Scientific Inference: Hypothesis Testing

Statistics for Data Science
CSE357 - Fall 2021

Goal of Data Science

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

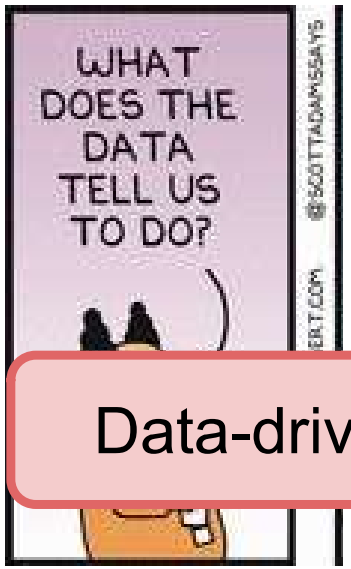
The Data Whisperer

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

Goal of Data Science

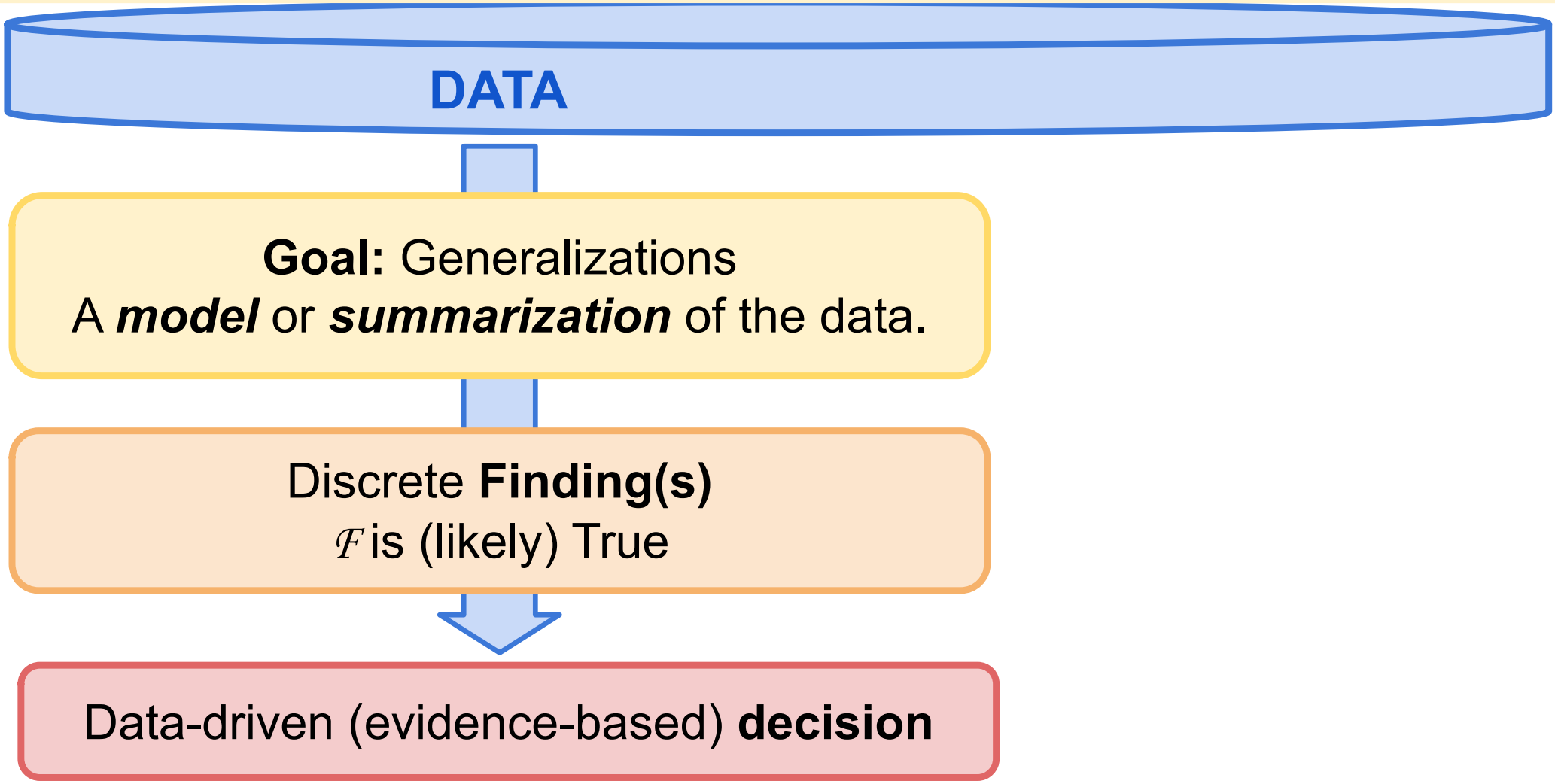
DATA

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

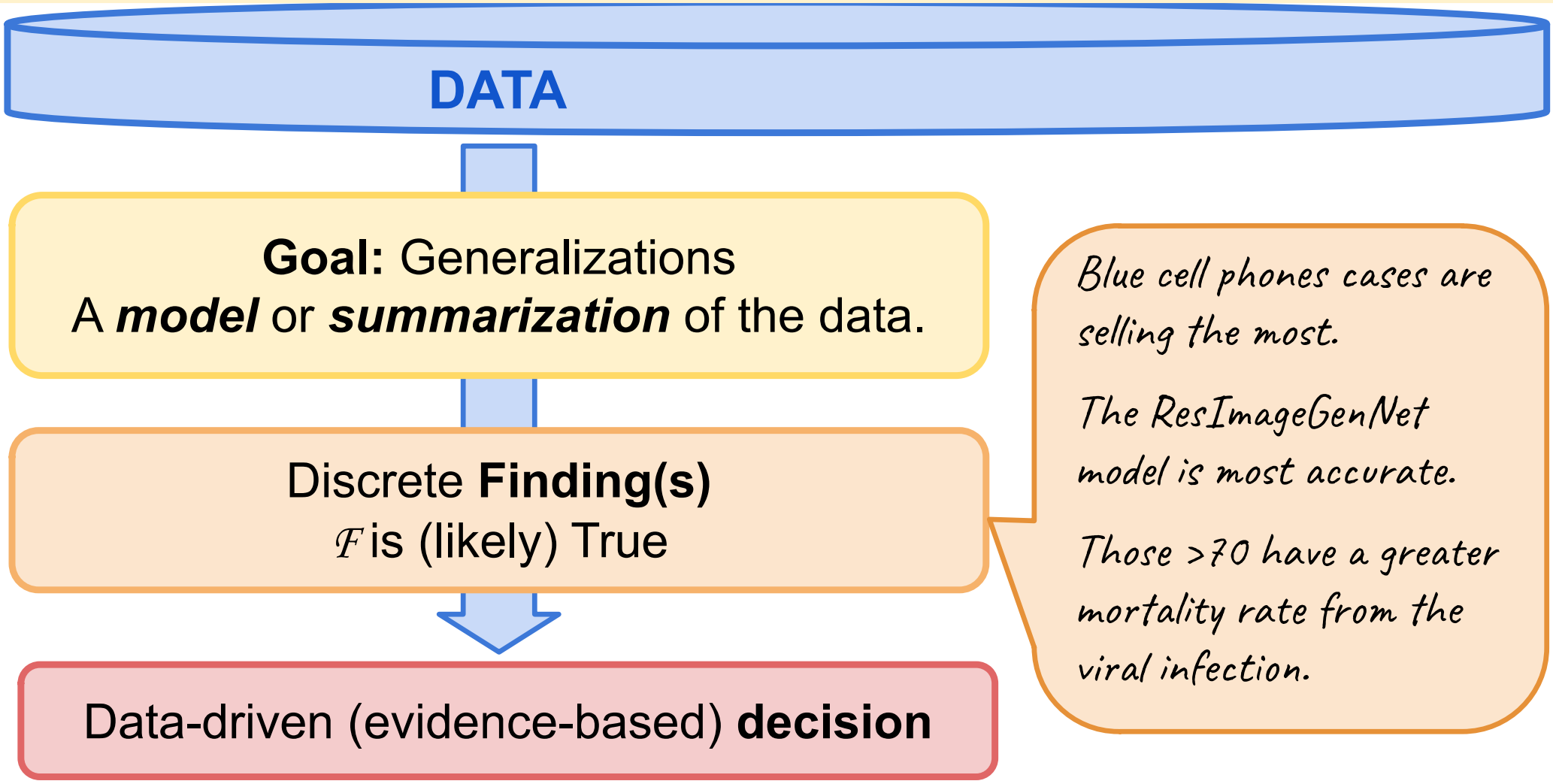


Data-driven (evidence-based) **decision**

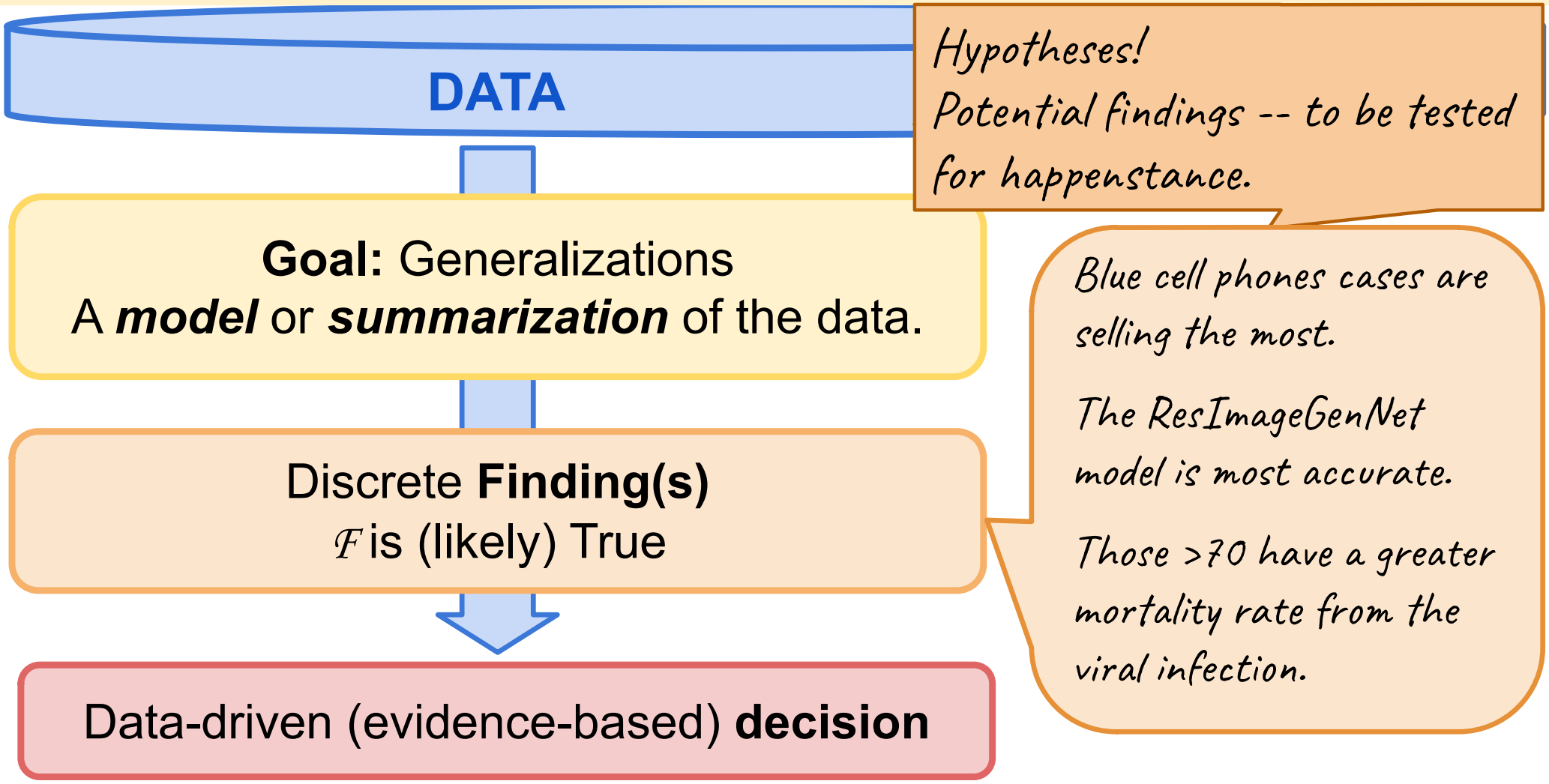
Goal of Data Science



Goal of Data Science



Goal of Data Science



The Data Whisperer

Hypotheses!
Potential findings -- to be tested for happenstance.

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

The Data Whisperer

Hypotheses!
Potential findings -- to be tested for happenstance.

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

The Data Whisperer

Hypotheses!
Potential findings -- to be tested for happenstance.

Conceptual Goal of Scientific Inference:
Determine the truth.

The Data Whisperer

Hypotheses!
Potential findings -- to be tested for happenstance.

Conceptual Goal of Scientific Inference: Determine the truth.

How?

- *Null Hypothesis Significance Testing (Neyman-Pearson)*
  confidence intervals (frequentist)
- *Likelihood-based Inference (Fischer)*
- *Bayesian Hypothesis Testing*
  confidence intervals (Bayesian)

Hypothesis Testing

Also known as… “Don’t be Dilbert’s Boss!”

Hypothesis -- something one asserts to be true.

Hypothesis Testing

Also known as… “Don’t be Dilbert’s Boss!”

Hypothesis -- something one asserts to be true.

Formally, two types:

H_0*: null hypothesis* -- some “default” value; “null”: nothing changes

H_1: the alternative -- the opposite of the null => a change or difference

Hypothesis Testing

H_0: null hypothesis -- some “default” value; “null”: nothing changes

H_1: the alternative -- the opposite of the null => a change or difference

Goal: Make sure what we observed was unlikely to happen by chance.

Thus, we want to know:

Given null, what is the probability of the observation or worse

3 rolls of 6-sided die: 3, 2, 6: observed ⅓; null: <=⅙

Hypothesis Testing

H_0: *null hypothesis* -- some "default" value; "null": nothing changes

H_1: the alternative -- the opposite of the null => a change or difference

Goal: Make sure what we observed was unlikely to happen by chance.

Thus, we want to know:

Given null, what is the probability of the observation or worse?

-> If low enough, then we "reject the null (H_0) in favor of H_1."

Hypothesis Testing

H_0: *null hypothesis* -- some "default" value; "null": nothing changes

H_1: the alternative -- the opposite of the null => a change or difference

Goal: Make sure what we observed was unlikely to happen by chance.

Thus, we want to know:

Given null, what is the probability of the observation or worse?

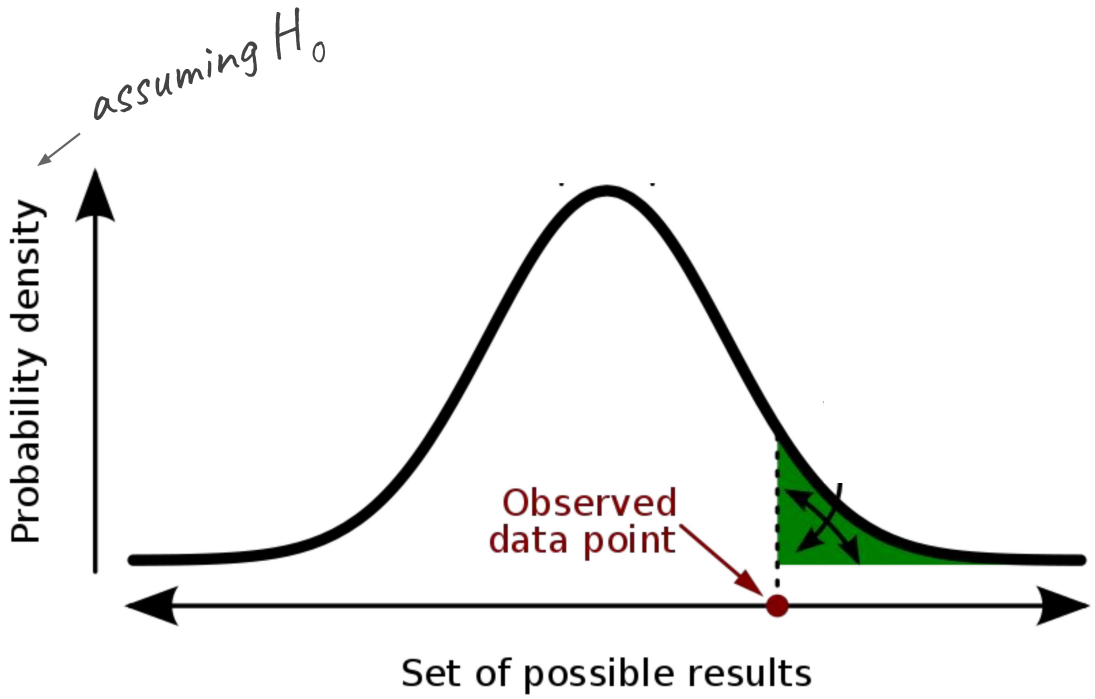
-> If low enough, then we "reject the null (H_0) in favor of H_1."

H_0: The blue case is not selling more than average.

Hypothesis Testing

P($D|H_0$): *Given null, what is the probability of the observed data or worse?*

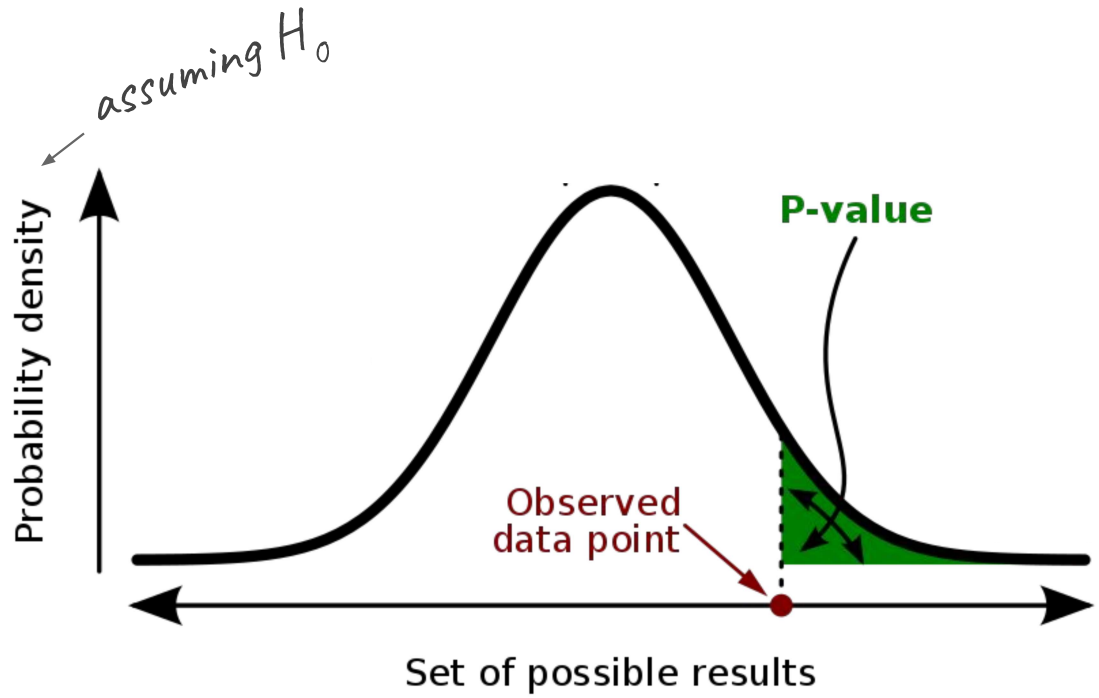
-> If low enough, then we "reject the null (H_0) in favor of H_1."



Hypothesis Testing

P($D|H_0$): *Given null, what is the probability of the observed data or worse?*

-> If low enough, then we "reject the null (H_0) in favor of H_1."

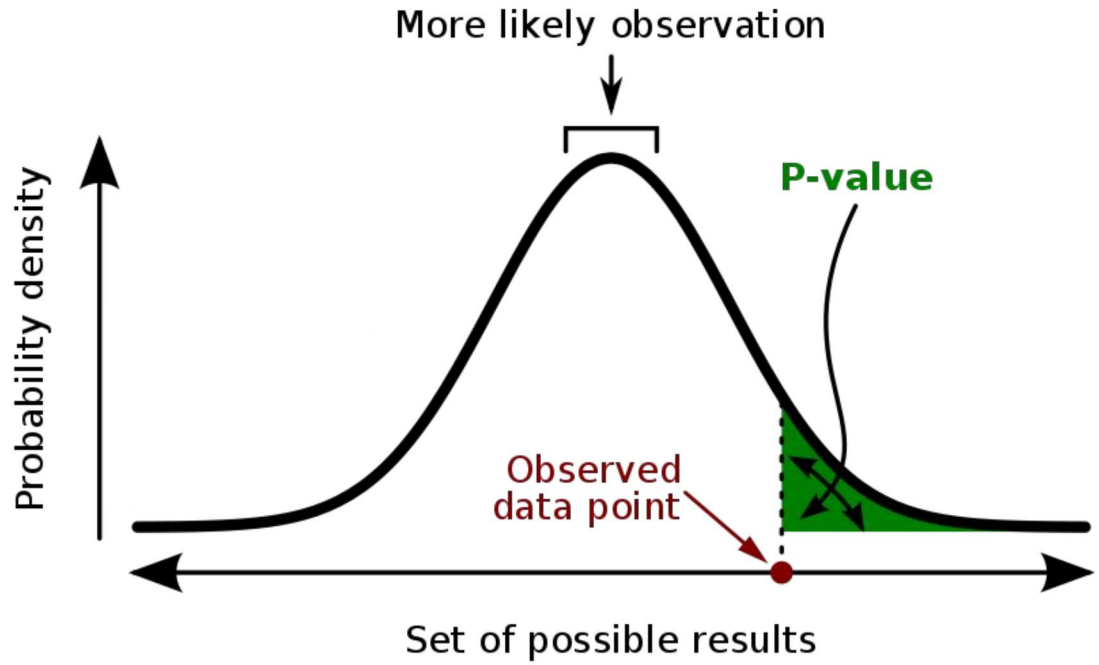


A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Hypothesis Testing

P($D|H_0$): *Given null, what is the probability of the observed data or worse?*

-> If low enough, then we "reject the null (H_0) in favor of H_1."

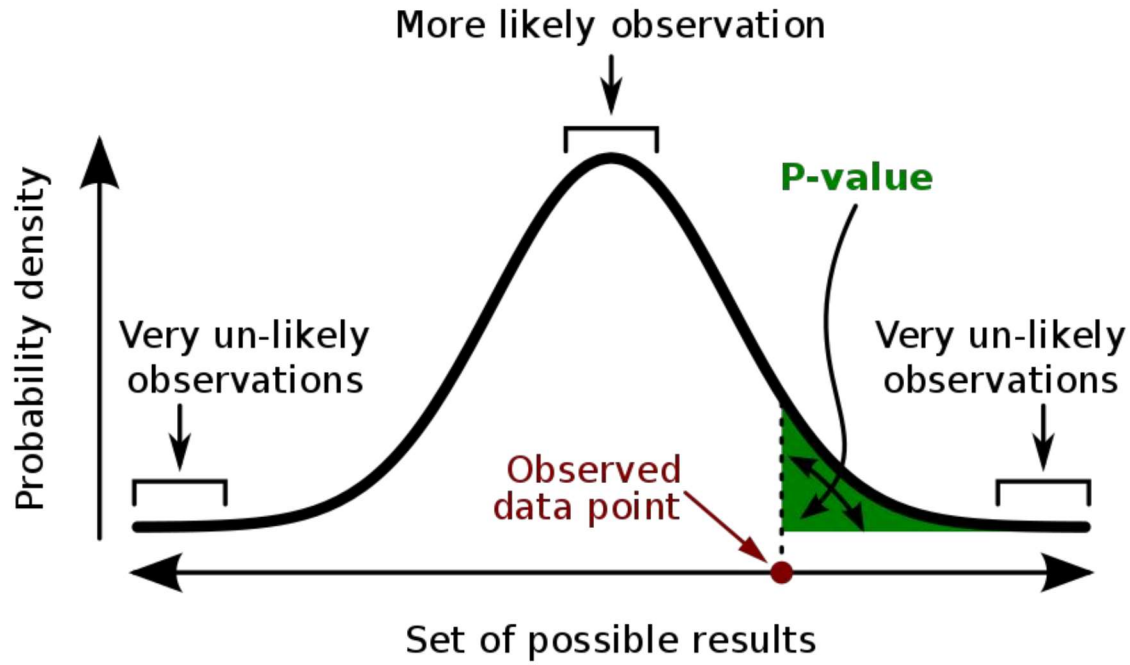


A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Hypothesis Testing

P($D|H_0$): *Given null, what is the probability of the observed data or worse?*

-> If low enough, then we "reject the null (H_0) in favor of H_1."



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

(thanks, Wikipedia)

The Hypothesis Test "Algorithm"

observations (i.e. data)

level of significance

Input: H_0, obs, α

Output: decision

Scenario: A third-party vendor makes a new Switch controller that they claim breaks less frequently than the default Switch controller. You suspect they are just trying to take advantage of a situation to sell more so **you hypothesize this new controller breaks moreu frequently.** The default Switch controller breaks 50% of the time.

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

observations (i.e. data)

level of significance

Input: H_0, obs, α

probability of what we observed or worse (i.e. more extreme)

$p(x \geq obs \mid H_0) < \alpha$

Output: decision

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

Input: H_0, obs, α

```
if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
```

Output: decision

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

Input: H_0, obs, α

Conditional is sometimes evaluated indirectly by first finding the "critical value" of some measurement such that: if measurement > critical_value then p(obs|H0) < α

```
if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
```

Output: decision

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

Input: H_0, obs, α

```
if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
```

Output: decision

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

Input: H_0, obs, α

Need to estimate

What is the distribution of values we would expect if the null was true?
-- the "null distribution"

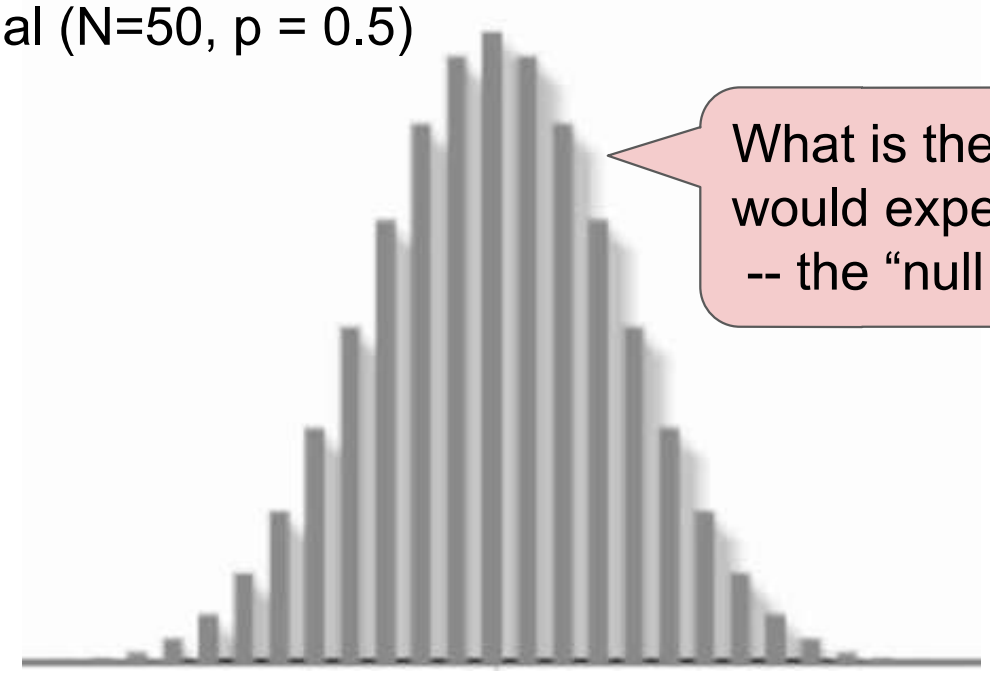
```
if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
```

Output: decision

H_0: The new Switch controller is not breaking more than the old one.

The Hypothesis Test "Algorithm"

Binomial (N=50, p = 0.5)
PMF



What is the distribution of values we would expect if the null was true?
-- the "null distribution"

25

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers

The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

null_dist = distribution of e

if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
Output: decision
```

What is the distribution of values we would expect if the null was true?
-- the "null distribution"

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers

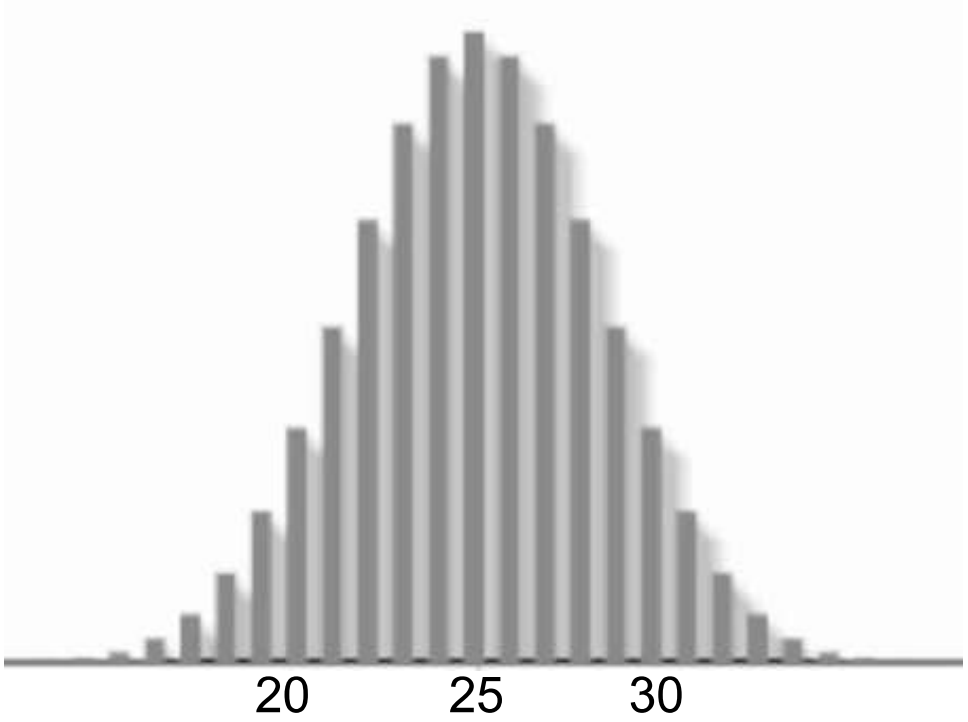
The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

null_dist = distribution of expected values under H0



if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
Output: decision
```

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers

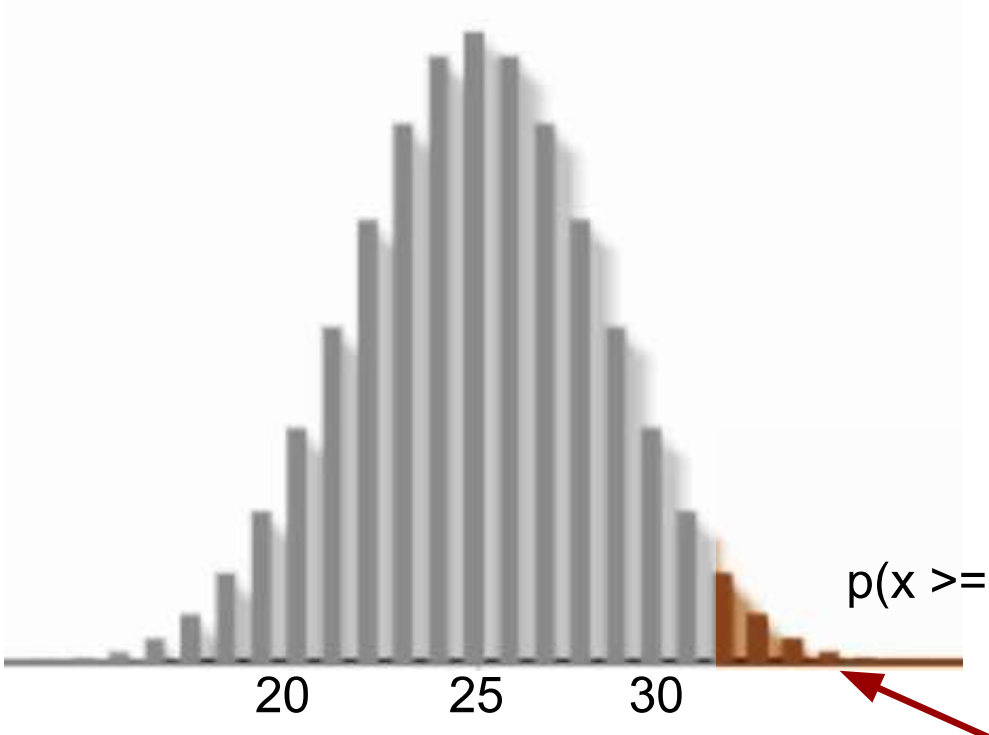
The Hypothesis Test "Algorithm"



H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"



p(x >= 32; Binomial(50, 0.5)) = 0.016

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

null_dist = distribution of expected values under H0

p(x>=obs | H0) =

if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
Output: decision
```

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

null_dist = distribution of expected values under H0

p(x>=obs | H0) = sum([pmf(null_dist, o)
                      for o in range(obs,)])
if p(x>=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
Output: decision
```

H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

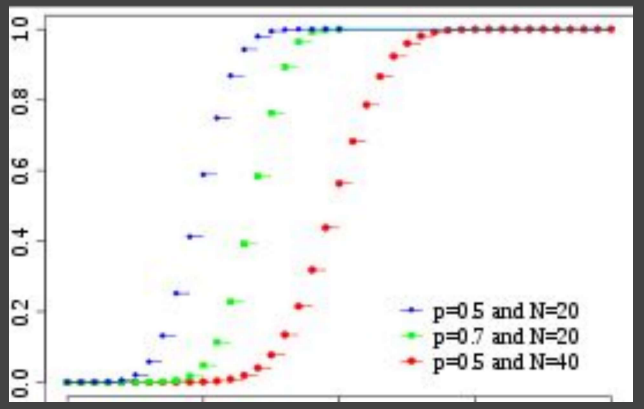
The Hypothesis Test "Algorithm"

Input: H_0, obs, α

null_dist = distribution of expected values under H_0

p(x>=obs | H_0) = 1 - cdf(null_dist, obs)

if p(x>=obs | H_0) < α:
    decision = "Reject H_0!"
else:
    decision = "Failed to reject H_0."
Output: decision



H_0: The new Switch controller is not breaking more than the old one.

50 tests; Thus, average would be 25 broken controllers ***Observed 32 broken***

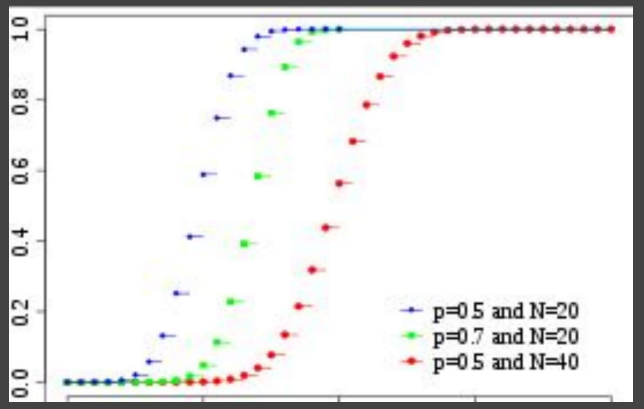
The Hypothesis Test "Algorithm"

Input: H_0, obs, α

null_dist = distribution of expected values under H_0

p(x>=obs | H_0) = 1 - cdf(null_dist, obs)

if p(x>=obs | H_0) < α:
    decision = "Reject H_0!"
else:
    decision = "Failed to reject H_0."
Output: decision



H_0: The new Switch controller is not breaking ~~more~~**less** than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

null_dist = distribution of expected values under H0

p(x<=obs | H0) = cdf(null_dist, obs)

if p(x<=obs | H0) < α:
    decision = "Reject H0!"
else:
    decision = "Failed to reject H0."
Output: decision
```

H_0: The new Switch controller is not breaking ~~more~~**less** than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"

```
Input: H0, obs, α

 null_dist = distribution of expected obs under H0
 p(x<=obs | H0) = cdf(null_dist, obs)
 if p(x<=obs | H0) < α:
    decision = "Reject H0!"
 else:
    decision = "Failed to reject H0."

Output: decision
```

H_0: The new Switch controller is not breaking ~~more~~**less** than the old one.

50 tests; Thus, average would be 25 broken controllers **Observed 32 broken**

The Hypothesis Test "Algorithm"

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected obs under H0
p(x<=obs | H0) = cdf(null_dist, obs)
if p(x<=obs | H0) < α:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

Output: decision

We were comparing counts or frequency.

Can we generalize to any sort of "score", any test statistic?

The Hypothesis Test “Algorithm”

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected ~~obs~~test_stat under H0
p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p(x<=obs_ts | H0) < α:
   decision = “Reject H0!”
else:
   decision = “Failed to reject H0.”
```

Output: decision

The *Hypothesis Test Algorithm*

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p < α:
   decision = “Reject H0!
else:
   decision = “Failed to reject H0 ”
```

Output: decision, **p**

the "p-value":
also often returned
along with decision

The *Hypothesis Test Algorithm*

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p < α:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

changes depending on tail needed for test.

Output: decision, **p**

(the complete version, without rejection regions)

The *Hypothesis Test Algorithm* ***with rejection regions***

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = cdf(null_dist, obs_ts)
reject_region = ppf(null_dist, α)
if p < α obs_ts in reject_region:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

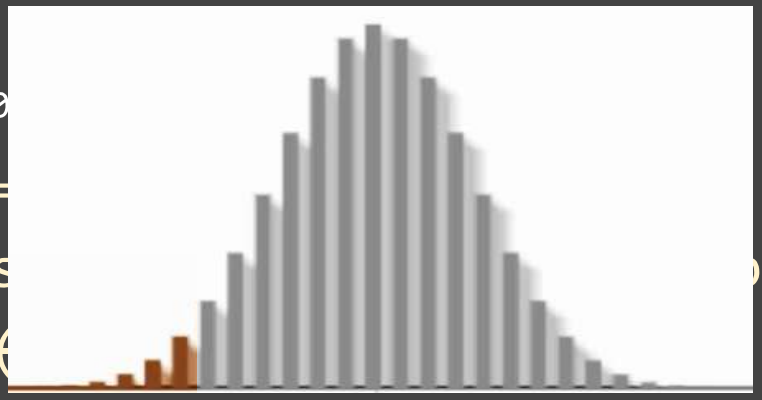
Output: decision, p

ppf: inverse of cdf: given probability (α), return x

rejection region: regions of pdf where null should be rejected.

The *Hypothesis Test Algorithm* ***with rejection regions***

Input: H_0



```
obs_ts =
null_dis                                  ected test_stat under H0
p = cdf(
reject_region = ppf(null_dist, α)
if p < α obs_ts in reject_region:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

Output: decision, p

ppf: inverse of cdf: given probability (α), return x

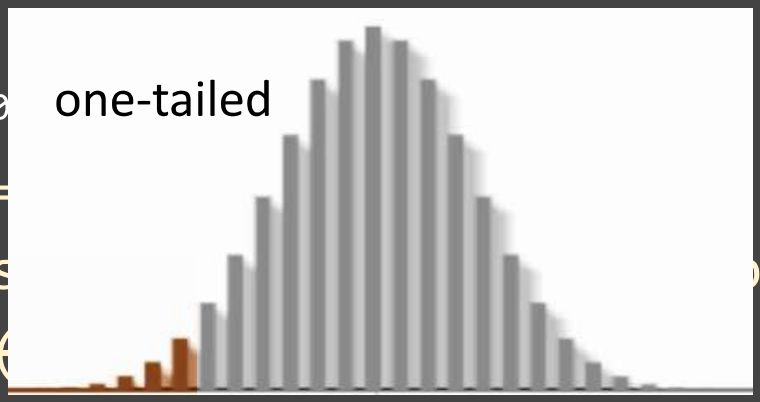
rejection region: regions of pdf where null should be rejected.

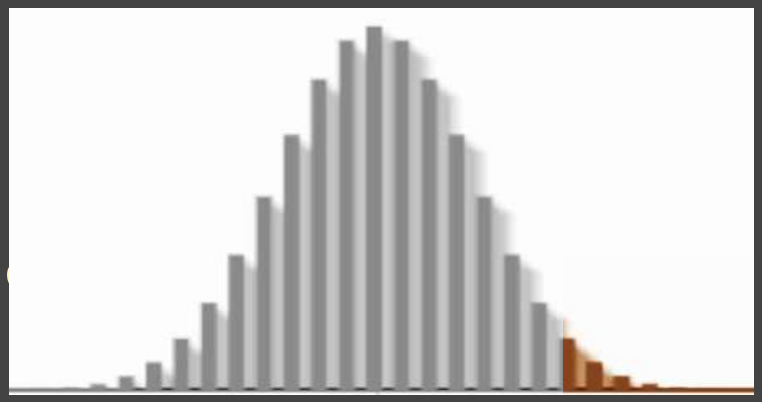
The *Hypothesis Test Algorithm* ***with rejection regions***

Input: H_0

```
obs_ts =
null_dis                                    be
p = cdf(
reject_region = ppf(null_dist, α)
if p < α obs_ts in reject_region:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

Output: decision, ~~p~~





ppf: inverse of cdf: given probability (α), return x

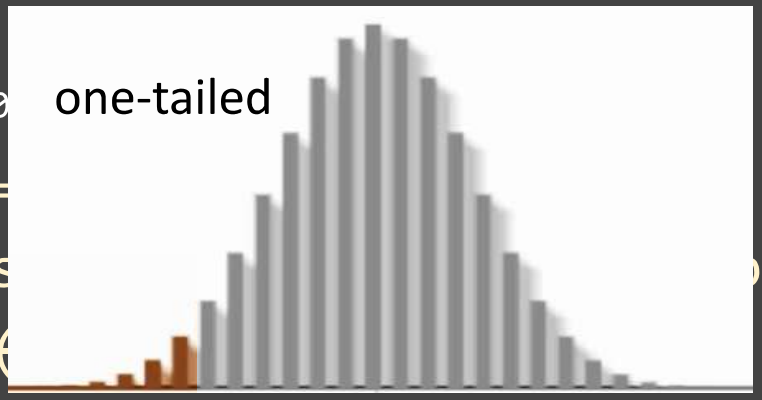
rejection region: regions of pdf where null should be rejected.

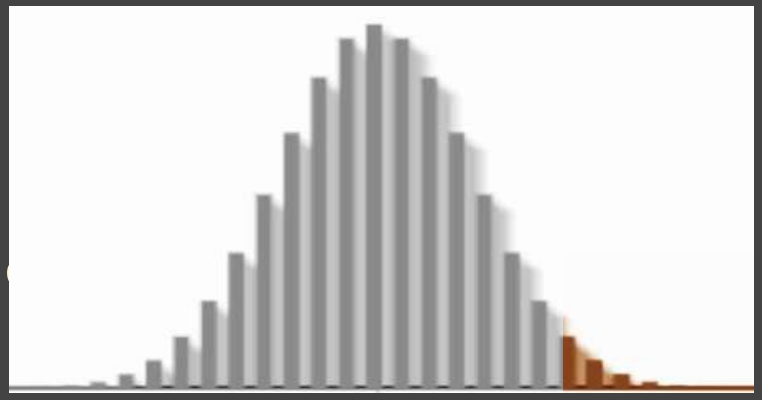
The *Hypothesis Test Algorithm* ***with rejection regions***

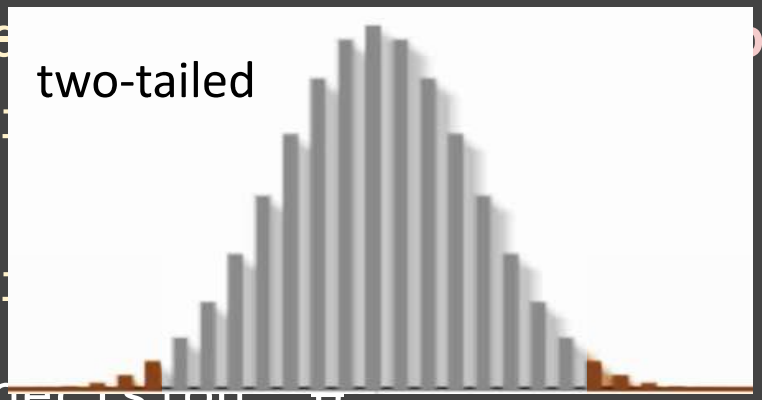
Input: H_0

```
obs_ts =
null_dis                                    be
p = cdf(
reject_region = ppf(null_dist, α)
if p < e                                    on:
   decis
else:
   decis                                    H0."
```

Output: decision, p







ppf: inverse of cdf: given probability (α), return x

rejection region: regions of pdf where null should be rejected.

total area of rejection region must be α

The *Hypothesis Test Algorithm* ***with rejection regions***

```
Input: H0, obs, α
 //presetup criteria to reject
 null_dist = distribution of expected test_stat under H0
 reject_region = ppf(null_dist, α)

 //test if meets criteria to reject
 obs_ts = test_stat(obs)
 if obs_ts in reject_region:
    decision = “Reject H0!”
 else:
    decision = “Failed to reject H0.”

Output: decision
```

(the complete version, with rejection regions)

Hypothesis Testing

Why?

Hypothesis Testing

Why?

A general framework for answering (yes/no) questions!

Hypothesis Testing

Why?

A general framework for answering (yes/no) questions!

- *Are height and baldness related?*
- *Is my deep predictive model better than the state of the art?*

Hypothesis Testing

Why?

A general framework for answering (yes/no) questions!

- *Are height and baldness related?*
- *Is my deep predictive model better than the state of the art?*
- *Is the heat index of a community related to poverty?*
- *Is the heat index of a community related to poverty* ***controlling for education rates****?*
- *Does my website receive a higher average number of monthly visitors?*

Hypothesis Testin

Failing to "reject the null" does not mean the null is true.

Why?

A general framework for answering (yes/**maybe**) questions!

- *Are height and baldness related?*
- *Is my deep predictive model better than the state of the art?*
- *Is the heat index of a community related to poverty?*
- *Is the heat index of a community related to poverty* ***controlling for education rates****?*
- *Does my website receive a higher average number of monthly visitors?*

Hypothesis Testir

Failing to "reject the null" does not mean the null is true. However, if the sample is large enough, it may be enough to say that the effect size (correlation, difference value, etc...) is not very meaningful.

Why?

A general framework for answering (yes/**maybe**) questions!

- *Are height and baldness related?*
- *Is my deep predictive model better than the state of the art?*
- *Is the heat index of a community related to poverty?*
- *Is the heat index of a community related to poverty* ***controlling for education rates****?*
- *Does my website receive a higher average number of monthly visitors?*

Hypothesis Testing

Why?

A general framework for answering (yes/**maybe**) questions!

- *Are height and baldness related?*
- *Is my deep predictive model better*
- *Is the heat index of a community*
- *Is the heat index of a community related to poverty* ***controlling for education rates?***
- *Does my website receive a higher average number of monthly visitors?*

What if we do not know the distribution?
Are their non-parametric techniques?
(like kernel density estimation)

Resampling Techniques

The bootstrap

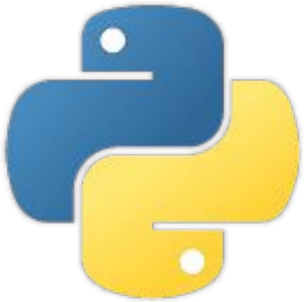
What if we do not know the distribution?
Are their non-parametric techniques?
(like kernel density estimation)

Resampling Techniques

The bootstrap

- What if we don't know the distribution?
- *Resample* many potential samples of what would happen for the null to establish an empirical distribution of the null.

Resample with replacement: for each *i* in *n* observations, put all observations in a hat and draw one and put it back in (all observations are equally likely).



Error Types

Type I, Type II Errors

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Error Types

significance level ("p-value") = P(type I error) = **P(Reject H_0 | H_0= True)**
(probability we are incorrect)

	H_0	H_A
Reject H_0	**P(Reject H_0 | H_0)**	

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Error Types

significance level ("p-value") = P(type I error) = **P(Reject H_0 | H_0= True)**
(probability we are incorrect)

power = 1 - P(type II error) = **P(Reject H_0 | H_1= True)**
(probability we are correct)

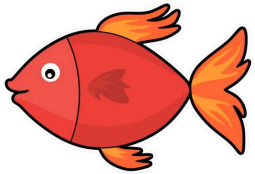
	H_0	H_A
Reject H_0	P(Reject H_0 | H_0)	P(Reject H_0 | H_A)

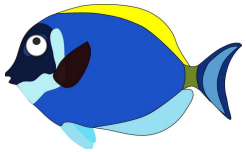
		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Bonferroni's Cats

General Question: Which fish do cats like?



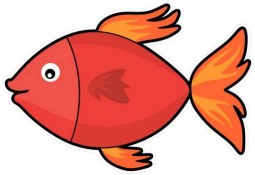


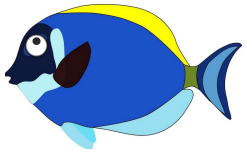
⋮

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats



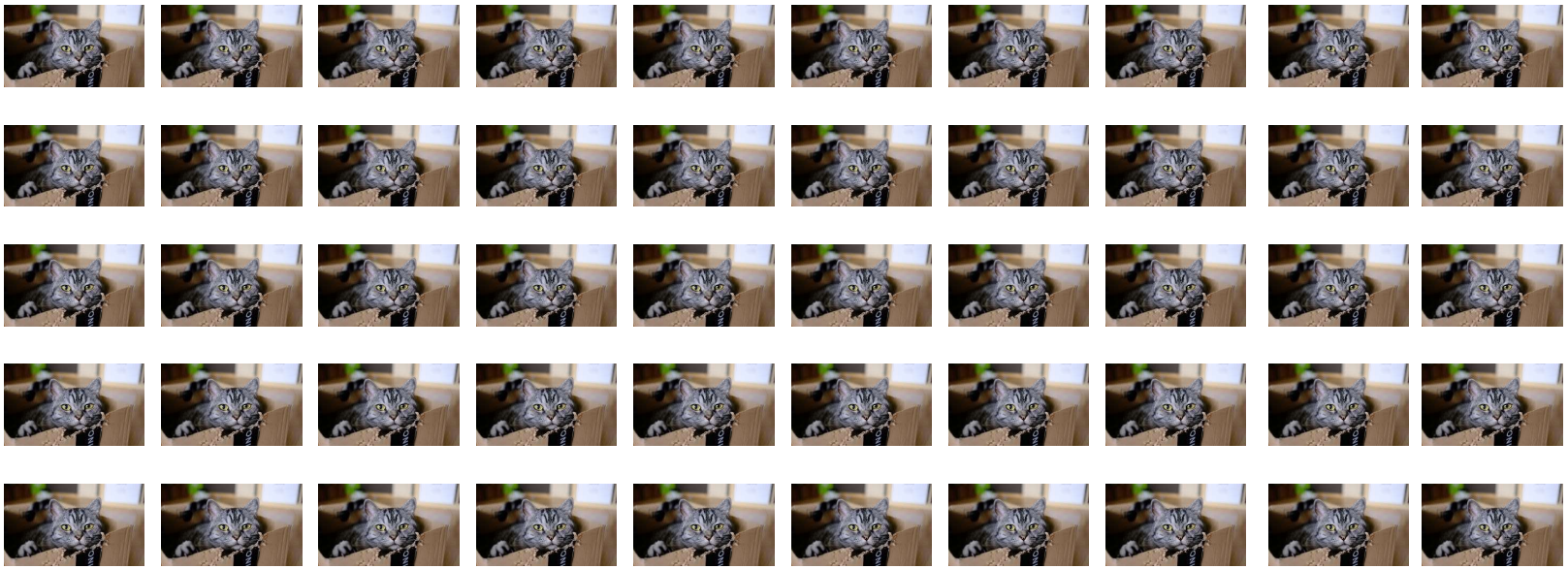


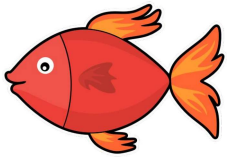
⋮

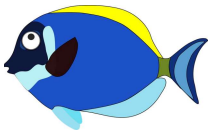
Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats





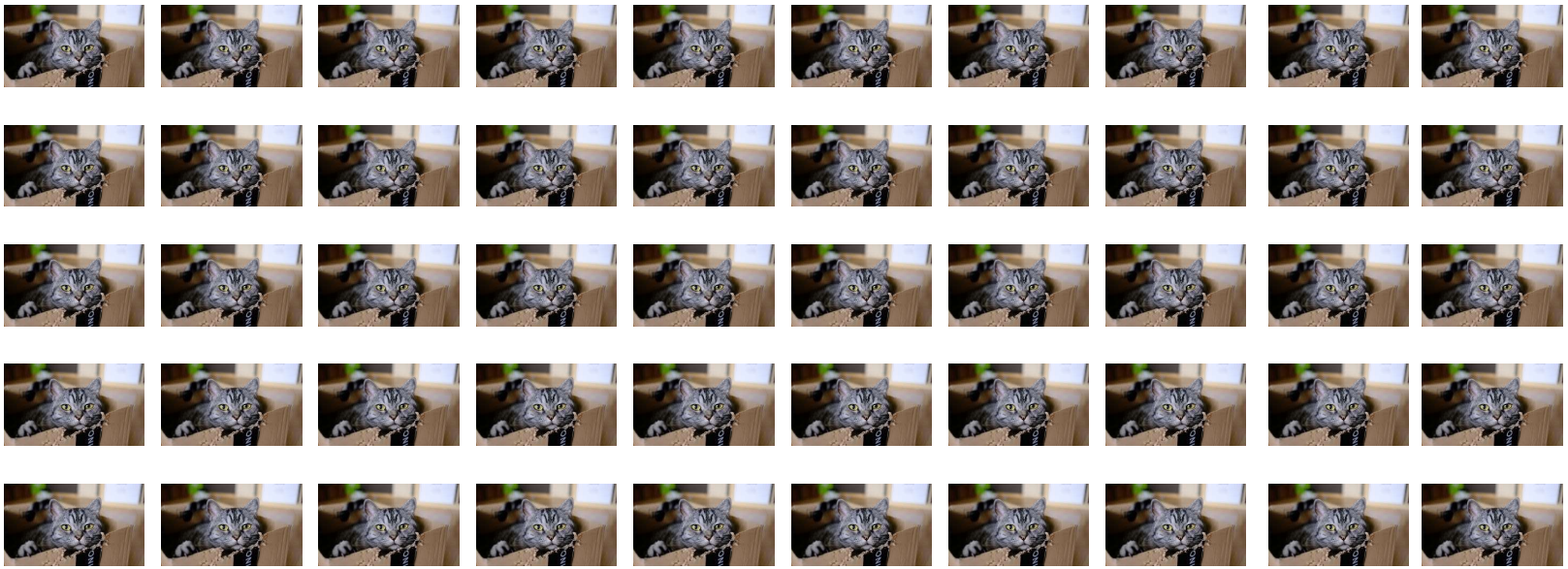


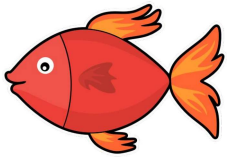
⋮

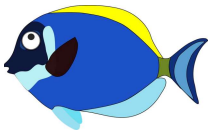
Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats





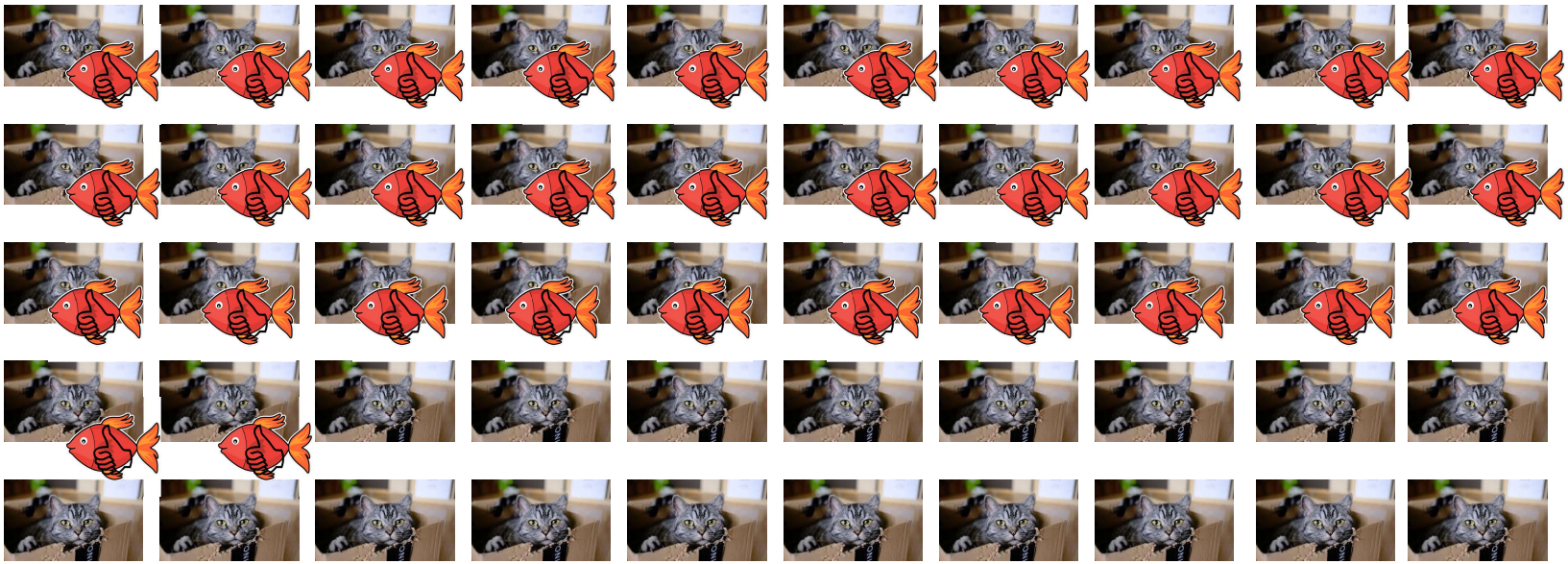


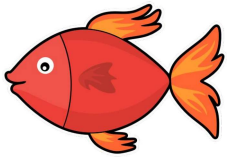
H_1: Most cats like redfish. H_0: Most cats don't like redfish.

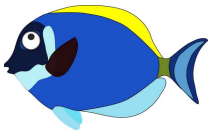
Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*





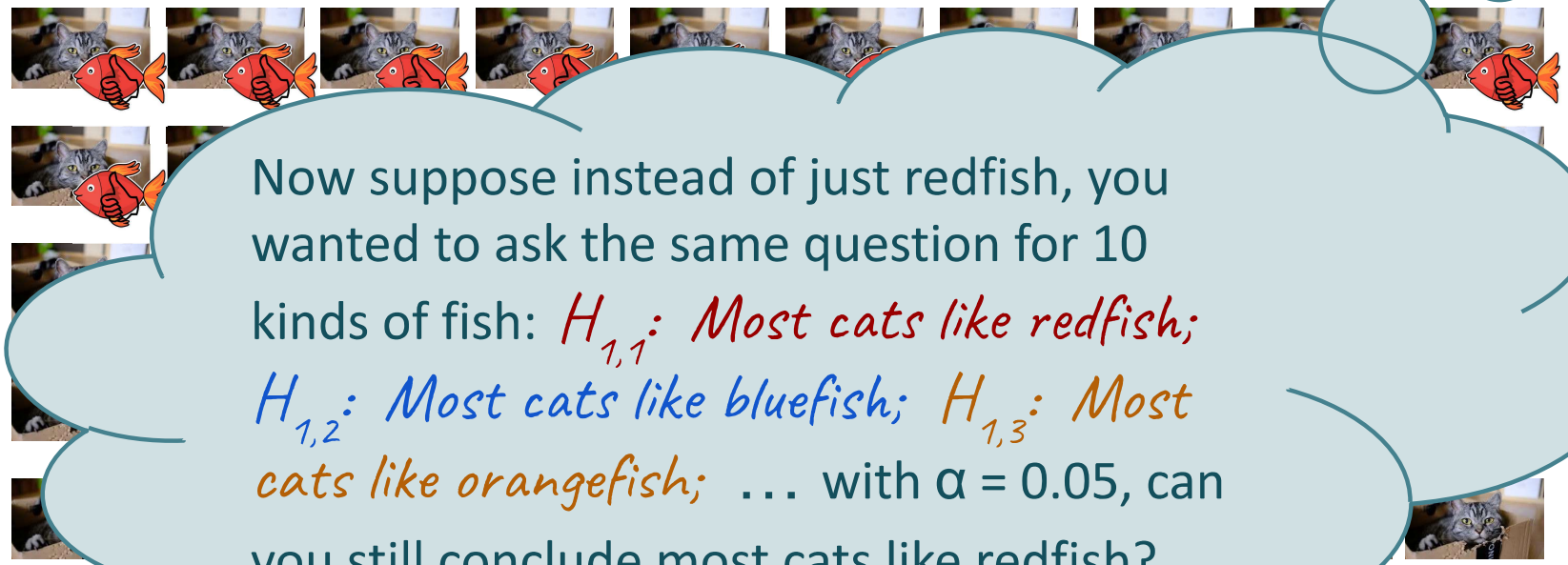


H_1: *Most cats like redfish.* H_0: *Most cats don't like redfish.*

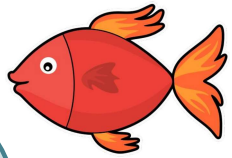
Bonferroni's Cats

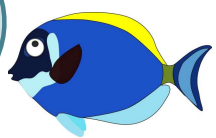
General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*



Now suppose instead of just redfish, you wanted to ask the same question for 10 kinds of fish: $H_{1,1}$: *Most cats like redfish;* $H_{1,2}$: *Most cats like bluefish;* $H_{1,3}$: *Most cats like orangefish;* … with α = 0.05, can you still conclude most cats like redfish?



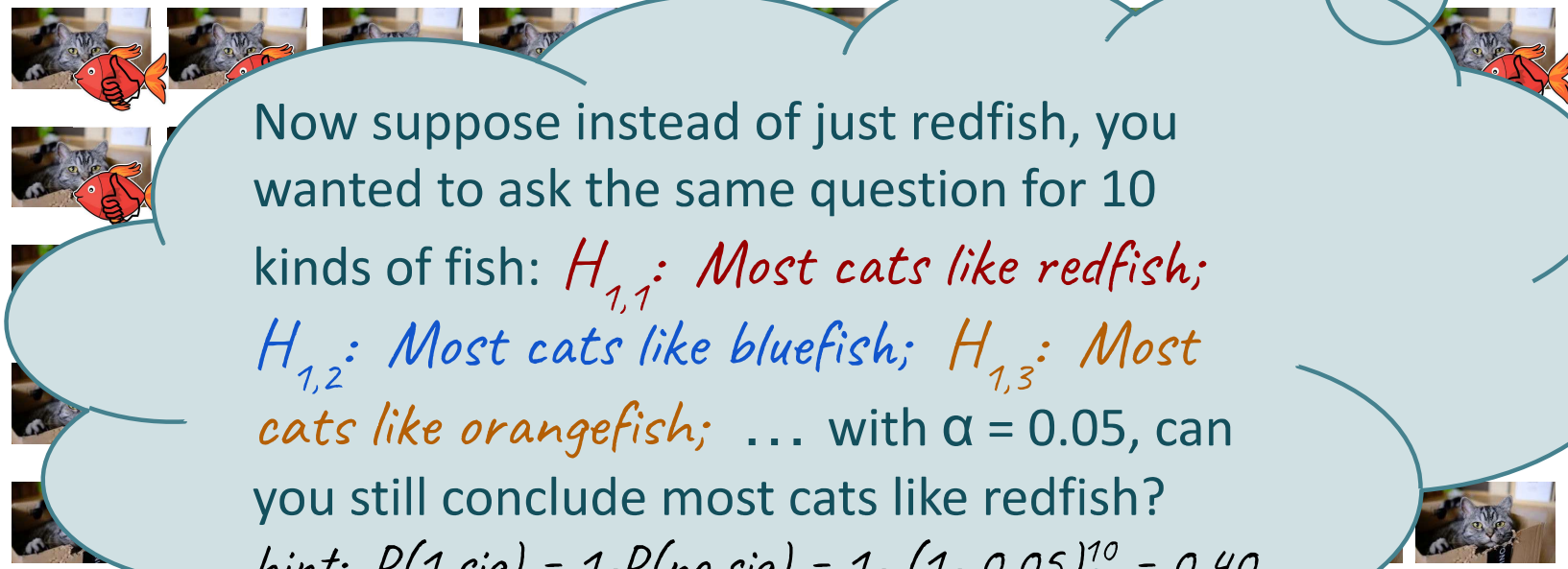


H_1: *Most cat… don't like redfish.*

Bonferroni's Cats

General Question: Which fish do cats like?

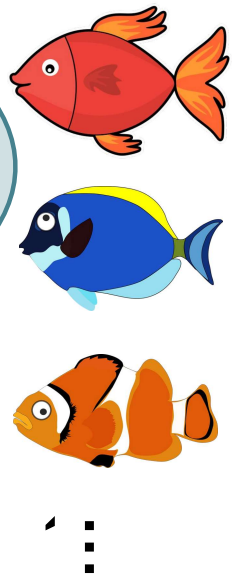
N = 50 cats; *32 like redfish; p = 0.016*



Now suppose instead of just redfish, you wanted to ask the same question for 10 kinds of fish: $H_{1,1}$: *Most cats like redfish;* $H_{1,2}$: *Most cats like bluefish;* $H_{1,3}$: *Most cats like orangefish;* ... with α = 0.05, can you still conclude most cats like redfish?

hint: $P(1\ sig) = 1 - P(no\ sig) = 1 - (1 - 0.05)^{10} = 0.40$

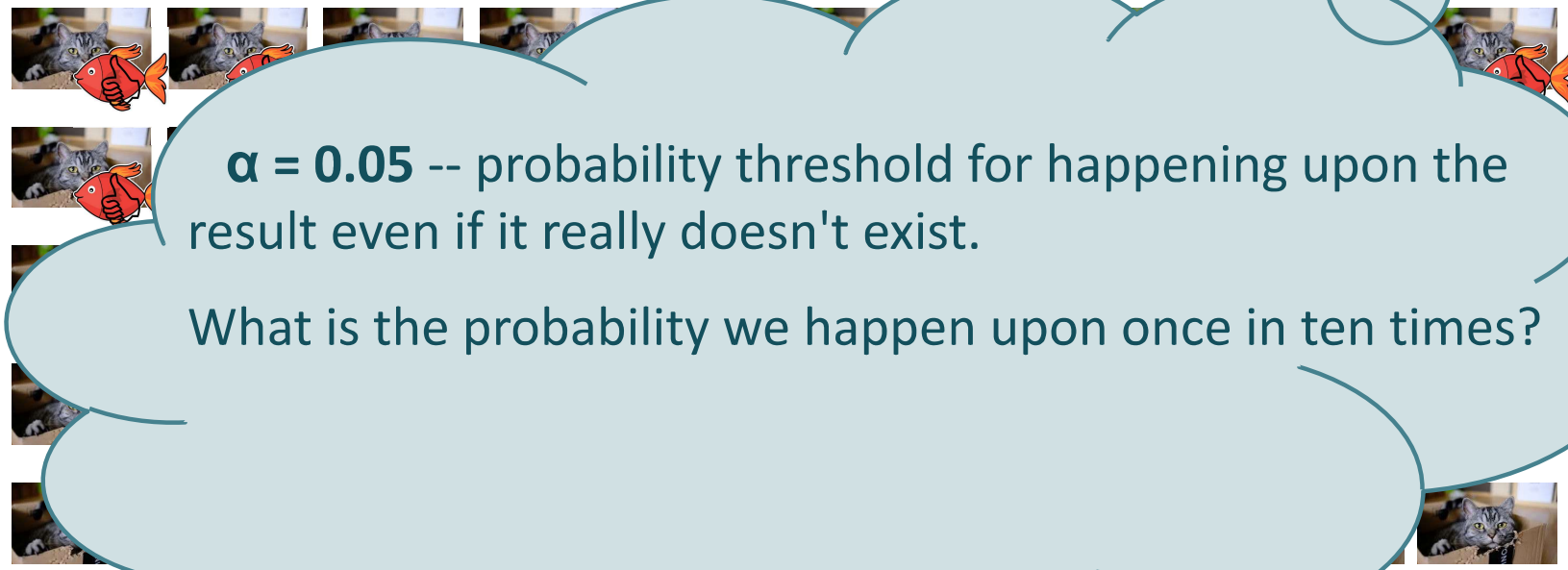
H_1: *Most cats* ... *don't like redfish.*



Bonferroni's Cats

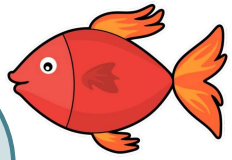
General Question: Which fish do cats like?

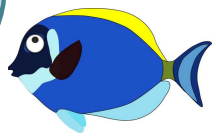
N = 50 cats; *32 like redfish; p = 0.016*



α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

What is the probability we happen upon once in ten times?



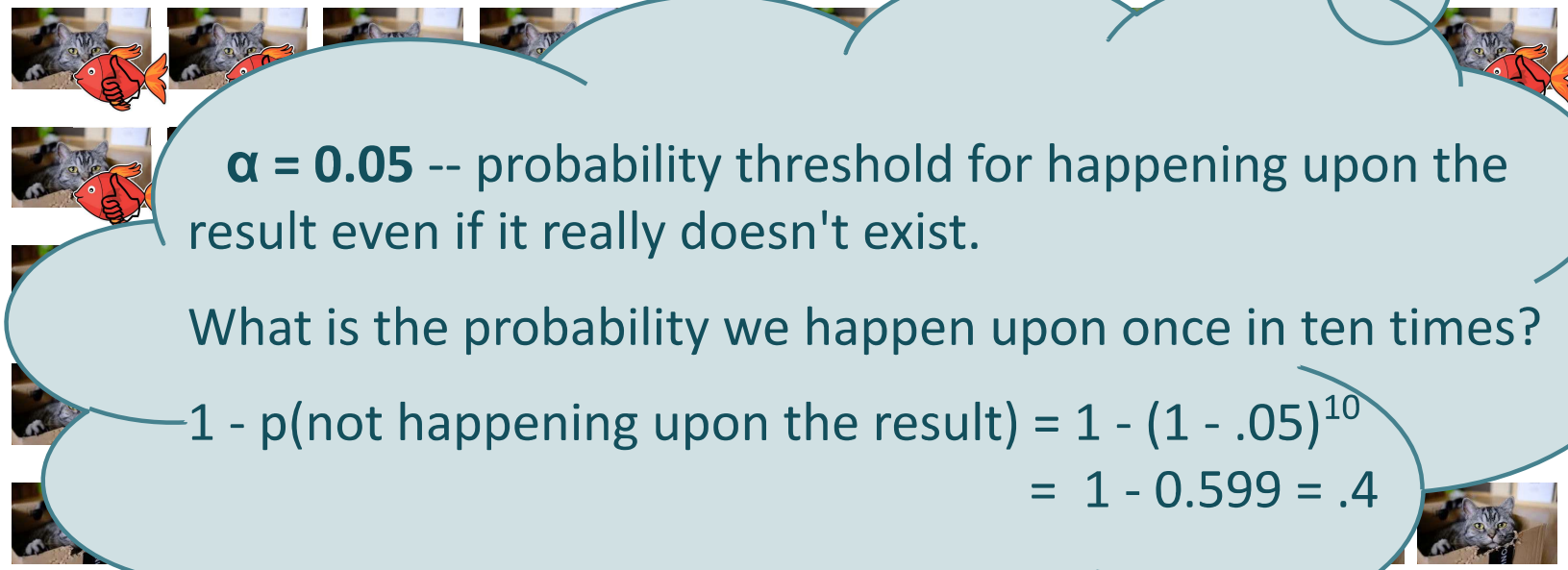


H_1: Most cats don't like redfish.

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*

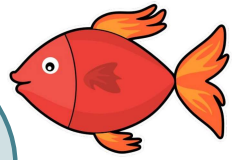


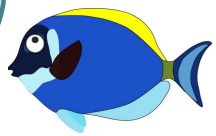
α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

What is the probability we happen upon once in ten times?

1 - p(not happening upon the result) = $1 - (1 - .05)^{10}$

$= 1 - 0.599 = .4$



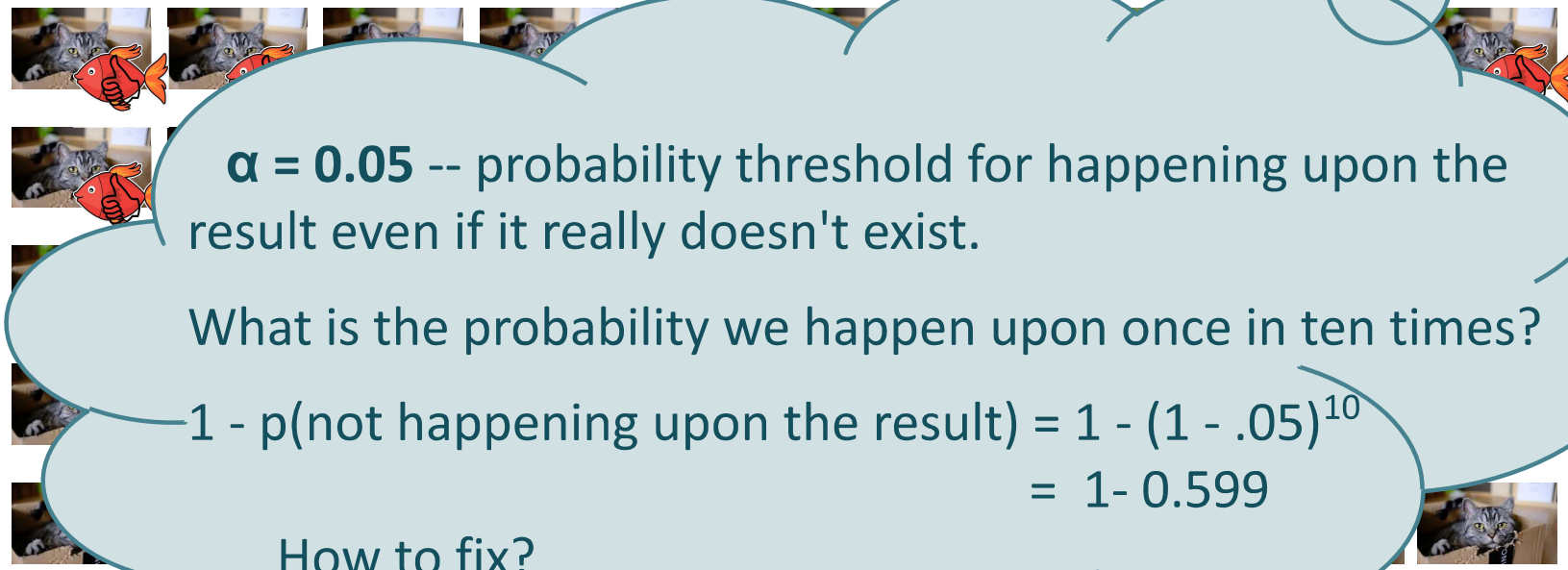


H_1: Most cats [illegible] don't like redfish.

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*



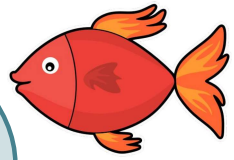
α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

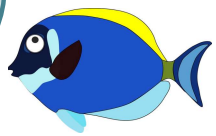
What is the probability we happen upon once in ten times?

1 - p(not happening upon the result) = $1 - (1 - .05)^{10}$

$= 1 - 0.599$

How to fix?



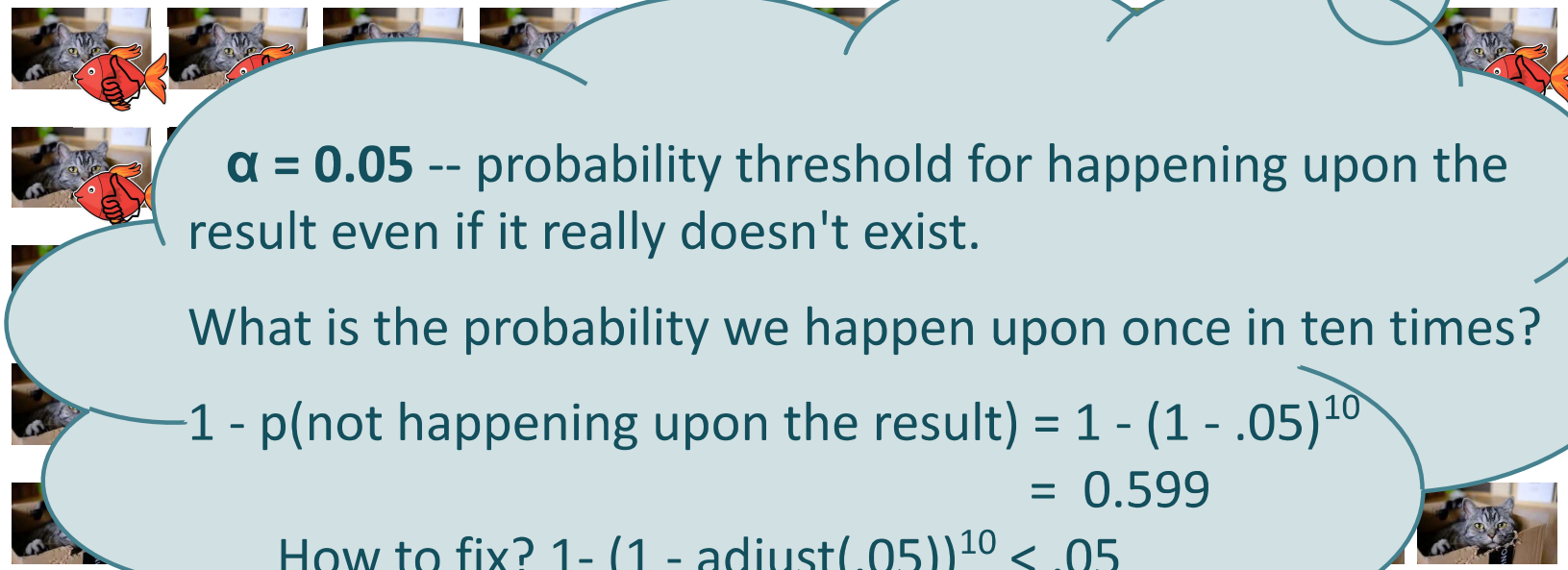


H_1: Most cats [illegible] don't like redfish.

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*



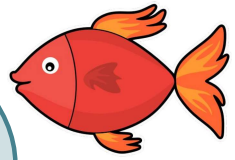
α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

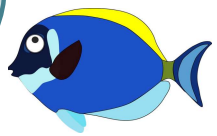
What is the probability we happen upon once in ten times?

1 - p(not happening upon the result) = $1 - (1 - .05)^{10}$

= 0.599

How to fix? $1 - (1 - \text{adjust}(.05))^{10} < .05$



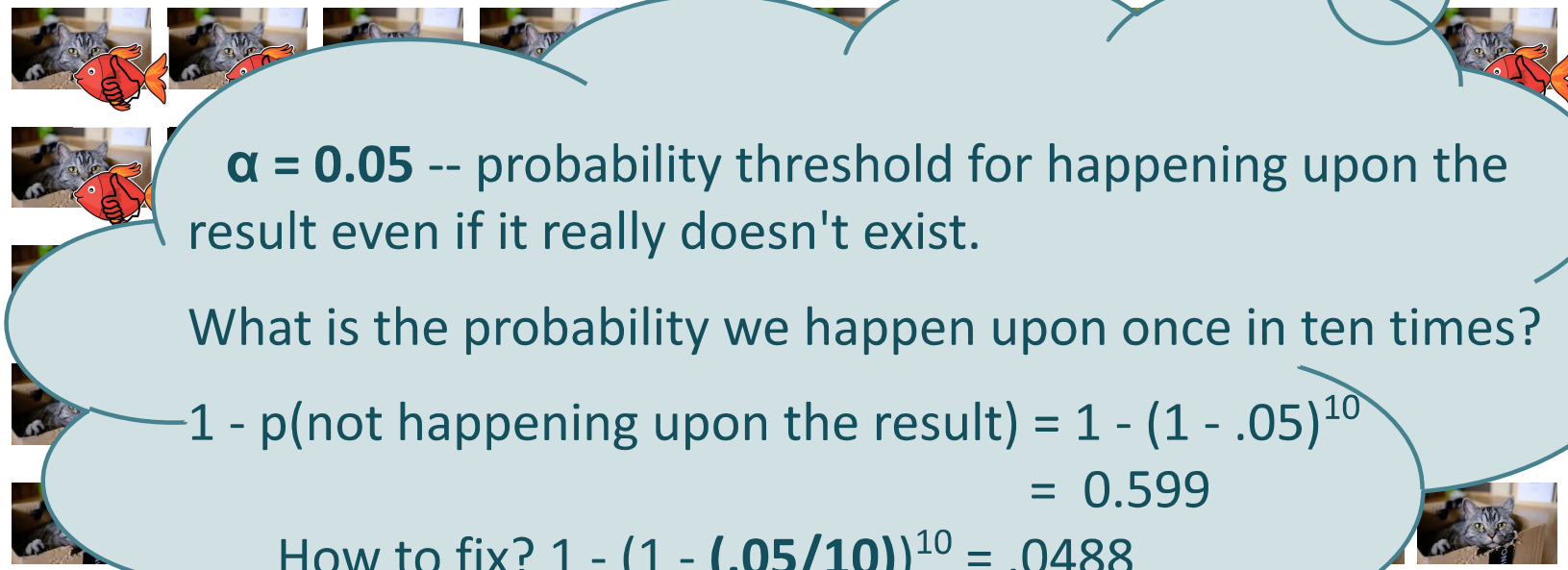


H_1: Most cats [illegible] don't like redfish.

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*



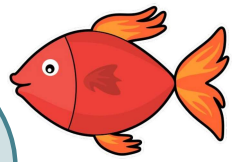
α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

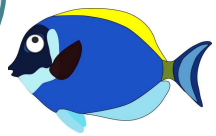
What is the probability we happen upon once in ten times?

1 - p(not happening upon the result) = $1 - (1 - .05)^{10}$

= 0.599

How to fix? $1 - (1 - \mathbf{(.05/10)})^{10} = .0488$



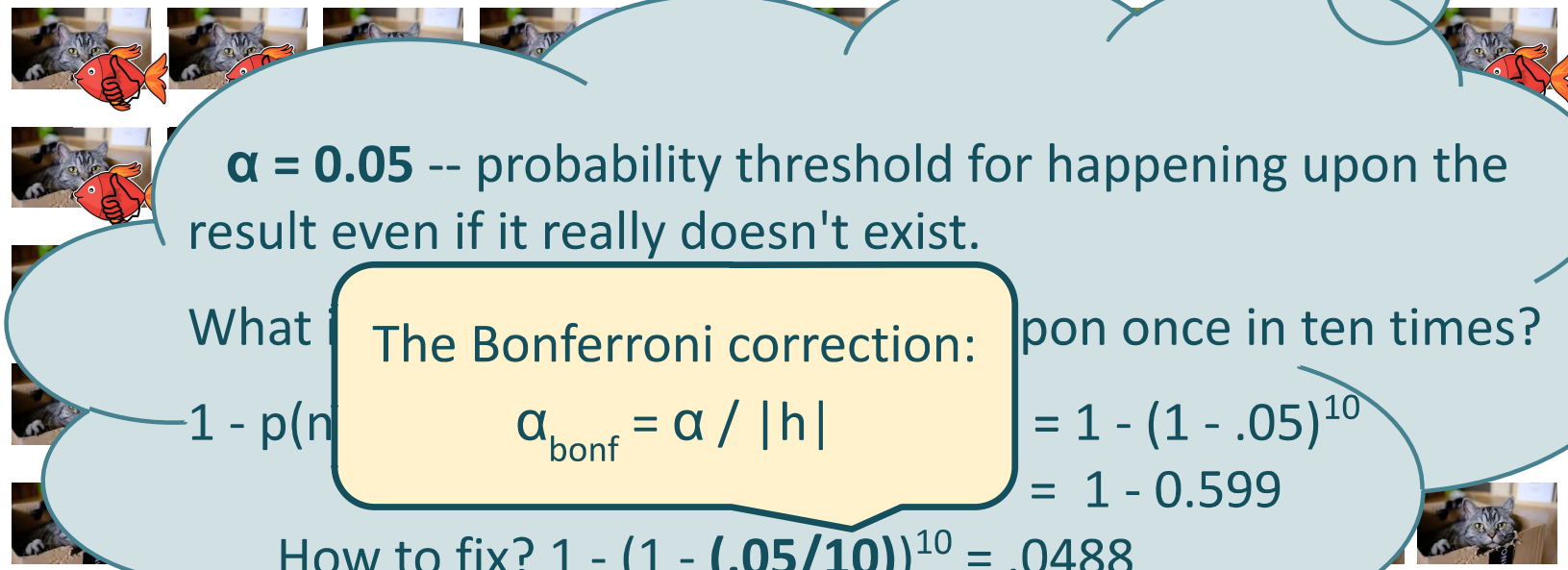


H_1: Most cats [illegible] don't like redfish.

Bonferroni's Cats

General Question: Which fish do cats like?

N = 50 cats; *32 like redfish; p = 0.016*



α = 0.05 -- probability threshold for happening upon the result even if it really doesn't exist.

What [illegible] pon once in ten times?

1 - p(n[illegible] = 1 - (1 - .05)^{10}

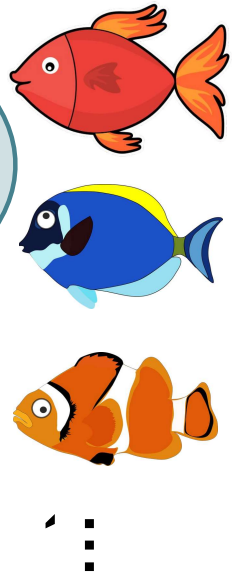
= 1 - 0.599

The Bonferroni correction:

$\alpha_{bonf} = \alpha / |h|$

How to fix? 1 - (1 - **(.05/10)**)^{10} = .0488

H_1: Most cats [illegible] don't like redfish.



Multi-test Correction

significance level ("p-value") = P(type I error) = **P(Reject H_0 | H_0)**
(probability we are incorrect)

power = 1 - P(type II error) = **P(Reject H_0 | H_1)**
(probability we are correct)

	H_0	H_A
Reject H_0	**P(Reject H_0 | H_0)**	**P(Reject H_0 | H_A)**

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Multi-test Correction

FWER: Family-wise error rate (Bonferroni Corrects)

The probability of making >=1 type 1 error.

$$FWER = Pr(\text{type1s}>0) = 1 - Pr(\text{type1s}=0) = 1 - (1 - a)^m$$

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Multi-test Correction

FWER: Family-wise error rate (Bonferroni Corrects)

The probability of making >=1 type 1 error.

$$FWER = Pr(\text{type1s}>0) = 1 - Pr(\text{type1s}=0) = 1 - (1 - a)^m$$

$1 - (1 - \mathbf{(.05/10)})^{10} = .488$

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Multi-test Correction

FWER: Family-wise error rate (Bonferroni corrects)

The probability of making >=1 type 1 error.

$$FWER = Pr(\text{type1s}>0) = 1 - Pr(\text{type1s}=0) = 1 - (1 - a)^m$$

FDR: False discovery rate (Benjamini-Hochberg corrects)

type1s / (type1s + correctRejects)

		True state of nature	
		H_0	H_A
Our decision	Reject H_0	Type I error	correct decision
	'Accept' H_0	correct decision	Type II error

(Orloff & Bloom, 2014)

Multi-test Correction

FWER: Family-wise error rate (Bonferroni corrects)

The probability of making >=1 type 1 error.

$$FWER = Pr(\text{type1s}>0) = 1 - Pr(\text{type1s}=0) = 1 - (1 - a)^m$$

FDR: False discovery rate (Benjamini-Hochberg corrects)

type1s / (type1s + correctRejects)

Proportion of false positives among *all* significant results.

(Orloff & Bloom, 2014)

We want to have good power:
What are the "knobs" we can turn to achieve it?

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p < α:
   decision = "Reject H0!"
else:
   decision = "Failed to reject H0."
```

Output: decision, **p-value**

(the complete version, without rejection regions)

What are the knobs we can turn?

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p < α:
   decision = "Reject H0"
else:
   decision = "Failed to reject H0."
```

Output: decision, **p-value**

p-value is a function of H0, obs

(the complete version, without rejection regions)

What are the knobs we can turn?

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = cdf(null_dist, obs_ts)
if p < α:
   decision = "Reject H0."
else:
   decision = "Failed to reject H0."
```

Output: decision, p-value

p-value is a function of H0 and obs;

(the complete version, without rejection regions)

What are the knobs we can turn?

Input: H_0, obs, α

```
obs_ts = test_stat(obs)
null_dist = distribution of expected test_stat under H0
p = p(x<=obs_ts | H0) = f(null_dist, obs_ts)
if p < α:
   decision = "Reject H0"
else:
   decision = "Failed to reject H0."
```

Output: decision, **p-value**

p-value is a function of H0 and obs_ts;

(the complete version, without rejection regions)

We want to have good power:
What are the "knobs" we can turn to achieve it?

p-value = func(obs_ts, H_0)

p-value is a function of H0 and obs_ts;

We want to have good power:
What are the "knobs" we can turn to achieve it?

p-value = func(obs_ts, H_0)

Scenario: A third-party vendor makes a new Switch controller that they claim breaks less frequently than the default Switch controller. You suspect they are just trying to take advantage of a situation to sell more so **you hypothesize this new controller breaks moreu frequently.** The default Switch controller breaks 50% of the time.

We want to have good power:
What are the "knobs" we can turn to achieve it?

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

Scenario: A third-party vendor makes a new Switch controller that they claim breaks less frequently than the default Switch controller. You suspect they are just trying to take advantage of a situation to sell more so **you hypothesize this new controller breaks moreu frequently.** The default Switch controller breaks 50% of the time.

We want to have good power:
What are the "knobs" we can turn to achieve it?

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased percent broken*

Scenario: A third-party vendor makes a new Switch controller that they claim breaks less frequently than the default Switch controller. You suspect they are just trying to take advantage of a situation to sell more so **you hypothesize this new controller breaks moreu frequently.** The default Switch controller breaks 50% of the time.

We want to have good power:
What are the "knobs" we can turn to achieve it?

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased percent broken*

= binomialp_observed - binomialp_null

Scenario: *A third-party vendor makes a new Switch controller that they claim breaks less frequently than the default Switch controller. You suspect they are just trying to take advantage of a situation to sell more so* ***you hypothesize this new controller breaks moreu frequently.*** *The default Switch controller breaks 50% of the time.*

Power Analysis for a Binomial

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased proportion broken*

= binomialp_observed - binomialp_null

H_0 is given. We want to know what N do we need for our study.

Power Analysis for a Binomial

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased proportion broken*

= binomialp_observed - binomialp_null

H_0 is given. We want to know what N do we need for our study.

What is the obs_ts (incr proportion) that we think matters? *let's say 0.10*

What is the p-value we want (prob of type-I error)? *let's say 0.05*

Power Analysis for a Binomial

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased proportion broken*

= binomialp_observed - binomialp_null

H_0 is given. We want to know what N do we need for our study.

What is the obs_ts (incr proportion) that we think matters? *let's say 0.10*

What is the p-value we want (prob of type-I error)? *let's say 0.05*

What is the power we want (prob of not type-II error)? *let's say 0.80*

Power Analysis for a Binomial

p-value = func(obs_ts, H_0)

H_0 = Binomial(binomialp=0.50, N=??)

obs_ts = proportion_broke - 0.5 *#increased proportion broken*

= binomialp_observed - binomialp_null

H_0 is given. We want to know what N do we need for our study.

What is the obs_ts (incr proportion) that we think matters? *let's say 0.10*

What is the p-value we want (prob of type-I error)? *let's say 0.05*

What is the power we want (prob of not type-II error)? *let's say 0.80*

N > ??

Hypothesis Testing with 2 Normals: T-test

Degrees of Freedom: the number of values that are free to vary

The number of observations available to measure a parameter in a distribution. In other words, what is the minimum *i*, such that given *i* observations one could determine the parameter?

df = N - 1, because we have used up 1 parameter to compute the mean.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{1,2}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Hypothesis Testing with 2 Normals: T-test

Degrees of Freedom: the number of values that are free to vary

The number of observations available to measure a parameter in a distribution. In other words, what is the minimum *i*, such that given *i* observations one could determine the parameter?

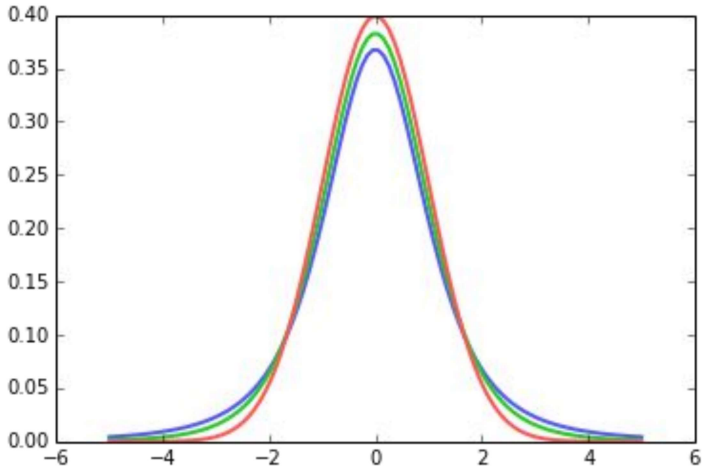
Statistical test is asking about generalizability to the population (or if we had infinite data).

Examples: mean, variance

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{1,2}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Hypothesis Testing with 2 Normals: T-test

Student's *t* distribution:
like normal but adjusted when low degrees of freedom.



$$df_{1,2} = \frac{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2}{\frac{(S_1^2/n_1)^2}{df_1} + \frac{(S_2^2/n_2)^2}{df_2}}$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

(assuming independent, different variance)

$$df = (n_1 - 1) + (n_2 - 1)$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{1,2}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

(assuming independent, similar variance)

T-Test Hypothesis Test “Algorithm”

Input: H_0, sample1, sample2, α

```
obs_t = compute_t(sample1, sample2)
df = compute_df(sample1, sample2)
p(t<=obs_ts | H0) = t.cdf(obs_t, df)
if p(x<=obs_ts | H0) < α:
   decision = “Reject H0!”
else:
   decision = “Accept H0.”
```

Output: decision

Confidence Intervals

Motivation: p-values tell a nice succinct story but neglect a lot of information.

Confidence Intervals

Motivation: p-values tell a nice succinct story but neglect a lot of information.

A mean is only an estimate.
Can we characterize how close it is likely to be to the "true" mean?

Confidence Intervals

Motivation: p-values tell a nice succinct story but neglect a lot of information.

A mean is only an estimate.
Can we characterize how close it is likely to be to the "true" mean?

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \mathrm{SE}_{\bar{x}} = \frac{s}{\sqrt{n}}$$

Confidence Intervals

Motivation: p-values tell a nice succinct story but neglect a lot of information.

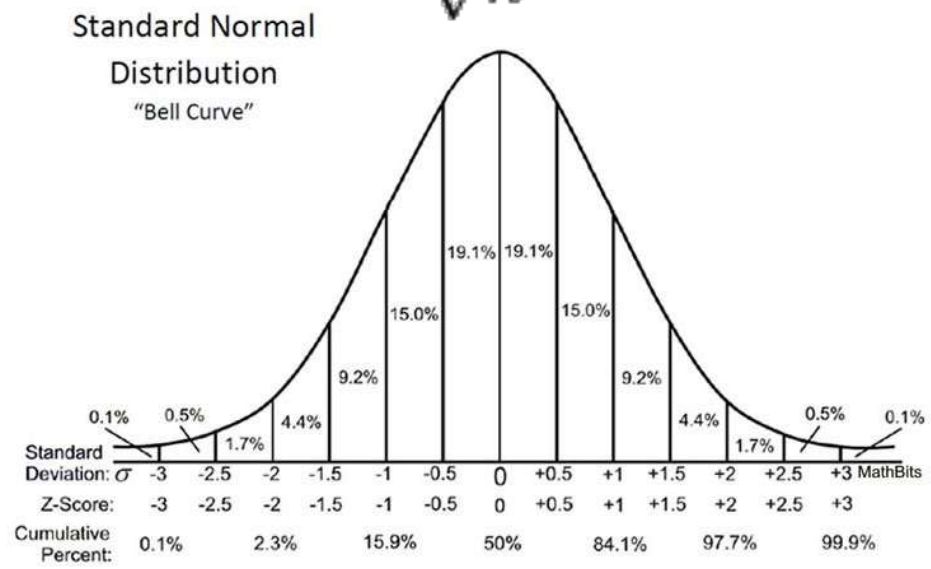
A mean is only an estimate.

Can we characterize how close it is likely to be to the "true" mean?

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\mathrm{SE}_{\bar{x}} = \frac{s}{\sqrt{n}}$$

$$z_i = \frac{x_i - \bar{X}}{s}$$



Confidence Intervals

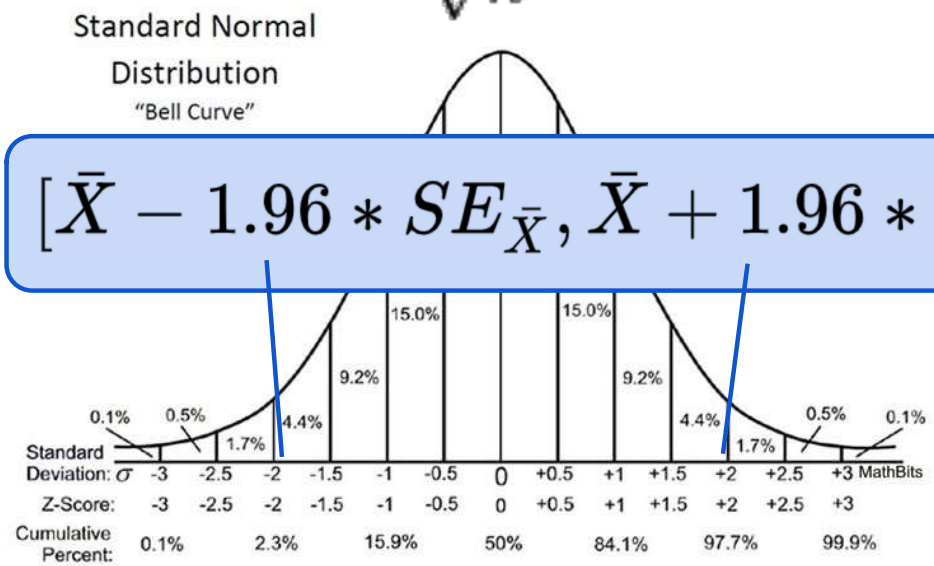
Motivation: p-values tell a nice succinct story but neglect a lot of information.

A mean is only an estimate.

Can we characterize how close it is likely to be to the "true" mean?

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \mathrm{SE}_{\bar{x}} = \frac{s}{\sqrt{n}} \qquad z_i = \frac{x_i - \bar{X}}{s}$$

$$\left[\bar{X} - 1.96 * SE_{\bar{X}}, \bar{X} + 1.96 * SE_{\bar{X}}\right]$$



Confidence Intervals

The bootstrap

- What if we don't know the distribution?
- *Resample* many potential distributions based on the observed data and find the range that CI% of the data fall in (e.g. mean).
- *Resample:* for each *i* in *n* observations, put all observations in a hat and draw one (all observations are equally likely).
- For confidence interval: grab the middle 95% of resampled means.

(see code)