Microdata API User Guide

August 22, 2024

TABLE OF CONTENTS

Purpose

The purpose of this guide is to provide an overview of the U.S. Census Bureau's microdata application programming interface (API) and walk through how to make queries. Through this guide, you will learn what the microdata API is, when to use it, and how to use it to get raw data and custom statistics.

What is Public Use Microdata?

Public use microdata is a set of raw data that have not yet been tabulated into a pre-made table. It is when the Census Bureau takes responses received from surveys, removes personal information so you cannot identify any specific person/household, and puts additional edits in place to protect the confidentiality of respondents. Then the Census Bureau provides the set of raw data for you to conduct your own statistical analysis. Working with microdata requires you to select variables, create categories (recode variables), determine output format, and apply statistical weights.

Before using microdata, here are a few tips:

- **Use pre-made statistics through the aggregate data API or pre-made tables on data.census.gov whenever possible:** The data will be more accurate in these statistics aggregated by the Census Bureau compared to a custom table you make. The process to find pre-made tables and statistics is also easier than creating custom statistics through public use microdata.
- **Geographic coverage in microdata is limited:** You can access data for the nation, region, division, and states. A few surveys provide data for public use microdata areas (PUMAs) and selected counties.

What is the Microdata API?

The microdata application programming interface (API) is a data service that allows software developers to access and use Census Bureau microdata within their applications. The service allows users to request results for individual microdata records or create custom statistics.

Like any data from the API, the microdata API is an efficient way to query data directly from Census Bureau servers with many advantages:

- No need to host data on own server
- Connects to statistical software like R and SAS
- Provides immediate access to most current data
- Facilitates building new mobile and Web applications
- Drives interactive data visualizations

Microdata Access on data.census.gov[1] is a prime example of an interactive application powered by the microdata API. All the data in Microdata Access is being requested in real time from the microdata API and fed on your screen in a user-friendly way. This allows you to access the data by navigating an intuitive system rather than making requests directly to the microdata API, which is the focus of this

---

[1] https://data.census.gov/mdat

guide. Please visit the Microdata Access Resources page[2] if you would like to learn how to use Microdata Access on data.census.gov.

In comparison to bulk downloads of microdata from the file transfer protocol (FTP) sites, the microdata API and Microdata Access on data.census.gov provide unique benefits. They both allow you to:

- Create custom statistics without using your own statistical software.
- Download a subset of microdata records instead of the entire dataset. Specifically, you can limit your data requests to get data only for specific variables or geographies. You can also apply universe restrictions prior to downloading the microdata records.

Dataset Types: Aggregate, Microdata, or N/A

The API provides access to many different types of data, and the API Discovery Tool[3] shows the complete list of datasets in the Census Data API.

When you visit the Discovery Tool, browse the column titled "Dataset Type" (see Figure 1). This will tell you whether an API dataset is:

- **Aggregate:** Provides data that the Census Bureau have tabulated to get results for a specific geography.
- **Microdata:** Provides untabulated, raw data. You can use these individual records to create custom statistics using the appropriate weights, universe, and geography.
- **N/A:** A timeseries dataset.

This user guide describes how to use the microdata API. If the statistics you need are available through an aggregate or timeseries dataset, we recommend using those datasets instead of the microdata API. To learn how to use aggregate and timeseries datasets, please see the API User Guide[4].

Title	Description	Vintage	Dataset Name	Dataset Type	Geography List	Variable List	Group List	Examples
American Community Survey: 1-Year Estimates: Data Profiles 1-Year	The American Community Survey (ACS) is a uswide survey designed to provide communities a fresh look at how they are changing. The ACS replaced the decennial census long form in 2010 and thereafter by collecting long form type information throughout the decade rather than only once every 10 years. Questionnaires are mailed to a sample of addresses to obtain information about households -- that is, about each person and the housing unit itself. The American Community Survey produces demographic, social, housing and economic estimates in the form of 1 and 5-year estimates based on population thresholds. The strength of the ACS is in estimating population and housing characteristics. The data profiles provide key estimates for each of the topic areas covered by the ACS for the us, all 50 states, the District of Columbia, Puerto Rico, every congressional district, every metropolitan area, and all counties and places with populations of 65,000 or more. Although the ACS produces population, demographic and housing unit estimates,it is the Census Bureau's Population Estimates Program that produces and disseminates the official estimates of the population for the us, states, counties, cities and towns, and estimates of housing units for states and counties. For 2010 and other decennial census years, the Decennial Census provides the official counts of population and housing units.	2019	acs› acs1› profile	Aggregate	geographies	variables	groups	examples
ACS 1-Year PUMS	The American Community Survey (ACS) Public Use Microdata Sample (PUMS) contains a sample of responses to the ACS. The ACS PUMS dataset includes variables for nearly every question on the survey, as well as many new variables that were derived after the fact from multiple survey responses (such as poverty status).Each record in the file represents a single person, or, in the household-level dataset, a single housing unit. In the person-level file, individuals are organized into households, making possible the study of people within the contexts of their families and other household members. Individuals living in Group Quarters, such as nursing facilities or college facilities, are also included on the person file. ACS PUMS data are available at the nation, state, and Public Use Microdata Area (PUMA) levels. PUMAs are special non-overlapping areas that partition each state into contiguous geographic units containing roughly 100,000 people each. ACS PUMS files for an individual year, such as 2019, contain data on approximately one percent of the United States population.	2019	acs› acs1› pums	Microdata	geographies	variables	groups	examples

Figure 1: Dataset Type Column in the API Discovery Tool

---

[2] https://www.census.gov/data/what-is-data-census-gov/guidance-for-data-users/how-to-materials-for-using-the-microdata-access.html
[3] https://api.census.gov/data.html
[4] https://www.census.gov/data/developers/guidance/api-user-guide.html

Types of Microdata API Queries: Raw Data or Custom Statistics

Depending on the type of API request you use, the microdata API provides results for untabulated records or custom statistics:

- **Raw, Untabulated Records: Use the get parameter *?get=* in your API queries**

  For example, this API query provides the raw data for sex, disability status, and person statistical weight for people without health insurance in Maryland, Virginia, and the District of Columbia: https://api.census.gov/data/2019/acs/acs1/pums**?get=**SEX,DIS,PWGTP&HICOV=2&for=state:11, 24,51. Step-by-step instructions on how to create this API query are covered later in this guide, but for now, let's focus on the results of the data request.

  ```
  [["SEX","DIS","PWGTP","HICOV","state"],
   ["1","2","88","2","51"],
   ["1","2","30","2","24"],
   ["1","1","68","2","24"],
   ["1","2","12","2","51"],
   ["1","2","13","2","51"],
   ["1","1","30","2","24"],
   ["1","2","89","2","51"],
   ["2","2","29","2","51"]
  ```

  Figure 2: API "Get" Query Results Provide Data for Individual Records
  api.census.gov/data/2019/acs/acs1/pums?get=SEX,DIS,PWGTP&HICOV=2&for=state:11,24,51

  The results show 8,589 records. Each record represents a single person and shows their sex, disability status, statistical weight, health insurance status, and state of residence. For example, you can see the first record of the API query in Figure 2 represents a single person whose characteristics include:
    - Male (SEX=1)
    - No disabilities (DIS=2)
    - Statistical weight of 88 (PWGTP=88)
    - No health insurance (HICOV=2)
    - Lives in Virginia (state=51)

- **Custom Statistics in a Table Format: Use the tabulate parameter *?tabulate=* in your API queries.**

  In your tabulate API queries, you will need to specify whether you want variables to appear in rows **&row+** or columns **&col+**, and you will often want to include a statistical weight in your query.

  For example, this API query provides results for sex by disability status for the population without health insurance in DC, MD, and VA: https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+for&row+SEX &row+DIS&HICOV=2&for=state:11,24,51. Step-by-step instructions on how to create this API

query are covered later in this guide, but for now, let's focus on the results of the data request.

```
[[{"state": "11"}, {"state": "24"}, {"state": "51"},"SEX", "DIS"],
[1591,13598,39616,"1","1"],
[675,8673,25014,"2","1"],
[16761,192542,354846,"1","2"],
[6893,150591,275677,"2","2"]]
```

Figure 3: API "Tabulate" Query Results Provide Data for Custom Statistics
api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+for&row+SEX&row+DIS&HICOV=2&for=state:11,24,51

The results provide a custom table. Each estimate from this table is a weighted tabulation that has been aggregated to represent the population without health insurance (HICOV=2) in each of the three states. From the first row of data in Figure 3, you can see in 2019 there were an estimated:

- o 1,591 men (SEX=1) with a disability (DIS=1) without health insurance in DC (state=11)
- o 13,598 men with a disability without health insurance in MD (state=24)
- o 39,616 men with a disability without health insurance in VA (state=51)

The next row of data in Figure 3 shows these same statistics for women (SEX=2). The third and fourth rows of data show the same information that the first two rows do, except that they represent people without a disability (DIS=2).

You can obtain these same results through Microdata Access on data.census.gov/mdat[5]. The data from the API query are shown below, and the tool also calculates totals.

Universe: **selected geographies:** District of Columbia, Maryland, Virginia; **Health insurance coverage recode (HICOV):** No health insurance coverage

Sex (SEX)	Total	Selected Geographies		
		District of Columbia	Maryland	Virginia
⌄ Total (4)	1,086,477	25,920	365,404	695,153
⌄ Total With a disability (2)	89,167	2,266	22,271	64,630
Male	54,805	1,591	13,598	39,616
Female	34,362	675	8,673	25,014
⌄ Total Without a disability (2)	997,310	23,654	343,133	630,523
Male	564,149	16,761	192,542	354,846
Female	433,161	6,893	150,591	275,677

Figure 4: Microdata Access Results
https://data.census.gov/mdat/#/search?ds=ACSPUMS1Y2019&cv=ucgid&rv=DIS,SEX&nv=HICOV(2)&wt=PWGTP&g=0400000US11,24,51

Terms Relevant to All Types of Microdata API Queries

Here are some core concepts that are key for all microdata API calls for custom statistics and raw data.

Base URL

Begin your API queries with the base URL. You can find it in the API Discovery Tool[6] for your dataset. The example below shows the base URL for the 2019 American Community Survey (ACS) 1-Year Public Use Microdata Sample (PUMS).

https://api.census.gov/data/2019/acs/acs1/pums

Figure 5: Base URL in the API Discovery Tool

The base URL consists of the:

- Host name: **https://api.census.gov/data**
- Vintage: **2019**
- Dataset name: **acs/acs1/pums**

Variables

Variables are the unit of data you are searching for. Each variable provides data for a range of values.

To explore the variables, click the variables link for a dataset in the API Discovery Tool as shown in Figure 6 for the 2019 ACS 1-Year PUMS dataset.

Figure 6: Variables Column in the API Discovery Tool

The API has three types of variables that you can browse by viewing the "Predicate Type" column on the API Variables page for your dataset (see Figure 7). The output of these variables may appear in a variety of alphanumeric and special characters, depending on the variable type:

- **String Variables** – Output may be any combination of letters, numbers, or special characters.
- **Integer (Int) Variables** – Output will be numbers, with or without decimals.
- **Float Variables** – Output will be numbers with decimals.

Census Data API: Variables in /data/2019/acs/acs1/pums/variables

Name	Label	Concept	Required	Attributes	Limit	Predicate Type	Group
ACCESS	Access to the Internet		not required		0	int	N/A
ACR	Lot size		not required		0	int	N/A
ADJHSG	Adjustment factor for housing dollar amounts (6 implied		not		0		N/A

Figure 7: Predicate Type Column in the Variables Page for each API Dataset

When you browse the available variables in the Discovery Tool, click a variable for more details. For example, the screenshot below shows details after clicking HICOV[7], the variable for the health insurance recode (see Figure 8). After clicking the variable, you can see that this variable has a predicate type of int (integer), which tells us that the values will be expressed numerically. In addition, it shows the output values for this variable. A value of 1 refers to people with health insurance coverage, and a value of 2 refers to people without health insurance coverage.

Figure 8: HICOV Variable Link and Details from the API Discovery Tool

Predicates to Restrict Your Universe

Predicates allow you to limit your universe by variable value or geography. They always start with **&**.

---

[7] https://api.census.gov/data/2019/acs/acs1/pums/variables/HICOV.json

Restrict Universe by Variable Value

You can use variable predicates to restrict the universe or population that your results include. Consider these examples from the 2019 ACS 1-Year PUMS[8] microdata API dataset.

Examples to restrict universe by a single variable value:

- *&HICOV=2* – restricts results to include data for the population without health insurance
- *&SCHL=16* – restricts results to include data for the population whose highest level of educational attainment is a high school diploma
- *&NAICSP=928110P2* – restricts results to include data for the population whose primary or most recent job is the United States Air Force

Examples to restrict universe by multiple variable values:

- *&SCHL=16&SCHL=17* or *&SCHL=16,17* – restricts universe to persons whose highest level of educational attainment is a regular high school diploma (SCHL=16) or GED or alternative credential (SCHL=17)
- *&SCHL=16:24* – restricts universe to people with a high school diploma or higher

> (!tip) You can use colons (:) to restrict results by a range of variable values for integer and float variables only. You cannot use colons with string variables.

Examples to restrict universe using variables with decimal values from the December 2020 Current Population Survey (CPS) Basic Monthly[9] microdata API dataset:

- *&PTERNHLY=-0.01* – restricts universe to people who do not have hourly earnings
- *&PTERNHLY=0.01:7.24* – restricts universe to people who earn $0.01 to $7.24 per hour

> (!tip) You must put a zero in front of decimal values between -1 and 1. For example, 0.01 and -0.01. The API query will not work if you try to shorten it to .01 or -.01.

Restrict Universe by Geography

Use geography predicates to specify the geography that your results include. The predicate ***&for*** specifies the primary geographic level. The predicate ***&in*** further restricts results for geographies that are nested within other geographies (example: public use microdata areas within a state).

Examples to get results for a single geography:

- *&for=state:22* – provides results for Louisiana

---

[8] https://api.census.gov/data/2019/acs/acs1/pums.html
[9] https://api.census.gov/data/2020/cps/basic/sep.html

- *&for=county:071&in=state:22* – provides results for Orleans Parish, Louisiana. Please be aware that data are only available for some counties in the microdata API for selected CPS datasets.
- *&for=public%20use%20microdata%20area:02400&in=state:22* – provides results for public use microdata area (PUMA) 02400 in Louisiana

Examples to get results for multiple geographies:

- *&for=state:22,28* – provides results for the states of Louisiana and Mississippi
- *&for=state:** – provides results for all states
- *&for=public%20use%20microdata%20area:02400,02401,02402&in=state:22* – provides results for PUMAs 02400, 02401, and 02402 in Louisiana
- *&for=public%20use%20microdata%20area:*&in=state:22* – provides results for all PUMAs in Louisiana

Browse the [Geographic Reference Files](#)[10] to explore the Federal Information Processing Standards (FIPS) geography codes to use in microdata API queries.

Additional Concepts for Tabulate Queries

In addition to the concepts described in the previous section, there are more concepts that are related exclusively to tabulate *?tabulate=* microdata API queries.

Weight Function

Because the microdata in the API are based on survey samples, you will need to apply a weight in order to create estimates that represent the overall population. Otherwise, the totals you calculate will only represent the people or households in the sample.

Include the weight function **weight(insert weight)** after **?tabulate=** in your API queries. For example, the below query applies the person weight **[PWGTP](#)**[11] to the tabulated variables, SEX and DIS. Including this in the query ensures that results represent the total number of people, and not just the number of people in the sample:
[https://api.census.gov/data/2019/acs/acs1/pums?tabulate=**weight(PWGTP)**&col+for&row+SEX&row+DIS&HICOV=2&for=state:11,24,51](https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+for&row+SEX&row+DIS&HICOV=2&for=state:11,24,51)

The other weight for this dataset is the housing weight [WGPT](#)[12], which weights the statistics to provide data for the number of households.

Determining Which Weight to Use

Each variable in the microdata API has a suggested weight. If you are not sure which weight to use for your API query, check out the suggested weight for each variable in the API call.

---

[10] https://www.census.gov/geographies/reference-files.html
[11] https://api.census.gov/data/2019/acs/acs1/pums/variables/PWGTP.json
[12] https://api.census.gov/data/2019/acs/acs1/pums/variables/WGTP.json

For example, clicking the variable DIS[13] in the 2019 ACS 1-Year PUMS shows the suggested weight is PWGTP, as shown in the screenshot below.

Figure 9: Variable Details for DIS Show Suggested Weight of PWGTP
api.census.gov/data/2019/acs/acs1/pums/variables/DIS.json

When all the variables in your query have the same suggested weight, it is recommended that you apply that weight in your API call.

Unweighted Counts

To get unweighted counts, simply omit the weight function from your API query. This will tell you the sample size that the custom statistics are based on.

Predicates to Place Variables in Columns and Rows

The goal of a tabulate API query is to create custom statistics in a table format. To create these custom statistics, you must specify the format of your results by placing variables in the columns and/or rows of your tabulate request.

Use the predicates **&col+** and **&row+** to place variables in the columns and rows. For example, in this request:

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+SEX&row+DIS**

- **&col+SEX** places the sex variable in the columns
- **&row+DIS** places the disability variable in the rows

You can see this reflected in the results for this API call:

```
[[{"SEX": "1"}, {"SEX": "2"},"DIS"],
[21099249,22181817,"1"],
[140495023,144463434,"2"]]
```

The first row of the API results tells us the format of our custom table. In this layout, the first column provides data for sex code 1 (male), the second column provides data for sex code 2 (female), and the third column indicates the disability code that each row refers to. The first row of data provides results for people with a disability (DIS=1) and the second row of data provides results for people without a disability (DIS=2).

---

[13] https://api.census.gov/data/2019/acs/acs1/pums/variables/DIS.json

The results tell us based on the 2019 ACS 1-Year PUMS, in the U.S. there were approximately:

- 21,099,249 men and 22,181,817 women with a disability
- 140,495,023 men and 144,463,434 women without a disability

Determining How to Place Variables in Columns and Rows

All variables that you place in the columns and rows will be cross tabulated by default. This means that placing the same set of variables in different combinations of columns and rows will provide the same set of statistics in a different layout. In other words, the decision on whether to include a variable in a column or a row is primarily cosmetic and will affect how you read the results, not the statistics themselves.

For example, in the screenshot below, you can see how different ways of placing the variables SEX and DIS in the columns and rows provides the same set of statistics in slightly different layouts.

&col+SEX&row+DIS →
```
[[{"SEX": "1"}, {"SEX": "2"},"DIS"],
[21099249,22181817,"1"],
[140495023,144463434,"2"]]
```

&col+DIS&row+SEX →
```
[[{"DIS": "1"}, {"DIS": "2"},"SEX"],
[21099249,140495023,"1"],
[22181817,144463434,"2"]]
```

&col+DIS&col+SEX →
```
[[{"DIS": "1", "SEX": "1"}, {"DIS": "2", "SEX": "1"}, {"DIS": "1", "SEX": "2"}, {"DIS": "2", "SEX": "2"}],
[21099249,140495023,22181817,144463434]]
```
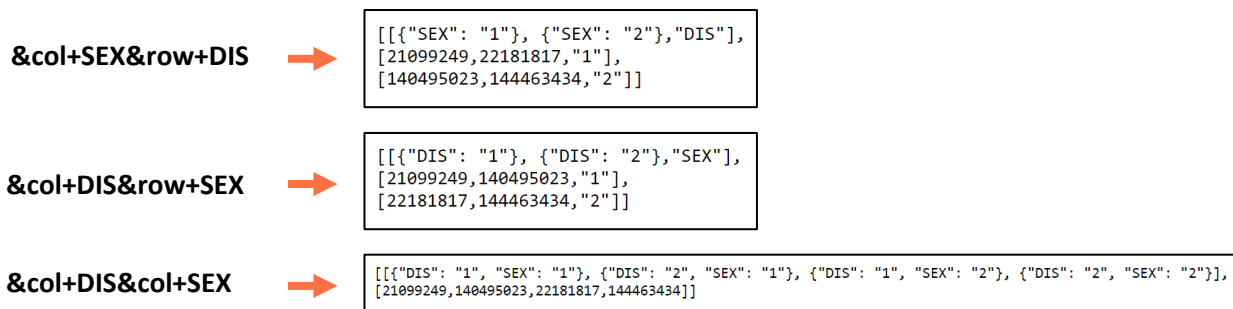
Figure 10: Custom Statistics for Sex by Disability in Different Layout Options
*api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+SEX&row+DIS***
*api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+DIS&row+SEX***
*api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+DIS&col+SEX***

Predicate to Place Geography in Columns or Rows

To get results for more than one geography, you must include the predicate **&col+for** or **&row+for** in your API call. This will place the geographies in the row or column or your table.

For example, when you add **&col+for** to the API query below, it provides data for your geographies in the columns:
https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+for**&row+SEX&row+DIS**&for=state:11,24,51**

The results for this query provide sex by disability data for the District of Columbia (state=11) in the first column, Maryland (state=24) in the second column and Virginia (state=51) in the third column.

```
[[{"state": "11"}, {"state": "24"}, {"state": "51"},"SEX", "DIS"],
[37840,327485,514387,"1","1"],
[46182,377087,544915,"2","1"],
[298100,2601638,3686473,"1","2"],
[323627,2739470,3789744,"2","2"]]
```

If you omit **&col+for** from this query, you will get sex by disability totals for the combined area spanning DC, MD, and VA. It will not provide results for each state individually.

Average Function

The default is that the microdata API will give you results for the total number of people or housing units in your universe. Some variables allow you to also get results for averages by including **avg(insert continuous variable)** in your API query. For example, adding **avg(AGEP)** to this query after the weight allows you to look at the average age of men and women with disabilities compared to those without disabilities:

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP),**avg(AGEP)**&col+SEX&row+ DIS

Be sure to place the average function after the tabulate parameter, using a comma to separate the weight function from the average function.

The microdata API allows you to calculate averages for continuous variables. It does not support use of the average function for categorical variables, because in many cases, the calculation of an average would not make sense or would be inappropriate for these variables.

You may also need to restrict your universe to calculate a meaningful average. Many variables will have values of 0 or -1 to indicate people or households that are not in the universe, and you will likely want to exclude these values from your universe when calculating an average.

Categorical vs Continuous Variables

Variables in the microdata API can be categorical or continuous variables. Categorical variables have a set number of pre-defined data values that are all individually listed in the "Values" section after you click the variable in the Discovery Tool. HICOV[14] is an example of a categorical variable, where a value of 1 represents people with health insurance coverage and a value of 2 represents people without health insurance coverage.

Continuous variables will have numerical output that falls within a specified range of values. You know you are working with a continuous variable anytime you see minimum and maximum numbers defined in the "Values" section after clicking the variable in the Discovery Tool (see Figure 11).

---

[14] https://api.census.gov/data/2019/acs/acs1/pums/variables/HICOV.json

INDP	Industry recode for 2018 and later based on 2017 IND codes
INSP	Fire/hazard/flood insurance (yearly amount, use ADJHSG to adjust INSP to constant dollars)
INTP	Interest, dividends, and net rental income past 12 months (signed, use ADJINC to adjust to constant dollars)
JWAP	Time of arrival at work - hour and minute
JWDP	Time of departure for work - hour and minute
JWMNP	Travel time to work
JWRIP	Vehicle occupancy
JWTRNS	Means of transportation to work
KIT	Complete kitchen facilities
LANP	Language spoken at home

```
{
  "name": "JWMNP",
  "label": "Travel time to work",
  "predicateType": "int",
  "group": "N/A",
  "limit": 0,
  "suggested-weight": "PWGTP",
  "values": {
    "item": {
      "0": "N/A (not a worker or worker
who worked at home)"
    }
    "range": [
      {
        "min": "1",
        "max": "200",
        "description": "1 to 200
minutes to get to work (Top-coded)"
      }
    ]
  }
}
```

Figure 11: JWMNP is a Continuous Variable That is Defined with a Range of Values

JWMNP[15] is an example of a continuous variable in the 2019 ACS 1-Year PUMS dataset that provides data for travel time to work. When you click the variable in the API Discovery Tool, it shows that this variable provides data for a range of values. The minimum value is a 1-minute commute time, and the maximum value is a 200-minute commute time. The details directly above the range highlighted in Figure 11 also show that a value of 0 represents people who are not in the universe for this variable, because they are either not a worker or they work from home.

Recoding Variables

The last consideration of your microdata API query is to determine if you need to recode any variables. This is a process where you can specify custom categories that your statistics represent rather than using the most-detailed, default categories. It is required when working with many continuous variables and is beneficial for categorical variables as well.

For example, suppose you are working with JWMNP[16], a continuous variable in the ACS PUMS microdata API for travel time to work. If you place it in the columns or rows of the API query without recoding it, you will get results for each possible value of the continuous variable. Consider this query that provides travel time to work broken out by sex.

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+SEX&row+JWMNP

After running the query, you will notice the default is to provide results for all 201 possible values of travel time to work in 1-minute increments. It begins with 0 for people who are not a worker or work from home and ends with a 200-minute travel time to work. This is likely more detail than what you need for your analysis. Recoding the variable allows you to define higher-level categories to get results for commuting, such as 15-min increments instead of 1-min increments. For example, this query provides results for the number of males and females with that have commute times within one of

---

[15] https://api.census.gov/data/2019/acs/acs1/pums/variables/JWMNP.json
[16] https://api.census.gov/data/2019/acs/acs1/pums/variables/JWMNP.json

these 8 categories: 0 (not a worker or work from home), 1-14 min, 15-29 min, 30-44 min, 45-59 min, 60-74 min, 75-89 min, or 90+ min

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+SEX**&row+JWMNP_RECODE&recode+JWMNP_RECODE=%7B%22b%22%3A%22JWMNP%22%2C%22d%22%3A%5B%5B00%5D%2C%5B%7B%22mn%22%3A1%2C%22mx%22%3A14%7D%5D%2C%5B%7B%22mn%22%3A15%2C%22mx%22%3A29%7D%5D%2C%5B%7B%22mn%22%3A30%2C%22mx%22%3A44%7D%5D%2C%5B%7B%22mn%22%3A45%2C%22mx%22%3A59%7D%5D%2C%5B%7B%22mn%22%3A60%2C%22mx%22%3A74%7D%5D%2C%5B%7B%22mn%22%3A75%2C%22mx%22%3A89%7D%5D%2C%5B%7B%22mn%22%3A90%2C%22mx%22%3A200%7D%5D%5D%7D**

IMPORTANT: The process to recode variables is complex and is described in detail in the appendix. We are considering ways to simplify the process to recode variables in the microdata API. If we implement a simplified syntax in the future, it will break the current way of recoding variables and users will need to update their microdata API queries to the new syntax for all of their queries that use recoded variables. To stay informed of these important API changes, please sign up for Census Data API email updates[17].

Only recoded continuous variables can be placed in the columns or rows of the microdata API query; otherwise, you will receive an error message. For example, if you try to place household income (HINCP[18]) in the columns/rows without recoding it, you will get an error message:
https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+HINCP

This is done because there is no way to present meaningful information to you without requiring you to define your desired categories up front. In the 2019 ACS 1-Year PUMS, HINCP values can range from a minimum of -$59,999 to a maximum of $9,999,999, which is much more than 1,000 possible values.

You can also recode categorical variables to create higher-level categories instead of getting results for the detailed categories that are provided by default. For example, consider the poverty level of persons variable (PERLIS[19]) from the 2020 Current Population Survey Annual Social and Economic Supplement (CPS ASEC) microdata API dataset. PERLIS provides 5 detailed categories of the poverty level of a person:

- -1 = Not in Poverty Universe
- 1 = Below Poverty Level
- 2 = 100-124 Percent of the Poverty Level
- 3 = 125-149 Percent of the Poverty Level
- 4 = 150 and Above the Poverty Level

You may not need this much detail in your analysis and may be interested in combining some of these categories together. For example, you may want to combine values 2, 3, and 4 together to create a "Not in Poverty" category. You can use the microdata API recode functionality to do this. See the appendix for details.

---

[17] https://public.govdelivery.com/accounts/USCENSUS/signup/15450
[18] https://api.census.gov/data/2019/acs/acs1/pums/variables/HINCP.json
[19] https://api.census.gov/data/2020/cps/asec/mar/variables/PERLIS.json

How To Create Microdata API Queries:

Explore the three exercises below that walk through the steps to create microdata API queries.

- Example 1: How to use the microdata API to get raw, untabulated records.
- Example 2: How to use the microdata API to get custom statistics.
- Example 3: How to use wildcards in the microdata API to get results for multiple geographies.

The functionality shown in each example is applicable across different surveys and programs in the microdata API, including ACS and CPS.

How To Use the Microdata API to Get Raw, Untabulated Data: ?get=

Use these steps to create API calls that provide the raw microdata records for your requested variables. For this example, we will request data for the untabulated records on sex, disability status, and person statistical weight for the population without health insurance in DC, MD, and VA based on the 2019 ACS 1-Year PUMS.

https://api.census.gov/data/2019/acs/acs1/pums?get=SEX,DIS,PWGTP&HICOV=2&for=state:11,24,51

1. Base API URL

   Begin your API queries with the base URL. You can find it in the API Discovery Tool for your dataset. The example below shows the base URL for the 2019 American Community Survey (ACS) 1-Year PUMS.

   https://api.census.gov/data/2019/acs/acs1/pums

Census API: Datasets in /data/2019/acs/acs1/pums and its descendants

Title	Description	Vintage	Dataset Name	Dataset Type	Geography List	Variable List	Group List	Examples	Developer Documentation	API Base URL
ACS 1-Year PUMS	The American Community Survey (ACS) Public Use Microdata Sample (PUMS) contains a sample of responses to the ACS. The ACS PUMS dataset includes variables for nearly every question on the survey, as well as many new variables that were derived after the fact from multiple survey responses (such as poverty status).Each record in the file represents a single person, or, in the household-level dataset, a single housing unit. In the person-level file, individuals are organized into households, making possible the study of people within the contexts of their families and other household members. Individuals living in Group Quarters, such as nursing facilities or college facilities, are also included on the person file. ACS PUMS data are available at the nation, state, and Public Use Microdata Area (PUMA) levels. PUMAs are special non-overlapping areas that partition each state into contiguous geographic units containing roughly 100,000 people each. ACS PUMS files for an individual year, such as 2020, contain data on approximately one percent of the United States population	2019	acs › acs1 › pums	Microdata	geographies	variables	groups	examples	documentation	https://api.census.gov/data/2019/acs/acs1/pums

Figure 12: Base URL Column in the API Discovery Tool for 2019 ACS 1-Year PUMS[20]

2. Add Get Statement
   Add the get function **?get=** to the API query. This specifies that you want raw data for untabulated records.

   https://api.census.gov/data/2019/acs/acs1/pums**?get=**

3. Add Your Variables and Statistical Weights
   Add your desired variables and weights, separated by a comma. In this dataset, **SEX** and **DIS** provide data for sex and disability status. **PWGTP** is the person-level statistical weight for each

---

record. See the "variables" section of the API Discovery Tool for this dataset[21] to explore a list of available variables.

https://api.census.gov/data/2019/acs/acs1/pums?get=**SEX,DIS,PWGTP**

4. Restrict Your Universe
   If you would like to limit your universe, add predicates to restrict your results by variable value. For example, the predicate **&HICOV=2** limits results to only include data for the population without health insurance.

   https://api.census.gov/data/2019/acs/acs1/pums?get=SEX,DIS,PWGTP**&HICOV=2**

5. Add Your Geography
   Add your geography. For example, the predicate **&for=state:11,24,51** limits results to only include data for the District of Columbia, Maryland, and Virginia. See the "examples" section of the API Discovery Tool for this dataset[22] to explore the available geographic levels.

   https://api.census.gov/data/2019/acs/acs1/pums?get=SEX,DIS,PWGTP&HICOV=2**&for=state:11, 24,51**

6. View Results
   The results show 8,589 records. Each record represents a single person and shows their sex, disability status, statistical weight, health insurance status, and state of residence. The first few rows of the output are displayed as follows. You can save these results as a .csv file or consume the data in its original JavaScript Object Notation (JSON) file format.

   ```
   [["SEX","DIS","PWGTP","HICOV","state"],
   ["1","1","57","2","51"],
   ["1","2","118","2","24"],
   ["1","2","114","2","11"],
   ["1","1","76","2","51"],
   ["1","1","73","2","51"],
   ["1","2","46","2","51"],
   ["1","2","51","2","51"],
   ["1","2","78","2","51"],
   ….
   ```

How To Use the Microdata API to Get Custom Statistics: ?tabulate=

Use these steps to create API calls that provide custom statistics in a table format. Many of the steps in tabulate queries are similar to the steps used in get queries, but there are key differences. In this example, we will find data on the total number of males and females by disability status who do not have health insurance in DC, MD, and VA based on the 2019 ACS 1-Year PUMS.

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&col+for&row+SEX&row+DIS&HICOV=2&for=state:11,24,51

---

[21] https://api.census.gov/data/2019/acs/acs1/pums/variables.html
[22] https://api.census.gov/data/2019/acs/acs1/pums/examples.html

1. Base API URL
   Begin your API queries with the base URL. You can find it in the API Discovery Tool for your dataset. The example below shows the base URL for the 2019 American Community Survey (ACS) 1-Year PUMS.

   https://api.census.gov/data/2019/acs/acs1/pums

Figure 13: Base URL Column in the API Discovery Tool for 2019 ACS 1-Year PUMS[23]

2. Add Tabulate Function

   Add the tabulate function **?tabulate=** to the API query. This specifies you want to create custom statistics in a table format.

   https://api.census.gov/data/2019/acs/acs1/pums**?tabulate=**

3. Add Statistical Weight
   Add a statistical weight by including **weight(insert weight)**. In this example, we will add **weight(PWGTP)** to the API query. Including this in the query ensures that results represent the total number of people and not just the number of people in the sample. See the determining which weight to use section of this user guide if you need assistance choosing the desired weight for your dataset.

   If you do not include a weight in your API query you will get results for the unweighted counts. This will tell you the sample size that your custom tabulation is based on.

   https://api.census.gov/data/2019/acs/acs1/pums?tabulate=**weight(PWGTP)**

4. Add Variables to the Rows and/or Columns
   Add variables to your table using the predicates &col+ and &row+ to place variables in the columns and rows of your table. In this example, we will add **&row+SEX** and **&row+DIS** to provide sex by disability status in the rows of our results. See the "variables" section of the API Discovery Tool for this dataset[24] to explore a list of available variables.

   https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&row+SEX&row+DIS**

---

5. Restrict Your Universe
   If you would like to limit your universe, add predicates to restrict your results by variable value. For example, the predicate **&HICOV=2** limits results to only include data for the population without health insurance.

   https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)&row+SEX&row+DIS**&HICOV=2**

6. Add Your Geography to the Universe and Table
   Add the geography in two places in the API query:

   - Add geography to the universe: Use the predicate **&for=**
   - Add geography to the table: Use the predicate **&col+for** or **&row+for**

   For example, the predicate **&for=state:11,24,51** limits results to only include data for DC, Maryland, and Virginia. Add the predicate **&col+for** to get data broken out for each state in the table. If you do not add a predicate to include "for" in the columns or rows of the table, you will only get data for the combined area of DC, MD, and VA rather than getting data for each state separately.
   See the "examples" section of the API Discovery Tool for this dataset[25] to explore the available geographic levels.

   https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&col+for**&row+SEX&row+DIS&HICOV=2**&for=state:11,24,51**

7. View Results

   The first row of the API results shows the format of our custom table. In this layout, the first three columns provide data for DC (state=11), MD (state=24) and VA (state=51) respectively. The 4th column indicates the sex value and the 5th column indicates the disability code.

   > [[{"state": "11"}, {"state": "24"}, {"state": "51"},"SEX", "DIS"],
   > [1591,13598,39616,"1","1"],
   > [675,8673,25014,"2","1"],
   > [16761,192542,354846,"1","2"],
   > [6893,150591,275677,"2","2"]]

   The first row of data shows that based on the 2019 ACS 1-Year PUMS, there were approximately 1,591 men with a disability without health insurance in DC, 13,598 in MD, and 39,616 in VA.

8. Optional: Add Average Function
   If you want results calculated as an average instead of totals, add the average function to the API query. For example, add a comma and **avg(AGEP)** after the weight function to get results for the average age of men and women without health insurance by disability status.

   https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP),**avg(AGEP)**&col+for&row+SEX&row+DIS&HICOV=2&for=state:11,24,51

---

[25] https://api.census.gov/data/2019/acs/acs1/pums/examples.html

[[{"state": "11", "AGEP": "average"}, {"state": "24", "AGEP": "average"}, {"state": "51", "AGEP": "average"},"SEX", "DIS"],
[32,39,43,"1","1"],
[38,49,47,"2","1"],
[34,35,34,"1","2"],
[35,35,34,"2","2"]]

The first row of the data shows that based on the 2019 ACS 1-Year PUMS, the average age for men with a disability that do not have health insurance is approximately 32 in DC, 39 in MD, and 43 in VA.

CPS vs ACS Microdata

Up to this point, most of the examples in this guide come from American Community Survey (ACS) microdata API datasets. The good news is that all the concepts and previous examples apply to microdata from other surveys and programs, such as the Current Population Survey (CPS). To illustrate this point, the next example will use the Current Population Survey Annual Social and Economic Supplement (CPS ASEC) dataset to show additional microdata API functionality: the use of wildcards (*****) to specify multiple geographies.

How To Use Wildcards in the Microdata API

Use these steps to create API calls that use the wildcard functionality to get results for multiple geographies. For this example, we will find the number of self-employed veterans across all states based on the 2020 CPS ASEC.

https://api.census.gov/data/2020/cps/asec/mar?tabulate=weight(MARSUPWT)&col+PEAFEVER&row+for&for=state:*&SEMP_YN=1

1. Base API URL
   Begin your API queries with the base URL. You can find it in the API Discovery Tool for your dataset. The example below shows the base URL for microdata from the 2020 CPS ASEC.

   https://api.census.gov/data/2020/cps/asec/mar

Census API: Datasets in /data/2020/cps/asec/mar and its descendants

Title	Description	Vintage	Dataset Name	Dataset Type	Geography List	Variable List	Group List	Examples	Developer Documentation	API Base URL
Current Population Survey Annual Social and Economic Supplement	The Annual Social and Economic Supplement or March CPS supplement is the primary source of detailed information on income and work experience in the United States. Numerous publications based on this survey are issued each year by the Bureaus of Labor Statistics and Census. A public-use microdata file is available for private researchers, who also produce many academic and policy-related documents based on these data.The Annual Social and Economic Supplement is used to generate the annual Population Profile of the United States, reports on geographical mobility and educational attainment, and detailed analysis of money income and poverty status. The labor force and work experience data from this survey are used to profile the U.S. labor market and to make employment projections.To allow for the same type of in-depth analysis of hispanics, additional hispanic sample units are added to the basic CPS sample in March each year. Additional weighting is also performed so that estimates can be made for households and families, in addition to persons.	2020	cps › asec › mar	Microdata	geographies	variables	groups	examples	documentation	https://api.census.gov/data/2020/cps/asec/mar

Figure 14: Base URL Column in the API Discovery Tool for [2020 CPS ASEC][26]

---

[26] https://api.census.gov/data/2020/cps/asec/mar.html

2. Add Tabulate Function
   Add the tabulate function **?tabulate=** to the API query. This specifies you want to create custom statistics in a table format.

   https://api.census.gov/data/2020/cps/asec/mar**?tabulate=**

3. Add Statistical Weight
   Add a statistical weight by including **weight(insert weight)**. In this example, we will add **weight(MARSUPWT)** to the API query. Including this in the query ensures that results represent the total number of people and not just the number of people in the sample. See the determining which weight to use section of this user guide if you need assistance choosing the desired weight for your dataset.

   If you do not include a weight in your API query you will get results for the unweighted counts. This will tell you the sample size that your custom tabulation is based on.

   https://api.census.gov/data/2020/cps/asec/mar?tabulate=**weight(MARSUPWT)**

4. Add Variables to the Rows and/or Columns
   Add variables to your table using the predicates &col+ and &row+ to place variables in the columns and rows of your table. In this example, we will add **&col+PEAFEVER** to provide veteran status in the columns of our results. See the "variables" section of the API Discovery Tool for this dataset[27] to explore a list of available variables.

   https://api.census.gov/data/2020/cps/asec/mar?tabulate=weight(MARSUPWT)**&col+PEAFEVER**

5. Restrict Your Universe
   If you would like to limit your universe, add predicates to restrict your results by variable value. For example, the predicate **&SEMP_YN=1** limits the results to only include data for the population that is self-employed.

   https://api.census.gov/data/2020/cps/asec/mar?tabulate=weight(MARSUPWT)&col+PEAFEVER **&SEMP_YN=1**

6. Add Your Geography to the Universe and Table
   Add the geography in two places in the API query:

   - Add geography to the universe: Use the predicate **&for=**
   - Add geography to the table: Use the predicate **&col+for** or **&row+for**

   For example, the predicate **&for=state:*** uses a wildcard to include data for all states. Add the predicate **&row+for** to get data broken out for each state in the table. If you do not add a predicate to include "for" in the columns or rows of the table, you will only get data for the combined area covering all states rather than getting data for each state separately.

---

[27] https://api.census.gov/data/2020/cps/asec/mar/variables.html

See the <u>"examples" section of the API Discovery Tool for this dataset</u>[28] to explore the available geographic levels.

https://api.census.gov/data/2020/cps/asec/mar?tabulate=weight(MARSUPWT)&col+PEAFEVER **&row+for**&SEMP_YN=1**&for=state:***

7. <u>View Results</u>
    The first row of the API results tells us the format of our custom table. In this layout, the first three columns provide data for non-veterans (PEAFEVER=2), veterans (PEAFEVER=1), and people not in universe for veteran status (PEAFEVER=-1). The 4th column indicates the two-digit state code.

    > [[{"PEAFEVER": "2"}, {"PEAFEVER": "1"}, {"PEAFEVER": "-1"},"state"],
    > [102606,13940,0,"01"],
    > [33801,4780,793,"02"],
    > [277858,16547,0,"04"],
    > [90825,6489,0,"05"],
    > ….

    The first row of the data shows that based on the 2020 CPS ASEC, there were approximately 102,606 self-employed non-veterans and 13,940 self-employed veterans in Alabama. There were not any self-employed individuals who were not-in-universe for veteran status in Alabama.

Limitations

Please be mindful of the following limitations when working with the microdata API.

Limitations for Get Queries

Queries for raw microdata with more than a few variables may not work unless you also restrict your geography or universe. This is because queries for raw microdata can request thousands or millions of records. When the data request is too large, it can cause the request to fail or time out.

If you are experiencing this issue, we recommend breaking up your API query into multiple API queries that only use a few variables at a time, or download the full set of data from the file transfer protocol (FTP) site for your dataset.

Limitations for Tabulate Queries

Queries for custom statistics may not satisfy all data needs due to some functionality limitations, challenges with the way variables are defined in a survey, and known defects.

- The process to recode variables is complex and subject to change. See the appendix for details.
- The microdata API does not currently have the functionality to:
  - Calculate medians, percentages, or margins of error
  - Apply inflation-adjustment variables like ADJINC to dollar values in ACS datasets
  - Provide data over time in a single API query
  - Calculate an average over time for monthly CPS datasets

---

[28] https://api.census.gov/data/2020/cps/asec/mar/examples.html

- Results are being incorrectly formatted as exponents in API queries that only have variables in the rows. Until this issue is resolved, we recommend placing at least one variable in the columns for all tabulate queries.
- Place of work PUMAs (POWPUMAs), migration PUMAs (MIGPUMAs), and dual-vintage PUMAs are extremely difficult to use in tabulate queries because they are not available as geography predicates. ACS datasets with dual-vintage PUMAs include the 2015, 2014, 2013, and 2012 ACS 5-Year PUMS.

Contact Us

If you have questions about the microdata API, please email us at census.data@census.gov.

Appendix: How To Recode Variables

The microdata API offers the ability to recode variables to create custom categories that your statistics represent, rather than using the most detailed categories that are provided by default. This appendix walks through the process on how to recode variables. For a detailed explanation of why you would want to recode variables, please see the Recoding Variables section of this guide

Please be aware that the process to recode variables in the microdata API is complex and is subject to change in the future. We are considering ways to simplify the process to recode variables in the microdata API. If we implement a simplified syntax in the future, it will break the current way of recoding variables and users will need to update their microdata API queries to the new syntax for all of their queries that use recoded variables. To stay informed of these important API changes, please sign up for Census Data API email updates[29].

The microdata API allows you to recode variables using the format below. The parts of the example in bold will change depending on the variable and values you would like to use in your recode. The remaining characters are required parts of the parameter that you must include every time you recode a variable.

Recode Format

&recode+**NEW_VARIABLE_ NAME**={"b":"**ORIGINAL_VARIABLE** ", "d":[[**CATEGORY 1],[CATEGORY 2]**]]}

In this structure:

- **NEW_VARIABLE_NAME** = A new label you must create for your recoded variable
- b = base variable
- **ORIGINAL_VARIABLE** = The name of the variable you recoded, as it appears on the variables page for a dataset in the microdata API
- d = definition
- **[CATEGORY 1],[CATEGORY 2]** = The definition of the number of categories for your recoded variable along with the values that each category represents

Example: Recoding AGEP to provide data for people age 0-11 and people age 12 and over:

&recode+**AGEGROUPS**={"b":"**AGEP**", "d":[[**0,1,2,3,4,5,6,7,8,9,10,11],[{"mn":12,"mx":99}]**]]}

- **AGEGROUPS** = A new label for the recoded variable
- **AGEP** = The original name of the age variable in the 2019 ACS 1-Year PUMS dataset
- **[0,1,2,3,4,5,6,7,8,9,10,11],[{"mn":12,"mx":99}]** = The definition of the recoded categories and values. This example has two categories: The first provides data for people age 0-11 and the second provides data for people age 12 and over.
- mn = minimum value of range
- mx = maximum value of range

---

[29] https://public.govdelivery.com/accounts/USCENSUS/signup/15450

Making Sense of the Recode Definition

Let's take a deep dive into the rules for defining recoded variables. This will explain why **[0,1,2,3,4,5,6,7,8,9,10,11],[{"mn":12,"mx":99}]** provides data for two categories: people age 0-11 and people age 12 and over.

Defining the number of categories

Square brackets and commas tell the API how many categories to include in a recoded variable.

[CATEGORY 1],[CATEGORY 2]

- Enclose each recoded category in square brackets []
- Separate each category with a comma

Defining values for each category

Individual values or ranges within a square bracket tell the API which values to include in a category.

Example: [0,1,2,3,4,5,6,7,8,9,10,11],[{"mn":12,"mx":99}]

Within each set of square brackets, you must:

- Specify one or more values that you want to include in the recoded category
- Use a comma to separate each value or range of values in your definition

You may define values in a few different ways if you are working with numerical values from integer or float variables. For example, you could create a category for people age 0-11 by:

- **Listing each value individually:** [0,1,2,3,4,5,6,7,8,9,10,11]
- **Defining a range:** [{"mn":0,"mx":11}]
- **A combination of both methods:** [0,1,{"mn":2,"mx":11}]

When defining categories for string variables, list out each value individually. String variables may contain a mixture of alphanumerical characters, which means it is not possible to define a logical range.

You must use straight double quotation marks **"** to enclose all non-numerical text that appears within the curly brackets in the recode parameter.

Encoding Special Characters

Once you recode the variable, you will need to encode the text that appears after the equals sign in the recode parameter. This is needed because some of the special characters such as the brackets and quotes cannot be read by the API. Encoding this part of the API query will replace the special characters with a set of standard characters that the API is able to read to represent the special characters.

Original Text:
&recode+AGEGROUPS=**{"b":"AGEP", "d":[[0,1,2,3,4,5,6,7,8,9,10,11],[{"mn":12,"mx":99}]]}**

Encoded Text:
&recode+AGEGROUPS=**%7B%22b%22%3A%22AGEP%22%2C%20%22d%22%3A%5B%5B0%2C1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%5D%2C%5B%7B%22mn%22%3A12%2C%22mx%22%3A99%7D%5D%5D%7D**

> (!) tip Please be sure to only encode the portion of the recode that appears after the equals sign in the recode parameter. This is the part within and including the curly brackets. If you encode other additional characters, the API query will not work.

In the example above:

Original Text	Encoded Text
{	%7B
}	%7D
[	%5B
]	%5B
"	%22
:	%3A
Blank space	%20

Rather than replacing these characters manually, we recommend using an encoding tool to transform the special characters all at once with a click of a button. Free encoding tools are easy to find on the internet and can complete this process for you very quickly.

Adding the Recode to Microdata API Query

The last step is to add a recoded variable to a tabulate query. Simply add the properly encoded recode parameter to your microdata API query, and the recoded variable to the columns or rows using the new variable name.

For example, suppose you want to look at health insurance status for people age 0-11 compared to people age 12 and over. You will add this to the API query:

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&row+AGEGROUPS**&col+HICOV**&recode+AGEGROUPS=%7B%22b%22%3A%22AGEP%22%2C%20%22d%22%3A%5B%5B0%2C1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%5D%2C%5B%7B%22mn%22%3A12%2C%22mx%22%3A99%7D%5D%5D%7D**

Notice this includes:

- The predicate to add the recoded variable to the rows of the results:
  &row+AGEGROUPS

- The properly encoded recode parameter:
  &recode+AGEGROUPS=%7B%22b%22%3A%22AGEP%22%2C%20%22d%22%3A%5B%5B0%2C1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%5D%2C%5B%7B%22mn%22%3A12%2C%22mx%22%3A99%7D%5D%5D%7D

The results provide a custom table based on the 2019 ACS 1-Year PUMS. The first row of data tells us there were an estimated 2,397,369 children age 0-11 without health insurance, and 45,154,935 children age 0-11 with health insurance. The second row of data provides these results for people 12 and over.

```
[[{"HICOV": "2"}, {"HICOV": "1"},"AGEGROUPS"],
[2397369,45154935,"1"],
[28605927,252081292,"2"]]
```

Additional Recode Examples

Below are some other examples of recoding variables. They show that the recode functionality can be used to recode more than two categories, multiple variables, and different types of continuous and categorical variables.

Travel Time to Work

Recode JWMNP[30] to get data for travel time to work broken out in 8 categories: 0 (not a worker or work from home), 1-14 min, 15-29 min, 30-44 min, 45-59 min, 60-74 min, 75-89 min, or 90+ min

Original Recode Parameter

&recode+COMMUTE={"b":"JWMNP","d":[[00],[{"mn":1,"mx":14}],[{"mn":15,"mx":29}],[{"mn":30,"mx":44}],[{"mn":45,"mx":59}],[{"mn":60,"mx":74}],[{"mn":75,"mx":89}],[{"mn":90,"mx":200}]]}

Encoded Recode Parameter

&recode+COMMUTE=%7B%22b%22%3A%22JWMNP%22%2C%22d%22%3A%5B%5B00%5D%2C%5B%7B%22mn%22%3A1%2C%22mx%22%3A14%7D%5D%2C%5B%7B%22mn%22%3A15%2C%22mx%22%3A29%7D%5D%2C%5B%7B%22mn%22%3A30%2C%22mx%22%3A44%7D%5D%2C%5B%7B%22mn%22%3A45%2C%22mx%22%3A59%7D%5D%2C%5B%7B%22mn%22%3A60%2C%22mx%22%3A74%7D%5D%2C%5B%7B%22mn%22%3A75%2C%22mx%22%3A89%7D%5D%2C%5B%7B%22mn%22%3A90%2C%22mx%22%3A200%7D%5D%5D%7D

API Query

https://api.census.gov/data/2019/acs/acs1/pums?tabulate=weight(PWGTP)**&row+COMMUTE**&col+SEX**&recode+COMMUTE=%7B%22b%22%3A%22JWMNP%22%2C%22d%22%3A%5B%5B00%5D%2C%5B%7B%22mn%22%3A1%2C%22mx%22%3A14%7D%5D%2C%5B%7B%22mn%22%3A15%2C%22mx%22%3A29%7D%5D%2C%5B%7B%22mn%22%3A30%2C%22mx%22%3A44%7D%5D%2C%5B%7B%22mn%22%3A45%2C%22mx%22%3A59%7D%5D%2C%5B%7B%22mn%22%3A60%2C%22mx%22%3A74%7D%5D%2C%5B%7B%22mn%22%3A75%2C%22mx%22%3A89%7D%5D%2C%5B%7B%22mn%22%3A90%2C%22mx%22%3A200%7D%5D%5D%7D**

Output

Results provide the total number of males and females in each of the 8 commuting categories. Specifically, the 8 COMMUTE values represent each of the categories you recoded, and they are ordered the same way you specified them. For example, COMMUTE=1 refers to people with a JWMNP value of 0 (not a worker or work from home) because it was the first category defined in the recode. COMMUTE=2 refers to people who have a 1-14 min travel time to work because it was the second category defined in the recode.

---

[30] https://api.census.gov/data/2019/acs/acs1/pums/variables/JWMNP.json

```
[{"SEX": "1"}, {"SEX": "2"},"COMMUTE"],
[83231328,97001692,"1"],
[18041733,18669298,"2"],
[27137539,25546648,"3"],
[17133339,14231984,"4"],
[7181282,5499790,"5"],
[5010477,3402857,"6"],
[901051,597099,"7"],
[2957523,1695883,"8"]]
```

The second row of the data shows that based on the 2019 ACS 1-Year PUMS, there were approximately 18,041,733 men and 18,669,298 women with commute times of 1-14 min in the United States.

Age and Poverty

Recode poverty by age from the 2020 CPS ASEC microdata:

- Recode A_AGE[31] to get data for 2 age categories: People 0-61 and people 62 and over
- Recode PERLIS[32] to get data for 2 poverty categories: In poverty and not in poverty

Original Recode Parameters

&recode+AGEGROUP={"b":"A_AGE","d":[[{"mn":0,"mx":61}],[{"mn":62,"mx":85}]]}

&recode+POVERTY={"b":"PERLIS","d":[[1],[2,3,4]]}

Encoded Recode Parameters

&recode+AGEGROUP=%7B%22b%22%3A%22A_AGE%22%2C%22d%22%3A%5B%5B%7B%22mn%22%3A0%2C%22mx%22%3A61%7D%5D%2C%5B%7B%22mn%22%3A62%2C%22mx%22%3A85%7D%5D%5D%7D

&recode+POVERTY=%7B%22b%22%3A%22PERLIS%22%2C%22d%22%3A%5B%5B1%5D%2C%5B2%2C3%2C4%5D%5D%7D

API Query

https://api.census.gov/data/2020/cps/asec/mar?tabulate=weight(MARSUPWT)**&col+POVERTY &row+AGEGROUP&recode+AGEGROUP=%7B%22b%22%3A%22A_AGE%22%2C%22d%22%3A %5B%5B%7B%22mn%22%3A0%2C%22mx%22%3A61%7D%5D%2C%5B%7B%22mn%22%3A62 %2C%22mx%22%3A85%7D%5D%5D%7D&recode+POVERTY=%7B%22b%22%3A%22PERLIS%22 %2C%22d%22%3A%5B%5B1%5D%2C%5B2%2C3%2C4%5D%5D%7D**

Output

Results provide the total number of people in poverty and the total number of people not in poverty across both age groups. In this output, the 2 AGEROUP values represent each of the categories you recoded, and they are ordered the same way you specified them. For example, AGEGROUP=1 refers to

---

[31] https://api.census.gov/data/2020/cps/asec/mar/variables/A_AGE.json
[32] https://api.census.gov/data/2020/cps/asec/mar/variables/PERLIS.json

people age 0-61 because it was the first category defined in the recode. AGEGROUP=2 refers to people age 62 and over because it was the second category defined in the recode.

[[{"POVERTY": "1"}, {"POVERTY": "2"},"AGEGROUP"],
[27942761,229606212,"1"],
[6041575,61163321,"2"]]

The first row of the data show that based on the 2020 CPS ASEC, there were approximately 27,942,761 people age 0-61 in poverty and 229,606,212 people age 0-61 not in poverty in the United States.

Hourly Earnings

Recode hourly earnings from the December 2020 CPS Basic Monthly microdata:

- Recode PTERNHLY[33] to get data for 3 wage categories: People with an hourly wage of $0.01-$7.24, people with an hourly wage of $7.25, and people with an hourly wage above $7.25.

Original Recode Parameter

&recode+HOURLYWAGE={"b":"PTERNHLY","d":[[{"mn":0.01,"mx":7.24}],[{"mn":7.25,"mx":7.25}],[{"mn":7.26,"mx":99.99}]]}

> (!) tip Remember to add a leading zero before decimal values between -1 and 1, such as 0.01.

Encoded Recode Parameter

&recode+HOURLYWAGE=%7B%22b%22%3A%22PTERNHLY%22%2C%22d%22%3A%5B%5B%7B%22mn%22%3A0.01%2C%22mx%22%3A7.24%7D%5D%2C%5B%7B%22mn%22%3A7.25%2C%22mx%22%3A7.25%7D%5D%2C%5B%7B%22mn%22%3A7.26%2C%22mx%22%3A99.99%7D%5D%5D%7D

API Query

https://api.census.gov/data/2020/cps/basic/dec?tabulate=weight(PWORWGT)**&col+HOURLYWAGE&recode+HOURLYWAGE=%7B%22b%22%3A%22PTERNHLY%22%2C%22d%22%3A%5B%5B%7B%22mn%22%3A0.01%2C%22mx%22%3A7.24%7D%5D%2C%5B%7B%22mn%22%3A7.25%2C%22mx%22%3A7.26%7D%5D%2C%5B%7B%22mn%22%3A7.2%2C%22mx%22%3A99.99%7D%5D%5D%7D**

Output

Results provide the total number of people in each of the 3 hourly wage categories: $0.01-$7.24 per hour, $7.25 per hour, and above $7.25 per hour. In this output, the 3 HOURLYWAGE values represent each of the categories you recoded, and they are ordered the same way you specified them. For example, HOURLYWAGE=1 refers to people with an hourly wage of $0.01-$7.24 per hour because it was the first category defined in the recode.

---

[33] https://api.census.gov/data/2020/cps/basic/sep/variables/PTERNHLY.json

[[{"HOURLYWAGE": "1"}, {"HOURLYWAGE": "2"}, {"HOURLYWAGE": "3"}],
[656108,152949,76224331]]

The first row of the data shows that based on the December 2020 CPS Basic Monthly, there were approximately 656,108 people with an hourly wage of $0.01-$7.24, 152,949 people with an hourly wage of $7.25, and 76,224,331 people with an hourly wage above $7.25.