

Appendix E: The original analysis pipeline as archived (verbatim listing)

Companion to Cieszewski (2026), MCFNS 18(1). The Python programs of the original project that produced the broad report (S1) and the final report, as archived in Appendix C (july_pipeline folder), unedited. They are listed for the documentary record of the workflow studied in the Article; the verification programs in Appendix D recompute every number of the Article independently of them.

Contents: 1. analyze.py (7,015 bytes); 2. analyze_ratios.py (1,180 bytes); 3. audit_check.py (2,020 bytes); 4. audit_check2.py (2,375 bytes); 5. audit_check3.py (1,743 bytes); 6. audit_extract.py (616 bytes); 7. build_docx.py (34,959 bytes); 8. build_docx2.py (29,972 bytes); 9. build_docx3.py (39,127 bytes); 10. build_docx4.py (39,170 bytes); 11. build_docx5.py (33,903 bytes); 12. build_docx6.bak.py (38,742 bytes); 13. build_docx6.py (98,279 bytes); 14. build_index.py (656 bytes); 15. build_report.py (9,340 bytes); 16. build_workbook2.py (8,133 bytes); 17. build_workbook3.py (17,415 bytes); 18. cc_defs.py (4,253 bytes); 19. chk_region.py (587 bytes); 20. compute_auth6.py (7,652 bytes); 21. compute_authorship.py (6,455 bytes); 22. compute_integrity.py (4,842 bytes); 23. compute_refs.py (3,673 bytes); 24. compute_refs2.py (3,047 bytes); 25. corr.py (3,051 bytes); 26. dedup_bettinger.py (26,954 bytes); 27. dedup_scholar.py (22,259 bytes); 28. diag_am_aq.py (2,646 bytes); 29. diag_singleauthor.py (1,787 bytes); 30. distinct.py (1,008 bytes); 31. dump_sheets.py (785 bytes); 32. explore2.py (1,472 bytes); 33. explore_forestry.py (1,681 bytes); 34. figs_auth6.py (16,525 bytes); 35. find_johnnash.py (1,126 bytes); 36. find_kahneman.py (1,048 bytes); 37. find_nash.py (883 bytes); 38. find_nash2.py (1,018 bytes); 39. find_refs.py (1,012 bytes); 40. find_uga.py (1,728 bytes); 41. inspect2.py (565 bytes); 42. inspect_cols.py (745 bytes); 43. make_figures.py (6,204 bytes); 44. mapping.py (62,311 bytes); 45. peek.py (506 bytes); 46. probe_final.py (3,975 bytes); 47. probe_head.py (2,792 bytes); 48. probe_nan.py (2,380 bytes); 49. process.py (5,185 bytes); 50. process_career.py (6,250 bytes); 51. ratios.py (6,197 bytes); 52. read_key.py (595 bytes); 53. recompute.py (3,108 bytes); 54. recompute_verify.py (9,531 bytes); 55. refine_data.py (7,755 bytes); 56. refine_explore.py (1,916 bytes); 57. refine_figures.py (12,108 bytes); 58. refine_figures2.bak.py (27,882 bytes); 59. refine_figures2.py (50,753 bytes); 60. refine_forestry.py (8,180 bytes); 61. refine_merged.py (6,320 bytes); 62. render_appx.py (726 bytes); 63. render_inst.py (619 bytes); 64. render_tabs.py (1,385 bytes); 65. report_data.py (3,310 bytes); 66. rootcause.py (894 bytes); 67. source_check.py (1,498 bytes); 68. split_batches.py (1,057 bytes); 69. uga_all.py (1,239 bytes); 70. uga_match.py (3,472 bytes); 71. validate_final.py (3,399 bytes); 72. validate_full.py (5,164 bytes); 73. validate_wb.py (1,662 bytes); 74. verify.py (4,811 bytes); 75. verify2.py (1,395 bytes); 76. verify_fields.py (1,255 bytes); 77. verify_final.py (1,727 bytes); 78. verify_fixes.py (2,576 bytes); 79. verify_forestry.py (2,855 bytes); 80. verify_merged.py (2,769 bytes); 81. verify_pdf.py (1,031 bytes); 82. verify_refine.py (2,509 bytes); 83. verify_report.py (2,590 bytes); 84. verify_restructure.py (2,837 bytes); 85. verify_sa.py (3,180 bytes); 86. xcheck.py (2,812 bytes)

1. analyze.py (SHA-256 85af7b422c503fc7...; 115 lines)

```
import pandas as pd, json
```

```
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
p = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df = pd.read_excel(p, sheet_name="Data")

# country code (ISO alpha-3, lowercase) -> (name, continent, UN sub-region)
CC = {
    "are": ("United Arab Emirates", "Asia", "Western Asia"), "arg": ("Argentina", "South America", "South America"),
    "aus": ("Australia", "Oceania", "Australia & New Zealand"), "aut": ("Austria", "Europe", "Western Europe"),
    "aze": ("Azerbaijan", "Asia", "Western Asia"), "bel": ("Belgium", "Europe", "Western Europe"),
    "bgd": ("Bangladesh", "Asia", "Southern Asia"), "bhr": ("Bahrain", "Asia", "Western Asia"),
    "bra": ("Brazil", "South America", "South America"), "brn": ("Brunei", "Asia", "South-eastern Asia"),
    "bwa": ("Botswana", "Africa", "Southern Africa"), "can": ("Canada", "North America", "Northern America"),
    "che": ("Switzerland", "Europe", "Western Europe"), "chl": ("Chile", "South America", "South America"),
    "chn": ("China", "Asia", "Eastern Asia"), "col": ("Colombia", "South America", "South America"),
    "cri": ("Costa Rica", "North America", "Central America"), "cuw": ("Curacao", "North America", "Caribbean"),
    "cze": ("Czechia", "Europe", "Eastern Europe"), "deu": ("Germany", "Europe", "Western Europe"),
    "dnk": ("Denmark", "Europe", "Northern Europe"), "dza": ("Algeria", "Africa", "Northern Africa"),
    "ecu": ("Ecuador", "South America", "South America"), "egy": ("Egypt", "Africa", "Northern Africa"),
```

```

"esp": ("Spain", "Europe", "Southern Europe"), "est": ("Estonia", "Europe", "Northern Europe"),
"eth": ("Ethiopia", "Africa", "Eastern Africa"), "fin": ("Finland", "Europe", "Northern Europe"),
"fra": ("France", "Europe", "Western Europe"), "gbr": ("United Kingdom", "Europe", "Northern Europe"),
"geo": ("Georgia", "Asia", "Western Asia"), "grc": ("Greece", "Europe", "Southern Europe"),
"grl": ("Greenland", "North America", "Northern America"), "hkg": ("Hong Kong", "Asia", "Eastern Asia"),
"hun": ("Hungary", "Europe", "Eastern Europe"), "idn": ("Indonesia", "Asia", "South-eastern Asia"),
"ind": ("India", "Asia", "Southern Asia"), "irl": ("Ireland", "Europe", "Northern Europe"),
"irn": ("Iran", "Asia", "Southern Asia"), "isr": ("Israel", "Asia", "Western Asia"),
"ita": ("Italy", "Europe", "Southern Europe"), "jor": ("Jordan", "Asia", "Western Asia"),
"jpn": ("Japan", "Asia", "Eastern Asia"), "ken": ("Kenya", "Africa", "Eastern Africa"),
"kor": ("South Korea", "Asia", "Eastern Asia"), "kwt": ("Kuwait", "Asia", "Western Asia"),
"ltu": ("Lithuania", "Europe", "Northern Europe"), "lva": ("Latvia", "Europe", "Northern Europe"),
"mex": ("Mexico", "North America", "Central America"), "mwi": ("Malawi", "Africa", "Eastern Africa"),
"mys": ("Malaysia", "Asia", "South-eastern Asia"), "nga": ("Nigeria", "Africa", "Western Africa"),
"nld": ("Netherlands", "Europe", "Western Europe"), "nor": ("Norway", "Europe", "Northern Europe"),
"npl": ("Nepal", "Asia", "Southern Asia"), "nzl": ("New Zealand", "Oceania", "Australia & New Zealand"),
"pak": ("Pakistan", "Asia", "Southern Asia"), "pan": ("Panama", "North America", "Central America"),
"per": ("Peru", "South America", "South America"), "phl": ("Philippines", "Asia", "South-eastern Asia"),
"pol": ("Poland", "Europe", "Eastern Europe"), "pri": ("Puerto Rico", "North America", "Caribbean"),
"prt": ("Portugal", "Europe", "Southern Europe"), "qat": ("Qatar", "Asia", "Western Asia"),
"rou": ("Romania", "Europe", "Eastern Europe"), "rus": ("Russia", "Europe", "Eastern Europe"),
"sau": ("Saudi Arabia", "Asia", "Western Asia"), "sgp": ("Singapore", "Asia", "South-eastern Asia"),
"srb": ("Serbia", "Europe", "Southern Europe"), "svk": ("Slovakia", "Europe", "Eastern Europe"),
"svn": ("Slovenia", "Europe", "Southern Europe"), "swe": ("Sweden", "Europe", "Northern Europe"),
"tha": ("Thailand", "Asia", "South-eastern Asia"), "tto": ("Trinidad and Tobago", "North America", "Caribbean"),
"tur": ("Turkey", "Asia", "Western Asia"), "twm": ("Taiwan", "Asia", "Eastern Asia"),
"uga": ("Uganda", "Africa", "Eastern Africa"), "ukr": ("Ukraine", "Europe", "Eastern Europe"),
"ury": ("Uruguay", "South America", "South America"), "usa": ("United States", "North America", "Northern America"),
"ven": ("Venezuela", "South America", "South America"), "zaf": ("South Africa", "Africa", "Southern Africa"),
"zmb": ("Zambia", "Africa", "Eastern Africa"),
}
missing = sorted(set(df["cntry"].dropna().unique()) - set(CC))
assert not missing, f"unmapped codes: {missing}"

df["country"] = df["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
df["continent"] = df["cntry"].map(lambda c: CC[c][1] if c in CC else "(unknown)")
df["region"] = df["cntry"].map(lambda c: CC[c][2] if c in CC else "(unknown)")
df["country"] = df["country"].fillna("(unknown)")
df["continent"] = df["continent"].fillna("(unknown)")
df["region"] = df["region"].fillna("(unknown)")

def field_of(m):
    m = str(m)
    has_f = "Forestry" in m; has_e = "Ecology" in m
    if has_f and has_e: return "Both"
    if has_f: return "Forestry"
    return "Ecology"
df["field"] = df["matched_subfield"].map(field_of)

YEARS = [2020, 2021, 2022, 2023, 2024]

def matrix(col, top=None):
    t = df.pivot_table(index=col, columns="year", values="authfull", aggfunc="count", fill_value=0)
    for y in YEARS:
        if y not in t.columns: t[y]=0
    t = t[YEARS]
    t["Total_allyears"] = t.sum(axis=1)
    t = t.sort_values(2024, ascending=False)
    if top: t = t.head(top)
    return t.reset_index()

out = {}
for dim in ["continent", "region", "country", "field"]:
    out[dim] = matrix(dim).to_dict(orient="records")
out["university_top60"] = matrix("inst_name", top=60).to_dict(orient="records")

# field x continent (2024) cross-tab
ct = df[df.year==2024].pivot_table(index="continent", columns="field", values="authfull", aggfunc="count",
fill_value=0)
ct["Total"] = ct.sum(axis=1)
out["field_by_continent_2024"] = ct.sort_values("Total", ascending=False).reset_index().to_dict(orient="records")

# headline
out["_meta"] = {
    "rows_per_year": {int(y): int((df.year==y).sum()) for y in YEARS},
    "distinct_countries": int(df["cntry"].nunique()),
    "distinct_institutions_2024": int(df[df.year==2024]["inst_name"].nunique()),
    "blank_cntry_rows": int(df["cntry"].isna().sum()),

```

```

    "blank_inst_rows": int(df["inst_name"].isna().sum()),
}
with open(f"{SCR}/rankings.json", "w", encoding="utf-8") as f:
    json.dump(out, f, indent=1, default=int)

# dump subnational institution lists for the mapping workflow (US, Canada, Australia)
for code, tag in [("usa", "us"), ("can", "can"), ("aus", "aus")]:
    sub = df[df.cntry==code]
    ic = sub.groupby("inst_name").size().sort_values(ascending=False)
    ic.to_csv(f"{SCR}/inst_{tag}.csv", header=["n"])
    print(f"{tag}: {len(ic)} distinct institutions, {int(ic.sum())} rows")

print("\n2024 continents:")
print(df[df.year==2024]["continent"].value_counts().to_string())
print("\n2024 top 12 countries:")
print(df[df.year==2024]["country"].value_counts().head(12).to_string())
print("\nfield totals (all years):", df["field"].value_counts().to_dict())
print("rankings.json + inst lists written")

```

2. analyze_ratios.py (SHA-256 836b7f60ad59779a...; 18 lines)

```

# -*- coding: utf-8 -*-
import json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
M=json.load(open(f"{SCR}/merged_data.json", encoding="utf-8"))
MF=M["forestry"]
rows=[]
for r in MF["country"]:
    S=sum(r["s"]); P=sum(r["p"]); C=sum(r["c"])
    if S<25: continue # >=5 scientists/yr for stable ratios
    rows.append((r["key"], S/5, P/S, C/S, C/P)) # avg scientists/yr, pubs/sci, cites/sci, cites/pub
print(f"{'country':<22}{'sci/yr':>7}{'pubs/sci':>10}{'cites/sci':>11}{'cites/pub':>11}")
for k, sy, pps, cps, cpp in sorted(rows, key=lambda x: -x[1]):
    print(f"{k:<22}{sy:>7.0f}{pps:>10.1f}{cps:>11.0f}{cpp:>11.1f}")
print("\n-- extremes among these countries --")
print("highest cites/pub:", sorted(rows, key=lambda x: -x[4])[5] and [(r[0], round(r[4], 1)) for r in sorted(rows, key=lambda x: -x[4])[5]])
print("lowest cites/pub:", [(r[0], round(r[4], 1)) for r in sorted(rows, key=lambda x: x[4])[5]])
print("highest pubs/sci :", [(r[0], round(r[2], 1)) for r in sorted(rows, key=lambda x: -x[2])[5]])
print("highest cites/sci:", [(r[0], round(r[3], 0)) for r in sorted(rows, key=lambda x: -x[3])[5]])

```

3. audit_check.py (SHA-256 e41916e9623b5ce1...; 48 lines)

```

import re

txt = open(r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\pdf_fulltext.txt", encoding="utf-8").read()

# Split into pages
pages = {}
for m in re.finditer(r"===== PAGE (\d+) =====\n(?:.?(?===== PAGE|\Z))", txt, re.S):
    pages[int(m.group(1))] = m.group(2)

def find_page(substr):
    hits = []
    for p, t in pages.items():
        if substr in t:
            hits.append(p)
    return hits

print("=== (1) NUMBERING CONTINUITY ===")
# Count table/figure captions
for label in ["Table B1", "Table B2", "Table B3", "Table B4", "Table C1", "Table C2", "Table C3", "Table C4",
              "Figure B1", "Figure B2", "Figure B3", "Figure B4", "Figure B5",
              "Figure C1", "Figure C2", "Figure C3", "Figure C4", "Figure C5"]:
    # Count as caption: label followed by . or space then capital
    cnt = len(re.findall(re.escape(label)+r"[\.\s]", txt))
    pgs = find_page(label)
    print(f"{label}: count={cnt} pages={pgs}")

print("\n=== Check for gaps: B6/C6, Table B5/C5 (should NOT exist) ===")
for label in ["Table B5", "Table C5", "Figure B6", "Figure C6", "Table B0", "Figure B0"]:
    cnt = len(re.findall(re.escape(label)+r"[\.\s]", txt))
    print(f"{label}: count={cnt}")

```

```

print("\n=== (2) SINGLE-AUTHOR METRIC UNITS in B/C ===")
# All-publications leftover units
for bad in ["Publications ('000)", "Citations (M)", "Publications (career papers)"]:
    pgs = find_page(bad)
    print(f"LEFTOVER-CHECK {bad!r}: pages={pgs}")

print("\n=== (7) LEFTOVER BAND LANGUAGE ===")
for bad in ["~10% bands", "10% bands", "top 20 individual", "21-100", "21-100", "band", "Band"]:
    pgs = find_page(bad)
    print(f"BAND-CHECK {bad!r}: pages={pgs}")

print("\n=== (3) top 25 (ranked by scientists) in country/inst tables ===")
for label in ["Table B3", "Table B4", "Table C3", "Table C4"]:
    # find the caption block
    idx = txt.find(label)
    block = txt[idx: idx+400].replace("\n", " ")
    print(f"{label}: {block[:260]}")

```

4. audit_check2.py (SHA-256 85fadbf29485c1e1...; 54 lines)

```

import re

txt = open(r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\pdf_fulltext.txt", encoding="utf-8").read()

pages = {}
for m in re.finditer(r"===== PAGE (\d+) =====\n(?:.?)?(?===== PAGE|\Z)", txt, re.S):
    pages[int(m.group(1))] = m.group(2)

print("=== 'band' context on pages 1,3,17,22 ===")
for p in [1,3,17,22]:
    for m in re.finditer(r".{40}band.{40}", pages[p], re.I):
        print(f"  p{p}: ...{m.group(0)}...")

print("\n=== (2b) Metric row labels used INSIDE B/C tables ===")
# In appendix B/C, the metric rows should be single-author labels
for label in ["Single-author papers", "Cites to SA papers", "Single-author papers (nps)", "Citations to single-author papers (ncs)", "Scientists (with single-author papers)"]:
    cnt = len(re.findall(re.escape(label), txt))
    print(f"  {label!r}: total occurrences={cnt}")

# Make sure no 'Publications' or 'Citations (M)' style rows appear on B/C pages
print("\n=== Scan B/C pages (31-46) for forbidden all-pub unit strings ===")
bc_pages = range(31,47)
for p in bc_pages:
    for bad in ["Publications ('000)", "Citations (M)", "Publications (career papers)", "Citations (career citations)"]:
        if bad in pages[p]:
            print(f"  FORBIDDEN {bad!r} on page {p}")
print("  (nothing printed above = clean)")

print("\n=== (4) Appendix titles ===")
for m in re.finditer(r"Appendix [ABC]\.^[^\n]*", txt):
    print("  ", m.group(0))

print("\n=== (5) Intro fractions & nps/ncs defs ===")
# Appendix B intro
bidx = txt.find("Appendix B. The Forestry Group")
bintro = txt[bidx: bidx+1300].replace("\n", " ")
print("  B-INTRO:", bintro[:1100])
print()
cidx = txt.find("Appendix C. The Ecology Group")
cintro = txt[cidx: cidx+1300].replace("\n", " ")
print("  C-INTRO:", cintro[:1100])

print("\n=== fraction strings ===")
for s in ["3,129 of the 3,271", "3,129 of 3,271", "615 of the 615", "615 of 615", "(96%)", "(100%)"]:
    print(f"  {s!r}: pages=[p for p in pages if s in pages[p]]")

print("\n=== (6) Abstract & Project Summary mention App B and C ===")
absidx = txt.find("Abstract")
absblk = txt[absidx: absidx+1600].replace("\n", " ")
print("  ABSTRACT tail:", absblk[-700:])

```

```

print()
psidx = txt.find("Project Summary")
psblk = txt[psidx: psidx+1200].replace("\n", " ")
print("  PROJECT SUMMARY:", psblk[:1000])

```

5. audit_check3.py (SHA-256 0a4b995daa27ab4c...; 38 lines)

```

import re
txt = open(r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\pdf_fulltext.txt", encoding="utf-8").read()
pages = {}
for m in re.finditer(r"===== PAGE (\d+) =====\n(?:.?)?(?===== PAGE|\Z)", txt, re.S):
    pages[int(m.group(1))] = m.group(2)

print("=== Full Abstract B/C sentence ===")
i = txt.find("Appendix A profiles the Ecology group")
print(txt[i:i+500].replace("\n", " "))

print("\n=== Every 'band' occurrence in whole doc (context) ===")
for m in re.finditer(r".{25}band\w*. {15}", txt, re.I):
    print("  ...", m.group(0).replace("\n", " "), "...")

print("\n=== B1 / C1 totals table caption + row labels ===")
for lab in ["Table B1", "Table C1"]:
    i = txt.find(lab)
    print(f"\n{lab}: \n", txt[i:i+430].replace("\n", " | "))

print("\n=== B2/C2 continent + B4/C4 institution caption full ===")
for lab in ["Table B2", "Table C2"]:
    i = txt.find(lab)
    print(f"\n{lab}: \n", txt[i:i+230].replace("\n", " "))

print("\n=== Figure B/C caption texts (verify single-author wording) ===")
for lab in ["Figure B1", "Figure B2", "Figure B3", "Figure B4", "Figure B5",
            "Figure C1", "Figure C2", "Figure C3", "Figure C4", "Figure C5"]:
    i = txt.find(lab)
    seg = txt[i:i+230].replace("\n", " ")
    print(f"  {seg}")

print("\n=== Confirm 'single-author' / 'single-authored' appears in EVERY B/C fig caption ===")
for lab in ["Figure B1", "Figure B2", "Figure B3", "Figure B4", "Figure B5",
            "Figure C1", "Figure C2", "Figure C3", "Figure C4", "Figure C5"]:
    i = txt.find(lab)
    seg = txt[i:i+230].lower()
    ok = ("single-author" in seg) or ("single-authored" in seg)
    print(f"  {lab}: single-author mention = {ok}")

```

6. audit_extract.py (SHA-256 640248c169ce6c52...; 17 lines)

```

import fitz
import json

doc = fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
print("PAGE COUNT:", doc.page_count)
out = []
for i, page in enumerate(doc):
    text = page.get_text()
    out.append((i+1, text))

# Dump full text with page markers
with open(r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\pdf_fulltext.txt", "w", encoding="utf-8") as f:
    for pageno, text in out:
        f.write(f"\n===== PAGE {pageno} =====\n")
        f.write(text)

print("WROTE fulltext")

```

7. build_docx.py (SHA-256 f03359566d6e5ef9...; 457 lines)

```

# -*- coding: utf-8 -*-
import json

```

```

from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.enum.section import WD_SECTION
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_Representation_Report.docx"
YEARS = [2020,2021,2022,2023,2024]
GREEN = "2F5233"; LIGHT = "E7EEE3"
rk = json.load(open(f"{SCR}/rankings.json", encoding="utf-8"))
D = json.load(open(f"{SCR}/report_data.json", encoding="utf-8"))

def gy(rec, y):
    for k in (str(y), y):
        if k in rec: return rec[k]
    return 0

doc = Document()
# ---- base styles ----
st = doc.styles["Normal"]; st.font.name = "Arial"; st.font.size = Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"), "Arial")
for i,(sz,col) in enumerate([(16, GREEN), (13, GREEN), (11, "333333")], start=1):
    h = doc.styles[f"Heading {i}"]
    h.font.name = "Arial"; h.font.size = Pt(sz); h.font.bold = True
    h.font.color.rgb = RGBColor.from_string(col)
doc.styles["Title"].font.name = "Arial"

sec = doc.sections[0]
sec.page_width, sec.page_height = Inches(8.5), Inches(11)
sec.top_margin = sec.bottom_margin = Inches(1); sec.left_margin = sec.right_margin = Inches(1)
CW = Inches(6.5)

def p(text="", size=11, bold=False, italic=False, align=None, color=None, after=6, before=0, style=None):
    par = doc.add_paragraph(style=style); r = par.add_run(text)
    r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold; r.italic=italic
    if color: r.font.color.rgb = RGBColor.from_string(color)
    if align: par.alignment = align
    par.paragraph_format.space_after = Pt(after); par.paragraph_format.space_before = Pt(before)
    return par

def body(text, size=11, after=8, justify=True):
    par = doc.add_paragraph(); r = par.add_run(text); r.font.name="Arial"; r.font.size=Pt(size)
    if justify: par.alignment = WD_ALIGN_PARAGRAPH.JUSTIFY
    par.paragraph_format.space_after = Pt(after); par.paragraph_format.line_spacing = 1.15
    return par

def bullet(text, size=11):
    par = doc.add_paragraph(style="List Bullet"); r = par.add_run(text)
    r.font.name="Arial"; r.font.size=Pt(size); par.paragraph_format.space_after = Pt(3)
    return par

def shade(cell, hexc):
    tcPr = cell._tc.get_or_add_tcPr(); sh = OxmlElement("w:shd")
    sh.set(qn("w:val"), "clear"); sh.set(qn("w:fill"), hexc); tcPr.append(sh)

def set_cell(cell, text, bold=False, white=False, size=9, align="left"):
    cell.text = ""; par = cell.paragraphs[0]
    par.alignment =
{"left":WD_ALIGN_PARAGRAPH.LEFT,"center":WD_ALIGN_PARAGRAPH.CENTER,"right":WD_ALIGN_PARAGRAPH.RIGHT}[align]
    r = par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb = RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after = Pt(1); par.paragraph_format.space_before = Pt(1)

def make_table(headers, rows, widths, aligns=None, caption=None, zebra=True):
    if caption: p(caption, size=9, italic=True, color="555555", after=3)
    t = doc.add_table(rows=1, cols=len(headers)); t.alignment = WD_TABLE_ALIGNMENT.CENTER
    t.autofit = False; t.allow_autofit = False
    t.style = "Table Grid"
    aligns = aligns or (["left"] + ["right"]*(len(headers)-1))
    for j, htext in enumerate(headers):
        set_cell(t.rows[0].cells[j], htext, bold=True, white=True, size=9, align="center")
        shade(t.rows[0].cells[j], GREEN)
    for i, row in enumerate(rows):

```

```

        cells = t.add_row().cells
        for j, val in enumerate(row):
            set_cell(cells[j], val, size=9, align=aligns[j])
            if zebra and i%2==1: shade(cells[j], LIGHT)
        for col, w in zip(t.columns, widths):
            for c in col.cells: c.width = w
        doc.add_paragraph().paragraph_format.space_after = Pt(2)
        return t

def figure(path, width, caption):
    par = doc.add_paragraph(); par.alignment = WD_ALIGN_PARAGRAPH.CENTER
    par.add_run().add_picture(path, width=width)
    par.paragraph_format.space_before = Pt(6); par.paragraph_format.space_after = Pt(2)
    p(caption, size=9, italic=True, color="555555", align=WD_ALIGN_PARAGRAPH.CENTER, after=10)

def footer_pagenums():
    ftr = sec.footer; par = ftr.paragraphs[0]; par.alignment = WD_ALIGN_PARAGRAPH.CENTER
    run = par.add_run("Page "); run.font.name="Arial"; run.font.size=Pt(8)
    for t, val in [ ("begin", None), ("instr", "PAGE"), ("end", None) ]:
        fld = XmlElement("w:fldChar") if t!="instr" else XmlElement("w:instrText")
        if t=="instr": fld.set(qn("xml:space"), "preserve"); fld.text=" PAGE "
        else: fld.set(qn("w:fldCharType"), val if False else t.replace("begin", "begin").replace("end", "end"))
        r = par.add_run(); r._r.append(fld); r.font.name="Arial"; r.font.size=Pt(8)
    run2 = par.add_run(" of "); run2.font.name="Arial"; run2.font.size=Pt(8)
    for t in ["begin", "instr", "end"]:
        if t=="instr":
            fld=XmlElement("w:instrText"); fld.set(qn("xml:space"), "preserve"); fld.text=" NUMPAGES "
        else:
            fld=XmlElement("w:fldChar"); fld.set(qn("w:fldCharType"), t)
            r=par.add_run(); r._r.append(fld); r.font.name="Arial"; r.font.size=Pt(8)
    footer_pagenums()

# ===== TITLE PAGE =====
for _ in range(2): doc.add_paragraph()
p("Global and Subnational Representation of Highly Cited Forestry and Ecology Researchers",
  size=20, bold=True, align=WD_ALIGN_PARAGRAPH.CENTER, color=GREEN, after=6)
p("A Five-Year Analysis (2020-2024) Of the Stanford / Ioannidis Career-Long Citation Database",
  size=13, italic=True, align=WD_ALIGN_PARAGRAPH.CENTER, color="555555", after=24)
p("Analytical Report", size=12, bold=True, align=WD_ALIGN_PARAGRAPH.CENTER, after=2)
p("Prepared 5 July 2026", size=11, align=WD_ALIGN_PARAGRAPH.CENTER, color="555555", after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3-v8)",
  size=9, align=WD_ALIGN_PARAGRAPH.CENTER, color="777777", after=24)

# Abstract
p("Abstract", size=13, bold=True, color=GREEN, after=4)
body("This report quantifies the geographic and institutional representation of the world's most highly cited "
  "forestry and ecology researchers, using the career-long author tables of the widely used Stanford / Ioannidis "
  "'top-scientists' citation database. Five annual releases (data years 2020-2024) were retrieved, and "
  "23,374 author-year records classified by the Science-Metrix scheme into the Forestry subfield (agriculture
domain) "
  "or the Ecology subfield (biology domain) were extracted and combined. Each record was resolved to a continent, "
  "
  "world sub-region and country from the author's current institutional affiliation, and records in the three "
  "largest federal countries (United States, Canada, Australia) were further resolved to state or province by "
  "geolocating each institution's main campus. The list is strongly concentrated in the Global North: in the 2024 "
  "
  "release, North America (46.9%) and Europe (35.4%) together account for more than four-fifths of all entries,
led by "
  "the United States (1,917 entries), the United Kingdom (426), Canada (417) and Australia (348). Within the
United "
  "States, California is the single largest source of entries, and a distributed block of federal science agencies "
  "
  "(led by the USDA Forest Service and U.S. Geological Survey) ranks second. The Swedish University of
Agricultural "
  "Sciences is the most represented single institution worldwide. Ecology researchers outnumber forestry
researchers "
  "roughly five to one throughout the period. While absolute counts are dominated by established scientific
powers, the "
  "fastest five-year growth is seen in China (+92%), Brazil (+48%) and several mid-sized European and Southern-
Hemisphere "
  "countries, whereas the United States and United Kingdom grew only single digits, hinting at a gradual
geographic "
  "diversification of the field. The nationality of authors and any qualitative-quantitative field split could not "
  "
  "be derived from the source data and are therefore not reported.", size=10, after=6)

# Keywords
kw = doc.add_paragraph(); r0 = kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)
r1 = kw.add_run("bibliometrics; scientometrics; citation metrics; composite citation indicator (c-score); ")

```

```

"Stanford/Ioannidis top-2% list; highly cited researchers; forestry; ecology; research geography; "
"Science-Metrix classification; Scopus; choropleth mapping.")
r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True

doc.add_paragraph().add_run().add_break() # page break to body
doc.paragraphs[-1].runs[0]._r.getparent().getparent() # noop
from docx.enum.text import WD_BREAK
# ensure a clean page break
pb = doc.add_paragraph(); pb.add_run().add_break(WD_BREAK.PAGE)

# ===== 1. INTRODUCTION / BACKGROUND =====
doc.add_heading("1. Background, Concept and Related Work", level=1)
body("Quantifying individual scientific impact through citation data has become routine in research evaluation, "
"  hiring, promotion and funding. The single most cited summary statistic, the h-index [1], is known to conflate "
"  productivity with impact, to scale with career length and field size, and to be inflated by self-citation and "
by "
"large co-author lists. To address these distortions, Ioannidis and colleagues proposed a standardized composite "
"
"citation indicator (the "c-score") that combines six complementary metrics-total citations; the h-index; "
"the co-authorship-adjusted Schreiber hm-index; and citations to papers as single author, as single-or-first "
author, "
"and as single-first-or-last author-each normalized within a paper's scientific field [2,3].")
body("Building on this indicator, the authors released a publicly available database that ranks roughly the top "
"  100,000 scientists (and all scientists in the top 2% of their subfield) across 22 scientific fields and 174 "
"  subfields, using a classification derived from the Science-Metrix journal ontology [3,4]. The database is "
published "
"in two forms: a "career-long" table, in which metrics are cumulative over a scientist's entire "
"publication record, and a "single-year" table, in which metrics reflect citations received in one calendar "
"year. Since its first release the database has been updated annually from a Scopus snapshot and has been "
downloaded "
"and cited many thousands of times; it is now one of the most widely used author-level bibliometric resources. "
This "
"report uses the career-long tables.")
body("Similar large-scale rankings exist but differ in unit of analysis and transparency. Institution-level "
rankings "
"such as the Academic Ranking of World Universities (Shanghai), the Times Higher Education World University "
Rankings, "
"and the Nature Index rank organizations rather than individuals. Clarivate's Highly Cited Researchers list "
"identifies influential authors but is proprietary and based on Essential Science Indicators "highly cited "
"papers." The Ioannidis database is distinctive in being author-level, field-normalized, openly licensed, and "
"reproducible from documented code, which makes it particularly suitable for secondary geographic analysis such "
as "
"the present study. Prior bibliometric work has repeatedly documented a concentration of scientific output in "
the "
"Global North; here we test and quantify that pattern specifically within forestry and ecology-two fields of "
"direct relevance to biodiversity, land management and climate policy-and add a subnational dimension that is "
"rarely reported.")

# ===== 2. DATA =====
doc.add_heading("2. Data Description", level=1)
body("The primary data are the five most recent career-long author tables (file prefix "
""Table_1_Authors_career") distributed through the dataset landing page at "
"elsevier.digitalcommonsdata.com/datasets/btchxktzyw. Each release is a versioned deposit in the Elsevier Data "
Repository (Mendeley Data) and corresponds to one data year, as summarized in Table 1. Each file is a large "
spreadsheet (~75-85 MB) containing one row per ranked scientist (about 186,000-230,000 rows) and 46-47 "
columns, together with a legend sheet. Fields include the author name, a single large-institution affiliation, "
the "
"ISO country code of that affiliation, first and last publication years, the six component citation metrics and "
the "
"composite c-score (reported both with and without self-citations), and the Science-Metrix classification: a "
primary "
"subfield (sm-subfield-1), an optional secondary subfield (sm-subfield-2), and the parent field (sm-field), each "
with "
"a fractional weight.")
body("From every release we extracted all rows whose primary or secondary subfield was Forestry or Ecology, added "
a "
"column recording the release year and which subfield matched, and concatenated the five releases into a single "
"
"combined database of 23,374 author-year records. Because year-specific columns are named differently across "
releases "
"(for example nc9620 versus nc9624 for career citations, or h20 versus h24 for the h-index), column names were "
"normalized to a common schema before stacking. The combined database is the analytical basis for this report "
and is "
"delivered alongside it as an Excel workbook and CSV file.")
make_table(
["Release (version)","Published","Data year","Scientists in file","Forestry+Ecology extracted"],
[["v3","19 Oct 2021","2020","186,177","4,344"],
["v5","03 Nov 2022","2021","194,983","4,516"],
["v6","04 Oct 2023","2022","204,643","4,646"],
["v7","16 Sep 2024","2023","217,097","4,825"],

```

```

["v8","19 Sep 2025","2024","230,333","5,043"]],
[Inches(1.3),Inches(1.2),Inches(1.0),Inches(1.6),Inches(1.9)],
aligns=["center","center","center","right","right"],
caption="Table 1. Source releases of the career-long database and records extracted for this study. Version 4 (a
"
"superseded 2021 release) was replaced by the corrected version 5. Each release reports citations accrued
through the "
"end of its data year; as of the August 2025 update no 2025 data-year table exists.")

# ===== 3. METHODS =====
doc.add_heading("3. Methods", level=1)
body("Retrieval and filtering. The dataset's public API was used to enumerate versions and obtain direct download
"
"URLs for the career tables of versions 3, 5, 6, 7 and 8. Files were processed in Python with the pandas and
openpyxl "
"libraries. A record was retained when either subfield field equalled "Forestry" or "Ecology" "
"(case-insensitive). Records matching both target subfields were flagged as "Both."")
body("Category scope. The analysis was scoped in consultation with the requester. "Forestry" maps directly to "
"the Science-Metrix Forestry subfield. "Life sciences" was operationalized narrowly as the Ecology subfield "
"only, explicitly excluding the broader Biology and Biomedical Research fields. A third requested theme
("natural "
"resource management") has no counterpart among the 174 Science-Metrix subfields—verified against every "
"release's classification table—and was therefore not included.")
body("Geographic assignment. Continent, United Nations M49 world sub-region and country were derived
deterministically "
"from the three-letter ISO affiliation code. For the three large federal countries in which subnational
geography is "
"meaningful—the United States, Canada and Australia—each distinct institution was additionally resolved to a "
"first-level administrative unit (state, province or territory) by geolocating the main campus or headquarters.
This "
"geolocation used a fan-out of language-model agents (twelve parallel workers over batched institution lists,
plus an "
"independent verification pass over the highest-weight institutions, which agreed on all of them). Distributed
national "
"agencies with no single dominant campus (for example the USDA Forest Service, the U.S. Geological Survey and
CSIRO) "
"were labelled "Federal / multi-site" and reported as a distinct category; institutions that could not be "
"confidently placed were labelled "Unknown."")
body("Unit of analysis. Because career metrics are cumulative and each release re-lists largely the same
scientists, "
"the natural counting unit is the author-year: a scientist is counted once for every release in which they
appear. "
"Counts are therefore reported per release year, and the 2024 release is used as the headline snapshot. Across
the "
"five releases the combined database contains 8,425 distinct author names, of whom 1,939 appear in all five
releases. "
"No cross-year de-duplication was applied to the ranking tables.")
body("Not derivable. The source records only the author's current institutional country; it contains no field for
"
"nationality, country of birth, or country of degree, and these cannot be reliably inferred. Nationality is
therefore "
"not reported. Likewise, both fields analysed here (Forestry and Ecology) are quantitative natural-science
fields, so a "
"quantitative-versus-qualitative contrast does not apply; a Forestry-versus-Ecology split is provided in its
place.")
body("Verification and visualization. All extraction counts were independently recomputed with an alternative "
"implementation, and every ranking table was reconciled against the known per-release totals. Figures were
produced "
"with matplotlib; choropleth maps use public-domain Natural Earth country boundaries [6] and a standard US-state
"
"boundary file, with colour intensity encoding the 2024 entry count on a graduated scale.")

# ===== 4. RESULTS =====
doc.add_heading("4. Results", level=1)

doc.add_heading("4.1 Field composition and growth over time", level=2)
fld = {r["field"]: r for r in rk["field"]}
eco24, for24, both24 = gy(fld["Ecology"],2024), gy(fld["Forestry"],2024), gy(fld["Both"],2024)
body(f"The combined database is dominated by ecology. In the 2024 release, {eco24:,} entries
({eco24/5043*100:.0f}%) "
f"were classified in Ecology, {for24:,} ({for24/5043*100:.0f}%) in Forestry, and {both24} ({both24/5043*100:.0f}
%) in "
"both subfields; the same roughly five-to-one ratio holds across all five years (Figure 1, Table 6). The list
grew "
"steadily from 4,344 entries in 2020 to 5,043 in 2024, a 16% increase over four years, with ecology and forestry
"
"expanding in step.")
figure(f"{SCR}/fig1_trend.png", Inches(5.6),
"Figure 1. Forestry and ecology entries in the career-long list by release year (stacked). "Both" denotes "
"authors whose primary and secondary subfields are both target subfields.")

```

```

doc.add_heading("4.2 Representation by continent and world region", level=2)
body("Representation is heavily concentrated in the Global North (Table 2, Figure 2). In 2024, North America and
"
"Europe together supplied 82% of all entries; Oceania-driven almost entirely by Australia and New Zealand-is "
"the third-ranked continent with a share far above its population, reflecting the strength of ecology and
forestry "
"research in those countries. Asia, South America and Africa together account for barely 9% of the list despite
"
"containing most of the world's forests and biodiversity.")
cont_rows = []
for r in rk["continent"]:
    if r["continent"]=="(unknown)": continue
    v24=gy(r,2024); cont_rows.append([r["continent"]]+[f"{gy(r,y):,}" for y in YEARS]+[f"{v24/5041*100:.1f}%"])
make_table(["Continent"]+[str(y) for y in YEARS]+["Share 2024"], cont_rows,
    [Inches(1.7)]+[Inches(0.82)]*5+[Inches(0.9)],
    aligns=["left"]+["right"]*6,
    caption="Table 2. Entries by continent and release year, with share of the 2024 release. (Two 2024 records had
no "
"resolvable country and are omitted from shares.)")
figure(f"{SCR}/fig2_worldmap.png", Inches(6.3),
"Figure 2. Global choropleth of 2024 career-list entries by country of affiliation. Darker greens indicate more
"
"highly cited forestry/ecology researchers; grey indicates no entries. Boundaries: Natural Earth [6].")

doc.add_heading("4.3 Representation by country", level=2)
usa24 = gy({r["country"]:r for r in rk["country"]})["United States"],2024)
body(f"The United States alone accounts for {usa24:,} entries-{usa24/5043*100:.0f}% of the 2024 list and more
than "
"four times the second-placed United Kingdom (Table 3, Figure 3). The United Kingdom, Canada and Australia form
a "
"close-packed second tier, followed by Germany and, notably, the Nordic countries: Sweden and Finland rank far
above "
"their population weight, a signature of their strong agricultural-university and forest-science traditions. The
"
"temporal picture (Figure 6) is more dynamic than the static ranking suggests: among countries with at least 20
"
"entries in 2020, China grew fastest (+92%), followed by Brazil (+48%), Italy (+44%) and South Africa (+40%),
whereas "
"the United States (+9%) and United Kingdom (+10%) grew only marginally. The list is thus dominated in absolute
terms "
"by the established powers but is slowly diversifying toward Asia, Latin America and the Southern Hemisphere.")
cmap = {r["country"]:r for r in rk["country"]}
g = D["country_growth_pct"]
ctry_rows = []
for r in rk["country"][:20]:
    nm=r["country"]; gr = g.get(nm)
    ctry_rows.append([nm, f"{gy(r,2020):,}", f"{gy(r,2024):,}", (f"{gr:+.0f}%" if gr is not None else "-")])
make_table(["Country","2020","2024","Growth 20-24"], ctry_rows,
    [Inches(2.6),Inches(1.1),Inches(1.1),Inches(1.4)], aligns=["left","right","right","right"],
    caption="Table 3. Top 20 countries by 2024 entries, with 2020 counts and four-year growth (growth shown for
countries "
"with ≥20 entries in 2020).")
figure(f"{SCR}/fig3_countries.png", Inches(5.8),
"Figure 3. Top 15 Countries by number of career-list entries in the 2024 release.")

doc.add_heading("4.4 Subnational representation: United States, Canada and Australia", level=2)
us = D["us_state"]; usm={r["subdivision"]:r for r in us}
ca = D["ca_prov"]; au = D["au_state"]
ca_top = [r for r in ca if r["subdivision"] not in ("Unknown","Federal/Multi-site")][:2]
au_top = [r for r in au if r["subdivision"] not in ("Unknown","Federal/Multi-site")][:2]
body(f"Within the United States (Table 4, Figure 4), California is by a wide margin the leading state "
f"({gy(usm['California'],2024)} entries in 2024), reflecting the University of California system and other large
"
"research universities. The second-ranked category is not a state at all but the distributed block of federal
science "
f"agencies ({gy(usm['Federal/Multi-site'],2024)} entries), underscoring the outsized role of the USDA Forest
Service, "
"U.S. Geological Survey and USDA Agricultural Research Service in American forestry and ecology. New York, North
"
"Carolina, Colorado, Oregon and Florida follow. About 6% of U.S. entries are at institutions that could not be
placed "
"to a state and are shown as "Unknown." In Canada, "
f"{ca_top[0]['subdivision']} and {ca_top[1]['subdivision']} lead; in Australia, "
f"{au_top[0]['subdivision']} and {au_top[1]['subdivision']} lead (Table 5).")
us_rows = []
for r in us[:12]:
    us_rows.append([r["subdivision"]]+[f"{gy(r,y)}" for y in YEARS])
make_table(["US state / category"]+[str(y) for y in YEARS], us_rows,
    [Inches(2.3)]+[Inches(0.84)]*5, aligns=["left"]+["right"]*5,

```

```

caption="Table 4. Top 12 United States states and categories by 2024 entries. "Federal / multi-site" = "
distributed national agencies; "Unknown" = institutions not placeable to a single state.")
figure(f"{SCR}/fig4_usmap.png", Inches(6.1),
"Figure 4. Choropleth of 2024 U.S. entries by state (contiguous states shown; Alaska 17 and Hawaii 16 entries
omitted "
"from the frame). Federal multi-site and unplaceable institutions are excluded from the map.")
ca_rows = [[r["subdivision"], f"{gy(r,2024)}"] for r in ca[:8]]
au_rows = [[r["subdivision"], f"{gy(r,2024)}"] for r in au[:8]]
merged = []
for i in range(max(len(ca_rows),len(au_rows))):
    left = ca_rows[i] if i<len(ca_rows) else ["",""]
    right= au_rows[i] if i<len(au_rows) else ["",""]
    merged.append(left+right)
make_table(["Canada province/terr.,"2024","Australia state/terr.,"2024"], merged,
[Inches(2.2),Inches(0.9),Inches(2.2),Inches(0.9)], aligns=["left","right","left","right"],
caption="Table 5. Leading Canadian provinces/territories and Australian states/territories by 2024 entries.")

doc.add_heading("4.5 Representation by institution", level=2)
uni = rk["university_top60"]
top_uni = uni[0]["inst_name"]; top_v = gy(uni[0],2024)
body(f"At the institutional level (Table 6, Figure 5), the single most represented organization in 2024 is the "
f"{top_uni} ({top_v} entries)–the Swedish University of Agricultural Sciences–ahead of the University of "
"British Columbia, the University of California, Davis, Oregon State University and Wageningen University &
Research. "
"The prominence of specialized agricultural and forestry universities (Sweden’s SLU, Wageningen, Oregon State) "
"shows how a single mission-focused institution can outrank the aggregate output of entire mid-sized countries.
U.S. "
"federal research bodies such as the USDA Forest Service also appear among the very top institutions.")
uni_rows = []
for r in uni[:20]:
    nm=r["inst_name"]; nm = nm if len(nm)<=46 else nm[:44]+"..."
    uni_rows.append([nm]+[f"{gy(r,y)}" for y in (2020,2024)]+[f"{gy(r,'Total_allyears')} or sum(gy(r,y) for y in
YEARS)}"]])
make_table(["Institution","2020","2024","All yrs"], uni_rows,
[Inches(3.6),Inches(0.9),Inches(0.9),Inches(1.0)], aligns=["left","right","right","right"],
caption="Table 6. Top 20 institutions worldwide by 2024 entries (career-long list). "All yrs" sums the five "
"releases. Only large institutions are individually named in the source data.")
figure(f"{SCR}/fig5_universities.png", Inches(5.9),
"Figure 5. Top 15 institutions by number of career-list entries in the 2024 release.")

doc.add_heading("4.6 Forestry versus ecology by year", level=2)
fr_rows = []
for r in rk["field"]:
    fr_rows.append([r["field"]]+[f"{gy(r,y):,}" for y in YEARS]+[f"{gy(r,'Total_allyears'):,}"])
make_table(["Field (subfield match)"+[str(y) for y in YEARS]+["All yrs"], fr_rows,
[Inches(1.9)]+[Inches(0.8)]*5+[Inches(0.9)], aligns=["left"]+[f"right"]*6,
caption="Table 6b. Entries by matched subfield and release year. "Both" = authors whose primary and "
"secondary subfields are both Forestry and Ecology.")
figure(f"{SCR}/fig6_growth.png", Inches(5.9),
"Figure 6. Four-year change (2020–2024) in entries for countries with at least 20 entries in 2020. Red bars mark
"
"the two largest established producers (United States, United Kingdom), which grew only single digits.")

# ===== 5. DISCUSSION =====
doc.add_heading("5. Discussion", level=1)
body("Why these data are interesting. The geography of highly cited researchers is a proxy for the geography of "
scientific capacity, prestige and resources. Mapping it for forestry and ecology is especially pointed because
the "
"subject matter–forests, biodiversity, ecosystems–is globally distributed and concentrated in the tropics, "
"yet the recognized expertise, as measured here, is concentrated in temperate, wealthy, English-publishing
countries. "
"The contrast between where forests are and where the most-cited forest scientists are affiliated is a
quantitative "
"entry point into long-standing debates about equity, "parachute" research, and capacity-building in "
"environmental science.")
body("Potential uses. The combined database and its rankings can support several concrete tasks: (i) research-
policy "
"and funding agencies can benchmark national or regional standing and track it over time; (ii) universities and
"
"research institutes can benchmark themselves against peers and identify recruitment targets; (iii) journal
editors "
"and conference organizers can use the affiliation geography to broaden reviewer and speaker pools beyond the
usual "
"centres; (iv) science-of-science researchers can study concentration, mobility and the emergence of new players
"
"(the China and Brazil growth signals here are a natural starting point); and (v) the subnational layer supports
"
"domestic science-policy analysis–for example, the strong showing of U.S. federal agencies and land-grant "
"universities in forestry is directly relevant to public-sector research funding.")

```

```

body("Caveats and limitations. Several boundaries on interpretation follow from the data and methods. First, the
"
"affiliation country is the author's current institution, not their nationality or origin; the concentration in
"
"wealthy countries partly reflects the international mobility of talent toward well-funded institutions. Second,
the "
"underlying citation data are from Scopus and inherit its coverage biases toward English-language and journal-
based "
"output, which disadvantages regions with strong grey-literature or non-English forestry traditions. Third, the
"
"counting unit is the author-year; because the same scientists recur across releases, year-on-year changes
reflect "
"both genuine new entrants and re-computation of the list, and should be read as trends rather than independent
"
"samples. Fourth, the subnational layer depends on institution-to-place geolocation: about 6% of U.S.
institutions "
"could not be confidently placed, and distributed federal agencies are deliberately pooled rather than assigned
to a "
"headquarters state. Finally, the list itself is a citation-based construct and shares the well-documented
limitations "
"of citation metrics; it should complement, not replace, expert judgement.")

# ===== 6. CONCLUSION =====
doc.add_heading("6. Conclusion", level=1)
body("Across five annual releases of the Stanford / Ioannidis career-long citation database, the population of "
"highly cited forestry and ecology researchers is large (23,374 author-year records; 8,425 distinct scientists)
and "
"strongly concentrated. North America and Europe supply more than four-fifths of all entries; the United States,
"
"United Kingdom, Canada and Australia are the leading countries; California and a block of U.S. federal science
"
"agencies lead within the United States; and the Swedish University of Agricultural Sciences is the most
represented "
"single institution. Ecology outnumbers forestry roughly five to one. Against this stable, Global-North-
dominated "
"backdrop, the fastest growth is occurring outside the traditional centres—most strikingly in China, but also in
"
"Brazil, Italy, South Africa and the Nordic and Southern-Hemisphere countries—suggesting a slow but measurable "
"geographic broadening of the field.")

# ===== 7. SUMMARY =====
doc.add_heading("7. Project Summary", level=1)
body("This project retrieved the five most recent career-long author tables of the Stanford / Ioannidis
standardized "
"citation database (data years 2020–2024), extracted every record classified in the Forestry or Ecology
subfield, "
"and merged them into a single combined database of 23,374 records. Each record was resolved to continent, world
"
"region and country, and—for the United States, Canada and Australia—to state or province via institution "
"geolocation. The report characterizes representation along each of these dimensions and by institution and
subfield, "
"with all counts independently verified against the source releases. Deliverables accompanying this document
are: the "
"combined Forestry+Ecology career database (Excel and CSV); a companion single-year Forestry database; a multi-
sheet "
"representation-ranking workbook; and the six figures reproduced here. Nationality and a qualitative-
quantitative "
"field split were requested but are not derivable from the source data and were omitted with explanation.")

# ===== 8. REFERENCES =====
doc.add_heading("8. References and Data Sources", level=1)
refs = [
"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. Proceedings of the "
"National Academy of Sciences, 102(46), 16569–16572. https://doi.org/10.1073/pnas.0507655102,"
"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite
across "
"scientific disciplines. PLoS Biology, 14(7), e1002501. https://doi.org/10.1371/journal.pbio.1002501,"
"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author
database "
"annotated for scientific field. PLoS Biology, 17(8), e3000384. https://doi.org/10.1371/journal.pbio.3000384,"
"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized "
"citation indicators. PLoS Biology, 18(10), e3000918. https://doi.org/10.1371/journal.pbio.3000918,"
"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open "
"scientific journal ontology. In Proceedings of the 13th International Conference of the ISSI (pp. 66–77). "
"(Science-Metrix classification.)",
"Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 - Countries. "
"https://www.naturalearthdata.com (public domain).",
"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized
"
"citation indicators" (Version 8) [Data set]. Elsevier Data Repository. https://doi.org/10.17632/btchxktzyw.8,"

```

```

]
for i,rf in enumerate(refs, 1):
    par = doc.add_paragraph(); par.paragraph_format.left_indent = Inches(0.35)
    par.paragraph_format.first_line_indent = Inches(-0.35); par.paragraph_format.space_after = Pt(6)
    r0=par.add_run(f"[{i}] "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(9.5)
    r1=par.add_run(rf); r1.font.name="Arial"; r1.font.size=Pt(9.5)
p("Data sources.", size=9.5, bold=True, before=4, after=2)
body("Primary data: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions
3, 5, "
"6, 7 and 8, downloaded July 2026 via the Elsevier Digital Commons / Mendeley Data public API. Subfield
taxonomy: "
"field/subfield threshold tables (Table_2) distributed within the same dataset. Base maps: Natural Earth (world)
and a "
"public US-state boundary file. Analysis and figures: Python (pandas, openpyxl, matplotlib).", size=9.5)

doc.save(OUT)
print("saved:", OUT)
print("paragraphs:", len(doc.paragraphs), "tables:", len(doc.tables))

```

8. build_docx2.py (SHA-256 4a6163a3af36e96f...; 338 lines)

```

# -*- coding: utf-8 -*-
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.enum.section import WD_ORIENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_Representation_Report.docx"
D = json.load(open(f"{SCR}/refine_data.json", encoding="utf-8"))
YEARS=[2020,2021,2022,2023,2024]
GREEN="2F5233"; LIGHT="E7EEE3"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"), "Arial")
for i,(sz,col) in enumerate([(16, GREEN), (13, GREEN), (11, "333333")], start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width, sec.page_height=Inches(8.5), Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="", size=11, bold=False, italic=False, align=None, color=None, after=6, before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t, size=10.5, after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
par.paragraph_format.line_spacing=1.13; return par
def bullet(t, size=10.5):
    par=doc.add_paragraph(style="List Bullet"); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size);
par.paragraph_format.space_after=Pt(2); return par
def shade(cell, hexc):
    tcPr=cell.tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"), "clear");
sh.set(qn("w:fill"), hexc); tcPr.append(sh)
def setcell(cell, text, bold=False, white=False, size=8, align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={"left":WD_ALIGN_PARAGRAPH.LEFT, "center":WD_ALIGN_PARAGRAPH.CENTER, "right":WD_ALIGN_PARAGRAPH.R
IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)
def table(headers, rows, widths, aligns=None, caption=None, band_rows=None, hsize=8, zebra=True):
    if caption: p(caption, size=8.5, italic=True, color="555555", after=3)
    band_rows=band_rows or set()
    t=doc.add_table(rows=1, cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
    aligns=aligns or (["left"]*len(headers)-1)
    for j,h in enumerate(headers):

```

```

        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=hsize,align="center");
shade(t.rows[0].cells[j],GREEN)
    for i,row in enumerate(rows):
        cells=t.add_row().cells
        for j,val in enumerate(row):
            setcell(cells[j],val,size=8,align=aligns[j],bold=(i in band_rows))
            if i in band_rows: shade(cells[j],BAND)
            elif zebra and i%2==1: shade(cells[j],LIGHT)
        for col,w in zip(t.columns,widths):
            for c in col.cells: c.width=w
    doc.add_paragraph().paragraph_format.space_after=Pt(2)
def figure(path,width,caption):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path,width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    p(caption,size=8.5,italic=True,color="555555",align=WD_ALIGN_PARAGRAPH.CENTER,after=10)

def fmt(metric,v):
    if metric=="scientists": return f"{int(round(v)):,}"
    if metric=="publications": return f"{v/1000:,.1f}"
    return f"{v/1e6:,.2f}"
UNIT={"scientists":"scientists","publications":"publications, thousands","citations":"citations, millions"}
SUF={"scientists":"","publications": "( '000)","citations": "( M)"}

def per_year_table(records, metric, keyhdr, caption, top=None, keyw=Inches(2.0), foldcountry=False):
    recs=records[:top] if top else records
    heads=[keyhdr]+[str(y)+SUF[metric] for y in YEARS]+["Total"+SUF[metric]]
    rows=[]
    for r in recs:
        key=r["key"]
        if foldcountry and r.get("country"): key=f'{key} ({r["country"]})'
        rows.append([key]+[fmt(metric,r[str(y)]) for y in YEARS]+[fmt(metric,r["total"])])
    ncol=len(heads); numw=(CW-keyw)/(ncol-1)
    table(heads,rows,[keyw]+[numw]*(ncol-1),aligns=["left"]+["right"]*(ncol-1),caption=caption)

def banded_table(banded, metric, keyhdr, caption, keyw=Inches(2.3), foldcountry=False):
    heads=[keyhdr]+[str(y)+SUF[metric] for y in YEARS]+["Total"+SUF[metric]]
    rows=[]; brows=set()
    for r in banded["head"]:
        key=r["key"]
        if foldcountry and r.get("country"): key=f'{key} ({r["country"]})'
        rows.append([key]+[fmt(metric,r[str(y)]) for y in YEARS]+[fmt(metric,r["total"])])
    for g in banded["bands"]:
        brows.add(len(rows))
        rows.append([f'Ranks {g["rank_from"]}-{g["rank_to"]} ({g["n"]})'+[fmt(metric,g[str(y)]) for y in YEARS]+[fmt(metric,g["total"])])
    ncol=len(heads); numw=(CW-keyw)/(ncol-1)
    table(heads,rows,[keyw]+[numw]*(ncol-1),aligns=["left"]+["right"]*(ncol-1),caption=caption,band_rows=brows)

def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=XmlElement("w:fldChar"); b.set(qn("w:fldCharType"),"begin")
        i=XmlElement("w:instrText"); i.set(qn("xml:space"),"preserve"); i.text=instr
        e=XmlElement("w:fldChar"); e.set(qn("w:fldCharType"),"end")
        for x in (b,i,e):
            r=par.add_run(); r.r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
    r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
    r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

TOT=D["totals"]
# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Representation of Highly Cited Forestry and Ecology Researchers by Scientists, Publications and Citations",
    size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)
p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020-2024) of the Stanford / Ioannidis
Career-Long Citation Database",
    size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)
p("Analytical Report – Revised Edition",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)
p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3-v8)",
    size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)
p("Abstract",size=13,bold=True,color=GREEN,after=4)
body(f"This report quantifies the geographic and institutional representation of the world's most highly cited
forestry and "
    f"ecology researchers along three complementary metrics—the number of listed scientists, the sum of their career
publications, "
    f"and the sum of their career citations—for each annual release of the Stanford / Ioannidis “top-scientists”
database ")

```

f"from data years 2020 to 2024. Across the five releases, 23,374 author-year records classified in the Forestry subfield "

f"(agriculture domain) or the Ecology subfield (biology domain) were extracted. In the 2024 release these "

f"{TOT['scientists']['2024'],} scientists account for {TOT['publications']['2024']/1000:,.0f} thousand career publications and "

f"{TOT['citations']['2024']/1e6:,.1f} million career citations. All three metrics are reported for every individual year and for "

f"every dimension—continent, world sub-region, country, first-level subdivision (U.S. state, Canadian province, Australian "

f"state), and institution (labelled with its country). Rankings list the top 20 members individually and then aggregate the "

f"remainder into cumulative bands each holding roughly ten percent of the residual total. A tree-chart maps the full field / "

f"sub-field / sub-sub-field taxonomy actually occupied by these authors, and a quantitative-versus-qualitative split is derived "

f"from each author's secondary methodological subfield. On every metric the list is dominated by the Global North—the United "

f"States, United Kingdom, Canada and Australia, and specialized agricultural universities such as the Swedish University of "

f"Agricultural Sciences—while the fastest growth appears in China, Brazil and several mid-sized nations. Maps show the "

f"five-year average only.",size=10)

kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)

r1=kw.add_run("bibliometrics; scientometrics; citation indicators; publication counts; Stanford/Ioannidis list; forestry; ecology; "

"research geography; Science-Metrix taxonomy; choropleth mapping; concentration/banding; quantitative methods."); r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True

doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

===== 1 BACKGROUND =====

doc.add_heading("1. Background, Concept and Related Work",level=1)

body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common single statistic, "

"the h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To standardize "

"comparison, Ioannidis and colleagues devised a composite indicator (the "c-score") combining six field-normalized "

"metrics [2,3] and published an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each "

"subfield) across 22 fields and 174 subfields defined by the Science-Metrix journal ontology [3,4]. The database appears in "

"career-long and single-year forms and is refreshed annually from Scopus. This report analyses the career-long tables.")

body("Whereas the original database is typically used to look up individuals, this report treats it as a population and asks how "

"that population is distributed—not only by head-count but also by the publications and citations those researchers carry, which "

"measure volume and influence rather than mere presence. Institution-level league tables (Shanghai, Times Higher Education, "

"Nature Index) and the proprietary Clarivate Highly Cited Researchers list address adjacent questions; the openness and "

"field-normalization of the Ioannidis data make it uniquely suited to the multi-metric, multi-scale geographic analysis "

"presented here for forestry and ecology—fields central to biodiversity, land management and climate policy.")

===== 2 DATA =====

doc.add_heading("2. Data Description",level=1)

body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset "

"doi:10.17632/btchxktzyw, versions 3, 5, 6, 7 and 8, corresponding to data years 2020-2024. Each ~75-85 MB spreadsheet holds "

"one row per ranked scientist (about 186,000-230,000 rows) with the author name, one large-institution affiliation, its ISO "

"country code, the six component citation metrics and composite c-score, the total career publication count (np) and career "

"citation count (nc), and the Science-Metrix classification (primary subfield, optional secondary subfield, and parent field). "

"From each release we kept every row whose primary or secondary subfield was Forestry or Ecology, yielding a combined database "

"of 23,374 author-year records that underlies this report and is delivered as an Excel workbook and CSV.")

table(["Metric \ Year"]+[str(y) for y in YEARS],

[["Scientists (list entries)"]+[f"{TOT['scientists']['str(y)'],}" for y in YEARS],

[["Publications (career papers)"]+[f"{TOT['publications']['str(y)'],}" for y in YEARS],

[["Citations (career citations)"]+[f"{TOT['citations']['str(y)'],}" for y in YEARS]],

[Inches(2.2)]+[Inches(0.9)]*5, aligns=["left"]+[["right"]]*5,

caption="Table 1. Combined Forestry+Ecology totals for each release year, by metric. Publications and citations are the sums of "

"each listed author's career totals; a scientist counts once per release year.")

===== 3 METHODS =====

doc.add_heading("3. Methods",level=1)

```

body("Metrics. Three quantities are tabulated for every group: the number of scientists (rows), publications (sum
of np) and "
"citations (sum of nc). Because career values are cumulative and each release re-lists largely the same people,
the counting "
"unit is the author-year; results are reported per release year, with maps showing the five-year average only.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For
the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus
(a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable
institutions as "
""Unknown." Institutions are labelled with their modal country.")
body("Ranking and banding. For countries and institutions, the top 20 by a given metric are listed individually;
the remainder "
"(institutions truncated at the top 100) is then aggregated into consecutive bands each accumulating about ten
percent of the "
"residual total, so the long tail is summarized without discarding it. Complete, unbanded rankings of all 83
countries and all "
"2,344 institutions-by scientists, publications and citations, for every year-are provided in the companion
workbook.")
body("Taxonomy and orientation. The tree-chart (Section 4.2) shows every domain, field and subfield actually
occupied by these "
"authors' primary or secondary classifications. A quantitative-versus-qualitative split labels an author "
""quantitative-oriented" when a primary or secondary subfield is a quantitative discipline-Statistics &
Probability, any "
"Engineering subfield, Geological & Geomatics Engineering (GIS/mensuration), Artificial Intelligence & Image
Processing (remote "
"sensing), Mathematics, or Bioinformatics. Requested terms such as Silviculture, Biometrics, Mensuration, Forest
Management, GIS "
"and Remote Sensing are forestry sub-disciplines that are not distinct labels in the Science-Matrix scheme; they
are captured "
"only insofar as they map onto these subfields, and Forestry/Ecology themselves are treated as the home fields
rather than as "
"quantitative or qualitative. Values were computed with Python (pandas); figures with matplotlib; maps use
Natural Earth "
"boundaries [6]. All totals were independently re-verified against the source releases.")

# ===== 4 RESULTS =====
doc.add_heading("4. Results",level=1)
doc.add_heading("4.1 Overall totals by metric and year",level=2)
body(f"All three metrics rose over the five years. Scientists grew 16% (4,344→5,043); career publications summed
by the cohort "
f"grew from {TOT['publications']['2020']/1000:,.0f}k to {TOT['publications']['2024']/1000:,.0f}k; and career
citations from "
f"{TOT['citations']['2020']/1e6:,.1f}M to {TOT['citations']['2024']/1e6:,.1f}M (Figure 1). Citations grew
fastest in relative "
"terms, reflecting both new entrants and the continual accrual of citations to established authors.")
figure(f"{SCR}/rfl_totals.png",Inches(6.7),"Figure 1. Cohort totals per release year for each metric: number of
scientists, summed career publications, summed career citations.")

doc.add_heading("4.2 Taxonomy tree: fields, sub-fields and sub-sub-fields present in the data",level=2)
tree=D["tree"]; nquant=sum(1 for v in tree.values() if v["quant"])
body(f"Although every author is a forestry or ecology researcher, their secondary classifications spread across
the whole of "
f"science. Figure 2 charts the {len(tree)} sub-sub-fields (Science-Matrix subfields) actually occupied by the
cohort, grouped "
f"under their {len({v['field'] for v in tree.values()})} sub-fields (fields) and 5 fields (domains). The two
target sub-sub-fields "
f"(Forestry, Ecology) are gold; {nquant} quantitative sub-sub-fields are marked in blue. Beyond the home
branches (Biology and "
"Agriculture-Fisheries-Forestry dominate), the largest secondary presences are in Earth & Environmental
Sciences, Enabling "
"Technologies (energy, biotechnology) and the Geomatics/Environmental Engineering subfields that carry the
quantitative, "
"GIS- and remote-sensing-oriented work.")
figure(f"{SCR}/rfl1_tree.png",Inches(6.9),"Figure 2. Tree-chart of the Field (Domain) → Sub-Field → Sub-Sub-Field
taxonomy occupied by the "
"Forestry+Ecology cohort. Right-hand numbers are author-records touching each node across 2020-2024 (an author
may appear under "
"two sub-sub-fields). Gold = target subfields; blue (») = quantitative subfields; grey = other.")

doc.add_heading("4.3 Field composition and quantitative vs qualitative orientation",level=2)
fld={r["key"]:r for r in D["field"]["scientists"]}
body(f"Ecology dominates the cohort: in 2024, {fld['Ecology']['2024']:} of {TOT['scientists']['2024']:}
scientists were in Ecology, "
f"{fld['Forestry']['2024']:} in Forestry and {fld['Both']['2024']} in both, a roughly five-to-one ecology-to-
forestry ratio that "
"holds on publications and citations too (Figure 3). Applying the secondary-subfield rule, only about six
percent of the cohort "

```

```

" is quantitative-oriented (Figure 4): forestry and ecology remain, in these data, predominantly field- and
organism-based rather "
" than methodological, though the quantitative minority is growing slightly.")
for m in ["scientists", "publications", "citations"]:
    per_year_table(D["field"][m], m, "Field", f"Table 2{'abc'[['scientists', 'publications', 'citations'].index(m)]}.
Field composition by {UNIT[m]}, per year.", keyw=Inches(1.6))
figure(f"{SCR}/rf2_field.png", Inches(6.7), "Figure 3. Field composition (Ecology / both / Forestry) per year, for
each metric.")
o={r["key"]:r for r in D["orient"]["scientists"]}
table(["Orientation"]+[str(y) for y in YEARS]+["Total"],
["Quantitative-oriented"]+[f'{o["Quantitative-oriented"][str(y)]:,}' for y in YEARS]+[f'{o["Quantitative-
oriented"]["total"]:,}'],
["Qualitative-oriented"]+[f'{o["Qualitative-oriented"][str(y)]:,}' for y in YEARS]+[f'{o["Qualitative-
oriented"]["total"]:,}']],
[Inches(2.0)]+[Inches(0.78)]*6, aligns=["left"]+["right"]*6,
caption="Table 3. Quantitative- vs qualitative-oriented scientists per year (by secondary Science-Metrix
subfield).")
figure(f"{SCR}/rf8_orient.png", Inches(6.7), "Figure 4. Quantitative- vs qualitative-oriented scientists per year,
for each metric.")

```

```

doc.add_heading("4.4 Representation by continent and world region", level=2)
body("On every metric, North America and Europe supply the great majority of the cohort, with Oceania (Australia
and New Zealand) a "
" strong third and Asia, South America and Africa trailing far behind despite holding most of the world's forests
and biodiversity "
" (Figure 5, maps in Figure 8).")
for m in ["scientists", "publications", "citations"]:
    per_year_table(D["continent"][m], m, "Continent", f"Table
4{'abc'[['scientists', 'publications', 'citations'].index(m)]}. Continents by {UNIT[m]}, per
year.", keyw=Inches(1.7))
figure(f"{SCR}/rf3_continent.png", Inches(6.9), "Figure 5. Continents by each metric, per release year.")
per_year_table(D["region"]["scientists"], "scientists", "World sub-region",
"Table 5. World sub-regions by number of scientists, per year (publications and citations by region are in the
workbook).", keyw=Inches(2.0))

```

```

doc.add_heading("4.5 Representation by country (top 20 individual, remainder banded)", level=2)
cs={r["key"]:r for r in D["country"]["scientists"]}
body(f"The United States leads on every metric ({cs['United States']['2024']:,} scientists in 2024), followed by
the United Kingdom, "
" Canada and Australia; Sweden and Finland punch far above their size. Tables 6a-c list the top 20 countries
individually and band "
" the remainder into ~10% cumulative groups; the per-year line charts (Figure 6) and the average maps (Figure 8)
show the spatial "
" pattern and the faster growth of China, Brazil and other emerging producers.")
for m in ["scientists", "publications", "citations"]:
    banded_table(D["country_banded"][m], m, "Country", f"Table
6{'abc'[['scientists', 'publications', 'citations'].index(m)]}. Countries by {UNIT[m]}: top 20 individual,
remainder in ~10% cumulative bands.", keyw=Inches(2.0))
figure(f"{SCR}/rf4_country.png", Inches(6.9), "Figure 6. Top-8 countries by each metric, per release year.")
figure(f"{SCR}/rf5_country_band.png", Inches(5.6), "Figure 7. Countries by scientists (average/yr): top-20
individual bars, remainder in ~10% cumulative bands.")
figure(f"{SCR}/rf9_worldmaps.png", Inches(6.2), "Figure 8. World choropleths of the five-year average per year for
each metric (scientists, publications, citations). Boundaries: Natural Earth [6].")

```

```

doc.add_heading("4.6 Subnational representation (United States, Canada, Australia)", level=2)
uss={r["key"]:r for r in D["us_state"]["scientists"]}
body(f"Within the United States, California leads on all metrics; a pooled block of federal agencies (USDA Forest
Service, USGS, "
" USDA-ARS) ranks second; New York, North Carolina, Colorado, Oregon and Florida follow (Figure 9 maps the
average of each "
" metric). In Canada, Ontario and British Columbia lead; in Australia, Queensland and New South Wales. Full per-
metric, per-year "
" subnational tables are in the workbook.")
per_year_table(D["us_state"]["scientists"], "scientists", "US state / category",
"Table 7. Top U.S. states/categories by number of scientists, per year (publications and citations by state are
in the workbook).", top=14, keyw=Inches(2.0))
figure(f"{SCR}/rf10_usmaps.png", Inches(6.0), "Figure 9. United States choropleths of the five-year average per
year for each metric (contiguous states; AK and HI omitted from frame).")

```

```

doc.add_heading("4.7 Representation by institution (with country; top 20 individual, ranks 21-100
banded)", level=2)
us=D["university"]["scientists"][0]
body(f"The Swedish University of Agricultural Sciences leads on scientists, with the University of British
Columbia, UC Davis, Oregon "
" State University and Wageningen close behind; U.S. federal research bodies and land-grant universities are
prominent. Tables "
" 8a-c list the top 20 institutions (with country) individually and band ranks 21-100 into ~10% cumulative
groups; per-year "
" trends are in Figure 10.")

```

```

for m in ["scientists","publications","citations"]:
    banded table(D["university_banded"][m],m,"Institution (country)",f"Table
8{'abc'[['scientists','publications','citations'].index(m)]}. Institutions by {UNIT[m]}: top 20 individual, ranks
21-100 in ~10% cumulative bands.",keyw=Inches(2.7),foldcountry=True)
figure(f"{SCR}/rf6_university.png",Inches(6.9),"Figure 10. Top-7 institutions by each metric, per release year.")
figure(f"{SCR}/rf7_university_band.png",Inches(5.6),"Figure 11. Institutions by scientists (average/yr): top-20
individual, ranks 21-100 in ~10% cumulative bands.")

# ===== 5 DISCUSSION =====
doc.add_heading("5. Discussion",level=1)
body("Adding publications and citations to the head-count sharpens rather than softens the picture: the same
countries and "
"institutions that field the most highly cited forestry and ecology researchers also carry, by construction,
most of the "
"cohort's papers and citations, so influence is at least as concentrated as presence. The gap between where
forests and "
"biodiversity are (largely tropical and Global South) and where the recognized, most-cited expertise is
affiliated (temperate, "
"wealthy, English-publishing) is a quantitative entry point into debates on research equity and capacity-
building.")
body("The multi-metric, multi-scale tables support several uses: funders and ministries can benchmark national
standing on head-count, "
"output and impact separately and track it yearly; universities can benchmark peers and target recruitment (the
banded tail shows "
"where the next tier sits); editors can widen reviewer and speaker pools using the country and subnational
geography; and "
"science-of-science researchers can study concentration (the banding makes the long tail explicit) and emergence
(China's and "
"Brazil's growth). The quantitative-orientation split flags the small but growing methodological community (GIS,
remote sensing, "
"statistics, engineering) within these fields.")
body("Limitations carry over and compound. Affiliation country is current institution, not nationality or origin.
Scopus coverage "
"biases toward English-language journals. Publication and citation sums inherit the skew of citation
distributions-a few "
"superstar authors dominate-so group totals are sensitive to individual outliers. The counting unit is the
author-year, so annual "
"changes mix genuine entry with re-computation. Subnational placement leaves ~6% of U.S. institutions "Unknown"
and pools federal "
"agencies. Finally, the quantitative/qualitative split is a coarse proxy: the forestry sub-disciplines the
reader named "
"(Silviculture, Mensuration, Forest Management...) are not separable in Science-Metrix and are approximated by
adjacent subfields.")

# ===== 6 CONCLUSION =====
doc.add_heading("6. Conclusion",level=1)
body("Measured by scientists, publications or citations, the population of highly cited forestry and ecology
researchers is large "
"and strongly concentrated in the Global North. North America and Europe dominate every metric; the United
States, United "
"Kingdom, Canada and Australia lead among countries; California and a block of U.S. federal agencies lead
subnationally; and the "
"Swedish University of Agricultural Sciences is the most represented institution. Ecology outnumbers forestry
about five to one, "
"and only ~6% of the cohort is quantitatively oriented. The banded rankings show a long, evenly graded tail
beneath the leaders, "
"and the growth signal-fastest in China, Brazil and mid-sized nations-points to a slow diversification that the
citation and "
"publication metrics corroborate as clearly as the head-count.")

# ===== 7 SUMMARY =====
doc.add_heading("7. Project Summary",level=1)
body("Five career-long releases of the Stanford / Ioannidis database (2020-2024) were filtered to Forestry and
Ecology, combined "
"into 23,374 records, and analysed by number of scientists, publications and citations across continent, region,
country, "
"subnational unit, institution (with country) and taxonomic subfield. Rankings list the top 20 individually and
band the rest by "
"~10% cumulative contribution; maps show the five-year average. Deliverables: this report; a comprehensive
workbook with the full "
"per-year, per-metric rankings of all 83 countries and all 2,344 institutions plus continent, region,
subnational, field, "
"orientation, taxonomy-tree and map-data sheets; and the combined career database (Excel and CSV).")

# ===== 8 REFERENCES =====
doc.add_heading("8. References and Data Sources",level=1)
refs=[
"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569-
16572. https://doi.org/10.1073/pnas.0507655102",

```

```

"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite
across scientific disciplines. PLoS Biology, 14(7), e1002501. https://doi.org/10.1371/journal.pbio.1002501",
"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author
database annotated for scientific field. PLoS Biology, 17(8), e3000384.
https://doi.org/10.1371/journal.pbio.3000384",
"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized
citation indicators. PLoS Biology, 18(10), e3000918. https://doi.org/10.1371/journal.pbio.3000918",
"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open
scientific journal ontology. Proc. 13th ISSI Conference, 66-77. (Science-Metrix classification.)",
"Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 - Countries.
https://www.naturalearthdata.com (public domain).",
"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized
citation indicators" (Version 8) [Data set]. Elsevier Data Repository. https://doi.org/10.17632/btchxktzyw.8",
]
for i,rf in enumerate(refs,1):
    par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);
par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)
a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)
b=par.add_run(rf); b.font.name="Arial"; b.font.size=Pt(9.5)
p("Data sources.",size=9.5,bold=True,before=4,after=2)
body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions
3/5/6/7/8, via the "
"Elsevier Digital Commons / Mendeley Data public API. Taxonomy: field/subfield threshold tables (Table_2) in the
same dataset. "
"Base maps: Natural Earth (world) and a public US-state boundary file. Analysis and figures: Python (pandas,
openpyxl, matplotlib).",size=9.5)

doc.save(OUT)
print("saved:",OUT,"| paragraphs:",len(doc.paragraphs),"| tables:",len(doc.tables))

```

9. build_docx3.py (SHA-256 8e5a98c9f5a50179...; 426 lines)

```

# -*- coding: utf-8 -*-
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.docx"
F = json.load(open(f"{SCR}/forestry_data.json", encoding="utf-8")) # main = Forestry
C = json.load(open(f"{SCR}/refine_data.json", encoding="utf-8")) # appendix = combined
YEARS=[2020,2021,2022,2023,2024]
GREEN="2F5233"; LIGHT="E7EEE3"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"),"Arial")
for i,(sz,col) in enumerate([(16,GREEN),(13,GREEN),(11,"333333")],start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width,sec.page_height=Inches(8.5),Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="",size=11,bold=False,italic=False,align=None,color=None,after=6,before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t,size=10.5,after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
par.paragraph_format.line_spacing=1.13; return par
def shade(cell,hexc):
    tcPr=cell._tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"),"clear");
sh.set(qn("w:fill"),hexc); tcPr.append(sh)
def setcell(cell,text,bold=False,white=False,size=8,align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={"left":WD_ALIGN_PARAGRAPH.LEFT,"center":WD_ALIGN_PARAGRAPH.CENTER,"right":WD_ALIGN_PARAGRAPH.R
IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)

```

```

def table(headers,rows,widths,aligns=None,caption=None,band_rows=None,hsize=8,zebra=True):
    if caption: p(caption,size=8.5,italic=True,color="555555",after=3)
    band_rows=band_rows or set()
    t=doc.add_table(rows=1,cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow autofit=False; t.style="Table Grid"
    aligns=aligns or (["left"]+["right"]*(len(headers)-1))
    for j,h in enumerate(headers):
        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=hsize,align="center");
    shade(t.rows[0].cells[j],GREEN)
    for i,row in enumerate(rows):
        cells=t.add_row().cells
        for j,val in enumerate(row):
            setcell(cells[j],val,size=8,align=aligns[j],bold=(i in band_rows))
            if i in band_rows: shade(cells[j],BAND)
            elif zebra and i%2==1: shade(cells[j],LIGHT)
        for col,w in zip(t.columns,widths):
            for c in col.cells: c.width=w
    doc.add_paragraph().paragraph_format.space_after=Pt(2)
def figure(path,width,caption):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path,width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    p(caption,size=8.5,italic=True,color="555555",align=WD_ALIGN_PARAGRAPH.CENTER,after=10)
def fmt(metric,v):
    if metric=="scientists": return f"{int(round(v)):}"
    if metric=="publications": return f"{v/1000:,.1f}"
    return f"{v/1e6:,.2f}"
UNIT={"scientists":"scientists","publications":"publications, thousands","citations":"citations, millions"}
SUF={"scientists":"","publications":'('000)',"citations":'('M)'}
ABC={"scientists":"a","publications":"b","citations":"c"}
def per_year_table(records,metric,keyhdr,caption,top=None,keyw=Inches(2.0),foldcountry=False):
    recs=records[:top] if top else records
    heads=[keyhdr]+[str(y)+SUF[metric] for y in YEARS]+["Total"+SUF[metric]]
    rows=[]
    for r in recs:
        key=r["key"]
        if foldcountry and r.get("country"): key=f'{key} ({r["country"]})'
        rows.append([key]+[fmt(metric,r[str(y)]) for y in YEARS]+[fmt(metric,r["total"])])
    ncol=len(heads); numw=(CW-keyw)/(ncol-1)
    table(heads,rows,[keyw]+[numw]*(ncol-1),aligns=["left"]+["right"]*(ncol-1),caption=caption)
def banded_table(banded,metric,keyhdr,caption,keyw=Inches(2.3),foldcountry=False):
    heads=[keyhdr]+[str(y)+SUF[metric] for y in YEARS]+["Total"+SUF[metric]]
    rows=[]; brows=set()
    for r in banded["head"]:
        key=r["key"]
        if foldcountry and r.get("country"): key=f'{key} ({r["country"]})'
        rows.append([key]+[fmt(metric,r[str(y)]) for y in YEARS]+[fmt(metric,r["total"])])
    for g in banded["bands"]:
        brows.add(len(rows))
        rows.append([f'Ranks {g["rank_from"]}-{g["rank_to"]} ({g["n"]})'+[fmt(metric,g[str(y)]) for y in YEARS]+
        [fmt(metric,g["total"])])
    ncol=len(heads); numw=(CW-keyw)/(ncol-1)
    table(heads,rows,[keyw]+[numw]*(ncol-1),aligns=["left"]+["right"]*(ncol-1),caption=caption,band_rows=brows)
def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=OxmlElement("w:fldChar"); b.set(qn("w:fldCharType"),"begin")
        i=OxmlElement("w:instrText"); i.set(qn("xml:space"),"preserve"); i.text=instr
        e=OxmlElement("w:fldChar"); e.set(qn("w:fldCharType"),"end")
        for x in (b,i,e):
            r=par.add_run(); r._r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
    r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
    r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

TF=F["totals"]; TC=C["totals"]
fc={r["key"]:r for r in F["country"]["scientists"]}
fcont={r["key"]:r for r in F["continent"]["scientists"]}
fo={r["key"]:r for r in F["orient"]["scientists"]}
fdual={r["key"]:r for r in F["dual"]["scientists"]}
funi=F["university"]["scientists"][0]
fus={r["key"]:r for r in F["us_state"]["scientists"]}
qhits=F["quant_subfield_hits"]

# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Global Representation of the World's Most-Cited Forestry Scientists by Number of Scientists, Publications and Citations",
    size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)
p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020-2024) of the Stanford / Ioannidis Career-Long Citation Database",

```

```

size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)
p("Analytical Report",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)
p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3-v8)",
size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)
p("Abstract",size=13,bold=True,color=GREEN,after=4)
body(f"This report quantifies the geographic and institutional representation of the world's most highly cited
forestry researchers "
f"along three complementary metrics—the number of listed scientists, the sum of their career publications, and
the sum of their "
f"career citations—for each annual release of the Stanford / Ioannidis "top-scientists" database from data years
2020 to 2024. "
f"From each release we retained every author placed in the Science-Metrix Forestry sub-sub-field (as primary or
secondary "
f"specialty), giving {int(sum(TF['scientists'].values())):,} author-year records; in the 2024 release these
{TF['scientists']['2024'],:,} "
f"forestry scientists carry {TF['publications']['2024']/1000:,.0f} thousand career publications and
{TF['citations']['2024']/1e6:,.1f} "
f"million career citations. All three metrics are reported for every individual year and for every dimension—
continent, world "
f"sub-region, country, first-level subdivision (U.S. state, Canadian province, Australian state) and institution
(labelled with its "
f"country). Rankings list the top 20 members individually and aggregate the remainder into cumulative bands each
holding roughly ten "
f"percent of the residual total; maps show the five-year average only. A tree-chart maps the field / sub-field /
sub-sub-field "
f"taxonomy the forestry cohort occupies, and a quantitative-versus-qualitative split is derived from each
author's secondary "
f"methodological subfield. The forestry list is dominated by the United States and Canada, with the Nordic
countries (Finland and "
f"Sweden) strongly over-represented relative to their size, and by specialized agricultural and forestry
universities. The parallel "
f"analysis of the combined Forestry+Ecology population—{int(sum(TC['scientists'].values())):,} records—on which
earlier drafts "
f"were based is retained in full in Appendix A.",size=10)
kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)
r1=kw.add_run("forestry; bibliometrics; scientometrics; citation indicators; publication counts;
Stanford/Ioannidis list; research "
"geography; Science-Metrix taxonomy; choropleth mapping; concentration/banding; quantitative forestry
(biometrics, mensuration, GIS, "
"remote sensing)."); r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

# ===== 1 BACKGROUND =====
doc.add_heading("1. Background, Concept and Related Work",level=1)
body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common
single statistic, the "
"h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To
standardize comparison, "
"Ioannidis and colleagues devised a composite indicator (the "c-score") combining six field-normalized metrics
[2,3] and "
"published an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each
subfield) across 22 fields "
"and 174 subfields defined by the Science-Metrix journal ontology [3,4]. The database appears in career-long and
single-year forms "
"and is refreshed annually from Scopus. This report analyses the career-long tables.")
body("Whereas the original database is typically used to look up individuals, this report treats one sub-sub-
field-Forestry-as a "
"population and asks how that population is distributed, not only by head-count but also by the publications and
citations those "
"researchers carry, which measure volume and influence rather than mere presence. Forestry is a small, applied,
globally distributed "
"field at the centre of timber economies, biodiversity, carbon and land-use policy, yet its most-cited expertise
has not, to our "
"knowledge, been mapped in this multi-metric way. Institution-level league tables (Shanghai, Times Higher
Education, Nature Index) "
"and the proprietary Clarivate Highly Cited Researchers list address adjacent questions; the openness and field-
normalization of the "
"Ioannidis data make it well suited to the analysis presented here. Because forestry and ecology overlap heavily
in practice, a "
"parallel analysis of the combined Forestry+Ecology population is preserved in Appendix A for readers who prefer
the broader frame.")

# ===== 2 DATA =====
doc.add_heading("2. Data Description",level=1)
body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset
doi:10.17632/btchxktzyw, "
"versions 3, 5, 6, 7 and 8, corresponding to data years 2020–2024. Each ~75–85 MB spreadsheet holds one row per
ranked scientist ")

```

```

"(about 186,000-230,000 rows) with the author name, one large-institution affiliation, its ISO country code, the
six component "
"citation metrics and composite c-score, the total career publication count (np) and career citation count (nc),
and the "
"Science-Metrix classification (primary subfield, optional secondary subfield, and parent field).")
body(f"For this report we kept every row whose primary or secondary subfield is Forestry, giving
{int(sum(TF['scientists'].values())):,} "
f"author-year records ({TF['scientists']['2020']:,} in 2020 rising to {TF['scientists']['2024']:,} in 2024). Of
these, "
f"{fdual.get('Forestry only',{}).get('total',0):,} are classified in Forestry alone and {fdual.get('Forestry &
Ecology',{}).get('total',0):,} "
"are jointly active in Ecology (Forestry being one of their two subfields). The wider Forestry+Ecology
extraction "
f"({int(sum(TC['scientists'].values())):,} records), analysed in Appendix A, is delivered together with this
report as an Excel "
"workbook and CSV. Table 1 gives the forestry totals for each release year and metric.")
table(["Metric \\ Year"]+[str(y) for y in YEARS],
[["Scientists (list entries)"]+[f"{TF['scientists'][str(y)]:,}" for y in YEARS],
["Publications (career papers)"]+[f"{TF['publications'][str(y)]:,}" for y in YEARS],
["Citations (career citations)"]+[f"{TF['citations'][str(y)]:,}" for y in YEARS]],
[Inches(2.2)]+[Inches(0.9)]*5, aligns=["left"]+["right"]*5,
caption="Table 1. Forestry cohort totals for each release year, by metric. Publications and citations are the
sums of each listed "
"author's career totals; a scientist counts once per release year.")

# ===== 3 METHODS =====
doc.add_heading("3. Methods",level=1)
body("Metrics. Three quantities are tabulated for every group: the number of scientists (rows), publications (sum
of np) and citations "
"(sum of nc). Because career values are cumulative and each release re-lists largely the same people, the
counting unit is the "
"author-year; results are reported per release year, with maps showing the five-year average only.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For
the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus
(a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable
institutions as "
""Unknown." Institutions are labelled with their modal country.")
body("Ranking and banding. For countries and institutions, the top 20 by a given metric are listed individually;
the remainder "
"(institutions truncated at the top 100) is then aggregated into consecutive bands each accumulating about ten
percent of the "
"residual total, so the long tail is summarized without discarding it. Complete, unbanded rankings of all "
f"{len(F['country']['scientists'])} forestry countries and all {len(F['university']['scientists'])} forestry
institutions-by "
"scientists, publications and citations, for every year-are in the companion workbook.")
body("Taxonomy and orientation. The tree-chart (Section 4.2) shows every domain, field and subfield occupied by
the forestry cohort's "
"primary or secondary classifications. A quantitative-versus-qualitative split labels an author "quantitative-
oriented" when a "
"primary or secondary subfield is a quantitative discipline. Following the reader's guidance, the quantitative
set comprises "
"Statistics & Probability; the Mathematical and Computational subfields (Applied, General and Numerical &
Computational Mathematics; "
"Computation Theory; Software and other Programming/ICT subfields); Operations Research; every Engineering
subfield including "
"Geological & Geomatics Engineering (GIS, mensuration and surveying) and Environmental Engineering; Artificial
Intelligence & Image "
"Processing (remote sensing); and Bioinformatics. Requested forestry sub-disciplines such as Silviculture,
Biometrics, Mensuration, "
"Forest Management, GIS and Remote Sensing are not distinct labels in the Science-Metrix scheme; they are
captured only where they "
"map onto these subfields, and Forestry itself is treated as the home field rather than as quantitative or
qualitative. Values were "
"computed with Python (pandas); figures with matplotlib; maps use Natural Earth boundaries [6]. All totals were
independently "
"re-verified against the source releases.")

# ===== 4 RESULTS =====
doc.add_heading("4. Results",level=1)
doc.add_heading("4.1 Overall totals by metric and year",level=2)
g_s=100*(TF['scientists']['2024']/TF['scientists']['2020']-1)
g_p=100*(TF['publications']['2024']/TF['publications']['2020']-1)
g_c=100*(TF['citations']['2024']/TF['citations']['2020']-1)
body(f"All three metrics rose over the five years (Figure 1). The number of listed forestry scientists grew
{g_s:.0f}% "
f"({TF['scientists']['2020']:,}→{TF['scientists']['2024']:,}); their summed career publications grew {g_p:.0f}%
"

```

```
f"({TF['publications']['2020']/1000:,.0f}k→{TF['publications']['2024']/1000:,.0f}k); and their summed career citations grew {g_c:.0f}% "
f"({TF['citations']['2020']/1e6:,.1f}M→{TF['citations']['2024']/1e6:,.1f}M). Citations grew fastest, reflecting both new entrants "
"and the continual accrual of citations to established authors.")
figure(f"{SCR}/ff1_change.png",Inches(6.6),"Figure 1. Growth of each metric for the forestry cohort, expressed as percent change from "
"2020; the actual value (scientists, thousands of publications, millions of citations) is labelled at each point.")
```

```
doc.add_heading("4.2 Taxonomy tree: fields, sub-fields and sub-sub-fields present in the forestry data",level=2)
tree=F["tree"]; nq=sum(1 for v in tree.values() if v["quant"])
body(f"Although every author here is a forestry researcher, their secondary classifications spread across science. Figure 2 charts the "
f"sub-sub-fields (Science-Matrix subfields) the forestry cohort actually occupies, grouped under their parent sub-fields (fields) and "
f"fields (domains). Besides Forestry itself, the strongest adjacent sub-sub-fields are Ecology, Plant Biology & Botany and Agronomy in "
"the life-science direction, and-carrying the quantitative, GIS- and remote-sensing-oriented work-Geological & Geomatics "
"Engineering and Environmental Engineering on the methodological side. The two home domains (Applied Sciences via "
"Agriculture-Fisheries-Forestry, and Natural Sciences via Biology) dominate, but eleven of the taxonomy's sub-sub-fields present "
"are quantitative (blue).")
figure(f"{SCR}/ff2_tree.png",Inches(6.9),"Figure 2. Tree-chart of the Field (Domain) → Sub-Field → Sub-Sub-Field taxonomy occupied by "
"the forestry cohort. Right-hand numbers are author-records touching each sub-sub-field across 2020-2024 (an author may appear under "
"two). Gold ● = Forestry (target); blue » = quantitative sub-sub-field; grey • = other/qualitative.")
```

```
doc.add_heading("4.3 Quantitative vs qualitative orientation",level=2)
body(f"Applying the secondary-subfield rule, {fo['Quantitative-oriented']['total']:,} of {int(sum(TF['scientists'].values())):,} "
f"author-records ({100*fo['Quantitative-oriented']['total']/sum(TF['scientists'].values()):.0f}%) are quantitative-oriented-a "
"somewhat higher share than in ecology, consistent with forestry's engineering and inventory traditions. The quantitative signal is "
"carried overwhelmingly by geomatics/GIS and environmental engineering, with statistics and operations research adding smaller "
"contributions (Table 2). Figure 3 shows the split is stable to slightly rising across the five years on all three metrics.")
grows=[k,f"{qhits[k]:,}"] for k in qhits]
table(["Quantitative sub-sub-field (secondary)","Author-records 2020-2024"],grows,
[Inches(3.6),Inches(3.1)],aligns=["left","right"],
caption="Table 2. Quantitative sub-sub-fields present among forestry authors (as primary or secondary specialty) and the number of "
"author-records touching each, 2020-2024. Sums exceed the quantitative-oriented head-count because a few authors touch two.")
o=F["orient"]
table(["Orientation"]+[str(y) for y in YEARS]+["Total"],
[["Quantitative-oriented"]+[f'{fo["Quantitative-oriented"][str(y):,}'] for y in YEARS]+[f'{fo["Quantitative-oriented"]['total']:,}'],
["Qualitative-oriented"]+[f'{fo["Qualitative-oriented"][str(y):,}'] for y in YEARS]+[f'{fo["Qualitative-oriented"]['total']:,}'],
[Inches(2.0)]+[Inches(0.78)]*6,aligns=["left"]+["right"]*6,
caption="Table 3. Quantitative- vs qualitative-oriented forestry scientists per year (by secondary Science-Matrix subfield).")
figure(f"{SCR}/ff4_orient.png",Inches(6.4),"Figure 3. Quantitative- vs qualitative-oriented forestry scientists per year, for each metric.")
```

```
doc.add_heading("4.4 Representation by continent and world region",level=2)
body(f"On every metric, North America and Europe supply the great majority of forestry's most-cited researchers "
f"(North America {fcont.get('North America',{}).get('2024',0):,} scientists in 2024; Europe {fcont.get('Europe',{}).get('2024',0):,}), "
f"with Oceania a distant third and Asia, South America and Africa trailing far behind despite holding most of the world's forests "
"(Figure 4; average maps in Figure 7).")
for m in ["scientists","publications","citations"]:
    per_year_table(F["continent"][m],m,"Continent",f"Table 4{ABC[m]}. Continents by {UNIT[m]}, per year.",keyw=Inches(1.7))
figure(f"{SCR}/ff3_continent.png",Inches(6.4),"Figure 4. Forestry representation by continent, by metric, per release year.")
per_year_table(F["region"]["scientists"],"scientists","World sub-region",
"Table 5. World sub-regions by number of forestry scientists, per year (publications and citations by region are in the workbook).",keyw=Inches(2.0))
```

```
doc.add_heading("4.5 Representation by country (top 20 individual, remainder banded)",level=2)
```

```

body(f"The United States leads forestry on every metric ({fc['United States']['2024']:,} scientists in 2024),
followed by Canada "
f"({fc.get('Canada',{}).get('2024',0):,}); the Nordic forestry powers Finland and Sweden, together with Germany,
form the next tier "
"and rank far above their population share. Tables 6a-c list the top 20 countries individually and band the
remainder into ~10% "
"cumulative groups; per-year trends are in Figure 5 and the banded profile in Figure 6.")
for m in ["scientists","publications","citations"]:
    banded_table(F["country_banded"][m],m,"Country",f"Table 6{ABC[m]}. Forestry countries by {UNIT[m]}: top 20
individual, remainder in ~10% cumulative bands.",keyw=Inches(2.0))
figure(f"{SCR}/ff5_country.png",Inches(6.4),"Figure 5. Top-8 forestry countries by each metric, per release
year.")
figure(f"{SCR}/ff6_country_band.png",Inches(6.2),"Figure 6. Forestry countries by scientists (average/yr): top-20
individual bars, remainder in ~10% cumulative bands.")
figure(f"{SCR}/ff9_worldmaps.png",Inches(6.0),"Figure 7. World choropleths of the five-year-average forestry
representation for each metric (scientists, publications, citations). Boundaries: Natural Earth [6].")

doc.add_heading("4.6 Subnational representation (United States, Canada, Australia)",level=2)
body(f"Within the United States, Oregon, California and a pooled block of federal agencies (USDA Forest Service,
USGS, USDA-ARS) lead "
"forestry representation, reflecting the country's public-lands forestry research system; the Pacific North-West
and Rocky Mountain "
"states are prominent. In Canada, British Columbia and Ontario lead; in Australia, Queensland and Victoria.
Figure 8 maps the "
"five-year average by state; full per-metric, per-year subnational tables are in the workbook.")
per_year_table(F["us_state"]["scientists"],"scientists","US state / category",
"Table 7. Top U.S. states/categories by number of forestry scientists, per year (publications and citations by
state are in the workbook).",top=14,keyw=Inches(2.0))
figure(f"{SCR}/ff10_usmaps.png",Inches(5.9),"Figure 8. United States choropleths of the five-year-average
forestry representation for each metric (contiguous states; AK and HI omitted from frame).")

doc.add_heading("4.7 Representation by institution (with country; top 20 individual, ranks 21-100
banded)",level=2)
body(f"The most-represented forestry institution is {funi['key']} ({funi.get('country','')}), followed by other
specialized "
"agricultural and forestry universities, North-American land-grant universities and national forest-research
agencies. Tables 8a-c "
"list the top 20 institutions (with country) individually and band ranks 21-100 into ~10% cumulative groups;
per-year trends are in "
"Figure 9 and the banded profile in Figure 10.")
for m in ["scientists","publications","citations"]:
    banded_table(F["university_banded"][m],m,"Institution (country)",f"Table 8{ABC[m]}. Forestry institutions by
{UNIT[m]}: top 20 individual, ranks 21-100 in ~10% cumulative bands.",keyw=Inches(2.7),foldcountry=True)
figure(f"{SCR}/ff7_university.png",Inches(6.4),"Figure 9. Top-7 forestry institutions by each metric, per release
year.")
figure(f"{SCR}/ff8_university_band.png",Inches(6.2),"Figure 10. Forestry institutions by scientists (average/yr):
top-20 individual, ranks 21-100 in ~10% cumulative bands.")

# ===== 5 DISCUSSION =====
doc.add_heading("5. Discussion",level=1)
body("Adding publications and citations to the head-count sharpens rather than softens the picture: the same
countries and institutions "
"that field the most highly cited forestry researchers also carry, by construction, most of the cohort's papers
and citations, so "
"influence is at least as concentrated as presence. The gap between where forests are (largely tropical and
Global South) and where "
"the recognized, most-cited forestry expertise is affiliated (temperate, wealthy, strongly Nordic and North-
American) is a "
"quantitative entry point into debates on research equity and forestry capacity-building in the tropics.")
body("The multi-metric, multi-scale tables support several uses: funders and forestry ministries can benchmark
national standing on "
"head-count, output and impact separately and track it yearly; forestry schools and research institutes can
benchmark peers and "
"target recruitment (the banded tail shows where the next tier sits); journal editors can widen reviewer and
speaker pools using the "
"country and subnational geography; and science-of-science researchers can study concentration and emergence.
The quantitative-"
"orientation split flags the growing methodological community-geomatics/GIS, remote sensing, environmental
engineering, statistics "
"and operations research-within a traditionally field-based discipline, which is relevant to curriculum design
and to the demand for "
"quantitative skills in modern forest inventory, monitoring and management.")
body("Limitations. Affiliation country is current institution, not nationality or origin. Scopus coverage biases
toward English-language "
"journals, which likely understates tropical and non-English forestry. Publication and citation sums inherit the
skew of citation "
"distributions-a few superstar authors dominate-so group totals are sensitive to individual outliers. The
counting unit is the "
"author-year, so annual changes mix genuine entry with re-computation. Subnational placement leaves some U.S.
institutions ")

```

"Unknown" and pools federal agencies. Finally, the quantitative/qualitative split is a coarse proxy: the forestry sub-disciplines "
 "named by the reader (Silviculture, Biometrics, Mensuration, Forest Management...) are not separable in Science-Metrix and are "
 "approximated by adjacent engineering, geomatics and statistics subfields.")

===== 6 CONCLUSION =====
 doc.add_heading("6. Conclusion",level=1)
 body(f"Measured by scientists, publications or citations, the population of highly cited forestry researchers is small "
 f"({TF['scientists'][:2024]:,} in 2024) but growing on every metric, and strongly concentrated in the Global North. North America "
 "and Europe dominate; the United States and Canada lead among countries, with Finland and Sweden conspicuously over-represented; "
 f"Oregon, California and a block of U.S. federal agencies lead subnationally; and {funi['key']} is the most represented institution. "
 f"About {100*fo['Quantitative-oriented'][:total]}/sum(TF['scientists'].values()):.0f}% of the cohort is quantitatively oriented, led by "
 "geomatics/GIS and environmental engineering, and that share is slowly rising. The banded rankings reveal a long, evenly graded tail "
 "beneath the leaders, and the citation and publication metrics corroborate the head-count's geography as clearly as they extend it.")

===== 7 SUMMARY =====
 doc.add_heading("7. Project Summary",level=1)
 body(f"Five career-long releases of the Stanford / Ioannidis database (2020-2024) were filtered to the Forestry sub-sub-field "
 f"({int(sum(TF['scientists'].values())):,} records) and analysed by number of scientists, publications and citations across continent, "
 "region, country, subnational unit, institution (with country) and taxonomic subfield. Rankings list the top 20 individually and band "
 "the rest by ~10% cumulative contribution; maps show the five-year average. The broader Forestry+Ecology analysis is retained in "
 "Appendix A. Deliverables: this report; a comprehensive workbook with the full per-year, per-metric rankings of all forestry "
 "countries and institutions plus continent, region, subnational, orientation, taxonomy-tree and map-data sheets (and the combined "
 "equivalents); and the combined career database (Excel and CSV).")

===== 8 REFERENCES =====
 doc.add_heading("8. References and Data Sources",level=1)
 refs=[
 "Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569-16572. <https://doi.org/10.1073/pnas.0507655102>",
 "Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite across scientific disciplines. PLoS Biology, 14(7), e1002501. <https://doi.org/10.1371/journal.pbio.1002501>",
 "Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author database annotated for scientific field. PLoS Biology, 17(8), e3000384. <https://doi.org/10.1371/journal.pbio.3000384>",
 "Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized citation indicators. PLoS Biology, 18(10), e3000918. <https://doi.org/10.1371/journal.pbio.3000918>",
 "Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open scientific journal ontology. Proc. 13th ISSI Conference, 66-77. (Science-Metrix classification.),
 "Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 - Countries. <https://www.naturalearthdata.com> (public domain).",
 "Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized citation indicators" (Version 8) [Data set]. Elsevier Data Repository. <https://doi.org/10.17632/btchxktzyw.8>,
]
 for i,rf in enumerate(refs,1):
 par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);
 par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)
 a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)
 b=par.add_run(rf); b.font.name="Arial"; b.font.size=Pt(9.5)
 p("Data sources.",size=9.5,bold=True,before=4,after=2)
 body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions 3/5/6/7/8, via the Elsevier "
 "Digital Commons / Mendeley Data public API. Taxonomy: field/subfield threshold tables (Table_2) in the same dataset. Base maps: "
 "Natural Earth (world) and a public US-state boundary file. Analysis and figures: Python (pandas, openpyxl, matplotlib).",size=9.5)

===== APPENDIX A: COMBINED =====
 doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
 doc.add_heading("Appendix A. Combined Forestry + Ecology Analysis",level=1)
 body(f"This appendix preserves, unchanged in substance, the broader analysis of the combined Forestry+Ecology population "
 f"({int(sum(TC['scientists'].values())):,} author-year records: the {int(sum(TF['scientists'].values())):,} forestry records above plus "

```

"all ecology records) on which earlier drafts of this report were based. It is provided for readers who prefer
to treat forestry and "
"ecology together—they overlap heavily in journals, methods and personnel. Method and metric definitions are
identical to the main "
"report; only the population differs.")
cc={r["key"]:r for r in C["country"]["scientists"]}
ccont={r["key"]:r for r in C["continent"]["scientists"]}
cfld={r["key"]:r for r in C["field"]["scientists"]}
cuni=C["university"]["scientists"][0]

doc.add_heading("A.1 Combined totals and field composition",level=2)
body(f"The combined cohort comprises {int(sum(TC['scientists'].values())):,} author-year records; in 2024 its
{TC['scientists']['2024']:,} "
f"scientists carry {TC['publications']['2024']/1000:,.0f}k publications and {TC['citations']['2024']/1e6:,.1f}M
citations. Ecology "
f"dominates: in 2024, {cfld['Ecology']['2024']:,} scientists were in Ecology alone, {cfld['Forestry']['2024']:,}
in Forestry alone and "
f"{cfld['Both']['2024']:,} in both—a roughly five-to-one ecology-to-forestry ratio that holds on publications
and citations too "
"(Figure A2). This is the central reason the combined geography (Figure A3–A5) leans more toward large ecology
countries "
"(the United Kingdom and Australia rise above Canada) than the forestry-only geography in the main report.")
table(["Metric \ Year"]+[str(y) for y in YEARS],
[["Scientists (list entries)"]+[f"{TC['scientists'][str(y)]:,}" for y in YEARS],
["Publications (career papers)"]+[f"{TC['publications'][str(y)]:,}" for y in YEARS],
["Citations (career citations)"]+[f"{TC['citations'][str(y)]:,}" for y in YEARS]],
[Inches(2.2)]+[Inches(0.9)]*5, aligns=["left"]+["right"]*5,
caption="Table A1. Combined Forestry+Ecology totals for each release year, by metric.")
for m in ["scientists","publications","citations"]:
    per_year_table(C["field"][m],m,"Field",f"Table A2{ABC[m]}. Field composition (Ecology / both / Forestry) by
{UNIT[m]}, per year.",keyw=Inches(1.6))
figure(f"{SCR}/cf1_change.png",Inches(6.4),"Figure A1. Combined cohort growth of each metric since 2020 (actual
values labelled).")
figure(f"{SCR}/cf2_field.png",Inches(6.4),"Figure A2. Combined field composition (Ecology / both / Forestry) per
year, for each metric.")

doc.add_heading("A.2 Combined taxonomy tree",level=2)
body("Figure A6 charts the full field/sub-field/sub-sub-field taxonomy occupied by the combined cohort's primary
and secondary "
"classifications—substantially larger than the forestry-only tree because ecology reaches broadly across
biology, environmental "
"science and the social sciences.")
figure(f"{SCR}/cf9_tree.png",Inches(6.9),"Figure A6. Tree-chart of the Domain → Field → Sub-Field taxonomy
occupied by the combined "
"Forestry+Ecology cohort (sub-sub-fields with ≥15 records shown). Gold • = target subfields (Forestry, Ecology);
blue » = "
"quantitative; grey • = other.")

doc.add_heading("A.3 Combined representation by continent and country",level=2)
body(f"North America and Europe again dominate ({ccont.get('North America',{}).get('2024',0):,} and "
f"{ccont.get('Europe',{}).get('2024',0):,} scientists in 2024), with the United States far ahead among countries
"
f"({cc['United States']['2024']:,} in 2024), followed by the United Kingdom, Canada and Australia. Tables A3–A4
give the banded "
"country rankings; Figures A3–A5 give per-year trends, the banded profile and the average world maps.")
for m in ["scientists","publications","citations"]:
    per_year_table(C["continent"][m],m,"Continent",f"Table A3{ABC[m]}. Combined continents by {UNIT[m]}, per
year.",keyw=Inches(1.7))
for m in ["scientists","publications","citations"]:
    banded_table(C["country_banded"][m],m,"Country",f"Table A4{ABC[m]}. Combined countries by {UNIT[m]}: top 20
individual, remainder in ~10% bands.",keyw=Inches(2.0))
figure(f"{SCR}/cf3_continent.png",Inches(6.4),"Figure A3. Combined representation by continent, by metric, per
year.")
figure(f"{SCR}/cf4_country.png",Inches(6.4),"Figure A4. Top-8 combined countries by each metric, per release
year.")
figure(f"{SCR}/cf5_country_band.png",Inches(6.2),"Figure A5a. Combined countries by scientists (avg/yr): top-20
individual, remainder banded.")
figure(f"{SCR}/cf7_worldmaps.png",Inches(6.0),"Figure A5b. World choropleths of the five-year-average combined
representation, per metric.")

doc.add_heading("A.4 Combined representation by institution and U.S. state",level=2)
body(f"The most-represented combined institution is {cuni['key']} ({cuni.get('country','')}); U.S. federal
research bodies and land-grant "
"universities are prominent. Table A5 gives the banded institution ranking; Figures A7–A8 give the banded
institution profile and "
"the average U.S. state maps.")
for m in ["scientists","publications","citations"]:
    banded_table(C["university_banded"][m],m,"Institution (country)",f"Table A5{ABC[m]}. Combined institutions by
{UNIT[m]}: top 20 individual, ranks 21–100 in ~10% bands.",keyw=Inches(2.7),foldcountry=True)

```

```

figure(f"{SCR}/cf6_university_band.png",Inches(6.2),"Figure A7. Combined institutions by scientists (avg/yr):
top-20 individual, ranks 21-100 banded.")
figure(f"{SCR}/cf8_usmaps.png",Inches(5.9),"Figure A8. United States choropleths of the five-year-average
combined representation, per metric.")
body("Combined-cohort commentary (preserved). Adding publications and citations to the head-count sharpened
rather than softened the "
      "combined picture; the same Global-North countries and institutions carried most of the papers and citations.
Ecology outnumbered "
      "forestry about five to one, and only about six percent of the combined cohort was quantitatively oriented-lower
than forestry "
      "alone, because ecology is more organism- and field-based. The gap between where forests and biodiversity are
(largely tropical) and "
      "where the most-cited expertise is affiliated (temperate, wealthy, English-publishing) was the combined
analysis's central equity "
      "observation, and the fastest growth appeared in China, Brazil and several mid-sized nations.")

doc.save(OUT)
print("saved:",OUT,"| paragraphs:",len(doc.paragraphs),"| tables:",len(doc.tables))

```

10. build_docx4.py (SHA-256 1473ab1388a087a8...; 415 lines)

```

# -*- coding: utf-8 -*-
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
OUT=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.docx"
M=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
F=json.load(open(f"{SCR}/forestry_data.json",encoding="utf-8"))
MF=M["forestry"]; MC=M["combined"]; TAX=M["taxonomy"]
YEARS=[2020,2021,2022,2023,2024]; YS=[str(y) for y in YEARS]
GREEN="2F5233"; LIGHT="EEF3EA"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"),"Arial")
for i,(sz,col) in enumerate([(16,GREEN),(13,GREEN),(11,"333333")],start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width,sec.page_height=Inches(8.5),Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="",size=11,bold=False,italic=False,align=None,color=None,after=6,before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t,size=10.5,after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
par.paragraph_format.line_spacing=1.13; return par
def shade(cell,hexc):
    tcPr=cell._tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"),"clear");
sh.set(qn("w:fill"),hexc); tcPr.append(sh)
def topborder(cell):
    tcPr=cell._tc.get_or_add_tcPr(); b=OxmlElement("w:tcBorders"); t=OxmlElement("w:top")
    t.set(qn("w:val"),"single"); t.set(qn("w:sz"),"6"); t.set(qn("w:color"),"9AA0A6"); b.append(t);
tcPr.append(b)
def setcell(cell,text,bold=False,white=False,size=8,align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={ "left":WD_ALIGN_PARAGRAPH.LEFT,"center":WD_ALIGN_PARAGRAPH.CENTER,"right":WD_ALIGN_PARAGRAPH.R
IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)
def caption(t): p(t,size=8.5,italic=True,color="555555",after=3)
def figure(path,width,cap):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path,width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    p(cap,size=8.5,italic=True,color="555555",align=WD_ALIGN_PARAGRAPH.CENTER,after=10)
def simple_table(headers,rows,widths,aligns=None,cap=None,hsize=8):

```

```

if cap: caption(cap)
t=doc.add_table(rows=1,cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
aligns=aligns or (["left"]+["right"]*(len(headers)-1))
for j,h in enumerate(headers):
    setcell(t.rows[0].cells[j],h,bold=True,white=True,size=hsize,align="center");
shade(t.rows[0].cells[j],GREEN)
for i,row in enumerate(rows):
    cells=t.add_row().cells
    for j,val in enumerate(row):
        setcell(cells[j],val,size=8,align=aligns[j])
        if i%2==1: shade(cells[j],LIGHT)
for col,w in zip(t.columns,widths):
    for c in col.cells: c.width=w
doc.add_paragraph().paragraph_format.space_after=Pt(2)

def merged_table(items, keyhdr, cap, keyw=Inches(1.55), foldcountry=False):
    """One table, 3 metric-rows per member. Summary col = avg for scientists, total for pubs & citations."""
    caption(cap)
    heads=[keyhdr,"Metric"]+YS+["5-yr"]
    t=doc.add_table(rows=1,cols=len(heads)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
    for j,h in enumerate(heads):
        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=8,align="center"); shade(t.rows[0].cells[j],GREEN)
    for mi,it in enumerate(items):
        key=it["key"]
        if foldcountry and it.get("country"): key=f'{key} ({it["country"]})'
        isband=it.get("band",False)
        s,pub,c=it["s"],it["p"],it["c"]
        triplet=([("Scientists",[f"{int(v):,}" for v in s],f"{sum(s)/5:,.0f}"),
                  ("Publications ('000)",[f"{v/1000:,.1f}" for v in pub],f"{sum(pub)/1000:,.1f}"),
                  ("Citations (M)",[f"{v/1e6:,.2f}" for v in c],f"{sum(c)/1e6:,.2f}"))
        for ri,(mlab,vals,summ) in enumerate(triplet):
            cells=t.add_row().cells
            setcell(cells[0],key if ri==0 else "",bold=(ri==0),size=8,align="left")
            setcell(cells[1],mlab,size=7.5,align="left")
            for j,v in enumerate(vals): setcell(cells[2+j],v,size=8,align="right")
            setcell(cells[7],summ,bold=True,size=8,align="right")
            for j in range(8):
                if isband: shade(cells[j],BAND)
                elif mi%2==1: shade(cells[j],LIGHT)
                if ri==0: topborder(cells[j])
            metricw=Inches(1.25); numw=(CW-keyw-metricw)/6
            for col,w in zip(t.columns,[keyw,metricw]+[numw]*6):
                for c in col.cells: c.width=w
            p("Summary Column: 5-year average for Scientists (largely the same people recur each release); 5-year total for Publications and Citations.",
              size=7.5,italic=True,color="777777",after=8)

def banded_items(bd): return bd["head"]+bd["bands"]
def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=OxmlElement("w:fldChar"); b.set(qn("w:fldCharType"),"begin")
        i=OxmlElement("w:instrText"); i.set(qn("xml:space"),"preserve"); i.text=instr
        e=OxmlElement("w:fldChar"); e.set(qn("w:fldCharType"),"end")
        for x in (b,i,e):
            r=par.add_run(); r._r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
        r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
        r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

TF=MF["totals"]; TC=MC["totals"]
fc={r["key"]:r for r in MF["country"]}
fcont={r["key"]:r for r in MF["continent"]}
fo={r["key"]:r for r in MF["orient"]}
funi=MF["university"][0]
qhits=F["quant_subfield_hits"]
def tot(d,m): return sum(d["totals"][m][y] for y in YS)
def yr(d,m,y): return d["totals"][m][str(y)]
qtot=sum(fo['Quantitative-oriented']['s']); qshare=100*qtot/tot(MF,'scientists')

# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Global Representation of the World's Most-Cited Forestry Scientists by Number of Scientists, Publications and Citations",
  size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)
p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020-2024) of the Stanford / Ioannidis Career-Long Citation Database",

```

```

size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)
p("Analytical Report",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)
p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3-v8)",
size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)
p("Abstract",size=13,bold=True,color=GREEN,after=4)
body(f"This report quantifies the geographic and institutional representation of the world's most highly cited
forestry researchers "
f"along three complementary metrics—the number of listed scientists, the sum of their career publications, and
the sum of their "
f"career citations—for each annual release of the Stanford / Ioannidis "top-scientists" database from data years
2020 to 2024. From "
f"each release we retained every author placed in the Science-Metrix Forestry sub-sub-field (as primary or
secondary specialty), "
f"giving {tot(MF,'scientists'):,} author-year records; in 2024 these {yr(MF,'scientists',2024):,} forestry
scientists carry "
f"{yr(MF,'publications',2024)/1000:,.0f} thousand career publications and {yr(MF,'citations',2024)/1e6:,.1f}
million career citations. "
f"Each dimension—continent, world sub-region, country, first-level subdivision (U.S. state, Canadian province,
Australian state) and "
f"institution (with country)—is reported for all three metrics in a single table per dimension and for every
individual year; the "
f"per-year graphs show the same, and the maps the five-year average. Rankings list the top 20 members
individually and aggregate the "
f"remainder into cumulative bands each holding roughly ten percent of the residual. The taxonomy tree isolates
the "
f"{TAX['n_primary']:,} primary-forestry researchers (whose first subfield is Forestry) and shows the secondary
specialties they "
f"branch into. The forestry list is dominated by the United States and Canada, with the Nordic countries
(Finland and Sweden) "
f"strongly over-represented and specialized agricultural/forestry universities leading; about {qshare:.0f}% of
the cohort is "
f"quantitatively oriented (geomatics/GIS, environmental engineering, statistics, operations research). The
parallel analysis of the "
f"combined Forestry+Ecology population ({tot(MC,'scientists'):,} records) is retained in full in Appendix
A.",size=10)
kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)
r1=kw.add_run("forestry; bibliometrics; scientometrics; citation indicators; publication counts;
Stanford/Ioannidis list; research "
"geography; Science-Metrix taxonomy; choropleth mapping; concentration/banding; quantitative forestry
(biometrics, mensuration, GIS, "
"remote sensing)."); r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

# ===== 1 BACKGROUND =====
doc.add_heading("1. Background, Concept and Related Work",level=1)
body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common
single statistic, the "
"h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To
standardize comparison, "
"Ioannidis and colleagues devised a composite indicator (the "c-score") combining six field-normalized metrics
[2,3] and published "
"an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each subfield) across
22 fields and 174 "
"subfields defined by the Science-Metrix journal ontology [3,4]. The database appears in career-long and single-
year forms and is "
"refreshed annually from Scopus. This report analyses the career-long tables.")
body("Whereas the original database is typically used to look up individuals, this report treats one sub-sub-
field-Forestry-as a "
"population and asks how that population is distributed, not only by head-count but also by the publications and
citations those "
"researchers carry, which measure volume and influence rather than mere presence. Forestry is a small, applied,
globally distributed "
"field at the centre of timber economies, biodiversity, carbon and land-use policy, yet its most-cited expertise
has not, to our "
"knowledge, been mapped in this multi-metric way. Institution-level league tables (Shanghai, Times Higher
Education, Nature Index) "
"and the proprietary Clarivate Highly Cited Researchers list address adjacent questions; the openness and field-
normalization of the "
"Ioannidis data make it well suited to the analysis presented here. Because forestry and ecology overlap heavily
in journals, methods "
"and personnel, a parallel analysis of the combined Forestry+Ecology population is preserved in Appendix A for
readers who prefer the "
"broader frame; the two are compared at each step below.")

# ===== 2 DATA =====
doc.add_heading("2. Data Description",level=1)
body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset
doi:10.17632/btchxktzyw, "

```

```

"versions 3, 5, 6, 7 and 8, corresponding to data years 2020–2024. Each ~75–85 MB spreadsheet holds one row per
ranked scientist "
"(about 186,000–230,000 rows) with the author name, one large-institution affiliation, its ISO country code, the
six component "
"citation metrics and composite c-score, the career publication count (np) and career citation count (nc), and the
Science-Metrix "
"classification: a primary subfield (sm-subfield-1), an optional secondary subfield (sm-subfield-2), and the
parent field (sm-field).")
body(f"For this report we kept every row whose primary or secondary subfield is Forestry, giving
{tot(MF,'scientists'):,} author-year "
f"records ({yr(MF,'scientists',2020):,} in 2020 rising to {yr(MF,'scientists',2024):,} in 2024). Of these,
{TAX['n_primary']:,} have "
"Forestry as their primary subfield (sm-subfield-1)—the core forestry researchers used for the taxonomy in
Section 4.2. The wider "
f"Forestry+Ecology extraction ({tot(MC,'scientists'):,} records), analysed in Appendix A, is delivered with this
report as an Excel "
"workbook and CSV. Table 1 gives the forestry totals for each release year and metric.")
simple_table(["Metric \ Year"]+YS+["5-yr¹"],
[["Scientists (list entries)"]+[f"{yr(MF,'scientists',y):,}" for y in YEARS]+[f"{tot(MF,'scientists')//5:,}"],
["Publications (career papers)"]+[f"{yr(MF,'publications',y):,}" for y in YEARS]+
f"{tot(MF,'publications'):,}"],
["Citations (career citations)"]+[f"{yr(MF,'citations',y):,}" for y in YEARS]+[f"{tot(MF,'citations'):,}"]],
[Inches(2.1)]+[Inches(0.82)]*6, aligns=["left"]+["right"]*6,
cap="Table 1. Forestry cohort totals for each release year, by metric. ¹Summary = 5-year average for scientists,
5-year total for "
"publications and citations. A scientist counts once per release year.")

# ===== 3 METHODS =====
doc.add_heading("3. Methods",level=1)
body("Metrics. Three quantities are tabulated for every group: the number of scientists (rows), publications (sum
of np) and citations "
"(sum of nc). Each dimension is presented in one table carrying all three metrics, with each member shown on
three lines. Because "
"career values are cumulative and each release re-lists largely the same people, the summary column reports the
five-year average for "
"scientists (a mean head-count) but the five-year total for publications and citations; results are given per
release year, and maps "
"show the five-year average.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For
the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus
(a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable ones as
"Unknown." "
"Institutions are labelled with their modal country.")
body("Ranking and banding. For countries and institutions, the top 20 by scientists are listed individually; the
remainder "
"(institutions truncated at the top 100) is aggregated into consecutive bands each accumulating about ten
percent of the residual "
"head-count, so the long tail is summarized without discarding it. Complete, unbanded rankings of all forestry
countries and "
"institutions—by scientists, publications and citations, for every year—are in the companion workbook.")
body("Taxonomy and orientation. The tree (Section 4.2) is built only from the core forestry researchers—those
whose primary subfield "
"(sm-subfield-1) is Forestry—and shows the secondary specialties (sm-subfield-2) they branch into, grouped by
the parent field of "
"each secondary subfield; nothing outside the disciplines actually present in the data is shown. A quantitative-
versus-qualitative "
"split labels an author "quantitative-oriented" when a primary or secondary subfield is a quantitative
discipline: Statistics & "
"Probability; the Mathematical and Computational subfields (Applied/General/Numerical Mathematics, Computation
Theory, Software and "
"other Programming/ICT); Operations Research; every Engineering subfield including Geological & Geomatics
Engineering (GIS, "
"mensuration, surveying) and Environmental Engineering; Artificial Intelligence & Image Processing (remote
sensing); and "
"Bioinformatics. Requested forestry sub-disciplines (Silviculture, Biometrics, Mensuration, Forest Management,
GIS, Remote Sensing) "
"are not distinct Science-Metrix labels and are captured only where they map onto these subfields. Values were
computed with Python "
"(pandas); figures with matplotlib; maps use Natural Earth boundaries [6]. All totals were independently re-
verified against the "
"source releases.")

# ===== 4 RESULTS =====
doc.add_heading("4. Results",level=1)
doc.add_heading("4.1 Overall totals by metric and year",level=2)
gs=100*(yr(MF,'scientists',2024)/yr(MF,'scientists',2020)-1)
gp=100*(yr(MF,'publications',2024)/yr(MF,'publications',2020)-1)
gc=100*(yr(MF,'citations',2024)/yr(MF,'citations',2020)-1)

```

```

body(f"All three metrics rose over the five years (Figure 1). Listed forestry scientists grew {gs:.0f}% "
  f"({yr(MF,'scientists',2020):,}{yr(MF,'scientists',2024):,}); their summed career publications grew {gp:.0f}% "
  f"({yr(MF,'publications',2020)/1000:,.0f}k→{yr(MF,'publications',2024)/1000:,.0f}k); and citations grew {gc:.0f}% "
  f"({yr(MF,'citations',2020)/1e6:,.1f}M→{yr(MF,'citations',2024)/1e6:,.1f}M). Citations grew fastest, reflecting
both new entrants and "
  "the continual accrual of citations to established authors. The combined Forestry+Ecology cohort (Appendix A,
Figure A1) grows at a "
  "very similar pace but is roughly four times larger, because ecology is the more populous field.")
figure(f"{SCR}/ff1_change.png",Inches(6.5),"Figure 1. Scientists, publications and citations for the forestry
cohort, as percent "
  "change from 2020, with the actual value labelled at each point (scientists; thousands of publications; millions
of citations).")

doc.add_heading("4.2 Taxonomy: Forestry and its neighbouring disciplines",level=2)
body(f"Figure 2 isolates the {TAX['n_primary']:,} core forestry researchers—those whose primary Science-Metrix
subfield is Forestry—and "
  "charts the secondary specialties they also work in, grouped by field. The dominant neighbour is Biology
(Ecology far ahead, then "
  "Plant Biology & Botany and Entomology); Agronomy, the chemistry of wood and biomaterials (Polymers), the
enabling technologies "
  "(Biotechnology, Materials, Energy) and the Earth-and-environmental sciences follow. The quantitative,
methodological neighbours are "
  "the Engineering subfields—Geological & Geomatics Engineering (GIS, mensuration and surveying) foremost, then
Environmental "
  "Engineering, with Operations Research and Civil Engineering—together with Statistics & Probability. This is the
taxonomy pertinent "
  "to forestry as practised in these data; disciplines that never co-occur with forestry do not appear.")
figure(f"{SCR}/ff2_tree.png",Inches(6.55),"Figure 2. Taxonomy tree of the core forestry researchers (primary
subfield = Forestry). "
  "Forestry (gold) branches to the parent field (green) of each secondary specialty, then to the secondary sub-
sub-field (blue » = "
  "quantitative). Right-hand numbers are author counts; minor branches are omitted for legibility (see note).")

doc.add_heading("4.3 Quantitative vs qualitative orientation",level=2)
body(f"Applying the secondary-subfield rule, {qtot:,} of {tot(MF,'scientists'):,} author-records "
  f"({100*qtot/tot(MF,'scientists'):.0f}%) are quantitative-oriented—a higher share than in ecology, consistent
with forestry's "
  "inventory, mensuration and engineering traditions. The signal is carried overwhelmingly by geomatics/GIS and
environmental "
  "engineering, with statistics and operations research adding smaller amounts (Table 3). Figure 3 shows both
orientations growing at a "
  "similar rate overall, with the quantitative minority a little faster in the middle years.")
merged_table(MF["orient"],"Orientation","Table 2. Scientists, publications and citations by orientation "
  "(quantitative- vs qualitative-oriented), per release year.",keyw=Inches(1.7))
simple_table(["Quantitative sub-sub-field (secondary)","Author-records 2020-2024"],
  [[k,f"{qhits[k]:,}"] for k in qhits],[Inches(3.6),Inches(3.1)],aligns=["left","right"],
  cap="Table 3. Quantitative sub-sub-fields present among forestry authors and the number of author-records
touching each, 2020-2024. "
  "Operations Research and the Mathematical/Computational and Engineering subfields are included per the analysis
definition.")
figure(f"{SCR}/ff4_orient.png",Inches(6.3),"Figure 3. Quantitative- vs qualitative-oriented forestry scientists,
as percent change "
  "from 2020, with actual head-counts labelled at each point (same style as Figure 1).")

doc.add_heading("4.4 Representation by continent and world region",level=2)
body(f"On every metric, North America and Europe supply the great majority of forestry's most-cited researchers "
  f"(North America {fcont['North America']['s'][-1]:,} scientists in 2024; Europe {fcont['Europe']['s'][-1]:,}),
with Oceania a distant "
  "third and Asia, South America and Africa trailing far behind despite holding most of the world's forests (Table
4, Figure 4; average "
  "maps in Figure 7). The combined Forestry+Ecology picture (Appendix A) is similar but tilts a little more to
Oceania and the United "
  "Kingdom, which are ecology strongholds.")
merged_table(MF["continent"],"Continent","Table 4. Scientists, publications and citations by continent, per
release year.",keyw=Inches(1.5))
merged_table(MF["region"],"World sub-region","Table 5. Scientists, publications and citations by world sub-
region, per release year.",keyw=Inches(1.9))
figure(f"{SCR}/ff3_continent.png",Inches(6.6),"Figure 4. Scientists, publications and citations by continent, per
release year (legend at right).")

doc.add_heading("4.5 Representation by country (top 20 individual, remainder banded)",level=2)
body(f"The United States leads forestry on every metric ({fc['United States']['s'][-1]:,} scientists in 2024),
followed by Canada "
  f"({fc['Canada']['s'][-1]:,}); the Nordic forestry powers Finland and Sweden, with Germany, form the next tier
and rank far above "
  "their population share. Table 6 lists the top 20 countries individually and bands the remainder into ~10%
cumulative groups; Figure 5 "

```

"shows per-year trends, Figure 6 the banded profile, and Figure 7 the average world maps. In the combined data (Appendix A) the "

"United Kingdom and Australia rise above Canada, pulled up by ecology.")

merged_table(banded_items(MF["country_banded"]), "Country", "Table 6. Scientists, publications and citations by country: top 20 "

"individual, remainder in ~10% cumulative bands (ranked by scientists).", keyw=Inches(1.7))

figure(f"{SCR}/ff5_country.png", Inches(6.6), "Figure 5. Scientists, publications and citations by country (top 8), per release year (legend at right).")

figure(f"{SCR}/ff6_country_band.png", Inches(6.2), "Figure 6. Scientists by country (average per year): top-20 individual bars, remainder in ~10% cumulative bands.")

figure(f"{SCR}/ff9_worldmaps.png", Inches(6.0), "Figure 7. Scientists, publications and citations by country – five-year-average world choropleths. Boundaries: Natural Earth [6].")

doc.add_heading("4.6 Subnational representation (United States, Canada, Australia)", level=2)

body("Within the United States, Oregon, California and a pooled block of federal agencies (USDA Forest Service, USGS, USDA-ARS) lead "

"forestry representation, reflecting the country's public-lands forestry research system; the Pacific North-West and Rocky Mountain "

"states are prominent. In Canada, British Columbia and Ontario lead; in Australia, Queensland and Victoria. Table 7 gives the top "

"U.S. states on all three metrics; Figure 8 maps the average. Full per-state, per-year tables for all three countries are in the "

"workbook.")

merged_table(MF["us_state"][:12], "U.S. state / category", "Table 7. Scientists, publications and citations by U.S. state (top 12), per release year.", keyw=Inches(1.7))

figure(f"{SCR}/ff10_usmaps.png", Inches(5.9), "Figure 8. Scientists, publications and citations by U.S. state – five-year-average choropleths (contiguous states; AK, HI omitted from frame).")

doc.add_heading("4.7 Representation by institution (with country; top 20 individual, ranks 21-100 banded)", level=2)

body(f"The most-represented forestry institution is {funi['key']} ({funi.get('country', '')}), followed by other specialized "

"agricultural and forestry universities, North-American land-grant universities and national forest-research agencies. Table 8 lists "

"the top 20 institutions (with country) individually and bands ranks 21-100 into ~10% cumulative groups; Figure 9 shows per-year "

"trends and Figure 10 the banded profile.")

merged_table(banded_items(MF["university_banded"]), "Institution (country)", "Table 8. Scientists, publications and citations by "

"institution: top 20 individual, ranks 21-100 in ~10% cumulative bands (ranked by scientists).", keyw=Inches(2.15), foldcountry=True)

figure(f"{SCR}/ff7_university.png", Inches(6.6), "Figure 9. Scientists, publications and citations by institution (top 7), per release year (legend at right).")

figure(f"{SCR}/ff8_university_band.png", Inches(6.2), "Figure 10. Scientists by institution (average per year): top-20 individual, ranks 21-100 in ~10% cumulative bands.")

===== 5 DISCUSSION =====

doc.add_heading("5. Discussion", level=1)

body("Reading the three metrics together sharpens rather than softens the forestry picture: the countries and institutions that field "

"the most highly cited forestry researchers also carry, by construction, most of the cohort's papers and citations, so influence is "

"at least as concentrated as head-count. The striking feature specific to forestry is the weight of Canada and the Nordic countries: "

"Finland and Sweden, tiny by population, rank among the leaders on all three metrics, reflecting deep national forestry-research "

"traditions and forest-based economies. The gap between where forests are (largely tropical and Global South) and where the "

"recognized forestry expertise is affiliated (temperate, wealthy, North-American and Nordic) is a quantitative entry point into "

"debates on research equity and forestry capacity-building in the tropics.")

body("How forestry compares with ecology (Appendix A). Placing forestry beside the combined Forestry+Ecology population is instructive. "

"Ecology is about four to five times larger, so the combined totals are dominated by it, and the combined geography leans toward "

"large ecology producers—the United Kingdom and Australia move ahead of Canada, and Oceania gains weight—whereas forestry alone is "

"more North-American and Nordic. Forestry is also proportionally more quantitative than the combined cohort (about "

f"{100*qtot/tot(MF, 'scientists'):.0f}% vs roughly six percent), because ecology is more organism- and field-based while forestry "

"carries the geomatics, remote-sensing, mensuration and engineering methods of forest inventory and management. Readers concerned "

"with the biodiversity-and-conservation frame will find the combined tables in Appendix A the better fit; readers concerned with "

"forest management, silviculture and the forest-products economy will find the forestry-only body of the report closer to the mark.")

body("Uses. The multi-metric, multi-scale tables let funders and forestry ministries benchmark national standing on head-count, output "

"and impact separately and track it yearly; forestry schools and research institutes benchmark peers and target recruitment (the "

"banded tail shows where the next tier sits); editors widen reviewer and speaker pools using the country and subnational geography; "

"and science-of-science researchers study concentration and emergence. The quantitative-orientation split flags the growing "

"methodological community-geomatics/GIS, remote sensing, environmental engineering, statistics and operations research-within a "

"traditionally field-based discipline, relevant to curriculum design and to the demand for quantitative skills in modern forest "

"inventory, monitoring and carbon accounting.")

body("Limitations. Affiliation country is current institution, not nationality or origin. Scopus coverage biases toward "

"English-language journals, which likely understates tropical and non-English forestry. Publication and citation sums inherit the "

"skew of citation distributions—a few superstar authors dominate—so group totals are sensitive to outliers. The counting unit is the "

"author-year, so annual changes mix genuine entry with re-computation. Subnational placement leaves some U.S. institutions "

"Unknown" and pools federal agencies. Finally, the quantitative/qualitative split is a coarse proxy: the forestry sub-disciplines "

"named by the reader (Silviculture, Biometrics, Mensuration, Forest Management...) are not separable in Science-Matrix and are "

"approximated by adjacent geomatics, engineering and statistics subfields.")

===== 6 CONCLUSION =====

doc.add_heading("6. Conclusion",level=1)

body(f"Measured by scientists, publications or citations, the population of highly cited forestry researchers is small "

f"({yr(MF,'scientists',2024):,} in 2024) but growing on every metric, and strongly concentrated in the Global North. North America "

"and Europe dominate; the United States and Canada lead among countries, with Finland and Sweden conspicuously over-represented; "

f"Oregon, California and a block of U.S. federal agencies lead subnationally; and {funi['key']} is the most represented institution. "

f"About {100*qtot/tot(MF,'scientists'):.0f}% of the cohort is quantitatively oriented, led by geomatics/GIS and environmental "

"engineering, and that share is slowly rising. Set against the combined Forestry+Ecology population in Appendix A, forestry alone is "

"smaller, more North-American and Nordic, and more methodological, while the combined frame is larger, more Anglo-Oceanian and more "

"organism-based; the two are complementary lenses on the same data, and the citation and publication metrics corroborate the "

"head-count's geography in both.")

===== 7 SUMMARY =====

doc.add_heading("7. Project Summary",level=1)

body(f"Five career-long releases of the Stanford / Ioannidis database (2020–2024) were filtered to the Forestry sub-sub-field "

f"({tot(MF,'scientists'):,} records) and analysed by number of scientists, publications and citations across continent, region, "

"country, subnational unit, institution (with country) and taxonomic neighbourhood. Each dimension is shown in one table carrying all "

"three metrics; rankings list the top 20 individually and band the rest by ~10% cumulative head-count; maps show the five-year "

"average. The broader Forestry+Ecology analysis is retained in Appendix A. Deliverables: this report; a comprehensive workbook with "

"full per-year, per-metric rankings of all forestry countries and institutions plus continent, region, subnational, orientation, "

"taxonomy and map-data sheets (and the combined equivalents); and the combined career database (Excel and CSV).")

===== 8 REFERENCES =====

doc.add_heading("8. References and Data Sources",level=1)

refs=[

"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569–16572. <https://doi.org/10.1073/pnas.0507655102>",

"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite across scientific disciplines. PLoS Biology, 14(7), e1002501. <https://doi.org/10.1371/journal.pbio.1002501>",

"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author database annotated for scientific field. PLoS Biology, 17(8), e3000384. <https://doi.org/10.1371/journal.pbio.3000384>",

"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized citation indicators. PLoS Biology, 18(10), e3000918. <https://doi.org/10.1371/journal.pbio.3000918>",

"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open scientific journal ontology. Proc. 13th ISSI Conference, 66–77. (Science-Matrix classification.)",

"Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 – Countries. <https://www.naturalearthdata.com> (public domain).",

"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized citation indicators" (Version 8) [Data set]. Elsevier Data Repository. <https://doi.org/10.17632/btchxktzyw.8>",

```

]
for i,rf in enumerate(refs,1):
    par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);
par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)
a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)
b=par.add_run(rf); b.font.name="Arial"; b.font.size=Pt(9.5)
p("Data sources.",size=9.5,bold=True,before=4,after=2)
body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions
3/5/6/7/8, via the Elsevier ")
    "Digital Commons / Mendeley Data public API. Taxonomy: field/subfield threshold tables (Table_2) in the same
dataset. Base maps: "
    "Natural Earth (world) and a public US-state boundary file. Analysis and figures: Python (pandas, openpyxl,
matplotlib).",size=9.5)

# ===== APPENDIX A: COMBINED =====
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
doc.add_heading("Appendix A. Combined Forestry + Ecology Analysis",level=1)
body(f"This appendix preserves the broader analysis of the combined Forestry+Ecology population
({tot(MC,'scientists'):,}) author-year "
f"records: the {tot(MF,'scientists'):,} forestry records above plus all ecology records) on which earlier drafts
were based. It is "
    "for readers who prefer to treat forestry and ecology together. Method and metric definitions are identical to
the main report; only "
    "the population differs. The main-text discussion (Section 5) compares the two throughout.")
cc={r["key"]:r for r in MC["country"]}; ccont={r["key"]:r for r in MC["continent"]}
cfl={r["key"]:r for r in MC["field_cls"]}; cuni=MC["university"][0]
doc.add_heading("A.1 Combined totals and field composition",level=2)
body(f"The combined cohort comprises {tot(MC,'scientists'):,} author-year records; in 2024 its
{yr(MC,'scientists',2024):,} scientists "
f"carry {yr(MC,'publications',2024)/1000:,.0f}k publications and {yr(MC,'citations',2024)/1e6:,.1f}M citations.
Ecology dominates: in "
f"2024, {cfl['Ecology']['s'][-1]:,} scientists were in Ecology alone, {cfl['Forestry']['s'][-1]:,} in Forestry
alone and "
f"{cfl['Both']['s'][-1]:,} in both—about five to one ecology-to-forestry, which is why the combined geography
leans toward large "
    "ecology countries.")
simple_table(["Metric \ Year"]+YS+["5-yr1"],
[[["Scientists (list entries)"]+[f"{yr(MC,'scientists',y):,}" for y in YEARS]+[f"{tot(MC,'scientists')/5:,}"],
["Publications (career papers)"]+[f"{yr(MC,'publications',y):,}" for y in YEARS]+
[f"{tot(MC,'publications'):,}"],
["Citations (career citations)"]+[f"{yr(MC,'citations',y):,}" for y in YEARS]+[f"{tot(MC,'citations'):,}"]],
[Inches(2.1)]+[Inches(0.82)]*6,aligns=["left"]+["right"]*6,
cap="Table A1. Combined Forestry+Ecology totals by release year and metric. 1avg for scientists, total for
publications/citations.")
merged_table(MC["field_cls"],"Field","Table A2. Scientists, publications and citations by field composition
(Ecology / Forestry / both), per release year.",keyw=Inches(1.5))
figure(f"{SCR}/cf1_change.png",Inches(6.3),"Figure A1. Scientists, publications and citations for the combined
cohort, as percent change from 2020 (actual values labelled).")
figure(f"{SCR}/cf2_field.png",Inches(6.4),"Figure A2. Field composition (Ecology / both / Forestry) per year, for
each metric.")
doc.add_heading("A.2 Combined representation by continent, country and institution",level=2)
body(f"North America and Europe again dominate ({ccont['North America']['s'][-1]:,} and {ccont['Europe']['s']
[-1]:,} scientists in 2024), "
f"the United States far ahead among countries ({cc['United States']['s'][-1]:,} in 2024), then the United
Kingdom, Canada and "
f"Australia. The most-represented institution is {cuni['key']} ({cuni.get('country','')}). Tables A3–A5 give the
merged continent and "
    "banded country/institution rankings; Figures A3–A6 give per-year trends, banded profiles and the average
maps.")
merged_table(MC["continent"],"Continent","Table A3. Scientists, publications and citations by continent
(combined), per release year.",keyw=Inches(1.5))
merged_table(banded_items(MC["country_banded"],"Country","Table A4. Scientists, publications and citations by
country (combined): top 20 individual, remainder in ~10% bands.",keyw=Inches(1.7))
merged_table(banded_items(MC["university_banded"],"Institution (country)","Table A5. Scientists, publications
and citations by institution (combined): top 20 individual, ranks 21–100 in ~10%
bands.",keyw=Inches(2.15),foldcountry=True)
figure(f"{SCR}/cf3_continent.png",Inches(6.6),"Figure A3. Scientists, publications and citations by continent
(combined), per year (legend at right).")
figure(f"{SCR}/cf4_country.png",Inches(6.6),"Figure A4. Scientists, publications and citations by country
(combined, top 8), per year (legend at right).")
figure(f"{SCR}/cf5_country_band.png",Inches(6.2),"Figure A5. Scientists by country (combined, average/yr): top-20
individual, remainder banded.")
figure(f"{SCR}/cf7_worldmaps.png",Inches(6.0),"Figure A6. Scientists, publications and citations by country
(combined) — five-year-average world choropleths.")
figure(f"{SCR}/cf6_university_band.png",Inches(6.2),"Figure A7. Scientists by institution (combined, average/yr):
top-20 individual, ranks 21–100 banded.")
figure(f"{SCR}/cf8_usmaps.png",Inches(5.9),"Figure A8. Scientists, publications and citations by U.S. state
(combined) — five-year-average choropleths.")

```

```
doc.save(OUT)
```

```
print("saved:",OUT,"| paragraphs:",len(doc.paragraphs),"| tables:",len(doc.tables))
```

11. build_docx5.py (SHA-256 6623a873f691fa06...; 361 lines)

```
# -*- coding: utf-8 -*-
# PRIMARY-FORESTRY report (sm-subfield-1==Forestry). No taxonomy tree, no quant/qual. Appendix A = combined.
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.docx"
M=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
MF=M["forestry"]; MC=M["combined"]
YEARS=[2020,2021,2022,2023,2024]; YS=[str(y) for y in YEARS]
GREEN="2F5233"; LIGHT="EEF3EA"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"),"Arial")
for i,(sz,col) in enumerate([(16,GREEN),(13,GREEN),(11,"333333")],start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width,sec.page_height=Inches(8.5),Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="",size=11,bold=False,italic=False,align=None,color=None,after=6,before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
    r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t,size=10.5,after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
    par.paragraph_format.line_spacing=1.13; return par
def shade(cell,hexc):
    tcPr=cell._tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"),"clear");
    sh.set(qn("w:fill"),hexc); tcPr.append(sh)
def topborder(cell):
    tcPr=cell._tc.get_or_add_tcPr(); b=OxmlElement("w:tcBorders"); t=OxmlElement("w:top")
    t.set(qn("w:val"),"single"); t.set(qn("w:sz"),"6"); t.set(qn("w:color"),"9AA0A6"); b.append(t);
    tcPr.append(b)
def setcell(cell,text,bold=False,white=False,size=8,align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={"left":WD_ALIGN_PARAGRAPH.LEFT,"center":WD_ALIGN_PARAGRAPH.CENTER,"right":WD_ALIGN_PARAGRAPH.R
IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)
def caption(t): p(t,size=8.5,italic=True,color="555555",after=3)
def figure(path,width,cap):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path,width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    p(cap,size=8.5,italic=True,color="555555",align=WD_ALIGN_PARAGRAPH.CENTER,after=10)
def simple_table(headers,rows,widths,aligns=None,cap=None,hsize=8):
    if cap: caption(cap)
    t=doc.add_table(rows=1,cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
    aligns=aligns or (["left"]+["right"]*(len(headers)-1))
    for j,h in enumerate(headers):
        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=hsize,align="center");
    shade(t.rows[0].cells[j],GREEN)
    for i,row in enumerate(rows):
        cells=t.add_row().cells
        for j,val in enumerate(row):
            setcell(cells[j],val,size=8,align=aligns[j])
            if i%2==1: shade(cells[j],LIGHT)
    for col,w in zip(t.columns,widths):
        for c in col.cells: c.width=w
    doc.add_paragraph().paragraph_format.space_after=Pt(2)
def merged_table(items, keyhdr, cap, keyw=Inches(1.55), foldcountry=False):
    caption(cap)
```

```

heads=[keyhdr,"Metric"]+YS+["5-yr¹"]
t=doc.add_table(rows=1,cols=len(heads)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
for j,h in enumerate(heads):
    setcell(t.rows[0].cells[j],h,bold=True,white=True,size=8,align="center"); shade(t.rows[0].cells[j],GREEN)
for mi,it in enumerate(items):
    key=it["key"]
    if foldcountry and it.get("country"): key=f'{key} ({it["country"]})'
    isband=it.get("band",False); s, pub, c=it["s"],it["p"],it["c"]
    triplet=[("Scientists",[f"{int(v):,}" for v in s],f"{sum(s)/5:,.0f}"),
              ("Publications ('000)",[f"{v/1000:,.1f}" for v in pub],f"{sum(pub)/1000:,.1f}"),
              ("Citations (M)",[f"{v/1e6:,.2f}" for v in c],f"{sum(c)/1e6:,.2f}")]
    for ri,(mlab,vals,summ) in enumerate(triplet):
        cells=t.add_row().cells
        setcell(cells[0],key if ri==0 else "",bold=(ri==0),size=8,align="left")
        setcell(cells[1],mlab,size=7.5,align="left")
        for j,v in enumerate(vals): setcell(cells[2+j],v,size=8,align="right")
        setcell(cells[7],summ,bold=True,size=8,align="right")
        for j in range(8):
            if isband: shade(cells[j],BAND)
            elif mi%2==1: shade(cells[j],LIGHT)
            if ri==0: topborder(cells[j])
        metricw=Inches(1.25); numw=(CW-keyw-metricw)/6
        for col,w in zip(t.columns,[keyw,metricw]+[numw]*6):
            for c in col.cells: c.width=w
    p("¹ Summary column: 5-year average for Scientists (largely the same people recur each release); 5-year total
for Publications and Citations.",
      size=7.5,italic=True,color="777777",after=8)
def banded_items(bd): return bd["head"]+bd["bands"]
def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=OxmlElement("w:fldChar"); b.set(qn("w:fldCharType"),"begin")
        i=OxmlElement("w:instrText"); i.set(qn("xml:space"),"preserve"); i.text=instr
        e=OxmlElement("w:fldChar"); e.set(qn("w:fldCharType"),"end")
        for x in (b,i,e):
            r=par.add_run(); r.r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
    r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
    r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

def tot(d,m): return sum(d["totals"][m][y] for y in YS)
def yr(d,m,y): return d["totals"][m][str(y)]
fc={r["key"]:r for r in MF["country"]}; fcont={r["key"]:r for r in MF["continent"]}; funi=MF["university"][0]

# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Global Representation of the World's Most-Cited Forestry Scientists by Number of Scientists, Publications and Citations",
  size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)
p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020-2024) of the Stanford / Ioannidis Career-Long Citation Database",
  size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)
p("Analytical Report",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)
p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktyzw (v3-v8)",
  size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)
p("Abstract",size=13,bold=True,color=GREEN,after=4)
body(f"This report quantifies the geographic and institutional representation of the world's most highly cited forestry researchers "
     f"along three complementary metrics—the number of listed scientists, the sum of their career publications, and the sum of their "
     f"career citations—for each annual release of the Stanford / Ioannidis “top-scientists” database from data years 2020 to 2024. The "
     f"population is defined narrowly: every author whose primary Science-Metrix subfield (sm-subfield-1) is Forestry, treating the "
     f"optional secondary subfield as irrelevant. This gives {tot(MF,'scientists'):,} author-year records; in 2024 these "
     f"{yr(MF,'scientists',2024):,} forestry scientists carry {yr(MF,'publications',2024)/1000:,.0f} thousand career publications and "
     f"{yr(MF,'citations',2024)/1e6:,.1f} million career citations. Each dimension—continent, world sub-region, country, first-level "
     f"subdivision (U.S. state, Canadian province, Australian state) and institution (with country)—is reported for all three metrics in a "
     f"single table and for every individual year; the per-year graphs show the same and the maps the five-year average. Rankings list the "
     f"top 20 members individually and aggregate the remainder into cumulative bands each holding roughly ten percent of the residual. The "
     f"list is dominated by the United States and Canada, with the Nordic countries (Finland and Sweden) strongly over-represented ")

```

```

f"relative to their size and specialized agricultural/forestry universities leading. A parallel analysis of the
broader combined "
f"Forestry+Ecology population ({tot(MC,'scientists'):,} records) is retained for comparison in Appendix
A.",size=10)
kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)
r1=kw.add_run("forestry; bibliometrics; scientometrics; citation indicators; publication counts;
Stanford/Ioannidis list; research "
"geography; choropleth mapping; concentration/banding; forest science."); r1.font.name="Arial";
r1.font.size=Pt(10); r1.italic=True
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

# ===== 1 BACKGROUND =====
doc.add_heading("1. Background, Concept and Related Work",level=1)
body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common
single statistic, the "
"h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To
standardize comparison, "
"Ioannidis and colleagues devised a composite indicator (the "c-score") combining six field-normalized metrics
[2,3] and published "
"an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each subfield) across
22 fields and 174 "
"subfields defined by the Science-Metrix journal ontology [3,4]. Each ranked author carries a primary subfield,
an optional "
"secondary subfield, and a parent field. The database appears in career-long and single-year forms and is
refreshed annually from "
"Scopus. This report analyses the career-long tables.")
body("Whereas the original database is typically used to look up individuals, this report treats one sub-sub-
field-Forestry-as a "
"population and asks how that population is distributed, not only by head-count but also by the publications and
citations those "
"researchers carry, which measure volume and influence rather than mere presence. To keep the population
unambiguously "forestry," "
"we include an author only when Forestry is their primary subfield, disregarding the secondary subfield
entirely. Forestry is a "
"small, applied, globally distributed field at the centre of timber economies, biodiversity, carbon and land-use
policy, yet its "
"most-cited expertise has not, to our knowledge, been mapped in this multi-metric way. Institution-level league
tables (Shanghai, "
"Times Higher Education, Nature Index) and the proprietary Clarivate Highly Cited Researchers list address
adjacent questions; the "
"openness and field-normalization of the Ioannidis data make it well suited to the analysis here. Because
forestry and ecology "
"overlap in practice, a broader combined analysis is retained in Appendix A for readers who prefer that frame.")

# ===== 2 DATA =====
doc.add_heading("2. Data Description",level=1)
body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset
doi:10.17632/btchxktzyw, "
"versions 3, 5, 6, 7 and 8, corresponding to data years 2020-2024. Each ~75-85 MB spreadsheet holds one row per
ranked scientist "
"(about 186,000-230,000 rows) with the author name, one large-institution affiliation, its ISO country code, the
six component "
"citation metrics and composite c-score, the career publication count (np) and career citation count (nc), and
the Science-Metrix "
"classification (primary subfield sm-subfield-1, secondary subfield sm-subfield-2, parent field sm-field).")
body(f"For this report we kept every row whose PRIMARY subfield (sm-subfield-1) is Forestry, treating the
secondary subfield as "
f"irrelevant, giving {tot(MF,'scientists'):,} author-year records ({yr(MF,'scientists',2020):,} in 2020 rising
to "
f"{yr(MF,'scientists',2024):,} in 2024). The broader Forestry+Ecology extraction analysed in Appendix A "
f"({tot(MC,'scientists'):,} records: any author whose primary or secondary subfield is Forestry or Ecology) is
delivered with this "
"report as an Excel workbook and CSV. Table 1 gives the forestry totals for each release year and metric.")
simple_table(["Metric \\ Year"]+YS+["5-yr¹"],
[["Scientists (list entries)"]+[f"{yr(MF,'scientists',y):,}" for y in YEARS]+[f"{tot(MF,'scientists')//5:,}"],
["Publications (career papers)"]+[f"{yr(MF,'publications',y):,}" for y in YEARS]+
f"{tot(MF,'publications'):,}"],
["Citations (career citations)"]+[f"{yr(MF,'citations',y):,}" for y in YEARS]+[f"{tot(MF,'citations'):,}"],
[Inches(2.1)]+[Inches(0.82)]*6, aligns=["left"]+["right"]*6,
cap="Table 1. Forestry cohort totals (primary subfield = Forestry) for each release year, by metric. ¹Summary =
5-year average for "
"scientists, 5-year total for publications and citations. A scientist counts once per release year.")

# ===== 3 METHODS =====
doc.add_heading("3. Methods",level=1)
body("Population. An author is included when the primary Science-Metrix subfield (sm-subfield-1) is Forestry; the
secondary subfield "
"(sm-subfield-2) is not used to include, exclude or characterize anyone. This yields a clean "core forestry"
population rather than "
"the wider set of authors who merely list forestry as a secondary interest.")

```

```

body("Metrics. Three quantities are tabulated for every group: the number of scientists (rows), publications (sum
of np) and citations "
"(sum of nc). Each dimension is presented in one table carrying all three metrics, with each member on three
lines. Because career "
"values are cumulative and each release re-lists largely the same people, the summary column reports the five-
year average for "
"scientists (a mean head-count) but the five-year total for publications and citations; results are given per
release year, and maps "
"show the five-year average.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For
the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus
(a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable ones as
"Unknown." "
"Institutions are labelled with their modal country.")
body("Ranking and banding. For countries and institutions, the top 20 by scientists are listed individually; the
remainder "
"(institutions truncated at the top 100) is aggregated into consecutive bands each accumulating about ten
percent of the residual "
"head-count, so the long tail is summarized without discarding it. Complete, unbanded rankings of all forestry
countries and "
"institutions-by scientists, publications and citations, for every year-are in the companion workbook. Values
were computed with "
"Python (pandas); figures with matplotlib; maps use Natural Earth boundaries [6]. All totals were independently
re-verified against "
"the source releases.")

# ===== 4 RESULTS =====
doc.add_heading("4. Results",level=1)
doc.add_heading("4.1 Overall totals by metric and year",level=2)
gs=100*(yr(MF,'scientists',2024)/yr(MF,'scientists',2020)-1)
gp=100*(yr(MF,'publications',2024)/yr(MF,'publications',2020)-1)
gc=100*(yr(MF,'citations',2024)/yr(MF,'citations',2020)-1)
body(f"All three metrics rose over the five years (Figure 1). Listed forestry scientists grew {gs:.0f}% "
f"({yr(MF,'scientists',2020):,}->{yr(MF,'scientists',2024):,}); their summed career publications grew {gp:.0f}% "
f"({yr(MF,'publications',2020)/1000:,.0f}k->{yr(MF,'publications',2024)/1000:,.0f}k); and citations grew {gc:.0f}
% "
f"({yr(MF,'citations',2020)/1e6:,.1f}M->{yr(MF,'citations',2024)/1e6:,.1f}M). Citations grew fastest, reflecting
both new entrants and "
"the continual accrual of citations to established authors. The broader Forestry+Ecology cohort (Appendix A)
grows at a similar pace "
"but is several times larger, because it also absorbs ecology and authors who list forestry only secondarily.")
figure(f"{SCR}/ff1_change.png",Inches(6.5),"Figure 1. Scientists, publications and citations for the forestry
cohort, as percent "
"change from 2020, with the actual value labelled at each point (scientists; thousands of publications; millions
of citations).")

doc.add_heading("4.2 Representation by continent and world region",level=2)
body(f"On every metric, North America and Europe supply the great majority of forestry's most-cited researchers "
f"(North America {fcont['North America']['s'][-1]:,} scientists in 2024; Europe {fcont['Europe']['s'][-1]:,}),
with Oceania a distant "
"third and Asia, South America and Africa trailing far behind despite holding most of the world's forests (Table
2, Figure 2; average "
"maps in Figure 5). The broader Forestry+Ecology picture (Appendix A) is similar but tilts a little more to
Oceania and the United "
"Kingdom, which are ecology strongholds.")
merged_table(MF["continent"],"Continent","Table 2. Scientists, publications and citations by continent, per
release year.",keyw=Inches(1.5))
merged_table(MF["region"],"World sub-region","Table 3. Scientists, publications and citations by world sub-
region, per release year.",keyw=Inches(1.9))
figure(f"{SCR}/ff3_continent.png",Inches(6.6),"Figure 2. Scientists, publications and citations by continent, per
release year (legend at right).")

doc.add_heading("4.3 Representation by country (top 20 individual, remainder banded)",level=2)
body(f"The United States leads forestry on every metric ({fc['United States']['s'][-1]:,} scientists in 2024),
followed by Canada "
f"({fc['Canada']['s'][-1]:,}); the Nordic forestry powers Finland and Sweden, with Germany, form the next tier
and rank far above "
"their population share. Table 4 lists the top 20 countries individually and bands the remainder into ~10%
cumulative groups; Figure 3 "
"shows per-year trends, Figure 4 the banded profile, and Figure 5 the average world maps. In the broader data
(Appendix A) the United "
"Kingdom and Australia rise above Canada, pulled up by ecology.")
merged_table(banded_items(MF["country_banded"],"Country","Table 4. Scientists, publications and citations by
country: top 20 "
"individual, remainder in ~10% cumulative bands (ranked by scientists).",keyw=Inches(1.7))
figure(f"{SCR}/ff5_country.png",Inches(6.6),"Figure 3. Scientists, publications and citations by country (top 8),
per release year (legend at right).")

```

```
figure(f"{SCR}/ff6_country_band.png",Inches(6.2),"Figure 4. Scientists by country (average per year): top-20 individual bars, remainder in ~10% cumulative bands.")
figure(f"{SCR}/ff9_worldmaps.png",Inches(6.0),"Figure 5. Scientists, publications and citations by country – five-year-average world choropleths. Boundaries: Natural Earth [6].")
```

```
doc.add_heading("4.4 Subnational representation (United States, Canada, Australia)",level=2)
body("Within the United States, Oregon, California and a pooled block of federal agencies (USDA Forest Service, USGS, USDA-ARS) lead "
"forestry representation, reflecting the country's public-lands forestry research system; the Pacific North-West and Rocky Mountain "
"states are prominent. In Canada, British Columbia and Ontario lead; in Australia, Queensland and Victoria. Table 5 gives the top "
"U.S. states on all three metrics; Figure 6 maps the average. Full per-state, per-year tables for all three countries are in the "
"workbook.")
merged_table(MF["us_state"][:12],"U.S. state / category","Table 5. Scientists, publications and citations by U.S. state (top 12), per release year.",keyw=Inches(1.7))
figure(f"{SCR}/ff10_usmaps.png",Inches(5.9),"Figure 6. Scientists, publications and citations by U.S. state – five-year-average choropleths (contiguous states; AK, HI omitted from frame).")
```

```
doc.add_heading("4.5 Representation by institution (with country; top 20 individual, ranks 21-100 banded)",level=2)
body(f"The most-represented forestry institution is {funi['key']} ({funi.get('country','')}), followed by other specialized "
"agricultural and forestry universities, North-American land-grant universities and national forest-research agencies. Table 6 lists "
"the top 20 institutions (with country) individually and bands ranks 21-100 into ~10% cumulative groups; Figure 7 shows per-year "
"trends and Figure 8 the banded profile.")
merged_table(banded_items(MF["university_banded"]), "Institution (country)", "Table 6. Scientists, publications and citations by "
"institution: top 20 individual, ranks 21-100 in ~10% cumulative bands (ranked by scientists).",keyw=Inches(2.15),foldcountry=True)
figure(f"{SCR}/ff7_university.png",Inches(6.6),"Figure 7. Scientists, publications and citations by institution (top 7), per release year (legend at right).")
figure(f"{SCR}/ff8_university_band.png",Inches(6.2),"Figure 8. Scientists by institution (average per year): top-20 individual, ranks 21-100 in ~10% cumulative bands.")
```

```
# ===== 5 DISCUSSION =====
```

```
doc.add_heading("5. Discussion",level=1)
body("Reading the three metrics together sharpens rather than softens the forestry picture: the countries and institutions that field "
"the most highly cited forestry researchers also carry, by construction, most of the cohort's papers and citations, so influence is "
"at least as concentrated as head-count. The striking feature specific to forestry is the weight of Canada and the Nordic countries: "
"Finland and Sweden, tiny by population, rank among the leaders on all three metrics, reflecting deep national forestry-research "
"traditions and forest-based economies. The gap between where forests are (largely tropical and Global South) and where the "
"recognized forestry expertise is affiliated (temperate, wealthy, North-American and Nordic) is a quantitative entry point into "
"debates on research equity and forestry capacity-building in the tropics.")
body("How the core-forestry frame compares with the broader one (Appendix A). Restricting the population to authors whose primary "
"subfield is Forestry produces a smaller, sharper cohort than the combined Forestry+Ecology set: ecology is far more populous, so the "
"combined totals are dominated by it and its geography leans toward large ecology producers—the United Kingdom and Australia move "
"ahead of Canada, and Oceania gains weight—whereas core forestry is more North-American and Nordic. Readers concerned with "
"biodiversity and conservation will find the combined tables in Appendix A the better fit; readers concerned with forest management, "
"silviculture and the forest-products economy will find this core-forestry body of the report closer to the mark.")
body("Uses. The multi-metric, multi-scale tables let funders and forestry ministries benchmark national standing on head-count, output "
"and impact separately and track it yearly; forestry schools and research institutes benchmark peers and target recruitment (the "
"banded tail shows where the next tier sits); editors widen reviewer and speaker pools using the country and subnational geography; "
"and science-of-science researchers study concentration and emergence.")
body("Limitations. Affiliation country is current institution, not nationality or origin. Scopus coverage biases toward "
"English-language journals, which likely understates tropical and non-English forestry. Publication and citation sums inherit the "
"skew of citation distributions—a few superstar authors dominate—so group totals are sensitive to outliers. The counting unit is the "
"author-year, so annual changes mix genuine entry with re-computation. Restricting to the primary subfield excludes researchers who "
```

```

"do substantial forestry work under a different primary label (they appear in the combined analysis of Appendix
A). Subnational "
"placement leaves some U.S. institutions "Unknown" and pools federal agencies.")

# ===== 6 CONCLUSION =====
doc.add_heading("6. Conclusion",level=1)
body(f"Measured by scientists, publications or citations, the population of core forestry researchers (primary
subfield = Forestry) is "
f"small ({yr(MF,'scientists',2024):,} in 2024) but growing on every metric, and strongly concentrated in the
Global North. North "
"America and Europe dominate; the United States and Canada lead among countries, with Finland and Sweden
conspicuously "
f"over-represented; Oregon, California and a block of U.S. federal agencies lead subnationally; and
{funi['key']} is the most "
"represented institution. Set against the broader Forestry+Ecology population in Appendix A, core forestry is
smaller, more "
"North-American and Nordic, while the combined frame is larger and more Anglo-Oceanian; the two are
complementary lenses on the same "
"data, and the citation and publication metrics corroborate the head-count's geography in both.")

# ===== 7 SUMMARY =====
doc.add_heading("7. Project Summary",level=1)
body(f"Five career-long releases of the Stanford / Ioannidis database (2020-2024) were filtered to authors whose
primary subfield is "
f"Forestry ({tot(MF,'scientists'):,} records) and analysed by number of scientists, publications and citations
across continent, "
"region, country, subnational unit and institution (with country). Each dimension is shown in one table carrying
all three metrics; "
"rankings list the top 20 individually and band the rest by ~10% cumulative head-count; maps show the five-year
average. The broader "
"Forestry+Ecology analysis is retained in Appendix A. Deliverables: this report; a comprehensive workbook with
full per-year, "
"per-metric rankings of all forestry countries and institutions plus continent, region, subnational and map-data
sheets (and the "
"combined equivalents); and the combined career database (Excel and CSV).")

# ===== 8 REFERENCES =====
doc.add_heading("8. References and Data Sources",level=1)
refs=[
"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569-
16572. https://doi.org/10.1073/pnas.0507655102",
"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite
across scientific disciplines. PLoS Biology, 14(7), e1002501. https://doi.org/10.1371/journal.pbio.1002501",
"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author
database annotated for scientific field. PLoS Biology, 17(8), e3000384.
https://doi.org/10.1371/journal.pbio.3000384",
"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized
citation indicators. PLoS Biology, 18(10), e3000918. https://doi.org/10.1371/journal.pbio.3000918",
"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open
scientific journal ontology. Proc. 13th ISSI Conference, 66-77. (Science-Metrix classification.)",
"Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 - Countries.
https://www.naturalearthdata.com (public domain).",
"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized
citation indicators" (Version 8) [Data set]. Elsevier Data Repository. https://doi.org/10.17632/btchxktzyw.8",
]
for i,rf in enumerate(refs,1):
    par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);
par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)
a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)
b=par.add_run(rf); b.font.name="Arial"; b.font.size=Pt(9.5)
p("Data sources.",size=9.5,bold=True,before=4,after=2)
body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions
3/5/6/7/8, via the Elsevier "
"Digital Commons / Mendeley Data public API. Base maps: Natural Earth (world) and a public US-state boundary
file. Analysis and "
"figures: Python (pandas, openpyxl, matplotlib).",size=9.5)

# ===== APPENDIX A: COMBINED =====
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
doc.add_heading("Appendix A. Combined Forestry + Ecology Analysis",level=1)
body(f"This appendix presents the broader Forestry+Ecology population for comparison: every author whose primary
or secondary subfield "
f"is Forestry or Ecology ({tot(MC,'scientists'):,} author-year records). Unlike the main report—which is
restricted to authors whose "
"primary subfield is Forestry—this appendix uses the wider either-subfield definition and includes all of
ecology. Method and metric "
"definitions are otherwise identical; only the population differs. Section 5 compares the two frames.")
cc={r["key"]:r for r in MC["country"]}; ccont={r["key"]:r for r in MC["continent"]}
cfl={r["key"]:r for r in MC["field_cls"]}; cuni=MC["university"][0]

```

```

doc.add_heading("A.1 Combined totals and field composition",level=2)
body(f"The combined cohort comprises {tot(MC,'scientists'):,} author-year records; in 2024 its
{yr(MC,'scientists',2024):,} scientists "
f"carry {yr(MC,'publications',2024)/1000:,.0f}k publications and {yr(MC,'citations',2024)/1e6:,.1f}M citations.
Ecology dominates: in "
f"2024, {cfl['Ecology']['s'][-1]:,} scientists were in Ecology, {cfl['Forestry']['s'][-1]:,} in Forestry and
{cfl['Both']['s'][-1]:,} "
"in both, about five to one ecology-to-forestry, which is why the combined geography leans toward large ecology
countries.")
simple_table(["Metric \ Year"]+YS+["5-yr¹"],
[["Scientists (list entries)"]+[f"{yr(MC,'scientists',y):,}" for y in YEARS]+[f"{tot(MC,'scientists')//5:,}"],
[["Publications (career papers)"]+[f"{yr(MC,'publications',y):,}" for y in YEARS]+
[f"{tot(MC,'publications'):,}"],
[["Citations (career citations)"]+[f"{yr(MC,'citations',y):,}" for y in YEARS]+[f"{tot(MC,'citations'):,}"]],
[Inches(2.1)]+[Inches(0.82)]*6,aligns=["left"]+["right"]*6,
cap="Table A1. Combined Forestry+Ecology totals by release year and metric. 'avg for scientists, total for
publications/citations.")
merged_table(MC["field_cls"],"Field","Table A2. Scientists, publications and citations by field composition
(Ecology / Forestry / both), per release year.",keyw=Inches(1.5))
figure(f"{SCR}/cf1_change.png",Inches(6.3),"Figure A1. Scientists, publications and citations for the combined
cohort, as percent change from 2020 (actual values labelled).")
figure(f"{SCR}/cf2_field.png",Inches(6.4),"Figure A2. Field composition (Ecology / both / Forestry) per year, for
each metric.")
doc.add_heading("A.2 Combined representation by continent, country and institution",level=2)
body(f"North America and Europe again dominate ({ccont['North America']['s'][-1]:,} and {ccont['Europe']['s']
[-1]:,} scientists in 2024), "
f"the United States far ahead among countries ({cc['United States']['s'][-1]:,} in 2024), then the United
Kingdom, Canada and "
f"Australia. The most-represented institution is {cuni['key']} ({cuni.get('country','')}). Tables A3-A5 give the
merged continent and "
"banded country/institution rankings; Figures A3-A8 give per-year trends, banded profiles and the average
maps.")
merged_table(MC["continent"],"Continent","Table A3. Scientists, publications and citations by continent
(combined), per release year.",keyw=Inches(1.5))
merged_table(banded_items(MC["country_banded"]), "Country","Table A4. Scientists, publications and citations by
country (combined): top 20 individual, remainder in ~10% bands.",keyw=Inches(1.7))
merged_table(banded_items(MC["university_banded"]), "Institution (country)","Table A5. Scientists, publications
and citations by institution (combined): top 20 individual, ranks 21-100 in ~10%
bands.",keyw=Inches(2.15),foldcountry=True)
figure(f"{SCR}/cf3_continent.png",Inches(6.6),"Figure A3. Scientists, publications and citations by continent
(combined), per year (legend at right).")
figure(f"{SCR}/cf4_country.png",Inches(6.6),"Figure A4. Scientists, publications and citations by country
(combined, top 8), per year (legend at right).")
figure(f"{SCR}/cf5_country_band.png",Inches(6.2),"Figure A5. Scientists by country (combined, average/yr): top-20
individual, remainder banded.")
figure(f"{SCR}/cf7_worldmaps.png",Inches(6.0),"Figure A6. Scientists, publications and citations by country
(combined) – five-year-average world choropleths.")
figure(f"{SCR}/cf6_university_band.png",Inches(6.2),"Figure A7. Scientists by institution (combined, average/yr):
top-20 individual, ranks 21-100 banded.")
figure(f"{SCR}/cf8_usmaps.png",Inches(5.9),"Figure A8. Scientists, publications and citations by U.S. state
(combined) – five-year-average choropleths.")

doc.save(OUT)
print("saved:",OUT,"| paragraphs:",len(doc.paragraphs),"| tables:",len(doc.tables))

```

12. build_docx6.bak.py (SHA-256 c2f2b172f7a9d730...; 403 lines)

```

# -*- coding: utf-8 -*-
# 2-GROUP forestry research report. Group F = AM=Forestry (3,271); Group E = AM=Ecology & AO=Forestry (615).
Union 3,886.
# Multi-root AQ->AM->AO taxonomy. Analyses separate the two groups. Appendix A = combined Forestry+Ecology.
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
OUT=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.docx"
M=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
MF=M["forestry"]; MFF=M["forestry_F"]; MFE=M["forestry_E"]; MC=M["combined"]
YEARS=[2020,2021,2022,2023,2024]; YS=[str(y) for y in YEARS]
GREEN="2F5233"; LIGHT="EEF3EA"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)

```

```

st.element.rPr.rFonts.set(qn("w: eastAsia"), "Arial")
for i, (sz, col) in enumerate([(16, GREEN), (13, GREEN), (11, "333333")], start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
    h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width, sec.page_height=Inches(8.5), Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="", size=11, bold=False, italic=False, align=None, color=None, after=6, before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
    r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t, size=10.5, after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
    par.paragraph_format.line_spacing=1.13; return par
def shade(cell, hexc):
    tcPr=cell._tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"), "clear");
    sh.set(qn("w:fill"), hexc); tcPr.append(sh)
def topborder(cell):
    tcPr=cell._tc.get_or_add_tcPr(); b=OxmlElement("w:tcBorders"); t=OxmlElement("w:top")
    t.set(qn("w:val"), "single"); t.set(qn("w:sz"), "6"); t.set(qn("w:color"), "9AA0A6"); b.append(t);
    tcPr.append(b)
def setcell(cell, text, bold=False, white=False, size=8, align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={"left":WD_ALIGN_PARAGRAPH.LEFT, "center":WD_ALIGN_PARAGRAPH.CENTER, "right":WD_ALIGN_PARAGRAPH.R
    IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)
def tight_margins(t, m=18):
    tblPr=t._tbl.get_or_add_tblPr(); mar=OxmlElement("w:tblCellMar")
    for side in ("left", "right", "top", "bottom"):
        e=OxmlElement(f"w:{side}"); e.set(qn("w:w"), str(m)); e.set(qn("w:type"), "dxa"); mar.append(e)
    tblPr.append(mar)
def caption(t): p(t, size=8.5, italic=True, color="555555", after=3)
def figure(path, width, cap):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path, width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    p(cap, size=8.5, italic=True, color="555555", align=WD_ALIGN_PARAGRAPH.CENTER, after=10)
def simple_table(headers, rows, widths, aligns=None, cap=None, hsize=9):
    if cap: caption(cap)
    t=doc.add_table(rows=1, cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"; tight_margins(t)
    aligns=aligns or (["left"]*len(headers)-1)
    for j, h in enumerate(headers):
        setcell(t.rows[0].cells[j], h, bold=True, white=True, size=hsize, align="center");
    shade(t.rows[0].cells[j], GREEN)
    for i, row in enumerate(rows):
        cells=t.add_row().cells
        for j, val in enumerate(row):
            setcell(cells[j], val, size=9.5, align=aligns[j])
            if i%2==1: shade(cells[j], LIGHT)
    for col, w in zip(t.columns, widths):
        for c in col.cells: c.width=w
    doc.add_paragraph().paragraph_format.space_after=Pt(2)
def merged_table(items, keyhdr, cap, keyw=Inches(1.5), foldcountry=False):
    caption(cap)
    heads=[keyhdr, "Metric"]+YS+["5-yr1"]
    t=doc.add_table(rows=1, cols=len(heads)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"; tight_margins(t)
    for j, h in enumerate(heads):
        setcell(t.rows[0].cells[j], h, bold=True, white=True, size=9, align="center"); shade(t.rows[0].cells[j], GREEN)
    for mi, it in enumerate(items):
        key=it["key"]
        if foldcountry and it.get("country"): key=f'{key} ({it["country"]})'
        isband=it.get("band", False); s, pub, c=it["s"], it["p"], it["c"]
        triplet=[("Scientists", [f"{int(v):,}" for v in s], f"{sum(s)/5:,.0f}"),
            ("Publications ('000)", [f"{v/1000:,.1f}" for v in pub], f"{sum(pub)/1000:,.1f}"),
            ("Citations (M)", [f"{v/1e6:,.2f}" for v in c], f"{sum(c)/1e6:,.2f}")]
    for ri, (mlab, vals, summ) in enumerate(triplet):
        cells=t.add_row().cells
        setcell(cells[0], key if ri==0 else "", bold=(ri==0), size=10, align="left")
        setcell(cells[1], mlab, size=8.5, align="left")
        for j, v in enumerate(vals): setcell(cells[2+j], v, size=10, align="right")
        setcell(cells[7], summ, bold=True, size=10, align="right")
        for j in range(8):
            if isband: shade(cells[j], BAND)

```

```

        elif mi%2==1: shade(cells[j],LIGHT)
        if ri==0: topborder(cells[j])
        # fixed narrow number columns, slightly wider summary, remaining width -> wide text (col 1)
        metricw=Inches(1.22); yearw=Inches(0.52); sumw=Inches(0.66); keyw=CW-metricw-5*yearw-sumw
        for col,w in zip(t.columns,[keyw,metricw]+[yearw]*5+[sumw]):
            for c in col.cells: c.width=w
        p("Summary column: 5-year average for Scientists (largely the same people recur each release); 5-year total
for Publications and Citations.",
        size=7.5,italic=True,color="777777",after=8)
def banded_items(bd): return bd["head"]+bd["bands"]
def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=OxmlElement("w:fldChar"); b.set(qn("w:fldCharType"),"begin")
        i=OxmlElement("w:instrText"); i.set(qn("xml:space"),"preserve"); i.text=instr
        e=OxmlElement("w:fldChar"); e.set(qn("w:fldCharType"),"end")
        for x in (b,i,e):
            r=par.add_run(); r._r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
        r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
        r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

def tot(d,m): return sum(d["totals"][m][y] for y in YS)
def yr(d,m,y): return d["totals"][m][str(y)]
fc={r["key"]:r for r in MF["country"]}; fcont={r["key"]:r for r in MF["continent"]}; funi=MF["university"][0]
nF=MFF["n"]; nE=MFE["n"]; nU=MF["n"]

# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Global Representation of the World's Most-Cited Forestry Scientists by Number of Scientists, Publications and
Citations",
    size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)
p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020-2024) of the Stanford / Ioannidis
Career-Long Citation Database",
    size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)
p("Analytical Report",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)
p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)
p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3-v8)",
    size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)
p("Abstract",size=13,bold=True,color=GREEN,after=4)
body(f"This report quantifies the geographic and institutional representation of the world's most highly cited
forestry researchers "
f"along three complementary metrics—the number of listed scientists, the sum of their career publications, and
the sum of their "
f"career citations—for each annual release of the Stanford / Ioannidis “top-scientists” database from data years
2020 to 2024. The "
f"forestry research cohort is assembled from two groups: the Forestry group, whose primary Science-Metrix
subfield is Forestry "
f"({nF:,} records), and the Ecology group, whose primary subfield is Ecology but whose secondary subfield is
Forestry ({nE:,} "
f"records), together {nU:,} author-year records analysed as two separate groups throughout. In 2024 these
{yr(MF,'scientists',2024):,} "
f"scientists carry {yr(MF,'publications',2024)/1000:,.0f} thousand career publications and
{yr(MF,'citations',2024)/1e6:,.1f} million "
f"career citations. Each dimension—continent, world sub-region, country, first-level subdivision (U.S. state,
Canadian province, "
f"Australian state) and institution (with country)—is reported for all three metrics in a single table and for
every year; maps show "
f"the five-year average. Rankings list the top 20 members individually and band the remainder in ~10% cumulative
groups. A "
f"multi-root taxonomy tree organizes the cohort by field (sm-field) → primary subfield (sm-subfield-1) →
secondary subfield "
f"(sm-subfield-2), which cleanly separates the Forestry-rooted and Ecology-rooted groups. The list is dominated
by the United States "
f"and Canada, with Finland and Sweden strongly over-represented. The broader combined Forestry+Ecology
population "
f"({tot(MC,'scientists'):,} records) is retained for comparison in Appendix A.",size=10)
kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)
r1=kw.add_run("forestry; ecology; bibliometrics; scientometrics; citation indicators; publication counts;
Stanford/Ioannidis list; "
"research geography; Science-Metrix taxonomy; choropleth mapping; concentration/banding.");
r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

# ===== 1 BACKGROUND =====
doc.add_heading("1. Background, Concept and Related Work",level=1)
body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common
single statistic, the "
"h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To
standardize comparison, "

```

"Ioannidis and colleagues devised a composite indicator (the "c-score") combining six field-normalized metrics [2,3] and published "

"an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each subfield) across 22 fields and 174 "

"subfields defined by the Science-Metrix journal ontology [3,4]. Each ranked author carries a primary subfield (sm-subfield-1), an "

"optional secondary subfield (sm-subfield-2), and a field-level label (sm-field). The database appears in career-long and single-year "

"forms and is refreshed annually from Scopus. This report analyses the career-long tables.")

body("Whereas the original database is typically used to look up individuals, this report treats forestry as a population and asks how "

"that population is distributed, not only by head-count but also by the publications and citations those researchers carry. Forestry "

"is a small, applied, globally distributed field at the centre of timber economies, biodiversity, carbon and land-use policy. Because "

"forestry and ecology overlap heavily in journals, methods and personnel, a subset of the most-cited forestry researchers are "

"classified primarily as ecologists but secondarily as foresters; the report keeps these as a distinct "Ecology group" alongside the "

"primary-forestry "Forestry group," and separates the two throughout. A broader combined Forestry+Ecology analysis is retained in "

"Appendix A.")

```
# ===== 2 DATA =====
doc.add_heading("2. Data Description",level=1)
body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset doi:10.17632/btchxktzyw, "
"versions 3, 5, 6, 7 and 8, for data years 2020-2024. Each ~75-85 MB spreadsheet holds one row per ranked scientist (about "
"186,000-230,000 rows) with the author name, one large-institution affiliation, its ISO country code, the six component citation "
"metrics and composite c-score, the career publication count (np) and career citation count (nc), and the Science-Metrix "
"classification: primary subfield (sm-subfield-1, column AM), secondary subfield (sm-subfield-2, AO) and field (sm-field, AQ).")
body(f"Selection. A record enters the forestry research cohort under either of two rules, kept as separate groups: "
f"(1) Forestry group - sm-subfield-1 = Forestry ({nF:,} records); (2) Ecology group - sm-subfield-1 = Ecology AND sm-subfield-2 = "
f"Forestry ({nE:,} records). The union is {nU:,} author-year records. We verified against the raw releases that sm-field (AQ) is a "
"field-level label taking only the 20 Science-Metrix field names and is never literally "Forestry" (the forestry field value is the "
"broader "Agriculture, Fisheries & Forestry"); AQ therefore adds no records beyond the two subfield rules but is used as the set of "
"roots for the taxonomy tree (Section 4.2). The broader Forestry+Ecology extraction analysed in Appendix A "
f"({tot(MC,'scientists'):,} records: any author with Forestry or Ecology in either subfield) is delivered with this report as an Excel "
"workbook and CSV. Table 1 gives the cohort totals per year and metric.")
simple_table(["Metric \ Year"]+YS+["5-yr¹"],
[["Scientists (list entries)"]+[f"{yr(MF,'scientists'),y}:" for y in YEARS]+[f"{tot(MF,'scientists')//5:,}"],
["Publications (career papers)"]+[f"{yr(MF,'publications'),y}:" for y in YEARS]+
[f"{tot(MF,'publications'):,}"],
["Citations (career citations)"]+[f"{yr(MF,'citations'),y}:" for y in YEARS]+[f"{tot(MF,'citations'):,}"]],
[Inches(2.1)]+[Inches(0.82)]*6, aligns=["left"]+["right"]*6,
cap=f"Table 1. Forestry research cohort totals (Forestry group {nF:,} + Ecology group {nE:,} = {nU:,}) for each release year, by "
"metric. ¹Summary = 5-year average for scientists, 5-year total for publications and citations.")
```

```
# ===== 3 METHODS =====
doc.add_heading("3. Methods",level=1)
body("Population and groups. The cohort is the union of the two selection rules above; every table and figure distinguishes the "
"Forestry group (primary subfield Forestry) from the Ecology group (primary Ecology, secondary Forestry). The overall geographic and "
"institutional rankings pool both groups; Section 4.2 and the workbook give the group split.")
body("Metrics. Three quantities are tabulated for every group: the number of scientists (rows), publications (sum of np) and citations "
"(sum of nc). Each dimension is presented in one table carrying all three metrics, with each member on three lines. Because career "
"values are cumulative and each release re-lists largely the same people, the summary column reports the five-year average for "
"scientists but the five-year total for publications and citations; results are per release year, and maps show the five-year average.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus (a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable ones as "Unknown.""")
```

```

body("Ranking and banding. For countries and institutions, the top 20 by scientists are listed individually; the remainder "
"(institutions truncated at the top 100) is aggregated into consecutive bands each holding about ten percent of the residual "
"head-count. Taxonomy. The tree (Figure 2) has multiple roots given by the field values (sm-field), each branching to the primary "
"subfield (sm-subfield-1 = Forestry or Ecology) and then to the secondary subfield (sm-subfield-2); this places the Forestry group "
"and the Ecology group on visibly distinct branches. Complete rankings are in the companion workbook. Values were computed with "
"Python (pandas); figures with matplotlib; maps use Natural Earth boundaries [6]. All totals were independently re-verified.")

# ===== 4 RESULTS =====
doc.add_heading("4. Results",level=1)
doc.add_heading("4.1 Overall totals by metric and year",level=2)
gs=100*(yr(MF,'scientists',2024)/yr(MF,'scientists',2020)-1)
gp=100*(yr(MF,'publications',2024)/yr(MF,'publications',2020)-1)
gc=100*(yr(MF,'citations',2024)/yr(MF,'citations',2020)-1)
body(f"All three metrics rose over the five years (Figure 1). Listed forestry scientists grew {gs:.0f}% "
f"({yr(MF,'scientists',2020):,}{yr(MF,'scientists',2024):,}); summed career publications grew {gp:.0f}% "
f"({yr(MF,'publications',2020)/1000:,.0f})k{yr(MF,'publications',2024)/1000:,.0f})k; and citations grew {gc:.0f}% "
f"({yr(MF,'citations',2020)/1e6:,.1f})M{yr(MF,'citations',2024)/1e6:,.1f})M. Citations grew fastest, reflecting new entrants and the "
"continual accrual of citations to established authors.")
figure(f"{SCR}/ff1_change.png",Inches(6.5),"Figure 1. Scientists, publications and citations for the forestry research cohort, as "
"percent change from 2020, with the actual value labelled at each point (scientists; thousands of publications; millions of citations).")

doc.add_heading("4.2 Composition by group and taxonomy",level=2)
gF={r["key"]:r for r in MF["group"]}
body(f"The cohort is {100*nF/nU:.0f}% Forestry group ({nF:,}) and {100*nE/nU:.0f}% Ecology group ({nE:,}). The Ecology group—authors "
"whose primary specialty is ecology but who also work in forestry—is smaller but non-trivial and carries proportionally more "
"citations, as ecology tends to be more highly cited. Table 2 gives the two groups on all three metrics; Figure 2 is the taxonomy "
"tree, whose multiple roots are the Science-Metrix fields (sm-field): the Forestry group appears as gold "
"Forestry" subfield nodes "
"(mostly under the Agriculture, Fisheries & Forestry root, some under Biology and other fields), while the Ecology group appears as "
"the teal "Ecology" node under the Biology root branching to a secondary Forestry leaf. Figure 3 shows the two groups per year.")
merged_table(MF["group"],"Group","Table 2. Scientists, publications and citations by group (Forestry group vs Ecology group), per release year.",keyw=Inches(1.9))
figure(f"{SCR}/ff2_tree.png",Inches(6.55),"Figure 2. Multi-root taxonomy of the forestry research cohort: field / sm-field (roots) → "
"primary subfield / sm-subfield-1 (gold = Forestry group, teal = Ecology group) → secondary subfield / sm-subfield-2 (blue » = "
"quantitative). Right-hand numbers are author counts; minor branches aggregated (see note).")
figure(f"{SCR}/ff4_group.png",Inches(5.4),"Figure 3. Scientists, publications and citations by group (Forestry group, Ecology group), "
"2020–2024, in a single panel. For each group there are three blocks—S (scientists, green), P (publications, blue), C (citations, "
"orange)—each with five thin columns for 2020–2024 (lighter–darker). Bar height is that value as a percentage of the metric's "
"five-year maximum across the two groups (tallest column of each metric = 100%); the absolute 2024 value is printed above each "
"block's last column.")

doc.add_heading("4.3 Representation by continent and world region",level=2)
body(f"On every metric, North America and Europe supply the great majority of the cohort (North America {fcont['North America']['s'][-1]:,} "
f"scientists in 2024; Europe {fcont['Europe']['s'][-1]:,}), with Oceania a distant third and Asia, South America and Africa trailing "
"far behind despite holding most of the world's forests (Table 3, Figure 4).")
merged_table(MF["continent"],"Continent","Table 3. Scientists, publications and citations by continent, per release year.",keyw=Inches(1.5))
merged_table(MF["region"],"World sub-region","Table 4. Scientists, publications and citations by world sub-region, per release year.",keyw=Inches(1.9))
figure(f"{SCR}/ff3_continent.png",Inches(6.7),"Figure 4. Scientists, publications and citations by continent, 2020–2024, in a single panel. "
"For each continent there are three blocks—S (scientists, green), P (publications, blue), C (citations, orange)—and each block holds "
"five thin columns for the years 2020–2024 (lighter–darker). Bar height is that value as a percentage of the metric's five-year "
"maximum across all continents, so each metric's single tallest column equals 100%; this lets the three differently-scaled metrics "

```

"share one axis. The absolute 2024 value is printed above each block's last (2024) column.")

```
doc.add_heading("4.4 Representation by country (top 20 individual, remainder banded)",level=2)
body(f"The United States leads on every metric ({fc['United States']['s'][-1]:,} scientists in 2024), followed by Canada "
    f"({fc['Canada']['s'][-1]:,}); Finland and Sweden, with Germany, form the next tier and rank far above their population share. Table 5 "
    "lists the top 20 countries individually and bands the remainder; Figure 5 shows the top eight per year and Figure 6 the banded profile.")
body("Counts alone hide productivity and impact, and the three count metrics are so correlated that mapping them side by side is "
    "uninformative. Figure 7 therefore maps three intensity ratios instead, which reveal patterns the head-count cannot: publications "
    "per scientist and citations per scientist are highest for large-output producers such as Russia, China and Brazil, whereas "
    "citations per publication-impact per paper-is highest for the United States, Canada, Brazil and parts of Oceania, and comparatively "
    "low for Russia and China (high output, lower per-paper impact). Ratios are shown only for countries with at least five listed "
    "scientists over the five years, to avoid single-author noise.")
merged_table(banded_items(MF["country_banded"]), "Country", "Table 5. Scientists, publications and citations by country: top 20 "
    "individual, remainder in ~10% cumulative bands (ranked by scientists).",keyw=Inches(1.7))
figure(f"{SCR}/ff5_country.png",Inches(6.7),"Figure 5. Scientists, publications and citations by country (top 8), 2020-2024, in a single "
    "panel. For each country there are three blocks-S (scientists, green), P (publications, blue), C (citations, orange)-each with five "
    "thin columns for 2020-2024 (lighter-darker). Bar height is the percentage of that metric's five-year maximum across the countries "
    "(tallest column of each metric = 100%); the absolute 2024 value is printed above each block's last column.")
figure(f"{SCR}/ff6_country_band.png",Inches(6.4),"Figure 6. Scientists by country (average per year): top-20 individual bars, remainder in ~10% cumulative bands.")
figure(f"{SCR}/ff9_worldmaps.png",Inches(6.1),"Figure 7. Output- and impact-intensity by country (forestry cohort, 2020-2024 averages): "
    "publications per scientist, citations per scientist, and citations per publication. These ratios differentiate countries that the "
    "three strongly-correlated count maps could not. Countries with fewer than five listed scientists across the five years are left "
    "grey. Boundaries: Natural Earth [6].")
```

```
doc.add_heading("4.5 Subnational representation (United States, Canada, Australia)",level=2)
body("Within the United States, Oregon, California and a pooled block of federal agencies (USDA Forest Service, USGS, USDA-ARS) lead, "
    "reflecting the public-lands forestry research system; the Pacific North-West and Rocky Mountain states are prominent. In Canada, "
    "British Columbia and Ontario lead; in Australia, Queensland and Victoria. Table 6 gives the top U.S. states; Figure 8 maps the "
    "average. Full per-state, per-year tables for all three countries are in the workbook.")
merged_table(MF["us_state"][:12], "U.S. state / category", "Table 6. Scientists, publications and citations by U.S. state (top 12), per release year.")
figure(f"{SCR}/ff10_usmaps.png",Inches(6.7),"Figure 8. Scientists, publications and citations by U.S. state - five-year-average choropleths, three compact panels (contiguous states; AK, HI omitted from frame).")
```

```
doc.add_heading("4.6 Representation by institution (with country; top 20 individual, ranks 21-100 banded)",level=2)
body(f"The most-represented institution is {funi['key']} ({funi.get('country','')}), followed by other specialized agricultural and "
    "forestry universities, North-American land-grant universities and national forest-research agencies. Table 7 lists the top 20 "
    "institutions (with country) individually and bands ranks 21-100; Figure 9 shows each of the top seven institutions' own trajectory "
    "and Figure 10 the banded profile.")
merged_table(banded_items(MF["university_banded"]), "Institution (country)", "Table 7. Scientists, publications and citations by "
    "institution: top 20 individual, ranks 21-100 in ~10% cumulative bands (ranked by scientists).",foldcountry=True)
figure(f"{SCR}/ff7_university.png",Inches(6.7),"Figure 9. Scientists, publications and citations by institution (top 7), 2020-2024. For "
    "each institution there are three blocks-S (scientists, green), P (publications, blue), C (citations, orange)-each with five thin "
    "columns for 2020-2024 (lighter-darker). Here each bar is a percentage of THAT institution's own five-year maximum for that metric, "
    "so every institution peaks at 100% at its own best year (bars are not comparable in height across institutions, which differ greatly "
    "in size-see Table 7 for absolute magnitudes); the absolute 2024 value is printed above each block's last column.")
figure(f"{SCR}/ff8_university_band.png",Inches(6.2),"Figure 10. Scientists by institution (average per year): top-20 individual, ranks 21-100 in ~10% cumulative bands.")
```

```

# ===== 5 DISCUSSION =====
doc.add_heading("5. Discussion",level=1)
body("Reading the three metrics together sharpens rather than softens the picture: the countries and institutions
that field the most "
"highly cited forestry researchers also carry, by construction, most of the cohort's papers and citations, so
influence is at least "
"as concentrated as head-count. The feature specific to forestry is the weight of Canada and the Nordic
countries: Finland and "
"Sweden, tiny by population, rank among the leaders, reflecting deep national forestry-research traditions and
forest-based "
"economies. The gap between where forests are (largely tropical and Global South) and where the recognized
forestry expertise is "
"affiliated (temperate, wealthy, North-American and Nordic) is a quantitative entry point into debates on
research equity and "
"forestry capacity-building in the tropics.")
body("The two groups. Separating the cohort by group shows that the primary-forestry Forestry group supplies the
great majority of "
f"records ({100*nF/nU:.0f}%) and dominates every geographic ranking, while the Ecology group ({100*nE/nU:.0f}%)—
ecologists who also "
"publish in forestry—is smaller but citation-dense and skews the citation totals upward relative to head-count.
The taxonomy tree "
"makes the distinction concrete: most forestry work is rooted in the Agriculture, Fisheries & Forestry field,
but a substantial "
"amount sits under Biology (both as forestry-primary authors whose field-level label is Biology, and as the
entire Ecology group), "
"and smaller amounts under Enabling Technologies, Chemistry, Earth & Environmental Sciences and Economics. This
cross-field spread is "
"why a single-subfield filter understates forestry's true footprint, and why the broader Forestry+Ecology frame
in Appendix A—"
f"{tot(MC,'scientists'),} records—remains a useful complement for biodiversity- and conservation-oriented
readers.")
body("Uses. The multi-metric, multi-scale, two-group tables let funders and forestry ministries benchmark
national standing on "
"head-count, output and impact separately and track it yearly; forestry schools benchmark peers and target
recruitment; editors "
"widened reviewer and speaker pools using the geography; and science-of-science researchers study concentration
and emergence, and the "
"forestry-ecology boundary specifically.")
body("Limitations. Affiliation country is current institution, not nationality or origin. Scopus coverage biases
toward "
"English-language journals, likely understating tropical and non-English forestry. Publication and citation sums
inherit the skew of "
"citation distributions—a few superstar authors dominate—so group totals are sensitive to outliers. The counting
unit is the "
"author-year, so annual changes mix genuine entry with re-computation. The two-group selection is deliberately
conservative on the "
"ecology side (it requires Forestry as the explicit secondary subfield); a looser rule would enlarge the Ecology
group. Subnational "
"placement leaves some U.S. institutions "Unknown" and pools federal agencies.")

# ===== 6 CONCLUSION =====
doc.add_heading("6. Conclusion",level=1)
body(f"Measured by scientists, publications or citations, the forestry research cohort-{nF:,} primary-forestry
authors plus {nE:,} "
f"ecology-rooted authors who also work in forestry, {nU:,} in all—is small but growing on every metric and
strongly concentrated in "
"the Global North. North America and Europe dominate; the United States and Canada lead among countries, with
Finland and Sweden "
f"conspicuously over-represented; Oregon, California and a block of U.S. federal agencies lead subnationally;
and {funi['key']} is the "
"most represented institution. The two groups behave consistently in geography but differ in citation intensity,
and the multi-root "
"taxonomy shows forestry's reach across the Agriculture and Biology fields and beyond. Set against the broader
Forestry+Ecology "
"population in Appendix A, this cohort is the sharper, forestry-centred lens; the two are complementary readings
of the same data.")

# ===== 7 SUMMARY =====
doc.add_heading("7. Project Summary",level=1)
body(f"Five career-long releases of the Stanford / Ioannidis database (2020-2024) were filtered to a forestry
research cohort of {nU:,} "
f"records—Forestry group ({nF:,}: primary subfield Forestry) plus Ecology group ({nE:,}: primary Ecology,
secondary Forestry)—and "
"analysed by number of scientists, publications and citations across continent, region, country, subnational
unit and institution "
"(with country), separating the two groups and mapping them with a multi-root field-subfield-subfield taxonomy.
Rankings list the top "
"20 individually and band the rest by ~10% cumulative head-count; maps show the five-year average. The broader
Forestry+Ecology ")

```

"analysis is in Appendix A. Deliverables: this report; a comprehensive workbook with full per-year, per-metric rankings and per-group "

"sheets; and the combined career database (Excel and CSV).")

===== 8 REFERENCES =====

doc.add_heading("8. References and Data Sources",level=1)

refs=[

"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569-16572. <https://doi.org/10.1073/pnas.0507655102>",

"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite across scientific disciplines. PLoS Biology, 14(7), e1002501. <https://doi.org/10.1371/journal.pbio.1002501>",

"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author database annotated for scientific field. PLoS Biology, 17(8), e3000384. <https://doi.org/10.1371/journal.pbio.3000384>",

"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized citation indicators. PLoS Biology, 18(10), e3000918. <https://doi.org/10.1371/journal.pbio.3000918>",

"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open scientific journal ontology. Proc. 13th ISSI Conference, 66-77. (Science-Metrix classification.)",

"Natural Earth (2024). Free vector and raster map data, 1:10m Admin 0 - Countries. <https://www.naturalearthdata.com> (public domain).",

"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized citation indicators" (Version 8) [Data set]. Elsevier Data Repository. <https://doi.org/10.17632/btchxktzyw.8>",

] for i,rf in enumerate(refs,1):

par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);

par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)

a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)

b=par.add_run(rf); b.font.name="Arial"; b.font.size=Pt(9.5)

p("Data sources.",size=9.5,bold=True,before=4,after=2)

body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions 3/5/6/7/8, via the Elsevier "

"Digital Commons / Mendeley Data public API. Field/subfield taxonomy: Table_2 threshold tables in the same dataset. Base maps: Natural "

"Earth (world) and a public US-state boundary file. Analysis and figures: Python (pandas, openpyxl, matplotlib).",size=9.5)

===== APPENDIX A: COMBINED =====

doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

doc.add_heading("Appendix A. Combined Forestry + Ecology Analysis",level=1)

body(f"This appendix presents the broader Forestry+Ecology population for comparison: every author whose primary or secondary subfield "

f"is Forestry or Ecology ({tot(MC,'scientists'),} author-year records). Unlike the main report—two groups selected by the rules in "

"Section 2—this appendix uses the wider either-subfield definition and includes all of ecology (not only the ecologists who also work "

"in forestry). Method and metric definitions are otherwise identical. Section 5 compares the frames.")

cc={r["key"]:r for r in MC["country"]}; ccont={r["key"]:r for r in MC["continent"]}

cfl={r["key"]:r for r in MC["field_cls"]}; cuni=MC["university"][0]

doc.add_heading("A.1 Combined totals and field composition",level=2)

body(f"The combined cohort comprises {tot(MC,'scientists'),} author-year records; in 2024 its

{yr(MC,'scientists',2024):,} scientists "

f"carry {yr(MC,'publications',2024)/1000:,.0f}k publications and {yr(MC,'citations',2024)/1e6:,.1f}M citations. Ecology dominates: in "

f"2024, {cfl['Ecology']['s'][-1]:,} scientists were in Ecology, {cfl['Forestry']['s'][-1]:,} in Forestry and {cfl['Both']['s'][-1]:,} "

"in both, about five to one ecology-to-forestry.")

simple_table(["Metric \ Year"]+YS+["5-yr"],

[["Scientists (list entries)"+[f"{yr(MC,'scientists',y):,}" for y in YEARS]+[f"{tot(MC,'scientists')//5:,}"],

[["Publications (career papers)"+[f"{yr(MC,'publications',y):,}" for y in YEARS]+

[f"{tot(MC,'publications'):,}"],

[["Citations (career citations)"+[f"{yr(MC,'citations',y):,}" for y in YEARS]+[f"{tot(MC,'citations'):,}"]],

[Inches(2.1)]+[Inches(0.82)]*6,aligns=["left"]+["right"]*6,

cap="Table A1. Combined Forestry+Ecology totals by release year and metric. 'avg for scientists, total for publications/citations.")

merged_table(MC["field_cls"],"Field","Table A2. Scientists, publications and citations by field composition (Ecology / Forestry / both), per release year.",keyw=Inches(1.5))

figure(f"{SCR}/cfl_change.png",Inches(6.3),"Figure A1. Scientists, publications and citations for the combined cohort, as percent change from 2020 (actual values labelled).")

figure(f"{SCR}/cf2_field.png",Inches(6.4),"Figure A2. Field composition (Ecology / both / Forestry) per year, for each metric.")

doc.add_heading("A.2 Combined representation by continent, country and institution",level=2)

body(f"North America and Europe again dominate ({ccont['North America']['s'][-1]:,} and {ccont['Europe']['s'][-1]:,} scientists in 2024), "

f"the United States far ahead ({cc['United States']['s'][-1]:,} in 2024), then the United Kingdom, Canada and Australia. The "

f"most-represented institution is {cuni['key']} ({cuni.get('country','')}). Tables A3-A5 give the merged continent and banded "

"country/institution rankings; Figures A3-A8 give per-year trends, banded profiles and average maps.")

merged_table(MC["continent"],"Continent","Table A3. Scientists, publications and citations by continent (combined), per release year.",keyw=Inches(1.5))

```

merged_table(banded_items(MC["country_banded"]), "Country", "Table A4. Scientists, publications and citations by
country (combined): top 20 individual, remainder in ~10% bands.", keyw=Inches(1.7))
merged_table(banded_items(MC["university_banded"]), "Institution (country)", "Table A5. Scientists, publications
and citations by institution (combined): top 20 individual, ranks 21-100 in ~10%
bands.", keyw=Inches(2.15), foldcountry=True)
figure(f"{SCR}/cf3_continent.png", Inches(6.6), "Figure A3. Scientists, publications and citations by continent
(combined), per year (legend at right).")
figure(f"{SCR}/cf4_country.png", Inches(6.6), "Figure A4. Scientists, publications and citations by country
(combined, top 8), per year (legend at right).")
figure(f"{SCR}/cf5_country_band.png", Inches(6.2), "Figure A5. Scientists by country (combined, average/yr): top-20
individual, remainder banded.")
figure(f"{SCR}/cf7_worldmaps.png", Inches(6.0), "Figure A6. Scientists, publications and citations by country
(combined) – five-year-average world choropleths.")
figure(f"{SCR}/cf6_university_band.png", Inches(6.2), "Figure A7. Scientists by institution (combined, average/yr):
top-20 individual, ranks 21-100 banded.")
figure(f"{SCR}/cf8_usmaps.png", Inches(5.9), "Figure A8. Scientists, publications and citations by U.S. state
(combined) – five-year-average choropleths.")

doc.save(OUT)
print("saved:", OUT, "| paragraphs:", len(doc.paragraphs), "| tables:", len(doc.tables))

```

13. build_docx6.py (SHA-256 6fcabe54ef4ddda3...; 982 lines)

```

# -*- coding: utf-8 -*-
# 2-GROUP forestry research report. Group F = AM=Forestry (3,271); Group E = AM=Ecology & AO=Forestry (615).
Union 3,886.
# Multi-root AQ->AM->AO taxonomy. Analyses separate the two groups. No broad-ecology appendix (both criteria
require Forestry).
import json
from docx import Document
from docx.shared import Pt, Inches, RGBColor
from docx.enum.text import WD_ALIGN_PARAGRAPH, WD_BREAK
from docx.enum.table import WD_TABLE_ALIGNMENT
from docx.oxml.ns import qn
from docx.oxml import OxmlElement

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
OUT=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.docx"
M=json.load(open(f"{SCR}/merged_data.json", encoding="utf-8"))
MF=M["forestry"]; MFF=M["forestry_F"]; MFE=M["forestry_E"]
R=json.load(open(f"{SCR}/ratios.json", encoding="utf-8"))
REFS=json.load(open(f"{SCR}/refs.json", encoding="utf-8"))
A6=json.load(open(f"{SCR}/auth6.json", encoding="utf-8"))
# ===== DATA section: RAW COUNTS only (no ratios, no citation sums) =====
CNTHDR=["Scientists /yr", "Solo pubs", "Senior pubs", "All pubs"]
def _cntcells(m):
    return [f"{m['sci_yr']:, .0f}", f"{m['solo_p']:, .0f}", f"{m['sen_p']:, .0f}", f"{m['all_p']:, .0f}"]
CNTNOTE=("Scientists = five-year average of listed author-year records; Solo = single-authored, Senior = single-
or "
        "first-authored, All = every publication – each count summed over the five releases. These are raw
selected "
        "counts; every per-author and per-publication rate is a derivation, reported in Sections 3-4.")
def count_table(members, keyhdr, cap, top=None, foldcountry=False):
    ms=members[:top] if top else members
    rows=[]
    for m in ms:
        lab=m["key"]
        if foldcountry and m.get("country"): lab=f'{lab} ({m["country"]})'
        rows.append([lab]+_cntcells(m))
    simple_table([keyhdr]+CNTHDR, rows, aligns=["left"]+["right"]*4, cap=cap)

# ===== RESULTS section: the NINE derivations = (solo, senior, all) x (pubs/au, cites/au, cites/pub)
=====
NINEHDR=["Solo p/au", "Solo c/au", "Solo c/pub", "Sr. p/au", "Sr. c/au", "Sr. c/pub", "All p/au", "All c/au", "All
c/pub"]
def _n9(m):
    return [f"{m['solo_ppa']:, .0f}", f"{m['solo_cpa']:, .0f}", f"{m['solo_cpp']:, .0f}",
            f"{m['sen_ppa']:, .0f}", f"{m['sen_cpa']:, .0f}", f"{m['sen_cpp']:, .0f}",
            f"{m['all_ppa']:, .0f}", f"{m['all_cpa']:, .0f}", f"{m['all_cpp']:, .0f}"]
NINENOTE=("Each authorship class–Solo (single-authored), Senior (single- or first-authored) and All–carries three
derived "
        "measures: publications per author (p/au), citations per author (c/au) and citations per publication
(c/pub), "
        "each formed from five-year sums over the listed author-year records. No raw citation total is shown.")
BLKHDR=["Population and authorship class", "Pubs / author", "Cites / author", "Cites / pub"]
def nine_block_rows(m, name):
    return [[f"{name} – solo", f"{m['solo_ppa']:, .0f}", f"{m['solo_cpa']:, .0f}", f"{m['solo_cpp']:, .0f}"],

```

```

        [f"{name} - senior",f"{m['sen_ppa']:, .0f}", f"{m['sen_cpa']:, .0f}", f"{m['sen_cpp']:, .0f}"],
        [f"{name} - all", f"{m['all_ppa']:, .0f}", f"{m['all_cpa']:, .0f}", f"{m['all_cpp']:, .0f}"]]
AUTH=json.load(open(f"{SCR}/authorship.json",encoding="utf-8"))
YEARS=[2020,2021,2022,2023,2024]; YS=[str(y) for y in YEARS]
GREEN="2F5233"; LIGHT="EEF3EA"; BAND="DCE6F0"
doc=Document()
st=doc.styles["Normal"]; st.font.name="Arial"; st.font.size=Pt(11)
st.element.rPr.rFonts.set(qn("w: eastAsia"), "Arial")
for i,(sz,col) in enumerate([(16, GREEN), (13, GREEN), (11, "333333")], start=1):
    h=doc.styles[f"Heading {i}"]; h.font.name="Arial"; h.font.size=Pt(sz); h.font.bold=True;
h.font.color.rgb=RGBColor.from_string(col)
sec=doc.sections[0]; sec.page_width, sec.page_height=Inches(8.5), Inches(11)
sec.top_margin=sec.bottom_margin=Inches(0.9); sec.left_margin=sec.right_margin=Inches(0.9)
CW=Inches(6.7)

def p(t="", size=11, bold=False, italic=False, align=None, color=None, after=6, before=0):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold;
    r.italic=italic
    if color: r.font.color.rgb=RGBColor.from_string(color)
    if align: par.alignment=align
    par.paragraph_format.space_after=Pt(after); par.paragraph_format.space_before=Pt(before); return par
def body(t, size=10.5, after=8):
    par=doc.add_paragraph(); r=par.add_run(t); r.font.name="Arial"; r.font.size=Pt(size)
    par.alignment=WD_ALIGN_PARAGRAPH.JUSTIFY; par.paragraph_format.space_after=Pt(after);
par.paragraph_format.line_spacing=1.13; return par
def shade(cell, hexc):
    tcPr=cell._tc.get_or_add_tcPr(); sh=OxmlElement("w:shd"); sh.set(qn("w:val"), "clear");
sh.set(qn("w:fill"), hexc); tcPr.append(sh)
def topborder(cell):
    tcPr=cell._tc.get_or_add_tcPr(); b=OxmlElement("w:tcBorders"); t=OxmlElement("w:top")
    t.set(qn("w:val"), "single"); t.set(qn("w:sz"), "6"); t.set(qn("w:color"), "9AA0A6"); b.append(t);
tcPr.append(b)
def setcell(cell, text, bold=False, white=False, size=8, align="left"):
    cell.text=""; par=cell.paragraphs[0]
    par.alignment={"left":WD_ALIGN_PARAGRAPH.LEFT, "center":WD_ALIGN_PARAGRAPH.CENTER, "right":WD_ALIGN_PARAGRAPH.R
IGHT}[align]
    r=par.add_run(str(text)); r.font.name="Arial"; r.font.size=Pt(size); r.bold=bold
    if white: r.font.color.rgb=RGBColor.from_string("FFFFFF")
    par.paragraph_format.space_after=Pt(0); par.paragraph_format.space_before=Pt(0)
def repeat_header(t):
    # mark the first row as a table header so it repeats when the table splits across pages
    trPr=t._tr.get_or_add_trPr(); h=OxmlElement("w:tblHeader"); h.set(qn("w:val"), "true"); trPr.append(h)
def tight_margins(t, m=18):
    tblPr=t._tbl.tblPr; mar=OxmlElement("w:tblCellMar")
    for side in ("left", "right", "top", "bottom"):
        e=OxmlElement(f"w:{side}"); e.set(qn("w:w"), str(m)); e.set(qn("w:type"), "dxa"); mar.append(e)
    tblPr.append(mar)
def fixed_layout(t, widths_in):
    # make explicit column widths authoritative (Word/LibreOffice otherwise auto-fit to content).
    # tblPr child order matters: tblW before tblLayout before tblCellMar (added later by tight_margins).
    tblPr=t._tbl.tblPr
    W=OxmlElement("w:tblW"); W.set(qn("w:w"), str(int(sum(widths_in)*1440))); W.set(qn("w:type"), "dxa");
tblPr.append(W)
    lay=OxmlElement("w:tblLayout"); lay.set(qn("w:type"), "fixed"); tblPr.append(lay)
    grid=t._tbl.find(qn("w:tblGrid"))
    if grid is not None:
        for gc, w in zip(grid.findall(qn("w:gridCol")), widths_in):
            gc.set(qn("w:w"), str(int(w*1440)))
# captions rendered at body size (10.5pt) per user request; italic + grey keep them distinct from body
CAPSIZE=10.5
def caption(t): p(t, size=CAPSIZE, italic=True, color="555555", after=3)
def figure(path, width, cap):
    par=doc.add_paragraph(); par.alignment=WD_ALIGN_PARAGRAPH.CENTER; par.add_run().add_picture(path, width=width)
    par.paragraph_format.space_before=Pt(6); par.paragraph_format.space_after=Pt(2)
    par.paragraph_format.keep_with_next=True # keep image with its caption across page breaks
    p(cap, size=CAPSIZE, italic=True, color="555555", align=WD_ALIGN_PARAGRAPH.CENTER, after=10)
CW_IN=6.7
def _strw(s, pt): # estimate rendered width of a string (inches) at point size pt, Arial-ish metrics
    em=pt/72.0; w=0.0
    for ch in str(s):
        if ch.isdigit(): w+=0.556
        elif ch in ",.": w+=0.25
        elif ch in "()[]": w+=0.333
        elif ch in " /-+": w+=0.30
        elif ch=="&": w+=0.667
        elif ch in "%": w+=0.90
        elif ch.isupper(): w+=0.70
        else: w+=0.50
    return w*em
def _cw(pairs, pad, minw): # pairs: list of (string, pt); width = widest + padding, floored at minw

```

```

return max(minw, max((_strw(s,pt) for s,pt in pairs), default=0.0)+pad)

def simple_table(headers,rows,widths=None,aligns=None,cap=None,hsize=9):
    if cap: caption(cap)
    t=doc.add_table(rows=1,cols=len(headers)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
    aligns=aligns or (["left"]+["right"]*(len(headers)-1))
    # content-aware widths: each column = widest cell/header; if total > page, scale to fit
    colw=[]
    for j in range(len(headers)):
        cells=[(headers[j],hsize)]+[(rows[i][j],9.5) for i in range(len(rows))]
        colw.append(_cw(cells,0.12,0.40))
    tot=sum(colw)
    if tot>CW_IN: colw=[w*CW_IN/tot for w in colw]
    fixed_layout(t,colw); tight_margins(t)
    for j,h in enumerate(headers):
        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=hsize,align="center");
shade(t.rows[0].cells[j],GREEN)
    repeat_header(t)
    for i,row in enumerate(rows):
        cells=t.add_row().cells
        for j,val in enumerate(row):
            setcell(cells[j],val,size=9.5,align=aligns[j])
            if i%2==1: shade(cells[j],LIGHT)
    for col,w in zip(t.columns,colw):
        for c in col.cells: c.width=Inches(w)
    doc.add_paragraph().paragraph_format.space_after=Pt(2)

def merged_table(items, keyhdr, cap, keyw=None, foldcountry=False,
                 mlabels=("Scientists","Publications ('000)","Citations (M)"),
                 cellfmt=(lambda v:f"{int(v):,}", lambda v:f"{v/1000:.1f}", lambda v:f"{v/1e6:.2f}"),
                 sumfmt=(lambda s:f"{sum(s)/5:,.0f}", lambda s:f"{sum(s)/1000:.1f}", lambda
s:f"{sum(s)/1e6:,.2f}")):
    caption(cap)
    heads=[keyhdr,"Metric"]+YS+["5-yr¹"]
    rowdata=[]
    for it in items:
        key=it["key"]
        if foldcountry and it.get("country"): key=f'{{key}} ({{it["country"]}})'
        s,pub,c=it["s"],it["p"],it["c"]
        triplet=[(mlabels[0],[cellfmt[0](v) for v in s], sumfmt[0](s)),
                 (mlabels[1],[cellfmt[1](v) for v in pub],sumfmt[1](pub)),
                 (mlabels[2],[cellfmt[2](v) for v in c], sumfmt[2](c))]
        rowdata.append((key,it.get("band",False),triplet))
    # content-aware widths: number cols sized to their widest value, remainder -> wide text col 1
    yd=[[] for _ in range(5)]; sd=[]; md=[]
    for _,_,tri in rowdata:
        for mlab,vals,summ in tri:
            md.append(mlab); sd.append(summ)
            for j in range(5): yd[j].append(vals[j])
    yw=[_cw([(YS[j],9)]+(v,10) for v in yd[j]),0.11,0.34) for j in range(5)]
    sw=_cw([("5-yr¹",9)]+(v,10) for v in sd],0.11,0.40)
    mw=_cw([("Metric",9)]+(m,8.5) for m in md],0.16,0.85)
    kw=max(1.35, CW_IN-mw-sum(yw)-sw)
    widths=[Inches(kw),Inches(mw)]+[Inches(x) for x in yw]+[Inches(sw)]
    t=doc.add_table(rows=1,cols=len(heads)); t.alignment=WD_TABLE_ALIGNMENT.CENTER
    t.autofit=False; t.allow_autofit=False; t.style="Table Grid"
    fixed_layout(t,[kw,mw]+yw+[sw]); tight_margins(t)
    for j,h in enumerate(headers):
        setcell(t.rows[0].cells[j],h,bold=True,white=True,size=9,align="center"); shade(t.rows[0].cells[j],GREEN)
    repeat_header(t)
    for mi,(key,isband,tri) in enumerate(rowdata):
        for ri,(mlab,vals,summ) in enumerate(tri):
            cells=t.add_row().cells
            setcell(cells[0],key if ri==0 else "",bold=(ri==0),size=10,align="left")
            setcell(cells[1],mlab,size=8.5,align="left")
            for j,v in enumerate(vals): setcell(cells[2+j],v,size=10,align="right")
            setcell(cells[7],summ,bold=True,size=10,align="right")
            for j in range(8):
                if isband: shade(cells[j],BAND)
                elif mi%2==1: shade(cells[j],LIGHT)
                if ri==0: topborder(cells[j])
    for col,w in zip(t.columns,widths):
        for cc in col.cells: cc.width=w
    p("¹ Summary column: 5-year average for Scientists (largely the same people recur each release); 5-year total
for Publications and Citations.",
      size=7.5,italic=True,color="777777",after=8)
def banded_items(bd): return bd["head"]+bd["bands"]
_MK={"scientists":"s","publications":"p","citations":"c"}

```

```

_MFMT={"scientists":lambda v:f"{int(round(v)):.0f}", "publications":lambda v:f"{v/1000:,.1f}", "citations":lambda
v:f"{v/1e6:,.2f}"}
_MCAP={"scientists":"Scientists", "publications":"Publications (thousands)", "citations":"Citations (millions)"}
def split_tables(members, keyhdr, tabnum, subject, top=25, foldcountry=False):
    # one table per metric (scientists / publications / citations); top-N ranked by that metric; last col = 5-yr
    average
    for mi, metric in enumerate(["scientists", "publications", "citations"]):
        k= _MK[metric]; fmt= _MFMT[metric]
        ranked=sorted(members, key=lambda it:-sum(it[k]))[:top]
        rows=[]
        for it in ranked:
            lab=it["key"]
            if foldcountry and it.get("country"): lab=f'{lab} ({it["country"]})'
            rows.append([lab]+[fmt(it[k][i]) for i in range(5)]+[fmt(sum(it[k])/5)])
        cap=(f"Table {tabnum}{'abc'[mi]}. { _MCAP[metric]} by {subject}, top {top} (ranked by the five-year
total). "
            "The last column is the five-year average.")
        simple_table([keyhdr]+YS+["Avg/yr"], rows, aligns=["left"]+["right"]*6, cap=cap)

# ---- intensity-ratio table: articles/author, citations/author, citations/article (quality vs quantity) ----
def _rfmt(m):
    return [f"{m['sci_yr']:.0f}", f"{m['art_per_auth']:.0f}", f"{m['cit_per_auth']:.0f}",
f"{m['cit_per_art']:.1f}"]
def ratio_table(members, keyhdr, cap, top=20, foldcountry=False, min_sci_yr=0.0):
    # members are pre-sorted by number of scientists (scale); show the largest `top` and their three ratios
    ms=[m for m in members if m["sci_yr"]>=min_sci_yr][:top]
    rows=[]
    for m in ms:
        lab=m["key"]
        if foldcountry and m.get("country"): lab=f'{lab} ({m["country"]})'
        rows.append([lab]+ _rfmt(m))
    simple_table([keyhdr, "Scientists / yr", "Articles / author", "Citations / author", "Citations / article"],
        rows, aligns=["left"]+["right"]*4, cap=cap)
def footer():
    par=sec.footer.paragraphs[0]; par.alignment=WD_ALIGN_PARAGRAPH.CENTER
    def fld(instr):
        b=OxmlElement("w:fldChar"); b.set(qn("w:fldCharType"), "begin")
        i=OxmlElement("w:instrText"); i.set(qn("xml:space"), "preserve"); i.text=instr
        e=OxmlElement("w:fldChar"); e.set(qn("w:fldCharType"), "end")
        for x in (b, i, e):
            r=par.add_run(); r._r.append(x); r.font.name="Arial"; r.font.size=Pt(8)
        r=par.add_run("Page "); r.font.name="Arial"; r.font.size=Pt(8); fld(" PAGE ")
        r=par.add_run(" of "); r.font.name="Arial"; r.font.size=Pt(8); fld(" NUMPAGES ")
    footer()

def tot(d, m): return sum(d["totals"][m][y] for y in YS)
def yr(d, m, y): return d["totals"][m][str(y)]
fc={r["key"]:r for r in MF["country"]}; fcont={r["key"]:r for r in MF["continent"]}; funi=MF["university"][0]
nF=MFF["n"]; nE=MFE["n"]; nU=MF["n"]

# ---- single-author appendices (B = Forestry group, C = Ecology group) ----
BSA=json.load(open(f"{SCR}/forestry_B_data.json", encoding="utf-8"))
CSA=json.load(open(f"{SCR}/forestry_C_data.json", encoding="utf-8"))
def recs_to_members(D, dim, ranked_keys=None):
    # convert refine forestry recs by metric ({metric:[recs]}) -> merged_table items ({key, country, s, p, c})
    idx={m:{r["key"]:r for r in D[dim][m]} for m in ["scientists", "publications", "citations"]}
    keys=ranked_keys if ranked_keys is not None else [r["key"] for r in D[dim]["scientists"]]
    out=[]
    for k in keys:
        rs=idx["scientists"].get(k); rp=idx["publications"].get(k); rc=idx["citations"].get(k)
        out.append({"key":k, "country":(rs.get("country") if rs else None),
            "s":[rs[str(y)] for y in YEARS] if rs else [0]*5,
            "p":[rp[str(y)] for y in YEARS] if rp else [0]*5,
            "c":[rc[str(y)] for y in YEARS] if rc else [0]*5})
    return out
def sa_topkeys(D, dim, metric="scientists"): return [r["key"] for r in D[dim+"_banded"][metric]["head"]]
def sa_at(D, dim, key, metric="scientists"):
    for r in D[dim][metric]:
        if r["key"]==key: return r["2024"]
    return 0
# single-author metric labels + cell/summary formatters for merged_table
SA_ML=("Scientists", "Single-author papers", "Cites to SA papers ('000)")
SA_CF=(lambda v:f"{int(v):.0f}", lambda v:f"{int(v):.0f}", lambda v:f"{v/1000:,.1f}")
SA_SF=(lambda s:f"{sum(s)/5:,.0f}", lambda s:f"{sum(s):.0f}", lambda s:f"{sum(s)/1000:,.1f}")

# ===== TITLE =====
for _ in range(2): doc.add_paragraph()
p("Global Representation of the World's Most-Cited Forestry Scientists by Number of Scientists, Publications and Citations",

```

size=19,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,color=GREEN,after=6)

p("A Five-Year, Multi-Metric Geographic and Institutional Analysis (2020–2024) of the Stanford / Ioannidis Career-Long Citation Database",

size=12.5,italic=True,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=22)

p("Analytical Report",size=12,bold=True,align=WD_ALIGN_PARAGRAPH.CENTER,after=2)

p("Prepared 6 July 2026",size=11,align=WD_ALIGN_PARAGRAPH.CENTER,color="555555",after=2)

p("Data source: Ioannidis, J. P. A. (2025), Elsevier Data Repository, doi:10.17632/btchxktzyw (v3–v8)",

size=9,align=WD_ALIGN_PARAGRAPH.CENTER,color="777777",after=20)

p("Abstract",size=13,bold=True,color=GREEN,after=4)

body(f"This report quantifies the geographic and institutional representation of the world’s most highly cited forestry researchers "

f"for each annual release of the Stanford / Ioannidis “top-scientists” database from data years 2020 to 2024, resolving every "

f"population’s output by authorship position. The cohort–termed “Forestry plus Ecology”–is assembled from two mutually exclusive "

f"groups, both of which require Forestry to be present: “Forestry” (primary Science-Metrix subfield = Forestry; {nF:,} records) and "

f"“Ecology” (primary subfield = Ecology with secondary subfield = Forestry; {nE:,} records), together {nU:,} author-year records "

f"analysed as two separate groups throughout. In 2024 these {yr(MF,'scientists',2024):,} scientists carry "

f"{yr(MF,'publications',2024)/1000:,.0f} thousand career publications and {yr(MF,'citations',2024)/1e6:,.1f} million career citations. "

"The Data section reports only raw counts–for every division, the number of scientists and of solo (single-authored), senior (single-

"or first-authored) and all publications; the Results derive, for each of those three authorship classes, three rates–publications per "

"author, citations per author and citations per publication–a 3 × 3 scheme of nine measures per entity, shown as 3 × 3 charts. No raw "

"citation total is reported. The central findings are that the United States leads the field by "

"scale and is also among the most cited per paper of the large producers, whereas China and Japan show the opposite high-volume, "

"low-per-paper-impact profile; that per-paper impact tracks discipline and geography rather than how much a population publishes, and "

"holds its solo-senior-all ordering across regions; that the ecology-rooted group is about twice as cited per paper as the "

"forestry-rooted group at every authorship level; and that ranking scientists by undiscriminated citations measures cited "

"collaboration more than individual authorship–narrowing to solo- or first-authored credit reorders them substantially. A multi-root "

"taxonomy tree organizes the cohort by field → primary subfield → secondary subfield, and Appendix A profiles the Ecology group "

"(criterion 2) on its own.",size=10)

kw=doc.add_paragraph(); r0=kw.add_run("Keywords: "); r0.bold=True; r0.font.name="Arial"; r0.font.size=Pt(10)

r1=kw.add_run("forestry; ecology; bibliometrics; scientometrics; citation indicators; authorship position; solo and senior authorship; "

"citations per paper; Stanford/Ioannidis list; research geography; Science-Metrix taxonomy; choropleth mapping."); r1.font.name="Arial"; r1.font.size=Pt(10); r1.italic=True

doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)

===== 1 BACKGROUND =====

doc.add_heading("1. Background, Concept and Related Work",level=1)

body("Citation-based rankings of individual scientists are widely used in research evaluation. The most common single statistic, the "

"h-index [1], conflates productivity with impact and scales with career length, field and co-authorship. To standardize comparison, "

"Ioannidis and colleagues devised a composite indicator (the “c-score”) combining six field-normalized metrics [2,3] and published "

"an openly licensed database ranking about the top 100,000 scientists (and the top 2% of each subfield) across 22 fields and 174 "

"subfields defined by the Science-Metrix journal ontology [3,4]. Each ranked author carries a primary subfield (sm-subfield-1), an "

"optional secondary subfield (sm-subfield-2), and a field-level label (sm-field). The database appears in career-long and single-year "

"forms and is refreshed annually from Scopus. This report analyses the career-long tables.")

body("Whereas the original database is typically used to look up individuals, this report treats forestry as a population and asks how "

"that population is distributed, not only by head-count but also by the publications and citations those researchers carry. Forestry "

"is a small, applied, globally distributed field at the centre of timber economies, biodiversity, carbon and land-use policy. Because "

"forestry and ecology overlap heavily in journals, methods and personnel, a subset of the most-cited forestry researchers are "

"classified primarily as ecologists but secondarily as foresters; the report keeps these as a distinct “Ecology” group alongside the "

"primary-forestry “Forestry” group, and separates the two throughout. Both groups, by construction, contain only authors who work in "

"forestry–there is no ecology-without-forestry group here.")

===== 2 DATA =====

doc.add_heading("2. Data Description",level=1)

body("The primary data are the five career-long author tables ("Table_1_Authors_career") from dataset doi:10.17632/btchxktzyw, "
"versions 3, 5, 6, 7 and 8, for data years 2020-2024. Each ~75-85 MB spreadsheet holds one row per ranked scientist (about "
"186,000-230,000 rows) with the author name, one large-institution affiliation, its ISO country code, the six component citation "
"metrics and composite c-score, the career publication count (np) and career citation count (nc), authorship-resolved counts-single-"
"authored papers and their citations (nps/ncs), single-or-first-authored (cpsf/ncsf) and single-first-or-last-authored (npsfl/ncsfl)-and "
"the Science-Metrix classification: primary subfield (sm-subfield-1, column AM), secondary subfield (sm-subfield-2, AQ) and field "
"(sm-field, AQ). All "
"counts are on the Scopus basis and inherit its journal-centric coverage; Section 5 (Table 10) validates that coverage externally "
"against Google Scholar for two cohort members and bounds the effect on the report's conclusions.")
body(f"Selection. A record enters the cohort under either of two rules, kept as separate groups: "
f"(1) "Forestry" - sm-subfield-1 = Forestry ({nF:,} records); (2) "Ecology" - sm-subfield-1 = Ecology AND sm-subfield-2 = "
f"Forestry ({nE:,} records). Both rules require Forestry, so the two groups are mutually exclusive and neither admits an author who is "
f"in ecology but not forestry; their sum, {nU:,} author-year records, is what we call "Forestry plus Ecology." We verified against the "
"raw releases that sm-field (AQ) is a field-level label taking only the 20 Science-Metrix field names and is never literally "
""Forestry" (the forestry field value is the broader "Agriculture, Fisheries & Forestry"); AQ therefore adds no records beyond the two "
"subfield rules but is used as the set of roots for the taxonomy tree (Figure 3, at the end of this section). The cohort is delivered with this report as an "
"Excel workbook (full per-year, per-metric rankings and per-group sheets) and CSV.")
body("This section presents only the raw selected counts: for every division, the number of listed scientists and the number of solo "
"(single-authored), senior (single- or first-authored) and all publications, summed over the five releases (Tables 1-7). These are the "
"data as selected; they involve no derivation beyond counting and classifying by authorship position. Every per-author and per-"
"publication rate-how much each population publishes per scientist, and how heavily its papers are cited-is a derivation and is reported "
"in the Results (Section 4), with the underlying definitions in the Methods (Section 3). Figure 1 shows the cohort's per-year growth; "
"full lists and the Canadian and Australian subdivisions are in the companion workbook.")
cby=A6["cohort_by"]
simple_table(["Release year"]+CNTHDR,
[[str(y),f"{cby['n']}[i]:,.0f}",f"{cby['solo_p']}[i]:,.0f}",f"{cby['sen_p']}[i]:,.0f}",f"{cby['all_p']}[i]:,.0f}"]
for i,y in enumerate(YEARS)],
aligns=["left"]+["right"]*4,
cap=f"Table 1. "Forestry plus Ecology" cohort (Forestry {nF:,} + Ecology {nE:,} = {nU:,} author-year records) by release year: scientist "
"and publication counts. Here "Scientists" is the number of listed author-year records in each release (not a five-year average); "
"Solo/Senior/All are the publication counts. "+CNTNOTE)
figure(f"{SCR}/ffl_change.png",Inches(6.3),"Figure 1. Growth of the cohort's solo, senior and all publication counts since 2020 (percent "
"change from the 2020 release, with the actual paper count labelled at each point). A per-year, descriptive view of the selected data.")
body("Composition by group. The cohort is "
f"{100*nF/nU:.0f}% Forestry group ({nF:,} records) and {100*nE/nU:.0f}% Ecology group ({nE:,}). Table 2 gives the scientist and "
"publication counts of the two groups; the field composition is shown by the taxonomy tree (Figure 3) at the end of this section. How "
"the two groups differ in per-author output and per-paper impact is a derivation and is taken up in Section 4.1.")
count_table(A6["div"]["group"],"Group","Table 2. Scientist and publication counts by group - Forestry (criterion 1) vs Ecology "
"(criterion 2), five-year values. "+CNTNOTE)
body("Geographic tables. Tables 3-5 give continent, world sub-region and country counts. North America and Europe supply most of the "
"cohort's output; the United States and Canada lead among countries, with Finland and Sweden strongly over-represented, and Central "
"Europe (Germany, Austria, Switzerland, Poland, Czechia, Slovakia, Hungary, Slovenia) broken out as its own sub-region, ranking third.")
count_table(A6["div"]["continent"],"Continent","Table 3. Scientist and publication counts by continent (five-year values).")
count_table(A6["div"]["region"],"World sub-region","Table 4. Scientist and publication counts by world sub-region (top 8 by output). "
"Central Europe is broken out of the UN Western/Eastern/Southern Europe groupings.",top=8)
count_table(A6["div"]["country"],"Country","Table 5. Scientist and publication counts by country: top 25 by output (five-year values).",top=25)
body("Subnational and institutional tables. Table 6 gives the leading U.S. states and Table 7 the most-represented institutions (with "
f"country). The most-represented single institution is {A6['div']['university'][0]['key']} "

```

f"({A6['div'] ['university'] [0].get('country', '')}). Full per-state tables and the complete institution list are
in the workbook.")
count_table(A6["div"] ["us_state"], "U.S. state / category", "Table 6. Scientist and publication counts by U.S.
state (top 15 by output).", top=15)
figure(f"{SCR}/ff10_usmaps.png", Inches(6.7), "Figure 2. Solo, senior and all publication counts by U.S. state –
five-year choropleths "
" (contiguous states; Alaska and Hawaii omitted from the frame). Boundaries: a public U.S. state file.")
count_table(A6["div"] ["university"], "Institution (country)", "Table 7. Scientist and publication counts by
institution: top 25 by output.", top=25, foldcountry=True)
body("Field composition (taxonomy). The two groups occupy visibly distinct branches of the Science-Metrix
taxonomy (Figure 3). Its "
"multiple roots are the fields (sm-field): the Forestry group appears as gold "Forestry" subfield nodes (mostly
under the Agriculture, "
"Fisheries & Forestry root, some under Biology and other fields), while the Ecology group appears as the teal
"Ecology" node under the "
"Biology root branching to a secondary Forestry leaf—showing that even a strictly forestry-defined population
reaches across many fields.")
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
figure(f"{SCR}/ff2_tree_1.png", Inches(6.9), "Figure 3 (part 1 of 2). Multi-root taxonomy of the forestry research
cohort – the two "
"principal roots (Agriculture, Fisheries & Forestry; Biology): field / sm-field (roots) → primary subfield / sm-
subfield-1 (gold = "
"Forestry group, teal = Ecology group) → secondary subfield / sm-subfield-2 (blue » = quantitative). Right-hand
numbers are author "
"counts; minor branches aggregated. Part 2 (remaining roots) follows.")
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
figure(f"{SCR}/ff2_tree_2.png", Inches(6.9), "Figure 3 (part 2 of 2). Remaining cross-field roots of the taxonomy
(Enabling & Strategic "
"Technologies, Chemistry, Earth & Environmental Sciences, Economics & Business): field / sm-field (roots) →
primary subfield "
"/ sm-subfield-1 → secondary subfield / sm-subfield-2. Same colour key as part 1; roots with fewer than 15
authors omitted (see note).")

# ===== 3 METHODS =====
doc.add_heading("3. Methods", level=1)
body("Population and groups. The cohort is the union of the two selection rules above; every table and figure
distinguishes the "
"Forestry group (primary subfield Forestry) from the Ecology group (primary Ecology, secondary Forestry). The
overall geographic and "
"institutional analyses pool both groups; the group split is in Table 2 and Section 4.1.")
body("Raw inputs (the Data section). Three quantities are read directly from the source and simply counted, per
division and per release "
"year: the number of listed scientists (author-year records) and, resolving each author's output by position,
the number of solo "
"(single-authored, dataset fields nps/ncs), senior (single- or first-authored, cpsf/ncsf—the papers an author
most directly drives) and "
"all (np/nc) publications. Section 2 tabulates these counts and nothing else; classifying a paper by authorship
position is a lookup, "
"not a derivation.")
body("Derived measures (the nine numbers, computed in Results). Every rate is formed in pre-processing from the
counts above and reported "
"in Section 4. For each of the three authorship classes—solo, senior, all—three measures are computed over the
five-year author-year "
"sums: publications per author ( $\Sigma$ class-publications ÷  $\Sigma$ scientists, the output a listed scientist carries),
citations per author "
"( $\Sigma$ class-citations ÷  $\Sigma$ scientists, total impact per listed scientist) and citations per publication ( $\Sigma$ class-
citations ÷  $\Sigma$ class-
"publications, per-paper impact). That is a 3 × 3 scheme—(solo, senior, all) × (per author, per author, per
paper)—giving nine derived "
"measures per entity; the report presents them as 3 × 3 charts (Section 4). No raw citation total is displayed.
Because career values "
"are cumulative and each release re-lists largely the same people, sums are over author-years and the scientist
count used as a "
"denominator is the five-year total of listed records.")
body("Authorship position, resolved further (Section 4.6). Beyond the nested solo/senior/all classes, each
author's output splits into "
"three mutually exclusive positions—single-authored, first-authored (of a multi-author paper) and other co-
authored (every paper on "
"which the author is neither sole nor first author). Last-authorship is not treated separately: this study asks
about the work a "
"scientist individually or leadingly produces (solo and senior), so all non-lead co-authorship is pooled as
"other.")
body("Geography. Continent, UN M49 world sub-region and country were derived from the ISO affiliation code. For
the United States, "
"Canada and Australia, each institution was resolved to a first-level subdivision by geolocating its main campus
(a verified "
"language-model pass); distributed federal agencies are pooled as "Federal/Multi-site" and unplaceable ones as
"Unknown." A few "
"source records carry no affiliation country code or no institution name; they are included in all cohort totals
but cannot appear "

```

```

"in the geographic or institutional breakdowns, so those tables can sum marginally below the totals.")
body("Authorship position. The database also reports, for each scientist, the number of and citations to their
single-authored papers "
      "(nps/ncs), their single-or-first-authored papers (cpsf/ncsf) and their single-first-or-last-authored papers
(npsfl/ncsfl). Because "
      "these sets are nested, Section 4.6 differences them into four mutually exclusive positions—single, first, last
and middle—and uses "
      "them to test how a scientist's rank depends on which authorship positions are credited. All position counts use
the all-citations "
      "basis, consistent with np and nc.")
body("Taxonomy and tools. The taxonomy tree (Figure 3) has multiple roots given by the field values (sm-field),
each branching to the "
      "primary subfield (sm-subfield-1 = Forestry or Ecology) and then to the secondary subfield (sm-subfield-2); this
places the two groups "
      "on visibly distinct branches. Complete rankings are in the companion workbook. Values were computed with Python
(pandas); figures "
      "with matplotlib; maps use Natural Earth boundaries [6]. All totals and ratios were independently re-verified.")

# ===== 4 RESULTS =====
def a6(dim,key):
    for m in A6["div"][dim]:
        if m["key"]==key: return m
    return {"solo_p":0,"solo_cpp":0,"sen_p":0,"sen_cpp":0,"all_p":0,"all_cpp":0,"sci_yr":0}
A6C=A6["cohort"]; A6GF=A6["group"]["Forestry"]; A6GE=A6["group"]["Ecology"]

doc.add_heading("4. Results – the nine derived measures of output and impact",level=1)
body("The Data section gave the raw counts. This section derives, for every population, how much its listed
scientists publish and how "
      "heavily that work is cited, resolved by authorship position. Each entity is read on nine measures—the three
authorship classes (solo, "
      "senior, all) crossed with three rates (publications per author, citations per author, citations per
publication)—presented as 3 × 3 "
      "charts. Two regularities run through all of them and are worth stating first. (i) Per-paper impact peaks at the
lead, not the sole, "
      "f"position: across the cohort solo papers draw about {A6C['solo_cpp']:.0f} citations each and all papers
{A6C['all_cpp']:.0f}, but the "
      "f"senior (single- or first-authored) set sits highest at {A6C['sen_cpp']:.0f}, because first-authored papers—an
author's own empirical "
      "and methodological work—are their most-cited. (ii) Output volume and per-paper impact are close to independent,
so a population's "
      "standing depends entirely on which of the nine numbers is asked.")

doc.add_heading("4.1 The cohort and the two groups",level=2)
body(f"Table 8 gives the cohort and the two groups on all nine measures, and Figure 4 shows the group contrast
cell by cell. The "
      "f"ecology-rooted group leads the forestry-rooted group on every one of the nine: it publishes more per author
(all {A6GE['all_ppa']:.0f} "
      "f"vs {A6GF['all_ppa']:.0f} papers), draws far more citations per author (all {A6GE['all_cpa']/1000:.1f}k vs
{A6GF['all_cpa']/1000:.1f}k) "
      "f"and is about twice as cited per paper at every class (solo {A6GE['solo_cpp']:.0f} vs {A6GF['solo_cpp']:.0f},
all {A6GE['all_cpp']:.0f} "
      "f"vs {A6GF['all_cpp']:.0f}). Within each population the citations-per-publication row rises from solo to senior
then dips for all—the "
      "lead-position peak noted above—while publications and citations per author climb monotonically as the class
widens, since wider classes "
      "contain more papers.")
simple_table(BLKHDR,
    nine_block_rows(A6C,"Whole cohort")+nine_block_rows(A6GF,"Forestry group")+nine_block_rows(A6GE,"Ecology
group"),
    aligns=["left"]+["right"]*3,
    cap="Table 8. The cohort and the two groups on the nine derived measures (five-year values), each population
resolved into solo, senior "
    "and all authorship classes. "+NINENOTE)
figure(f"{SCR}/ff_ratio_group.png",Inches(6.4),"Figure 4. Forestry vs Ecology group – the 3 × 3 scheme: rows are
the authorship classes "
    "(solo, senior, all), columns the three rates (publications per author, citations per author, citations per
publication), five-year "
    "values. The Ecology group is higher in every one of the nine cells.")

doc.add_heading("4.2 By continent and world region",level=2)
_na=a6('continent','North America'); _as=a6('continent','Asia'); _sa=a6('continent','South America')
body(f"Across continents (Figure 5) the three rates tell different stories. On citations per publication North
America leads at every "
      "f"class (all {_na['all_cpp']:.0f}) and Asia trails (all {_as['all_cpp']:.0f}) despite carrying the most papers
per author (all "
      "f({_as['all_ppa']:.0f}), against North America's {_na['all_ppa']:.0f})—the clearest volume-versus-impact contrast
in the data. A forensic "
      "f"oddity: South America and Africa post the highest solo publications per author ({_sa['solo_ppa']:.0f} and "

```

```
f"{a6('continent','Africa')['solo_ppa']:.0f}, against ~8 elsewhere) and South America the highest solo citations
per author "
f"({_sa['solo_cpa']:.0f})--not because the regions are prolific but because each lists only a handful of hyper-
selected scientists whose "
"individual careers are very large; the per-author means are inflated by tiny denominators and must be read as
such. At the sub-region "
"level (Figure 6) the same split holds: Northern America and Western Europe lead on per-paper impact, Eastern
Asia on volume.")
figure(f"{SCR}/ff_ratio_continent.png",Inches(6.9),"Figure 5. By continent – the 3 × 3 scheme (rows: solo,
senior, all; columns: "
"publications per author, citations per author, citations per publication; continents ordered by output, five-
year values). Asia's "
"high output but low per-paper impact, and South America's and Africa's tiny-base per-author spikes, stand
out.")
figure(f"{SCR}/ff_ratio_region.png",Inches(6.9),"Figure 6. By world sub-region (the eight most productive) – the
same 3 × 3 scheme "
"(five-year values).")
```

```
doc.add_heading("4.3 By country",level=2)
_us=a6('country','United States'); _cn=a6('country','China')
body(f"Country by country (Figures 7–8) the United States—by far the largest producer—is also among the most
cited per paper at every "
f"class (solo {_us['solo_cpp']:.0f}, senior {_us['sen_cpp']:.0f}, all {_us['all_cpp']:.0f}), so it leads on both
output volume and "
f"per-paper impact. China is its mirror image: high output per author (all {_cn['all_ppa']:.0f} papers) but the
lowest citations per "
f"paper of the large cohorts (all {_cn['all_cpp']:.0f}), with Japan sharing the low per-paper impact. Figure 7
isolates a single, "
"revealing slice—each country's solo-paper impact against its all-paper impact. Most large cohorts sit near the
diagonal (solo and "
"overall impact track each other), but the scatter above the line is itself informative: where a country's solo
papers out-cite its "
"average paper, its most individual work is its strongest, whereas countries below the line lean on
collaboration for their impact.")
figure(f"{SCR}/ff_ratio_scatter.png",Inches(6.5),"Figure 7. A single revealing slice of the nine measures: solo-
paper impact (citations "
"per solo paper, vertical) against all-paper impact (citations per paper overall, horizontal), by country;
bubble area = number of "
"scientists; countries with at least five scientists per year. The dashed line is equality. Points above it
(e.g. Portugal, Norway) are "
"countries whose sole-authored work is cited more heavily than their average paper—their most individual output
is their strongest; the "
"United States sits high on both axes at large scale, China low on both. The vertical spread at a given
horizontal position shows that "
"overall per-paper impact does not fix solo impact—the two are related but far from identical.")
figure(f"{SCR}/ff9_worldmaps.png",Inches(6.9),"Figure 8. The full 3 × 3 scheme mapped: nine world choropleths,
rows solo/senior/all × "
"columns publications per author, citations per author, citations per publication (five-year values; countries
with fewer than five "
"listed scientists left grey). Boundaries: Natural Earth [6].")
```

```
doc.add_heading("4.4 By U.S. state",level=2)
body("Within the United States (counts in Table 6) the pooled federal-agency block and the leading forestry
states carry most of the "
"output; on the derived rates per-paper impact is generally high across the authorship classes, consistent with
the national profile, "
"though the small per-state samples make the per-author means noisy. Full per-state derived measures (and the
Canadian and Australian "
"subdivisions) are in the workbook.")
```

```
doc.add_heading("4.5 By institution",level=2)
body("Among the most-represented institutions (Figure 9) the clearest separation is again in per-paper impact
rather than output: broad "
"research universities reach the highest citations per publication at every authorship class, while several of
the specialized forestry "
"agencies and agricultural universities that top the output counts are markedly lower. Publications per author,
by contrast, does not "
"sort the institutions in any consistent way—the head-count leaders are not the most prolific per person,
confirming at the institution "
"level the volume-impact independence seen among countries.")
figure(f"{SCR}/ff_ratio_institution.png",Inches(6.9),"Figure 9. By institution (the ten most productive) – the 3
× 3 scheme "
"(five-year values). Output per author varies without tracking per-paper impact; the institutions separate most
clearly on citations "
"per publication.")
```

```
doc.add_heading("4.6 Authorship position and the attribution of credit",level=2)
_acohort=AUTH["blocks"]["cohort"]
def _apc(pos): return AUTH["share_p"]["cohort"][pos]
```

```

def _acc(pos): return AUTH["share_c"] ["cohort"] [pos]
def _acpp(g,pos): return AUTH["cpp"] [g] [pos]
_cq=AUTH["corr"]; _ov=AUTH["overlap"]
body("The nine measures still credit an author equally for a paper whether they wrote it alone, led it, or
appeared as one of many "
"co-authors. Resolving position one step further splits each author's output into three mutually exclusive
classes, ordered from most to "
"least individually attributable: single-authored, first-authored (of a multi-author paper) and other co-
authored (every paper on which "
"the author is neither sole nor first author-last-authorship, which is not part of this study, is pooled here).
So resolved, the "
f"cohort's cited output is overwhelmingly collaborative (Table 9, Figure 10): sole-authored work is just
{_acc('single'):.0f}% of its "
f" Citations and {_apc('single'):.0f}% of its papers; first-authorship adds another {_acc('first'):.0f}%, so the
two lead positions "
f"together account for about {_acc('single')+_acc('first'):.0f}% of citations, while the remaining
~{_acc('other'):.0f}% accrues to "
"other co-authored (collaborative) work. The two groups differ only slightly in this split (Figure 10), though
the Ecology group's "
" Citations are, as everywhere, denser per paper.")
simple_table(["Authorship position","Share of papers (%)","Share of citations (%)","Citations per paper"],
[["Single-authored",f"{_apc('single'):.0f}",f"{_acc('single'):.0f}",f"{_acpp('cohort','single'):.1f}"],
["First-authored (multi-
author)",f"{_apc('first'):.0f}",f"{_acc('first'):.0f}",f"{_acpp('cohort','first'):.1f}"],
["Other co-authored (not
lead)",f"{_apc('other'):.0f}",f"{_acc('other'):.0f}",f"{_acpp('cohort','other'):.1f}"],
["All positions (undiscriminated)", "100","100",f"{_acpp('cohort','all'):.1f}"]],
aligns=["left"]+["right"]*3,
cap="Table 9. The cohort's five-year output resolved by authorship position (2020-2024). Categories are mutually
exclusive and derived "
"from the database's nested position fields—single (nps/ncs) and single-or-first (cpsf/ncsf)—so single + first +
other = all; "other" "
"is every non-lead co-authored paper (last- and middle-author positions pooled, as last-authorship is outside
this study).")
figure(f"{SCR}/ff_authcomp.png",Inches(6.4),"Figure 10. Where the cohort's citations come from, by authorship
position: single-, first- "
"and other-co-authored shares for the whole cohort and the two groups (five-year totals). Sole-authored work is
a small slice; most "
" Citations accrue to non-lead co-authored work. Segment labels are rounded independently and may not sum to
exactly 100.")
body("Because the report's headline counts do not discriminate authorship, they largely rank scientists by cited
collaboration rather "
"than by individual contribution—and the two orderings are almost unrelated. Ranking the "
f"{AUTH['n2024']:,} scientists listed in 2024 by total citations reproduces the "lead"-authorship ranking
(single or first author) only "
f"moderately (Spearman  $\rho$  = {_cq['lead: single+first (ncsf)']:.2f}) and bears essentially no relation to the
ranking by single-authored "
f" Citations ( $\rho$  = {_cq['single (ncs)']:.2f}): of the 25 most-cited scientists overall, only {_ov['single
(ncs)']} are among the 25 "
"most-cited on their sole-authored work (Figure 11). The more one narrows toward sole individual contribution,
the less an "
"undiscriminated citation count reveals about it.")
figure(f"{SCR}/ff_authrank.png",Inches(6.2),"Figure 11. How the undiscriminated (total-citation) ranking of the
cohort's scientists "
"relates to rankings that credit only certain authorship positions: Spearman correlation and top-25 overlap fall
sharply as the credited "
"set narrows from lead (single + first) to single-authored citations.")
body("The inference is direct. A citation ranking that ignores authorship position—like most of this report, and
like the composite "
"indicator that defines the source list itself—measures chiefly how much cited collaboration a scientist takes
part in, not how much "
"they individually author. That is a legitimate measure of participation in impactful research; but a reader who
means individual "
"contribution must read the single- and first-authored columns, where the ordering is very different. Section 5
makes the reversal "
"concrete for two scientists at the same institution.")

# ===== 5 DISCUSSION =====
doc.add_heading("5. Discussion",level=1)
_us_d=a6('country','United States')
body("The head-count geography (Data section) tells a familiar story—forestry's recognized expertise is
concentrated in the wealthy "
"temperate North, with Canada and the Nordic countries (Finland and Sweden, tiny by population) ranking far
above their size, and the "
"tropical Global South, where most of the world's forests are, largely absent. The gap between where forests
grow and where forestry "
"science is affiliated is a quantitative entry point into debates on research equity and capacity-building. But
the more novel finding "
"comes from resolving output by authorship position, which is why this report foregrounds the solo-senior-all
structure.")

```

```

body("Per-paper impact travels with discipline and geography, not with how much a population publishes. Within
every region the ordering "
"Is the same—solo papers cited least, senior (single- or first-authored) papers most, all papers between—and it
is stable across the "
"five releases (Section 4.1). What moves is the level: North America and western Europe are the most cited per
paper at every authorship "
f"tier (North America {a6('continent','North America')['all_cpp']:.0f} citations per paper overall), while
eastern Asia combines the "
f"highest output per scientist with the lowest per-paper impact (Asia {a6('continent','Asia')['all_cpp']:.0f}).
Publishing more does not "
"in itself lower per-paper quality; the heavy variation in citations per paper is disciplinary and geographic,
not a by-product of volume.")
body("U.S. pre-eminence is scale. The United States leads every head-count and output ranking by fielding the
largest number of elite "
f"scientists, and it is also among the most cited per paper of the large producers (solo
{us_d['solo_cpp']:.0f}, senior
f"{us_d['sen_cpp']:.0f}, all {us_d['all_cpp']:.0f} citations per paper). China and Japan are its mirror image—
high output per scientist, "
"the lowest per-paper impact of the large cohorts. The lesson for benchmarking is that a country's rank depends
on which of the nine "
"numbers is asked, so solo, senior and all output must be read together rather than collapsed into one league
table.")
body("The two groups differ in kind, not just degree. The ecology-rooted group is markedly more citation-dense
than the forestry-rooted "
f"group at every authorship level—about {A6GE['all_cpp']/A6GF['all_cpp']:.1f}× the citations per paper overall
({A6GE['all_cpp']:.0f} vs "
f"{A6GF['all_cpp']:.0f}), and a comparable multiple on solo and senior papers (Table 8, Figure 4)—so the roughly
one-sixth of the cohort "
"that is ecology-primary contributes a disproportionate share of its citations. This is the clearest single
driver of quality "
"differences in the whole dataset and reflects a real "
"disciplinary fact: ecology journals and ecological questions attract heavier citation than applied
silviculture. The taxonomy tree "
"makes the boundary concrete, with the Ecology group sitting under the Biology root and the Forestry group
mostly under Agriculture, "
"Fisheries & Forestry but reaching into Biology, Chemistry, Earth & Environmental Sciences and Economics.")
body("A forensic reading of the nine measures across the maps, charts and tables surfaces five regularities the
headline counts hide. "
f"(1) Impact peaks at the lead, not the sole, position: for the cohort and in nearly every division, citations
per paper are highest for "
f"senior work ({A6C['sen_cpp']:.0f} for the cohort), lowest for pure-solo work ({A6C['solo_cpp']:.0f}), with the
all-paper figure "
f"between ({A6C['all_cpp']:.0f})—a scientist's first-authored papers are their most-cited output, and pooling in
sole-authored work "
f"pulls the average down. (2) Volume and per-paper impact are near-orthogonal: Asia carries the most papers per
author "
f"({a6('continent','Asia')['all_ppa']:.0f}) yet is least cited per paper ({a6('continent','Asia')
['all_cpp']:.0f}), North America "
f"inverts it ({a6('continent','North America')['all_ppa']:.0f} papers, {a6('continent','North America')
['all_cpp']:.0f}) "
"citations each), and neither the country nor the institution charts show a consistent volume-versus-quality
trade-off. (3) The "
f"per-author means for the smallest regions are artefacts of selection, not productivity: South America and
Africa post solo "
f"publications per author two to three times any Global-North region ({a6('continent','South America')
['solo_ppa']:.0f} and "
f"{a6('continent','Africa')['solo_ppa']:.0f} against ~8) only because each lists a handful of hyper-cited
veterans—large numerators over "
"tiny denominators—so those cells must be read as small-sample noise, not regional strength. (4) The ecology
premium is total, not "
"partial: the Ecology group leads the Forestry group in all nine cells (Figure 4), the single largest structured
effect in the data. "
f"(5) Solo impact is not predictable from overall impact at the individual scale: among the three Georgia
reference scientists citations "
f"per solo paper run from {A6['bett']['solo_cpp']:.0f} (Bettinger) to {A6['peterson']['solo_cpp']:.0f}
(Peterson) with no tidy relation "
"to their all-paper rates, so a scientist's most individual work has to be measured directly rather than
inferred. Together these say "
"what the rank analysis says numerically: the axis you choose determines the answer.")
# ---- reference scientists: three cohort members; Kahneman + Nash discussed in text ----
_ci=REFS["cies"]; _be=REFS["bett"]; _bn=REFS["bench"]
# Daniel Kahneman, from the raw career database (Princeton; Experimental Psychology; NOT in the forestry cohort)
KAH={"name":"Kahneman, Daniel","np":196,"nc":112018,"h":88,"rank":218,"sfrank":4,"sfN":36527}
KAH["cpp"]=KAH["nc"]/KAH["np"]
def _ord(n):
    return f"{n:},{th' if 10<=n%100<=20 else {1:'st',2:'nd',3:'rd'}.get(n%10,'th'))"
_cirank=_ci["sfrank"]; _berank=_be["sfrank"]; _sfN=_ci["sfN"]
_A6ci=A6["cies"]; _A6be=A6["bett"]; _A6pe=A6["peterson"]
body("Reference scientists. It is instructive to place individual careers against these population statistics on
the same nine measures. "

```

"Three are members of the cohort, all at the University of Georgia: the present author, Chris J. Cieszewski, and Pete Bettinger (both "f"primary-Forestry authors, in roughly the top one percent of the {sfN:,} ranked Forestry-subfield authors worldwide-Bettinger "f"{_ord_berank}}, Cieszewski {_ord_cirank}} in 2024), and Chris J. Peterson, a forest-disturbance ecologist (criterion 2, primary "Ecology with secondary Forestry). Table 10 and Figure 12 give the three quantitatively against the University-of-Georgia, Forestry-"group, cohort and U.S. benchmarks. Two further scientists mark the extremes and are discussed in the text rather than tabulated because "they sit outside the forestry cohort: the mathematician John Forbes Nash Jr. (not in the database at all) and the psychologist Daniel "Kahneman (near the very top of it).")

body(f"The three Georgia colleagues reproduce the cohort-level pattern within a single institution, and the nine-number view sharpens it. "f"Bettinger is a high-volume producer-({_A6be['all_ppa']:.0f}) career papers, {_A6be['all_ppa']/_A6ci['all_ppa']:.1f}) times Cieszewski's "f"{_A6ci['all_ppa']:.0f}-yet per paper the order reverses: Cieszewski's papers average about {_A6ci['all_cpp']:.0f}) citations each "f"against Bettinger's {_A6be['all_cpp']:.0f}). The gap widens as one narrows to individual work: on solo papers Cieszewski draws "f"{_A6ci['solo_cpp']:.0f}) citations each and Bettinger only {_A6be['solo_cpp']:.0f}-Bettinger's sole-authored output is barely cited, "f"his impact being overwhelmingly collaborative. Peterson, the ecologist, sits above both at every class ({_A6pe['all_cpp']:.0f}) "f"citations per paper overall, {_A6pe['solo_cpp']:.0f}) per solo paper), a single-person instance of the ecology-over-forestry premium " "(Section 4.1).")

```
def _refrow9(m): return [f"{m['solo_ppa']:.0f}",f"{m['solo_cpa']:.0f}",f"{m['solo_cpp']:.0f}",
                        f"{m['sen_ppa']:.0f}", f"{m['sen_cpa']:.0f}", f"{m['sen_cpp']:.0f}",
                        f"{m['all_ppa']:.0f}", f"{m['all_cpa']:.0f}", f"{m['all_cpp']:.0f}"]
```

simple_table(["Reference"]+NINEHDR,
 [[f"Cieszewski, Chris J. (this author)"+_refrow9(_A6ci),
 [f"Bettinger, Pete"+_refrow9(_A6be),
 [f"Peterson, Chris J. (ecology-forestry)"+_refrow9(_A6pe),
 ["University of Georgia"+_refrow9(a6('university','University of Georgia')),
 ["Forestry group"+_refrow9(A6GF),
 ["Whole cohort"+_refrow9(A6C),
 ["United States"+_refrow9(_us_d)],
 aligns=["left"]+["right"]*9,
 cap="Table 10. The three University-of-Georgia reference scientists on the nine derived measures (career values, 2024 release), against "population benchmarks (five-year values). For an individual, publications-per-author is the career paper count and citations-per-author "the career citation count; for a population both are per-scientist means. All cells are comparable across rows."+NINENOTE)

figure(f"{SCR}/ff_refs.png",Inches(6.3),"Figure 12. The three cohort reference scientists in the (publications per author, citations per "publication) plane, with each scientist's solo, senior and all placements marked and joined by a line. Peterson (ecologist) sits "highest on per-paper impact at every class; Cieszewski's impact falls as co-authored work is added (solo is his strongest); Bettinger "is high-volume with low per-paper impact, and his solo papers (~7 citations each) are barely cited. Two external Nobel laureates lie "off the frame and are discussed in the text: John F. Nash (23 papers, not in the dataset) and Daniel Kahneman (196 papers, ~571 "citations each, near the top of the dataset).")

body("John Nash is the limiting case of the selective axis, and worth stating precisely because the dataset omits him. His 23 papers "include several of the most influential works of twentieth-century mathematics and economics: the 1951 article "Non-cooperative "games" [7], drawn from a 28-page doctoral thesis, defined the Nash equilibrium and is among the most-cited papers in the mathematical "social sciences, cited tens of thousands of times. Its economic content directly qualified Adam Smith's "invisible hand": Nash showed "that agents each pursuing rational self-interest can settle on an equilibrium that is collectively sub-optimal-the prisoner's dilemma "is the textbook case-so individual self-interest need not deliver the social optimum Smith supposed. That one short paper, and a "handful of others, carry an impact per unit of output that no country, institution or individual anywhere in this report approaches.")

body(f"Nash's absence from the list is itself the lesson. A career of 23 papers-however foundational-yields a modest h-index (an "h-index cannot exceed the paper count) and a composite score dominated by output, so a purely metric-based ranking places a "field-founding laureate below thousands of prolific mid-tier scientists, and a coverage- and volume-sensitive database omits him "entirely. The same structural effect is visible in miniature within the cohort: Bettinger's h-index "f"({_be['h2024']}) exceeds Cieszewski's ({_ci['h2024']}) chiefly because he has published far more, not because his papers are "

"individually more cited—per paper the reverse holds. h-type and composite indicators systematically cap the low-volume/high-impact "

"profile, the very h-index limitation that motivated the composite indicator in the first place [1,2]. The practical lesson repeats "

"the section's theme at the scale of individuals: which scientist 'ranks highest' depends entirely on the axis chosen, so quantity and "

"quality must be reported separately.")

body(f"Kahneman marks the opposite pole from Nash, and from inside the database. His {KAH['np']:}, indexed papers are almost exactly "

f"Bettinger's count ({_be['np2024']:}), but they carry {KAH['nc']:}, citations—about {KAH['cpp']:.0f} per paper, some "

f"{KAH['cpp']/_be['cpp2024']:.0f} times Bettinger's {_be['cpp2024']:.0f} and {KAH['cpp']/_bn['cohort'] ['cart']:.0f} times the cohort's "

f"per-paper mean of {_bn['cohort']['cart']:.0f}—for an h-index of {KAH['h']} and a place {_ord(KAH['sfrank'])} of {KAH['sfN']:}, in his "

f"Experimental-Psychology subfield and {_ord(KAH['rank'])} of the roughly 200,000 scientists in the whole list. He is the profile "

"forestry's applied literature never reaches—high output and extreme per-paper impact together—and the mirror image of Nash: the "

"database seats Kahneman at its summit yet omits Nash entirely, because it registers heavily-cited, mainstream, indexed work and loses "

"sparse, foundational or non-indexed work, the same bias Table 11 quantifies. Both are here as personal reference points—Kahneman "

"taught the author at UBC in 1985, as Bettinger is his present colleague—and both show that a single citation number, stripped of "

"authorship and of the venues a database happens to cover, ranks scientists on an axis that need not be the one a reader has in mind.")

_aci=AUTH["cies"]; _abe=AUTH["bett"]

body("Authorship position makes the same point personal, and reverses it. Of Bettinger's "

f"[_abe['nc']:}, citations only {_abe['single']['c']} (from {_abe['single']['p']} sole-authored papers) are his in the strictest "

f"sense; another {_abe['first']['c']:}, come from first-authored papers and the remaining {_abe['other']['c']:}, from other "

f"co-authored (non-lead) work. Cieszewski's smaller total ({_aci['nc']:}) is proportionally more his own—"

f"[_aci['single']['c']} single-authored and {_aci['first']['c']} first-authored citations. The effect on rank is decisive "

f"(Section 4.6): by total citations Bettinger ({_ord(_abe['rank_all'])} of the {AUTH['n2024']:}, scientists listed in 2024) sits well "

f"above Cieszewski ({_ord(_aci['rank_all'])}); by single-authored citations the order reverses, Cieszewski rising to "

f"[_ord(_aci['rank_single'])} and Bettinger falling to {_ord(_abe['rank_single'])}. Which of the two 'contributed more' is not settled "

"by the indiscriminated count; it depends entirely on whether one credits sole authorship or total participation.")

body("Both Georgia foresters (Bettinger and Cieszewski) sit below the cohort's per-paper mean of "

f"[_bn['cohort']['cart']:.0f} citations per article, and below the U.S. mean of {_bn['us']['cart']:.0f}—as expected for applied, "

"quantitative forestry, whose growth-and-yield, geomatics and forest-planning literature is cited less heavily per paper than the "

f"ecology-leaning work that lifts the cohort average (Section 4.1); their ecologist colleague Peterson, by contrast, sits above it. "

f"Their shared department, the University of Georgia, averages "

f"[_bn['uga']['cart']:.0f} citations per article across its roughly {_bn['uga']['sci_yr']:.0f} listed forestry scientists per year—a "

"figure the two of them largely set (the University of Georgia's full roster of listed members meeting the two criteria over 2020–2024 "

"is given in Table 12, below). They also fall on different branches of the taxonomy (Figure 3): Bettinger's secondary specialty is recorded "

"as Ecology, although his published work centres on forest management planning, spatial optimisation and GIS—a reminder that the "

"Science-Metrix secondary label does not always capture the quantitative character of an author's work; Cieszewski's, by contrast, is "

"Geomatics Engineering, one of the quantitative sub-sub-fields, matching a programme in forest biometrics and growth-and-yield "

"modelling. It is a distinction on which a close colleague's first-hand knowledge of the work adds what the citation record alone cannot.")

---- external validation of database coverage: two members' records vs Google Scholar ----

_cicov=100*_ci['nc2024']/4131; _becov=100*_be['nc2024']/10619

body("External validation of coverage: a case study. Because every number in this report inherits the coverage of the underlying "

"database (Scopus), it is worth confronting the database record of the two University-of-Georgia scientists compared above—the present "

"author and Bettinger, whose full publication lists are independently visible—with their public Google Scholar profiles (retrieved July "

"2026; Table 11). Google Scholar's raw lists are inflated by duplicate entries, translations of the same paper into other languages, and "

"non-article items (software files, conference websites, section headings); de-duplicating them—done reproducibly by title—reduces "

f"Cieszewski's 211 listed entries to about 170 unique works (one handbook alone is listed six times) and Bettinger's 361 listed entries "

"to about 300. Against those de-duplicated corpora the database understates both careers, but by strikingly different amounts. For "

f"Cieszewski the two sources roughly agree in proportion: the database holds {_ci['nc2024']:}, of his {4131:}, Google Scholar citations "

f"({_cicov:.0f}%) and about {100*_ci['np2024']/170:.0f}% of his works. For Bettinger the database holds "

f"({_be['nc2024']:}, of {10619:}, citations-only {_becov:.0f}%-even though it indexes roughly two-thirds of his works, because his "

"citations come heavily from books, proceedings and other documents Scopus does not index as citing sources. The two sources are biased "

"in the same direction (Scopus is journal-centric and under-covers proceedings, books, reports and regional-language outlets) but to "

"author-specific degrees. On the recent window both agree well: the database career counts grew by "

f"({_ci['np2024']-_ci['np2020']}) (Cieszewski) and {_be['np2024']-_be['np2020']}) (Bettinger) papers across the 2020-2024 releases, matching "

"their Google-Scholar output at the resolution the analysis uses.")

simple_table(["Measure", "Cieszewski: DB", "Cieszewski: Scholar", "Bettinger: DB", "Bettinger: Scholar"],

[[["Listed works (Scholar raw)", "-", "211", "-", "361"],

["Unique works (career)", f"({_ci['np2024']:},)", "~170", "{:,.format(_be['np2024']), "~300",

["Career citations", f"({_ci['nc2024']:},)", "4,131", f"({_be['nc2024']:},)", "10,619"],

["h-index", f"({_ci['h2024']})", "29", f"({_be['h2024']})", "49"],

["i10-index", "-", "60", "-", "174"],

["Scopus share of citations", f"({_cicov:.0f}%)", "(=100%)", f"({_becov:.0f}%)", "(=100%)"]],

aligns=["left", "right", "right", "right", "right"],

cap="Table 11. External validation of database coverage: the two University-of-Georgia reference scientists' records in the career "

"database (DB, 2024 release) versus their public Google Scholar profiles (July 2026). Google Scholar lists 211 works for Cieszewski and "

"361 for Bettinger; de-duplicating by title (removing repeated listings, translations and non-article items-one Cieszewski handbook is "

"listed six times) reduces these to about 170 and 300 unique works respectively. Differences reflect source coverage, not errors in "

"either source.")

body("The implication is a disclaimer with a bounded-but not negligible-scope. Absolute magnitudes (papers, citations, and the ratios "

"built from them) are Scopus-basis values and understate an author's true output and impact, the more so where the work or its citing "

"literature is proceedings-, book- or region-heavy; on another source they would move upward, and by author-specific amounts. The "

"comparative analyses are computed within one consistently-indexed source, so they are distorted only insofar as coverage bias differs "

"between the entities compared. That differential is not always small-Bettinger's citation coverage (~33%) is well below Cieszewski's "

"(~45%), a real difference between two colleagues in the same department-but in every comparison this report actually draws it preserves "

"the direction of the result: Bettinger leads Cieszewski on total citations in both sources, and by a wider margin in Google Scholar "

"(2.6x versus 1.9x), so a Scholar-basis analysis would sharpen, not reverse, the total-citation ranking. The single-authored reversal of "

"Section 4.6 is a within-database structural fact about authorship credit that an all-source citation count cannot itself confirm or "

"deny. Discrepancies relative to a Google-Scholar-basis analysis are therefore expected-material for the absolute levels, but "

"direction-preserving for the comparisons.")

body("The University of Georgia roster makes the coverage and data-quality issues concrete at the smallest scale. Over the five releases "

"the database attaches five distinct names to UGA under the two criteria, but two are spurious and are set aside here: the criterion-1 "

"records for I. D. Cave and J.-P. Schütz carry a "University of Georgia" affiliation that is evidently a Scopus assignment error-both are "

"eminent non-UGA wood scientists (Cave a New Zealand wood-physics researcher, Schütz an ETH-Zürich silviculturist), and both are among "

"the blank-country records the Methods flag. That leaves the three genuine UGA cohort members of Table 12: under criterion 1 (primary "

"subfield Forestry) the present author and Pete Bettinger, both of the Warnell School of Forestry and Natural Resources; under criterion "

"2 (primary Ecology, secondary Forestry) the forest-disturbance ecologist Chris J. Peterson of the Department of Plant Biology. The "

"database records only the institution, so the departments here are added from public university sources; the two spurious affiliations "

"are a reminder that even the affiliation field-on which all of this report's geography rests-carries occasional error.")

simple_table(["Name", "Crit.", "Primary → secondary subfield", "Department (added from public sources)"],

[[["Cieszewski, Chris J. (this author)", "1", "Forestry → Geomatics Engineering", "Warnell School of Forestry and Natural Resources"],

["Bettinger, Pete", "1", "Forestry → Ecology", "Warnell School of Forestry and Natural Resources"],

["Peterson, Chris J.", "2", "Ecology → Forestry", "Department of Plant Biology"]],

aligns=["left", "center", "left", "left"],

cap="Table 12. The three genuine University-of-Georgia members of the cohort, by selection criterion (1 = primary subfield Forestry; "

"2 = primary Ecology with secondary Forestry), 2020-2024, with name variants across releases consolidated. The source carries only the "

"institution, not the department; departments are added from public UGA sources. Two further criterion-1 records the source attaches to "

"UGA-I. D. Cave (secondary Microscopy) and J.-P. Schütz (secondary Evolutionary Biology)-are internationally-known non-UGA wood "

"scientists whose "University of Georgia" affiliation appears to be an indexing artifact, and are excluded above.")

body("A paradox at the source. The University of Georgia's presence on this list is the smallest the analysis can register while still "

"being non-zero: two primary-Forestry scientists (criterion 1) and a single ecologist working in forestry (criterion 2)-and, once the "

"two spurious affiliations of Table 12 are set aside, that is the whole of the University of Georgia's forestry cohort as the 'Forestry' "

"classification counts it (a fuller Warnell tally, across other subfield labels, follows below). Against its context the figure is jarring. The "

"University of Georgia is the state's flagship land-grant institution and the nation's first state-chartered public university (1785, "

"the birthplace of public higher education in America); its Warnell School of Forestry and Natural Resources, endowed in 1906, is the "

"oldest forestry school in the American South. And Georgia is, by a wide margin, the number-one forestry state in the United States: it "

"holds more commercial timberland than any other state-about 22 million acres-leads the nation in timber harvested and in forest-product "

"exports, and supports a forest industry of roughly forty billion dollars a year. That the flagship university of the country's foremost "

"forestry state should field only two or three of the world's most-cited forestry scientists is, on its face, hard to reconcile.")

body("Part of the shortfall is measurement, and the report's own findings say so. The list ranks only the most-cited two percent or so of "

"scientists, on a Scopus basis that-as Table 11 documents-systematically under-covers the conference proceedings, books, technical "

"reports and extension literature in which applied, quantitative and management-oriented forestry, the working core of a school like "

"Warnell, largely appears. The same coverage shortfall that roughly halved the present author's indexed citations depresses the "

"visibility of exactly this kind of forestry. The count of two therefore understates the science the university actually does; the real "

"forestry research effort is larger than the metric can see.")

body("Yet measurement does not dissolve the paradox, and the present author contends that the residue is symptomatic of a deeper malaise "

"in the university system. Across American higher education the past half-century has seen administrative ranks expand far faster than "

"faculty or research capacity: from 1975 to 2005 administrators grew by roughly 85 percent and administrative staff by some 240 percent, "

"and between 1993 and 2009 administrative positions grew about 60 percent-on the order of ten times the growth rate of tenured "

"faculty-with professional-staff growth outpacing faculty hiring across every type of institution [8]. The argument advanced here is "

"that this proliferation of administration, with the incentives and the diversion of resources it imposes, has come at the expense of "

"the sustained, discipline-deep research programmes that produce highly-cited science-that a flagship's vanishing presence in global "

"forestry science, in the very state where forestry matters most, is a visible symptom of a system that has grown its bureaucracy while "

"starving the scholarly enterprise it exists to serve. This is an interpretation, not a demonstration: the data assembled here cannot "

"isolate cause, and other factors-retirements and hiring cycles, the field's proceedings-heavy applied character, the metric's own "

"biases-contribute. But the juxtaposition is itself a datum. The country's premier forestry state; its oldest southern forestry school; "

"its flagship, land-grant, first-chartered university-and, on the world's most-used ranking of scientific impact, almost no one. Any "

"honest account of the health of the academic research enterprise has to explain that.")

body("Full disclosure, however, requires a correction to the headline count-and it cuts the other way. The Science-Matrix taxonomy files "

"each scientist under a single primary subfield, and for a natural-resources faculty that scatters people across labels. Checking the "

"affiliations of all 280 University-of-Georgia scientists on the 2024 list against the Warnell School's own roster shows that several "

"Warnell faculty appear not as "Forestry" but under Plant Biology, Fisheries, Agronomy, veterinary parasitology, and even Recreation and "

"Tourism (Table 13). Counted properly, the school's presence on the list is about eight, not two: R. O. Teskey (tree physiology, filed "

"under Plant Biology), D. Markewitz (forest soils, filed under Agronomy), G. D. Grossman and G. S. Helfman (fisheries, both emeritus), M. J. Yabsley "

"(wildlife disease, filed under parasitology) and K. M. Woosnam (parks and tourism), alongside the two Forestry-classified foresters. "

"The dispersion is the report's own taxonomy point (Figure 3) in miniature: a single-subfield label cannot hold a discipline as broad "

"as forestry and natural resources, so any tally keyed to "Forestry" alone undercounts the field, at UGA and everywhere. Eight "

"highly-cited scientists is still a modest showing for the flagship of the nation's first forestry state—the paradox is softened, not "

"dissolved—but the honest figure is eight, and the two-scientist headline is in part an artifact of classification.")

_WN=A6["warnell"]

_wn_rows=[

 ("Yabsley, Michael J.", "Mycology & Parasitology → Veterinary Sci.", "Yabsley, Michael"),

 ("Teskey, Robert O.", "Plant Biology & Botany → Forestry", "Teskey, Robert"),

 ("Woosnam, Kyle M.", "Sport, Leisure & Tourism → Reg. Planning", "Woosnam, Kyle"),

 ("Grossman, Gary D. (emeritus)", "Fisheries → Ecology", "Grossman, Gary"),

 ("Markewitz, Daniel", "Agronomy & Agriculture → Forestry", "Markewitz, Daniel"),

 ("Helfman, Gene S. (emeritus)", "Fisheries → Behavioural Science", "Helfman, Gene")]

_WNHDR=["Solo pubs", "Solo c/pub", "Senior pubs", "Sr. c/pub", "All pubs", "All c/pub"]

def _wncells(m): return

[f"{m['solo_p']:.0f}", f"{m['solo_cpp']:.0f}", f"{m['sen_p']:.0f}", f"{m['sen_cpp']:.0f}",

 f"{m['all_p']:.0f}", f"{m['all_cpp']:.0f}"]

simple_table(["Name", "Listed subfield (primary → secondary)"]+_WNHDR,

 [[disp, sub]+_wncells(_WN[k]) for disp, sub, k in _wn_rows],

 aligns=["left", "left"]+["right"]*6,

 cap="Table 13. Warnell School of Forestry and Natural Resources faculty on the 2024 list who are classified under a non-Forestry "

 "primary subfield, by publication class (career values, 2024 release; Solo = single-authored, Senior = single-or first-authored, All "

 "= every paper; c/pub = citations per publication, no citation totals shown). Identified by matching all 280 University-of-Georgia "

 "listees against the Warnell roster and public sources; the database "

 "carries no department, so this is a best-effort identification and may omit joint USDA-Forest-Service or Savannah-River-Ecology-Laboratory "

 "appointments affiliated elsewhere in the source. Grossman and Helfman are emeritus. With the two Forestry-classified foresters "

 "(Bettinger, Cieszewski) the school's presence on the list is about eight.")

body("Nonetheless, the core figure stays glaring. Whatever the exact tally—two scientists in the top "Forestry" recognition category, "

 "about eight across the whole school—it is conspicuously thin for the flagship university of a state where forestry is no marginal "

 "pursuit but one of the largest agricultural sectors of all: by the industry's economic footprint, on the order of forty billion dollars "

 "a year and roughly 140,000 jobs, it stands second only to poultry. A discipline of that economic weight, housed at the institution "

 "charged above all others with serving it, ought to field far more than two of its scientists among the world's most-cited. That it does "

 "not is the datum with which this section began, and enlarging the count from two to eight does not enlarge it enough to dissolve the "

 "point.")

INTG=json.load(open(f"{SCR}/integrity.json", encoding="utf-8"))

body("Robustness and integrity of the cohort. The database carries several diagnostics—beyond output and impact—that test how trustworthy "

 "and how concentrated the cohort's citation record is; the section closes by reading them (Table 14, Figure 13). Three point the same "

 f"way: the cohort is clean. Self-citation is moderate (median {INTG['self_med']:.0f}% of an author's citations, none above "

 f"{INTG['self_max']:.0f}%, only {INTG['self_gt25']} of {INTG['n']} authors above 25%); citation concentration shows no anomaly (the "

 f"ratio of citations to distinct citing papers is {INTG['cprat_med']:.1f} at the median and never exceeds {INTG['cprat_max']:.1f}, well "

 f"below the range that flags citation-farming); and retraction exposure is negligible—just {INTG['rw_papers']} retracted papers across "

 f"all {INTG['n']} authors ({INTG['rw_auth']} authors), with {INTG['rw_citing']} authors citing any retracted work. Visibility is high: "

 f"the median author has had {INTG['pctcited_med']:.0f}% of their papers cited at least once, so impact is broad, not confined to a few "

 "blockbusters.")

simple_table(["Diagnostic (2024 release, n = %d)"%INTG['n'], "Value"],

 [["Self-citation rate – median / 90th pct / max", f"{INTG['self_med']:.0f}% / {INTG['self_p90']:.0f}% / {INTG['self_max']:.0f}%"],

 ["Authors with self-citation above 25%", f"{INTG['self_gt25']} ({100*INTG['self_gt25']/INTG['n']:.0f}%)],

 ["Citations per distinct citing paper (cprat) – median / max", f"{INTG['cprat_med']:.2f} / {INTG['cprat_max']:.2f}"],

 ["Retracted papers in cohort / authors affected", f"{INTG['rw_papers']} / {INTG['rw_auth']}"],

 ["Papers ever cited – median share per author", f"{INTG['pctcited_med']:.0f}%"],

 ["h-index / hm-index (coauthorship-adjusted) – median", f"{INTG['h_med']:.0f} / {INTG['hm_med']:.0f} (hm ≈ {INTG['hmratio_med']:.2f}·h)"],

 ["Citation inequality – Gini / top-10% / top-1% share", f"{INTG['gini']:.2f} / {INTG['top10']:.0f}% / {INTG['top1']:.0f}%"],

 ["Career age (2024 – first publication year) – median", f"{INTG['age_med']:.0f} yr"]],

 aligns=["left", "right"],

 cap="Table 14. Robustness, integrity and distribution diagnostics for the cohort (2024 release, 854 authors). Self-citation, citation "

 "concentration and retraction exposure are screening variables, not misconduct labels; the cohort is unremarkable on all three. The "

"hm-index fractionally counts multi-authored papers [10], so $hm \approx \frac{1}{2}h$ quantifies the cohort's heavy collaboration. Retraction columns are "
 "populated only in the 2023-2024 releases.")
 figure(f"{SCR}/ff_integrity.png", Inches(6.7), "Figure 13. Left: Lorenz curve of career citations across the 854 listed scientists—even "
 f"within this citation elite the top tenth holds about a third of all citations (Gini {INTG['gini']:.2f}), so means are pulled by a heavy "
 "right tail. Right: the coauthorship-adjusted hm-index against the h-index; the cloud sits along the $hm = \frac{1}{2}h$ line, so accounting for "
 "multi-authorship roughly halves the index. The three University-of-Georgia reference scientists are marked.")
 body("Two readings follow. Methodologically, the concentration and the h-to-hm gap reinforce the report's recurring caution—population "
 "means are pulled by a heavy right tail and by collaboration, so the medians, per-paper rates and authorship-resolved figures used "
 "throughout are the honest summaries, not raw totals or a single index, in line with responsible-metrics guidance that indicators be "
 "read with context and never collapsed into one score [11]. Substantively, the integrity screen matters for a cohort of "
 "elite authors precisely because citation-based selection can reward gaming: that this forestry cohort shows low self-citation, no "
 "citation-concentration anomaly and almost no retraction exposure means its standing rests on ordinary, broadly-cited work. The "
 f"reference scientists are likewise unremarkable (self-citation {INTG['refs']['Peterson']['self']:.0f})- {INTG['refs']['Bettinger']['self']:.0f}%), "
 f"and the h-to-hm halving holds for each: Bettinger's h of {INTG['refs']['Bettinger']['h']:.0f} falls to an hm of "
 f"{INTG['refs']['Bettinger']['hm']:.0f} once his heavy co-authorship is discounted—the same collaboration signal Section 4.6 read from "
 "his authorship positions.")
 body("Caveats on the comparisons. All of these authors already sit at the top of the citation distribution, so every comparison is among "
 "an elite; citations accumulate with career length, so nations or institutions with older cohorts look more impactful; per-paper impact "
 "is shaped by field mix (the ecology effect above), publication venue and language; and the small-cohort leaders (Netherlands, Brazil, "
 "Portugal) rest on a handful of authors and should not be over-read. Affiliation geography credits the current host institution, not "
 "origin or training.")
 body("Uses. Reporting solo, senior and all output with its per-paper impact lets funders and forestry ministries benchmark national "
 "standing on volume and on per-paper impact independently, and at the level of individually-authored work rather than total "
 "collaboration; forestry schools benchmark peers and target recruitment; editors widen reviewer and speaker pools using the geography; "
 "and science-of-science researchers can study the authorship-credit structure and the forestry-ecology boundary directly.")
 body("Limitations. Affiliation country is current institution, not nationality or origin. Scopus coverage biases toward "
 "English-language journals and under-covers proceedings, books and reports, likely understating tropical, non-English and "
 "applied-technical forestry—the case study above (Table 10) quantifies the effect for two authors at roughly two- to three-fold on "
 "citations, and shows the coverage shortfall is author-specific yet direction-preserving for the report's comparisons. Publication and "
 "citation sums inherit the skew of "
 "citation distributions—a few superstar authors dominate—so ratios for small divisions are volatile (hence the minimum-size filters). "
 "The counting unit is the author-year, so annual changes mix genuine entry with re-computation. The two-group selection is "
 "deliberately conservative on the ecology side; a looser rule would enlarge the Ecology group. Subnational placement leaves some U.S. "
 "institutions "Unknown" and pools federal agencies.")

===== 6 CONCLUSION =====
 doc.add_heading("6. Conclusion", level=1)
 body(f"By head-count the forestry research cohort—{nF:,} primary-forestry authors plus {nE:,} ecology-rooted authors who also work in "
 f"forestry, {nU:,} in all—is small, growing, and strongly concentrated in the Global North: North America and Europe dominate, the "
 f"United States and Canada lead among countries, Finland and Sweden are conspicuously over-represented, and {funi['key']} is the most "
 "represented institution (Data section). The analytical contribution, however, is in resolving output by authorship position. Reading "
 "solo, senior and all publications with their citations per paper together shows that per-paper impact travels with discipline and "
 "geography, not with how much a population publishes; that the United States leads by scale and is also among the most cited per paper "
 "of the large producers, while China and Japan show the opposite high-volume, low-per-paper-impact profile; that the ecology-rooted "

"group is about twice as cited per paper as the forestry-rooted group at every authorship level; and that an undiscriminated citation "

"ranking measures cited collaboration more than individual authorship, so narrowing to solo- or first-authored credit reorders "

"scientists substantially. Which country, institution or individual looks strongest depends on which of the nine measures one asks "

"for—so they must be reported together.")

===== 7 SUMMARY =====

doc.add_heading("7. Project Summary",level=1)

body(f"Five career-long releases of the Stanford / Ioannidis database (2020–2024) were filtered to a "Forestry plus Ecology" cohort of "

f"({nU:,}) records—Forestry ({nF:,}): primary subfield Forestry) plus Ecology ({nE:,}): primary Ecology with secondary Forestry)—both "

"groups requiring forestry. The Data section (Tables 1–7) gives only raw counts—the number of scientists and of solo (single-authored), "

"senior (single- or first-authored) and all publications for every division (group, continent, world sub-region, country, U.S. state and "

"institution)—with the two groups distinguished and a multi-root field-subfield-subfield taxonomy. The Results (Section 4) derive, for "

"each authorship class, three rates—publications per author, citations per author, citations per publication—a 3 × 3 scheme of nine "

"measures shown as 3 × 3 charts, and read the patterns: per-paper impact tracks discipline and geography rather than publication volume, "

"and holds its solo-senior-all ordering across regions. A further analysis (Section 4.6) resolves each author's output into single, first "

"and other (non-lead) co-authored positions and shows that an undiscriminated citation ranking measures cited collaboration rather than "

"individual contribution (ranking by total citations is essentially uncorrelated with ranking by single-authored citations). The "

"Discussion adds individual scientists as concrete reference points—the author and two UGA colleagues (all in the cohort); the "

"mathematician John Nash as an external low-volume/high-impact case the database omits entirely; and Daniel Kahneman as an in-database "

"high-impact extreme—showing the same authorship-credit reversal at the scale of a single career; it also lists the University of "

"Georgia's cohort members by criterion (Table 12) and bounds the database's Scopus coverage against Google Scholar (Table 11); and a "

"closing robustness screen (Table 14, Figure 13) confirms low self-citation, no citation-concentration anomaly and negligible retraction "

"exposure, while quantifying the cohort's heavy collaboration ($hm \approx \frac{1}{2}h$) and citation concentration (Gini 0.45). Appendix "

"A profiles the Ecology group on its own terms; Appendix B notes that "

"single-authored (solo) output is carried in the "Solo" columns of every table throughout. Deliverables: this report; a comprehensive "

"workbook with full per-year, per-metric rankings and per-group sheets; and the source career database (Excel and CSV).")

===== 8 REFERENCES =====

doc.add_heading("8. References and Data Sources",level=1)

refs=[

"Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. PNAS, 102(46), 16569–16572. <https://doi.org/10.1073/pnas.0507655102>",

"Ioannidis, J. P. A., Klavans, R., & Boyack, K. W. (2016). Multiple citation indicators and their composite across scientific disciplines. PLoS Biology, 14(7), e1002501. <https://doi.org/10.1371/journal.pbio.1002501>",

"Ioannidis, J. P. A., Baas, J., Klavans, R., & Boyack, K. W. (2019). A standardized citation metrics author database annotated for scientific field. PLoS Biology, 17(8), e3000384. <https://doi.org/10.1371/journal.pbio.3000384>",

"Ioannidis, J. P. A., Boyack, K. W., & Baas, J. (2020). Updated science-wide author databases of standardized citation indicators. PLoS Biology, 18(10), e3000918. <https://doi.org/10.1371/journal.pbio.3000918>",

"Archambault, É., Beauchesne, O. H., & Caruso, J. (2011). Towards a multilingual, comprehensive and open scientific journal ontology. Proc. 13th ISSI Conference, 66–77. (Science-Metrix classification.),

"Natural Earth (2024). Free vector and raster map data, 1:110m Admin 0 – Countries. <https://www.naturalearthdata.com> (public domain).",

"Nash, J. F. (1951). Non-cooperative games. Annals of Mathematics, 54(2), 286–295. <https://doi.org/10.2307/1969529> (Nash equilibrium; Nobel Memorial Prize in Economic Sciences 1994).",

"Ginsberg, B. (2011). The Fall of the Faculty: The Rise of the All-Administrative University and Why It Matters. Oxford University Press. See also Desrochers, D. M., & Kirshstein, R. (2014), Labor Intensive or Labor Expensive? Changing Staffing and Compensation Patterns in Higher Education, Delta Cost Project / American Institutes for Research.",

"Ioannidis, J. P. A. (2025). August 2025 data-update for "Updated science-wide author databases of standardized citation indicators" (Version 8) [Data set]. Elsevier Data Repository. <https://doi.org/10.17632/btchxktzyw.8>",

"Schreiber, M. (2008). To share the fame in a fair way, hm modifies h for multi-authored manuscripts. New Journal of Physics, 10, 040201. <https://doi.org/10.1088/1367-2630/10/4/040201> (coauthorship-adjusted h-index).",

"Hicks, D., Wouters, P., Waltman, L., de Rijcke, S., & Rafols, I. (2015). The Leiden Manifesto for research metrics. Nature, 520, 429–431. <https://doi.org/10.1038/520429a>. See also DORA (2013), San Francisco Declaration on Research Assessment, <https://sfidora.org>.",

]

for i,rf in enumerate(refs,1):

```

par=doc.add_paragraph(); par.paragraph_format.left_indent=Inches(0.35);
par.paragraph_format.first_line_indent=Inches(-0.35); par.paragraph_format.space_after=Pt(6)
a=par.add_run(f"[{i}] "); a.bold=True; a.font.name="Arial"; a.font.size=Pt(9.5)
b=par.add_run(f); b.font.name="Arial"; b.font.size=Pt(9.5)
p("Data sources.",size=9.5,bold=True,before=4,after=2)
body("Primary: career-long author tables (Table_1_Authors_career), dataset doi:10.17632/btchxktzyw, versions
3/5/6/7/8, via the Elsevier "
"Digital Commons / Mendeley Data public API. Field/subfield taxonomy: Table_2 threshold tables in the same
dataset. Base maps: Natural "
"Earth (world) and a public US-state boundary file. External coverage validation (Table 11): the public Google
Scholar profiles of the "
"two reference scientists, https://scholar.google.com/citations?user=npjSdoAAAAJ (Cieszewski) and "
"https://scholar.google.com/citations?user=rvCOS60AAAAJ (Bettinger), retrieved July 2026 and de-duplicated by
title. Analysis and "
"figures: Python (pandas, openpyxl, matplotlib).",size=9.5)

# ===== APPENDIX A: THE ECOLOGY GROUP (CRITERION 2) =====
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
doc.add_heading("Appendix A. The Ecology Group (Criterion 2) in Detail",level=1)
GPE=A6["group_pack"]["Ecology"]; eby=GPE["by"]
body(f"This appendix profiles only the Ecology group—selection criterion 2: authors whose primary Science-Metrix
subfield is Ecology "
f"and whose secondary subfield is Forestry ({nE:,} author-year records, {100*nE/nU:.0f})% of the {nU:,}-record
cohort). These are "
"ecologists who also work in forestry; like the Forestry group, every one of them has Forestry as a subfield, so
this is still a "
"strictly forestry-defined population—just the ecology-primary half of it. Following the main report, Section
A.1 gives its raw counts "
"and Section A.2 its derived nine-measure premium; full per-country and per-institution breakdowns are in the
companion workbook.")
doc.add_heading("A.1 Counts by release year and continent",level=2)
body("The Ecology group is even more Global-North-concentrated than the cohort as a whole—North America and
Europe supply almost all of "
"it. Tables A1 and A2 give its raw scientist and publication counts by release year and by continent; the
derived rates follow in A.2.")
simple_table(["Release year"]+CNTHDR,
[[str(y),f"{eby['n'][i]:,.0f}",f"{eby['solo_p'][i]:,.0f}",f"{eby['sen_p'][i]:,.0f}",f"{eby['all_p'][i]:,.0f}"]
for i,y in enumerate(YEARS)],
aligns=["left"]+["right"]*4,
cap="Table A1. Ecology group (criterion 2) scientist and publication counts by release year (scientists = listed
records that year). "+CNTNOTE)
count_table(GPE["continent"],"Continent","Table A2. Ecology group – scientist and publication counts by continent
(five-year values, ordered by output).")
doc.add_heading("A.2 The ecology impact premium (derived)",level=2)
body(f"On the derived rates the group's defining feature is per-paper impact. Its solo papers draw about
{A6GE['solo_cpp']:.0f} citations "
f"each against the Forestry group's {A6GF['solo_cpp']:.0f}, its senior papers {A6GF['sen_cpp']:.0f} against
{A6GF['sen_cpp']:.0f}, and "
f"across all papers {A6GE['all_cpp']:.0f} against {A6GF['all_cpp']:.0f}—roughly a two-fold premium at every
authorship class—while it "
f"also publishes more per author (all {A6GE['all_ppa']:.0f} vs {A6GF['all_ppa']:.0f} papers). The full nine-
measure comparison against "
"the Forestry group is Table 8 and Figure 4 of the main report. Country- and institution-level rates are not
tabulated here: with only "
f"about {A6GE['sci_yr']:.0f} scientists per year the group has too few members per country to give stable per-
author means, so the "
"reliable comparison for this group is at the group and continent level.")

# ===== APPENDIX B: NOTE ON SINGLE-AUTHOR (SOLO) PUBLICATIONS =====
doc.add_paragraph().add_run().add_break(WD_BREAK.PAGE)
doc.add_heading("Appendix B. Note on single-author (solo) publications",level=1)
body("An earlier version of this report carried two further appendices replicating the whole analysis for single-
authored publications "
"only. In the presentation adopted throughout, single-authored output is no longer a separate analysis: solo is
the first of the three "
"authorship classes carried everywhere—its publication counts in the Data section (the "Solo pubs" column of
Tables 1-7 and A1-A2) and "
"its derived rates (solo publications per author, citations per author and citations per publication) as the top
row of every 3 × 3 "
"Results chart and of the reference-scientist table (Table 10). The solo figures can be read directly against
the senior and all—"
"publication figures alongside them, which is more informative than the standalone tabulation those appendices
provided; the separate "
"appendices are therefore no longer needed.")
body("Two features of solo output stand out across those tables. First, single-authored work is a small slice of
the cohort—about "
f"{100*A6C['solo_p']/A6C['all_p']:.0f}% of its papers are solo—so absolute solo counts run far below the all-
publication counts. "
f"Second, across the cohort solo papers are cited somewhat less per paper ({A6C['solo_cpp']:.0f}) than senior
({A6C['sen_cpp']:.0f}) or "

```

```

f"all papers ({A6C['all_cpp']:.0f}) (Section 4.1), consistent with sole-authored work often being reviews or
conceptual pieces rather "
"than large empirical collaborations. But the between-group and geographic orderings that hold for all papers
hold for solo papers too: "
"the Ecology group out-cites the Forestry group, and North America out-cites Asia, on solo work as on the rest.
The complete "
"per-scientist single-authored rankings by division are in the companion workbook.")

doc.save(OUT)
print("saved:",OUT,"| paragraphs:",len(doc.paragraphs),"| tables:",len(doc.tables))

```

14. build_index.py (SHA-256 fe76e73d4350e303...; 13 lines)

```

import pandas as pd
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
rows = []
for tag, country in [("us", "United States"), ("can", "Canada"), ("aus", "Australia")]:
    d = pd.read_csv(f"{SCR}/inst_{tag}.csv")
    d.columns = ["inst", "n"]
    for _, r in d.iterrows():
        rows.append({"country": country, "inst": r["inst"], "n": int(r["n"])})
idx = pd.DataFrame(rows)
idx.insert(0, "id", range(len(idx)))
idx.to_csv(f"{SCR}/inst_index.csv", index=False, encoding="utf-8")
print("total institutions to map:", len(idx))
print(idx.groupby("country").size().to_dict())

```

15. build_report.py (SHA-256 74f9e03a3bb6288e...; 192 lines)

```

import json, pandas as pd
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill, Alignment, Border, Side
from openpyxl.utils import get_column_letter

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS = [2020, 2021, 2022, 2023, 2024]

rankings = json.load(open(f"{SCR}/rankings.json", encoding="utf-8"))
subdiv = {int(k): v for k, v in json.load(open(f"{SCR}/inst_subdiv.json", encoding="utf-8")).items()}
idx = pd.read_csv(f"{SCR}/inst_index.csv")
idx["subdivision"] = idx["id"].map(subdiv).fillna("Unknown")
lookup = {(r.country, r.inst): r.subdivision for r in idx.itertuples()}

df = pd.read_excel(CAREER, sheet_name="Data")
CN = {"usa": "United States", "can": "Canada", "aus": "Australia"}
def subdiv_of(row):
    c = CN.get(row["cntry"])
    if not c: return None
    return lookup.get((c, row["inst_name"]), "Unknown")
df["subdivision"] = df.apply(subdiv_of, axis=1)

def field_of(m):
    m=str(m); f="Forestry" in m; e="Ecology" in m
    return "Both" if (f and e) else ("Forestry" if f else "Ecology")
df["field"] = df["matched_subfield"].map(field_of)

# ----- styling -----
ARIAL = "Arial"
hdr_font = Font(name=ARIAL, size=10, bold=True, color="FFFFFF")
hdr_fill = PatternFill("solid", start_color="2F5233")
title_font = Font(name=ARIAL, size=14, bold=True, color="1F3A26")
sub_font = Font(name=ARIAL, size=10, italic=True, color="555555")
base_font = Font(name=ARIAL, size=10)
bold_font = Font(name=ARIAL, size=10, bold=True)
thin = Side(style="thin", color="D9D9D9")
border = Border(bottom=thin)
wb = Workbook()

def style_header(ws, row, ncol):
    for c in range(1, ncol+1):
        cell = ws.cell(row=row, column=c)
        cell.font = hdr_font; cell.fill = hdr_fill

```

```

        cell.alignment = Alignment(horizontal="center" if c>1 else "left", vertical="center")

def write_matrix(ws, start_row, records, keycol, keyhdr, note=None, share=True, total_label=None):
    """records: list of dicts with keycol + year cols + Total_allyears."""
    dfm = pd.DataFrame(records)
    if dfm.empty:
        return start_row+1
    cols = [keycol] + [str(y) for y in YEARS] + ["Total (all yrs)"]
    if share: cols += ["Share 2024"]
    r = start_row
    for j, h in enumerate(cols, 1):
        ws.cell(row=r, column=j, value=h)
    style_header(ws, r, len(cols))
    tot2024 = sum(rec.get(2024, rec.get("2024",0)) for rec in records)
    r += 1
    for rec in records:
        y = {yy: rec.get(yy, rec.get(str(yy),0)) for yy in YEARS}
        ws.cell(row=r, column=1, value=rec[keycol]).font = base_font
        for j, yy in enumerate(YEARS, 2):
            cc = ws.cell(row=r, column=j, value=int(y[yy])); cc.font = base_font
            tcell = ws.cell(row=r, column=2+len(YEARS), value=int(rec.get("Total_allyears", sum(y.values()))))
            tcell.font = bold_font
        if share:
            sc = ws.cell(row=r, column=3+len(YEARS), value=(y[2024]/tot2024 if tot2024 else 0))
            sc.font = base_font; sc.number_format = "0.0%"
        for j in range(1, len(cols)+1):
            ws.cell(row=r, column=j).border = border
        r += 1
    # total row
    ws.cell(row=r, column=1, value=(total_label or "TOTAL")).font = bold_font
    for j, yy in enumerate(YEARS, 2):
        tv = sum(int(rec.get(yy, rec.get(str(yy),0))) for rec in records)
        ws.cell(row=r, column=j, value=tv).font = bold_font
    ws.cell(row=r, column=2+len(YEARS), value=sum(int(rec.get("Total_allyears",0)) for rec in records)).font = bold_font
    if share:
        ws.cell(row=r, column=3+len(YEARS), value=1.0).number_format="0.0%"
        ws.cell(row=r, column=3+len(YEARS)).font = bold_font
    r += 1
    if note:
        ws.cell(row=r, column=1, value=note).font = sub_font
        r += 1
    return r+1

def subnat_matrix(country_code):
    sub = df[df.cntry==country_code]
    t = sub.pivot_table(index="subdivision", columns="year", values="authfull", aggfunc="count", fill_value=0)
    for y in YEARS:
        if y not in t.columns: t[y]=0
    t = t[YEARS]; t["Total_allyears"]=t.sum(axis=1)
    t = t.sort_values(2024, ascending=False).reset_index()
    return t.to_dict(orient="records")

# ----- Sheet: Summary -----
ws = wb.active; ws.title = "Summary"
ws.column_dimensions["A"].width = 4
for col in "BCDEFGH": ws.column_dimensions[col].width = 16
ws.column_dimensions["B"].width = 34
ws["B2"] = "Stanford / Ioannidis Citation List - Representation Report"; ws["B2"].font = title_font
ws["B3"] = "Career-long database, Forestry + Ecology subfields, data years 2020-2024"; ws["B3"].font = sub_font
m = rankings["_meta"]
rpy = {int(k): v for k, v in m["rows_per_year"].items()}
rows = [
    ("", ""),
    ("Scope", "Authors appearing on each annual career-list release (2020-2024). Unit = one scientist in one release year."),
    ("Total list entries", f"{sum(rpy.values()):,} across 5 releases"),
    ("Entries per release", " ".join(f"{y}={rpy[y]:,}" for y in YEARS)),
    ("Distinct countries", str(m["distinct_countries"])),
    ("Distinct institutions (2024)", str(m["distinct_institutions_2024"])),
    ("", ""),
    ("Fields present", "Forestry (subfield) and Ecology (subfield). Both are QUANTITATIVE natural-science fields."),
    ("Quant vs qualitative", "No qualitative field exists in this subset, so that split is not applicable. A Forestry-vs-Ecology breakdown is provided instead (see 'By Field')."),
    ("Nationality", "DROPPED. The source records only current institutional country (cntry); no nationality, birth country, or degree-country field exists, and it cannot be reliably derived."),
    ("Subnational (states)", "US states, Canadian provinces, Australian states derived from each institution's main-campus location (LLM geocoding + verification of top institutions). Distributed federal agencies shown as 'Federal/Multi-site.'"),

```

```

("Counting note", "Career metrics are cumulative; the same scientist recurs across release years. Counts are
per-release-year, so a person is counted once per year they appear (no cross-year de-duplication)."),
("Source", "Ioannidis, J.P.A., Elsevier Data Repository doi:10.17632/btchxktzyw, versions 3/5/6/7/8."),
]
r = 5
for k, v in rows:
    ws.cell(row=r, column=2, value=k).font = bold_font
    c = ws.cell(row=r, column=3, value=v); c.font = base_font; c.alignment = Alignment(wrap_text=True,
vertical="top")
    ws.merge_cells(start_row=r, start_column=3, end_row=r, end_column=8)
    ws.row_dimensions[r].height = 30
    r += 1

# ----- dimension sheets -----
def new_sheet(name):
    s = wb.create_sheet(name)
    s.column_dimensions["A"].width = 40
    for col in "BCDEFGHI": s.column_dimensions[col].width = 13
    return s

s = new_sheet("By Continent")
write_matrix(s, 1, rankings["continent"], "continent", "Continent")

s = new_sheet("By World Region")
write_matrix(s, 1, rankings["region"], "region", "World sub-region")

s = new_sheet("By Country")
write_matrix(s, 1, rankings["country"], "country", "Country")

s = new_sheet("By US State")
write_matrix(s, 1, subnat_matrix("usa"), "subdivision", "US state",
note="State = main-campus location of each author's institution. 'Federal/Multi-site' = distributed
national agencies (USDA Forest Service, USGS, etc.).")

s = new_sheet("By Canada Province")
write_matrix(s, 1, subnat_matrix("can"), "subdivision", "Province/Territory")

s = new_sheet("By Australia State")
write_matrix(s, 1, subnat_matrix("aus"), "subdivision", "State/Territory")

s = new_sheet("By University (Top 60)")
write_matrix(s, 1, rankings["university_top60"], "inst_name", "Institution",
note="Top 60 institutions worldwide by 2024 count. 'Large institutions only' are named in the
source; blank-institution rows excluded here.")

s = new_sheet("By Field")
r = write_matrix(s, 1, rankings["field"], "field", "Field (subfield match)", share=True)
# field x continent crosstab 2024
r += 1
s.cell(row=r, column=1, value="Field x Continent (2024 release)").font = title_font
r += 1
ctrecs = rankings["field_by_continent_2024"]
fcols = [c for c in ctrecs[0].keys() if c != "continent"]
hdr = ["Continent"] + fcols
for j, h in enumerate(hdr, 1): s.cell(row=r, column=j, value=h)
style_header(s, r, len(hdr)); r += 1
for rec in ctrecs:
    s.cell(row=r, column=1, value=rec["continent"]).font = base_font
    for j, fc in enumerate(fcols, 2):
        s.cell(row=r, column=j, value=int(rec[fc])).font = (bold_font if fc=="Total" else base_font)
    r += 1

# freeze header rows on data sheets
for name in wb.sheetnames:
    if name != "Summary":
        wb[name].freeze_panes = "A2"

dest = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_Representation_Report.xlsx"
wb.save(dest)
print("saved report:", dest)

# quick console essence
def top(records, key, n=5):
    return [(rec[key], rec.get(2024, rec.get("2024",0))) for rec in records[:n]]
print("continents 2024:", top(rankings["continent"],"continent",6))
print("countries 2024 top8:", top(rankings["country"],"country",8))

```

```

print("US states 2024 top8:", [(x["subdivision"], x[2024]) for x in subnat_matrix("usa")[:8]])
print("universities 2024 top8:", top(rankings["university_top60"],"inst_name",8))
print("fields:", top(rankings["field"],"field",3))

```

16. build_workbook2.py (SHA-256 e92658cd87883293...; 133 lines)

```

# -*- coding: utf-8 -*-
import json
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill, Alignment
from openpyxl.utils import get_column_letter
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_Representation_Report.xlsx"
D = json.load(open(f"{SCR}/refine_data.json", encoding="utf-8"))
YEARS = [2020,2021,2022,2023,2024]
GREEN="2F5233"; LIGHT="E7EEE3"
hf=Font(name="Arial",size=10,bold=True,color="FFFFFF"); hfill=PatternFill("solid",start_color=GREEN)
bf=Font(name="Arial",size=10); bb=Font(name="Arial",size=10,bold=True)
wb=Workbook(); wb.remove(wb.active)
INTFMT="#,##0"

def sheet_metric_block(ws, start, title, records, keyhdr, metric, with_country=False, intfmt=INTFMT):
    r=start
    c=ws.cell(row=r,column=1,value=title); c.font=Font(name="Arial",size=11,bold=True,color=GREEN); r+=1
    cols=[keyhdr]+(["Country"] if with_country else [])+[str(y) for y in YEARS]+["Total","Avg/yr"]
    for j,h in enumerate(cols,1):
        cc=ws.cell(row=r,column=j,value=h); cc.font=hf; cc.fill=hfill
        cc.alignment=Alignment(horizontal="left" if j==1 else "center")
    r+=1; first=r
    for i,rec in enumerate(records):
        col=1
        ws.cell(row=r,column=col,value=rec["key"]).font=bf; col+=1
        if with_country: ws.cell(row=r,column=col,value=rec.get("country","")).font=bf; col+=1
        for y in YEARS:
            cc=ws.cell(row=r,column=col,value=rec[str(y)]); cc.font=bf; cc.number_format=intfmt; col+=1
            cc=ws.cell(row=r,column=col,value=rec["total"]); cc.font=bb; cc.number_format=intfmt; col+=1
            cc=ws.cell(row=r,column=col,value=round(rec["avg"],1)); cc.font=bf; cc.number_format="#,##0.0"; col+=1
            if i%2==1:
                for j in range(1,len(cols)+1): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
        r+=1
    ws.freeze_panes=ws.cell(row=first,column=2)
    w0 = 34 if not with_country else 30
    ws.column_dimensions["A"].width=w0
    if with_country: ws.column_dimensions["B"].width=18
    off = 3 if with_country else 2
    for j in range(off,len(cols)+1): ws.column_dimensions[get_column_letter(j)].width=12
    return r+2

# --- Summary ---
ws=wb.create_sheet("Summary"); ws.column_dimensions["A"].width=30
for col in "BCDEFG": ws.column_dimensions[col].width=15
ws["A1"]="Stanford / Ioannidis Citation List - Forestry + Ecology Representation (career-long)";
ws["A1"].font=Font(name="Arial",size=13,bold=True,color=GREEN)
ws["A2"]="Breakdowns by scientists, publications and citations, per release year 2020-2024";
ws["A2"].font=Font(name="Arial",size=10,italic=True,color="555555")
tot=D["totals"]; r=4
ws.cell(row=r,column=1,value="Metric").font=hf; ws.cell(row=r,column=1).fill=hfill
for j,y in enumerate(YEARS,2): c=ws.cell(row=r,column=j,value=str(y)); c.font=hf; c.fill=hfill
r+=1
for m,label in [("scientists","Scientists (list entries)"),("publications","Publications (sum of career papers)"),("citations","Citations (sum of career citations)"]):
    ws.cell(row=r,column=1,value=label).font=bf
    for j,y in enumerate(YEARS,2):
        c=ws.cell(row=r,column=j,value=tot[m][str(y)]); c.font=bf; c.number_format=INTFMT
    r+=1
notes=[
    "",
    "Metric definitions: Scientists = number of ranked authors on the list. Publications = sum of each author's career paper count (np). Citations = sum of each author's career citations (nc).",
    "Counting unit = author-year: each scientist is counted once per release year in which they appear. No cross-year de-duplication.",
    "Universities are grouped by name and labelled with their modal country. Full lists (all 2,344 institutions; all 83 countries) are on the per-metric sheets.",
    "Quantitative-oriented = author whose primary or secondary Science-Metrix subfield is a quantitative discipline (Statistics, Engineering, Geomatics/GIS, AI & Image Processing/RS, Mathematics, Bioinformatics).",
    "Source: Ioannidis, doi:10.17632/btchxktzyw, career tables v3/v5/v6/v7/v8.",
]

```

```

for n in notes:
    ws.cell(row=r,column=1,value=n).font=Font(name="Arial",size=9,italic=True,color="555555")
    ws.merge_cells(start_row=r,start_column=1,end_row=r,end_column=7); r+=1

# --- Continent / Region / Field / Orientation (all 3 metrics stacked) ---
def multi_metric_sheet(name, key, keyhdr, with_country=False):
    ws=wb.create_sheet(name); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By CITATIONS")]:
        r=sheet_metric_block(ws,r,f"{name} - {lab}",D[key][m],keyhdr,m,with_country=with_country)
    return ws
multi_metric_sheet("Continent","continent","Continent")
multi_metric_sheet("World Region","region","World sub-region")
multi_metric_sheet("Field","field","Field (subfield match)")
multi_metric_sheet("Orientation","orient","Quant/Qual orientation")
multi_metric_sheet("US States","us_state","US state / category")
multi_metric_sheet("Canada & Australia","ca_prov","Canada province/territory") # ca; append au below
wsCA=wb["Canada & Australia"]; # add Australia blocks
r=wsCA.max_row+3
for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By CITATIONS")]:
    r=sheet_metric_block(wsCA,r,f"Australia - {lab}",D["au_state"][m],"Australia state/territory",m)

# --- Country: one sheet per metric (full 83) ---
for m,lab in [("scientists","Scientists"),("publications","Publications"),("citations","Citations")]:
    ws=wb.create_sheet(f"Country by {lab}")
    sheet_metric_block(ws,1,f"All countries ranked by {lab.lower()} (per release year)",D["country"][m],"Country",m)

# --- University: one sheet per metric (full 2,344, with country) ---
for m,lab in [("scientists","Scientists"),("publications","Publications"),("citations","Citations")]:
    ws=wb.create_sheet(f"University by {lab}")
    sheet_metric_block(ws,1,f"All institutions ranked by {lab.lower()} (with country, per release year)",D["university"][m],"Institution",m,with_country=True)

# --- Taxonomy tree sheet ---
ws=wb.create_sheet("Taxonomy Tree")
for j,h in enumerate(["Domain (Field)","Field (Sub-Field)","Subfield (Sub-Sub-Field)","Records 2020-24","Avg/yr","Quantitative?","Target subfield?"],1):
    c=ws.cell(row=1,column=j,value=h); c.font=hf; c.fill=hfill
tree=D["tree"]
rows=sorted(tree.items(), key=lambda kv: (kv[1]["domain"],kv[1]["field"],-kv[1]["total"]))
r=2
for sf,info in rows:
    ws.cell(row=r,column=1,value=info["domain"]).font=bf
    ws.cell(row=r,column=2,value=info["field"]).font=bf
    cc=ws.cell(row=r,column=3,value=sf); cc.font=(bf if info["target"] else bf)
    c4=ws.cell(row=r,column=4,value=info["total"]); c4.font=bf; c4.number_format=INTFMT
    ws.cell(row=r,column=5,value=round(info["avg"],1)).number_format="#.##0.0"
    ws.cell(row=r,column=6,value=("Yes" if info["quant"] else "")).font=bf
    ws.cell(row=r,column=7,value=("Yes" if info["target"] else "")).font=bf
    if r%2==1:
        for j in range(1,8): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
    r+=1
ws.freeze_panes="A2"
for col,w in zip("ABCDEFGH",(26,32,34,14,10,13,15)): ws.column_dimensions[col].width=w

# --- Map data (avg) ---
ws=wb.create_sheet("Map data (avg)")
ws.cell(row=1,column=1,value="Average per release year (2020-2024). ISO3 for world; state for US.").font=Font(name="Arial",size=9,italic=True,color="555555")
hdr=["Key"]+[f"World {m}" for m in ["scientists","publications","citations"]]
for j,h in enumerate(hdr,1): c=ws.cell(row=2,column=j,value=h); c.font=hf; c.fill=hfill
mw=D["map_world_avg"]; keys=sorted(mw["scientists"], key=lambda k:-mw["scientists"][k]); r=3
for k in keys:
    ws.cell(row=r,column=1,value=k).font=bf
    for j,m in enumerate(["scientists","publications","citations"],2):
        c=ws.cell(row=r,column=j,value=round(mw[m].get(k,0),1)); c.number_format="#.##0.0"; c.font=bf
    r+=1
ws.column_dimensions["A"].width=10
for col in "BCD": ws.column_dimensions[col].width=16

wb.save(OUT)
print("saved workbook:", OUT, "sheets:", len(wb.sheetnames))
print(wb.sheetnames)

```

17. build_workbook3.py (SHA-256 daa6492dac37f633...; 245 lines)

```
# -*- coding: utf-8 -*-
import json
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill, Alignment
from openpyxl.utils import get_column_letter
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\ea9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
OUT = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
F = json.load(open(f"{SCR}/forestry_data.json", encoding="utf-8"))
YEARS=[2020,2021,2022,2023,2024]
GREEN="2F5233"; LIGHT="E7EEE3"; GOLD="C6902F"
hf=Font(name="Arial",size=10,bold=True,color="FFFFFF"); hfill=PatternFill("solid",start_color=GREEN)
bf=Font(name="Arial",size=10); bb=Font(name="Arial",size=10,bold=True)
INTFMT="#.##0"
wb=Workbook(); wb.remove(wb.active)

def block(ws,start,title,records,keyhdr,with_country=False):
    r=start
    ws.cell(row=r,column=1,value=title).font=Font(name="Arial",size=11,bold=True,color=GREEN); r+=1
    cols=[keyhdr]+(["Country"] if with_country else [])+[str(y) for y in YEARS]+["Total","Avg/yr"]
    for j,h in enumerate(cols,1):
        cc=ws.cell(row=r,column=j,value=h); cc.font=hf; cc.fill=hfill
        cc.alignment=Alignment(horizontal="left" if j==1 else "center")
    r+=1; first=r
    for i,rec in enumerate(records):
        col=1; ws.cell(row=r,column=col,value=rec["key"]).font=bf; col+=1
        if with_country: ws.cell(row=r,column=col,value=rec.get("country","")).font=bf; col+=1
        for y in YEARS:
            cc=ws.cell(row=r,column=col,value=rec[str(y)]); cc.font=bf; cc.number_format=INTFMT; col+=1
            cc=ws.cell(row=r,column=col,value=rec["total"]); cc.font=bb; cc.number_format=INTFMT; col+=1
            cc=ws.cell(row=r,column=col,value=round(rec["avg"],1)); cc.font=bf; cc.number_format="#.##0.0"; col+=1
            if i%2==1:
                for j in range(1,len(cols)+1): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
        r+=1
    ws.freeze_panes=ws.cell(row=first,column=2)
    ws.column_dimensions["A"].width=30 if with_country else 34
    if with_country: ws.column_dimensions["B"].width=18
    off=3 if with_country else 2
    for j in range(off,len(cols)+1): ws.column_dimensions[get_column_letter(j)].width=12
    return r+2

def multi(pfx,D,name,key,keyhdr,with_country=False):
    ws=wb.create_sheet(f"{pfx}{name}"); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By CITATIONS")]:
        r=block(ws,r,f"{name} - {lab}",D[key][m],keyhdr,with_country=with_country)
    return ws

def group(pfx,D,is_forestry,sm2=True):
    if not is_forestry:
        multi(pfx,D,"Field","field","Field (subfield match)")
    multi(pfx,D,"Continent","continent","Continent")
    multi(pfx,D,"World Region","region","World sub-region")
    if sm2:
        multi(pfx,D,"Orientation","orient","Quant/Qual orientation")
    if is_forestry and sm2:
        multi(pfx,D,"Dual field","dual","Forestry-only vs dual")
    multi(pfx,D,"US States","us_state","US state / category")
    wsCA=multi(pfx,D,"Canada & Australia","ca_prov","Canada province/territory")
    r=wsCA.max_row+3
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By CITATIONS")]:
        r=block(wsCA,r,f"Australia - {lab}",D["au_state"][m],"Australia state/territory")
    for m,lab in [("scientists","Scientists"),("publications","Publications"),("citations","Citations")]:
        ws=wb.create_sheet(f"{pfx}Country by {lab}")
        block(ws,1,f"All {len(D['country'][m])} countries by {lab.lower()} (per release year)",D["country"][m],"Country")
    for m,lab in [("scientists","Scientists"),("publications","Publications"),("citations","Citations")]:
        ws=wb.create_sheet(f"{pfx}University by {lab}")
        block(ws,1,f"All {len(D['university'][m])} institutions by {lab.lower()} (with country, per year)",D["university"][m],"Institution",with_country=True)
    # tree (only for combined; forestry taxonomy dropped with sm-subfield-2)
    if sm2:
        ws=wb.create_sheet(f"{pfx}Taxonomy Tree")
        for j,h in enumerate(["Domain (Field)","Field (Sub-Field)","Subfield (Sub-Sub-Field)","Records 2020-24","Avg/yr","Quantitative?","Target?"],1):
            c=ws.cell(row=1,column=j,value=h); c.font=hf; c.fill=hfill
            rows=sorted(D["tree"].items(), key=lambda kv:(kv[1]["domain"],kv[1]["field"],-kv[1]["total"])); r=2
```

```

for sf,info in rows:
    ws.cell(row=r,column=1,value=info["domain"]).font=bf
    ws.cell(row=r,column=2,value=info["field"]).font=bf
    ws.cell(row=r,column=3,value=sf).font=(bb if info["target"] else bf)
    ws.cell(row=r,column=4,value=info["total"]).number_format=INTFMT
    ws.cell(row=r,column=5,value=round(info["avg"],1)).number_format="#,##0.0"
    ws.cell(row=r,column=6,value=("Yes" if info["quant"] else "")).font=bf
    ws.cell(row=r,column=7,value=("Yes" if info["target"] else "")).font=bf
    if r%2==1:
        for j in range(1,8): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
    r+=1
ws.freeze_panes="A2"
for col,w in zip("ABCDEFGH",(26,32,34,14,10,13,10)): ws.column_dimensions[col].width=w
# map data
ws=wb.create_sheet(f"{pfx}Map data (avg)")
ws.cell(row=1,column=1,value="Average per release year (2020-2024). ISO3 = world country; name = US
state. ").font=Font(name="Arial",size=9,italic=True,color="555555")
for j,h in enumerate(["ISO3"]+[f"World {m}" for m in ["scientists","publications","citations"]],1):
    c=ws.cell(row=2,column=j,value=h); c.font=hf; c.fill=hfill
mw=D["map_world_avg"]; keys=sorted(mw["scientists"],key=lambda k:-mw["scientists"][k]); r=3
for k in keys:
    ws.cell(row=r,column=1,value=k).font=bf
    for j,m in enumerate(["scientists","publications","citations"],2):
        ws.cell(row=r,column=j,value=round(mw[m].get(k,0),1)).number_format="#,##0.0"
    r+=1
ws.column_dimensions["A"].width=10
for col in "BCD": ws.column_dimensions[col].width=16

# --- Index sheet ---
ws=wb.create_sheet("Index"); ws.column_dimensions["A"].width=34
for col in "BCDEFG": ws.column_dimensions[col].width=15
ws["A1"]="Stanford / Ioannidis Citation List - Forestry Representation (career-long, 2020-2024)";
ws["A1"].font=Font(name="Arial",size=13,bold=True,color=GREEN)
ws["A2"]="Sheets prefixed 'F ' = the 'Forestry plus Ecology' cohort (counts by division). 'F Intensity ratios' +
'F Ratios by ...' = the three intensity ratios (articles/author, citations/author, citations/article) backing the
report's analytical sections. 'F Group split' + 'F Grp ...' = the two groups (Forestry=criterion 1,
Ecology=criterion 2); 'SA-B '/'SA-C ' = single-author-only data for Appendices B (Forestry) and C (Ecology):
publications=single-author papers (nps), citations=cites to those papers (ncs).";
ws["A2"].font=Font(name="Arial",size=10,italic=True,color="555555")
r=4
for lab,D in [("Forestry plus Ecology cohort (criteria 1+2)",F)]:
    ws.cell(row=r,column=1,value=lab).font=Font(name="Arial",size=11,bold=True,color=GOLD); r+=1
    ws.cell(row=r,column=1,value="Metric").font=hf; ws.cell(row=r,column=1).fill=hfill
    for j,y in enumerate(YEARS,2): c=ws.cell(row=r,column=j,value=str(y)); c.font=hf; c.fill=hfill
    r+=1
    for m,label in [("scientists","Scientists (list entries)"),("publications","Publications (sum np)"),
("citations","Citations (sum nc)"]):
        ws.cell(row=r,column=1,value=label).font=bf
        for j,y in enumerate(YEARS,2):
            ws.cell(row=r,column=j,value=D["totals"][m][str(y)]).number_format=INTFMT
        r+=1
    r+=1
notes=[
    "Metric definitions: Scientists = ranked authors on the list. Publications = sum of each author's career paper
count (np). Citations = sum of each author's career citations (nc).",
    "Counting unit = author-year: each scientist counted once per release year they appear. No cross-year de-
duplication.",
    "Cohort = 'Forestry plus Ecology' = criterion 1 (sm-subfield-1 = Forestry) + criterion 2 (sm-subfield-1 =
Ecology AND sm-subfield-2 = Forestry). Both require Forestry; there is no ecology-without-forestry group.",
    "Quantitative-oriented = author whose primary/secondary subfield is a quantitative discipline: Statistics &
Probability; Mathematical/Computational (Applied/General/Numerical Math, Computation Theory, Software & other
ICT); Operations Research; all Engineering incl. Geological & Geomatics (GIS/mensuration) and Environmental
Engineering; AI & Image Processing (remote sensing); Bioinformatics.",
    "Universities grouped by name, labelled with modal country. Federal/Multi-site = distributed national agencies;
Unknown = unplaceable.",
    "Source: Ioannidis, doi:10.17632/btchxktzyw, career tables v3/v5/v6/v7/v8. Prepared 6 July 2026.",
]
for n in notes:
    ws.cell(row=r,column=1,value=n).font=Font(name="Arial",size=9,italic=True,color="555555")
    ws.merge_cells(start_row=r,start_column=1,end_row=r,end_column=7); r+=1

group("F ",F,is_forestry=True,sm2=False) # "Forestry plus Ecology" cohort geographic sheets

# --- per-group + multi-root taxonomy sheets from merged data ---
MM=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
def merged_to_recs(members,metric):
    k={"scientists":"s","publications":"p","citations":"c"}[metric]; out=[]
    for it in members:
        rec={"key":it["key"],"country":it.get("country",""),"total":sum(it[k]),"avg":round(sum(it[k])/5,1)}
        for i,y in enumerate(YEARS): rec[str(y)]=it[k][i]

```

```

        out.append(rec)
    out.sort(key=lambda r:-r["total"]); return out
# group split (Forestry group vs Ecology group) + each group's countries
ws=wb.create_sheet("F Group split"); r=1
for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By CITATIONS")]:
    r=block(ws,r,f"Group (Forestry vs Ecology) - {lab}",merged_to_recs(MM['forestry']['group'],m),"Group")
for gname,gkey in [{"F Grp Forestry countries","forestry_F"},("F Grp Ecology countries","forestry_E")]:
    ws=wb.create_sheet(gname); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By
CITATIONS")]:
        r=block(ws,r,f"{gname[6:]} - {lab}",merged_to_recs(MM[gkey]['country'],m),"Country")
for gname,gkey in [{"F Grp Forestry institutions","forestry_F"},("F Grp Ecology institutions","forestry_E")]:
    ws=wb.create_sheet(gname); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By PUBLICATIONS"),("citations","By
CITATIONS")]:
        r=block(ws,r,f"{gname[6:]} - {lab}",merged_to_recs(MM[gkey]
['university'],m),"Institution",with_country=True)
# multi-root taxonomy sheet (AQ -> AM -> AO)
ws=wb.create_sheet("F Taxonomy (AQ-AM-AO)"); QS=set(MM['taxonomy']['quant_set'])
for j,h in enumerate(["Field (sm-field / AQ)","Primary subfield (sm-subfield-1 / AM)","Secondary subfield (sm-
subfield-2 / AO)","Authors","Quant?"),1):
    c=ws.cell(row=1,column=j,value=h); c.font=hf; c.fill=hfill
roots=MM['taxonomy']['roots']; r=2
for aq in sorted(roots,key=lambda x:-sum(sum(a.values()) for a in roots[x].values())):
    for am in sorted(roots[aq],key=lambda x:-sum(roots[aq][x].values())):
        for ao in sorted(roots[aq][am],key=lambda x:-roots[aq][am][x]):
            ws.cell(row=r,column=1,value=aq).font=bf; ws.cell(row=r,column=2,value=am).font=bf
            ws.cell(row=r,column=3,value=ao).font=bf
            ws.cell(row=r,column=4,value=roots[aq][am][ao]).number_format=INTFMT
            ws.cell(row=r,column=5,value=("Yes" if ao in QS else "")).font=bf
            if r%2==1:
                for j in range(1,6): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
            r+=1
ws.freeze_panes="A2"
for col,w in zip("ABCDE",(30,26,32,12,10)): ws.column_dimensions[col].width=w

# --- intensity-ratio sheets (articles/author, citations/author, citations/article) ---
RJ=json.load(open(f"{SCR}/ratios.json",encoding="utf-8"))
def ratio_block(ws,start,title,members,keyhdr,with_country=False):
    r=start
    ws.cell(row=r,column=1,value=title).font=Font(name="Arial",size=11,bold=True,color=GREEN); r+=1
    cols=[keyhdr]+(["Country"] if with_country else
[["Scientists/yr","Articles/author","Citations/author","Citations/article"]
    for j,h in enumerate(cols,1):
        cc=ws.cell(row=r,column=j,value=h); cc.font=hf; cc.fill=hfill
        cc.alignment=Alignment(horizontal="left" if j==1 else "center")
    r+=1; first=r
    for i,m in enumerate(members):
        col=1; ws.cell(row=r,column=col,value=m["key"]).font=bf; col+=1
        if with_country: ws.cell(row=r,column=col,value=m.get("country","")).font=bf; col+=1
        ws.cell(row=r,column=col,value=round(m["sci_yr"],1)).number_format="#,##0.0"; col+=1
        ws.cell(row=r,column=col,value=round(m["art_per_auth"],1)).number_format="#,##0.0"; col+=1
        ws.cell(row=r,column=col,value=round(m["cit_per_auth"],0)).number_format=INTFMT; col+=1
        ws.cell(row=r,column=col,value=round(m["cit_per_art"],1)).number_format="#,##0.0"; col+=1
        if i%2==1:
            for j in range(1,len(cols)+1): ws.cell(row=r,column=j).fill=PatternFill("solid",start_color=LIGHT)
        r+=1
    ws.freeze_panes=ws.cell(row=first,column=2)
    ws.column_dimensions["A"].width=30 if with_country else 34
    if with_country: ws.column_dimensions["B"].width=18
    off=3 if with_country else 2
    for j in range(off,len(cols)+1): ws.column_dimensions[get_column_letter(j)].width=15
    return r+2
# summary sheet (cohort + groups + the quantity/quality correlations)
ws=wb.create_sheet("F Intensity ratios"); r=1
ws.cell(row=r,column=1,value="Intensity ratios - five-year values, author-year basis. Ratio = sum of numerator /
sum of denominator over 2020-2024.").font=Font(name="Arial",size=9,italic=True,color="555555"); r+=2
co=RJ["cohort"]["overall"]; gg=RJ["group"]
summ=[{"key":"Whole cohort (Forestry + Ecology)","co"},
{"key":"Forestry group (criterion 1)","ggg["Forestry"]},
{"key":"Ecology group (criterion 2)","ggg["Ecology"]}
r=ratio_block(ws,r,"Cohort and groups",summ,"Population")
cq=RJ["corr"]["country"]; iq=RJ["corr"]["university"]
for line in [f"Quantity vs per-paper quality (articles/author vs citations/article): countries n={cq['n']} (>=
{cq['min_sci_yr']:.0f} sci/yr) Pearson r={cq['pearson']:.2f}, Spearman {cq['spearman']:.2f}; institutions
n={iq['n']} r={iq['pearson']:.2f}.",
f"Quantity vs total impact (articles/author vs citations/author): countries Pearson
r={cq['pearson_q_vs_totimpact']:.2f}.",
"Interpretation: output and per-paper quality are largely independent; output tracks total citations
per author."]:

```

```

        ws.cell(row=r,column=1,value=line).font=Font(name="Arial",size=9,italic=True,color="555555")
        ws.merge_cells(start_row=r,start_column=1,end_row=r,end_column=6); r+=1
r+=1
r=ratio_block(ws,r,"By continent",RJ["continent"],"Continent")
r=ratio_block(ws,r,"By world sub-region",RJ["region"],"World sub-region")
ws2=wb.create_sheet("F Ratios by country")
ratio_block(ws2,1,f"All {len(RJ['country'])} countries - intensity ratios (ordered by
scientists)",RJ["country"],"Country")
ws3=wb.create_sheet("F Ratios by institution")
ratio_block(ws3,1,f"All {len(RJ['university'])} institutions - intensity ratios (ordered by
scientists)",RJ["university"],"Institution",with_country=True)
ws4=wb.create_sheet("F Ratios by US state")
ratio_block(ws4,1,"U.S. states / categories - intensity ratios (ordered by scientists)",RJ["us_state"],"U.S.
state / category")

# --- single-author appendix sheets (App B = Forestry group, App C = Ecology group) ---
# publications = single-author papers (nps); citations = citations to those papers (ncs); scientists have >=1
single-author paper.
for tag,fn,gl in [("SA-B Forestry","forestry_B_data.json","Forestry group (criterion 1), single-author"),
                  ("SA-C Ecology","forestry_C_data.json","Ecology group (criterion 2), single-author")]:
    D=json.load(open(f"{SCR}/{fn}",encoding="utf-8"))
    totsci=sum(D["totals"] ["scientists"] [str(y)] for y in YEARS)
    def _note(ws,r,dim):
        miss=totsci-sum(rec["total"] for rec in D[dim] ["scientists"])
        if miss:
            ws.cell(row=r,column=1,value=f"Note: {miss} author-year record(s) lack a {'country code' if
dim=='country' else 'institution name'} in the source data; "
            "they are included in the group totals but cannot be listed here, so column sums are
marginally below the totals."
            ).font=Font(name="Arial",size=9,italic=True,color="8A5A00")
        return r+2
    ws=wb.create_sheet(f"{tag} countries"); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By SINGLE-AUTHOR PAPERS"),("citations","By
CITES TO SA PAPERS")]:
        r=block(ws,r,f"{gl} - countries {lab}",D["country"] [m],"Country")
        _note(ws,r,"country")
    ws=wb.create_sheet(f"{tag} institutions"); r=1
    for m,lab in [("scientists","By SCIENTISTS"),("publications","By SINGLE-AUTHOR PAPERS"),("citations","By
CITES TO SA PAPERS")]:
        r=block(ws,r,f"{gl} - institutions {lab}",D["university"] [m],"Institution",with_country=True)
        _note(ws,r,"university")

wb.save(OUT)
print("saved workbook:",OUT,"| sheets:",len(wb.sheetnames))
maxlen=max(len(s) for s in wb.sheetnames); print("max sheet-name length:",maxlen)
print(wb.sheetnames)

```

18. cc_defs.py (SHA-256 1db009f64c0b7d5b...; 48 lines)

```

# country code (ISO alpha-3, lowercase) -> (name, continent, sub-region)
# NOTE: sub-regions follow UN M49 EXCEPT a dedicated "Central Europe" (CE-9: Poland, Germany, Austria,
# Switzerland, Czechia, Slovakia, Hungary, Slovenia, Liechtenstein) is broken out of Western/Eastern/Southern
# Europe per the user's Central-Europe definition (explicitly NOT Eastern Europe).
CC = {
    "are": ("United Arab Emirates", "Asia", "Western Asia"), "arg": ("Argentina", "South America", "South America"),
    "aus": ("Australia", "Oceania", "Australia & New Zealand"), "aut": ("Austria", "Europe", "Central Europe"),
    "aze": ("Azerbaijan", "Asia", "Western Asia"), "bel": ("Belgium", "Europe", "Western Europe"),
    "bgd": ("Bangladesh", "Asia", "Southern Asia"), "bhr": ("Bahrain", "Asia", "Western Asia"),
    "bra": ("Brazil", "South America", "South America"), "brn": ("Brunei", "Asia", "South-eastern Asia"),
    "bwa": ("Botswana", "Africa", "Southern Africa"), "can": ("Canada", "North America", "Northern America"),
    "che": ("Switzerland", "Europe", "Central Europe"), "chl": ("Chile", "South America", "South America"),
    "chn": ("China", "Asia", "Eastern Asia"), "col": ("Colombia", "South America", "South America"),
    "cri": ("Costa Rica", "North America", "Central America"), "cuw": ("Curacao", "North America", "Caribbean"),
    "cze": ("Czechia", "Europe", "Central Europe"), "deu": ("Germany", "Europe", "Central Europe"),
    "dnk": ("Denmark", "Europe", "Northern Europe"), "dza": ("Algeria", "Africa", "Northern Africa"),
    "ecu": ("Ecuador", "South America", "South America"), "egy": ("Egypt", "Africa", "Northern Africa"),
    "esp": ("Spain", "Europe", "Southern Europe"), "est": ("Estonia", "Europe", "Northern Europe"),
    "eth": ("Ethiopia", "Africa", "Eastern Africa"), "fin": ("Finland", "Europe", "Northern Europe"),
    "fra": ("France", "Europe", "Western Europe"), "gbr": ("United Kingdom", "Europe", "Northern Europe"),
    "geo": ("Georgia", "Asia", "Western Asia"), "grc": ("Greece", "Europe", "Southern Europe"),
    "grl": ("Greenland", "North America", "Northern America"), "hkg": ("Hong Kong", "Asia", "Eastern Asia"),
    "hun": ("Hungary", "Europe", "Central Europe"), "idn": ("Indonesia", "Asia", "South-eastern Asia"),
    "ind": ("India", "Asia", "Southern Asia"), "irl": ("Ireland", "Europe", "Northern Europe"),
    "irn": ("Iran", "Asia", "Southern Asia"), "isr": ("Israel", "Asia", "Western Asia"),
    "ita": ("Italy", "Europe", "Southern Europe"), "jor": ("Jordan", "Asia", "Western Asia"),
    "jpn": ("Japan", "Asia", "Eastern Asia"), "ken": ("Kenya", "Africa", "Eastern Africa"),
    "kor": ("South Korea", "Asia", "Eastern Asia"), "kwt": ("Kuwait", "Asia", "Western Asia"),
    "ltu": ("Lithuania", "Europe", "Northern Europe"), "lva": ("Latvia", "Europe", "Northern Europe"),

```

```

"mex": ("Mexico", "North America", "Central America"), "mwi": ("Malawi", "Africa", "Eastern Africa"),
"mys": ("Malaysia", "Asia", "South-eastern Asia"), "nga": ("Nigeria", "Africa", "Western Africa"),
"nld": ("Netherlands", "Europe", "Western Europe"), "nor": ("Norway", "Europe", "Northern Europe"),
"npl": ("Nepal", "Asia", "Southern Asia"), "nzl": ("New Zealand", "Oceania", "Australia & New Zealand"),
"pak": ("Pakistan", "Asia", "Southern Asia"), "pan": ("Panama", "North America", "Central America"),
"per": ("Peru", "South America", "South America"), "phl": ("Philippines", "Asia", "South-eastern Asia"),
"pol": ("Poland", "Europe", "Central Europe"), "pri": ("Puerto Rico", "North America", "Caribbean"),
"prt": ("Portugal", "Europe", "Southern Europe"), "qat": ("Qatar", "Asia", "Western Asia"),
"rou": ("Romania", "Europe", "Eastern Europe"), "rus": ("Russia", "Europe", "Eastern Europe"),
"sau": ("Saudi Arabia", "Asia", "Western Asia"), "sgp": ("Singapore", "Asia", "South-eastern Asia"),
"srb": ("Serbia", "Europe", "Southern Europe"), "svk": ("Slovakia", "Europe", "Central Europe"),
"svn": ("Slovenia", "Europe", "Central Europe"), "swe": ("Sweden", "Europe", "Northern Europe"),
"tha": ("Thailand", "Asia", "South-eastern Asia"), "tto": ("Trinidad and Tobago", "North America", "Caribbean"),
"tur": ("Turkey", "Asia", "Western Asia"), "tw": ("Taiwan", "Asia", "Eastern Asia"),
"uga": ("Uganda", "Africa", "Eastern Africa"), "ukr": ("Ukraine", "Europe", "Eastern Europe"),
"ury": ("Uruguay", "South America", "South America"), "usa": ("United States", "North America", "Northern America"),
"ven": ("Venezuela", "South America", "South America"), "zaf": ("South Africa", "Africa", "Southern Africa"),
"zmb": ("Zambia", "Africa", "Eastern Africa"),
}

```

19. chk_region.py (SHA-256 b5b1d6e1a6b6e28c...; 11 lines)

```

import json
M=json.load(open("merged_data.json",encoding="utf-8"))
print("=== World sub-region (Forestry plus Ecology, 3886), scientists ===")
for r in M["forestry"]["region"]:
    print(f"    {r['key']}<26} total={sum(r['s']):>5}    2024={r['s'][-1]}")
# which countries are Central Europe
from cc_defs import CC
ce=[v[0] for k,v in CC.items() if v[2]=="Central Europe"]
print("\nCentral Europe countries in map:", sorted(ce))
fc={r["key"]:sum(r["s"]) for r in M["forestry"]["country"]}
print("CE countries present in cohort (scientists all-yr):", {c:fc[c] for c in sorted(ce) if c in fc})

```

20. compute_auth6.py (SHA-256 5bd3ac08e9d4b369...; 135 lines)

```

# -*- coding: utf-8 -*-
# NINE-number scheme for every division and named individual.
# Authorship classes: solo (single, nps/ncs), senior (single+first, cpsf/ncsf), all (np/nc).
# For each class, three measures (author = listed author-year record; sums over 5 releases):
#     pubs/author = Epub_class / Eauthors ; cites/author = Ecite_class / Eauthors ; cites/pub = Ecite_class /
# Epub_class.
# The Data section shows only the RAW COUNTS (scientists, solo/senior/all publication counts); every ratio is a
# derivation reported in Results. No raw citation SUM is displayed anywhere.
import json, numpy as np, pandas as pd
from cc_defs import CC
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]; F=["np","nc","nps","ncs","cpsf","ncsf"]

raw=pd.read_excel(RAW,sheet_name="Data")
for c in F: raw[c]=pd.to_numeric(raw[c],errors="coerce").fillna(0)
raw["country"]=raw["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
raw["continent"]=raw["cntry"].map(lambda c: CC[c][1] if c in CC else "(unknown)")
raw["region"]=raw["cntry"].map(lambda c: CC[c][2] if c in CC else "(unknown)")
subdiv={int(k):v for k,v in json.load(open(f"{SCR}/inst_subdiv.json",encoding="utf-8")).items()}
idx=pd.read_csv(f"{SCR}/inst_index.csv"); idx["sd"]=idx["id"].map(subdiv).fillna("Unknown")
look={r.country,r.inst:r.sd for r in idx.itertuples()}
CN={"usa":"United States","can":"Canada","aus":"Australia"}
raw["subdivision"]=raw.apply(lambda r: look.get((CN.get(r["cntry"])),r["inst_name"],"Unknown") if r["cntry"] in CN else None,axis=1)
s1=raw["sm-subfield-1"].astype(str).str.strip(); s2=raw["sm-subfield-2"].astype(str).str.strip()
gF=(s1=="Forestry"); gE=(s1=="Ecology") & (s2=="Forestry")
df=raw[gF|gE].copy(); df["group"]=np.where(df["sm-subfield-1"].astype(str).str.strip()=="Forestry","Forestry","Ecology")

def nine(frame):
    S=lambda c: float(frame[c].sum())
    n=len(frame)
    sp,snc=S("nps"),S("ncs"); ep,enc=S("cpsf"),S("ncsf"); ap,anc=S("np"),S("nc")
    r=lambda num,den:(num/den if den else 0.0)
    return {"sci_yr":n/5.0,"n":n,
            "solo_p":sp,"sen_p":ep,"all_p":ap,
            "solo_ppa":r(sp,n),"sen_ppa":r(snc,n),"solo_cpp":r(snc,sp),
            "sen_ppa":r(ep,n),"sen_cpa":r(enc,n),"sen_cpp":r(enc,ep),
            "all_ppa":r(ap,n),"all_cpa":r(anc,n),"all_cpp":r(anc,ap)}
# raw pub counts (Data)
# solo derivations (Results)
# senior derivations
# all derivations

```

```

def members(frame,keycol,with_country=False):
    out=[]
    ctry=frame.groupby(keycol)["country"].agg(lambda s:s.value_counts().index[0]) if with_country else None
    for k,g in frame.groupby(keycol):
        m={"key":k, **nine(g)}
        if with_country: m["country"]=ctry.get(k,"")
        out.append(m)
    out.sort(key=lambda m:-m["all_p"])
    return out

DIV={}
DIV["continent"]=members(df[df.continent!="(unknown)"],"continent")
DIV["region"]=members(df[df.region!="(unknown)"],"region")
DIV["country"]=members(df[df.country!="(unknown)"],"country")
DIV["us_state"]=members(df[df.cntry=="usa"],"subdivision")
DIV["university"]=members(df[df["inst_name"].notna()], "inst_name",with_country=True)
DIV["group"]=members(df,"group")

def totals_block(frame):
    b=nine(frame); b["key"]="Whole cohort"
    keys=["solo_p","sen_p","all_p","solo_ppa","solo_cpa","solo_cpp","sen_ppa","sen_cpa","sen_cpp",
        "all_ppa","all_cpa","all_cpp","n"]
    by={k:[] for k in keys}
    for y in YEARS:
        s=nine(frame[frame.year==y])
        for k in by: by[k].append(s[k])
    return b,by
COH,COH_by=totals_block(df)
GF=nine(df[df.group=="Forestry"]); GF["key"]="Forestry group"
GE=nine(df[df.group=="Ecology"]); GE["key"]="Ecology group"

def group_pack(frame):
    b,by=totals_block(frame)
    return {"overall":b,"by":by,
        "continent":members(frame[frame.continent!="(unknown)"],"continent"),
        "country":members(frame[frame.country!="(unknown)"],"country"),
        "university":members(frame[frame["inst_name"].notna()], "inst_name",with_country=True)}
GPACK={"Forestry":group_pack(df[df.group=="Forestry"]), "Ecology":group_pack(df[df.group=="Ecology"])}

# ---- named individuals (latest release row) ----
def person_from(frame_full,namekey,label=None,note=None):
    h=frame_full[frame_full["authfull"].astype(str).str.contains(namekey,case=False,na=False)].copy()
    if len(h)==0: return None
    h["year"]=h["year"].astype(int) if "year" in h else 2024
    r=h.sort_values("year").iloc[-1] if "year" in h else h.iloc[0]
    def v(c): return float(pd.to_numeric(pd.Series([r[c]]),errors="coerce").fillna(0).iloc[0])
    sp,snc=v("nps"); v("ncs"); ep,enc=v("cpsf"); v("ncsf"); ap,anc=v("np"); v("nc")
    rr=lambda num,den:(num/den if den else 0.0)
    return {"name":label or str(r["authfull"]), "note":note,
        "solo_p":sp, "sen_p":ep, "all_p":ap,
        "solo_ppa":sp, "solo_cpa":snc, "solo_cpp":rr(snc,sp), # individual: /author == own count
        "sen_ppa":ep, "sen_cpa":enc, "sen_cpp":rr(enc,ep),
        "all_ppa":ap, "all_cpa":anc, "all_cpp":rr(anc,ap),
        "sf1":str(r["sm-subfield-1"]), "sf2":str(r["sm-subfield-2"])}
cies=person_from(df,"Cieszewski"); bett=person_from(df,"Bettinger")
pete=person_from(df[df.group=="Ecology"],"Peterson, Chris") # 3rd UGA forester (criterion 2)

# external individuals + hidden-Warnell from the FULL 2024 career file
full=pd.read_excel(f"{SCR}/career_2024.xlsx",sheet_name="Data")
import re
ren={c:("nc" if re.fullmatch(r"nc\d+",str(c)) and "(ns)" not in str(c)) else
    "np" if re.fullmatch(r"np\d+",str(c)) else str(c)} for c in full.columns}
full=full.rename(columns=ren)
for c in F: full[c]=pd.to_numeric(full[c],errors="coerce").fillna(0)
full["year"]=2024
KAH=person_from(full,"Kahneman","Kahneman, Daniel")
WARN={k:person_from(full,k) for k in ["Teskey, Robert","Markewitz, Daniel","Grossman, Gary","Helfman,
Gene","Yabsley, Michael","Woosnam, Kyle"]}

# ---- map data ----
# per-country (ISO3 = uppercased source code) nine layers, ≥5 author-years over 2020-2024
LAYERS=["solo_ppa","solo_cpa","solo_cpp","sen_ppa","sen_cpa","sen_cpp","all_ppa","all_cpa","all_cpp"]
mc={L:[] for L in LAYERS}
for code,g in df[df.cntry.notna()].groupby("cntry"):
    if len(g)<5: continue
    s=nine(g); iso=str(code).upper()
    for L in LAYERS: mc[L][iso]=s[L]

```

```

# per-US-state solo/senior/all publication COUNTS (Data section, counts only)
mus={"solo_p":{}, "sen_p":{}, "all_p":{}}
for st, g in df[df.cntry=="usa"].groupby("subdivision"):
    if st in ("Unknown", None, "Federal/Multi-site"): continue
    s=nine(g)
    for k in mus: mus[k][st]=s[k]

OUT={"years":YEARS, "div":DIV, "cohort":COH, "cohort_by":COH_by, "group":{"Forestry":GF, "Ecology":GE},
     "group_pack":GPack, "map_country":mc, "map_us":mus,
     "cies":cies, "bett":bett, "peterson":pete, "kahneman":KAH, "warnell":WARN}
json.dump(OUT, open(f"{SCR}/auth6.json", "w", encoding="utf-8"), default=float)

def show(m, lab):
    print(f"  {lab:<24} solo {m['solo_ppa']:>6.1f}p/au {m['solo_cpa']:>7.0f}c/au {m['solo_cpp']:>5.0f}c/p | "
          f"sen {m['sen_ppa']:>6.1f} {m['sen_cpa']:>7.0f} {m['sen_cpp']:>5.0f} | all {m['all_ppa']:>6.1f} "
          f"{m['all_cpa']:>7.0f} {m['all_cpp']:>5.0f}")
    print("COHORT / GROUPS:"); [show(m, m['key']) for m in (COH, GF, GE)]
    print("\nCONTINENTS:"); [show(m, m['key']) for m in DIV['continent']]
    print("\nREFERENCE INDIVIDUALS:")
    for m in [cies, bett, pete, KAH]:
        if m: show(m, m['name'])
    print("\nsaved auth6.json")

```

21. compute_authorship.py (SHA-256 ab49e9b3a5b18013...; 107 lines)

```

# -*- coding: utf-8 -*-
# Authorship-position analysis for the forestry cohort.
# Dataset fields (all-citations basis): np/nc = all; nps/ncs = single; cpsf/ncsf = single+first;
# npsfl/ncsfl = single+first+last. Nested sets -> exact position decomposition per author:
#   single = nps ; first(multi) = cpsf-nps ; last(senior) = npsfl-cpsf ; middle = np-npsfl.
import json, numpy as np, pandas as pd
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020, 2021, 2022, 2023, 2024]
df=pd.read_excel(RAW, sheet_name="Data")
need=["np", "nc", "nps", "ncs", "cpsf", "ncsf", "npsfl", "ncsfl"]
print("cols present:", [c for c in need if c in df.columns])
for c in need: df[c]=pd.to_numeric(df[c], errors="coerce").fillna(0)
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
gF=(s1=="Forestry"); gE=(s1=="Ecology") & (s2=="Forestry")
df=df[gF|gE].copy(); df["group"]=np.where(df["sm-subfield-1"].astype(str).str.strip()=="Forestry", "Forestry", "Ecology")
print("cohort author-years:", len(df), "| per year:", {y: int((df.year==y).sum()) for y in YEARS})

# position decomposition (papers p_*, citations c_*). Last-authored is NOT part of this research:
# collapse everything beyond the lead (single-or-first) authorship into one "other co-authored" class.
df["p_single"]=df["nps"]; df["c_single"]=df["ncs"]
df["p_first"]=df["cpsf"]-df["nps"]; df["c_first"]=df["ncsf"]-df["ncs"]
df["p_other"]=df["np"]-df["cpsf"]; df["c_other"]=df["nc"]-df["ncsf"] # = last + middle (co-authored, not lead)
# sanity: non-negative, and sum to totals
neg=((df[["p_first", "p_other", "c_first", "c_other"]]<0).sum())
print("negative cells (should be ~0):", dict(neg))
psum=(df[["p_single"]+df["p_first"]+df["p_other"]-df["np"]]).abs().sum()
csum=(df[["c_single"]+df["c_first"]+df["c_other"]-df["nc"]]).abs().sum()
print("decomposition residual papers/cites (should be 0):", int(psum), int(csum))

POS=["single", "first", "other"]
def agg_block(frame):
    out={}
    for pos in POS:
        out[pos]={ "p":int(frame[f"p_{pos}"].sum()), "c":int(frame[f"c_{pos}"].sum()) }
    out["all"]={ "p":int(frame["np"].sum()), "c":int(frame["nc"].sum()) }
    # author-years (scientists) with >=1 paper in each position
    for pos in POS: out[pos]["nauth"]=int((frame[f"p_{pos}"]>0).sum())
    return out
COH=agg_block(df); GF=agg_block(df[df.group=="Forestry"]); GE=agg_block(df[df.group=="Ecology"])

def cpp(block, pos): return (block[pos]["c"]/block[pos]["p"]) if block[pos]["p"]>0 else 0.0
def share_c(block, pos): return 100*block[pos]["c"]/block["all"]["c"]
def share_p(block, pos): return 100*block[pos]["p"]/block["all"]["p"]

# ----- RANK-SHIFT ANALYSIS on the 2024 release -----
d24=df[df.year==2024].copy()
def avg_rank(x): # descending rank (1 = highest), average ties
    x=np.asarray(x, float); order=np.argsort(-x); r=np.empty(len(x)); r[order]=np.arange(1, len(x)+1)

```

```

# average ties
import collections
idx=collections.defaultdict(list)
for i,v in enumerate(x): idx[v].append(i)
for v,ii in idx.items():
    if len(ii)>1:
        r[ii]=np.mean([r[i] for i in ii])
return r
def spearman(a,b):
    ra,rb=avg_rank(a),avg_rank(b); return float(np.corrcoef(ra,rb)[0,1])
# ranking bases (by citations): lead(ncsf = single+first) and single(ncs) only – last-authored dropped.
bases={"lead": single+first (ncsf)":d24["ncsf"].values,
       "single (ncs)":d24["ncs"].values,
       "first-authored cites":d24["c_first"].values}
nc=d24["nc"].values
corr={k:spearman(nc,v) for k,v in bases.items()}
# top-25 overlap: of top-25 by nc, how many are in top-25 by each basis
def top_set(x,n=25):
    return set(np.argsort(-np.asarray(x,float))[:n])
t_all=top_set(nc)
overlap={k:len(t_all & top_set(v)) for k,v in bases.items()}
# biggest rank climbers/fallers between all vs single ranking
r_all=avg_rank(nc); r_single=avg_rank(d24["ncs"].values)
d24=d24.assign(r_all=r_all,r_single=r_single,shift=r_all-r_single) # +ve => ranks HIGHER on single (climbs)
N=len(d24)

# reference scientists' decomposition (2024)
def person_decomp(namekey):
    h=d24[d24["authfull"].astype(str).str.contains(namekey,case=False,na=False)]
    if len(h)==0: return None
    r=h.iloc[0]
    return {"name":str(r["authfull"]), "np":int(r["np"]), "nc":int(r["nc"]),
           "single":{"p":int(r["p_single"]), "c":int(r["c_single"])},
           "first":{"p":int(r["p_first"]), "c":int(r["c_first"])},
           "other":{"p":int(r["p_other"]), "c":int(r["c_other"])},
           "rank_all":int(r["r_all"]), "rank_single":int(r["r_single"])
    }
cies=person_decomp("Cieszewski"); bett=person_decomp("Bettinger")

OUT={"blocks":{"cohort":COH,"Forestry":GF,"Ecology":GE},
     "corr":corr,"overlap":overlap,"n2024":N,
     "cies":cies,"bett":bett,
     "cpp":{"g:{pos:cpp(b,pos) for pos in POS+["all"]} for g,b in [("cohort",COH),("Forestry",GF),
     ("Ecology",GE)]},
     "share_c":{"g:{pos:share_c(b,pos) for pos in POS} for g,b in [("cohort",COH),("Forestry",GF),
     ("Ecology",GE)]},
     "share_p":{"g:{pos:share_p(b,pos) for pos in POS} for g,b in [("cohort",COH),("Forestry",GF),
     ("Ecology",GE)]}}
json.dump(OUT,open(f"{SCR}/authorship.json","w",encoding="utf-8"))

print("\n=== COHORT citation share by authorship position (5-yr) ===")
for pos in POS: print(f" {pos:<7}: {share_c(COH,pos):5.1f}% of citations | {share_p(COH,pos):5.1f}% of papers |
{cpp(COH,pos):6.1f} cites/paper")
print(f" all : 100.0% | {COH['all']['c']:}, cites, {COH['all']['p']:}, papers | {cpp(COH,'all'):.1f}
cites/paper")
print("\n=== group cites/paper by position ===")
for g,b in [("Forestry",GF),("Ecology",GE)]:
    print(f" {g:<9}: "+ " ".join(f"{pos} {cpp(b,pos):.0f}" for pos in POS+["all"]))
print("\n=== RANK CORRELATION (Spearman) of the undiscriminated ranking (by nc) vs each basis, 2024 (n=%d)
===%N)
for k in bases: print(f" nc vs {k:<32}: rho={corr[k]:+.2f} | top-25 overlap {overlap[k]}/25")
print("\n=== reference scientists (2024) authorship decomposition ===")
for p in [cies,bett]:
    if p: print(f" {p['name']}: np={p['np']} nc={p['nc']:}, | single {p['single']['p']}p/{p['single']['c']}c |
first {p['first']['p']}p/{p['first']['c']}c | other {p['other']['p']}p/{p['other']['c']}c | rank_all
{p['rank_all']} vs rank_single {p['rank_single']}")
print("\nsaved authorship.json")

```

22. compute_integrity.py (SHA-256 d43412ebd27324b7...; 77 lines)

```

# -*- coding: utf-8 -*-
# Robustness / integrity / distribution diagnostics for the cohort (2024 release), following the
# "beyond the headline number" bibliometric guidance: self-citation, citation concentration (cprat),
# retraction exposure, visibility efficiency (% papers cited), coauthorship-adjusted h (hm), and
# citation inequality (Gini, top-decile share). Writes integrity.json + a 2-panel figure ff_integrity.png.
import json, numpy as np, pandas as pd
import matplotlib; matplotlib.use("Agg")
import matplotlib.pyplot as plt

```

```

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
GREEN="#2F5233"; GOLD="#C6902F"; TEAL="#2C6E9B"; ECOL="#3E8E7E"
d=pd.read_excel(RAW,sheet_name="Data")
s1=d["sm-subfield-1"].astype(str).str.strip(); s2=d["sm-subfield-2"].astype(str).str.strip()
d=d[(s1=="Forestry") | ((s1=="Ecology") & (s2=="Forestry"))].copy()
d=d[d.year==2024].copy()
for c in ["np","nc","h","hm","self%","npciting","cprat","np_cited","np_rw","nc_to_rw","nc_rw","firstyr"]:
    d[c]=pd.to_numeric(d[c],errors="coerce")
n=len(d)
d["pctcited"]=100*d["np_cited"]/d["np"]
d["hmratio"]=d["hm"]/d["h"]
d["age"]=2024-d["firstyr"]
def gini(x):
    x=np.sort(np.asarray(x,float)); k=len(x)
    return float((2*np.arange(1,k+1)-k-1).dot(x)/(k*x.sum()))
nc=np.sort(d["nc"].dropna().values.astype(float))
G=gini(nc); top10=float(nc[int(.9*n)].sum()/nc.sum()); top1=float(nc[int(.99*n)].sum()/nc.sum())
OUT={"n":n,
     "self_med":float(d["self%"].median()*100),"self_p90":float(d["self%"].quantile(.9)*100),
     "self_max":float(d["self%"].max()*100),"self_gt25":int((d["self%"]>0.25).sum()),
     "cprat_med":float(d["cprat"].median()),"cprat_max":float(d["cprat"].max()),"cprat_gt2":int((d["cprat"]>2).sum())
     ,
     "rw_auth":int((d["np_rw"]>0).sum()),"rw_papers":int(d["np_rw"].fillna(0).sum()),"rw_citing":int((d["nc_to_rw"]>0).sum()),
     "pctcited_med":float(d["pctcited"].median()),
     "h_med":float(d["h"].median()),"hm_med":float(d["hm"].median()),"hmratio_med":float(d["hmratio"].median()),
     "gini":G,"top10":top10*100,"top1":top1*100,
     "age_med":float(d["age"].median()),"age_early":float(100*(d["age"]<20).mean())}
def ref(nm):
    r=d[d["authfull"].astype(str).str.contains(nm,case=False,na=False)]
    if not len(r): return None
    r=r.iloc[0]
    return {"h":float(r["h"]),"hm":float(r["hm"]),"self":float(r["self%"]*100),
           "cprat":float(r["cprat"]),"pctcited":float(100*r["np_cited"]/r["np"]),"age":float(2024-r["firstyr"])}
OUT["refs"]={"Cieszewski":ref("Cieszewski"),"Bettinger":ref("Bettinger"),"Peterson":ref("Peterson, Chris")}
json.dump(OUT,open(f"{SCR}/integrity.json","w"),default=float)

# ---- figure: (a) Lorenz curve of citations, (b) h vs hm scatter ----
plt.rcParams.update({"font.family":"Arial","font.size":12})
fig,axs=plt.subplots(1,2,figsize=(11.0,4.7))
# Lorenz
cum=np.concatenate([ [0],np.cumsum(nc)/nc.sum()]); p=np.linspace(0,1,len(cum))
axs[0].plot([0,1],[0,1],"--",color="#888",lw=1.1)
axs[0].plot(p,cum,color=GREEN,lw=2.6)
axs[0].fill_between(p,cum,p,color=GREEN,alpha=0.08)
axs[0].set_xlabel("Cumulative share of scientists (poorest-richest)",fontsize=11.5)
axs[0].set_ylabel("Cumulative share of citations",fontsize=11.5)
axs[0].set_title(f"Citation concentration - Gini {G:.2f}",color=GREEN,fontweight="bold",fontsize=13,pad=6)
axs[0].annotate(f"top 10% hold {top10*100:.0f}% of citations",(0.03,0.92),xycoords="axes fraction",fontsize=10.5,color="#444",style="italic")
axs[0].set_xlim(0,1); axs[0].set_ylim(0,1); axs[0].spines[["top","right"]].set_visible(False)
# h vs hm
h=d["h"].values; hm=d["hm"].values
axs[1].scatter(h,hm,s=14,c="#4C9F70",alpha=0.45,edgecolor="none")
hi=np.nanmax(h)
axs[1].plot([0,hi],[0,hi],"--",color="#888",lw=1.0,label="h = hm (no coauthors)")
axs[1].plot([0,hi],[0,0.5*hi],":",color="#A63D40",lw=1.4,label="hm = ½ h (cohort median)")
rc={"Cieszewski":GOLD,"Bettinger":TEAL,"Peterson":ECOL}
for nm,col in rc.items():
    r=OUT["refs"][nm]
    if r: axs[1].scatter([r["h"]],[r["hm"]],s=130,c=col,edgecolor="#333",linewidth=1.0,zorder=5,
                        marker={"Cieszewski":"*", "Bettinger":"D", "Peterson":"o"}[nm],label=nm)
axs[1].set_xlabel("h-index",fontsize=11.5); axs[1].set_ylabel("hm-index (coauthorship-adjusted)",fontsize=11.5)
axs[1].set_title("Coauthorship roughly halves the h-index",color=GREEN,fontweight="bold",fontsize=13,pad=6)
axs[1].legend(fontsize=9,loc="upper left",frameon=False); axs[1].spines[["top","right"]].set_visible(False)
fig.tight_layout(); fig.savefig(f"{SCR}/ff_integrity.png",dpi=200,bbox_inches="tight"); plt.close(fig)

print("saved integrity.json + ff_integrity.png")
for k,v in OUT.items():
    if k!="refs": print(f" {k}: {v}")
print(" refs:",OUT["refs"])

```

23. compute_refs.py (SHA-256 3c4baf9437e14cb...; 67 lines)

```

# -*- coding: utf-8 -*-
# Reference-scientist comparison: Cieszewski (author) & Bettinger, both UGA, vs cohort/US/UGA benchmarks.

```

```

import pandas as pd, json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
df=pd.read_excel(RAW,sheet_name="Data")
R=json.load(open(f"{SCR}/ratios.json",encoding="utf-8"))

def person(namekey):
    h=df[df["authfull"].astype(str).str.contains(namekey,case=False,na=False)].copy()
    by=(int(r["year"]):r for _,r in h.iterrows())
    def g(y,c):
        return (None if y not in by else by[y][c])
    out={"name":str(by[2024]["authfull"]),
        "inst":str(by[2024]["inst_name"]), "cntry":str(by[2024]["cntry"]),
        "sf1":str(by[2024]["sm-subfield-1"]), "sf2":str(by[2024]["sm-subfield-2"]),
        "np":{y:int(g(y,"np")) for y in YEARS if y in by},
        "nc":{y:int(g(y,"nc")) for y in YEARS if y in by}}
    # subfield rank (2024)
    for col in ["rank sm-subfield-1","sm-subfield-1 count","rank","c"]:
        if col in by[2024]: out[col]=(None if pd.isna(by[2024][col]) else float(by[2024][col]))
    out["np2024"]=out["np"][2024]; out["nc2024"]=out["nc"][2024]
    out["cpp2024"]=out["nc2024"]/out["np2024"]
    out["np2020"]=out["np"][2020]; out["nc2020"]=out["nc"][2020]
    out["cpp2020"]=out["nc2020"]/out["np2020"]
    return out

cies=person("Cieszewski"); bett=person("Bettinger")

# quant set membership of each secondary subfield
QSET=set(json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))["taxonomy"]["quant_set"])
cies["sf2_quant"]=cies["sf2"] in QSET; bett["sf2_quant"]=bett["sf2"] in QSET

# benchmarks from ratios.json
def uni(name):
    for m in R["university"]:
        if m["key"]==name: return m
    return None
COH=R["cohort"]["overall"]; GF=R["group"]["Forestry"]
def ctry(k):
    for m in R["country"]:
        if m["key"]==k: return m
UGA=uni("University of Georgia")
US=ctry("United States")

refs={"cies":cies,"bett":bett,
    "bench":{"cohort":{"art":COH["art_per_auth"],"cit":COH["cit_per_auth"],"cart":COH["cit_per_auth"]},
        "forestry":{"art":GF["art_per_auth"],"cit":GF["cit_per_auth"],"cart":GF["cit_per_auth"]},
        "us":{"art":US["art_per_auth"],"cit":US["cit_per_auth"],"cart":US["cit_per_auth"]},
        "uga":
        ({"art":UGA["art_per_auth"],"cit":UGA["cit_per_auth"],"cart":UGA["cit_per_auth"],"sci_yr":UGA["sci_yr"]} if UGA
        else None)}}
json.dump(refs,open(f"{SCR}/refs.json","w",encoding="utf-8"))

def show(p,lab):
    print(f"\n{lab}: {p['name']} | {p['inst']} ({p['cntry']}) | AM={p['sf1']} / AO={p['sf2']} (quant
    sec={p['sf2_quant']})")
    print(f"    np: {p['np']}")
    print(f"    nc: {p['nc']}")
    print(f"    2024: {p['np2024']} papers, {p['nc2024']:,} citations, {p['cpp2024']:.1f} cites/paper (2020:
    {p['cpp2020']:.1f})")
    print(f"    growth 2020->2024: np {p['np2020']}>{p['np2024']} (+{100*(p['np2024']/p['np2020']-1):.0f}%), nc
    {p['nc2020']:,}>{p['nc2024']:,} (+{100*(p['nc2024']/p['nc2020']-1):.0f}%))")
    print(f"    Forestry-subfield rank (2024): {p.get('rank sm-subfield-1')} of {p.get('sm-subfield-1 count')}")
    show(cies,"AUTHOR (Cieszewski)"); show(bett,"COLLEAGUE (Bettinger)")
    print("\n--- benchmarks (five-year ratios) ---")
    b=refs["bench"]
    for k,v in b.items():
        if v: print(f"    {k:<9}: art/auth {v['art']:.0f} | cit/auth {v['cit']:.0f} | cit/art {v['cart']:.1f}" + (f" |
        sci/yr {v.get('sci_yr'):.0f}" if v.get('sci_yr') else ""))
    print("\nUGA in institution list:", "YES" if UGA else "NO")
    print("saved refs.json")

```

24. compute_refs2.py (SHA-256 5242348f7a6e2245...; 47 lines)

```

# -*- coding: utf-8 -*-
# Rebuild refs.json with THREE reference scientists:

```

```

# cies/bett from the forestry cohort file; nash (J.E. Nash, hydrologist) from the full career files.
import pandas as pd, json, re
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
df=pd.read_excel(RAW,sheet_name="Data")
R=json.load(open(f"{SCR}/ratios.json",encoding="utf-8"))

def person(namekey):
    h=df[df["authfull"].astype(str).str.contains(namekey,case=False,na=False)].copy()
    by={int(r["year"]):r for _,r in h.iterrows()}
    out={"name":str(by[2024]["authfull"]),"inst":str(by[2024]["inst_name"]),"cntry":str(by[2024]["cntry"]),"sf1":str(by[2024]["sm-subfield-1"]),"sf2":str(by[2024]["sm-subfield-2"]),"np":{str(y):int(by[y]["np"]) for y in YEARS}}
    r24=by[2024]
    out["h2024"]=int(r24["h"]); out["rank2024"]=int(r24["rank"]); out["c2024"]=float(r24["c"]);
    out["sfrank"]=int(r24["rank sm-subfield-1"]); out["sfN"]=int(r24["sm-subfield-1 count"]);
    out["self"]=float(r24["self%"]) if "self%" in r24 else None
    out["np2024"]=out["np"]["2024"]; out["nc2024"]=out["nc"]["2024"]
    out["np2020"]=out["np"]["2020"]; out["nc2020"]=out["nc"]["2020"]
    out["cpp2024"]=out["nc2024"]/out["np2024"]; out["cpp2020"]=out["nc2020"]/out["np2020"]
    return out
cies=person("Cieszewski"); bett=person("Bettinger")

# NOTE: J.F. Nash (game theorist) is NOT in the Stanford list - verified; handled as an external
# prose-only limiting case in the report. Only the two cohort scientists get computed rows here.
QSET=set(json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))["taxonomy"]["quant_set"])
cies["sf2_quant"]=cies["sf2"] in QSET; bett["sf2_quant"]=bett["sf2"] in QSET
def uni(name):
    for m in R["university"]:
        if m["key"]==name: return m
def ctry(k):
    for m in R["country"]:
        if m["key"]==k: return m
COH=R["cohort"]["overall"]; GFr=R["group"]["Forestry"]; UGA=uni("University of Georgia"); US=ctry("United States")
refs={"cies":cies,"bett":bett,"bench":{"cohort":{"art":COH["art_per_auth"],"cit":COH["cit_per_auth"],"cart":COH["cit_per_auth"]},"forestry":{"art":GFr["art_per_auth"],"cit":GFr["cit_per_auth"],"cart":GFr["cit_per_auth"]},"us":{"art":US["art_per_auth"],"cit":US["cit_per_auth"],"cart":US["cit_per_auth"]},"uga":{"art":UGA["art_per_auth"],"cit":UGA["cit_per_auth"],"cart":UGA["cit_per_auth"],"sci_yr":UGA["sci_yr"]}}}
json.dump(refs,open(f"{SCR}/refs.json","w",encoding="utf-8"))
for k in ["cies","bett"]:
    p=refs[k]
    print(f"{k}: {p['name']} | h={p['h2024']} rank={p['rank2024']:,} c={p['c2024']:.3f} | np {p['np']} | nc2024={p['nc2024']:,} cpp={p['cpp2024']:.1f} | sfrank {p['sfrank']:,}/{p['sfN']:,}")
print("saved refs.json (2 cohort refs; John Nash handled in prose)")

```

25. corr.py (SHA-256 ff5fc0403115e1c4...; 73 lines)

```

import openpyxl, math, sys
from collections import defaultdict
sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford")
from cc_defs import CC

XLSX = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
wb = openpyxl.load_workbook(XLSX, read_only=True, data_only=True)
ws = wb['Data']
rows = ws.iter_rows(min_row=1, values_only=True)
hdr = list(next(rows)); idx = {h:i for i,h in enumerate(hdr)}
i_cntry=idx['cntry']; i_np=idx['np']; i_nc=idx['nc']; i_sf1=idx['sm-subfield-1']; i_sf2=idx['sm-subfield-2']; i_inst=idx['inst_name']

ctry=defaultdict(lambda:[0,0.0,0.0]) # S,P,C
inst=defaultdict(lambda:[0,0.0,0.0])
for r in rows:
    sf1=r[i_sf1]; sf2=r[i_sf2]
    if sf1 is None: continue
    if not ((sf1=='Forestry') or (sf1=='Ecology' and sf2=='Forestry')): continue
    npv=float(r[i_np] or 0); ncv=float(r[i_nc] or 0)
    info=CC.get(str(r[i_cntry]).lower()) if r[i_cntry] else None
    cname=info[0] if info else None
    if cname:
        a=ctry[cname]; a[0]+=1; a[1]+=npv; a[2]+=ncv

```

```

    inm=r[i_inst]
    if inm:
        b=inst[inm]; b[0]+=1; b[1]+=npv; b[2]+=ncv

def pearson(x,y):
    n=len(x); mx=sum(x)/n; my=sum(y)/n
    sxy=sum((a-mx)*(b-my) for a,b in zip(x,y))
    sxx=sum((a-mx)**2 for a in x); syy=sum((b-my)**2 for b in y)
    return sxy/math.sqrt(sxx*syy)

def ranks(v):
    order=sorted(range(len(v)), key=lambda i:v[i])
    r=[0.0]*len(v); i=0
    while i<len(v):
        j=i
        while j+1<len(v) and v[order[j+1]]==v[order[i]]: j+=1
        avg=(i+j)/2.0+1
        for k in range(i,j+1): r[order[k]]=avg
        i=j+1
    return r

def spearman(x,y):
    return pearson(ranks(x),ranks(y))

# Countries with >=5 scientists/yr => S>=25
csetl={n:v for n,v in ctry.items() if v[0]>=25}
print('N countries (S>=25, i.e. >=5/yr):', len(csetl))
names=sorted(csetl, key=lambda n:-csetl[n][0])
apa=[]; cpar=[]; cpa=[]
for n in names:
    S,P,C=csetl[n]
    apa.append(P/S); cpar.append(C/P); cpa.append(C/S)
    print(f' {n:16s} S={S:5.0f} art/auth={P/S:8.2f} cit/art={C/P:7.2f} cit/auth={C/S:9.1f}')
print()
print('Country art/auth vs cit/art Pearson =', round(pearson(apa,cpar),4), ' Spearman =',
round(spearman(apa,cpar),4))
print('Country art/auth vs cit/auth Pearson =', round(pearson(apa,cpa),4))

# Institutions with >=3 scientists/yr => S>=15
isel={n:v for n,v in inst.items() if v[0]>=15}
print('\nN institutions (S>=15, i.e. >=3/yr):', len(isel))
iapa=[]; icpar=[]
for n in sorted(isel, key=lambda n:-isel[n][0]):
    S,P,C=isel[n]
    iapa.append(P/S); icpar.append(C/P)
print('Institution art/auth vs cit/art Pearson =', round(pearson(iapa,icpar),4), ' Spearman =',
round(spearman(iapa,icpar),4))

# sensitivity: try >5/yr strict (S>25) for countries
csetl2={n:v for n,v in ctry.items() if v[0]/5.0>5}
a2=[v[1]/v[0] for v in csetl2.values()]; c2=[v[2]/v[1] for v in csetl2.values()]
print('\n[sensitivity] countries S/5>5 strict, N=',len(csetl2),'Pearson art/auth vs
cit/art=',round(pearson(a2,c2),4))

```

26. dedup_bettinger.py (SHA-256 95ed768d4dceb0dd...; 352 lines)

```

# -*- coding: utf-8 -*-
# De-duplicate Pete Bettinger's Google Scholar list (user=rvCOS60AAAAJ, fetched July 2026, sorted by pubdate).
# Titles truncated to ~90 chars (ample for exact-duplicate detection); artifacts/software/CV/name items removed.
import re, unicodedata
E = [
    # page 1 (2026-2019)
    "AI Technology Use in North American Forestry",
    "Review of post-settlement influences on eastern United States forest landscapes: The rise of the",
    "Navigating uncertainty: Timberland management with variable fertilization responses",
    "Linking environmental sustainability in forest companies to ecosystem services: A systematic",
    "Two orders of wall-to-wall geographic data",
    "Using optimization for maximizing future utility of ecosystem services",
    "The geography of forest management",
    "Field Note: A Case Study of Generative AI Responses Concerning Infrared Reflectance of Evergreen",
    "The quality of outcomes from tabu search when enhanced to address contemporary forest harvest",
    "Opposing effects of warming and increased precipitation on site productivity of planted larch",
    "Approaches for simulating alternative futures of complex forested landscapes: A review",
    "Using optimization for maximizing the future utility of ecosystem services",
    "Competition in mixed Scots pine and Pyrenean oak stands modifies allometry and partially affects",
    "Forest Harvest Scheduling. From Linear Programming to Heuristic Search",
    "Quality of Threshold Accepting Heuristic Search Results When Attempting to Solve Forest Harvest",

```

"Semi-automatic stand delineation based on very-high-resolution orthophotographs and topographic",
 "Uncertainty analysis of linear and non-linear regression models in the modeling of water quality",
 "Sustainability language found in forest plans and its mathematical modeling potential",
 "Associations between forest harvest scheduling and artificial intelligence",
 "Mapping percent canopy cover using individual tree- and area-based procedures that are based on",
 "OLAF: An Assessment of Online Learning Outcomes Among Forestry and Natural Resource Management",
 "A forest management evaluation system for small private forest landowners",
 "Estimating forest inventory information for the Talladega National Forest using airborne laser",
 "Technology to Assist Land Management: User Satisfaction with an Online Forest Management System",
 "How to maximize the joint benefits of timber production and carbon sequestration for rural areas",
 "Stand spatial structural diversity: Developing and validating a novel index",
 "Dear AI, Can you help me estimate basal area from a prism sample?",
 "Dear AI, What is a board foot?",
 "Forest management planning technologies in the light of climate change",
 "Mapping Ecological Condition Classes of a Natural Pine-Dominated National Forest in the Southeast",
 "Approaches for Addressing Spatial Connectivity of Final Harvests Within Forest Harvest Scheduling",
 "The contributions of stand characteristics on carbon sequestration potential are triple that of",
 "Modelling Spatial Connectivity of Forest Harvest Areas: Exact and Heuristic Approaches",
 "Transaction costs and investment interest in the U.S. South and the Pacific Northwest timberland",
 "Estimating timber supply elasticity of private forest landowners in the US South",
 "Recreation potential assessment at Tamarix forest reserves: A method based on multi-criteria",
 "The effects of nearby trees on the positional accuracy of GNSS receivers in a forest environment",
 "Sustainability language in forest management plans: A comparative analysis for public forests of",
 "Periodic harvests, rather than passive conservation, increase the carbon balance of boreal",
 "Usefulness and need for digital technology to assist forest management: Summary of findings from",
 "Comparison between graph theory connectivity indices and landscape connectivity metrics for",
 "Geographic information system skills for foresters and natural resource managers",
 "Introducing a New Approach in Stand Tending Planning and Thinning Block Designation by Using Mixed",
 "Evaluating the accuracy of remote dendrometers in tree diameter measurements at breast height",
 "Handbook of land and tree measurements",
 "Timberland transaction costs: survey results and insights",
 "Estimating Timber Supply Elasticity of Private Forest Landowners in South Georgia",
 "Maximizing the canopy openness for replanted Korean pine seedlings under secondary forests in",
 "Virtually exhausted: Experiences in hybrid conference management during the pandemic",
 "Timber business in West Africa: a review and outlook",
 "A hierarchical binary process model to assess deviation from desired ecological condition across",
 "Mission Impossible: Positions determined by basic mapping-grade and recreation-grade GNSS",
 "A theoretical minimum solution for heuristics: The case of the spatial timber harvest problem",
 "Habitat suitability modelling of rare turkeybeard (*Xerophyllum Asphodeloides*) species in the",
 "Urban land use cover changes in three developed cities of the United States: San Diego, Denver",
 "A simulation-optimization modeling approach for conjunctive water use management in a semi-arid",
 "Optimizing rotation lengths for maximizing carbon balance of larch plantations in northeast China",
 "Forest sustainability in state forest management plans: A content analysis",
 "Forester preferences on revising Silvics of North America: A survey of registered foresters in",
 "Optimizing neighborhood-based stand spatial structure: Four cases of boreal forests",
 "Spatial modeling of forest stand susceptibility to logging operations",
 "Introduction to forestry and natural resources",
 "Machine Learning Modeling of Forest Road Construction Costs",
 "A combination of biotic and abiotic factors and diversity determine productivity in natural",
 "Stakeholder perceptions on the need for updated tree species distributions maps",
 "Machine learning modeling of forest road construction costs",
 "Ten-year estimation of Oriental beech (*Fagus orientalis* Lipsky) volume increment in natural",
 "Forestry research in the Middle East: A bibliometric analysis",
 "An integrated multi-criteria decision analysis and optimization modeling approach to spatially",
 "Variation in results of three biology-focused search engines: A case study using North American",
 "Random forest regression model for estimation of the growing stock volumes in Georgia, USA, using",
 "A decision making approach for assignment of ecosystem services to forest management units: A",
 "Estimating the optimal internal carbon prices for balancing forest wood production and carbon",
 "Overview of the 12th Southern Forestry and Natural Resource Management GIS Conference",
 "Contributions of Donald J. Lipscomb to early forestry GIS and low cost applications to address",
 "Mapping forest disturbances between 1987-2016 using all available time series Landsat TM/ETM+",
 "Estimation of tree heights in an uneven-aged, mixed forest in northern Iran using artificial",
 "Tree species rankings based on publication rate (North America)",
 "Analysis of fire frequency on the Talladega National Forest, USA, 1998-2018",
 "A comparative analysis of five forest certification programs",
 "Using mixed integer goal programming in final yield harvest planning: A case study from the",
 "Effect of working forest conservation easements on surrounding property values",
 "Towards a characterization of working forest conservation easements in Georgia, USA",
 "Timberland transaction costs and due diligence: a literature review and assessment of research",
 "The applicability of recreation-grade GNSS receiver (GPS watch, Suunto Ambit Peak 3) in a",
 "Preferences of motorcyclists to views of managed, rural southern United States landscapes",
 "Mapping Human and Natural Systems",
 "The terms foresters and planners in the United States use to infer sustainability in forest",
 "Key challenges in forest management",
 "Achieving sustainable management of boreal and temperate forests",
 "k 近傍法と FIA データによる米国ジョージア州沿岸部における森林蓄積量の推定",
 "Preliminary analysis of forest stand disturbances in Coastal Georgia (USA) using Landsat time",
 "Review of the effects of conservation easements on surrounding property values",
 "Proceedings of the 12th Southern Forestry and Natural Resources GIS Conference: December 9-10",
 "Growth and yield models for uneven-aged forest stands managed under a selection system in northern",
 "Smartphone GPS accuracy study in an urban environment",

"Evaluation of sites for reestablishment of the American chestnut (*Castanea dentata*) in northeast",
 "Land cover transitions in the United States South: 2007-2013",
 "GNSS use in forestry - A multi-national survey from Iran, Slovakia and southern USA",
 # page 2 (2018-2010)
 "Analysis of temporal changes in land cover and landscape metrics of a managed forest in the west",
 "Proceedings of the 11th Southern Forestry and Natural Resources GIS Conference: December 11-12",
 "Review of the effects of conservation easements on surrounding property values",
 "A perspective of the mapping technology knowledge and skills required of entry-level forestry job",
 "Synthesis of what we have learned studying GPS in forests",
 "Mixed-effects modeling for tree height prediction models of Oriental beech in the Hyrcanian",
 "Decision support approaches in adaptive forest management",
 "Mapping potential motorized sightseeing recreation supply across broad privately-owned landscapes",
 "Follow-up study of the importance of mapping technology knowledge and skills for entry-level",
 "GIS-based sustainable assessment of crop residues for optimal siting of biogas plants",
 "Reflections on the number of independent solutions for forest spatial harvest scheduling problems",
 "Analysis of temporal changes in land cover and landscape metrics of a managed forest in the west b",
 "Estimation of urban woody vegetation cover using multispectral imagery and LiDAR",
 "Overview of the third Special Section of papers from the 10th Southern Forestry and Natural",
 "LMSCHED. BAS",
 "Bettinger, Pete PhD. pdf",
 "Effects of forest thinning on static horizontal positions collected with a mapping-grade GNSS",
 "Forests Special Issue Decision Support Approaches in Adaptive Forest Management",
 "Forest planning heuristics - Current recommendations and research opportunities for s-metaheurist",
 "Spatial forest plan development using heuristic processes seeded with a relaxed linear programming",
 "Assessment of potential agricultural and short-rotation forest bioenergy crop establishment sites",
 "Estimation of soil moisture across broad landscapes of Georgia and South Carolina using the",
 "Mapping the spatial distribution of forest growth characteristics using different geostatistical",
 "Stumpage prices: A review of influential factors",
 "Auerbach's wilderness medicine e-book",
 "A brief overview of the second special section of papers from the Tenth Southern Forestry and",
 "Nonlinear mixed-effects branch diameter and length models for natural Dahurian larch",
 "SOFIE (Scheduling of Forest Investment Endeavors), version 1.0",
 "Also Mentioned",
 "An overview of the 10th Southern Forestry and Natural Resource Management GIS Conference",
 "Optimisation in forest management",
 "Proceedings of the 10th Southern Forestry and Natural Resources GIS Conference: December 7-8",
 "Assessment of geographic information system (GIS) skills employed by graduates from three forestry",
 "Finding feasible harvest zones in mountainous areas using integrated, spatial multi-criteria",
 "Evaluating the neighborhood, hybrid and reversion search techniques of simulated annealing",
 "Nonlinear mixed-effects branch diameter and length models for natural Dahurian larch (*Larix*",
 "Time efficiency of selected types of adjacency constraints in solving unit restriction models",
 "The importance of mapping technology knowledge and skills for students seeking entry-level forestr",
 "A comparison of two sampling approaches for assessing the urban forest canopy cover from aerial",
 "A comparison of a neighborhood search technique for forest spatial harvest scheduling problems",
 "Using mixed integer multi-objective goal programming for stand tending block designation",
 "Spatial forest harvest scheduling for areas involving carbon and timber management goals",
 "Spatial forest harvest scheduling for areas involving carbon and timber management goals",
 "Tree-level harvest optimization for structure-based forest management based on the species mingling",
 "Assessing the accuracy of tree diameter measurements collected at a distance",
 "Modeling harvest scheduling in multifunctional planning of forests for longterm water yield",
 "Forest carbon conservation and management: integration with sustainable forest management",
 "Terrain evaluation model to identify landing locations in mountain areas",
 "The role of the US mayors in addressing climate change and urban forests",
 "An optimization model to solve skidding problem steep slope terrain",
 "Estimating urban forest carbon sequestration potential in the southern United States",
 "Forest tenure and sustainable forest management",
 "Forest Plans of North America",
 "Response and recovery after severe wind storms using hierarchical open vehicle routing",
 "How a GNSS receiver is held may affect static horizontal position accuracy",
 "Search reversion within s-metaheuristics: impacts illustrated with a forest planning problem",
 "Urban tree cover change in Detroit and Atlanta, USA, 1951-2010",
 "A Brief Overview of the 9th Southern Forestry and Natural Resources GIS Conference",
 "Proceedings of the 9th Southern Forestry and Natural Resources GIS Conference: December 8-10",
 "Assessment of a Global Land Cover Classification Allocated Across the Landscape of Georgia (USA)",
 "Dynamic accuracy of recreation-grade GPS receivers in oak-hickory forests",
 "Evolution of modern forest management planning in the Republic of Turkey",
 "Efficient assessments of urban tree planting potential within or near the southern Piedmont region",
 "Micro-detail comparative forest site analysis using high-resolution satellite imagery",
 "Wood quality assessment of tree trunk from the tree branch sample and auxiliary data based on NIR",
 "Forest management planning technology issues posed by climate change",
 "Using optimization techniques in designing forest roads and road networks",
 "HATT (Heuristic Algorithm Teaching Tool), version 2.0",
 "Assessing Potential Urban Tree Planting Sites in the Piedmont of the United States",
 "Forest management issues of the southern United States and comparisons with Turkey",
 "Trees at Work: Solving Critical Issues with Urban Forest Canopy",
 "Pitfalls and potential of particle swarm optimization for contemporary spatial forest planning",
 "Relative vulnerability of the forests along southeastern African coasts to cyclones",
 "Reflections on the estimation of stand-level forest characteristics using Landsat satellite",
 "Static Horizontal Positions Determined with a Consumer-Grade GNSS Receiver",
 "Ecologically appropriate restoration thinning in the Northwest Forest Plan area",
 "Proceedings of the 8th Southern Forestry and Natural Resources GIS Conference: December 11-13",

"Determining the Relative Vulnerability of Forests to Tropical Cyclones",
 "Application of a new multi-valued particle swarm optimization to forest harvest schedule",
 "Influence of the juxtaposition of trees on consumer-grade GPS position quality",
 "Cypress dome characteristics within one county of the Southern United States: a case study of",
 "The Conclusion of Perspectives on New Developments of Decision Support Systems for Sustainable",
 "Decision Support Tools for Sustainable Forest Management. 2010 Workshop on Decision Support",
 "Carbon Sinks and Climate Change, Forests In The Fight Against Global Warming",
 "Forest Management in a Climate Change Era: Options for Planning",
 "Perspectives on New Developments of Decision Support Systems for Sustainable Forest Management",
 "Knowledge gained and gaps suggested from recent forestry-related GPS research",
 "Global navigation satellite system research in forest management: A summary of horizontal, vertical",
 "The impact of prescribed fire on longleaf pine productivity",
 "Forest planning desk reference: Terminology and examples",
 "A prototype model for estimating the location of forest damage from tropical cyclones",
 "Informed development of meta heuristics for spatial forest planning problems",
 "Southern Forestry and Natural Resources GIS Conference (SOFOR GIS), Athens, Georgia, USA",
 "The 2009 Southern Forestry and Natural Resources GIS Conference",
 "One Year's Experience with a Recreation-Grade GPS Receiver",
 "Perceptions of foresters of wind damage in Mississippi forests",
 "Analytical Decisionmaking Considerations for Upgrading or Changing Geographic Information System",
 "Using bobcat habitat suitability to prioritize habitat preservation on a suburbanizing barrier",
 "An overview of methods for incorporating wildfires into forest planning models",
 "Assessing Forest Fragmentation in Southern US Industrial Forest Plans that Accommodate Different",
 # page 3 (2008-2001)
 "Cross-Sectoral Policy Developments in Forestry, Yves C. Dube and Franz Schmithusen (eds.)",
 "Field-based Near-Infrared Reflectance Spectroscopy for Spatially Explicit Quantification of Forest",
 "Tabu search experience in forest management and planning",
 "Habitat and commodity production trade-offs in Coastal Oregon",
 "Status of GPS Research at the Whitehall GPS Test Course",
 "Spatial optimisation-computational methods",
 "Simulated annealing. Encyclopedia of Ecology",
 "Warnell School of Forest Resources",
 "Natural resources: economics, management and policy",
 "Effects of arranging forest fuel reduction treatments in spatial patterns on hypothetical",
 "Surveys of GIS usage and needs among natural resource managers, Extension agents, and land use",
 "Geographic information systems: Applications in natural resource management",
 "Synthesis of knowledge of hazardous fuels management in loblolly pine forests",
 "Back to the future part 2: Surveying geospatial technology needs of Georgia Agriculture and Natural",
 "Back to the future part 1: Surveying geospatial technology needs of Georgia land use planners",
 "Assessment of three heuristics for developing large-scale spatial forest harvest scheduling plans",
 "Estimating the effects of adjacency and green-up constraints on landowners of different sizes and",
 "An assessment of geographic information system skills used by field-level natural resource",
 "Heuristics in forest planning",
 "General management practices for hazardous fuels management in loblolly pine (Pinus taeda) forests",
 "Landscape-level optimization using tabu search and stand density-related forest management",
 "Cumulative ecological and socio-economic effects of forest policies in coastal Oregon",
 "A historical perspective on the use of GIS and remote sensing in natural resource management, as",
 "Additional insight into the performance of a new heuristic for solving spatially constrained",
 "Simulating forest structure, timber production, and socioeconomic effects in a multi-owner",
 "On Integrating Water Yield Models with Forest Planning Efforts",
 "Estimation of above-ground biomass with remotely sensed imagery: A brief literature review",
 "Evaluating the salvage value of fire-killed timber by helicopter-effects of yarding distance and",
 "An economic and landscape evaluation of the green-up rules for California, Oregon, and Washington",
 "Historical disturbance regimes as a reference for forest policy in a multiowner province: a",
 "ASSESSING GPS ACCURACY, WAAS, AND A CHOKE RING ANTENNA",
 "Proceedings of the 5th Southern Forestry and Natural Resources GIS Conference, Asheville, North",
 "ABOVE-GROUND BIOMASS ESTIMATION IN A FORESTLAND USING A LANDSAT THEMATIC MAPPER IMAGE WITH SUPERVIS",
 "Assessing GPS accuracy, WAAS, and a choke ring antenna solution in a southern hardwood forest",
 "A selective bibliography of scientific literature on hazardous fuel management in loblolly pine",
 "Forest Carbon Estimation Protocol for the State of Georgia",
 "Aboveground biomass estimation in a forestland using a Landsat thematic mapper image with",
 "Linking multiple tools: An American case",
 "A new heuristic method for solving spatially constrained forest planning problems based on",
 "Measures of change in projected landscape conditions",
 "The effects of spatial harvest scheduling constraints on the value of a medium-sized forest",
 "A density-dependent stand-level optimization approach for deriving management prescriptions for",
 "Spatial optimization of fuel management activities",
 "A hierarchical spatial framework for forest landscape planning",
 "Systems Analysis in Forest Resources: Proceedings of the 2003 Symposium, October 7-9, 2003",
 "Proceedings of the 4th Southern Forestry and Natural Resources GIS Conference: Athens, Georgia; De",
 "Examination of tradeoffs between timber production and spotted owl habitat in coastal Oregon",
 "HATT: the heuristic algorithm teaching tool for advanced forest planning courses",
 "SIMULATOR (LAMPS)",
 "Determining thinning regimes to reach stand density targets for any-aged stand management in the",
 "Application of spatial technologies in wildlife biology",
 "Simulating the Implications of Alternative Policies for a Large Forested Landscape",
 "The SafeD forest landscape planning model",
 "Second International Conference on Forest Measurements and Quantitative Methods and Management & The",
 "Hastening the return of complex forests following fire: the consequences of delay",
 "The key literature of, and trends in, forest-level management planning in North America, 1950-2001",
 "Landscape management policy simulator (LAMPS), version 1.1. USER GUIDE. Research Contribution 43",

"Finding the Vertical Alignment of Forest Roads Using Several Heuristic Techniques",
 "Geographic Information Systems: Applications in Forestry and Natural Resources Management",
 "Spatial forest plan development with ecological and economic goals",
 "A Vegetation Similarity Index Driven by Distance-dependent Spatial Information",
 "GIS: An updated primer on a powerful management tool",
 "Spatial forest planning: To adopt, or not to adopt?",
 "A heuristic process for solving mixed-integer land management and transportation models",
 "Spatial scheduling of forest management activities using a dynamic deterministic harvest block",
 "CONGRUENT MANAGEMENT OF MULTIPLE RESOURCES: PROCEEDINGS FROM THE WOOD COMPATIBILITY INITIATIVE",
 "An evaluation of tradeoffs between wood production and ecological integrity in the Oregon Coast",
 "Challenges to Integrating Wood Production Biodiversity, and Other Socioeconomic Values at Broad",
 "P, E2-Intensive Forest Management Systems Addressing Economic, Environmental, and Social Goals",
 "A SIMULATION MODEL FOR EVALUATING FOREST POLICIES ON LARGE LANDSCAPES WITH MULTIPLE OWNERS",
 "Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest",
 "Assessing the ecological consequences of forest policies in a multi-ownership province in Oregon",
 "Precision Forestry: Proceedings of the First International Precision Forestry Cooperative Symposium",
 "Eight heuristic planning techniques applied to three increasingly difficult wildlife planning",
 "Challenges and opportunities for linking the modeling of forest vegetation dynamics with landscape",
 "First International Precision Forestry Symposium Agenda",
 # page 4 (undated tail)
 "Guangyu Wang",
 "Regional Comparison of Forest Management: Snapshots in Time 1900-2000",
 "eYield",
 "2003 Symposium for Systems Analysis in Forest Resources October 7-9, Skamania Lodge, Stevenson",
 "CS Decisions-Teaching",
 "University of Georgia",
 "Designing green landscapes",
 "Stakeholder Perceptions on the Need for Updated Tree Species Distribution Maps",
 "Unraveling nonlinear and asymmetric price transmission in US South timber markets: a smooth",
 "ORMANCILIKTA SEKTOREL PLANLAMANIN 50. YILI ULUSLARARASI SEMPOZYUMU BILDIRILER KITABI",
 "Forest Plans of North America",
 "13th Southern Forestry and Natural Resource Management GIS Conference",
 "Welcome to the Council on Forest Engineering (COFE) Publications Website",
 "Evaluating the performance of neighborhood search techniques of simulated annealing in three",
 "Integrating Spatial Aquatic Habitat Goals Into A Planning Process Considering Harvesting System",
 "COMPARISON OF LANDSCAPE PLANNING MODELS",
 "User Satisfaction with an Online Forest Management Tool and Feedback from Stakeholders",
 "Research on an iconic cat pinpoints land for preservation",
 "ISSUES FACING FOREST PLANNING IN THE SOUTHERN UNITED STATES",
]

```
def strip_diac(s):
    return "".join(c for c in unicodedata.normalize("NFKD", s) if not unicodedata.combining(c))

def norm(t):
    t = strip_diac(t).lower().replace("u.s.", "us")
    t = re.sub(r"^[a-z0-9 ]", " ", t)
    return re.sub(r"s+", " ", t).strip()

# artifacts: software files, CV/name/affiliation/website fragments, section headers, non-Bettinger book
ART_EXACT = {norm(x) for x in [
    "LMSCHED. BAS", "Bettinger, Pete PhD. pdf", "Also Mentioned", "SOFIE (Scheduling of Forest Investment Endeavors),
    version 1.0",
    "HATT (Heuristic Algorithm Teaching Tool), version 2.0", "SIMULATOR (LAMPS)", "eYield", "Guangyu Wang", "University
    of Georgia",
    "Warnell School of Forest Resources", "CS Decisions-Teaching", "Welcome to the Council on Forest Engineering
    (COFE) Publications Website",
    "Research on an iconic cat pinpoints land for preservation", "Auerbach's wilderness medicine e-book",
    "Landscape management policy simulator (LAMPS), version 1.1. USER GUIDE. Research Contribution 43",
    "Mapping Human and Natural Systems", "Designing green landscapes",
    ]}
# Japanese/Turkish-language translation versions of English papers
TRANSLATIONS = {norm(x) for x in [
    "k近傍法と FIA データによる米国ジョージア州沿岸部における森林蓄積量の推定",
    "ORMANCILIKTA SEKTOREL PLANLAMANIN 50. YILI ULUSLARARASI SEMPOZYUMU BILDIRILER KITABI",
    ]}
def is_jp(t): # any CJK char
    return any(' ' <= c <= 'ㇿ' or '一' <= c <= '𐤟' for c in t)

raw=len(E)
artifacts,kept=[],[]
for t in E:
    n=norm(t)
    if n in ART_EXACT or n in TRANSLATIONS or is_jp(t) or len(n)<6:
        artifacts.append(t)
    else:
        kept.append(t)
seen,uniq,dups={},[],[]
for t in kept:
    n=norm(t)
    if n in seen: dups.append(t)
```

```

else: seen[n]=t; uniq.append(t)

print(f"RAW transcribed entries:      {raw}")
print(f" - artifacts/software/names:  {len(artifacts)}")
print(f" - translations (JP/TR):       {sum(1 for t in E if is_jp(t) or norm(t) in TRANSLATIONS)}")
print(f" - exact/near duplicates:        {len(dups)}")
print(f"= unique works:                  {len(uniq)}")
print("\nDuplications removed:")
for a in dups: print("  ", a[:75])
print("\nArtifacts removed:")
for a in artifacts: print("  ", a[:75])

```

27. dedup_scholar.py (SHA-256 5a4297cf4f8e33df...; 291 lines)

```

# -*- coding: utf-8 -*-
# De-duplicate Chris J. Cieszewski's Google Scholar list (211 raw entries, fetched July 2026,
# user=npjJSDoAAAAJ, sorted by pubdate). Reproducible: normalize titles, drop parsing artifacts,
# collapse exact/near-exact duplicates, and flag Polish-language translation versions of English papers.
import re, unicodedata

# --- raw entries as "title" strings (year not needed for de-dup) ---
ENTRIES = [
# page 1 (2024 -> 2005)
"Interconnected Biodiversity: Mapping the Relationship Between Tree and Breeding Bird Species Diversity in Georgia's Forests, USA",
"SOLUTIONS TO THE BASE-AGE VARIANT MODELS",
"The effects of nearby trees on the positional accuracy of GNSS receivers in a forest environment",
"Mission Impossible: Positions determined by basic mapping-grade and recreation-grade GNSS receivers cannot emulate the actual spatial pattern of trees",
"NEAREST NEIGHBOR BIAS IN THE SUBSTITUTION OF MISSING VALUES",
"UTADA: Unified Theory of the Algebraic Differences Approaches-Derivation of Dynamic Site Equations from Yield-Site Relationships",
"Estimating biomass and carbon storage by Georgia Forest types and species groups using the FIA data diameters, basal areas, site indices, and Total Heights",
"Random Forest regression model for estimation of the growing stock volumes in Georgia, USA, using dense Landsat time series and FIA dataset",
"ALS-based detection of past human activities in the Bialowieza Forest-New evidence of unknown remains of past agricultural systems",
"A comparative analysis of five forest certification programs",
"The applicability of recreation-grade GNSS receiver (GPS watch, Suunto Ambit Peak 3) in a forested and an open area compared to a mapping-grade receiver (Trimble Juno T41)",
"Mapping forest disturbances between 1987-2016 using all available time series Landsat TM/ETM+ imagery: Developing a reliable methodology for Georgia, United States",
"Mapping of recreation suitability in the Belgrad Forest Stands",
"To Act or not to Act Bialowieza Forest under Conflicting Ecological Paradigms",
"Evaluation of sites for the reestablishment of the American chestnut (Castanea dentata) in northeast Georgia, USA",
"Preliminary analysis of forest stand disturbances in Coastal Georgia (USA) using Landsat time series stacked imagery",
"Special Issue on SELECTED EXAMPLES OF RECENT TRENDS IN COMPUTING",
"Comparing properties of self-referencing models based on nonlinear-fixed-effects versus nonlinear-mixed-effects modeling approaches",
"Proceedings of the 11th Southern Forestry and Natural Resources Management GIS Conference",
"Overview of the 2017 11th Southern Forestry and Natural Resources GIS Conference",
"Proceedings of the 11th Southern Forestry and Natural Resources GIS Conference: December 11-12, 2017, Athens, Ga",
"Taxonomy for the optimization in forest management: a review and assessment",
"Handbook of land and tree measurements",
"Estimation of soil moisture across broad landscapes of Georgia and South Carolina using the triangle method applied to MODIS satellite imagery",
"Estimation of premature forests in Georgia (USA) using US Forest Service FIA data and Landsat imagery",
"Assessment of potential agricultural and short-rotation forest bioenergy crop establishment sites in Jackson County, Florida, USA",
"Estimation of premature forests in Georgia (USA) using U.S. Forest Service FIA data and Landsat imagery",
"Empirical equations for estimating aboveground biomass of Betula pendula growing on former farmland in central Poland",
"The importance of mapping technology knowledge and skills for students seeking entry-level forestry positions: Evidence from job advertisements",
"Optimisation in forest management",
"Proceedings of the 10th Southern Forestry and Natural Resources GIS Conference: December 7-8, 2015, Athens, Ga",
"Forest Plans of North America",
"Vegetation Damage, Fuel Droplet Distribution and Airplane Height",
"Synopsis of forest management plans of North America",
"The analysis of Pine stumpage prices based on timber sale characteristics of the Southern United States",
"Assessing the Accuracy of Tree Diameter Measurements Collected at a Distance",
"Multi-source K-nearest neighbor, Mean Balanced forest inventory of Georgia",
"Evolution of modern forest management planning in the Republic of Turkey",
"Micro-detail comparative forest site analysis using high-resolution satellite imagery",

```

"Wood quality assessment of tree trunk from the tree branch sample and auxiliary data based on NIR Spectroscopy and SilviScan",
 "Ocena wlasciwosci Drewna Brzozowego za Pomoca Spektroskopii NIR i SilviScan dla Probek z Polski USA i Smolenska",
 "Mikro-Detaliczna Analiza Porownawcza Obszaru Lesnego Przy Uzyciu Zdjec Satelitarnych Wysokiej Rozdzielczosci",
 "Forest management issues of the southern United States and comparisons with Turkey",
 "Pitfalls and potential of particle swarm optimization for contemporary spatial forest planning",
 "Generalization of the Coefficient of Determination or R-square and its application to Self-Referencing Models",
 "In Memory of René Marek Cieszewski",
 "Generalization of the coefficient of determination or R^2 and its application to self-referencing models",
 "Handbook of land and tree measurements",
 "Reflections on the estimation of stand-level forest characteristics using Landsat satellite imagery",
 "Selected Examples of Advanced FIA Data Analysis",
 "Selected Examples of Advanced FIA Data Analysis",
 "The Conclusion of Perspectives on New Developments of Decision Support Systems for Sustainable Forest Management",
 "Comparing the use of three dendrometers for measuring diameters at breast height",
 "Decision Support Tools for Sustainable Forest Management. 2010 Workshop on Decision Support Systems in Sustainable Forest Management, Lisbon, Portugal, 19-21 April 2010",
 "Perspectives on New Developments of Decision Support Systems for Sustainable Forest Management",
 "Perspectives on new developments of decision support systems for sustainable forest management",
 "Handbook of land and tree measurements",
 "Spatially Explicit Biomass Supply Sustainability Analysis for Bioenergy Mill Siting in Georgia, USA",
 "Handbook of land and tree measurements",
 "Spatial regression modeling of tree height-diameter relationships",
 "Trends in spatial forest planning",
 "Impacts of management intensity and harvesting practices on long-term forest resource sustainability in Georgia",
 "Assessment of stream management zones and road beautifying buffers in Georgia based on remote sensing and various ground inventory data",
 "Closest spectral fit for removing clouds and cloud shadows",
 "Sensitivity analysis on long-term fiber supply simulations in Georgia",
 "The Foundation of Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS)",
 "A comparison of GPS performance in a southern hardwood forest: Exploring low-cost solutions for forestry applications",
 "Handbook of land and tree measurements",
 "New functions and programs in Hypermap software development for internet-based displaying of FIA data",
 "2007 Southern Mensurationists Conference, Hilton Garden Inn Beachfront, Orange Beach, Alabama, 12-15 November 2007",
 "The Foundation of Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS)",
 "Preliminary results of spatial modeling of selected forest health variables in Georgia",
 "Large area forest inventory using Landsat ETM+: A geostatistical approach",
 "Generalized algebraic difference approach derivation of dynamic site equations with polymorphism and variable asymptotes from exponential and logarithmic functions",
 "Handbook of land and tree measurements",
 "Parameter estimation of base-age invariant site index models: which data structure to use?—A discussion",
 "A linear mixed-effects model of biomass and volume of trees using Landsat ETM+ images",
 "K nearest neighbor method for forest inventory using remote sensing data",
 "New dynamic site equation that fits best the Schwappach data for Scots pine (*Pinus sylvestris* L.) in Central Europe",
 "The editors of European Journal of Forest Research would like to thank the following referees for their unselfish work for the journal in 2006",
 "Second International Conference on Forest Measurements and Quantitative Methods and Management, Hot Springs, Arkansas, USA, 15-18 June 2004",
 "Inventory Updates and Related Topics: From Special Sampling Techniques to Stochastic Analyses",
 "Base-age invariance properties of two techniques for estimating the parameters of site index models",
 "Analysis of different base models and methods of site model derivation for Scots pine",
 "Hot Springs Conference on Forest Growth and Yield",
 "USING GEOSPATIAL METHODS FOR DERIVATION OF FINE SPATIAL RESOLUTION FOREST INVENTORY FROM GROUND INVENTORY DATA AND LANDSAT IMAGERY",
 "TEMPORAL ANALYSIS OF LANDSAT SATELLITE IMAGERY FOR LAND COVER CHANGE TRACKING IN SOUTHEASTERN GEORGIA",
 "Georgia Carbon Sequestration Registry, Forest Carbon Estimation Protocol Technical Guidelines, Draft",
 "Using geo-spatial methods for derivation of fine resolution forest inventory in Georgia from ground inventory data and Landsat imagery",
 "Spatial clusters and variability analysis of tree mortality",
 "Second International Conference on Measurements and Quantitative Methods and Management & the 2004 Southern Mensurationists Meeting",
 "Applying geostatistics for investigations of forest ecosystems using remote sensing imagery",
 "A finite mixture distribution approach for characterizing tree diameter distributions by natural social class in pure even-aged Scots pine stands in Poland",
 "Distributing FIA inventory information onto segmented Landsat Thematic Mapper images stratified with industrial ground data",
 "Proceedings of the Fourth Annual Forest Inventory and Analysis Symposium: Meeting Jointly with the Southern Forest Mensurationists: New Orleans, Louisiana, November 19-21, 2002",
 "Distributing FIA Information onto Segmented Landsat Thematic Mapper Images Stratified",
 # page 2 (2005 -> 1988)
 "A new flexible GADA based dynamic site equation with polymorphism and variable asymptotes",
 "Use of Semivariances for Studies of Landsat TM Image Textural Properties of Loblolly Pine Forests",
 "Impact of Definitions of FIA Variables and Compilation Procedures on Inventory Compilation Results in Georgia",
 "Comparison of programs used for FIA inventory information dissemination and spatial representation",
 "Self-referencing taper curves for loblolly pine",

"Analysis of Pooled FIA and Remote Sensing Data for Fiber Supply Assessment",
 "Semivariogram analysis of Landsat 5 TM textural data for loblolly pine forests",
 "A three-step approach to model tree mortality in the State of Georgia",
 "Using FIA and GIS data to estimate areas and volumes of potential stream management zones",
 "Impact of stream management zones and road beautifying buffers on long-term fiber supply in Georgia",
 "Distributing FIA information onto segmented Landsat Thematic Mapper images",
 "Proof of Concept for an Approach to a Finer Resolution Inventory",
 "One- and Two-point Principle Dynamic Site Equations from Richards, Weibull exponential functions",
 "Second International Conference on Forest Measurements and Quantitative Methods and Management",
 "Impact of adjacency and green up constraints on regional wood supply",
 "2003 Subject and Author Index",
 "Spatially explicit sustainability analysis of long-term fiber supply in Georgia, USA",
 "Intensive forest management in the US South",
 "Use of geostatistical methods for classification of forest ecosystems using satellite imagery",
 "Using a camera as an angle gauge in angle-count sampling",
 "The use of geostatistical methods for remote-sensing based determination of inventory measures",
 "Selected problems in spatially explicit sustainability analysis of forest resources in Georgia USA",
 "Carbon Sequestration-A Georgia Fact Sheet-CFB# 13",
 "GADA derivation of dynamic site equations with polymorphism and variable asymptotes",
 "Intensywna gospodarka lesna na Poludniu Stanow Zjednoczonych Ameryki Polnocnej",
 "Zastosowanie metod geostatystycznych do wyznaczania cech taksacyjnych i parametrów biofizycznych lasów",
 "Wykorzystanie metod geostatystycznych do klasyfikacji ekosystemów leśnych",
 "Three methods for avoiding impacts of incompatible site index and height prediction models",
 "Developing a well-behaved dynamic site equation using a modified Hossfeld IV function",
 "Growth and yield models for second rotation loblolly pine plantations in the piedmont",
 "Semivariances of Landsat TM satellite imagery texture of forested areas",
 "Wyprowadzanie ogólnych dynamicznych równań bonitacyjnych za pomocą uniwersalnej metody różnic",
 "A simple method for derivation of biologically based implicit growth and yield equations",
 "Semiwariacje tekstury obrazów satelitarnych Landsat TM obszarów leśnych",
 "Site index model for Scots pine based on Szymkiewicz yield tables",
 "Derivation of Generic Dynamic Site Equations Using Generalized Algebraic Difference Approach",
 "Impact of intensive management practices on long-term simulations of forest resources in Georgia",
 "Comparing fixed-and variable-base-age site equations having single versus multiple asymptotes",
 "Long-term sustainability analysis of fiber supply in Georgia",
 "ON LONG-TERM SIMULATIONS OF FOREST",
 "Generalized Method of Algebraic Differences deriving dynamic site index equations",
 "A dynamic form of the anamorphic site index model for Scots pine in Poland",
 "Uniwersalna Metoda Różnic Algebraicznych: uogólnione wyprowadzanie dynamicznych równań",
 "Updated Base Age Invariant Height Growth Models for Loblolly Pine",
 "Uniwersalna metoda różnic algebraicznych: uogólnione wyprowadzanie dynamicznych równań",
 "Dynamiczna forma anamorficznego modelu bonitacyjnego dla sosny pospolitej w Polsce",
 "A Dynamic Equation for a Sitka Spruce Height- Age Model",
 "Proceedings of the fourth annual Forest Inventory and Analysis symposium",
 "Various aspects of sustainability analysis in Georgia",
 "Selected technical aspects of a statewide forest resource modeling project in Georgia",
 "Inventory Updates and Related Topics-From Imputations to Mixed Models",
 "Solutions for reducing bias in inventory projections when using incompatible site index models",
 "Forest inventory in the United States of America",
 "January/janvier",
 "Revue canadienne de recherche forestière",
 "Wielkoobszarowa inwentaryzacja lasów w Stanach Zjednoczonych Ameryki Polnocnej",
 "Long-term sustainability of fiber supply in Georgia-a preliminary approach",
 "Forest Inventory-Introduction",
 "Long-term sustainability analysis of wood supply in Georgia-Preliminary Approach",
 "Three methods of deriving advanced dynamic site equations on inland Douglas-fir site curves",
 "Development of a well-behaved site-index equation: jack pine in north-central Ontario",
 "Analytical site index solution for the generalized log-logistic height equation",
 "A Method for Deriving Theory Based Base Age Invariant Models",
 "Practical methods for estimating non-biased parameters in self-referencing growth and yield models",
 "Generalized algebraic difference approach: theory based derivation of dynamic site equations",
 "Using 30-meter satellite data to estimate basal areas of intensively managed loblolly pine stands",
 "Generalized Algebraic Difference Approach: Theory Based Derivation of",
 "Proceedings...: Chris J. Cieszewski",
 "A comparison of sample unit designs in the national inventory of the US",
 "Base-age invariance and inventory projections",
 "Examples of practical methods for unbiased parameter estimation in self-referencing functions",
 "An implementation of new dynamic site index models in a company's timber supply analysis",
 "e-s8} j BP s Y 8s \${sjjk VIs} \$ t 3\$-IT sR TV BI jR B zs \$ sAj T sR TV yt% t B k _ B{\$ BtR",
 "The Algebraic Difference Approach Improves Fixed Base-age Site Index Models",
 "An Example of Mathematical Adaptation of Base Age Specific Models",
 "qbB V}} Bs {YRB y8} B% \$ t yt% t B k _ B{\$ Bt eJZs \$ BtR Bt e-s8} j R BP _Zaj \$ RY I BI jR",
 "Three Approaches to Derivation of Inventory Projection Equations",
 "Improving Inventory Projections",
 "First International Conference on Measurements and Quantitative Methods and Management",
 "Implementation of a base age invariant height model for lodgepole pine",
 "The method for deriving theory-based base-age invariant polymorphic site equations",
 "Estimating forest product yield with Monte Carlo integration",
 "Biomass estimation for temperate broadleaf forests of the United States using inventory data",
 "A multiproduct growth and yield model for the circumboreal aspens",
 "Statistical analysis of error propagation in national level carbon budgets",
 "Adjusting lodgepole pine site index for density related height growth reduction",

```

"Stockability, growth, and yield of the circumboreal aspens",
"Development of a variable density height-growth-model through defining multidimensional spaces",
"Modeling density-related lodgepole pine height growth, using Czarnowski's stand dynamics theory",
"Preliminary site-index height growth curves for eleven timber species in Saskatchewan",
"Predicting density-related lodgepole pine height growth in Alberta for thinning applications",
"Predicting density-related lodgepole pine height growth in Alberta for thinning applications",
"Towards optimal design of nonlinear regression models",
"Grand Rapids, Minnesota",
"Modeling growth and yield of aspen in western Canada",
"Polymorphic height and site index curves for the major tree species in Alberta",
"Forestry Canada Edmonton, Alberta",
"Modeling growth and yield of aspen in western Canada symposium proceedings",
"Polymorphic height and site index curves for lodgepole pine in Alberta",
"The stand dynamics of lodgepole pine",
# page 3 (201-211)
"The Stand Dynamics of Lodgepole Pine",
"Note of Appreciation",
"Self-thinning and Stockability of the Circumboreal Aspens (Populus tremuloides Michx., and P. tremula L.)",
"Framework for Analysis of Error Propagation in National Carbon Budgets",
"Wielkoobszarowa inwentaryzacja lasów w Stanach Zjednoczonych Ameryki Północnej Forest inventory in the United States of America",
"Three Methods for Avoiding the Impacts of Incompatible Site Index and Height Prediction Models Demonstrated on Jack Pine Curves for Ontario",
"Recherche forestière",
"Michx., P. tremula L.",
"Paul Schroeder, Sandra Brown, Jiangming Mo, Richard Birdsey, and Chris Cieszewski",
"Geostatistical approach to mortality pattern analysis using remote sensing and large-scale inventory ground measurements",
"Site index modeling in Poland: its history and current directions",
]

def strip_diac(s):
    return "".join(c for c in unicodedata.normalize("NFKD", s) if not unicodedata.combining(c))

def norm(t):
    t = strip_diac(t).lower()
    t = t.replace("u.s.", "us").replace("r-square", "r2").replace("r²", "r2").replace("r2 ", "r2 ")
    t = re.sub(r"^[a-z0-9 ]", " ", t)
    t = re.sub(r"\s+", " ", t).strip()
    return t

# --- artifacts: journal names, section headings, author-list fragments, OCR garbage, indexes ---
ARTIFACT_EXACT = {
    "January/janvier", "Revue canadienne de recherche forestière", "Recherche forestière",
    "Grand Rapids, Minnesota", "Forestry Canada Edmonton, Alberta", "2003 Subject and Author Index",
    "Proceedings...: Chris J. Cieszewski", "Note of Appreciation", "Michx., P. tremula L.",
    "Paul Schroeder, Sandra Brown, Jiangming Mo, Richard Birdsey, and Chris Cieszewski",
    "ON LONG-TERM SIMULATIONS OF FOREST", "Carbon Sequestration-A Georgia Fact Sheet-CFB# 13",
}

def is_ocr_garbage(t):
    letters = re.findall(r"[A-Za-z]", t)
    weird = re.findall(r"[{}$%]", t)
    return len(weird) >= 4 and len(weird) > 0.05*len(t)

# --- Polish-language translation versions of English papers (normalized) ---
POLISH_TRANSLATIONS = {norm(x) for x in [
    "Ocena wlasciwosci Drewna Brzozowego za Pomoca Spektroskopii NIR i SilviScan dla Probek z Polski USA i Smolenska",
    "Mikro-Detaliczna Analiza Porownawcza Obszaru Lesnego Przy Uzyciu Zdjec Satelitarnych Wysokiej Rozdzielczosci",
    "Intensywna gospodarka lesna na Poludniu Stanow Zjednoczonych Ameryki Polnocnej",
    "Zastosowanie metod geostatycznych do wyznaczania cech taksacyjnych i parametrow biofizycznych lasow",
    "Wykorzystanie metod geostatystycznych do klasyfikacji ekosystemow leśnych",
    "Wyprowadzanie ogólnych dynamicznych równań bonitacyjnych za pomocą uniwersalnej metody różnic",
    "Semiwariancje tekstury obrazów satelitarnych Landsat TM obszarów leśnych",
    "Uniwersalna Metoda Różnic Algebraicznych: uogólnione wyprowadzanie dynamicznych równań",
    "Uniwersalna metoda różnic algebraicznych: uogólnione wyprowadzanie dynamicznych równań",
    "Dynamiczna forma anamorficznego modelu bonitacyjnego dla sosny pospolitej w Polsce",
    "Wielkoobszarowa inwentaryzacja lasow w Stanach Zjednoczonych Ameryki Polnocnej",
]}

raw = len(ENTRIES)
artifacts, kept = [], []
for t in ENTRIES:
    if t in ARTIFACT_EXACT or is_ocr_garbage(t):
        artifacts.append(t)
    else:
        kept.append(t)

# collapse exact-normalized duplicates
seen, uniq, exact_dups = {}, [], []

```

```

for t in kept:
    n = norm(t)
    if n in seen:
        exact_dups.append(t)
    else:
        seen[n] = t; uniq.append(t)

# among the unique-normalized set, count Polish translation versions
polish = [t for t in uniq if norm(t) in POLISH_TRANSLATIONS]
uniq_no_pol = [t for t in uniq if norm(t) not in POLISH_TRANSLATIONS]

print(f"RAW entries:                {raw}")
print(f" - non-article artifacts:    {len(artifacts)}")
print(f" - exact/near duplicates:      {len(exact_dups)}")
print(f"= unique titles (incl. Polish): {len(uniq)}")
print(f" - Polish translation versions: {len(polish)}")
print(f"= unique works (English/merged): {len(uniq_no_pol)}")
print("\n--- Handbook occurrences:", sum(1 for t in ENTRIES if norm(t)==norm('Handbook of land and tree
measurements'))))
print("\nArtifacts removed:")
for a in artifacts: print("    ", a[:70])
print("\nExact/near duplicates removed:")
for a in exact_dups: print("    ", a[:70])
print("\nPolish translation versions (of English papers):")
for a in polish: print("    ", a[:70])

```

28. diag_am_aq.py (SHA-256 58b930c09939c1d6...; 46 lines)

```

# -*- coding: utf-8 -*-
import openpyxl, pandas as pd
from openpyxl.utils import get_column_letter
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudex\c--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
FILES={"2020":"career_2020.xlsx", "2021":"career_2021.xlsx", "2022":"career_2022.xlsx", "2023":"career_2023.xlsx", "2024":"career_2024.xlsx"}

# 1) column-letter map for the 2024 raw source file
wb=openpyxl.load_workbook(f"{SCR}/career_2024.xlsx",read_only=True)
ws=wb["Data"]; hdr=[c for c in next(ws.iter_rows(max_row=1,values_only=True))]
print("=== 2024 raw source column letters for sm-* headers ===")
for i,h in enumerate(hdr):
    if isinstance(h,str) and h.startswith("sm-"):
        print(f"    {get_column_letter(i+1)} = {h}")
wb.close()

# 2) across all raw files: distinct sm-field values, and the three counts
allfields=set()
tot={"AM_forestry":0,"AO_forestry":0,"AQ_forestry":0,"AM_or_AQ":0,"AM_or_AO":0,"field_AFF":0,"rows":0}
peryear={}
for y,fn in FILES.items():
    df=pd.read_excel(f"{SCR}/{fn}",sheet_name="Data",
                    usecols=["sm-subfield-1","sm-subfield-2","sm-field"],dtype=str)
    s1=df["sm-subfield-1"].str.strip(); s2=df["sm-subfield-2"].str.strip(); fd=df["sm-field"].str.strip()
    allfields|=set(fd.dropna().unique())
    am=(s1=="Forestry"); ao=(s2=="Forestry"); aq=(fd=="Forestry")
    aff=(fd=="Agriculture, Fisheries & Forestry")
    c={"AM_forestry":int(am.sum()),"AO_forestry":int(ao.sum()),"AQ_forestry":int(aq.sum()),
      "AM_or_AQ":int((am|aq).sum()),"AM_or_AO":int((am|ao).sum()),"field_AFF":int(aff.sum()),"rows":len(df)}
    peryear[y]=c
    for k in tot: tot[k]+=c[k]
    print(f"{y}: AM=Forestry {c['AM_forestry']}, AO=Forestry {c['AO_forestry']}, AQ(sm-field)=Forestry {c['AQ_forestry']}, "
          f"AM|AQ {c['AM_or_AQ']}, AM|AO {c['AM_or_AO']}, sm-field=Agri/Fish/Forestry {c['field_AFF']}")

print("\n=== POOLED 2020-2024 ===")
for k,v in tot.items(): print(f"    {k}: {v},")

print("\n=== Is 'Forestry' ever a value of sm-field (AQ)? ===")
print("    'Forestry' in distinct sm-field values:", "Forestry" in allfields)
print("    distinct sm-field values (count):", len(allfields))
print("    sm-field values containing 'Forest':", sorted(f for f in allfields if "forest" in f.lower()))

# 3) for AM==Forestry rows, what sm-field do they carry?
df24=pd.read_excel(f"{SCR}/career_2024.xlsx",sheet_name="Data",usecols=["sm-subfield-1","sm-field"],dtype=str)
sub=df24[df24["sm-subfield-1"].str.strip()=="Forestry"]
print("\n=== 2024: sm-field values among AM(sm-subfield-1)=Forestry rows ===")

```

```
print(sub["sm-field"].str.strip().value_counts().to_dict())
```

29. diag_singleauthor.py (SHA-256 eb6c5ba3ace18456...; 29 lines)

```
# -*- coding: utf-8 -*-
import pandas as pd, numpy as np
CAREER=r"C:\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
df=pd.read_excel(CAREER,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
df=df[(s1=="Forestry")|((s1=="Ecology")&(s2=="Forestry"))].copy()
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
df["group"]=np.where(s1=="Forestry","Forestry","Ecology")
for col in ["np","nps","nc","ncs"]:
    df[col]=pd.to_numeric(df[col],errors="coerce").fillna(0)
print("dtypes:",{c:str(df[c].dtype) for c in ["np","nps","nc","ncs"]})
print("nps stats:
min",df.nps.min(),"max",df.nps.max(),"mean",round(df.nps.mean(),1),"zeros",int((df.nps==0).sum()),"of",len(df))
for g in ["Forestry","Ecology"]:
    d=df[df.group==g]
    dsa=d[d.nps>0]
    print(f"\n=== {g} (criterion) : {len(d)} author-year records; with nps>0: {len(dsa)}
    ({100*len(dsa)/len(d):.0f}%) ===")
    print(" by year (all / with nps>0 / sum nps / sum ncs):")
    for y in YEARS:
        dy=d[d.year==y]; dyy=dy[dy.nps>0]
        print(f" {y}: n={len(dy):>4} nps>0={len(dyy):>4} Enps={int(dy.nps.sum()):>6}
Enncs={int(dy.ncs.sum()):>9}")
    print(" distinct scientists (by authfull) with nps>0, 2024:",dsa[dsa.year==2024]["authfull"].nunique())
# top single-author countries/insts quick peek (2024)
from cc_defs import CC
df["country"]=df["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
for g in ["Forestry","Ecology"]:
    d=df[(df.group==g)&(df.nps>0)&(df.year==2024)]
    print(f"\n{g} single-author top-6 countries by Enps (2024):")
    print(d.groupby("country").nps.sum().sort_values(ascending=False).head(6).to_string())
```

30. distinct.py (SHA-256 6fbde7604abab18e...; 36 lines)

```
import openpyxl
from collections import Counter

path = "career_2024.xlsx"
wb = openpyxl.load_workbook(path, read_only=True)
ws = wb["Data"]
it = ws.iter_rows(values_only=True)
header = [str(c).strip() for c in next(it)]
i1 = header.index("sm-subfield-1")
i2 = header.index("sm-subfield-2")
ifld = header.index("sm-field")

subs = Counter()
fields = Counter()
for row in it:
    if row is None:
        continue
    if row[i1]:
        subs[str(row[i1]).strip()] += 1
    if row[i2]:
        subs[str(row[i2]).strip()] += 1
    if row[ifld]:
        fields[str(row[ifld]).strip()] += 1
wb.close()

print("=== distinct sm-field values ===")
for k in sorted(fields):
    print(f"{fields[k]:>8} {k}")

print("\n=== subfields matching key terms ===")
for k in sorted(subs):
    kl = k.lower()
    if any(t in kl for t in ("ecolog", "forest", "natural resource", "resource manag", "environ", "conserv",
"wildlife", "fish")):
        print(f"{subs[k]:>8} {k}")
```

```
print(f"\ntotal distinct subfields: {len(subs)}")
```

31. dump_sheets.py (SHA-256 1ab17ab9d0c16d48...; 18 lines)

```
from openpyxl import load_workbook
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
SHEETS = ["SA-B Forestry countries", "SA-B Forestry institutions",
          "SA-C Ecology countries", "SA-C Ecology institutions"]
wb = load_workbook(WB, data_only=True)
for s in SHEETS:
    ws = wb[s]
    print("\n" + "="*70)
    print(f"SHEET: {s}  dims={ws.dimensions}  max_row={ws.max_row}  max_col={ws.max_column}")
    print("="*70)
    for r in range(1, ws.max_row+1):
        row = []
        for c in range(1, min(ws.max_column, 9)+1):
            v = ws.cell(row=r, column=c).value
            row.append(" " if v is None else str(v))
        # only print non-empty rows
        if any(x != " " for x in row):
            print(f"r{r:>3}: " + " | ".join(row))
```

32. explore2.py (SHA-256 58dc73a6c8465dd5...; 26 lines)

```
import pandas as pd, json
p = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df = pd.read_excel(p, sheet_name="Data")
print("shape", df.shape)
print("cols", list(df.columns)[:12])
print("years", sorted(df["year"].unique()))
print("distinct cntry codes:", df["cntry"].nunique())
print(sorted(df["cntry"].dropna().unique().tolist()))
print("\nrows with blank cntry:", df["cntry"].isna().sum())
print("rows with blank inst_name:", df["inst_name"].isna().sum())
print("\nmismatched subfield values:")
print(df["matched_subfield"].value_counts().to_dict())
print("\nsm-field values (should be Ag/Fish/Forestry + Biology):")
print(df["sm-field"].value_counts().to_dict())
d24 = df[df.year==2024]
print("\n2024 rows:", len(d24))
us24 = d24[d24.cntry=="usa"]
print("2024 USA rows:", len(us24), "distinct USA institutions:", us24["inst_name"].nunique())
print("distinct institutions overall (2024):", d24["inst_name"].nunique())
# dump distinct US institutions with counts for state-mapping workflow
us_all = df[df.cntry=="usa"]
inst_counts = us_all.groupby("inst_name").size().sort_values(ascending=False)
print("\ndistinct USA institutions (all years):", len(inst_counts))
inst_counts.to_csv(r"C:\Users\ADMINI~1\AppData\Local\Temp\Claude\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\us_institutions.csv", header=["n_rows_allyears"])
print("top 15 US institutions:")
print(inst_counts.head(15).to_string())
```

33. explore_forestry.py (SHA-256 ec1739fde360a560...; 30 lines)

```
# -*- coding: utf-8 -*-
import pandas as pd
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df = pd.read_excel(CAREER, sheet_name="Data")
YEARS=[2020,2021,2022,2023,2024]
sf1=df["sm-subfield-1"].astype(str).str.strip(); sf2=df["sm-subfield-2"].astype(str).str.strip()
df["is_forestry"]=(sf1=="Forestry")|(sf2=="Forestry")
fo=df[df.is_forestry].copy()
print("Forestry cohort rows:", len(fo), "of", len(df))
print("per year:", {y:int((fo.year==y).sum()) for y in YEARS})
print("pubs 2024 (k):", round(fo[fo.year==2024]["np"].sum()/1000,1), "| cites 2024 (M):", round(fo[fo.year==2024]
["nc"].sum()/1e6,2))

# the OTHER subfield among forestry authors (the non-Forestry one)
def other(r):
    a=str(r["sm-subfield-1"]).strip(); b=str(r["sm-subfield-2"]).strip()
    if a=="Forestry": return b
    if b=="Forestry": return a
    return "(none)"
fo["other"]=fo.apply(other,axis=1)
```

```

print("\nTop secondary subfields among Forestry authors (all yrs):")
print(fo["other"].value_counts().head(25).to_string())

# distinct subfields present in forestry cohort (both slots)
pres=sorted(set(fo["sm-subfield-1"].astype(str).str.strip())|set(fo["sm-subfield-2"].astype(str).str.strip()))
pres=[p for p in pres if p and p!="nan"]
print("\ndistinct subfields present in Forestry cohort:", len(pres))

# check requested quant additions
for probe in ["Operations Research","Computation Theory & Mathematics","Numerical & Computational
Mathematics","Software Engineering","Distributed Computing","Statistics & Probability","Geological & Geomatics
Engineering","Artificial Intelligence & Image Processing"]:
    print(f"    present? {probe!r}:", probe in pres)

```

34. figs_auth6.py (SHA-256 f82a3bdb8a5a54e4...; 255 lines)

```

# -*- coding: utf-8 -*-
# Figures for the NINE-number scheme: (solo, senior, all) x (pubs/author, cites/author, cites/pub).
# Data-section figures: ff1_change (per-year COUNT trend), ff10_usmaps (pub-count choropleths).
# Results 3x3 grids: ff_ratio_group / _continent / _region / _institution ; ff9_worldmaps (9 maps);
ff_ratio_scatter.
# Authorship ($4.6, no last): ff_authcomp, ff_authrank. Discussion: ff_refs (solo/senior/all placements).
import json, numpy as np
import matplotlib; matplotlib.use("Agg")
import matplotlib.pyplot as plt
import matplotlib.patches as mp
from matplotlib.collections import PolyCollection
from matplotlib.colors import BoundaryNorm
import matplotlib.cm as cm
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
plt.rcParams.update({"font.family":"Arial","figure.dpi":200,"font.size":12,"axes.titlesize":13,
"axes.labelsize":12,"xtick.labelsize":11,"ytick.labelsize":11,"legend.fontsize":11})
YEARS=[2020,2021,2022,2023,2024]
GREEN="#2F5233"; SOLO="#2F5233"; SEN="#2C6E9B"; ALLC="#C6902F"; GREY="#9AA0A6"; GOLD="#C6902F"; TEAL="#2C6E9B"
A6=json.load(open(f"{SCR}/auth6.json",encoding="utf-8"))
AU=json.load(open(f"{SCR}/authorship.json",encoding="utf-8"))
# classes (rows) and metrics (columns) of the 3x3
CLASSES=[("solo","Solo",SOLO),("sen","Senior",SEN),("all","All",ALLC)]
METRICS=[("ppa","Pubs / author"),("cpa","Cites / author"),("cpp","Cites / pub")]
def kf(cls,met): return f"{cls}_{met}"
def cfmt(met,v):
    if met=="cpa": return f"{v/1000:.1f}k" if v>=1000 else f"{v:.0f}"
    if met=="ppa": return f"{v:.0f}" if v>=10 else f"{v:.1f}"
    return f"{v:.0f}"

def _rshorten(s):
    s=str(s)
    for a,b in [ ("University","Univ."), ("Institute","Inst."), ("Technology","Tech."),
        ("Agricultural Sciences","Agric. Sci."), ("Sciences","Sci."), ("Northern America","N. America"),
        ("Australia & New Zealand","Australia & NZ"), ("South America","S. America") ]:
        s=s.replace(a,b)
    return s if len(s)<=34 else s[:33]+"..."

# ----- Data-section: per-year COUNT trend -----
PSPEC=[("solo_p","Solo papers",SOLO),("sen_p","Senior papers",SEN),("all_p","All papers",ALLC)]
def fig_change(fname,title):
    by=A6["cohort_by"]
    fig,ax=plt.subplots(figsize=(8.6,4.6))
    for key,short,col in PSPEC:
        v=by[key]; pc=[100*(x/v[0]-1) for x in v]
        ax.plot(YEARS,pc,marker="o",ms=7,lw=2.6,color=col,label=f"{short} ({v[0]:.0f}→{v[-1]:.0f})",zorder=3)
        ax.annotate(f"{v[-1]:.0f}",(YEARS[-1],pc[-1]),textcoords="offset
points",xytext=(6,0),ha="left",fontSize=10.5,color=col,fontweight="bold")
        ax.set_xticks(YEARS); ax.set_ylabel("Change since 2020 (%)",fontSize=12.5)
        ax.axhline(0,color="#bbb",lw=0.9); ax.spines[["top","right"]].set_visible(False)
        ax.legend(fontsize=11,loc="upper left",frameon=False)
        ax.set_title(title,color=GREEN,fontweight="bold",fontSize=14,pad=8)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- Results 3x3: two GROUPS (grouped vertical bars per cell) -----
def fig_group3x3(fname,title):
    groups=["Forestry","Ecology"]; gcol={"Forestry":GOLD,"Ecology":TEAL}
    fig,axs=plt.subplots(3,3,figsize=(10.8,9.2))
    for ri,(cls,clab,ccol) in enumerate(CLASSES):
        for ci,(met,mlab) in enumerate(METRICS):
            ax=axs[ri,ci]; vals=[A6["group"][g][kf(cls,met)] for g in groups]

```

```

        ax.bar([0,1],vals,color=[gcol[g] for g in groups],width=0.62)
        for xi,v in zip([0,1],vals):
            ax.annotate(cfmt(met,v),(xi,v),textcoords="offset points",xytext=(0,3),ha="center",va="bottom",
                        fontsize=11,fontweight="bold",color="#333")
        ax.set_xticks([0,1]); ax.set_xticklabels(groups,fontsize=10.5)
        ax.set_ylim(0,max(vals)*1.20); ax.spines[["top","right"]].set_visible(False);
ax.tick_params(axis="y",labelsize=9)
        if ri==0: ax.set_title(mlab,fontsize=12.5,color="#333",fontweight="bold",pad=8)
        if ci==0: ax.set_ylabel(clab,fontsize=13,color=ccol,fontweight="bold")
fig.suptitle(title,fontsize=15,color=GREEN,fontweight="bold",y=1.005)
fig.tight_layout(rect=[0,0,1,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- Results 3x3: a DIVISION (thin horizontal bars across categories per cell) -----
def fig_div3x3(members,fname,title,n=None,foldcountry=False):
    ms=members[:n] if n else members
    ms=ms[:-1]
    labels=[_rshorten(m["key"]+(f' ({m["country"]})' if foldcountry and m.get("country") else "")) for m in ms]
    yy=np.arange(len(ms))
    fig,axs=plt.subplots(3,3,figsize=(11.4,max(6.0,0.34*len(ms)*3+2.2)),sharey=True)
    for ri,(cls,clab,ccol) in enumerate(CLASSES):
        for ci,(met,mlab) in enumerate(METRICS):
            ax=axs[ri,ci]; vals=[m[kf(cls,met)] for m in ms]
            ax.barh(yy,vals,color=ccol,height=0.7,alpha=0.92)
            ax.set_yticks(yy)
            if ci==0: ax.set_yticklabels(labels,fontsize=10)
            ax.tick_params(axis="y",length=0,pad=2); ax.tick_params(axis="x",labelsize=9)
            for yi,v in zip(yy,vals): ax.annotate(" "+cfmt(met,v),
            (v,yi),va="center",ha="left",fontsize=8.5,color="#333")
            ax.set_xlim(0,max(vals)*1.22); ax.spines[["top","right"]].set_visible(False)
            if ri==0: ax.set_title(mlab,fontsize=12.5,color="#333",fontweight="bold",pad=8)
            if ci==0: ax.set_ylabel(clab,fontsize=13,color=ccol,fontweight="bold")
fig.suptitle(title,fontsize=15,color=GREEN,fontweight="bold",y=1.002)
fig.tight_layout(rect=[0,0,1,0.99]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- Results: country scatter (solo vs all impact) -----
def fig_scatter(fname,title,min_sci_yr=5.0):
    pts=[m for m in A6["div"] if m["sci_yr"]>=min_sci_yr and m["all_p"]>0 and m["solo_p"]>0]
    x=np.array([m["all_cpp"] for m in pts]); y=np.array([m["solo_cpp"] for m in pts])
    sci=np.array([m["sci_yr"] for m in pts])
    fig,ax=plt.subplots(figsize=(8.8,6.4))
    lo=min(x.min(),y.min())-6; hi=max(x.max(),y.max())+8
    ax.plot([lo,hi],[lo,hi],ls="--",color="#888",lw=1.1,zorder=1)
    area=90+(sci/sci.max())*2100
    ax.scatter(x,y,s=area,c="#4C9F70",alpha=0.50,edgecolor="#2F5233",linewidth=1.0,zorder=3)
    NUDGE={"United States":(0,15),"China":(0,-15),"Canada":(20,-12),"Finland":(-12,12),"Sweden":(-20,-12),
            "Germany":(18,-6),"Australia":(24,-2),"France":(0,13),"Spain":(-14,-12),"Japan":(0,-14),
            "Netherlands":(0,14),"Brazil":(0,13),"Italy":(-14,-11),"United Kingdom":(0,12),"Norway":(-8,11),
            "Austria":(22,3),"Switzerland":(18,9),"Portugal":(2,-14),"Denmark":(-6,11),"New Zealand":(0,-14)}
    for m in pts:
        dx,dy=NUDGE.get(m["key"],(0,11))
        ha="left" if dx>2 else ("right" if dx<-2 else "center"); va="bottom" if dy>0 else "top"
        ax.annotate(m["key"],(m["all_cpp"],m["solo_cpp"]),textcoords="offset points",xytext=(dx,dy),ha=ha,va=va,
                    fontsize=10.5,color="#222",fontweight=("bold" if m["sci_yr"]>=25 else "normal"),zorder=5,
                    bbox=dict(boxstyle="round,pad=0.10",fc="white",ec="none",alpha=0.70))
    ax.set_xlim(lo,hi); ax.set_ylim(lo,hi)
    ax.set_xlabel("Citations per paper – all publications",fontsize=13)
    ax.set_ylabel("Citations per solo paper",fontsize=13)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=14,pad=10)
    ax.spines[["top","right"]].set_visible(False)
    ax.text(0.015,0.975,"above the line: solo papers out-cite the country's average
paper",transform=ax.transAxes,
            fontsize=9.5,color="#2F5233",ha="left",va="top",style="italic")
    ax.text(0.985,0.06,"dashed line: equal solo and overall impact",transform=ax.transAxes,fontsize=9.5,
            color="#888",ha="right",va="bottom",style="italic")
    ax.text(0.985,0.975,"bubble area scaled to no. of
scientists",transform=ax.transAxes,fontsize=9.5,color="#888",ha="right",va="top")
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- maps -----
def geoms(f):
    g=f["geometry"]
    if not g: return []
    return [g["coordinates"]] if g["type"]=="Polygon" else (g["coordinates"] if g["type"]=="MultiPolygon" else
[])
def nice_bins(vals):
    v=sorted(x for x in vals if x>0)
    if not v: return [0,1]
    qs=np.quantile(v,[0,.2,.4,.6,.8,.95,1.0]); b=sorted(set(int(round(x)) for x in qs))
    if len(b)<3: b=[0,max(1,int(v[-1]))]
    if b[0]>0: b=[0]+b

```

```

return b
def draw_panel(ax, feats, keyfn, values, title, xlim=None, ylim=None, fscale=1.0):
    bins=nice_bins(list(values.values()))
    if len(bins)<3: bins=[0,1,2]
    cmap=plt.get_cmap("YlGn",len(bins)-1); norm=BoundaryNorm(bins,cmap.N)
    bg,fg,fc=[[],[],[]]
    for f in feats:
        k=keyfn(f); v=values.get(k,0)
        for poly in geoms(f):
            ext=np.asarray(poly[0])
            if v and v>0: fg.append(ext); fc.append(cmap(norm(v)))
            else: bg.append(ext)
    ax.add_collection(PolyCollection(bg,facecolors="#EFEFEA",edgecolors="#BBB",linewidths=.2))
    if fg: ax.add_collection(PolyCollection(fg,facecolors=fc,edgecolors="#888",linewidths=.2))
    ax.autoscale()
    if xlim: ax.set_xlim(*xlim)
    if ylim: ax.set_ylim(*ylim)
    ax.axis("off"); ax.set_title(title,fontsize=10.5*fscale,color=GREEN,fontweight="bold")
    sm=cm.ScalarMappable(norm=norm,cmap=cmap); sm.set_array([])
    cb=plt.colorbar(sm,ax=ax,orientation="horizontal",fraction=0.05,pad=0.02,aspect=28)
    cb.set_tick_params(labelsize=8*fscale); cb.set_ticks(bins); cb.set_ticklabels([f"{int(b)}" for b in bins])
WORLD=json.load(open(f"{SCR}/world_110m.geojson",encoding="utf-8"))
WF=[f for f in WORLD["features"] if f["properties"].get("CONTINENT")!="Antarctica"]
def iso3(f):
    p=f["properties"]; a=p.get("ISO_A3","-99"); return p.get("ADM0_A3") if a in ("-99",None,"") else a
US=json.load(open(f"{SCR}/us_states.geojson",encoding="utf-8"))
UF=[f for f in US["features"] if f["properties"]["name"] not in ("Alaska","Hawaii","Puerto Rico")]
def fig_worldmaps3x3(fname):
    mc=A6["map_country"]
    fig,axs=plt.subplots(3,3,figsize=(13.2,9.6))
    for ri,(cls,clab,ccol) in enumerate(CLASSES):
        for ci,(met,mlab) in enumerate(METRICS):
            ax=axs[ri,ci]
            draw_panel(ax,WF,iso3,mc[kf(cls,met)],f"{clab} - {mlab}",ylim=(-60,85),fscale=0.98)
    fig.suptitle("World: (solo, senior, all) x (pubs/author, cites/author, cites/paper), 2020-2024",
        fontsize=14,color=GREEN,fontweight="bold",y=1.005)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)
def fig_usmaps(fname):
    mu=A6["map_us"]
    fig,axs=plt.subplots(1,3,figsize=(11.0,3.3))
    for ax,(key,short,col) in zip(axs,PSPEC):
        draw_panel(ax,UF,lambda f:f["properties"]["name"],mu[key],f"{short} (5-yr total)",xlim=(-125,-
66),ylim=(24,50),fscale=1.55)
    fig.suptitle("United States - solo, senior and all publications by
state",fontsize=10.5*1.55,color=GREEN,fontweight="bold",y=1.05)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- $4.6 authorship (no last): single / first / other -----
APOS=[("single","Single-authored",SOLO),("first","First-authored",SEN),("other","Other co-authored","#B8B8B8")]
def fig_authcomp(fname,title):
    groups=["Ecology","Forestry","cohort"]
    glabel={"cohort":"Whole cohort","Forestry":"Forestry group","Ecology":"Ecology group"}
    fig,ax=plt.subplots(figsize=(8.4,3.5))
    for gi,g in enumerate(groups):
        left=0
        for pos,lab,col in APOS:
            w=AU["share_c"][g][pos]
            ax.barh(gi,w,left=left,color=col,edgecolor="white",height=0.62)
            if w>=4.5:
                ax.annotate(f"{w:.0f}%",(left+w/2,gi),ha="center",va="center",fontsize=12,
                    color=("white" if pos!="other" else "#333"),fontweight="bold")
            left+=w
    ax.set_yticks(range(len(groups))); ax.set_yticklabels([glabel[g] for g in groups],fontsize=12.5)
    ax.set_xlim(0,100); ax.set_xlabel("Share of citations (%)",fontsize=12.5)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=14,pad=8)
    ax.spines[["top","right","left"]].set_visible(False); ax.tick_params(axis="y",length=0)
    handles=[mp.Patch(color=c,label=l) for _,l,c in APOS]
    ax.legend(handles=handles,loc="lower center",bbox_to_anchor=(0.5,-0.36),ncol=3,frameon=False,fontsize=11.5)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)
def fig_authrank(fname,title):
    rows=[("single + first ("lead" authorship)","lead: single+first (ncsf)",SEN),
        ("single only","single (ncs)","#A63D40")]
    corr=AU["corr"]; ov=AU["overlap"]
    y=np.arange(len(rows))[:-1]
    fig,ax=plt.subplots(figsize=(8.6,2.8))
    for yi,(lab,key,col) in zip(y,rows):
        v=corr[key]; ax.barh(yi,v,color=col,height=0.5)
        ax.annotate(f" ρ = {v:+.2f} ({ov[key]} of the top 25 overlap)",(max(v,0),yi),va="center",ha="left",
            fontsize=11.5,color="#333")
    ax.axvline(0,color="#888",lw=1)

```

```

ax.set_yticks(y); ax.set_yticklabels([r[0] for r in rows], fontsize=12)
ax.set_xlim(-0.2, 1.2); ax.set_xlabel("Spearman correlation with the undiscriminated (total-citation) ranking
of the 854 scientists", fontsize=11)
ax.set_title(title, color=GREEN, fontweight="bold", fontsize=13.5, pad=8)
ax.spines[["top", "right", "left"]].set_visible(False); ax.tick_params(axis="y", length=0)
fig.tight_layout(); fig.savefig(fname, dpi=200, bbox_inches="tight"); plt.close(fig)

# ----- Discussion: reference scientists, solo/senior/all placements -----
def fig_refs(fname, title):
    refs=[("Cieszewski, Chris J. (this author)", A6["cies"], GOLD),
          ("Bettinger, Pete", A6["bett"], TEAL),
          ("Peterson, Chris J. (ecology-forestry)", A6["peterson"], "#3E8E7E")]
    fig, ax=plt.subplots(figsize=(8.8, 5.8))
    STAGE=[("solo", "Solo", 90, "o"), ("sen", "Senior", 150, "s"), ("all", "All", 300, "*")]
    for name, m, col in refs:
        xs=[m[f"{c}_ppa"] for c, _, _ in STAGE]; ys=[m[f"{c}_cpp"] for c, _, _ in STAGE]
        ax.plot(xs, ys, "-", color=col, lw=1.6, alpha=0.55, zorder=2)
        for (c, clab, sz, mk), xv, yv in zip(STAGE, xs, ys):
            ax.scatter([xv], [yv], s=sz, c=col, marker=mk, edgecolor="#333", linewidth=0.8, zorder=4)
            ax.annotate(name.split(" ")[0], (xs[-1], ys[-1]), textcoords="offset points", xytext=(8, 4),
                        fontsize=11, fontweight="bold", color=col)
    # legend for the three stages (shape)
    for c, clab, sz, mk in STAGE:
        ax.scatter([], [], s=sz, c="#666", marker=mk, label=clab)
    ax.legend(title="authorship class", fontsize=10.5, title_fontsize=10.5, loc="upper right", frameon=False)
    ax.set_xlabel("Publications per author (career output)", fontsize=13)
    ax.set_ylabel("Citations per publication", fontsize=13)
    ax.set_title(title, color=GREEN, fontweight="bold", fontsize=14, pad=8)
    ax.spines[["top", "right"]].set_visible(False)
    ax.text(0.985, 0.02, "a line links each scientist's solo → senior → all positions", transform=ax.transAxes,
            fontsize=9.5, color="#888", ha="right", va="bottom", style="italic")
    fig.tight_layout(); fig.savefig(fname, dpi=200, bbox_inches="tight"); plt.close(fig)

fig_change(f"{SCR}/ff1_change.png", "Forestry cohort – growth of solo, senior and all publication counts since
2020")
fig_usmaps(f"{SCR}/ff10_usmaps.png")
fig_worldmaps3x3(f"{SCR}/ff9_worldmaps.png")
fig_group3x3(f"{SCR}/ff_ratio_group.png", "Forestry vs Ecology group – the nine measures")
fig_div3x3(A6["div"] ["continent"], f"{SCR}/ff_ratio_continent.png", "By continent – the nine measures")
fig_div3x3(A6["div"] ["region"], f"{SCR}/ff_ratio_region.png", "By world sub-region (top 8 by output) – the nine
measures", n=8)
fig_div3x3(A6["div"] ["university"], f"{SCR}/ff_ratio_institution.png", "By institution (top 10 by output) – the
nine measures", n=10, foldcountry=True)
fig_scatter(f"{SCR}/ff_ratio_scatter.png", "Solo-paper impact vs all-paper impact, by country")
fig_authcomp(f"{SCR}/ff_authcomp.png", "Where the cohort's citations come from, by authorship position")
fig_authrank(f"{SCR}/ff_authrank.png", "How the total-citation ranking relates to lead- and solo-authored
rankings")
fig_refs(f"{SCR}/ff_refs.png", "Reference scientists: solo, senior and all placements")
print("regenerated figures (nine-number + 3x3 grids + maps + authorship + refs)")

```

35. find_johnnash.py (SHA-256 ccc33af686c877dc...; 18 lines)

```

# -*- coding: utf-8 -*-
# Is John Forbes Nash Jr. (game theory / Mathematics / Economics) anywhere in the Stanford list?
import pandas as pd
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
found=False
for y in [2020, 2021, 2022, 2023, 2024]:
    df=pd.read_excel(f"{SCR}/career_{y}.xlsx", sheet_name="Data")
    m=df[df["authfull"].astype(str).str.contains("Nash", case=False, na=False)]
    # any Nash in Mathematics or Economics fields?
    me=m[m["sm-field"].astype(str).str.contains("Mathematics|Economics", case=False, na=False)]
    print(f"== {y}: {len(m)} Nash rows; {len(me)} in Math/Econ ==")
    for _, r in me.iterrows():
        print(f" {r['authfull'][:26]} | {r['sm-subfield-1']} / {r['sm-field']} | {str(r['inst_name'][:34])}")
    # explicit John / J F
    jf=m[m["authfull"].astype(str).str.contains(r"Nash, \s*John|Nash, \s*J\.? \s*", case=True, na=False, regex=True)]
    for _, r in jf.iterrows():
        print(f" [J/F] {r['authfull'][:22]} | {r['sm-subfield-1']} / {r['sm-field']} | {str(r['inst_name']
[:30])}")
print("\nDONE")

```

36. find_kahneman.py (SHA-256 a7da8bc8f9f2f8b7...; 13 lines)

```

# -*- coding: utf-8 -*-
import pandas as pd, re

```

```

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
for y in [2024,2020]:
    d=pd.read_excel(f"{SCR}/career_{y}.xlsx",sheet_name="Data")
    m=d[d["authfull"].astype(str).str.contains("Kahneman",case=False,na=False)]
    npcol=[c for c in d.columns if re.fullmatch(r"np\d+",str(c))][0]
    nccol=[c for c in d.columns if re.fullmatch(r"nc\d+",str(c)) and "(ns)" not in str(c)][0]
    hcol=[c for c in d.columns if re.fullmatch(r"h\d+",str(c)) and "(ns)" not in str(c)][0]
    print(f"== career_{y}: {len(m)} Kahneman row(s) ==")
    for _,r in m.iterrows():
        print(f"    {r['authfull']} | {r['inst_name']} ({r['cntry']}) | AM={r['sm-subfield-1']} / AO={r['sm-subfield-2']} / {r['sm-field']}")
        print(f"        np={r[npcol]} nc={r[nccol]} h={r[hcol]} rank={r['rank']} c={r['c']:.3f} sfrank={r['rank sm-subfield-1']}/{r['sm-subfield-1 count']} first/last yr={r['firstyr']}-{r['lastyr']} self%={r['self%']}")

```

37. find_nash.py (SHA-256 113e1197a1fb1f6a...; 13 lines)

```

# -*- coding: utf-8 -*-
# Find "Nash, John" in the full career database (all ranked scientists, not just forestry cohort).
import pandas as pd
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
cols=["authfull","inst_name","cntry","np","nc","firstyr","lastyr","sm-subfield-1","sm-subfield-2","sm-field","rank","c","rank sm-subfield-1","sm-subfield-1 count","self%"]
for y in [2024,2020]:
    print(f"==== career_{y}.xlsx =====")
    df=pd.read_excel(f"{SCR}/career_{y}.xlsx",sheet_name="Data")
    hits=df[df["authfull"].astype(str).str.contains(r"^Nash",case=False,na=False,regex=True)]
    use=[c for c in cols if c in hits.columns]
    with pd.option_context("display.max_columns",None,"display.width",260,"display.max_colwidth",34):
        print(hits[use].to_string(index=False))
    print()

```

38. find_nash2.py (SHA-256 a843f7e393a4b17f...; 14 lines)

```

# -*- coding: utf-8 -*-
# All "Nash, J*" across all 5 career files, with np/nc (year-embedded column names normalized).
import pandas as pd, re
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
for y in [2020,2021,2022,2023,2024]:
    df=pd.read_excel(f"{SCR}/career_{y}.xlsx",sheet_name="Data")
    npcol=[c for c in df.columns if re.fullmatch(r"np\d+",str(c)) or str(c).startswith("np60")]
    nccol=[c for c in df.columns if re.fullmatch(r"nc\d+",str(c)) and "(ns)" not in str(c)]
    if y==2020: print("np-like cols:",[c for c in df.columns if str(c).startswith(("np","nc"))][12:])
    hits=df[df["authfull"].astype(str).str.match(r"Nash,\s*J*",case=False,na=False)].copy()
    print(f"\n== {y} ==")
    for _,r in hits.iterrows():
        npv=[r[c] for c in npcol[:1]]; ncv=[r[c] for c in nccol[:1]]
        print(f"    {r['authfull']}<28) {str(r['inst_name'][:44])<44) {r['cntry']} np={npv} nc={ncv} sfl={r['sm-subfield-1']}")

```

39. find_refs.py (SHA-256 efb008e91098bec8...; 15 lines)

```

# -*- coding: utf-8 -*-
# Locate the two reference scientists (Cieszewski, Bettinger) in the raw cohort and extract per-year metrics.
import pandas as pd, json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df=pd.read_excel(RAW,sheet_name="Data")
print("cols:",list(df.columns))
print("rows:",len(df),"| years:",sorted(df["year"].unique()))
for name in ["Cieszewski","Bettinger"]:
    hits=df[df["authfull"].astype(str).str.contains(name,case=False,na=False)]
    print(f"\n===== {name}: {len(hits)} row(s) =====")
    cols=["authfull","year","np","nc","cntry","inst_name","sm-subfield-1","sm-subfield-2","sm-field"]
    cols=[c for c in cols if c in hits.columns]
    with pd.option_context("display.max_columns",None,"display.width",240,"display.max_colwidth",40):
        print(hits[cols].sort_values("year").to_string(index=False))

```

40. find_uga.py (SHA-256 1f322aa2f52e15b6...; 26 lines)

```

# -*- coding: utf-8 -*-

```

```

# All UGA-affiliated cohort members meeting criterion 1 (Forestry) or 2 (Ecology&Forestry), latest release
(2024).
import pandas as pd, json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
RAW=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df=pd.read_excel(RAW,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
gF=(s1=="Forestry"); gE=(s1=="Ecology") & (s2=="Forestry")
df=df[gF|gE].copy(); df["crit"]=["1" if x else "2" for x in gF[gF|gE]]

# Which institution field(s) name UGA? Show any inst containing 'Georgia'
inst=df["inst_name"].astype(str)
print("Institutions containing 'Georgia' (in cohort):")
for v,c in inst[inst.str.contains("Georgia",case=False,na=False)].value_counts().items():
    print(f"    {c:>4} {v}")
print("\nColumns that might hold department:", [c for c in df.columns if any(k in c.lower() for k in
["dept","depart","affil","inst"])])

uga=df[inst.str.fullmatch(r"University of Georgia",case=False,na=True) | (inst=="University of Georgia")].copy()
print(f"\n=== UGA cohort members (any year 2020-2024): {uga['authfull'].nunique()} distinct names, {len(uga)}
author-years ===")
# latest record per person
uga["year"]=uga["year"].astype(int)
latest=uga.sort_values("year").groupby("authfull",as_index=False).tail(1)
cols=["authfull","crit","sm-subfield-1","sm-subfield-2","sm-field","np","nc","year"]
latest=latest.sort_values(["crit","authfull"])
with pd.option_context("display.max_columns",None,"display.width",240,"display.max_colwidth",44):
    print(latest[cols].to_string(index=False))

```

41. inspect2.py (SHA-256 cce2e6d404f6c316...; 13 lines)

```

from openpyxl import load_workbook
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
wb = load_workbook(WB, data_only=True)

print("### INSTITUTIONS SHEET layout (first block, rows 1-8, cols 1-9) ###")
ws = wb["SA-B Forestry institutions"]
for r in range(1,9):
    print(f"r{r}:", [ws.cell(row=r,column=c).value for c in range(1,10)])

print("\n### COUNTRIES SHEET header (rows 1-4, cols 1-8) ###")
ws2 = wb["SA-B Forestry countries"]
for r in range(1,5):
    print(f"r{r}:", [ws2.cell(row=r,column=c).value for c in range(1,9)])

```

42. inspect_cols.py (SHA-256 38d39fda8a20c474...; 20 lines)

```

# -*- coding: utf-8 -*-
import pandas as pd
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
xl=pd.ExcelFile(CAREER)
print("SHEETS:",xl.sheet_names)
df=pd.read_excel(CAREER,sheet_name="Data",nrows=5)
print("N COLS:",len(df.columns))
cols=list(df.columns)
print("ALL COLUMNS:")
for c in cols: print("    ",repr(c))
print("\n--- columns mentioning single-author-ish tokens ---")
for c in cols:
    lc=str(c).lower()
    if any(t in lc for t in ["nps","ncs","single","sole","cpsf","sf","sfl","frac"]):
        print("    ",repr(c))
# Key sheet legend if present
if "Key" in xl.sheet_names:
    k=pd.read_excel(CAREER,sheet_name="Key")
    print("\n--- KEY sheet (legend) ---")
    print(k.to_string()[:4000])

```

43. make_figures.py (SHA-256 22c9c5972775522e...; 116 lines)

```

import json
import matplotlib
matplotlib.use("Agg")

```

```

import matplotlib.pyplot as plt
from matplotlib.collections import PolyCollection
from matplotlib.colors import BoundaryNorm
from matplotlib import cm
import matplotlib.patches as mpatches
import numpy as np

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claudes\C--D-Dropbox-ClaudeCodeTest\ee9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
plt.rcParams.update({"font.family": "Arial", "font.size": 10, "figure.dpi": 200})
D = json.load(open(f"{SCR}/report_data.json", encoding="utf-8"))
YEARS = [2020,2021,2022,2023,2024]
GREEN = "#2F5233"; ECO = "#4C9F70"; BOTH = "#B7CE63"

# ---- Fig 1: entries per year by field (stacked area) ----
fy = {r["field"]: [r[str(y)] if str(y) in r else r[y] for y in YEARS] for r in D["field_year"]}
fig, ax = plt.subplots(figsize=(6.5,3.4))
ax.stackplot(YEARS, fy["Ecology"], fy["Both"], fy["Forestry"],
             labels=["Ecology","Both subfields","Forestry"], colors=[ECO,BOTH,GREEN], alpha=.92)
tot = [sum(v) for v in zip(fy["Ecology"],fy["Both"],fy["Forestry"])]
for x,y in zip(YEARS,tot): ax.annotate(f"{y:,}", (x,y),textcoords="offset
points",xytext=(0,4),ha="center",fontSize=8)
ax.set_xticks(YEARS); ax.set_ylabel("List entries"); ax.set_ylim(0,5600)
ax.legend(loc="lower right", frameon=False, fontSize=8)
ax.spines[["top","right"]].set_visible(False)
fig.tight_layout(); fig.savefig(f"{SCR}/fig1_trend.png", dpi=200); plt.close(fig)

# ---- Fig 3: top 15 countries 2024 ----
cc = [(r["country"], r["2024"]) if "2024" in r else r[2024] for r in D["rankings"]["country"][:15]]
fig, ax = plt.subplots(figsize=(6.5,4.2))
names=[x[0] for x in cc][::-1]; vals=[x[1] for x in cc][::-1]
ax.barh(names, vals, color=GREEN)
for i,v in enumerate(vals): ax.annotate(f" {v:,}", (v,i),va="center",fontSize=8)
ax.set_xlabel("List entries, 2024 release"); ax.set_xlim(0, max(vals)*1.12)
ax.spines[["top","right"]].set_visible(False)
fig.tight_layout(); fig.savefig(f"{SCR}/fig3_countries.png", dpi=200); plt.close(fig)

# ---- Fig 5: top 15 institutions 2024 ----
uu = [(r["inst_name"], r["2024"]) if "2024" in r else r[2024] for r in D["rankings"]["university_top60"][:15]]
fig, ax = plt.subplots(figsize=(6.5,4.6))
names=[x[0] if len(x[0])<=52 else x[0][:50]+"..." for x in uu][::-1]; vals=[x[1] for x in uu][::-1]
ax.barh(names, vals, color=ECO)
for i,v in enumerate(vals): ax.annotate(f" {v}", (v,i),va="center",fontSize=8)
ax.set_xlabel("List entries, 2024 release"); ax.set_xlim(0, max(vals)*1.12)
ax.spines[["top","right"]].set_visible(False)
fig.tight_layout(); fig.savefig(f"{SCR}/fig5_universities.png", dpi=200); plt.close(fig)

# ---- Fig 6: growth 2020->2024 ----
g = D["country_growth_pct"]
items = list(g.items())[:10] + [(k,g[k]) for k in ("United Kingdom","United States") if k in g]
fig, ax = plt.subplots(figsize=(6.5,3.8))
names=[x[0] for x in items][::-1]; vals=[x[1] for x in items][::-1]
cols=[GREEN if n not in ("United Kingdom","United States") else "#A63D40" for n in names]
ax.barh(names, vals, color=cols)
for i,v in enumerate(vals): ax.annotate(f" {v:+.0f}%", (v,i),va="center",fontSize=8)
ax.set_xlabel("Change in list entries, 2020 → 2024 (%) (countries with ≥20 entries in 2020)")
ax.set_xlim(0, max(vals)*1.15)
ax.spines[["top","right"]].set_visible(False)
fig.tight_layout(); fig.savefig(f"{SCR}/fig6_growth.png", dpi=200); plt.close(fig)

# ---- map helpers ----
def geoms(feature):
    g = feature["geometry"]
    if g is None: return []
    if g["type"]=="Polygon": return [g["coordinates"]]
    if g["type"]=="MultiPolygon": return g["coordinates"]
    return []

def draw_map(features, key_fn, values, bins, cmap_name, fname, title, legend_title, xlim=None, ylim=None,
figsize=(6.9,3.9)):
    cmap = plt.get_cmap(cmap_name, len(bins)-1)
    norm = BoundaryNorm(bins, cmap.N)
    fig, ax = plt.subplots(figsize=figsize)
    polys_bg, polys, cols = [], [], []
    for f in features:
        k = key_fn(f)
        v = values.get(k)
        for poly in geoms(f):

```

```

        ext = np.asarray(poly[0])
        if v is None or v == 0:
            polys_bg.append(ext)
        else:
            polys.append(ext); cols.append(cmap(norm(v)))
    ax.add_collection(PolyCollection(polys_bg, facecolors="#EFEFEA", edgecolors="#BBBBBB", linewidths=.3))
    if polys:
        ax.add_collection(PolyCollection(polys, facecolors=cols, edgecolors="#888888", linewidths=.3))
    ax.autoscale(); ax.set_aspect("auto" if xlim is None else "auto")
    if xlim: ax.set_xlim(*xlim)
    if ylim: ax.set_ylim(*ylim)
    ax.axis("off")
    handles = [mpatches.Patch(facecolor=cmap(norm((bins[i]+bins[i+1])/2)), edgecolor="#888",
        label=f"{bins[i]:,}-{bins[i+1]-1:,}" if bins[i+1]-1>bins[i] else f"{bins[i]:,}")
        for i in range(len(bins)-1)]
    handles.append(mpatches.Patch(facecolor="#EFEFEA", edgecolor="#BBB", label="0 / no data"))
    ax.legend(handles=handles, loc="lower left", fontsize=7, frameon=False, title=legend_title, title_fontsize=8)
    fig.tight_layout(); fig.savefig(f"{SCR}/{fname}", dpi=200, bbox_inches="tight"); plt.close(fig)

# ---- Fig 2: world choropleth ----
world = json.load(open(f"{SCR}/world_110m.geojson", encoding="utf-8"))
feats = [f for f in world["features"] if f["properties"].get("CONTINENT") != "Antarctica"]
def iso3(f):
    p=f["properties"]
    a=p.get("ISO_A3","-99")
    return p.get("ADM0_A3") if a in ("-99",None,"") else a
draw_map(feats, iso3, D["map_2024_iso3"], [1,5,10,25,50,100,250,500,2000], "YlGn",
    "fig2_worldmap.png", "", "Entries (2024)", ylim=(-60,85))

# ---- Fig 4: US states choropleth ----
us = json.load(open(f"{SCR}/us_states.geojson", encoding="utf-8"))
usfeats = [f for f in us["features"] if f["properties"]["name"] not in ("Alaska","Hawaii","Puerto Rico")]
draw_map(usfeats, lambda f: f["properties"]["name"], D["map_2024_us_state"],
    [1,5,10,25,50,100,300], "YlGn", "fig4_usmap.png", "", "Entries (2024)",
    xlim=(-125,-66), ylim=(24,50), figsize=(6.9,4.1))
ak = D["map_2024_us_state"].get("Alaska",0); hi = D["map_2024_us_state"].get("Hawaii",0); pr =
D["map_2024_us_state"].get("Puerto Rico",0)
print("AK/HI/PR (footnote):", ak, hi, pr)
print("figures done")

```

44. mapping.py (SHA-256 2531dd787ec3d910...; 1064 lines)

```

# -*- coding: utf-8 -*-
# id -> subdivision, determined from geographic knowledge of main campus/HQ.
M = {}

# ----- UNITED STATES (ids 0-728) -----
M[0]="California"      # UC Davis
M[1]="Oregon"          # Oregon State University (Corvallis)
M[2]="Federal/Multi-site" # USDA Forest Service
M[3]="Washington"      # University of Washington (Seattle)
M[4]="Wisconsin"        # UW-Madison
M[5]="Minnesota"        # UMN Twin Cities
M[6]="Michigan"         # Michigan State (East Lansing)
M[7]="Colorado"         # Colorado State (Fort Collins)
M[8]="New York"         # Cornell (Ithaca)
M[9]="North Carolina"   # NC State (Raleigh)
M[10]="Florida"         # UF (Gainesville)
M[11]="Georgia"         # UGA (Athens)
M[12]="Federal/Multi-site" # USGS
M[13]="Colorado"        # CU Boulder
M[14]="North Carolina"  # Duke (Durham)
M[15]="California"      # UC Santa Cruz
M[16]="Arizona"         # U Arizona (Tucson)
M[17]="California"      # UC Santa Barbara
M[18]="California"      # Stanford
M[19]="California"      # UC Berkeley
M[20]="California"      # UCLA
M[21]="Michigan"        # U Michigan Ann Arbor
M[22]="Illinois"        # UIUC
M[23]="Arizona"         # Arizona State (Tempe)
M[24]="Connecticut"     # Yale (New Haven)
M[25]="Pennsylvania"    # Penn State (University Park)
M[26]="California"      # UC Riverside
M[27]="Federal/Multi-site" # USDA ARS
M[28]="Alaska"          # UA Fairbanks
M[29]="Virginia"        # U Virginia (Charlottesville)
M[30]="Idaho"           # U Idaho (Moscow)

```

M[31]="Utah" # U Utah (Salt Lake City)
M[32]="Tennessee" # UT Knoxville
M[33]="Arizona" # Northern Arizona (Flagstaff)
M[34]="Montana" # U Montana (Missoula)
M[35]="New Jersey" # Princeton
M[36]="Maine" # U Maine (Orono)
M[37]="New York" # Institute of Ecosystem Studies (Cary, Millbrook NY)
M[38]="Connecticut" # UConn (Storrs)
M[39]="Utah" # Utah State (Logan)
M[40]="Texas" # Texas A&M (College Station)
M[41]="Massachusetts" # UMass Amherst
M[42]="Washington" # WSU Pullman
M[43]="Federal/Multi-site" # USFS Rocky Mountain Research Station (Fort Collins-based but distributed)
M[44]="New York" # SUNY ESF (Syracuse)
M[45]="New Hampshire" # UNH Durham
M[46]="California" # USFS Pacific Southwest Research Station (Albany CA)
M[47]="Virginia" # Virginia Tech (Blacksburg)
M[48]="Indiana" # Purdue (West Lafayette)
M[49]="New Mexico" # UNM (Albuquerque)
M[50]="Kansas" # Kansas State (Manhattan)
M[51]="North Carolina" # UNC Chapel Hill
M[52]="Texas" # UT Austin
M[53]="Maryland" # UMD College Park
M[54]="Wyoming" # U Wyoming (Laramie)
M[55]="Louisiana" # LSU (Baton Rouge)
M[56]="Wisconsin" # USDA Forest Products Laboratory (Madison WI)
M[57]="California" # Dept of ESPM = UC Berkeley
M[58]="Oregon" # U Oregon (Eugene)
M[59]="Indiana" # Indiana U Bloomington
M[60]="Kansas" # U Kansas (Lawrence)
M[61]="Hawaii" # UH Manoa
M[62]="Massachusetts" # Harvard
M[63]="Missouri" # U Missouri (Columbia)
M[64]="Alabama" # Auburn
M[65]="Maryland" # Smithsonian Environmental Research Center (Edgewater MD)
M[66]="Iowa" # Iowa State (Ames)
M[67]="California" # UC Irvine
M[68]="Oregon" # USDA ARS Corvallis Forestry Sciences Lab
M[69]="New York" # Columbia
M[70]="Unknown" # "School of Life Sciences" generic
M[71]="New York" # Wildlife Conservation Society (Bronx NY)
M[72]="Maryland" # UMD Center for Environmental Science (Cambridge MD)
M[73]="New Jersey" # Rutgers-New Brunswick
M[74]="Ohio" # Ohio State (Columbus)
M[75]="Oregon" # USFS Pacific Northwest Research Station (Portland OR)
M[76]="Federal/Multi-site" # USDA
M[77]="New York" # Stony Brook
M[78]="Nevada" # UN Reno
M[79]="Vermont" # U Vermont (Burlington)
M[80]="Florida" # Florida State (Tallahassee)
M[81]="Montana" # Montana State (Bozeman)
M[82]="South Carolina" # U South Carolina (Columbia)
M[83]="Virginia" # Nature Conservancy HQ (Arlington VA)
M[84]="Massachusetts" # Tufts (Medford)
M[85]="District of Columbia" # Smithsonian NMNH
M[86]="Pennsylvania" # U Penn (Philadelphia)
M[87]="New York" # American Museum of Natural History (NYC)
M[88]="Federal/Multi-site" # USDA ARS Rocky Mountain Research Station
M[89]="Oklahoma" # U Oklahoma (Norman)
M[90]="Oklahoma" # Oklahoma State (Stillwater)
M[91]="Kentucky" # U Kentucky (Lexington)
M[92]="Massachusetts" # Boston University
M[93]="Massachusetts" # Ecosystems Center = MBL Woods Hole MA
M[94]="Colorado" # Rocky Mountain Biological Laboratory (Gothic CO)
M[95]="Federal/Multi-site" # USGS Western Region
M[96]="New York" # Syracuse University
M[97]="Unknown" # Smithsonian Tropical Research Institute (Panama) - not US/CA/AU subdivision
M[98]="Illinois" # U Illinois at Chicago
M[99]="California" # Bodega Marine Laboratory (UC Davis, Bodega Bay CA)
M[100]="California" # San Diego State
M[101]="Ohio" # Miami University (Oxford OH)
M[102]="Florida" # U Miami (Coral Gables FL)
M[103]="New Hampshire" # Dartmouth (Hanover)
M[104]="Illinois" # Field Museum (Chicago)
M[105]="Federal/Multi-site" # USFS Northern Research Station
M[106]="Illinois" # U Chicago
M[107]="Montana" # U Montana Missoula
M[108]="Rhode Island" # Brown (Providence)
M[109]="Federal/Multi-site" # US EPA
M[110]="California" # UC Merced
M[111]="California" # Cal Poly Humboldt (Arcata)

M[112]="Tennessee" # ORNL Environmental Sciences Division (Oak Ridge TN)
M[113]="Vermont" # U Vermont
M[114]="Georgia" # Georgia Tech (Atlanta)
M[115]="North Carolina" # Wake Forest (Winston-Salem)
M[116]="Unknown" # "College of Science" generic
M[117]="Nebraska" # U Nebraska-Lincoln
M[118]="Hawaii" # Hawaii Institute of Marine Biology (UH)
M[119]="Colorado" # NCAR (Boulder CO)
M[120]="Massachusetts" # Northeastern (Boston)
M[121]="California" # UC San Diego
M[122]="Idaho" # USDA ARS Moscow Forestry Sciences Lab (Moscow ID)
M[123]="Minnesota" # UMN Duluth
M[124]="Georgia" # Emory (Atlanta)
M[125]="Florida" # Florida Institute of Technology (Melbourne)
M[126]="Virginia" # George Mason (Fairfax)
M[127]="Florida" # Florida Museum of Natural History (Gainesville)
M[128]="Florida" # USF Tampa
M[129]="Florida" # Florida International (Miami)
M[130]="Texas" # Texas Tech (Lubbock)
M[131]="California" # Scripps Institution of Oceanography (La Jolla)
M[132]="Illinois" # SIU Carbondale
M[133]="South Dakota" # South Dakota State (Brookings)
M[134]="Michigan" # Michigan Tech (Houghton)
M[135]="Ohio" # Ohio University (Athens)
M[136]="Virginia" # NSF (Alexandria VA)
M[137]="New Mexico" # Santa Fe Institute
M[138]="Missouri" # UMSL Dept of Biology (St. Louis)
M[139]="New Mexico" # New Mexico State (Las Cruces)
M[140]="Ohio" # Wright State (Dayton)
M[141]="Tennessee" # Oak Ridge National Laboratory
M[142]="Colorado" # Warner College of Natural Resources = Colorado State
M[143]="Massachusetts" # Woodwell Climate Research Center (Falmouth MA)
M[144]="Oklahoma" # U Oklahoma
M[145]="District of Columbia" # Smithsonian Institution
M[146]="Massachusetts" # Marine Biological Laboratory (Woods Hole)
M[147]="South Dakota" # USGS EROS (Sioux Falls SD)
M[148]="Federal/Multi-site" # NOAA
M[149]="Alabama" # U Alabama (Tuscaloosa)
M[150]="Arkansas" # U Arkansas (Fayetteville)
M[151]="Florida" # Rosenstiel School (U Miami, Key Biscayne)
M[152]="Massachusetts" # Clark University (Worcester)
M[153]="Maryland" # Towson
M[154]="Connecticut" # Yale School of the Environment
M[155]="Montana" # U Montana College of Forestry
M[156]="Oregon" # USGS Forest and Rangeland Ecosystem Science Center (Corvallis OR)
M[157]="Washington" # Western Washington University (Bellingham)
M[158]="Washington" # Pacific Northwest National Laboratory (Richland WA)
M[159]="North Carolina" # NCASI HQ (Cary NC)
M[160]="Indiana" # Notre Dame
M[161]="District of Columbia" # Georgetown
M[162]="Federal/Multi-site" # USGS Central Region
M[163]="California" # CSU Northridge
M[164]="New York" # Binghamton University
M[165]="Florida" # Archbold Biological Station (Venus FL)
M[166]="Unknown" # "College of the Environment" generic
M[167]="Maryland" # NASA Goddard (Greenbelt MD)
M[168]="Maryland" # Patuxent Wildlife Research Center (Laurel MD)
M[169]="Washington" # NOAA NMFS Northwest Regional Office (Seattle)
M[170]="Texas" # Rice (Houston)
M[171]="New York" # Cornell Lab of Ornithology (Ithaca)
M[172]="Unknown" # "College of Agriculture" generic
M[173]="District of Columbia" # Resources For The Future (DC)
M[174]="Maryland" # USDA ARS Beltsville (MD)
M[175]="Virginia" # Virginia Institute of Marine Science (Gloucester Point)
M[176]="North Carolina" # Duke Marine Lab (Beaufort NC)
M[177]="Massachusetts" # UMass Boston
M[178]="Louisiana" # LSU AgCenter (Baton Rouge)
M[179]="Florida" # Florida Gulf Coast (Fort Myers)
M[180]="Louisiana" # Tulane (New Orleans)
M[181]="Ohio" # U Toledo
M[182]="Colorado" # Fort Collins Science Center (USGS, Fort Collins CO)
M[183]="Texas" # Texas State (San Marcos)
M[184]="District of Columbia" # Carnegie Institution of Washington (HQ DC)
M[185]="District of Columbia" # National Zoo (DC)
M[186]="Maryland" # College of Computer, Mathematical & Natural Sciences = UMD College Park
M[187]="Michigan" # Western Michigan (Kalamazoo)
M[188]="Missouri" # WUSTL (St. Louis)
M[189]="California" # Moss Landing Marine Labs (CA)
M[190]="Missouri" # Missouri Botanical Garden (St. Louis)
M[191]="Colorado" # U Colorado Museum of Natural History (Boulder)
M[192]="Unknown" # "School of Veterinary Medicine" generic

M[193]="Rhode Island" # U Rhode Island (Kingston)
M[194]="California" # California Academy of Sciences (SF)
M[195]="Rhode Island" # URI Narragansett Bay Campus
M[196]="Ohio" # College of Wooster (OH)
M[197]="North Carolina" # UNC Greensboro
M[198]="Texas" # U Houston
M[199]="Maine" # Bowdoin (Brunswick ME)
M[200]="Florida" # UF IFAS
M[201]="Mississippi" # Mississippi State
M[202]="Kentucky" # U Kentucky College of Agriculture
M[203]="North Dakota" # U North Dakota (Grand Forks)
M[204]="Massachusetts" # Harvard Forest (Petersham MA)
M[205]="Nevada" # UNLV
M[206]="Kansas" # KU Natural History Museum (Lawrence)
M[207]="Unknown" # "College of Arts & Sciences" generic
M[208]="Texas" # U North Texas (Denton)
M[209]="South Carolina" # Clemson
M[210]="Missouri" # UMSL
M[211]="North Carolina" # NCASI
M[212]="California" # NASA Ames (Moffett Field CA)
M[213]="Florida" # USDA ARS Invasive Plant Research Lab (Fort Lauderdale FL)
M[214]="Tennessee" # Vanderbilt (Nashville)
M[215]="Wisconsin" # UW-Milwaukee
M[216]="Utah" # BYU (Provo)
M[217]="South Carolina" # Savannah River Ecology Lab (Aiken SC)
M[218]="Georgia" # UGA College of Ag Athens
M[219]="Unknown" # "School of Biological Sciences" generic
M[220]="New York" # Cornell CALS
M[221]="North Carolina" # UNC School of Medicine (Chapel Hill)
M[222]="Arizona" # ASU Downtown Phoenix
M[223]="Federal/Multi-site" # National Park Service
M[224]="New Jersey" # School of Env and Biological Sciences = Rutgers
M[225]="California" # "University of California" (system, Oakland) -> California
M[226]="Indiana" # Indiana State (Terre Haute)
M[227]="Massachusetts" # Woods Hole Oceanographic Institution
M[228]="Texas" # UT Arlington
M[229]="North Carolina" # UNC Wilmington
M[230]="Massachusetts" # The Ecosystems Center (MBL Woods Hole)
M[231]="District of Columbia" # Global Environment Facility (DC)
M[232]="Unknown" # "College of Arts and Sciences" generic
M[233]="Alaska" # NOAA Alaska Fisheries Science Center (Seattle-based HQ though) -> Alaska? NMFS AFSC is in Seattle WA. Use Washington.
M[234]="Federal/Multi-site" # USDA ARS Sugarbeet Research Unit
M[235]="New York" # NYU
M[236]="District of Columbia" # National Geographic Society (DC)
M[237]="District of Columbia" # National Zoological Park (DC)
M[238]="Iowa" # U Iowa (Iowa City)
M[239]="Delaware" # U Delaware (Newark)
M[240]="Louisiana" # UL Lafayette
M[241]="Michigan" # Central Michigan (Mount Pleasant)
M[242]="New York" # City College of New York
M[243]="Pennsylvania" # Franklin and Marshall (Lancaster)
M[244]="California" # H.T. Harvey & Associates (Los Gatos CA)
M[245]="Federal/Multi-site" # U.S. Forest Service
M[246]="Connecticut" # Williams-Mystic (Mystic CT)
M[247]="New York" # William H. Miner Agricultural Research Institute (Chazy NY)
M[248]="New York" # U Rochester
M[249]="Pennsylvania" # U Pittsburgh
M[250]="California" # CSU Fullerton
M[251]="Florida" # CSA Ocean Sciences (Stuart FL)
M[252]="New York" # Bard College (Annandale-on-Hudson NY)
M[253]="Federal/Multi-site" # US Fish & Wildlife Service
M[254]="Louisiana" # U New Orleans
M[255]="California" # Farallon Institute (Petaluma CA)
M[256]="Unknown" # WrightLink Consulting - unclear
M[257]="California" # Whittier College (CA)
M[258]="District of Columbia" # American University (DC)
M[259]="Virginia" # Ferrum College (VA)
M[260]="Arkansas" # Winrock International (Little Rock AR)
M[261]="West Virginia" # USFS Northeastern Research Station Morgantown (WV)
M[262]="Virginia" # Sweet Briar College (VA)
M[263]="Florida" # Stetson (DeLand FL)
M[264]="Federal/Multi-site" # US Dept of Energy
M[265]="Illinois" # Illinois State (Normal)
M[266]="Illinois" # Illinois State Museum (Springfield)
M[267]="New York" # CUNY
M[268]="Illinois" # Brookfield Zoo (Chicago area)
M[269]="Texas" # Texas A&M AgriLife Research
M[270]="Unknown" # "Monitoring and Research Department" generic
M[271]="Washington" # UW Tacoma
M[272]="Colorado" # Rocky Mountain Tree-Ring Research (Fort Collins CO)

M[273]="California" # Lawrence Berkeley National Laboratory
M[274]="Unknown" # "Research Support" generic
M[275]="Minnesota" # Macalester (St. Paul)
M[276]="Unknown" # Robert A. Monserud Consulting - unclear (person, likely OR/Portland) -> Unknown
M[277]="California" # Sonoma State
M[278]="Pennsylvania" # Villanova
M[279]="Virginia" # Virginia Commonwealth (Richmond)
M[280]="Indiana" # IPFW (Fort Wayne)
M[281]="Oregon" # Southern Oregon (Ashland)
M[282]="Arizona" # Southwest Watershed Research Center (USDA ARS, Tucson AZ)
M[283]="Colorado" # Jonah Ventures (Boulder CO)
M[284]="Washington" # Cascadia Research Collective (Olympia WA)
M[285]="Minnesota" # St. Cloud State (MN)
M[286]="Alaska" # UA Museum of the North (Fairbanks)
M[287]="Pennsylvania" # Drexel (Philadelphia)
M[288]="New York" # EcoHealth Alliance (NYC)
M[289]="Unknown" # EcoIsoMix.com - unclear
M[290]="Arkansas" # U Arkansas at Monticello
M[291]="Unknown" # Seastar Scientific - unclear
M[292]="Unknown" # Shields Engineering LLC - unclear
M[293]="Unknown" # Silvaware Inc - unclear
M[294]="Pennsylvania" # Juniata (Huntingdon PA)
M[295]="Ohio" # Kent State
M[296]="Florida" # U West Florida (Pensacola)
M[297]="Wisconsin" # UW Oshkosh
M[298]="California" # Earth Innovation Institute (San Francisco CA)
M[299]="Virginia" # Conservation and Research Center (National Zoo, Front Royal VA)
M[300]="Florida" # Dynamac Corporation (Cape Canaveral FL / Rockville MD) -> Unknown? Dynamac HQ
historically FL. Use Florida.
M[301]="Indiana" # Earlham College (Richmond IN)
M[302]="Unknown" # RTK Consulting Services - unclear
M[303]="Alabama" # Samford (Birmingham AL)
M[304]="Virginia" # Old Dominion (Norfolk)
M[305]="Washington" # Microsoft Corporation (Redmond WA)
M[306]="Massachusetts" # Mount Holyoke (South Hadley MA)
M[307]="Unknown" # ProFor Consulting - unclear
M[308]="Washington" # NOAA Northwest Fisheries Science Center (Seattle)
M[309]="Wisconsin" # Marquette (Milwaukee)
M[310]="Unknown" # "Department of Human Ecology" generic
M[311]="Ohio" # Denison (Granville OH)
M[312]="Connecticut" # Connecticut Agricultural Experiment Station (New Haven)
M[313]="Oregon" # Oregon Institute of Marine Biology (Charleston OR)
M[314]="Massachusetts" # New England Wild Flower Society (Framingham MA)
M[315]="North Dakota" # North Dakota State (Fargo)
M[316]="Illinois" # Northern Illinois (DeKalb)
M[317]="Florida" # Nova Southeastern (Fort Lauderdale)
M[318]="Oregon" # Oregon Natural Desert Association (Bend OR)
M[319]="Pennsylvania" # Allegheny College (Meadville PA)
M[320]="Delaware" # USDA ARS Beneficial Insects Introduction Research Unit (Newark DE)
M[321]="Arizona" # USDA ARS RMRS Flagstaff Lab
M[322]="Massachusetts" # Brandeis (Waltham)
M[323]="Arizona" # Borderlands Restoration Network (Patagonia AZ)
M[324]="Connecticut" # Wesleyan (Middletown)
M[325]="North Carolina" # Western Carolina (Cullowhee)
M[326]="Montana" # Clearwater Resource Council (Montana)
M[327]="Oregon" # US EPA NHEERL (Corvallis OR is a major site; HQ RTP NC) -> lab distributed, but named as
Western Ecology Div likely Corvallis. Use Oregon.
M[328]="Virginia" # Conservation International (Arlington VA)
M[329]="Hawaii" # U Hawaii at Hilo
M[330]="Unknown" # "Dept of Landscape Architecture & Environmental Planning" generic
M[331]="Oregon" # USDA ARS Horticultural Crops Research Unit (Corvallis OR)
M[332]="New York" # Queens College CUNY
M[333]="Virginia" # RESOLVE (Washington DC area / Arlington) -> District of Columbia
M[334]="Texas" # Re:wild (Austin TX)
M[335]="New York" # Rensselaer Polytechnic (Troy NY)
M[336]="California" # SDSU College of Sciences
M[337]="Unknown" # SABER Consulting - unclear
M[338]="Missouri" # Saint Louis Zoo
M[339]="District of Columbia" # Earth Policy Institute (DC)
M[340]="Unknown" # Conservation Planning Technologies - unclear
M[341]="Indiana" # Corteva Agriscience (Indianapolis IN)
M[342]="Unknown" # Project Drawdown (San Francisco CA) -> California
M[343]="Oregon" # Portland State
M[344]="Unknown" # "College of Engineering" generic
M[345]="Florida" # Moffitt Cancer Center (Tampa)
M[346]="Utah" # Natural History Museum of Utah (Salt Lake City)
M[347]="Hawaii" # National Tropical Botanical Garden (Kalaheo HI)
M[348]="Maryland" # SESYNC (Annapolis MD)
M[349]="California" # NOAA NMFS Southwest Regional Office (Long Beach CA)
M[350]="Tennessee" # U Memphis
M[351]="Maine" # Irland Group (Maine forestry consultancy)

M[352]="Montana" # Salish Kootenai College (Pablo MT)
M[353]="Maryland" # UMBC
M[354]="Montana" # Center for Large Landscape Conservation (Bozeman MT)
M[355]="Unknown" # Environmental Forestry Consultants - unclear
M[356]="California" # Formation Environmental LLC (Sacramento CA) -> California
M[357]="California" # Global Footprint Network (Oakland CA)
M[358]="Maryland" # HHMI (Chevy Chase MD)
M[359]="Massachusetts" # Harvard SEAS
M[360]="Oklahoma" # U Tulsa
M[361]="District of Columbia" # George Washington University (DC)
M[362]="Massachusetts" # I.C.T. Nisbet & Company (North Falmouth MA)
M[363]="Idaho" # Idaho State (Pocatello)
M[364]="California" # Humboldt State (Arcata CA)
M[365]="Nebraska" # U Nebraska Omaha
M[366]="Wyoming" # Institute for Ethnomedicine (Jackson WY)
M[367]="Indiana" # Indiana U School of Medicine-Terre Haute
M[368]="Maryland" # Johns Hopkins Carey Business School (Baltimore)
M[369]="Massachusetts" # JFK School of Government (Harvard)
M[370]="Louisiana" # Southeastern Louisiana (Hammond)
M[371]="Virginia" # Sustainable Waters (Virginia)
M[372]="Wisconsin" # USGS National Wildlife Health Center (Madison WI)
M[373]="Unknown" # "State Univ" generic
M[374]="Unknown" # Environmental Pulse Institute - unclear
M[375]="California" # USDA ARS Western Regional Research Center (Albany CA)
M[376]="Unknown" # "Faculty of Arts and Sciences" generic
M[377]="Louisiana" # Tulane School of Science and Engineering
M[378]="Montana" # FireTree Wildland Fire Sciences LLC (Missoula MT likely) -> Montana
M[379]="Florida" # Florida Institute for Conservation Science
M[380]="Michigan" # Grand Valley State (Allendale MI)
M[381]="Mississippi" # USDA ARS Stoneville (MS)
M[382]="Ohio" # The Davey Tree Expert Company (Kent OH)
M[383]="Montana" # The Science and Conservation Center (Billings MT)
M[384]="New York" # Brookhaven National Laboratory (Upton NY)
M[385]="Unknown" # Clymene Enterprises - unclear
M[386]="Arkansas" # U Arkansas at Little Rock
M[387]="Unknown" # "Section of Ecology, Behavior & Evolution" generic (UCSD) -> California
M[388]="Washington" # Shannon Point Marine Center (WWU, Anacortes WA)
M[389]="California" # San Francisco State
M[390]="Arkansas" # Eastern Ozarks Forestry Council (Arkansas)
M[391]="Unknown" # Universidad de Puerto Rico (Puerto Rico - not US state) -> Unknown
M[392]="New Hampshire" # University System of New Hampshire
M[393]="Ohio" # U Akron
M[394]="Unknown" # "School of Natural Resources" generic
M[395]="California" # Lang Railsback & Associates (Arcata CA)
M[396]="Washington" # Salmon Recovery Funding Board (WA state agency)
M[397]="California" # Point Blue Conservation Science (Petaluma CA)
M[398]="Colorado" # USDA APHIS National Wildlife Research Center (Fort Collins CO)
M[399]="Massachusetts" # Boston College (Chestnut Hill)
M[400]="Unknown" # Lora Consulting LLC - unclear
M[401]="Missouri" # Saint Louis University
M[402]="Maine" # Maine College of Engineering and Computing (U Maine)
M[403]="Ohio" # U Cincinnati
M[404]="New York" # Rockefeller University (NYC)
M[405]="Missouri" # Missouri State (Springfield)
M[406]="Washington" # UW School of Oceanography
M[407]="Utah" # U Utah College of Science
M[408]="Federal/Multi-site" # Dept of Agriculture Forest Service
M[409]="New Jersey" # Atlantic Health (Morristown NJ)
M[410]="North Dakota" # USDA ARS Red River Valley Ag Research Center (Fargo ND)
M[411]="Unknown" # "Deceased 1 November 2015"
M[412]="Unknown" # "Department of Entomology" generic
M[413]="Alabama" # Dauphin Island Sea Laboratory (AL)
M[414]="Alaska" # Alaska Ecoscience (Fairbanks AK)
M[415]="Oregon" # Amnis Opes Institute (Corvallis OR)
M[416]="California" # Aquarium of the Pacific (Long Beach CA)
M[417]="Illinois" # Argonne National Laboratory (Lemont IL)
M[418]="Federal/Multi-site" # US National Park Service
M[419]="South Carolina" # Belle W. Baruch Institute (USC, Georgetown SC)
M[420]="Idaho" # USFS Boise National Forest (ID)
M[421]="North Carolina" # UNC Asheville
M[422]="Unknown" # Cellulose Sciences International - unclear
M[423]="New York" # CUNY
M[424]="New York" # Buffalo Museum of Science
M[425]="Unknown" # WildBeecology Consulting - unclear
M[426]="Unknown" # USFS Forestry Science Lab (generic) -> Federal/Multi-site
M[427]="Minnesota" # USFS North Central Research Station (St. Paul MN) -> historically St Paul; distributed ->
Federal/Multi-site
M[428]="Kansas" # The Land Institute (Salina KS)
M[429]="Michigan" # Eastern Michigan (Ypsilanti)
M[430]="New York" # SUNY System (Albany NY)
M[431]="Unknown" # Huntington Consulting - unclear

M[432]="Maine" # University of New England USA (Biddeford ME)
M[433]="Illinois" # Illinois Wesleyan (Bloomington)
M[434]="California" # Altman Plants (Vista CA)
M[435]="Alabama" # IFDC (Muscle Shoals AL)
M[436]="Unknown" # "LLC" generic
M[437]="Wisconsin" # UW-Stevens Point
M[438]="California" # Jet Propulsion Laboratory (Pasadena CA)
M[439]="District of Columbia" # EcoAgriculture Partners (Washington DC)
M[440]="California" # Energy & Resources Group (UC Berkeley)
M[441]="Montana" # Jaronski Mycological Consulting LLC (Sidney MT area) -> Montana
M[442]="Unknown" # "Division of Biological Sciences" generic
M[443]="California" # UC Division of Agriculture and Natural Resources (Davis/Oakland CA)
M[444]="Unknown" # Universidad de Puerto Rico Rio Piedras -> Unknown (PR)
M[445]="Tennessee" # UT College of Veterinary Medicine (Knoxville)
M[446]="Rhode Island" # U Rhode Island
M[447]="Maine" # Atlantic Salmon Federation Inc (also NB Canada; US office Brunswick ME) -> Maine
M[448]="Hawaii" # East-West Center (Honolulu HI)
M[449]="Oregon" # Oregon Health & Science University (Portland)
M[450]="California" # NCEAS (Santa Barbara CA)
M[451]="Kentucky" # College of Agriculture, Food and Environment (U Kentucky)
M[452]="Unknown" # OurResearch (nonprofit, distributed) -> Unknown
M[453]="Florida" # New College of Florida (Sarasota)
M[454]="District of Columbia" # World Resources Institute (DC)
M[455]="District of Columbia" # World Wildlife Fund (US HQ DC)
M[456]="West Virginia" # USDA ARS Appalachian Fruit Research Station (Kearneysville WV)
M[457]="Florida" # U Central Florida (Orlando)
M[458]="Georgia" # Renewable Bioproducts Institute (Georgia Tech, Atlanta)
M[459]="Washington" # College of Forest Resources (UW) -> Washington
M[460]="Wisconsin" # UW-Eau Claire
M[461]="Illinois" # Morton Arboretum (Lisle IL)
M[462]="Tennessee" # Oak Ridge Associated Universities (Oak Ridge TN)
M[463]="Michigan" # NOAA Great Lakes Environmental Research Laboratory (Ann Arbor MI)
M[464]="Florida" # Mote Marine Laboratory & Aquarium (Sarasota FL)
M[465]="Unknown" # "College of Natural Sciences and Mathematics" generic
M[466]="District of Columbia" # POLICY Project (Futures Group, DC) -> Unknown? Use District of Columbia
M[467]="District of Columbia" # Ocean Conservancy (DC)
M[468]="Florida" # Mote Marine Laboratory (Sarasota FL)
M[469]="Illinois" # Northwestern (Evanston)
M[470]="Maryland" # U Maryland School of Medicine (Baltimore)
M[471]="Massachusetts" # Woodwell Climate Research Center Inc (Falmouth MA)
M[472]="Federal/Multi-site" # USGS (Retired)
M[473]="North Carolina" # UNC Gillings School of Global Public Health (Chapel Hill)
M[474]="Federal/Multi-site" # USGS (retired)
M[475]="Unknown" # "Department of Environmental Resources Engineering" generic (Humboldt) -> California
M[476]="Colorado" # Colorado State University System
M[477]="Hawaii" # College of Tropical Agriculture and Human Resources (UH Manoa)
M[478]="Massachusetts" # UMass Dartmouth
M[479]="Kansas" # Land Institute (Salina KS)
M[480]="Texas" # Lamar University (Beaumont TX)
M[481]="Michigan" # U Michigan
M[482]="New Jersey" # Rutgers
M[483]="Indiana" # Luddy School (Indiana U Bloomington)
M[484]="Louisiana" # Louisiana Tech (Ruston)
M[485]="Pennsylvania" # Lehigh (Bethlehem)
M[486]="Indiana" # Rose-Hulman (Terre Haute)
M[487]="Connecticut" # UConn Avery Point (Groton)
M[488]="Colorado" # U Colorado Denver
M[489]="Missouri" # St. Louis University
M[490]="Tennessee" # Vanderbilt Law School
M[491]="Tennessee" # Vanderbilt School of Medicine
M[492]="Mississippi" # U Mississippi (Oxford)
M[493]="District of Columbia" # Society for Conservation Biology (DC)
M[494]="Rhode Island" # Division of Biology and Medicine (Brown) -> Rhode Island
M[495]="Alaska" # UA Anchorage
M[496]="California" # UC Cooperative Extension
M[497]="Texas" # Sam Houston State (Huntsville TX)
M[498]="California" # San Diego Zoo Wildlife Alliance
M[499]="Unknown" # "School of Engineering" generic
M[500]="Unknown" # EcoTipping Points Project - unclear
M[501]="California" # Energy and Resources Group (UC Berkeley)
M[502]="Oregon" # Eastern Oregon Agricultural Research Center (Burns OR)
M[503]="District of Columbia" # EcoAgriculture International (DC) -> District of Columbia
M[504]="California" # Island Conservation (Santa Cruz CA)
M[505]="California" # Vibrant Data Labs (San Francisco CA)
M[506]="New York" # St. Lawrence University (Canton NY)
M[507]="Colorado" # Walter Scott Jr College of Engineering (Colorado State)
M[508]="Kansas" # Weed Science Society of America (Lawrence KS HQ) -> Unknown? WSSA HQ Westminster CO /
varies -> Unknown
M[509]="Washington" # School of Environmental and Forest Sciences (UW)
M[510]="Georgia" # Woodruff School of Mechanical Engineering (Georgia Tech)
M[511]="Unknown" # "Independent Researcher"

M[512]="Illinois" # Illinois Natural History Survey (Champaign)
M[513]="Illinois" # Illinois Medical Society -> Illinois
M[514]="District of Columbia" # IFPRI (DC)
M[515]="Maryland" # Johns Hopkins University (Baltimore)
M[516]="Ohio" # John Carroll University (University Heights OH)
M[517]="Maryland" # Fogarty International Center (NIH, Bethesda MD)
M[518]="Pennsylvania" # Bucknell (Lewisburg)
M[519]="Unknown" # "Co" generic
M[520]="New Jersey" # The College of New Jersey (Ewing)
M[521]="North Carolina" # UNC Asheville
M[522]="Illinois" # Chicago Botanic Garden (Glencoe IL)
M[523]="Washington" # Friday Harbor Laboratories (UW, San Juan Island WA)
M[524]="Florida" # Herbert Wertheim College of Engineering (U Florida)
M[525]="California" # Center for the Analysis of Sustainable Agricultural Systems (Kensington CA)
M[526]="California" # The Institute for Bird Populations (Petaluma CA)
M[527]="Unknown" # GreenValue Enterprises LLC - unclear
M[528]="Federal/Multi-site" # USDA-ARS
M[529]="Massachusetts" # Harvard Graduate School of Design
M[530]="Alabama" # U South Alabama (Mobile)
M[531]="Colorado" # Wildlife Protection Solutions (Denver CO)
M[532]="Hawaii" # East West Centre (Honolulu)
M[533]="Unknown" # "Fisheries Research Group" generic
M[534]="California" # Caltech (Pasadena)
M[535]="New York" # Cary Institute of Ecosystem Studies (Millbrook NY)
M[536]="Connecticut" # Bartlett Tree Research Laboratories (Charlotte NC HQ) -> North Carolina
M[537]="Unknown" # Biophysical Economics Institute - unclear
M[538]="Arizona" # USDA ARS U.S. Arid-Land Agricultural Research Center (Maricopa AZ)
M[539]="Washington" # Weyerhaeuser Company (Seattle WA HQ)
M[540]="Georgia" # Zoo Atlanta
M[541]="District of Columbia" # American Geophysical Union (DC)
M[542]="Michigan" # Altarum Institute (Ann Arbor MI)
M[543]="New York" # Applied Biomathematics (Setauket NY)
M[544]="Unknown" # AquaNova LLC - unclear
M[545]="Maine" # Archipelago Consulting (Portland ME)
M[546]="Virginia" # William & Mary (Williamsburg)
M[547]="Massachusetts" # Woods Hole Research Center (Falmouth MA)
M[548]="District of Columbia" # Worldwatch Institute (DC)
M[549]="Kansas" # USDA ARS Center for Grain and Animal Health Research (Manhattan KS)
M[550]="Federal/Multi-site" # USDA ARS Office of National Programs
M[551]="Utah" # USDA ARS Poisonous Plant Research Laboratory (Logan UT)
M[552]="Federal/Multi-site" # USDA ARS Soil and Water Management Unit
M[553]="Federal/Multi-site" # USFS Northern Research Station
M[554]="California" # UCI School of Medicine (Irvine)
M[555]="Florida" # UF College of Veterinary Medicine (Gainesville)
M[556]="District of Columbia" # UNEP DEWA North America (Washington DC) -> District of Columbia
M[557]="Texas" # UNT College of Engineering (Denton)
M[558]="Federal/Multi-site" # USGS
M[559]="District of Columbia" # US National Academy of Education (DC)
M[560]="California" # USC Gould School of Law (Los Angeles)
M[561]="District of Columbia" # USDA ARS Washington DC
M[562]="Idaho" # USFS Boise Aquatic Sciences Lab (ID)
M[563]="Mississippi" # USFS Center for Bottomland Hardwoods Research (Stoneville MS)
M[564]="Mississippi" # U Southern Mississippi (Hattiesburg)
M[565]="North Carolina" # Trinity College of Arts & Sciences (Duke)
M[566]="New York" # The Graduate Center (CUNY)
M[567]="New York" # The Safina Center (Setauket NY)
M[568]="Ohio" # U Akron
M[569]="Texas" # Texas A&M
M[570]="Unknown" # "The College of Natural Resources" generic
M[571]="California" # The David and Lucile Packard Foundation (Los Altos CA)
M[572]="New York" # Doctorate-Granting Institution of CUNY
M[573]="Massachusetts" # The Ecosystems Center MBL (Woods Hole)
M[574]="Federal/Multi-site" # USFS Southern Research Station
M[575]="Mississippi" # USFS Utilization of Southern Forest Resources -> Federal/Multi-site
M[576]="Federal/Multi-site" # USDA Natural Resources Conservation Service
M[577]="Oregon" # USGS Forest and Rangeland Ecosystems Science Center (Corvallis OR)
M[578]="Federal/Multi-site" # US Dept of the Interior
M[579]="California" # Stanford Woods Institute for the Environment
M[580]="Unknown" # "State University" generic
M[581]="Arizona" # Steve Sanghi College of Engineering (ASU? Sanghi = Microchip; ASU) -> Arizona
M[582]="Unknown" # Systematic Ecology - unclear
M[583]="Texas" # Tarleton State (Stephenville TX)
M[584]="District of Columbia" # Tennenbaum Marine Observatories Network (Smithsonian) -> District of Columbia
M[585]="California" # Tetra Tech Inc (Pasadena CA HQ)
M[586]="Texas" # Texas A&M Corpus Christi
M[587]="Unknown" # Soil Microbial Systems Laboratory (USDA) -> Federal/Multi-site
M[588]="Unknown" # Sound Solutions for Sustainable Science - unclear
M[589]="Illinois" # Species Conservation Toolkit Initiative (Chicago Zoological Society) -> Illinois
M[590]="California" # Stanford Doerr School of Sustainability
M[591]="California" # Stanford Engineering
M[592]="California" # Universidad Estatal de San Diego = San Diego State

M[593]="Alaska" # UA Southeast (Juneau)
M[594]="Unknown" # "School of Science" generic
M[595]="Montana" # Science and Conservation Center (Billings MT)
M[596]="New Mexico" # Sevilleta LTER (University of New Mexico)
M[597]="Virginia" # Smithsonian Conservation Biology Institute (Front Royal VA)
M[598]="Maryland" # Smithsonian Environmental Research Station (Edgewater MD)
M[599]="District of Columbia" # Smithsonian National Zoo and Conservation Biology Institute (DC)
M[600]="Oklahoma" # Sam Noble Oklahoma Museum of Natural History (Norman)
M[601]="California" # Santa Clara University
M[602]="Unknown" # "School of Natural Resources and Environment" generic
M[603]="Wisconsin" # Rockwell Automation HQ (Milwaukee WI)
M[604]="Unknown" # Rolling Contact Engineering - unclear
M[605]="Vermont" # Rubenstein School of Environment and Natural Resources (U Vermont)
M[606]="California" # Rausser College of Natural Resources (UC Berkeley)
M[607]="Florida" # Rayonier Inc Forest Research Center (Yulee/Fernandina FL)
M[608]="Unknown" # Renewable Resources Associates - unclear
M[609]="Unknown" # "Retired"
M[610]="Unknown" # "Retired/Independent Researcher"
M[611]="Hawaii" # Pacific Rim Conservation (Honolulu HI)
M[612]="Washington" # Pacific Wildland Fire Sciences Laboratory (USFS, Seattle WA)
M[613]="Pennsylvania" # Penn State Shenango
M[614]="Oregon" # Portland VA Medical Center
M[615]="North Carolina" # Pratt School of Engineering (Duke)
M[616]="New Jersey" # Princeton School of Public and International Affairs
M[617]="Montana" # Pyrologix (Missoula MT)
M[618]="California" # Open Philanthropy (San Francisco CA)
M[619]="Oregon" # PNW Research Station (USFS, Portland OR)
M[620]="New York" # Pace University (NYC)
M[621]="California" # Pacific Ecoinformatics and Computational Ecology Lab (Berkeley CA)
M[622]="Illinois" # Negaunee Institute for Plant Conservation (Chicago Botanic Garden)
M[623]="Illinois" # Negaunee Integrative Research Center (Field Museum, Chicago)
M[624]="Montana" # Norm Asbjornson College of Engineering (Montana State)
M[625]="Federal/Multi-site" # Northern Research Station (USFS)
M[626]="Oregon" # OHSU School of Medicine (Portland)
M[627]="Hawaii" # U Hawaii Windward Community College
M[628]="Kansas" # U Kansas School of Law
M[629]="Georgia" # National Center for Emerging and Zoonotic Infectious Diseases (CDC, Atlanta)
M[630]="Maryland" # NHGRI (NIH, Bethesda MD)
M[631]="Maryland" # NIH (Bethesda MD)
M[632]="California" # Natural History Museum of Los Angeles County
M[633]="Federal/Multi-site" # NOAA NMFS
M[634]="Washington" # NOAA National Marine Mammal Laboratory (Seattle)
M[635]="Federal/Multi-site" # NASA
M[636]="Massachusetts" # Manomet Center for Conservation Sciences (Plymouth MA)
M[637]="Unknown" # Marine Resource Consultant - unclear
M[638]="Maine" # U Maine at Farmington
M[639]="Maryland" # U Maryland
M[640]="Maryland" # U Maryland (US)
M[641]="Maryland" # U Maryland Baltimore (UMB)
M[642]="Oklahoma" # Langston University (OK)
M[643]="Unknown" # "Limnology and Oceanography Letters" journal -> Unknown
M[644]="Unknown" # "Ltd." generic
M[645]="New York" # Mailman School of Public Health (Columbia)
M[646]="Hawaii" # Makawahi Cave Reserve (Kauai HI)
M[647]="Maryland" # Johns Hopkins School of Medicine (Baltimore)
M[648]="Unknown" # "Independent scientist"
M[649]="Washington" # Institute for Health Metrics and Evaluation (Seattle)
M[650]="Unknown" # Institute for Seabird Research and Conservation - unclear
M[651]="Washington" # Institute for Systems Biology (Seattle)
M[652]="Nebraska" # Institute of Agriculture and Natural Resources (U Nebraska-Lincoln)
M[653]="Florida" # Integrated Mathematical Oncology Dept (Moffitt, Tampa)
M[654]="Unknown" # "Interdisciplinary Research Building" generic
M[655]="Georgia" # International Plant Nutrition Institute (Peachtree Corners GA)
M[656]="Nebraska" # U Nebraska at Omaha
M[657]="Pennsylvania" # Huck Institutes of the Life Sciences (Penn State)
M[658]="Massachusetts" # I.C.T. Nisbet and Company (North Falmouth MA)
M[659]="Idaho" # Idaho State College of Science and Engineering
M[660]="Illinois" # Illinois Audubon Society
M[661]="Unknown" # "Inc" generic
M[662]="Florida" # Inc. Forest Research Center (Rayonier) -> Florida
M[663]="New Jersey" # Grant F. Walton Center for Remote Sensing (Rutgers)
M[664]="Iowa" # Grinnell College (IA)
M[665]="Oregon" # H.J. Andrews Experimental Forest (OR)
M[666]="Florida" # Harbor Branch Oceanographic Institute at FAU (Fort Pierce FL)
M[667]="Massachusetts" # Harvard T.H. Chan School of Public Health
M[668]="Pennsylvania" # Hawk Mountain Sanctuary (Kempton PA)
M[669]="New York" # Hofstra (Hempstead NY)
M[670]="Unknown" # "Forest Research Center" generic
M[671]="Massachusetts" # Global Water Policy Project (Amherst MA)
M[672]="Texas" # Global Wildlife Conservation (Austin TX) = Re:wild
M[673]="Unknown" # "Graduate Program in Ecology and Evolutionary Biology" generic

M[674]="Rhode Island" # URI College of Engineering
M[675]="Washington" # Evergreen State College (Olympia WA)
M[676]="Virginia" # Falls Church (VA)
M[677]="Florida" # Florida Atlantic University (Boca Raton)
M[678]="South Dakota" # Ecdysis Foundation (Estelline SD)
M[679]="New York" # Environmental Defense Fund (NYC HQ)
M[680]="Unknown" # Dept of Integrative Biol generic -> Unknown
M[681]="Arkansas" # Donaghey College of STEM (U Arkansas Little Rock)
M[682]="North Carolina" # Duke University School of Medicine
M[683]="Virginia" # Duke and Virginia Tech Universities -> ambiguous, VT in VA, Duke in NC -> Unknown
M[684]="California" # D'Arrigo Bros of California (Salinas CA)
M[685]="New York" # Earth Institute (Columbia)
M[686]="North Carolina" # East Carolina University (Greenville NC)
M[687]="Mississippi" # U Southern Mississippi
M[688]="Unknown" # University of St Andrews (Scotland, UK) -> Unknown
M[689]="Tennessee" # U Tennessee (Knoxville)
M[690]="Tennessee" # UT College of Veterinary Medicine
M[691]="Tennessee" # UT Institute of Agriculture
M[692]="Oregon" # Decision Research (Eugene OR)
M[693]="Unknown" # "Department of Chemical Engineering" generic
M[694]="Unknown" # "Department of Ecology and Evolution" generic
M[695]="Illinois" # Dept of Ecology and Evolution, U Chicago
M[696]="Unknown" # "Department of Ecology, Behavior & Evolution" generic (UCSD) -> California
M[697]="New Jersey" # Dept of Ecology, Evolution, and Natural Resources (Rutgers)
M[698]="Oregon" # Conservation Biology Institute (Corvallis OR)
M[699]="Unknown" # Conservation Science Global - unclear
M[700]="Unknown" # "Conservation Science Program" generic
M[701]="Unknown" # "College of Engineering and Computer Science" generic
M[702]="Unknown" # "College of Engineering and Physical Sciences" generic
M[703]="Unknown" # "College of Engineering and Science" generic
M[704]="Unknown" # "College of Liberal Arts and Sciences" generic
M[705]="Unknown" # "College of Natural Sciences" generic
M[706]="Unknown" # "College of Science and Engineering" generic
M[707]="Tennessee" # Vanderbilt Law School (Nashville)
M[708]="California" # CASAS Global (Kensington CA)
M[709]="New Hampshire" # ChairUniversity of New Hampshire
M[710]="California" # Chapman University (Orange CA)
M[711]="Florida" # Coconut Grove (Miami FL)
M[712]="California" # CSU Bakersfield
M[713]="California" # CSU Los Angeles
M[714]="California" # Audubon Canyon Ranch (Marshall CA)
M[715]="Maine" # Bates College (Lewiston ME)
M[716]="Wyoming" # Brain Chemistry Labs (Jackson WY)
M[717]="Washington" # Washington State University (Pullman)
M[718]="Washington" # WSU Puyallup Research and Extension Center
M[719]="Kansas" # Weed Science Society of America -> Unknown (varies) -> Unknown
M[720]="Wyoming" # Western EcoSystems Technology Inc (Cheyenne WY)
M[721]="Wyoming" # Western Ecosystems Technology Incorporated (Cheyenne WY)
M[722]="Michigan" # Western Michigan's College of Engineering
M[723]="Montana" # Whitebark Pine Ecosystem Foundation (Missoula MT)
M[724]="California" # WildAid (San Francisco CA)
M[725]="California" # XCEL Engineering Inc (Oak Ridge TN actually) -> Tennessee
M[726]="Alaska" # Alaska Science Center (USGS, Anchorage AK)
M[727]="Massachusetts" # AIR Worldwide Corporation (Boston MA)
M[728]="Louisiana" # A.B. Freeman School of Business (Tulane, New Orleans)

Corrections / overrides after reflection:

M[233]="Washington" # NOAA Alaska Fisheries Science Center HQ is Seattle WA
M[327]="Oregon" # US EPA NHEERL Western Ecology Division (Corvallis OR)
M[333]="District of Columbia" # RESOLVE (DC)
M[342]="California" # Project Drawdown
M[391]="Unknown" # Universidad de Puerto Rico
M[426]="Federal/Multi-site"
M[427]="Federal/Multi-site"
M[466]="District of Columbia"
M[475]="California"
M[508]="Unknown" # WSSA
M[536]="North Carolina"
M[556]="District of Columbia"
M[575]="Federal/Multi-site"
M[587]="Federal/Multi-site"
M[683]="Unknown"
M[719]="Unknown"
M[725]="Tennessee"

----- CANADA (ids 729-866) -----

M[729]="British Columbia" # UBC (Vancouver)
M[730]="Ontario" # U Toronto
M[731]="Alberta" # U Alberta (Edmonton)
M[732]="Ontario" # U Guelph

M[733]="British Columbia" # Simon Fraser (Burnaby)
M[734]="Quebec" # McGill (Montreal)
M[735]="Federal/Multi-site" # Canadian Forest Service
M[736]="British Columbia" # U Victoria
M[737]="Quebec" # Universite Laval (Quebec City)
M[738]="Ontario" # Queen's (Kingston)
M[739]="Alberta" # U Calgary
M[740]="Nova Scotia" # Dalhousie (Halifax)
M[741]="Ontario" # Western University (London ON)
M[742]="New Brunswick" # U New Brunswick (Fredericton)
M[743]="Ontario" # Carleton (Ottawa)
M[744]="Saskatchewan" # U Saskatchewan (Saskatoon)
M[745]="Quebec" # UQAM (Montreal)
M[746]="Ontario" # U Waterloo
M[747]="Ontario" # U Ottawa
M[748]="Quebec" # U Sherbrooke
M[749]="Ontario" # Trent (Peterborough ON)
M[750]="Manitoba" # U Manitoba (Winnipeg)
M[751]="Ontario" # York (Toronto)
M[752]="Quebec" # FPInnovations (HQ Pointe-Claire QC) -> Quebec
M[753]="Quebec" # FPInnovations - Paprican (Pointe-Claire QC)
M[754]="Federal/Multi-site" # Environment and Climate Change Canada
M[755]="Newfoundland and Labrador" # Memorial University (St. John's)
M[756]="Quebec" # U Montreal
M[757]="Alberta" # U Lethbridge
M[758]="Federal/Multi-site" # Fisheries and Oceans Canada
M[759]="Ontario" # Great Lakes Forestry Centre (CFS, Sault Ste. Marie ON)
M[760]="British Columbia" # Ministry of Forests and Range (BC)
M[761]="Ontario" # Lakehead (Thunder Bay ON)
M[762]="British Columbia" # Pacific Forestry Centre (CFS, Victoria BC)
M[763]="Nova Scotia" # Dalhousie Faculty of Medicine
M[764]="Federal/Multi-site" # Agriculture and Agri-Food Canada
M[765]="Quebec" # Centre d'Etudes Nordiques Rimouski (QC)
M[766]="British Columbia" # UBC Okanagan (Kelowna)
M[767]="British Columbia" # U Northern BC (Prince George)
M[768]="Saskatchewan" # U Regina
M[769]="British Columbia" # Pacific Biological Station DFO (Nanaimo BC)
M[770]="Quebec" # UQAC (Chicoutimi)
M[771]="Ontario" # Brock (St. Catharines ON)
M[772]="Alberta" # Lethbridge Research and Development Centre (AAFC, Lethbridge AB)
M[773]="Quebec" # Concordia (Montreal)
M[774]="Ontario" # U Western Ontario (London)
M[775]="Quebec" # UQTR (Trois-Rivieres)
M[776]="Federal/Multi-site" # Environment Canada
M[777]="Alberta" # Northern Forestry Centre (CFS, Edmonton AB)
M[778]="Quebec" # UQAT (Rouyn-Noranda QC)
M[779]="Ontario" # Ontario Agricultural College (U Guelph)
M[780]="Ontario" # U Windsor
M[781]="Nova Scotia" # Acadia (Wolfville NS)
M[782]="Ontario" # McMaster (Hamilton ON)
M[783]="Quebec" # Ministere des Forets (Quebec)
M[784]="Ontario" # Ontario Ministry of Natural Resources
M[785]="Quebec" # UQAT
M[786]="Alberta" # Alberta Research Council (Edmonton)
M[787]="British Columbia" # Royal Roads University (Victoria BC)
M[788]="New Brunswick" # St. Andrews Biological Station DFO (St. Andrews NB)
M[789]="Ontario" # Ottawa Research and Development Centre (AAFC, Ottawa)
M[790]="British Columbia" # Wildlife Genetics International (Nelson BC)
M[791]="Ontario" # Musee Canadien de la Nature (Ottawa/Gatineau) -> collections Gatineau QC, HQ Ottawa
ON -> Ontario
M[792]="Unknown" # BHP Billiton (mining multinational) -> Unknown
M[793]="Nova Scotia" # Atlantic Canada Conservation Data Centre (Sackville NB actually) -> New Brunswick
M[794]="British Columbia" # Fibria Innovations Ltd (Burnaby BC)
M[795]="Ontario" # Wilfrid Laurier (Waterloo ON)
M[796]="Quebec" # UQAR (Rimouski)
M[797]="Unknown" # JHB Consulting - unclear
M[798]="Quebec" # Ministere des Forets de la Faune et des Parcs du Quebec
M[799]="Federal/Multi-site" # Natural Resources Canada
M[800]="Unknown" # Nursery To Forest Solutions - unclear (Ontario likely) -> Ontario
M[801]="Quebec" # Polytechnique Montreal
M[802]="Nova Scotia" # Bedford Institute of Oceanography DFO (Dartmouth NS)
M[803]="British Columbia" # Trinity Western University (Langley BC)
M[804]="Alberta" # Wild Rose Fire Behaviour (Alberta)
M[805]="Ontario" # Universite Laurentienne (Sudbury ON)
M[806]="Unknown" # Resilience Center - unclear
M[807]="Ontario" # BJ Stocks Wildfire Investigations (Sault Ste Marie ON)
M[808]="British Columbia" # Natural Resources and Environmental Studies Institute (UNBC)
M[809]="New Brunswick" # Canadian Rivers Institute (UNB, Fredericton NB)
M[810]="Manitoba" # Freshwater Institute DFO (Winnipeg MB)
M[811]="Ontario" # Institute of Environmental and Interdisciplinary Science (unclear) -> Unknown
M[812]="Quebec" # McGill Macdonald Campus (Ste-Anne-de-Bellevue QC)

M[813]="Ontario" # Musee canadien de la nature -> Ontario
M[814]="New Brunswick" # Mount Allison (Sackville NB)
M[815]="Quebec" # UQO (Gatineau QC)
M[816]="British Columbia" # Thompson Rivers University (Kamloops BC)
M[817]="Saskatchewan" # U Saskatchewan Western College of Veterinary Medicine
M[818]="Federal/Multi-site" # Agriculture et Agroalimentaire Canada
M[819]="Ontario" # McMaster Faculty of Science
M[820]="New Brunswick" # J.D. Irving Limited (Saint John NB)
M[821]="Nova Scotia" # Faculty of Agriculture Dalhousie (Truro NS)
M[822]="Federal/Multi-site" # Geological Survey of Canada
M[823]="British Columbia" # Pacific Science Enterprise Centre (DFO, West Vancouver BC)
M[824]="British Columbia" # Vancouver Island University (Nanaimo BC)
M[825]="Ontario" # Royal Ontario Museum (Toronto)
M[826]="British Columbia" # Laskeek Bay Conservation Society (Haida Gwaii BC)
M[827]="Ontario" # Great Lakes Institute for Environmental Research (U Windsor)
M[828]="Nova Scotia" # Dalhousie Faculty of Management
M[829]="Ontario" # Institute of Environmental Change & Society (U Regina actually) -> Saskatchewan
M[830]="Unknown" # Adaptive Knowledge Management - unclear
M[831]="Alberta" # Arctic Institute of North America (U Calgary)
M[832]="Alberta" # Alberta Forestry and Park
M[833]="Alberta" # Alberta Caribou Committee
M[834]="Ontario" # Biodiversity Institute of Ontario (U Guelph)
M[835]="Ontario" # Ontario Forest Research Institute (Sault Ste Marie)
M[836]="Ontario" # NurseryToForest Solutions -> Ontario
M[837]="Newfoundland and Labrador" # Northwest Atlantic Fisheries Centre DFO (St. John's NL)
M[838]="Quebec" # Ministere des Ressources naturelles et des Forets du Quebec
M[839]="Quebec" # MRNF
M[840]="Quebec" # Ministere des Ressources Naturelles et des Forets
M[841]="Unknown" # "MRC" generic
M[842]="Ontario" # McMaster Faculty of Engineering
M[843]="Unknown" # "Ltd." generic
M[844]="Alberta" # McEachran Institute (Alberta) -> Unknown; associated with veterinary -> Unknown
M[845]="British Columbia" # Laskeek Bay Conservation Society Office (BC)
M[846]="Quebec" # CHU de Quebec - Universite Laval
M[847]="British Columbia" # Government of British Columbia
M[848]="Alberta" # Government of Alberta
M[849]="Unknown" # GWA Forestry and Applied Biosciences Consulting - unclear
M[850]="Unknown" # "Independent Scholar"
M[851]="Unknown" # "Department of Biology" generic
M[852]="Quebec" # Dairy and Swine Research and Development Centre (AAFC, Sherbrooke QC)
M[853]="Ontario" # Cascadero Copper Corporation (Toronto ON) -> Ontario
M[854]="Quebec" # Loyola Sustainability Research Centre (Concordia, Montreal)
M[855]="Unknown" # Statistical Scientific Editing - unclear
M[856]="British Columbia" # Summerland Research and Development Centre (AAFC, Summerland BC)
M[857]="Quebec" # Saint-Jean-sur-Richelieu Research and Development Centre (AAFC, QC)
M[858]="Nova Scotia" # Shark Research Foundation Inc -> Unknown; based NS? -> Unknown
M[859]="Ontario" # Research and Collections Canadian Museum of Nature -> Ontario
M[860]="Ontario" # Ontario Ministry of the Environment
M[861]="Quebec" # Quebec Research and Development Centre (AAFC, Quebec City)
M[862]="British Columbia" # Southwest British Columbia Grizzly Bear Project
M[863]="Unknown" # The Peoples Seas Initiative - unclear
M[864]="Nova Scotia" # Saint Mary's University (Halifax)
M[865]="Unknown" # University of Maine (this is under Canada country tag but is US Maine) -> Maine
M[866]="Yukon" # Yukon University (Whitehorse)

Canada corrections

M[791]="Ontario"
M[793]="New Brunswick" # ACCDC HQ Sackville NB
M[800]="Ontario"
M[811]="Unknown"
M[813]="Ontario"
M[829]="Saskatchewan"
M[830]="Unknown"
M[844]="Unknown"
M[849]="Unknown"
M[853]="Ontario"
M[858]="Unknown"
M[859]="Ontario"
M[863]="Unknown"
M[865]="Maine"

----- AUSTRALIA (ids 867-997) -----
M[867]="Queensland" # James Cook University (Townsville)
M[868]="Queensland" # U Queensland (Brisbane)
M[869]="Federal/Multi-site" # CSIRO
M[870]="New South Wales" # U Sydney
M[871]="Tasmania" # U Tasmania (Hobart)
M[872]="New South Wales" # UNSW Sydney
M[873]="Australian Capital Territory" # ANU (Canberra)
M[874]="Victoria" # Monash (Melbourne)

M[875]="New South Wales" # Macquarie (Sydney)
M[876]="Queensland" # Griffith (Brisbane/Gold Coast)
M[877]="Victoria" # U Melbourne
M[878]="Western Australia" # UWA (Perth)
M[879]="Australian Capital Territory" # ANU Research School of Biology
M[880]="Victoria" # La Trobe (Melbourne)
M[881]="Australian Capital Territory" # Fenner School (ANU)
M[882]="New South Wales" # Southern Cross University (Lismore NSW)
M[883]="Queensland" # Australian Institute of Marine Science (Townsville QLD)
M[884]="South Australia" # U Adelaide
M[885]="Australian Capital Territory" # CSIRO Land and Water -> CSIRO division -> Federal/Multi-site
M[886]="Victoria" # Deakin School of Life and Environmental Sciences (Geelong VIC)
M[887]="South Australia" # Flinders (Adelaide)
M[888]="New South Wales" # Charles Sturt (Bathurst NSW)
M[889]="New South Wales" # Hawkesbury Institute for the Environment (Western Sydney U, NSW)
M[890]="New South Wales" # U New England Australia (Armidale NSW)
M[891]="Federal/Multi-site" # CSIRO
M[892]="Northern Territory" # Charles Darwin University (Darwin NT)
M[893]="Queensland" # QUT (Brisbane)
M[894]="Australian Capital Territory" # U Canberra
M[895]="New South Wales" # Western Sydney University
M[896]="Western Australia" # Curtin (Perth)
M[897]="Victoria" # School of Ecosystem and Forest Science (U Melbourne)
M[898]="New South Wales" # NSW Department of Primary Industries
M[899]="Victoria" # Deakin University (Geelong VIC)
M[900]="Tasmania" # Institute for Marine and Antarctic Studies (U Tasmania, Hobart)
M[901]="New South Wales" # University of Technology Sydney
M[902]="Unknown" # "School of Biosciences" generic (U Melbourne likely) -> Victoria
M[903]="Western Australia" # Murdoch (Perth)
M[904]="Queensland" # U Sunshine Coast (QLD)
M[905]="Federal/Multi-site" # CSIRO Sustainable Ecosystems
M[906]="Federal/Multi-site" # CSIRO Oceans and Atmosphere
M[907]="New South Wales" # U Wollongong (NSW)
M[908]="Western Australia" # Conservation and Attractions (WA DBCA)
M[909]="Victoria" # RMIT (Melbourne)
M[910]="Federal/Multi-site" # CSIRO Forestry and Forest Products
M[911]="Victoria" # Swinburne (Melbourne)
M[912]="Unknown" # "School of Geography" generic
M[913]="New South Wales" # Australian Museum (Sydney)
M[914]="Unknown" # Consulting Geomorphologist - unclear
M[915]="Queensland" # U Southern Queensland (Toowoomba)
M[916]="Victoria" # Faculty of Medicine Nursing and Health Sciences (Monash)
M[917]="Western Australia" # Western Australian Museum (Perth)
M[918]="Northern Territory" # American-Australian Scientific Expedition to Arnhem Land (NT) -> historic; Arnhem Land NT
M[919]="South Australia" # Adelaide Dental School (U Adelaide)
M[920]="Queensland" # Reef and Rainforest Research Centre (Cairns QLD)
M[921]="Unknown" # EastCoast Geospatial Consultants - unclear
M[922]="New South Wales" # School of Environmental and Life Sciences (U Newcastle NSW)
M[923]="Unknown" # Environmental Sensing Systems - unclear
M[924]="Queensland" # Department of Environment and Science (QLD)
M[925]="New South Wales" # Faculty of Science, Medicine and Health (U Wollongong)
M[926]="Tasmania" # Tasmanian Institute of Agriculture (U Tasmania)
M[927]="Unknown" # THE MULLION GROUP PTY LTD (Canberra ACT) -> Australian Capital Territory
M[928]="South Australia" # CRC for Australian Weed Management (Adelaide SA)
M[929]="Unknown" # "Whithycombe" garbled -> Unknown
M[930]="Unknown" # "Faculty of Science" generic
M[931]="Unknown" # Forsci Pty Ltd - unclear (Tasmania forestry) -> Tasmania? -> Unknown
M[932]="Queensland" # AJ Ash and Associates (rangeland, QLD) -> Unknown
M[933]="Unknown" # Crop Variety Testing Group - unclear
M[934]="Queensland" # Department of Transport and Main Roads (QLD)
M[935]="Victoria" # Forest and Wood Products Australia (Melbourne VIC)
M[936]="Unknown" # "College of Science and Engineering" generic
M[937]="Federal/Multi-site" # CSIRO Ecosystem Sciences
M[938]="Federal/Multi-site" # CSIRO Plant Industry
M[939]="New South Wales" # U Sydney Business School
M[940]="Tasmania" # Tasmanian Museum and Art Gallery (Hobart)
M[941]="New South Wales" # Taronga Conservation Society Australia (Sydney)
M[942]="Tasmania" # Tasmania Museum and Art Gallery
M[943]="Unknown" # "1 Caddy Avenue" address -> Unknown
M[944]="Australian Capital Territory" # Institute for Applied Ecology (U Canberra)
M[945]="Queensland" # Queensland Department of Agriculture and Fisheries
M[946]="Queensland" # Queensland Herbarium (Brisbane)
M[947]="South Australia" # South Australian Research and Development Institute
M[948]="New South Wales" # NSW Rural Fire Service
M[949]="New South Wales" # Port Stephens Fisheries Institute (NSW DPI)
M[950]="Victoria" # Faculty of Veterinary and Agricultural Sciences (U Melbourne)
M[951]="Queensland" # Sugarbag Bees (Brisbane QLD)
M[952]="New South Wales" # U Newcastle Australia (NSW)
M[953]="Unknown" # Hydrology and Catchment Management - unclear
M[954]="Tasmania" # Australian Antarctic Division (Kingston TAS)

```

M[955]="Unknown"                # Chessman Ecology - unclear (NSW) -> Unknown
M[956]="Queensland"             # CQUniversity Australia (Rockhampton QLD)
M[957]="New South Wales"        # Forest Science, Department of Primary Industries (NSW)
M[958]="Australian Capital Territory" # "Australian Capital Territory" literal
M[959]="Unknown"                # FORSCI Pty. Ltd. -> Unknown
M[960]="Australian Capital Territory" # Geoscience Australia (Canberra ACT)
M[961]="Unknown"                # "School of Mathematics and Statistics" generic
M[962]="Unknown"                # Silviscan Pty Ltd -> Unknown
M[963]="Australian Capital Territory" # Australian Commissioner to the International Whaling Commission ->
Unknown
M[964]="Queensland"             # A.J. Ash -> Unknown
M[965]="Queensland"             # Bond University (Gold Coast QLD)
M[966]="Unknown"                # "Independent Scholar"
M[967]="Victoria"               # Global Evergreening Alliance (Melbourne VIC)
M[968]="Unknown"                # Forest and fire science consultant
M[969]="Unknown"                # Forest Hydrology and Catchment Management Consultant
M[970]="Unknown"                # "Forest Science" generic
M[971]="Unknown"                # "Faculty of Medicine and Health" generic
M[972]="Tasmania"               # CCAMLR (Hobart TAS)
M[973]="New South Wales"        # Australian Cotton Research Institute (Narrabri NSW)
M[974]="Victoria"               # Bushfire and Natural Hazards CRC (Melbourne VIC)
M[975]="Victoria"               # Bureau of Meteorology Australia (Melbourne HQ)
M[976]="Queensland"             # CSIRO Agriculture Flagship Brisbane -> Federal/Multi-site (CSIRO)
M[977]="Queensland"             # CQUniversity Brisbane
M[978]="Victoria"               # CSIRO Futures -> Federal/Multi-site
M[979]="New South Wales"        # CSIRO Plant Industry and CRC Weed Management -> Federal/Multi-site
M[980]="Western Australia"      # Cervantes Agritech Pty Limited (Cervantes WA)
M[981]="New South Wales"        # Charles Sturt University Wagga Wagga (NSW)
M[982]="Victoria"               # Agriculture Victoria
M[983]="Queensland"             # ARC Centre of Excellence for Coral Reef Studies (JCU, Townsville QLD)
M[984]="Unknown"                # Aquamarine Australia - unclear
M[985]="Western Australia"      # Science and Conservation Division (WA DBCA)
M[986]="Western Australia"      # Minderoo Foundation (Perth WA)
M[987]="New South Wales"        # NSW Primary Industries
M[988]="Queensland"             # Queensland Department of Environment and Science
M[989]="Queensland"             # Queensland Department of Primary Industries
M[990]="Unknown"                # Know Your Wood - unclear
M[991]="Queensland"             # Queensland Museum (Brisbane)
M[992]="Unknown"                # R. E. Johannes Pty Ltd (Tasmania) -> Tasmania
M[993]="Unknown"                # Ross Analytics Pty Ltd - unclear
M[994]="Queensland"             # U Queensland Remote Sensing Research Centre
M[995]="Unknown"                # "The Faculty of Science & Engineering" generic
M[996]="New South Wales"        # Sydney Institute of Marine Science
M[997]="Unknown"                # "Withycombe" garbled -> Unknown

# Australia corrections
M[885]="Federal/Multi-site"
M[902]="Victoria"
M[905]="Federal/Multi-site"
M[906]="Federal/Multi-site"
M[910]="Federal/Multi-site"
M[918]="Northern Territory"
M[927]="Australian Capital Territory"
M[931]="Unknown"
M[932]="Unknown"
M[937]="Federal/Multi-site"
M[938]="Federal/Multi-site"
M[963]="Unknown"
M[964]="Unknown"
M[976]="Federal/Multi-site"
M[978]="Federal/Multi-site"
M[979]="Federal/Multi-site"
M[992]="Tasmania"

import json
print(json.dumps(M))

```

45. peek.py (SHA-256 353d5d93282786ea...; 14 lines)

```

import openpyxl, sys

for year in ("2020", "2024"):
    path = f"singleyr_{year}.xlsx"
    wb = openpyxl.load_workbook(path, read_only=True)
    print(f"== {path} sheets: {wb.sheetnames}")
    for name in wb.sheetnames:
        ws = wb[name]
        print(f"-- sheet '{name}' dims={ws.max_row}x{ws.max_column}")

```

```

        for i, row in enumerate(ws.iter_rows(max_row=3, values_only=True)):
            print(" ", [str(v)[:40] for v in row][:45])
            if i >= 2:
                break
wb.close()

```

46. probe_final.py (SHA-256 5c05bd017739e88d...; 75 lines)

```

# -*- coding: utf-8 -*-
# Final corroboration: exclude blank inst/cntry the way the pipeline does, recount distinct dims,
# and verify full continent nps/ncs + banded heads.
import sys, json
from collections import defaultdict
import pandas as pd
sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford")
from cc_defs import CC
XLSX = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
JDIR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
YEARS=["2020","2021","2022","2023","2024"]

df=pd.read_excel(XLSX,sheet_name="Data",engine="openpyxl")
df.columns=[str(c).strip() for c in df.columns]
for c in ("nps","ncs"): df[c]=pd.to_numeric(df[c],errors="coerce").fillna(0)
df["sm1"]=df["sm-subfield-1"].astype(str).str.strip()
df["sm2"]=df["sm-subfield-2"].astype(str).str.strip()

def blank(v): return pd.isna(v) or str(v).strip().lower() in ("","nan","none")

def run(name, sub, jfile):
    J=json.load(open(f"{JDIR}\\{jfile}",encoding="utf-8"))
    # distinct institutions EXCLUDING blank
    insts=set(str(x).strip() for x in sub["inst_name"] if not blank(x))
    j_insts=set(rec["key"] for rec in J["university"]["scientists"])
    print(f"{name}: distinct institutions (excl blank) recomp={len(insts)} json={len(j_insts)} "
          f"'{OK}' if insts==j_insts else 'DIFF'")
    if insts!=j_insts:
        print(" only-recomp:",sorted(insts-j_insts)[:5],"only-json:",sorted(j_insts-insts)[:5])
    # distinct countries excluding blank/unmapped
    ctys=set()
    for v in sub["cntry"]:
        if blank(v): continue
        code=str(v).strip().lower()
        if code in CC: ctys.add(CC[code][0])
    j_ctys=set(rec["key"] for rec in J["country"]["scientists"])
    print(f"{name}: distinct countries (excl blank) recomp={len(ctys)} json={len(j_ctys)} "
          f"'{OK}' if ctys==j_ctys else 'DIFF'")

    # full continent nps & ncs reconcile
    def cont(v):
        c=str(v).strip().lower()
        return CC[c][1] if c in CC else None
    cn_nps=defaultdict(lambda:defaultdict(int)); cn_ncs=defaultdict(lambda:defaultdict(int))
    for _,r in sub.iterrows():
        co=cont(r["cntry"])
        if co is None: continue
        y=str(int(r["year"]))
        cn_nps[co][y]+=int(r["nps"]); cn_ncs[co][y]+=int(r["ncs"])
    jnps={rec["key"]:rec for rec in J["continent"]["publications"]}
    jncs={rec["key"]:rec for rec in J["continent"]["citations"]}
    bad=0
    for co in set(cn_nps)|set(jnps):
        for y in YEARS:
            if cn_nps.get(co,{}).get(y,0)!=jnps.get(co,{}).get(y,0):
                print(f" NPS MISMATCH {co} {y}"); bad+=1
            if cn_ncs.get(co,{}).get(y,0)!=jncs.get(co,{}).get(y,0):
                print(f" NCS MISMATCH {co} {y}"); bad+=1
    print(f"{name}: continent NPS+NCS all-years reconcile: '{OK}' if bad==0 else str(bad)+' mismatches'")

    # banded university head top-25 by scientists: compare key+2024 order
    head=J["university_banded"]["scientists"]["head"]
    # recompute top-25 by 2024 scientists excluding blank
    isci=defaultdict(lambda:defaultdict(int))
    for _,r in sub.iterrows():
        if blank(r["inst_name"]): continue
        isci[str(r["inst_name"]).strip()][str(int(r["year"]))]+=1

```

```

# json head is top-25 by 'total' (scientists total across years) per context? verify by total
recomp_by_total=sorted(isci.keys(), key=lambda k:(-sum(isci[k].get(y,0) for y in YEARS),k))[:25]
jhead_keys=[h["key"] for h in head]
match_total=sum(1 for k in recomp_by_total if k in jhead_keys)
print(f"{name}: banded univ head top-25 (by total) overlap with json head = {match_total}/25")

run("Forestry-SA", df[(df["sm1"]=="Forestry")&(df["nps"]>0)], "forestry_B_data.json")
run("Ecology-SA", df[(df["sm1"]=="Ecology")&(df["sm2"]=="Forestry")&(df["nps"]>0)], "forestry_C_data.json")

```

47. probe_head.py (SHA-256 b42741bca2eb636a...; 44 lines)

```

# -*- coding: utf-8 -*-
# Verify the JSON banded university head is internally consistent: each head record's per-year
# scientists must equal our raw recount for that institution. That is the real test (not ranking).
import sys, json
from collections import defaultdict
import pandas as pd
sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford")
XLSX = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
JDIR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
YEARS=["2020","2021","2022","2023","2024"]
df=pd.read_excel(XLSX,sheet_name="Data",engine="openpyxl")
df.columns=[str(c).strip() for c in df.columns]
for c in ("nps","ncs"): df[c]=pd.to_numeric(df[c],errors="coerce").fillna(0)
df["sm1"]=df["sm-subfield-1"].astype(str).str.strip()
df["sm2"]=df["sm-subfield-2"].astype(str).str.strip()
def blank(v): return pd.isna(v) or str(v).strip().lower() in ("","nan","none")

def check(name, sub, jfile):
    J=json.load(open(f"{JDIR}\\{jfile}",encoding="utf-8"))
    isci=defaultdict(lambda:defaultdict(int)); inps=defaultdict(lambda:defaultdict(int));
    incs=defaultdict(lambda:defaultdict(int))
    for _,r in sub.iterrows():
        if blank(r["inst_name"]): continue
        k=str(r["inst_name"]).strip(); y=str(int(r["year"]))
        isci[k][y]+=1; inps[k][y]+=int(r["nps"]); incs[k][y]+=int(r["ncs"])
    bad=0
    for metric,src in (("scientists",isci),("publications",inps),("citations",incs)):
        for h in J["university_banded"][metric]["head"]:
            k=h["key"]
            for y in YEARS:
                if src.get(k,{}).get(y,0)!=h.get(y,0):
                    print(f"    {name} banded[{metric}] {k} {y}: recomp={src.get(k,{}).get(y,0)}")
    json={h.get(y)}; bad+=1
    rt=sum(src.get(k,{}).get(y,0) for y in YEARS)
    if rt!=h.get("total"):
        print(f"    {name} banded[{metric}] {k} TOTAL recomp={rt} json={h.get('total')}"); bad+=1
    # verify head is truly the top-25 by scientists 2024-count -> confirm the 25th >= any excluded
    sci_head=J["university_banded"]["scientists"]["head"]
    head_keys=set(h["key"] for h in sci_head)
    # rank all insts the way spec says: by scientists (use total across years as the volume measure the head carries)
    print(f"{name}: banded university head all-metrics per-year+total reconcile: {'OK' if bad==0 else str(bad)+' mismatches'}")
    # also country banded head
    return bad

check("Forestry-SA", df[(df["sm1"]=="Forestry")&(df["nps"]>0)], "forestry_B_data.json")
check("Ecology-SA", df[(df["sm1"]=="Ecology")&(df["sm2"]=="Forestry")&(df["nps"]>0)], "forestry_C_data.json")

```

48. probe_nan.py (SHA-256 88283e6b3479db09...; 44 lines)

```

# -*- coding: utf-8 -*-
import sys, json
import pandas as pd
sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford")
from cc_defs import CC
XLSX = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
JDIR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"

df = pd.read_excel(XLSX, sheet_name="Data", engine="openpyxl")
df.columns=[str(c).strip() for c in df.columns]

```

```

df["nps"]=pd.to_numeric(df["nps"],errors="coerce").fillna(0)
df["sm1"]=df["sm-subfield-1"].astype(str).str.strip()
fsa=df[(df["sm1"]=="Forestry") & (df["nps"]>0)].copy()

# rows with blank/nan inst_name
blank = fsa[fsa["inst_name"].isna() | (fsa["inst_name"].astype(str).str.strip().isin(["", "nan", "None"]))]
print("Forestry-SA rows with blank/nan inst_name:", len(blank))
cols=["year", "authfull", "inst_name", "cntry", "nps", "ncs"]
print(blank[cols].to_string())

# also blank cntry
blankc = fsa[fsa["cntry"].isna() | (fsa["cntry"].astype(str).str.strip().isin(["", "nan", "None"]))]
print("\nForestry-SA rows with blank/nan cntry:", len(blankc))
print(blankc[cols].to_string())

# Does JSON university dim contain any 'nan'/blank key?
JB=json.load(open(f"{JDIR}\\forestry_B_data.json",encoding="utf-8"))
ukeys=[rec["key"] for rec in JB["university"]["scientists"]]
suspicious=[k for k in ukeys if str(k).strip().lower() in ("nan", "none", "") ]
print("\nJSON B university keys that look blank/nan:", suspicious)

# sum of scientists across university dim per year vs totals (does dropping nan-inst break university totals?)
usci=JB["university"]["scientists"]
sum24=sum(rec.get("2024",0) for rec in usci)
print("JSON B: sum of university 2024 scientists =", sum24, " totals 2024 =", JB["totals"]["scientists"]["2024"])
for y in ["2020", "2021", "2022", "2023", "2024"]:
    s=sum(rec.get(y,0) for rec in usci)
    print(f" university sum {y} = {s} totals {y} = {JB['totals']['scientists'][y]} diff={JB['totals']
['scientists'][y]-s}")

# same check for country dim
csci=JB["country"]["scientists"]
for y in ["2020", "2021", "2022", "2023", "2024"]:
    s=sum(rec.get(y,0) for rec in csci)
    print(f" country sum {y} = {s} totals {y} = {JB['totals']['scientists'][y]} diff={JB['totals']
['scientists'][y]-s}")

```

49. process.py (SHA-256 4823abd40be58dae...; 135 lines)

```

import re
import openpyxl
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill
from openpyxl.utils import get_column_letter

META = {
    "2020": ("3", "Table_1_Authors_singleyr_2020_wopp_extracted_202108.xlsx"),
    "2021": ("5", "Table_1_Authors_singleyr_2021_pubs_since_1788_wopp_extracted_202209b.xlsx"),
    "2022": ("6", "Table_1_Authors_singleyr_2022_pubs_since_1788_wopp_extracted_202310.xlsx"),
    "2023": ("7", "Table_1_Authors_singleyr_2023_pubs_since_1788_wopp_extracted_202408.xlsx"),
    "2024": ("8", "Table_1_Authors_singleyr_2024_pubs_since_1788_wopp_extracted_202508.xlsx"),
}

def norm(col):
    c = str(col).strip()
    c = re.sub(r"^hm\d\d", "hm", c)
    c = re.sub(r"^h\d\d", "h", c)
    c = re.sub(r"^np60\d\d", "np", c)
    c = re.sub(r"^nc\d{4}", "nc", c)
    c = re.sub(r"cited\d{4}", "cited", c)
    return c

union_cols = []
rows_by_year = {}
totals = {}
key_desc = {}

for year in META:
    path = f"singleyr_{year}.xlsx"
    wb = openpyxl.load_workbook(path, read_only=True)

    kws = wb["Key"]
    for r in kws.iter_rows(min_row=2, values_only=True):
        if r and r[0]:
            key_desc.setdefault(norm(r[0]), r[2] if len(r) > 2 else None)

```

```

ws = wb["Data"]
it = ws.iter_rows(values_only=True)
header = [norm(c) for c in next(it)]
for c in header:
    if c not in union_cols:
        union_cols.append(c)
i1 = header.index("sm-subfield-1")
i2 = header.index("sm-subfield-2")

matched, total = [], 0
for row in it:
    if row is None or all(v is None for v in row):
        continue
    total += 1
    s1 = str(row[i1]).strip().lower() if row[i1] else ""
    s2 = str(row[i2]).strip().lower() if row[i2] else ""
    if s1 == "forestry" or s2 == "forestry":
        matched.append(dict(zip(header, row)))
wb.close()
rows_by_year[year] = matched
totals[year] = total
print(f"{year}: {total} authors scanned, {len(matched)} Forestry rows", flush=True)

out = Workbook()
arial = Font(name="Arial", size=10)
hdr_font = Font(name="Arial", size=10, bold=True, color="FFFFFF")
hdr_fill = PatternFill("solid", start_color="2F5233")

ws = out.active
ws.title = "Forestry"
cols = ["year"] + union_cols
ws.append(cols)
for cell in ws[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
n = 0
for year in META:
    for rec in rows_by_year[year]:
        ws.append([int(year)] + [rec.get(c) for c in union_cols])
        n += 1
for row in ws.iter_rows(min_row=2):
    for cell in row:
        cell.font = arial
ws.freeze_panes = "A2"
ws.auto_filter.ref = f"A1:{get_column_letter(len(cols))}{n+1}"
ws.column_dimensions["B"].width = 28
ws.column_dimensions["C"].width = 34
for idx, cname in enumerate(cols, 1):
    if cname.startswith("sm-"):
        ws.column_dimensions[get_column_letter(idx)].width = 24

kw = out.create_sheet("Key")
kw.append(["FIELD", "DESCRIPTION"])
kw.append(["year", "single-year citation data year (added during aggregation; from file name)"])
for f in union_cols:
    kw.append([f, key_desc.get(f)])
kw.append([])
kw.append(["Note: year-specific column names were normalized across files, e.g. nc2020/nc2424 -> nc, h20/h24 -> h, np6020/np6024 -> np."])
kw.append(["Columns np_d/nc_d appear only in 2020-2022 files; np_rw/nc_rw/nc_to_rw only in 2023-2024 files; blank where not provided."])
for row in kw.iter_rows():
    for cell in row:
        cell.font = arial
for cell in kw[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
kw.column_dimensions["A"].width = 26
kw.column_dimensions["B"].width = 110

sw = out.create_sheet("Sources")
sw.append(["data year", "dataset version", "original file", "authors in file", "Forestry rows extracted"])
for year, (ver, fname) in META.items():
    sw.append([int(year), int(ver), fname, totals[year], len(rows_by_year[year])])
sw.append([])
sw.append(["Dataset: Ioannidis, John P.A. (2025), 'August 2025 data-update for Updated science-wide author databases of standardized citation indicators'"])

```

```

sw.append(["Elsevier Data Repository, doi:10.17632/btchxktyzw - single-year (singleyr) tables from versions 3, 5,
6, 7, 8; downloaded 2026-07-05."])
sw.append(["Filter: sm-subfield-1 = 'Forestry' OR sm-subfield-2 = 'Forestry' (Science-Metrix subfield
classification)."])
sw.append(["Note: each release reports single-year citations for the prior calendar year; no 2025 single-year
table exists as of the August 2025 update (v8)."])
for row in sw.iter_rows():
    for cell in row:
        cell.font = arial
for cell in sw[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
for col, w in zip("ABCDE", (10, 16, 72, 16, 22)):
    sw.column_dimensions[col].width = w

dest = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_singleyr_2020-2024.xlsx"
out.save(dest)
print(f"Saved {dest}: {n} rows, {len(cols)} columns", flush=True)

import csv
with open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_singleyr_2020-2024.csv", "w", newline="",
encoding="utf-8-sig") as f:
    w = csv.writer(f)
    w.writerow(cols)
    for year in META:
        for rec in rows_by_year[year]:
            w.writerow([int(year)] + [rec.get(c) for c in union_cols])
print("CSV saved", flush=True)

```

50. process_career.py (SHA-256 97473629fe0b03d0...; 146 lines)

```

import re
import csv
import openpyxl
from openpyxl import Workbook
from openpyxl.styles import Font, PatternFill
from openpyxl.utils import get_column_letter

META = {
    "2020": ("3", "Table_1_Authors_career_2020_wopp_extracted_202108.xlsx"),
    "2021": ("5", "Table_1_Authors_career_2021_pubs_since_1788_wopp_extracted_202209b.xlsx"),
    "2022": ("6", "Table_1_Authors_career_2022_pubs_since_1788_wopp_extracted_202310.xlsx"),
    "2023": ("7", "Table_1_Authors_career_2023_pubs_since_1788_wopp_extracted_202408.xlsx"),
    "2024": ("8", "Table_1_Authors_career_2024_pubs_since_1788_wopp_extracted_202508.xlsx"),
}
TARGETS = {"forestry", "ecology"}

def norm(col):
    c = str(col).strip()
    c = re.sub(r"^\hm\d\d", "hm", c)
    c = re.sub(r"^\h\d\d", "h", c)
    c = re.sub(r"^\np60\d\d", "np", c)
    c = re.sub(r"^\nc\d{4}", "nc", c)
    c = re.sub(r"^\cited\d{4}", "cited", c)
    return c

union_cols = []
rows_by_year = {}
totals = {}
key_desc = {}

for year in META:
    path = f"career_{year}.xlsx"
    wb = openpyxl.load_workbook(path, read_only=True)

    kws = wb["Key"]
    for r in kws.iter_rows(min_row=2, values_only=True):
        if r and r[0]:
            key_desc.setdefault(norm(r[0]), r[2] if len(r) > 2 else None)

    ws = wb["Data"]
    it = ws.iter_rows(values_only=True)
    header = [norm(c) for c in next(it)]
    for c in header:
        if c not in union_cols:
            union_cols.append(c)

```

```

i1 = header.index("sm-subfield-1")
i2 = header.index("sm-subfield-2")

matched, total = [], 0
for row in it:
    if row is None or all(v is None for v in row):
        continue
    total += 1
    s1 = str(row[i1]).strip().lower() if row[i1] else ""
    s2 = str(row[i2]).strip().lower() if row[i2] else ""
    hit = []
    if s1 in TARGETS: hit.append(row[i1])
    if s2 in TARGETS and (not row[i1] or str(row[i2]).strip().lower() != s1): hit.append(row[i2])
    if hit:
        rec = dict(zip(header, row))
        rec["_match"] = "; ".join(dict.fromkeys(str(h).strip() for h in hit))
        matched.append(rec)
wb.close()
rows_by_year[year] = matched
totals[year] = total
fcount = sum(1 for r in matched if "forestry" in r["_match"].lower())
ecount = sum(1 for r in matched if "ecology" in r["_match"].lower())
print(f"{year}: {total} authors scanned, {len(matched)} matched (Forestry~{fcount}, Ecology~{ecount})",
flush=True)

out = Workbook()
arial = Font(name="Arial", size=10)
hdr_font = Font(name="Arial", size=10, bold=True, color="FFFFFF")
hdr_fill = PatternFill("solid", start_color="2F5233")

ws = out.active
ws.title = "Data"
cols = ["year", "matched_subfield"] + union_cols
ws.append(cols)
for cell in ws[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
n = 0
for year in META:
    for rec in rows_by_year[year]:
        ws.append([int(year), rec["_match"]] + [rec.get(c) for c in union_cols])
        n += 1
for row in ws.iter_rows(min_row=2):
    for cell in row:
        cell.font = arial
ws.freeze_panes = "A2"
ws.auto_filter.ref = f"A1:{get_column_letter(len(cols))}{n+1}"
ws.column_dimensions["B"].width = 18
ws.column_dimensions["C"].width = 28
ws.column_dimensions["D"].width = 34
for idx, cname in enumerate(cols, 1):
    if cname.startswith("sm-"):
        ws.column_dimensions[get_column_letter(idx)].width = 24

kw = out.create_sheet("Key")
kw.append(["FIELD", "DESCRIPTION"])
kw.append(["year", "career-data release year (added during aggregation; from file name). Career metrics are cumulative through end of that year."])
kw.append(["matched_subfield", "which target subfield matched: Forestry and/or Ecology (from sm-subfield-1 / sm-subfield-2)"])
for f in union_cols:
    kw.append([f, key_desc.get(f)])
kw.append([])
kw.append(["Note: year-specific column names normalized across files, e.g. nc9620/nc9624 -> nc, h20/h24 -> h, np6020/np6024 -> np, cited9620/9624 -> cited."])
kw.append(["Columns np_d/nc_d appear only in 2020-2022 files; np_rw/nc_to_rw/nc_rw only in 2023-2024 files; blank where a file did not provide them."])
for row in kw.iter_rows():
    for cell in row:
        cell.font = arial
for cell in kw[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
kw.column_dimensions["A"].width = 26
kw.column_dimensions["B"].width = 110

sw = out.create_sheet("Sources")
sw.append(["data year", "dataset version", "original file", "authors in file", "matched rows extracted"])
for year, (ver, fname) in META.items():

```

```

    sw.append([int(year), int(ver), fname, totals[year], len(rows_by_year[year])])
sw.append([])
sw.append(["Dataset: Ioannidis, John P.A. - 'Updated science-wide author databases of standardized citation
indicators' (Elsevier Data Repository, doi:10.17632/btchxktzyw)."])
sw.append(["Files: Table_1_Authors_career (career-long tables) from versions 3, 5, 6, 7, 8; downloaded 2026-07-
05."])
sw.append(["Filter: sm-subfield-1 = 'Forestry'/'Ecology' OR sm-subfield-2 = 'Forestry'/'Ecology' (Science-Metrix
subfield classification)."])
sw.append(["Category mapping agreed with user: Forestry = subfield 'Forestry'; 'Life Sciences' = subfield
'Ecology' (Biology/Biomedical excluded); 'Environment/Natural Resource Management' dropped (no such category
exists in this taxonomy's 174 subfields)."])
sw.append(["Coverage: data years 2020-2024 are the five annual career releases available. No 2025 data-year table
exists; the most recent update (August 2025, version 8) carries career data through end of 2024."])
for row in sw.iter_rows():
    for cell in row:
        cell.font = arial
for cell in sw[1]:
    cell.font = hdr_font
    cell.fill = hdr_fill
for col, w in zip("ABCDE", (10, 16, 72, 16, 22)):
    sw.column_dimensions[col].width = w

dest = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
out.save(dest)
print(f"Saved {dest}: {n} rows, {len(cols)} columns", flush=True)

with open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.csv", "w", newline="",
encoding="utf-8-sig") as f:
    w = csv.writer(f)
    w.writerow(cols)
    for year in META:
        for rec in rows_by_year[year]:
            w.writerow([int(year), rec["_match"]] + [rec.get(c) for c in union_cols])
print("CSV saved", flush=True)

```

51. ratios.py (SHA-256 9bcaada9266cdb86...; 113 lines)

```

# -*- coding: utf-8 -*-
# Intensity ratios for the forestry cohort: articles/author, citations/author, citations/article.
# Ratios are computed on 5-year SUMS (author-years): art/auth = ΣP/ΣS, cit/auth = ΣC/ΣS, cit/art = ΣC/ΣP.
# Also: cohort & per-year ratios, group (Forestry vs Ecology) ratios, and the quantity-vs-quality
# regularity (correlation of articles/author against citations/article across countries & institutions).
import json, numpy as np
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
YEARS=[2020,2021,2022,2023,2024]; YS=[str(y) for y in YEARS]
M=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
MF=M["forestry"]; MFE=M["forestry_E"]

def ratios_of(S,P,C):
    S=float(S); P=float(P); C=float(C)
    return {"S":S,"P":P,"C":C,"sci_yr":S/5.0,
            "art_per_auth":(P/S if S else 0.0),
            "cit_per_auth":(C/S if S else 0.0),
            "cit_per_art":(C/P if P else 0.0)}

def member_ratios(members, with_country=False):
    out=[]
    for it in members:
        S=sum(it["s"]); P=sum(it["p"]); C=sum(it["c"])
        r={"key":it["key"], **ratios_of(S,P,C)}
        if with_country: r["country"]=it.get("country","")
        out.append(r)
    out.sort(key=lambda r:-r["S"]) # rank by scale (author-years) by default
    return out

def dim_ratios(D, dim, with_country=False):
    return member_ratios(D[dim], with_country=with_country)

def cohort_ratios(D):
    T=D["totals"]
    S=sum(T["scientists"][y] for y in YS); P=sum(T["publications"][y] for y in YS); C=sum(T["citations"][y] for y
in YS)
    overall=ratios_of(S,P,C)
    by_year={"art_per_auth":[],"cit_per_auth":[],"cit_per_art":[]}
    for y in YS:
        s=T["scientists"][y]; p=T["publications"][y]; c=T["citations"][y]

```

```

        by_year["art_per_auth"].append(p/s if s else 0)
        by_year["cit_per_auth"].append(c/s if s else 0)
        by_year["cit_per_art"].append(c/p if p else 0)
    return {"overall":overall,"by_year":by_year}

def spearman(x,y):
    def rank(a):
        order=np.argsort(a); r=np.empty(len(a)); r[order]=np.arange(len(a))
        # average ties
        a=np.asarray(a,float)
        for v in set(a):
            idx=np.where(a==v)[0]
            if len(idx)>1: r[idx]=r[idx].mean()
        return r
    return float(np.corrcoef(rank(x),rank(y))[0,1])

def regularity(members, min_sci_yr=5.0):
    pts=[m for m in members if m["sci_yr"]>=min_sci_yr and m["P"]>0]
    x=np.array([m["art_per_auth"] for m in pts]); y=np.array([m["cit_per_auth"] for m in pts])
    return {"n":len(pts),"min_sci_yr":min_sci_yr,
            "pearson":float(np.corrcoef(x,y)[0,1]) if len(pts)>2 else None,
            "spearman":spearman(x,y) if len(pts)>2 else None,
            # quantity vs total-impact (art/auth vs cit/auth): are prolific authors more cited overall?
            "pearson_q_vs_totimpact":float(np.corrcoef([m["art_per_auth"] for m in pts],[m["cit_per_auth"] for m
in pts])[0,1]) if len(pts)>2 else None}

country_r=dim_ratios(MF,"country")
uni_r=dim_ratios(MF,"university",with_country=True)
OUT={
    "years":YEARS,
    "cohort":cohort_ratios(MF),
    "cohort_E":cohort_ratios(MFE),
    "group":{r["key"]:r for r in member_ratios(MF["group"])},
    "continent":dim_ratios(MF,"continent"),
    "region":dim_ratios(MF,"region"),
    "country":country_r,
    "us_state":dim_ratios(MF,"us_state"),
    "university":uni_r,
    # Ecology-group (Appendix A) dimension ratios
    "E_continent":dim_ratios(MFE,"continent"),
    "E_country":dim_ratios(MFE,"country"),
    "E_group_vs_forestry":{"Forestry":{r["key"]:r for r in member_ratios(MF["group"])},{"Forestry":
        "Ecology":{r["key"]:r for r in member_ratios(MF["group"])},{"Ecology":}},
    "corr":{"country":regularity(country_r,5.0),
            "university":regularity(uni_r,3.0)},
}
json.dump(OUT,open(f"{SCR}/ratios.json","w",encoding="utf-8"))

# ---- console report to ground the prose ----
co=OUT["cohort"]["overall"]
print(f"COHORT overall: {co['art_per_auth']:.1f} articles/author | {co['cit_per_auth']:.0f} citations/author |
{co['cit_per_art']:.1f} citations/article")
print("Cohort by year (art/auth, cit/auth, cit/art):")
for i,y in enumerate(YEARS):
    b=OUT["cohort"]["by_year"]; print(f" {y}: {b['art_per_auth'][i]:.1f} | {b['cit_per_auth'][i]:.0f} |
{b['cit_per_art'][i]:.1f}")
print("\nGROUP:")
for g in ["Forestry","Ecology"]:
    r=OUT["group"][g]; print(f" {g:<9} sci/yr {r['sci_yr']:.0f}: {r['art_per_auth']:.1f} art/auth |
{r['cit_per_auth']:.0f} cit/auth | {r['cit_per_art']:.1f} cit/art")
print("\nCONTINENTS:")
for r in OUT["continent"]:
    print(f" {r['key']:<15} sci/yr {r['sci_yr']:>4.0f}: {r['art_per_auth']:>6.1f} art/auth |
{r['cit_per_auth']:>7.0f} cit/auth | {r['cit_per_art']:>5.1f} cit/art")
print("\nTOP COUNTRIES (by scientists), ratios:")
print(f" { 'country':<20}{ 'sci/yr':>7}{ 'art/auth':>9}{ 'cit/auth':>10}{ 'cit/art':>9}")
for r in OUT["country"][:20]:
    print(f" {r['key']:<20}{r['sci_yr']:>7.0f}{r['art_per_auth']:>9.1f}{r['cit_per_auth']:>10.0f}
{r['cit_per_art']:>9.1f}")
c=OUT["corr"]["country"]; u=OUT["corr"]["university"]
print(f"\nREGULARITY (quantity vs quality):")
print(f" countries (n={c['n']}, >={c['min_sci_yr']:.0f} sci/yr): art/auth vs cit/art Pearson r={c['pearson']:.
+.2f}, Spearman={c['spearman']:.+2f}")
print(f" art/auth vs cit/auth Pearson r={c['pearson_q_vs_totimpact']:.
+.2f}")
print(f" institutions (n={u['n']}, >={u['min_sci_yr']:.0f} sci/yr): art/auth vs cit/art Pearson
r={u['pearson']:.+2f}, Spearman={u['spearman']:.+2f}")
# extremes for prose
cc=[r for r in OUT["country"] if r["sci_yr"]>=5]

```

```

print("\n highest cit/art:",[(r["key"],round(r["cit_per_art"],1)) for r in sorted(cc,key=lambda r:-
r["cit_per_art"])[5]])
print(" lowest cit/art:",[(r["key"],round(r["cit_per_art"],1)) for r in sorted(cc,key=lambda
r:r["cit_per_art"])[5]])
print(" highest art/auth:",[(r["key"],round(r["art_per_auth"],0)) for r in sorted(cc,key=lambda r:-
r["art_per_auth"])[5]])
print(" highest cit/auth:",[(r["key"],round(r["cit_per_auth"],0)) for r in sorted(cc,key=lambda r:-
r["cit_per_auth"])[5]])
print("\nDONE -> ratios.json")

```

52. read_key.py (SHA-256 6aaeb9d5293dd14f...; 11 lines)

```

# -*- coding: utf-8 -*-
import pandas as pd
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
xl=pd.ExcelFile(f"{SCR}/career_2024.xlsx")
print("SHEETS:",xl.sheet_names)
for sh in xl.sheet_names:
    if sh.lower()!="data":
        k=pd.read_excel(f"{SCR}/career_2024.xlsx",sheet_name=sh,header=None)
        print(f"\n===== {sh} ({k.shape}) =====")
        with pd.option_context("display.max_rows",None,"display.max_colwidth",120,"display.width",200):
            print(k.to_string(index=False,header=False))

```

53. recompute.py (SHA-256 9990fcd3eb65e556...; 84 lines)

```

import openpyxl, math, json, sys
sys.path.insert(0, r'C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford')
from cc_defs import CC

```

```

XLSX = r'C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx'
wb = openpyxl.load_workbook(XLSX, read_only=True, data_only=True)
ws = wb['Data']

```

```

rows = ws.iter_rows(min_row=1, values_only=True)
hdr = list(next(rows))
idx = {h: i for i, h in enumerate(hdr)}
# columns
i_year = idx['year']; i_cntry = idx['cntry']; i_np = idx['np']; i_nc = idx['nc']
i_sf1 = idx['sm-subfield-1']; i_sf2 = idx['sm-subfield-2']

```

```

records = [] # (year, cntry, np, nc, sf1, sf2)
for r in rows:
    sf1 = r[i_sf1]; sf2 = r[i_sf2]
    if sf1 is None:
        continue
    forestry = (sf1 == 'Forestry') or (sf1 == 'Ecology' and sf2 == 'Forestry')
    if not forestry:
        continue
    np_v = r[i_np] or 0
    nc_v = r[i_nc] or 0
    records.append((r[i_year], r[i_cntry], float(np_v), float(nc_v), sf1, sf2))

```

```

print('cohort rows (author-years):', len(records))

```

```

def agg(recs):
    S = len(recs)
    P = sum(x[2] for x in recs)
    C = sum(x[3] for x in recs)
    return S, P, C, (P/S if S else 0), (C/S if S else 0), (C/P if P else 0)

```

```

def show(name, recs):
    S,P,C,apa,cpa,cpar = agg(recs)
    print(f'{name:28s} S={S:6d} P={P:12.0f} C={C:14.0f} art/auth={apa:8.3f} cit/auth={cpa:11.3f}
cit/art={cpar:8.3f}')
    return dict(S=S,P=P,C=C,apa=apa,cpa=cpa,cpar=cpar)

```

```

print('\n=== WHOLE COHORT ===')
res={}
res['cohort']=show('Whole cohort', records)

```

```

print('\n=== GROUPS ===')

```

```

forestry_g = [x for x in records if x[4]=='Forestry']
ecology_g = [x for x in records if x[4]=='Ecology' and x[5]=='Forestry']
res['Forestry']=show('Forestry group', forestry_g)
res['Ecology']=show('Ecology group', ecology_g)
print('check split:', len(forestry_g)+len(ecology_g), '==', len(records))

print('\n=== CONTINENT ===')
from collections import defaultdict
by_cont = defaultdict(list)
unmapped = set()
for x in records:
    cc = x[1]
    info = CC.get(str(cc).lower()) if cc else None
    if info is None:
        unmapped.add(cc)
        cont = 'UNMAPPED'
    else:
        cont = info[1]
    by_cont[cont].append(x)
for cont in sorted(by_cont, key=lambda k:-len(by_cont[k])):
    res.setdefault('cont', {})[cont]=show(cont, by_cont[cont])
print('unmapped country codes:', unmapped)

print('\n=== COUNTRIES (top 25 by scientists) ===')
by_ctry = defaultdict(list)
for x in records:
    info = CC.get(str(x[1]).lower()) if x[1] else None
    name = info[0] if info else f'?{x[1]}'
    by_ctry[name].append(x)
country_res={}
for name in sorted(by_ctry, key=lambda k:-len(by_ctry[k]))[:25]:
    country_res[name]=show(name, by_ctry[name])
res['country']=country_res

# store full country aggregates + per-year for correlations
json.dump({
    'records_by_country': {n:[[x[0],x[2],x[3]] for x in v] for n,v in by_ctry.items()},
}, open(r'C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford\recomp_records.json','w'))
print('\nDONE')

```

54. recompute_verify.py (SHA-256 1a97091a2569a445...; 202 lines)

```

# -*- coding: utf-8 -*-
# Independent recomputation of single-author aggregates from the raw cohort.xlsx.
import sys, json, math
from collections import defaultdict
import pandas as pd

sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford")
from cc_defs import CC

XLSX = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
JDIR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
YEARS = ["2020", "2021", "2022", "2023", "2024"]

df = pd.read_excel(XLSX, sheet_name="Data", engine="openpyxl")
# normalize
df.columns = [str(c).strip() for c in df.columns]
for col in ("nps", "ncs", "np", "nc"):
    df[col] = pd.to_numeric(df[col], errors="coerce").fillna(0)
df["year"] = pd.to_numeric(df["year"], errors="coerce").astype("Int64")
df["sm1"] = df["sm-subfield-1"].astype(str).str.strip()
df["sm2"] = df["sm-subfield-2"].astype(str).str.strip()

print("Total rows in Data:", len(df))
print("Year distribution:", dict(df["year"].value_counts().sort_index()))

# Two groups
forestry = df[df["sm1"] == "Forestry"].copy()
ecology = df[(df["sm1"] == "Ecology") & (df["sm2"] == "Forestry")].copy()
print("Forestry (crit1) rows:", len(forestry))
print("Ecology (crit2) rows:", len(ecology))

```

```

# restrict to nps>0
fsa = forestry[forestry["nps"] > 0].copy()
esa = ecology[ecology["nps"] > 0].copy()
print("Forestry-SA (nps>0) rows:", len(fsa))
print("Ecology-SA (nps>0) rows:", len(esa))

# continent mapping
def continent_of(code):
    c = str(code).strip().lower()
    if c in CC: return CC[c][1]
    return None

findings = []

def approx(a, b):
    return int(a) == int(b)

def verify_group(name, sub, jfile):
    print("\n" + "="*70)
    print("GROUP:", name, " json:", jfile)
    J = json.load(open(f"{JDIR}\\{jfile}", encoding="utf-8"))
    local_findings = []

    # ---- (0) totals sanity: scientists per year == count of nps>0 rows ----
    tot_sci = {y: 0 for y in YEARS}
    tot_nps = {y: 0 for y in YEARS}
    tot_ncs = {y: 0 for y in YEARS}
    for _, r in sub.iterrows():
        y = str(int(r["year"]))
        if y in tot_sci:
            tot_sci[y] += 1
            tot_nps[y] += int(r["nps"])
            tot_ncs[y] += int(r["ncs"])
    print("Recomputed totals scientists:", tot_sci)
    print("JSON totals scientists:", J["totals"]["scientists"])
    print("Recomputed totals publications(nps):", tot_nps)
    print("JSON totals publications:", J["totals"]["publications"])
    print("Recomputed totals citations(ncs):", tot_ncs)
    print("JSON totals citations:", J["totals"]["citations"])
    for y in YEARS:
        if tot_sci[y] != J["totals"]["scientists"].get(y):
            local_findings.append(f"{name}: totals scientists {y} recomputed={tot_sci[y]} json={J['totals']["scientists"].get(y)}")
        if tot_nps[y] != J["totals"]["publications"].get(y):
            local_findings.append(f"{name}: totals publications {y} recomputed={tot_nps[y]} json={J['totals']["publications"].get(y)}")
        if tot_ncs[y] != J["totals"]["citations"].get(y):
            local_findings.append(f"{name}: totals citations {y} recomputed={tot_ncs[y]} json={J['totals']["citations"].get(y)}")

    # nps==0 leak check: full group scientists per year (no nps filter) minus nps>0
    zero_rows = sub[sub["nps"] == 0]
    if len(zero_rows) > 0:
        local_findings.append(f"{name}: sub passed to verify_group contained {len(zero_rows)} rows with nps==0 (LEAK)")

    # ---- (a) CONTINENT per-year scientists / sum(nps) / sum(ncs) ----
    c_sci = defaultdict(lambda: defaultdict(int))
    c_nps = defaultdict(lambda: defaultdict(int))
    c_ncs = defaultdict(lambda: defaultdict(int))
    unmapped = set()
    for _, r in sub.iterrows():
        y = str(int(r["year"]))
        cont = continent_of(r["cntry"])
        if cont is None:
            unmapped.add(str(r["cntry"]))
            continue
        c_sci[cont][y] += 1
        c_nps[cont][y] += int(r["nps"])
        c_ncs[cont][y] += int(r["ncs"])
    if unmapped:
        local_findings.append(f"{name}: UNMAPPED country codes: {sorted(unmapped)}")

    # build json continent lookups
    def jmap(dimlist):
        return {rec["key"]: rec for rec in dimlist}

```

```

j_c_sci = jmap(J["continent"]["scientists"])
j_c_nps = jmap(J["continent"]["publications"])
j_c_ncs = jmap(J["continent"]["citations"])

all_conts = set(c_sci) | set(j_c_sci)
print(f"\n(a) CONTINENTS recomputed={sorted(c_sci)} json={sorted(j_c_sci)}")
for cont in sorted(all_conts):
    for y in YEARS:
        rv = c_sci.get(cont, {}).get(y, 0)
        jv = j_c_sci.get(cont, {}).get(y, 0)
        if rv != jv:
            local_findings.append(f"{name}: continent[{cont}] scientists {y} recomputed={rv} json={jv}")
        rvp = c_nps.get(cont, {}).get(y, 0)
        jvp = j_c_nps.get(cont, {}).get(y, 0)
        if rvp != jvp:
            local_findings.append(f"{name}: continent[{cont}] publications {y} recomputed={rvp} json={jvp}")
        rvc = c_ncs.get(cont, {}).get(y, 0)
        jvc = j_c_ncs.get(cont, {}).get(y, 0)
        if rvc != jvc:
            local_findings.append(f"{name}: continent[{cont}] citations {y} recomputed={rvc} json={jvc}")
# print a compact continent table
for cont in sorted(all_conts):
    rr = [c_sci.get(cont, {}).get(y, 0) for y in YEARS]
    jj = [j_c_sci.get(cont, {}).get(y, 0) for y in YEARS]
    print(f"    {cont:16s} sci recomp={rr} json={jj} {'OK' if rr==jj else 'MISMATCH'}")

# ---- (b) TOP-10 INSTITUTIONS by 2024 scientist count ----
inst_sci = defaultdict(lambda: defaultdict(int))
inst_nps = defaultdict(lambda: defaultdict(int))
inst_ncs = defaultdict(lambda: defaultdict(int))
inst_ctype = {}
for _, r in sub.iterrows():
    y = str(int(r["year"]))
    inst = str(r["inst_name"]).strip()
    inst_sci[inst][y] += 1
    inst_nps[inst][y] += int(r["nps"])
    inst_ncs[inst][y] += int(r["ncs"])
top10 = sorted(inst_sci.keys(), key=lambda k: (-inst_sci[k].get("2024", 0), k))[:10]
print("\n(b) TOP-10 INSTITUTIONS by 2024 scientist count (recomputed):")
for inst in top10:
    s24 = inst_sci[inst].get("2024", 0)
    snps = sum(inst_nps[inst].get(y, 0) for y in YEARS)
    sncs = sum(inst_ncs[inst].get(y, 0) for y in YEARS)
    print(f"    {inst[:45]:45s} 2024sci={s24:4d} sumNPS={snps:6d} sumNCS={sncs:8d}")

# compare to json university dim
j_u_sci = {rec["key"]: rec for rec in J["university"]["scientists"]}
j_u_nps = {rec["key"]: rec for rec in J["university"]["publications"]}
j_u_ncs = {rec["key"]: rec for rec in J["university"]["citations"]}
print("    -- compare vs JSON university dim --")
for inst in top10:
    jr = j_u_sci.get(inst)
    if jr is None:
        local_findings.append(f"{name}: top-inst '{inst}' MISSING from json university.scientists")
        print(f"    MISSING IN JSON: {inst}")
        continue
    r24 = inst_sci[inst].get("2024", 0)
    if r24 != jr.get("2024", 0):
        local_findings.append(f"{name}: inst[{inst}] 2024 scientists recomputed={r24} json={jr.get('2024')}")
    # sum(nps)/sum(ncs): json rec 'total' is sum across years for that metric
    rnps = sum(inst_nps[inst].get(y, 0) for y in YEARS)
    jnps = j_u_nps.get(inst, {}).get("total")
    if jnps is not None and rnps != jnps:
        local_findings.append(f"{name}: inst[{inst}] sum(nps) recomputed={rnps} json_total={jnps}")
    rncs = sum(inst_ncs[inst].get(y, 0) for y in YEARS)
    jncs = j_u_ncs.get(inst, {}).get("total")
    if jncs is not None and rncs != jncs:
        local_findings.append(f"{name}: inst[{inst}] sum(ncs) recomputed={rncs} json_total={jncs}")

# ---- (c) distinct countries and institutions ----
r_countries = set()
for _, r in sub.iterrows():
    cont = continent_of(r["cntry"])
    if cont is not None:
        r_countries.add(CC[str(r["cntry"]).strip().lower()][0])
r_insts = set(str(x).strip() for x in sub["inst_name"])
j_countries = set(rec["key"] for rec in J["country"]["scientists"])
j_insts = set(rec["key"] for rec in J["university"]["scientists"])
print(f"\n(c) distinct COUNTRIES recomputed={len(r_countries)} json={len(j_countries)}")

```

```

print(f"    distinct INSTITUTIONS recomputed={len(r_insts)} json={len(j_insts)}")
if len(r_countries) != len(j_countries):
    local_findings.append(f"{name}: distinct countries recomputed={len(r_countries)} json={len(j_countries)}");
only-in-recomp={sorted(r_countries-j_countries)}; only-in-json={sorted(j_countries-r_countries)}")
    if len(r_insts) != len(j_insts):
        extra_r = sorted(r_insts - j_insts)
        extra_j = sorted(j_insts - r_insts)
        local_findings.append(f"{name}: distinct institutions recomputed={len(r_insts)} json={len(j_insts)}");
only-in-recomp(n={len(extra_r)})={extra_r[:10]}; only-in-json(n={len(extra_j)})={extra_j[:10]}")

print("\nFINDINGS for", name, ":", len(local_findings))
for f in local_findings:
    print("  !", f)
return local_findings

f1 = verify_group("Forestry-SA", fsa, "forestry_B_data.json")
f2 = verify_group("Ecology-SA", esa, "forestry_C_data.json")

print("\n" + "#"*70)
print("TOTAL FINDINGS:", len(f1)+len(f2))
if not (f1 or f2):
    print("ALL RECONCILED EXACTLY.")

```

55. refine_data.py (SHA-256 8328e9239af26d92...; 126 lines)

```

# -*- coding: utf-8 -*-
import json, pandas as pd, numpy as np
from cc_defs import CC
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS = [2020,2021,2022,2023,2024]

df = pd.read_excel(CAREER, sheet_name="Data")
df["country"] = df["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
df["continent"] = df["cntry"].map(lambda c: CC[c][1] if c in CC else "(unknown)")
df["region"] = df["cntry"].map(lambda c: CC[c][2] if c in CC else "(unknown)")

subdiv = {int(k): v for k,v in json.load(open(f"{SCR}/inst_subdiv.json", encoding="utf-8")).items()}
idx = pd.read_csv(f"{SCR}/inst_index.csv"); idx["sd"] = idx["id"].map(subdiv).fillna("Unknown")
look = {(r.country, r.inst): r.sd for r in idx.itertuples()}
CN = {"usa": "United States", "can": "Canada", "aus": "Australia"}
df["subdivision"] = df.apply(lambda r: look.get((CN.get(r["cntry"]), r["inst_name"]), "Unknown") if r["cntry"] in CN else None, axis=1)

def field_of(m):
    m=str(m); f="Forestry" in m; e="Ecology" in m
    return "Both (F & E)" if (f and e) else ("Forestry only" if f else "Ecology only")
df["field"] = df["matched_subfield"].map(field_of)

# ---- quantitative subfields (Science-Metrix labels) per user's guidance ----
QUANT = {
    "Applied Mathematics", "General Mathematics", "Numerical & Computational Mathematics", "Statistics & Probability",
    "Artificial Intelligence & Image Processing", "Computation Theory & Mathematics", "Computer Hardware &
    Architecture",
    "Distributed Computing", "Information Systems", "Medical Informatics", "Networking & Telecommunications", "Software
    Engineering",
    "Aerospace & Aeronautics", "Automobile Design & Engineering", "Biomedical Engineering", "Chemical Engineering",
    "Civil Engineering", "Electrical & Electronic Engineering", "Environmental Engineering", "Geological & Geomatics
    Engineering",
    "Industrial Engineering & Automation", "Mechanical Engineering & Transports", "Mining & Metallurgy", "Operations
    Research",
    "Bioinformatics",
    # user-named terms not present as Science-Metrix labels (kept for transparency; match nothing in the data):
    "Silviculture", "Biometrics", "Mensuration", "Forest Management", "Remote Sensing", "GIS",
}
def is_quant(row):
    return (str(row["sm-subfield-1"]).strip() in QUANT) or (str(row["sm-subfield-2"]).strip() in QUANT)
df["orient"] = np.where(df.apply(is_quant, axis=1), "Quantitative-oriented", "Qualitative-oriented")

def agg(frame, keycol, metric):
    val, fn = ("authfull", "count") if metric=="scientists" else (("np", "sum") if metric=="publications" else
    ("nc", "sum"))
    t = frame.pivot_table(index=keycol, columns="year", values=val, aggfunc=fn, fill_value=0)
    for y in YEARS:
        if y not in t.columns: t[y]=0

```

```

t = t[YEARS]; t["total"]=t.sum(axis=1); t["avg"]=t[YEARS].mean(axis=1).round(1)
return t

def recs_by_metric(frame, keycol, extra_country=False):
    """Return {metric: [records sorted by that metric's total desc]}."""
    inst_country = None
    if extra_country:
        inst_country = frame.groupby(keycol)["country"].agg(lambda s: s.value_counts().index[0])
    out = {}
    for m in ["scientists","publications","citations"]:
        a = agg(frame, keycol, m).sort_values("total", ascending=False)
        lst = []
        for k in a.index:
            r = dict(key=k, **{str(y): int(a.loc[k,y]) for y in YEARS},
                    total=int(a.loc[k,"total"]), avg=float(a.loc[k,"avg"]))
            if extra_country: r["country"] = inst_country.get(k, "")
            lst.append(r)
        out[m] = lst
    return out

def band(records_list, n_individual=20, cap=None, target_frac=0.10):
    recs = records_list[:cap] if cap else records_list
    head = recs[:n_individual]; tail = recs[n_individual:]
    R = sum(r["total"] for r in tail); target = target_frac*R
    bands, cur, acc, start = [], [], 0, n_individual+1
    def close(cur, a, b):
        g = {"rank_from":a, "rank_to":b, "n":len(cur),
            "total":sum(r["total"] for r in cur), "avg":round(sum(r["avg"] for r in cur),1)}
        for y in YEARS: g[str(y)] = sum(r[str(y)] for r in cur)
        return g
    for r in tail:
        cur.append(r); acc += r["total"]
        if R>0 and acc >= target:
            bands.append(close(cur, start, start+len(cur)-1)); start += len(cur); cur=[]; acc=0
    if cur: bands.append(close(cur, start, start+len(cur)-1))
    return {"head": head, "bands": bands, "remainder_total": R, "n_total": len(records_list)}

OUT = {"years": YEARS}
OUT["continent"] = recs_by_metric(df[df.continent!="(unknown)"], "continent")
OUT["region"] = recs_by_metric(df[df.region!="(unknown)"], "region")
OUT["country"] = recs_by_metric(df[df.country!="(unknown)"], "country")
OUT["field"] = recs_by_metric(df, "field")
OUT["orient"] = recs_by_metric(df, "orient")
OUT["university"] = recs_by_metric(df[df["inst_name"].notna()], "inst_name", extra_country=True)
for code,name in [("usa","us_state"), ("can","ca_prov"), ("aus","au_state")]:
    OUT[name] = recs_by_metric(df[df.cntry==code], "subdivision")

# banded (top-20 individual + ~10% cumulative bands) for country (all) and university (top 100)
OUT["country_banded"] = {m: band(OUT["country"][m], n_individual=20, cap=None) for m in OUT["country"]}
OUT["university_banded"] = {m: band(OUT["university"][m], n_individual=20, cap=100) for m in OUT["university"]}

# ---- maps: average over years ----
OUT["map_world_avg"]={}; OUT["map_us_avg"]={}
d = df[df.country!="(unknown)"]
for m in ["scientists","publications","citations"]:
    a = agg(d, "cntry",m); OUT["map_world_avg"][m] = {str(k).upper(): float(a.loc[k,"avg"]) for k in a.index}
    au = agg(df[df.cntry=="usa"], "subdivision",m); OUT["map_us_avg"][m] = {str(k): float(au.loc[k,"avg"]) for k in au.index}

# ---- taxonomy tree (present subfields only) ----
th = pd.read_excel(f"{SCR}/thresholds_career_2024.xlsx", sheet_name="SubField")
th.columns=[c.strip() for c in th.columns]
sf2fd = {r["Subfield"].strip(): (str(r["Domain"]).strip(),str(r["Field"]).strip()) for _,r in th.iterrows() if
isinstance(r["Subfield"],str)}
sf1 = df["sm-subfield-1"].astype(str).str.strip(); sf2 = df["sm-subfield-2"].astype(str).str.strip()
present = [p for p in sorted(set(sf1)|set(sf2)) if p and p!="nan"]
tree={}
for sf in present:
    mask=(sf1==sf)|(sf2==sf); dom,fld=sf2fd.get(sf, ("unmapped"), ("unmapped"))
    tree[sf]={ "domain":dom, "field":fld, "total":int(mask.sum()), "avg":round(int(mask.sum())/5,1),
        "quant":sf in QUANT, "target":sf in ("Forestry","Ecology")}
OUT["tree"]=tree
qhits={sf:int(((sf1==sf)|(sf2==sf)).sum()) for sf in QUANT if sf in present}
OUT["quant_subfield_hits"]=dict(sorted(qhits.items(),key=lambda kv:-kv[1]))
OUT["totals"]={m:{str(y):(int((df.year==y).sum()) if m=="scientists" else int(df[df.year==y]["np" if
m=="publications" else "nc"].sum())) for y in YEARS} for m in ["scientists","publications","citations"]}

json.dump(OUT, open(f"{SCR}/refine_data.json","w",encoding="utf-8"), default=float)

```

```

print("universities:", len(OUT["university"]["scientists"]), "| countries:", len(OUT["country"]["scientists"]))
print("tree subfields:", len(tree), "| quant present:", len(qhits))
print("orient (all-yr scientists):", {r["key"]:r["total"] for r in OUT["orient"]["scientists"]})
for m in ["scientists","publications","citations"]:
    b=OUT["university_banded"][m]; print(f"uni band[{m}]: head={len(b['head'])} bands={len(b['bands'])}
    sizes={g['n'] for g in b['bands']}")
b=OUT["country_banded"]["scientists"]; print("country band[sci]: bands=", [ (g['rank_from'],g['rank_to'],g['n'])
for g in b['bands']])
print("saved refine_data.json")

```

56. refine_explore.py (SHA-256 a82f4126bff13a7b...; 33 lines)

```

import pandas as pd
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df = pd.read_excel(CAREER, sheet_name="Data")
print("cols:", [c for c in df.columns])
print("\nnp (publications) null:", df["np"].isna().sum(), "min/median/max:", df["np"].min(), df["np"].median(),
df["np"].max())
print("\nc (citations) null:", df["nc"].isna().sum(), "min/median/max:", df["nc"].min(), df["nc"].median(),
df["nc"].max())
print("2024 totals: scientists", int((df.year==2024).sum()),
      "| publications", int(df[df.year==2024]["np"].sum()),
      "| citations", int(df[df.year==2024]["nc"].sum()))

# thresholds subfield -> (domain, field)
th = pd.read_excel(f"{SCR}/thresholds_career_2024.xlsx", sheet_name="SubField")
th.columns = [c.strip() for c in th.columns]
print("\nthresholds cols:", list(th.columns), "rows:", len(th))
sf2fd = {r["Subfield"].strip(): (r["Domain"], r["Field"]) for _,r in th.iterrows() if
instance(r["Subfield"],str)}

# distinct subfields present in our data (sf1 or sf2)
present = set()
for col in ["sm-subfield-1","sm-subfield-2"]:
    present |= set(x.strip() for x in df[col].dropna().unique())
present = {p for p in present if p}
print("\ndistinct subfields present (sf1 or sf2):", len(present))
missing = [p for p in present if p not in sf2fd]
print("present-but-not-in-thresholds:", missing)
# build tree counts
domains = {}
for sf in present:
    dom, fld = sf2fd.get(sf, ("(unmapped)","(unmapped)"))
    domains.setdefault(dom, []).setdefault(fld, []).append(sf)
print("\ndomains:", len(domains), "| fields:", sum(len(v) for v in domains.values()), "| subfields:",
len(present))
for dom in domains:
    print(f" {dom}: {sum(len(s) for s in domains[dom].values())} subfields across {len(domains[dom])} fields")

```

57. refine_figures.py (SHA-256 81b0e21ff973366a...; 212 lines)

```

# -*- coding: utf-8 -*-
import json, numpy as np
import matplotlib; matplotlib.use("Agg")
import matplotlib.pyplot as plt
from matplotlib.collections import PolyCollection
from matplotlib.colors import BoundaryNorm
import matplotlib.cm as cm
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
plt.rcParams.update({"font.family":"Arial","font.size":9,"figure.dpi":200})
D=json.load(open(f"{SCR}/refine_data.json",encoding="utf-8"))
YEARS=[2020,2021,2022,2023,2024]
GREEN="#2F5233"; ECO="#4C9F70"; BOTH="#B7CE63"; BLUE="#2C6E9B"; GREY="#9AA0A6"; GOLD="#C6902F"
METLAB={"scientists":"Scientists","publications":"Publications","citations":"Citations"}
PAL=["#2F5233","#4C9F70","#2C6E9B","#C6902F","#A63D40","#7B6CA6","#3E8E7E","#B7654C","#8DA750","#5B7DB1"]

def yvals(rec): return [rec[str(y)] for y in YEARS]

def lines_panel(ax, records, n, title, ylabel):
    for i,rec in enumerate(records[:n]):
        ax.plot(YEARS, yvals(rec), marker="o", ms=3, lw=1.6, color=PAL[i%len(PAL)], label=str(rec["key"])[2:22])
    ax.set_title(title, fontsize=10, color=GREEN, fontweight="bold")
    ax.set_xticks(YEARS); ax.set_ylabel(ylabel); ax.spines[["top","right"]].set_visible(False)

```

```

ax.legend(fontsize=6.5, frameon=False, ncol=2, loc="upper left")

# ---- F1: totals per year (3 metrics, indexed) ----
fig,axs=plt.subplots(1,3,figsize=(7.2,2.5))
for ax,m in zip(axs,["scientists","publications","citations"]):
    vals=[D["totals"][m][str(y)] for y in YEARS]
    ax.plot(YEARS,vals,marker="o",color=GREEN,lw=2)
    for x,v in zip(YEARS,vals): ax.annotate(f"{v/1000:.0f}k" if v>=10000 else f"{v:,}",(x,v),textcoords="offset
points",xytext=(0,4),ha="center",fontsize=6.5)
    ax.set_title(METLAB[m],fontsize=10,color=GREEN,fontweight="bold"); ax.set_xticks(YEARS)
    ax.spines[["top","right"]].set_visible(False); ax.set_ylim(0,max(vals)*1.18)
fig.tight_layout(); fig.savefig(f"{SCR}/rf1_totals.png",dpi=200); plt.close(fig)

# ---- F2: field composition per year (3 metrics stacked) ----
fig,axs=plt.subplots(1,3,figsize=(7.2,2.7))
order=["Ecology","Both","Forestry"]; colors=[ECO,BOTH,GREEN]
for ax,m in zip(axs,["scientists","publications","citations"]):
    recs={r["key"]:r for r in D["field"][m]}
    series=[yvals(recs[k]) for k in order]
    ax.stackplot(YEARS,*series,labels=order,colors=colors,alpha=.92)
    ax.set_title(METLAB[m],fontsize=10,color=GREEN,fontweight="bold"); ax.set_xticks(YEARS)
    ax.spines[["top","right"]].set_visible(False)
axs[0].legend(fontsize=6.5,frameon=False,loc="upper left")
fig.tight_layout(); fig.savefig(f"{SCR}/rf2_field.png",dpi=200); plt.close(fig)

# ---- F3: continents over years ----
fig,axs=plt.subplots(1,3,figsize=(7.4,3.0))
for ax,m in zip(axs,["scientists","publications","citations"]):
    lines_panel(ax, D["continent"][m], 6, METLAB[m], "count")
fig.suptitle("Representation by continent, per release year",fontsize=10.5,color=GREEN,fontweight="bold",y=1.02)
fig.tight_layout(); fig.savefig(f"{SCR}/rf3_continent.png",dpi=200,bbox_inches="tight"); plt.close(fig)

# ---- F4: top countries over years ----
fig,axs=plt.subplots(1,3,figsize=(7.4,3.0))
for ax,m in zip(axs,["scientists","publications","citations"]):
    lines_panel(ax, D["country"][m], 8, METLAB[m], "count")
fig.suptitle("Top-8 countries by metric, per release year",fontsize=10.5,color=GREEN,fontweight="bold",y=1.02)
fig.tight_layout(); fig.savefig(f"{SCR}/rf4_country.png",dpi=200,bbox_inches="tight"); plt.close(fig)

# ---- F5: countries banded (scientists, avg/yr) ----
def banded_bar(fname, banded, title, unit="avg/yr scientists"):
    head=banded["head"]; bands=banded["bands"]
    labels=[str(h["key"])[1:26] for h in head]+[f"#{{g['rank_from']}}-{{g['rank_to']}} ({{g['n']}})" for g in bands]
    vals=[h["avg"] for h in head]+[g["avg"] for g in bands]
    cols=[GREEN]*len(head)+[BLUE]*len(bands)
    fig,ax=plt.subplots(figsize=(7.0,6.2))
    y=np.arange(len(labels))[:-1]
    ax.barh(y,vals,color=cols)
    ax.set_yticks(y); ax.set_yticklabels(labels,fontsize=7)
    for yi,v in zip(y,vals): ax.annotate(f"{{v:,.0f}}", (v,yi),va="center",fontsize=6.5)
    ax.set_xlabel(unit); ax.set_title(title,fontsize=10.5,color=GREEN,fontweight="bold")
    ax.spines[["top","right"]].set_visible(False); ax.set_xlim(0,max(vals)*1.12)
    import matplotlib.patches as mp
    ax.legend(handles=[mp.Patch(color=GREEN,label="Top-20 (individual)"),mp.Patch(color=BLUE,label="Banded groups
(~10% each)"]],
    fontsize=7,frameon=False,loc="lower right")
    fig.tight_layout(); fig.savefig(fname,dpi=200); plt.close(fig)
banded_bar(f"{SCR}/rf5_country_band.png", D["country_banded"]["scientists"],
    "Countries: top-20 individual, remainder banded (by scientists)")

# ---- F6: top universities over years ----
fig,axs=plt.subplots(1,3,figsize=(7.4,3.2))
for ax,m in zip(axs,["scientists","publications","citations"]):
    lines_panel(ax, D["university"][m], 7, METLAB[m], "count")
fig.suptitle("Top-7 institutions by metric, per release year",fontsize=10.5,color=GREEN,fontweight="bold",y=1.02)
fig.tight_layout(); fig.savefig(f"{SCR}/rf6_university.png",dpi=200,bbox_inches="tight"); plt.close(fig)

# ---- F7: universities banded (top 100) ----
banded_bar(f"{SCR}/rf7_university_band.png", D["university_banded"]["scientists"],
    "Universities: top-20 individual, ranks 21-100 banded (by scientists)")

# ---- F8: orientation quant/qual over years (3 metrics grouped bars) ----
fig,axs=plt.subplots(1,3,figsize=(7.2,2.6))
for ax,m in zip(axs,["scientists","publications","citations"]):
    recs={r["key"]:r for r in D["orient"][m]}
    qual=yvals(recs["Qualitative-oriented"]); quant=yvals(recs["Quantitative-oriented"])
    ax.bar(YEARS,qual,color=ECO,label="Qualitative-oriented")

```

```

    ax.bar(YEARS, quant, bottom=qual, color=BLUE, label="Quantitative-oriented")
    ax.set_title(METLAB[m], fontsize=10, color=GREEN, fontweight="bold"); ax.set_xticks(YEARS)
    ax.spines[["top", "right"]].set_visible(False)
    axs[0].legend(fontsize=6.5, frameon=False, loc="upper left")
    fig.tight_layout(); fig.savefig(f"{SCR}/rf8_orient.png", dpi=200); plt.close(fig)

# ===== MAPS (average over years) =====
def geoms(f):
    g=f["geometry"]
    if not g: return []
    return [g["coordinates"]] if g["type"]=="Polygon" else (g["coordinates"] if g["type"]=="MultiPolygon" else
[])
def nice_bins(vals):
    v=sorted(x for x in vals if x>0)
    if not v: return [0,1]
    qs=np.quantile(v, [0,.2,.4,.6,.8,.95,1.0])
    b=sorted(set(int(round(x)) for x in qs))
    if len(b)<3: b=[0,max(1,int(v[-1]))]
    if b[0]>0: b=[0]+b
    return b
def draw_panel(ax, feats, keyfn, values, title, xlim=None, ylim=None):
    bins=nice_bins(list(values.values()));
    if len(bins)<3: bins=[0,1,2]
    cmap=plt.get_cmap("YlGn", len(bins)-1); norm=BoundaryNorm(bins, cmap.N)
    bg, fg, fc=[], [], []
    for f in feats:
        k=keyfn(f); v=values.get(k, 0)
        for poly in geoms(f):
            ext=np.asarray(poly[0])
            if v and v>0: fg.append(ext); fc.append(cmap(norm(v)))
            else: bg.append(ext)
    ax.add_collection(PolyCollection(bg, facecolors="#EFEFEA", edgecolors="#BBB", linewidths=.2))
    if fg: ax.add_collection(PolyCollection(fg, facecolors=fc, edgecolors="#888", linewidths=.2))
    ax.autoscale()
    if xlim: ax.set_xlim(*xlim)
    if ylim: ax.set_ylim(*ylim)
    ax.axis("off"); ax.set_title(title, fontsize=9.5, color=GREEN, fontweight="bold")
    sm=cm.ScalarMappable(norm=norm, cmap=cmap); sm.set_array([])
    cb=plt.colorbar(sm, ax=ax, orientation="horizontal", fraction=0.05, pad=0.02, aspect=30)
    cb.ax.tick_params(labelsize=6); cb.set_ticks(bins)
    cb.set_ticklabels([f"{int(b)}" for b in bins])

world=json.load(open(f"{SCR}/world_110m.geojson", encoding="utf-8"))
wf=[f for f in world["features"] if f["properties"].get("CONTINENT")!="Antarctica"]
def iso3(f):
    p=f["properties"]; a=p.get("ISO_A3", "-99"); return p.get("ADM0_A3") if a in ("-99", None, "") else a
fig, axs=plt.subplots(3, 1, figsize=(6.6, 9.2))
for ax, m in zip(axs, ["scientists", "publications", "citations"]):
    draw_panel(ax, wf, iso3, D["map_world_avg"][m], f"World - average {METLAB[m].lower()} per year (2020-2024)", ylim=(-60, 85))
fig.tight_layout(); fig.savefig(f"{SCR}/rf9_worldmaps.png", dpi=200, bbox_inches="tight"); plt.close(fig)

us=json.load(open(f"{SCR}/us_states.geojson", encoding="utf-8"))
uf=[f for f in us["features"] if f["properties"]["name"] not in ("Alaska", "Hawaii", "Puerto Rico")]
fig, axs=plt.subplots(3, 1, figsize=(6.6, 9.0))
for ax, m in zip(axs, ["scientists", "publications", "citations"]):
    draw_panel(ax, uf, lambda f:f["properties"]["name"], D["map_us_avg"][m], f"United States - average {METLAB[m].lower()} per year", xlim=(-125, -66), ylim=(24, 50))
fig.tight_layout(); fig.savefig(f"{SCR}/rf10_usmaps.png", dpi=200, bbox_inches="tight"); plt.close(fig)

# ===== TREE-CHART (Domain -> Field -> Subfield, present-in-data) =====
tree=D["tree"]
# organize
byd={}
for sf, info in tree.items():
    byd.setdefault(info["domain"], []).setdefault(info["field"], []).append((sf, info))
DOM_ORDER=["Applied Sciences", "Natural Sciences", "Health Sciences", "Economic & Social Sciences", "Arts & Humanities", "(unmapped)"]
doms=[d for d in DOM_ORDER if d in byd]+[d for d in byd if d not in DOM_ORDER]
# count rows to size figure
total_leaves=sum(len(sfs) for d in byd for sfs in byd[d].values())
total_fields=sum(len(byd[d]) for d in byd)
lines=len(doms)+total_fields+total_leaves
# 2-column layout
col_lines=(lines+1)//2
fig_h=max(9, col_lines*0.135)
fig, axs=plt.subplots(1, 2, figsize=(13.2, fig_h))
# flatten into drawing ops
ops=[] # (level, text, info)

```

```

for d in doms:
    dtot=sum(i["total"] for f in byd[d] for _,i in byd[d][f])
    ops.append((0,f"{d}",{"tot":dtot}))
    for f in sorted(byd[d], key=lambda ff:-sum(i["total"] for _,i in byd[d][ff])):
        ftot=sum(i["total"] for _,i in byd[d][f])
        ops.append((1,f"{f}",{"tot":ftot}))
        for sf,info in sorted(byd[d][f], key=lambda t:-t[1]["total"]):
            ops.append((2,sf,info))
half=(len(ops)+1)//2
# ensure we don't split a domain awkwardly: find nearest domain boundary near half
cut=half
for i in range(half,len(ops)):
    if ops[i][0]==0: cut=i; break
chunks=[ops[:cut],ops[cut:]]
X={0:0.01,1:0.05,2:0.11}
for ax,chunk in zip(axes,chunks):
    ax.set_xlim(0,1); ax.set_ylim(0,max(len(chunk),1)+4); ax.axis("off"); ax.invert_yaxis()
    for i,(lvl,text,info) in enumerate(chunk):
        y=i+0.5
        if lvl==0:
            ax.text(X[0],y,f"{text}",fontsize=8.5,fontweight="bold",color="#222",va="center")
            ax.text(0.985,y,f"{info['tot']:,}",fontsize=7,color="#222",va="center",ha="right",fontweight="bold")
        elif lvl==1:
            ax.text(X[1],y,f"{text}",fontsize=7.5,color=GREEN,va="center",fontweight="bold")
            ax.text(0.985,y,f"{info['tot']:,}",fontsize=6.5,color=GREEN,va="center",ha="right")
        else:
            col = GOLD if info["target"] else (BLUE if info["quant"] else "#444")
            mk = "●" if info["target"] else ("»" if info["quant"] else "• ")
            ax.text(X[2],y,f"{mk}{text}",fontsize=6.6,color=col,va="center")
            ax.text(0.985,y,f"{info['total']:,}",fontsize=6.2,color="#666",va="center",ha="right")
import matplotlib.patches as mp
axes[0].legend(handles=[mp.Patch(color=GOLD,label="Target sub-sub-field (Forestry/Ecology)"),
                        mp.Patch(color=BLUE,label="Quantitative sub-sub-field (»)"),
                        mp.Patch(color="#444",label="Other (qualitative)")],
                fontsize=7.5,frameon=False,loc="lower left")
fig.suptitle("Taxonomy of Forestry+Ecology authors: Field (Domain) → Sub-Field → Sub-Sub-Field - right numbers
= author-records 2020-2024",
            fontsize=9.5,color=GREEN,fontweight="bold",y=0.995)
fig.tight_layout(rect=[0,0,1,0.985]); fig.savefig(f"{SCR}/rf11_tree.png",dpi=200,bbox_inches="tight");
plt.close(fig)

print("figures written:", total_leaves, "leaves,", total_fields, "fields,", len(doms), "domains; tree fig
h=",round(fig_h,1))

```

58. refine_figures2.bak.py (SHA-256 ce7eabe1cdd3b989...; 441 lines)

```

# -*- coding: utf-8 -*-
# Big-font figures. Forestry set (prefix ff_) for main report; combined set (prefix cf_) for appendix.
import json, numpy as np
import matplotlib; matplotlib.use("Agg")
import matplotlib.pyplot as plt
import matplotlib.patches as mp
from matplotlib.collections import PolyCollection, LineCollection
from matplotlib.colors import BoundaryNorm
import matplotlib.cm as cm
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
# ---- global BIG-font style (about the same size as 10-11pt body text on the page) ----
plt.rcParams.update({
    "font.family":"Arial","figure.dpi":200,
    "font.size":13,"axes.titlesize":15,"axes.labelsize":13,
    "xtick.labelsize":12,"ytick.labelsize":12,"legend.fontsize":12})
YEARS=[2020,2021,2022,2023,2024]
GREEN="#2F5233"; ECO="#4C9F70"; BOTH="#B7CE63"; BLUE="#2C6E9B"; GREY="#9AA0A6"; GOLD="#C6902F"
METLAB={"scientists":"Scientists","publications":"Publications","citations":"Citations"}
PAL=["#2F5233","#4C9F70","#2C6E9B","#C6902F","#A63D40","#7B6CA6","#3E8E7E","#B7654C","#8DA750","#5B7DB1"]
FORE=json.load(open(f"{SCR}/forestry_data.json",encoding="utf-8"))
COMB=json.load(open(f"{SCR}/refine_data.json",encoding="utf-8"))
MERGED=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))

def yv(rec): return [rec[str(y)] for y in YEARS]
def scaled(metric,v): # human units for annotations
    if metric=="scientists": return f"{int(round(v)):,}"
    if metric=="publications": return f"{v/1000:,.0f}k"
    return f"{v/1e6:,.1f}M"
def axis_units(metric): return {"scientists":"scientists","publications":"publications","citations":"citations"}
[metric]

```

```

# ----- Figure 1: single-panel % change, actual numbers on points, legend OUTSIDE -----
def fig_change(D,fname,title):
    fig,ax=plt.subplots(figsize=(7.4,4.6))
    maxpct=max([D["totals"][m][str(YEARS[-1])]/D["totals"][m]["2020"]-1)*100 for m in
["scientists","publications","citations"]])
    for i,m in enumerate(["scientists","publications","citations"]):
        base=D["totals"][m]["2020"]; vals=[D["totals"][m][str(y)] for y in YEARS]
        pct=[(v/base-1)*100 for v in vals]
        ax.plot(YEARS,pct,marker="o",ms=7,lw=2.6,color=PAL[i],label=METLAB[m],zorder=3)
        # distinct vertical offsets per metric so the labels never stack where lines meet at 2020
        dy={"scientists":-18,"publications":14,"citations":30}[m]
        va="top" if dy<0 else "bottom"
        for j,(x,p,v) in enumerate(zip(YEARS,pct,vals)):
            ax.annotate(scaled(m,v),(x,p),textcoords="offset points",xytext=(0,dy),
                ha="center",va=va,fontSize=11,color=PAL[i],fontWeight="bold")
        ax.axhline(0,color="#999",lw=1,ls="--",zorder=1)
        ax.set_xticks(YEARS); ax.set_ylabel("Change since 2020 (%)")
        ax.set_title(title,color=GREEN,fontWeight="bold",pad=10)
        ax.spines[["top","right"]].set_visible(False)
        ax.set_ylim(-14, maxpct*1.30)
        ax.legend(loc="upper center",bbox_to_anchor=(0.5,-0.13),ncol=3,frameon=False)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- Figure-1-style % change with TWO series (e.g. quant vs qual orientation) -----
def fig_change_2series(D,fname,title,series):
    recs={r["key"]:r for r in D["orient"]["scientists"]}
    fig,ax=plt.subplots(figsize=(7.4,4.6))
    allp=[]
    for k,_ in series:
        base=recs[k]["2020"]; allp+=[(recs[k][str(y)]/base-1)*100 for y in YEARS]
    for (k,col),dy in zip(series,[16,-20]):
        base=recs[k]["2020"]; vals=[recs[k][str(y)] for y in YEARS]; pct=[(v/base-1)*100 for v in vals]
        ax.plot(YEARS,pct,marker="o",ms=7,lw=2.6,color=col,label=k)
        va="bottom" if dy>0 else "top"
        for x,pp,v in zip(YEARS,pct,vals):
            ax.annotate(f"{v:,}",(x,pp),textcoords="offset
points",xytext=(0,dy),ha="center",va=va,fontSize=11,color=col,fontWeight="bold")
        ax.axhline(0,color="#999",lw=1,ls="--")
        ax.set_xticks(YEARS); ax.set_ylabel("Change since 2020 (%)")
        ax.set_title(title,color=GREEN,fontWeight="bold",pad=10); ax.spines[["top","right"]].set_visible(False)
        ax.set_ylim(min(allp+[0])-9, max(allp)+11)
        ax.legend(loc="upper center",bbox_to_anchor=(0.5,-0.13),ncol=2,frameon=False)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- per-year, 3 metrics stacked VERTICALLY; legend OUTSIDE to the right -----
def fig_lines_vert(D,key,n,fname,suptitle):
    fig,axs=plt.subplots(3,1,figsize=(9.2,9.4))
    for ax,m in zip(axs,["scientists","publications","citations"]):
        for i,rec in enumerate(D[key][m][:n]):
            ax.plot(YEARS,yv(rec),marker="o",ms=5,lw=2.2,color=PAL[i%len(PAL)],label=str(rec["key"])[26])
        ax.set_title(METLAB[m],fontSize=14,color=GREEN,fontWeight="bold",loc="left")
        ax.set_xticks(YEARS); ax.set_ylabel(axis_units(m)); ax.spines[["top","right"]].set_visible(False)
        ax.legend(fontsize=10.5,frameon=False,ncol=1,loc="center left",bbox_to_anchor=(1.005,0.5))
    fig.suptitle(suptitle,fontSize=15,color=GREEN,fontWeight="bold",y=0.997)
    fig.tight_layout(rect=[0,0,0.80,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- orientation / field stacked bars, 3 metrics vertical -----
def fig_stack_vert(D,key,order,colors,fname,suptitle):
    fig,axs=plt.subplots(3,1,figsize=(7.4,8.6))
    for ax,m in zip(axs,["scientists","publications","citations"]):
        recs={r["key"]:r for r in D[key][m]}
        bottom=np.zeros(len(YEARS))
        for k,c in zip(order,colors):
            if k not in recs: continue
            vals=np.array(yv(recs[k]),float)
            ax.bar(YEARS,vals,bottom=bottom,color=c,label=k); bottom+=vals
        ax.set_title(METLAB[m],fontSize=14,color=GREEN,fontWeight="bold",loc="left")
        ax.set_xticks(YEARS); ax.set_ylabel(axis_units(m)); ax.spines[["top","right"]].set_visible(False)
    axs[0].legend(fontsize=11,frameon=False,loc="upper left",ncol=len(order))
    fig.suptitle(suptitle,fontSize=15,color=GREEN,fontWeight="bold",y=0.997)
    fig.tight_layout(rect=[0,0,1,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- ONE compact graph: categories x (5 year-columns per metric), Y = % of each metric's max -----
def fig_onebar(D,key,fname,title,figw=11.0,n=None,catsize=12,percat=False):
    metrics=["scientists","publications","citations"]
    MCMAP={"scientists":"Greens","publications":"Blues","citations":"YlOrBr"}
    shades={m:[plt.get_cmap(MCMAP[m])(0.33+0.15*i) for i in range(5)] for m in metrics} # 2020->2024 light->dark
    conts=[r["key"] for r in D[key]["scientists"]]
    if n: conts=conts[:n]

```

```

val={m:{r["key"]:r for r in D[key][m]} for m in metrics}
# percat: normalize each bar to ITS OWN category-metric 5-yr max (every category peaks at 100%)
# else: normalize each metric to the global max across categories (only the largest category hits 100%)
if percat:
    denom=lambda m,c: max(max(val[m][c][str(y)] for y in YEARS),1e-9)
else:
    gmax={m:max(val[m][c][str(y)] for c in conts for y in YEARS) for m in metrics}
    denom=lambda m,c: max(gmax[m],1e-9)
def afmt(m,v):
    if m=="scientists": return f"{int(round(v)):,}"
    if m=="publications": return f"{v/1000:.1f}k" if v<10000 else f"{v/1000:.0f}k"
    return f"{v/1e6:.2f}M" if v<1e6 else f"{v/1e6:.1f}M"
fig,ax=plt.subplots(figsize=(figw,4.9))
bw=1.0; gmetric=0.7; gcont=2.2
x=0.0; cont_ticks=[]; mtick_x=[]; mtick_lab=[]
for c in conts:
    c0=x
    for m in metrics:
        m0=x
        for i,y in enumerate(YEARS):
            v=val[m][c][str(y)]; pct=100*v/denom(m,c)
            ax.bar(x,pct,width=bw,color=shades[m][i],edgecolor="white",linewidth=0.3)
            if y==2024:
                ax.annotate(afmt(m,v),(x,pct),textcoords="offset points",xytext=(0,3),
                    ha="center",va="bottom",rotation=90,fontsize=9.5,color="#333")
                x+=bw
            mtick_x.append((m0+x-bw)/2); mtick_lab.append(m[0].upper()) # S / P / C under each block
            x+=gmetric
        cont_ticks.append((c0+x-gmetric-bw/2)/2 + (c0)/2 if False else (c0+(x-gmetric))/2)
        x+=gcont
ax.set_xticks(mtick_x); ax.set_xticklabels(mtick_lab,fontsize=9.5,color="#666")
# continent labels on a secondary line
def wrap2(s,width=17):
    s=str(s)
    if len(s)<=width: return s
    mid=len(s)//2; l=s.rfind(" ",0,mid); r=s.find(" ",mid)
    pos=l if (l>0 and (r<0 or mid-l<=r-mid)) else r
    return (s[:pos]+\n"+s[pos+1:]) if pos>0 else (s[:width-1]+"...")
for xt,c in zip(cont_ticks,conts):
    ax.annotate(wrap2(c),(xt,0),xycoords=("data","axes fraction"),textcoords="offset points",
        xytext=(0,-
22),ha="center",va="top",fontsize=catsize,fontweight="bold",color=GREEN,annotation_clip=False)
ax.set_ylabel("% of each category's own 2020-2024 maximum" if percat else "% of each metric's 2020-2024
maximum")
ax.set_ylim(0,118)
ax.set_title(title,color=GREEN,fontweight="bold",pad=12)
ax.spines[["top","right"]].set_visible(False)
handles=[mp.Patch(color=plt.get_cmap(MCMAP[m])(0.85),label=METLAB[m]) for m in metrics]
ax.legend(handles=handles,loc="upper right",frameon=False,fontsize=11.5,ncol=1,
    handlelength=1.1,borderaxespad=0.2)
fig.tight_layout(rect=[0,0.10,1,1]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- banded horizontal bar (avg/yr) -----
def fig_band(banded,fname,title,unit,figh=None,fs=11):
    head=banded["head"]; bands=banded["bands"]
    labels=[str(h["key"])[0:30] for h in head]+[f"#{g['rank_from']}-{g['rank_to']}" for g in bands]
    vals=[h["avg"] for h in head]+[g["avg"] for g in bands]
    cols=[GREEN]*len(head)+[BLUE]*len(bands)
    fig,ax=plt.subplots(figsize=(7.4, figh or max(6.5,0.32*len(labels)+1)))
    y=np.arange(len(labels))[::-1]; ax.barh(y,vals,color=cols,height=0.82)
    ax.set_yticks(y); ax.set_yticklabels(labels,fontsize=fs); ax.tick_params(axis="y",pad=1)
    for yi,v in zip(y,vals): ax.annotate(f" {v:,.0f}",(v,yi),va="center",fontsize=fs-1)
    ax.set_xlabel(unit); ax.set_title(title,fontsize=14,color=GREEN,fontweight="bold")
    ax.spines[["top","right"]].set_visible(False); ax.set_xlim(0,max(vals)*1.14)
    ax.legend(handles=[mp.Patch(color=GREEN,label="Top-20 (individual)"),mp.Patch(color=BLUE,label="Banded groups
(~10% each)"),
        frameon=False,loc="upper center",bbox_to_anchor=(0.5,-0.05),ncol=2,fontsize=11)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ===== MAPS =====
def geoms(f):
    g=f["geometry"]
    if not g: return []
    return [g["coordinates"]] if g["type"]=="Polygon" else (g["coordinates"] if g["type"]=="MultiPolygon" else
[])
def nice_bins(vals):
    v=sorted(x for x in vals if x>0)
    if not v: return [0,1]
    qs=np.quantile(v,[0,.2,.4,.6,.8,.95,1.0]); b=sorted(set(int(round(x)) for x in qs))
    if len(b)<3: b=[0,max(1,int(v[-1]))]

```

```

    if b[0]>0: b=[0]+b
    return b
def draw_panel(ax, feats, keyfn, values, title, xlim=None, ylim=None):
    bins=nice_bins(list(values.values()))
    if len(bins)<3: bins=[0,1,2]
    cmap=plt.get_cmap("YlGn", len(bins)-1); norm=BoundaryNorm(bins, cmap.N)
    bg, fg, fc=[], [], []
    for f in feats:
        k=keyfn(f); v=values.get(k,0)
        for poly in geoms(f):
            ext=np.asarray(poly[0])
            if v and v>0: fg.append(ext); fc.append(cmap(norm(v)))
            else: bg.append(ext)
    ax.add_collection(PolyCollection(bg, facecolors="#EFEFEA", edgecolors="#BBB", linewidths=.2))
    if fg: ax.add_collection(PolyCollection(fg, facecolors=fc, edgecolors="#888", linewidths=.2))
    ax.autoscale()
    if xlim: ax.set_xlim(*xlim)
    if ylim: ax.set_ylim(*ylim)
    ax.axis("off"); ax.set_title(title, fontsize=13.5, color=GREEN, fontweight="bold")
    sm=cm.ScalarMappable(norm=norm, cmap=cmap); sm.set_array([])
    cb=plt.colorbar(sm, ax=ax, orientation="horizontal", fraction=0.05, pad=0.02, aspect=32)
    cb.ax.tick_params(labelsize=10.5); cb.set_ticks(bins); cb.set_ticklabels([f"{int(b)}" for b in bins])
WORLD=json.load(open(f"{SCR}/world_110m.geojson", encoding="utf-8"))
WF=[f for f in WORLD["features"] if f["properties"].get("CONTINENT")!="Antarctica"]
def iso3(f):
    p=f["properties"]; a=p.get("ISO_A3", "-99"); return p.get("ADM0_A3") if a in ("-99", None, "") else a
US=json.load(open(f"{SCR}/us_states.geojson", encoding="utf-8"))
UF=[f for f in US["features"] if f["properties"]["name"] not in ("Alaska", "Hawaii", "Puerto Rico")]
def fig_worldmaps(D, fname, label):
    fig, axs=plt.subplots(3,1, figsize=(7.0,9.6))
    for ax, m in zip(axs, ["scientists", "publications", "citations"]):
        draw_panel(ax, WF, iso3, D["map_world_avg"][m], f"World - avg {METLAB[m].lower()} /yr ({label}, 2020-2024)", ylim=(-60,85))
    fig.tight_layout(); fig.savefig(fname, dpi=200, bbox_inches="tight"); plt.close(fig)
def fig_usmaps(D, fname, label):
    fig, axs=plt.subplots(1,3, figsize=(11.0,3.3))
    for ax, m in zip(axs, ["scientists", "publications", "citations"]):
        draw_panel(ax, UF, lambda f: f["properties"]["name"], D["map_us_avg"][m], f"Avg {METLAB[m].lower()} /yr", xlim=(-125,-66), ylim=(24,50))
    fig.suptitle(f"United States - five-year averages ({label})", fontsize=13, color=GREEN, fontweight="bold", y=1.03)
    fig.tight_layout(); fig.savefig(fname, dpi=200, bbox_inches="tight"); plt.close(fig)

# ----- INTENSITY ratio maps (differentiate what count-maps cannot): output & impact per unit -----
def fig_ratio_worldmaps(D, fname, label, min_sci_total=5):
    sci=D["map_world_avg"]["scientists"]; pub=D["map_world_avg"]["publications"]; cit=D["map_world_avg"]["citations"]
    def rat(num, den):
        return {k:num[k]/den[k] for k in num if den.get(k,0)>0 and sci.get(k,0)*5>=min_sci_total}
    panels=[("Publications per scientist (career output intensity)", rat(pub, sci)),
            ("Citations per scientist (career impact per person)", rat(cit, sci)),
            ("Citations per publication (impact per paper)", rat(cit, pub))]
    fig, axs=plt.subplots(3,1, figsize=(7.0,9.6))
    for ax, (t, vals) in zip(axs, panels):
        draw_panel(ax, WF, iso3, vals, f"{t} - {label}, 2020-2024", ylim=(-60,85))
    fig.tight_layout(); fig.savefig(fname, dpi=200, bbox_inches="tight"); plt.close(fig)

# ===== REAL TREE GRAPH (branched) =====
DOM_ORDER=["Applied Sciences", "Natural Sciences", "Health Sciences", "Economic & Social Sciences", "Arts & Humanities", "(unmapped)"]
def draw_tree(D, fname, title, leaf_threshold=10):
    tree=D["tree"]
    byd={}
    omitted=0; omit_hits=0
    for sf, info in tree.items():
        if info["total"]<leaf_threshold and not info["target"]:
            omitted+=1; omit_hits+=info["total"]; continue
        byd.setdefault(info["domain"], []).setdefault(info["field"], []).append((sf, info))
    doms=[d for d in DOM_ORDER if d in byd]+[d for d in byd if d not in DOM_ORDER]
    leaf_rows=[]; dom_span={}; fld_span={}
    yy=0.0
    for d in doms:
        d0=yy
        for f in sorted(byd[d], key=lambda ff:-sum(i["total"] for _, i in byd[d][ff])):
            f0=yy
            for sf, info in sorted(byd[d][f], key=lambda t:-t[1]["total"]):
                leaf_rows.append((d, f, sf, info, yy)); yy+=1
            fld_span[(d, f)]=(f0, yy-1)
            yy+=0.5
        dom_span[d]=(d0, yy-1.5)

```

```

        yy+=1.0
nrows=yy
fig_h=max(7.5,0.315*len(leaf_rows)+1.6)
fig,ax=plt.subplots(figsize=(9.0,fig_h))          # wide canvas -> on-page fonts ~body size
ax.set_xlim(0,1); ax.set_ylim(-1.5,nrows+0.5); ax.invert_yaxis(); ax.axis("off")
# column x-anchors and bus (elbow) x-positions
XD,XF,XL,XCNT=0.005,0.185,0.560,0.988
BUSD,BUSF=0.150,0.545
def ymid(a,b): return (a+b)/2
segs=[]
for d in doms:                                     # domain -> fields
    dy=ymid(*dom_span[d])
    ax.text(XD,dy,d,fontsize=13,fontweight="bold",color="#141414",va="center")
    for f in byd[d]:
        fy=ymid(*fld_span[(d,f)])
        segs+=[[ (BUSD,fy), (BUSD,fy) ], [ (BUSD,fy), (XF-0.004,fy) ]]
    segs+=[[ (XD+0.002,dy), (BUSD,dy) ]]
for (d,f),(a,b) in fld_span.items():               # field -> leaves
    fy=ymid(a,b)
    ax.text(XF,fy,f,fontsize=12,color=GREEN,fontweight="bold",va="center")
    for (_,sf,info,ly) in [r for r in leaf_rows if r[0]==d and r[1]==f]:
        segs+=[[ (BUSF,fy), (BUSF,ly) ], [ (BUSF,ly), (XL-0.004,ly) ]]
    segs+=[[ (XF+0.004,fy), (BUSF,fy) ]]
for (d,f,sf,info,ly) in leaf_rows:                # leaves + counts
    col=GOLD if info["target"] else (BLUE if info["quant"] else "#3a3a3a")
    mk="●" if info["target"] else ("»" if info["quant"] else "•")
    name=sf if len(sf)<=34 else sf[:33]+"..."
    ax.text(XL,ly,f"{mk} {name}",fontsize=11.5,color=col,va="center")
    ax.text(XCNT,ly,f"{info['total']:,}",fontsize=11,color="#555",va="center",ha="right")
ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.1,zorder=0))
note=(f"{'omitted' if len(segs)>len(leaf_rows)-len(segs) else ''} records total) "
      "omitted for legibility; full list in the companion workbook." if omitted else "")
if note: fig.text(0.5,0.006,note,fontsize=10.5,color="#777",ha="center",style="italic")
fig.legend(handles=[mp.Patch(color=GOLD,label="Target sub-sub-field (Forestry)"),
                    mp.Patch(color=BLUE,label="Quantitative sub-sub-field (»)"),
                    mp.Patch(color="#3a3a3a",label="Other / qualitative (•)"),],
           frameon=False,fontsize=11.5,ncol=3,loc="lower center",bbox_to_anchor=(0.5,0.028))
fig.suptitle(title,fontsize=14,color=GREEN,fontweight="bold",y=0.998)
fig.tight_layout(rect=[0,0.055,1,0.982]); fig.savefig(fname,dpi=200); plt.close(fig)
return len(leaf_rows),omitted

```

```

# ===== REAL TREE: primary-forestry (sm-subfield-1==Forestry) =====
def draw_forestry_tree(tax,fname,title,big=10,quant_min=5):
    branches=tax["branches"]; QSET=set(tax["quant_set"])
    def keep(sf,n): return (n>=big) or (sf in QSET and n>=quant_min)
    kept={}; omitted=0; omit_hits=0
    for f in branches:
        leaves={sf:n for sf,n in branches[f].items() if keep(sf,n)}
        drop={sf:n for sf,n in branches[f].items() if not keep(sf,n)}
        omitted+=len(drop); omit_hits+=sum(drop.values())
        if leaves: kept[f]=leaves
    fields=sorted(kept,key=lambda f:-sum(kept[f].values()))
    leaf_rows=[]; fld_span={}; yy=0.0
    for f in fields:
        f0=yy
        for sf in sorted(kept[f],key=lambda x:-kept[f][x]):
            leaf_rows.append((f,sf,kept[f][sf],yy)); yy+=1
            fld_span[f]=(f0,yy-1); yy+=0.6
    nrows=yy; rootY=(0+nrows-1.6)/2
    fig_h=max(7.0,0.33*len(leaf_rows)+1.6)
    fig,ax=plt.subplots(figsize=(9.2,fig_h))
    ax.set_xlim(0,1); ax.set_ylim(-1.4,nrows+0.3); ax.invert_yaxis(); ax.axis("off")
    XR,XF,XL,XCNT=0.01,0.26,0.60,0.988; BUSR,BUSF=0.235,0.575
    segs=[]
    ax.text(XR,rootY,"Forestry",fontsize=14.5,fontweight="bold",color=GOLD,va="center")
    ax.text(XR,rootY+1.05,f"primary specialty; n={tax['n_primary']:,}",fontsize=9.5,color="#777",va="center")
    for f in fields:
        fy=(fld_span[f][0]+fld_span[f][1])/2
        segs+=[[ (BUSR,rootY), (BUSR,fy) ], [ (BUSR,fy), (XF-0.004,fy) ]]
    segs+=[[ (XR+0.11,rootY), (BUSR,rootY) ]]
    for f in fields:
        a,b=fld_span[f]; fy=(a+b)/2
        ax.text(XF,fy,f,fontsize=11.5,color=GREEN,fontweight="bold",va="center")
        for (ff,sf,n,ly) in [r for r in leaf_rows if r[0]==f]:
            segs+=[[ (BUSF,fy), (BUSF,ly) ], [ (BUSF,ly), (XL-0.004,ly) ]]
        segs+=[[ (XF+0.004,fy), (BUSF,fy) ]]
    for (f,sf,n,ly) in leaf_rows:
        q=sf in QSET; col=BLUE if q else "#3a3a3a"; mk="»" if q else "•"
        name=sf if len(sf)<=34 else sf[:33]+"..."
        ax.text(XL,ly,f"{mk} {name}",fontsize=11,color=col,va="center")

```

```

        ax.text(XCNT,ly,f"{n:,.}",fontsize=10.5,color="#555",va="center",ha="right")
    ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.1,zorder=0))
    note=(f"Minor secondary specialties (<{big} authors, and non-quantitative) omitted for legibility: "
        f"{omitted} sub-sub-fields, {omit_hits:,} authors. Full list in the workbook.") if omitted else ""
    if note: fig.text(0.5,0.006,note,fontsize=10,color="#777",ha="center",style="italic")
    fig.legend(handles=[mp.Patch(color=GOLD,label="Forestry (primary sub-field, sm-subfield-1)"),
        mp.Patch(color=GREEN,label="Field of secondary specialty (sm-field of sm-subfield-2)"),
        mp.Patch(color=BLUE,label="Quantitative secondary sub-sub-field (>)")],
        frameon=False,fontsize=10.5,ncol=1,loc="lower left",bbox to anchor=(0.01,0.01))
    fig.suptitle(title,fontsize=13.5,color=GREEN,fontweight="bold",y=0.998)
    fig.tight_layout(rect=[0,0.02,1,0.982]); fig.savefig(fname,dpi=200); plt.close(fig)
    return len(leaf_rows),omitted

# ===== MULTI-ROOT TREE: AQ(sm-field) -> AM(sm-subfield-1) -> AO(sm-subfield-2) =====
TEAL="#2C6E9B"
def draw_multiroot_tree(tax,fname,title,root_min=15,leaf_min=12,quant_min=5):
    roots=tax["roots"]; QSET=set(tax["quant_set"])
    root_tot={aq:sum(a.values()) for a in roots[aq].values()}
    keep=[aq for aq in sorted(roots,key=lambda x:-root_tot[x]) if root_tot[aq]>=root_min]
    drop_roots=[aq for aq in roots if aq not in keep]; drop_n=sum(root_tot[aq] for aq in drop_roots)
    leafrows=[]; am_span=[]; aq_span=[]; yy=0.0
    for aq in keep:
        aq0=yy
        for am in sorted(roots[aq],key=lambda x:-sum(roots[aq][x].values())):
            am0=yy; aos=roots[aq][am]; CAP=15
            kept=sorted([(ao,n) for ao,n in aos.items() if n>=leaf_min or (ao in QSET and
n>=quant_min)],key=lambda t:-t[1])
            dropped=[(ao,n) for ao,n in aos.items() if not (n>=leaf_min or (ao in QSET and n>=quant_min))]
            dropped+=kept[CAP:]; kept=kept[:CAP] # cap leaves per branch
            for ao,n in kept: leafrows.append((aq,am,ao,n,yy)); yy+=1
            if dropped: leafrows.append((aq,am,f"({len(dropped)} minor)",sum(n for _,n in dropped),yy)); yy+=1
            am_span[(aq,am)]=(am0,yy-1); yy+=0.5
        aq_span[aq]=(aq0,yy-1.5); yy+=1.0
    nrows=yy
    fig_h=max(7.5,0.28*len(leafrows)+1.9)
    fig,ax=plt.subplots(figsize=(9.4,fig_h))
    ax.set_xlim(0,1); ax.set_ylim(-1.5,nrows+0.3); ax.invert_yaxis(); ax.axis("off")
    XQ,XM,XL,XC=0.005,0.30,0.60,0.99; BUSQ,BUSM=0.27,0.575
    def ymid(sp): return (sp[0]+sp[1])/2
    segs=[]
    for aq in keep: # AQ roots -> AM
        qy=ymid(aq_span[aq])
        name=aq if len(aq)<=30 else aq[:29]+"..."
        ax.text(XQ,qy,name,fontsize=11.5,fontweight="bold",color="#141414",va="center")
        for am in roots[aq]:
            if (aq,am) not in am_span: continue
            my=ymid(am_span[(aq,am)]); segs+=[[ (BUSQ,qy), (BUSQ,my)], [(BUSQ,my), (XM-0.004,my)]]
            segs+=[[ (XQ+0.005,qy), (BUSQ,qy)]]
    for (aq,am),sp in am_span.items(): # AM -> AO leaves
        my=ymid(sp); amcol=GOLD if am=="Forestry" else TEAL
        ax.text(XM,my,f"{am}",fontsize=11,fontweight="bold",color=amcol,va="center")
        for (a,m,ao,n,ly) in [r for r in leafrows if r[0]==aq and r[1]==am]:
            segs+=[[ (BUSM,my), (BUSM,ly)], [(BUSM,ly), (XL-0.004,ly)]]
            segs+=[[ (XM+0.004,my), (BUSM,my)]]
    for (aq,am,ao,n,ly) in leafrows: # AO leaves + counts
        if ao.startswith("("): col="#9AA0A6"; mk=""
        elif ao=="Forestry": col=GOLD; mk="● "
        elif ao in QSET: col=BLUE; mk="» "
        else: col="#3a3a3a"; mk="• "
        nm=ao if len(ao)<=32 else ao[:31]+"..."
        ax.text(XL,ly,f"{mk}{nm}",fontsize=10.5,color=col,va="center")
        ax.text(XC,ly,f"{n:,.}",fontsize=10,color="#555",va="center",ha="right")
    ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.05,zorder=0))
    note=(f"Roots (sm-field) with <{root_min} authors omitted: {len(drop_roots)} fields, {drop_n:,} authors. "
        f"Minor secondary specialties aggregated per branch. Full lists in the workbook.")
    fig.text(0.5,0.005,note,fontsize=9.5,color="#777",ha="center",style="italic")
    fig.legend(handles=[mp.Patch(color=GOLD,label="AM = Forestry (Forestry group)"),
        mp.Patch(color=TEAL,label="AM = Ecology (Ecology group -> secondary Forestry)"),
        mp.Patch(color=BLUE,label="Quantitative secondary sub-sub-field (>)")],
        frameon=False,fontsize=10.5,ncol=1,loc="lower left",bbox to anchor=(0.01,0.015))
    fig.suptitle(title,fontsize=12,color=GREEN,fontweight="bold",y=0.997)
    fig.tight_layout(rect=[0,0.04,1,0.978]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)
    return len(leafrows),len(keep)

# ===== BUILD FORESTRY (main = union: Forestry group + Ecology group) =====
fig_change(FORE,f"{SCR}/ff1_change.png","Forestry research cohort - growth of each metric since 2020 (actual
values labelled)")
draw_multiroot_tree(MERGED["taxonomy"],f"{SCR}/ff2_tree.png",
    "Forestry research taxonomy: field (AQ) -> primary subfield (AM) -> secondary subfield (AO)")
fig_onebar(FORE,"group",f"{SCR}/ff4_group.png",

```

```

"Forestry research cohort: scientists, publications and citations by group (2020-2024)",figw=6.4)
fig_onebar(FORE,"continent",f"{SCR}/ff3_continent.png",
"Forestry: scientists, publications and citations by continent (2020-2024)")
fig_onebar(FORE,"country",f"{SCR}/ff5_country.png",
"Forestry: scientists, publications and citations by country (top 8, 2020-2024)",figw=13.6,n=8,catsize=11)
fig_band(FORE["country_banded"]["scientists"],f"{SCR}/ff6_country_band.png","Forestry countries: top-20
individual, remainder ~10% bands","avg scientists / yr",figh=5.1,fs=8.5)
fig_onebar(FORE,"university",f"{SCR}/ff7_university.png",
"Forestry: scientists, publications and citations by institution (top 7, % of each institution's own
maximum)",
figw=13.2,n=7,catsize=9.5,percat=True)
fig_band(FORE["university_banded"]["scientists"],f"{SCR}/ff8_university_band.png","Forestry institutions: top-20
individual, ranks 21-100 ~10% bands","avg scientists / yr")
fig_ratio_worldmaps(FORE,f"{SCR}/ff9_worldmaps.png","forestry cohort")
fig_usmaps(FORE,f"{SCR}/ff10_usmaps.png","Forestry")

# ===== BUILD COMBINED (appendix) =====
fig_change(COMB,f"{SCR}/cf1_change.png","Combined Forestry+Ecology - growth of each metric since 2020 (actual
values labelled)")
fig_stack_vert(COMB,"field",["Ecology","Both","Forestry"],[ECO,BOTH,GREEN],f"{SCR}/cf2_field.png","Combined:
field composition (Ecology / both / Forestry), per year")
fig_lines_vert(COMB,"continent",6,f"{SCR}/cf3_continent.png","Combined: representation by continent, per release
year")
fig_lines_vert(COMB,"country",8,f"{SCR}/cf4_country.png","Combined: top-8 countries by metric, per release year")
fig_band(COMB["country_banded"]["scientists"],f"{SCR}/cf5_country_band.png","Combined countries: top-20
individual, remainder ~10% bands","avg scientists / yr")
fig_band(COMB["university_banded"]["scientists"],f"{SCR}/cf6_university_band.png","Combined institutions: top-20
individual, ranks 21-100 ~10% bands","avg scientists / yr")
fig_worldmaps(COMB,f"{SCR}/cf7_worldmaps.png","Combined")
fig_usmaps(COMB,f"{SCR}/cf8_usmaps.png","Combined")
draw_tree(COMB,f"{SCR}/cf9_tree.png","Combined Forestry+Ecology taxonomy present in the data",leaf_threshold=15)

print("figures written OK (primary-forestry; no tree/orient figures)")

```

59. refine_figures2.py (SHA-256 852f5789f67f853d...; 732 lines)

```

# -*- coding: utf-8 -*-
# Big-font figures. Forestry set (prefix ff_) for main report; combined set (prefix cf_) for appendix.
import json, numpy as np
import matplotlib; matplotlib.use("Agg")
import matplotlib.pyplot as plt
import matplotlib.patches as mp
from matplotlib.collections import PolyCollection, LineCollection
from matplotlib.colors import BoundaryNorm
import matplotlib.cm as cm
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
# ---- global BIG-font style (about the same size as 10-11pt body text on the page) ----
plt.rcParams.update({
    "font.family":"Arial","figure.dpi":200,
    "font.size":13,"axes.titlesize":15,"axes.labelsize":13,
    "xtick.labelsize":12,"ytick.labelsize":12,"legend.fontsize":12})
YEARS=[2020,2021,2022,2023,2024]
GREEN="#2F5233"; ECO="#4C9F70"; BOTH="#B7CE63"; BLUE="#2C6E9B"; GREY="#9AA0A6"; GOLD="#C6902F"
METLAB={"scientists":"Scientists","publications":"Publications","citations":"Citations"}
PAL=["#2F5233","#4C9F70","#2C6E9B","#C6902F","#A63D40","#7B6CA6","#3E8E7E","#B7654C","#8DA750","#5B7DB1"]
FORE=json.load(open(f"{SCR}/forestry_data.json",encoding="utf-8"))
MERGED=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8"))
RATIOS=json.load(open(f"{SCR}/ratios.json",encoding="utf-8"))

def yv(rec): return [rec[str(y)] for y in YEARS]
def scaled(metric,v): # human units for annotations
    if metric=="scientists": return f"{int(round(v)):,}"
    if metric=="publications": return f"{v/1000:,.0f}k"
    return f"{v/1e6:,.1f}M"
def axis_units(metric): return {"scientists":"scientists","publications":"publications","citations":"citations"}
[metric]

# ----- Figure 1: single-panel % change, actual numbers on points, legend OUTSIDE -----
def fig_change(D,fname,title,vaifmt=None):
    vf=vaifmt or scaled
    fig,ax=plt.subplots(figsize=(7.4,4.6))
    metrics=["scientists","publications","citations"]
    maxpct=max((D["totals"][m][str(YEARS[-1])]/D["totals"][m]["2020"]-1)*100 for m in metrics)
    # stagger the end-of-line series labels when two lines finish close together (else they overprint)
    finals=[m:(D["totals"][m][str(YEARS[-1])]/D["totals"][m]["2020"]-1)*100 for m in metrics]
    yrange=(maxpct+max((0.30*maxpct,8)))+14

```

```

enddy={m:0 for m in metrics}
order=sorted(metrics,key=lambda m:finals[m])
for lo,hi in zip(order,order[1:]):
    if finals[hi]-finals[lo] < 0.06*yrange:
        enddy[lo]-=7; enddy[hi]+=7
for i,m in enumerate(metrics):
    base=D["totals"][m]["2020"]; vals=[D["totals"][m][str(y)] for y in YEARS]
    pct=[(v/base-1)*100 for v in vals]
    ax.plot(YEARS,pct,marker="o",ms=7,lw=2.6,color=PAL[i],zorder=3)
    # distinct vertical offsets per metric so the labels never stack where lines meet at 2020
    dy={"scientists":-18,"publications":14,"citations":30}[m]
    va="top" if dy<0 else "bottom"
    for j,(x,p,v) in enumerate(zip(YEARS,pct,vals)):
        ax.annotate(vf(m,v),(x,p),textcoords="offset points",xytext=(0,dy),
            ha="center",va=va,fontSize=11,color=PAL[i],fontWeight="bold",zorder=5,
            bbox=dict(boxstyle="round,pad=0.12",fc="white",ec="none",alpha=0.8))
    # series label at the end of its line (replaces legend)
    ax.annotate(METLAB[m],(YEARS[-1],pct[-1]),textcoords="offset points",xytext=(14,enddy[m]),
        ha="left",va="center",fontSize=13,color=PAL[i],fontWeight="bold",annotation_clip=False,
        bbox=dict(boxstyle="round,pad=0.10",fc="white",ec="none",alpha=0.75))
ax.axhline(0,color="#999",lw=1,ls="--",zorder=1)
ax.set_xticks(YEARS); ax.set_ylabel("Change since 2020 (%)")
ax.set_title(title,color=GREEN,fontWeight="bold",pad=12)
ax.spines[["top","right"]].set_visible(False)
# headroom above the top point so value labels never punch into the title (matters for low-growth panels)
ax.set_ylim(-14, maxpct + max(0.30*maxpct, 8)); ax.set_xlim(YEARS[0]-0.15,YEARS[-1]+0.85)
fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

```

----- Figure-1-style % change with TWO series (e.g. quant vs qual orientation) -----

```

def fig_change_2series(D,fname,title,series):
    recs={r["key"]:r for r in D["orient"]["scientists"]}
    fig,ax=plt.subplots(figsize=(7.4,4.6))
    allp=[]
    for k,_ in series:
        base=recs[k]["2020"]; allp+=[(recs[k][str(y)]/base-1)*100 for y in YEARS]
    for (k,col),dy in zip(series,[16,-20]):
        base=recs[k]["2020"]; vals=[recs[k][str(y)] for y in YEARS]; pct=[(v/base-1)*100 for v in vals]
        ax.plot(YEARS,pct,marker="o",ms=7,lw=2.6,color=col,label=k)
        va="bottom" if dy>0 else "top"
        for x,pp,v in zip(YEARS,pct,vals):
            ax.annotate(f"{v:,.1f}",(x,pp),textcoords="offset
points",xytext=(0,dy),ha="center",va=va,fontSize=11,color=col,fontWeight="bold")
        ax.axhline(0,color="#999",lw=1,ls="--")
        ax.set_xticks(YEARS); ax.set_ylabel("Change since 2020 (%)")
        ax.set_title(title,color=GREEN,fontWeight="bold",pad=10); ax.spines[["top","right"]].set_visible(False)
        ax.set_ylim(min(allp+[0])-9, max(allp)+11)
        ax.legend(loc="upper center",bbox_to_anchor=(0.5,-0.13),ncol=2,frameon=False)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

```

----- per-year, 3 metrics stacked VERTICALLY; legend OUTSIDE to the right -----

```

def fig_lines_vert(D,key,n,fname,suptitle):
    fig,axs=plt.subplots(3,1,figsize=(9.2,9.4))
    for ax,m in zip(axs,["scientists","publications","citations"]):
        for i,rec in enumerate(D[key][m][:n]):
            ax.plot(YEARS,yv(rec),marker="o",ms=5,lw=2.2,color=PAL[i%len(PAL)],label=str(rec["key"])[26])
        ax.set_title(METLAB[m],fontSize=14,color=GREEN,fontWeight="bold",loc="left")
        ax.set_xticks(YEARS); ax.set_ylabel(axis_units(m)); ax.spines[["top","right"]].set_visible(False)
        ax.legend(fontsize=10.5,frameon=False,ncol=1,loc="center left",bbox_to_anchor=(1.005,0.5))
    fig.suptitle(suptitle,fontSize=15,color=GREEN,fontWeight="bold",y=0.997)
    fig.tight_layout(rect=[0,0,0.80,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

```

----- orientation / field stacked bars, 3 metrics vertical -----

```

def fig_stack_vert(D,key,order,colors,fname,suptitle):
    fig,axs=plt.subplots(3,1,figsize=(7.4,8.6))
    for ax,m in zip(axs,["scientists","publications","citations"]):
        recs={r["key"]:r for r in D[key][m]}
        bottom=np.zeros(len(YEARS))
        for k,c in zip(order,colors):
            if k not in recs: continue
            vals=np.array(yv(recs[k]),float)
            ax.bar(YEARS,vals,bottom=bottom,color=c,label=k); bottom+=vals
        ax.set_title(METLAB[m],fontSize=14,color=GREEN,fontWeight="bold",loc="left")
        ax.set_xticks(YEARS); ax.set_ylabel(axis_units(m)); ax.spines[["top","right"]].set_visible(False)
        axs[0].legend(fontsize=11,frameon=False,loc="upper left",ncol=len(order))
    fig.suptitle(suptitle,fontSize=15,color=GREEN,fontWeight="bold",y=0.997)
    fig.tight_layout(rect=[0,0,1,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

```

----- ONE compact graph: categories × (5 year-columns per metric), Y = % of each metric's max -----

```

def
fig_onebar(D,key,fname,title,figw=11.0,n=None,catsize=12,percat=False,share=False,disp=6.7,figh=5.8,labfrac=0.28,
afmt=None):
    # fonts sized so that after the figure is scaled from `figw` down to display width `disp`,
    # in-figure text lands at ~body size on the page. S(pt) = on-page pt -> in-figure pt.
    K=figw/disp
    def S(pt): return pt*K
    metrics=["scientists","publications","citations"]
    MCMAP={"scientists":"Greens","publications":"Blues","citations":"YlOrBr"}
    shades={m:[plt.get_cmap(MCMAP[m])(0.33+0.15*i) for i in range(5)] for m in metrics} # 2020->2024 light->dark
    conts=[r["key"] for r in D[key]["scientists"]]
    if n: conts=conts[:n]
    val={m:{r["key"]:r for r in D[key][m]} for m in metrics}
    # share: each bar as % of the combined total of ALL categories that metric-year (common denominator; shares
    sum to 100%)
    # percat: normalize each bar to ITS OWN category-metric 5-yr max (every category peaks at 100%)
    # else: normalize each metric to the global max across categories (only the largest category hits 100%)
    if share: # denominator = total of ALL categories (not just the top-n shown), so a top-n view sums to <100%
        tot={m:{y:sum(val[m][cc][str(y)] for cc in val[m]) for y in YEARS} for m in metrics}
        denom=lambda m,c,y: max(tot[m][y],1e-9)
    elif percat:
        denom=lambda m,c,y: max(max(val[m][c][str(yy)] for yy in YEARS),1e-9)
    else:
        gmax={m:max(val[m][c][str(y)] for c in conts for y in YEARS) for m in metrics}
        denom=lambda m,c,y: max(gmax[m],1e-9)
    def _afmt_default(m,v):
        if m=="scientists": return f"{int(round(v))}:"
        if m=="publications": return f"{v/1000:.1f}k" if v<10000 else f"{v/1000:.0f}k"
        return f"{v/1e6:.2f}M" if v<1e6 else f"{v/1e6:.1f}M"
    afmt=afmt or _afmt_default
    fig,ax=plt.subplots(figsize=(figw,figh))
    bw=1.0; gmetric=0.7; gcont=2.2
    x=0.0; cont_ticks=[]; mtick_x=[]; mtick_lab=[]
    for c in conts:
        c0=x
        for m in metrics:
            m0=x
            for i,y in enumerate(YEARS):
                v=val[m][c][str(y)]; pct=100*v/denom(m,c,y)
                ax.bar(x,pct,width=bw,color=shades[m][i],edgecolor="white",linewidth=0.3)
                if y==2024:
                    ax.annotate(afmt(m,v),(x,pct),textcoords="offset points",xytext=(0,3),
                               ha="center",va="bottom",rotation=90,fontsize=S(9.5),color="#333")
                x+=bw
            mtick_x.append((m0+x-bw)/2); mtick_lab.append(m[0].upper()) # S / P / C under each block
            x+=gmetric
        cont_ticks.append((c0+x-gmetric-bw/2)/2 + (c0)/2 if False else (c0+(x-gmetric))/2)
        x+=gcont
    ax.set_xticks(mtick_x); ax.set_xticklabels(mtick_lab,fontsize=S(9.5),color="#666")
    ABBR={"Northern America":"N. America","Northern Europe":"N. Europe","Western Europe":"W. Europe",
          "Southern Europe":"S. Europe","Eastern Europe":"E. Europe","Eastern Asia":"E. Asia",
          "Southern Asia":"S. Asia","South-eastern Asia":"SE Asia","Western Asia":"W. Asia",
          "South America":"S. America","Central America":"C. America","Northern Africa":"N. Africa",
          "Eastern Africa":"E. Africa","Southern Africa":"S. Africa","Australia & New Zealand":"Australia & NZ",
          "United States":"United States","United Kingdom":"United Kingdom"}
    def shorten(s):
        s=ABBR.get(str(s),str(s))
        for a,b in [("University","Univ."),("Institute","Inst."),("Natural Resources","Nat. Res."),
                    ("Technology","Tech."),("Agricultural Sciences","Agric. Sci."),("Sciences","Sci.")]:
            s=s.replace(a,b)
        return s
    def wrap2(s,width=11):
        s=shorten(s)
        if len(s)<=width: return s
        mid=len(s)//2; l=s.rfind(" ",0,mid); r=s.find(" ",mid)
        pos=l if (l>0 and (r<0 or mid-l<=r-mid)) else r
        if pos<=0: return s
        return s[:pos]+"\\n"+s[pos+1:]
    for xt,c in zip(cont_ticks,conts):
        ax.annotate(wrap2(c),(xt,0),xycoords=("data","axes fraction"),textcoords="offset points",
                    xytext=(0,-
16),ha="center",va="top",fontsize=S(10),fontweight="bold",color=GREEN,annotation_clip=False)
    ax.set_ylabel("% share of total" if share else
                  ("% of category's own max" if percat else "% of metric's max"),fontsize=S(10))
    ax.set_ylim(0,118 if not share else max(60,max(100*val[m][c][str(y)]/denom(m,c,y) for m in metrics for c in
    conts for y in YEARS)*1.30))
    ax.set_title(title,color=GREEN,fontweight="bold",pad=(30 if percat else 8),fontsize=S(11.5))
    ax.spines[["top","right"]].set_visible(False)
    ax.tick_params(axis="y",labelsize=S(9.5))
    handles=[mp.Patch(color=plt.get_cmap(MCMAP[m])(0.85),label=METLAB[m]) for m in metrics]
    if percat: # all bars reach 100% -> put legend just above the plot (no empty corner)

```

```

        ax.legend(handles=handles,loc="lower center",bbox_to_anchor=(0.5,1.005),ncol=3,frameon=False,
                  fontsize=8(10),columnspacing=1.8,handlelength=1.2)
    else:
        # share/global: rightmost categories are short -> legend sits in the empty upper-right
        ax.legend(handles=handles,loc="upper right",frameon=False,fontsize=8(10),handlelength=1.2)
    fig.tight_layout(rect=[0,labfrac,1,1]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- horizontal bar ranking (avg/yr), straight top-N -----
def fig_band(banded,fname,title,unit,figh=None,fs=11):
    head=banded["head"]
    labels=[str(h["key"])[1:30] for h in head]
    vals=[h["avg"] for h in head]
    fig,ax=plt.subplots(figsize=(7.4, figh or max(6.5,0.32*len(labels)+1)))
    y=np.arange(len(labels))[:1]; ax.barh(y,vals,color=GREEN,height=0.82)
    ax.set_yticks(y); ax.set_yticklabels(labels,fontsize=fs); ax.tick_params(axis="y",pad=1)
    for yi,v in zip(y,vals): ax.annotate(f" {v:,.1f}" if 0<v<1 else f" {v:,.0f}",
    (v,yi),va="center",fontsize=fs-1)
    ax.set_xlabel(unit,fontsize=fs+1,labelpad=3); ax.set_title(title,fontsize=fs+5,color=GREEN,fontweight="bold")
    ax.spines[["top","right"]].set_visible(False); ax.set_xlim(0,max(vals)*1.16)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ===== MAPS =====
def geoms(f):
    g=f["geometry"]
    if not g: return []
    return [g["coordinates"]] if g["type"]=="Polygon" else (g["coordinates"] if g["type"]=="MultiPolygon" else
    [])
def nice_bins(vals):
    v=sorted(x for x in vals if x>0)
    if not v: return [0,1]
    qs=np.quantile(v,[0,.2,.4,.6,.8,.95,1.0]); b=sorted(set(int(round(x)) for x in qs))
    if len(b)<3: b=[0,max(1,int(v[-1]))]
    if b[0]>0: b=[0]+b
    return b
def draw_panel(ax,feats,keyfn,values,title,xlim=None,ylim=None,fscale=1.3):
    bins=nice_bins(list(values.values()))
    if len(bins)<3: bins=[0,1,2]
    cmap=plt.get_cmap("YlGn",len(bins)-1); norm=BoundaryNorm(bins,cmap.N)
    bg,fg,fc=[],[],[]
    for f in feats:
        k=keyfn(f); v=values.get(k,0)
        for poly in geoms(f):
            ext=np.asarray(poly[0])
            if v and v>0: fg.append(ext); fc.append(cmap(norm(v)))
            else: bg.append(ext)
    ax.add_collection(PolyCollection(bg,facecolors="#E0E0E0",edgecolors="#BBB",linewidths=.2))
    if fg: ax.add_collection(PolyCollection(fg,facecolors=fc,edgecolors="#888",linewidths=.2))
    ax.autoscale()
    if xlim: ax.set_xlim(*xlim)
    if ylim: ax.set_ylim(*ylim)
    ax.axis("off"); ax.set_title(title,fontsize=10.5*fscale,color=GREEN,fontweight="bold")
    sm=cm.ScalarMappable(norm=norm,cmap=cmap); sm.set_array([])
    cb=plt.colorbar(sm,ax=ax,orientation="horizontal",fraction=0.05,pad=0.02,aspect=32)
    cb.ax.tick_params(labelsize=9*fscale); cb.set_ticks(bins); cb.set_ticklabels([f"{int(b)}" for b in bins])
    WORLD=json.load(open(f"{SCR}/world_110m.geojson",encoding="utf-8"))
    WF=[f for f in WORLD["features"] if f["properties"].get("CONTINENT")!="Antarctica"]
    def iso3(f):
        p=f["properties"]; a=p.get("ISO_A3","-99"); return p.get("ADM0_A3") if a in ("-99",None,"") else a
    US=json.load(open(f"{SCR}/us_states.geojson",encoding="utf-8"))
    UF=[f for f in US["features"] if f["properties"]["name"] not in ("Alaska","Hawaii","Puerto Rico")]
    def fig_worldmaps(D,fname,label):
        fig,axs=plt.subplots(3,1,figsize=(7.0,9.6))
        for ax,m in zip(axs,["scientists","publications","citations"]):
            draw_panel(ax,WF,iso3,D["map_world_avg"][m],f"World - avg {METLAB[m].lower()}/yr ({label}, 2020-
    2024)",ylim=(-60,85))
        fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)
    def fig_usmaps(D,fname,label):
        # compact 1x3; figw 11 shrinks to ~6.7 on page -> fscale 1.64
        fig,axs=plt.subplots(1,3,figsize=(11.0,3.3))
        for ax,m in zip(axs,["scientists","publications","citations"]):
            draw_panel(ax,UF,lamba f:f["properties"]["name"],D["map_us_avg"][m],f"Avg {METLAB[m].lower()}/
    yr",xlim=(-125,-66),ylim=(24,50),fscale=1.64)
        fig.suptitle(f"United States - five-year averages
    ({label})",fontsize=10.5*1.64,color=GREEN,fontweight="bold",y=1.05)
        fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ----- INTENSITY ratio maps (differentiate what count-maps cannot): output & impact per unit -----
def fig_ratio_worldmaps(D,fname,label,min_sci_total=5):
    sci=D["map_world_avg"]["scientists"]; pub=D["map_world_avg"]["publications"]; cit=D["map_world_avg"]
    ["citations"]
    def rat(num,den):
        return {k:num[k]/den[k] for k in num if den.get(k,0)>0 and sci.get(k,0)*5>=min_sci_total}

```

```

panels=[("Publications per scientist (career output intensity)",rat(pub,sci)),
        ("Citations per scientist (career impact per person)",rat(cit,sci)),
        ("Citations per publication (impact per paper)",rat(cit,pub))]
fig,axs=plt.subplots(3,1,figsize=(7.0,9.6))
for ax,(t,vals) in zip(axs,panels):
    draw_panel(ax,WF,iso3,vals,f"{t} - {label}", 2020-2024,ylim=(-60,85),fscale=1.15)
fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ===== INTENSITY-RATIO FIGURES (quality vs quantity) =====
R_ART="#2F5233"; R_CAU="#2C6E9B"; R_CAR="#C6902F"
RSPEC=[("art_per_auth","Articles per author","Articles / author",R_ART,lambda v:f"{v:.0f}"),
        ("cit_per_auth","Citations per author","Citations / author",R_CAU,lambda v:f"{v/1000:.1f}k"),
        ("cit_per_art","Citations per article","Citations / article",R_CAR,lambda v:f"{v:.0f}")]

def fig_ratio_trend(R,fname,title):
    # the cohort's three intensity ratios over 2020->2024, one panel each (own scale), actual values labelled.
    # a small "+NN%" growth badge summarizes each ratio's five-year change.
    by=R["cohort"]["by_year"]
    fig,axs=plt.subplots(1,3,figsize=(11.0,3.9))
    for ax,(key,name,short,col,vf) in zip(axs,RSPEC):
        vals=by[key]; g=100*(vals[-1]/vals[0]-1)
        ax.plot(YEARS,vals,marker="o",ms=8,lw=2.8,color=col,zorder=3)
        for x,v in zip(YEARS,vals):
            ax.annotate(vf(v),(x,v),textcoords="offset points",xytext=(0,9),ha="center",va="bottom",
                        fontsize=11.5,color=col,fontweight="bold")
        ax.set_title(short,fontsize=14.5,color=col,fontweight="bold",pad=8)
        ax.set_xticks([2020,2022,2024]); ax.tick_params(axis="x",labelsize=11)
        lo=min(vals); hi=max(vals); ax.set_ylim(lo-0.30*(hi-lo)-1, hi+0.30*(hi-lo)+1)
        ax.set_yticks([]); ax.spines[["top","right","left"]].set_visible(False)
        ax.annotate(f"+{g:.0f}% (2020-2024)",(0.5,-0.19),xycoords="axes fraction",ha="center",va="top",
                    fontsize=11,color="#555",style="italic")
    fig.suptitle(title,fontsize=15,color=GREEN,fontweight="bold",y=1.03)
    fig.tight_layout(rect=[0,0.04,1,0.97]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

def fig_ratio_scatter(R,fname,title,min_sci_yr=5.0):
    # quality (citations/article) vs quantity (articles/author); bubble area = number of scientists;
    # quadrant lines at the cohort means. This is the report's central quality-vs-quantity figure.
    pts=[m for m in R["country"] if m["sci_yr"]>min_sci_yr and m["P"]>0]
    x=np.array([m["art_per_auth"] for m in pts]); y=np.array([m["cit_per_art"] for m in pts])
    sci=np.array([m["sci_yr"] for m in pts])
    mx=R["cohort"]["overall"]["art_per_auth"]; my=R["cohort"]["overall"]["cit_per_art"]
    fig,ax=plt.subplots(figsize=(8.8,6.4))
    xlo,xhi=x.min()-14,x.max()+22; ylo,yhi=y.min()-5,y.max()+7
    # faint quadrant shading (top-left = selective/high-impact; bottom-right = volume-oriented)
    ax.axvspan(xlo,mx,ymax=(my-ylo)/(yhi-ylo),ymin=0,color="#2F5233",alpha=0.05,zorder=0)
    ax.axvspan(mx,xhi,ymin=0,ymax=(my-ylo)/(yhi-ylo),color="#C6902F",alpha=0.06,zorder=0)
    ax.axvline(mx,color="#888",lw=1.1,ls="--",zorder=1); ax.axhline(my,color="#888",lw=1.1,ls="--",zorder=1)
    area=90+(sci/sci.max())*2100
    ax.scatter(x,y,s=area,c="#4C9F70",alpha=0.50,edgecolor="#2F5233",linewidth=1.0,zorder=3)
    # label every country; the central cluster is dense, so nudge problem labels off their bubbles
    # (cohort is locked, so fixed per-country offsets are stable). dx sign sets horizontal anchoring.
    NUDGE={"Norway":(-8,11),"Australia":(30,-3),"Switzerland":(18,9),"Germany":(-11,-15),
           "Finland":(-14,13),"Canada":(22,-14),"Sweden":(-22,-13),"Austria":(24,3),"Italy":(-16,-11),
           "France":(24,-6),"Spain":(27,1),"United States":(0,17),"China":(-24,11),"Portugal":(2,-15),
           "South Africa":(4,12),"Japan":(0,-15),"Netherlands":(0,-16),"Brazil":(0,13),"Denmark":(-6,11),
           "New Zealand":(0,-14),"Slovenia":(-10,11),"United Kingdom":(0,12)}
    for i,m in enumerate(pts):
        dx,dy=NUDGE.get(m["key"],(0, 10 if i%2==0 else -13))
        ha="left" if dx>2 else ("right" if dx<-2 else "center")
        va="bottom" if dy>0 else "top"
        ax.annotate(m["key"],(m["art_per_auth"],m["cit_per_art"]),textcoords="offset points",
                    xytext=(dx,dy),ha=ha,va=va,fontsize=10.5,
                    color="#222",fontweight=("bold" if m["sci_yr"]>25 else "normal"),zorder=5,
                    bbox=dict(boxstyle="round,pad=0.10",fc="white",ec="none",alpha=0.70))
    ax.set_xlim(xlo,xhi); ax.set_ylim(ylo,yhi)
    ax.set_xlabel("Quantity → articles per author (career output)",fontsize=13)
    ax.set_ylabel("Quality → citations per article",fontsize=13)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=14,pad=10)
    ax.spines[["top","right"]].set_visible(False)
    # quadrant captions placed in empty bands (top-left corner; right-middle just below the mean line)
    ax.text(0.015,0.975,"selective · high-impact",transform=ax.transAxes,fontsize=10.5,style="italic",
            color="#2F5233",ha="left",va="top")
    ax.text(0.985,0.46,"high-volume ·\nlower per-paper",transform=ax.transAxes,fontsize=10.5,style="italic",
            color="#A9781F",ha="right",va="top")
    ax.annotate(f"cohort mean\n({mx:.0f}, {my:.0f})",(mx,my),textcoords="offset points",xytext=(7,6),
                fontsize=9.5,color="#666",ha="left",va="bottom")
    ax.text(0.985,0.975,"bubble area scaled to no. of",transform=ax.transAxes,fontsize=9.5,color="#888",ha="right",va="top")
    ax.text(0.985,0.975,"scientists",transform=ax.transAxes,fontsize=9.5,color="#888",ha="right",va="top")
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

```

```

def _rshorten(s):
    s=str(s)
    for a,b in [("University","Univ."),("Institute","Inst."),("Technology","Tech."),
                ("Agricultural Sciences","Agric. Sci."),("Sciences","Sci."),("Northern America","N. America"),
                ("Australia & New Zealand","Australia & NZ"),("South America","S. America")]:
        s=s.replace(a,b)
    return s if len(s)<=37 else s[:36]+"..."

def fig_ratio_hbars(members,fname,title,n=None,min_sci_yr=0.0,foldcountry=False,rank_by=None):
    # three stacked horizontal-bar panels (one per ratio); categories shared, ordered by scientists (or rank_by)
    ms=[m for m in members if m["sci_yr"]>=min_sci_yr]
    if rank_by: ms=sorted(ms,key=lambda m:-m[rank_by])
    if n: ms=ms[:n]
    ms=ms[::-1]
    labels=[_rshorten(m["key"]+(f' ({m["country"]})' if foldcountry and m.get("country") else "")) for m in ms]
    yy=np.arange(len(ms))
    fig,axs=plt.subplots(1,3,figsize=(11.2,max(3.6,0.46*len(ms)+1.7)))
    for ax,(key,name,short,col,vf) in zip(axs,RSPEC):
        vals=[m[key] for m in ms]
        ax.barh(yy,vals,color=col,height=0.74,alpha=0.92)
        ax.set_yticks(yy)
        ax.set_yticklabels(labels if ax is axs[0] else [""]*len(ms),fontsize=12)
        ax.tick_params(axis="y",length=0,pad=2); ax.tick_params(axis="x",labelsize=11)
        for yi,v in zip(yy,vals):
            ax.annotate(" "+vf(v),(v,yi),va="center",ha="left",fontsize=11,color="#333")
        ax.set_title(short,fontsize=14,color=col,fontweight="bold")
        ax.set_xlim(0,max(vals)*1.20); ax.spines[["top","right"]].set_visible(False)
    fig.suptitle(title,fontsize=15.5,color=GREEN,fontweight="bold",y=1.005)
    fig.tight_layout(rect=[0,0,1,0.985]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

def fig_ref_scatter(refs,fname,title):
    # the two cohort reference scientists on the quantity(x=articles)-quality(y=cites/article) plane,
    # against the cohort-mean crosshair and the U.S. + shared-institution (UGA) means.
    # (John F. Nash is external / not in the dataset – discussed in prose, marked as an off-frame arrow.)
    cies=refs["cies"]; bett=refs["bett"]; b=refs["bench"]
    mx=b["cohort"]["art"]; my=b["cohort"]["cart"]
    fig,ax=plt.subplots(figsize=(8.4,5.6))
    ax.axvline(mx,ls="--",color="#bbb",lw=1.1,zorder=1); ax.axhline(my,ls="--",color="#bbb",lw=1.1,zorder=1)
    ax.annotate(f"cohort mean\n({mx:.0f} articles, {my:.0f} cites/paper)",(mx,my),textcoords="offset points",
               xytext=(8,8),fontsize=10,color="#888",ha="left",va="bottom")
    for x,y,lab,dx,dy,ha in [(b["us"]["art"],b["us"]["cart"],"U.S. mean",8,2,"left"),
                             (b["uga"]["art"],b["uga"]["cart"],"UGA mean",8,-4,"left")]:
        ax.scatter([x],[y],s=80,c="#9AA0A6",marker="s",zorder=3)
        ax.annotate(lab,(x,y),textcoords="offset
points",xytext=(dx,dy),fontsize=10.5,color="#666",ha=ha,va="center")
    ax.scatter([cies["np2024"]],
[cies["cpp2024"]],s=420,c=GOLD,marker="x",edgecolor="#7a5a12",linewidth=1.1,zorder=5)
    ax.scatter([bett["np2024"]],
[bett["cpp2024"]],s=200,c=TEAL,marker="D",edgecolor="#1c4a68",linewidth=1.1,zorder=5)
    ax.annotate(f"Cieszewski (this author)\n{cies['np2024']:.0f} papers, {cies['cpp2024']:.0f} cites/paper",
               (cies["np2024"],cies["cpp2024"]),textcoords="offset
points",xytext=(0,16),ha="center",va="bottom",
               fontsize=11.5,fontweight="bold",color="#222")
    ax.annotate(f"Bettinger\n{bett['np2024']:.0f} papers, {bett['cpp2024']:.0f} cites/paper",
               (bett["np2024"],bett["cpp2024"]),textcoords="offset
points",xytext=(0,17),ha="center",va="bottom",
               fontsize=11.5,fontweight="bold",color="#222")
    # John F. Nash lies far off the top-left (23 papers, thousands of cites each) – mark the direction, not a
    point
    ax.annotate("J. F. Nash – off scale\n(23 papers, not in the dataset;\nthousands of citations each)",
               xy=(6,60.5),xytext=(14,52),fontsize=10.5,color="#6e2325",fontstyle="italic",ha="left",va="top",
               arrowprops=dict(arrowstyle="-|>",color="#6e2325",lw=1.4))
    # Daniel Kahneman: in the database, ~196 papers and ~571 cites/paper – straight off the top above Bettinger
    ax.annotate("Kahneman – off scale (in the dataset, near its top)\n196 papers, ~571 citations per paper",
               xy=(200,61),xytext=(158,59.5),fontsize=10.5,color="#2C6E9B",fontstyle="italic",ha="right",va="top",
               arrowprops=dict(arrowstyle="-|>",color="#2C6E9B",lw=1.4))
    ax.set_xlim(0,235); ax.set_ylim(8,62)
    ax.set_xlabel("Quantity → articles (career output)",fontsize=13)
    ax.set_ylabel("Quality → citations per article",fontsize=13)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=14,pad=8)
    ax.spines[["top","right"]].set_visible(False)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

def fig_group_ratio(R,fname,title):
    # Forestry vs Ecology group on the three ratios, as grouped bars normalized within each ratio
    groups=["Forestry","Ecology"]; gcol={"Forestry":GOLD,"Ecology":TEAL} if False else
{"Forestry": "#C6902F","Ecology": "#2C6E9B"}

```

```

fig,axs=plt.subplots(1,3,figsize=(9.0,4.0))
for ax,(key,name,short,col,vf) in zip(axs,RSPEC):
    vals=[R["group"][g][key] for g in groups]
    bars=ax.bar([0,1],vals,color=[gcol[g] for g in groups],width=0.64)
    for xi,v in zip([0,1],vals):
        ax.annotate(vf(v),(xi,v),textcoords="offset points",xytext=(0,3),ha="center",va="bottom",
                    fontsize=12,fontweight="bold",color="#333")
    ax.set_xticks([0,1]); ax.set_xticklabels(groups,fontsize=12)
    ax.set_title(short,fontsize=13.5,color=col,fontweight="bold")
    ax.set_ylim(0,max(vals)*1.20); ax.spines[["top","right"]].set_visible(False)
    ax.tick_params(axis="y",labelsize=10.5)
fig.suptitle(title,fontsize=15,color=GREEN,fontweight="bold",y=1.02)
fig.tight_layout(rect=[0,0,1,0.97]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ===== AUTHORSHIP-POSITION FIGURES =====
APOS=[("single","Single-authored","#2F5233"),("first","First-authored","#2C6E9B"),
      ("last","Last-authored","#C6902F"),("middle","Middle / other","#B8B8B8")]
def fig_authorship_composition(A,fname,title):
    groups=["Ecology","Forestry","cohort"] # cohort drawn on top
    glabel={"cohort":"Whole cohort","Forestry":"Forestry group","Ecology":"Ecology group"}
    fig,ax=plt.subplots(figsize=(8.4,3.7))
    for gi,g in enumerate(groups):
        left=0
        for pos,lab,col in APOS:
            w=A["share_c"][g][pos]
            ax.barh(gi,w,left=left,color=col,edgecolor="white",height=0.62)
            if w>=4.5:
                ax.annotate(f"{w:.0f}%",(left+w/2,gi),ha="center",va="center",fontsize=12,
                           color="white" if pos!="middle" else "#333",fontweight="bold")
            left+=w
    ax.set_yticks(range(len(groups))); ax.set_yticklabels([glabel[g] for g in groups],fontsize=12.5)
    ax.set_xlim(0,100); ax.set_xlabel("Share of citations (%)",fontsize=12.5)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=14,pad=8)
    ax.spines[["top","right","left"]].set_visible(False); ax.tick_params(axis="y",length=0)
    handles=[mp.Patch(color=c,label=l) for _,l,c in APOS]
    ax.legend(handles=handles,loc="lower center",bbox_to_anchor=(0.5,-0.40),ncol=4,frameon=False,fontsize=11.5)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

def fig_rank_correlation(A,fname,title):
    rows=[("single + first + last ("key" authorship)","key: single+first+last (ncsf1)","#2F5233"),
          ("single + first ("lead" authorship)","lead: single+first (ncsf)","#2C6E9B"),
          ("single only","single (ncs)","#A63D40")]
    corr=A["corr"]; ov=A["overlap"]
    y=np.arange(len(rows))[:-1]
    fig,ax=plt.subplots(figsize=(8.6,3.5))
    for yi,(lab,key,col) in zip(y,rows):
        v=corr[key]; ax.barh(yi,v,color=col,height=0.58)
        ax.annotate(f" ρ = {v:.2f} ({ov[key]} of the top 25 overlap)",(max(v,0),yi),va="center",ha="left",
                   fontsize=11.5,color="#333")
    ax.axvline(0,color="#888",lw=1)
    ax.set_yticks(y); ax.set_yticklabels([r[0] for r in rows],fontsize=12)
    ax.set_xlim(-0.2,1.2); ax.set_xlabel("Spearman correlation with the undiscriminated (total-citation) ranking
of the 854 scientists",fontsize=11.5)
    ax.set_title(title,color=GREEN,fontweight="bold",fontsize=13.5,pad=8)
    ax.spines[["top","right","left"]].set_visible(False); ax.tick_params(axis="y",length=0)
    fig.tight_layout(); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)

# ===== REAL TREE GRAPH (branched) =====
DOM_ORDER=["Applied Sciences","Natural Sciences","Health Sciences","Economic & Social Sciences","Arts &
Humanities","(unmapped)"]
def draw_tree(D,fname,title,leaf_threshold=10):
    tree=D["tree"]
    byd={}
    omitted=0; omit_hits=0
    for sf,info in tree.items():
        if info["total"]<leaf_threshold and not info["target"]:
            omitted+=1; omit_hits+=info["total"]; continue
        byd.setdefault(info["domain"],[]).setdefault(info["field"],[]).append((sf,info))
    doms=[d for d in DOM_ORDER if d in byd]+[d for d in byd if d not in DOM_ORDER]
    leaf_rows=[]; dom_span=[]; fld_span={}
    yy=0.0
    for d in doms:
        d0=yy
        for f in sorted(byd[d],key=lambda ff:-sum(i["total"] for _,i in byd[d][ff])):
            f0=yy
            for sf,info in sorted(byd[d][f],key=lambda t:-t[1]["total"]):
                leaf_rows.append((d,f,sf,info,yy)); yy+=1
            fld_span[(d,f)]=(f0,yy-1)
            yy+=0.5
        dom_span[d]=(d0,yy-1.5)

```

```

        yy+=1.0
nrows=yy
fig_h=max(7.5,0.315*len(leaf_rows)+1.6)
fig,ax=plt.subplots(figsize=(9.0,fig_h))          # wide canvas -> on-page fonts ~body size
ax.set_xlim(0,1); ax.set_ylim(-1.5,nrows+0.5); ax.invert_yaxis(); ax.axis("off")
# column x-anchors and bus (elbow) x-positions
XD,XF,XL,XCNT=0.005,0.185,0.560,0.988
BUSD,BUSF=0.150,0.545
def ymid(a,b): return (a+b)/2
segs=[]
for d in doms:                                     # domain -> fields
    dy=ymid(*dom_span[d])
    ax.text(XD,dy,d,fontsize=13,fontweight="bold",color="#141414",va="center")
    for f in byd[d]:
        fy=ymid(*fld_span[(d,f)])
        segs+=[[ (BUSD,fy), (BUSD,fy) ], [ (BUSD,fy), (XF-0.004,fy) ]]
    segs+=[[ (XD+0.002,dy), (BUSD,dy) ]]
for (d,f),(a,b) in fld_span.items():               # field -> leaves
    fy=ymid(a,b)
    ax.text(XF,fy,f,fontsize=12,color=GREEN,fontweight="bold",va="center")
    for (_,sf,info,ly) in [r for r in leaf_rows if r[0]==d and r[1]==f]:
        segs+=[[ (BUSF,fy), (BUSF,ly) ], [ (BUSF,ly), (XL-0.004,ly) ]]
    segs+=[[ (XF+0.004,fy), (BUSF,fy) ]]
for (d,f,sf,info,ly) in leaf_rows:                 # leaves + counts
    col=GOLD if info["target"] else (BLUE if info["quant"] else "#3a3a3a")
    mk="●" if info["target"] else ("»" if info["quant"] else "•")
    name=sf if len(sf)<=34 else sf[:33]+"..."
    ax.text(XL,ly,f"{mk} {name}",fontsize=11.5,color=col,va="center")
    ax.text(XCNT,ly,f"{info['total']}:,"),fontsize=11,color="#555",va="center",ha="right")
ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.1,zorder=0))
note=(f"{'omitted'} rarer sub-sub-fields (<{leaf_threshold} records, {omit_hits}, records total) "
      "omitted for legibility; full list in the companion workbook.") if omitted else ""
if note: fig.text(0.5,0.006,note,fontsize=10.5,color="#777",ha="center",style="italic")
fig.legend(handles=[mp.Patch(color=GOLD,label="Target sub-sub-field (Forestry)"),
                    mp.Patch(color=BLUE,label="Quantitative sub-sub-field (»)"),
                    mp.Patch(color="#3a3a3a",label="Other / qualitative (•)"),],
            frameon=False,fontsize=11.5,ncol=3,loc="lower center",bbox_to_anchor=(0.5,0.028))
fig.suptitle(title,fontsize=14,color=GREEN,fontweight="bold",y=0.998)
fig.tight_layout(rect=[0,0.055,1,0.982]); fig.savefig(fname,dpi=200); plt.close(fig)
return len(leaf_rows),omitted

```

===== REAL TREE: primary-forestry (sm-subfield-1==Forestry) =====

```

def draw_forestry_tree(tax,fname,title,big=10,quant_min=5):
    branches=tax["branches"]; QSET=set(tax["quant_set"])
    def keep(sf,n): return (n>=big) or (sf in QSET and n>=quant_min)
    kept={}; omitted=0; omit_hits=0
    for f in branches:
        leaves={sf:n for sf,n in branches[f].items() if keep(sf,n)}
        drop={sf:n for sf,n in branches[f].items() if not keep(sf,n)}
        omitted+=len(drop); omit_hits+=sum(drop.values())
        if leaves: kept[f]=leaves
    fields=sorted(kept,key=lambda f:-sum(kept[f].values()))
    leaf_rows=[]; fld_span={}; yy=0.0
    for f in fields:
        f0=yy
        for sf in sorted(kept[f],key=lambda x:-kept[f][x]):
            leaf_rows.append((f,sf,kept[f][sf],yy)); yy+=1
            fld_span[f]=(f0,yy-1); yy+=0.6
    nrows=yy; rootY=(0+nrows-1.6)/2
    fig_h=max(7.0,0.33*len(leaf_rows)+1.6)
    fig,ax=plt.subplots(figsize=(9.2,fig_h))
    ax.set_xlim(0,1); ax.set_ylim(-1.4,nrows+0.3); ax.invert_yaxis(); ax.axis("off")
    XR,XF,XL,XCNT=0.01,0.26,0.60,0.988; BUSR,BUSF=0.235,0.575
    segs=[]
    ax.text(XR,rootY,"Forestry",fontsize=14.5,fontweight="bold",color=GOLD,va="center")
    ax.text(XR,rootY+1.05,f"primary specialty; n={tax['n_primary']}:,"),fontsize=9.5,color="#777",va="center")
    for f in fields:
        fy=(fld_span[f][0]+fld_span[f][1])/2
        segs+=[[ (BUSR,rootY), (BUSR,fy) ], [ (BUSR,fy), (XF-0.004,fy) ]]
    segs+=[[ (XR+0.11,rootY), (BUSR,rootY) ]]
    for f in fields:
        a,b=fld_span[f]; fy=(a+b)/2
        ax.text(XF,fy,f,fontsize=11.5,color=GREEN,fontweight="bold",va="center")
        for (ff,sf,n,ly) in [r for r in leaf_rows if r[0]==f]:
            segs+=[[ (BUSF,fy), (BUSF,ly) ], [ (BUSF,ly), (XL-0.004,ly) ]]
        segs+=[[ (XF+0.004,fy), (BUSF,fy) ]]
    for (f,sf,n,ly) in leaf_rows:
        q=sf in QSET; col=BLUE if q else "#3a3a3a"; mk="»" if q else "•"
        name=sf if len(sf)<=34 else sf[:33]+"..."
        ax.text(XL,ly,f"{mk} {name}",fontsize=11,color=col,va="center")

```

```

        ax.text(XCNT,ly,f"{n:,"},", fontsize=10.5,color="#555",va="center",ha="right")
    ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.1,zorder=0))
    note=(f"Minor secondary specialties (<{big} authors, and non-quantitative) omitted for legibility: "
        f"{omitted} sub-sub-fields, {omit_hits:,"} authors. Full list in the workbook.") if omitted else ""
    if note: fig.text(0.5,0.006,note,fontsize=10,color="#777",ha="center",style="italic")
    fig.legend(handles=[mp.Patch(color=GOLD,label="Forestry (primary sub-field, sm-subfield-1)"),
        mp.Patch(color=GREEN,label="Field of secondary specialty (sm-field of sm-subfield-2)"),
        mp.Patch(color=BLUE,label="Quantitative secondary sub-sub-field (>)")],
        frameon=False,fontsize=10.5,ncol=1,loc="lower left",bbox_to_anchor=(0.01,0.01))
    fig.suptitle(title,fontsize=13.5,color=GREEN,fontweight="bold",y=0.998)
    fig.tight_layout(rect=[0,0.02,1,0.982]); fig.savefig(fname,dpi=200); plt.close(fig)
    return len(leaf_rows),omitted

# ===== MULTI-ROOT TREE: AQ(sm-field) -> AM(sm-subfield-1) -> AO(sm-subfield-2) =====
TEAL="#2C6E9B"
def
draw_multiroot_tree(tax,fname,title,root_min=15,leaf_min=12,quant_min=5,only_roots=None,row_h=0.28,show_note=True
):
    roots=tax["roots"]; QSET=set(tax["quant_set"])
    root_tot={aq:sum(sum(a.values()) for a in roots[aq].values()) for aq in roots}
    keep=[aq for aq in sorted(roots,key=lambda x:-root_tot[x]) if root_tot[aq]>=root_min]
    drop_roots=[aq for aq in roots if aq not in keep]; drop_n=sum(root_tot[aq] for aq in drop_roots)
    if only_roots is not None: # render only this subset of roots (for multi-page splits)
        want=set(only_roots); keep=[aq for aq in keep if aq in want]
    leafrows=[]; am_span={}; aq_span={}; yy=0.0
    for aq in keep:
        aq0=yy
        for am in sorted(roots[aq],key=lambda x:-sum(roots[aq][x].values())):
            am0=yy; aos=roots[aq][am]; CAP=15
            kept=sorted([(ao,n) for ao,n in aos.items() if n>=leaf_min or (ao in QSET and
n>=quant_min)],key=lambda t:-t[1])
            dropped=[(ao,n) for ao,n in aos.items() if not (n>=leaf_min or (ao in QSET and n>=quant_min))]
            dropped+=kept[CAP:]; kept=kept[:CAP] # cap leaves per branch
            for ao,n in kept: leafrows.append((aq,am,ao,n,yy)); yy+=1
            if dropped: leafrows.append((aq,am,f"({len(dropped)} minor)",sum(n for _,n in dropped),yy)); yy+=1
            am_span[(aq,am)]=(am0,yy-1); yy+=0.5
        aq_span[aq]=(aq0,yy-1.5); yy+=1.0
    nrows=yy
    fig_h=max(6.0,row_h*len(leafrows)+1.9)
    fig,ax=plt.subplots(figsize=(9.4,fig_h))
    ax.set_xlim(0,1); ax.set_ylim(-1.5,nrows+0.3); ax.invert_yaxis(); ax.axis("off")
    XQ,XM,XL,XC=0.005,0.30,0.60,0.99; BUSQ,BUSM=0.27,0.575
    def ymid(sp): return (sp[0]+sp[1])/2
    segs=[]
    for aq in keep: # AQ roots -> AM
        qy=ymid(aq_span[aq])
        name=aq if len(aq)<=30 else aq[:29]+"..."
        ax.text(XQ,qy,name,fontsize=11.5,fontweight="bold",color="#141414",va="center")
        for am in roots[aq]:
            if (aq,am) not in am_span: continue
            my=ymid(am_span[(aq,am)]); segs+=[[ (BUSQ,qy), (BUSQ,my)], [(BUSQ,my), (XM-0.004,my)]]
            segs+=[[ (XQ+0.005,qy), (BUSQ,qy)]]
    for (aq,am),sp in am_span.items(): # AM -> AO leaves
        my=ymid(sp); amcol=GOLD if am=="Forestry" else TEAL
        ax.text(XM,my,f"{am}",fontsize=11,fontweight="bold",color=amcol,va="center")
        for (a,m,ao,n,ly) in [r for r in leafrows if r[0]==aq and r[1]==am]:
            segs+=[[ (BUSM,my), (BUSM,ly)], [(BUSM,ly), (XL-0.004,ly)]]
            segs+=[[ (XM+0.004,my), (BUSM,my)]]
    for (aq,am,ao,n,ly) in leafrows: # AO leaves + counts
        if ao.startswith("("): col="#9AA0A6"; mk=""
        elif ao=="Forestry": col=GOLD; mk="● "
        elif ao in QSET: col=BLUE; mk="» "
        else: col="#3a3a3a"; mk="• "
        nm=ao if len(ao)<=32 else ao[:31]+"..."
        ax.text(XL,ly,f"{mk}{nm}",fontsize=10.5,color=col,va="center")
        ax.text(XC,ly,f"{n:,"},",fontsize=10,color="#555",va="center",ha="right")
    ax.add_collection(LineCollection(segs,colors="#AEB9B2",linewidths=1.05,zorder=0))
    if show_note:
        note=(f"Roots (sm-field) with <{root_min} authors omitted: {len(drop_roots)} fields, {drop_n:,"} authors.
        "
        "Minor secondary specialties aggregated per branch. Full lists in the workbook.")
        fig.text(0.5,0.005,note,fontsize=9.5,color="#777",ha="center",style="italic")
    fig.legend(handles=[mp.Patch(color=GOLD,label="AM = Forestry (Forestry group)"),
        mp.Patch(color=TEAL,label="AM = Ecology (Ecology group -> secondary Forestry)"),
        mp.Patch(color=BLUE,label="Quantitative secondary sub-sub-field (>)")],
        frameon=False,fontsize=10.5,ncol=1,loc="lower left",bbox_to_anchor=(0.01,0.015))
    fig.suptitle(title,fontsize=12,color=GREEN,fontweight="bold",y=0.997)
    fig.tight_layout(rect=[0,0.04,1,0.978]); fig.savefig(fname,dpi=200,bbox_inches="tight"); plt.close(fig)
    return len(leafrows),len(keep)

```

```

# ===== BUILD FORESTRY (main = union: Forestry group + Ecology group) =====
fig_change(FORE,f"{SCR}/ff1_change.png","Forestry research cohort – growth of each metric since 2020 (actual values labelled)")
# --- taxonomy tree split across two pages (was overflowing one page) ---
_TAX=MERGED["taxonomy"]; _PART1=["Agriculture, Fisheries & Forestry","Biology"]
_rt={aq:sum(sum(a.values()) for a in _TAX["roots"][aq].values()) for aq in _TAX["roots"]}
_keep=[aq for aq in sorted(_rt,key=lambda x:-_rt[x]) if _rt[aq]>=15]
_part2=[aq for aq in _keep if aq not in _PART1]
draw_multiroot_tree(_TAX,f"{SCR}/ff2_tree_1.png",
    "Forestry research taxonomy (part 1 of 2) – principal roots: field (AQ) → primary subfield (AM) → secondary subfield (AO)",
    only_roots=_PART1,row_h=0.42,show_note=False)
draw_multiroot_tree(_TAX,f"{SCR}/ff2_tree_2.png",
    "Forestry research taxonomy (part 2 of 2) – cross-field roots: field (AQ) → primary subfield (AM) → secondary subfield (AO)",
    only_roots=_part2,row_h=0.42,show_note=True)
fig_onebar(FORE,"group",f"{SCR}/ff4_group.png",
    "Forestry cohort: group share of scientists, publications and citations (2020-2024)",figw=6.0,disp=5.7,figh=4.6,labfrac=0.16,share=True)
fig_onebar(FORE,"continent",f"{SCR}/ff3_continent.png",
    "Forestry: continent share of scientists, publications and citations (2020-2024)",figw=7.6,disp=6.7,figh=4.9,labfrac=0.19,share=True)
fig_onebar(FORE,"region",f"{SCR}/ff_region.png",
    "Forestry: world sub-region share of scientists, publications and citations (top 8, 2020-2024)",figw=9.2,n=8,disp=6.7,figh=5.4,labfrac=0.26,share=True)
fig_onebar(FORE,"country",f"{SCR}/ff5_country.png",
    "Forestry: country share of scientists, publications and citations (top 8, 2020-2024)",figw=9.2,n=8,disp=6.7,figh=5.0,labfrac=0.20,share=True)
fig_band(FORE["country_banded"]["scientists"],f"{SCR}/ff6_country_band.png","Forestry countries: top 25 by average scientists per year","avg scientists / yr",figh=6.6,fs=11)
fig_onebar(FORE,"university",f"{SCR}/ff7_university.png",
    "Forestry: scientists, publications and citations by institution (top 5, % of each institution's own maximum)",
    figw=9.0,n=5,disp=6.7,figh=5.9,labfrac=0.32,percat=True)
fig_band(FORE["university_banded"]["scientists"],f"{SCR}/ff8_university_band.png","Forestry institutions: top 25 by average scientists per year","avg scientists / yr",figh=6.8,fs=11)
fig_ratio_worldmaps(FORE,f"{SCR}/ff9_worldmaps.png","Forestry plus Ecology")
fig_usmaps(FORE,f"{SCR}/ff10_usmaps.png","Forestry plus Ecology")
# ---- intensity-ratio figures (quality vs quantity) ----
fig_ratio_trend(RATIOS,f"{SCR}/ff_ratio_trend.png",
    "Forestry cohort: the three intensity ratios rose steadily, 2020-2024 (each panel on its own scale)")
fig_group_ratio(RATIOS,f"{SCR}/ff_ratio_group.png",
    "Forestry group vs Ecology group: output and impact intensity (five-year values)")
fig_ratio_hbars(RATIOS["continent"],f"{SCR}/ff_ratio_continent.png",
    "Intensity ratios by continent (five-year values, ordered by number of scientists)")
fig_ratio_hbars(RATIOS["region"],f"{SCR}/ff_ratio_region.png",
    "Intensity ratios by world sub-region (top 8 by scientists, five-year values)",n=8)
fig_ratio_scatter(RATIOS,f"{SCR}/ff_ratio_scatter.png",
    "Quality vs quantity by country (forestry cohort, 2020-2024; ≥5 scientists per year)")
fig_ratio_hbars(RATIOS["university"],f"{SCR}/ff_ratio_institution.png",
    "Intensity ratios for the 12 most-represented institutions (five-year values)",n=12,foldcountry=True)
# reference-scientist comparison (Discussion): two individuals vs cohort/US/UGA benchmarks
_REFS=json.load(open(f"{SCR}/refs.json",encoding="utf-8"))
fig_ref_scatter(_REFS,f"{SCR}/ff_refs.png",
    "Two cohort reference scientists on the quality-quantity plane (2024 career values)")
# authorship-position figures
_AUTH=json.load(open(f"{SCR}/authorship.json",encoding="utf-8"))
fig_authorship_composition(_AUTH,f"{SCR}/ff_authcomp.png",
    "Where the cohort's citations come from, by authorship position (2020-2024)")
fig_rank_correlation(_AUTH,f"{SCR}/ff_authrank.png",
    "Ranking scientists by total citations vs by authorship-discriminated citations")

# ===== APPENDIX A: the Ecology group only (criterion 2 = AM=Ecology & AO=Forestry, 615) =====
ECO=json.load(open(f"{SCR}/forestry_E_data.json",encoding="utf-8"))
fig_change(ECO,f"{SCR}/ef1_change.png","Ecology group (criterion 2) – growth of each metric since 2020 (actual values labelled)")
fig_onebar(ECO,"continent",f"{SCR}/ef2_continent.png","Ecology group: continent share of scientists, publications and citations (2020-2024)",figw=7.6,disp=6.7,figh=4.9,labfrac=0.19,share=True)
fig_onebar(ECO,"country",f"{SCR}/ef3_country.png","Ecology group: country share of scientists, publications and citations (top 8, 2020-2024)",figw=9.2,n=8,disp=6.7,figh=5.0,labfrac=0.20,share=True)
fig_band(ECO["country_banded"]["scientists"],f"{SCR}/ef4_country_band.png","Ecology-group countries: top 25 by average scientists per year","avg scientists / yr",figh=6.6,fs=11)
fig_band(ECO["university_banded"]["scientists"],f"{SCR}/ef5_university_band.png","Ecology-group institutions: top 25 by average scientists per year","avg scientists / yr",figh=6.6,fs=11)
fig_ratio_hbars(RATIOS["E_continent"],f"{SCR}/ef_ratio_continent.png",
    "Ecology group: intensity ratios by continent (five-year values, ordered by number of scientists)")

# ===== APPENDIX B: Forestry group (criterion 1), SINGLE-AUTHOR publications only =====
# metrics: scientists = list entries with >=1 single-authored paper; publications = single-author papers (nps);

```

```

# citations = citations to those single-author papers (ncs).
SA_VF=lambda m,v:(f"{int(round(v))}:" if m=="scientists" else (f"{v/1000:,.1f}k" if m=="publications" else
f"{v/1000:,.0f}k"))
SA_AF=lambda m,v:(f"{int(round(v))}:" if m=="scientists" else (f"{int(round(v))}:" if m=="publications" else
f"{v/1000:,.0f}k"))
BSA=json.load(open(f"{SCR}/forestry_B_data.json",encoding="utf-8"))
fig_change(BSA,f"{SCR}/bf1_change.png","Forestry group (criterion 1), single-author publications – growth of each
metric since 2020 (actual values labelled)",valfmt=SA_VF)
fig_onebar(BSA,"continent",f"{SCR}/bf2_continent.png","Forestry group, single-author: continent share of
scientists, single-author papers and their citations (2020-
2024)",figw=7.6,disp=6.7,figh=4.9,labfrac=0.19,share=True,afmt=SA_AF)
fig_onebar(BSA,"country",f"{SCR}/bf3_country.png","Forestry group, single-author: country share of scientists,
single-author papers and their citations (top 8, 2020-
2024)",figw=9.2,n=8,disp=6.7,figh=5.0,labfrac=0.20,share=True,afmt=SA_AF)
fig_band(BSA["country_banded"]["scientists"],f"{SCR}/bf4_country_band.png","Forestry single-author: top 25
countries by average scientists per year","avg scientists / yr",figh=6.6,fs=11)
fig_band(BSA["university_banded"]["scientists"],f"{SCR}/bf5_university_band.png","Forestry single-author: top 25
institutions by average scientists per year","avg scientists / yr",figh=6.8,fs=11)

# ===== APPENDIX C: Ecology group (criterion 2), SINGLE-AUTHOR publications only =====
CSA=json.load(open(f"{SCR}/forestry_C_data.json",encoding="utf-8"))
fig_change(CSA,f"{SCR}/cf1_change.png","Ecology group (criterion 2), single-author publications – growth of each
metric since 2020 (actual values labelled)",valfmt=SA_VF)
fig_onebar(CSA,"continent",f"{SCR}/cf2_continent.png","Ecology group, single-author: continent share of
scientists, single-author papers and their citations (2020-
2024)",figw=7.6,disp=6.7,figh=4.9,labfrac=0.19,share=True,afmt=SA_AF)
fig_onebar(CSA,"country",f"{SCR}/cf3_country.png","Ecology group, single-author: country share of scientists,
single-author papers and their citations (top 8, 2020-
2024)",figw=9.2,n=8,disp=6.7,figh=5.0,labfrac=0.20,share=True,afmt=SA_AF)
fig_band(CSA["country_banded"]["scientists"],f"{SCR}/cf4_country_band.png","Ecology single-author: top 25
countries by average scientists per year","avg scientists / yr",figh=6.4,fs=11)
fig_band(CSA["university_banded"]["scientists"],f"{SCR}/cf5_university_band.png","Ecology single-author: top 25
institutions by average scientists per year","avg scientists / yr",figh=6.6,fs=11)

print("figures written OK (main = Forestry plus Ecology; App A = Ecology group; App B = Forestry single-author;
App C = Ecology single-author)")

```

60. refine_forestry.py (SHA-256 197677243197da31...; 118 lines)

```

# -*- coding: utf-8 -*-
# Build Forestry-ONLY data (main report). Combined data stays in refine_data.json (appendix).
import json, pandas as pd, numpy as np
from cc_defs import CC
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS = [2020,2021,2022,2023,2024]

df = pd.read_excel(CAREER, sheet_name="Data")
sf1=df["sm-subfield-1"].astype(str).str.strip(); sf2=df["sm-subfield-2"].astype(str).str.strip()
# Corrected selection: (AM=Forestry) OR (AM=Ecology AND AO=Forestry)
df = df[(sf1=="Forestry") | ((sf1=="Ecology") & (sf2=="Forestry"))].copy()
df["country"] = df["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
df["continent"] = df["cntry"].map(lambda c: CC[c][1] if c in CC else "(unknown)")
df["region"] = df["cntry"].map(lambda c: CC[c][2] if c in CC else "(unknown)")

subdiv = {int(k): v for k,v in json.load(open(f"{SCR}/inst_subdiv.json", encoding="utf-8")).items()}
idx = pd.read_csv(f"{SCR}/inst_index.csv"); idx["sd"] = idx["id"].map(subdiv).fillna("Unknown")
look = {(r.country, r.inst): r.sd for r in idx.itertuples()}
CN = {"usa":"United States","can":"Canada","aus":"Australia"}
df["subdivision"] = df.apply(lambda r: look.get((CN.get(r["cntry"]), r["inst_name"]), "Unknown") if r["cntry"] in
CN else None, axis=1)

# 2-group split: Forestry group (AM=Forestry) vs Ecology group (AM=Ecology & AO=Forestry)
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
df["group"]=np.where(s1=="Forestry","Forestry","Ecology")
# numeric metric columns: np/nc (all publications) and nps/ncs (single-author publications & their citations)
for _c in ["np","nc","nps","ncs"]:
    df[_c]=pd.to_numeric(df[_c],errors="coerce").fillna(0).astype(int)

# ---- quantitative subfields (Science-Matrix labels); incl. Operations Research + computational/programming ----
QUANT = {
    # Mathematics & Statistics (Mathematical)
    "Applied Mathematics","General Mathematics","Numerical & Computational Mathematics","Statistics & Probability",
    # ICT (Programming / Computational)
    "Artificial Intelligence & Image Processing","Computation Theory & Mathematics","Computer Hardware &
Architecture",

```

```

"Distributed Computing","Information Systems","Medical Informatics","Networking & Telecommunications","Software Engineering",
# Engineering (incl. Operations Research; Geomatics ~ GIS/mensuration; Environmental Eng.)
"Aerospace & Aeronautics","Automobile Design & Engineering","Biomedical Engineering","Chemical Engineering",
"Civil Engineering","Electrical & Electronic Engineering","Environmental Engineering","Geological & Geomatics Engineering",
"Industrial Engineering & Automation","Mechanical Engineering & Transports","Mining & Metallurgy","Operations Research",
"Bioinformatics",
# user-named forestry sub-disciplines NOT present as Science-Matrix labels (kept for transparency; match nothing):
"Silviculture","Biometrics","Mensuration","Forest Management","Remote Sensing","GIS","Programming",
}
def is_quant(row):
    return (str(row["sm-subfield-1"]).strip() in QUANT) or (str(row["sm-subfield-2"]).strip() in QUANT)
df["orient"] = np.where(df.apply(is_quant, axis=1), "Quantitative-oriented", "Qualitative-oriented")

def agg(frame, keycol, metric, pubcol="np", citecol="nc"):
    val, fn = ("authfull","count") if metric=="scientists" else ((pubcol,"sum") if metric=="publications" else (citecol,"sum"))
    t = frame.pivot_table(index=keycol, columns="year", values=val, aggfunc=fn, fill_value=0)
    for y in YEARS:
        if y not in t.columns: t[y]=0
    t = t[YEARS]; t["total"]=t.sum(axis=1); t["avg"]=t[YEARS].mean(axis=1).round(1)
    return t

def recs_by_metric(frame, keycol, extra_country=False, pubcol="np", citecol="nc"):
    inst_country=None
    if extra_country:
        inst_country = frame.groupby(keycol)["country"].agg(lambda s: s.value_counts().index[0])
    out={}
    for m in ["scientists","publications","citations"]:
        a=agg(frame,keycol,m,pubcol,citecol).sort_values("total",ascending=False); lst=[]
        for k in a.index:
            r=dict(key=k,**{str(y):int(a.loc[k,y]) for y in YEARS},total=int(a.loc[k,"total"]),avg=float(a.loc[k,"avg"]))
            if extra_country: r["country"]=inst_country.get(k,"")
            lst.append(r)
        out[m]=lst
    return out

def band(records_list, n_individual=25, cap=None, target_frac=0.10):
    # straight top-N ranking (cumulative bands dropped per user request)
    return {"head":records_list[:n_individual],"bands":[],"remainder_total":0,"n_total":len(records_list)}

th=pd.read_excel(f"{SCR}/thresholds_career_2024.xlsx",sheet_name="SubField"); th.columns=[c.strip() for c in th.columns]
sf2fd={r["Subfield"].strip():(str(r["Domain"]).strip(),str(r["Field"]).strip()) for _,r in th.iterrows() if isinstance(r["Subfield"],str)}

def build(dfx, outfile, cohort_name, with_group=True, pubcol="np", citecol="nc"):
    OUT={"years":YEARS,"cohort":cohort_name}
    rbm=lambda frame,keycol,**kw: recs_by_metric(frame,keycol,pubcol=pubcol,citecol=citecol,**kw)
    OUT["continent"]=rbm(dfx[dfx.continent!="(unknown)"],"continent")
    OUT["region"]=rbm(dfx[dfx.region!="(unknown)"],"region")
    OUT["country"]=rbm(dfx[dfx.country!="(unknown)"],"country")
    if with_group: OUT["group"]=rbm(dfx,"group")
    OUT["orient"]=rbm(dfx,"orient")
    OUT["university"]=rbm(dfx[dfx["inst_name"].notna()], "inst_name",extra_country=True)
    for code,name in [("usa","us_state"),("can","ca_prov"),("aus","au_state")]:
        OUT[name]=rbm(dfx[dfx.centry==code],"subdivision")
    OUT["country_banded"]={m:band(OUT["country"][m],25,None) for m in OUT["country"]}
    OUT["university_banded"]={m:band(OUT["university"][m],25,100) for m in OUT["university"]}
    OUT["map_world_avg"]={}; OUT["map_us_avg"]={}
    d=dfx[dfx.country!="(unknown)"]
    for m in ["scientists","publications","citations"]:
        a=agg(d,"centry",m,pubcol,citecol); OUT["map_world_avg"][m]={str(k).upper():float(a.loc[k,"avg"]) for k in a.index}
        au=agg(dfx[dfx.centry=="usa"],"subdivision",m,pubcol,citecol); OUT["map_us_avg"][m]={str(k):float(au.loc[k,"avg"]) for k in au.index}
    S1=dfx["sm-subfield-1"].astype(str).str.strip(); S2=dfx["sm-subfield-2"].astype(str).str.strip()
    present=[p for p in sorted(set(S1)|set(S2)) if p and p!="nan"]
    tree={}
    for sf in present:
        mask=(S1==sf)|(S2==sf); dom,fld=sf2fd.get(sf, ("(unmapped)","(unmapped)"))
        tree[sf]={ "domain":dom,"field":fld,"total":int(mask.sum()), "avg":round(int(mask.sum())/5,1),
            "quant":sf in QUANT,"target":sf=="Forestry"}
    OUT["tree"]=tree

```

```

OUT["totals"]={m:{str(y):(int((dfx.year==y).sum())) if m=="scientists" else int(dfx[dfx.year==y][pubcol if
m=="publications" else citecol].sum())) for y in YEARS} for m in ["scientists","publications","citations"]}
json.dump(OUT,open(outfile,"w",encoding="utf-8"),default=float)
print(f"[cohort_name]: {int(sum(OUT['totals']['scientists'].values()))} records | countries
{len(OUT['country']['scientists'])} | universities {len(OUT['university']['scientists'])}")
return OUT

build(df, f"{SCR}/forestry_data.json", "Forestry plus Ecology")
build(df[df.group=="Ecology"].copy(), f"{SCR}/forestry_E_data.json", "Ecology (criterion 2)", with_group=False)

# ---- Single-author appendices (B = Forestry group, C = Ecology group): only single-authored publications ----
# scientists = list entries with >=1 single-authored paper (nps>0); publications = sum of nps (single-author
papers);
# citations = sum of ncs (citations to those single-author papers). Same geography/method as the main report.
dfF_sa=df[(df.group=="Forestry") & (df["nps"]>0)].copy()
dfE_sa=df[(df.group=="Ecology") & (df["nps"]>0)].copy()
build(dfF_sa, f"{SCR}/forestry_B_data.json", "Forestry (criterion 1) - single-author publications",
with_group=False, pubcol="nps", citecol="ncs")
build(dfE_sa, f"{SCR}/forestry_C_data.json", "Ecology (criterion 2) - single-author publications",
with_group=False, pubcol="nps", citecol="ncs")

```

61. refine_merged.py (SHA-256 3983d02915f8e50e...; 94 lines)

```

# -*- coding: utf-8 -*-
# Corrected 2-group forestry selection + merged 3-metric tables + multi-root AQ->AM->AO taxonomy.
# Group F (Forestry-rooted): sm-subfield-1 == Forestry
# Group E (Ecology-rooted): sm-subfield-1 == Ecology AND sm-subfield-2 == Forestry
import json, pandas as pd, numpy as np
from cc_defs import CC
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
raw=pd.read_excel(CAREER,sheet_name="Data")
raw["country"]=raw["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")
raw["continent"]=raw["cntry"].map(lambda c: CC[c][1] if c in CC else "(unknown)")
raw["region"]=raw["cntry"].map(lambda c: CC[c][2] if c in CC else "(unknown)")
subdiv=(int(k):v for k,v in json.load(open(f"{SCR}/inst_subdiv.json",encoding="utf-8")).items())
idx=pd.read_csv(f"{SCR}/inst_index.csv"); idx["sd"]=idx["id"].map(subdiv).fillna("Unknown")
look={(r.country,r.inst):r.sd for r in idx.itertuples()}
CN={"usa":"United States","can":"Canada","aus":"Australia"}
raw["subdivision"]=raw.apply(lambda r: look.get((CN.get(r["cntry"]),r["inst_name"]), "Unknown") if r["cntry"] in
CN else None,axis=1)
s1=raw["sm-subfield-1"].astype(str).str.strip(); s2=raw["sm-subfield-2"].astype(str).str.strip()

QUANT={"Applied Mathematics","General Mathematics","Numerical & Computational Mathematics","Statistics &
Probability",
"Artificial Intelligence & Image Processing","Computation Theory & Mathematics","Computer Hardware &
Architecture",
"Distributed Computing","Information Systems","Medical Informatics","Networking & Telecommunications","Software
Engineering",
"Aerospace & Aeronautics","Automobile Design & Engineering","Biomedical Engineering","Chemical
Engineering","Civil Engineering",
"Electrical & Electronic Engineering","Environmental Engineering","Geological & Geomatics Engineering",
"Industrial Engineering & Automation","Mechanical Engineering & Transports","Mining & Metallurgy","Operations
Research","Bioinformatics",
"Silviculture","Biometrics","Mensuration","Forest Management","Remote Sensing","GIS","Programming"}
raw["orient"]=np.where(s1.isin(QUANT)|s2.isin(QUANT),"Quantitative-oriented","Qualitative-oriented")

# --- the two selection criteria (both require Forestry) ---
gF_mask=(s1=="Forestry") # criterion 1: AM=Forestry -> "Forestry"
gE_mask=(s1=="Ecology") & (s2=="Forestry") # criterion 2: AM=Ecology AND AO=Forestry -> "Ecology"
forestry_mask=gF_mask|gE_mask # 1 + 2 = "Forestry plus Ecology"
GF="Forestry"; GE="Ecology"
raw["group"]=np.where(gF_mask,GF,np.where(gE_mask,GE,"(other)"))

def years_arr(piv,k): return [int(piv.loc[k,y]) if (k in piv.index and y in piv.columns) else 0 for y in YEARS]
def dim_members(frame,keycol,with_country=False,order=None):
ps=frame.pivot_table(index=keycol,columns="year",values="authfull",aggfunc="count",fill_value=0)
pp=frame.pivot_table(index=keycol,columns="year",values="np",aggfunc="sum",fill_value=0)
pc=frame.pivot_table(index=keycol,columns="year",values="nc",aggfunc="sum",fill_value=0)
ctry=frame.groupby(keycol)["country"].agg(lambda s:s.value_counts().index[0]) if with_country else None
mem=[]
for k in ps.index:
mem.append({"key":k,"country":(ctry.get(k,"") if with_country else None),
"s":years_arr(ps,k),"p":years_arr(pp,k),"c":years_arr(pc,k)})
if order: mem.sort(key=lambda m: order.index(m["key"]) if m["key"] in order else 99)

```

```

        else: mem.sort(key=lambda m:-sum(m["s"]))
        return mem
def band(mem,n_ind=25, cap=None, frac=0.10):
    # straight top-N ranking (cumulative bands dropped per user request)
    return {"head":mem[:n_ind], "bands":[]}

def cohort(mask,label):
    df=raw[mask].copy()
    out={"n":int(len(df)),
        "totals":{m:{str(y):(int((df.year==y).sum()) if m=="scientists" else int(df[df.year==y]["np" if
m=="publications" else "nc"].sum())) for y in YEARS} for m in ["scientists","publications","citations"]}}
    for dim,col in [("continent","continent"), ("region","region"), ("country","country")]:
        out[dim]=dim_members(df[df[col]!="(unknown)"],col)
    out["us_state"]=dim_members(df[df.cntry=="usa"],"subdivision")
    out["orient"]=dim_members(df,"orient")
    if label=="forestry": out["group"]=dim_members(df,"group",order=[GF,GE])
    out["country_banded"]=band(out["country"],25,None)
    uni=dim_members(df[df["inst_name"].notna()], "inst_name",with_country=True)
    out["university"]=uni; out["university_banded"]=band(uni,25,100)
    out["ca_prov"]=dim_members(df[df.cntry=="can"],"subdivision")
    out["au_state"]=dim_members(df[df.cntry=="aus"],"subdivision")
    return out

DATA={"years":YEARS,
    "forestry":cohort(forestry_mask,"forestry"), # "Forestry plus Ecology" = 1+2 (the whole cohort)
    "forestry_F":cohort(gF_mask,"forestry_F"), # "Forestry" = criterion 1
    "forestry_E":cohort(gE_mask,"forestry_E")} # "Ecology" = criterion 2

# ---- multi-root taxonomy: AQ (sm-field) -> AM (sm-subfield-1) -> AO (sm-subfield-2) ----
cdf=raw[forestry_mask]
roots={} # {AQ: {AM: {AO: count}}}
group_of_field={}
for aq,am,ao,grp in zip(cdf["sm-field"].astype(str).str.strip(),
    cdf["sm-subfield-1"].astype(str).str.strip(),
    cdf["sm-subfield-2"].astype(str).str.strip(),
    cdf["group"]):
    ao=ao if ao and ao!="nan" else "(no secondary)"
    roots.setdefault(aq,{}).setdefault(am,{}).setdefault(ao,0); roots[aq][am][ao]+=1
    group_of_field[aq]=grp
DATA["taxonomy"]={"quant_set":sorted(QUANT),"n_total":int(len(cdf)),
    "roots":roots,"group_of_field":group_of_field}

json.dump(DATA,open(f"{SCR}/merged_data.json","w",encoding="utf-8"),default=int)
MF=DATA["forestry"]
print("Forestry+Ecology n:",MF["n"],"| Forestry (1):",DATA["forestry_F"]["n"],"| Ecology (2):",DATA["forestry_E"]
["n"])
print("group per-year totals:",{g["key"]:sum(g["s"]) for g in MF["group"]})
print("taxonomy roots (AQ):",{aq:sum(sum(a.values()) for a in roots[aq].values()) for aq in roots})
print("countries:",len(MF["country"]),"| universities:",len(MF["university"]))

```

62. render_appx.py (SHA-256 82492bae4a1ef9b0...; 17 lines)

```

# -*- coding: utf-8 -*-
import fitz
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
want={"Appendix B. The Forestry Group":None,"Table B3":None,"Appendix C. The Ecology Group":None,"Table C3":None}
for i,p in enumerate(doc):
    t=p.get_text()
    for k in want:
        if want[k] is None and k in t: want[k]=i
print("pages:",want)
out=[]
for k,pg in want.items():
    if pg is None: continue
    pix=doc[pg].get_pixmap(dpi=105)
    fn=f"{SCR}/_apx_{k.split('.')[0]}.replace(' ','').png"
    pix.save(fn); out.append(fn)
print("\n".join(out))

```

63. render_inst.py (SHA-256 82a7f2fa12bdd831...; 14 lines)

```

# -*- coding: utf-8 -*-
import fitz
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")

```

```

SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
want={"Table B4":None,"Table C4":None}
for i,p in enumerate(doc):
    t=p.get_text()
    for k in want:
        if want[k] is None and k in t: want[k]=i
print("pages:",want)
for k,pg in want.items():
    if pg is None: continue
    doc[pg].get_pixmap(dpi=105).save(f"{SCR}/_apx_{k.replace(' ','')}.png")
    print(f"{SCR}/_apx_{k.replace(' ','')}.png")

```

64. render_tabs.py (SHA-256 51e2312d67c9a811...; 30 lines)

```

# -*- coding: utf-8 -*-
import fitz
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
# find pages containing Table 6a and Table 7a
want={"Table 6a":None,"Table 7a":None}
for i,p in enumerate(doc):
    t=p.get_text()
    for k in want:
        if want[k] is None and k in t: want[k]=i
print("pages:",want)
# render those pages plus the next one
out=[]
for k,pg in want.items():
    if pg is None: continue
    for j in (pg,pg+1):
        if j<doc.page_count:
            pix=doc[j].get_pixmap(dpi=110)
            fn=f"{SCR}/_chk_{k.replace(' ','')}_{j}.png"
            pix.save(fn); out.append(fn)
print("\n".join(out))
# also count rows: for each split table caption, count subsequent numeric-first-token lines until next 'Table' or 'Figure'
import re
txt="\n".join(p.get_text() for p in doc)
for cap in ["Table 6a","Table 6b","Table 6c","Table 7a","Table 7b","Table 7c"]:
    seg=txt.split(cap,1)[1]
    seg=re.split(r"(Table \d|Figure \d)",seg,1)[0]
    # count lines that look like a data row label (non-empty, not the header row)
    rows=[l for l in seg.splitlines() if l.strip() and l.strip() not in
("2020","2021","2022","2023","2024","Avg/yr")]
    print(cap,"approx content lines:",len(rows))

```

65. report_data.py (SHA-256 fc91ec1b21ce905a...; 69 lines)

```

import json, pandas as pd
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS = [2020,2021,2022,2023,2024]

rankings = json.load(open(f"{SCR}/rankings.json", encoding="utf-8"))
subdiv = {int(k): v for k, v in json.load(open(f"{SCR}/inst_subdiv.json", encoding="utf-8")).items()}
idx = pd.read_csv(f"{SCR}/inst_index.csv")
idx["sd"] = idx["id"].map(subdiv).fillna("Unknown")
look = {(r.country, r.inst): r.sd for r in idx.itertuples()}

df = pd.read_excel(CAREER, sheet_name="Data")
CN = {"usa":"United States","can":"Canada","aus":"Australia"}
df["subdivision"] = df.apply(lambda r: look.get((CN.get(r["cntry"]), r["inst_name"]), "Unknown") if r["cntry"] in CN else None, axis=1)
def field_of(m):
    m=str(m); f="Forestry" in m; e="Ecology" in m
    return "Both" if (f and e) else ("Forestry" if f else "Ecology")
df["field"] = df["matched_subfield"].map(field_of)

out = {"rankings": rankings}

def tab(sub, col):
    t = sub.pivot_table(index=col, columns="year", values="authfull", aggfunc="count", fill_value=0)

```

```

for y in YEARS:
    if y not in t.columns: t[y]=0
    t = t[YEARS]; t["total"]=t.sum(axis=1)
    return t.sort_values(2024, ascending=False).reset_index().to_dict(orient="records")

out["us_state"] = tab(df[df.cntry=="usa"], "subdivision")
out["ca_prov"] = tab(df[df.cntry=="can"], "subdivision")
out["au_state"] = tab(df[df.cntry=="aus"], "subdivision")

# distinct scientists (dedup by author name within country group) for narrative
out["unique_authors_allyears"] = int(df["authfull"].nunique())
out["persistent_all5"] = int((df.groupby("authfull")["year"].nunique()==5).sum())

# growth 2020->2024 for top countries
c = pd.DataFrame(rankings["country"])
c = c.set_index("country")
growth = {}
for name in c.index:
    a, b = c.loc[name, "2020"] if "2020" in c.columns else c.loc[name, 2020], c.loc[name, 2024] if 2024 in
c.columns else c.loc[name, "2024"]
    if a and a >= 20:
        growth[name] = round((b-a)/a*100, 1)
out["country_growth_pct"] = dict(sorted(growth.items(), key=lambda kv: -kv[1]))

# world map data: 2024 count per country (ISO3 upper)
d24 = df[df.year==2024]
cnt = d24.groupby("cntry").size()
out["map_2024_iso3"] = {k.upper(): int(v) for k, v in cnt.items()}

# US map data
us24 = d24[d24.cntry=="usa"]
out["map_2024_us_state"] = us24["subdivision"].value_counts().to_dict()

# field by year
out["field_year"] = tab(df, "field")

# per-capita style: entries per country 2024 already there. Also field split of top countries 2024
fc = d24.pivot_table(index="cntry", columns="field", values="authfull", aggfunc="count", fill_value=0)
fc["Total"] = fc.sum(axis=1)
fc = fc.sort_values("Total", ascending=False).head(12)
out["field_by_country_2024"] = fc.reset_index().to_dict(orient="records")

json.dump(out, open(f"{SCR}/report_data.json", "w", encoding="utf-8"), indent=1, default=int)
print("unique authors:", out["unique_authors_allyears"], "| in all 5 releases:", out["persistent_all5"])
print("top growth (>=20 in 2020):", list(out["country_growth_pct"].items())[0:6])
print("bottom growth:", list(out["country_growth_pct"].items())[-3:])
print("saved report_data.json")

```

66. rootcause.py (SHA-256 8b32eeca665b14e7...; 17 lines)

```

import json
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford"
YEARS=["2020","2021","2022","2023","2024"]
def load(n):
    with open(f"{SCR}\{n}", encoding="utf-8") as f: return json.load(f)
B=load("forestry_B_data.json")

# Sum the JSON country dimension itself per year, compare to totals
for dim in ["country","university","continent"]:
    print(f"\n--- B dimension '{dim}' summed over all records vs totals ---")
    for met in ["scientists","publications","citations"]:
        recs=B[dim][met]
        s={y:sum(int(r.get(y,0) or 0) for r in recs) for y in YEARS}
        tot={y:B["totals"][met][y] for y in YEARS}
        diff={y:(s[y],tot[y]) for y in YEARS if s[y]!=tot[y]}
        print(f"    {met}: dimsum={s}")
        print(f"    totals={tot} {'MATCH' if not diff else 'DIFF '+str(diff)}")

```

67. source_check.py (SHA-256 f5b96db5f069eb67...; 31 lines)

```

import pandas as pd, sys
sys.path.insert(0, r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-
a33f-6af155a73519\scratchpad\stanford")

```

```

XL = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df = pd.read_excel(XL, sheet_name="Data")
print("cols:", list(df.columns))
# Forestry group criterion 1
f = df[df["sm-subfield-1"]=="Forestry"].copy()
# single author only
f = f[f["nps"]>0].copy()
print("Forestry-SA record count:", len(f))

for y in [2021,2023]:
    fy = f[f["year"]==y]
    print(f"\nYEAR {y}: total SA scientists={len(fy)}")
    # rows with missing/blank country
    badc = fy[fy["cntry"].isna() | (fy["cntry"].astype(str).str.strip()=="")]
    print(f" rows with blank cntry: {len(badc)}")
    # rows with missing institution
    badi = fy[fy["inst_name"].isna() | (fy["inst_name"].astype(str).str.strip()=="")]
    print(f" rows with blank inst_name: {len(badi)}")
    # Also check if country code maps in cc_defs
    try:
        from cc_defs import CC
        unmapped = fy[~fy["cntry"].isin(CC.keys()) & fy["cntry"].notna()]
        print(f" rows with cntry code NOT in CC map: {len(unmapped)}")
        if len(unmapped)>0:
            print(" codes:", sorted(set(unmapped['cntry'].astype(str))))
        if len(badc)>0:
            print(" blank-country authors:", badc[['authfull','inst_name','np','nps','ncs']].values.tolist())
    except Exception as e:
        print(" cc_defs import error:", e)

```

68. split_batches.py (SHA-256 ff544db568d7eb32...; 21 lines)

```

import pandas as pd, json, os
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\ea9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
BATCHDIR = os.path.join(SCR, "batches")
os.makedirs(BATCHDIR, exist_ok=True)
idx = pd.read_csv(f"{SCR}/inst_index.csv")
B = 85
batches = []
for i in range(0, len(idx), B):
    part = idx.iloc[i:i+B]
    fn = os.path.join(BATCHDIR, f"batch_{i//B:02d}.csv")
    part.to_csv(fn, index=False, encoding="utf-8")
    batches.append(fn)
# verify list: top institutions by scientist-rows per country (drive most of the ranking)
vlist = []
for country, k in [("United States",30), ("Canada",12), ("Australia",12)]:
    sub = idx[idx.country==country].nlargest(k, "n")
    vlist += sub[["id","country","inst","n"]].to_dict(orient="records")
manifest = {"dir": SCR, "batches": batches, "verifyList": vlist}
with open(f"{SCR}/manifest.json","w",encoding="utf-8") as f:
    json.dump(manifest, f)
print(json.dumps({"n_batches": len(batches), "n_verify": len(vlist), "manifest": f"{SCR}/manifest.json"}))

```

69. uga_all.py (SHA-256 b989c47ccd0b1954...; 19 lines)

```

# -*- coding: utf-8 -*-
# ALL University of Georgia scientists on the Stanford list (2024 career release), with subfields.
# Goal: find Warnell-School people classified under NON-Forestry subfields.
import pandas as pd, re, json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\ea9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
d=pd.read_excel(f"{SCR}/career_2024.xlsx",sheet_name="Data")
npc=[c for c in d.columns if re.fullmatch(r"np\d+",str(c))][0]
ncc=[c for c in d.columns if re.fullmatch(r"nc\d+",str(c)) and "(ns)" not in str(c)][0]
uga=d[d["inst_name"].astype(str).str.fullmatch(r"University of Georgia",na=False)].copy()
print(f"UGA (2024) total on list: {len(uga)}")
uga=uga.sort_values("rank sm-subfield-1")
cols=["authfull","sm-subfield-1","sm-subfield-2","sm-field",npc,ncc]
out=uga[cols].rename(columns={npc:"np",ncc:"nc"})
out.to_csv(f"{SCR}/uga_all_2024.csv",index=False)
print("\n=== all UGA scientists (2024), sorted by subfield ===")
with
pd.option_context("display.max_rows",None,"display.max_columns",None,"display.width",260,"display.max_colwidth",40):
    print(out.to_string(index=False))

```

```
print("\n=== subfield tally ===")
print(out["sm-subfield-1"].value_counts().to_string())
```

70. uga_match.py (SHA-256 c306cd2dea3acb81...; 29 lines)

```
# -*- coding: utf-8 -*-
# Match the 280 UGA Stanford-list scientists against the Warnell faculty roster (last name + first initial).
import pandas as pd, re
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudio\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
uga=pd.read_csv(f"{SCR}/uga_all_2024.csv")
warnell_raw="""Abney Rebecca|Abrams Jesse|Albanese Brett|Aubrey Doug|Bailey Robert L.|Bargeron Chuck|Barnes Brittany|Beasley James|Beckwith Julian|Bettinger Pete|Bolding Chad|Boley Bynum|Bongarten Bruce|Borders Bruce|Boring Lindsay|Bothwell Helen|Brantley Steven|Brawner Jeremy|Bringolf Robert|Bullock Bronson|Carroll John|Castleberry Steven B.|Chamberlain Michael|Chandler Richard|Cherry Michael|Clabo David|Coder Kim|Conner Mike|Conrad Joe|Conroy Michael|Cook Walter|Cooper Robert|Cordell Ken|Covert Sarah F.|Crawford Daniel|D'Angelo Gino|Dahlen Joe|Daniels Richard|DeVault Travis|Dickens David|Drummond Daniel|Elkins Duncan|Evaristo Jaivime|Ford Miniat Chelcy|Fox Adam|Fuhrman Nick|Gandhi Kamal|Ghimire Ramesh|Gladfelter Heather|Gordon Jason|Green Gary|Greene Dale|Hamel Martin|Harris Tom|Hazelton Peter|Hepinstall-Cymerman Jeff|Hernandez Sonia M.|Hitchner Sarah|Hoopes Lisa|Hubbard William|Irwin Brian|Izlar Robert|Jackson Rhett|Jenkins Christopher|Jennings Cecil|Johnson Dan|Kane Michael|Kilgo John|Kim Caroline|King Elizabeth|Klepzig Kier|Kluever Bryan Michael|Kohl Michel|Korhonen Jaana|Li Yanshu|Loudermilk Eva Louise|Lucardi Rima Dee|Lutz Jack|Maerz John|Marchinton Larry|Markewitz Daniel|Martin James|McCarty Elizabeth|McKay Steven Kyle|Mctague John|Mendell Brooks|Merkle Scott|Merry Krista|Miller Karl|Moore Clinton|Moorhead David|Naha Dipanjan|Nibbelink Nate|Nutter Wade|O'Brien Joseph|Peterson James|Petty Jeffrey Todd|Pienaar Elizabeth|Ponder Skeet|Poudyal Neelam|Rhodes Olin Jr.|Rushing Clark|Schelhas John|Schimleck Laurence|Shamblin Brian|Shelton James|Shiver Barry|Simon Troy|Siry Jacek|Smith Lora|Spear Stephen|Sutter Lori|Tarrant Michael|Teitelbaum Claire|Terhune Theron Menees|Teskey Robert|Tsai C.J.|Tuberville Tracey|Villari Caterina|Vogt James|Warren Bob|White Eric Michael|Wigley Thomas|Wilde Susan|Woosnam Kyle|Yabsley Michael|Yang Sheng-I|Zhao Dehai|Zhou Ran"""
# also known EMERITUS Warnell not in the current directory
emeritus="Grossman Gary D.|Helfman Gene S.|Clutter Michael|Harris Larry"
def keyset(txt):
    s=set()
    for n in txt.split("|"):
        p=n.strip().split()
        if len(p)>=2: s.add((p[0].lower(), p[1][0].lower()))
    return s
W=keyset(warnell_raw)|keyset(emeritus)
def ugakey(a):
    m=re.match(r"([^\,]+),\s*(\w)",str(a)); return (m.group(1).strip().lower(), m.group(2).lower()) if m else (None,None)
uga["k"]=uga["authfull"].map(ugakey)
uga["warnell"]=uga["k"].map(lambda k: k in W)
hit=uga[uga["warnell"]].copy()
hit["is_forestry"]=(hit["sm-subfield-1"]=="Forestry")|((hit["sm-subfield-1"]=="Ecology")&(hit["sm-subfield-2"]=="Forestry"))
print(f"Candidate Warnell matches among the 280 UGA list: {len(hit)}")
print("\n=== NON-Forestry-classified candidates (the 'hidden Warnell') ===")
nf=hit[~hit["is_forestry"]].sort_values("nc",ascending=False)
for _,r in nf.iterrows():
    print(f" {r['authfull'][:<26]} {r['sm-subfield-1'][:<32]} {r['sm-subfield-2'][:<24]} np={r['np']:>4} nc={r['nc']:>6}")
print("\n=== Forestry-classified (already in cohort) ===")
for _,r in hit[hit["is_forestry"]].iterrows():
    print(f" {r['authfull'][:<26]} {r['sm-subfield-1']} / {r['sm-subfield-2']}")
```

71. validate_final.py (SHA-256 2aa660720939124c...; 76 lines)

```
import json
from openpyxl import load_workbook

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claudio\C--D-Dropbox-ClaudeCodeTest\9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
YEARS = ["2020","2021","2022","2023","2024"]

def load(name):
    with open(f"{SCR}\\{name}", encoding="utf-8") as f:
        return json.load(f)
B = load("forestry_B_data.json"); C = load("forestry_C_data.json")
def jt(D):
    t=D["totals"]; return {m:{y:t[m][y] for y in YEARS} for m in ["scientists","publications","citations"]}
jB=jt(B); jC=jt(C)

wb = load_workbook(WB, data_only=True)
```

```

BLOCK_METRIC = {"By SCIENTISTS":"scientists","By SINGLE-AUTHOR PAPERS":"publications","By CITES TO SA
PAPERS":"citations"}

# (json totals, dimension, expected count, year-start-col, keycol)
def dimcount(D,dim): return len(D[dim]["scientists"])
CFG = {
    "SA-B Forestry countries": (jB,"country", dimcount(B,"country"), 2, 1),
    "SA-B Forestry institutions": (jB,"university", dimcount(B,"university"),3, 1), # keycol=Institution(1),
Country=2, years 3-7
    "SA-C Ecology countries": (jC,"country", dimcount(C,"country"), 2, 1),
    "SA-C Ecology institutions": (jC,"university", dimcount(C,"university"),3, 1),
}

def is_int(v):
    if isinstance(v,bool): return False
    if isinstance(v,int): return True
    if isinstance(v,float): return v==int(v)
    return False

findings=[]
for sheet, (jt,dim,expcount,ystart,keycol) in CFG.items():
    ws=wb[sheet]
    print(f"\n=== {sheet} (years start col {ystart}) ===")
    blocks=[]
    for r in range(1,ws.max_row+1):
        a=ws.cell(row=r,column=1).value
        if isinstance(a,str):
            for lab,met in BLOCK_METRIC.items():
                if lab in a: blocks.append((r,lab,met))
    if len(blocks)!=3:
        findings.append(f"[{sheet}] expected 3 blocks found {len(blocks)}")
    brows=[b[0] for b in blocks]
    for i,(hr,lab,met) in enumerate(blocks):
        end=brows[i+1] if i+1<len(brows) else ws.max_row+1
        colsum={y:0 for y in YEARS}; nrows=0; nonint=0; firstbad=None
        for r in range(hr+2,end):
            k=ws.cell(row=r,column=keycol).value
            if k is None or (isinstance(k,str) and k.strip()==""): continue
            nrows+=1
            for off,y in enumerate(YEARS):
                v=ws.cell(row=r,column=ystart+off).value
                if v is None: v=0
                if not is_int(v):
                    nonint+=1
                    if firstbad is None: firstbad=(r,y,v)
                else: colsum[y]+=int(v)
        exp=jt[met]
        mism={y:(colsum[y],exp[y]) for y in YEARS if colsum[y]!=exp[y]}
        print(f" {lab}->{met}: rows={nrows} sums={colsum} {'OK' if not mism else 'MISMATCH '+str(mism)}")
        if mism: findings.append(f"[{sheet}] '{lab}' ({met}) colsum mismatch (got,exp): {mism}")
        if nonint: findings.append(f"[{sheet}] '{lab}' {nonint} non-int, first={firstbad}")
        if met=="scientists" and dim=="country" and nrows!=expcount:
            findings.append(f"[{sheet}] scientists rows {nrows} != JSON #countries {expcount}")
        if met=="scientists":
            print(f" (rows {nrows} vs JSON {dim} count {expcount})")

print("\n"+"#"*60)
if findings:
    print("FINDINGS:")
    for f in findings: print(" -",f)
else:
    print("NO FINDINGS - ALL CHECKS PASS")

```

72. validate_full.py (SHA-256 842f214ea7119cdf...; 117 lines)

```

import json
from openpyxl import load_workbook

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
YEARS = ["2020","2021","2022","2023","2024"]

def load(name):
    with open(f"{SCR}\\{name}", encoding="utf-8") as f:
        return json.load(f)
B = load("forestry_B_data.json")

```

```

C = load("forestry_C_data.json")

def jtotals(D):
    t = D["totals"]
    return {m: {y: t[m][y] for y in YEARS} for m in ["scientists", "publications", "citations"]}
jB = jtotals(B); jC = jtotals(C)
ncountriesB = len(B["country"]["scientists"]); ncountriesC = len(C["country"]["scientists"])
nunivB = len(B["university"]["scientists"]); nunivC = len(C["university"]["scientists"])

wb = load_workbook(WB, data_only=True)

# expected block header labels (metric name mapping to JSON totals metric)
BLOCK_METRIC = {
    "By SCIENTISTS": "scientists",
    "By SINGLE-AUTHOR PAPERS": "publications",
    "By CITES TO SA PAPERS": "citations",
}

# map sheet -> (json obj, totals dict, dimension expected counts)
SHEET_CFG = {
    "SA-B Forestry countries": (jB, "country", ncountriesB),
    "SA-B Forestry institutions": (jB, "university", nunivB),
    "SA-C Ecology countries": (jC, "country", ncountriesC),
    "SA-C Ecology institutions": (jC, "university", nunivC),
}

findings = []

def is_int(v):
    if isinstance(v, bool): return False
    if isinstance(v, int): return True
    if isinstance(v, float): return v == int(v)
    return False

for sheet, (jtot, dim, expected_count) in SHEET_CFG.items():
    ws = wb[sheet]
    print("\n" + "="*72)
    print(f"SHEET: {sheet} max_row={ws.max_row} max_col={ws.max_column}")
    # non-empty?
    nonempty = ws.max_row > 1 and ws.max_column >= 1
    # find block header rows in col A
    blocks = [] # (header_row, label, metric)
    for r in range(1, ws.max_row+1):
        a = ws.cell(row=r, column=1).value
        if a and isinstance(a, str):
            for label, metric in BLOCK_METRIC.items():
                if label in a:
                    blocks.append((r, label, metric))
    print("Blocks found:", [(r,l) for r,l,m in blocks])
    if len(blocks) != 3:
        findings.append(f"[{sheet}] expected 3 stacked blocks, found {len(blocks)}: {[l for _,l,_ in blocks]}")

# for each block, parse header row (should be next row) and data rows until blank/next block
block_rows = [b[0] for b in blocks]
for idx, (hr, label, metric) in enumerate(blocks):
    # column-header row expected at hr+1
    hdr = [ws.cell(row=hr+1, column=c).value for c in range(1,9)]
    exp_hdr = [None, "2020", "2021", "2022", "2023", "2024", "Total", "Avg/yr"]
    # col1 label may be 'Country'/'Institution' etc - check cols 2..8
    hdr_ok = all(str(hdr[i]).strip() == exp_hdr[i] for i in range(1,8))
    if not hdr_ok:
        findings.append(f"[{sheet}] block '{label}' header row mismatch: got {hdr[1:8]}")
    # data starts hr+2, ends before next block header (or max_row)
    end = block_rows[idx+1] if idx+1 < len(block_rows) else ws.max_row+1
    # collect data rows (col A non-empty, not header)
    colsum = {y:0 for y in YEARS}
    nrows = 0
    noninteger = 0
    firstbad = None
    for r in range(hr+2, end):
        a = ws.cell(row=r, column=1).value
        if a is None or (isinstance(a, str) and a.strip() == ""):
            continue
        nrows += 1
        for ci, y in enumerate(YEARS, start=2):
            v = ws.cell(row=r, column=ci).value
            if v is None: v = 0

```

```

        if not is_int(v):
            noninteger += 1
            if firstbad is None: firstbad=(r,y,v)
        else:
            colsum[y] += int(v)
        # also check Total col is int
        tv = ws.cell(row=r, column=7).value
        if tv is not None and not is_int(tv):
            noninteger += 1
            if firstbad is None: firstbad=(r,"Total",tv)
    # compare colsum to json totals
    exp = jtots[metric]
    mism = {y:(colsum[y],exp[y]) for y in YEARS if colsum[y]!=exp[y]}
    status = "OK" if not mism else "MISMATCH"
    print(f"  block '{label}' -> metric {metric}: rows={nrows} colsums={colsum} {status}")
    if mism:
        findings.append(f"[{sheet}] block '{label}' ({metric}) column-sum mismatch (got,expected): {mism}")
    if noninteger:
        findings.append(f"[{sheet}] block '{label}' has {noninteger} non-integer value(s), first={firstbad}")
    # row-count check for scientists block on countries sheets
    if metric=="scientists":
        print(f"    scientists block rows={nrows}, JSON dim '{dim}' count={expected_count}")
        if dim=="country" and nrows != expected_count:
            findings.append(f"[{sheet}] scientists block row-count {nrows} != JSON #countries {expected_count}")

print("\n" + "#"*72)
print("FINDINGS:" if findings else "NO FINDINGS - ALL CHECKS PASS")
for f in findings:
    print(" -", f)

```

73. validate_wb.py (SHA-256 eab9d025ba6ec076...; 51 lines)

```

import json, sys
from openpyxl import load_workbook

SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claudes\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
YEARS = ["2020","2021","2022","2023","2024"]

def load(name):
    with open(f"{SCR}\\{name}", encoding="utf-8") as f:
        return json.load(f)

B = load("forestry_B_data.json")
C = load("forestry_C_data.json")

def jtots(D):
    t = D["totals"]
    out = {}
    for metric in ["scientists","publications","citations"]:
        out[metric] = {y: t[metric][y] for y in YEARS}
    return out

jB = jtots(B)
jC = jtots(C)

# number of countries in JSON (scientists dimension)
def ncountries(D):
    return len(D["country"]["scientists"])
def nuniv(D):
    return len(D["university"]["scientists"])

print("=== JSON TOTALS ===")
print("B scientists", jB["scientists"])
print("B publications", jB["publications"])
print("B citations", jB["citations"])
print("C scientists", jC["scientists"])
print("C publications", jC["publications"])
print("C citations", jC["citations"])
print("B #countries(scientists dim):", ncountries(B), " #univ:", nuniv(B))
print("C #countries(scientists dim):", ncountries(C), " #univ:", nuniv(C))

wb = load_workbook(WB, data_only=True)

```

```

print("\n=== ALL SHEETS ===")
for ws in wb.sheetnames:
    print(repr(ws))

SHEETS = ["SA-B Forestry countries", "SA-B Forestry institutions",
          "SA-C Ecology countries", "SA-C Ecology institutions"]

print("\n=== SHEET EXISTENCE ===")
for s in SHEETS:
    print(s, "EXISTS" if s in wb.sheetnames else "MISSING")

```

74. verify.py (SHA-256 4dfe52a6b95b3a0b...; 107 lines)

```

import openpyxl
from collections import Counter

def col_idx(ws):
    hdr = [str(c).strip() for c in next(ws.iter_rows(max_row=1, values_only=True))]
    return hdr, {name: hdr.index(name) for name in ("sm-subfield-1", "sm-subfield-2")}

def scan_source(path, targets):
    """Independent re-scan: return per-year style Counter of matches and a set of author keys."""
    wb = openpyxl.load_workbook(path, read_only=True)
    ws = wb["Data"]
    hdr, idx = col_idx(ws)
    i1, i2 = idx["sm-subfield-1"], idx["sm-subfield-2"]
    keys, cat = set(), Counter()
    n = 0
    for row in ws.iter_rows(min_row=2, values_only=True):
        if row is None or all(v is None for v in row):
            continue
        n += 1
        vals = {str(row[i1]).strip().lower() if row[i1] else "",
                str(row[i2]).strip().lower() if row[i2] else ""}
        m = vals & targets
        if m:
            keys.add((row[0], row[1])) # authfull, inst_name
            for t in m:
                cat[t] += 1
    wb.close()
    return n, keys, cat

def read_agg(path):
    wb = openpyxl.load_workbook(path, read_only=True)
    ws = wb["Data"] if "Data" in wb.sheetnames else wb[wb.sheetnames[0]]
    hdr = [str(c).strip() for c in next(ws.iter_rows(max_row=1, values_only=True))]
    yi = hdr.index("year")
    mi = hdr.index("matched_subfield") if "matched_subfield" in hdr else None
    sli = hdr.index("sm-subfield-1"); s2i = hdr.index("sm-subfield-2")
    rows = []
    for row in ws.iter_rows(min_row=2, values_only=True):
        if row is None or all(v is None for v in row):
            continue
        rows.append(row)
    wb.close()
    return hdr, rows, yi, mi, sli, s2i

print("==== CAREER (Forestry + Ecology) =====")
targets = {"forestry", "ecology"}
CAREER = {
    "2020": "career_2020.xlsx", "2021": "career_2021.xlsx", "2022": "career_2022.xlsx",
    "2023": "career_2023.xlsx", "2024": "career_2024.xlsx",
}
src_keys_by_year, src_counts = {}, {}
for y, p in CAREER.items():
    n, keys, cat = scan_source(p, targets)
    src_keys_by_year[y] = keys
    src_counts[y] = (n, len(keys), dict(cat))
    print(f"  source {y}: scanned={n}, matched_rows={len(keys)}, by_subfield={dict(cat)}")

hdr, rows, yi, mi, sli, s2i = read_agg(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx")
agg_by_year = Counter(r[yi] for r in rows)
print(f"  aggregate total rows: {len(rows)}  cols: {len(hdr)}")
print(f"  aggregate by year: {dict(sorted(agg_by_year.items()))}")

```

```

bad_label, bad_match = 0, 0
for r in rows:
    s1 = str(r[s1i]).strip().lower() if r[s1i] else ""
    s2 = str(r[s2i]).strip().lower() if r[s2i] else ""
    if not ({s1, s2} & targets):
        bad_match += 1
    lab = str(r[mi]).lower()
    labset = {x.strip() for x in lab.replace(";", " ").split()}
    present = {t for t in targets if t in {s1, s2}}
    if not (labset & {t for t in present}):
        bad_label += 1
print(f" rows where neither subfield is a target (should be 0): {bad_match}")
print(f" rows with wrong/missing matched_subfield label (should be 0): {bad_label}")
ok = all(agg_by_year[int(y)] == src_counts[y][1] for y in CAREER)
print(f" per-year aggregate == independent source count: {ok}")
for y in CAREER:
    a, b = agg_by_year[int(y)], src_counts[y][1]
    if a != b:
        print(f" MISMATCH {y}: aggregate={a} source={b}")

print("\n===== SINGLEYR (Forestry only) =====")
tS = {"forestry"}
SINGLE = {
    "2020": "singleyr_2020.xlsx", "2021": "singleyr_2021.xlsx", "2022": "singleyr_2022.xlsx", "2023": "singleyr_2023.xlsx",
    "2024": "singleyr_2024.xlsx"
}
s_counts = {}
for y, p in SINGLE.items():
    n, keys, cat = scan_source(p, tS)
    s_counts[y] = len(keys)
    print(f" source {y}: scanned={n}, forestry_rows={len(keys)}")
wb = openpyxl.load_workbook(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_singleyr_2020-2024.xlsx",
read_only=True)
ws = wb["Forestry"]
hdr2 = [str(c).strip() for c in next(ws.iter_rows(max_row=1, values_only=True))]
yi2 = hdr2.index("year"); slb = hdr2.index("sm-subfield-1"); s2b = hdr2.index("sm-subfield-2")
srows = [r for r in ws.iter_rows(min_row=2, values_only=True) if r and not all(v is None for v in r)]
wb.close()
sby = Counter(r[yi2] for r in srows)
badS = sum(1 for r in srows if "forestry" not in {str(r[slb]).strip().lower() if r[slb] else "",
str(r[s2b]).strip().lower() if r[s2b] else ""})
print(f" aggregate total rows: {len(srows)} by year: {dict(sorted(sby.items()))}")
print(f" rows where neither subfield is Forestry (should be 0): {badS}")
okS = all(sby[int(y)] == s_counts[y] for y in SINGLE)
print(f" per-year aggregate == independent source count: {okS}")
for y in SINGLE:
    if sby[int(y)] != s_counts[y]:
        print(f" MISMATCH {y}: aggregate={sby[int(y)]} source={s_counts[y]}")

print("\nDONE")

```

75. verify2.py (SHA-256 43c17b2b95cdf8a...; 35 lines)

```

import openpyxl
from collections import Counter

def scan(path, targets):
    wb = openpyxl.load_workbook(path, read_only=True)
    ws = wb["Data"]
    hdr = [str(c).strip() for c in next(ws.iter_rows(max_row=1, values_only=True))]
    i1, i2 = hdr.index("sm-subfield-1"), hdr.index("sm-subfield-2")
    rowcount = 0
    keyc = Counter()
    for row in ws.iter_rows(min_row=2, values_only=True):
        if row is None or all(v is None for v in row):
            continue
        vals = {str(row[i1]).strip().lower() if row[i1] else "",
                str(row[i2]).strip().lower() if row[i2] else ""}
        if vals & targets:
            rowcount += 1
            keyc[(row[0], row[1])] += 1
    wb.close()
    dups = {k: v for k, v in keyc.items() if v > 1}
    return rowcount, dups

print("== career (forestry+ecology): matched ROW counts and duplicate (author,inst) pairs ==")
for y in ("2020", "2021", "2022", "2023", "2024"):
    rc, dups = scan(f"career_{y}.xlsx", {"forestry", "ecology"})

```

```

print(f"{y}: matched_rows={rc}, duplicate_pairs={len(dups)}")
for k, v in list(dups.items())[:10]:
    print(f"    {v}x {k}")

print("== singleyr (forestry) ==")
for y in ("2022", "2024"):
    rc, dups = scan(f"singleyr_{y}.xlsx", {"forestry"})
    print(f"{y}: matched_rows={rc}, duplicate_pairs={len(dups)}")
    for k, v in list(dups.items())[:10]:
        print(f"    {v}x {k}")

```

76. verify_fields.py (SHA-256 b1a45bb9abea1446...; 22 lines)

```

# -*- coding: utf-8 -*-
import pandas as pd
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
df=pd.read_excel(CAREER,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
hasF=(s1=="Forestry")|(s2=="Forestry")
hasE=(s1=="Ecology")|(s2=="Ecology")
print("=== COMBINED cohort (has Forestry OR Ecology in either subfield) - Appendix Fig A2 ===")
print(f"{'':16}{'2024':>8}{'all-yr':>9}")
d24=df["year"]==2024
def cnt(mask): return int(mask.sum()), int((mask & d24).sum())
for name,mask in [("Ecology only", hasE & ~hasF),
                  ("Forestry only", hasF & ~hasE),
                  ("Both (F & E)", hasF & hasE),
                  ("-- Forestry TOTAL (only+both)", hasF),
                  ("-- Ecology TOTAL (only+both)", hasE)]:
    a,y=cnt(mask); print(f"{name:32}{y:>8}{a:>9}")
print("\n=== MAIN forestry cohort (Fig 3 groups) ===")
gF=(s1=="Forestry"); gE=(s1=="Ecology")&(s2=="Forestry")
for name,mask in [("Forestry group (AM=Forestry)",gF), ("Ecology group (AM=Ecology&AO=Forestry)",gE)]:
    a,y=cnt(mask); print(f"{name:40}{y:>8}{a:>9}")
print("\nIn combined cohort, is Forestry-total > Ecology-total?", int(hasF.sum())>int(hasE.sum()))

```

77. verify_final.py (SHA-256 b5095e9d9c0a68ba...; 26 lines)

```

# -*- coding: utf-8 -*-
import fitz, re, json
from openpyxl import load_workbook
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
txt="\n".join(p.get_text() for p in doc)
print("PAGES:",doc.page_count)
checks=[
    ("Methods unattributed clause","no affiliation country code or no institution name" in txt),
    ("App B unattributed note","3 author-year records lack a country code" in txt and "2 lack an institution name"
in txt),
    ("App C has NO note (0 missing)","author-year record" not in txt.split("Appendix C.")[1] if "Appendix C." in txt
else False),
    ("Tables B1-B4 present",all(f"Table B{i}" in txt for i in range(1,5))),
    ("Tables C1-C4 present",all(f"Table C{i}" in txt for i in range(1,5))),
    ("Figures B1-B5 present",all(f"Figure B{i}" in txt for i in range(1,6))),
    ("Figures C1-C5 present",all(f"Figure C{i}" in txt for i in range(1,6))),
]
for name,ok in checks: print(f"  [{ 'OK ' if ok else 'FAIL' }] {name}")
# workbook notes
wb=load_workbook(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx",read_only=True)
for sheet,expect in [("SA-B Forestry countries",True),("SA-B Forestry institutions",True),
                    ("SA-C Ecology countries",False),("SA-C Ecology institutions",False)]:
    ws=wb[sheet]; found=False
    for row in ws.iter_rows(values_only=True):
        if row and row[0] and "lack a" in str(row[0]): found=True; break
    print(f"  [{ 'OK ' if found==expect else 'FAIL' }] {sheet}: note {'present' if found else 'absent'} (expected
{'present' if expect else 'absent'})")
wb.close()

```

78. verify_fixes.py (SHA-256 4ba97956304e162a...; 44 lines)

```

# -*- coding: utf-8 -*-
import fitz, re
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
pages=[p.get_text() for p in doc]

```

```

txt="\n".join(pages)
print("PAGES:",doc.page_count)

checks=[
# narrative fixes
("MAJOR1 region: Western Europe named impact leader","Western Europe lead on impact intensity" in txt),
("MAJOR1: Northern Europe NOT claimed impact leader","Northern Europe lead on impact" not in txt),
("MAJOR2 institution: no 'output-oriented' claim","more output-oriented" not in txt),
("MAJOR2: institutions 'markedly lower' impact wording","markedly lower" in txt),
("MAJOR2: institution values non-zero (rr ok)", not re.search(r"Wageningen 0[,\\.\\s]", txt)),
("MINOR3 US: not 'lowest among the leaders'", "lowest among the leaders" not in txt),
("MINOR3 US: 'below Canada, Finland, Australia and Germany'", "below Canada, Finland, Australia and Germany" in
txt),
("MINOR4 Japan: not 'China and Japan are high-volume'", "China and Japan are high-volume" not in txt),
("LOW: 'one-seventh' removed","one-seventh" not in txt),
("LOW: 'about a sixth' / 'one-sixth' present","a sixth of the cohort" in txt or "one-sixth" in txt),
("LOW Brazil qualifier","highest among the cohorts that clear the five-scientist filter" in txt),
("nit: no 'Table B-5'/'C-5'", "Table B-5" not in txt and "Table C-5" not in txt),
# structure: taxonomy moved to Data
("Taxonomy Fig 3 present","Figure 3 (part 1 of 2)" in txt),
("Data has 'Field composition (taxonomy)'", "Field composition (taxonomy)" in txt),
("Results 4.1 is the ratios (not composition)", "4.1 The three intensity ratios" in txt),
("No stray '4.6 ' heading","4.6 Intensity" not in txt),
("Appendix A lean: 'A.1 Data description'", "A.1 Data description" in txt),
("Appendix A: only Tables A1-A3","Table A4" not in txt and "Table A5" not in txt),
]
allok=True
for name,ok in checks:
    print(f"    [{ 'OK ' if ok else 'FAIL' }] {name}"); allok=allok and ok

# institution values in prose (make sure rr didn't render 0)
m=re.search(r"Wageningen (\d+)", txt)
print("\nWageningen cit/art printed in prose:", m.group(1) if m else "NOT FOUND")
m2=re.search(r"Swedish University of Agricultural Sciences (\d+)", txt)
print("SLU cit/art printed:", m2.group(1) if m2 else "NOT FOUND")
# table numbering
mt=sorted(set(int(x) for x in re.findall(r"\bTable (\d+)[a-z]?\. ",txt)))
mf=sorted(set(int(x) for x in re.findall(r"\bFigure (\d+)\b",txt)))
print("Main tables:",mt)
print("Main figures:",sorted(set(mf)))
print("\nOverall:", "ALL FIX CHECKS PASS" if allok else "SOME FAILED")

```

79. verify_forestry.py (SHA-256 9dc01496ac2fb9cf...; 50 lines)

```

# -*- coding: utf-8 -*-
import json, pandas as pd, openpyxl
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
WB=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
YEARS=[2020,2021,2022,2023,2024]
F=json.load(open(f"{SCR}/forestry_data.json",encoding="utf-8"))
C=json.load(open(f"{SCR}/refine_data.json",encoding="utf-8"))

df=pd.read_excel(CAREER,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
fo=df[(s1=="Forestry")|(s2=="Forestry")]
print("== Independent source recompute (Forestry cohort) ==")
ok=True
for y in YEARS:
    sub=fo[fo.year==y]
    s=len(sub); p=int(sub["np"].sum()); c=int(sub["nc"].sum())
    js,jp,jc=F["totals"]["scientists"][str(y)],F["totals"]["publications"][str(y)],F["totals"]["citations"]
    [str(y)]
    match=(s==js and p==jp and c==jc); ok&=match
    print(f"    {y}: scientists {s}=={js}?{s==js}   pubs {p}=={jp}?{p==jp}   cites {c}=={jc}?{c==jc}")
print("Forestry totals reconcile:",ok)

print("\n== Banding integrity (head + bands == full ranking total) ==")
for keyname,banded_key,capped in [("country","country_banded",None),("university","university_banded",100)]:
    for m in ["scientists","publications","citations"]:
        full=F[keyname][m]; b=F[banded_key][m]
        full_sum=sum(r["total"] for r in full[:capped] if capped else full)
        banded_sum=sum(r["total"] for r in b["head"])+sum(g["total"] for g in b["bands"])
        print(f"    {keyname} {m}: head+bands={banded_sum} == top{capped} or 'all'={full_sum}?
{banded_sum==full_sum}")

```

```

print("\n== Dimension full-sums == per-year totals (Forestry) ==")
for dim in ["continent","country","us_state"]:
    for m in ["scientists"]:
        col_tot={y:sum(r[str(y)] for r in F[dim][m]) for y in YEARS}
        # continent/country exclude unknown; compare to totals only for country+continent(with unknown small)
        print(f"    {dim} {m} per-year:",[col_tot[y] for y in YEARS])
print("    Forestry per-year totals:",[F["totals"]["scientists"][str(y)] for y in YEARS])

print("\n== Combined totals sanity ==")
print("    combined 2024 sci/pub/cit:",C["totals"]["scientists"]["2024"],C["totals"]["publications"]
["2024"],C["totals"]["citations"]["2024"])
print("    forestry+ecology-only check: 2024 combined scientists =",C["totals"]["scientists"]["2024"])

print("\n== Workbook formula-error scan ==")
wb=openpyxl.load_workbook(WB); errs=0
for ws in wb:
    for row in ws.iter_rows():
        for cc in row:
            if isinstance(cc.value,str) and cc.value.startswith("#") and cc.value.endswith("!"): errs+=1
print("    formula-error cells:",errs,"| sheets:",len(wb.sheetnames))
print("DONE")

```

80. verify_merged.py (SHA-256 5919a614fe1ba3c4...; 47 lines)

```

# -*- coding: utf-8 -*-
import json, pandas as pd, openpyxl
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
WB=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
M=json.load(open(f"{SCR}/merged_data.json",encoding="utf-8")); MF=M["forestry"]
YEARS=[2020,2021,2022,2023,2024]
df=pd.read_excel(CAREER,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
gF=(s1=="Forestry"); gE=(s1=="Ecology") & (s2=="Forestry")
fo=df[gF|gE] # UNION forestry research cohort
print("group sizes: F(AM=Forestry)=",int(gF.sum()),"E(AM=Ecology&AO=Forestry)=",int(gE.sum()),"union=",int((gF|gE).sum()))
print("== UNION forestry per-year totals vs merged JSON ==")
ok=True
for i,y in enumerate(YEARS):
    sub=fo[fo.year==y]
    s,p,c=len(sub),int(sub["np"].sum()),int(sub["nc"].sum())
    js,jp,jc=MF["totals"]["scientists"][str(y)],MF["totals"]["publications"][str(y)],MF["totals"]["citations"][str(y)]
    m=(s==js and p==jp and c==jc); ok&=m
    print(f"    {y}: {s}=={js},{p}=={jp},{c}=={jc} -> {m}")
print("totals OK:",ok)

# banded country: head+bands scientists total == full country scientists total
bd=MF["country_banded"]
hb=sum(sum(m["s"]) for m in bd["head"])+sum(sum(g["s"]) for g in bd["bands"])
full=sum(sum(m["s"]) for m in MF["country"])
print("country banded scientists head+bands:",hb,"== full:",full,"->",hb==full)
# same for pubs/cites
for metric in ["p","c"]:
    hb2=sum(sum(m[metric]) for m in bd["head"])+sum(sum(g[metric]) for g in bd["bands"])
    full2=sum(sum(m[metric]) for m in MF["country"])
    print(f"    country banded {metric}: {hb2} == {full2} -> {hb2==full2}")

# multi-root taxonomy total == union n
TAX=M["taxonomy"]; tsum=sum(roots_n for aq in TAX["roots"] for am in TAX["roots"][aq] for roots_n in TAX["roots"]
[aq][am].values())
print("multi-root taxonomy leaf-sum:",tsum,"== union n:",TAX["n_total"],"->",tsum==TAX["n_total"])
# group split sums to union
gsum=sum(sum(g["s"]) for g in MF["group"]); print("group split scientists sum:",gsum,"== union
total:",sum(MF["totals"]['scientists'][str(y)] for y in YEARS),"->",gsum==sum(MF["totals"]['scientists'][str(y)]
for y in YEARS))

# workbook opens + formula errors
wb=openpyxl.load_workbook(WB); errs=0
for ws in wb:
    for row in ws.iter_rows():
        for cc in row:
            if isinstance(cc.value,str) and cc.value.startswith("#") and cc.value.endswith("!"): errs+=1
print("workbook sheets:",len(wb.sheetnames),"| formula-error cells:",errs,"| has F Forestry Taxonomy:", "F
Forestry Taxonomy" in wb.sheetnames)

```

```
print("DONE")
```

81. verify_pdf.py (SHA-256 44b77a5b3225eeca...; 20 lines)

```
# -*- coding: utf-8 -*-
import fitz, re
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
txt=""\n".join(p.get_text() for p in doc)
print("PAGES:",doc.page_count)
print("--- table captions ---")
for pat in ["Table 5.", "Table 6a", "Table 6b", "Table 6c", "Table 7a", "Table 7b", "Table 7c", "Table A3", "Table A4"]:
    m=re.search(re.escape(pat)+r"[\n]*",txt)
    print(f"[{'OK' if m else 'MISS'}] {(m.group(0)[:95] if m else pat)}")
print("--- band/top-20 leftovers (should be none) ---")
found=False
for bad in ["band", "Band", "~10%", "top 20", "top-20", "21-100", "cumulative", "remainder"]:
    c=txt.count(bad)
    if c:
        found=True
        i=txt.find(bad); ctx=txt[max(0,i-45):i+45].replace("\n", " ")
        print(f" {bad!r}: {c}x ...{ctx}...")
if not found: print(" (clean - no band language)")
print("--- 'Avg/yr' header occurrences:",txt.count("Avg/yr"),"(expect 6: tables 6a-c,7a-c) ---")
# count top-25 style rows in a split table: find Table 7a block and count numeric rows
```

82. verify_refine.py (SHA-256 e0d7ef8386802ac2...; 47 lines)

```
import json, pandas as pd
from cc_defs import CC
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudio\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
D=json.load(open(f"{SCR}/refine_data.json",encoding="utf-8"))
YEARS=[2020,2021,2022,2023,2024]
# independent recompute from career file
df=pd.read_excel(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx",sheet_name="Data")
tot={m:{y:0 for y in YEARS} for m in ["scientists","publications","citations"]}
for y in YEARS:
    s=df[df.year==y]
    tot["scientists"][y]=len(s); tot["publications"][y]=int(s["np"].sum()); tot["citations"][y]=int(s["nc"].sum())
ok=True
def chk(name,cond):
    global ok
    print(("PASS" if cond else "FAIL"),name); ok=ok and cond

# 1 totals match
for m in tot:
    chk(f"totals[{m}]", all(int(D["totals"][m][str(y)])==tot[m][y] for y in YEARS))

# 2 dimension per-year sums equal totals (for full-coverage dims)
def dim_sum(dim,m,y): return sum(r[str(y)] for r in D[dim][m])
for dim in ["country","field","orient"]:
    for m in ["scientists","publications","citations"]:
        chk(f"{dim} per-yr sum==totals [{m}]", all(round(dim_sum(dim,m,y))==round(tot[m][y]) for y in YEARS))
# continent excludes 2 unknown-country rows -> allow small diff on scientists
for m in ["scientists"]:
    diff=tot[m][2024]-dim_sum("continent",m,2024)
    chk(f"continent 2024 within unknowns (diff={diff})", 0<=diff<=5)

# 3 banded reconstruction == full metric total
for dim,key,cap in [("country","country_banded",None),("university","university_banded",100)]:
    for m in ["scientists","publications","citations"]:
        b=D[key][m]
        recon=sum(r["total"] for r in b["head"])+sum(g["total"] for g in b["bands"])
        full=sum(r["total"] for r in (D[dim][m][:cap] if cap else D[dim][m]))
        chk(f"{key} recon==top{cap or 'all'} [{m}] ({recon} vs {full})", round(recon)==round(full))

# 4 university totals (all) sum to grand total (each author's np/nc counted once per year under its inst)
for m in ["scientists","publications","citations"]:
    us=sum(r["total"] for r in D["university"][m])
    # exclude 7 blank-inst rows from grand total
    blank=df[df["inst_name"].isna()]
    gt=sum(tot[m][y] for y in YEARS) - (len(blank) if m=="scientists" else int(blank["np"].sum()) if m=="publications" else int(blank["nc"].sum()))
    chk(f"university total==grand-blank [{m}] ({us} vs {gt})", round(us)==round(gt))
```

```
print("\nALL PASS" if ok else "\nSOME FAILED")
```

83. verify_report.py (SHA-256 31fc0aa92727458f...; 59 lines)

```
import json, pandas as pd, openpyxl
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
REP = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_Representation_Report.xlsx"
CAREER = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
EXPECT={2020:4344,2021:4516,2022:4646,2023:4825,2024:5043}

subdiv = {int(k):v for k,v in json.load(open(f"{SCR}/inst_subdiv.json",encoding="utf-8")).items()}
idx = pd.read_csv(f"{SCR}/inst_index.csv"); idx["sd"]=idx["id"].map(subdiv)
look={ (r.country,r.inst):r.sd for r in idx.itertuples()}
df=pd.read_excel(CAREER,sheet_name="Data")
CN={"usa":"United States","can":"Canada","aus":"Australia"}

wb=openpyxl.load_workbook(REP,data_only=True)
print("sheets:",wb.sheetnames)

def col_totals(sheet):
    ws=wb[sheet]; rows=list(ws.iter_rows(values_only=True)); hdr=rows[0]
    yi={str(y):hdr.index(str(y)) for y in YEARS}
    # sum data rows until a TOTAL row
    tot={y:0 for y in YEARS}
    for r in rows[1:]:
        if r[0] in ("TOTAL",None): continue
        for y in YEARS:
            v=r[yi[str(y)]]
            if isinstance(v,(int,float)): tot[y]+=v
    return tot

# full-coverage dims must equal release sizes
for sheet in ["By Continent","By Country","By Field"]:
    t=col_totals(sheet)
    ok=all(t[y]==EXPECT[y] for y in YEARS)
    print(f"{sheet}: per-year {[t[y] for y in YEARS]} == release sizes? {ok}")

# subnational sheets equal that country's per-year totals
for code,sheet in [{"usa","By US State"}, {"can","By Canada Province"}, {"aus","By Australia State"}]:
    sub=df[df.cntry==code]; exp={y:int((sub.year==y).sum()) for y in YEARS}
    t=col_totals(sheet)
    ok=all(t[y]==exp[y] for y in YEARS)
    print(f"{sheet}: {[t[y] for y in YEARS]} == country totals {[exp[y] for y in YEARS]}? {ok}")

# independent recompute of 2024 continent + top country
CC=pd.read_csv # noop
d24=df[df.year==2024]
print("\nindep 2024 USA rows:",int((d24.cntry=='usa').sum()))
usd=d24[d24.cntry=='usa'].copy()
usd["sd"]=usd["inst_name"].map(lambda x: look.get(("United States",x),"Unknown"))
print("indep US state top6 (2024):")
print(usd["sd"].value_counts().head(6).to_string())

# formula error scan (should be none; no formulas)
wb2=openpyxl.load_workbook(REP)
errs=0
for ws in wb2:
    for row in ws.iter_rows():
        for c in row:
            if isinstance(c.value,str) and c.value.startswith("#") and c.value.endswith("!"): errs+=1
print("\nformula-error cells:",errs)
print("DONE")
```

84. verify_restructure.py (SHA-256 0ffed0bbea33181a...; 53 lines)

```
# -*- coding: utf-8 -*-
import fitz, re, json
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
pages=[p.get_text() for p in doc]
```

```

txt="\n".join(pages)
print("PAGES:",doc.page_count)

def find_all(pat):
    return sorted(set(re.findall(pat,txt)),key=lambda s:(len(s),s))

# main tables 1..13, figures 1..10
def nums(prefix):
    got={}
    for m in re.finditer(rf"\b{prefix} (\d+) [a-z]?\. ",txt):
        got[int(m.group(1))]=got.get(int(m.group(1)),0)+1
    return got
mt=nums("Table"); mf=nums("Figure")
print("Main Table numbers present:",sorted(mt))
print("Main Figure numbers present:",sorted(mf))
# appendix tables/figs
for L in "ABC":
    at=sorted(set(int(x) for x in re.findall(rf"Table {L} (\d+)\. ",txt)))
    af=sorted(set(int(x) for x in re.findall(rf"Figure {L} (\d+)\. ",txt)))
    print(f"Appendix {L}: tables {at} | figures {af}")

checks=[
    (" $ headings - Data holds counts","This section presents the full set of absolute counts" in txt),
    ("Results retitled to intensity","Results - the intensity of output and impact" in txt),
    ("ratio defs in Methods","articles per author = " in txt.replace("\n"," ") or "Articles per author" in txt),
    ("quality-vs-quantity heading","Quality versus quantity by country" in txt),
    ("correlation reported (r =)","bool(re.search(r"r = [\+-]\d\.\d\d",txt))",
    ("Table 8 cohort+group ratios","Intensity ratios for the whole cohort and the two groups" in txt),
    ("scatter figure caption","Quality versus quantity by country: citations per article" in txt),
    ("Ecology density claim","citation-dense" in txt),
    ("Appendix B ratio subsection","B.4 Output and impact intensity" in txt),
    ("Appendix C ratio subsection","C.4 Output and impact intensity" in txt),
    ("Appendix A intensity heading","A.2 Output and impact intensity" in txt),
    ("NO stale 'Tables 6a','7a'", "Table 6a" not in txt and "Table 7a" not in txt),
    ("NO stale 'Tables 7a-c'", "7a-c" not in txt and "6a-c" not in txt),
    ("NO stale '~10% bands'", "~10% bands" not in txt),
    ("US maps now Figure 2","Figure 2. Scientists, publications and citations by U.S. state" in txt),
    ("taxonomy now Figure 3","Figure 3 (part 1 of 2)" in txt),
]
print("\nCHECKS:")
allok=True
for name,ok in checks:
    print(f" [{ 'OK ' if ok else 'FAIL' }] {name}"); allok=allok and ok

# sanity: count-metric captions (Publications ('000)/Citations (M)) should still appear in Data/appendix count
tables
print("\nCitations (M)' occurrences (count tables):", txt.count("Citations (M)"))
print("'Citations / article' occurrences (ratio tables):", txt.count("Citations / article"))
print("\nOverall:", "ALL STRUCTURE CHECKS PASS" if allok else "SOME CHECKS FAILED")

```

85. verify_sa.py (SHA-256 3392003c68d25587...; 54 lines)

```

# -*- coding: utf-8 -*-
# Independent verification of the single-author appendices (B, C):
# (1) recompute SA aggregates straight from the raw career xlsx and compare to the JSON the pipeline produced;
# (2) check the PDF has the right appendix structure + captions.
import json, re, pandas as pd, numpy as np
SCR=r"C:\Users\ADMINI~1\AppData\Local\Temp\claudc\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
CAREER=r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Ecology_career_2020-2024.xlsx"
YEARS=[2020,2021,2022,2023,2024]
from cc_defs import CC
df=pd.read_excel(CAREER,sheet_name="Data")
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
df=df[(s1=="Forestry") | ((s1=="Ecology") & (s2=="Forestry"))].copy()
s1=df["sm-subfield-1"].astype(str).str.strip(); s2=df["sm-subfield-2"].astype(str).str.strip()
df["group"]=np.where(s1=="Forestry","Forestry","Ecology")
for c in ["nps","ncs"]: df[c]=pd.to_numeric(df[c],errors="coerce").fillna(0).astype(int)
df["country"]=df["cntry"].map(lambda c: CC[c][0] if c in CC else "(unknown)")

ok=True
def check(name,a,b):
    global ok
    good=(a==b)
    ok=ok and good
    print(f" [{ 'OK ' if good else 'FAIL' }] {name}: pipeline={a} independent={b}")

```

```

for grp,jf,letter in [("Forestry","forestry_B_data.json","B"),("Ecology","forestry_C_data.json","C")]:
    print(f"\n=== Appendix {letter}: {grp} group, single-author (nps>0) ===")
    d=df[(df.group==grp)&(df.nps>0)]
    J=json.load(open(f"{SCR}/{jf}",encoding="utf-8"))
    for y in YEARS:
        dy=d[d.year==y]
        check(f"{y} scientists", J["totals"]["scientists"][str(y)], int(len(dy)))
        check(f"{y} SA papers ", J["totals"]["publications"][str(y)], int(dy.nps.sum()))
        check(f"{y} SA cites ", J["totals"]["citations"][str(y)], int(dy.ncs.sum()))
    # top-5 countries by scientists in 2024 (compare ordering+counts)
    d24=d[d.year==2024]
    ind=d24.groupby("country").size().sort_values(ascending=False).head(5)
    jc={r["key"]:r["2024"] for r in J["country"]["scientists"]}
    print(" top-5 countries by scientists (2024): independent vs pipeline")
    for cty,cnt in ind.items():
        check(f" {cty}", jc.get(cty,0), int(cnt))

print("\n=== PDF structure ===")
import fitz
doc=fitz.open(r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.pdf")
txt=""
for p in doc:
    txt+=p.get_text()
print("PAGES:",doc.page_count)
for pat in ["Appendix B. The Forestry Group","Appendix C. The Ecology Group",
            "Table B1","Table B2","Table B3","Table B4","Table C1","Table C2","Table C3","Table C4",
            "Figure B1","Figure B5","Figure C1","Figure C5","Single-author papers","Cites to SA papers"]:
    print(f" [{ 'OK ' if pat in txt else 'MISS' }] {pat}")
# confirm B1 totals row appears with the expected 2024 single-author paper count for Forestry (5,448) and Ecology (1,402)
for grp,val in [("Forestry-SA 2024 papers 5,448","5,448"),("Ecology-SA 2024 papers 1,402","1,402")]:
    print(f" [{ 'OK ' if val in txt else 'MISS' }] {grp}")
print("\nRESULT:", "ALL DATA CHECKS PASSED" if ok else "*** SOME DATA CHECKS FAILED ***")

```

86. xcheck.py (SHA-256 8ce6b3e2c14df056...; 66 lines)

```

import json
from openpyxl import load_workbook
SCR = r"C:\Users\ADMINI~1\AppData\Local\Temp\claude\C--D-Dropbox-ClaudeCodeTest\e9e03403-d331-4f33-a33f-6af155a73519\scratchpad\stanford"
WB = r"C:\D\Dropbox\ClaudeCodeTest\StanfordList_Forestry_Representation_Report.xlsx"
YEARS=["2020","2021","2022","2023","2024"]
def load(n):
    with open(f"{SCR}\{n}",encoding="utf-8") as f: return json.load(f)
B=load("forestry_B_data.json"); C=load("forestry_C_data.json")
wb=load_workbook(WB, data_only=True)

# Verify workbook Total and Avg columns are internally consistent, and
# verify Total column sum equals grand total across 5 yrs for a couple blocks.
def check_totals_avg(sheet, ystart, keycol):
    ws=wb[sheet]
    bad=0; checked=0
    # find first block header
    for r in range(1,ws.max_row+1):
        a=ws.cell(row=r,column=1).value
        if isinstance(a,str) and "By SCIENTISTS" in a:
            hr=r; break
    r=hr+2
    while r<=ws.max_row:
        k=ws.cell(row=r,column=keycol).value
        if k is None or (isinstance(k,str) and k.strip()==""): break
        vals=[ws.cell(row=r,column=ystart+o).value or 0 for o in range(5)]
        tot=ws.cell(row=r,column=ystart+5).value
        avg=ws.cell(row=r,column=ystart+6).value
        exp_tot=sum(int(v) for v in vals)
        exp_avg=exp_tot/5
        if tot!=exp_tot: bad+=1
        if abs(float(avg)-exp_avg)>0.05: bad+=1
        checked+=1
        r+=1
    print(f"{sheet}: scientists block rows checked={checked}, Total/Avg mismatches={bad}")

check_totals_avg("SA-B Forestry countries",2,1)
check_totals_avg("SA-B Forestry institutions",3,1)
check_totals_avg("SA-C Ecology countries",2,1)
check_totals_avg("SA-C Ecology institutions",3,1)

# Confirm workbook country rows == JSON country dimension records exactly (B scientists)
def match_dim(sheet, jobj, dim, ystart, keycol):

```

```

ws=wb[sheet]
recs=jobj[dim]["scientists"]
jmap={r["key"]:r for r in recs}
for rr in range(1,ws.max_row+1):
    a=ws.cell(row=rr,column=1).value
    if isinstance(a,str) and "By SCIENTISTS" in a: hr=rr; break
r=hr+2; mism=0; n=0
while r<=ws.max_row:
    k=ws.cell(row=r,column=keycol).value
    if k is None or (isinstance(k,str) and k.strip()==""): break
    n+=1
    jr=jmap.get(k)
    if jr is None:
        mism+=1; print(" KEY not in JSON:",k)
    else:
        for o,y in enumerate(YEARS):
            wv=ws.cell(row=r,column=ystart+o).value or 0
            if int(wv)!=int(jr.get(y,0) or 0):
                mism+=1; print(f" cell diff {k} {y}: wb={wv} json={jr.get(y)}")
        r+=1
print(f"{sheet}: matched {n} rows to JSON {dim} scientists, cell mismatches={mism}")

match_dim("SA-B Forestry countries",B,"country",2,1)
match_dim("SA-C Ecology countries",C,"country",2,1)

```