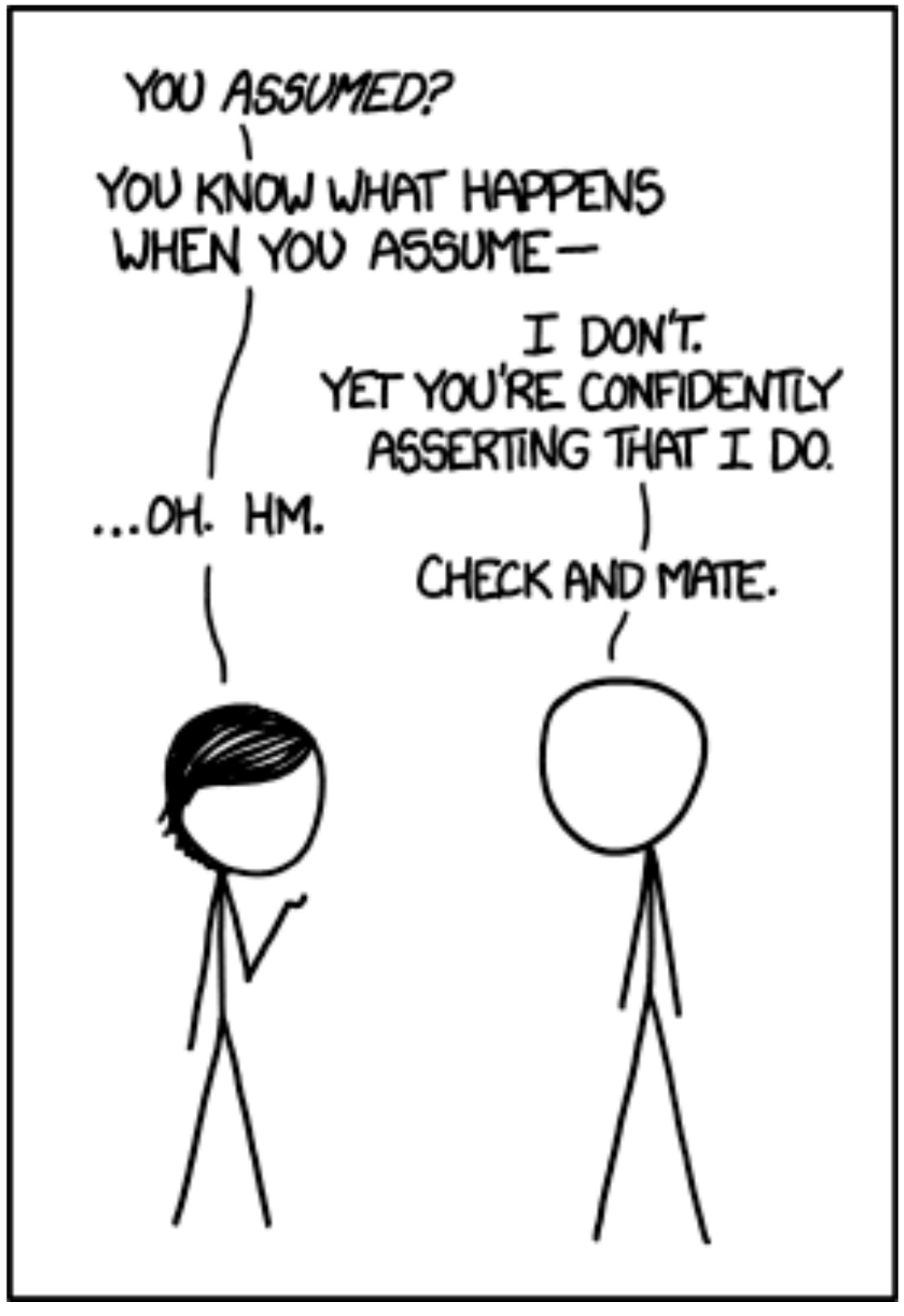
25 Model assumptions and reporting



03/09/2020

Logistics

Zoom rules

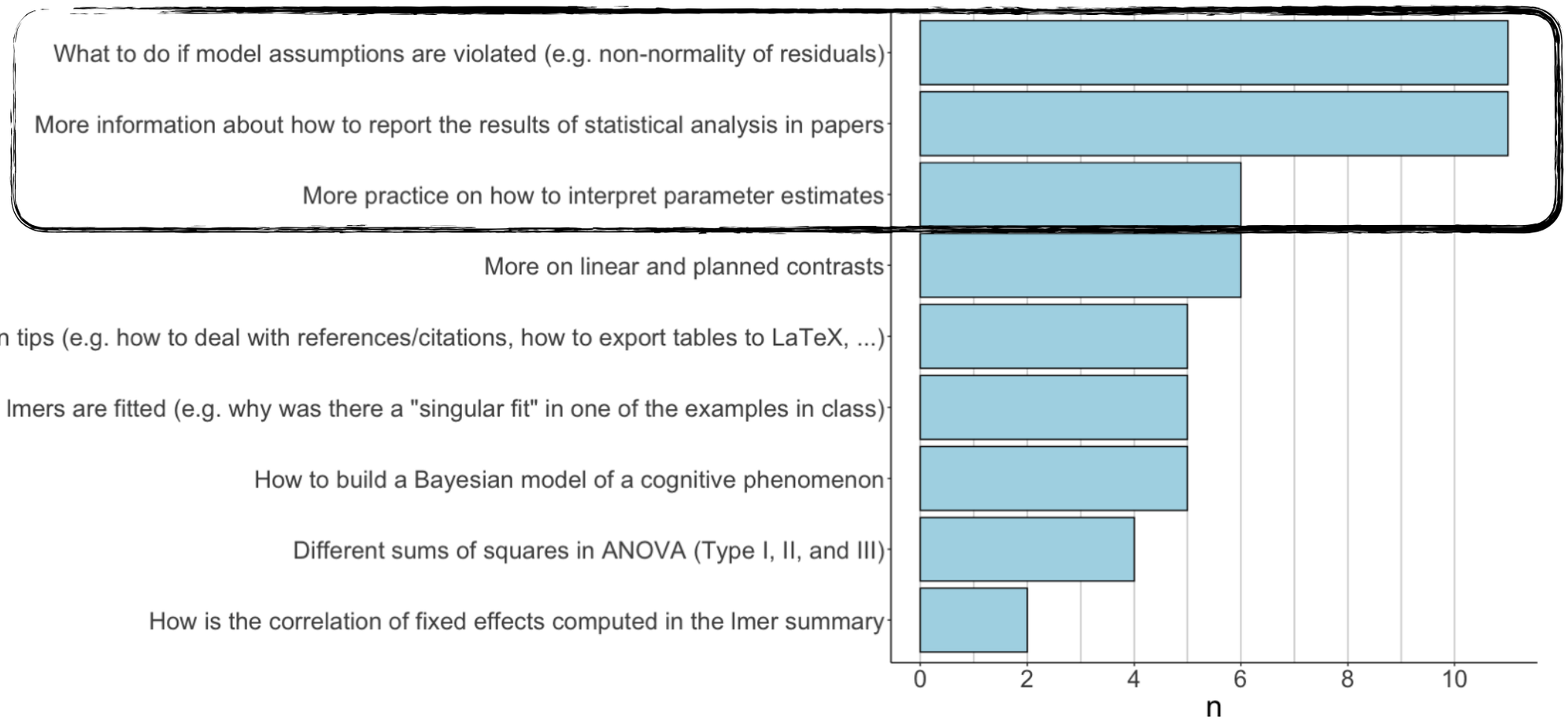
- keep your microphone muted
- ask questions via the chat
- Andrew Nam will read out your questions

Homework 6


How many hours did you spend on hw3?
number of students
0
1
2
3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
>15
yay, this wasn't too long ...

Choose your own adventure

focus of today



Things that came up

Nice resource!

Pronunciation of `lmer`/`glmer`/etc.

- `lmer`: I have heard “ell emm ee arr” (i.e. pronouncing each letter); “elmer” (probably most common); and “lemur”
- `glmer`: “gee ell emm ee arr”, “gee elmer”, “glimmer”, or “gleamer”
- for `lme` and `nlme` people just seem to spell out the names (rather than saying e.g. “lemmy” and “nelmy”)

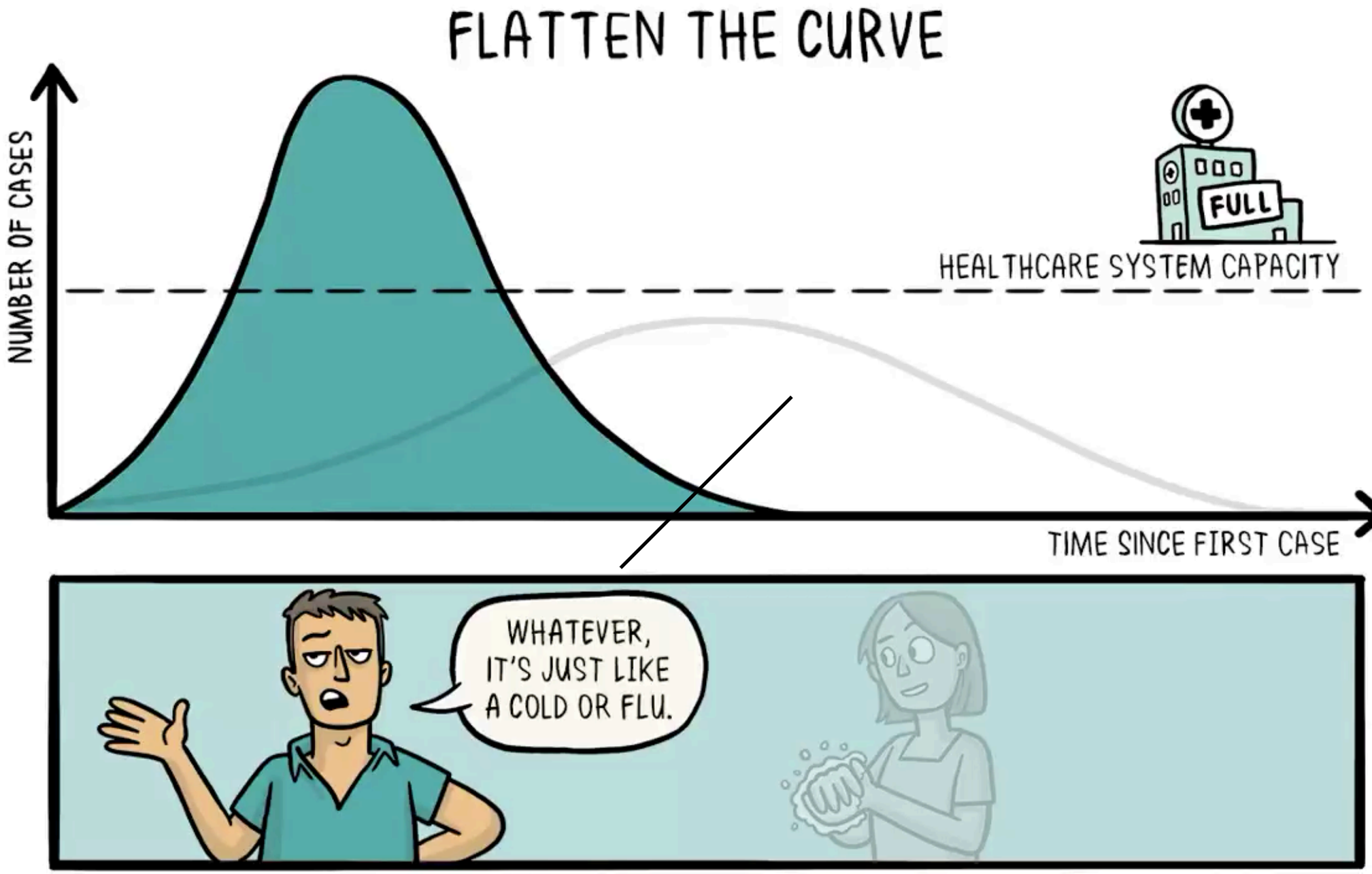
GLMM FAQ

Ben Bolker and others

09 Jan 2020

https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html

Covid-19


FLATTEN THE CURVE
NUMBER OF CASES
FULL
HEALTHCARE SYSTEM CAPACITY
TIME SINCE FIRST CASE
WHATEVER, IT'S JUST LIKE A COLD OR FLU.
@SIOUXSIEW @XTOTL @THESPINOFFTV
'ADAPTED FROM THOMAS SPLETTSTÖßER (@SPLETTE) AND THE CDC'

Covid-19

Adjusted age-specific case fatality ratio during the COVID-19 epidemic in Hubei, China, January and February 2020

Julien Riou (julien.riou@ispm.unibe.ch), Anthony Hauser (anthony.hauser@ispm.unibe.ch), Michel J. Counotte (michel.counotte@ispm.unibe.ch) and Christian L. Althaus (christian.althaus@ispm.unibe.ch)

Institute of Social and Preventive Medicine, University of Bern, Switzerland

Abstract. The coronavirus disease 2019 (COVID-19) epidemic that originated in Wuhan, China has spread to more than 60 countries. We estimated the age-specific case fatality ratio (CFR) by fitting a transmission model to data from China, accounting for underreporting of cases and the time delay to death. Overall CFR among all infections was 1.6% (1.4-1.8%) and increased considerable for the elderly, highlighting the expected burden for populations with further expansion of the COVID-19 epidemic around the globe.

epidemiologist

model the age-specific case fatality ratio using

stan

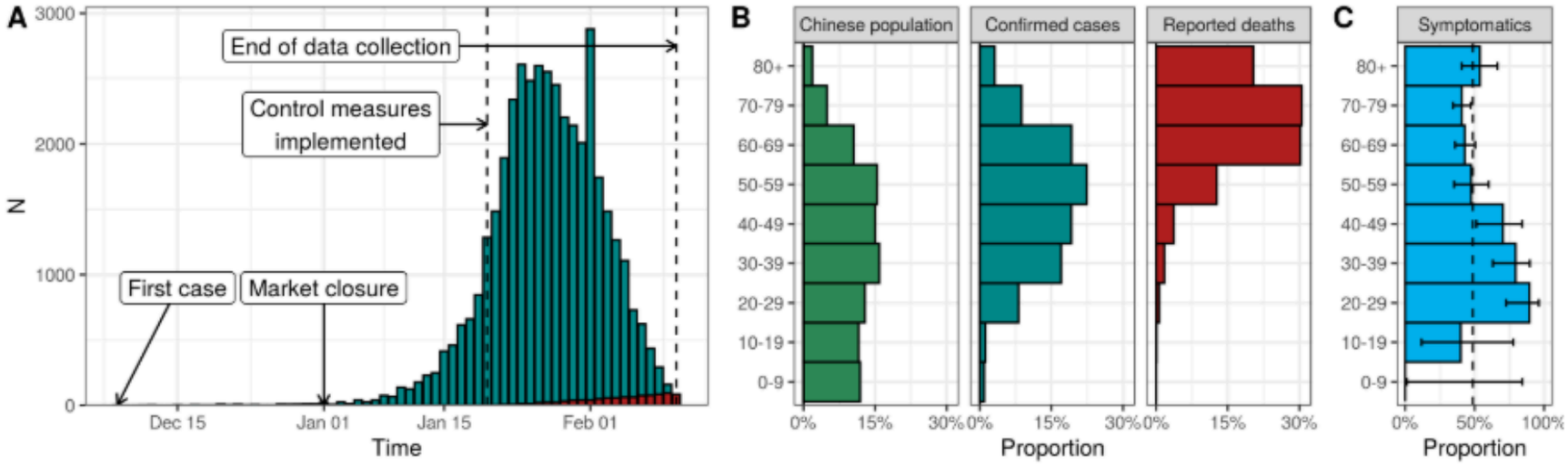
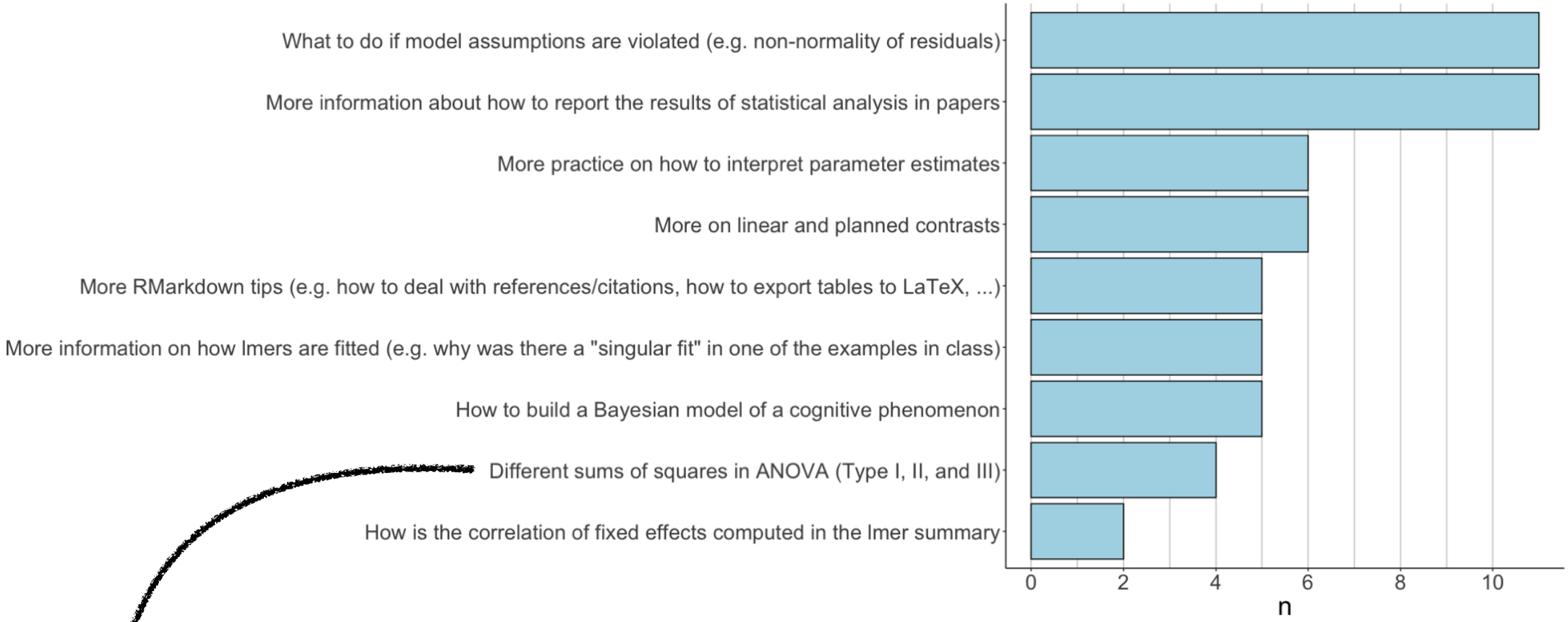


Figure. (A) Reported confirmed cases of COVID-19 in the Hubei province by date of disease onset (blue) and reported deaths (red) from 8 December 2019 to 11 February 2020. (B) Age distribution of the Chinese population compared to that of confirmed cases of and deaths due to COVID-19. (C) Proportion of individuals infected by COVID-19 showing symptoms among passengers of the Diamond Princess ship (with 95% credible interval).

https://github.com/jriou/covid_adjusted_cfr

Sums of squares in ANOVA



https://stats.stackexchange.com/questions/20452/how-to-interpret-type-i-type-ii-and-type-iii-anova-and-manova

https://stats.stackexchange.com/questions/13241/the-order-of-variables-in-anova-matters-doesnt-it

some useful posts

Plan for today

- What we've learned
- What should I do now?
- What to do when assumptions are violated?
- How to report statistical results?

What we've learned

Learning goals

What you will learn

You will learn how to **use R** to …

- read, wrangle, and analyze data
- make publication-ready plots

Understand the philosophy behind null **hypothesis significance testing (NHST)** and **Bayesian statistics** through …

- running computer simulations and visualizing the results

Formulate **research questions as statistical models** and …

- determine which models work for different situations
- check that the model's assumptions are met, how much it matters, and what to do if assumptions aren't met

Communicate what you have learned about your data …

- in short presentations in class, showcasing your visualization and analysis
- in written reports

Contribute to open and **reproducible science** through …

- adopting good coding practices
- sharing your data and research reports online

What will we learn?

Weeks 1-3 **1. Use R!**

- Data visualization
- Data manipulation/wrangling
- Understand key statistical concepts
  - Simulation, manipulation, visualization

2. Build models

Weeks 4-7
- Formulate hypotheses as statistical models

Weeks 8-9
- Bayesian statistics

3. Report results

all the time (& Week 10)
- Reproducible research

What we've covered

data wrangling and visualization

probability, causality, simulation

linear model

power analysis

model comparison

linear mixed effects models

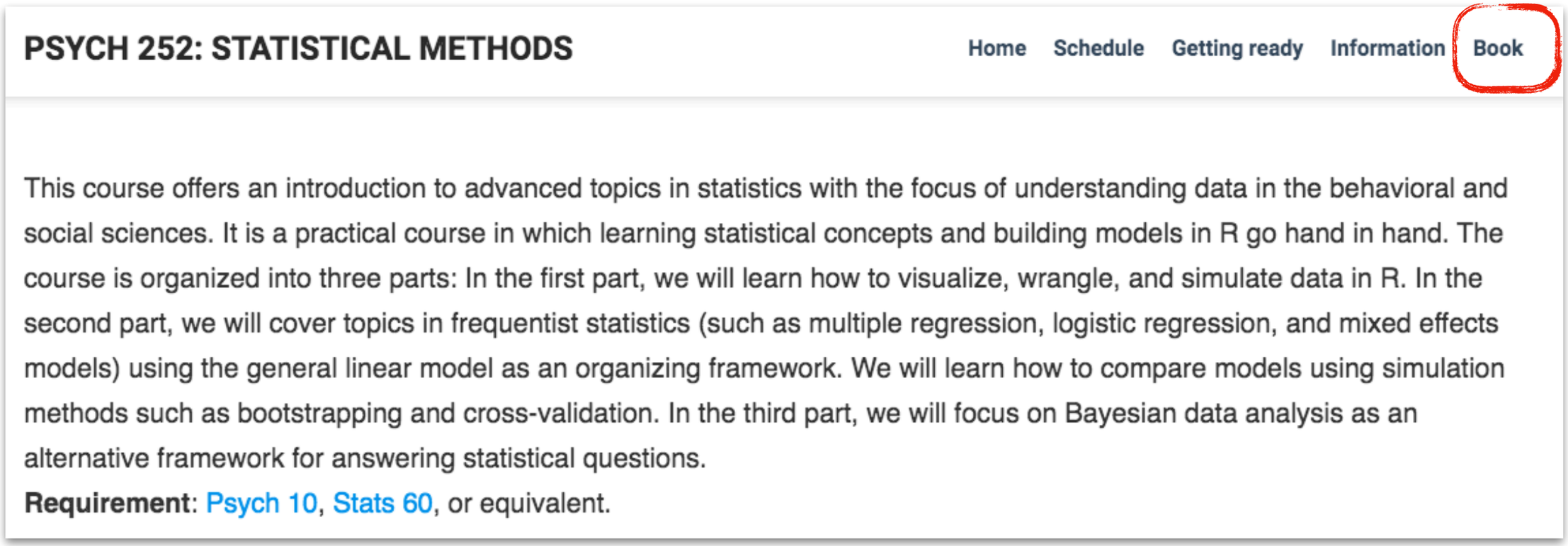
logistic regression

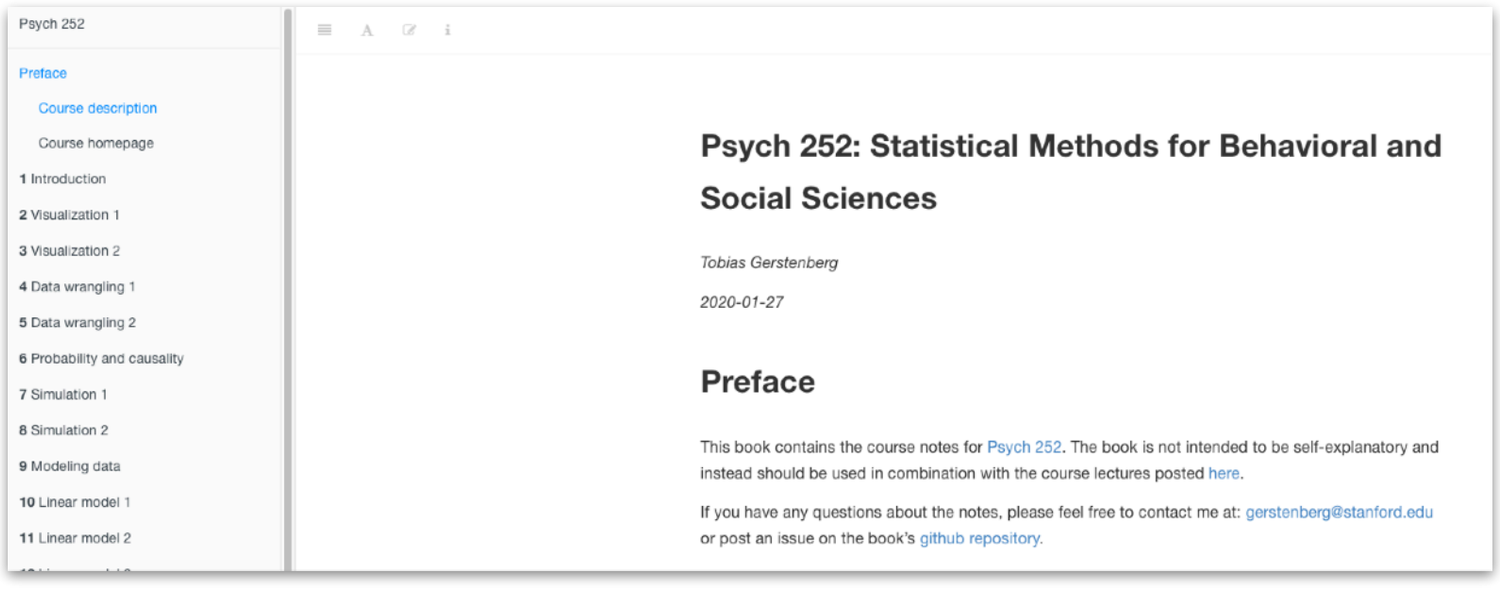
Bayesian data analysis

model assumptions and reporting

Day	Date	Topic
Monday	January 6th	Introduction
Wednesday	January 8th	Visualization I
Friday	January 10th	Visualization II
Monday	January 13th	Data wrangling I
Tuesday	January 14th	Homework section: Visualization
Wednesday	January 15th	Data wrangling II
Thursday	January 16th	Application section: Visualization & Data wrangling
Friday	January 17th	Probability
Monday	January 20th	No class (Martin Luther King Jr. Day)
Tuesday	January 21st	Homework section: Data wrangling
Wednesday	January 22nd	Simulation I
Thursday	January 23rd	Application section: Probability & Simulation
Friday	January 24th	Simulation II
Monday	January 27th	Modeling data
Tuesday	January 28th	Homework section: Simulation
Wednesday	January 29th	Linear model I
Thursday	January 30th	Application section: Modeling data
Friday	January 31st	Linear model II
Monday	February 3rd	Linear model III
Tuesday	February 4th	Homework section: Linear model
Wednesday	February 5th	Linear model IV
Thursday	February 6th	Application section: Linear model
Friday	February 7th	Power analysis
Monday	February 10th	Mediation and moderation
Wednesday	February 12th	No class
Thursday	February 13th	Midterm due
Friday	February 14th	Model comparison
Monday	February 17th	No class (Presidents' Day)
Tuesday	February 18th	Homework section: Project proposals
Wednesday	February 19th	Linear mixed effects models I
Thursday	February 20th	Application section: Model comparison
		Project proposal due
Friday	February 21st	Linear mixed effects models II
Monday	February 24th	Linear mixed effects models III
Tuesday	February 25th	Homework section: Model comparison
Wednesday	February 26th	Generalized linear model
Thursday	February 27th	Application section: Linear mixed effects models
Friday	February 28th	Bayesian data analysis I
Monday	March 2nd	Bayesian data analysis II
Tuesday	March 3rd	Homework section: Linear mixed effects models
Wednesday	March 4th	Bayesian data analysis III
Thursday	March 5th	Application section: Bayesian data analysis
Friday	March 6th	Bayesian data analysis IV
Monday	March 9th	Course summary and outlook
Tuesday	March 10th	Homework section: Bayesian data analysis & Final projects
Wednesday	March 11th	Guest lecture: Johannes Eichstaedt
Friday	March 13th	Guest lecture: TAs in action
Wednesday	March 18th	Final project presentations
Friday	March 20th	Final project report due

I'll keep updating the course notes!





https://psych252.github.io/

What shall I do now?

Psych 253

Basics / Course Schedule & Syllabus / Software Tools

High-Dimensional Methods for Behavioral and Neural Data
Spring 2019, Stanford University

Introduction to high-dimensional data analysis and machine learning methods for use in the behavioral and neurosciences, including: supervised methods such as SVMs and logistic classification, linear, nonlinear and multi-level regression, and their associated feature selection and regularization techniques; unsupervised methods such as dimension reduction, factor analysis, manifold learning, and clustering; statistical methods such as signal detection, bootstrapping, and reliability theory; metrics for model/data comparison such as representational similarity analysis; and general model parameter estimation approaches using first and second-order optimization methods. The course will involve analysis of several real-world large-scale behavioral and neural datasets. Students will learn how to both use existing models from Python-based statistical data analysis packages (such as pandas, numpy, scipy, and scikit-learn) as well to build, optimize, and estimate their own custom models using the Tensorflow framework.

Time: Mon./Wed. 1:30p - 2:50p

- focus on high-dimensional data
- classification
- representational similarity analysis
- clustering ...
- learn Python!

Dan Yamins

Russ Poldrack

http://web.stanford.edu/class/cs109/schedule.html

Johannes Eichstaedt's class

Human-Centered Artificial Intelligence
Stanford University

Home About People Research Courses News & Insights Events AI Index Contact

People

- Leadership
- Advisory Council
- Distinguished Fellows
- Fellows
- Faculty
- Staff

Johannes Eichstaedt

Johannes C. Eichstaedt is a computational social scientist and has just joined the Psychology department at Stanford as faculty and HAI as a Junior Fellow. Johannes obtained his Ph.D. in Psychology and has been a Senior Research Associate at the Positive Psychology Center at the University of Pennsylvania. In 2011 he co-founded and led the World Well-Being Project, bringing together computer scientists and psychologists, which has since attracted $3.9m in funding. Before joining the social sciences, Johannes did research in particle physics with an M.S. from the University of Chicago. In 2014, he was elected an Emerging Leader in Science & Society by the American Association for Advancement of Science (AAAS). In his non-academic time he practices Tai Chi and goes on long-distance hikes.

Dr. Eichstaedt's Homepage

CS109: Probability for Computer Scientists

The class starts by providing a fundamental grounding in combinatorics, and then quickly moves into the basics of probability theory. We will then cover many essential concepts in probability theory, including particular probability distributions, properties of probabilities, and mathematical tools for analyzing probabilities. Finally, the last third of the class will focus on data analysis and Machine Learning as a means for seeing direct applications of probability in this exciting and quickly growing subfield of computer science.

- learn more about probability theory through programming
- gain a deeper understanding of the fundamental underlying concepts

http://web.stanford.edu/class/cs109/schedule.html

Advanced regression analysis

EDUC 326: Advanced Regression Analysis

Social science researchers often deal with complex data and research questions that traditional statistics models like linear regression cannot adequately address. This course offers the opportunity to understand and apply two widely used types of advanced regression analysis that allow the examination of 1) multilevel data structures (multilevel models) and 2) multivariate research questions (structural equation models).

Terms: Spr | Units: 3-4 | Grading: Letter or Credit/No Credit

Instructors: Smith, S. (PI)

Schedule for EDUC 326

- multilevel models
- structural equation models

Data Challenge Lab

Where students develop their data skills by solving a progression of increasingly difficult challenges

ENGR 150: Data Challenge Lab
Terms: Win, Spr | Units: 5
Instructors: Bill Behrman, Hadley Wickham

Prof. Tidyverse

https://datalab.stanford.edu/challenge-lab

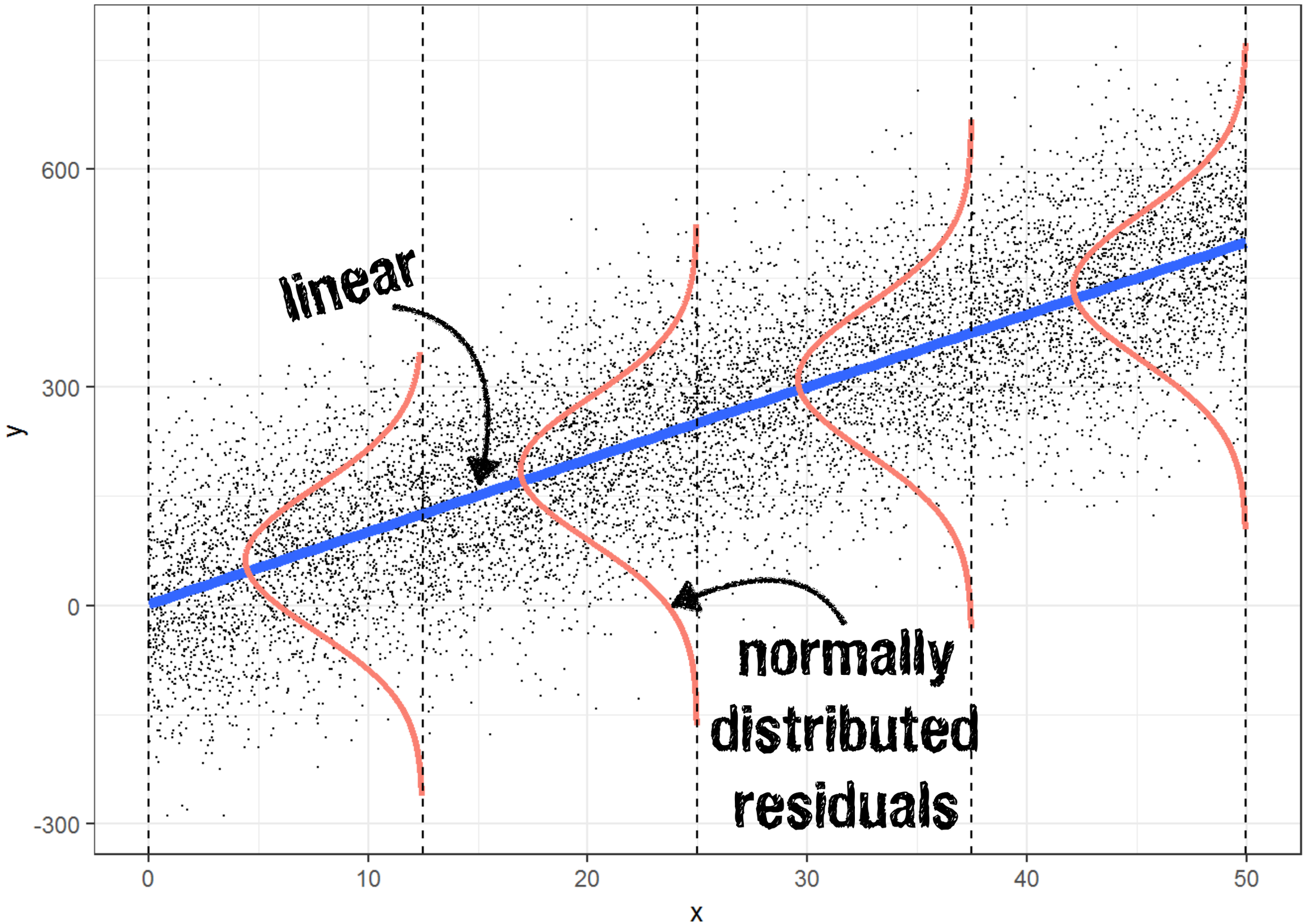
What to do when assumptions are violated?

All models are wrong, but some are useful – George Box

Linear regression assumptions

- **variable type**
  - dependent variable: quantitative, continuous, unbounded (ideally)
  - independent variables: quantitative or categorical
- **non-zero variance**
  - predictors need to vary
- **no perfect multicollinearity**
  - no perfect linear relationship between two or more predictors
  - predictor variables shouldn't correlated too highly
- **linear and additive**
  - the mean values of the outcome variable for each increment of the predictor(s) lie along a straight line
- **independence of residuals**
- **independence of the outcome variable**
  - all the values of the outcome variables are independent from each other
- **homogeneity of variance**
  - at each level of the predictor variable, the variance of the residuals should be constant
- **normally distributed residuals**

Linear regression assumptions



What when assumptions are violated?

- Influential data points
- Linear and additive
- Data transformation
- Independence
- Non-parametric analysis
- Simulation methods

What when assumptions are violated?

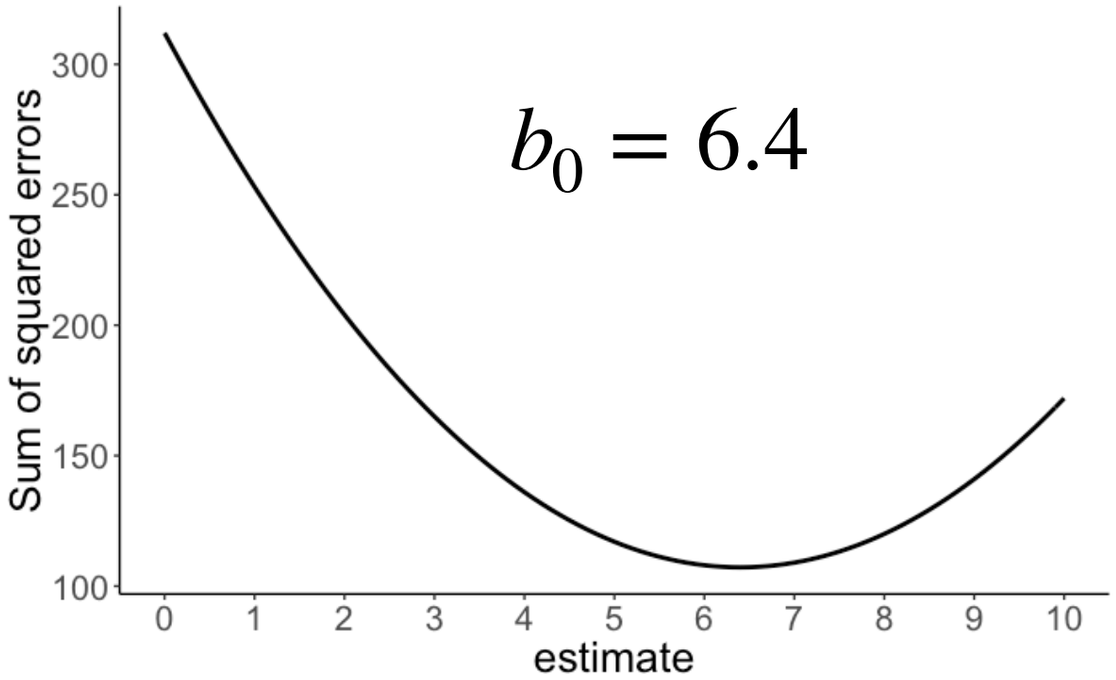
- **Influential data points**
- Linear and additive
- Data transformation
- Independence
- Non-parametric analysis
- Simulation methods

Influential data points

flash from the past

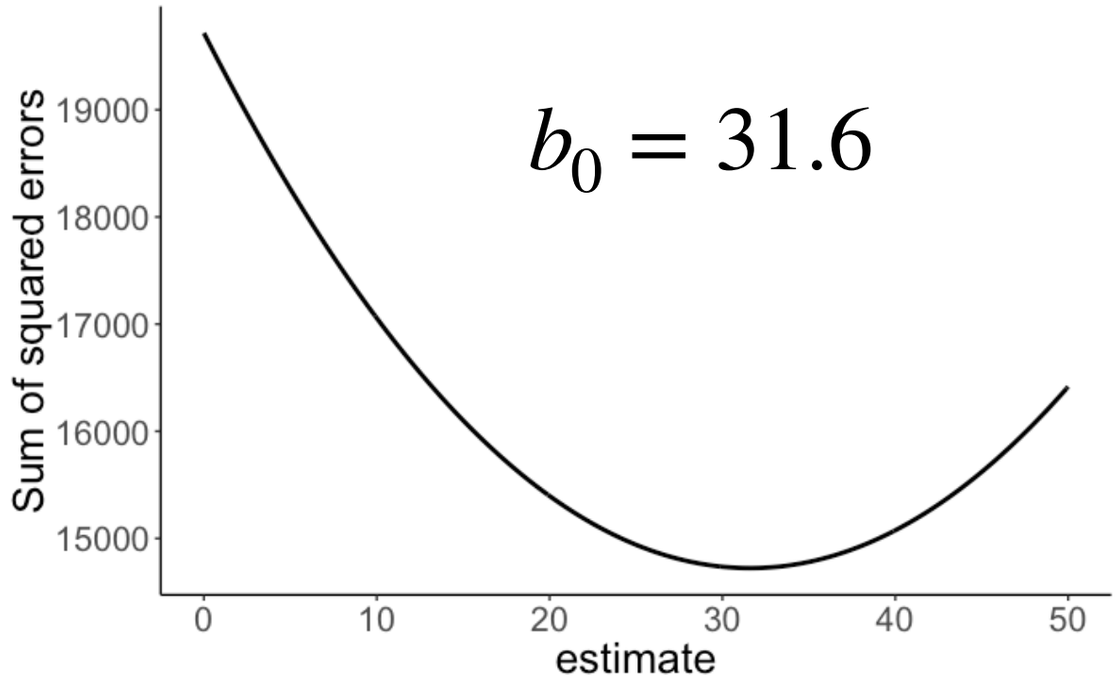
$$Y_i = b_0 + e_i \qquad \text{ERROR} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Y_i - b_0)^2$$

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



$b_0 = 6.4$

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



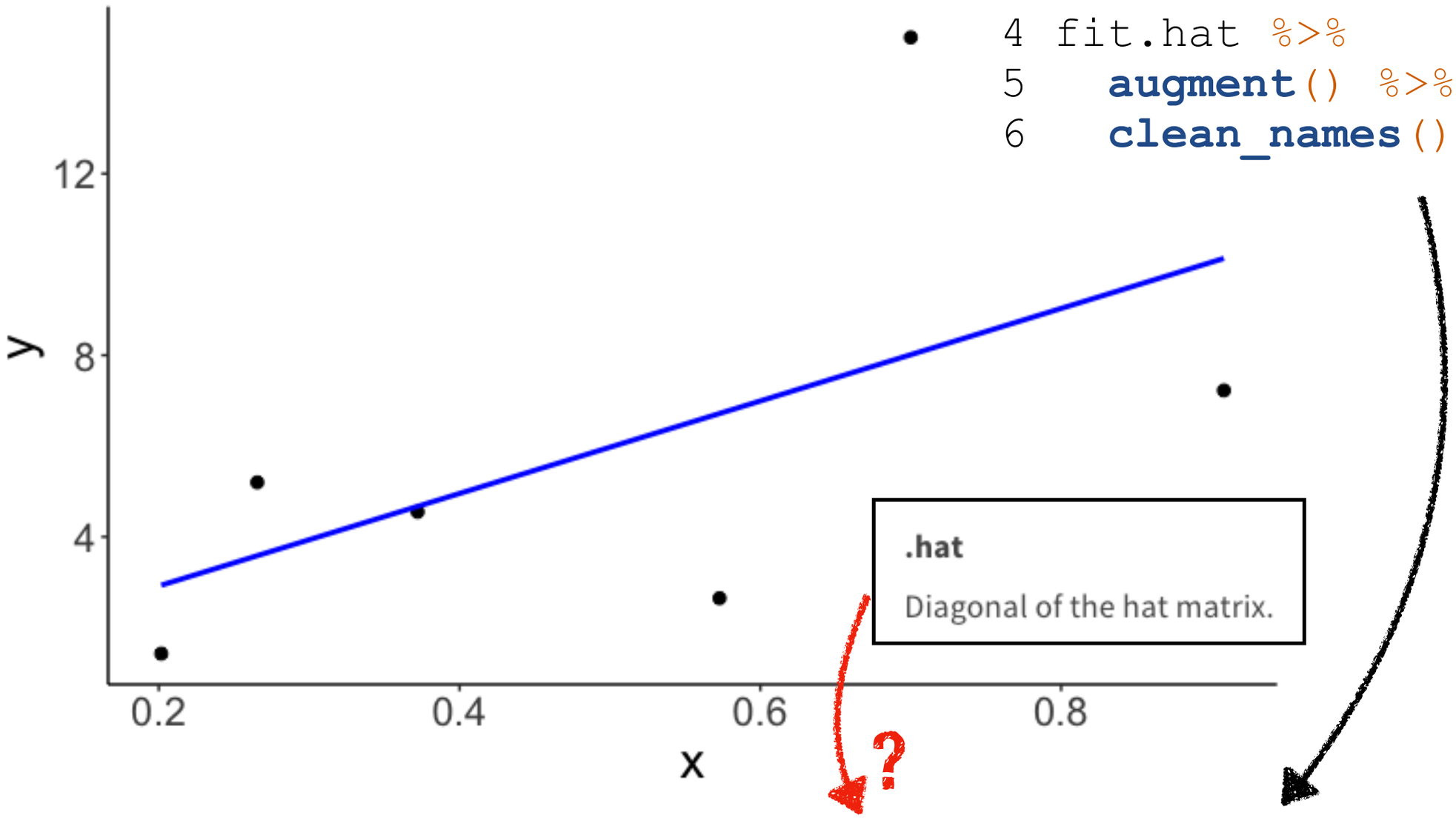
$b_0 = 31.6$

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

is strongly affected by outliers!

y-hat

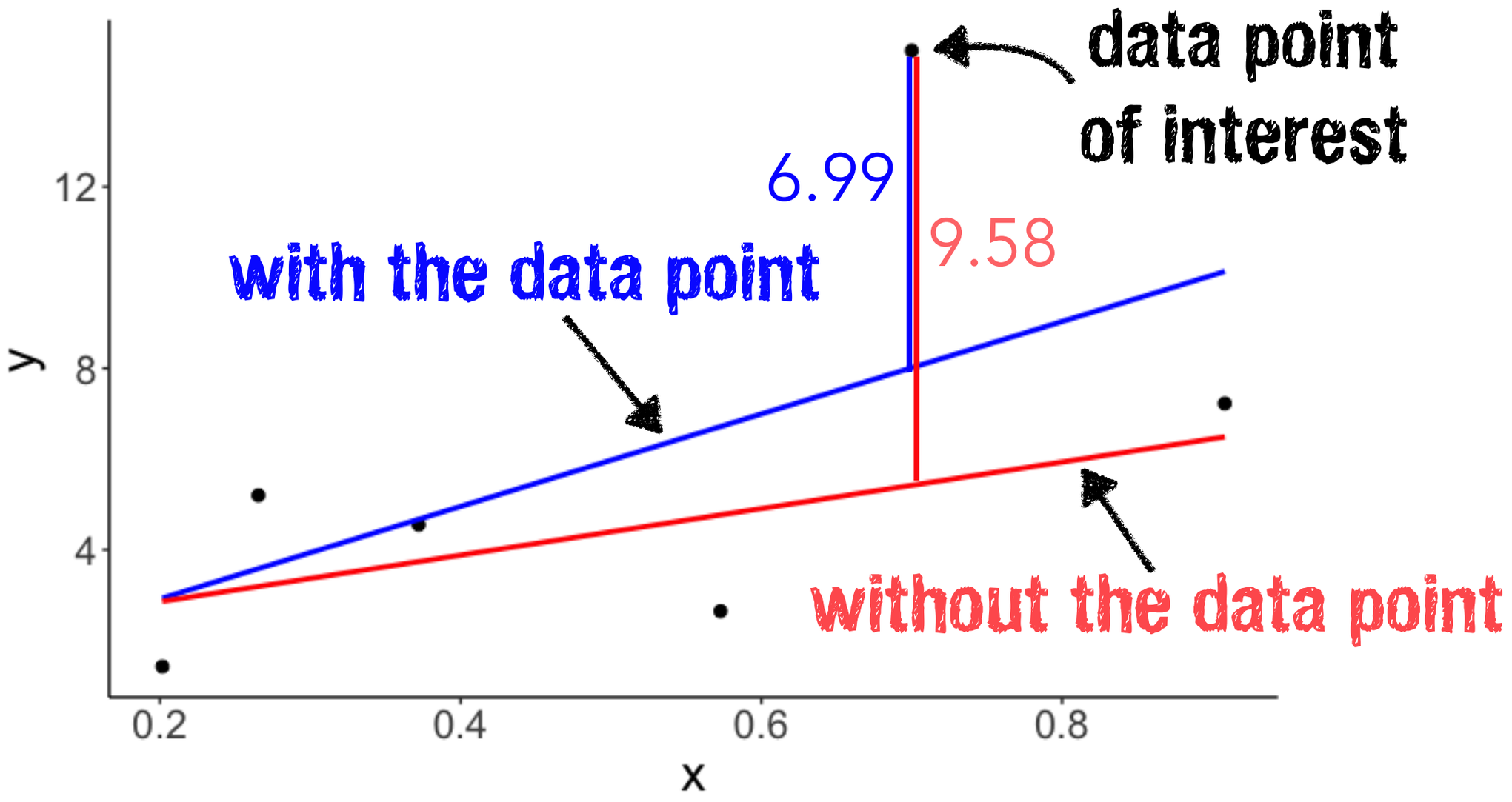
```
1 fit.hat = lm(formula = y ~ x,
2                data = df.hat)
3
4 fit.hat %>%
5   augment() %>%
6   clean_names()
```



y	x	fitted	se_fit	resid	hat	sigma	cooksd	std_resid
5.20	0.27	3.58	2.50	1.62	0.32	5.00	0.05	0.44
4.55	0.37	4.67	2.05	-0.12	0.21	5.12	0.00	-0.03
2.65	0.57	6.72	1.88	-4.07	0.18	4.42	0.11	-1.01
7.22	0.91	10.13	3.46	-2.91	0.61	4.37	0.84	-1.05
1.43	0.20	2.93	2.85	-1.51	0.41	5.00	0.07	-0.44
15.00	0.70	8.01	2.31	6.99	0.27	1.98	0.63	1.84

y-hat

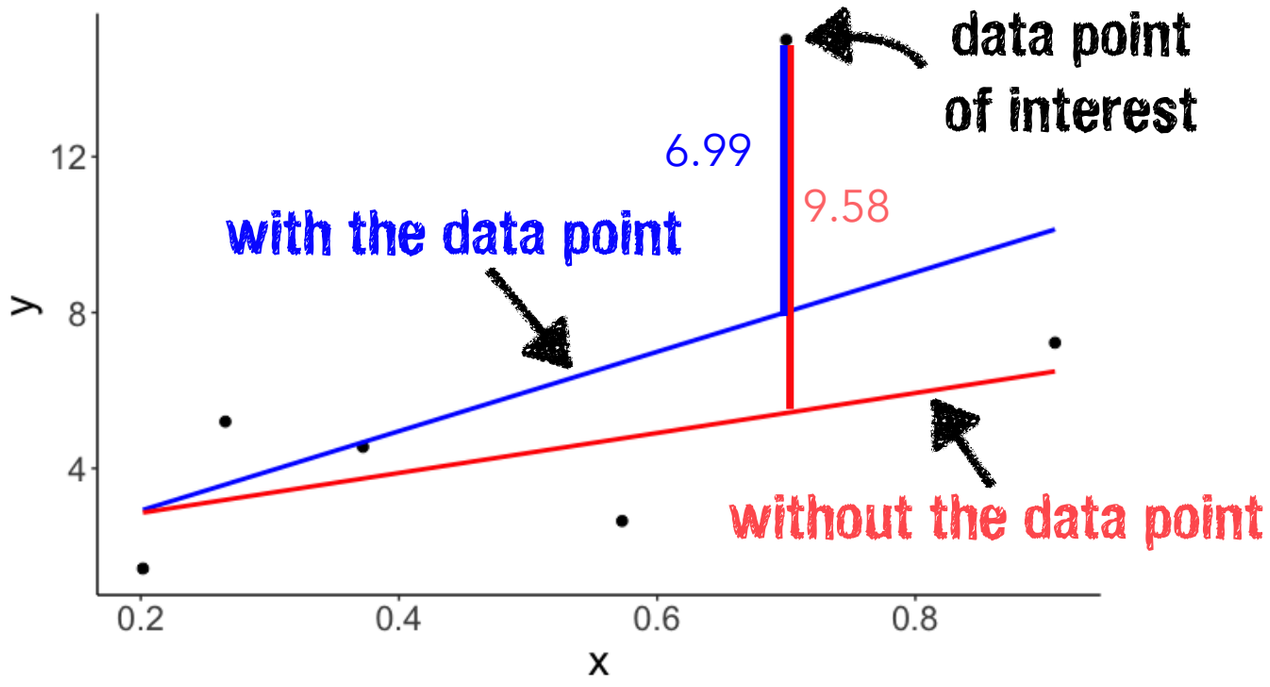
y-hat is a measure of influence: how much of a difference does a data point make for the model parameters?



$$\text{y-hat}_i = 1 - \frac{\text{residual for data point i based on of model fitted with data point i}}{\text{residual for data point i based on of model fitted without data point i}}$$

y-hat

```
 1 fit.hat_with = lm(formula = y ~ x,
 2                   data = df.hat)
 3
 4 fit.hat_without = lm(formula = y ~ x,
 5                      data = df.hat %>%
 6                        filter(index != 6))
 7
 8 residual_without = fit.hat_without %>%
 9   augment(newdata = df.hat) %>%
10   clean_names() %>%
11   mutate(resid = y - fitted) %>%
12   filter(row_number() == 6) %>%
13   pull(resid)
14
15 residual_with = fit.hat %>%
16   augment() %>%
17   clean_names() %>%
18   filter(row_number() == 6) %>%
19   pull(resid)
20
21 hat = 1 - (residual_with/residual_without)
22 hat
```



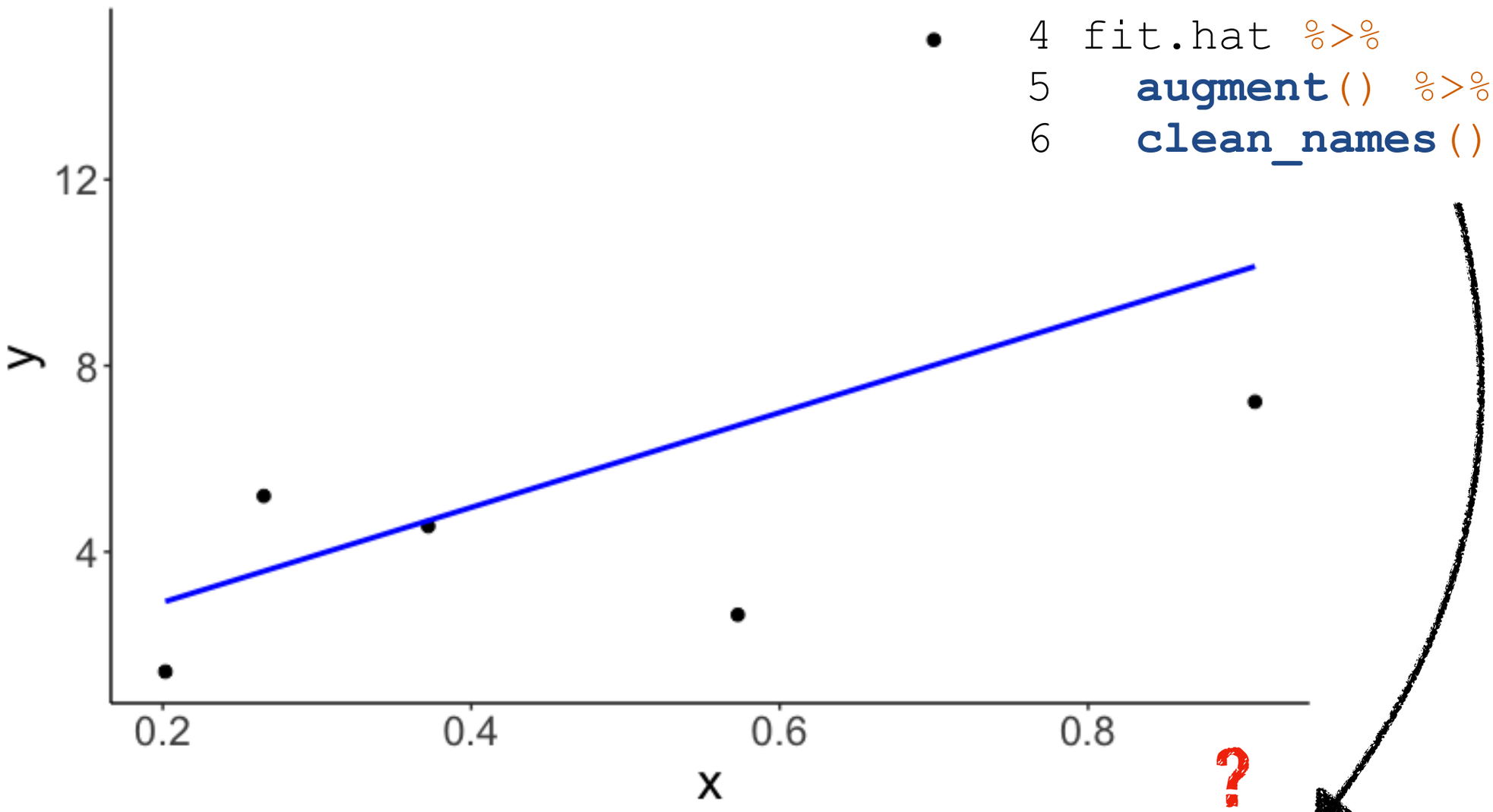
y	x	fitted	se_fit	resid	hat
5.20	0.27	3.58	2.50	1.62	0.32
4.55	0.37	4.67	2.05	-0.12	0.21
2.65	0.57	6.72	1.88	-4.07	0.18
7.22	0.91	10.13	3.46	-2.91	0.61
1.43	0.20	2.93	2.85	-1.51	0.41
15.00	0.70	8.01	2.31	6.99	0.27

0.27

$$\text{y-hat}_i = 1 - \frac{\text{residual for data point i based on of model fitted with data point i}}{\text{residual for data point i based on of model fitted without data point i}}$$

cook's distance

```
1 fit.hat = lm(formula = y ~ x,
2                data = df.hat)
3
4 fit.hat %>%
5   augment() %>%
6   clean_names()
```



y	x	fitted	se_fit	resid	hat	sigma	cooksd	std_resid
5.20	0.27	3.58	2.50	1.62	0.32	5.00	0.05	0.44
4.55	0.37	4.67	2.05	-0.12	0.21	5.12	0.00	-0.03
2.65	0.57	6.72	1.88	-4.07	0.18	4.42	0.11	-1.01
7.22	0.91	10.13	3.46	-2.91	0.61	4.37	0.84	-1.05
1.43	0.20	2.93	2.85	-1.51	0.41	5.00	0.07	-0.44
15.00	0.70	8.01	2.31	6.99	0.27	1.98	0.63	1.84

Cook's distance

squared standardized residual

y-hat

how much of an outlier?

how much influence?

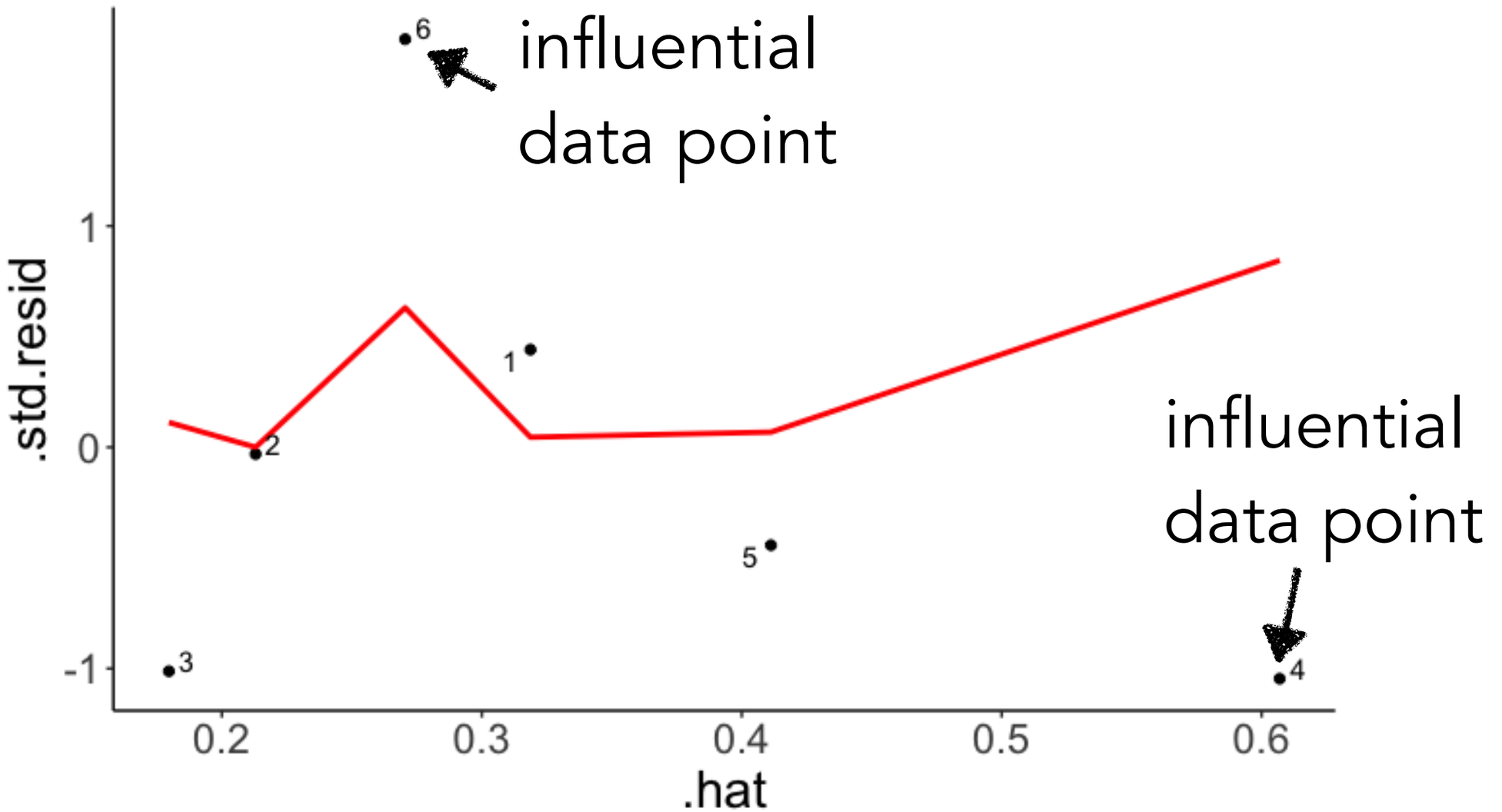
$$D_i = \frac{e_{Si}^2}{k+1} \times \frac{h_i}{1-h_1} = \frac{1.84^2}{1+1} \times \frac{0.27}{1-0.27} = 0.63$$

number of parameters in the model (excluding the intercept)

y	x	fitted	se_fit	resid	hat	cooksd	std_resid
5.20	0.27	3.58	2.50	1.62	0.32	0.05	0.44
4.55	0.37	4.67	2.05	-0.12	0.21	0.00	-0.03
2.65	0.57	6.72	1.88	-4.07	0.18	0.11	-1.01
7.22	0.91	10.13	3.46	-2.91	0.61	0.84	-1.05
1.43	0.20	2.93	2.85	-1.51	0.41	0.07	-0.44
15.00	0.70	8.01	2.31	6.99	0.27	0.63	1.84

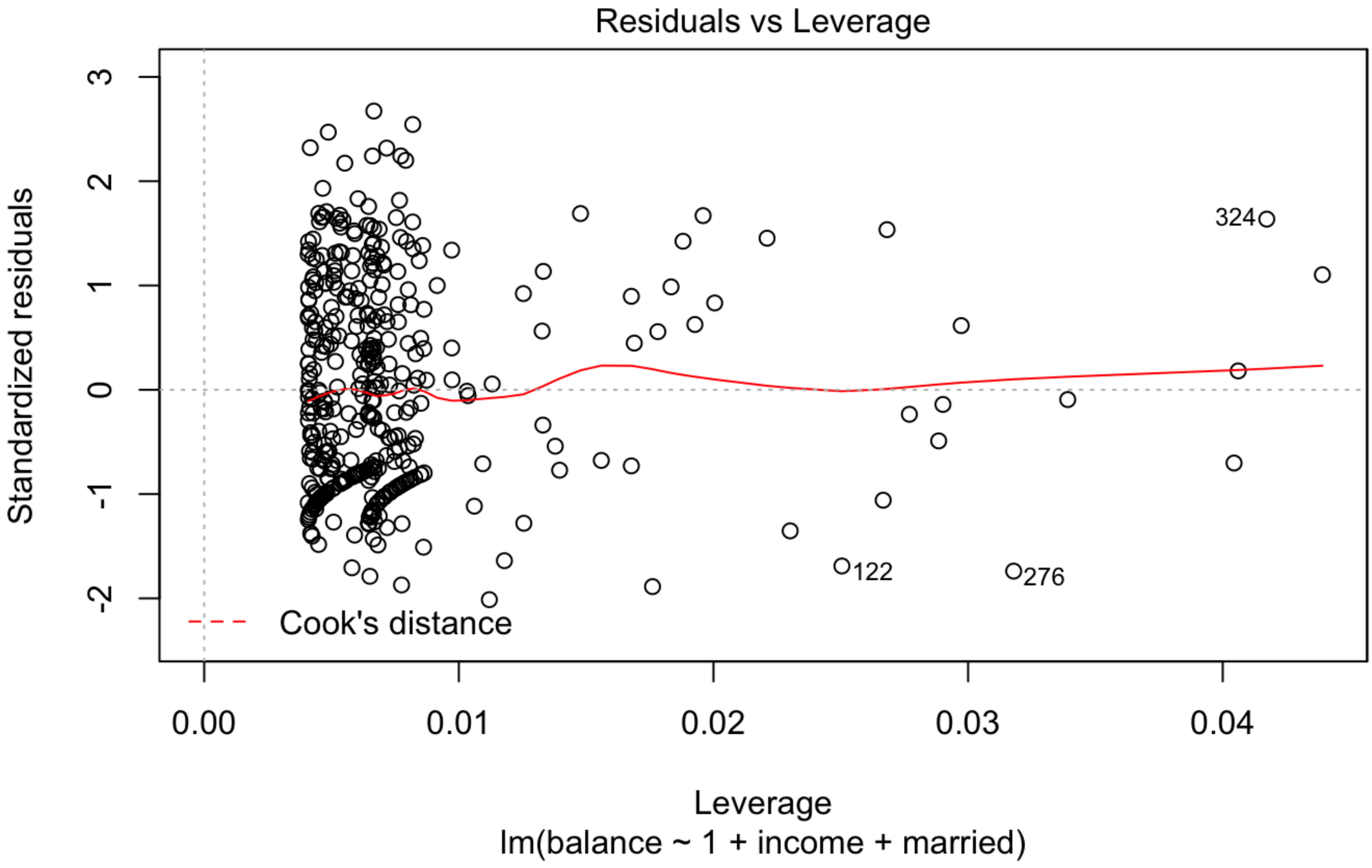
Visualize influential data points

```
1  fit.hat %>%
2    augment() %>%
3    mutate(index = 1:n()) %>%
4    ggplot(aes(x = .hat,
5               y = .std.resid)) +
6    geom_point() +
7    geom_line(aes(y = .cooksd),
8              color = "red",
9              size = 1) +
10   geom_text_repel(mapping = aes(label = index))
```



Visualize influential data points

```
1 fit.hat %>%
2   plot()
```



What when assumptions are violated?

- Influential data points
- **Linear and additive**
- Data transformation
- Independence
- Non-parametric analysis
- Simulation methods

Linear and additive

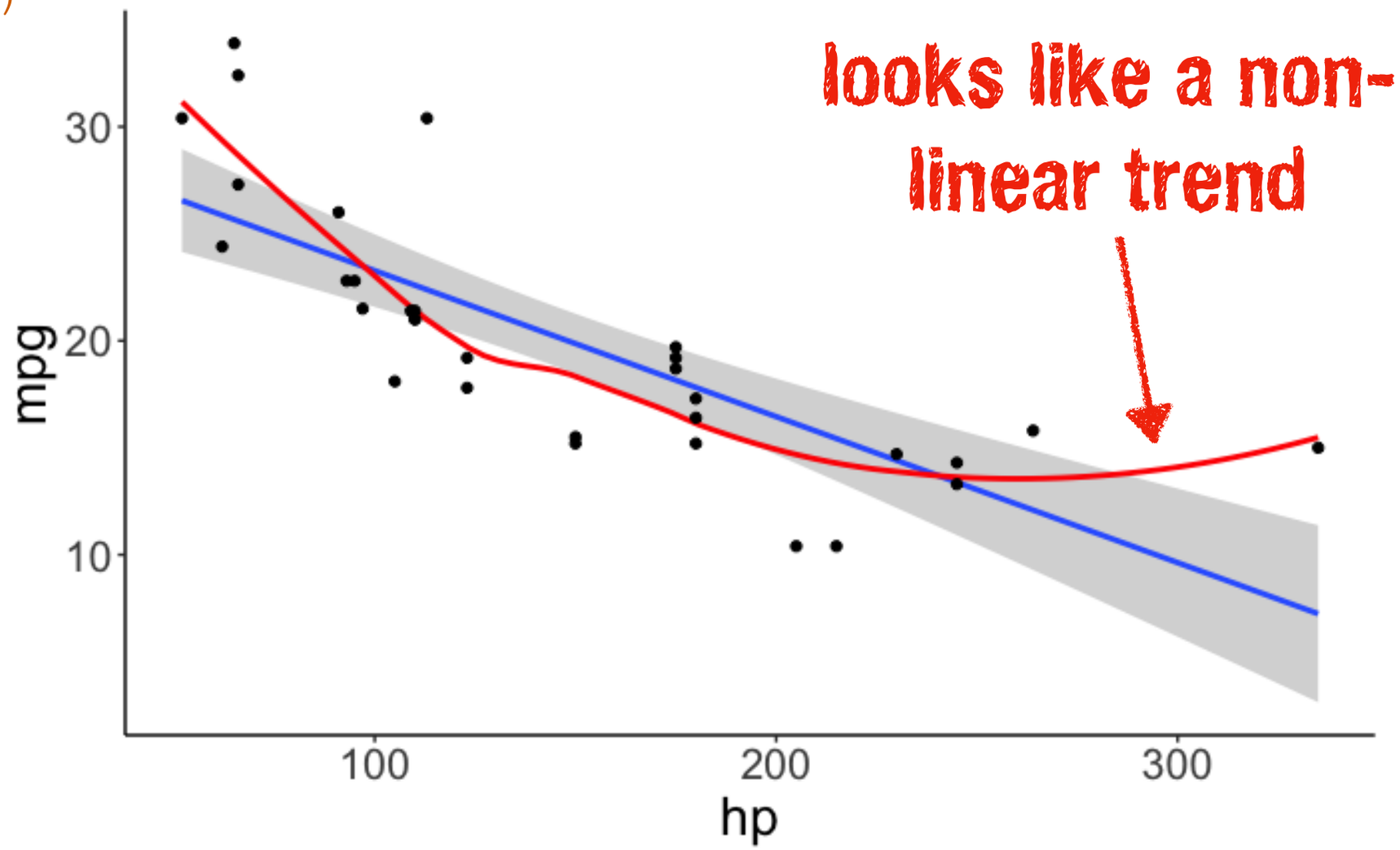
The outcome variable is explained as a linear and additive combination of the predictors.

Can we predict miles per gallon (mpg) with horsepower (hp)?

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	car
Mazda RX4	21.0	6	160	110	3.90	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.88	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.21	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1

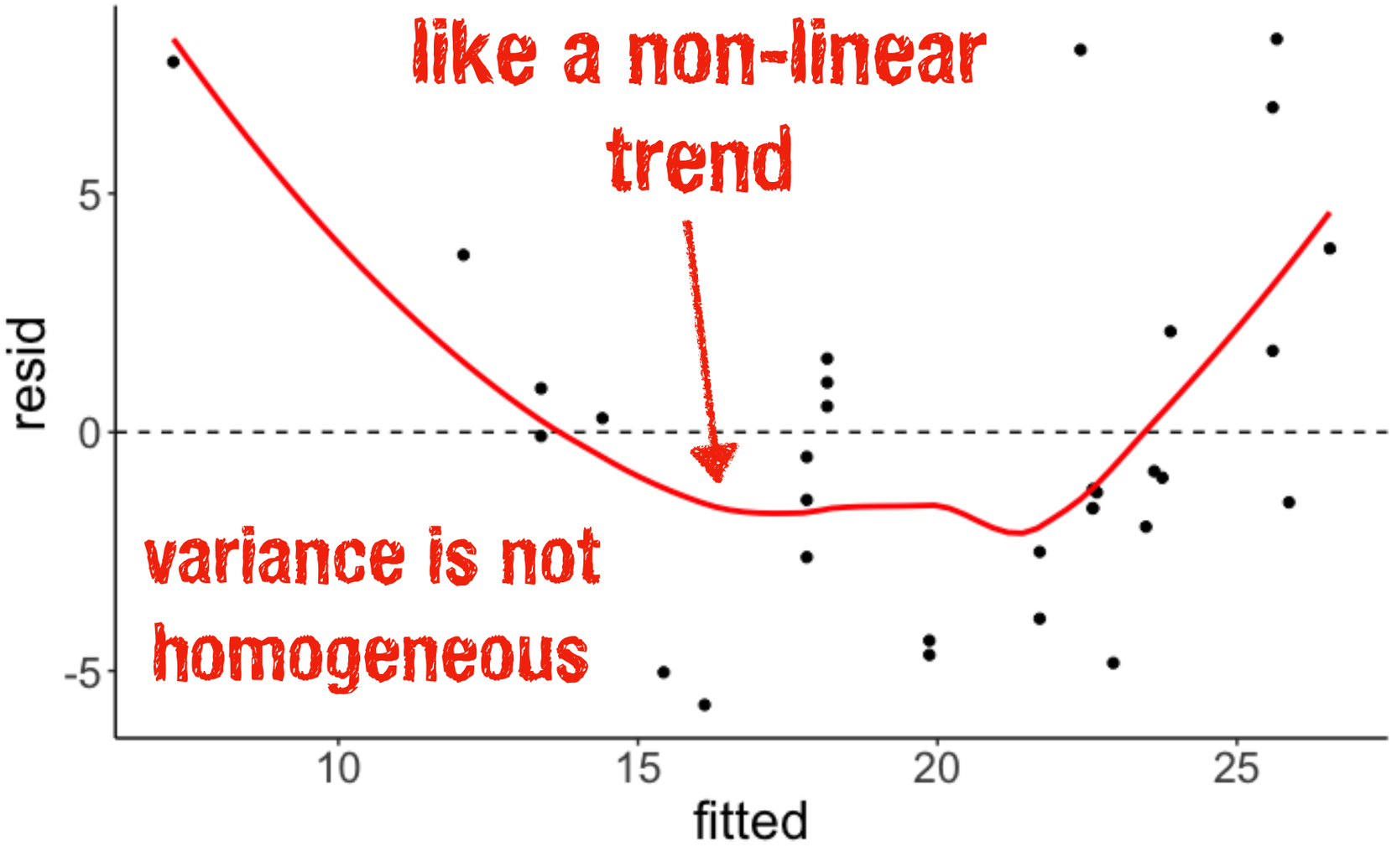
Linear and additive

```
fit.car = lm(formula = mpg ~ 1 + hp,
             data = df.car)

ggplot(data = df.car,
       mapping = aes(x = hp,
                     y = mpg)) +
  geom_smooth(method = "lm") +
  geom_smooth(color = "red",
              se = F) +
  geom_point()
```



Linear and additive

```
fit.car %>%
  augment() %>%
  clean_names() %>%
  ggplot(data = .,
         mapping = aes(x = fitted,
                       y = resid)) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point() +
  geom_smooth(color = "red",
              se = F)
```



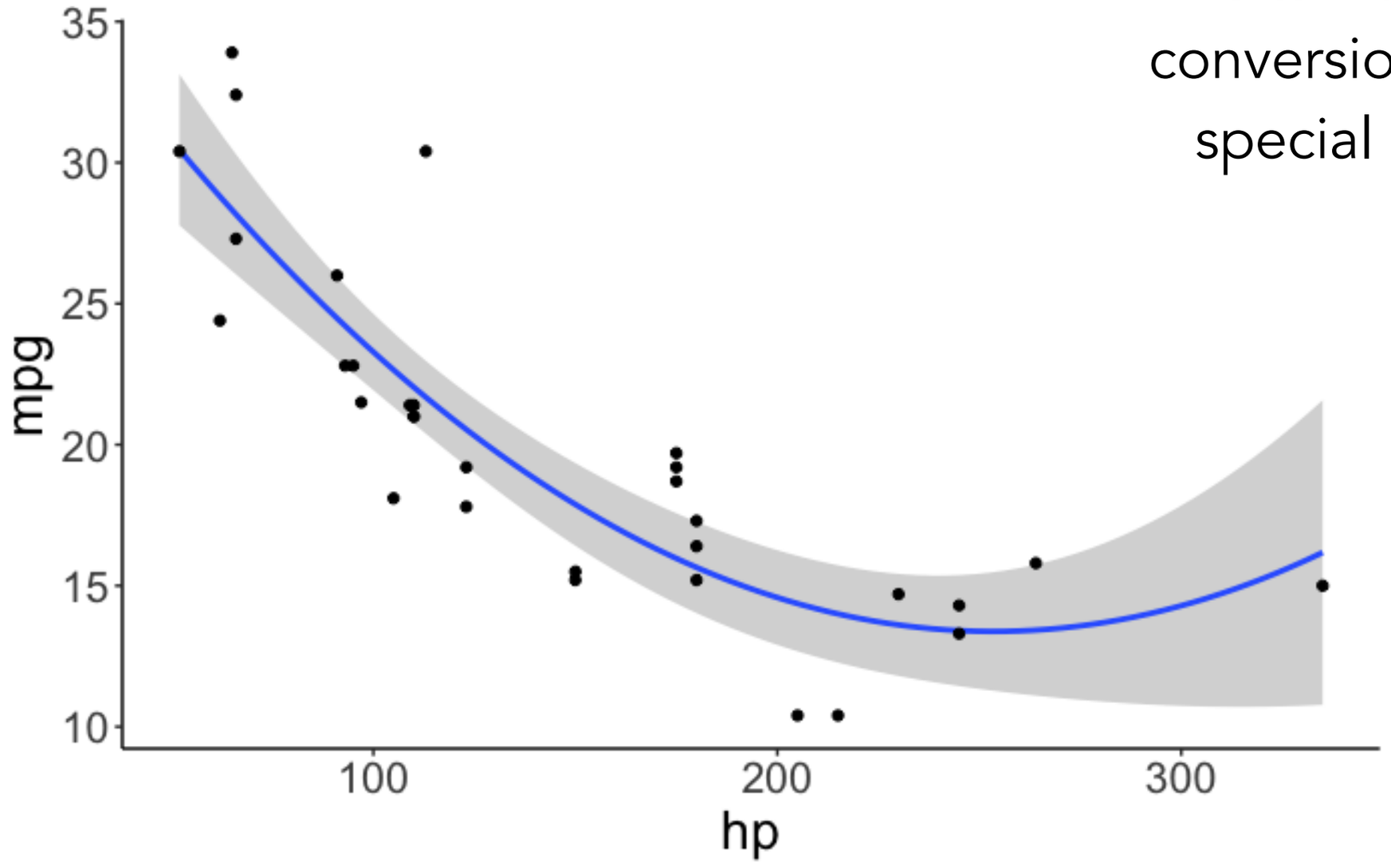
Linear and additive

Solution: Include a squared predictor

```
1 ggplot(data = df.car,
2        mapping = aes(x = hp,
3                      y = mpg)) +
4   geom_smooth(method = "lm",
5               formula = y ~ 1 + x + I(x^2)) +
6   geom_point()
```

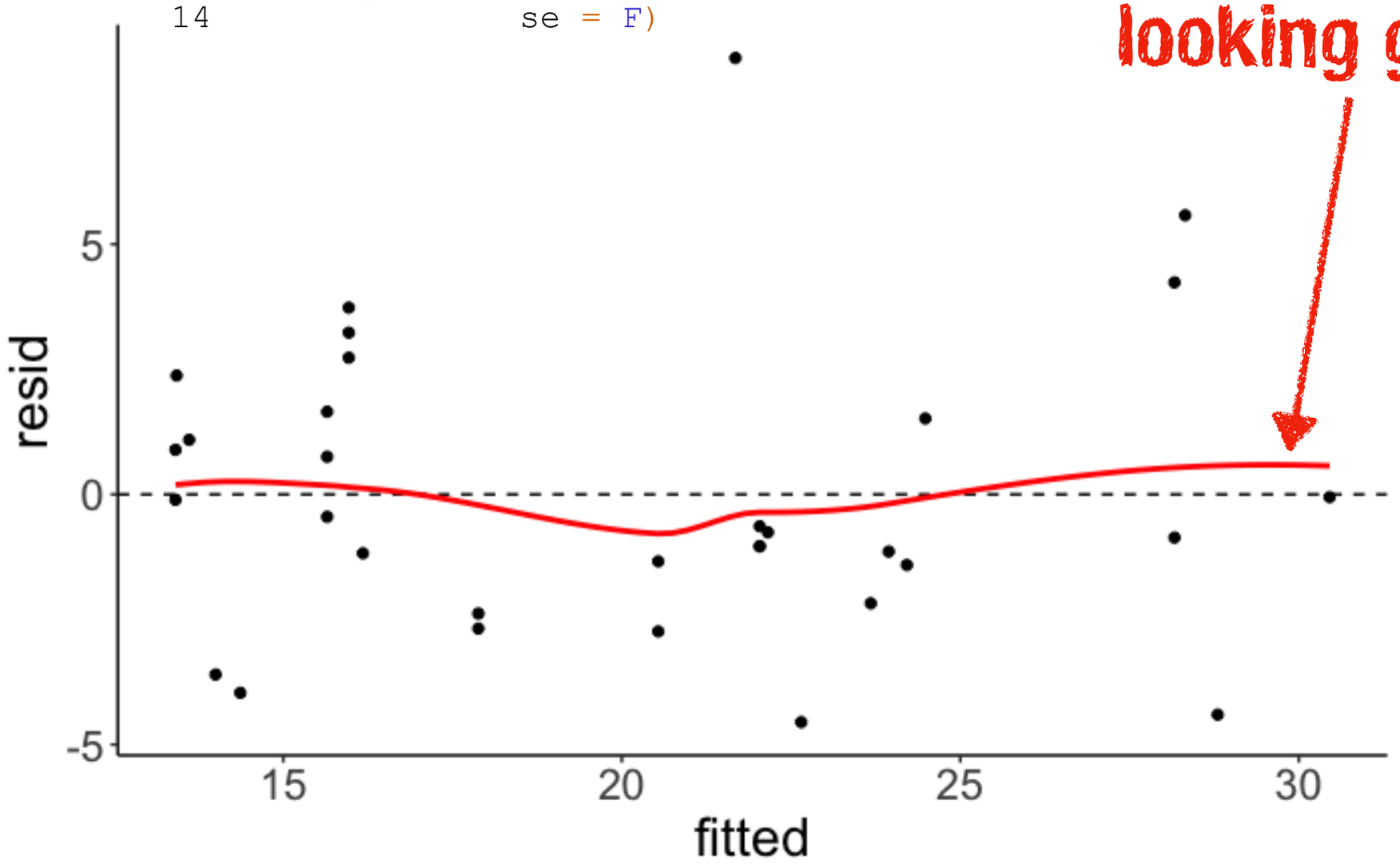
^ has a special meaning in formulas

inhibits interpretation/ conversion (e.g. that special meaning)



Linear and additive **Solution**: Include a squared predictor

```
fit.car2 = lm(formula = mpg ~ 1 + hp + I(hp^2),
              data = df.car)

fit.car2 %>%
  augment() %>%
  clean_names() %>%
  ggplot(data = .,
         mapping = aes(x = fitted,
                       y = resid)) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point() +
  geom_smooth(color = "red",
              se = F)
```



What when assumptions are violated?

- Influential data points
- Linear and additive
- **Data transformation**
- Independence
- Non-parametric analysis
- Simulation methods

Data transformation

Per capita gross domestic product in US dollars.

Infant deaths by age 1 year per 1000 live births.

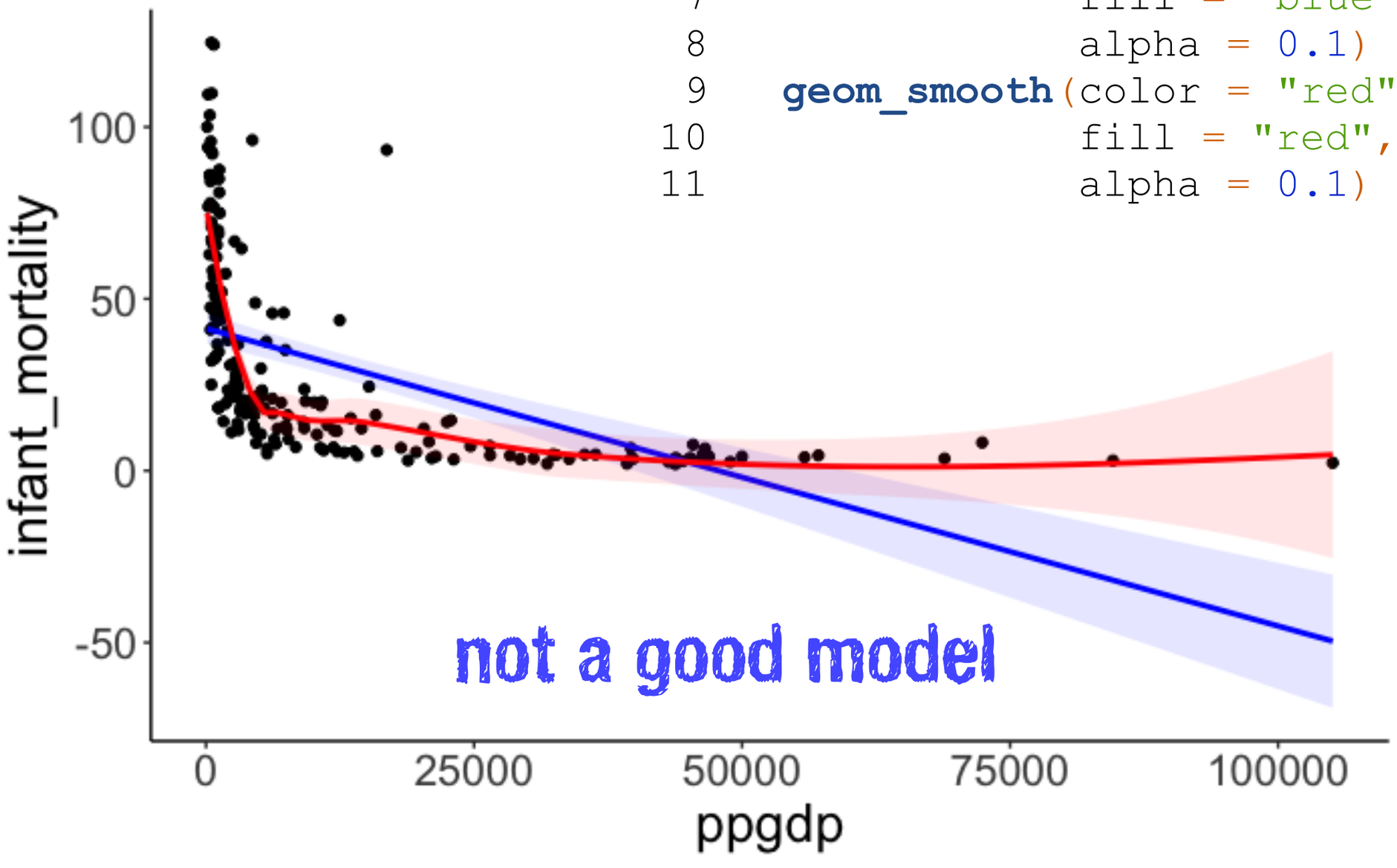
	region	group	fertility	ppgdp	life_exp_f	pct_urban	infant_mortality
Afghanistan	Asia	other	5.97	499.0	49.49	23	124.53
Albania	Europe	other	1.52	3677.2	80.40	53	16.56
Algeria	Africa	africa	2.14	4473.0	75.00	67	21.46
Angola	Africa	africa	5.13	4321.9	53.17	59	96.19
Argentina	Latin Amer	other	2.17	9162.1	79.89	93	12.34

```
1 fit.mortality1 = lm(formula = infant_mortality ~ ppgdp,
2                     data = df.un)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 41.3780016  2.2157454  18.675  < 2e-16 ***
ppgdp       -0.0008656  0.0001041  -8.312 1.73e-14 ***
```

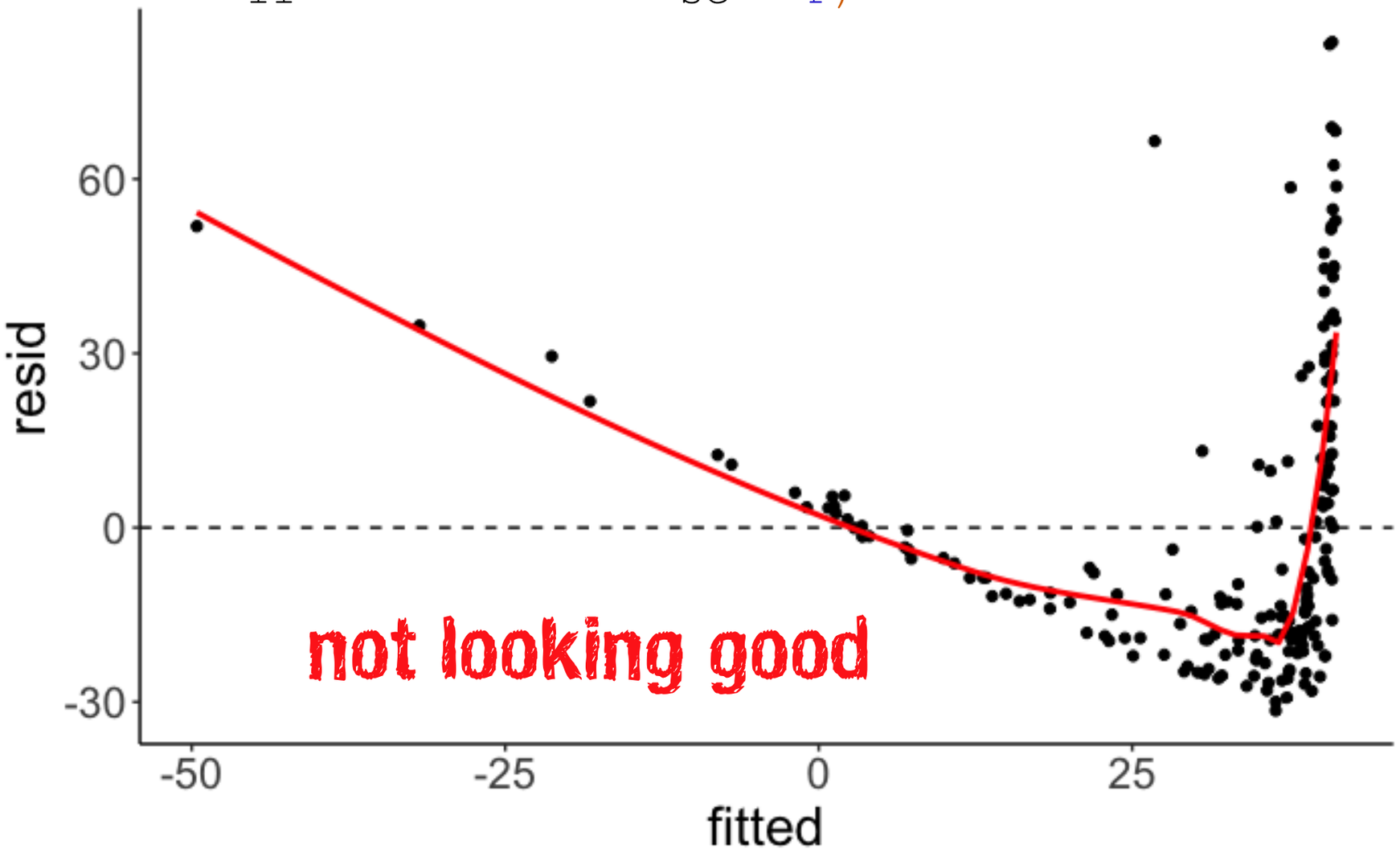
Data transformation

```
ggplot(data = df.un,
       mapping = aes(x = ppgdp,
                     infant_mortality)) +
  geom_point() +
  geom_smooth(method = "lm",
              color = "blue",
              fill = "blue",
              alpha = 0.1) +
  geom_smooth(color = "red",
              fill = "red",
              alpha = 0.1)
```



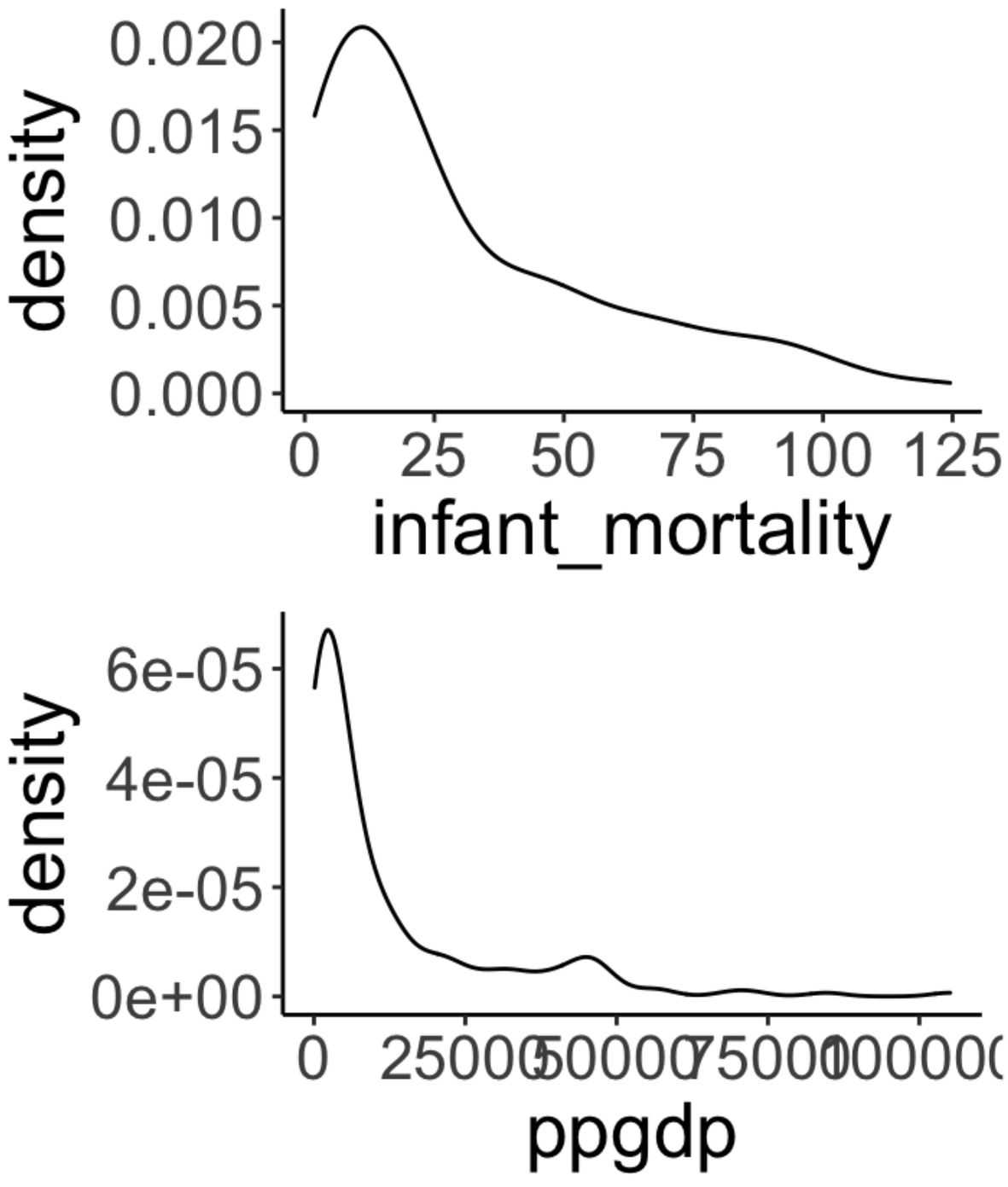
Data transformation

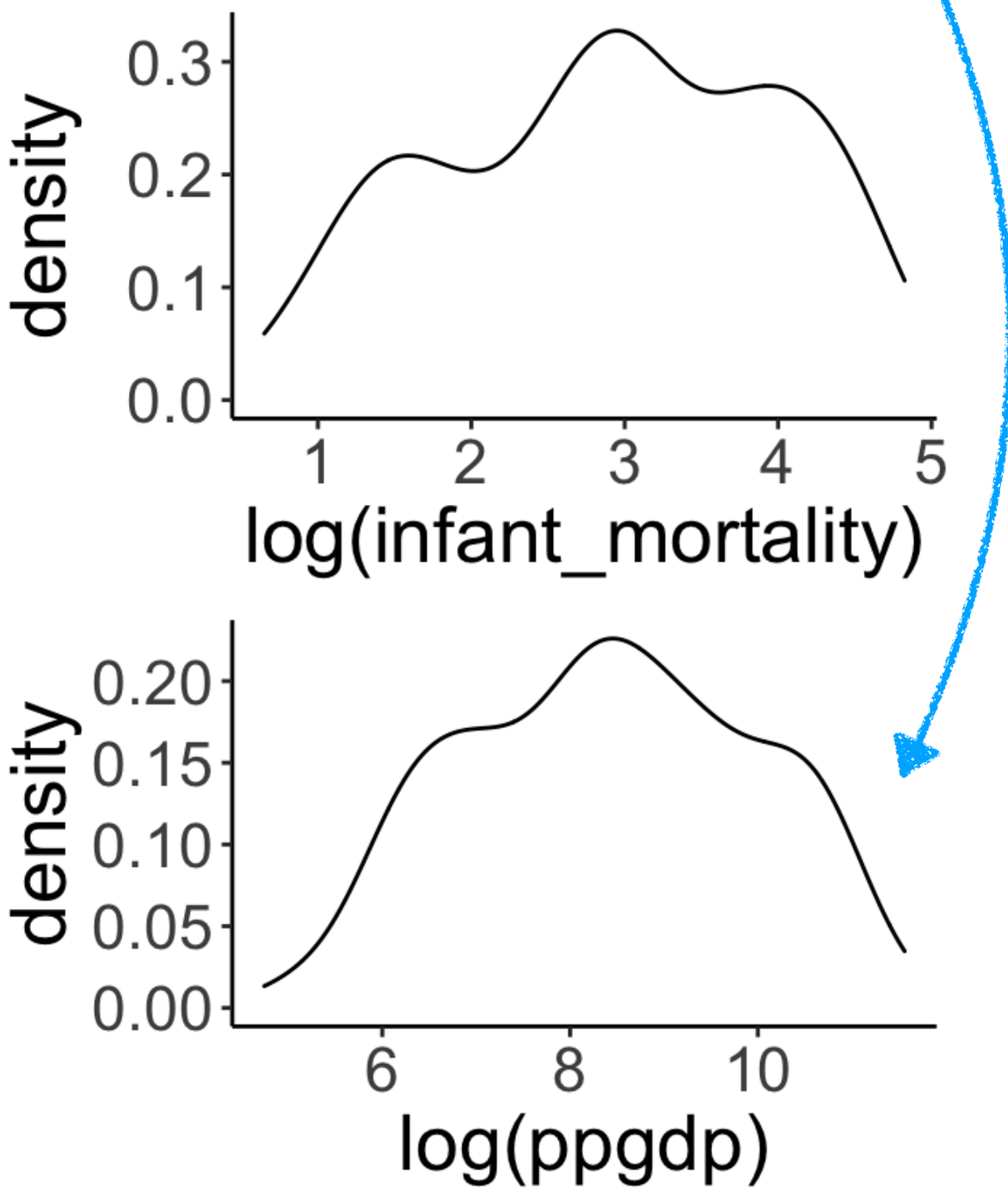
```
fit.mortality1 %>%
  augment() %>%
  clean_names() %>%
  ggplot(data = .,
         mapping = aes(x = fitted,
                       y = resid)) +
  geom_hline(yintercept = 0,
             linetype = 2) +
  geom_point() +
  geom_smooth(color = "red",
              se = F)
```



Data transformation

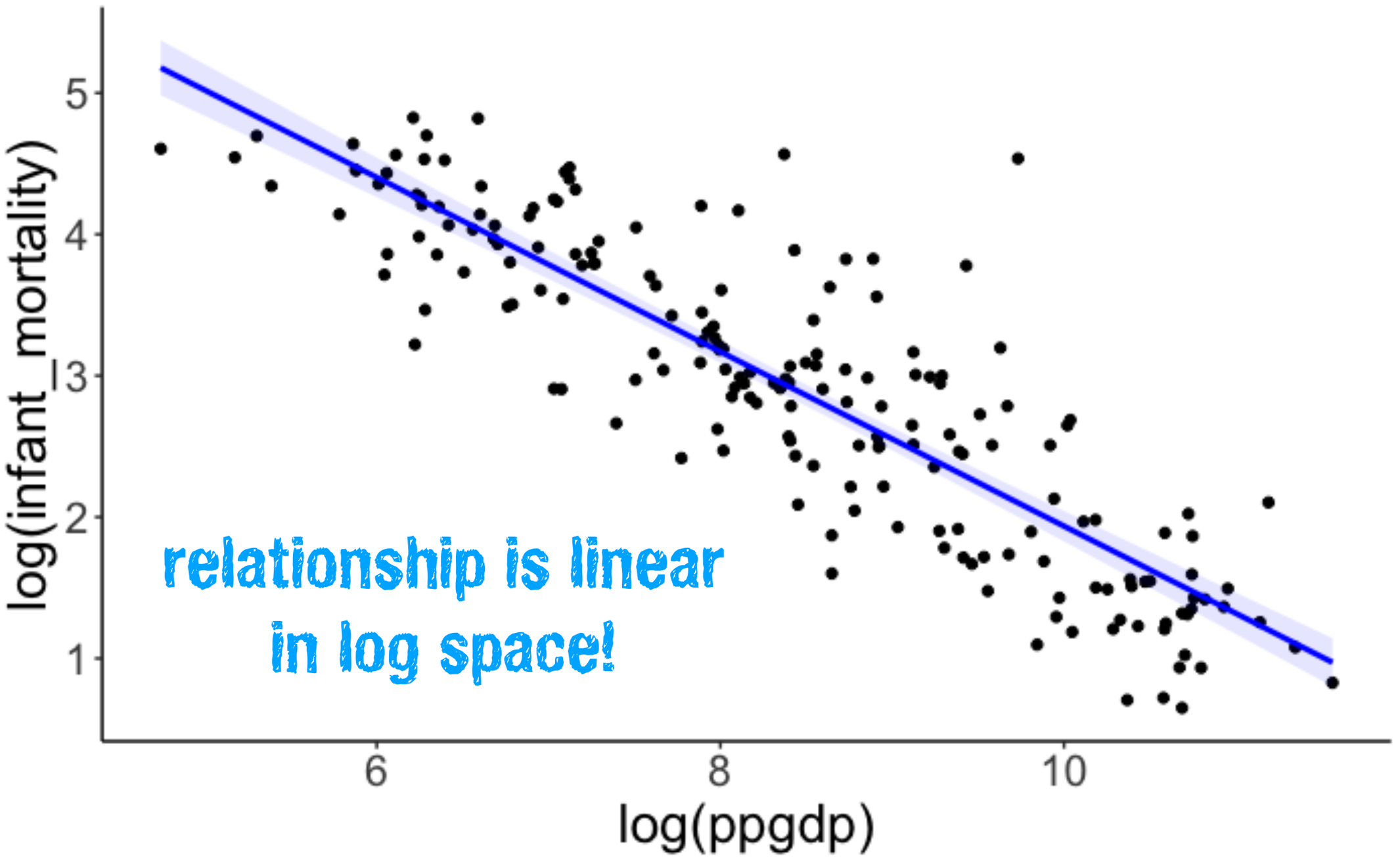
look more like normal distributions





Data transformation

```
fit.mortality2 = lm(formula = log(infant_mortality) ~ log(ppgdp),
                    data = df.un)
```



Data transformation

```
1 fit.mortality2 = lm(formula = log(infant_mortality) ~ log(ppgdp),
2                     data = df.un)
```

```
Call:
lm(formula = log(infant_mortality) ~ log(ppgdp), data = df.un)

Residuals:
     Min       1Q   Median       3Q      Max
-1.16789 -0.36738 -0.02351  0.24544  2.43503

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.10377    0.21087   38.43   <2e-16 ***
log(ppgdp)  -0.61680    0.02465  -25.02   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5281 on 191 degrees of freedom
Multiple R-squared:  0.7662,    Adjusted R-squared:  0.765
F-statistic: 625.9 on 1 and 191 DF,  p-value: < 2.2e-16
```

how to interpret the parameter?

predicted % change in `y`

for a 1% increase in `x`

Data transformation

from the `ggeffects` package

```
1 fit.mortality2 %>%
2   ggpredict(terms = "ppgdp [exp]")
```

to "untransform "the log-transformed data

```
Model has log-transformed response. Back-
transforming predictions to original response scale.
Standard errors are still on the log-scale.

# Predicted values of infant_mortality
# x = ppgdp

          x | Predicted |   SE |            95% CI
-------------------------------------------------
     114.80 |    177.36 | 0.10 | [146.33, 214.98]
     598.80 |     64.03 | 0.06 | [ 56.64,  72.39]
    1239.80 |     40.87 | 0.05 | [ 37.09,  45.04]
    2882.30 |     24.29 | 0.04 | [ 22.48,  26.25]
    4495.80 |     18.47 | 0.04 | [ 17.14,  19.89]
    7522.40 |     13.44 | 0.04 | [ 12.43,  14.54]
   14497.30 |      8.97 | 0.05 | [  8.17,   9.85]
1.05095e+05 |      2.64 | 0.09 | [  2.23,   3.13]
```

Data transformation

$$y = \exp(8.1 + (-0.62) \cdot \log(4))$$

x	Predicted	SE	95% CI
4	1406.29	0.18	[993.36, 1990.87]
5	1225.46	0.17	[874.79, 1716.72]
6	1095.12	0.17	[788.48, 1521.01]
7	995.79	0.16	[722.18, 1373.06]
8	917.06	0.16	[669.27, 1256.60]
9	852.80	0.16	[625.82, 1162.11]
10	799.15	0.16	[589.35, 1083.62]
12	714.15	0.15	[531.18, 960.13]

transform back from log space

$$\log(y) = \beta_0 + \beta_1 x$$

$$\exp(\log(y)) = \exp(\beta_0 + \beta_1 x)$$

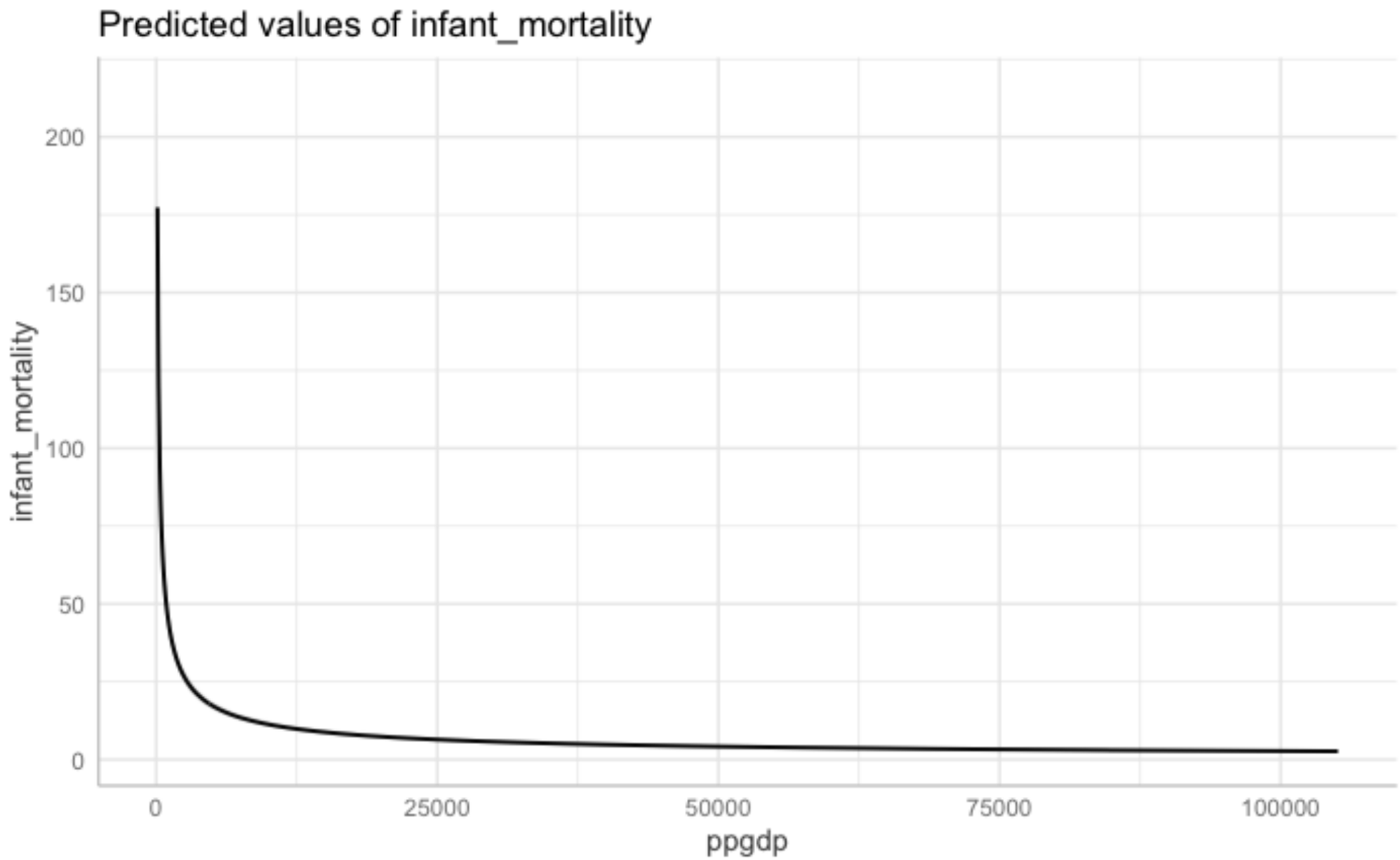
$$y = \exp(\beta_0 + \beta_1 x)$$

```
Call:
lm(formula = log(infant_mortality) ~ log(ppgdp), data
= df.un)

Residuals:
     Min       1Q   Median       3Q      Max
-1.16789 -0.36738 -0.02351  0.24544  2.43503

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.10377    0.21087   38.43   <2e-16 ***
log(ppgdp)  -0.61680    0.02465  -25.02   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
0.1 ‘ ’ 1

Residual standard error: 0.5281 on 191 degrees of
freedom
Multiple R-squared:  0.7662,   Adjusted R-squared:
0.765
F-statistic: 625.9 on 1 and 191 DF,  p-value: <
2.2e-16
```

https://data.library.virginia.edu/interpreting-log-transformations-in-a-linear-model/

Data transformation

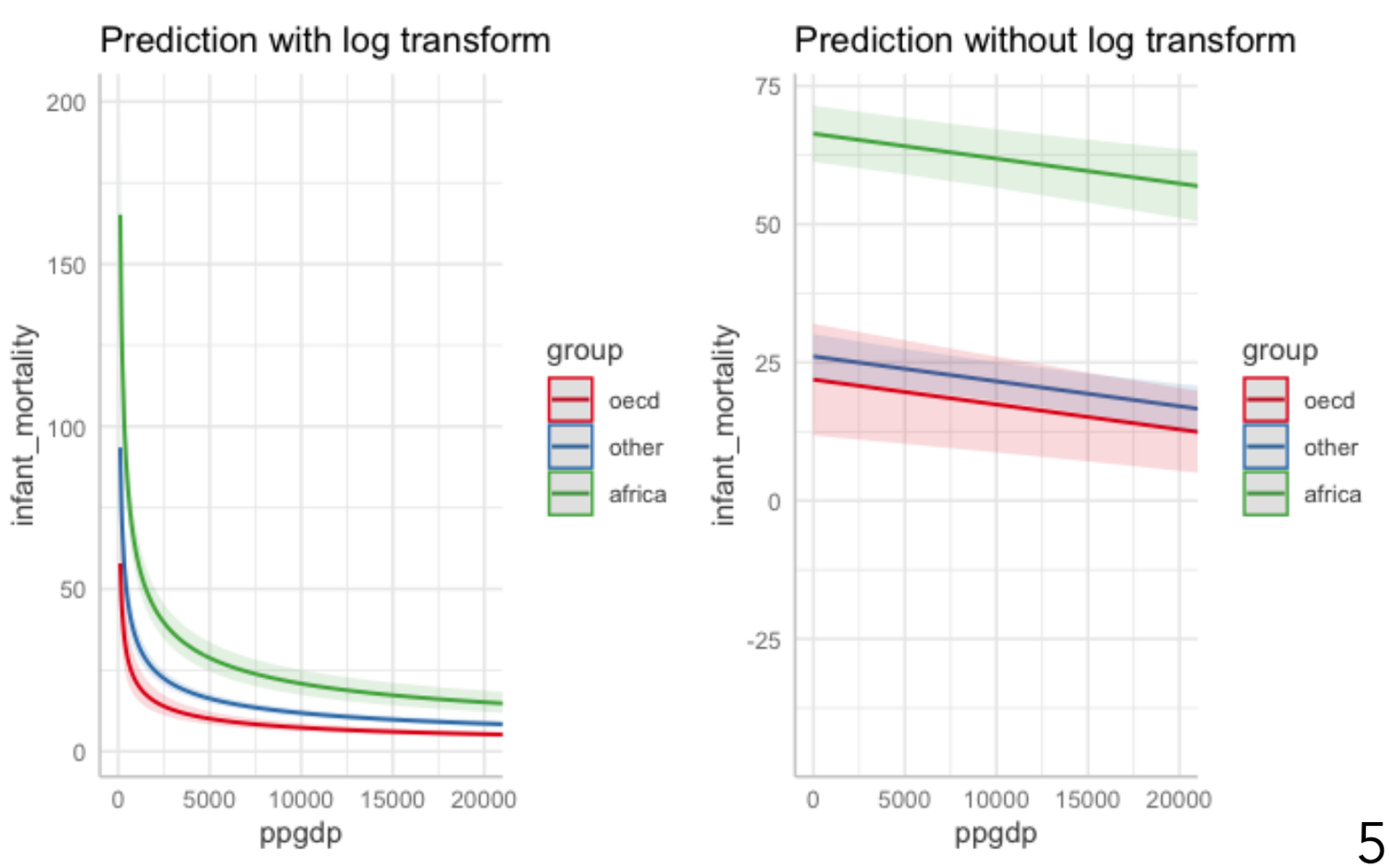
```
1 fit.mortality2 %>%
2   ggpredict(terms = "ppgdp [exp]") %>%
3   plot()
```



Data transformation

```r
# with log transforms
fit.mortality5 = lm(formula = log(infant_mortality) ~ log(ppgdp) + group,
                    data = df.un)

# without log transforms
fit.mortality6 = lm(formula = infant_mortality ~ ppgdp + group,
                    data = df.un)

p1 = ggpredict(fit.mortality5,
               terms = c("ppgdp [exp]", "group")) %>%
  plot() +
  labs(title = "Prediction with log transform") +
  coord_cartesian(xlim = c(0, 20000))

p2 = ggpredict(fit.mortality6,
               terms = c("ppgdp", "group")) %>%
  plot() +
  labs(title = "Prediction without log transform") +
  coord_cartesian(xlim = c(0, 20000))

p1 + p2
```

multiple predictors



What when assumptions are violated?

- Influential data points
- Linear and additive
- Data transformation
- **Independence**
- Non-parametric analysis
- Simulation methods

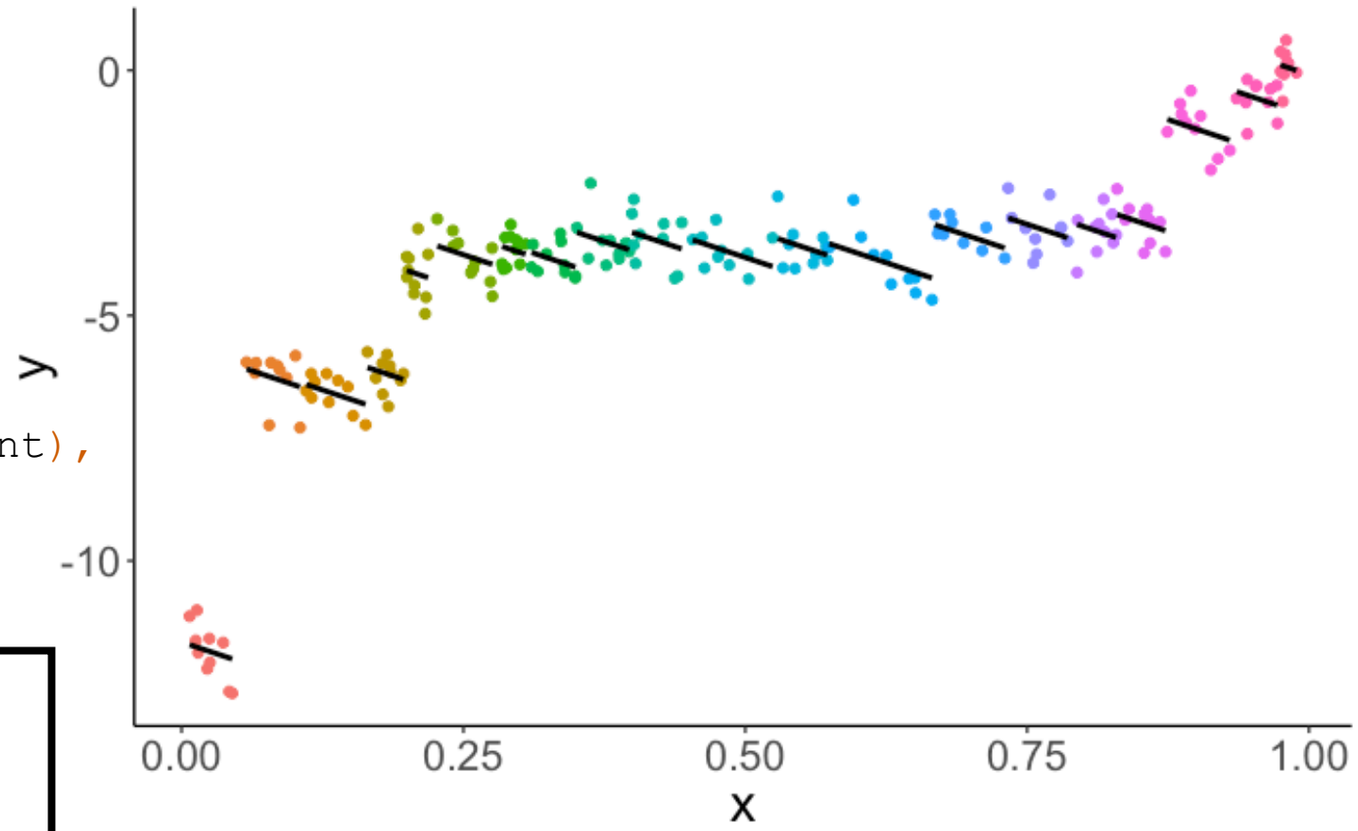
Independence

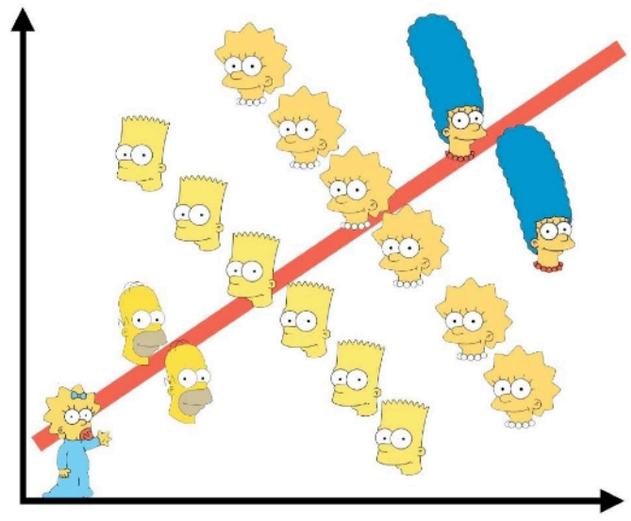
- difficult/impossible to check based on model diagnostics (e.g. residual plots)
- we need to know based on how the data was generated

Simpson's paradox

```
1 lmer(formula = y ~ 1 + x + (1 | participant),
2        data = df.simpson) %>%
3    summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + x + (1 | participant)
   Data: df.simpson

REML criterion at convergence: 345.1

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.43394 -0.59687  0.04493  0.62694  2.68828

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 21.4898  4.6357
 Residual                 0.1661  0.4075
Number of obs: 200, groups:  participant, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.1577     1.3230  -0.119
x            -7.6678     1.6572  -4.627

Correlation of Fixed Effects:
  (Intr)
x -0.621
```





negative (!) relationship between x and y

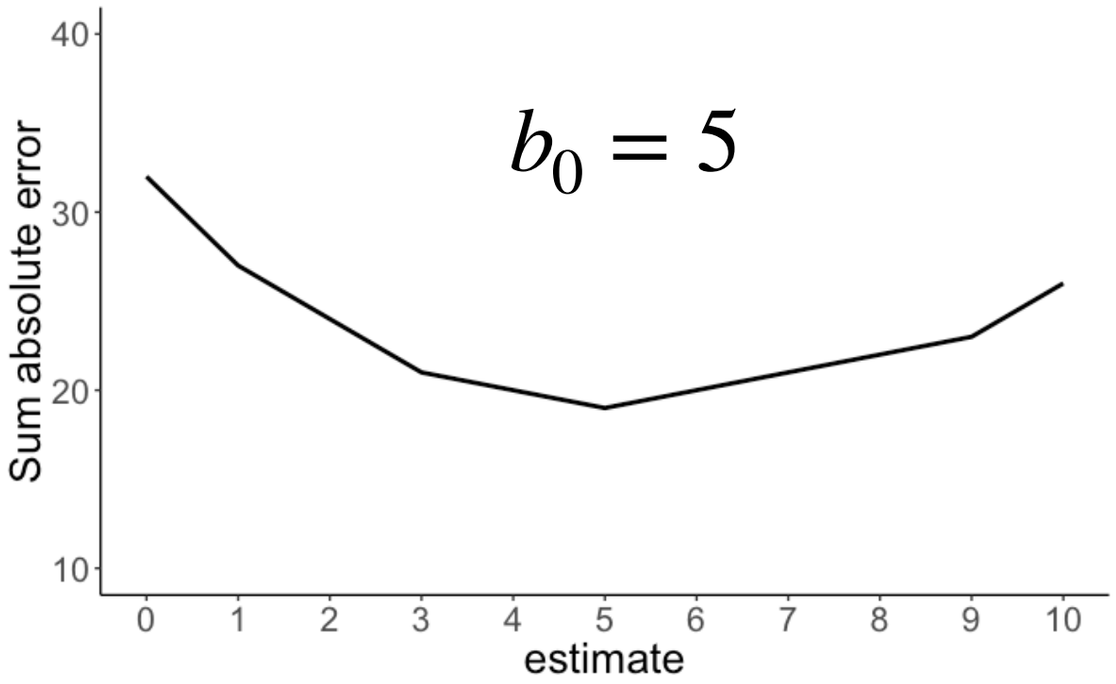
What when assumptions are violated?

- Influential data points
- Linear and additive
- Data transformation
- Independence
- **Non-parametric analysis**
- Simulation methods

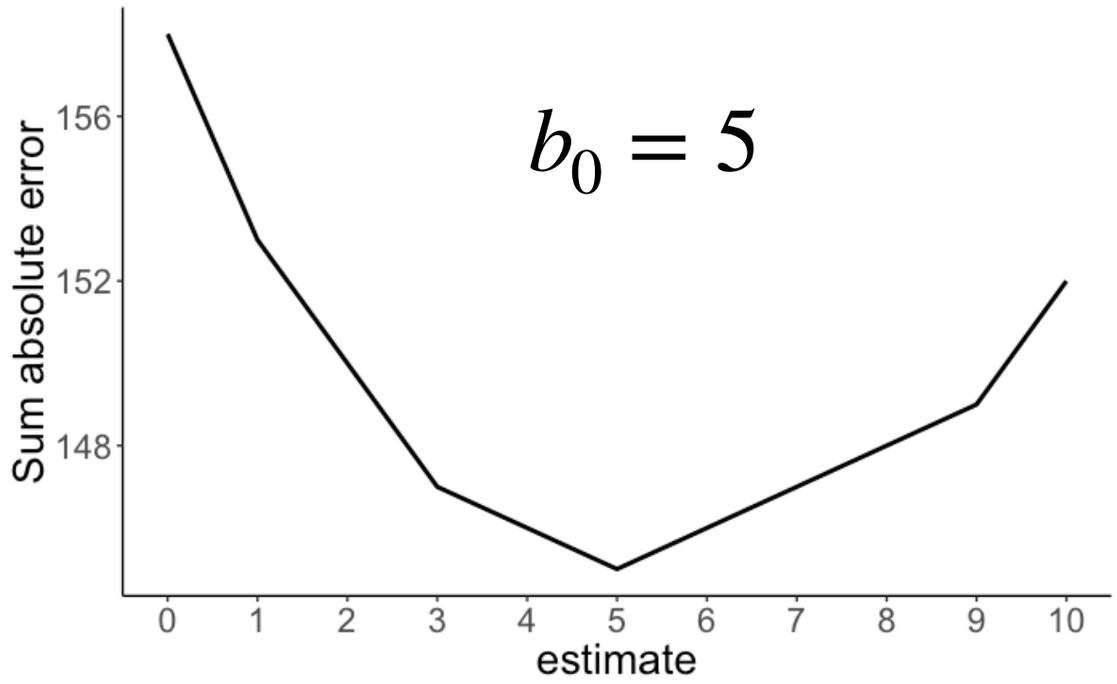
Sum of absolute errors

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

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



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

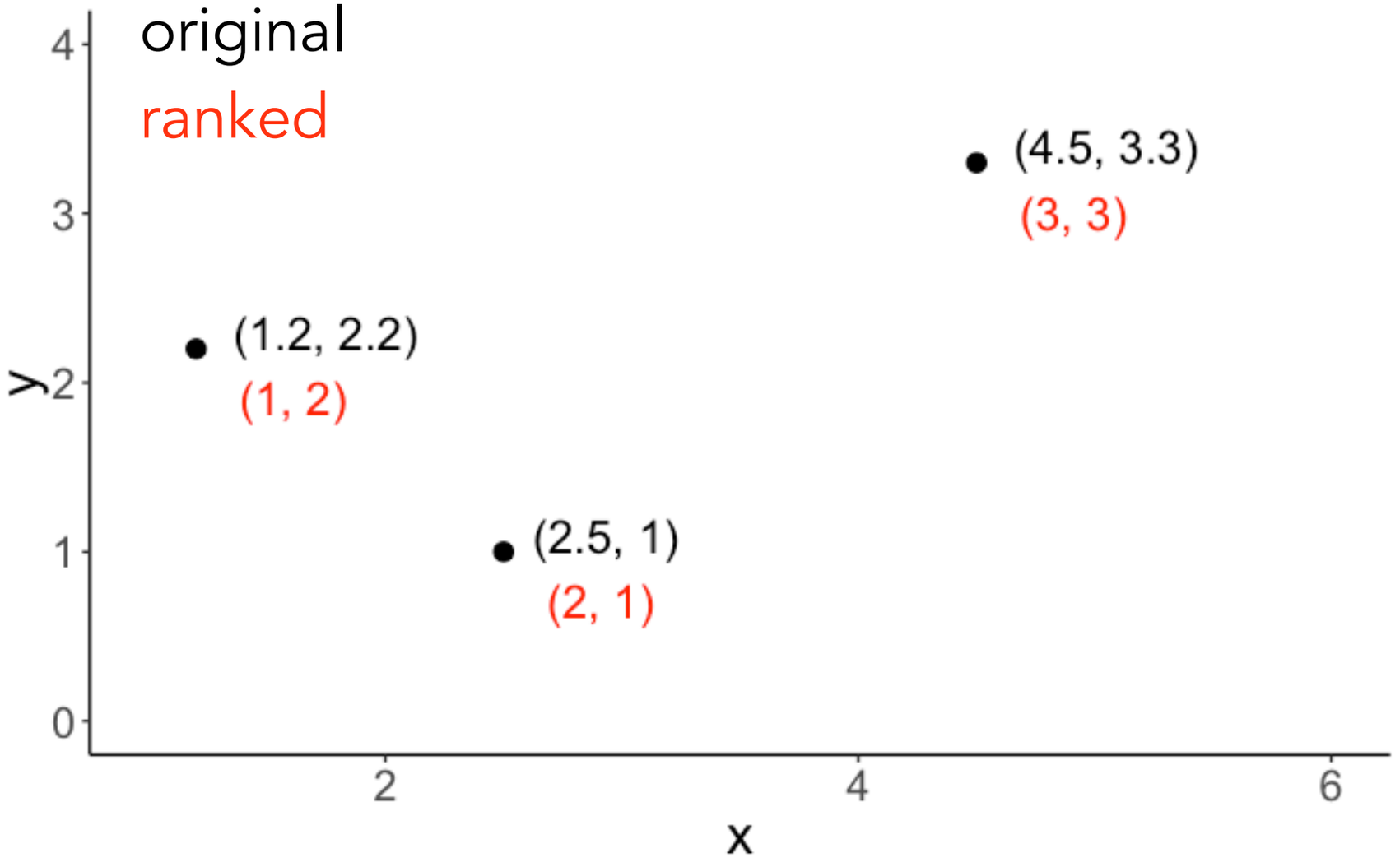


the **median** minimizes the sum of absolute errors

is robust to outliers!

Spearman rank order correlation

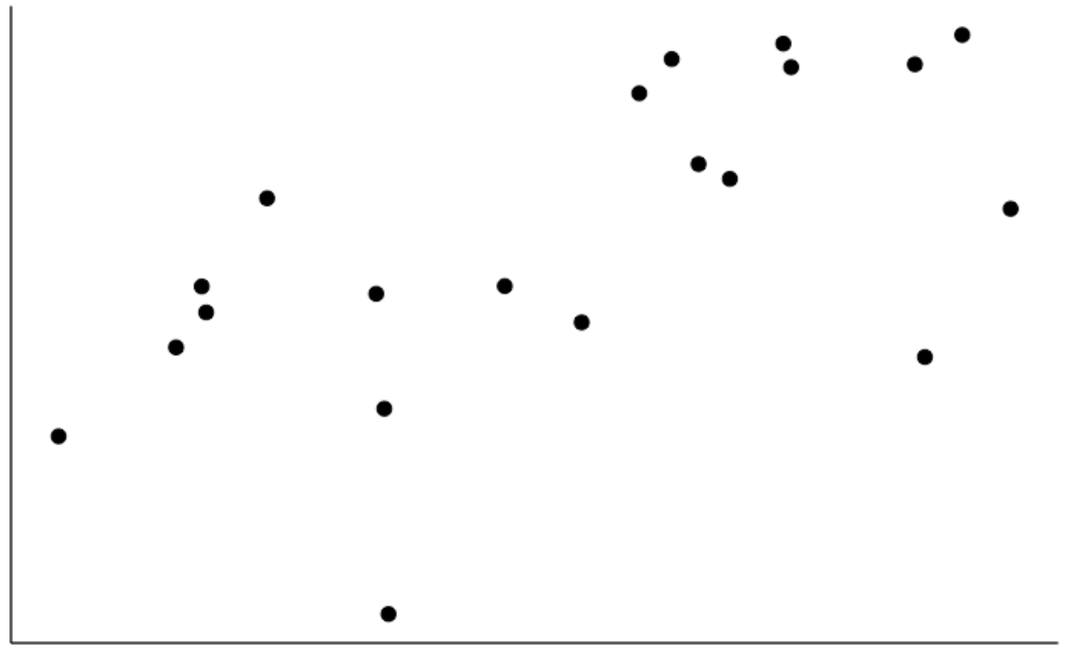
- transform original data into **ranks**
- calculate correlation on the ranked data



Spearman rank order correlation

- transform original data into ranks
- calculate correlation on the ranked data

x	y	x_rank	y_rank
0.27	1.14	5	12
0.37	0.97	6	8
0.57	0.92	10	6
0.91	0.85	18	4
0.20	0.98	3	9
0.90	1.39	17	17
0.94	1.44	19	20
0.66	1.40	12	18
0.63	1.33	11	15
0.06	0.71	1	2

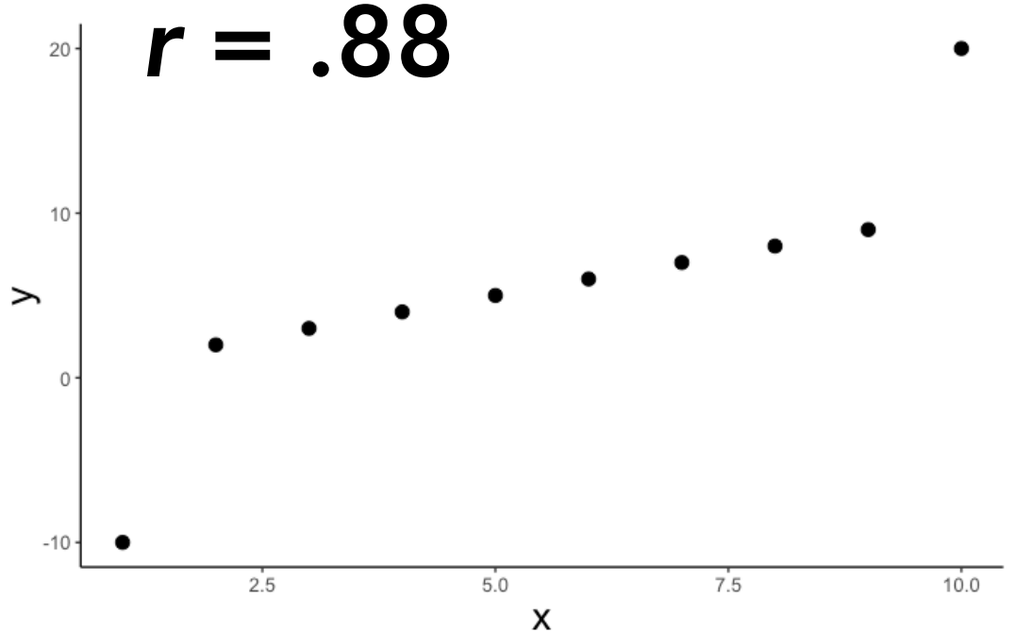


r	spearman	r_ranks
0.609	0.595	0.595

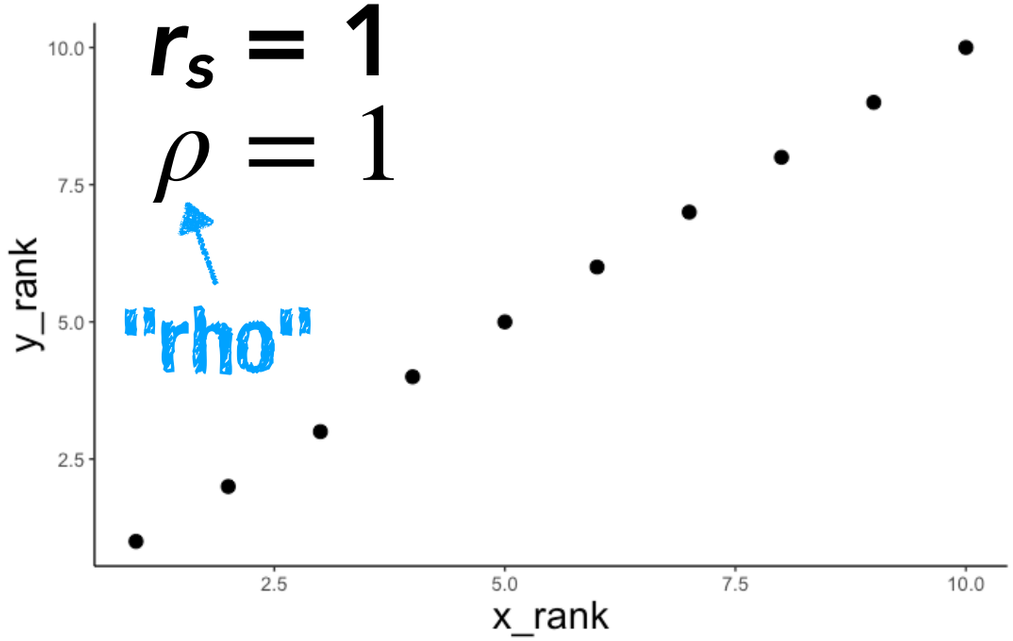
```
1 # correlation
2 df.spearman %>%
3   summarize(r = cor(x, y, method = "pearson"),
4             spearman = cor(x, y, method = "spearman"),
5             r_ranks = cor(x_rank, y_rank))
```

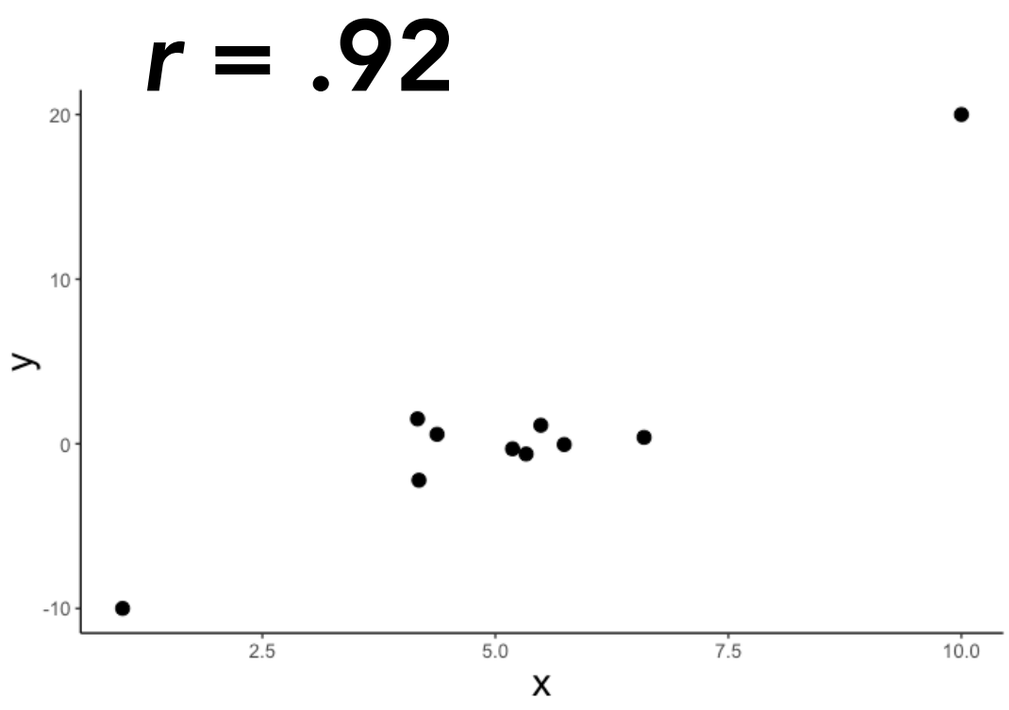
Spearman rank order correlation

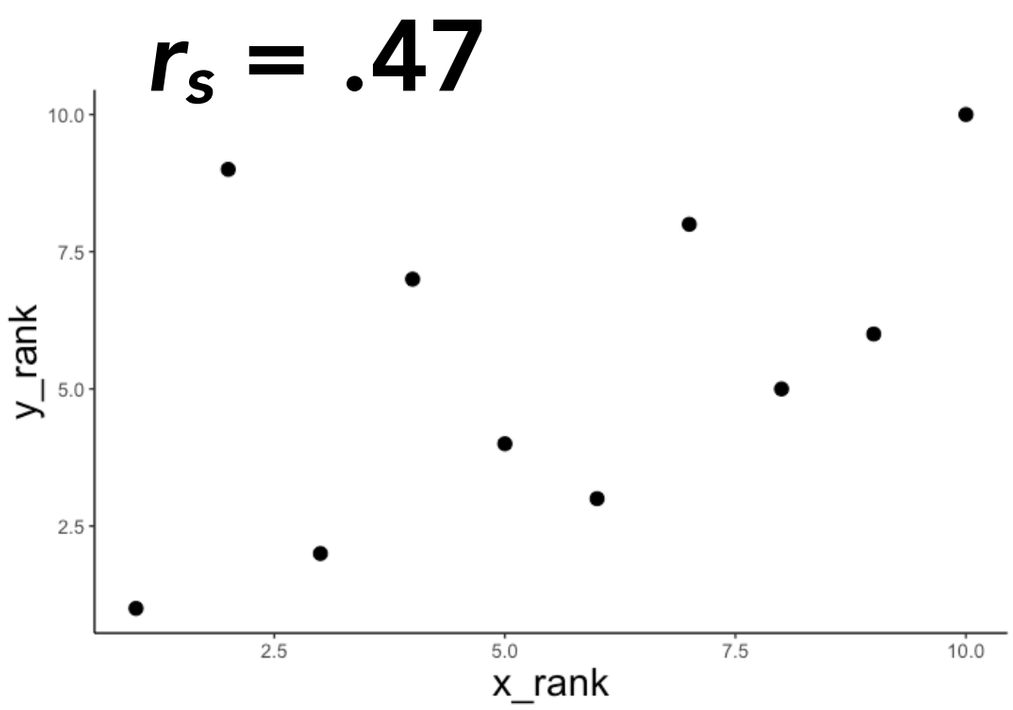
original



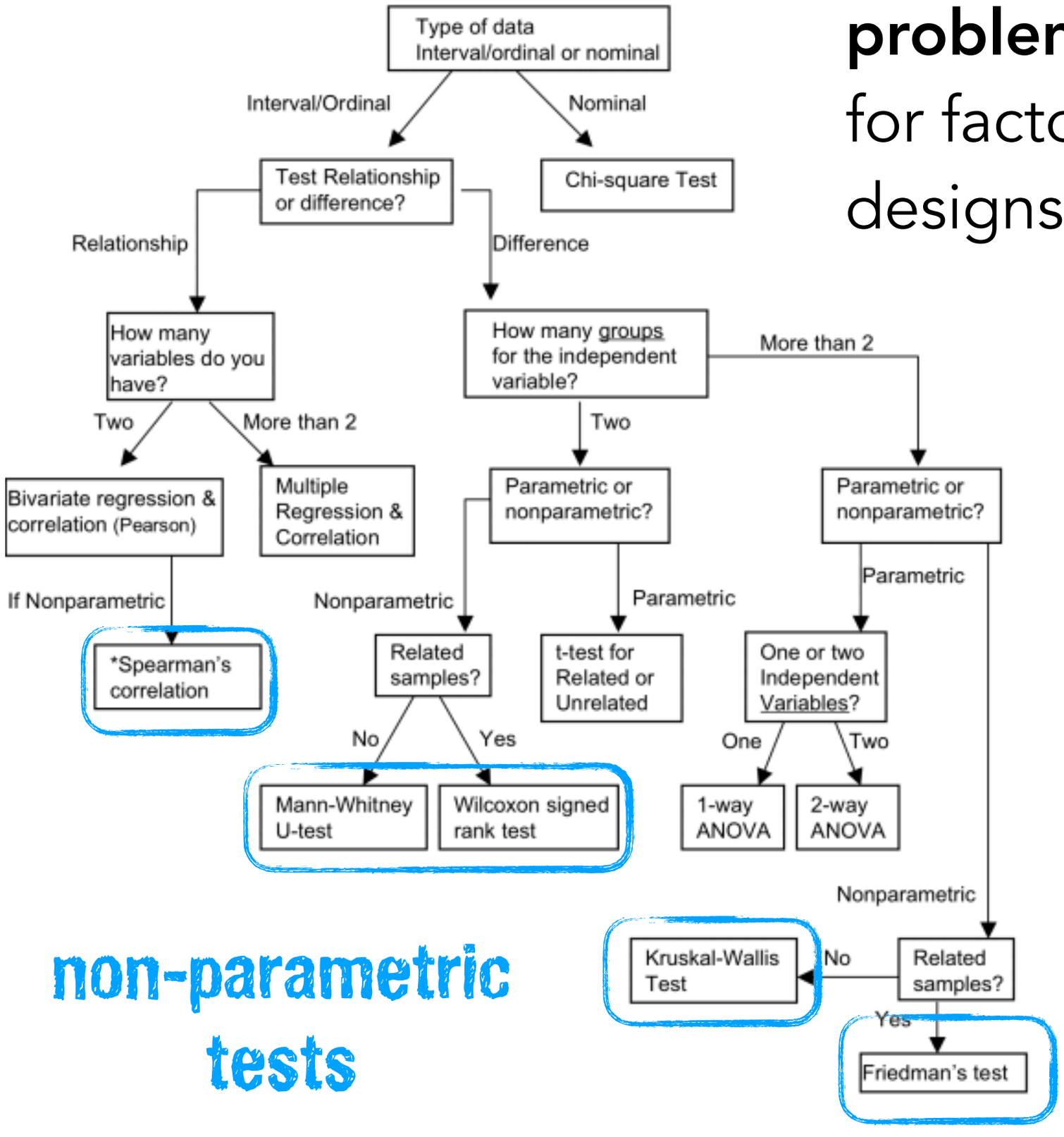
ranked







Non-parametric analysis

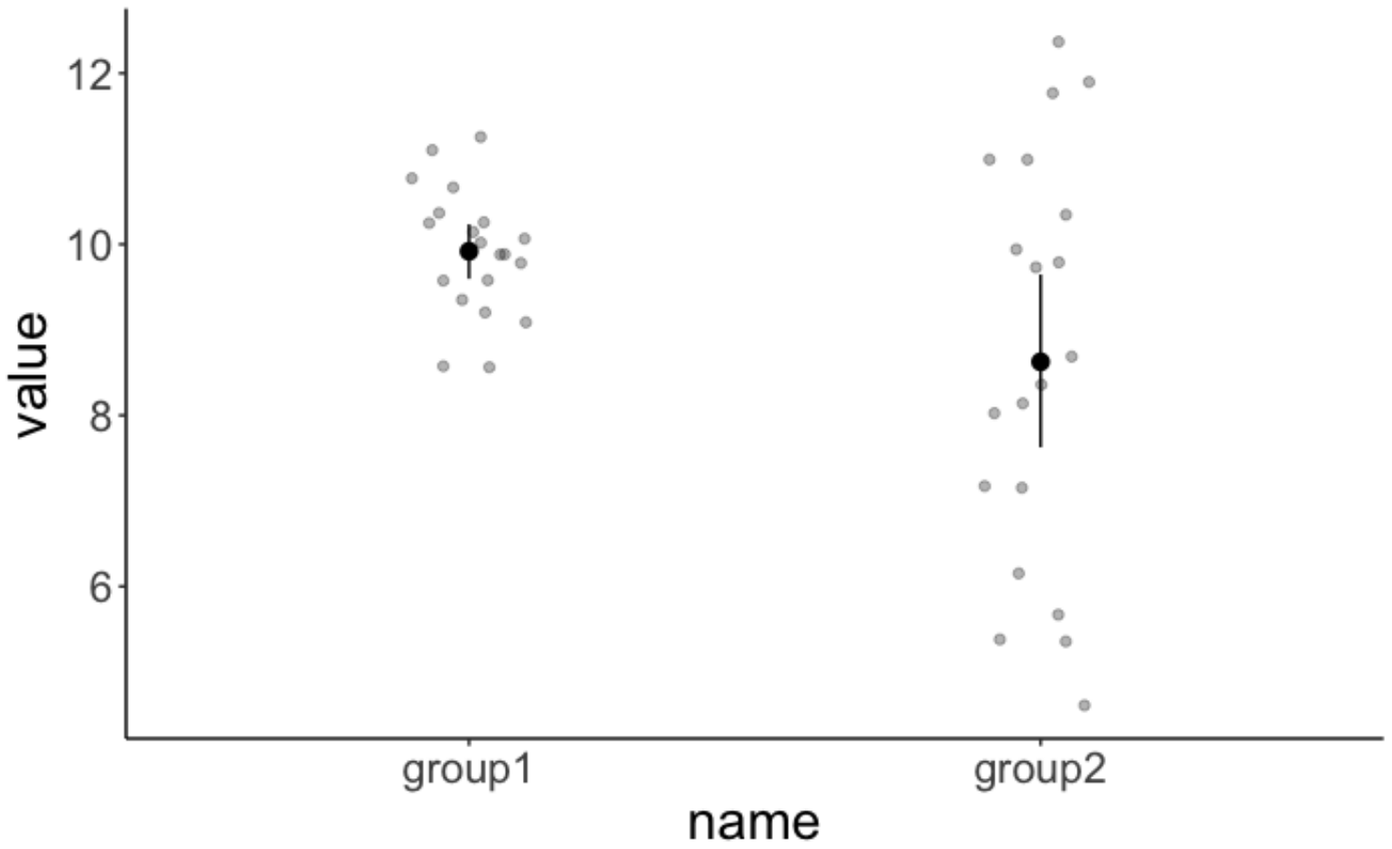


problem: we can't use these for factorial designs / designs with many predictors

Non-parametric analysis

```
df.ttest = tibble(group1 = rnorm(n = 20, mean = 10, sd = 1),
                  group2 = rnorm(n = 20, mean = 8, sd = 3)) %>%
  pivot_longer(cols = everything()) %>%
  mutate(participant = 1:n())
```

```
ggplot(data = df.ttest,
       mapping = aes(x = name,
                     y = value)) +
  geom_point(alpha = 0.3,
             position = position_jitter(width = 0.1)) +
  stat_summary(fun.data = "mean_cl_boot")
```

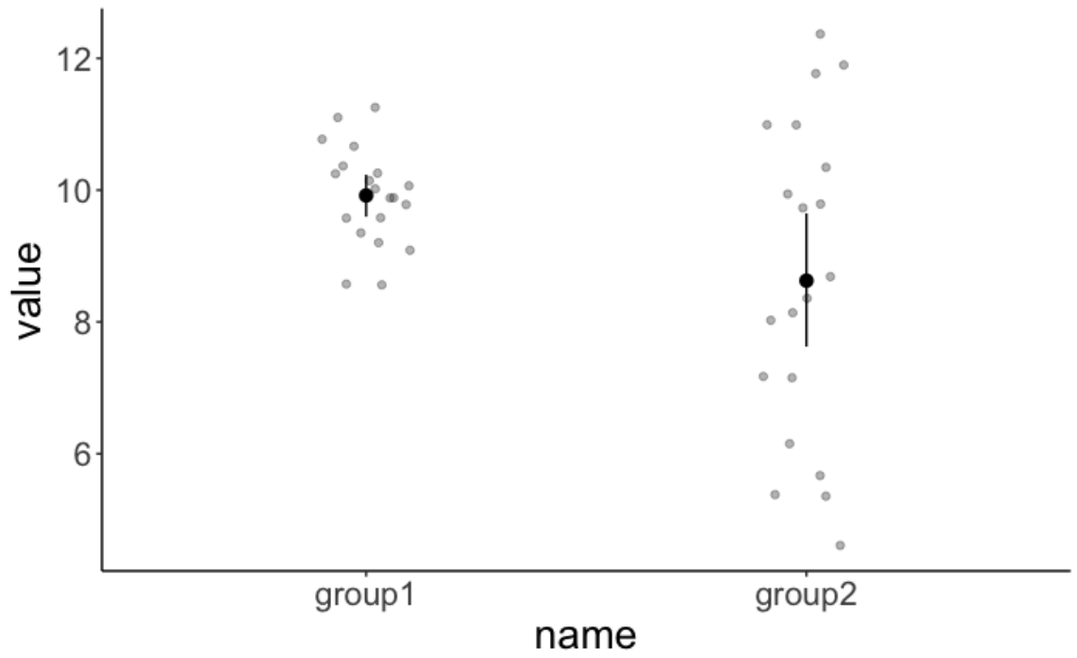


Non-parametric analysis

parametric test

```
1 t.test(formula = value ~ name,
2          data = df.ttest)
```

```
    Welch Two Sample t-test

data:  value by name
t = 2.2941, df = 22.553, p-value = 0.03145
alternative hypothesis: true difference in
means is not equal to 0
95 percent confidence interval:
 0.1257182 2.4588539
sample estimates:
mean in group group1 mean in group group2
            9.918508             8.626222
```



- parametric tests generally have more power
- non-parametric tests are more conservative

non-parametric test

```
1 wilcox.test(formula = value ~ name,
2          data = df.ttest)
```

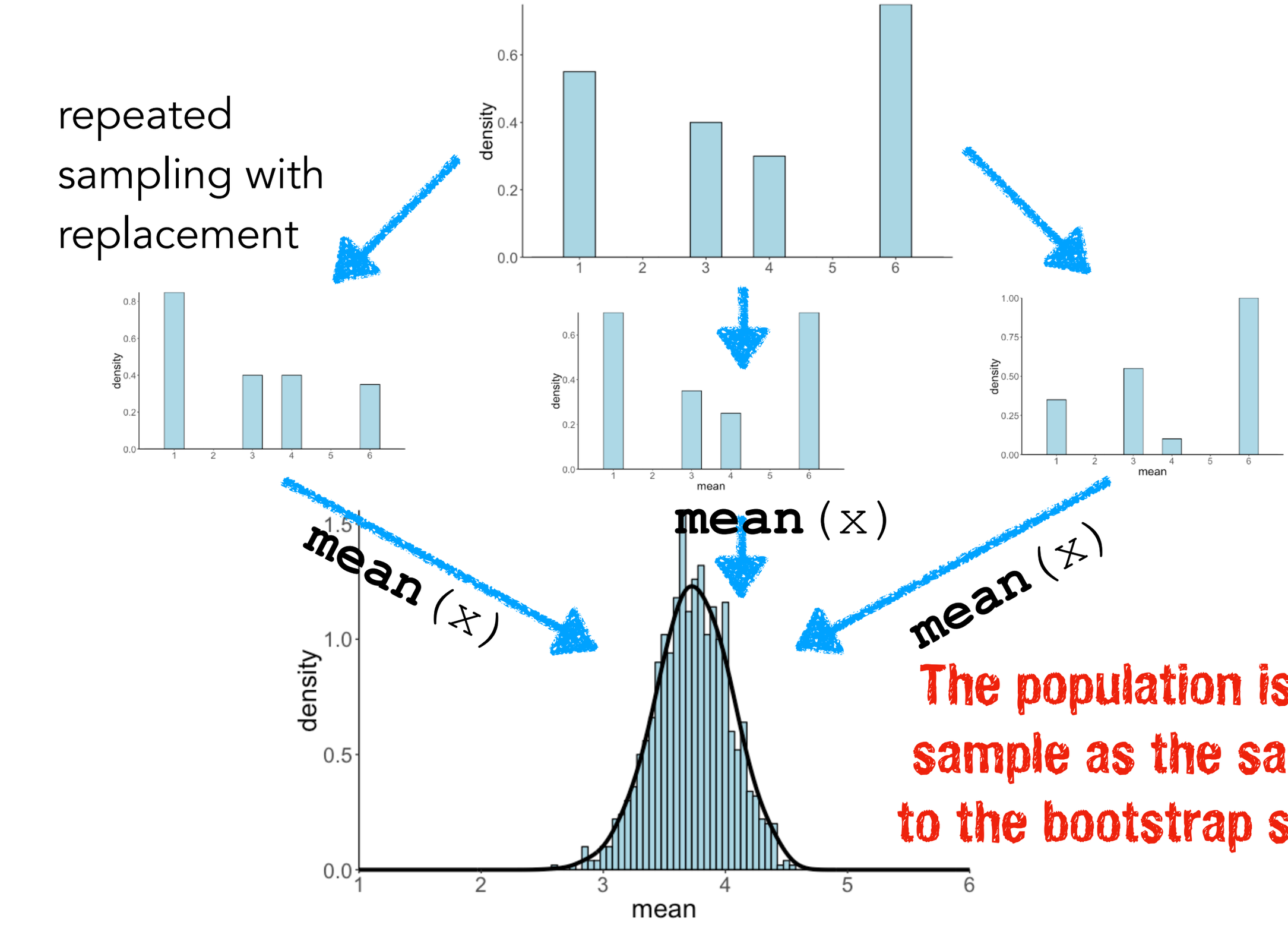
```
1 Wilcoxon rank sum test
2
3 data:  value by name
4 W = 262, p-value = 0.0965
5 alternative hypothesis: true location shift is not equal to 0
```

What when assumptions are violated?

- Influential data points
- Linear and additive
- Data transformation
- Independence
- Non-parametric analysis
- **Simulation methods**

Simulation methods

repeated sampling with replacement

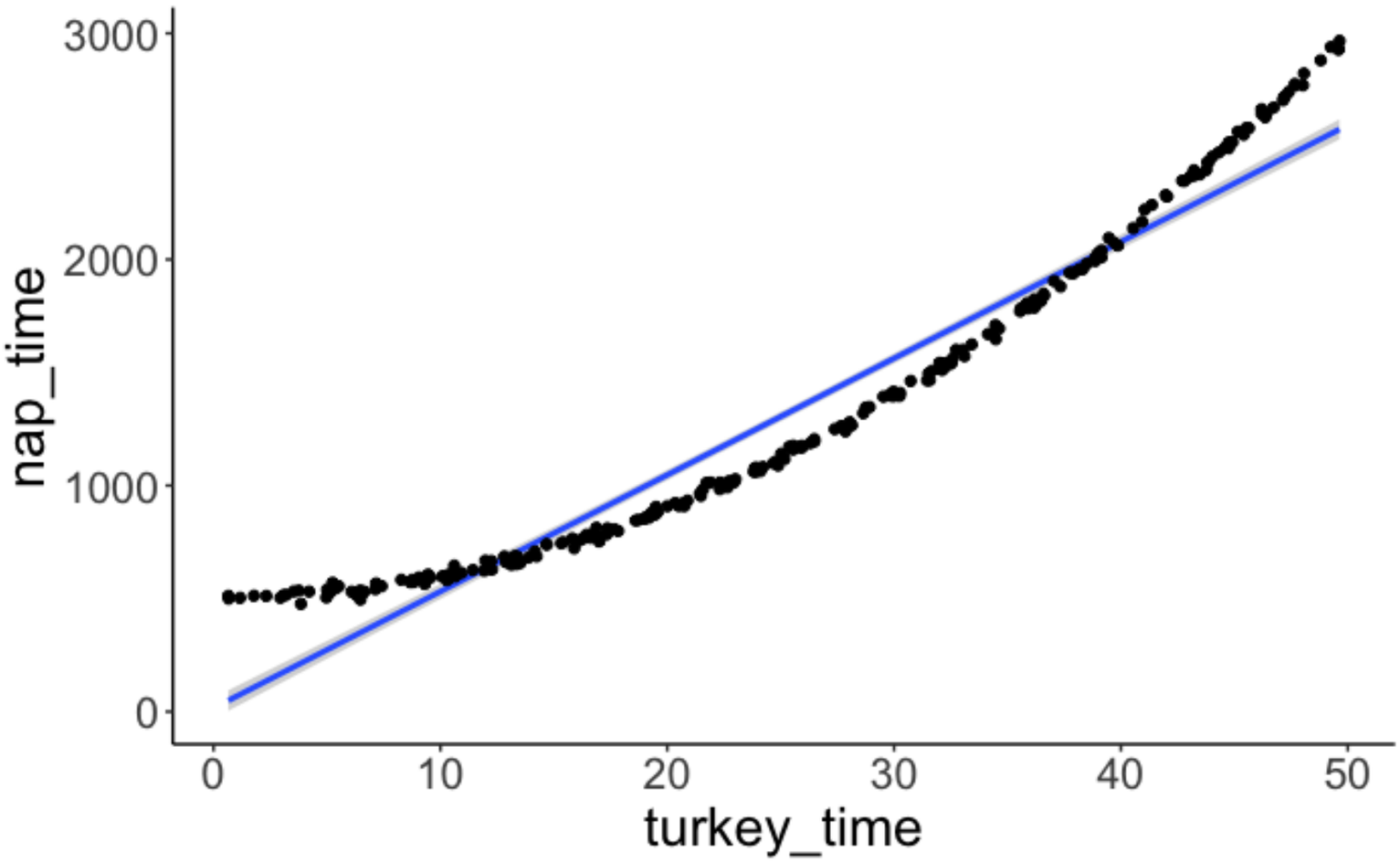


The population is to the sample as the sample is to the bootstrap samples.

Bootstrapping regression models

```
# make reproducible
set.seed(1)

n = 250
df.turkey = tibble(turkey_time = runif(n = n, min = 0, max = 50),
                   nap_time = 500 + turkey_time ^ 2 + rnorm(n, sd = 16))
```

time spent napping (in unrealistic units)



time spent eating turkey

Bootstrapping regression models

```r
fit.turkey = lm(formula = nap_time ~ 1 + turkey_time,
                data = df.turkey)

summary(fit.turkey)
```

```
Call:
lm(formula = nap_time ~ 1 + turkey_time, data = df.turkey)

Residuals:
    Min      1Q  Median      3Q     Max
-212.82 -146.78  -55.17  125.74  462.52

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  15.4974    23.3827   0.663    0.508
turkey_time  51.5746     0.8115  63.557   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 172.4 on 248 degrees of freedom
Multiple R-squared:  0.9422, Adjusted R-squared:  0.9419
F-statistic:  4039 on 1 and 248 DF,  p-value: < 2.2e-16
```

Bootstrapping regression models

```
1 fit.turkey = lm(formula = nap_time ~ 1 + turkey_time,
2                 data = df.turkey)
```

```
1 boot.turkey = Boot(fit.turkey)
```

bootstrapping

parametric

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	15.50	23.38	0.66	0.51	-30.56	61.55
turkey_time	51.57	0.81	63.56	0.00	49.98	53.17

bootstrapping

larger confidence intervals

term	statistic	bias	std.error	conf.low	conf.high
(Intercept)	15.50	-1.13	29.33	-47.64	71.64
turkey_time	51.57	0.02	1.06	49.35	53.61

What when assumptions are violated?

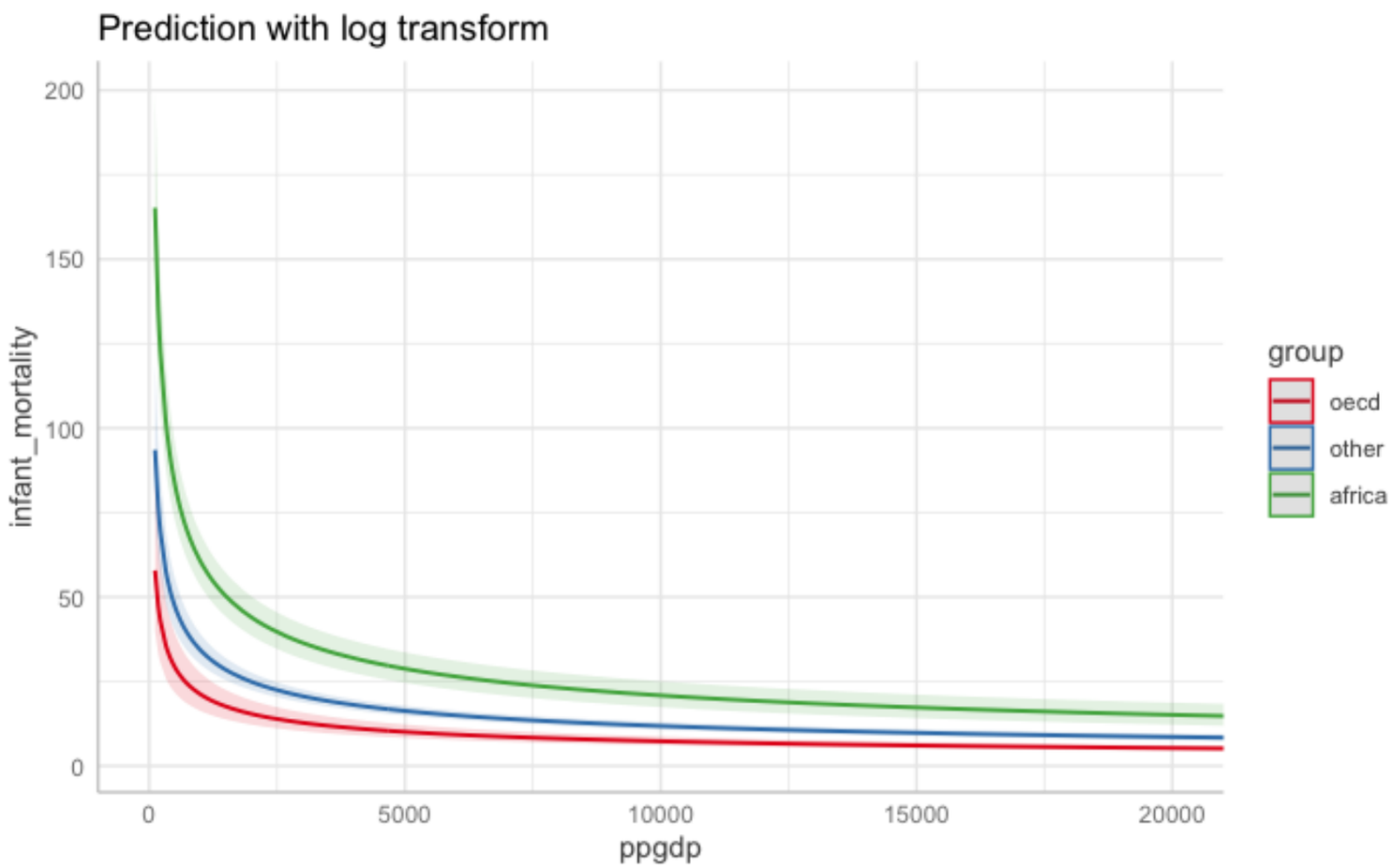
- Influential data points
- Linear and additive
- Data transformation
- Independence
- Non-parametric analysis
- Simulation methods

build Bayesian models that better capture the data-generating process!

How to report statistical results?

Multiple regression

Plots



Tables

	log(infant mortality)			
Predictors	*Estimates*	*CI*	*Statistic*	*p*
(Intercept)	6.26	5.61 – 6.90	19.25	**<0.001**
ppgdp [log]	-0.46	-0.52 – -0.40	-15.28	**<0.001**
group [other]	0.48	0.26 – 0.70	4.36	**<0.001**
group [africa]	1.05	0.76 – 1.34	7.15	**<0.001**
Observations	193			
R^2 / R^2 adjusted	0.819 / 0.816			

Text

We log-transformed the predictor "per capita gross domestic product" (ppgdp) and the dependent variable (infant mortality). A country's ppgdp is a significant predictor of infant mortality (see Table). [give one or two concrete examples to help with interpretation]

https://cran.r-project.org/web/packages/sjPlot/vignettes/tab_model_estimates.html

Ran out of steam ....

I will make a nice RMarkdown file with examples for how to report results!

Thanks to you!

Psych 252 Team

Tyler Bonnen

Andrew Nam

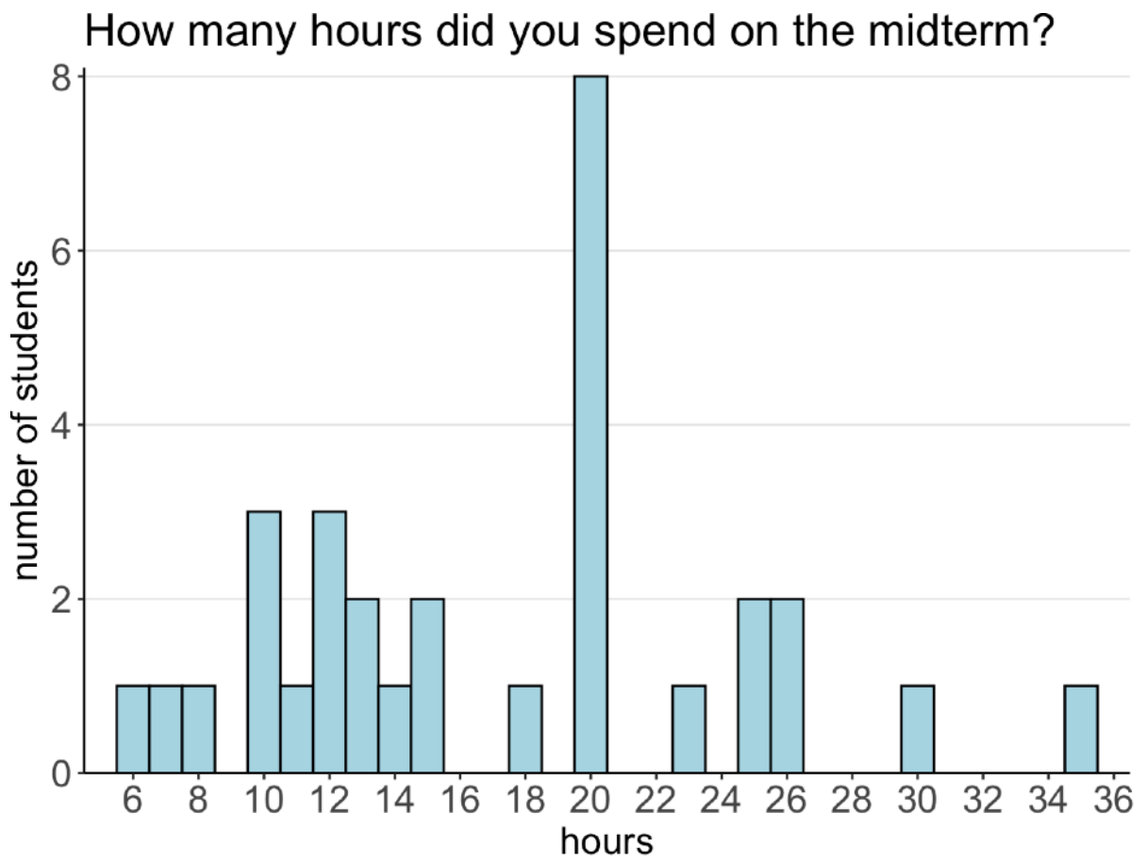
Jinxiao Zhang

All of you!

Shilaan Alzahawi
Ruth Elisabeth Appel
Xubo Cao
Cristina Isabel Ceballos
Zonghe Chua
Aaron Chuey
Will Somers Clapp
Ayo Daniel Dada
Brendan Fereday
Kayla Good
Hope Marie Harrington
Lindsey Hasak
Philip Hernandez
Rebecca Hinds
Hanseul Jun
Summer Jung
Insub Kim
Emily Kubota
Angela Yuson Lee
Effie Li
Chelsea Lide
Mufan Luo
Samina Lutfeali
Marijn Nura Mado
Ashish Mehta
Leili Mortazavi
Elizabeth Mortenson
Maggie Perry
Erika Petersen O Farrill
Jacob William Keith Ritchie
Daphna Lee Spivack
Preeti Srinivasan
Lily Steyer
Nicky Nicholas Sullivan
Omar Patricio Vasquez Duque
Jeremy John Walters
Rondeline Michelle Williams
Justin Ping Yuan

~~Fear~~ of statistics

no more



Thanks for adopting a growth mindset!

fixed mindset:
students believe their basic abilities, their intelligence, their talents, are just fixed traits.

growth mindset:
students *understand* that their talents and abilities can be developed through effort, good teaching and persistence

Vision for this class

In "A Vision for Stanford", university president Marc Tessier-Lavigne states that Stanford wants to be

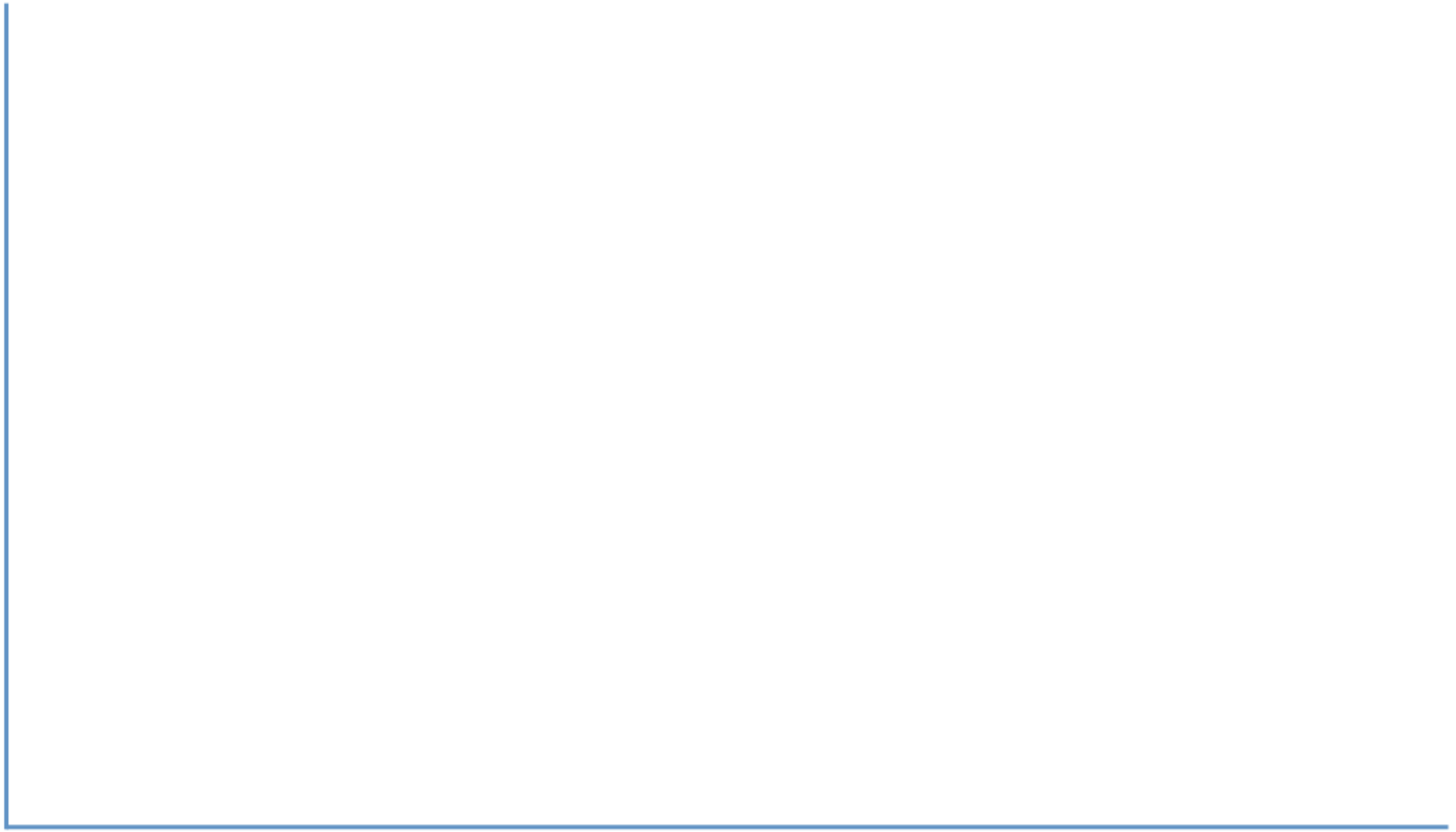
"an inspired, inclusive and collaborative community of diverse scholars, students and staff, where all are supported and empowered to thrive."

Thanks for making it happen!

We're looking forward to your presentations!

Feedback

How was the pace of today's class?



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!