

LECTURE 7

Seeing data

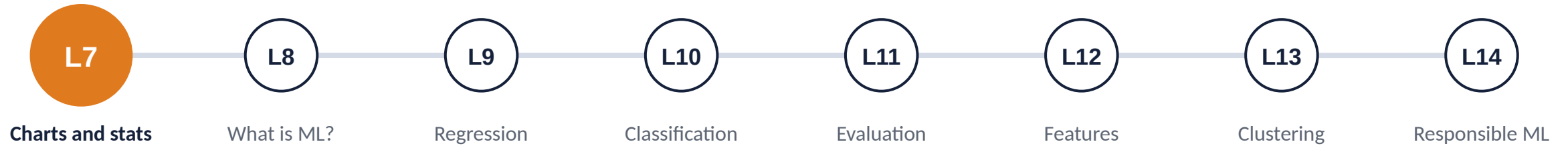
Charts and basic statistics: the foundation of machine learning

Today's key line of code

```
df["length"].hist()  
df["length"].describe()
```

The road ahead: machine learning

Today



Today you will

- 1 Summarise a column with mean, median and spread
- 2 Calculate a mean and standard deviation by hand
- 3 Draw histograms, bar charts, scatter plots and box plots
- 4 Measure how two columns move together with correlation
- 5 Know why correlation is not causation

Mean and median: two kinds of average

- `mean()` adds everything and divides by the count
- `median()` is the middle value after sorting
- One extreme value pulls the mean, not the median
- Real data often has extremes: report both

Code

```
import pandas as pd
marks = pd.Series([55, 62, 68, 70, 71, 74, 78, 99])
print(marks.mean())
print(marks.median())
marks2 = pd.Series([55, 62, 68, 70, 71, 74, 78, 300])
print(marks2.mean())
print(marks2.median())
```

Output

```
72.125
70.5
97.25
70.5
```

By hand: mean and standard deviation

Standard deviation measures how far values typically are from the mean. Fill each column, then add up the last one.

Code

```
values = [2, 4, 4, 4, 5, 5, 7, 9]
# mean = sum / count = 40 / 8 = 5
# variance = sum of squared gaps / count
#           = 32 / 8 = 4
# standard deviation = square root = 2
```

value	value - mean	(value - mean) ²
2	-3	9
4	-1	1
4	-1	1
4	-1	1
5	0	0
5	0	0
7	2	4
9	4	16
sum = 40		sum = 32

Click to reveal the next row. Predict it first!

describe(): the whole summary at once

- count, mean, standard deviation (std)
- min and max: the range
- 25%, 50%, 75%: quartiles (50% = median)
- pandas `std()` divides by count - 1, so 2.14, not 2

Code

```
values = pd.Series([2, 4, 4, 4, 5, 5, 7, 9])  
print(values.std().round(2))  
print(values.describe().round(2))
```

Output

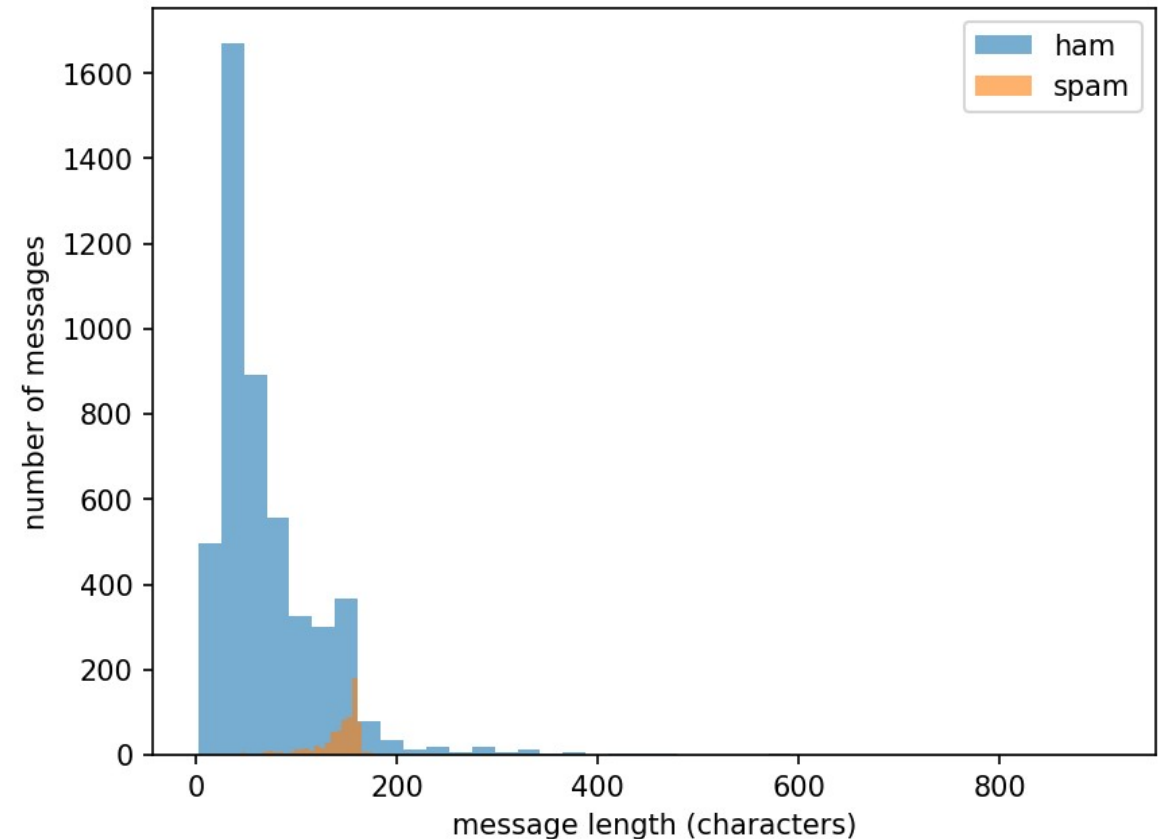
```
2.14  
count      8.00  
mean       5.00  
std        2.14  
min        2.00  
25%        4.00  
50%        4.50  
75%        5.50  
max        9.00  
dtype: float64
```

Histogram: the shape of one column

A histogram groups values into bins and counts each bin. Spam and normal messages have very different length shapes.

Code

```
import matplotlib.pyplot as plt
df = pd.read_csv("sms.tsv", sep="\t", header=None,
                 names=["label", "message"])
df["length"] = df["message"].str.len()
ham = df[df["label"] == "ham"]["length"]
spam = df[df["label"] == "spam"]["length"]
plt.hist(ham, bins=40, alpha=0.6, label="ham")
plt.hist(spam, bins=40, alpha=0.6, label="spam")
plt.xlabel("message length (characters)")
plt.ylabel("number of messages")
plt.legend()
plt.show()
```



Watch it: how a histogram is built

WATCH FOR

- Each mark drops into the bin covering its value
- Bar height = how many values in that bin
- The shape appears as values accumulate
- `plt.hist` does all of this in one line

CHARTS, STEP BY STEP

How a histogram is built

40 exam marks arrive one at a time.

Each one drops into the bin that covers its value.

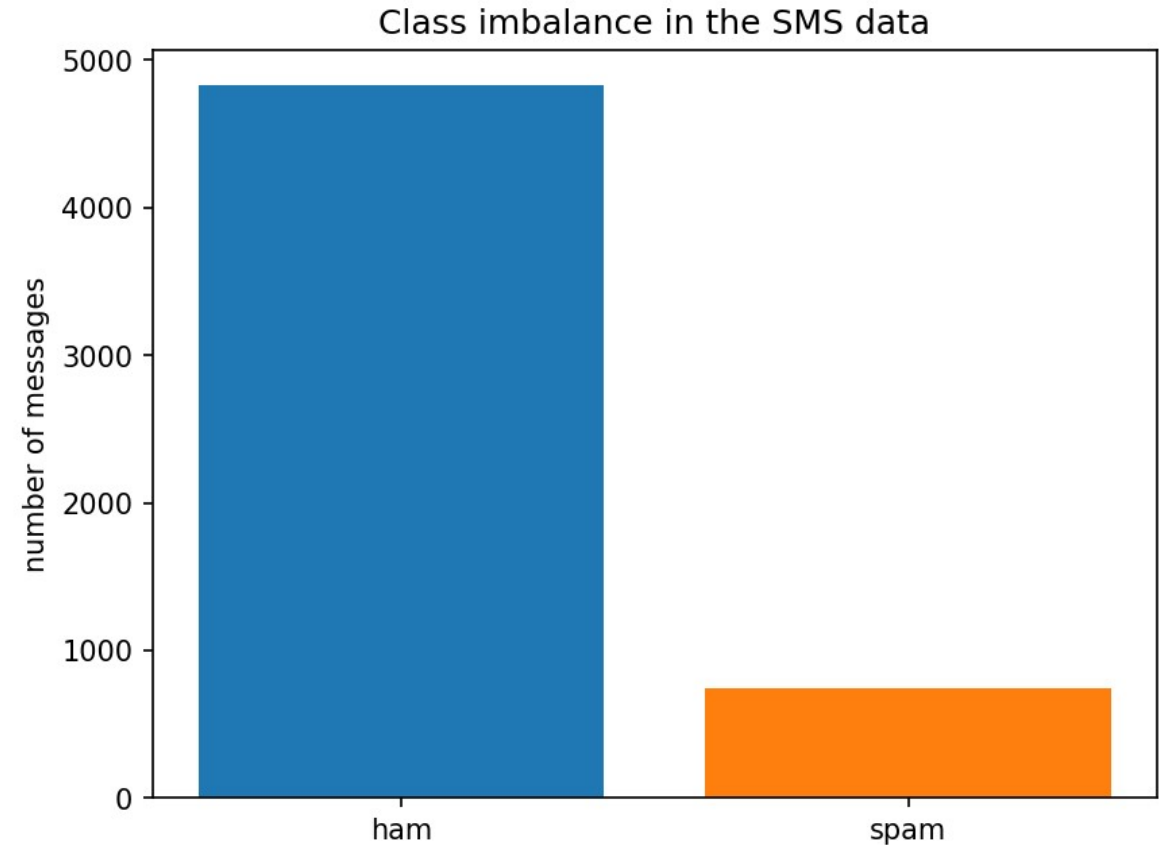
Click the video to play, click again to pause. Silent: talk the class through it.

Bar chart: counts per category

Use a bar chart to compare counts of categories. Always label the axes.

Code

```
counts = df["label"].value_counts()
plt.bar(counts.index, counts.values,
        color=["tab:blue", "tab:orange"])
plt.ylabel("number of messages")
plt.title("Class imbalance in the SMS data")
plt.show()
```



Scatter plot: two columns at once

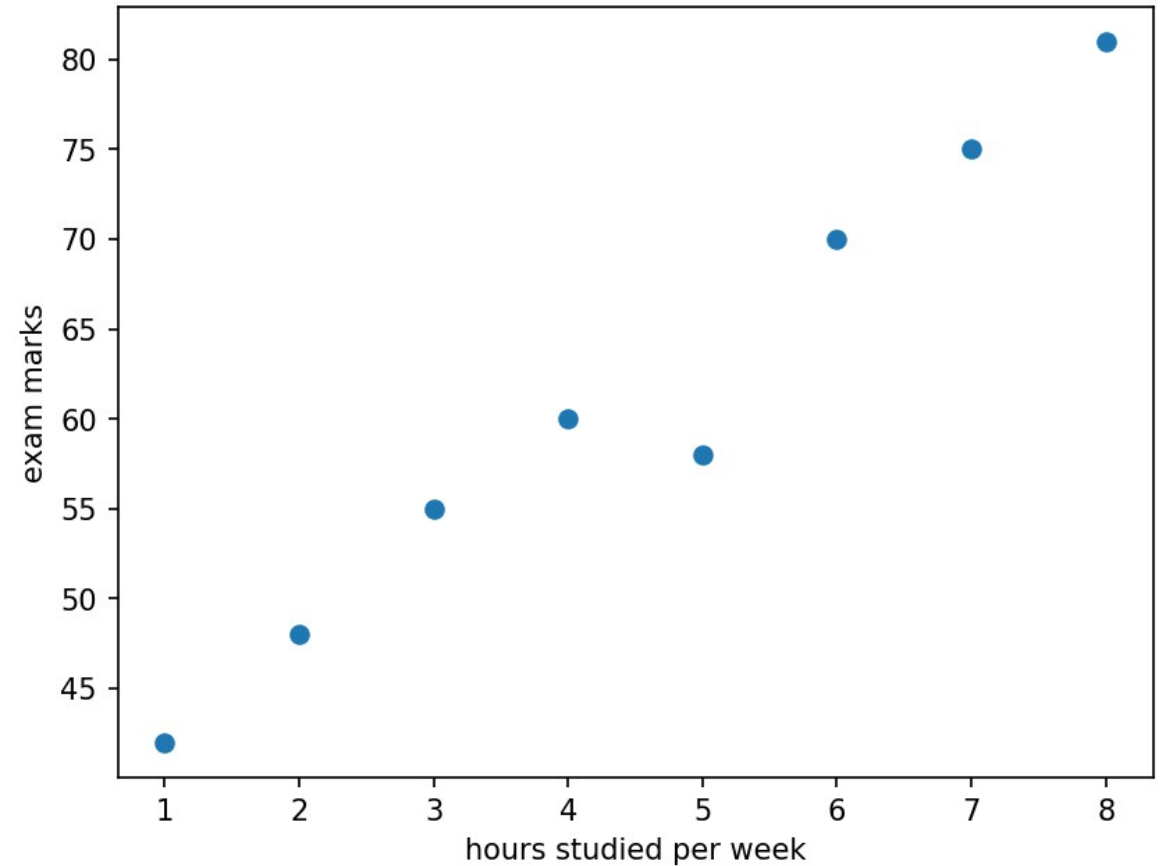
Each dot is one student. The pattern shows how marks change with hours studied (a small invented example).

Code

```
hours = [1, 2, 3, 4, 5, 6, 7, 8]
marks = [42, 48, 55, 60, 58, 70, 75, 81]
plt.scatter(hours, marks)
plt.xlabel("hours studied per week")
plt.ylabel("exam marks")
plt.show()
r = pd.Series(hours).corr(pd.Series(marks))
print(round(r, 3))
```

Output

0.983



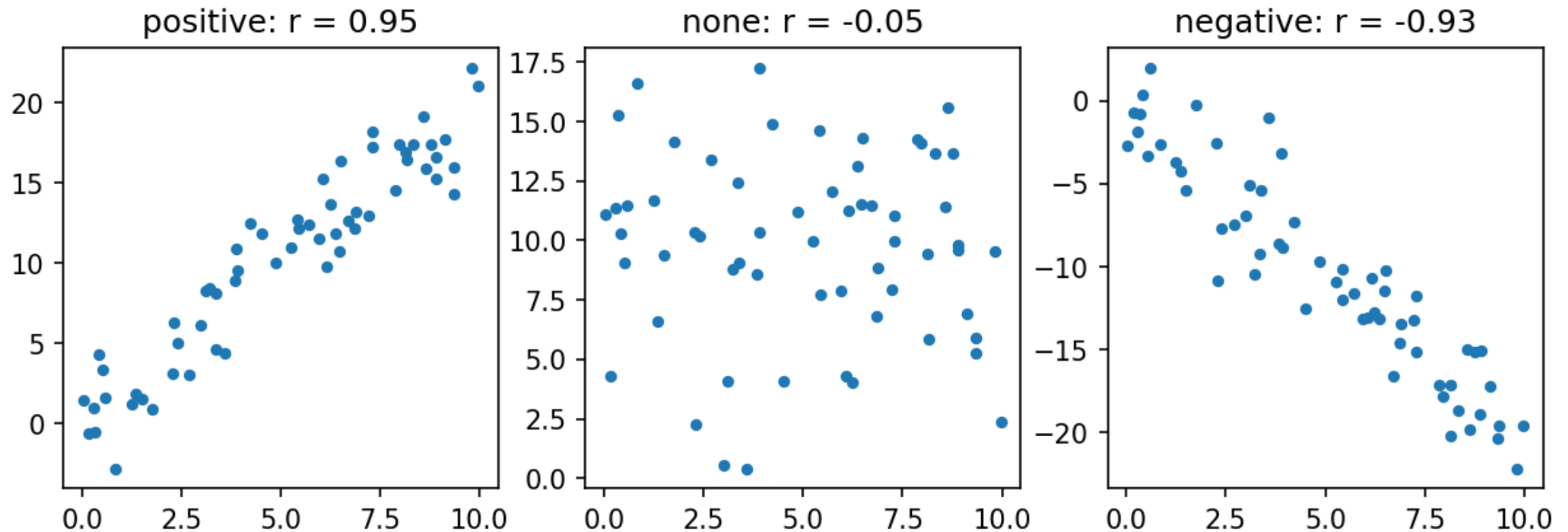
Correlation: how strongly two columns move together

Correlation r runs from -1 to +1. Close to +1: rise together. Close to -1: one rises as the other falls. Near 0: no straight-line pattern.

Code

```
import numpy as np
rng = np.random.default_rng(0)
x = rng.uniform(0, 10, 60)
ys = {"positive": 2 * x + rng.normal(0, 2, 60),
      "none": rng.normal(10, 4, 60),
      "negative": -2 * x + rng.normal(0, 2, 60)}
fig, axes = plt.subplots(1, 3, figsize=(10, 3))
for ax, (name, y) in zip(axes, ys.items()):
    r = np.corrcoef(x, y)[0, 1]
    ax.scatter(x, y, s=12)
    ax.set_title(f"{name}: r = {r:.2f}")
plt.show()
```

Correlation: how strongly two columns move together: the result



Watch it: correlation as the points tighten

WATCH FOR

- Same points, less and less noise
- r climbs towards +1 as the line tightens
- Tilting the slope down makes r negative
- Adding noise again weakens it

STATISTICS, STEP BY STEP

Correlation as the points tighten

The same 60 points, with less and less noise.

Watch r climb towards +1, then flip sign.

Click the video to play, click again to pause. Silent: talk the class through it.

Correlation is not causation

1

Ice cream and drowning

Both rise in hot weather. Ice cream does not cause drowning; heat drives both.

2

Hidden third factor

A correlation can come from something you did not measure. Ask: what else could explain it?

3

Why it matters for ML

Models use correlations to predict. Prediction is useful, but it does not prove one thing causes another.

Box plot: spread and outliers at a glance

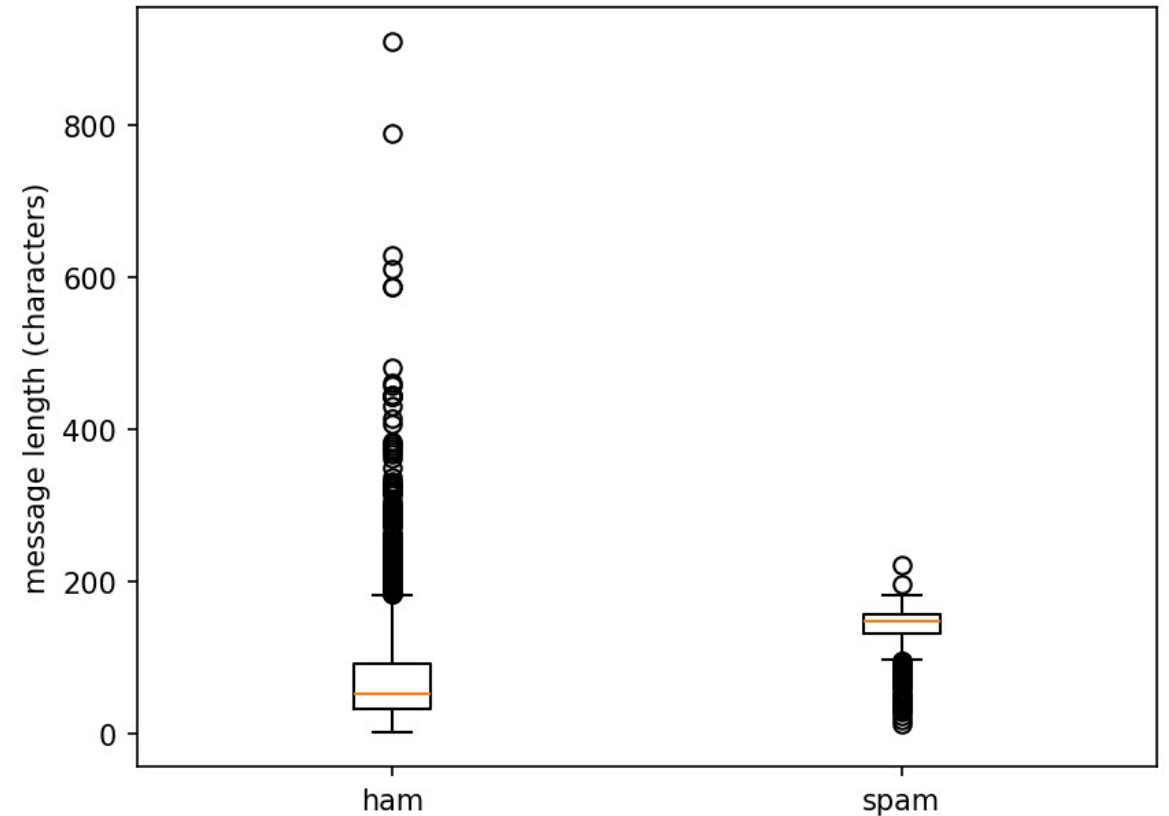
The box holds the middle 50% of values, the line is the median, and dots are unusual values (outliers).

Code

```
plt.boxplot([ham, spam], labels=["ham", "spam"])
plt.ylabel("message length (characters)")
plt.show()
print(ham.median(), spam.median())
```

Output

```
52.0 149.0
```



Your turn

10 minutes

- 1 Load students.csv with `pd.read_csv`
- 2 Print `describe()` for the hours column
- 3 Draw a histogram of attendance
- 4 Draw a scatter plot of hours against prior_gpa and print the correlation

Code

```
st = pd.read_csv("students.csv")
print(st["hours"].describe())
plt.hist(st["attendance"], bins=15)
plt.show()

...
```

Starter code: replace the ... with your own code

Common mistakes with charts and statistics

Read the last line of an error first. It names the problem.

What you typed	What Python says	What it means
<pre>plt.scatter([1, 2, 3], [4, 5])</pre>	<pre>ValueError: x and y must be the same size</pre>	x and y must have the same number of values.
<pre>print(df["label"].mean())</pre>	<pre>TypeError: Cannot perform reduction 'mean' with string dtype</pre>	You cannot average text. Use <code>value_counts()</code> for categories.
<pre>plt.hist(df["lenght"])</pre>	<pre>KeyError: 'lenght'</pre>	Typo in the column name. Check <code>df.columns</code> .

Interactive lab: Charts Lab

In the lab you can

1 Mean, median, spread

Drag one value away and watch the mean move, not the median.

2 Histogram

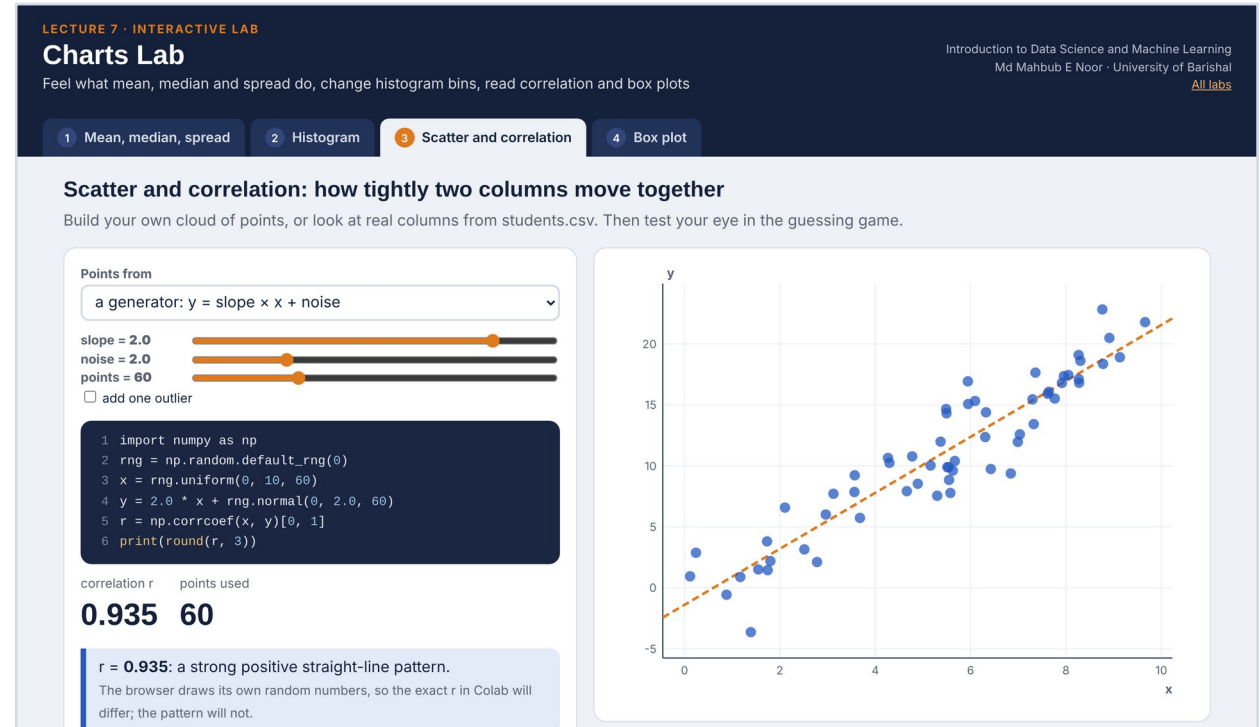
Change the bins and see the shape change.

3 Scatter and correlation

Build point clouds, read students.csv, guess r .

4 Box plot

Quartiles, whiskers and outliers, compared by group.



Open the lab ▶

Works offline in any browser

Labs_open_in_browser/
L7_Charts_Lab.html

Today's toolkit

<code>s.mean(), s.median()</code>	Two kinds of average
<code>s.std()</code>	Typical distance from the mean
<code>s.describe()</code>	Full numeric summary
<code>plt.hist(x, bins=)</code>	Shape of one number column

<code>plt.bar(names, counts)</code>	Compare categories
<code>plt.scatter(x, y)</code>	Relationship of two columns
<code>a.corr(b)</code>	Strength of a straight-line pattern
<code>plt.boxplot([a, b])</code>	Compare spread of groups

Homework 7

Due before Lecture 8 • [date]

1 By hand

Compute the mean and standard deviation (divide by count) of 3, 5, 7, 9, 11 in a table like today's. Check with numpy: `np.std(values)`.

2 Median vs mean

Make a list of 7 prices, then replace the largest with a value 10 times bigger. Print mean and median before and after. Explain.

3 Four charts

From students.csv: a histogram, a bar chart of dept counts, a scatter plot and a box plot of hours by passed. Label every axis.

4 Correlation table

Print `st[["hours", "attendance", "prior_gpa"]].corr()`. Which pair is most correlated?

5 Causation

Write one example of two things that are correlated but where one does not cause the other.

Submit on [LMS / Google Classroom]: your notebook file HW7_<your ID>.ipynb, or a Colab link set to “Anyone with the link can view”.

Recap: Lecture 7 in 90 seconds

RECAP

Lecture 7 in 90 seconds

A short recap of today's lecture. Pause any time.

Click to play or pause. Watch it again at home to revise.