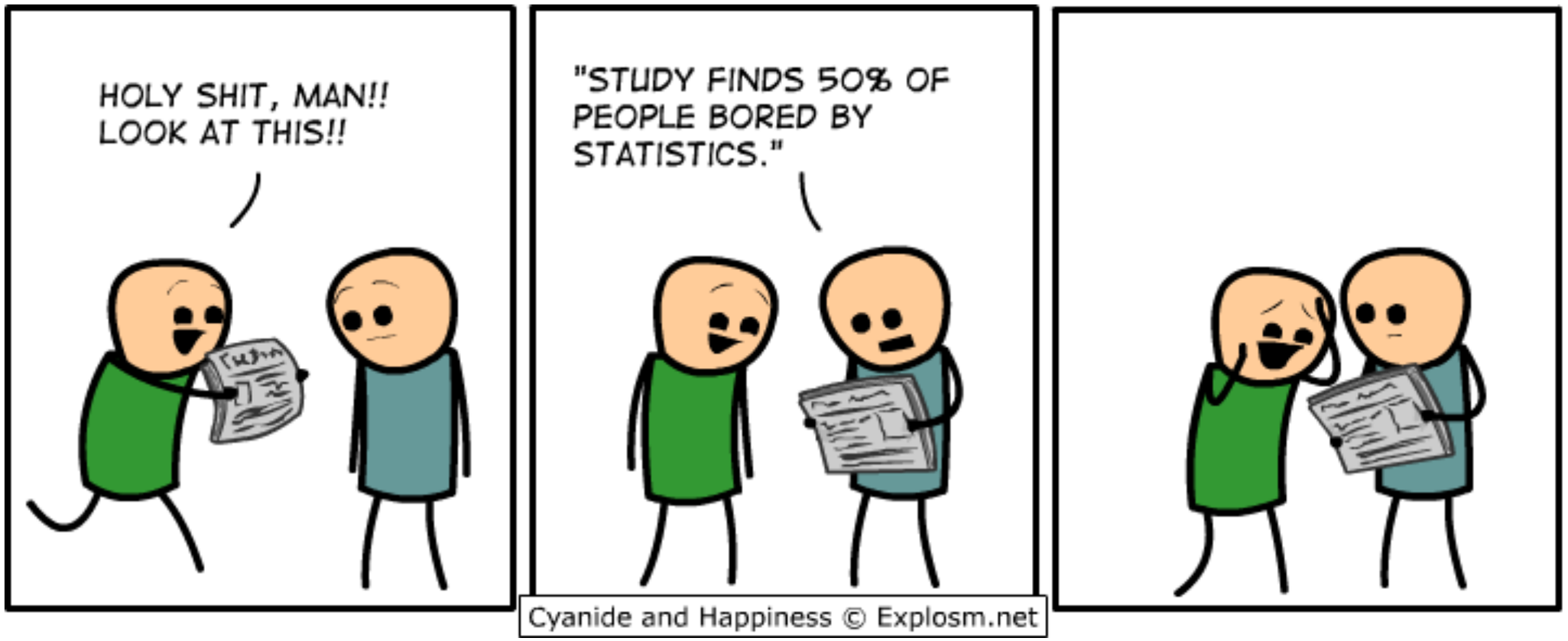
# Linear model 4




02/06/2019

# Logistics

# Start time

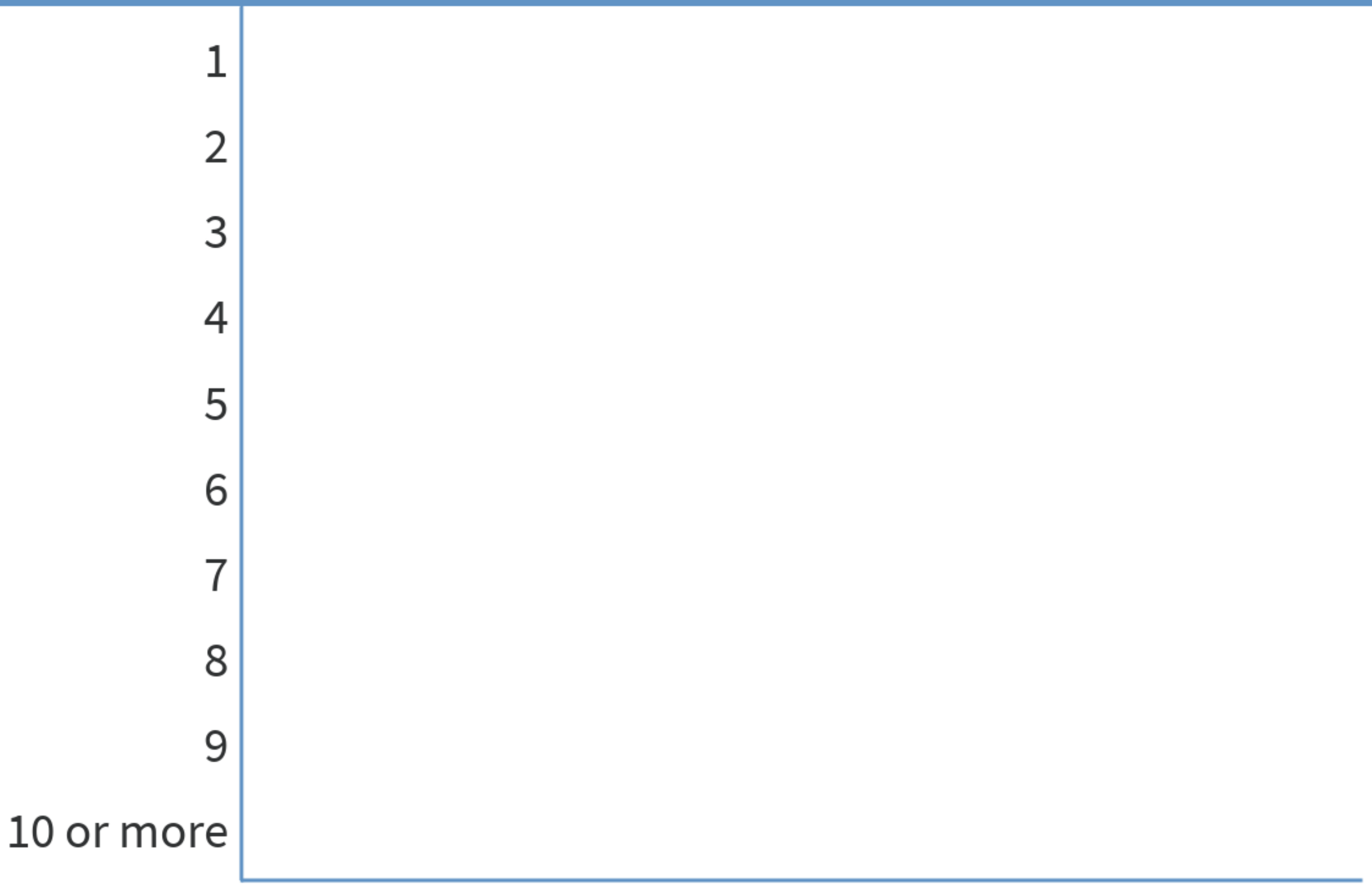
Start time

**Class now starts at 10:30 am**

# Homework

# How do you feel about the data camp class "Correlation and Regression" ?

# How many hours did it take you to complete Homework 3?



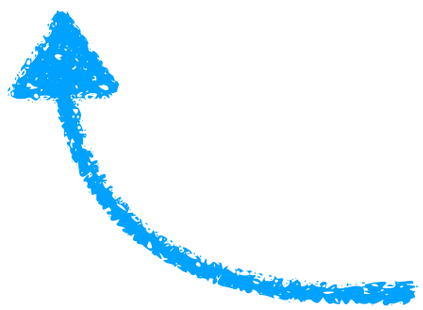
# Midterm

# Midterm

Will be released on Monday 11th at noon (after class), and will be due on **Wednesday 13th at noon**.

There will be no class on Wednesday!

Sections next week will discuss the midterm solutions.



**You should go!**

No homework this week

# Code scripts

# R code scripts ...

I've been running behind but will update shortly ...

# Plan for today

- ANOVA interpretation
- Contrasts
  - Planned comparisons
- Coding schemes
  - Dummy Coding
  - Effect coding
- Linear model assumptions
  - how to check them
  - what to do if they are violated

# ANOVA interpretation

# One-way ANOVA

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

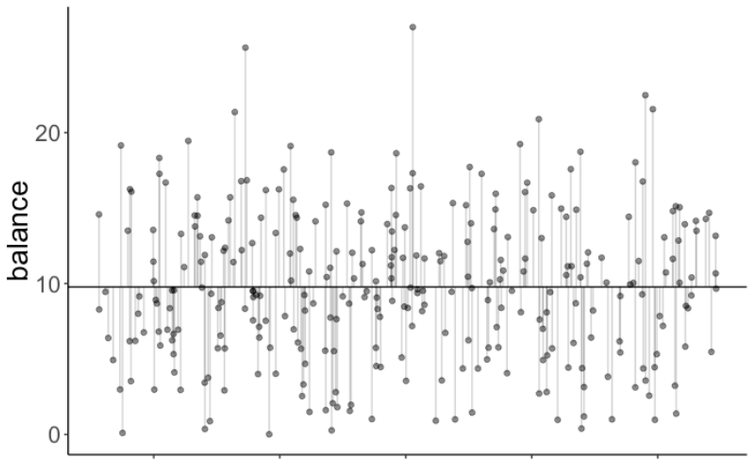
```
Analysis of Variance Table

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?**

# One-way ANOVA

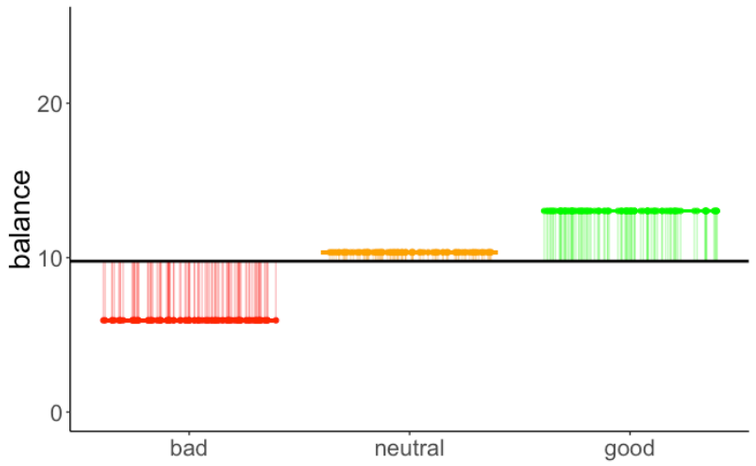
## Variance decomposition

### Total variance



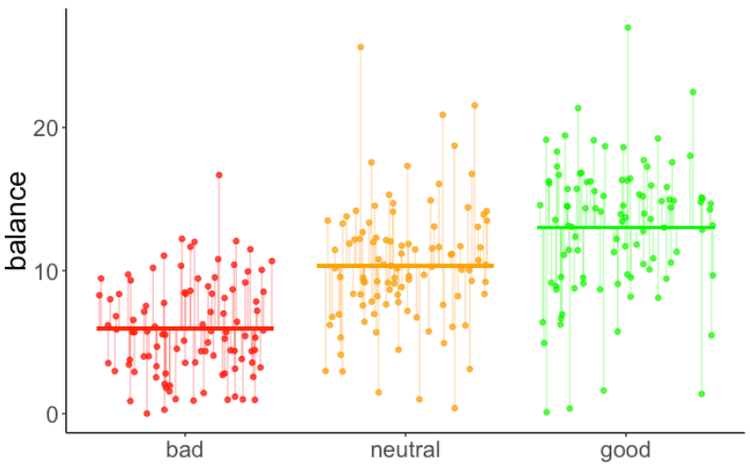

$SS_{total}$

### Model variance




$SS_{model}$

### Residual variance




$SS_{residual}$

| variance_total | variance_model | variance_residual |
|---|---|---|
| 7580 | 2559 | 5021 |

# One-way ANOVA

```
1 df.poker %>%
2   mutate(mean_grand = mean(balance)) %>%
3   group_by(hand) %>%
4   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
```

# Two-way ANOVA

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

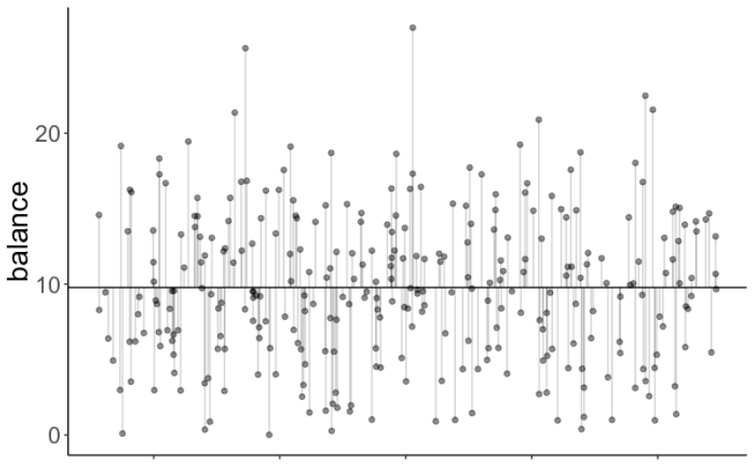
```
Analysis of Variance Table

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

**What do these mean?**

# Two-way ANOVA

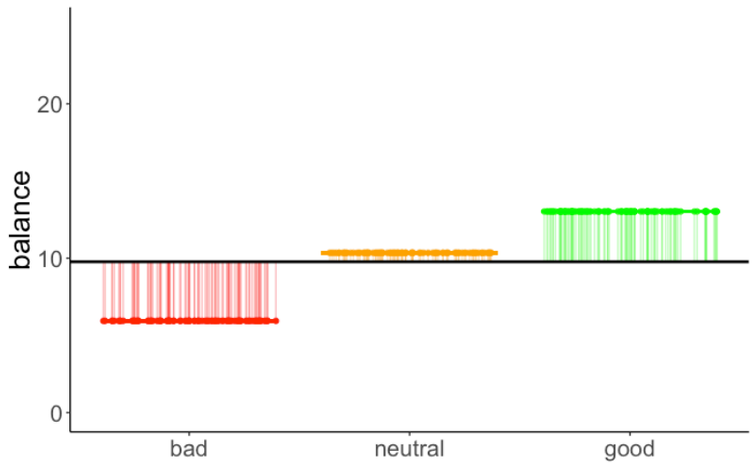
## Variance decomposition

### Total variance



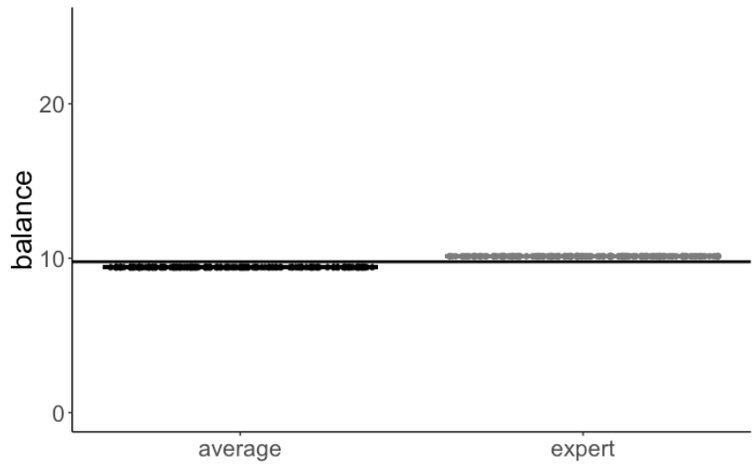

$SS_{total}$

### Hand variance




$SS_{hand}$

### Skill variance




$SS_{skill}$

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

# Two-way ANOVA

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

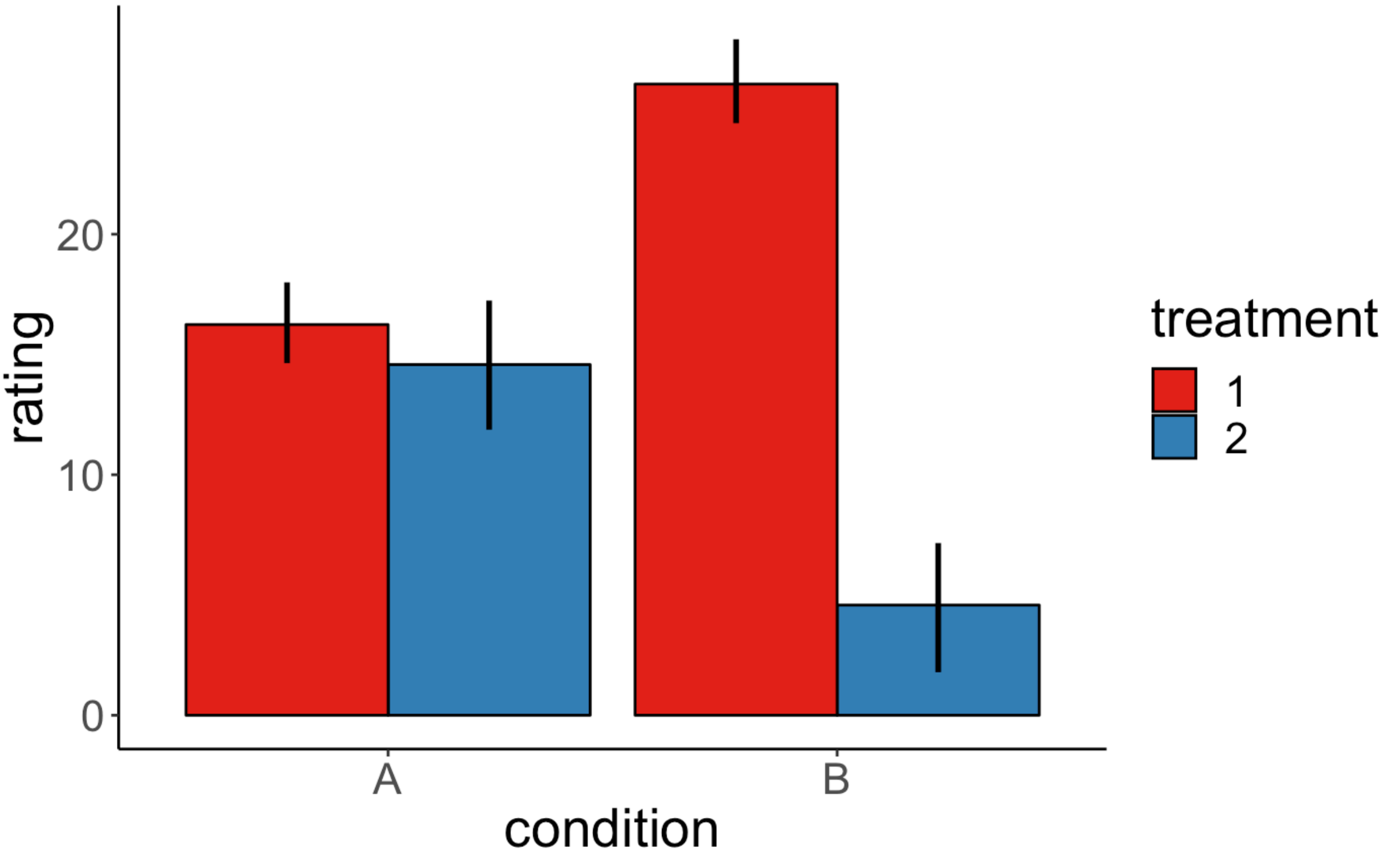
| participant | skill | hand | balance | mean_grand | mean_skill | mean_hand |
|---|---|---|---|---|---|---|
| 1 | expert | bad | 4.00 | 9.77 | 10.13 | 5.94 |
| 2 | expert | bad | 5.55 | 9.77 | 10.13 | 5.94 |
| 51 | expert | neutral | 11.74 | 9.77 | 10.13 | 10.35 |
| 52 | expert | neutral | 10.04 | 9.77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

| variance_total | variance_skill | variance_hand | variance_residual |
|---|---|---|---|
| 7580 | 39 | 2559 | 4981 |

```
Analysis of Variance Table

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

# Simulating linear model data

**How can I make my own?**



# Simulating linear model data

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

# set parameters
nsamples = 30

b0 = 15
b1 = 10 # main effect of condition
b2 = 0 # main effect of treatment
b3 = -20 # interaction effect
sd = 5

```

df.data

| condition | treatment | rating |
|---|---|---|
| A | 1 | 11.87 |
| A | 2 | 15.92 |
| A | 1 | 10.82 |
| A | 2 | 22.98 |
| B | 1 | 21.87 |
| B | 2 | 5.92 |
| B | 1 | 20.82 |
| B | 2 | 12.98 |

# Inferring the parameters

$$\text{rating}_i = \beta_0 + \beta_1 \cdot \text{condition}_i + \beta_2 \cdot \text{treatment}_i + \beta_3 \cdot (\text{condition}_i \times \text{treatment}_i) + \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(\mu = 0, \sigma^2 = 5)$$

```
b0 = 15
b1 = 10 # main effect of condition
b2 = 0 # main effect of treatment
b3 = -20 # interaction effect
sd = 5

rating = b0 + b1 * condition + b2 * treatment +
         (b3 * condition * treatment) +
         rnorm(nsamples, sd = sd)) %>%


# infer parameters
lm(formula = rating ~ 1 + condition + treatment +
               condition:treatment,
    data = df.data) %>%
   summary()
```

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 16.244      1.194  13.608  < 2e-16 ***
conditionB                  10.000      1.688   5.924 2.02e-07 ***
treatment2                  -1.662      1.688  -0.985    0.329
conditionB:treatment2      -20.000      2.387  -8.378 1.86e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# ANOVA

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

# Beware of unbalanced designs

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

```
Analysis of Variance Table

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

**flipped the order**

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

```
Analysis of Variance Table

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

# If you want to reproduce SPSS

- There are different kinds of ANOVAs, for which the sums of squares are calculated differently.
- This makes a difference when we have an unbalanced design (i.e. some of the cell sizes are unequal).
- We won't discuss how the different sums of squares are computed.
- If you want to reproduce what SPSS spits out, then do the following ...

# If you want to reproduce SPSS

```r
library("car")

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

**now the order doesn't matter ...**

```
Anova 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,
             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
```

# ANOVA

- We won't be discussing the different techniques to run ANOVAs in more detail.
- In practice, you'll almost never want to run an ANOVA. You'll want to ask more specific questions using planned contrasts.
- We will learn how to use linear mixed effects models (which are much more powerful and flexible than ANOVAs).
- Linear mixed effects models:
  - have no trouble with unbalanced designs
  - can capture dependencies in your data

# Contrasts

# 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
```

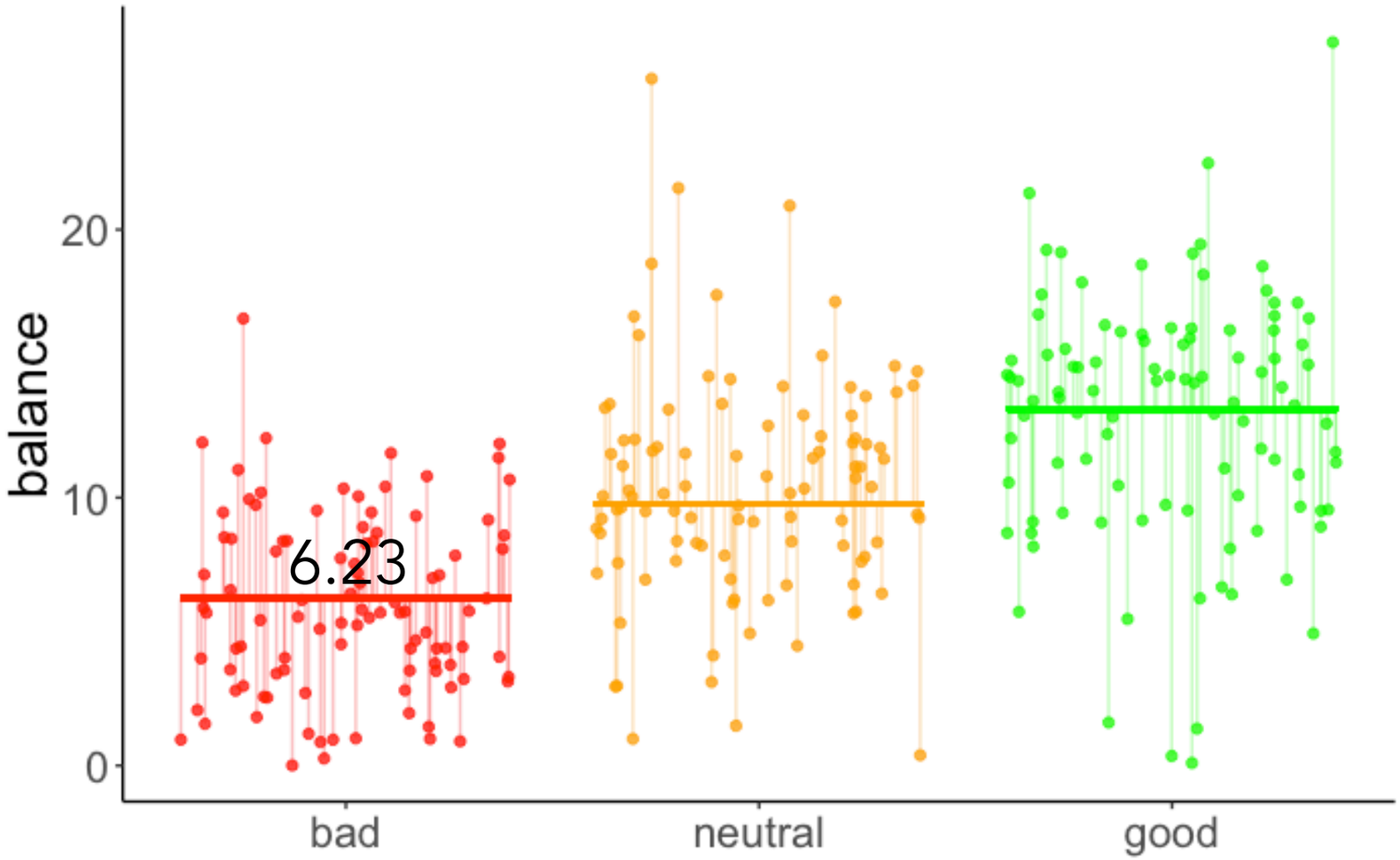
grand mean

significant contrast

# Contrasts

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

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



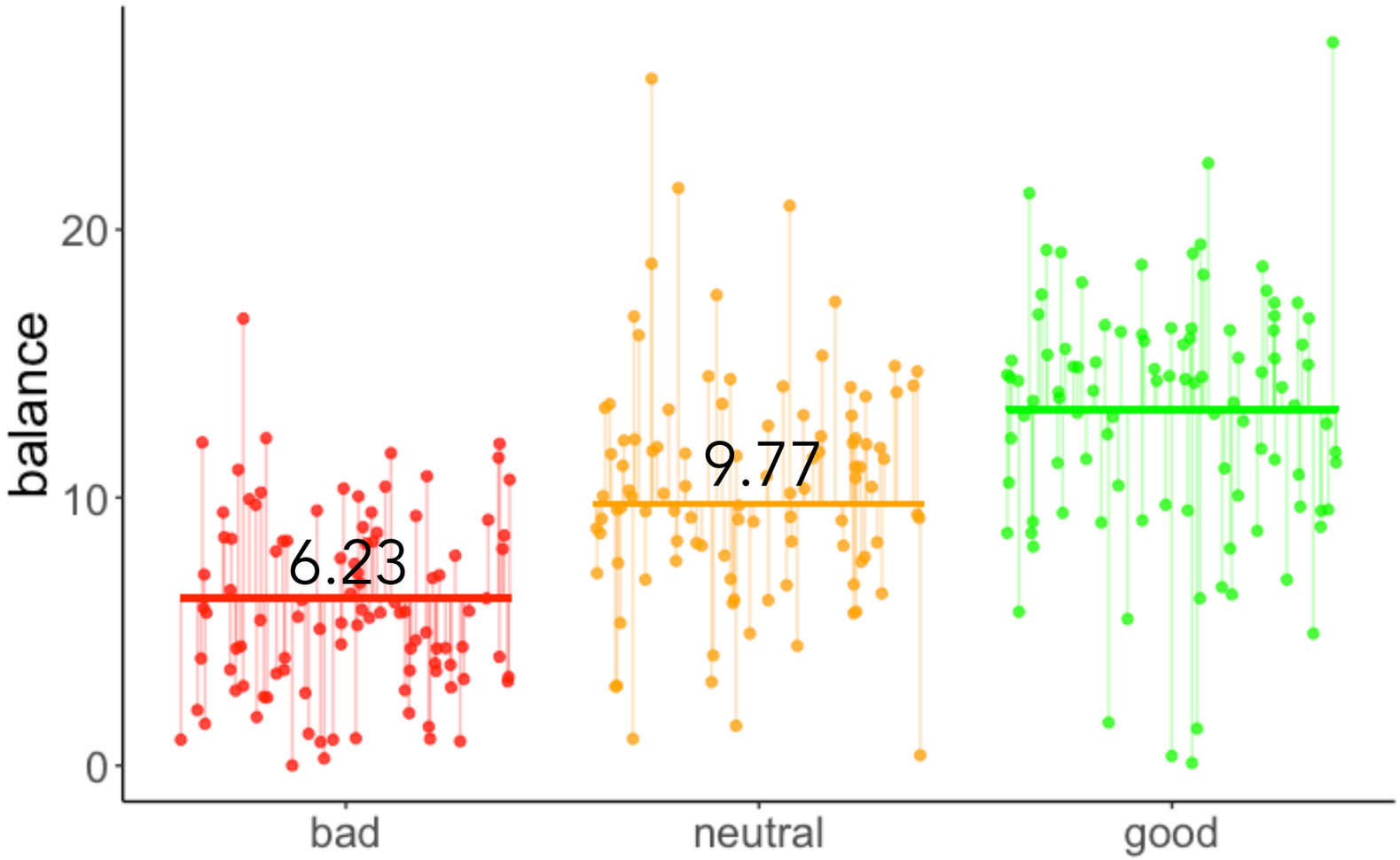

**if contrast == -1**

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

# Contrasts

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

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



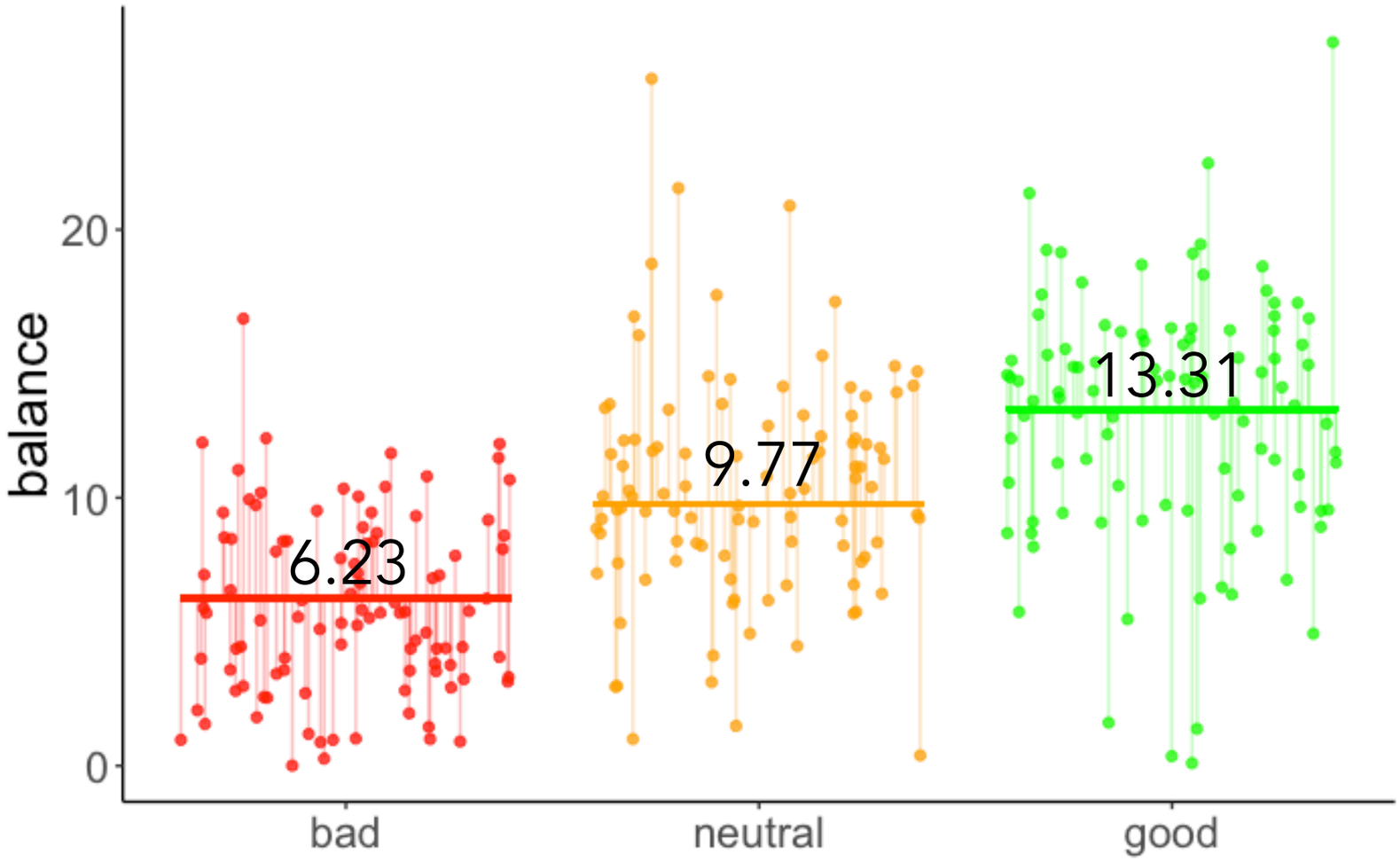

**if contrast == 0**

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

# Contrasts

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

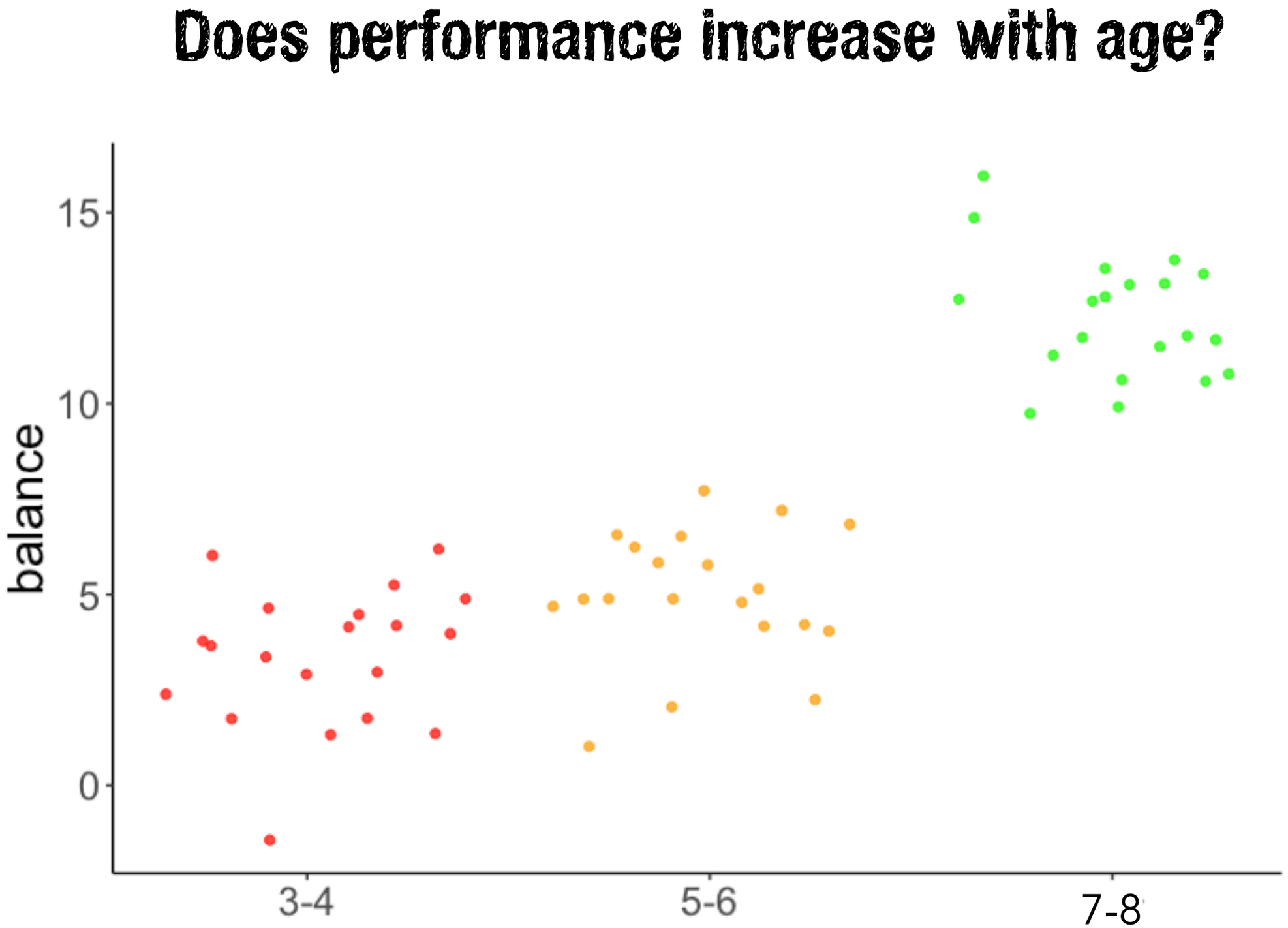
$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{hand\_contrast}$$




**if contrast == 1**

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

# Contrasts



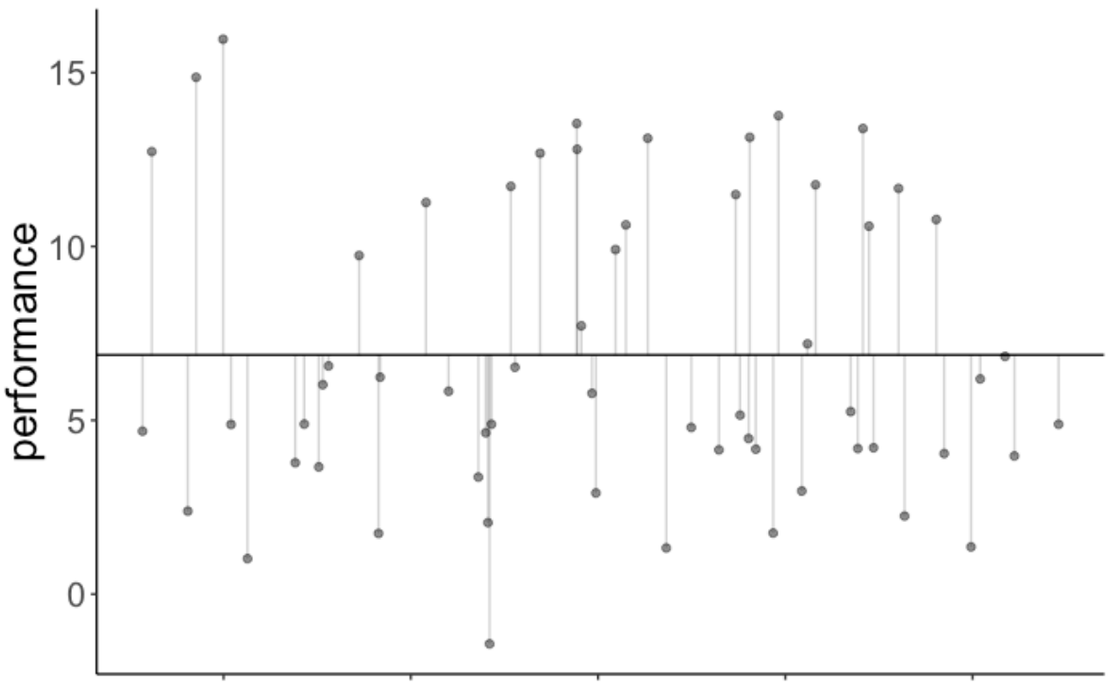

Data from a hypothetical developmental study

# Contrasts

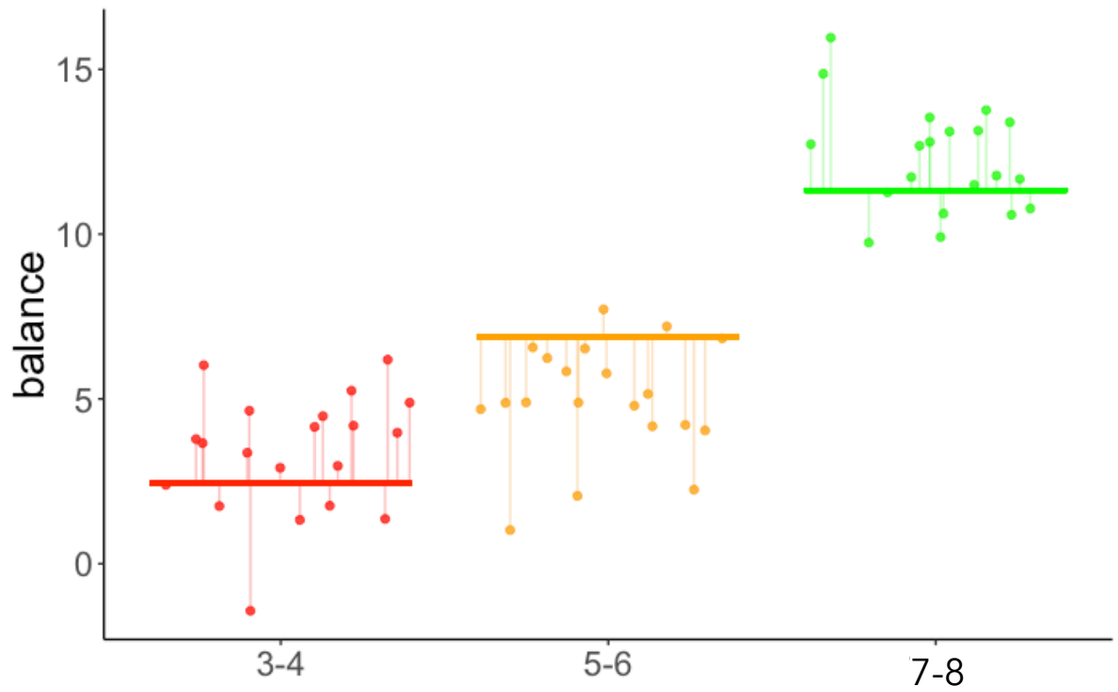
**Does performance increase with age?**

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

**Compact model**



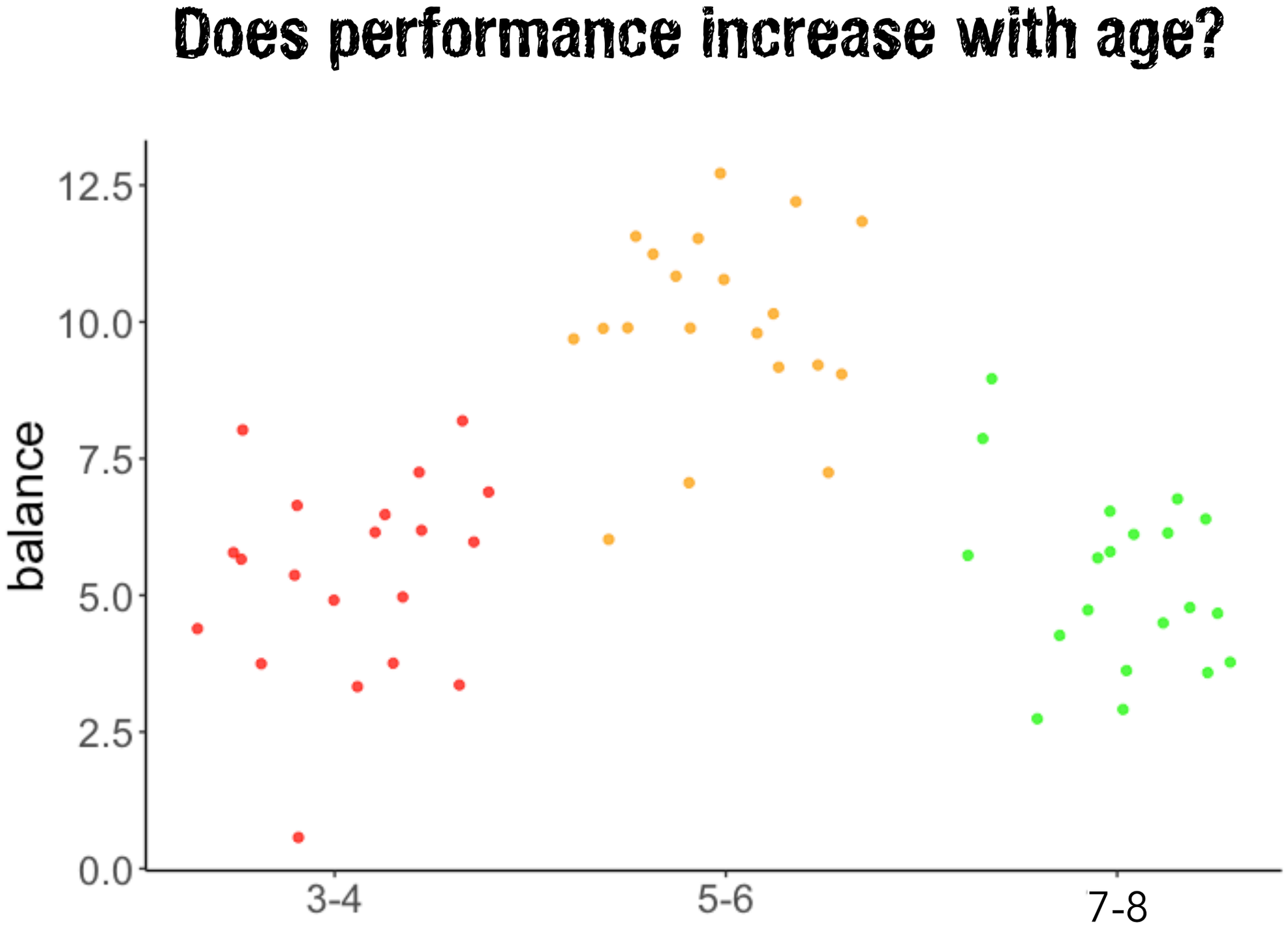

**Augmented model**




**Model comparison**

**p < .001**

# Contrasts



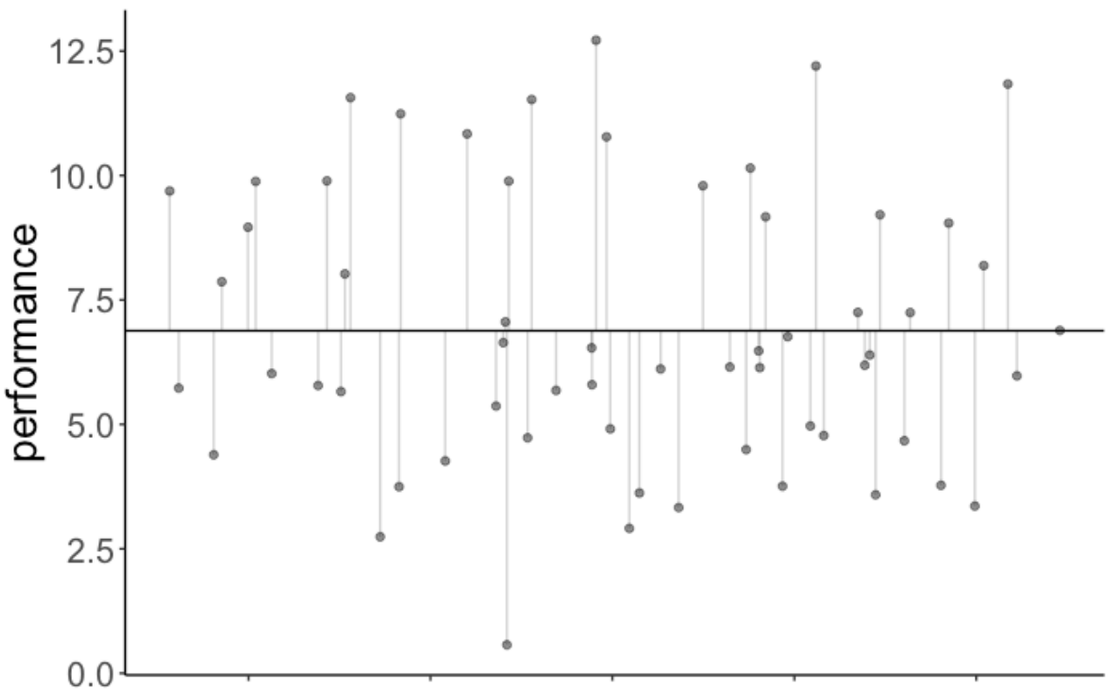

Data from another hypothetical developmental study

# Contrasts

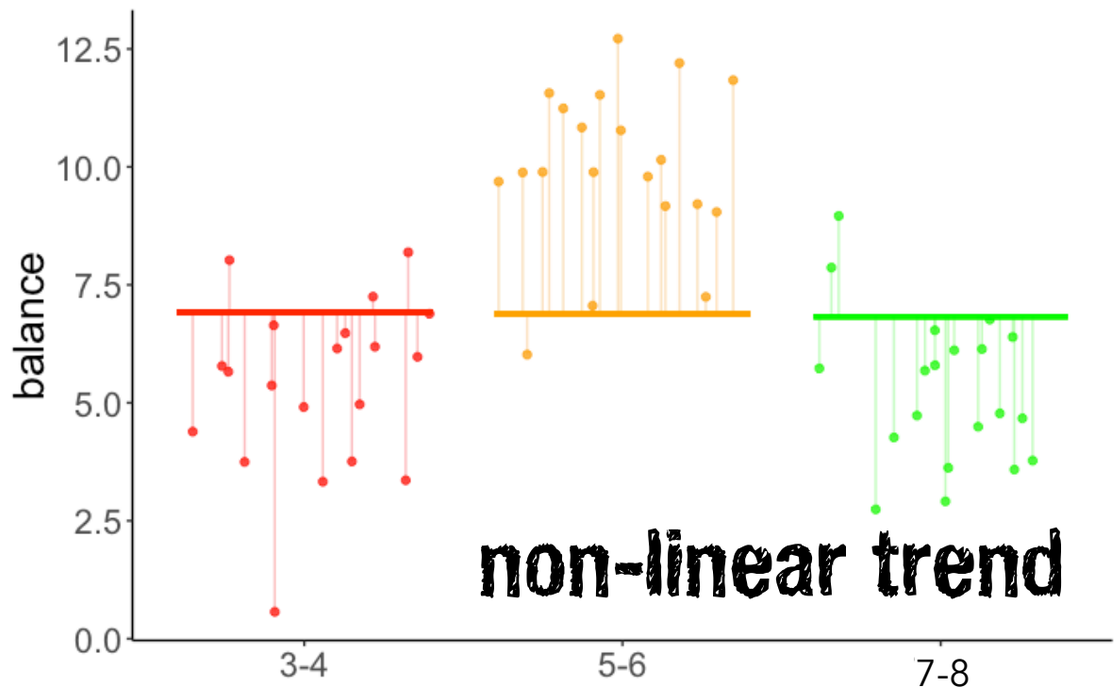
**Does performance increase with age?**

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

**Compact model**




**Augmented model**




**Model comparison**

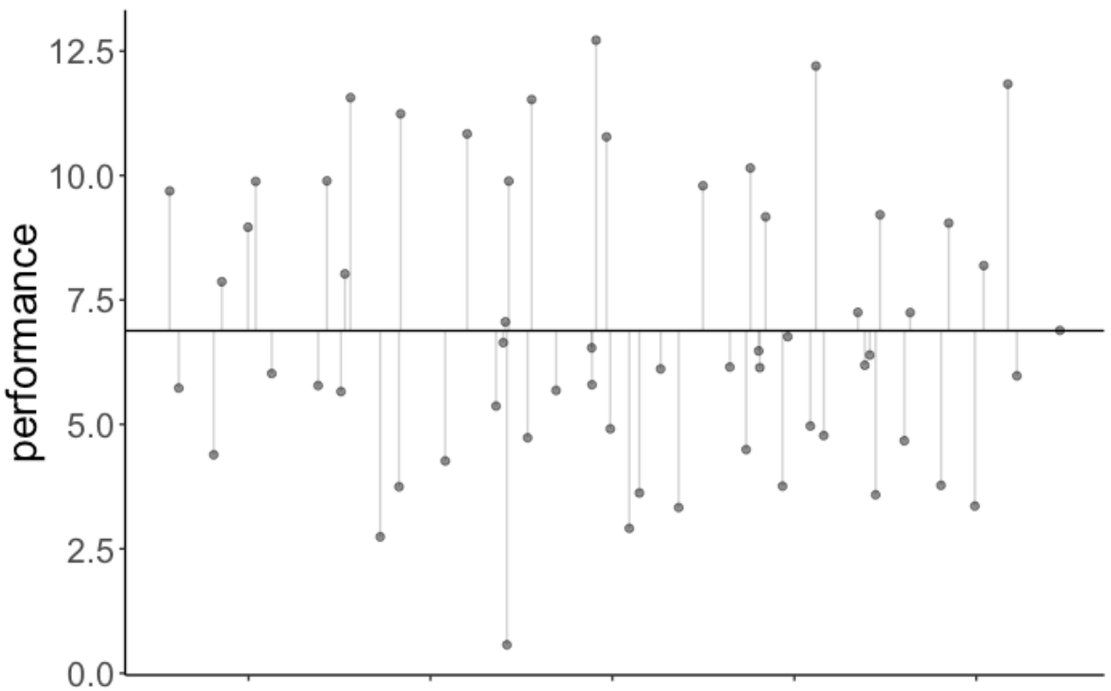
**p = .8508**

# Contrasts

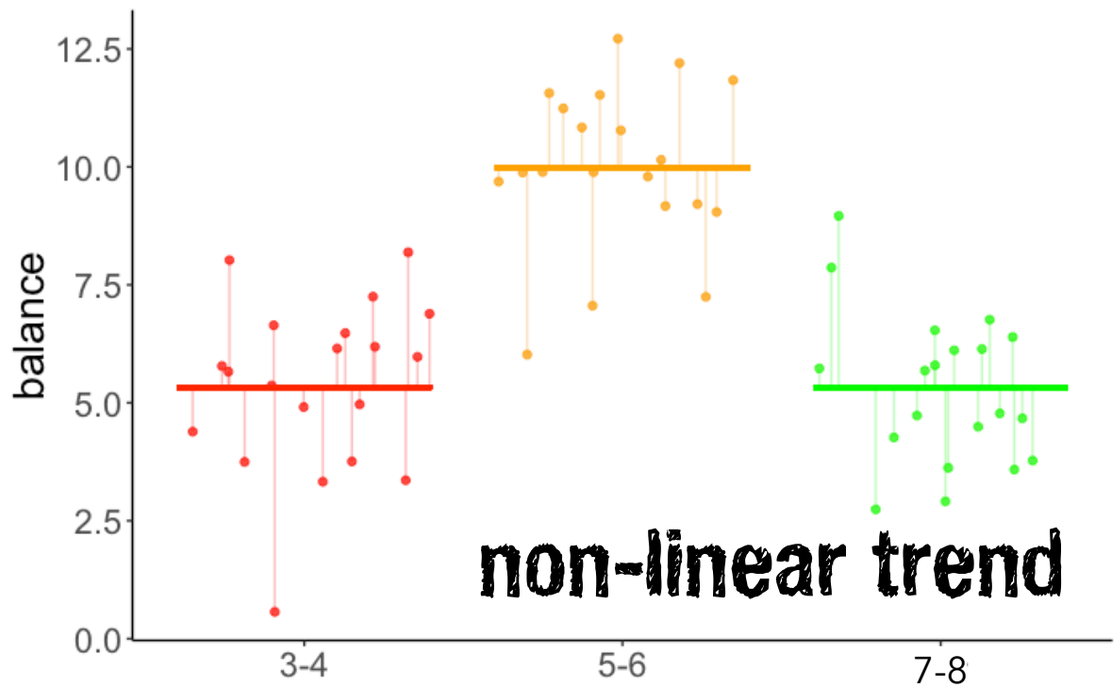
**Does performance increase with age?**

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

**Compact model**




**Augmented model**

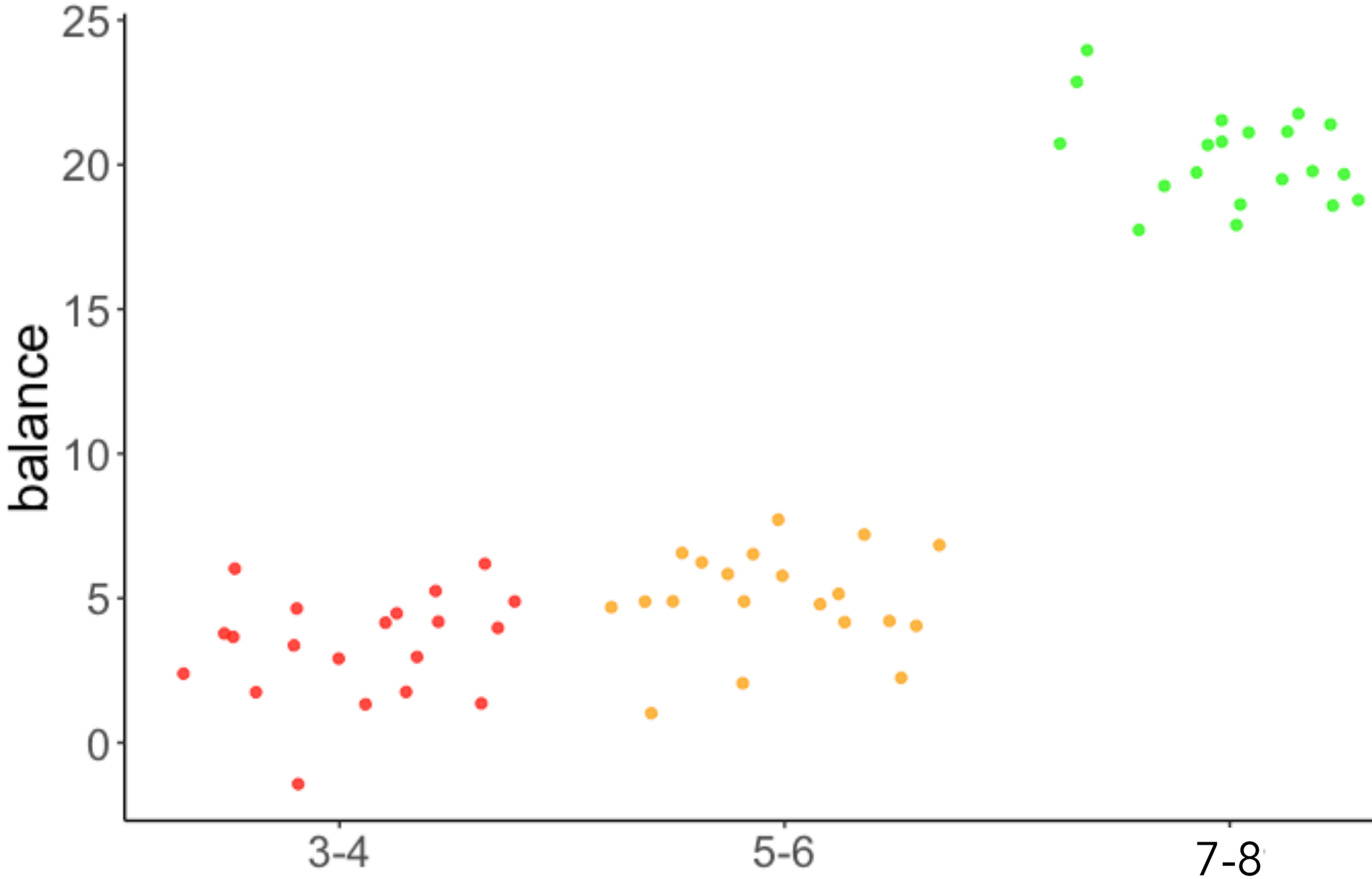



**Model comparison**

**p < .001**

# Contrasts

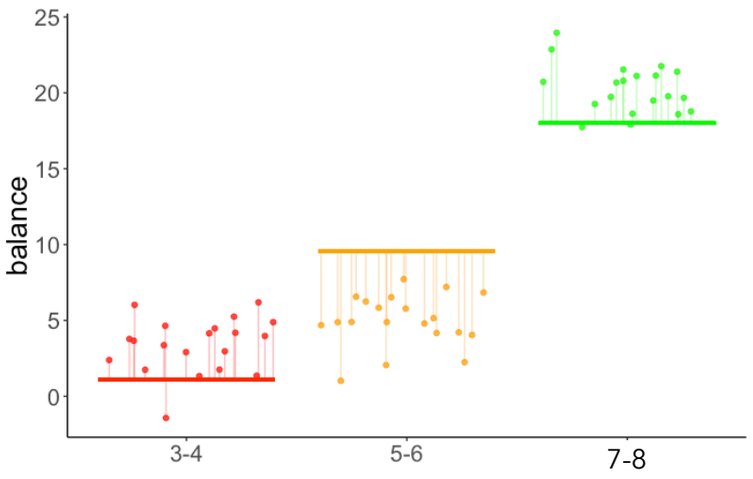
**Does performance increase with age?**



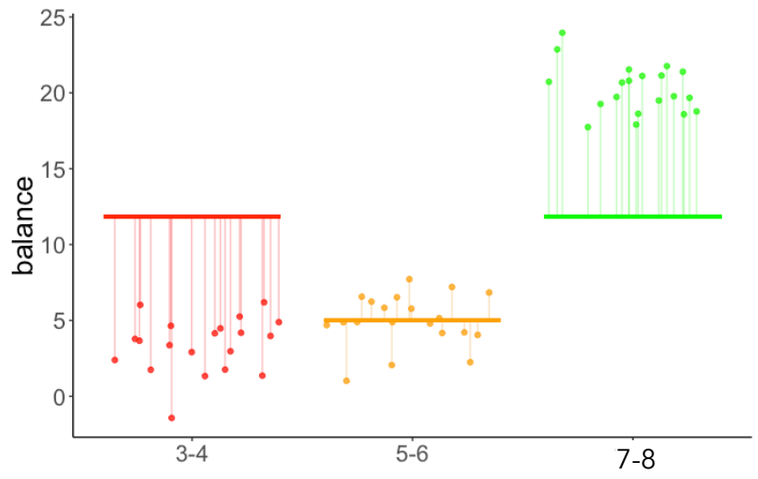

**Data from yet another hypothetical developmental study**

# Contrasts

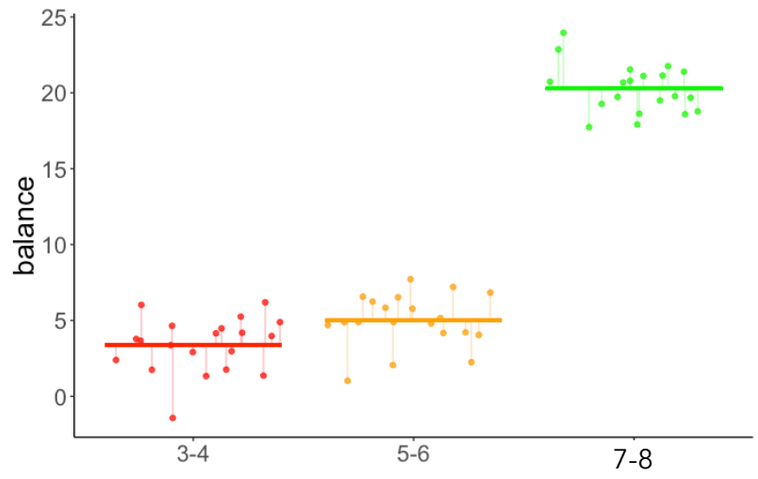
## Linear contrast




## Quadratic contrast




## Linear & Quadratic contrast




```
1 lm(formula = performance ~ group_contrast + I(group_contrast^2),
2     data = df.contrast)
```

```
Call:
lm(formula = performance ~ group_contrast + I(group_contrast^2),
    data = df.contrast)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8104 -0.9555  0.0742  1.1504  3.6832

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.9871     0.3872   12.88   <2e-16 ***
group_contrast        8.4483     0.2738   30.86   <2e-16 ***
I(group_contrast^2)   6.8423     0.4742   14.43   <2e-16 ***

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.732 on 57 degrees of freedom
Multiple R-squared:  0.9532,    Adjusted R-squared:  0.9515
F-statistic: 580.1 on 2 and 57 DF,  p-value: < 2.2e-16
```

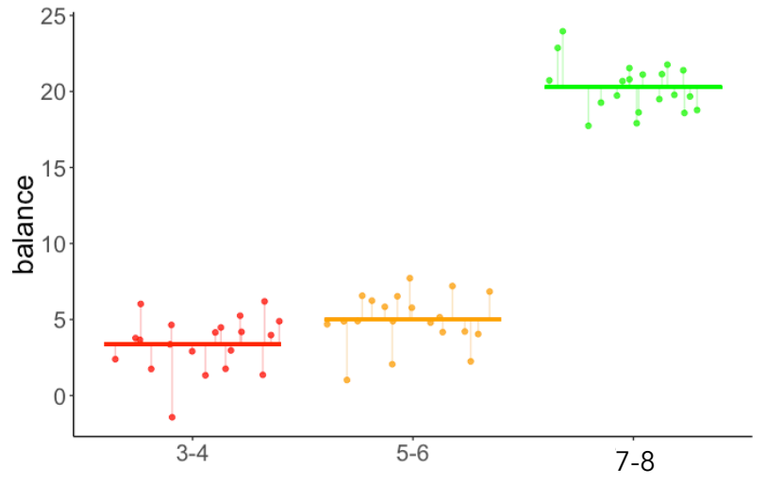
Inhibit Interpretation/
Conversion of Objects

# Contrasts

| term | estimate |
|---|---|
| (Intercept) | 4.99 |
| group_contrast | 8.45 |
| I(group_contrast^2) | 6.84 |

| contrast | 3-4 | 5-6 | 10+ |
|---|---|---|---|
| group_contrast | -1 | 0 | 1 |
| group_contrast2 | 1 | 0 | 1 |

**Linear & Quadratic contrast**




$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group\_contrast}_i + b_2 \cdot \text{group\_contrast2}_i$$

**if group == "3-4"**

$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group\_contrast} + b_2 \cdot \text{group\_contrast2}$$

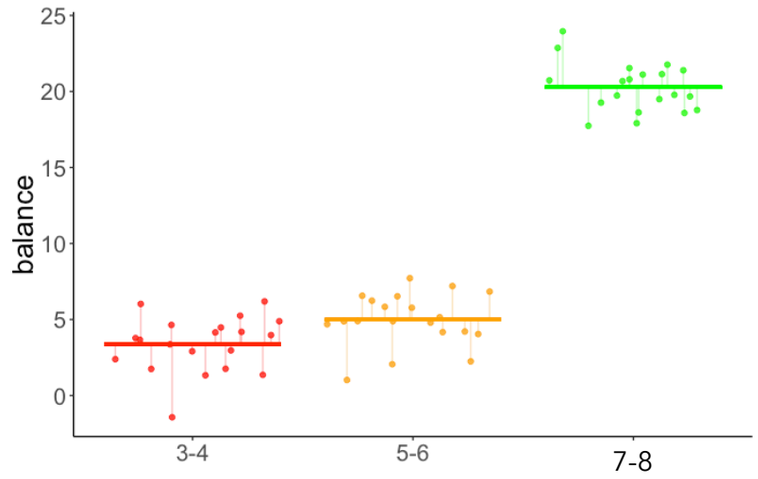
| group | performance | group_contrast | group_contrast2 |
|---|---|---|---|
| 3-4 | 1.75 | -1 | 1 |
| 3-4 | 3.37 | -1 | 1 |
| 3-4 | 1.33 | -1 | 1 |
| 5-6 | 6.84 | 0 | 0 |
| 5-6 | 6.56 | 0 | 0 |
| 5-6 | 5.15 | 0 | 0 |
| 10+ | 19.67 | 1 | 1 |
| 10+ | 19.49 | 1 | 1 |
| 10+ | 21.39 | 1 | 1 |

# Contrasts

| term | estimate |
|---|---|
| (Intercept) | 4.99 |
| group_contrast | 8.45 |
| I(group_contrast^2) | 6.84 |

| contrast | 3-4 | 5-6 | 10+ |
|---|---|---|---|
| group_contrast | -1 | 0 | 1 |
| group_contrast2 | 1 | 0 | 1 |

## Linear & Quadratic contrast




$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group\_contrast}_i + b_2 \cdot \text{group\_contrast2}_i$$

**if group == "5-6"**

$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group\_contrast} + b_2 \cdot \text{group\_contrast2}$$

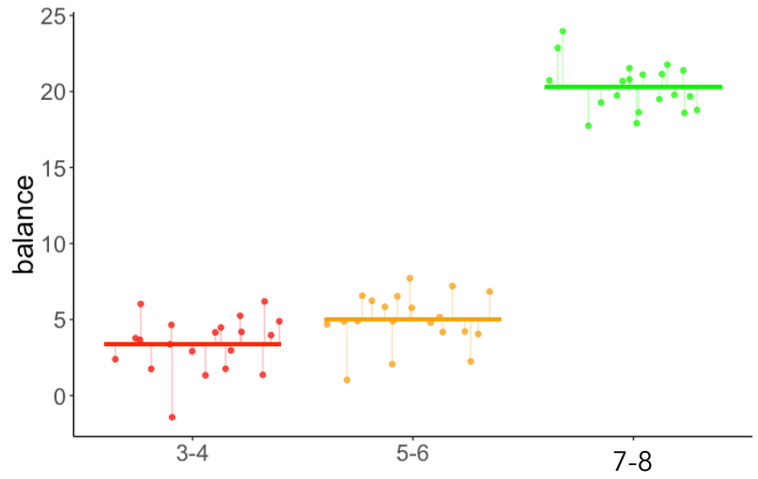
| group | performance | group_contrast | group_contrast2 |
|---|---|---|---|
| 3-4 | 1.75 | -1 | 1 |
| 3-4 | 3.37 | -1 | 1 |
| 3-4 | 1.33 | -1 | 1 |
| 5-6 | 6.84 | 0 | 0 |
| 5-6 | 6.56 | 0 | 0 |
| 5-6 | 5.15 | 0 | 0 |
| 10+ | 19.67 | 1 | 1 |
| 10+ | 19.49 | 1 | 1 |
| 10+ | 21.39 | 1 | 1 |

# Contrasts

| term | estimate |
|---|---|
| (Intercept) | 4.99 |
| group_contrast | 8.45 |
| I(group_contrast^2) | 6.84 |

| contrast | 3-4 | 5-6 | 10+ |
|---|---|---|---|
| group_contrast | -1 | 0 | 1 |
| group_contrast2 | 1 | 0 | 1 |

## Linear & Quadratic contrast




$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group\_contrast}_i + b_2 \cdot \text{group\_contrast2}_i$$

**if group == "10+"**

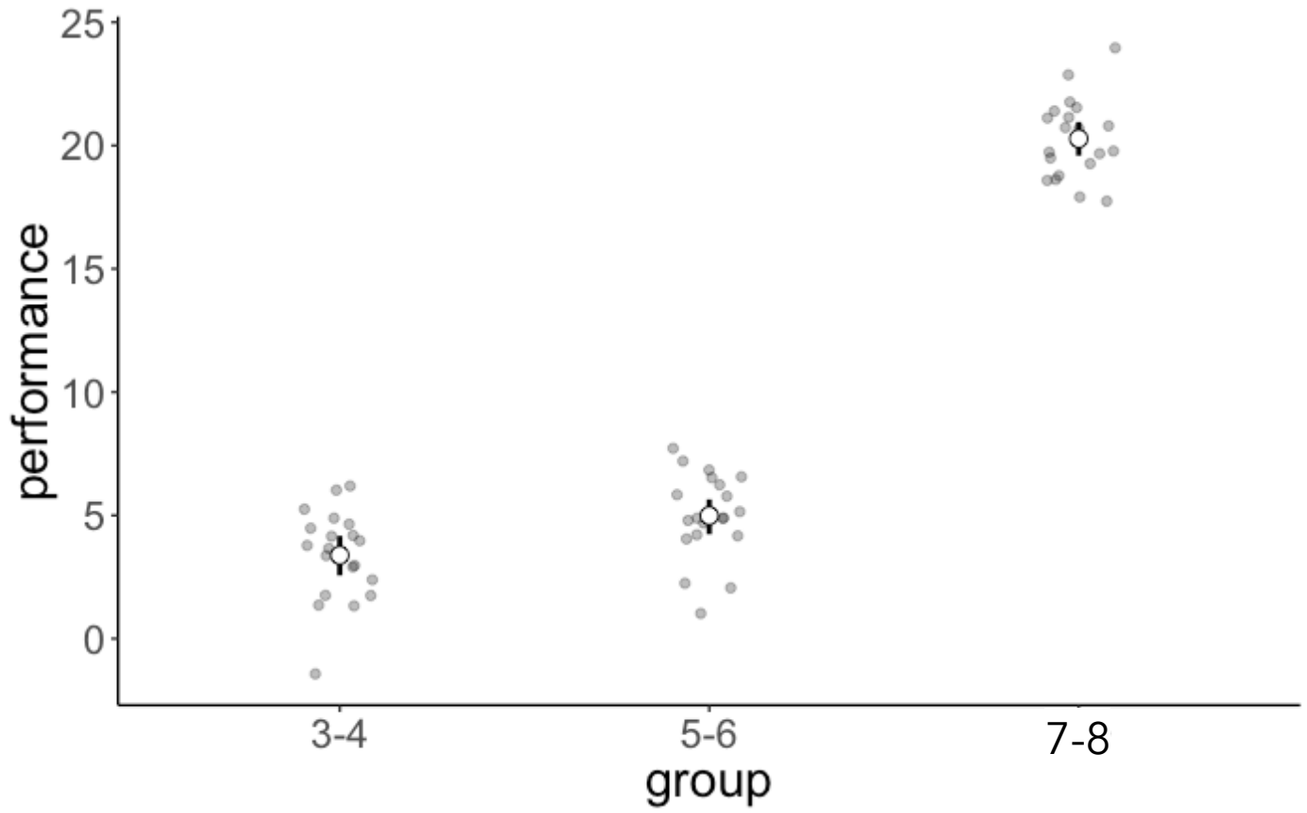
$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group\_contrast} + b_2 \cdot \text{group\_contrast2}$$

| group | performance | group_contrast | group_contrast2 |
|---|---|---|---|
| 3-4 | 1.75 | -1 | 1 |
| 3-4 | 3.37 | -1 | 1 |
| 3-4 | 1.33 | -1 | 1 |
| 5-6 | 6.84 | 0 | 0 |
| 5-6 | 6.56 | 0 | 0 |
| 5-6 | 5.15 | 0 | 0 |
| 10+ | 19.67 | 1 | 1 |
| 10+ | 19.49 | 1 | 1 |
| 10+ | 21.39 | 1 | 1 |

# Reporting results

Children's performance increased with age $t(56) = 30.86$, $p < .001$.

The effect of age on performance was non-linear $t(56) = 14.43$, $p < .001$.




```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            4.9871     0.3872   12.88   <2e-16 ***
group_contrast         8.4483     0.2738   30.86   <2e-16 ***
I(group_contrast^2)    6.8423     0.4742   14.43   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

**possible follow up tests ...**

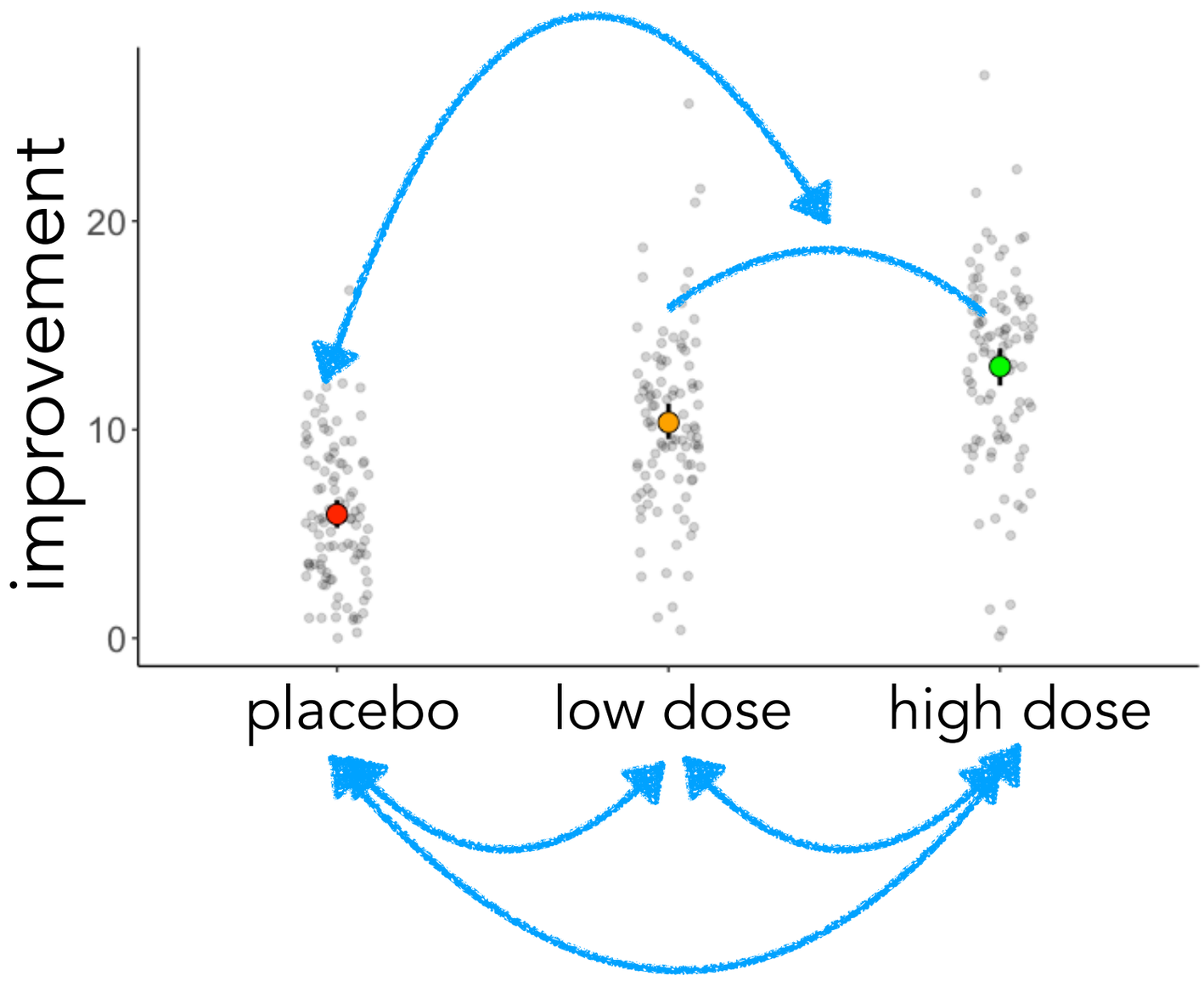
While the performance between 3-4 year olds, and 5-6 year olds did not differ significantly, 7-8 year olds performed significantly better than the other age groups.

# Planned comparisons

# Planned contrasts

**must be defined in advance**

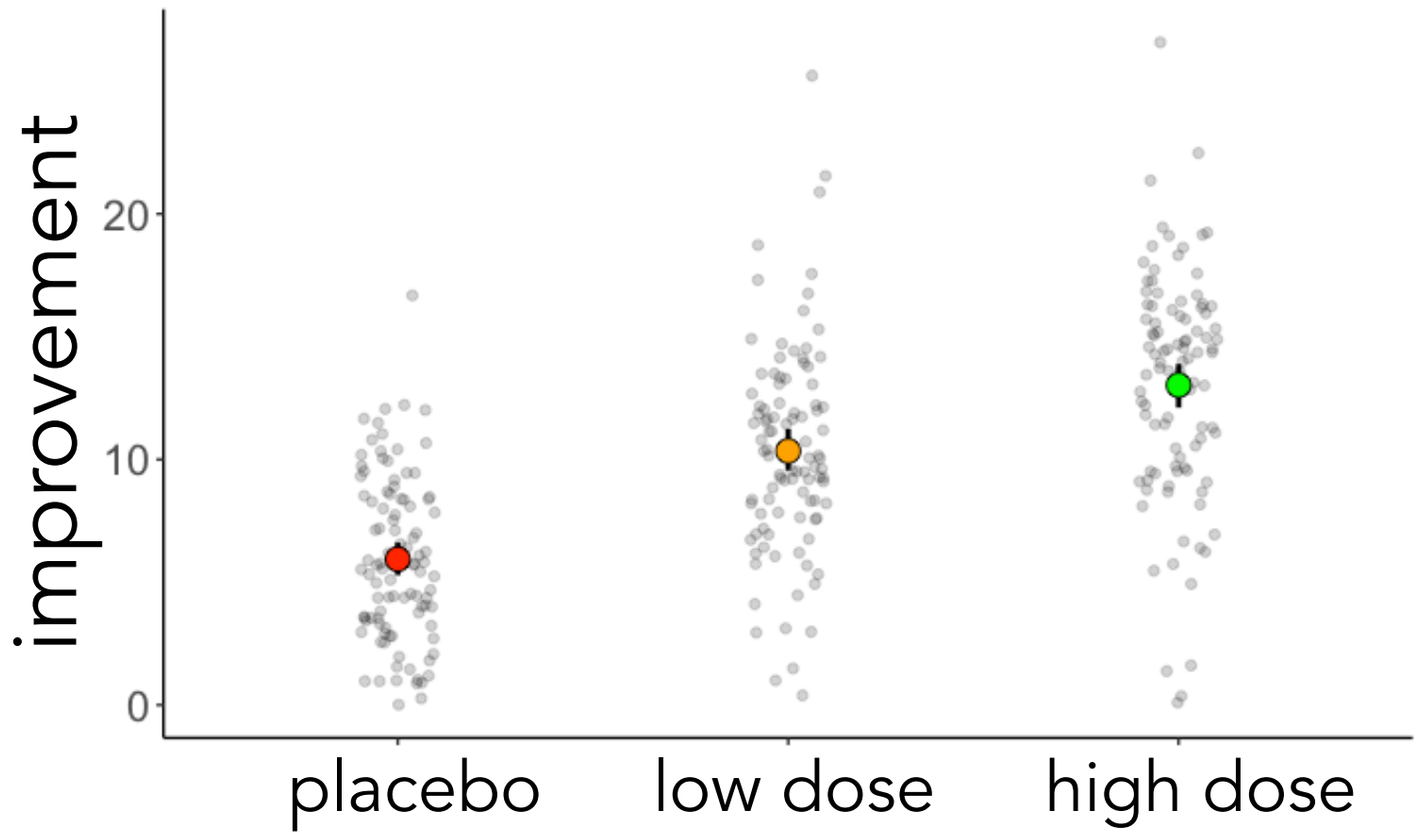
1. Is the treatment different from the placebo?
2. Do the two treatments differ from each other?



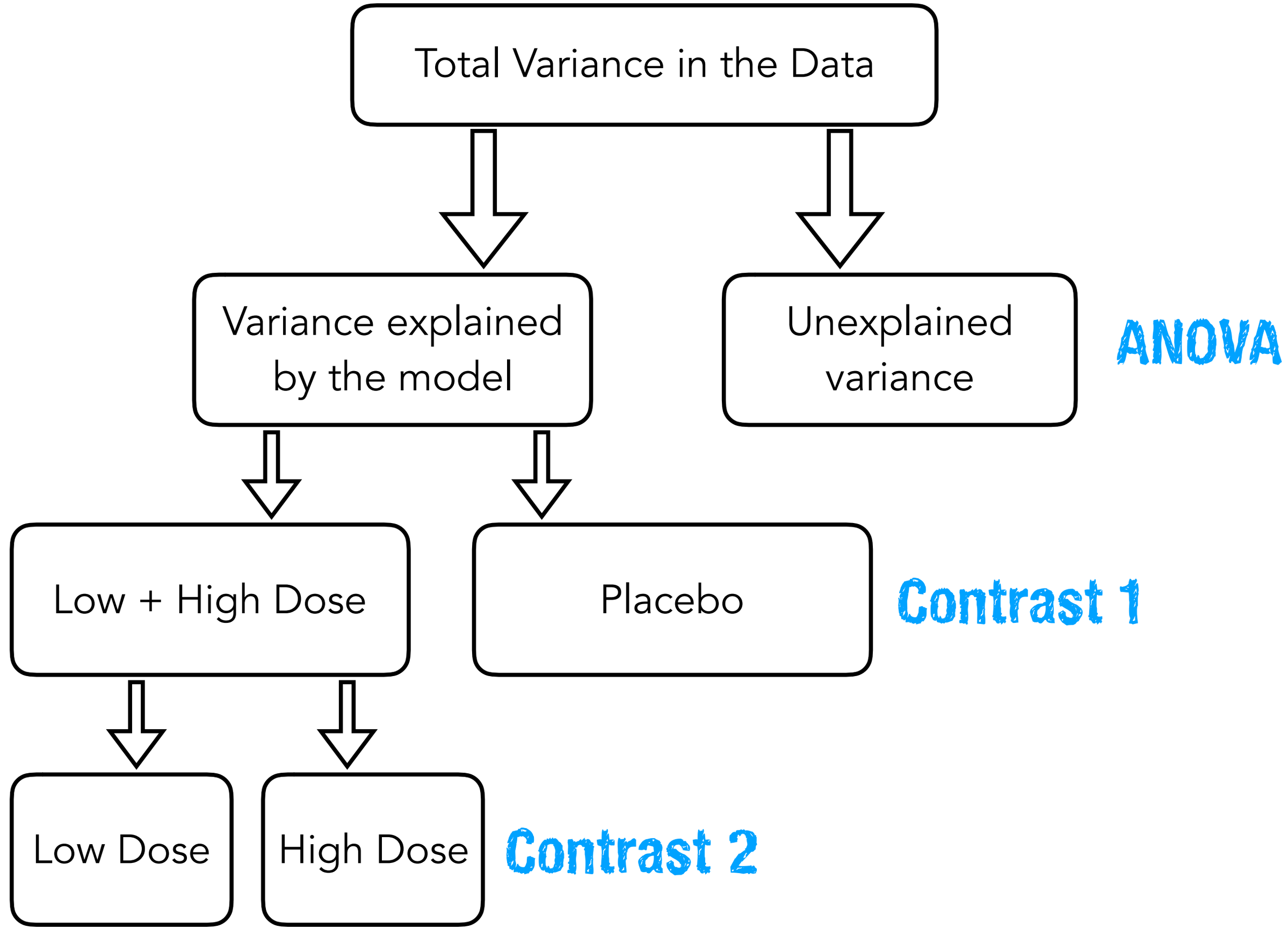

six possible comparisons

if we test all hypotheses, we increase the chance of making a type I error

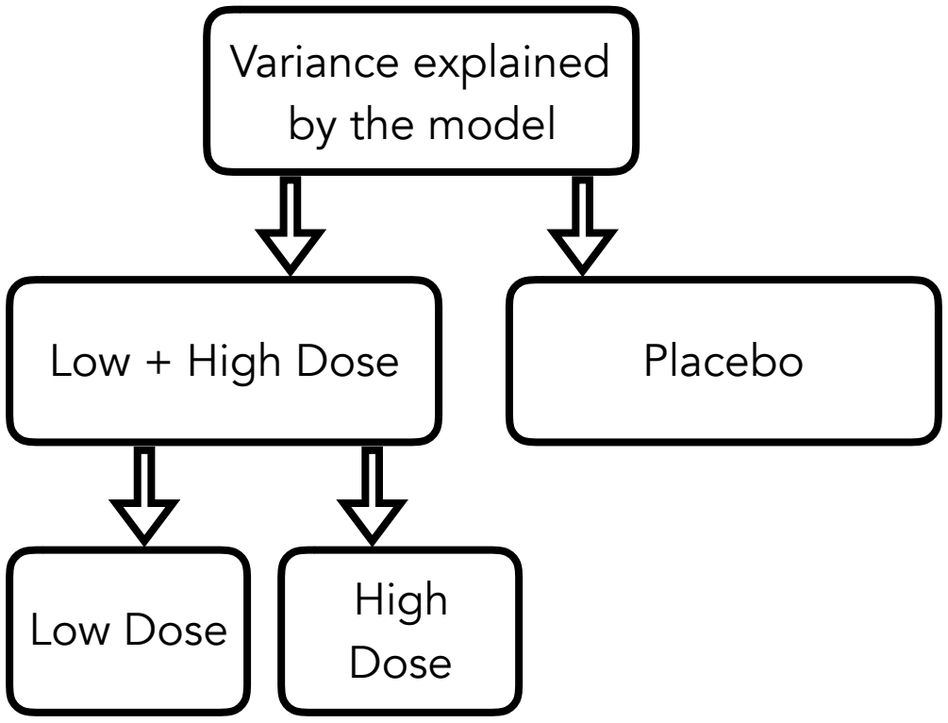
# Planned contrasts




Orthogonal contrasts allow us to partition the variance explained by the ANOVA model into a maximum of (#treatment - 1) meaningful and targeted comparisons involving different combinations of means.


Total Variance in the Data
Variance explained by the model
Unexplained variance
ANOVA
Low + High Dose
Placebo
Contrast 1
Low Dose
High Dose
Contrast 2

# Planned contrasts



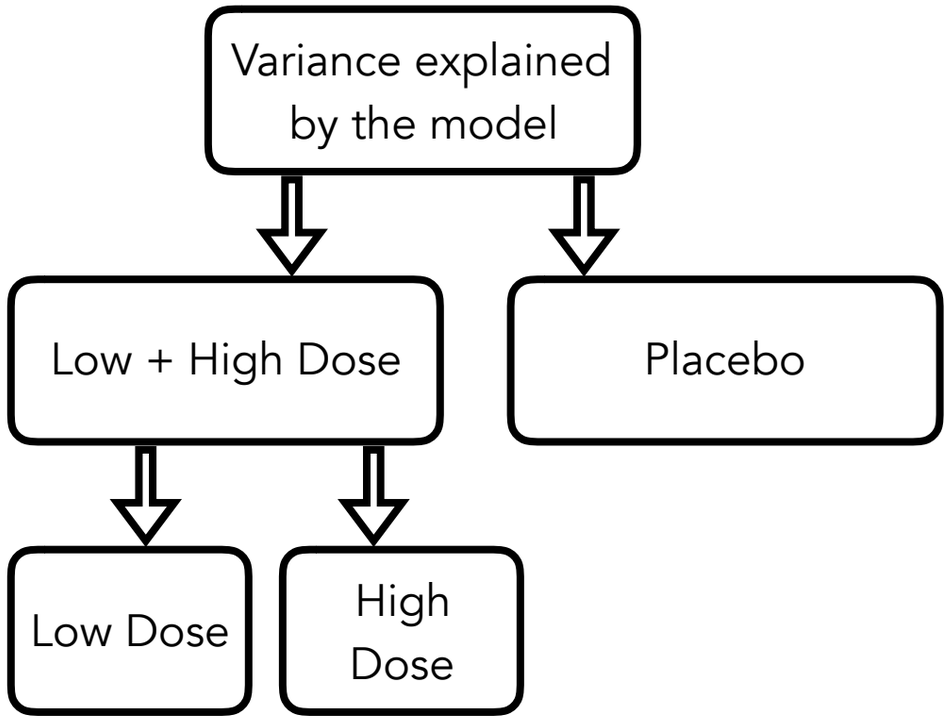

- Each contrast must compare only two "chunks" of variation
- Once a group has been singled out in a contrast, it can't be used in another contrast

# Defining orthogonal contrast codes

1. Groups coded with positive weights will be compared against groups with negative weights
2. Assign a 0 to groups who are not involved in a comparison
3. The sum of weights for each comparison should be 0
4. The product of the weights for any pair comparisons should sum to 0

# Defining contrast codes

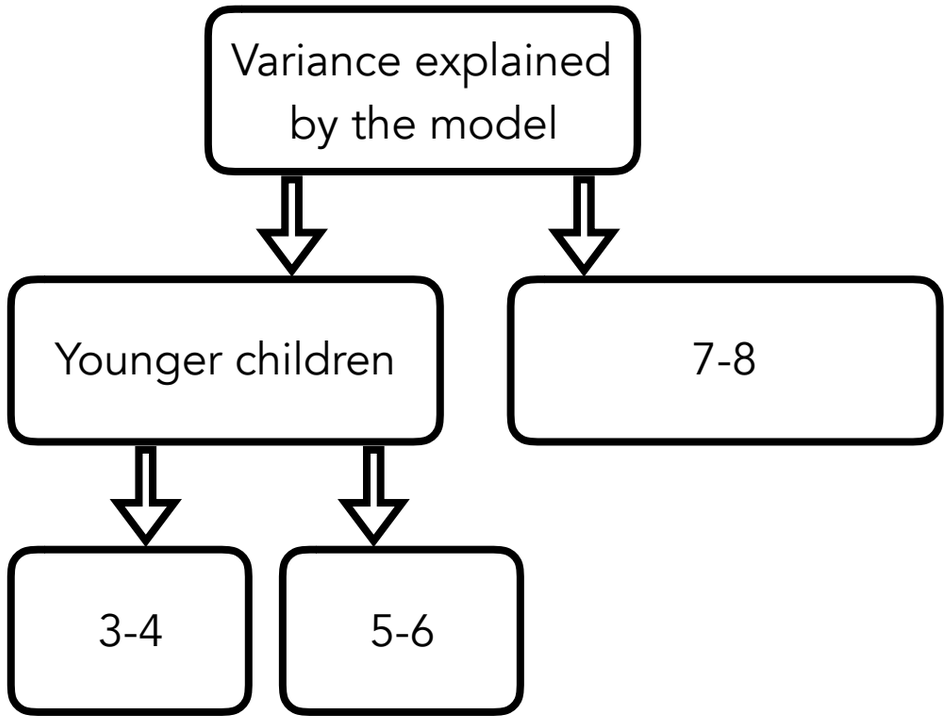
- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0




| contrast | Low | High | Placebo | sum |
|---|---|---|---|---|
| Dose vs Placebo | 0.5 | 0.5 | -1 | 0 |
| Low vs. high | -1 | 1 | 0 | 0 |
| Product | -0.5 | 0.5 | 0 | 0 |

# Defining contrast codes

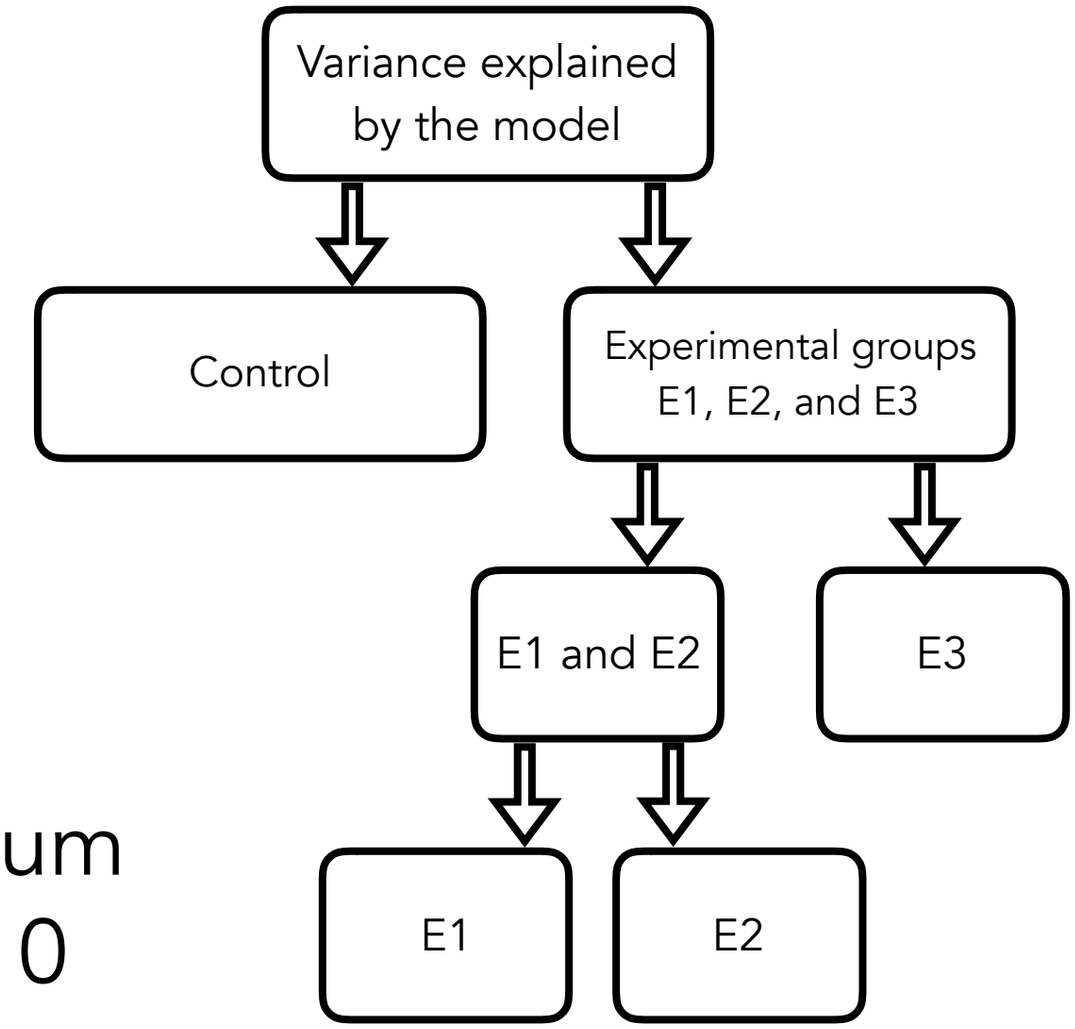
- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0




| contrast | 3-4 | 5-6 | 7-8 | sum |
|---|---|---|---|---|
| Young vs. old | -0.5 | -0.5 | 1 | 0 |
| 3-4 vs. 5-6 | -1 | 1 | 0 | 0 |
| Product | 0.5 | -0.5 | 0 | 0 |

# Defining contrast codes

- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0



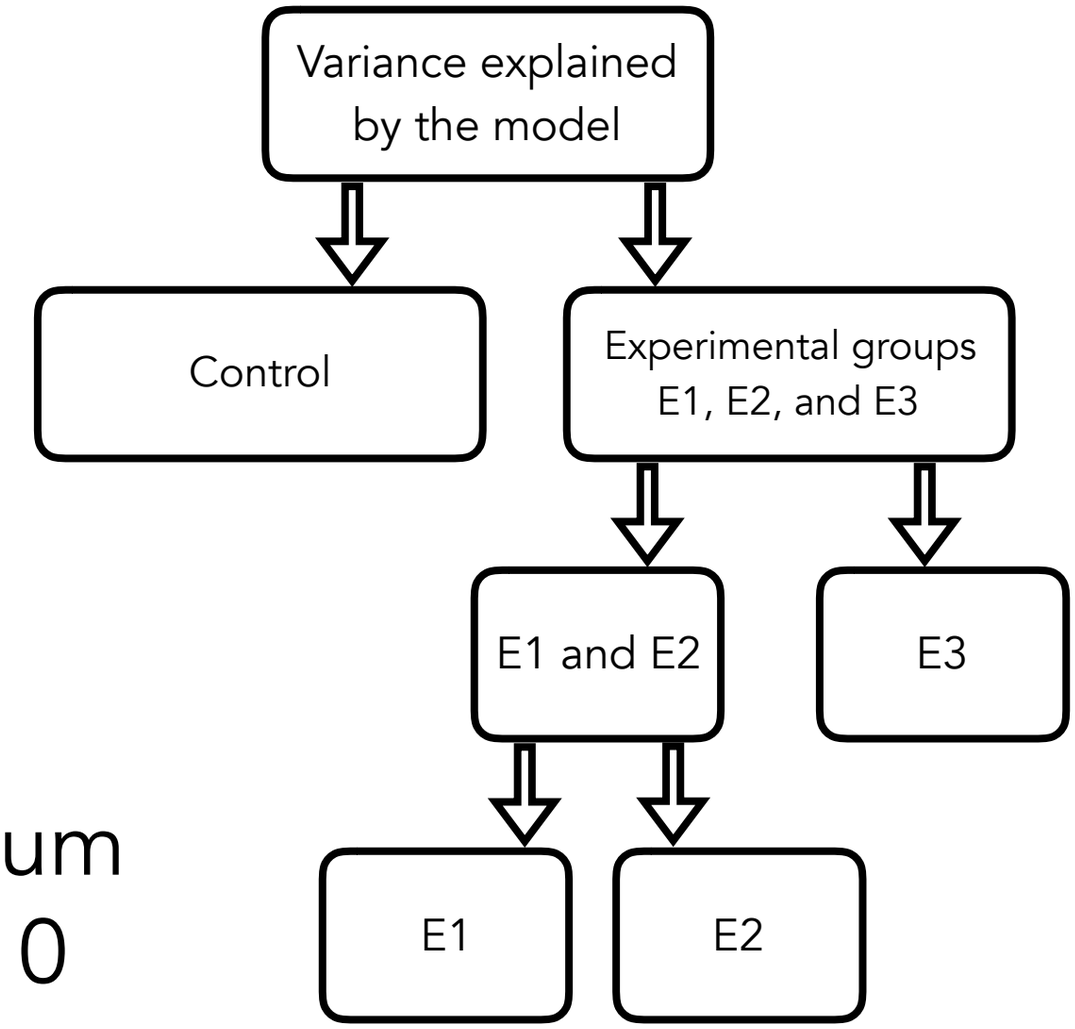

| contrast | Control | E1 | E2 | E3 | sum |
|---|---|---|---|---|---|
| Control vs. E | -3 | 1 | 1 | 1 | 0 |
| E1&2 vs. E3 | 0 | -1 | -1 | 2 | 0 |
| E1 vs. E2 | 0 | -1 | 1 | 0 | 0 |

## Check for products

| contrast | Control | E1 | E2 | E3 | |
|---|---|---|---|---|---|
| Control vs. E | -3 | 1 | 1 | 1 | |
| E1&2 vs. E3 | 0 | -1 | -1 | 2 | sum |
| product | 0 | -1 | -1 | 2 | 0 |

# Defining contrast codes

- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0



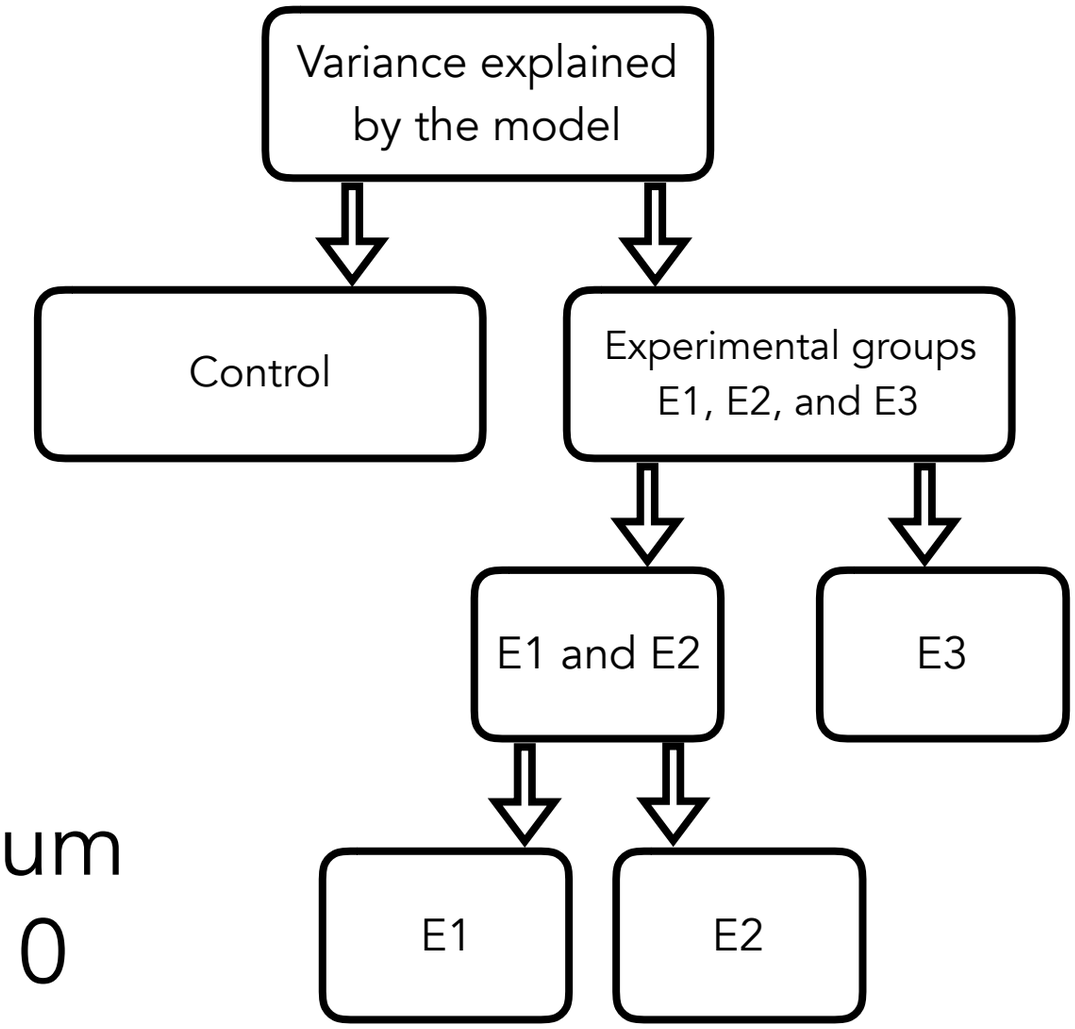

| contrast | Control | E1 | E2 | E3 | sum |
|---|---|---|---|---|---|
| Control vs. E | -3 | 1 | 1 | 1 | 0 |
| E1&2 vs. E3 | 0 | -1 | -1 | 2 | 0 |
| E1 vs. E2 | 0 | -1 | 1 | 0 | 0 |

## Check for products

| contrast | Control | E1 | E2 | E3 | sum |
|---|---|---|---|---|---|
| Control vs. E | -3 | 1 | 1 | 1 | |
| E1 vs. E2 | 0 | -1 | 1 | 0 | |
| product | 0 | -1 | 1 | 0 | 0 |

# Defining contrast codes

- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0



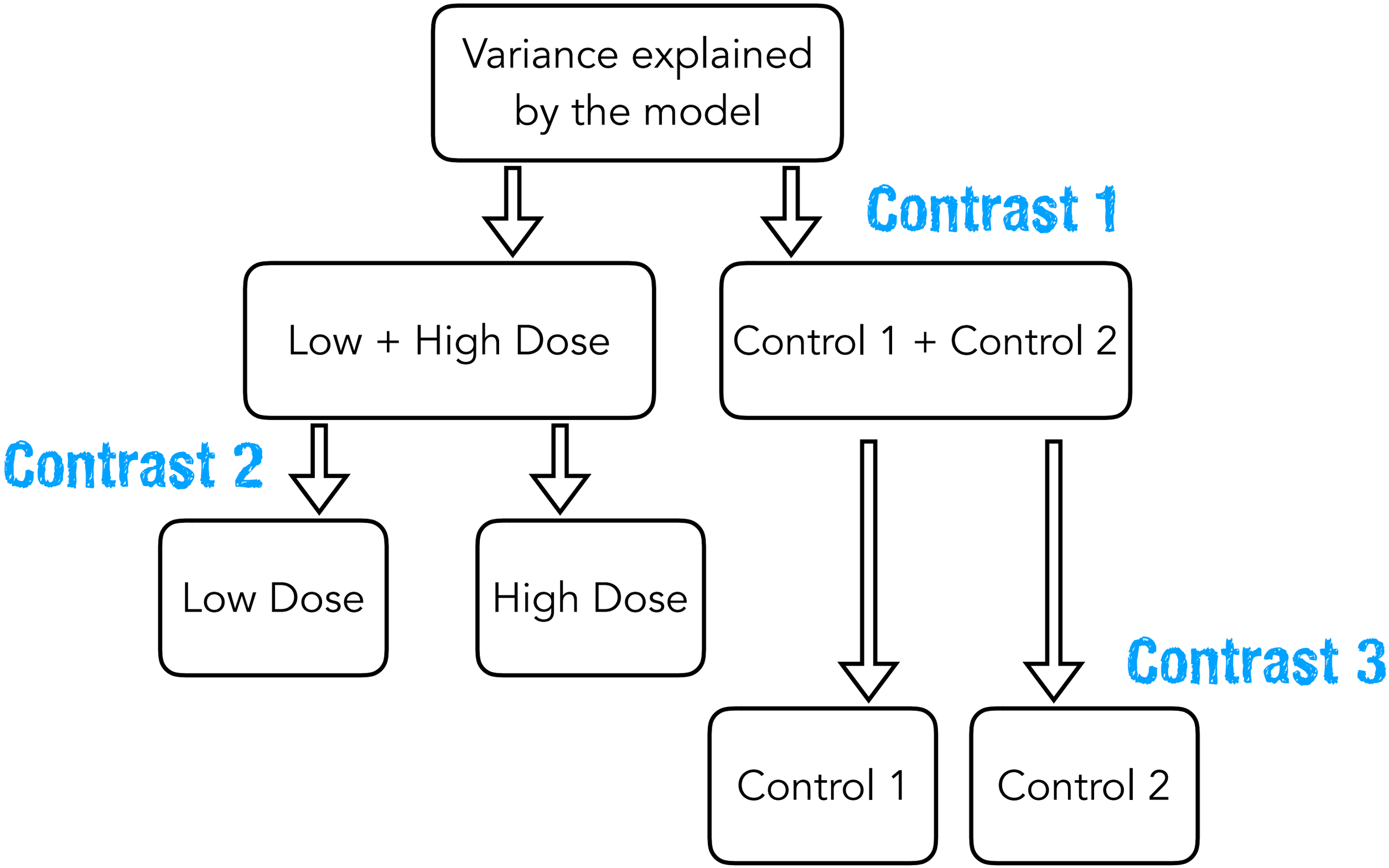

| contrast | Control | E1 | E2 | E3 | sum |
|---|---|---|---|---|---|
| Control vs. E | -3 | 1 | 1 | 1 | 0 |
| E1&2 vs. E3 | 0 | -1 | -1 | 2 | 0 |
| E1 vs. E2 | 0 | -1 | 1 | 0 | 0 |

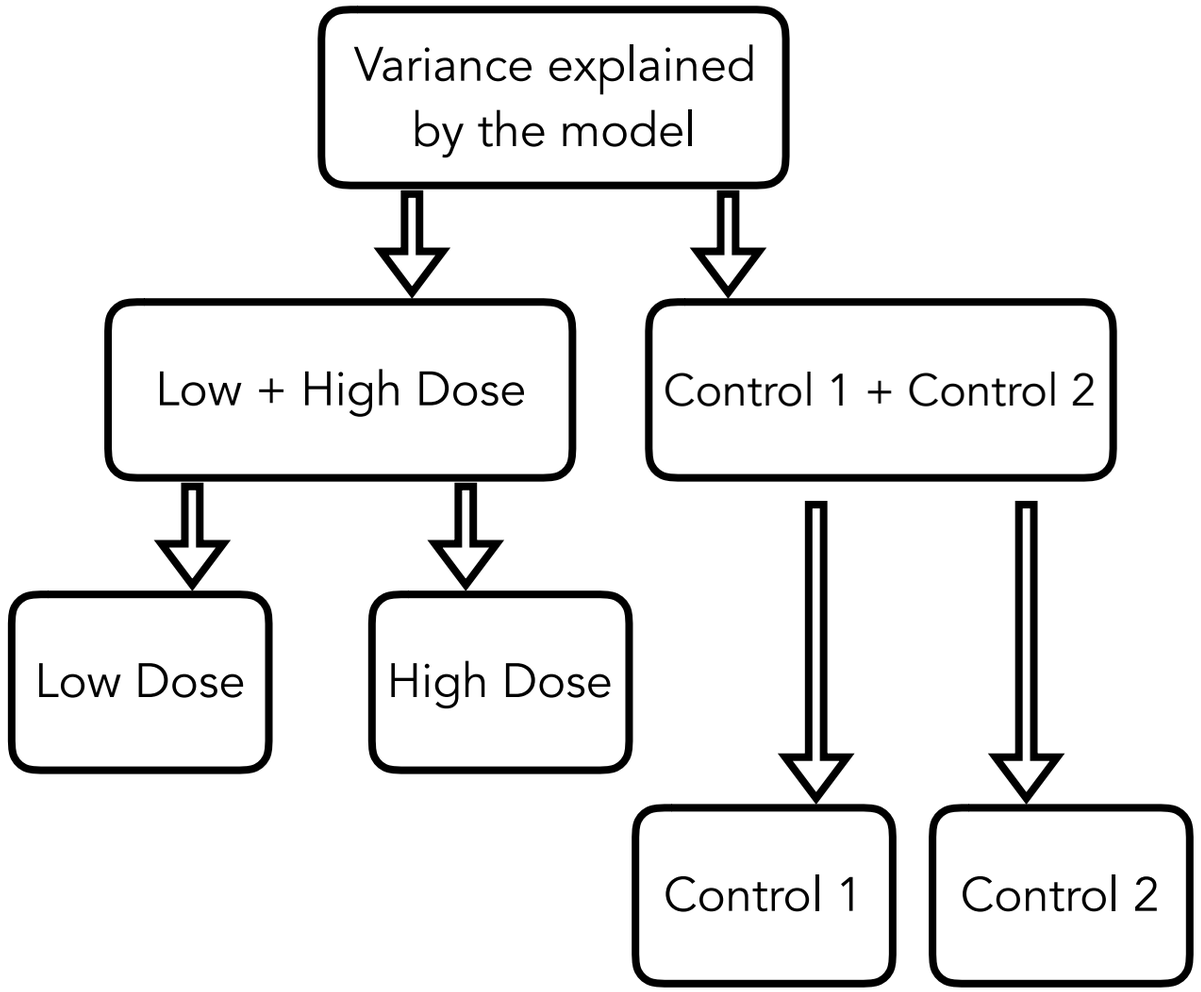
contrasts are orthogonal

## Check for products

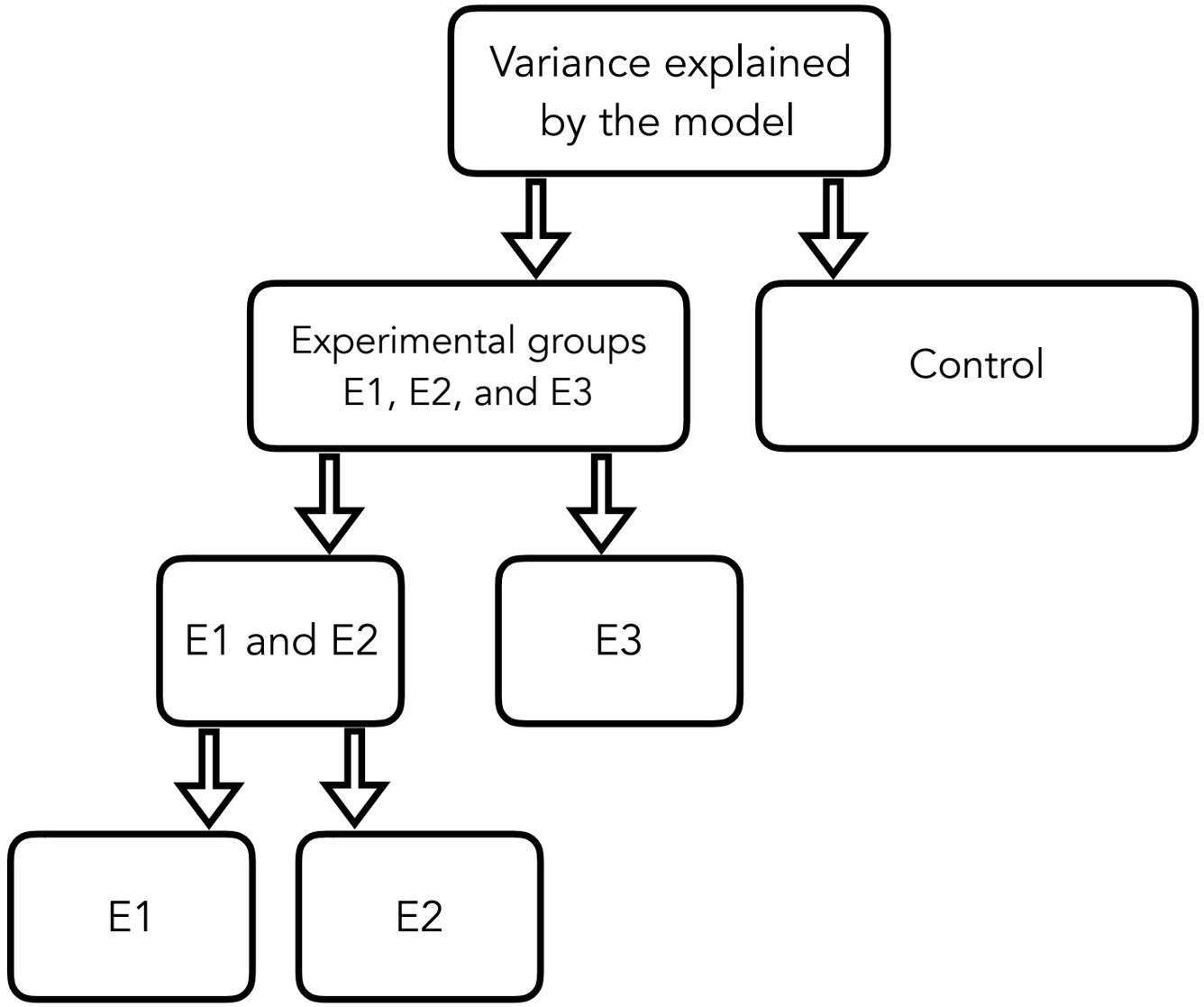
| contrast | Control | E1 | E2 | E3 |
|---|---|---|---|---|
| E1&2 vs. E3 | 0 | -1 | -1 | 2 |
| E1 vs. E2 | 0 | -1 | 1 | 0 |
| product | 0 | 1 | -1 | 0 |

sum 0


Variance explained by the model
Contrast 1
Low + High Dose
Control 1 + Control 2
Contrast 2
Low Dose
High Dose
Contrast 3
Control 1
Control 2




| contrast | low | high | C1 | C2 |
|---|---|---|---|---|
| low/high vs. C1/C2 | | | | |
| low vs. high | | | | |
| C1 vs. C2 | | | | |




| contrast | E1 | E2 | E2 | C |
|---|---|---|---|---|
| E1/E2/E3 vs. C | | | | |
| E1/E2 vs. E3 | | | | |
| E1 vs. E2 | | | | |

# Planned contrasts in R

```r
library("emmeans")

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

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

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

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

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

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

```
[1] "3-4" "5-6" "7-8"
 contrast       estimate        SE df t.ratio p.value
 young_vs_old  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
```

# Post hoc tests

```r
library("emmeans")

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

# post hoc tests
leastsquare = emmeans(fit, "group")
pairs(leastsquare,
      adjust = "bonferroni")
```

```
contrast    estimate        SE df t.ratio p.value
 3-4 - 5-6  -1.606009 0.5475962 57  -2.933  0.0145
 3-4 - 10+ -16.896546 0.5475962 57 -30.856  <.0001
 5-6 - 10+ -15.290537 0.5475962 57 -27.923  <.0001

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

**all pairwise tests between groups**

# Post hoc tests

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

# post hoc tests
leastsquare = emmeans(fit, c("hand", "skill"))
pairs(leastsquare,
      adjust = "bonferroni")
```

the poker data

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

P value adjustment: bonferroni method for 15 tests

all pairwise tests between groups

that's a lot of tests!

... not

# Coding schemes

# 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 |

# Effect coding

# Effect coding

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

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

**reference**

grand mean = 9.77

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

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

**reference**

grand mean = 9.77

**Note: The last level in each factor is dropped.**

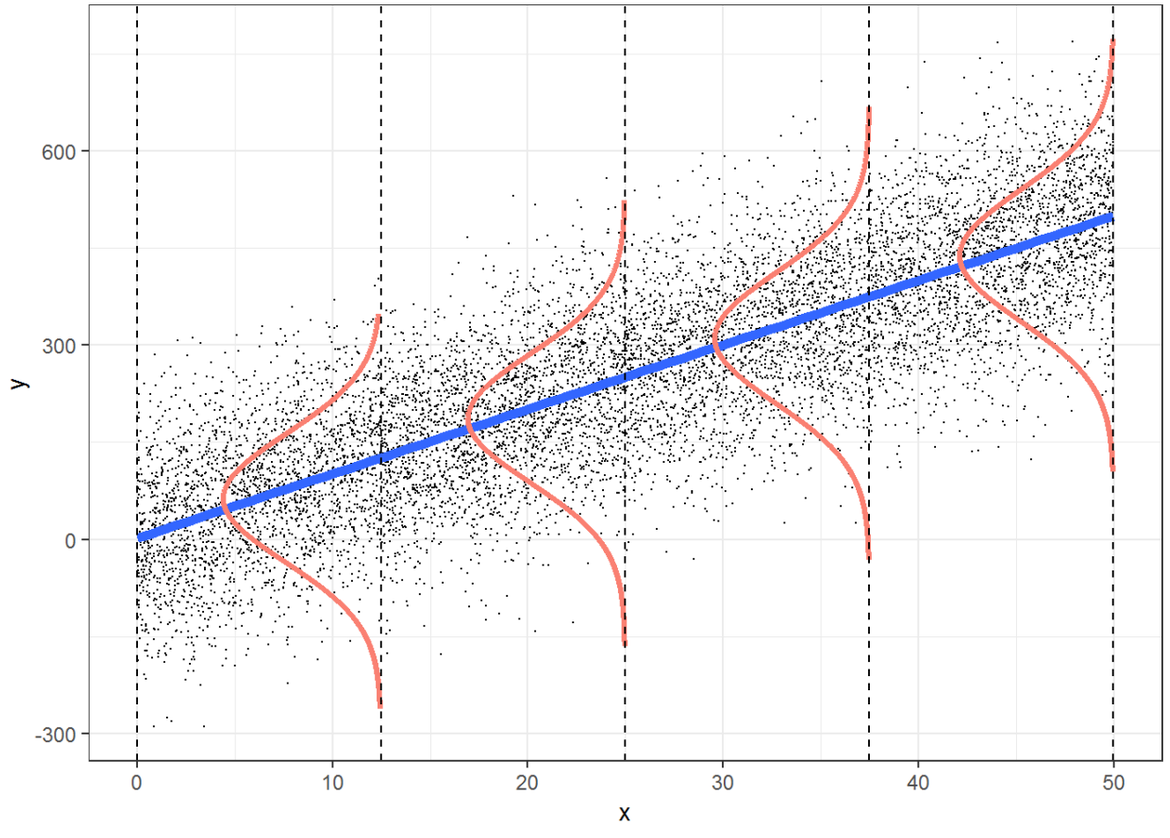
# Coding schemes

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

# Linear model assumptions

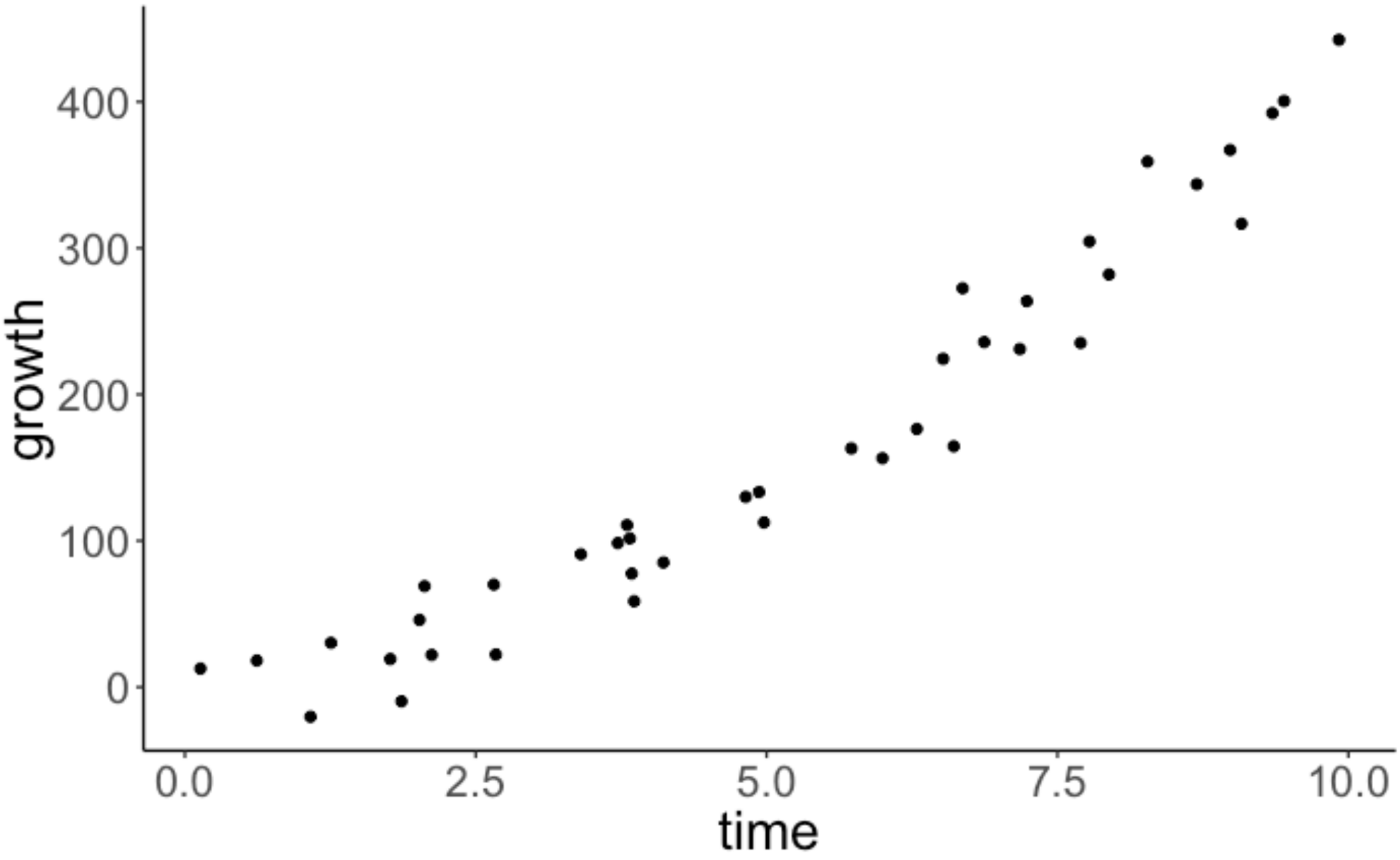
# Assumptions of linear model

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



# Distribution of errors

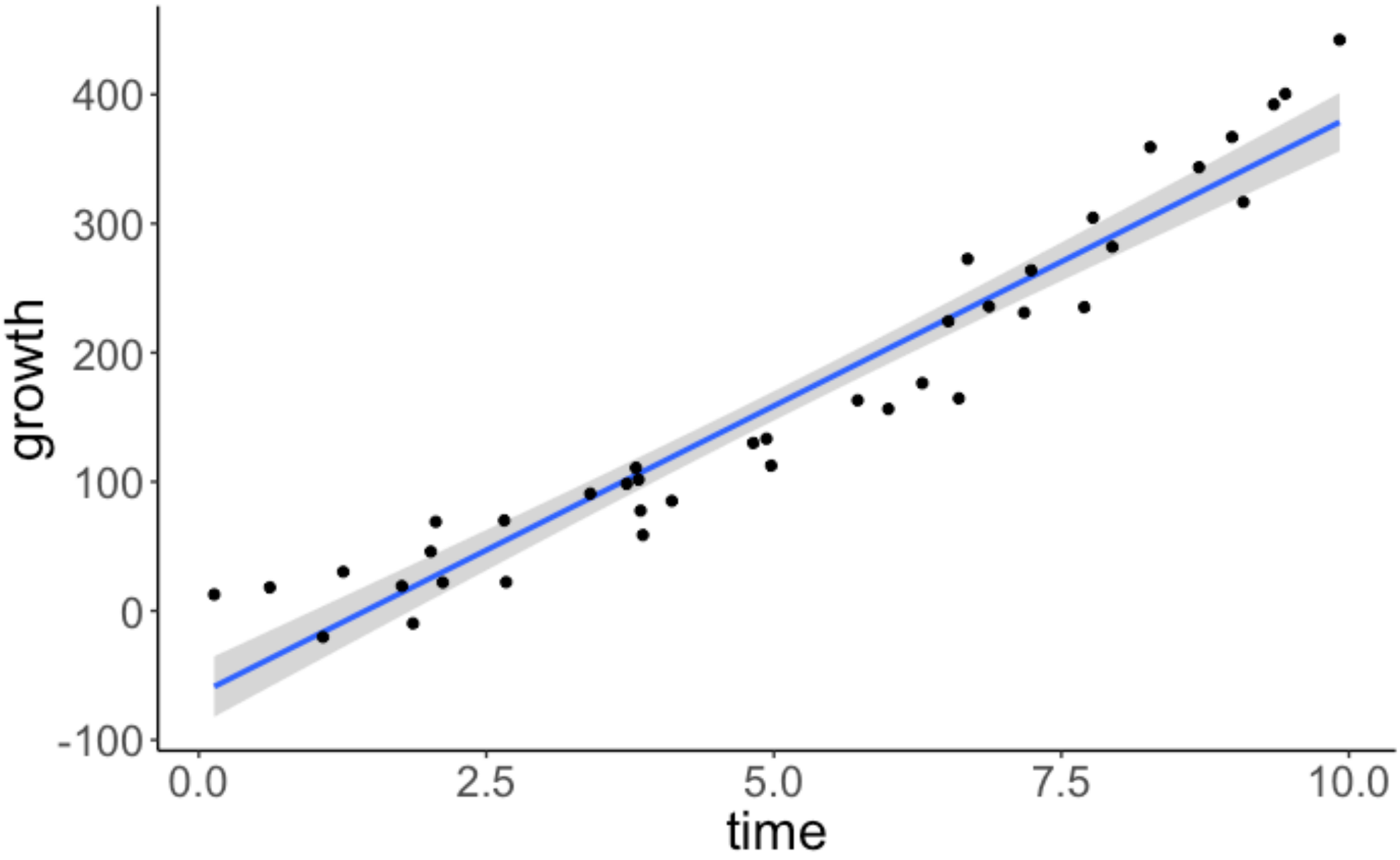
# Distribution of errors




**simulated data set**

# Distribution of errors

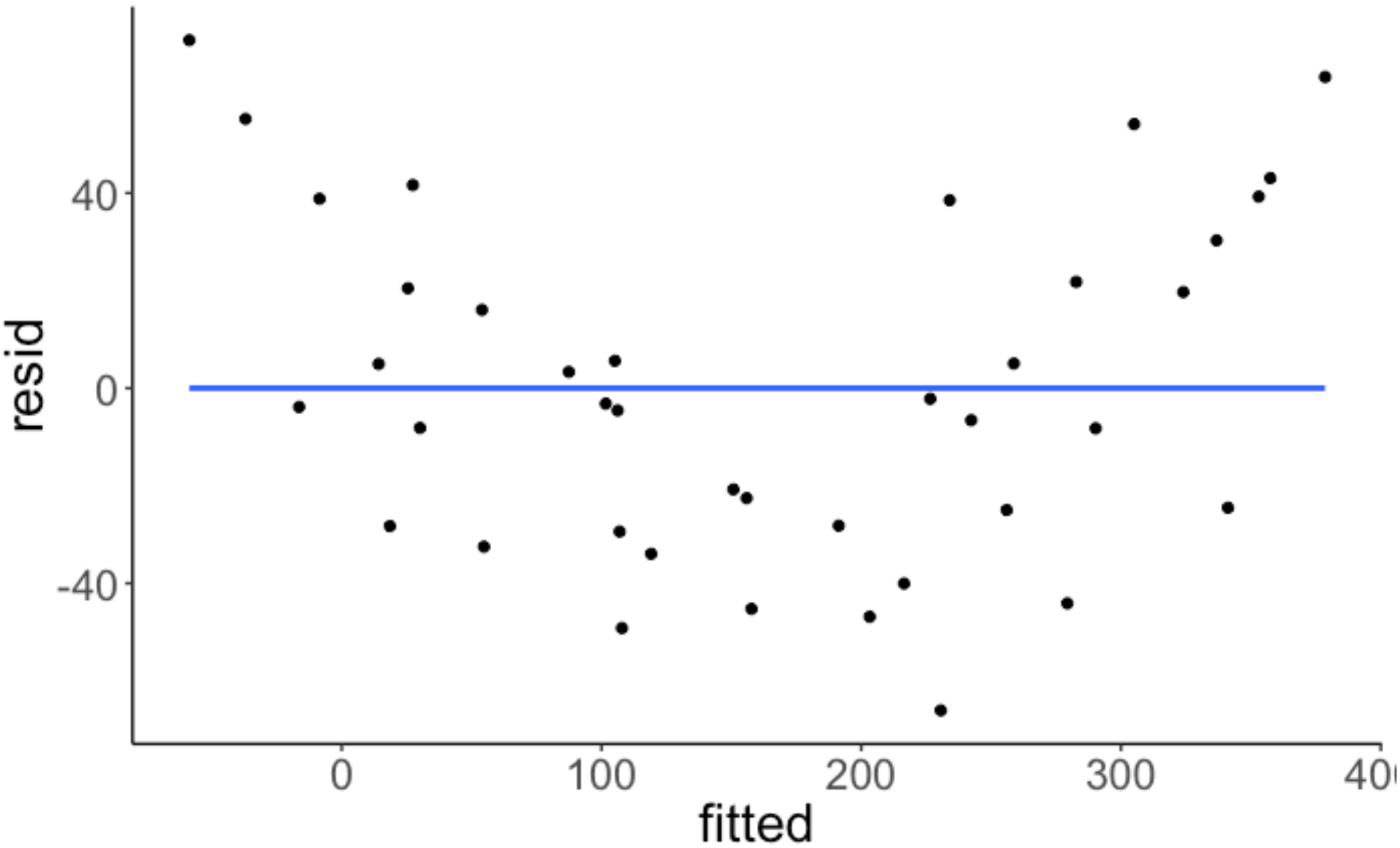
```
lm(formula = growth ~ 1 + time, data = df.growth)
```




**simulated data set**

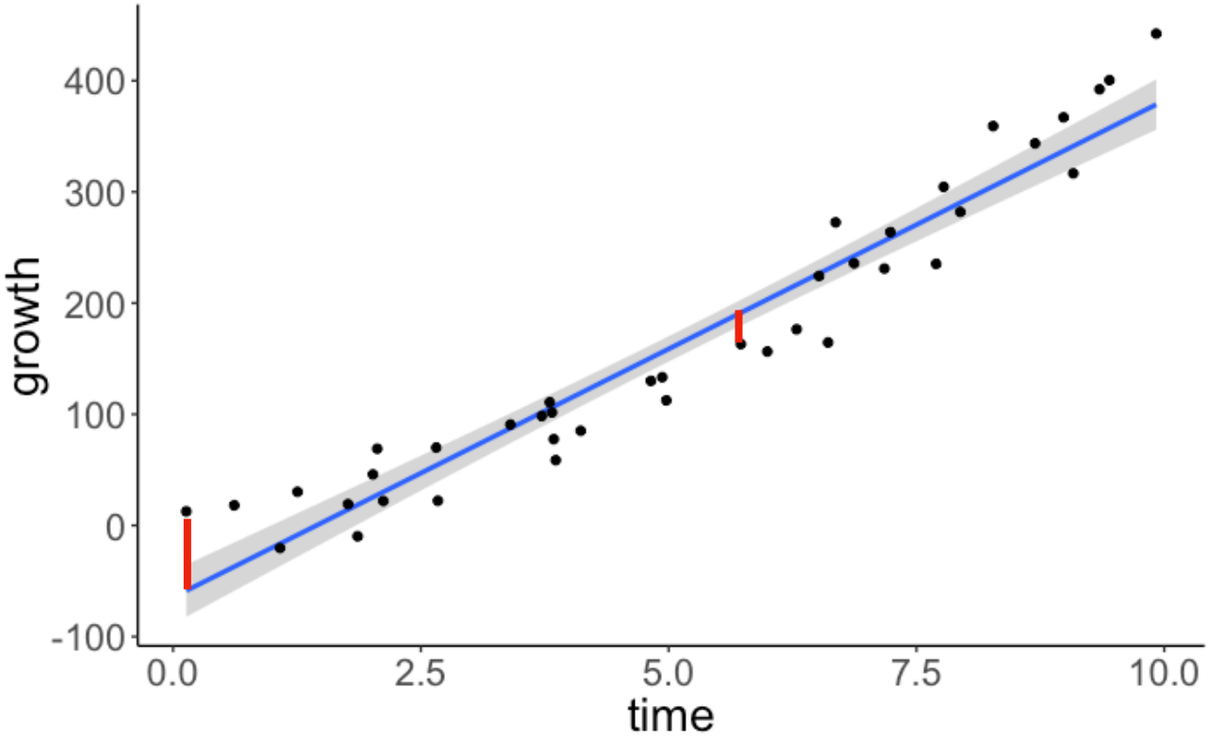
# Distribution of errors

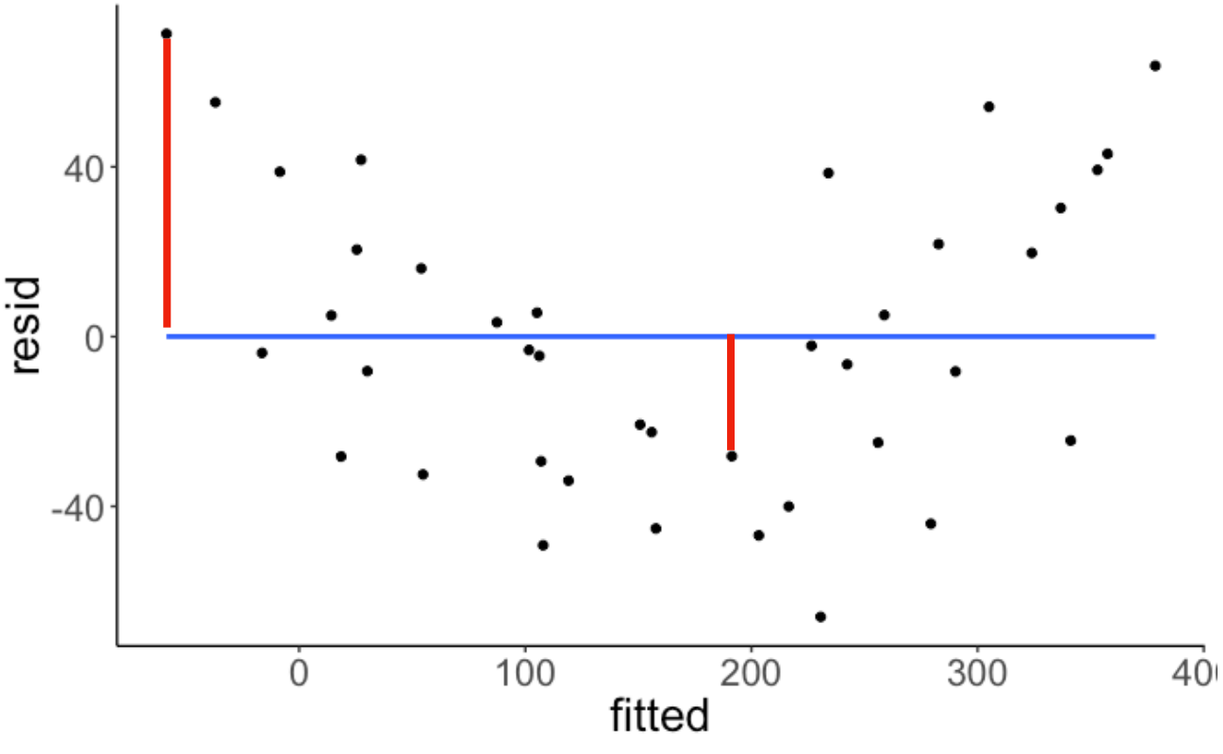
```
lm(formula = growth ~ 1 + time, data = df.growth)
```




**Residual plot**

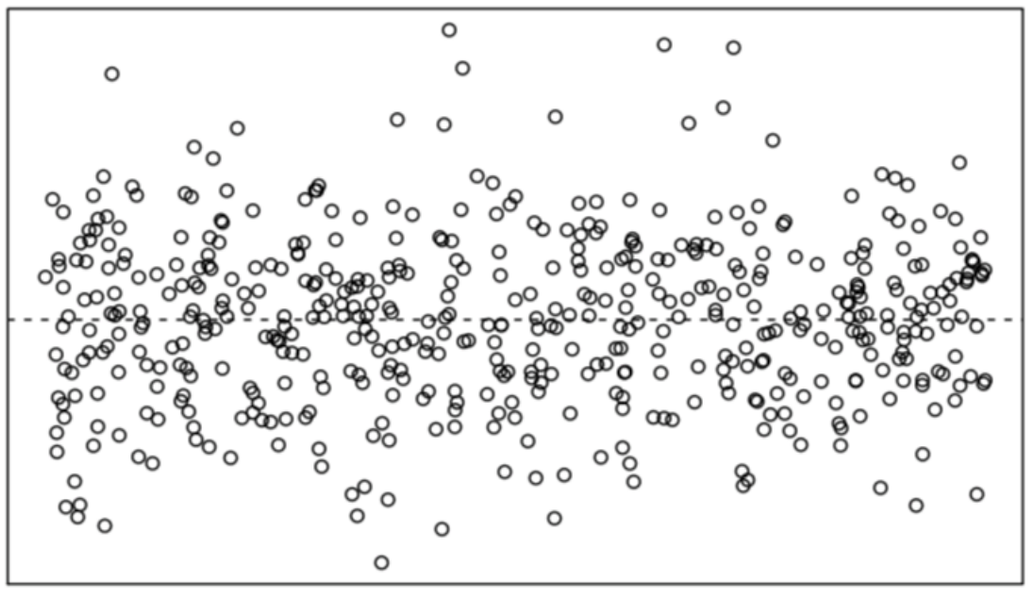
# Distribution of errors

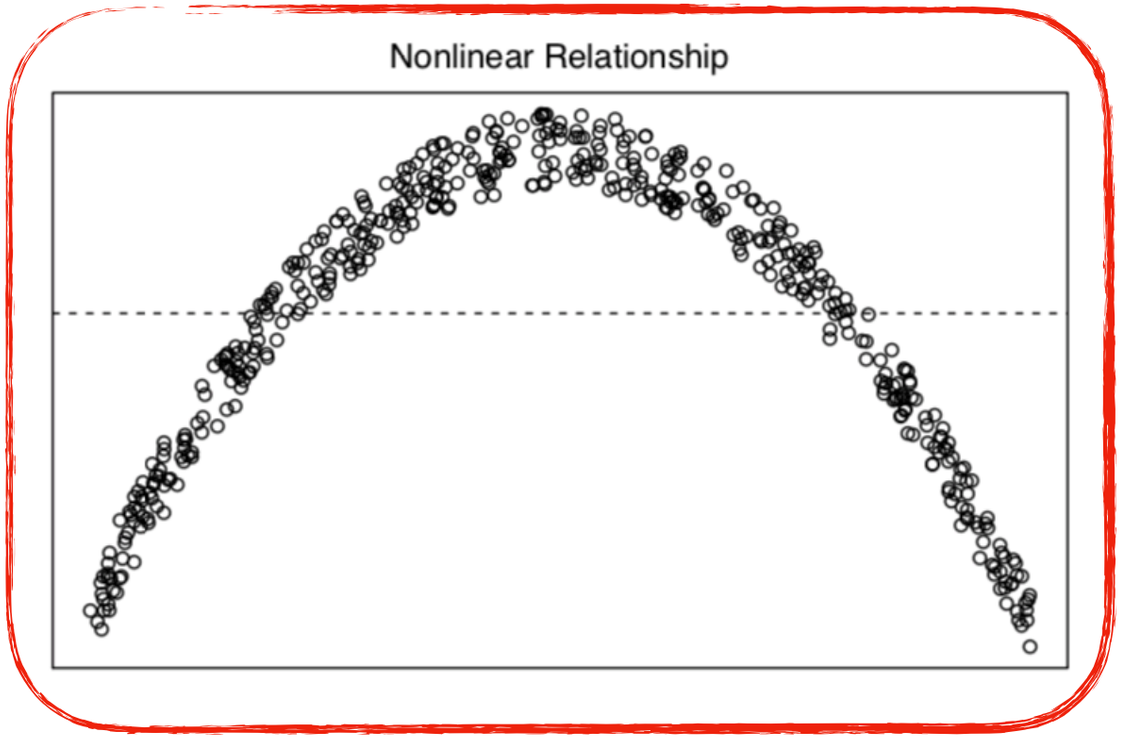


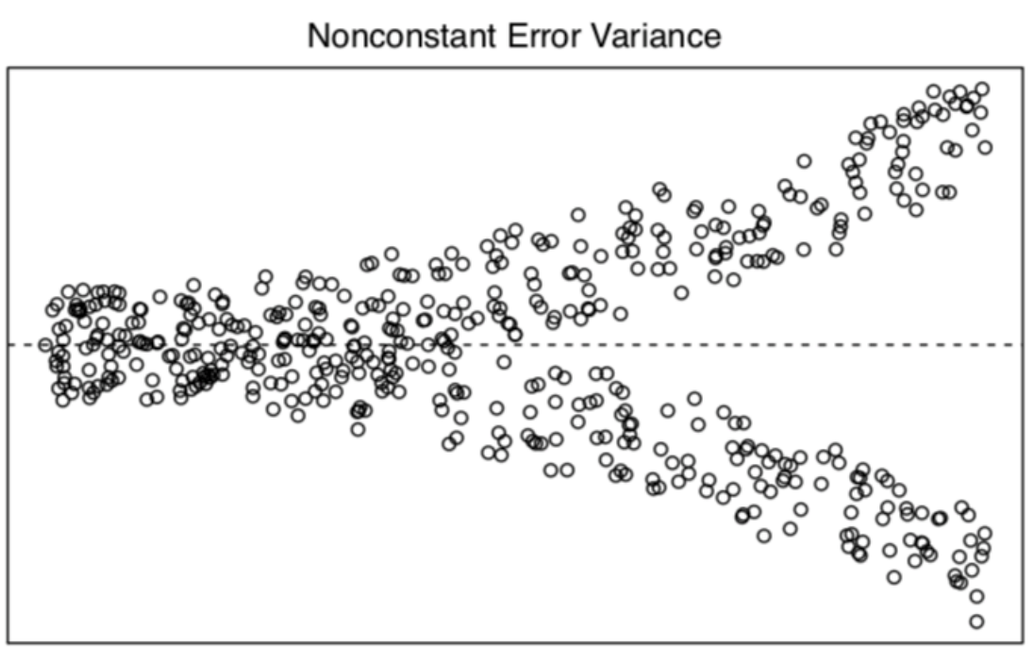



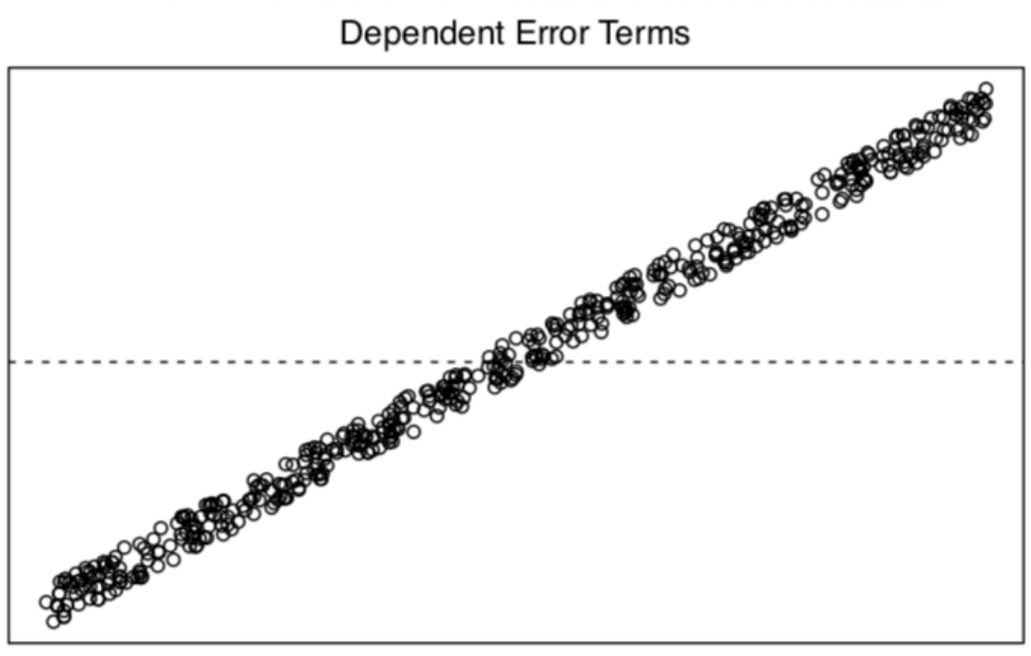

**Residual plot**

# Distribution of errors



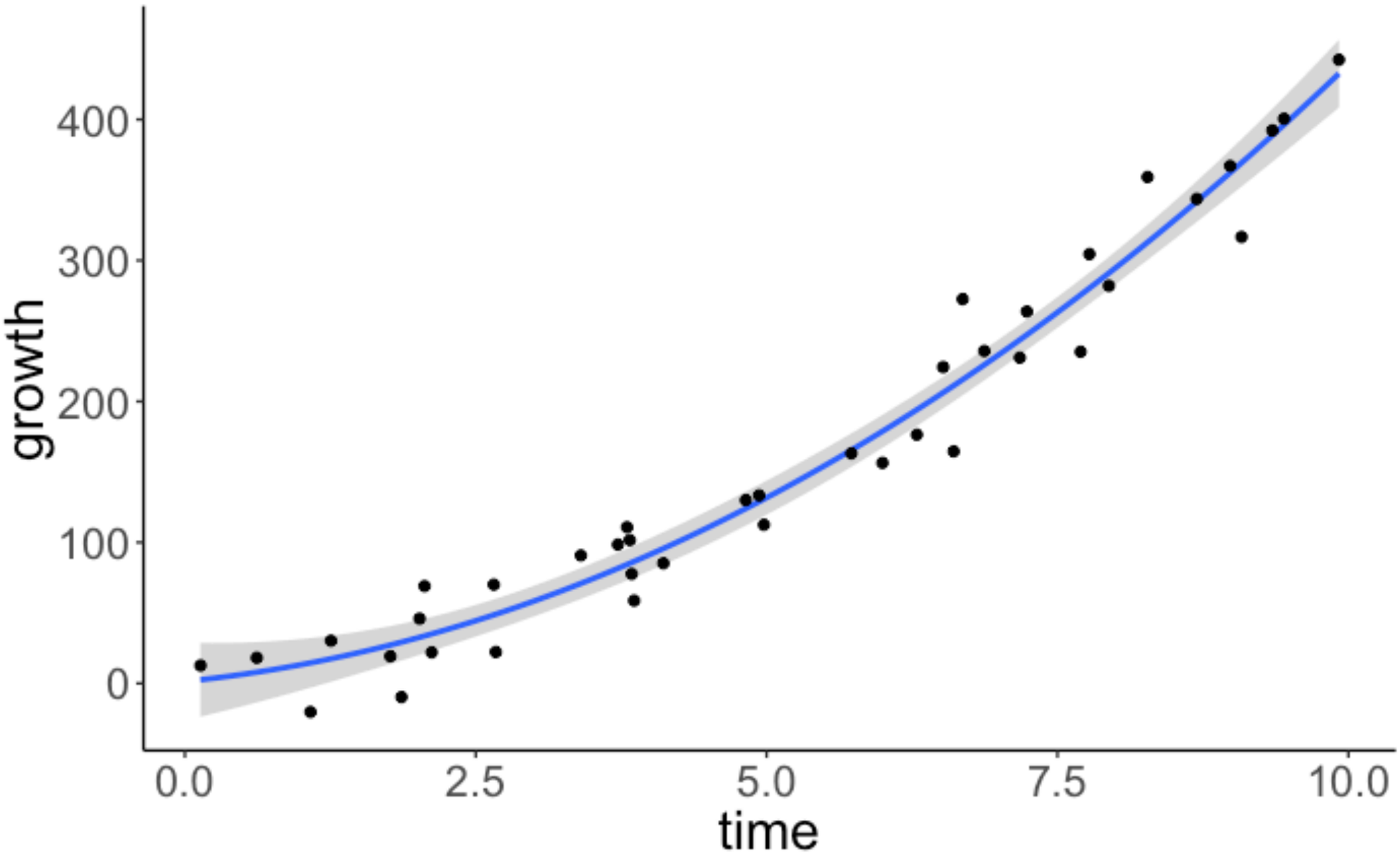



# Distribution of errors

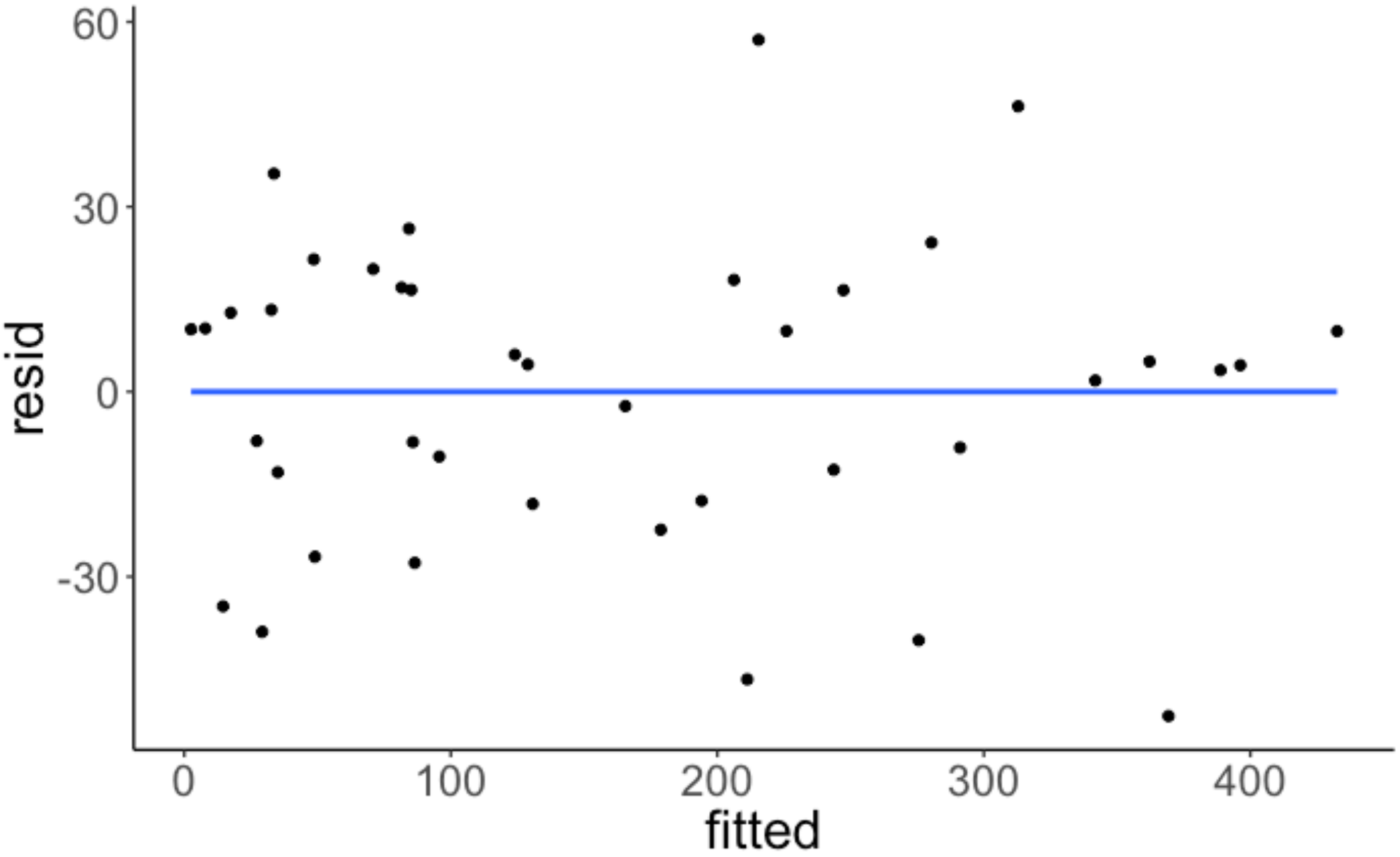
```
lm(formula = growth ~ 1 + time + I(time^2), data = df.growth)
```




**simulated data set**

# Distribution of errors

```
lm(formula = growth ~ 1 + time + I(time^2), data = df.growth)
```




**Residual plot**

# Distribution of errors

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.36      13.6065   0.100    0.921
time           8.39       6.0612   1.384    0.175
I(time^2)      3.54       0.5738   6.163 3.77e-07 ***
```

$$\widehat{\text{growth}}_i = b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2$$

**if time == 1:**

$$\widehat{\text{growth}}_i = b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2$$

# Distribution of errors

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.36      13.6065   0.100    0.921
time           8.39       6.0612   1.384    0.175
I(time^2)      3.54       0.5738   6.163 3.77e-07 ***
```

$$\widehat{\text{growth}}_i = b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2$$

**if time == 7:**

$$\widehat{\text{growth}}_i = b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2$$

# Simpsons paradox

# Simpson's paradox

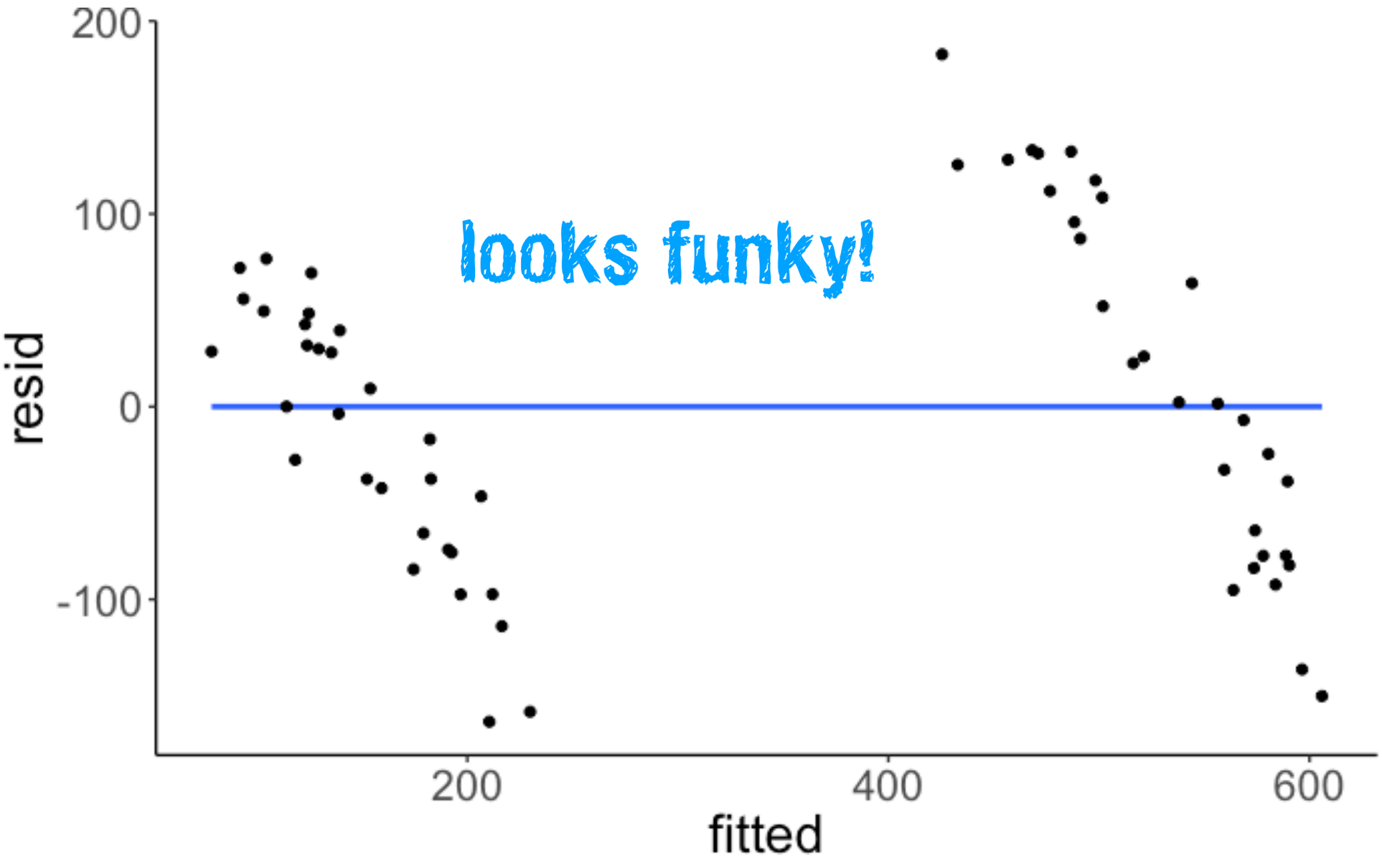
```
1 lm(formula = y ~ x, data = df.simpsons) %>%
2   summary()
```

```
Call:
lm(formula = y ~ x, data = df.simpson)

Residuals:
     Min       1Q   Median       3Q      Max
-155.558  -82.438   -9.204   76.255  201.910

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -96.2554    32.5768  -2.955  0.00452 **
x             7.1558     0.4808  14.883  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 99.23 on 58 degrees of freedom
Multiple R-squared:  0.7925,    Adjusted R-squared:  0.7889
F-statistic: 221.5 on 1 and 58 DF,  p-value: < 2.2e-16
```

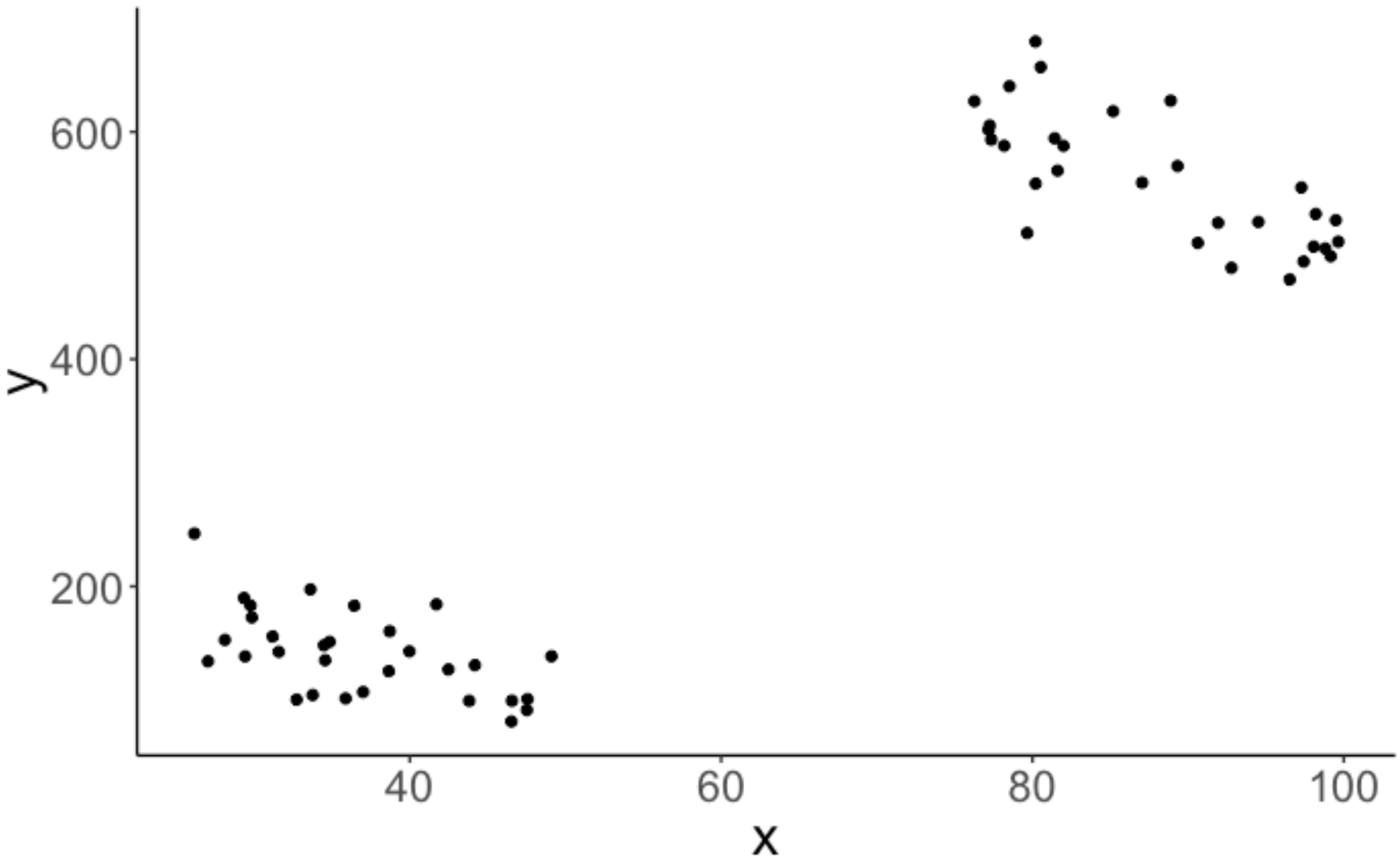
There is a significant positive relationship between x and y.

**you should take a look at the data first!**

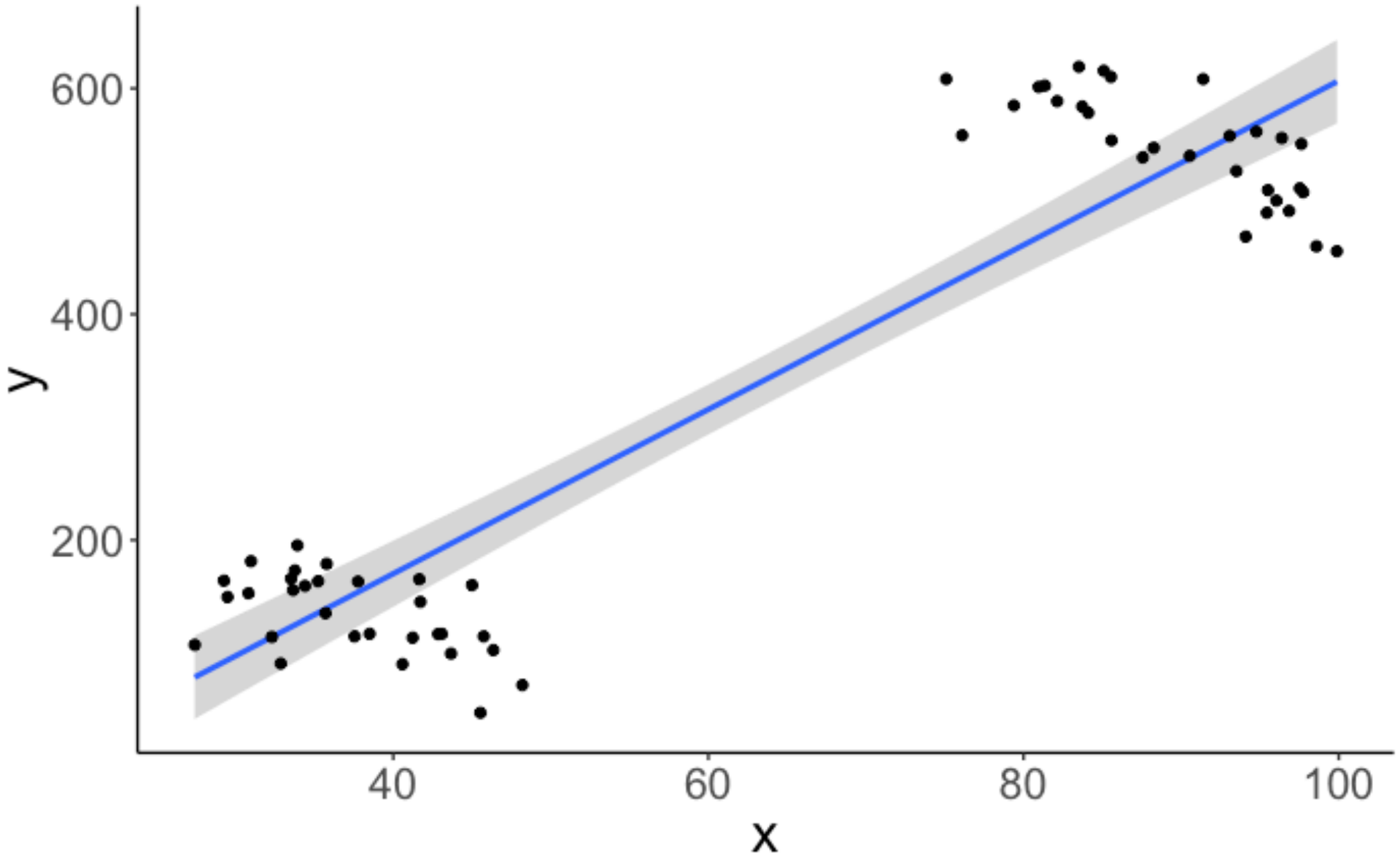
# Simpson's paradox



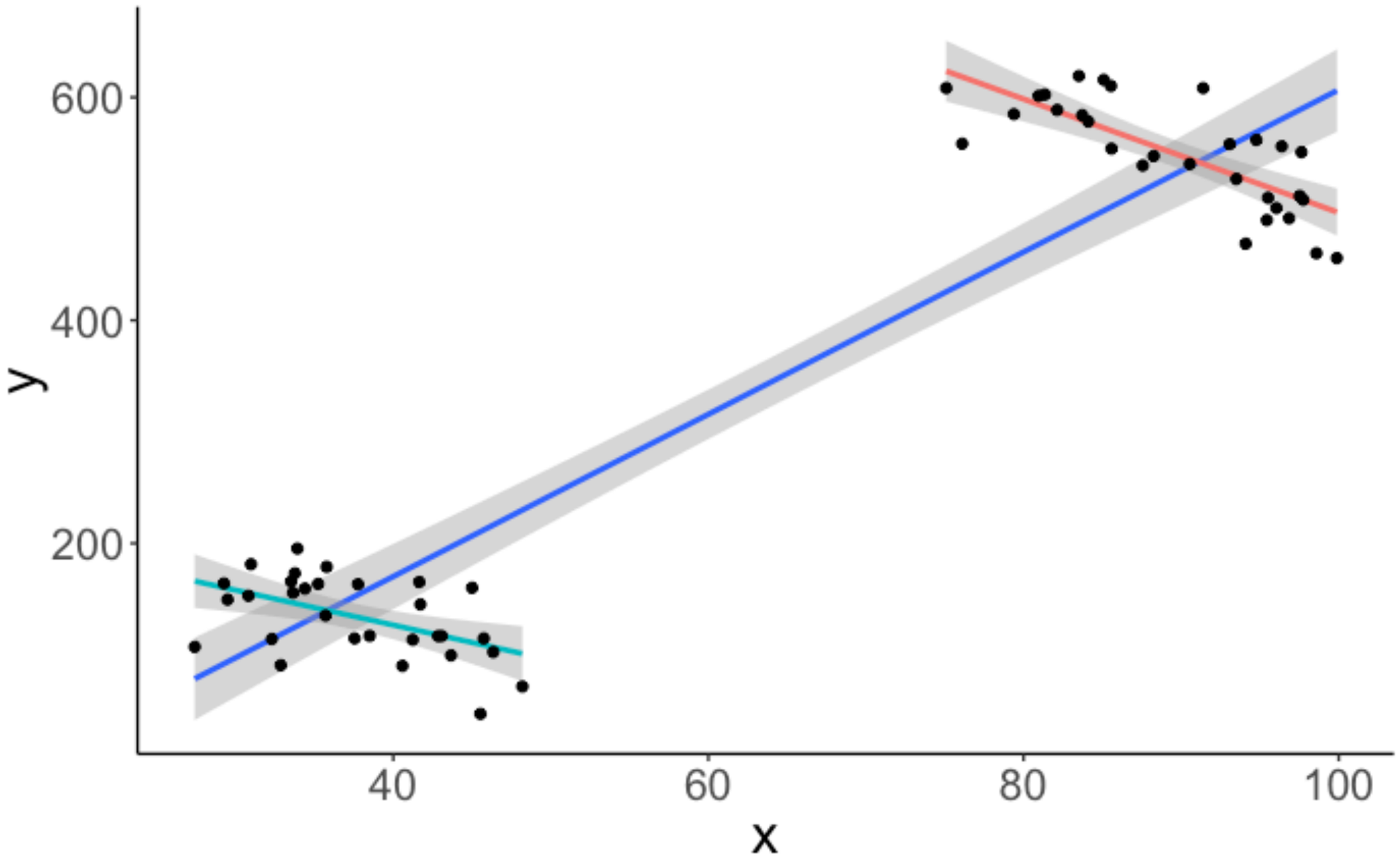
# Simpson's paradox



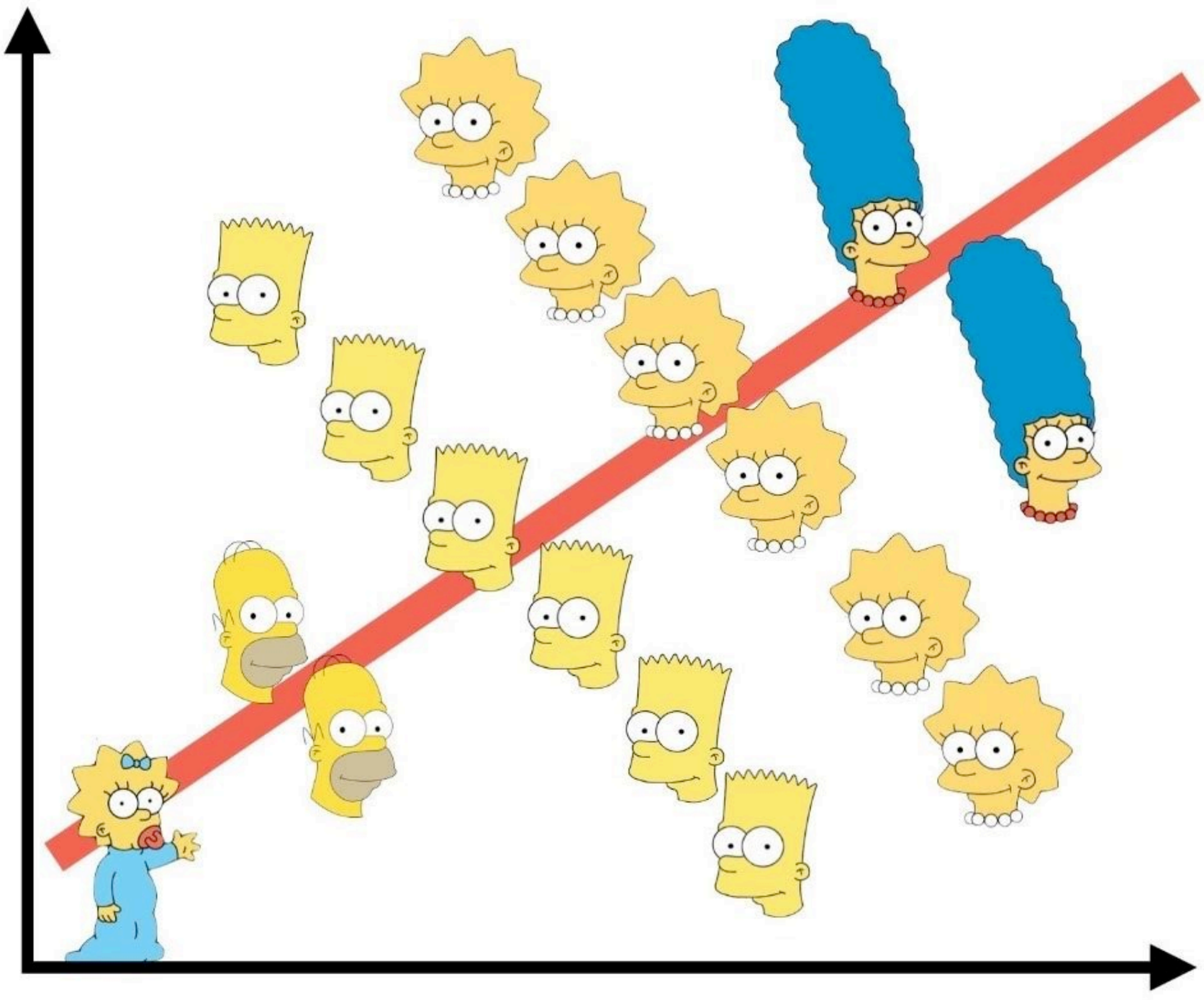
# Simpson's paradox



# Simpson's paradox



# Simpson's paradox



# Simpson's paradox

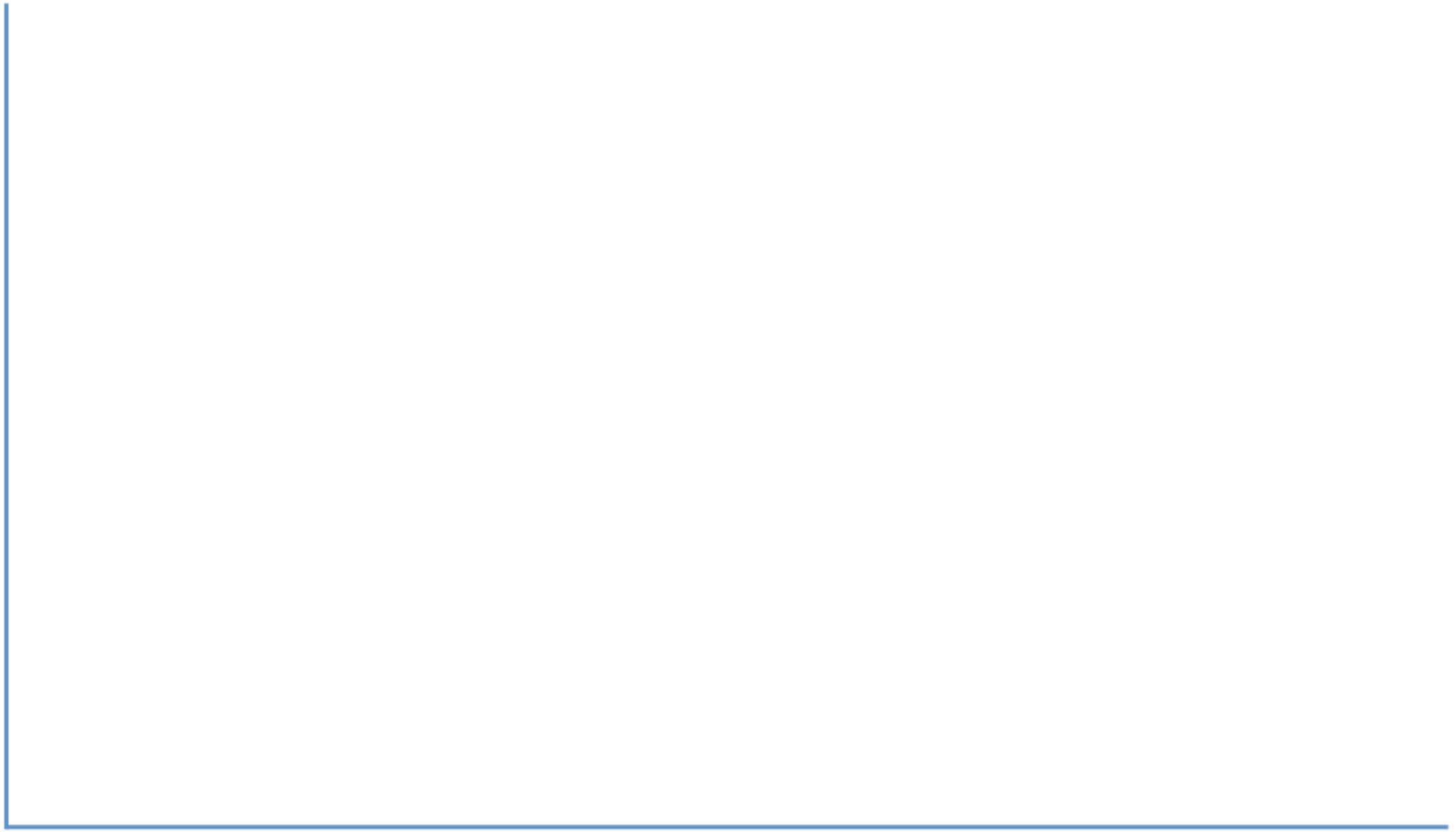
- an overall positive relationship might be negative when considered separately for each group (or vice versa)
- always look at the data first!
- whether the overall effect, or the within-group effect is more meaningful depends

# Summary

- ANOVA interpretation
- Contrasts
  - Planned comparisons
- Coding schemes
  - Dummy Coding
  - Effect coding
- Linear model assumptions
  - how to check them
  - what to do if they are violated

# Feedback

# How was the pace of today's class?



# How happy were you with today's class overall?

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

# Thank you!