From raw data to results you can trust

Programming & Data Management, Part II (Python) · Lecture 1

Magnus Lodefalk
Örebro University

NA446A, Örebro University

Autumn 2026

Today's goals

By the end of today you can:

1. Open a Python project in VS Code that has its own environment, and run a script
2. Organise a project so that someone else can rerun it
3. Load a dataset in pandas and inspect it: size, first rows, summary statistics

From last time

Answer on your own first, then compare with a neighbour

1. In Stata, how do you load a dataset and look at its first five rows?
2. In MATLAB with Kamil, what did a variable such as `x = 5` hold?
3. Where on your computer is the file you worked on last week?

How Part II works

- Four sessions, each the same three steps: a short lecture, a lab, an exercise sheet
- Exercises go from easy to hard; solutions follow three days later
- Everything is on the course site: `magnuslodefalk.com/na446a`
- The written exam tests the course's learning outcomes; the labs are where you practise them

Today's lab: install, open the project, run your first script. Help is in the room.

One recipe, four sessions

- Every session adds one step. Today: set up the project, then load and inspect.

A famous result no one could check

- Reinhart and Rogoff (2010, *AER*): growth falls sharply once public debt exceeds 90% of GDP. Policymakers quoted it in the austerity debate
- 2013: a graduate student, Thomas Herndon, tries to replicate it as homework and cannot
- With the authors' spreadsheet, he finds that the key average used 15 of the 20 countries, plus excluded data and unusual weights
- Corrected, average growth above 90% debt is 2.2%, not -0.1%
- The authors accepted the spreadsheet error but disputed how much it mattered. Either way, no one could check the result until the spreadsheet was shared.

Herndon, Ash and Pollin (2013), PERI Working Paper 322; BBC News, 20 April 2013.

Which regression produced the key result?

Questions every empirical economist has asked at 11 pm:

- Where is the original raw data file?
- How did I define that control variable two years ago?
- Why do the summary statistics look so strange for the outcome?
- Which of the regressions produced the result in the paper?

All four have the same cure: a project anyone can rerun from raw data to results.

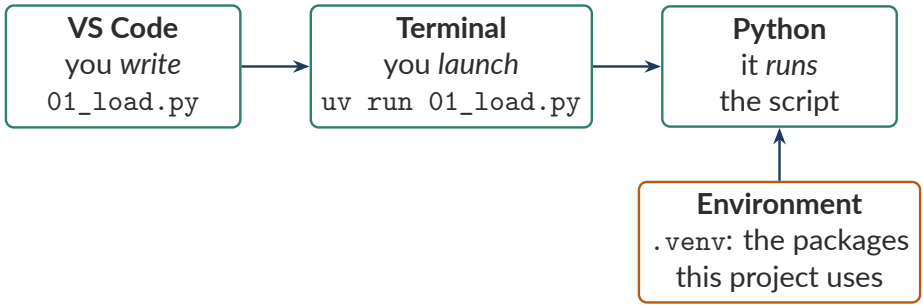
Seven principles for credible empirical work

Today we practise the three in teal

1. **Credibility:** results survive scrutiny; log choices, test robustness
2. **Reproducibility:** same results on rerun; one run-all script
3. **Efficiency:** automate, reuse, refactor
4. **Transparency:** a legible path from data to results
5. **Data integrity:** raw data is read-only; check early
6. **Organisation:** one folder layout: raw, code, modified, output
7. **Version control:** a history you can trust

Consistent with the Data and Code Availability Standard that the AEA follows.

Four tools, one chain



- "ModuleNotFoundError" almost always means the terminal is not using the project's environment.

One folder layout, every project

`data/raw/`	the original data, never edited
`data/processed/`	data your code made
`scripts/`	scripts, numbered in the order they run
`output/`	tables, figures and logs
`README.md`	how to rerun the project
`pyproject.toml`	the package versions
`run_all.py`	runs every script, in order

- Delete `data/processed/` and `output/`, run `run_all.py`: everything comes back. That is the test of a good project.

Check your understanding

Answer on your own first, then compare with a neighbour

You drop three outliers and save the result with `cars.to_csv("data/raw/auto.csv")`. What went wrong?

From Stata commands to Python objects

Stata: one dataset in memory; commands act on it

```
use auto, clear
summarize mpg
```

Python: each dataset is an object; you ask it to act

```
cars = pd.read_csv(path)
cars["mpg"].describe()
```

- You can have many datasets open at once, each with its own name: `cars`, `wages`, `firms`.

Step 1: load the data and check its size

```
from pathlib import Path
import pandas as pd

RAW = Path("data/raw")
cars = pd.read_csv(RAW / "auto.csv")
cars.shape
```

```
(392, 9)
```

`RAW` points to the raw-data folder; `read_csv` reads the file into a DataFrame; `shape` gives (rows, columns).

- 392 cars and 9 variables. Check the size every time you load data: it is the first sign that something went wrong.

Step 2: look at the first rows

```
cars[["name", "mpg", "weight", "origin"]].head(3)
```

```
                          name   mpg  weight  origin
0  chevrolet chevelle malibu  18.0    3504       1
1          buick skylark 320  15.0    3693       1
2         plymouth satellite  18.0    3436       1
```

- The left column is the row index, starting at 0, not 1. Python counts from zero.

Step 3: summary statistics

```
cars[["mpg", "weight"]].describe().round(1)
```

```
          mpg  weight
count   392.0   392.0
mean     23.4  2977.6
std       7.8   849.4
min       9.0  1613.0
50%      22.8  2803.5
max      46.6  5140.0
```

- Same numbers as Stata's `summarize`; the median (50%) comes for free.

Rows 25% and 75% omitted here to fit the slide.

Check your understanding

Answer on your own first, then compare with a neighbour

Predict the result before you run it: `cars.groupby("origin")["mpg"].mean()`. Which origin do you expect to have the highest mean mpg, and how would you write it in Stata?

Five rules for using AI in this course

The same rules in every part of the course

1. **Try first yourself:** write in a comment what the code should do
2. **Use AI to explain, debug or suggest an alternative**, not to write the whole script
3. **Test** what it gives you on a case where you know the answer
4. **Explain** every line, and where the code stops working
5. **Own the result:** mark lines that came from AI; you answer for all of them

- Why: in a trial with about 1,000 students, ChatGPT raised practice grades by 48% but lowered grades on a later exam without AI by 17%, and the students did not notice (Bastani et al., 2025, *PNAS*).

Today's lab

1. Install the tools and open the course project in VS Code *goal 1*
2. Set up the four folders; run today's Steps 1–3 as one script *goal 2*
3. Fix an AI-written script: `lab01_fix_the_bug.py` runs without an error but gives a wrong answer. Find it by testing a case you know *goal 3*

– Step 3 practises rules 3 and 4: code that runs is not code that is right.

Stuck? Paste the course tutor prompt (course site, AI page) into your chatbot first: it gives hints, not answers.

Summary

1. Each project has its own environment; the terminal must use it
2. Raw data is read-only; code writes only to `data/processed/` and `output/`
3. In pandas each dataset is an object: `shape`, `head()`, `describe()`

Before next time: watch Video 4 and read the Turrell chapter (the parts named on the Readings page); try the self-test without AI.

Test yourself before next time

Without AI: this is what you should manage on your own

1. Draw the chain from VS Code to the Python interpreter. Where do packages live?
2. Why is `data/raw/` read-only, and where does cleaned data go?
3. What does `cars.shape` return, and why check it after every load?
4. Write the pandas line that gives the mean weight by origin.

Appendix

When the setup fails

- **“ModuleNotFoundError: pandas”:** the terminal is not in the project’s environment. Activate it, or pick the interpreter ending in `.venv` in VS Code
- **“File not found”:** VS Code is open on the wrong folder. Open the project folder itself, and use paths relative to it
- **Nothing works:** delete the `.venv` folder and type `uv sync` again; for today, work with a neighbour and email me

Stata and Python terms

Python / data science	Stata
DataFrame	the dataset in memory
row, record, instance	observation
column, feature	variable
target	dependent variable
missing value `<NA>`, `NaN`	.
method: `cars.describe()`	command: `summarize`
package: `pip install`	`ssc install`

References and further reading

- Alexander, R. (2013). Reinhart, Rogoff… and Herndon: The student who caught out the profs. *BBC News*, 20 April.
- Bastani, H., O. Bastani, A. Sungu, H. Ge, Ö. Kabakcı and R. Mariman (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. *PNAS* 122(26): e2422633122.
- Bryan, K. (2025). Social Science PhD Tech Stack. Memo, Rotman School, University of Toronto.
- Dreber, A. and M. Johannesson (2025). A framework for evaluating reproducibility and replicability in economics. *Economic Inquiry* 63(2): 338–356.