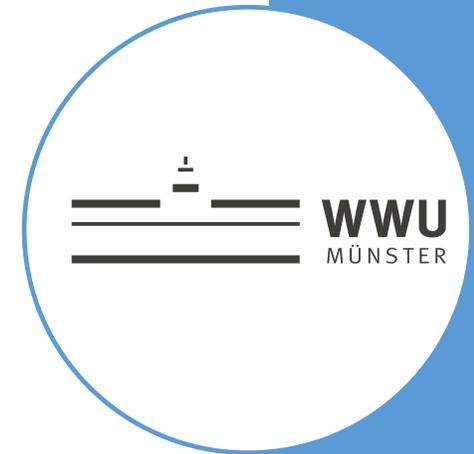


# Tutorial zum Erstellen eines psycholinguistischen Experiments

Vom Pretest bis zur Auswertung am Beispiel eines Priming-Experiments zur Verarbeitung ambiger Wörter

Dieses Tutorial ist im Rahmen des Master-Seminars „Psycholinguistik“, Wintersemester 2021/22, an der Universität Münster entstanden



# Erste Schritte

- Nachdem Du die **Forschungsliteratur** gesichtet hast und eine Fragestellung samt Hypothesen abgeleitet hast, musst Du Dich für eine Methode entscheiden
- Unsere **Fragestellung** lautete z. B.:
  - Wie wird die dominante Bedeutung ambiger Wörter (z. B. die Bedeutung *BESTECK* des Wortes *Gabel*) in Abhängigkeit der Bedeutungsfrequenz verarbeitet?
- Unsere **Hypothesen** waren die folgenden:
  - Ambige Wörter, deren dominante Lesart sehr frequent ist, werden wie nicht-ambige Wörter verarbeitet.
  - Ambige Wörter, deren dominante Lesart weniger frequent ist, werden langsamer als nicht-ambige Wörter verarbeitet.
  - Je weniger frequent die dominante Lesart eines ambigen Wortes ist, desto langsamer wird das Wort verarbeitet.
- Wir haben uns für ein Priming-Experiment mit einer lexical decision task entschieden
- Welche experimentelle Methode für Dich in Frage kommt, hängt von Deiner individuellen Fragestellung ab

# Erste Schritte

- Bei einem Priming-Experiment mit einer lexical decision task werden den ProbandInnen nacheinander Wortpaare präsentiert (z. B. *Besteck – Gabel*)
- Bei den Wortpaaren, die für unsere Fragestellung wichtig sind, besteht eine semantische Relation zwischen den Wörtern → das erste Wort (*Besteck*) ist mit der dominanten Bedeutung des zweiten, ambigen Wortes (*Gabel*) verbunden
- Wenn man also das Wort *Besteck* liest, sollte es einem leichter fallen, die damit verbundene Bedeutung des Wortes *Gabel* zu aktivieren (anders wäre es, wenn man das Wort *Besteck* durch das Wort *Fahrrad* ersetzen würde)
- Um unser Material für das Experiment zu erstellen, brauchen wir also folgendes:
  - Ambige Wörter, die auch wirklich ambig sind
  - Informationen dazu, welche die dominante, also die stärkste Bedeutung eines ambigen Wortes ist und wie stark, also wie frequent diese Bedeutung tatsächlich ist
- Um an diese Informationen zu kommen, starten wir mit einem **Pretest**

# 1. Pretest

# Was ist ein Pretest?

- Mit einem Pretest wird im Allgemeinen die Durchführbarkeit eines Experiments getestet; er dient der Verbesserung der Qualität von Erhebungsinstrumenten (wie z. B. Fragebögen) und des Forschungsdesigns
- Hier meinen wir mit Pretest aber etwas anderes: Wir führen einen Test durch, um Daten zu erheben, die wir für die Erstellung unseres Wortmaterials benötigen
- Wir wollen mit unserem Pretest herausfinden, ob ambige Wörter wirklich ambig sind und wie frequent ihre dominante Lesart ist
- Dafür haben wir uns zunächst ambige und nicht-ambige Wörter überlegt und diese in einer Excel-Tabelle zusammengetragen

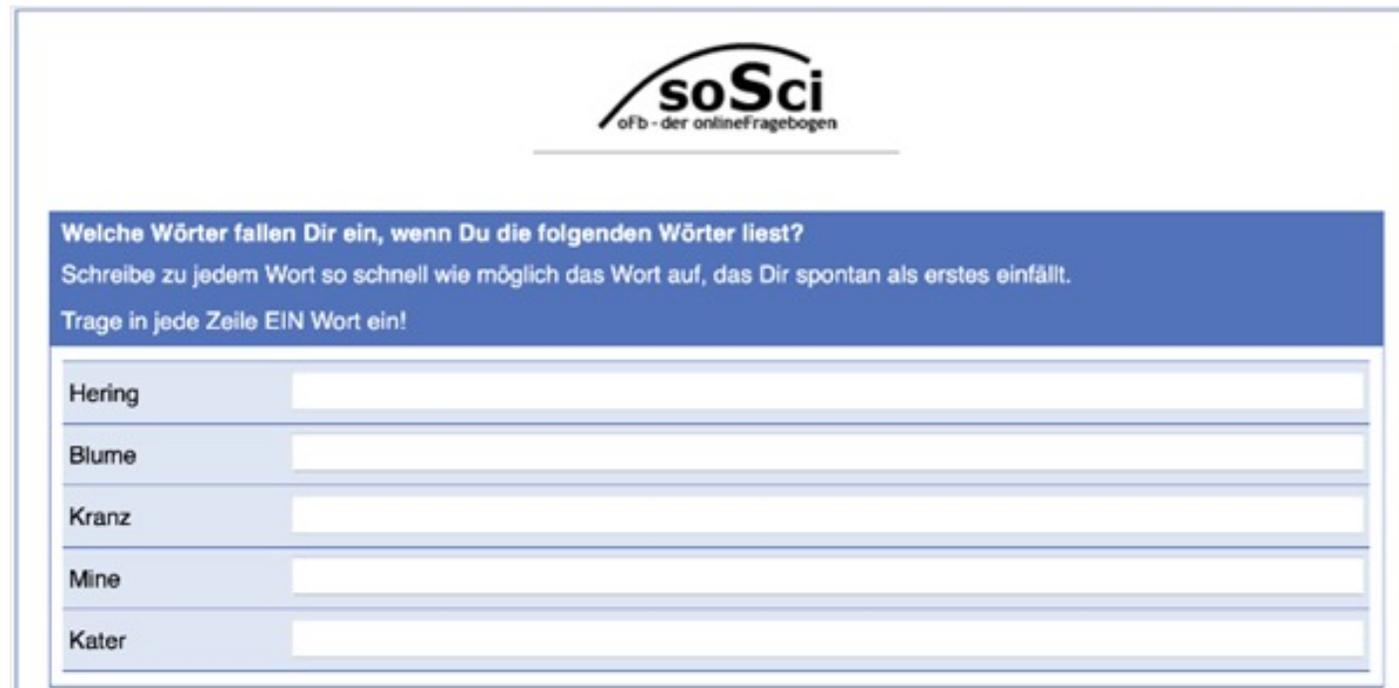
- **Schritt 1: Sammlung von Untersuchungsmaterialien** (z. B. Sammlung ambiger und nicht-ambiger Wörter in einer Excel-Tabelle)

|    | A        | B       | C        | D     | E      | F      | G                 | H                   | I                   | J             |
|----|----------|---------|----------|-------|--------|--------|-------------------|---------------------|---------------------|---------------|
| 1  | Wort     | Wortart | Frequenz | Länge | Silben | Status | dominante Le      | Assoziation         | Assoziation_2       | Art der Ambig |
| 2  | Absatz   | Nomen   | 4879     | 6     | 2      | ambig  | Text              | Computer            | Schuh               | Homonym       |
| 3  | Abstand  | Nomen   | 2898     | 7     | 2      | ambig  | Raum              | Distanz             | Zurückhaltung       | Polysem       |
| 4  | Abzug    | Nomen   |          | 5     | 2      | ambig  | Dampfzug          | Waffe               |                     | Homonym       |
| 5  | Akt      | Nomen   | 3836     | 3     | 1      | ambig  | Handlung          | Theater             | Kunst               | Homonym       |
| 6  | Ampel    | Nomen   |          | 5     | 2      | ambig  | Verkehrsampel     | Blumenampel         | Auto                | Polysem       |
| 7  | Angel    | Nomen   |          | 5     | 2      | ambig  | Fischen           | bei Tür und Fenster |                     | Homonym       |
| 8  | Anhänger | Nomen   |          | 8     | 3      | ambig  | Schlüsselanhänger | Schlüsselanhänger   | Mitglied            | Polysem       |
| 9  | Apfel    | Nomen   |          | 5     | 2      | ambig  | Obst              | Obst                | Auge; Pferdekot     | Polysem       |
| 10 | Ball     | Nomen   | 1388     | 4     | 1      | ambig  | Spielutensil      | Tor                 | Tanz                | Homonym       |
| 11 | Band     | Nomen   | 4859     | 4     | 1      | ambig  | Faden             | Gummi               | Buch                | Homonym       |
| 12 | Bande    | Nomen   |          | 5     | 2      | ambig  | Gruppe            | Gruppe              | Spielfeldbegrenzung | Homonym       |
| 13 | Bank     | Nomen   | 6573     | 4     | 1      | ambig  | Sitzgelegenheit   | Stuhl               | Geld                | Homonym       |
| 14 | Bar      | Nomen   | 752      | 3     | 1      | ambig  | Gaststätte        | Cocktail            | Druck               | Homonym       |
| 15 | Barren   | Nomen   |          | 6     | 2      | ambig  | Turngerät         | Goldbarren          | Turnen              | Homonym       |
| 16 | Bar      | Nomen   |          | 4     | 1      | ambig  | im Gesicht        | Schnurrbart         | am Schlüssel        | Homonym       |

Achtung! Man arbeitet immer im Spalten-Format; so kann man jedem Wort beliebig viele Eigenschaften zuweisen (z. B. Frequenz, Länge, etc.), wobei jede neue Eigenschaft in einer neuen Spalte hinzugefügt wird

Achtung! Da man anschließend viele Wörter aussortiert, sollte man in der ersten Runde wesentlich mehr Wörter zusammentragen, als man letztendlich braucht

- Schritt 2: Auswahl bestimmter Wörter aus den Untersuchungsmaterialien (z. B. Auswahl ambiger und nicht-ambiger Wörter mit 1-2 Silben und 4-6 Buchstaben)
- Schritt 3: Überlegungen zum Pretest-Aufbau (z. B. zwei verschiedene Listen mit je 50 Wörtern; pro Liste 10 Seiten mit 5 Wörtern/Seite; keine zeitliche Begrenzung; intuitives, manuelles Aufschreiben der ersten Assoziation mit dem gegebenen Wort)
- Schritt 4: Programmierung des Pretests (z. B. mit SoSci-Survey, siehe Abbildung)



The image shows a screenshot of a web-based survey interface. At the top center is the logo for 'soSci ofb - der onlineFragebogen'. Below the logo is a blue header bar containing the following text: 'Welche Wörter fallen Dir ein, wenn Du die folgenden Wörter liest?' followed by 'Schreibe zu jedem Wort so schnell wie möglich das Wort auf, das Dir spontan als erstes einfällt.' and 'Trage in jede Zeile EIN Wort ein!'. Below the header is a table with five rows, each containing a word and an empty text input field for the response.

| Welche Wörter fallen Dir ein, wenn Du die folgenden Wörter liest?                                |                      |
|--|----------------------|
| Schreibe zu jedem Wort so schnell wie möglich das Wort auf, das Dir spontan als erstes einfällt. |                      |
| Trage in jede Zeile EIN Wort ein!  |                      |
| Hering   | <input type="text"/> |
| Blume  | <input type="text"/> |
| Kranz  | <input type="text"/> |
| Mine   | <input type="text"/> |
| Kater  | <input type="text"/> |

# SoSci

Link: [soscisurvey.de](https://soscisurvey.de)



## SoSci Survey – die Lösung für eine professionelle Onlinebefragung

Sie möchten eine professionelle Onlinebefragung einfach, schnell und zuverlässig durchführen? SoSci Survey bietet genau das als White Label Lösung, nach deutschem Datenschutz und barrierefrei.

Und wenn Sie einen anspruchsvolleren Onlinefragebogen benötigen, kann SoSci Survey seine Stärken erst richtig ausspielen: Einbindung von Bildern, Audio und Video, frei programmierbare Filterführung, vollständig anpassbare Layouts, kontrollierte Randomisierung für wissenschaftliche Experimente, u.v.m.



- 
- ✓ Server in Deutschland
  - ✓ Betreiber in Deutschland
  - ✓ Software aus Deutschland
-

- Schritt 5: Auswertung der Ergebnisse (z. B. manuelle Kategorisierung der genannten Assoziationen in verschiedene Lesarten; Spalte J: Assoziationen der einzelnen ProbandInnen zum Wort *Hering*, Spalte K: manuelle Kategorisierung der einzelnen Assoziationen in die Lesarten *Fisch* oder *Zelt*)

|    | J         | K      | L           | M       | N           | O           | P         | Q        | R        | S       |
|----|-----------|--------|-------------|---------|-------------|-------------|-----------|----------|----------|---------|
| 1  | 1: Hering | Lesart | 1: Blume    | Lesart  | 1: Kranz    | Lesart      | 1: Mine   | Lesart   | 1: Kater | Lesart  |
| 2  | Fisch     | Fisch  | Pflanze     | Pflanze | Weihnachten | Blumenkranz | Bombe     | Bombe    | Tier     | Tier    |
| 3  | Fisch     | Fisch  | Sonnenblume | Pflanze | Advent      | Blumenkranz | Arbeit    | Berg     | Alkohol  | Alkohol |
| 4  | Fisch     | Fisch  | Baum        | Pflanze | Advent      | Blumenkranz | Kohle     | Berg     | Alkohol  | Alkohol |
| 5  | Fisch     | Fisch  | Tulpe       | Pflanze | Weihnachten | Blumenkranz | Gold      | Berg     | Katze    | Tier    |
| 6  | Fisch     | Fisch  | Rose        | Pflanze | Grab        | Blumenkranz | Meins     | Englisch | Katze    | Tier    |
| 7  | Fisch     | Fisch  | Rose        | Pflanze | Hochzeit    | Blumenkranz | Gesicht   | Gesicht  | Katze    | Tier    |
| 8  | Angel     | Fisch  | Biene       | Pflanze | Advent      | Blumenkranz | Kohle     | Berg     | Garfield | Tier    |
| 9  | Zelten    | Zelt   | Gelb        | Pflanze | Aufhängen   | Blumenkranz | Kulli     | Stift    | Alkohol  | Alkohol |
| 10 | Fisch     | Fisch  | Sonnenblume | Pflanze | Advent      | Blumenkranz | Bombe     | Bombe    | Alkohol  | Alkohol |
| 11 | Rollmops  | Fisch  | Biene       | Pflanze | Kerzen      | Blumenkranz | Bergbau   | Berg     | Alkohol  | Alkohol |
| 12 | Fisch     | Fisch  | Bunt        | Pflanze | Advent      | Blumenkranz | Bleistift | Stift    | Alkohol  | Alkohol |
| 13 | Fisch     | Fisch  | Frisch      | Pflanze | Weihnachten | Blumenkranz | Kohle     | Berg     | Katze    | Tier    |
| 14 | Fisch     | Fisch  | Rose        | Pflanze | Haar        | Blumenkranz | Gefahr    | Bombe    | Katze    | Tier    |
| 15 | Fisch     | Fisch  | Wiese       | Pflanze | Advent      | Blumenkranz | Bleistift | Stift    | Tier     | Tier    |



## 2. Erstellung des Experiments

# Was benötigst Du für die Durchführung Deines Experiments?

- Material, das Du untersuchen möchtest (z. B. Liste ambiger Wörter, basierend auf dem Pretest)
- Material, das als Kontrollbedingung dient (z. B. Liste nicht-ambiger Wörter)
- Material, das als Füllmaterial dient, um von der eigentlichen Aufgabe abzulenken (z. B. Liste mit Nicht-Wörtern)
- ProbandInnen
- Experiment-Software (z. B. *Gorilla* für Experimente, die man online durchführen kann)
- Bei Experimenten in Präsenz: Ein Labor/ einen ruhigen Raum, einen Computer
- Evtl. Geld, um die ProbandInnen auszuzahlen

# Aufbau unseres Priming Experiments

- Bei einem Priming-Experiment braucht man Wortpaare: ein Prime und ein Target (z. B. *Besteck – Gabel*)
- Prime = Wort, welches semantisch mit der dominanten Bedeutung des ambigen Wortes in Verbindung steht; darf nicht ambig sein
- Target = ambiges Wort, dessen dominante Bedeutung durch das Prime aktiviert werden soll
- Um die Verarbeitung der ambigen Wörter mit der Verarbeitung nicht ambiger Wörter zu vergleichen, haben wir außerdem Wortpaare integriert, deren Targets nicht ambig waren (z. B. *All – Mond*)
- Die Aufgabe der ProbandInnen bestand darin, zu entscheiden, ob es sich bei dem Target um ein Wort handelt oder nicht; aus diesem Grund brauchten wir zur Hälfte Wortpaare, deren Targets Wörter waren und zur Hälfte Wortpaare, deren Targets Nicht-Wörter waren
- Nicht-Wörter = Wort-Targets, bei denen jeweils ein Buchstabe verändert wurde, um ein Nicht-Wort zu kreieren, was ein potenzielles Wort des Deutschen sein kann (z. B. *Gubel*)

# Aufbau unseres Priming Experiments

- *Pretask* (Übungsdurchlauf)
  - 2 Paare mit Wort-Targets und 2 Paare mit Nicht-Wort-Targets (z. B. *Streifen – Zebra, Brille – Zubra*)
- *Main task* (eigentliches Experiment)
  - 30 ambige Wörter als Targets; Gruppierung nach Dominanz, basierend auf dem Pretest
    - 10 Wörter stark dominant (90–100%)
    - 10 Wörter mittel dominant (70–80%)
    - 10 Wörter schwach dominant (50–60%)
  - 30 nicht ambige Wörter als Targets,
  - 60 Nicht-Wörter als Targets

= insgesamt 120 Wortpaare (Prime – Target)

→ alle Wörter ungefähr gleich lang (ein- bzw. zweisilbig), Frequenz vergleichbar (Mittelwert)
- Kritische Bedingungen:
  - 1) Wort vs. Nicht-Wort
  - 2) ambiges Wort vs. nicht ambiges Wort
  - 3) ambiges Wort stark dominant vs. ambiges Wort mittel dominant vs. ambiges Wort schwach dominant)

# Aufbau unseres Priming Experiments

- Prime-Wörter
  - Für die ambigen Wörter aus dem Pretest übernommen, Begünstigung der dominanten Lesart
  - Für nicht ambige Wörter und Nicht-Wörter selber erstellt
  - Vergleichbare Länge zum Target
  - Darf kein Wort sein, das bereits als Target vorkommt
- Gemessen wurde
  - Korrektheit der ja/nein-Entscheidung
  - Reaktionszeit
- Erstellung des Experiments mit der Software *Gorilla*

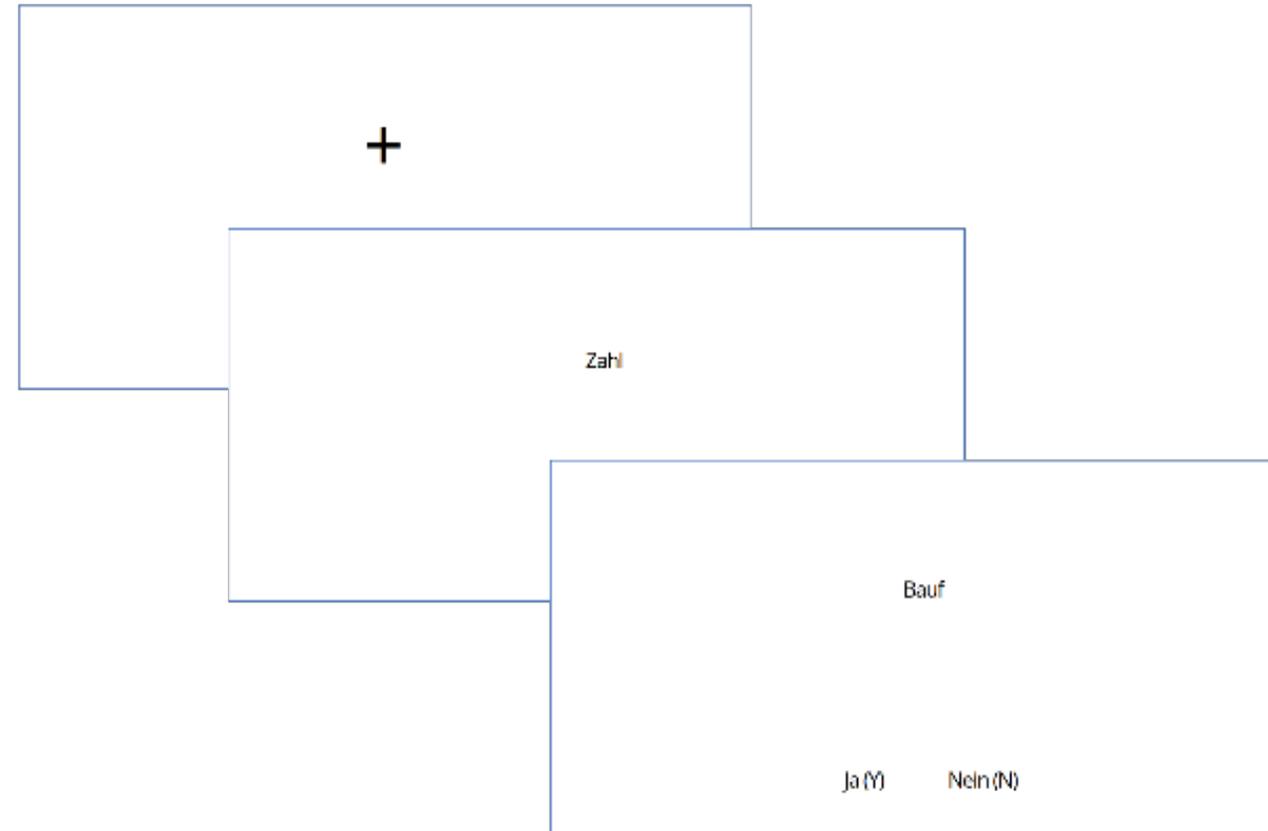
# Unser Priming Experiment bei *Gorilla*

- Begrüßung
- Demografische Daten
  - Geschlecht
  - Alter
  - Muttersprache
  - Wohnort oder Aufenthaltsort
  - Sprachkenntnisse
- Einverständniserklärung
- *Pretask* (Übungsdurchlauf)
- *Main task* (eigentliches Experiment)

# Unser Priming Experiment bei *Gorilla*

## Durchführung

- Am Anfang jedes Wortpaares wurde den Teilnehmenden ein Fixationskreuz für 1000 ms präsentiert
- Dann folgte das Prime-Wort (*Zahl*), welches für 300 ms sichtbar war.
- Danach folgte das Target-Wort (*Bauf*) und die Entscheidung, ob es sich um ein Wort oder Nicht-Wort handelt
- Nach anklicken von Y oder N wurde 200 ms später Feedback gegeben, ob die Antwort korrekt oder falsch war
- Nach jedem Wortpaar folgte ein leerer Bildschirm für 1000 ms und danach folgte das nächste Wortpaar, beginnend mit dem Fixationskreuz



# Verwendung von *Gorilla*

- Erstellung eines Accounts
  - Link: <https://gorilla.sc/>
- Erstellung eines Projekts
  - + *create a project*

Tipp: Wenn Du Dir einen Account über die Uni erstellst, ist dieser kostenlos! Frag einfach beim Team vom LiLEx-Labor nach!  
[https://www.uni-muenster.de/Germanistik/Linguistisches\\_Labor/](https://www.uni-muenster.de/Germanistik/Linguistisches_Labor/)

The screenshot shows the GORILLA web interface. At the top, there is a red header with the GORILLA logo on the left and the user name 'Lene Siefert' on the right. Below the header, a breadcrumb trail reads 'Home > Projects > Beispielexperiment'. The main content area is titled 'Beispielexperiment' and includes a 'Settings' icon and a red '+ Create' button. There are three sections: 'Experiments', 'Tasks & Questionnaires', and 'Open Materials'. Each section has a table with columns for 'Name' and 'Description'. To the right of each section is an 'Open Archive' button. The 'Complete Participants' section is also visible.

# Literatur und Links zur Hilfestellung

- Literatur zu Experimenten
  - Albert, Ruth & Marx, Nicole. 2010. Empirisches Arbeiten in Linguistik und Sprachlehrforschung: Anleitung zu quantitativen Studien von der Planungsphase bis zum Forschungsbericht. Tübingen: Gunter Narr.
  - Meindl, Claudia. 2011. Methodik für Linguisten: Eine Einführung in Statistik und Versuchsplanung. Tübingen: Narr.
- Gorilla
  - <https://support.gorilla.sc/support/getstarted>
  - <https://support.gorilla.sc/support/walkthrough/tutorial-videos>
  - <https://support.gorilla.sc/support/>

# 3. Auswertung der Ergebnisse



Programmiersprache



Entwicklungsumgebung und grafische  
Benutzeroberfläche für R

# Folgendes erklären wir Dir auf den nächsten Folien

- Was musst du vorbereiten, um Deine Daten mit R auszuwerten?
- Wie kannst du mit Deinen Daten umgehen?
- Wie erstellst Du Diagramme?
- Wie kannst Du einen Feinschliff vornehmen und das Layout Deiner Diagramme verändern?
- Wo kannst Du Dir weitere Hilfe holen (weiterführende Literatur/Links)?

# Vorbereitung

- 1) Herunterladen von R und RStudio: <https://www.r-project.org> → Download → CRAN → jeweiliges Institut auswählen, z. B. Germany, Münster → <https://cran.uni-muenster.de/>
- 2) Erstellen eines R-Projekts: R öffnen und Projekt direkt in gewünschtem Dateiordner abspeichern



Zur einfachen Benutzung sollten allen Dateien, die für R benötigt werden, im selben R-Projekt-Ordner abgespeichert werden: R arbeitet mit lokalen Bezügen, sodass der Ordnerpfad nicht auf jedem Computer neu definiert werden muss (stell sicher, dass Du dich immer im richtigen R-Projekt befindest!)

- 3) Benötigte Pakete installieren (z. B. ggplot2; tidyverse):

```
install.packages("Name des Pakets")
```

Achte hier auf das  
Setzen der  
Anführungszeichen!

Pakete in die Bibliothek von R laden (damit R auch auf sie zugreifen kann)

```
library(Name des Pakets)
```

Hier benötigst Du  
keine  
Anführungszeichen  
mehr

- 4) Einlesen der Rohdaten in R als csv-Datei (kann auch als anderes Dateiformat eingelesen werden, dafür müssen jedoch spezielle packages in R geladen werden)  
→ je nach verwendetem Programm für Experiment (z.B. *Gorilla*) werden die Ergebnisse und die soziodemografischen Daten der ProbandInnen in zwei verschiedenen Dateien erstellt: beide lädst Du in R (sie werden dann als zwei verschiedene Tabs im Programm dargestellt, siehe nächste Folie)

```
Name Datensatz <- read.csv("Name der csv-Datei",  
header=T)
```

Hiermit definierst und benennst Du den Datensatz, den Du als csv-Datei in R eingelesen hast (beliebig benennbar)

...so sieht das dann aus

Hiermit kannst Du markierte Befehle auch manuell ausführen

Im **Environment** werden Deine definierten Datensätze im Laufe des Skripts aufgelistet

Das ist das **Skript**: hier schreibst Du alles rein

Nachdem im Skript der Befehl zum Ausführen gegeben wurde (cmd/strg + enter) wird in der **Konsole** das Skript ausgeführt

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for installing packages and reading a CSV file. The 'Run' button is circled in red.
- Console:** Shows the output of the script execution, including variable assignments and data rows.
- Environment Pane:** Shows the current environment state, which is currently empty.
- Files Pane:** Shows the file explorer with a list of files including .RData, .Rhistory, and ambig.plot.pdf.

```
7
8 ##### die Pakete installieren, die man braucht #####
9 install.packages("ggplot2")
10 library(ggplot2)
11
12 ##### Daten einlesen #####
13 data_2 <- read.csv("Rohdaten_Experiment_2.csv", header=T)
14
15
16
17
17:1 # Daten einlesen
```

Console output:

```
~/Documents/Master/3. Semester/Psycho/Experiment ambige Wörter/
$ randomise_trials : logi NA NA NA NA NA NA ...
$ display          : chr "" "Begrüßung" "Tasks" "Tasks" ...
$ prime           : chr "" "" "Ort" "Ort" ...
$ target          : chr "" "" "Zeltel" "Zeltel" ...
$ Antwort         : chr "" "" "Nein" "Nein" ...
$ X               : logi NA NA NA NA NA NA ...
> ##### zu Beginn alles, was sich noch in der Ablage befindet, löschen #####
#####
```

| Name           | Size    |
|----------------|---------|
| ..             |         |
| .RData         | 1.4 MB  |
| .Rhistory      | 20.9 KB |
| ambig.plot.pdf | 4.5 KB  |

# Arbeiten mit den Daten

5) Die eingelesene csv-Datei enthält viele Daten und Spalten, die Du nicht immer für Deine Auswertung brauchst (z. B. genutzter Browser der ProbandInnen, Datum und Uhrzeit der Durchführung, Art des digitalen Endgeräts bei der Durchführung, etc.)

→ diese kannst Du zuerst entfernen, um eine bessere Übersicht zu erhalten

Mit diesen  
Befehlen löschst  
Du nicht benötigte  
Spalten

Damit Du weißt, wie die  
Spalten heißen, die du  
entfernen möchtest, klickst  
Du hier auf den Datensatz,  
den Du in Schritt 3  
eingelassen hast

The screenshot shows the RStudio interface. The main editor window contains R code for removing columns from a dataset. The code is as follows:

```
48 ##### Spalten entfernen, die wir nicht brauchen #####  
49 data$X = NULL  
50 data$X...Event.Index = NULL  
51 data$UTC.Timestamp = NULL  
52 data$UTC.Date = NULL  
53 data$Local.Timestamp = NULL  
54 data$Local.Timezone = NULL  
55 data$Experiment.ID = NULL  
56 data$Experiment.Version = NULL  
57 data$Tree.Node.Key = NULL  
58 data$Repeat.Key = NULL  
59 data$Schedule.ID = NULL  
60 data$Participant.Public.ID = NULL  
61 data$Participant.Starting.Group = NULL  
62 data$Participant.Completion.Code = NULL  
63 data$Participant.External.Session.ID = NULL  
64 data$Participant.OS = NULL  
65 data$Participant.Browser = NULL
```

The status bar at the bottom indicates: 31:1 # Liefert einen Überblick über die Daten R Script

The Environment pane on the right shows the following data overview:

| Environment | History                    | Connections | Tuto |
|-------------|----------------------------|-------------|------|
| R           | Global Environment         |             |      |
| Data        |                            |             |      |
| data_2      | 29947 obs. of 52 variab... |             |      |

The Files pane at the bottom right shows a file explorer view with the following files:

| Name      | Size    |
|-----------|---------|
| ..        |         |
| .RData    | 1.4 MB  |
| .Rhistory | 20.9 KB |



Mit diesem Befehl  
kannst Du Dir eine  
Übersicht Deiner  
Daten in der Konsole  
anzeigen lassen:  
*str(Name des  
Datensatzes, den  
Du dir anzeigen  
lassen willst)*

Beim Klicken auf data\_2 wird ein neuer Tab geöffnet, in dem die eingelesene Tabelle angezeigt wird

|    | Event.Index | UTC.Timestamp | UTC.Date            | Local.Timestamp | Local.Timezone | Local.Date       |
|----|-------------|---------------|---------------------|-----------------|----------------|------------------|
| 1  | 1           | 1.641465e+12  | 06/01/2022 10:31:02 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 2  | 2           | 1.641465e+12  | 06/01/2022 10:31:05 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 3  | 3           | 1.641465e+12  | 06/01/2022 10:31:06 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 4  | 4           | 1.641465e+12  | 06/01/2022 10:31:06 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 5  | 5           | 1.641465e+12  | 06/01/2022 10:31:08 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 6  | 6           | 1.641465e+12  | 06/01/2022 10:31:09 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 7  | 7           | 1.641465e+12  | 06/01/2022 10:31:10 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 8  | 8           | 1.641465e+12  | 06/01/2022 10:31:10 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 9  | 9           | 1.641465e+12  | 06/01/2022 10:31:12 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 10 | 10          | 1.641465e+12  | 06/01/2022 10:31:13 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 11 | 11          | 1.641465e+12  | 06/01/2022 10:31:14 | 1.641465e+12    |                | 1 06/01/2022 11: |
| 12 | 12          | 1.641465e+12  | 06/01/2022 10:31:14 | 1.641465e+12    |                | 1 06/01/2022 11: |

Showing 1 to 12 of 29,947 entries, 52 total columns



Mit diesem Befehl kannst Du zur besseren Übersicht Deine Spaltennamen ändern:

```
colnames(Name Datensatz) [which(names(Name Datensatz) == "bisheriger Spaltenname")] <- "neuer Spaltenname"
```

## 6) Falls Du Zeilen hast, die Du nicht brauchst...

| .ID    | Trial.Number | Zone.Type                | Reaction.Time | Response | Correct | randomise_blocks | pr |
|--------|--------------|--------------------------|---------------|----------|---------|------------------|----|
| 126383 | BEGIN TASK   |                          | NA            |          | 0       |                  |    |
| 126383 | 1            | continue_button          | 3236.00       |          | 0       |                  | NA |
| 126383 | 1            | fixation                 | 1000.02       |          | 0       | 84               | Or |
| 126383 | 1            | timelimit_screen         | 300.00        |          | 0       | 84               | Or |
| 126383 | 1            | response_keyboard_single | 1758.00       | Nein     | 1       | 84               | Or |
| 126383 | 1            | timelimit_screen         | 1000.02       |          | 0       | 84               | Or |
| 126383 | 2            | fixation                 | 1000.02       |          | 0       | 97               | Ke |
| 126383 | 2            | timelimit_screen         | 300.00        |          | 0       | 97               | Ke |
| 126383 | 2            | response_keyboard_single | 1189.00       | Nein     | 1       | 97               | Ke |
| 126383 | 2            | timelimit_screen         | 1000.02       |          | 0       | 97               | Ke |
| 126383 | 3            | fixation                 | 1000.02       |          | 0       | 115              | Ta |
| 126383 | 3            | timelimit_screen         | 300.02        |          | 0       | 115              | Ta |

Showing 1 to 12 of 28,979 entries, 11 total columns

z. B. die Zeile „BEGIN TASK“ enthält keine Informationen, daher kann sie eliminiert werden

...entfernst Du sie so:

```

95
96 - ##### Zeilen entfernen, die wir nicht brauchen #####
97 - ##### mit neuem Datensatz! #####
98 data1 = data[data$Trial.Number != 'BEGIN TASK',]
99 data1 = data1[data1$Trial.Number != 'END TASK',]
100

```

Achte immer darauf, dass Du den richtigen Datensatz angibst, den du verändern willst!



Erstelle Dir zur besseren Übersicht für jede Änderung, die Du im Datensatz vornimmst, einen neuen Datensatz → so kannst Du Deine Schritte besser nachvollziehen

## 7) Im nächsten Schritt kodierst Du deine Daten im Datensatz



Alle Daten im Datensatz sind sogenannte **Vektoren**

Mithilfe von Vektoren kann R rechnen, indem es in deren **Inhalt** schaut

Es gibt verschiedene Arten von Vektoren, die jedoch zunächst manuell bestimmt werden müssen

- a) **numeric**: numerische Vektoren sind eine Liste von Nummern, wie bspw. Reaktionszeiten, die alle hintereinander aufgelistet werden
- b) **character**: bestehen aus Buchstaben und bilden eine Liste von Text → enthalten Informationen wie Alter, Geschlecht, Dialekte, usw.
- c) **factor**: sind quasi „Zusammenfassungen“ der oben genannten Vektoren, sie können aus Zahlen oder Buchstaben bestehen und zeigen die verschiedenen Arten (level) an, die es gibt, bspw. gibt es zwei Arten (level) von Antworten, nämlich „1“ (für „korrekt“) und „0“ (für „falsch“) oder es gibt 60 Arten von Participant Private IDs, nämlich die Nummern aller 60 ProbandInnen

```
##### Variablen kodieren #####
```

```
Name Datensatz$Participant.Private.ID = as.factor(Name  
Datensatz$Participant.Private.ID)
```

```
Name Datensatz$Trial.Number = as.numeric(Name Datensatz$Trial.Number)
```

```
Name Datensatz$correct = as.character(Name Datensatz$correct)
```

Hier ein nützlicher Befehl, um mit Deinen Daten zu arbeiten

- Wenn Du neue Variablen in Deinen Datensatz hinzufügen willst (eine neue Spalte kreieren willst):

```
Name Datensatz[Name Datensatz$target == "Mond", "ambig"] <- "nichtambig"
```

Achte immer darauf, dass Du den richtigen Datensatz angibst, den du verändern willst!

Dieses Element bildet den Ausgangspunkt für die neue Variable: es wurde im Experiment als target gezeigt und wird durch die neue Variable und die Sub-Variable "nichtambig" genauer charakterisiert

Das ist deine neu kodierte Variable/ Kategorie, die dann als neue Spalte im Datensatz erscheint

Dieses Element soll zur neuen Kategorie „ambig“ gelistet werden und fungiert quasi als deren Sub-Kategorie: es definiert, dass das Wort „Mond“ nicht-ambig ist

Wenn das Wort bspw. „Ball“ wäre, würde hier an dieser Stelle „ambig“ stehen

So sieht die neue Variable/ Kategorie dann in Deinem Datensatz aus

|      | Response | Correct | randomise_blocks | prime | target | Antwort | lex  | ambig      |
|------|----------|---------|------------------|-------|--------|---------|------|------------|
| 53.0 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |
| 87.3 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |
| 02.2 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |
| 97.0 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |
| 57.0 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |
| 10.0 | Ja       | 1       | 1                | All   | Mond   | Ja      | wort | nichtambig |

Showing 1 to 6 of 57 entries, 13 total columns (filtered from 3,366 total entries)

## 8) Anzeigen deskriptiver statistischer Kennzahlen

- Wenn Du Dir einen Überblick über Deine Stichprobe verschaffen möchtest (in diesem Beispiel über alle Reaktionszeiten)

`mean(Name Datensatz$rt)` → Durchschnitt aller Werte

`min(Name Datensatz$rt)` → schnellste Zeit

`max(Name Datensatz$rt)` → langsamste Zeit

`sort(Name Datensatz$rt)` → Reaktionszeiten werden sortiert (von schnell zu langsam)

`summary(Name Datensatz$rt)` → hier wird nochmal alles auf einen Blick angezeigt: Median, Durchschnitt, max. und minimale Geschwindigkeit

rt steht für „reaction time“

Wenn Du nach dieser Übersicht Reaktionszeiten entfernen möchtest, geht das mit diesem Befehl:

```
Name Datensatz [Name  
Datensatz$rt < 2000, ]
```

Hier kannst du definieren, welche Reaktionszeiten entfernt werden sollen

9) So schaust Du dir spezifische Mittelwerte an:

```
tapply(Name Datensatz$rt, Name Datensatz$ambig, mean)
```

Hier bestimmst Du, von welchen Variablen aus  
Deinem Datensatz Du Dir die Durchschnittswerte  
anzeigen lassen willst, bspw. hier wurden die  
Mittelwerte der Reaktionszeiten (*rt*) der Variable  
*ambig* extrahiert

mean =  
Durchschnitt

# Erstellen von Diagrammen

10) Wie in Schritt 8 erstellst Du zunächst eine Zusammenfassung der Daten, die in Deinem Diagramm abgebildet werden sollen

```
lex.sum <- summarySEwithin(Name Datensatz,  
measurevar="rt",  
withinvars="lex",  
idvar="vp")
```

Name der Kategorie/  
Variable, die Du in  
einem Diagramm  
abbilden willst

Informationen der  
Kategorien, die das  
Diagramm enthalten soll

11) Deine Funktion für ein Diagramm könnte so beispielhaft aussehen → abgebildet sind die Grundlagen (in den weiteren Schritten erfährst Du, wie Du diese modifizierst)

Um Diagramme zu erstellen → lade z. B. package `ggplot2` in deine library (siehe Schritt 3)

Das Diagramm nutzt die Informationen Deiner Zusammenfassung aus Schritt 10

Hier definierst Du, welche Informationen auf der x- und y-Achse abgebildet werden sollen

Informationen über Art, Position, Breite und Farbe der Balken

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt))+  
  geom_bar(width=.5, position=position_dodge(), stat="identity",  
  color = "black", fill = "dodgerblue4")+  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.15,  
  position=position_dodge(.5), colour="navy blue")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Wort vs. Nicht-Wort")+  
  labs(x = "", y = "Reaktionszeiten in ms")  
lex.plot
```

Eingabe des Diagramm-Titels

Skalierung der Skala auf der y-Achse

Informationen über Art, Position, Breite und Farbe des Konfidenzintervalls

Beschriftung der Achsen

In diesem letzten Schritt führst Du die Funktion aus und das Diagramm erscheint

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity",  
color = "black", fill = "dodgerblue4") +  
geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.15,  
position=position_dodge(.5), colour="navy blue") +  
  coord_cartesian(ylim=c(500,1000)) +  
  ggtitle("Wort vs. Nicht-Wort") +  
  labs(x = "", y = "Reaktionszeiten in ms")
```

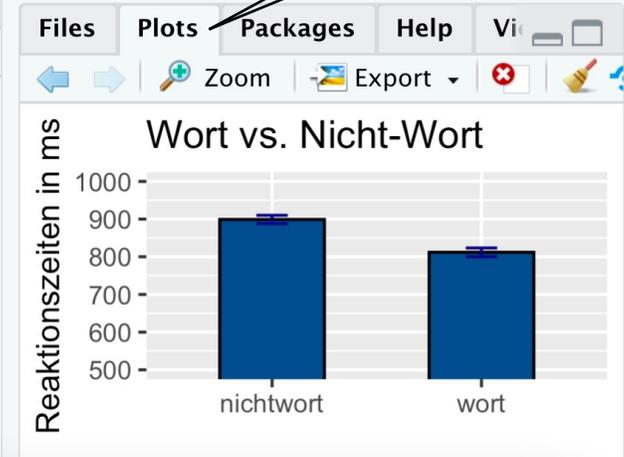


**Wichtig:** Zwischen den einzelnen Funktionsbausteinen muss immer ein + (rot markiert) geschrieben werden, damit es eine vollständige Funktion ergibt, die ausgeführt werden kann

```
287
288 lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt))+
289   geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black", f
290   geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.15, position=position_dodge(.5)
291   coord_cartesian(ylim=c(500,1000))+
292   ggtitle("Wort vs. Nicht-Wort") +
293   labs(x = "", y = "Reaktionszeiten in ms")
294 lex.plot
295 pdf(file="lex.plot.pdf", width=8, height=8)
296 lex.plot
297 dev.off()
295:25 # 1. Plot: lex
```

```
~/Documents/Master/3. Semester/Psycho/Experiment ambige Wörter/
+ coord_cartesian(ylim=c(500,1000))+
+ ggtitle("Wort vs. Nicht-Wort") +
+ labs(x = "", y = "Reaktionszeiten in ms")
> lex.plot
> pdf(file="lex.plot.pdf", width=8, height=8)
> lex.plot
> dev.off()
RStudioGD
  2
>
```

| Environment            |                       | History | Connections |
|------------------------|-----------------------|---------|-------------|
| R   Global Environment |                       |         |             |
| data                   | 28979 obs. of 11 v... |         |             |
| data_2                 | 29947 obs. of 52 v... |         |             |
| data_cor...            | 6834 obs. of 12 va... |         |             |
| data1                  | 28860 obs. of 11 v... |         |             |
| data2                  | 7200 obs. of 12 va... |         |             |
| data3                  | 3366 obs. of 13 va... |         |             |
| data4                  | 1697 obs. of 14 va... |         |             |
| datatrim               | 6625 obs. of 12 va... |         |             |
| lex.plot               | List of 9             |         |             |



Hier erscheinen dann Deine ausgeführten Diagramme

# Feinschliff: Veränderungen des Layouts

## 12) Ändern des Titels

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black",  
fill = "dodgerblue4") +  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2, position=position_dodge(.5),  
colour="grey")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Wort vs. Nichtwort") + → Titel hier beliebig ändern  
  labs(x = "", y = "Reaktionszeiten in ms")  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

### 13) Ändern der Beschriftung der x-Achse (nachträglich und manuell)

→ Im Ausgangsskript wird die x-Achse so beschriftet, wie sie im Laufe des Skripts definiert wurde → dies kann nach Belieben jedoch noch nachträglich manuell geändert werden

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color =  
"black", fill = "dodgerblue4") +  
  scale_x_discrete(labels=c("Nicht-Wort", "Wort")) +  
  xlab("Names") + → Hinzufügen dieses Funktionsbausteins und beliebige  
Änderung des grün markierten  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2,  
position=position_dodge(.5), colour="grey") +  
  coord_cartesian(ylim=c(500,1000)) +  
  ggtitle("Wort vs. Nichtwort") +  
  labs(x = "", y = "Reaktionszeiten in ms")  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

Im blau  
markierten  
Bereich wird  
die y-Achse  
beschriftet

## 14) Ändern der Balkenfarbe

### a) einfarbig

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color =  
  "black", fill = "dodgerblue4") + → beliebige Änderung der Farbe (Farben  
siehe z.B. http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf )  
  scale_x_discrete(labels=c("Nicht-Wort", "Wort"))+  
  xlab("Names")+  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2,  
  position=position_dodge(.5), colour="grey")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Wort vs. Nichtwort") +  
  labs(x = "", y = "Reaktionszeiten in ms")  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

grün markiert:  
Farbe der  
Balkenumrandung  
(kann auch  
gelöscht werden,  
dann keine  
Umrandung)

Hinzufügen: fill = der Wert, der für x eingetragen ist → hier demnach dominanz

## b) Farbverlauf einer Farbe (Nutzen der Brewer-Farbpalette)

```
dominanz.plot = ggplot(dominanz.sum, mapping=aes(x=dominanz, y=rt, fill=dominanz)) +  
  scale_x_discrete(limits = positions)+  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black", fill =  
  "dodgerblue4") +  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2, position=position_dodge(.5), colour="grey")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Dominanz der Bedeutungen ambiger Wörter") +  
  scale_fill_brewer(palette = "Greens")+  
  guides(fill = "none")+  
  labs(x = "", y = "Reaktionszeiten in ms")  
dominanz.plot  
pdf(file="dominanz.plot.pdf", width=8, height=8)  
dominanz.plot  
dev.off()
```

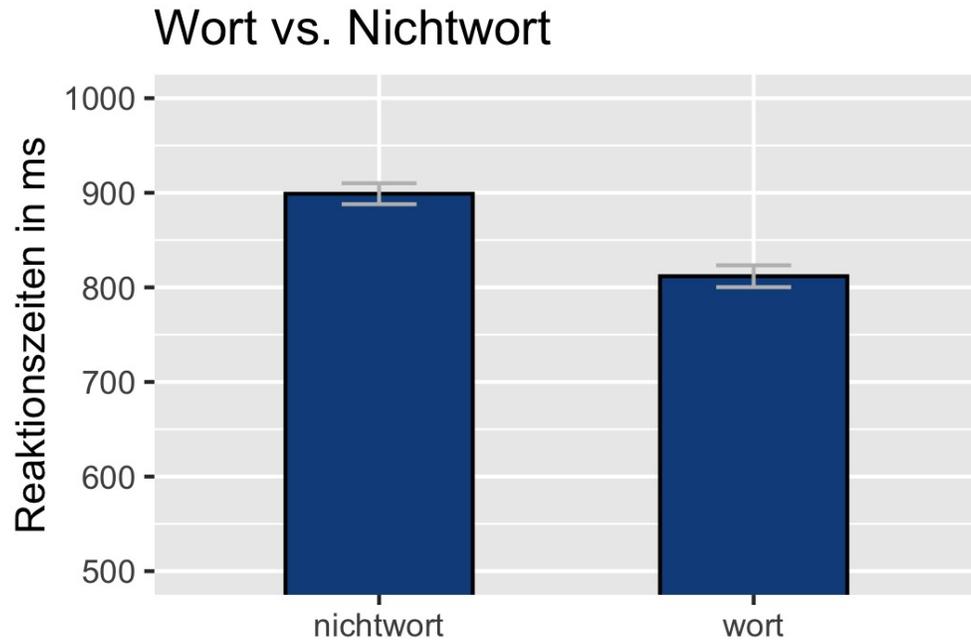
Löschen

Hinzufügen dieses Funktionsbausteins  
**!!Achtung!!** vorher  
Paket installieren  
(wird für Farbpalette  
benötigt):

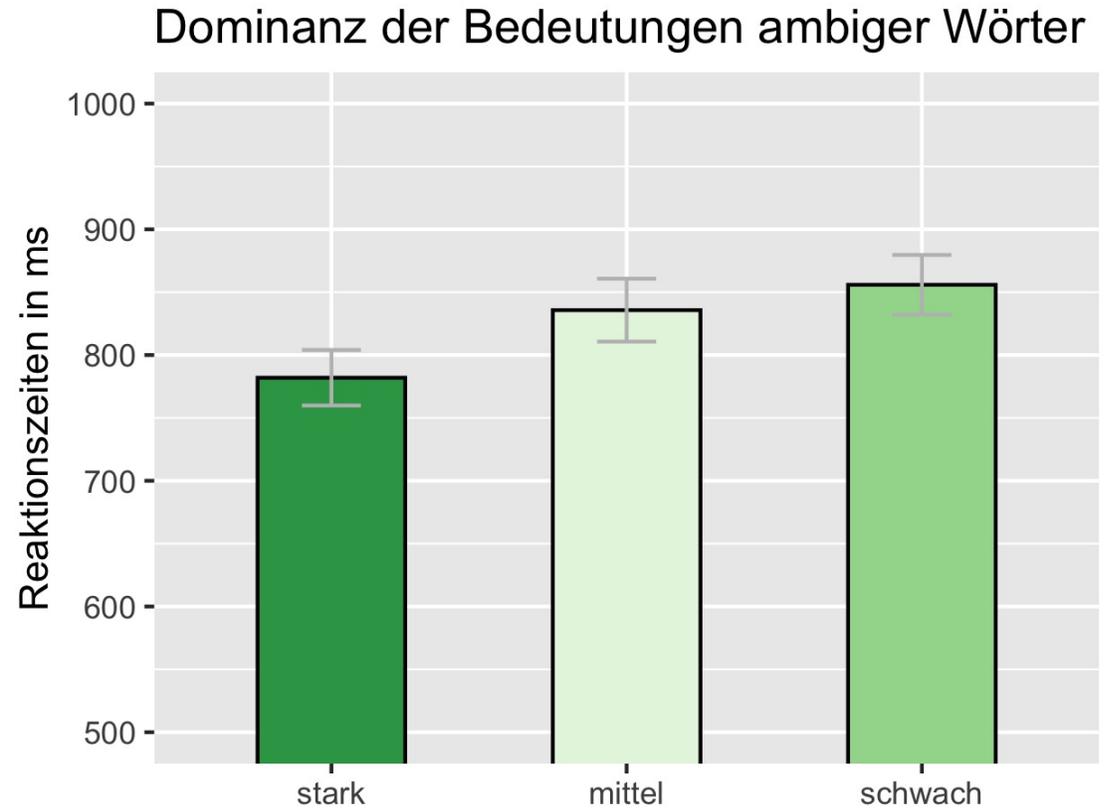
```
install.packages  
("RColorBrewer")
```

Brewer-Farbpaletten  
siehe z. B.  
<https://www.r-graph-gallery.com/38-rcolorbrewers-palettes.html>

## Beispiel für a)



## Beispiel für b)



### c) verschiedene Farben (manuelles Aussuchen der Farben)

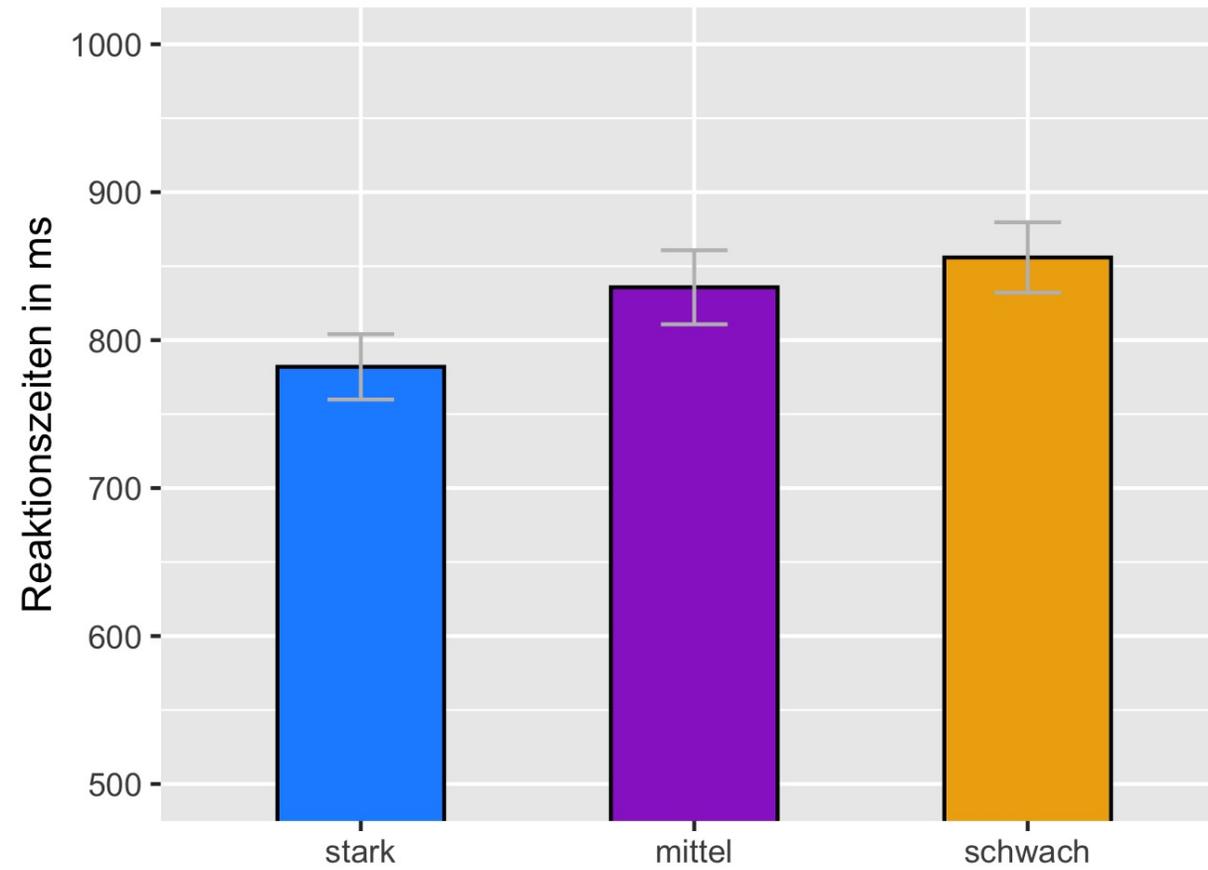
```
dominanz.plot = ggplot(dominanz.sum, mapping=aes(x=dominanz, y=rt, fill=dominanz)) +  
  scale_x_discrete(limits = positions)+  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black", fill=  
  "dodgerblue4") +  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2, position=position_dodge(.5),  
  colour="grey")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Dominanz der Bedeutungen ambiger Wörter") +  
  scale_fill_manual(values = c("darkorchid", "darkgoldenrod2", "dodgerblue1"))+  
  guides(fill = "none")+  
  labs(x = "", y = "Reaktionszeiten in ms")  
dominanz.plot  
pdf(file="dominanz.plot.pdf", width=8, height=8)  
dominanz.plot  
dev.off()
```

Hinzufügen dieses Funktionsbausteins

Farben siehe z. B.  
<http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

## Beispiel für c)

### Dominanz der Bedeutungen ambiger Wörter

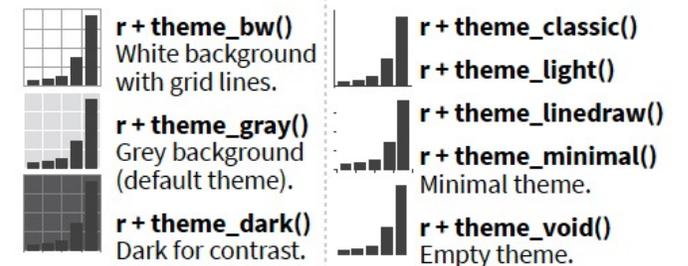


## 15) Ändern des Themes (Plot-Hintergrund)

→ Hintergrund der Beispiele bisher: Standardeinstellung von R

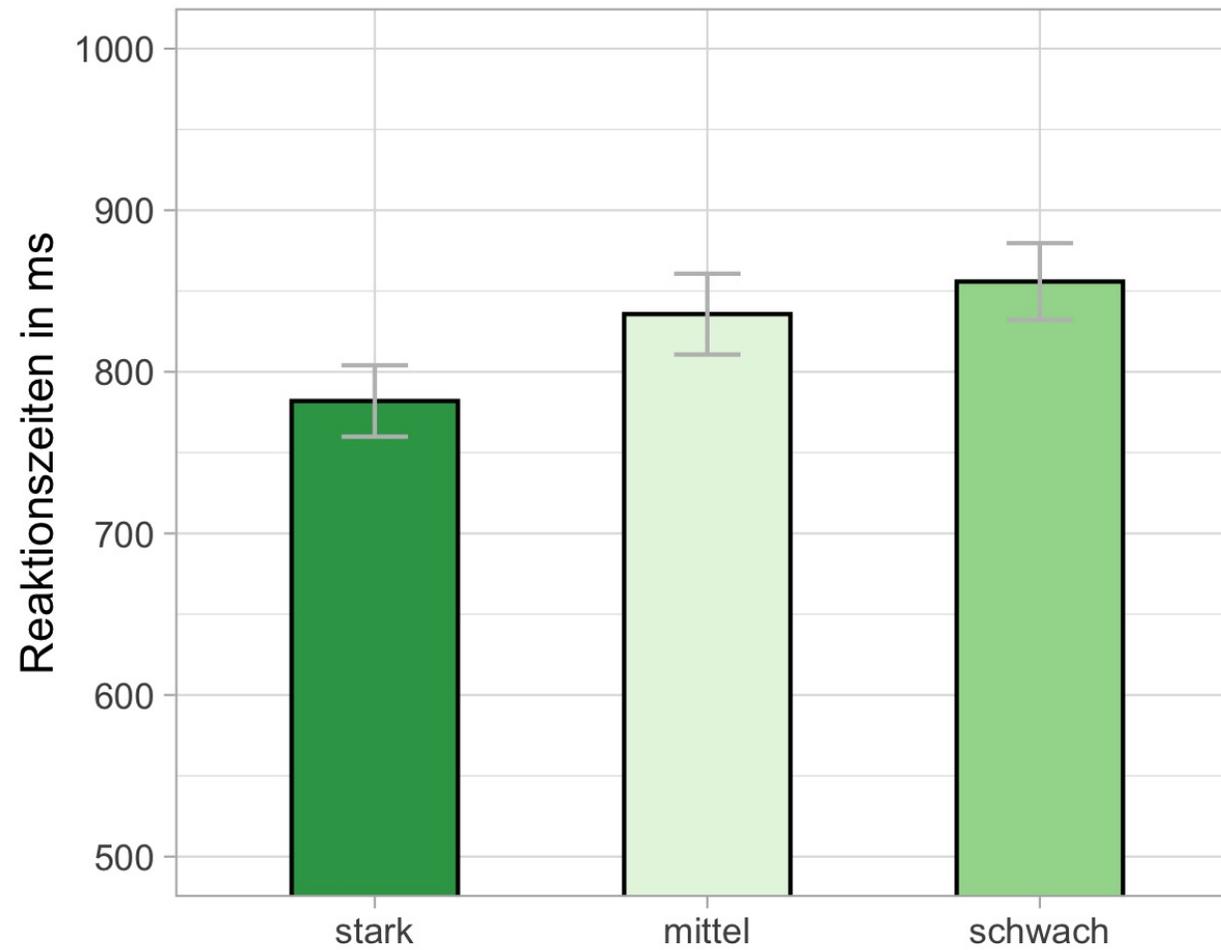
```
dominanz.plot = ggplot(dominanz.sum, mapping=aes(x=dominanz, y=rt, fill=dominanz)) +  
  scale_x_discrete(limits = positions)+  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black") +  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.2, position=position_dodge(.5),  
  colour="grey")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Dominanz der Bedeutungen ambiger Wörter") +  
  scale_fill_brewer(palette = "Greens")+  
  guides(fill = "none")+  
  labs(x = "", y = "Reaktionszeiten in ms")+  
  theme_light()  
dominanz.plot  
pdf(file="dominanz.plot.pdf", width=8, height=8)  
dominanz.plot  
dev.off()
```

Hinzufügen bspw. dieses Funktionsbausteins,  
weitere:



## Beispiel für 4)

### Dominanz der Bedeutungen ambiger Wörter

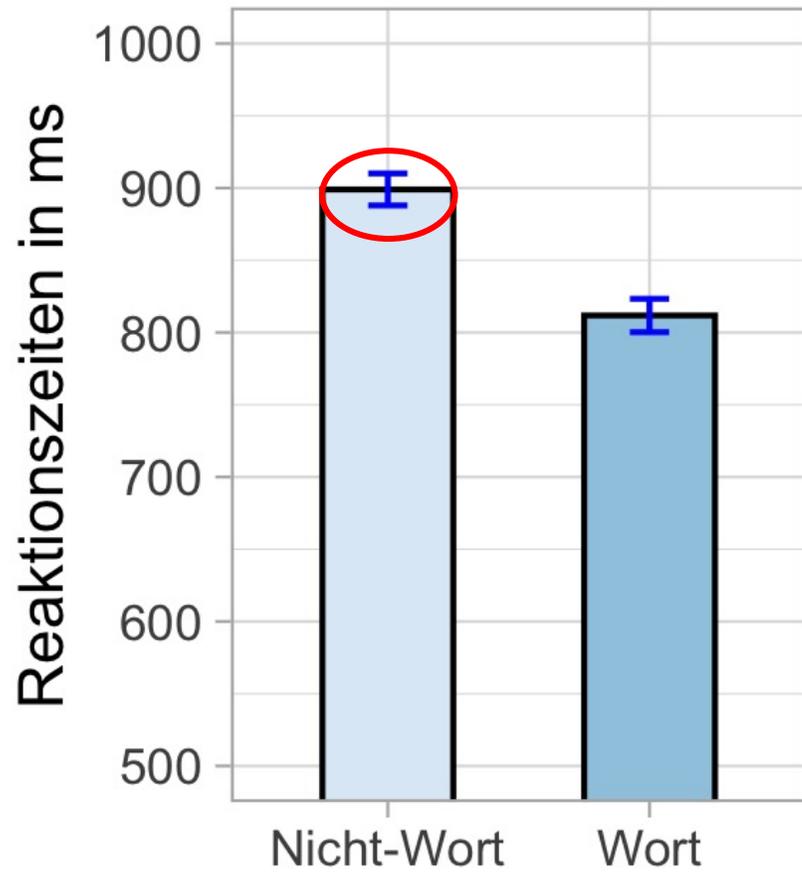


## 16) Ändern des Konfidenzintervalls

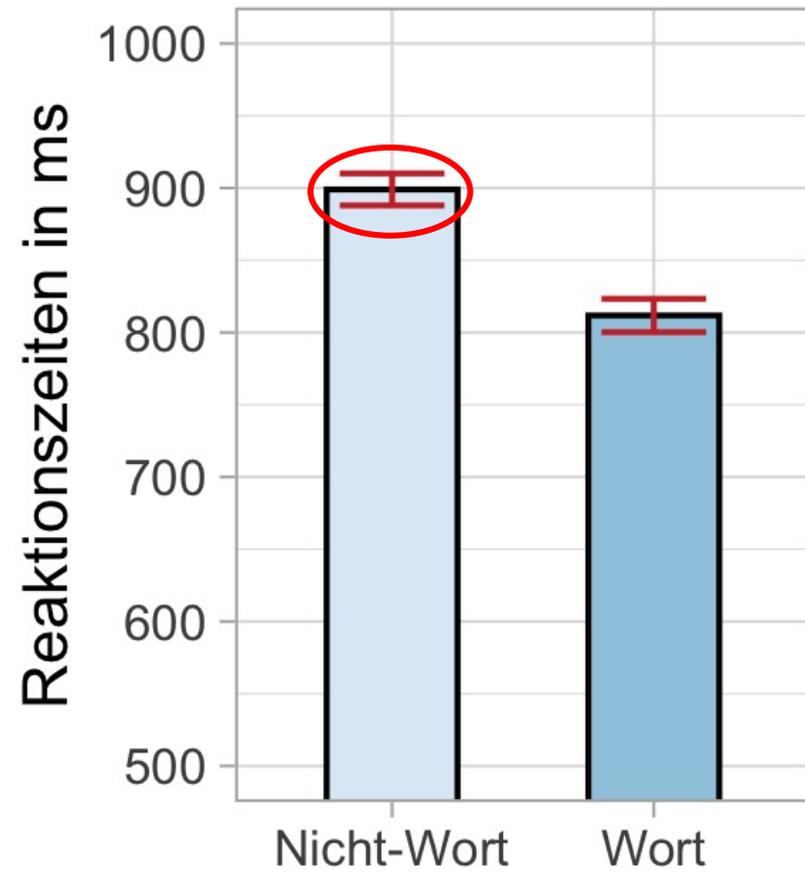
```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt, fill = lex)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black") +  
  scale_x_discrete(labels=c("Nicht-Wort", "Wort"))+  
  xlab("Names")+  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.15, position=position_dodge(.5),  
  colour="blue")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Wort vs. Nichtwort") +  
  labs(x = "", y = "Reaktionszeiten in ms")+  
  scale_fill_brewer(palette = "Blues")+  
  guides(fill = "none")+  
  theme_light()  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

The diagram consists of two callout boxes with black outlines and white backgrounds. The first callout box, labeled "Verändern der Farbe", has two lines pointing to the "colour="blue"" parameter in the `geom_errorbar` function. The second callout box, labeled "Verändern der Breite", has one line pointing to the "width=.15" parameter in the same function.

### Wort vs. Nichtwort



### Wort vs. Nichtwort



## 17) Ändern der Breite der Balken

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt, fill = lex)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black") +  
  scale_x_discrete(labels=c("Nicht-Wort", "Wort"))+  
  xlab("Names")+  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.4, position=position_dodge(.5),  
colour="brown3")+  
  coord_cartesian(ylim=c(500,1000))+  
  ggtitle("Wort vs. Nichtwort") +  
  labs(x = "", y = "Reaktionszeiten in ms")+  
  scale_fill_brewer(palette = "Blues")+  
  guides(fill = "none")+  
  theme_light()  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

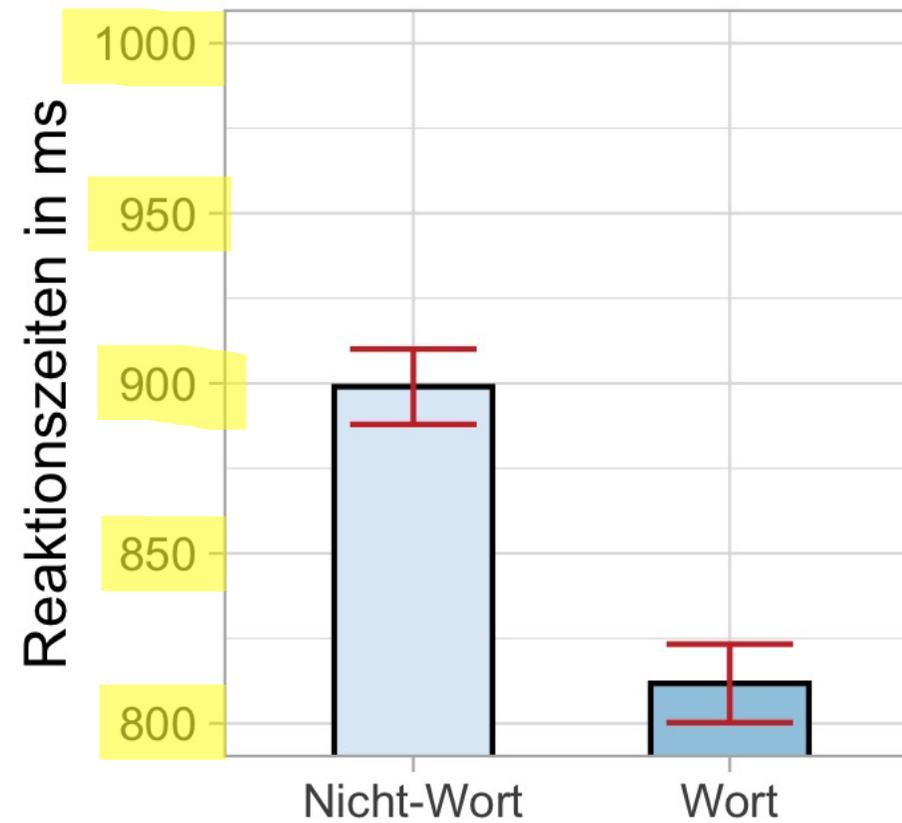
Je größer die Zahl,  
desto dicker; je  
kleiner, desto  
dünner

## 18) Ändern der Größe der Skala

```
lex.plot = ggplot(lex.sum, mapping=aes(x=lex, y=rt, fill = lex)) +  
  geom_bar(width=.5, position=position_dodge(), stat="identity", color = "black") +  
  scale_x_discrete(labels=c("Nicht-Wort", "Wort"))+  
  xlab("Names")+  
  geom_errorbar(aes(ymin=rt-ci, ymax=rt+ci), width=.4, position=position_dodge(.5),  
colour="brown3")+  
  coord_cartesian(ylim=c(800,1000))+  
  ggtitle("Wort vs. Nichtwort") +  
  labs(x = "", y = "Reaktionszeiten in ms")+  
  scale_fill_brewer(palette = "Blues")+  
  guides(fill = "none")+  
  theme_light()  
lex.plot  
pdf(file="lex.plot.pdf", width=8, height=8)  
lex.plot  
dev.off()
```

## Veränderte Skala

### Wort vs. Nichtwort



# Weiterführende Literatur/Links

- Einführungsliteratur: Winter, Bodo (2020): *Statistics for Linguists: An Introduction Using R*, New York: Routledge.
- RStudio Cheatsheets: auf komprimierten Cheatsheets sind die Funktionen der am frequentesten genutzten Packages übersichtlich aufgelistet (es werden regelmäßig neue Cheatsheets veröffentlicht)
  - Link: <https://www.rstudio.com/resources/cheatsheets/>
- Google is your friend – ask the R community! – das hilft wirklich immer, denn es gibt für ein Problem immer mehrere Lösungsansätze, die jemand schon einmal bereitgestellt hat
  - Wichtig für die Google-Suchanfrage: am besten auf Englisch stellen und immer „R“ mit angeben, da es noch eine Reihe anderer Programmiersprachen gibt!