Output presentation

Log files

- The simplest form of preserving your results are so-called **Log**-files.

! We recommend to always use log files for basic output documentation.

- They store everything passed to your output window, e.g., in a `.txt` file.
- You have to initialize a log file by specifying a storage path.

! We already created a log folder in our project folder (see header).

- You can turn your log file **on** and **off** in your code.

! Stata can only open one log file at a time. Thus, always use `capture log close` at the beginning of any do-file.

- Good practice: Log do-files separately.

A make over for your regression output

- Stata's default regression output does not look like anything published in academic papers.
- Use `esttab` to present a nicely formatted table.
- Stata stores estimates in an `e-list`. `esttab` uses these stored values and arranges them for you.
- `esttab` uses the **estout** engine and features basically any formatting option you can think of. You will need to check out the `help` file.
- If you estimate more than one model, store your results by using `eststo model_name` and pass all models to `esttab`.
- You may export tables to `.tex`, `.rtf`, `.csv`, ... files.

! Using the input command in LaTeX allows for an easy integration of your output into your paper. This way, all of our output tables are updated automatically.

Produce publication-style tables

- Let's compare Stata's default regression output with a simple LaTeX adaptation:

```
. regress math c.write i.gender, vce(robust)

Linear regression                               Number of obs     =        200
                                                F(2, 197)         =      72.35
                                                Prob > F          =     0.0000
                                                R-squared         =     0.4190
                                                Root MSE          =     7.1774

------------------------------------------------------------------------------
             |               Robust
        math |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       write |   .6612119   .0549737    12.03   0.000     .5527994    .7696245
             |
      gender |
      female |  -3.770626   1.070417    -3.52   0.001    -5.881573   -1.65968
       _cons |   19.80453   2.771554     7.15   0.000     14.33881    25.27026
------------------------------------------------------------------------------
```

Figure 1: Main specification

	(1) Math	(2) Full
female	-3.771***	-2.024*
	[1.070]	[0.971]
writing score	0.661***	0.389***
	[0.055]	[0.066]
reading score		0.385***
		[0.057]
Constant	19.805***	13.098***
	[2.772]	[2.657]
Observations	200	200
R^2	0.419	0.525

Standard errors in brackets

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

- With just two lines of code, we can make very neat regression tables!

More table!

- **esttab** only works with results stored in an `e()-list`. However, when we use commands such as **summarize** everything is stored in an `r()-list`. So what do we do?
- Use **estpost** to store the results of an `r()-list` in an `e()-list`.
- Afterwards, **esttab** can access these results and display nicely formatted descriptive tables based on **summarize**, **tabulate** and other commands.
- You need to specify the statistics that you want to include into your table using `cell()`.
- Extra brackets in `cell()` make Stata print statistics in wide format instead of below each other.

Ok, if you really don't want to use LaTeX...

- For users of Microsoft Word, there are some neat packages for formatting your output tables and exporting them to Word.
- Install `outreg2` via `ssc install outreg2`.
- `outreg2` also accesses stored estimates, so use the command `eststo` after estimating your models.
- We cannot cover everything here. There is a nice tutorial at thedatahall.com

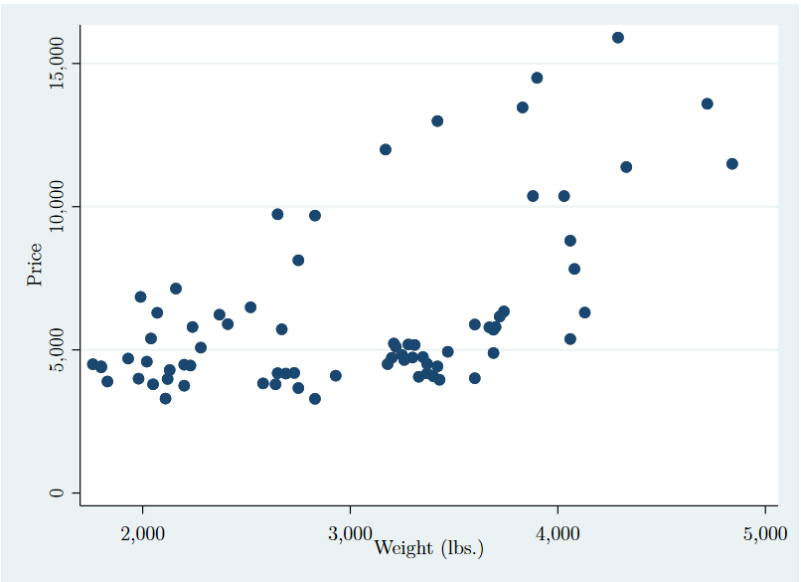
Graphs

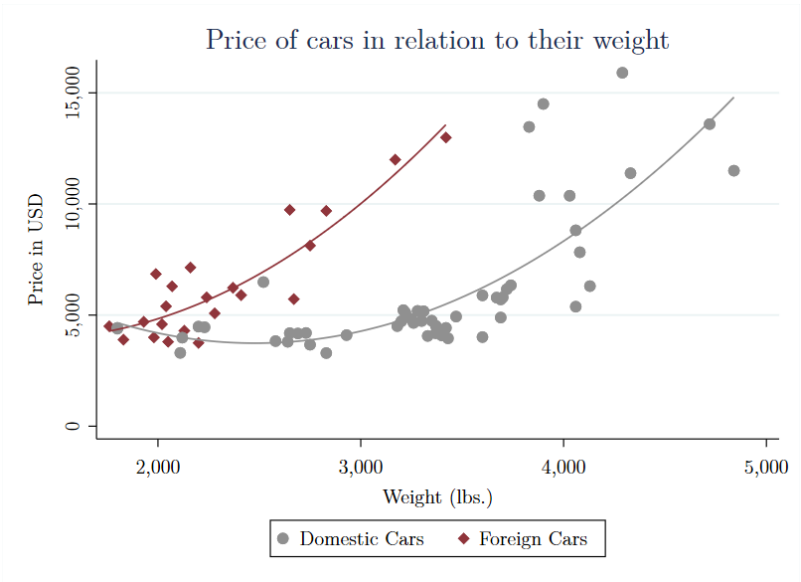
- Stata's tools for graphical visualization are equally powerful and complex.
- You can display almost anything as a `twoway graph`.
- `twoway` lets you combine different plot types, e.g., line plots, bar plots, scatter plots, fitted lines, ... all in one graph.
- You may use **if**-conditions for each plot separately.
- Formatting options include options for...
  - ... changing the looks of individual plots within a graph,
  - ... adjusting the x- and y-axis,
  - ... including text fragments and auxiliary lines,
  - ... setting background colors and adjusting plot margins.

! For bar graphs, Stata offers the `graph bar` command. However, you can display any bar graph as a `twoway bar` when adjusting your data accordingly.

Example twoway graph

- Let's compare Stata's default plot with a formatted graph in which we plot car prices against car weight.





- You will need to work intensively with `help twoway` to design nice graphs.

Marginsplots

- You are able to directly create plots from your **postestimation** analysis by using the command **marginsplot**.

- **marginsplot** uses the results from your prior **margins** command. Any group- and/or value-specific predictions will be passed to **marginsplot**.

- For formatting, the command relies on similar options as **twoway**.

Honorable mentions

- Check out the `help` files of the following non-twoway plot types to check if they are better suited to handle your specific case:
  - `graph bar`
  - `graph pie`
  - `graph dot`
  - `coefplot`
  - `histogram`
  - `sts graph`
  - ...