

ECON-C4100 - Capstone: Econometrics I

Session 1: Stata & \LaTeX

Jaakko Markkanen

Doctoral candidate Jaakko Markkanen

Economics dept. Aalto

jaakko.m.markkanen@aalto.fi

Office hours: on appointment.

Recap: Problem Sets and Exercises

- 5 graded problem sets and 6 exercise sessions.
- Problem sets are published a week before the deadline. All deadlines are before the start of the next exercise session (14:00 EET).
 - First one is out, DL next Friday!
- Points are granted for an honest effort!
- Detailed instructions and deadlines are found on [MyCourses](#).

The objective

- In addition to the theory of econometrics, we want you to learn how to apply what you've learned → empirical exercises.
- This year we also encourage you to learn and master some of the modern technologies used both in data science and in research from the beginning.
- Topics to be covered: \LaTeX , Markdown and outputting results with html-files, using Git and version control in research...

- In short, [Stata](#) is a statistical software and a programming language used widely especially in Economics.
- We use Stata in our empirical exercises and applications during the course but you can also use any other statistical software (like R). However, support is given only for Stata.
- Aalto offers a free Stata licence to all students. You can download the software directly from [this website](#) (Aalto credentials required!).
- It's spelled Stata not STATA!

Why Stata and not R? - The basic answer

- A common question without a definite answer.
- Stata is still the *lingua franca* in empirical Economics. Most of faculty in Aalto and around the world use Stata.
 - "*Because we use it, so should you!*"
 - Lots of built in and user written packages useful in applied econometrics. For instance: [rdrobust](#) and [reghdfe](#)
- In comparison to R, basic regression techniques are relatively easy to learn in Stata.

Why Stata and not R? - The hardcore answer

- Since Stata 16, there is a build in Python integration. In other words, it's Stata on steroids.
 - **Pandas** for data manipulation and Stata for estimation!
 - Most students learn Python in their introduction to programming courses.
- Build in **parallelization** for most of the possible functions.
 - → *much* faster than single core processing.
 - Should not matter on this course though...

Why Stata and not R? - The hardcore answer

- However, R is a great tool: [ggplot2](#) is probably the best graphics package available anywhere.
- For some users who program their own estimators, Stata is not the best choice. However, this is a no-issue for most of us.
- There are lots of MOOCs available both for R and Python/Pandas. For now, we're stuck with Stata.
- Which ever software you choose, the difference is marginal!

- The most important command of Stata is the `help` command. No joke, it opens the Stata documentation.
 - Besides the function options and etc, the Stata documentation offers excellent documentation on statistical concepts. For instance, type in `help regress`.
- As usual, Google search is a great tool. Unorthodox to other programming languages, most online help about Stata are not on [Stack Overflow](#) but on [Statalist](#).



Figure: Why ducks are so great.

- Stata has an easy-to-use GUI for most operations. It also prints the used command which makes the GUI useful in learning Stata.
- However, we recommend you start using Stata directly by writing the code to the command line or most ideally into the do-file editor.
- Do-files are the files where the user's code is written and which Stata executes.

- Some template files are provided on MyCourses.
- Otherwise, the course Stata materials are published [here](#).
- You can request additional Stata topics & tricks to be covered in these sessions.

- L^AT_EX is a typesetting system that makes it easy to to write elegant publication-like documents → widely used in academia.
- More controlled and sophisticated input for equations, graphs and tables than in MS Office.
- Automated managing and formatting of citations and references with with [BibL^AT_EX](#) and [Zotero](#) or [Mendeley](#).
- L^AT_EX takes time to master but it's totally worth it. Start now and you will never regret it!

- There are multiple editors for L^AT_EX. The "best" editor available is [Overleaf](#) which is cloud-based. Aalto offers a premium subscription plan for students who register with their aaltomail.
- Other L^AT_EX desktop-based clients are also available, one example is [Texmaker](#).
- There are various templates available for homework problem sets or thesis documents.
- L^AT_EX is all about flexibility. You can spend a lot of time on formatting margins, fonts, spacing and basically everything. Sticking with the default options can often be the right choice.

Markdown and dynamic documents

- Oftentimes you want to present both your code and your results. How to do this most efficiently?
- The answer: Markdown and dynamic documents.
- Modern IDEs for data scientists allow the user to write both the input and the output directly to html or pdf.
- This is the current best practice to present your empirical work. Example platforms include [Jupyter](#) and [RMarkdown](#).

Markdown and dynamic documents with Stata

- Stata has a built in functionality for creating dynamic Markdown files.
- A do-file template for dynamic html-files is provided on the course webpage.
- You can also use the excellent [Markstat](#) package.
 - Requires some extra work to install but is easier to use notation-wise.
- For the bold ones, there is another way: [using Jupyter with Stata kernel developed by Kyle Barron](#).
 - This allows you to use Stata with [other IDEs](#) than the standard Stata build in do-file editor.
 - Cons: somewhat difficult and prone to unsolvable errors.