Python Cheatsheet

MATH60033A · Session 02 · Exploratory data analysis

Thierry Warin, PhD · HEC Montréal · A2026

Everything you need to type, in one place. The first four sections are the same every week; the last one is every name from sessions 1 to 2.

Starting Python in VS Codium

Open the course folder once — **File > Open Folder**, then pick the repository. Everything below assumes you are inside it.

Open a terminal with **View > Terminal** (`Ctrl` + `` ` `` on Windows and Linux, `Cmd` + `` ` `` on macOS). Create the environment the first time only:

```
python3 -m venv .venv      # macOS
python3 -m venv .venv      # Linux
py -m venv .venv           # Windows (PowerShell)
```

Then activate it — **every time you open a new terminal**:

```
source .venv/bin/activate  # macOS
source .venv/bin/activate  # Linux
.venv\Scripts\activate     # Windows (PowerShell)
```

You will see `(.venv)` at the start of the prompt. If you do not, nothing below will find pandas. Install the packages once, with the environment active:

```
pip install -r requirements.txt
```

Finally tell the editor which Python to use: `Ctrl`/`Cmd` + `Shift` + `P`, type **Python: Select Interpreter**, and choose the one inside `.venv`.

A new file, and running it

File > New File, then save it with a `.py` ending — `session02.py`, say. Write your code, save it, and in the terminal:

```
python session02.py     # macOS, Linux and Windows alike, inside .venv
```

Inside the environment the command is `python` on all three platforms. Outside it, macOS and Linux need `python3` and Windows needs `py`, which is one more reason to activate first.

For trying one line at a time, type `python` on its own to get the `>>>` prompt, and `exit()` to leave. The editor's **Run** triangle does the same thing as the command above.

The four things you always do

```
import pandas as pd

# 1 - build a DataFrame
df = pd.DataFrame({"country": ["FR", "DE", "IT"],
                   "gdp_pc_eur": [37_000, 46_000, 32_000]})

# 2 - look at what is in it
df.head()          # the first five rows
df.shape           # (rows, columns)
df.dtypes          # what type each column holds
df.info()          # types, and how many values are missing
```

```
df.describe()      # count, mean, sd and quartiles, per numeric column

# 3 - read a file in
df = pd.read_csv("data.csv")
df = pd.read_excel("data.xlsx")                    # needs openpyxl - see below
df = pd.read_parquet("data/spine/core.parquet")

import qmib
df = qmib.load("core")                             # a course dataset, by name

# 4 - write a CSV out
df.to_csv("out.csv", index=False)   # index=False, or you get a stray column
```

Installing a package you do not have

Everything this course needs is already in `requirements.txt`. When you meet a `ModuleNotFoundError` anyway, or a line above says a reader needs something extra, install it **into the environment** — never outside it:

```
pip install openpyxl                # one package, by name
pip install -r requirements.txt     # or the whole course list again
```

Two rules, and both are about where it lands:

- Your prompt must show `(.venv)` **first**. `pip -V` prints the path it is about to install into; that path has to contain `.venv`.
- Type `pip`, not `pip3`, and never `sudo`. Inside an activated environment `pip` is already the environment's own.

Then check it arrived:

```
pip show openpyxl                   # name, version and location - or nothing at all
python -c "import openpyxl; print(openpyxl.__version__)"
```

What you have learned so far

Sessions 1 to 2, in the order you met them. Every line below has run on the course data.

Session 1 · Syllabus and Europe 2031

The workstation itself — the editor, the environment, and loading a course dataset with one line. That is section 1 above.

Session 2 · Exploratory data analysis

```
# -- Load and shape -----------------------------------------------
core = qmib.load("core")            # a course dataset, by name
core = pd.read_parquet("data/spine/core.parquet")
d = core.dropna(subset=["gdp_pc_eur", "productivity_idx"])
# subset= matters: without it a row missing ANY column goes
both = left.merge(right, on=["geo", "time"], how="inner")
n_rows, n_cols = core.shape

# -- Describe -----------------------------------------------------
gdp.mean(), gdp.median()      # the gap between them is the finding
gdp.std(ddof=1), gdp.var(ddof=1)  # ddof=1 divides by n-1
gdp.quantile([0.25, 0.5, 0.75])    # IQR is the third minus the first
from scipy import stats
stats.trim_mean(gdp, 0.05)        # mean after cutting 5% off each tail
```