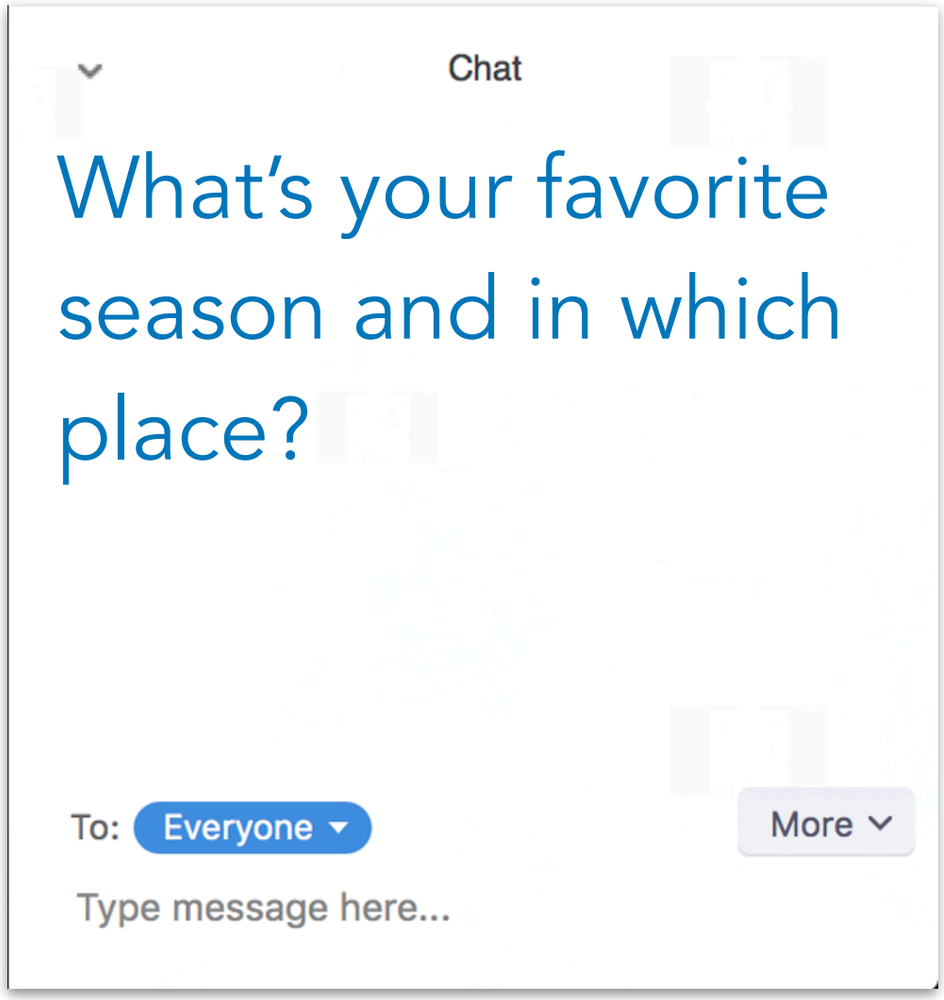
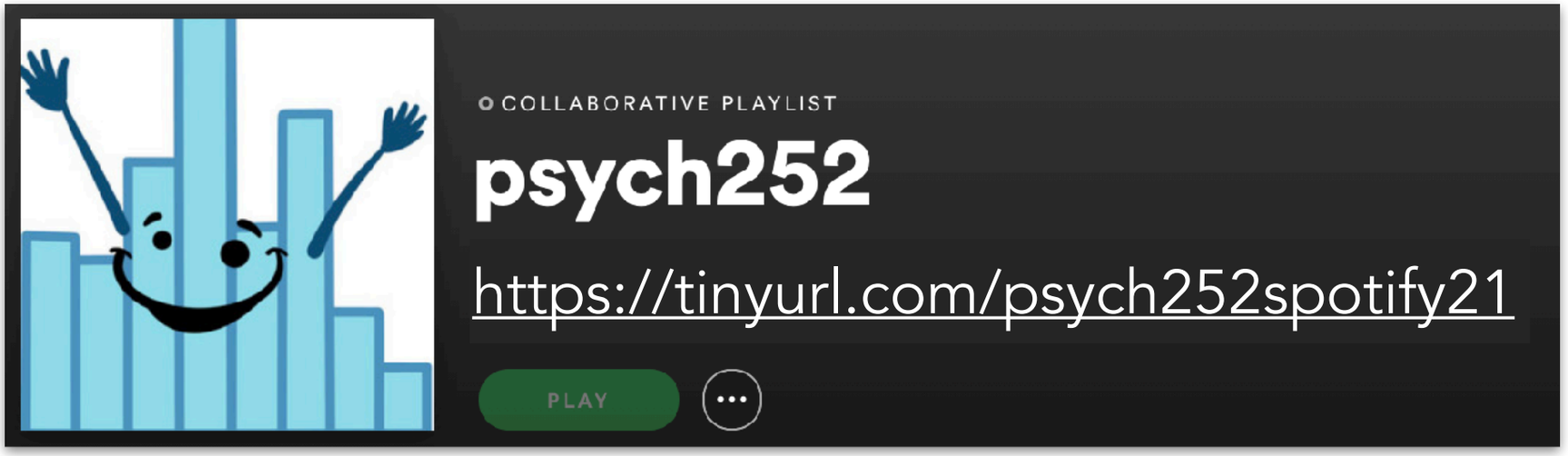
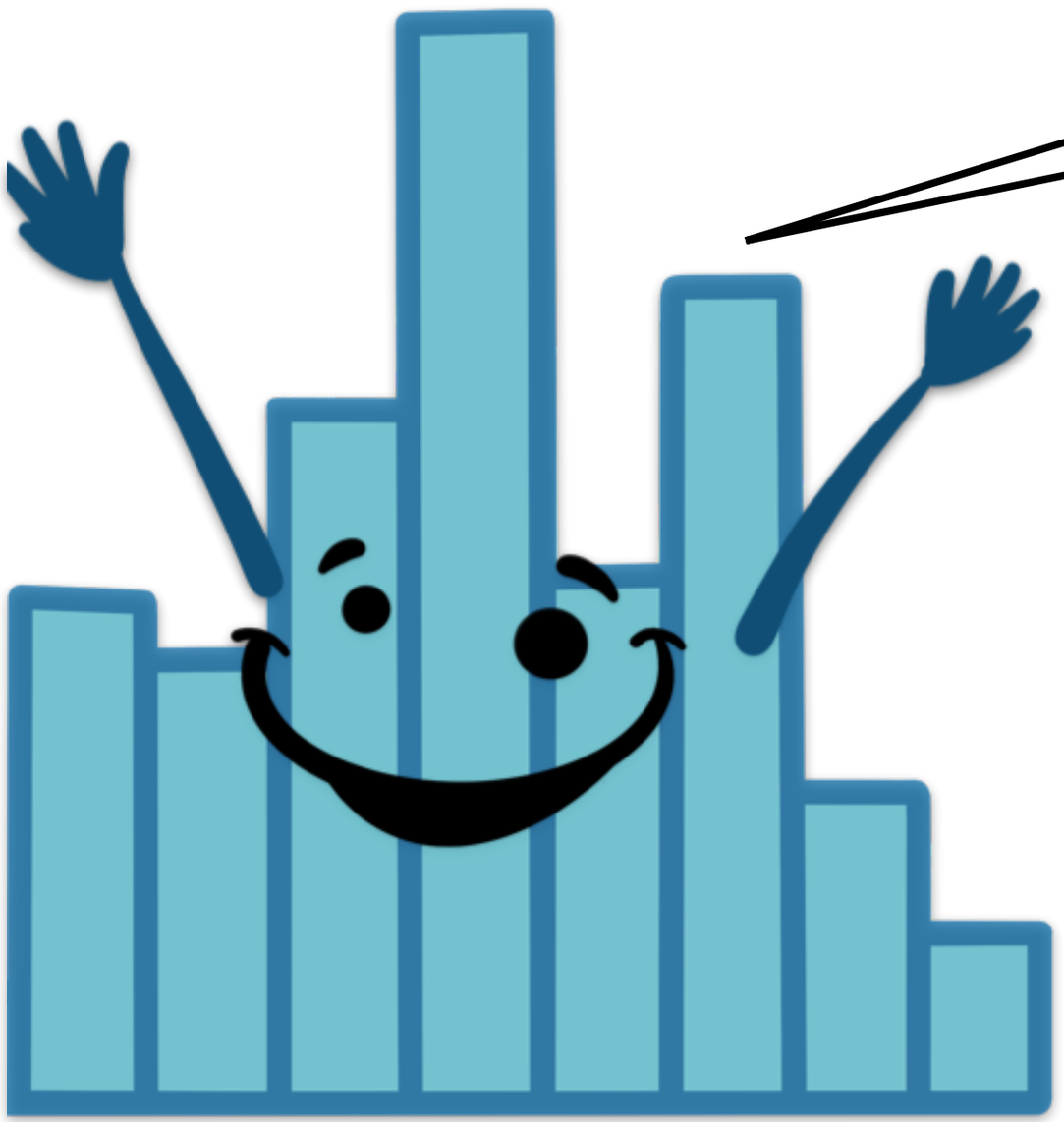
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Modeling data





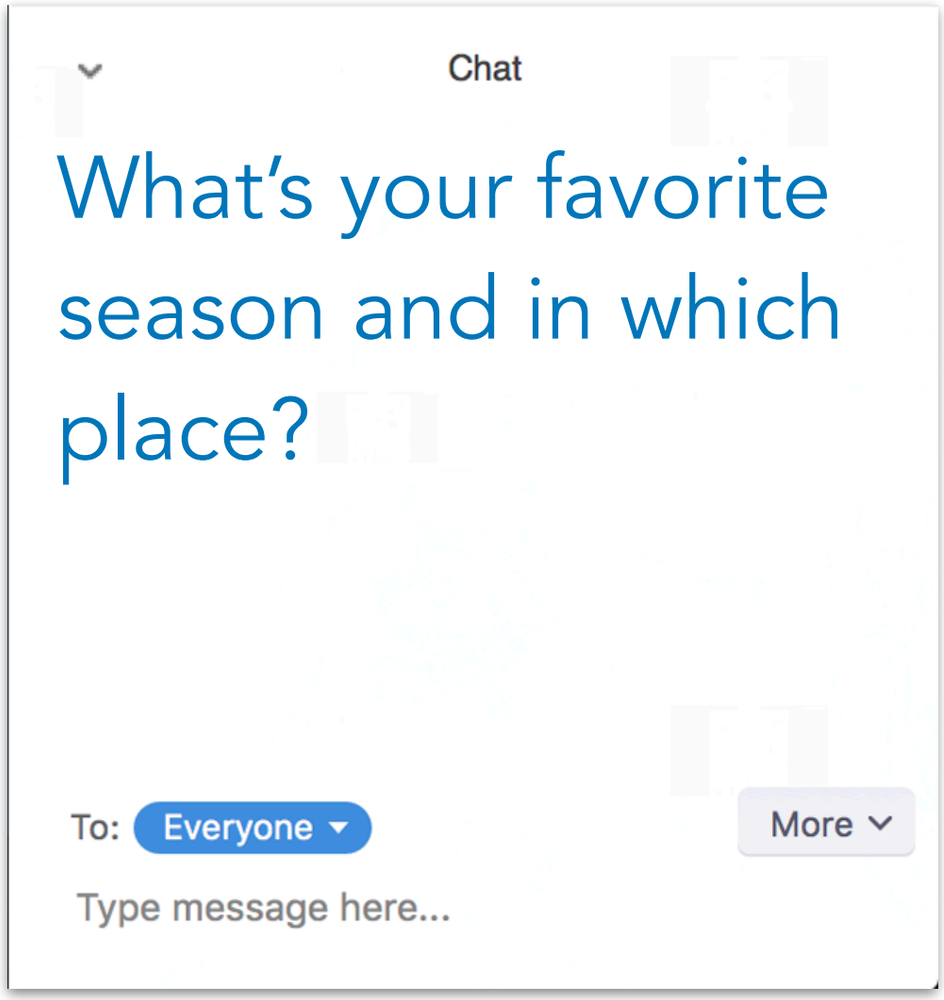
02/01/2021

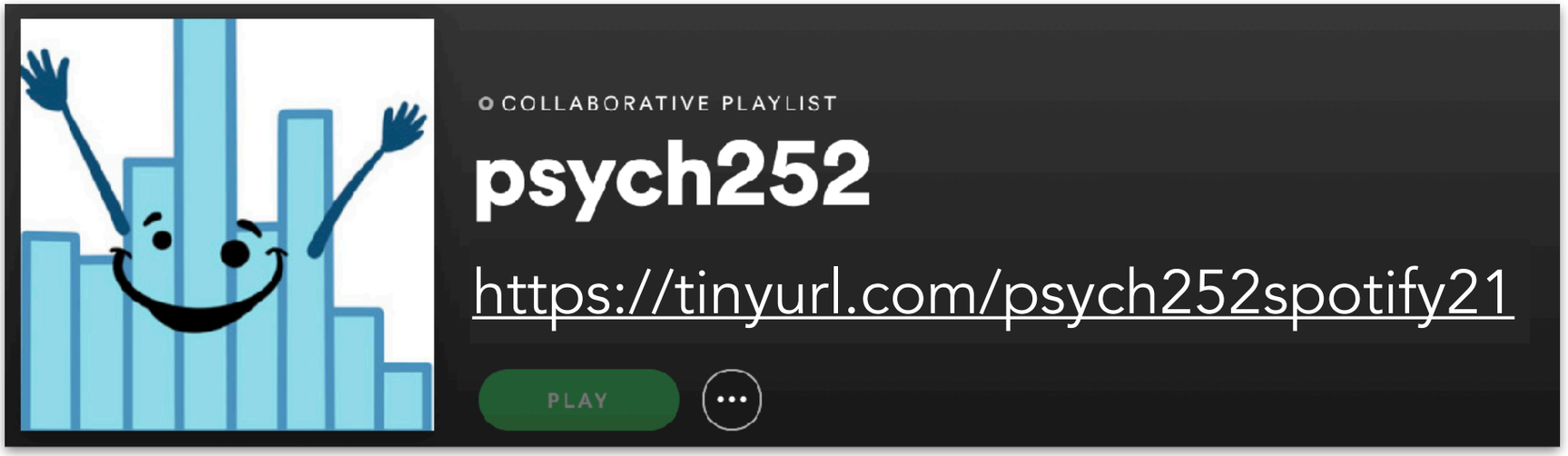


Remember to record the lecture!

PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Modeling data

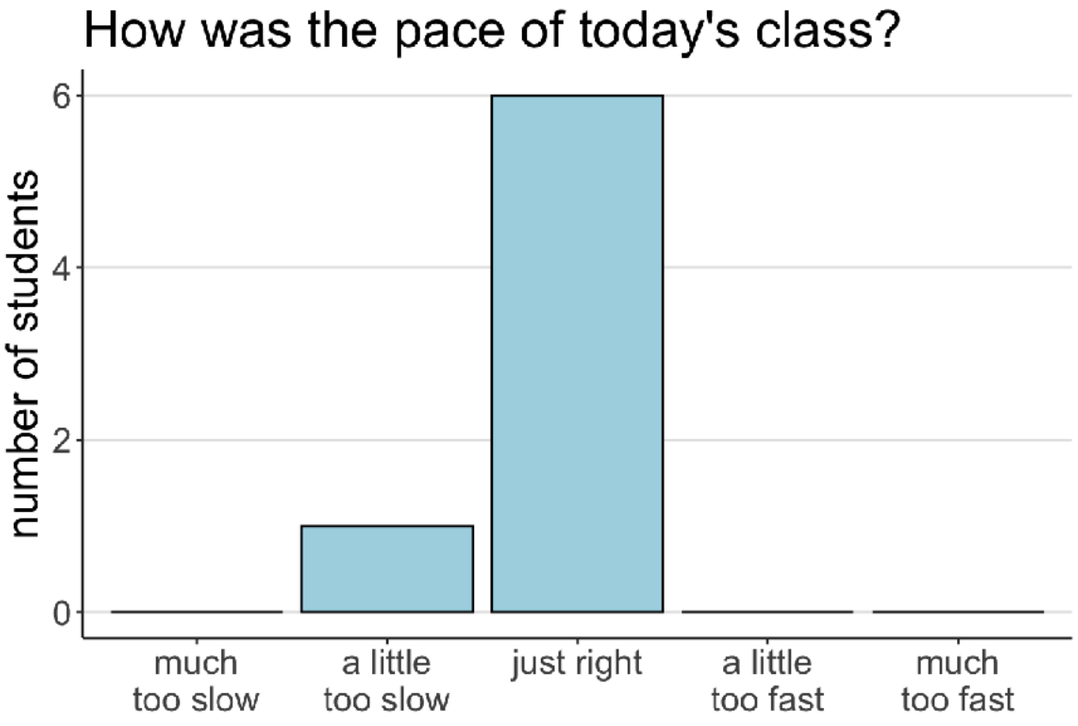


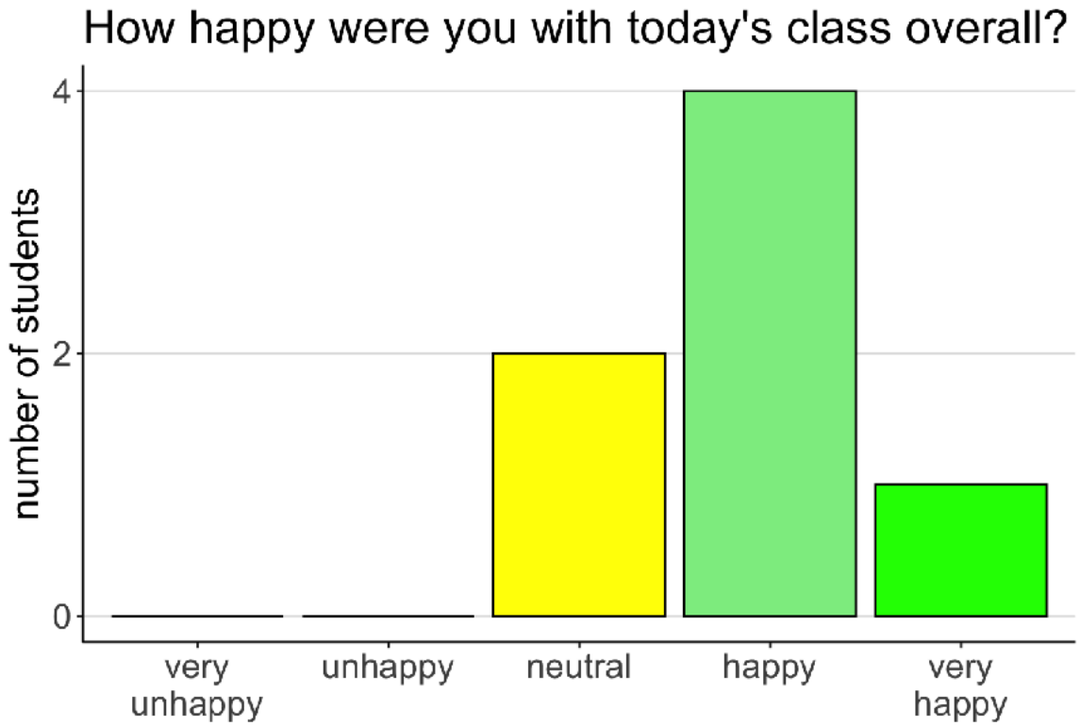


02/01/2021

Feedback

Your feedback





i'll try to elicit feedback early enough, so it would be great if you could stick around. it helps me a lot because it's so difficult to read the room.

Your feedback

Toni is my auto-correct alter ego

Toni, your explanation of p-values was the most sensical explanation I have ever heard.

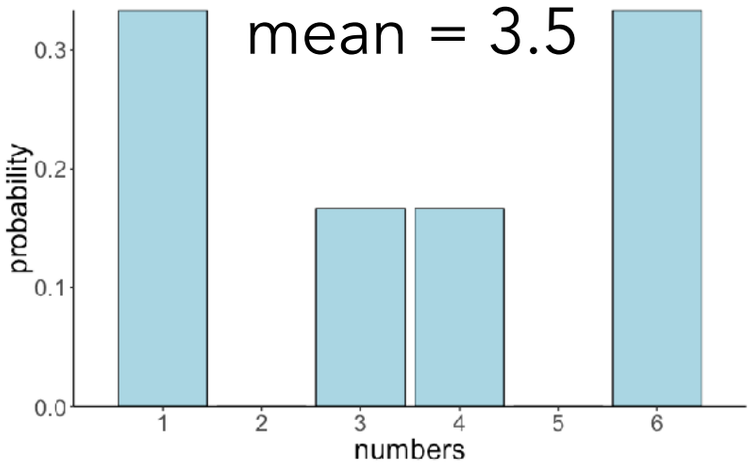
I find it a little difficult to go back and forth between the R code and the concept. Not sure if it's possible but if we could finish a review of the concept and then go over the R code related to that concept it'd help me out I think

I will try and keep it conceptual first, and then illustrate via R if I think it'll be useful.

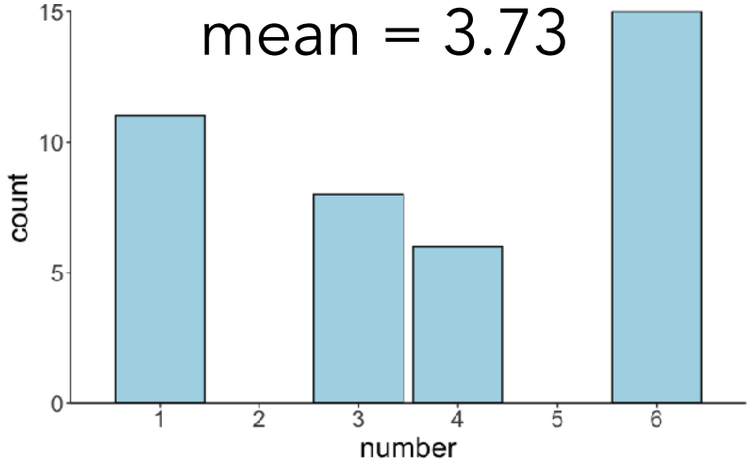
Things that came up ...

Bootstrapping and sampling distribution

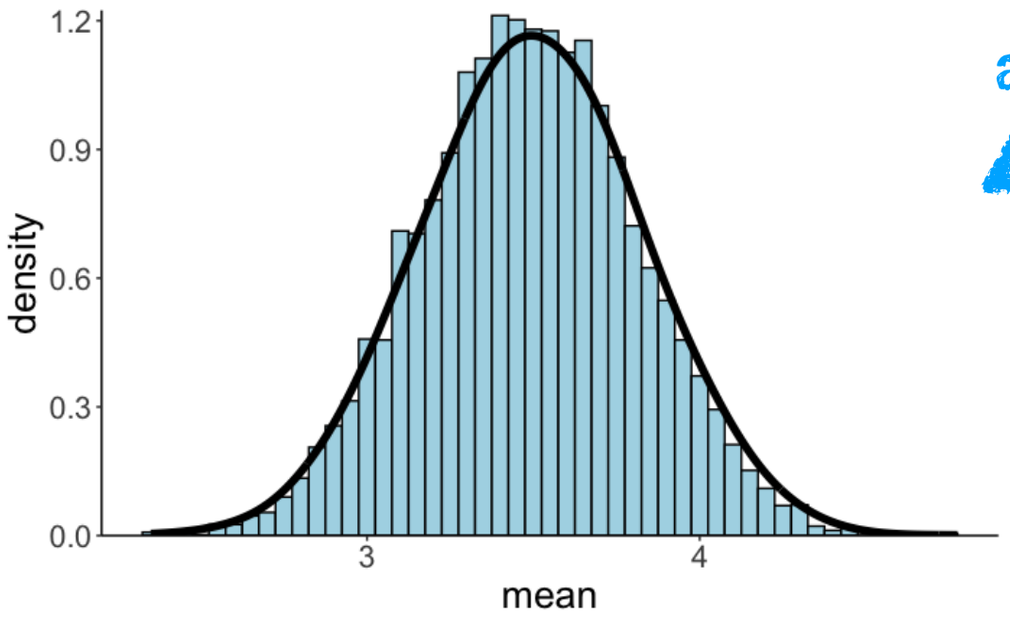
1) Sampling distribution



approximates

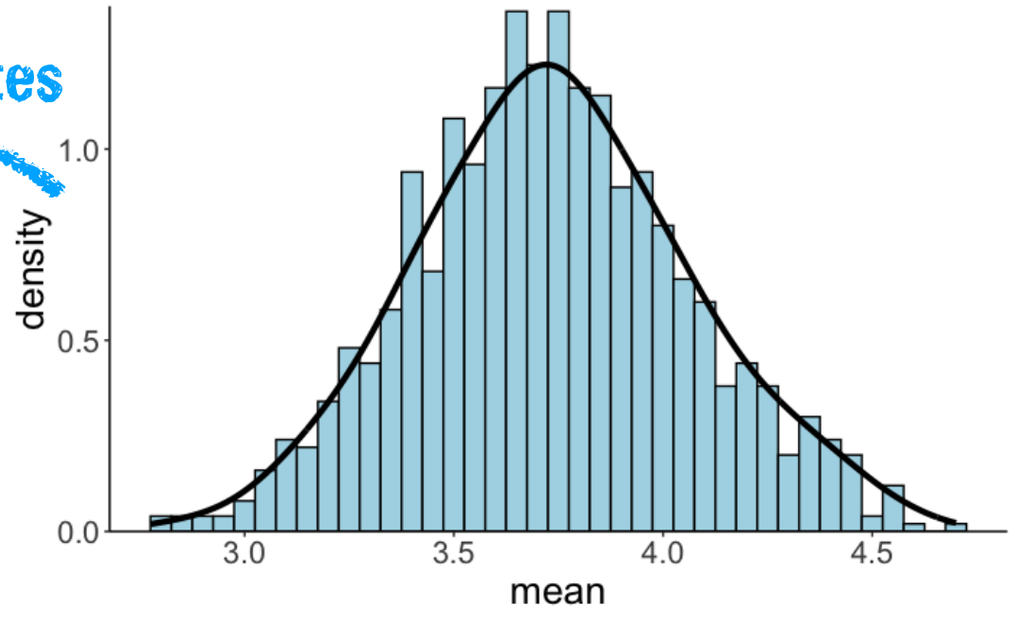


generate from the true population distribution



approximates

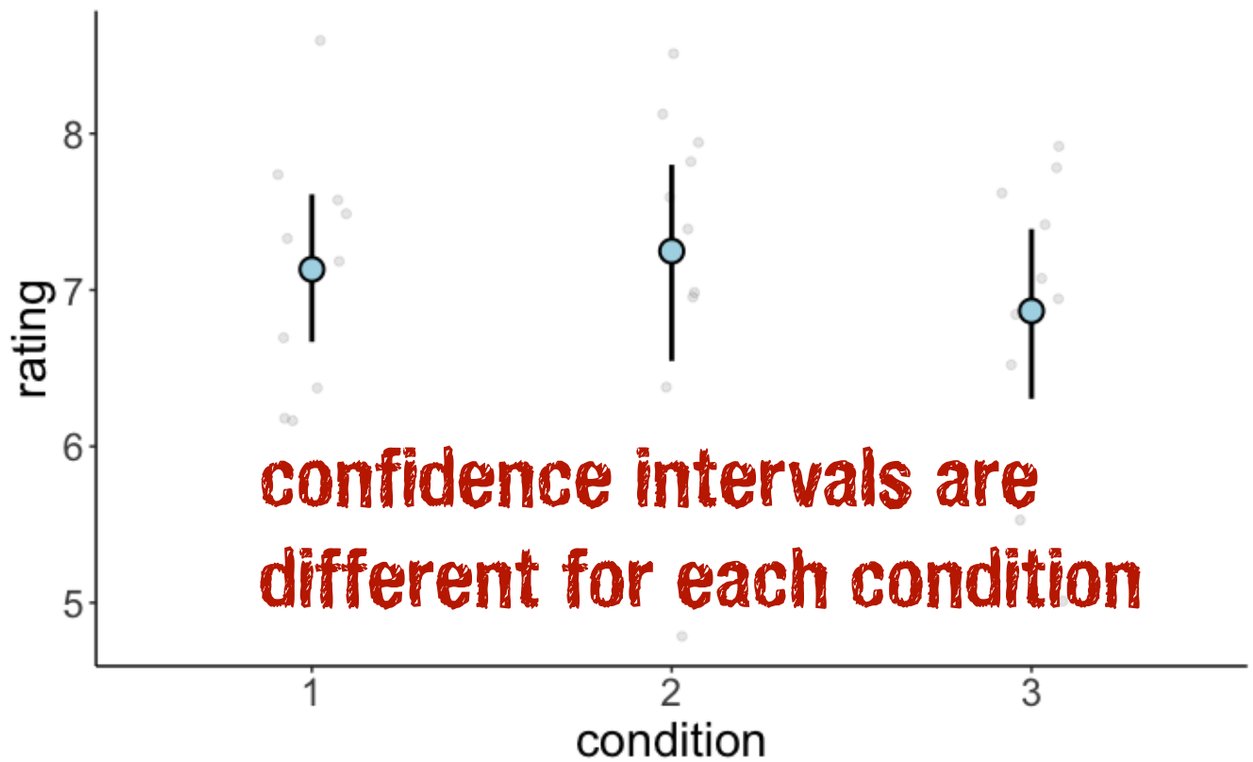
generate from this sample via bootstrapping



mean_cl_boot() explained

```r
set.seed(1)

n = 10 # sample size per group
k = 3 # number of groups

df.data = tibble(participant = 1:(n*k),
                 condition = as.factor(rep(1:k, each = n)),
                 rating = rnorm(n*k, mean = 7, sd = 1))

ggplot(data = df.data,
       mapping = aes(x = condition,
                     y = rating)) +
  geom_point(alpha = 0.1,
             position = position_jitter(width = 0.1, height = 0)) +
  stat_summary(fun.data = "mean_cl_boot",
               shape = 21,
               size = 1,
               fill = "lightblue")
```

participant	condition	rating
1	1	6.37
2	1	7.18
3	1	6.16
4	1	8.60
5	1	7.33

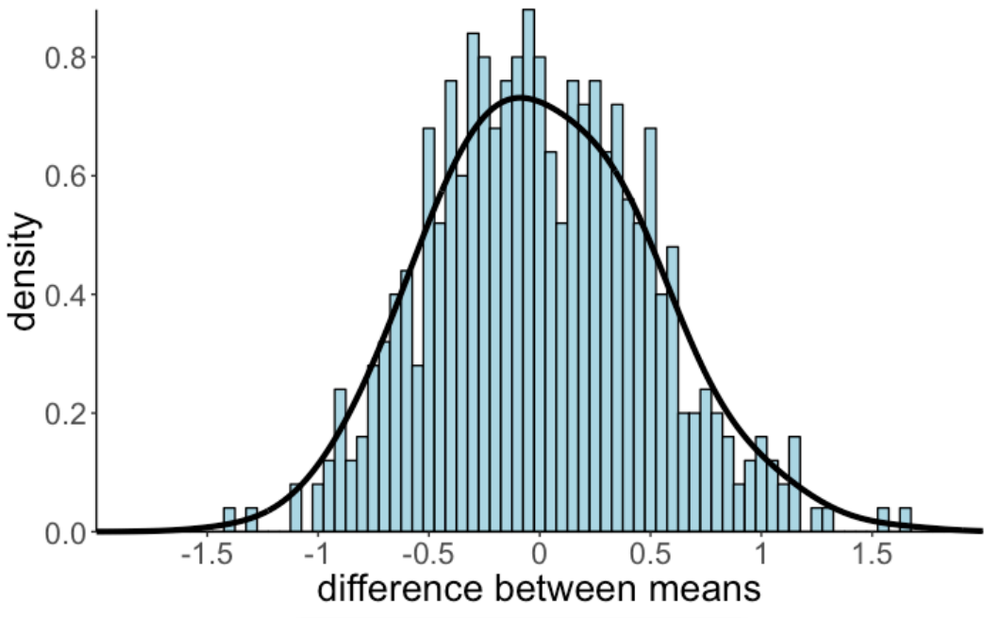


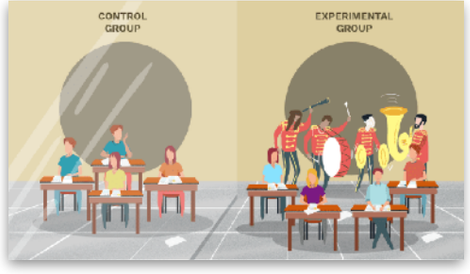
Bootstrapping and sampling distribution

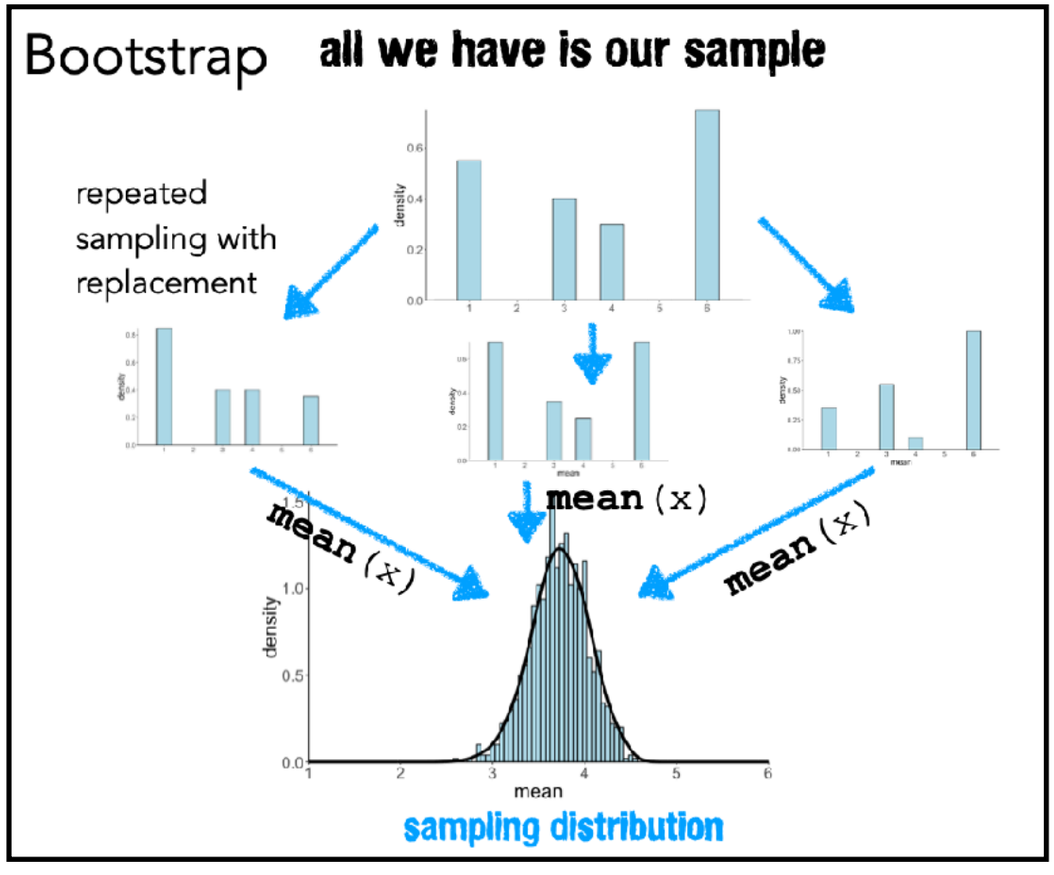
2) What kinds of sampling distributions?

we can generate a sampling distribution for any measure, for example: mean, standard deviation, difference between two conditions, t-statistic, F-statistic, ...

Permutation test

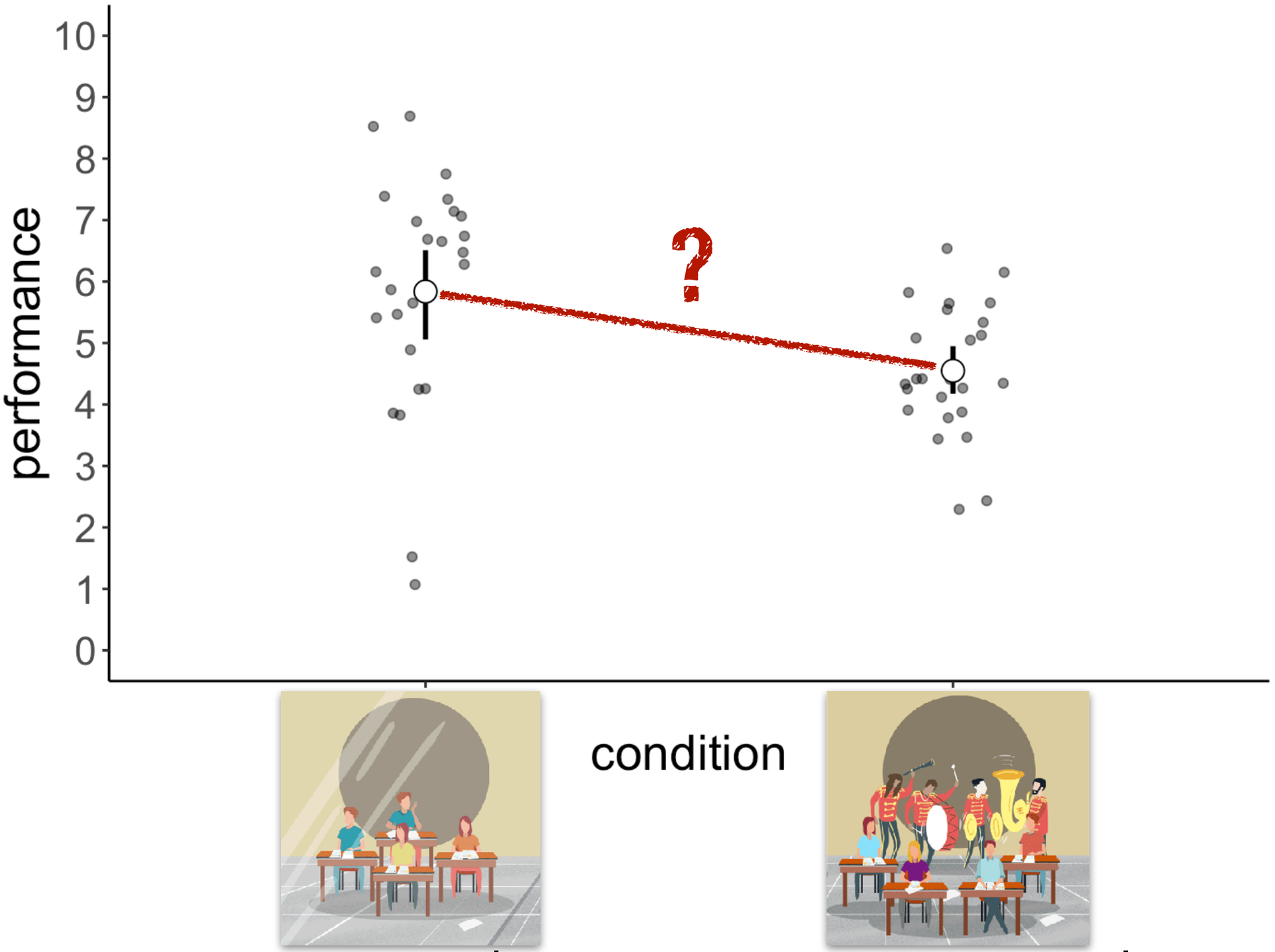




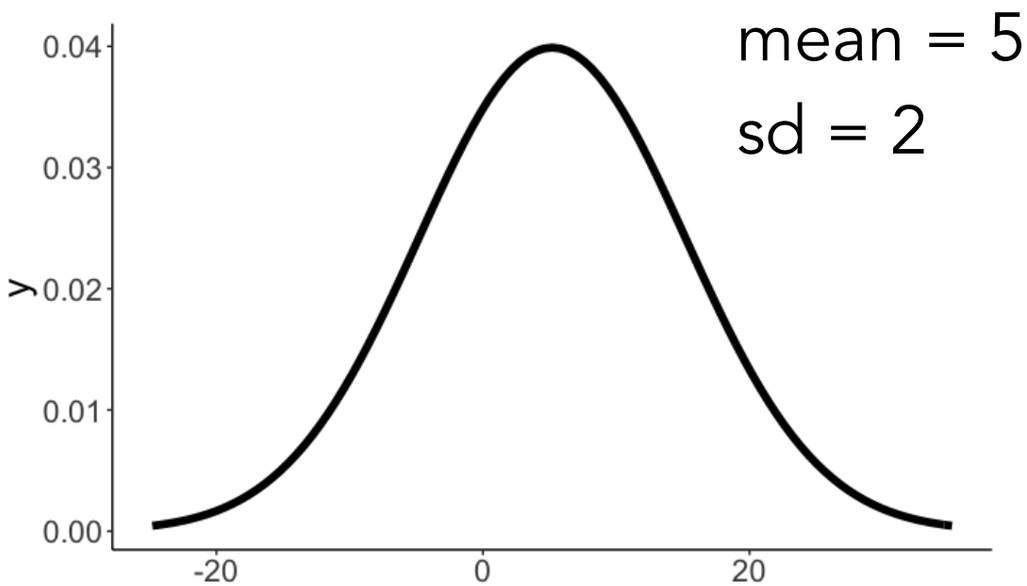


two sample t-test

Is the difference in performance statistically significant?



two-sample t-test

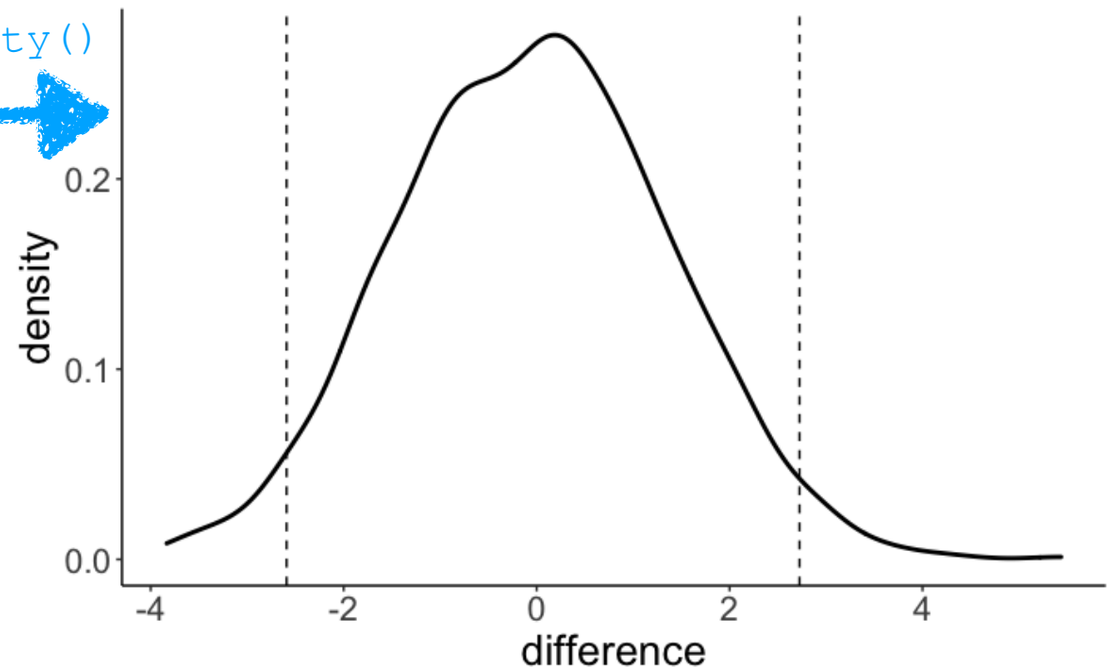



```r
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 2

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

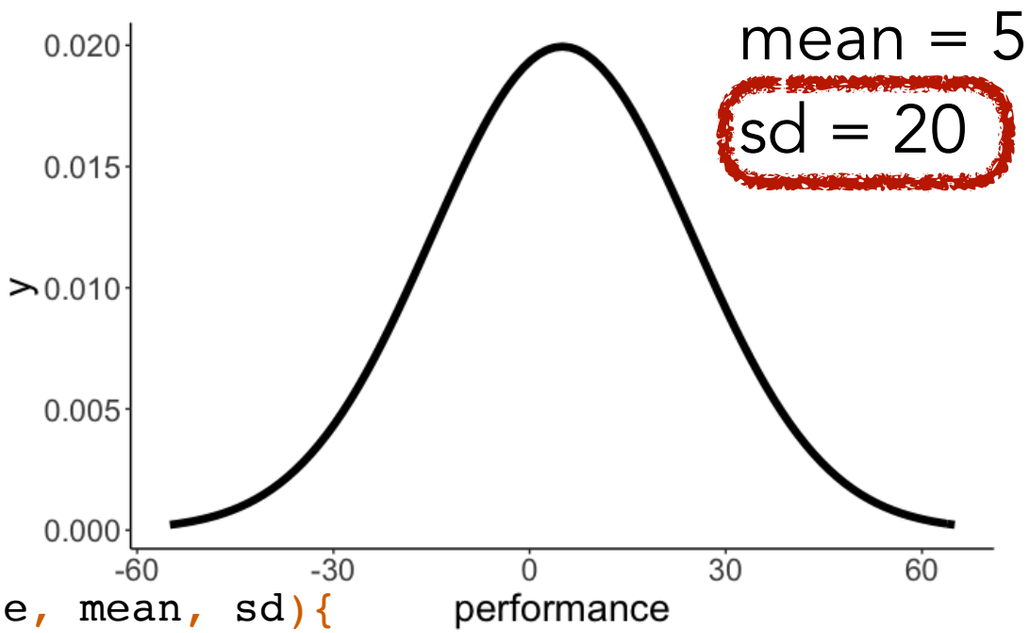
simulation	sample1	sample2	difference
1	6.28	5.16	1.13

geom_density()

What if sd = 20?



two-sample t-test

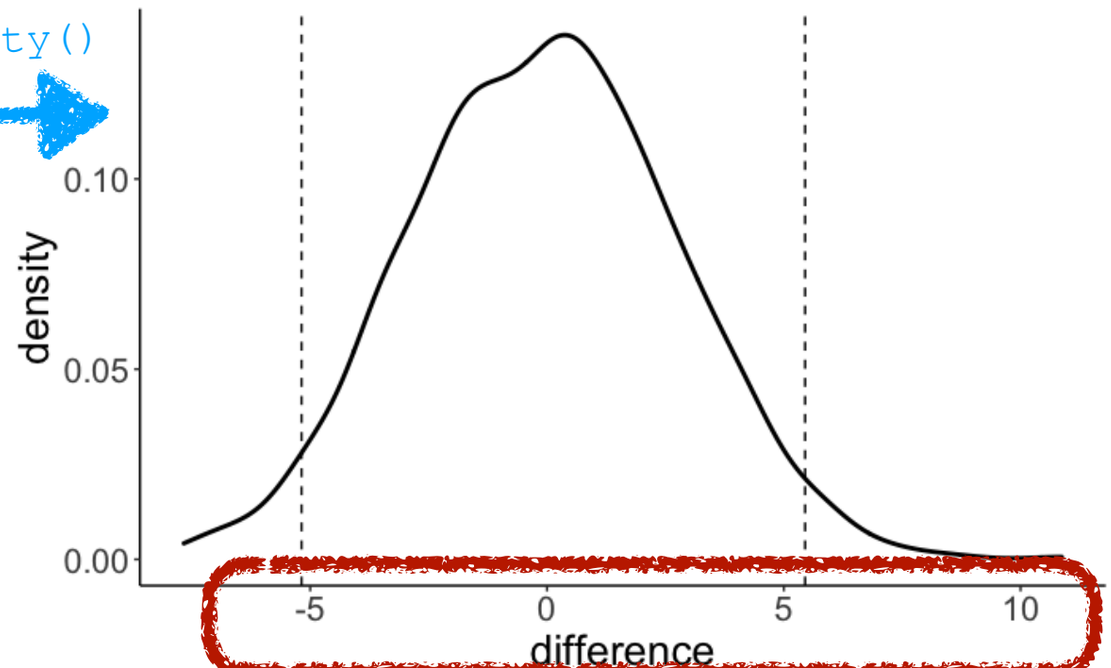



```r
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 20

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

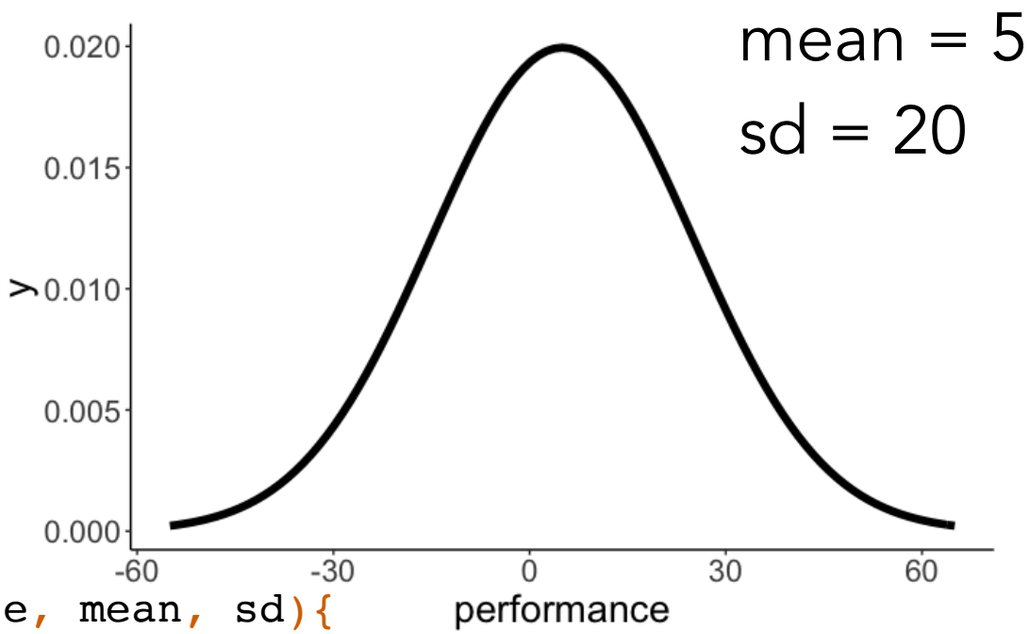
What if sample size N = 1000?

simulation	sample1	sample2	difference
1	7.18	4.93	2.25
2	4.24	5.57	-1.32
3	5.59	8.99	-3.40
4	6.03	4.71	1.32
5	4.22	2.48	1.73
⋮			

geom_density()



two-sample t-test



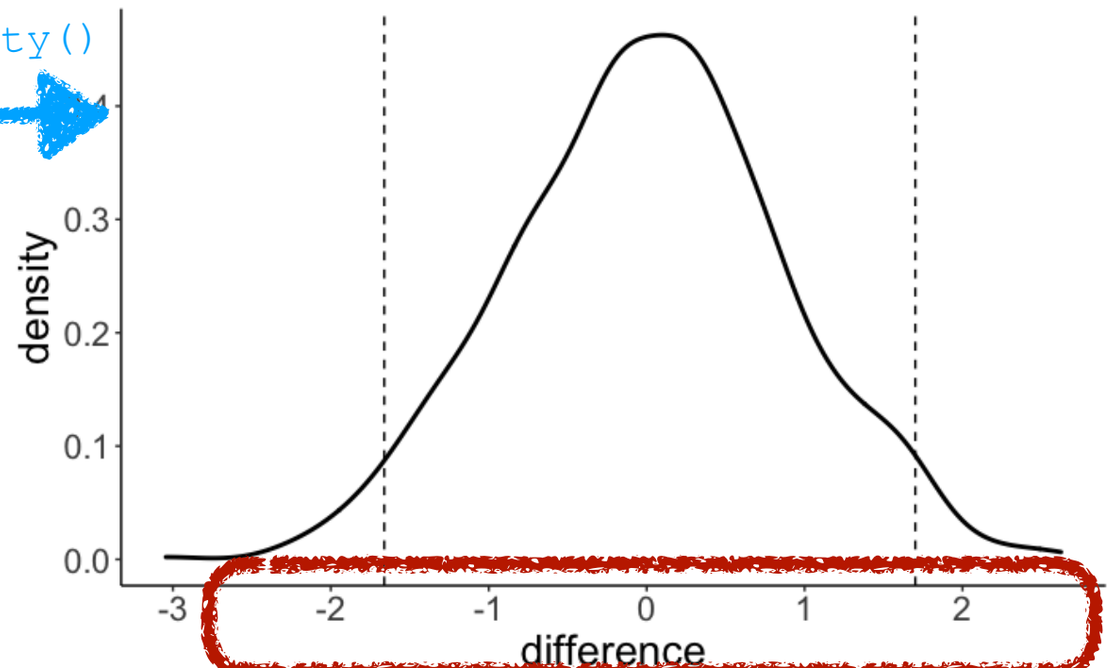

```r
set.seed(1)

n_simulations = 1000
sample_size = 1000
mean = 5
sd = 20

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

simulation	sample1	sample2	difference
1	4.77	4.77	0.00
2	4.67	5.10	-0.43
3	5.31	5.87	-0.56
4	5.33	6.28	-0.94
5	4.60	5.52	-0.92

⋮

geom_density()

What if sample size N = 1000?



two-sample t-test

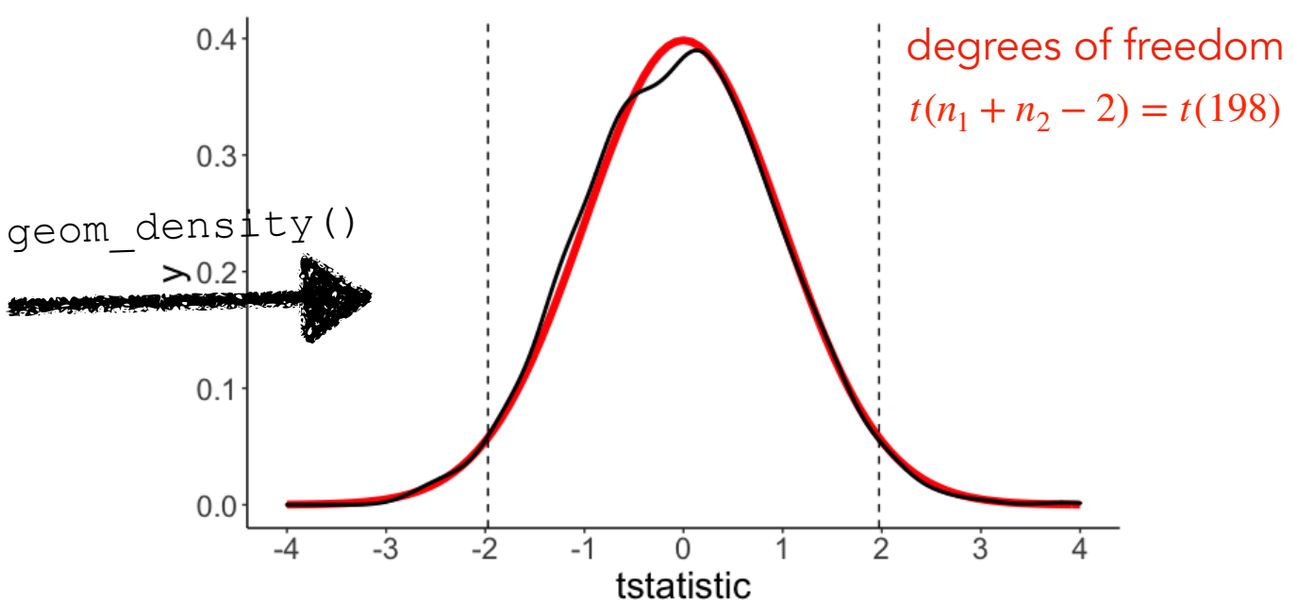
difference in means (relative to the standard deviation and sample size)

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2(\frac{1}{n_1} + \frac{1}{n_2})}}$$

pooled sample variance

```
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 2

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2,
         # assuming the same standard deviation in each sample
         tstatistic = difference / sqrt(sd^2 * (2/sample_size)))
```

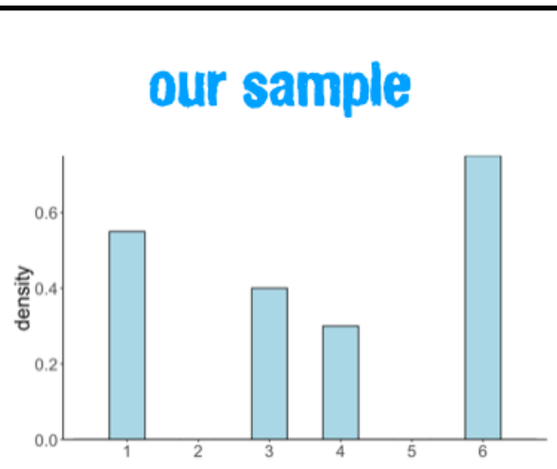
simulation	sample1	sample2	difference	tstatistic
1	5.22	4.99	0.23	0.80
2	4.92	5.06	-0.13	-0.47
3	5.06	5.40	-0.34	-1.20
4	5.10	4.97	0.13	0.47
5	4.92	4.75	0.17	0.61



Bootstrapping and sampling distribution

3) What is the standard error

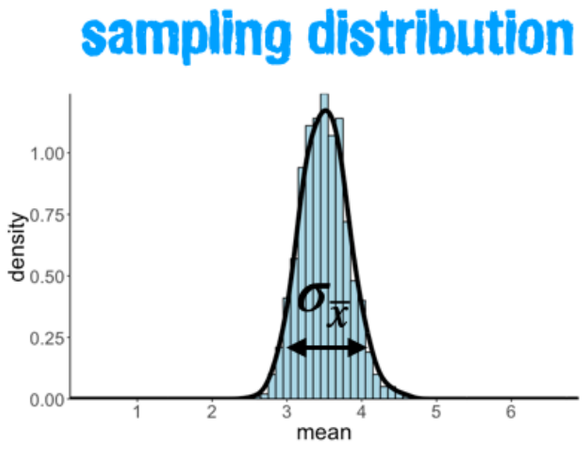
the standard error refers to the standard deviation of the sampling distribution



standard deviation

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}}$$

gives a sense for how well the mean captures the data



standard error of the mean

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$$

$\sigma_{\bar{x}}$ decreases as:

σ decreases

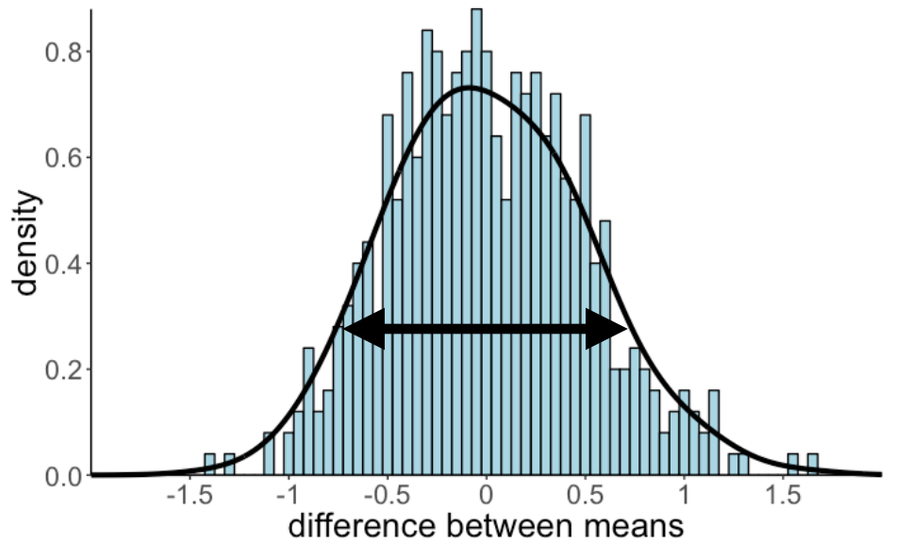
N increases

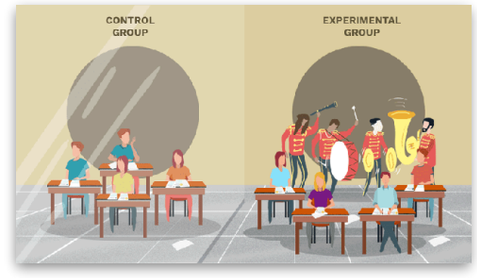
$\hat{\sigma} = s$ for large enough samples (> 30)

we are more confident in our inference with larger samples, and less variance

59

standard error of the difference between two means





Open questions

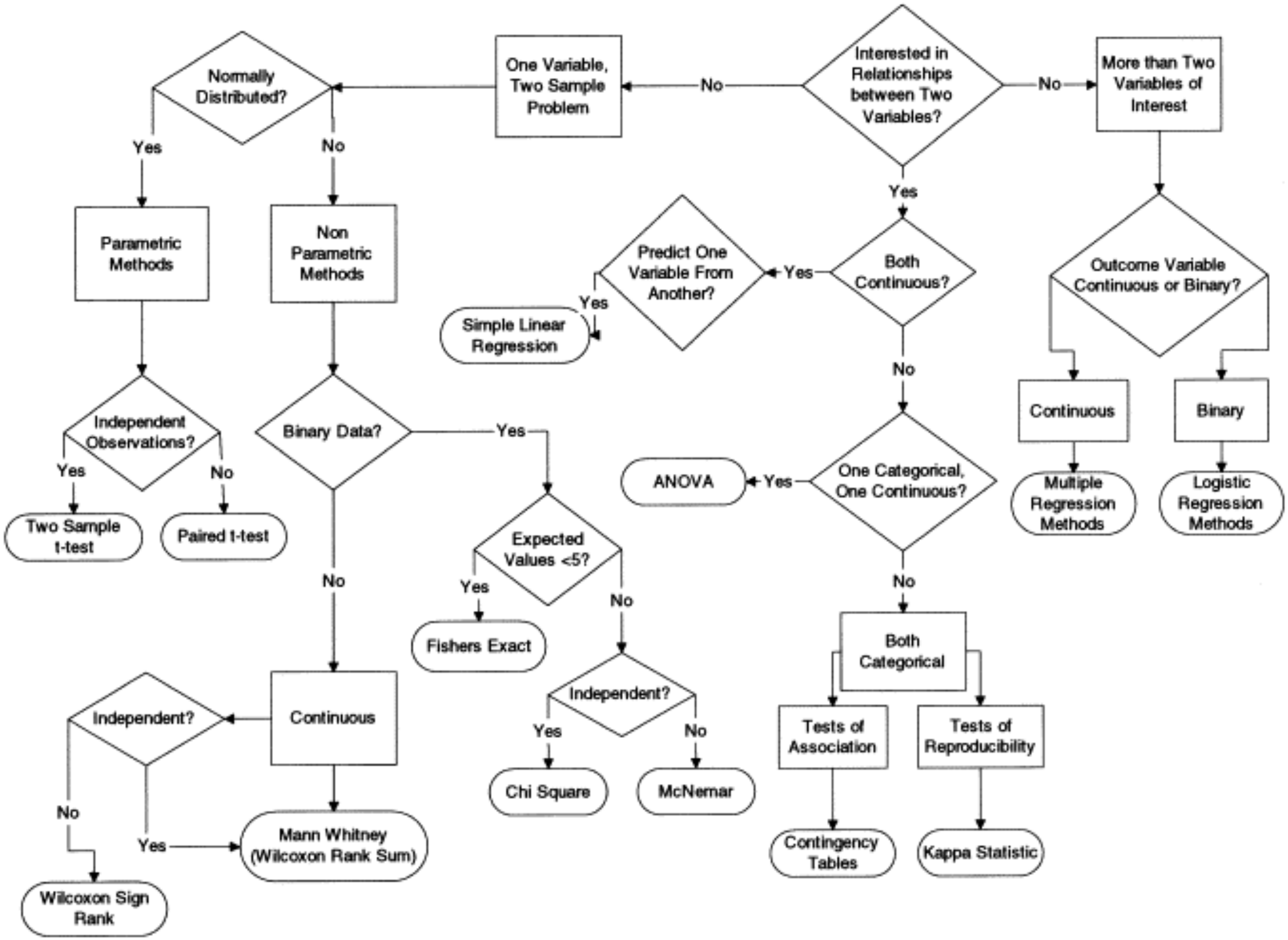
- why not always do bootstrapping?

Plan for today

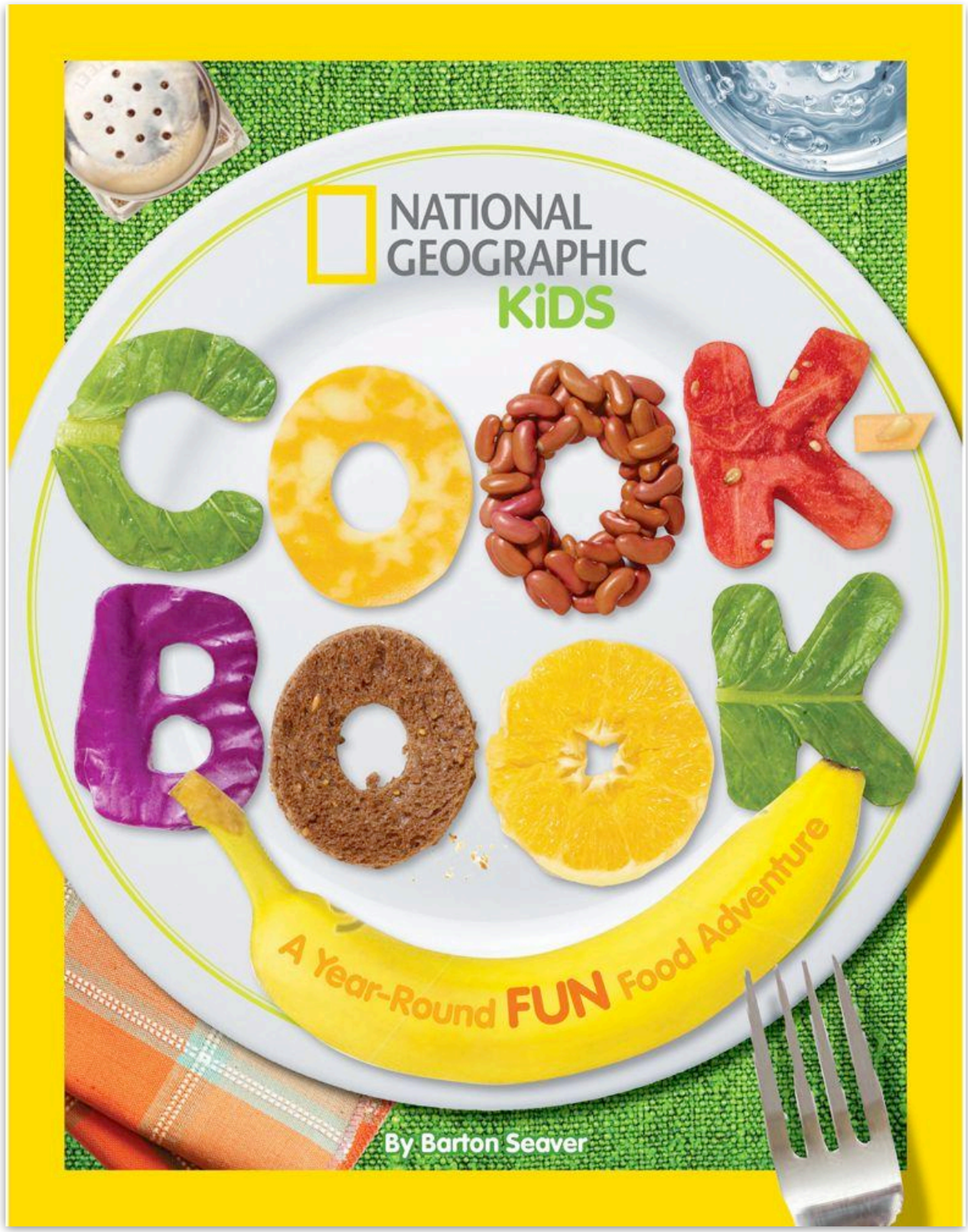
- Cookbook vs. Model Comparison
- Modeling data
- Definitions of error and parameter estimates
- Models of error
- Statistical inferences about parameter values

Cookbook vs. Model Comparison

The cookbook approach

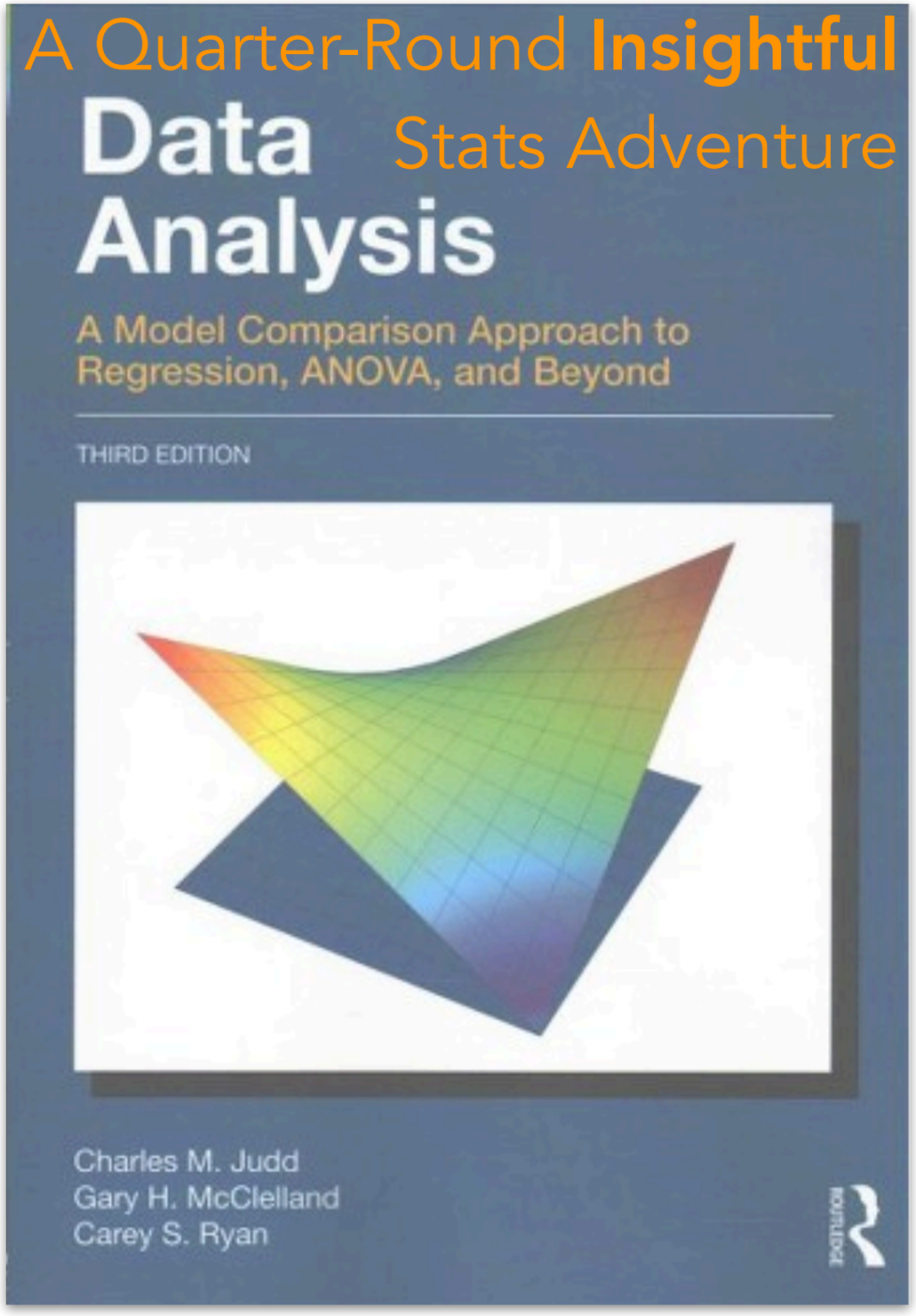


The cookbook approach



- many statistics textbooks are organized in this way
- works reasonably well if what we want to cook is in the book
- leaves us with no idea what to do if we can't find a recipe

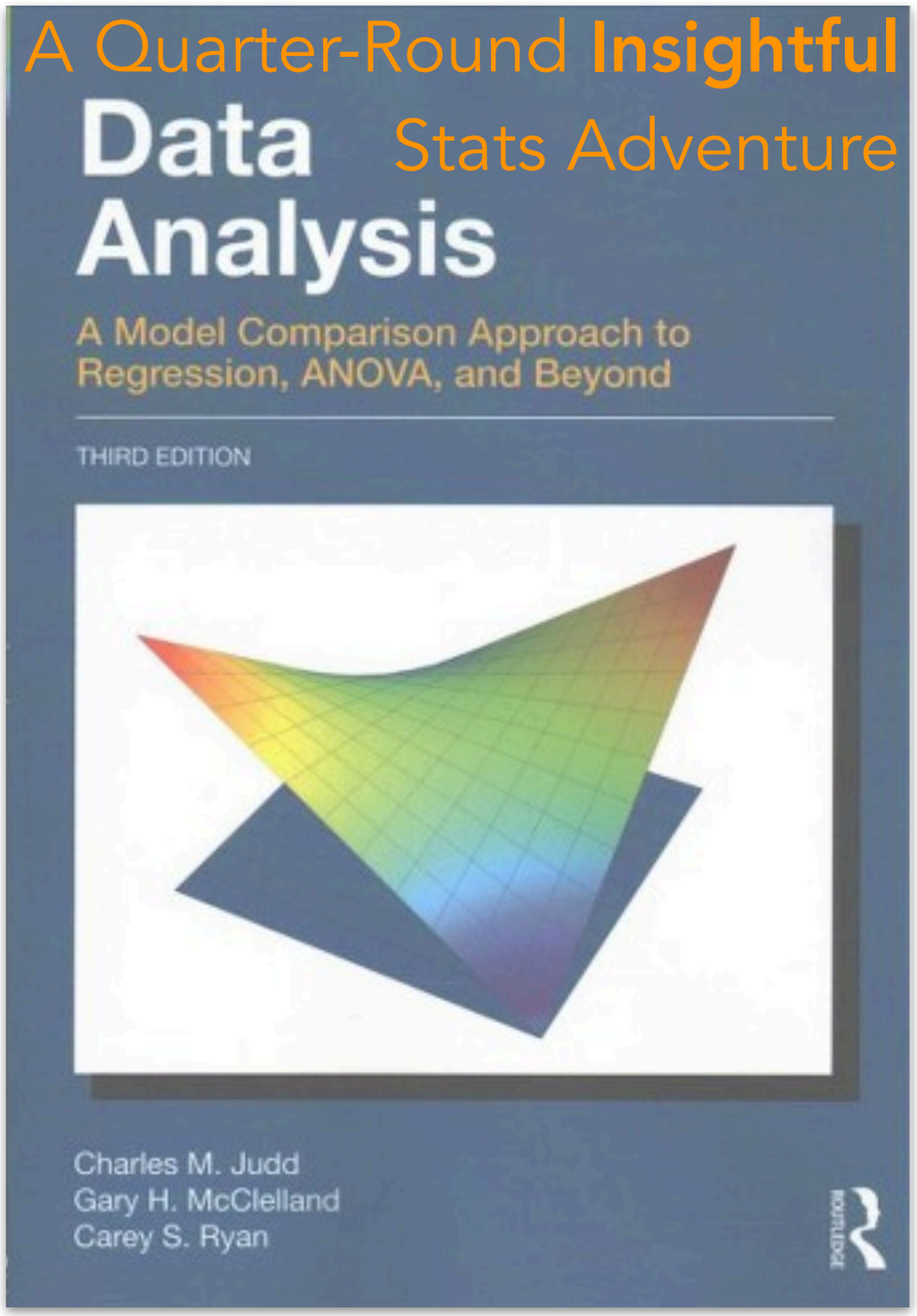
Model comparison approach



Judd, C. M., McClelland, G. H., & Ryan, C. S. (2011). *Data analysis: A model comparison approach*. Routledge.

Model comparison approach

- more flexible approach
- hopefully generates better insight
- thinking of statistical analysis as modeling
- allows for a smoother transition into Bayesian data analysis, and probabilistic modeling more generally



Judd, C. M., McClelland, G. H., & Ryan, C. S. (2011). *Data analysis: A model comparison approach*. Routledge.

Modeling data

$$\text{Data} = \text{Model} + \text{Error}$$

what's a good model?

how shall we define this?

= residual: the part that's left over after we have used the model to predict/explain the data

$$\text{Error} = \text{Data} - \text{Model}$$

to reduce error we can:

improve the quality of the data

e.g. run good experiments

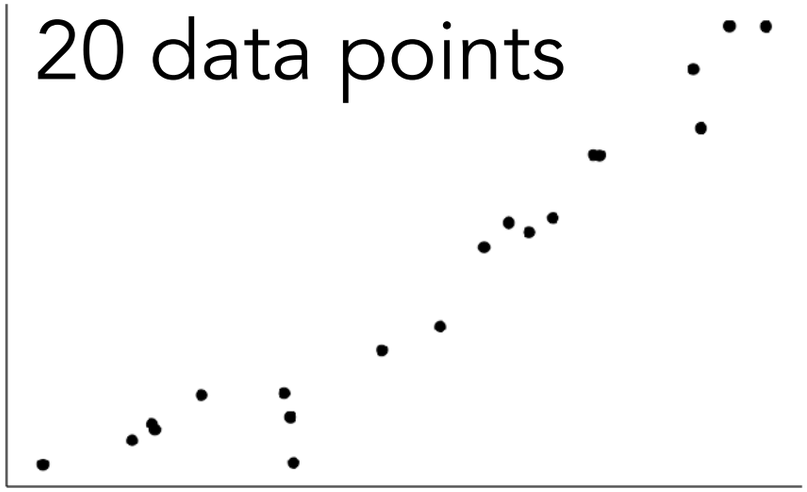
improve the model

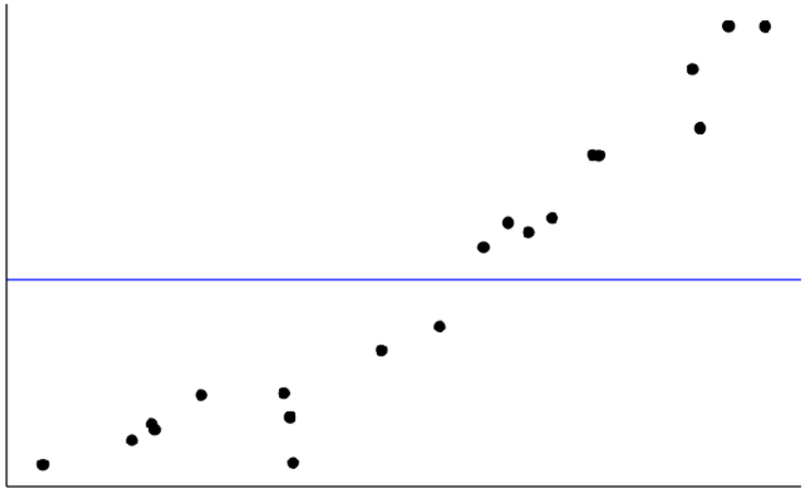
e.g. make predictions conditional on additional information

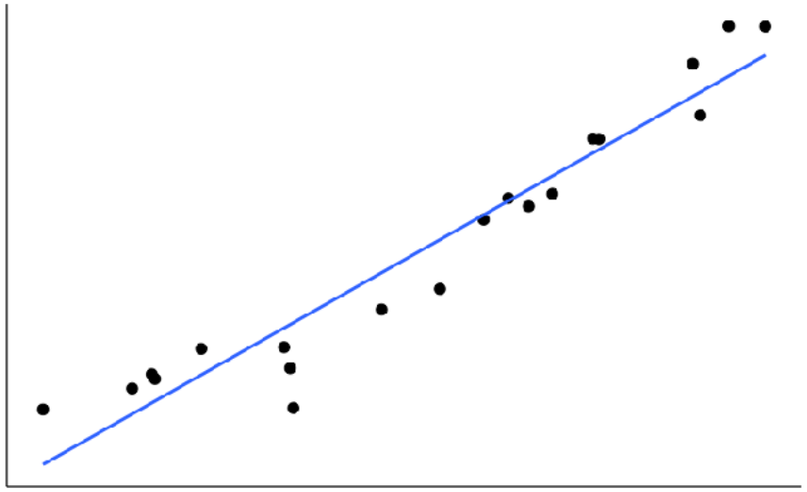
Error = Data - Model

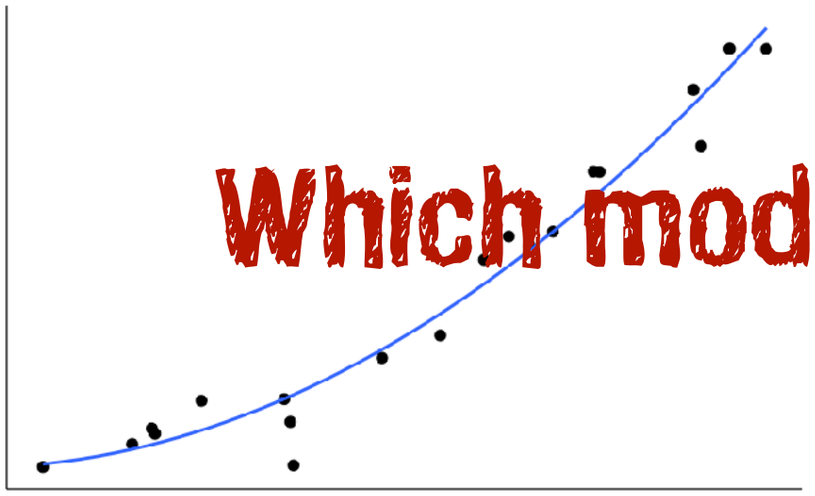
- we build models with parameters, and fit those parameters to minimize error
- adding additional parameters to the model will always improve the model fit and reduce error
- fundamental trade-off between **simplicity** and **accuracy**

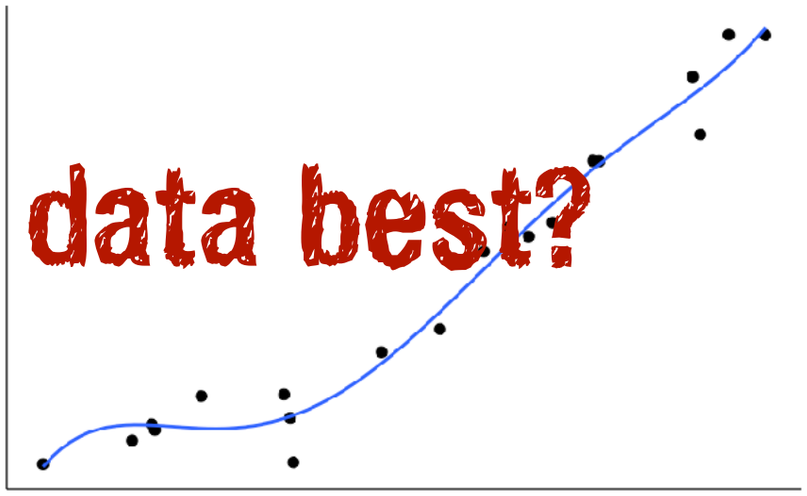
20 data points



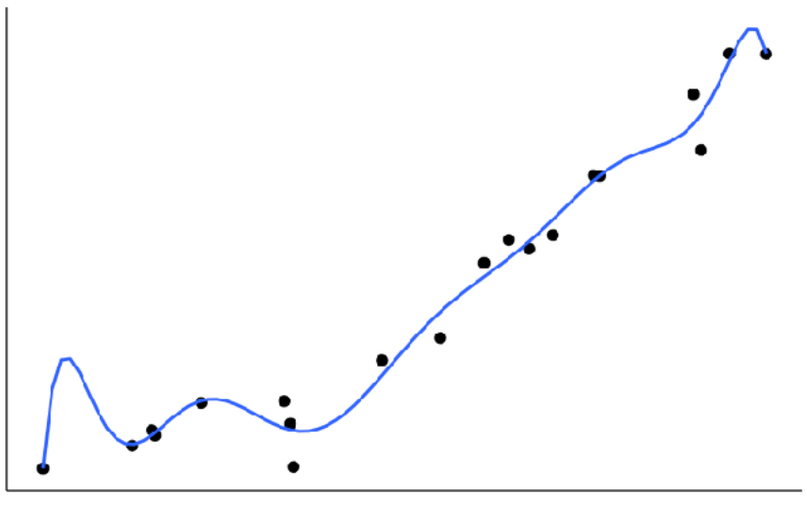


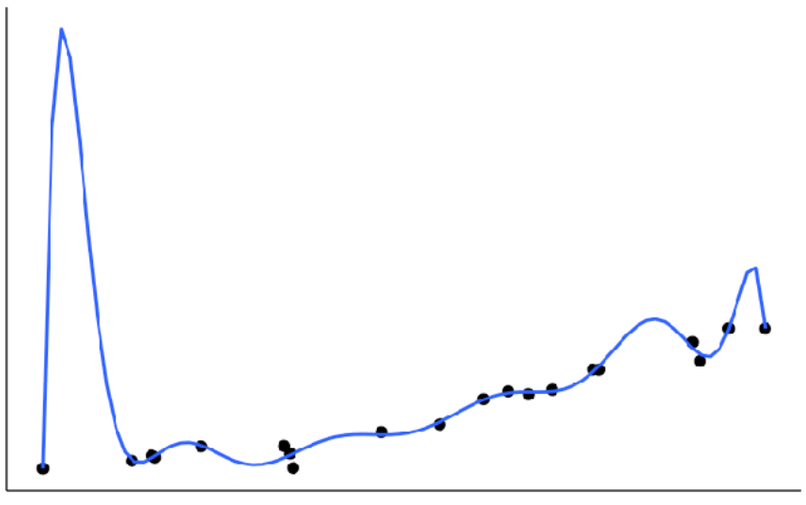


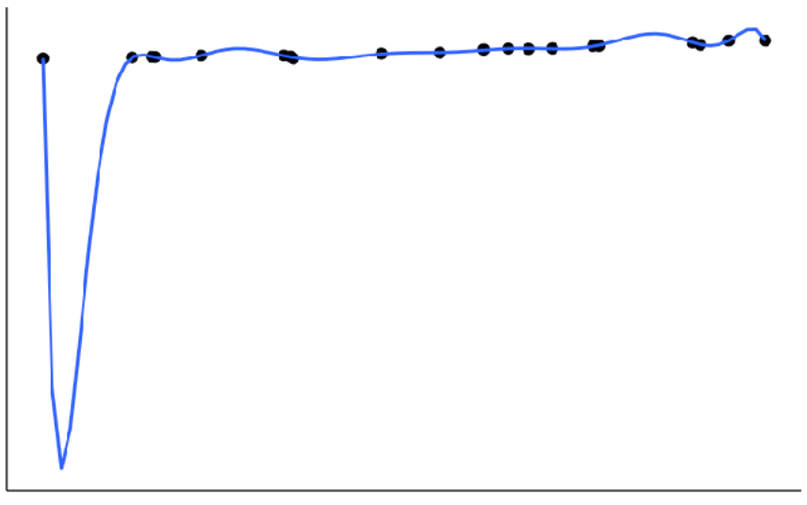




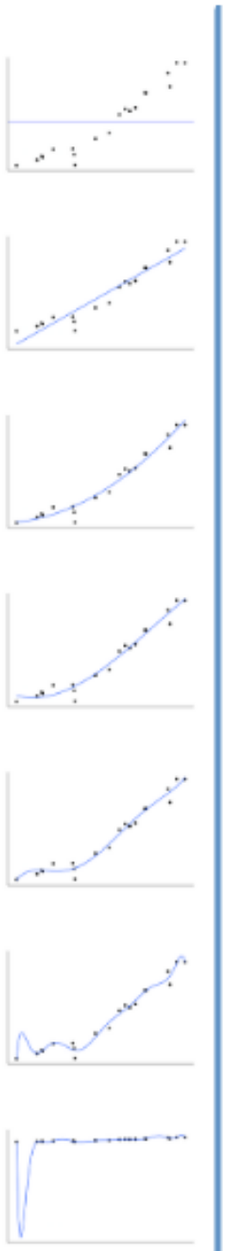
Which model describes the data best?







Which model describes the data best




THE BEST WAY TO
EXPLAIN OVERFITTING
imgflip.com

Example

Percentage of households that had internet access in the year 2013 by US state

i	internet	state	college	auto	density
1	79.0	AK	28.0	1.2	1.2
2	63.5	AL	23.5	1.3	94.4
3	60.9	AR	20.6	1.7	56.0
4	73.9	AZ	27.4	1.3	56.3
5	77.9	CA	31.0	0.8	239.1
6	79.4	CO	37.8	1.0	48.5
7	77.5	CT	37.2	1.0	738.1
8	74.5	DE	29.8	1.1	460.8
9	74.3	FL	27.2	1.2	350.6
10	72.2	GA	28.3	1.1	168.4

set before looking at the data

$\text{model}_1: \quad Y_i = 75 + \text{ERROR}$

worth it?

fit to the data

$\text{model}_2: \quad Y_i = \beta_0 + \text{ERROR}$

worth it?

additional predictor

$\text{model}_3: \quad Y_i = \beta_0 + \beta_1 X_i + \text{ERROR}$

fit to the data

Compact model

model_C: $Y_i = \beta_0 + \text{ERROR}$

Augmented model

model_A: $Y_i = \beta_0 + \beta_1 X_i + \text{ERROR}$

$$\text{ERROR}(C) \geq \text{ERROR}(A)$$

Proportional reduction in error (PRE)

$$\text{PRE} = \frac{\text{ERROR}(C) - \text{ERROR}(A)}{\text{ERROR}(C)}$$

Compact model

model_C: $Y_i = \beta_0 + \text{ERROR}$

$$\text{ERROR}(C) = 50$$

Augmented model

model_A: $Y_i = \beta_0 + \beta_1 X_i + \text{ERROR}$

$$\text{ERROR}(A) = 30$$

Proportional reduction in error (PRE)

$$\text{PRE} = 1 - \frac{\text{ERROR}(A)}{\text{ERROR}(C)}$$

$$= 1 - \frac{30}{50} = .40$$

Increasing the complexity of the model reduced the error by 40%. **worth it?**

worth it?

Compact model

model_C: $Y_i = \beta_0 + \text{ERROR}$

Augmented model

model_A: $Y_i = \beta_0 + \beta_1 X_i + \text{ERROR}$

Proportional reduction in error (PRE)

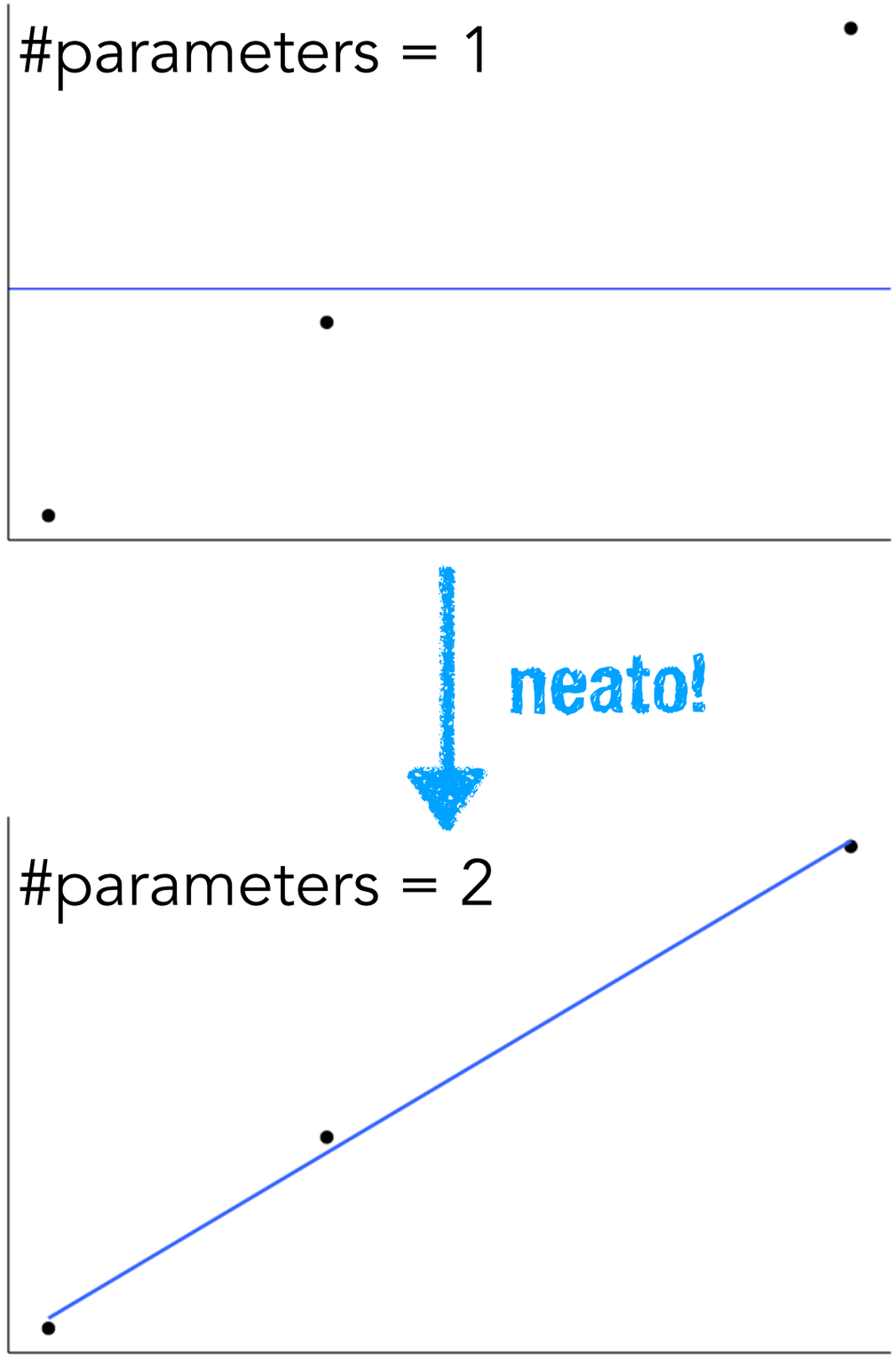
$$\text{PRE} = 1 - \frac{\text{ERROR}(A)}{\text{ERROR}(C)}$$

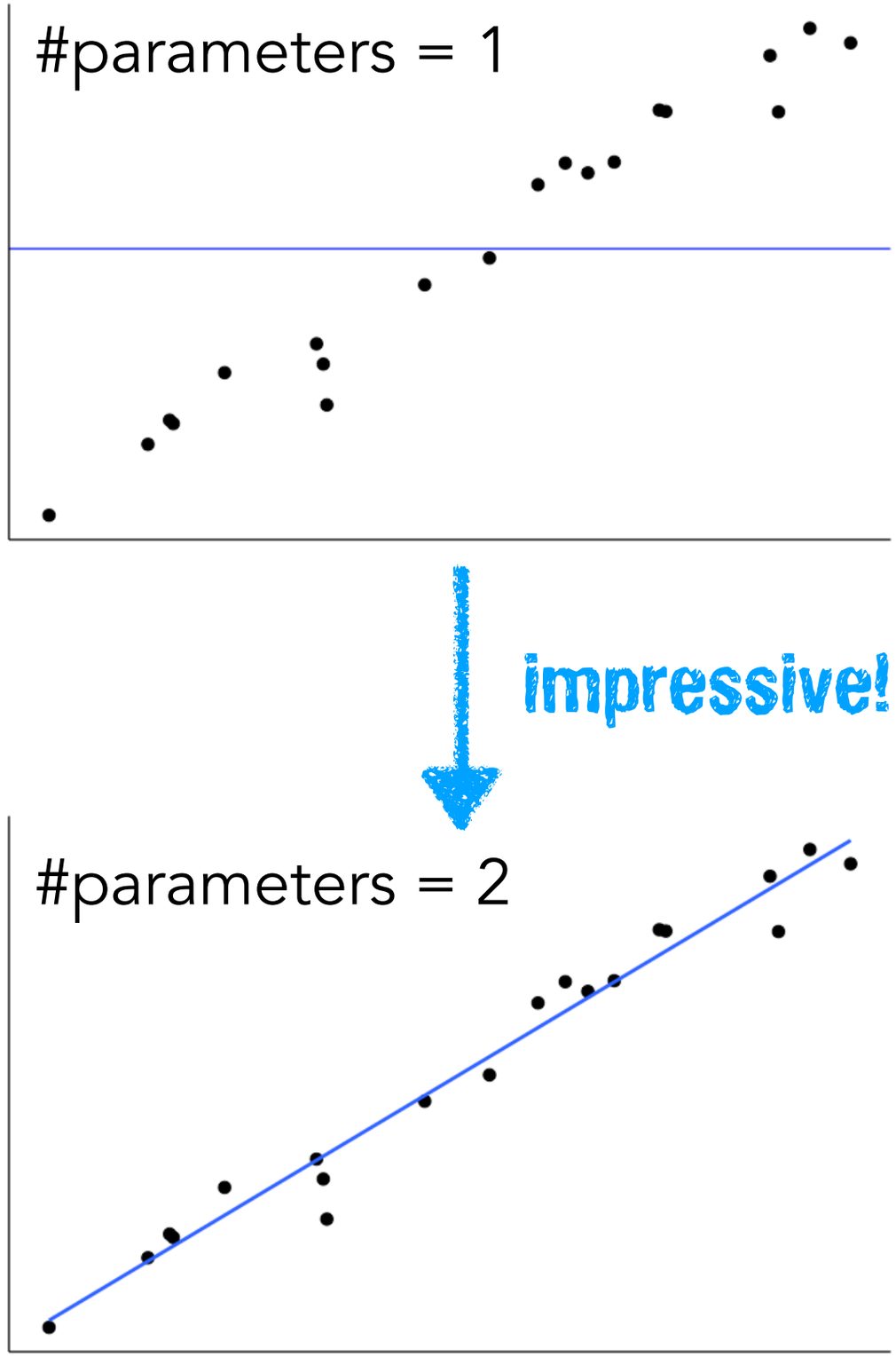
$$= 1 - \frac{30}{50} = .40$$

- to answer the **worth it?** question we need inferential statistics (define ERROR, sampling distributions, etc.)
- more likely to be **worth it** if:
  1. **PRE** is high
  2. the number of additional parameters in A compared to C is low (**PRE per additional parameter**)
  3. the number of parameters that could have been added to model_C to create model_A but were not is high

more impressed if the number of observations n is much greater than the number of parameters

PRE per parameter for different *n*





General procedure

- for any question we want to ask about our DATA
  - we define $model_C$ and $model_A$
  - compare the models using PRE
  - determine whether PRE is **worth it**
- in standard frequentist lingo:
  - $model_C = H_0$ (null hypothesis)
  - $model_A = H_1$ (alternative hypothesis)

model comparison

- hypothesis test:
  - H_0: **all** the parameters that are included in $model_A$ but not in $model_C$ are 0
  - H_1: **not all** the parameters that are included in $model_A$ but not in $model_C$ are 0

Notation

DATA = MODEL + ERROR

$$Y_i = \beta_0 + \epsilon_i$$ simple model (true parameters)

$$Y_i = b_0 + e_i$$ simple model (estimated parameters)

$$\hat{Y}_i = b_0$$

college

$$Y_i = b_0 + b_1 X_{i1} + e_i$$ more complex model

Percentage of households that had internet access in the year 2013 by US state

i	internet	state	college	auto	density
1	79.0	AK	28.0	1.2	1.2
2	63.5	AL	23.5	1.3	94.4
3	60.9	AR	20.6	1.7	56.0
4	73.9	AZ	27.4	1.3	56.3
5	77.9	CA	31.0	0.8	239.1
6	79.4	CO	37.8	1.0	48.5
7	77.5	CT	37.2	1.0	738.1
8	74.5	DE	29.8	1.1	460.8
9	74.3	FL	27.2	1.2	350.6
10	72.2	GA	28.3	1.1	168.4

Greek letters β or ε represent the true but unknowable parameters in the population.

Roman letters ***b*** or ***e*** represent estimates of these parameters using our DATA.

Definitions of error and parameter estimates

Definitions of error and parameter estimates

1. How should individual errors be aggregated into a summary index ERROR?
   - sum of absolute errors
   - sum of squared errors
   - count of errors
2. What's the best estimator of the data for each kind of error?
3. Which error shall we choose?

Simple model

$$Y_i = \beta_0 + \epsilon_i$$

$$Y_i = b_0 + e_i$$

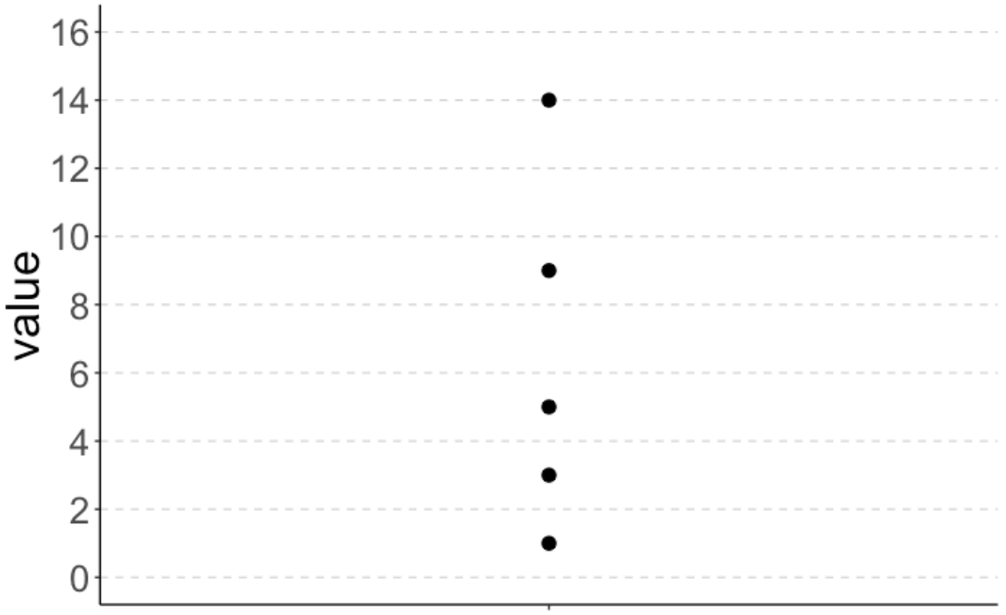
depends on how the error is defined!

what value is the best model of the data?

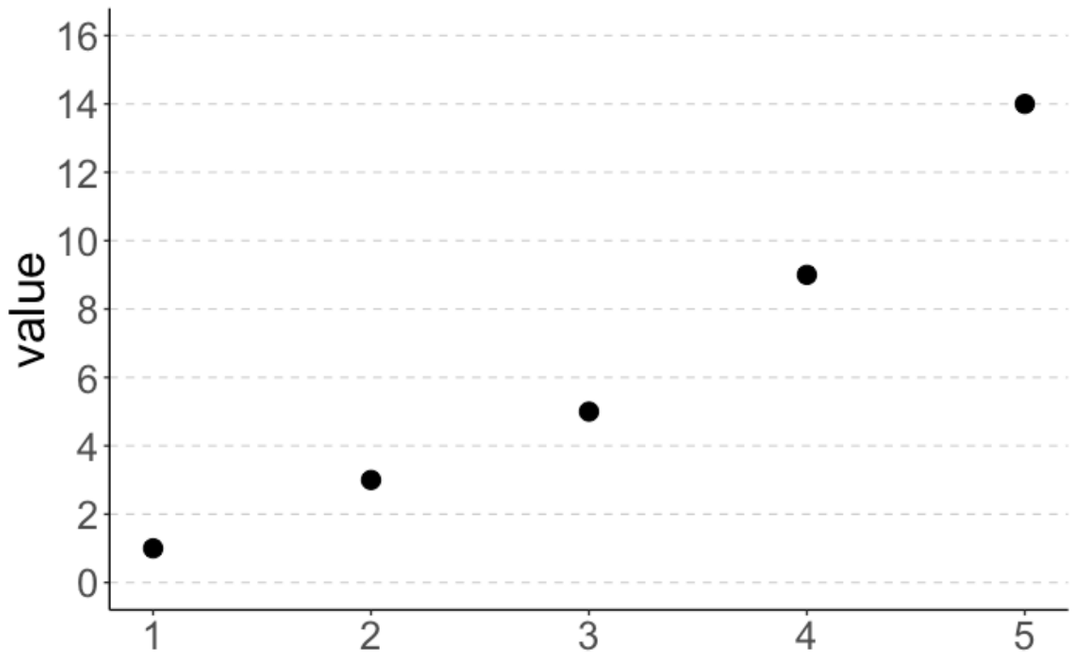
DATA

Error = sum of absolute errors

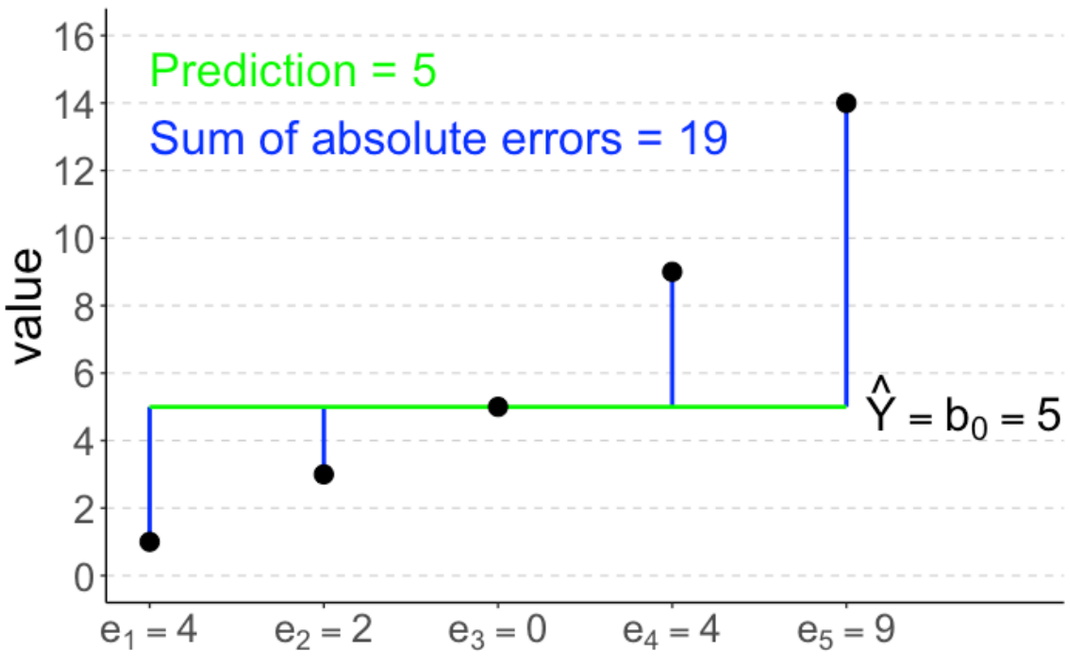
Model

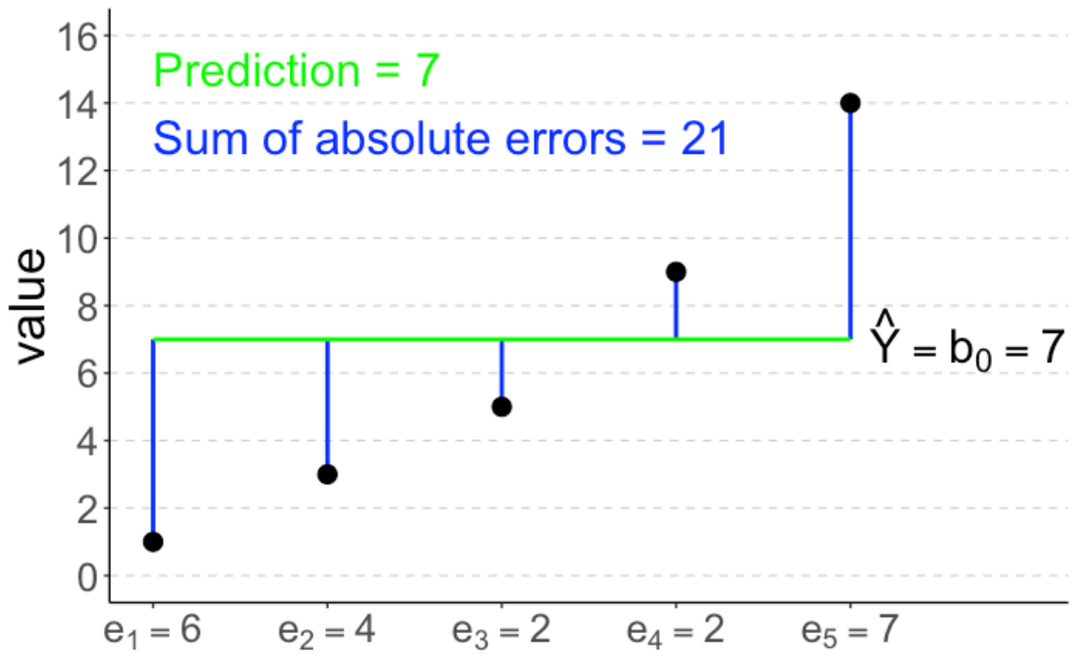


`c(1, 3, 5, 9, 14)`

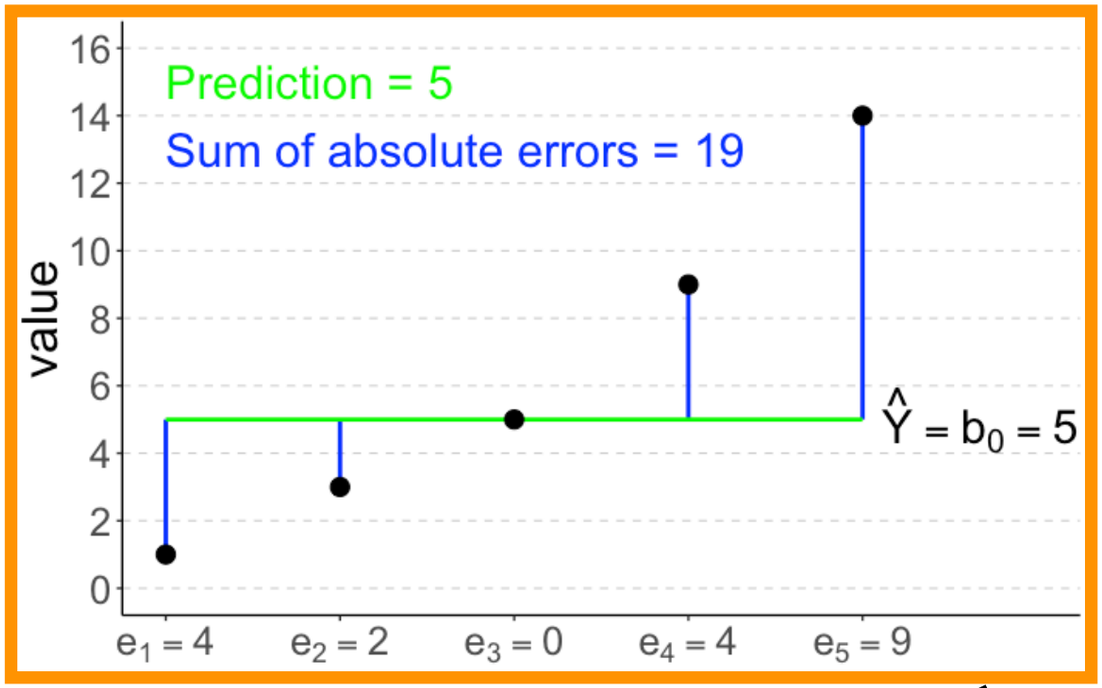


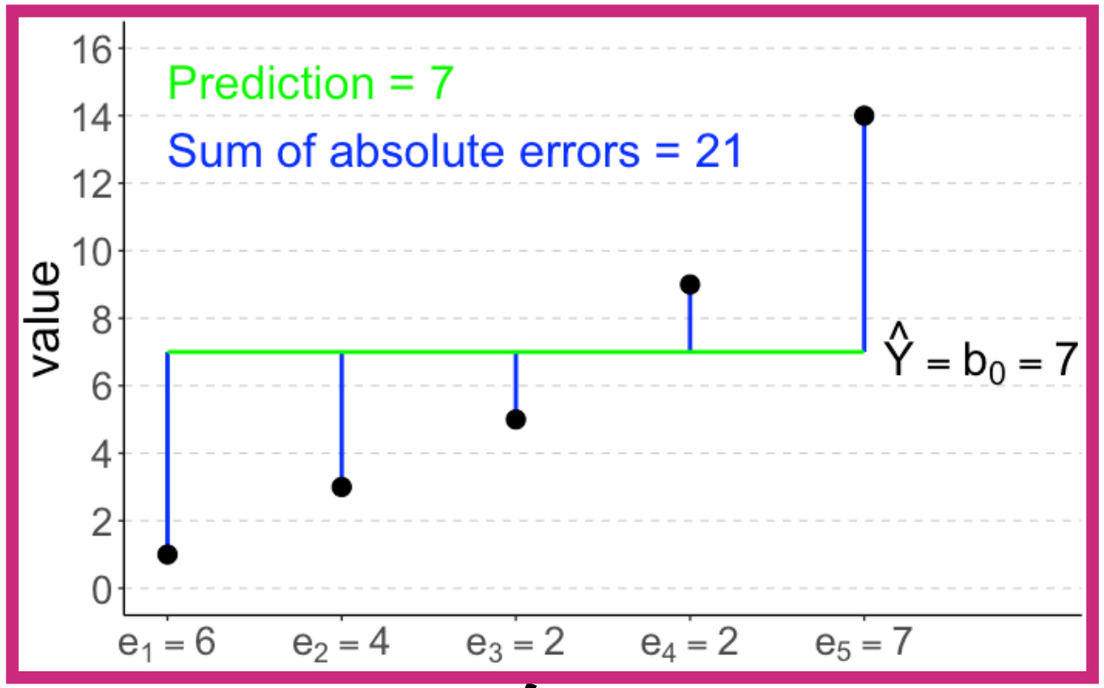
Error as the sum of **line lengths**

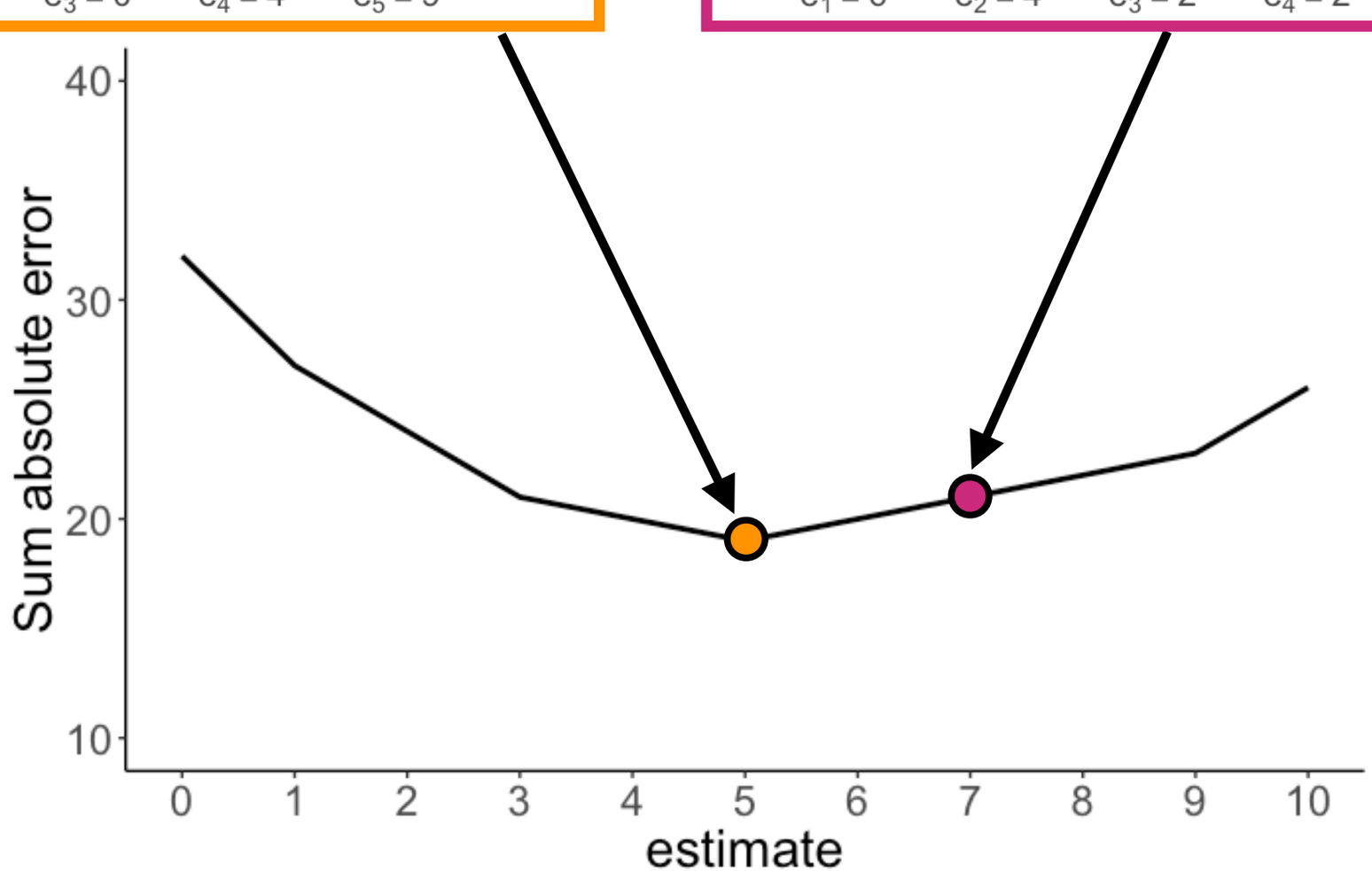




What's the best simple model b_0 for this error measure?

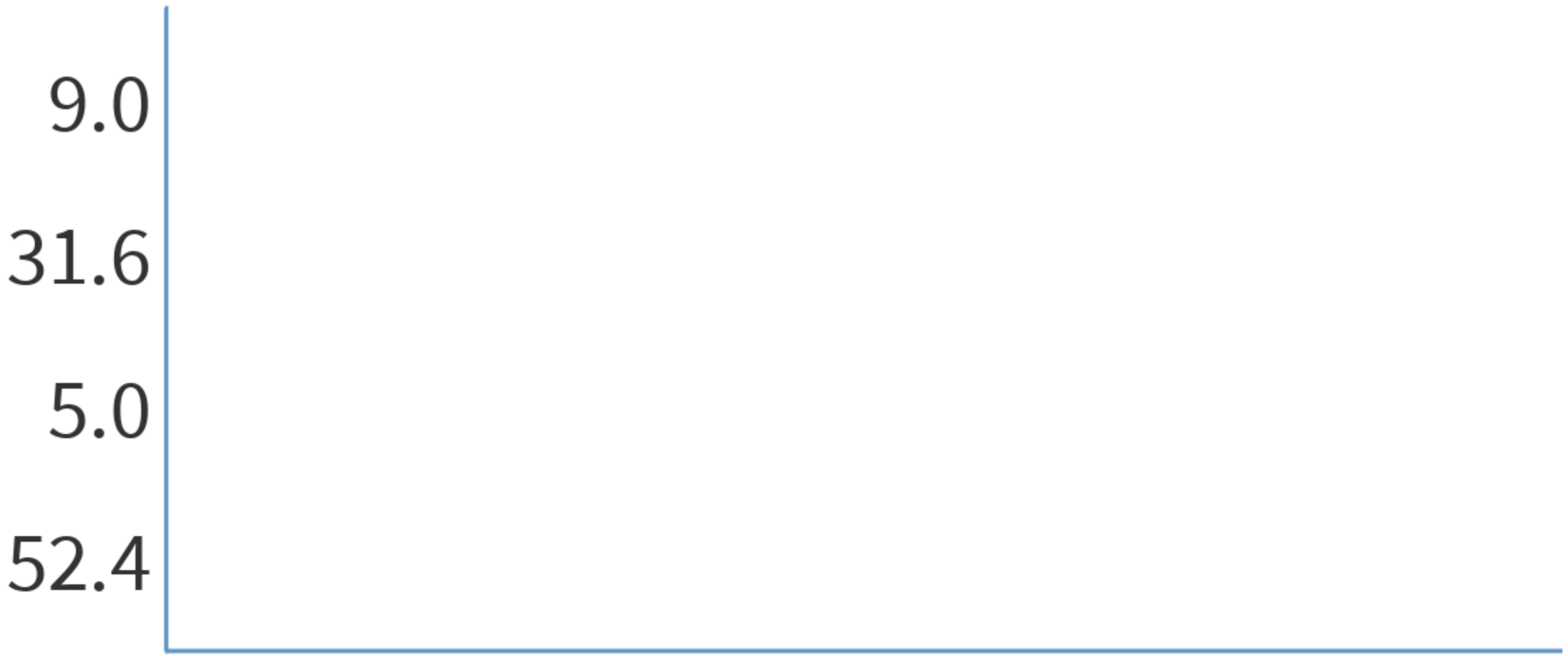






What if the blue value was 140 instead of 14?
What would be the best estimate of b_0 then?

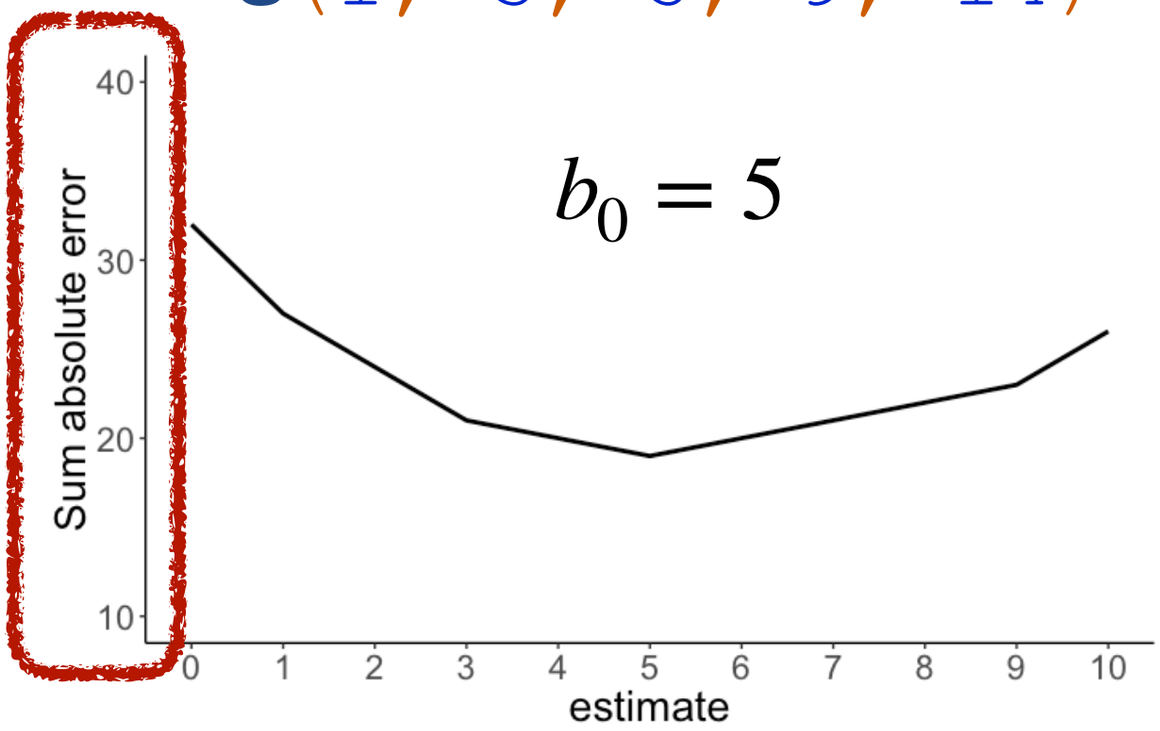
What's the estimate that minimizes the sum of absolute errors for the values 1, 3, 5, 9, 140?



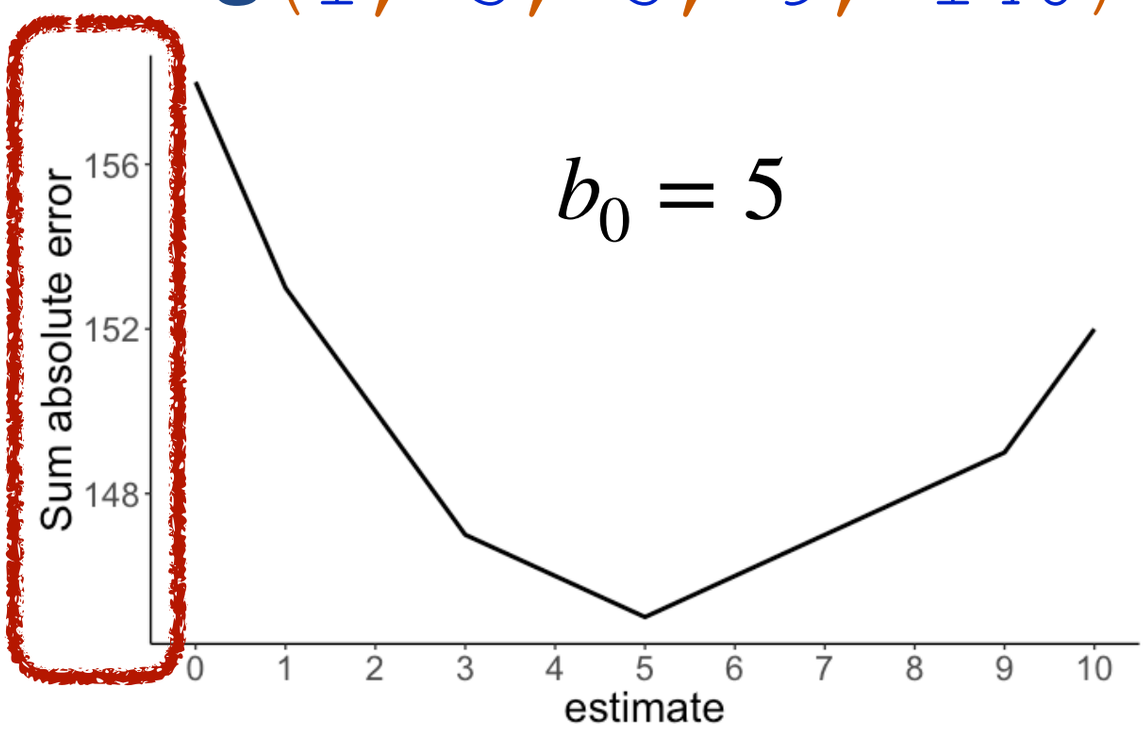
Sum of absolute errors

$$Y_i = b_0 + e_i \qquad \text{ERROR} = \sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |Y_i - b_0|$$

`c(1, 3, 5, 9, 14)`



`c(1, 3, 5, 9, 140)`



the **median** minimizes the sum of absolute errors

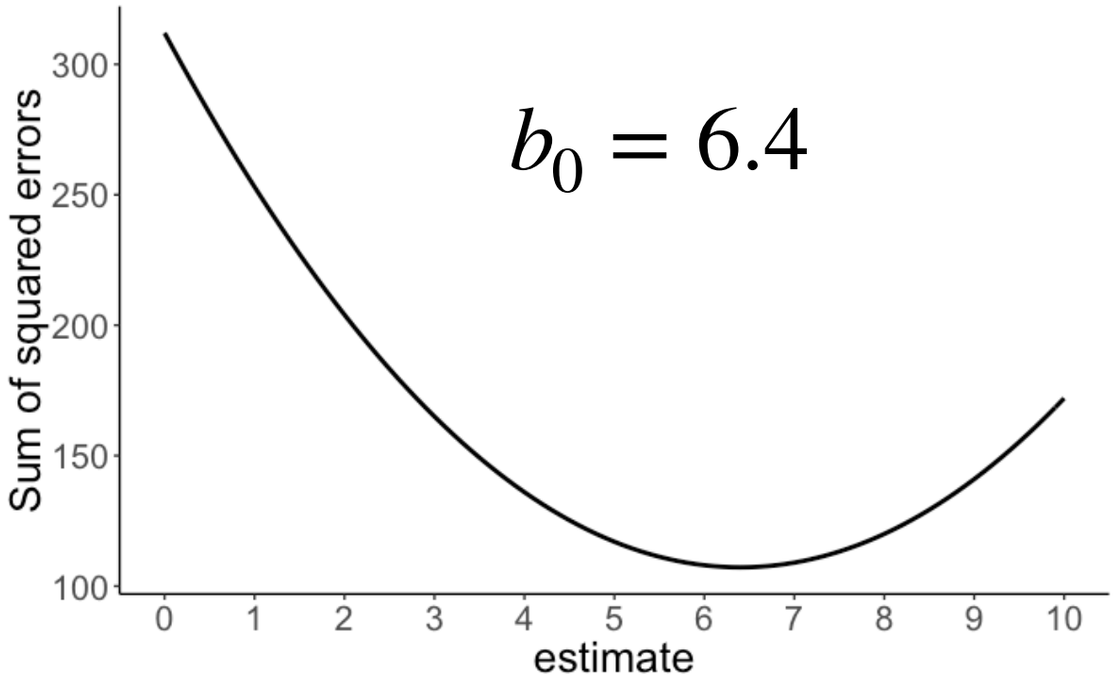
is robust to outliers!

Error = sum of squared errors

Sum of squared errors

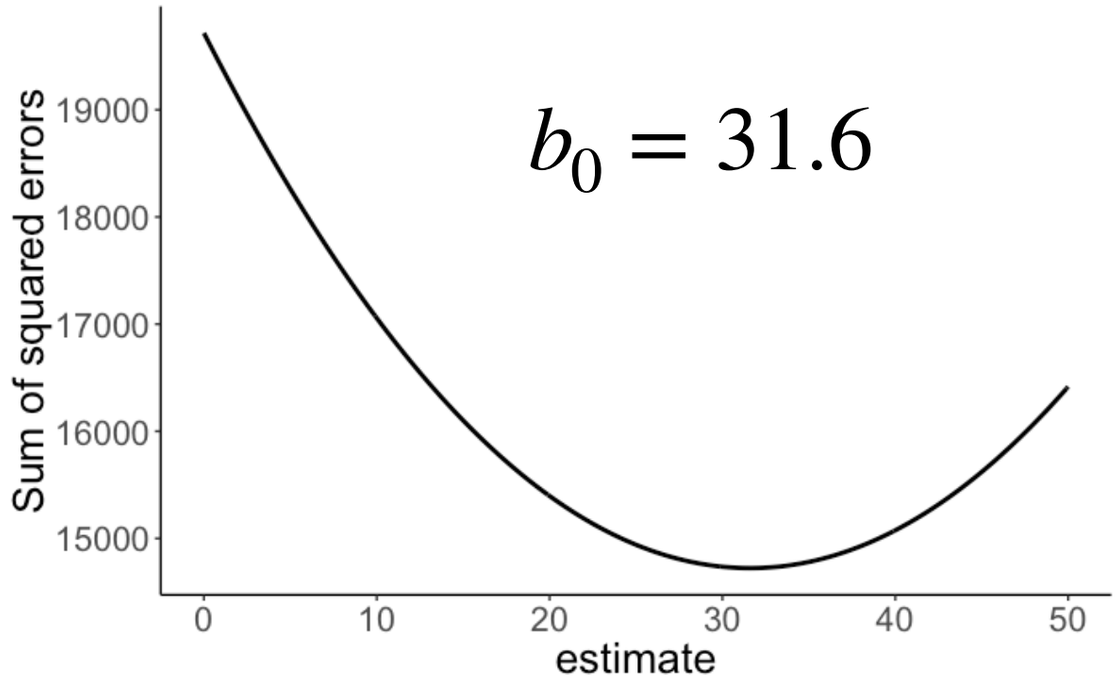
$$Y_i = b_0 + e_i \qquad \text{ERROR} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Y_i - b_0)^2$$

```
c(1, 3, 5, 9, 14)
```



$$b_0 = 6.4$$

```
c(1, 3, 5, 9, 140)
```



$$b_0 = 31.6$$

the **mean** minimizes the sum of squared errors

is strongly affected by outliers!

Error = count of errors

$$Y_i = b_0 + e_i \qquad \text{ERROR} = \sum_{i=1}^{n} I(e_i) = \sum_{i=1}^{n} I(Y_i - b_0)$$

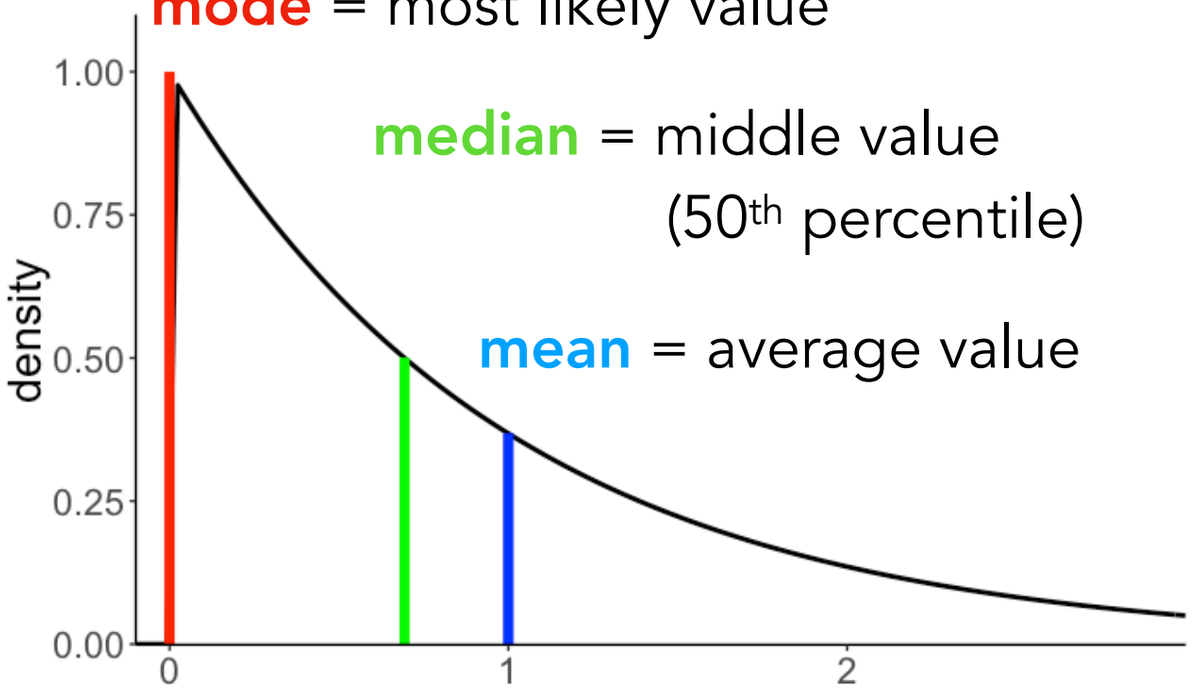
the **mode** minimizes the count of errors

Quick recap

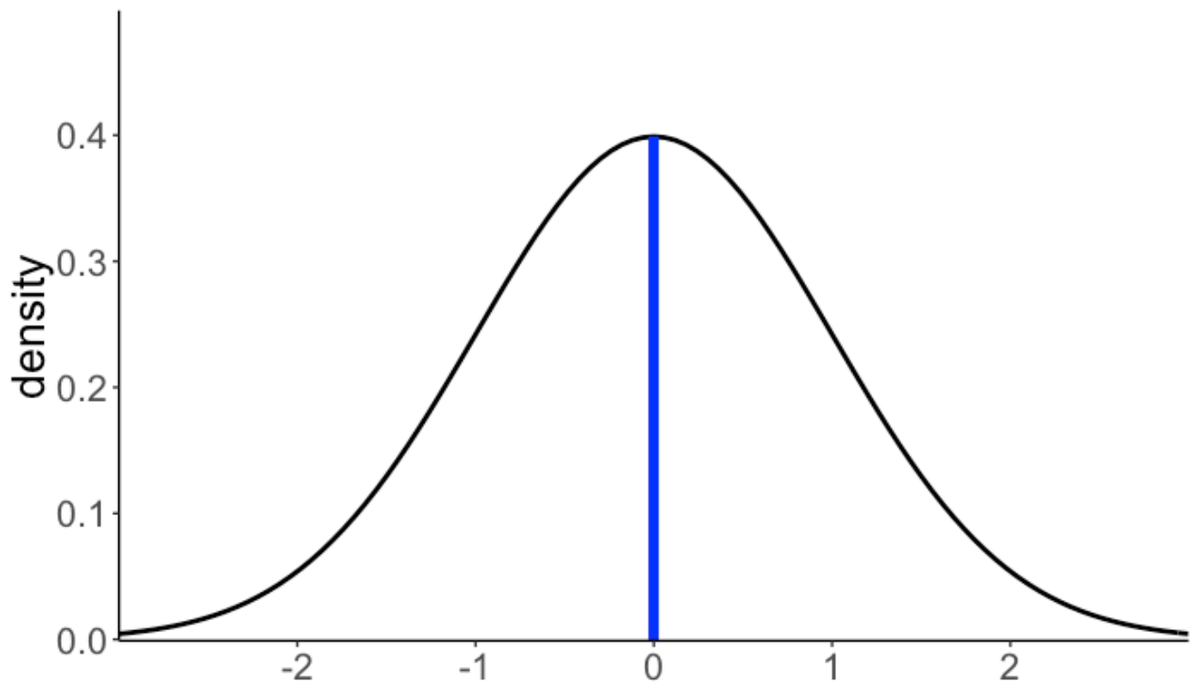
mode = most likely value

median = middle value (50th percentile)

mean = average value



exponential distribution



normal distribution

Error definition	Best estimator
Count of errors	Mode = most frequent value
Sum of absolute errors	Median = middle observation of all values
Sum of squared errors	Mean = average of all values

Models of error

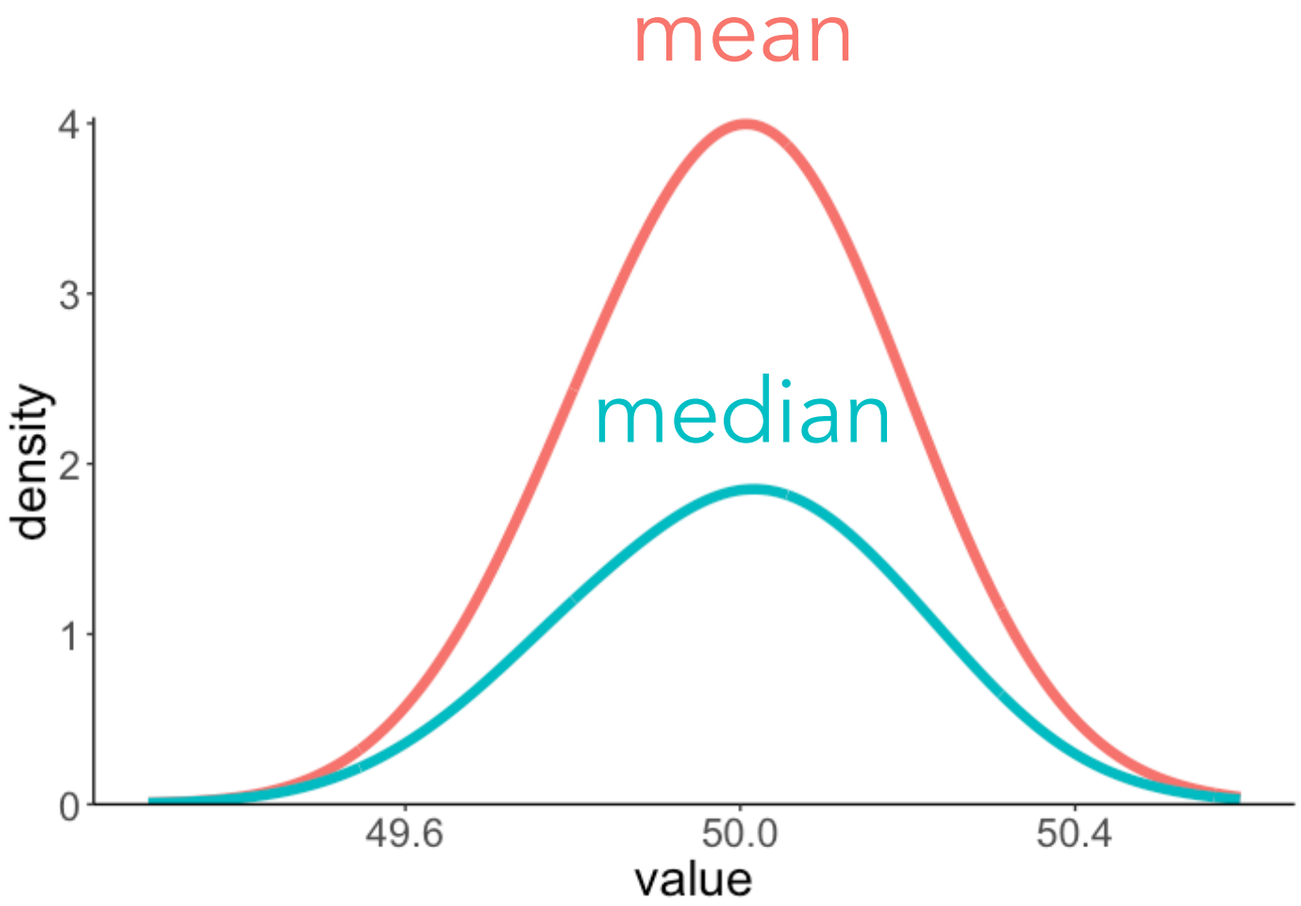
which model for error shall we choose?

Sampling distributions

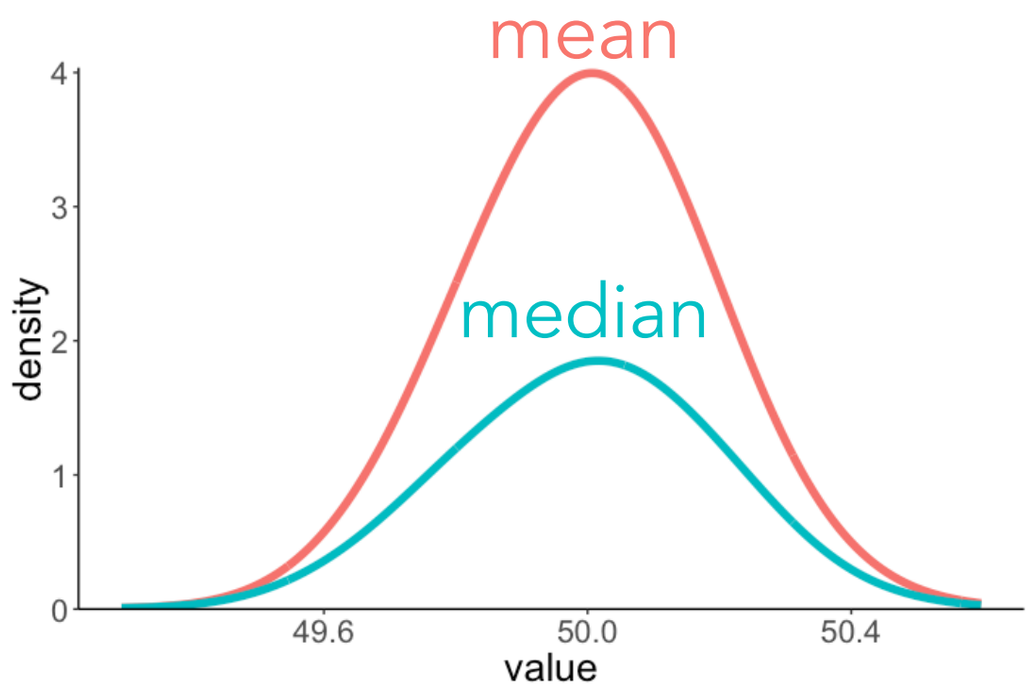
$$Y_i = 50 + \epsilon$$ **the true model**

Recipe

- take *m* samples of size *n*
- for each sample, calculate the `mean()` and `median()`
- plot the distribution (histogram, or density)



Properties of estimators



- **Unbiasedness**
  - does the average value of distribution match the true value?
- **Efficiency**
  - how precisely does the estimator capture the true value for a given sample size?
- **Consistency**
  - how does the estimators precision change as the sample size increases?

$$Y_i = 50 + \epsilon$$

but how was the error generated in the true model?

$$\epsilon \sim \mathcal{N}(\mu = 0, \sigma)$$

I assumed normally distributed errors!

justification?

The central limit theorem!

the distribution of the sum of individual error components will approximate a normal distribution

https://seeing-theory.brown.edu/probability-distributions/index.html#section3

Central limit theorem

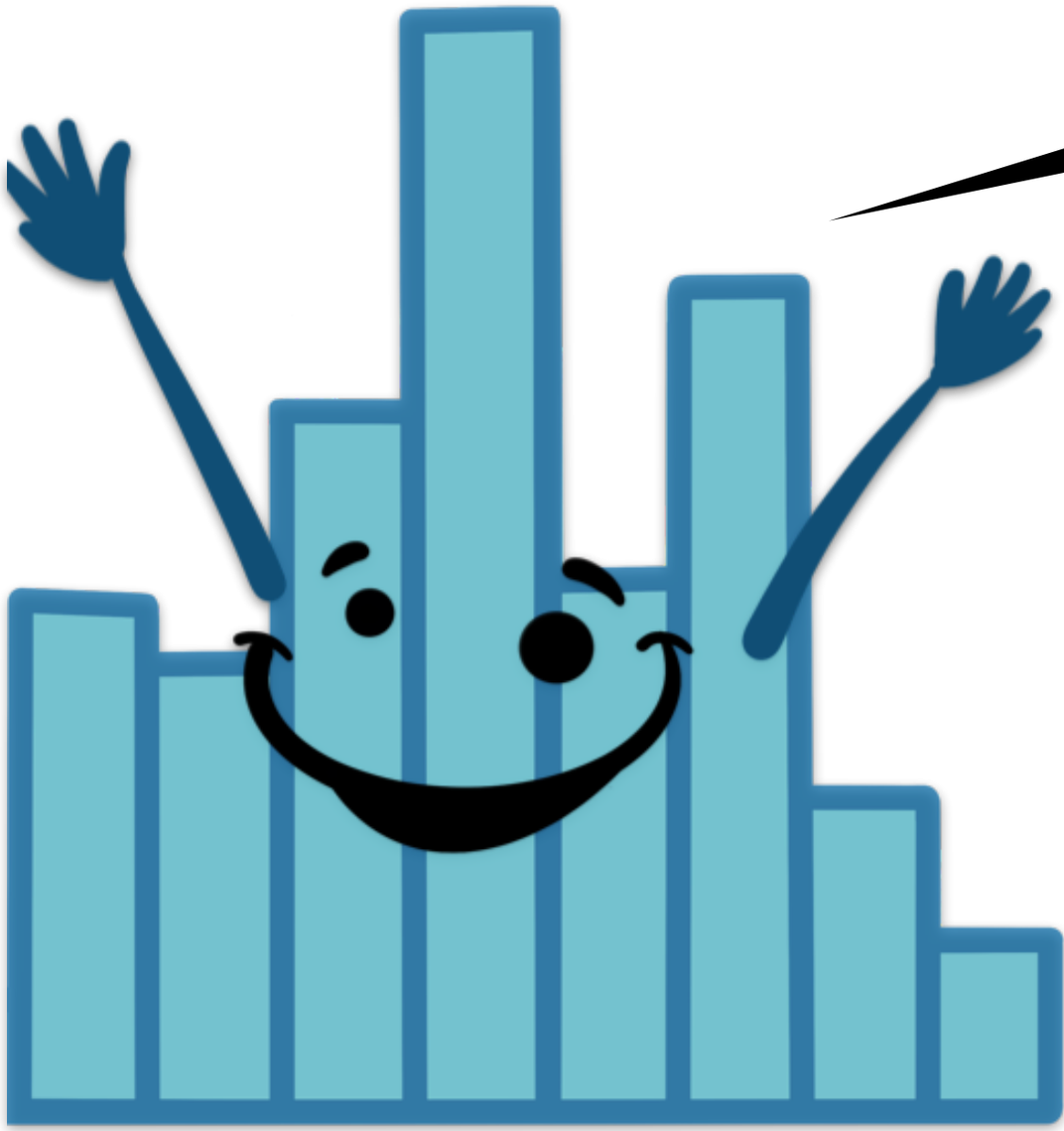
Quick recap

$$Y_i = \beta_0 + \epsilon_i \qquad Y_i = b_0 + e_i$$

sum of squared errors

mean

- Central limit theorem suggests that (very often) *errors are normally distributed*
- the mean is the *most efficient* (and unbiased) estimator when errors are normally distributed
- the mean minimizes the *sum of squared errors*



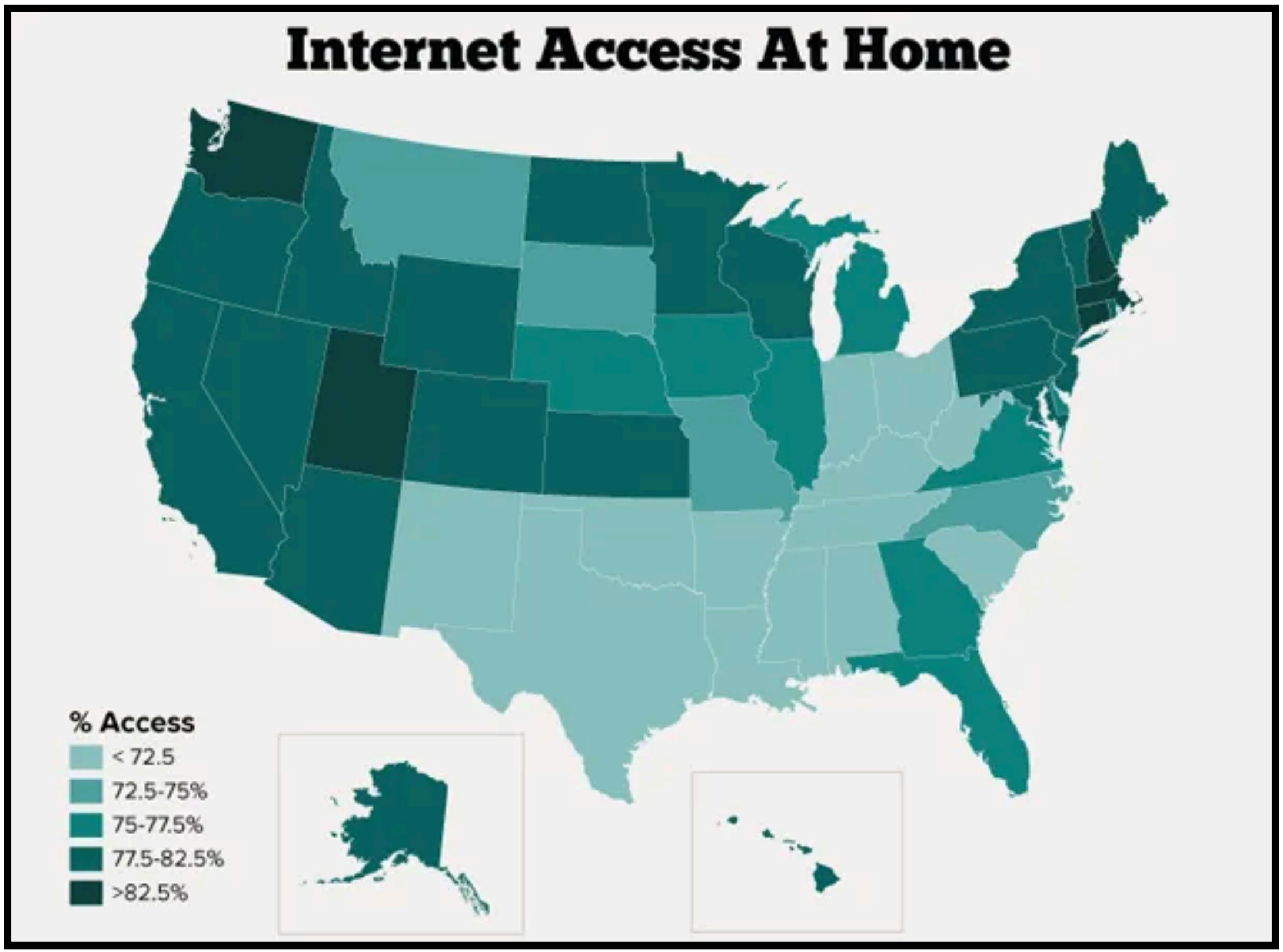
01:00

stretch break!

Statistical inferences about parameter values

Procedure

1. Start with research question
2. Formulate hypothesis as a comparison between compact and augmented model
3. Fit parameters in each model
4. Calculate the proportional reduction of error (PRE)
5. Decide whether PRE is **worth it**



http://thedailyviz.com/2012/07/19/mapping-internet-access-in-u-s-homes/

Research question and hypotheses

Is the average percentage of internet users per state significantly different from 75%?

Model$_C$: $Y_i = B_0 + \epsilon_i$

0 parameters

$Y_i = 75 + e_i$

Model$_A$: $Y_i = \beta_0 + \epsilon_i$

1 parameter

$Y_i = b_0 + e_i$

$= \overline{Y} + e_i$

i	internet	state	college	auto	density
1	79.0	AK	28.0	1.2	1.2
2	63.5	AL	23.5	1.3	94.4
3	60.9	AR	20.6	1.7	56.0
4	73.9	AZ	27.4	1.3	56.3
5	77.9	CA	31.0	0.8	239.1
6	79.4	CO	37.8	1.0	48.5
7	77.5	CT	37.2	1.0	738.1
8	74.5	DE	29.8	1.1	460.8
9	74.3	FL	27.2	1.2	350.6
10	72.2	GA	28.3	1.1	168.4

Fit parameters and calculate PRE

C: $Y_i = 75 + e_i$ **A**: $Y_i = \overline{Y} + e_i$

i	state	internet	compact_b	compact_se	augmented_b	augmented_se
1	AK	79.0	75	16.00	72.81	38.37
2	AL	63.5	75	132.25	72.81	86.60
3	AR	60.9	75	198.81	72.81	141.75
4	AZ	73.9	75	1.21	72.81	1.20
5	CA	77.9	75	8.41	72.81	25.95
6	CO	79.4	75	19.36	72.81	43.48
7	CT	77.5	75	6.25	72.81	22.03
8	DE	74.5	75	0.25	72.81	2.87
9	FL	74.3	75	0.49	72.81	2.23
10	GA	72.2	75	7.84	72.81	0.37

$$\text{PRE} = 1 - \frac{\text{ERROR}(A)}{\text{ERROR}(C)}$$

$$= 1 - \frac{1355}{1595} \approx .15$$

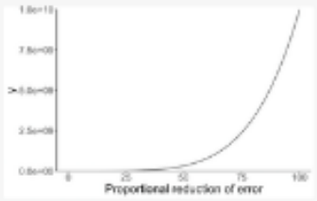
Model A has 15% less error than Model C.

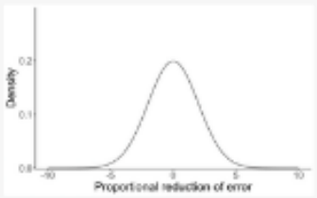
SSE(C) = 1595 SSE(A) = 1355

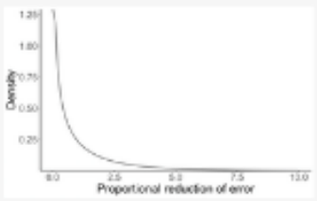
Decide whether it's **worth it**

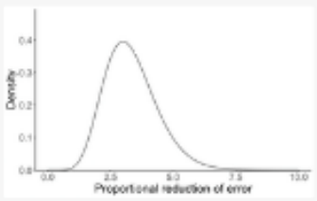
- PRE is the estimate of an unknown true reduction of error η^2
- we need a sampling distribution of PRE
  - a distribution of what PRE would look like if Model C (our H_0) were true
  - we could just simulate such a sampling distribution ...

What do you expect the sampling distribution of PRE to look like?



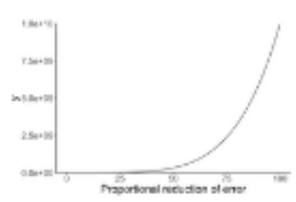


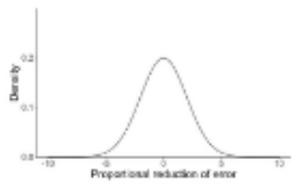


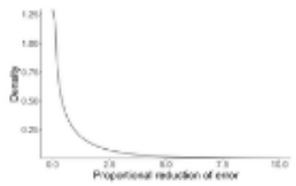


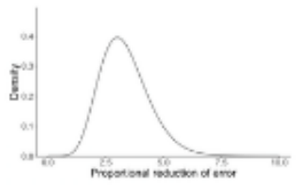
Total Results: 0

What do you expect the sampling distribution of PRE to look like?









Decide whether it's **worth it**

- PRE is the estimate of an unknown true reduction of error η^2
- we need a sampling distribution of PRE
  - a distribution of what PRE would look like if Model C (our H_0) were true
  - we could just simulate such a sampling distribution ...
- PRE is closely related to the *F* statistic!

Decide whether it's **worth it**

- To compute the *F* statistic, we need:
  - PRE
  - number of parameters in Model C (PC) and Model A (PA)
  - number of observations *n*

- more likely to be **worth it** if:
  1. **PRE** is high
  2. the number of additional parameters in A compared to C is low (**PRE per additional parameter**)
  3. the number of parameters that could have been added to $model_C$ to create $model_A$ but were not

difference in parameters between models A and C

$$F = \frac{\text{PRE}/(\text{PA} - \text{PC})}{(1 - \text{PRE})/(n - \text{PA})}$$

number of observations vs. parameters in Model A

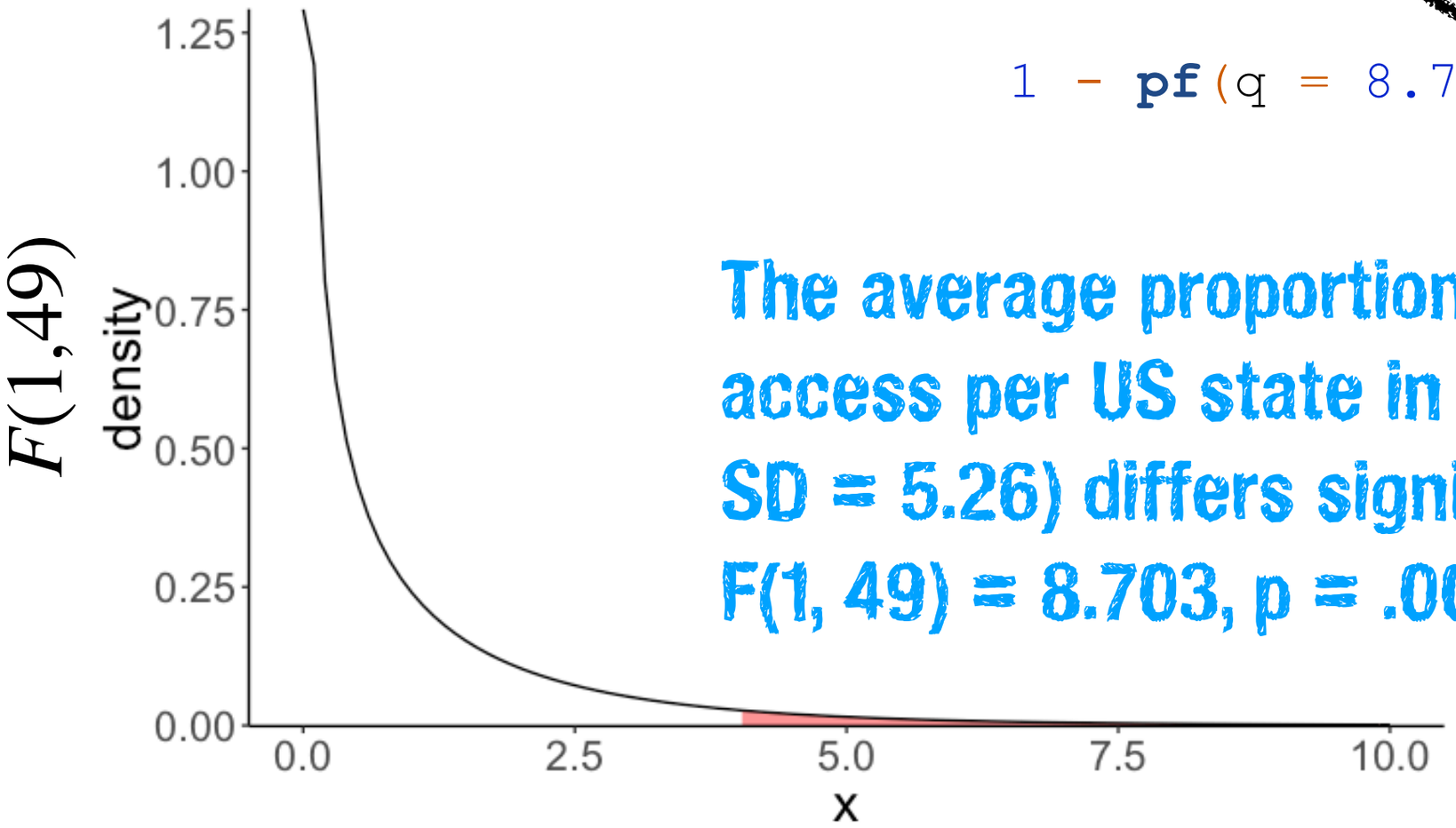
Decide whether it's worth it

Note: I've used the approximated PRE here (PRE = 0.15), but I'm reporting the F value for the non-approximated PRE value.

$$F = \frac{\text{PRE}/(\text{PA} - \text{PC})}{(1 - \text{PRE})/(n - \text{PA})}$$

$$= \frac{.15/(1 - 0)}{(1 - .15)/(50 - 1)} = 8.703 \quad p = .00486$$

```
1 - pf(q = 8.703, df1 = 1, df2 = 49)
```



The average proportion of internet access per US state in 2003 (M = 72.8, SD = 5.26) differs significantly from 75%, F(1, 49) = 8.703, p = .005.

we just performed a one sample t-test ...

```
t.test(df.internet$internet, mu = 75)
```

One Sample t-test

data: df.internet$internet
t = -2.9502, df = 49, p-value = **0.00486**
alternative hypothesis: true mean is not equal to 75

but ...

- we will apply the general procedure we went through to day to a large range of different situations
- thinking of hypothesis testing as model comparison is (hopefully more) intuitive
- this approach still works even if we can't find a recipe in our favorite stats cook book

Generating a sampling distribution for PRE

Decide whether it's **worth it**

- we have to construct a sampling distribution of PRE assuming that H_0 is true
- and then compare the observed value of PRE to that distribution

Population distribution

$$Y_i = 75 + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}(\mu = 0, \sigma = 5)$$

Model C

$$Y_i = 75 + e_i$$

0 parameters

Model A

$$Y_i = \overline{Y} + e_i$$

1 parameter

Sampling distribution of PRE

```r
# simulation parameters
n_samples = 1000
sample_size = 50
mu = 75 # true mean of the distribution
sigma = 5 # true standard deviation of the errors

# function to draw samples from the population distribution
fun.draw_sample = function(sample_size, mu, sigma){
  sample = mu + rnorm(sample_size, mean = 0, sd = sigma)
}

# draw samples
samples = n_samples %>%
  replicate(fun.draw_sample(sample_size, mu, sigma)) %>%
  t() # transpose the resulting matrix (i.e. flip rows and columns)
```

sample	index	number
1	1	75.30
1	2	72.06
1	3	77.66
1	4	67.41
1	5	76.53
1	6	67.32
1	7	73.50
1	8	72.36
1	9	71.74
1	10	74.72
⋮		

Sampling distribution of PRE

```
# put samples in data frame and compute PRE
df.samples = samples %>%
  as_tibble(.name_repair = ~ 1:ncol(samples)) %>%
  mutate(sample = 1:n()) %>%
  pivot_longer(cols = -sample,
               names_to = "index",
               values_to = "value") %>%
  mutate(compact = 75) %>%
  group_by(sample) %>%
  mutate(augmented = mean(value))
```

sample	index	value	compact	augmented
1	1	73.43	75	74.75
1	2	76.38	75	74.75
1	3	79.92	75	74.75
1	4	72.33	75	74.75
1	5	77.75	75	74.75
2	1	79.84	75	73.92
2	2	78.44	75	73.92
2	3	79.49	75	73.92
2	4	71.81	75	73.92
2	5	79.57	75	73.92
3	1	78.99	75	74.93
3	2	67.28	75	74.93
3	3	77.74	75	74.93
3	4	73.73	75	74.93
3	5	73.49	75	74.93

Sampling distribution of PRE

```
# put samples in data frame and compute PRE
df.samples = samples %>%
  as_tibble(.name_repair = ~ 1:ncol(samples)) %>%
  mutate(sample = 1:n()) %>%
  pivot_longer(cols = -sample,
               names_to = "index",
               values_to = "value") %>%
  mutate(compact = 75) %>%
  group_by(sample) %>%
  mutate(augmented = mean(value)) %>%
  summarize(sse_compact = sum((value - compact)^2),
            sse_augmented = sum((value - augmented)^2),
            pre = 1 - sse_augmented/sse_compact)
```

calculate SSE for each model

calculate PRE

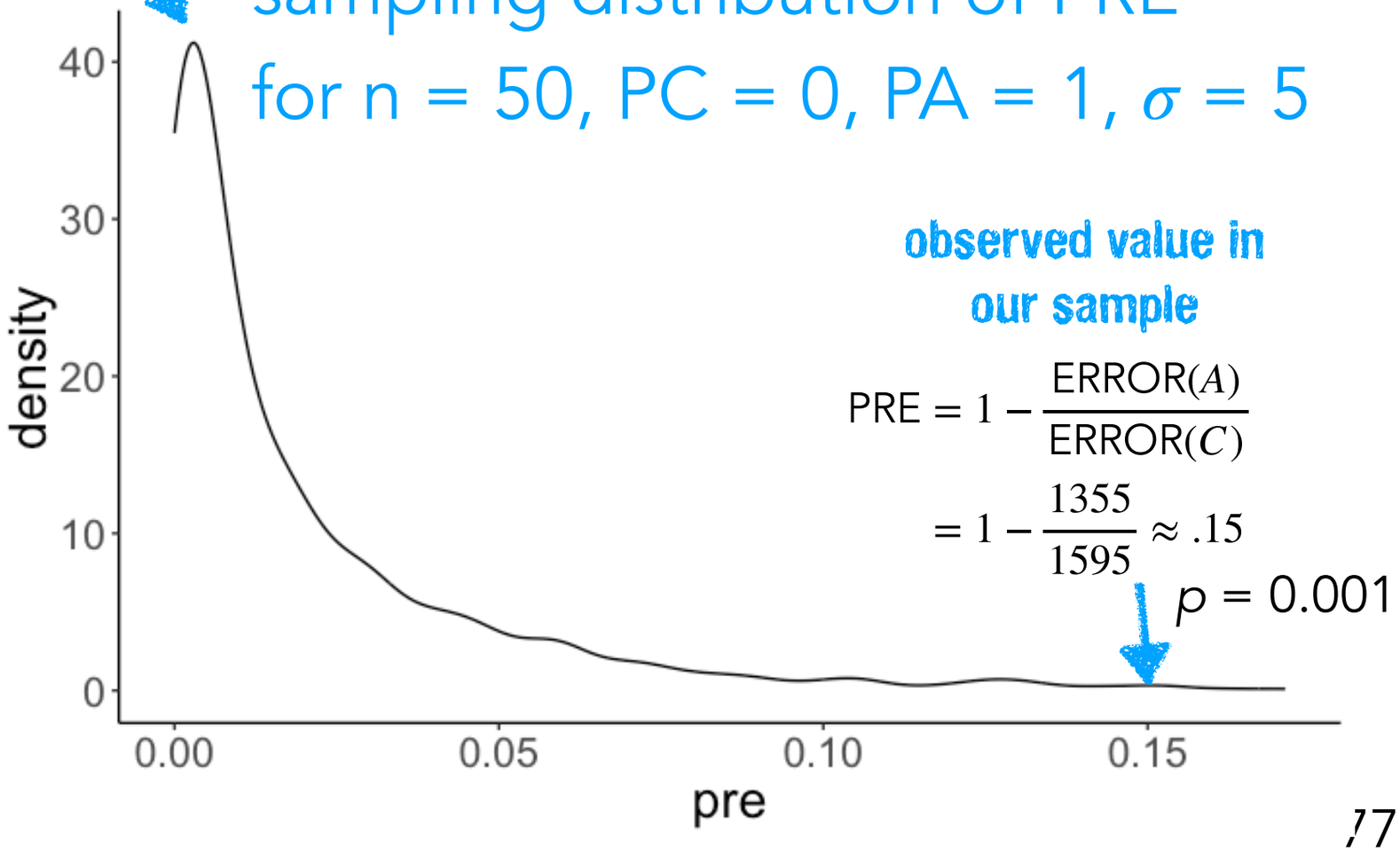
sample	sse_compact	sse_augmented	pre
1	1244.66	1241.52	0.00
2	953.25	894.95	0.06
3	1083.60	1083.32	0.00
4	888.06	798.19	0.10
5	1246.57	1233.25	0.01

Sampling distribution of PRE

```
29 # sampling distribution for PRE
30 ggplot(data = df.samples,
31        mapping = aes(x = pre)) +
32   stat_density(geom = "line")
33
34 # p-value for our sample
35 df.samples %>%
36   summarize(p_value = sum(pre >= df.summary$pre)/n())
```

sample	sse_compact	sse_augmented	pre
1	1244.66	1241.52	0.00
2	953.25	894.95	0.06
3	1083.60	1083.32	0.00
4	888.06	798.19	0.10
5	1246.57	1233.25	0.01

sampling distribution of PRE
for n = 50, PC = 0, PA = 1, $\sigma = 5$

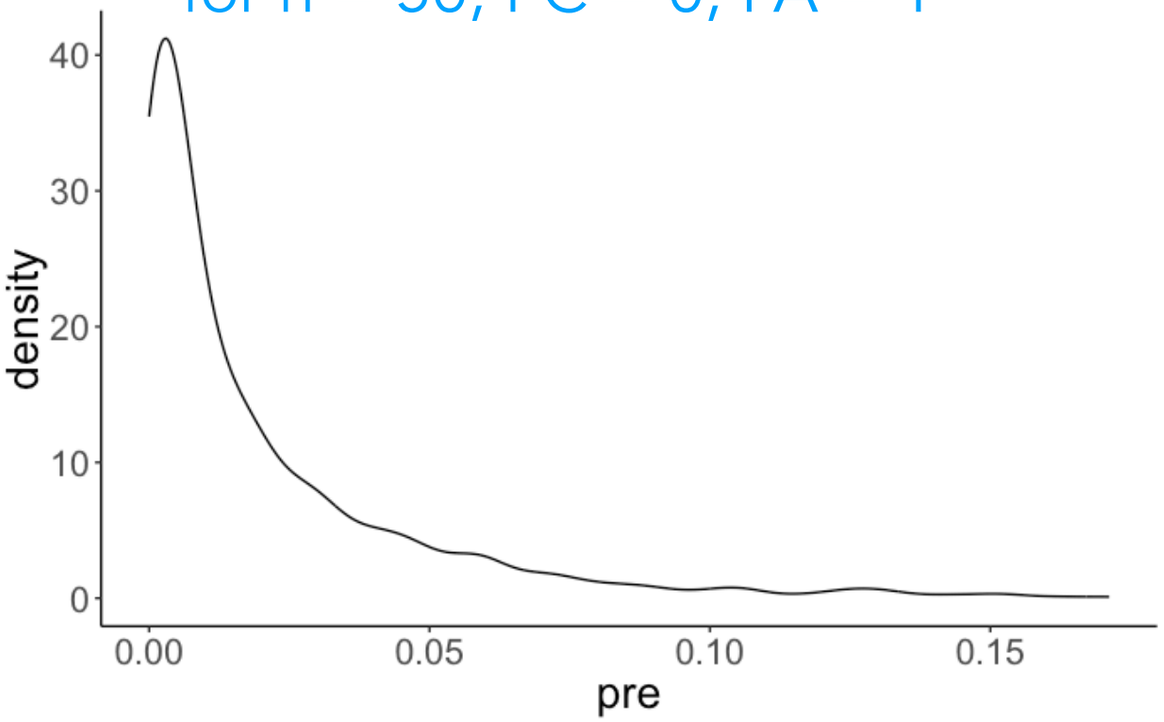


Sampling distribution of PRE

deterministic mapping

sampling distribution of PRE

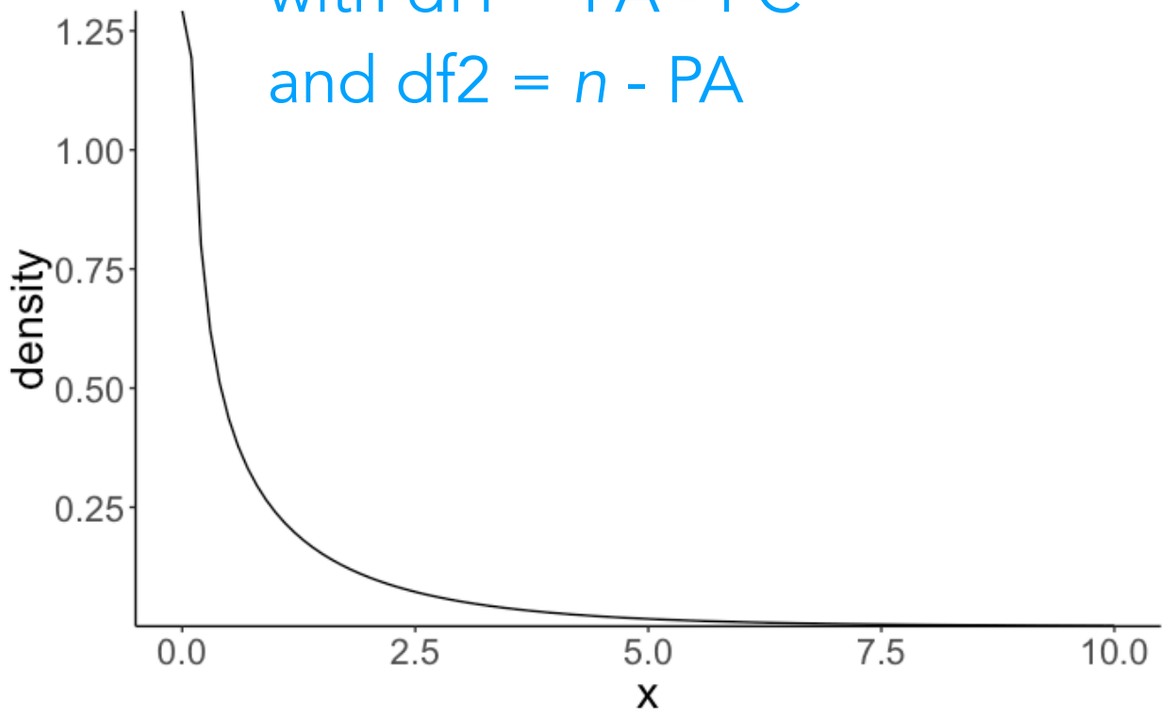
for n = 50, PC = 0, PA = 1



***F*(df1, df2) distribution**

with df1 = PA - PC

and df2 = *n* - PA



we use the F-distribution since it comes with R (and is the standard statistic to report)

The general procedure

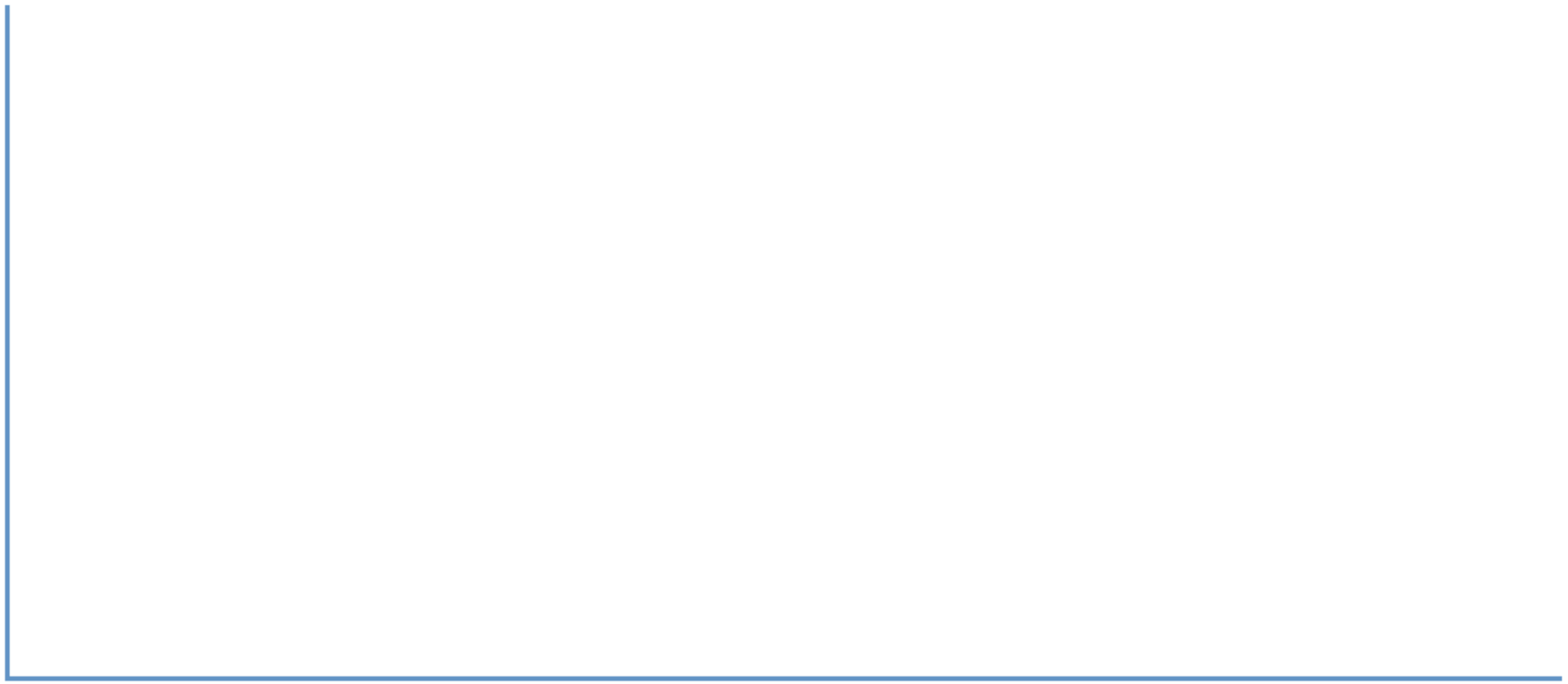
1. Define H_0 as Model C (compact) and H_1 as Model A (augmented)
2. Fit model parameters to the data
3. Calculate the proportional reduction of error (PRE) in our sample
4. Decide whether the augmented model is **worth it** by comparing the observed PRE in our sample to the sampling distribution of PRE (assuming that H_0 is true)

Summary

- Cookbook vs. Model Comparison
- Modeling data
- Definitions of error and parameter estimates
- Models of error
- Statistical inferences about parameter values

Feedback

How was the pace of today's class?



much too slow

a little too slow

just right

a little too fast

much too fast

How happy were you with today's class overall?

What did you like about today's class? What could be improved next time?

Thank you!