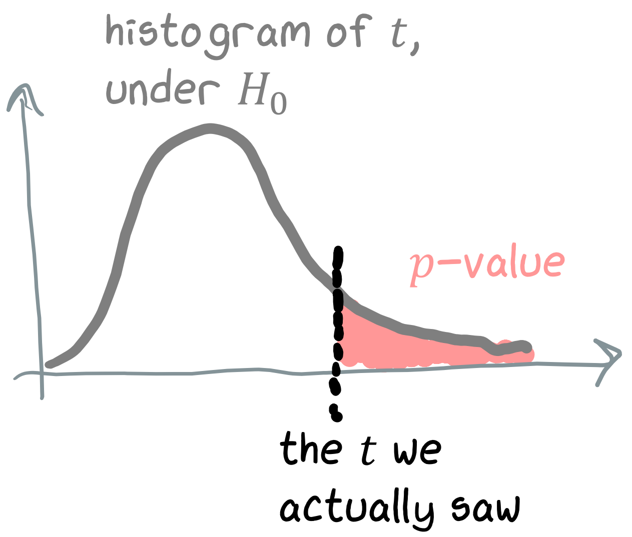
§9.3 Hypothesis testing

Hypothesis testing asks whether a proposed probability model H_0 could plausibly have generated the dataset.

- The p-value is the probability that an outcome as extreme as what we actually saw might have come about by chance, if H_0 were true.
- A low p-value suggests we should reject H_0.
- "Extreme" is measured by a test statistic t, which is up to us to choose.



QUESTION.
Why do you think we define the p-value this way, rather than defining it to be "the probability of the t that we actually saw"?

Hypothesis testing is good for questions that we can cast as
"Does the evidence suggest rejecting H_0?"

- Is my probability model a good enough fit for the dataset?
- Is my new algorithm better than the standard one?
- Does this new UI allow users to do their task faster than before?
- Is this drug effective, compared to placebo?

H_0: the data was generated by my model

comparing groups of readings

H_0: all groups come from the same distribution

There's a common way to set out hypothesis tests for comparing groups (as well as for many similar tasks), called the Neyman-Pearson approach.

Neyman-Pearson hypothesis testing

Let x be the dataset.

Propose a general parametric model H_1, and express H_0 as a restriction on one or more parameters

1. Choose a test statistic based on mle estimates of the parameters of H_0 and H_1
2. Define a random synthetic dataset X^*, what we might see if H_0 were true.
3. Let p be the probability (assuming H_0 to be true) of seeing $t(X^*)$ as or more extreme than the observed $t(x)$.

A low p-value is a sign that H_0 should be rejected.

General model

H_1: $X_i \sim N(a, \sigma^2)$

$Y_i \sim N(b, \sigma^2)$

$Z_i \sim N(c, \sigma^2)$

H_0: $a = b = c$. All samples $\sim N(\mu, \sigma^2)$

$\left.\begin{array}{l}\hat{a} = \bar{x} \\ \hat{b} = \bar{y} \\ \hat{c} = \bar{z}\end{array}\right\}$ mles under H_1

$\hat{\mu} = \overline{\text{concat}(x, y, z)}$

can I invent a test statistic using these four parameters, which is liable to be bigger if H_0 is false?

More generally:

$$t = \frac{\max_{\text{params of } H_1} \Pr(\text{data} \mid H_1)}{\max_{\text{params of } H_0} \Pr(\text{data} \mid H_0)}$$

Exercise 9.3.2 (Equality of group means).
We are given three groups of observations from three different systems

$$x = [7.2, 7.3, 7.8, 8.2, 8.8, 9.5]$$
$$y = [8.3, 8.5, 9.2]$$
$$z = [7.4, 8.5, 9.0]$$

Do all three groups have the same mean?

fitted pars for full model

Test statistic:

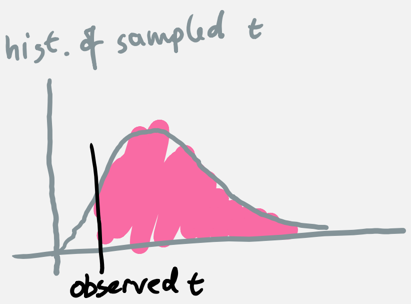
$$t = (\hat{a} - \hat{\mu})^2 + (\hat{b} - \hat{\mu})^2 + (\hat{c} - \hat{\mu})^2$$

Fitted under H_0

```
# 1. Define test statistic
def t(x,y,z):
    μ = np.mean(np.concatenate([x,y,z]))
    a,b,c = [np.mean(v) for v in [x,y,z]]
    return (a-μ)**2 + (b-μ)**2 + (c-μ)**2

# 2. To generate a synthetic dataset, assuming H₀ ...
xyz = np.concatenate([x,y,z])
μ̂ = np.mean(xyz)
σ̂ = np.sqrt(np.mean((xyz-μ̂)**2))
def rxyz_star():
    return (np.random.normal(size=len(x), loc=μ̂, scale=σ̂),
            np.random.normal(size=len(y), loc=μ̂, scale=σ̂),
            np.random.normal(size=len(z), loc=μ̂, scale=σ̂))

# 3. Sample the test statistic, find the p-value
t_ = np.array([t(*rxyz_star()) for _ in range(10000)])
p = np.mean(t_>=t(x,y,z))
```



EXERCISE.

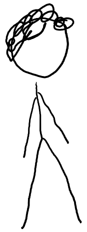
Consider the data for IA student marks.

a. What's a sensible H_0 to test?

b. What's a natural test statistic?

c. How might we generate a random synthetic dataset?

gender	mark
F	17
F	14
M	18
O	11
M	17
⋮	⋮



(a) Introduce a richer model H_1: Mark $\sim \mu_{\text{gender}} + N(0,\sigma^2)$

let H_0 be: $\mu_F = \mu_M = \mu_O$

(b) $t = (\hat\mu_F - \hat\mu)^2 + (\hat\mu_M - \hat\mu)^2 + (\hat\mu_O - \hat\mu)^2$ where $\hat\mu_F, \hat\mu_M, \hat\mu_O$ are MLE under H_1 and $\hat\mu$ is MLE under H_0

(c) Parametric resampling. Under H_0, Marks $\sim \hat\mu + N(0, \hat\sigma^2)$.

Conclusion: for the real marks from last year, $p = 0.71\%$ So we reject H_0.

NON-PARAMETRIC RESAMPLING

(a) H_0: marks for all three genders are drawn from the same distribution.

(c) If H_0 is true, then the best fit is the empirical distribution of all marks (concatenated together).
Let's simply resample from this.

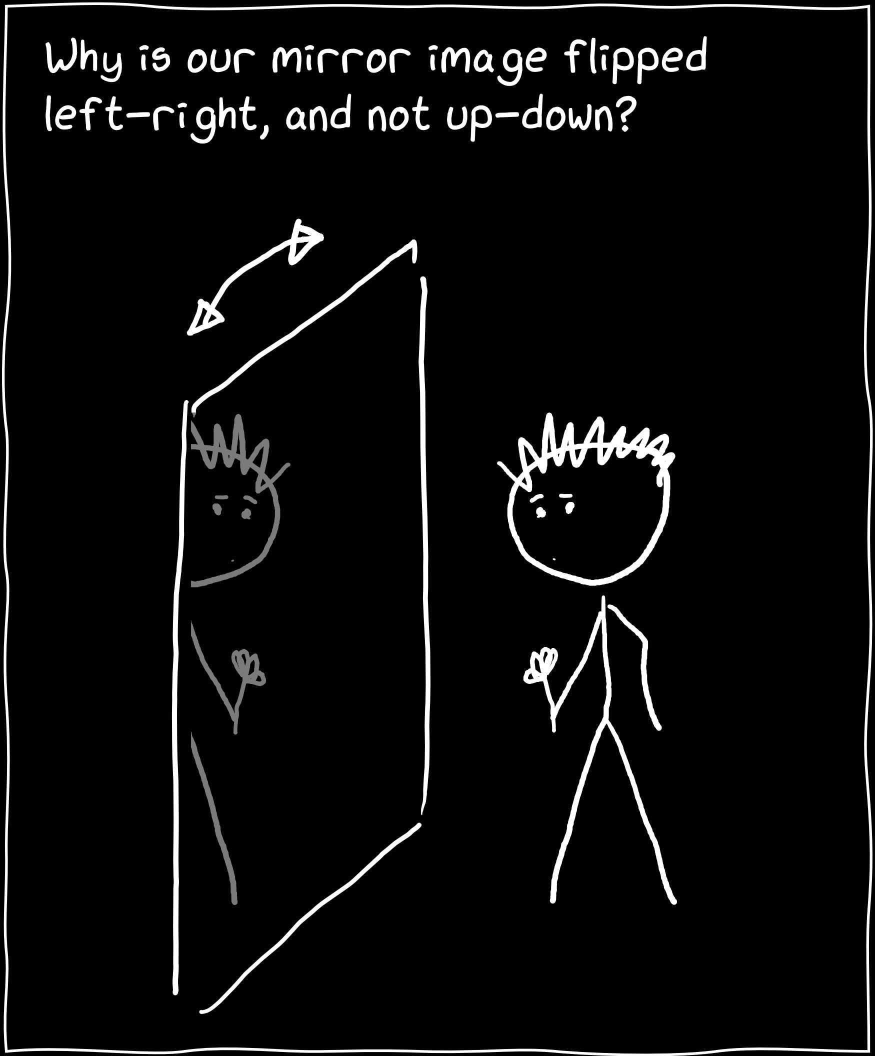
Conclusion: $p = 0.80\%$

PERMUTATION TESTING

(a) H_0: you'd get the same mark regardless of your gender.

(c) Imagine a parallel universe where every student gets assigned a random gender (25 Women, 110 Men, 5 Other).
Simulate this parallel universe by randomly permuting the gender column.

Conclusion: $p = 0.82\%$


Why is our mirror image flipped left-right, and not up-down?

IB Data Science syllabus

Using a probability model to describe data → Models that depend on linear combinations of features → Parameter interpretation and identifiability

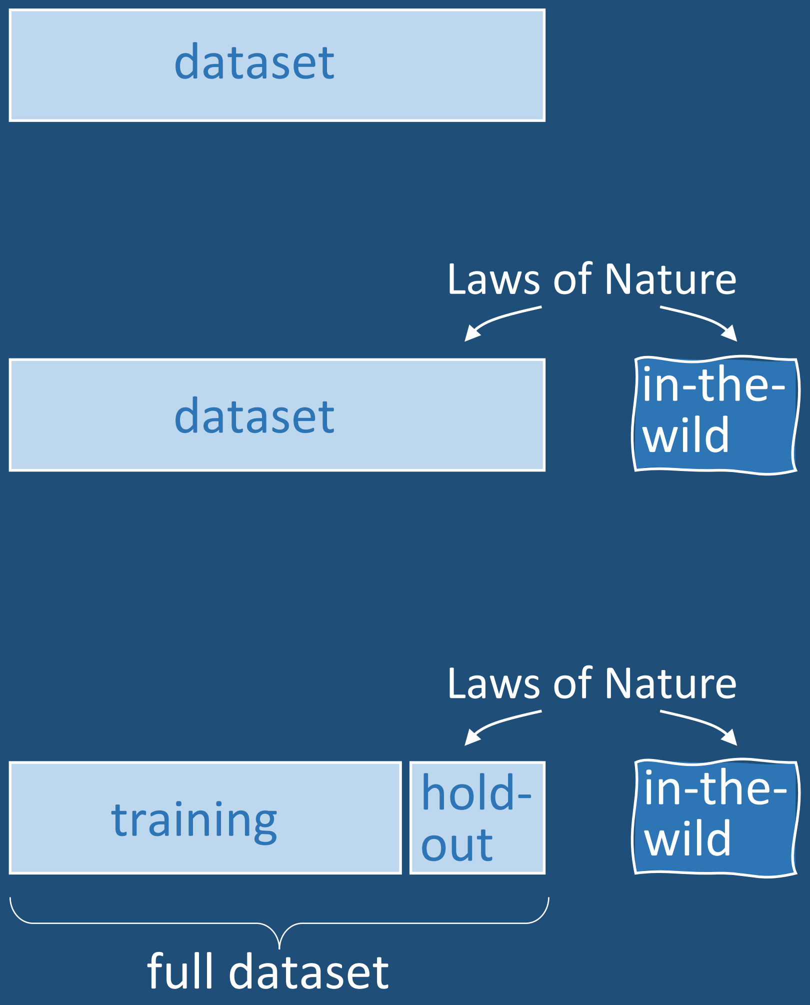
Fitting a model's unknown parameters using MLE → Fitting via least squares (when appropriate)

Reasoning about uncertainty →

Part III
(*non-examinable)

"Induction is the glory of Science and the scandal of Philosophy."

C.D. Broad, 1926



- Maximum likelihood estimation gives us a model that fits the training dataset

But how well will our model work on new data?
("The challenge of induction.")

- Bayesianism and frequentism address this by making careful claims about the Laws of Nature that generated the dataset.

- Alternatively, we could simply say "The performance on in-the-wild data is approximately the performance on holdout data."

Table 2: Results on HotpotQA distractor (dev). (+hyperlink) means usage of extra hyperlink data in Wikipedia. Models beginning with "–" are ablation studies without the corresponding design.

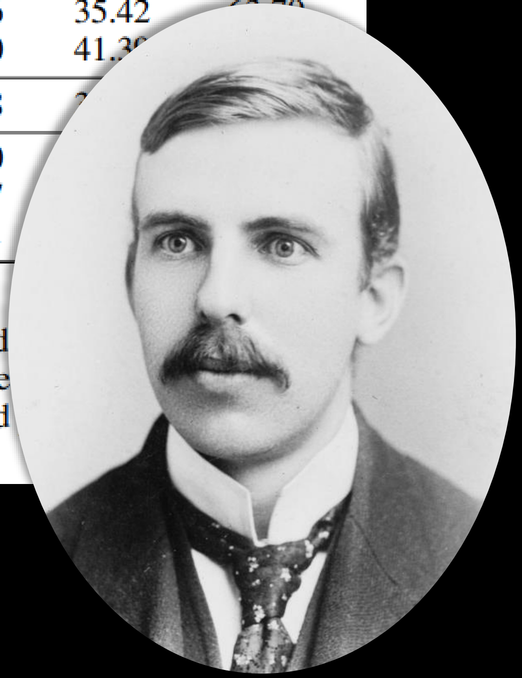
Model	Ans EM	Ans F_1	Sup EM	Sup F_1	Joint EM	Joint F_1
Baseline [53]	45.60	59.02	20.32	64.49	10.83	40.16
DecompRC [29]	55.20	69.63	N/A	N/A	N/A	N/A
QFE [30]	53.86	68.06	57.75	84.49	34.63	59.61
DFGN [36]	56.31	69.69	51.50	81.62	33.62	59.82
SAE [45]	60.36	73.58	56.93	84.63	38.81	64.96
SAE-large	66.92	79.62	61.53	86.86	45.36	**71.45**
HGN [14] (+hyperlink)	66.07	79.36	60.33	87.33	43.57	71.03
HGN-large (+hyperlink)	69.22	82.19	62.76	88.47	47.11	**74.21**
BERT (sliding window) variants						
BERT Plus	55.84	69.76	42.88	80.74	27.13	58.23
LQR-net + BERT	57.20	70.66	50.20	82.42	31.18	59.99
GRN + BERT	55.12	68.98	52.55	84.06	32.88	60.31
EPS + BERT	60.13	73.31	52.55	83.20	35.40	63.41
LQR-net 2 + BERT	60.20	73.78	56.21	84.09	36.56	63.68
P-BERT	61.18	74.16	51.38	82.76	35.42	[illegible]
EPS + BERT(large)	63.29	76.36	58.25	85.60	[illegible]	[illegible]
CogLTX	65.09	78.72	56.15	85.78	[illegible]	[illegible]
– multi-step reasoning	62.00	75.39	51.74	83.10	[illegible]	[illegible]
– rehearsal & decay	61.44	74.99	7.74	47.37	[illegible]	[illegible]
– train-test matching	63.20	77.21	52.57	84.21	[illegible]	[illegible]

Results. Table 2 shows that CogLTX outperforms most of previous method [illegible]
solutions on the leaderboard.[4] These solutions basically follow the frame [illegible]
results from sliding windows by extra neural networks, leading to bounded [illegible]
to insufficient interaction across paragraphs.

Most ML papers don't state an inductive claim.

Perhaps the authors haven't thought hard enough to be able to state one?

Perhaps they prefer to leave you, the reader, to make the inference?



"All science is either physics or stamp-collecting."

Ernest Rutherford (1871–1937)

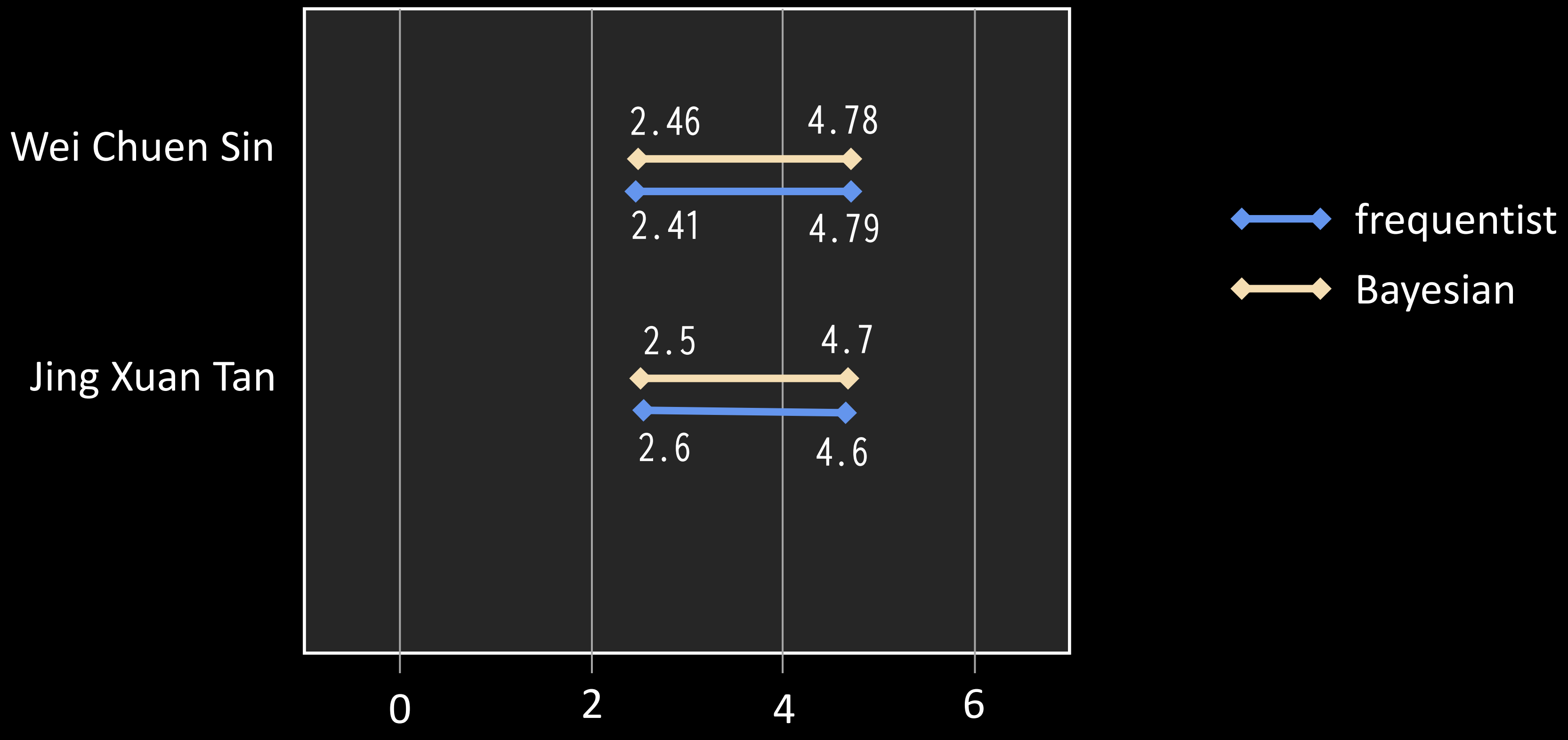
	model selection	**confidence intervals for model parameters**	**confidence intervals for predictions**
BAYESIANIST 1	Given two models, each with a prior weight, use the data to reweight the models	✓	✓
EMPIRICIST 2	Given two models, prefer the one that works better on holdout data	Who cares?	?
FREQUENTIST 3	Given a model, is it a good enough explanation of the data?	✓	✓

4

My personal approach:

1. If there's anything for which I have a justified prior belief, put it into my model as a random variable
2. Choose between competing models empirically
3. Check my final model using frequentist tests
4. Read off confidence intervals, using Bayesianism or frequentistism as appropriate.

Challenge: find a 95% confidence interval for the rate of temperate increase in Cambridge from 1985 to the present, in °C/century



Is CoPilot Bayesianist or frequentist?

```python
import numpy as np

x = np.array([7.2, 7.3, 7.8, 8.2, 8.8, 9.5])

# Assuming x is sampled from an Exponential distribution with rate lambda,
# find a 95% confidence interval for lambda.

def confint(x):
    n = len(x)
    xbar = np.mean(x)
    s = np.std(x, ddof=1)
    t = 2.776
    return xbar - t*s/np.sqrt(n), xbar + t*s/np.sqrt(n)
```

CoPilot likes to give me a frequentist confidence interval for the mean of a Gaussian distribution, but I can’t persuade it to give any other answer.

```python
import numpy as np

x = np.array([7.2, 7.3, 7.8, 8.2, 8.8, 9.5])
y = np.array([3, 2.5, 7.3])

# Test if x and y come from the same distribution
def test_same_distribution(x, y):
    # Test if x and y come from the same distribution
    from scipy.stats import ks_2samp
    ks = ks_2samp(x, y)
    print(ks)
    if ks[1] > 0.05:
        print('Same distribution')
    else:
        print('Different distribution')
```

CoPilot knows a few library calls for hypothesis testing, but it doesn’t know any substance.

ChatGPT4 gives textbook-like dumps with many different choices, but gives spurious answers (including hallucinated library calls) when I ask for specifics.