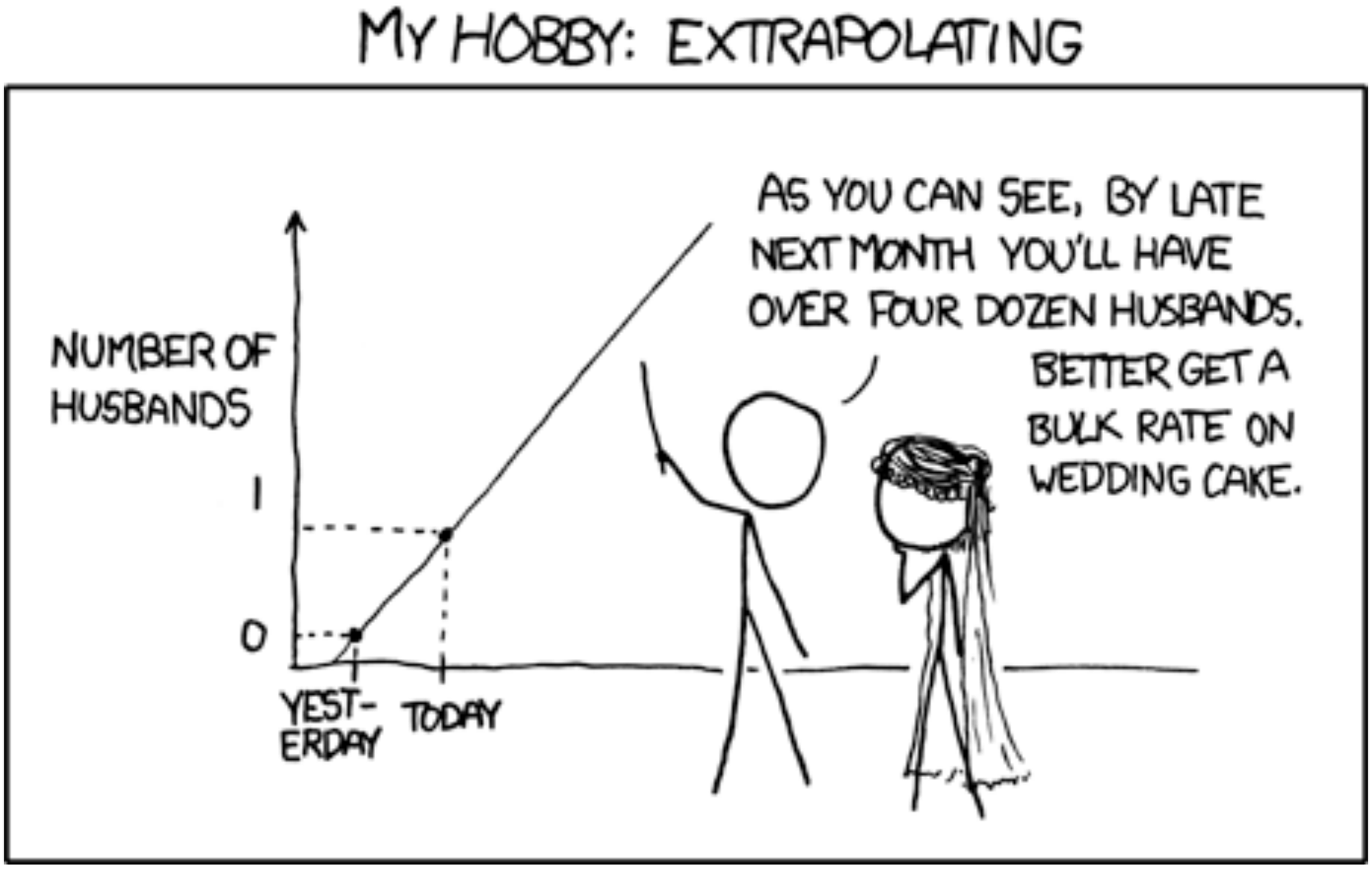
Linear model 1



01/29/2020

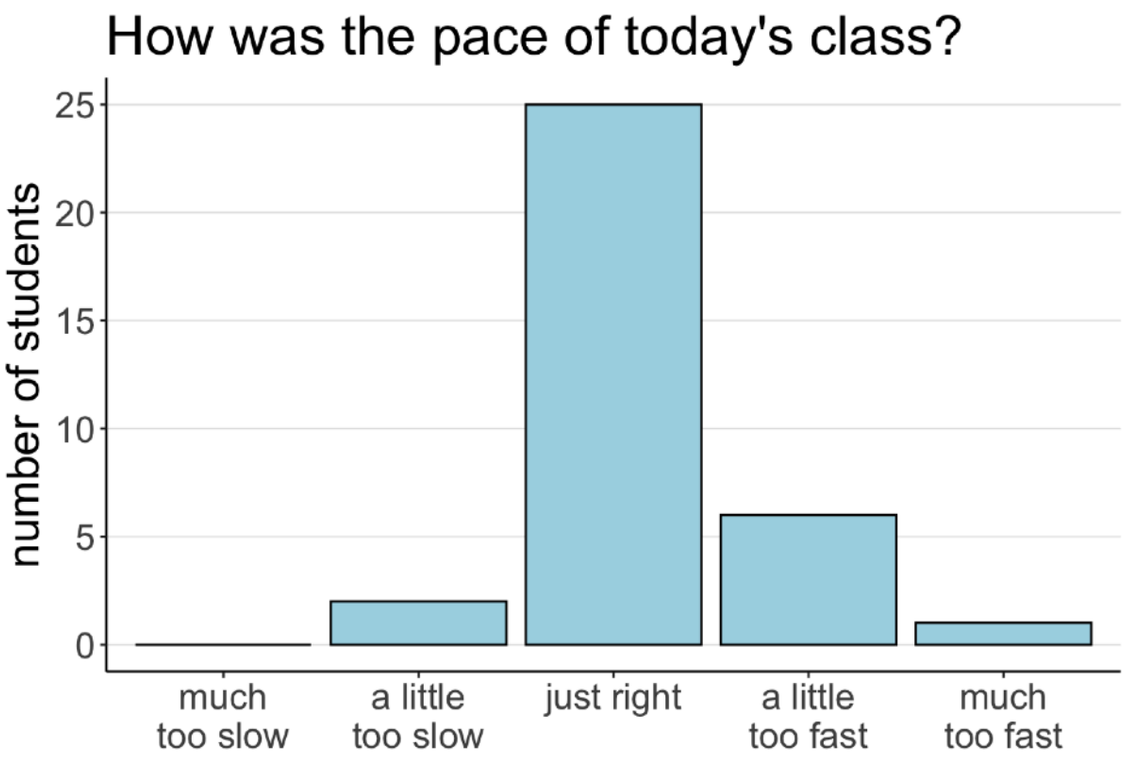
Logistics

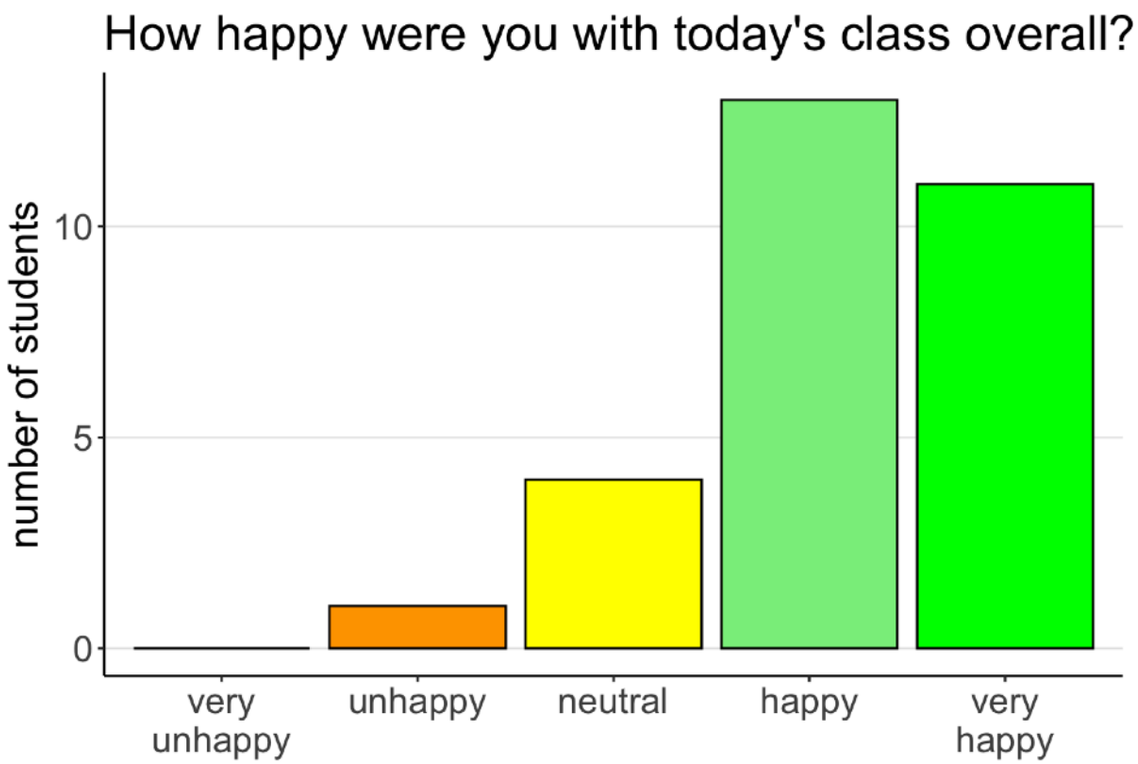
Your feedback

Homework 2


How many hours did you spend on homework 2?
number of students
0
2
4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
or more

Your feedback





Your feedback

I found the modeling-based approach very intuitive and I think it helped me understand the conceptual basis of statistical testing more so than other classes I've taken. Thanks.

I think the lecture is still too theoretical with not enough concrete examples….it makes it hard to follow.

you have survived the theory part, things will get more practical now

Your feedback

I didn’t fully understand the utility of the f statistic. I like the simple heuristic of modeling as a process of increasing accuracy while minimizing complexity, but I didn’t fully understand why it’s important to minimize complexity. A short explanation might fix things for me.

check out application section tomorrow!

Application section

Application section

Psych252 section 3: modeling data

Jinxiao Zhang

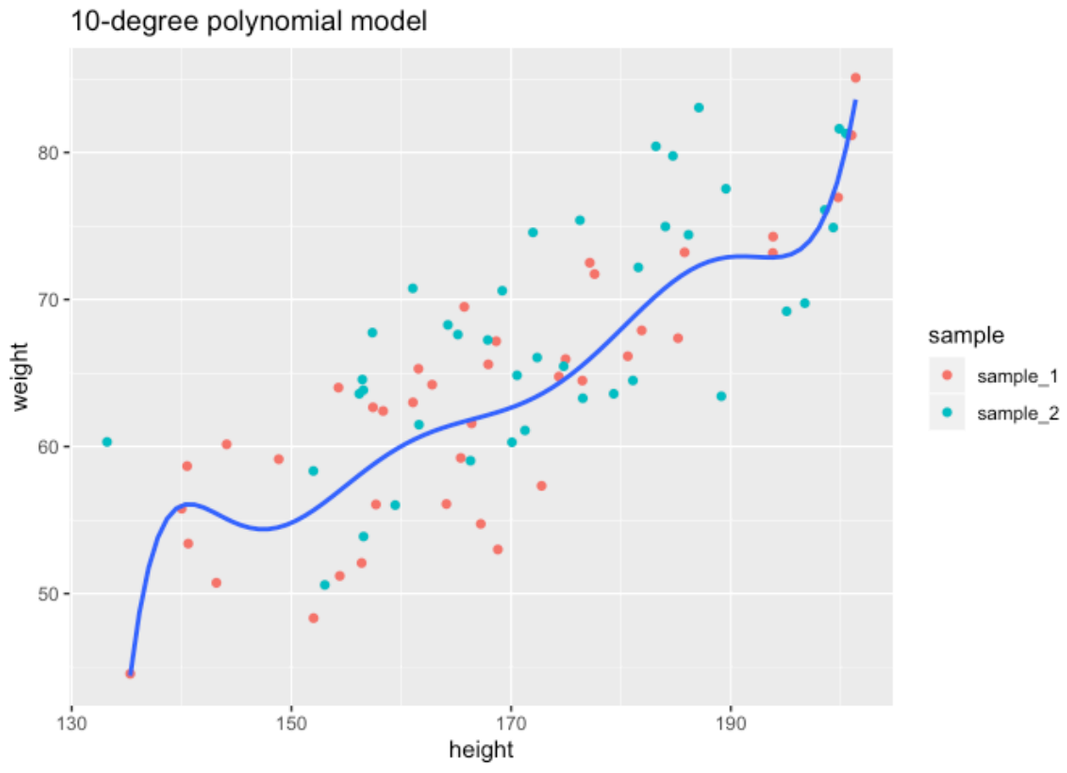
January 30, 2020

Load packages and set plotting theme

```
library("knitr")      # for knitting RMarkdown
library("tidyverse")  # for wrangling, plotting, etc.
```

What is overfitting?

In statistics, overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably" (wikipedia). Overfitting is briefly mentioned in the lecture. In this section, we are going to get a deeper understanding of this sentence together by experimenting fitting different models to the same data.



Plan for today

- Quick review of statistical inference in frequentist statistics
- Correlation
  - Pearson's moment correlation
  - Spearman's rank correlation
- Regression
  - The conceptual tour
  - The R route

Quick review of statistical inference in frequentist statistics

The general procedure

1. Start with research question
2. Formulate hypothesis as a comparison between compact and augmented model
3. Define H_0 as Model C (compact) and H_1 as Model A (augmented)
4. Fit model parameters to the data
5. Calculate the proportional reduction of error (PRE) in our sample
6. 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)

Research question and hypotheses

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

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

0 parameters

$Y_i = 75 + e_i$

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

1 parameter

$Y_i = b_0 + e_i$

$= \bar{Y} + e_i$

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

Decide whether it's **worth it**

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

Population distribution

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

Model C

$$Y_i = 75 + e_i$$

0 parameters

Model A

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

1 parameter

Sampling distribution of PRE

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

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

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

samples

1000 columns

50 rows

Sampling distribution of PRE

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

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

Sampling distribution of PRE

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

calculate SSE for each model

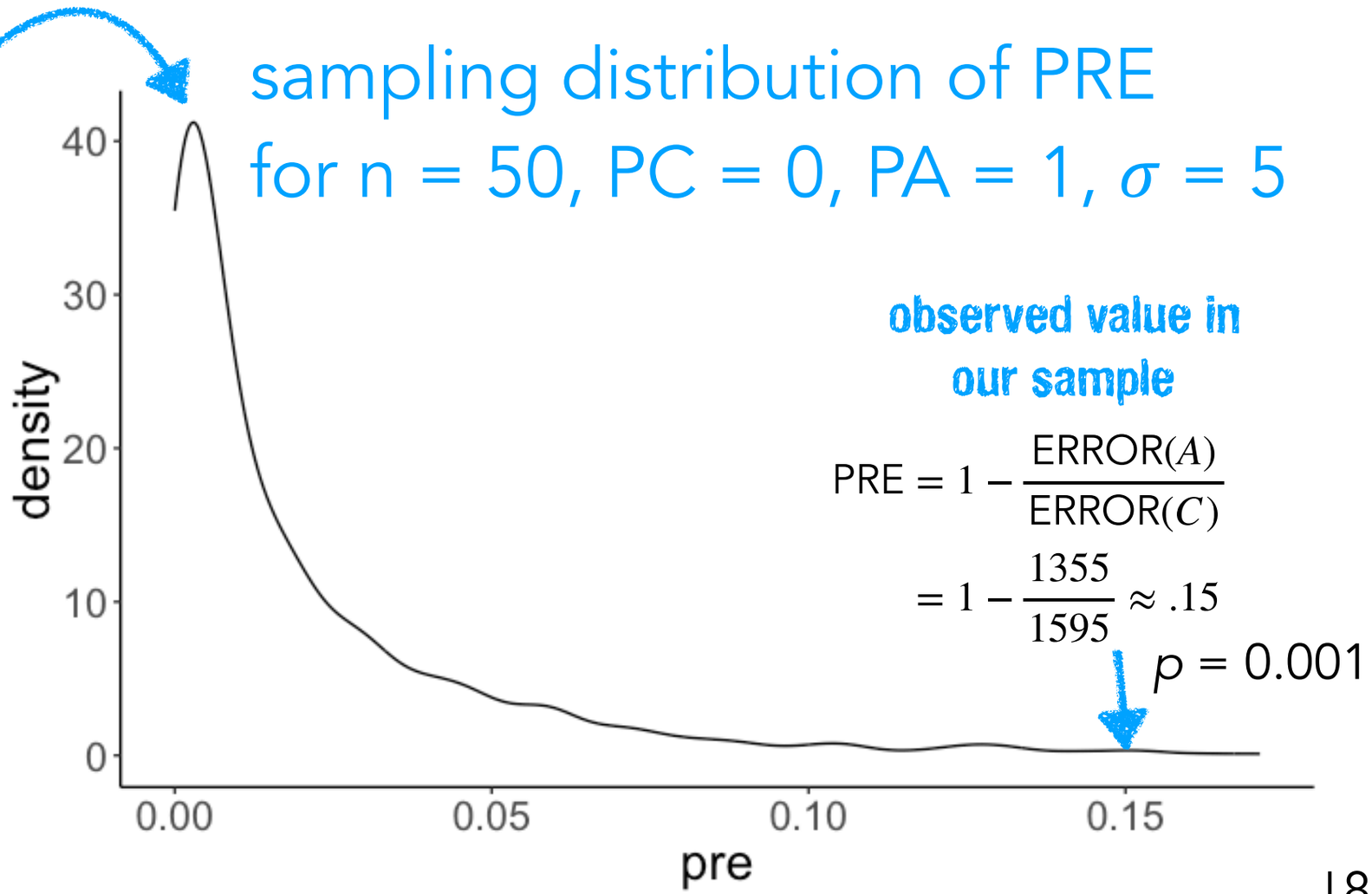
calculate PRE

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

Sampling distribution of PRE

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

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

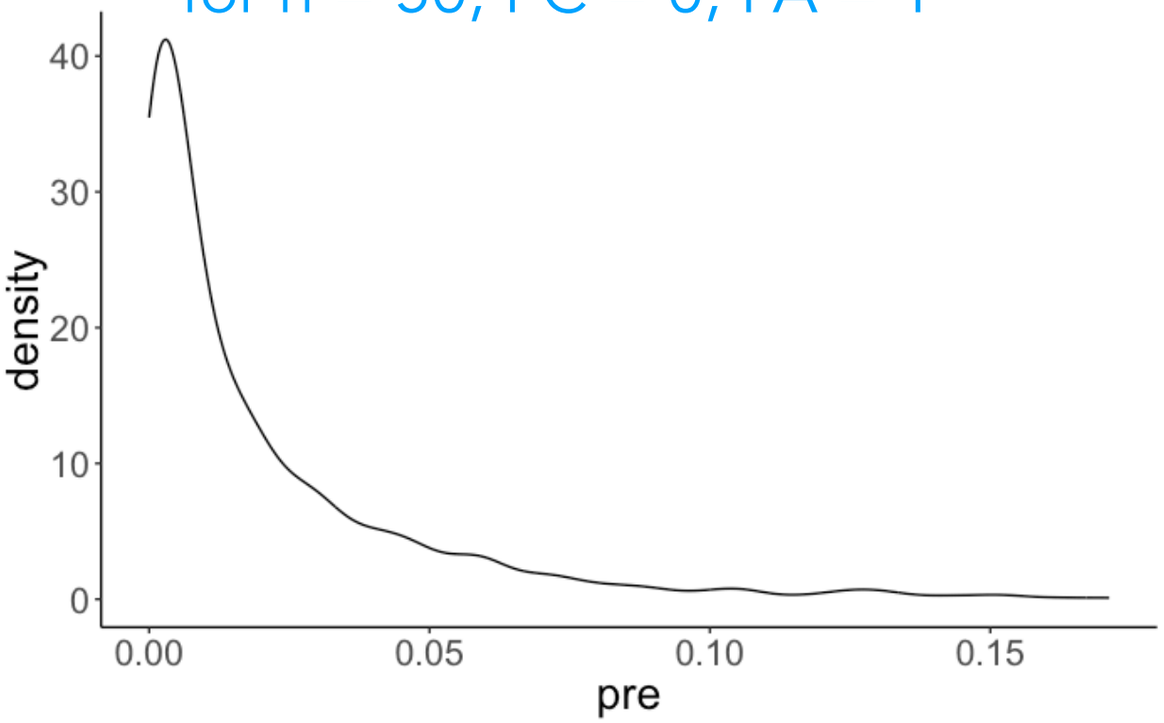


Sampling distribution of PRE

deterministic mapping

sampling distribution of PRE

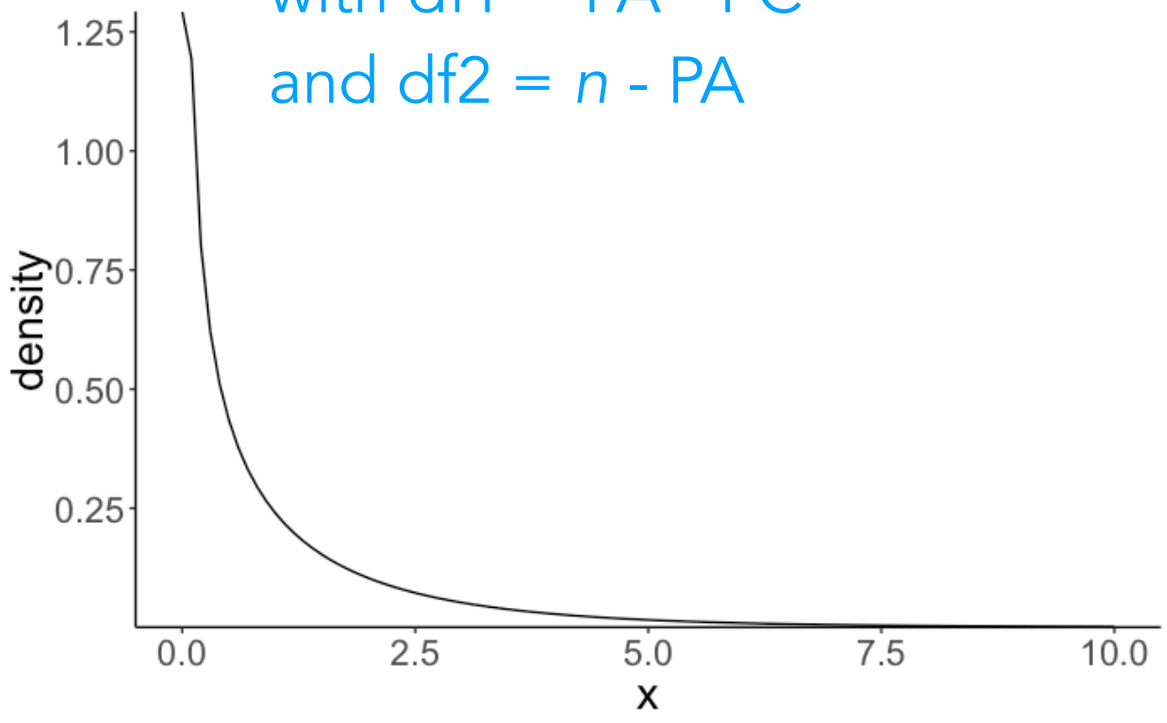
for n = 50, PC = 0, PA = 1



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

with df1 = PA - PC

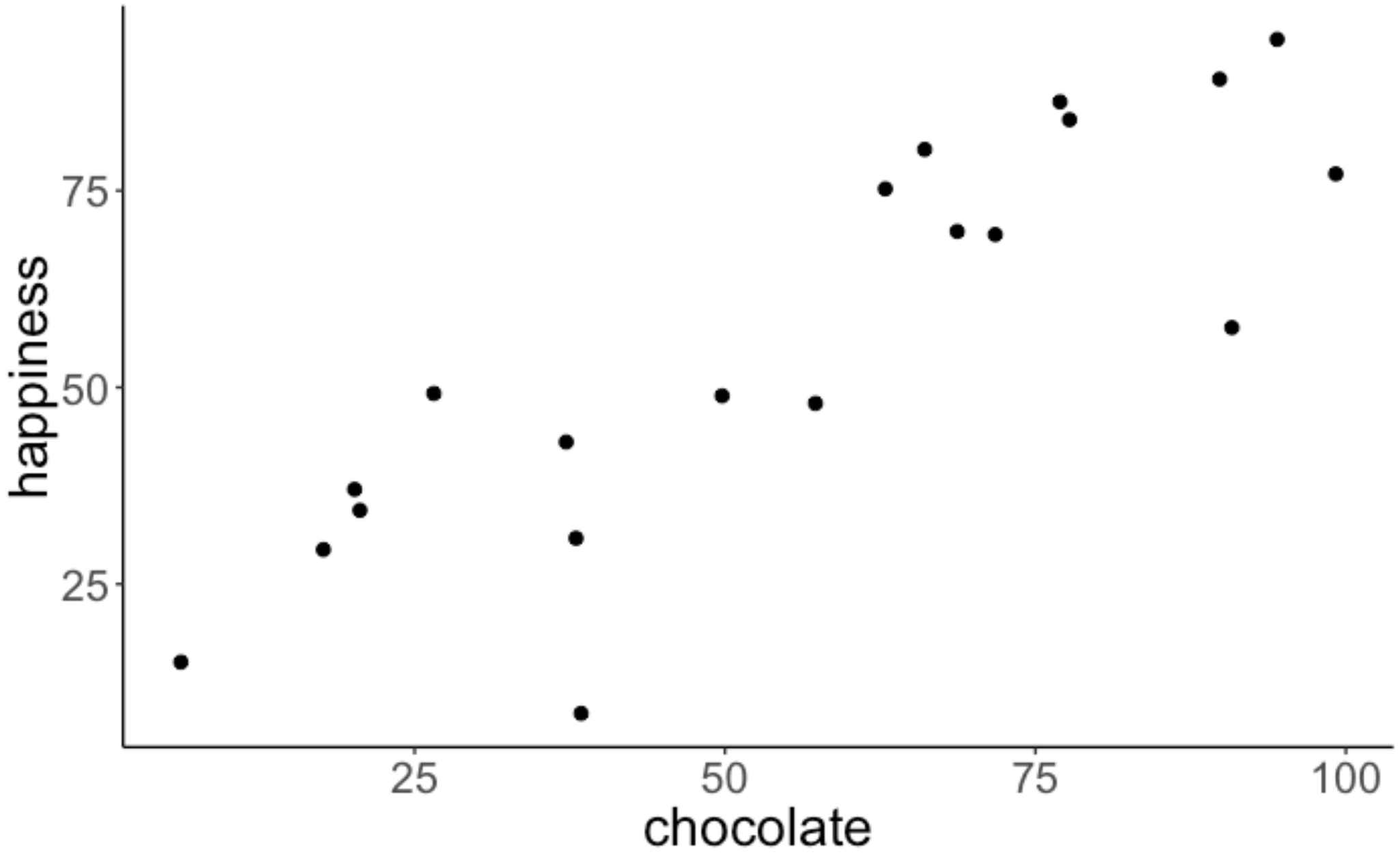
and df2 = *n* - PA



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

Correlation

How to best characterize the relationship between x and y by a single number?

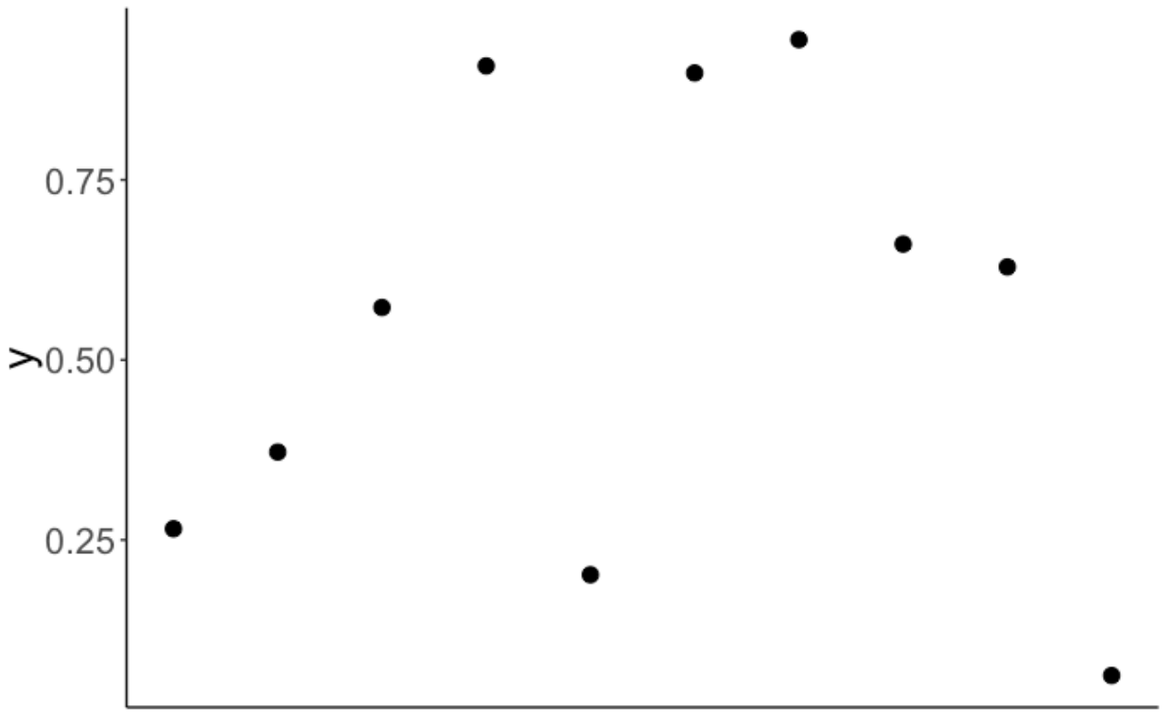


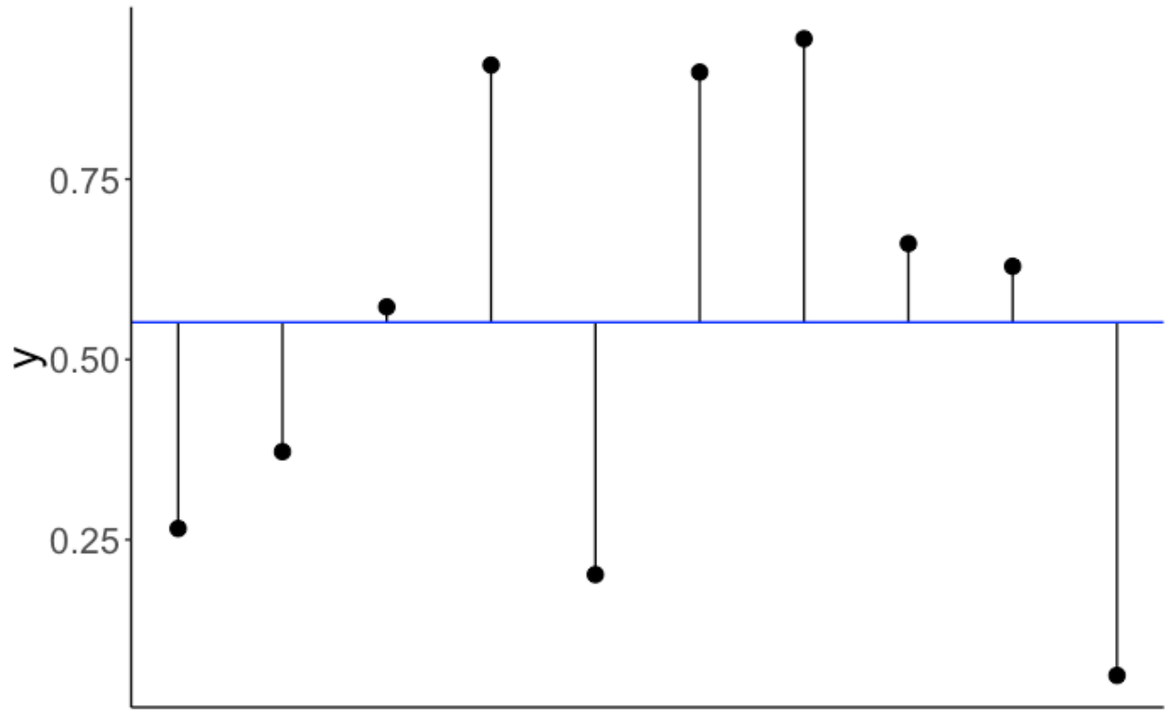
correlation = a measure of the relationship between two variables

sample variance

$$Var(Y) = \sum_{i=1}^{n} \frac{(Y_i - \overline{Y})^2}{n-1}$$

sum of squared errors



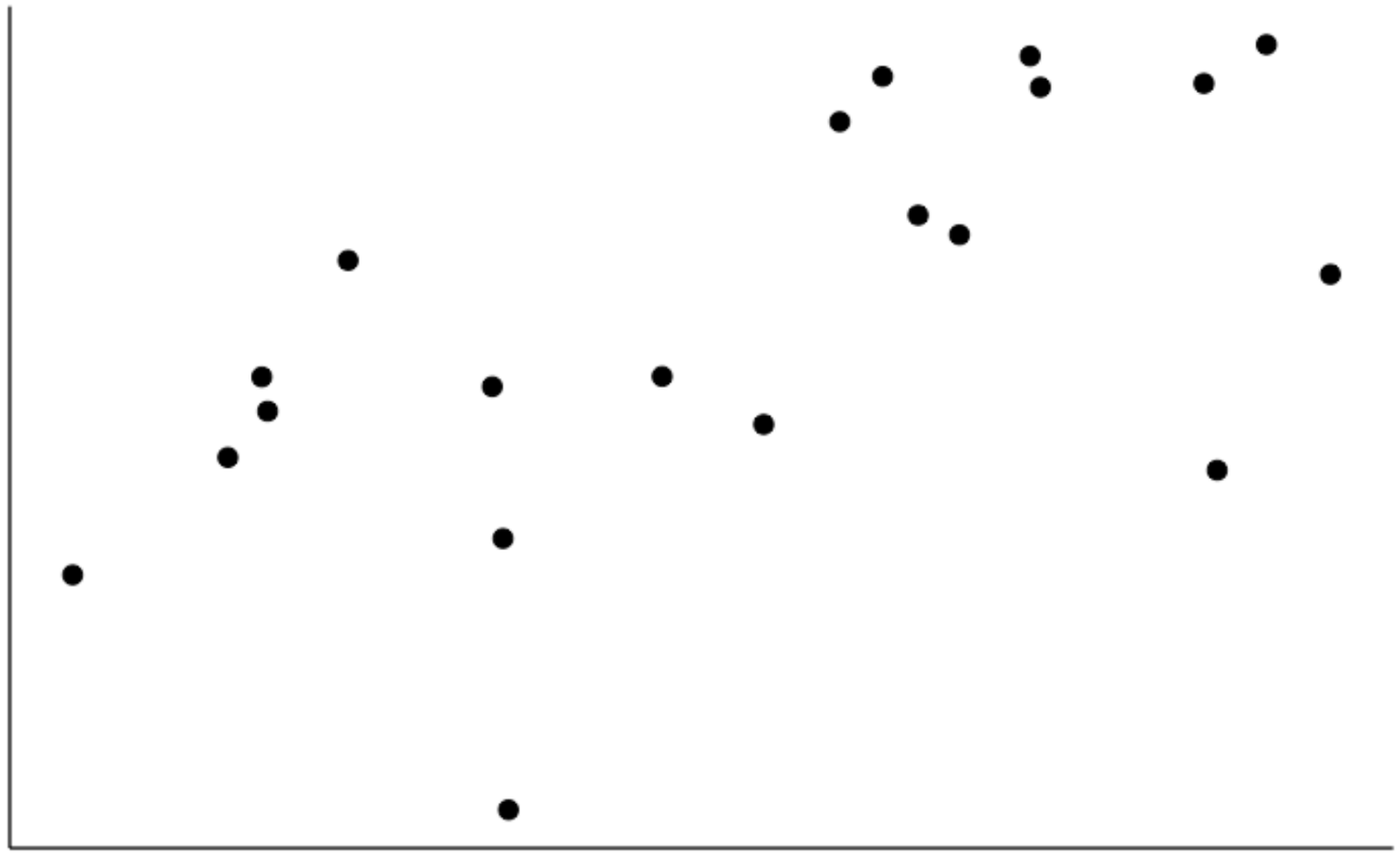


(I was too lazy to draw rectangles ...)

How well does the mean capture the data?

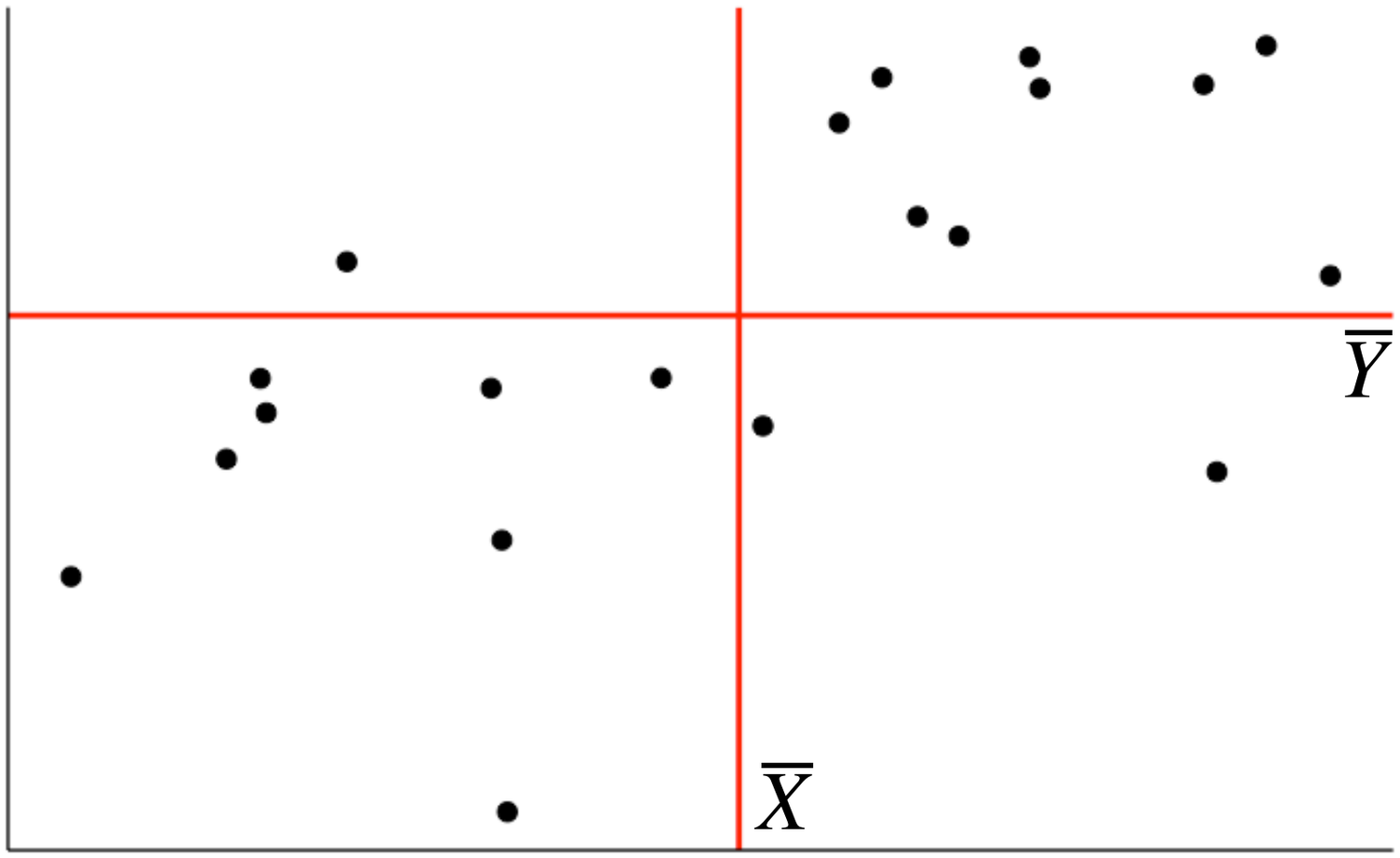
sample covariance

$$Cov(X, Y) = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{n - 1}$$



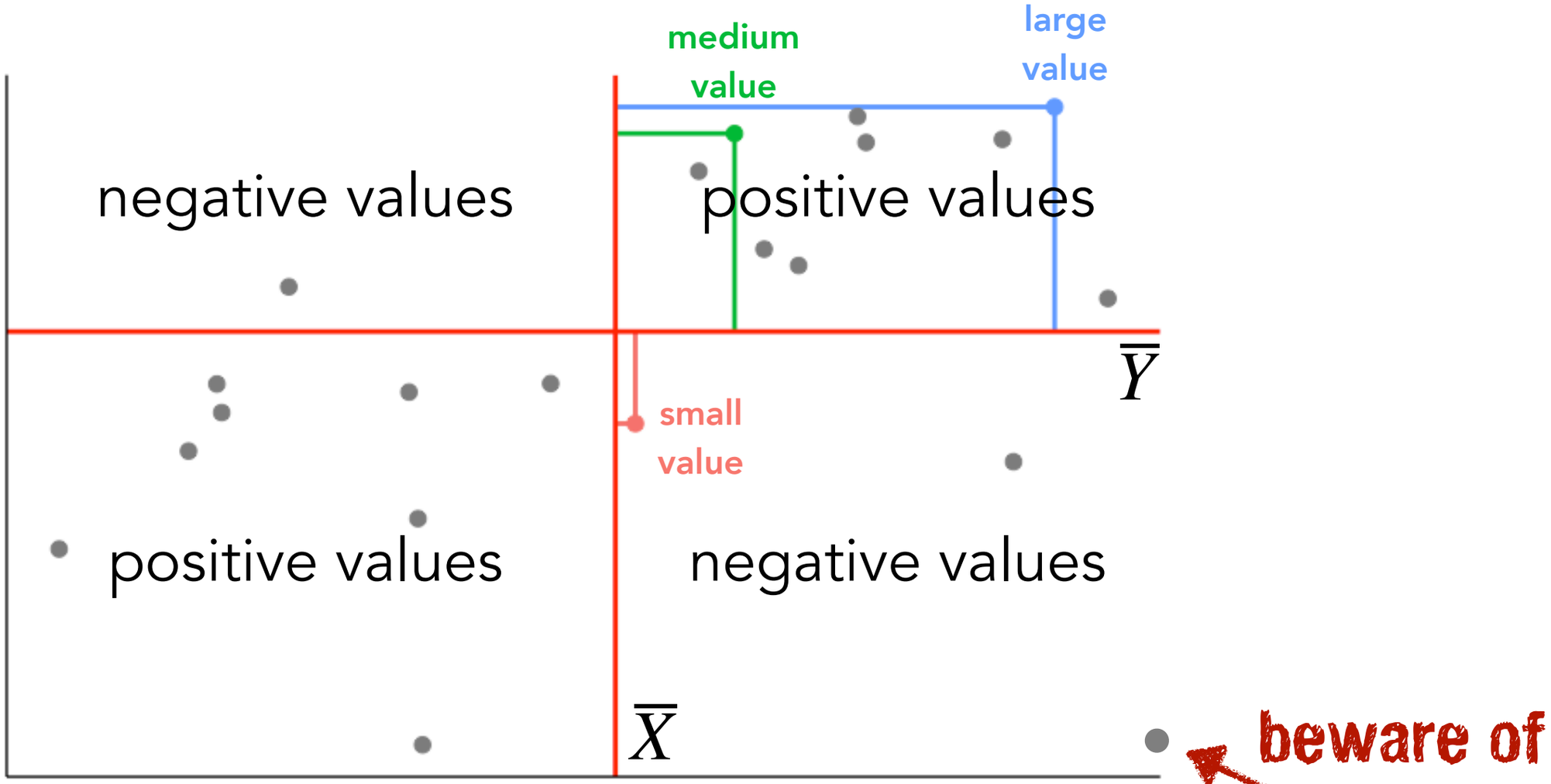
sample covariance

$$Cov(X, Y) = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{n - 1}$$



sample covariance

$$Cov(X, Y) = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{n - 1}$$



sample covariance

$$Cov(X, Y) = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{n - 1}$$

depends on the scale of the variables

sample correlation coefficient

the $n - 1$s cancel out

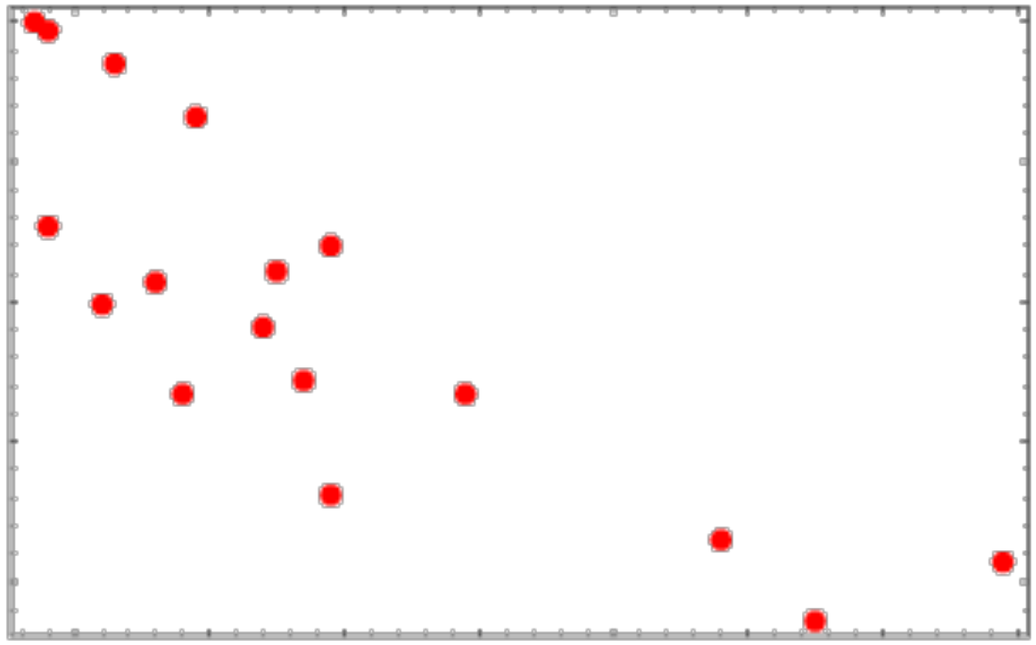
$$r(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \cdot \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}}$$

standardized covariation
(dividing by the standard deviations)

Properties of the Pearson correlation

- standardized: $-1 \leq r \leq 1$
- scale independent (for both X and Y)
- commutativity: $r(X, Y) = r(Y, X)$ ← association not causation
- sign determines the direction of dependence
- captures **linear dependence** only

Who is the correlation champion?



The faster you respond the more points you get!

-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

Who is the correlation champion?

Get ready to compete!

In what range is the correlation coefficient?

- -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

In what range is the correlation coefficient?

- -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

In what range is the correlation coefficient?

- -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

Leaderboard

In what range is the correlation coefficient?

- -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

In what range is the correlation coefficient?

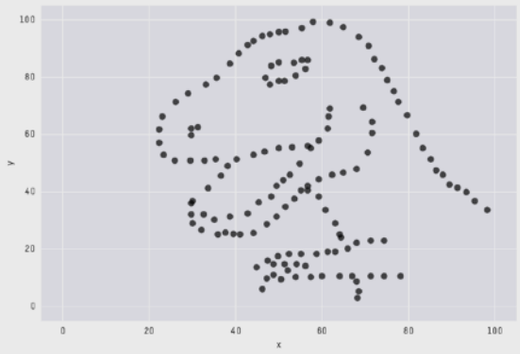
- -1 : -0.75
- -0.75 : -0.5
- -0.5 : -0.25
- -0.25 : 0
- 0 : 0.25
- 0.25 : 0.5
- 0.5 : 0.75
- 0.75 : 1

In what range is the correlation coefficient?

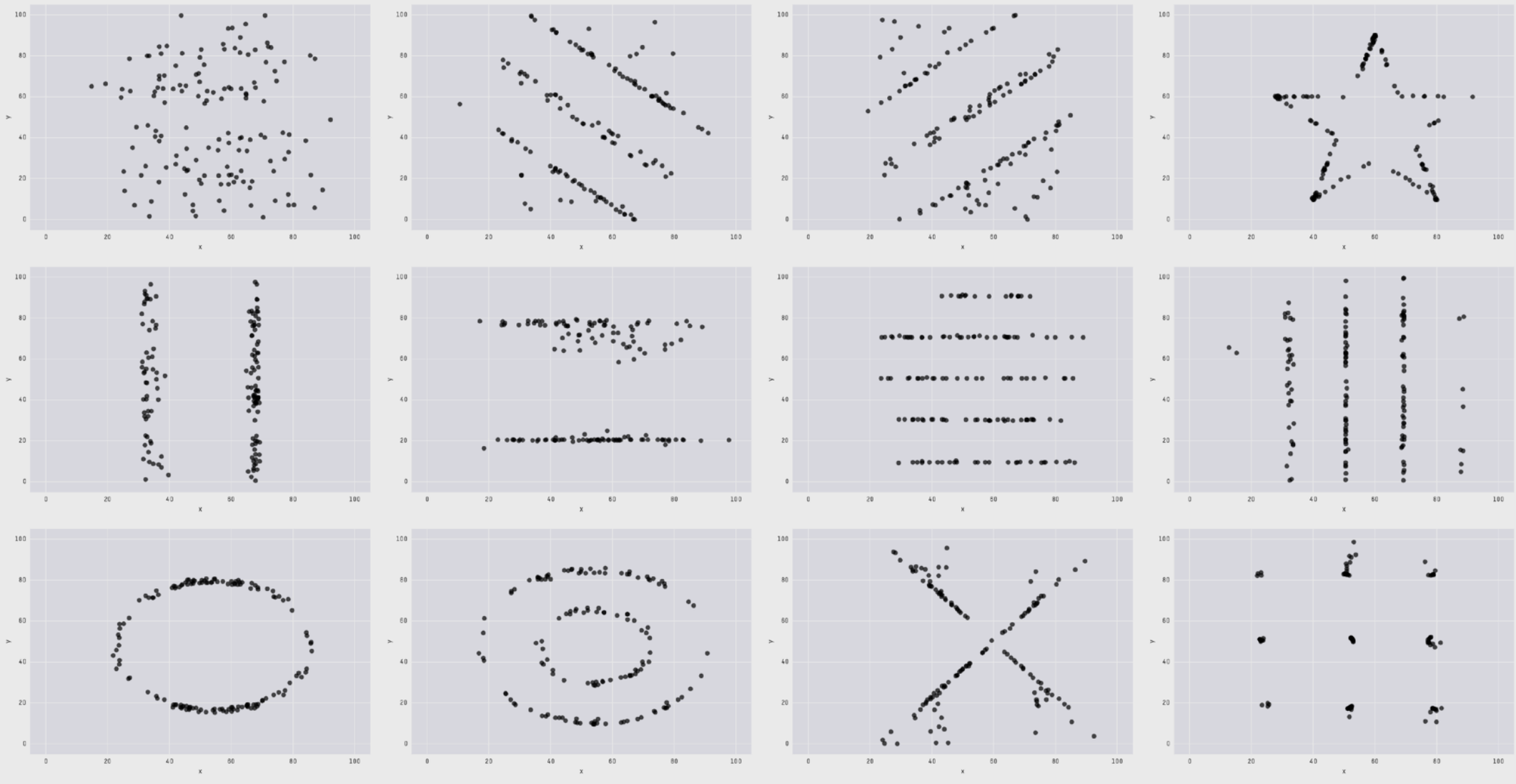
- -1 : -0.75
- -0.75 : -0.5
- -0.5 : -0.25
- -0.25 : 0
- 0 : 0.25
- 0.25 : 0.5
- 0.5 : 0.75
- 0.75 : 1

Leaderboard

Be careful about interpreting correlations!



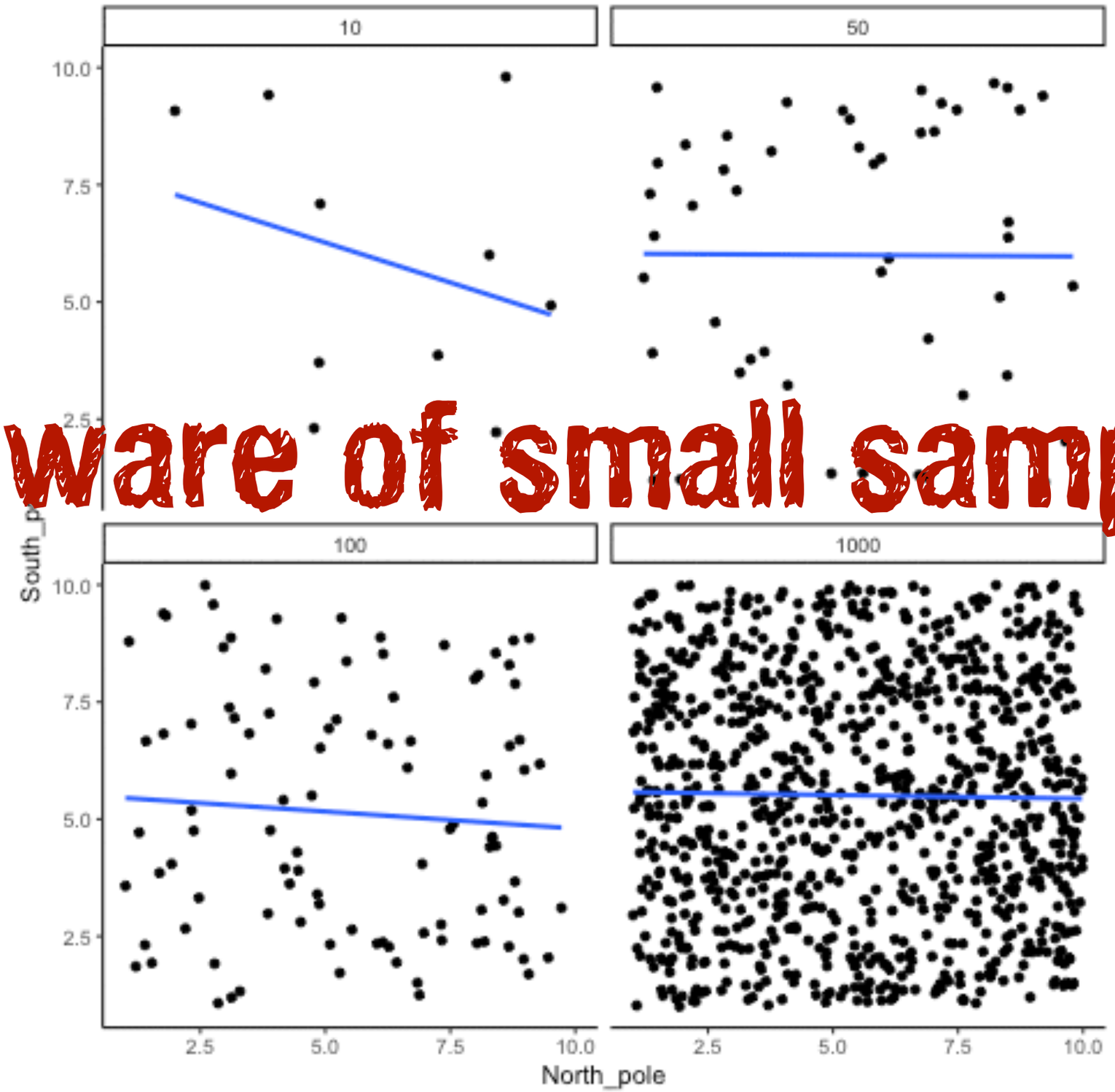
X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06



always visualize the data ...

$$n = [10,\ 50,\ 100,\ 1000]$$

$$X \sim \mathcal{U}(\min = 0,\ \max = 10)$$
$$Y \sim \mathcal{U}(\min = 0,\ \max = 10)$$

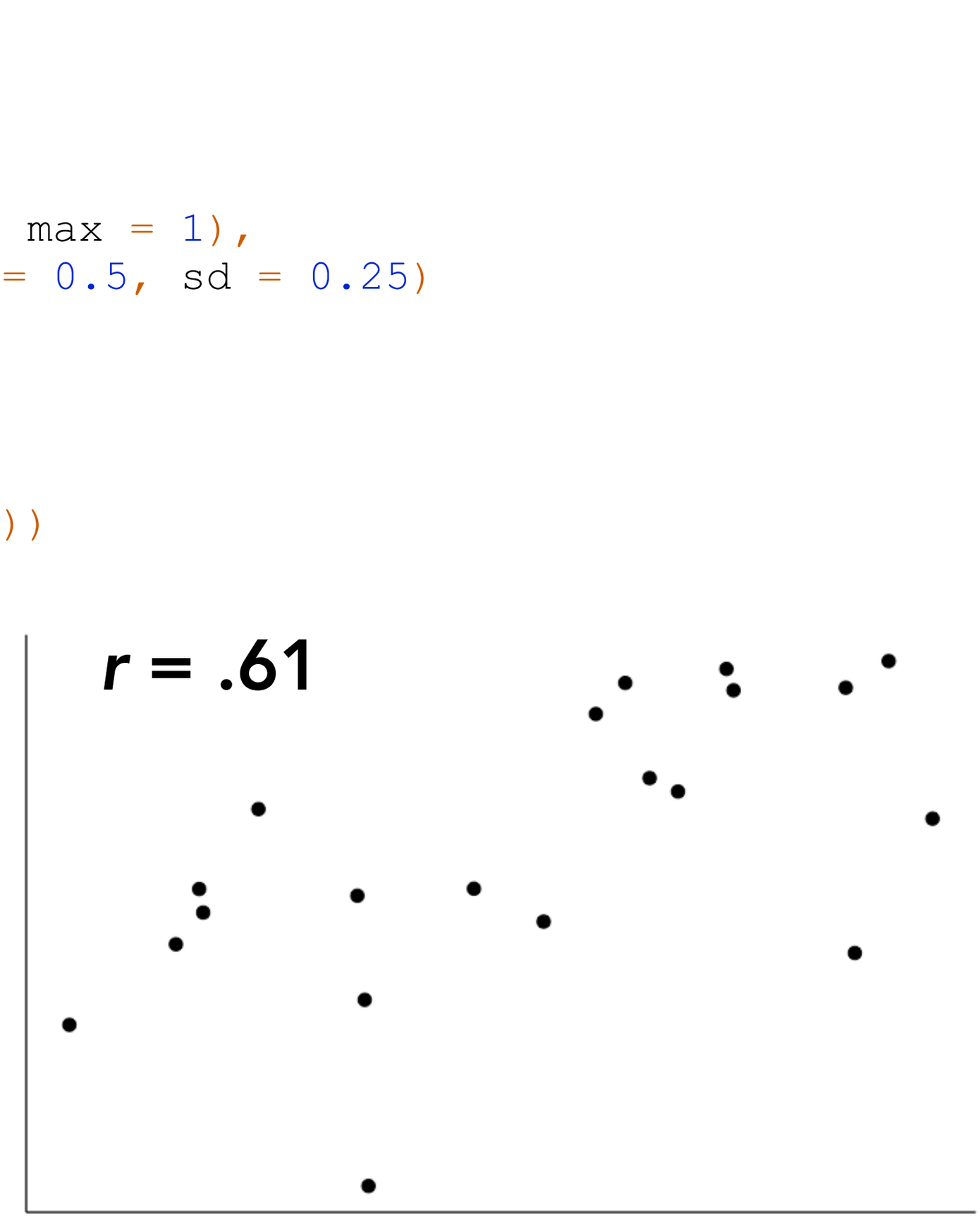


Beware of small samples!

in R

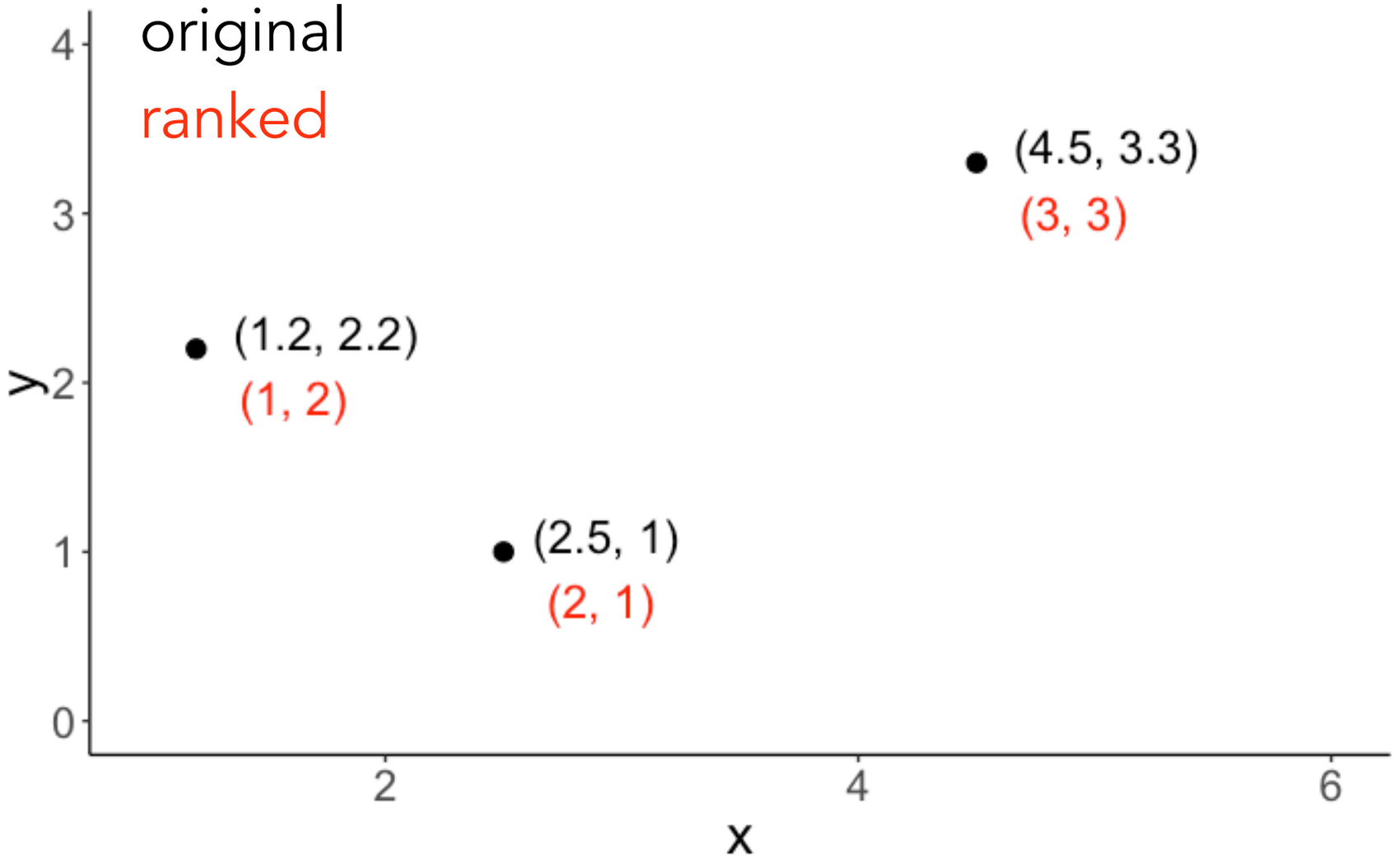
```r
# data set
df.correlation = tibble(
  x = runif(20, min = 0, max = 1),
  y = x + rnorm(x, mean = 0.5, sd = 0.25)
)

# correlation
df.correlation %>%
  summarize(r = cor(x, y))
```

function to calculate correlation



Spearman rank order correlation

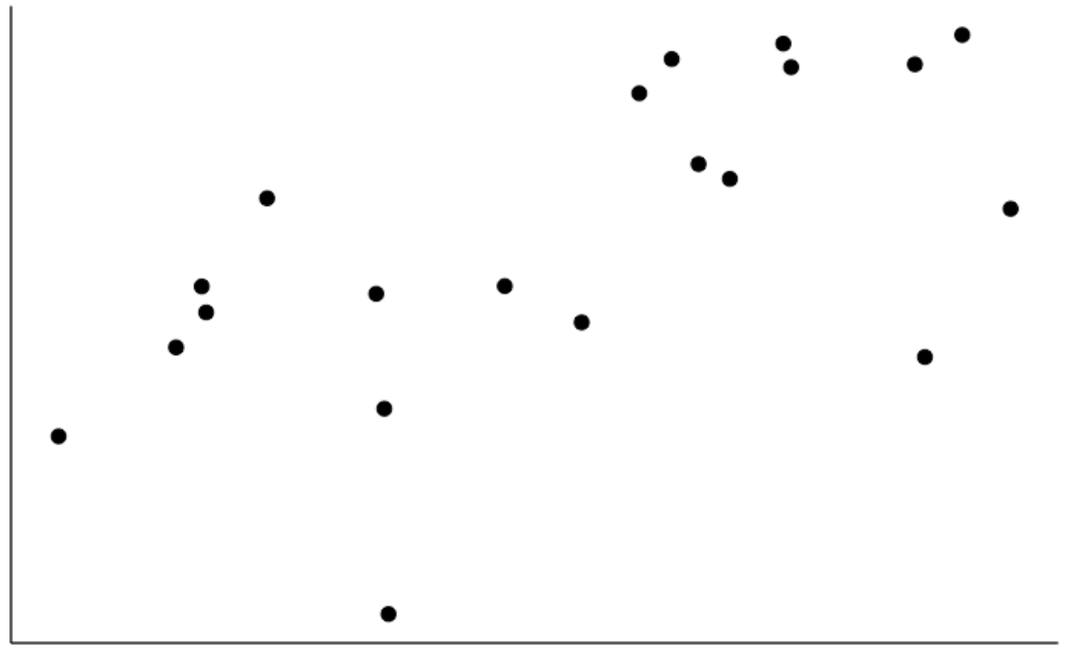
- transform original data into ranks
- calculate correlation on the ranked data



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

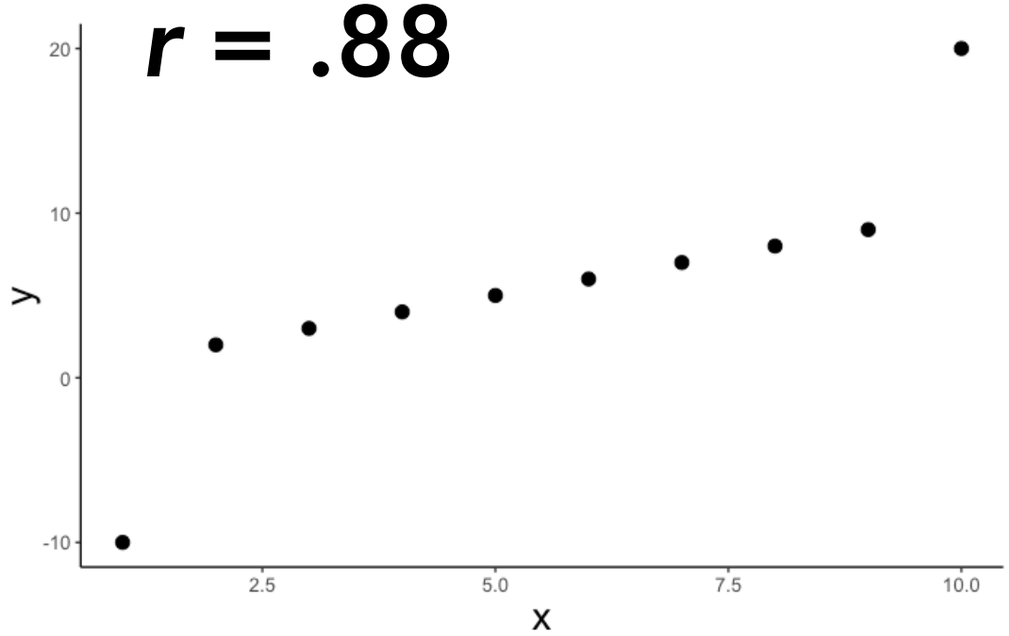


r	spearman	r_ranks
0.609	0.595	0.595

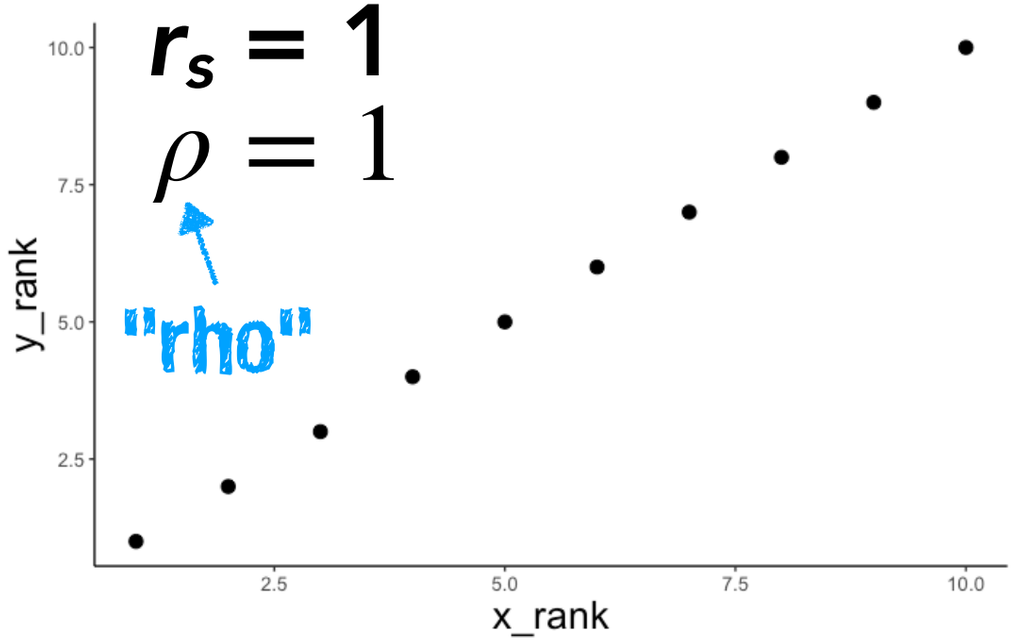
```
1 # correlation
2 df.spearman %>%
3   summarize(r = cor(x, y, method = "pearson"),
4             spearman = cor(x, y, method = "spearman"),
5             r_ranks = cor(x_rank, y_rank))
```

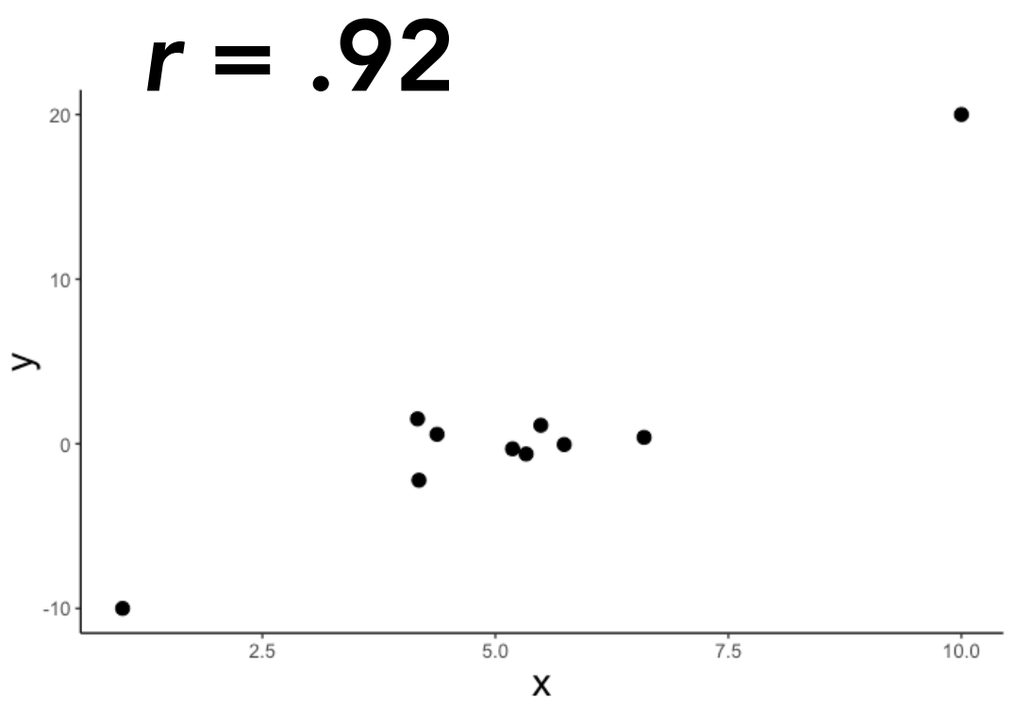
Spearman rank order correlation

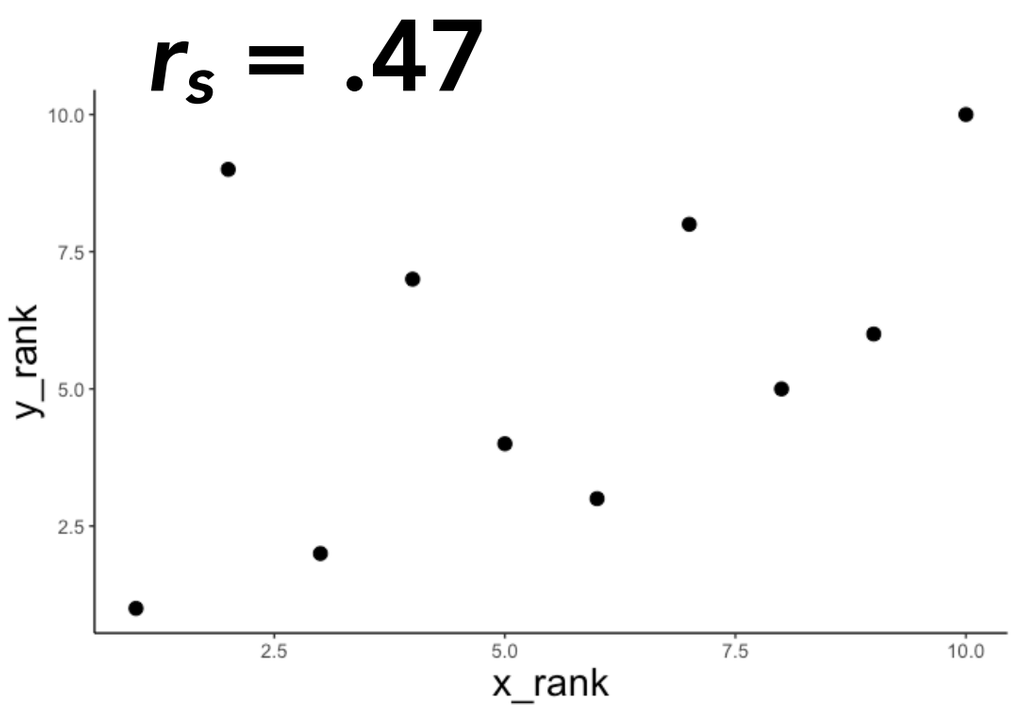
original



ranked







Pearson vs. Spearman

- Pearson's *r* captures the extent to which the relationship between two variable is **linear**
- Spearman's ρ 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

Regression

The conceptual tour

Linear model: Simple regression

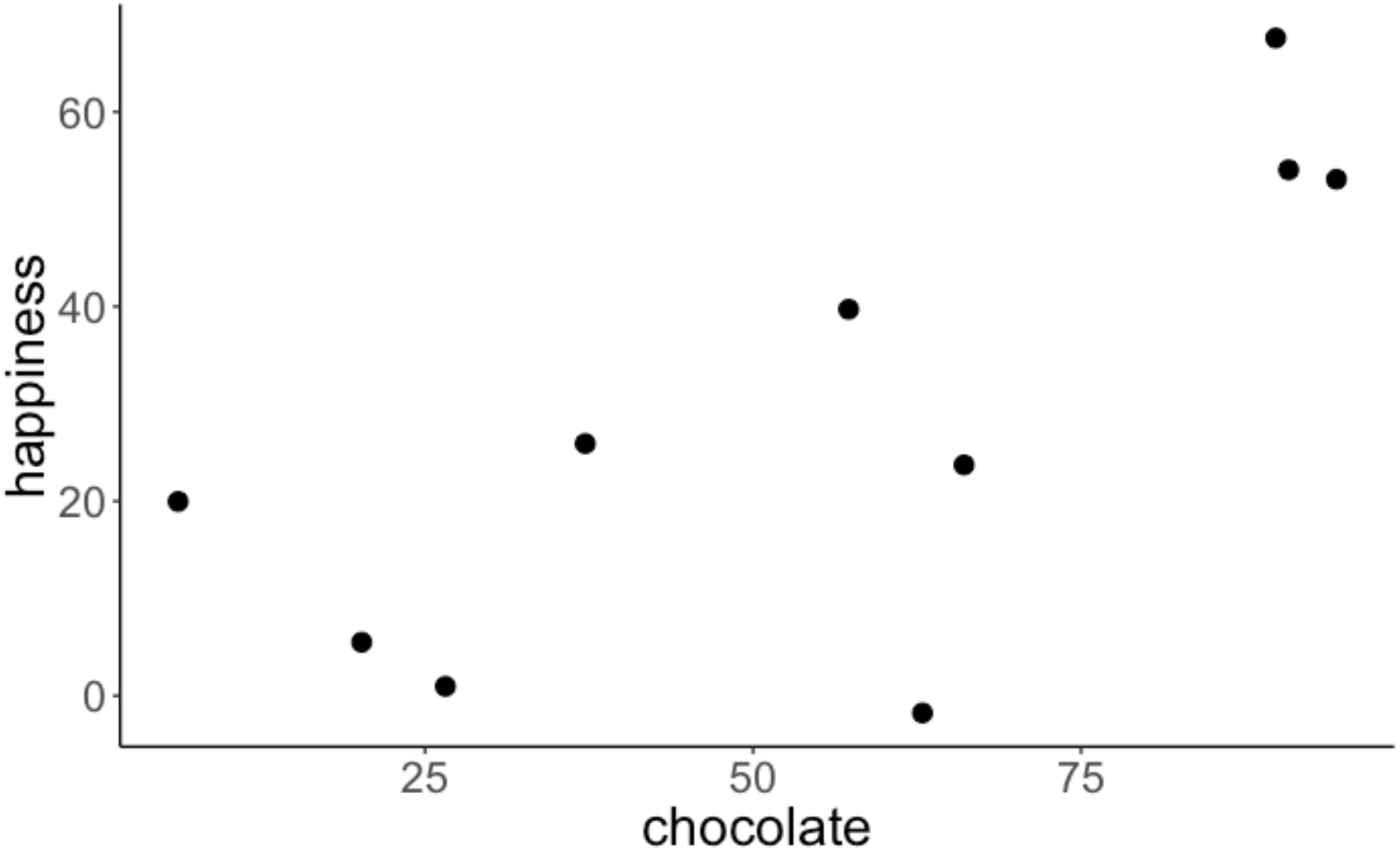
$$\texttt{Data} = \texttt{Model} + \texttt{Error}$$

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

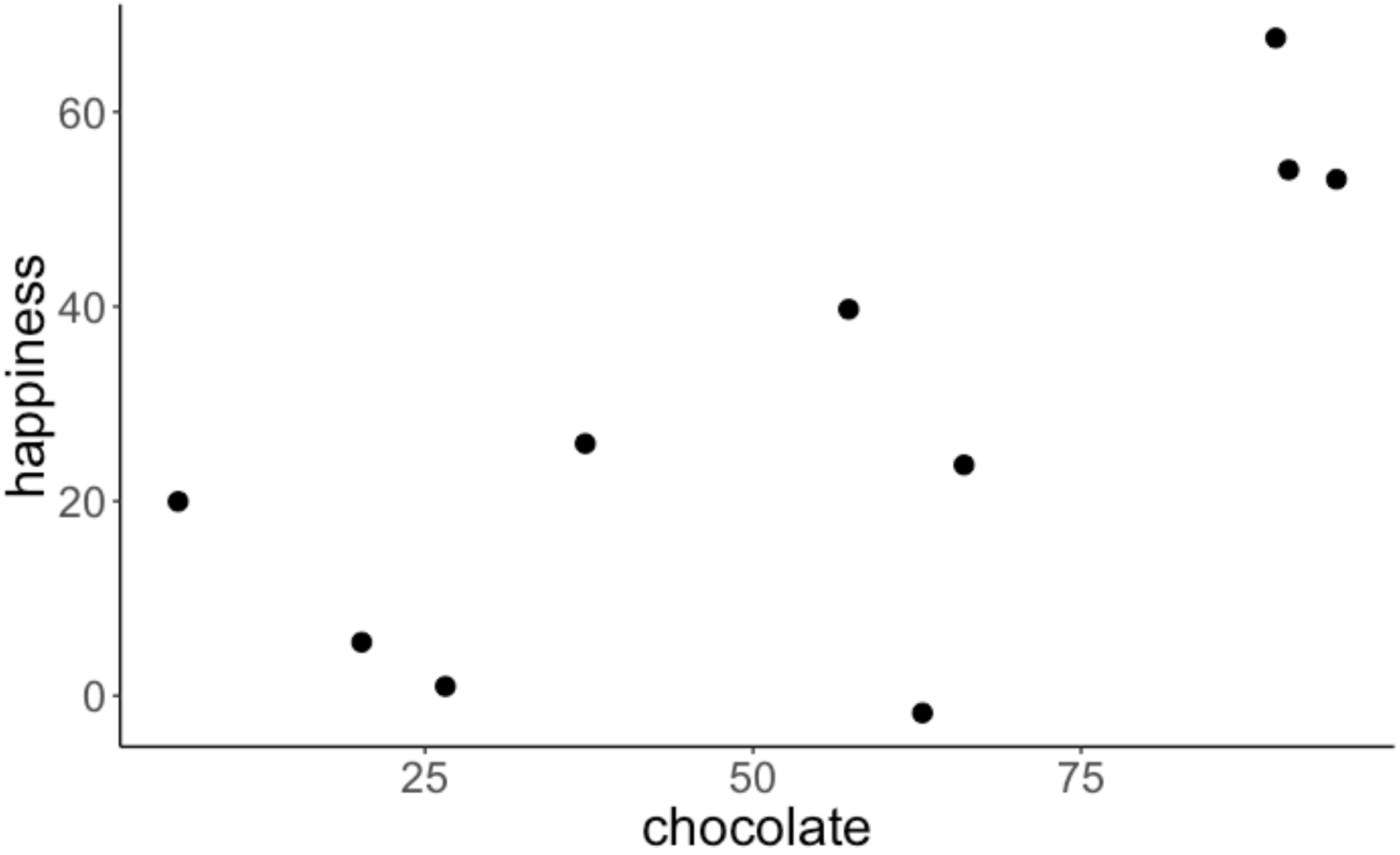
$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$

the model is a linear combination of predictors

Does chocolate make us happy?



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

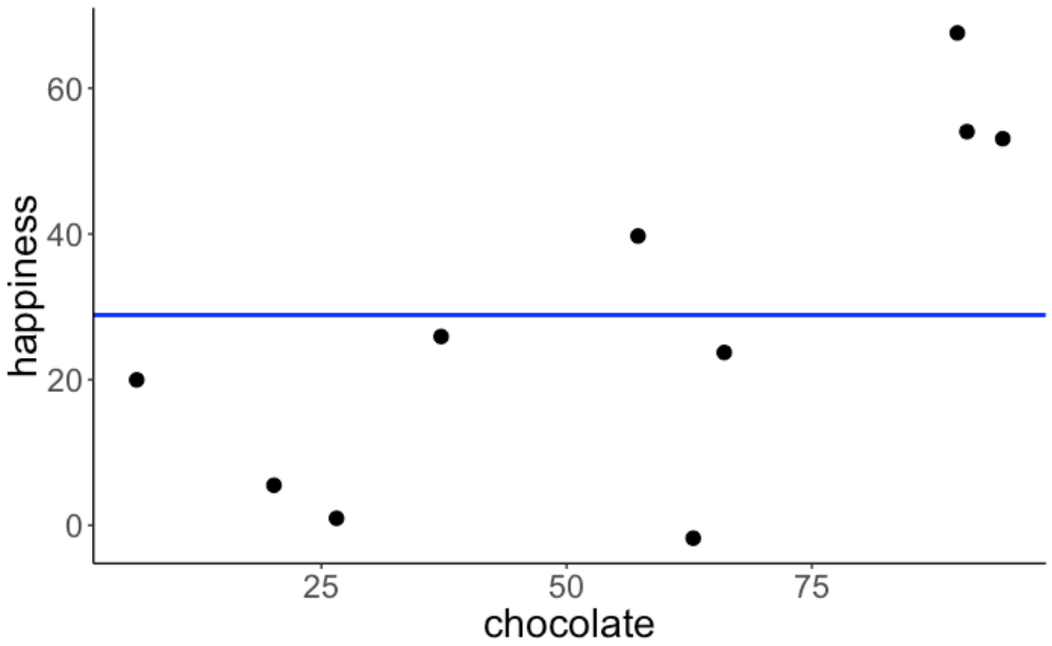
1. Define H_0 as Model C (compact) and H_1 as Model A (augmented)
2. **Fit model parameters to the data**
3. Calculate the proportional reduction of error (PRE) in our sample
4. Decide whether the augmented model is **worth it** by comparing the observed PRE in our sample to the sampling distribution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

$$Y_i = 28.88 + e_i$$

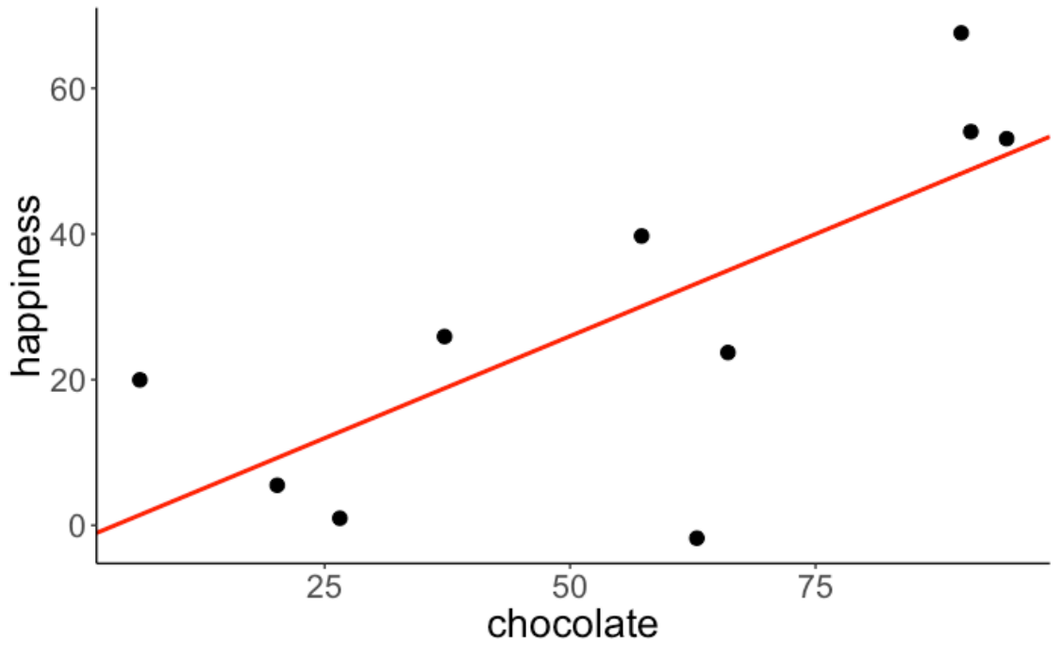
H_1: Chocolate consumption and happiness are related.

Model A

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

chocolate consumption

Model prediction



Fitted model

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

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)

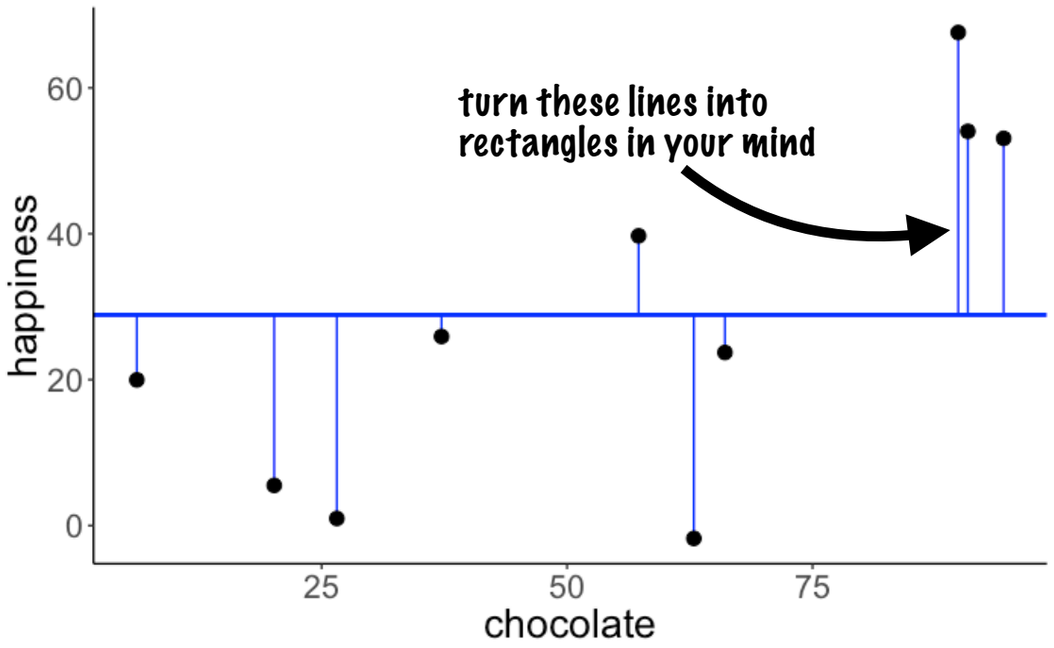
Calculate PRE

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

Both models were fit to minimize the sum of squared errors

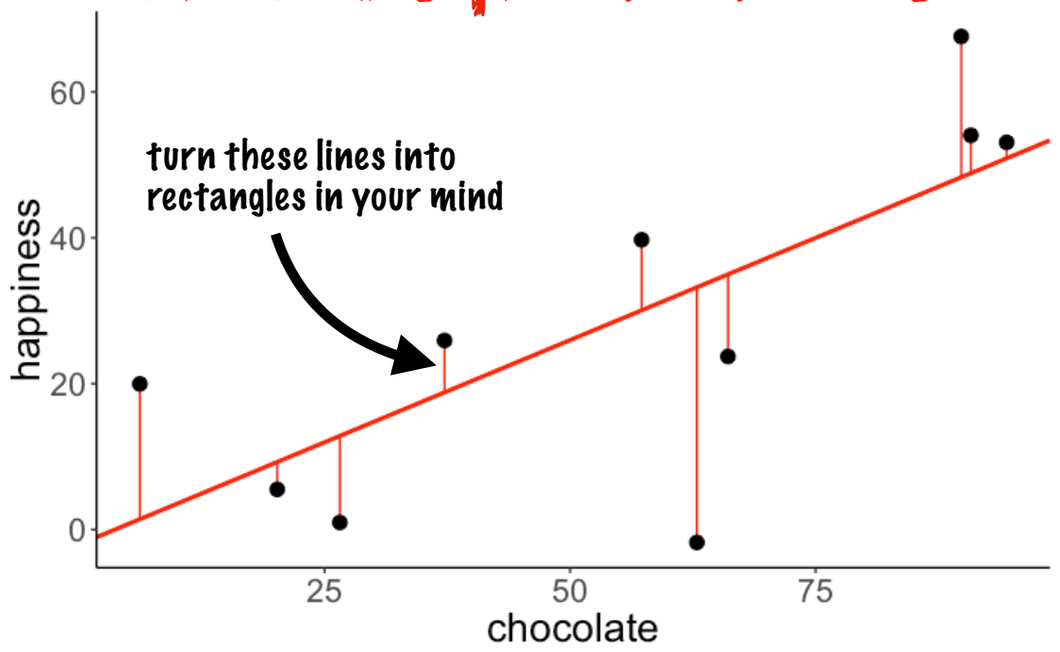
OLS = Ordinary **least squares** regression

Sum of squared errors



SSE(C) = 5215.016

Sum of squared errors



SSE(A) = 2396.946

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

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 $model_C$ to create $model_A$ but were not

difference in parameters between models A and C

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

number of observations vs. parameters in Model A

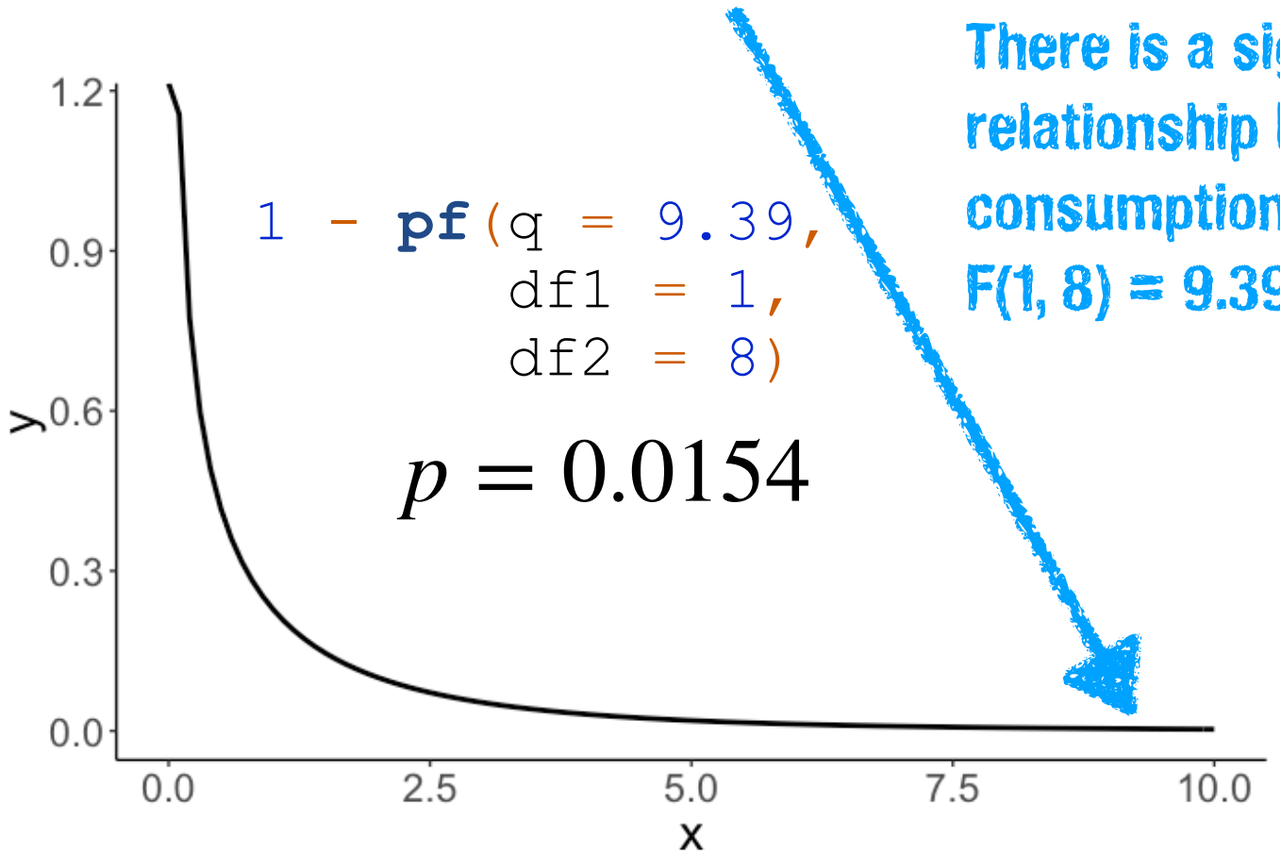
Decide whether it's **worth it**

- To compute the *F* statistic, we need:
  - PRE = 0.54
  - PC = 1
  - PA = 2
  - $n = 10$

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

$$= \frac{0.54/(2 - 1)}{(1 - 0.54)/(10 - 2)}$$

$$= 9.39$$



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

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

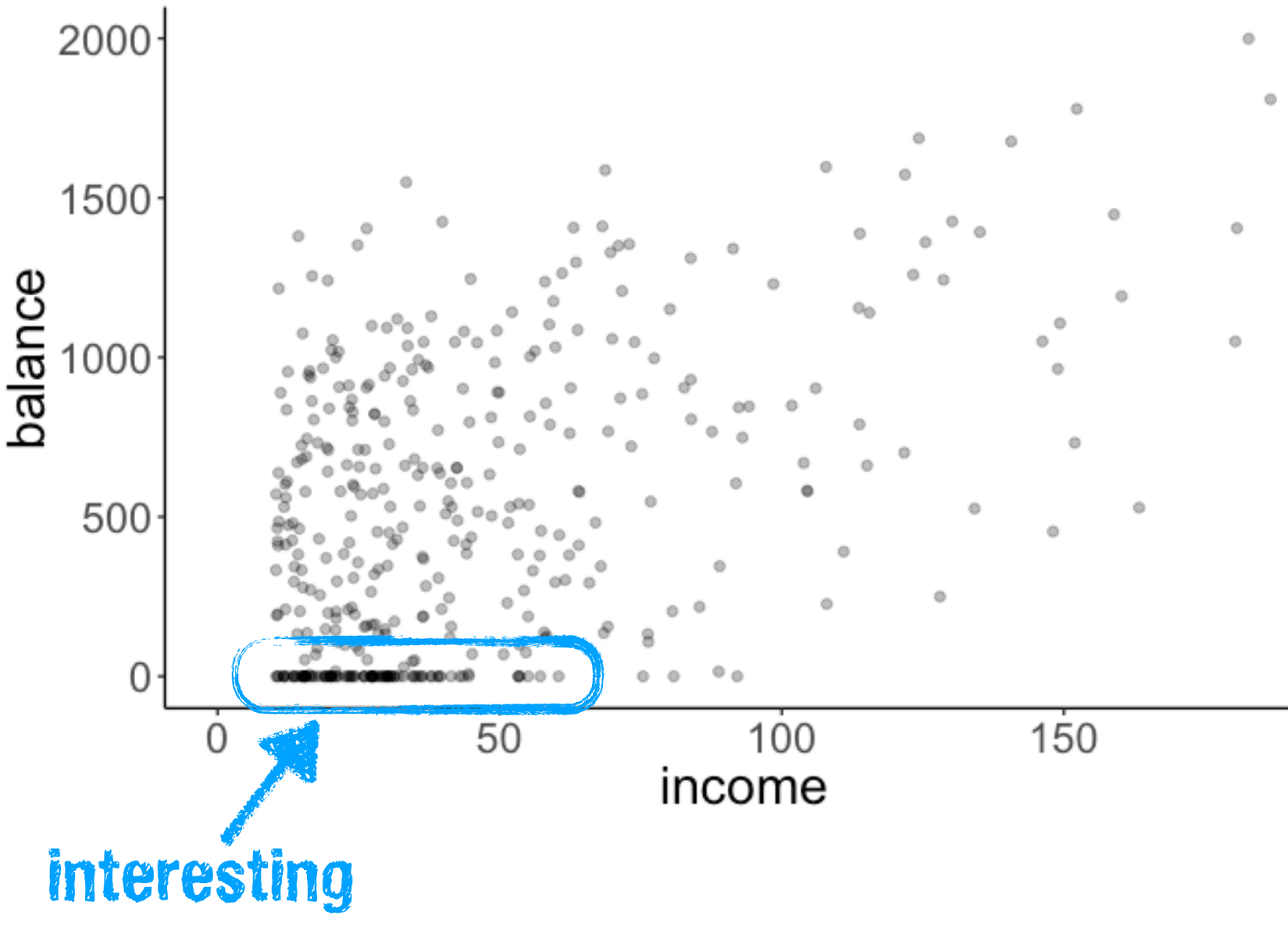
variable	description
income	in thousand dollars
limit	credit limit
rating	credit rating
cards	number of credit cards
age	in years
education	years of education
gender	male or female
student	student or not
married	married or not
ethnicity	African American, Asian, Caucasian
balance	average credit card debt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

`nrow(df.credit) = 400`

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

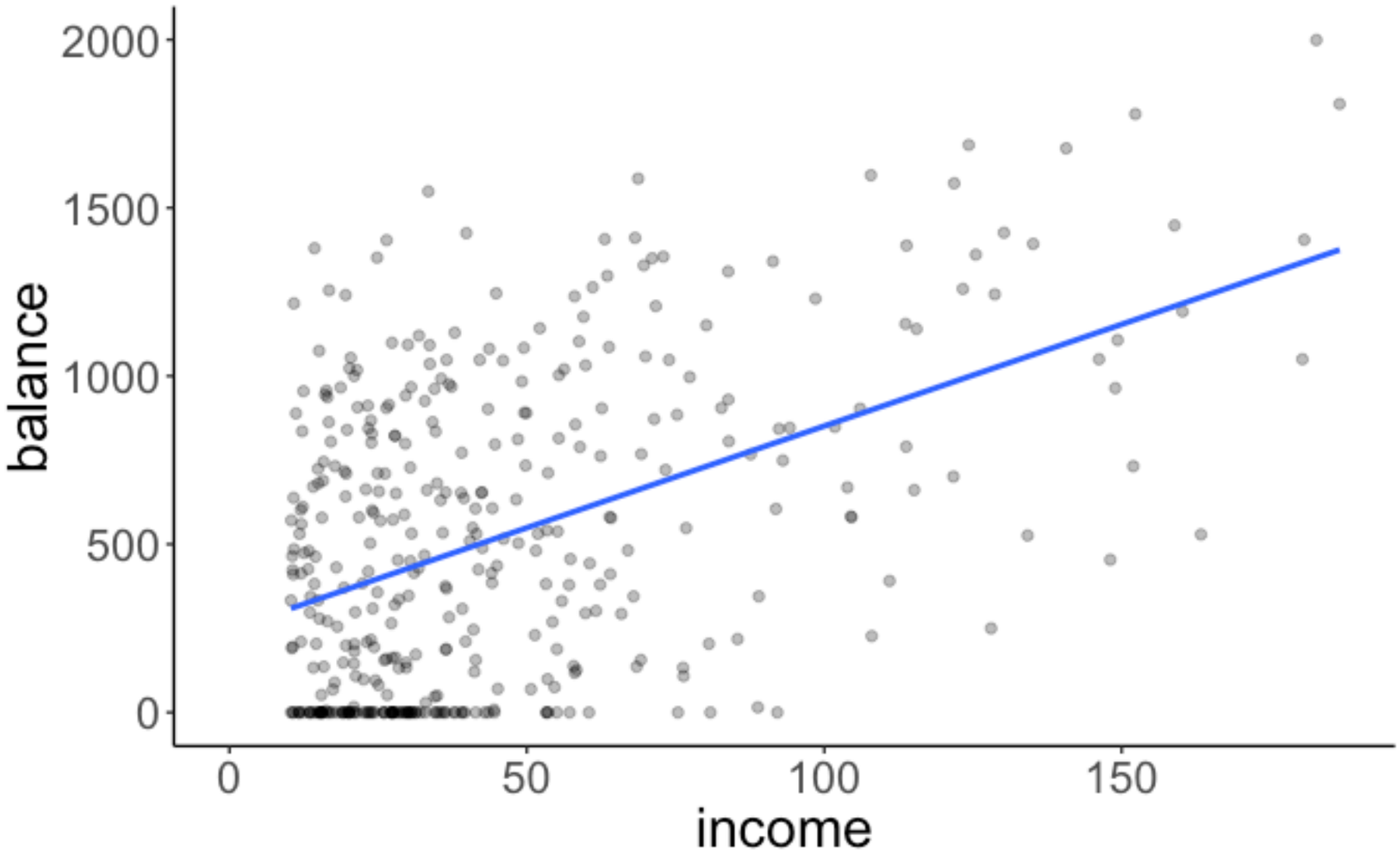
Always plot the data first ...

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

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

(doesn't need to be specified explicitly)

`lm()`

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

print(fit)
```

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

Coefficients:
(Intercept)       income
    246.515        6.048
```

parameter estimates → which minimize the squared error between model and data

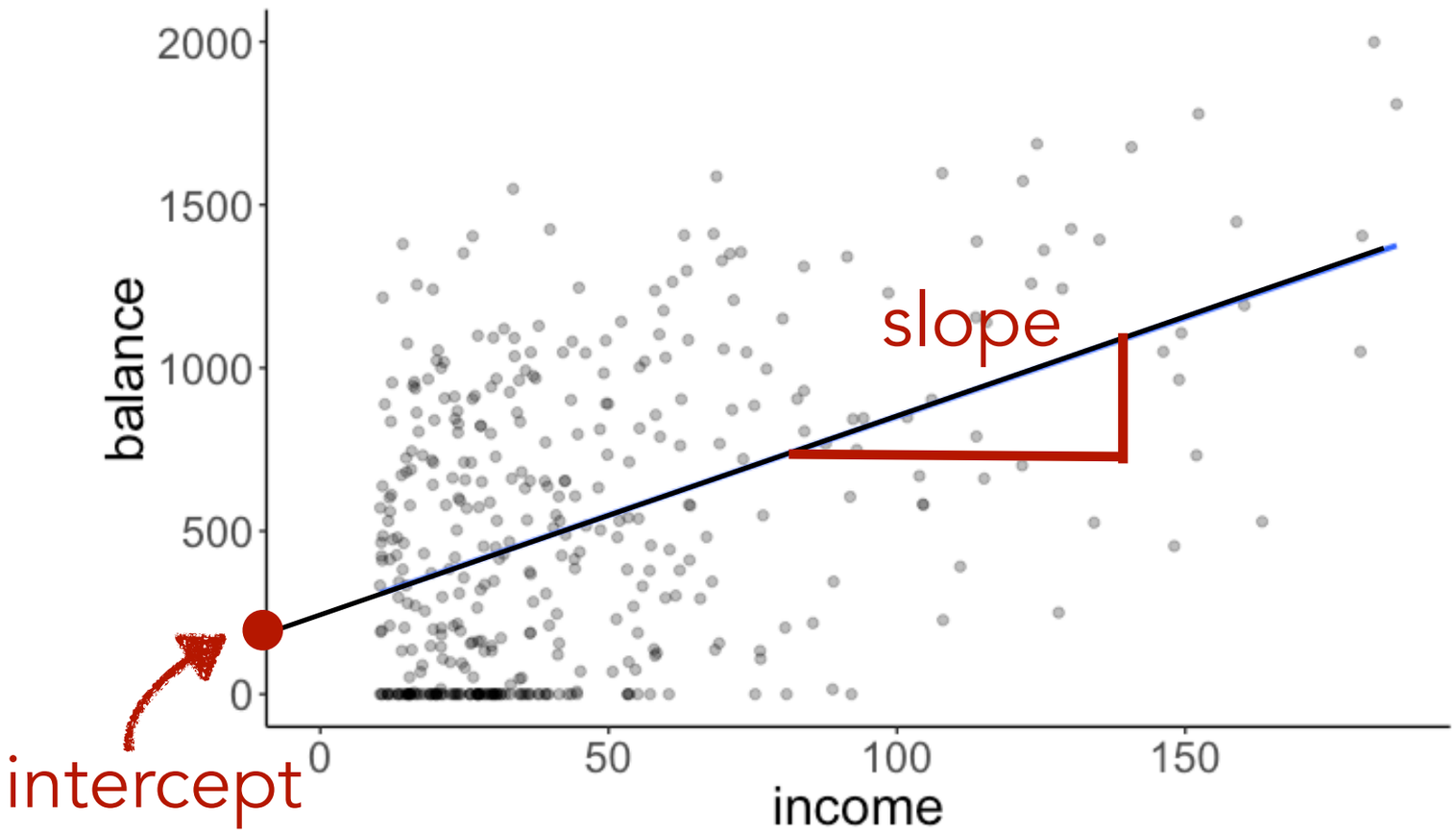
Interpreting regression parameters

```
Coefficients:
(Intercept)          income
    246.515           6.048
```

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

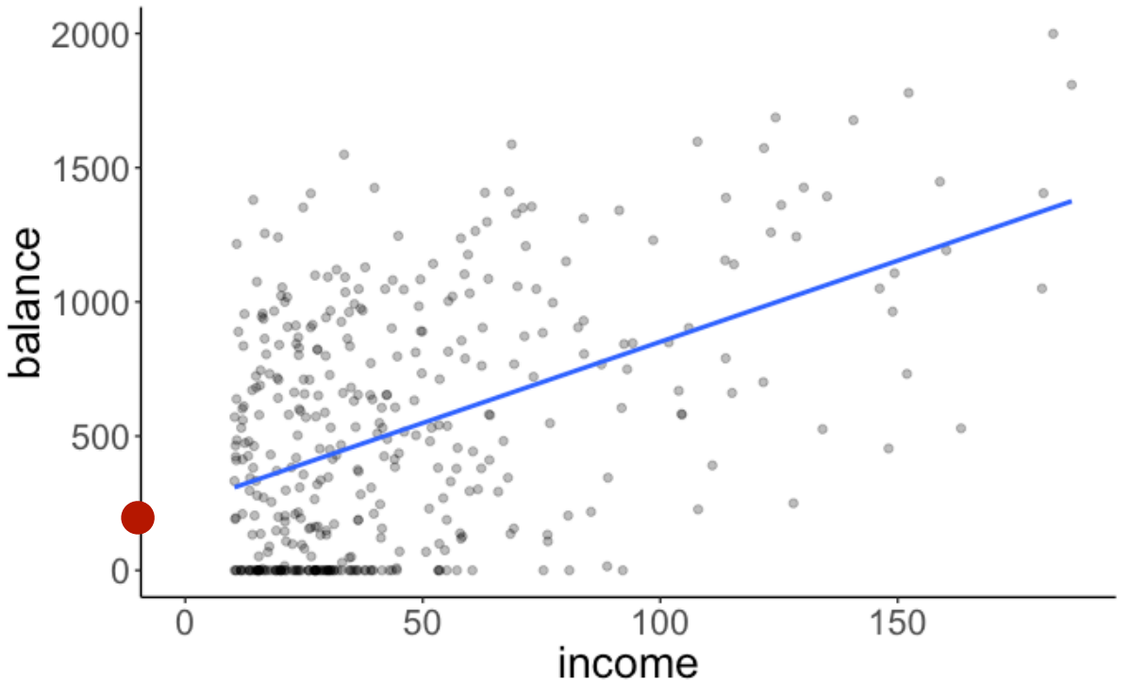
$$\text{balance}_i = b_0 + b_1 \cdot \text{income}_i + e_i$$

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



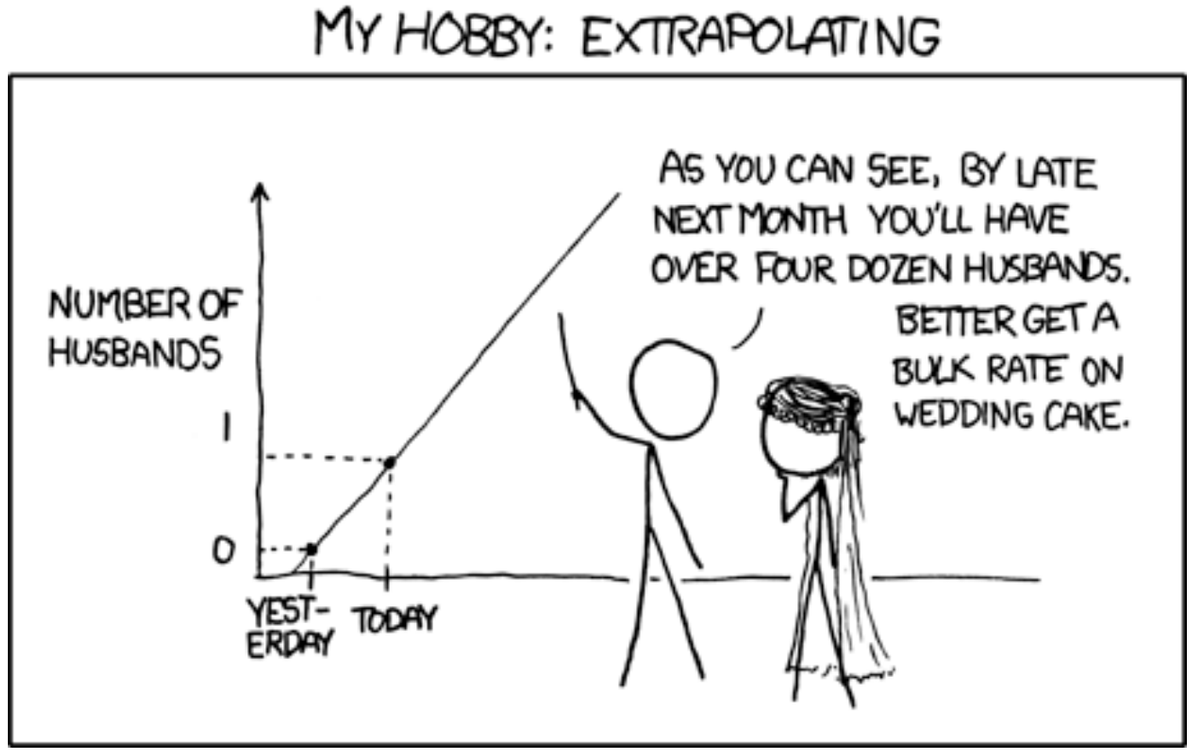
For each additional thousand dollars income, a person's average credit card debt increases by $6.05.

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, ...)

comic from slide 1



Centering the predictor

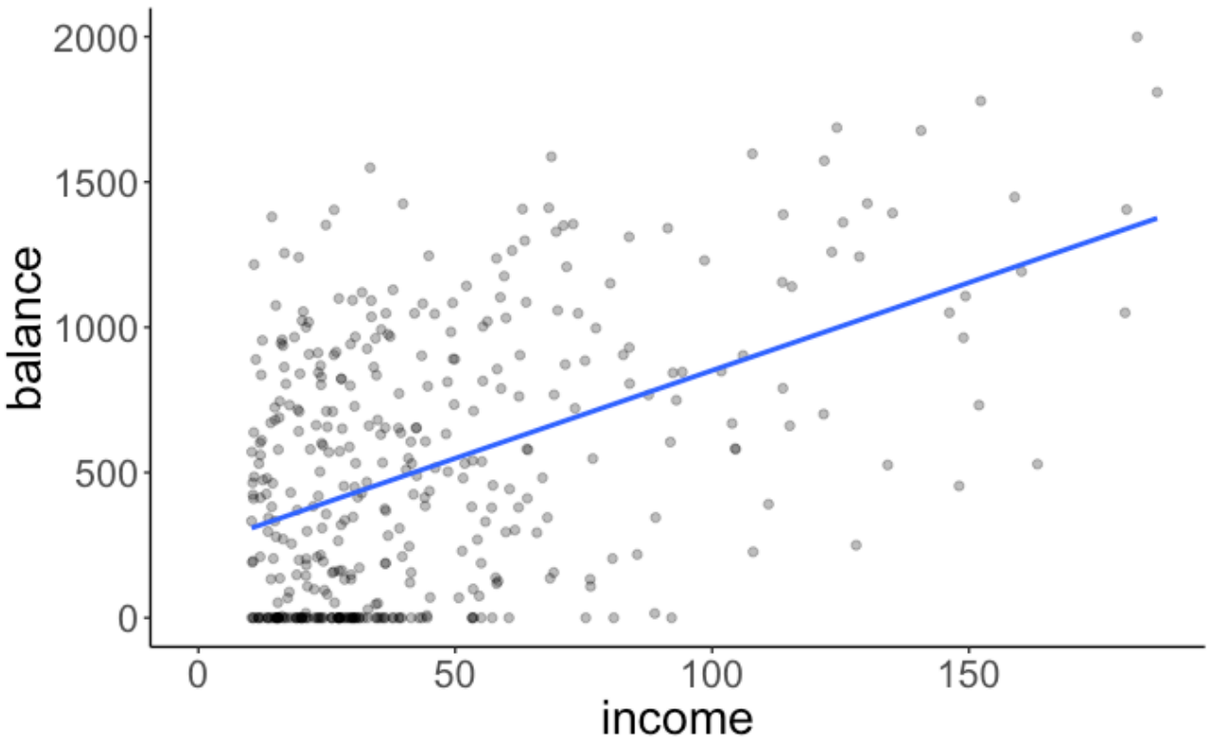
```
df.credit %>%
  mutate(income_centered = income - mean(income)) %>%
  select(balance, income, income_centered)
```

balance	income	income_centered
333	14.89	-30.33
903	106.03	60.81
580	104.59	59.37
964	148.92	103.71
331	55.88	10.66
1151	80.18	34.96
203	21.00	-24.22
872	71.41	26.19
279	15.12	-30.09
1350	71.06	25.84

Centering predictors

uncentered

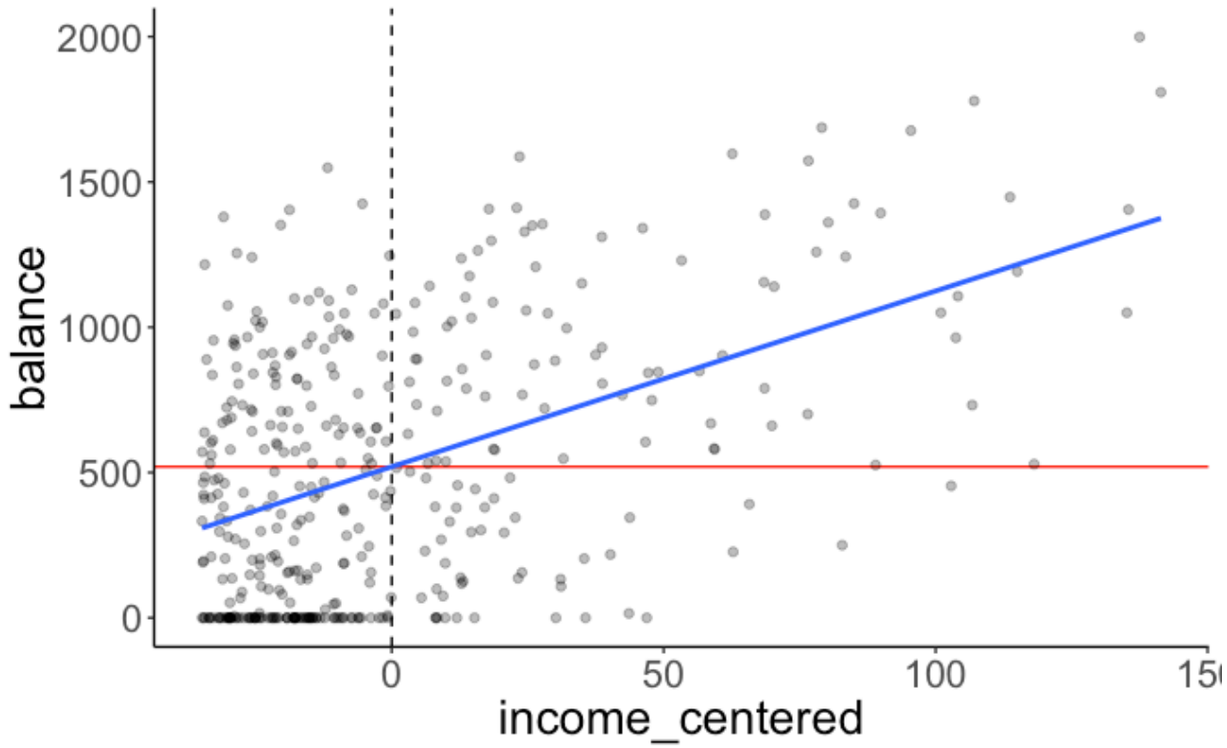
$$\text{balance}_i = 246.515 + 6.048 \cdot \text{income}_i + e_i$$



intercept = predicted value if income is 0

centered

$$\text{balance}_i = 520.015 + 6.048 \cdot \text{income\_centered}_i + e_i$$



intercept = predicted value if income is **mean**(income)

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

`summary()`

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

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

balance	income	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.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

summary()

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

```
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

```
1  fit %>%
2    augment() %>%
3    clean_names() %>%
4    summarize(
5      min = min(resid),
6      first_quantile = quantile(resid, 0.25),
7      median = median(resid),
8      third_quantile = quantile(resid, 0.75),
9      max = max(resid)
10   )
```

min	first_quantile	median	third_quantile	max
-803.64	-348.99	-54.42	331.75	1100.25

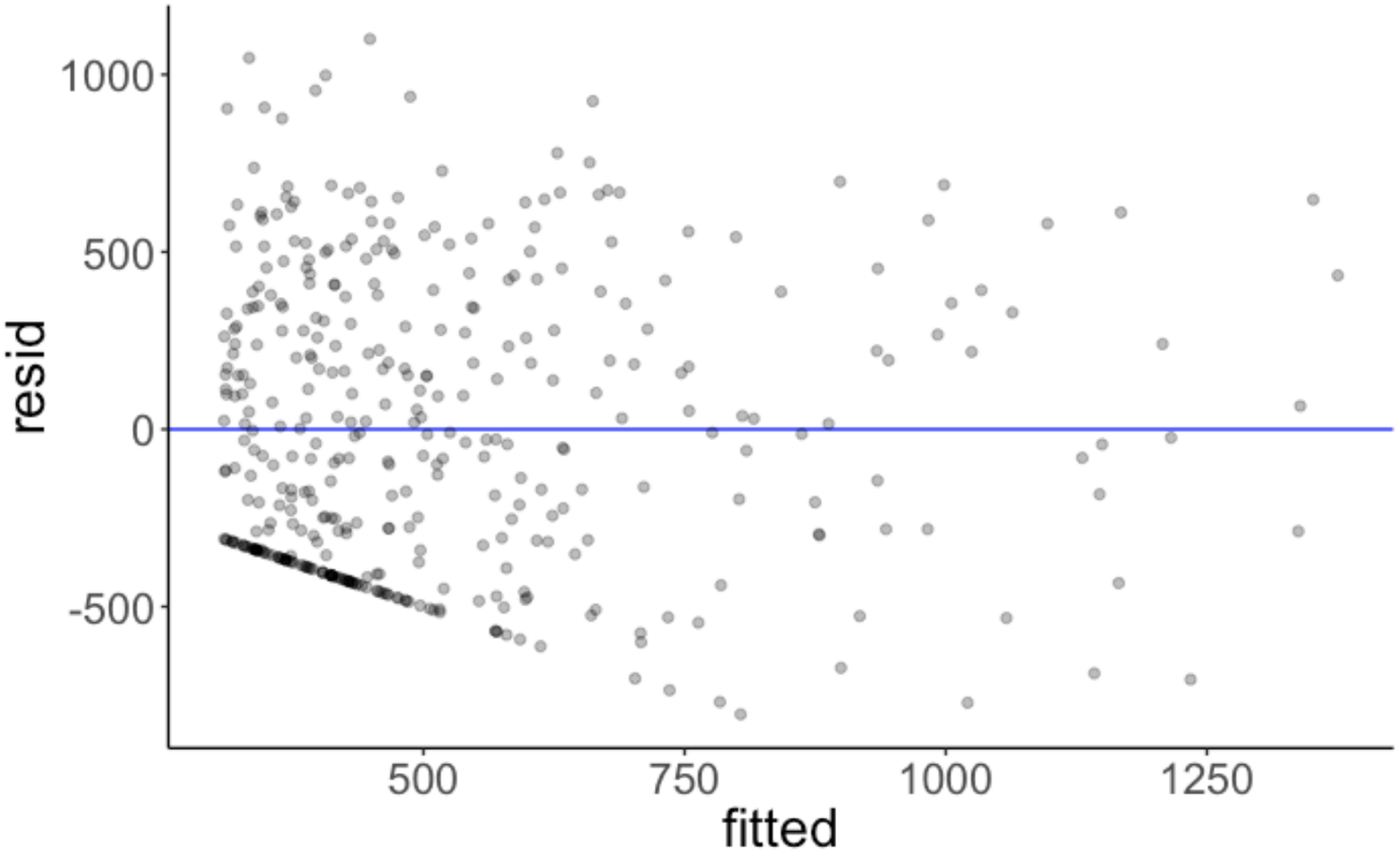
`summary()`

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

```
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



summary()

```
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)
```

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

summary()

```
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

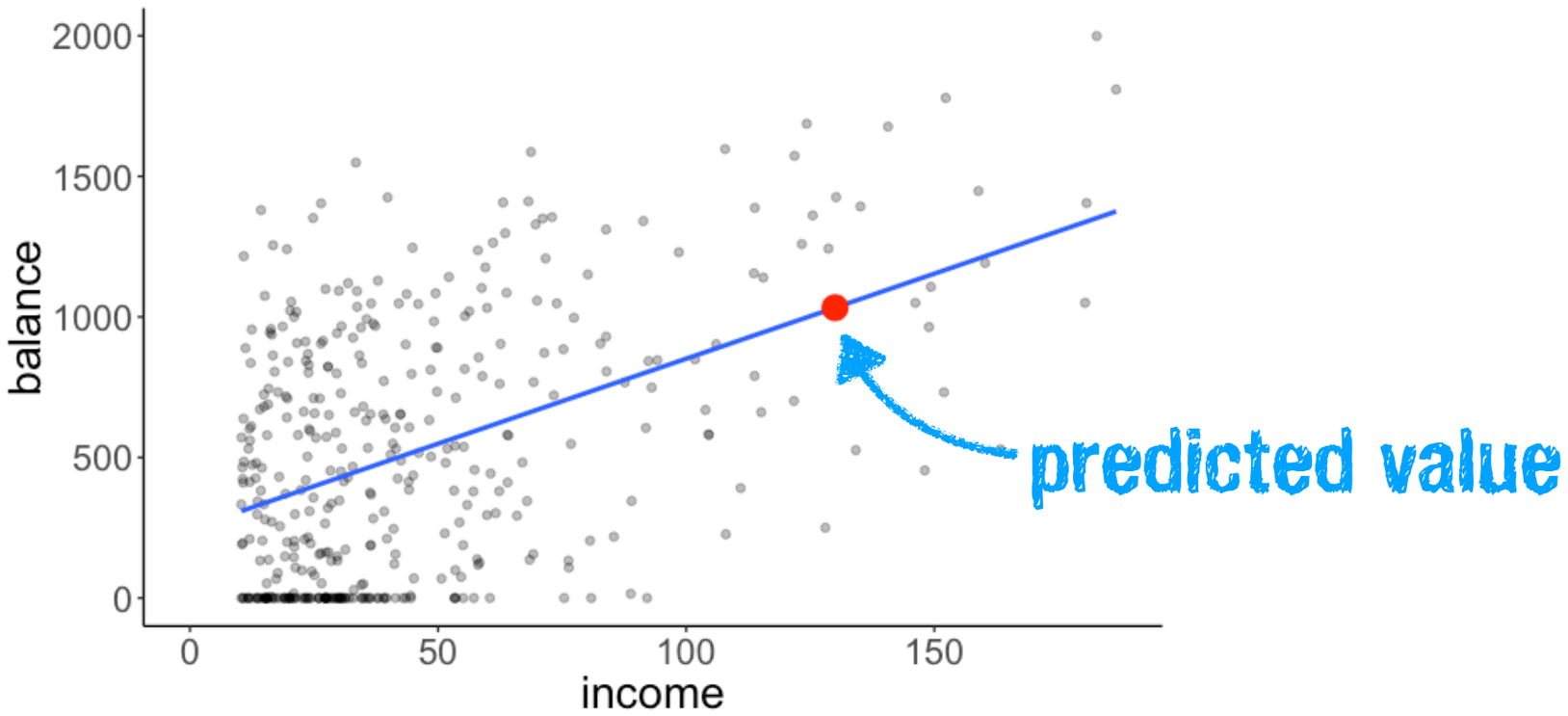
Making predictions

```
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))
```

$$\widehat{\text{balance}} = 246.515 + 6.048 \cdot 130$$
$$= 1032.755$$



Hypothesis test

Compact Model

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

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

Augmented Model

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

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

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

```
Analysis of Variance Table

Model 1: balance ~ 1
Model 2: balance ~ 1 + income
  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

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

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

---

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)

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

Hypothesis test

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

```
Analysis of Variance Table

Model 1: balance ~ 1
Model 2: balance ~ 1 + income
  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
```

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

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

- in the case of a simple regression PRE (proportion of reduced error) is identical to R^2 (variance explained)
- and R^2 is directly related to the correlation coefficient *r*

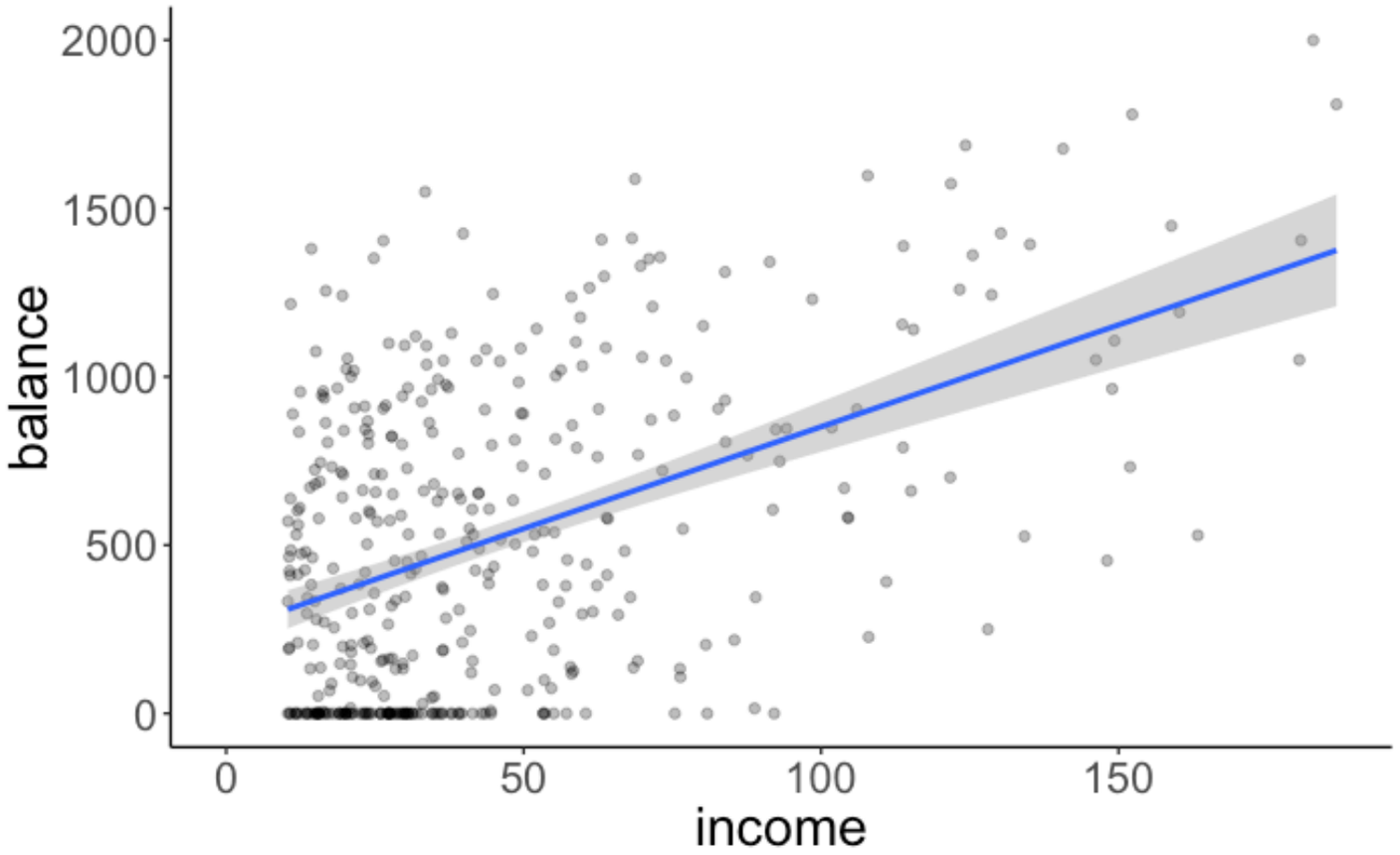
```
cor(df.credit$balance,
    df.credit$income)
```

$$R^2 = 0.215$$

$$r = .463$$

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).

The general procedure

Test whether the intercept is significantly different from 0

tells R to remove the intercept

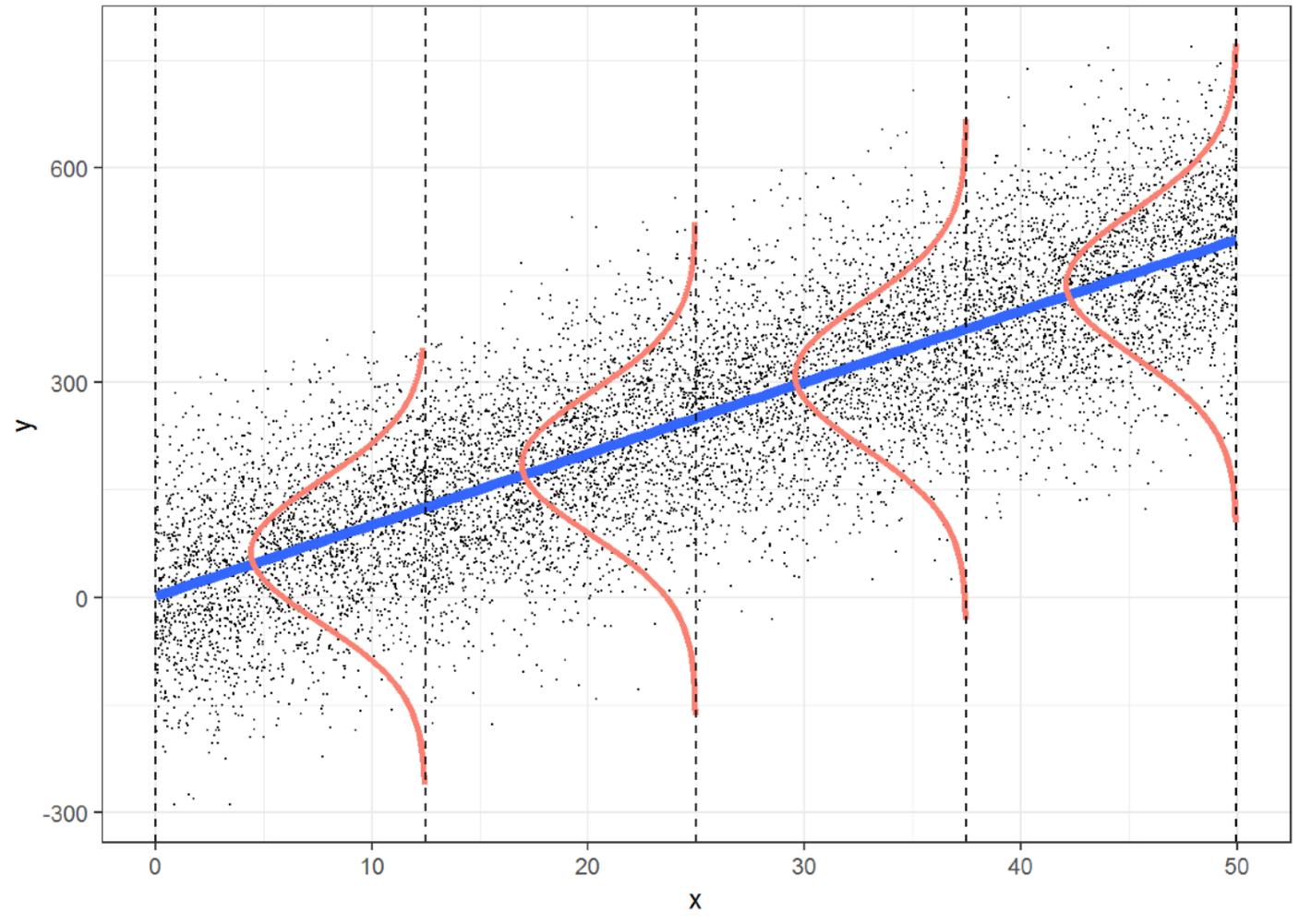
```
fit_c = lm(formula = balance ~ -1 + income,
           data = df.credit)
```

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

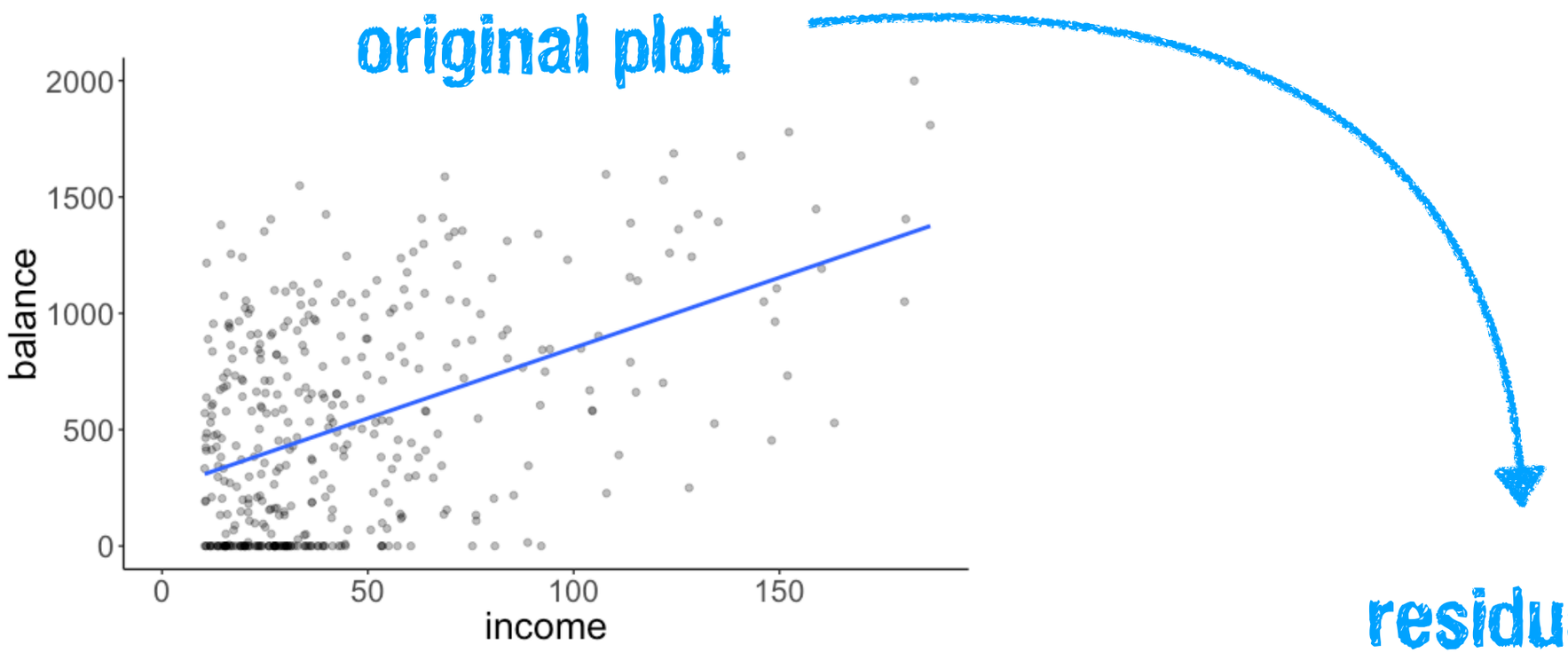
```
anova(fit_c, fit_a)
```

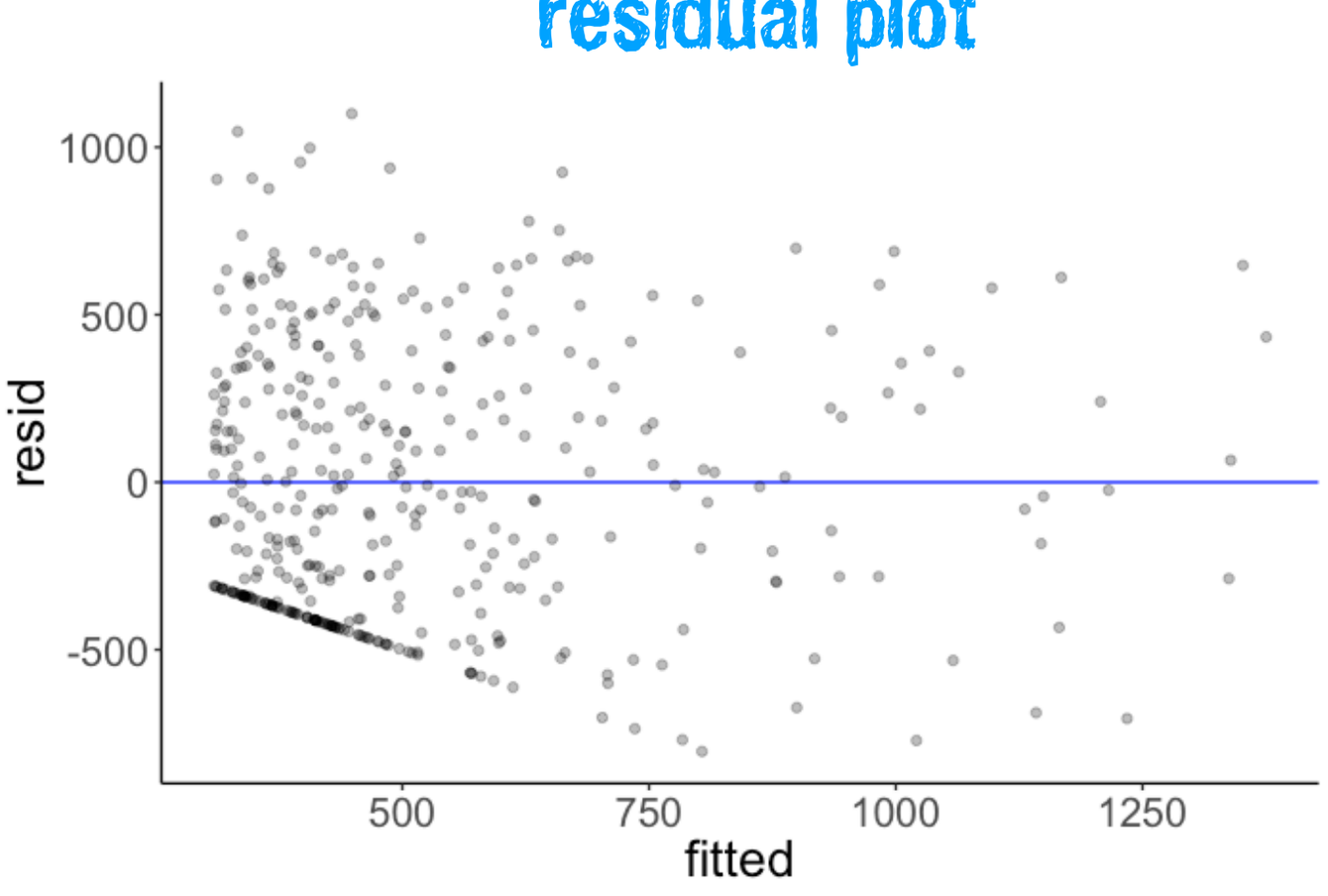
Model assumptions

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

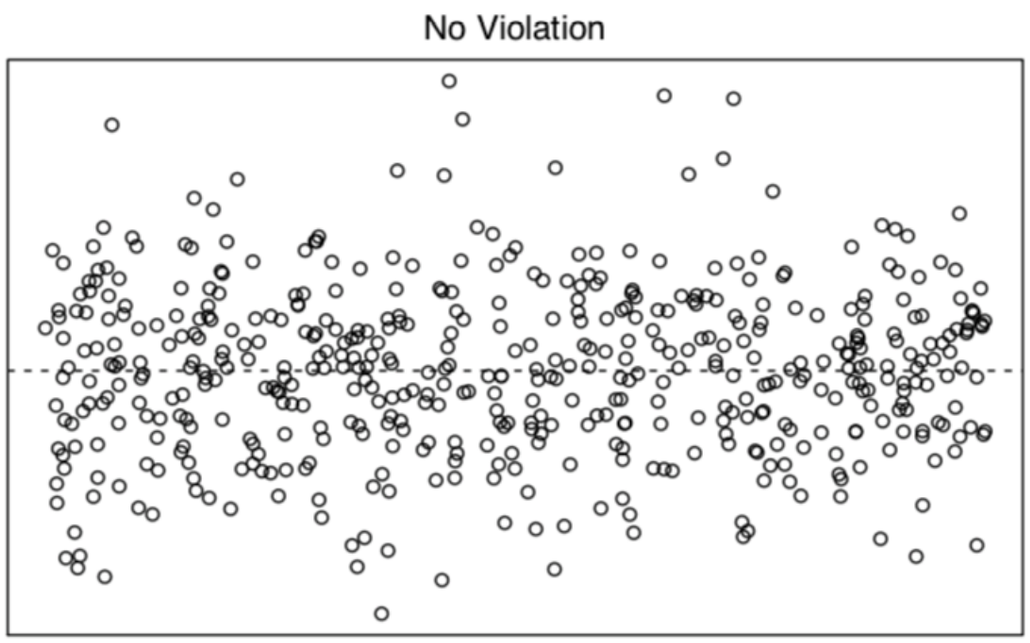


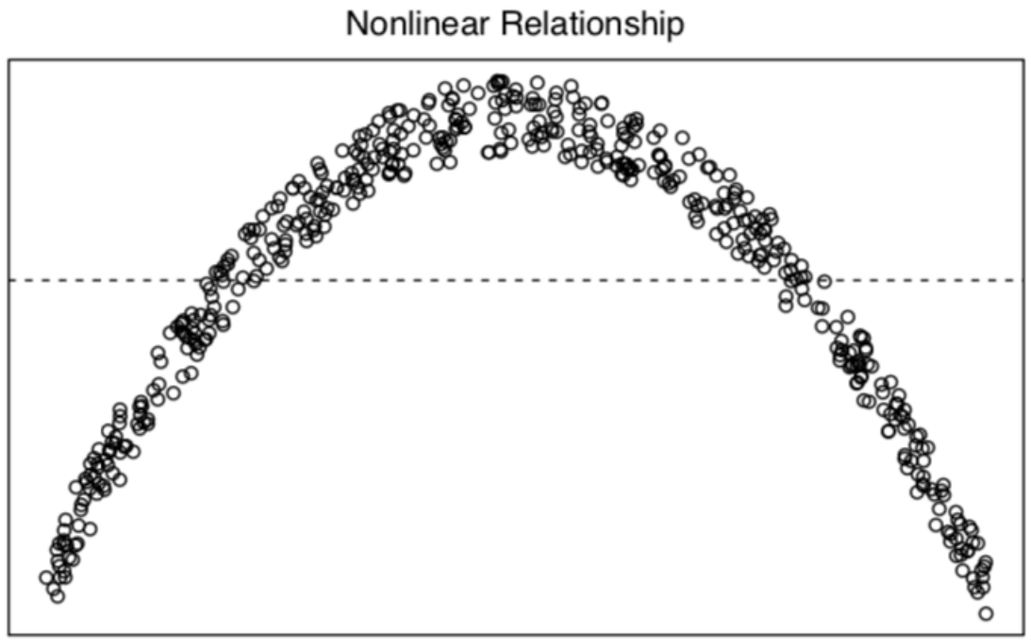
Model assumptions

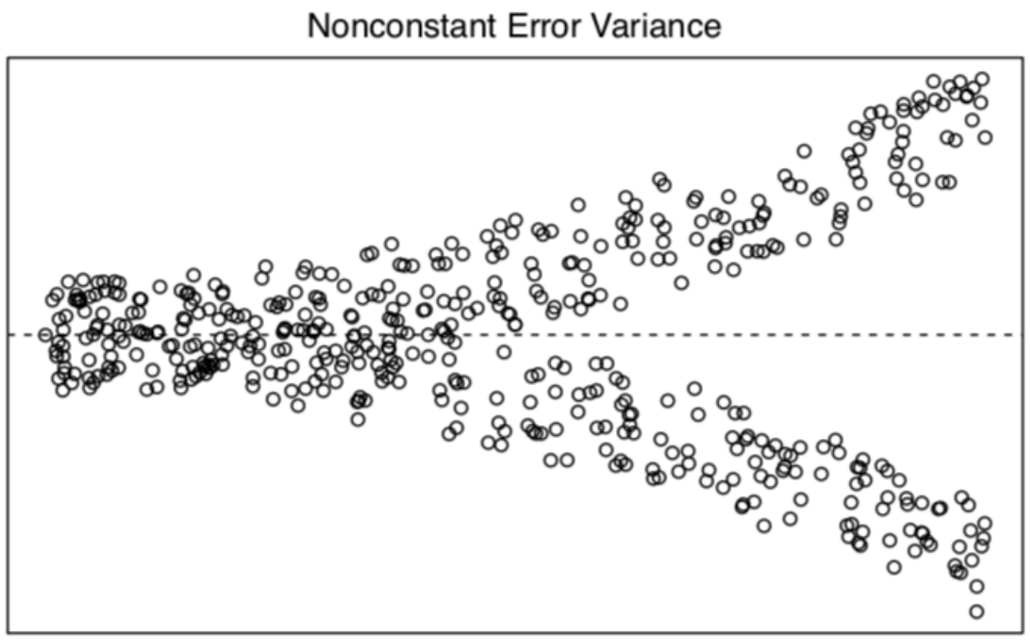


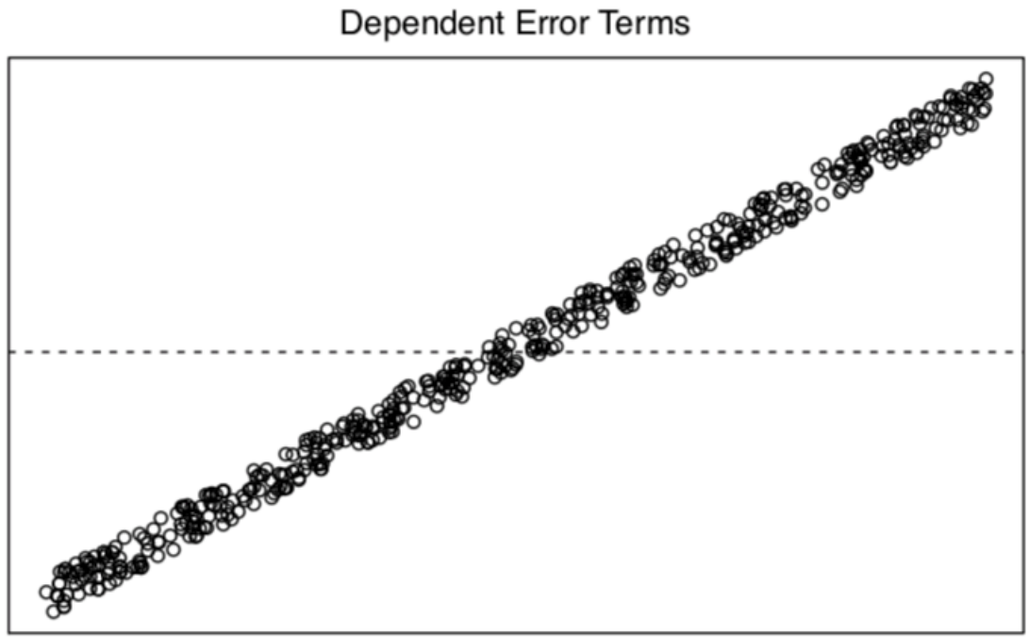


Model assumptions







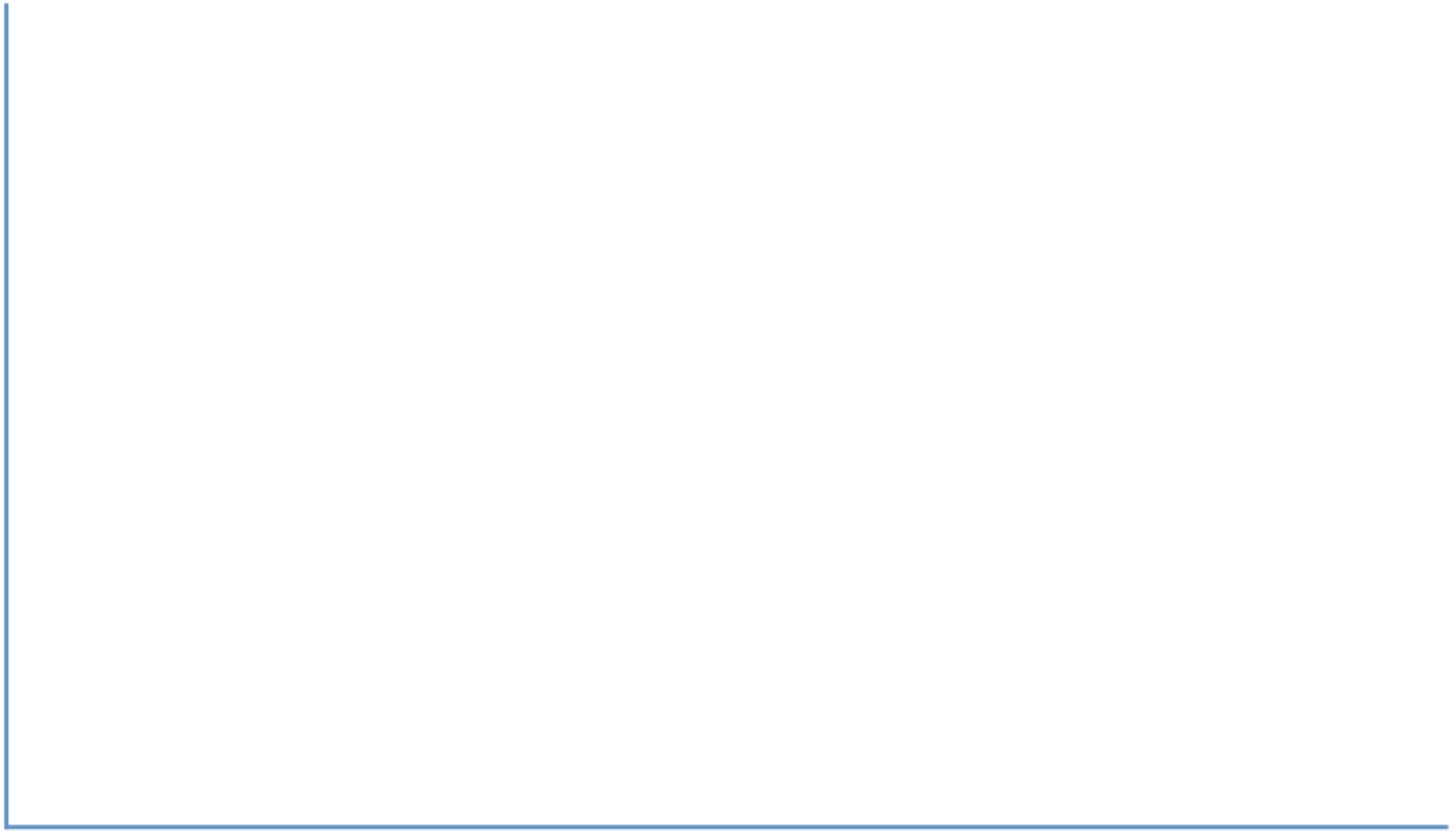


Summary

- Quick review of statistical inference in frequentist statistics
- Correlation
  - Covariation
  - Pearson's moment correlation
  - Spearman's rank correlation
- Regression
  - The conceptual tour
  - The R route

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!