25 PYTHON DATA ANALYSIS EXERCISES

Student Workbook — Problems Only

Five progressive learning modules. Work independently before reading the provided solutions. Data are invented; not actual student or environmental measurements.

MODULE 1: PYTHON FUNDAMENTALS

Exercise 1: Calculate the Average Examination Score (Beginner)

Learning objective

Calculate a mean from a Python list using sum() and len().

Problem and sample data

Five fictional test scores are provided. Find their arithmetic average and describe what it represents. Data: scores = [62, 75, 81, 92, 70]

Starter code

```
scores = [62,75,81,92,70]
# Calculate average
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Replace the highest score with 100 and recalculate the average.

Exercise 2: Find Maximum and Minimum Daily Temperatures (Beginner)

Learning objective

Use min() and max() to summarize extremes.

Problem and sample data

Identify the coolest and warmest fictional day. Data: temps = [21,24,19,27,23] °C

Starter code

```
temps = [21,24,19,27,23]
# Print coolest and warmest
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Calculate the range (maximum minus minimum).

Exercise 3: Count Scores Reaching a Threshold (Beginner)

Learning objective

Use a loop and an if condition to count qualifying observations.

Problem and sample data

Count fictional scores of at least 75, including scores exactly equal to 75. Data: scores = [48,65,75,90,59,75]

Starter code

```
count = 0
for score in scores:
    # test score
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Count values strictly greater than 75 and explain the change.

Exercise 4: Calculate Electricity Use Percentage Change (Beginner)

Learning objective

Compute percentage change using an initial value and a new value.

Problem and sample data

A fictional household used 240 kWh in one month and 198 kWh the next. Find percentage change. Data: old = 240 kWh; new = 198 kWh

Starter code

```
old,new=240,198
# Percentage change
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Guard against old=0 and describe why percentage change is then undefined.

Exercise 5: Write a Reusable Mean Function (Beginner)

Learning objective

Define a function with a parameter and return value; handle empty inputs.

Problem and sample data

Write a function to average any non-empty numeric list and test it with three values. Data: values=[3,6,9]

Starter code

```
def safe_mean(values):
    # handle empty list
    pass
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Write a second function returning both the mean and sample size.

MODULE 2: NUMPY

Exercise 6: Create and Inspect a NumPy Array (Developing)

Learning objective

Learn array construction, shape, size and dimension count.

Problem and sample data

Create an array of five fictional rainfall measurements. Report its structure. Data: rain=[12,5,0,18,9] mm

Starter code

```
import numpy as np
rain = np.array([...])
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Create a two-row array and compare ndim and shape.

Exercise 7: Convert Celsius Temperatures to Fahrenheit (Developing)

Learning objective

Apply scalar arithmetic to all elements of a NumPy array.

Problem and sample data

Convert three fictional Celsius readings to Fahrenheit using vectorization. Data: Celsius: [0,20,37]

Starter code

```
c = np.array([0,20,37])
# Calculate F
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Convert -10°C and test whether the output is 14°F.

Exercise 8: Find NumPy Mean, Median and Standard Deviation (Developing)

Learning objective

Compare central tendency and population/sample spread.

Problem and sample data

Compute mean, median, population SD and sample SD for four numbers. Data: x=[2,4,6,8]

Starter code

```
x=np.array([2,4,6,8])
# statistics
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Change the data to [2,4,6,20] and compare the mean with the median.

Exercise 9: Select Readings Above a Fictional Limit (Developing)

Learning objective

Use Boolean indexing to find selected values.

Problem and sample data

From a fictional environmental sensor, select readings strictly above 40 units. This is not a safety threshold. Data: readings=[21,35,44,18,55]

Starter code

```
readings=np.array([...])
# Boolean mask
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Count qualifying readings and express the fraction of all five.

Exercise 10: Reshape Twelve Observations into Four Weeks (Developing)

Learning objective

Reshape an array while preserving the element count.

Problem and sample data

Twelve observations were collected three days per week for four weeks. Form a 4×3 array and calculate weekly totals. Data: values=1 to 12

Starter code

```
values=np.arange(1,13)
# reshape, sums
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Compute daily totals across four weeks using axis=0.

MODULE 3: PANDAS

Exercise 11: Build a Student Pandas DataFrame (Developing)

Learning objective

Create a labeled table with distinct rows and columns.

Problem and sample data

Represent four invented learners, subjects and scores in a DataFrame. Data: Asha Math 78; Leo Science 85; Mina Math 91; Noah Science 74

Starter code

```
import pandas as pd
# Build df
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add an Age column with four fictional ages.

Exercise 12: Filter Students Who Scored at Least 80 (Developing)

Learning objective

Apply a Boolean mask to a Pandas DataFrame.

Problem and sample data

Select student names with scores of at least 80 from the same four-row table. Data: Asha 78; Leo 85; Mina 91; Noah 74

Starter code

```
df = pd.DataFrame({...})
# filter
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Change the threshold to 75 and check the new selection.

Exercise 13: Sort Examination Results from Highest to Lowest (Developing)

Learning objective

Use sort_values to order records without confusing order and ranking.

Problem and sample data

Sort the fictional four-row table descending by Score. Data: Scores: Asha 78; Leo 85; Mina 91; Noah 74

Starter code

```
ordered = df.sort_values(...)
# names
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add a tied score and explore rank(method="min").

Exercise 14: Compute Average Scores by Subject (Developing)

Learning objective

Group observations by category and calculate group means.

Problem and sample data

Find the Math and Science means for the shared fictional student table. Data: Math 78,91; Science 85,74

Starter code

```
grouped = df.groupby(...)
# means
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Also report count and minimum by subject.

Exercise 15: Calculate Percentage Scores in a New Column (Developing)

Learning objective

Use vectorized Pandas arithmetic with row-specific denominators.

Problem and sample data

Three fictional tasks have different maximum marks. Calculate earned percentages. Data: earned=[76,60,45]; maximum=[100,80,50]

Starter code

```
df["Percentage"] = ...
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Compute average percentage across the three tasks and explain its meaning.

MODULE 4: CSV AND CLEANING

Exercise 16: Read a Reproducible CSV File (Developing)

Learning objective

Write then read a small CSV and inspect its structure.

Problem and sample data

Create the supplied rainfall CSV, then load it using read_csv. Data: TownA 18; TownB 24; TownC 11; TownD 30 (mm)

Starter code

```
pd.DataFrame(...).to_csv("town-rainfall.csv")
# read
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Change the filename and check Path.cwd() before reading.

Exercise 17: Count Missing Values in Two Fields (Developing)

Learning objective

Use isna and sums to inspect completeness.

Problem and sample data

Count missing Score and Attendance entries in a fictional table. Data: Score=[80,None,72]; Attendance=[94,88,None]

Starter code

```
df.isna().sum()
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Describe one plausible reason why missingness may not be random.

Exercise 18: Normalize Inconsistent Subject Labels (Developing)

Learning objective

Clean spaces and map known abbreviations explicitly.

Problem and sample data

Standardize five category labels into Mathematics or Chemistry. Data: [" Mathematics ","math","MATH","Chemistry"," chemistry "]

Starter code

```
labels = pd.Series([...])
# normalize
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add a previously unseen label and detect unmapped values.

Exercise 19: Remove Confirmed Exact Duplicate Rows (Developing)

Learning objective

Use duplicated() and drop_duplicates() after validating repeated records.

Problem and sample data

A small table contains the same A,8 record twice. Identify and remove only exact repetition. Data: A 8; B 6; A 8; C 7

Starter code

```
df.duplicated()
# clean
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add A with 9 hours and justify whether it is a duplicate.

Exercise 20: Export a Cleaned CSV and Read It Back (Developing)

Learning objective

Use to_csv(index=False) and verify the exported rows.

Problem and sample data

Write a cleaned three-record table and confirm its shape after rereading. Data: A 8; B 6; C 7

Starter code

```
clean.to_csv("cleaned.csv",index=False)
# verify
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Compare the reloaded DataFrame with clean using equals().

MODULE 5: CHARTS AND CAPSTONE

Exercise 21: Create a Labeled Bar Chart (Intermediate Beginner)

Learning objective

Use Figure and Axes to compare category magnitudes.

Problem and sample data

Plot three fictional subject averages as vertical bars and save the image. Data: English 78; Mathematics 73; Science 81

Starter code

```
fig,ax=plt.subplots()
# bars
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add numeric labels to each bar.

Exercise 22: Plot Chronological Rainfall Changes (Intermediate Beginner)

Learning objective

Create a line plot with observations in time order.

Problem and sample data

Plot rainfall across five fictional months and save the figure. Data: Jan–May [42,50,31,62,55] mm

Starter code

```
ax.plot(months,rain,marker="o")
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add a horizontal reference line at 50 mm.

Exercise 23: Plot Study Hours Against Examination Scores (Intermediate Beginner)

Learning objective

Use a scatter plot for two numeric variables and interpret cautiously.

Problem and sample data

Plot paired fictional observations without claiming causality. Data: Hours [1,2,3,4,5]; Scores [52,61,66,78,88]

Starter code

```
ax.scatter(hours,scores)
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Add a sixth observation that breaks the simple trend.

Exercise 24: Summarize Fictional Sales by Region (Intermediate Beginner)

Learning objective

Calculate revenue and group total and mean sale using groupby().agg().

Problem and sample data

Summarize a five-row synthetic shop dataset by North and South. Data: North: Book 3×15, Pen 5×2, Book 1×15; South: Book 2×15, Pen 4×2

Starter code

```
sales["Revenue"]=...
# group aggregate
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Calculate average revenue per record and total units by region.

Exercise 25: Student Performance Data Analysis Challenge (Intermediate Beginner)

Learning objective

Combine data import, quality checks, grouping, charting and responsible interpretation.

Problem and sample data

Complete a full mini-project: create or import the fictional CSV, inspect missing values and duplicates, retain valid scored rows, calculate overall and subject means, identify scores of 80+, generate a chart and export a verified clean CSV. Data: 12 distinct scored fictional student records (P01–P12) across Math, Science and English; an extra exact repeat of P03 and a P13 row with missing score. Attendance for P06 is missing.

Starter code

```
raw=pd.read_csv("exercise25-student-performance-raw.csv")
# inspect, clean, summarize, chart, export
```

Your challenge

Predict the output, write and test a complete solution, and explain the result in one or two sentences.

Extension task

Restart a new kernel, rerun the project and write a 100-word conclusion acknowledging its synthetic sample.
