Session 06:
Simple Lineare Regression

Dominic Schmitz & Janina Esser

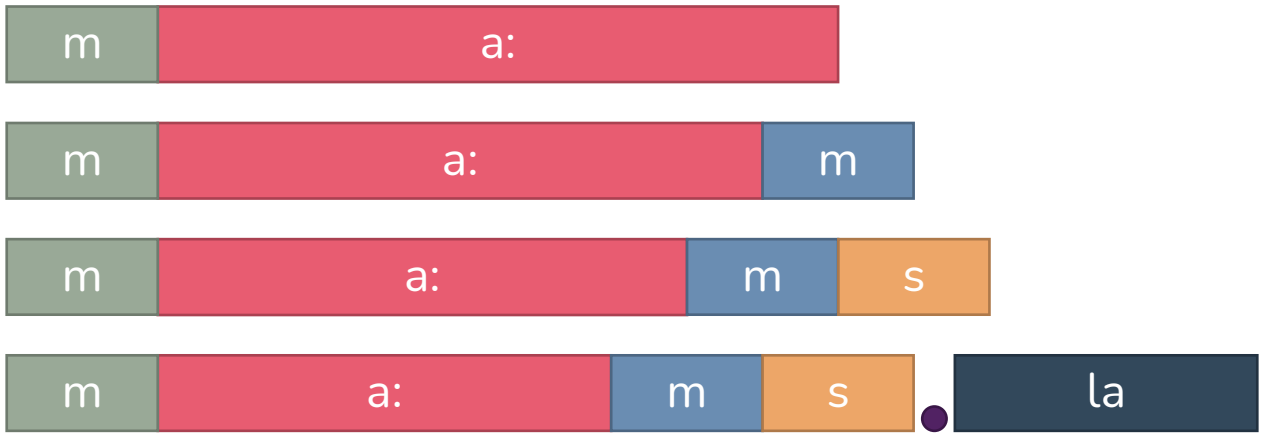
Verein für Diversität in der Linguistik

Beispieldaten

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

Compensatory Vowel Shortening in German[1]

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



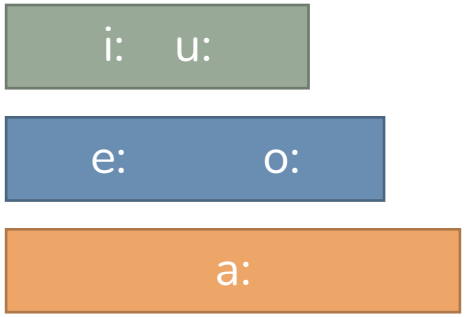
[1] Schmitz, D., Cho, H.-E., & Niemann, H. (2018). Vowel shortening in German as a function of syllable structure. Proceedings 13. Phonetik Und Phonologie Tagung (P&P13), 181–184.

Beispieldaten

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

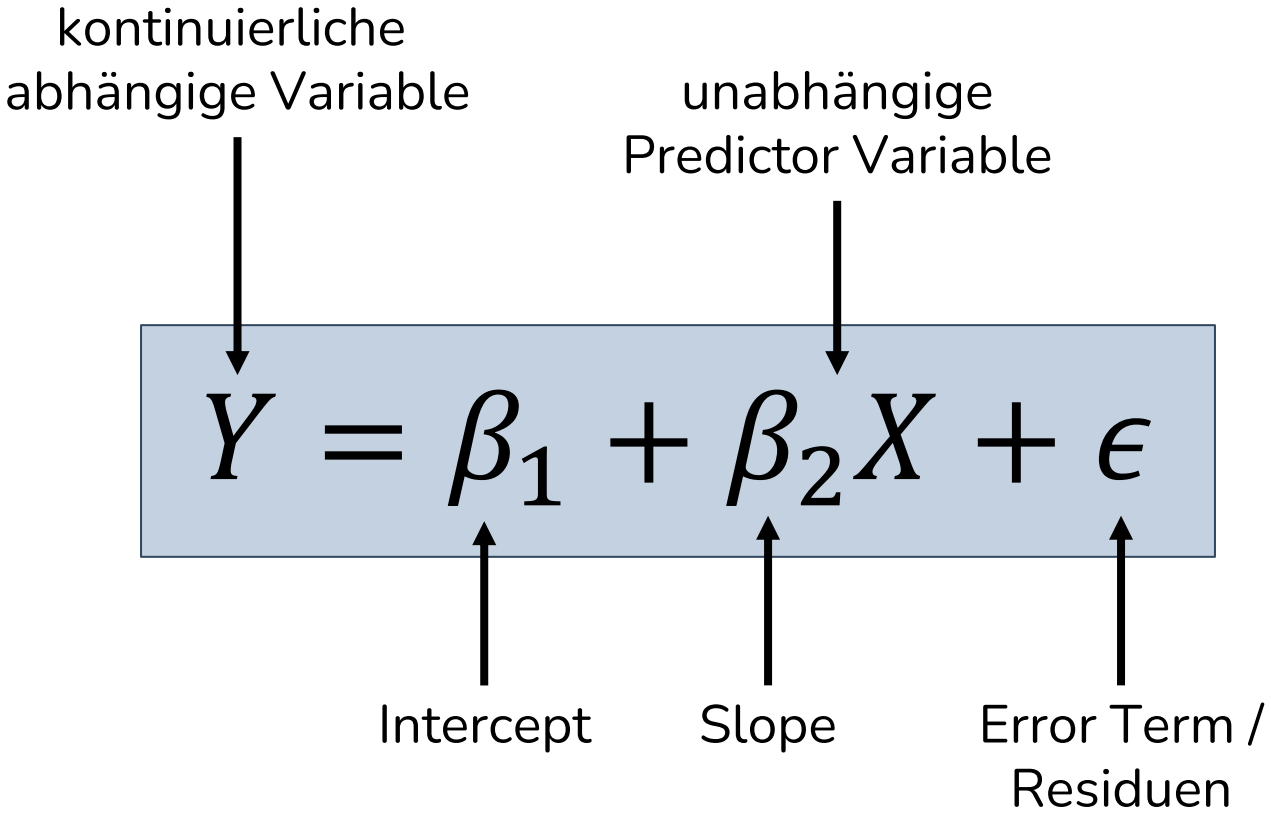
Compensatory Vowel Shortening in German[1]

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

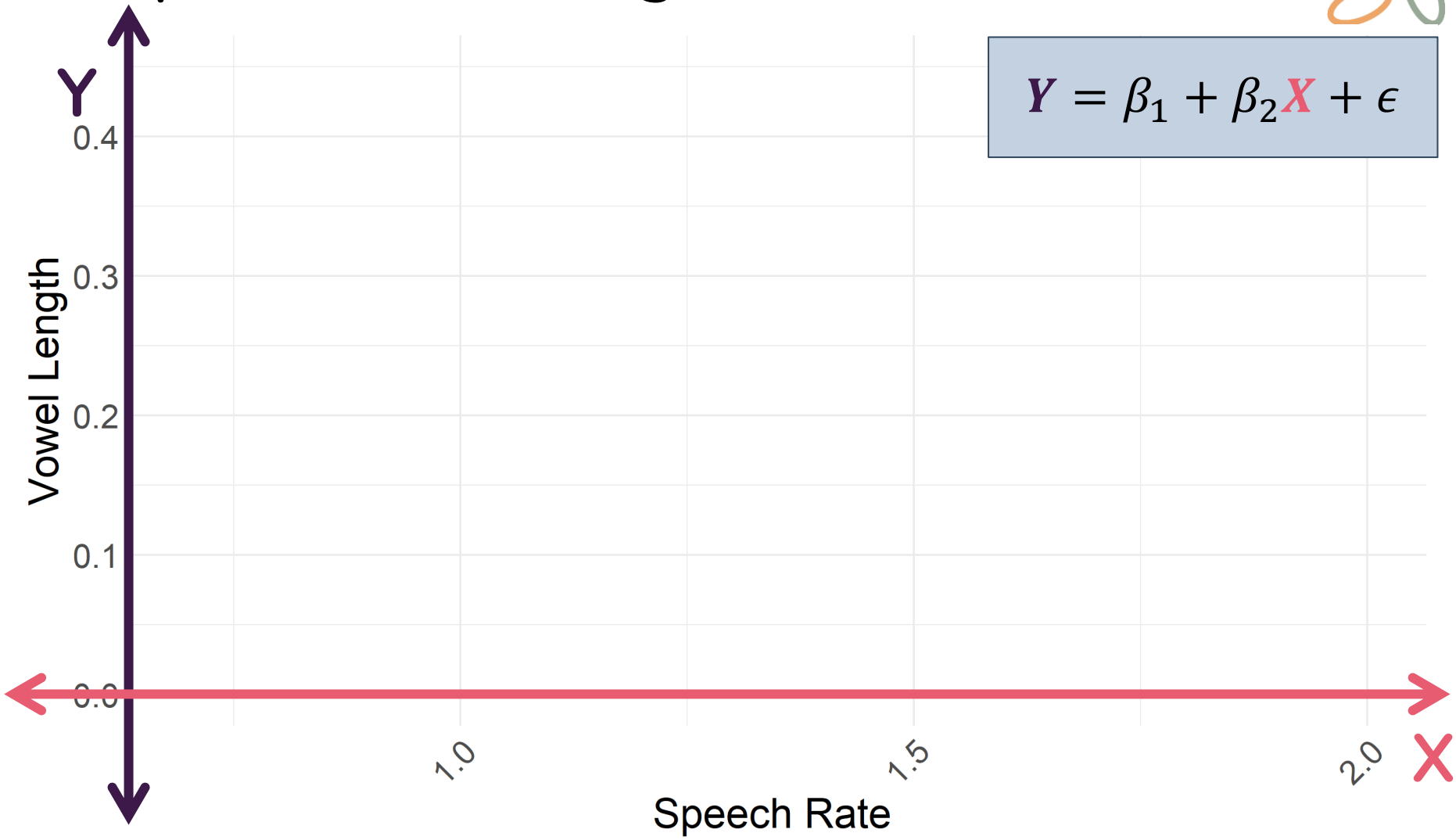


[1] Schmitz, D., Cho, H.-E., & Niemann, H. (2018). Vowel shortening in German as a function of syllable structure. Proceedings 13. Phonetik Und Phonologie Tagung (P&P13), 181–184.

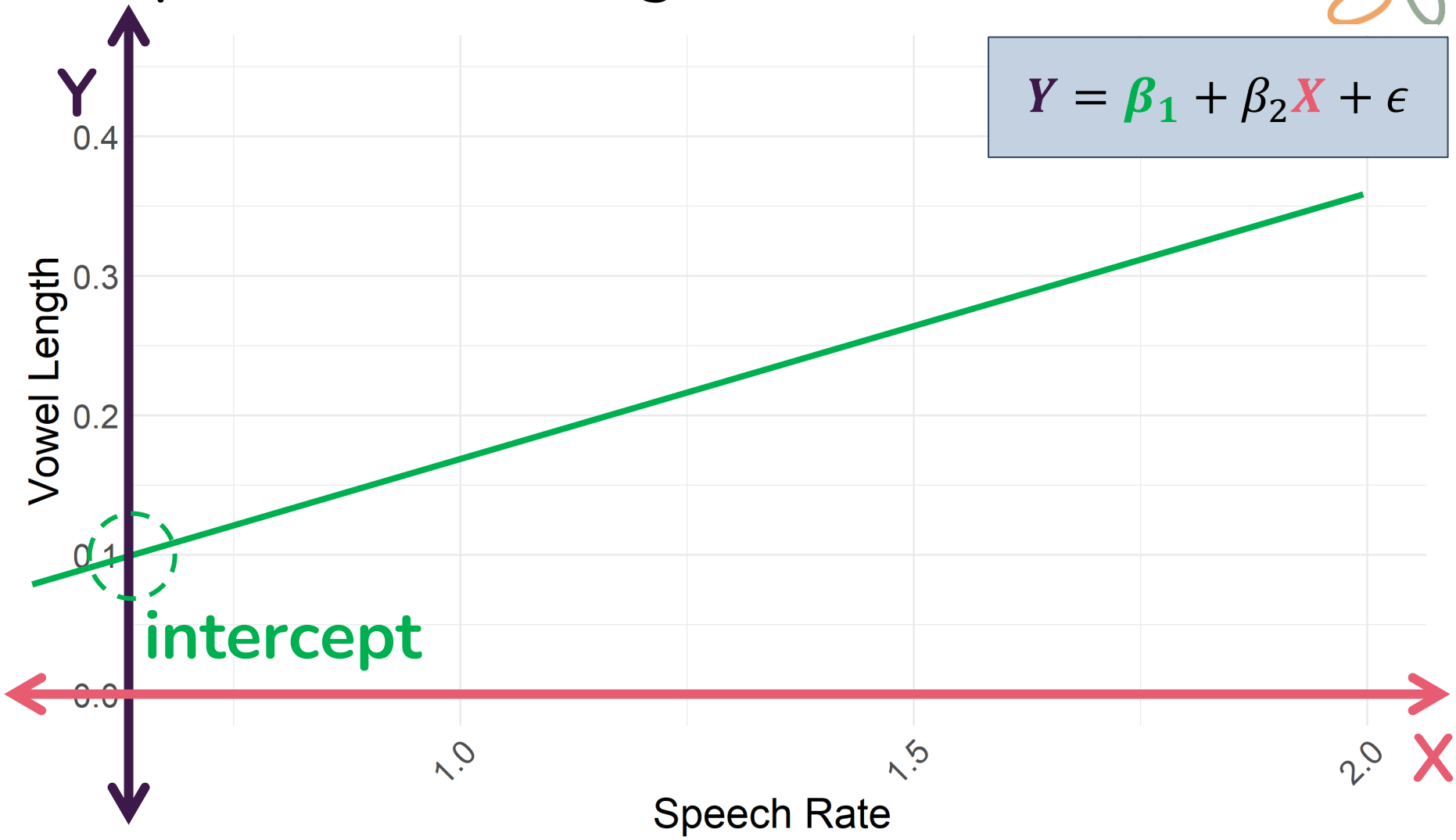
Simple Lineare Regression: Formel



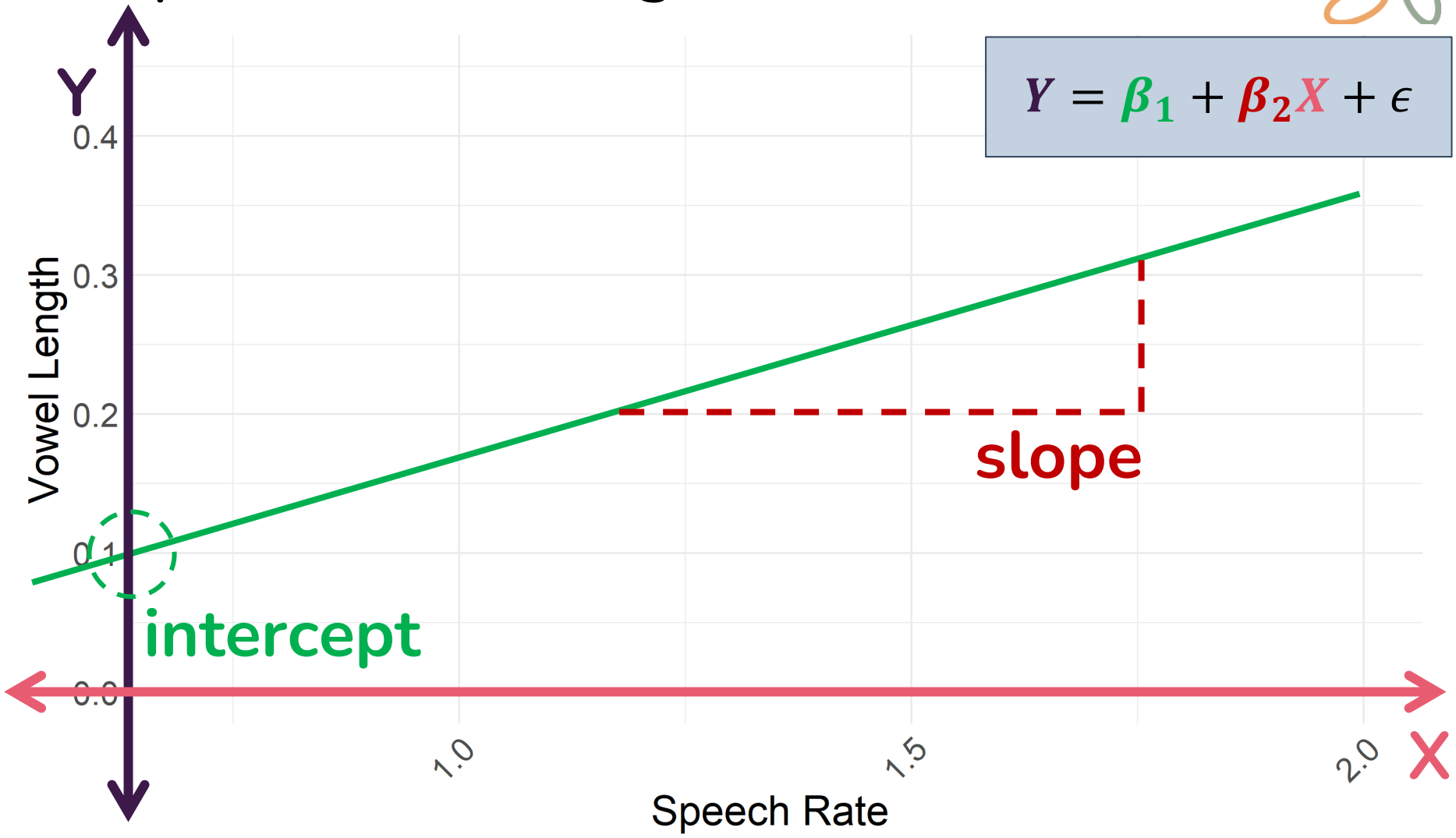
Simple Lineare Regression: Formel



Simple Lineare Regression: Formel

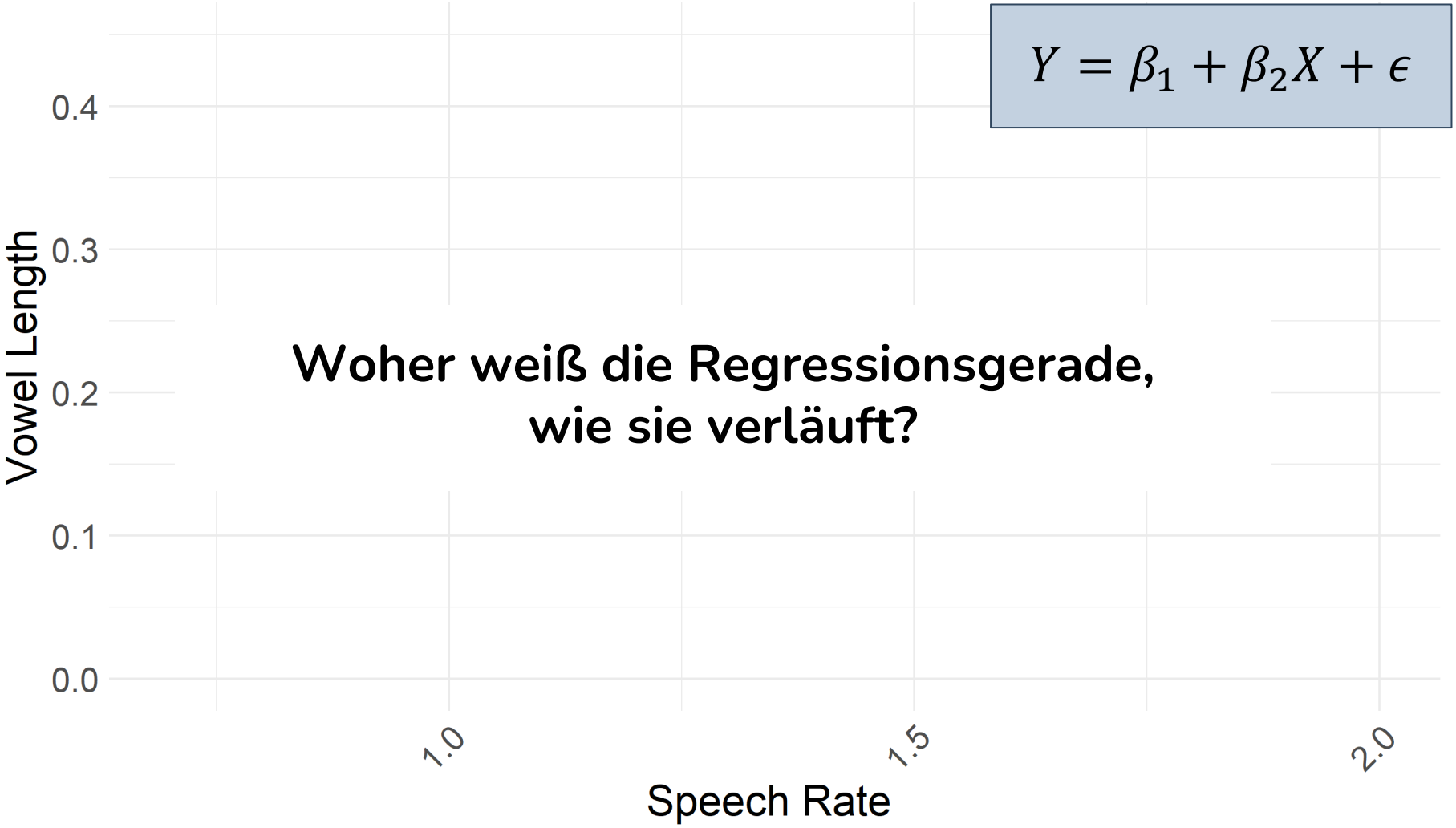


Simple Lineare Regression: Formel



Simple Lineare Regression: Formel

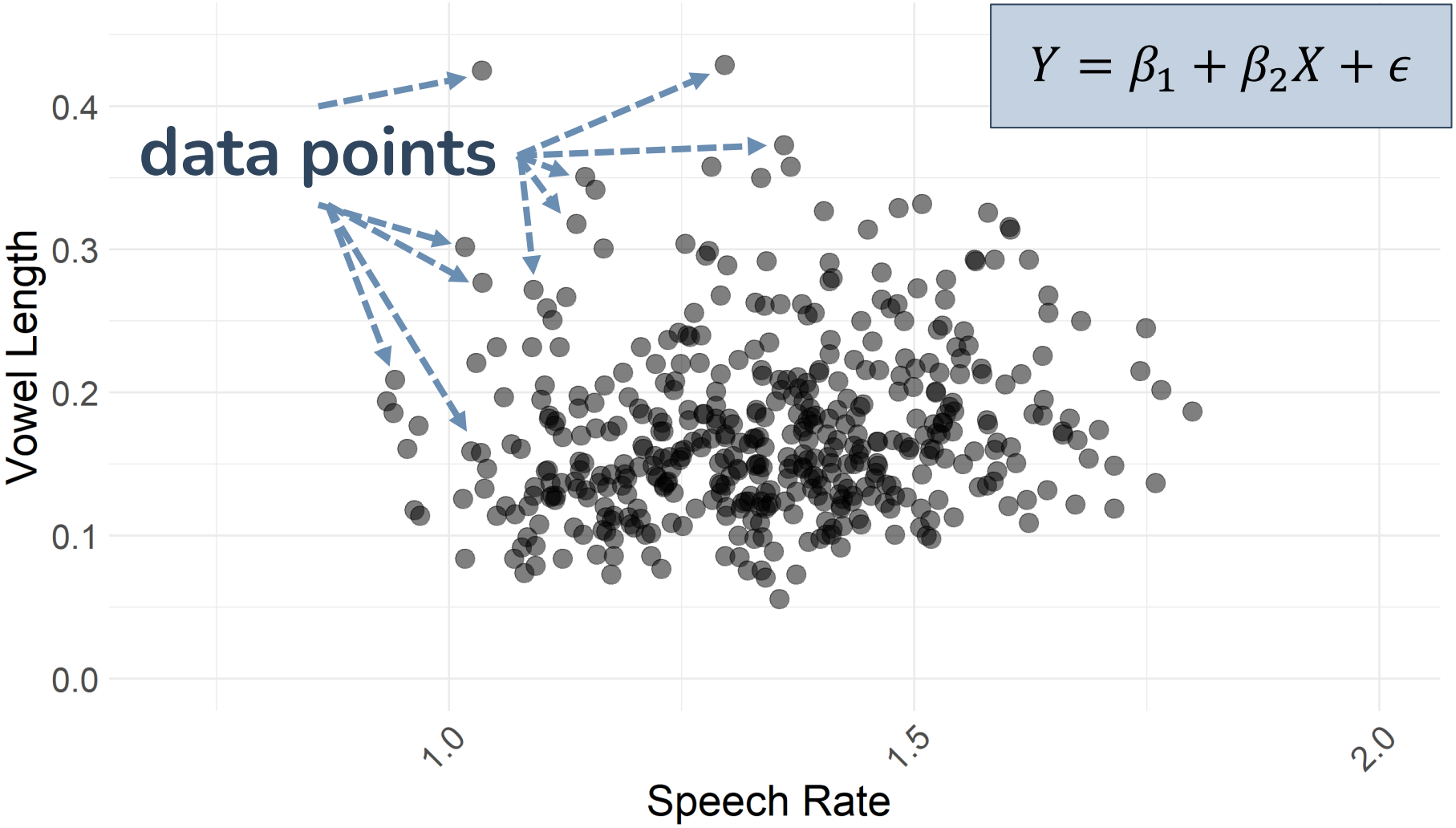
$$Y = \beta_1 + \beta_2 X + \epsilon$$



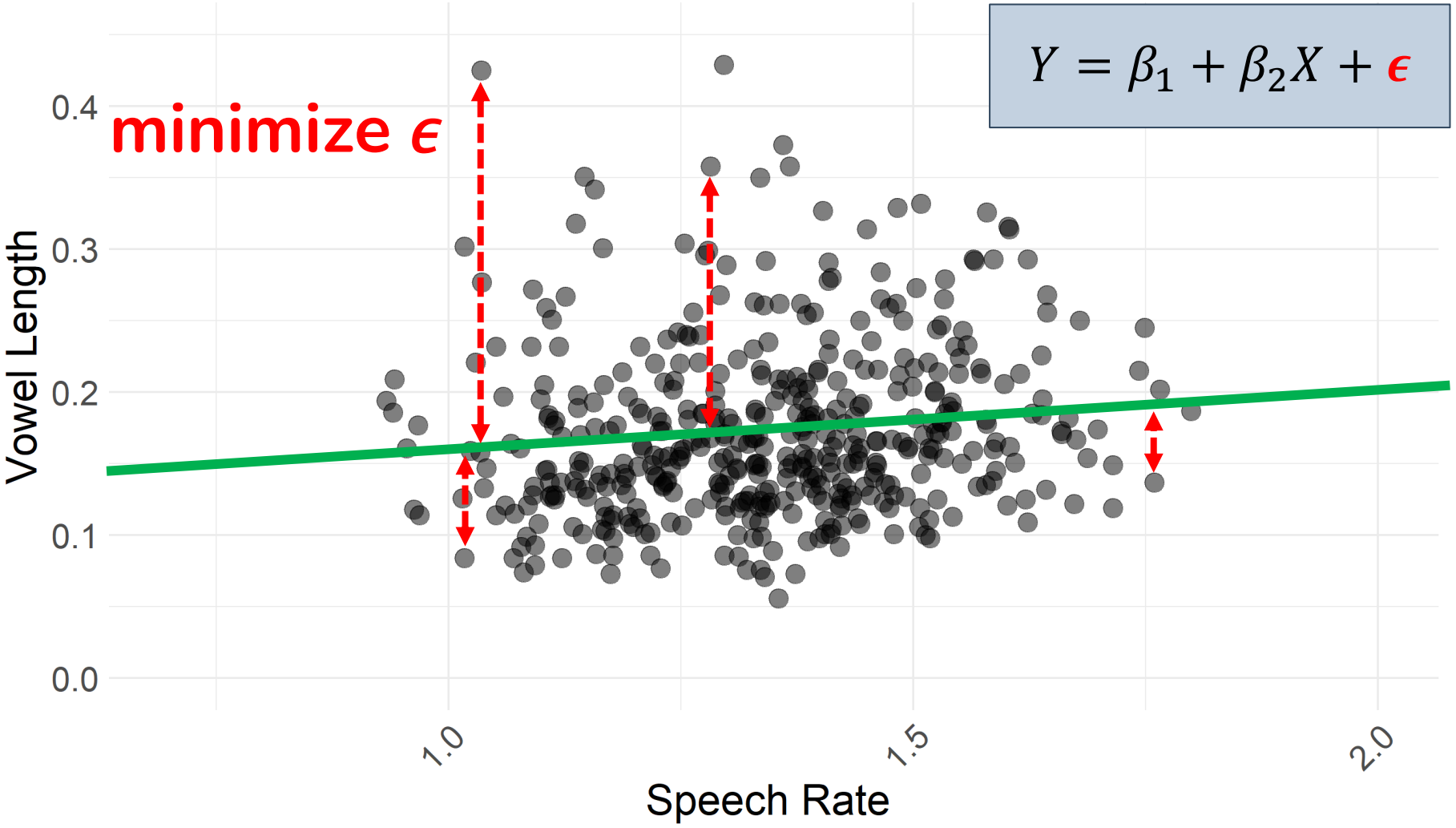
Woher weiß die Regressionsgerade, wie sie verläuft?

Simple Lineare Regression: Formel

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



Simple Lineare Regression: Formel



Simple Lineare Regression in R

- In R erstellt man ein simples lineares Regressionsmodell

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

- mit folgendem Befehl und folgender Syntax:

```
lm(Y ~ X, data)
```

- Intercept und Slope berechnet R indem es die Residuen zwischen tatsächlichen Datenpunkten und der Regressionsgeraden minimiert

Simple Lineare Regression in R

- Beispiel: vowel duration modelliert durch speech rate

```
model = lm(duration ~ rate, data)
```

- Nach der Berechnung erhalten wir folgende Information zum Modell:

```
Call:
lm(formula = duration ~ rate, data = data)

Coefficients:
(Intercept)         rate
    0.22301     -0.03687
```

Simple Lineare Regression in R

- Beispiel: vowel duration modelliert durch speech rate

```
model = lm(duration ~ rate, data)
```

- Nach der Berechnung erhalten wir folgende Information zum Modell:

```
Call:
lm(formula = duration ~ rate, data = data)

Coefficients:
(Intercept)         rate
    0.22301     -0.03687
```

intercept slope

Simple Lineare Regression in R

- Einen *p*-Wert erhalten wir mit der `anova()` Funktion:

```
anova(model)
Analysis of Variance Table

Response: duration
           Df  Sum Sq   Mean Sq F value  Pr(>F)
rate        1 0.01787 0.0178734  4.8468 0.02821 *
Residuals 446 1.64468 0.0036876
```

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rate	1	0.01787	0.0178734	4.8468	0.02821	*
Residuals	446	1.64468	0.0036876			

- **Degrees of Freedom**

  The number of independent pieces of information that went into calculating the estimate of said factor.

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rate	1	0.01787	0.0178734	4.8468	0.02821	*
Residuals	446	1.64468	0.0036876			

- **Squared Sum**

  The higher the value, the more important the factor is to the model.

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rate	1	0.01787	0.0178734	4.8468	0.02821	*
Residuals	446	1.64468	0.0036876			

- **Squared Mean**

  The higher the value, the more important the factor is to the model.

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rate	1	0.01787	0.0178734	4.8468	0.02821	*
Residuals	446	1.64468	0.0036876			

- **Fisher Value**

  The higher the value, the more influence the factor has on the dependent variable.

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
rate	1	0.01787	0.0178734	4.8468	0.02821 *
Residuals	446	1.64468	0.0036876		

- **Probability Value**

  Indicates whether an included factor has a significant influence on the dependent variable.

Simple Linear Regression in R

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rate	1	0.01787	0.0178734	4.8468	0.02821	*
Residuals	446	1.64468	0.0036876			

- **Residuals**

  The deviation/error not explained by the independent variables/factors.

  $\rightarrow \epsilon$

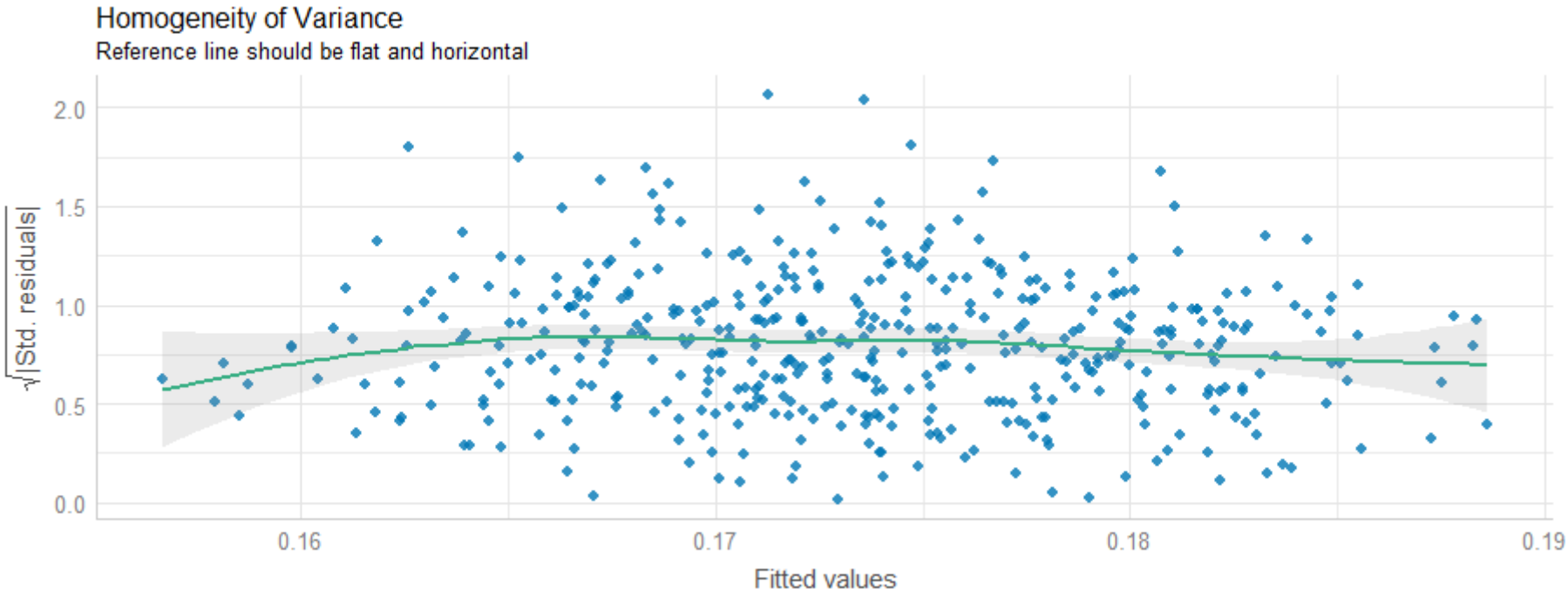
Assumptions

- Laut unserem Modell sinkt die Vowel Duration significant, wenn die Speaking Rate ansteigt
- Allerdings wissen wir gar nicht, ob unser Modell zuverlässig ist – wir haben nicht überprüft, ob es den **Assumptions** linearer Regression folgt:
  - Linearity
  - Homoscedasticity
  - Normality
  - Independence

Assumptions: Linearity

- Assumption:

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

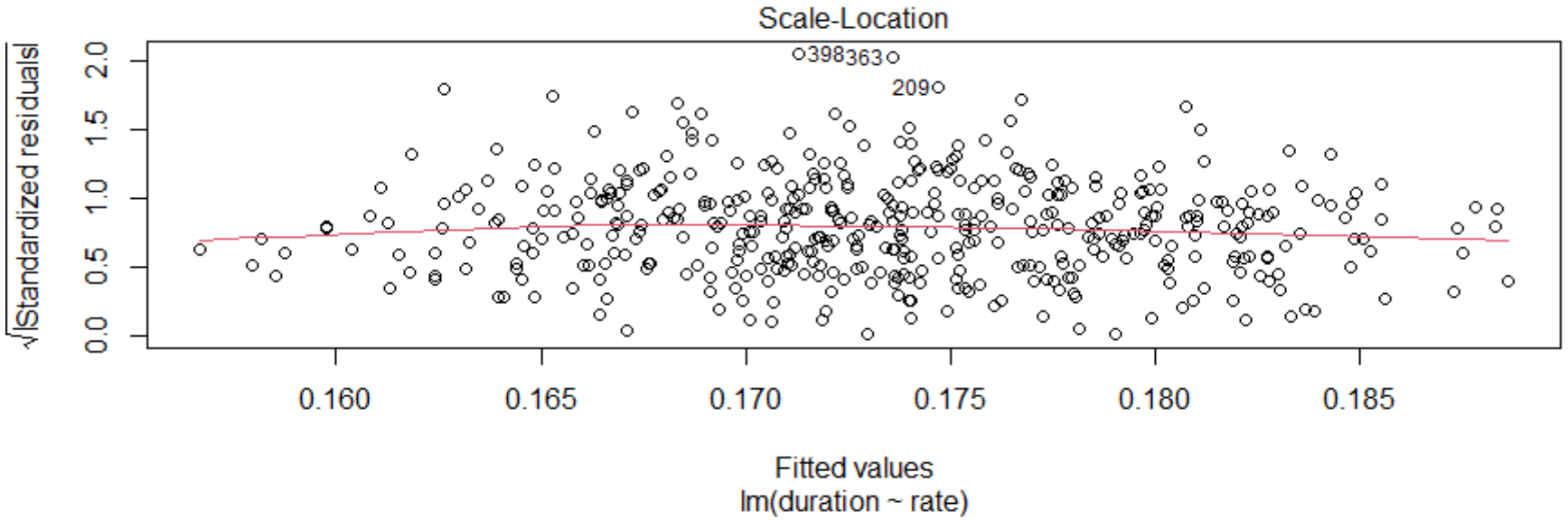


- The line should be horizontal and flat.

Assumptions: Homoscedasticity

- Assumption:

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

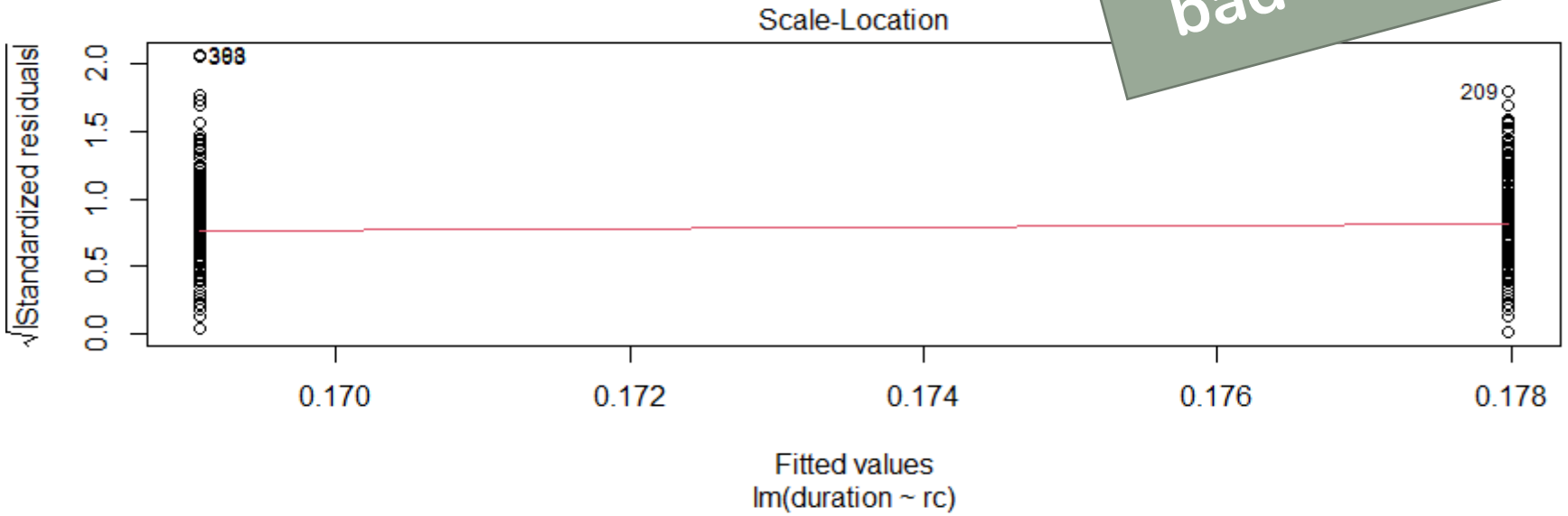


- Data should be spread equally around the line, with no obvious patterns visible.

Assumptions: Homoscedasticity

- Assumption:

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

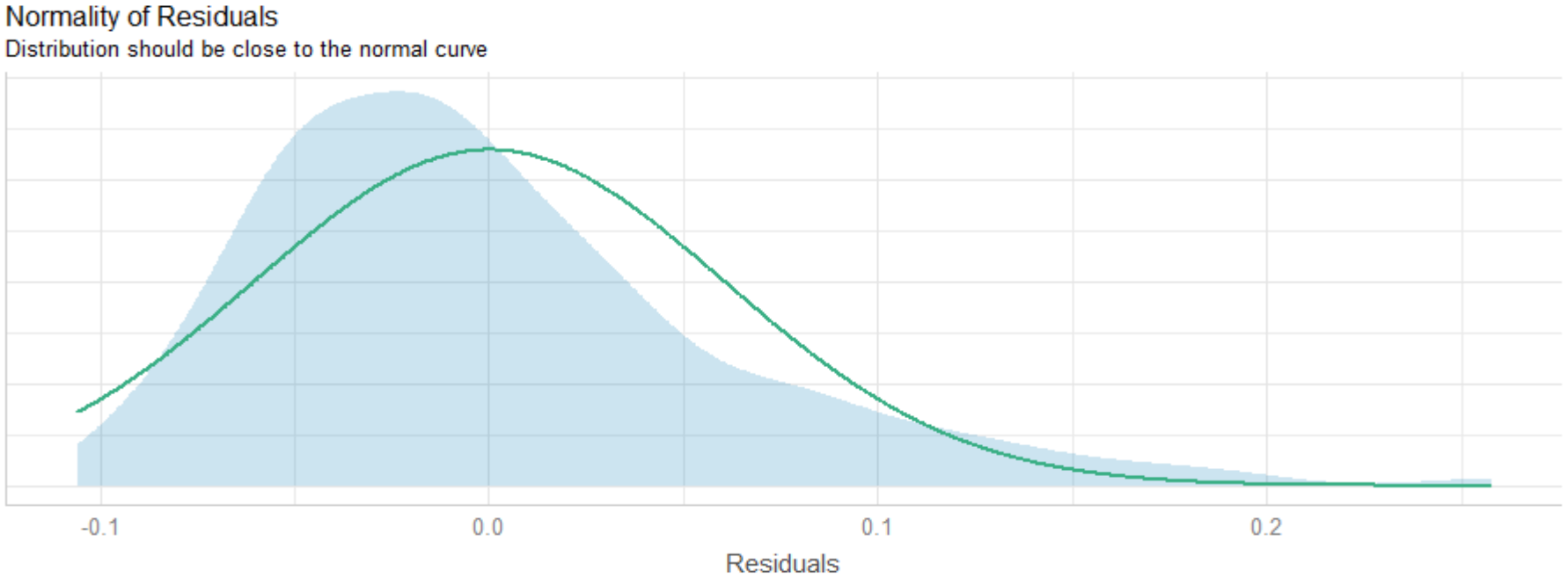


- Data should be spread equally around the line, with no obvious patterns visible.

Assumptions: Normality

- Assumption:

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

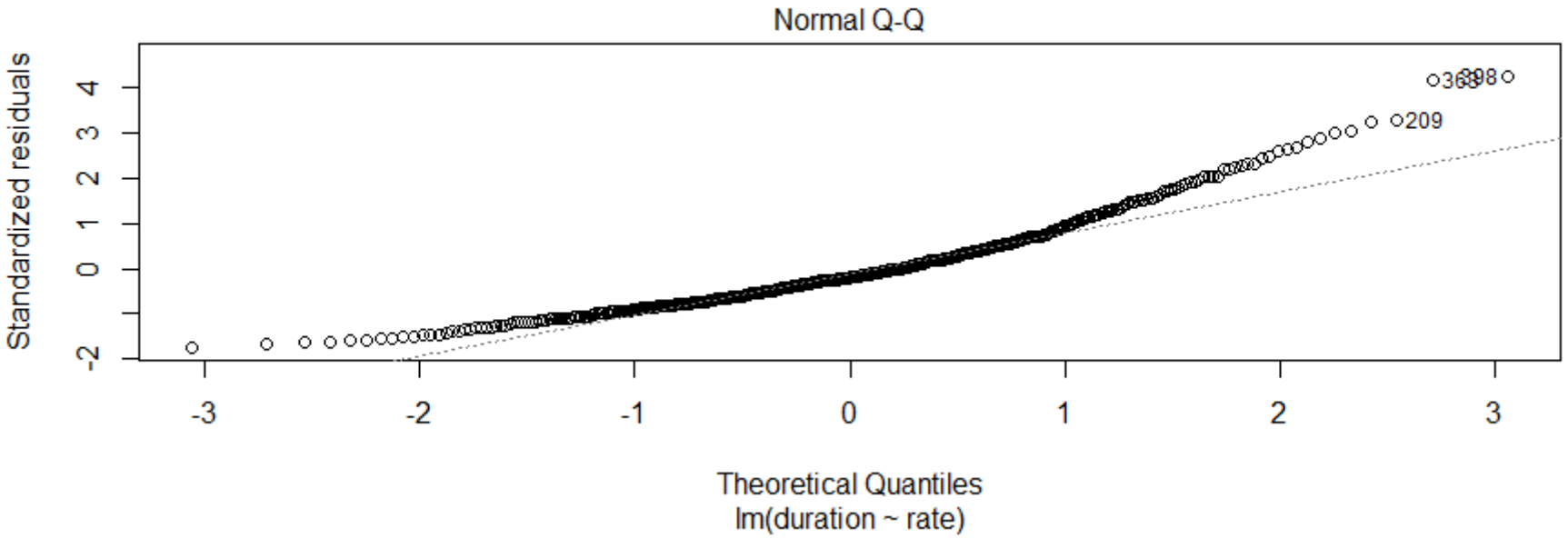


- The distribution of a linear model's residuals should follow a normal distribution.

Assumptions: Normality

- Assumption:

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



- Residual points should follow the line.

Assumptions: Independence

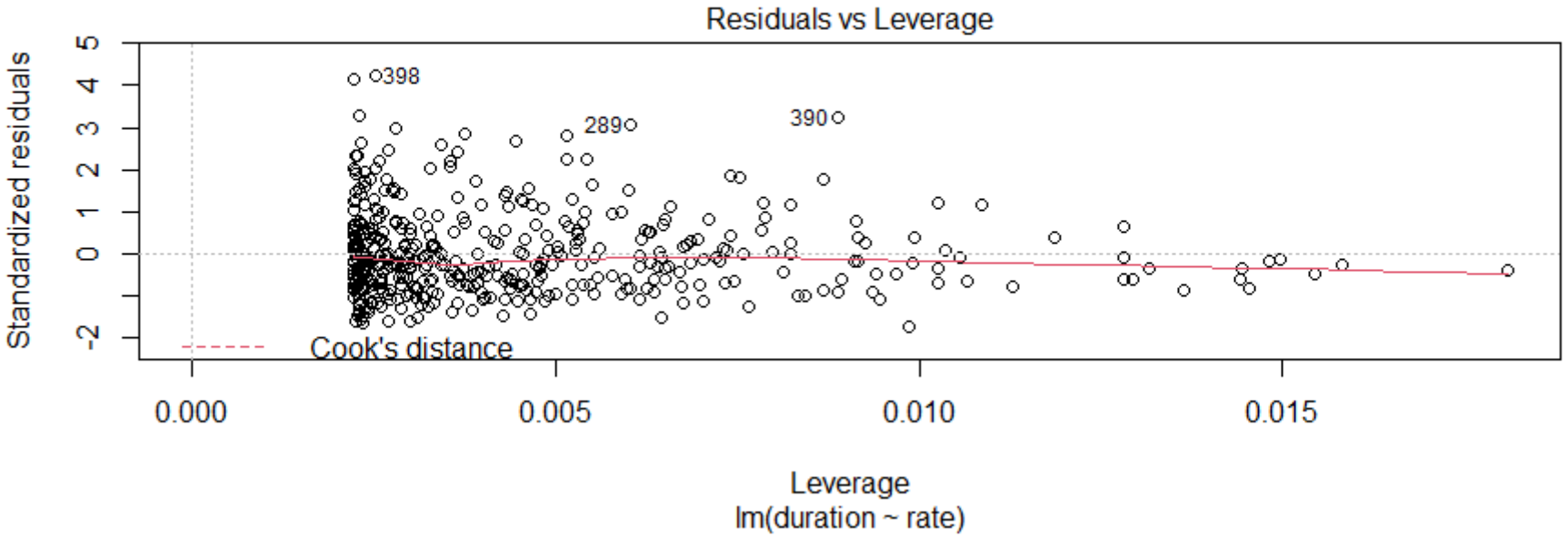
- Assumption:

  Observations are independent of each other.

- Independence cannot be checked visually
- It is an assumption that you can test by examining the study design

Extra: Influential Data Points

- Cook‘s Distance:
  - A measure of the influence of each observation on the regression coefficients
  - Any observation for which the Cook's distance is close to 1, or that is substantially larger than other Cook's distances requires investigation.



Dependent Variable Distribution Check

- Lineare Regressionsmodelle sind zuverlässiger, wenn ihre abhängige Variable einer Normaldistribution folgt
- Daher sollte man vor dem Erstellen von Modellen überprüfen, ob die abhängige Variable diese Voraussetzung erfüllt
- Falls die Verteilung fernab einer Normalverteilung ist, ist es ratsam die Variable zu transformieren
- In seltenen Fällen hilft keine Transformation dabei, die Variable näher an eine Normalverteilung zu bringen – hier kann Lineare Regression dennoch genutzt werden

Dependent Variable Distribution Check

- Wie wir bereits gelernt haben, kann man die Verteilung einer Variable mit einem Shapiro-Wilk Test überprüfen
- Je höher der *p*-Wert, desto normaler verteilt die Variable

```
shapiro.test(data$duration)

        Shapiro-Wilk normality test

data:  data$duration
W = 0.93844, p-value = 1.171e-12
```

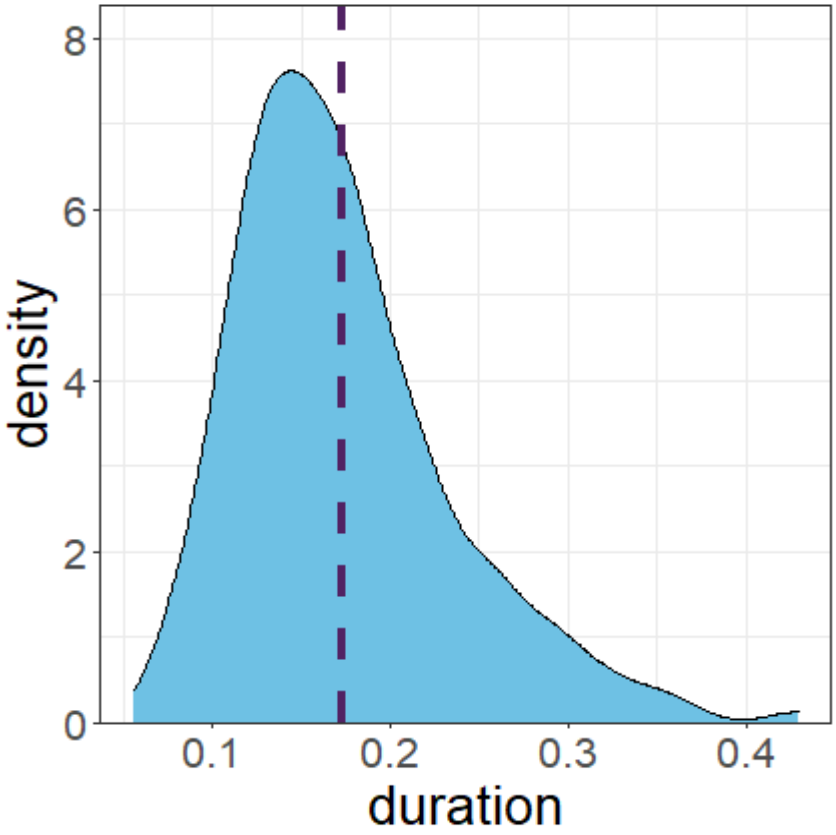
Dependent Variable Distribution Check

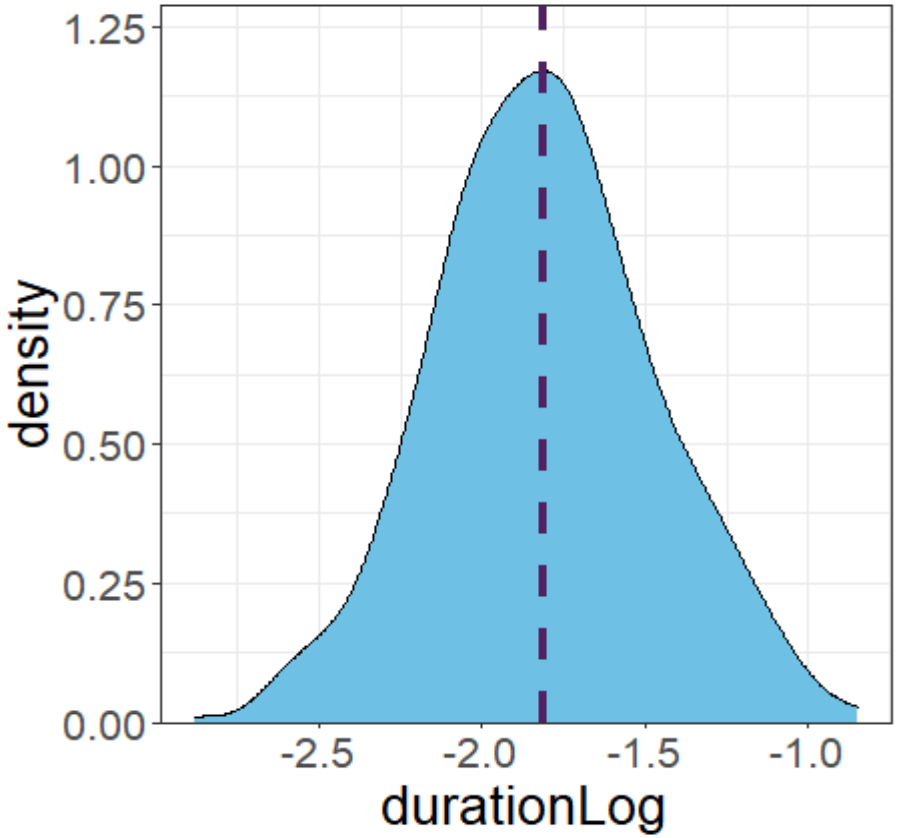
- Duration ist nicht normal verteilt; der *p*-Wert ist extrem niedrig
- Daher erstellen wir eine log-transformierte (= logarithmierte) Version

```
data$durationLog = log(data$duration)

shapiro.test(data$durationLog)

        Shapiro-Wilk normality test

data:  data$duration
W = 0.99762, p-value = 0.7798
```

Dependent Variable Distribution Check

- Eine Visualisierung zeigt deutlich, dass die transformierte Variable normalverteilter ist





Dependent Variable Distribution Check

- Wenn wir das zuvor erstellte Modell nun mit der log-transformierten Duration-Variable erneut erstellen, finden wir eine Verbesserung für die Normality of Residuals Assumption