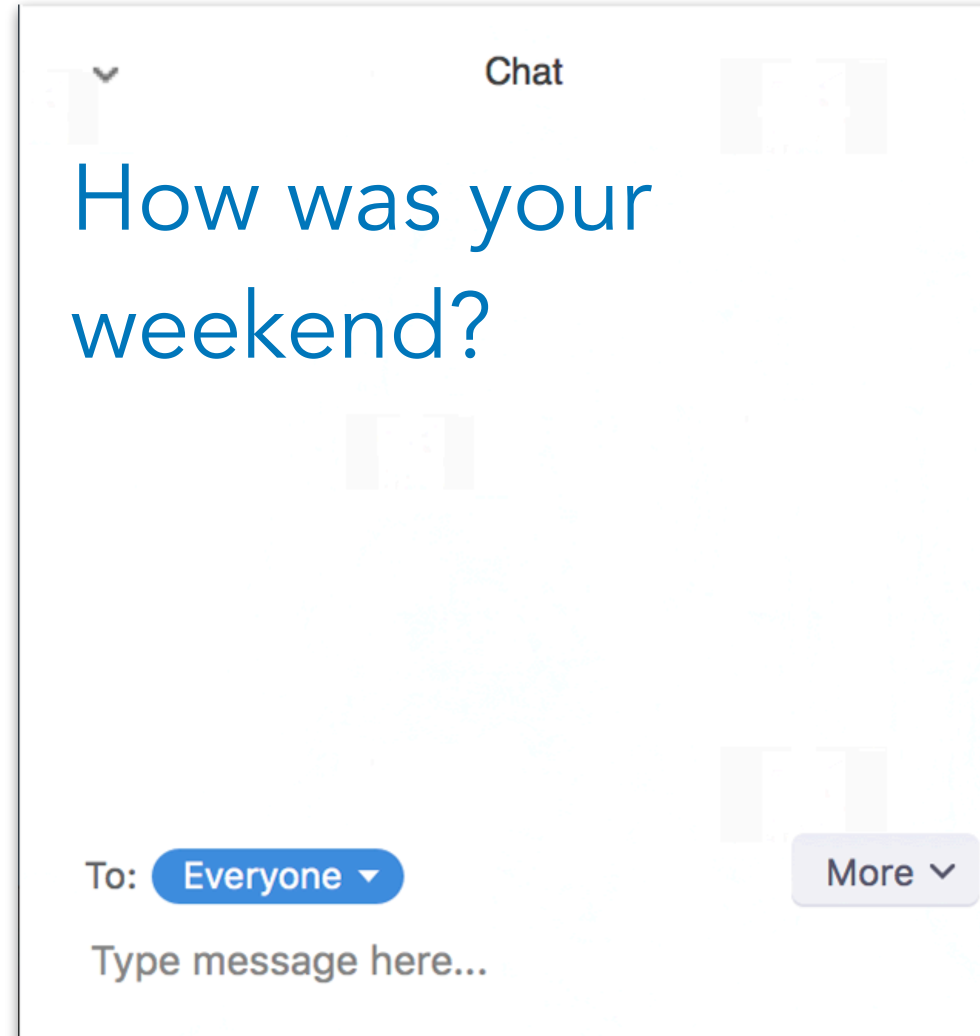
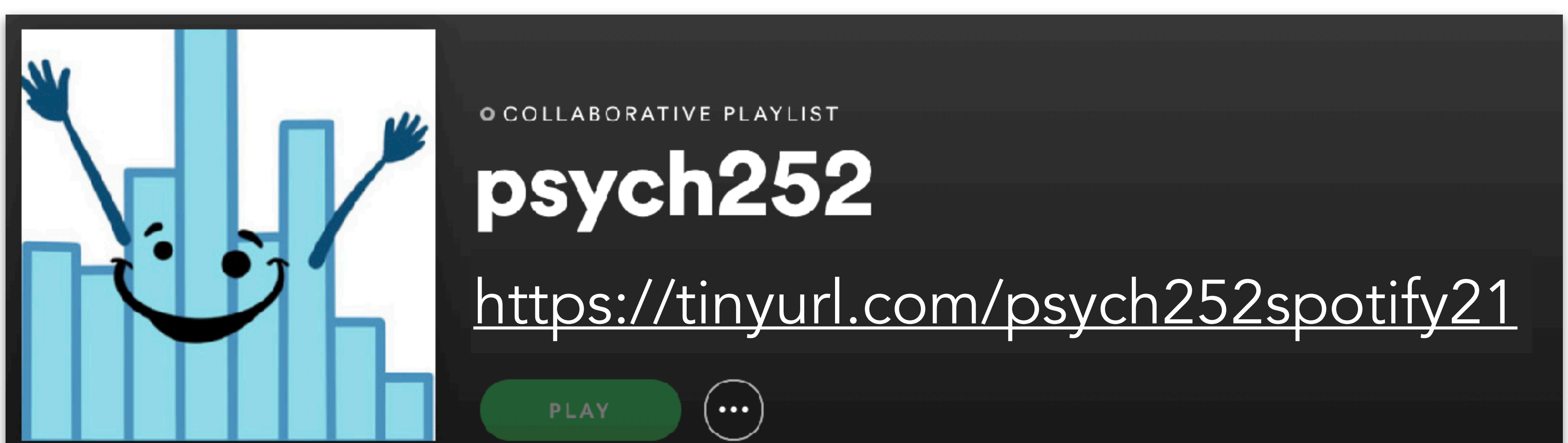
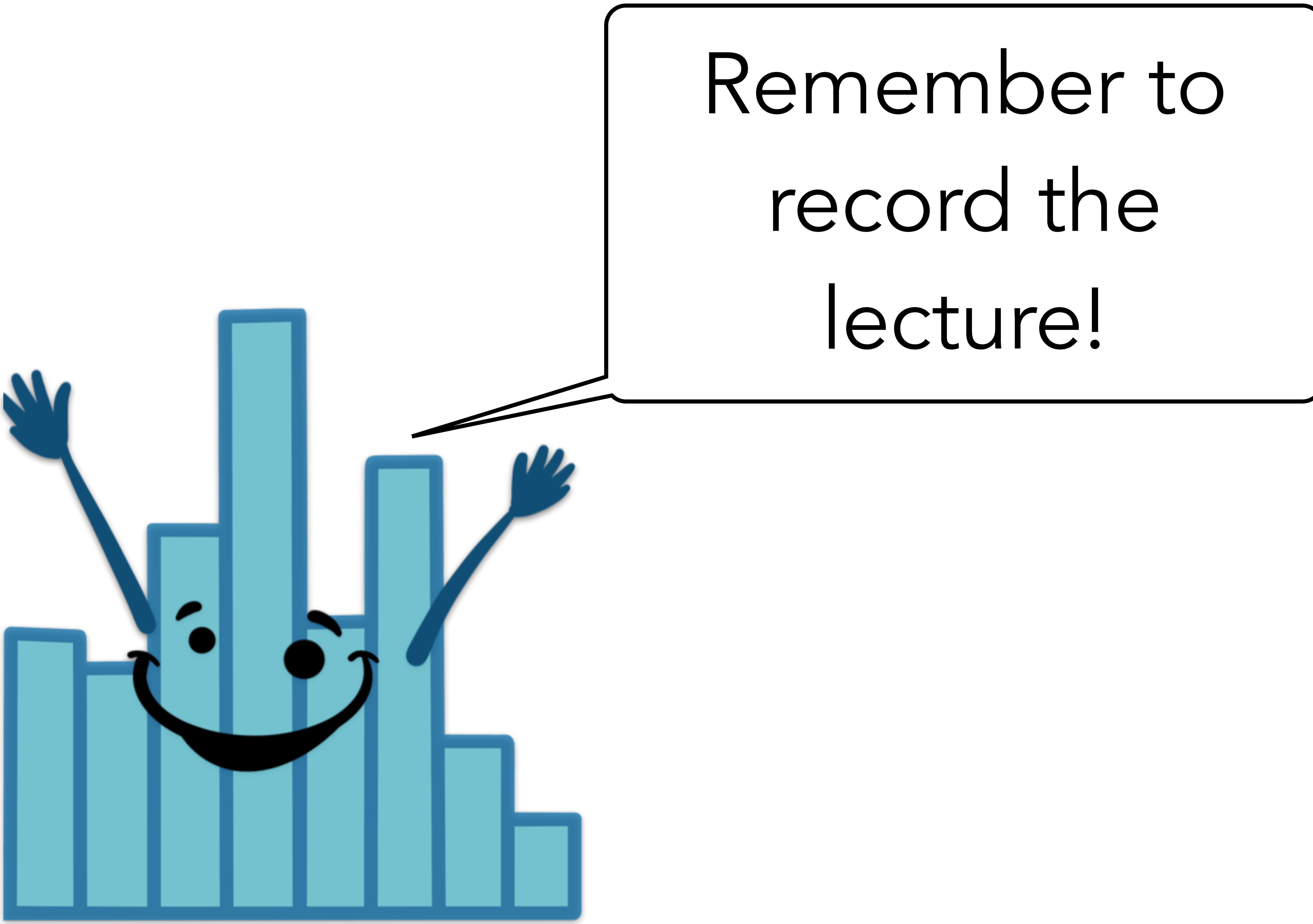
Linear model 3

If she loves you more each and every day,
by linear regression she hated you before you met.



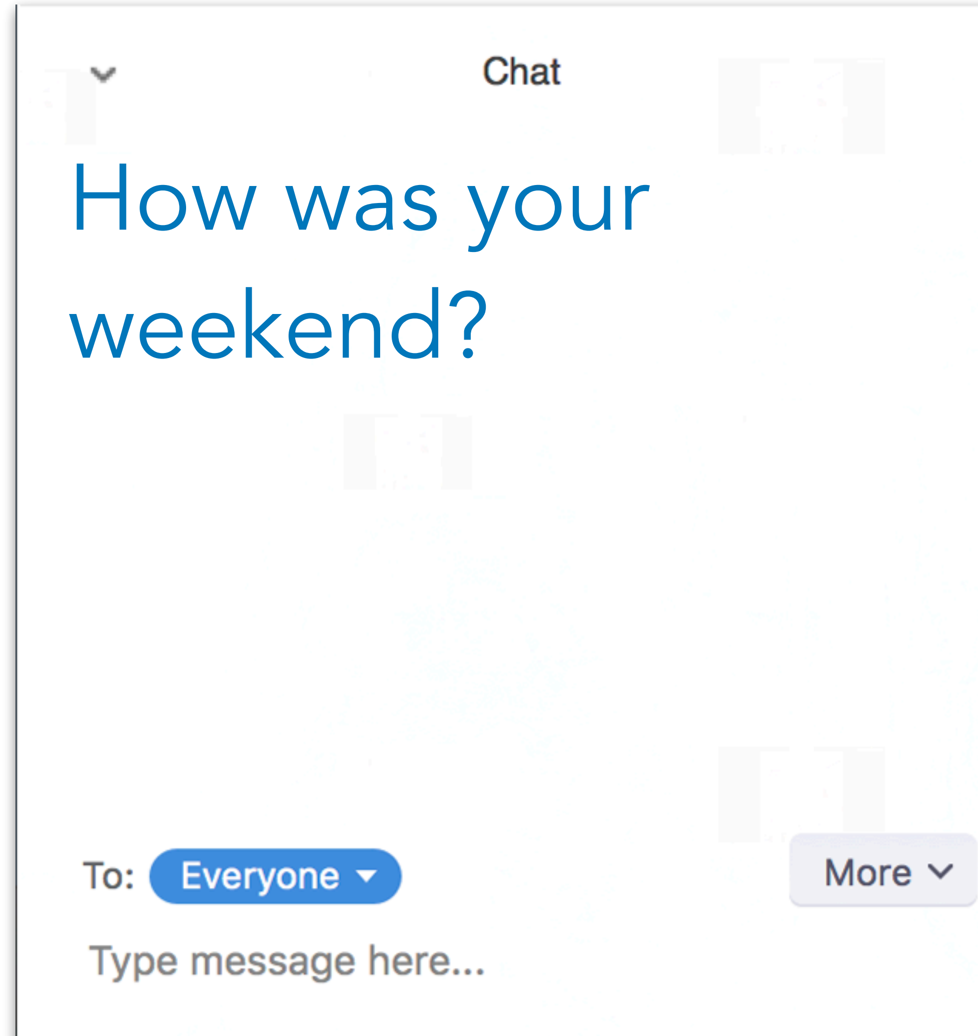


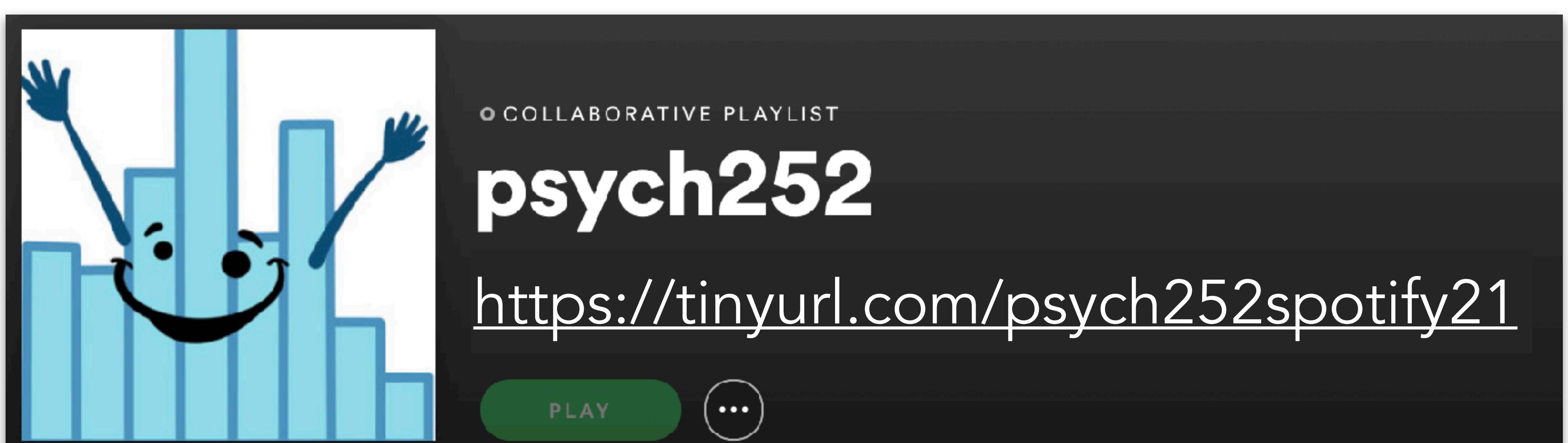
02/08/2021


Remember to record the lecture!

Linear model 3

If she loves you more each and every day, by linear regression she hated you before you met.





02/08/2021

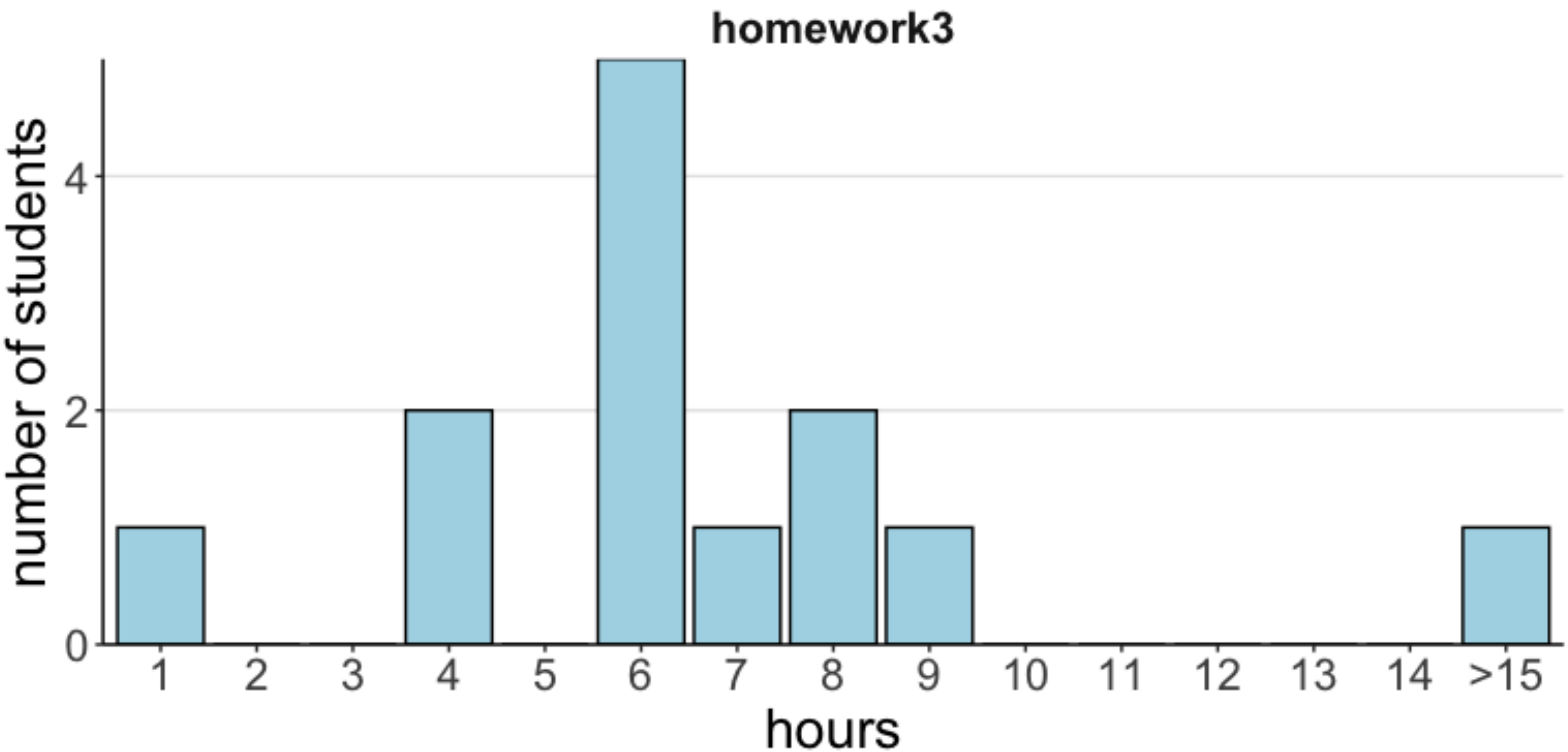
Logistics

Feedback

i liked having a short discussion about multiple regression, it helped to clarify why we do the things we do in stats and get at the bigger picture. i was wondering if it would be possible to **have a stretch break earlier in the class**? 2:10 always feels very far away

good idea! will move the stretch break forward

Homework 3


homework3
number of students
0
2
4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
>15
hours

Midterm

will be released after class on Friday

will be due on Thursday 18th, at 8pm

we won't have class on Wednesday 17th

so you have some more time to work on the midterm

just like a homework, you'll submit a knitted pdf
but: this time you'll have to work on your own

post any questions you have on Piazza and make sure your question **is only visible to the teaching team**

late policy: for each hour late until midnight, we will subtract 2% from your points (so 2% for 1h late, 4% for 2h late, ...); if you submit after midnight but before Friday 19th at 8pm, we will subtract 20% of your points; 0 points afterwards.

slip days don't apply to the midterm

`linear_model2.Rmd` notes **now updated** on canvas

Class 11

Tobias Gerstenberg

February 5th, 2021

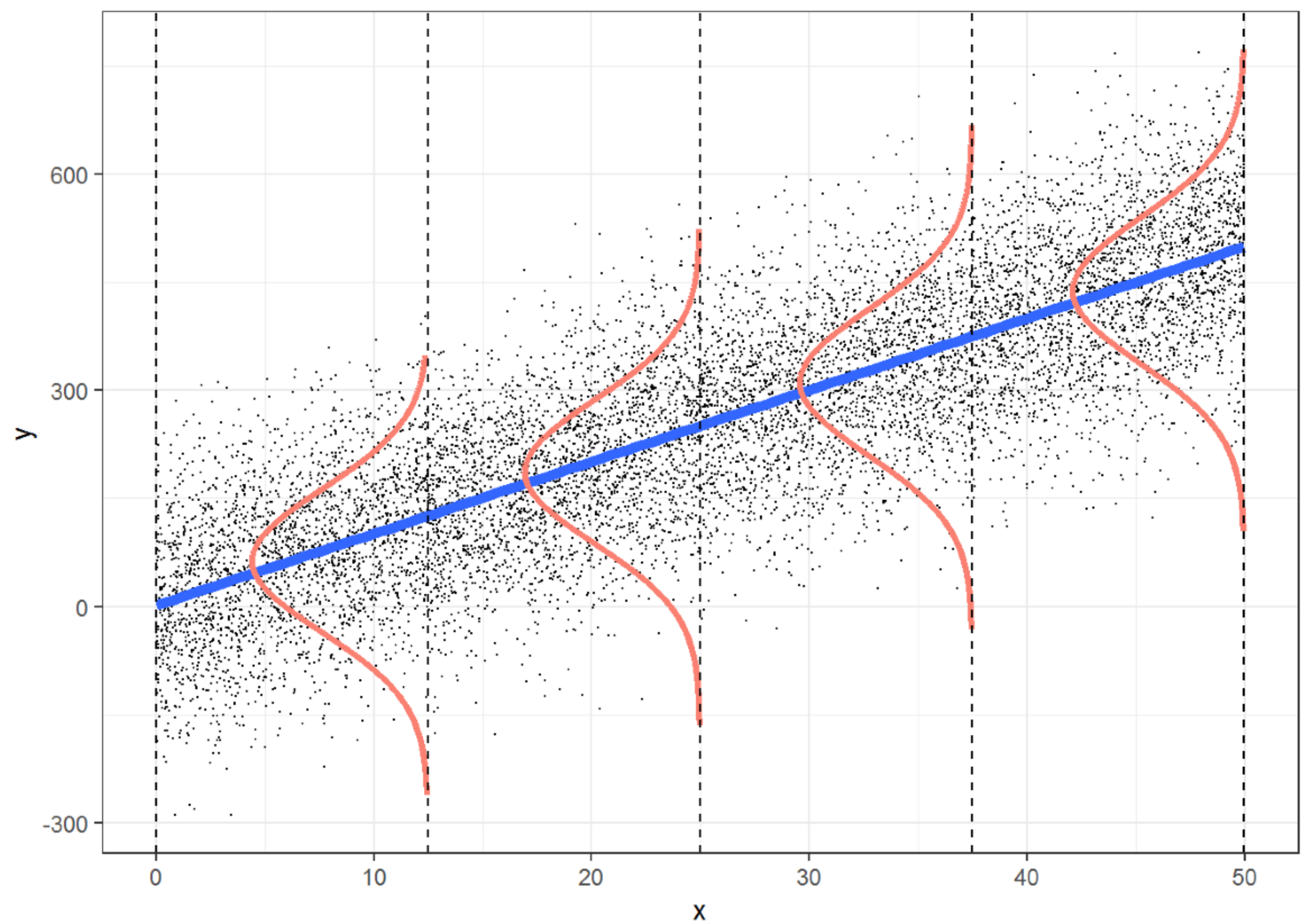
Things that came up

Normality assumption

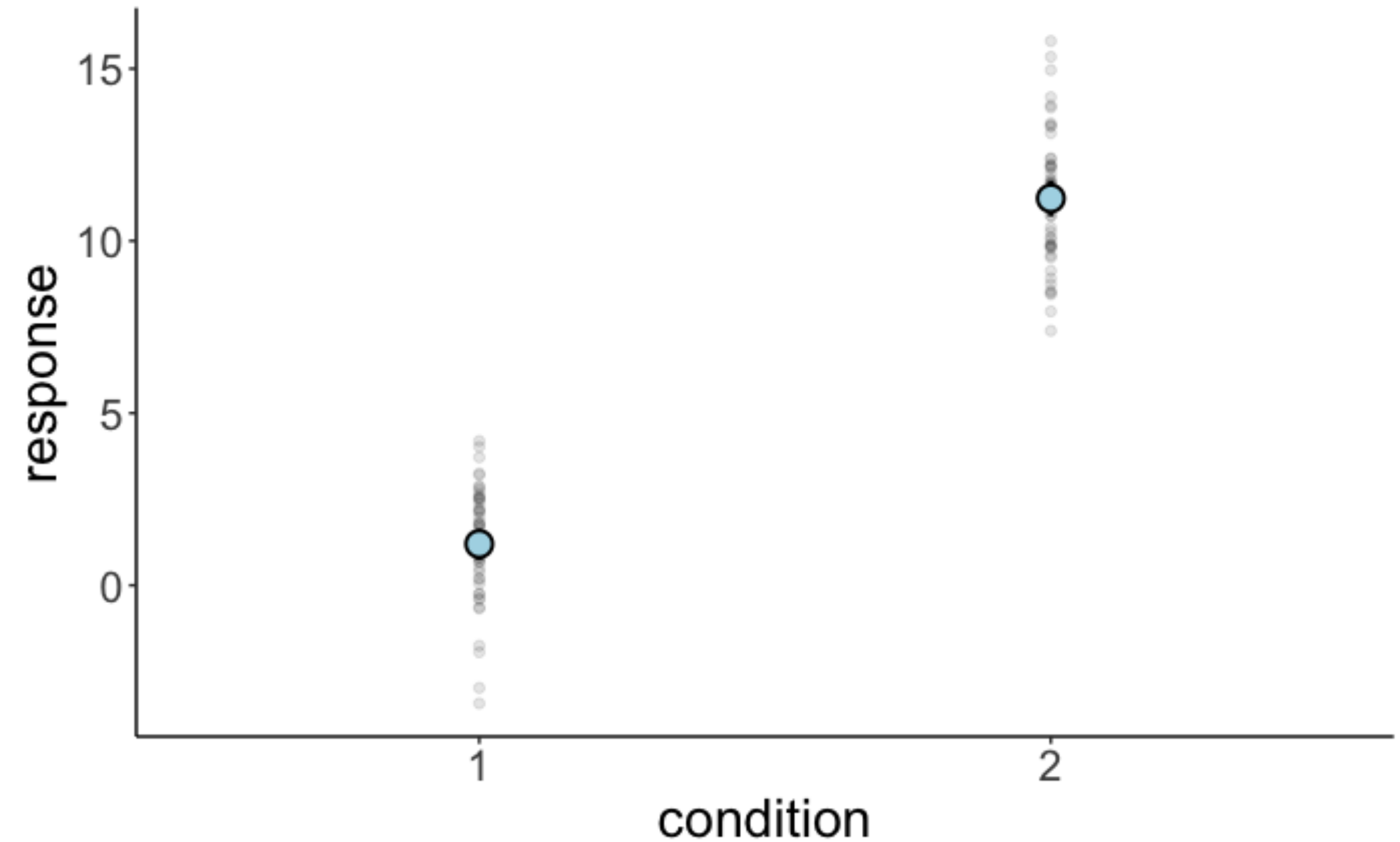
Model assumptions of simple regression

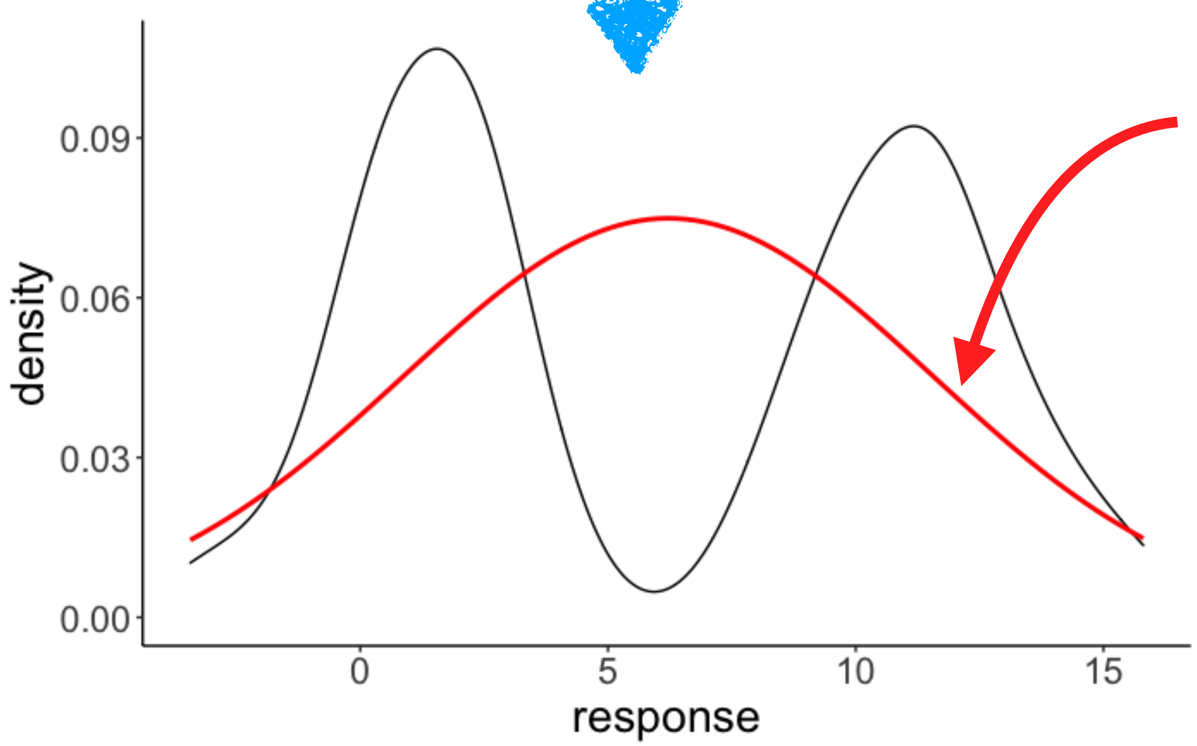
- independent observations
- Y is continuous
- errors are normally distributed
- errors have constant variance
- error terms are uncorrelated

the dependent variable doesn't need to be normally distributed!



Normality assumption





normal distribution fit to the dependent variable

```r
# test for normality of the dependent variable
shapiro.test(df.data$response)
```

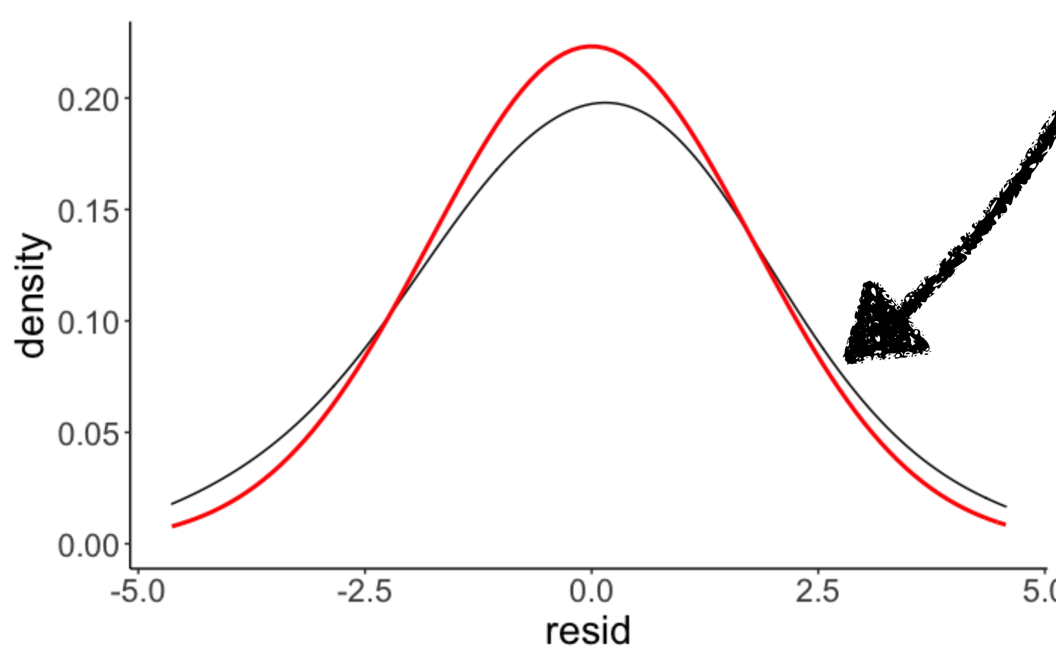
```
	Shapiro-Wilk normality test

data:  df.data$response
W = 0.90153, p-value = 1.669e-06
```

significantly different from a normal distribution

Normality assumption

```r
# fit the model to the data
fit_model = lm(formula = response ~ 1 + condition,
               data = df.data)

df.fit = fit_model %>%
  augment()
```

response	condition	.fitted	.resid	.std.resid	.hat	.sigma	.cooksd
-0.25	1	1.20	-1.45	-0.81	0.02	1.81	0.01
1.37	1	1.20	0.17	0.09	0.02	1.81	0.00
-0.67	1	1.20	-1.87	-1.05	0.02	1.80	0.01
4.19	1	1.20	2.99	1.67	0.02	1.79	0.03
1.66	1	1.20	0.46	0.26	0.02	1.81	0.00



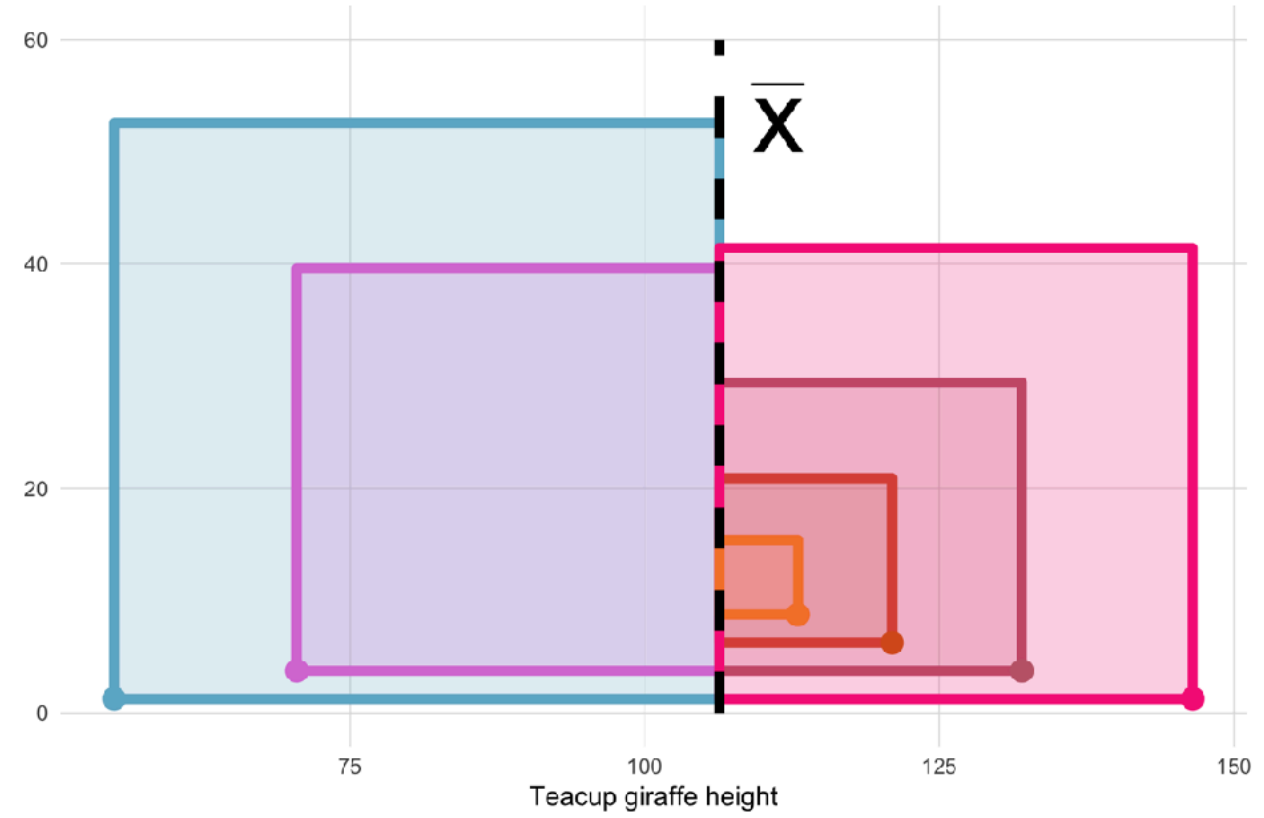
n-1 in the standard deviation

n-1 in standard deviation

$$\text{sd}(Y) = \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \overline{Y})^2}{\boxed{n-1}}}$$

definition of the standard deviation

sum of squares = squared deviation from the mean

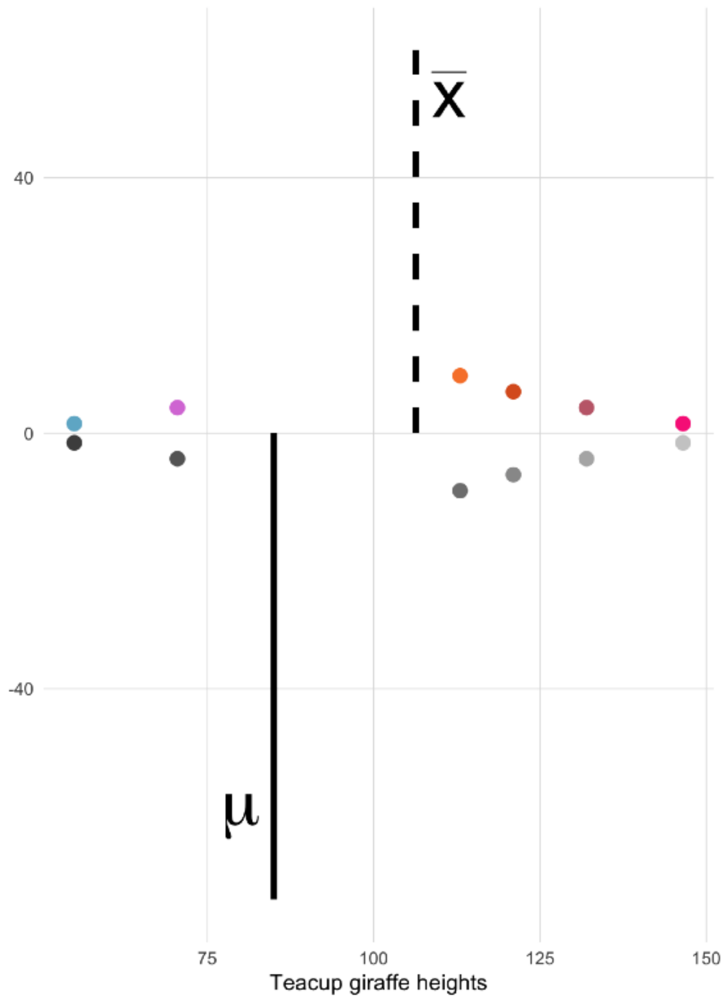


https://tinystats.github.io/teacups-giraffes-and-statistics/04_variance.html#population_vs_sample_((n)_vs_(n-1))

n-1 in standard deviation

$$\text{sd}(Y) = \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \overline{Y})^2}{\boxed{n - 1}}}$$

definition of the standard deviation



$\bar{x}$ = sample mean

- the sample mean will be somewhat different from the population mean
- this means, we are likely to underestimate the true variance

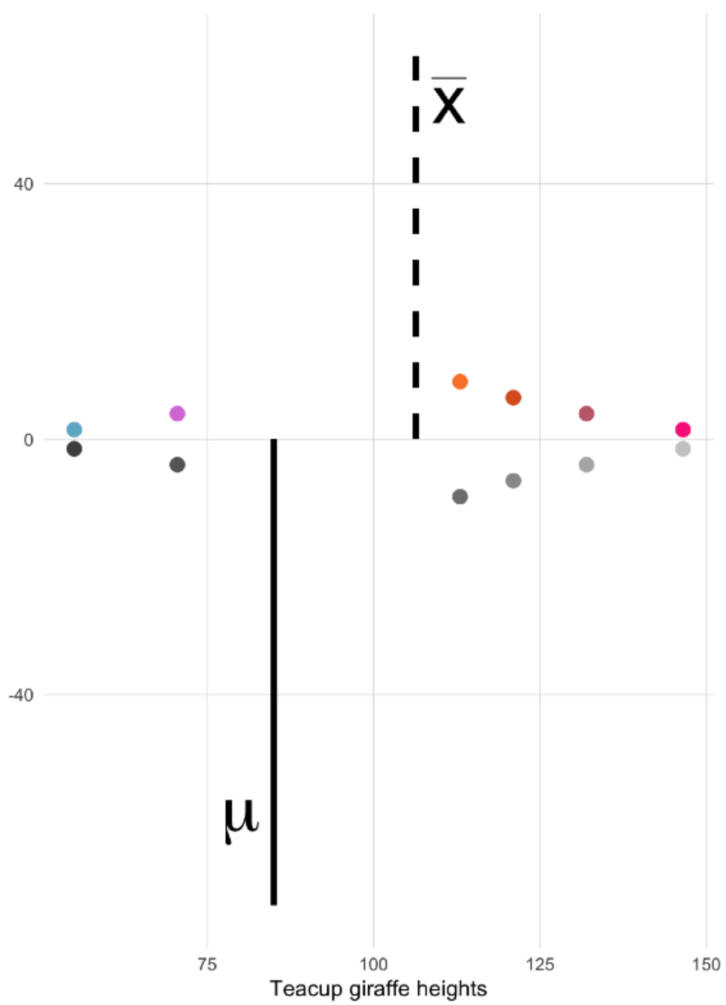
μ = population mean

https://tinystats.github.io/teacups-giraffes-and-statistics/04_variance.html#population_vs_sample_((n)_vs_(n-1))

n-1 in standard deviation

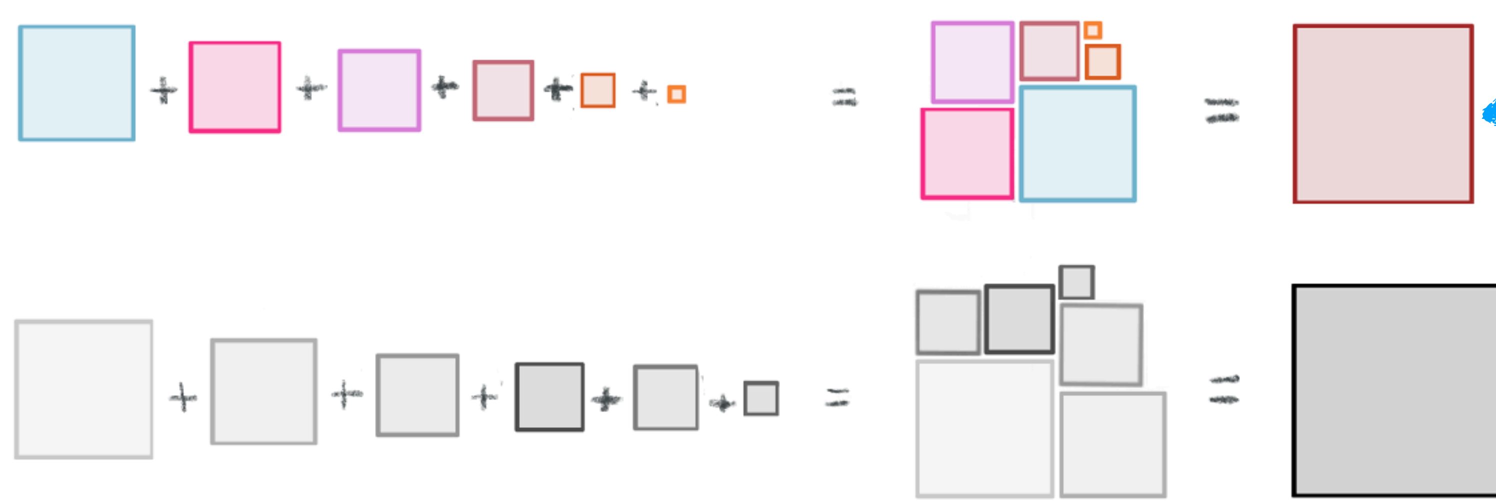
$$\mathrm{sd}(Y) = \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \overline{Y})^2}{\boxed{n-1}}}$$

definition of the standard deviation



$\bar{x}$ = sample mean

this area is smaller



μ = population mean

when *not* to bootstrap

when *not* to bootstrap

What are examples where a “naive bootstrap” fails?

Asked 9 years, 9 months ago Active 7 years, 1 month ago Viewed 15k times

94

Suppose I have a set of sample data from an unknown or complex distribution, and I want to perform some inference on a statistic T of the data. My default inclination is to just generate a bunch of bootstrap samples with replacement, and calculate my statistic T on each bootstrap sample to create an estimated distribution for T.

What are examples where this is a bad idea?

3 Answers

Active Oldest Votes

75

If the quantity of interest, usually a functional of a distribution, is reasonably smooth and your data are i.i.d., you're usually in pretty safe territory. Of course, there are other circumstances when the bootstrap will work as well.

What it means for the bootstrap to "fail"

Broadly speaking, the purpose of the bootstrap is to construct an approximate sampling distribution for the statistic of interest. It's not about actual estimation of the parameter. So, if the statistic of interest (under some rescaling and centering) is $\hat{X}_n$ and $\hat{X}_n \to X_\infty$ in distribution, we'd like our

https://stats.stackexchange.com/questions/9664/what-are-examples-where-a-naive-bootstrap-fails

when *not* to bootstrap

3 Answers

Active | Oldest | Votes

75

If the quantity of interest, usually a functional of a distribution, is reasonably smooth and your data are i.i.d., you're usually in pretty safe territory. Of course, there are other circumstances when the bootstrap will work as well.

What it means for the bootstrap to "fail"

Broadly speaking, the purpose of the bootstrap is to construct an approximate sampling distribution for the statistic of interest. It's not about actual estimation of the parameter. So, if the statistic of interest (under some rescaling and centering) is $\hat{X}_n$ and $\hat{X}_n \to X_\infty$ in distribution, we'd like our

example: estimate the 99th percentile of a distribution

In summary, the bootstrap fails (miserably) in this case. Things tend to go wrong when dealing with parameters at the edge of the parameter space.

References

Unfortunately, the subject matter is nontrivial, so none of these are particularly easy reads.

https://stats.stackexchange.com/questions/9664/what-are-examples-where-a-naive-bootstrap-fails

Plan for today

- Interactions
    - one continuous and one binary categorical variable
    - understanding the `lm()` output
- Analysis of Variance (ANOVA)
    - categorical predictor that has more than two levels (One-way ANOVA)
        - follow-up tests
    - multiple categorical predictors (N-way ANOVA)
        - interpreting parameters
        - Who is the ANOVA champ?
    - unbalanced designs

Interactions

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

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

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

Is the relationship between level of income and balance different for students than it is for non-students?

Compact Model

$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i$$

Augmented Model

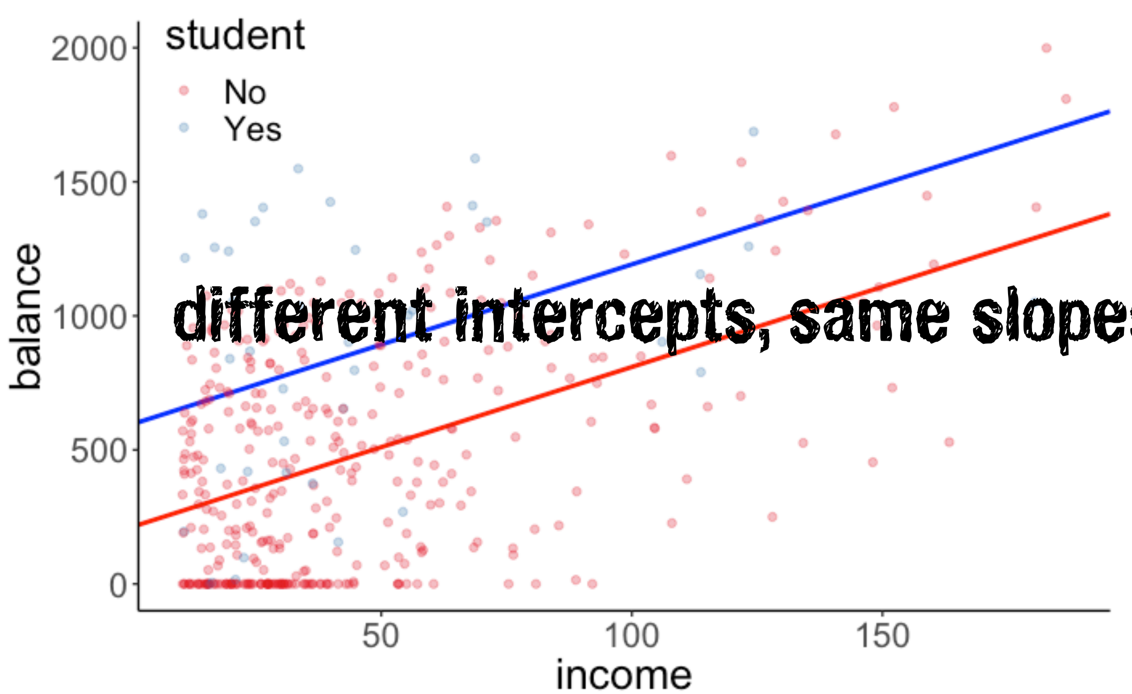
$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i + b_3 (\text{income}_i \times \text{student}_i)$$

H$_0$: The relationship between income and balance is the same for students and non-students.

Model C

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

Model prediction



Fitted model

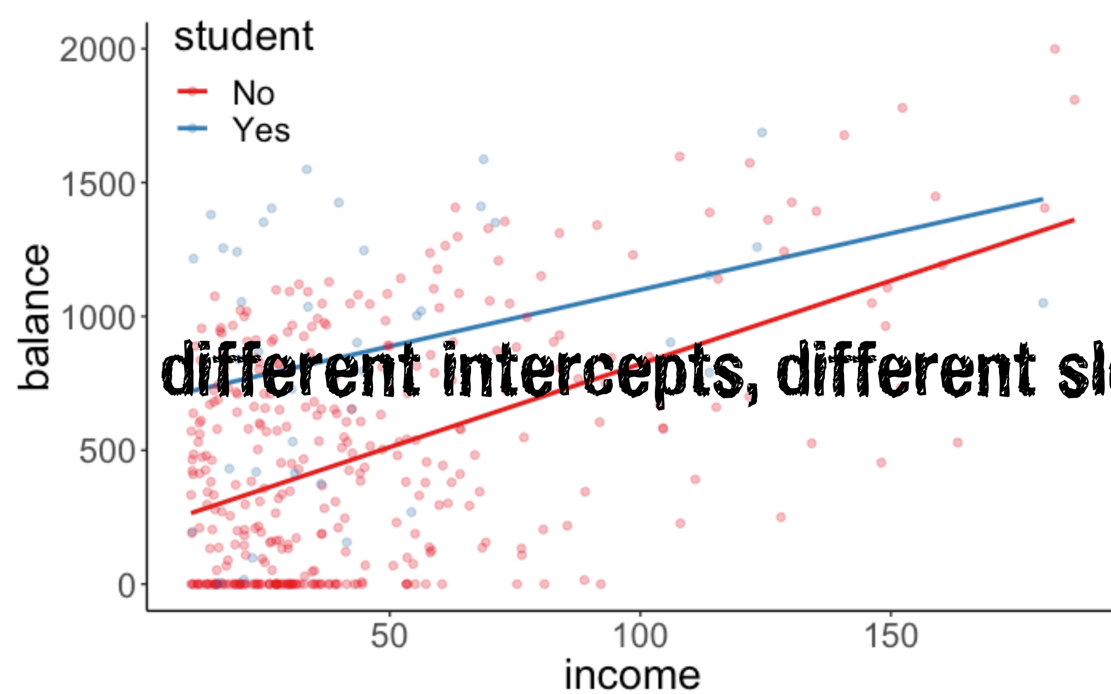
$$\widehat{\text{balance}}_i = 211.14 + 5.98 \cdot \text{income}_i + 382.67 \cdot \text{student}_i$$

H$_1$: The relationship between income and balance differs between students and non-students.

Model A

$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i + b_3(\text{income}_i \times \text{student}_i)$$

Model prediction



Fitted model

$$\widehat{\text{balance}}_i = 200.62 + 6.22 \cdot \text{income}_i + 476.68 \cdot \text{student}_i - 2.00 \cdot (\text{income}_i \times \text{student}_i)$$

Worth it?

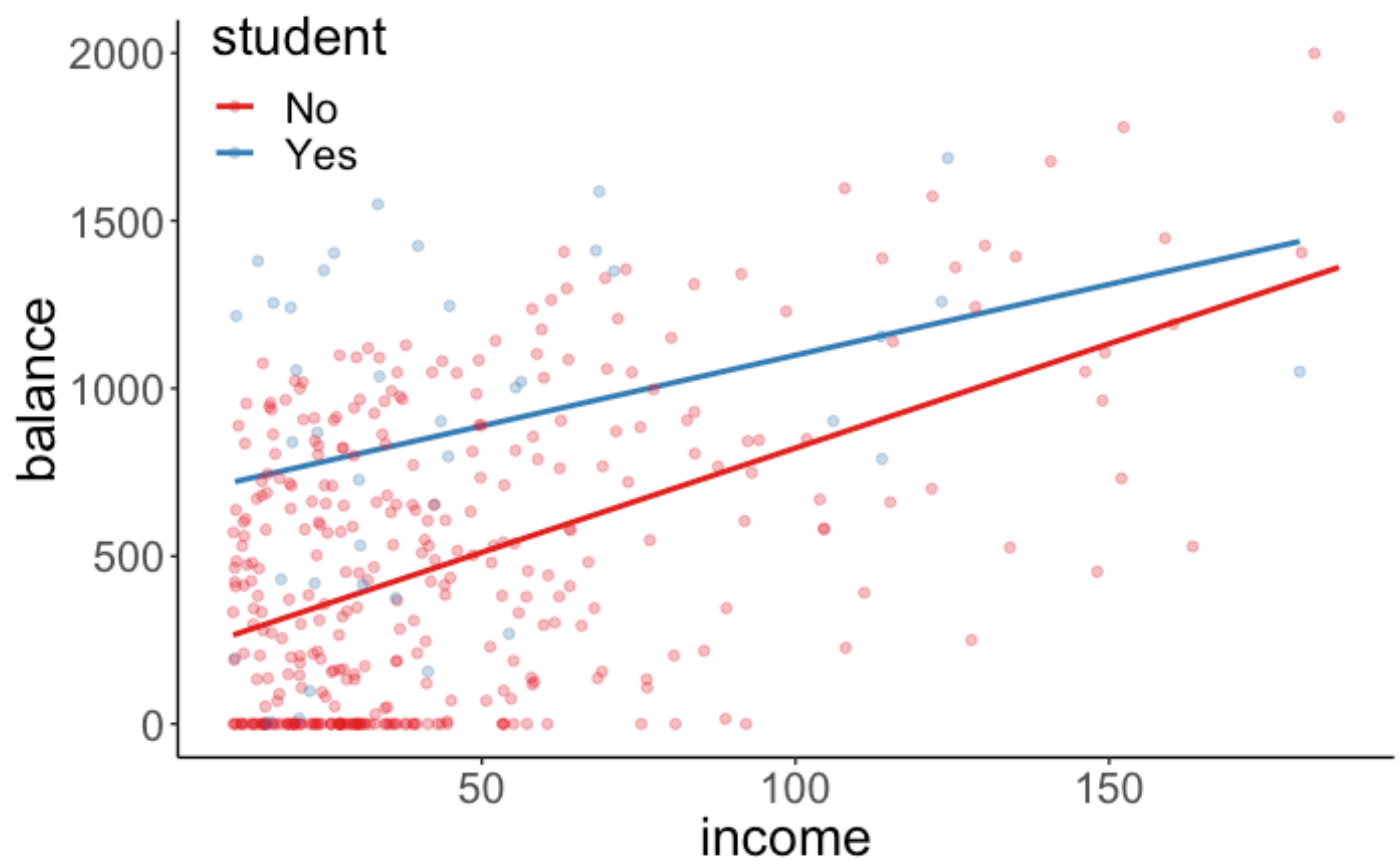
Is the relationship between level of income and balance different for students than it is for non-students?

```
# fit models
fit_c = lm(formula = balance ~ income + student, data = df.credit)
fit_a = lm(formula = balance ~ income * student, data = df.credit)

# F-test
anova(fit_c, fit_a)
```

```
Analysis of Variance Table

Model 1: balance ~ income + student
Model 2: balance ~ income * student
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1    397 60939054
2    396 60734545  1    204509 1.3334 0.2489
```

not worth it!

Interpretation



$$\widehat{\text{balance}}_i = 200.62 + 6.22 \cdot \text{income}_i + 476.68 \cdot \text{student}_i - 2.00 \cdot (\text{income}_i \times \text{student}_i)$$

if student = "No" $\widehat{\text{balance}}_i = 200.62 + 6.22 \cdot \text{income}_i$

if student = "Yes"

$$\begin{aligned}\widehat{\text{balance}}_i &= 200.62 + 6.22 \cdot \text{income}_i + 476.68 \cdot 1 - 2.00 \cdot (\text{income}_i \times 1) \\ &= 677.3 + 6.22 \cdot \text{income}_i - 2.00 \cdot \text{income}_i \\ &= 677.3 + 4.22 \cdot \text{income}_i\end{aligned}$$

Interpretation

```
fit1 = lm(formula = balance ~ income + student + income:student, data = df.credit)
```

Explicitly encode the interaction

```
df.credit %>%
  mutate(student_dummy = ifelse(student == "No", 0, 1)) %>%
  mutate(income_student = income * student_dummy) %>%
  select(balance, income, student, student_dummy, income_student)
```

balance	income	student	student_dummy	income_student
333	14.89	No	0	0.00
903	106.03	Yes	1	106.03
580	104.59	No	0	0.00
964	148.92	No	0	0.00
331	55.88	No	0	0.00
1151	80.18	No	0	0.00
203	21.00	No	0	0.00
872	71.41	No	0	0.00
279	15.12	No	0	0.00
1350	71.06	Yes	1	71.06

```
fit2 = lm(formula = balance ~ income + student + income_student, data = df.credit)
```

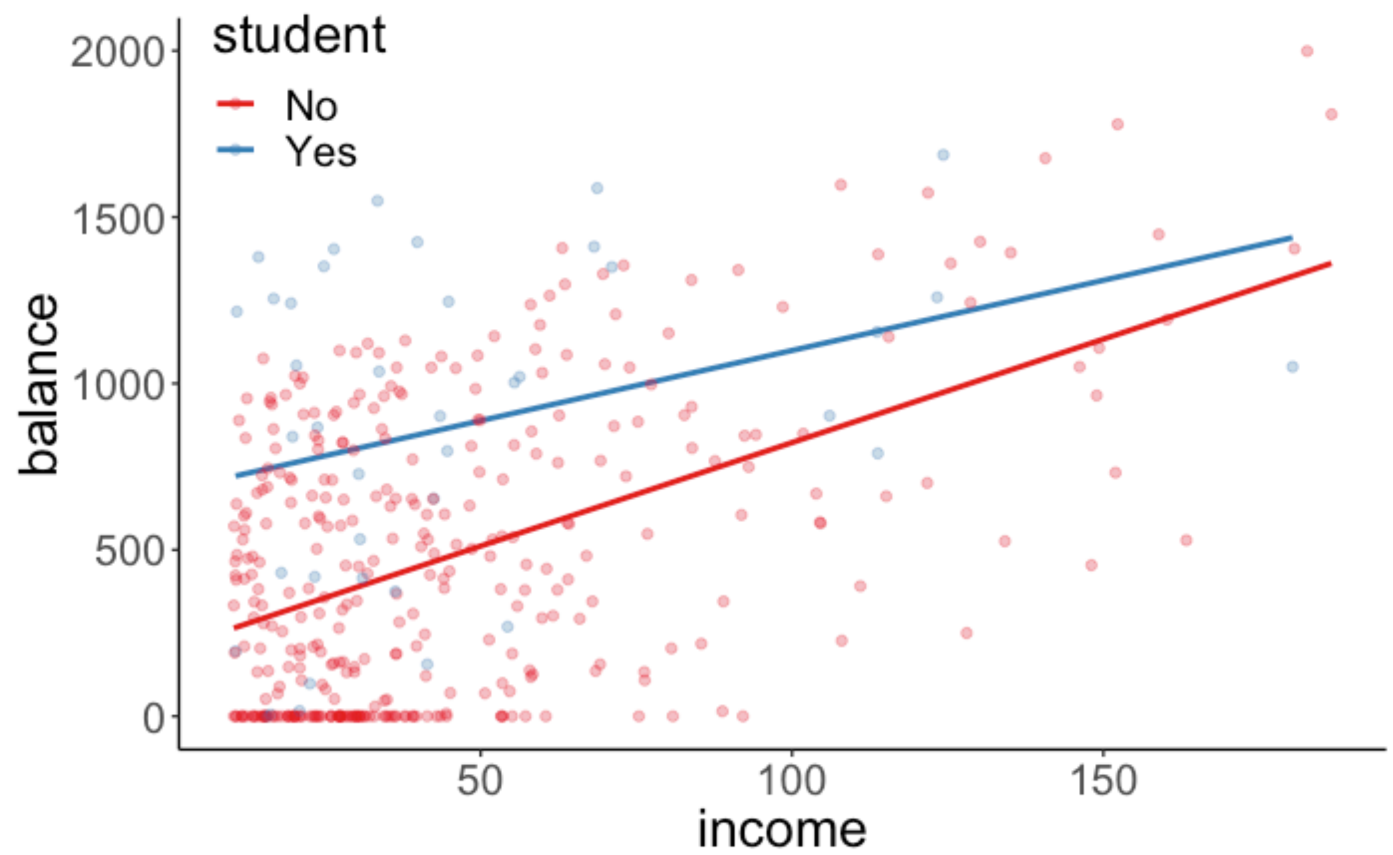
`fit1` and `fit2` are identical!

How to report results of interaction

There is no significant difference in the relationship between income and balance for students versus non-students, $F(1, 396) = 1.33$, $p = 0.25$.

For *students*, an increase in $1000 income is associated with an increase in $4.21 of average credit card balance.

For *non-students*, an increase in $1000 income is associated with an increase in $6.22 of average credit card balance.



`lm()` output

lm() output

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

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

Residuals:
    Min      1Q  Median      3Q     Max
-773.39 -325.70  -41.13  321.65  814.04

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       200.6232    33.6984   5.953 5.79e-09 ***
income              6.2182     0.5921  10.502  < 2e-16 ***
studentYes        476.6758   104.3512   4.568 6.59e-06 ***
income:studentYes  -1.9992     1.7313  -1.155    0.249
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Residual standard error: 391.6 on 396 degrees of freedom
Multiple R-squared:  0.2799, Adjusted R-squared:  0.2744
F-statistic:  51.3 on 3 and 396 DF,  p-value: < 2.2e-16
```

?

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

anova(fit_c, fit_a)
```

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

anova(fit_c, fit_a)
```

lm() output

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

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

Residuals:
    Min      1Q  Median      3Q     Max
-773.39 -325.70  -41.13  321.65  814.04

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       200.6232    33.6984   5.953 5.79e-09 ***
income              6.2182     0.5921  10.502  < 2e-16 ***
studentYes        476.6758   104.3512   4.568 6.59e-06 ***
income:studentYes  -1.9992     1.7313  -1.155    0.249
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Residual standard error: 391.6 on 396 degrees of freedom
Multiple R-squared:  0.2799, Adjusted R-squared:  0.2744
F-statistic:  51.3 on 3 and 396 DF,  p-value: < 2.2e-16
```

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

anova(fit_c, fit_a)
```

`lm()` output

```r
lm(formula = balance ~ income + student + income:student, data = df.credit) %>%
  summary()
```

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

Residuals:
    Min      1Q  Median      3Q     Max
-773.39 -325.70  -41.13  321.65  814.04

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        200.6232    33.6984   5.953 5.79e-09 ***
income               6.2182     0.5921  10.502  < 2e-16 ***
studentYes         476.6758   104.3512   4.568 6.59e-06 ***
income:studentYes   -1.9992     1.7313  -1.155    0.249
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 391.6 on 396 degrees of freedom
Multiple R-squared:  0.2799,	Adjusted R-squared:  0.2744
F-statistic:  51.3 on 3 and 396 DF,  p-value: < 2.2e-16
```

- runs many hypothesis tests at the same time (many of which we may not be interested in)
- increases the danger of making a type-I error (incorrectly rejecting the H_0)
- will not give us p-values for mixed effects models ...

The model comparison approach

- allows to formulate hypotheses as specific comparisons between candidate models
- is more flexible: we could test a model with 2 predictors vs. one with 4 predictors
- gives us insight into the underlying statistical procedure

```
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
```

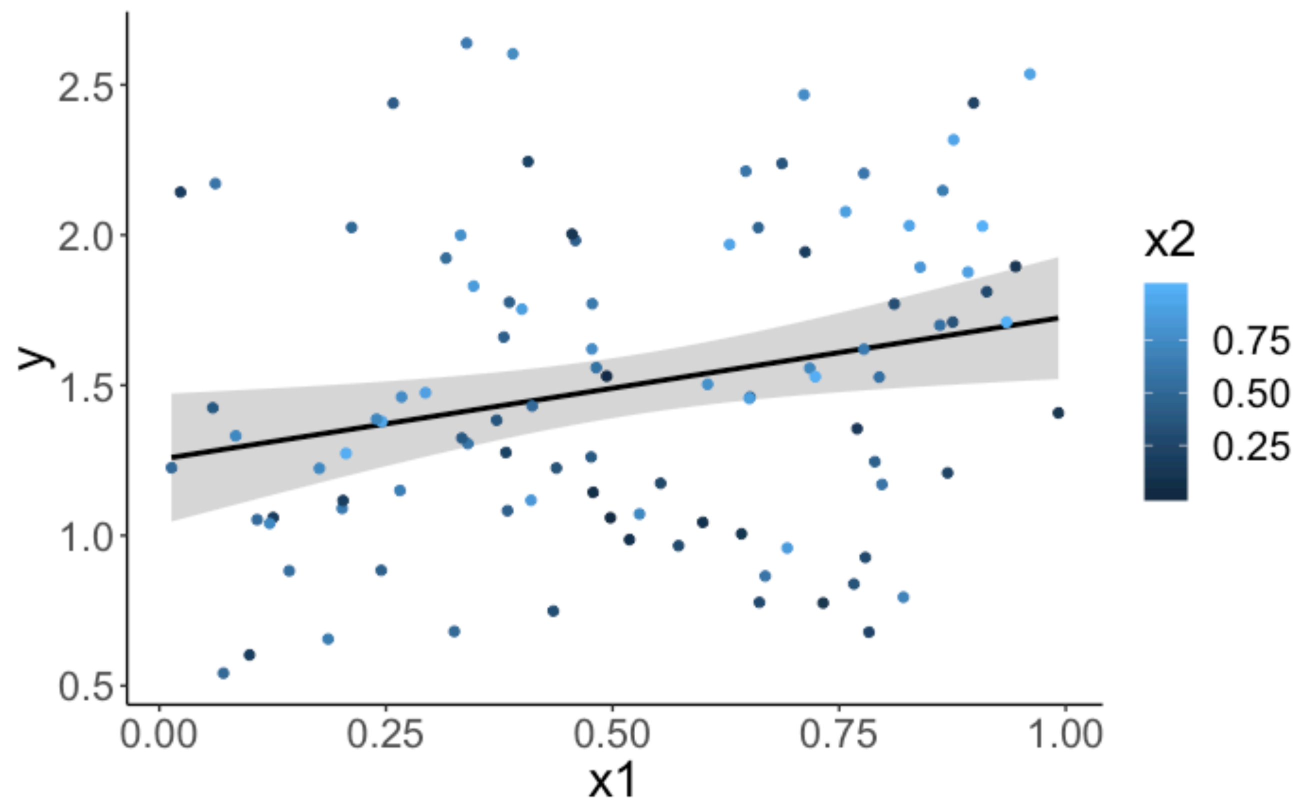
`anova()` gives me *F*s

but `lm()` gives me *t*s ?

```
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
```

F vs. t in `lm()` output

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

# parameters
sample_size = 100
b0 = 1
b1 = 0.5
b2 = 0.5
sd = 0.5

# sample
df.data = tibble(
  participant = 1:sample_size,
  x1 = runif(sample_size, min = 0, max = 1),
  x2 = runif(sample_size, min = 0, max = 1),
  # simple additive model
  y = b0 + b1 * x1 + b2 * x2 + rnorm(sample_size, sd = sd)
)
```



$$Y_i = b_0 + b_1 \cdot x1_i + b_2 \cdot x2_i + e_i$$

F vs. t in `lm()` output

```
Call:
lm(formula = y ~ x1 + x2, data = df.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.9290 -0.3084 -0.0716  0.2676  1.1659 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.9953     0.1395   7.133 1.77e-10 ***
x1            0.4654     0.1817   2.561  0.01198 *  
x2            0.5072     0.1789   2.835  0.00558 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4838 on 97 degrees of freedom
Multiple R-squared:  0.1327,	Adjusted R-squared:  0.1149 
F-statistic: 7.424 on 2 and 97 DF,  p-value: 0.001
```

Is x1 a significant predictor, controlling for x2?

F vs. t in `lm()` output

without x1

```r
# fit models
model_compact = lm(formula = y ~ 1 + x2,
                   data = df.data)
```

with x1

```r
model_augmented = lm(formula = y ~ 1 + x1 + x2,
                     data = df.data)

# compare models using the F-test
anova(model_compact, model_augmented)
```

```
Analysis of Variance Table

Model 1: y ~ 1 + x2
Model 2: y ~ 1 + x1 + x2
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     98 24.235
2     97 22.700  1    1.5347 6.558 0.01198 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

F vs. t in `lm()` output

```
Call:
lm(formula = y ~ x1 + x2, data = df.data)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9290 -0.3084 -0.0716  0.2676  1.1659

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.9953     0.1395   7.133 1.77e-10 ***
x1            0.4654     0.1817   2.561  0.01198 *
x2            0.5072     0.1789   2.835  0.00558 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4838 on 97 degrees of freedom
Multiple R-squared:  0.1327,  Adjusted R-squared:  0.1149
F-statistic: 7.424 on 2 and 97 DF,  p-value: 0.001
```

deterministic mapping between t and F

$$t^2 = F$$

$$2.561^2 = 6.558$$

```
Analysis of Variance Table

Model 1: y ~ 1 + x2
Model 2: y ~ 1 + x1 + x2
  Res.Df    RSS Df Sum of Sq     F  Pr(>F)
1     98 24.235
2     97 22.700  1    1.5347 6.558 0.01198 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Breakout rooms

Task:

- What's an ANOVA again?

Discuss with your group.

Size: ~3 people

Time: 4 minutes

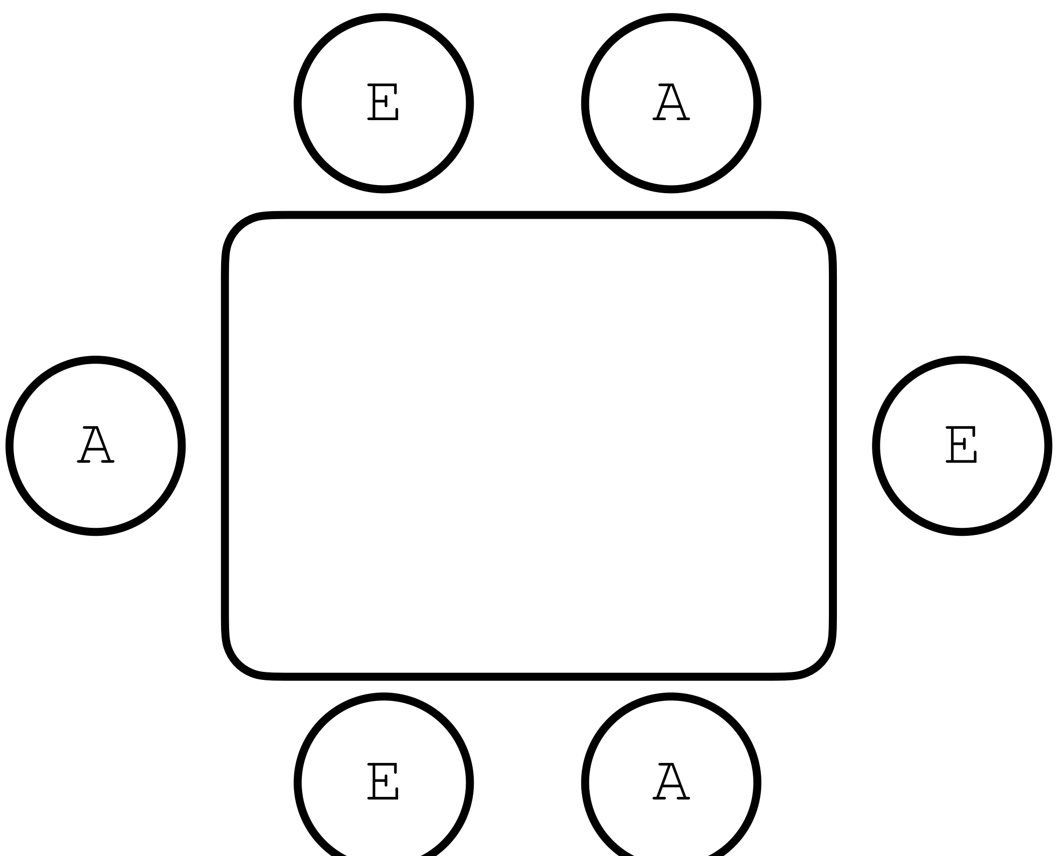
Report: I'll ask a few of you to share with the larger group.

Categorical predictor with more than two levels

What's the role of skill vs. chance in poker?

Abstract

Adopting a quasi-experimental approach, the present study examined the extent to which the influence of poker playing skill was more important than card distribution. Three average players and three experts sat down at a six-player table and played **60 computer-based** hands of the poker variant ''Texas Hold'em'' for money. In each hand, one of the average players and one expert received (a) better-than-average cards (winner's box), (b) average cards (neutral box) and (c) worse-than-average cards (loser's box). The standardized manipulation of the card distribution controlled the factor of chance to determine differences in performance between the average and expert groups. Overall, 150 individuals participated in a ''fixed- limit'' game variant, and 150 individuals participated in a ''no-limit'' game variant.



During the game, one expert player and one average player received

(a) the winning hand 15 times and the losing hand 5 times (winner's box condition)

(b) the winning hand 10 times and the losing hand 10 times (neutral box condition)

(c) the winning hand 5 times and the losing hand 15 times (loser's box condition)

Data set for today

participant	skill	hand	limit	balance
1	expert	bad	fixed	4.00
2	expert	bad	fixed	5.55
26	expert	bad	none	5.52
27	expert	bad	none	8.28
51	expert	neutral	fixed	11.74
52	expert	neutral	fixed	10.04
76	expert	neutral	none	21.55
77	expert	neutral	none	3.12
101	expert	good	fixed	10.86
102	expert	good	fixed	8.68

skill = expert/average

hand = bad/neutral/good

limit = fixed/none

balance = final balance in Euros

2 (skill) x 3 (hand) x 2 (limit) design

25 participants per condition

n = 300

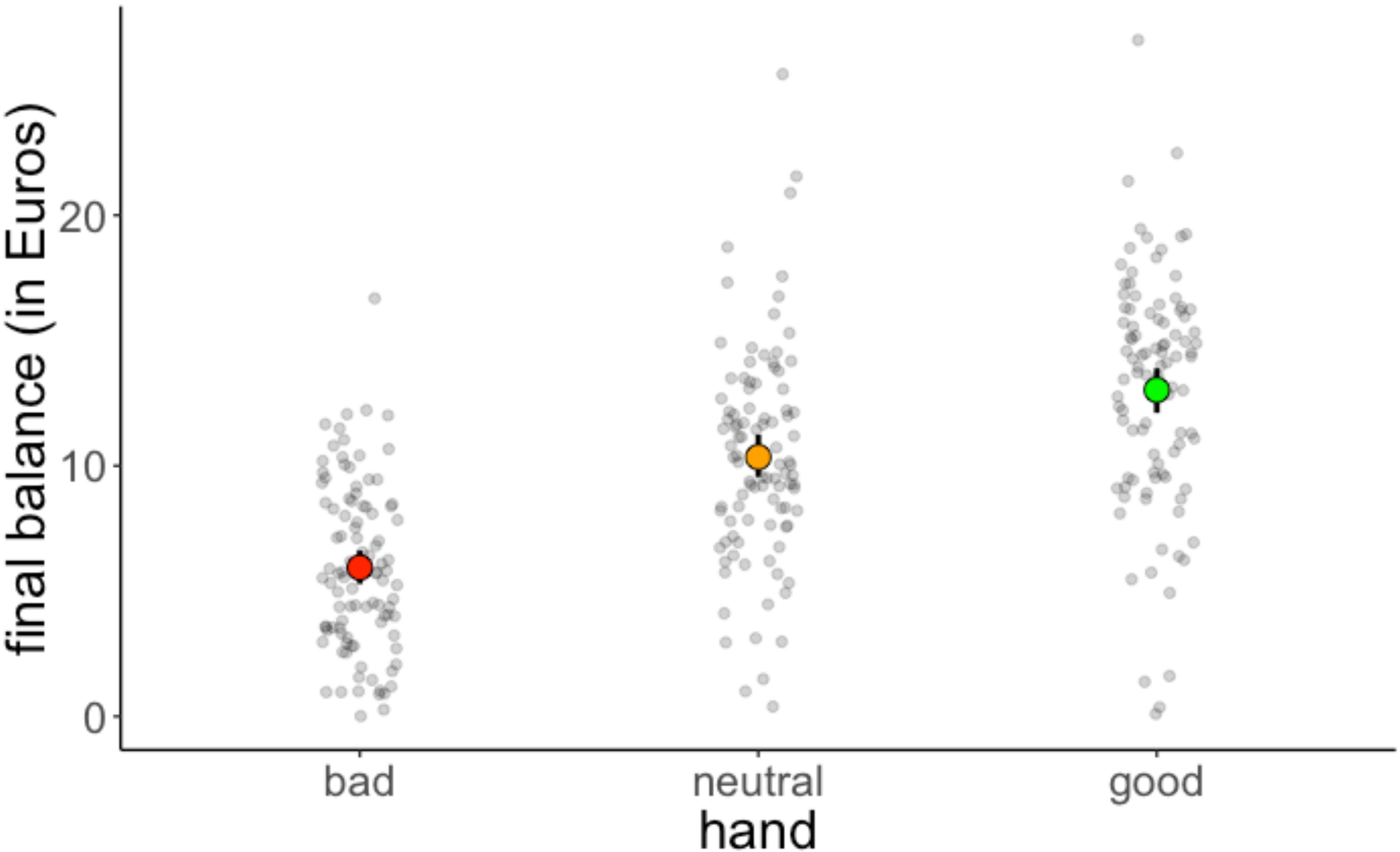
Meyer, G., von Meduna, M., Brosowski, T., & Hayer, T. (2012). Is poker a game of skill or chance? A quasi-experimental study. *Journal of Gambling Studies*

Do better hands win more money?

participant	skill	hand	limit	balance
1	expert	bad	fixed	4.00
2	expert	bad	fixed	5.55
26	expert	bad	none	5.52
27	expert	bad	none	8.28
51	expert	neutral	fixed	11.74
52	expert	neutral	fixed	10.04
76	expert	neutral	none	21.55
77	expert	neutral	none	3.12
101	expert	good	fixed	10.86
102	expert	good	fixed	8.68

hand = {bad, neutral, good}

Visualize the data first

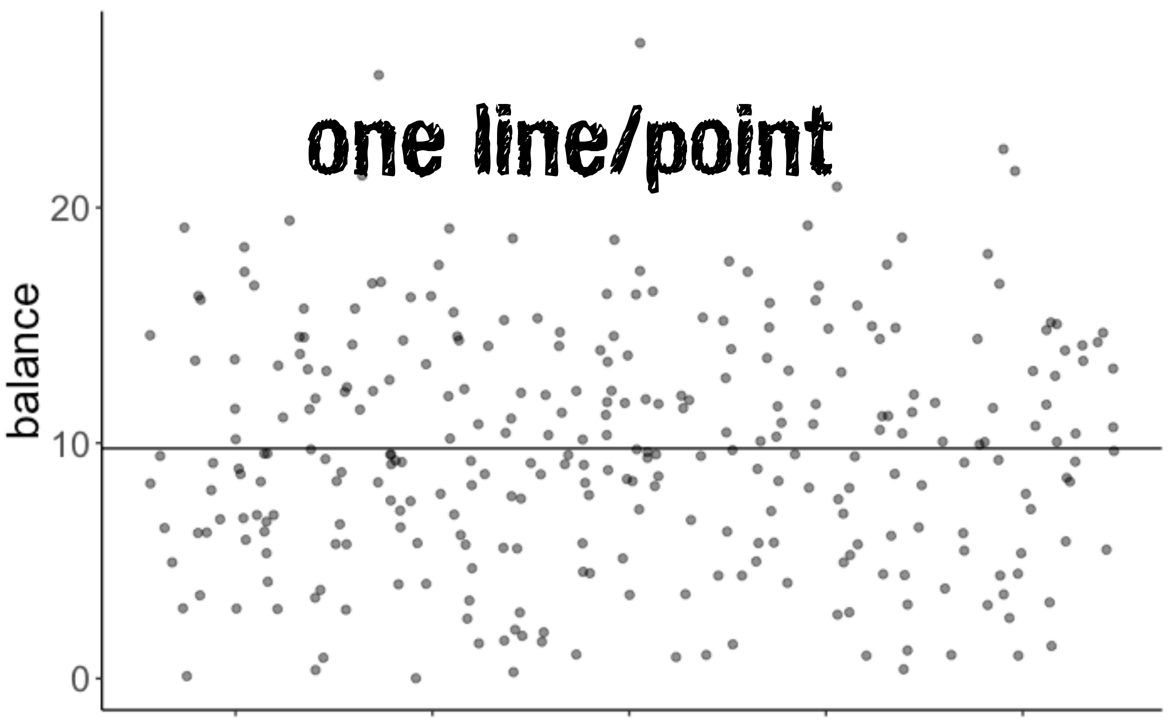


H$_0$: Card quality does not affect the final balance.

Model C

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

Model prediction



Fitted model

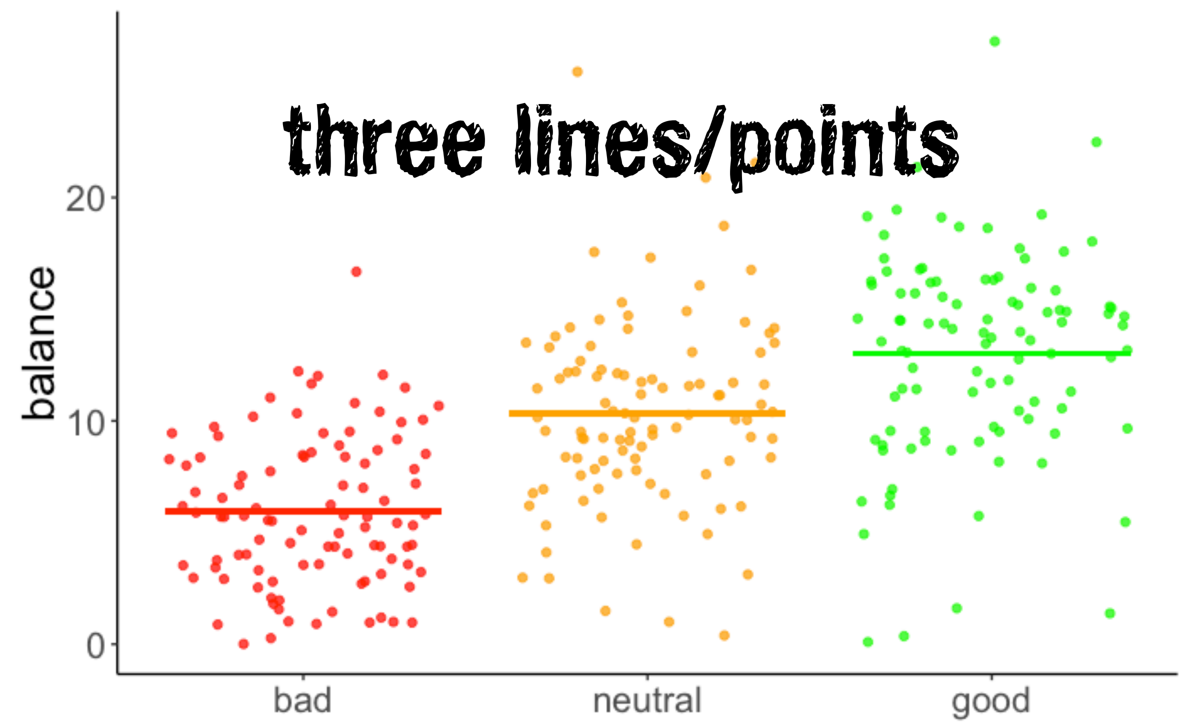
$$\widehat{\text{balance}}_i = 9.77$$

H$_1$: Card quality affects the final balance.

Model A

$$\text{balance}_i = \beta_0 + \beta_1 \text{hand\_neutral}_i + \beta_2 \text{hand\_good}_i + \epsilon$$

Model prediction



Fitted model

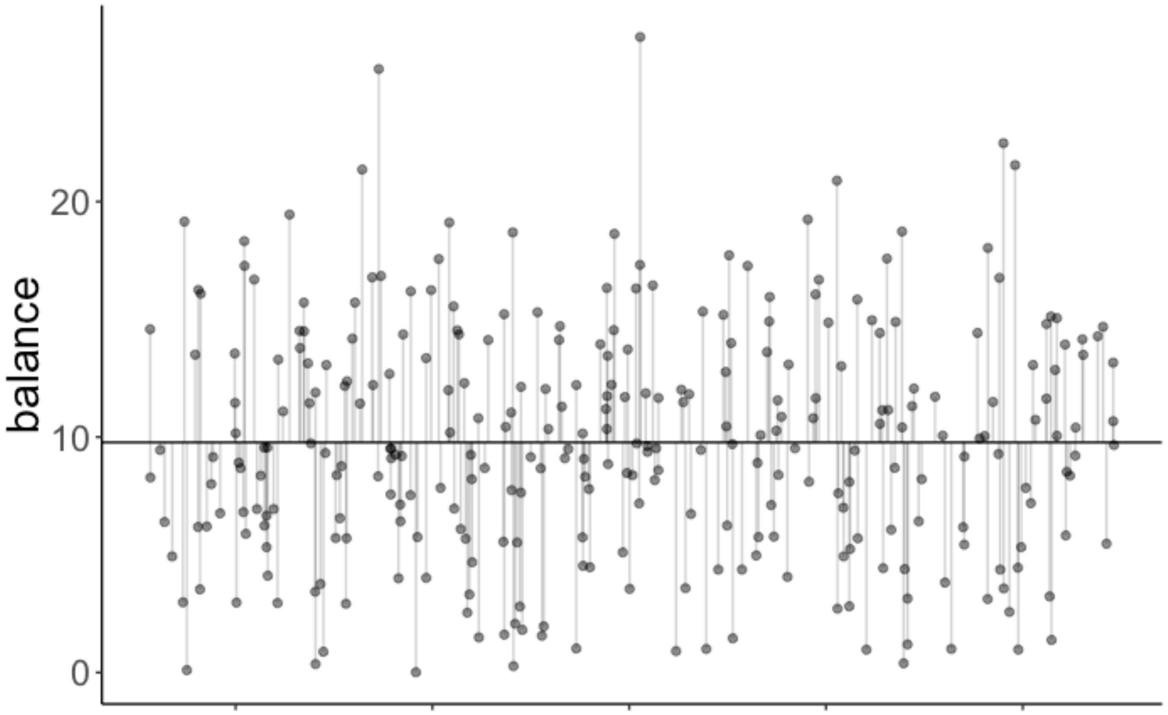
$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$

H_0: Card quality does not affect the final balance.

Model C

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

Model prediction



SSE(C) = 7580

Fitted model

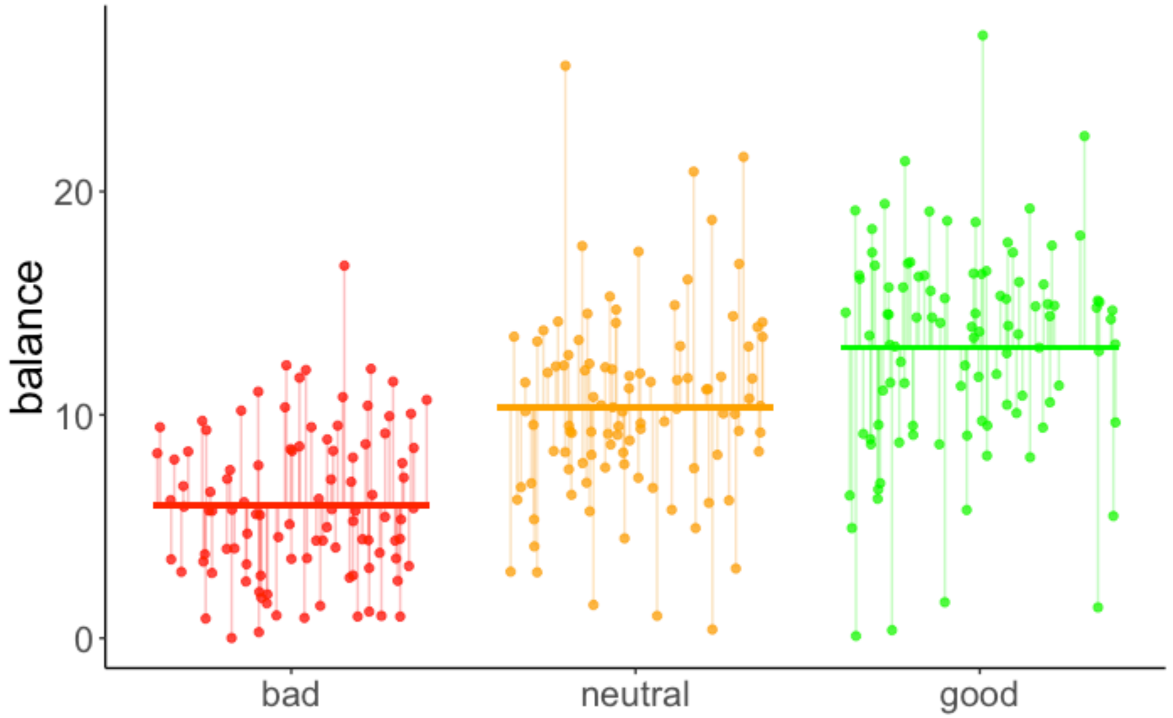
$$\widehat{\text{balance}}_i = 9.77$$

H_1: Card quality affects the final balance.

Model A

$$\text{balance}_i = \beta_0 + \beta_1 \text{hand\_neutral}_i + \beta_2 \text{hand\_good}_i + \epsilon$$

Model prediction



SSE(A) = 5021

Fitted model

$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$

Does card quality affect the final balance?

SSE(C) = 7580

SSE(A) = 5021

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

worth it?

$$= 1 - \frac{5021}{7580} \approx 0.34$$

```r
# fit the models
fit_c = lm(formula = balance ~ 1, data = df.poker)
fit_a = lm(formula = balance ~ hand, data = df.poker)

# compare via F-test
anova(fit_c, fit_a)
```

```
Analysis of Variance Table

Model 1: balance ~ 1
Model 2: balance ~ hand
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    299 7580.0
2    297 5020.6  2    2559.4 75.703 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpreting the results

```
lm(formula = balance ~ 1 + hand, data = df.poker)
```

```
Call:
lm(formula = balance ~ hand, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.9264  -2.5902  -0.0115   2.6573  15.2834 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.9415     0.4111  14.451  < 2e-16 ***
handneutral   4.4051     0.5815   7.576 4.55e-13 ***
handgood      7.0849     0.5815  12.185  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.111 on 297 degrees of freedom
Multiple R-squared:  0.3377, Adjusted R-squared:  0.3332 
F-statistic:  75.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

Dummy coding

```r
df.poker %>%
  mutate(hand_neutral = ifelse(hand == "neutral", 1, 0),
         hand_good = ifelse(hand == "good", 1, 0))
```

participant	hand	hand_neutral	hand_good	balance
31	bad	0	0	12.22
46	bad	0	0	12.06
50	bad	0	0	16.68
76	neutral	1	0	21.55
87	neutral	1	0	20.89
89	neutral	1	0	25.63
127	good	0	1	26.99
129	good	0	1	21.36
283	good	0	1	22.48

for a variable with k levels, we need k-1 dummy variables for encoding

same same, but different

```r
lm(formula = balance ~ 1 + hand_neutral + hand_good + data = df.poker)

lm(formula = balance ~ 1 + hand, data = df.poker)
```

Interpreting the results

regression coefficients encode differences between group means

term	estimate	std.error	statistic	p.value
(Intercept)	5.941	0.411	14.451	0
handneutral	4.405	0.581	7.576	0
handgood	7.085	0.581	12.185	0

$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$

participant	hand	hand_neutral	hand_good	balance
31	bad	0	0	12.22
46	bad	0	0	12.06
50	bad	0	0	16.68
76	neutral	1	0	21.55
87	neutral	1	0	20.89
89	neutral	1	0	25.63
127	good	0	1	26.99
129	good	0	1	21.36
283	good	0	1	22.48

if hand == "bad":

$$\widehat{\text{balance}}_i = 5.94$$

if hand == "neutral":

$$\widehat{\text{balance}}_i = 5.94 + 4.41 = 10.35$$

if hand == "good":

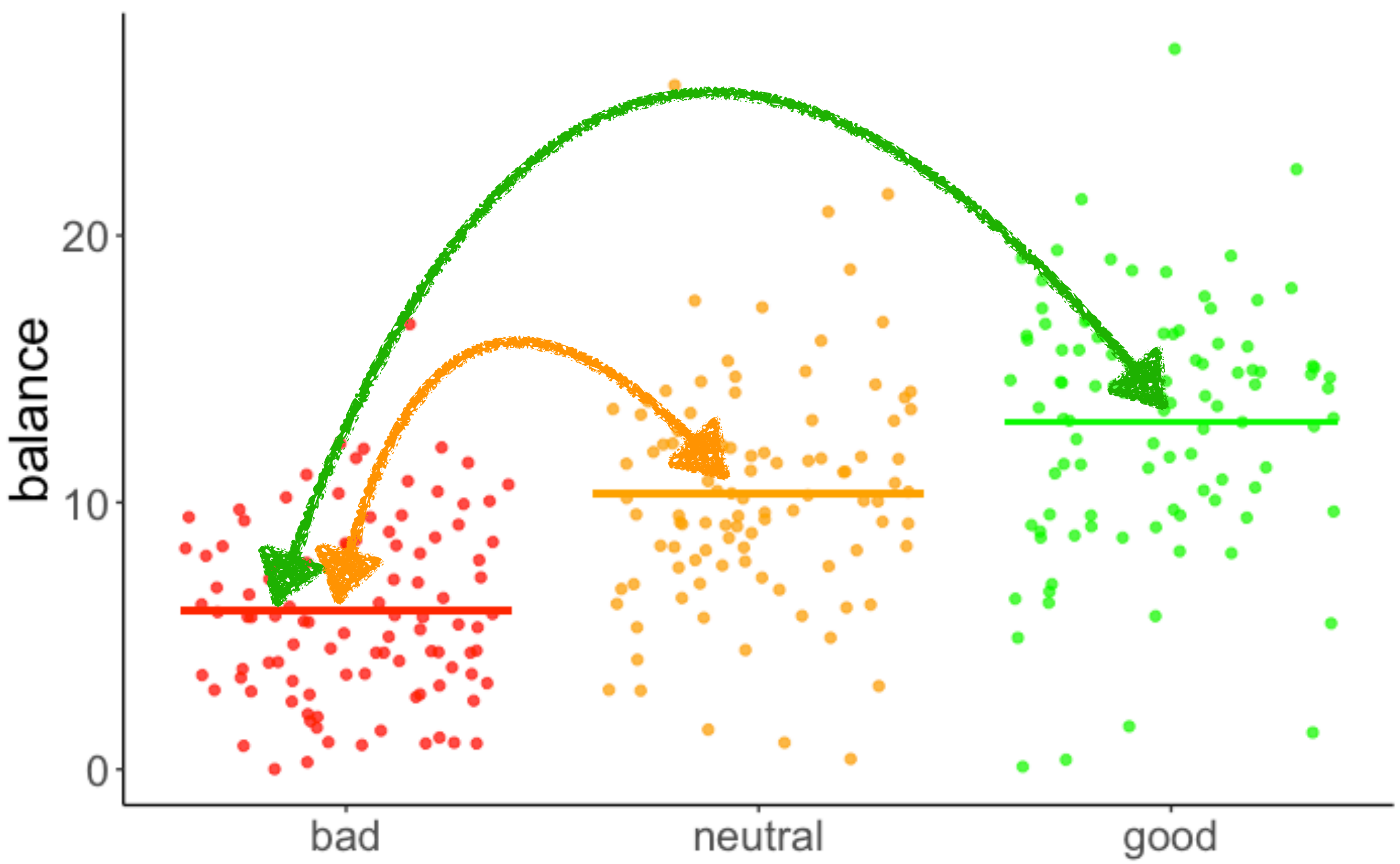
$$\widehat{\text{balance}}_i = 5.94 + 7.08 = 13.02$$

Interpreting the results

regression coefficients encode differences between group means

term	estimate	std.error	statistic	p.value
(Intercept)	5.941	0.411	14.451	0
handneutral	4.405	0.581	7.576	0
handgood	7.085	0.581	12.185	0

$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$



if hand == "bad":

$$\widehat{\text{balance}}_i = 5.94$$

if hand == "neutral":

$$\widehat{\text{balance}}_i = 5.94 + 4.41 = 10.35$$

if hand == "good":

$$\widehat{\text{balance}}_i = 5.94 + 7.08 = 13.02$$

One-way ANOVA

```
lm(formula = balance ~ hand, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value    Pr(>F)
hand        2 2559.4  1279.7  75.703 < 2.2e-16 ***
Residuals 297 5020.6    16.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What do these mean?

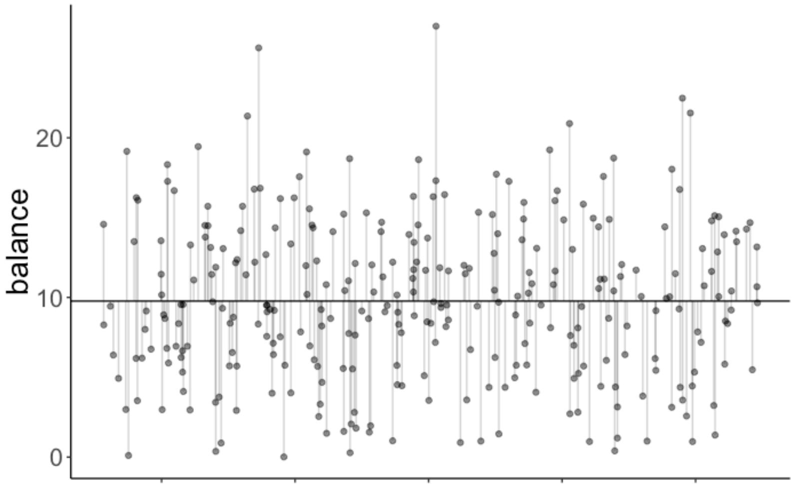
```
# fit the models
fit_c = lm(formula = balance ~ 1, data = df.poker)
fit_a = lm(formula = balance ~ hand, data = df.poker)

# compare via F-test
anova(fit_c, fit_a)
```

```
Analysis of Variance Table

Model 1: balance ~ 1
Model 2: balance ~ hand
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    299 7580.0
2    297 5020.6  2    2559.4 75.703 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One-way ANOVA

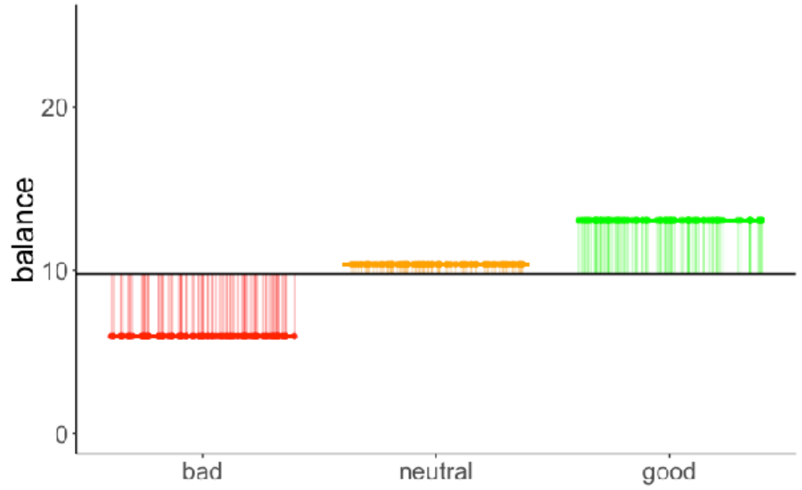
Variance decomposition

Total variance



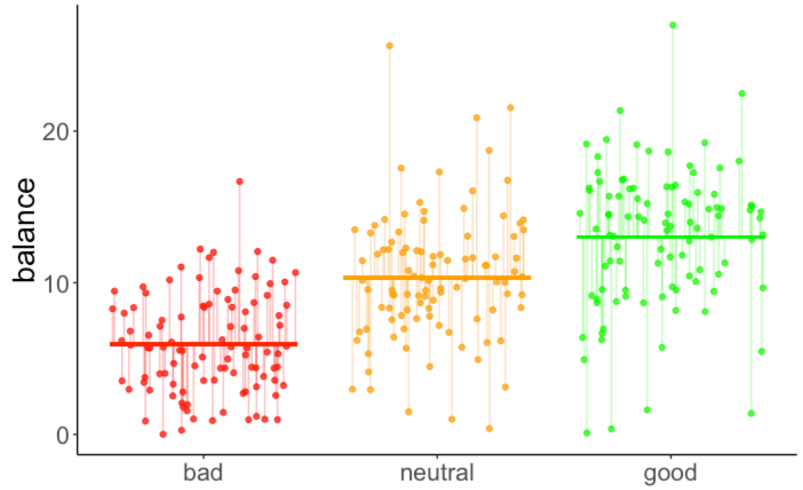
SS_{total}

Model variance



SS_{model}

Residual variance



$SS_{residual}$

variance_total	variance_model	variance_residual
7580	2559	5021

One-way ANOVA

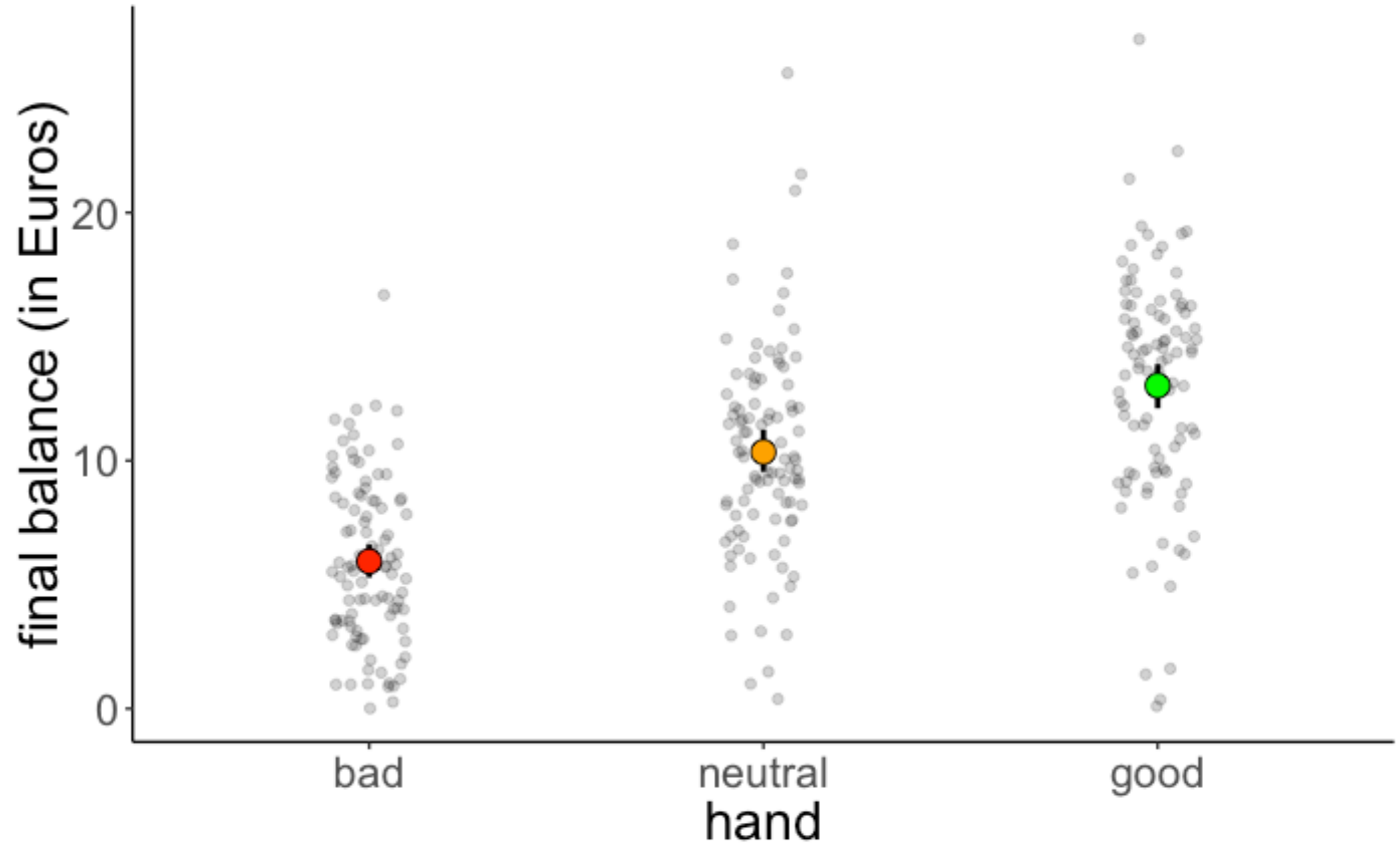
```
df.poker %>%
  mutate(mean_grand = mean(balance)) %>%
  group_by(hand) %>%
  mutate(mean_group = mean(balance)) %>%
```

participant	hand	balance	mean_grand	mean_group
1	bad	4.00	9.771	5.941
2	bad	5.55	9.771	5.941
3	bad	9.45	9.771	5.941
51	neutral	11.74	9.771	10.347
52	neutral	10.04	9.771	10.347
53	neutral	9.49	9.771	10.347
101	good	10.86	9.771	13.026
102	good	8.68	9.771	13.026
103	good	14.36	9.771	13.026

variance_total	variance_model	variance_residual
7580	2559	5021

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value    Pr(>F)
hand        2 2559.4  1279.7  75.703 < 2.2e-16 ***
Residuals 297 5020.6    16.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

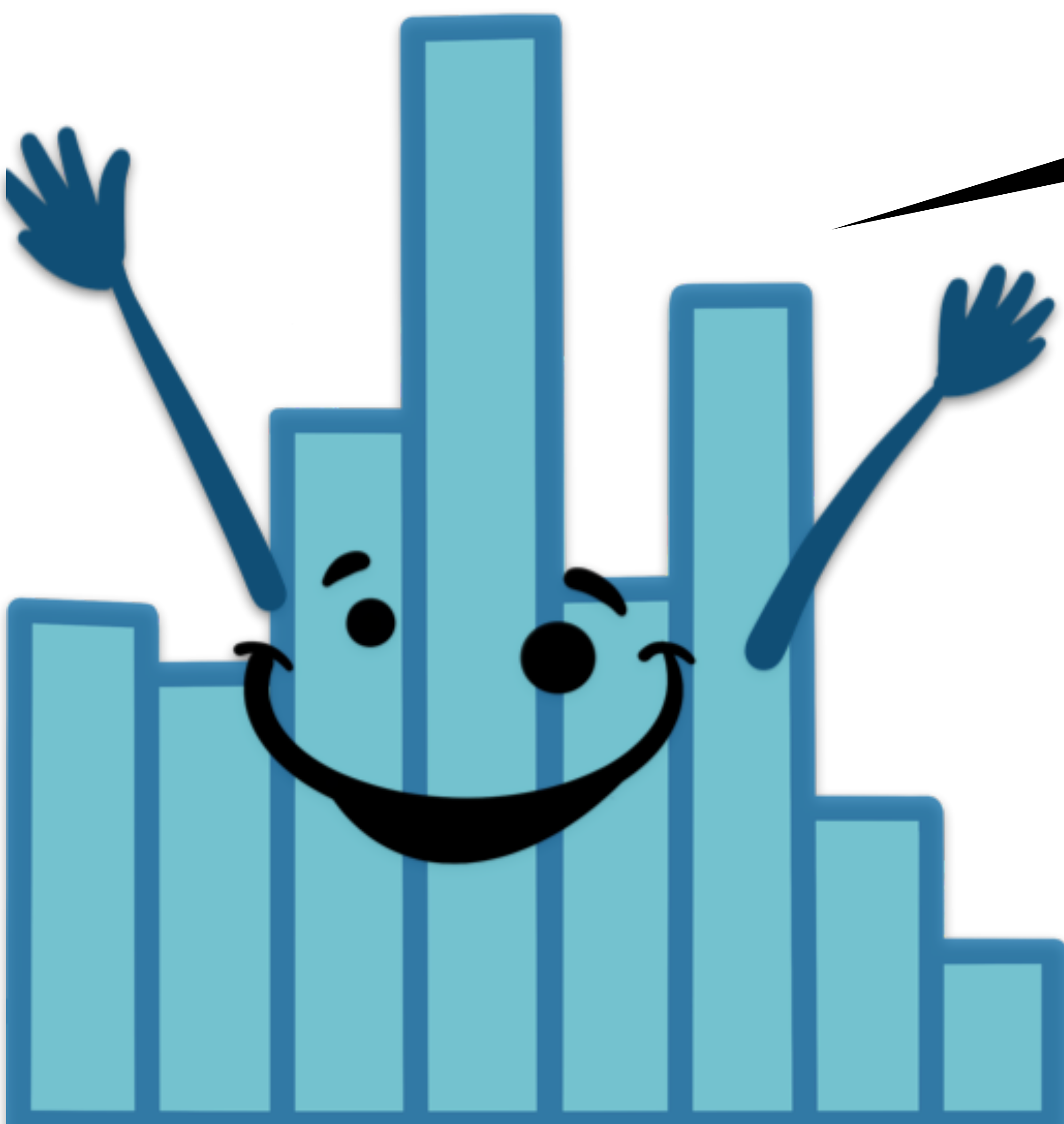
Reporting an ANOVA



The final balance differed significantly as a function of the quality of a player's hand (i.e. whether the hand was bad, neutral, or good), $F(2, 297) = 75.703$, $p < .001$.

01:00

stretch break!



Follow-up tests

nice tutorial

https://timmastny.rbind.io/blog/tests-pairwise-categorical-mean-emmeans-contrast/#cheatsheet

Asking more specific questions

Is there a difference in the final balance between bad hands and neutral hands?

```r
df.poker %>%
  filter(hand %in% c("bad", "neutral")) %>%
  lm(formula = balance ~ hand,
     data = .) %>%
  summary()
```

```
Call:
lm(formula = balance ~ hand, data = .)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.9566 -2.5078 -0.2365  2.4410 15.2834 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.9415     0.3816  15.570  < 2e-16 ***
handneutral   4.4051     0.5397   8.163 3.76e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.816 on 198 degrees of freedom
Multiple R-squared:  0.2518,	Adjusted R-squared:  0.248 
F-statistic: 66.63 on 1 and 198 DF,  p-value: 3.758e-14
```

Interpreting the results

```
lm(formula = balance ~ hand, data = df.poker)
```

```
Call:
lm(formula = balance ~ hand, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.9264  -2.5902  -0.0115   2.6573  15.2834 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.9415     0.4111  14.451  < 2e-16 ***
handneutral   4.4051     0.5815   7.576 4.55e-13 ***
handgood      7.0849     0.5815  12.185  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.111 on 297 degrees of freedom
Multiple R-squared:  0.3377, Adjusted R-squared:  0.3332 
F-statistic:  75.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

What does this summary not tell us?

Model comparison

Is there a difference in the final balance between neutral hands and good hands?

Model C

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

Model A

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

(after having removed bad hands from the data set)

Asking more specific questions

Is there a difference in the final balance between neutral hands and good hands?

```
df.poker %>%
  filter(hand %in% c("neutral", "good")) %>%
  lm(formula = balance ~ hand,
     data = .) %>%
  summary()
```

```
Call:
lm(formula = balance ~ hand, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9264  -2.7141   0.2585   2.7184  15.2834

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.3466     0.4448   23.26  < 2e-16 ***
handgood      2.6798     0.6291    4.26 3.16e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.448 on 198 degrees of freedom
Multiple R-squared:  0.08396,   Adjusted R-squared:  0.07933
F-statistic: 18.15 on 1 and 198 DF,  p-value: 3.158e-05
```

Asking more specific questions

Is there a difference in the final balance between neutral hands and good hands?

```
df.poker %>%
  mutate(hand = fct_relevel(hand, "neutral")) %>%
  lm(formula = balance ~ hand,
       data = .) %>%
  summary()
```

same same, but different

```
Call:
lm(formula = balance ~ hand, data = .)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.9264  -2.5902  -0.0115   2.6573  15.2834 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  10.3466     0.4111  25.165  < 2e-16 ***
handbad      -4.4051     0.5815  -7.576 4.55e-13 ***
handgood      2.6798     0.5815   4.609 6.02e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.111 on 297 degrees of freedom
Multiple R-squared:  0.3377,	Adjusted R-squared:  0.3332 
F-statistic:  75.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

Is there a difference between bad hands vs. other hands?

```
df.poker %>%
  mutate(hand_other = ifelse(hand %in% c("neutral", "good"), 1, 0)) %>%
  lm(balance ~ 1 + hand_other,
     data = .) %>%
  summary()
```

```
Call:
lm(formula = balance ~ 1 + hand_other, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5865  -2.6203  -0.1815   2.8285  15.3035

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.9415     0.4249   13.98   <2e-16 ***
hand_other    5.7450     0.5204   11.04   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.249 on 298 degrees of freedom
Multiple R-squared:  0.2903,	Adjusted R-squared:  0.2879
F-statistic: 121.9 on 1 and 298 DF,  p-value: < 2.2e-16
```

df.poker

participant	hand	hand_other	balance
31	bad	0	12.22
46	bad	0	12.06
50	bad	0	16.68
76	neutral	1	21.55
87	neutral	1	20.89
89	neutral	1	25.63
127	good	1	26.99
129	good	1	21.36
283	good	1	22.48

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_other}_i$$

if hand == bad: $\widehat{\text{balance}}_i = b_0 = 5.94$

if hand != bad: $\widehat{\text{balance}}_i = b_0 + b_1 = 5.94 + 5.75 = 11.69$

group means

bad	neutral	good
5.94	10.35	13.03

Multiple categorical predictors

Do skill level and quality of cards affect the final balance?

participant	skill	hand	limit	balance
1	expert	bad	fixed	4.00
2	expert	bad	fixed	5.55
26	expert	bad	none	5.52
27	expert	bad	none	8.28
51	expert	neutral	fixed	11.74
52	expert	neutral	fixed	10.04
76	expert	neutral	none	21.55
77	expert	neutral	none	3.12
101	expert	good	fixed	10.86
102	expert	good	fixed	8.68

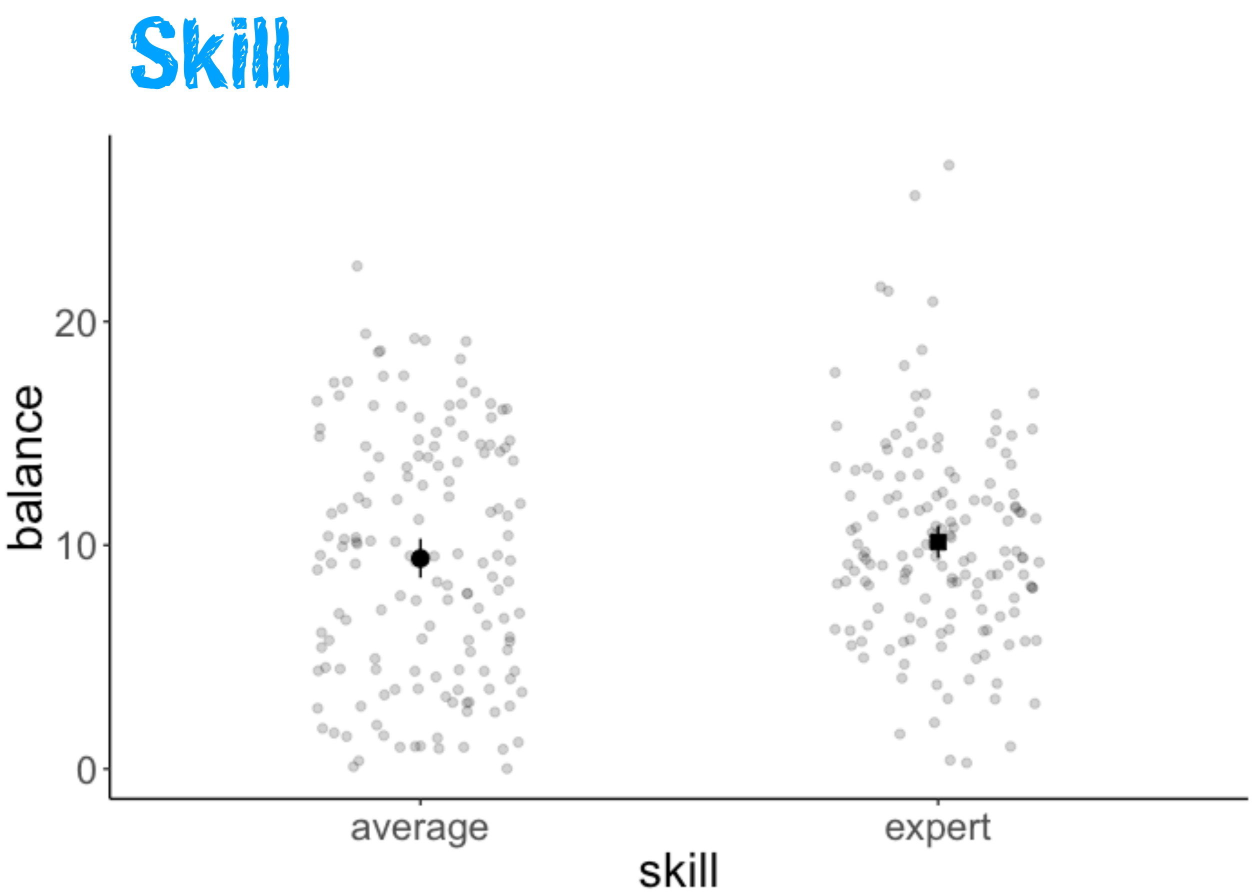
Why not just fit separate models?

One testing whether skill level affects the final balance, and one testing whether quality of cards affects the final balance?

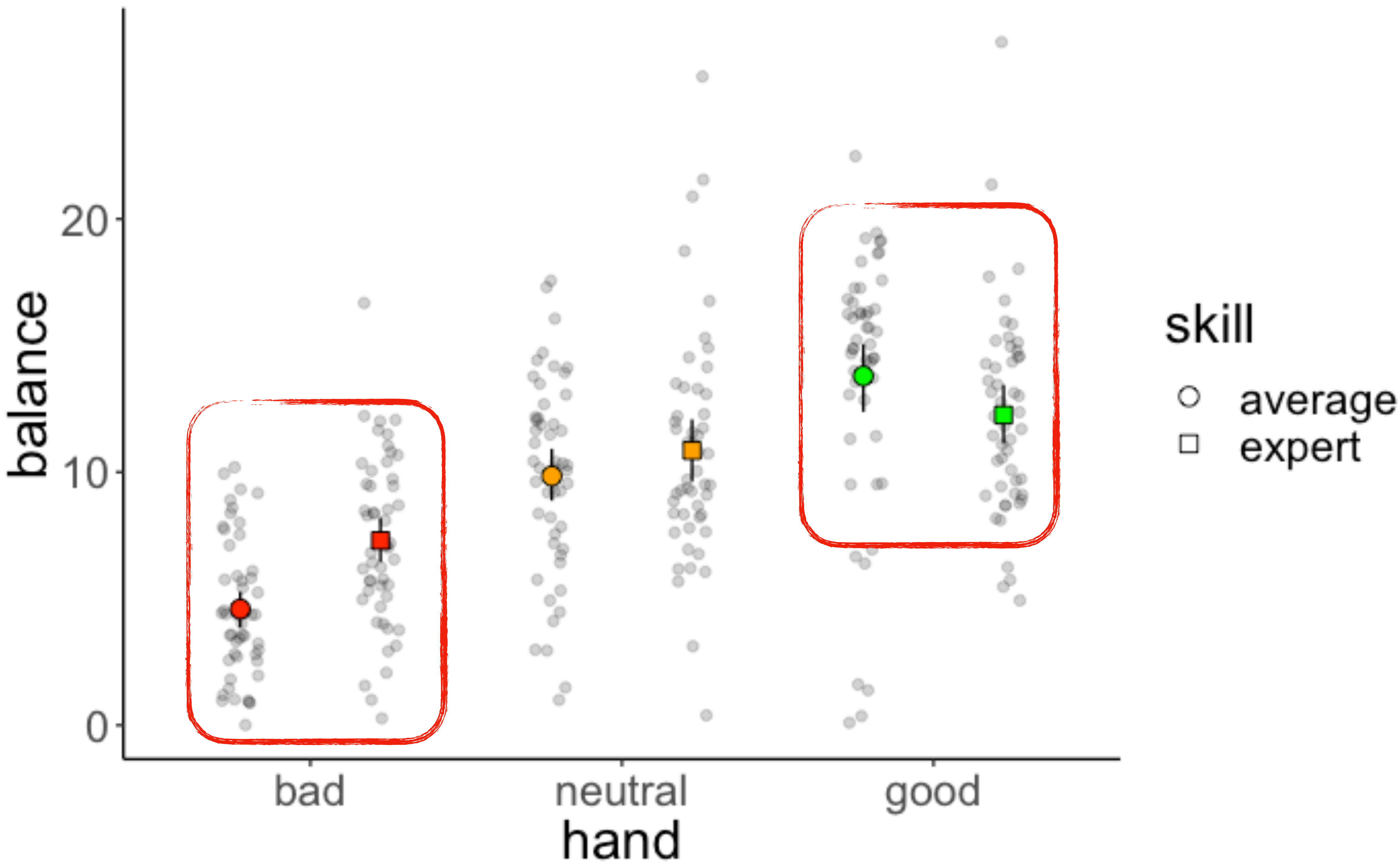
Interested in interactions!

Does the effect of one variable depend on the other?

Visualize the data



Visualize the data



Fit a model

```r
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  summary()
```

```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.6976  -2.4740   0.0348   2.4644  14.7806 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)               4.5866     0.5686   8.067 1.85e-14 ***
handneutral               5.2572     0.8041   6.538 2.75e-10 ***
handgood                  9.2110     0.8041  11.455  < 2e-16 ***
skillexpert               2.7098     0.8041   3.370 0.000852 ***
handneutral:skillexpert  -1.7042     1.1372  -1.499 0.135038    
handgood:skillexpert     -4.2522     1.1372  -3.739 0.000222 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.02 on 294 degrees of freedom
Multiple R-squared:  0.3731,Adjusted R-squared:  0.3624 
F-statistic: 34.99 on 5 and 294 DF,  p-value: < 2.2e-16
```

Interpretation

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)               4.5866     0.5686   8.067 1.85e-14 ***
handneutral               5.2572     0.8041   6.538 2.75e-10 ***
handgood                  9.2110     0.8041  11.455  < 2e-16 ***
skillexpert               2.7098     0.8041   3.370 0.000852 ***
handneutral:skillexpert  -1.7042     1.1372  -1.499 0.135038    
handgood:skillexpert     -4.2522     1.1372  -3.739 0.000222 ***
```

group means

skill	bad	neutral	good
average	4.59	9.84	13.80
expert	7.30	10.85	12.26

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_neutral}_i + b_2 \cdot \text{hand\_good}_i + b_3 \cdot \text{skill\_expert}_i + \\ b_4 \cdot \text{hand\_neutral:skill\_expert}_i + b_5 \cdot \text{hand\_good:skill\_expert}_i$$

hand = bad, skill = average

$$\widehat{\text{balance}}_i = b_0 = 4.59$$

hand = neutral, skill = average

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_neutral}_i = 4.59 + 5.26 = 9.85$$

hand = good, skill = expert

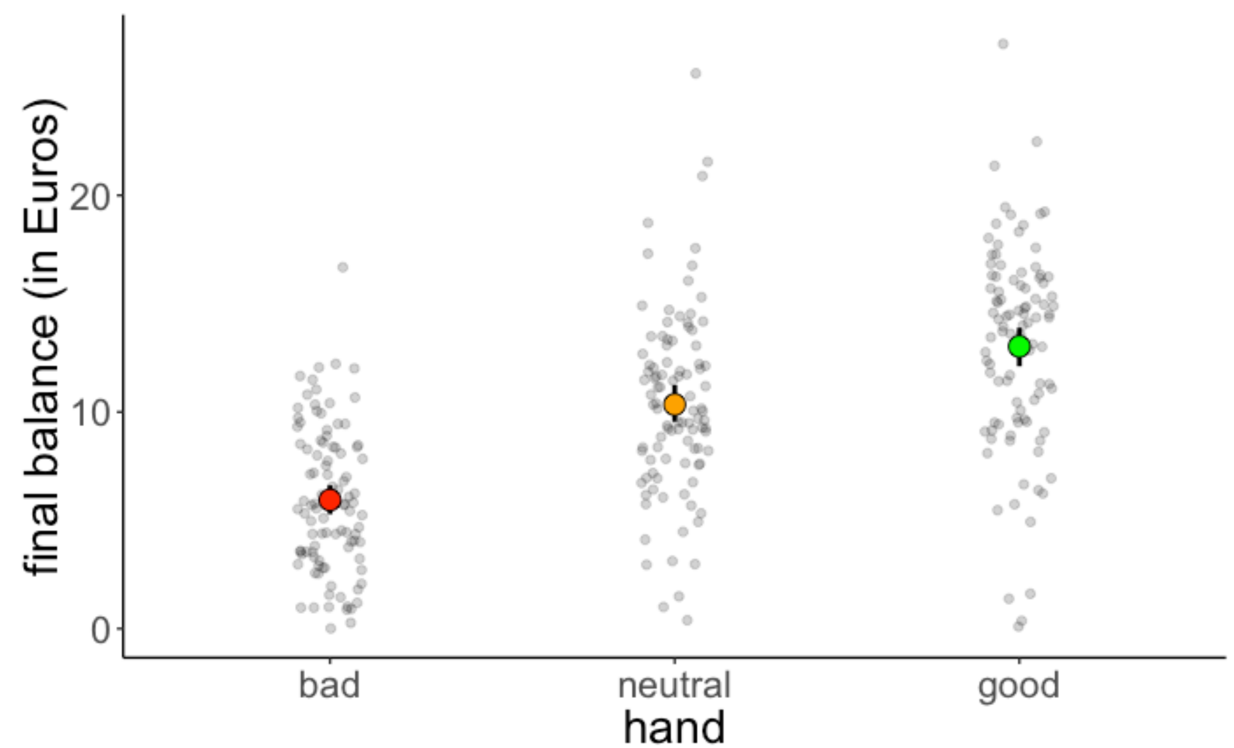
$$\widehat{\text{balance}}_i = b_0 + b_2 \cdot \text{hand\_good}_i + b_3 \cdot \text{skill\_expert}_i + b_5 \cdot \text{hand\_good:skill\_expert}_i \\ = 12.26$$

Analysis of variance

```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
            Df  Sum Sq Mean Sq F value    Pr(>F)    
hand         2 2559.4 1279.70 79.1692 < 2.2e-16 ***
skill        1   39.3   39.35  2.4344 0.1197776    
hand:skill   2  229.0  114.49  7.0830 0.0009901 ***
Residuals  294 4752.3   16.16                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

main effect of hand



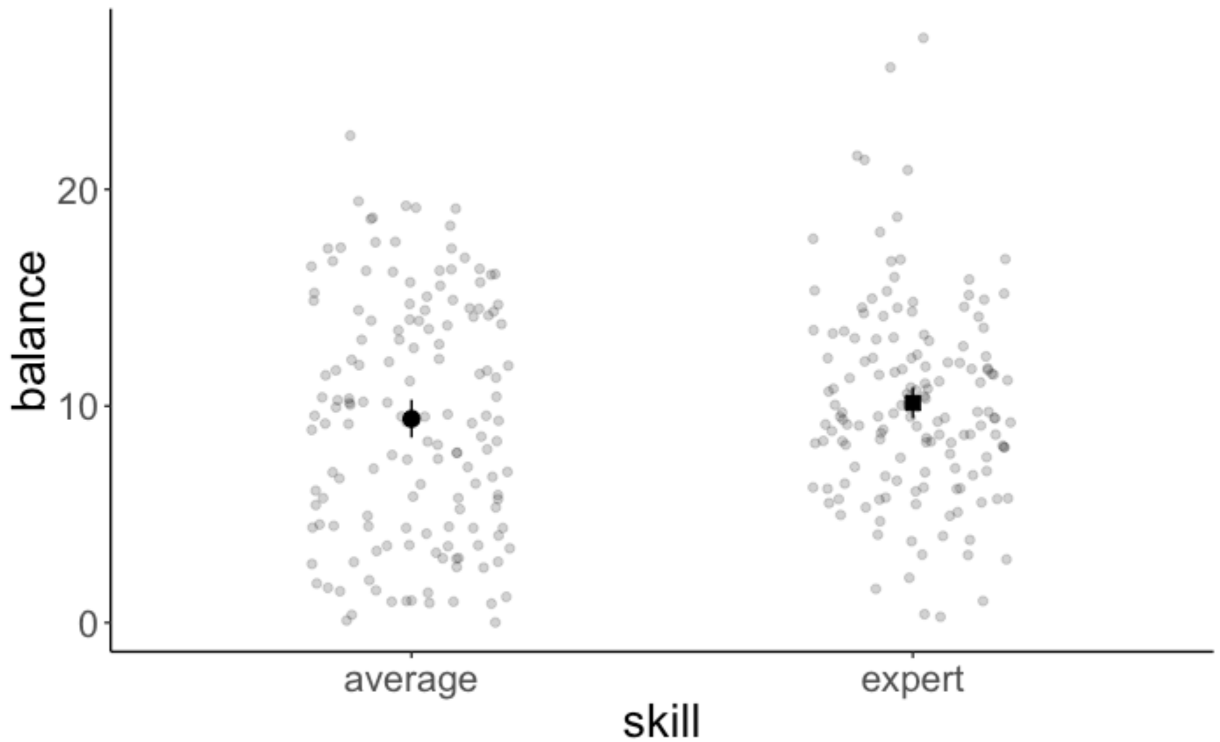
Analysis of variance

```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
             Df Sum Sq Mean Sq F value    Pr(>F)    
hand          2 2559.4 1279.70 79.1692 < 2.2e-16 ***
skill         1   39.3   39.35  2.4344 0.1197776    
hand:skill    2  229.0  114.49  7.0830 0.0009901 ***
Residuals   294 4752.3   16.16                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

main effect of `hand`

no main effect of `skill`



Analysis of variance

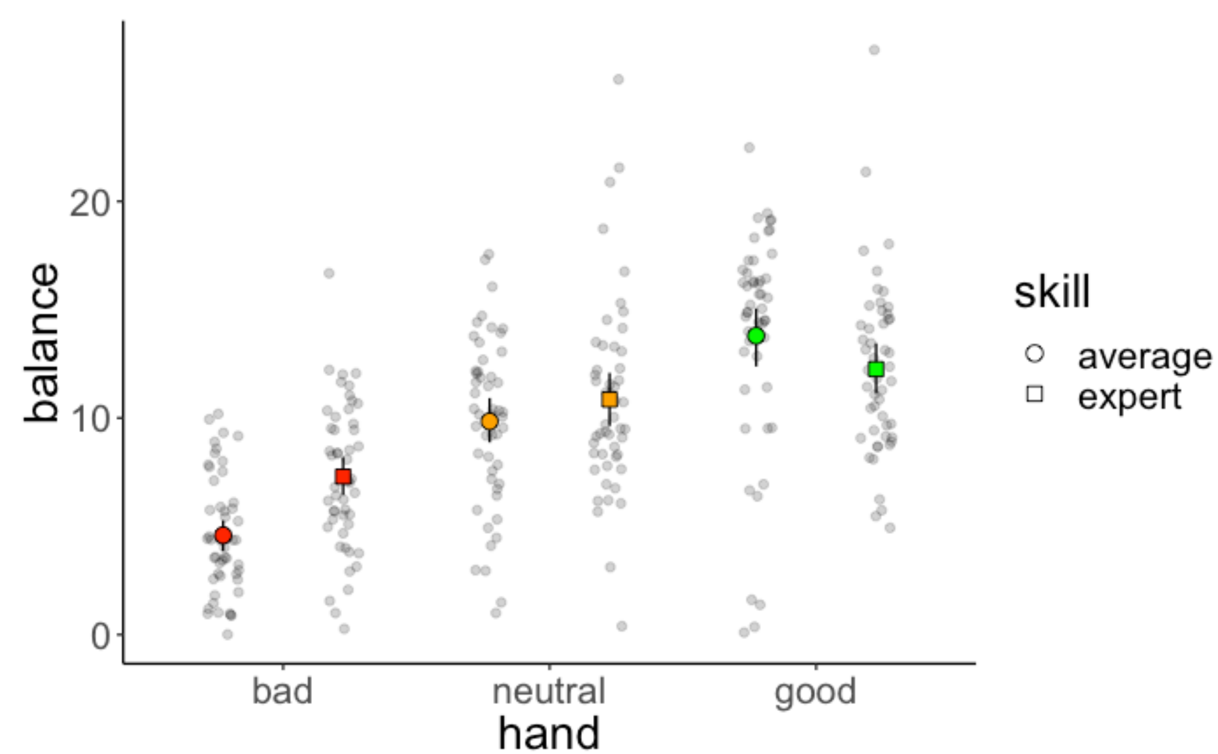
```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
            Df Sum Sq Mean Sq F value    Pr(>F)    
hand         2 2559.4 1279.70 79.1692 < 2.2e-16 ***
skill        1   39.3   39.35  2.4344 0.1197776    
hand:skill   2  229.0  114.49  7.0830 0.0009901 ***
Residuals  294 4752.3   16.16                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

main effect of `hand`

no main effect of `skill`

interaction between `hand` and `skill`



Two-way ANOVA

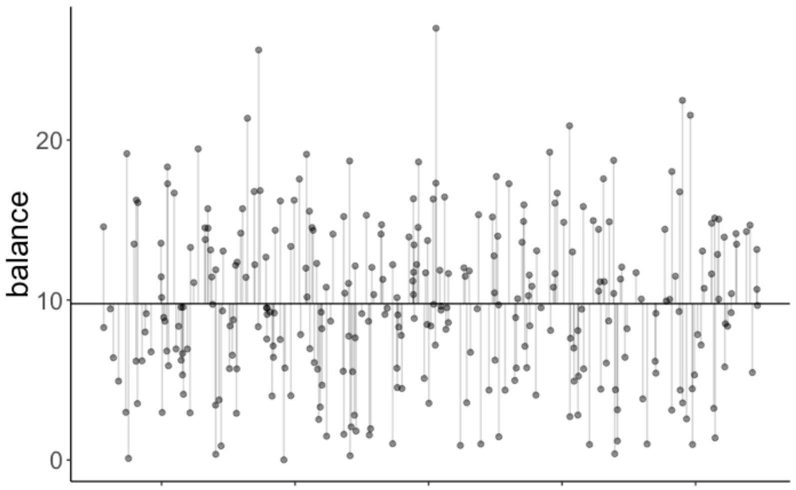
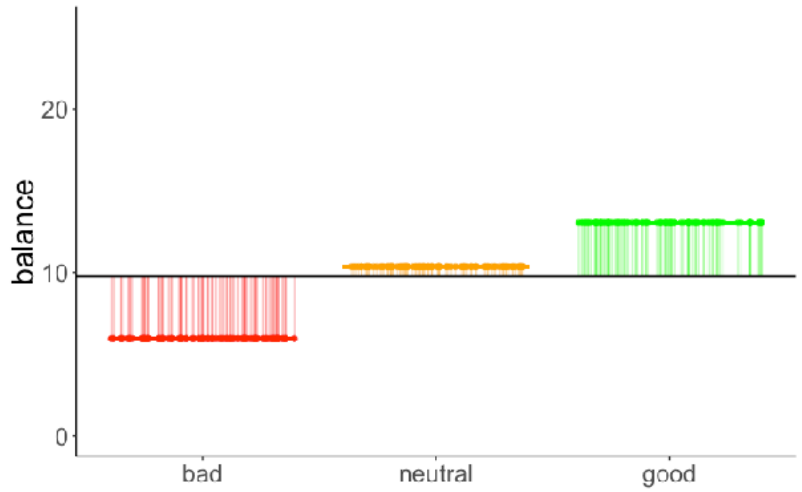
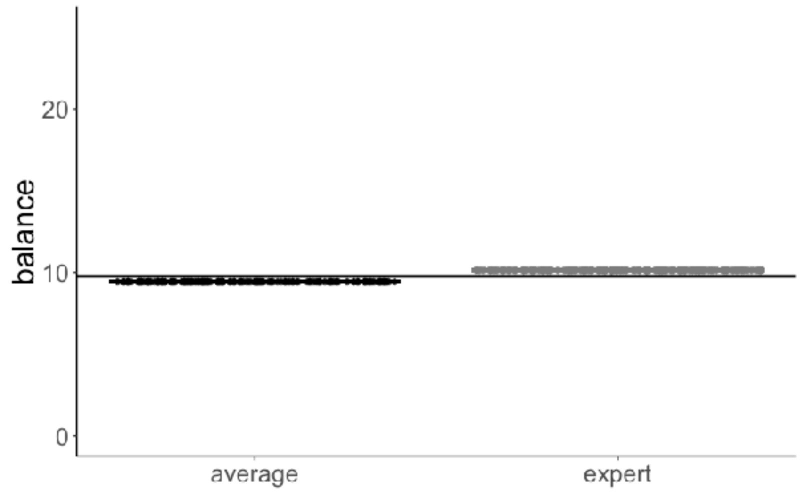
```
lm(formula = balance ~ hand + skill, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value Pr(>F)
hand        2 2559.4 1279.70 76.0437 <2e-16 ***
skill       1   39.3   39.35  2.3383 0.1273
Residuals 296 4981.2   16.83
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What do these mean?

Two-way ANOVA

Variance decomposition

Total variance	Hand variance	Skill variance
		
SS_{total}	SS_{hand}	SS_{skill}

$$SS_{residual} = SS_{total} - SS_{hand} - SS_{skill}$$

Two-way ANOVA

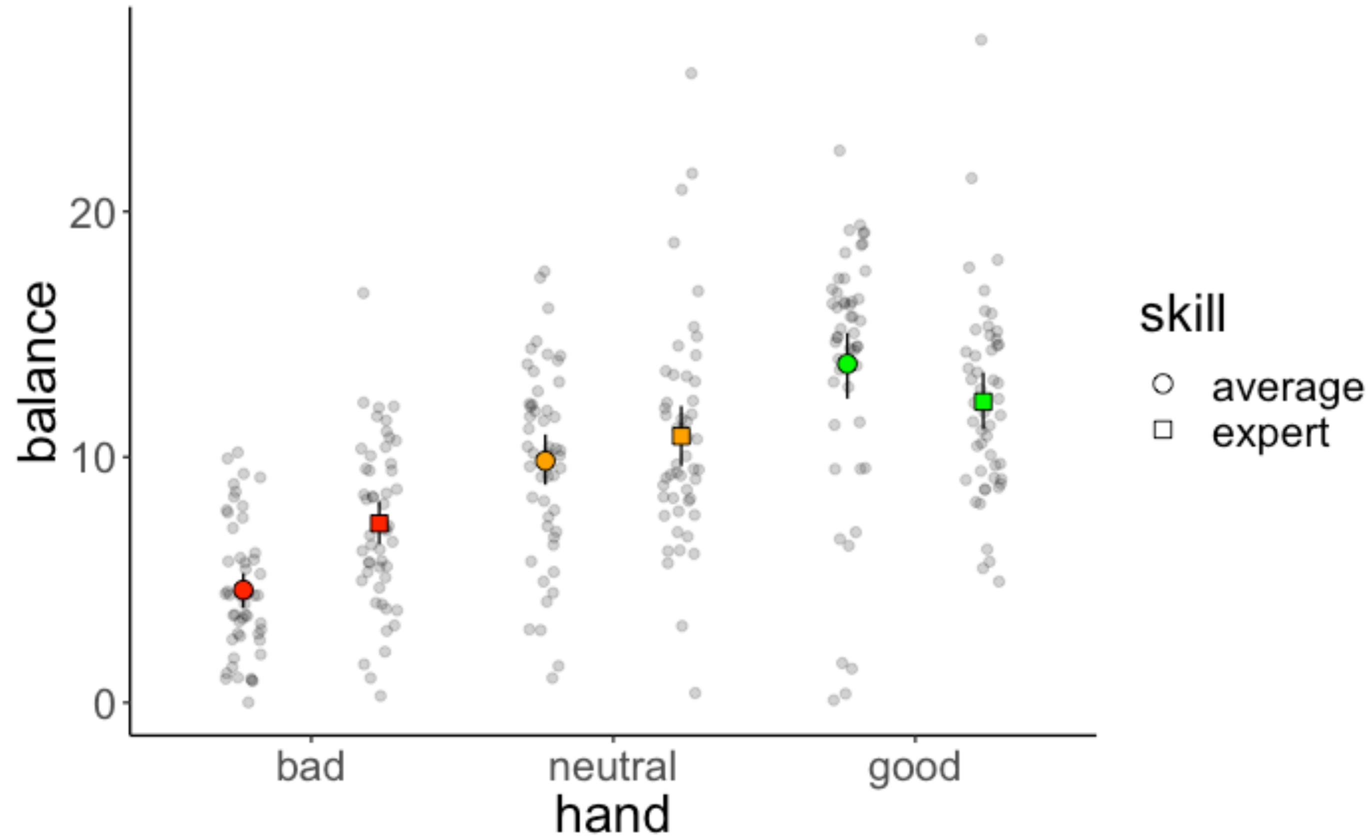
```
df.poker %>%
  mutate(mean_grand = mean(balance)) %>%
  group_by(skill) %>%
  mutate(mean_skill = mean(balance)) %>%
  group_by(hand) %>%
  mutate(mean hand = mean(balance)) %>%
```

participant	skill	hand	balance	mean_grand	mean_skill	mean_hand
1	expert	bad	4.00	9.77	10.13	5.94
2	expert	bad	5.55	9.77	10.13	5.94
51	expert	neutral	11.74	9.77	10.13	10.35
52	expert	neutral	10.04	9.77	10.13	10.35
101	expert	good	10.86	9.77	10.13	13.03
102	expert	good	8.68	9.77	10.13	13.03
151	average	bad	4.37	9.77	9.41	5.94
152	average	bad	3.58	9.77	9.41	5.94
201	average	neutral	6.42	9.77	9.41	10.35
202	average	neutral	14.18	9.77	9.41	10.35

variance_total	variance_skill	variance_hand	variance_residual
7580	39	2559	4981

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value Pr(>F)
hand        2 2559.4 1279.70 76.0437 <2e-16 ***
skill       1   39.3   39.35  2.3383 0.1273
Residuals 296 4981.2   16.83
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reporting the results



There was no main effect of skill $F(1, 294) = 2.43$, $p = .12$. The final balance of average (M = 9.41, SD = 5.51) and expert poker players (M = 10.13, SD = 4.50) did not differ significantly.

However, the quality of a player's hand significantly affected the final balance F(2, 294) = 79.17, p < .001. The final balance for good hands (M = 13.03, SD = 4.65) was significantly greater than for neutral hands (M = 10.35, SD = 4.24), and the balance for neutral hands was significantly higher than for bad hands (M = 5.94, SD = 3.34).

There was also a significant interaction between the quality of a player's hand and the player's skill level F(2, 294) = 7.08, p < .001. Whereas for bad hands, average players had a lower final balance than experts, for good hands, average players had a higher final balance than experts.

Interpreting parameters

Beware of misinterpretation

```
lm(formula = balance ~ hand, data = df.poker)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.9415     0.4111  14.451  < 2e-16 ***
handneutral   4.4051     0.5815   7.576 4.55e-13 ***
handgood      7.0849     0.5815  12.185  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

reference category

bad	neutral	good
5.94	10.35	13.03

```
lm(formula = balance ~ hand * skill, data = df.poker)
```

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               4.5866     0.5686   8.067 1.85e-14 ***
handneutral               5.2572     0.8041   6.538 2.75e-10 ***
handgood                  9.2110     0.8041  11.455  < 2e-16 ***
skillexpert               2.7098     0.8041   3.370 0.000852 ***
handneutral:skillexpert  -1.7042     1.1372  -1.499 0.135038
handgood:skillexpert     -4.2522     1.1372  -3.739 0.000222 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

reference category

skill	bad	neutral	good
average	4.59	9.84	13.80
expert	7.30	10.85	12.26

Effect coding

```
lm(formula = balance ~ hand, data = df.poker,
   contrasts = list(hand = "contr.sum"))
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.7715     0.2374  41.165   <2e-16 ***
hand1        -3.8300     0.3357 -11.409   <2e-16 ***
hand2         0.5751     0.3357   1.713   0.0877 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

reference

grand mean = 9.77

```
lm(formula = balance ~ hand * skill, data = df.poker,
   contrasts = list(hand = "contr.sum", skill = "contr.sum"))
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     9.7715     0.2321  42.096  < 2e-16 ***
hand1          -3.8300     0.3283 -11.667  < 2e-16 ***
hand2           0.5751     0.3283   1.752  0.08083 .
skill1         -0.3622     0.2321  -1.560  0.11978
hand1:skill1   -0.9927     0.3283  -3.024  0.00271 **
hand2:skill1   -0.1406     0.3283  -0.428  0.66867
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

reference

grand mean = 9.77

Note: The last level in each factor is dropped.

Parameter interpretation

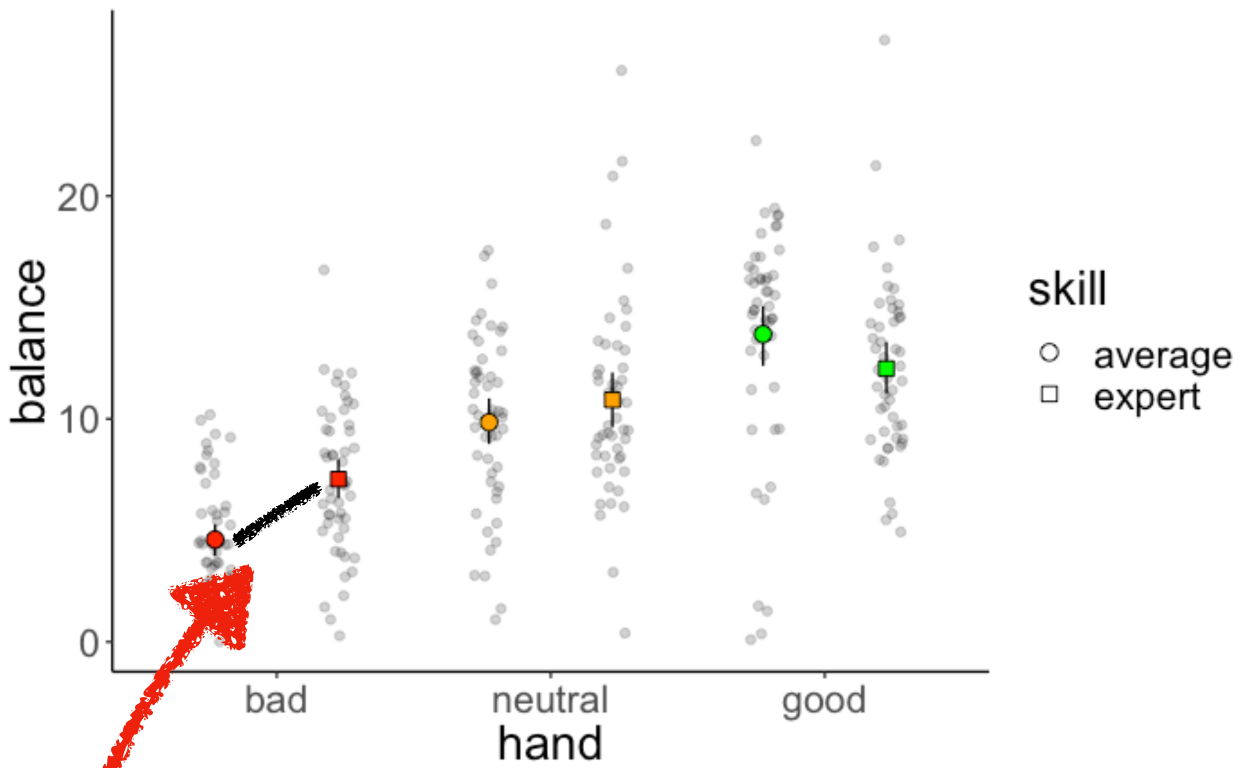
```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  summary()
```

```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.6976  -2.4740   0.0348   2.4644  14.7806 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)               4.5866     0.5686   8.067 1.85e-14 ***
handneutral               5.2572     0.8041   6.538 2.75e-10 ***
handgood                  9.2110     0.8041  11.455  < 2e-16 ***
skillexpert               2.7098     0.8041   3.370 0.000852 ***
handneutral:skillexpert  -1.7042     1.1372  -1.499 0.135038    
handgood:skillexpert     -4.2522     1.1372  -3.739 0.000222 ***
---
Sig                                                        ' 1

Residual standard error: 4.02 on 294 degrees of freedom
Multiple R-squared:  0.3731,Adjusted R-squared:  0.3624 
F-statistic: 34.99 on 5 and 294 DF,  p-value: < 2.2e-16
```

acute danger of misinterpretation!!

~~there was a significant effect of skill~~

Parameter interpretation




```
lm(formula = balance ~ hand * skill,
   data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
            Df Sum Sq Mean Sq F value    Pr(>F)
hand         2 2559.4 1279.70 79.1692 < 2.2e-16 ***
skill        1   39.3   39.35  2.4344 0.1197776
hand:skill   2  229.0  114.49  7.0830 0.0009901 ***
Residuals  294 4752.3   16.16
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1
```

there was no main effect of skill!

```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max
-13.6976  -2.4740   0.0348   2.4644  14.7806

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               4.5866     0.5686   8.067 1.85e-14 ***
handneutral               5.2572     0.8041   6.538 2.75e-10 ***
handgood                  9.2110     0.8041  11.455  < 2e-16 ***
skillexpert               2.7098     0.8041   3.370 0.000852 ***
handneutral:skillexpert  -1.7042     1.1372  -1.499 0.135038
handgood:skillexpert     -4.2522     1.1372  -3.739 0.000222 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.02 on 294 degrees of freedom
Multiple R-squared:  0.3731,	Adjusted R-squared:  0.3624
F-statistic: 34.99 on 5 and 294 DF,  p-value: < 2.2e-16
```

hand	average	expert	difference
bad	4.59	7.3	2.71

is this difference significantly different from 0?

Effects in an ANOVA

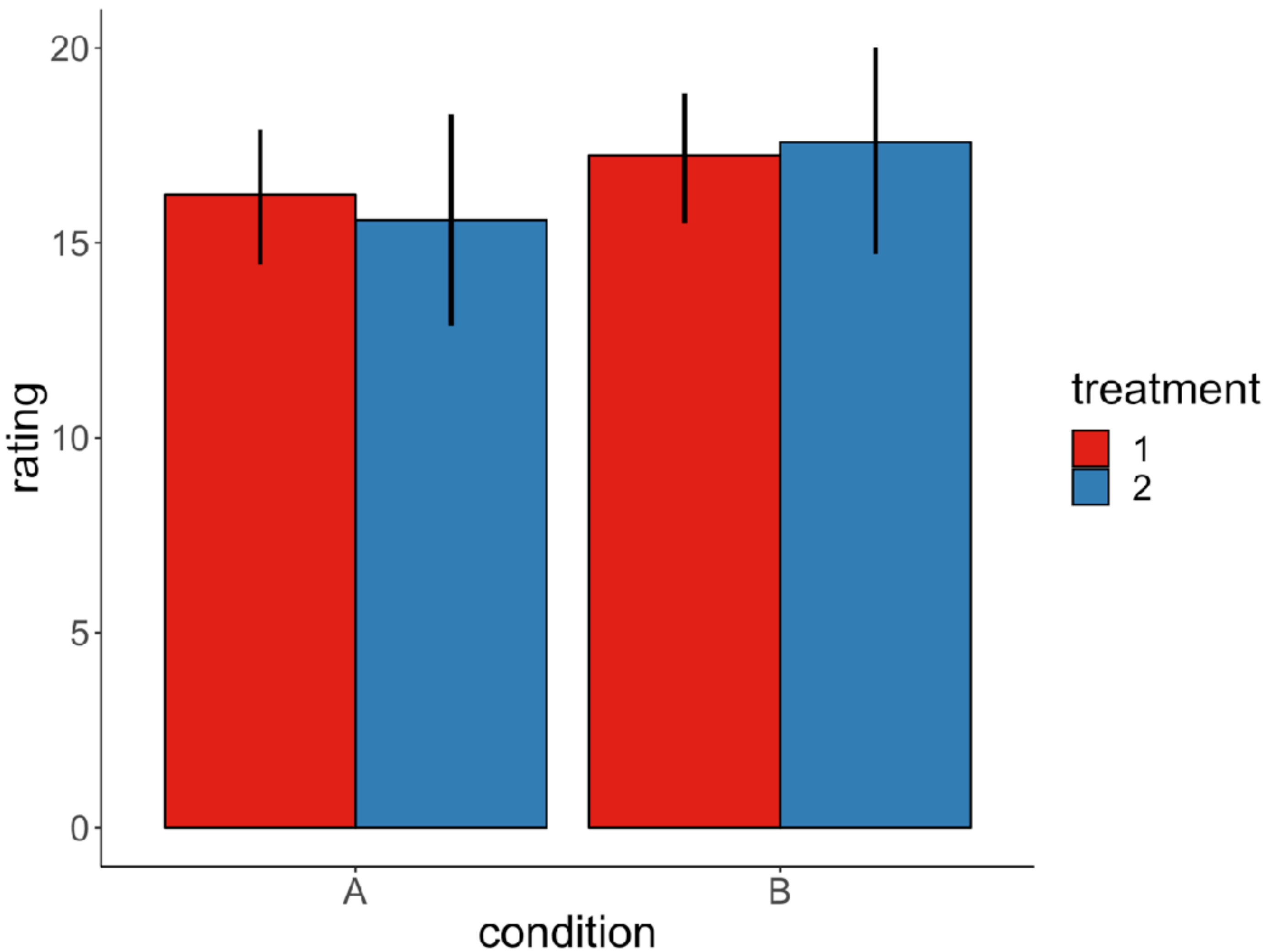
- **main effect**: effect of one independent variable on the dependent variable
- **interaction effect**: when the effect of one independent variable depends on the level of another
- **simple effect**: comparison between two specific cell means

https://opentext.wsu.edu/carriecuttler/chapter/9-2-main-effects-and-interactions/

Who is the ANOVA champ?

Who is the ANOVA champ?

Which effects are significant?



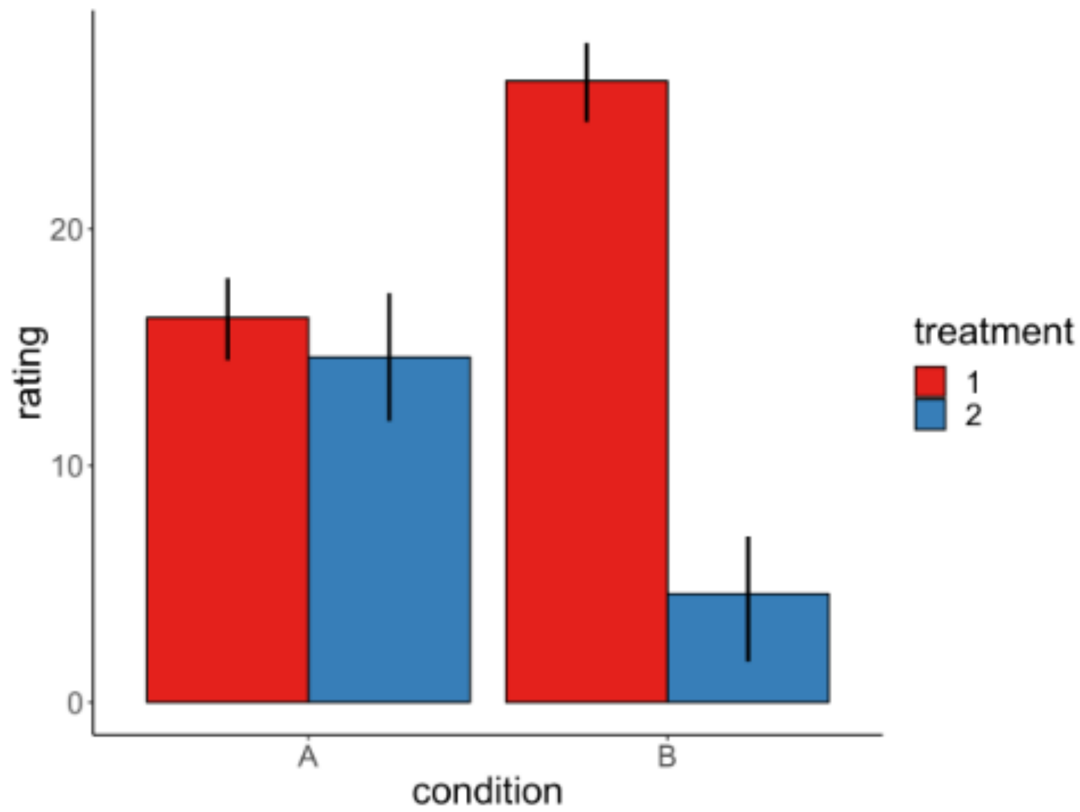
- Condition
- Treatment
- Condition x Treatment **interaction effect**
- Condition, Treatment **two main effects**
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

The winner gets to choose the song for next class!

Who is the ANOVA champ?

Get ready to compete!

Which effects are significant?



- Condition
- Treatment
- Condition x Treatment
- Condition, Treatment
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

Which effects are significant?

- Condition
- Treatment
- Condition x Treatment
- Condition, Treatment
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

Which effects are significant?

- Condition
- Treatment
- Condition x Treatment
- Condition, Treatment
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

Leaderboard

Which effects are significant?

- Condition
- Treatment
- Condition x Treatment
- Condition, Treatment
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

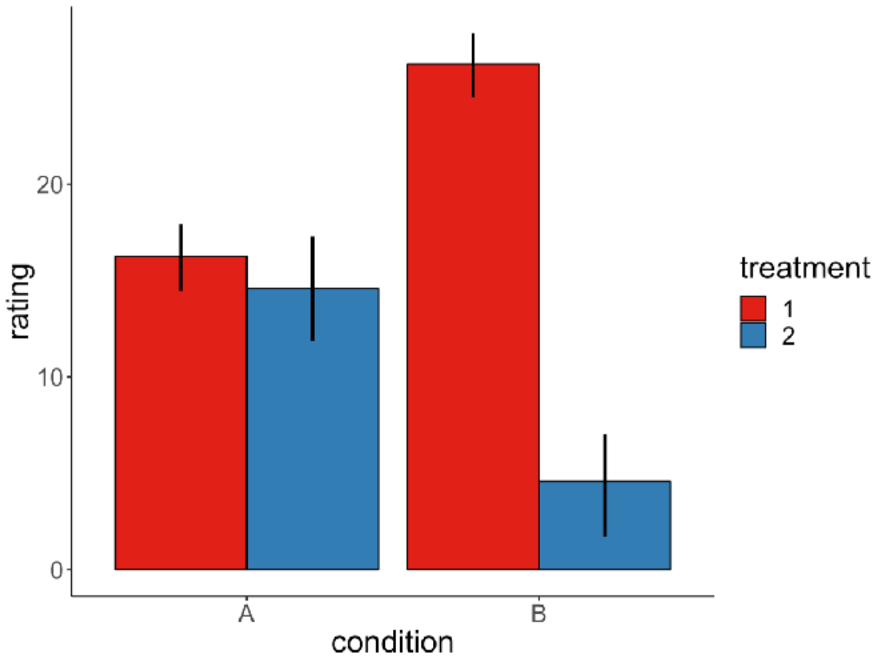
Which effects are significant?

- Condition
- Treatment
- Condition x Treatment
- Condition, Treatment
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

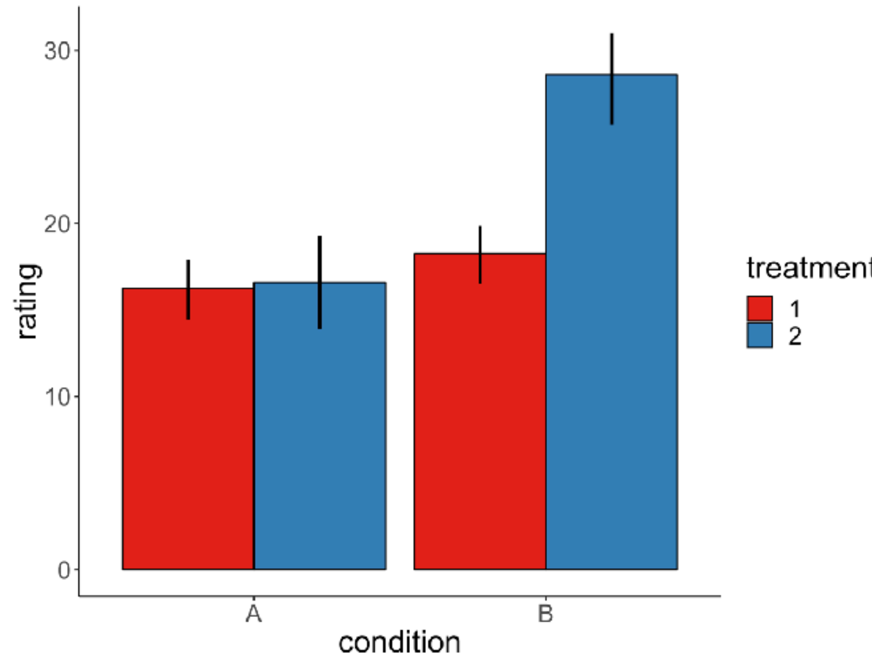
Leaderboard

Solution

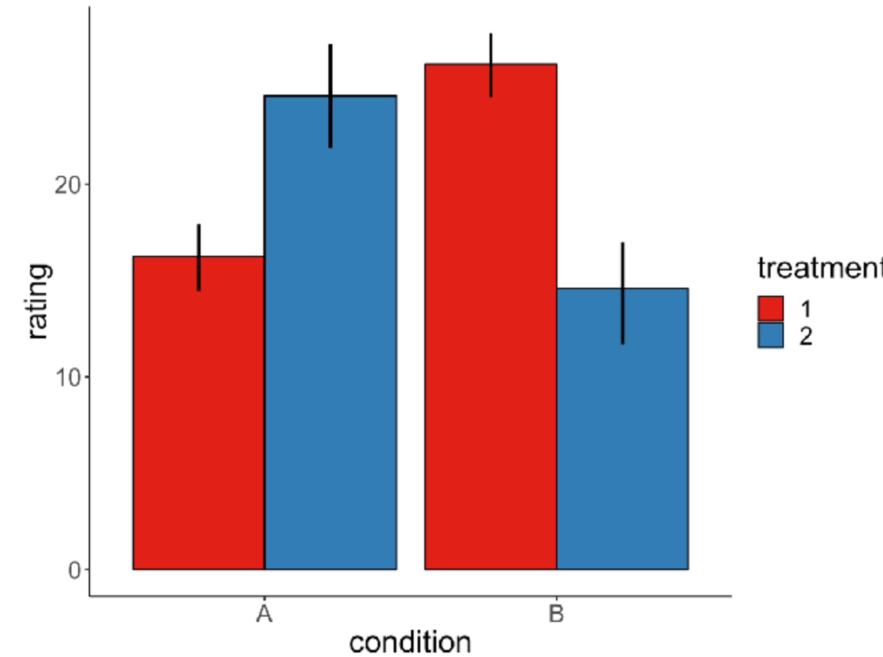
Treatment
Condition x Treatment



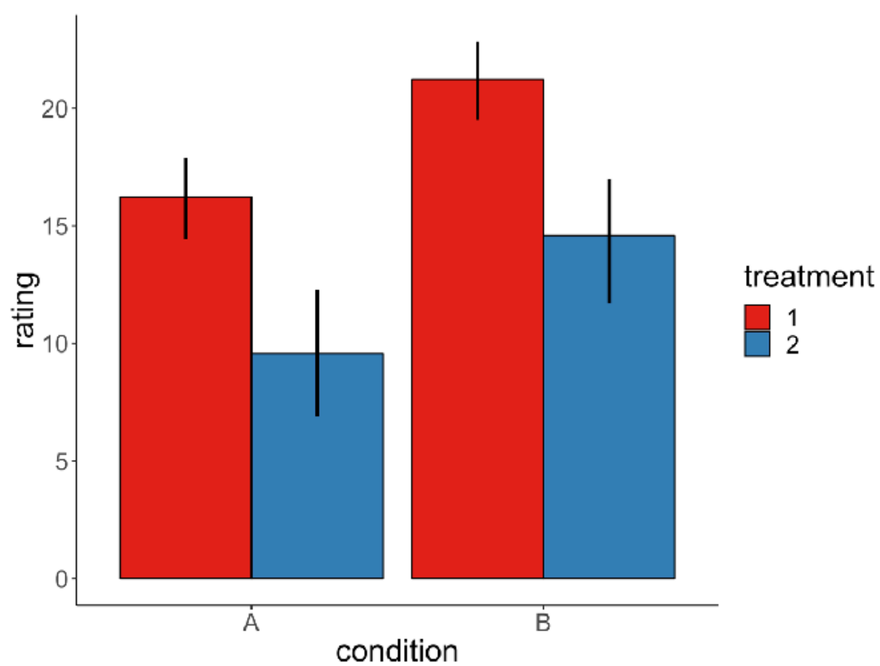
Condition,
Treatment,
Condition x Treatment



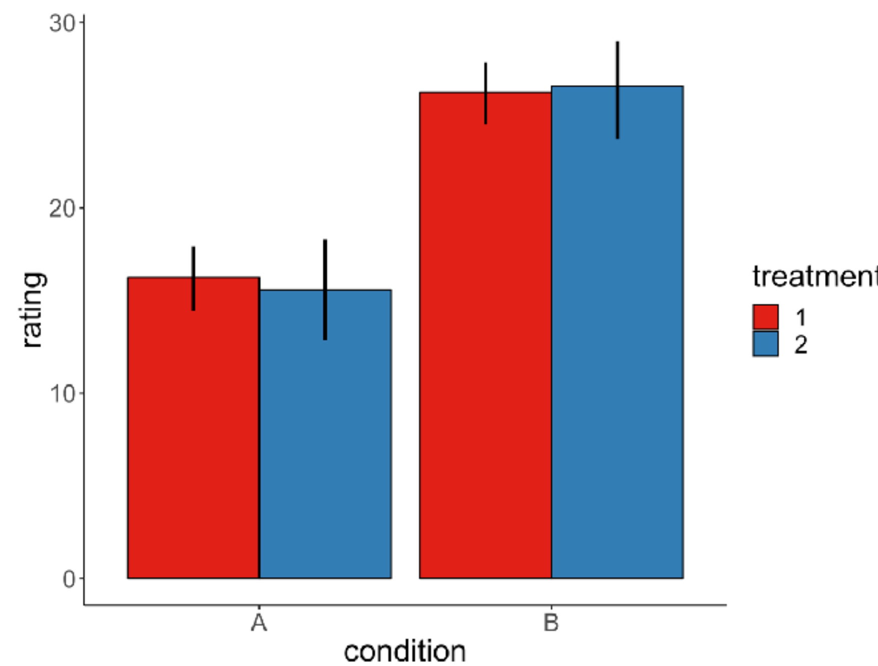
Condition x Treatment



Condition
Treatment



Condition



Unbalanced designs

not the same number of
participants in each cell

ANOVA

- for these examples, I've assumed a balanced design (i.e. the same number of observations in each of the different factor levels)
- things get funky when we have an unbalanced design

https://towardsdatascience.com/anovas-three-types-of-estimating-sums-of-squares-don-t-make-the-wrong-choice-91107c77a27a

Beware of unbalanced designs

```r
lm(formula = balance ~ skill + hand, data = df.poker.unbalanced) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value  Pr(>F)
skill       1   74.3   74.28  4.2904 0.03922 *
hand        2 2385.1 1192.57 68.8827 < 2e-16 ***
Residuals 286 4951.5   17.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

flipped the order

```r
lm(formula = balance ~ hand + skill, data = df.poker.unbalanced) %>%
  anova()
```

```
Analysis of Variance Table

Response: balance
           Df Sum Sq Mean Sq F value Pr(>F)
hand        2 2419.8 1209.92 69.8845 <2e-16 ***
skill       1   39.6   39.59  2.2867 0.1316
Residuals 286 4951.5   17.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The different sums of squares

Two-Way ANOVA is ANOVA with 2 independent variables.

Three different methodologies for splitting variation exist: Type I, Type II and Type III Sums of Squares. They do not give the same result in case of unbalanced data.

Type I, Type II and Type III ANOVA have different outcomes!

https://towardsdatascience.com/anovas-three-types-of-estimating-sums-of-squares-don-t-make-the-wrong-choice-91107c77a27a

Type I Sums of Squares

Type I Sums of Squares are Sequential, so the order of variables in the models makes a difference. This is rarely what we want in practice!

Sums of Squares are Mathematically defined as:

- SS(A) for independent variable A
- SS(B | A) for independent variable B
- SS(AB | B, A) for the interaction effect

caution: this is what `anova()` uses by default

https://towardsdatascience.com/anovas-three-types-of-estimating-sums-of-squares-don-t-make-the-wrong-choice-91107c77a27a

Type II Sums of Squares

Type II Sums of Squares should be used if there is no interaction between the independent variables.

Sums of Squares are Mathematically defined as:

- SS(A | B) for independent variable A
- SS(B | A) for independent variable B
- No interaction effect

however, often not used in practice ...
(mostly because we are interested in interaction effects)

Type III Sums of Squares

The Type III Sums of Squares are also called partial sums of squares again another way of computing Sums of Squares:

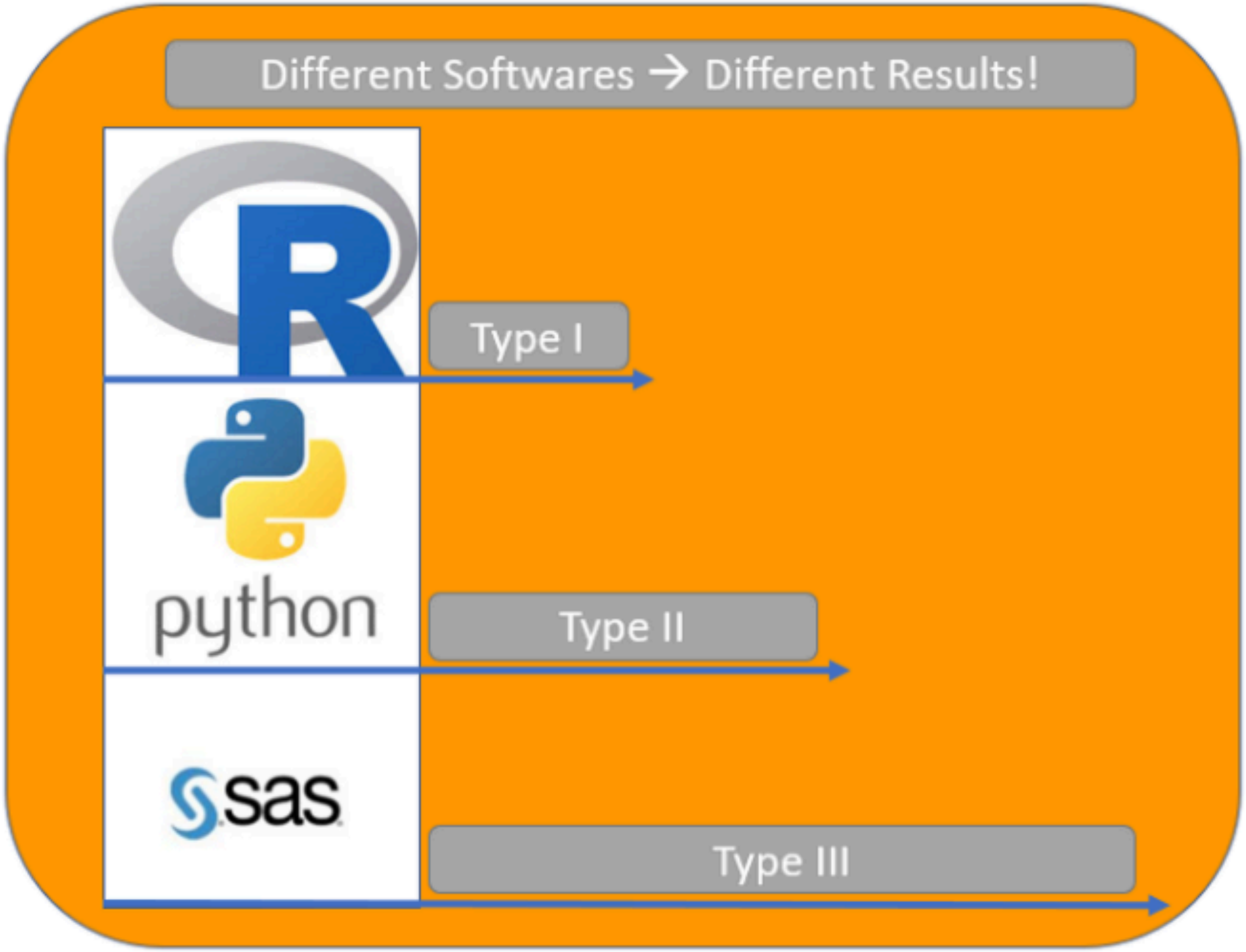
- Like Type II, the Type III Sums of Squares are not sequential, so the order of specification does not matter.
- Unlike Type II, the Type III Sums of Squares do specify an interaction effect.

Sums of Squares are Mathematically defined as:

- SS(A | B, AB) for independent variable A
- SS(B | A, AB) for independent variable B

this is the default in the literature (e.g. SPSS uses it)

Default sums of squares ...



Default Types of Sums of Squares for different programming languages

not great for reproducibility ...

If you want to reproduce SPSS

```r
library("car")

lm(formula = balance ~ hand * skill,
   data = df.poker.unbalanced,
   contrasts = list(hand = "contr.sum",
                    skill = "contr.sum")) %>%
  Anova(type = "3")
```

load the "car" package

set the contrasts

run Anova (capital A) with type "3" for the sum of squares

```
Anova Table (Type III tests)

Response: balance
             Sum Sq  Df   F value Pr(>F)
(Intercept) 27629.1   1 1595.8482 <2e-16 ***
hand         2385.1   2   68.8827 <2e-16 ***
skill          39.6   1    2.2867 0.1316
Residuals    4951.5 286
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you want to reproduce SPSS

```r
library("car")

lm(formula = balance ~ skill * hand,
   data = df.poker.unbalanced,
   contrasts = list(hand = "contr.sum",
                    skill = "contr.sum")) %>%
  Anova(type = "3")
```

now the order doesn't matter ...

```
nova Table (Type III tests)

Response: balance
             Sum Sq  Df   F value Pr(>F)
(Intercept) 27629.1   1 1595.8482 <2e-16 ***
skill          39.6   1    2.2867 0.1316
hand         2385.1   2   68.8827 <2e-16 ***
Residuals    4951.5 286
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you want to reproduce SPSS

```r
library("afex")

fit = aov_ez(id = "participant",
             dv = "balance",
             data = df.poker.unbalanced,
             between = c("hand", "skill"))
fit$Anova
```

```
Contrasts set to contr.sum for the following variables: hand, skill
Anova Table (Type III tests)

Response: dv
             Sum Sq  Df   F value    Pr(>F)    
(Intercept) 27781.3   1 1676.9096 < 2.2e-16 ***
hand         2285.3   2   68.9729 < 2.2e-16 ***
skill          48.9   1    2.9540 0.0867525 .  
hand:skill    246.5   2    7.4401 0.0007089 ***
Residuals    4705.0 284                        
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you want to reproduce SPSS

- There are different kinds of ANOVAs, for which the sums of squares are calculated differently.
- This makes a difference when we have an unbalanced design (i.e. some of the cell sizes are unequal).
- For **unbalanced designs**, make sure to use **type III** sums of squares.
- When interested in interactions, make sure to set the contrasts to **sum contrasts** (= effect coding), rather than using the default dummy coding.

Plan for today

- Interactions
  - one continuous and one binary categorical variable
  - understanding the `lm()` output
- Analysis of Variance (ANOVA)
  - categorical predictor that has more than two levels (One-way ANOVA)
    - follow-up tests
  - multiple categorical predictors (N-way ANOVA)
    - interpreting parameters
    - Who is the ANOVA champ?
  - unbalanced designs

Feedback

How was the pace of today's class?

much too slow

a little too slow

just right

a little too fast

much too fast

How happy were you with today's class overall?

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

Thank you!