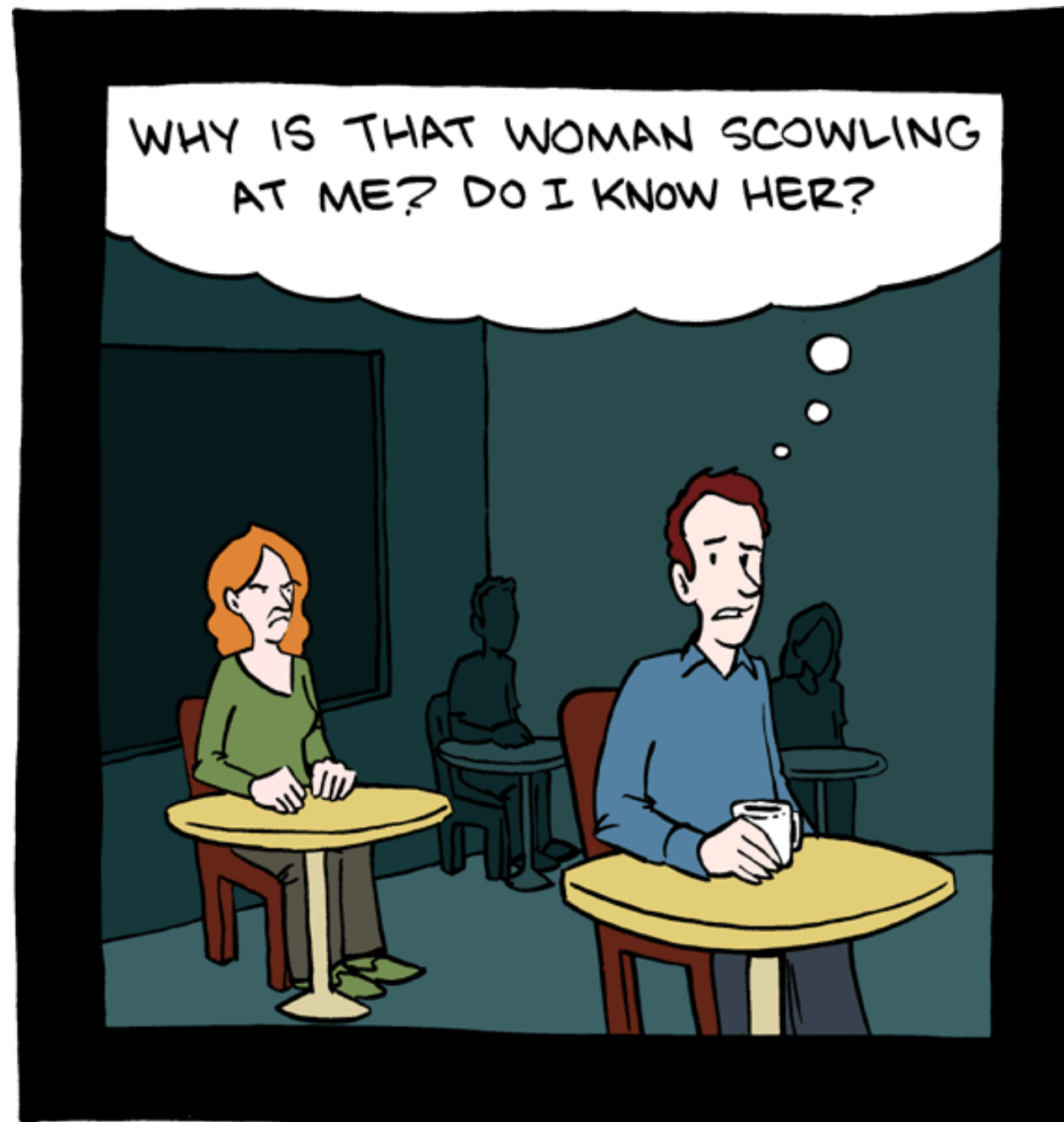


Linear model 3



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

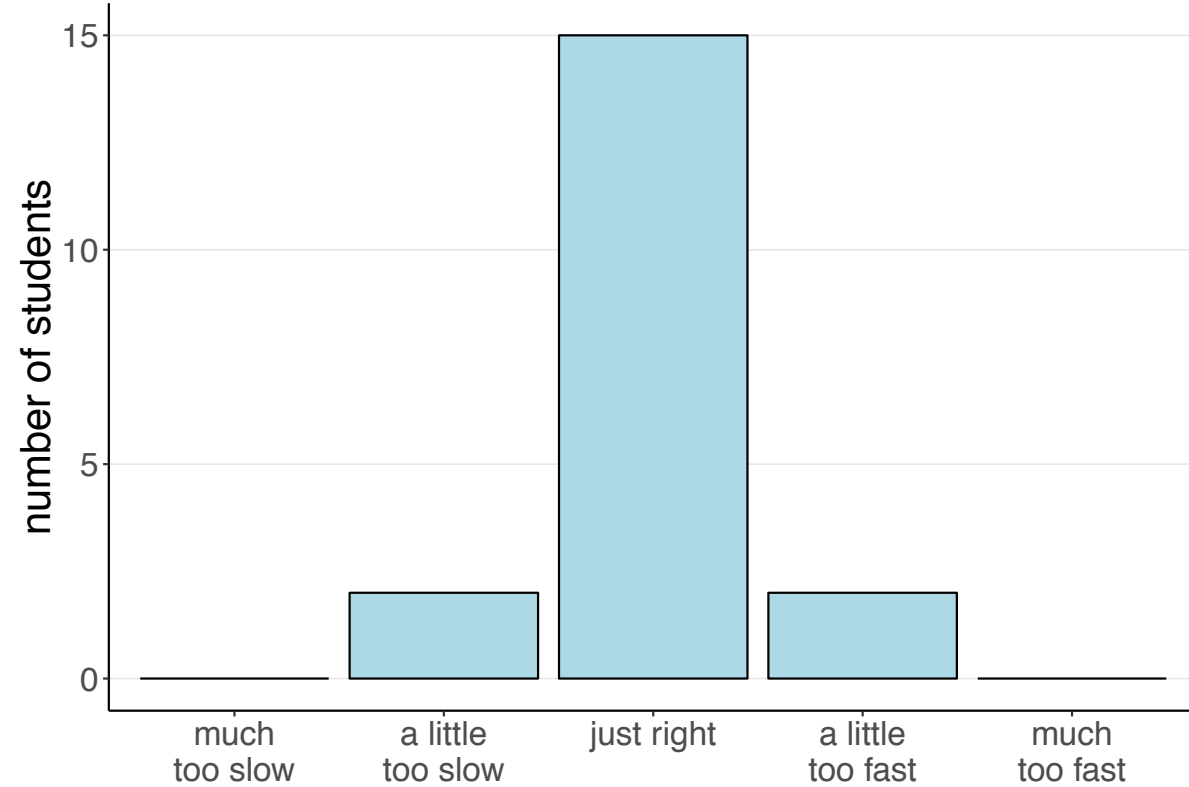
02/04/2019

Logistics

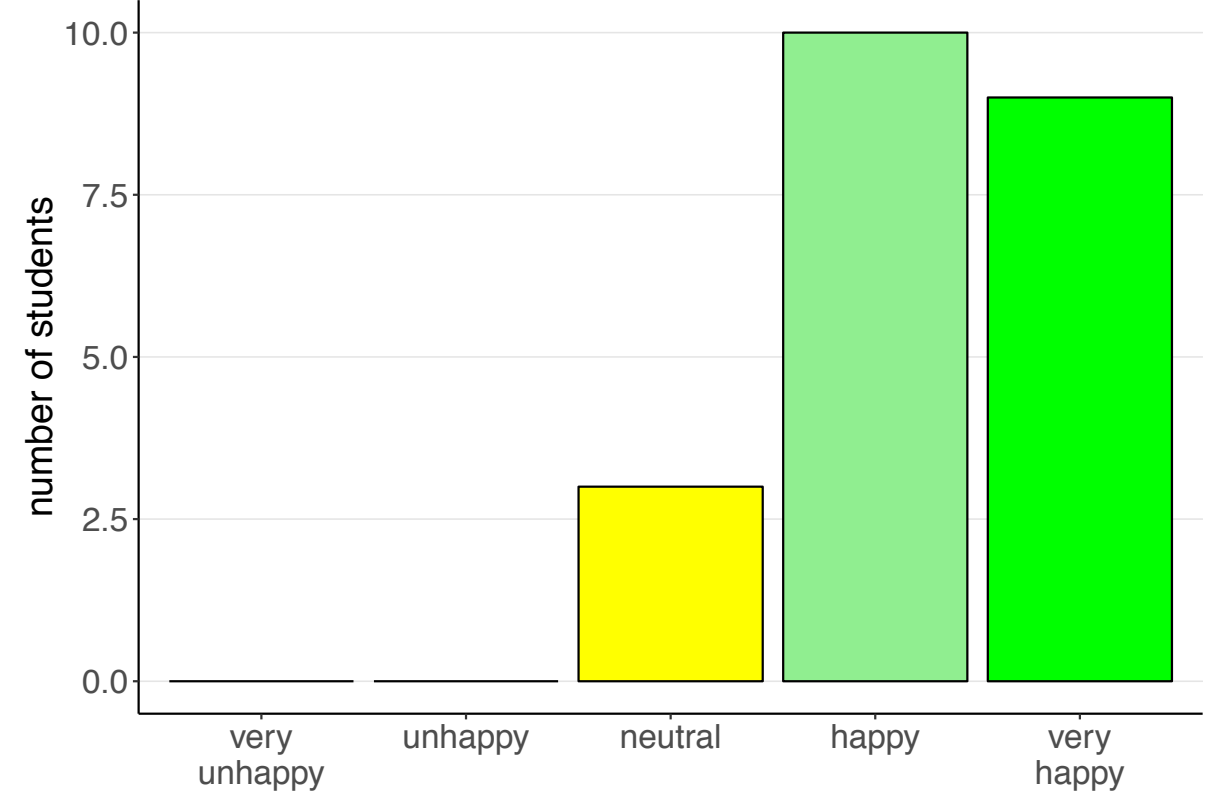
Your feedback

Your feedback

How was the pace of today's class?



How happy were you with today's class overall?



Your feedback

Class was good. Would appreciate a list of ways in which we might misinterpret importance of regression, as well as why we should care.

we'll talk about problem cases on Wednesday

Homework 4

~~Efficient estimator of the population variance~~

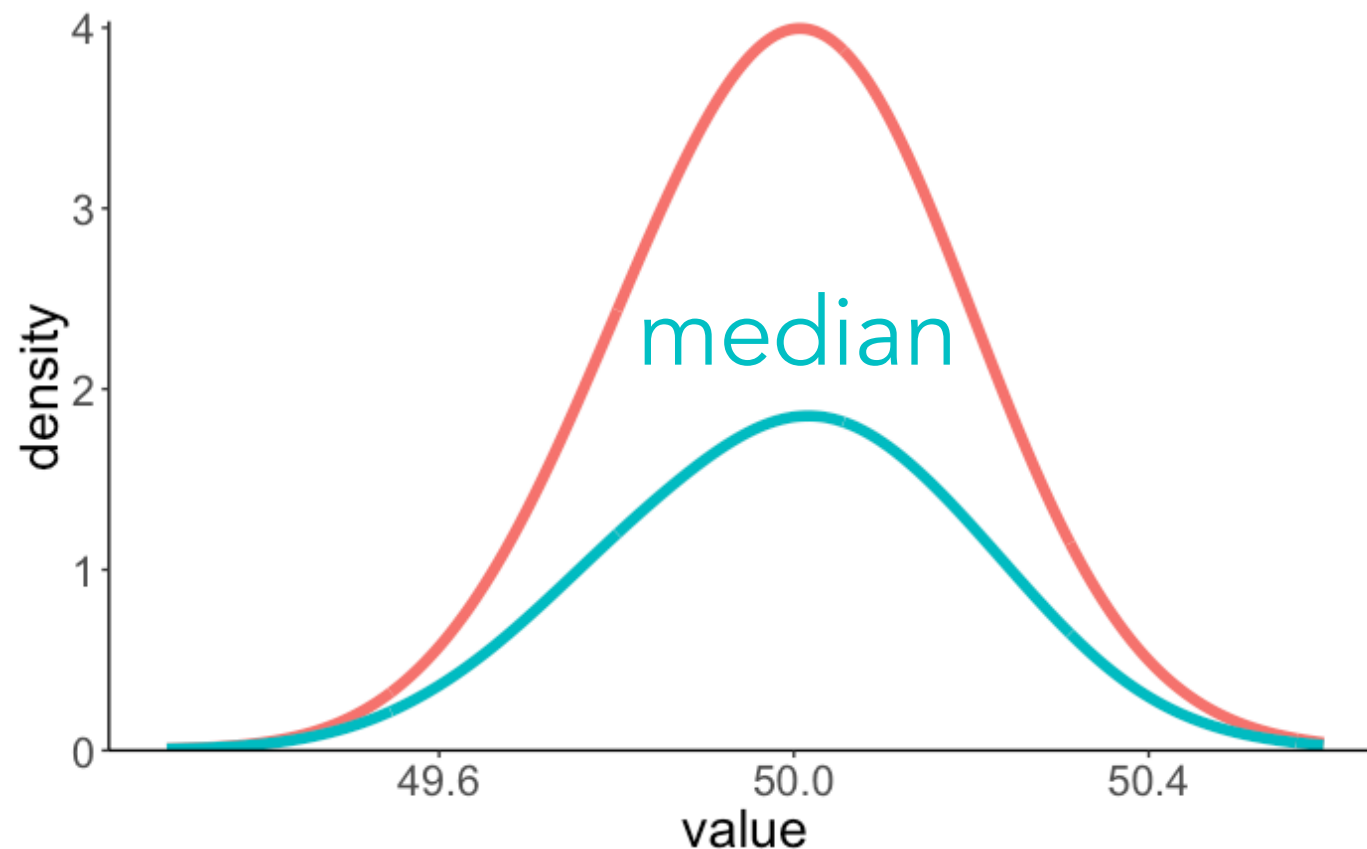
~~Show that the sample variance is an efficient estimator of the population variance. Compare the sample variance $s^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n-1}$ with the *median absolute deviation* as an estimator of the population variance. Use the same procedure as above when you tried to figure out whether s^2 was an unbiased estimator.~~

population distribution

$$Y_i = 50 + \epsilon$$

mean

median



But ...

The *median absolute deviation* (MAD) is a biased estimator of the population standard deviation. It's even a biased estimator of the population MAD. So comparing efficiency doesn't make that much sense ...

Thanks to Mark Roman Miller!!

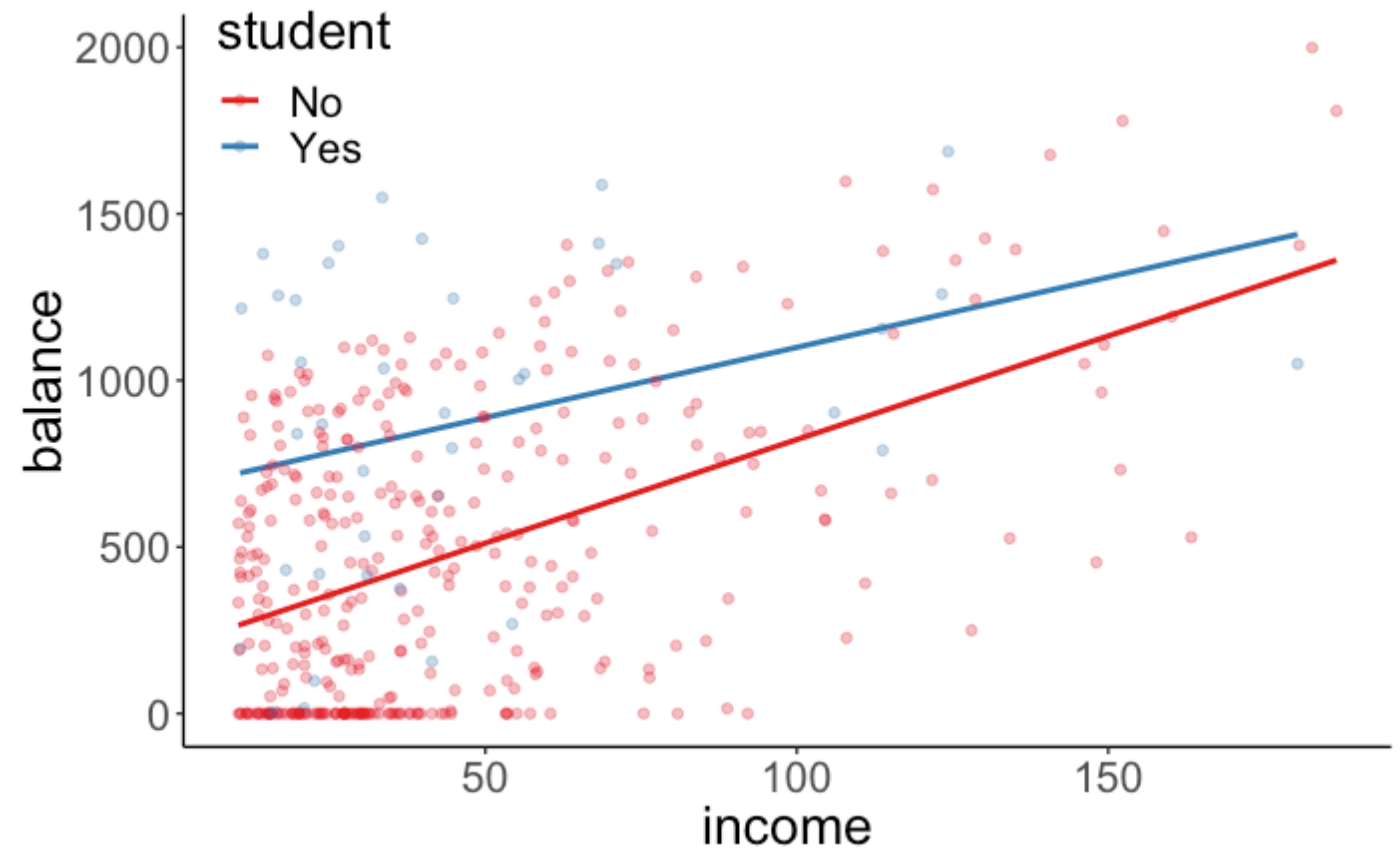
Things that came up in class

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

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

```
1 fit_c = lm(formula = balance ~ student + income:student, data = df.credit)
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)
3
4 anova(fit_c, fit_a)
```

```
1 fit_c = lm(formula = balance ~ income + student, data = df.credit)
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)
3
4 anova(fit_c, fit_a)
```

lm() output

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

```
1 fit_c = lm(formula = balance ~ 1, data = df.credit)  
2 fit_a = lm(formula = balance ~ income + student + income:student, data = df.credit)  
3  
4 anova(fit_c, fit_a)
```

lm() output

```
1 lm(formula = balance ~ income + student + income:student, data = df.credit) %>%  
2 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
- 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

Another clarification

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

Explicitly encode the interaction

```
1 df.credit %>%  
2   mutate(student_dummy = ifelse(student == "No", 0, 1)) %>%  
3   mutate(income_student = income * student_dummy) %>%  
4   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!

And now for something completely different ...



How the BBC Visual and Data Journalism team works with graphics in R

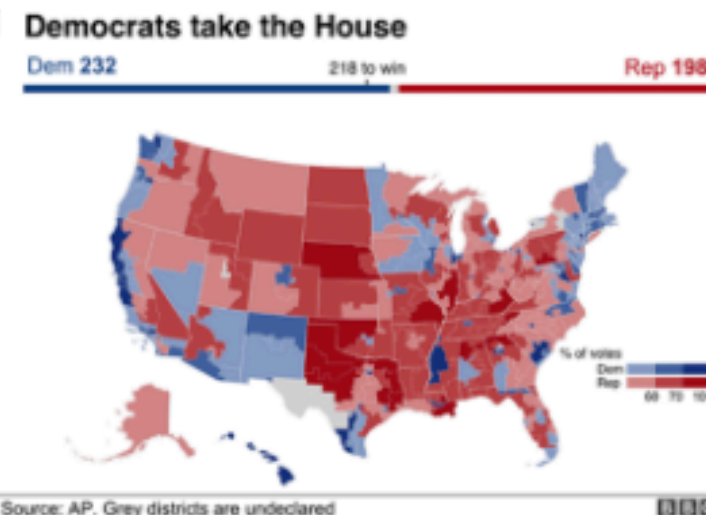
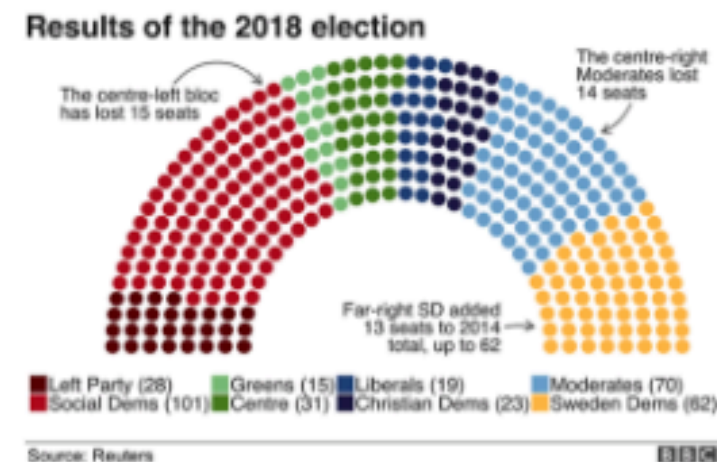
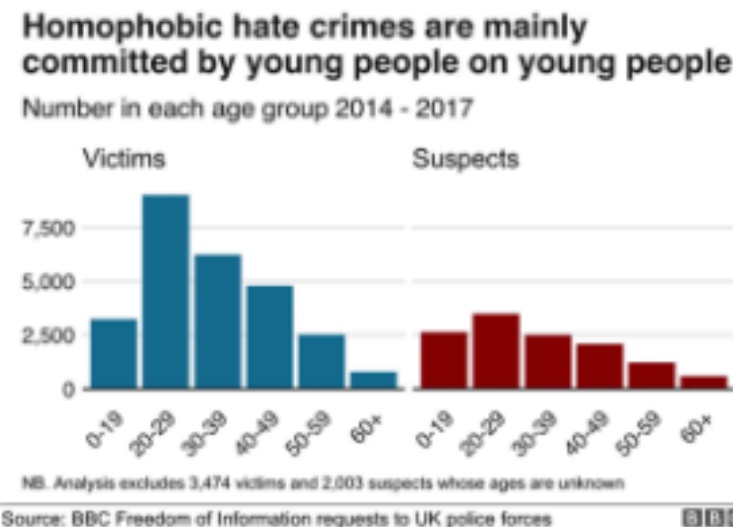
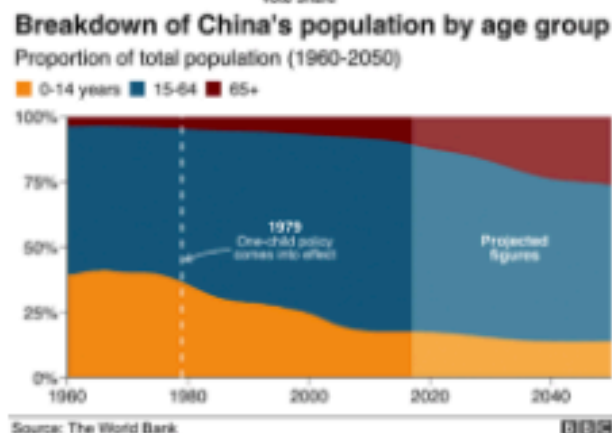
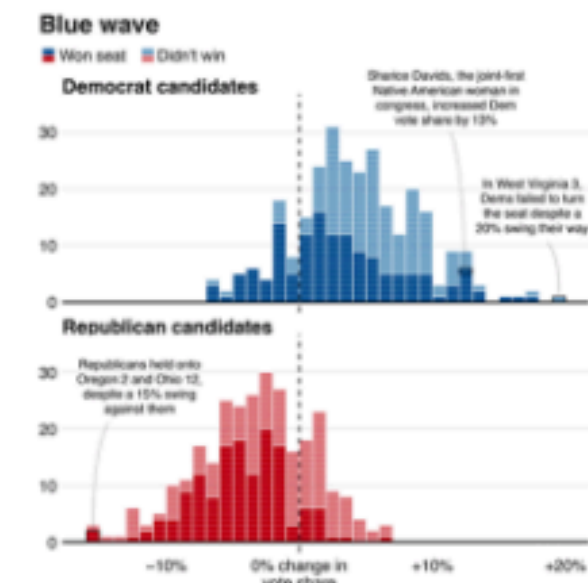
the BBC uses `ggplot2`



BBC Visual and Data Journalism

[Follow](#)

Feb 1 · 8 min read



R Markdown doc!

bbc / bbplot

Code Issues 2 Pull requests 2 Projects 0 Wiki Insights

R package that helps create and export ggplot2 charts in the style used by the BBC News data team

12 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

nassosstilianou adds link to Github pages Latest commit 83abfe3 18 days ago

R pushes repo to Github 19 days ago

chart_examples updates chart examples 19 days ago

data pushes repo to Github 19 days ago

man pushes repo to Github 19 days ago

README.md

BBPLOT

This repo contains the functions of the `bbplot` package, which once installed locally, provides helpful functions for creating and exporting graphics made in ggplot in the style used by the BBC News data team.

Blue wave

Won seat Didn't win

Democrat candidates

Sharice Davis, the joint first Native American woman in congress, increased Dem vote share by 13%

In West Virginia 3, Dems failed to turn the seat despite a 20% swing their way

Republican candidates

Republicans held onto Oregon 2 and Ohio 12, despite a 15% swing against them

Where penalties are saved

World Cup shootout misses and saves, 1982-2014

Roberto Baggio's penalty miss in the 1994 final against Brazil

MPs rejected Theresa May's deal by 230 votes

For (119) Against (249)

Earnings vary across units even within subjects

Impact on men's earnings relative to the average degree

Fast-growing cities face worse climate risks

Population growth 2018-2035 over climate change vulnerability

Africa Asia Americas Europe Oceania

Less vulnerable More vulnerable

Source: AP, 19/01 ET

How to create BBC style graphics

- Make a line chart
- Make a multiple line chart
- Make a bar chart
- Make a stacked bar chart
- Make a grouped bar chart
- Make a dumbbell chart
- Make a histogram
- Make changes to the legend
- Make changes to the axes
- Add annotations
- Work with small multiples
- Do something else entirely

BBC Visual and Data Journalism cookbook for R graphics

Last updated: 2019-01-24

How to create BBC style graphics

At the BBC data team, we have developed an R package and an R cookbook to make the process of creating publication-ready graphics in our in-house style using R's ggplot2 library a more reproducible process, as well as making it easier for people new to R to create graphics.

The cookbook below should hopefully help anyone who wants to make graphics like these:

Blue wave

Won seat Didn't win

Democrat candidates

Sharice Davis, the joint first Native American woman in congress, increased Dem vote share by 13%

In West Virginia 3, Dems failed to turn the seat despite a 20% swing their way

Republican candidates

Republicans held onto Oregon 2 and Ohio 12, despite a 15% swing against them

Where penalties are saved

World Cup shootout misses and saves, 1982-2014

Roberto Baggio's penalty miss in the 1994 final against Brazil

MPs rejected Theresa May's deal by 230 votes

For (119) Against (249)

Earnings vary across units even within subjects

Impact on men's earnings relative to the average degree

Fast-growing cities face worse climate risks

Population growth 2018-2035 over climate change vulnerability

Africa Asia Americas Europe Oceania

Less vulnerable More vulnerable

Source: AP, 19/01 ET

We'll get to how you can put together the various elements of these graphics, but let's get the admin out of the way first...

Load all the libraries you need

A few of the steps in this cookbook - and to create charts in R in general - require certain packages to be installed and loaded. So that you do not have to install and load them one by one, you can use the `p_load` function in the `pacman` package to load them all at once with the following code.

```
#This line of code installs the pacman page if you do not have it installed - if you do, it s
imply loads the package
if(!require(pacman))install.packages("pacman")
```

<https://github.com/bbc/bbplot>

<https://bbc.github.io/rcookbook/>

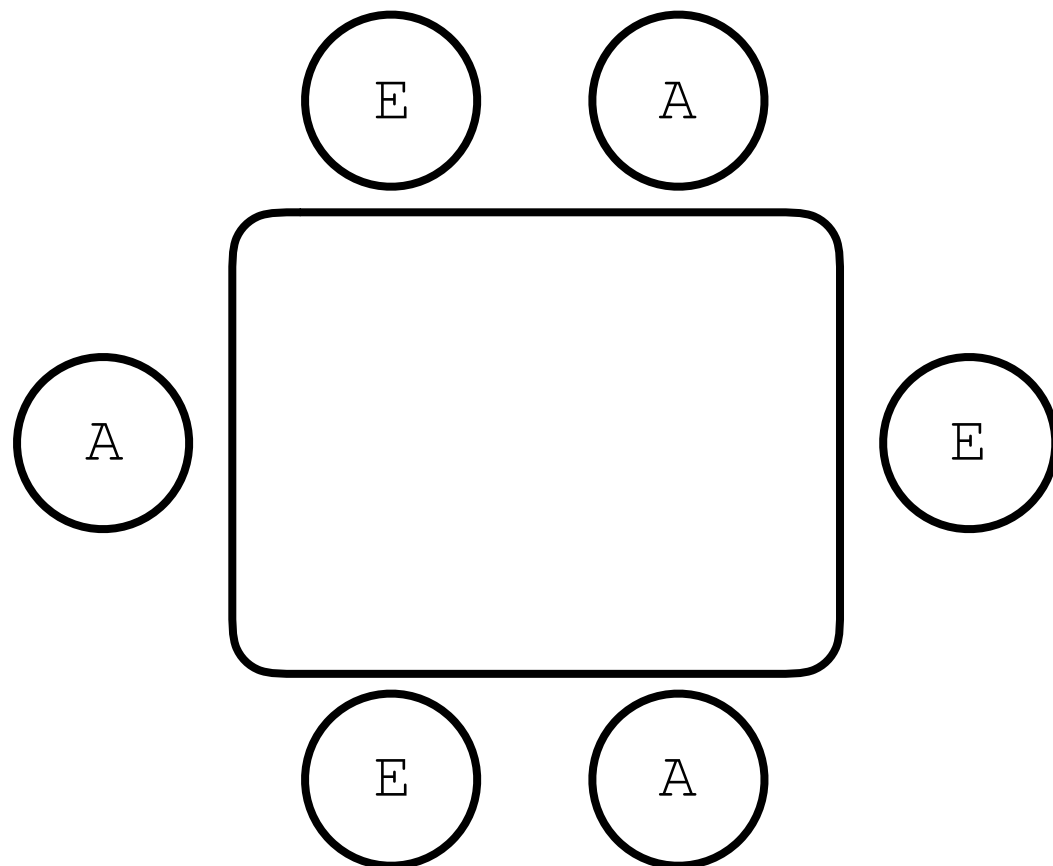
Plan for today

- Linear model with ...
 - categorical predictor that has more than two levels (One-way ANOVA)
 - multiple categorical predictors (N-way ANOVA)
 - dummy coding vs. effect coding
 - planned comparisons

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*

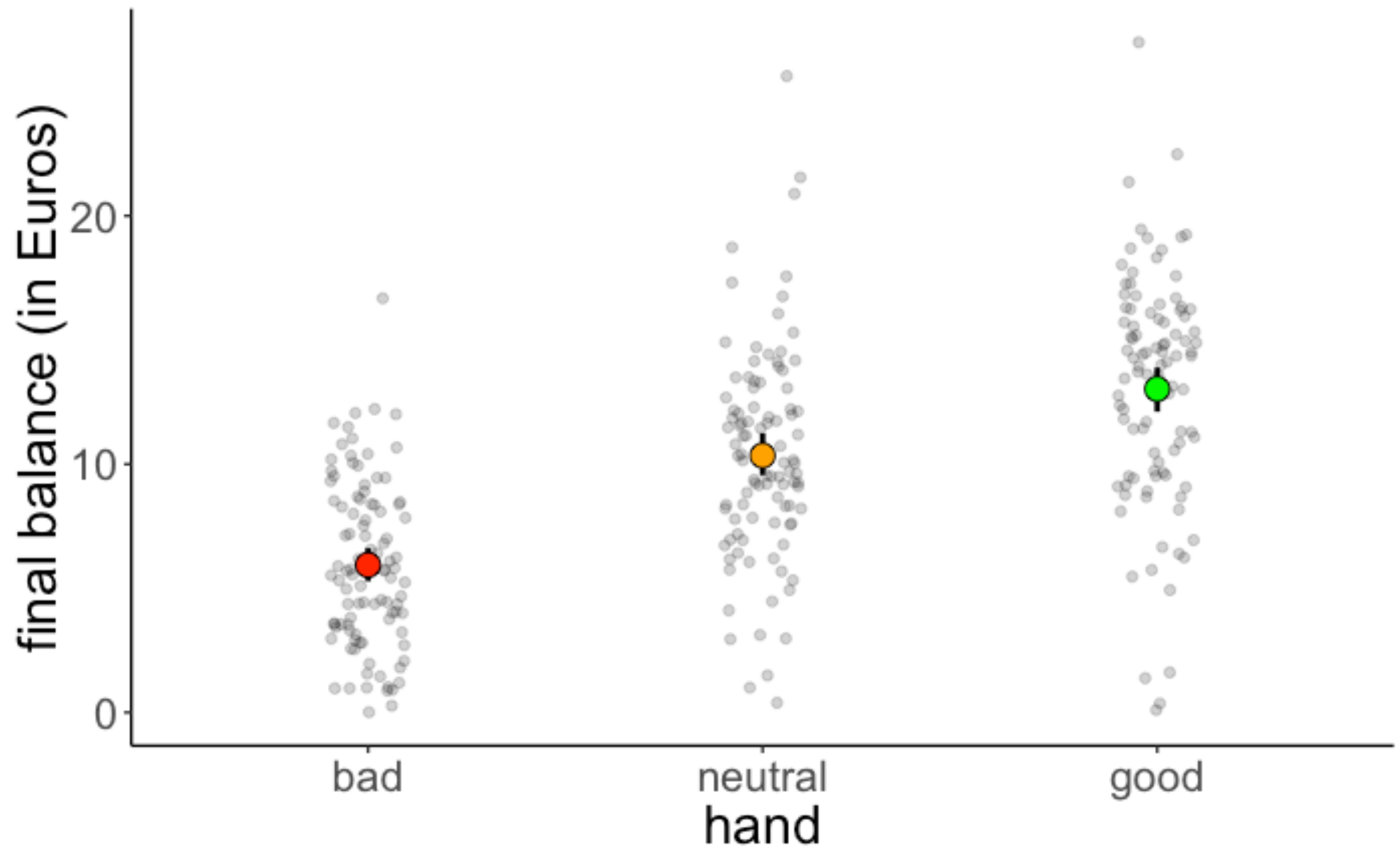
Categorical predictor with more than two levels

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



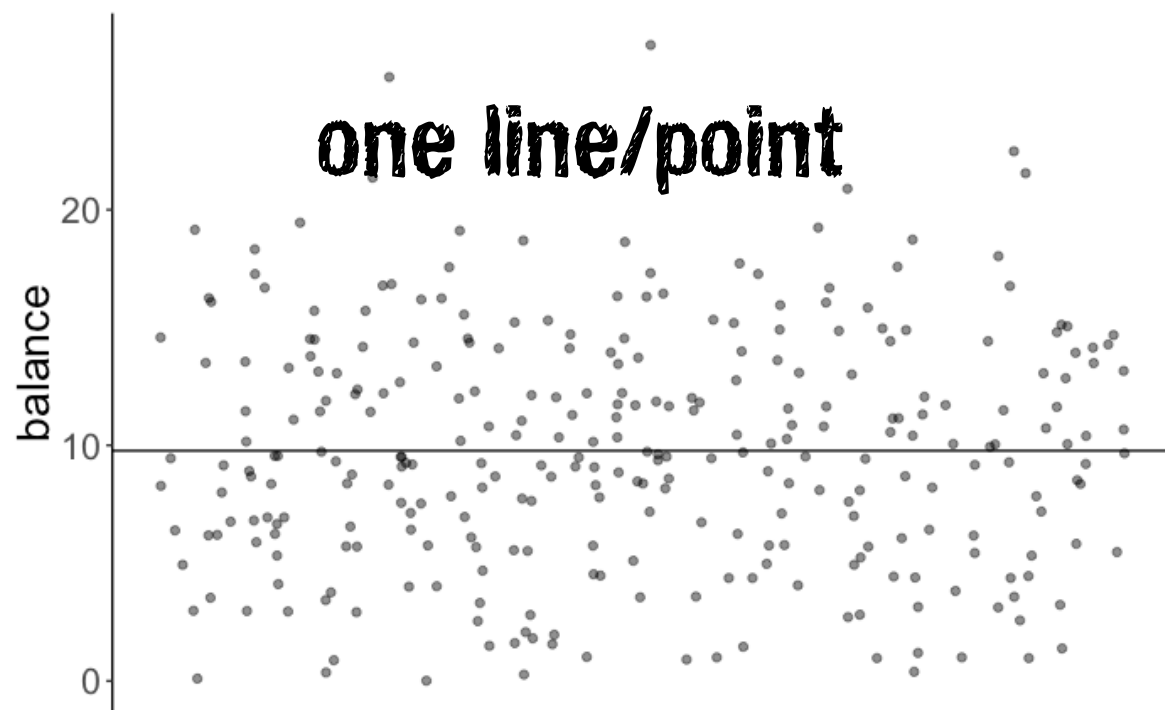
H_0 : Card quality does not affect the final balance.

Model C

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

Model prediction

one line/point



Fitted model

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

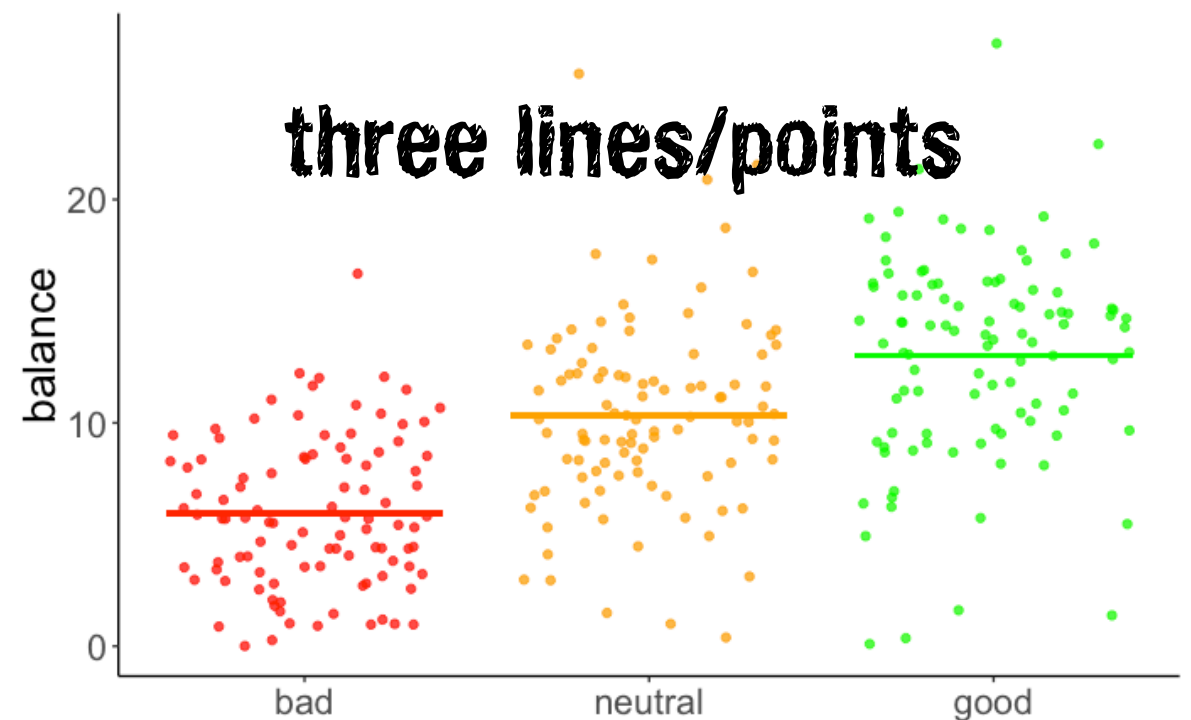
H_1 : Card quality affects the final balance.

Model A

$$\text{balance}_i = \beta_0 + \beta_1 \text{hand_neutral}_i + \beta_2 \text{hand_good}_i + \epsilon$$

Model prediction

three lines/points



Fitted model

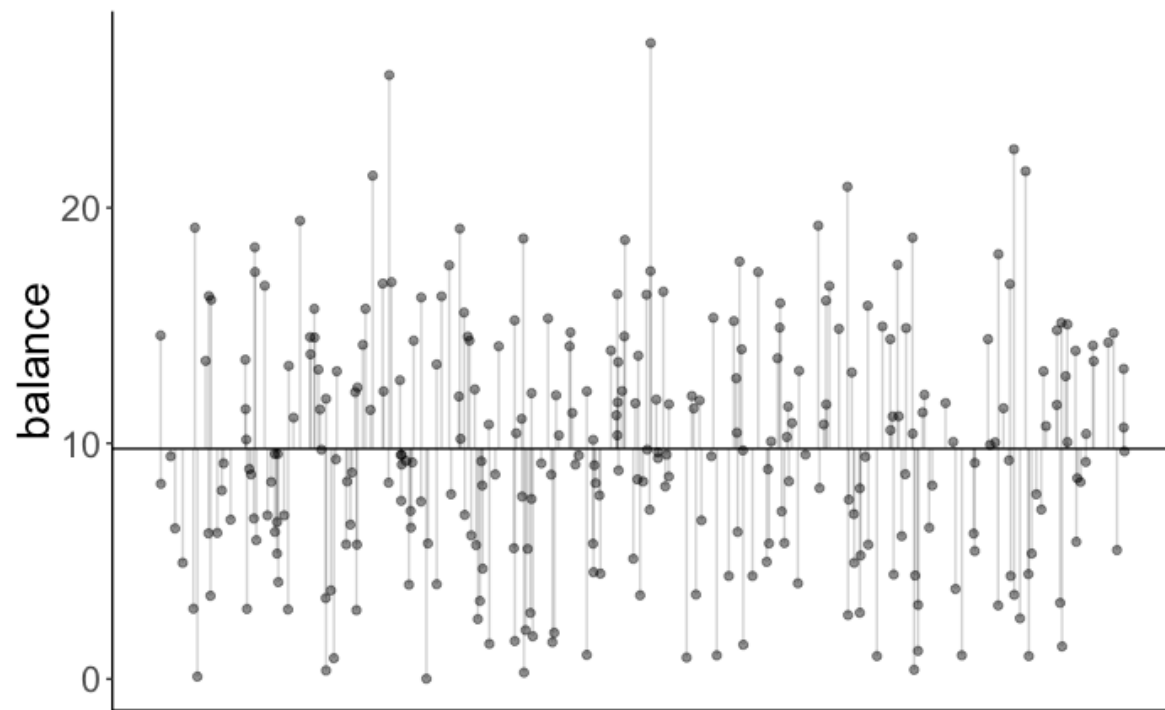
$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand_neutral}_i + 7.08 \cdot \text{hand_good}_i$$

H_0 : Card quality does not affect the final balance.

Model C

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

Model prediction



$$\text{SSE}(C) = 7580$$

Fitted model

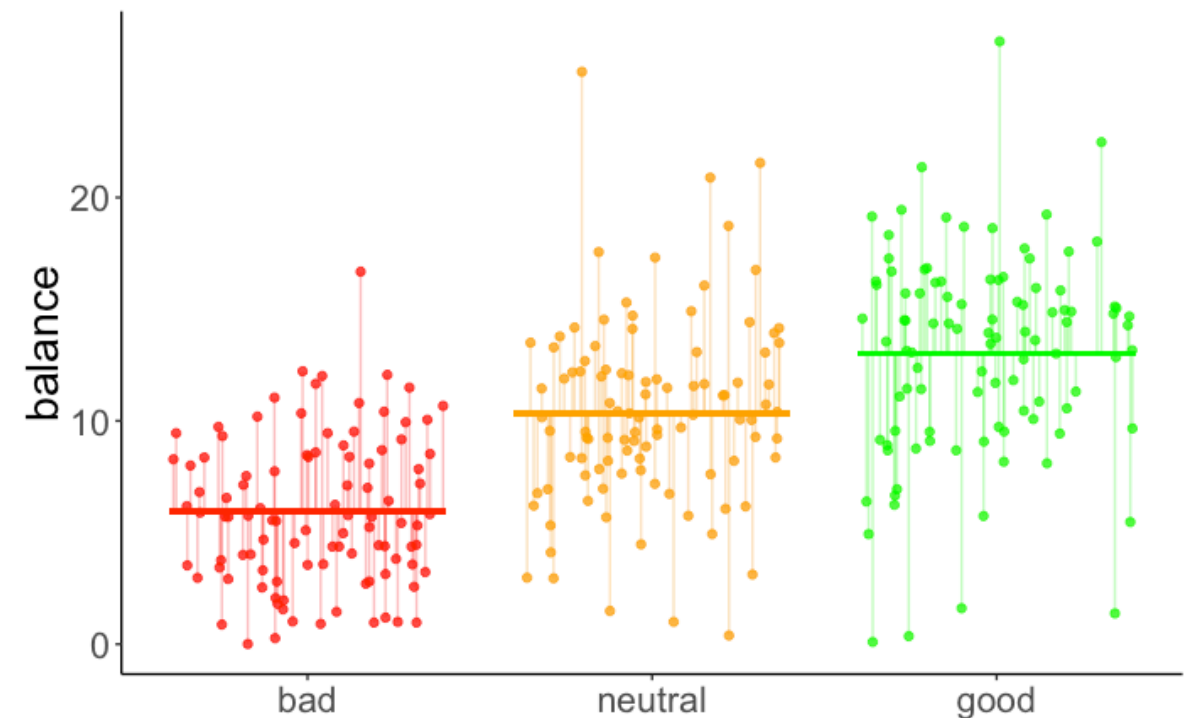
$$\widehat{\text{balance}}_i = 9.77$$

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



$$\text{SSE}(A) = 5021$$

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?

$$\text{SSE}(C) = 7580$$

$$\text{SSE}(A) = 5021$$

$$\text{PRE} = 1 - \frac{\text{SSE}(A)}{\text{SSE}(C)} \quad \text{worth it?}$$

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

```
1 # fit the models
2 fit_c = lm(formula = balance ~ 1, data = df.poker)
3 fit_a = lm(formula = balance ~ hand, data = df.poker)
4
5 # compare via F-test
6 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

```
1 df.poker %>%  
2   mutate(hand_neutral = ifelse(hand == "neutral", 1, 0),  
3         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

same same,
but different

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

```
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":

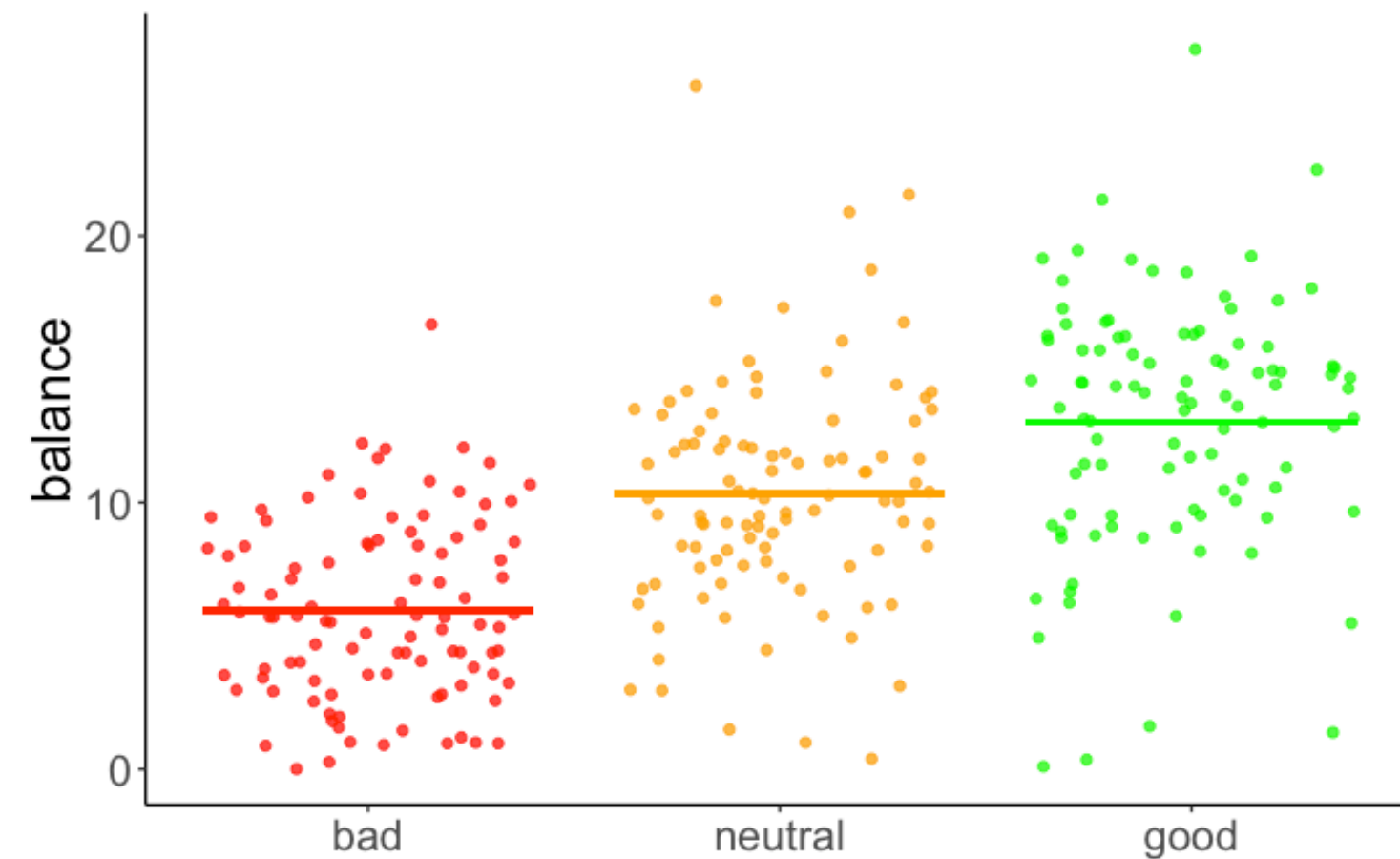
$$\widehat{\text{balance}}_i = 5.94 + 7.08 = 13.02$$

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":

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

Follow-up tests

Asking more specific questions

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

```
1 df.poker %>%
2   filter(hand %in% c("bad", "neutral")) %>%
3   lm(formula = balance ~ hand,
4       data = .) %>%
5   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: **What does this summary not tell us?**

	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

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?

```
1 df.poker %>%
2   filter(hand %in% c("neutral", "good")) %>%
3   lm(formula = balance ~ hand,
4       data = .) %>%
5   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?

```
1 df.poker %>%
2   mutate(hand = fct_relevel(hand, "neutral")) %>%
3   lm(formula = balance ~ hand,
4     data = .) %>%
5   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
```

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

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

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

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

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?

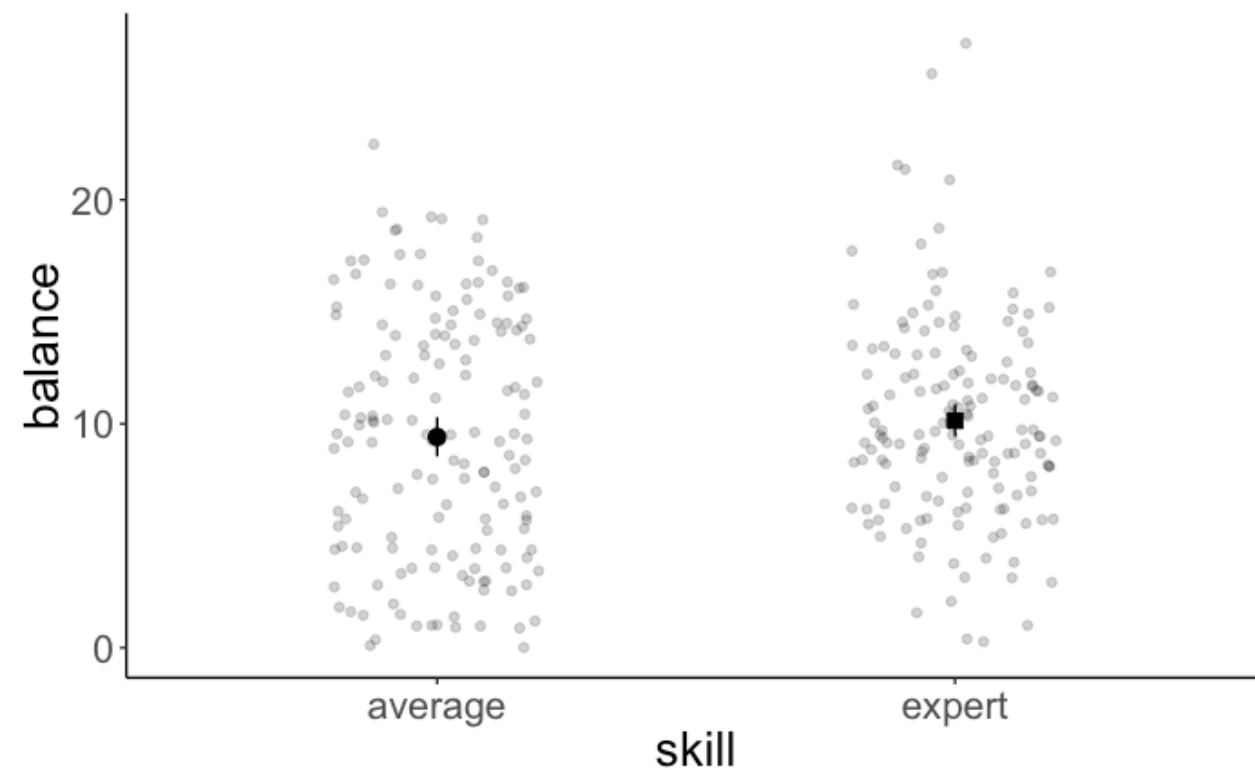
One testing whether skill level affects the final balance,
and one testing whether quality of cards affects the final balance?

Interested in interactions!

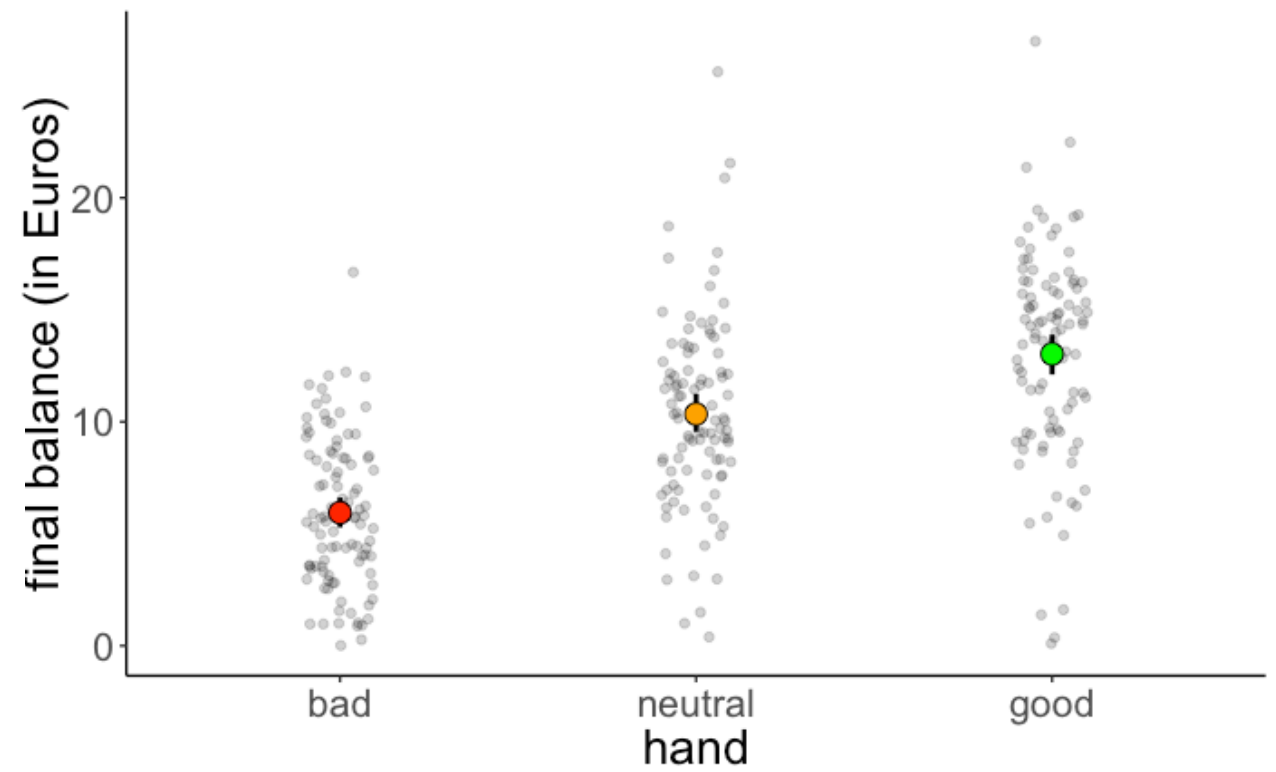
Does the effect of one variable depend on the other?

Visualize the data

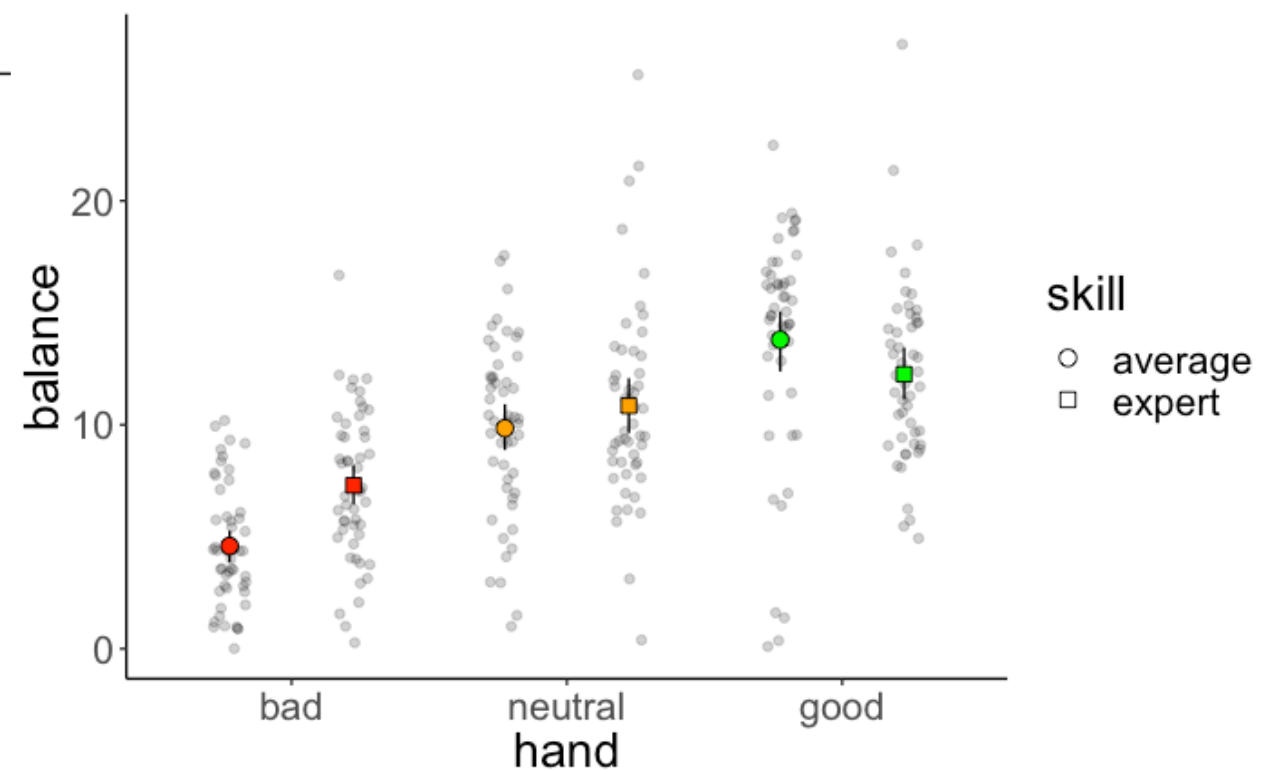
Skill



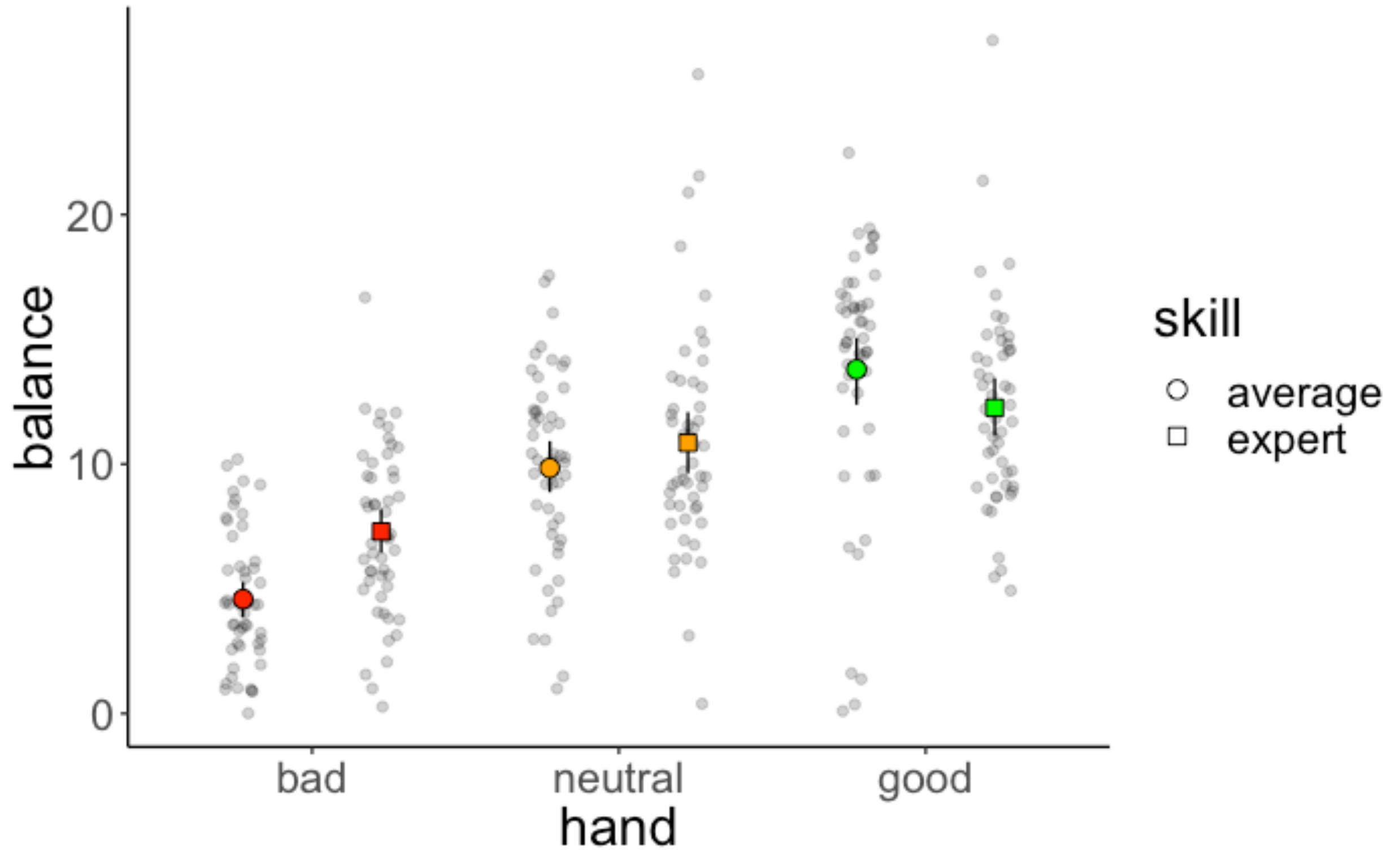
Quality of hand



Quality of hand & Skill



Visualize the data



Fit a model

```
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

$$\begin{aligned} \widehat{\text{balance}}_i &= b_0 + b_2 \cdot \text{hand_good}_i + b_3 \cdot \text{skill_expert}_i + b_5 \cdot \text{hand_good:skill_expert}_i \\ &= 12.26 \end{aligned}$$

Analysis of variance

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

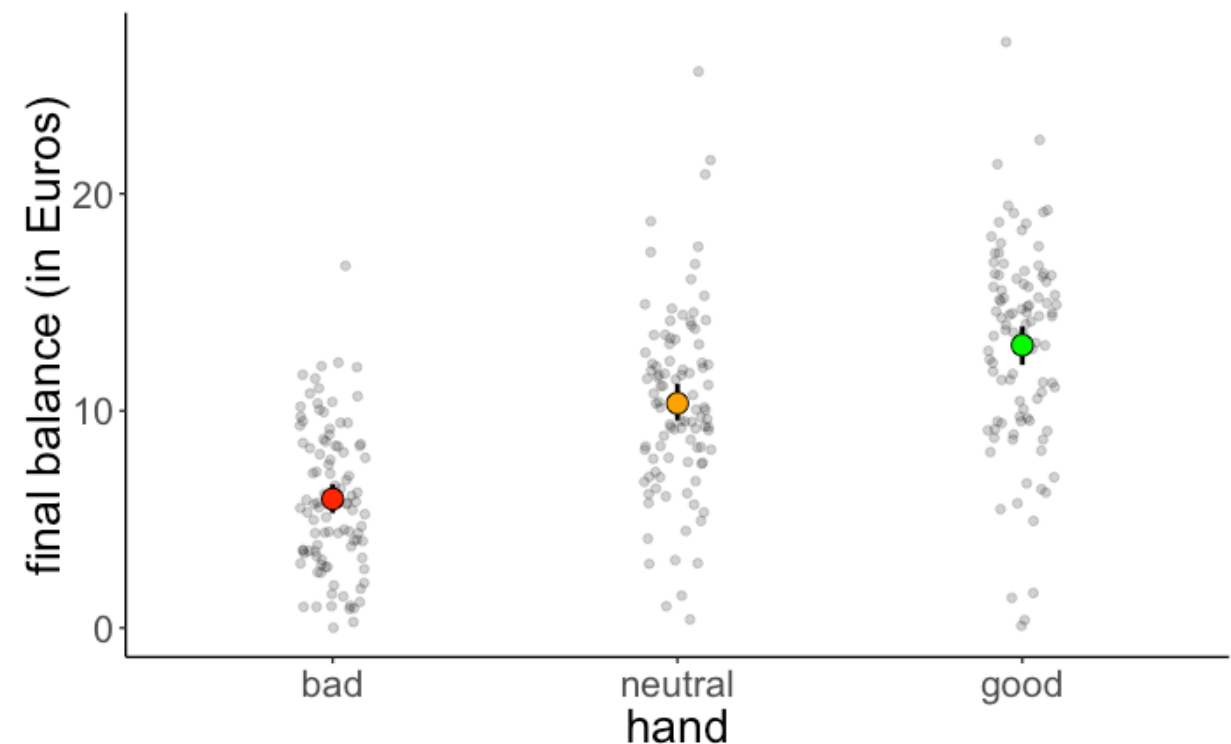
Analysis of Variance Table

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

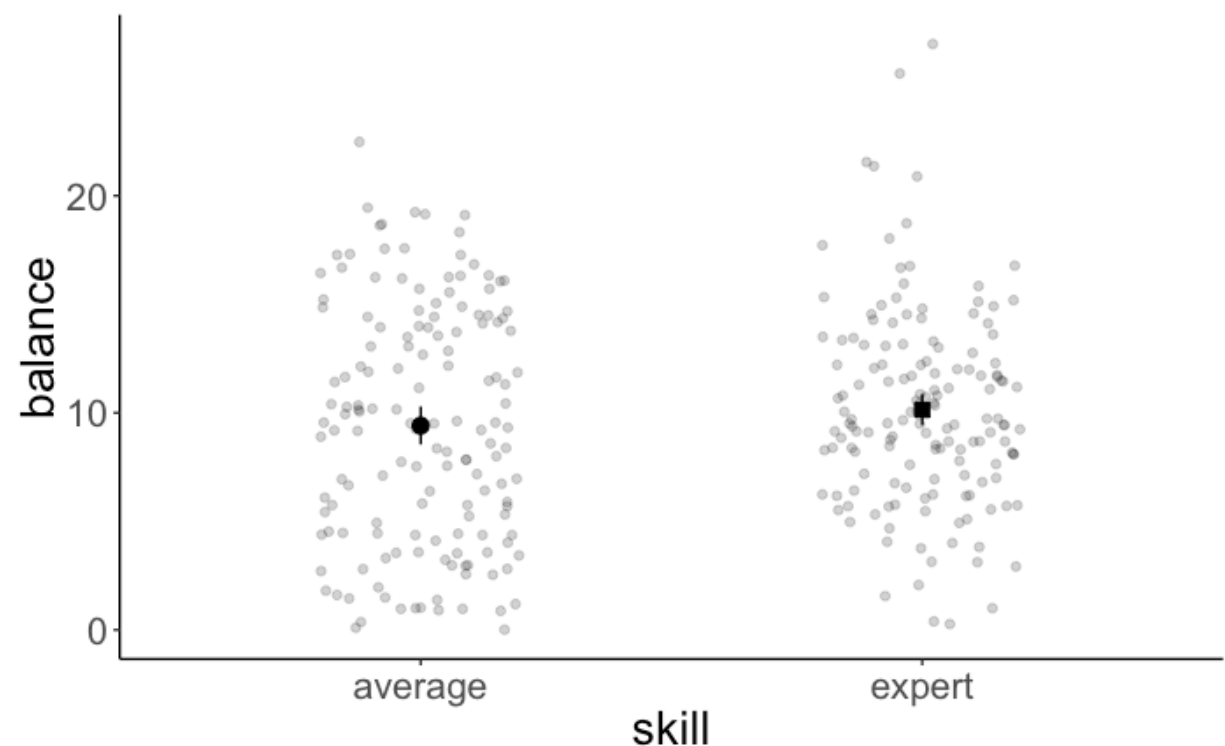
Response: balance

	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

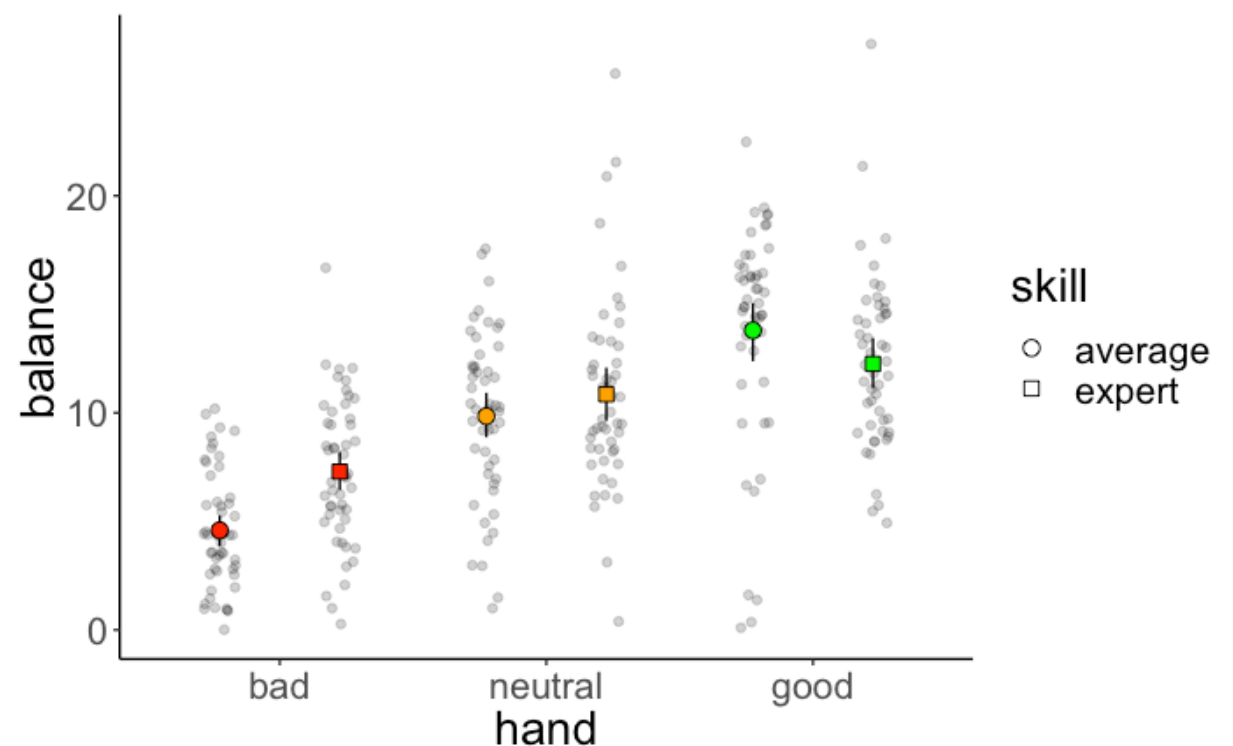
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
hand	2	2559.4	1279.70	79.1692	< 2.2e-16	***
skill	1	39.3	39.35	2.4344	0.1197776	
hand:skill	2	229.0	114.49	7.0830	0.0009901	***
Residuals	294	4752.3	16.16			

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

main effect of hand

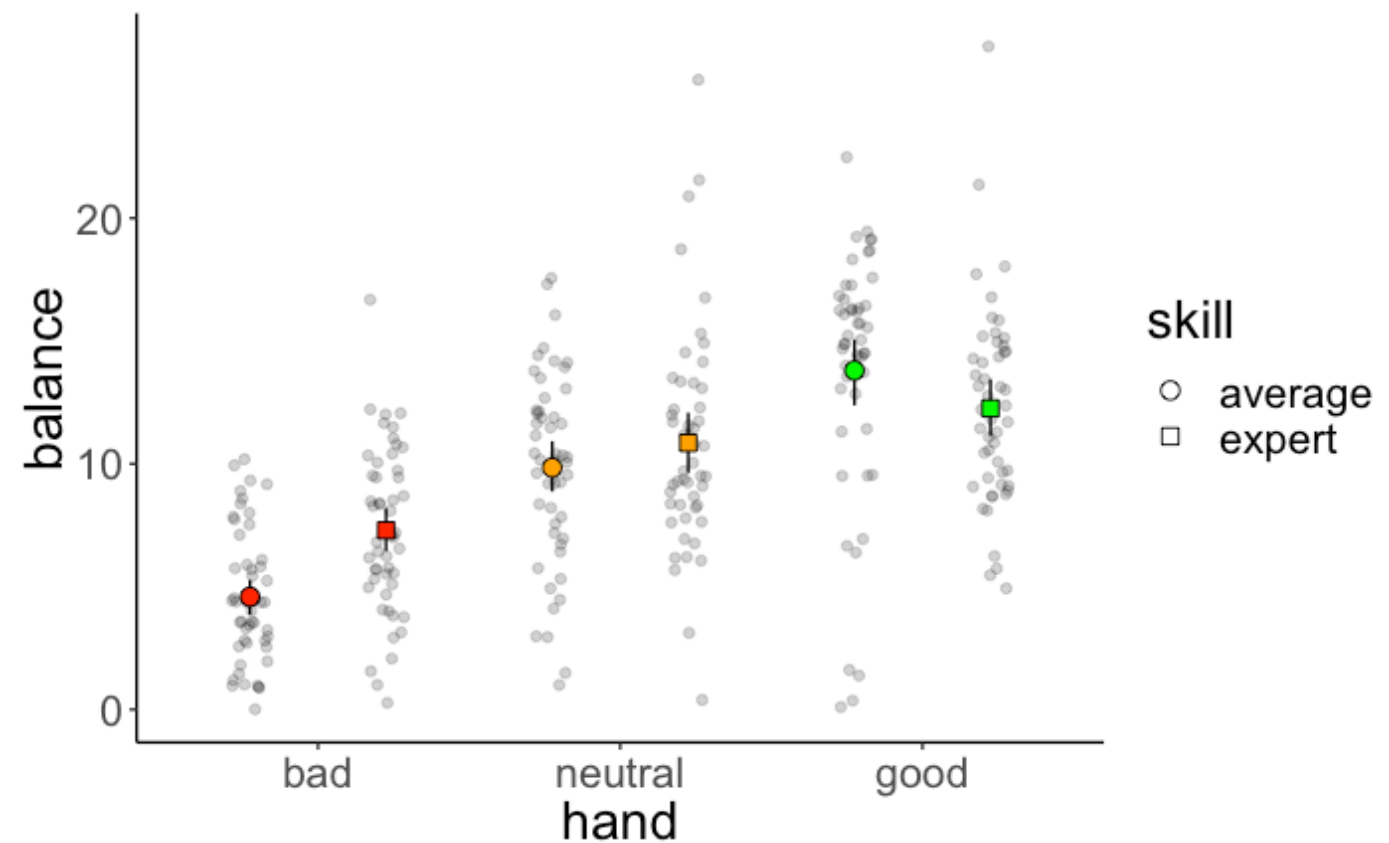
no main effect of skill

interaction between hand
and skill



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.

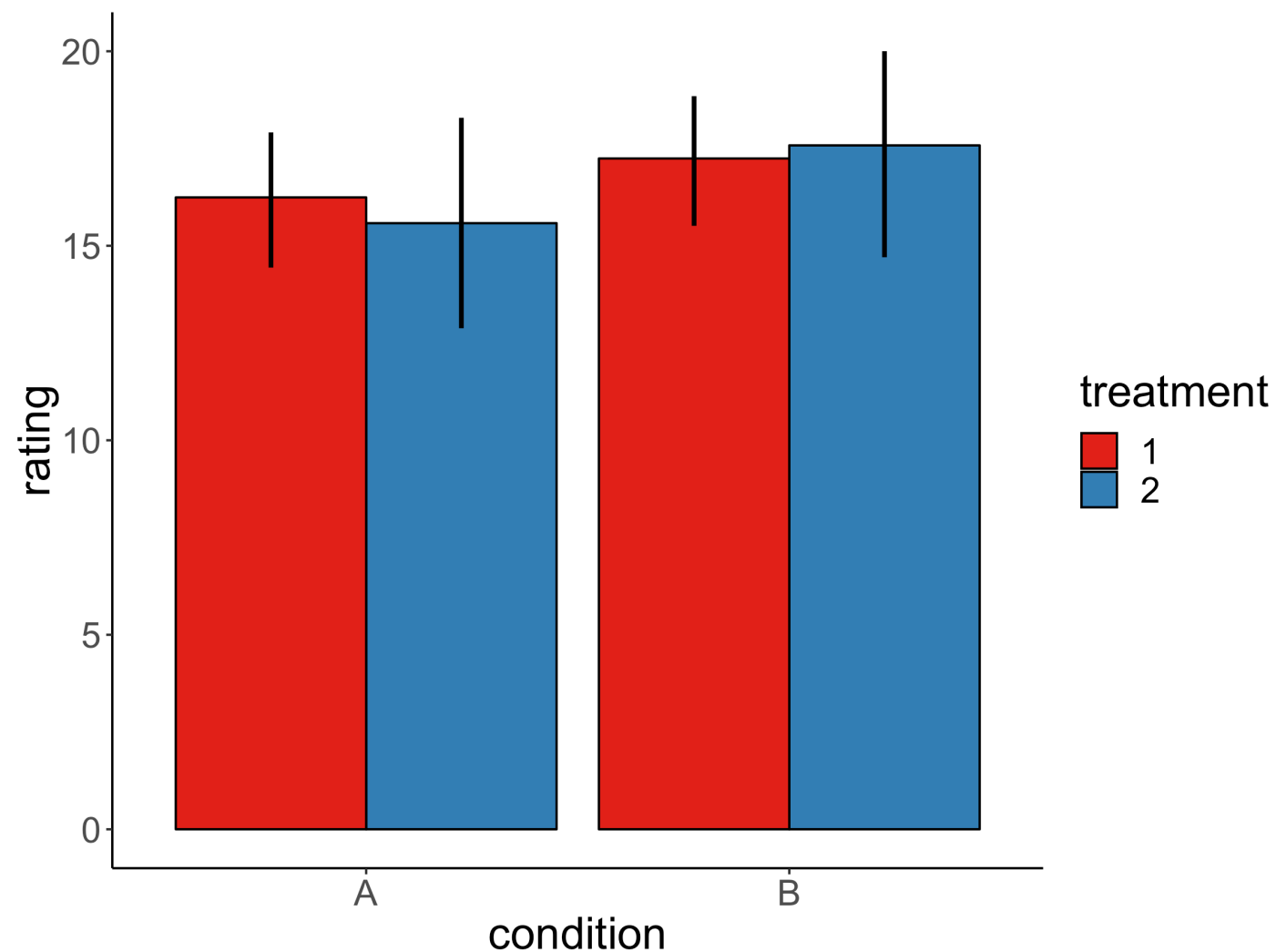


However, the quality of a player's hand significantly affected the final balance $F(2, 294) = 79.17, p < .001$. The final balance for good hands ($M = 13.03, SD = 4.65$) was significantly greater than for neutral hands ($M = 10.35, SD = 4.24$), and the balance for neutral hands was significantly higher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

Who is the ANOVA champ?

Which effects are significant?



Condition

Treatment

Condition x Treatment **interaction effect**

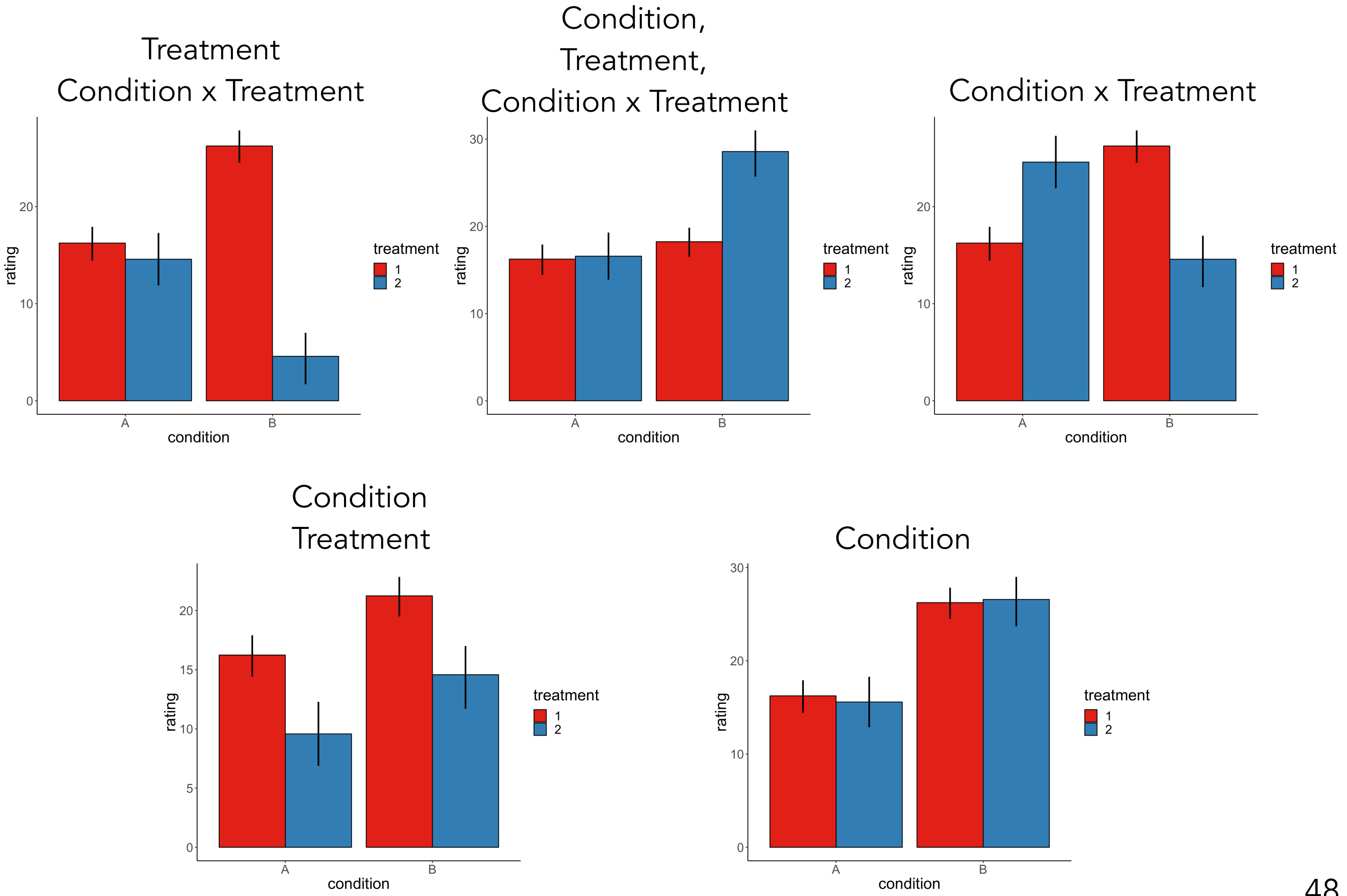
Condition, Treatment **two main effects**

Condition, Condition x Treatment

Treatment, Condition x Treatment

Condition, Treatment, Condition x Treatment

Solution



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!