

Session 07: Multiple Lineare Regression

Dominic Schmitz & Janina Esser

Verein für Diversität in der Linguistik

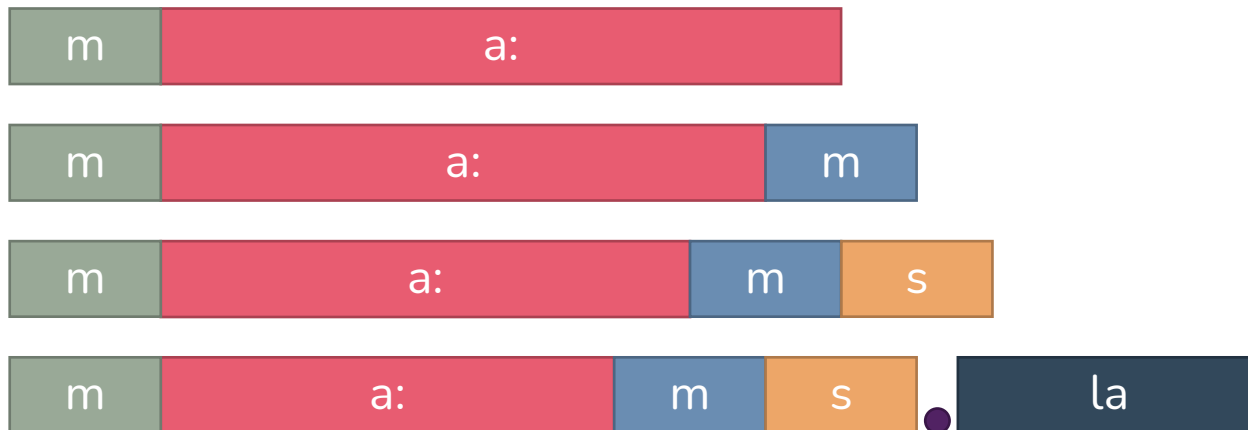
Beispieldaten



- Für die folgenden Beispiele werden wir Daten folgender Studie nutzen:

Compensatory Vowel Shortening in German¹

- Stressed Vowels sind kürzer je nachdem wie viele Konsonanten ihnen folgen:



¹ Schmitz, D., Cho, H.-E., & Niemann, H. (2018). Vowel shortening in German as a function of syllable structure. Proceedings 13. Phonetik Und Phonologie Tagung (P&P13), 181–184.

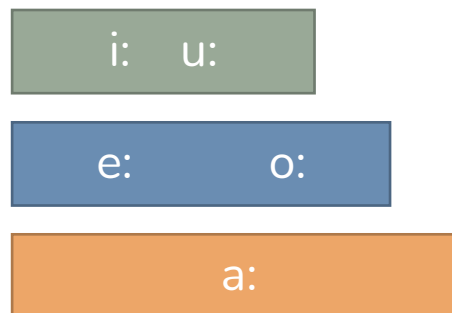
Beispieldaten



- Für die folgenden Beispiele werden wir Daten folgender Studie nutzen:

Compensatory Vowel Shortening in German¹

- Unabhängig von diesem Vowel Shortening gilt, dass offene Vokale länger sind als halb-offene Vokale, und halb-offene Vokale sind länger als geschlossene Vokale:



¹ Schmitz, D., Cho, H.-E., & Niemann, H. (2018). Vowel shortening in German as a function of syllable structure. Proceedings 13. Phonetik Und Phonologie Tagung (P&P13), 181–184.

Simple Lineare Regression



kontinuierliche
abhängige Variable

unabhängige
Predictor Variable

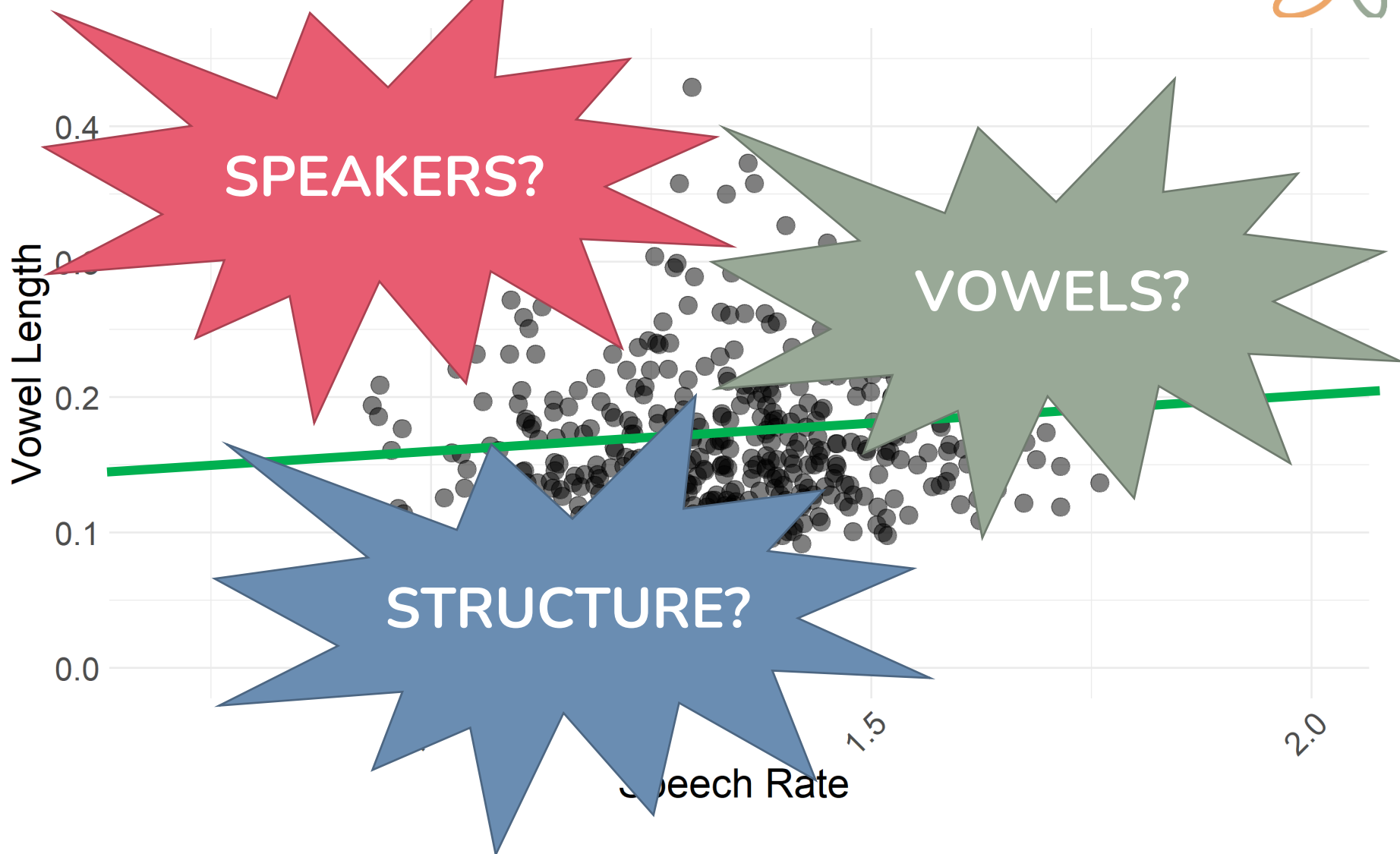
$$Y = \beta_1 + \beta_2 X + \epsilon$$

Intercept

Slope

Error Term /
Residuen

(Zu) Simple Lineare Regression



Simple Lineare Regression: Formel



kontinuierliche
abhängige Variable

unabhängige
Predictor Variable

$$Y = \beta_1 + \beta_2 X + \epsilon$$

Intercept

Slope

Error Term /
Residuen

Multiple Lineare Regression: Formel



kontinuierliche abhängige Variable

unabhängige Predictor Variable 1

unabhängige Predictor Variable 2

unabhängige Predictor Variable i

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i + \epsilon$$

Intercept

Slope von Variable 1

Slope von Variable 2

Slope von Variable i

Error Term / Residuen

Multiple Lineare Regression in R



- Mehr Variablen = mehr Zeitaufwand
- Typische Schritte bei Multipler Linearer Regression sind
 1. Distribution der abhängigen Variable überprüfen
 2. „volles“ Modell erstellen
 3. „bestes“ Modell finden
 4. Assumptions überprüfen
 5. Modell interpretieren

1: Distribution der abhängigen Variable



- Wie wir bereits wissen, nutzen wir hierzu den Shapiro-Wilk Test
- Die abhängige Variable in unserem Beispiel, duration, ist nicht normalverteilt
- Daher nutzen wir wieder eine log-transformierte Version der Variable, durationLog

2: „Volles“ Modell



- Unsere abhängige Variable ist `durationLog`
- Als nächstes müssen wir die unabhängigen Variablen identifizieren, die wir nutzen möchten
- In diesem Beispiel sind es die folgenden Variablen:
 - `structure` i.e. coda structure
 - `vowel` i.e. vowel quality
 - `rate` i.e. speech rate
 - `number` i.e. slide number during experiment

2: „Volles“ Modell



- Erstellen des „vollen“ Modells:

```
model = lm(durationLog ~ structure +  
            vowel +  
            rate +  
            number,  
            data)
```

3: „Bestes“ Modell



- Theoretisch müssten wir nun alle möglichen Variabel-Kombinationen testen um das “beste” Modell zu finden
- Allerdings ist dieser Vorgang manuell durchgeführt fehleranfällig und zeitaufwendig (und macht wirklich keinen Spaß)
- Zum Glück gibt es eine Funktion, die diesen Schritt übernimmt:

`step(model)`

3: „Bestes“ Modell



```
> step(model)
```

3: „Bestes“ Modell



```
> step(model)
```

```
Start: AIC=-1167.31
```

Akaike Information Criterion

The lower, the better the model fit

```
durationLog ~ structure + vowel + rate + number
```

3: „Bestes“ Modell



```
> step(model)
```

Start: **AIC=-1167.31**

Akaike Information Criterion

The lower, the better the model fit

```
durationLog ~ structure + vowel + rate + number
```

	Df	Sum of Sq	RSS	AIC	
- number	1	0.0536	31.839	-1168.55	a model without number
<none>			31.786	-1167.31	
- rate	1	0.8500	32.636	-1157.48	
- vowel	4	3.4109	35.197	-1129.64	a model without vowel
- structure	2	14.9708	46.756	-998.41	

3: „Bestes“ Modell



Step: AIC=-1168.55

best model found by the
step() function and its AIC

durationLog ~ structure + vowel + rate

	Df	Sum of Sq	RSS	AIC
<none>			31.839	-1168.55
- rate	1	0.8416	32.681	-1158.86
- vowel	4	3.4070	35.246	-1131.01
- structure	2	14.9881	46.827	-999.73

additional proof that
further reduction is not
improving model fit

3: „Bestes“ Modell



call:

best model found by the
step() function and its call

```
lm(formula = durationLog ~ structure + vowel + rate, data = data)
```

Coefficients:

(Intercept)	structureopen	structuresingle	vowele
-1.5062	0.4340	0.1219	-0.1441
voweli	vowelo	vowelu	rate
-0.2374	-0.1229	-0.2365	-0.2532

model coefficients – we will
take a closer look in step 5

4: Assumptions überprüfen



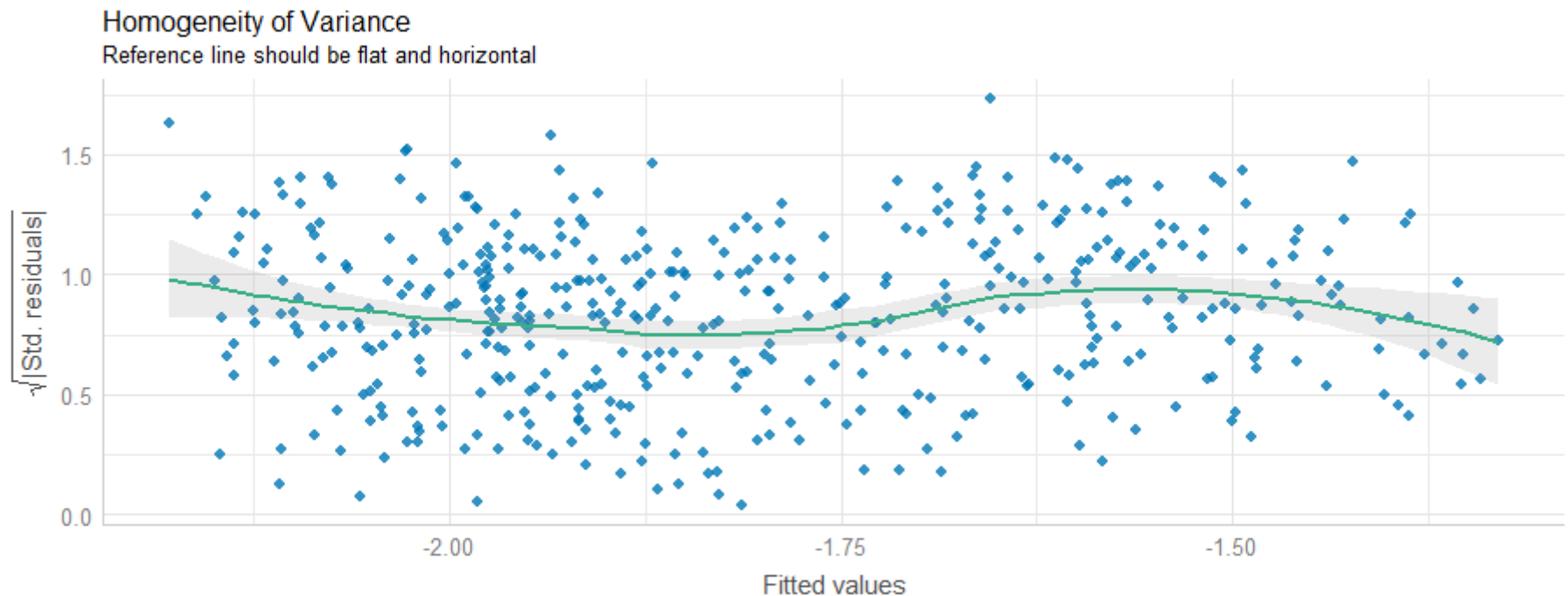
- Multiple Lineare Regression folgt den gleichen Assumptions, denen auch Simple Lineare Regression folgt
 - ▶ Linearity
 - ▶ Homoscedasticity
 - ▶ Normality
 - ▶ Independence

4: Assumptions überprüfen



- Linearity Assumption:

The relationship between X and the mean of Y is linear.



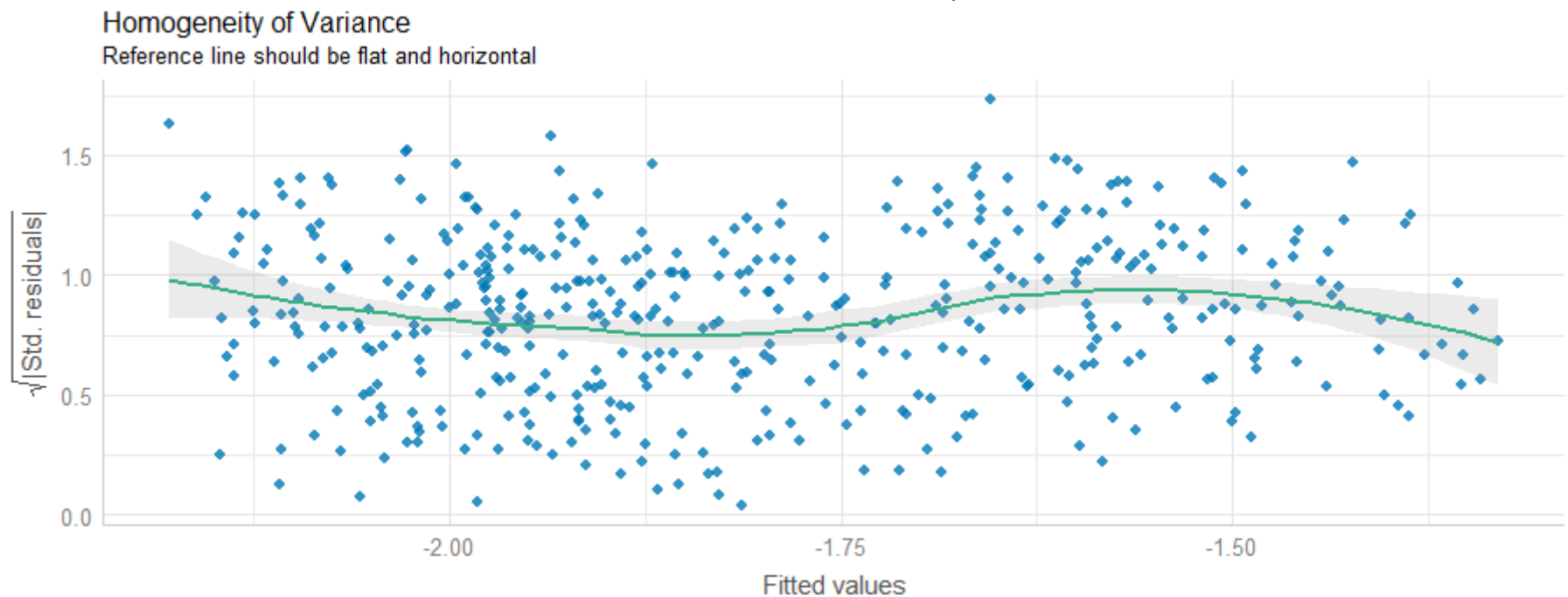
- The line should be horizontal and flat.

4: Assumptions überprüfen



- Homoscedasticity Assumption:

The variance of residuals is the same for any value of X.



- Data should be spread equally around the line, with no obvious patterns visible.

4: Assumptions überprüfen

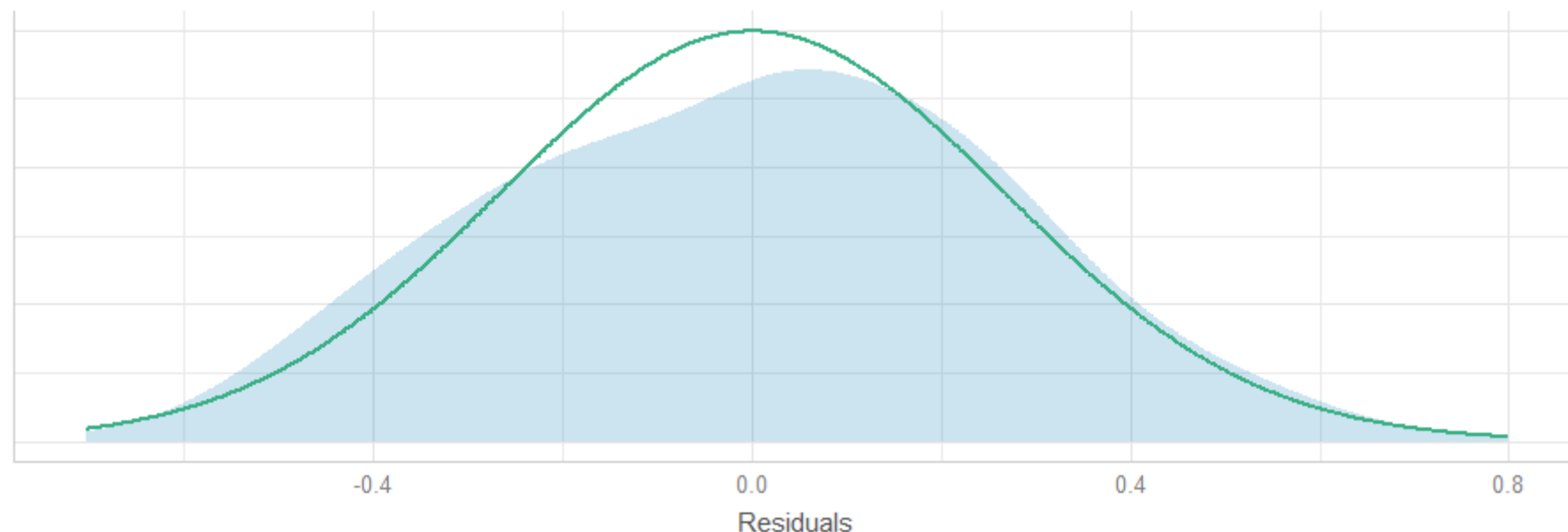


- Normality Assumption:

For any fixed value of X , Y is normally distributed.

Normality of Residuals

Distribution should be close to the normal curve



- The distribution of a linear model's residuals should follow a normal distribution.

5: Interpretation



- Generell sind wir an zwei Dingen interessiert:
 1. die p -Werte der einzelnen Predictors
 2. die Effekte der einzelnen Predictors

5: Interpretation – p -Werte



1. Mit der `anova()` Funktion erhalten wir p -Werte

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
structure	2	15.131	7.5654	104.4874	< 2.2e-16	***
vowel	4	3.507	0.8767	12.1079	2.41e-09	***
rate	1	0.842	0.8416	11.6241	0.0007112	***
Residuals	439	31.786	0.0724			

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.12186	0.03117	3.910	0.000107	***
vowele	-0.14406	0.04033	-3.572	0.000393	***
voweli	-0.23739	0.04035	-5.883	7.97e-09	***
vowelo	-0.12292	0.04034	-3.048	0.002446	**
vowelu	-0.23653	0.04033	-5.864	8.87e-09	***
rate	-0.25324	0.07425	-3.410	0.000708	***

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16 ***
structureopen	0.43395	0.05112	13.947	< 2e-16 ***
structuresingle	0			***
vowele	-0			***
voweli	-0			***
vowelo	-0			**
vowelu	-0			***
rate	-0.25524	0.07425	-3.410	0.000708 ***

contains the 'baseline' levels of all factors, i.e.

structure:double

vowel:a

plus the 'starting point' of the numerical predictor(s)

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.02112	13.947	< 2e-16	***
structuresingle	0.43395	0.02112	13.947	< 2e-16	***
vowele	-0.23739	0.04035	-5.883	7.97e-09	***
voweli	-0.23739	0.04035	-5.883	7.97e-09	***
vowelo	-0.12292	0.04034	-3.048	0.002446	**
vowelu	-0.23653	0.04033	-5.864	8.87e-09	***
rate	-0.25324	0.07425	-3.410	0.000708	***

structure:double + vowel:a + rate:start

estimated mean of durationLog

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.43395	0.03112	13.947	< 2e-16	***
vowele	-0.23739	0.04035	-5.883	7.97e-09	***
voweli	-0.12292	0.04034	-3.048	0.002446	**
vowelo	-0.23653	0.04033	-5.864	8.87e-09	***
vowelu	-0.25324	0.07425	-3.410	0.000708	***

structure:double + vowel:a + rate:start
standard error of that mean

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

structure:double + vowel:a + rate:start

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.12186	0.03117	3.910	0.000107	***
vowele	0.14406	0.04033	3.572	0.000393	***
voweli	0.14406	0.04033	3.572	0.000393	***
vowelo	0.14406	0.04033	3.572	0.000393	***
vowelu	0.14406	0.04033	3.572	0.000393	***
rate	0.14406	0.04033	3.572	0.000393	***

to obtain the estimated mean value of `durationLog` in `structure:single` words, we have to add its estimate to the intercept, i.e.

$$-1.50620 + 0.12186 = -1.38434$$

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

structure:single + vowel:a + rate:start

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16 ***
structureopen	0.43395	0.03112	13.947	< 2e-16 ***
structuresingle	0.12186	0.03117	3.910	0.000107 ***
vowele	-0.14406	0.04033	-3.572	0.000393 ***
voweli	-0.23739	0.04035	-5.883	7.97e-09 ***
vowelo				
vowelu				
rate				

to obtain the estimated mean value of `durationLog` in `structure:single` & `vowel:i` words, we have to add both estimates to the intercept, i.e.

$$-1.50620 + 0.12186 - 0.23739 = -1.62173$$

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

`structure:double + vowel:a + rate:start`

Estimate

(Intercept) -1.50620

structureopen 0.43395

structuresingle 0.12186

vowele -0.14406

voweli -0.23739

vowelo -0.12292

vowelu -0.23653

rate -0.25324

durationLog is

- significantly longer in open coda words

- significantly longer in simple coda words

than in complex coda words

**

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

`structure:double + vowel:a + rate:start`

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.12186	0.03117	3.910	0.000107	***
vowele	-0.14406				***
voweli	-0.23739				***
vowelo	-0.12292				**
vowelu	-0.23653				***
rate	-0.25324				***

durationLog is

- significantly shorter in words with all other vowels, i.e. /e, i, o, u/

than in words with /a/

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.12186	0.03117	3.910	0.000107	***
vowele	-0.14406	0.04033	-3.572	0.000393	***
voweli	0.1883	0.03112	6.048	7.97e-09	***
vowelo	0.1048	0.03117	3.364	0.002446	**
vowelu	0.23653	0.04033	-5.864	8.87e-09	***
rate	-0.25324	0.07425	-3.410	0.000708	***

the higher the speaking rate, the lower the value of `durationLog`

Step 5: Interpretation – Effects



2. Mit der `summary()` Funktion können wir einen Blick auf die einzelnen Effekte der Predictors werfen

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.50620	0.10486	-14.364	< 2e-16	***
structureopen	0.43395	0.03112	13.947	< 2e-16	***
structuresingle	0.12186	0.03117	3.910	0.000107	***
vowele	-0.14406	0.04033	-3.572	0.000393	***
voweli	-0.23739	0.04035	-5.883	7.97e-09	***
vowelo	-0.12292	0.04034	-3.048	0.002446	**
vowelu	-0.23653	0.04033	-5.864	8.87e-09	***
rate	-0.25324	0.07425	-3.410	0.000708	***

Step 5: Interpretation – Effects



Der s.g. „Tukey-Contrast“ zeigt uns die Unterschiede innerhalb eines kategorischen Predictors

```
> tukey(model = mdl_fin, predictor = structure)
```

	Estimate	Std. Error	t value	Pr(> t)	
open - double == 0	0.43395	0.03112	13.95	< 1e-04	***
single - double == 0	0.12186	0.03117	3.91	0.00031	***
single - open == 0	-0.31209	0.03111	-10.03	< 1e-04	***