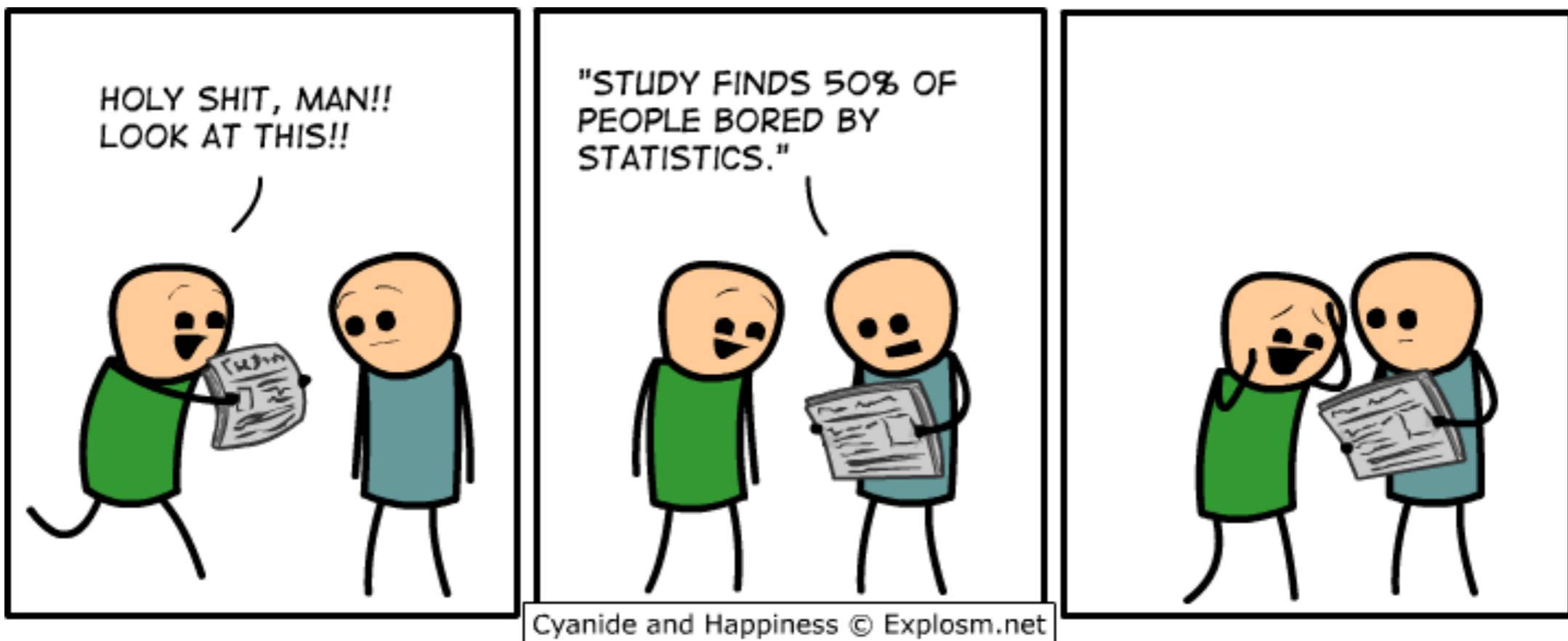


Linear model 4



02/06/2019

Logistics

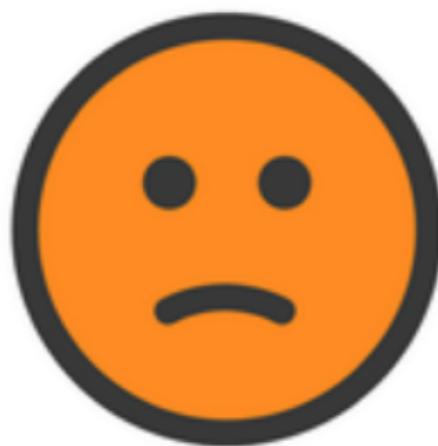
Start time

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?



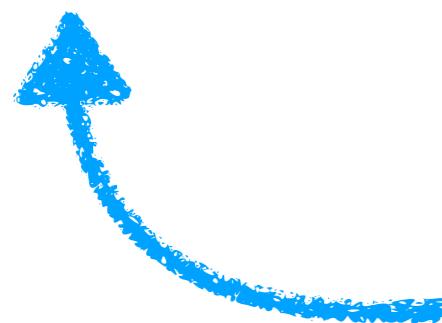
Midterm

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		

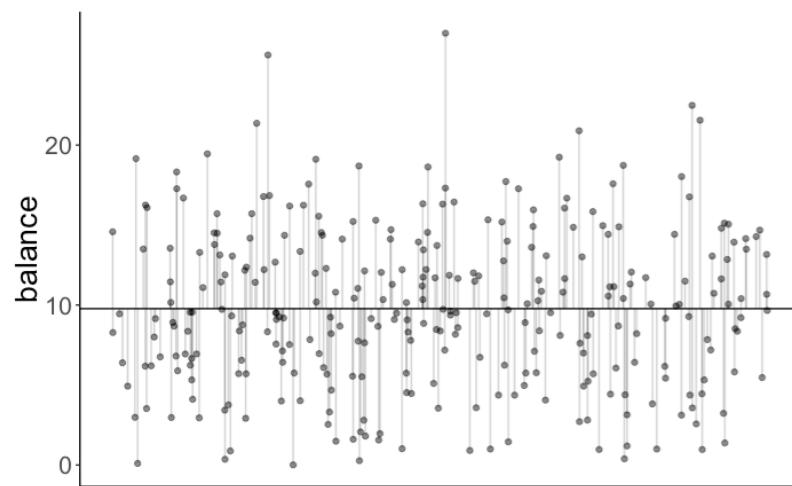
```
---
```

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

One-way ANOVA

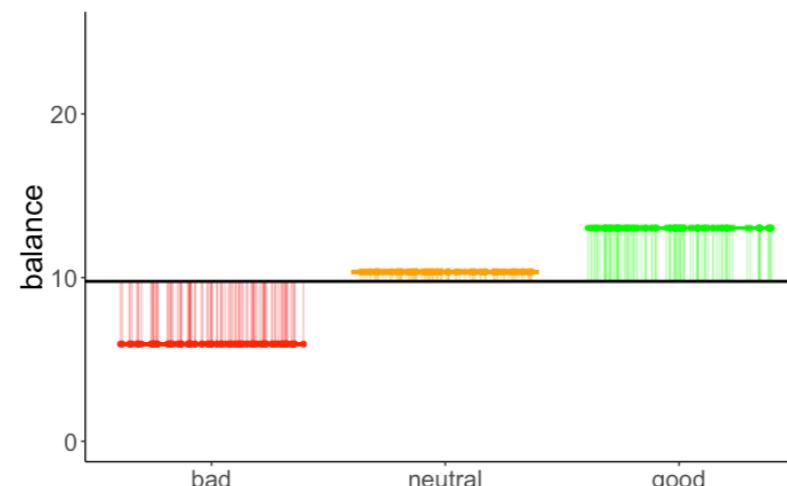
Variance decomposition

Total variance



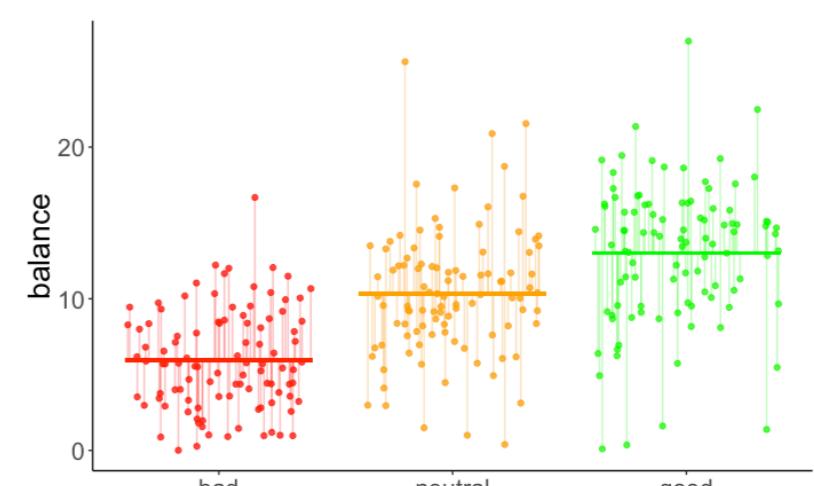
SS_{total}

Model variance



SS_{model}

Residual variance



SS_{residual}

<code>variance_total</code>	<code>variance_model</code>	<code>variance_residual</code>
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)) %>%
5   ungroup() %>%
6   summarize(variance_total = sum((balance - mean_grand)^2),
7             variance_model = sum((mean_group - mean_grand)^2),
8             variance_residual = variance_total - variance_model)
```

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

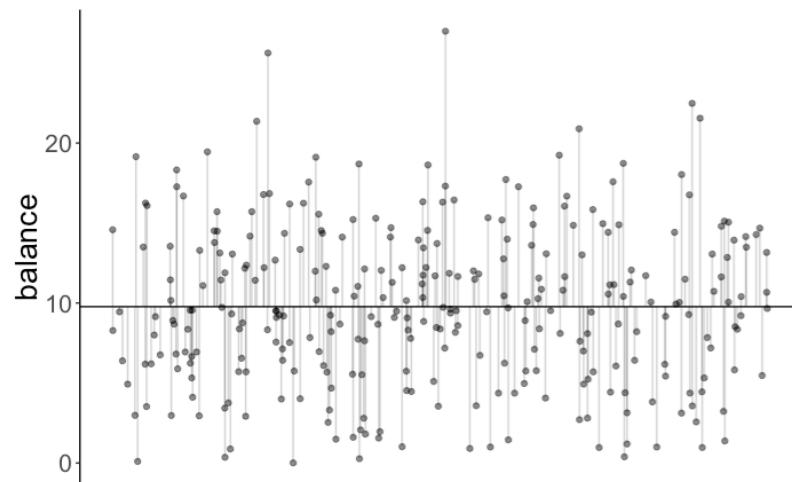
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
hand	2	2559.4	1279.70	76.0437	<2e-16 ***
skill	1	39.3	39.35	2.3383	0.1273
Residuals	296	4981.2	16.83		

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

Two-way ANOVA

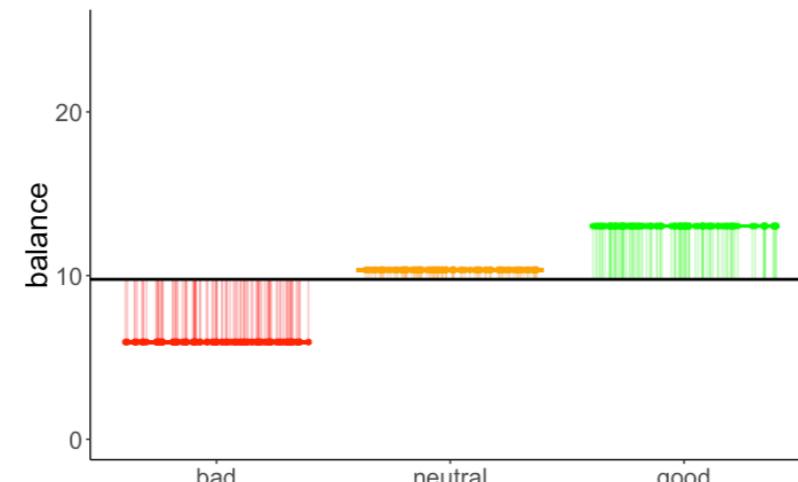
Variance decomposition

Total variance



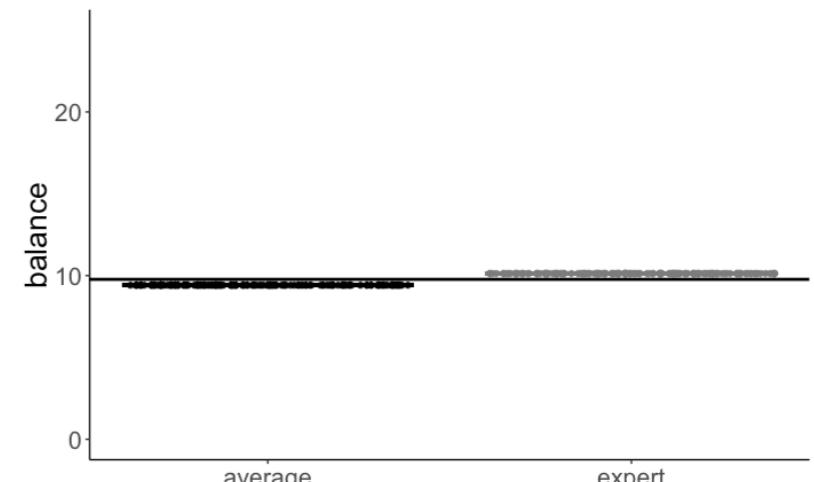
SSE_{total}

Hand variance



SSE_{hand}

Skill variance



SSE_{skill}

$$SSE_{\text{residual}} = SSE_{\text{total}} - SSE_{\text{hand}} - SSE_{\text{skill}}$$

Two-way ANOVA

```

1 df.poker %>%
2   mutate(mean_grand = mean(balance)) %>%
3   group_by(skill) %>%
4   mutate(mean_skill = mean(balance)) %>%
5   group_by(hand) %>%
6   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)	Signif. codes:
hand	2	2559.4	1279.70	76.0437	<2e-16 ***	0 '***'
skill	1	39.3	39.35	2.3383	0.1273	0.001 '**'
Residuals	296	4981.2	16.83			0.01 '*' 0.05 '.' 0.1 ' '
<hr/>						
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '						

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

```
1 library("car") ← load the "car" package  
2  
3 lm(formula = balance ~ hand + skill,  
4      data = df.poker,  
5      contrasts = c("contr.sum", "contr.poly")) %>%  
6 Anova(type = "3")
```

← 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)	2261.2	1	130.6038	<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

```
1 library("car")
2
3 lm(formula = balance ~ skill + hand,
4     data = df.poker,
5     contrasts = c("contr.sum", "contr.poly")) %>%
6 Anova(type = "3")
```

now the order doesn't matter ...

Anova Table (Type III tests)

Response: balance

	Sum Sq	Df	F value	Pr (>F)	
(Intercept)	2261.2	1	130.6038	<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				

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?



bad vs. neutral

neutral vs. good

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

Contrasts

```
1 df.poker = df.poker %>%
2   mutate(hand_contrast = factor(hand,
3                                 levels = c("bad", "neutral", "good"),
4                                 labels = c(-1, 0, 1)),
5   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

```
1 df.poker = df.poker %>%
2   mutate(hand_contrast = factor(hand,
3                                 levels = c("bad", "neutral", "good"),
4                                 labels = c(-1, 0, 1)),
5   hand_contrast = hand_contrast %>% as.character() %>% as.numeric())
6
7 fit = lm(formula = balance ~ hand_contrast,
8         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

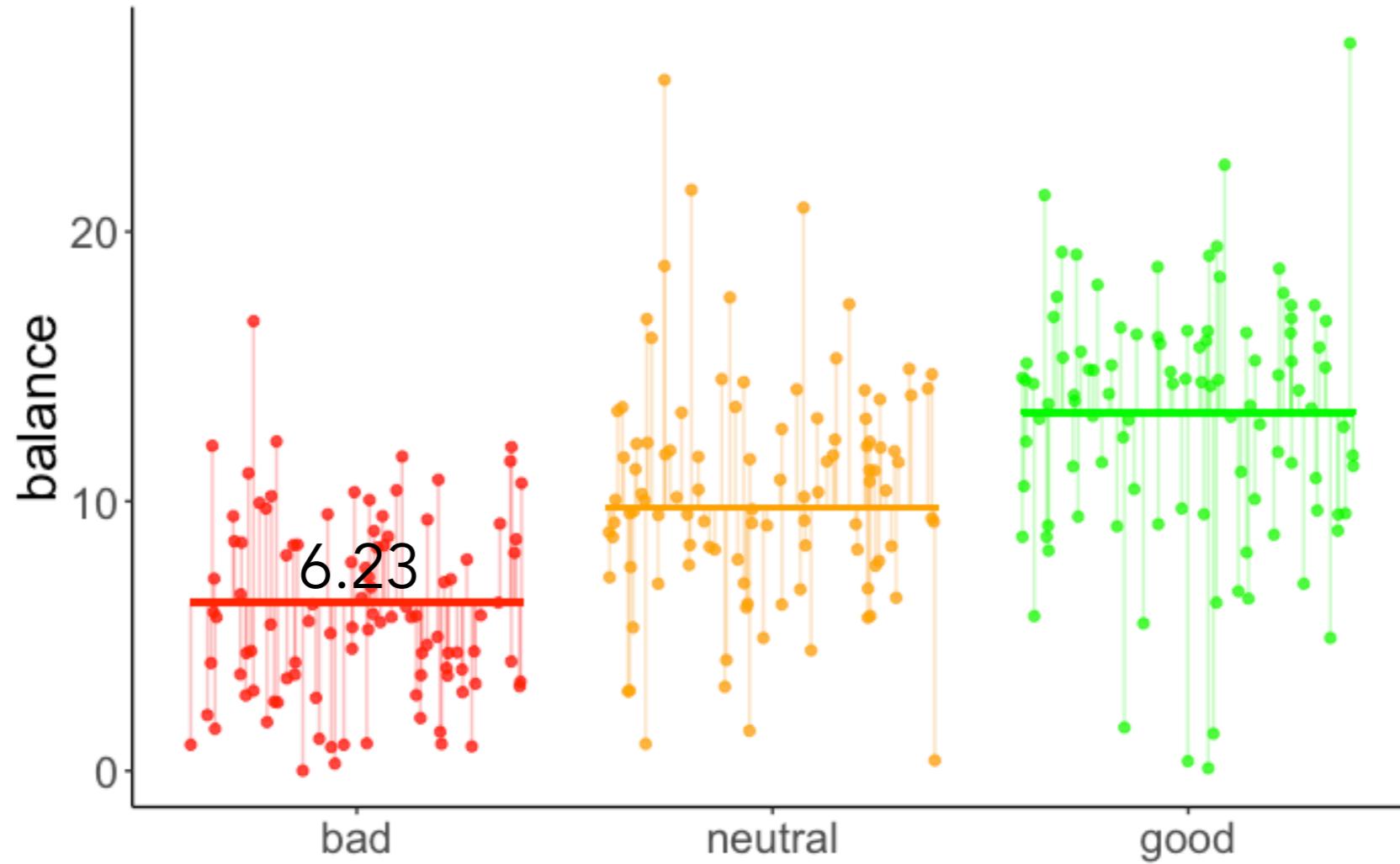
grand mean

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

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



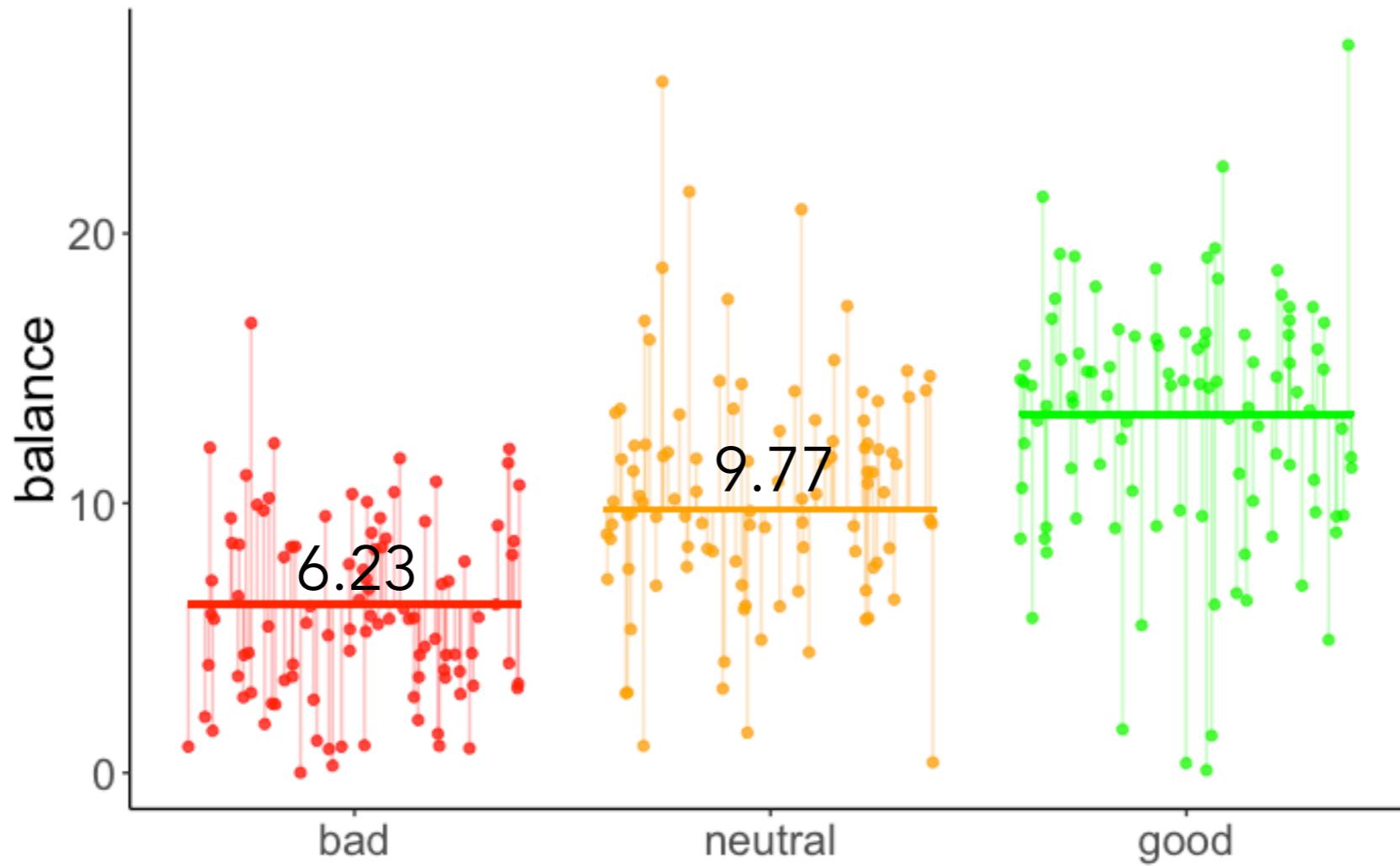
if `contrast == -1`

$$\begin{aligned}\widehat{\text{balance}}_i &= b_0 + b_1 \cdot \text{hand_contrast}_i \\ &= 9.77 + (-1) \cdot 3.54 = 6.23\end{aligned}$$

Contrasts

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

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



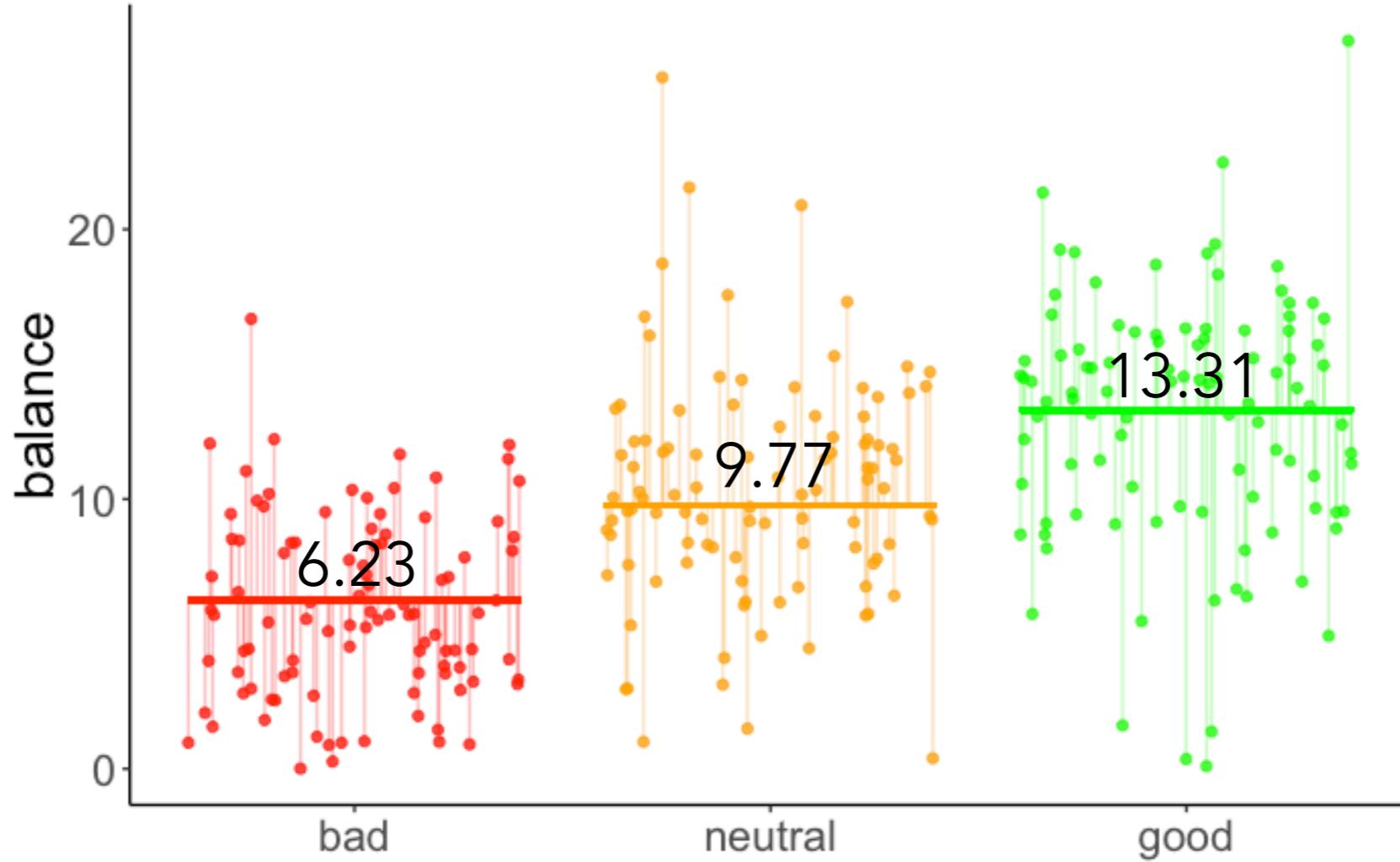
if $\text{contrast} == 0$

$$\begin{aligned}\widehat{\text{balance}}_i &= b_0 + b_1 \cdot \text{hand_contrast}_i \\ &= 9.77 + 0 \cdot 3.54 = 9.77\end{aligned}$$

Contrasts

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

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

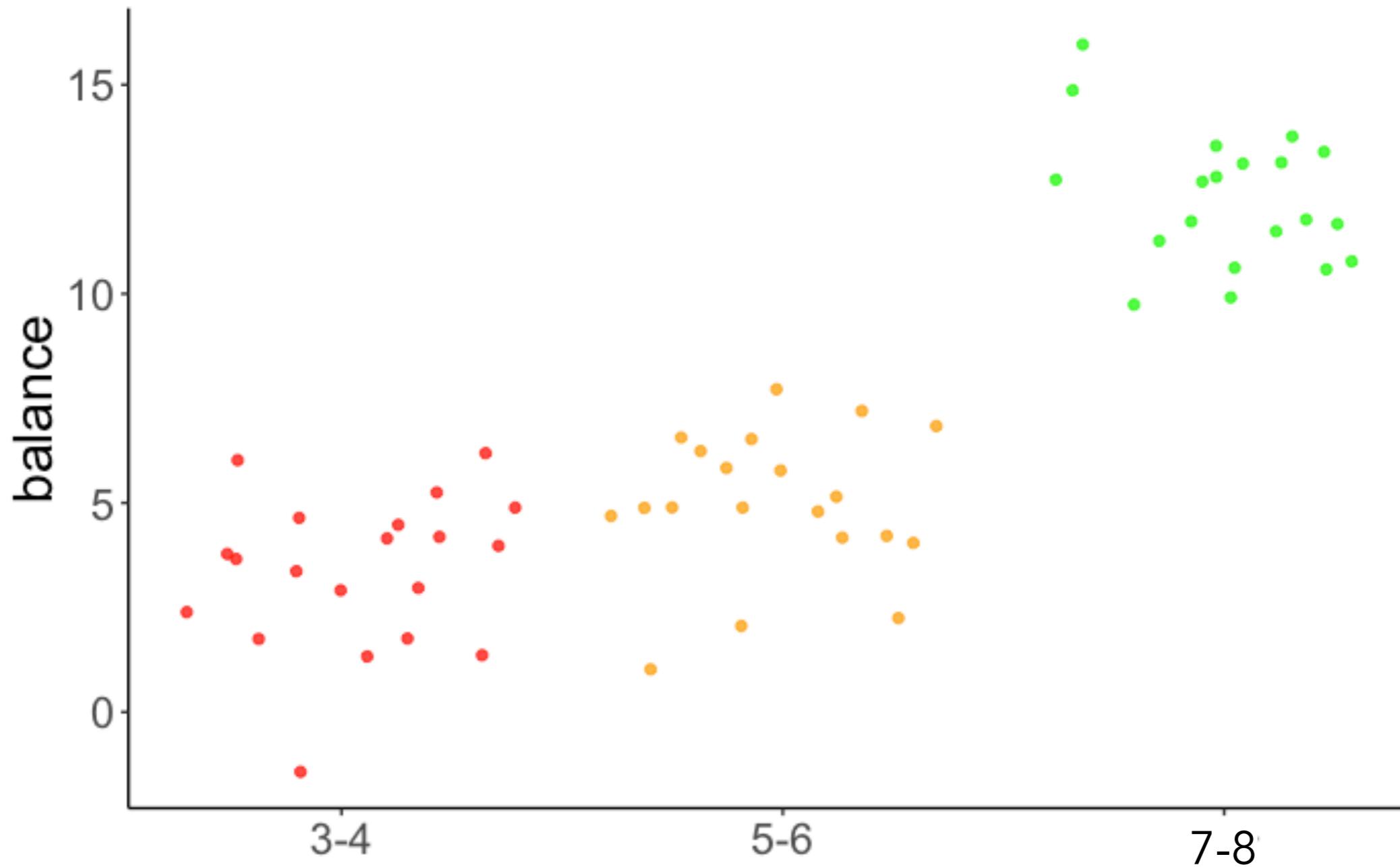


if `contrast == 1`

$$\begin{aligned}\widehat{\text{balance}}_i &= b_0 + b_1 \cdot \text{hand_contrast}_i \\ &= 9.77 + 1 \cdot 3.54 = 13.31\end{aligned}$$

Contrasts

Does performance increase with age?



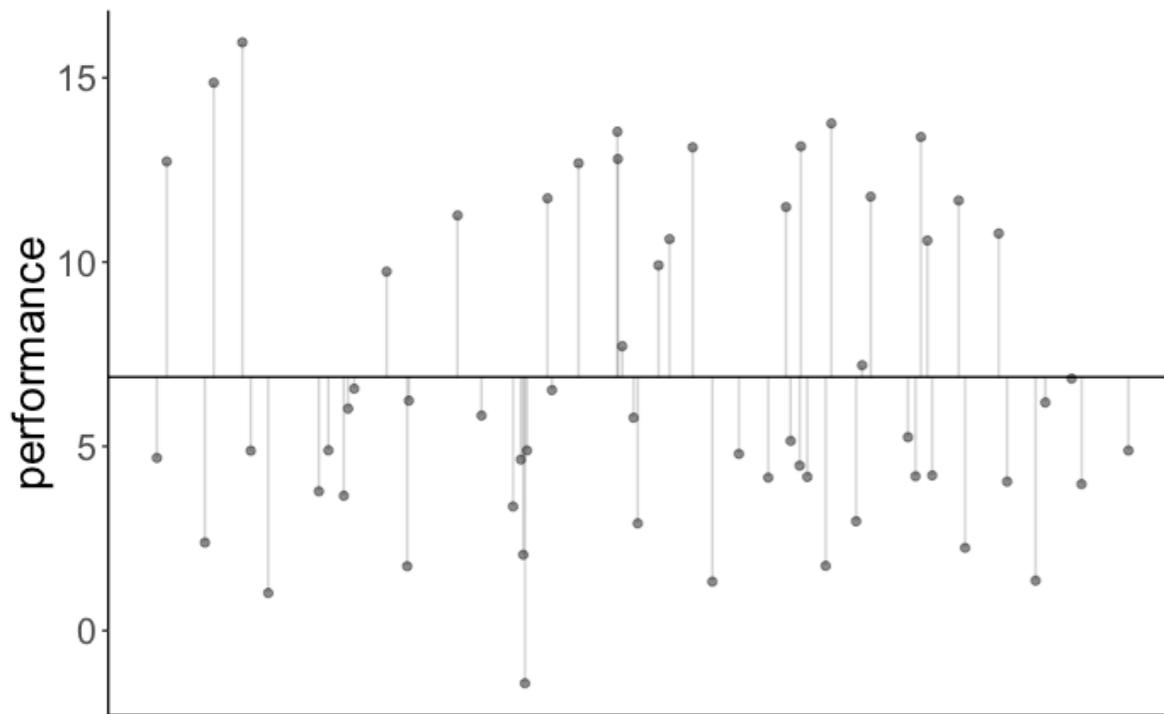
Data from a hypothetical developmental study

Contrasts

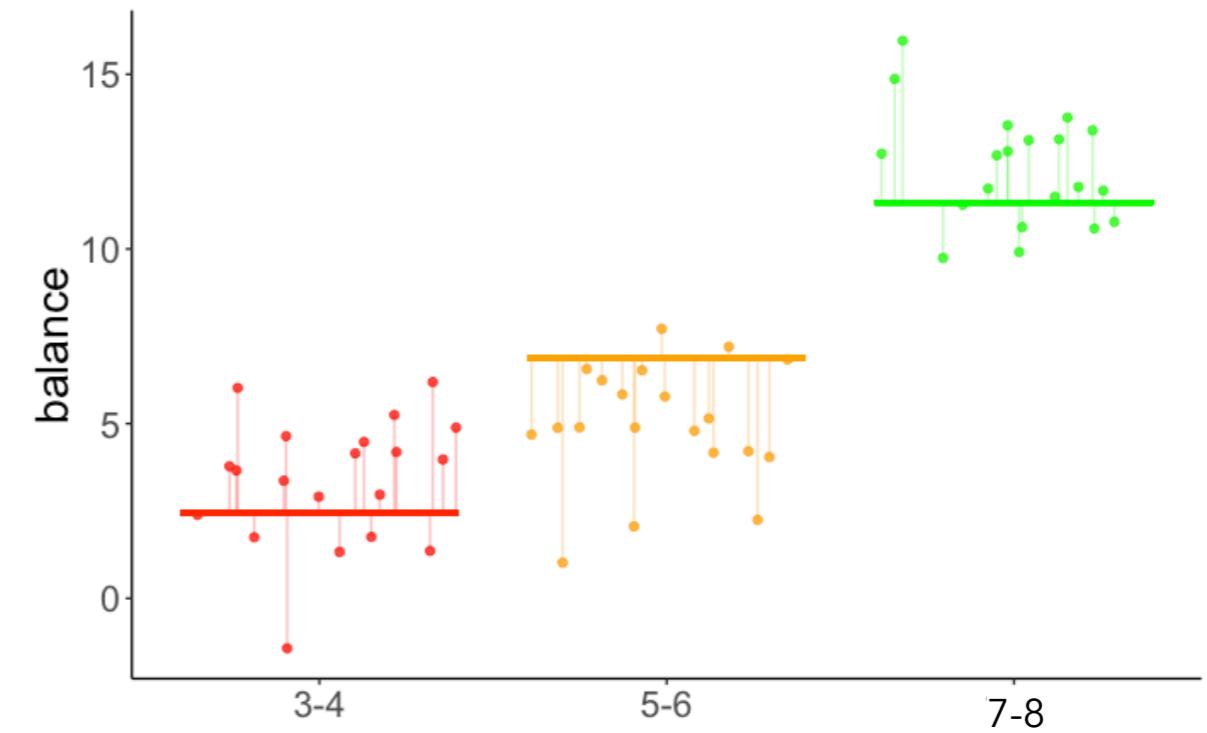
Does performance increase with age?

contrasts = c(-1, 0, 1)

Compact model



Augmented model

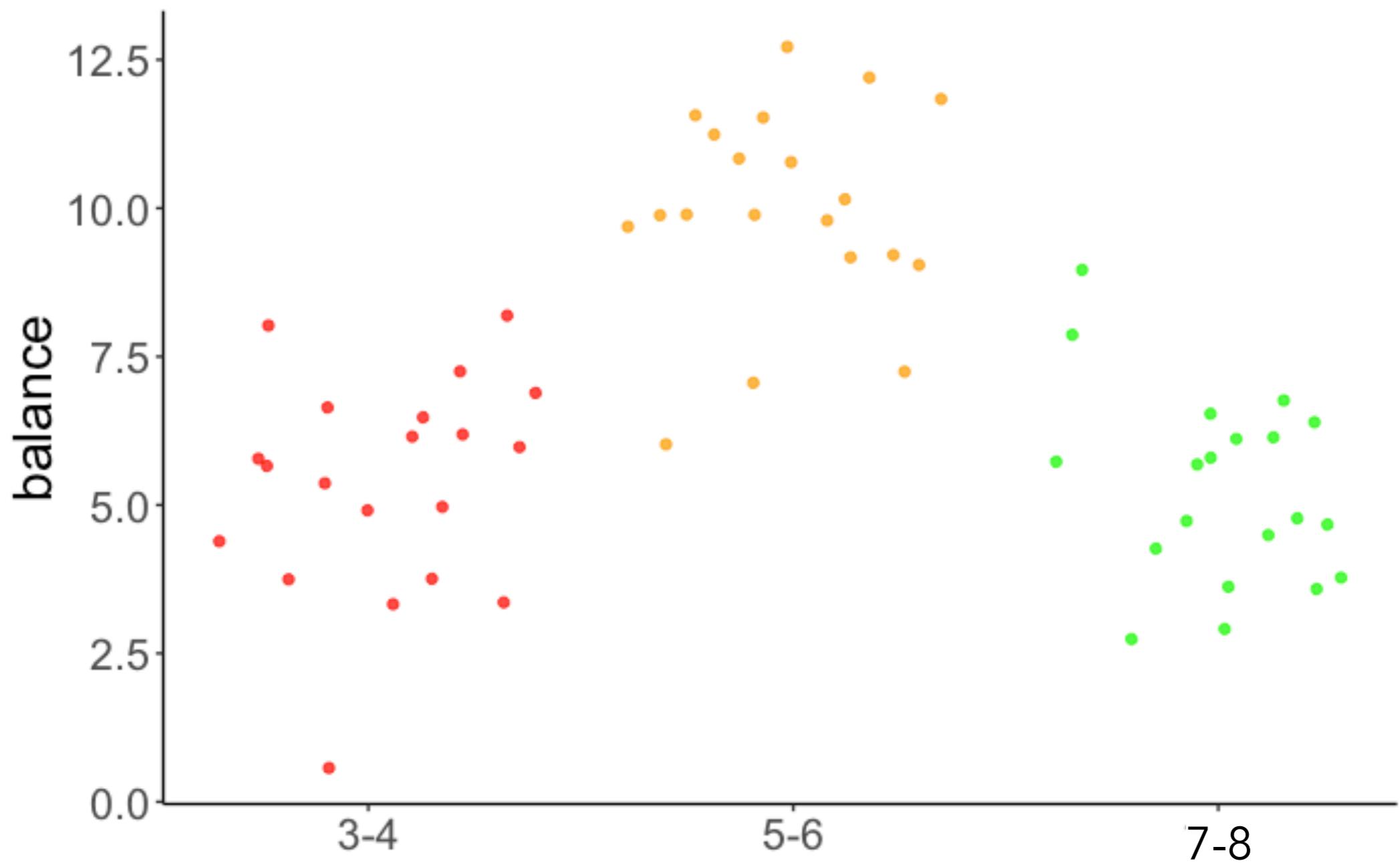


Model comparison

p < .001

Contrasts

Does performance increase with age?



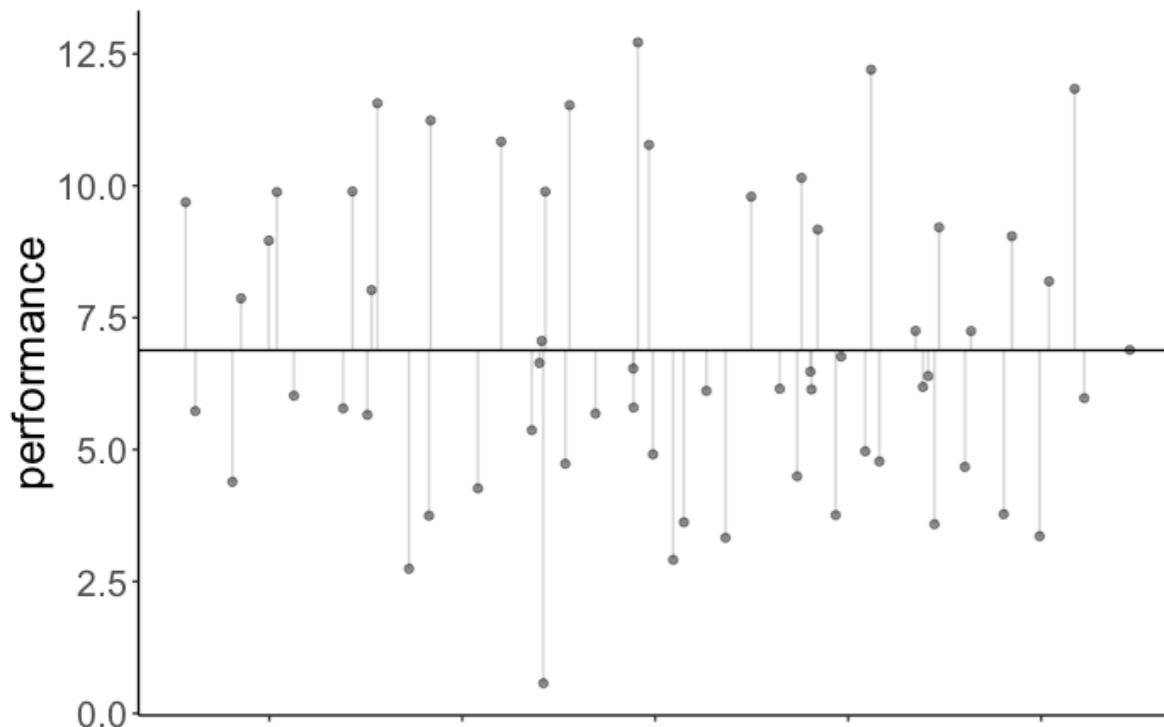
Data from another hypothetical developmental study

Contrasts

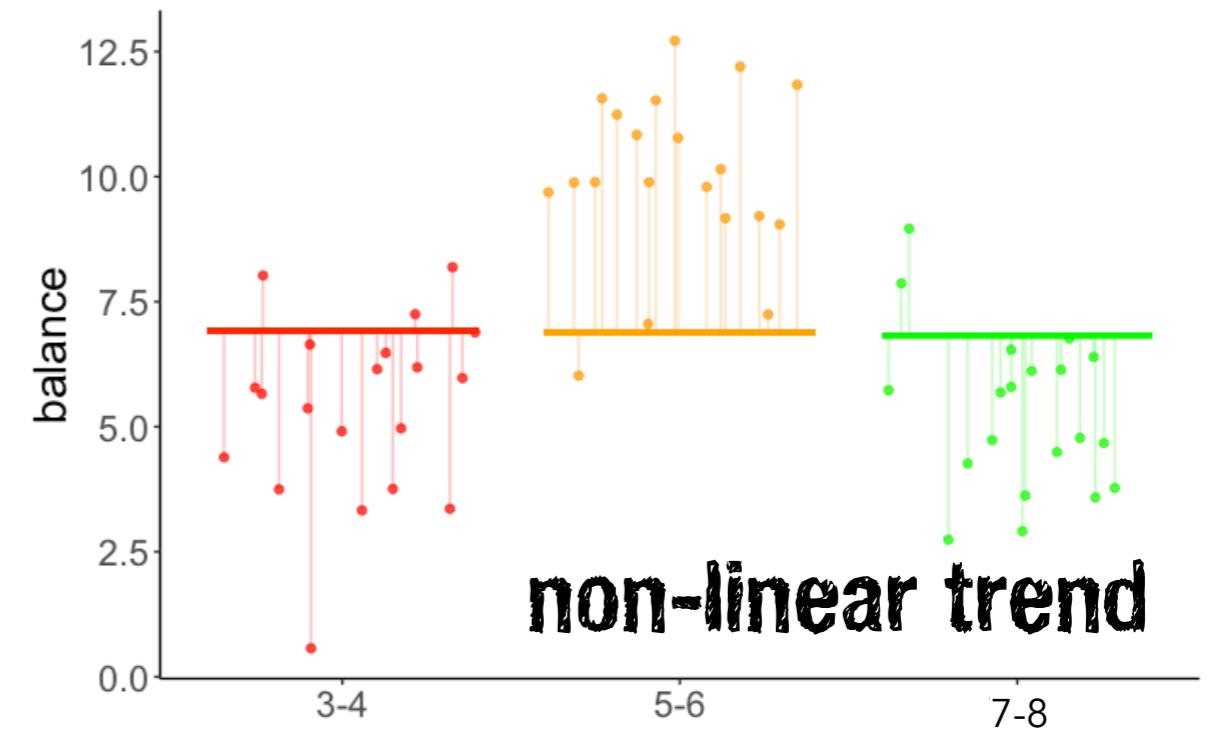
Does performance increase with age?

contrasts = c(-1, 0, 1)

Compact model



Augmented model



Model comparison

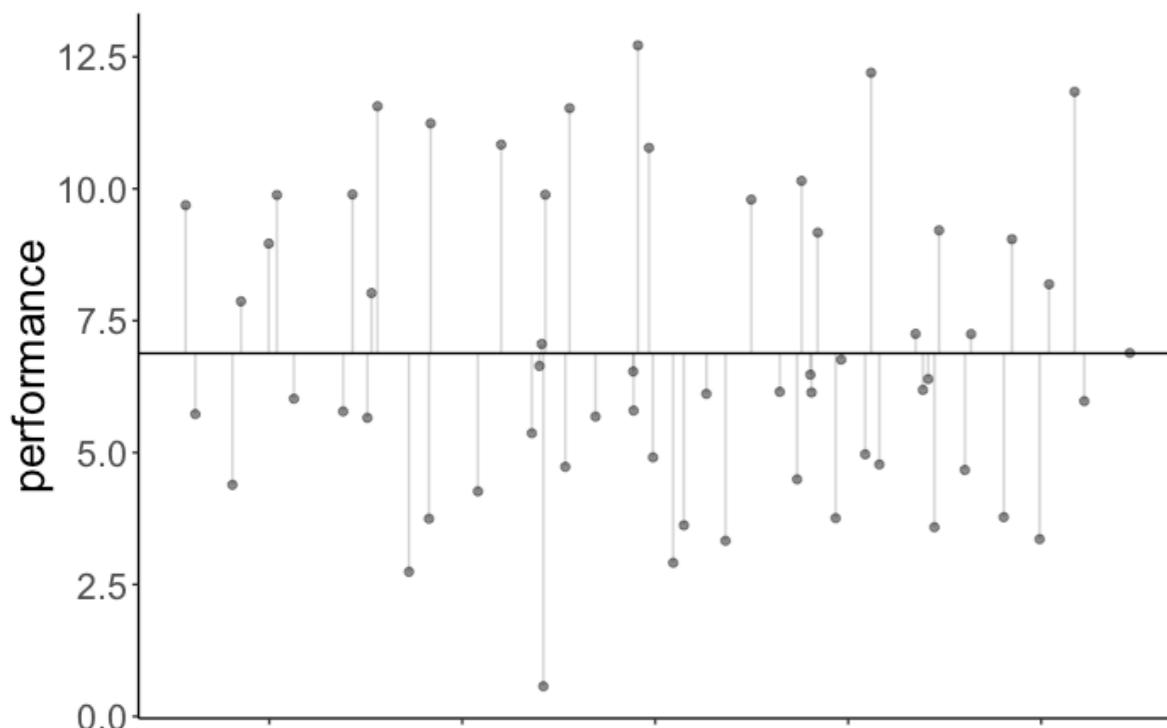
p = .8508

Contrasts

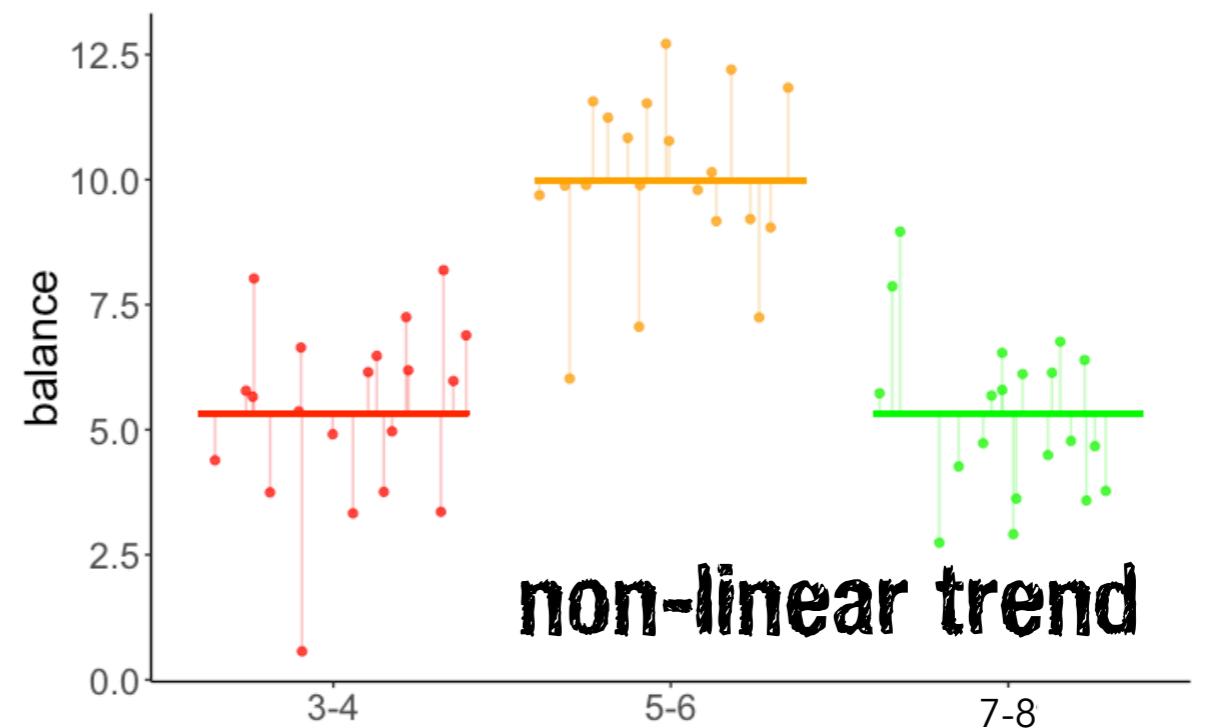
Does performance increase with age?

contrasts = c(1, 0, 1)

Compact model



Augmented model

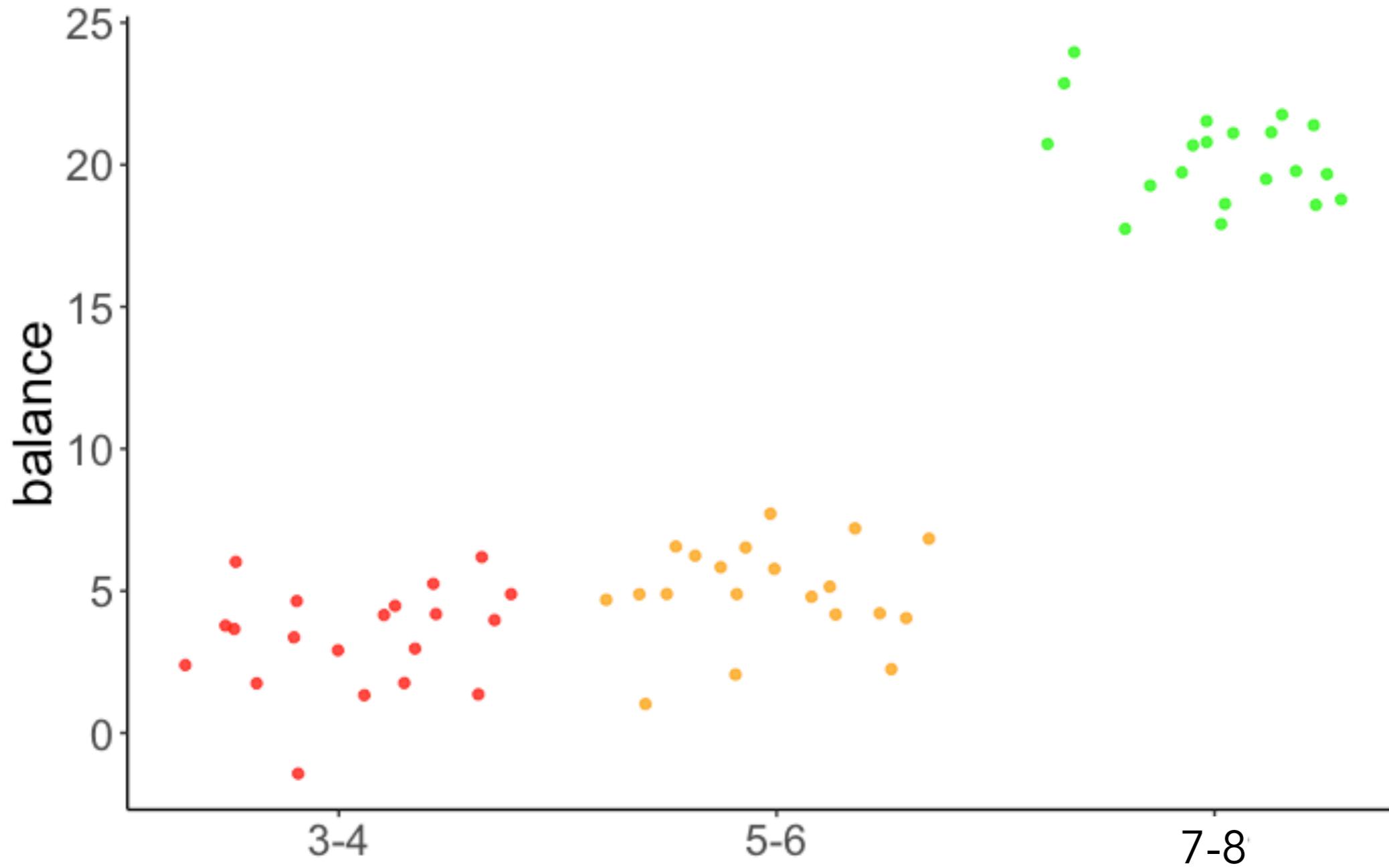


Model comparison

$p < .001$

Contrasts

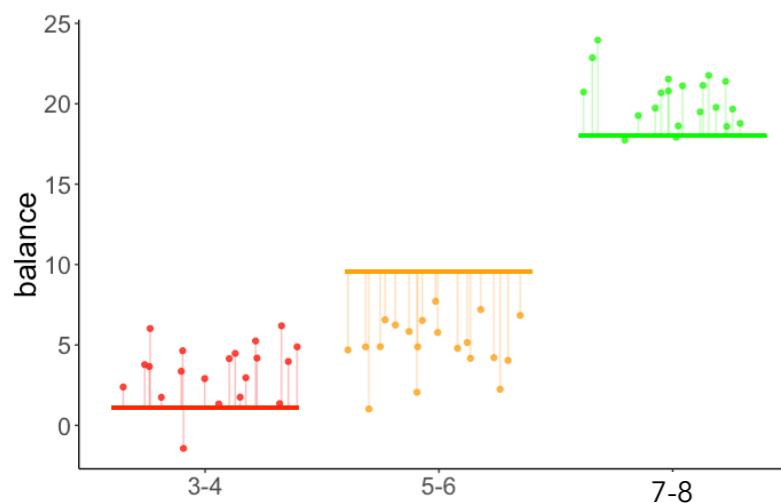
Does performance increase with age?



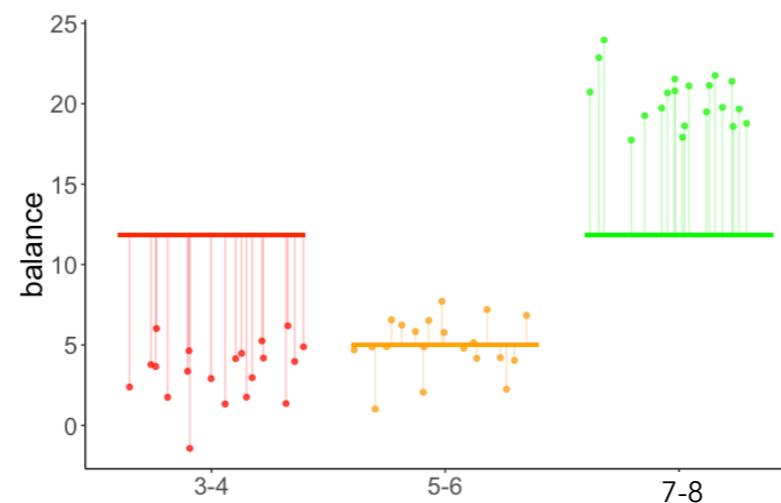
Data from yet another hypothetical developmental study

Contrasts

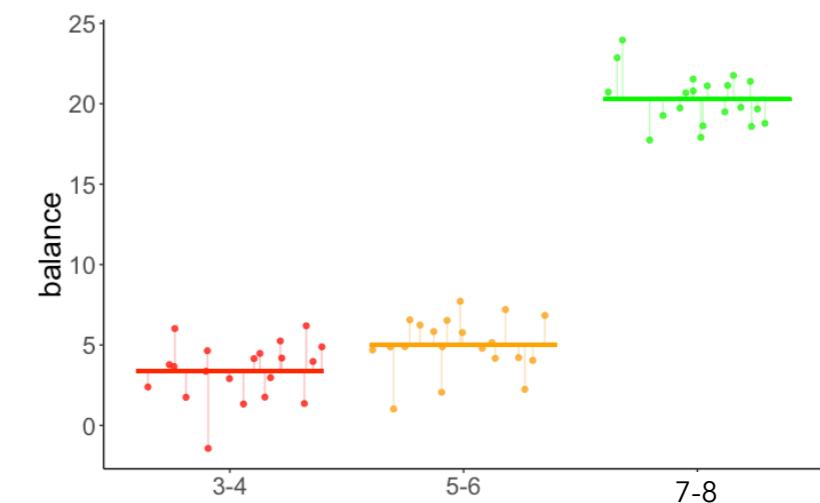
Linear contrast



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

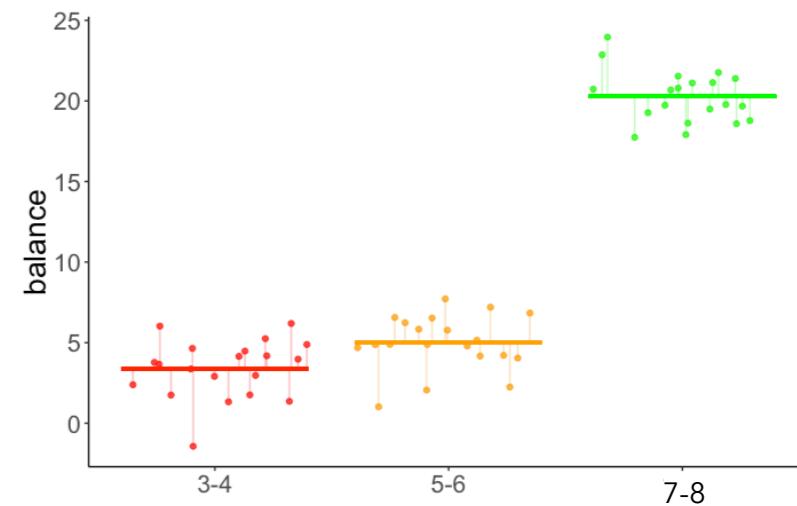
contrast	3-4	5-6	10+
group_contrast	-1	0	1
group_contrast2	1	0	1

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group_contrast}_i + b_2 \cdot \text{group_contrast2}_i$$

If group == "3-4"

$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group_contrast} + b_2 \cdot \text{group_contrast2}$$

Linear & Quadratic contrast



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

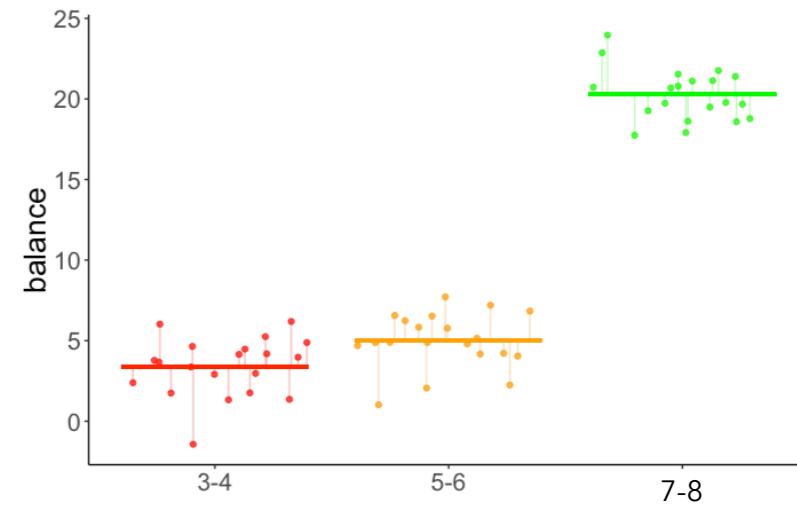
contrast	3-4	5-6	10+
group_contrast	-1	0	1
group_contrast2	1	0	1

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group_contrast}_i + b_2 \cdot \text{group_contrast2}_i$$

if group == "5-6"

$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group_contrast} + b_2 \cdot \text{group_contrast2}$$

Linear & Quadratic contrast



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

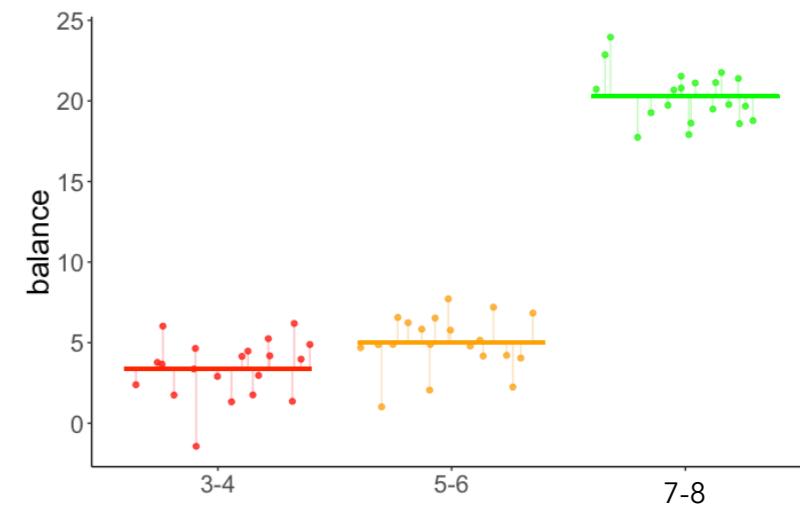
contrast	3-4	5-6	10+
group_contrast	-1	0	1
group_contrast2	1	0	1

$$\widehat{\text{balance}}_i = b_0 + b_1 \cdot \text{group_contrast}_i + b_2 \cdot \text{group_contrast2}_i$$

if group == "10+"

$$\widehat{\text{balance}} = b_0 + b_1 \cdot \text{group_contrast} + b_2 \cdot \text{group_contrast2}$$

Linear & Quadratic contrast

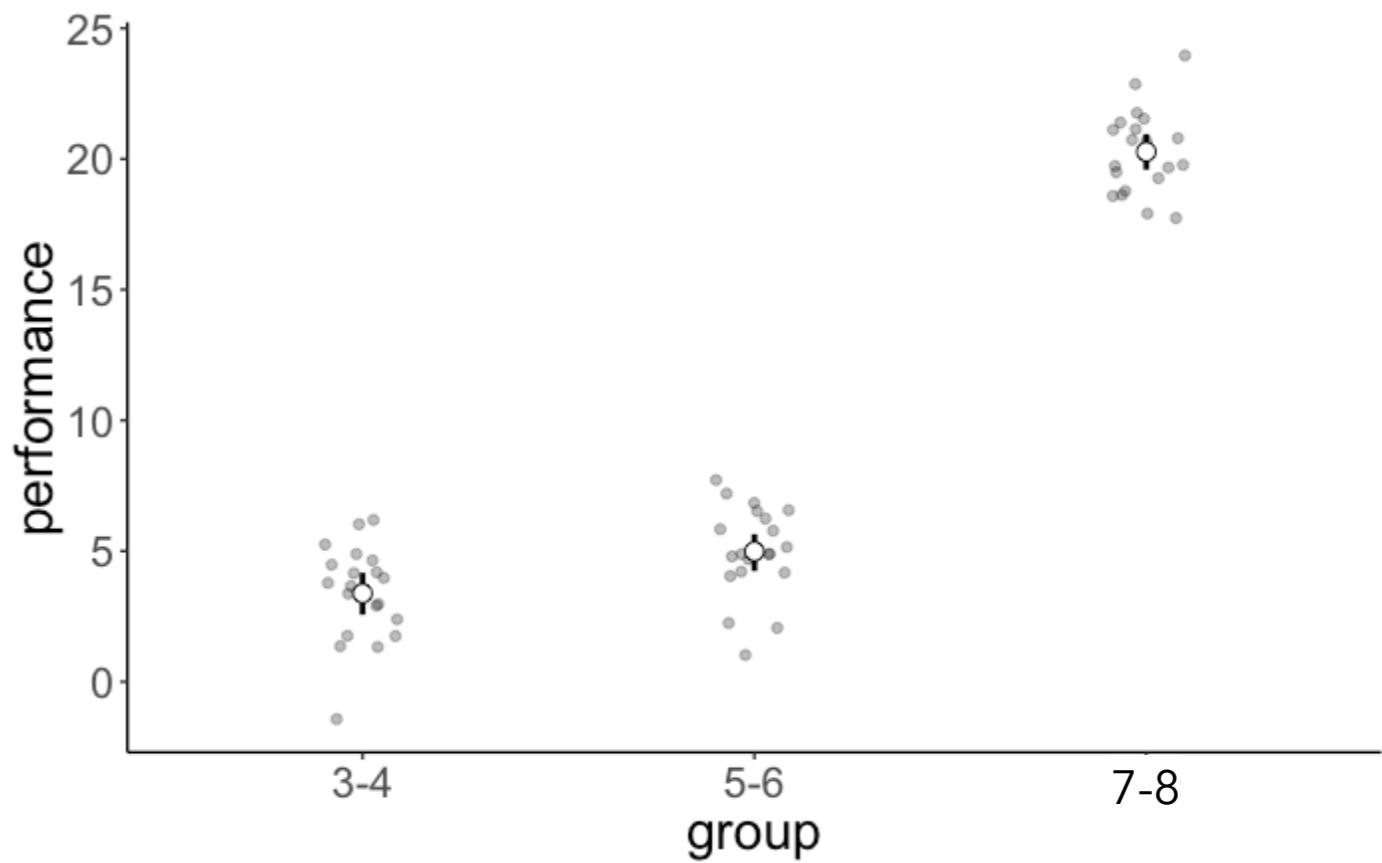


group	performance	group_contrast	group_contrast2
3-4	1.75	-1	1
3-4	3.37	-1	1
3-4	1.33	-1	1
5-6	6.84	0	0
5-6	6.56	0	0
5-6	5.15	0	0
10+	19.67	1	1
10+	19.49	1	1
10+	21.39	1	1

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 ...

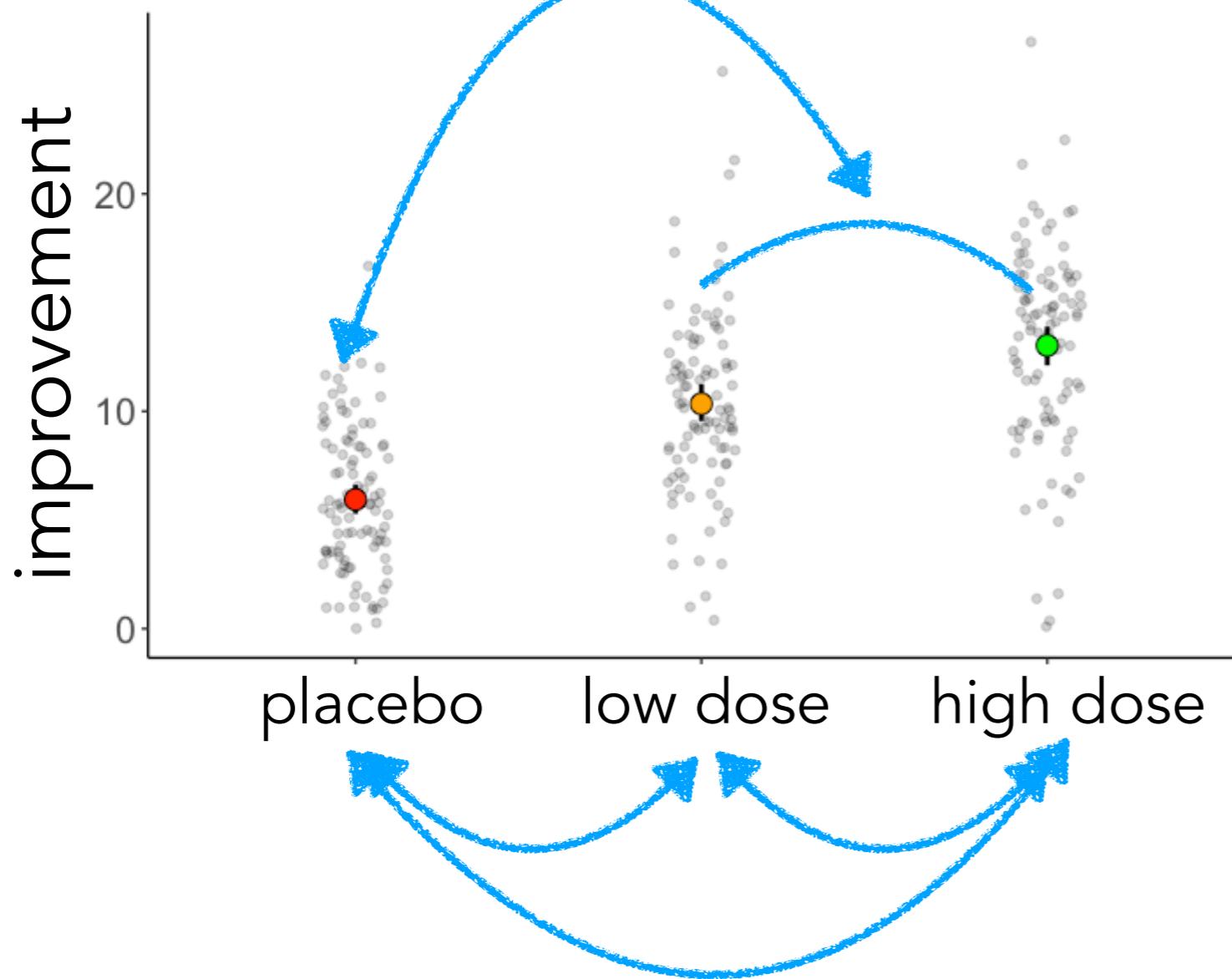
While the performance between 3-4 year olds, and 5-6 year olds did not differ significantly, 7-8 year olds performed significantly better than the other age groups.

Planned comparisons

Planned contrasts

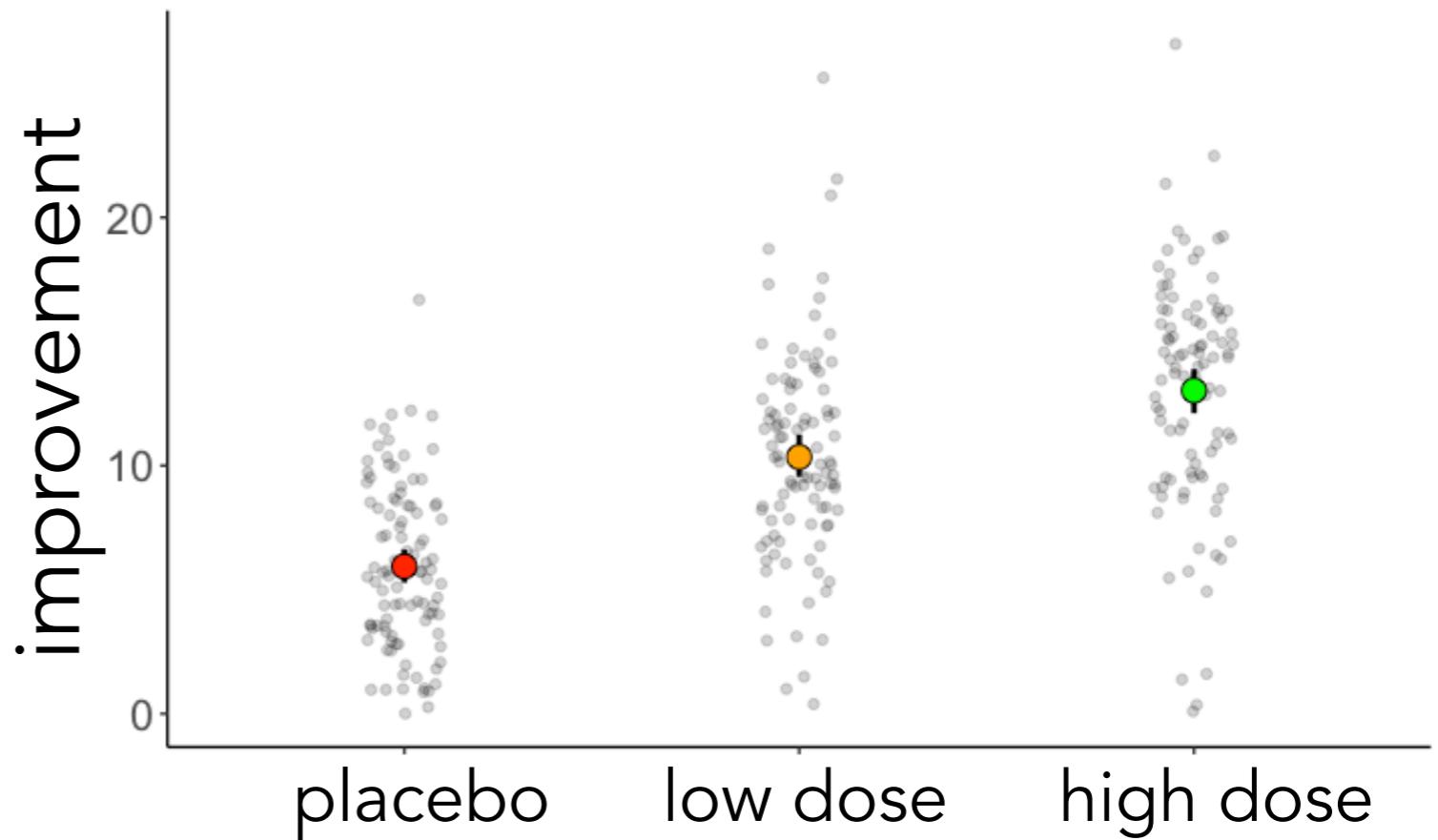
must be defined in advance

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

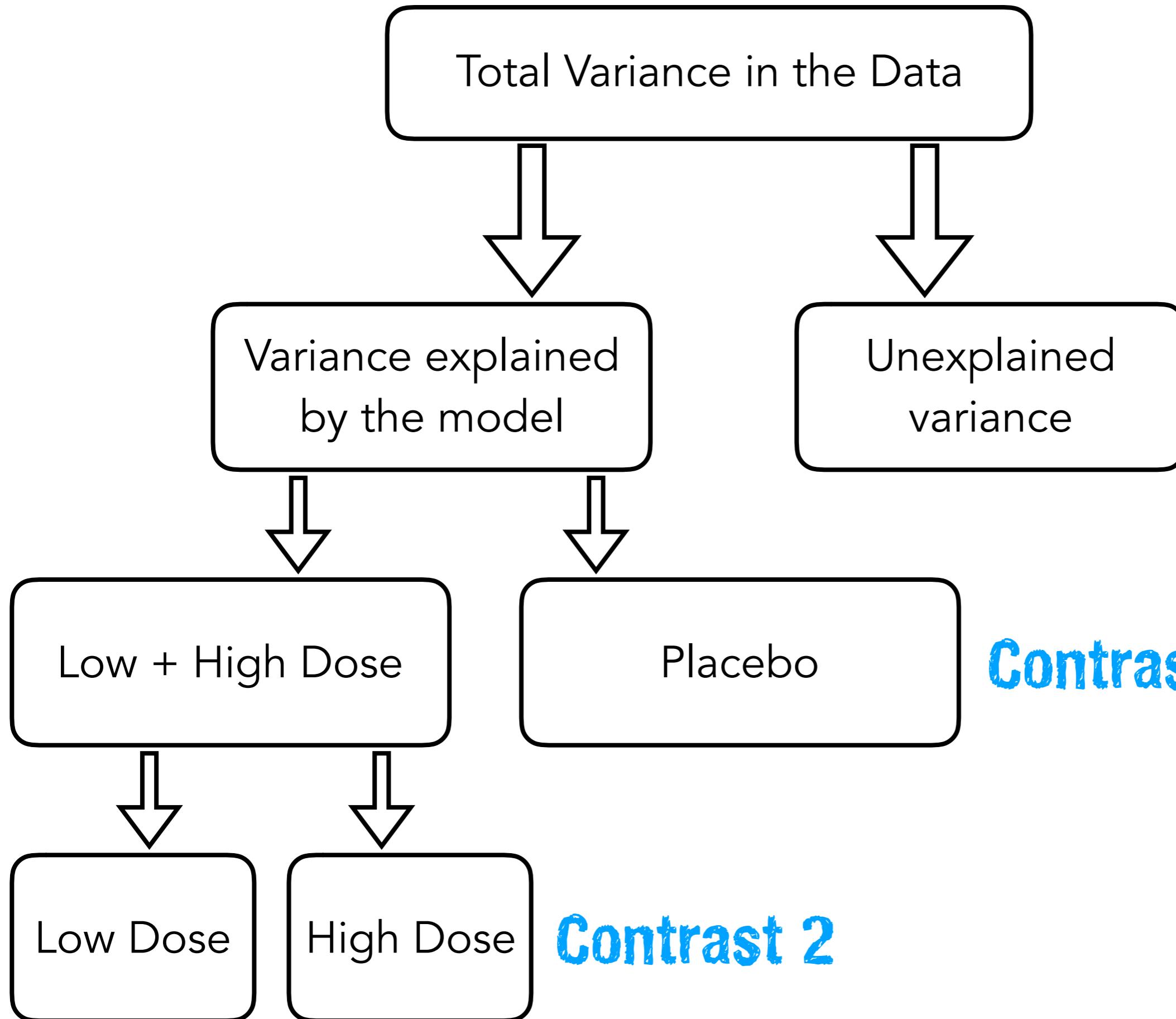


six possible comparisons
if we test all hypotheses,
we increase the chance
of making a type I error

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.

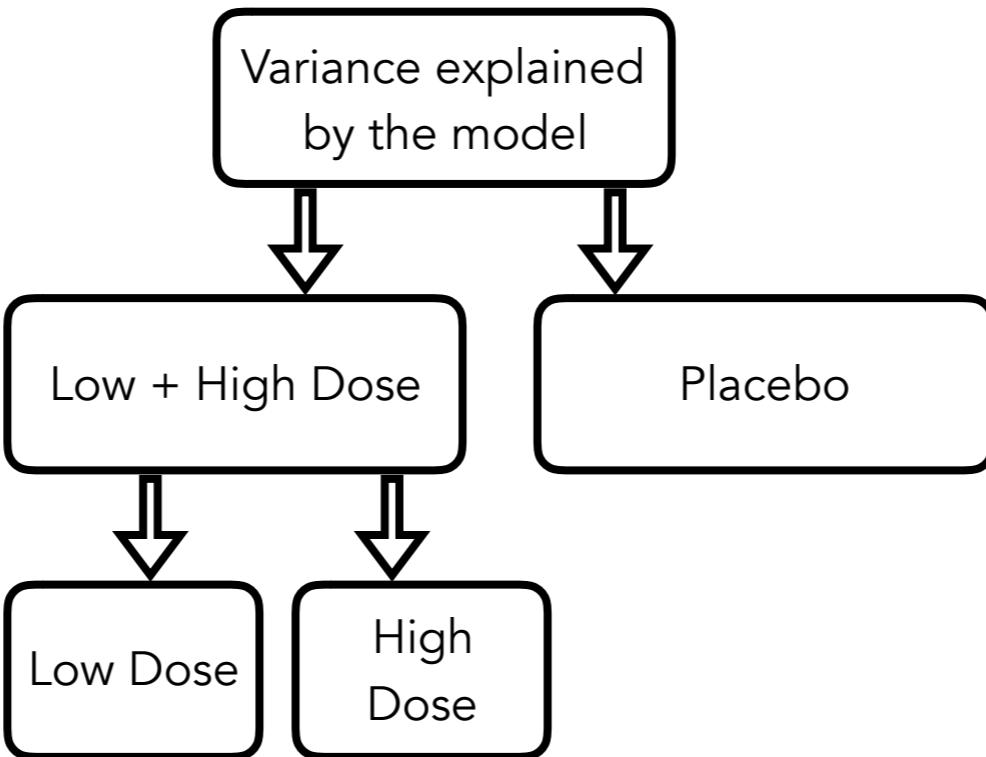


ANOVA

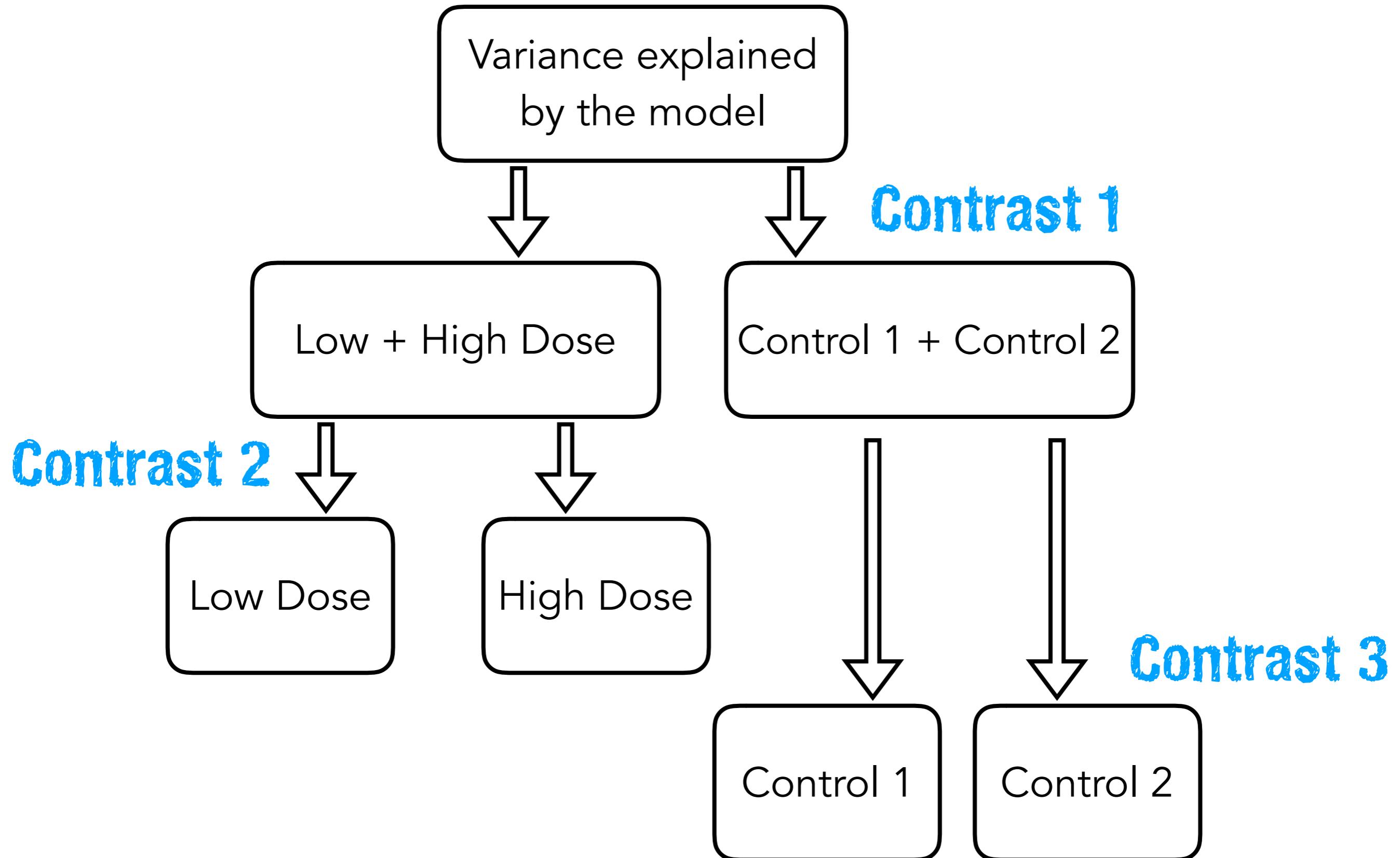
Contrast 1

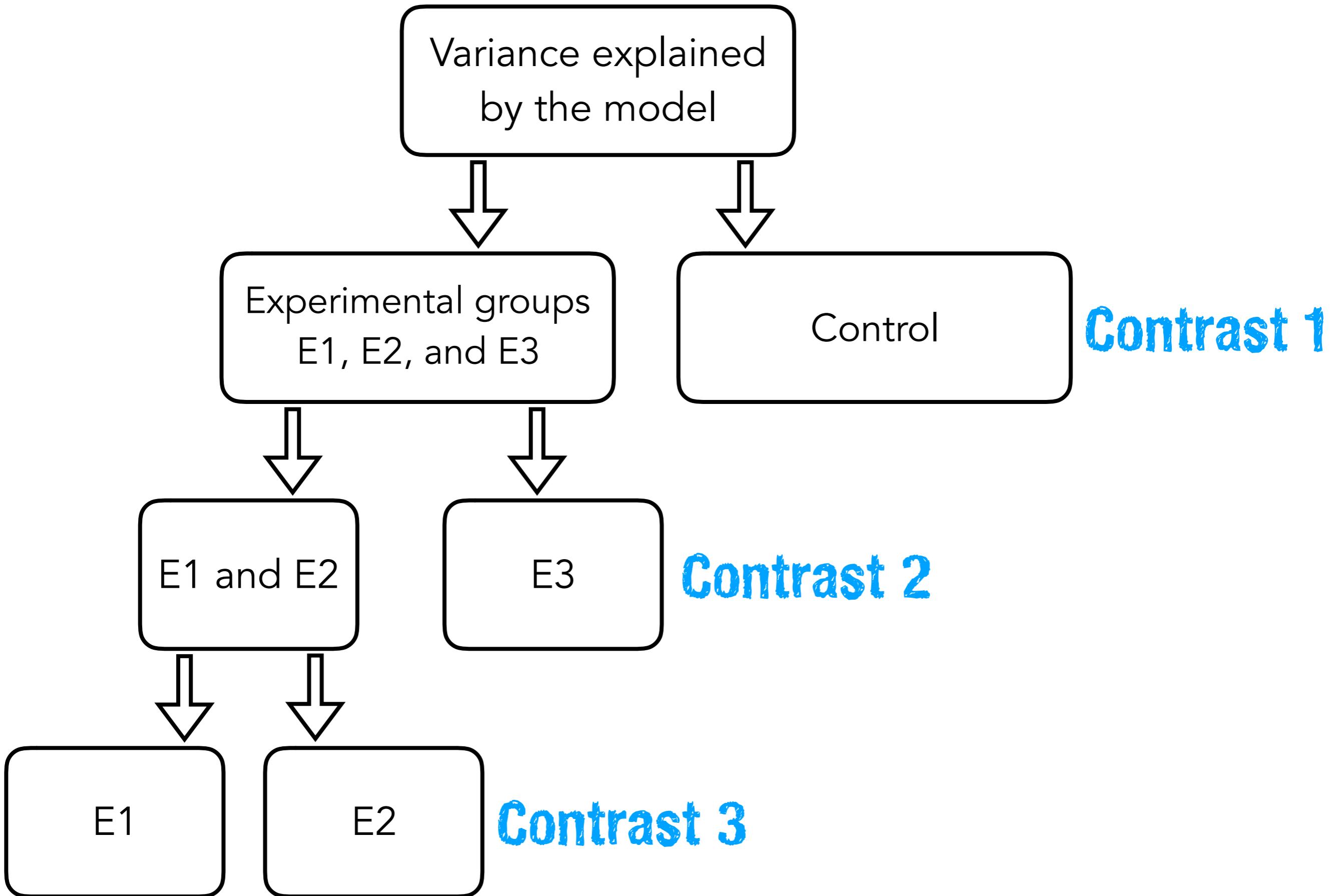
Contrast 2

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



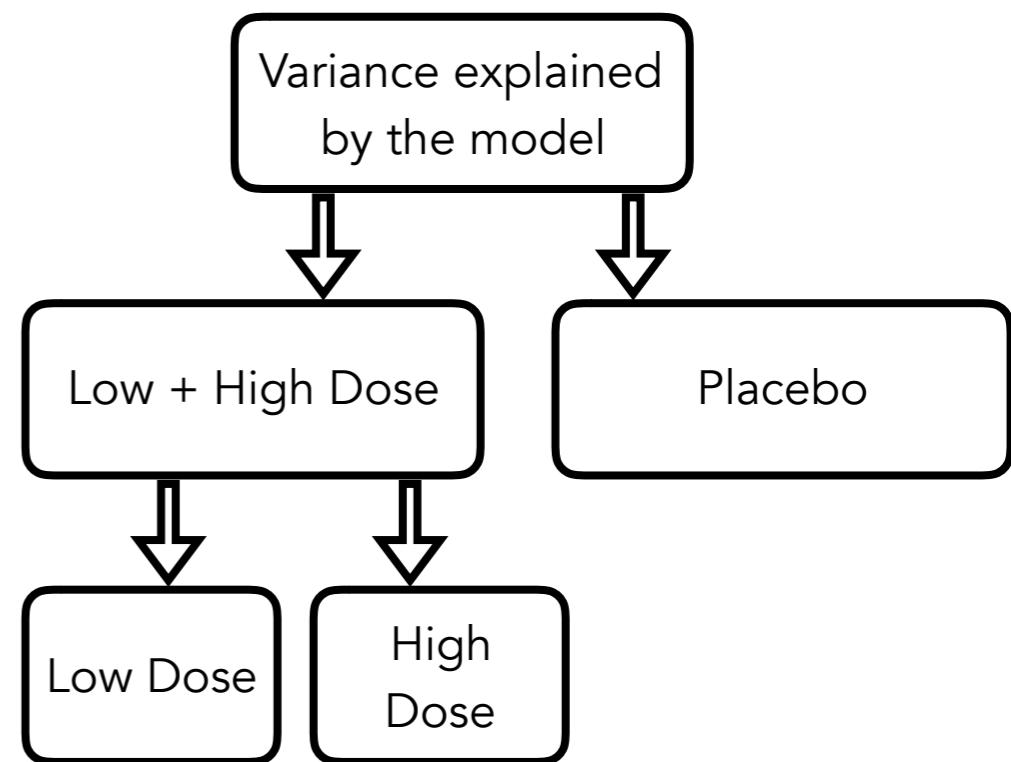


Defining orthogonal contrast codes

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

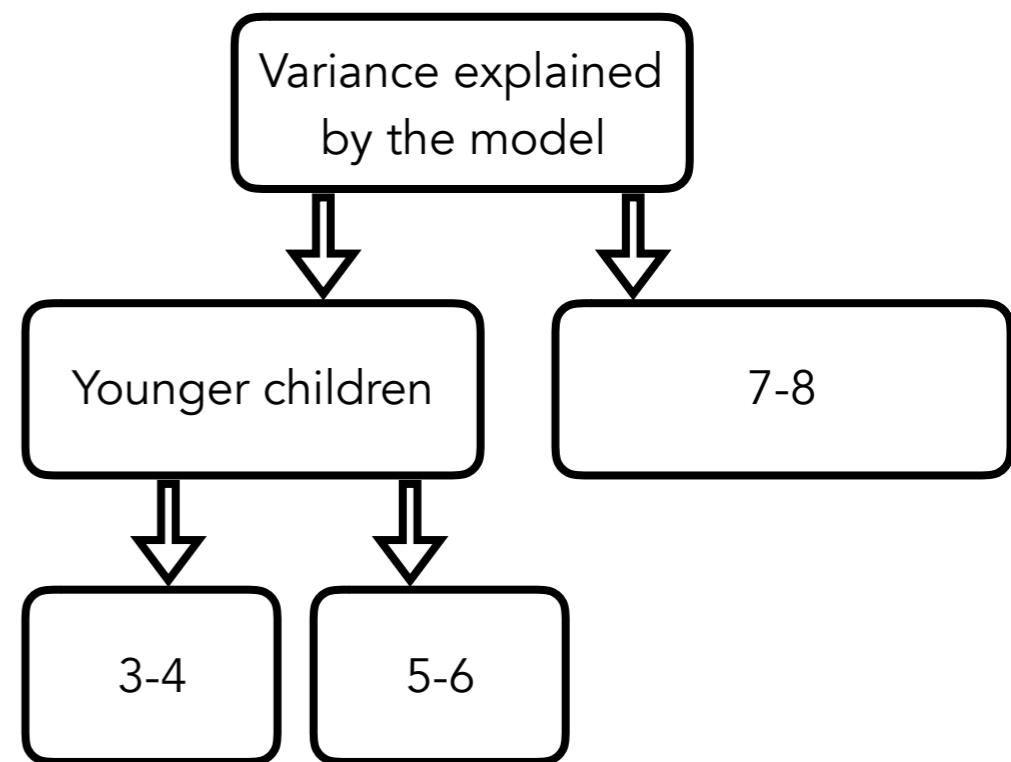


contrast	Low	High	Placebo	SUM
Dose vs Placebo	0.5	0.5	-1	0
Low vs. high	-1	1	0	0

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

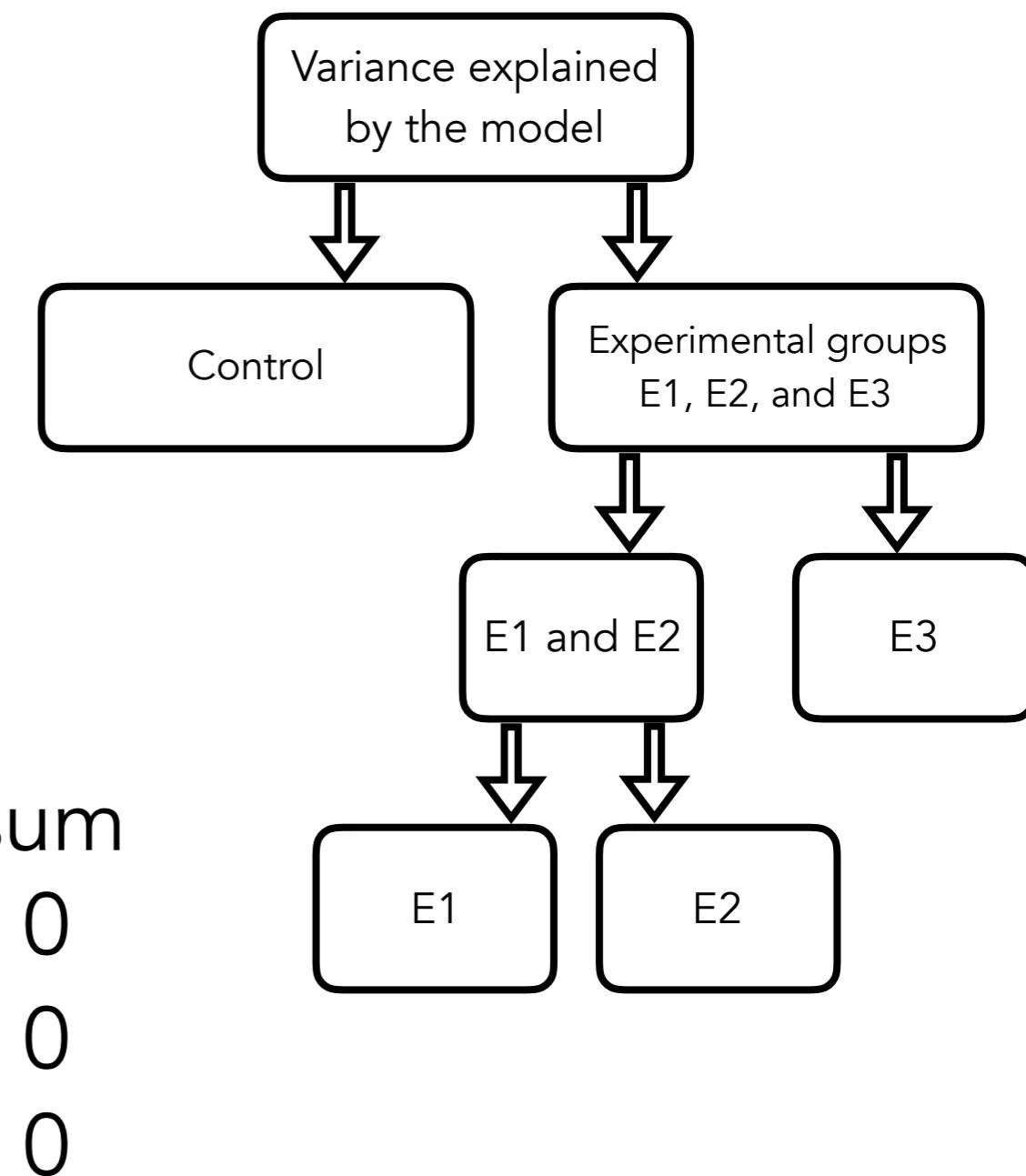


contrast	3-4	5-6	7-8	sum
Young vs. old	-0.5	-0.5	1	0
3-4 vs. 5-6	-1	1	0	0
Product	0.5	-0.5	0	0

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



Check for products

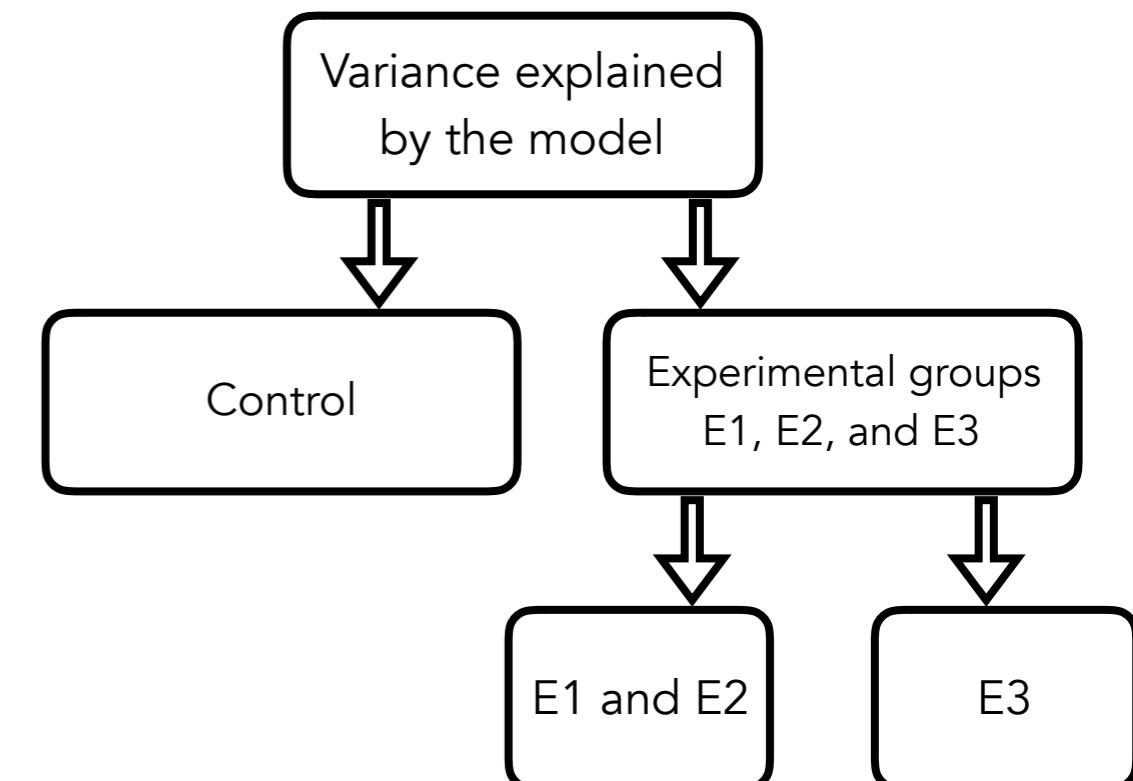
contrast	Control	E1	E2	E3
Control vs. E	-3	1	1	1
E1&2 vs. E3	0	-1	-1	2

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



sum
0
0
0

Check for products

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

sum
0

product 0 -1 1 0

Defining contrast codes

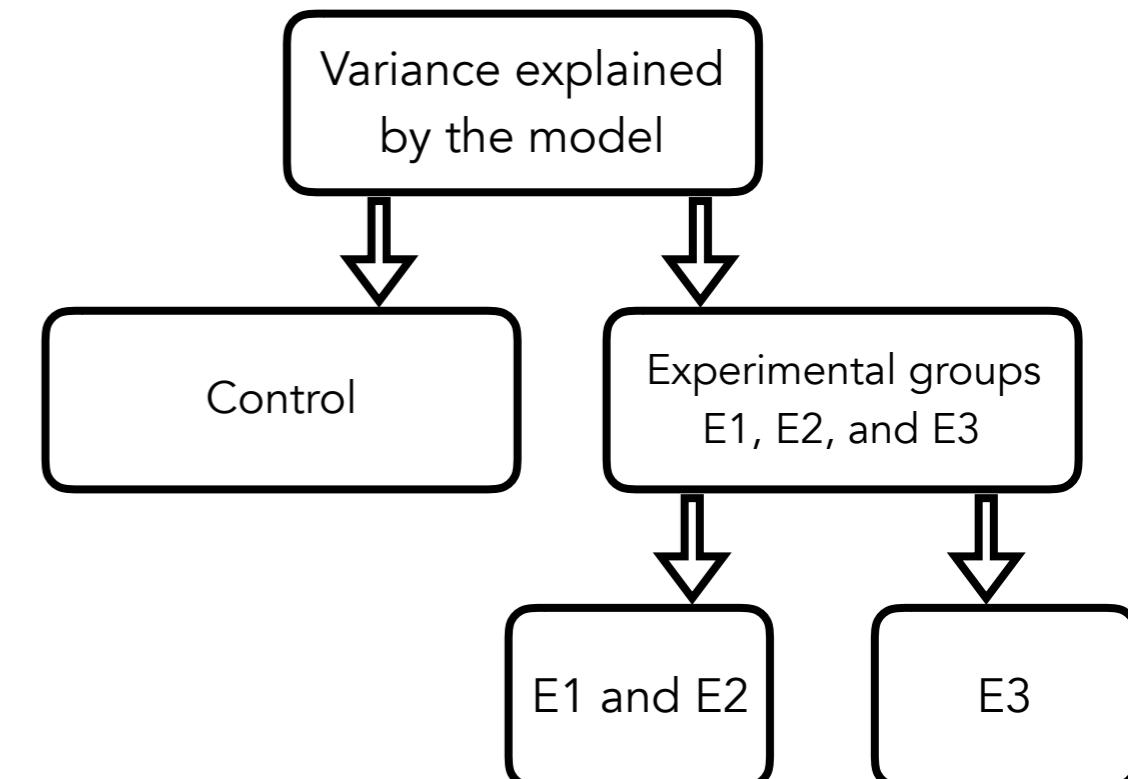
- The sum of weights for each comparison should be 0
- The product of the weights for any pair comparisons should sum to 0

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

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 -1 0



sum
0
0
0



sum
0

contrasts are
orthogonal

Planned contrasts in R

```
1 library("emmeans")
2
3 fit = lm(formula = performance ~ group,
4           data = df.development)
5
6 # check factor levels
7 levels(df.development$group) [1] "3-4" "5-6" "7-8"
8
9 # define the contrasts of interest
10 contrasts = list(young_vs_old = c(-0.5, -0.5, 1),
11                   three_vs_five = c(-1, 1, 0))
12
13 # compute estimated marginal means
14 leastsquare = emmeans(fit, "group")
15
16 # run analyses
17 contrast(leastsquare,
18            contrasts,
19            adjust = "bonferroni")
```

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

```
1 library("emmeans")
2
3 fit = lm(formula = performance ~ group,
4           data = df.development)
5
6 # post hoc tests
7 leastsquare = emmeans(fit, "group")
8 pairs(leastsquare,
9        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

```
1 # fit the model  
2 fit = lm(formula = balance ~ hand + skill,  
3           data = df.poker)  
4  
5 # post hoc tests  
6 leastsquare = emmeans(fit, c("hand", "skill"))  
7 pairs(leastsquare,  
8        adjust = "bonferroni")
```

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

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 ' '

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 ' '

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.

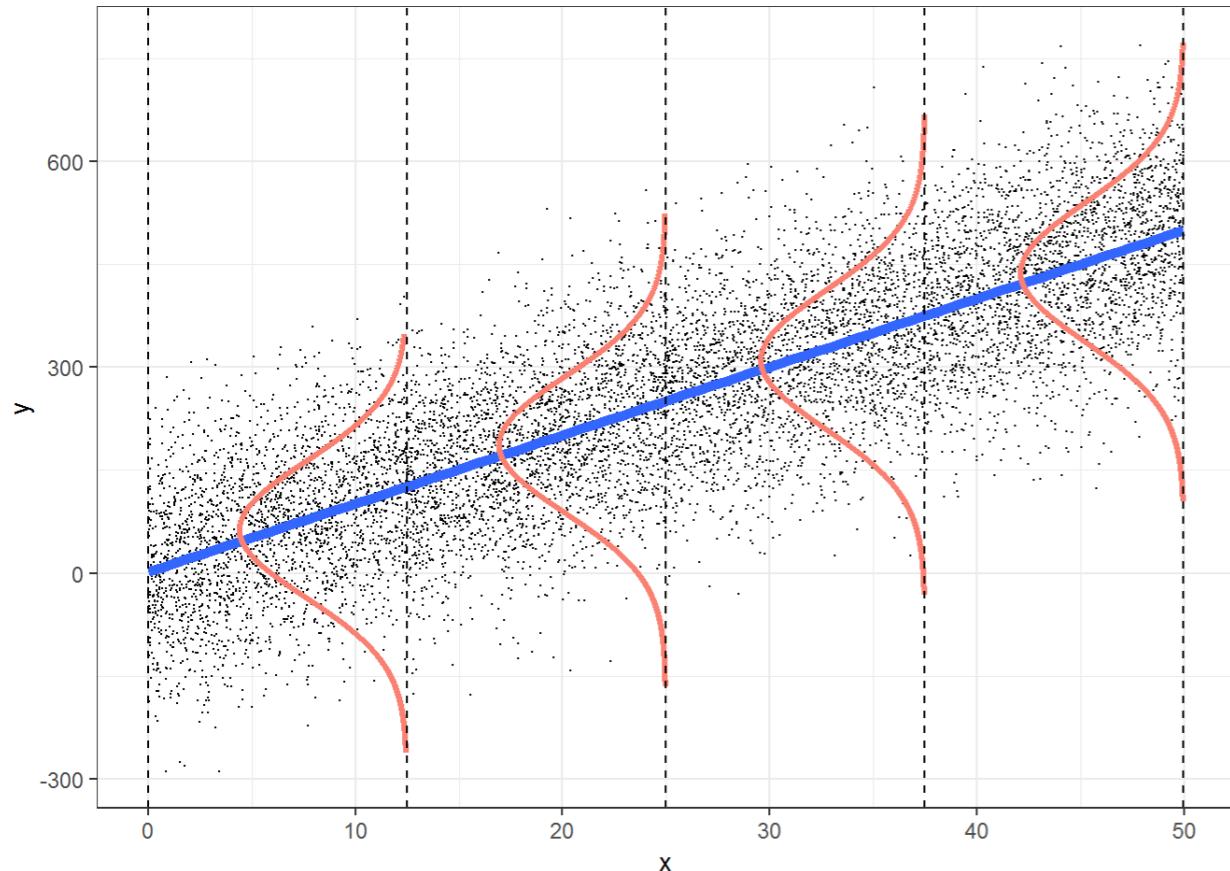
Coding schemes

- **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

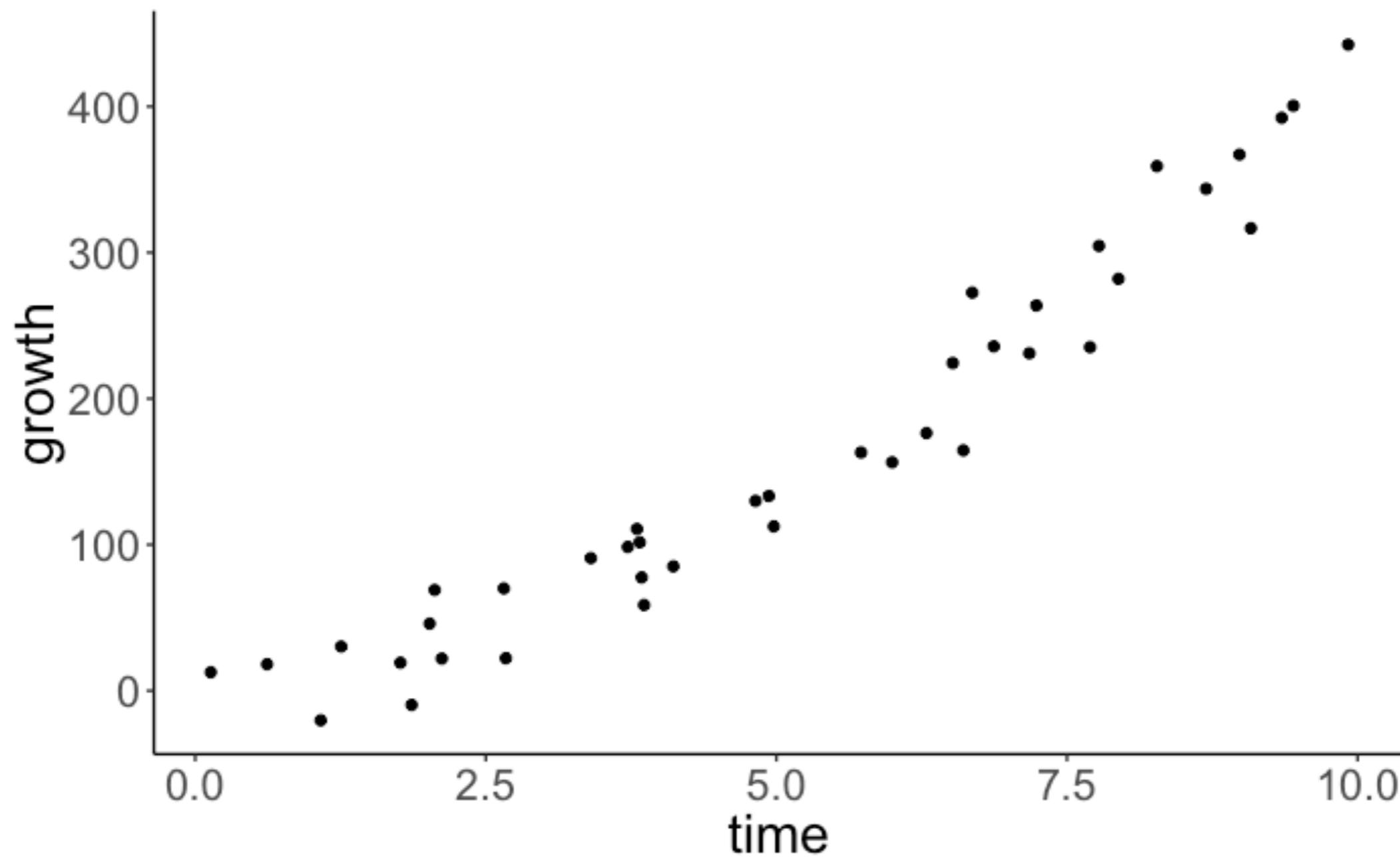
Assumptions of linear model

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



Distribution of errors

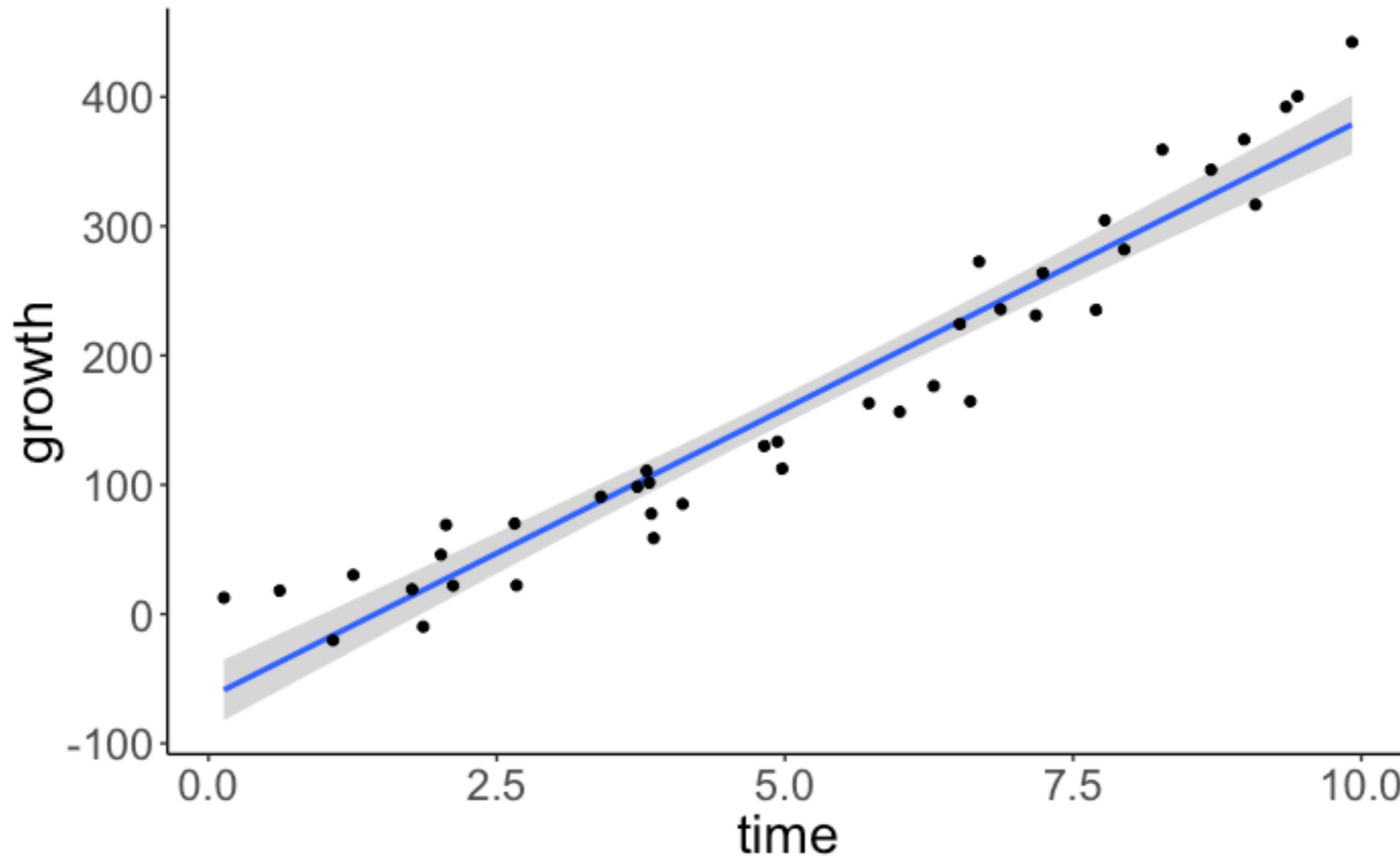
Distribution of errors



simulated data set

Distribution of errors

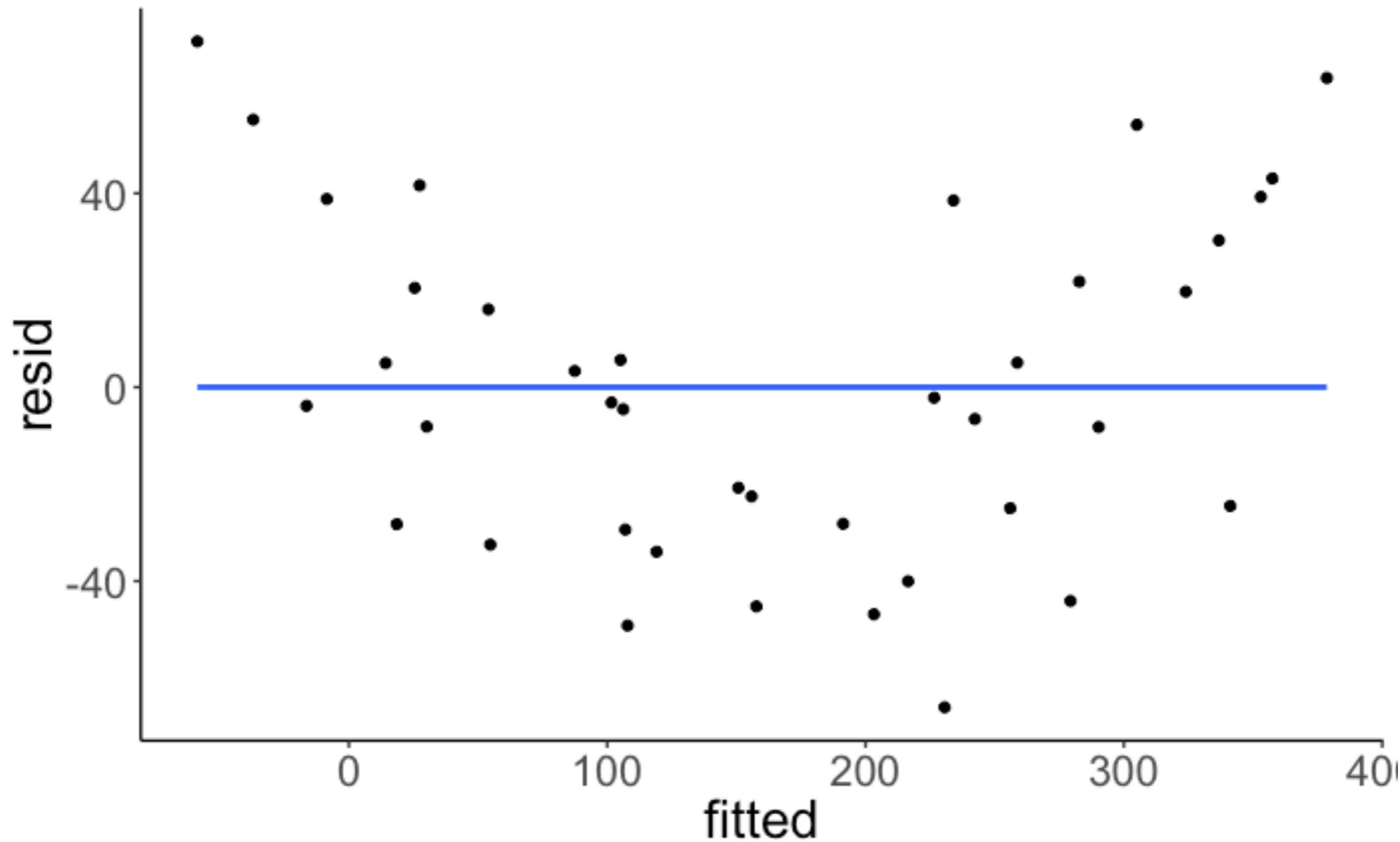
`lm(formula = growth ~ 1 + time, data = df.growth)`



simulated data set

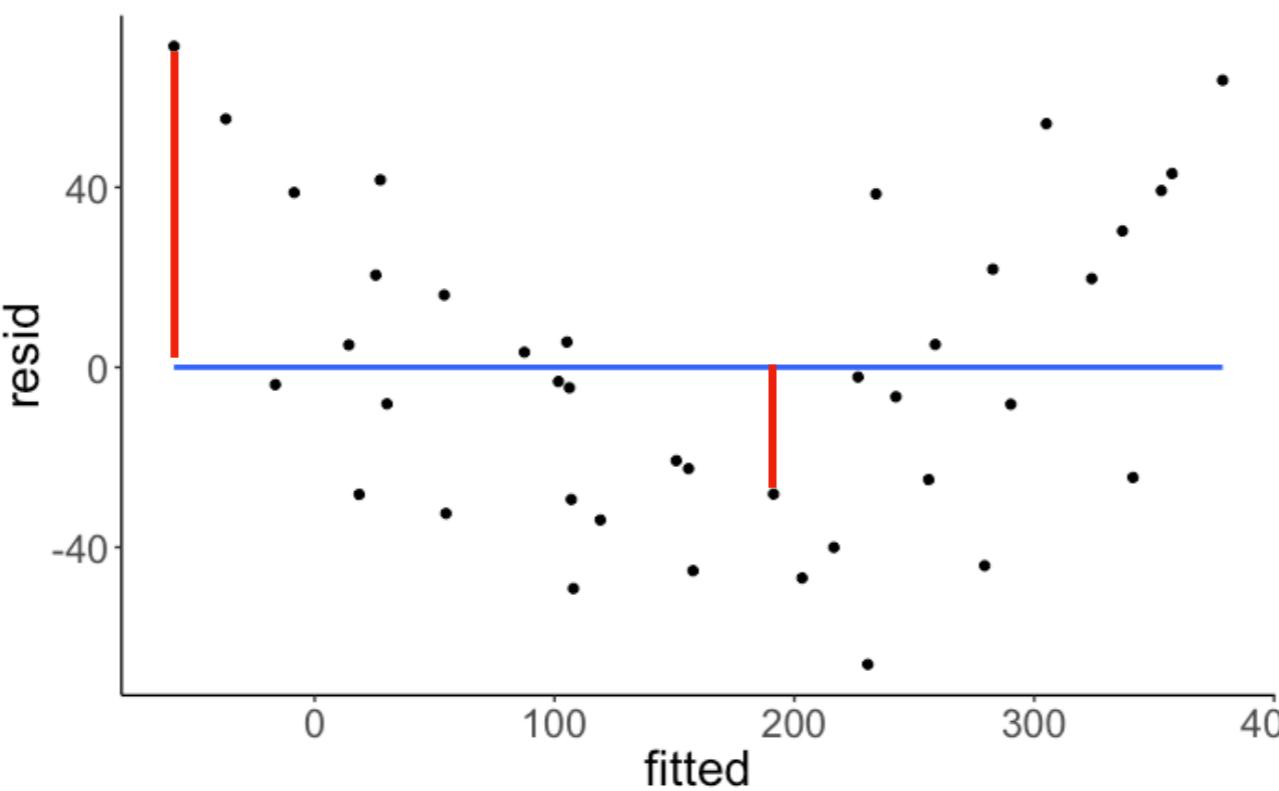
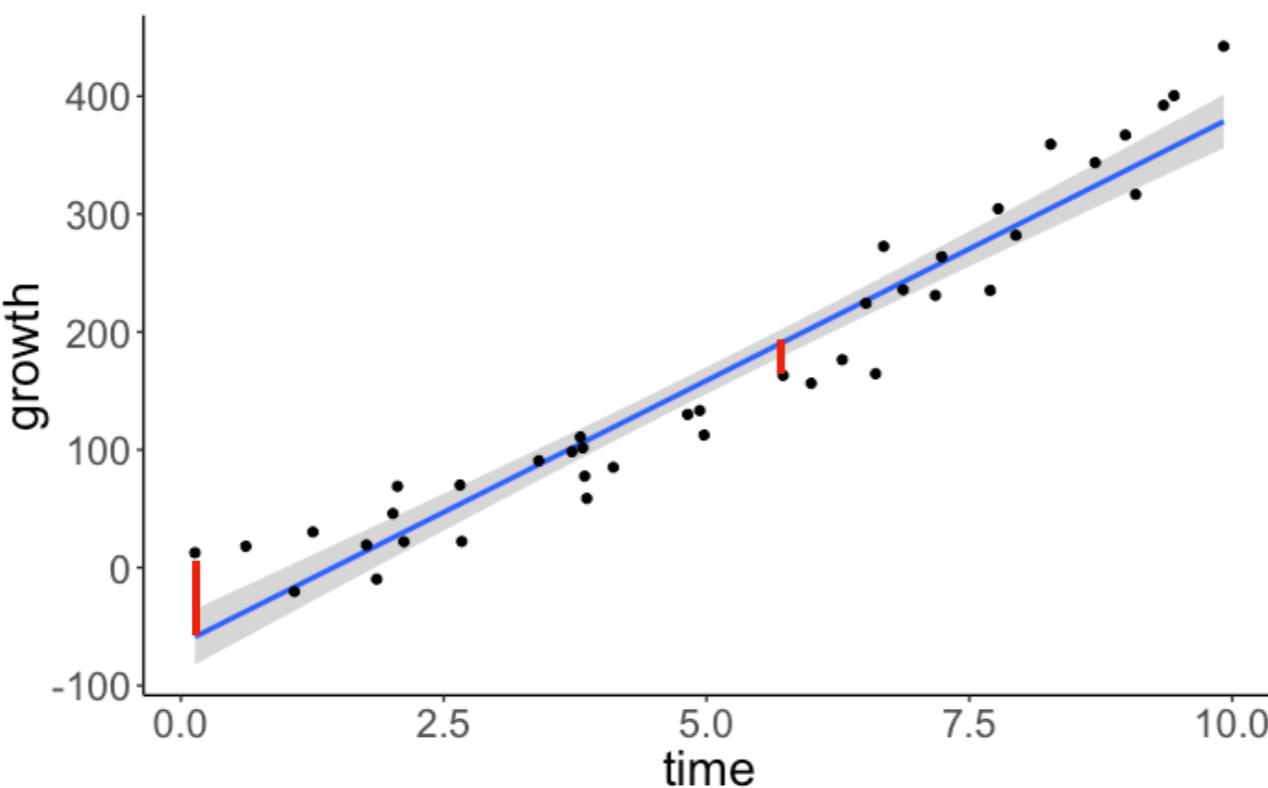
Distribution of errors

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



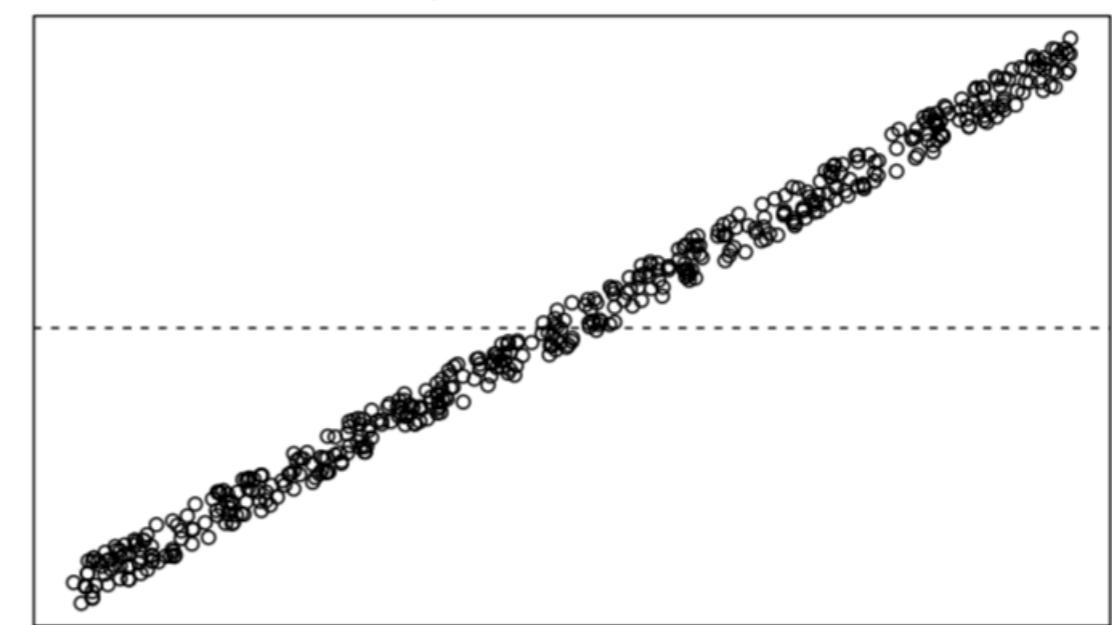
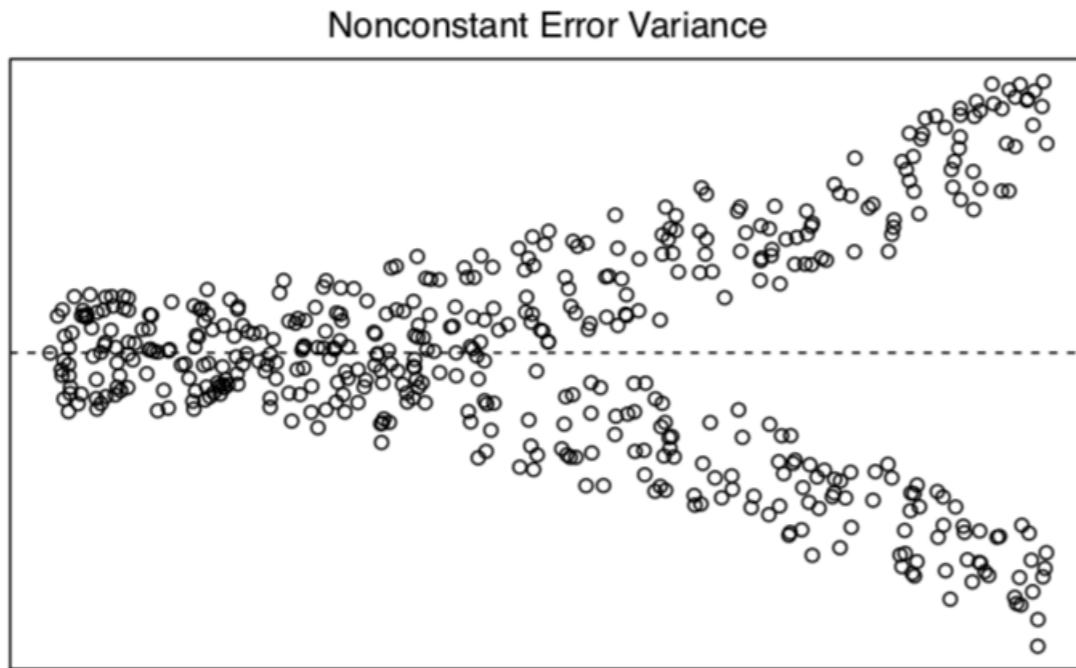
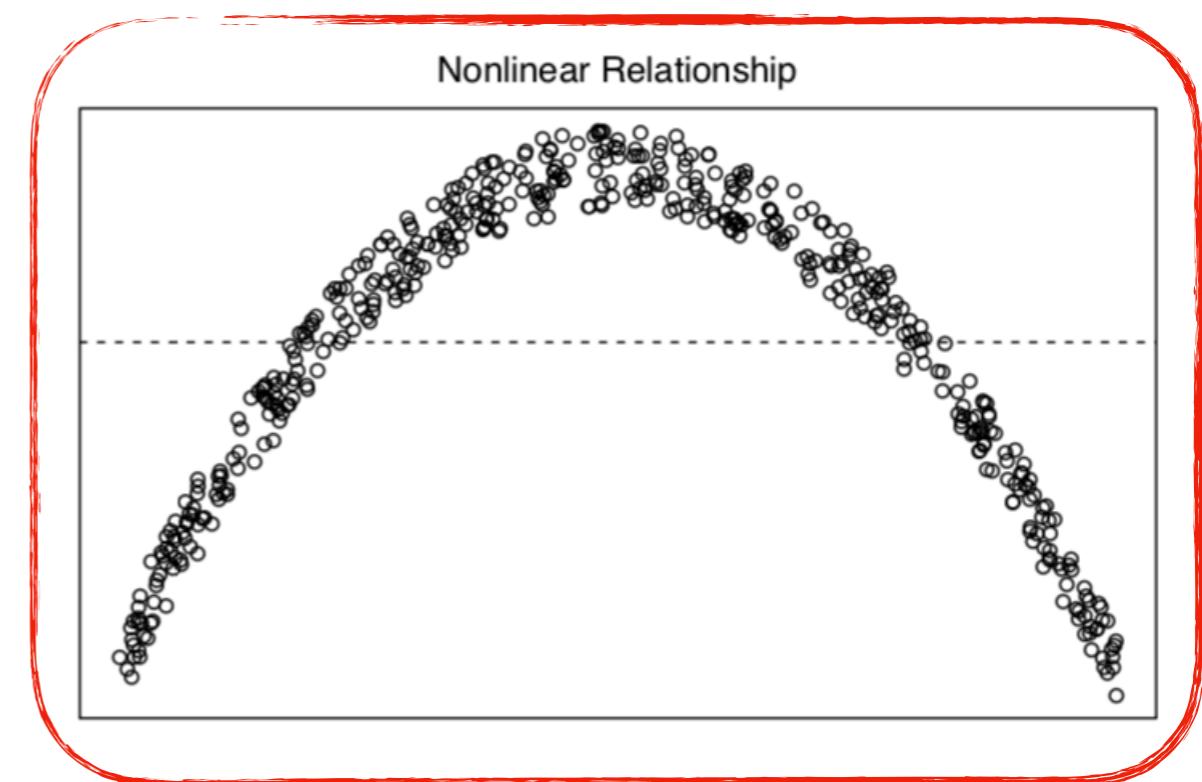
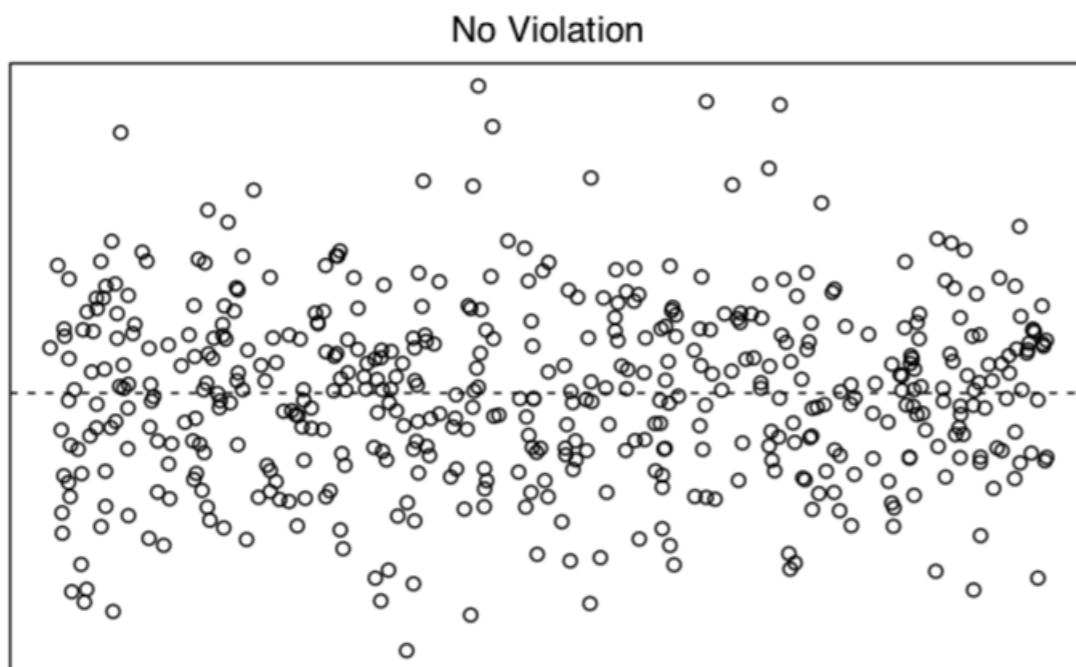
Residual plot

Distribution of errors



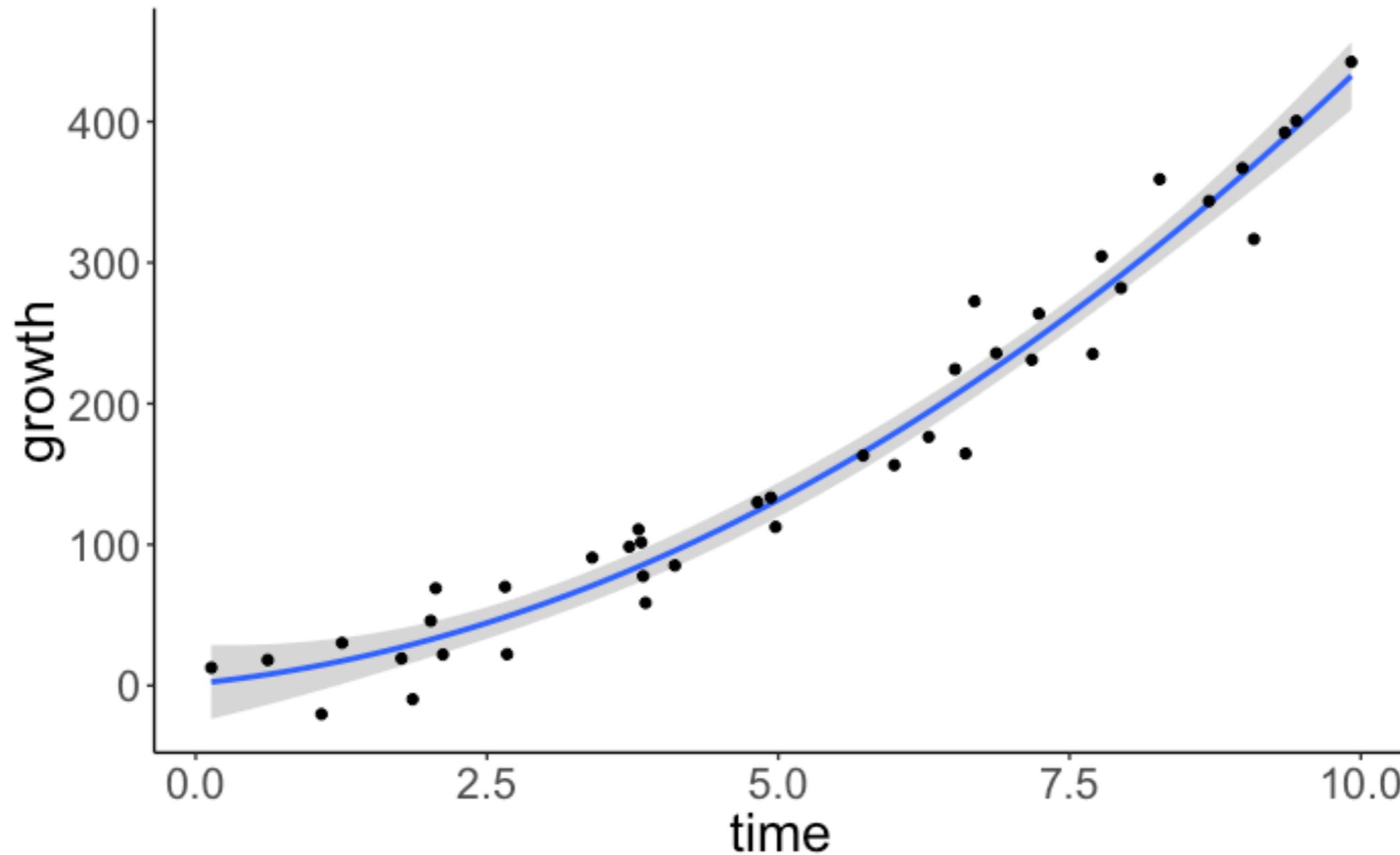
Residual plot

Distribution of errors



Distribution of errors

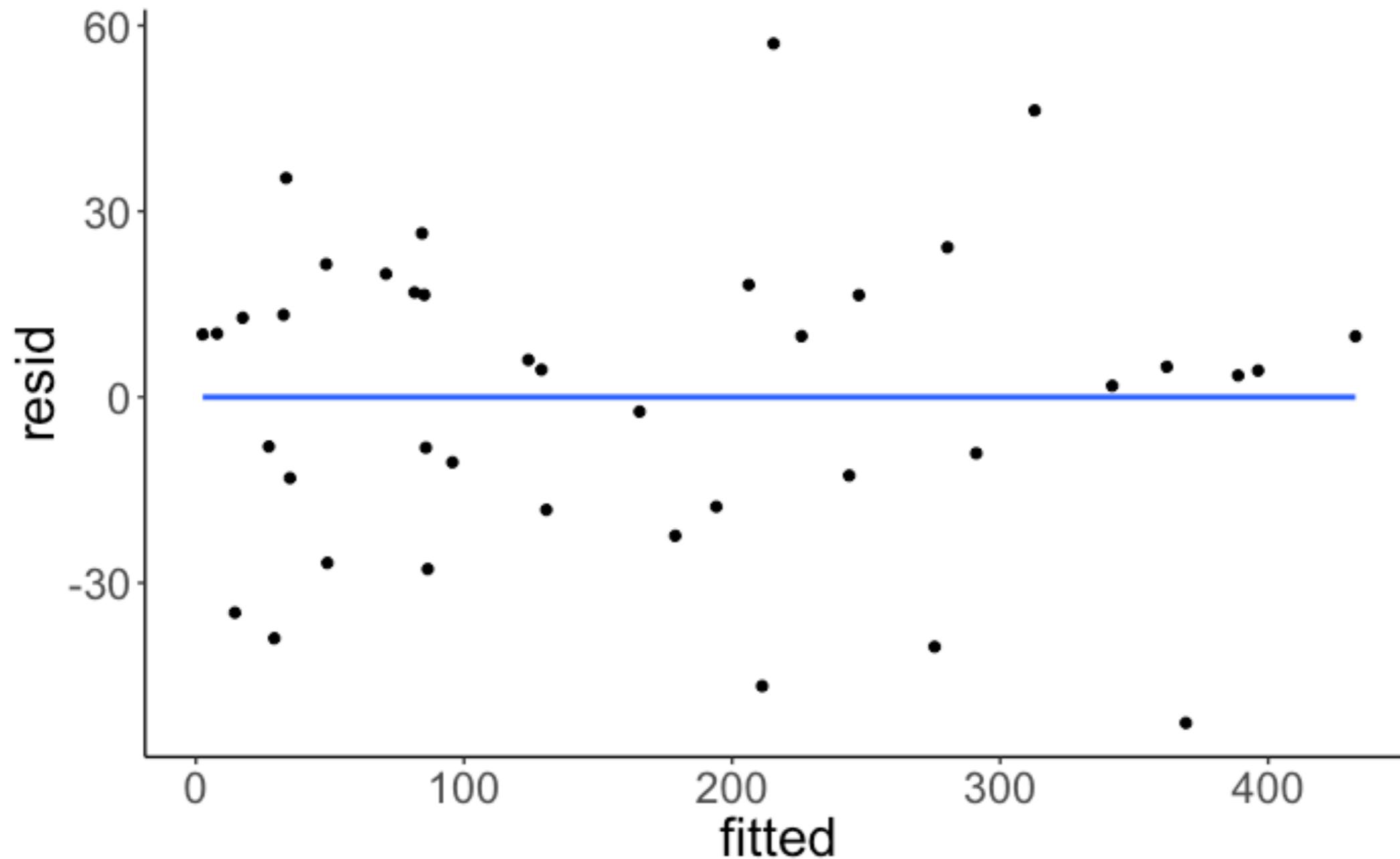
`lm(formula = growth ~ 1 + time + I(time^2), data = df.growth)`



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:

$$\begin{aligned}\widehat{\text{growth}}_i &= b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2 \\ &= b_0 + b_1 \cdot 1 + b_2 \cdot 1 \\ &= 1.36 + 8.39 \cdot 1 + 3.54 \cdot 1 \\ &= 13.29\end{aligned}$$

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:

$$\begin{aligned}\widehat{\text{growth}}_i &= b_0 + b_1 \cdot \text{time}_i + b_2 \cdot \text{time}_i^2 \\ &= b_0 + b_1 \cdot 7 + b_2 \cdot 7 \\ &= 1.36 + 8.39 \cdot 7 + 3.54 \cdot 7 \\ &= 84.87\end{aligned}$$

Simpsons paradox

Simpson's paradox

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

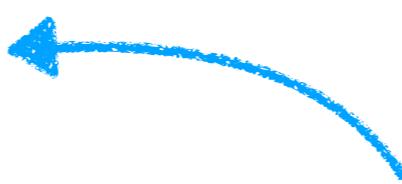
```
Call:
lm(formula = y ~ x, data = df.simpson)

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 

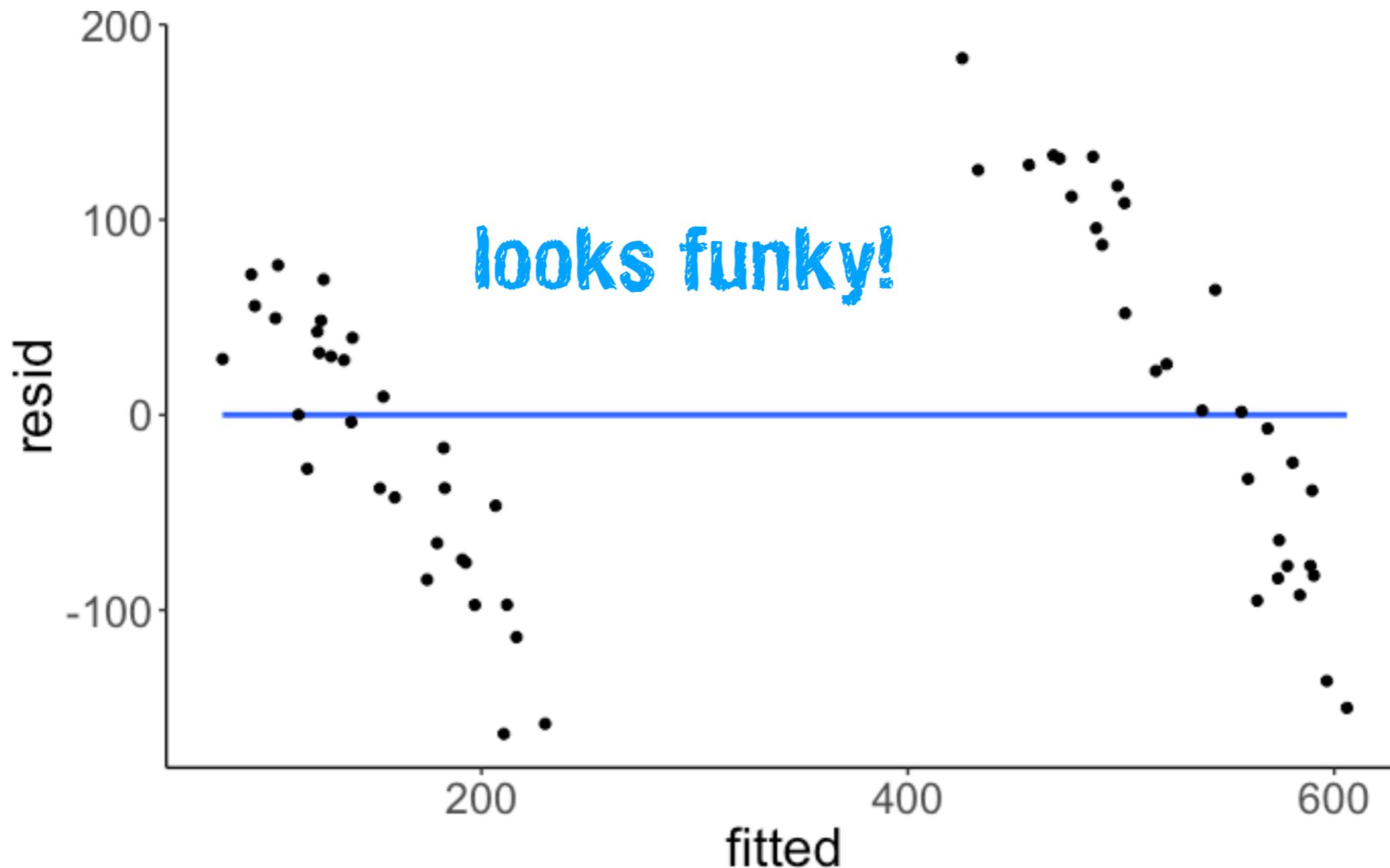
Residual standard error: 99.23 on 58 degrees of freedom
Multiple R-squared:  0.7925,    Adjusted R-squared:  0.7889 
F-statistic: 221.5 on 1 and 58 DF,  p-value: < 2.2e-16
```

There is a significant positive relationship
between x and y.

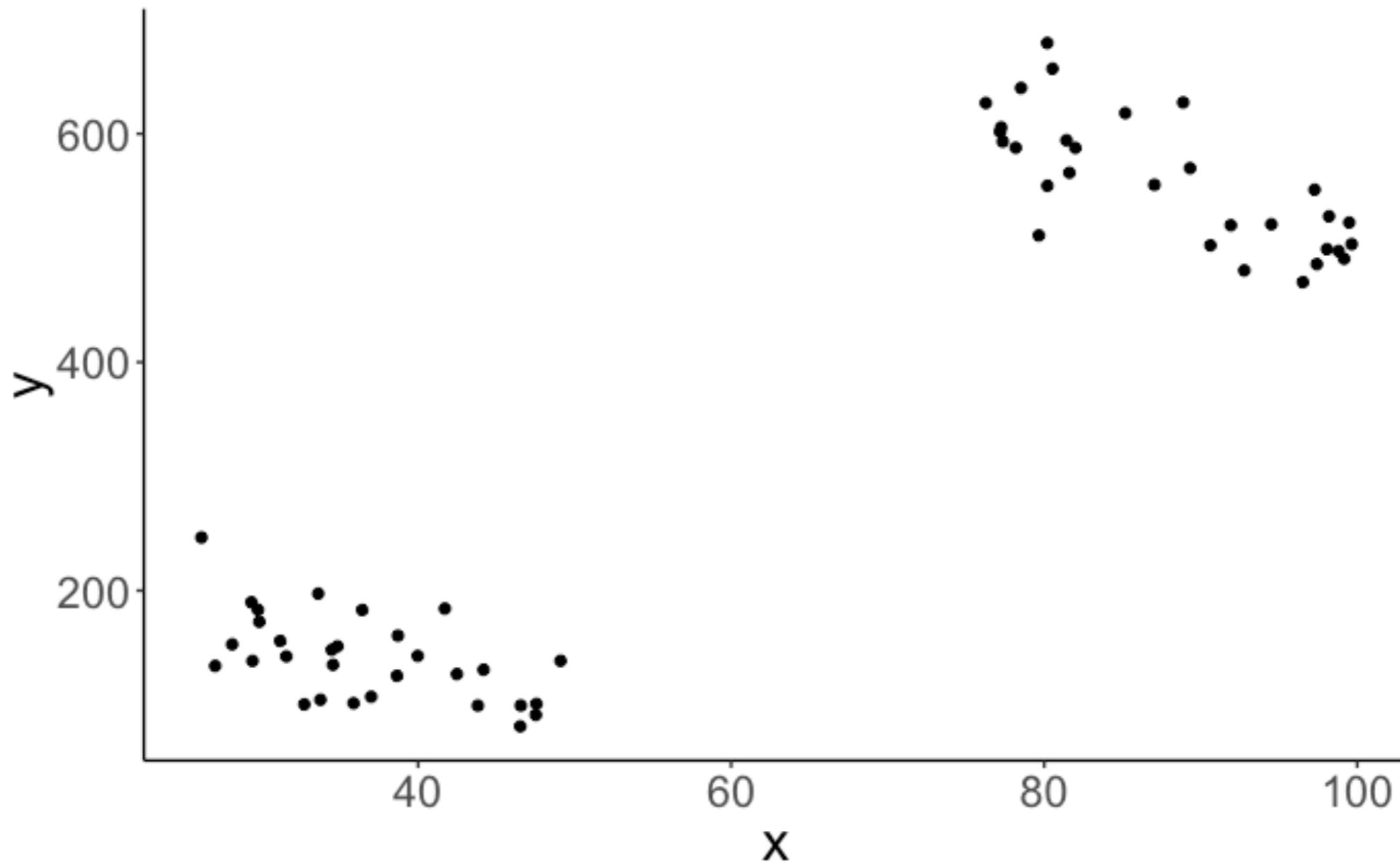


**you should take a look
at the data first!**

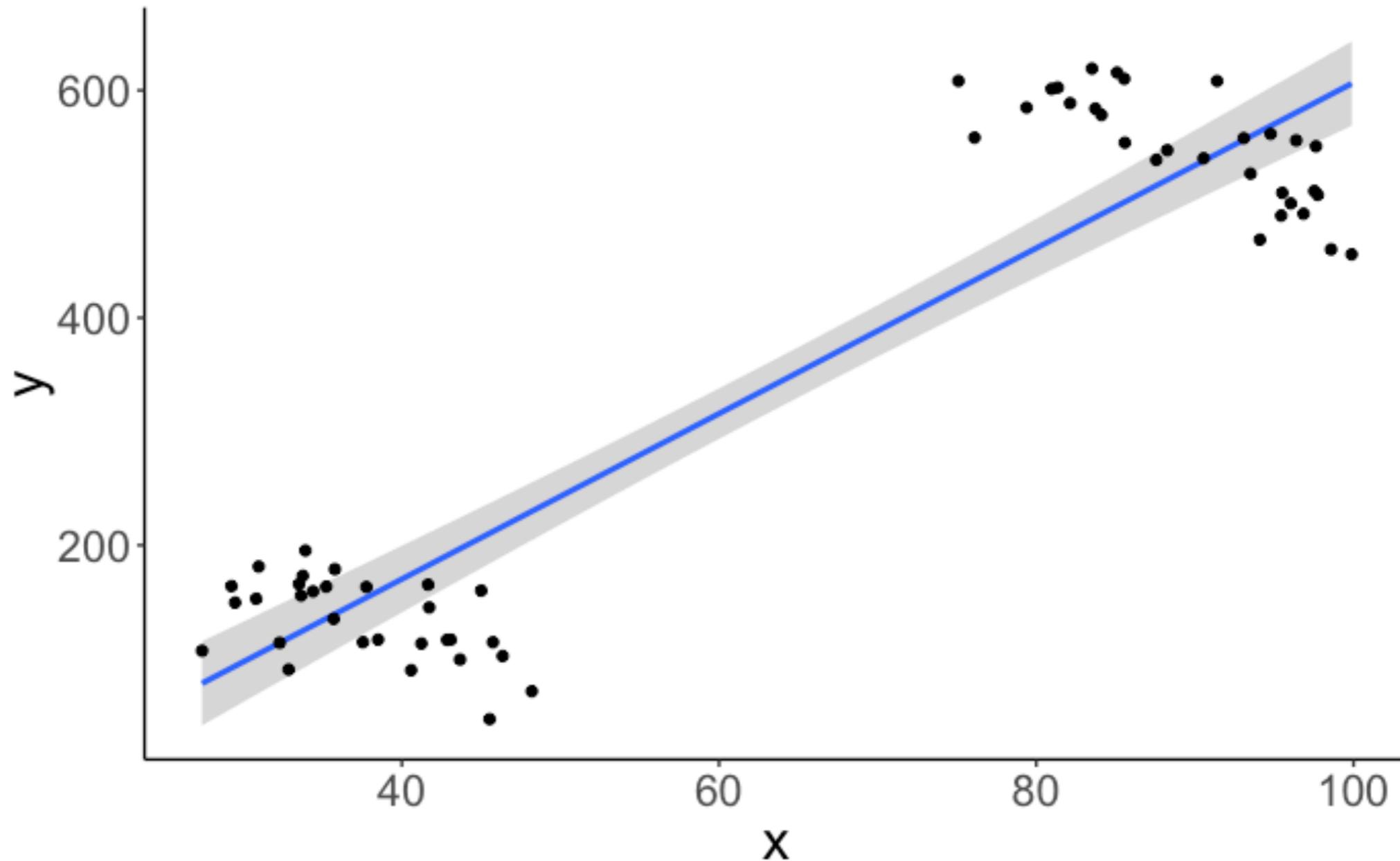
Simpson's paradox



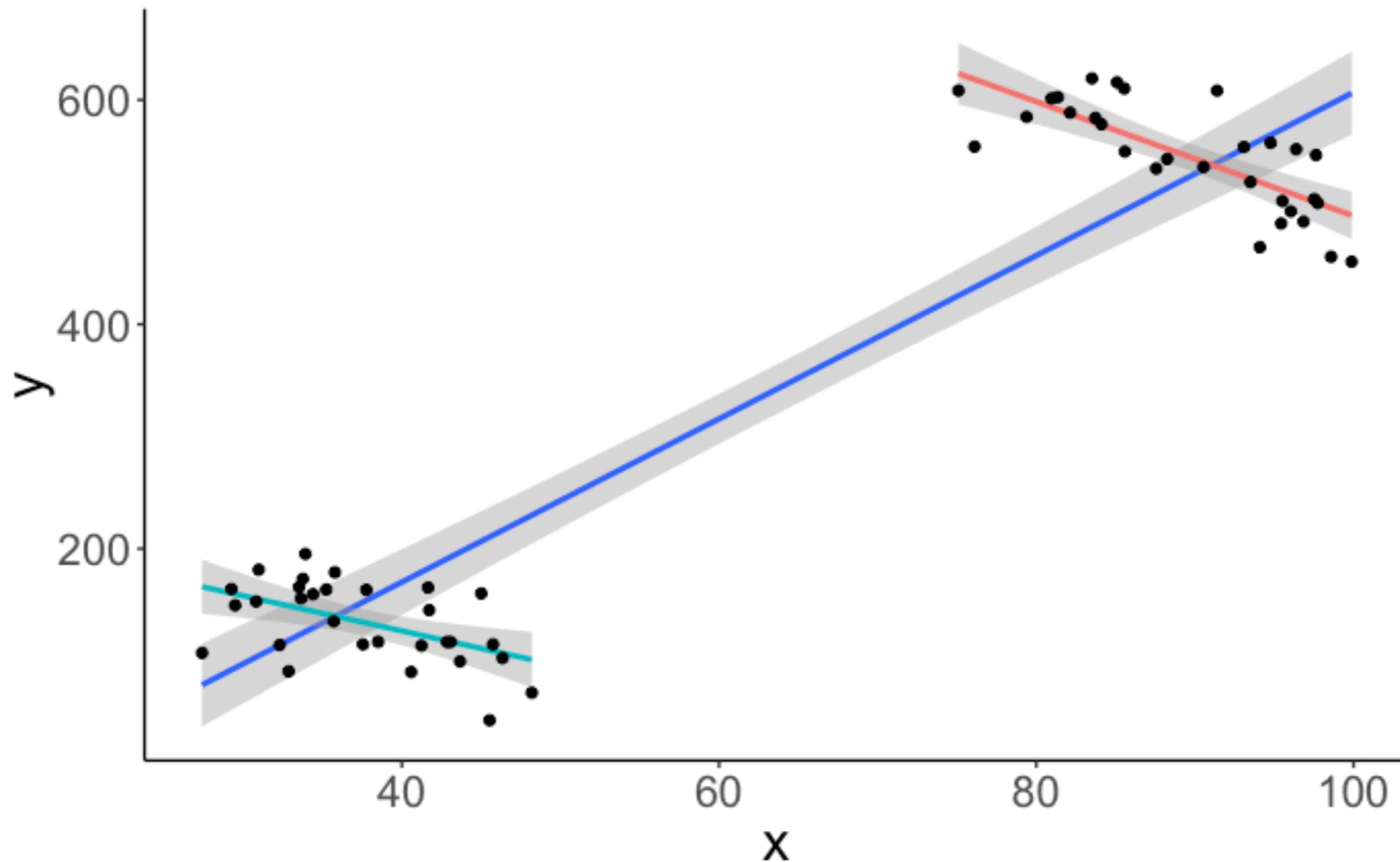
Simpson's paradox



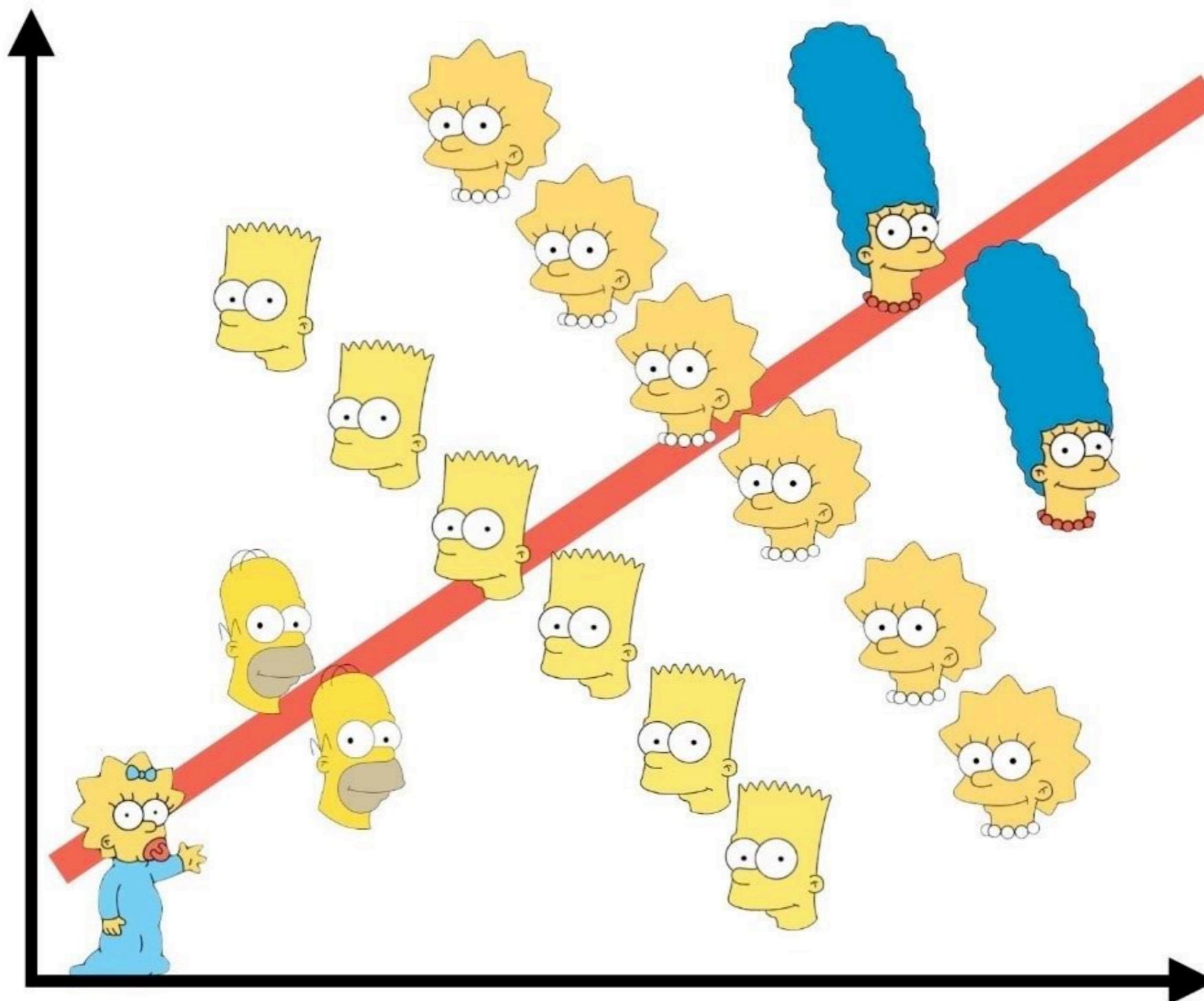
Simpson's paradox



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?

much a little just a little much
too too right too too
slow slow

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!