

Accessing Data

There are many different websites and portals for accessing census data. As it's free and in the public domain, organizations take the data and package it in different ways to make it more accessible for certain groups of users and to add additional features, such as derived datasets, data visualizations, and map-based interfaces. Many portals make it easier to access data by filtering it down to certain subsets so that there is less to wade through. The Census Reporter was created by journalists to provide access to just the most recent iteration of the ACS, with convenient search mechanisms, infographics, and the ability to create basic shaded-area maps. The Missouri Census Data Center provides many different applications (for the entire US, not just Missouri). Its data profile tools provide just the five profile tables from the DEC and ACS in a few simple clicks, with the ability to compare up to four geographies. It flags ACS data for reliability based on its CV and provides graphs and charts for many indicators. Many state and local government agencies will repackage data for their jurisdictions on their websites, making it easier for local residents to access. There are also a number of proprietary library databases that focus on providing census and other government datasets, which we'll mention in the next chapter.

Librarians should consider the nature of each researcher's request and their technical capabilities when suggesting a specific tool or website and match the needs of the user to the resource. Some users would benefit from a simple source that entails just pointing and clicking, while others need a fuller range of capabilities and access to a greater number of datasets. It's always best to present users with a range of options so they can choose what works best for them.

In the following sections we will look at data.census.gov and the Census Bureau's API, which are two of the Bureau's primary portals for accessing a full range of datasets. [Data.census.gov](https://data.census.gov) has both basic and advanced features that can satisfy the needs of novice and advanced users, while the API is suitable for users

with a background in scripting or coding. We will briefly mention where you can find data briefs and research reports that summarize population trends.

Open Census Data Portals

Data.census.gov

<https://data.census.gov/cedsci/>

Census Data APIs

<https://www.census.gov/data/developers/data-sets.html>

Missouri Census Data Center

<https://mcdc.missouri.edu/>

Census Reporter

<https://censusreporter.org/>

Also check state and local government sources: regional planning, city planning, economic development, population divisions, labor departments.

Data.census.gov

[Data.census.gov](https://data.census.gov) is the Census Bureau's primary portal for accessing many of its datasets. Launched a few years prior to the 2020 census, it was designed to provide a modern search-based interface for accessing data. The main page has a search box where users can type in a geography, topic, or table number to retrieve results that can be subsequently browsed and filtered. The search works well for casual browsing, and in particular for retrieving profiles on places (lots of data for one place) with helpful infographics.

The search concept has its limits as you are searching across data tables as opposed to documents

and text, which limits the effectiveness of keyword searching. Given the large volume of available data for dozens of datasets for dozens of years, a filter-based approach works best. By clicking on the advanced search option, users can limit the range of datasets to a small, usable selection that can be browsed through and downloaded more easily. Using the advanced search requires knowledge of the datasets and basic census concepts, all of which we have covered thus far. The five advanced search filters are as follows:

1. **Dataset.** Select the specific dataset to eliminate the others. For example, if you want data on educational attainment, that's something that's captured only in the ACS. If you want data for all the counties in a state, many counties have fewer than 65,000 people, so you would need to use the five-year ACS. Then you would choose between Profiles, Subject Tables, or Detailed Tables.
2. **Year.** A simple selection that filters out quite a bit. Choose the latest year if that's what you need. If you want decennial data, then your choice would be either 2020 or 2010. For the ACS, the year represents the one-year series and the last year in the five-year series, so the 2019 ACS refers to the 2019 one-year and 2015–2019 five-year estimates.
3. **Geography.** The nesting rules covered in chapter 3 largely apply here for selecting areas within other areas. Select all counties in the state or nation, or all census tracts in a county or state, or all places in a state, or select specific areas one by one.
4. **Topic.** This is the most open-ended of the options. Try selecting topics that you think are relevant to the specific variables you are looking for. If you are looking for DEC or ACS data profile tables, you can skip this step as there are only a few tables.
5. **Industrial Code.** Important for the business datasets if you want statistics for a particular industry.

Once the filters are selected and applied, an intermediary page appears that provides top results for tables, maps (as basic web mapping capabilities are part of the platform), and web pages on the Census Bureau website. Click right through to the tables to see all the options returned, and from there you can preview each table or start keyword searching if there are too many results. Download the data as a CSV for machine-readable files where each row is a geography and the variables are stored in columns. The CSV can be imported into a spreadsheet, scripting language, or any program. Download the data as an Excel spreadsheet for a human-readable version that resembles what's depicted on the screen. This format is fine for presentation and for looking up statistics, while the CSV is better for data processing, visualization, and analysis.

The ability to filter tables exists at multiple points in the process. You can filter at the beginning by doing

the advanced search. Or, if you do a basic search and click your way through to a table, you can apply the filters on the table results screen. So if you did a basic search for educational attainment and got to a relevant table for the United States, in the table results you could apply a filter for a specific geography and year.

For users who want an exact listing of every single table to definitively evaluate available content, they can go to the individual program web pages for either the DEC or ACS on the Census Bureau's website. In the technical documentation section for each series there are PDF reports as well as Excel spreadsheets that list every table along with its ID number. The basic data.census.gov search works well for known-item searches; enter the table's ID number, move to the table results page, and from there apply filters for geography and year. For smaller datasets like the Population Estimates and Business Patterns, their program websites will have direct links to either spreadsheets or CSV files that you can browse through and download, bypassing data.census.gov altogether.

Census Bureau API

An application programming interface (API) allows users to tap into data repositories through a script or program, which offers a number of benefits. In using a GUI-based portal like data.census.gov, users must manually point and click to retrieve and download data. The download might include multiple tables that would need to be filtered or stitched together. A number of other manual tasks for processing and preparing the data for analysis may follow, likely in a spreadsheet package. Unless the user documented and shared the steps they took, their process would be opaque to subsequent viewers of the data. In contrast, by accessing an API through a script, coders can create highly specific queries to retrieve exactly what they want, and that data is pulled directly into their script where it can be processed, analyzed, or visualized and subsequently output to any number of data formats. The script allows for the automation of processes and serves as documentation that describes exactly how data was retrieved and manipulated. There is a growing need for librarians to become familiar with using APIs to assist researchers with accessing datasets, and this skill is a logical extension to the work many data librarians are increasingly doing (White and Powell 2019; Adams 2018).

There is a learning curve for writing scripts, but once you have a grasp on the basics, the technical process of retrieving data through an API is straightforward. The bigger challenge is having an understanding of census concepts, as these are used for structuring the data and must be understood for making requests. In a REST API, the user builds links

```

# Python Script for Census API
import requests, csv

#Set variables
year='2019'
dsouce='pep'
dname='population'
cols='NAME,POP,DATE_DESC'
state='10'
county='*'
dcode='2,12'
keyfile='census_key.txt'
outfile='pop2019_de_counties.txt'

base_url = f'https://api.census.gov/data/{year}/{dsouce}/{dname}'

#Read api key in from file
with open(keyfile) as key:
    api_key=key.read().strip()

#Retrieve data, print output to screen
data_url = f'{base_url}?get={cols}&DATE_CODE={dcode}&for=county:{county}\
    &in=state:{state}&key={api_key}'
response=requests.get(data_url)
popdata=response.json()
for record in popdata:
    print(record)

#Write data to CSV
with open(outfile, 'w', newline='') as writefile:
    writer = csv.writer(writefile, quoting=csv.QUOTE_ALL, delimiter=',')
    writer.writerows(popdata)

```

to data they wish to retrieve. Specific attributes that describe the data are passed into the link as variables. The program passes that URL to the API to request that specific data. If successful, the API returns the data to the user in a container that can be manipulated in the script. The methods and functions of the specific programming language are used to loop through the data that's returned to extract, process, visualize, or output it to a specific format.

Consider the sample script above, which uses the popular Python language to retrieve recent population estimates for the three counties of Delaware. At the top of the script, two modules are imported that provide additional functionality to the core Python language. The Requests module is popular for working with APIs, while the CSV module is used for manipulating data stored in a plain, delimited-text format. In the next portion of the script, a number of variables are set that will be passed into a URL. Every census program (DEC, ACS, estimates, etc.) has an API, and every iteration and series in each program has a specific web page that documents what its API

provides. These pages describe how the data is structured, illustrate what variables and geographies are available for the given dataset, and provide examples. In this example, the year, dataset (PEP for Population Estimates Program), and a data name that varies with each census program are provided. In the Population Estimates, there are options for total population, components of change, and characteristics. Below the variables, a base URL that has these components is constructed; the variable names that appear in braces will be replaced by the actual variables stored at the top of the script.

More specific variables for the request are defined next. The specific columns to retrieve must appear as one string of text separated by a comma. For the geography, the ANSI/FIPS code that uniquely identifies Delaware is provided, and an asterisk for the county to return all counties in Delaware; alternatively specific counties could be passed as a string of text with the county ANSI/FIPS codes separated by a comma. Counties nest within states, so the request must be made in this manner. The dcode variable is a date

```
# Output from Python Script for Census API
['NAME', 'POP', 'DATE_DESC', 'DATE_CODE', 'state', 'county']
['Kent County, Delaware', '162350', '4/1/2010 population estimates base',
'2', '10', '001']
['Kent County, Delaware', '180786', '7/1/2019 population estimate', '12',
'10', '001']
['New Castle County, Delaware', '538484', '4/1/2010 population estimates
base', '2', '10', '003']
['New Castle County, Delaware', '558753', '7/1/2019 population estimate',
'12', '10', '003']
['Sussex County, Delaware', '197103', '4/1/2010 population estimates base',
'2', '10', '005']
['Sussex County, Delaware', '234225', '7/1/2019 population estimate', '12',
'10', '005']
```

code that is specific to this data file and indicates the years to retrieve; these codes are provided in the online documentation.

Most APIs require you to register with the organization in order to use them, and you must agree to its terms of service. Registering for a census API key is simple and free and can be done from the main census API website. The API key is an alphanumeric string that you must append to your requests to identify that it's coming from you. It's a best practice to not embed an API key in a script, but to store it in a file that you read in so that it's not exposed to others. When uploading scripts into a repository like GitHub, the key file should be ignored so that it is not included in the repository. In this example, the key filename is stored as a variable, and the file is stored in the same folder as the script.

Once the base URL is created, the file is opened and the key is read in. A fuller data URL is built by appending additional variables to it. This isn't strictly necessary; you could build one big link all at once, but this approach makes the code a bit more readable. If this script were longer, this approach would also allow for additional requests without having to re-create the base. Once the variables are passed into the URL, the actual link looks like this:

```
https://api.census.gov/data/2019
/pep/population?get=NAME,POP,DATE
_DESC&DATE_CODE=2,12&for=county:*&in=s
tate:10&key=APIKEYGOESHERE
```

Next, this URL is passed to the internet using the Requests module, and the outcome is saved in a variable called response. If all went well, the response can be read as a JSON object, which is a standard data format used by APIs to return data. The Census Bureau uses a simplified form of JSON, where the data is returned as a nested list, with each sublist

representing a single record and each object in the list as a variable that we requested, separated by a comma. The script loops through the list and prints each record to the screen; the first record has the column headers, while the following records contain the data, in this case one record for each county and year as shown in the example above. Some variables were returned that weren't specifically requested, such as a date description that goes along with the date code, and the state and county ANSI/FIPS codes. At this point, you could loop through these records and manipulate them however you would like. In this example, we loop through the records and write each row out to a CSV file.

This is a minimally working example, and there are countless variations you can employ when writing your own programs. One common addition would be to incorporate protocols for handling errors. With the Requests module you can request status codes to indicate whether the API service is currently available and whether or not a request was correctly formed or allowed prior to processing data. For large requests it's a best practice to incorporate try and exception blocks, where you try to connect and retrieve data but do something else when an error occurs. Larger data requests will require you to loop through and do multiple iterations. To accommodate errors like an interruption in service, you would want to structure your code to hold on to data that has been retrieved and identify where the process has stopped so you can pick up where you left off when launching the script again. The default location for reading and writing files is the folder where the script is stored. Python's OS module can be used for navigating your file system and specifying different input and output locations.

Python is a popular, general purpose, open source scripting language and a good place to start if you are new to coding, but many programming languages can be used to make API requests. The R statistical language is a popular alternative, particularly for

researchers who plan to do statistical analysis. A recent issue of *Library Technology Reports* provides a crash course in R (Glowacka-Musial 2021), and Python and R have been reviewed for their suitability for librarians who wish to expand their programming skills (White and Powell 2019).

While using an API provides obvious benefits, there are certain scenarios where other options would be best. If a particular census table has all of the necessary data for a particular need, it's easier to simply download it from data.census.gov and then read the CSV file into a script for processing. Some datasets, such as the Population Estimates and Business Patterns, provide ready-to-use CSVs and spreadsheets that are straightforward to download and would save the time of writing out a script to do the job. On the other side of the spectrum, if you need a lot of data, like all the DEC or ACS tables for all geographies in a state, the Census Bureau's FTP site allows you to download this data in bulk. You can subsequently load it into a statistical package or relational database.

Reports and Data Summaries

While some users are looking for census data for doing their own analyses and supporting their writing or research, others may be seeking reports that summarize and provide context for the data. What are current population trends in the United States? What is the geographic concentration of different racial and ethnic groups? Are incomes growing or declining? Where are Americans moving? The Census Bureau publishes an extensive series of reports in its online library. Some are recurring series, such as changes in income year by year, while others represent special topics, such as the impact of the Great Recession on school enrollment. Most of the reports are tied to the study of a specific dataset, and you can filter reports by dataset, year, and topic. With each DEC, a series of reports is issued that studies population change, aging, changes in the composition and location of each racial and ethnic group, changing household and family relationships, and a summary of housing characteristics.

In addition to the Census Bureau, a number of nonprofit think tanks such as the Brookings Institute

and the Pew Research Center regularly analyze census data and publish their latest findings in reports and extended blog posts. Demographic research centers at universities across the country will also publish data briefs and technical reports that study either national trends or local ones based on where the college is located.

Census Reports and Analysis

Census Bureau Reports

<https://www.census.gov/library/publications.html>

Brookings Institute: Demographics and Population

<https://www.brookings.edu/topic/demographics-population/>

Pew Research Center

<https://www.pewresearch.org/>

Population Reference Bureau

<https://www.prb.org/what-we-do/focus-areas/u-s-census-american-community-survey-ac/>

Carsey School of Public Policy, University of New Hampshire

<https://carsey.unh.edu/publications>

References

- Adams, Richard Manly, Jr. 2018. "Overcoming Disintermediation: A Call for Librarians to Learn to Use Web Service APIs." *Library Hi Tech*, 36, no. 1: 180–90.
- Glowacka-Musial, Monika. 2021. "Data Visualization with R for Digital Collections." *Library Technology Reports* 57, no 1 (January).
- White, Philip, and Susan Powell. 2019. "Code-Literacy for GIS Librarians: A Discussion of Languages, Use Cases, and Competencies." *Journal of Map and Geography Libraries* 15: 45–67.