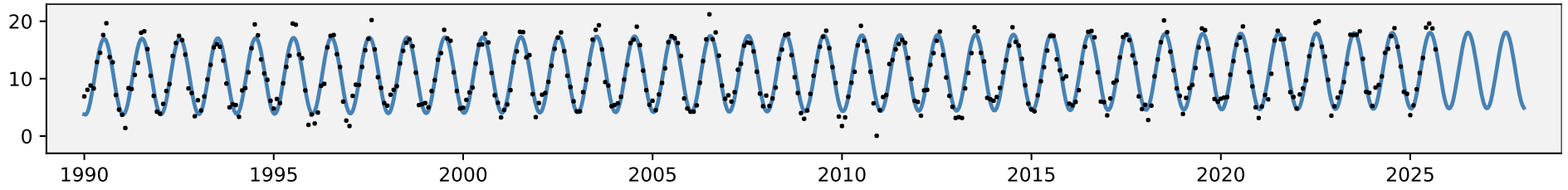
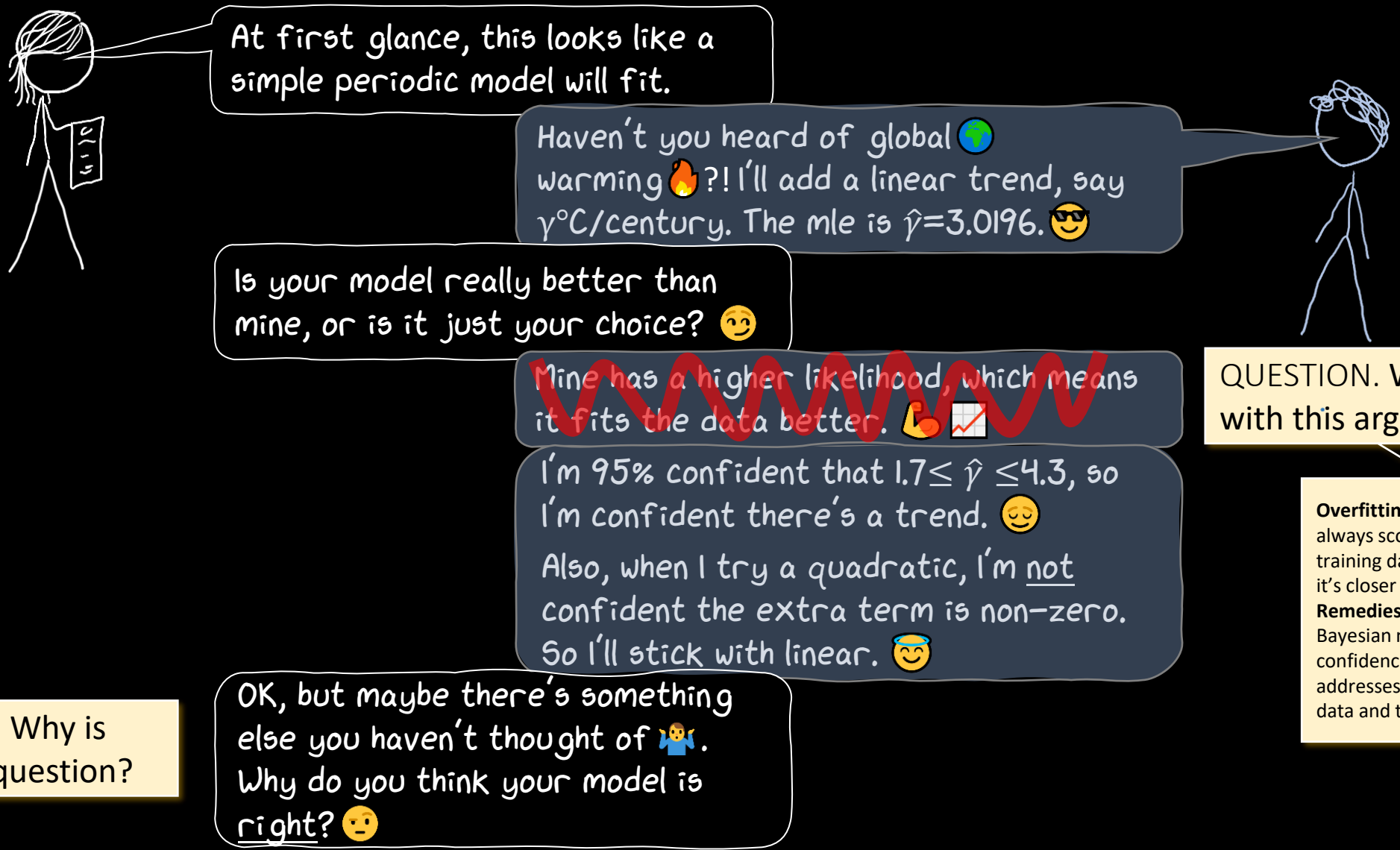
Iterative model development





QUESTION. What's wrong with this argument?

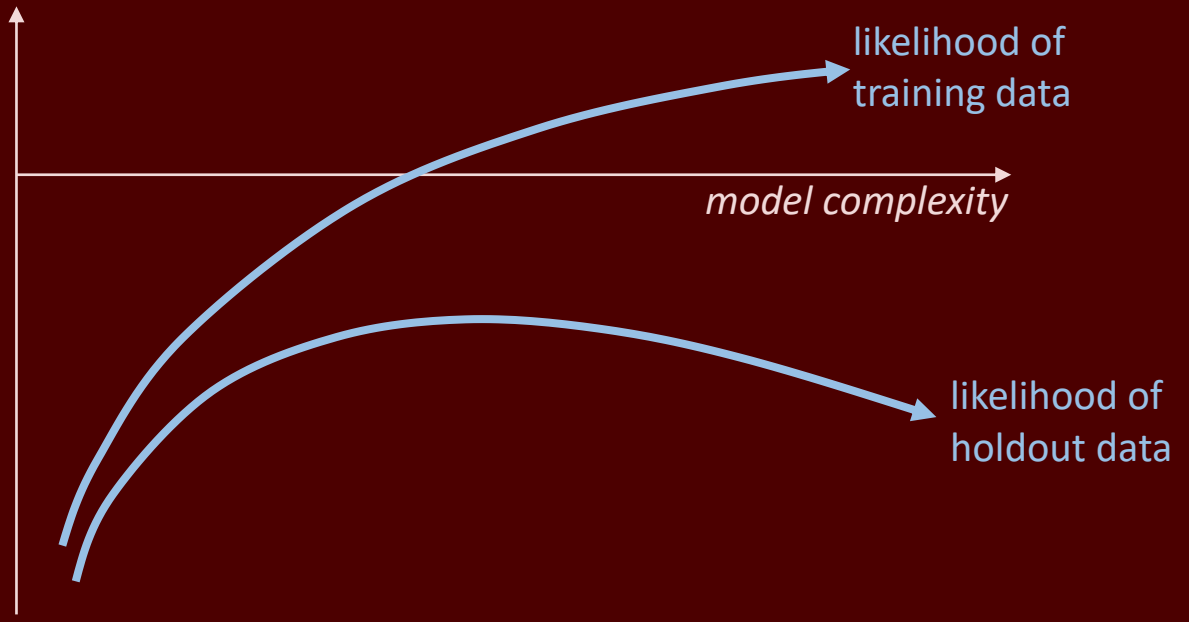
QUESTION. Why is this a silly question?

Overfitting! A more complex model always scores a higher likelihood on the training dataset. But that doesn't mean it's closer to the true distribution.
Remedies: holdout comparison, or Bayesian model choice, frequentist confidence intervals ... anything that addresses the difference between the data and the truth.

OVERFITTING AND HOLDOUT EVALUATION

A too-complex model will typically fit the dataset well,
but generalize poorly because it doesn't match the true probability model.

We can measure this by using holdout data
to approximate the true distribution.



```python
# Periodic model
model0 = sklearn.linear_model.LinearRegression()
def X0(t): return np.column_stack([np.sin(2*π*t), np.cos(2*π*t)])
model0.fit(X0(df.t), df.temp)
```

```python
# Model with linear trend
model1 = sklearn.linear_model.LinearRegression()
def X1(t): return np.column_stack([np.sin(2*π*t), np.cos(2*π*t), (t-2000)/100])
model1.fit(X1(df.t), df.temp)
model1.coef_[-1]
```

```python
# Compare log likelihoods
def loglik(ε):
    σ = np.sqrt(np.mean(ε**2))
    return - 1/2*np.sqrt(2*π*σ**2) - 1/(2*σ**2)*np.mean(ε**2)

loglik(df.temp - model0.predict(X0(df.t))), loglik(df.temp - model1.predict(X1(df.t)))
```

```python
# Periodic model
model0 = sklearn.linear_model.LinearRegression()
def X0(t): return np.column_stack([np.sin(2*π*t), np.cos(2*π*t)])
model0.fit(X0(df.t), df.temp)
```

```python
# Model with linear trend
model1 = sklearn.linear_model.LinearRegression()
def X1(t): return np.column_stack([np.sin(2*π*t), np.cos(2*π*t), (t-2000)/100])
model1.fit(X1(df.t), df.temp)
model1.coef_[-1]
```

```python
# Confidence interval for the trend term

# 1. Define the readout statistic, yhat
m = sklearn.linear_model.LinearRegression()
x = X1(df.t)
def trend(temp): return m.fit(x,temp).coef_[-1]

# 2. Fit the model, and generate resampled data based on the fitted model
pred1 = model1.predict(X1(df.t))
ε = df.temp - model1.predict(X1(df.t))
σ = np.sqrt(np.mean(ε**2))
def tempstar(): return np.random.normal(loc=pred1, scale=σ)

# 3. Find the spread of the readout statistic
γ_ = [trend(tempstar()) for _ in range(20000)]
lo,hi = np.quantile(γ_,[.025,.975])
lo,hi
```

Once we've tried all the models we can think of and chosen the best, what next?

1. Eyeball our model's fit, to see if we can spot any specific improvements
    - Compute the residuals $\varepsilon_i = y_i - \text{pred}(x_i)$ and look for patterns
    - More generally, compute the likelihood of each datapoint, and investigate datapoints that our model thinks are unlikely
2. Ask: is it plausible that our model might have generated the dataset?

RESEARCH QUESTION

Can you taste the difference between milk-first versus tea-first?

EXPERIMENT

Make 4 cups of each style, shuffle them, and ask taster to label them

RESEARCH QUESTION

Can you taste the difference between milk-first versus tea-first?

EXPERIMENT

Make 4 cups of each style, shuffle them, and ask the taster to label them

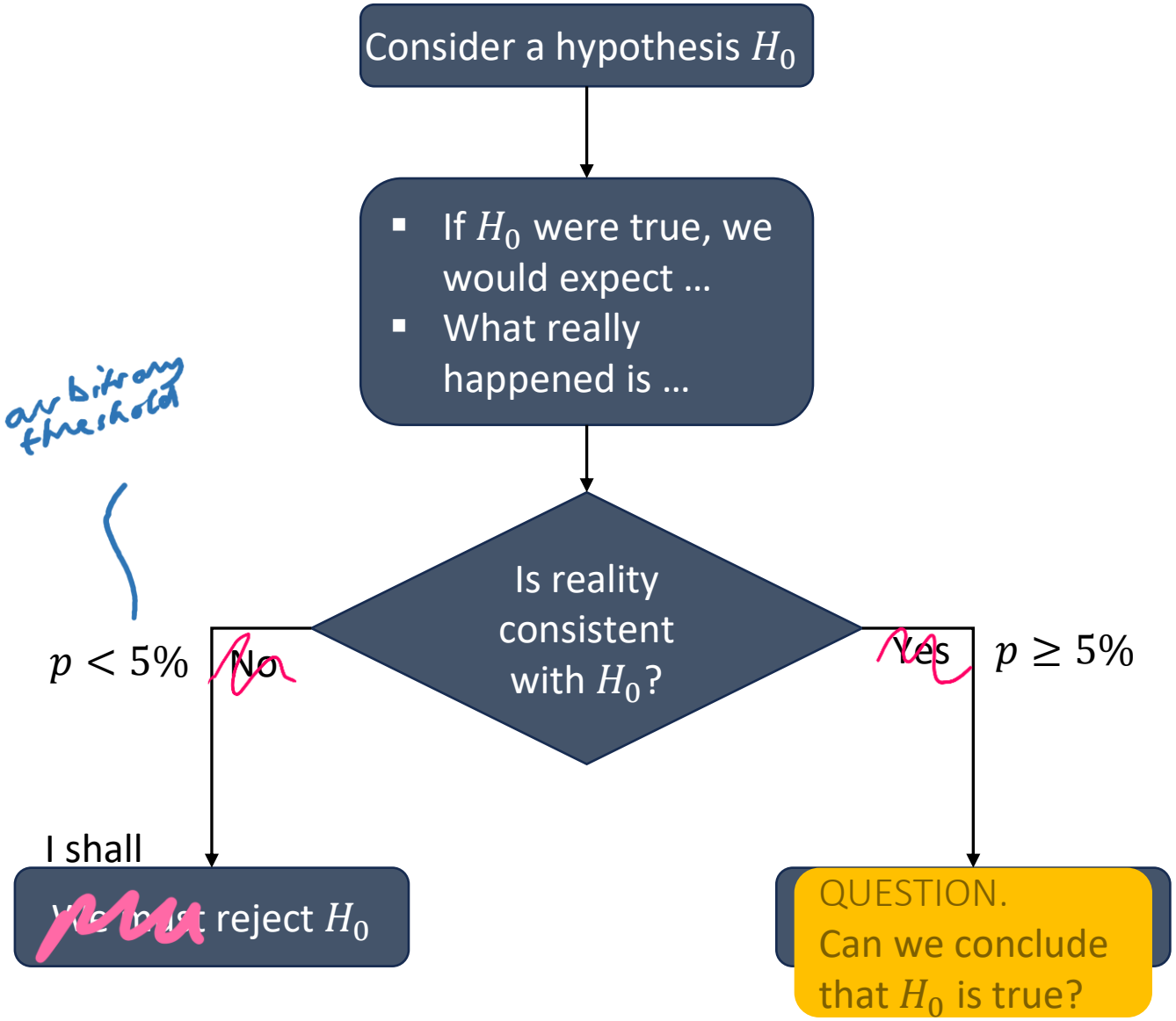
DATASET

8 pairs of (truth,label)

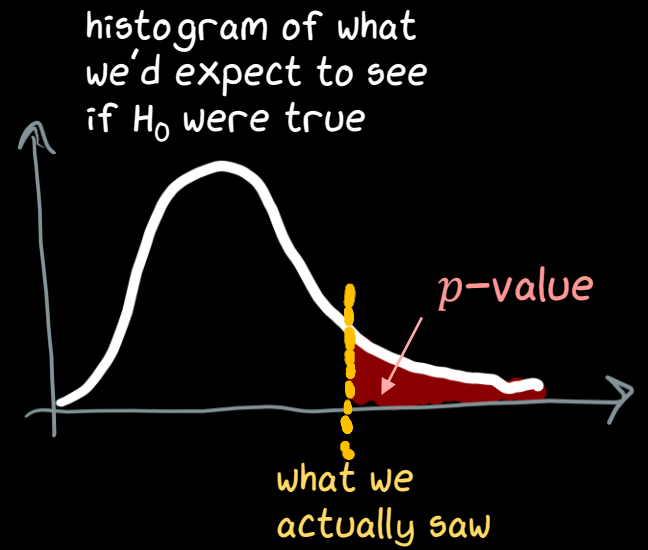
HYPOTHESIS / MODEL

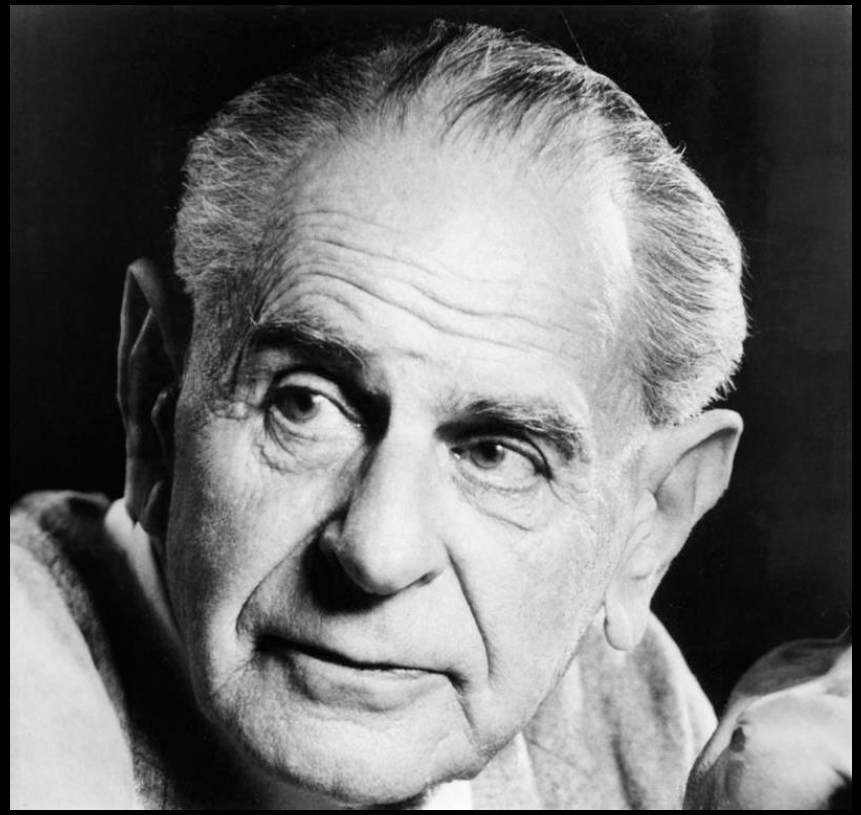
There's no difference. Thus the dataset arose by chance, from purely random choice

Hypothesis testing asks whether a proposed probability model H_0 is consistent with the dataset.



- Because of noise, this isn't a simple yes/no decision.
- It's about *degree of consistency,* which we measure by the p-value. Instead of yes/no, we can go by $p < 5\%$ / $p \geq 5\%$.
- The p-value is the probability, according to the model H_0, of seeing something at least as extreme as what we actually saw.





"Every genuine scientific theory must be falsifiable. It is easy to obtain evidence in support of virtually any theory; the evidence only counts if it is the positive result of a genuinely risky prediction."

Karl Popper (1902–1994)

Fisher's hypothesis testing

Let x be the dataset.

State a null hypothesis H_0, i.e. a probability model for the dataset

1. Choose a test statistic
$$t : \text{dataset} \mapsto \mathbb{R}$$

2. Define a random synthetic dataset X^*, what we might see if H_0 were true.

3. Look at the histogram of $t(X^*)$.
Let p be the probability of seeing a value as extreme or more so than the observed $t(x)$.

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

cup id	truth	taster
1	milk	milk
2	tea	tea
⋮		

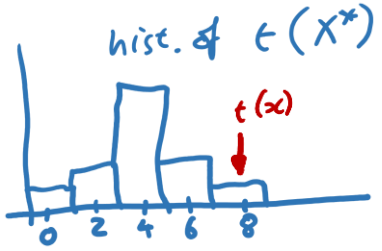
x = taster's assignment of labels

H_0: the taster can't tell the difference and just assigned labels randomly

t(x) = #correct

def X*(): return random permutation of {t,t,t,t,m,m,m,m}

What would be the distribution of the test statistic if H_0 were true?



$p = \mathbb{P}(t(X^*) \geq t(x)) = 1.4\%$

p<5% so we shall reject H_0.
"The evidence against H_0 is statistically significant (p=1.4%)"

Example 9.6.2.

I have a dataset with readings from two groups, $x = [x_1, \ldots, x_m]$ and $y = [y_1, \ldots, y_n]$. Test whether the two groups are significantly different, using the test statistic $\bar{y} - \bar{x}$.

Dataset: (x, y)

H_0: x and y are samples from the same distribution.

Test statistic: given in the question.

```
1  # 1. Define the test statistic
2  def t(x,y): return np.mean(y) - np.mean(x)
3
4  # 2. To generate a synthetic dataset, assuming H0, ...
5  xy = np.concatenate([x,y])
6  def rxy_star():
7      return (np.random.choice(xy, size=len(x)),
8              np.random.choice(xy, size=len(y)))
9
10 # 3. Sample the test statistic under H0; find p-value for observed data
11 t_ = np.array([t(*rxy_star()) for _ in range(10000)])
12 p = …
```

Example 9.3.1.

I have a dataset with readings from two groups, $x = [x_1, \dots, x_m]$ and $y = [y_1, \dots, y_n]$. Test whether the two groups are significantly different, using the test statistic $\bar{y} - \bar{x}$.

H_0: x, y are both sampled from $N(\mu, \sigma^2)$ (μ, σ unknown).

EQUIVALENTLY

General model:

X sampled from $N(\mu, \sigma^2)$

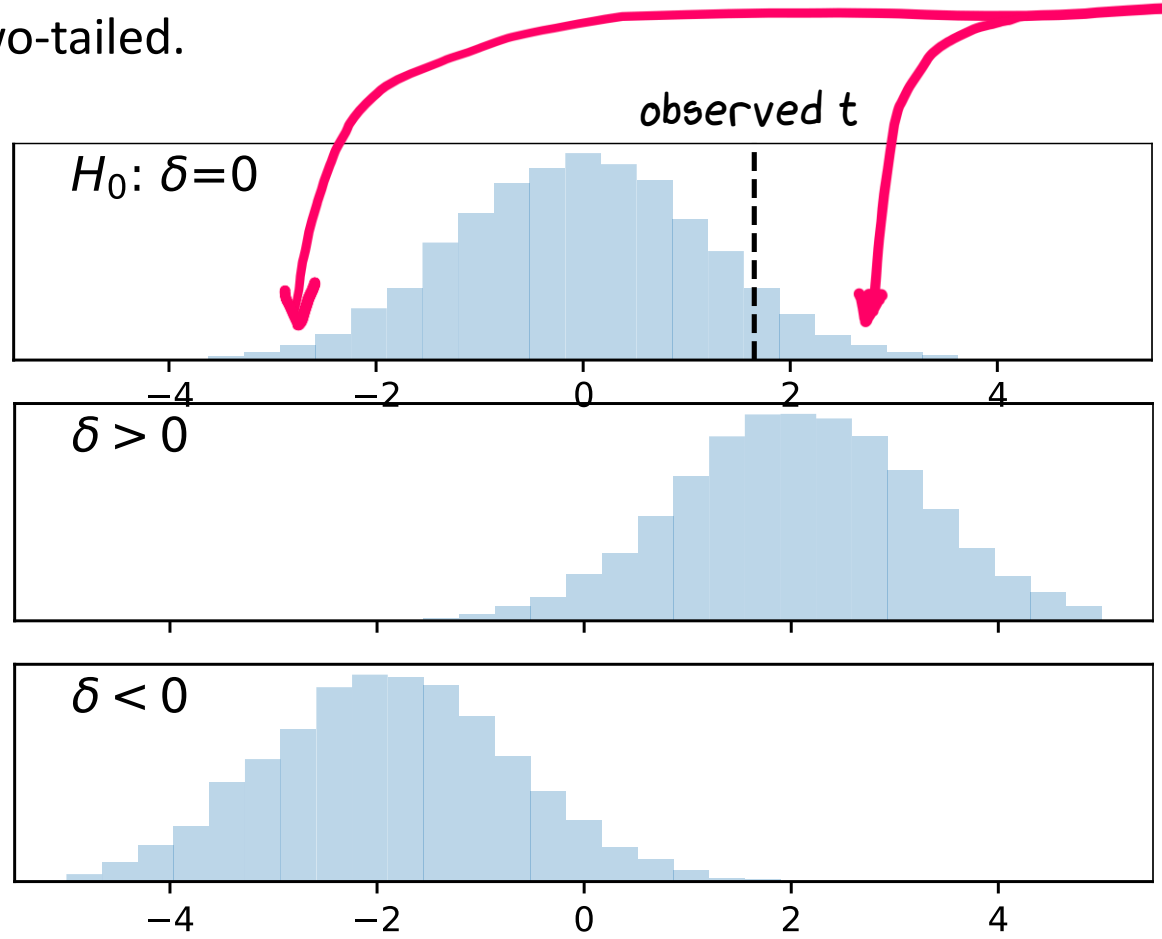
Y sampled from $N(\mu+\delta, \sigma^2)$

H_0: $\delta = 0$.

```
1  # 1. Define the test statistic
2  def t(x,y): return np.mean(y) - np.mean(x)
3
4  # 2. To generate a synthetic dataset, assuming H0, ...
5  xy = np.concatenate([x,y])
6  μ̂ = np.mean(xy)
7  σ̂ = np.sqrt(np.mean((xy - μ̂)**2))
8  def rxy_star():
9      return (np.random.normal(loc=μ̂, scale=σ̂, size=len(x)),
10             np.random.normal(loc=μ̂, scale=σ̂, size=len(y)))
11
12 # 3. Sample the test statistic under H0; find p-value for observed data
13 t_ = np.array([t(*rxy_star()) for _ in range(10000)])
14 p = 2 * min(np.mean(t_ >= t(x,y)), np.mean(t_ <= t(x,y)))
```

What counts as ‘more extreme’?

- Plot the histogram for $t(X^*)$, assuming H_0 is true
- Also plot the histogram for some scenarios where H_0 is false
- Do the alternatives push $t(X^*)$ bigger, or smaller, or either? This determines what ‘more extreme’ means — either one-tailed or two-tailed.

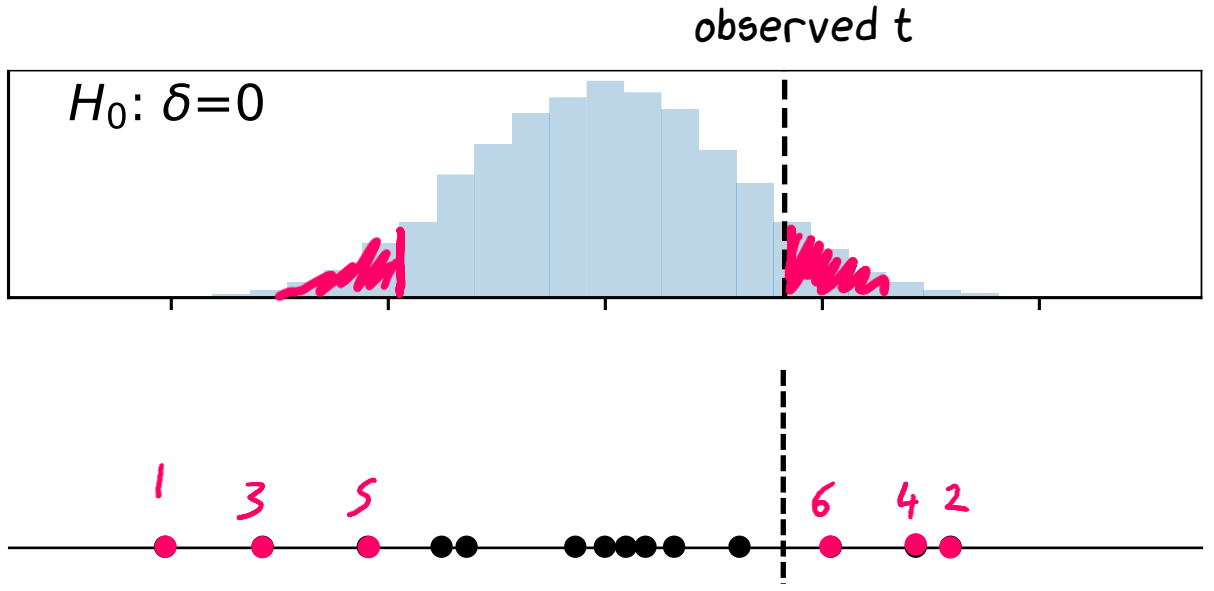


If the observed t lies at either extreme, it's evidence against H_0: δ=0

How do we compute p for a two-tailed test?

The p-value is

$$\mathbb{P}\left(\begin{array}{c} t(X^*) \text{ at least} \\ \text{as extreme as } t(x) \end{array} \;\middle|\; H_0 \text{ is true}\right)$$



"6 of my samples of t(X*,Y*) are more extreme than t(x,y)."

```
p = 2 * min(np.mean(t_ >= t(x,y)), np.mean(t_ <= t(x,y)))
```

Where do test statistics come from?

Indeed, why do we even need test statistics?
We want to know if the dataset is plausible according to H_0,
so why not simply measure its likelihood under H_0?

—this is answered in §9.4 of lecture notes.

Example 9.3.1.
I have a dataset with readings from two groups, $x = [x_1, \dots, x_m]$ and $y = [y_1, \dots, y_n]$. Test whether the two groups are significantly different, using the test statistic $\bar{y} - \bar{x}$.

In some problems, it's natural to express H_0 as a *constraint* on the parameters of a more general model H_1.

H_1: $x \sim N(\mu, \sigma^2)$
$y \sim N(\nu, \sigma^2)$

H_0: $\mu = \nu$

- **For the test statistic:**
  estimate the parameters under H_1, and let t measure how close they are to meeting the constraint.

$\hat{\mu} = \bar{x}$, $\hat{\nu} = \bar{y}$

$t = |\hat{\nu} - \hat{\mu}| = |\bar{y} - \bar{x}|$ one-tailed test reject H_0 if t large

$t = \bar{y} - \bar{x}$, two-tailed test.

- **For resampling:**
  Fit H_0 by maximizing the likelihood of the parameters under the constraint. Then, sample from this fitted model.

$\max_{\mu, \nu, \sigma : \mu = \nu} \log \Pr(\text{data}; \mu, \nu, \sigma)$

equiv. to $\max_{\lambda, \sigma} \log \Pr(\text{data}; \lambda, \lambda, \sigma)$

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?

QUESTION. Why did I choose a common variance?

ANSWER: I didn't have to, and it'd be wiser to propose an H_1 that has different variances.

(Though, with such a tiny dataset, it's extremely unlikely we'll see evidence that rejects the hypothesis of equal variance.)

H_1: $X \sim N(a, \sigma^2)$ $Y \sim N(b, \sigma^2)$ $Z \sim N(c, \sigma^2)$

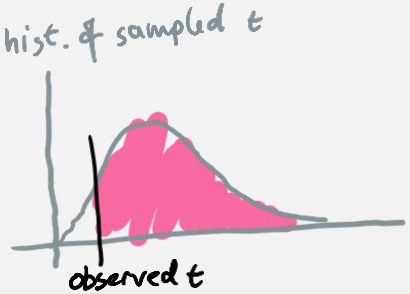
H_0: $a = b = c$

$\hat{a} = \bar{x}$ $\hat{b} = \bar{y}$ $\hat{c} = \bar{z}$

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

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

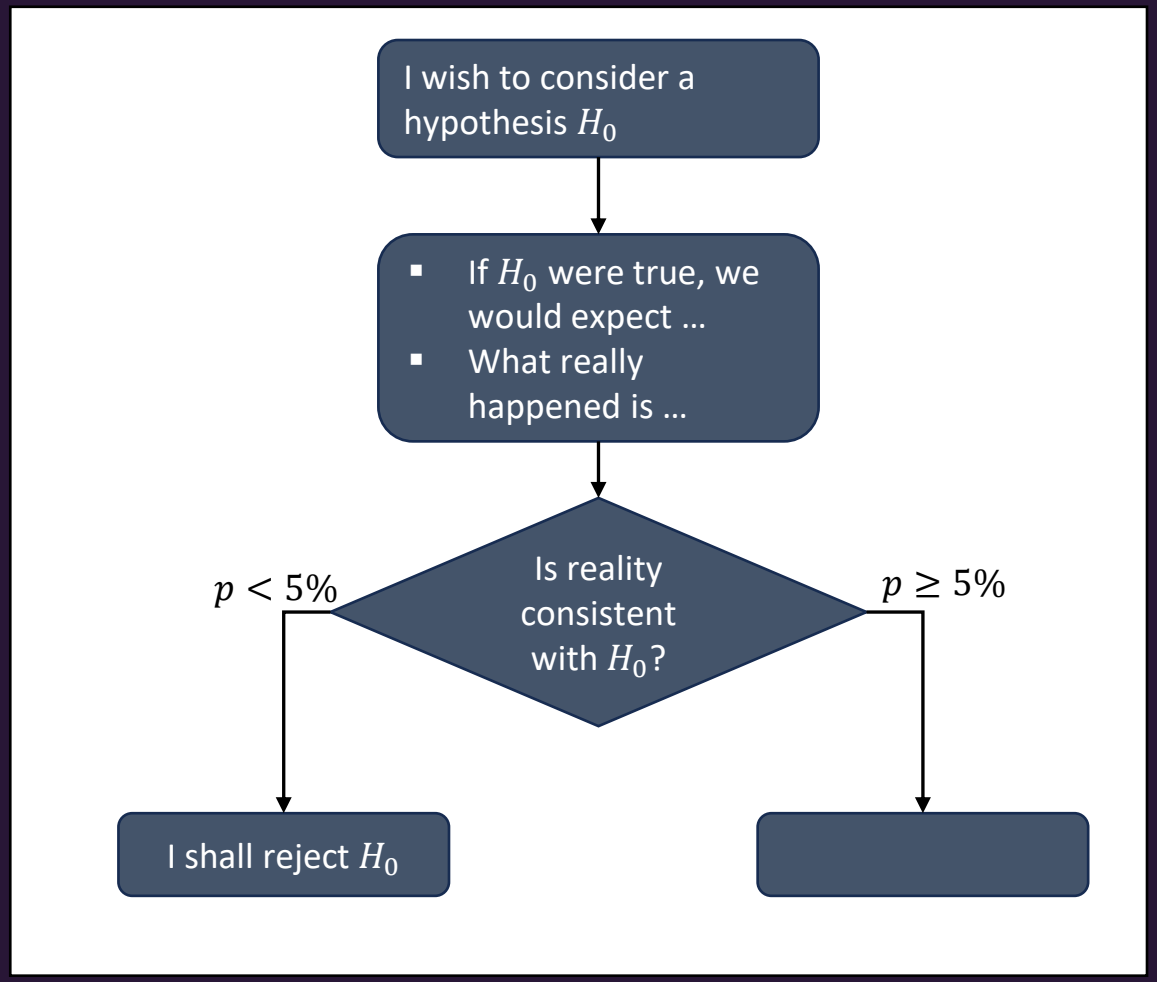
where $\hat{\mu}$ is MLE of H_0: all $N(\mu, \sigma^2)$




```python
# 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 H_0 ...
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))
```



"Data science is quantitative rhetoric"

We only get a definite publishable conclusion if we reject H_0.

Anything we want to argue, we have to phrase it as "reject H_0" for a suitable H_0.

EXERCISE.

Here are marks for IA Algorithms questions.
I think there might be a gender bias.
What H_0 do I choose?

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

H_0: I think everyone gets pretty much the same marks, regardless of gender

Null hypothesis:

Let's introduce a richer model, H_1: $\text{Mark} \sim \mu_{\text{gender}} + N(0, \sigma^2)$
and express H_0 as a constraint: $\mu_F = \mu_M = \mu_O$

H_1: I think gender affects marks

Test statistic: $t = (\hat{\mu}_F - \hat{\mu})^2 + (\mu_M - \hat{\mu})^2 + (\hat{\mu}_O - \hat{\mu})^2$ where $\hat{\mu}_F, \hat{\mu}_M, \hat{\mu}_O$ are MLE under H_1 and $\hat{\mu}$ is MLE under H_0

Resampling: Fit H_0 which says $\text{Mark} \sim N(\mu, \sigma^2)$ then sample from this fitted $N(\hat{\mu}, \hat{\sigma}^2)$

Here are marks for IA Algorithms questions.
I think there might be a gender bias.
What H_0 do I choose?

Under the general H_1 model

$$\text{Mark} \sim \mu_{\text{gender}} + N(0, \sigma^2)$$

I propose $H_0: \mu_M = \mu_F = \mu_O$

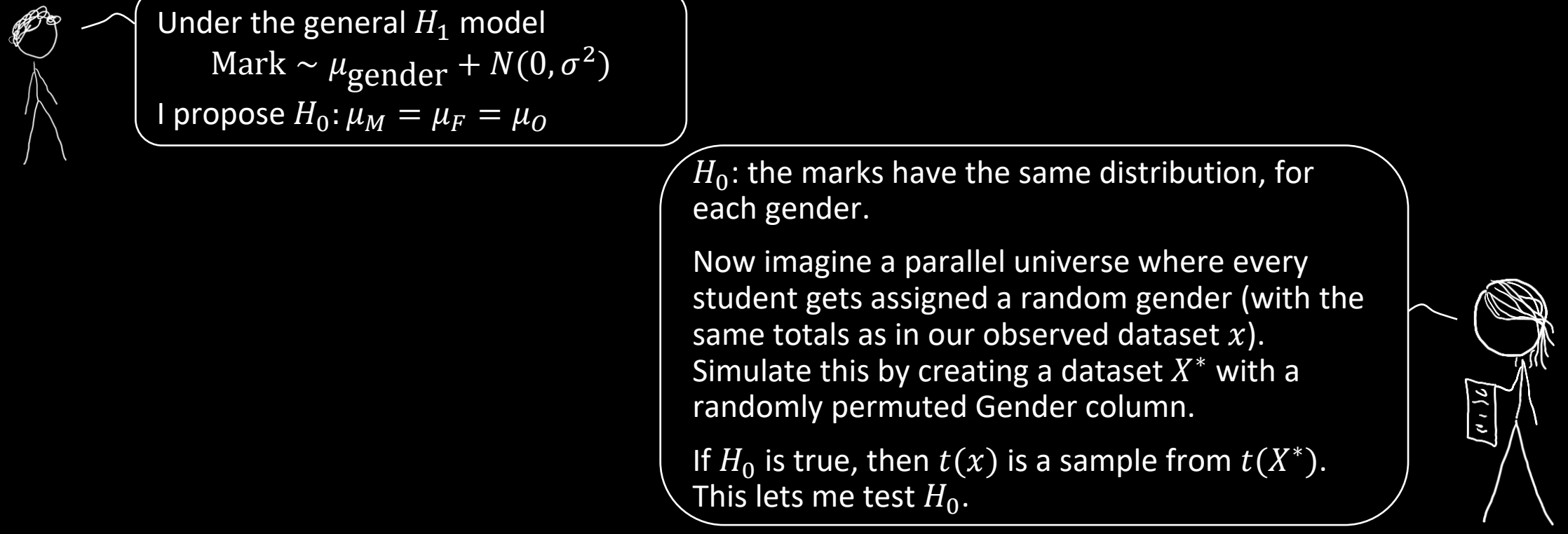
gender	mark
F	17
F	14
M	18
O	11
M	17
⋮	⋮

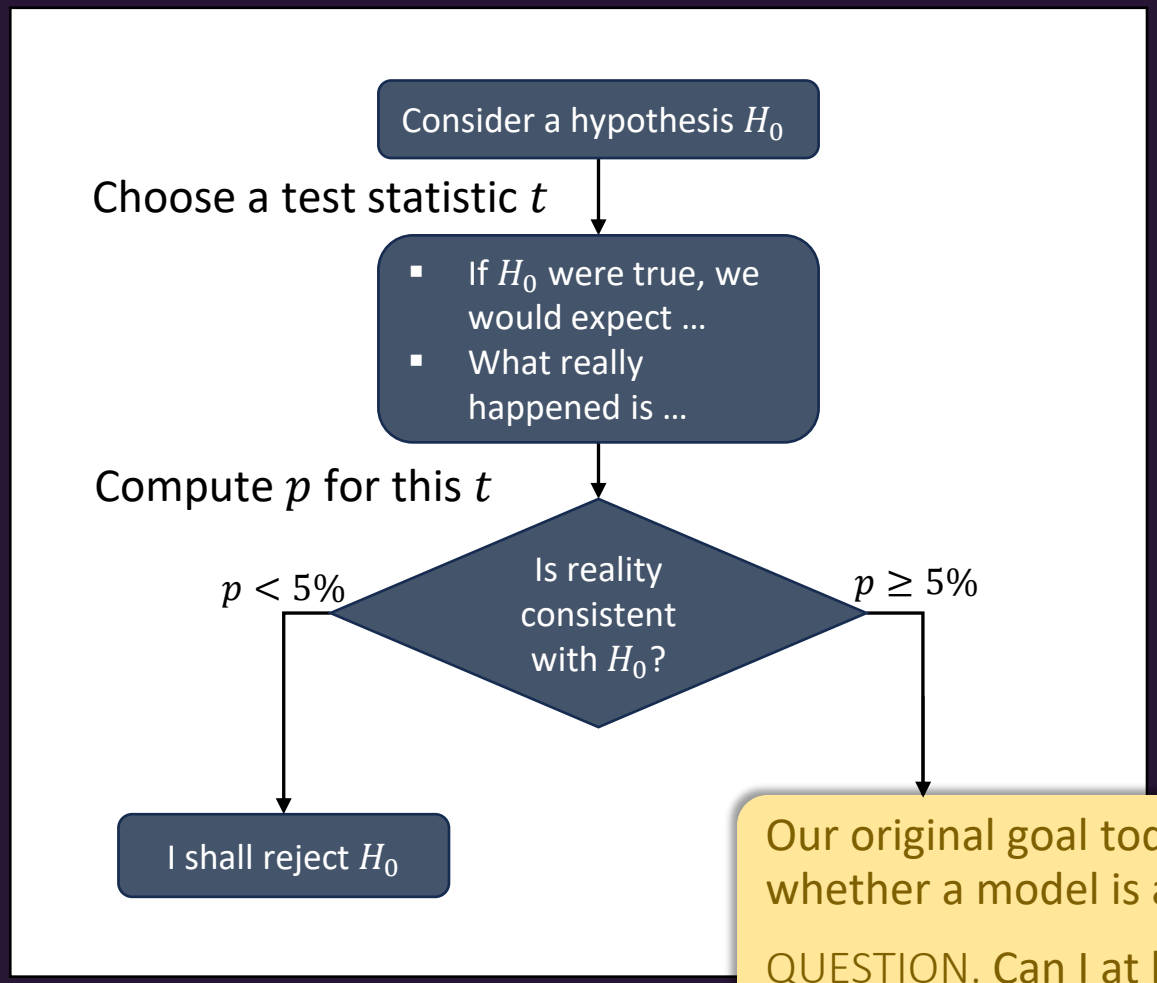
QUESTION. Suppose we reject H_0. Does this mean we believe that the means μ_g are not all equal?

- Our H_0 claims several things at the same time:
  (means are equal) & (variances are equal) & (all are Gaussian) & (all are independent)
- If we reject H_0, we reject it in its entirety.
  We might be rejecting it because of evidence against any one or more of its subclaims.

What makes a good hypothesis test?

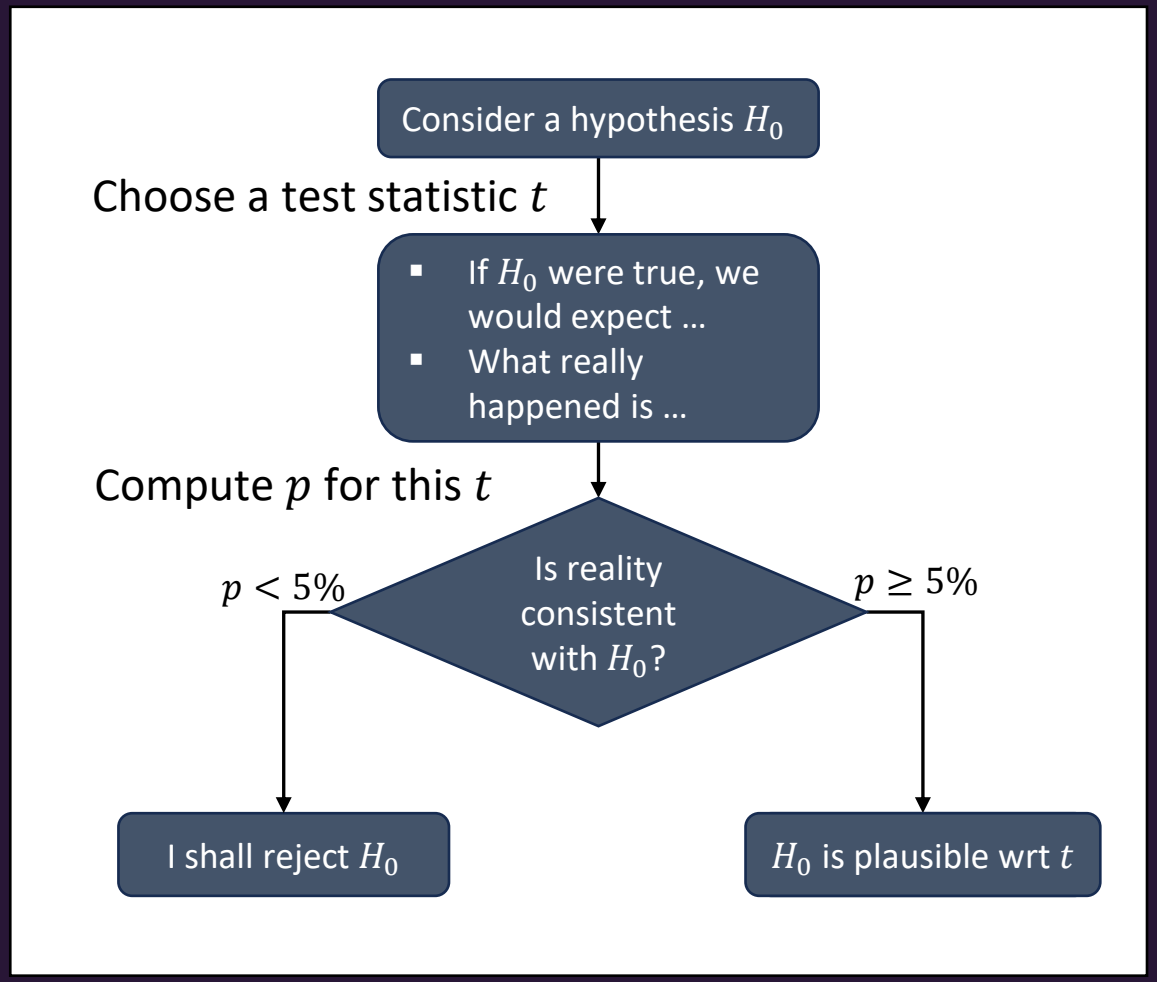
- Your H_0 is credible to your audience.
  If you propose a non-credible H_0 and then reject it — who cares?
- Your H_0 matches the research question you want to answer,
  and doesn't bring in contentious subclaims.
- Your resampling method assumes H_0 and nothing more.





Our original goal today was to decide whether a model is adequate.

QUESTION. Can I at least conclude that "H_0 could plausibly have generated the dataset"?



Different t are sensitive to different violations of H_0.

If I settle for my H_0 and then someone comes up with a better model, I lose. So it's on me to test H_0 using several test statistics.

NEXT LECTURE

THE GREAT GENERALIZATION

SMACKDOWN

BAYESIANIST V FREQUENTIST