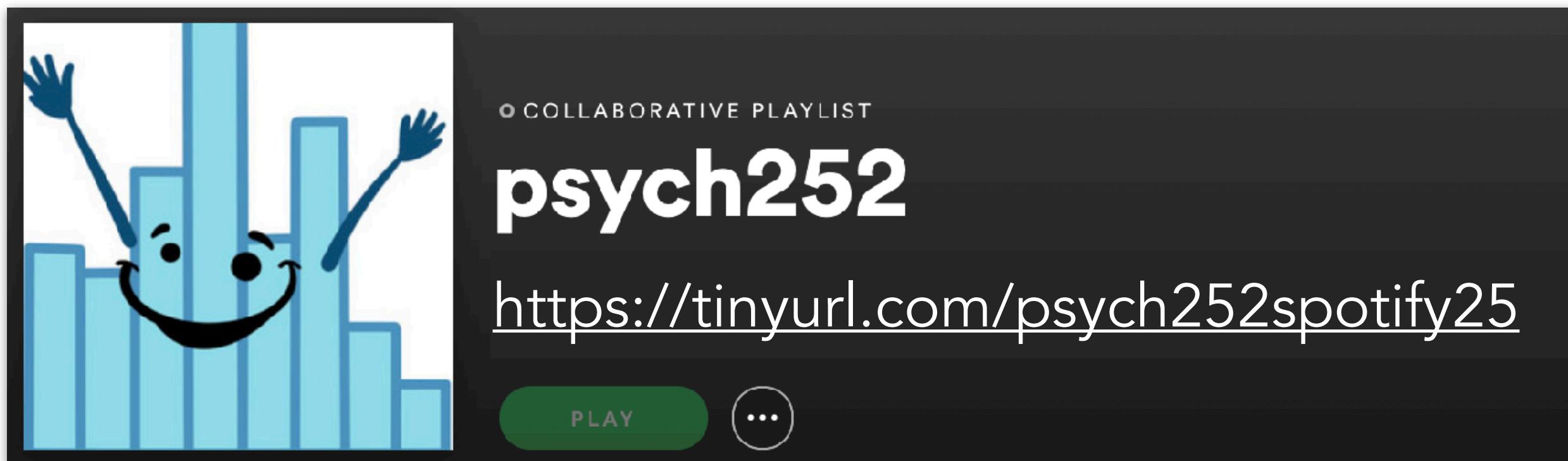
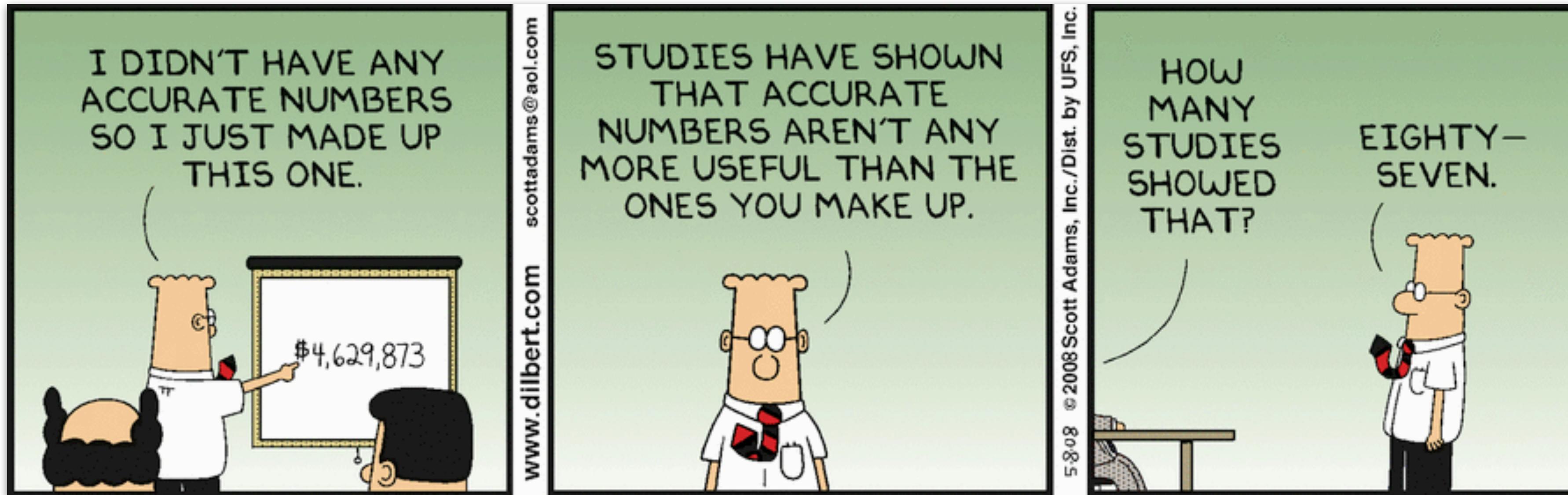


# Linear model 2



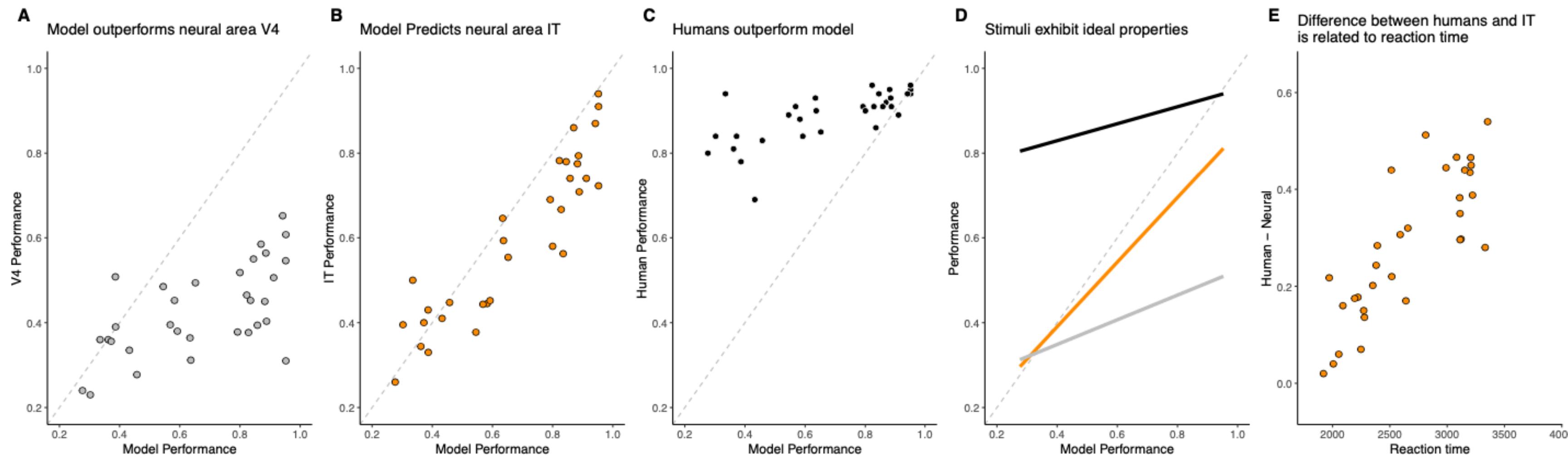
01/31/2025

# **Logistics**

# **Homework 2**

# Homework 2

Grades are posted, and solutions are on Canvas.



# **Homework 4**

# Homework 4: Modeling data

## Homework 4

My name goes here

Collaborator 1

Collaborator 2

January 30, 2025

This homework is due by **Thursday, February 6th, 8:00pm**. Upload both the rendered pdf file (`4_modeling_data.pdf`) and the Rmarkdown code file (`4_modeling_data.rmd`) to Canvas.

### I. Simple linear regression and prediction

In this section and the next, we'll be revisiting the credit dataset.

```
# Load data
df.credit = read_csv("data/credit.csv") %>%
  clean_names()
```

#### 1. Explore and visualize the data

Use the `ggpairs()` function from the `GGally` package to make scatterplots of all pairwise combinations of the numeric variables.

Tip: use `progress = FALSE` within the function to suppress unwanted progress bars of each plot

```
### YOUR CODE HERE ###
```

```
#####
```

That's perhaps a bit too congested to be useful. Recreate the plot, focusing on just ~5 of the variables that seem interesting to you.

```
### YOUR CODE HERE ###
```

```
#####
```

Does your plot tell you anything interesting about the data? Briefly describe one observation.

YOUR ANSWER HERE:

#### 2. Simple linear regression

You decide you want to test the relationship between credit limit (`limit`) and average credit card debt (`balance`).

a) First, let's visualize this relationship. Create a scatterplot of `balance` (on the y-axis) as a function of `limit` (on the x-axis). Set the transparency of the points to 0.3.

```
### YOUR CODE HERE ###
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

```
#####
```

# Midterm

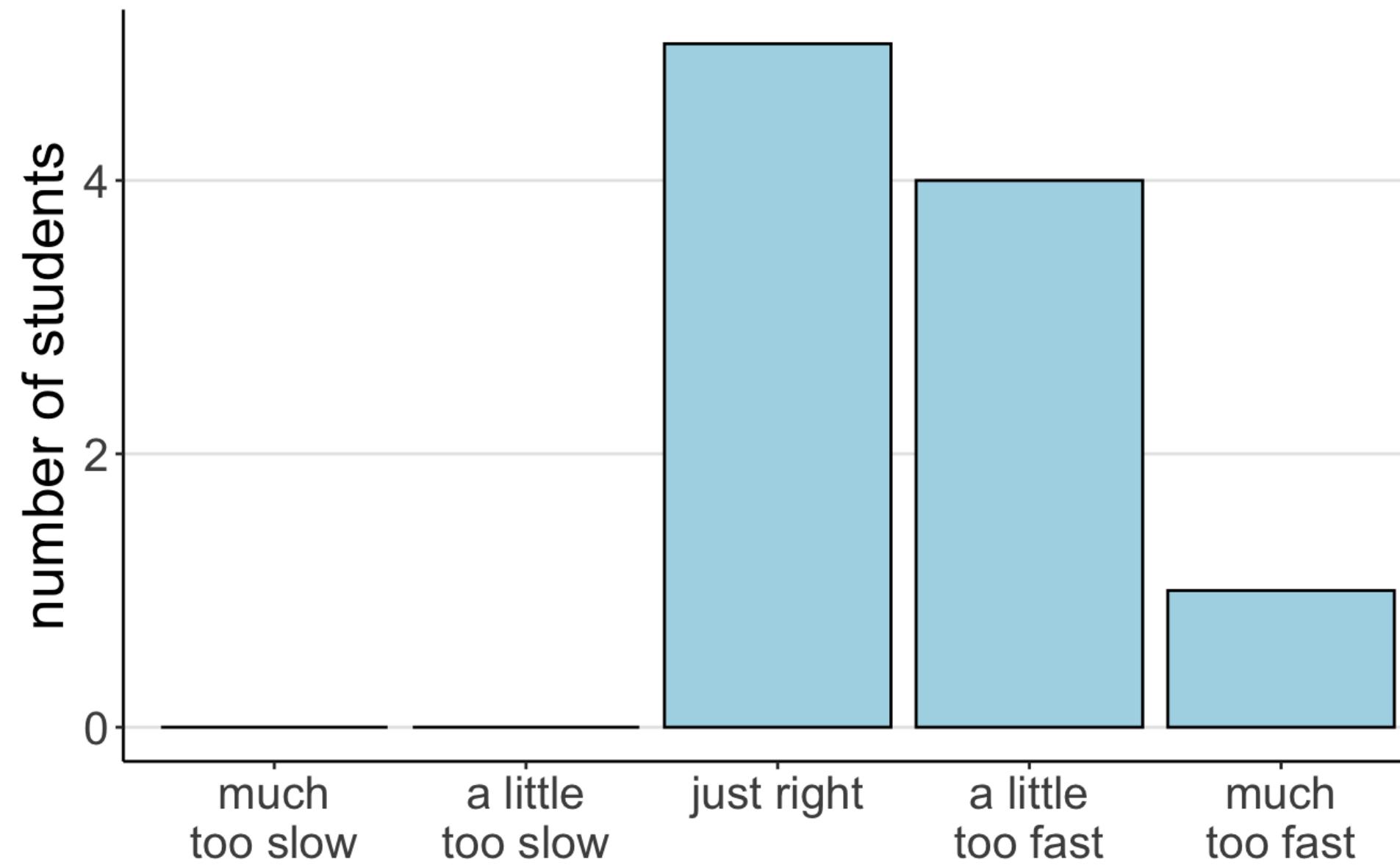
# Midterm

- will be released Friday next week
- midterm is take-home, open exam (you can use all the course notes, all of the internet, etc.)
- is due on **Friday, February 14th at 8pm** (stretch days can't be used for the midterm)
- midterm is like the other homeworks but:
  - it's longer (there won't be class on Wednesday that week)
  - you'll need to work on it on your own
- there won't be office hours or sections in the week that the midterm is due
- ask questions about the midterm on Ed Discussion (please **ask private first** -- we will then respond and, if appropriate, check with you whether we can share with the rest of the class)

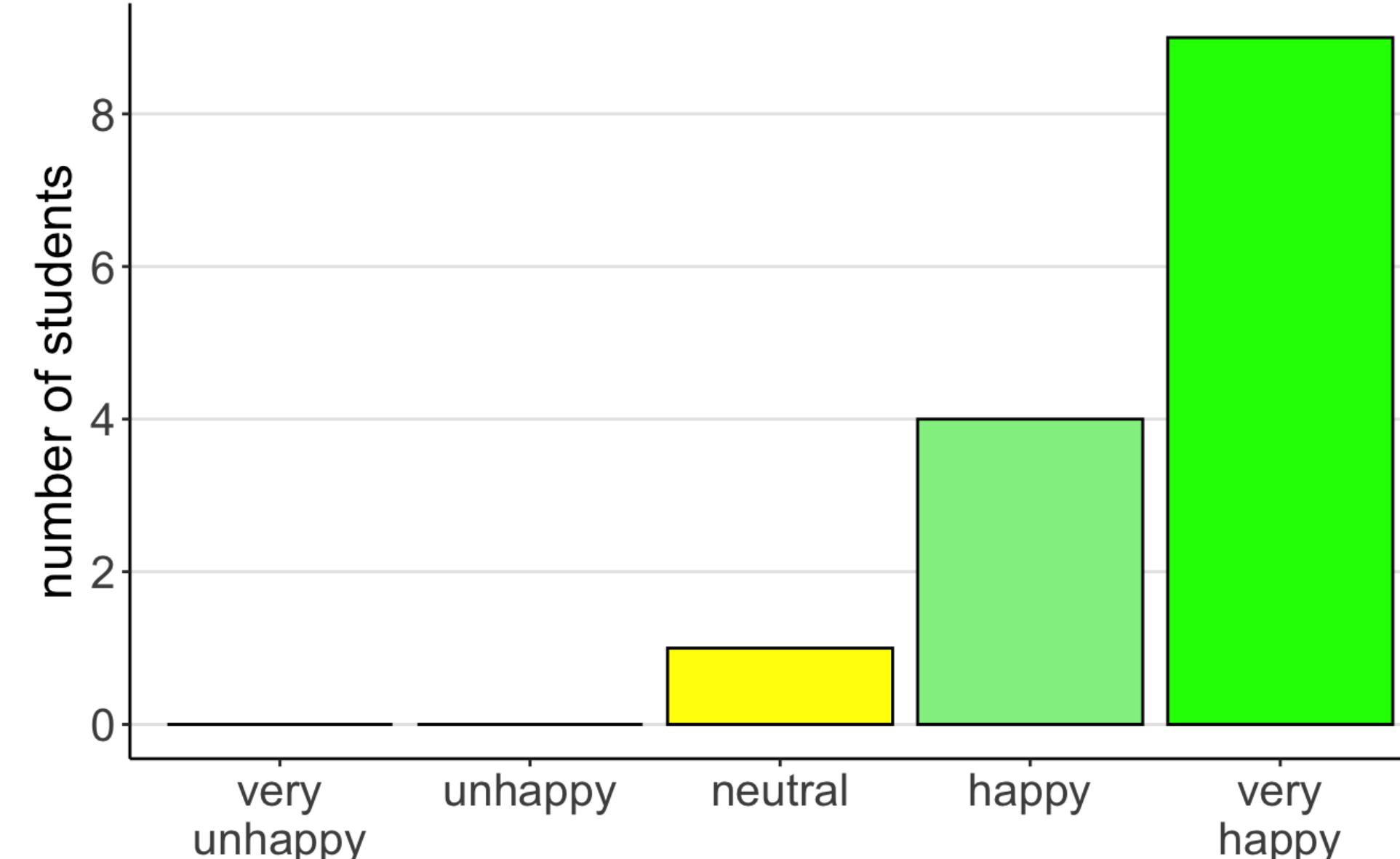
# **Feedback**

# Your feedback

How was the pace of today's class?



How happy were you with today's class overall?



The explanations of concepts and examples are very clear and widely accessible to students with varying stats and programming background. I appreciate the focus on developing both intuition and practical skills, in contrast to many other stats class that seem to prioritize one or the other.

The phase was a **tad too fast**. Also I really really **appreciate the jokes and analogies!** (I really hope they continue) That's the only way I can get through all these STATS :(

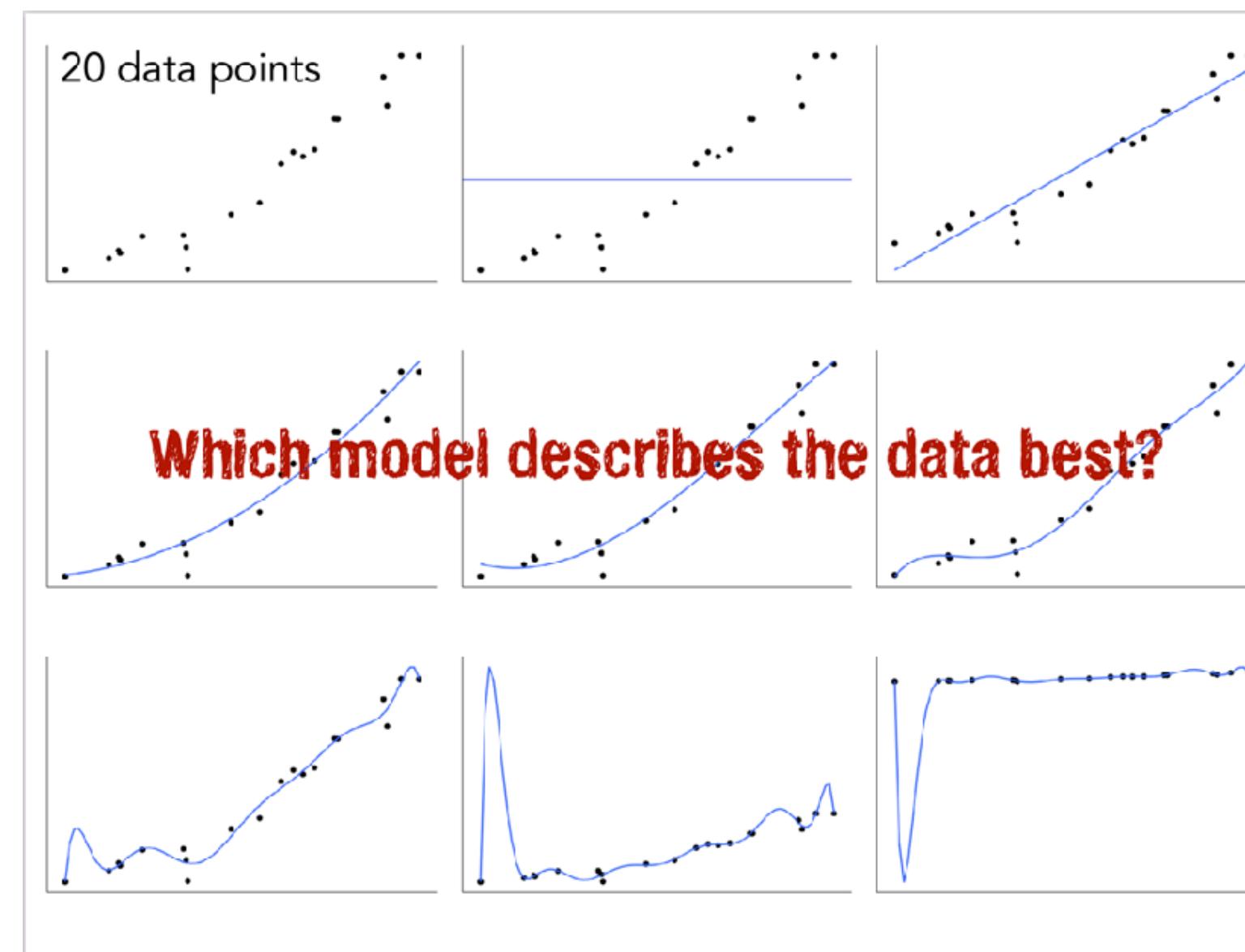
Please **standardize grading** among TAs. maybe TAs can proactively walk around during section and offer help. Please **write in plain speak what the equations are** - helpful to have them in English

# Plan for today

- Quick recap
- Generating a sampling distribution for PRE
- Correlation
- Regression
  - The conceptual tour
  - The R route

# **Quick recap**

# Quick recap: Linear model 1



Compact model

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

Augmented model

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

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

Proportional reduction in error (PRE)

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

- 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  $\text{model}_C$  to create  $\text{model}_A$  but were not is high

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

# Quick recap: Linear model 1

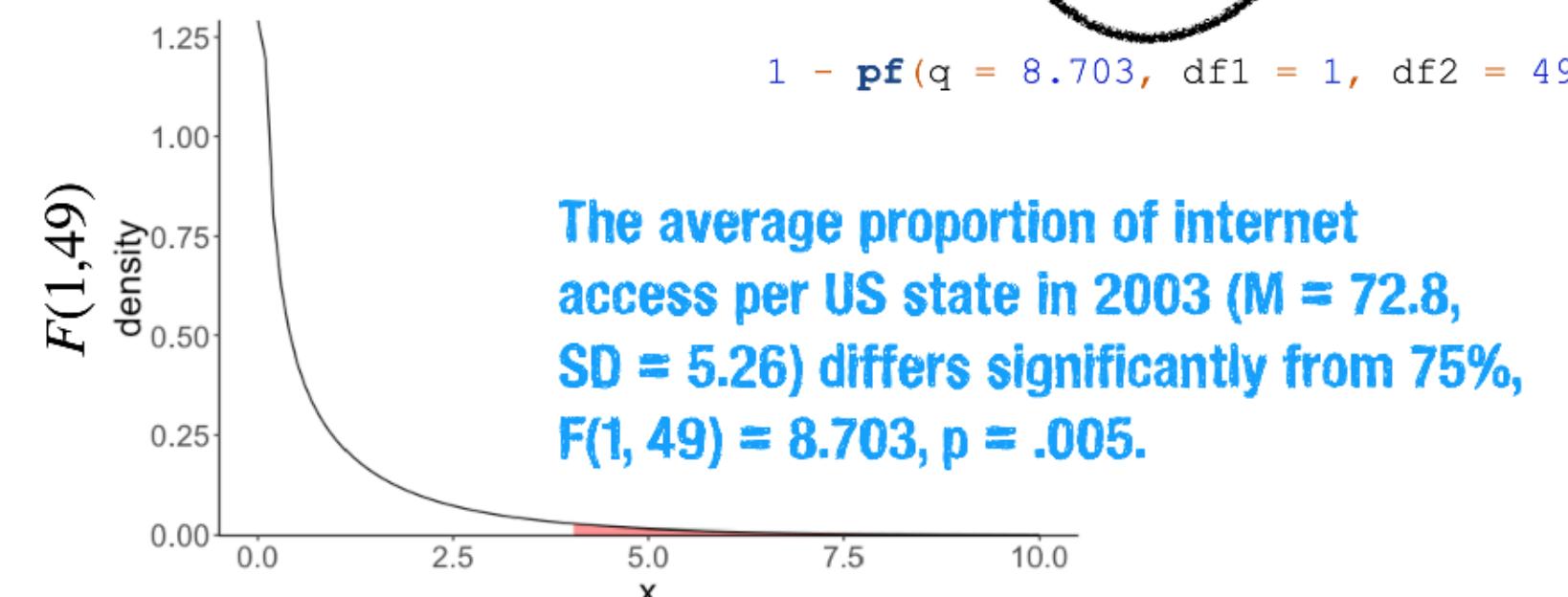
## Procedure

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**

Decide whether it's **worth it**

$$F = \frac{\text{PRE}/(\text{PA} - \text{PC})}{(1 - \text{PRE})/(n - \text{PA})}$$
$$= \frac{.15/(1 - 0)}{(1 - .15)/(50 - 1)} = 8.703 \quad p = .00486$$

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



## Research question and hypotheses

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

Model<sub>C</sub>:  $Y_i = B_0 + \epsilon_i$   
**0 parameters**

$$Y_i = 75 + \epsilon_i$$

Model<sub>A</sub>:  $Y_i = \beta_0 + \epsilon_i$   
**1 parameter**

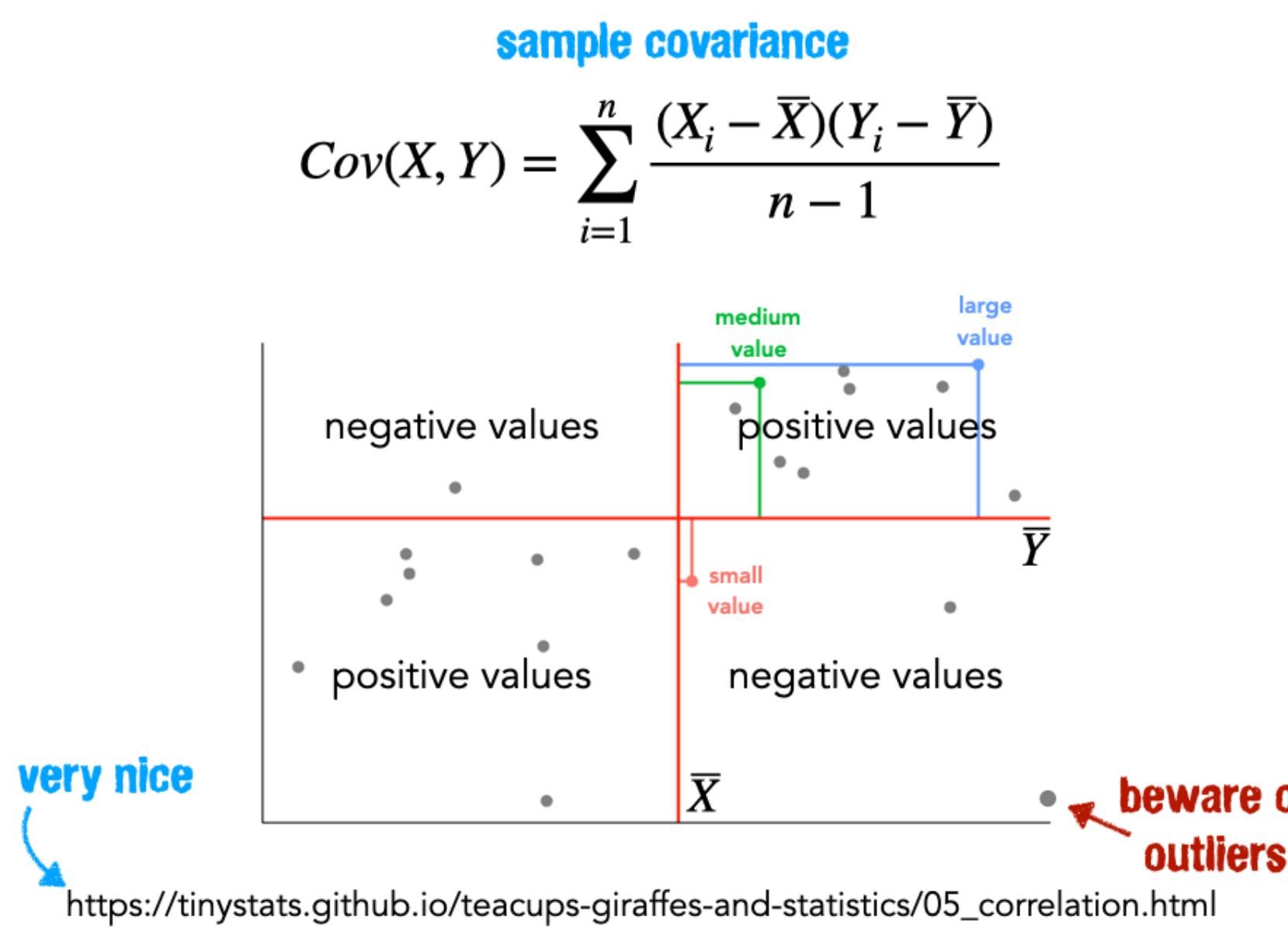
$$Y_i = b_0 + \epsilon_i$$
$$= \bar{Y} + \epsilon_i$$

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

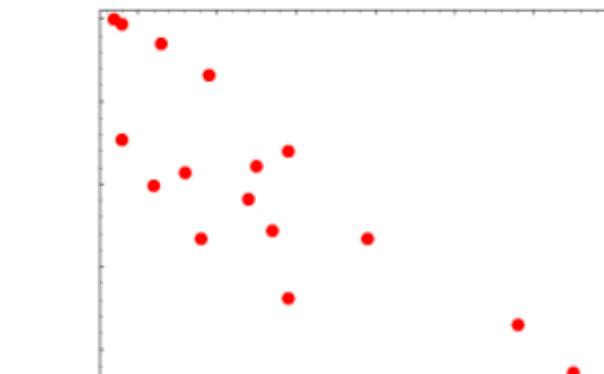
## One Sample t-test

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

# Quick recap: Linear model 1



Who is the correlation champion?

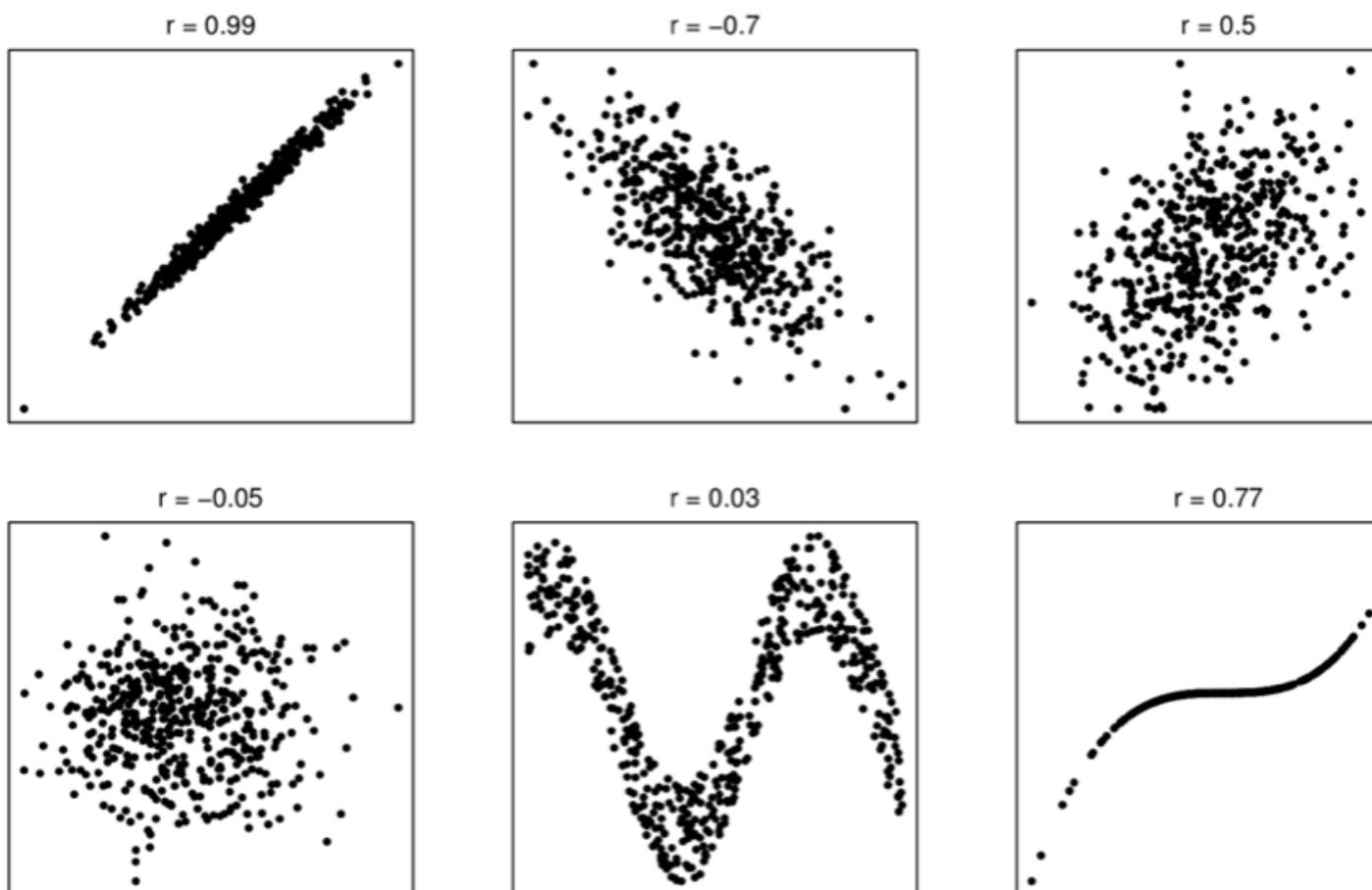


Winner gets chocolate!

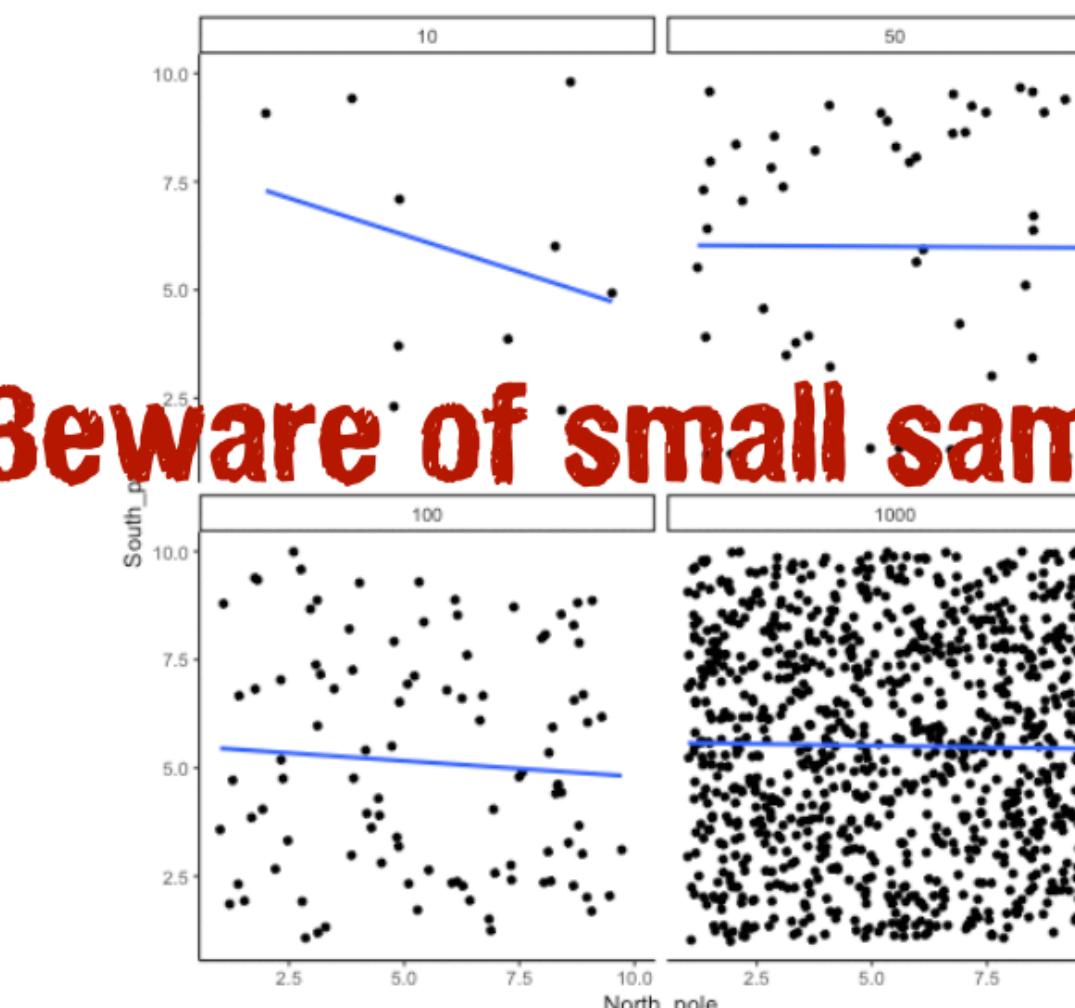
-1 : -0.75
-0.75 : -0.5
-0.5 : -0.25
-0.25 : 0
0 : 0.25
0.25 : 0.5
0.5 : 0.75
0.75 : 1

48

## Solution



$$n = [10, 50, 100, 1000] \quad X \sim \mathcal{U}(\min = 0, \max = 10) \quad Y \sim \mathcal{U}(\min = 0, \max = 10)$$



15

59

# **Generating a sampling distribution for PRE**

# Fit parameters and calculate PRE

$$\mathbf{C: } Y_i = 75 + e_i \quad \mathbf{A: } Y_i = \bar{Y} + e_i$$

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

$$\begin{aligned} \text{PRE} &= 1 - \frac{\text{ERROR}(A)}{\text{ERROR}(C)} \\ &= 1 - \frac{1355}{1595} \approx .15 \end{aligned}$$

Model A has  
15% less error  
than Model C.

$$\text{SSE(C)} = 1595 \quad \text{SSE(A)} = 1355$$

# 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

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

### Model C

$$Y_i = 75 + e_i$$

0 parameters

### Model A

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

1 parameter

# Sampling distribution of PRE

```
1 # simulation parameters
2 n_samples = 1000
3 sample_size = 50
4 mu = 75 # true mean of the distribution
5 sigma = 5 # true standard deviation of the errors
6
7 # function to draw samples from the population distribution
8 fun.draw_sample = function(sample_size, mu, sigma) {
9   sample = mu + rnorm(sample_size, mean = 0, sd = sigma)
10 }
11
12 # draw samples
13 samples = n_samples %>%
14   replicate(fun.draw_sample(sample_size, mu, sigma)) %>%
15   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

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

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

# Sampling distribution of PRE

```

1 # put samples in data frame and compute PRE
2 df.samples = samples %>%
3   as_tibble(.name_repair = ~ 1:ncol(samples)) %>%
4   mutate(sample = 1:n()) %>%
5   pivot_longer(cols = -sample,
6                 names_to = "index",
7                 values_to = "value") %>%
8   mutate(compact = 75) %>%
9   group_by(sample) %>%
10  mutate(augmented = mean(value)) %>%
11  summarize(sse_compact = sum((value - compact)^2),
12            sse_augmented = sum((value - augmented)^2),
13            pre = 1 - sse_augmented/sse_compact)

```

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

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

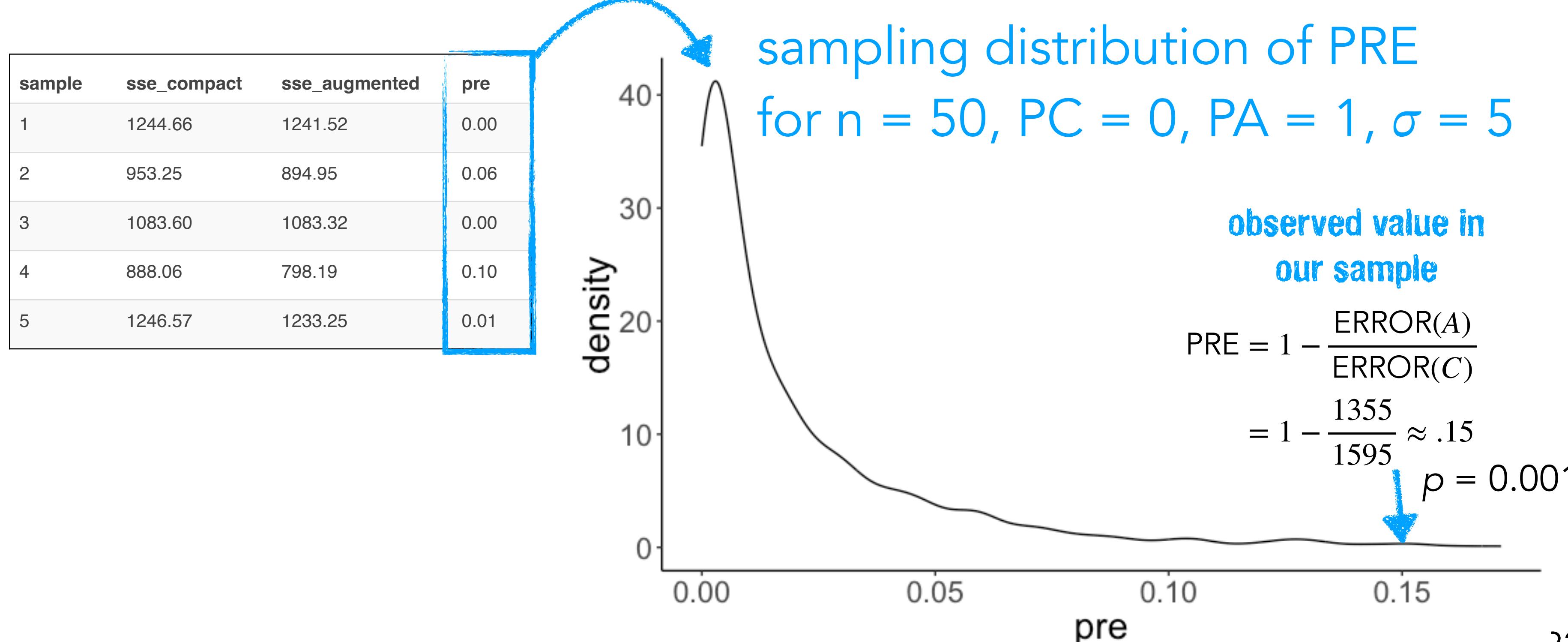
calculate SSE  
for each model

calculate PRE

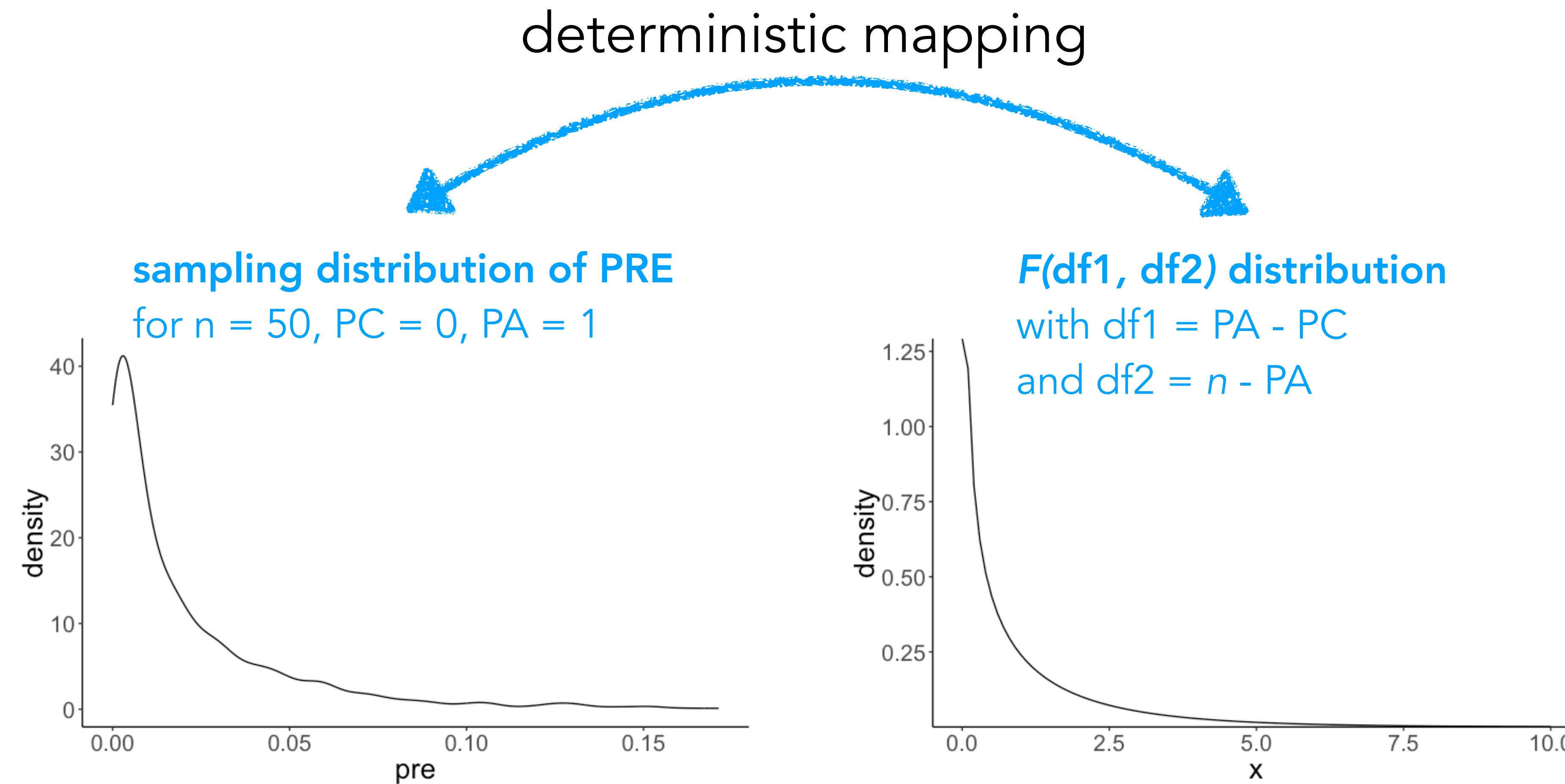
$$PRE = 1 - \frac{SSE_A}{SSE_C}$$

# 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())
```



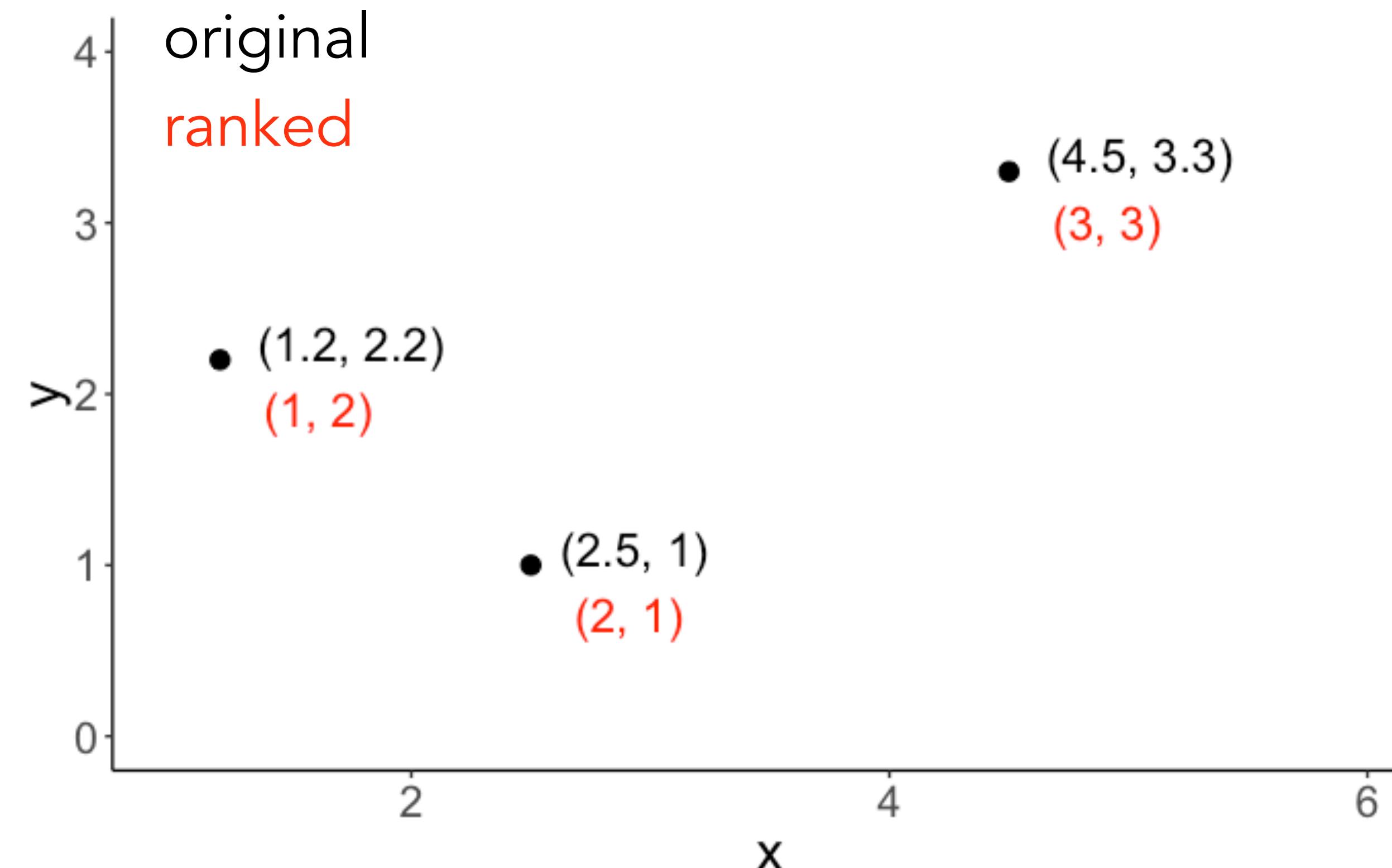
# Sampling distribution of PRE



# Correlation

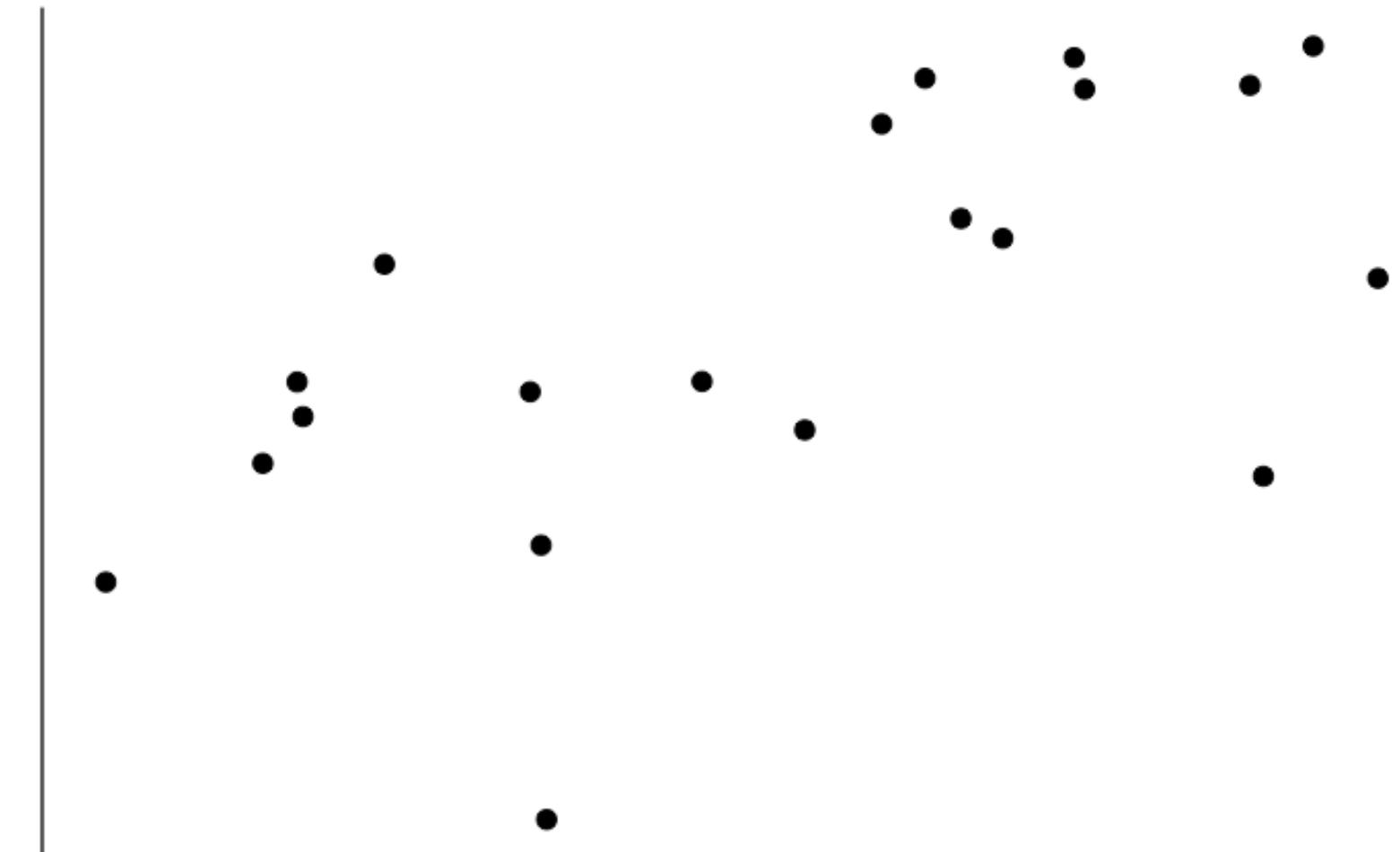
# Spearman rank order correlation

- 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



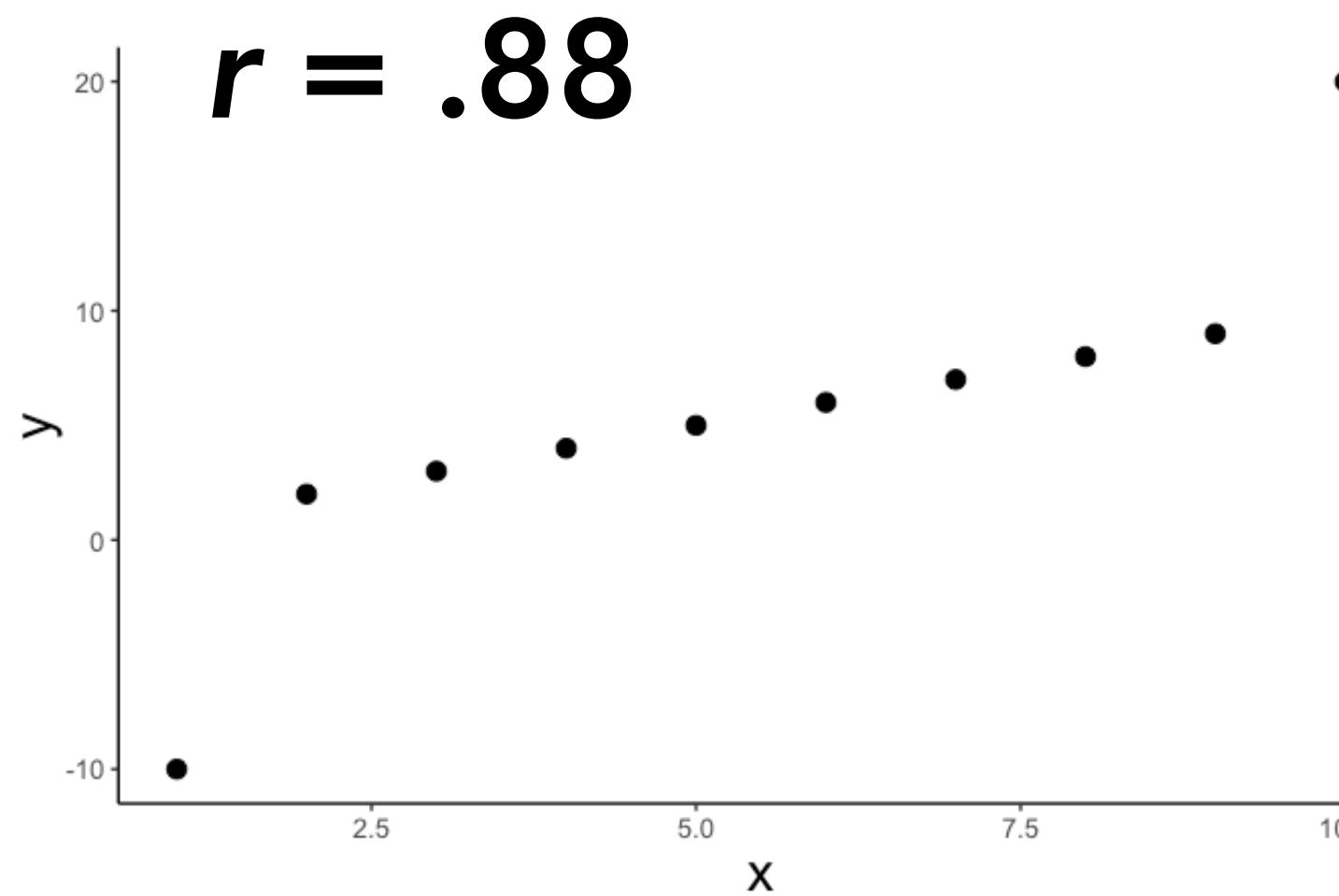
x	y	x_rank	y_rank
0.27	1.14	5	12
0.37	0.97	6	8
0.57	0.92	10	6
0.91	0.85	18	4
0.20	0.98	3	9
0.90	1.39	17	17
0.94	1.44	19	20
0.66	1.40	12	18
0.63	1.33	11	15
0.06	0.71	1	2

r	spearman	r_ranks
0.609	0.595	0.595

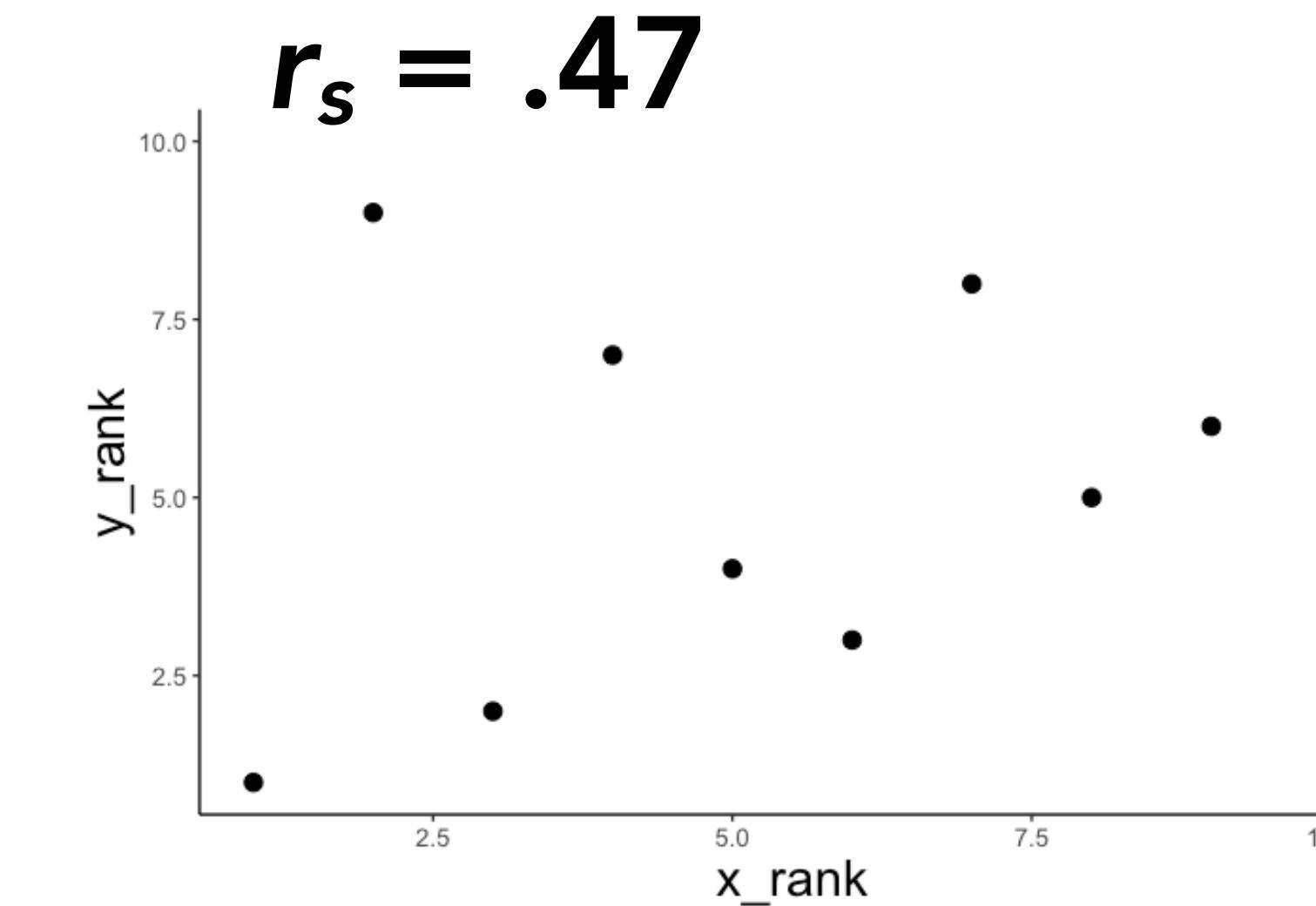
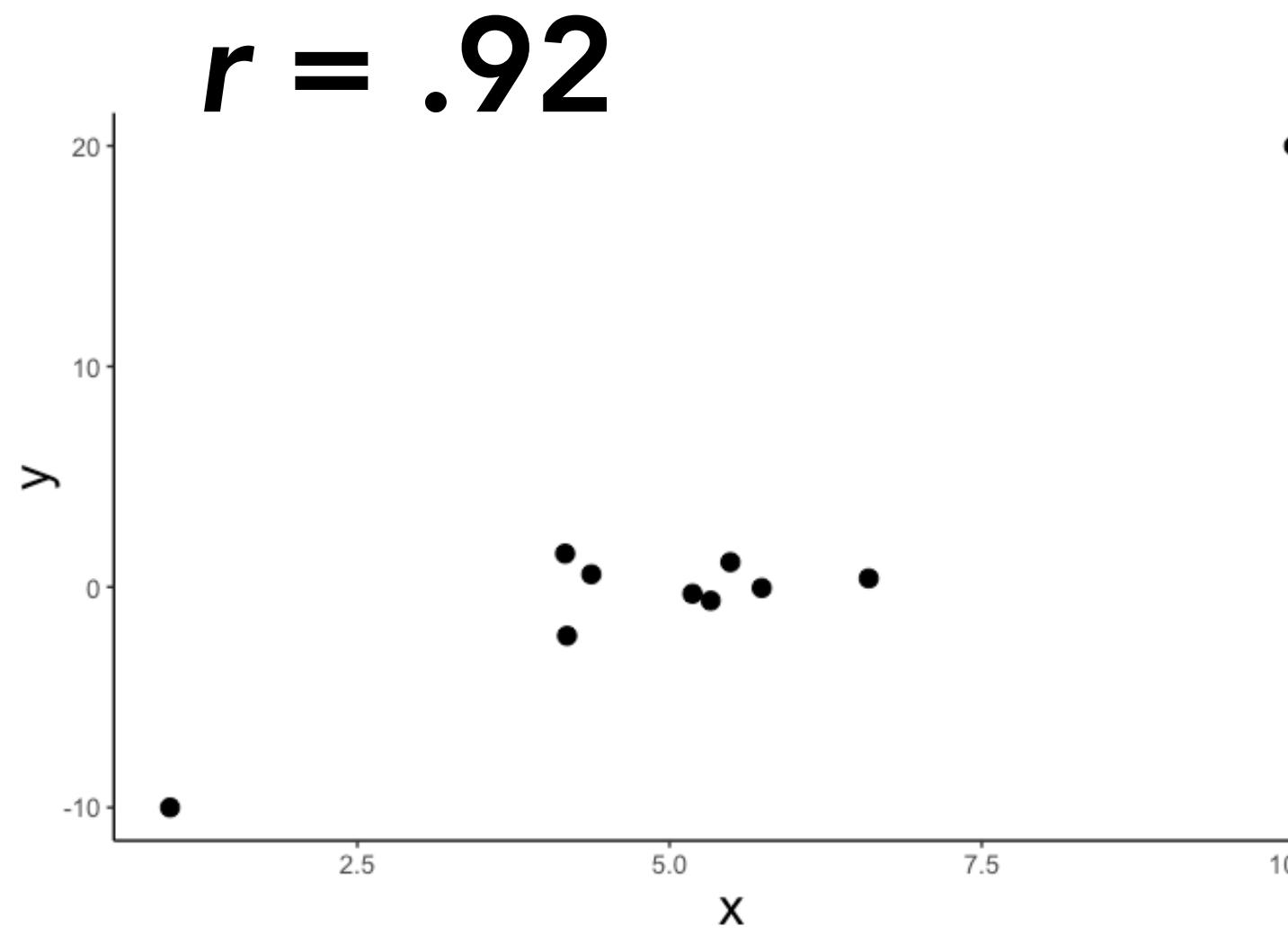
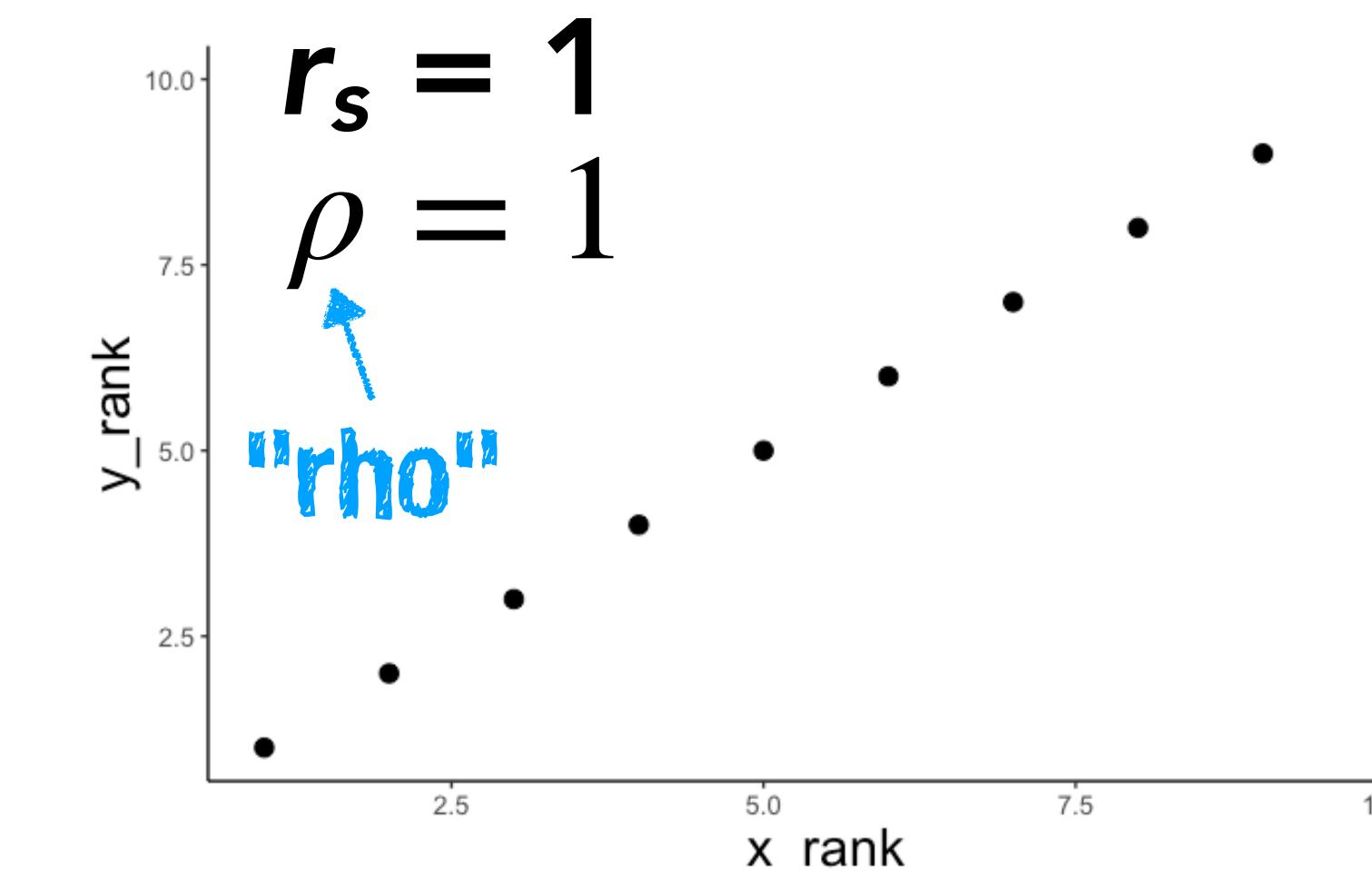
```
1 # correlation
2 df.spearman %>%
3   summarize(r = cor(x, y, method = "pearson"),
4             spearman = cor(x, y, method = "spearman"),
5             r_ranks = cor(x_rank, y_rank, method = "pearson"))
```

# Spearman rank order correlation

original



ranked



# Pearson vs. Spearman

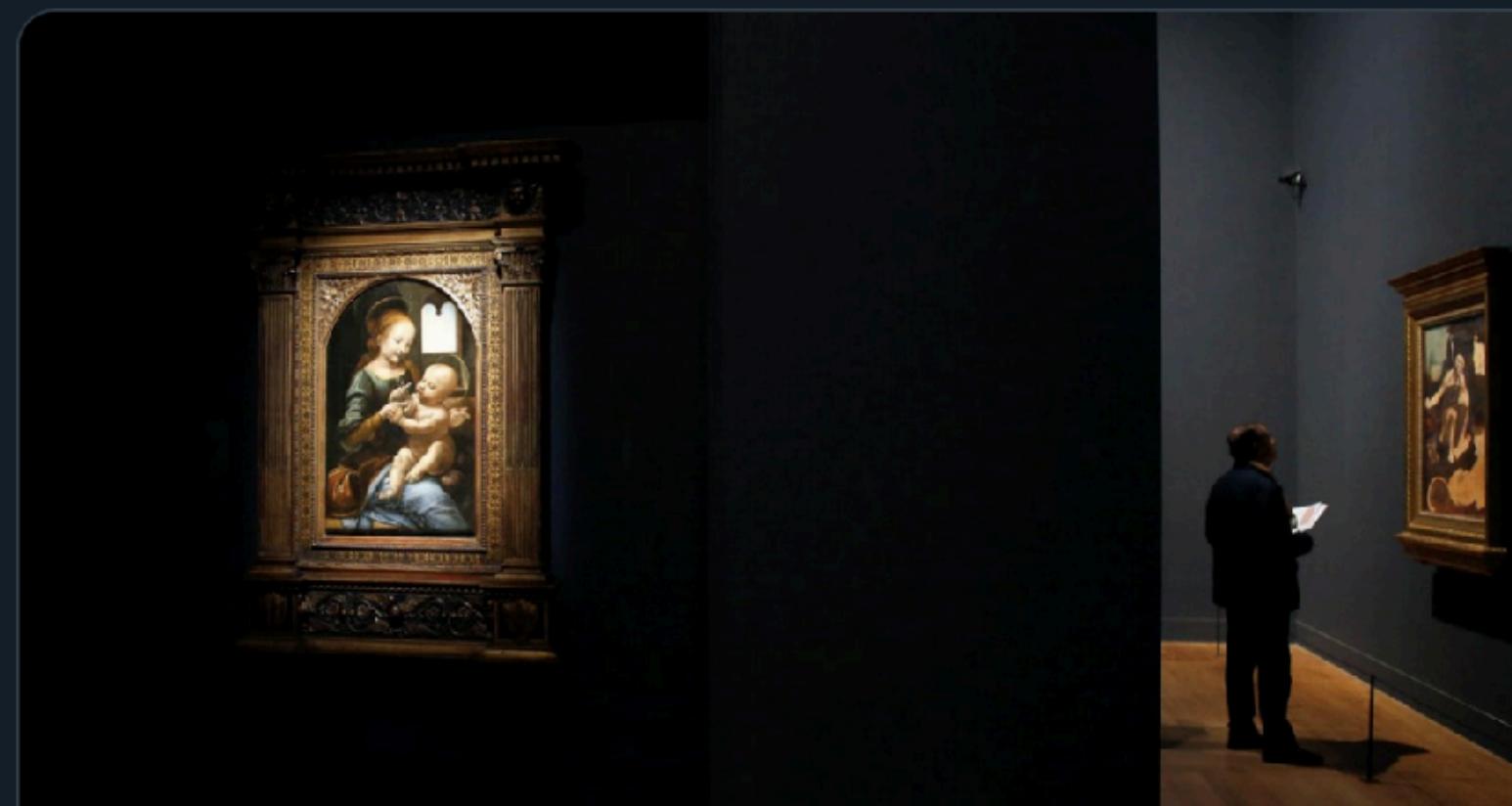
- Pearson's  $r$  captures the extent to which the relationship between two variable is **linear**
- Spearman's  $\rho$  captures the extent to which the relationship between two variables is **monotonic**
- What's better?
  - depends on the context
  - Spearman is robust to outliers, but it throws away (potentially useful) information

# CORRELATION IS NOT CAUSATION



NYT Health  
@NYTHealth

Want to live longer? Try going to the opera. Researchers in Britain have found that people who reported going to a museum or concert even once a year lived longer than those who didn't.



Another Benefit to Going to Museums? You May Live Longer

Researchers in Britain found that people who go to museums, the theater and the opera were less likely to die in the study period than those who didn't.

[nytimes.com](#)

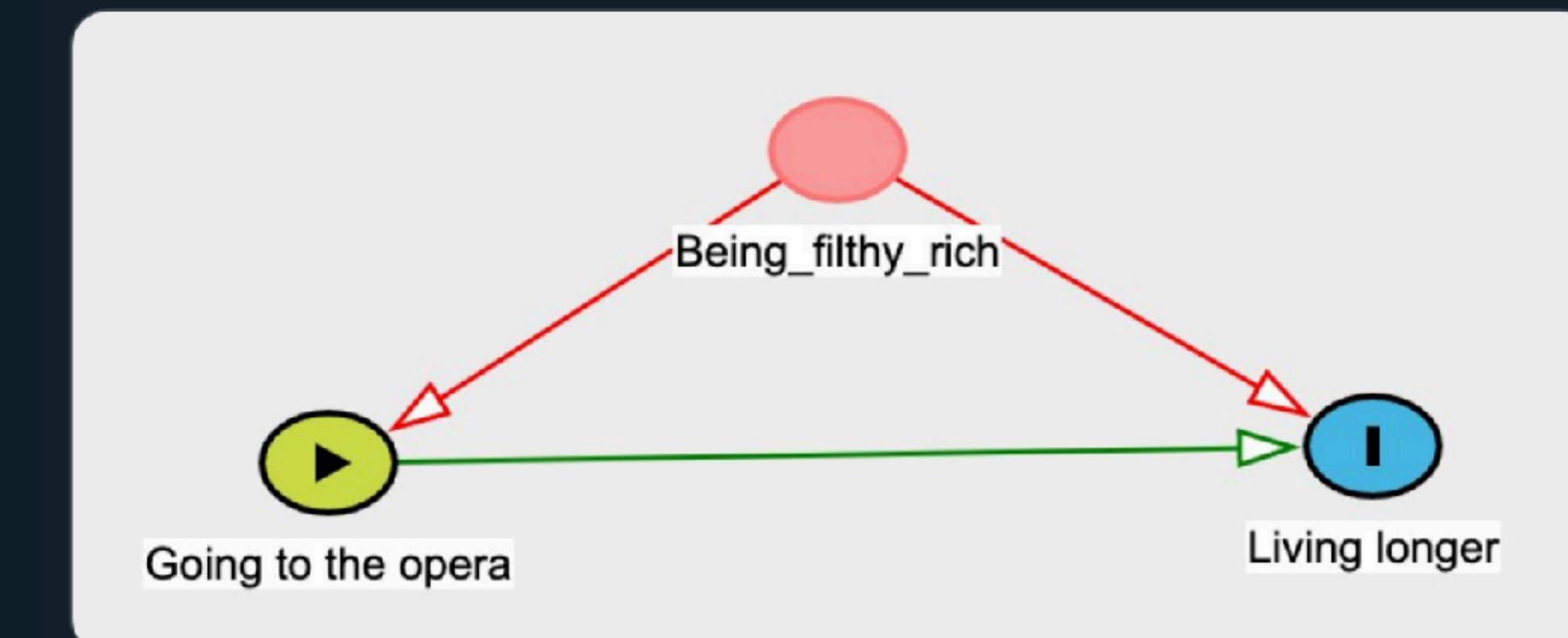
9:19 AM · Dec 22, 2019 · SocialFlow

336 Retweets 1.3K Likes



Andrew Heiss  
@andrewheiss

ooh ooh i can draw the dag for this one!



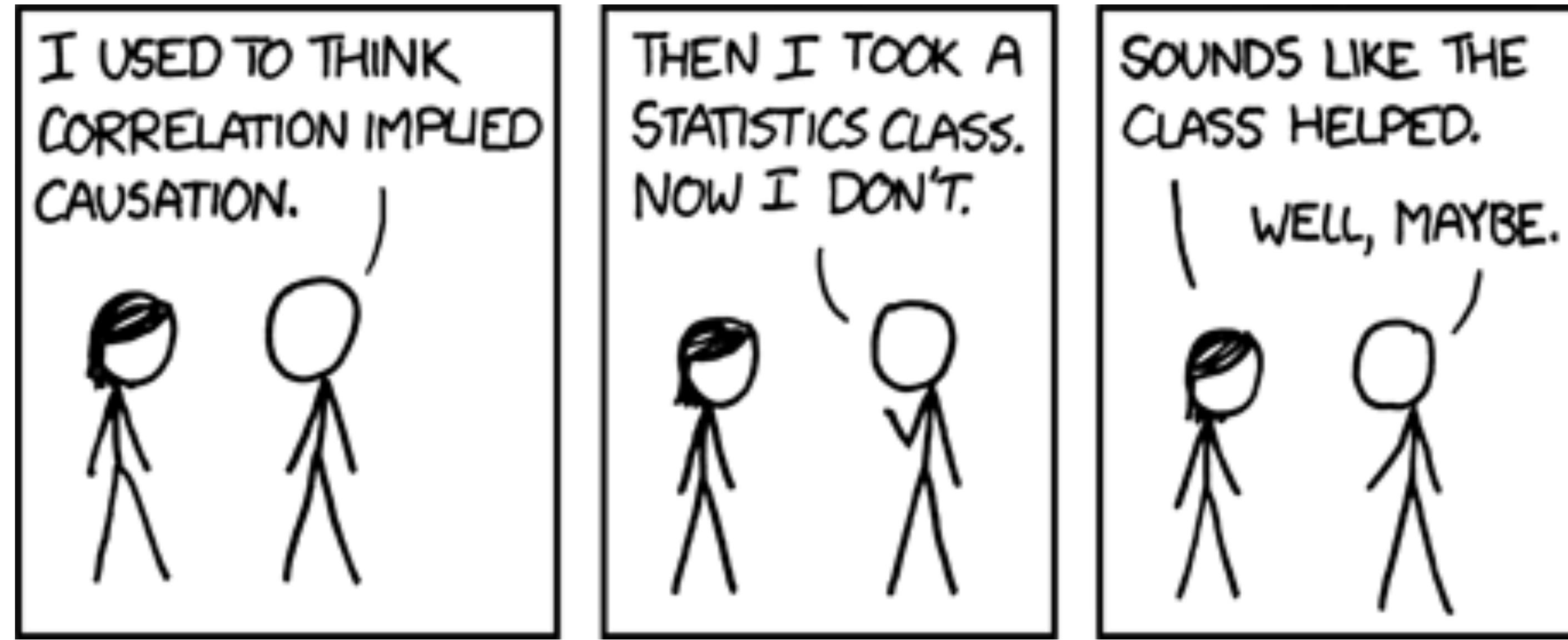
NYT Health @NYTHealth · Dec 22, 2019

Want to live longer? Try going to the opera. Researchers in Britain have found that people who reported going to a museum or concert even once a year lived longer than those who didn't. [nyti.ms/2Q9AmZV](#)

2:47 PM · Dec 22, 2019 · Twitter Web App

[View Tweet activity](#)

837 Retweets 3.9K Likes



- correlations suggest that there is some causal relationship
- but this relationship need not be a direct causal relationship from A to B (or from B to A)

more about causation in a later class

# **Regression**

# **The conceptual tour**

# Linear model: Simple regression

Data = Model + Error

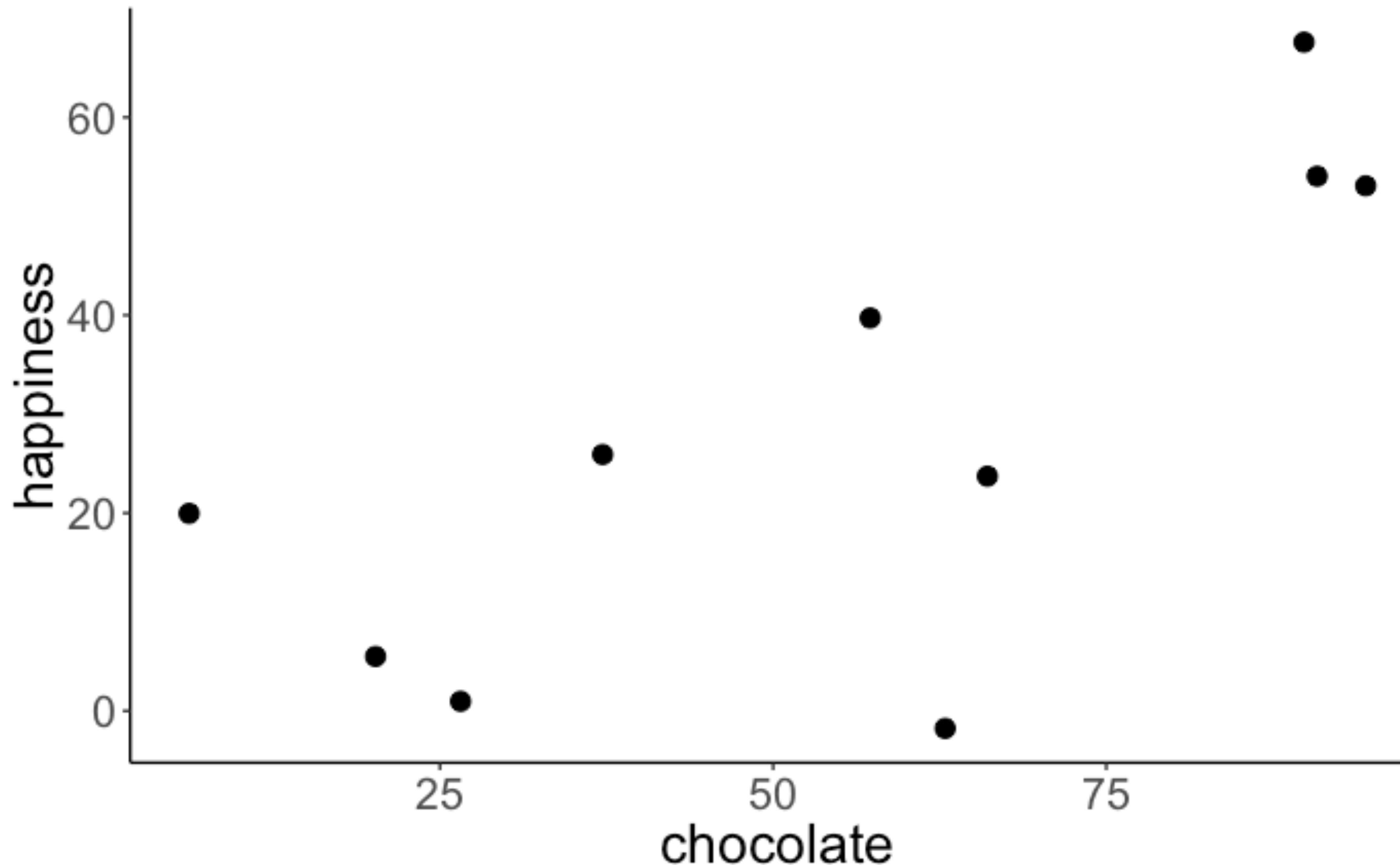
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$



the model is a linear  
combination of predictors

# Is there a relationship between chocolate consumption and happiness?



# 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)

# 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)

$H_0$ : Chocolate consumption and happiness are unrelated.

### Model C

$$Y_i = \beta_0 + \epsilon_i$$

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

and

$$\beta_1 = 0$$

$H_1$ : Chocolate consumption and happiness are related.

### Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

chocolate  
consumption

# The general procedure

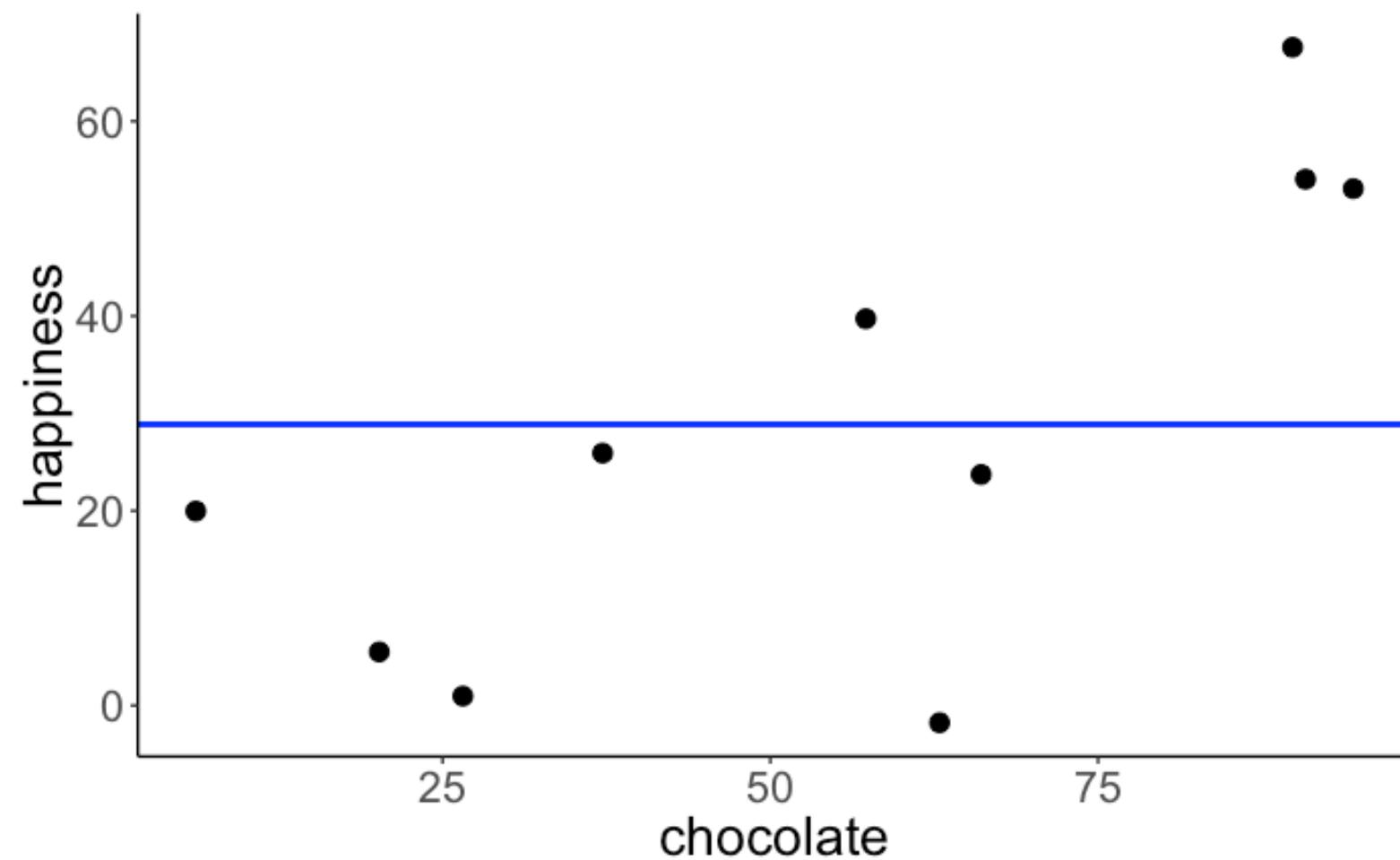
1. Define  $H_0$  as Model C (compact) and  $H_1$  as Model A (augmented)
- 2. Fit model parameters to the data**
3. Calculate the proportional reduction of error (PRE) in our sample
4. Decide whether the augmented model is **worth it** by comparing the observed PRE in our sample to the sampling distribution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

$H_0$ : Chocolate consumption and happiness are unrelated.

### Model C

$$Y_i = \beta_0 + \epsilon_i$$

### Model prediction



### Fitted model

$$Y_i = 28.88 + e_i$$

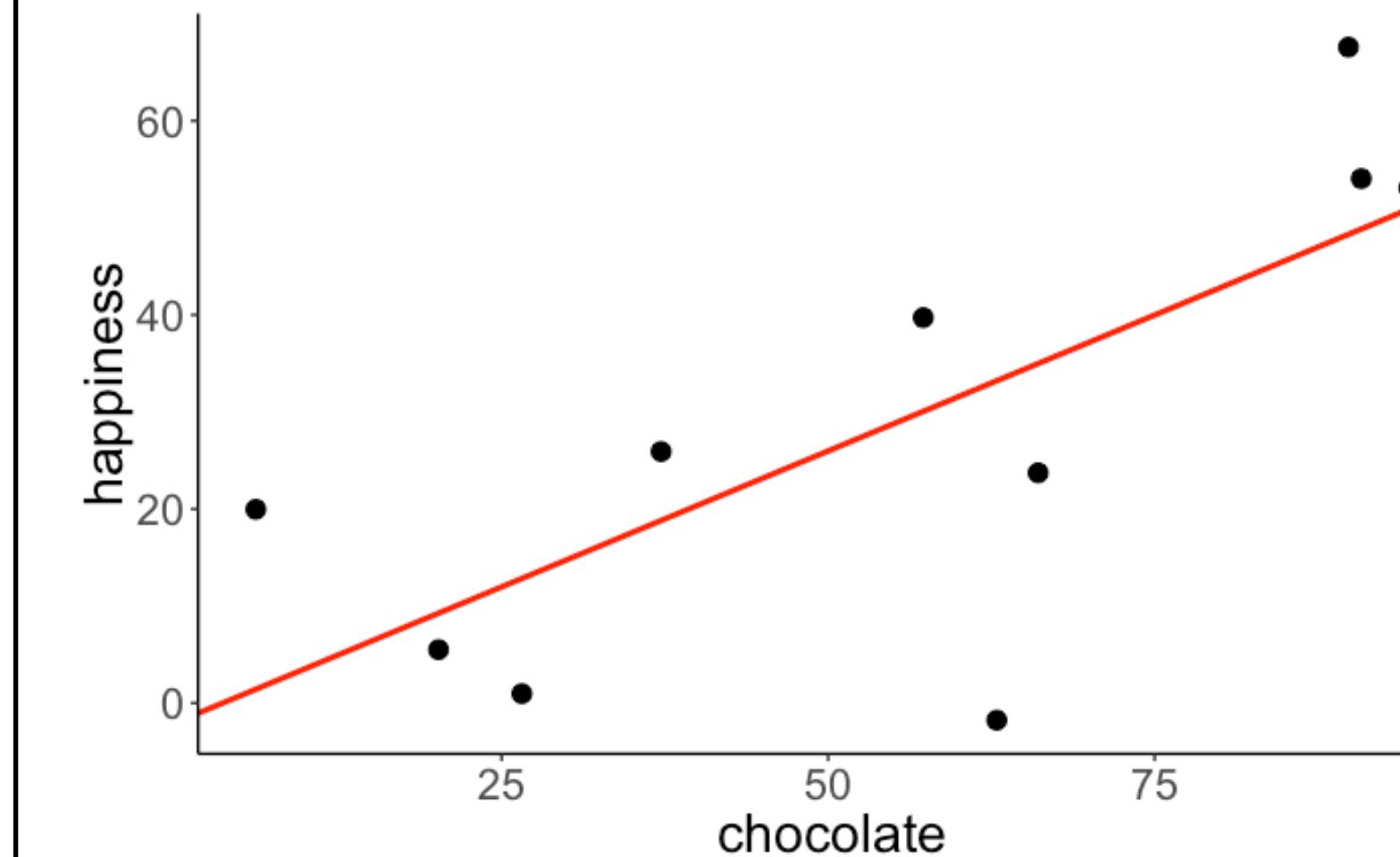
$H_1$ : Chocolate consumption and happiness are related.

### Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

chocolate  
consumption

### Model prediction



### Fitted model

$$Y_i = -2.04 + 0.56X_i + e_i$$

# The general procedure

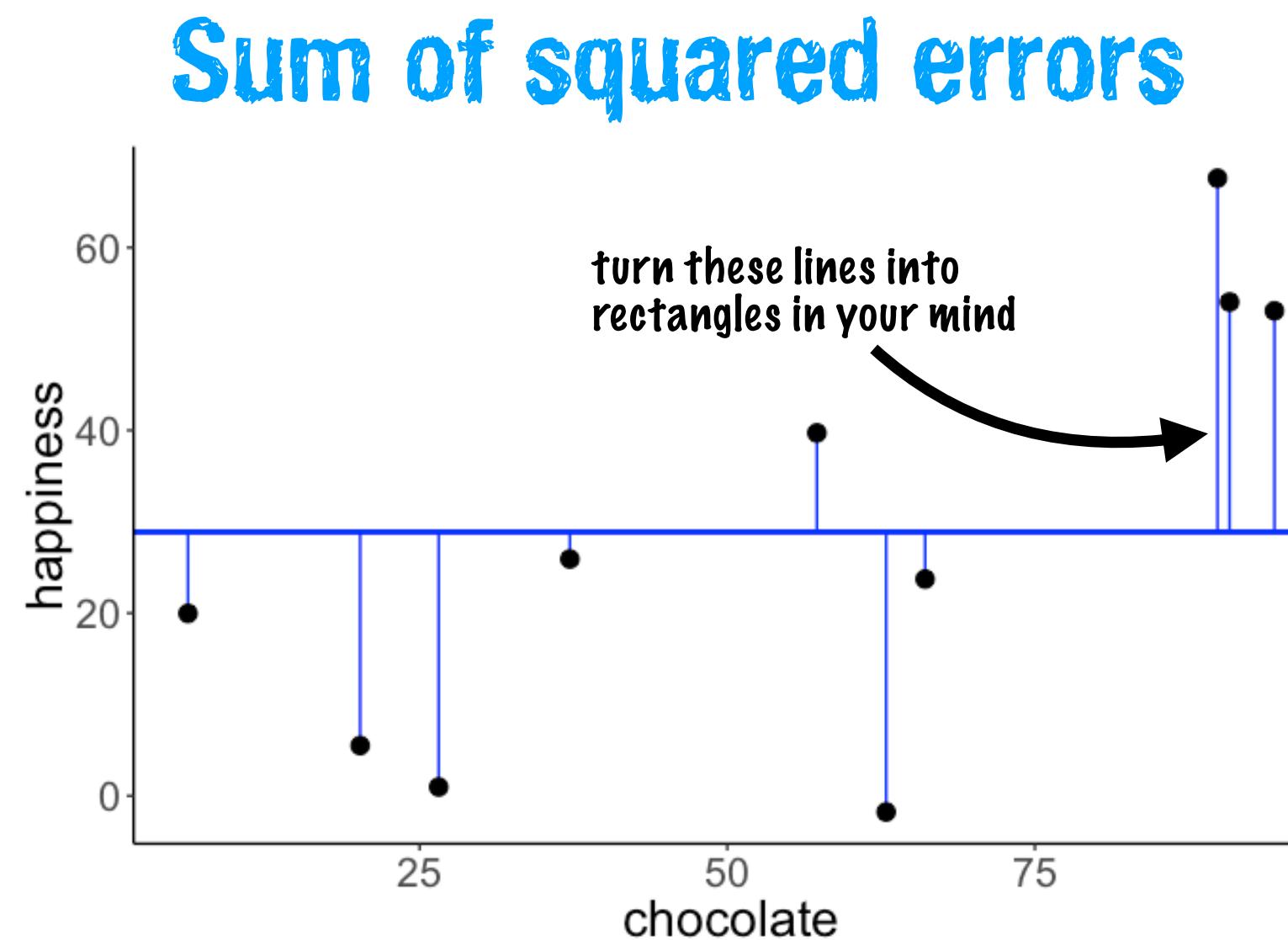
1. Define  $H_0$  as Model C (compact) and  $H_1$  as Model A (augmented)
2. Fit model parameters to the data
3. **Calculate the proportional reduction of error (PRE) in our sample**
4. Decide whether the augmented model is **worth it** by comparing the observed PRE in our sample to the sampling distribution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

# Calculate PRE

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

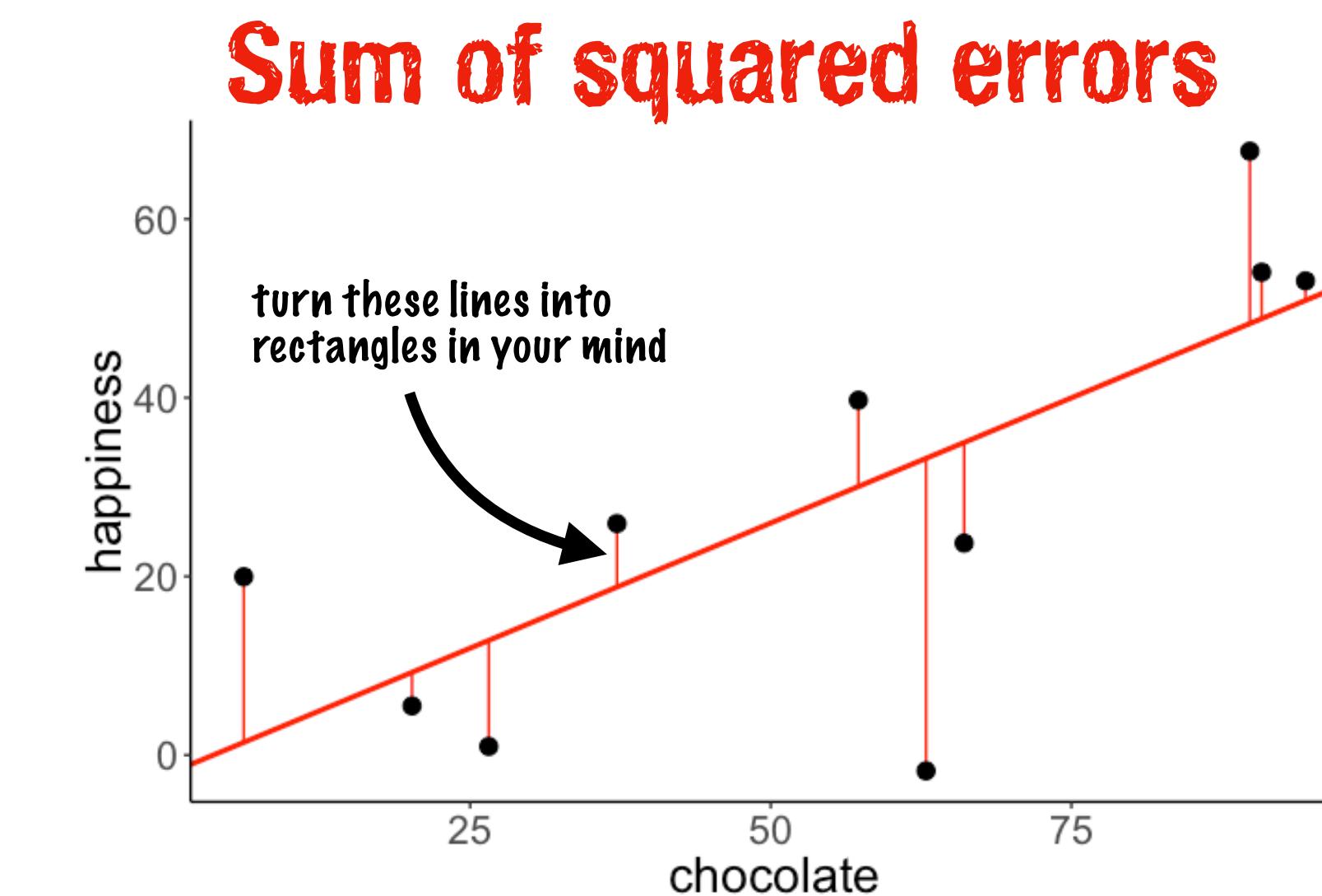
Both models were fit to minimize the sum of squared errors

OLS = Ordinary **least squares** regression



$$\text{SSE}(C) = 5215.016$$

$$\text{PRE} = 1 - \frac{2396.946}{5215.016} \approx 0.54$$



$$\text{SSE}(A) = 2396.946$$

**The augmented model  
reduces the error by 54%.**

# 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)

# 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<sub>C</sub> to create model<sub>A</sub> but were not

**difference in parameters  
between models A and C**

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

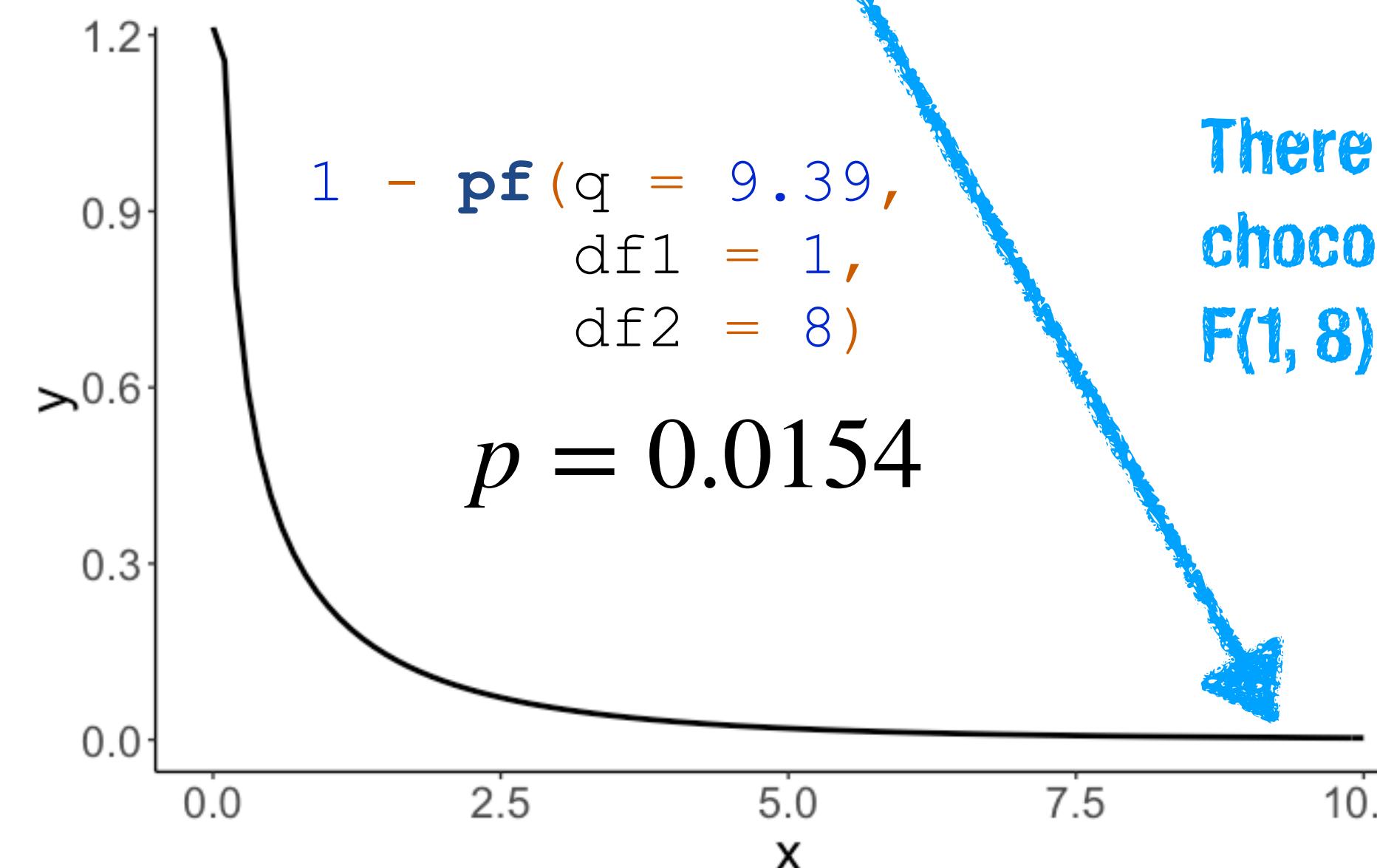
**number of observations  
vs. parameters in Model**

# Decide whether it's **worth it**

- To compute the  $F$  statistic, we need:

- PRE = 0.54
- PC = 1
- PA = 2
- $n$  = 10

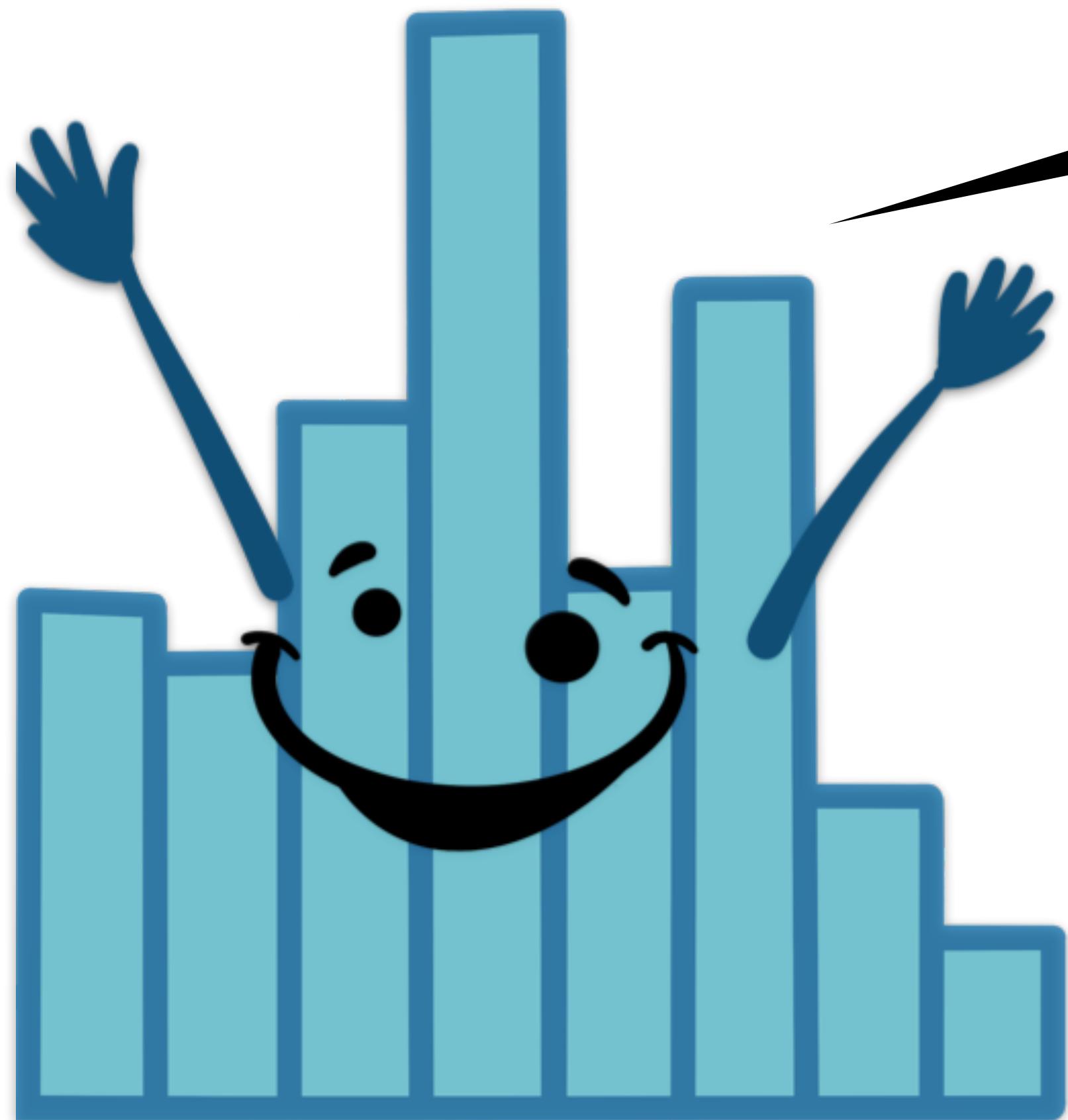
$$\begin{aligned} F &= \frac{\text{PRE}/(\text{PA} - \text{PC})}{(1 - \text{PRE})/(n - \text{PA})} \\ &= \frac{0.54/(2 - 1)}{(1 - 0.54)/(10 - 2)} \\ &= 9.39 \end{aligned}$$



**There is a significant relationship between chocolate consumption and happiness**  
 $F(1, 8) = 9.39, p = .0154.$

02:00

stretch break!



I LIKE TO KICK... STRETCH... AND KICK!

# **The R route**

# Credit card debt



# Credit data set

df.credit

index	income	limit	rating	cards	age	education	gender	student	married	ethnicity	balance
1	14.89	3606	283	2	34	11	Male	No	Yes	Caucasian	333
2	106.03	6645	483	3	82	15	Female	Yes	Yes	Asian	903
3	104.59	7075	514	4	71	11	Male	No	No	Asian	580
4	148.92	9504	681	3	36	11	Female	No	No	Asian	964
5	55.88	4897	357	2	68	16	Male	No	Yes	Caucasian	331
6	80.18	8047	569	4	77	10	Male	No	No	Caucasian	1151
7	21.00	3388	259	2	37	12	Female	No	No	African American	203
8	71.41	7114	512	2	87	9	Male	No	No	Asian	872
9	15.12	3300	266	5	66	13	Female	No	No	Caucasian	279
10	71.06	6819	491	3	41	19	Female	Yes	Yes	African American	1350

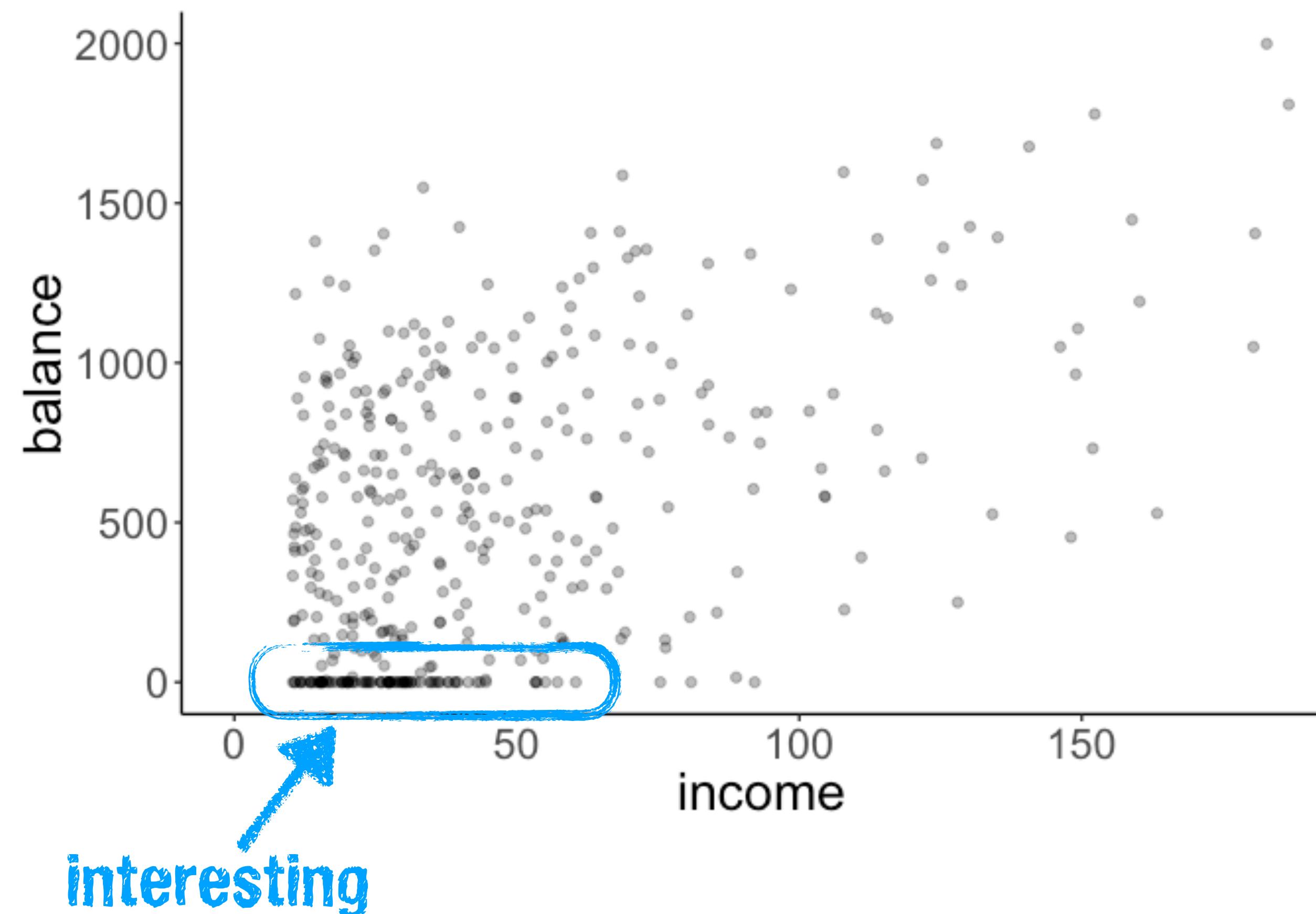
variable	description
income	in thousand dollars
limit	credit limit
rating	credit rating
cards	number of credit cards
age	in years
education	years of education
gender	male or female
student	student or not
married	married or not
ethnicity	African American, Asian, Caucasian
balance	average credit card debt in dollars

**nrow(df.credit) = 400**

**Is there a relationship between income and the average credit card debt?**

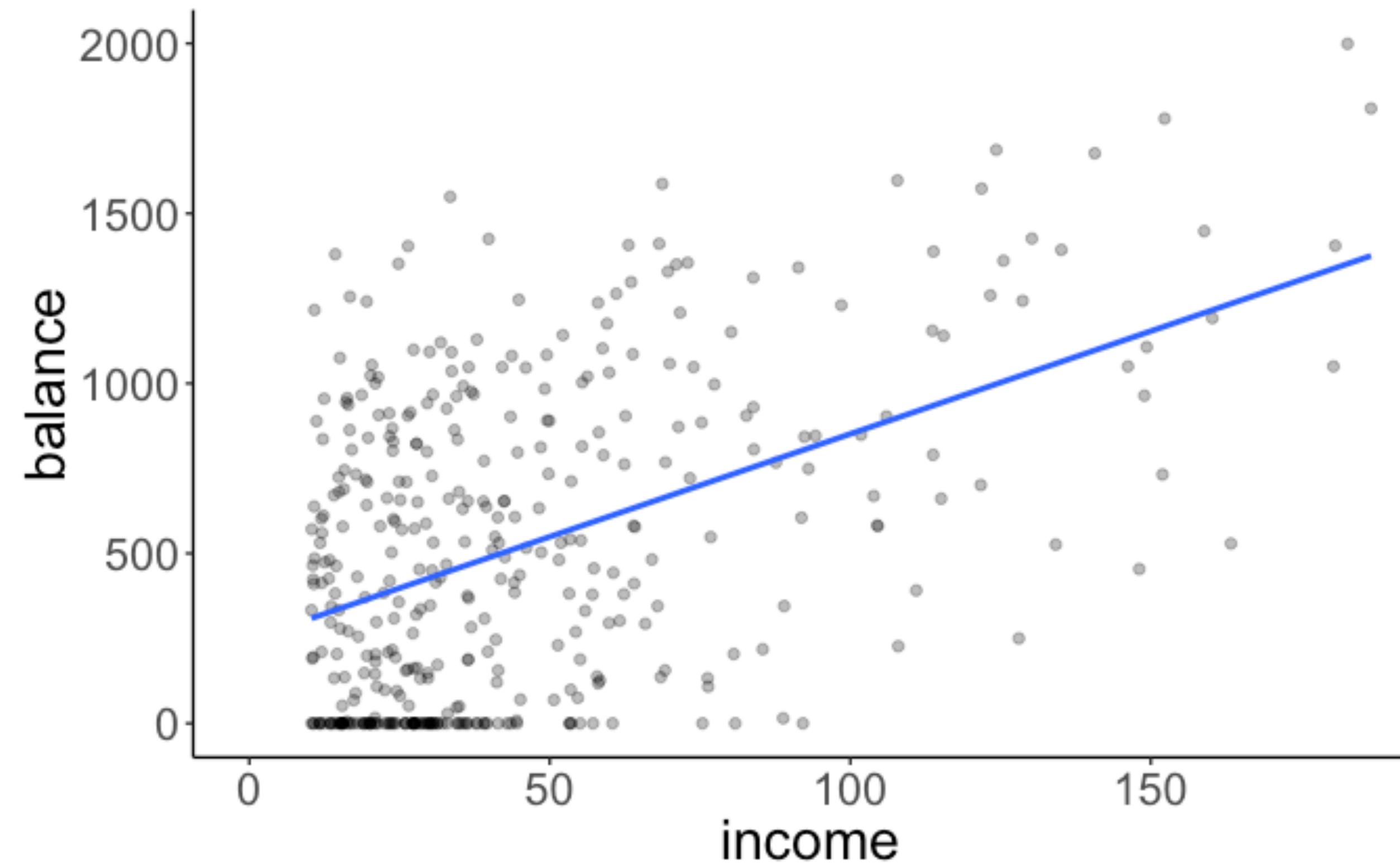
# Always plot the data first ...

```
1 ggplot(data = df.credit,  
2         mapping = aes(x = income,  
3                             y = balance)) +  
4     geom_point(alpha = 0.3)
```



# Always plot the data first ...

```
1 ggplot(data = df.credit,  
2         mapping = aes(x = income,  
3                             y = balance)) +  
4 geom_point(alpha = 0.3) +  
5 geom_smooth(method = "lm", se = F)
```



# Linear model

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$\text{balance}_i = b_0 + b_1 \cdot \text{income}_i + e_i$$

```
fit = lm(formula = balance ~ 1 + income, data = df.credit)
```

**outcome**      **intercept**      **predictor**      **data**

(doesn't need to be  
specified explicitly,  
but please **do it**  
**anyhow!**)

# **lm ( )**

```
fit = lm(balance ~ 1 + income, data = df.credit)
```

```
print(fit)
```

```
Call:  
lm(formula = balance ~ 1 + income, data = df.credit)
```

```
Coefficients:
```

	income
(Intercept)	246.515
income	6.048

**parameter estimates** → which minimize the sum of squared errors between model and data

# Interpreting regression parameters

Coefficients:

(Intercept)

246.515

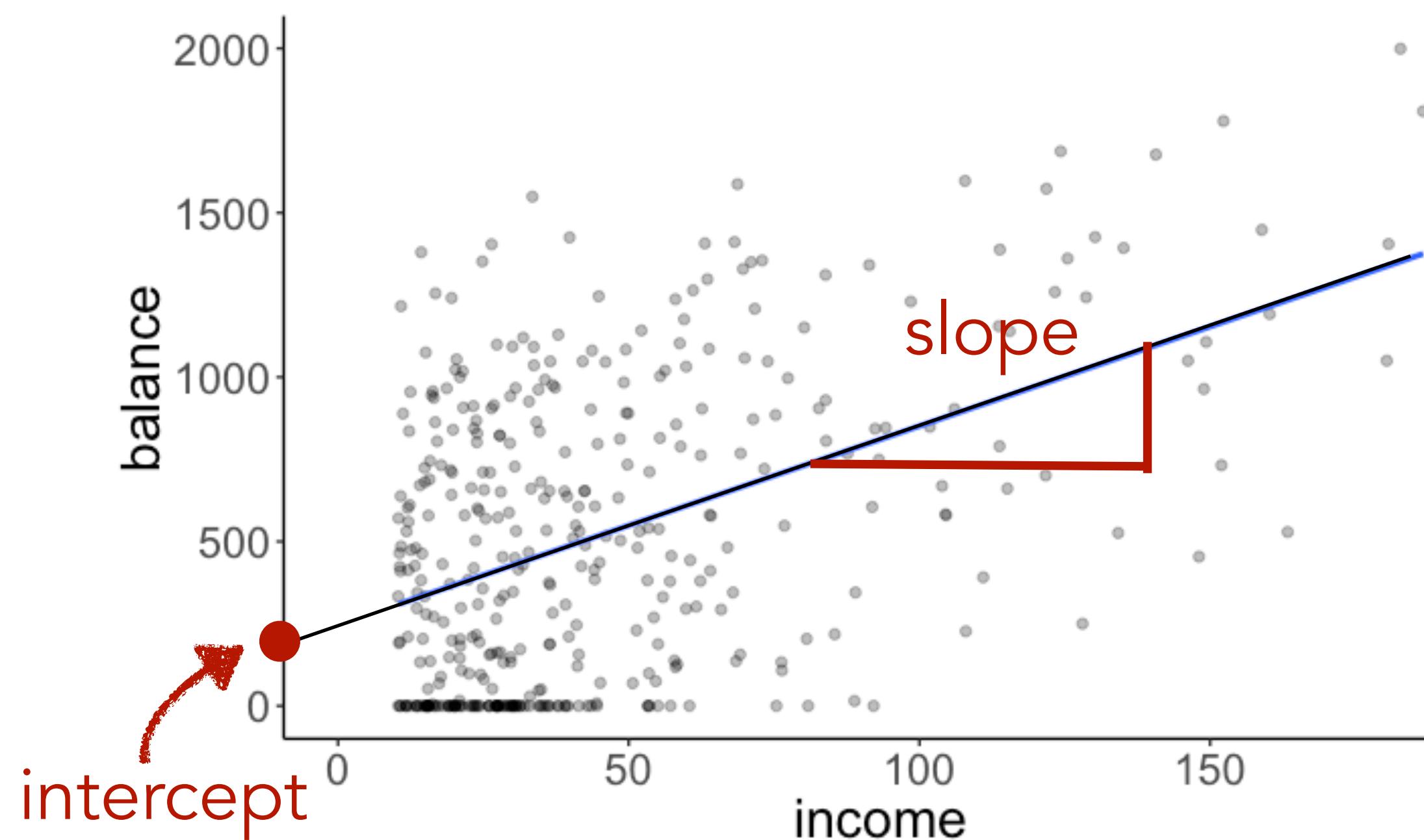
income

6.048

variable	description
income	in thousand dollars
balance	average credit card debt in dollars

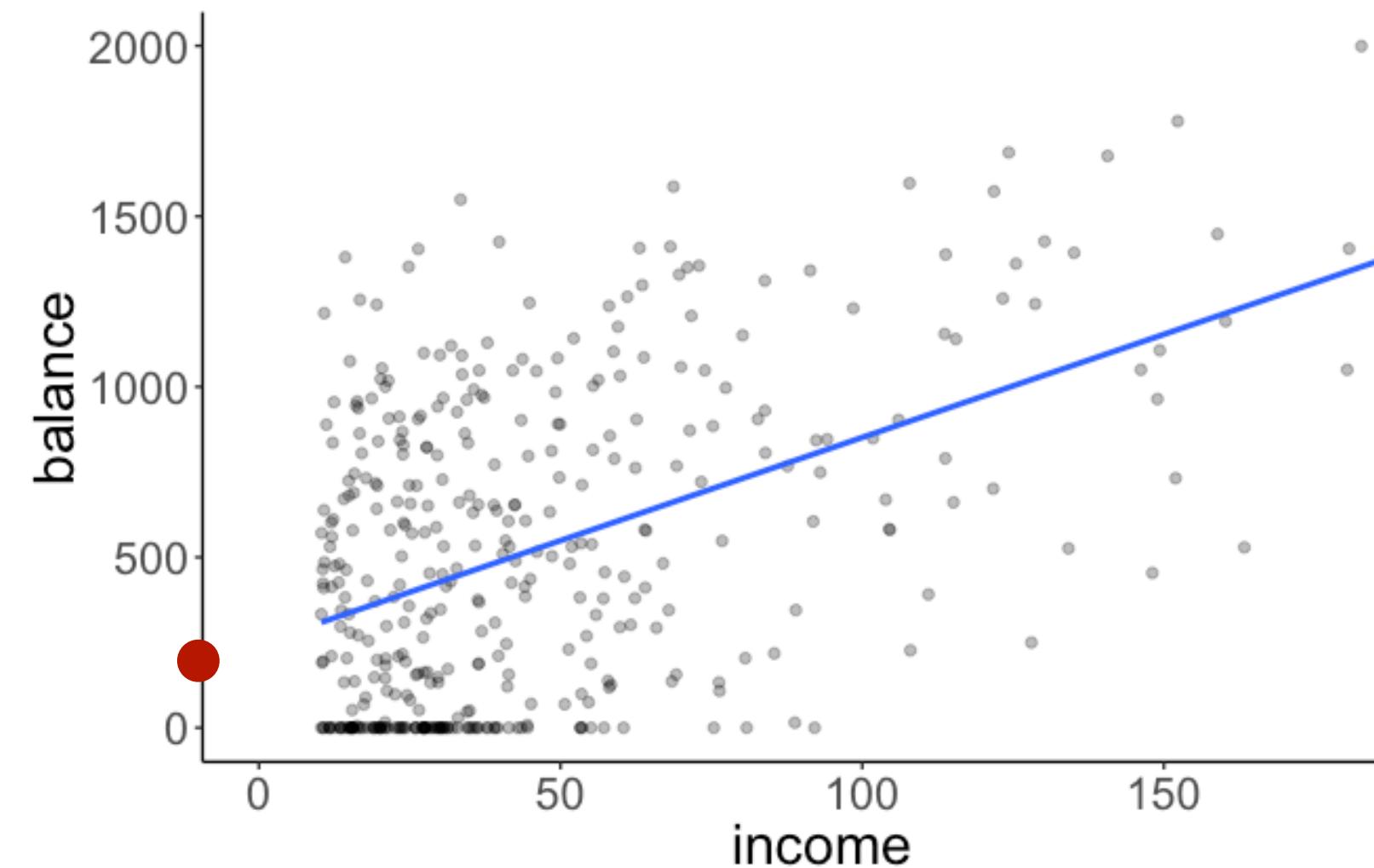
$$\text{balance}_i = b_0 + b_1 \cdot \text{income}_i + e_i$$

$$\text{balance}_i = 246.515 \cdot 1 + 6.048 \cdot \text{income}_i + e_i$$

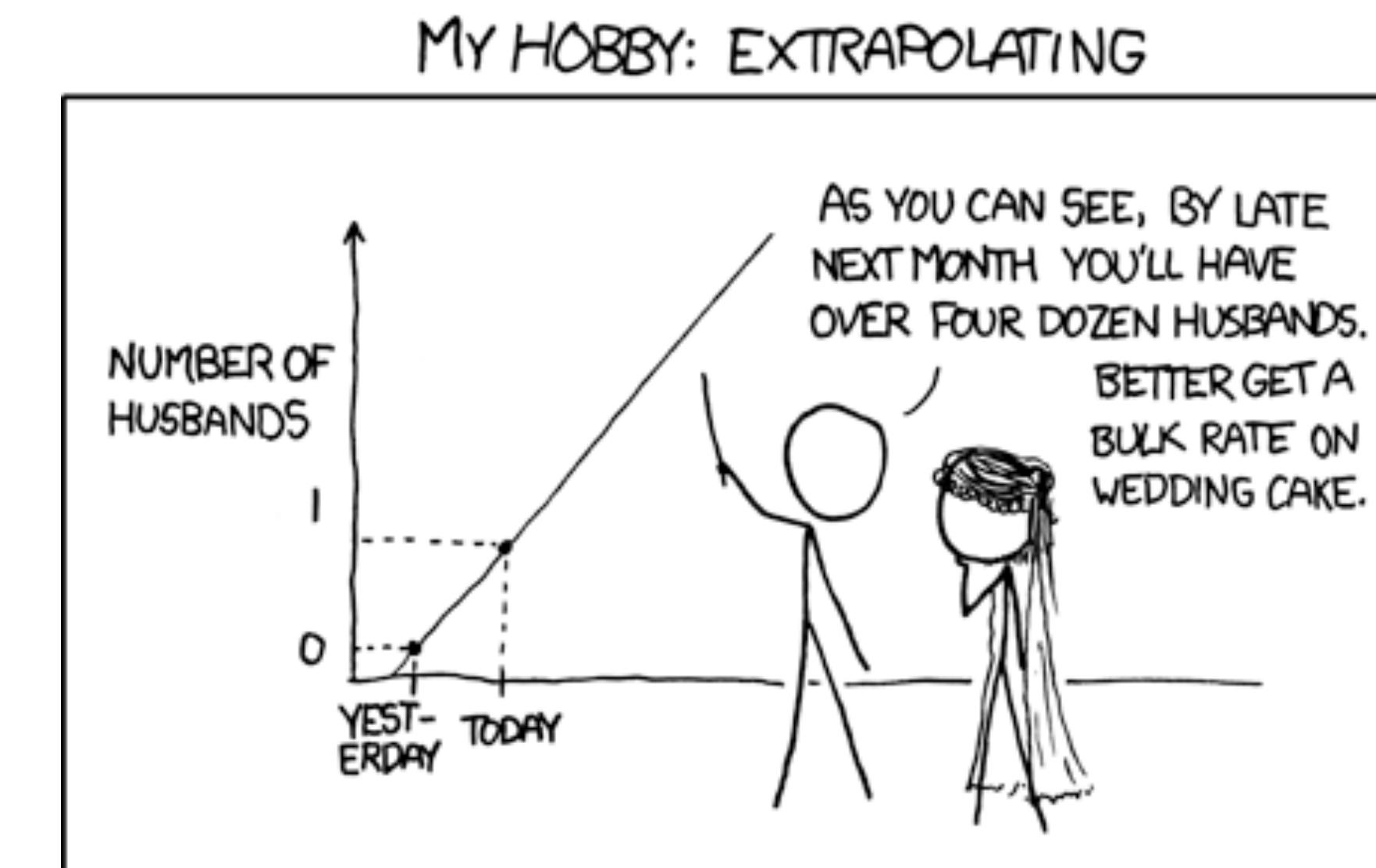


For each additional thousand dollars income, a person's average credit card balance is predicted to increase by \$6.05.

# Be careful about extrapolating predictions



- intercept is often outside the range of predictor values
- sometimes doesn't make sense (e.g. age = 0, height = 0, ...)



# summary()

```
1 lm(balance ~ 1 + income, data = df.credit) %>%
2   summary()
```

```
Call:
lm(formula = balance ~ 1 + income, data = df.credit)

Residuals:
    Min      1Q  Median      3Q     Max 
-803.64 -348.99 -54.42  331.75 1100.25 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 246.5148   33.1993   7.425  6.9e-13 *** 
income       6.0484    0.5794  10.440 < 2e-16 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 407.9 on 398 degrees of freedom
Multiple R-squared:  0.215, Adjusted R-squared:  0.213 
F-statistic: 109 on 1 and 398 DF,  p-value: < 2.2e-16
```



```
library("broom")
```

helps with tidying up  
model objects in R

**augment()** adds columns to the original data such as predictions, residuals and cluster assignments

**tidy()** summarizes a model's statistical findings such as coefficients of a regression

**glance()** provides a one-row summary of model-level statistics

broom: turn messy model outputs  
into tidy TIBBLES!



@allison\_horst

# augment()

fit %>%  
augment()

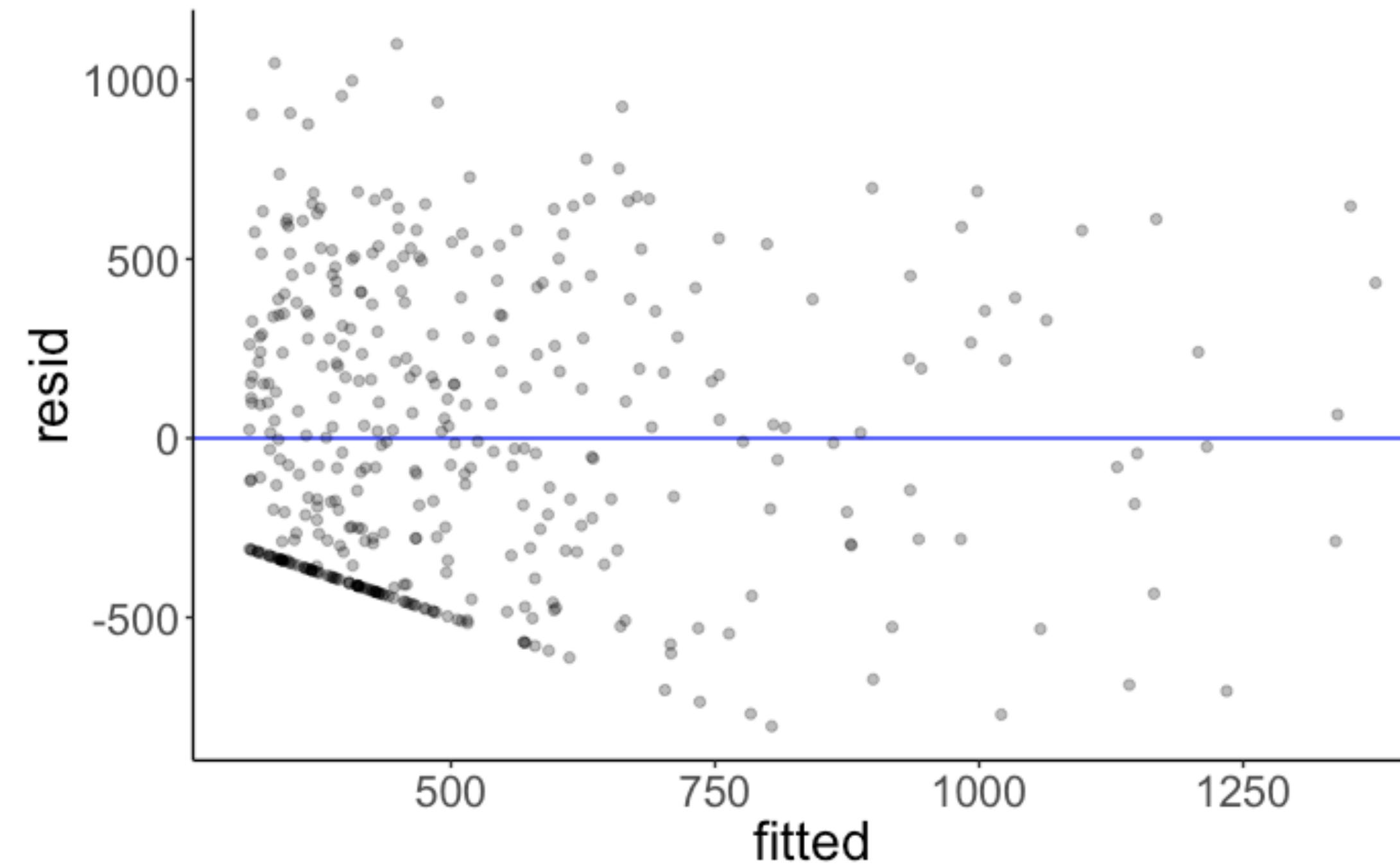
balance	income	.fitted	.se.fit	.resid	.hat	.sigma	.cooksdi	.std.resid
333	14.89	336.58	26.92	-3.58	0.00	408.38	0.00	-0.01
903	106.03	887.79	40.71	15.21	0.01	408.38	0.00	0.04
580	104.59	879.13	39.99	-299.13	0.01	408.10	0.00	-0.74
964	148.92	1147.26	63.45	-183.26	0.02	408.27	0.00	-0.45
331	55.88	584.51	21.31	-253.51	0.00	408.18	0.00	-0.62
1151	80.18	731.47	28.74	419.53	0.00	407.83	0.00	1.03
203	21.00	373.51	24.76	-170.51	0.00	408.29	0.00	-0.42
872	71.41	678.42	25.42	193.58	0.00	408.26	0.00	0.48
279	15.12	338.00	26.83	-59.00	0.00	408.37	0.00	-0.14
1350	71.06	676.32	25.30	673.68	0.00	406.97	0.01	1.65

# augment()

```
fit %>%  
  augment()
```

balance	income	.fitted	.resid
333	14.89	336.58	-3.58
903	106.03	887.79	15.21
580	104.59	879.13	-299.13
964	148.92	1147.26	-183.26
331	55.88	584.51	-253.51
1151	80.18	731.47	419.53
203	21.00	373.51	-170.51
872	71.41	678.42	193.58
279	15.12	338.00	-59.00
1350	71.06	676.32	673.68

## Residual plot



# tidy()

```
1 lm(balance ~ 1 + income, data = df.credit) %>%
2   summary()
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 246.5148    33.1993   7.425  6.9e-13 ***
income       6.0484     0.5794  10.440 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
1 fit %>%
2   tidy(conf.int = TRUE)
```

a data frame, yay!

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	246.51	33.20	7.43	0	181.25	311.78
income	6.05	0.58	10.44	0	4.91	7.19

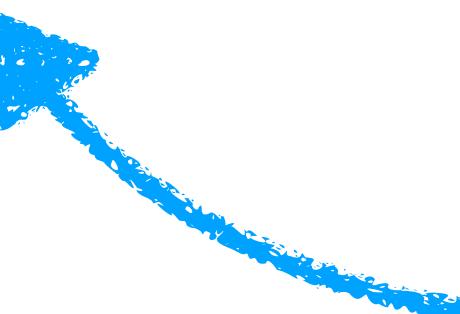
# glance()

```
1 lm(balance ~ 1 + income, data = df.credit) %>%  
2   summary()
```

```
Residual standard error: 407.9 on 398 degrees of freedom  
Multiple R-squared:  0.215, Adjusted R-squared:  0.213  
F-statistic: 109 on 1 and 398 DF, p-value: < 2.2e-16
```

```
1 fit %>%  
2   glance()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
0.21	0.21	407.86	108.99	0	2	-2970.95	5947.89	5959.87	66208745	398



useful model summary  
(we will learn later what  
the different values mean)

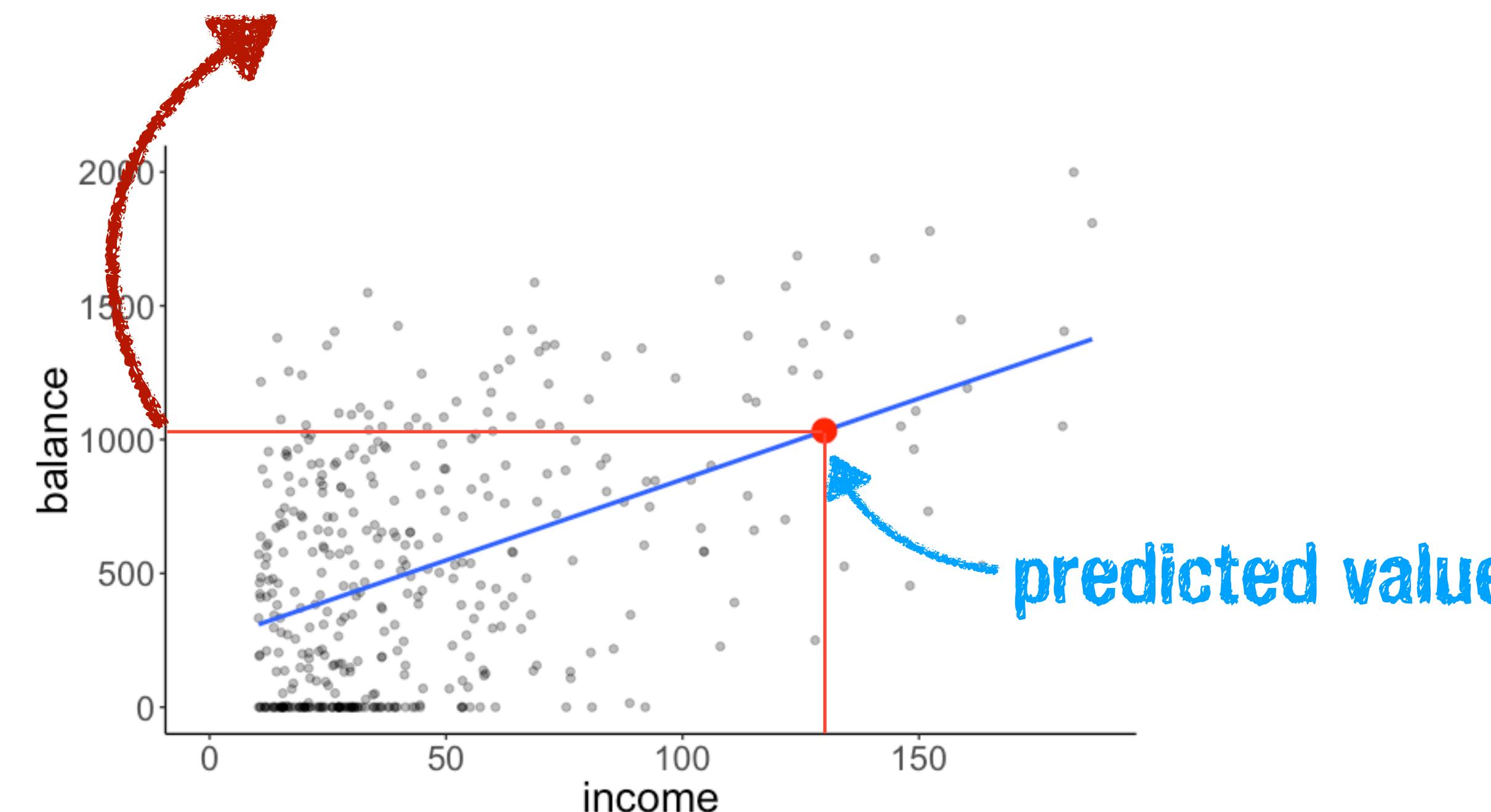
# Making predictions

```
fit = lm(balance ~ 1 + income, data = df.credit)
```

$$\text{balance}_i = 246.515 + 6.048 \cdot \text{income}_i + e_i$$

```
augment(fit, newdata = tibble(income = 130))
```

$$\widehat{\text{balance}} = 246.515 + 6.048 \cdot 130$$



# Hypothesis test

## Compact Model

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

```
fit_c = lm(formula = balance ~ 1,  
           data = df.credit)
```

## Augmented Model

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

```
fit_a = lm(formula = balance ~ 1 + income,  
           data = df.credit)
```

**anova (fit\_c, fit\_a)**

Analysis of Variance Table

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	399	84339912				
2	398	66208745	1	18131167	108.99 < 2.2e-16 ***	

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# 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)

```
fit_c = lm(formula = balance ~ 1,  
           data = df.credit)
```

```
fit_a = lm(formula = balance ~ 1 + income,  
           data = df.credit)
```

```
anova(fit_c, fit_a)
```

# Hypothesis test

**anova (fit\_c, fit\_a)**

Analysis of Variance Table

Model	balance ~ 1	balance ~ 1 + income
1	399	84339912
2	398	66208745
---		

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	84339912	1	18131167	108.99	< 2.2e-16 ***
---					

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

$$PRE = 1 - \frac{66208745}{84339912} \approx 0.215$$

The augmented model reduces the error by 21.5%.

```
lm(balance ~ 1 + income, data = df.credit) %>%  
  summary()
```

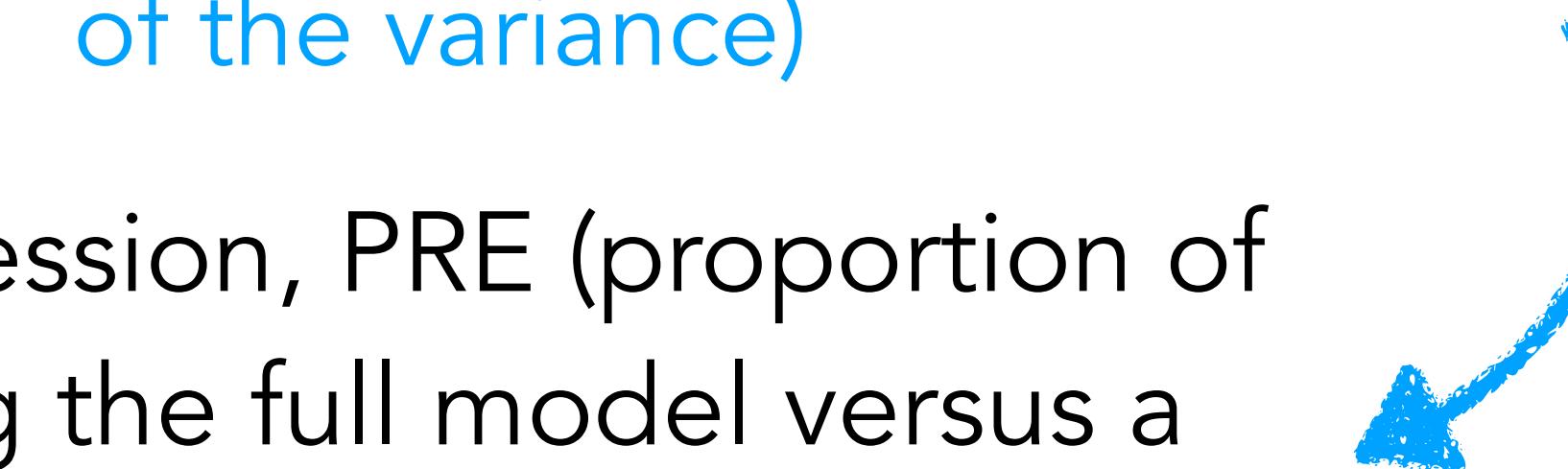
$R^2$

Residual standard error:	407.9	on 398 degrees of freedom
Multiple R-squared:	0.215,	Adjusted R-squared: 0.213
F-statistic:	109	on 1 and 398 DF, p-value: < 2.2e-16

# Hypothesis test

the **compact model** predicts the mean (which doesn't explain any of the variance)

- in the case of a simple regression, PRE (proportion of reduced error) of comparing the full model versus a model with just an intercept is identical to  $R^2$  (variance explained)
- and  $R^2$  is directly related to the correlation coefficient  $r$



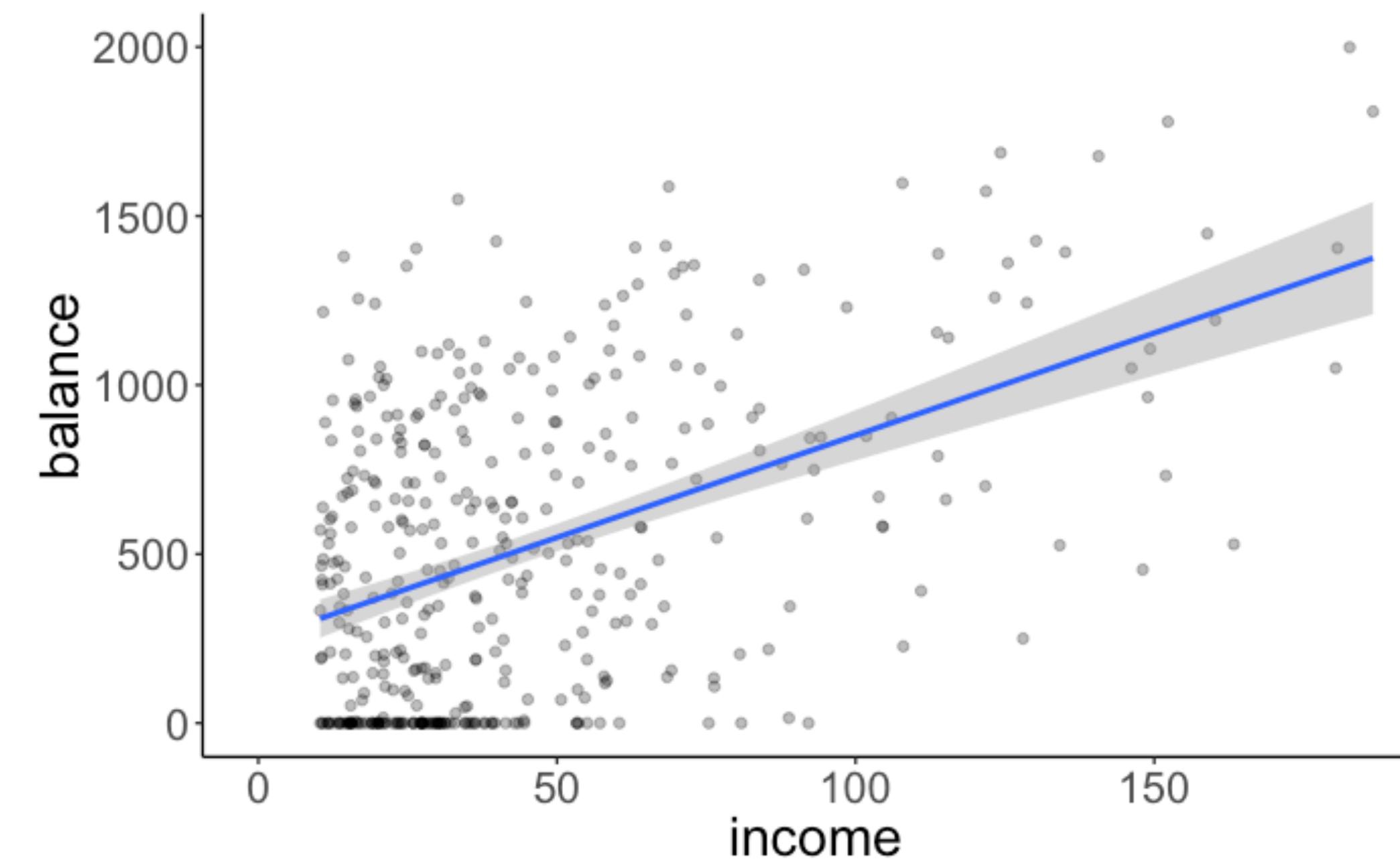
```
cor(df.credit$balance,  
df.credit$income)
```

$$R^2 = 0.215$$

$$r = .463$$

effect size measure

# Reporting the results

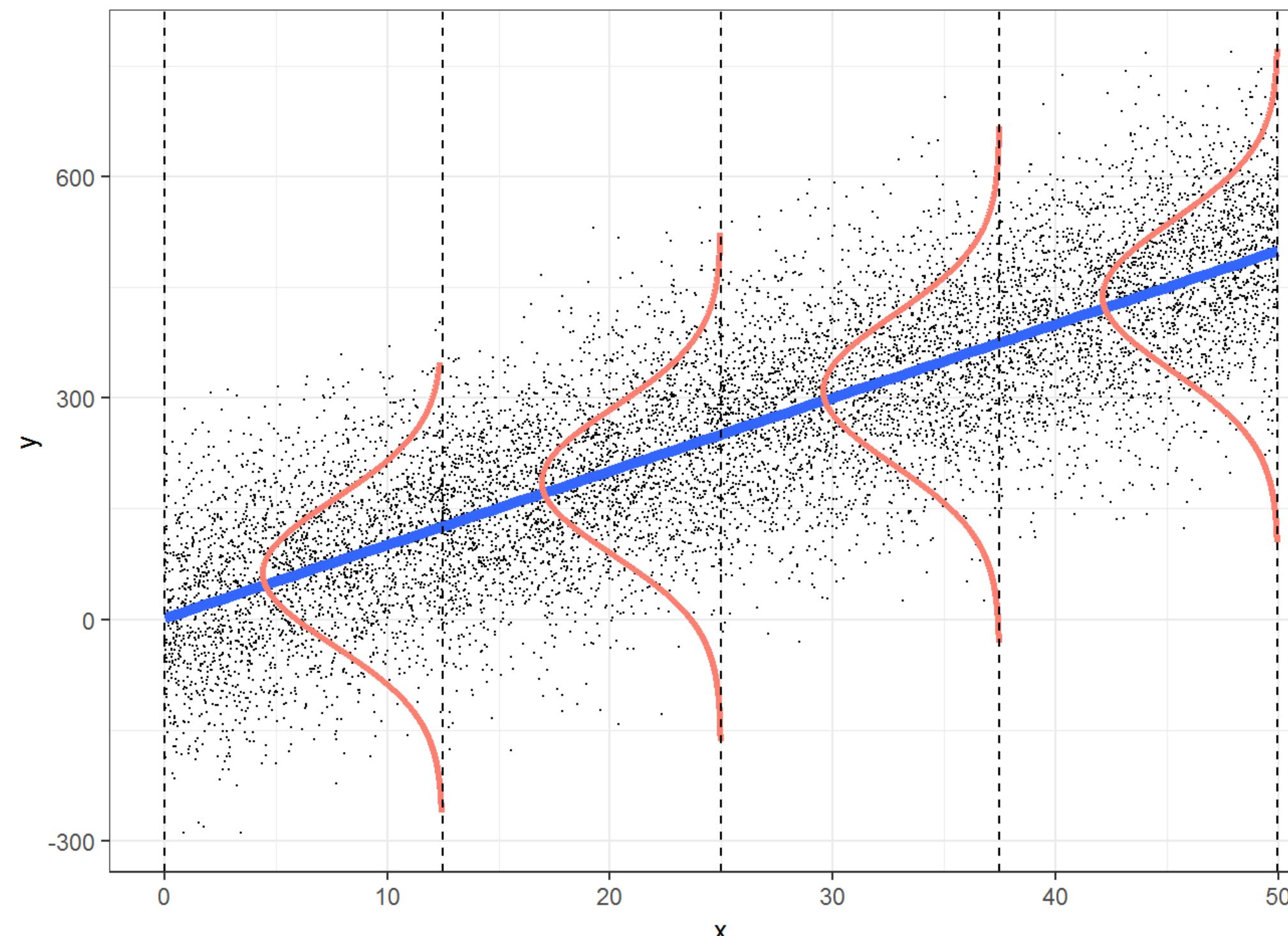


There is a significant relationship between a person's income and the average balance on their credit cards  $F(1, 389) = 108.99, p < .001, r = .463$ .

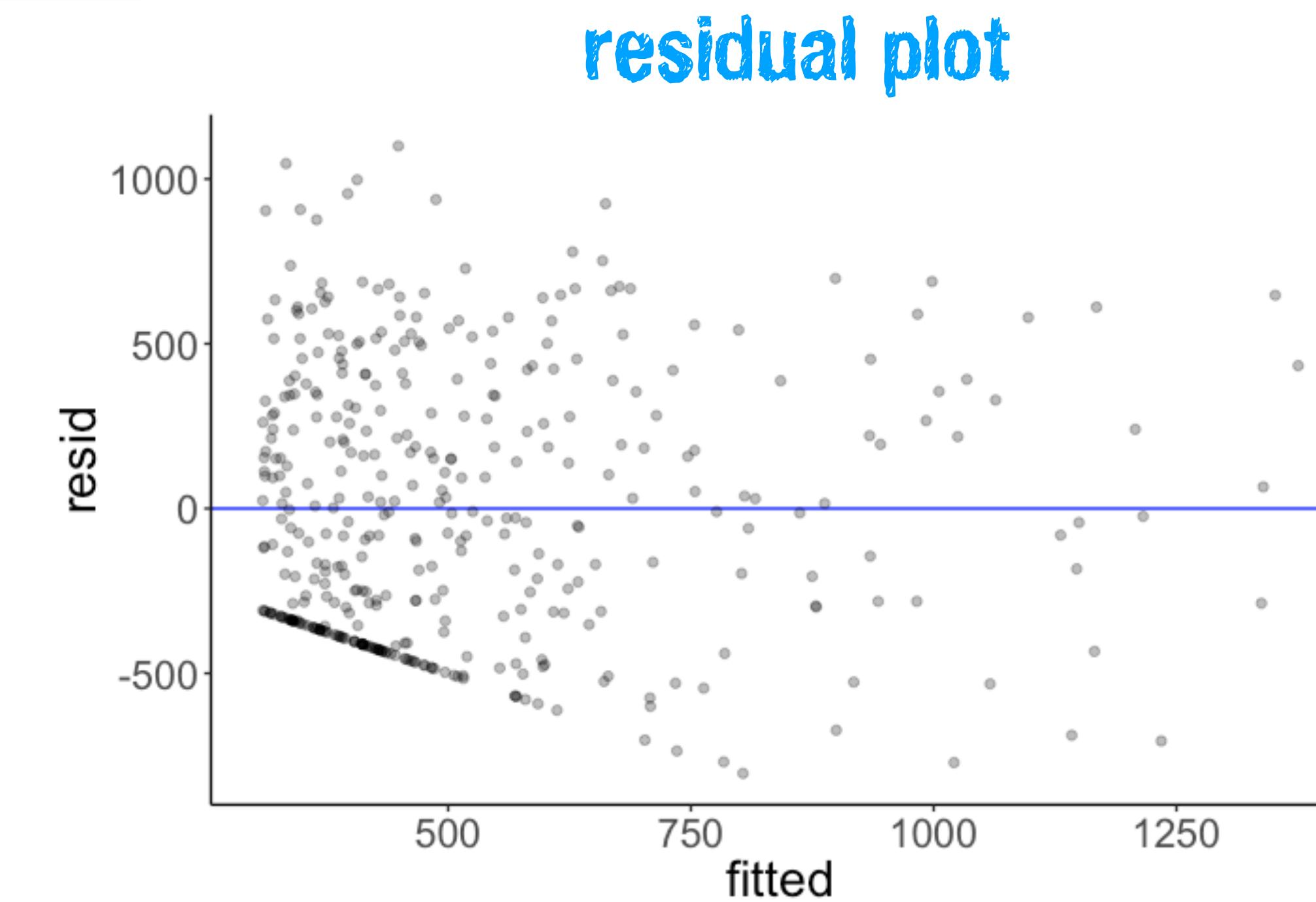
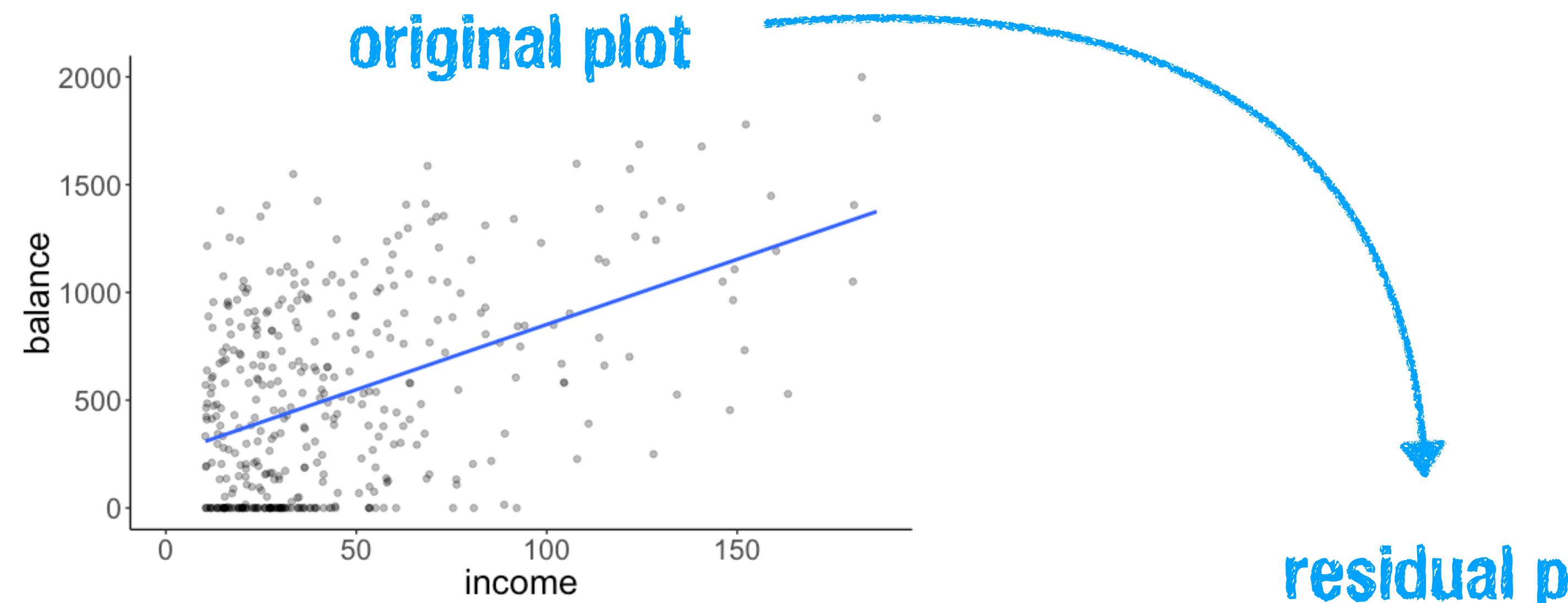
With each additional \$1000 of income, the average balance is predicted to increase by \$6.05 [4.91, 7.19] (95% CI).

# Model assumptions

- independent observations
- $Y$  is continuous
- errors are normally distributed
- errors have constant variance
- error terms are uncorrelated

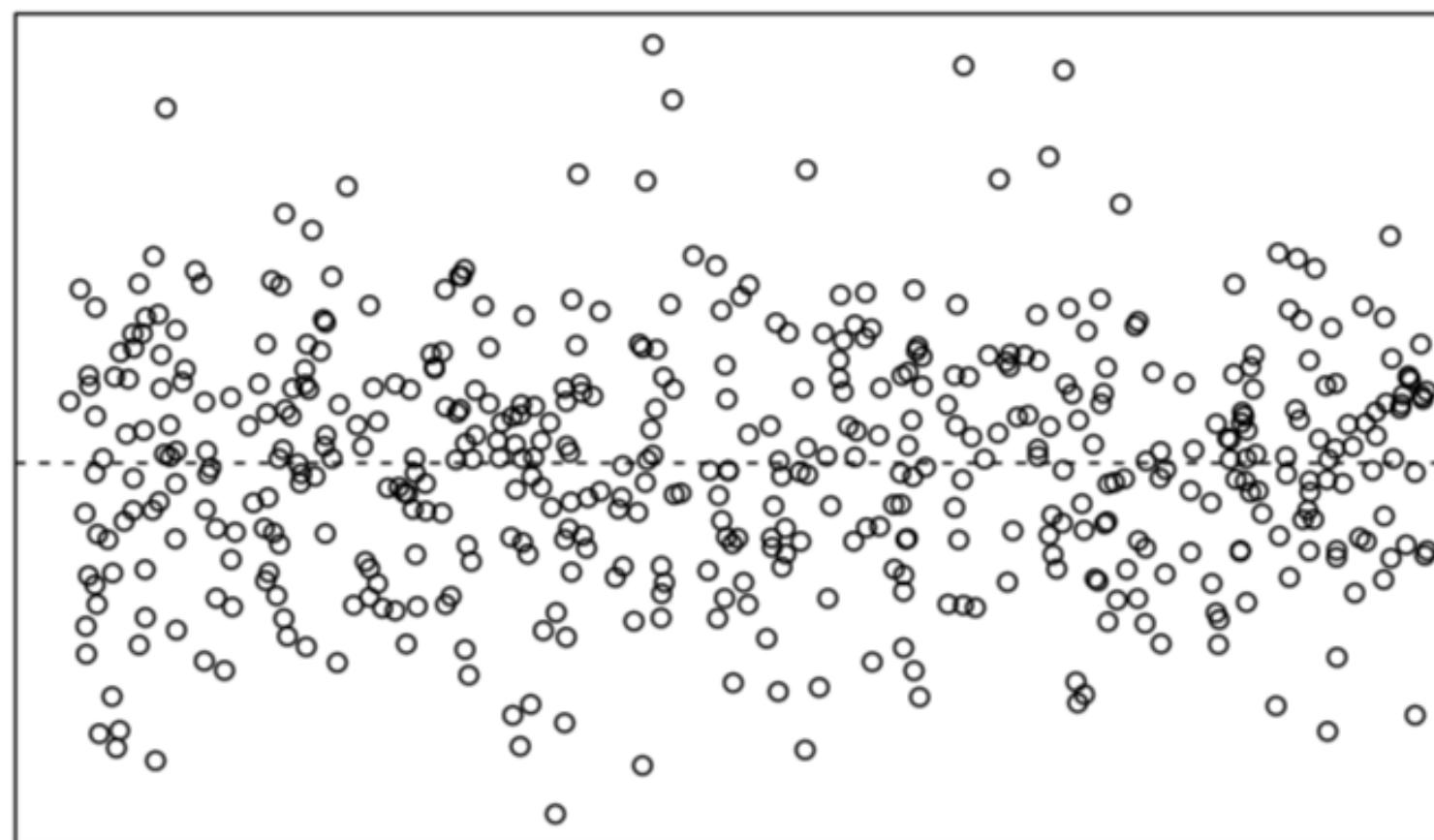


# Model assumptions

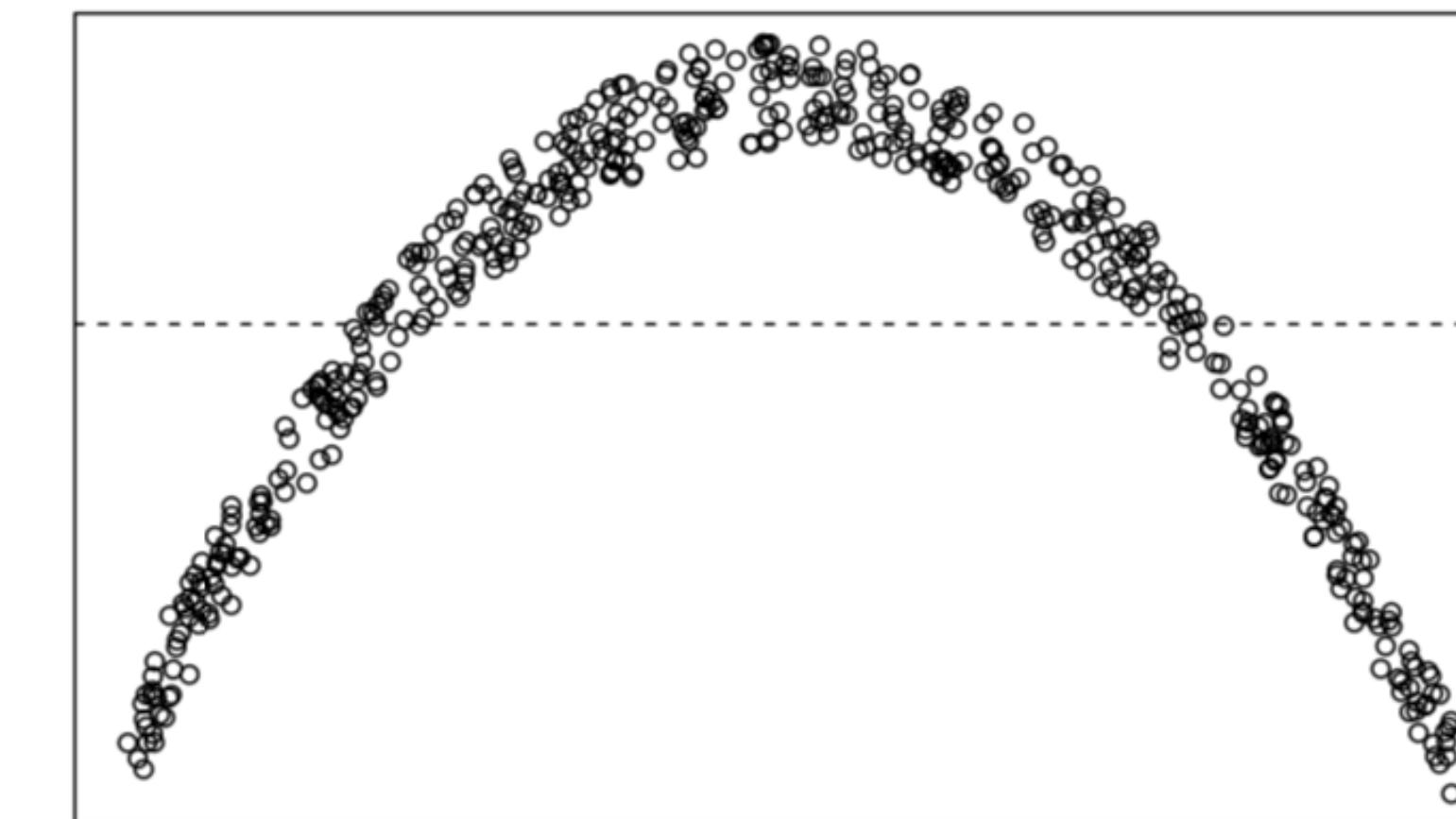


# Model assumptions

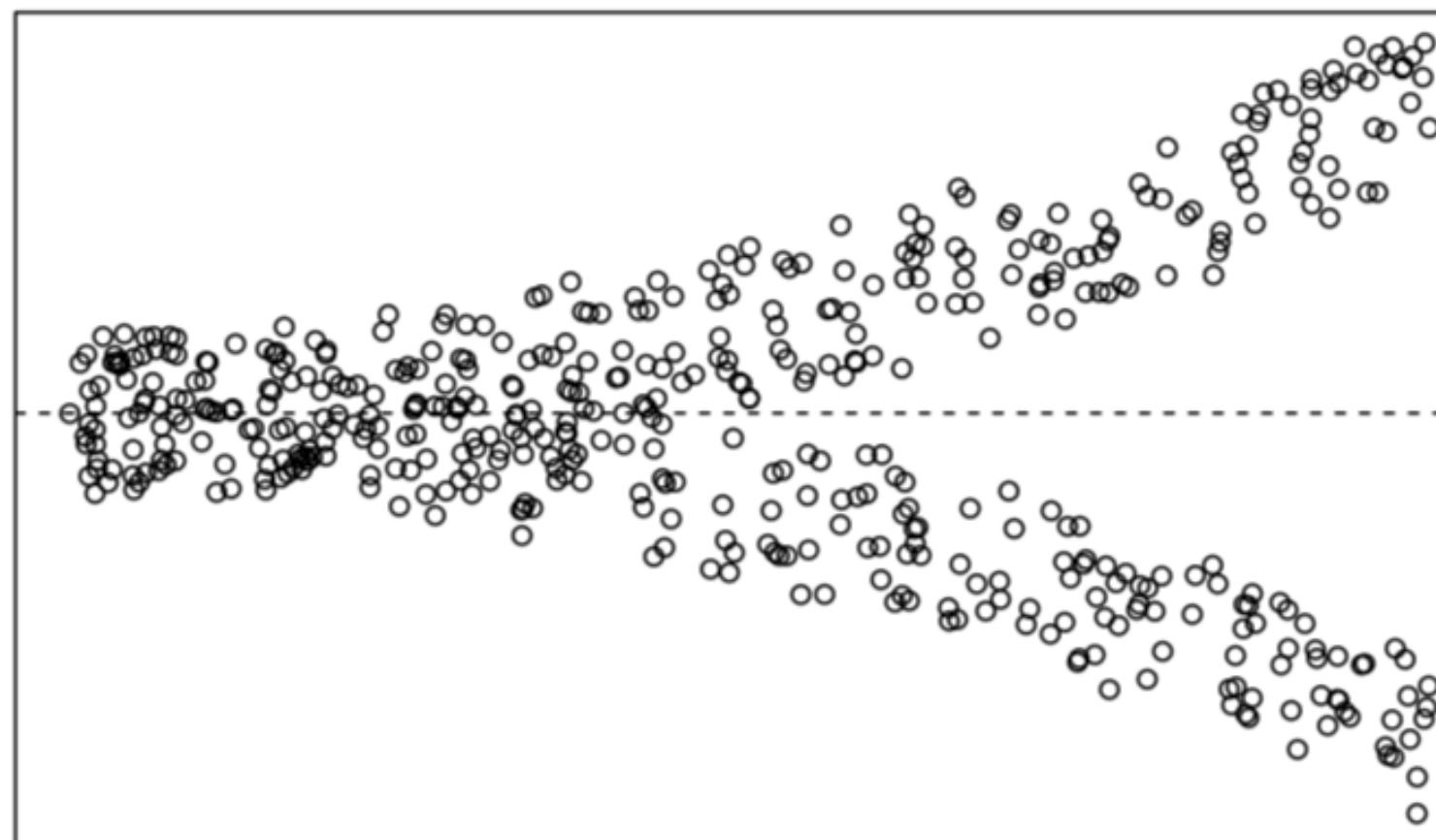
No Violation



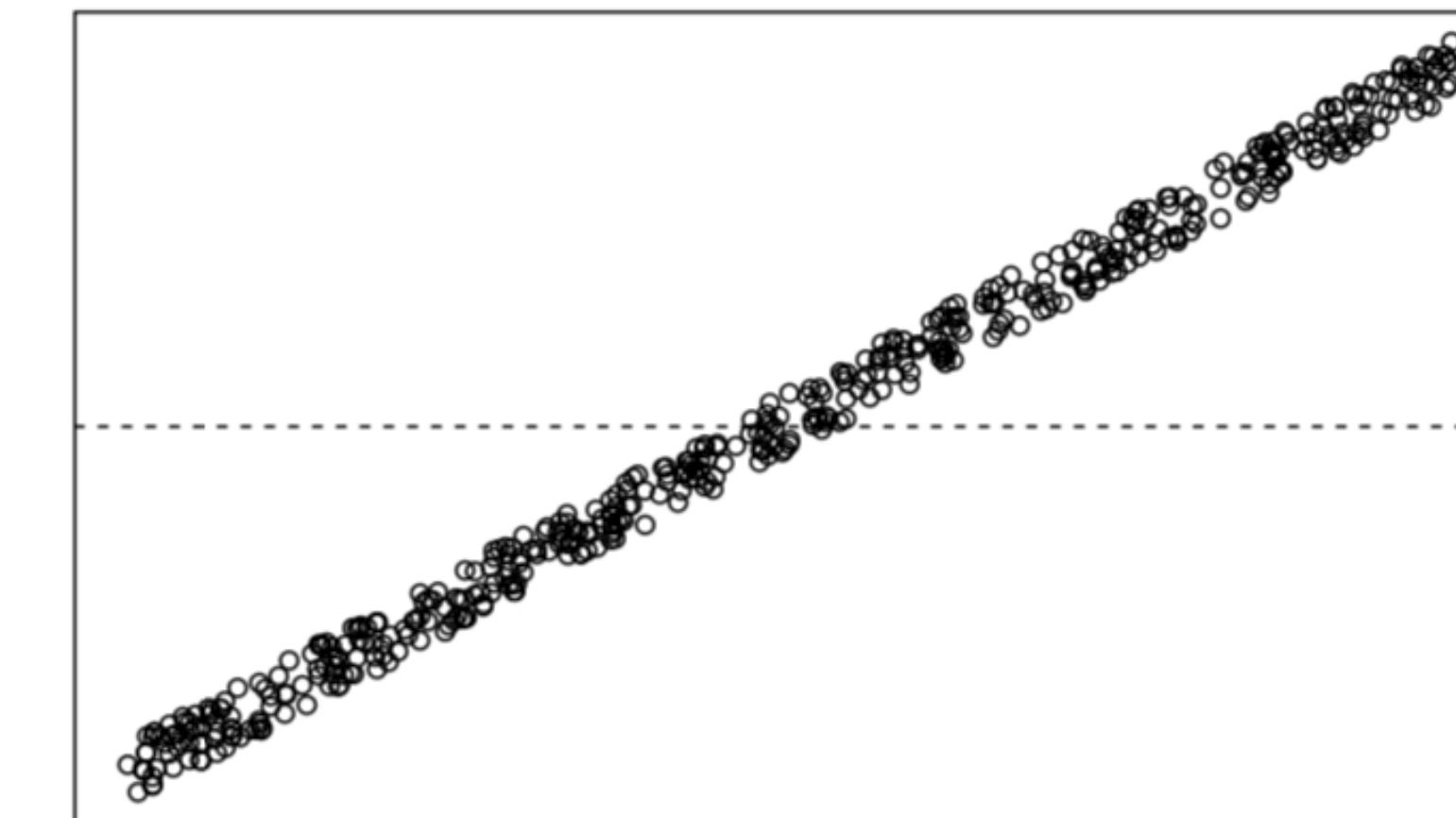
Nonlinear Relationship



Nonconstant Error Variance



Dependent Error Terms



# Summary

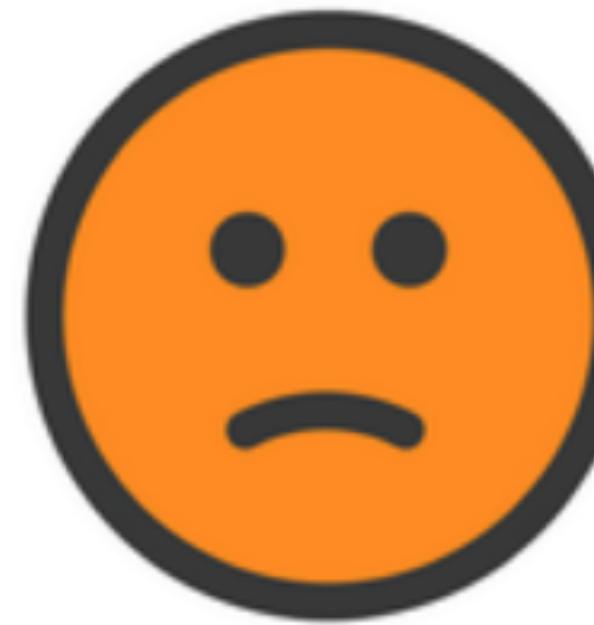
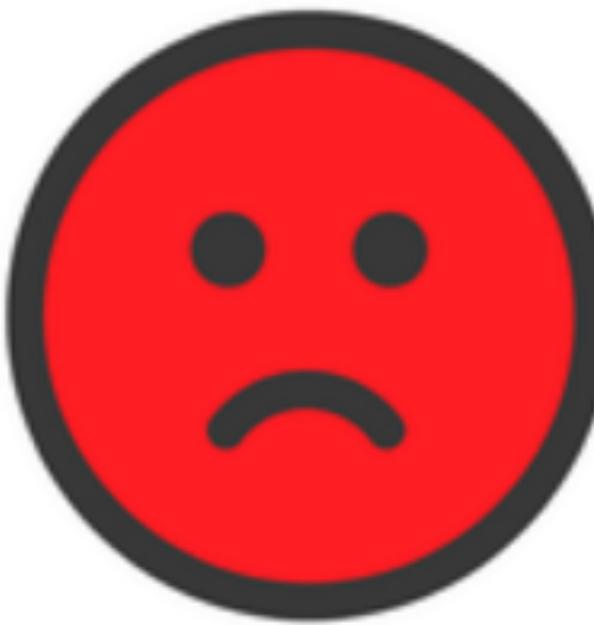
- Quick recap
- Generating a sampling distribution for PRE
- Correlation
- Regression
  - The conceptual tour
  - The R route

# **Feedback**

# How was the pace of today's class?

much      a little      just      a little      much  
too      too      right      too      too  
slow      slow           fast      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!**