

US Census IV&V  
Independent Verification of Census Bureau Data Releases

Troy Altus

2025-12-31



# Table of contents

<b>1</b>	<b>How to Read a Census Release</b>	<b>1</b>
<b>2</b>	<b>How to Read a Census Release</b>	<b>3</b>
2.1	The Press Release Pipeline . . . . .	3
2.2	The Three Main Surveys . . . . .	3
2.3	The Verification Workflow . . . . .	4
2.4	The Census API in Practice . . . . .	4
2.5	Summary . . . . .	5
2.6	Further Reading . . . . .	5
<b>3</b>	<b>Preface</b>	<b>7</b>
	<b>Preface</b>	<b>9</b>
	What This Is . . . . .	9
	The Census API . . . . .	9
	How to Add a New Note . . . . .	10
<b>4</b>	<b>Education Enrollment, 2024</b>	<b>11</b>
4.1	The Survey . . . . .	11
4.2	The Data Pull . . . . .	11
4.3	Headline Figures . . . . .	13
4.4	The Recovery Context . . . . .	14
4.5	The Gender Gap . . . . .	15
4.6	The Black Enrollment Surge . . . . .	15
4.7	Peer Review: What the Release Gets Right and Wrong . . . . .	16
	4.7.1 Strengths of the methodology . . . . .	16
	4.7.2 Concerns . . . . .	16
	4.7.3 Suggested improvements . . . . .	17
	4.7.4 Potential follow-on projects . . . . .	17
4.8	Summary . . . . .	18
	<b>References</b>	<b>19</b>
	<b>Appendices</b>	<b>21</b>
<b>A</b>	<b>Census API Utilities</b>	<b>21</b>

<b>Census API Utilities</b>	<b>23</b>
Core API Wrapper . . . . .	23
ACS Education Variables (Reference) . . . . .	25
Weighted Estimate Helper . . . . .	26

## Chapter 1

# How to Read a Census Release



## Chapter 2

# How to Read a Census Release

### Learning Objectives

- Understand how Census Bureau press releases are structured and what they typically claim
- Know the difference between ACS, CPS, and Decennial Census data products
- Understand the Census API well enough to pull a table and compare it to a headline figure

## 2.1 The Press Release Pipeline

The Census Bureau publishes hundreds of data releases each year. The public-facing version is a press release with a headline, a few bullet points, and a link to detailed tables. The statistical version is a set of tables, microdata files, and an API endpoint that returns the underlying estimates.

The headline figure is always accurate in the sense that it correctly describes what the tables say. What the press release does not always make clear is which survey produced the number, what population was counted, what years are being compared, and what the margin of error is. Those details are in the technical documentation, and they matter.

A release that says “college enrollment reached X million” might be using the October CPS supplement (household survey, smaller sample, asks about current enrollment) or the ACS (larger sample, asks about enrollment in the past 12 months). These are not the same question. The difference is rarely explained in the press release and is always explained in the methodology.

## 2.2 The Three Main Surveys

Understanding which survey produced a number is the first step in verifying it.

**American Community Survey (ACS)** Annual survey of approximately 3.5 million households. The gold standard for detailed demographic and economic characteristics. Published in one-year (geographies 65,000 population) and five-year (all geographies) editions. Variables include

education, income, housing, employment, language, citizenship. API endpoint: `/data/acs/acs1` and `/data/acs/acs5`.

**Current Population Survey (CPS)** Monthly household survey conducted jointly by the Bureau of Labor Statistics and Census. The primary source for labor force statistics. Supplemental surveys run annually on specific topics: school enrollment (October), income and poverty (March), voting (November). Smaller sample than ACS; better for tracking month-to-month change. API endpoint: `/data/cps/<supplement>/<month>`.

**Decennial Census** Full count (or near-full count) every ten years. Last conducted 2020. Used for apportionment, redistricting, and as a benchmark for survey weighting. API endpoint: `/data/dec/`.

Most education enrollment figures in Census press releases come from the CPS October School Enrollment supplement.

## 2.3 The Verification Workflow

Each note in this series follows the same sequence:

1. CLAIM - what the press release said, verbatim
2. SOURCE - which survey, which year, which variable group
3. PULL - API call that retrieves the relevant estimate
4. COMPARE - press release figure vs. API result
5. OBSERVE - what the data can do beyond the headline

Step 4 should reproduce the headline figure exactly. If it does not, the discrepancy is documented and explained (usually it is a rounding convention or a population restriction that the press release does not spell out). The interesting work is step 5: what else does this data product contain, what geographies can be disaggregated, what time series goes back how far, and what follow-on questions are worth pursuing.

## 2.4 The Census API in Practice

```
import requests
import os
import pandas as pd

def census_get(dataset, year, variables, geography, api_key=None):
    """
    Minimal Census API wrapper. Returns a DataFrame.

    Parameters
    -----
    dataset : str    e.g. "acs/acs1", "cps/school/oct"
    year    : int    e.g. 2024
    variables : list  e.g. ["NAME", "B15003_001E"]
    geography : str   e.g. "us:1", "state:*"
    api_key  : str   defaults to CENSUS_API_KEY env var
```

```

"""
key = api_key or os.getenv("CENSUS_API_KEY", "")
base = f"https://api.census.gov/data/{year}/{dataset}"
params = {
    "get": ",".join(variables),
    "for": geography,
    "key": key,
}
r = requests.get(base, params=params, timeout=30)
r.raise_for_status()
data = r.json()
return pd.DataFrame(data[1:], columns=data[0])

def list_variables(dataset, year):
    """
    Print available variable groups for a dataset/year.
    Useful for exploration.
    """
    url = f"https://api.census.gov/data/{year}/{dataset}/groups.json"
    r = requests.get(url, timeout=30)
    r.raise_for_status()
    groups = r.json().get("groups", [])
    df = pd.DataFrame(groups)[["name", "description"]].sort_values("name")
    return df

```

### API Key Setup

The Census API key is stored in macOS Keychain and loaded by `~/secrets/load_keys.sh` at login. In Python: `os.getenv("CENSUS_API_KEY")`. If running on a new machine, get a free key at [api.census.gov/data/key\\_signup.html](https://api.census.gov/data/key_signup.html).

## 2.5 Summary

Census press releases present accurate but compressed descriptions of survey results. Verifying them requires knowing which survey produced the number, locating the variable in the API, and pulling the estimate directly. The three main sources are the ACS, the CPS, and the Decennial Census; most education and labor statistics in press releases come from CPS supplements. Each note in this series walks through that verification workflow for one release, with observations on what the underlying data product can support beyond the headline.

## 2.6 Further Reading

- [Census Bureau API documentation](#) — complete list of available datasets and variables
- [Census Academy](#) — free training on using Census data products
- Citro, C.F. & Michael, R.T. (eds.) *Measuring Poverty: A New Approach*. National

Academies Press, 1995. Background on how survey-based poverty and enrollment estimates are constructed.

## Chapter 3

## Preface



# Preface

## What This Is

The US Census Bureau sends email announcements when new data releases go out — surveys completed, tables published, headline figures highlighted. This series takes those announcements at face value and then checks them.

The goal is not adversarial. The Bureau is a competent statistical agency and its headline figures are generally accurate. The goal is to understand *how* the numbers were produced: which survey, which variables, which population definition, which suppression rules, and what the API returns when you ask the same question the press release answered. That understanding is worth having on its own terms, and it surfaces questions worth pursuing as standalone projects.

Each note follows the same structure: the claim, the source, the API pull, the comparison, and observations about what the data can and cannot do. Notes accumulate as the emails arrive.

## The Census API

The US Census Bureau exposes most of its data through a public REST API at `api.census.gov`. An API key is required but free. The key used in this analysis is stored in macOS Keychain and loaded into the shell environment; code accesses it via `os.getenv("CENSUS_API_KEY")`.

The main endpoints used here are:

Endpoint	Dataset	Notes
<code>/data/acs/acs1</code>	American Community Survey 1-year	Annual, geography 65K pop
<code>/data/acs/acs5</code>	American Community Survey 5-year	Annual, all geographies
<code>/data/cps/school/oct</code>	Current Population Survey — School Enrollment	October supplement
<code>/data/dec/dp</code>	Decennial Census Data Profiles	2020 onward

Variable names, group codes, and geography parameters are documented at `api.census.gov/data/<year>/<dat`. The API is not always intuitive. Part of what these notes capture is what you learn navigating it.

## How to Add a New Note

When a new Census email arrives worth checking:

1. Add a new `.qmd` file: `NN-<topic>-<year>.qmd`
2. Register it in `_quarto.yml` under `chapters`:
3. Use the template structure from any existing note: `claim` → `source` → `pull` → `compare` → `observations`
4. Run `pixi run render` and push

The bibliography accumulates in `references.bib`. API utility functions accumulate in `appendix-a-code.qmd`.

---

*Troy Altus 2026*

# Chapter 4

## Education Enrollment, 2024

### This Note

**Release:** US Census Bureau, 2024 School Enrollment data (CPS October supplement)  
**Survey:** Current Population Survey — October School Enrollment Supplement, 2024  
**Records pulled:** 99,045 (weighted to national population) **Verdict:** Headline figures reproduced. Notable findings on gender gap, Black enrollment surge, and continued below-peak college recovery merit closer attention than a typical press release affords.

### 4.1 The Survey

The Census Bureau’s school enrollment figures come from the **Current Population Survey (CPS) October Supplement** — a monthly household survey that, every October, adds a battery of questions about school attendance. The reference period is the survey week, typically the week containing October 15. Respondents report whether household members are currently enrolled, at what level, and whether full- or part-time.

This is not the same as counting students. It is counting *households reporting students*, weighted to the national population. The distinction matters and is addressed in the review section below.

### 4.2 The Data Pull

All analysis below uses the Census Bureau’s public API at [api.census.gov/data/2024/cps/school/oct](https://api.census.gov/data/2024/cps/school/oct). Key variables:

Variable	Description
PESCHLVL	School level: 1 = High school, 2 = College/university
PESCH35	Enrolled, age 3–5 (nursery / preK / kindergarten)
PESCH614	Enrolled, age 6–14
PESCHFT	College enrollment type: 1 = Full-time, 2 = Part-time

Variable	Description
PESEX	Sex: 1 = Male, 2 = Female
PTDTRACE	Race (detailed)
PEHSPNON	Hispanic origin: 1 = Hispanic, 2 = Non-Hispanic
PWSSWGT	Final person weight — use this for all population estimates

```

import os, json
import pandas as pd
import numpy as np
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

# Load pre-pulled data (cached from API pull)
df24 = pd.read_pickle("data/processed/cps_school_2024.pkl")
df23 = pd.read_pickle("data/processed/cps_school_2023.pkl")
with open("data/processed/trend_2019_2024.json") as f:
    trend = json.load(f)

W = "PWSSWGT"

def wtot(df, mask):
    """Weighted total for a boolean mask, in millions."""
    return df[mask][W].sum() / 1e6

# 2024 core estimates
e35_24 = wtot(df24, df24["PESCH35"]==1)
e614_24 = wtot(df24, df24["PESCH614"]==1)
ehs_24 = wtot(df24, df24["PESCHLVL"]==1)
ecol_24 = wtot(df24, df24["PESCHLVL"]==2)
eft_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PESCHFT"]==1))
ept_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PESCHFT"]==2))
em_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PESEX"]==1))
ef_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PESEX"]==2))
ehisp_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PEHSPNON"]==1))
ewh_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PTDTRACE"]==1) & (df24["PEHSPNON"]==2))
ebl_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PTDTRACE"]==2))
eas_24 = wtot(df24, (df24["PESCHLVL"]==2) & (df24["PTDTRACE"]==4))

# 2023 comparison
ecol_23 = wtot(df23, df23["PESCHLVL"]==2)
ehs_23 = wtot(df23, df23["PESCHLVL"]==1)
e35_23 = wtot(df23, df23["PESCH35"]==1)
ebl_23 = wtot(df23, (df23["PESCHLVL"]==2) & (df23["PTDTRACE"]==2))

```

```
em_23 = wtot(df23, (df23["PESCHLVL"]==2) & (df23["PESEX"]==1))
ef_23 = wtot(df23, (df23["PESCHLVL"]==2) & (df23["PESEX"]==2))
```

### 4.3 Headline Figures

The table below reproduces the top-line estimates from 2024 and compares them to 2023.

```
rows = [
    ("Nursery / preschool / kindergarten (age 3-5)", e35_24, wtot(df23, df23["PESCH35"]==1),
    ("Age 6-14", e614_24, wtot(df23, df23["PESCH614"]==1),
    ("High school", ehs_24, ehs_23),
    ("College - total", ecol_24, ecol_23),
    (" Full-time", eft_24, wtot(df23, (df23["PESCHLVL"]==2) & (df23["PESEX"]==1)),
    (" Part-time", ept_24, wtot(df23, (df23["PESCHLVL"]==2) & (df23["PESEX"]==2))
]

tbl = pd.DataFrame(rows, columns=["Category", "2024 (M)", "2023 (M)"])
tbl["Change (M)"] = tbl["2024 (M)"] - tbl["2023 (M)"]
tbl["Change (%)"] = (tbl["Change (M)"] / tbl["2023 (M)"]) * 100
for c in ["2024 (M)", "2023 (M)", "Change (M)"]:
    tbl[c] = tbl[c].map("{:.3f}".format)
tbl["Change (%)"] = tbl["Change (%)"].map("{:+.1f}%".format)

tbl.style.set_properties(**{"text-align": "left"})
```

Table 4.1: 2024 CPS October School Enrollment — API-derived estimates

Category	2024 (M)	2023 (M)	Change (M)	Change (%)
0 Nursery / preschool / kindergarten (age 3–5)	7.723	7.433	0.290	+3.9%
1 Age 6–14	35.596	35.631	-0.035	-0.1%
2 High school	10.256	10.115	0.142	+1.4%
3 College — total	17.358	17.233	0.124	+0.7%
4 Full-time	13.602	13.385	0.217	+1.6%
5 Part-time	3.756	3.848	-0.092	-2.4%

#### Key takeaways from the table:

- College enrollment ticked up **+0.7%** (+124,000) year-over-year, continuing a slow recovery from the 2022 trough.
- Full-time college enrollment grew +1.6%; part-time fell –2.4%, suggesting a shift toward full-time status rather than net new students.
- Nursery/preschool enrollment jumped **+3.9%** — the largest single-year increase in recent years.
- High school enrollment rose modestly (+1.4%).

## 4.4 The Recovery Context

The press release headline of “college enrollment up” is accurate but incomplete without the five-year view. Enrollment peaked before COVID, dropped through 2022, and has only partially recovered.

```

years = [int(y) for y in sorted(trend.keys())]
col_total = [trend[str(y)]["college"] for y in years]
female     = [trend[str(y)]["female"]  for y in years]
male       = [trend[str(y)]["male"]    for y in years]

fig = go.Figure()
fig.add_trace(go.Scatter(x=years, y=col_total, name="Total",
    line=dict(color="#4f8ef7", width=2.5),
    mode="lines+markers", marker=dict(size=7)))
fig.add_trace(go.Scatter(x=years, y=female, name="Female",
    line=dict(color="#a47ee8", width=2, dash="solid"),
    mode="lines+markers", marker=dict(size=6)))
fig.add_trace(go.Scatter(x=years, y=male, name="Male",
    line=dict(color="#e06c4a", width=2, dash="solid"),
    mode="lines+markers", marker=dict(size=6)))

fig.add_vline(x=2020, line_dash="dot", line_color="gray",
    annotation_text="COVID-19", annotation_position="top right")

fig.update_layout(
    xaxis_title="Year", yaxis_title="Enrolled (millions)",
    legend=dict(orientation="h", y=1.1),
    yaxis=dict(range=[6, 20]),
    template="plotly_white", height=400,
    margin=dict(t=40, b=40)
)
fig.show()

```

Unable to display output for mime type(s): text/html

(a) College enrollment trend 2019–2024, by sex (CPS October, API-derived)

Unable to display output for mime type(s): text/html

(b)

Figure 4.1

Total college enrollment in 2024 (**17.36M**) remains **4.0% below the 2019 level** (18.09M). The press release’s positive YoY framing is accurate; the incomplete recovery is the more important story.

## 4.5 The Gender Gap

The most striking structural feature of these data is not the headline number — it is the male-female enrollment split.

```
gender_ratio = [trend[str(y)]["female"] / (trend[str(y)]["female"] + trend[str(y)]["male"])
                for y in years]

fig = make_subplots(rows=1, cols=2,
                    subplot_titles=("Enrollment by sex (M)", "Female share (%)"))

fig.add_trace(go.Bar(x=years, y=female, name="Female",
                     marker_color="#a47ee8"), row=1, col=1)
fig.add_trace(go.Bar(x=years, y=male, name="Male",
                     marker_color="#e06c4a"), row=1, col=1)

fig.add_trace(go.Scatter(x=years, y=gender_ratio, name="Female %",
                         line=dict(color="#a47ee8", width=2.5), mode="lines+markers",
                         marker=dict(size=7), showlegend=False), row=1, col=2)
fig.add_hline(y=50, line_dash="dot", line_color="gray", row=1, col=2)

fig.update_layout(barmode="group", template="plotly_white",
                  height=380, legend=dict(orientation="h", y=1.12),
                  margin=dict(t=50, b=40))
fig.show()
```

Unable to display output for mime type(s): text/html

Figure 4.2: College enrollment by sex, 2019–2024 (CPS October)

In 2024, **57.7% of college students are female** — 10.02M women vs. 7.34M men. Male enrollment has fallen every year since 2019, declining **9.9% over five years** (8.14M → 7.34M). Female enrollment has held roughly flat before ticking up in 2024.

The gender gap is mentioned in most enrollment press releases but rarely foregrounded. At 57.7/42.3, it is larger than most education-policy discussions acknowledge.

## 4.6 The Black Enrollment Surge

```
black = [trend[str(y)]["black"] for y in years]
hisp = [trend[str(y)]["hisp"] for y in years]

fig = go.Figure()
fig.add_trace(go.Scatter(x=years, y=black, name="Black",
                        line=dict(color="#e06c4a", width=2.5), mode="lines+markers", marker=dict(size=7)))
fig.add_trace(go.Scatter(x=years, y=hisp, name="Hispanic",
                        line=dict(color="#4ec9a0", width=2.5), mode="lines+markers", marker=dict(size=7)))
fig.add_vline(x=2020, line_dash="dot", line_color="gray")
fig.update_layout(
```

```

axis_title="Year", yaxis_title="Enrolled (millions)",
template="plotly_white", height=380,
legend=dict(orientation="h", y=1.1),
margin=dict(t=40, b=40)
)
fig.show()

```

Unable to display output for mime type(s): text/html

Figure 4.3: Black and Hispanic college enrollment trend, 2019–2024

Black college enrollment rose **+10.3%** year-over-year (2.56M → 2.82M) — the largest single-year increase of any demographic group in the 2024 data. This is worth flagging as both encouraging and analytically suspect.

Encouraging: the 2022 figure (2.37M) represented a multi-year low; 2024’s 2.82M is now above the pre-COVID 2019 level (2.77M).

Analytically suspect: a 10.3% single-year shift from a household survey with ~99,000 records is large. The CPS does not publish standard errors through the API. Without replicate-weight-based variance estimates, it is not possible to determine whether this jump is statistically significant or an artifact of sampling variability in a smaller demographic cell.

## 4.7 Peer Review: What the Release Gets Right and Wrong

### 4.7.1 Strengths of the methodology

**Continuity.** The CPS October supplement has asked comparable questions since 1947. The trend data exists and is publicly accessible. That is unusual and valuable.

**Timeliness.** Results are available within roughly six months of the reference week — faster than the ACS.

**Disaggregation.** The microdata support breakdowns by age, sex, race, Hispanic origin, family income, and school level simultaneously. The API exposes most of this without needing to download the full public-use file.

### 4.7.2 Concerns

**1. October reference week is not enrollment.** The survey asks whether a person is currently enrolled in the week of the survey. Students who enrolled in September and dropped by October are missed. Students who enrolled late are missed. Community college students on non-standard calendars are systematically undercounted. A press release that says “X million are enrolled in college” should say “X million reported being enrolled during the week of October 15–21.”

**2. Self-report with no institutional validation.** CPS enrollment figures are never cross-checked against IPEDS (the Integrated Postsecondary Education Data System), which is an institutional headcount census. IPEDS typically shows 1–2M more college students than CPS, partly due to the reference-period issue and partly due to definitional differences (IPEDS counts

international students on visas; CPS sampling may undercount them). The press release never mentions this discrepancy.

**3. No standard errors reported.** The headline figures in a Census enrollment release are point estimates. The CPS uses a complex sample design with stratification and clustering; naively treating the data as simple random sample understates variance by 30–50% for subgroup estimates. The Bureau publishes replicate weights for proper variance estimation, but they are not accessible through the API. A release that says Black college enrollment rose 10.3% without a confidence interval is incomplete.

**4. Headcounts, not rates.** Reporting enrollment in millions obscures demographic change because the relevant population denominators are shifting. A press release noting that Black college enrollment hit 2.82M should also note the Black 18–24 population size in 2024 vs. 2023. The enrollment *rate* is the more informative statistic.

**5. Part-time decline understated.** Part-time enrollment fell 2.4%. This likely reflects either (a) financial pressure pushing students to full-time to finish faster or (b) part-time students dropping out entirely. These are opposite interpretations. The press release does not investigate.

### 4.7.3 Suggested improvements

Issue	Recommended fix
No uncertainty quantification	Publish 90% confidence intervals alongside point estimates; use BRR replicate weights
No IPEDS cross-check	Include a one-paragraph reconciliation with IPEDS Fall enrollment each year
Headcounts only	Add enrollment <i>rates</i> (enrolled / civilian noninstitutional population by age group)
October snapshot framing	State explicitly that figures reflect enrollment status in one reference week, not full-year enrollment
Gender gap underemphasized	The 57.7/42.3 female/male split warrants a dedicated section, not a footnote

### 4.7.4 Potential follow-on projects

- **Enrollment rates by age group:** pull population denominators from ACS 1-year and compute enrollment rates; cleaner than headcounts for trend analysis
- **CPS vs. IPEDS reconciliation:** compare CPS college headcounts to IPEDS Fall enrollment by year; document and explain the gap
- **Part-time to full-time shift:** disaggregate the full-time/part-time trend by demographic; test whether the shift is concentrated in specific groups
- **Gender gap over time:** the 57.7% female share is the highest on record; build a longer time series and test for structural break around 2010

## 4.8 Summary

The 2024 CPS October School Enrollment data show college enrollment at 17.36M — up 0.7% from 2023 but 4.0% below the 2019 peak. Full-time enrollment grew while part-time declined. Female students now account for 57.7% of college enrollment, with male enrollment down nearly 10% since 2019. Black college enrollment surged 10.3% YoY to 2.82M, recovering above pre-COVID levels, though the absence of standard errors in the API release makes statistical significance uncertain.

The Bureau's methodology is sound for what it measures, but press release framing omits uncertainty quantification, institutional cross-validation, and enrollment rates. The API is well-structured and reproduces the headline figures; the replicate weights needed for proper variance estimation require the full microdata files.

# References



## Appendix A

# Census API Utilities



# Census API Utilities

Reusable functions for pulling and processing Census Bureau data. All functions use `eval: false` — import from this appendix or copy into a note's code cells.

## Core API Wrapper

```
import os
import requests
import pandas as pd

def census_get(dataset, year, variables, geography, api_key=None):
    """
    Pull a Census API table. Returns a DataFrame.

    Parameters
    -----
    dataset      : str      e.g. "acs/acs1", "cps/school/oct", "dec/dp"
    year         : int      e.g. 2024
    variables    : list     e.g. ["NAME", "B15003_001E"]
    geography    : str      e.g. "us:1", "state:*", "county:*&in=state:45"
    api_key     : str      defaults to CENSUS_API_KEY env var

    Returns
    -----
    pd.DataFrame with numeric columns coerced where possible

    Examples
    -----
    # National college enrollment from ACS 1-year
    df = census_get("acs/acs1", 2023, ["NAME", "B14001_008E"], "us:1")

    # State-level K-12 enrollment
    df = census_get("acs/acs5", 2023,
                   ["NAME", "B14001_003E", "B14001_004E"],
                   "state:*")
    """
```

```

key = api_key or os.getenv("CENSUS_API_KEY", "")
base = f"https://api.census.gov/data/{year}/{dataset}"
params = {"get": ",", "join(variables)", "for": geography, "key": key}
r = requests.get(base, params=params, timeout=30)
r.raise_for_status()
data = r.json()
df = pd.DataFrame(data[1:], columns=data[0])
# Coerce numeric columns
for col in df.columns:
    if col not in ("NAME", "state", "county", "us"):
        df[col] = pd.to_numeric(df[col], errors="ignore")
return df

def census_variables(dataset, year, filter_str=None):
    """
    List available variables for a dataset/year.

    Parameters
    -----
    dataset      : str    e.g. "acs/acs1"
    year         : int
    filter_str   : str    optional substring filter on variable name or label

    Returns
    -----
    pd.DataFrame with columns: name, label, concept, predicateType
    """
    url = f"https://api.census.gov/data/{year}/{dataset}/variables.json"
    r = requests.get(url, timeout=30)
    r.raise_for_status()
    raw = r.json()["variables"]
    rows = [{"name": k, **{f: v.get(f, "") for f in
        ["label", "concept", "predicateType"]}}
        for k, v in raw.items()]
    df = pd.DataFrame(rows).sort_values("name").reset_index(drop=True)
    if filter_str:
        mask = (df["name"].str.contains(filter_str, case=False) |
            df["label"].str.contains(filter_str, case=False) |
            df["concept"].str.contains(filter_str, case=False))
        df = df[mask].reset_index(drop=True)
    return df

def census_groups(dataset, year):
    """
    List available variable groups (table prefixes) for a dataset/year.

```

```

"""
url = f"https://api.census.gov/data/{year}/{dataset}/groups.json"
r = requests.get(url, timeout=30)
r.raise_for_status()
groups = r.json().get("groups", [])
return pd.DataFrame(groups)[["name", "description"]].sort_values("name")

```

## ACS Education Variables (Reference)

```

# Commonly used ACS education enrollment groups
ACS_EDUCATION = {
    # School enrollment by level (universe: population 3+ years)
    "B14001": "School enrollment by level of school",
    # B14001_001E Total
    # B14001_002E Enrolled in school
    # B14001_003E Enrolled in nursery school, preschool
    # B14001_004E Enrolled in kindergarten
    # B14001_005E Enrolled in grade 1 to grade 4
    # B14001_006E Enrolled in grade 5 to grade 8
    # B14001_007E Enrolled in grade 9 to grade 12
    # B14001_008E Enrolled in college, undergraduate years
    # B14001_009E Graduate or professional school

    # Educational attainment (population 25+)
    "B15003": "Educational attainment for population 25+",
    # B15003_001E Total
    # B15003_017E Regular high school diploma
    # B15003_022E Bachelor's degree
    # B15003_023E Master's degree
    # B15003_025E Doctorate degree

    # School enrollment by age and sex (more granular)
    "B14003": "Sex by school enrollment by type of school by age",
}

# CPS October supplement - school enrollment variables
CPS_SCHOOL = {
    "PESCHLVL": "School level (1=nursery, 2=K-12, 3=college)",
    "PESCHFT": "Full-time/part-time college enrollment",
    "PEEDUCA": "Highest level of education completed",
    "HEFAMINC": "Family income (recode)",
    "PWSSWGT": "Final person weight (use for population estimates)",
    "PRTAGE": "Age",
    "PESEX": "Sex (1=male, 2=female)",
    "PTDTRACE": "Race (detailed)",
    "PEHSPNON": "Hispanic origin",

```

```
}
```

## Weighted Estimate Helper

```
def weighted_total(df, value_col, weight_col, group_col=None):
    """
    Compute weighted population total from CPS microdata.

    Parameters
    -----
    df          : pd.DataFrame  CPS microdata
    value_col   : str           column to filter on (e.g. "PESCHLVL")
    weight_col  : str           person weight column (e.g. "PWSSWGT")
    group_col   : str           optional groupby column

    Returns
    -----
    pd.Series or float: weighted total(s) in persons (divide by 1000 for thousands)
    """
    if group_col:
        return df.groupby(group_col)[weight_col].sum()
    return df[weight_col].sum()
```