

Applied Social Data Science

Coding Camp

Setup • Orientation

Elena Karagianni, PhD Candidate

karagiae@tcd.ie

 September 8–12, 2025

Department of Political Science, Trinity College Dublin

Schedule

10am - 1pm

- Monday: Introduction
- **Tuesday: R Basics + Good practices**

10am - 12pm

- Thursday: Python basics
- Friday: How to write up and report - \LaTeX

Good practices for learning data science

- Aim for steady, constant progress: you will gain familiarity better by doing a little every day.

Good practices for learning data science

- Aim for steady, constant progress: you will gain familiarity better by doing a little every day.
- Learn by doing: code is a language, and it's always easier to learn a language by using it – so find some data that interests you, and put your skills to work.

Good practices for learning data science

- Aim for steady, constant progress: you will gain familiarity better by doing a little every day.
- Learn by doing: code is a language, and it's always easier to learn a language by using it – so find some data that interests you, and put your skills to work.
- Take advantage of online learning resources: sites like Datacamp, Udemy, and Kaggle can help improve your data science skills. Online free-access books too!



Good practices for doing data science

Organize your files!

Use folders to organize your projects. Set a working directory for each project and keep all related files inside it.

R / RStudio

- Set directory: `setwd("path")`
- Check directory: `getwd()`
- Use **Projects** in RStudio to assign directories automatically

Good practices for doing data science

Organize your files!

Use folders to organize your projects. Set a working directory for each project and keep all related files inside it.

R / RStudio

- Set directory: `setwd("path")`
- Check directory: `getwd()`
- Use **Projects** in RStudio to assign directories automatically

Python / Spyder

- Import package: `import os`
- Change directory:
`os.chdir("path")`
- In Spyder: adjust using the file explorer bar (top right)

Work *with* your IDE

Why use an IDE?

IDEs (Integrated Development Environments) make coding easier by improving workflow, efficiency, and organization. Take the time to learn your IDE, and experiment to find one that suits you best.

Work *with* your IDE

Why use an IDE?

IDEs (Integrated Development Environments) make coding easier by improving workflow, efficiency, and organization. Take the time to learn your IDE, and experiment to find one that suits you best.

- Learn your IDE's keyboard shortcuts — they save time and effort

Work *with* your IDE

Why use an IDE?

IDEs (Integrated Development Environments) make coding easier by improving workflow, efficiency, and organization. Take the time to learn your IDE, and experiment to find one that suits you best.

- Learn your IDE's keyboard shortcuts — they save time and effort
- Avoid bad habits, especially relying on automatic settings (e.g. auto-preserve workspace)

Work *with* your IDE

Why use an IDE?

IDEs (Integrated Development Environments) make coding easier by improving workflow, efficiency, and organization. Take the time to learn your IDE, and experiment to find one that suits you best.

- Learn your IDE's keyboard shortcuts — they save time and effort
- Avoid bad habits, especially relying on automatic settings (e.g. auto-preserve workspace)

In general...

Work **with** your IDE, not **for** it. Relying too much on automatic features (e.g. auto-restore workspace) can make your scripts unreliable.

The importance of a good script



Use scripts

Why scripts?

- Record and repeat your analysis
- Share your work with others
- Keep track of progress over time
- Recycle code to save time and effort

Use scripts

Why scripts?

- Record and repeat your analysis
- Share your work with others
- Keep track of progress over time
- Recycle code to save time and effort

A good script...

- begins with the required packages/libraries/modules
- is well-commented to explain the workflow
- is saved in the project's working directory and points there when creating/saving files
- uses meaningful names for objects

Naming objects

Stick to a style!

In any data science project you will create numerous objects: datasets, variables, functions, etc. Consistent naming makes it easier for **you** to revisit old scripts and for **others** to understand your work.

Naming objects

Stick to a style!

In any data science project you will create numerous objects: datasets, variables, functions, etc. Consistent naming makes it easier for **you** to revisit old scripts and for **others** to understand your work.

- `this_is_snake_case`
- `THIS_IS_UPPER_SNAKE_CASE`
- `thisIsCamelCase`
- `ThisIsPascalCase`
- `this-is-kebab-case`

Naming objects

Stick to a style!

In any data science project you will create numerous objects: datasets, variables, functions, etc. Consistent naming makes it easier for **you** to revisit old scripts and for **others** to understand your work.

- `this_is_snake_case`
- `THIS_IS_UPPER_SNAKE_CASE`
- `thisIsCamelCase`
- `ThisIsPascalCase`
- `this-is-kebab-case`

Remember:

Not all case styles will work in all languages. For example,
`dont.do.this.in.python`