Stata How-to:
Load and Save Data

Patrick Blanchenay

2021-12-27

Contents

1. Preliminary: working directory, relative and absolute filepaths

As mentioned in Stata How-To: Get Started , for each Stata project, you should always set a working directory. The project directory acts as the default folder for all operations that involve finding files or saving files (datasets, graphs, etc.) Unless you explicitly specify full file paths, this is where Stata will look for datasets, and where Stata will save what you tell it to save (an output, a graph, a log file, etc.). I recommend that you create a new folder for each Stata project, and set this as the working directory at the beginning of your do-file, using **`cd "C:\...\and\on "`**.

The working directory also serves as the base point for *relative file paths*. Relative file paths are so called because they specify a path relative to the working directory.

Suppose you specified the working directory as `C:\Users\Patrick\myworkingdirectory`, and ask Stata to look for a file `mydataset.dta`; Stata will look for that file into the working directory, that is, `C:\Users\Patrick\myworkingdirectory\mydataset.dta`. If instead you ask Stata to look for a file `original_data\mydataset.dta`; Stata will look into the subolder `original_data` of the working directory, that is, `C:\Users\Patrick\myworkingdirectory\original_data\mydataset.dta`.[1]

If you need Stata to find a file in a folder that is not part of your working directory, you can specify the full file path, also called *absolute file path*. For instance: `C:\Users\Patrick\anotherfolder\dataset2.dta`. If most of your projet file are in your working directory or one of its subfolders, you should not need this.

[1] Because of bad life decisions, Windows computers use backslashes (\) in its file paths, while almost all other operating systems (including MacOS) use forward slashes (/). On Windows, Stata will accept either, so if you share a project with someone working on a Mac, always use forward slashes: `C:/Users/Patrick/mydirectory`.

2. Loading or importing data

2.1. Stata DTA datasets

Datasets in Stata are stored in a proprietary format called DTA, with the `*.dta` extension. Loading a dataset is done with **`use`** followed by the name of the file or its path, between double quotes (" "). For instance:

```
// loading the dataset mydataset.dta from the working directory
use "mydatataset.dta"
```

If you specified the working directory as `C:\Users\Patrick\myworkingdirectory`, the command above tries to load the dataset `C:\Users\Patrick\myworkingdirectory\mydataset.dta`.

2.2. CSV file

Publicly available datasets are often provided in CSV format, with `.csv` extension.[2] To import these, save it somewhere in your working directory, then use **`import delimited`** followed by the name and/or path of the file between double quotes (" "):

```
import delimited "original_data/mydownloadedCSV.csv"
```

2.3. Clearing memory

Stata can only hold one dataset in memory at a time. If you have a dataset already loaded and need to load another one instead (either DTA or CSV), you need to use the **`clear`** option to authorize Stata to clear the memory before loading the new one.

```
sysuse auto, clear
use "mydatataset2.dta", clear
import delimited "myCSVfile.csv", clear
```

If you omit the **`clear`** option, Stata will generate an error and stop, because it always asks for consent before erasing data from memory.

Warning

The option **`clear`** clears whichever data is in Stata's memory. Unless you have saved this data before clearing memory, there is no way to recover it. If you load a dataset, make some modifications to it (generate new variables for instance), then clear memory without saving this modified dataset first, you will have to redo these modifications, recreate those variables, etc. every time you load the dataset (not a problem if all your modifications are recorded in a do-file, right?).

[2] CSV = Comma-Separated Values, a text file that can be read by many programs, including Stata.

2.4. System datasets

Stata includes several example datasets, why can be opened using the **sysuse** command followed by the name of the dataset. These "system" datasets are particularly useful to illustrate how certain commands work; in Stata's official documentation, examples often rely on the "auto" dataset, which contains data on various cars from 1978.

```
// loading the included "auto" dataset
sysuse auto, clear
```

When loading a system dataset, you do not need to have specified a working directory.

3. Saving or exporting data files

3.1. As a Stata DTA file

To save the data *currently in memory* in Stata DTA format you can use the **save** command:

```
save "mydmodifiedata.dta"
```

This will save the data in the working directory, in my case it would create the file `C:\Users\Patrick\myworkingdirecto mymodifieddata.dta`.

> **Warning**
>
> Never, ever, *ever* over-write your original dataset. If you must save a dataset after manipulating it, use a different filename.

3.2. Exporting to CSV or Excel format

We can also export the data to a comma-delimited CSV file using **export delimited** followed by **using** and the filename (or file path) to which you wish to save:

```
export delimited using "mydata.csv"
```

To export to an Excel XLSX or XLS file:

```
export excel using "mydata.xlsx", sheet("Sheet1") ///
        firstrow(variables)
```

where the **sheet("Sheet1")** option specifies that the data should be outputted to the "Sheet1" worksheet and **firstrow(variables)** tells Stata to place the variable names in the first row. You can change that option by **firstrow(varlabels)** if you want to use the variable labels (easier to read, but longer, and more difficult to import back into Stata).

3.3. Overwriting existing files

Stata will prevent you to overwrite existing files, unless you specify the **replace** option. If you run a do-file several times, you will likely need to include that, since previous runs will have already created the file you are trying to save.

```
save "mydmodifiedata.dta", replace
export excel using "mydata.xlsx", sheet("Sheet1") ///
        firstrow(variables) replace
```

Warning

Excel files cannot be overwritten while they are open. So make sure to close your Excel files before running your do-file.

Good Practice

Use meaningful filenames from the start because nothing is more confusing than `mydata_modified_1final_v3.dta`.

4. Practice

Create a folder on your computer, and save the dataset `CPS_2016.dta` to that folder. Open Stata, and create a new do-file in that folder. In that do-file:

1. ☐ At the top, write a comment to describe what this do-file does.
2. ☐ Set the working directory to the folder you just created.
3. ☐ Load the dataset `CPS_2016.dta`.
4. ☐ Using your file explorer, create a subfolder called `CSVexports` in your working directory; in your do-file, export the dataset as a CSV to that subfolder.
5. ☐ Load the system dataset "auto".
6. ☐ Drop the variable `make` with the command **`drop make`**; save that dataset to your working directory.
7. ☐ Drop the variable `mpg` and overwrite the dataset you had created in the step before.
8. ☐ Create a subfolder called `ExcelExports` in your working directory; export the dataset (without `make` or `mpg`) as an Excel file in that subfolder.
9. ☐ Open the Excel file you just created and without closing it, re-run the do-file. What does Stata say?
10. ☐ Close the Excel file. Re-run the do-file from the start. Do you need to add **`replace`** options anywhere? Do so until your do-file can run entirely from the start without any error message.
11. ☐ Import the CSV file you created in step 4..