Session 2: Datasets in Stata

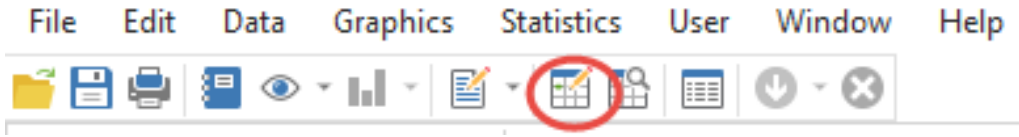
Manuel V. Montesinos

Computation Brush-Up Course
Competition and EPP Master Programs

Fall 2020

Creating a Dataset Manually

Manually type or paste date into the **data editor**.



You can use the **shortcuts**:

- Type `edit` into the command window.
- Press Ctrl+v.
- `browse` does not allow you to alter the data; `edit` does.

You can **rename variables** and write brief descriptions using the Properties Window. Otherwise, type `rename oldvarname newvarname`.

Save your dataset: click on *Save* in the toolbar menu, or press Ctrl+s, or type `save filename` in the command window. If you want to save in a new directory, type `C:/newpath/filename.dta [, replace]`

Using Existing Datasets

You can use the following **shortcuts** to open an existing dataset:

- *File/Open* on the Stata menu bar or *Open* on the toolbar.
- Press Ctrl+o.
- Type `use filename` into the command window (don't forget to clear the previous the dataset).
- Type `use C:/"path..."/filename.dta [, clear]` to open a file outside your working directory.

There are many datasets provided by StataCorp. You can download them by typing `webuse filename`.

Some particular datasets are also provided with the standard Stata package and are saved into your local disk. Type in `sysuse auto` for an example.

Reading Data from Other Sources

Commands to import data in different formats:

- From **Excel**:
  - `import excel filename`
  - Option `firstrow` treats first row of Excel data as variable names.

- From **text files** with clearly defined column delimiters (commas, tabs, semicolons...):
  - `import delimited filename`
  - Option `varnames(n)` treats first n rows as variable names.

Old Stata versions may require different commands.

Missing Values

Missing numeric data are coded using a dot (.).

As a general rule, commands that perform computations of any type handle missing data by omitting those observations.

Missing text data are coded using an empty string (“”).

Take into account that data from external sources may code missing values differently.

Examples of Public Datasets

OECD: data from the Organization for Economic Cooperation and Development for the so-called advanced economies.

Eurostat: provides statistical information to the European Union institutions.

World Development Indicators (World Bank): data for all countries from 1970 to present on agriculture, aid effectiveness, debt, education, energy and mining, environment, finance, health, infrastructure, labor, and poverty.

World Economic Outlook Database (IMF): selected macroeconomic data and projections on national accounts, inflation, unemployment rates, balance of payments, fiscal indicators, trade, and commodity prices. Data available from 1980.

Examples of Public Datasets

United Nations Common Database (UN): selected data series from numerous specialized international data sources for all countries.

Integrated Public Use Microdata Series (IPUMS) – International: 159 samples of census microdata from 55 countries containing records for 325 million individuals.

IPUMS – USA: harmonized data on people in the U.S. census and American Community Survey, from 1850 to the present.

IPUMS – CPS: harmonized data on people in the Current Population Survey, every March from 1962 to the present.

National Bureau of Economic Research (NBER): the NBER Macro History Database contains 3,500 economic time series. The NBER site also provides other datasets.

Exercise

1. Surf the *Database* section in Eurostat (here).

2. Find the table *educ ilev* and download it as a csv file.

3. Import the dataset in Stata.

4. Handle missing values (if needed) and numeric variables stored as strings.

Appending and Merging Datasets

You can use the command `append` to join an existing dataset to the one loaded in Stata:

```
use filename
append using filename2
```

We use `append` when the two datasets have the same set of variables for a different set of observations.

If you want to add variables to a specific dataset from a different file, you can use `merge`:

```
merge 1:1 varname using filename2
merge m:1 varname using filename2
```

`merge` requires that the two datasets have at least one variable in common (`varname`).