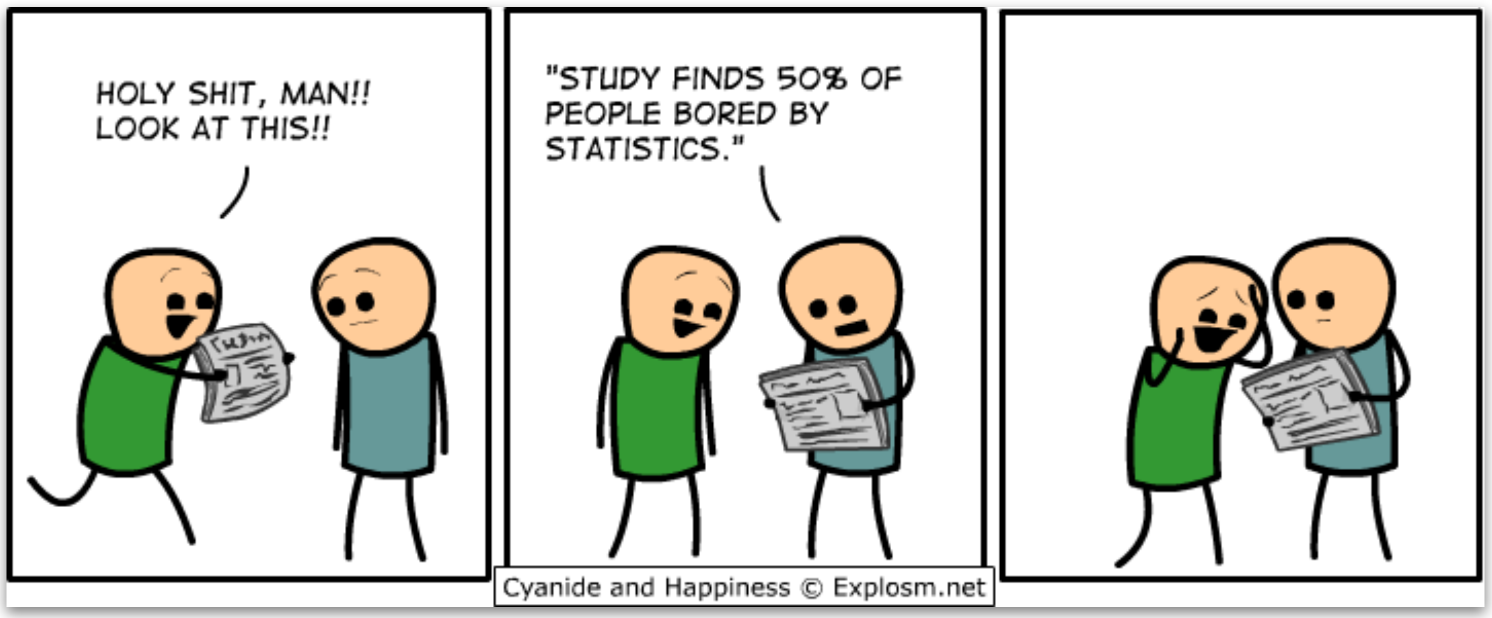
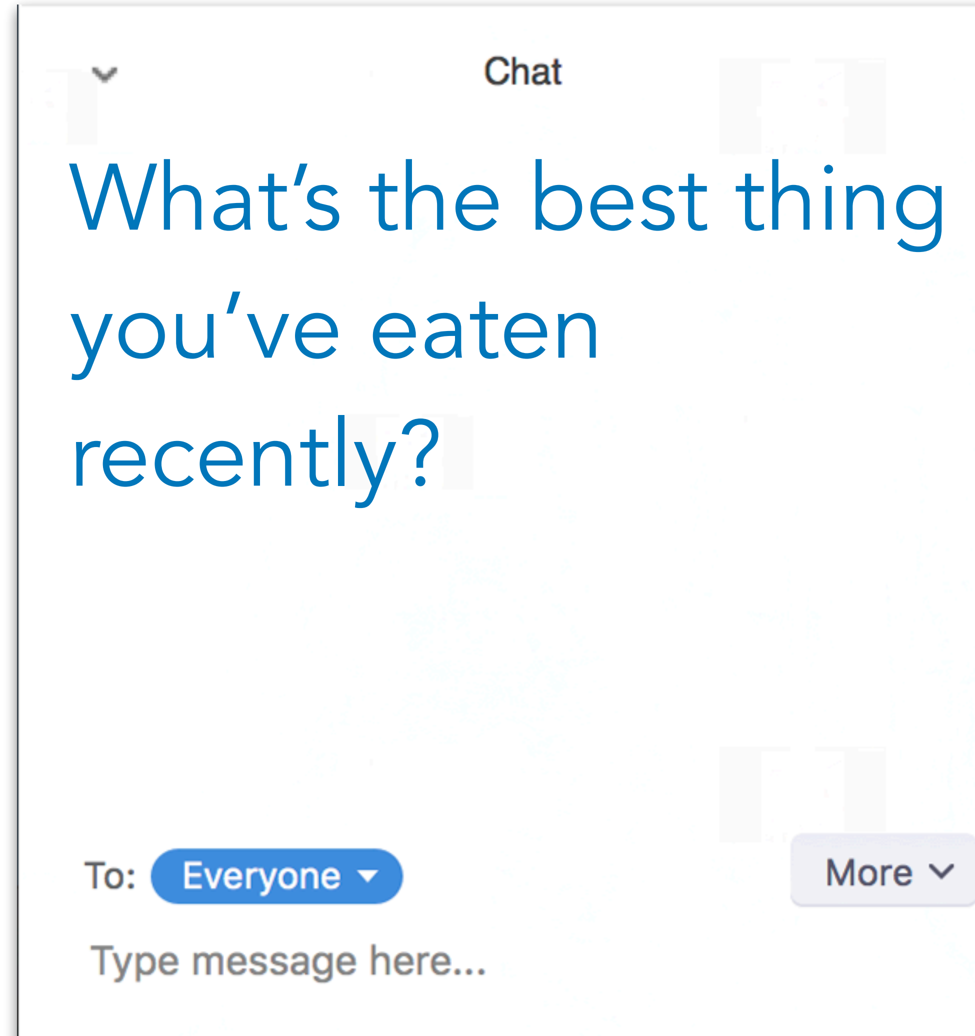
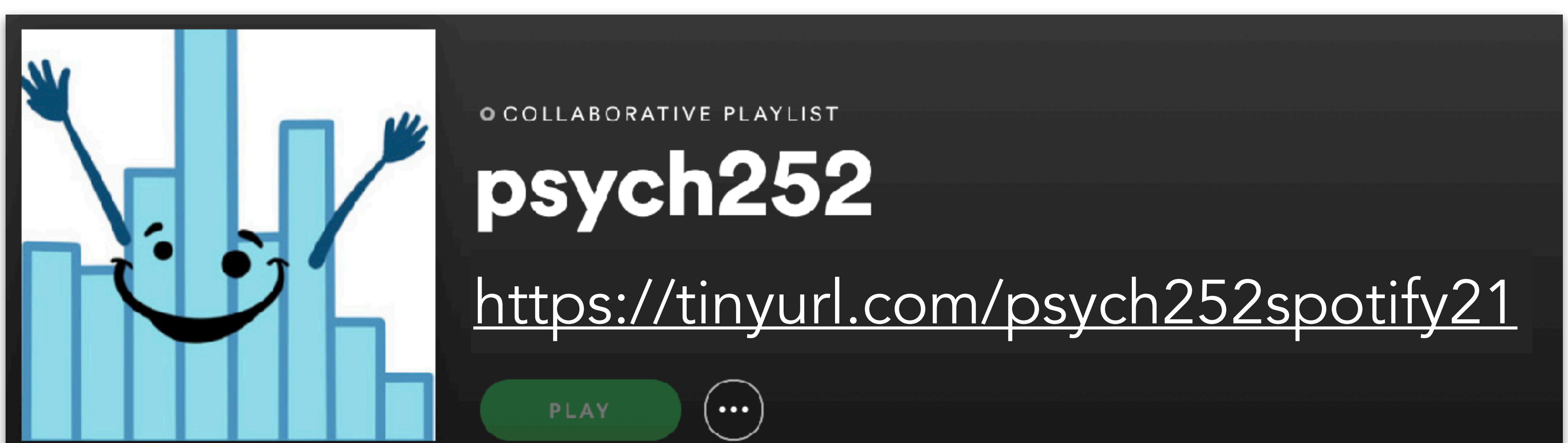
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Linear model 4







We're listening to "Slip" by "Shubh Saran" submitted by Ben

02/02/2022

Plan for today

- Quick recap
- `lm()` output
- Analysis of Variance (ANOVA)
    - multiple categorical predictors (N-way ANOVA)
        - interpreting parameters
    - unbalanced designs
- Linear contrasts
    - testing specific hypotheses with linear contrasts
    - `emmeans` for handling linear contrasts in R

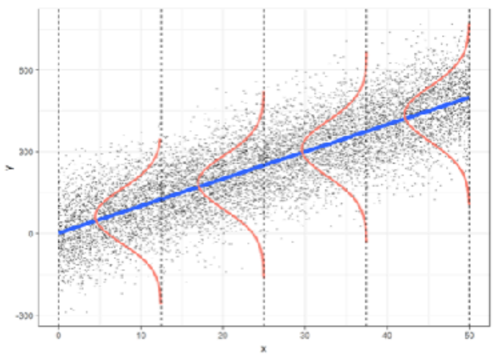
Quick recap

Quick recap: Multiple regression

Assumptions of **multiple** regression

- independent observations
- Y is continuous
- errors are normally distributed
- errors have constant variance
- error terms are uncorrelated
- **no multicollinearity**

predictors in the model should not be highly correlated with each other



18

H₀: Radio ads and sales are not related once we control for TV ads.

Model C

$\text{sales}_i = b_0 + b_1 \cdot \text{tv}_i + e_i$

H₁: Radio ads and sales are related even when we control for TV ads.

Model A

$\text{sales}_i = b_0 + b_1 \cdot \text{tv}_i + b_2 \cdot \text{radio}_i + e_i$

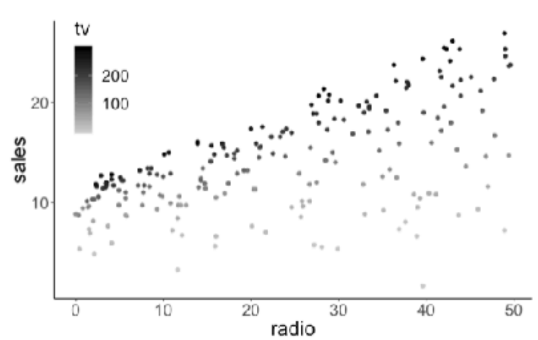
```r
# fit the models
fit_c = lm(sales ~ 1 + tv, data = df.ads)
fit_a = lm(sales ~ 1 + tv + radio, data = df.ads)

# do the F test
anova(fit_c, fit_a)
```

we reject the H₀

```
Analysis of Variance Table

Model 1: sales ~ 1 + tv
Model 2: sales ~ 1 + tv + radio
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1    198 2102.53
2    197  556.91  1    1545.6 546.74 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

26

Reporting results



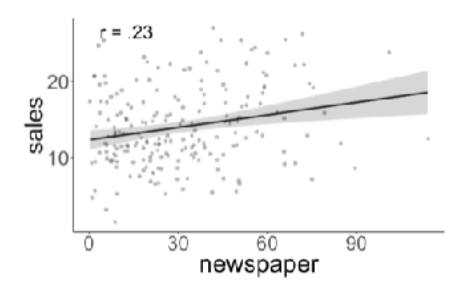
There is a significant relationship between sales and radio ads, controlling for TV ads $F(1, 197) = 546.74$, $p < .001$.

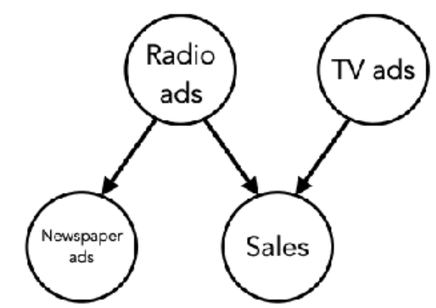
Holding TV ads fixed, an increase in \$1000 on radio ads is predicted to increase sales by 190 units [170, 200] (95% confidence intervals).

31

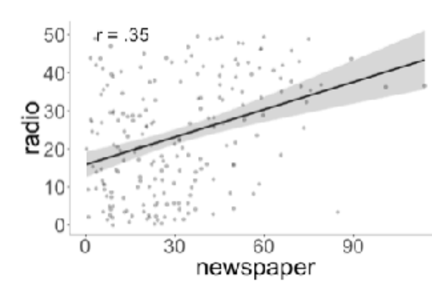
Are newspaper ads and sales related when controlling for radio ads and TV ads?

Relationship between newspaper ads and sales

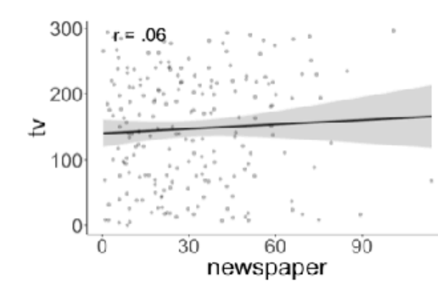




Relationship between newspaper and radio ads



Relationship between newspaper and TV ads



38

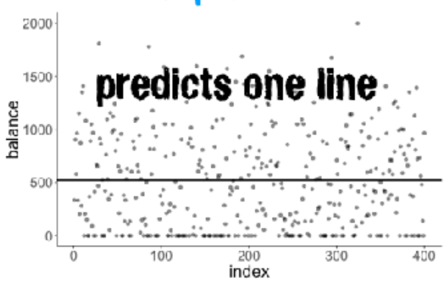
Quick recap: Categorical predictors

H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



Fitted model

$$Y_i = 520.02 + e_i$$

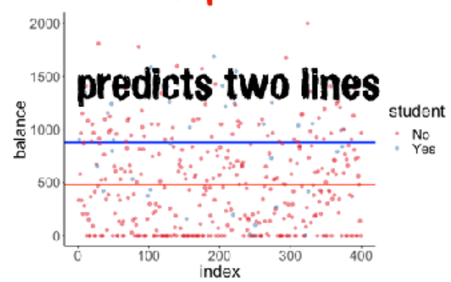
H_1: Students and non-students have different balances.

Model A

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

student

Model prediction



Fitted model

$$Y_i = 480.37 + 396.46 X_i + e_i$$

41

Interpreting the model

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

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

Residuals:
    Min      1Q  Median      3Q     Max
-876.82 -458.82  -40.87  341.88 1518.63

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   480.37      23.43   20.50  < 2e-16 ***
studentYes    396.46      74.10    5.35 1.49e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 444.6 on 398 degrees of freedom
Multiple R-squared:  0.06709,	Adjusted R-squared:  0.06475
F-statistic: 28.62 on 1 and 398 DF,  p-value: 1.488e-07
```

45

Dummy coding

46

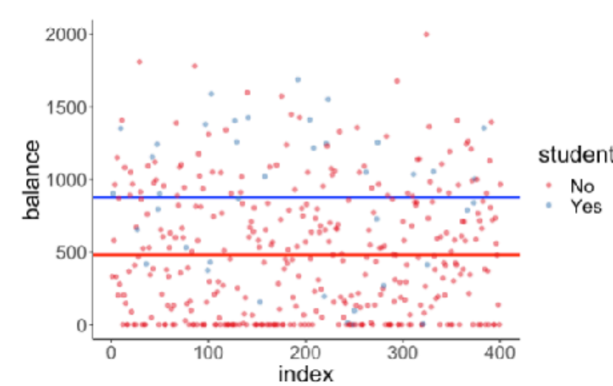
Dummy coding

$$\hat{Y}_i = 480.37 + 396.46 \cdot \text{student\_dummy}_i$$

if student = "No" $\hat{Y}_i = 480.37$

if student = "Yes" $\hat{Y}_i = 480.37 + 396.46 = 876.83$

student	student_dummy
No	0
Yes	1
No	0
No	0
No	0
No	0
No	0
No	0
No	0
Yes	1



48

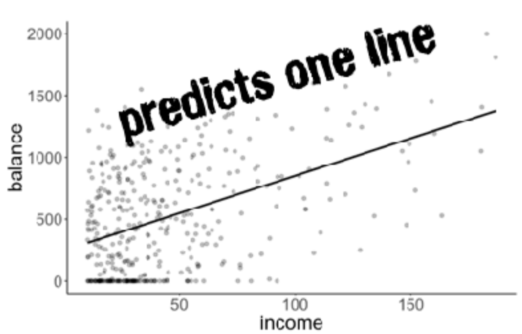
Quick recap: Categorical and continuous predictors

H_0: Students and non-students have the same balance, when controlling for income.

Model C

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

Model prediction



Fitted model

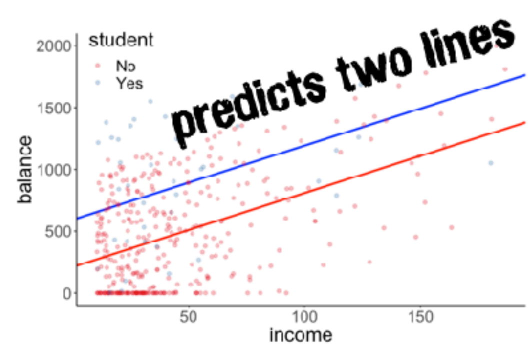
$$\widehat{\text{balance}}_i = 246.515 + 6.048 \cdot \text{income}_i$$

H_1: Students and non-students have different balances, when controlling for income.

Model A

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

Model prediction



Fitted model

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

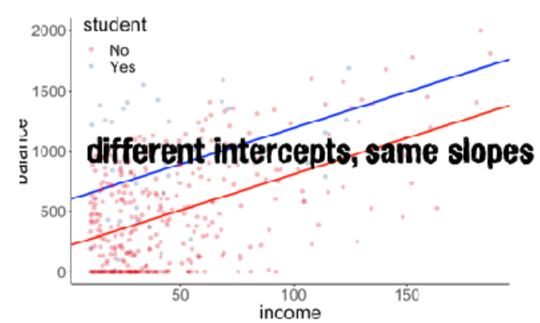
52

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

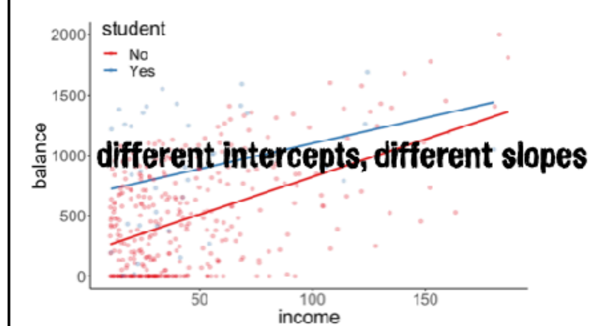
$$\widehat{\text{balance}}_i = 211.14 + 5.98 \cdot \text{income}_i + 382.67 \cdot \text{student}_i$$

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

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

59

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

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

if student = "Yes"

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

61

`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)
```

```
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
```

deterministic mapping between t and F

$$t^2 = F$$

$$10.44^2 = 108.99$$

`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
```

`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
- 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

`lm()` output

```
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
```

what does this mean?

not the overall effect of income!!

instead the predicted effect of income for non-students!

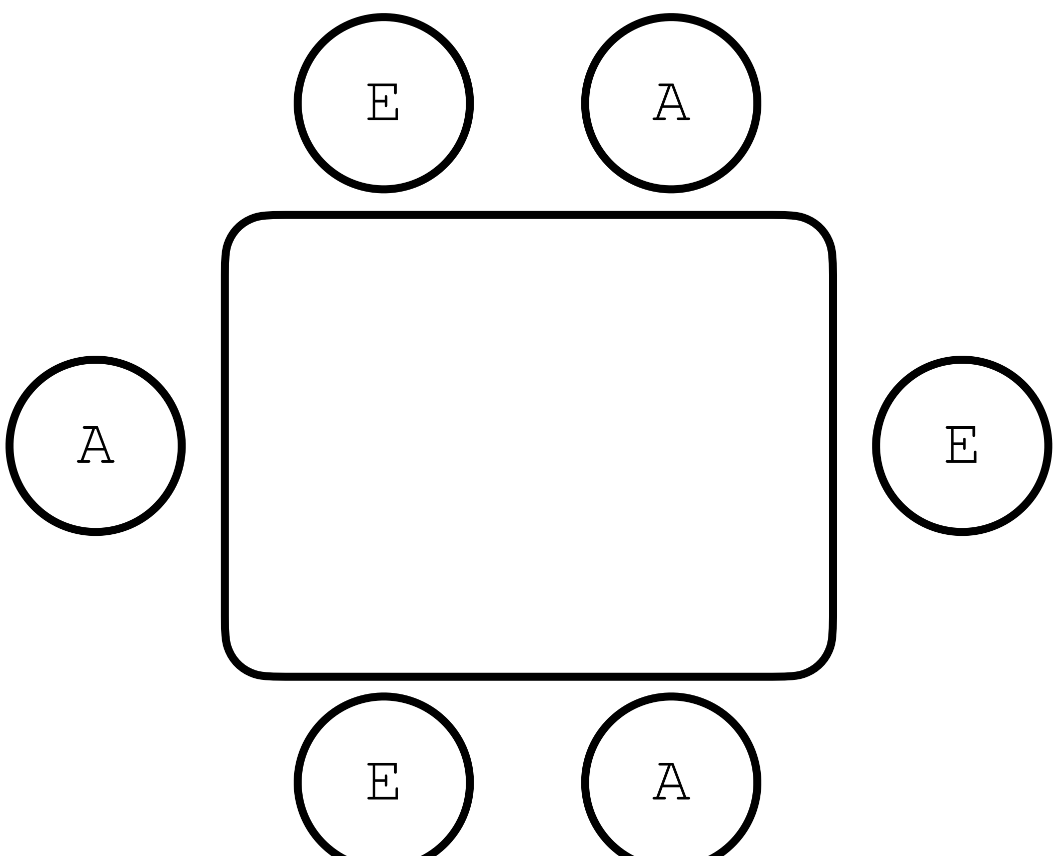
we'll talk more about the difference between simple/conditional effects and main effects next time!

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

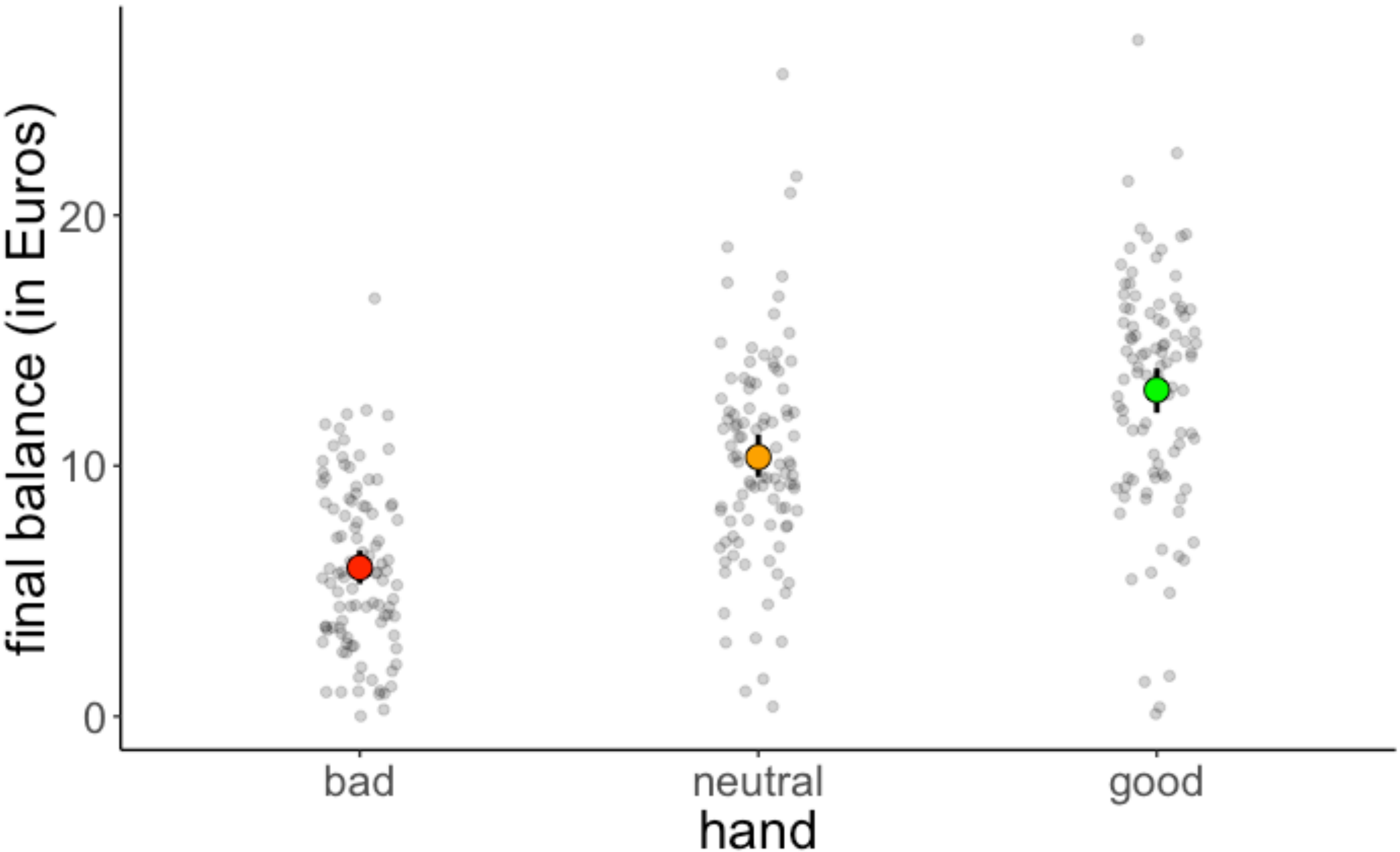
Meyer, G., von Meduna, M., Brosowski, T., & Hayer, T. (2012). Is poker a game of skill or chance? A quasi-experimental study. *Journal of Gambling Studies*

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

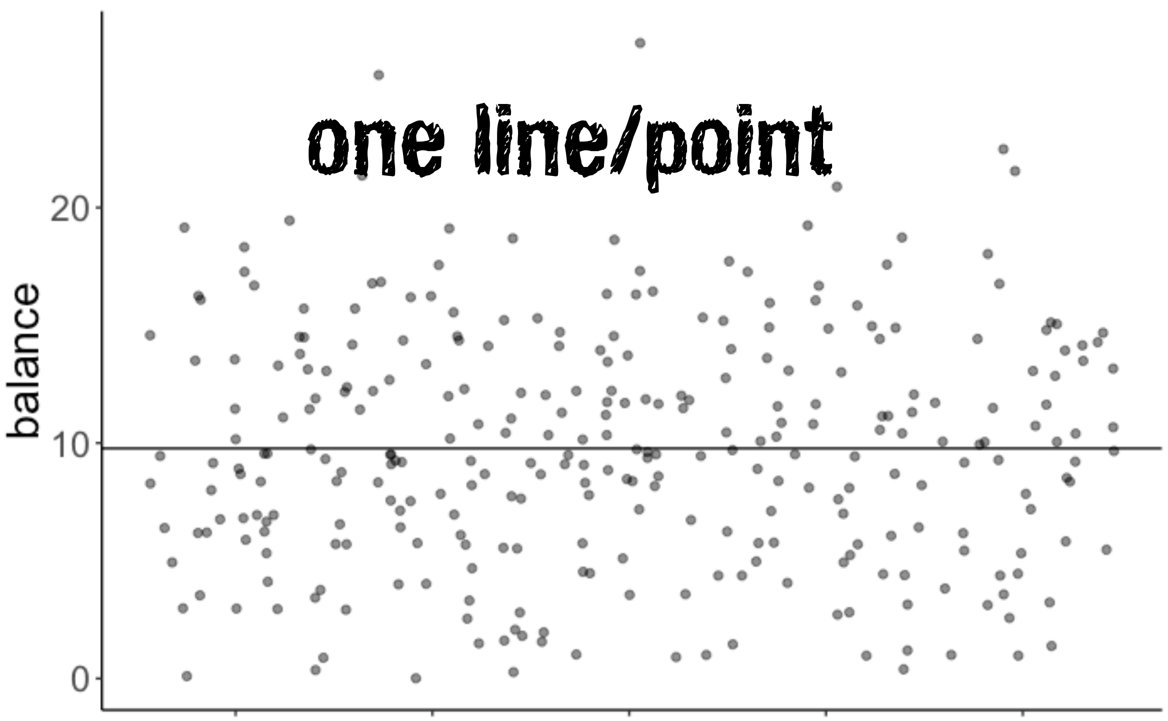


H_0: Card quality does not affect the final balance.

Model C

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

Model prediction



Fitted model

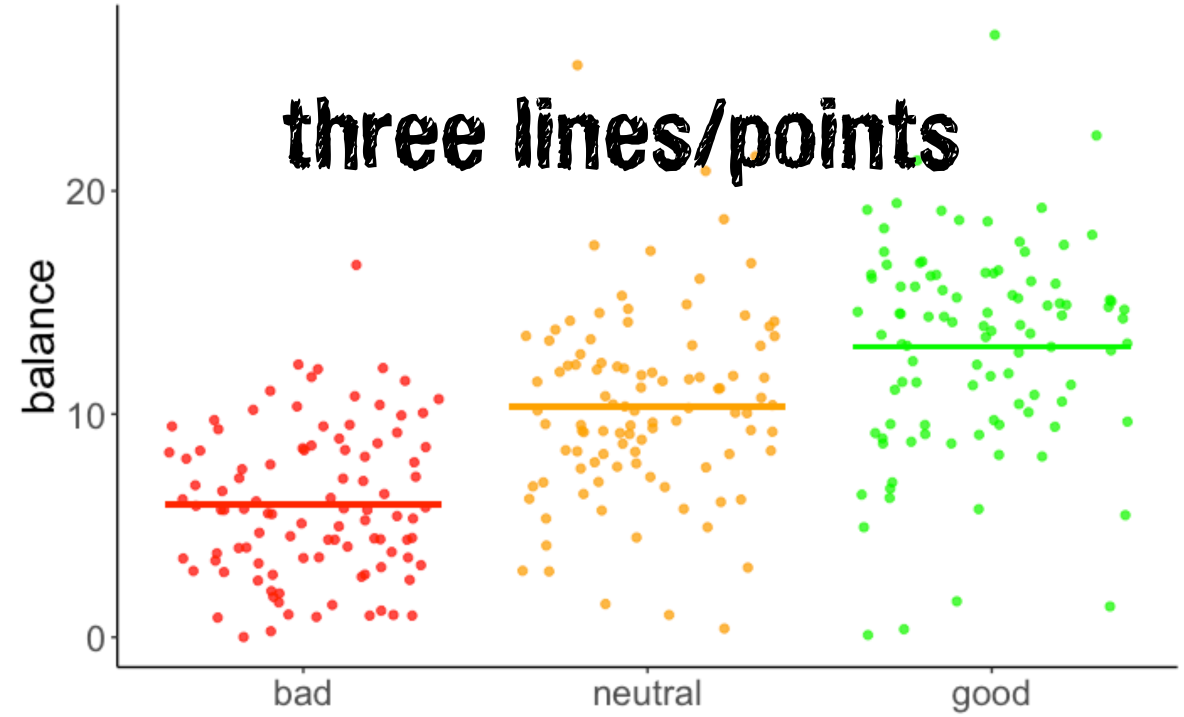
$$\widehat{\text{balance}}_i = 9.77$$

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

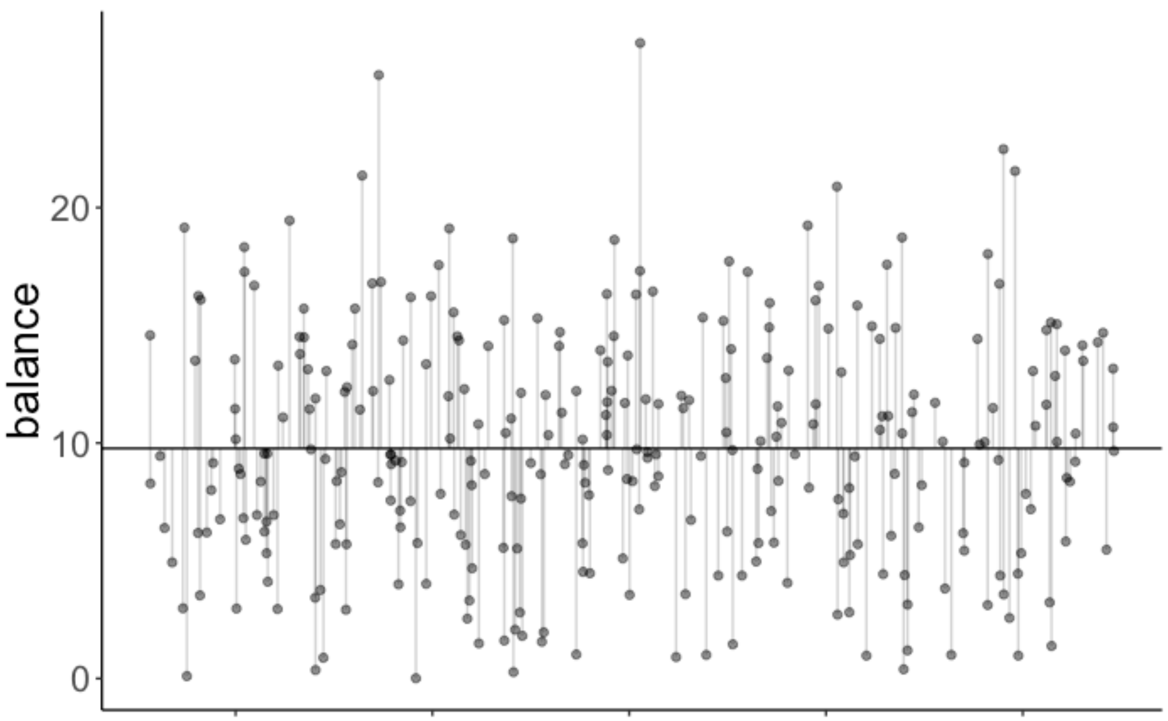
$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$

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

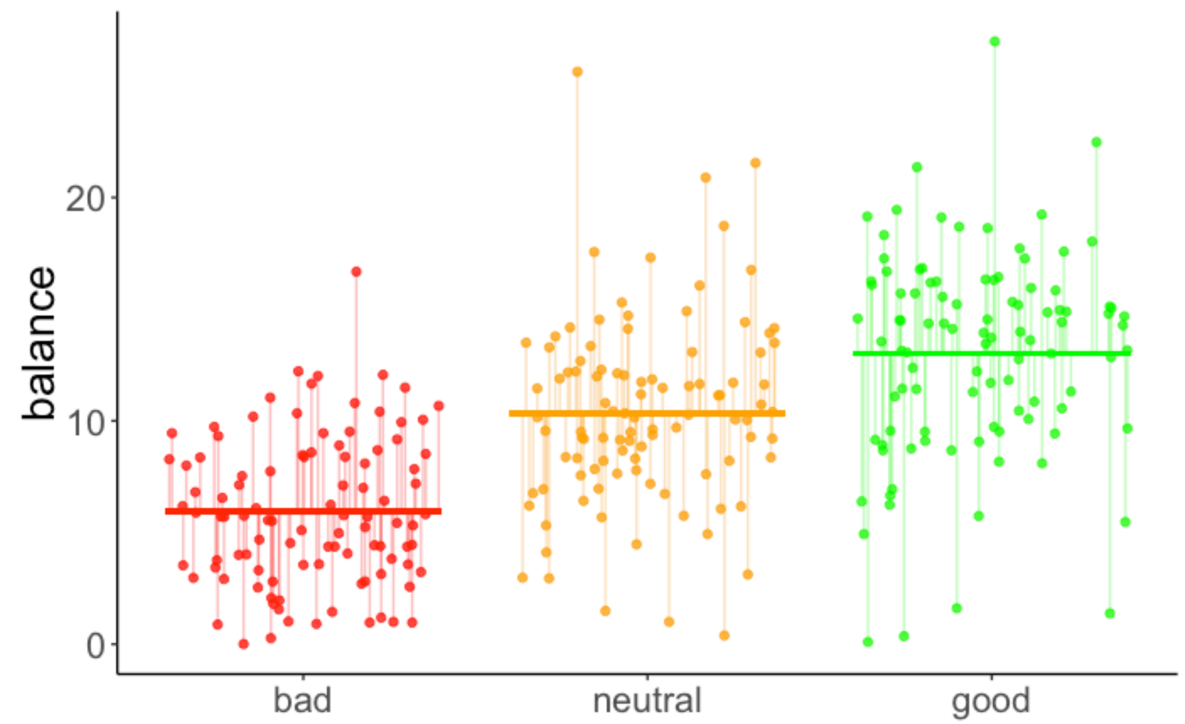
$$\widehat{\text{balance}}_i = 9.77$$

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

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

regression coefficients encode differences between group means

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

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":

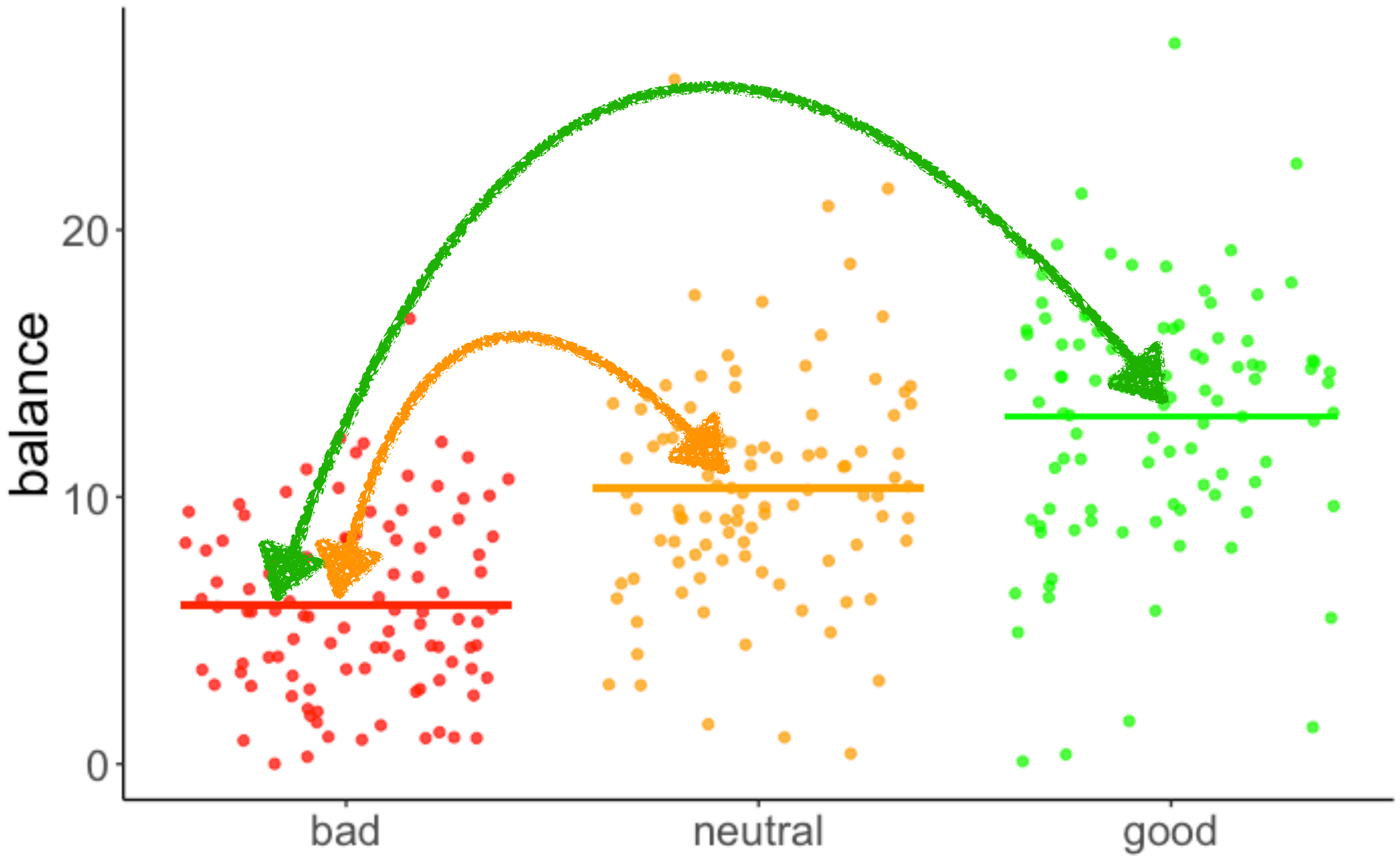
$$\widehat{\text{balance}}_i = 5.94 + 7.08 = 13.02$$

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?

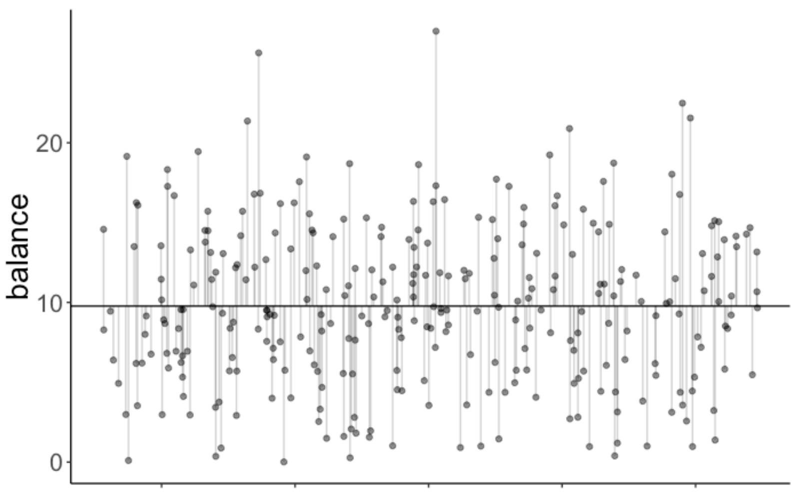
```
# fit the models
fit_c = lm(formula = balance ~ 1, data = df.poker)
fit_a = lm(formula = balance ~ hand, data = df.poker)

# 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

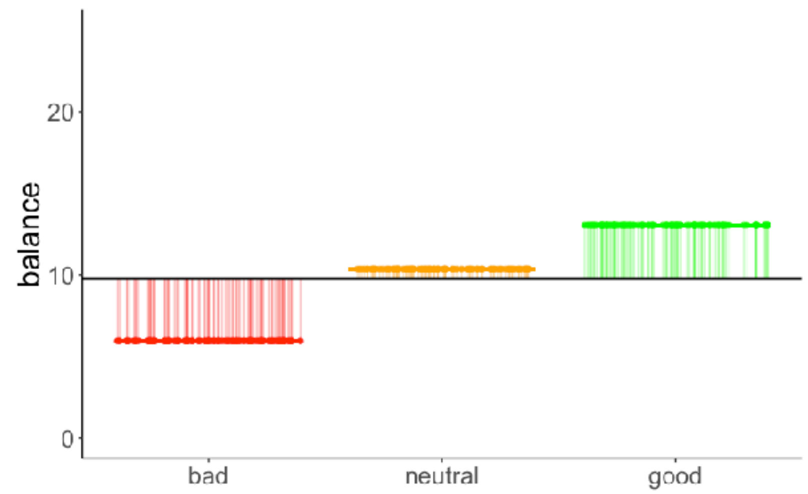
Variance decomposition

Total variance



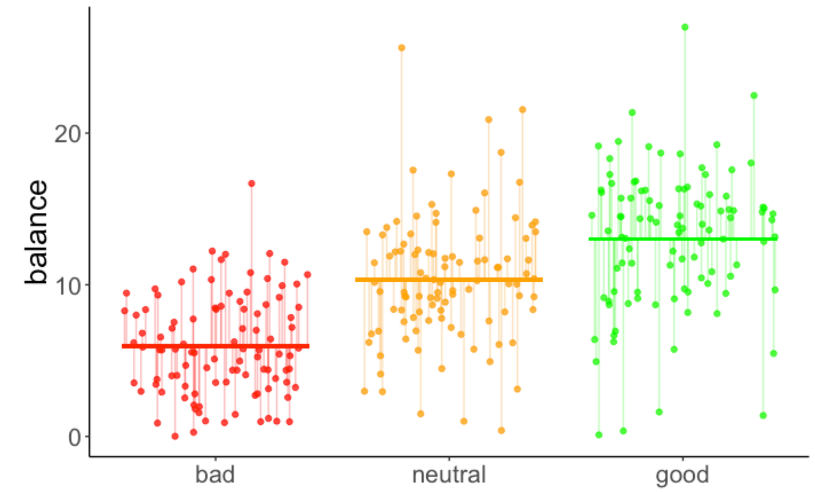
SS_{total}

Model variance



SS_{model}

Residual variance



$SS_{residual}$

variance_total	variance_model	variance_residual
7580	2559	5021

One-way ANOVA

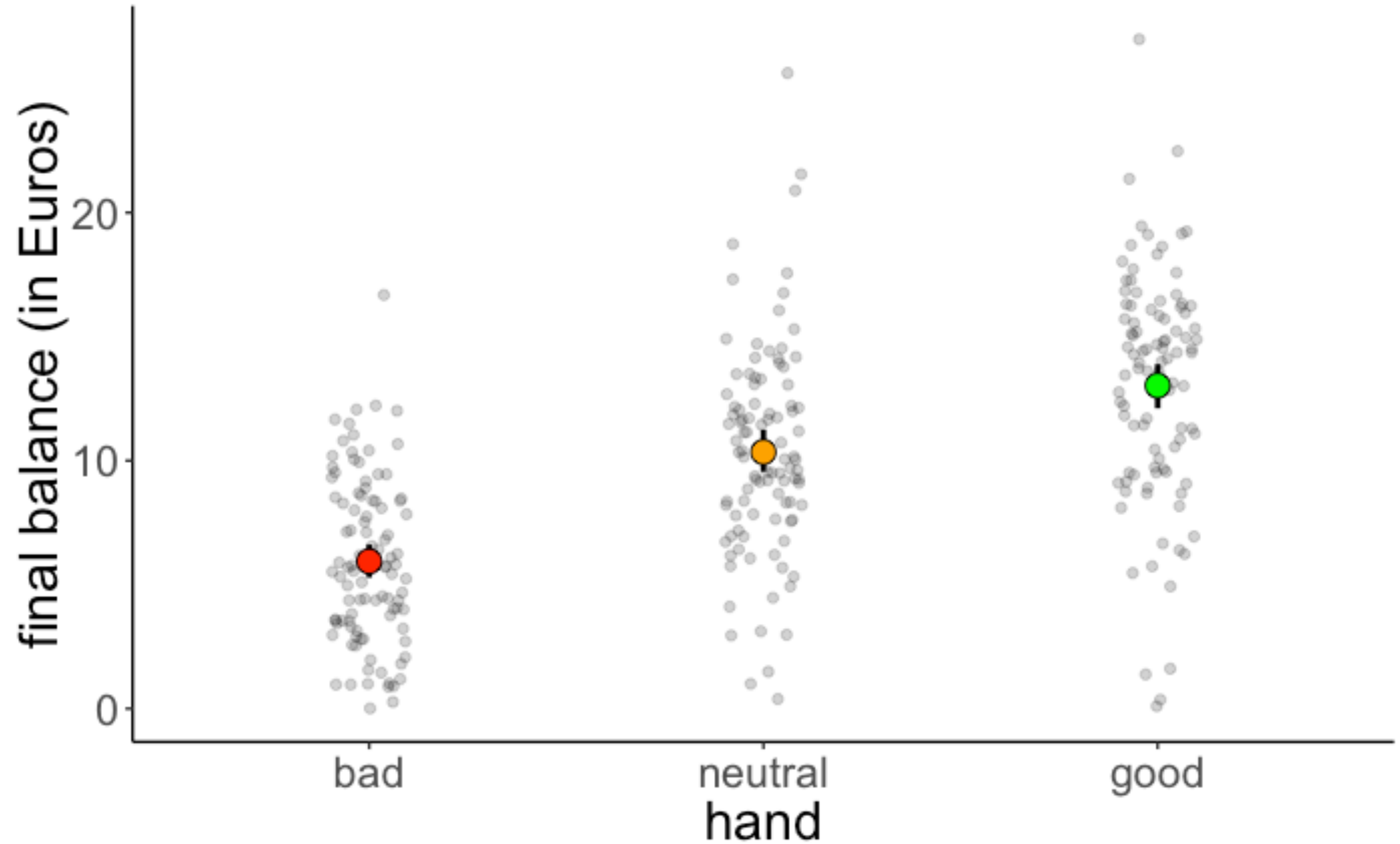
```
df.poker %>%
  mutate(mean_grand = mean(balance)) %>%
  group_by(hand) %>%
  mutate(mean_group = mean(balance)) %>%
```

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

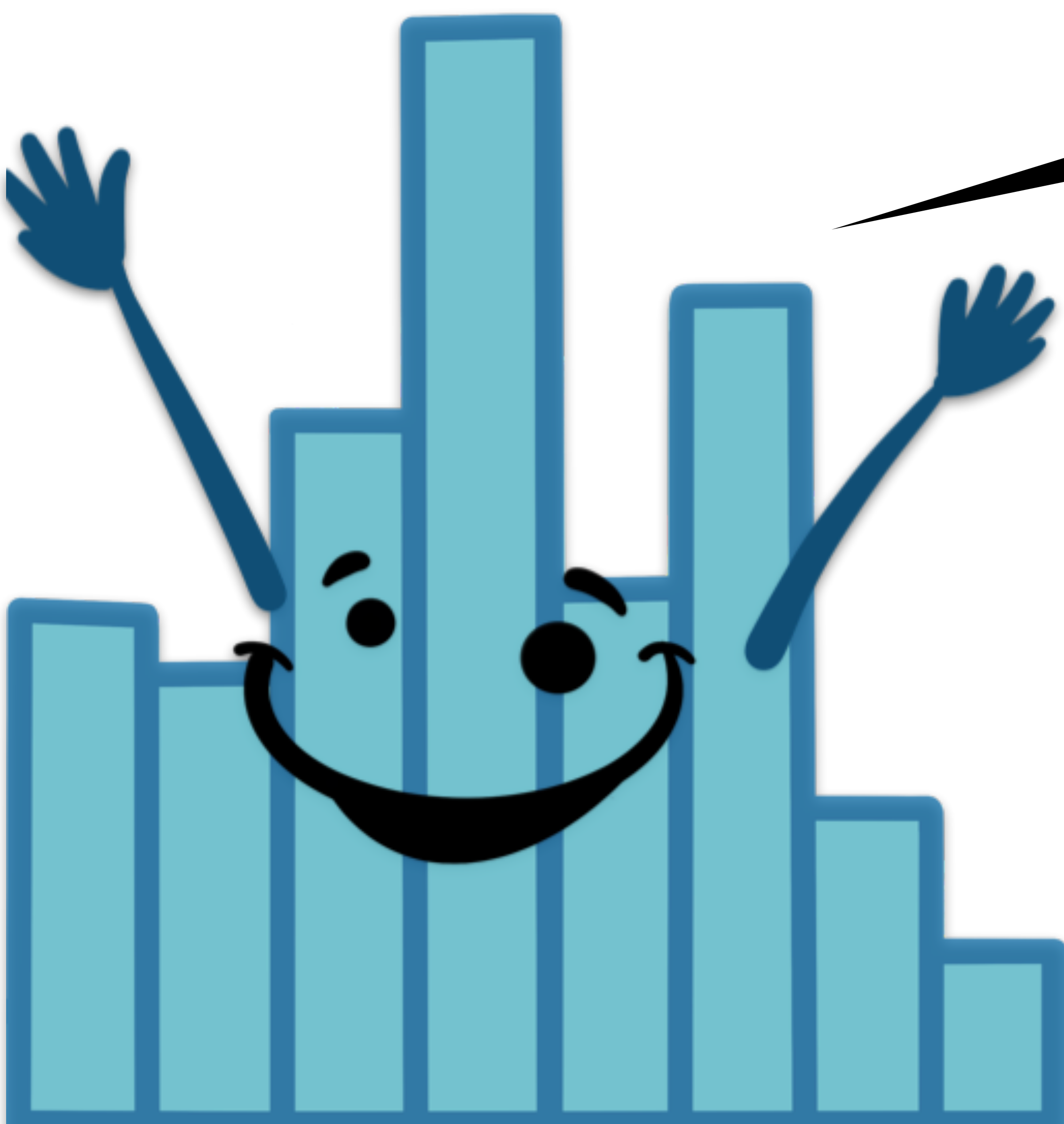


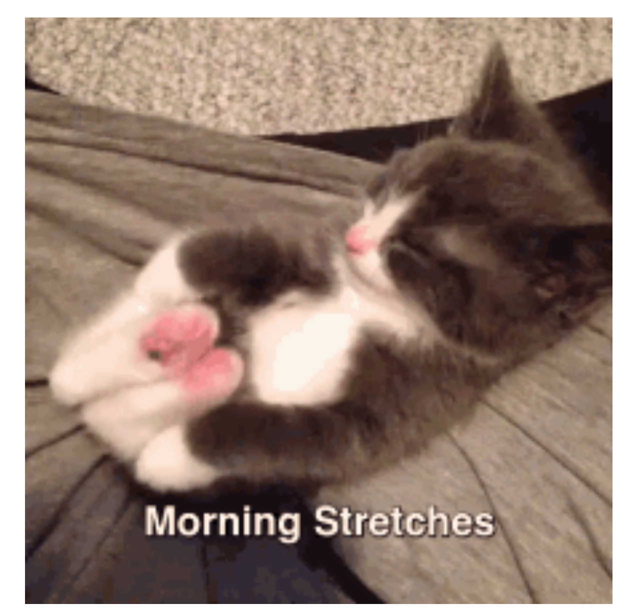
The final balance differed significantly as a function of the quality of a player's hand (i.e. whether the hand was bad, neutral, or good), $F(2, 297) = 75.703$, $p < .001$.

We're listening to
"Atlas Novus" by
"Scale the Summit"
submitted by Ben

02:00

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?

```r
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

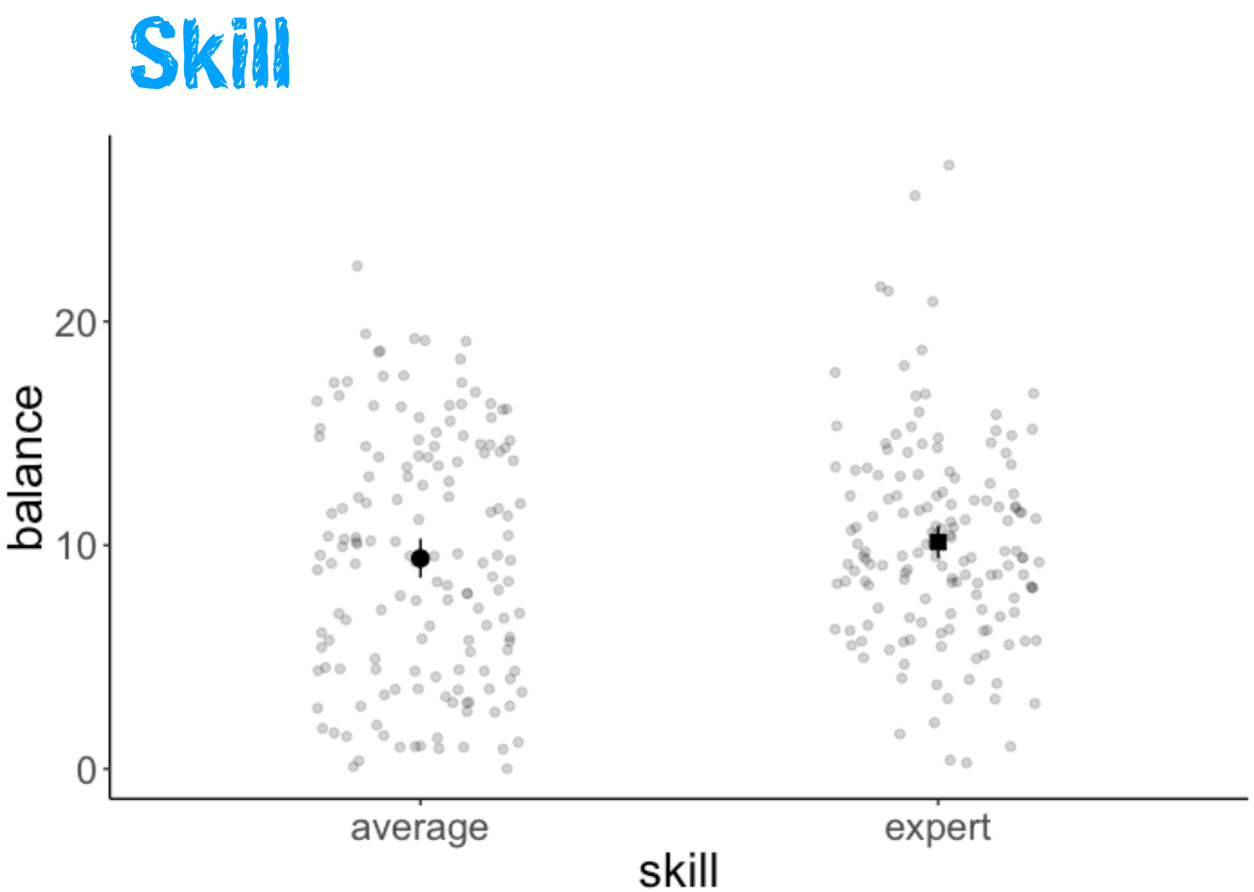
Why not just fit separate models?

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

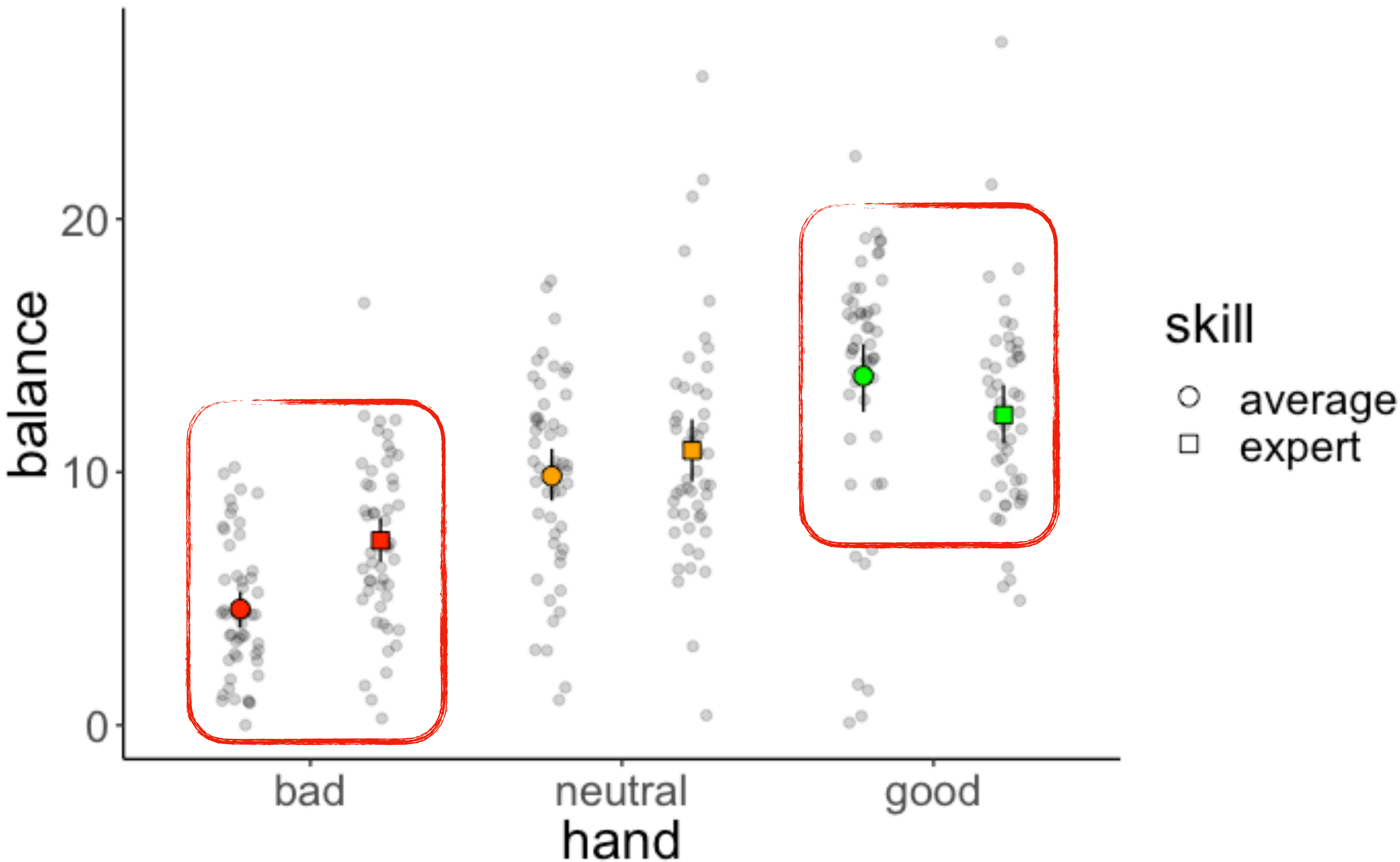
Interested in interactions!

Does the effect of one variable depend on the other?

Visualize the data



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

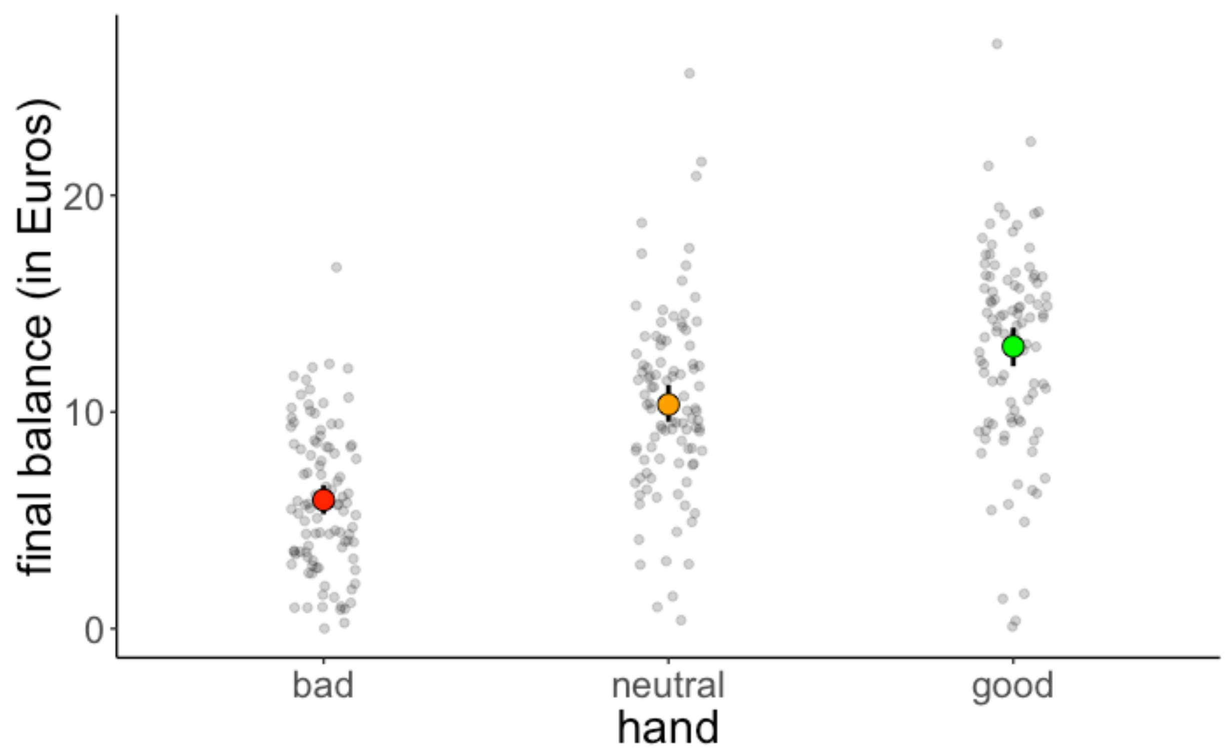
$$\widehat{\text{balance}}_i = b_0 + b_2 \cdot \text{hand\_good}_i + b_3 \cdot \text{skill\_expert}_i + b_5 \cdot \text{hand\_good:skill\_expert}_i = 12.26$$

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



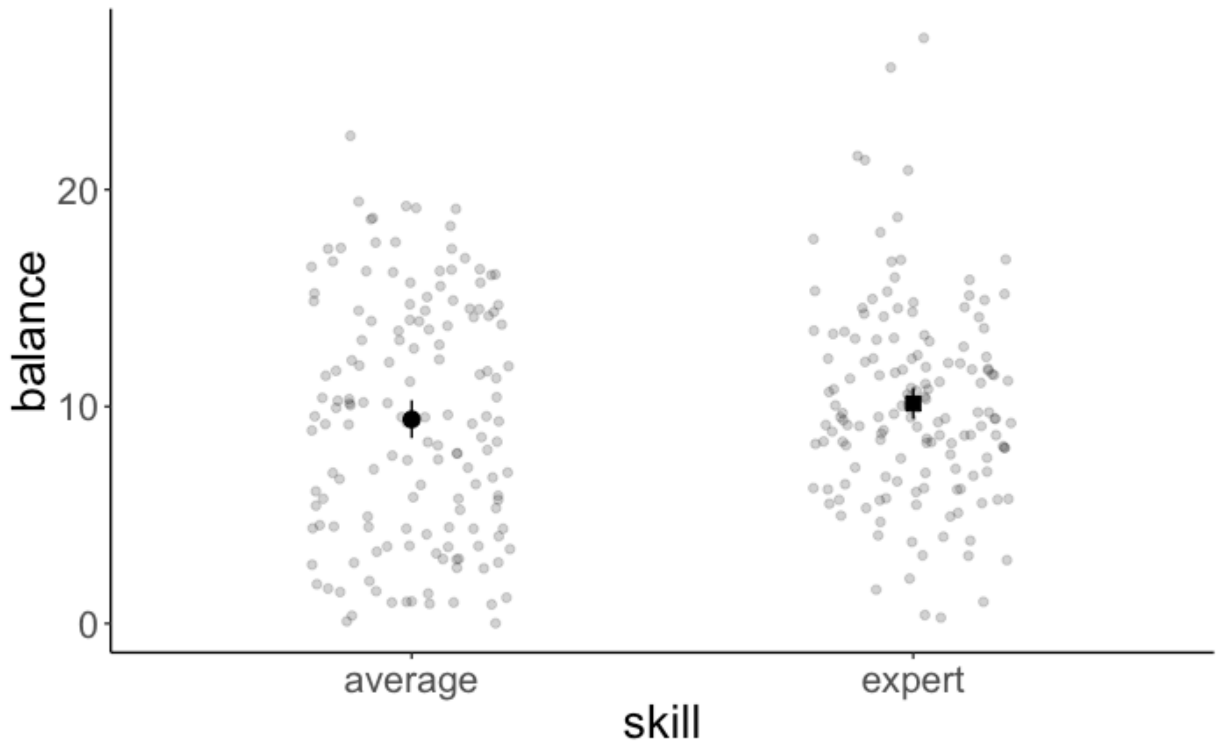
Analysis of variance

```
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

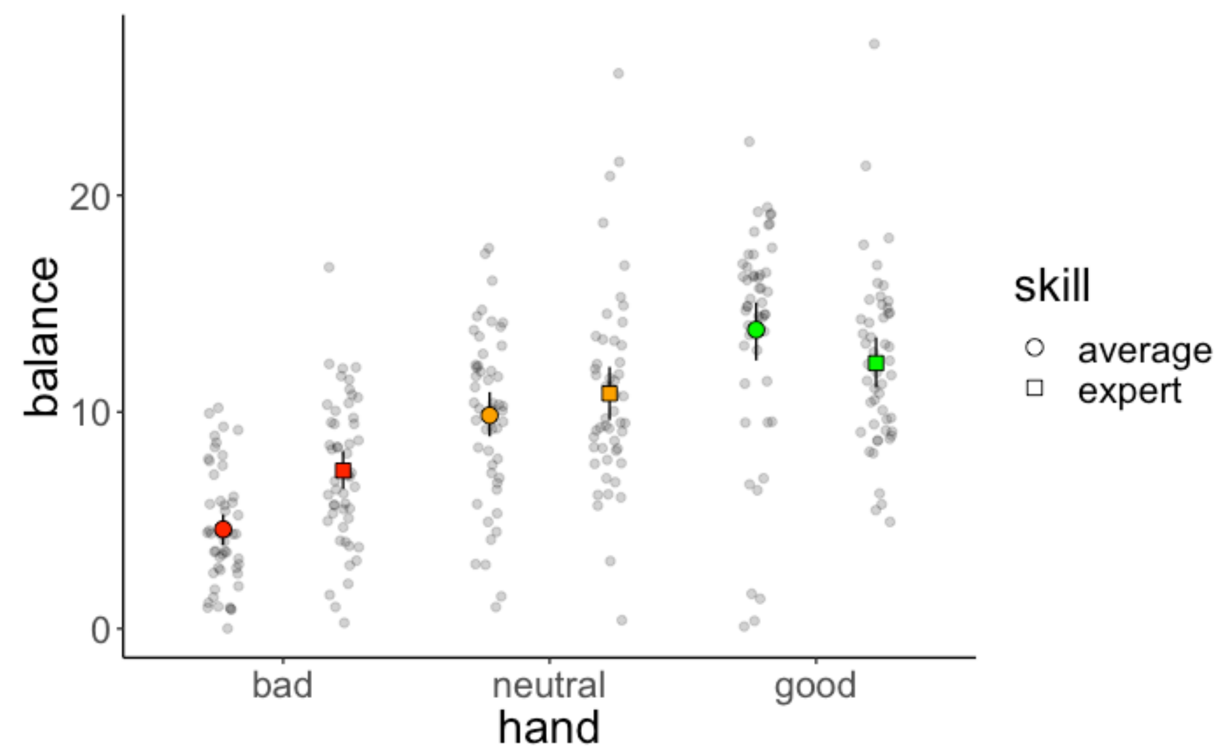
```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
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

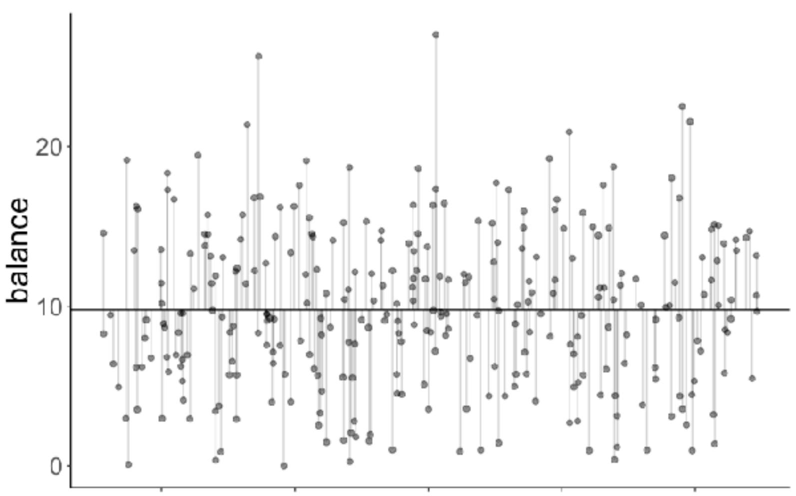
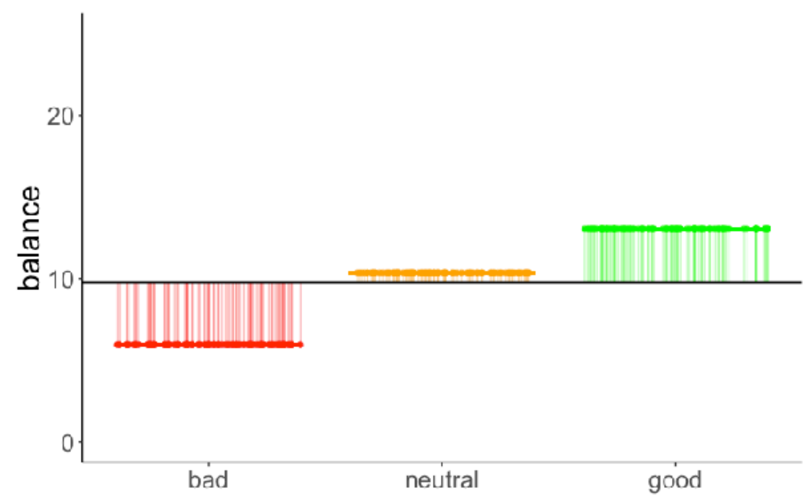
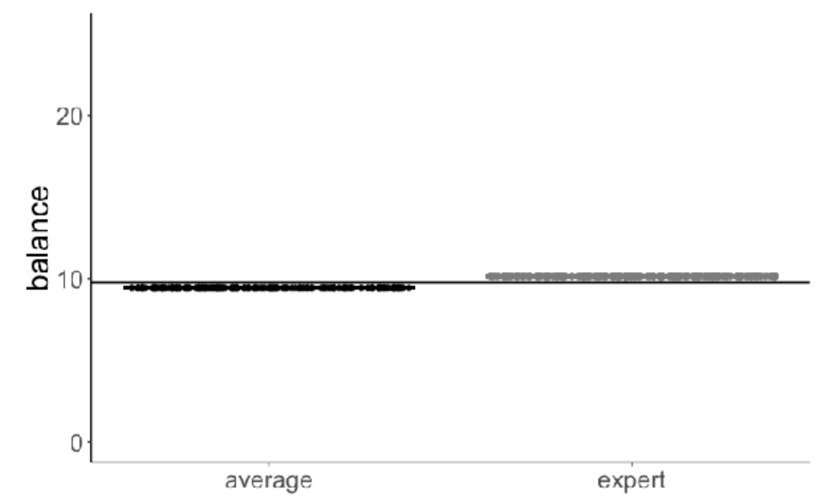
```
lm(formula = balance ~ hand + skill, data = df.poker) %>%
  anova()
```

```
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?

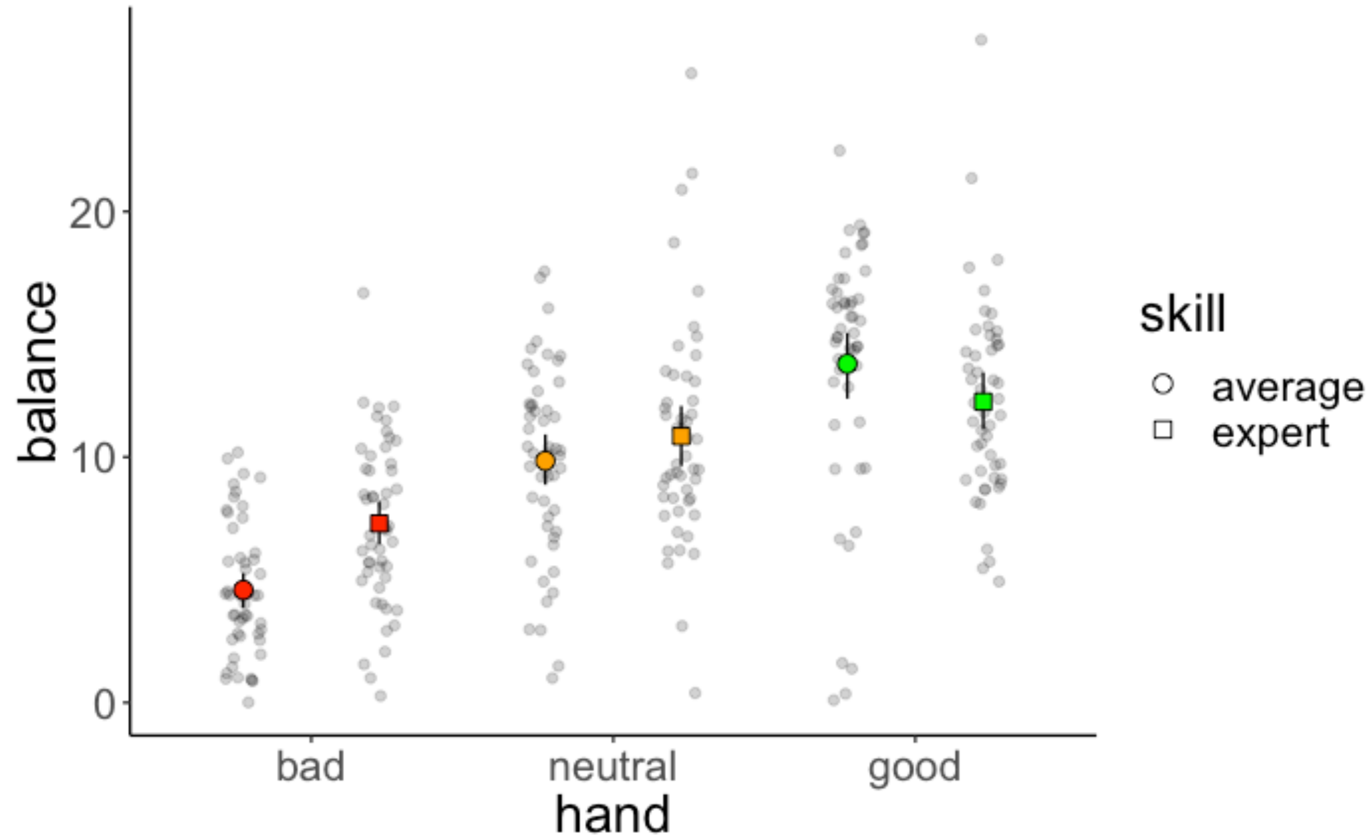
Two-way ANOVA

Variance decomposition

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

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

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

Parameter interpretation

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

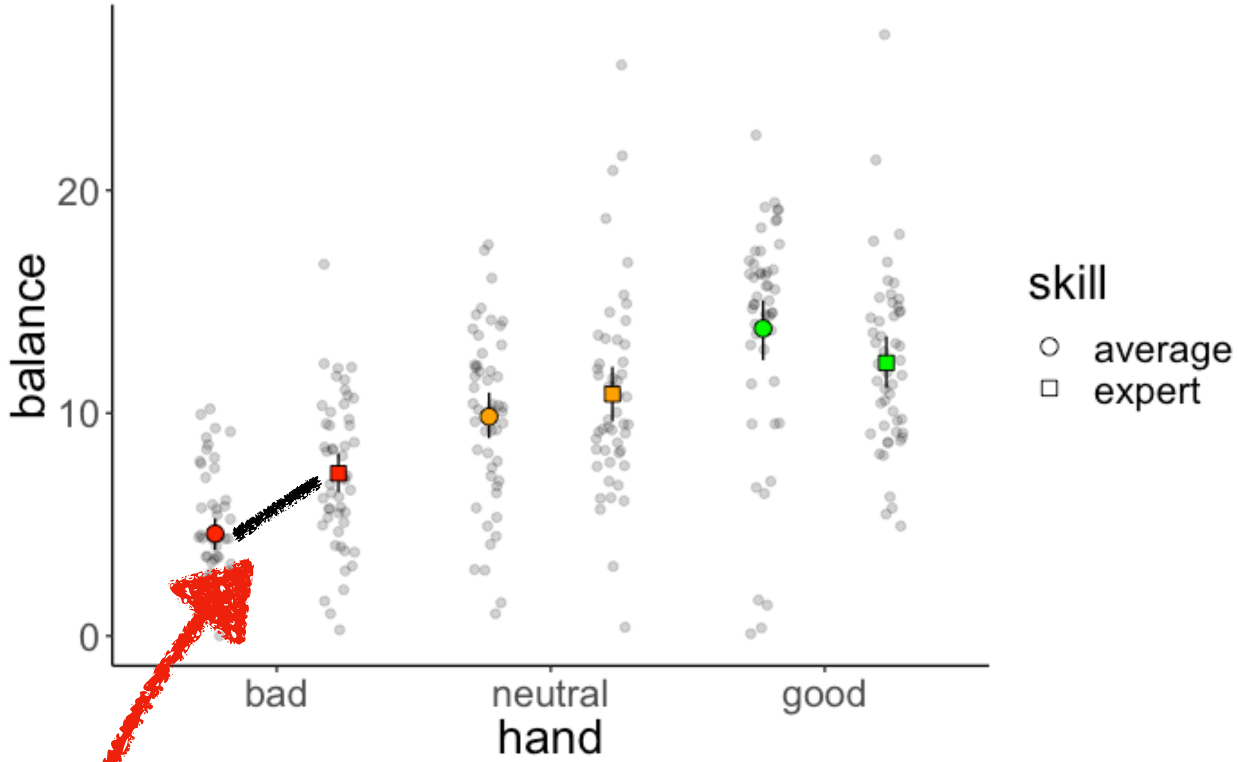
```
Call:
lm(formula = balance ~ hand * skill, data = df.poker)

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 ***
---
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~~

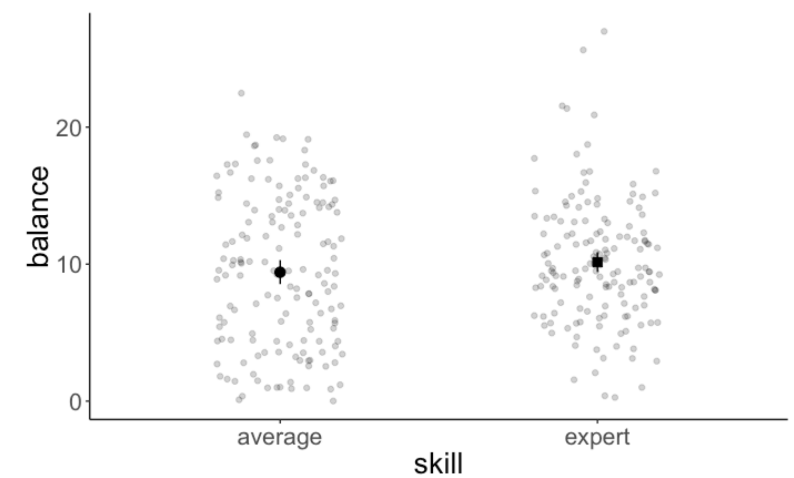
Parameter interpretation

```r
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
```

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
```

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

```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:

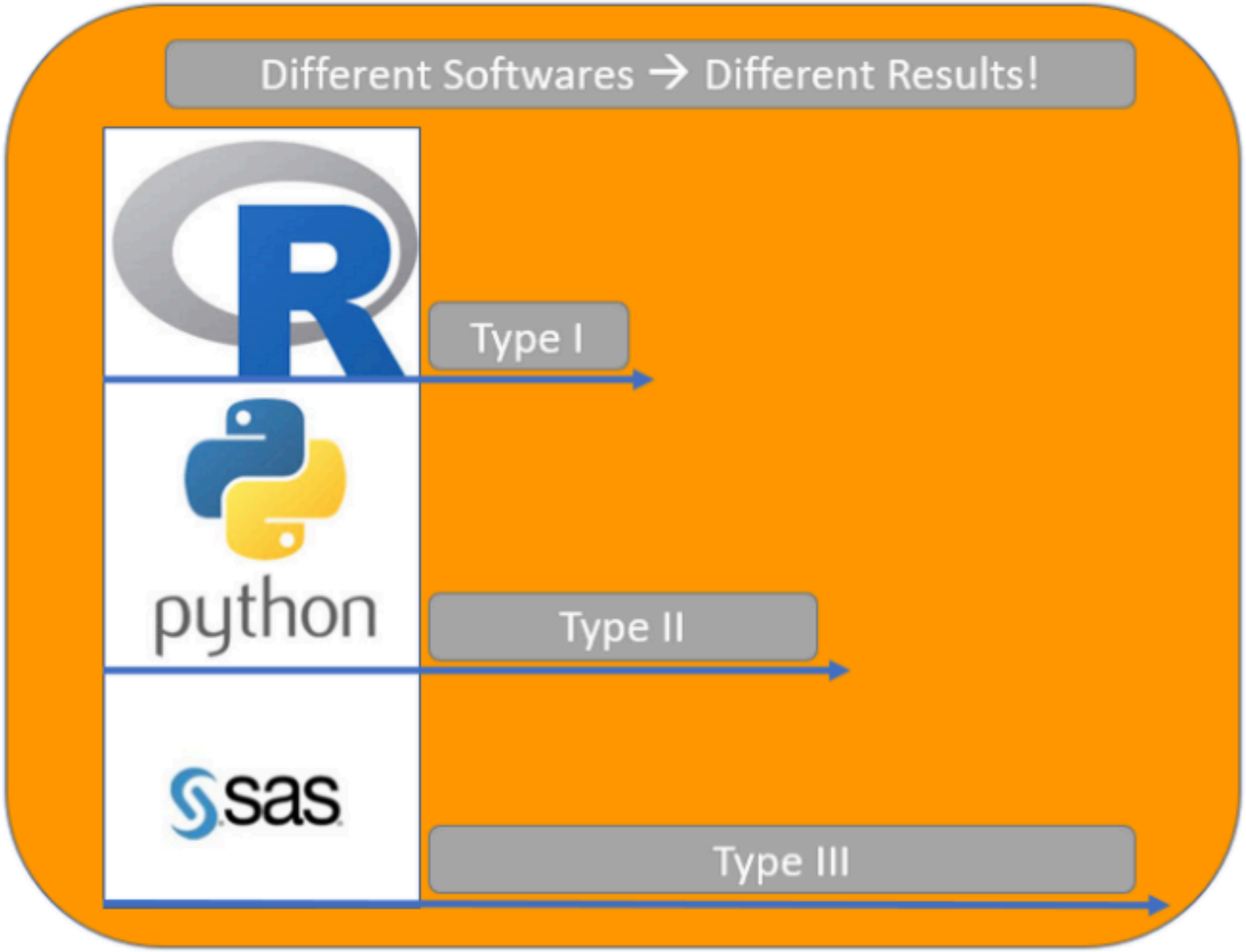
- Like Type II, the Type III Sums of Squares are not sequential, so the order of specification does not matter.
- Unlike Type II, the Type III Sums of Squares do specify an interaction effect.

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

very important!!

```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. 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

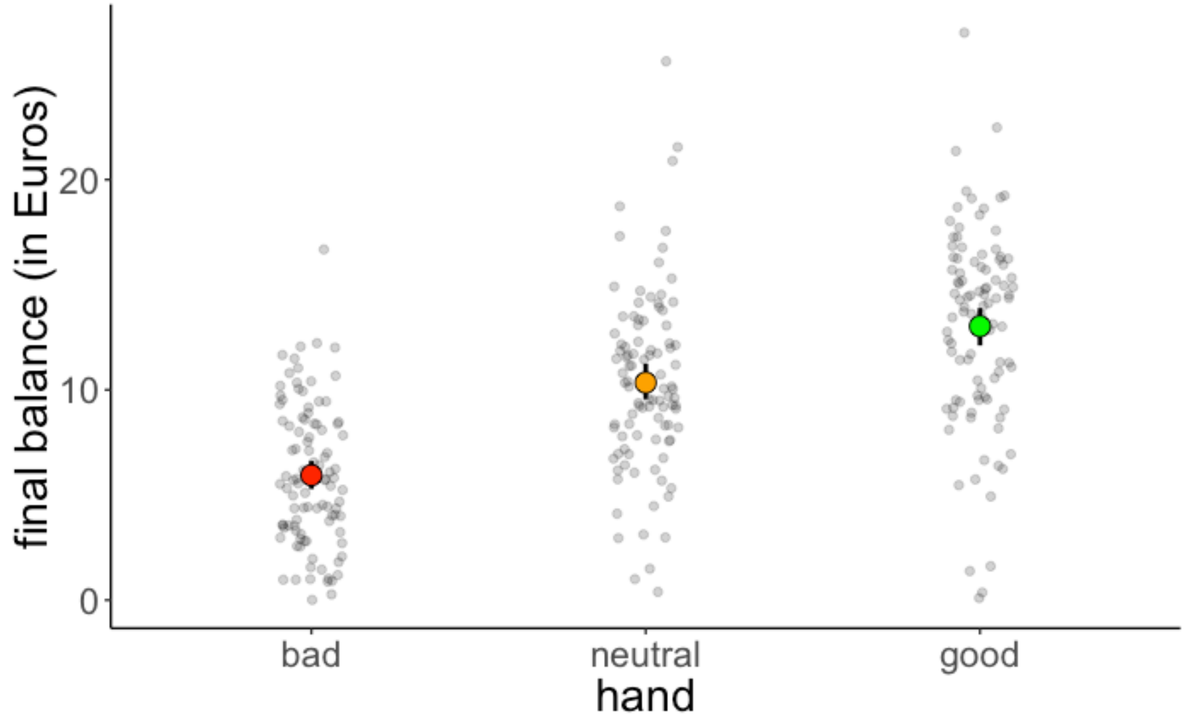
- **Dummy coding:**
  - the reference category is coded as 0
  - represented by the intercept
  - all other categories are compared to this reference category

this is what we want to do in ANOVAs

- **Effect coding:**
  - the intercept is the grand mean
  - all other categories are compared to the grand mean

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
```

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

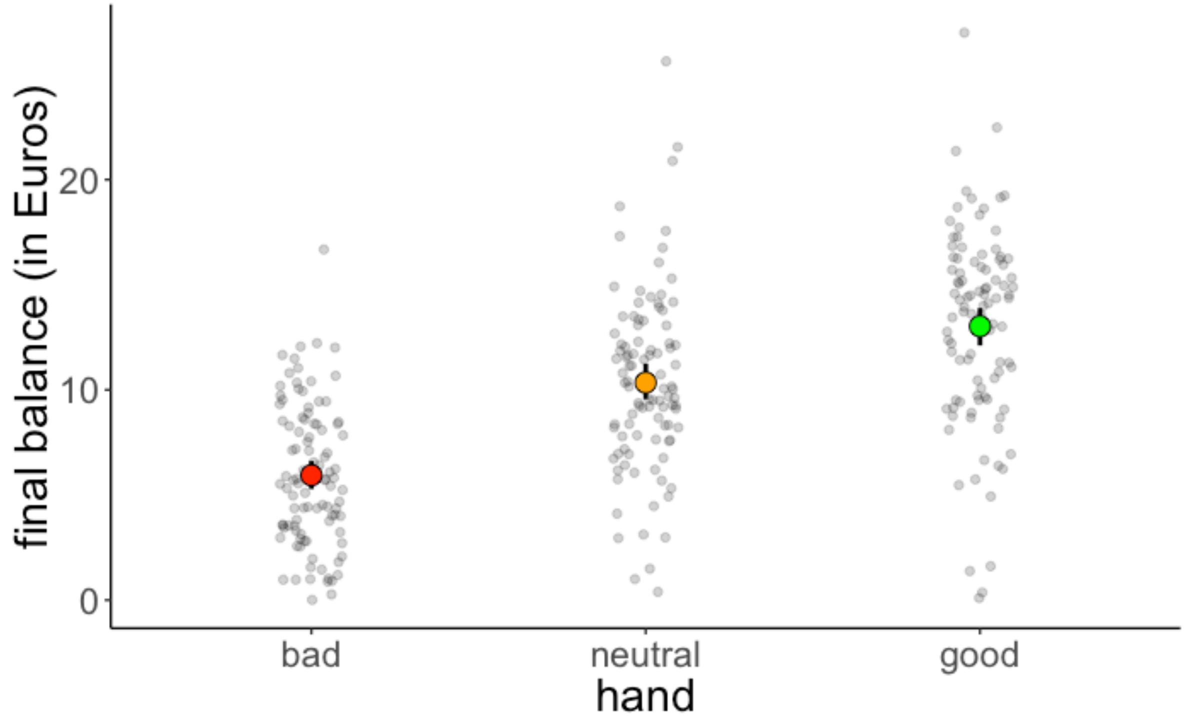
```
lm(formula = balance ~ hand * skill, data = df.poker)
```

```
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

```
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
```

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

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

Effect coding

```r
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

```r
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
```

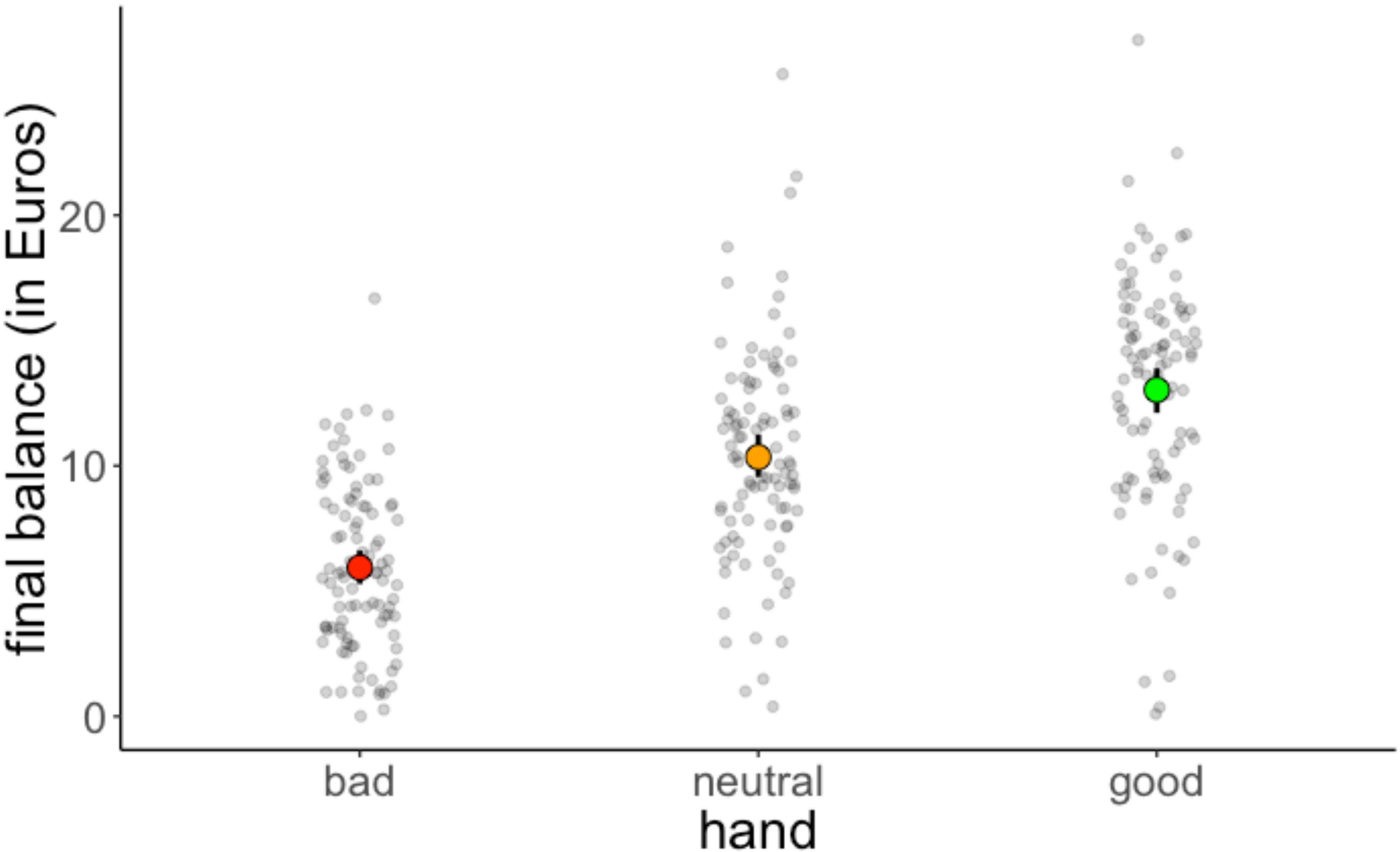
reference

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

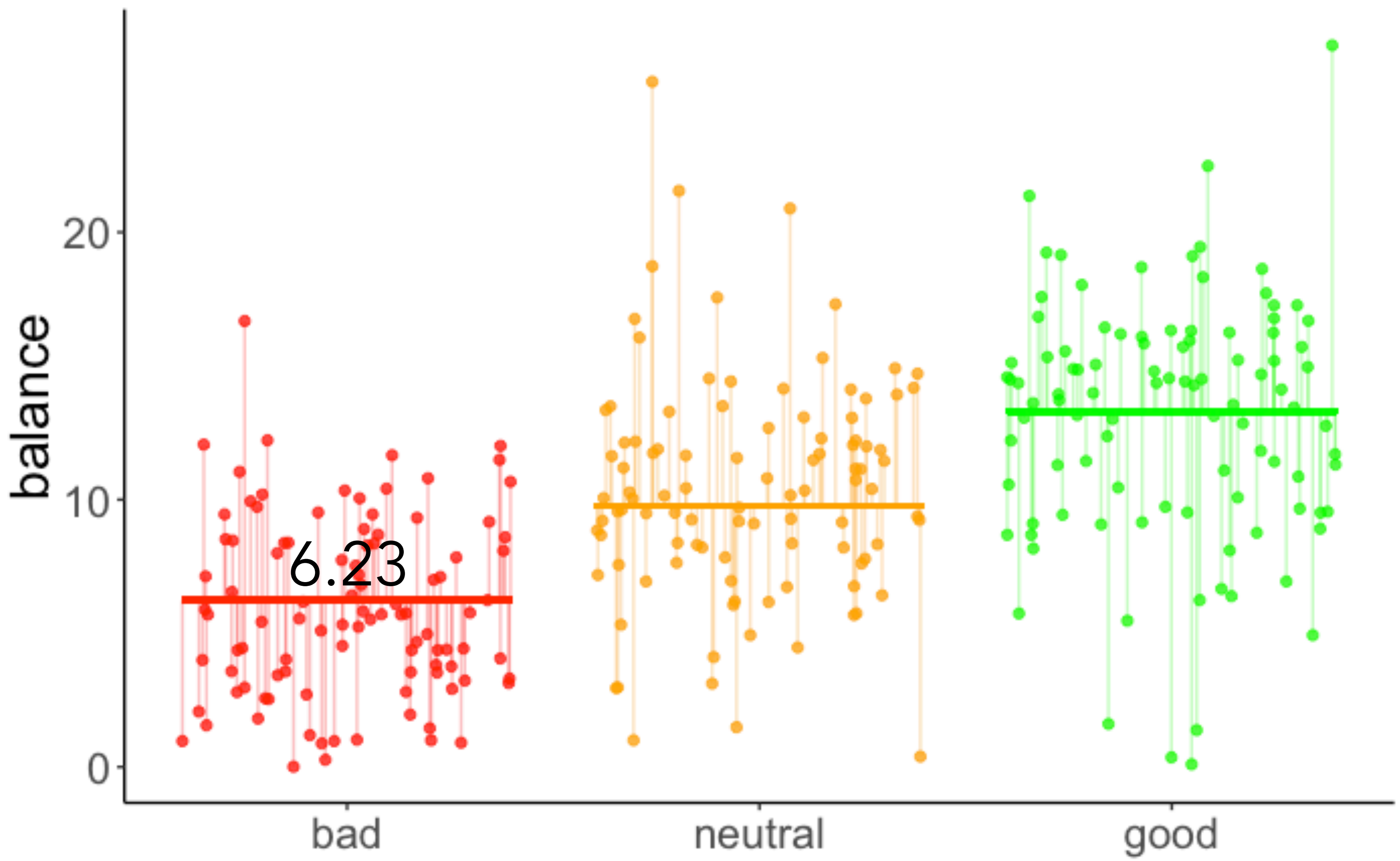
significant contrast

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

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



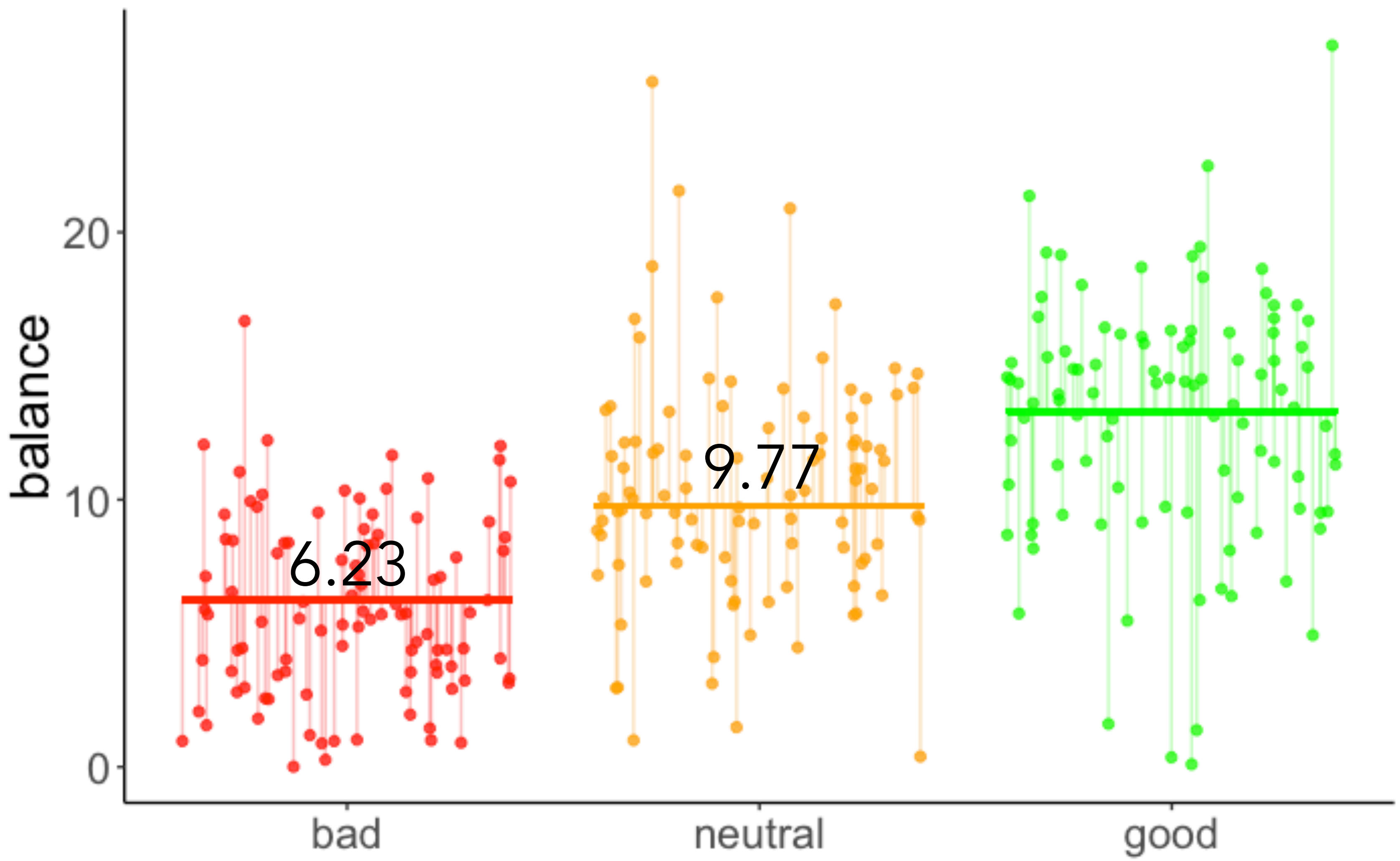
if contrast == -1

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

Contrasts

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

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



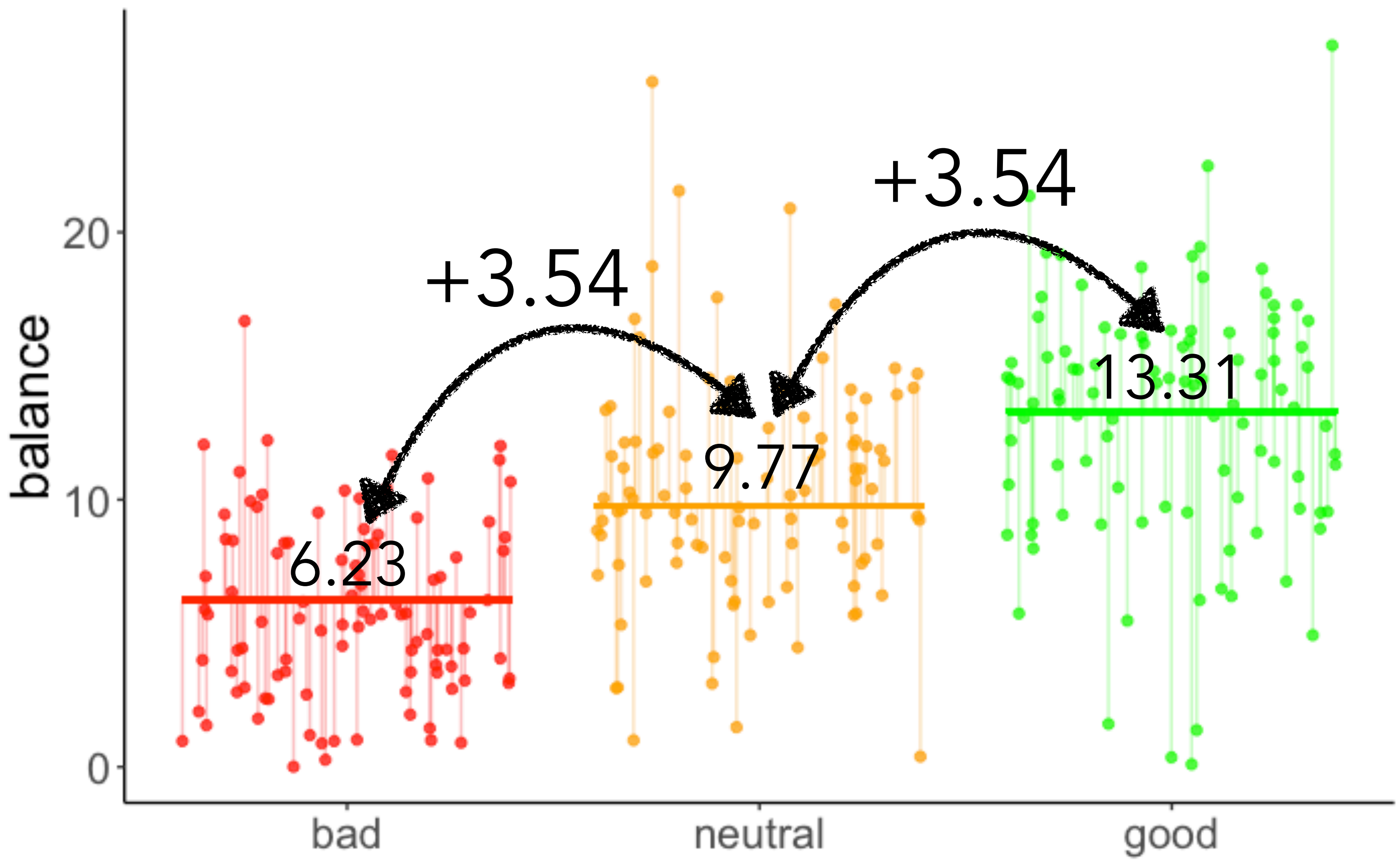
if contrast == 0

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

Contrasts

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

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

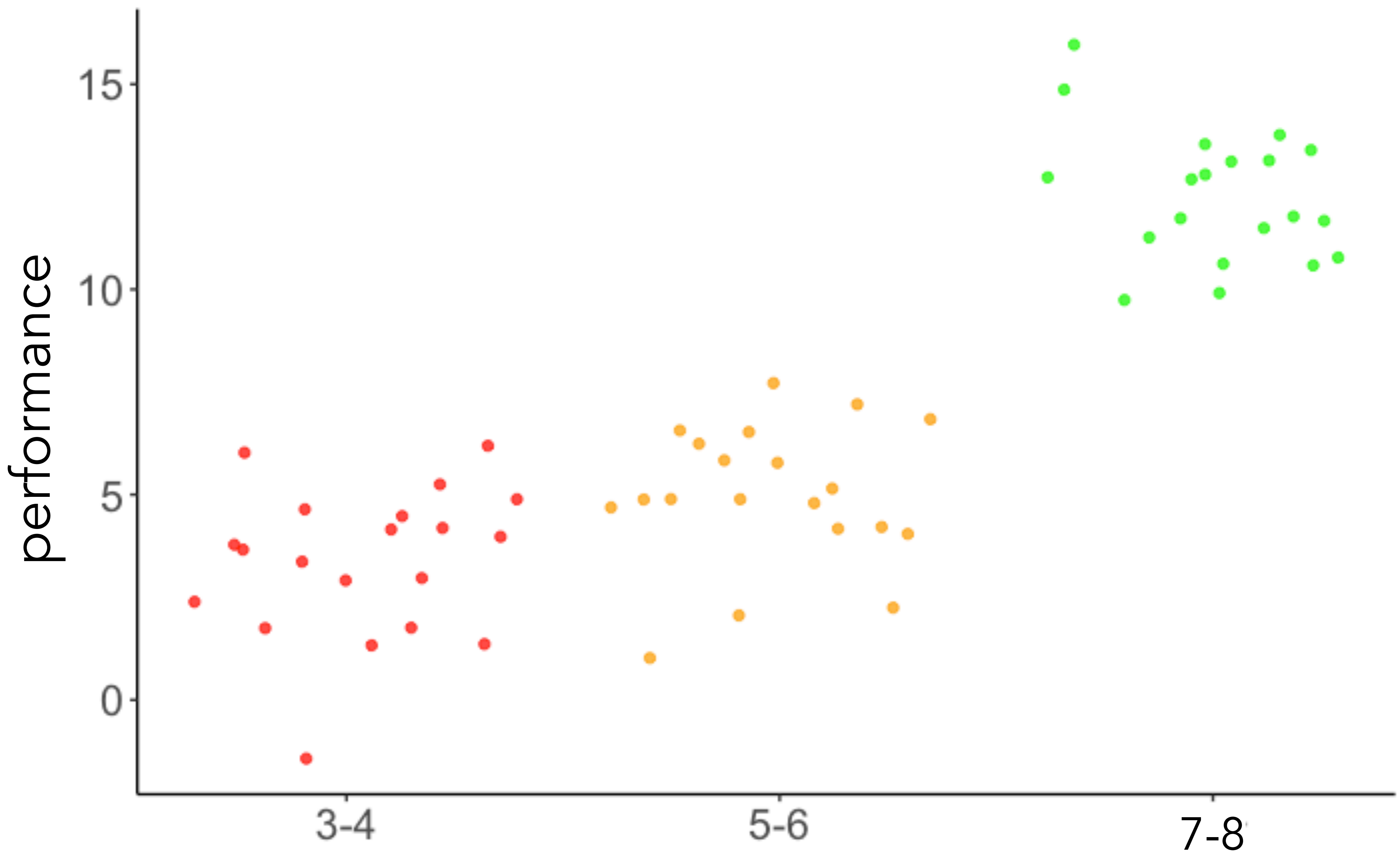


if contrast == 1

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

Contrasts

Does performance increase with age?

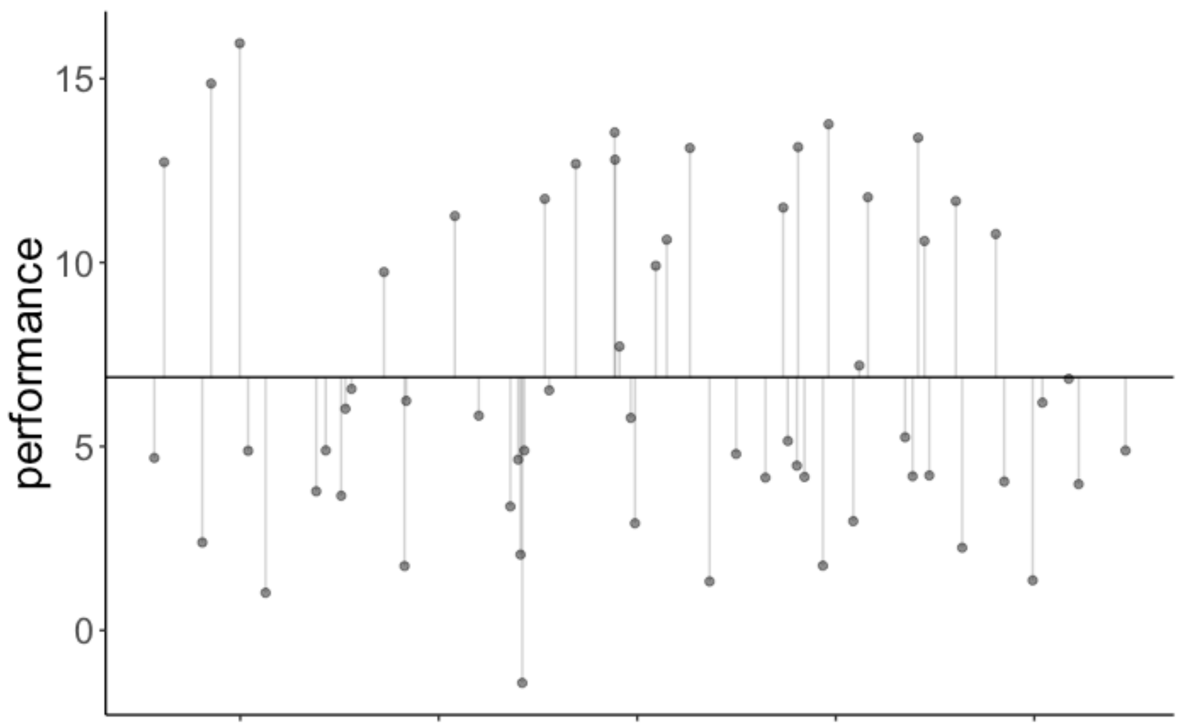


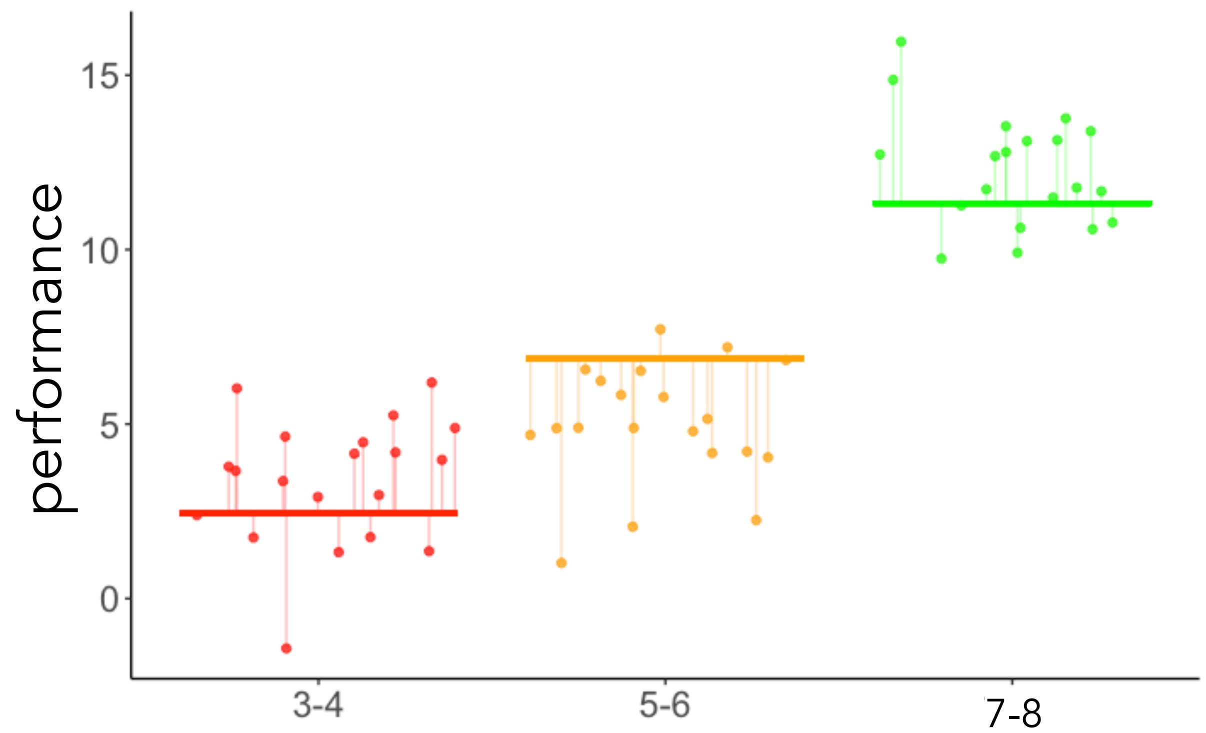
Data from a hypothetical developmental study

Contrasts

Does performance increase with age?

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



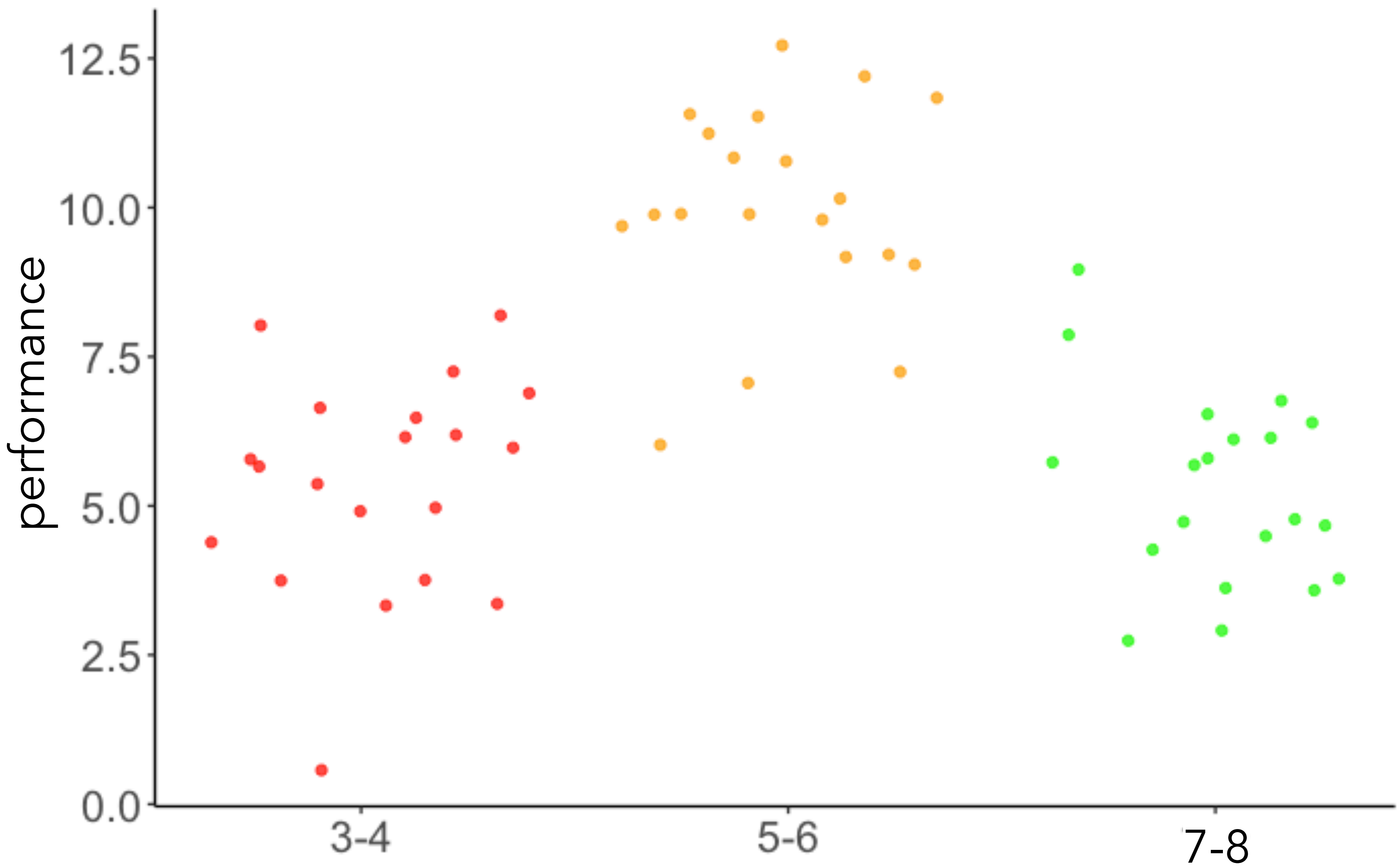


Model comparison

$p < .001$

Contrasts

Does performance increase with age?



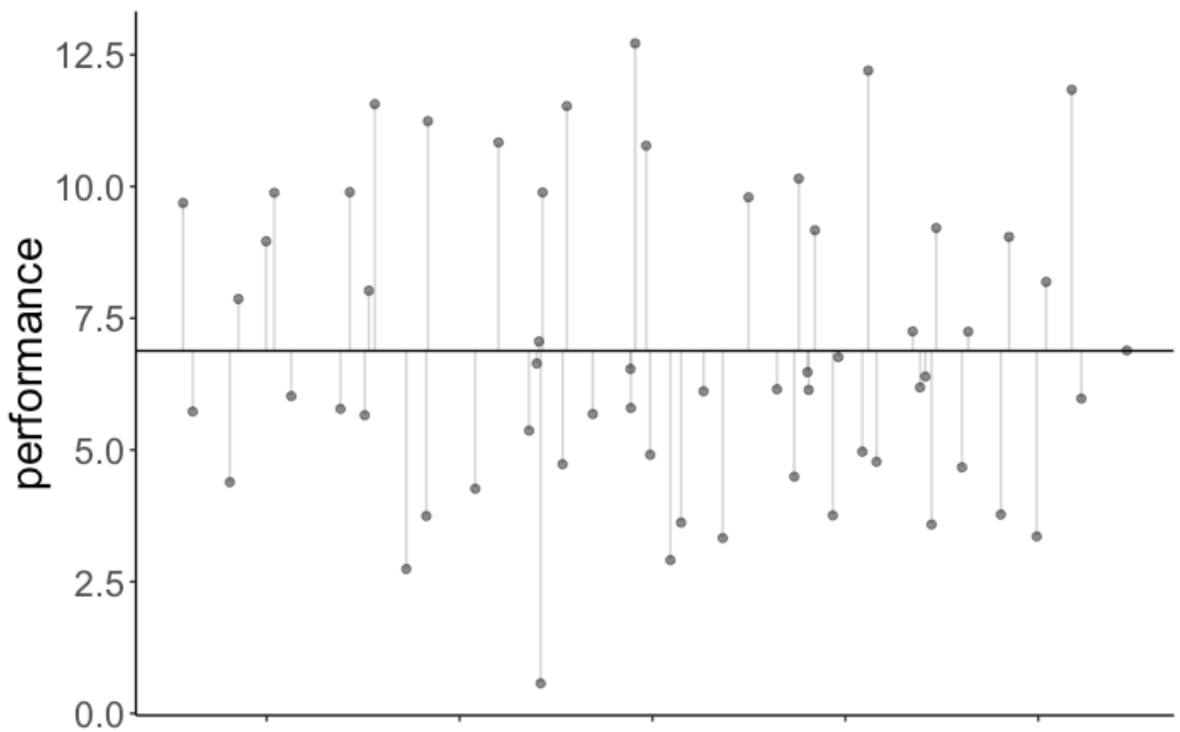
Data from another hypothetical developmental study

Contrasts

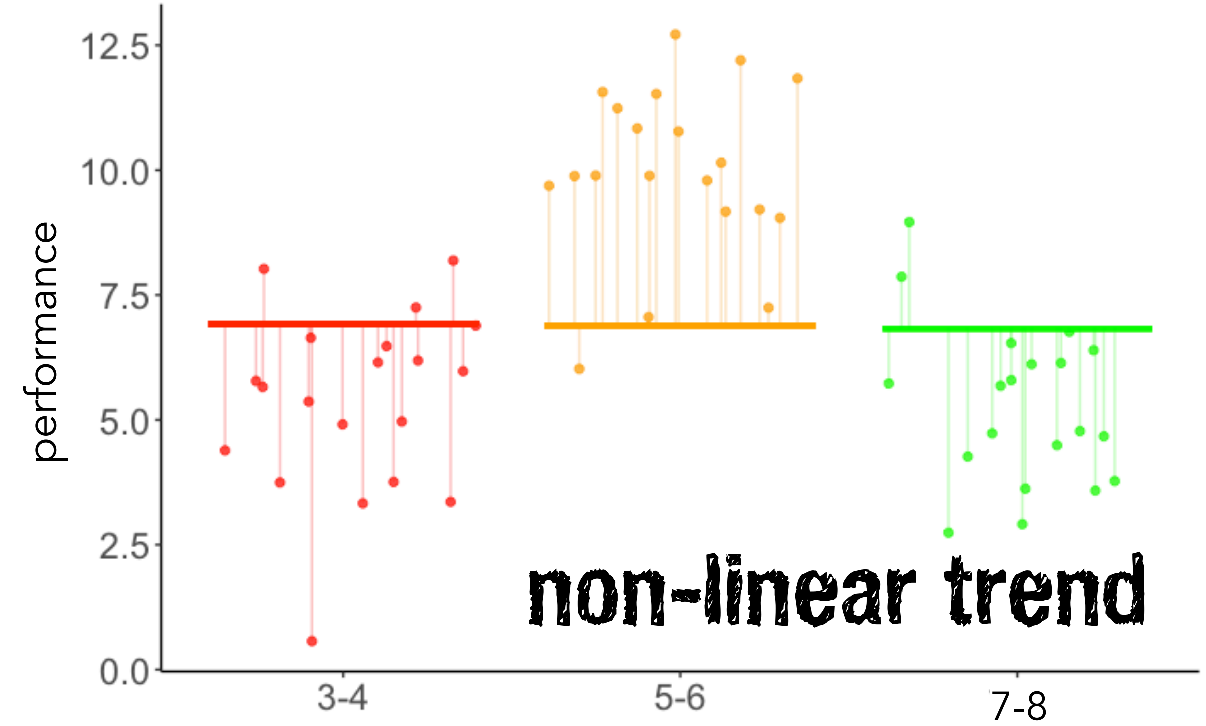
Does performance increase with age?

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

Compact model



Augmented model



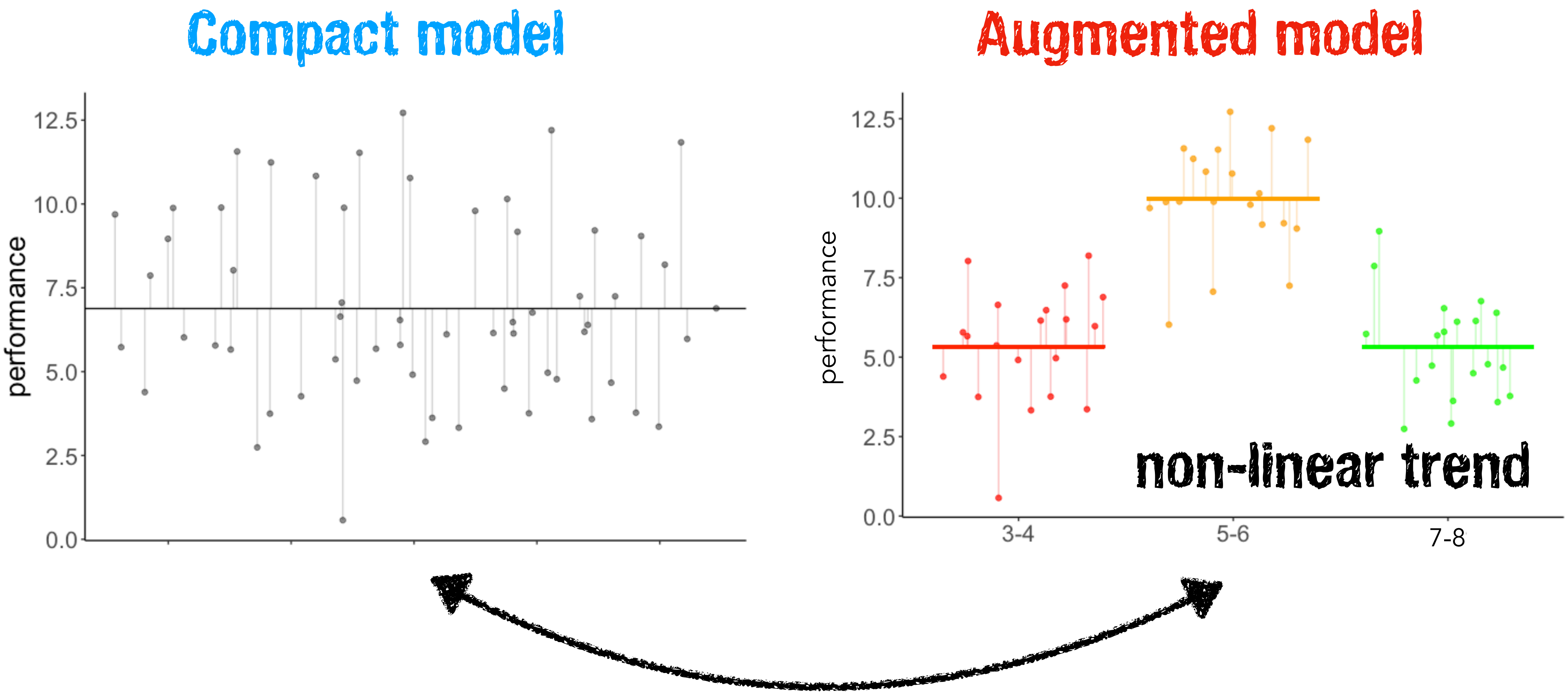
Model comparison

p = .8508

Contrasts

Does performance increase with age?

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



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.

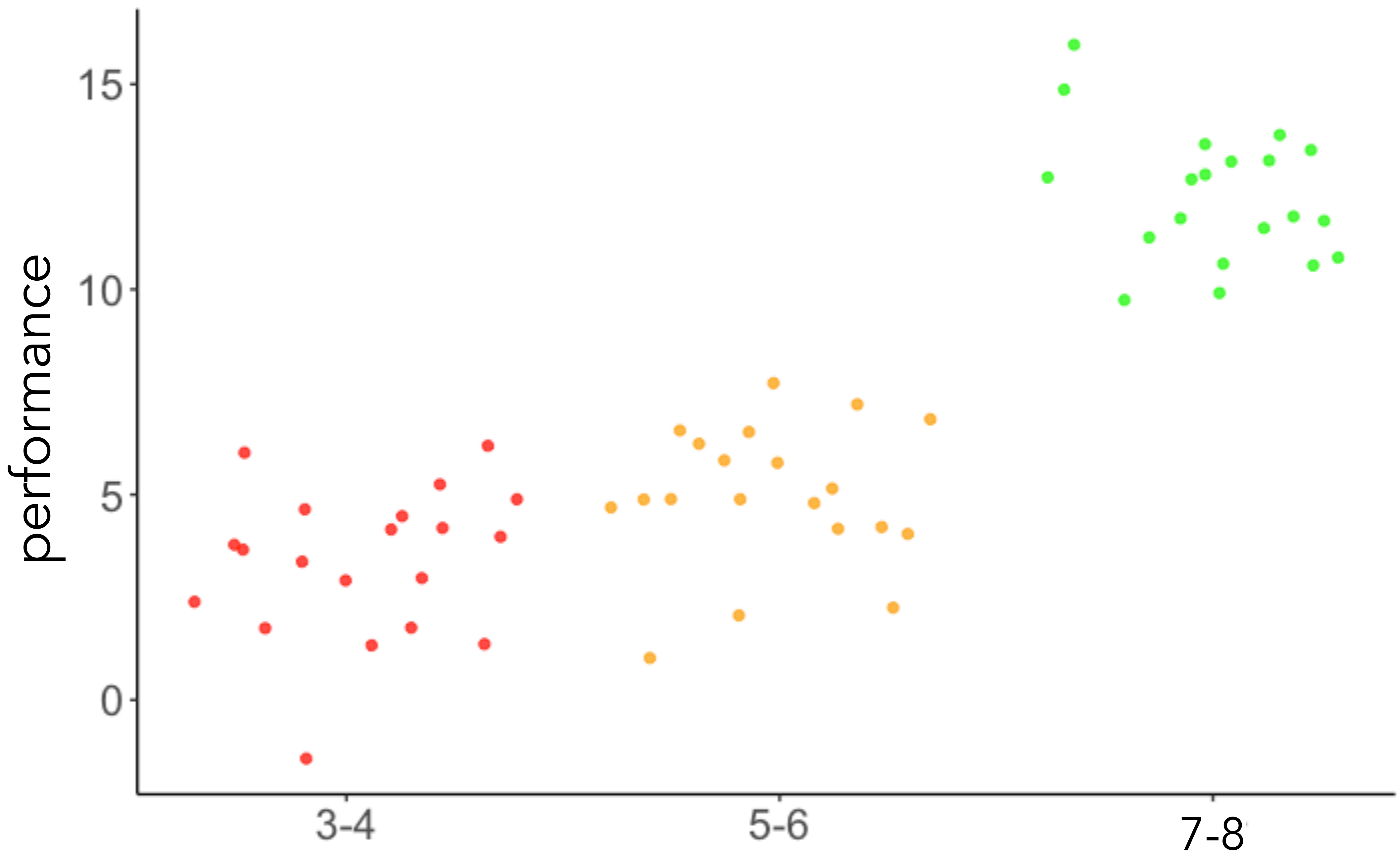
`emmeans` works with `lm`, `glm`, and the Bayesian friends in `brms` and `rstanarm`, so the process is applicable no matter the tool.

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

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

Contrasts

Does performance increase with age?



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

```
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

```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"

# 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))
```

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

```r
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")
```

hypothesis tests are the same!

```
[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
```

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

```r
# fit the model
fit = lm(formula = balance ~ hand + skill,
         data = df.poker)

# post hoc tests
fit %>%
  emmeans(pairwise ~ hand + skill,
          adjust = "bonferroni") %>%
  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

that's a lot of tests!

... not

all pairwise tests between groups

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?

Summary

- Quick recap
- `lm()` output
- Analysis of Variance (ANOVA)
  - multiple categorical predictors (N-way ANOVA)
    - interpreting parameters
  - 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?

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?

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

Thank you!