

# First Steps with Stata

Frank Donnelly, GIS & Data Librarian, Brown University

Feb 7, 2023

[https://libguides.brown.edu/gis\\_data\\_tutorials/stata](https://libguides.brown.edu/gis_data_tutorials/stata)

---

## Introduction

This tutorial provides a cursory introduction to working with data in Stata, with a focus on navigating the software interface and learning basic commands for loading, describing, and modifying data. The examples were written using Stata SE 17.

### Conventions used in this tutorial:

- Summaries of steps appear in **bold face**.
- Names of windows, tabs, and tools appear in *italic face*.
- Names of files, variables, and commands, and software output appear in typewriter face.
- Commands that you are prompted to type appear in **highlighted typewriter face**.

Before you begin, download the sample data that accompanies this tutorial, from the link at the top of this page. The default working directory for Stata is the documents subfolder, under your username folder. Move the sample data to this folder, and unzip the ZIP file to extract its contents:

**Windows:** Select the ZIP, right click, *Extract All*. Modify the path to unzip directly to C:\Users\yourname\Documents\ to avoid creating a duplicate nested folder. Or, if you have the 7-Zip utility, simply select the ZIP, right click, and choose *7-ZIP - Extract Here*.

**Mac:** Double-click on the ZIP file to open it. By doing so, it automatically extracts.

The sample data comes from two sources, and is stored in the `stata_sample_data` folder in subfolders named for each source. Documentation is included for both datasets.

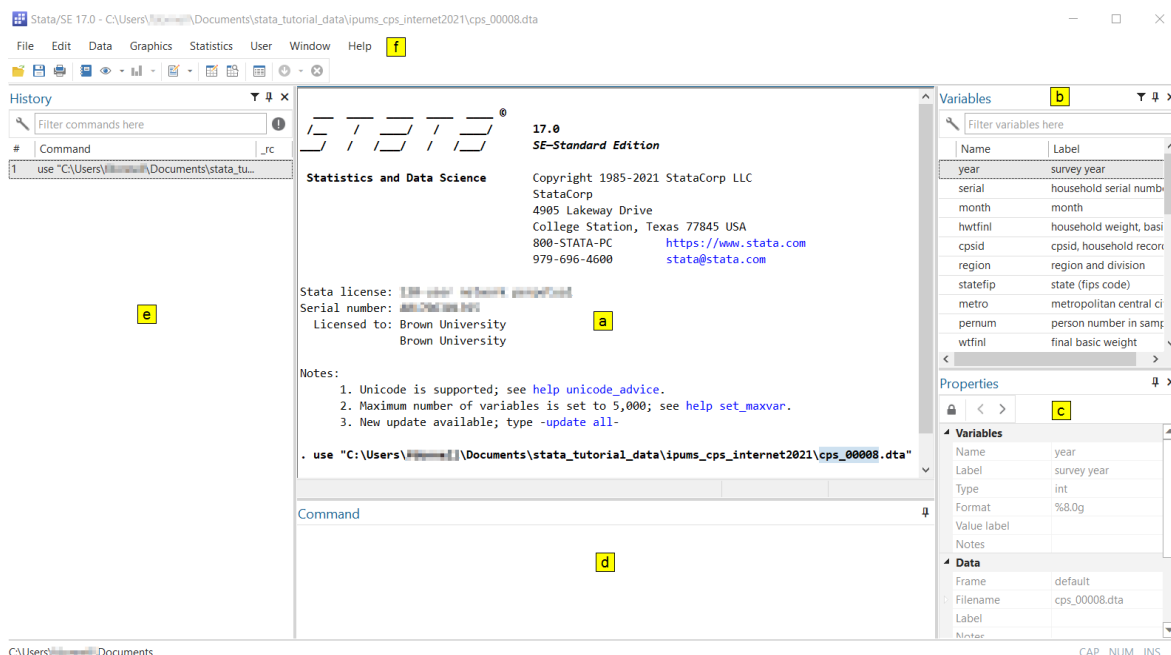
**ipums\_cps\_internet2021:** Microdata samples from the US Census Current Population Survey (CPS), "Computer and Internet Use Supplement" Nov 2021, extracted from IPUMS CPS in a Stata dta format: <https://cps.ipums.org/cps/>.

**acs\_internet\_2021:** State-level summary data from the US Census American Community Survey (ACS), 5-year 2017-2021 series, table B28005 "Age by Presence of Computer and Types of Internet Subscription in Household" in a csv format: <https://data.census.gov/>.

# 1 Basics

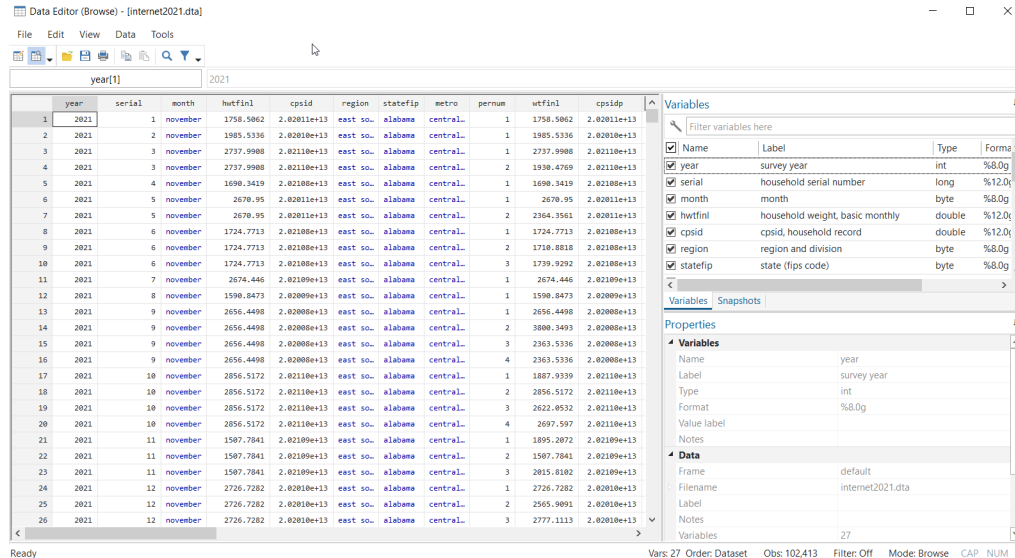
This section provides an overview of the Stata interface, and demonstrates how to load a Stata dta file. We will work with the sample microdata file from the CPS on internet users. Microdata represents individual survey responses.

1. **Launch Stata:** Use the *Start Menu* (in Windows) or the *Apps* menu (on Mac) to launch Stata.
2. **Load a .dta file:** Data (.dta) files are the default file format for Stata. In menu bar at the top, go to *File - Open*, navigate to the `stata_tutorial_data` folder, and in the `ipums_cps_internet2021` subfolder select the `cps_00008.dta` file to open it.
3. **Overview of the Stata interface:**
  - (a) *Results:* displays output of all executed commands.
  - (b) *Variables:* shows names and labels of all variables in a dataset.
  - (c) *Properties:* provides a dataset summary and details for a variable selected in *Properties*.
  - (d) *Commands:* prompt for typing commands, pressing enter runs them.
  - (e) *History:* a running history of all operations performed during the current session; clicking on an entry inserts that command into the *Commands* window.
  - (f) *Menu Bar:* shortcuts for performing several operations (an alternative to typing commands).

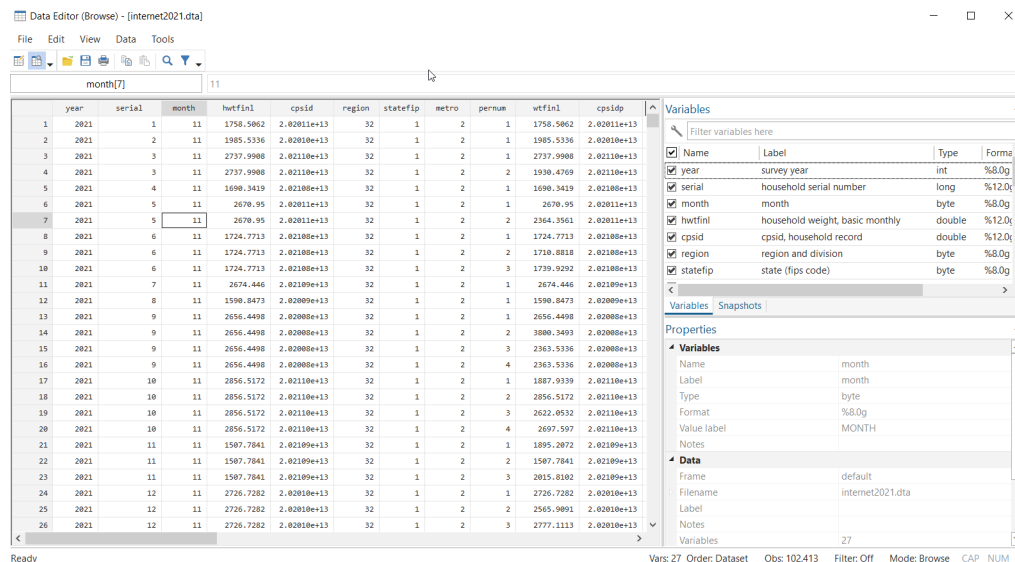


4. **Save a copy of the file:** It's a best practice to copy your original file, and then work on the copy. On the menu bar go to *File - Save As*. Name the file `internet2021` and save it in the `ipums_cps_internet2021` folder. Notice that the path above the menu bar now indicates that you are working in this new copy.

- Open the Data Browser:** To see what the data looks like, on the menu bar go to *Data - Data Editor - Data Editor Browse*. Notice the *Browser* also contains a *Variables* and *Properties* window, which displays variable names, labels, and data types (see the following section for information about data types).



- Viewing labels:** All variables and values have labels, which are human read-able names that provide context. For example, in the *Properties* window the variable *serial* has the label *household serial number*. If you click on any November entry under the *month* column in the table, the actual value of 11 is shown at the top of the table. To display the values instead of the labels, right click on any cell in the table and choose *Data - Value labels - Hide all value labels*. The table now shows the values instead of the labels (when typing commands, you must reference values and not labels). You can repeat the previous operation to un-hide the labels. Close the *Browser* when finished.



## Data Types

Each variable is assigned a data type, which specifies its allowable values and types of operations that can be performed.

**str#:** text string of a specific number of characters, up to a max of approx 2000

**strL:** text string that can store up to 2 bil characters

**byte:** whole number between -100 and 127

**int:** whole number between -32767 and 32640

**long:** whole number between -2.1 bil and 2.1 bil

**float:** a large decimal number

**double:** an enormous decimal number

## 2 Commands

Most Stata users use commands to view, process, and analyze data, rather than relying on the point and click tools in the menus. This section illustrates some of the most fundamental commands for describing, summarizing, and modifying data. Commands in Stata have this basic structure:

```
command varname(s) [if varname==value] [,options]
```

Where the command is the operation you want to perform, followed by the name or names of variables you wish to perform them on. This is followed optionally by an if statement that expresses specific criteria, and by optional qualifiers for specific commands (separated from the rest of the statement with a comma).

Typing the command in the *Command* window and pressing <enter> executes the command. Most commands have an abbreviated or short form, such as `tab` for `tabulate` and `desc` for `describe`. The following examples use the long form.

### Describe Data

1. **Describe:** Type: `describe` to display information about the dataset in the *Results* window. This information is similar to what appears in the *Variables* and *Properties* windows.

```
Observations:      102,413
Variables:         27                               2 Feb 2023 14:12
-----
Variable      Storage   Display   Value
  name         type     format    label      Variable label
-----
year           int       %8.0g          survey year
```

```

serial      long      %12.0g      household serial number
month       byte      %8.0g      MONTH      month
...

```

2. **Describe short:** Type: `describe, short` to display just the observation and summary counts.
3. **Describe variables:** Get a description for specific variables by listing them. Type: `describe age race`.

Variable name	Storage type	Display format	Display label	Value Variable label
age	byte	%8.0g	AGE	age
race	int	%8.0g	RACE	race

4. **Codebook:** Generate a list of all variables, with their values, labels, and counts, by typing: `codebook`.
5. **Variables in codebook:** To print the codebook entry for specific variables, list them. Type: `codebook cinethp`.

```

-----
cinethp                                person accesses internet at home
-----

Type: Numeric (byte)
Label: CINETHP

Range: [1,99]                          Units: 1
Unique values: 3                        Missing .: 0/102,413

Tabulation: Freq.    Numeric  Label
              23,937      1    no
              75,191      2    yes
              3,285      99   niu

```

6. **List:** Print records. For big datasets, limit output to the first ten. Type: `list in f/10`.
7. **List specific variables:** To limit the output list to certain variables, add them to the command. Type: `list statefip age sex race cinethp in f/5`.

```

+-----+
| statefip  age      sex    race    cinethp |
+-----+
1. | alabama   76    female  white     yes |
2. | alabama   68    female  black     yes |
3. | alabama   24     male   black     no  |
4. | alabama   56    female  black     no  |
5. | alabama   80    female  white     yes |
+-----+

```

8. **Add conditions:** To limit results you can add conditions with `if`. Use `&` to express AND and `|` to express OR. Use `==` to express equality and `!=` to express inequality. You need to consult the codebook to identify the values you want to reference. Type:

```
list statefip age sex race cinethp if age <= 18 & statefip == 44
& cinethp != 99.
```

```

| statefip  age      sex    race    cinethp |
+-----+
78550. | rhode is   10     male   white     no  |
78551. | rhode is   13    female  white     no  |
78556. | rhode is    4     male   white     no  |
78585. | rhode is   13    female  white     yes |
78587. | rhode is   10     male   white     yes |
+-----+

```

## Summarize Data

1. **Tabulate:** Summarize a list of variables. The optional `sort` command sorts by frequency. Type: `tabulate cinethp, sort`.

```

person |
accesses |
internet at |
home |      Freq.      Percent      Cum.
-----+-----
yes |      75,191      73.42      73.42
no  |      23,937      23.37      96.79
niu |       3,285       3.21     100.00
-----+-----
Total |     102,413     100.00

```

2. **Cross tabulate:** Add additional variables. Type: `tabulate cinethp sex`.

person				
accesses				
internet		sex		
at home		male	female	Total
-----+-----+-----				
no		11,881	12,056	23,937
yes		36,307	38,884	75,191
niu		1,754	1,531	3,285
-----+-----+-----				
Total		49,942	52,471	102,413

3. **Tabulate with conditions:** You can also specify criteria. Type:  
`tabulate cinethp sex if statefip==44.`

person				
accesses				
internet		sex		
at home		male	female	Total
-----+-----+-----				
no		105	98	203
yes		319	355	674
niu		15	17	32
-----+-----+-----				
Total		439	470	909

4. **Frequencies:** Add the option for row or col to add percentages for rows or columns. Type:  
`tabulate cinethp sex if statefip==44, col`

person				
accesses				
internet		sex		
at home		male	female	Total
-----+-----+-----				
no		105	98	203
		23.92	20.85	22.33
-----+-----+-----				
yes		319	355	674
		72.67	75.53	74.15
-----+-----+-----				
niu		15	17	32
		3.42	3.62	3.52
-----+-----+-----				

Total		439	470		909
		100.00	100.00		100.00

5. **Sum:** For interval-ratio variables, use sum to generate a summary. Type: `sum age`.

Variable		Obs	Mean	Std. dev.	Min	Max
-----+						
age		102,413	41.18927	23.53935	0	85

6. **Sum multiple variables:** Type: `sum age year month`.

7. **Sum with detail:** Add the detail option for more information. Type: `sum age, detail`. Note the 50% value represents the median.

age					
-----					
Percentiles		Smallest			
1%	1	0			
5%	4	0			
10%	9	0	Obs	102,413	
25%	21	0	Sum of wgt.	102,413	
50%	41		Mean	41.18927	
			Std. dev.	23.53935	
		Largest			
75%	61	85			
90%	73	85	Variance	554.1011	
95%	79	85	Skewness	.0080476	
99%	85	85	Kurtosis	1.872457	

## Modifying Data

1. **Create new columns:** Create a new column with a calculated value. Calculate the year each person was born, in a new column called yrborn. Type: `generate yrborn=2021-age`. Then type: `list age sex race yrborn in f/5`.

	age	sex	race	yrborn	
	-----				
1.	76	female	white	1945	
2.	68	female	black	1953	
3.	24	male	black	1997	
4.	56	female	black	1965	
5.	80	female	white	1941	
	+-----+				



2. **Add variable label:** Type: `label variable yrborn "birth year"` to add a label for the new column. Then type: `describe yrborn`.

Variable name	Storage type	Display format	Value Variable label
yrborn	float	%9.0g	birth year

3. **Recode values:** This command allows you to re-classify values, which is often necessary for summarizing data more effectively. It's a best practice to create these in a new variable rather than overwriting an existing one. The notation `#/#` indicates ("values x through y") inclusive. Recode the metro categories into fewer categories. Type: `recode metro (1=1 "Non-metro") (2/4=2 "Metro") (else=9 "Unknown"), generate(newmetro)`. Then type: `codebook newmetro`.

```
(58522 differences between metro and newmetro)
```

```
-----
newmetro      RECODE of metro (metropolitan central city status)
-----
```

```
Type: Numeric (byte)
Label: newmetro
```

```

      Range: [1,9]
Unique values: 3
      Units: 1
Missing .: 0/102,413

```

Tabulation: Freq.	Numeric	Label
18,971	1	Non-metro
82,508	2	Metro
934	9	Unknown

4. **Replace values:** This command allows you to simply replace values in-place. For example (don't type the following), the syntax is: `replace var1=newvalue if var1==oldvalue`.
5. **Renaming and dropping variables:** The syntax for renaming: `rename varname, newname`, and dropping: `drop varname`.
6. **Dropping records:** Can be accomplished with the `drop / if` commands, if a variable is or is not equal to something, greater or less than, etc. Syntax: `drop if varname [criteria]`.

### 3 Saving Output

You begin a particular session when you launch Stata. As you modify a data file, you can save it along the way. When you end a session, the output in the *Results* window is lost unless you explicitly capture it. Here are a few options for saving summary output, but first - some general commands to keep in mind:

- `pwd` prints your current working directory, which by default is your username/documents folder on your computer. When outputting files, if you specify a file name but no path, your output will be stored in this location. You can change the working directory by going to *File - Change working directory*.
- `cls` clears the contents of the *Results* window.
- `clear` clears the current dataset that is in memory.
- *Copy / Paste*: Manually copy information from your *Results* window, and paste it into a document.
- *Translate*: You can output all content that's currently in the *Results* window, and save it to a text file using `translate @results myfile.txt`.
- *Log*: You can turn on the log, which records all actions that you take in a session. `log using mylog.txt`. When you're finished recording, `close log`.

### 4 Do Files

Rather than typing commands one by one, you can compile them into a script and run them as a batch program. In Stata these scripts are called Do files, which are a series of Stata commands saved in a plain text document. Stata has a built-in Do file editor that we will use, but Do files can be created in any plain text editor. The example illustrates several methods for adding comments to files (notes that are not executed), as well as triple forward slashes that can break long commands across multiple lines for readability, without impacting execution (remember, the <enter> key indicates that a line should be executed).

1. **Open the Do-File Editor:** Go to *Window - Do-File Editor - New Do-file Editor*.
2. **Open a Do file:** Go to *File - Open*, navigate to the `stata_tutorial_data` folder and `ipums_cps_internet2021` subfolder, and select `region_summary.do` to open it.
3. **Inspect the Do file:** The Do file is printed below. It loads our IPUMS CPS internet dta file, recodes the region column (which actually contains the nine Census divisions of the US) into the four Census regions, computes summaries of internet users by region, writes the results out to a text file, and saves the dta file. It also uses the weights in the CPS file to calculate total population estimates (sample weights are used to indicate how many cases an individual case would represent in a total population).

```

/* Create four census regions from census divisions and summarize internet
users at home from the 2021 CPS from IPUMS */

* Load the file, use forward slashes in paths
cls
use stata_tutorial_data\ipums_cps_internet2021\internet2021.dta, clear

* Create new column called region4
recode region (11/12=1 "Northeast") (21/22=2 "Midwest") ///
(31/33=3 "South") (41/42=4 "West") (97=9 "Unknown"), generate(region4)

* Data summaries
codebook region4
tabulate region4
tabulate region4 cinethp, row
tabulate cinethp region4 [iweight=cisupwt] // use weight to create pop est

/* Print info in results window to file (saved in the default
directory user/documents)*/

translate @Results region4_summary.txt

save internet2021.dta

```

4. **Run the Do file:** Go to *Tools - Execute*. Close the editor and return to the main Stata interface to see the result. You can also go outside of Stata into your file system, and in your username/documents folder click on the region4\_summary.txt file to open it and view the summary.
5. **Did the Do file fail to run?:** Verify that the Do file refers to the same file name as your dta file, and is stored in the same folder location specified in the path. Alternatively, you could remove the use command, and the Do file will attempt to run on the file you have currently loaded in the session. Likewise, you could remove the save command, and save the file manually once the operation is complete.
6. **Alternative for executing Do files:** If you don't need to edit or see a Do file but simply want to execute one, you can do that by going to *File - Do*.

## 5 Importing Data

While many large data repositories and government surveys publish data in a Stata .dta format, this is relatively uncommon on smaller platforms and sites geared to the general public. These sources will typically provide Excel spreadsheet files, or plain text formats where records are stored on

individual lines, and variables are separated by a character called a delimiter. CSV files (comma separated values) are a common delimited text format.

We will import a csv file of state-level summary data on internet access from the Census ACS. Most of the tables that you download from data.census.gov will come with two header rows: one with variable names and another with labels. We cannot import both, so in the example below we will illustrate how to skip the label row. If you are ever working with csv data that has just a single header row, you can disregard that step.

This section concludes with another example of a Do file. When working with data that's not initially packaged in a .dta format, you will often have to clean and reorganize it prior to use.

1. **Import delimited text data:** Go to *File - Import - Text data*. For *File to import*, hit the folder button and browse to the stata\_tutorial\_data folder, acs\_internet2021 subfolder, and select ACSDT5Y2021.B28005-Data.csv.
2. **Specify import parameters:** This csv file has two header rows; we are only allowed to have one. Change *First row variable names* to *Always*. On the right, click *Override row and column ranges...*, specify that the *First* rows should be 3 and hit *OK*. This tells Stata to start reading from the 3rd row where the data begins, and allows it to properly identify data types for the variables.

File to import: C:\Users\... \Documents\stata\_tutorial\_data\acs\_internet2021\ACSDT5Y2021.B28005-Data.csv

Delimiter: Automatic

First row as variable names: Always

Variable name case: Lower

Floating point precision: Use default float

Text encoding: Automatic ISO-8859-1

Quote stripping: Automatic

Quote binding: Loose

Treat sequential delimiters as one: ☒

Override row and column ranges... \*

Numeric parsing rules

Use locale based parsing for numbers: ☒

Locale: English (United States)

Override decimal separator:

Override grouping separator:

Preview:

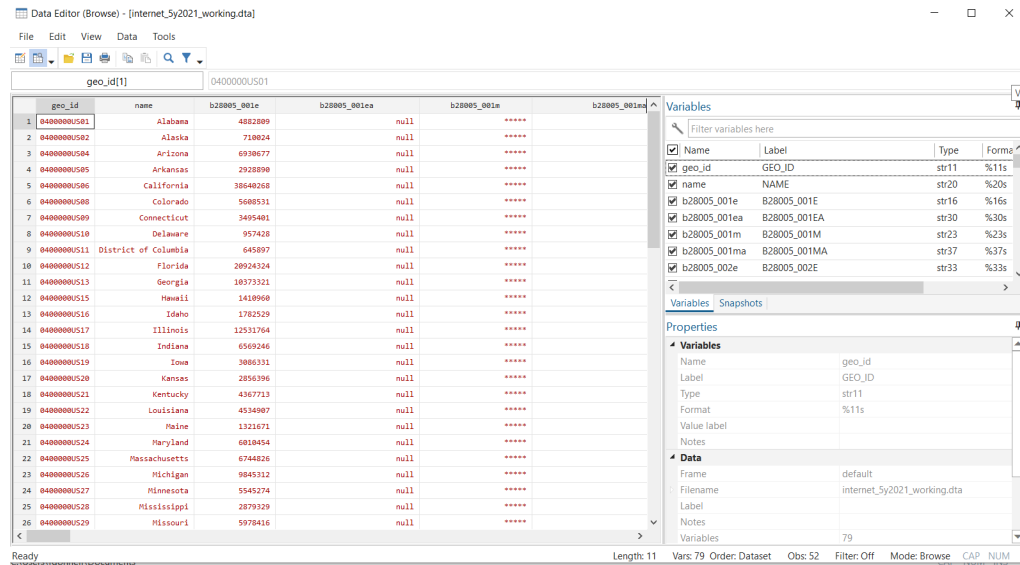
#	geo_id	name	b28005_00...	b28005_001...	b28005_001...	b28005_001m...	b28005_00...	b28	Variable	Type
3	0400000US01	Alabama	4882809 null	*****	*****	*****	1120374 null	geo_id	str	
4	0400000US02	Alaska	710024 null	*****	*****	*****	182238 null	name	str	
5	0400000US04	Arizona	6930677 null	*****	*****	*****	1611069 null	b28005_001e	numeric	
6	0400000US05	Arkansas	2928890 null	*****	*****	*****	703613 null	b28005_001ea	str	
7	0400000US06	California	38640268 null	*****	*****	*****	8971663 null	b28005_001m	str	
8	0400000US08	Colorado	5608531 null	*****	*****	*****	1259951 null	b28005_001ma	str	
9	0400000US09	Connecticut	3495401 null	*****	*****	*****	742807 null	b28005_002e	numeric	
10	0400000US10	Delaware	957428 null	*****	*****	*****	206985 null	b28005_002ea	str	
11	0400000US11	District of Columbia	645897 null	*****	*****	*****	124488 null	b28005_002m	numeric	
12	0400000US12	Florida	20924324 null	*****	*****	*****	4251158 null	b28005_002ma	str	
13	0400000US13	Georgia	10373321 null	*****	*****	*****	2528004 null	b28005_003e	numeric	
14	0400000US15	Hawaii	1410960 null	*****	*****	*****	309496 null			
15	0400000US16	Idaho	1782529 null	*****	*****	*****	457749 null			
16	0400000US17	Illinois	12531764 null	*****	*****	*****	2882403 null			

To change the data type for a column, right-click on the selected column and choose the appropriate type.  
Note: importing string data as numeric can result in loss of data.

OK Cancel Submit

3. **Check other parameters and load:** The *Delimiter* is a comma, so we can leave the automatic setting alone. Likewise, the *Variable case name* is set to *lower*, which is desirable (making it easier to reference variables). Hit *OK* to load the file.
4. **Save the file as a .dta:** Go to *File - Save As*, and in the acs\_internet subfolder save the file as internet\_5y2021\_orig.dta.

5. **Save the file again!:** The file in our previous step is now our original. We don't want to modify it (if we make a mistake, we'd have to repeat the import step). Save a copy of the file as `internet_5y2021_working.dta`.
6. **Open the Data Browser:** Go to *Data - Data Editor - Data Editor Browse*. Unlike our CPS file which represented microdata, this ACS file represents summary data, aggregated by state. There are four columns for each variable that end in a suffix: estimate (e), annotation (a), margin of error(m), and margin of error annotation (ma). Close the browser.



7. **Open Do file.** To demonstrate some clean up steps, we will run a do file to drop the annotation columns (as they contain no data in this example), and add labels for the remaining columns. Go to *Window - Do-File Editor - New Do-file Editor*. Go to *File - Open*, and in the `acs_internet2021` subfolder, select `acs_labels.do` to open it.
8. **Run Do File.** This Do file drops all variables that end in 'ea' and 'ma' using the asterisk character as a wildcard. The label designations were not created by hand; Excel was used to concatenate data in the `ACS_DT5Y2021.B28005-Column-Metadata.csv` file (see the Excel spreadsheet in the project folder that illustrates how this was done). Go to *Tools - Execute* to run the file. Close the editor and return to the main Stata interface to see the result.

describe

Contains data from C:\Users\uname\Documents\stata\_tutorial\_data  
 \acs\_internet2021\internet\_5y2021\_working.dta

Observations: 52  
 Variables: 41 4 Feb 2023 17:07

---

Variable name	Storage type	Display format	Value label	Variable label
------------------	-----------------	-------------------	----------------	----------------

```

-----
geo_id      str11   %11s           Geography
name        str20   %20s           Geographic Area Name
b28005_001e str16   %16s           Estimate!!Total:
b28005_001m str23   %23s           Margin of Error!!Total:
...

```

9. **Save your file:** Go to *File - Save*, and overwrite the current file to save it. This Do file did not explicitly save the file - or load it. By running it, we assumed it would run on the file in memory.
10. **Next steps...:** Return to sections 1 and 2 in this tutorial, and repeat some of those steps on this current ACS file to describe and summarize it.

## Notes about ACS estimates

An example of interpreting an ACS estimate and its margin of error, using data in our sample file: We are 90% confident that the population under 18 in Alaska was 182,238 +/- 272 between the years 2017 and 2021. Alternatively, we could say the under 18 population in Alaska was between 181,966 and 182,510 during this period. When summing ACS estimates across categories or geographies, we can calculate a new margin of error (moe) by taking the square root of the sum of the squares of each moe for the estimates we are combining.

## 6 To Learn More

This tutorial was designed to provide a basic introduction to the Stata interface and commands, primarily for viewing and modifying data. For a fuller treatment, see the following resources.

**Stata User Docs:** The official docs, brief *Getting Started* introductions and the full *User Guide*.  
<https://www.stata.com/features/documentation/>

**Stata Cheat Sheets:** Concise summaries of commands.  
<https://www.stata.com/bookstore/stata-cheat-sheets/>

**GSU Library Stata Guide:** A concise introduction with examples.  
<https://research.library.gsu.edu/Stata>

**UCLA Learning Modules :** A thorough series of tutorials.  
<https://stats.oarc.ucla.edu/stata/modules/>

**Bittmann, Felix.** (2019). *STATA: A Really Short Introduction*. De Gruyter Oldenbourg: Berlin.

**Longest, Kyle C.** (2020). *Using STATA for Quantitative Analysis*, 3rd ed. SAGE Publications: Los Angeles.