# Recap of the Python part

## Programming & Data Management, Part II (Python)

Magnus Lodefalk
*Örebro University*

NA446A, Örebro University

Autumn 2026



# What the exam asks, and how to practise it

| | Part | Minutes |
|---|---|---|
| 1 | the four sessions, one slide each | 6 |
| 2 | what the exam looks like | 2 |
| 3 | three questions: predict on paper, then we trace them | 6 |
| 4 | how to prepare, and your questions | 6 |

- The exam asks you to read code and predict what it does, without a computer. That is what we practise today.

# The four sessions

# Session 1: a project someone else can rerun

- Project layout: raw data in `data/raw/`, never edited; code in `scripts/`; paths built from the project folder (`ROOT / "data" / "raw"`), so the project runs on any computer
- Python basics: `"20" + "24"` joins text; lists count from 0; a slice `x[1:3]` stops *before* position 3
- Objects, not commands: `df.head()` asks the object to act; `df.sort_values("x")` changes nothing until you write `df = df.sort_values("x")`

- In pandas, nothing changes unless you assign (`replace`).

# Session 2: look at the data before you use it

- The intake checklist: type, missing values, range. One text value such as `"****"` makes the whole column text
- Sentinel codes (999, $-9$) are numbers, so `isna()` cannot see them; set them to missing with `.mask()` or `.where()`
- Aggregate: `groupby().agg()` gives one row per group; `groupby().transform()` keeps every row and adds the group value

- The errors that matter most give no error message: check type, missing and range every time.

# Session 3: combine tables without losing rows

- Keys: unique and never missing; the same type in both tables (`1` and `"1"` do not match)
- Merge audit: `how="left"`, `validate="m:1"`, `indicator=True`; predict the number of rows before you run
- Wide and long: `melt` gives one row per unit and column melted; years taken from column names are text. A lag is `groupby("pid")["x"].shift(1)`

- Before every merge: is the key unique, and how many rows do you expect?

# Session 4: get data, and rebuild it all

- API: save the raw answer first; check that every row arrived (`raise_for_status()`, a count against the metadata)
- Dates: `to_datetime`, `resample("QE")` with a stated rule (mean for prices, sum for flows), drop incomplete quarters; year-on-year growth is `pct_change(4)` for quarters
- Scale up: functions, loops over files, `run_all.py`, the regression and DuckDB are how you work, but not on this year's exam

- Save the raw answer first, and check that every row arrived.

# The exam

# What the exam looks like

- One exam for both parts: WISEflow, in an exam hall, 2 hours, no aids; 100 points, 50 to pass
- The Python part: 16 questions, 40 points; some give 2 points, some 3, and the paper shows the points at each question
- Not on the exam this year: your own functions, loops over files, `run_all.py`, the regression, DuckDB
- One correct answer per question; a wrong answer costs nothing, so answer them all
- Most questions show a few lines of code; a few are on the readings
- The Blackboard practice exam is from an older course: use the sample exam on the course site

- No computer: the skill is to read the code and predict.

# Three questions

# Predict on paper: how many rows does `df` have?

Alone, one minute; then hands up for A–D

```
df = pd.DataFrame({"region": ["N", "N", "S"],
                   "wage": [100, 200, 90]})
df["mean_wage"] = (df.groupby("region")["wage"]
                     .transform("mean"))
```

A. 2 B. 6 C. 3 D. 1

# Predict on paper: how many rows does `df` have?

Alone, one minute; then hands up for A–D

```
df = pd.DataFrame({"region": ["N", "N", "S"],
                   "wage": [100, 200, 90]})
df["mean_wage"] = (df.groupby("region")["wage"]
                     .transform("mean"))
```

A. 2 B. 6 C. 3 D. 1

**Answer: C, 3 rows.** `transform` keeps every row and gives each the mean of its region: 150, 150, 90 (Stata: `bysort region: egen`). `agg` would collapse to 2 rows.

# Predict on paper: how many rows does `long` have?

Alone, one minute; then hands up for A–D

```
wide = pd.DataFrame({"pid": [1, 2, 3],
                     "wage2023": [100, 110, 90],
                     "wage2024": [105, 112, 95]})
long = wide.melt(id_vars="pid", var_name="year",
                 value_name="wage")
```

A. 3  B. 2  C. 9  D. 6

# Predict on paper: how many rows does `long` have?

Alone, one minute; then hands up for A–D

```
wide = pd.DataFrame({"pid": [1, 2, 3],
                     "wage2023": [100, 110, 90],
                     "wage2024": [105, 112, 95]})
long = wide.melt(id_vars="pid", var_name="year",
                 value_name="wage")
```

A. 3  B. 2  C. 9  D. 6

**Answer: D, 6 rows.** One row per person and melted column: $3 \times 2 = 6$ (Stata: `reshape long`). And `year` holds the text `"wage2023"`: a number only after you clean and convert it. The reverse, long to wide, is `pivot` (Stata: `reshape wide`).

# Predict on paper: what does the last line print?

Alone, one minute; then hands up for A–D

```
origin = pd.Series([1, 3, 2])
names = {1: "US", 2: "Europe", 3: "Japan"}
region = origin.map(names)
print(region.iloc[1])
```

A. Europe  B. 3  C. US  D. Japan

# Predict on paper: what does the last line print?

Alone, one minute; then hands up for A–D

```
origin = pd.Series([1, 3, 2])
names = {1: "US", 2: "Europe", 3: "Japan"}
region = origin.map(names)
print(region.iloc[1])
```

A. Europe  B. 3  C. US  D. Japan

**Answer: D, Japan.** `map` looks up each code in the dictionary; `iloc[1]` is the *second* row (Python counts from 0), code 3.

# How to prepare

1. Do the sample exam on the course site in 45 minutes, on paper, then check
2. Watch the six videos (course site, *Videos*) and answer the questions under each before you open the answers
3. Do each session's self-test slide without AI
4. Redo the labs from scratch, without AI: predict each output before you run
5. Revisit the readings (course site, *Readings*): one short text per session; the exam has one or two questions on them

– Predict, then check. That is the exam, and it is how the course works.