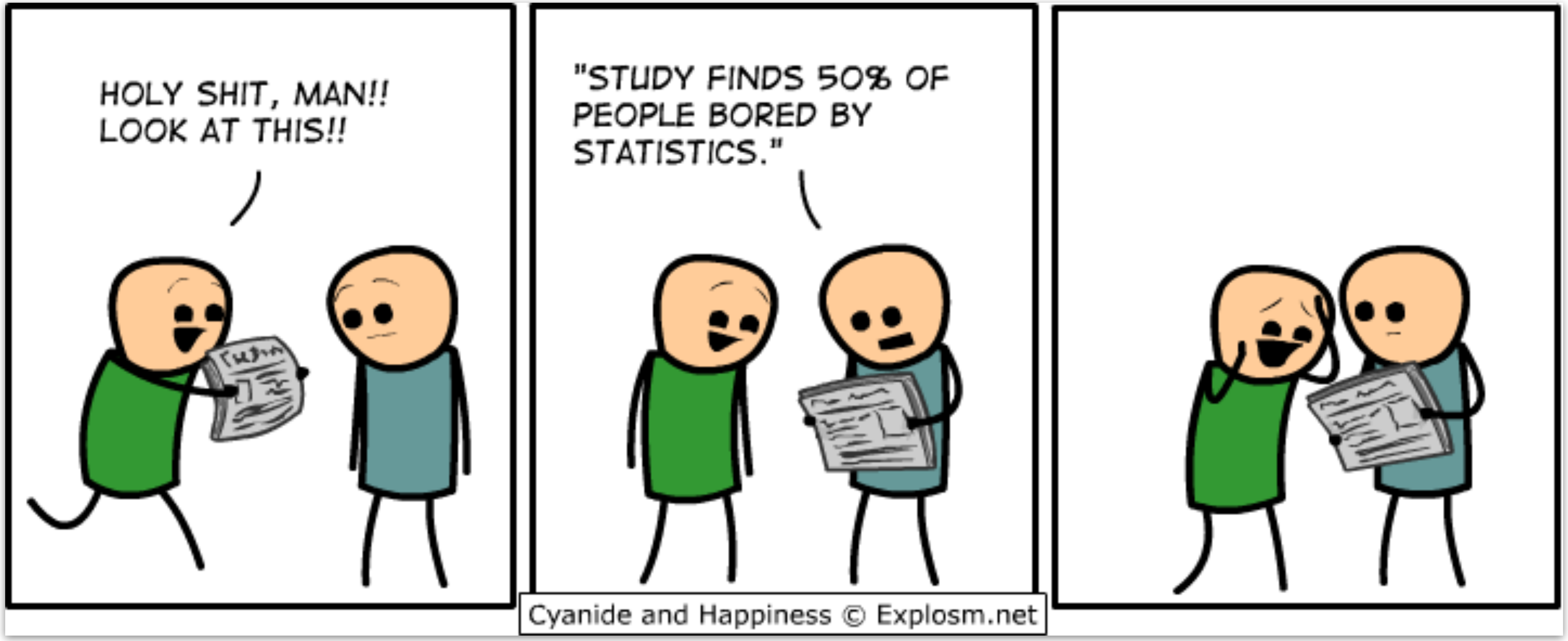
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

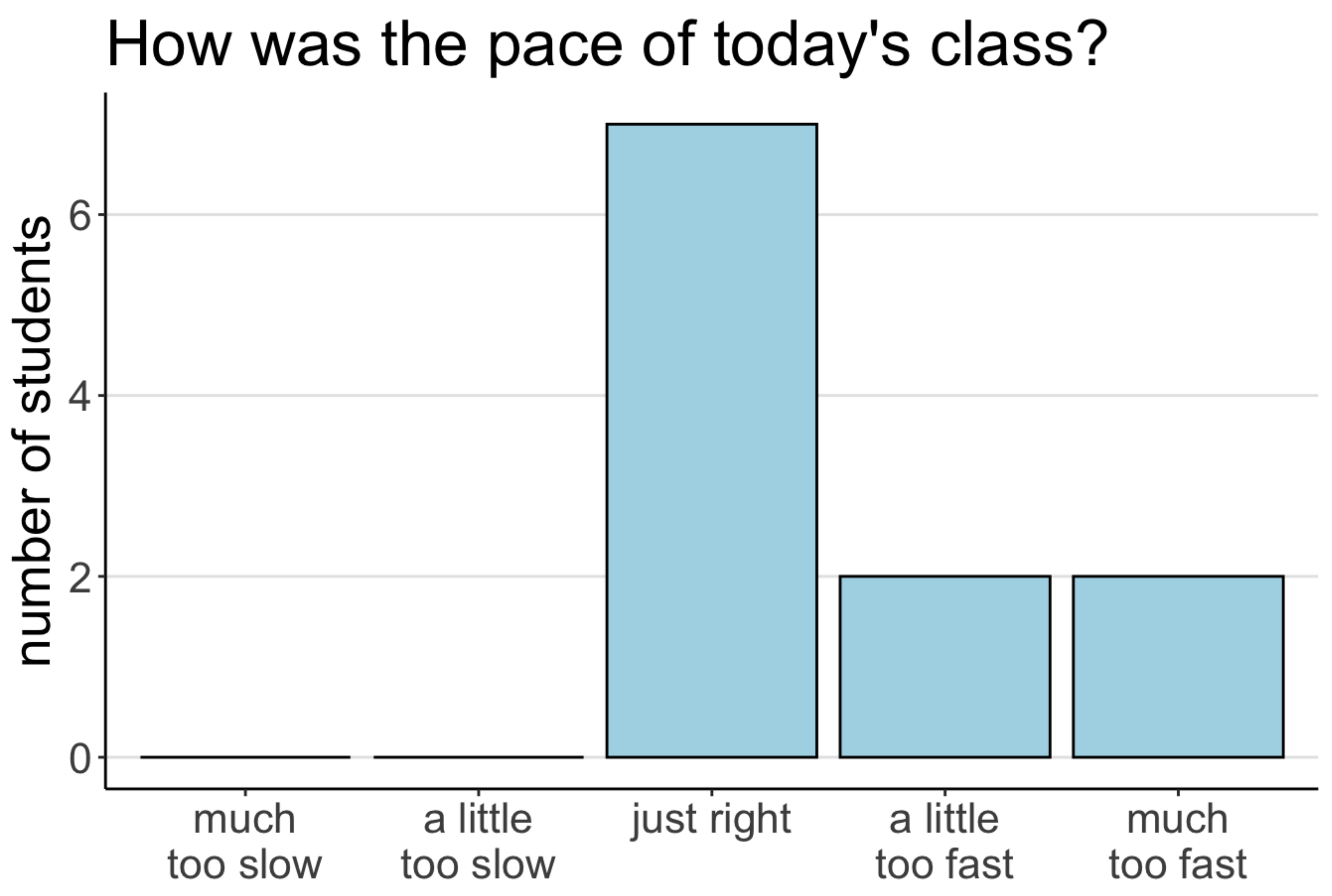
Linear model 4

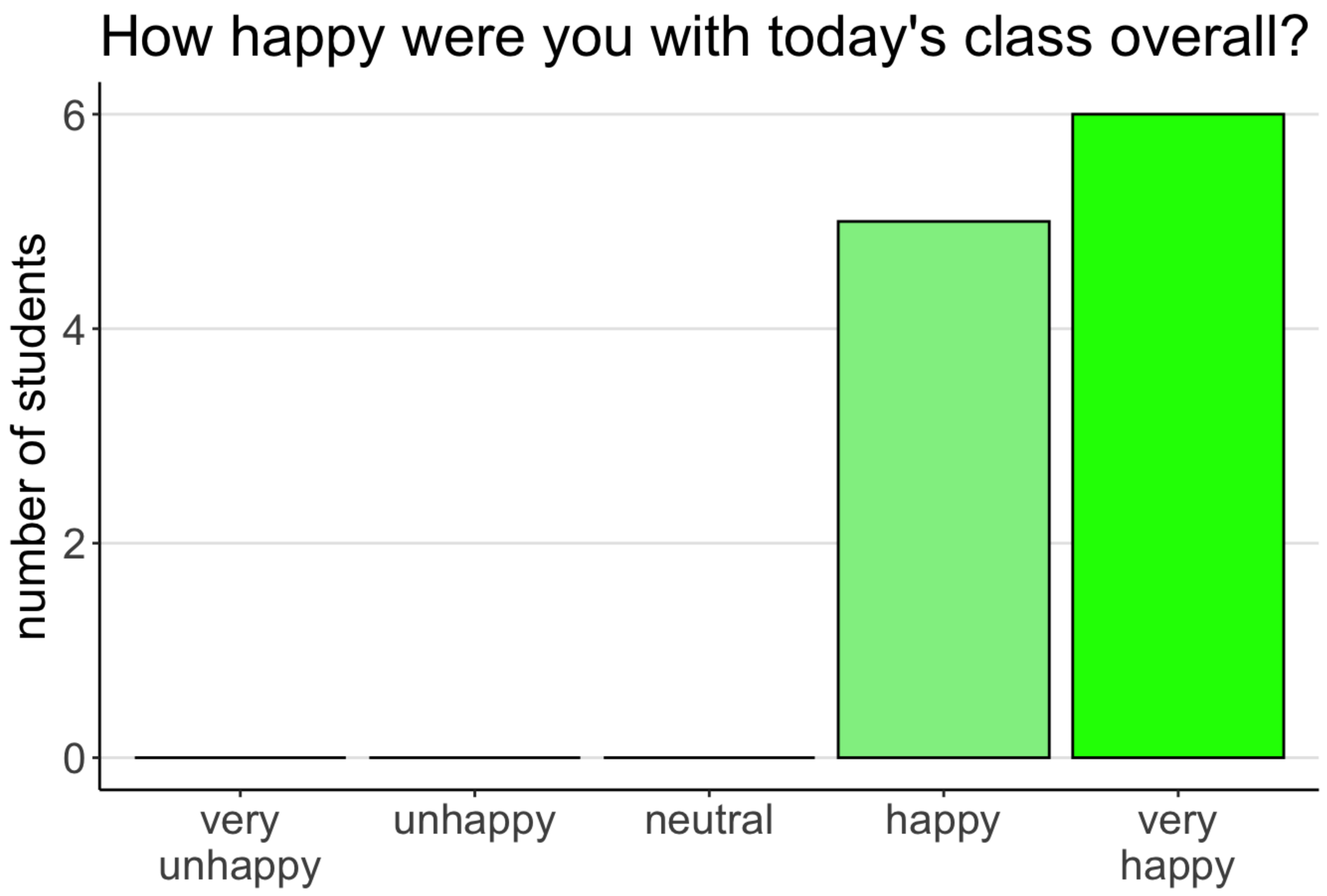


02/05/2025

Feedback

Feedback





Showing the relation between graph, formula, and R was very helpful

It's really helpful when you show the output and put a red square around the key parameters.

I enjoyed the mini history lesson about how we got from t-test to generalized SEM, etc. Stats is cool and you all make it fun to learn :)

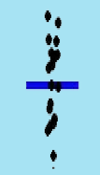
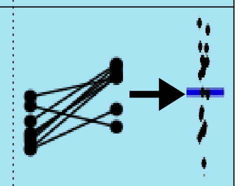
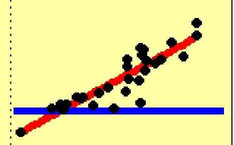
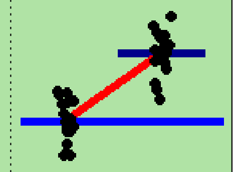
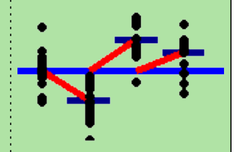
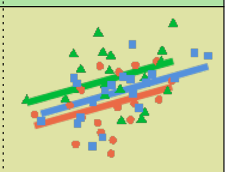
Relation between RSS, Residual Standard Error, Multiple R-Squared, Adjusted variants?

Things that came up

Common statistical tests are linear models

Last updated: 28 June, 2019.Also check out the Python version!

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise **y_2-y_1** differences. - (Same, but it predicts the *signed rank* of **y_2-y_1**.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)[A] gls(y ~ 1 + G_2, weights=…[B])[A] lm(signed_rank(y) ~ 1 + G_2)[A]	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +...)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +...+ G_N)[A] lm(rank(y) ~ 1 + G_2 + G_3 +...+ G_N)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +...+ G_N + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 +...+ G_N + S_2 + S_3 +...+ S_K + G_2*S_2 + G_3*S_3 + … + G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: **$G_{2 \text{ to } N}$** is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for **$S_{2 \text{ to } K}$** for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2 + G_3*S_3 +...+ G_N*S_K, family=...)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments:* `glm(model, family=poisson())` *As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +...+ G_N, family=...)[A]	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

[A] See the note to the two-way ANOVA for explanation of the notation.
[B] Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.

Jonas Kristoffer Lindeløv
https://lindeloev.net

Logistics

Plan for today

- Quick recap
- Interaction
- `lm()` output
- 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

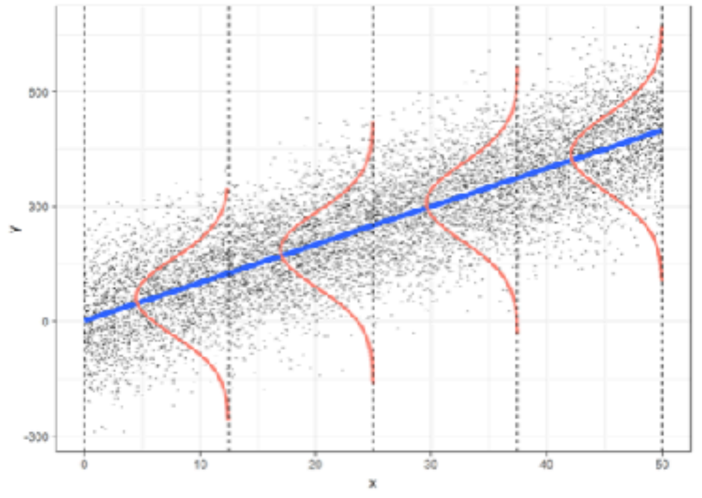
Quick recap

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

H$_0$: Radio ads and sales are not related once we control for TV ads.

H$_1$: Radio ads and sales are related even when we control for TV ads.

Model C

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

Model A

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

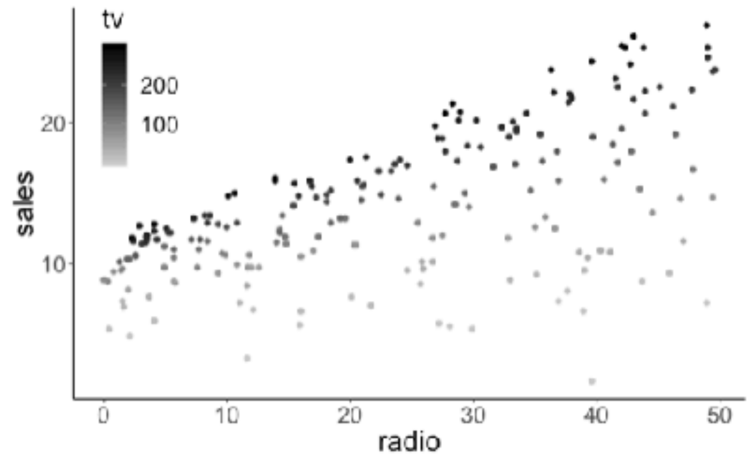
```r
# fit the models
fit_c = lm(sales ~ 1 + tv, data = df.ads)
fit_a = lm(sales ~ 1 + tv + radio, data = df.ads)

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

we reject the H$_0$

```
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



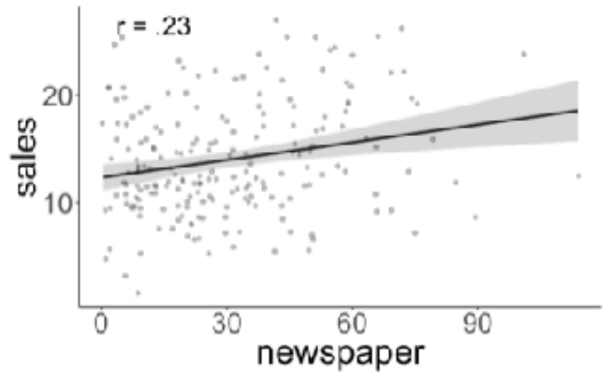
There is a significant relationship between sales and radio ads, controlling for TV ads $F(1, 197) = 546.74$, $p < .001$.

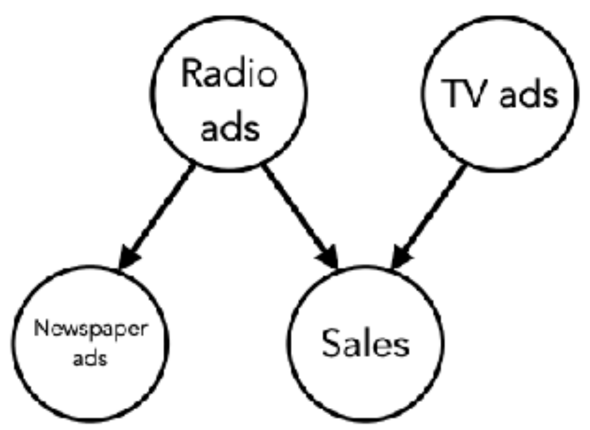
Holding TV ads fixed, an increase in $1000 on radio ads is predicted to increase sales by 190 units [170, 200] (95% confidence intervals).

31

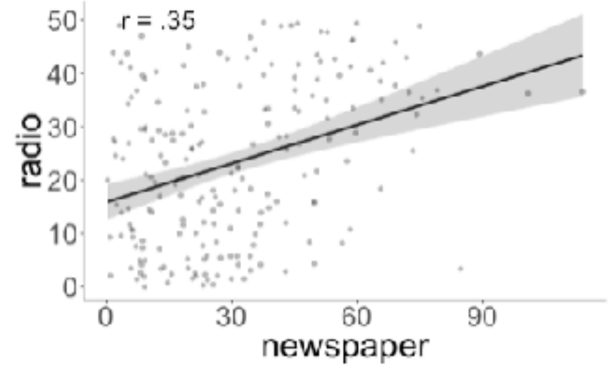
Are newspaper ads and sales related when controlling for radio ads and TV ads?

Relationship between newspaper ads and sales

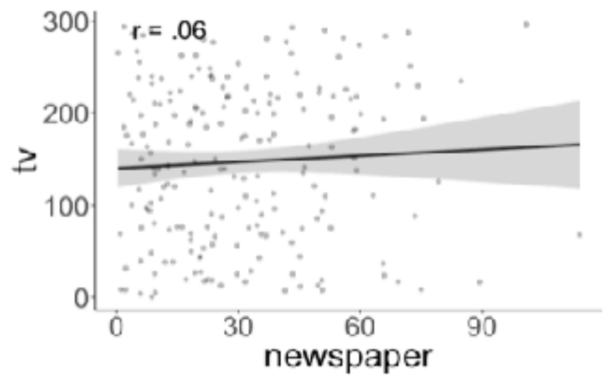




Relationship between newspaper and radio ads

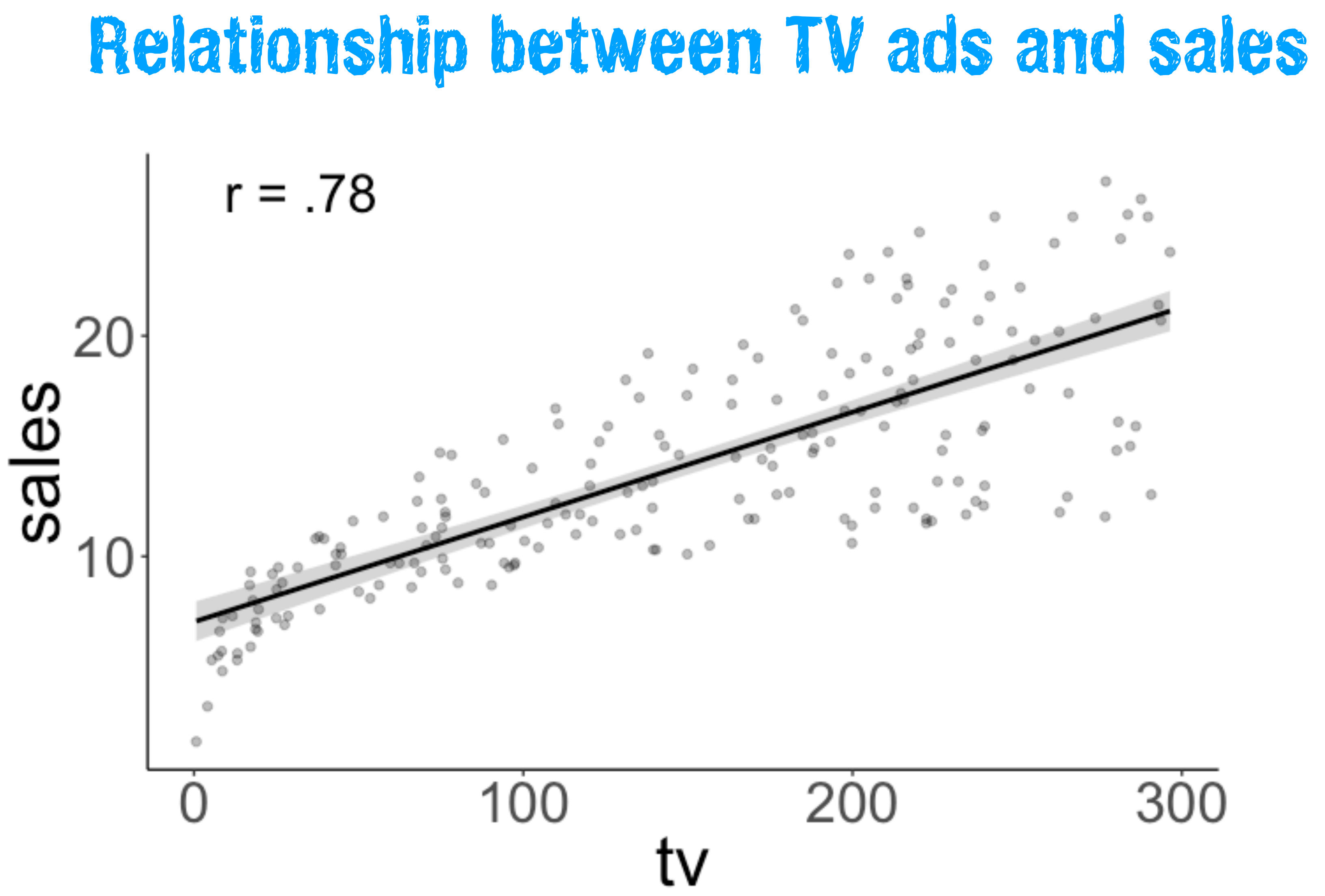


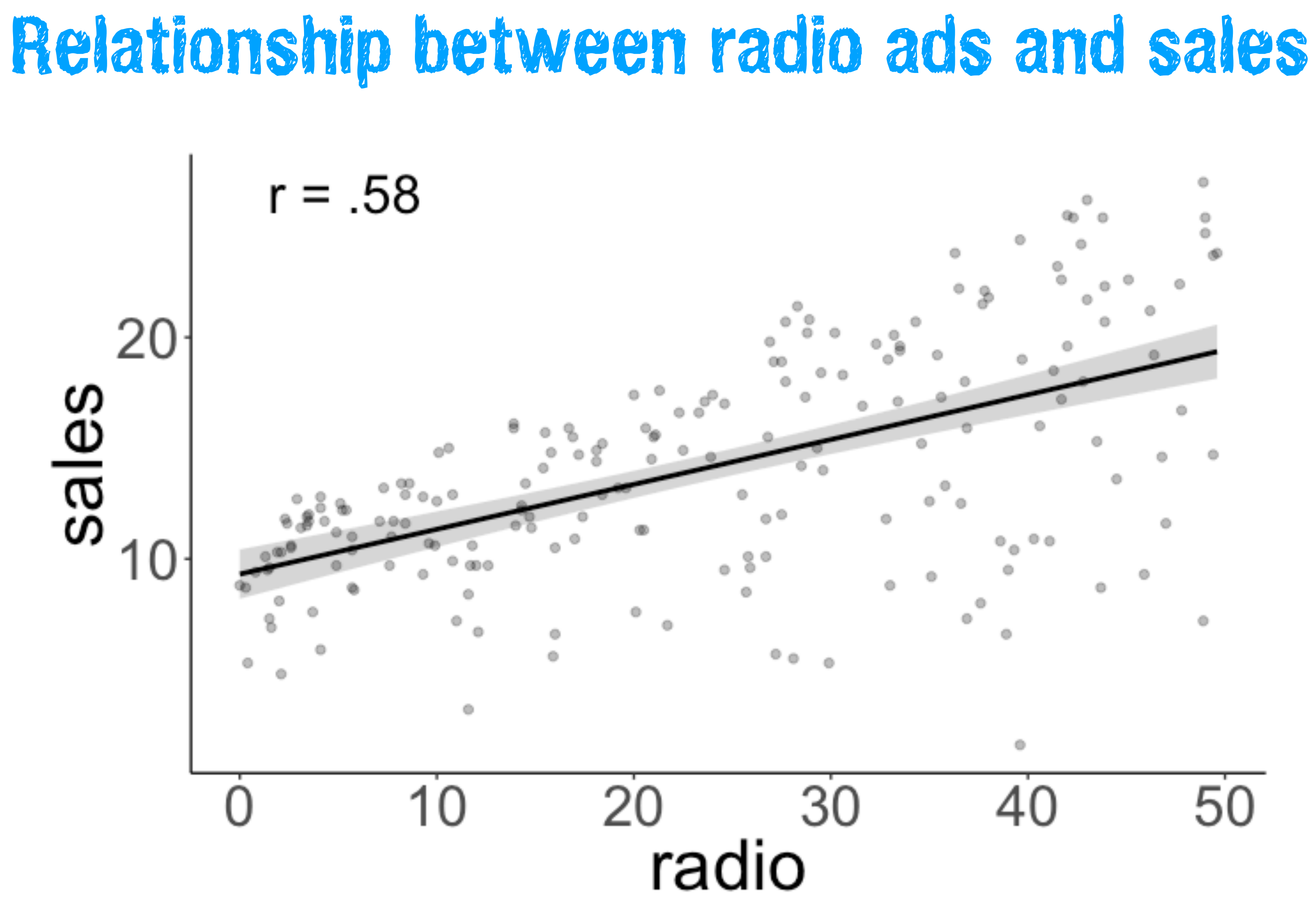
Relationship between newspaper and TV ads



38

Why can't I just run several simple regressions?

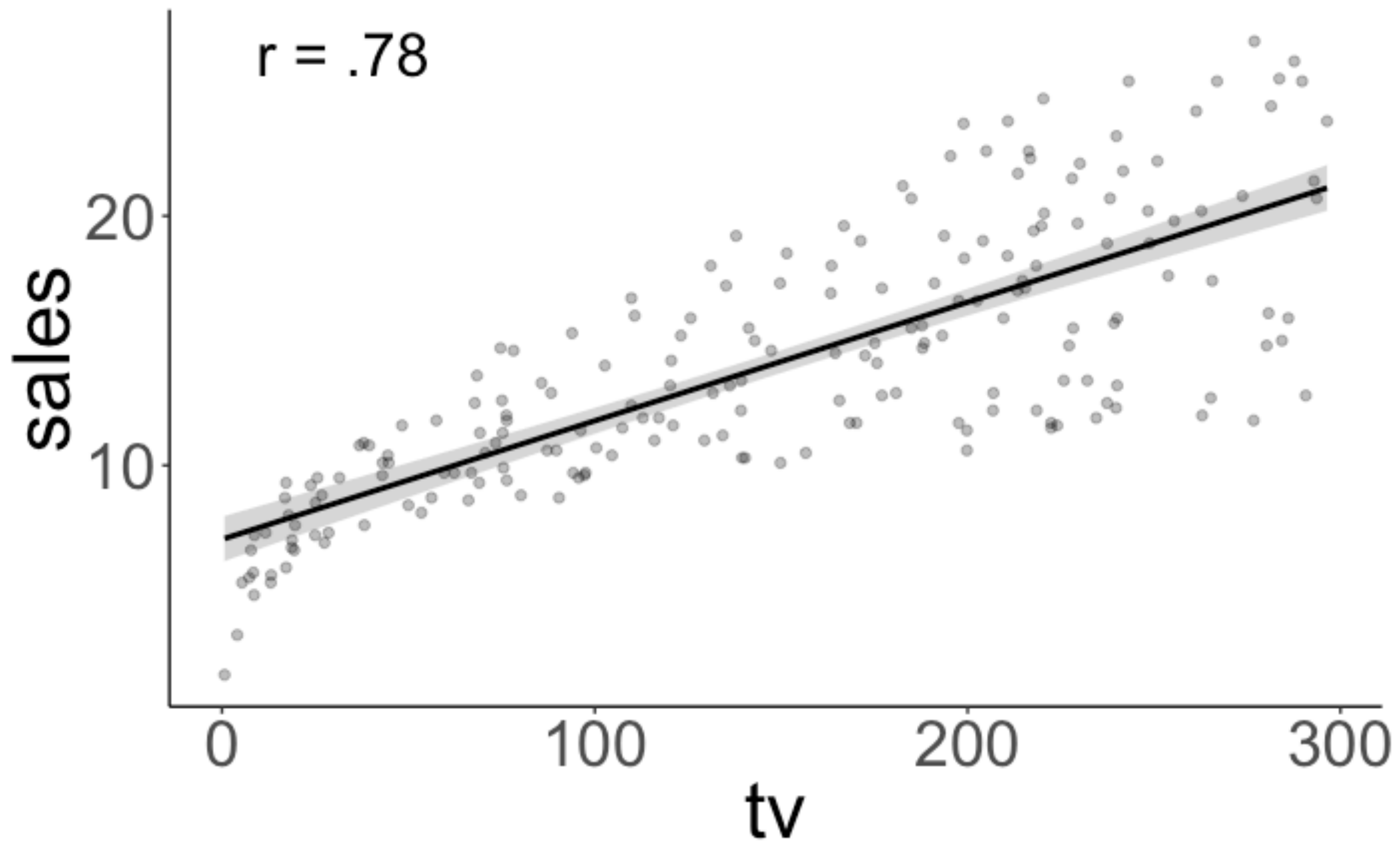




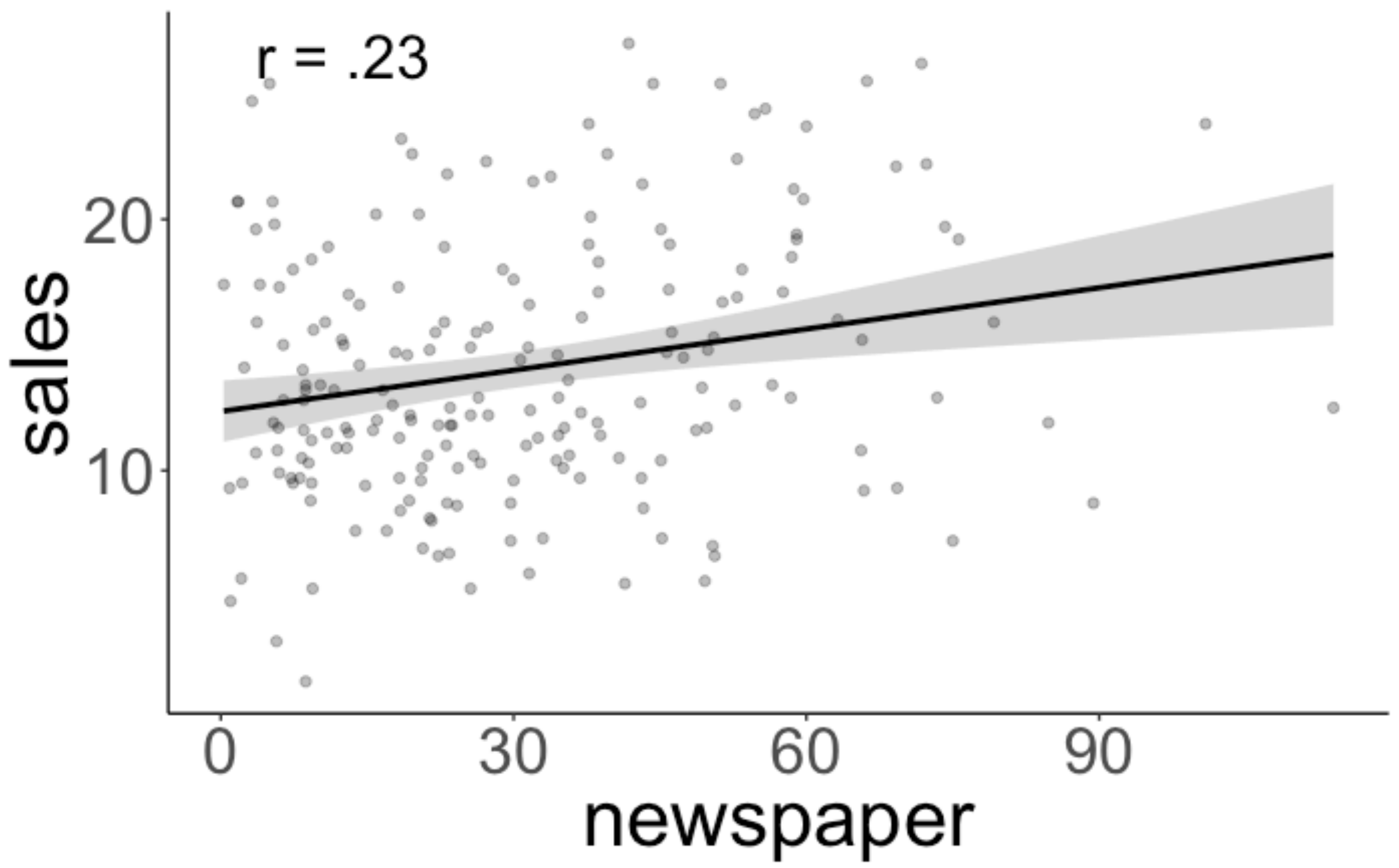
We found that both TV ads and radio ads were related to sales.
But did we need to run a multiple regression? Could we have just looked at correlations?

Correlations are bivariate - ignore the rest of the world
Multiple Regression parameters are UNIQUE effects - parameters indicate the unique variance accounted for by each variable (not the shared part)

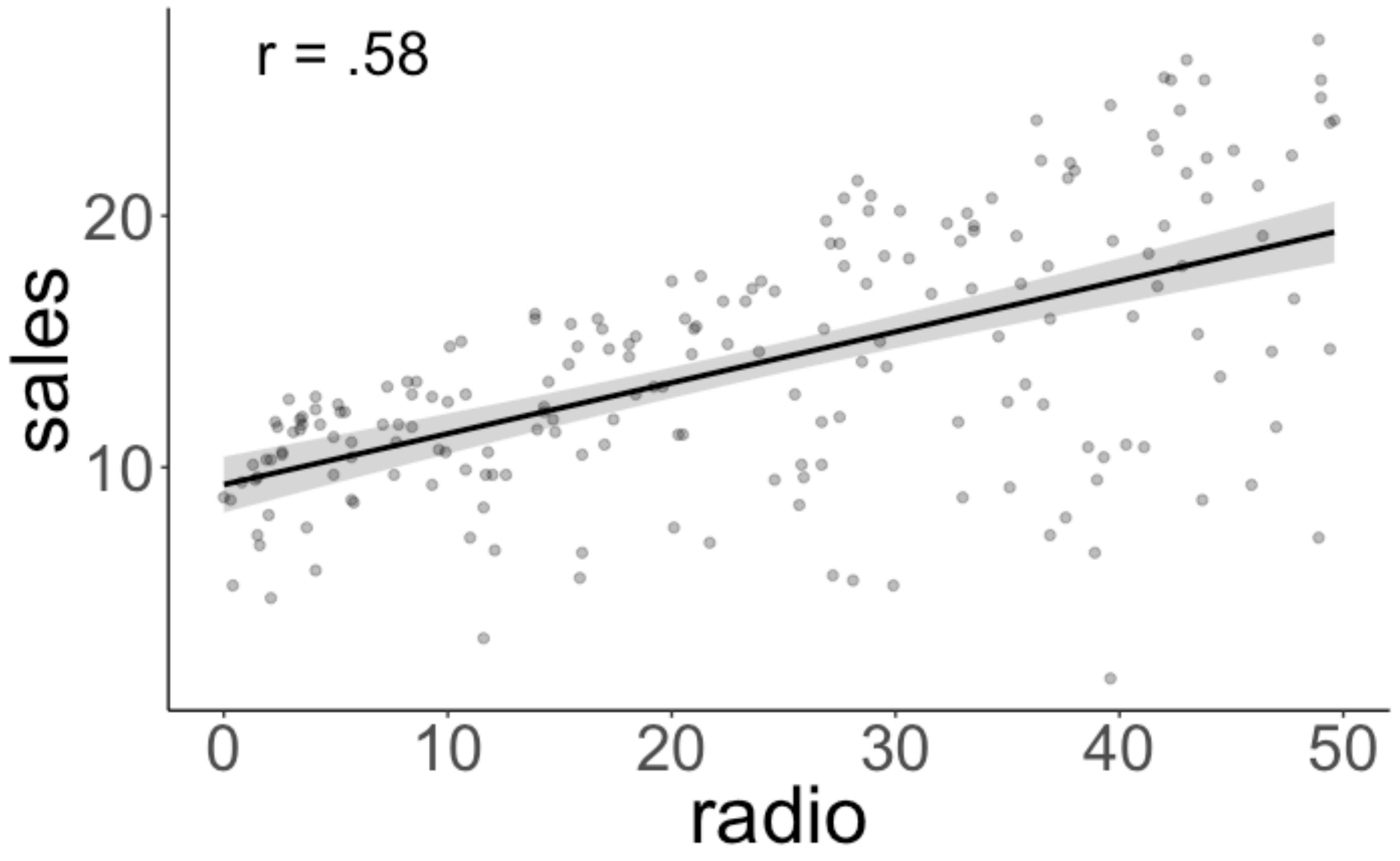
Relationship between TV ads and sales



Relationship between newspaper ads and sales



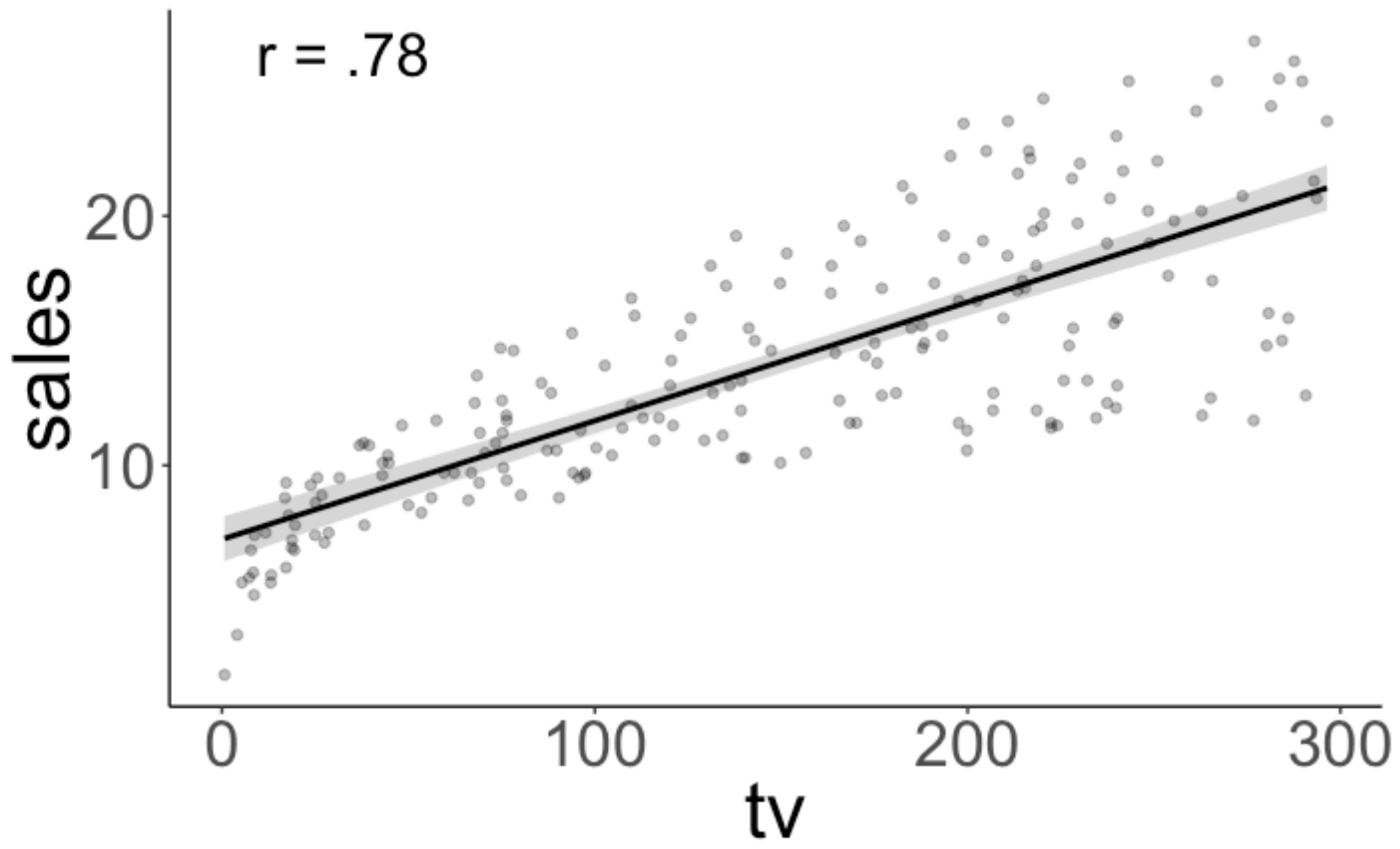
Relationship between radio ads and sales



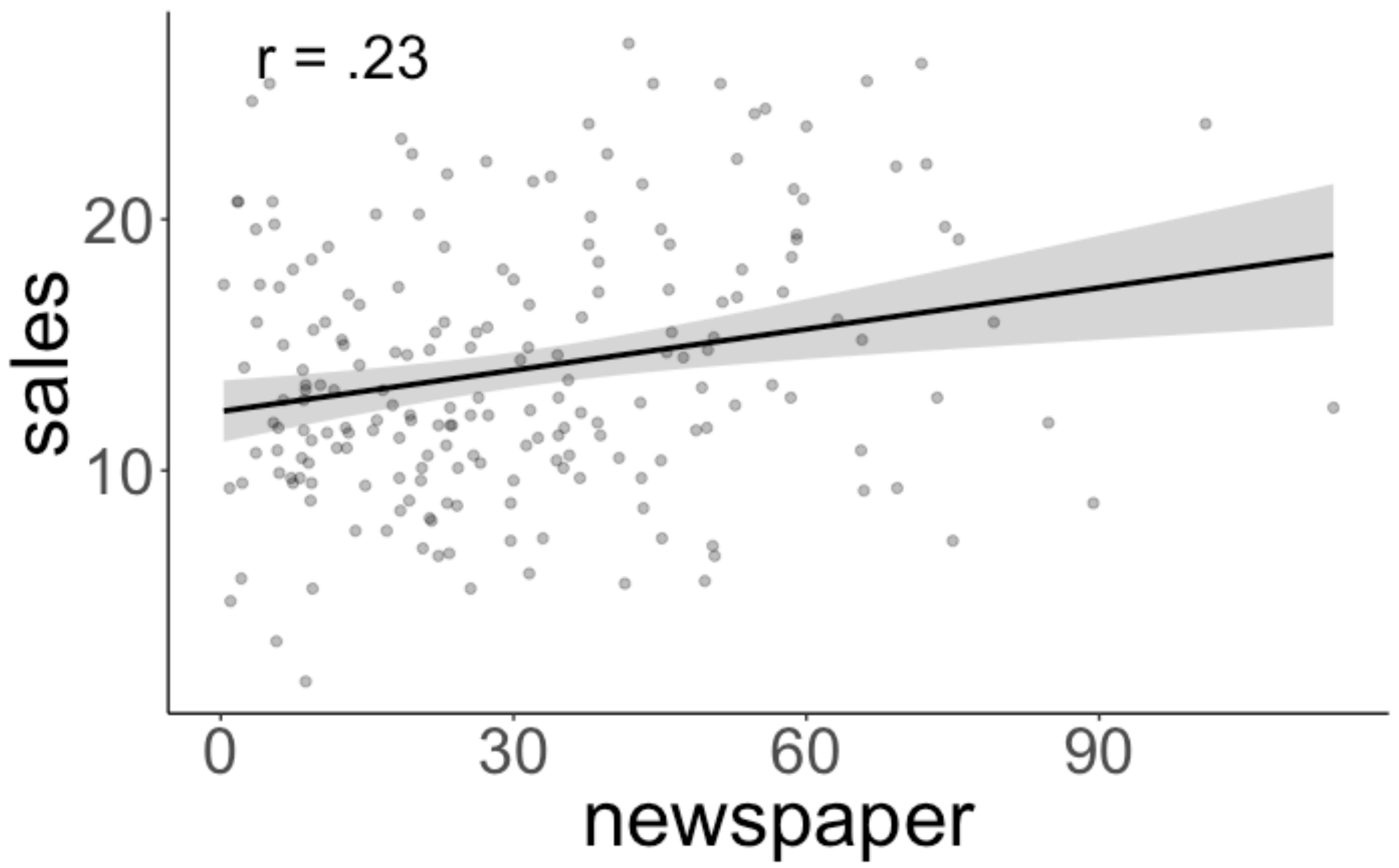
We found that both TV ads and radio ads were related to sales.
But did we need to run a multiple regression? Could we have just looked at correlations?

Correlations are bivariate - ignore the rest of the world
Multiple Regression parameters capture UNIQUE effects - each parameters indicates the unique variance accounted for by that variable (and not the shared part, the shared part is in the R2)

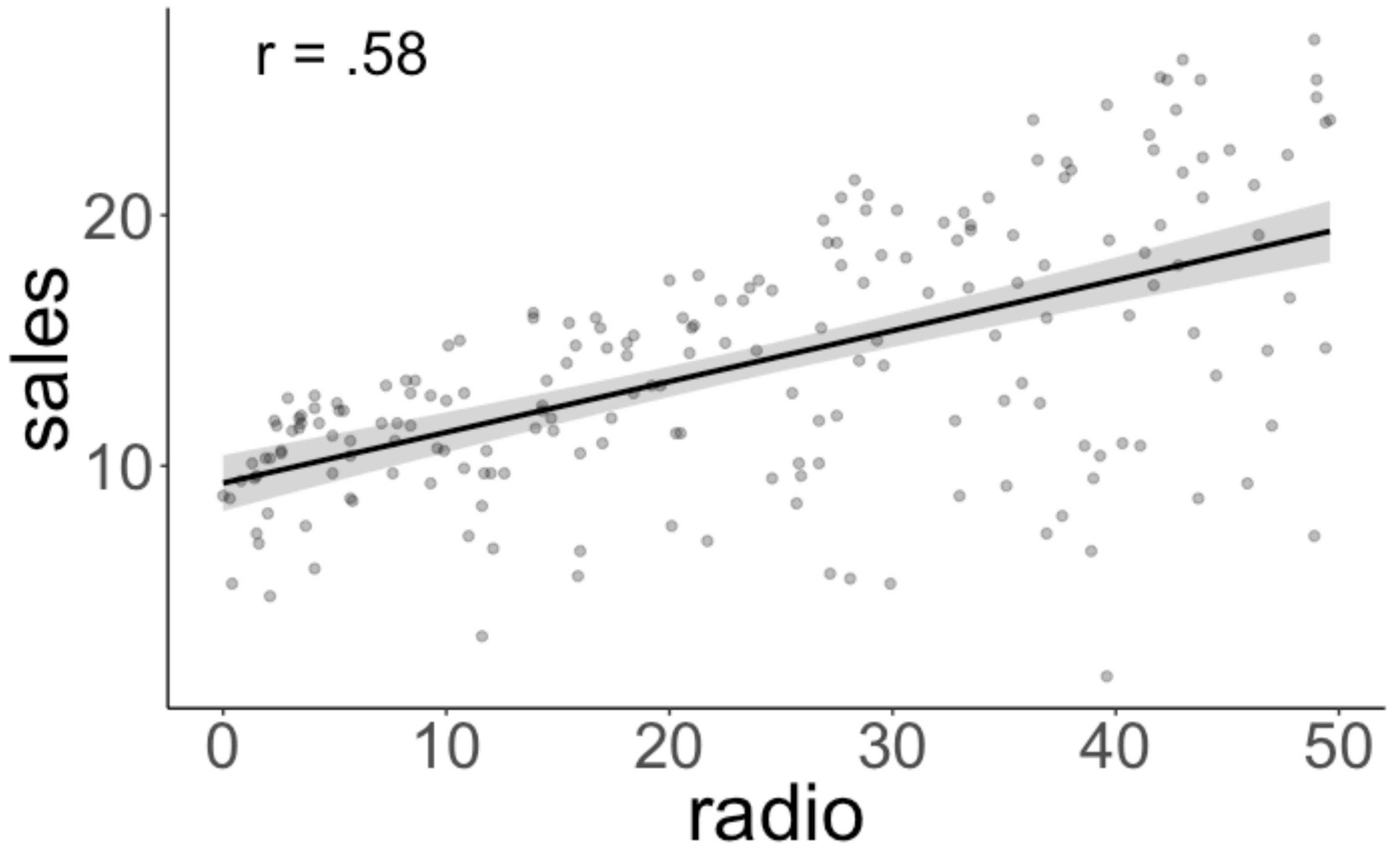
Relationship between TV ads and sales

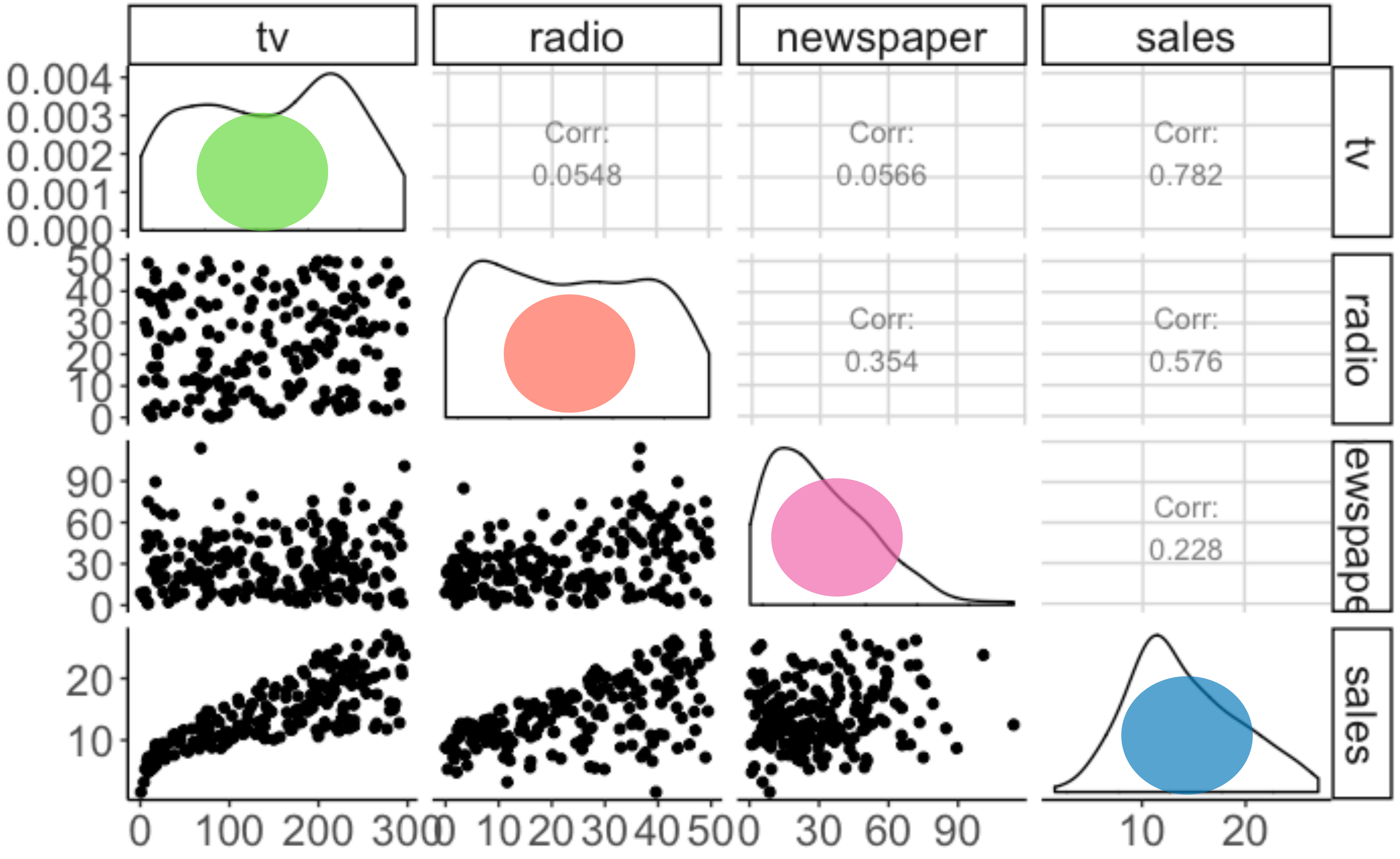


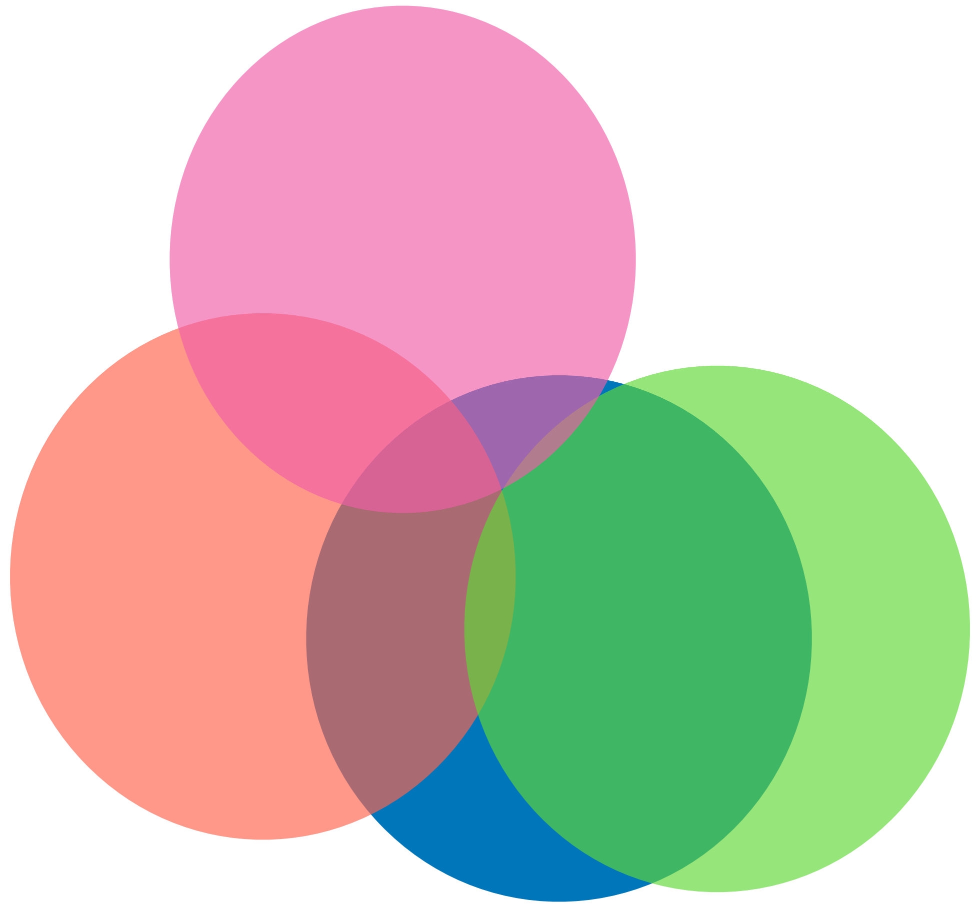
Relationship between newspaper ads and sales



Relationship between radio ads and sales







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

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

```
Analysis of Variance Table

Model 1: sales ~ 1 + tv + radio
Model 2: sales ~ 1 + tv + radio + newspaper
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    197 556.91
2    196 556.83  1  0.088717 0.0312 0.8599
```

it's not worth it

```
sales ~ 1 + tv
sales ~ 1 + tv + newspaper
```

it's worth it

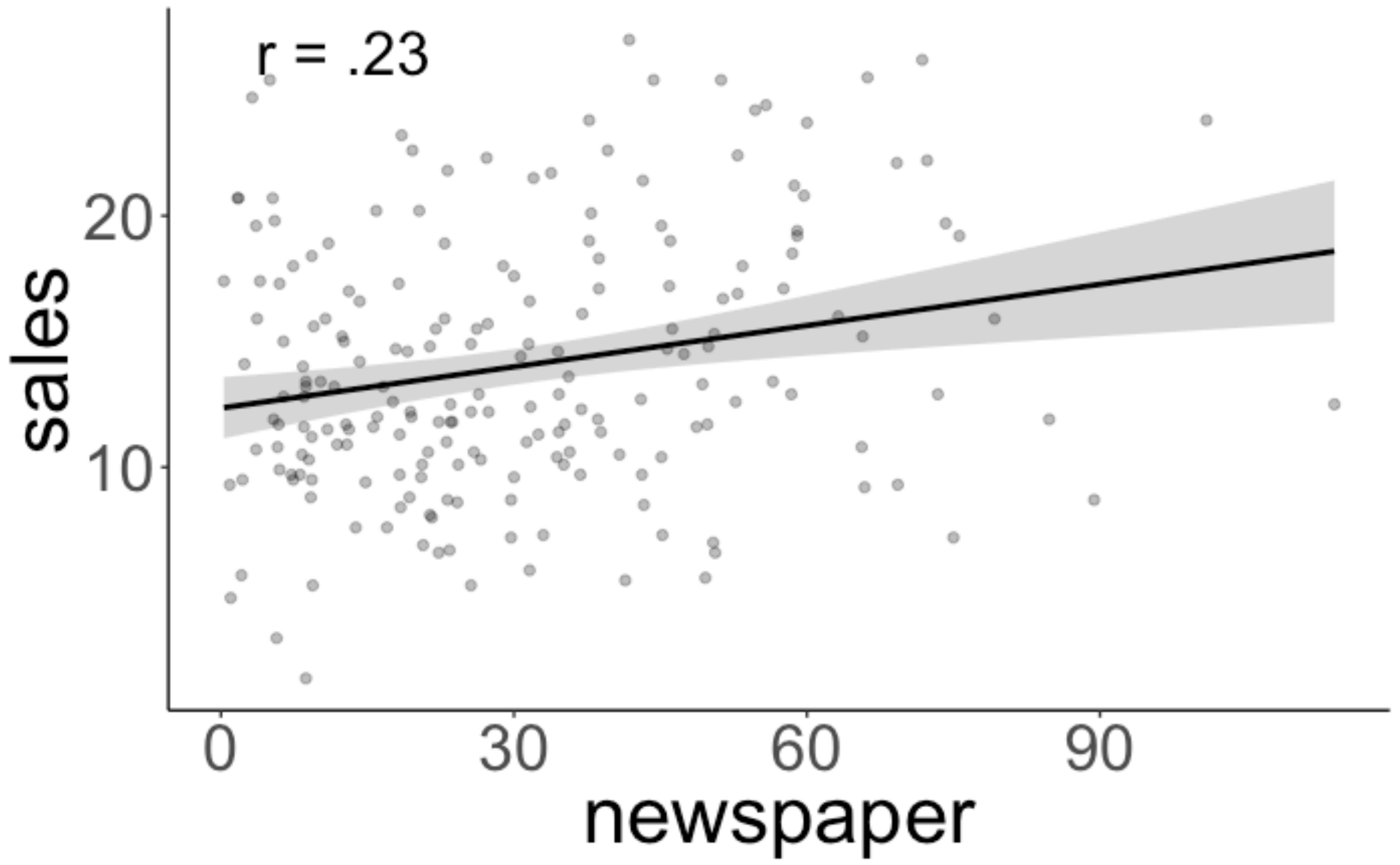
```
sales ~ 1 + radio
sales ~ 1 + radio + newspaper
```

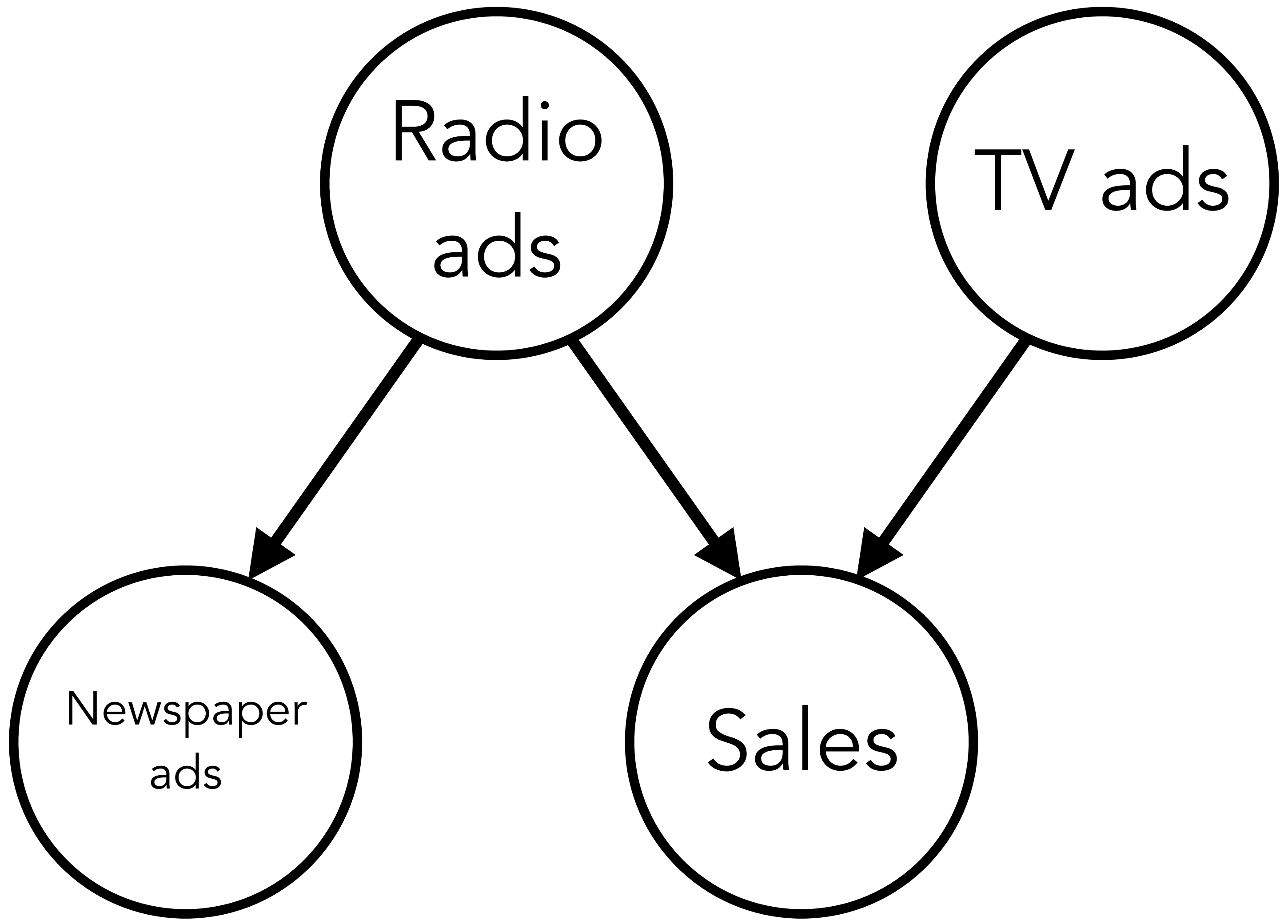
it's not worth it

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

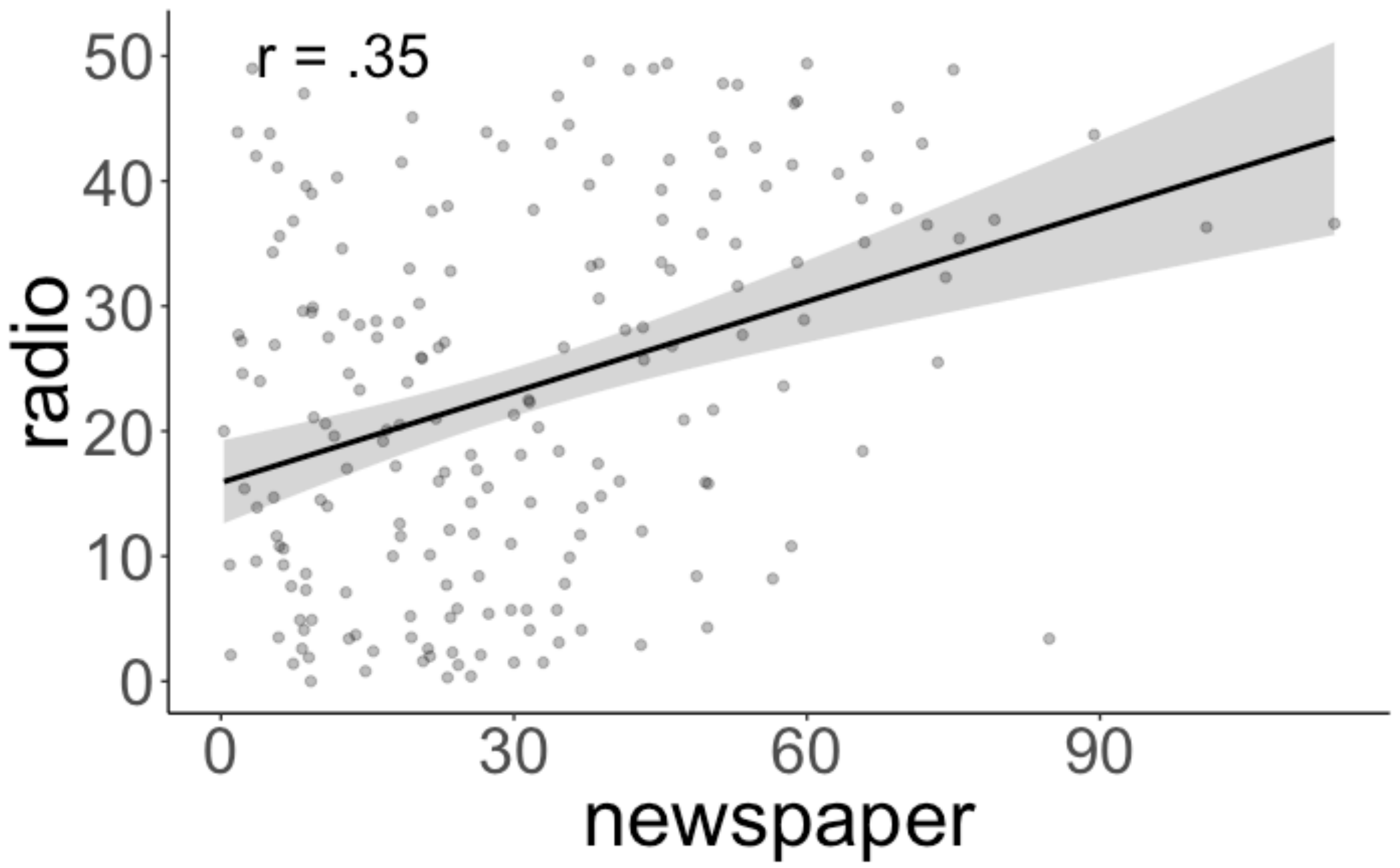
Can begin to develop a causal picture of the system of variables
Post-hoc theory development

Relationship between newspaper ads and sales

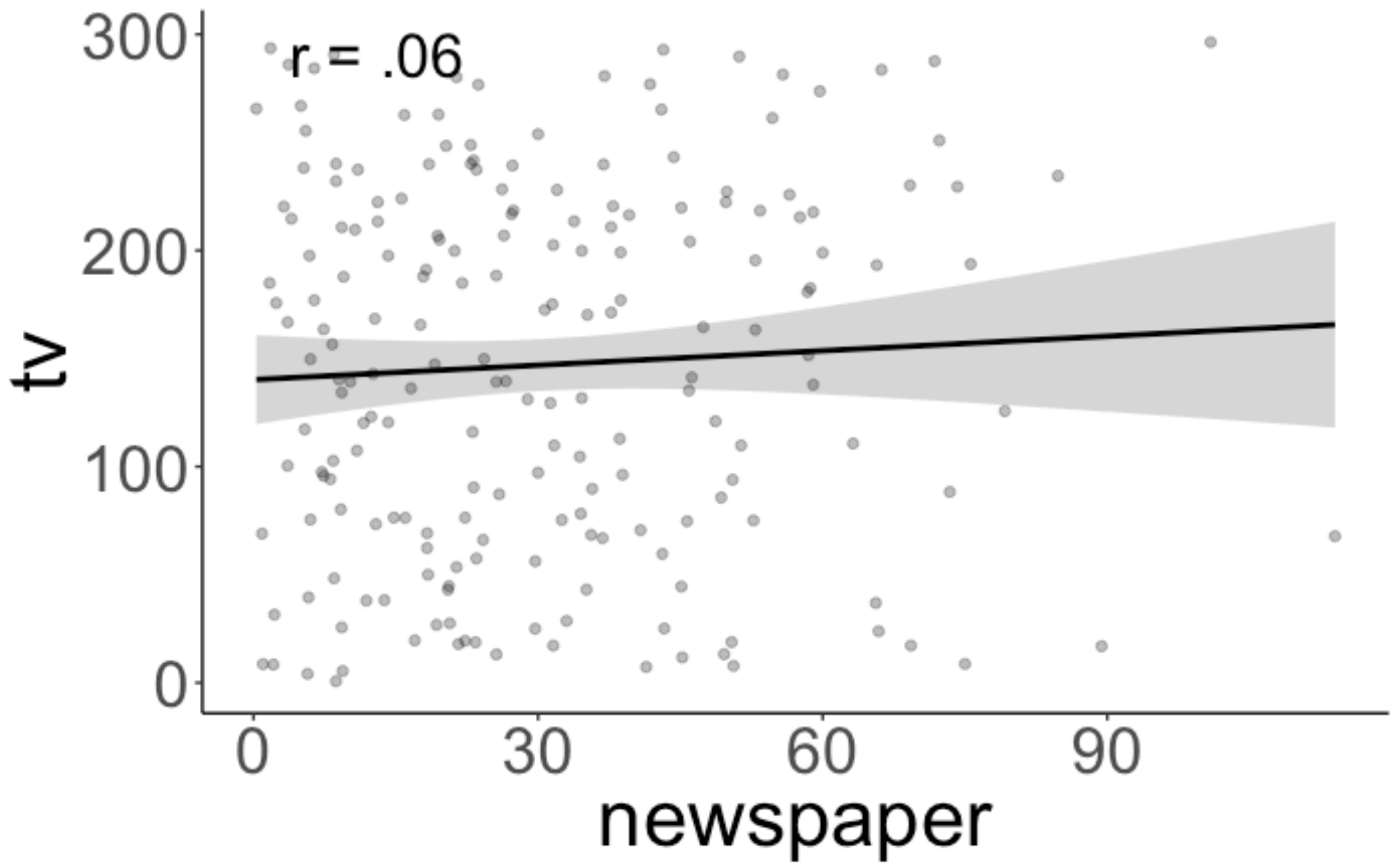




Relationship between newspaper and radio ads



Relationship between newspaper and TV ads



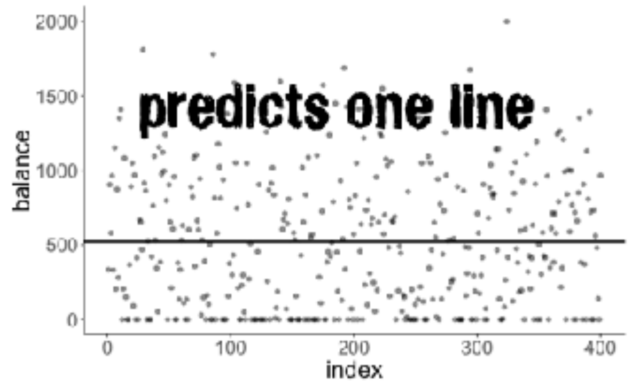
Quick recap: Categorical predictors

H$_0$: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



Fitted model

$$Y_i = 520.02 + e_i$$

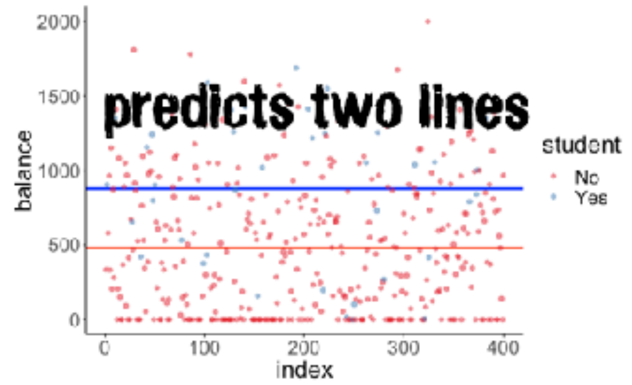
H$_1$: Students and non-students have different balances.

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

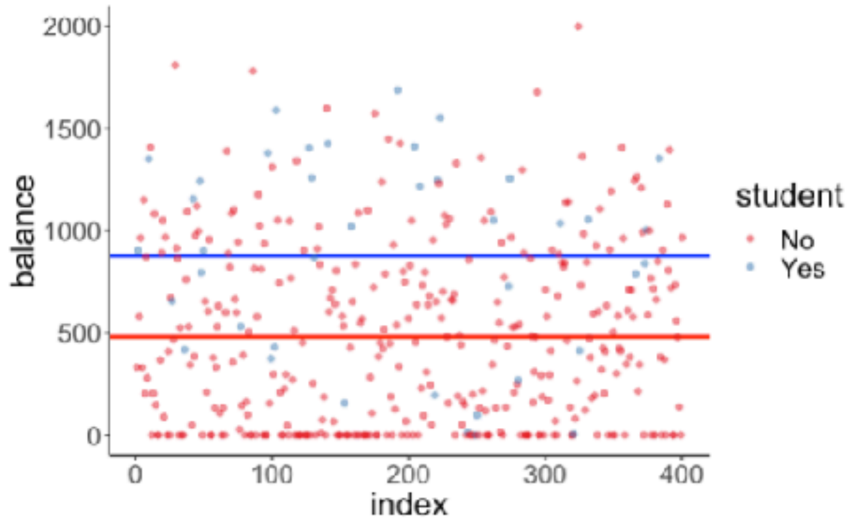
Dummy coding

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

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

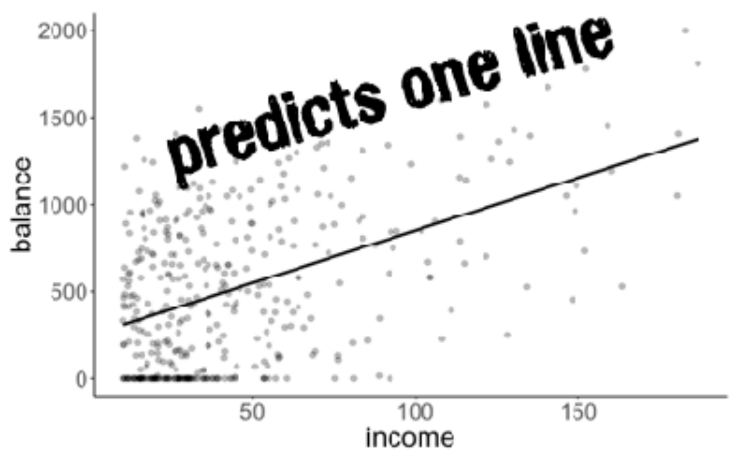
Quick recap: Categorical and continuous predictors

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

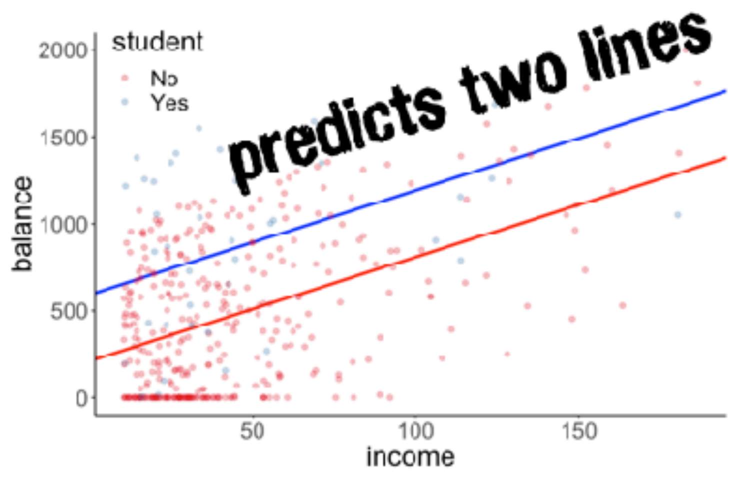
$$\widehat{\text{balance}}_i = 246.515 + 6.048 \cdot \text{income}_i$$

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

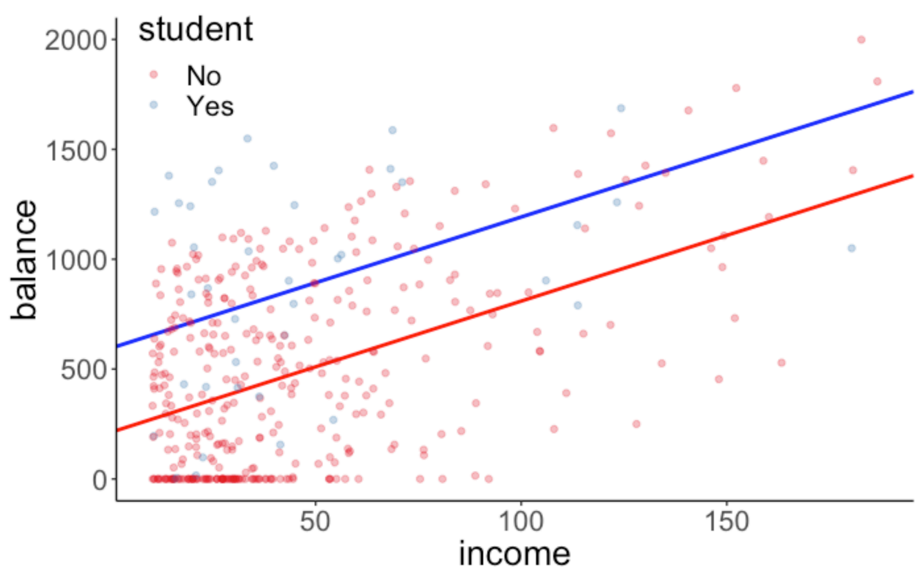


Fitted model

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

52

Interpretation



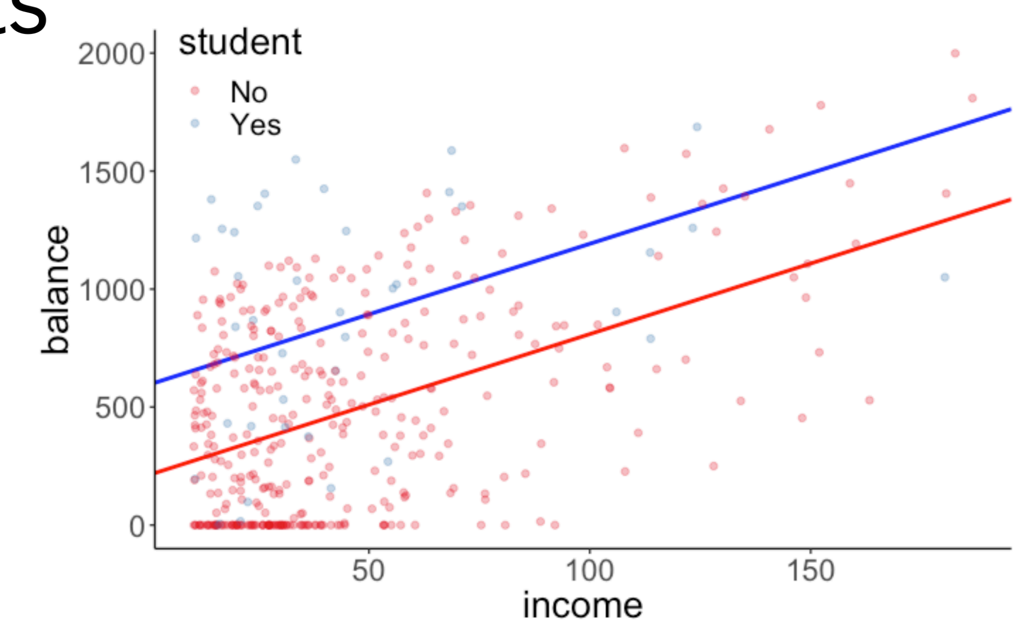
$$\widehat{\text{balance}}_i = 211.14 + 5.98 \cdot \text{income}_i + 382.67 \cdot \text{student}_i$$

if student = "No" $\widehat{\text{balance}}_i = 211.14 + 5.98 \cdot \text{income}_i$

if student = "Yes"
$$\begin{aligned}\widehat{\text{balance}}_i &= 211.14 + 5.98 \cdot \text{income}_i + 382.67 \\ &= 211.14 + 382.67 + 5.98 \cdot \text{income}_i \\ &= 593.81 + 5.98 \cdot \text{income}_i\end{aligned}$$

53

Reporting the results



Controlling for income, students have a significantly higher average credit card balance (*Mean* = 876.83, *SD* = 490.00) than non-students (*Mean* = 480.37, *SD* = 439.41), $F(1, 397) = 34.331$, $p < .001$.

Interactions

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

Compact Model

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

R-formula for these?

Augmented Model

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

Interaction ~ Moderation

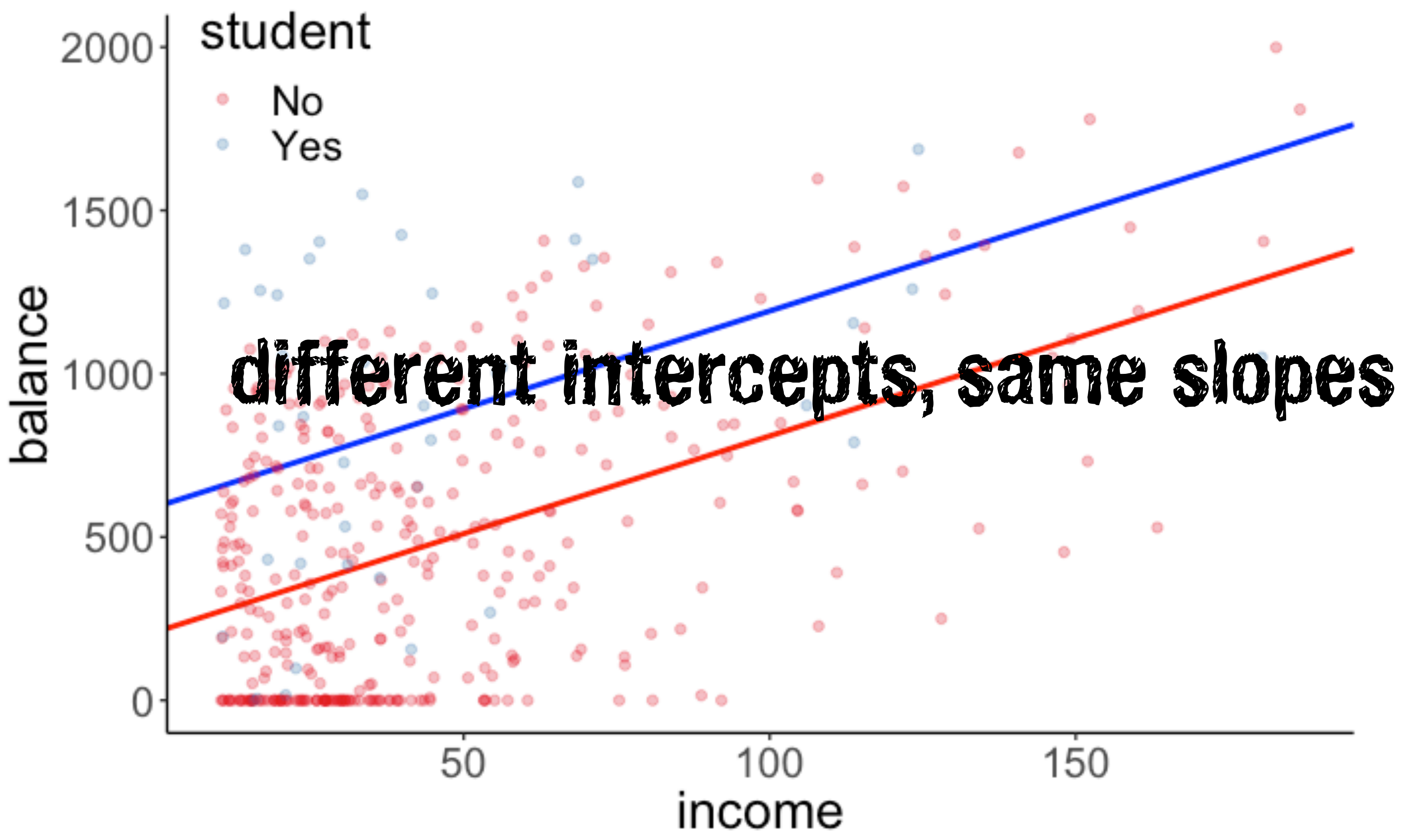
Does student status moderate the relationship between level of income and balance?

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

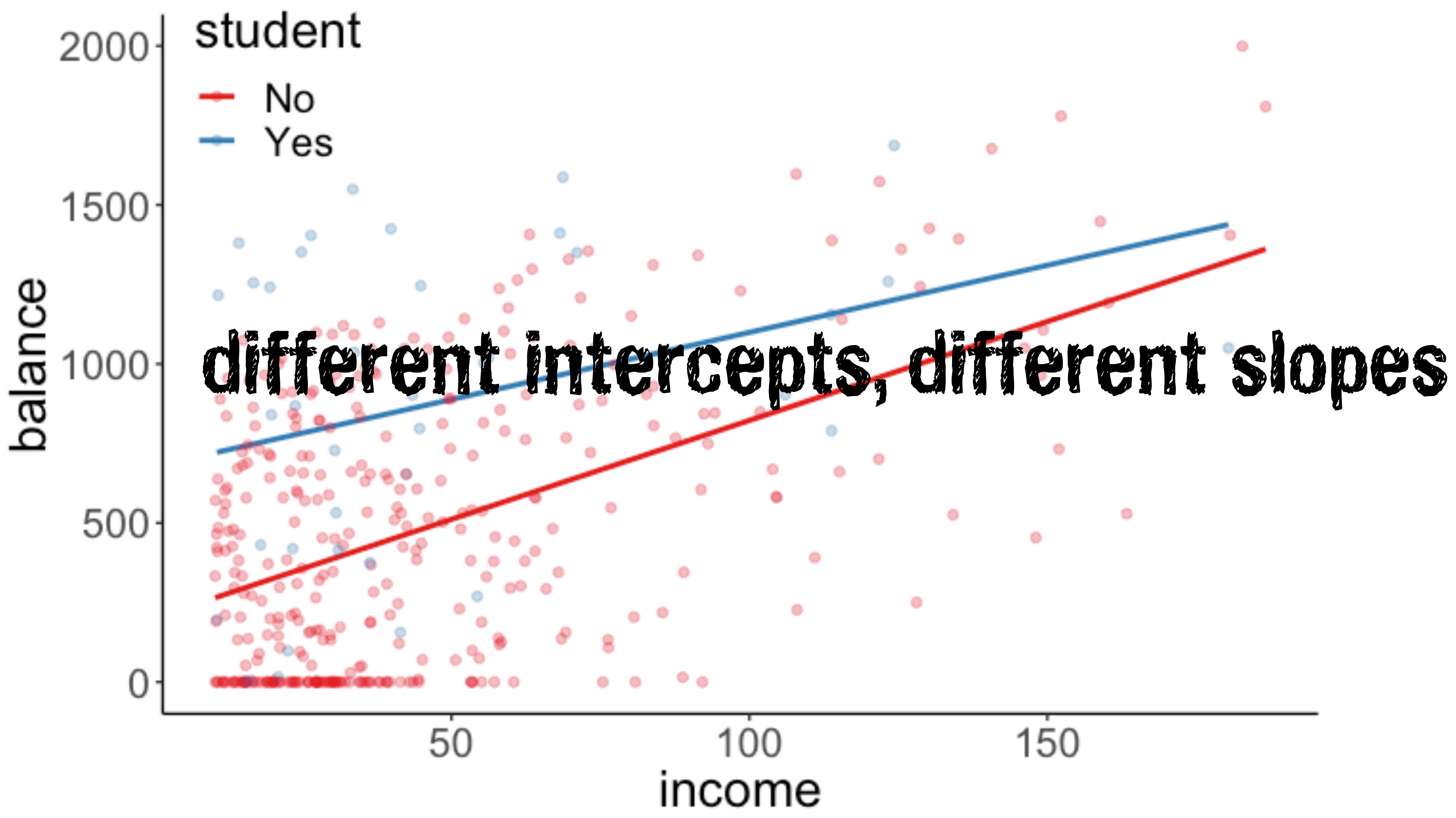
$$\widehat{\text{balance}}_i = 211.14 + 5.98 \cdot \text{income}_i + 382.67 \cdot \text{student}_i$$

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

Model A

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

Model prediction



Fitted model

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

Worth it?

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

```
# fit models
fit_c = lm(formula = balance ~ 1 + income + student, data = df.credit)
fit_a = lm(formula = balance ~ 1 + income * student, data = df.credit)

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

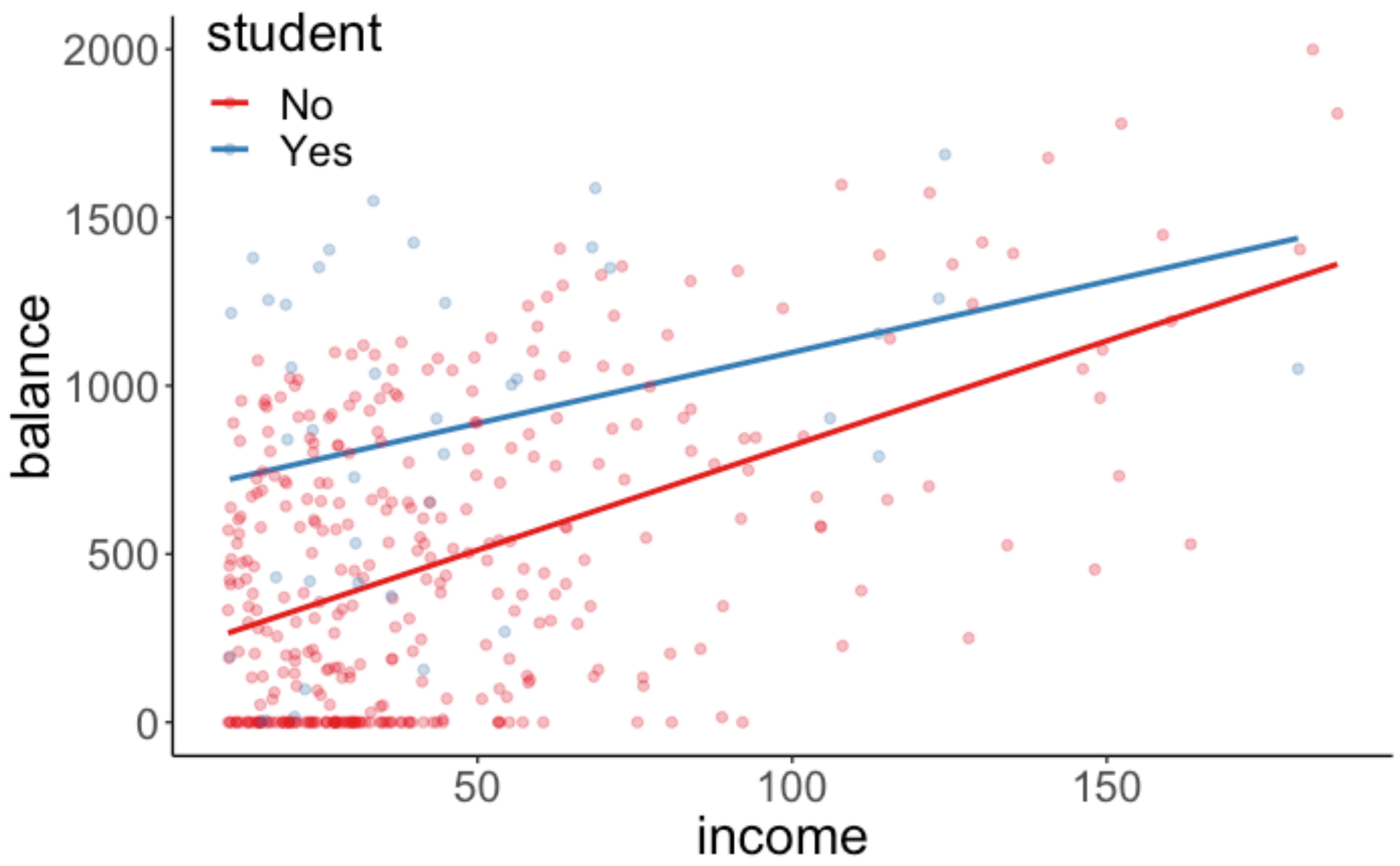
Equations for these?

```
Analysis of Variance Table

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

not worth it!

```
# alternative coding for model a
fit_a = lm(formula = balance ~ 1 + income + student + income:student, data = df.credit)
```

Interpretation



$$\widehat{\text{balance}}_i = 200.62 + 6.22 \cdot \text{income}_i + 476.68 \cdot \text{student}_i - 2.00 \cdot (\text{income}_i \times \text{student}_i) + 0$$

if student = "No" $\widehat{\text{balance}}_i = 200.62 + 6.22 \cdot \text{income}_i + (476.68 \cdot 0) - 2.00 \cdot (\text{income}_i \times 0)$

if student = "Yes"

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

Same as calculating interaction variable by hand?

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

Explicitly encode the interaction

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

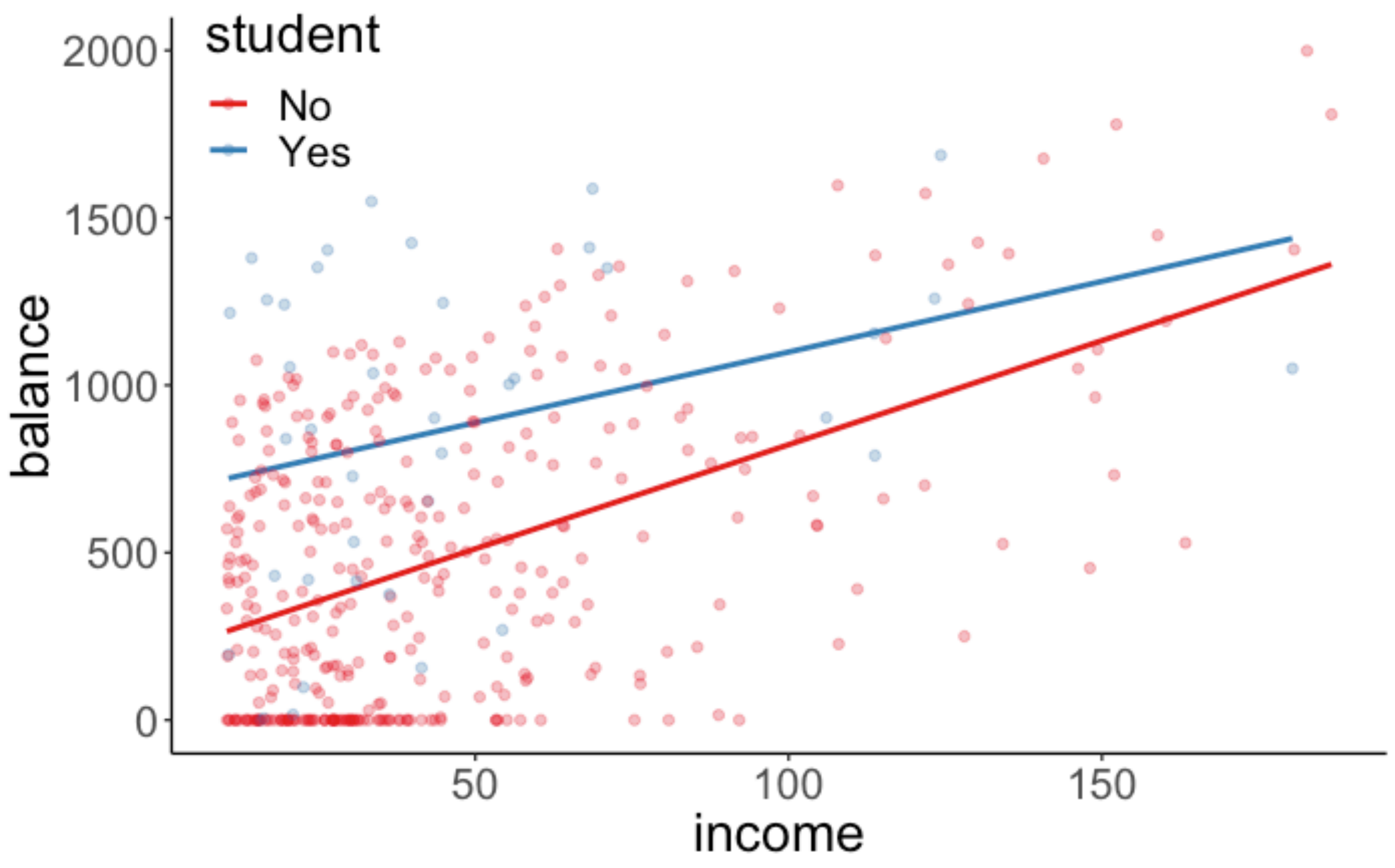
balance	income	student	student_dummy	income_student
333	14.89	No	0	0.00
903	106.03	Yes	1	106.03
580	104.59	No	0	0.00
964	148.92	No	0	0.00
331	55.88	No	0	0.00
1151	80.18	No	0	0.00
203	21.00	No	0	0.00
872	71.41	No	0	0.00
279	15.12	No	0	0.00
1350	71.06	Yes	1	71.06

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

`fit1` and `fit2` are identical!

How to report results of interaction

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



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

For *students*, an increase in $1000 income is associated with an increase in $4.22 of average credit card balance.

$4.22 = ($6.22 - $2.00) = (b2 + b3)

Alternative:
There was no evidence that student status *moderated* the relationship between income and balance, *F*(1, 396) = 1.33, p = 0.25.

`lm()` output

`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
```

?

```r
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)
```

```r
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

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

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

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

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

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

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

very important

```
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 (student = 0)

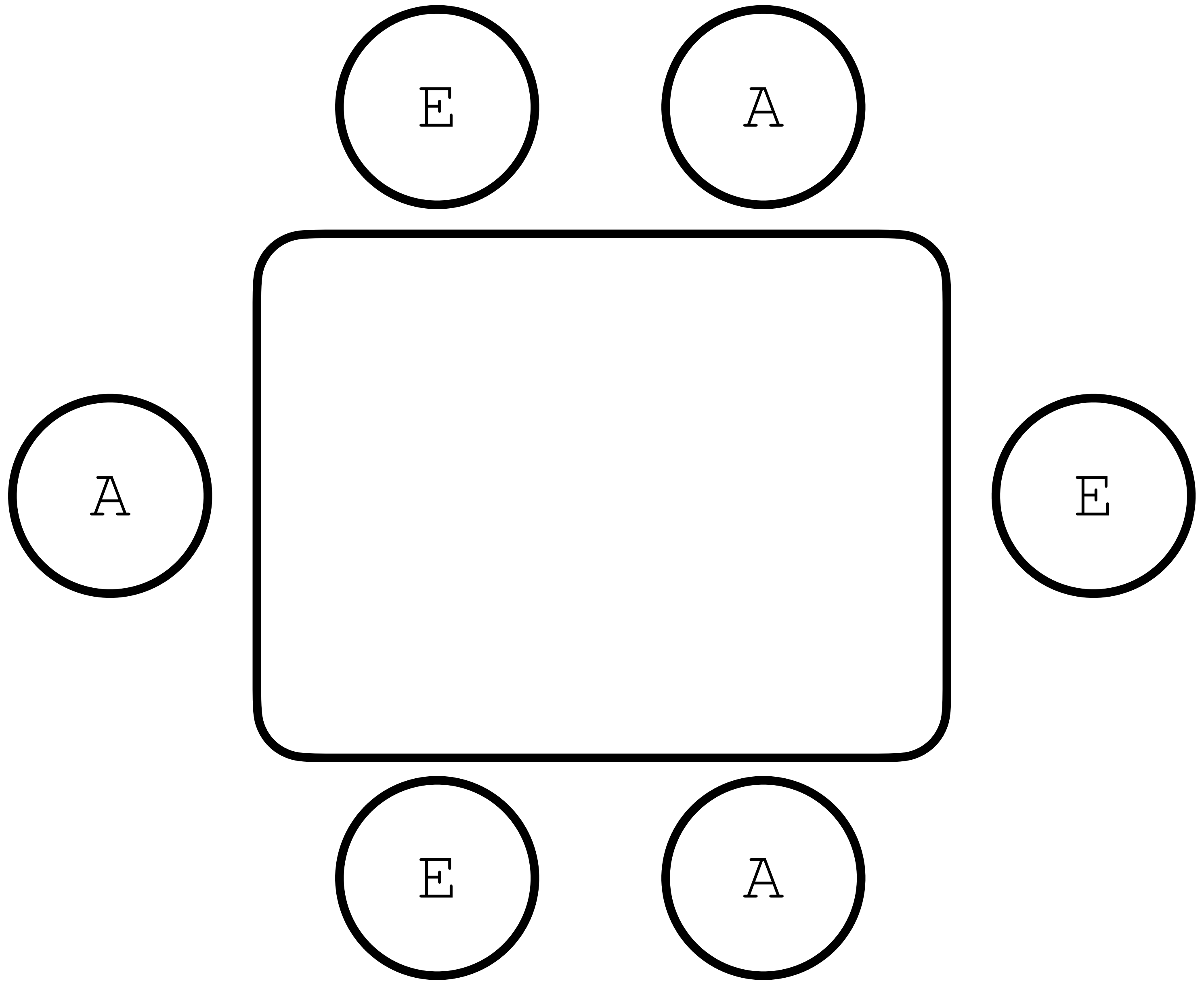
we'll talk more about the difference between simple/conditional effects and main effects soon!

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*

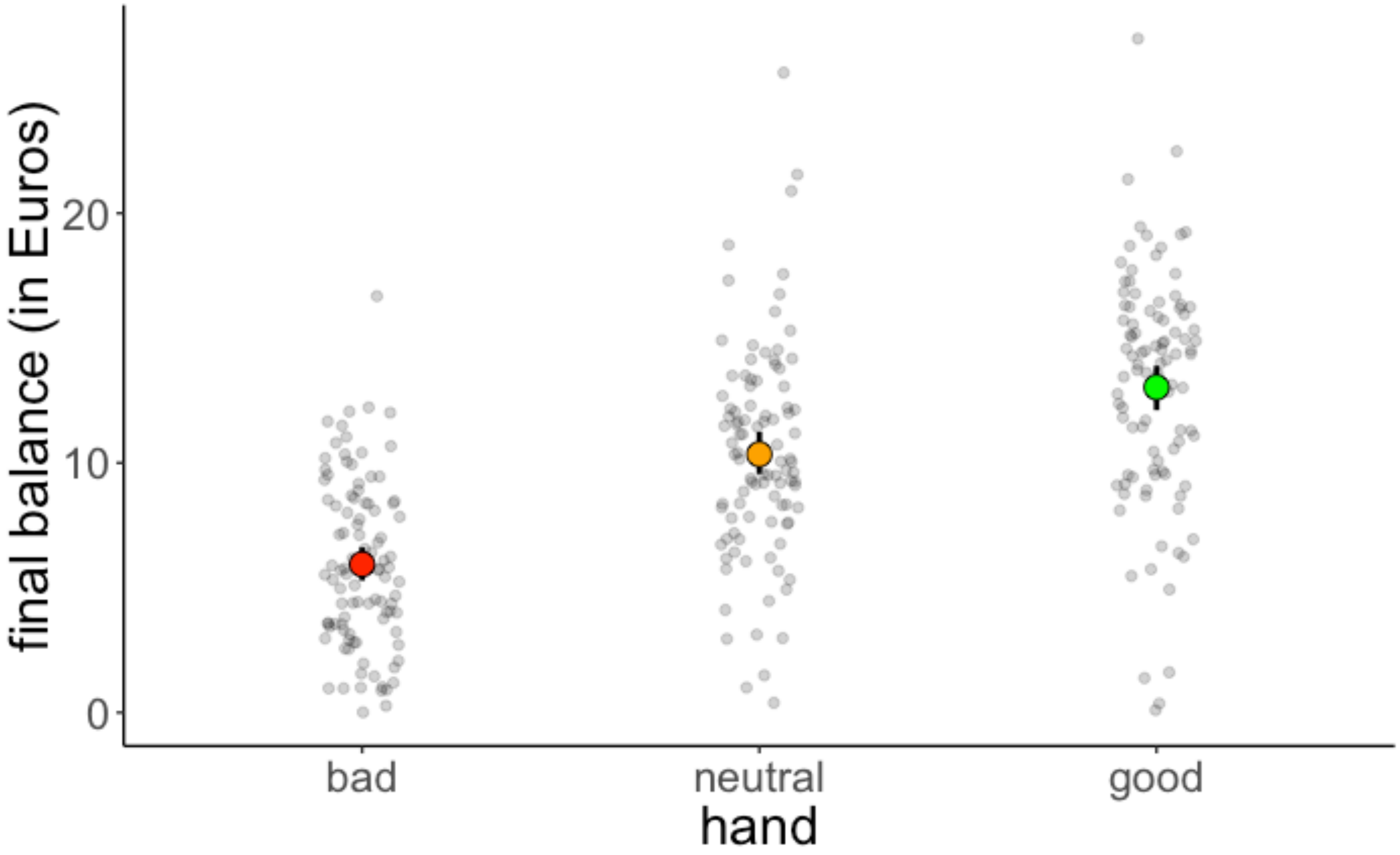
Do better hands win more money?

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}

Does card quality affect the final balance?

Visualize the data first



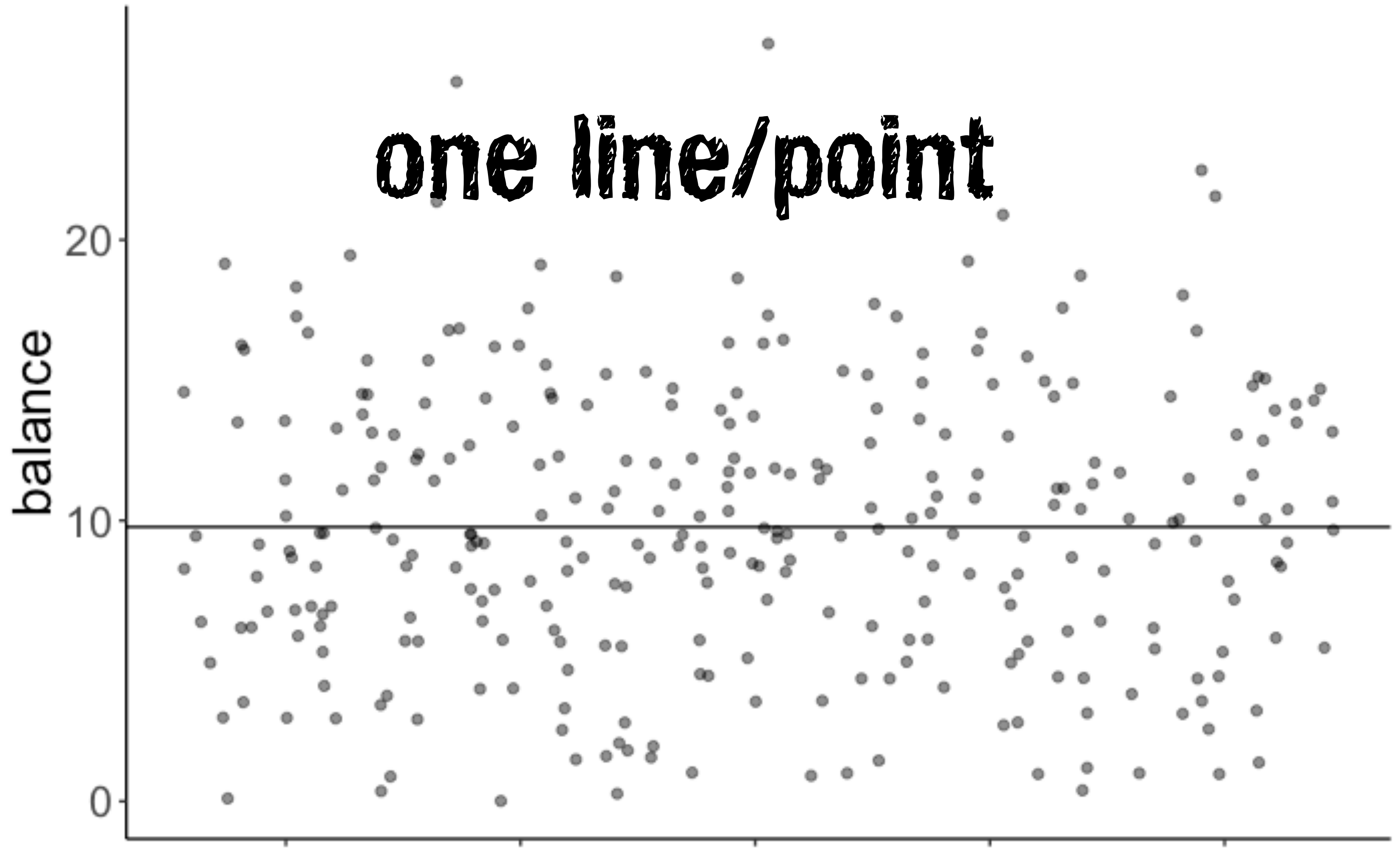
card quality

H_0: Card quality does not affect the final balance.

Model C

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

Model prediction



Fitted model

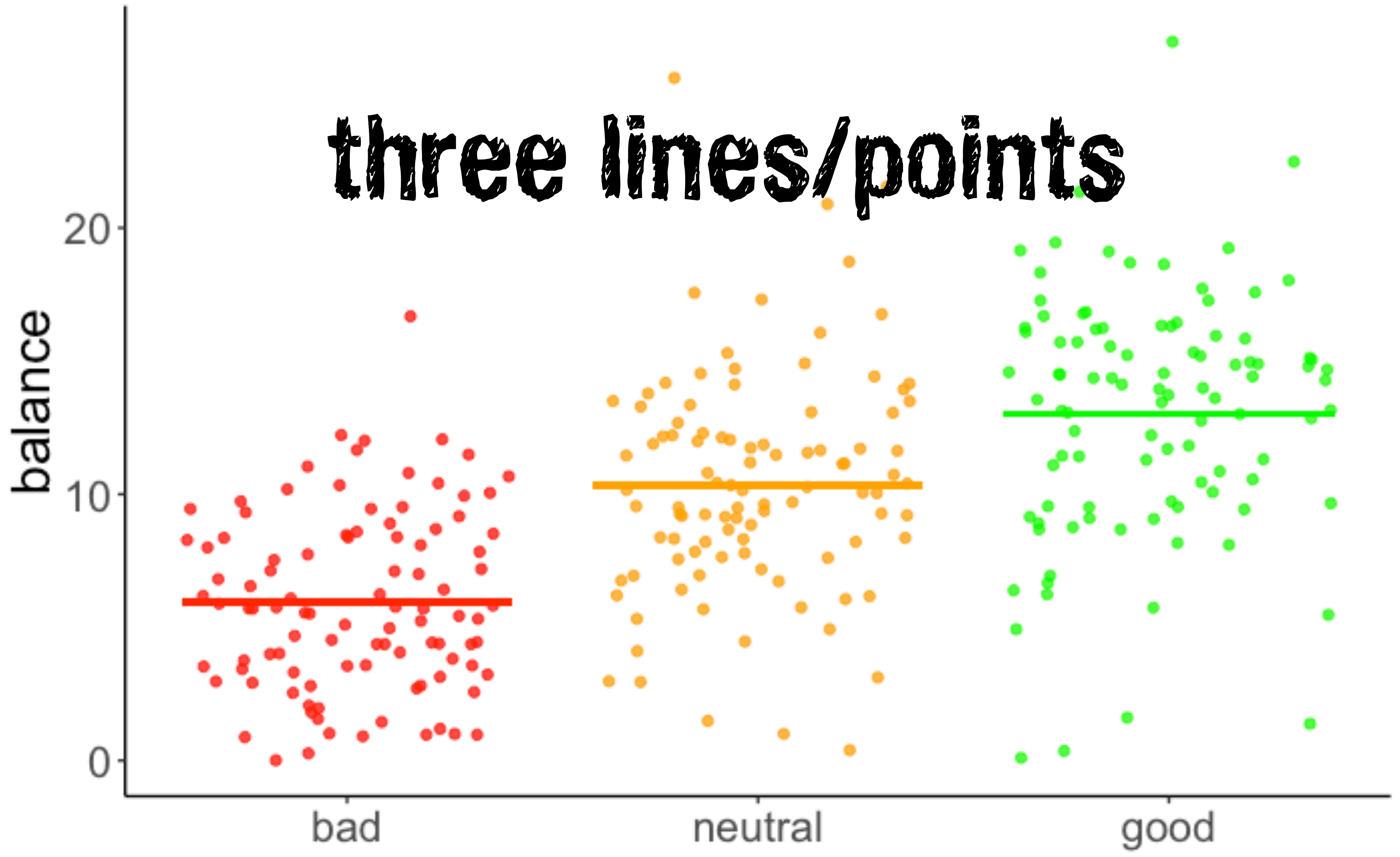
$$\widehat{\text{balance}}_i = 9.77$$

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

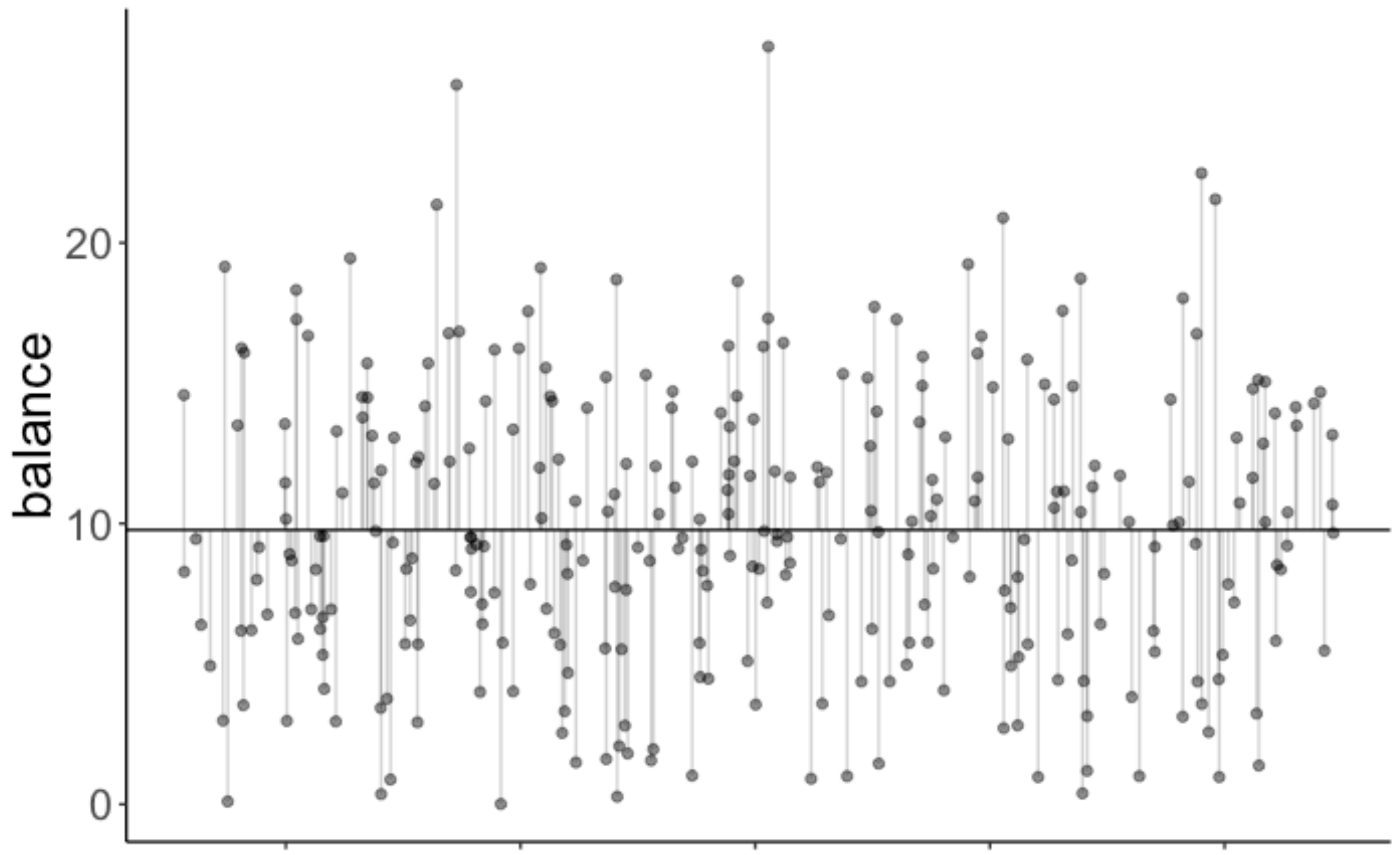
$$\widehat{\text{balance}}_i = 5.94 + 4.41 \cdot \text{hand\_neutral}_i + 7.08 \cdot \text{hand\_good}_i$$

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

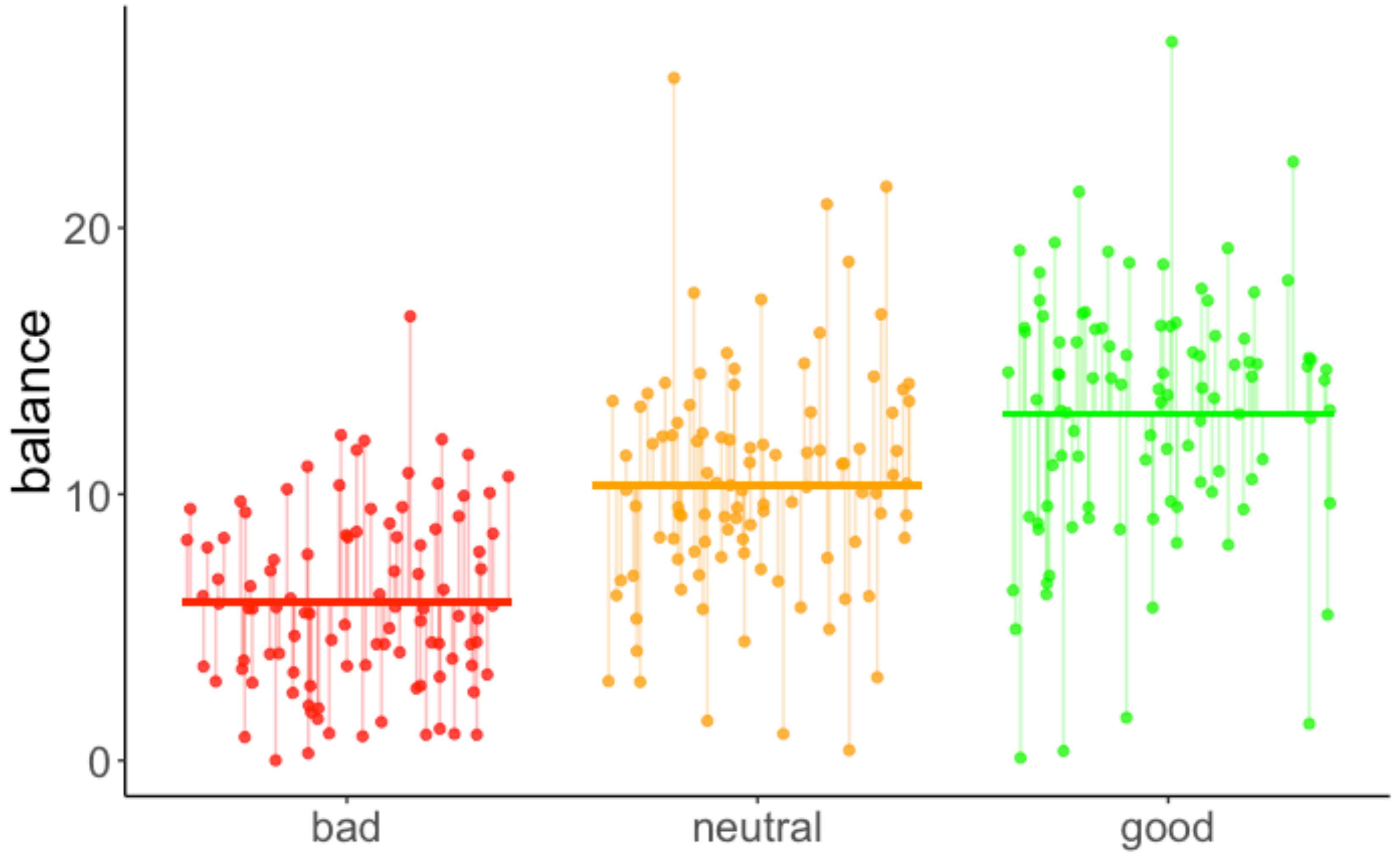
$$\widehat{\text{balance}}_i = 9.77$$

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 ~ 1 + 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

```r
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
```

Interpreting the results

regression coefficients encode differences between group means

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

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

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

if hand == "bad":

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

if hand == "neutral":

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

if hand == "good":

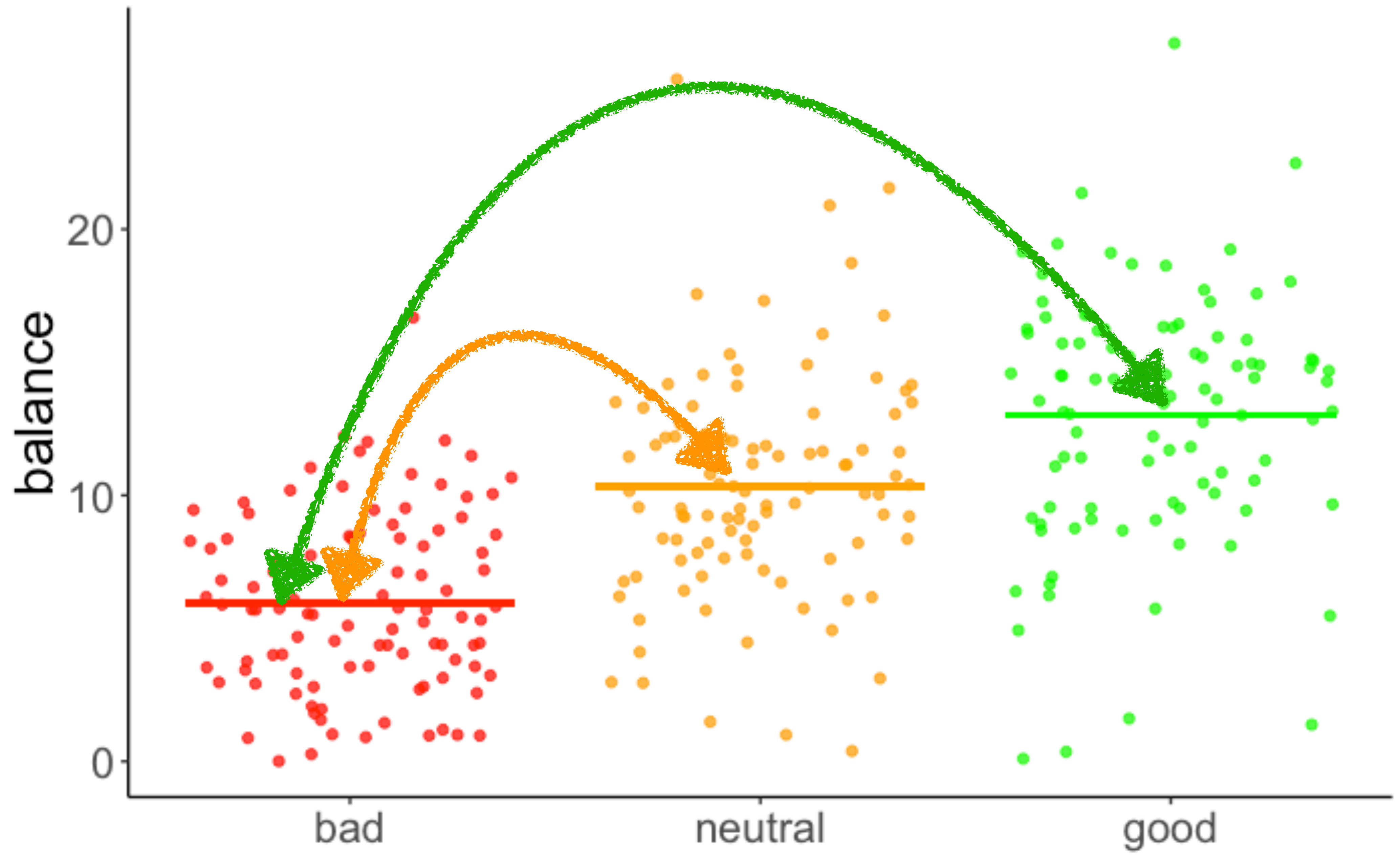
$$\widehat{\text{balance}}_i = 5.94 + 7.08 = 13.02$$

Interpreting the results

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

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



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

```r
lm(formula = balance ~ 1 + 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?

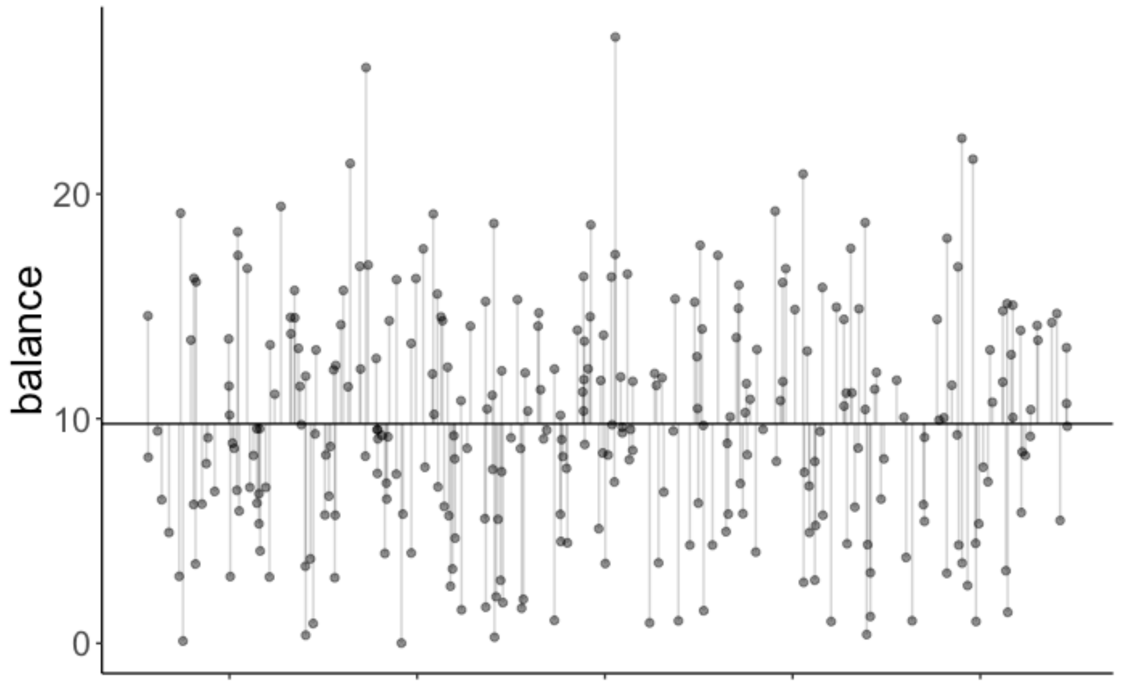
```r
# fit the models
fit_c = lm(formula = balance ~ 1, data = df.poker)
fit_a = lm(formula = balance ~ hand, data = df.poker)

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

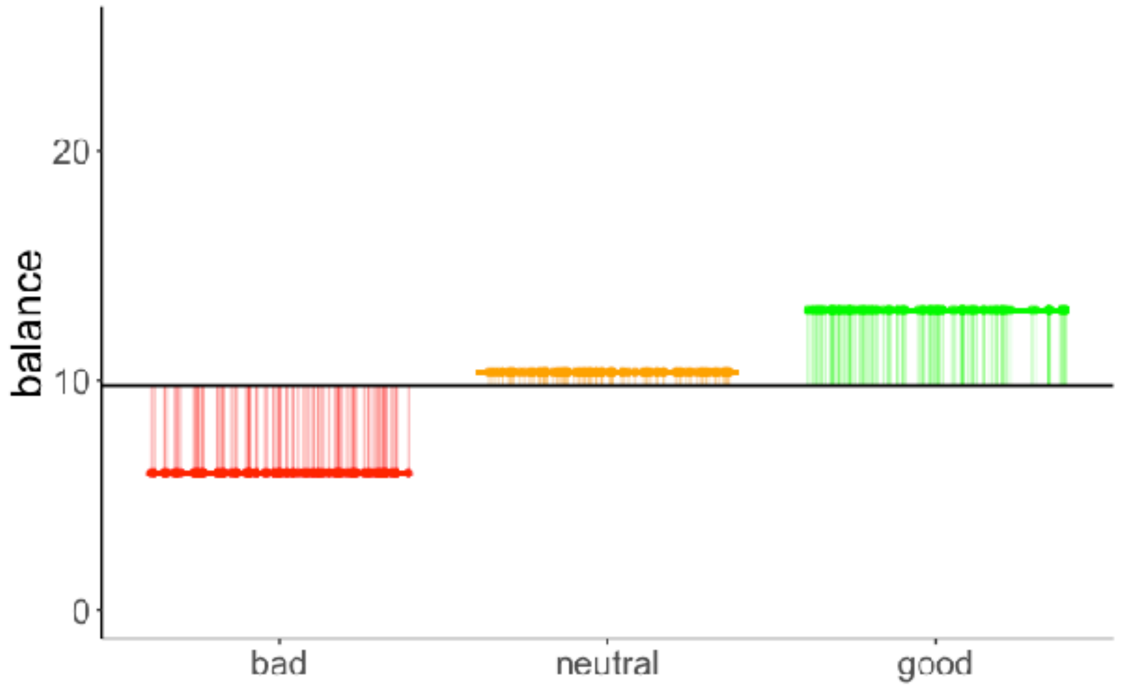
```
Analysis of Variance Table

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
```

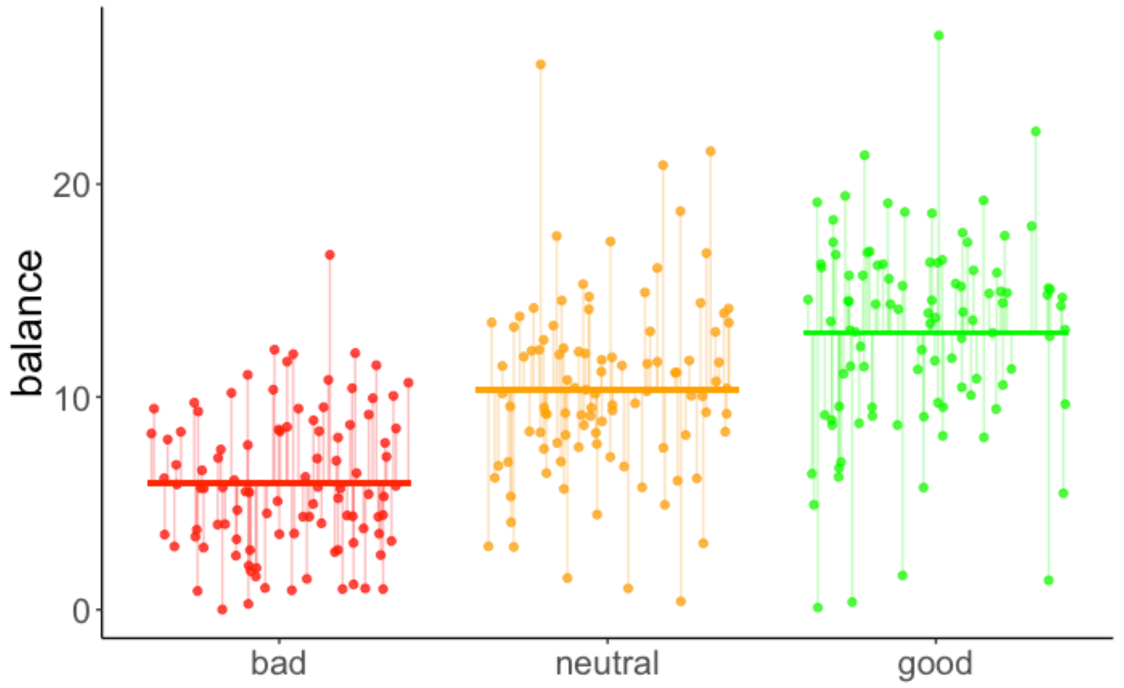
One-way ANOVA

Variance decomposition

Total variance	Model variance	Residual variance





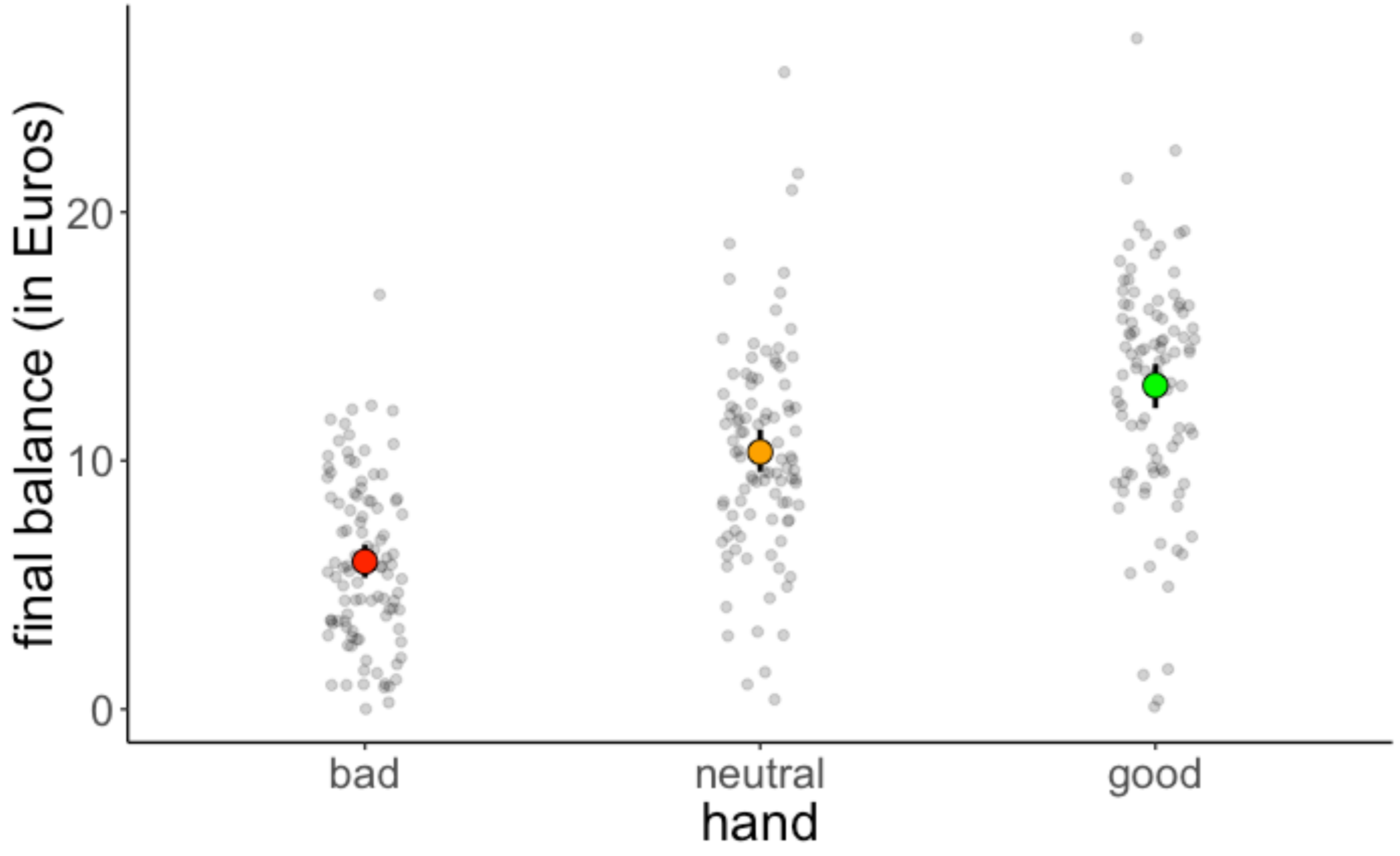


SS_{total} SS_{model} $SS_{residual}$

variance_total	variance_model	variance_residual
7580	2559	5021

`Data = Model + Error`

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), $F(2, 297) = 75.703$, $p < .001$.

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?

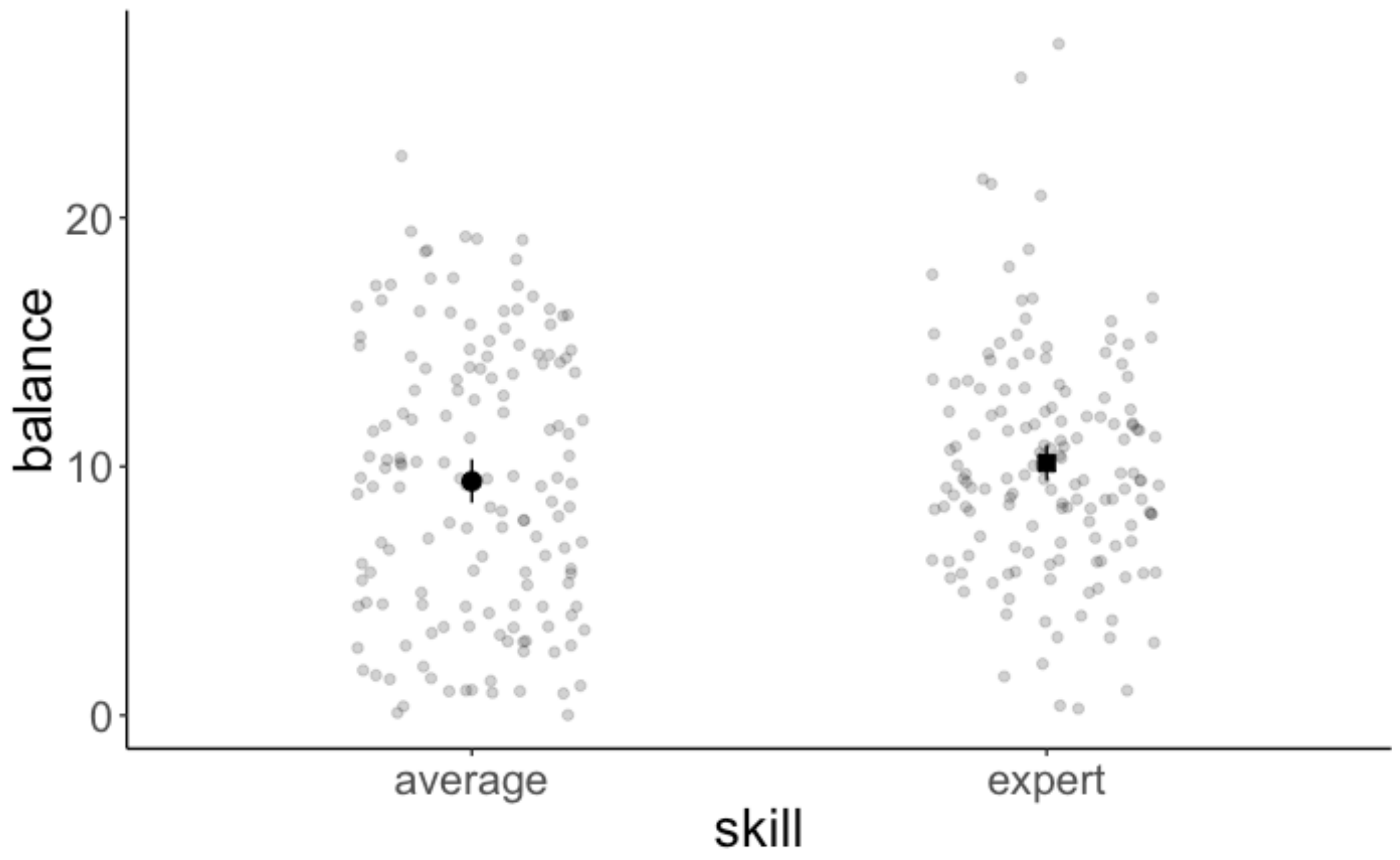
One testing whether skill level affects the final balance, and one testing whether quality of cards affects the final balance?

Interested in interactions!

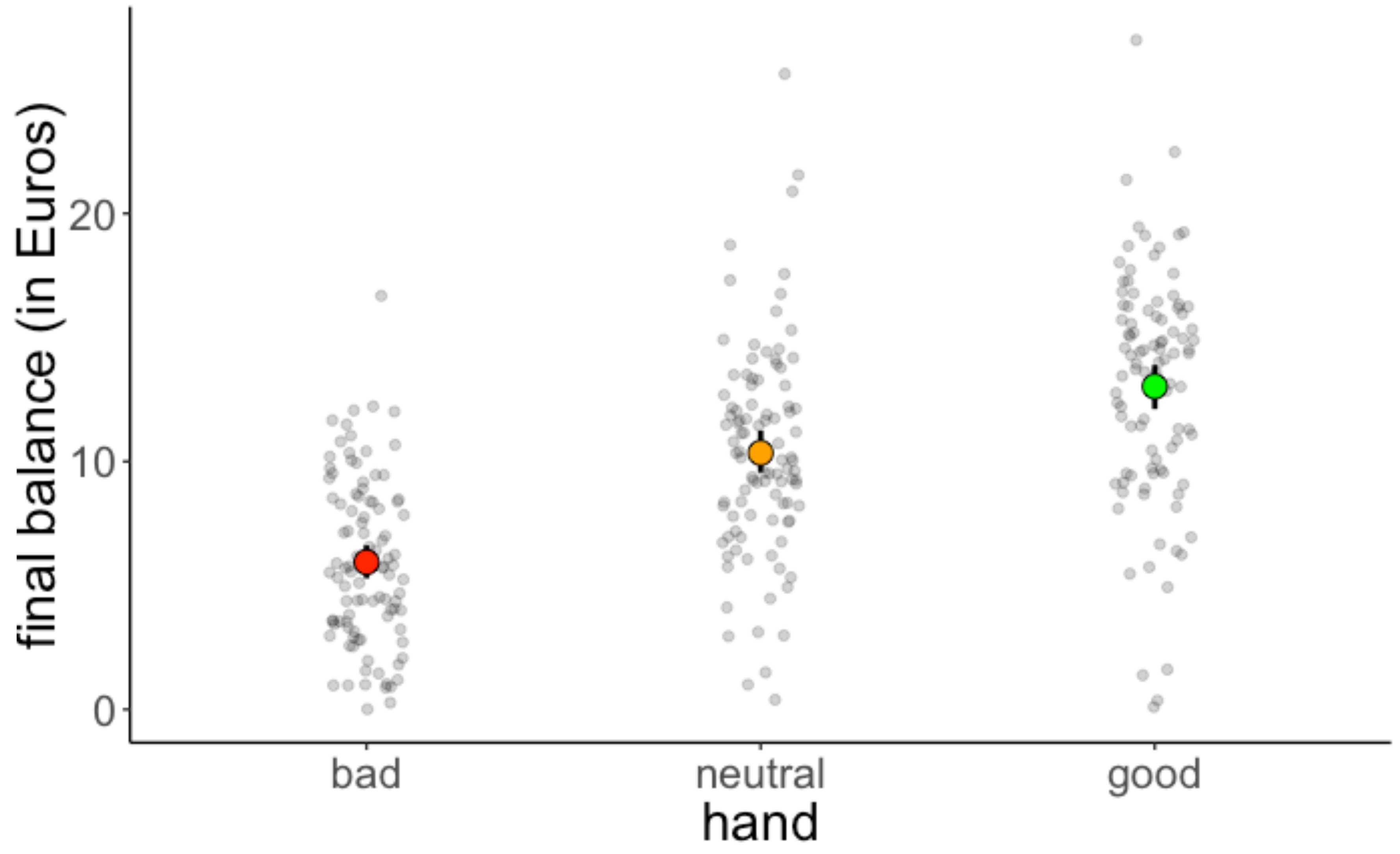
Does the effect of one variable depend on the other?

Visualize the data

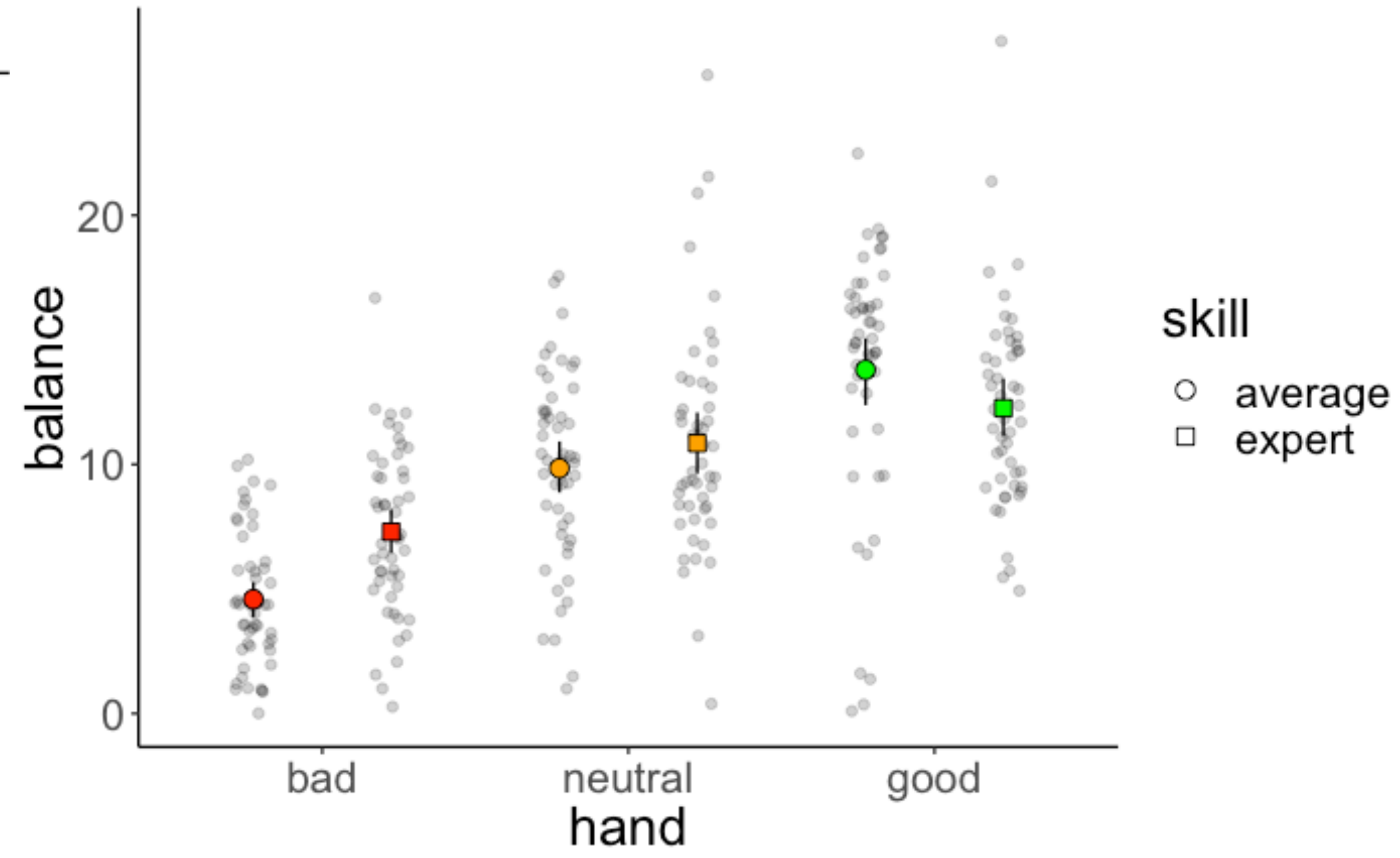
Skill



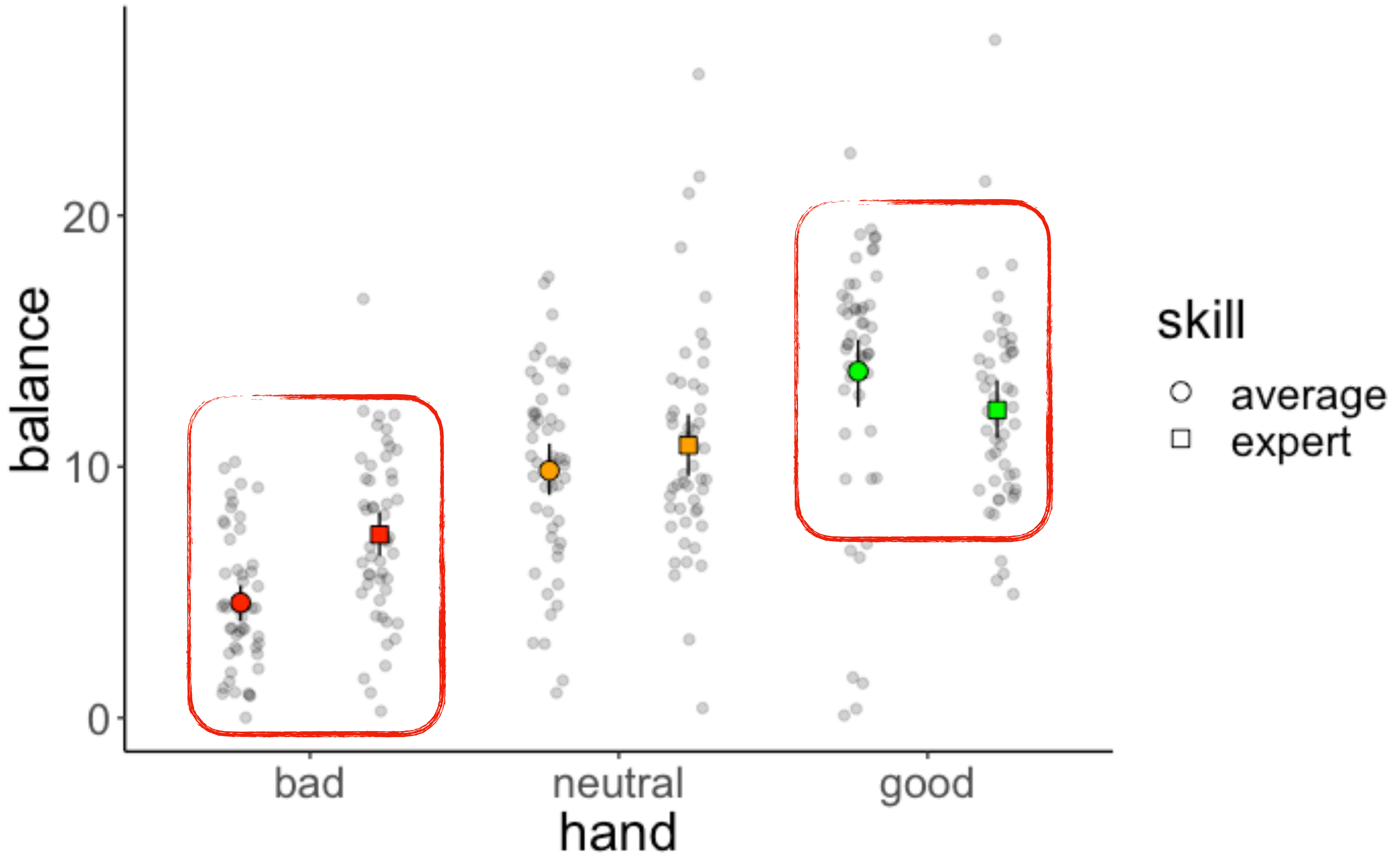
Quality of hand



Quality of hand & Skill



Visualize the data

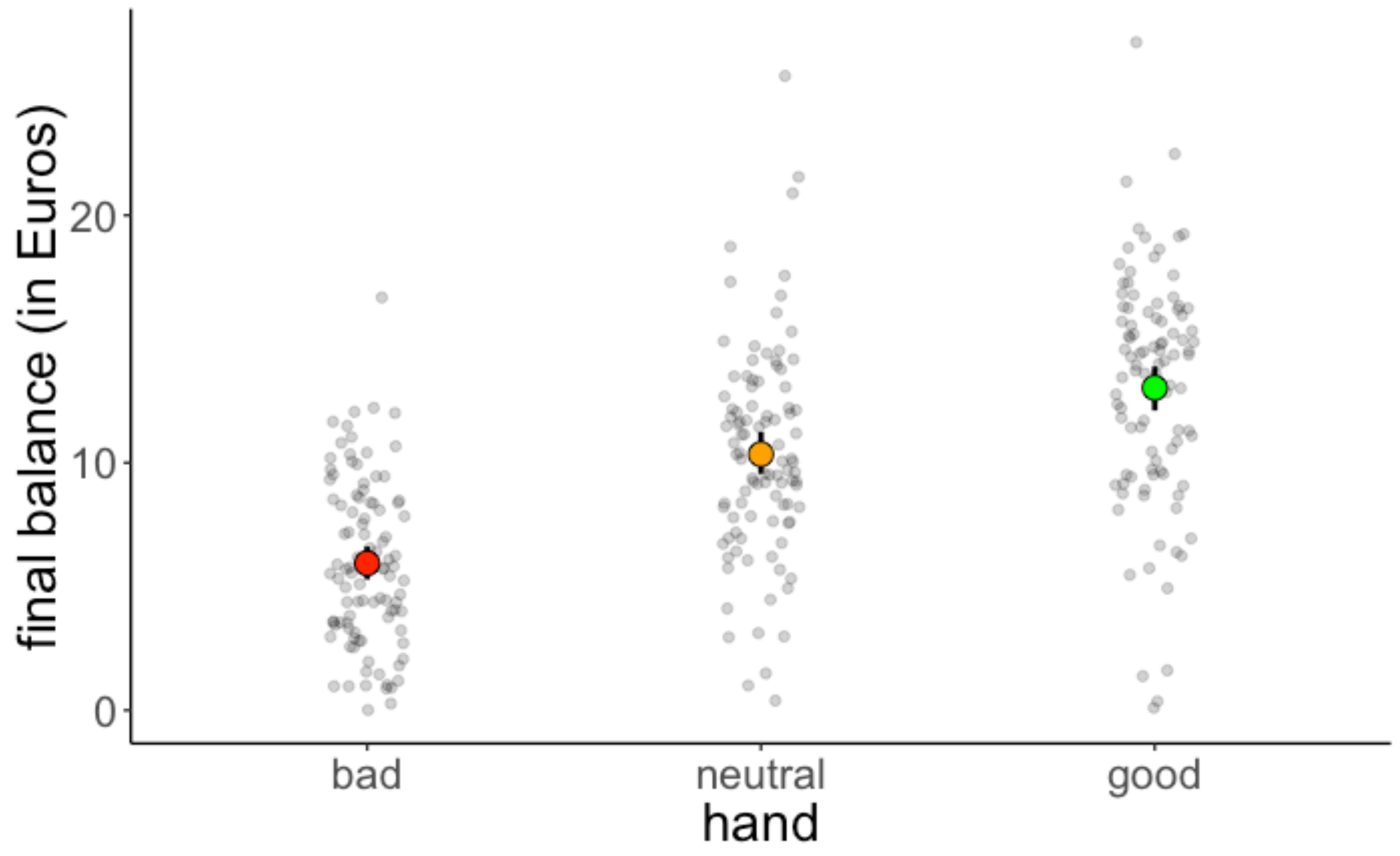


Analysis of variance

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

```
Analysis of Variance Table

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

main effect of hand



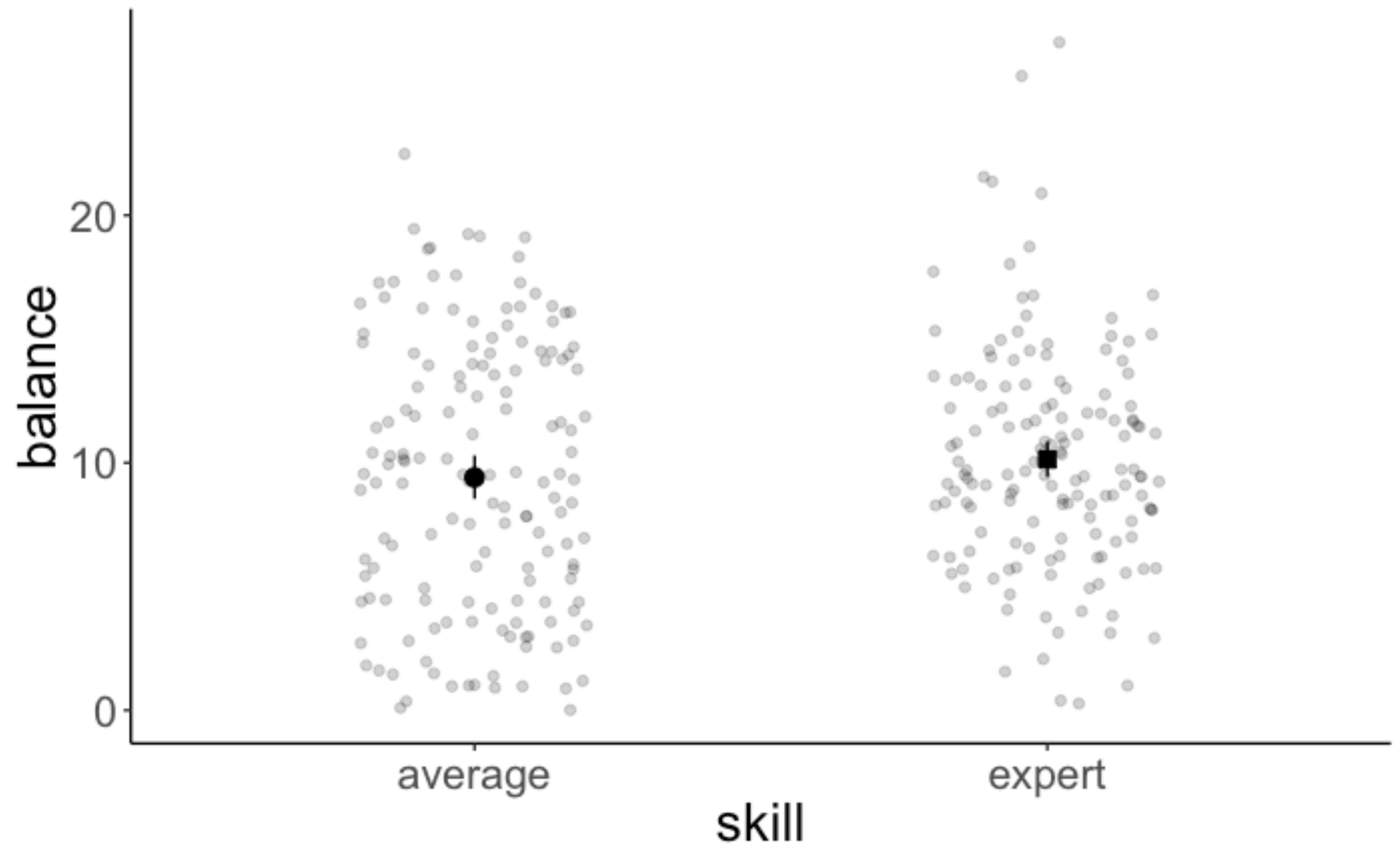
Analysis of variance

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

```
Analysis of Variance Table

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

main effect of `hand`

no main effect of `skill`



Analysis of variance

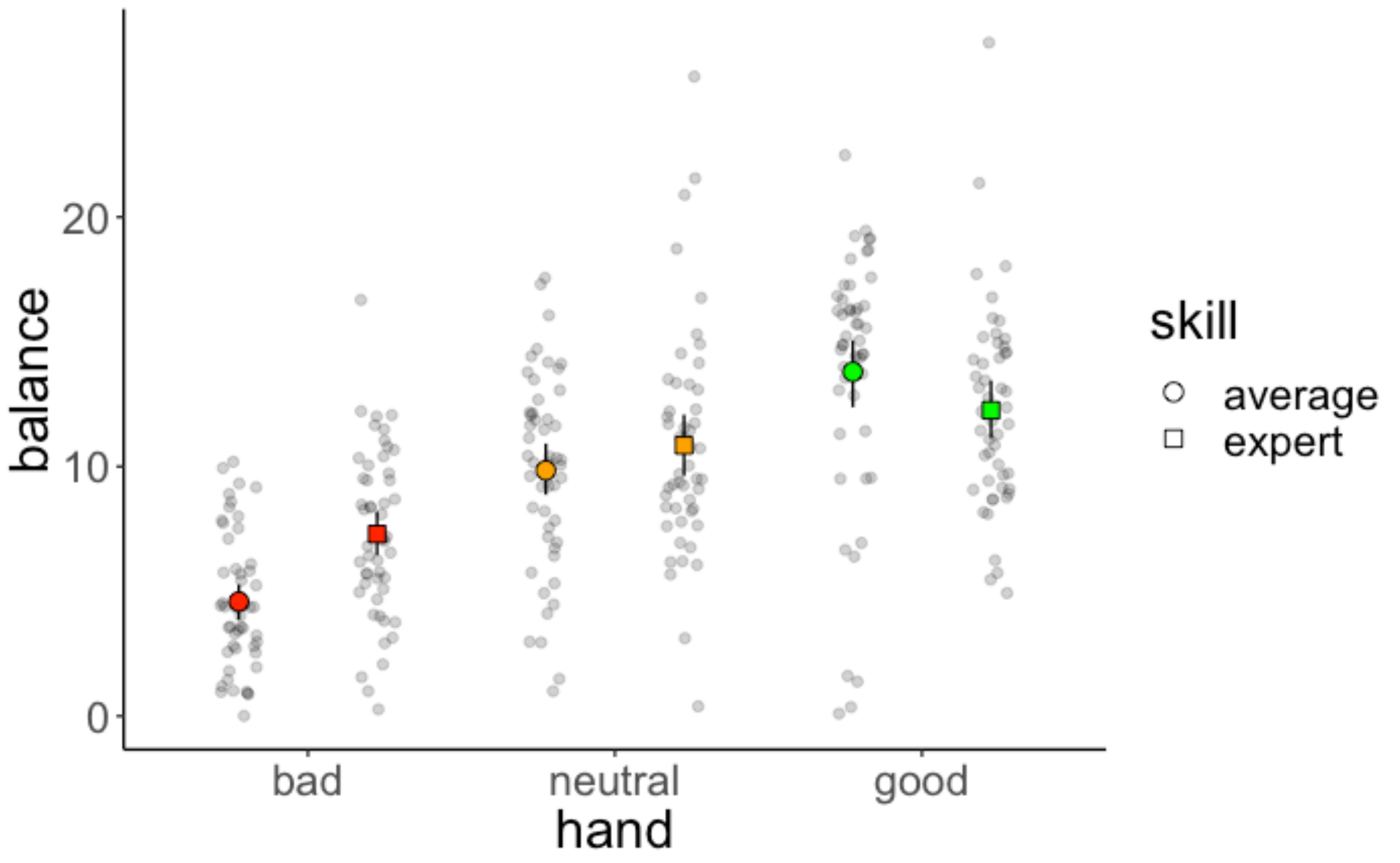
```
lm(formula = balance ~ hand * skill, data = df.poker) %>%
  anova()
```

```
Analysis of Variance Table

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

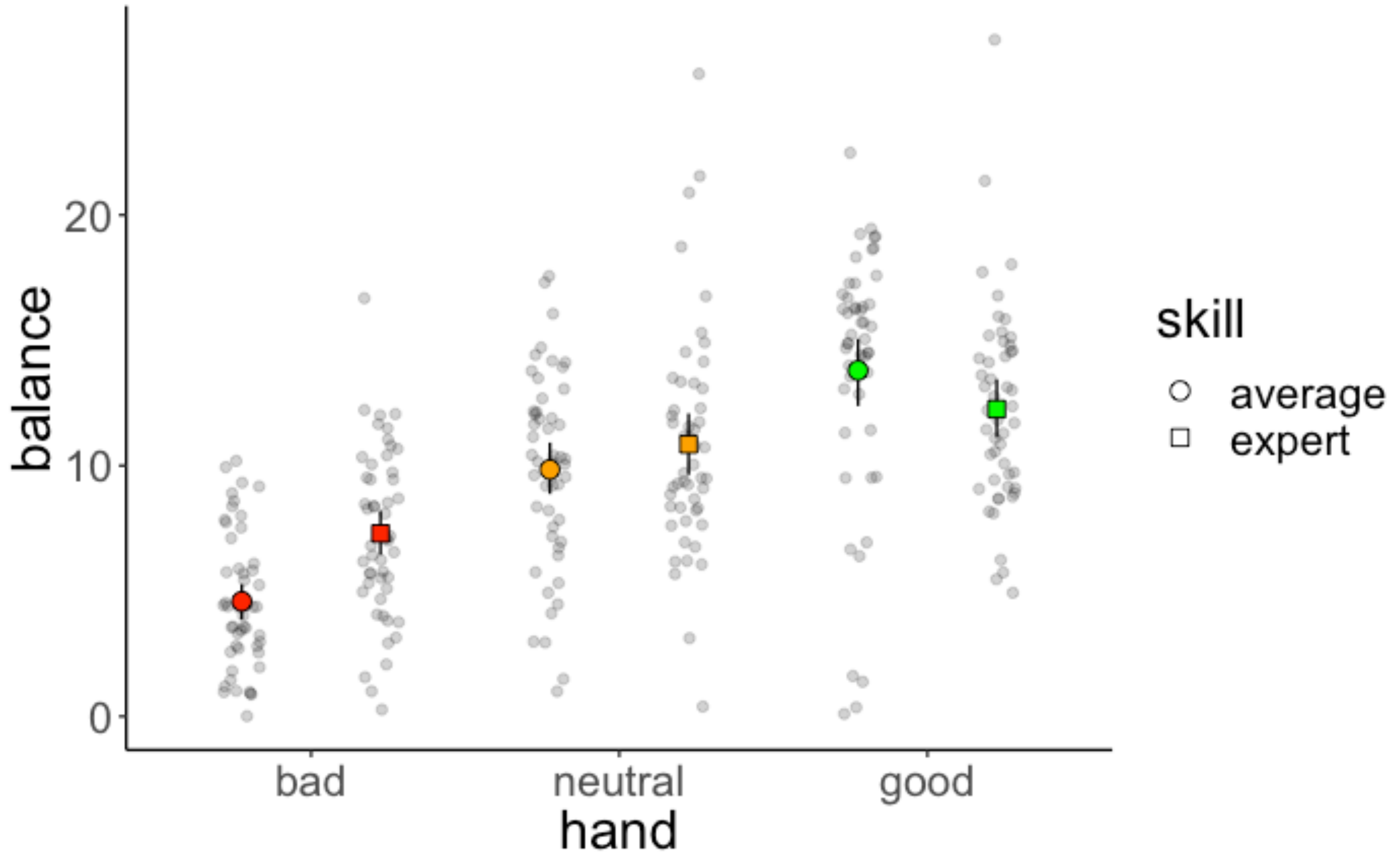
main effect of `hand`

no main effect of `skill`

interaction between `hand` and `skill`



Reporting the results



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

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

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

Interpreting parameters

Parameter interpretation

```r
lm(formula = balance ~ 1 + hand * skill, data = df.poker) %>%
  summary()
```

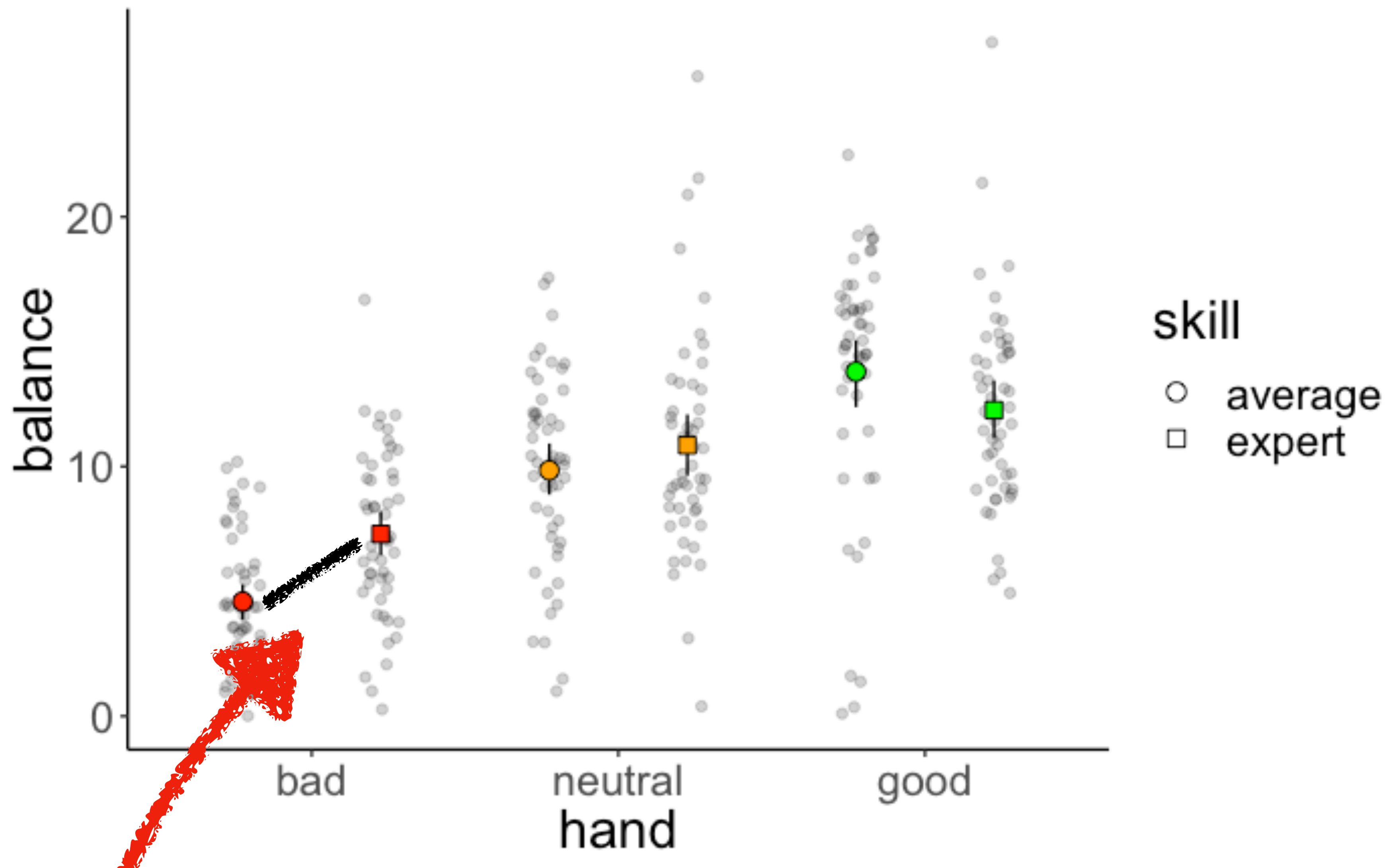
```
Call:
lm(formula = balance ~ 1 + hand * skill, data = df.poker)

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 ***
---
Sign                                                         1

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

acute danger of misinterpretation!!

~~there was a significant effect of skill~~

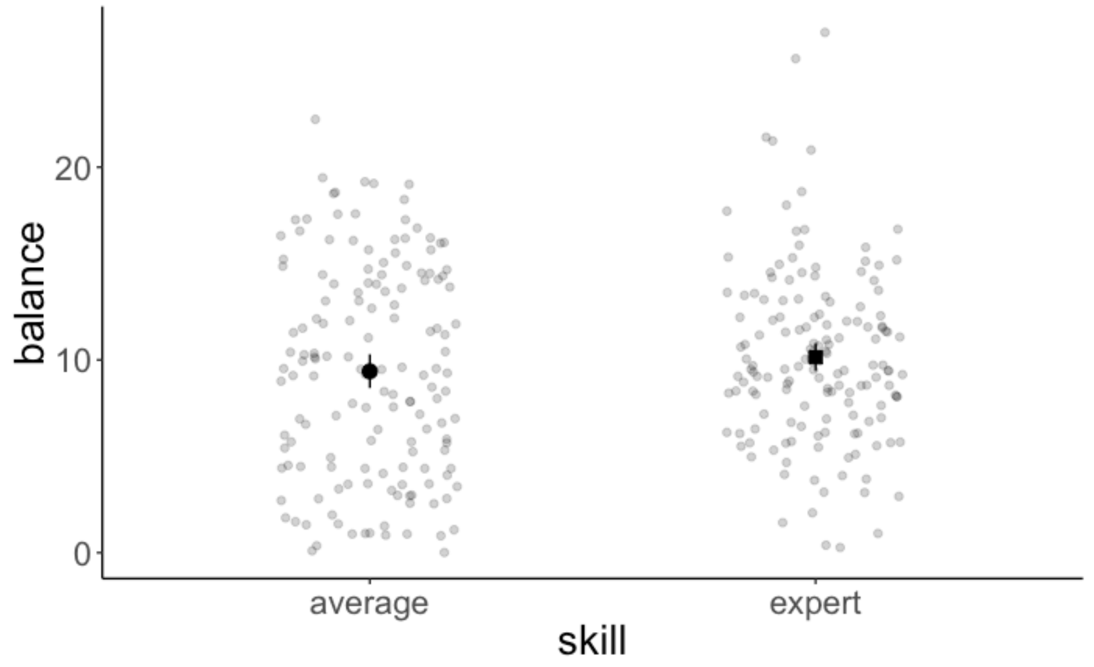
Parameter interpretation

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




```
Analysis of Variance Table

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

there was no main effect of skill!




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

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

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

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

is this difference significantly different from 0?

hand	average	expert
bad	4.59	7.3

Different effect terms

- **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

```r
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

```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
```

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()
```

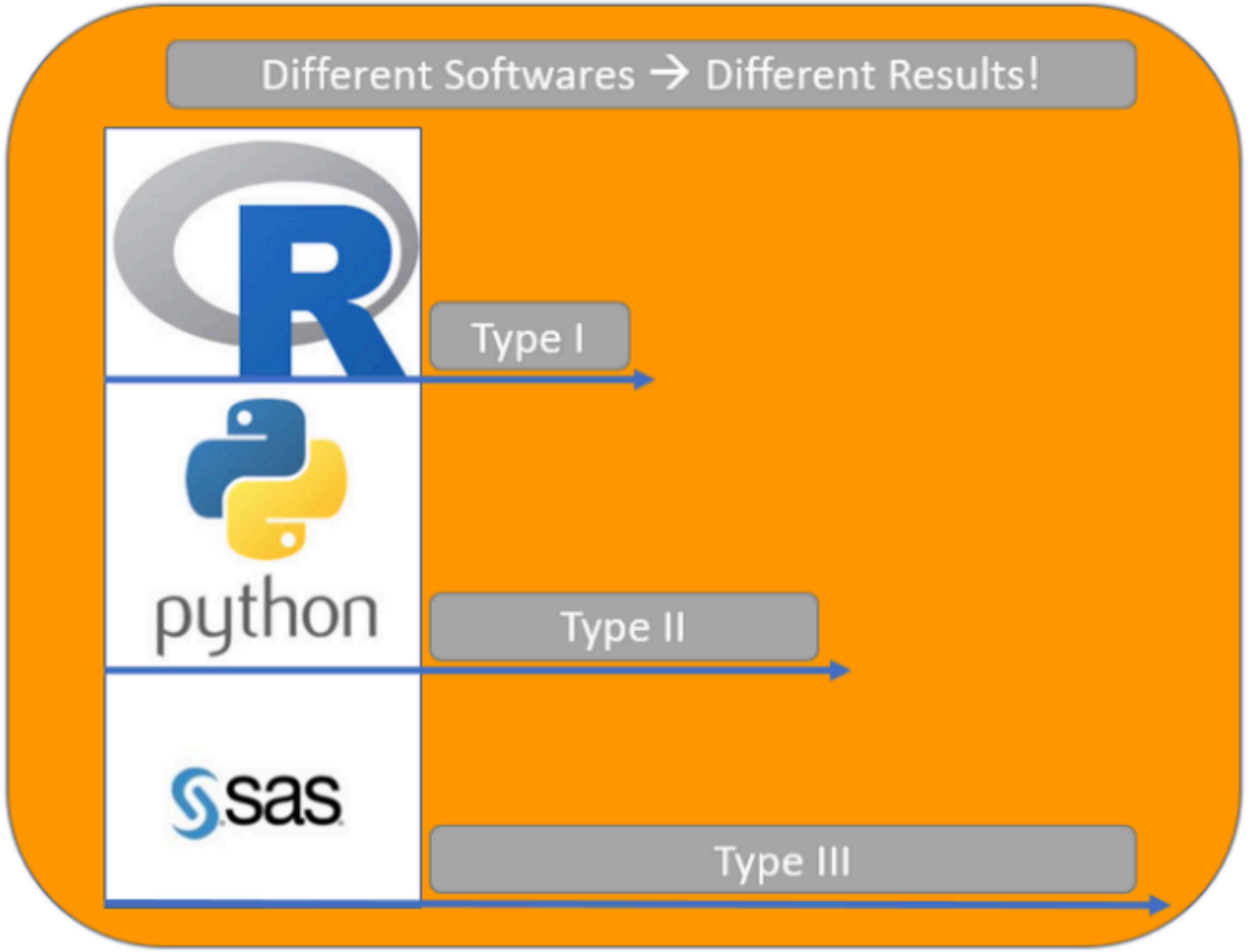
```
Analysis of Variance Table

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

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!

Default sums of squares ...



Default Types of Sums of Squares for different programming languages

not great for reproducibility ...

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, SAS, Stata etc use it)

Route I: Using "afex"

```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
```

Route II: Using "emmeans"

preferred route!!

```
library("emmeans")

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

very handy function

```
model term df1 df2 F.ratio p.value
 hand          2 284  68.973  <.0001
 skill         1 284   2.954  0.0868
 hand:skill    2 284   7.440  0.0007
```

Same same ...

Route I: Using "afex"

```
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
```

Route II: Using "emmeans"

preferred route!!

```
library("emmeans")

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

very handy function

```
model term df1 df2 F.ratio p.value
 hand         2 284  68.973  <.0001
 skill        1 284   2.954  0.0868
 hand:skill   2 284   7.440  0.0007
```

... but different

can come apart when we deal with repeated observations, but we'll deal with that later!

Unbalanced design

- 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).

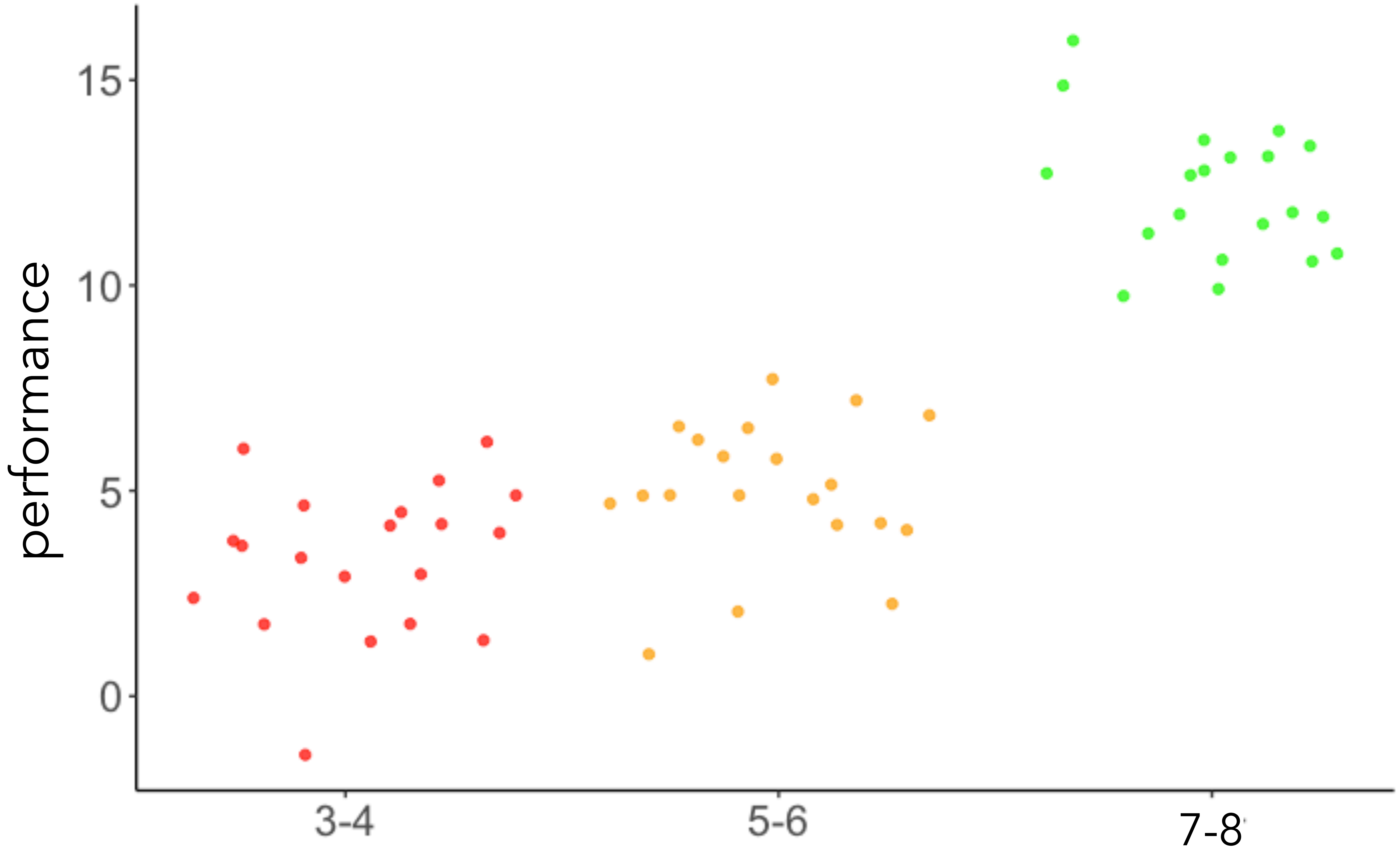
`joint_tests()` is your friend!

Linear contrasts

Testing (more) specific hypotheses with linear contrasts

Contrasts

Does performance increase with age?



Data from a hypothetical developmental study

Does performance increase with age?

ANOVA

Does performance differ between age groups?

post-hoc tests

3-4 vs. 5-6

5-6 vs. 7-8

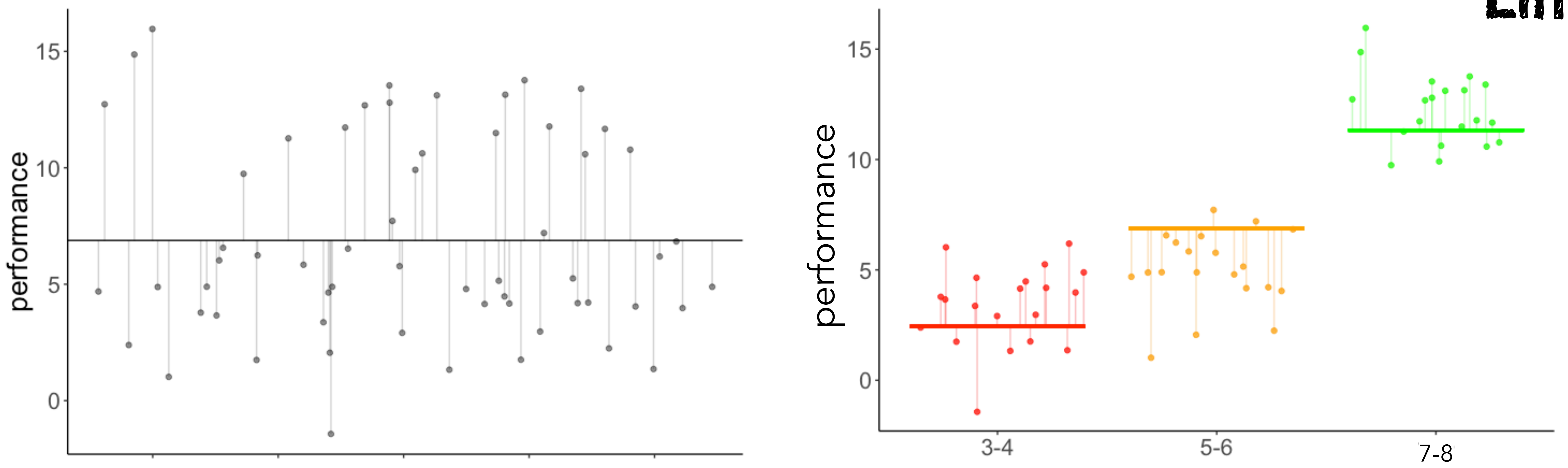
Is there are more direct way of asking this question with a statistical model?

Contrasts

Does performance increase with age?

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

Compact model **Augmented model** **Linear contrast**



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.

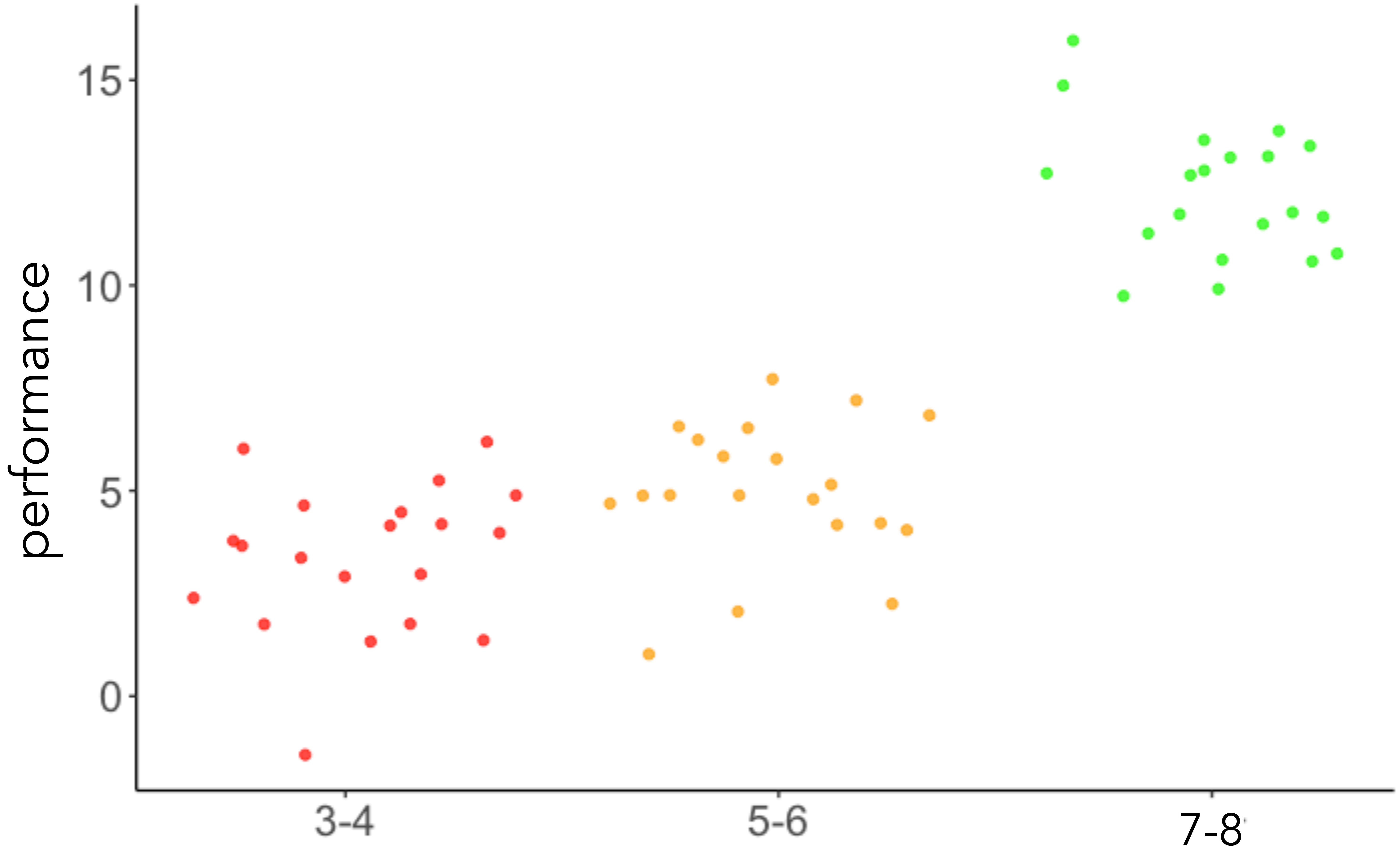
`emmeans` works with `lm`, `glm`, and the Bayesian friends in `brms` and `rstanarm`, so the process is applicable no matter the tool.

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

```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)

# 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))
```

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

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")
```

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

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
```

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(-1, -1, 2),
                 three_vs_five = c(-1, 1, 0))

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

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
```

Defining contrasts

- groups that we don't want to include in the comparison get a 0
- groups that we want to compare with one another should sum to 0
- this also means that all the contrasts together should sum to 0

Example:

```
contrasts = list(young_vs_old = c(-1, -1, 2),
                 three_vs_five = c(-1, 1, 0))
```

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?

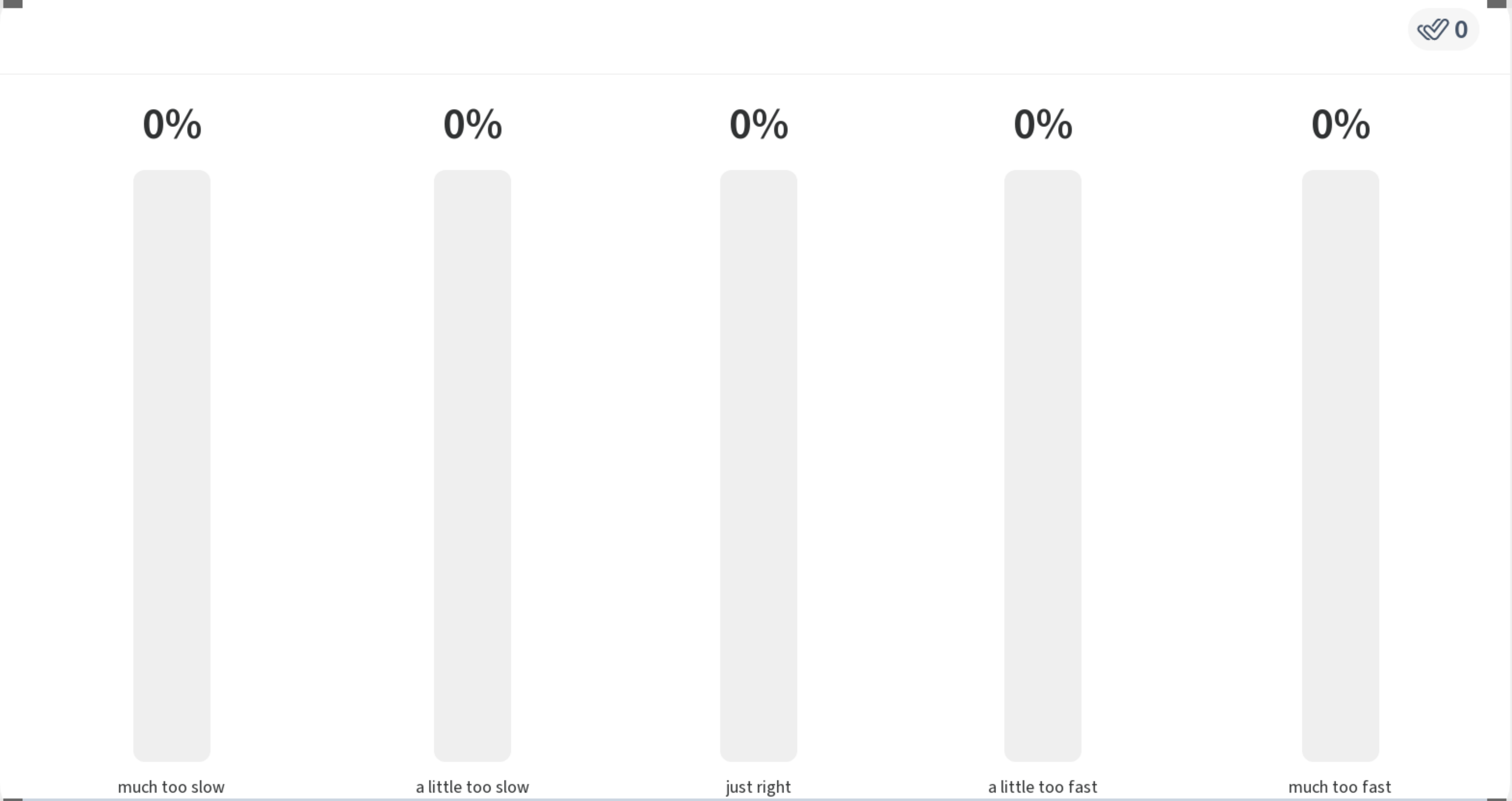
Plan for today

- Quick recap
- Interaction
- `lm()` output
- 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

0

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**


0
0%
0%
0%
0%
0%
much too slow
a little too slow
just right
a little too fast
much too fast

Thank you!