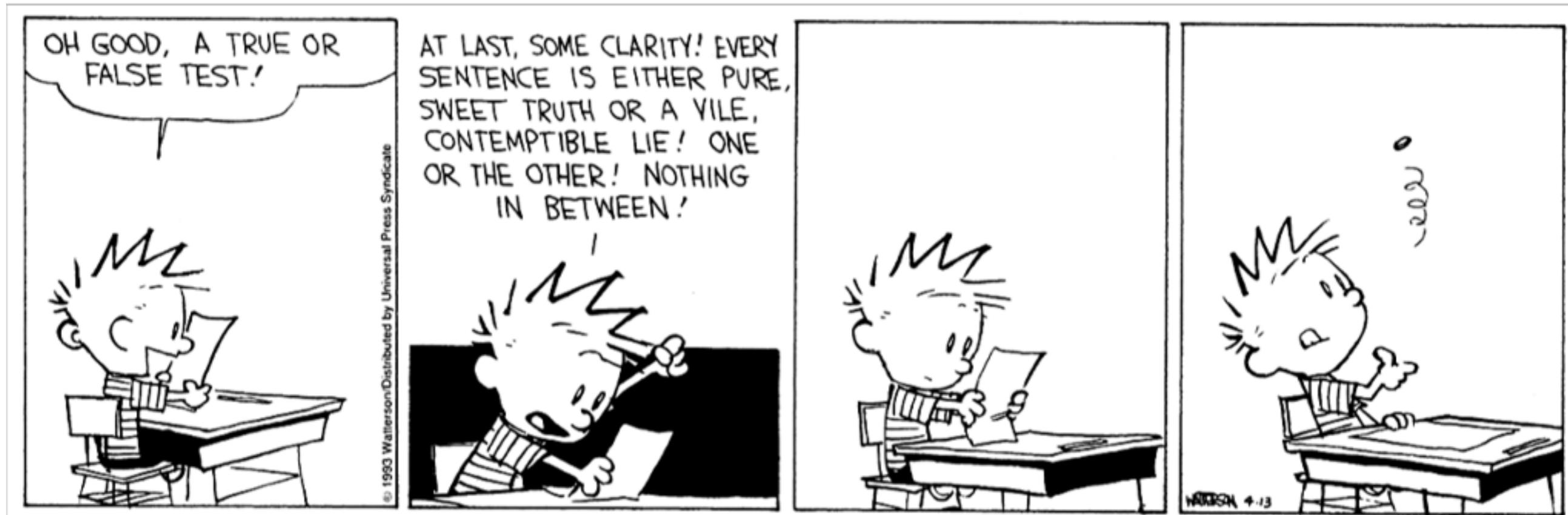


Model comparison



02/15/2019

Logistics

Final project

Project proposal

The project proposal is due on
Thursday, February 21st at 8pm

Project proposal

find some teammates!

The screenshot shows the Piazza platform interface. On the left, there's a feed of posts. A red arrow points from the word "teammates!" in the title to the first post in the feed, which is titled "Search for Teammates!". This post is marked as private and was created on 11/5/18 at 11:58AM. The content asks if others are looking for teammates for project work. To the right of this post is a "private note" titled "Search for Teammates!". This note is also private and visible only to instructors. It contains sections for "INSTRUCTORS:" and "STUDENTS:". The "INSTRUCTORS:" section has a "Make Post Public" button. The "STUDENTS:" section has a "Mark all requests as closed" button. At the bottom of the note, there's a link to "add new post" and fields for name and email.

PIAZZA PSYCH 252 ▾ Q & A Resources Statistics Manage Class Tobias Gerstenberg

hw1 hw2 hw3 hw4 logistics midterm_exam rstudio

Unread Updated Unresolved Following

New Post Search or add a post...

PINNED

■ Private **Search for Teammates!** 11/5/18

TODAY

■ Private **Q19** 5:16PM
For Q19, do we have to use the same model (fit = lm(formula = salary ~ 1 + school, data = df.placement) ? Or, can we

■ Private **error with skim and knitting ...** 4:06PM
For question 13, I was using head and skim to show a quick look at the data set (without visualizing). It appears, skim

Z score for kids in question 11? 3:03PM
We are supposed to control for the variable "kids" in question 11. In question 10 we created a z-score for the v

error removing package to install tidyv... 2:46PM
When trying to install tidyverse using install.packages("tidyverse"), I get the error: Error in install.packages

Q16 1:39PM
Q16 asks "Show the predictions of the model together with the data. There is no need to show the individual data poi
• 1 Unresolved Followup

■ Private **Anova descriptives** 12:36PM
What is the best way to get the descriptive statistics of a two-way ANOVA, for example in question 15 when there is an i

Do we have to create visualizations for... 10:24AM
The calculations can be done without visualizing, but if it is required, I can add in.

YESTERDAY

■ Private **Q19 - do we need any R code?** 10:51PM
Do we just describe our contrasts and hypotheses, or do we also actually run the R code to test it?

Q8 vs. Q5 10:03PM
Question 8 asks us "Compare the augmented model in (Question 3 -- which includes

private note ★ 3 views Actions

Search for Teammates!

INSTRUCTORS:

This post is currently private and visible only to instructors.
If you anticipate your students needing to find teammates for project work, make this post public.

Make Post Public

You can mark all open requests as closed.

Mark all requests as closed

Need to form teams? Create a post below to initiate a search and we'll notify you via email when others respond.

add new post:

I'm one student looking for more people to work with.
 I'm from a group looking for more students.

*Name Tobias Gerstenberg *Email gerstenberg@stanford.edu

*About Me Introduce yourself. What kind of teammate(s) are you looking for?
(Things you could include: your location, grad/undergrad, when you're available... help people get to know you!)

Submit

This private post is only visible to Instructors

edit good note 0 Updated 3 months ago by Piazza Team

Project proposal

IT'S TIME FOR A...

GROUP ASSIGNMENT!!

**Members of teams
will all the get
same grade!**



Didn't attend
any group
meetings



Doesn't
understand
the material



Gave the
presentation
but obviously
didn't know
what he was
even saying

**maximum 3 team
members**



Who is
this guy



"You can
use my
printer"

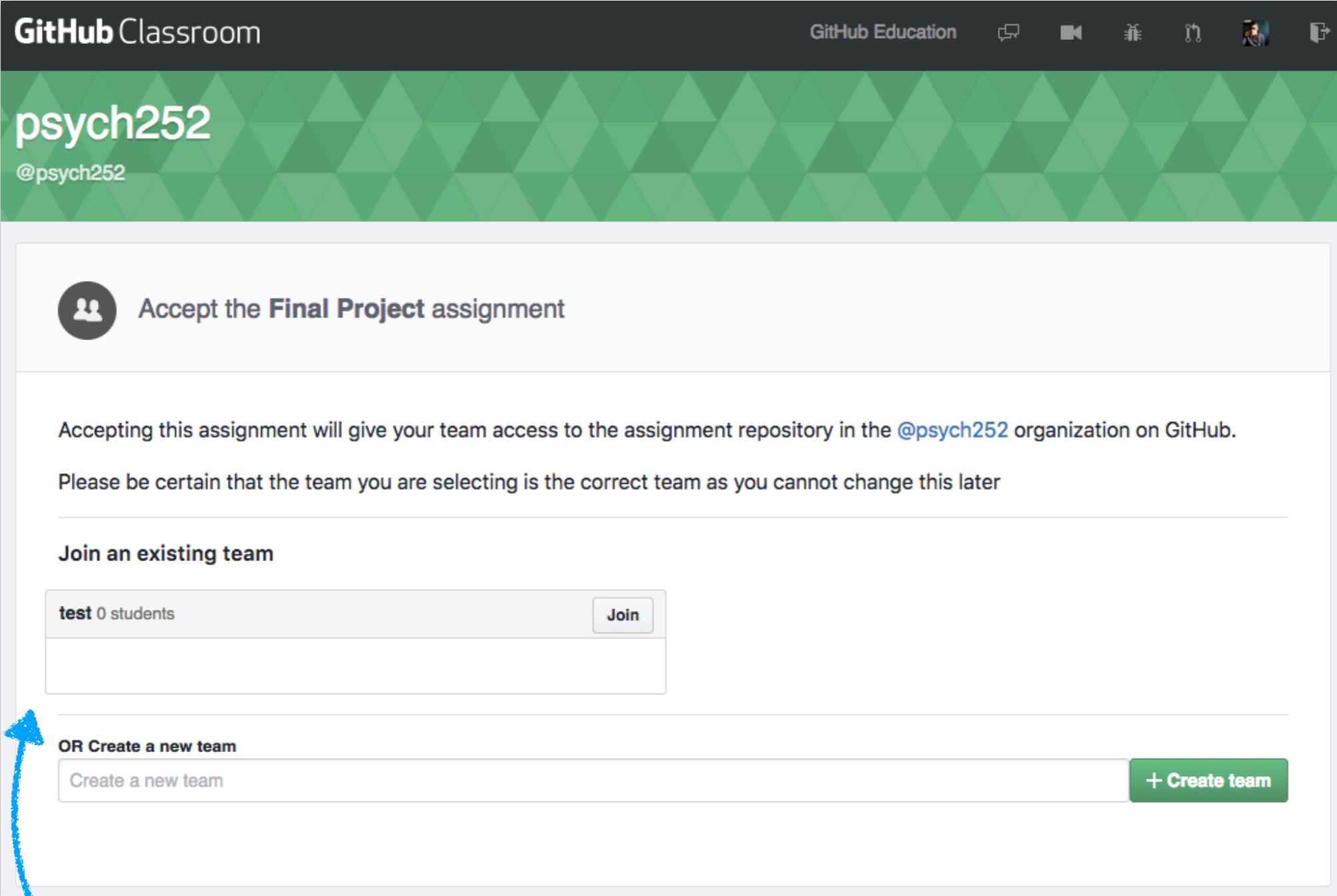


Did all the
research, wrote
paper, composed
presentation

Project proposal

<https://tinyurl.com/psych252github>

Project proposal



The screenshot shows the GitHub Classroom interface for the organization `psych252`. At the top, there's a navigation bar with `GitHub Classroom`, `GitHub Education`, and various icons. Below the header, the organization name `psych252` and handle `@psych252` are displayed against a green geometric background. The main content area contains a section titled `Accept the Final Project assignment` with a user icon. It explains that accepting the assignment will give the team access to a repository in the `@psych252` organization on GitHub. A note cautions users to ensure they select the correct team, as changes cannot be made later. Below this, there are two sections: `Join an existing team` (showing one team named `test` with 0 students and a `Join` button) and `OR Create a new team` (with a `Create a new team` button and a `+ Create team` button). A blue arrow points from the bottom text to the `OR Create a new team` section.

join a team or make a new one

Project proposal

The screenshot shows a GitHub repository page for the user 'psych252'. The repository name is 'final-projects' and it is private. The page displays basic repository statistics: 5 commits, 1 branch, 0 releases, and 1 contributor (tobiasgerstenberg). A 'Clone or download' button is visible. Below the stats, a list of files and their commit history is shown:

File	Commit Message	Time Ago
code/R	initial commit	20 minutes ago
data	initial commit	20 minutes ago
figures	initial commit	20 minutes ago
papers	initial commit	20 minutes ago
presentation	initial commit	20 minutes ago
writeup	initial commit	20 minutes ago
.gitignore	initial commit	20 minutes ago
README.md	.	14 minutes ago

Below the file list, there is a section titled 'Final project' which contains a note about starter code and general points. The 'General points' section lists several guidelines for folder and file names, relative paths, organization, and specific notes about empty folders.

Final project

Starter code for your final project.

General points

- for folder and file names:
 - don't use white space in either folder or filenames, use an underscore "_" instead
 - (almost always) use lower case only
- always use relative paths in your code
 - for example, to save a figure from an R script inside the `code/R/` folder the path should be `"../../figures/figure_name.pdf"`
- keep your folder structure organized
 - we recommend adhering to the folder structure in this repository
 - more complex projects may have additional folders such as `videos/`, `papers/`, ...
- note: some of the folders are empty except for a `.keep` file
 - the `.keep` file is just there to make sure that github includes the otherwise empty folder
 - feel free to delete the `.keep` file once you've added another file to that folder

Repository structure

- each team will have their own private github repository
- all work on your final project should happen within this repository
- sections next week will give a quick **github** tour
- post on Piazza in case you experience any problems getting set up

Project proposal

RMarkdown template



```
---
```

```
title: "A catchy project title goes here"
subtitle: "My team's name goes here"
author: "The team members' names go here"
date: "`r Sys.time()`"
urlcolor: blue # to show hyperlinks in blue when printed as pdf
```

```
# edit the output format below
# output: html_document # use this to render to html
output: pdf_document # use this to render to pdf
```

```
---
```

```
# Instructions
```

The project proposal is due on Thursday, February 21st at 8pm. It should not be longer than 500 words. It may contain code (code doesn't count toward the word limit).

```
## Research question
```

- What's your main research question?
- Which hypotheses are you trying to test?

```
## Methods
```

- Describe the data that you have. Are they based on an experiment, or will you work with an existing data set?

```
## Analysis
```

- What analyses are you planning on using?

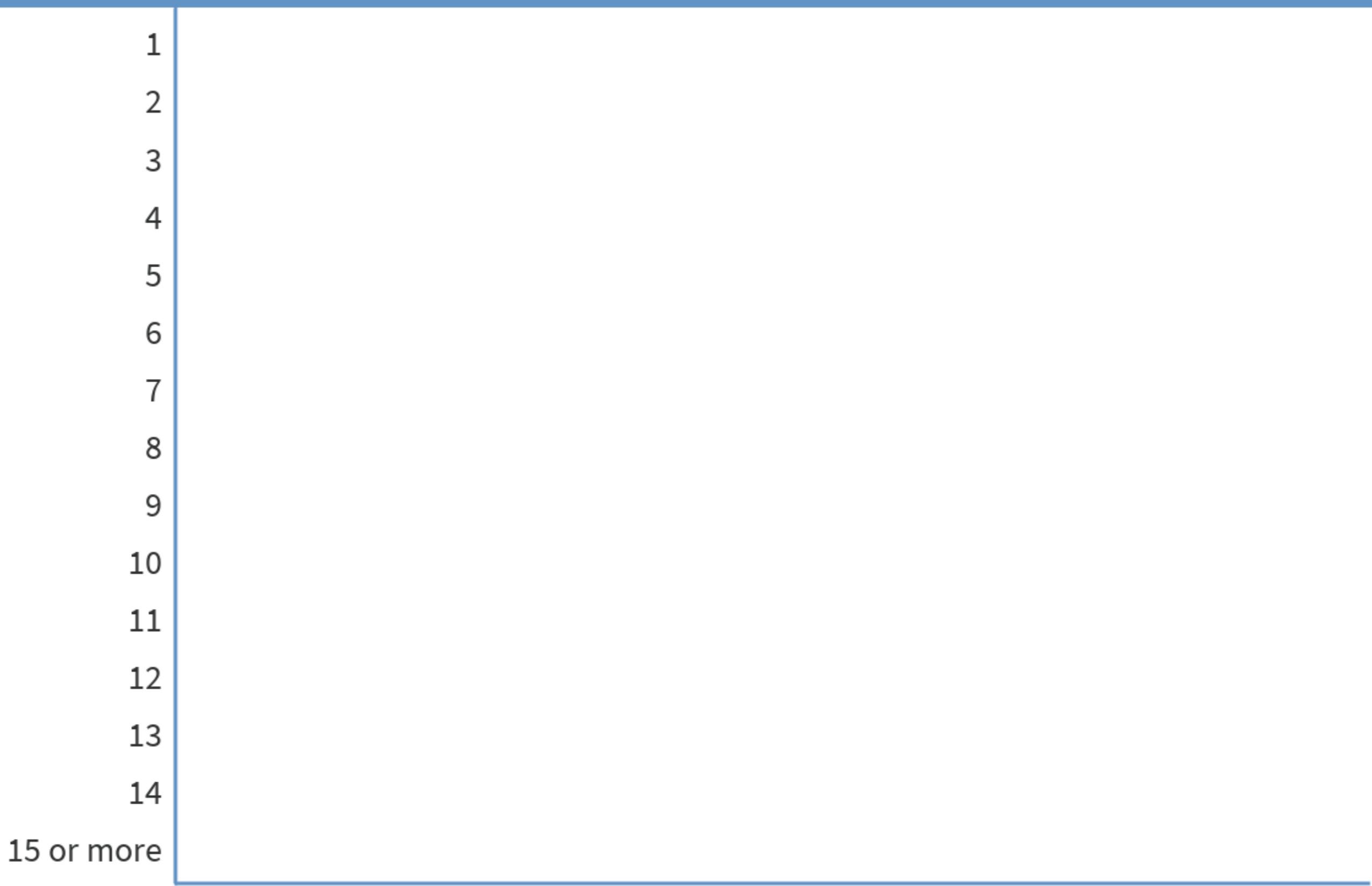
```
## Report
```

- What figures and/or tables are you planning on showing?

Upload the
pdf to canvas

Midterm

How many hours did it take you to complete the midterm?



Plan for today

- Quick power review
- Determining sample size
- Model comparison
 - Cross-validation
 - AIC and BIC

Quick power review

Type I Error



Type II Error



H_0 : Not pregnant. H_1 : Pregnant.

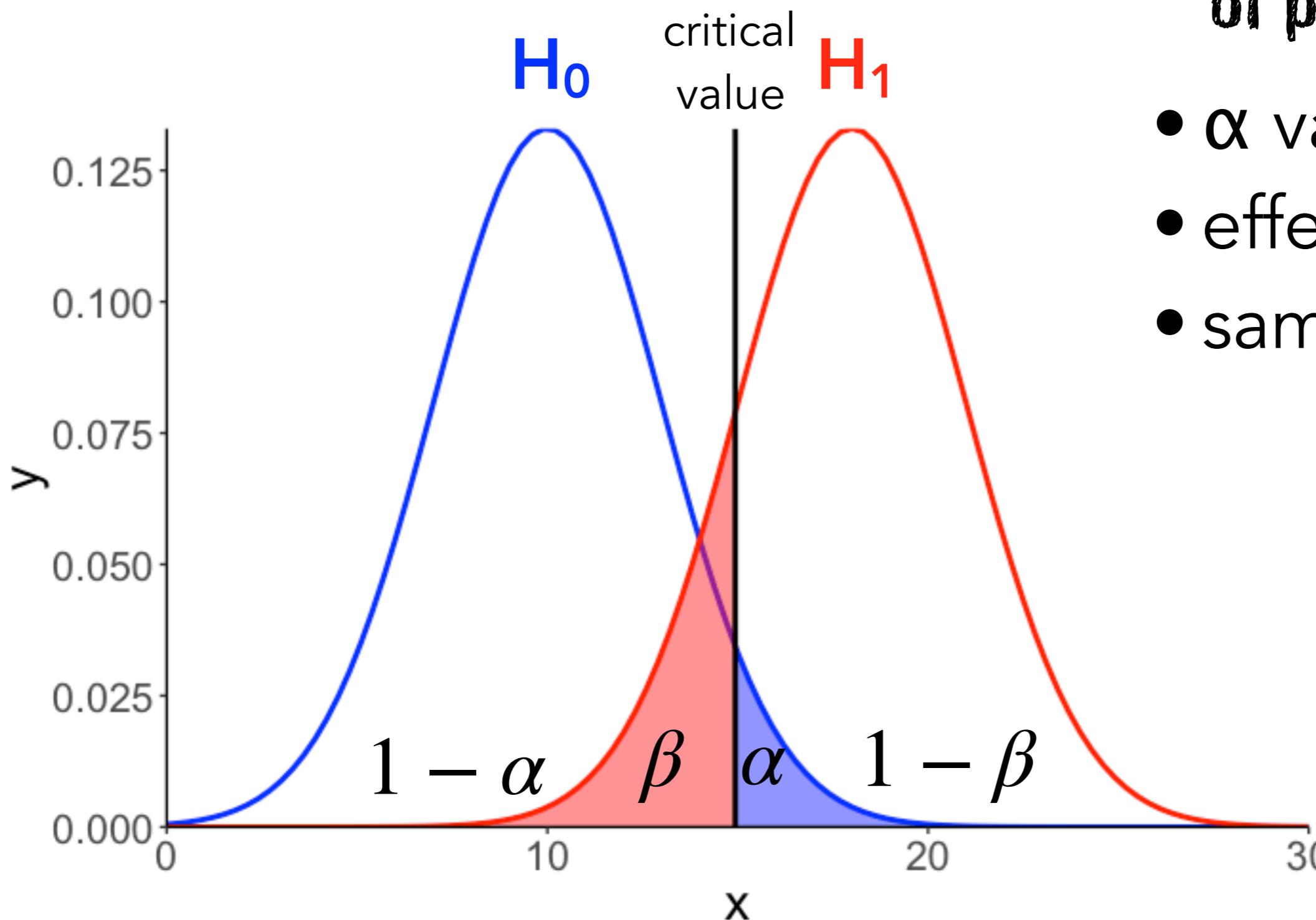
Type I Error: Falsely rejecting the null hypothesis (even though it is true).

Type II Error: Failing to reject the null hypothesis (even though it is false).

What affects power?

Determinants of power

- α value
- effect size
- sample size

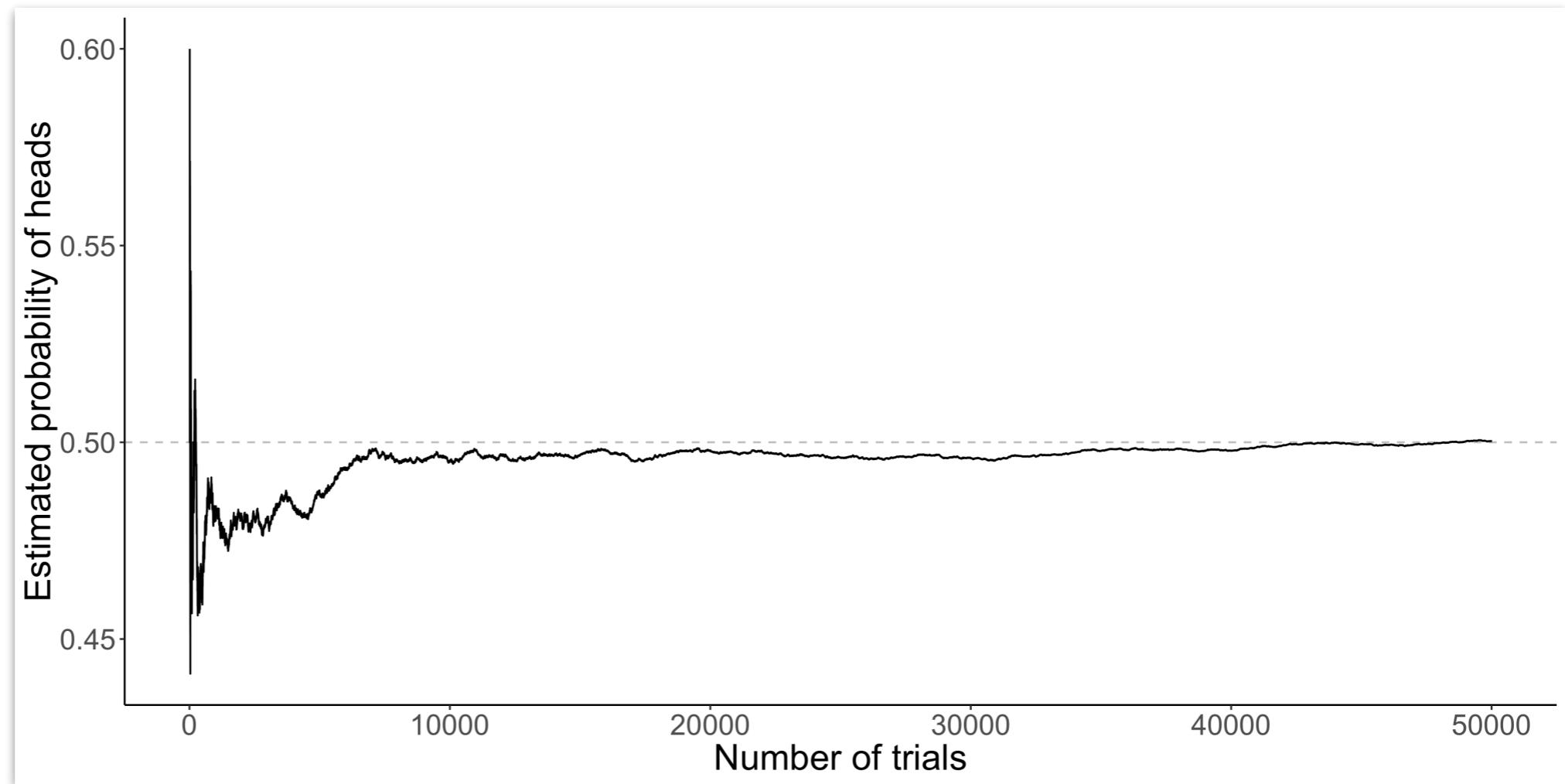


Determining sample size

How many participants do I need to run to have a good chance of detecting a true effect?

Frequentist interpretation

Probabilities = **long-range frequencies**



law of large numbers = empirical probability will approximate the true probability as the sample size increases

Example

$H_0: p(\text{coin} = \text{heads}) = 0.5$

$H_1: p(\text{coin} = \text{heads}) = 0.75$

$\alpha = 0.05$

$1 - \beta = 0.8$

How many times should we flip the coin to have a high probability (or power), say 0.80, of correctly rejecting H_0 if our coin is indeed loaded to land heads 75% of the time?

```
1 library("pwr")
2 pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),
3             sig.level = 0.05,
4             power = 0.80,
5             alternative = "greater")
```

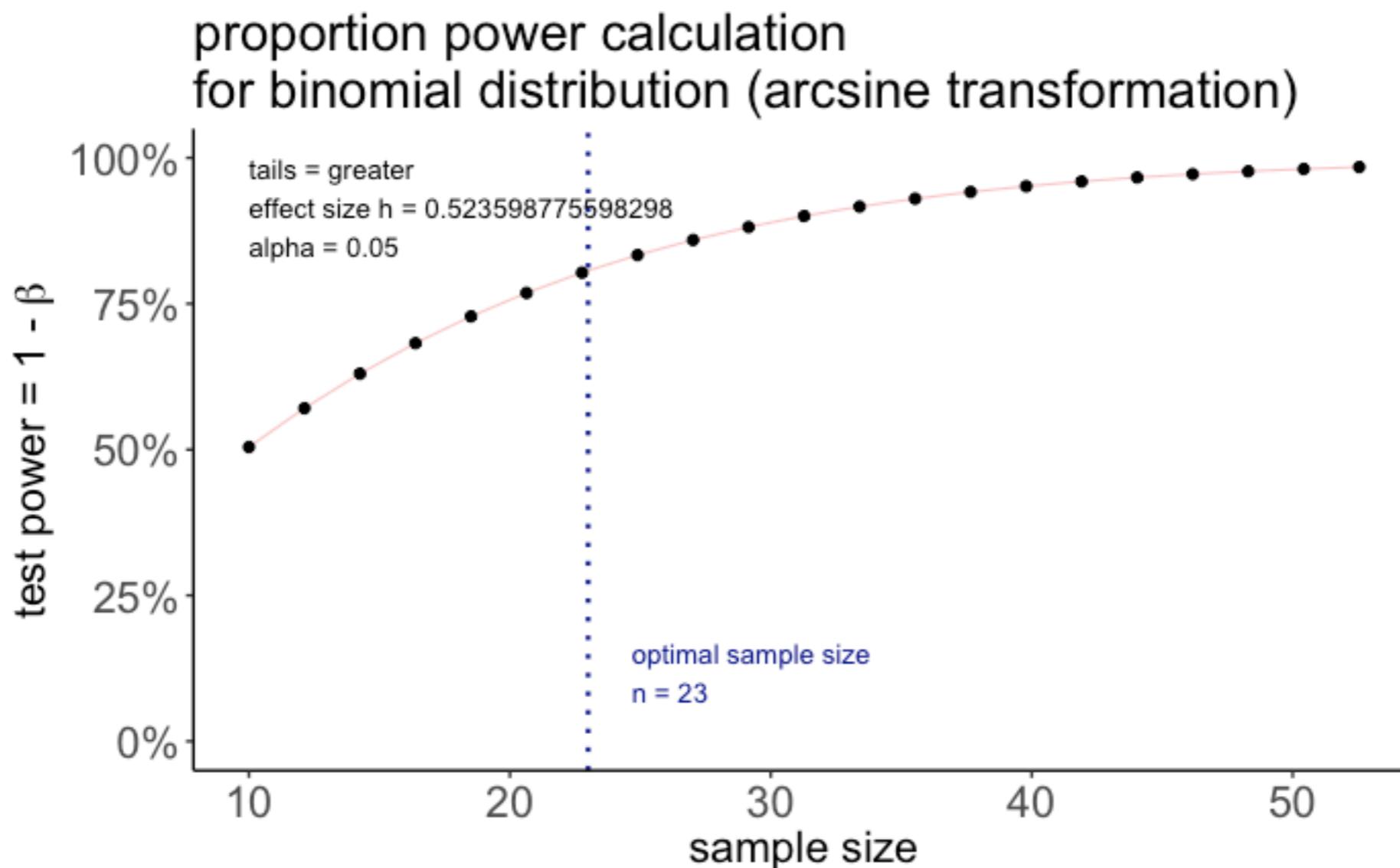
We need to flip it 23 times.

```
proportion power
calculation for binomial
distribution (arcsine
transformation)

h = 0.5235988
n = 22.55126
sig.level = 0.05
power = 0.8
alternative = greater
```

Example

```
1 pwr.p.test(h = ES.h(p1 = 0.75, p2 = 0.50),  
2               sig.level = 0.05,  
3               power = 0.80,  
4               alternative = "greater") %>%  
5   plot()
```



Power simulation recipe

- assume:
 - α , n , effect size
- simulate a large number of data sets of size n with the specified effect size
- for each data set, run a statistical test to calculate the p-value for a given α
- determine the probability of rejecting the H_0 (given that H_1 is true)

Let's simulate

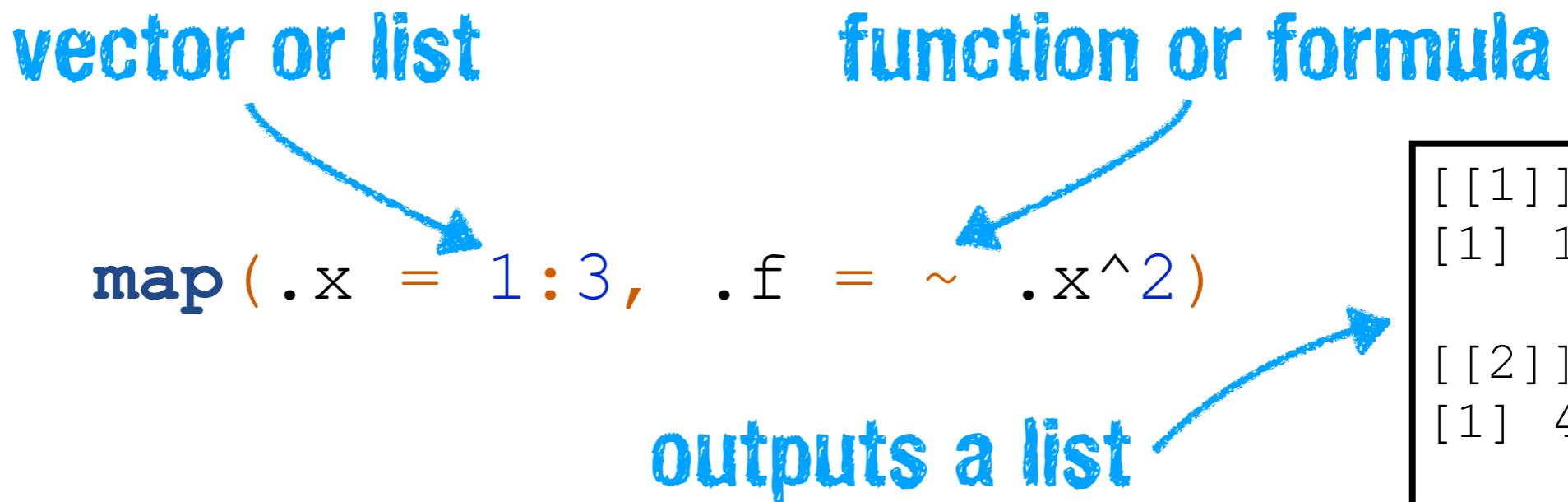
```
library("purrr")
```



automatically loaded with
library("tidyverse")

map ()

map ()



- **map**(list, function) applies a function to each element of the list
- it's a unified version of the many different **apply**() functions in base R
- you already know a cousin of **map**(): **replicate**()
- using **map**() allows one to avoid writing loops
- it's extremely powerful in combination with data frames

map ()

vector or list

`map (.x = 1:3, .f = ~ .x^2)`

function or formula

outputs a list

[[1]]
[1] 1
[[2]]
[1] 4
[[3]]
[1] 9

same same but different

`map (.x = 1:3, .f = function (.x) .x^2)`

anonymous function

map ()

vector or list

`map (.x = 1:3, .f = ~ .x^2)`

function or formula

outputs a list

[[1]]
[1] 1
[[2]]
[1] 4
[[3]]
[1] 9

different versions of map ()

differ in what they output

`map dbl (.x = 1:3, .f = ~ .x^2)`

outputs a vector

[1] 1 4 9

map ()

same same but different

```
map (.x = 1:3, .f = ~ .x^2)
```

```
map (1:3, ~ .x^2)
```

```
map (1:3, ~ .^2)
```

using a function

```
square = function(x) {x^2}
```

```
map (1:3, square)
```

map2 ()

with multiple arguments

```
map_dbl (.x = 1:3, .f = ~ .x^2)
```

```
# with multiple arguments
```

```
map2_dbl (.x = 1:3, .y = 1:3, .f = ~ .x * .y)
```

Let's simulate ...

```
1 # number of simulations
2 n_simulations = 1000
3
4 # run simulation
5 df.power = crossing(n = seq(10, 50, 1),
6                         simulation = 1:n_simulations,
7                         p = c(0.75, 0.8, 0.85)) %>%
```

set up simulation grid

n	simulation	p
10	1	0.75
10	1	0.80
10	1	0.85
10	2	0.75
10	2	0.80
10	2	0.85
10	3	0.75
10	3	0.80
10	3	0.85
10	4	0.75

Let's simulate ...

```
1 # number of simulations
2 n_simulations = 1000
3
4 # run simulation
5 df.power = crossing(n = seq(10, 50, 1),
6                     simulation = 1:n_simulations,
7                     p = c(0.75, 0.8, 0.85)) %>%
8   mutate(index = 1:n()) %>%
9   mutate(response = rbinom(n = n(), size = n, prob = p)) %>%
```

generate
random data

n	simulation	p	index	response
10	1	0.75	1	6
10	2	0.75	1	10
10	3	0.75	1	9
10	4	0.75	1	8
10	5	0.75	1	7
10	6	0.75	1	7
10	7	0.75	1	10

Let's simulate ...

```
1 # number of simulations
2 n_simulations = 1000
3
4 # run simulation
5 df.power = crossing(n = seq(10, 50, 1),
6                         simulation = 1:n_simulations,
7                         p = c(0.75, 0.8, 0.85)) %>%
8   mutate(index = 1:n()) %>%
9   mutate(response = rbinom(n = n(), size = n, prob = p)) %>%
10  group_by(index, simulation, p) %>%
11  nest() %>%
```


put data in a
list column

	index	simulation	p	data
1	1	1	0.75	list(n = 10, response = 7)
2	2	1	0.80	list(n = 10, response = 8)
3	3	1	0.85	list(n = 10, response = 9)
4	4	2	0.75	list(n = 10, response = 9)
5	5	2	0.80	list(n = 10, response = 9)
6	6	2	0.85	list(n = 10, response = 10)
7	7	3	0.75	list(n = 10, response = 9)
8	8	3	0.80	list(n = 10, response = 5)
9	9	3	0.85	list(n = 10, response = 6)
10	10	4	0.75	list(n = 10, response = 7)

Let's simulate ...

▲	index	simulation	p	data	fit
1	1	1	0.75	list(n = 10, response = 7)	list(statistic = c(`number of successes` = 7), param...
2	2	1	0.80	list(n = 10, response = 9)	list(statistic = c(`number of successes` = 9), param...
3	3	1	0.85	list(n = 10, response = 9)	list(statistic = c(`number of successes` = 9), param...
4	4	2	0.75	list(n = 10, response = 8)	list(statistic = c(`number of successes` = 8), param...
5	5	2	0.80	list(n = 10, response = 8)	list(statistic = c(`number of successes` = 8), param...
6	6	2	0.85	list(n = 10, response = 10)	list(statistic = c(`number of successes` = 10), param...
7	7	3	0.75	list(n = 10, response = 7)	list(statistic = c(`number of successes` = 7), param...
8	8	3	0.80	list(n = 10, response = 8)	list(statistic = c(`number of successes` = 8), param...
9	9	3	0.85	list(n = 10, response = 9)	list(statistic = c(`number of successes` = 9), param...
10	10	4	0.75	list(n = 10, response = 6)	list(statistic = c(`number of successes` = 6), param...

```
12 mutate(fit = map(data,
13   ~ binom.test(x = .$response,
14     n = .$n,
15     p = 0.5,
16     alternative = "two.sided")),
```

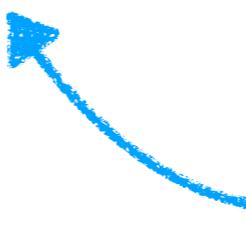
run binomial
test for each
cell in the data
column

map(list, function)
applies function to each
element of the list

Let's simulate ...

index	simulation	p	p.value	n	response
1	1	0.75	1.093750e-01	10	8
2	2	0.80	1.953125e-03	10	10
3	3	0.85	3.437500e-01	10	7
4	4	0.75	2.148438e-02	10	9
5	5	0.80	2.148438e-02	10	9
6	6	0.85	3.437500e-01	10	7
7	7	0.75	3.437500e-01	10	7
8	8	0.80	1.093750e-01	10	8
9	9	0.85	2.148438e-02	10	9
10	10	0.75	2.148438e-02	10	9

```
13 mutate(fit = map(data,
14   ~ binom.test(x = .$response,
15     n = .$n,
16     p = 0.5,
17     alternative = "two.sided")),
18   p.value = map_dbl(fit, ~ .$p.value)) %>%
19 unnest(data) %>%
20 select(-fit)
```



extract p-values

Let's simulate ...

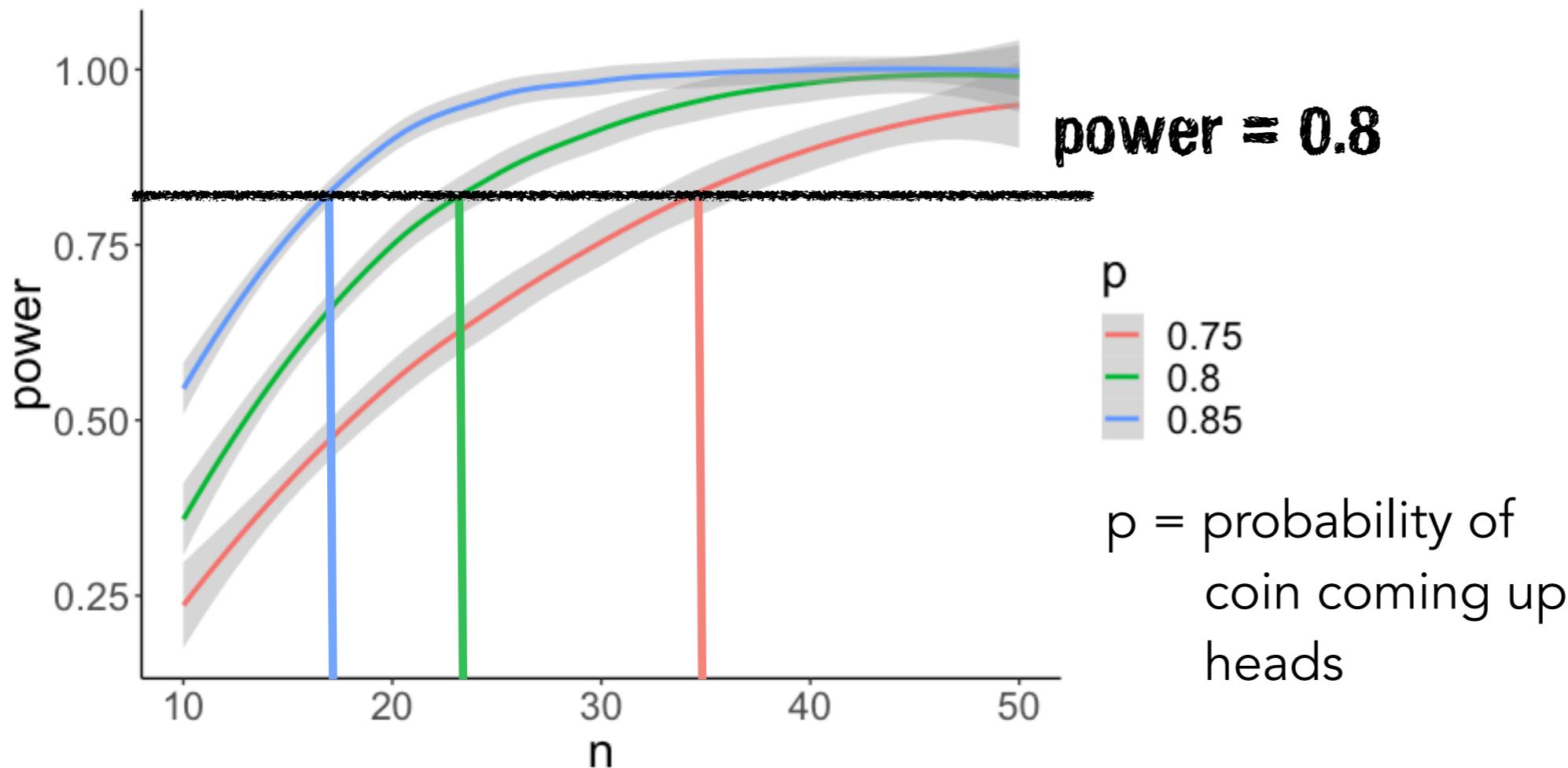
```
1 # data frame for plot  
2 df.plot = df.power %>%  
3   group_by(n, p) %>%  
4   summarize(power = sum(p.value < 0.05) / n())
```

probability of rejecting
 H_0 when H_1 is true

Power

$$1 - \beta$$

$p(\text{reject } H_0 | H_1 \text{ is true})$



n	p	power
10	0.75	0.24
10	0.8	0.38
10	0.85	0.58
11	0.75	0.24
11	0.8	0.40
11	0.85	0.50
12	0.75	0.45
12	0.8	0.55
12	0.85	0.82
13	0.75	0.26

Let's simulate ...

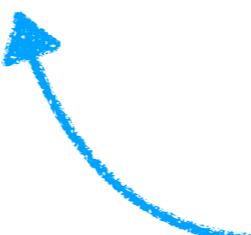
```
1 # analytic solution  
2 pwr.p.test(h = ES.h(0.5, 0.75),  
3               power = 0.8,  
4               alternative = "two.sided")
```

proportion power calculation for binomial distribution (arcsine transformation)

```
h = 0.5235988  
n = 28.62923  
sig.level = 0.05  
power = 0.8  
alternative = two.sided
```

```
1 # based on simulations  
2 df.plot %>%  
3   filter(p == 0.75, near(power, 0.8, tol = 0.02))
```

n	p	power
30	0.75	0.81



cool function that filters on close values



Studio[®]

time

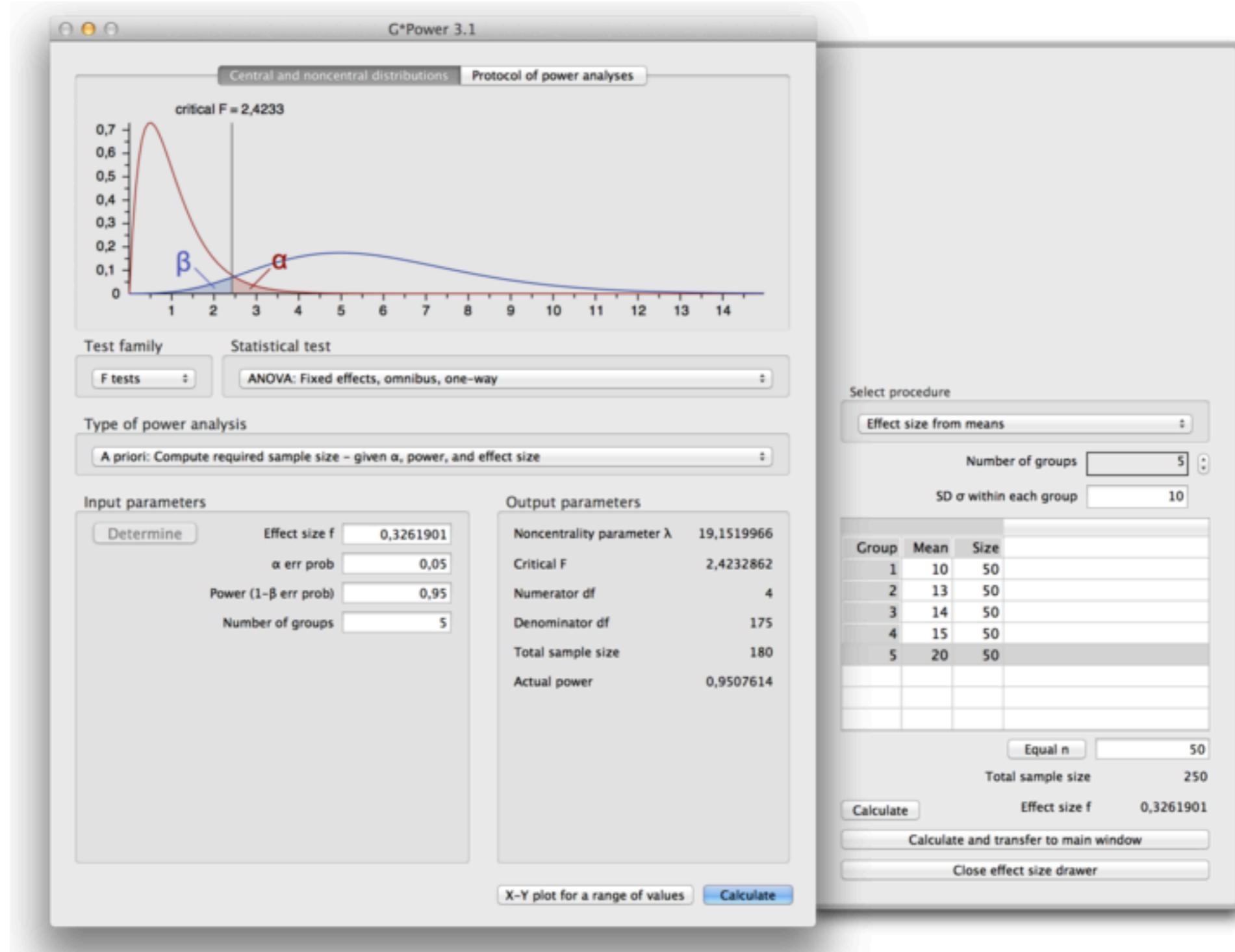
Let's simulate ...

- here, I've used a very simple example (Binomial test)
- but: we can use the same recipe for any statistical test that we are planning on running

Power simulation recipe

- assume:
 - α , n , effect size
- simulate a large number of data sets of size n with the specified effect size
- for each data set, run a statistical test to calculate the p-value for a given α
- determine the probability of rejecting the H_0 (given that H_1 is true)

G*Power 3.1: Alternative software for power calculations



<http://www.gpower.hhu.de/>

Model comparison

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)

Any problems with our approach?

sometimes it doesn't work ...

Model C

$$\text{balance}_i = \beta_0 + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{student}_i + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{student}_i + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{student}_i + \epsilon_i$$

Model A

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{student}_i + \beta_2 \cdot \text{age}_i + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{student}_i + \beta_2 \cdot \text{age}_i + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{age}_i + \epsilon_i$$

$$\text{balance}_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{degree}_i + \epsilon_i$$

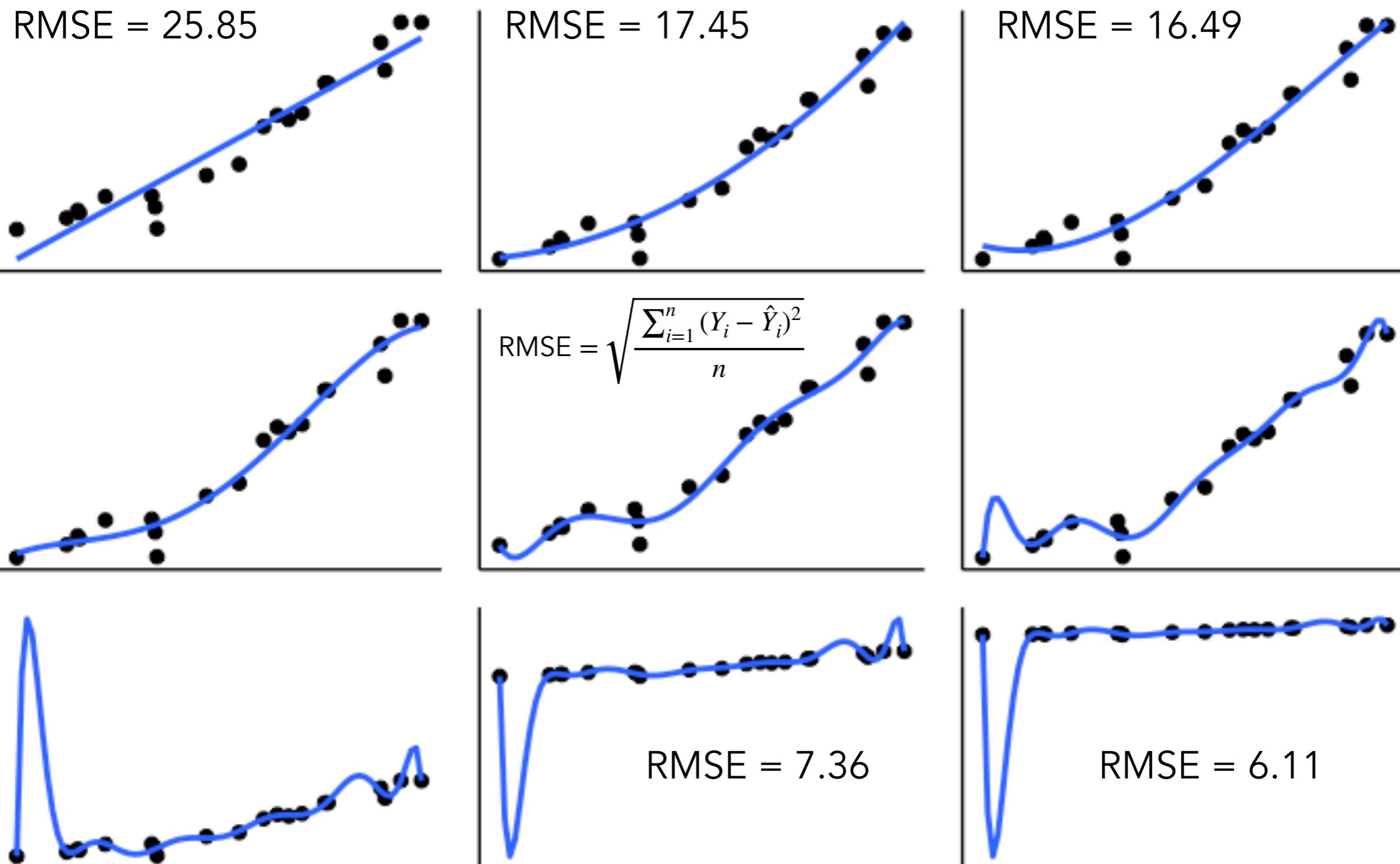


Tools for model comparison

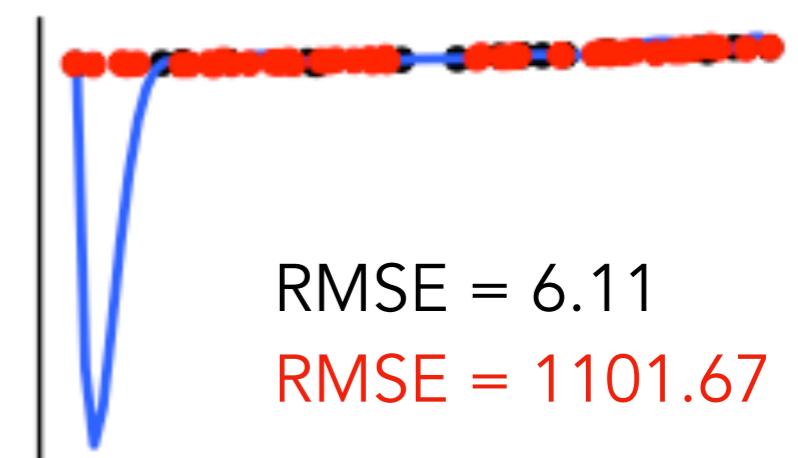
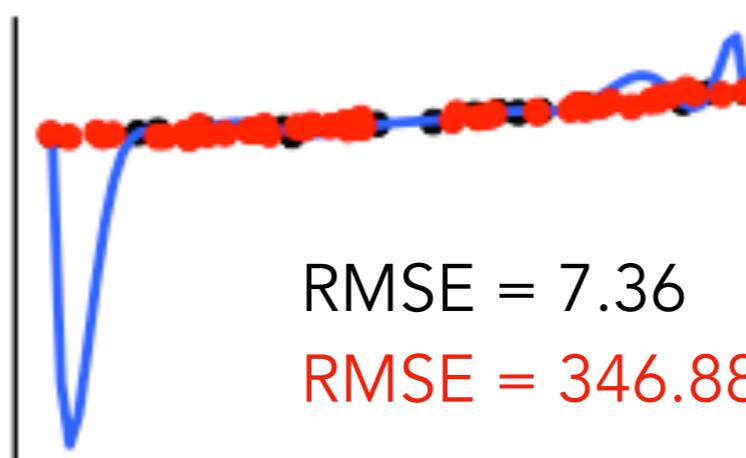
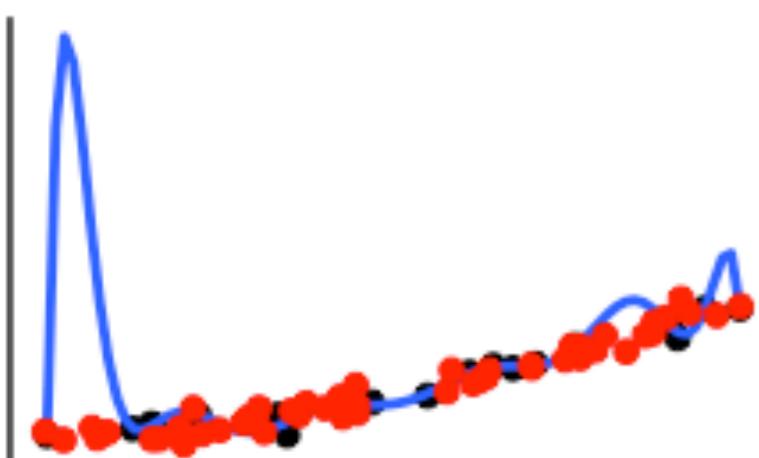
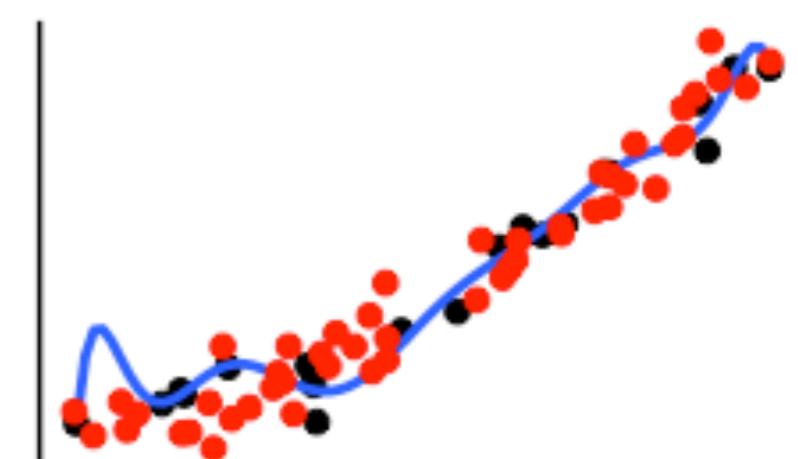
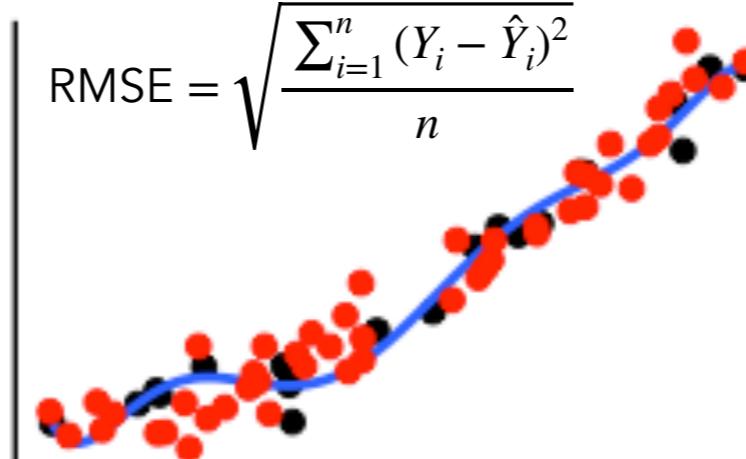
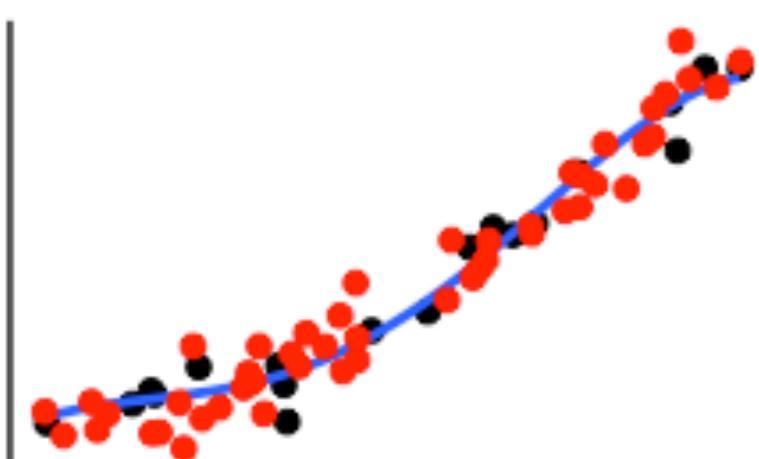
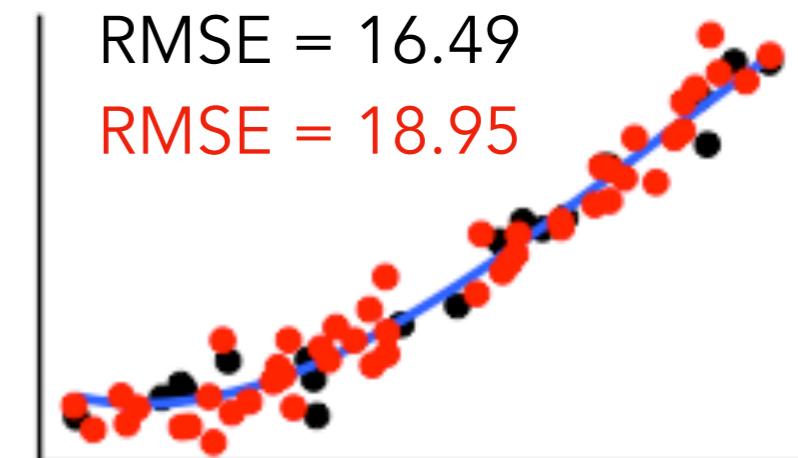
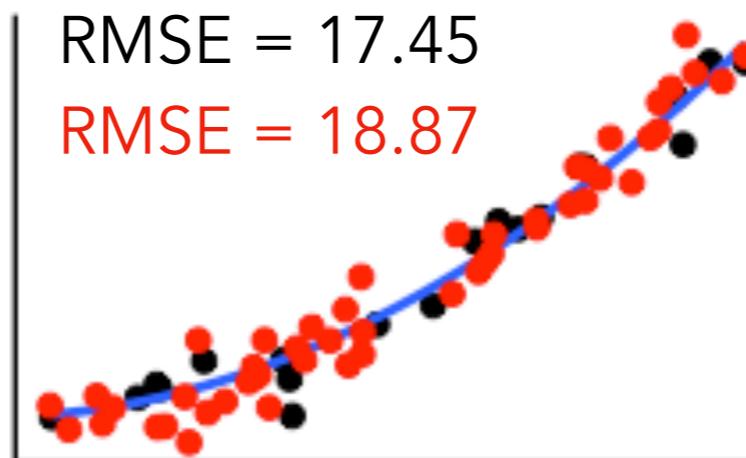
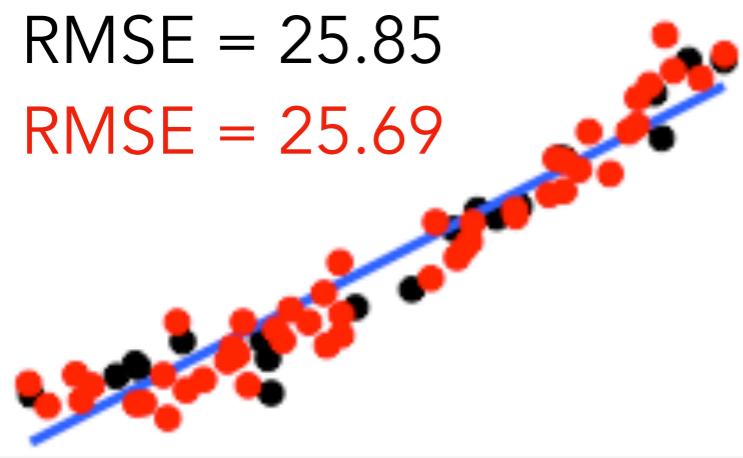
- **anova()** : compare a compact model with an augmented model via the F-test
 - problem: only works for nested models (where the augmented model contains all the predictors of the compact model and more)
- **What if we want to compare models that aren't nested?**
 - Cross-validation
 - AIC and BIC
 - Bayesian data analysis (we'll get there soon)

Cross-validation

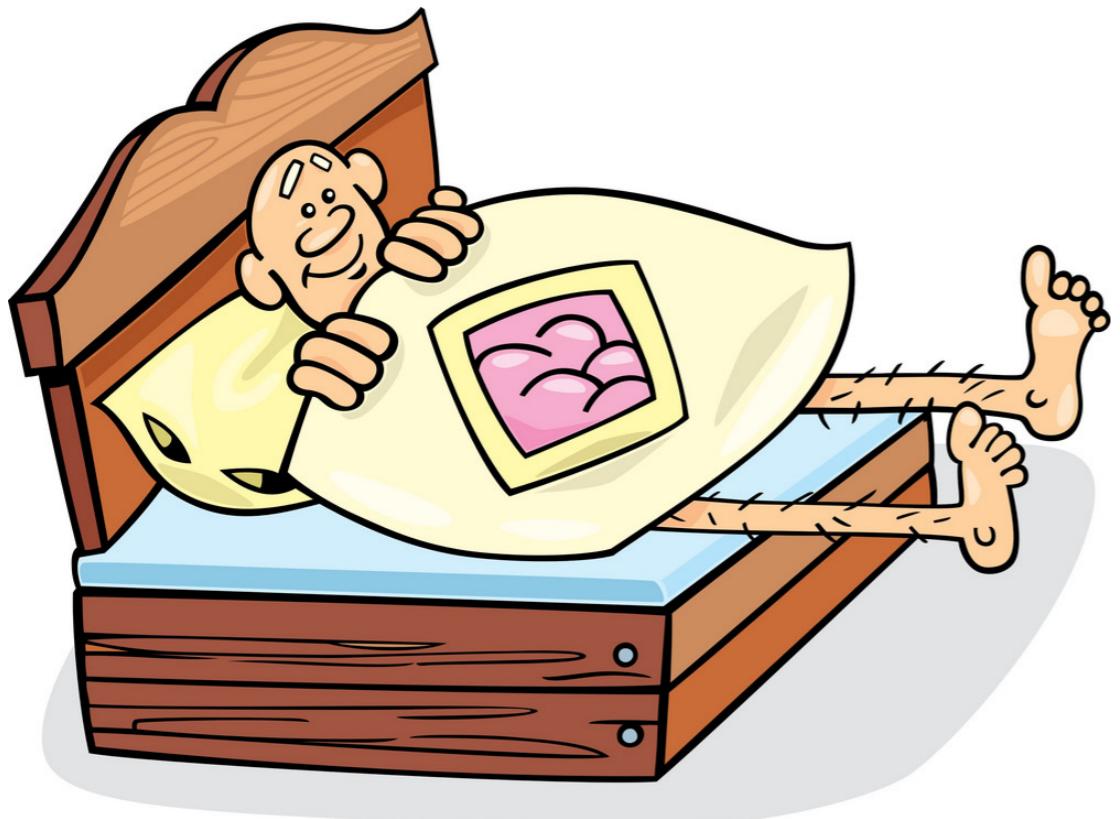
Which model describes the data best?



Which model describes the data best?



Underfitting vs. Overfitting



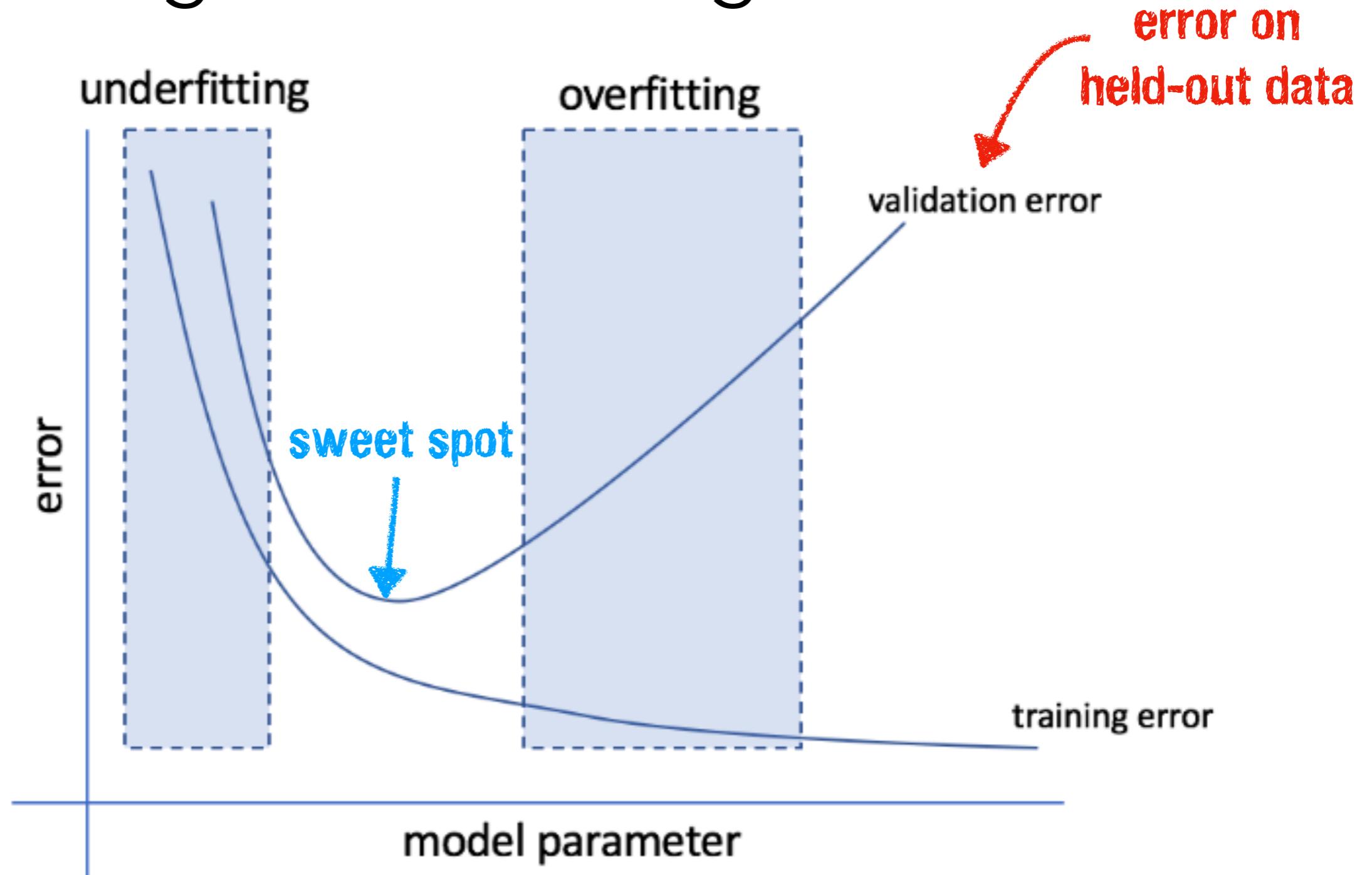
**ONE WAY TO EXPLAIN
UNDERFITTING**



Underfitting vs. Overfitting

- a good model should:
 - explain the actual data well
 - predict future data well
- bias-variance tradeoff:
 - **bias** = error from erroneous assumptions in the model, high bias can cause a model to miss the relevant relations between predictors and outcome underfitting
 - **variance** = error from sensitivity to small fluctuations in the data, high variance can cause a model to fit the random **noise** in the data overfitting

Underfitting vs. Overfitting



in machine learning, the goal is often to find the sweet spot between underfitting and overfitting

Leave-one-out crossvalidation



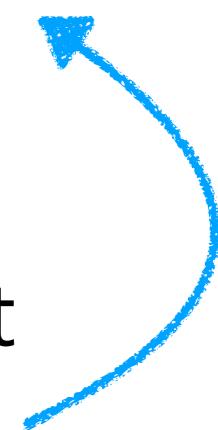
LOO

Leave One
Out

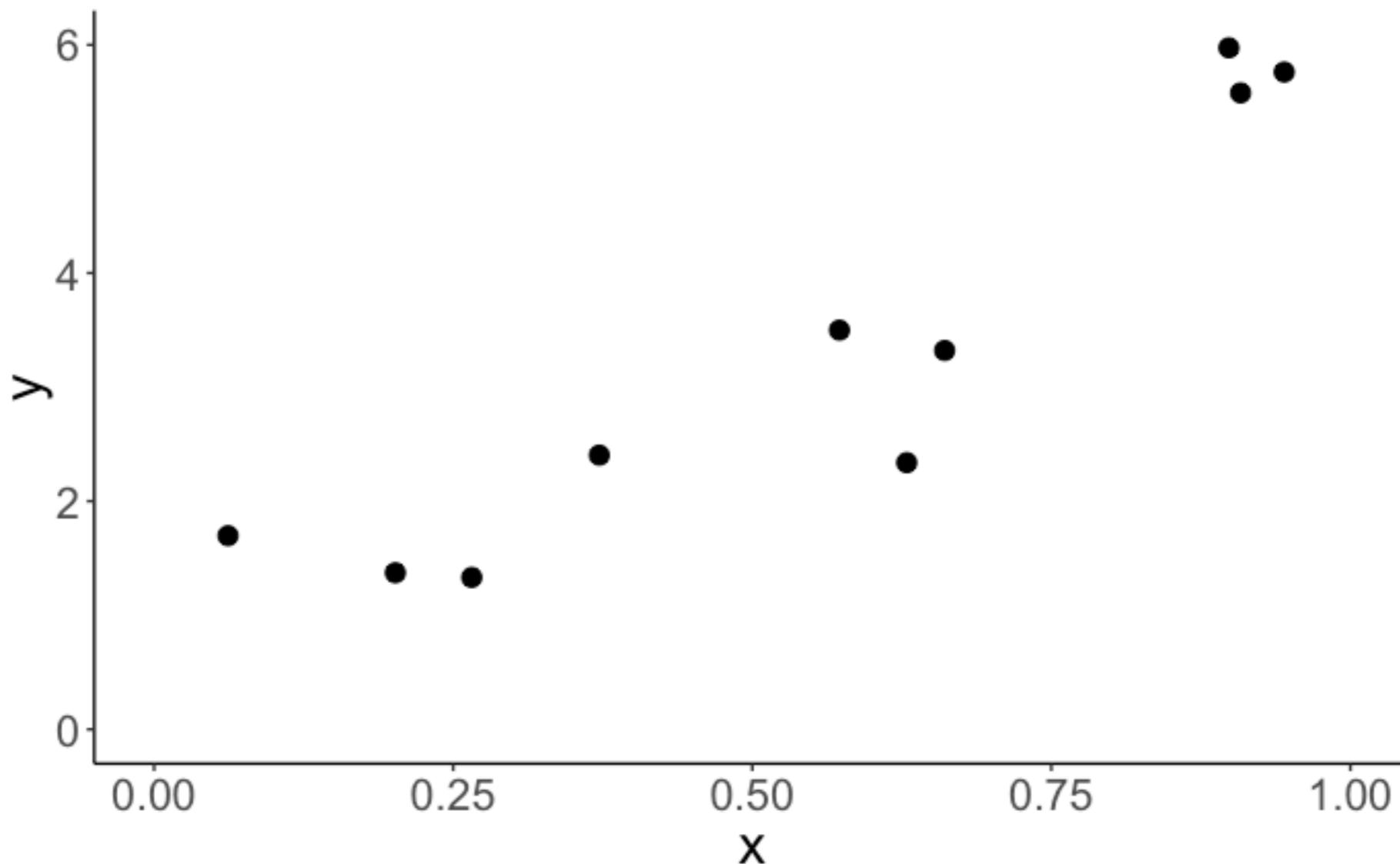
non-inspirational quote



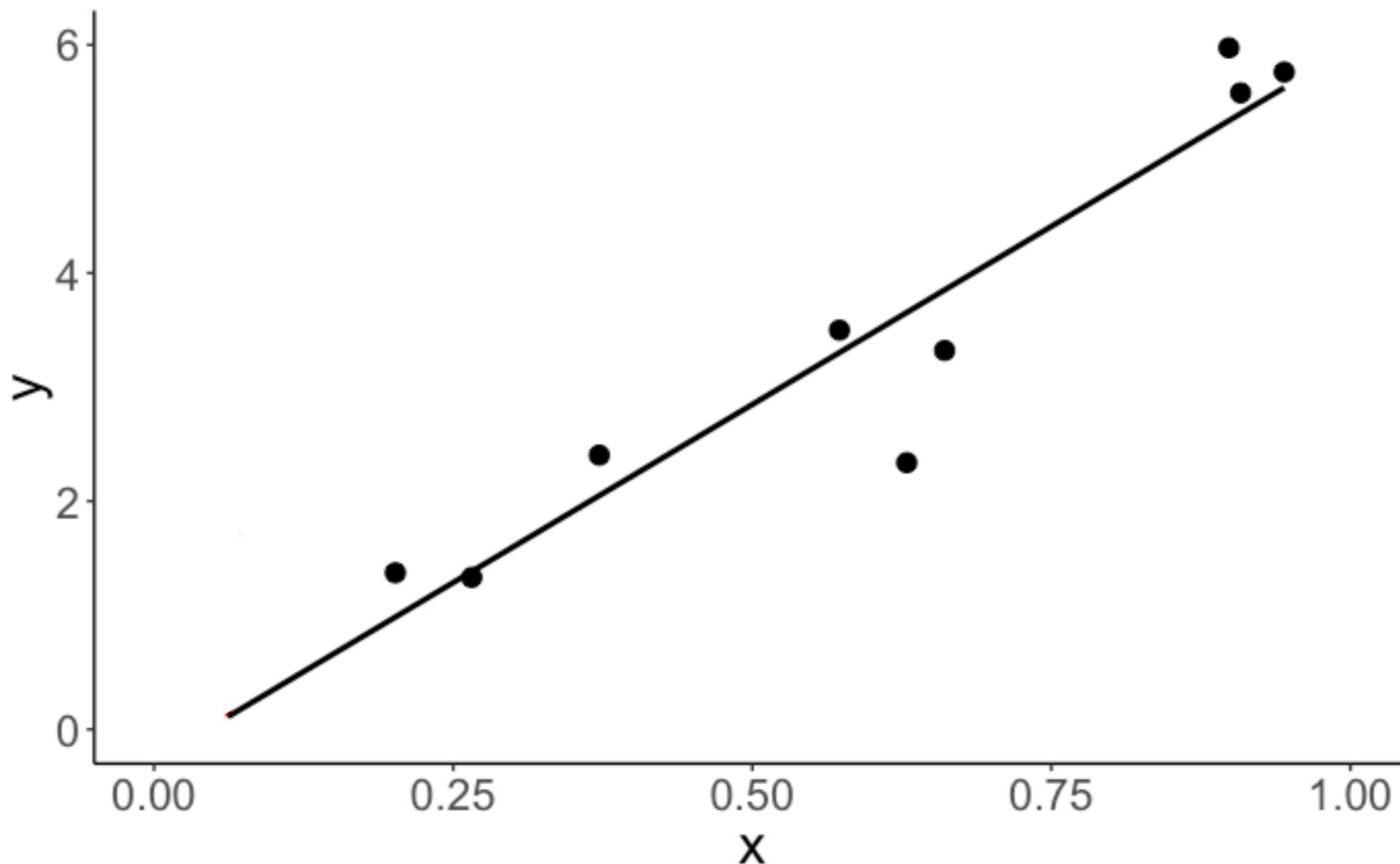
Leave one out cross-validation

- train the model on all the data points except for one
 - calculate the prediction error for the held-out data point
- repeat for all data points**
- 

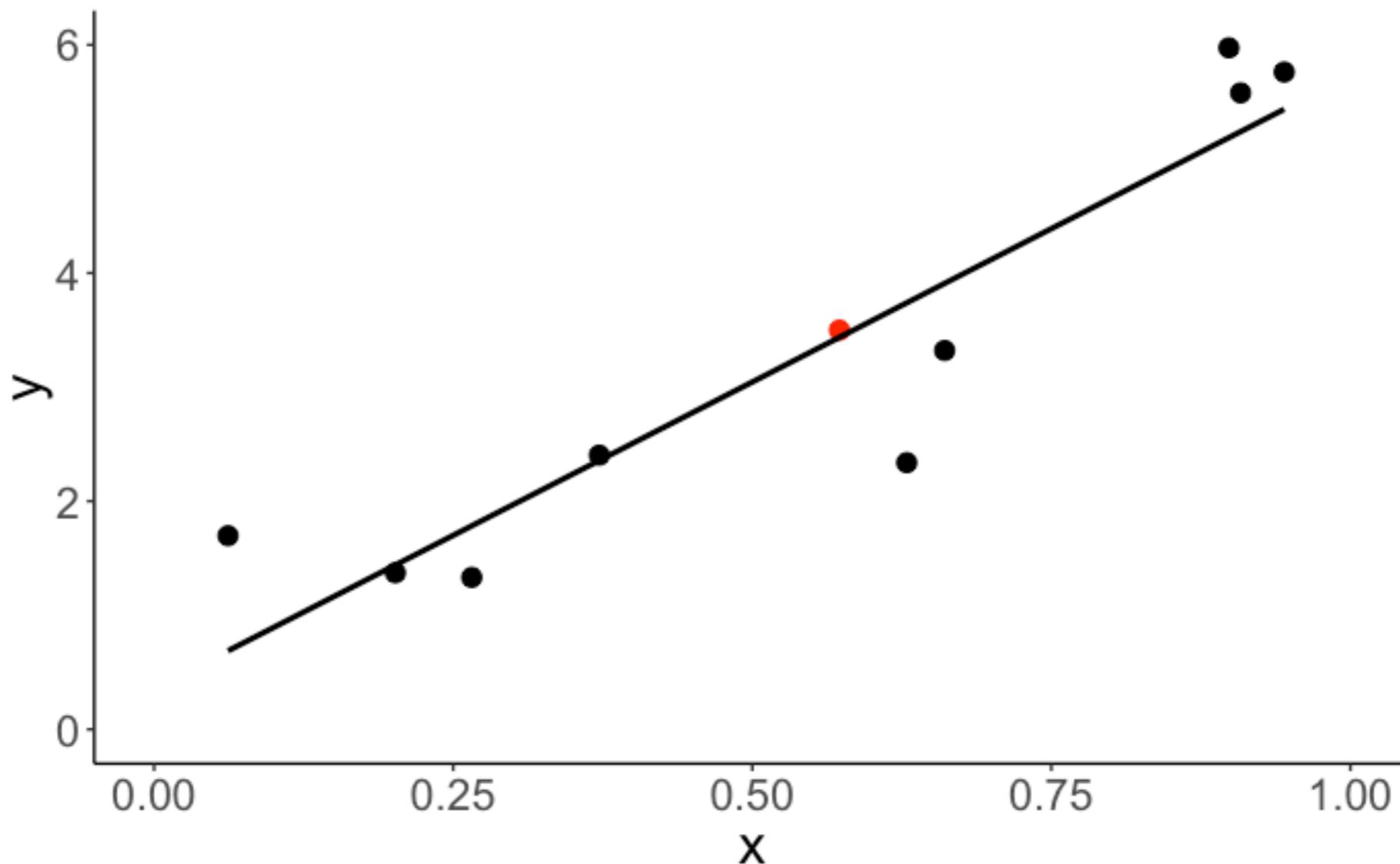
Leave one out cross-validation



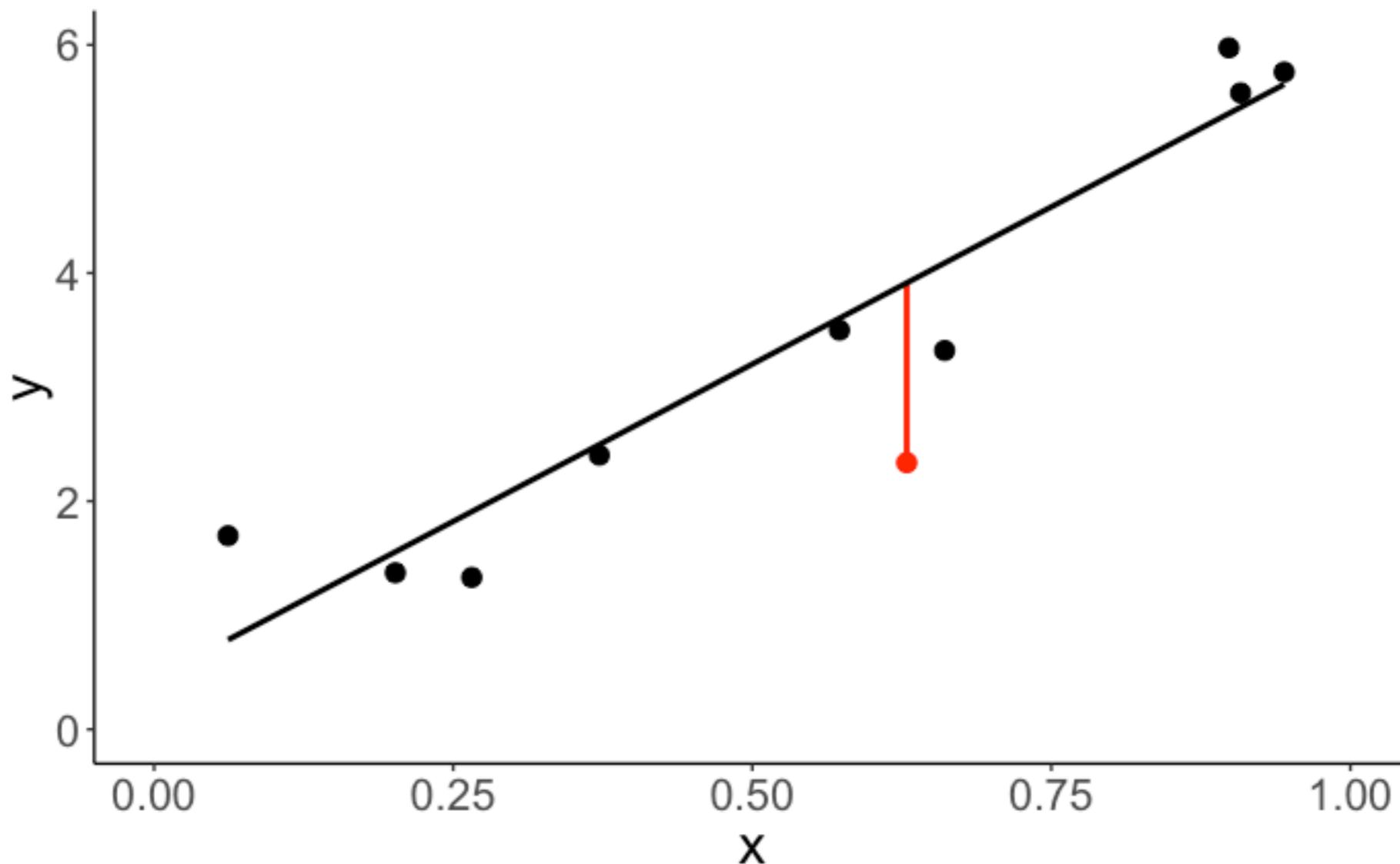
Leave one out cross-validation



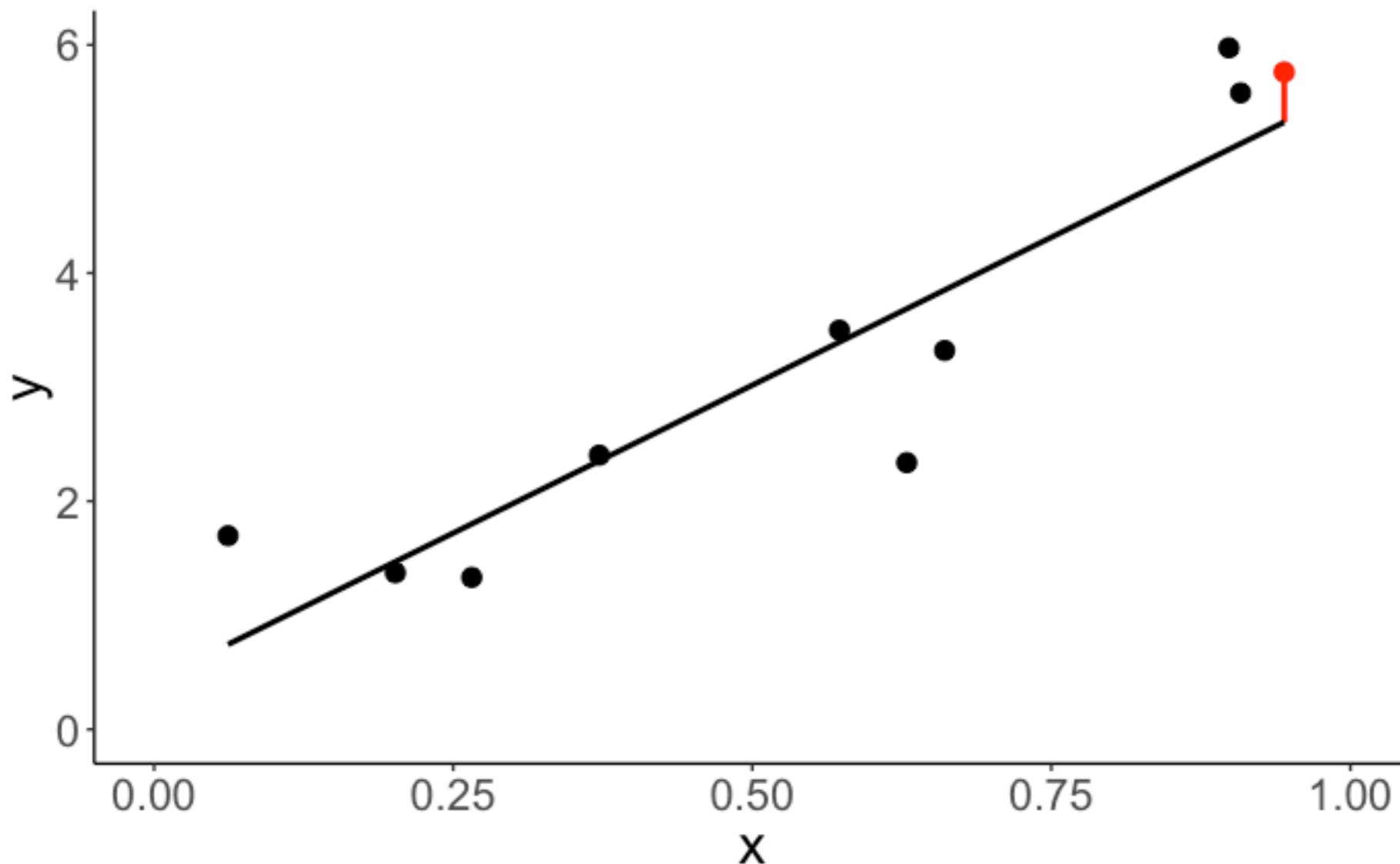
Leave one out cross-validation



Leave one out cross-validation

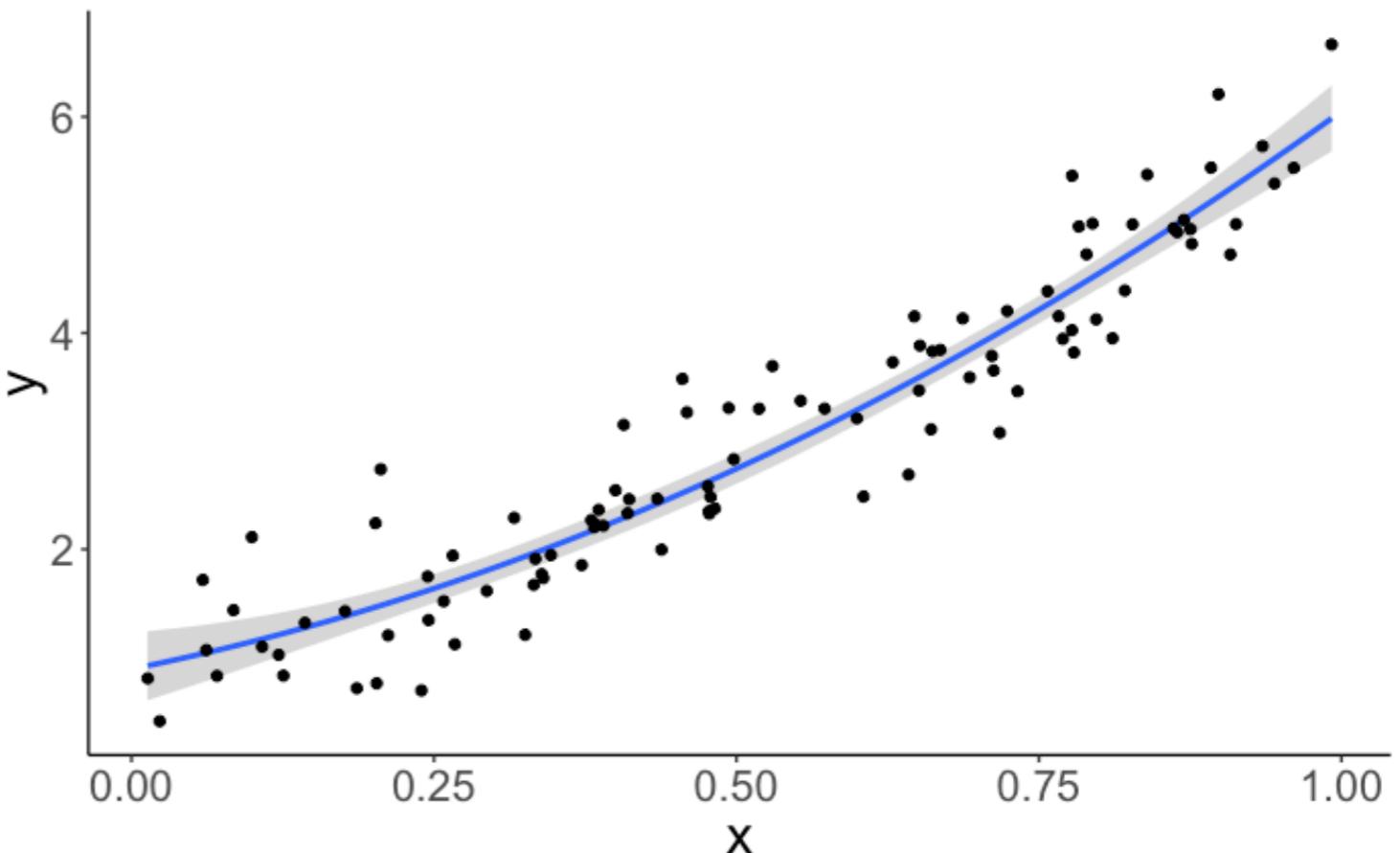


Leave one out cross-validation



Leave-one out crossvalidation

```
1 # make example reproducible
2 set.seed(1)
3
4 # parameters
5 sample_size = 100
6 b0 = 1
7 b1 = 2
8 b2 = 3
9 sd = 0.5
10
11 # sample
12 df.data = tibble(
13   participant = 1:sample_size,
14   x = runif(sample_size, min = 0, max = 1),
15   y = b0 + b1*x + b2*x^2 + rnorm(sample_size, sd = sd)
16 )
```



Leave-one out crossvalidation

ground truth

$$y_i = 1 + 2 \cdot x_i + 3 \cdot x_i^2 + e$$

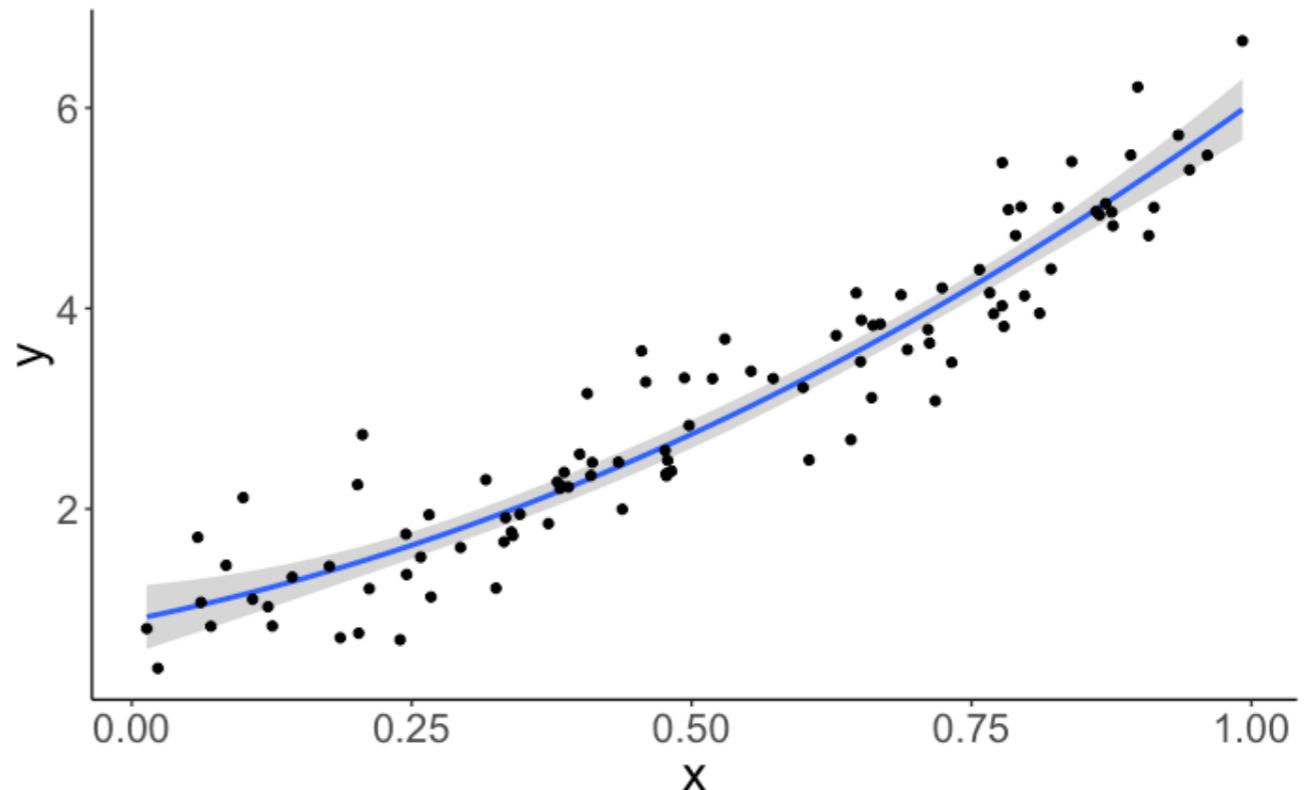
$$e \sim \mathcal{N}(\text{mean} = 0, \text{sd} = 0.5)$$

candidate models

simple $\hat{y}_i = b_0 + b_1 \cdot x_i$

correct $\hat{y}_i = b_0 + b_1 \cdot x_i + b_2 \cdot x_i^2$

complex $\hat{y}_i = b_0 + b_1 \cdot x_i + b_2 \cdot x_i^2 + b_3 \cdot x_i^3$



we could do an F-test here since the models are nested ...

Leave-one out crossvalidation

```
1 library("modelr")
2
3 df.cross = df.data %>%
4   crossv_loo() %>%
5   mutate(model_simple = map(train, ~ lm(y ~ 1 + x, data = .)),
6         model_correct = map(train, ~ lm(y ~ 1 + x + I(x^2), data = .)),
7         model_complex = map(train, ~ lm(y ~ 1 + x + I(x^2) + I(x^3), data = .))) %>%
8   gather("model", "fit", contains("model")) %>%
9   mutate(rmse = map2_dbl(fit, train, rmse))
```

model	mean_rmse
simple	0.65
correct	0.42
complex	0.41

complex model has the lowest
error on the training data

Leave-one out crossvalidation

```
1 library("modelr")
2
3 df.cross = df.data %>%
4   crossv_loo() %>%
```

splits the data set into training and test

	train	test	.id
1	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	1
2	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	2
3	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	3
4	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	4
5	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	5
6	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	6
7	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	7
8	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	8
9	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	9
10	list(data = list(participant = 1:10, x = c(0.265508663...))	list(data = list(participant = 1:10, x = c(0.265508663...))	10

each entry is simply a pointer to the data set plus some indices .idx for the filtered data points

Leave-one out crossvalidation

```
1 library("modelr")
2
3 df.cross = df.data %>%
4   crossv_loo() %>%
5   mutate(model_simple = map(train, ~ lm(y ~ 1 + x, data = .)),
6         model_correct = map(train, ~ lm(y ~ 1 + x + I(x^2), data = .)),
7         model_complex = map(train, ~ lm(y ~ 1 + x + I(x^2) + I(x^3), data = .))) %>%
```

fit three different models to the training data set

Leave-one out crossvalidation

```
1 library("modelr")
2
3 df.cross = df.data %>%
4   crossv_loo() %>%
5   mutate(model_simple = map(train, ~ lm(y ~ 1 + x, data = .)),
6         model_correct = map(train, ~ lm(y ~ 1 + x + I(x^2), data = .)),
7         model_complex = map(train, ~ lm(y ~ 1 + x + I(x^2) + I(x^3), data = .))) %>%
8   gather("model", "fit", contains("model")) %>%
9   mutate(rmse = map2_dbl(fit, test, rmse))
```

calculate the root mean squared error for each model on the test data set

```
1 df.cross %>%
2   group_by(model) %>%
3   summarize(mean_rmse = mean(rmse))
```

model	mean_rmse
simple	0.65
correct	0.48
complex	0.70

the correct model has the lowest prediction error



Studio[®]

time

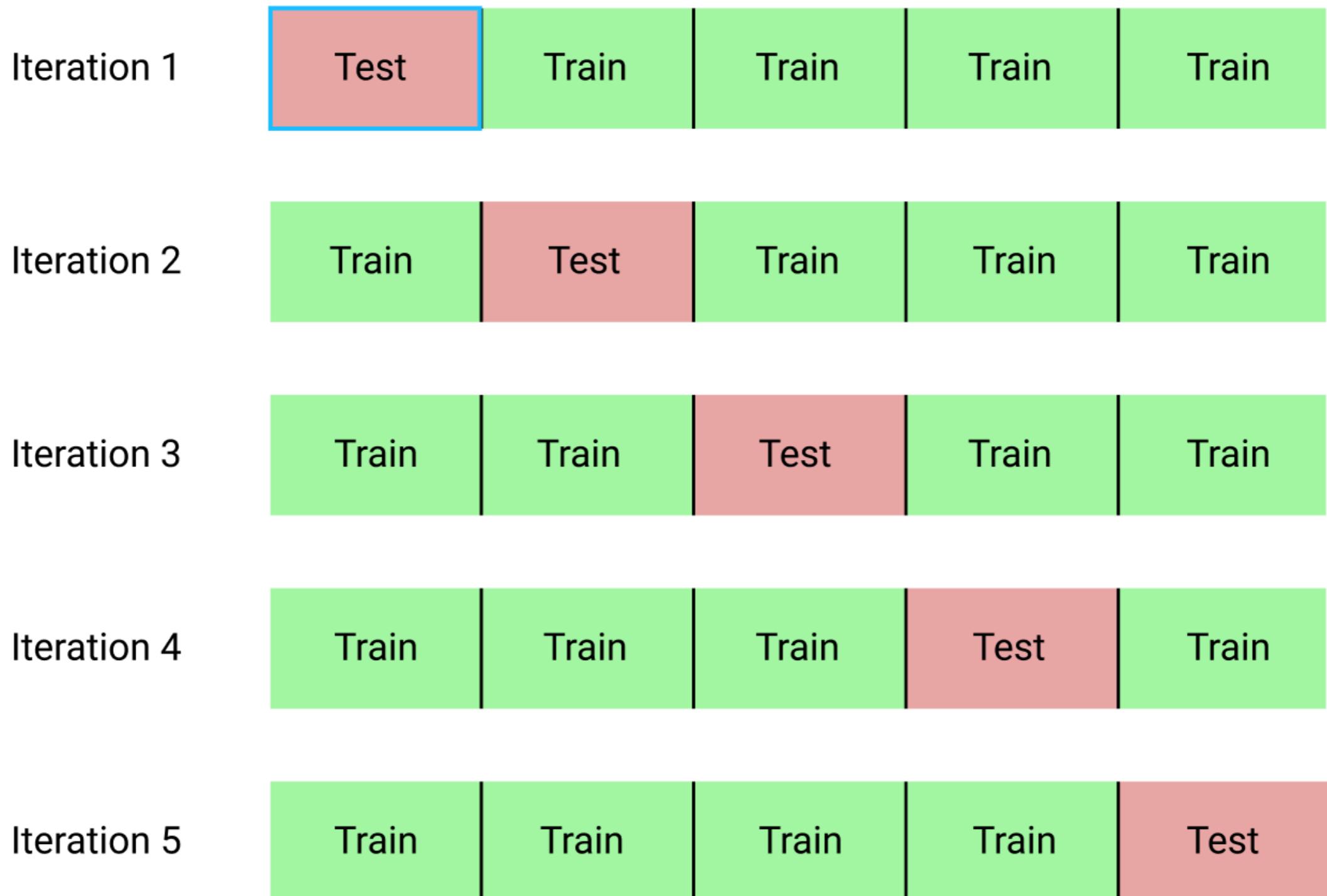
Leave-one out crossvalidation

- **problem:** can be computationally expensive since it requires fitting the model n times

k-fold cross validation

k-fold crossvalidation

Full data set



k-fold crossvalidation

k-fold crossvalidation

```
1 df.cross = df.data %>%
2   crossv_kfold(k = 10) %>%
3   mutate(model_simple = map(train, ~ lm(y ~ 1 + x, data = .)),
4         model_correct = map(train, ~ lm(y ~ 1 + x + I(x^2), data = .)),
5         model_complex = map(train, ~ lm(y ~ 1 + x + I(x^2) + I(x^3), data = .))) %>%
6   gather("model", "fit", contains("model")) %>%
7   mutate(rsquare = map2_dbl(fit, test, rsquare))
```

this wouldn't work for LOO
since we only have one data
point in the test data...

using R² as a measure

model	median_rsquare
simple	0.839
correct	0.865
complex	0.860

the correct model accounts for
the most variance in the test data

k-fold vs. leave-one-out crossvalidation

- LOO:
 - trained on **more** data
 - more variance
 - less bias
- k-fold:
 - trained on **less** data
 - less variance
 - more bias

Monte Carlo crossvalidation

Monte Carlo crossvalidation

random splits into
training and test data

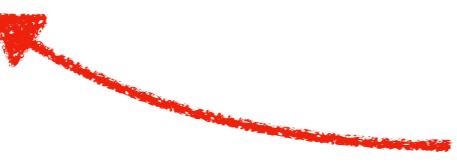
```
1 df.cross = df.data %>%
2   crossv_mc(n = 50, test = 0.5) %>% # number of samples, and percentage of test
3   mutate(model_simple = map(train, ~ lm(y ~ 1 + x, data = .x)),
4         model_correct = map(train, ~ lm(y ~ 1 + x + I(x^2), data = .x)),
5         model_complex = map(train, ~ lm(y ~ 1 + x + I(x^2) + I(x^3), data = .))) %>%
6   gather("model", "fit", contains("model")) %>%
7   mutate(rmse = map2_dbl(fit, test, rmse))
```

- splits can also be done in a **stratified** way
- for example, generate training data that has the same percentage of cases from each group
- fit data from some participants, test on data from other participants, ...

AIC and BIC

AIC and BIC

- AIC = Akaike Information Criterion
- BIC = Bayesian Information Criterion

 **not that much Bayesian about it ...**

$$\text{AIC} = 2k - 2 \ln(\hat{L})$$

$$\text{BIC} = \ln(n)k - 2 \ln(\hat{L})$$

\hat{L} = maximized value of the likelihood function of the model

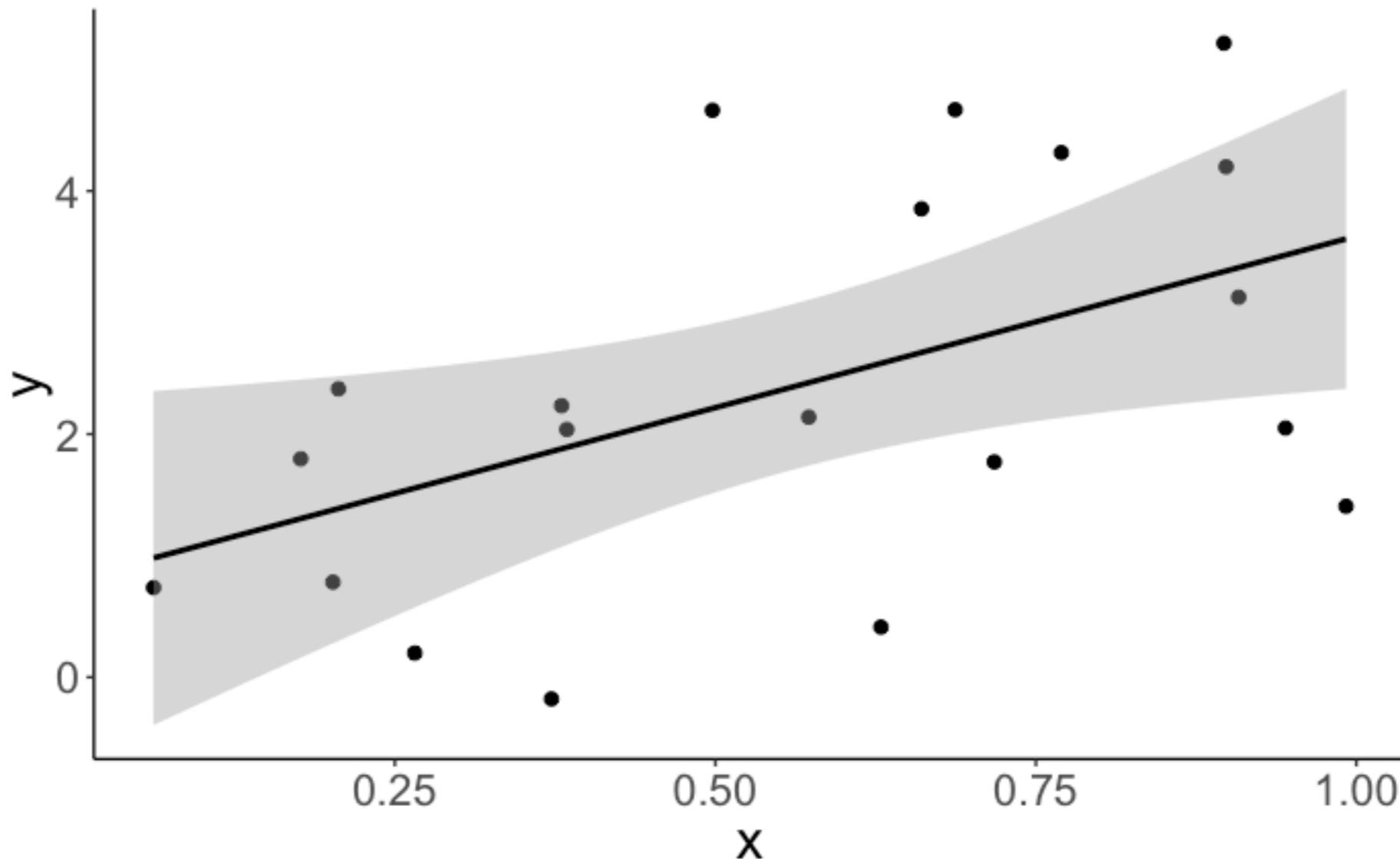
k = number of parameters in the model

n = number of observations

AIC and BIC

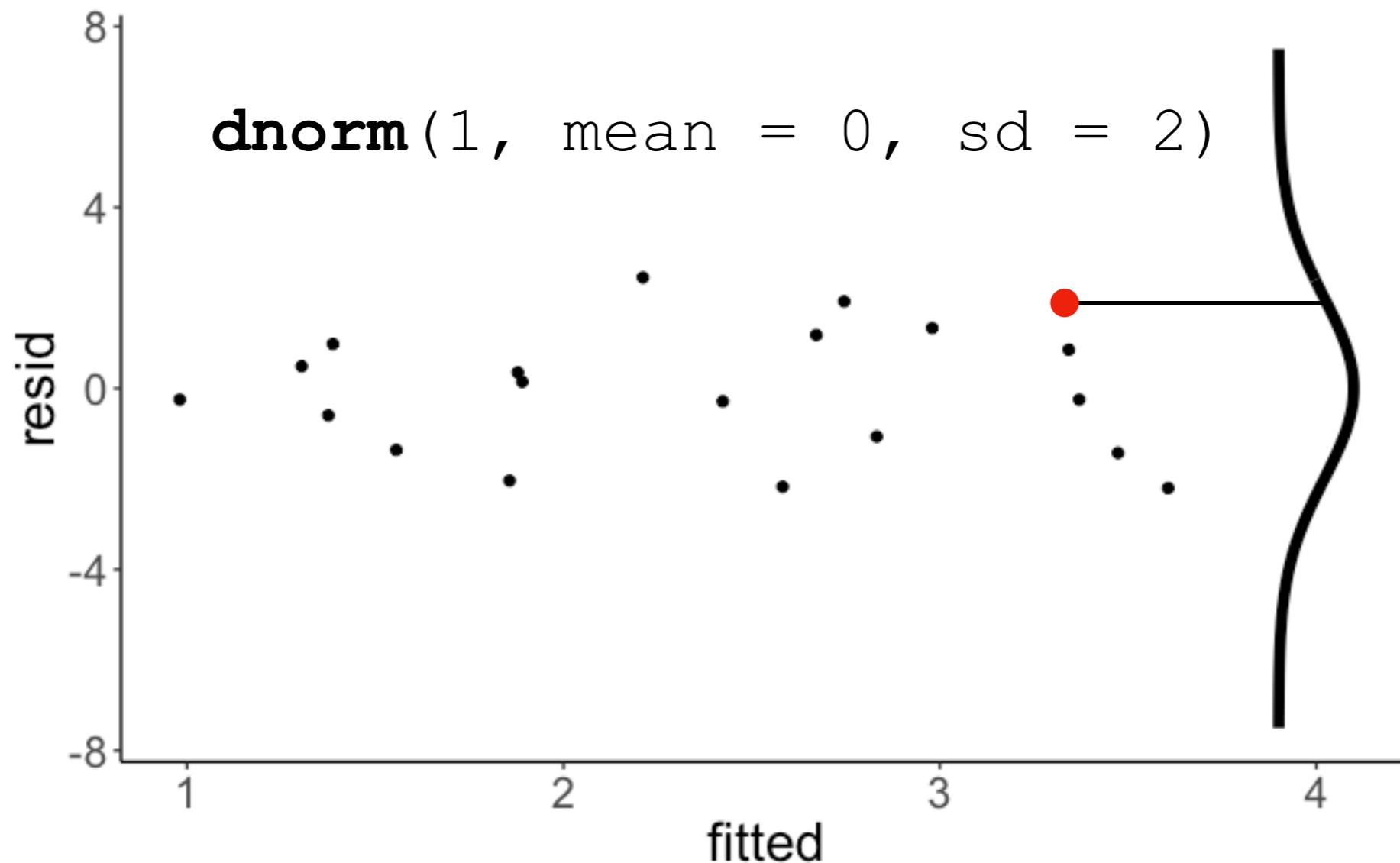
- How do we get the likelihood of our model?
 - in a linear regression, minimizing least squares is equivalent to maximizing the likelihood of the data given the model
- Assumption of the model:
 - residuals are normally distributed with:
 - mean = 0 and sd = sigma
 - calculate overall likelihood by computing the likelihood of each residual, and then multiplying

AIC and BIC



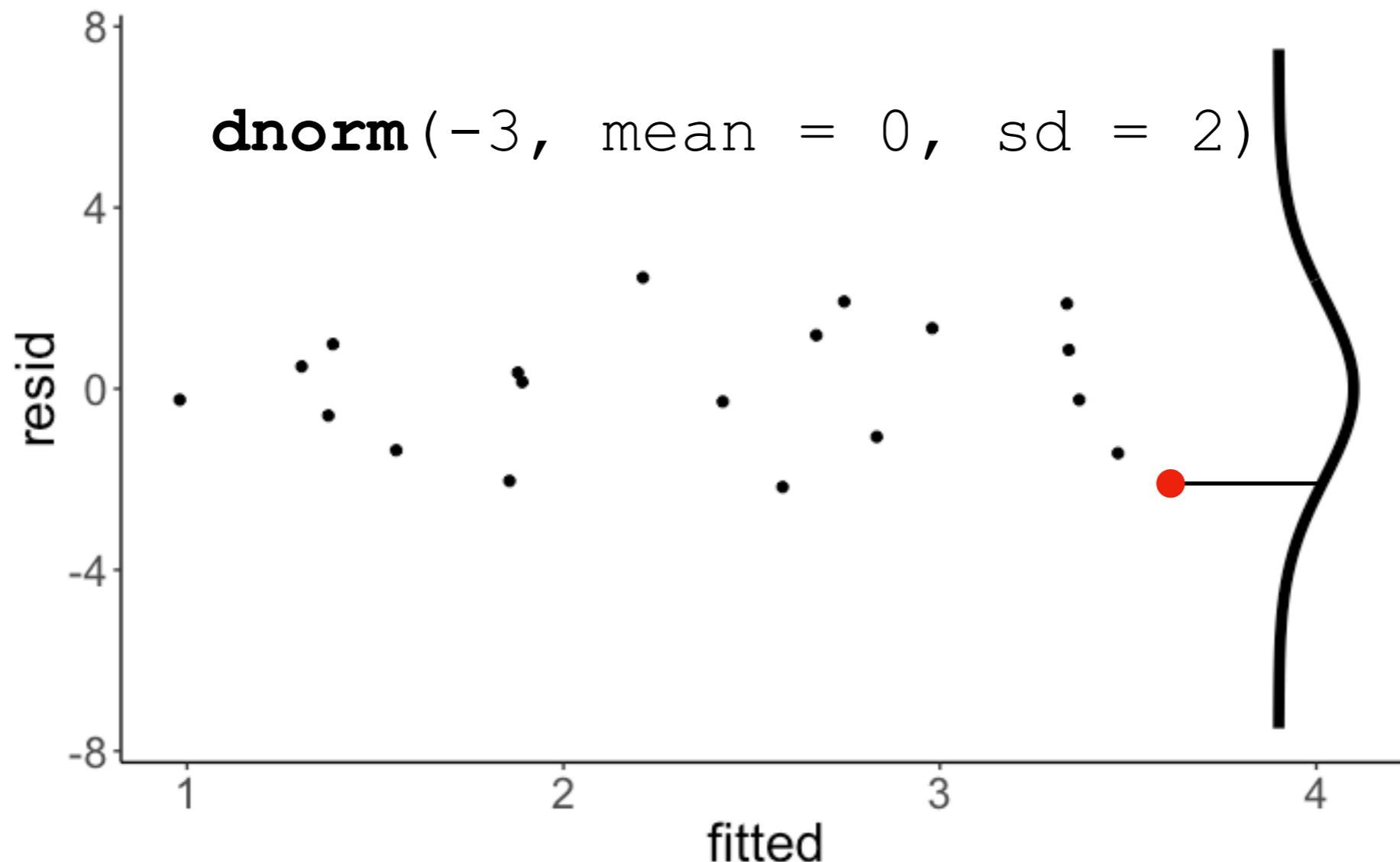
AIC and BIC

Residual plot



AIC and BIC

Residual plot



since the data points are independent, we can calculate the overall likelihood by multiplying the likelihood of each observation

AIC and BIC

```

1 # generate some data
2 df.like = tibble(
3   x = runif(20, min = 0, max = 1),
4   y = 1 + 3 * x + rnorm(20, sd = 2)
5 )
6
7 # fit the model
8 fit = lm(formula = y ~ x,
9           data = df.like)
10
11 # model summary
12 fit %>%
13   glance()

```

`dnorm(4.20, mean = 0, sd = 2.15) = 0.02`

x	y	fitted	resid	likelihood
0.41	5.74	1.53	4.20	0.02
0.91	4.86	4.80	0.06	0.18
0.29	0.98	0.80	0.18	0.18
0.46	0.71	1.87	-1.16	0.16
0.33	-0.34	1.05	-1.39	0.15
0.65	0.82	3.11	-2.29	0.11
0.26	-1.35	0.57	-1.92	0.12
0.48	4.75	2.00	2.75	0.08
0.77	4.96	3.86	1.11	0.16
0.08	0.80	-0.55	1.35	0.15

inferred standard deviation of the error

$$\sum_{i=1}^n \ln(\text{likelihood})$$

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
0.38	0.35	2.17	11.2	0	2	-42.78	91.56	94.55	84.43	18

$e \sim \mathcal{N}(\text{mean} = 0, \text{sd} = 2.17)$

AIC and BIC

- for both AIC and BIC, *lower* is better!
- neither provide a test of a model in the sense of testing a null hypothesis
 - AIC or BIC tell us nothing about the absolute quality of a model, only the quality relative to other models
- The BIC generally penalizes free parameters more strongly than the Akaike information criterion, though it depends on the size of n and relative magnitude of n and k .

ΔBIC	Evidence against higher BIC
0 to 2	Not worth more than a bare mention
2 to 6	Positive
6 to 10	Strong
>10	Very Strong

What shall I use when?

- Use it all!
- ideally, the different measures provide converging evidence

Table 2

Summary of the model results. Values for r and RMSE indicate means (with 5% and 95% quantiles in parentheses) based on 100 split-half cross-validation runs. BIC scores are based on running the models on the full data set.

Model	r	RMSE	BIC
Difference & pivotality	.86 (.66, .95)	10.56 (6.17, 17.21)	158.59
Difference	.70 (.30, .90)	26.92 (16.4, 40.6)	209.74
Pivotality	.63 (.41, .77)	14.23 (11.39, 17.54)	199.53
Optimality	.66 (.42, .84)	14.55 (10.54, 17.91)	199.47

Note: BIC = Bayesian Information Criterion (lower values indicate better model performance).

Summary

- Quick power review
- Determining sample size
 - use `library("pwr")`
 - write your own simulation
- Model comparison
 - Cross-validation
 - different methods
 - the power of `library("purrr")`
 - AIC and BIC
 - trading off maximum likelihood and model complexity
 - least squares and maximum likelihood

Feedback

How was the pace of today's class?

much a little just a little much
too too right too too
slow slow

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!