Exploratory Data Analysis

Task 1 Count the right thing | 12 minutes

Work individually in Python. Create a script named STUDENTID_Lab5.py. Complete the TODOs, run your code, and explain your results to the TA during the lab. All required data are included in this manual; no separate dataset download is needed.

Lab duration 2 hours: Tasks 1-6, 80 minutes; break, 10 minutes; Task 7, 30 minutes. This lab practises the Week 5 concepts. All student records are synthetic.

Table	One row represents	Repeated student ID
students	One student	Should be unique
enrolments	One course registration in one term	Can be correct
activity	One student in one week	Can be correct

1. Compare registrations with students

Student A passes three courses; student B takes one course and fails. There are four registrations and two students. A pass means a final grade of at least 50 points.

```
import pandas as pd
students = pd.DataFrame({"id": ["A", "B"],
                         "mode": ["Full time", "Part time"]})
enrolments = pd.DataFrame([
    ["A", "C1", "T1", 60], ["A", "C2", "T1", 70],
    ["A", "C3", "T1", 80], ["B", "C1", "T1", 40]
], columns=["id", "course", "term", "grade"])
enrolments["passed"] = enrolments["grade"] >= 50
# TODO: passes = enrolments["passed"].sum()
# TODO: print the fraction and percentage of registrations passed
# TODO: count students who passed at least one course
```

Small example A contributes three passing registrations but only one student who passed at least one course. For the second calculation, use enrolments.groupby("id")["passed"].any().sum().

Check The registration rate is 3/4 = 75%; the student rate under the stated definition is 1/2 = 50%. Explain why both can be correct.

2. Attach student information without multiplying rows

```
joined = enrolments.merge(students, on="id", how="left",
                          validate="many_to_one", indicator=True)
print("Rows before and after:", len(enrolments), len(joined))
print("Unmatched:", joined["_merge"].ne("both").sum())
# TODO: print students["id"].is_unique
# TODO: check duplicates using ["id", "course", "term"]
```

Check There must be four joined rows and zero unmatched registrations. Explain why a repeated ID in enrolments is allowed, while repeating A in students could multiply the join.

Show the TA Your two fractions, row counts and one complete question that names the population, period and unit. IDs identify records; their digits are not measured quantities.

If needed, install the libraries in your local terminal: **python -m pip install pandas numpy matplotlib**. Keep using the same script for the next tasks.

Fundamental Data Science - NUU

Describe the distribution

Task 2 | 16 minutes

1. See what one large value changes

```
from statistics import mean, median
logins = [1, 2, 3, 4, 40]
print("Mean:", mean(logins), "Median:", median(logins))
# TODO: replace 40 with 400 in a NEW list and repeat
```

Check The original mean is 10 and median is 3. Predict the new values before running the code. Which summary moves more? Why is a large observation not automatically an error?

Small example The middle of the sorted five values is the third value, 3. The mean uses the sum of all five values, so 40 affects it directly.

2. Equal means can hide different spread

```
A = [50, 55, 60, 65, 70]
B = [20, 40, 60, 80, 100]
print("A:", mean(A), max(A) - min(A))
# TODO: print the same two summaries for B
```

Check Both means are 60 points. The ranges are 20 and 80 points. State which group is more spread out and why the mean alone is insufficient.

For 1,200 students, the entry-score quartiles are Q1 = 58.3 and Q3 = 73.6. Calculate **IQR = Q3 - Q1**. Explain what the middle half means. These are student entry scores, not registration final grades.

Small example IQR uses the two quartile boundaries; full range uses the maximum and minimum. They describe different parts of the spread.

3. Build a histogram

Grades: **42, 47, 53, 58, 61, 64, 65, 78**. First count the grades in 40 to below 50, 50 to below 60, 60 to below 70, and 70 to 80. The final interval includes its right endpoint.

```
import numpy as np
import matplotlib.pyplot as plt
grades = [42, 47, 53, 58, 61, 64, 65, 78]
edges = [40, 50, 60, 70, 80]
print("Bin counts:", np.histogram(grades, bins=edges)[0])
plt.hist(grades, bins=edges, edgecolor="black")
plt.xlabel("Final grade in points")
plt.ylabel("Number of registrations")
plt.show()
# TODO: repeat with edges [40, 60, 80]
```

Check The four original bin counts are 2, 2, 3, 1; they sum to 8. Changing bins changes the picture, not the grades. State what each axis counts and which interval is most populated.

Choose a plot Would you use a histogram or a categorical bar chart for programme counts? Explain using the kind of variable. Sketch the bar chart for CS = 4 and Economics = 2 students.

Show the TA Your two login summaries, both score ranges, the IQR, histogram counts and an explanation of why bins touch in a histogram.

Fundamental Data Science · NUU

Calculate correlation by hand

Task 3 | 18 minutes across two pages

Pearson correlation r summarizes the direction and strength of a **linear** association between two numerical variables. Each row must be a matched (x, y) pair.

1. Use Pearson's correlation formula

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^{n}(x_i - \bar{x})^2\right]\left[\sum_{i=1}^{n}(y_i - \bar{y})^2\right]}}$$

n counts paired rows. Bars mean averages: sum each variable and divide by n. The summation sign means add across all pairs. Correlation ranges from -1 to +1.

Both variables must vary. If either variable is constant, the denominator is zero and r is **undefined**, not zero.

2. Complete one calculation manually before coding

Use **x = [1, 2, 3, 4]** and **y = [1, 3, 2, 4]**. Calculate both means, then complete the table: dx = x minus its mean; dy = y minus its mean. A calculator is allowed; **do not run Python yet**.

x	y	dx	dy	dx x dy	dx squared	dy squared
1	1	-1.5	-1.5	2.25	2.25	2.25
2	3	___	___	___	___	___
3	2	___	___	___	___	___
4	4	___	___	___	___	___
Sum		___	___	___	___	___

Small example The x mean is (1 + 2 + 3 + 4) / 4 = 2.5, so the first dx = 1 - 2.5 = -1.5. Check the y mean and complete the other rows.

Write on paper Both means, all entries, the three formula sums, the substituted fraction and r. Interpret its sign and whether the relationship is perfectly linear. Both deviation totals should be zero.

3. Only now verify your manual result with code

```
x_hand = np.array([1, 2, 3, 4], dtype=float)
y_hand = np.array([1, 3, 2, 4], dtype=float)
dx = x_hand - x_hand.mean()
dy = y_hand - y_hand.mean()
numerator = np.sum(dx * dy)
denominator = np.sqrt(np.sum(dx ** 2) * np.sum(dy ** 2))
print("Formula result:", numerator / denominator)
print("NumPy check:", np.corrcoef(x_hand, y_hand)[0, 1])
```

Show the TA Your handwritten calculation, then the code output. If they differ, check the pairing, means, signs and sums. Continue with three datasets on the next page.

Fundamentals of Data Science | NUU

Compare three correlation patterns

Task 3 continued | Positive, negative and zero linear correlation

4. Predict, plot and calculate for all three datasets

Use the common x row with each y row to form **three paired datasets**. Before running code, predict the sign of r for each. Here x is a mathematical input, not a student ID.

x	-3	-2	-1	0	1	2	3
Dataset A y = x	-3	-2	-1	0	1	2	3
Dataset B y = -x	3	2	1	0	-1	-2	-3
Dataset C y = x squared	9	4	1	0	1	4	9

```
x = np.arange(-3, 4)
datasets = {"A": x, "B": -x, "C": x ** 2}
fig, axes = plt.subplots(1, 3, figsize=(9, 3))
for ax, (name, y) in zip(axes, datasets.items()):
    ax.scatter(x, y)
    ax.set(xlabel="x", ylabel="y", title=name)
    # TODO: compute r with np.corrcoef(x, y)[0, 1]
    # TODO: print the dataset name and its r
plt.tight_layout()
plt.show()
```

Check A: r = +1, perfect positive linear correlation. B: r = -1, perfect negative linear correlation. C: r = 0, zero linear correlation, up to rounding. For each, describe what happens to y as x increases.

Small example In C, x = -2 and x = 2 both give y = 4. Its exact U-shaped relationship remains even though r = 0. Explain why zero correlation does not establish that variables are independent or unrelated.

5. Keep the plot and the statistic on the same population

The lecture showed a scatter plot of 50 selected DS100 students. The reported correlation of about 0.48 was calculated using **all 1,200 DS100 students**.

Answer in your own words:
1. Does one point represent a student, a week, or a course total?
2. Can we claim that r = 0.48 is the correlation of those 50 plotted points?
3. Why should a description include the number of score pairs?

Small example A statement such as "Among all 1,200 DS100 students, entry score and final grade have a positive linear association" names the observations being summarized.

6. Rewrite an unsupported claim

A colleague says: "The correlation is positive, so increasing entry score will cause a higher final grade." Rewrite this using only the evidence the plot and observational records support. Give one other factor that could relate to both scores.

Check Use language such as "is associated with". A correlation describes how recorded values vary together; changing one value is a different question.

Show the TA Your manual calculation, all three scatter plots and correlations, one sentence about the zero-correlation curve and your revised entry-score claim.

Fundamental Data Science · NUU

Keep unknown outcomes separate

Task 4 | 10 minutes

1. Choose the denominator for the question

Consider the 100-student cohort below. Every student will eventually have either a pass or a fail, but some outcomes have not yet been recorded.

Known pass	Known fail	Unknown	Whole cohort
60	20	20	100

```
passed, failed, unknown = 60, 20, 20
known = passed + failed
total = known + unknown
print("Known-outcome rate:", 100 * passed / known)
# TODO: lowest whole-cohort rate = 100 * passed / total
# TODO: highest rate = 100 * (passed + unknown) / total
```

Small example The known-outcome question counts 60 passes among 80 recorded outcomes. The whole-cohort question has denominator 100 because it includes the 20 unknown cases.

Check Known-outcome rate = 75%. The eventual whole-cohort rate can range from 60% to 80%. State which assumption gives each endpoint.

Explain why these two statements are unsupported:
1. "Exactly 75% of the whole cohort passed."
2. "The 20 unknown outcomes should be counted as failures."

These bounds cover possible outcomes of the unknown cases. They are **not a confidence interval**. Missing outcomes can also be unevenly distributed between groups.

2. See why a small denominator matters

The synthetic course data contain **1 pass among 9 part-time ST110 registrations**. Compare this with a hypothetical group of **100 passes among 900 registrations**. Both start at the same percentage.

```
small_pass, small_n = 1, 9
large_pass, large_n = 100, 900
small_rate = 100 * small_pass / small_n
print("Small original rate:", small_rate)
# TODO: calculate the large original rate
# TODO: change ONE failure to a pass in each hypothetical group
# TODO: keep n fixed and report both new rates
# TODO: subtract each old rate from its new rate
```

Small example One changed outcome in a group of n changes the rate by **100/n percentage points**. Compare the hypothetical changes while keeping the original counts as your reference.

Check One changed outcome moves the small-group rate by about 11.1 percentage points, and the large-group rate by about 0.111 percentage points. Explain the difference using the denominators.

3. Report a rate as a complete sentence

Write a sentence that includes the subgroup, passes / n, percentage and a caution about the small count. Then write a second sentence that accurately reports the known-outcome rate above.

Show the TA The three unknown-outcome fractions, both rate changes and your two reporting sentences. Keep percentage points separate from relative percent change.

Fundamental Data Science · NUU

Compare overall and within groups

Task 5 | 18 minutes

Each table entry is **passes / registrations**, across all recorded terms. Every outcome is known. The quantitative-service group contains MA110, CH110 and ST110.

Course group	Full time	Part time
Quantitative service	305 / 1,027	7 / 38
Other nine courses	2,754 / 3,081	422 / 519
All courses	3,059 / 4,108	429 / 557

1. Calculate each comparison

```
ft_pass = np.array([305, 2754])
ft_n = np.array([1027, 3081])
pt_pass = np.array([7, 422])
pt_n = np.array([38, 519])
ft_rates = ft_pass / ft_n
# TODO: pt_rates = pt_pass / pt_n
print("FT within groups (%):", 100 * ft_rates)
# TODO: print PT within groups and both pooled percentages
# Pooled proportion = sum(passes) / sum(registrations)
```

Check Full time is higher within both broad groups; part time is higher overall. Overall rates are about 74.46% and 77.02%. Calculate the part-time minus full-time gap in percentage points. This reversal is Simpson's paradox.

2. Explain the reversal with group sizes

```
ft_weights = ft_n / ft_n.sum()
print("FT group weights:", ft_weights)
print("FT reconstructed (%):", 100 * sum(ft_weights * ft_rates))
# TODO: compute PT weights and reconstruct its pooled rate
# TODO: compare both modes using a common 50:50 mix:
# equal_mix_rate = 0.5 * first_rate + 0.5 * second_rate
```

Small example Full-time weights are 1,027/4,108 = 0.25 and 3,081/4,108 = 0.75. Do not use a simple average of group percentages to reconstruct a pooled rate unless the groups have equal sizes.

Check The part-time other-course weight is about 0.932. Explain why placing more weight on that higher-pass group can reverse the pooled ranking. Use unrounded rates in calculations.

One course is an exception to the broader pattern: **EC102** has 207/240 full-time and 30/34 part-time passes. Calculate both percentages. Why is "full time leads in every individual course" inaccurate?

Task 6 Say what the evidence supports

6 minutes

1. Rewrite "Part-time study causes better performance." Include overall rates, course mix and a causal limitation.
2. Students choose whether to join a support programme. Why might readiness affect both participation and passing, and why might other differences remain after grouping?
3. For later prediction, explain why feature selection uses training data, one student stays in one split, and the final test waits until choices are fixed.

Show the TA Rates, weights, the common-mix comparison and your explanation. **Take a 10-minute break** before Task 7.

Fundamentals of Data Science · NUU

Combined student analysis

Task 7 | 30 minutes across two pages

Question: How do recorded pass rates compare between students who chose support and students who did not? Readiness was measured **before** support began. Final grade was measured afterward. Participation was not randomized.

The table contains **16 students**. Each ID appears once and every final grade is known. A pass means a final grade of at least 50 points.

ID	Support	Readiness	Final grade
S01	Yes	Lower	40
S02	Yes	Lower	45
S03	Yes	Lower	50
S04	Yes	Lower	55
S05	Yes	Lower	35
S06	Yes	Lower	48
S07	Yes	Higher	72
S08	Yes	Higher	82
S09	No	Lower	30
S10	No	Lower	44
S11	No	Higher	49
S12	No	Higher	60
S13	No	Higher	65
S14	No	Higher	70
S15	No	Higher	80
S16	No	Higher	90

Fundamental Data Science · NUU

1. Enter the printed table

```
rows = [
 ["S01","Yes","Lower",40], ["S02","Yes","Lower",45],
 ["S03","Yes","Lower",50], ["S04","Yes","Lower",55],
 ["S05","Yes","Lower",35], ["S06","Yes","Lower",48],
 ["S07","Yes","Higher",72], ["S08","Yes","Higher",82],
 ["S09","No", "Lower",30], ["S10","No", "Lower",44],
 ["S11","No", "Higher",49],["S12","No", "Higher",60],
 ["S13","No", "Higher",65],["S14","No", "Higher",70],
 ["S15","No", "Higher",80],["S16","No", "Higher",90]
]
df = pd.DataFrame(rows, columns=["id","support","readiness","grade"])
df["passed"] = df["grade"] >= 50
# TODO: print row count, unique IDs and missing-grade count
```

Check 16 students, unique IDs and zero missing grades. There are 8 students with support and 8 without. A grade of 50 is a pass.

2. Predict before grouping

Count passes by hand in each support group. Predict which group leads overall, then predict whether that conclusion holds within both readiness levels. Continue on the next page.

Calculate and defend the comparison

Task 7 continued | Keep using the same 16-student table

3. Calculate the pooled and within-readiness rates

```
overall = df.groupby("support")["passed"].agg(["sum", "count"])
overall["rate_pct"] = 100 * overall["sum"] / overall["count"]
print(overall)
within = df.groupby(["readiness", "support"])["passed"].agg(
    ["sum", "count"])
# TODO: add a rate_pct column to within and print it
```

Small example For support = Yes and readiness = Lower, there are 2 passes among 6 students. Build the other three fractions yourself.

Report the overall gap as **support percentage minus no-support percentage**. Does the ranking reverse when readiness is considered? Name the pattern and explain it using counts.

4. Reconstruct with weights and compare the same mix

For each support group, calculate the lower- and higher-readiness shares. Multiply each share by its own readiness-specific pass rate and add. The result must equal the pooled proportion from Step 3.

Small example Supported students have lower-readiness weight 6/8 and higher-readiness weight 2/8. Reconstruct using (6/8) x (2/6) + (2/8) x (2/2). Work out the no-support expression yourself.

Then use a hypothetical **50:50 readiness mix for both groups**. Compute 0.5 x lower-readiness rate + 0.5 x higher-readiness rate. Explain why this is a different comparison from the observed pooled rate and why it still does not prove a causal effect.

5. Inspect one distribution individually

If the last digit of your student ID is even, inspect the **Yes** support group; if it is odd, inspect the **No** support group. Use your actual group in the code below.

```
selected = df.loc[df["support"] == "Yes", "grade"]
# TODO: use "No" above if your student ID ends with an odd digit
print("n:", len(selected), "mean:", selected.mean())
# TODO: print median, minimum, maximum and range
plt.hist(selected, bins=[0, 20, 40, 60, 80, 100], edgecolor="black")
# TODO: label grade in points on x and student count on y
plt.show()
```

Give your group, n, mean, median and range. State what the histogram shows. Grades are binned with left endpoints included; the last bin includes 100. Explain why the mean is not itself the pass rate.

6. Write a short defensible conclusion

Write 4-6 sentences: **observation** with pass counts, denominators and rates; **interpretation** with the readiness mixture and reversal; **limitation** covering the observational design and small groups. Give one plausible pre-support difference that readiness alone may miss.

7. Show the TA and explain

Show your data checks, pooled and within-group tables, weights, common-mix rates and individual histogram. Explain one denominator and one limitation in your own words. Save your completed script and calculations.

Final check Every rate names its denominator; each pair of weights sums to 1; no record was changed to create a reversal; all plot axes name units; and your conclusion separates association from causation.

Fundamental Data Science - NUU