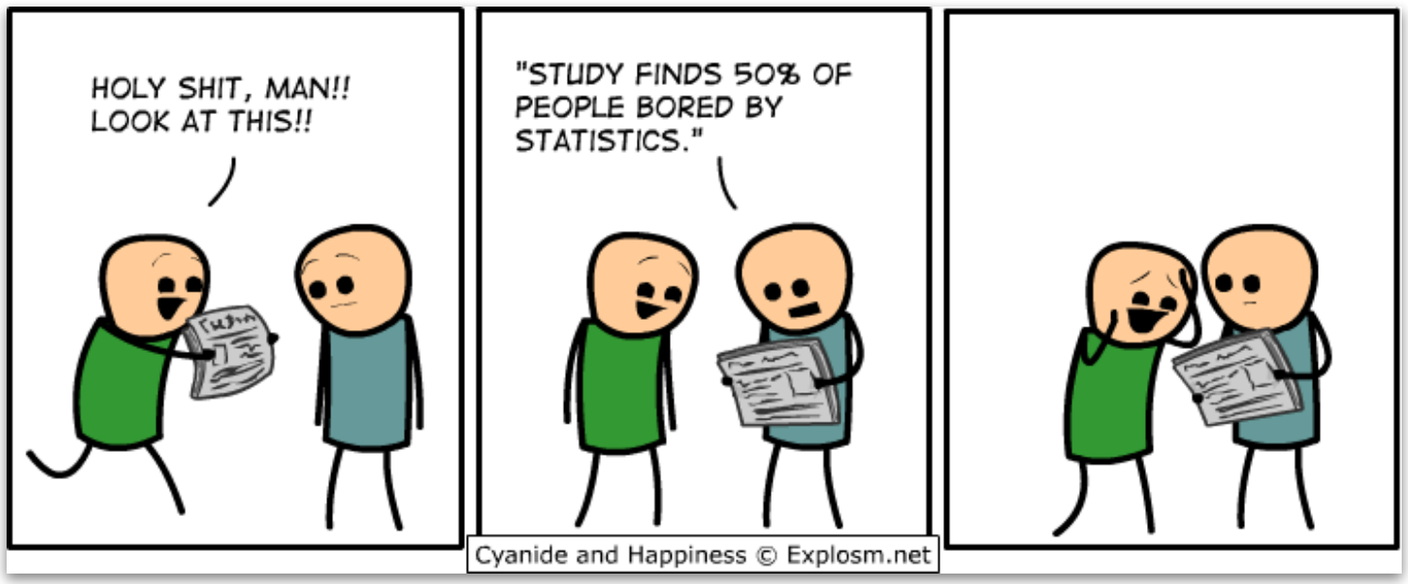
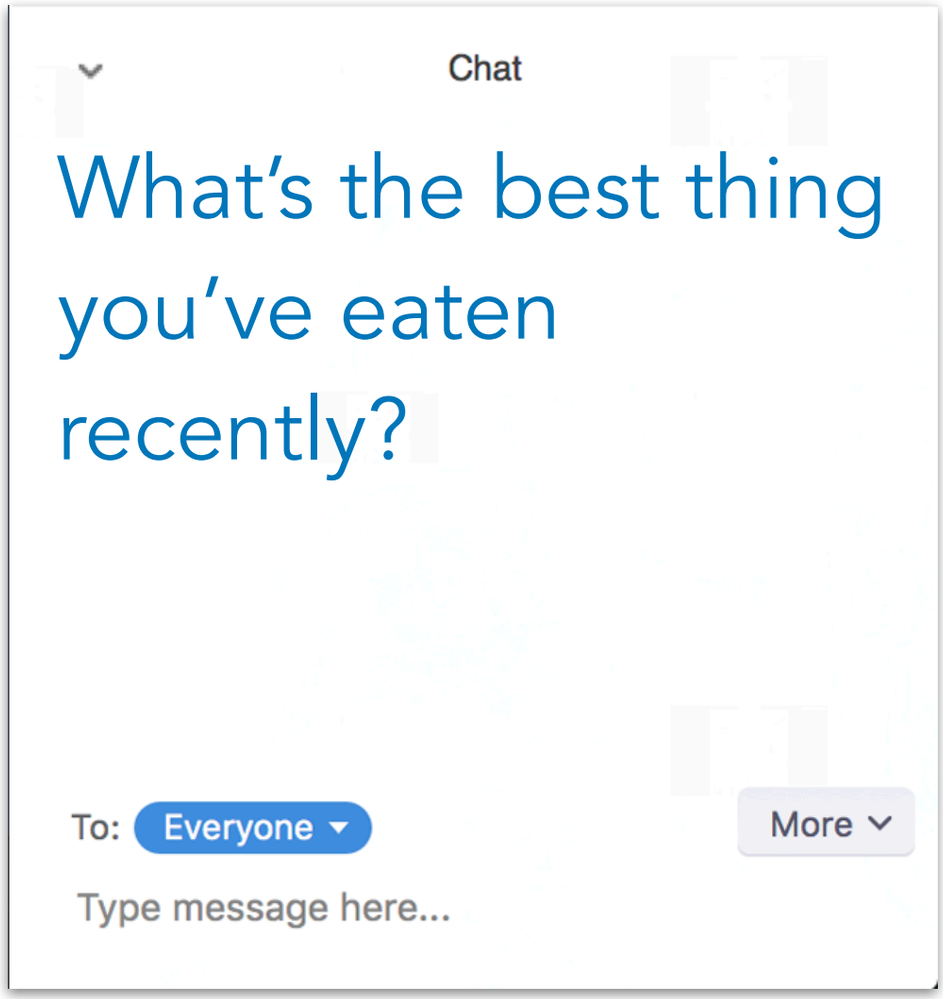
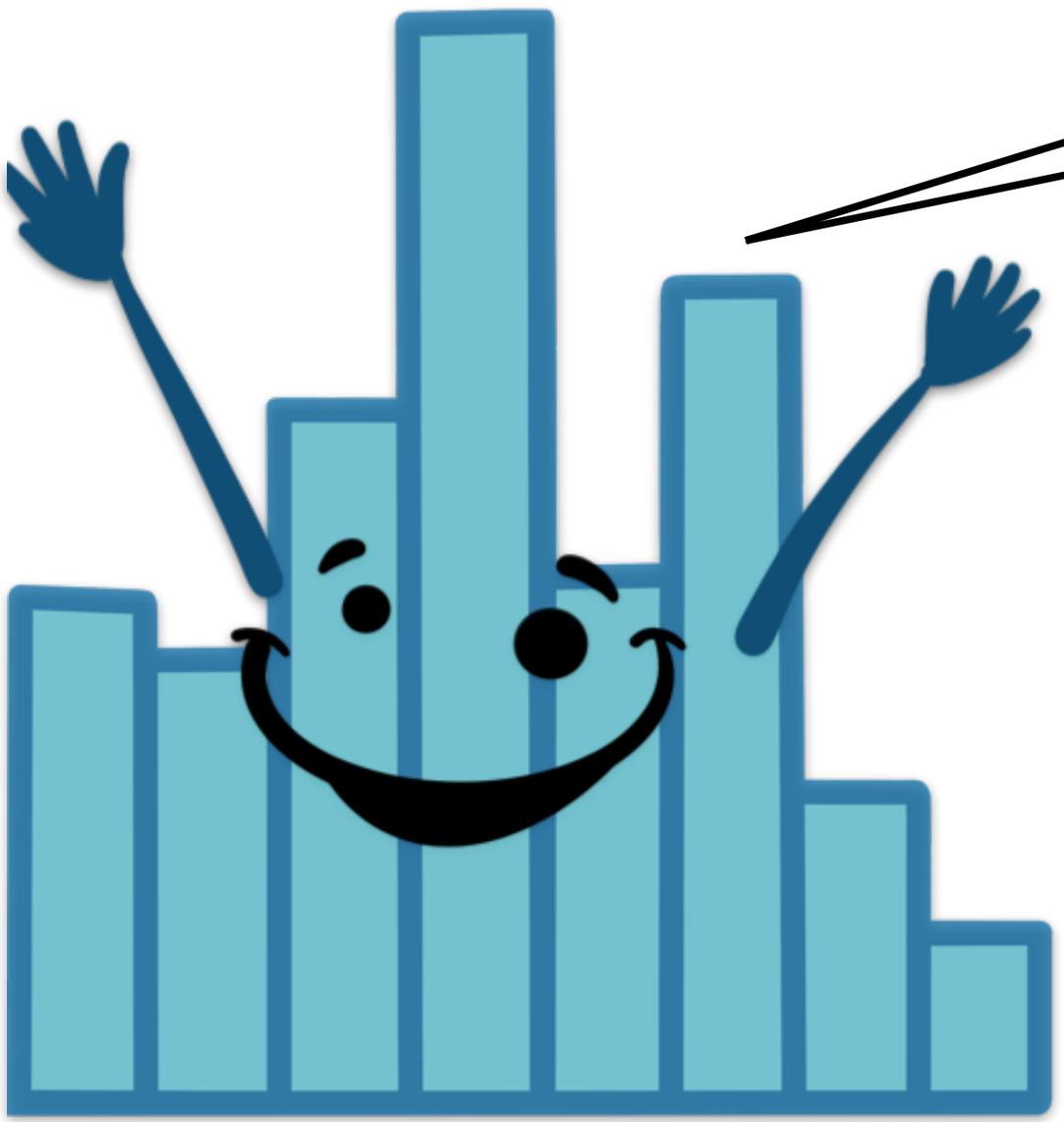
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Linear model 4





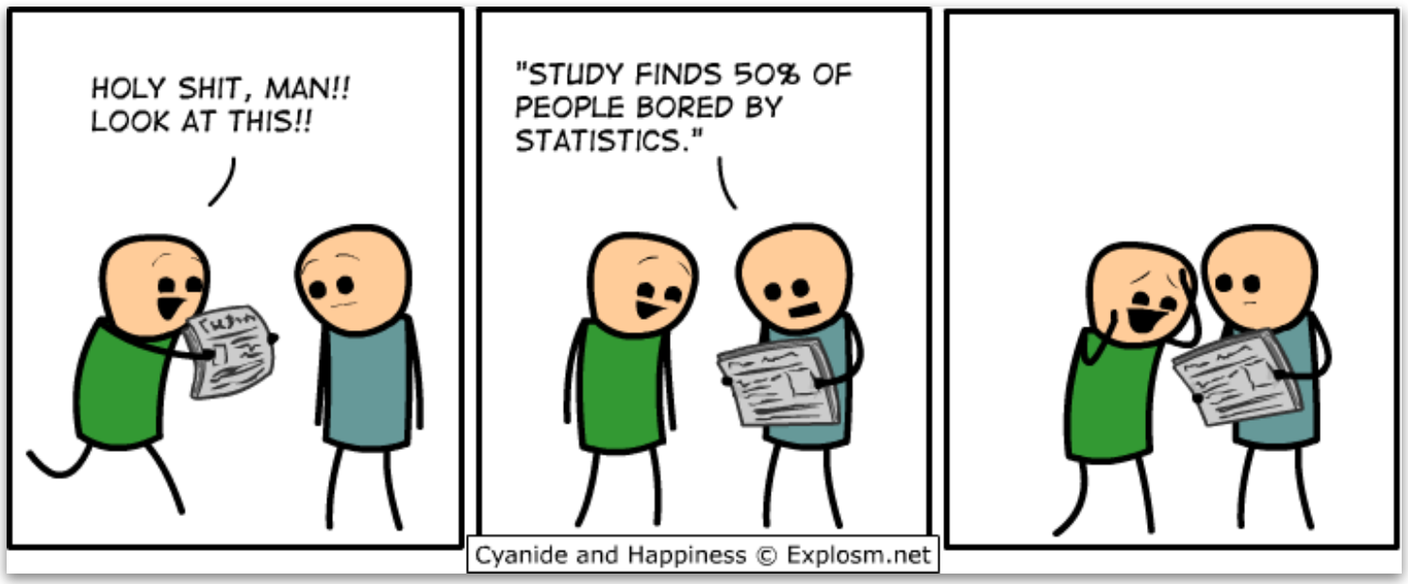
02/10/2021

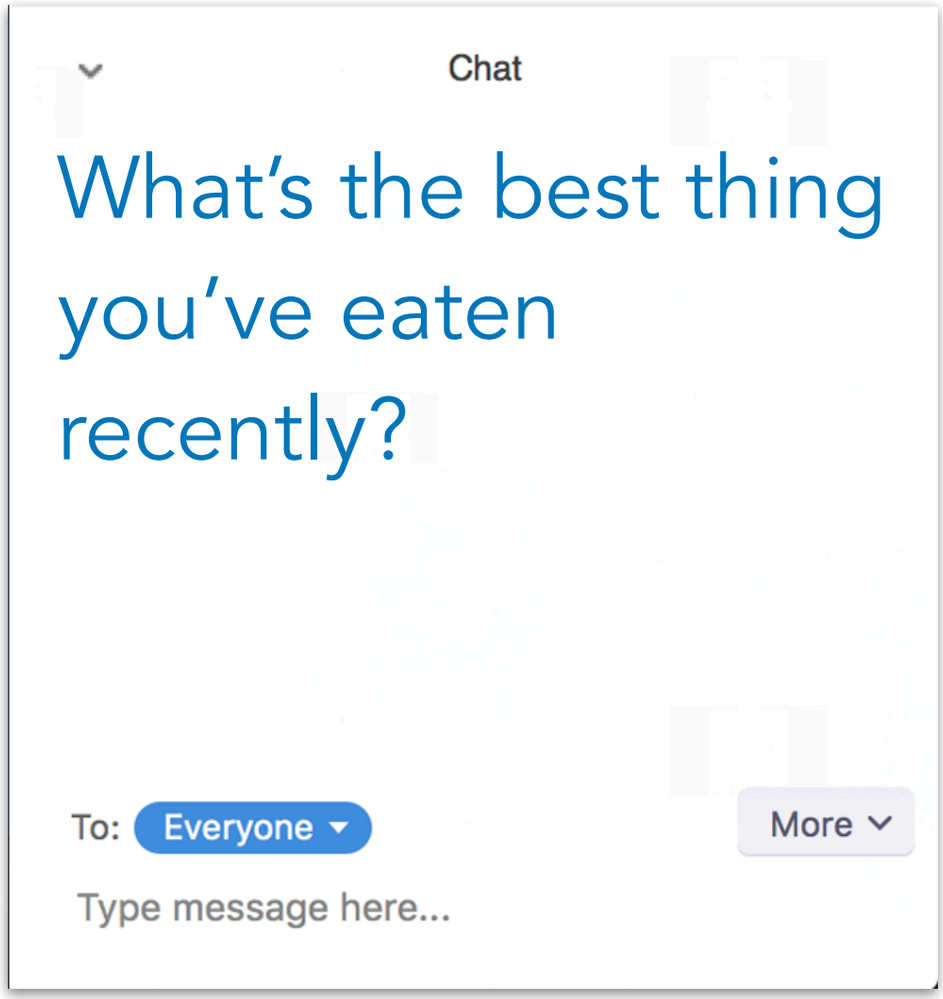


Remember to record the lecture!

PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Linear model 4



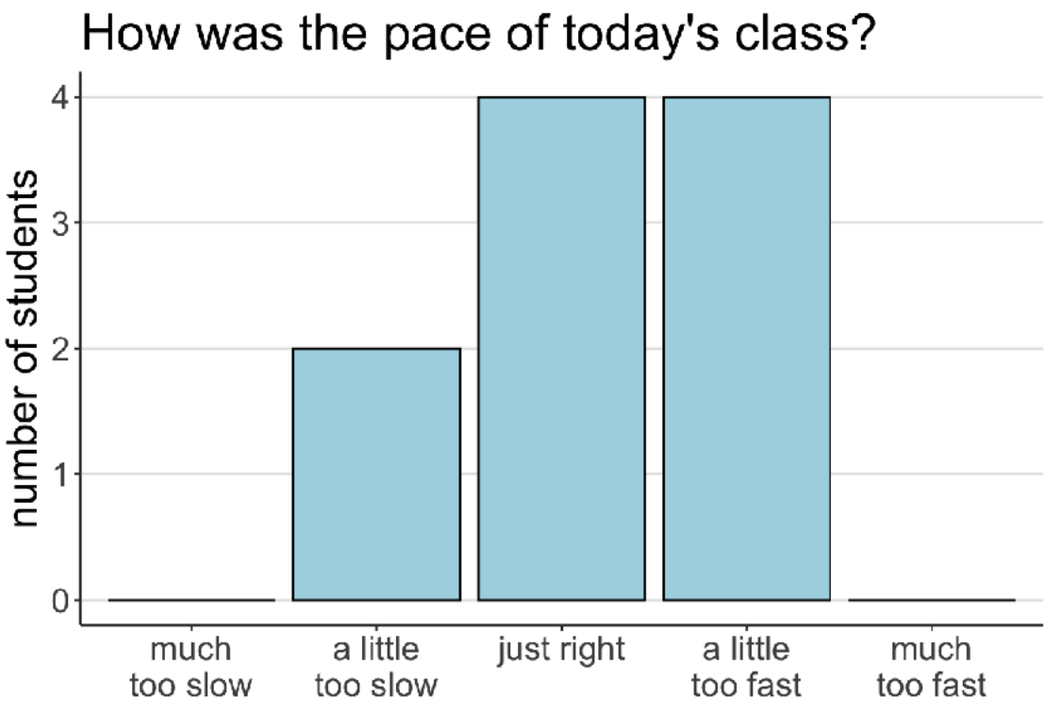


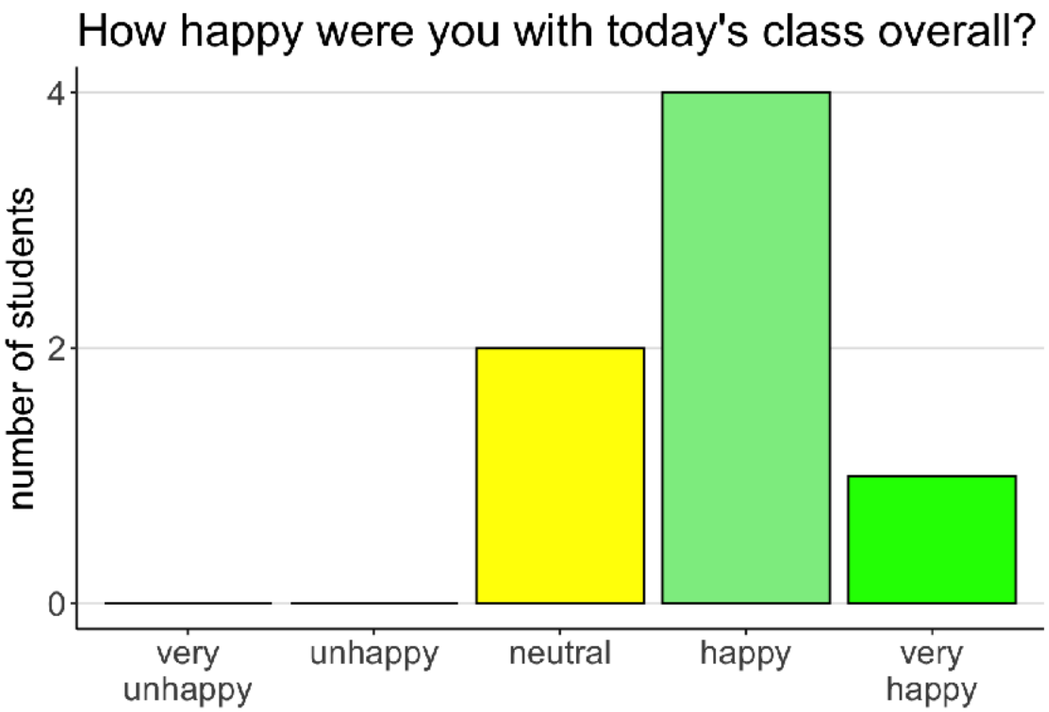
02/10/2021

Logistics

Feedback

Feedback


How was the pace of today's class?
number of students
4
3
2
1
0
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?
number of students
4
2
0
very unhappy
unhappy
neutral
happy
very happy

Feedback

sometimes you use random stats words that you haven't explain which can make it a little difficult to follow what you're saying. Like today, you said **"gaussian nose"** but we haven't learned what that means so in order to understand what you're saying I have to try to figure out what that means. It would be great if you could give quick 1-sentence summaries of new terms

$Y_i = 50 + \epsilon$ $\epsilon \sim \mathcal{N}(\mu = 0, \sigma)$

but how was the error generated in the true model?

I assumed normally distributed errors!

justification?

The central limit theorem!

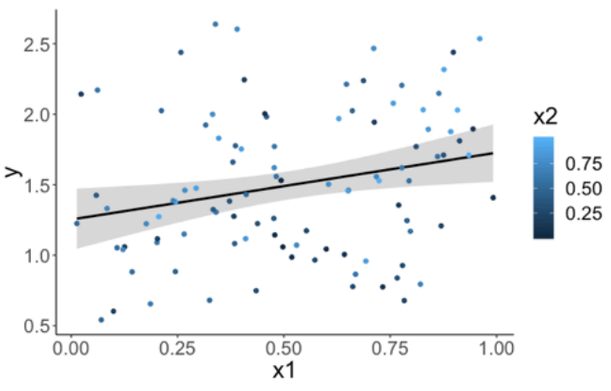
the distribution of the sum of individual error components will approximate a normal distribution

https://seeing-theory.brown.edu/probability-distributions/index.html#section3 55

Normality assumption

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)
)
```

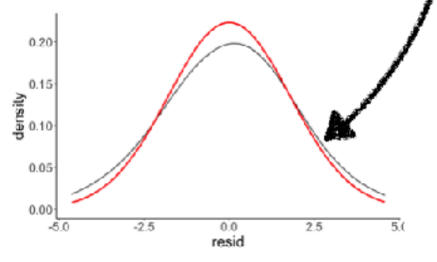


$Y_i = b_0 + b_1 \cdot x1_i + b_2 \cdot x2_i + e_i$

Gaussian noise

```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
-0.57	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



13

Feedback

do you have info on how long the midterm took students to complete last year?

we will significantly shorten the midterm this time!

Feedback

just a general question - how many problem sets are there in this class? Will there be problem sets at the same time as final project assignments later in the quarter, or will problem sets phase out?

we will have 6 problem sets in total

there will be no problem set for the last week of class

last week of class

Monday	March 8th	Bayesian data analysis II
Wednesday	March 10th	Bayesian data analysis III
Friday	March 12th	Guest lecture: Matthew Kay
Monday	March 15th	Guest lecture: Nilam Ram
Wednesday	March 17th	Final project presentations
Friday	March 19th	Final project presentations

Plan for today

- Analysis of Variance (ANOVA)
  - multiple categorical predictors (N-way ANOVA)
    - interpreting parameters
    - Who is the ANOVA champ?
    - unbalanced designs
- Linear contrasts
  - testing specific hypotheses with linear contrasts
  - `emmeans` for handling linear contrasts in R

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?

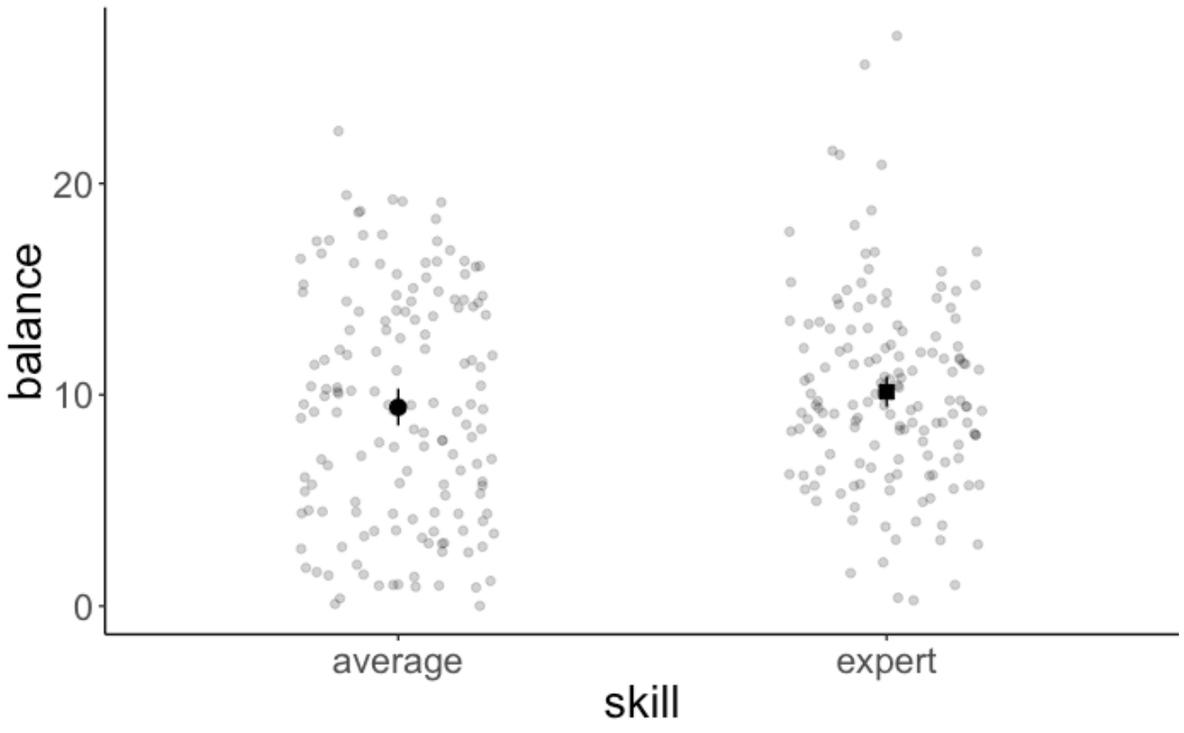
One testing whether skill level affects the final balance, and one testing whether quality of cards affects the final balance?

Interested in interactions!

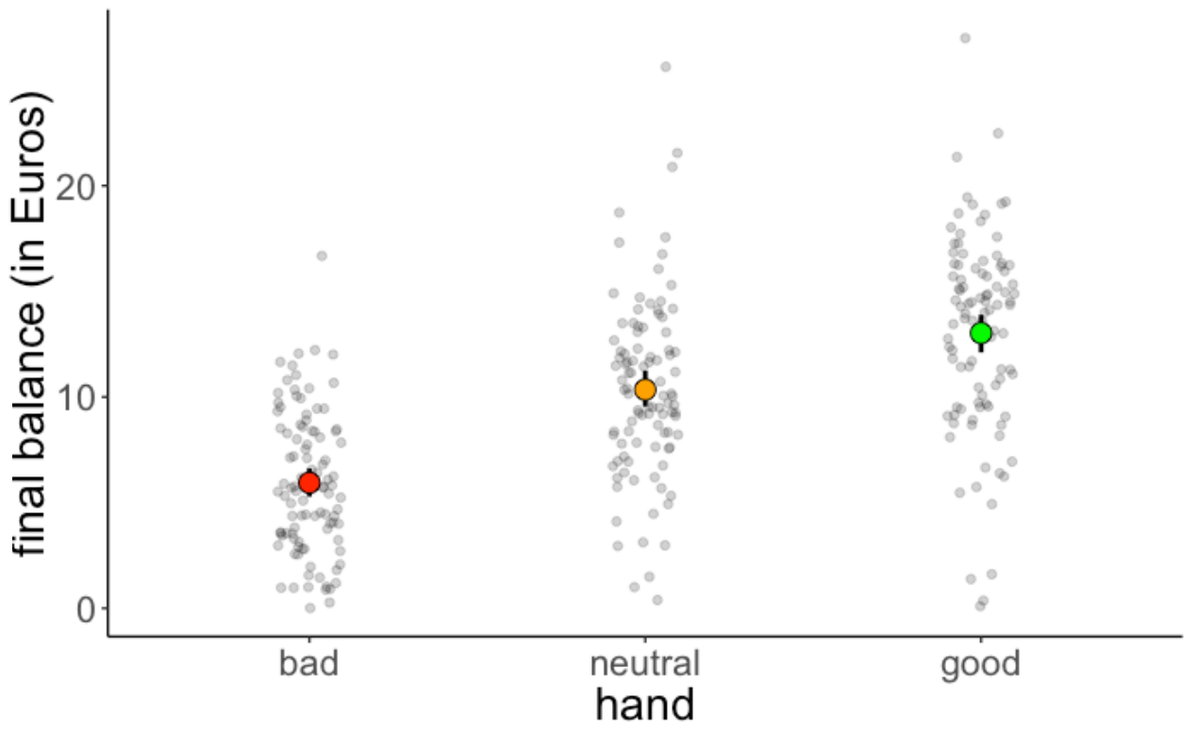
Does the effect of one variable depend on the other?

Visualize the data

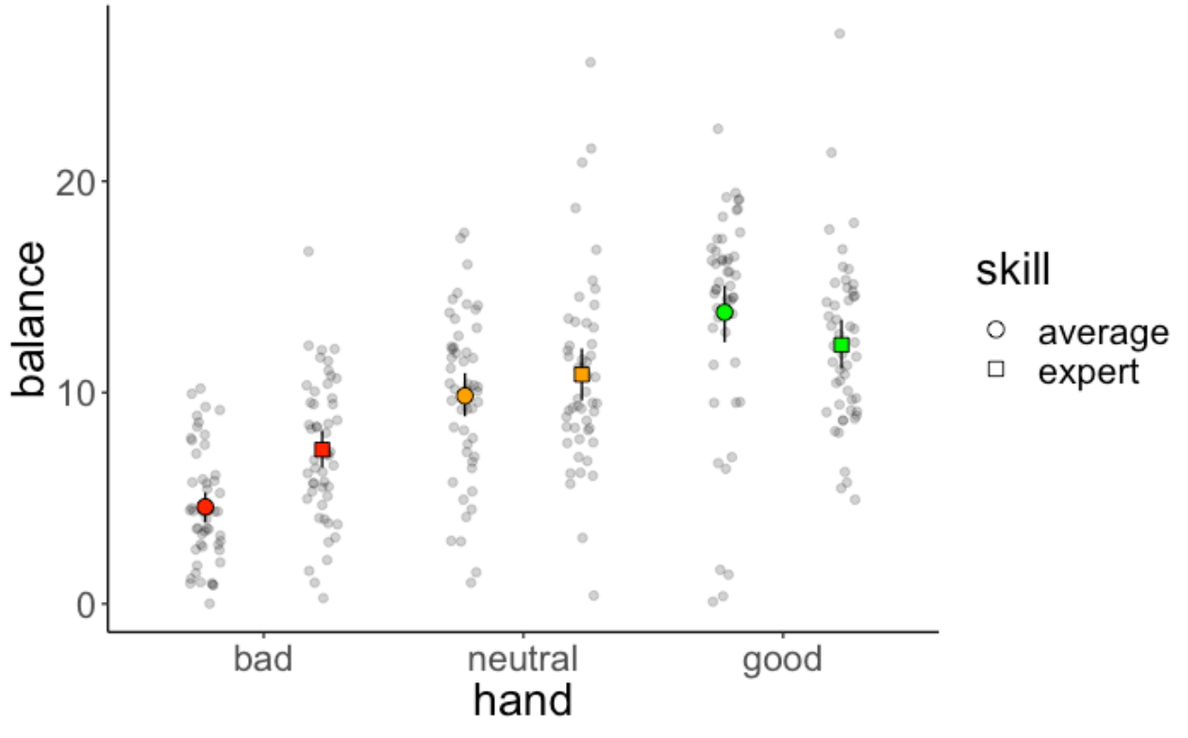
Skill



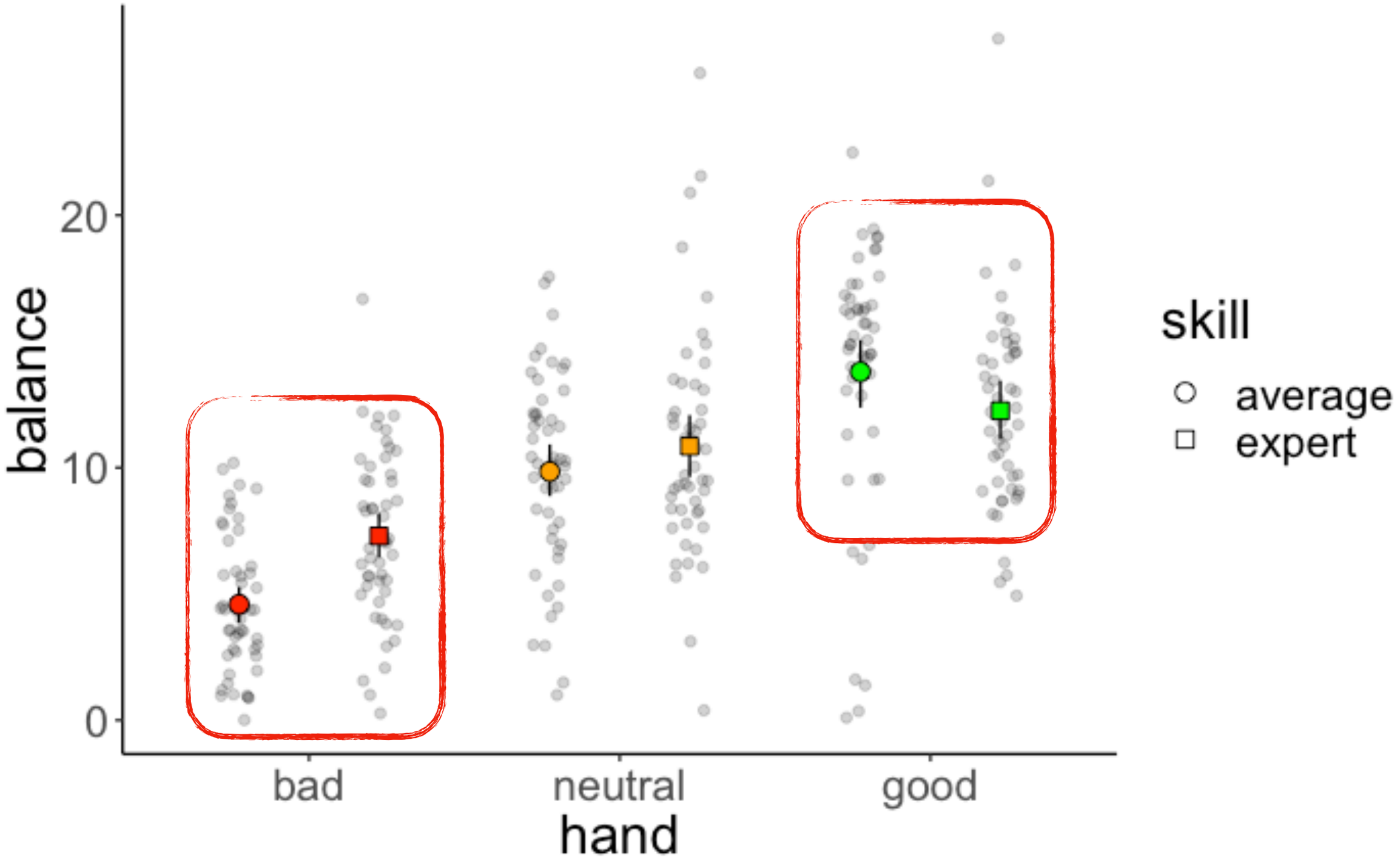
Quality of hand



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

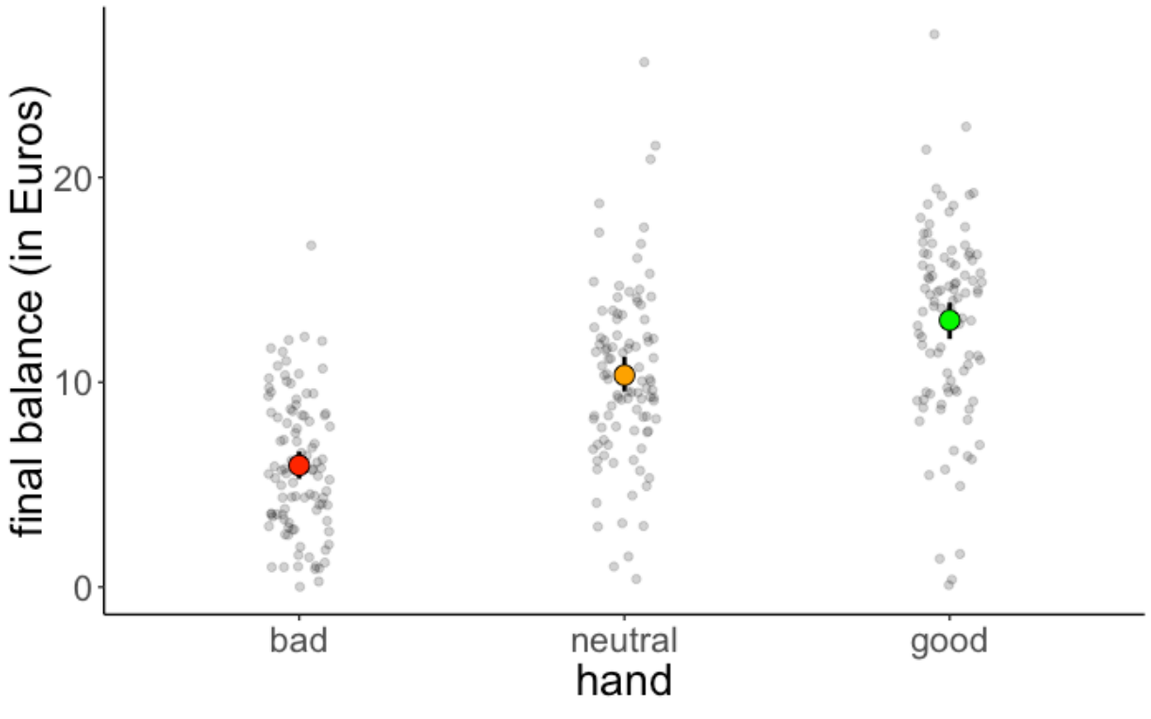
$$\widehat{\text{balance}}_i = b_0 + b_2 \cdot \text{hand\_good}_i + b_3 \cdot \text{skill\_expert}_i + b_5 \cdot \text{hand\_good:skill\_expert}_i = 12.26$$

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`



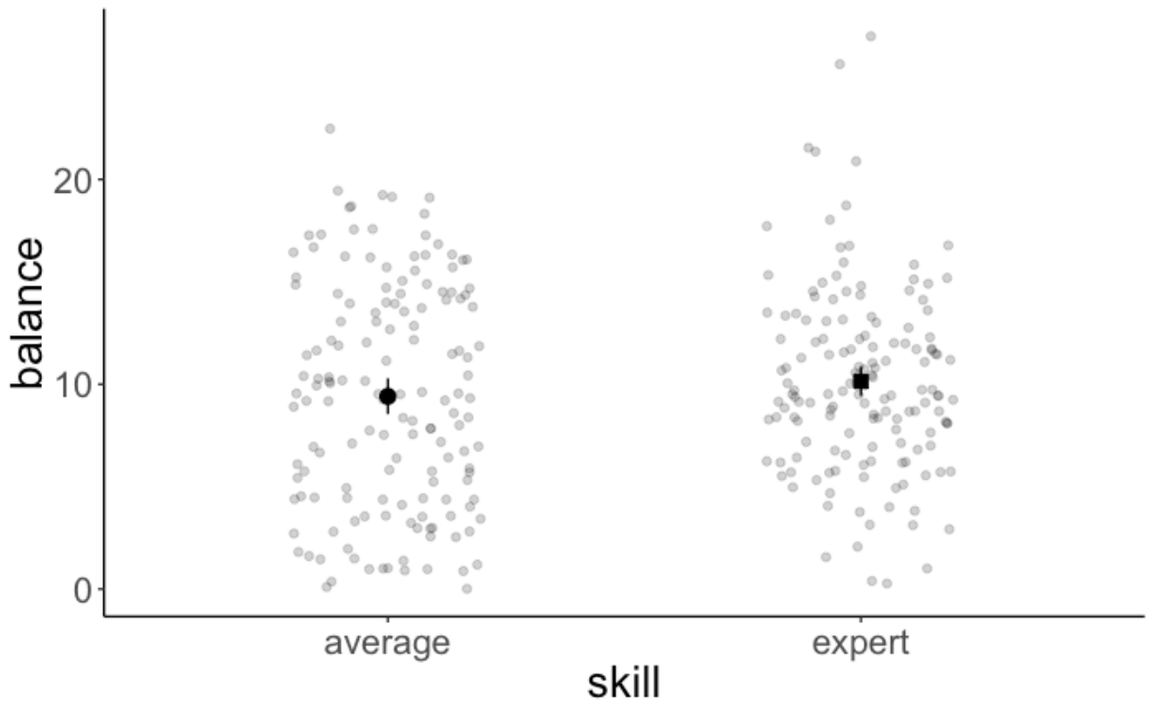
Analysis of variance

```
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

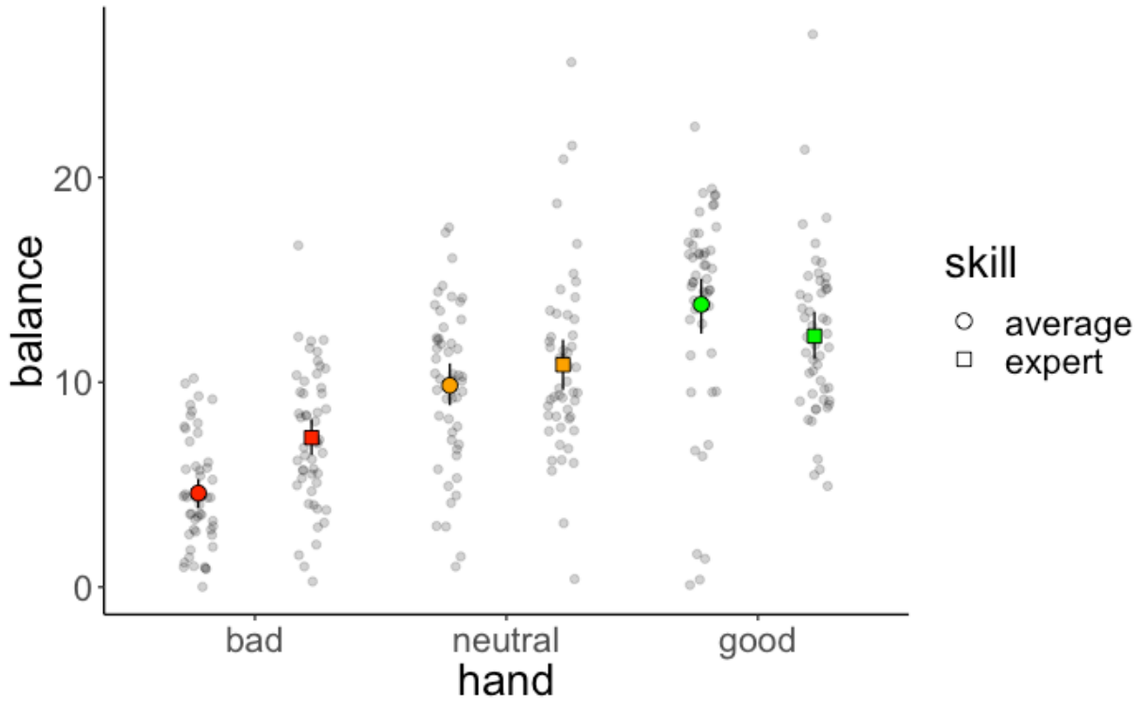
```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
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()
```

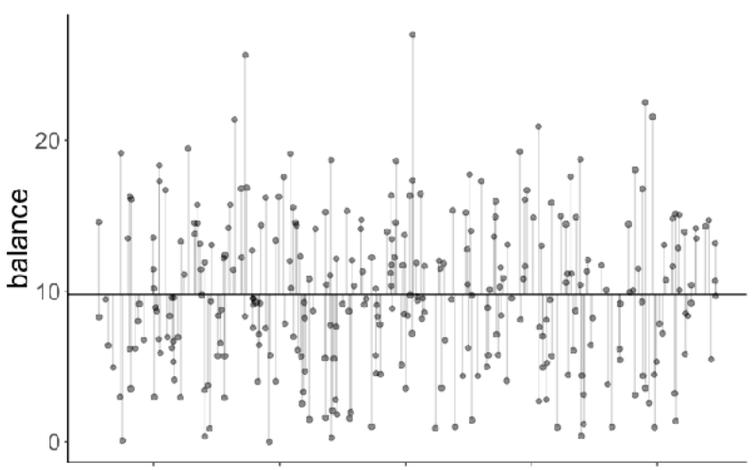
```
Analysis of Variance Table

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

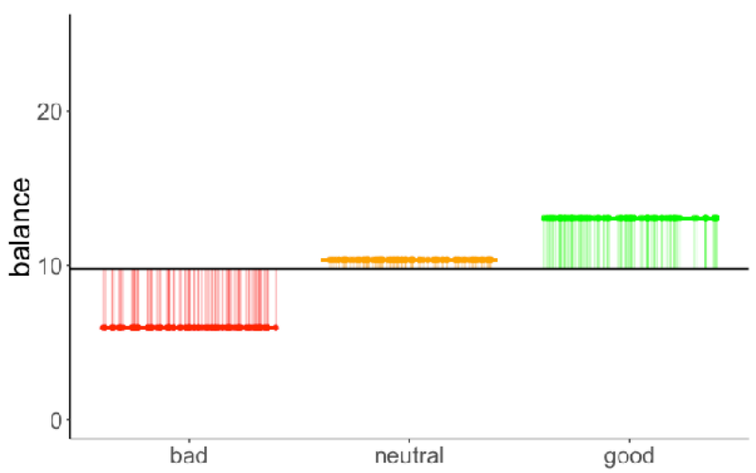
Variance decomposition

Total variance



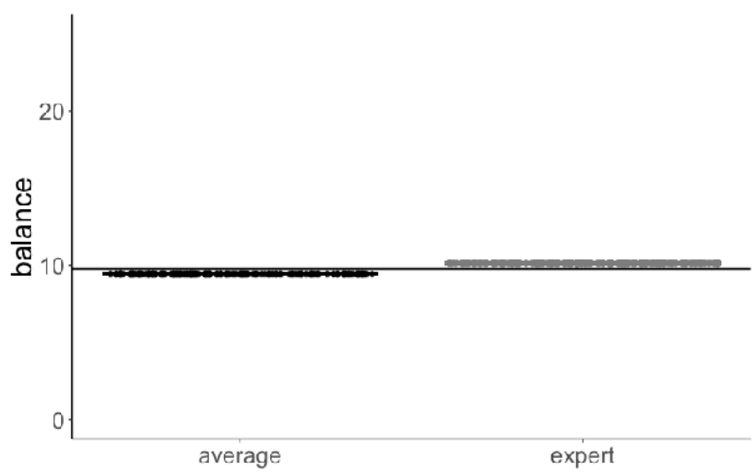
SS_{total}

Hand variance



SS_{hand}

Skill variance

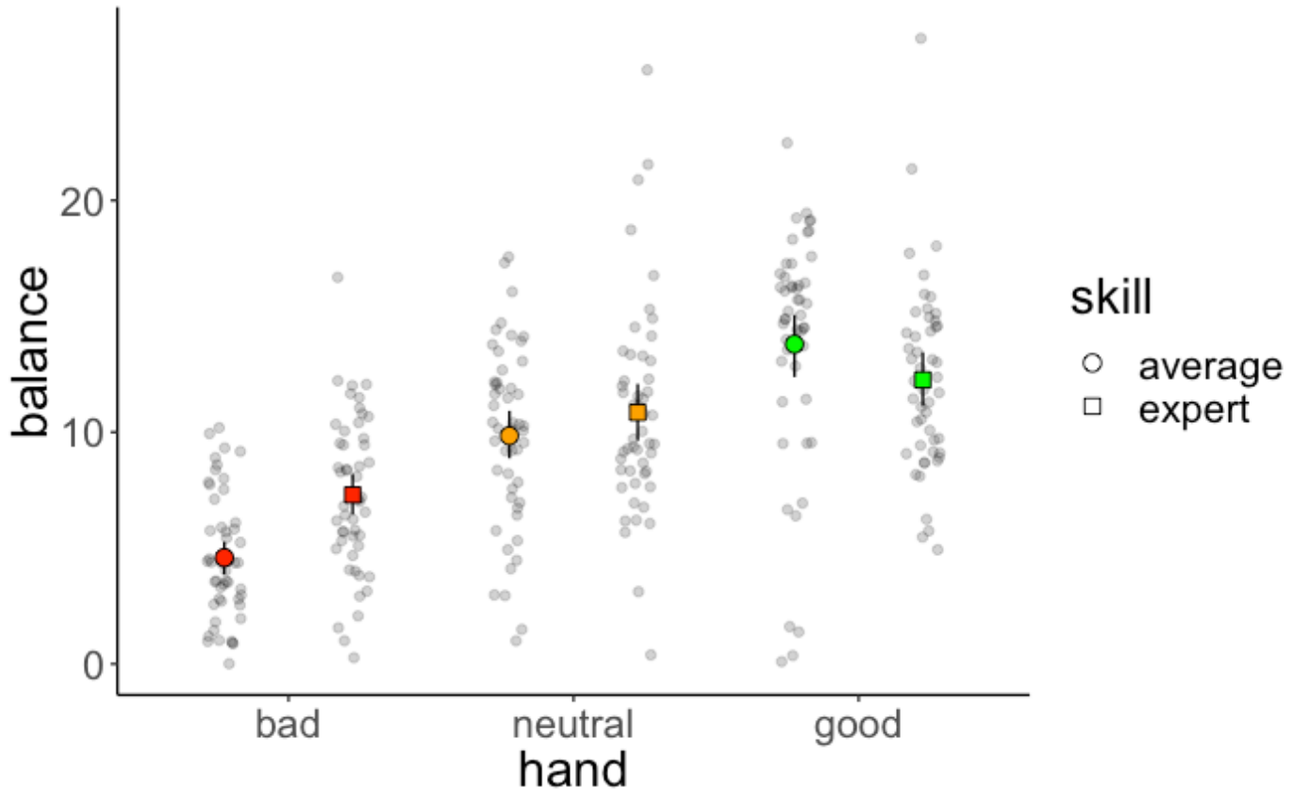


SS_{skill}

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

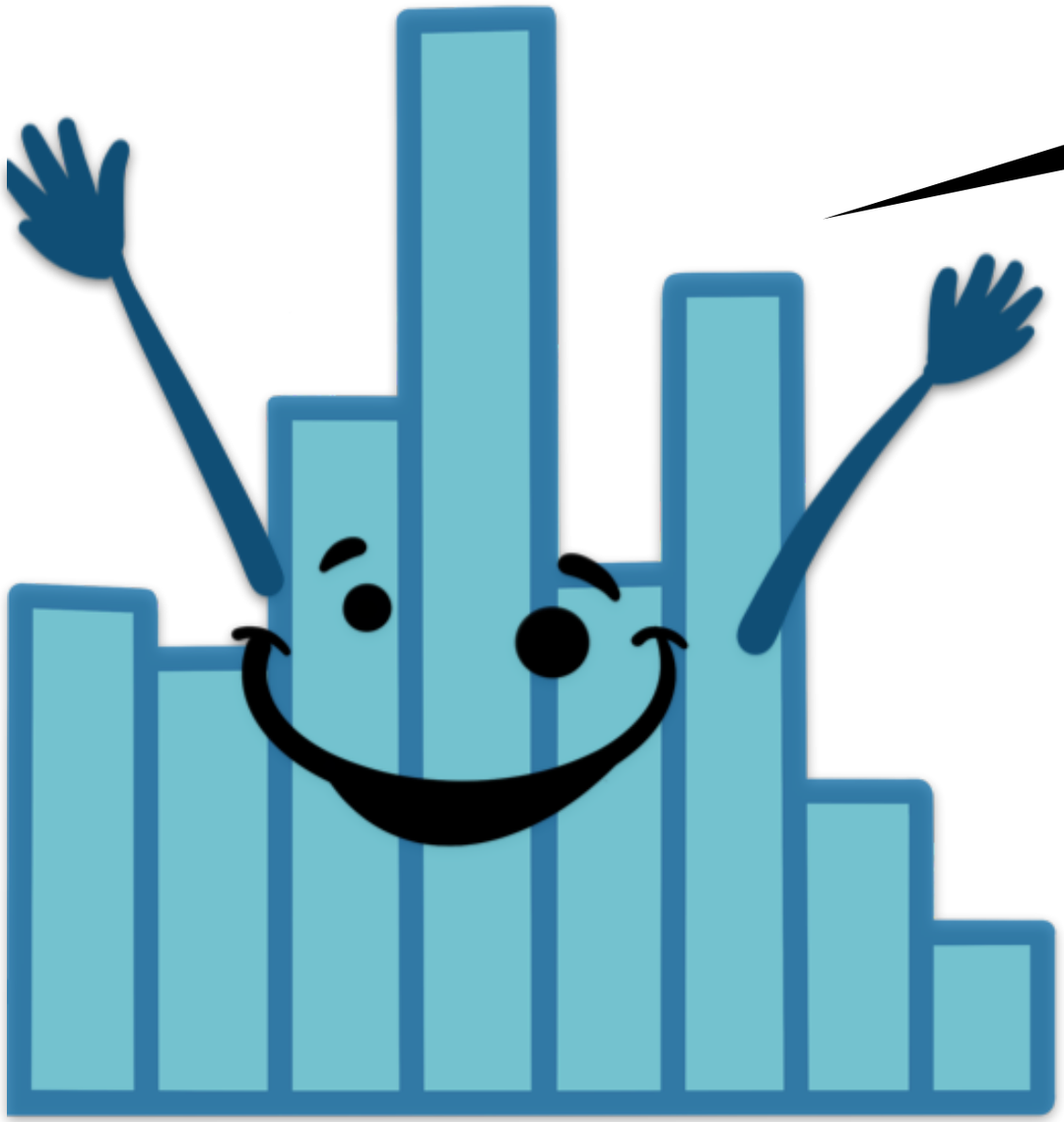
Reporting the results

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.



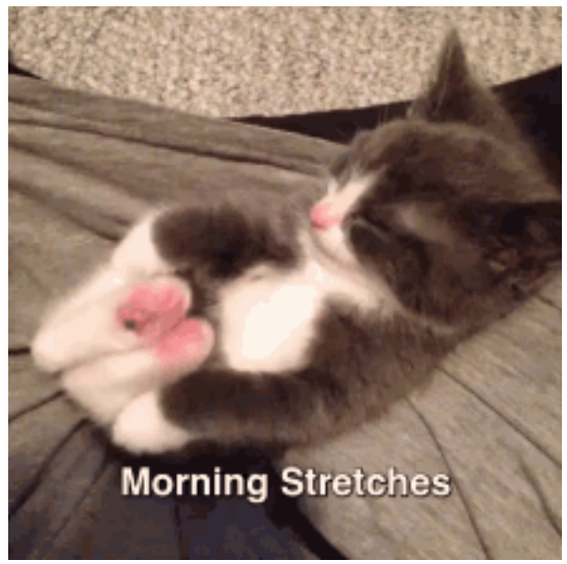
However, the quality of a player's hand significantly affected the final balance $F(2, 294) = 79.17$, $p < .001$. The final balance for good hands ($M = 13.03$, $SD = 4.65$) was significantly greater than for neutral hands ($M = 10.35$, $SD = 4.24$), and the balance for neutral hands was significantly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



01:00

stretch break!



Interpreting parameters

Parameter interpretation

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

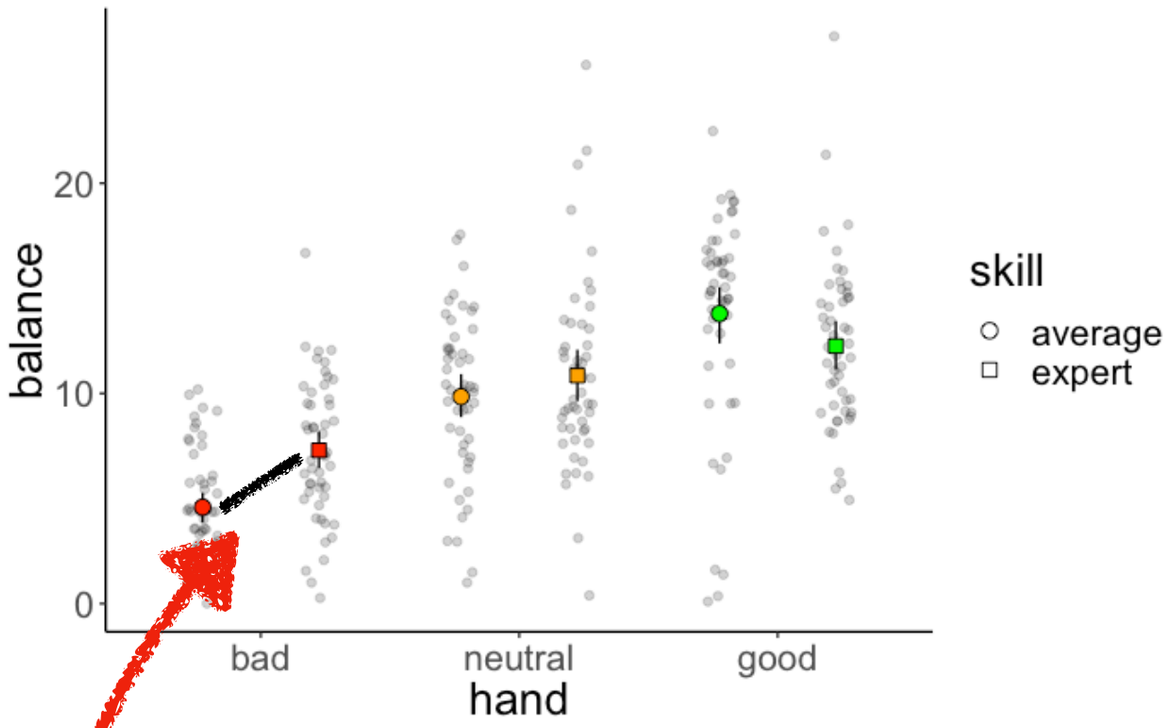
```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

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
```

acute danger of misinterpretation!!

```
' 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
```

~~there was a significant effect of skill~~

Parameter interpretation

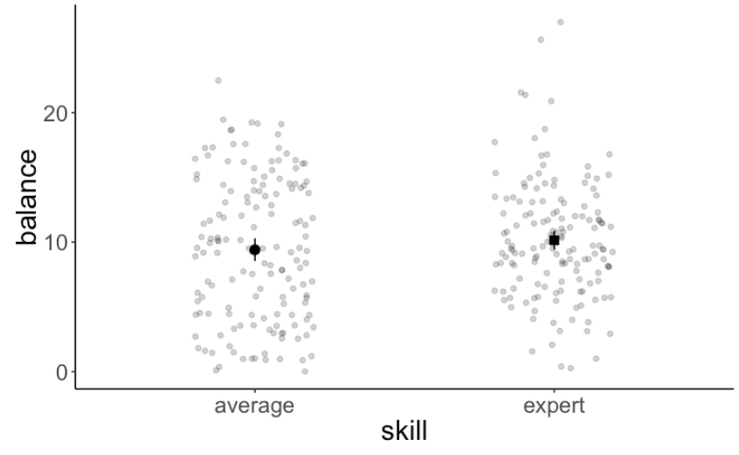



```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

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
```

```
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!



is this difference significantly different from 0?

hand	average	expert	difference
bad	4.59	7.3	2.71

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/

Interpreting parameters

`lm()` gives simple effects

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

```
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
```

`anova()` gives main effects, and interactions

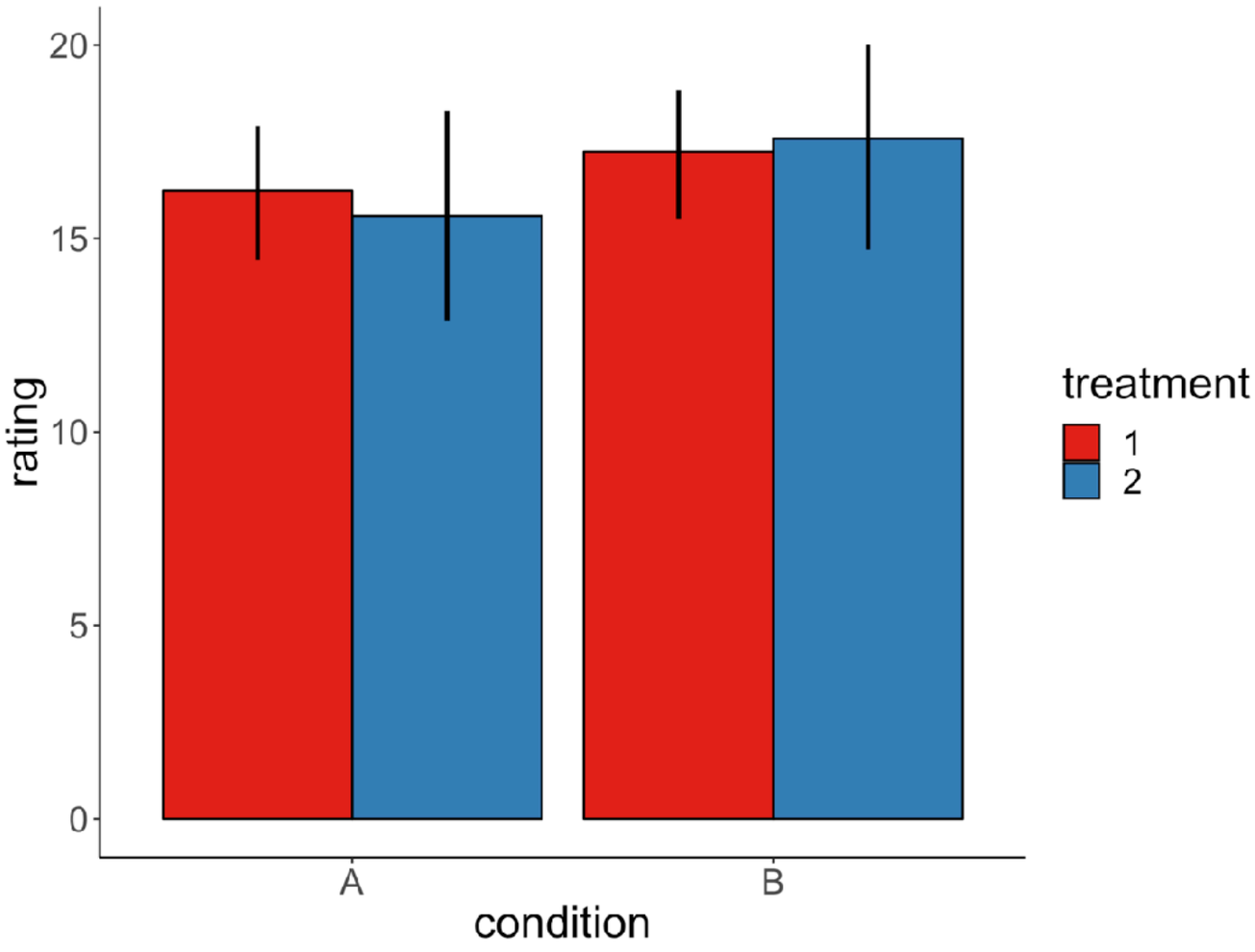
```
lm(formula = balance ~ hand * skill,
   data = df.poker) %>%
  anova()
```

```
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
```

Who is the ANOVA champ?

Who is the ANOVA champ?

Which effects are significant?



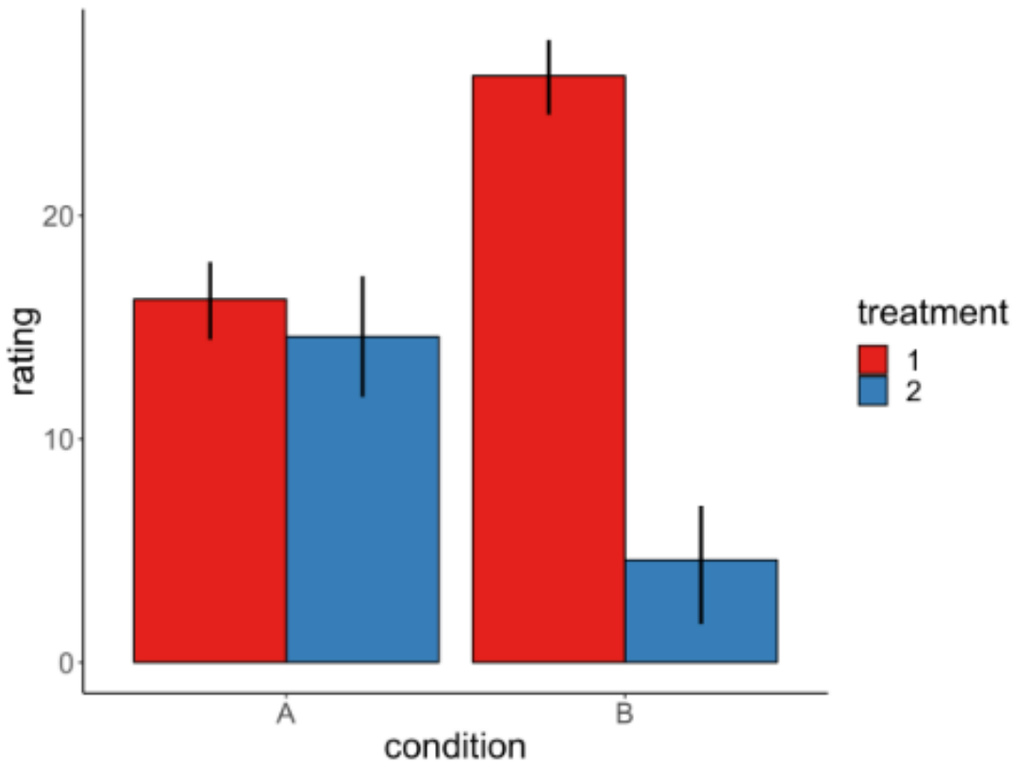
- Condition
- Treatment
- Condition x Treatment — **interaction effect**
- Condition, Treatment — **two main effects**
- Condition, Condition x Treatment
- Treatment, Condition x Treatment
- Condition, Treatment, Condition x Treatment

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

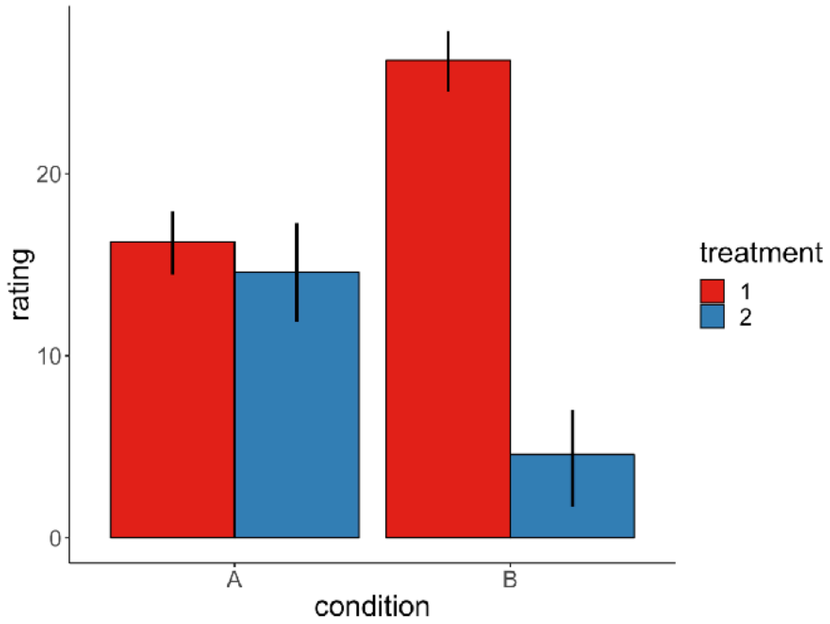
Which effects are significant?

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

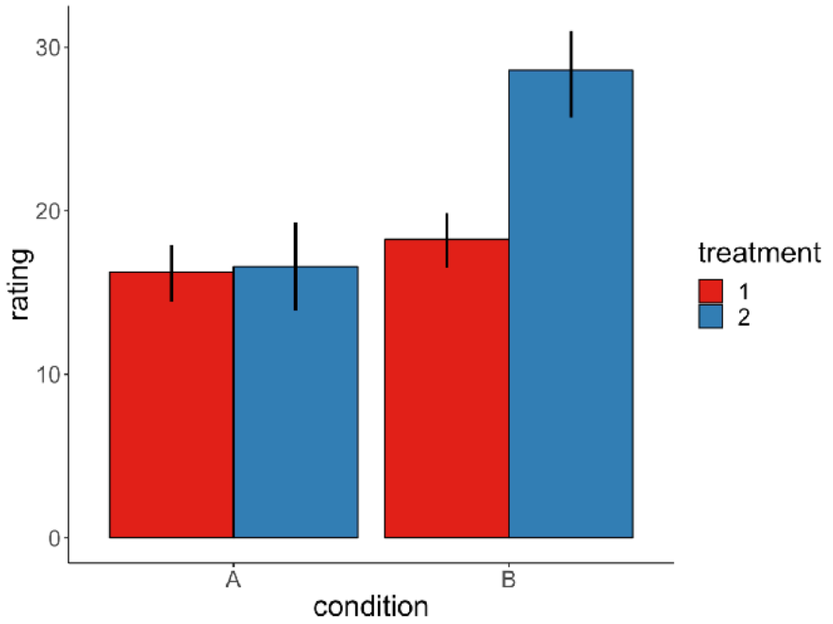
Leaderboard

Solution

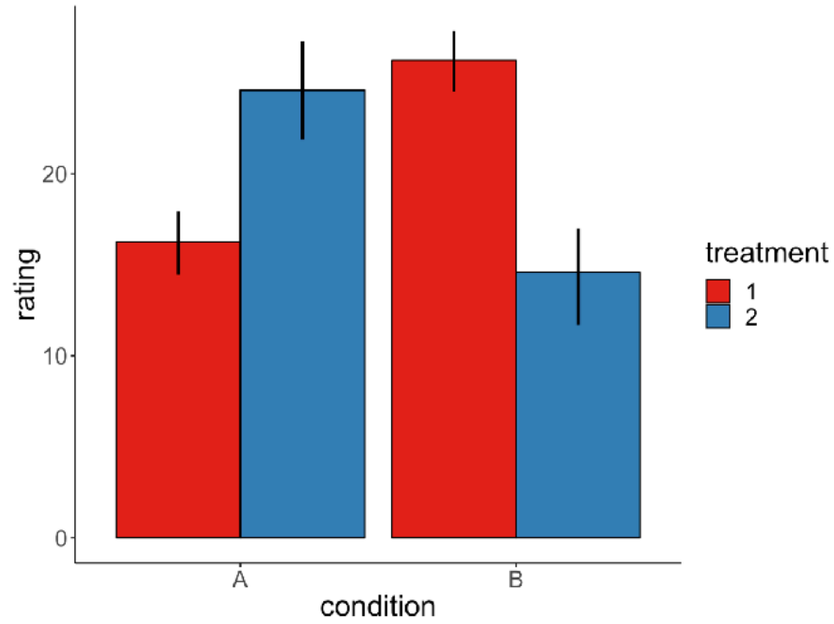
Treatment
Condition x Treatment



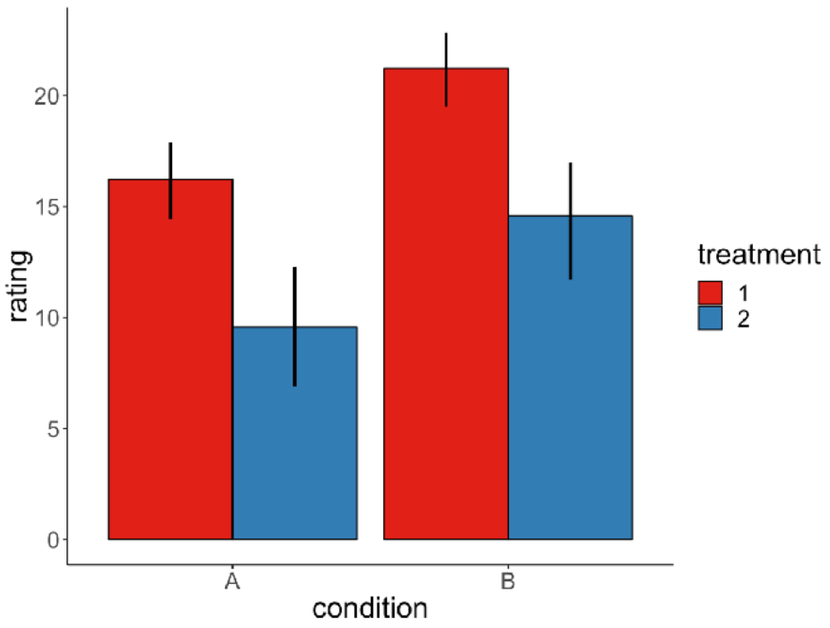
Condition,
Treatment,
Condition x Treatment



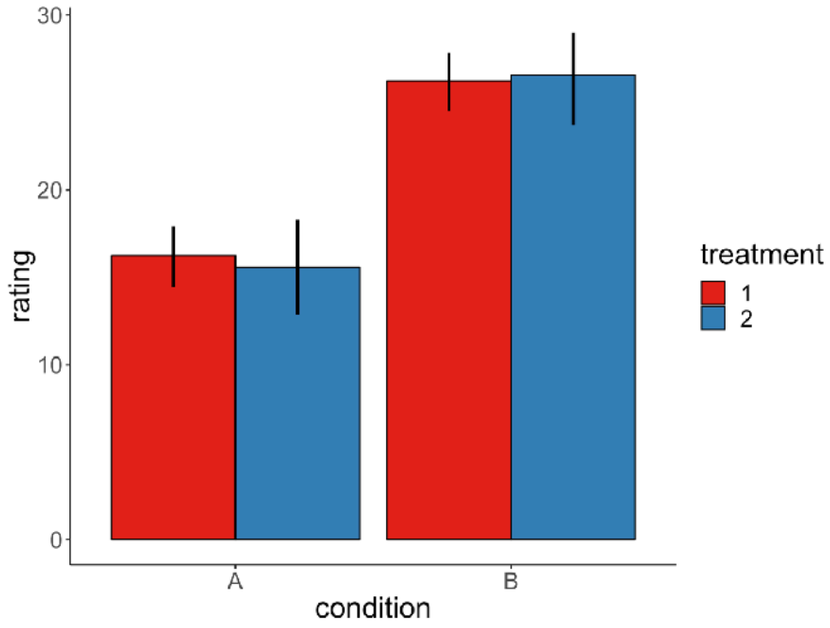
Condition x Treatment



Condition
Treatment

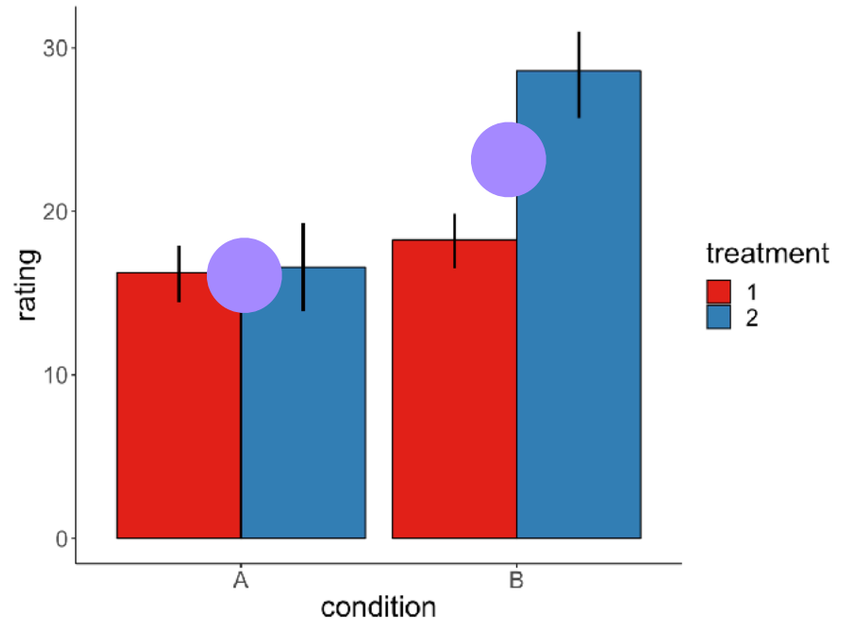


Condition



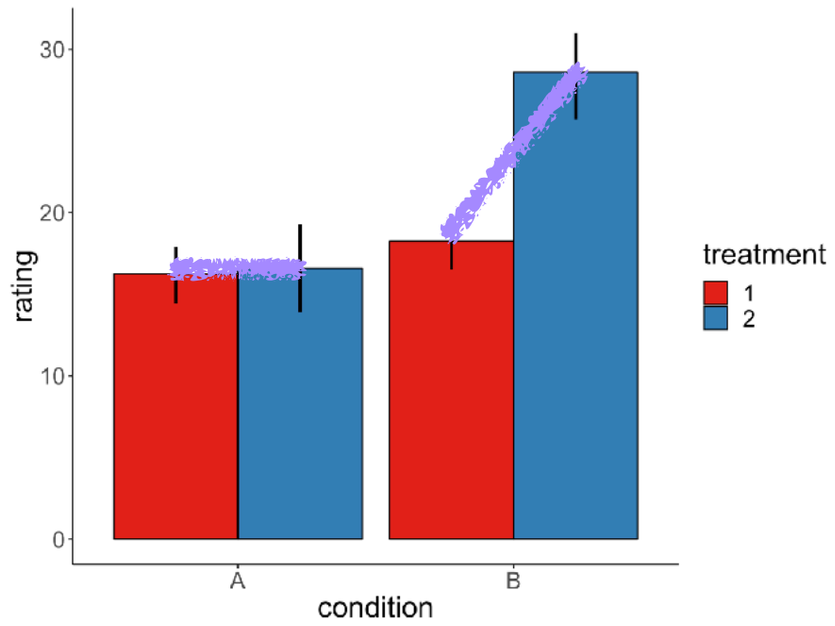
Solution

to detect main effects, try to visualize what the averaged group means would look like



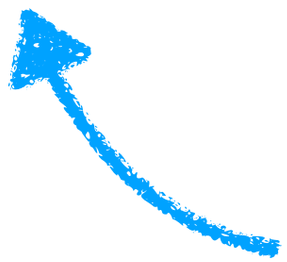
main effect of condition

to detect interaction effects, try to visualize whether the slopes are different from each other



interaction effect

Unbalanced designs



not the same number of participants in each cell

ANOVA

- for all the examples so far, 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

```
1 lm(formula = balance ~ skill + hand, data = df.poker.unbalanced) %>%
2   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

```
1 lm(formula = balance ~ hand + skill, data = df.poker.unbalanced) %>%
2   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:

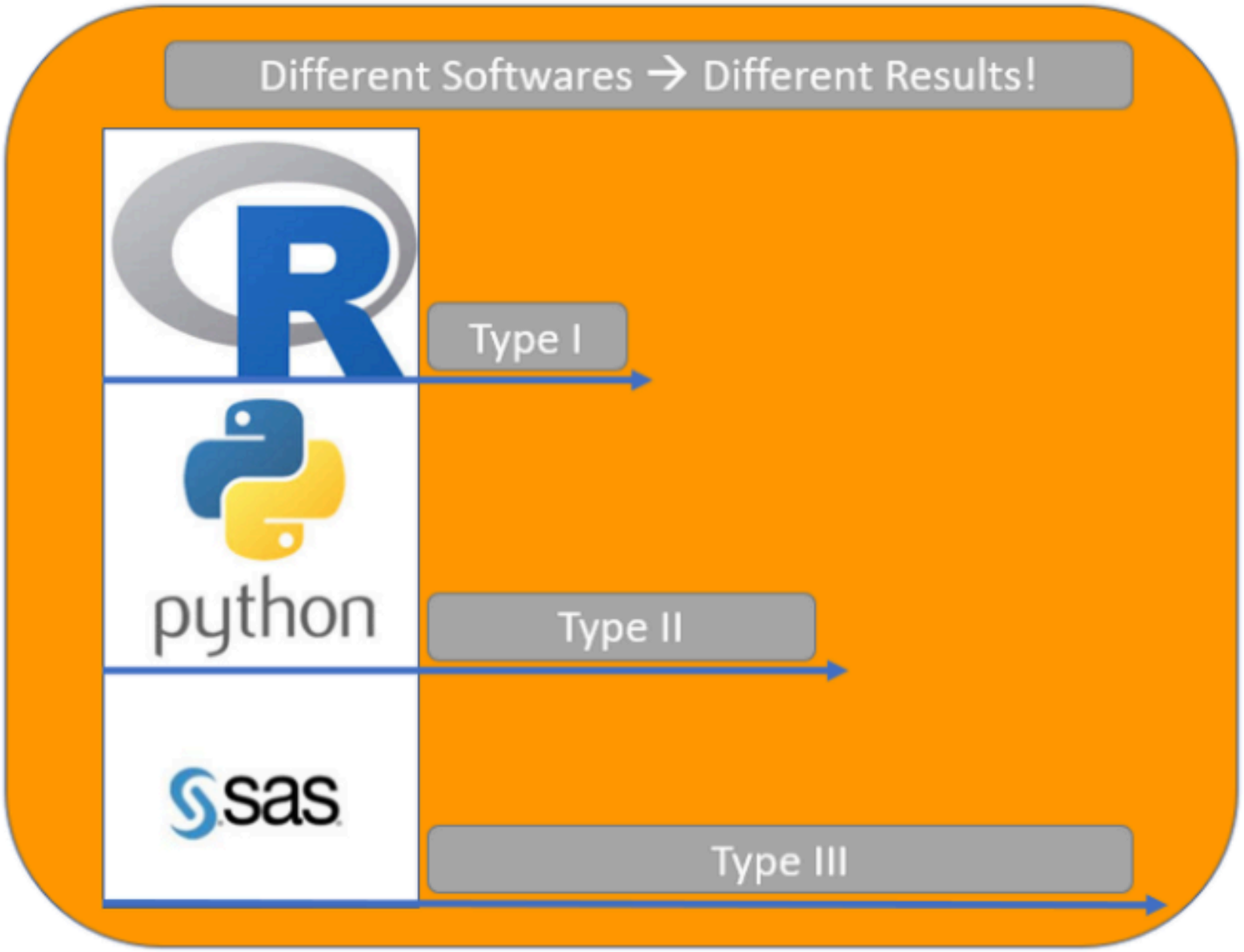
- Like Type II, the Type III Sums of Squares are not sequential, so the order of specification does not matter.
- Unlike Type II, the Type III Sums of Squares do specify an interaction effect.

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

very important!!

load the "car" package

set the contrasts

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

```r
library("car")

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

```
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. the number of participants is not the same for each cell in our design).
- 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**, rather than using the default dummy coding scheme.

Linear contrasts

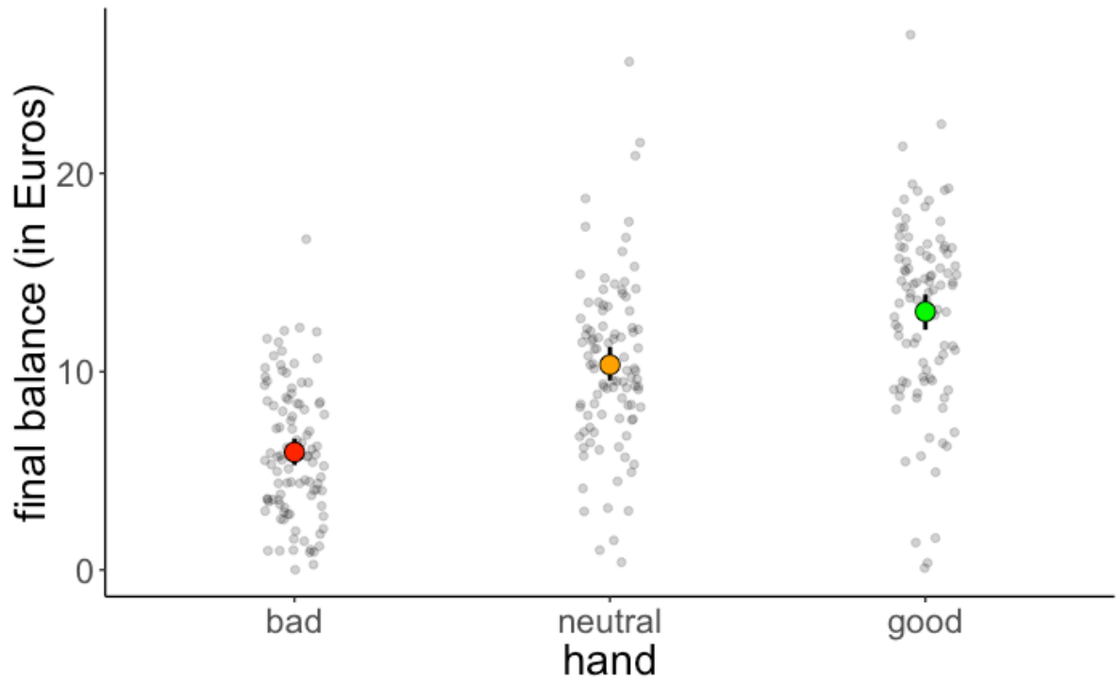
Coding schemes

- **Dummy coding:**
  - the reference category is coded as 0
  - represented by the intercept
  - all other categories are compared to this reference category
- **Effect coding:**
  - the intercept is the grand mean
  - all other categories are compared to the grand mean

this is what we want to do in ANOVAs

Schad, D. J., Vasishth, S., Hohenstein, S., & Kliegl, R. (2020). How to capitalize on a priori contrasts in linear (mixed) models: A tutorial. *Journal of Memory and Language*, 110, 104038.

Understanding dummy coding



how can we encode this information as a numeric predictor?

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

contrast coding

```
fit = 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
```

```
1 model.matrix(fit) %>%
2   as_tibble() %>%
3   distinct()
```

	(Intercept)	handneutral	handgood
bad	1	0	0
neutral	1	1	0
good	1	0	1

Understanding dummy coding

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

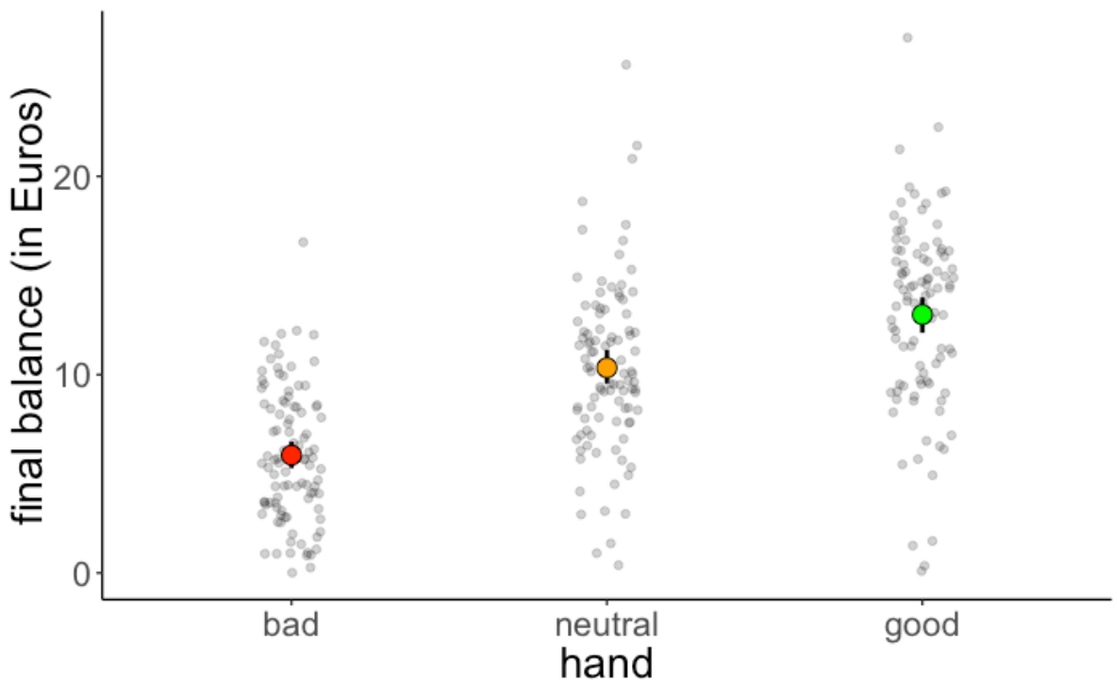
```
lm(formula = balance ~ hand * skill, data = df.poker)
```

```
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

Understanding effect coding



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

```r
fit = lm(formula = balance ~ 1 + hand,
         contrasts = list(hand = "contr.sum"),
         data = df.poker)
```

```
Call:
lm(formula = balance ~ 1 + hand, data = df.poker, contrasts =
list(hand = "contr.sum"))

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)   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

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
```

```r
model.matrix(fit) %>%
  as_tibble() %>%
  distinct()
```

	(Intercept)	hand1	hand2
bad	1	1	0
neutral	1	0	1
good	1	-1	-1

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
```

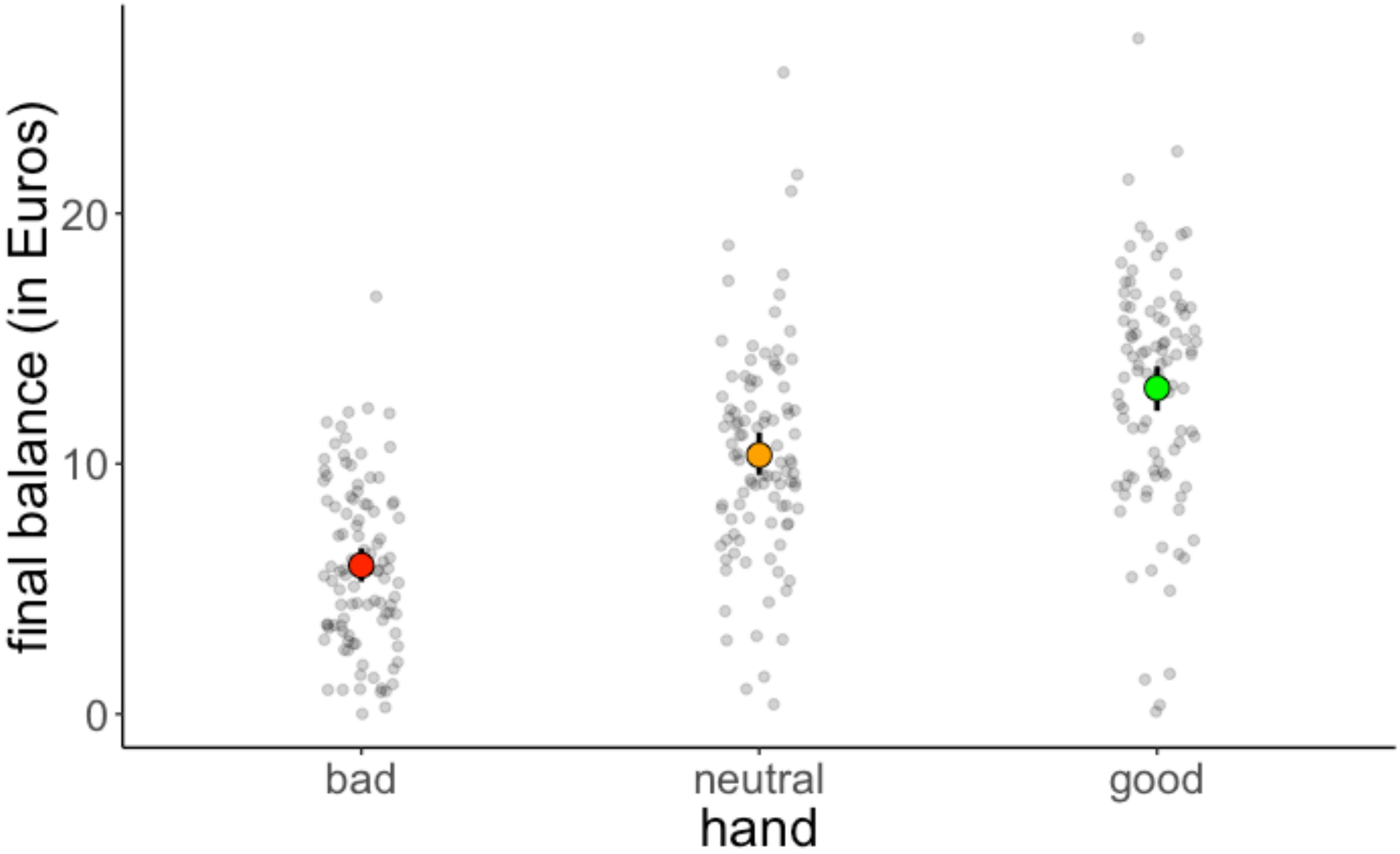
reference

grand mean = 9.77

Note: The last level in each factor is dropped.

Testing (more) specific hypotheses with linear contrasts

Do better hands win more money?



Do better hands win more money?

ANOVA

Does card quality affect the final balance?

post-hoc tests

bad vs. **neutral**

neutral vs. **good**

Is there are more direct way of asking this question with a statistical model?

Contrasts

```
df.poker = df.poker %>%
  mutate(hand_contrast = factor(hand,
                                levels = c("bad", "neutral", "good"),
                                labels = c(-1, 0, 1)),
         hand_contrast = hand_contrast %>% as.character() %>% as.numeric())
```

participant	hand	balance	hand_contrast
1	bad	4.00	-1
2	bad	5.55	-1
3	bad	9.45	-1
51	neutral	11.74	0
52	neutral	10.04	0
53	neutral	9.49	0
101	good	10.86	1
102	good	8.68	1
103	good	14.36	1

Contrasts

```r
df.poker = df.poker %>%
  mutate(hand_contrast = factor(hand,
                                levels = c("bad", "neutral", "good"),
                                labels = c(-1, 0, 1)),
         hand_contrast = hand_contrast %>% as.character() %>% as.numeric())

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

```
Call:
lm(formula = balance ~ hand_contrast, data = df.fit)

Residuals:
    Min      1Q  Median      3Q     Max
-13.214  -2.684  -0.019   2.444  15.858

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.7715     0.2381   41.03   <2e-16 ***
hand_contrast  3.5424     0.2917   12.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.125 on 298 degrees of freedom
Multiple R-squared:  0.3311,  Adjusted R-squared:  0.3289
F-statistic: 147.5 on 1 and 298 DF,  p-value: < 2.2e-16
```

mean in neutral condition

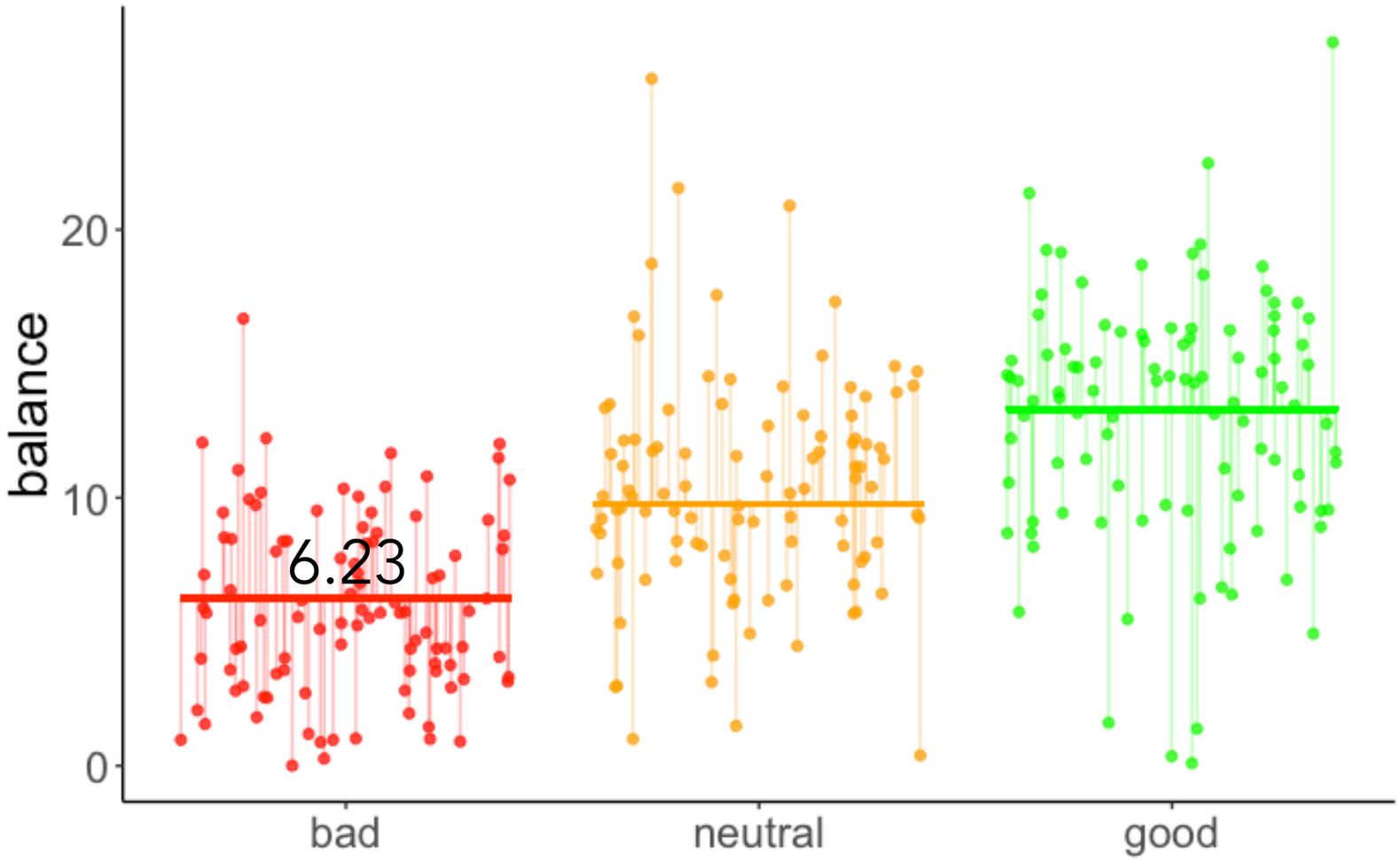
significant contrast

Understanding linear contrasts

Contrasts

name	estimate	std.error	statistic	p.value
intercept	9.77	0.24	41.03	0
contrast	3.54	0.29	12.15	0

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



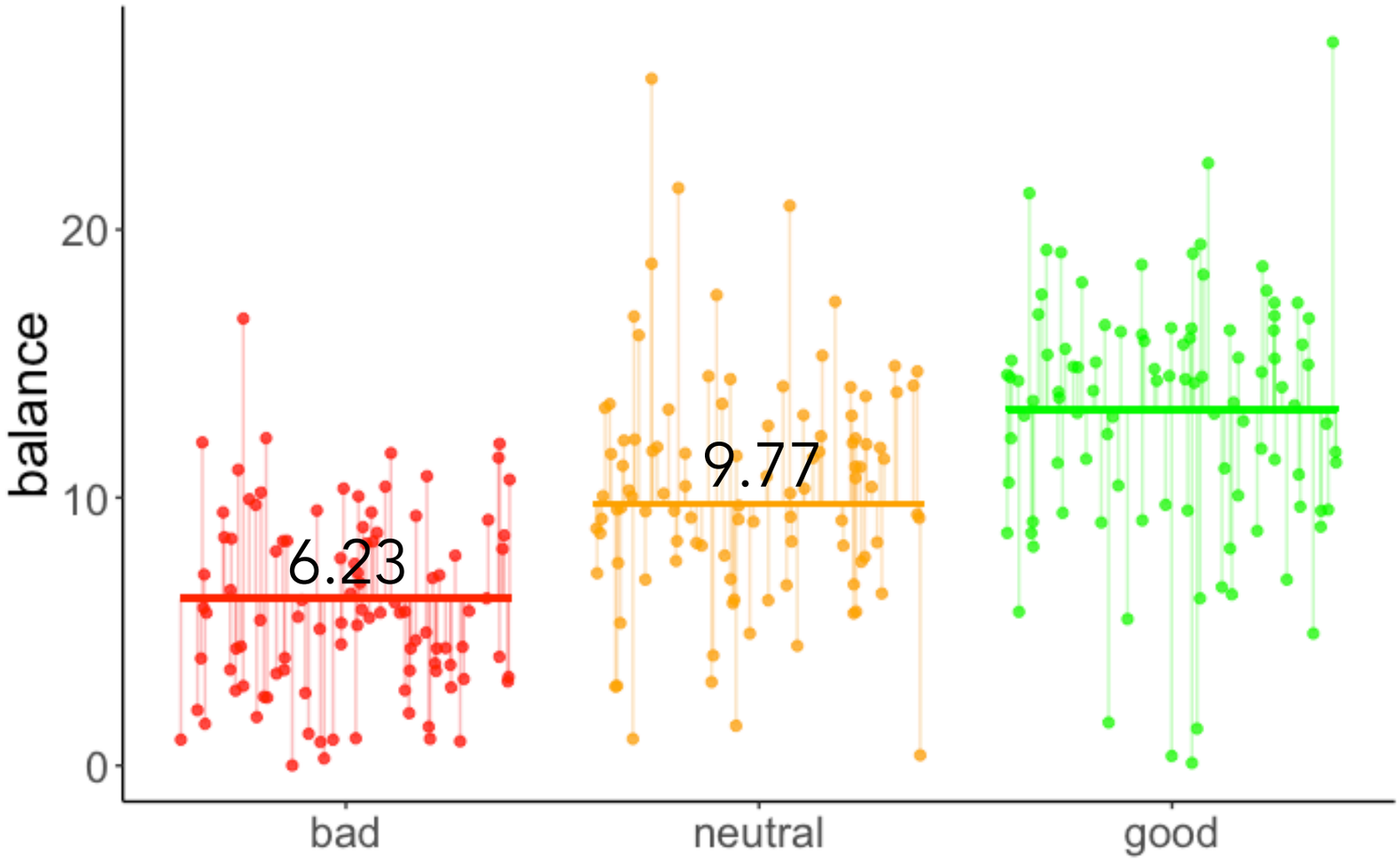
if contrast == -1

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_contrast}_i$$
$$= 9.77 + (-1) \cdot 3.54 = 6.23$$

Contrasts

name	estimate	std.error	statistic	p.value
intercept	9.77	0.24	41.03	0
contrast	3.54	0.29	12.15	0

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



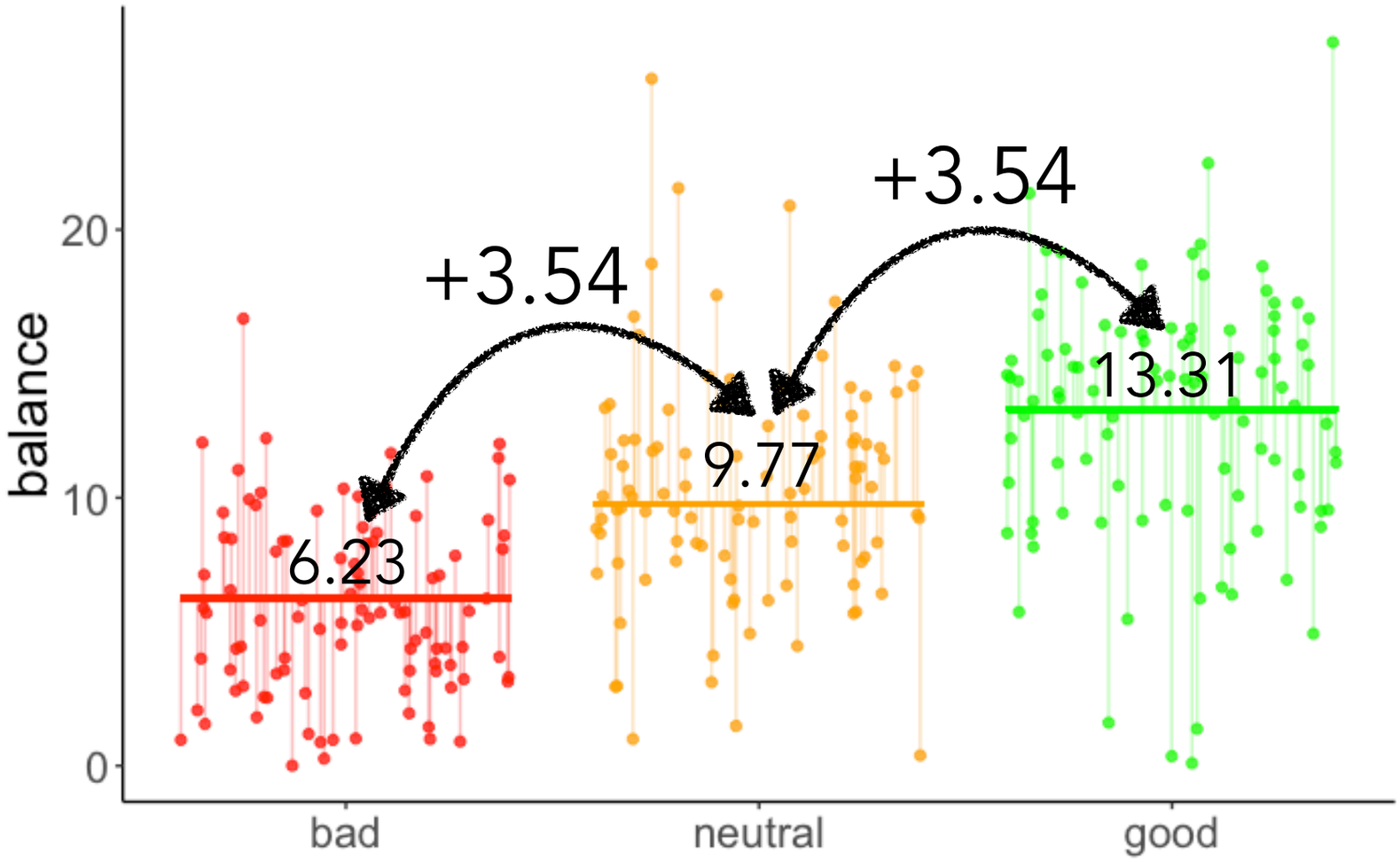
if contrast == 0

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_contrast}_i$$
$$= 9.77 + 0 \cdot 3.54 = 9.77$$

Contrasts

name	estimate	std.error	statistic	p.value
intercept	9.77	0.24	41.03	0
contrast	3.54	0.29	12.15	0

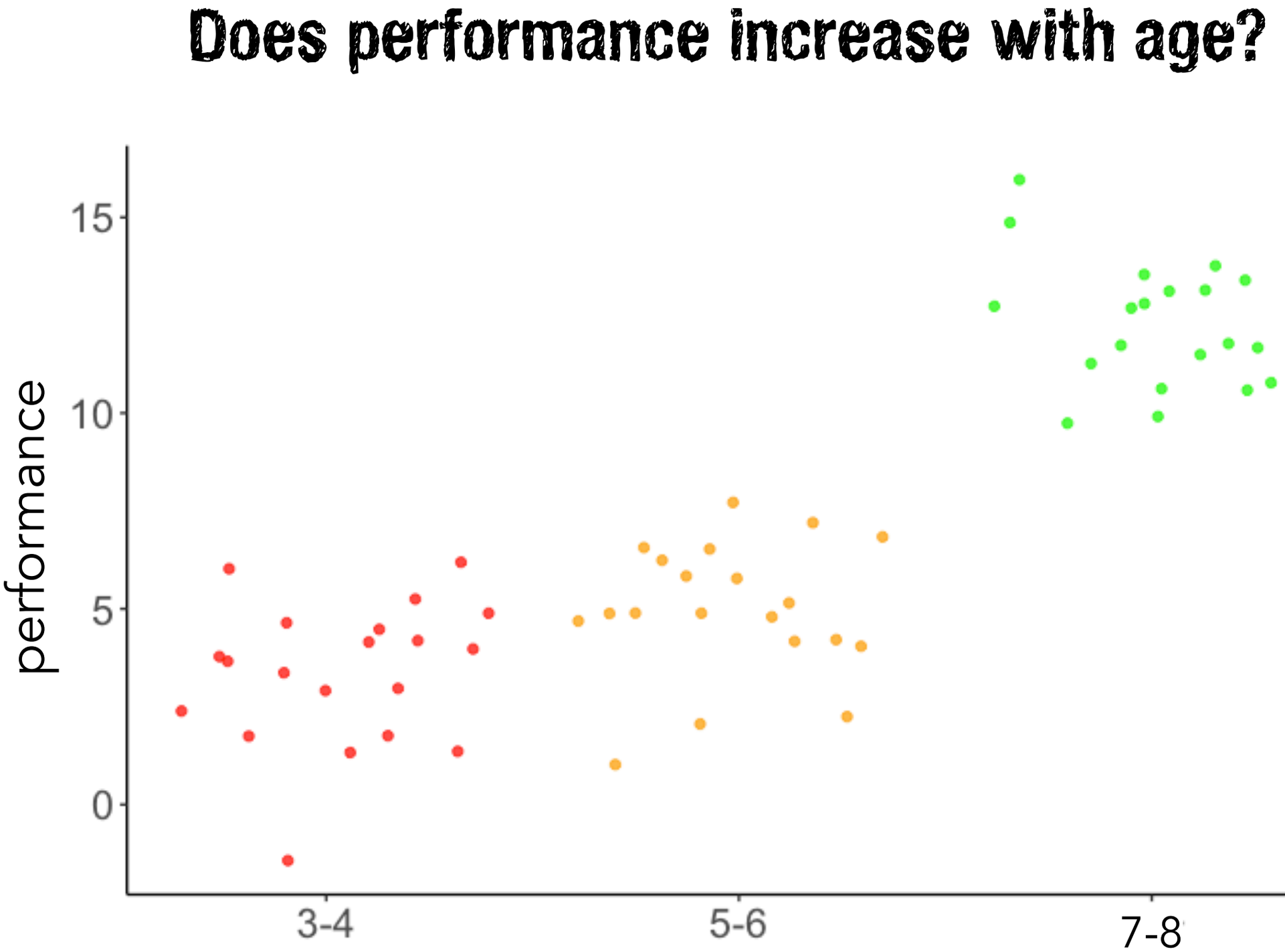
$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_contrast}$$



if contrast == 1

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_contrast}_i$$
$$= 9.77 + 1 \cdot 3.54 = 13.31$$

Contrasts



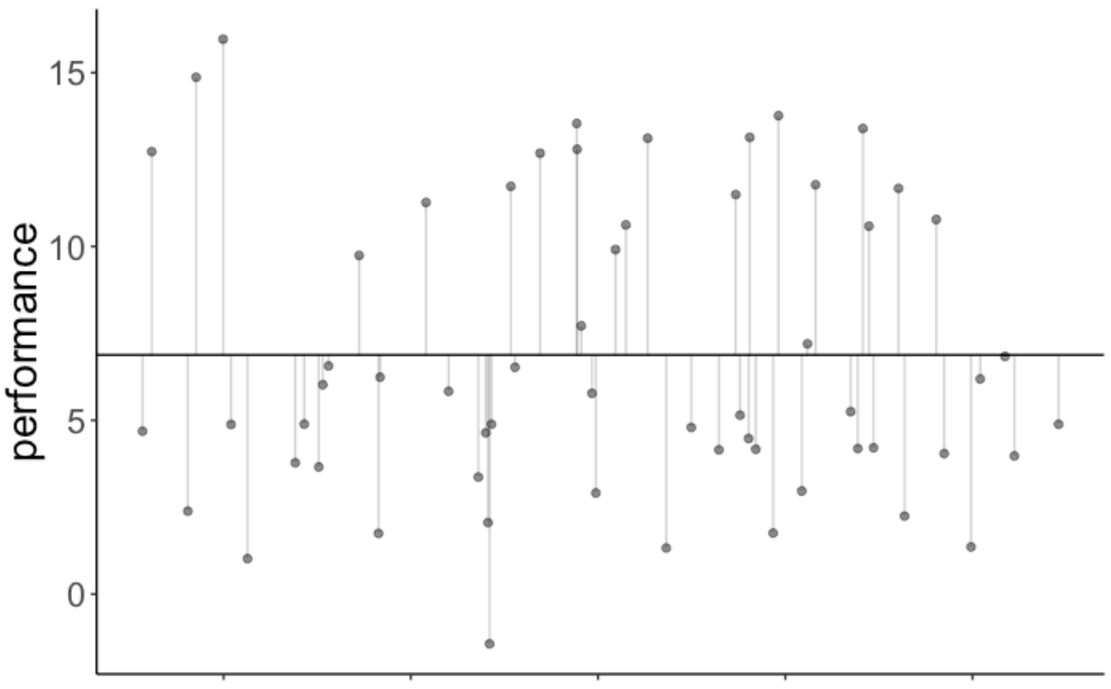
Data from a hypothetical developmental study

Contrasts

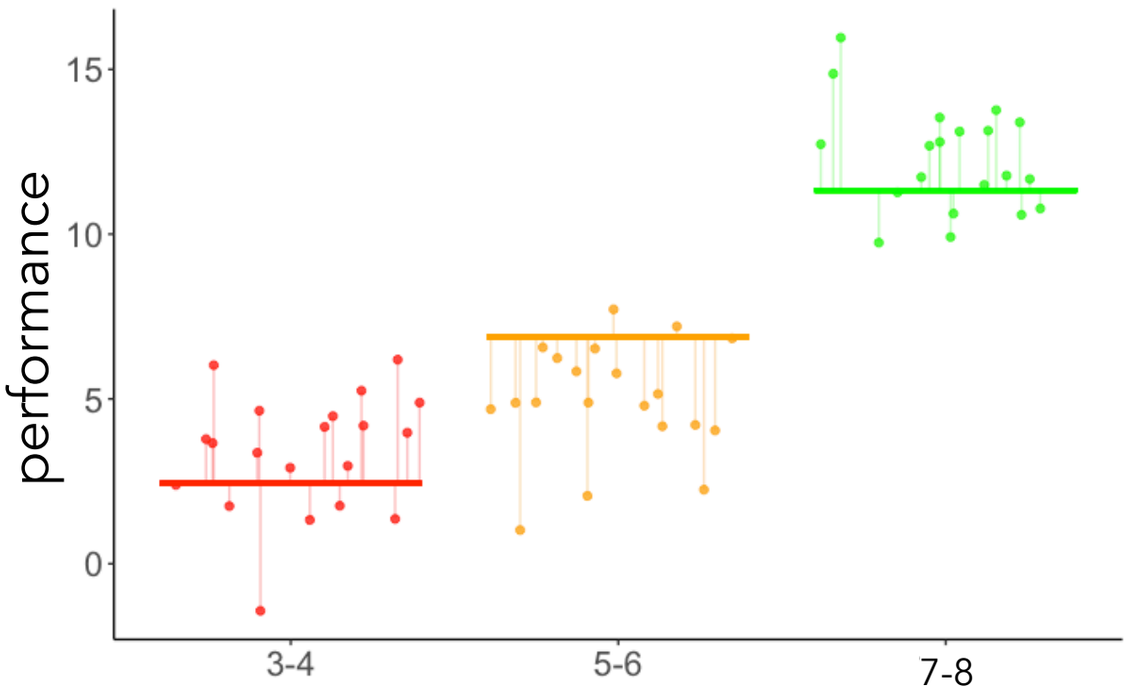
Does performance increase with age?

```
contrasts = c(-1, 0, 1)
```

Compact model



Augmented model

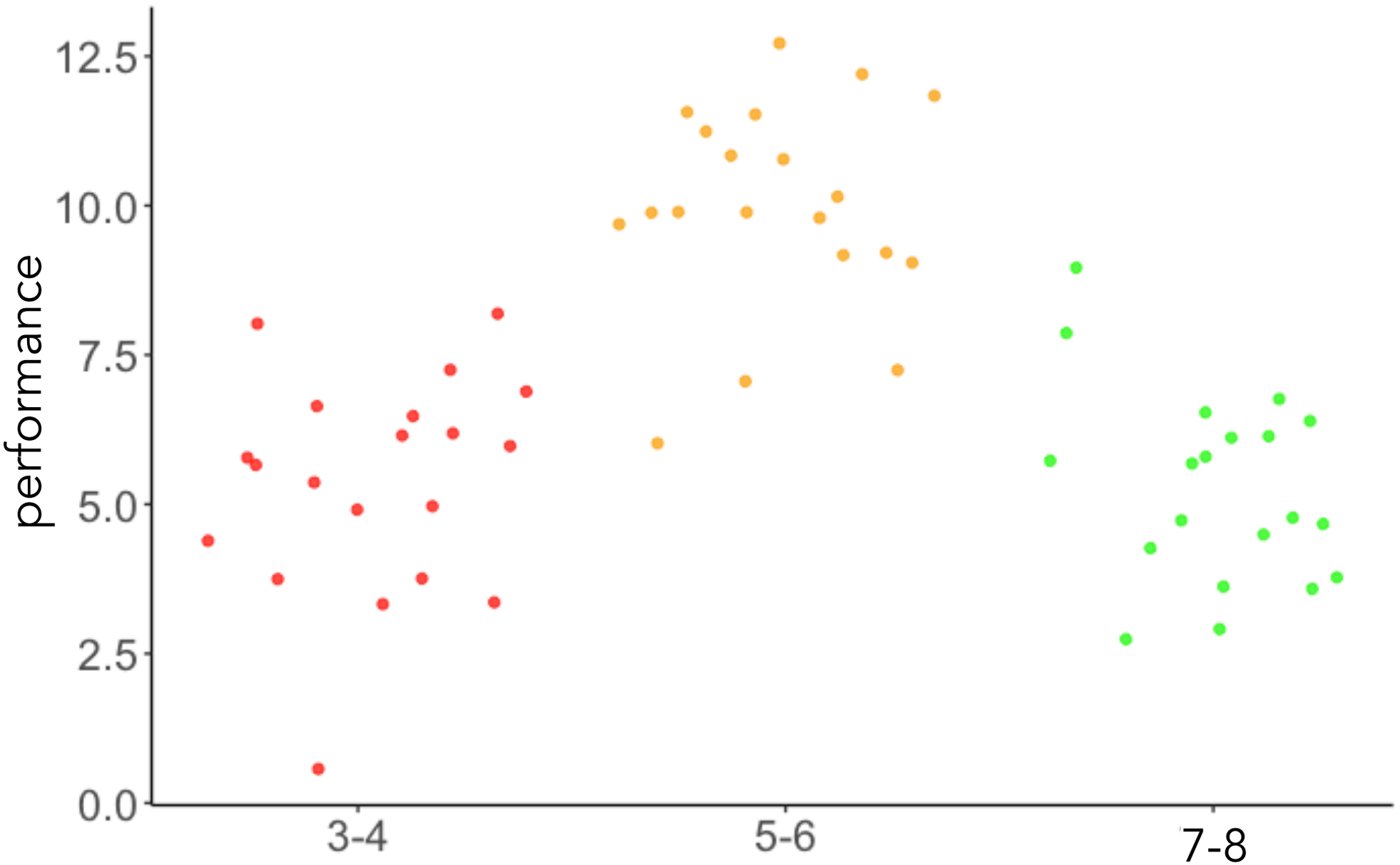


Model comparison

p < .001

Contrasts

Does performance increase with age?



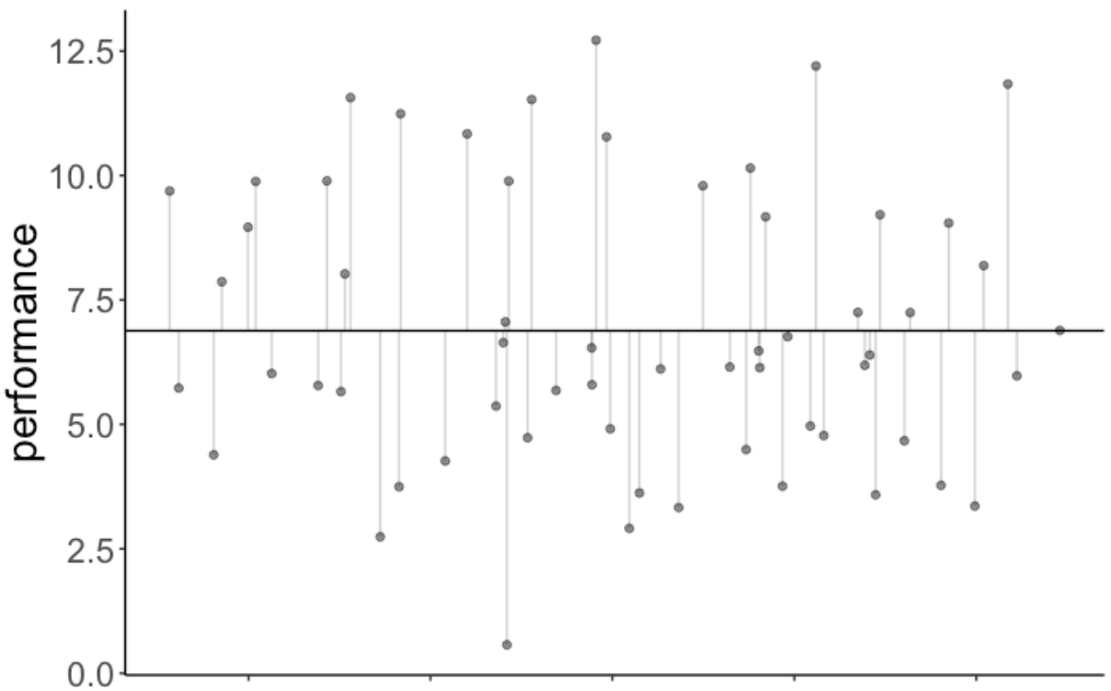
Data from another hypothetical developmental study

Contrasts

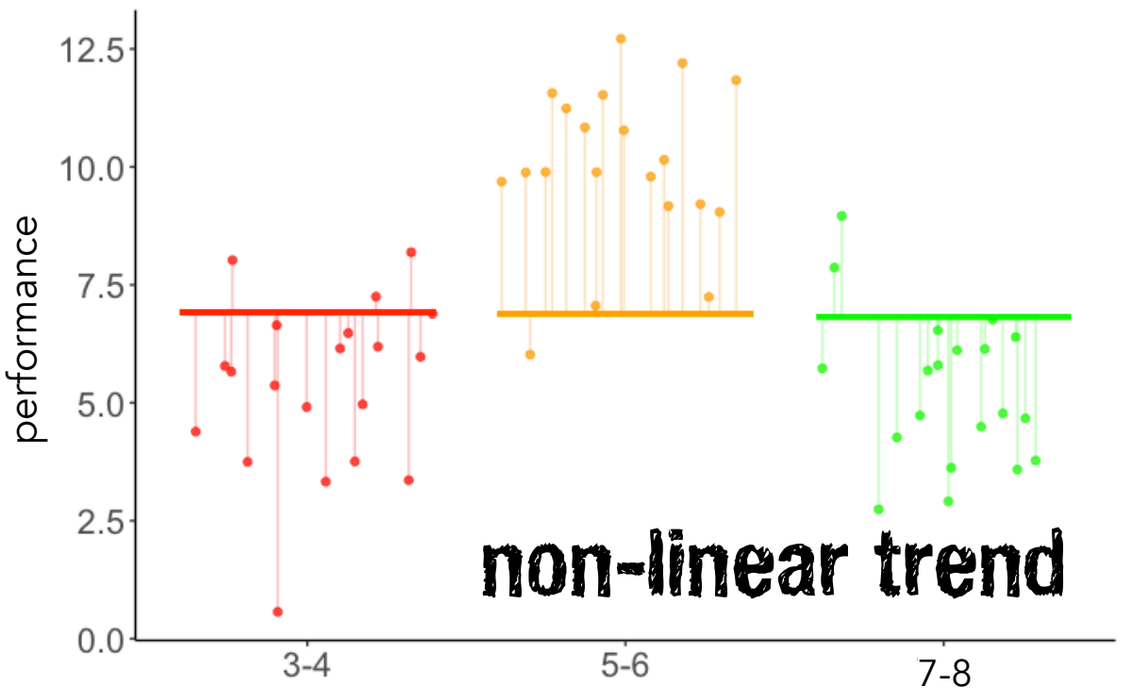
Does performance increase with age?

```
contrasts = c(-1, 0, 1)
```

Compact model



Augmented model



Model comparison

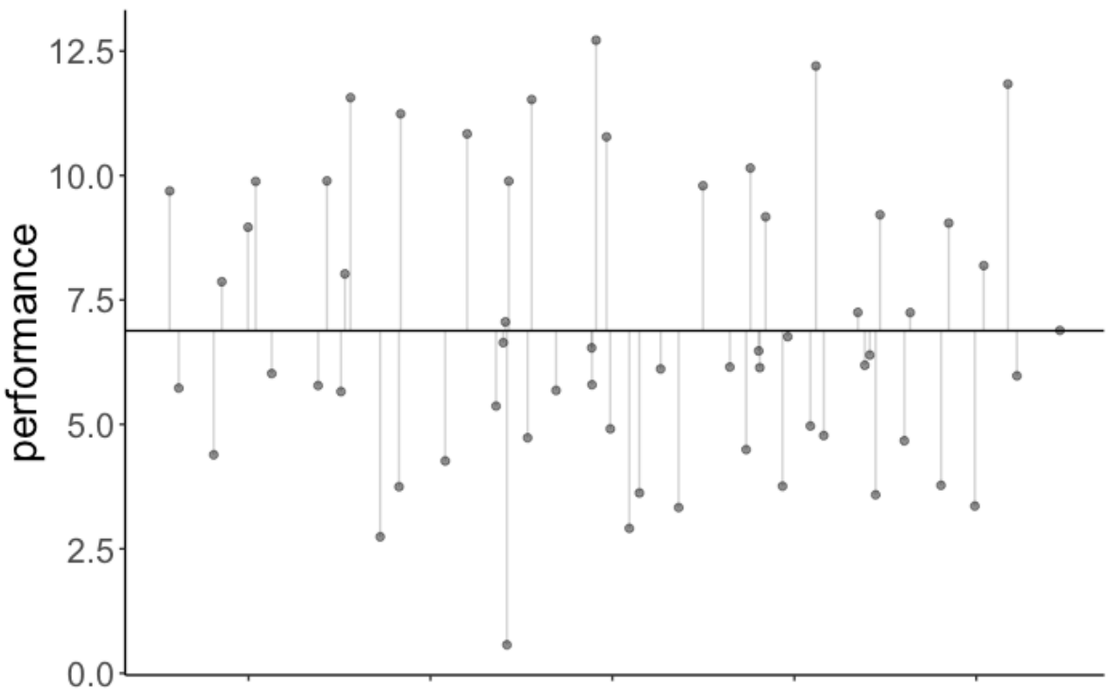
p = .8508

Contrasts

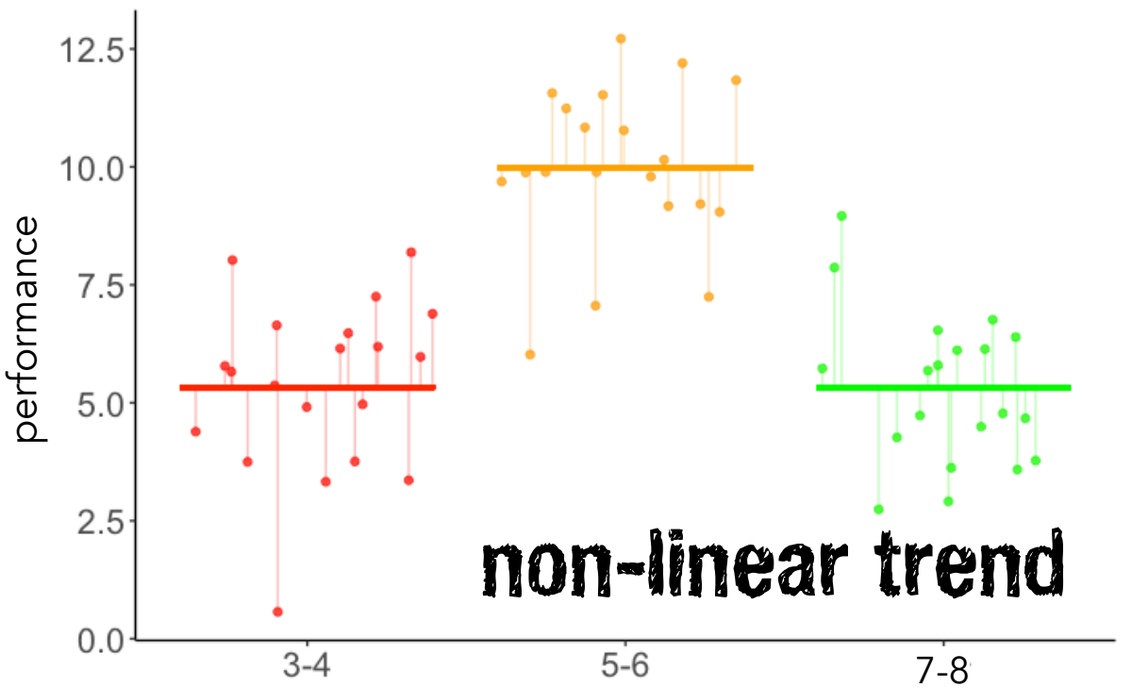
Does performance increase with age?

```
contrasts = c(-1, 2, -1)
```

Compact model



Augmented model



Model comparison

p < .001

`emmeans` for handling linear contrasts in R

Linear contrasts

~~How to use contrasts in R~~

In short: don't bother.[1]

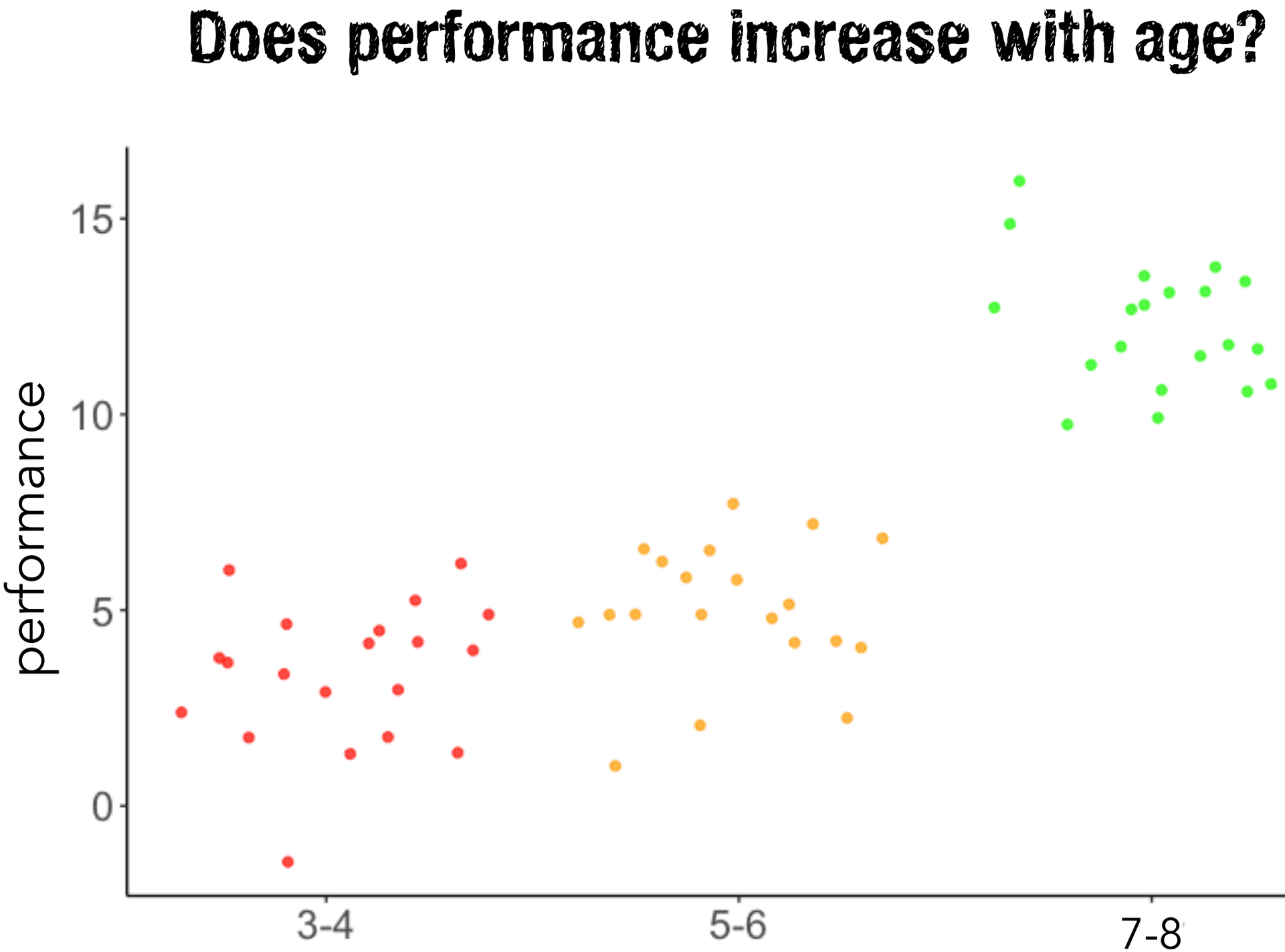
Like many before me, one of my stats classes technically "taught" me contrasts. But I didn't get the point and using them was cumbersome, so I promptly ignored them for years.

Luckily for me, someone came along and fixed the situation: `emmeans`. `emmeans` frames contrasts as a question you pose to a model: you can ask for all pairwise comparisons and get back that. `lm` and `summary` treat the same problem as fitting abstract coefficients, and you are left to answer your own question.

`emmeans` works with `lm`, `glm`, and the Bayesian friends in `brms` and `rstanarm`, so the process is applicable no matter the tool.

And you don't have to learn (much) about contrasts to take advantage of it.

Contrasts



Data from a hypothetical developmental study

Linear contrasts in R

```r
library("emmeans") # for calculating contrasts

#  fit the linear model
fit = lm(formula = performance ~ group,
         data = df.development)
```

fit linear model

Linear contrasts in R

```r
library("emmeans") # for calculating contrasts

#  fit the linear model
fit = lm(formula = performance ~ group,
         data = df.development)

# check factor levels
levels(df.development$group)
```

```
[1] "3-4" "5-6" "7-8"
```

check factor levels before defining contrasts

Linear contrasts in R

```
1  library("emmeans") # for calculating contrasts
2
3  #  fit the linear model
4  fit = lm(formula = performance ~ group,
5           data = df.development)
6
7  # check factor levels
8  levels(df.development$group) [1] "3-4" "5-6" "7-8"
9
10 # define the contrasts of interest
11 contrasts = list(young_vs_old = c(-0.5, -0.5, 1),
12                   three_vs_five = c(-0.5, 0.5, 0))
```

set up linear contrasts

Linear contrasts in R

```r
library("emmeans") # for calculating contrasts

#  fit the linear model
fit = lm(formula = performance ~ group,
         data = df.development)

# check factor levels
levels(df.development$group)

# define the contrasts of interest
contrasts = list(young_vs_old = c(-0.5, -0.5, 1),
                 three_vs_five = c(-0.5, 0.5, 0))

# compute significance test on contrasts
fit %>%
  emmeans("group",
          contr = contrasts,
          adjust = "bonferroni") %>%
  pluck("contrasts")
```

compute the results

```
[1] "3-4" "5-6" "7-8"
 contrast       estimate        SE df t.ratio p.value
 young_vs_old  16.093541 0.4742322 57  33.936  <.0001
 three_vs_five  1.606009 0.5475962 57   2.933  0.0097

P value adjustment: bonferroni method for 2 tests
```

Interpreting the coefficients

```
fit = lm(formula = performance ~ group,
         data = df.development)

# check factor levels
levels(df.development$group)

# define the contrasts of interest
contrasts = list(young_vs_old = c(-1, -1, 2),
                 three_vs_five = c(-0.5, 0.5, 0))

# compute estimated marginal means
leastsquare = emmeans(fit, "group")

# run analyses
contrast(leastsquare,
         contrasts,
         adjust = "bonferroni")
```

```
[1] "3-4" "5-6" "7-8"
 contrast      estimate    SE df t.ratio p.value
 young_vs_old    32.187 0.948 57 33.936  <.0001
 three_vs_five    0.803 0.274 57  2.933  0.0097

P value adjustment: bonferroni method for 2 tests
```

hypothesis tests are the same!

Post hoc tests

```r
fit = lm(formula = performance ~ group,
         data = df.development)

# pairwise differences between all the groups
fit %>%
  emmeans(pairwise ~ group) %>%
  pluck("contrasts")
```

all pairwise tests between groups

```
contrast    estimate        SE df t.ratio p.value
 3-4 - 5-6  -1.606009 0.5475962 57  -2.933  0.0145
 3-4 - 7-8 -16.896546 0.5475962 57 -30.856  <.0001
 5-6 - 7-8 -15.290537 0.5475962 57 -27.923  <.0001

P value adjustment: bonferroni method for 3 tests
```

Post hoc tests

```
1 # fit the model
2 fit = lm(formula = balance ~ hand + skill,
3           data = df.poker)
4
5 # post hoc tests
6 fit %>%
7   emmeans(pairwise ~ hand + skill,
8           adjust = "bonferroni") %>%
9   pluck("contrasts")
```

the poker data

contrast	estimate	SE	df	t.ratio	p.value
bad,average - neutral,average	-4.381023	0.6051766	286	-7.239	<.0001
bad,average - good,average	-7.060823	0.6051766	286	-11.667	<.0001
bad,average - bad,expert	-0.740385	0.4896119	286	-1.512	1.0000
bad,average - neutral,expert	-5.121408	0.7611327	286	-6.729	<.0001
bad,average - good,expert	-7.801208	0.7611327	286	-10.249	<.0001
neutral,average - good,average	-2.679800	0.5884403	286	-4.554	0.0001
neutral,average - bad,expert	3.640638	0.7953578	286	4.577	0.0001
neutral,average - neutral,expert	-0.740385	0.4896119	286	-1.512	1.0000
neutral,average - good,expert	-3.420185	0.7654945	286	-4.468	0.0002
good,average - bad,expert	6.320438	0.7953578	286	7.947	<.0001
good,average - neutral,expert	1.939415	0.7654945	286	2.534	0.1774
good,average - good,expert	-0.740385	0.4896119	286	-1.512	1.0000
bad,expert - neutral,expert	-4.381023	0.6051766	286	-7.239	<.0001
bad,expert - good,expert	-7.060823	0.6051766	286	-11.667	<.0001
neutral,expert - good,expert	-2.679800	0.5884403	286	-4.554	0.0001

P value adjustment: bonferroni method for 15 tests

all pairwise tests between groups

that's a lot of tests!

... not

Contrasts

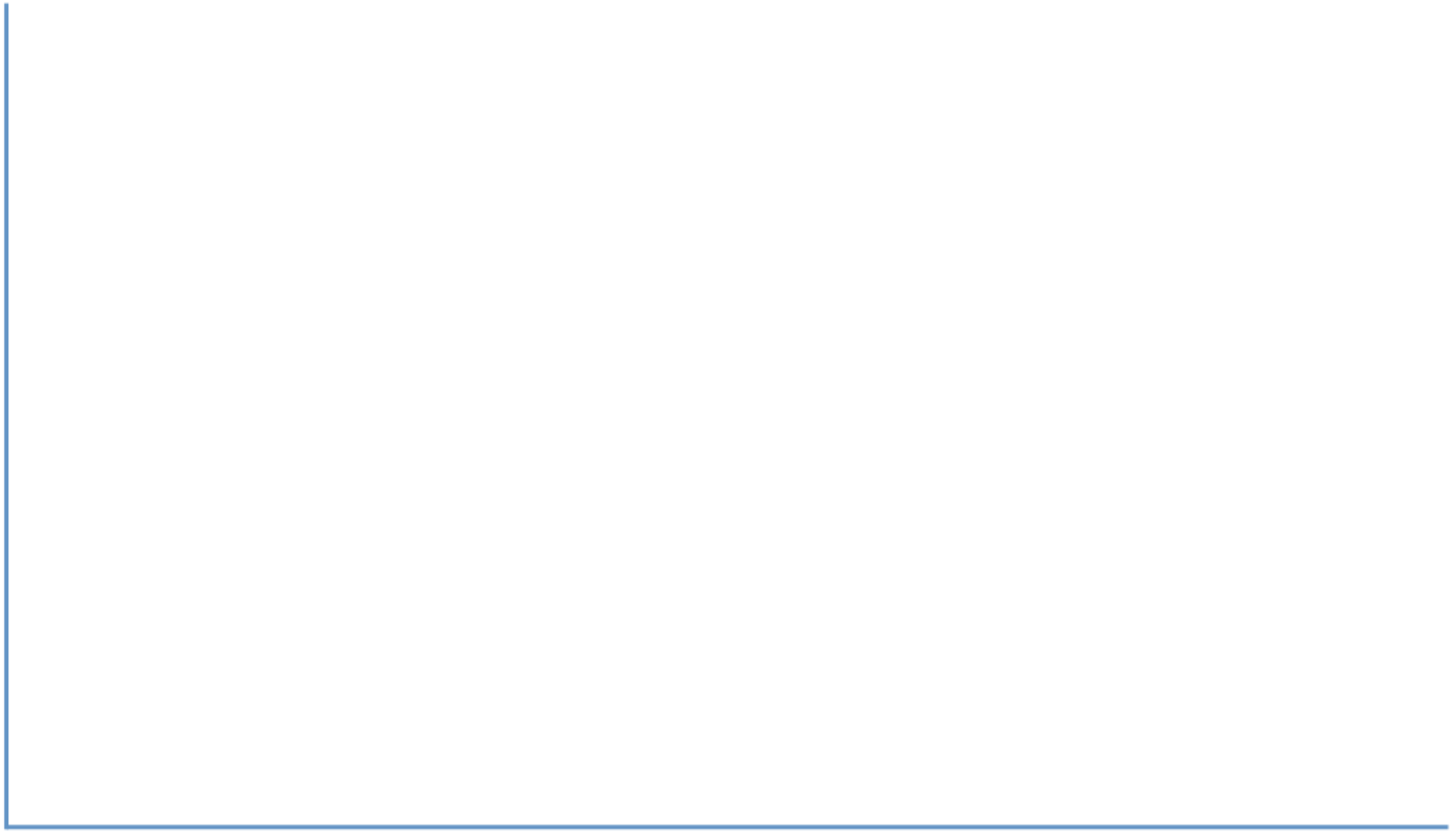
- linear contrasts allow us to ask more specific questions of our data
- rather than asking whether any of the group means are significantly different from each other (ANOVA), we can ask questions such as:
  - Does performance increase with age?
  - Is the overall performance in Condition B and C better from the performance in Condition A?

Plan for today

- Analysis of Variance (ANOVA)
  - multiple categorical predictors (N-way ANOVA)
    - interpreting parameters
    - Who is the ANOVA champ?
    - unbalanced designs
- Linear contrasts
  - testing specific hypotheses with linear contrasts
  - `emmeans` for handling linear contrasts in R

Feedback

How was the pace of today's class?



How happy were you with today's class overall?

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

Thank you!