Loading data into Stata

Stata's .dta Files

- Data files stored in Stata's format are known as **.dta** files.
  - Remember that scripts of your code are **do-files** and usually have a **.do** extension.
- Also remember: Stata can only store one instance of data. However, you can open multiple instances of Stata. A new instance will open when:
  - You open a new **.dta** file.
  - You open another do-file.
- Be careful of opening too many instances of Stata, as this can get messy quickly.

Loading and Saving .dta Files

- The command `use` loads **.dta** files.
  - Source files may be stored on a hard drive or accessed over the internet (e.g.,by using a web address).
  - Using macros (globals) simplifies `pathing` (see video on Setting up Stata)
  - You have to `clear` the memory before loading a new data set.
- You can also `save` data in Stata's **.dta** format.
  - Stata will not overwrite existing files on its own.
  - The option `replace` will overwrite an existing file with the same name.
- The extension **.dta** may be omitted when using `use` and `save`.

```
use "${data}\auto.dta"
webuse auto, clear
save "${output}\auto_new.dta", replace
```

Importing Excel Data Sets

- Stata is able to load other data formats.
- The command `import excel` is used to import Excel data
- An Excel filename is required (with `path`, if not located in working directory) after the keyword `using`.
- Use the option `sheet()` to open a particular spreadsheet.
- Use the option `firstrow` if variable names are on the first row of the selected spreadsheet.
- Use the option `clear` to overwrite the currently stored data.

```
import excel using "${data}\myfile.xlsx", sheet("mysheet") ///
    firstrow clear
```

Importing .csv Data Sets

- Comma-separated values files are also commonly used.
- Use `import delimited` to read in **.csv** files.
  - Use the option `delimiters(";")` to specify **;** as the delimiter used in the source file
  - Per default, Stata checks for **tab**s (`"\t"`) and **comma**s (`","`)
- Syntax and options are very similar to `import excel`.
  - There is no need for the option `sheet()` here.
  - In **.csv** files the first row usually contains variable names.

```
import delimited using "${data}\myfile.csv", clear
```

Preparing Data for Import

- To get data into Stata cleanly, make sure the data in your Excel file or .csv file have the following properties:
  - Each column (variable) should have the same number of rows (observations)
  - No graphs or summary statistics (means, row totals, etc.)
  - Missing data should be left as blanks or missing data codes, e.g., -999 (see command `mvdecode`)
  - Variable names should contain only alphanumeric characters and, as separators, underscores (`_`) or periods (`.`)
  - Variable names cannot start with a number.
- Try to convert as many variables to numeric as possible to make them applicable for most Stata commands.
- Problems may be handled within Stata, but this may be too difficult for beginners.

No strings attached

- If Stata encounters a problem with the content of a variable, it will import this variable as a `String` even if you want it to be numeric.
- Use **`destring`** to try to convert string variables into numerics.

  `help destring`

- If Stata throws an error message, it is probably because your variable contains some symbols that cannot be converted to numerics directly.
  - E.g., problems may arise if Excel and Stata display decimals (comma vs. dot) differently.
- The option `force` allows you to overwrite the error message. Any erroneous values will be set to **missing** (`.`).

! Using `force` may be useful, but is a potential pitfall if you do not know your data well. Double-check if everything works as intended.