Lecture 15 | pandas – Part I – Exercises

Guided exercise | Laboratory exercises | Independent exercises

ISEG Lisbon School of Economics & Management Universidade de Lisboa

Guided | Asset price snapshot

CONTEXT

An analyst receives a CSV snapshot of asset prices. Some readings are missing; the goal is to inspect the data, adopt a simple cleaning policy and produce a shortlist

INPUT | `data/prices_snapshot.csv`

Fields: asset (identifier), date (text), price (numeric, possibly missing)

EXPECTED OUTPUT

A compact table containing **asset**, **cleaned price** and **converted price in EUR** for assets **with price at least 100**

Guided | Understand the data first

SAMPLE CSV

asset	date	price
AAPL	2026-01-02	105
MSFT	2026-01-02	[missing]
GOOG	2026-01-02	130
TSLA	2026-01-02	85
AMZN	2026-01-02	[missing]

FIRST INSPECTION

- Use `.read_csv()` to load the file, then inspect `.head()`, `.shape` and `.dtypes`
- Use `.isna()` to identify which prices are missing

Guided | Cleaning and selection

PROCESS

- Create a copy of the original DataFrame, keeping the source unchanged
- Apply `.ffill()` to the price column: a missing entry takes the preceding valid price
- Build a Boolean mask selecting prices greater than or equal to 100
- Create `price_eur` using the illustrative conversion 1 USD = 0.92 EUR
- Display only asset, price and price_eur for selected rows

CHECK

- Cleaned prices: 105, 105, 130, 85, 85
- Selected assets: AAPL, MSFT, GOOG
- Discuss: is forward fill defensible for a snapshot of different assets?

Guided | Key code to complete

- STARTER: `starters/01_guided_price_cleanup.py`

- The starter provides imports, input path, output labels and blank operations marked TODO
- Fill in the inspection, missing-value mask, cleaning, selection and computed column while following the demonstration

- The sample is intentionally artificial: the objective is to expose the assumption, not recommend this policy for market data

Laboratory 1 | Student spending survey

CONTEXT

An anonymized survey records weekly spending and study hours. Identify a subset of students and estimate their daily spending

INPUT: `data/student_spending.csv`

Columns: `student, group, study_hours, weekly_spend`

YOUR REPORT SHOULD

- Show a small preview and the table dimensions
- Display just student and `weekly_spend` for all respondents
- Keep students studying at least 8 hours AND spending less than 80 per week
- Create `daily_spend = weekly_spend / 7` in the selected table
- Show `student`, `weekly_spend` and `daily_spend` without altering the source data

Laboratory 1 | Example and checks

EXAMPLE RECORDS

student	group	study_hours	weekly_spend
S01	A	10	70
S02	B	5	60
S03	A	8	75

EXPECTED BEHAVIOUR

- S01 and S03 satisfy the two conditions in this sample; S02 does not
- In the complete file, three students qualify: S01, S03, S06
- For S01, daily_spend = 70 / 7 = 10.0

- Check that students meeting only one condition are excluded

Laboratory 1 | Implementation hints

STARTER | `starters/02_lab_student_survey.py`

HINTS

- Use `.read_csv()` and `.head()` for input and inspection
- Use `.loc[:, ["student", "weekly_spend"]]` to select columns
- Build one Boolean condition per requirement and combine with &
- Each comparison must be enclosed in parentheses
- Use `.loc[mask].copy()` to obtain a separate selection
- Assign the new calculated column to that selection

SELF-CHECK

- Exactly three rows should remain; retain the CSV input row order

Laboratory 2 | Repair sensor readings

CONTEXT

A temperature sensor logged one reading per day, but some measurements are missing. Compare how three replacement policies alter the reconstructed data

INPUT | `data/sensor_daily.csv`

Columns: day, temperature; a blank cell means a missing reading

OUTPUT

- Keep the original temperature column untouched
- Add three columns: `previous, next, linear`
- Use `.ffill(), .bfill()` and `.interpolate()` respectively
- Display only the dates whose original temperature is missing

Laboratory 2 | Example of three policies

EXAMPLE

day	temperature	previous	next	linear
Jan 02	20	20	20	20
Jan 03	[missing]	20	26	23
Jan 04	26	26	26	26

WHAT TO VERIFY

There are three missing original measurements

- For Jan 03, the three strategies yield 20, 26 and 23
- For Jan 06 and Jan 08, inspect the corresponding results

- Question: which strategy would you choose for a physical signal changing smoothly over time?

Laboratory 2 | Implementation hints

STARTER | starters/03_lab_sensor_repair.py

HINTS

- Count missing values with `.isna().sum()` on the temperature Series
- Apply each replacement method to the same original Series, not to one already modified
- Assign each result to a newly named DataFrame column
- Use a mask on original temperature to show only originally missing rows

- INTERPRETATION
- Default `.interpolate()` treats the rows as equally spaced
- Do not assume a fill strategy is appropriate without considering the data
- You do not need to generate a chart: focus on the pandas operations

Independent 1 | Product price shortlist

SITUATION

A shop manager needs a discounted shortlist of products that can be offered immediately, checking their availability recorded in a CSV file.

INPUT | `data/products.csv`

Fields: `product, category, price, available`

REQUIRED RESULT

- Show the dataset dimensions
- Select products in the Electronics category that are available and cost at most 90
- Calculate `discounted_price` after a 10% discount on the qualifying selection
- Print only product, price and `discounted_price`

- No starter is provided; the original product prices must remain unchanged

Independent 1 | Example and acceptance

EXAMPLE

product	category	price	available
P02	Electronics	75	True
P04	Electronics	85	False
P05	Electronics	90	True

EXPECTED

- P02 qualifies and becomes 67.50 after discount
- P04 does not qualify because it is unavailable
- P05 qualifies and becomes 81.00

- In the supplied dataset the final shortlist must contain P02 and P05 only, because available products

Independent 2 | Inventory restoration

SITUATION

A store tracks daily stock by product. Some readings are missing; the reorder threshold comes from a separate settings file

INPUT FILES

- data/inventory_log.csv: product, date, stock
- data/inventory_settings.txt: minimum_stock=8

- REQUIREMENTS
- Read the threshold from the TXT file
- Fill each product’s missing stock with its own previous recorded value
- Add reorder_needed indicating that a known stock is below the threshold
- Print records requiring reordering and export the complete table to inventory_checked.csv

Independent 2 | Boundary cases and output

ILLUSTRATION

```
product      date       stock
A          Jan 01          11
A          Jan 02   [missing]    → 11
A          Jan 03           6    → reorder
C          Jan 01   [missing]    → remains missing
C          Jan 02           7    → reorder
```

IMPORTANT RULES

- The file is sorted by product and date, but observations from different products must never be mixed
- If no preceding value exists for a product, leave that stock missing
- A missing stock cannot automatically count as below the threshold
- The exported CSV should contain the restored values and the new indicator

Independent 3 | Branch sales report

CONTEXT

A company wants a reusable weekly sales shortlist from `data/branch_sales.csv` (branch, week, sales, refunds)

- Write `make_report(input_path, minimum_sales)` to:
- Load the CSV and exclude records with missing branch or sales
- Treat missing refunds as zero and calculate `net_sales = sales - refunds`
- Keep rows with net_sales at least minimum_sales
- Output `branch`, `week` and `net_sales` to `qualifying_sales.csv`
- Return the number of exported records

- The function must be called from a small interactive program

Independent 3 | Interface and checks

PROGRAM INTERFACE

- Prompt the user for the minimum net sales threshold
- Convert it to a number using try/except; on invalid input print a message and stop, without a retry loop
- Call `make_report()` and show how many rows were exported

REFERENCE EXAMPLE

- With threshold 100, qualifying rows are B01/W1, B01/W2 and B03/W2
- B01/W2 has missing refunds, therefore net_sales equals 110
- B02/W2 has missing sales and must be ignored

- Use only previously taught techniques: functions, exceptions, file I/O and pandas Part I