Statistische Grundlagen

Eine Einführung mit Beispielen aus der Sprachforschung

Jan Vanhove

Universität Freiburg/Fribourg, Schweiz
Departement für Mehrsprachigkeitsforschung und Fremdsprachendidaktik

jan.vanhove@unifr.ch
https://janhove.github.io

komplett überarbeitete Version: Sommer 2018
letzte Änderung: September 2019

Vorwort

Das vorliegende Skript ist eine Neubearbeitung eines gleichnamigen Skriptes, das ich 2015 auf meiner Website (`https://janhove.github.io`) veröffentlicht habe. Es teilt das Ziel seines Vorgängers: Forschenden in Bereichen wie den einzelsprachigen Philologien (Anglistik, Germanistik, Romanistik, usw.), der theoretischen oder angewandten Linguistik und der Mehrsprachigkeitsforschung statistische Grundkenntnisse zu vermitteln, die ihnen sowohl bei der Lektüre quantitativer Forschungsberichte als auch bei der Gestaltung und Auswertung eigener Studien nützlich sein werden. Dabei wird davon ausgegangen, dass die Lesenden bisher nur geringfügige Erfahrung mit quantitativer Datenanalyse hatten.

Mit dieser Neubearbeitung versuche ich Folgendes zu bewirken:

- Die Erfahrung hat gezeigt, dass Studierende oft Schwierigkeiten haben, ihre eigenen Datensätze so zu gestalten, dass diese einfach in einem Computerprogramm ausgewertet werden können. Kapitel 2 ist daher diesen Schritten gewidmet.
- Der Fokus im Skript liegt auf dem *Schätzen* relevanter Informationen anhand von Stichproben (z.B. das Mittel einer Population oder die Form des Zusammenhangs von zwei Variablen) und dem Quantifizieren der *Unsicherheit* dieser Schätzung. Teil II (Kapitel 3 bis 6) arbeitet auf dieses Ziel hin.
- Konzepte wie Stichprobenfehler und Konfidenzintervalle werden hier hauptsächlich anhand von Simulationen und verwandten Methoden (*bootstrapping*) illustriert. Hiervon verspreche ich mir, dass sie diese Konzepte besser veranschaulichen und die ihnen zu Grunde liegenden Annahmen klarer darlegen als eine rein mathematische Erklärung.
- Im dritten Teil wird das allgemeine lineare Modell vorgestellt als ein Rahmen, in dem man Informationen schätzen kann und die Unsicherheit dieser Schätzung quantifizieren kann.
- Auch wenn ich den gelegentlichen Nutzen von Signifikanztests nicht abstreiten will, halte ich diese vermehrt für überverwendet. Um zu vermeiden, dass Lesende den Eindruck erhalten, dass p-Werte das A und O einer statistischen Analyse sein sollten, werden diese daher erst im vierten Teil, unter Begleitung vieler Wenn und Aber, besprochen.

- Der Kurs, für den dieses Skript als Grundlage dient, dauert nur 14 Wochen. Dies reicht nicht, um das ganze Spektrum von der Berechnung eines Medians hin bis zur Verwendung komplexer Mehrebenenmodelle zu besprechen. Im fünften Teil erhalten Lesende daher Empfehlungen, wo sie sich über solche Verfahren selber schlau machen können, falls sie diese für ihre eigene Arbeit brauchen. Einige meiner gelegentliche Blogeinträge unter `https://janhove.github.io/` können auch als Grundlage zum erweiterten Selbststudium dienen.

Danke an Isabelle Udry, die mich auf sprachliche Fehler und inhaltliche Unklarheiten hingewiesen habe. Diese habe ich dann prompt durch andere ersetzt. Auch für jegliche weitere Hinweise wäre ich dankbar.

Jan Vanhove
Freiburg, Schweiz, Juli 2018

Inhaltsverzeichnis

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding'
sampler used
```

Teil I

Software und Datenaufbereitung

Kapitel 1

Software

1.1 R installieren

Um alle Schritte in diesem Skript mitverfolgen zu können, brauchen Sie die Gratis-Software R.

Laden Sie dieses Programm unter `http://www.r-project.org` herunter und installieren Sie es. Wählen Sie als Sprache bitte Englisch und kreuzen Sie im Fenster SSelect ComponentsMessage translations"*nicht* an. Wenn Sie dieses Kästchen ankreuzen, ist es nämlich möglich, dass die Fehlermeldungen, die Sie unvermeidlich erhalten werden, auf Deutsch oder Französisch angezeigt werden. Um im Internet nach Lösungen zu suchen, ist es aber besser, wenn die Fehlermeldungen auf Englisch angezeigt werden.

1.2 RStudio installieren

Es lohnt sich, zusätzlich zu R auch RStudio herunterzuladen. Dies ist ein benutzerfreundlicheres Interface, in dem Sie mit R arbeiten können. RStudio ist ebenfalls kostenlos. Laden Sie es unter `http://rstudio.com` herunter und installieren Sie es.

Öffnen Sie RStudio. Ihr Bildschirm soll jetzt so aussehen wie in Abbildung 1.1. Wenn Sie statt vier Quadranten nur drei sehen, klicken Sie auf `File, New File, R Script`.

Eine kurze Beschreibung:

- Links unten sehen Sie die R-Konsole. Vermeiden Sie möglichst, hier direkt Befehle einzutragen.
- Links oben sehen Sie einen Texteditor, wo Sie Befehle und Kommentare eintippen können. Anstatt Ihre Befehle in die Konsole einzutragen, sollten Sie diese hier eintragen. Das macht es einfacher, Tippfehler aufzudecken. Ausserdem können Sie diese Skripts speichern, sodass Sie Ihre Analyse nachher reproduzieren können.

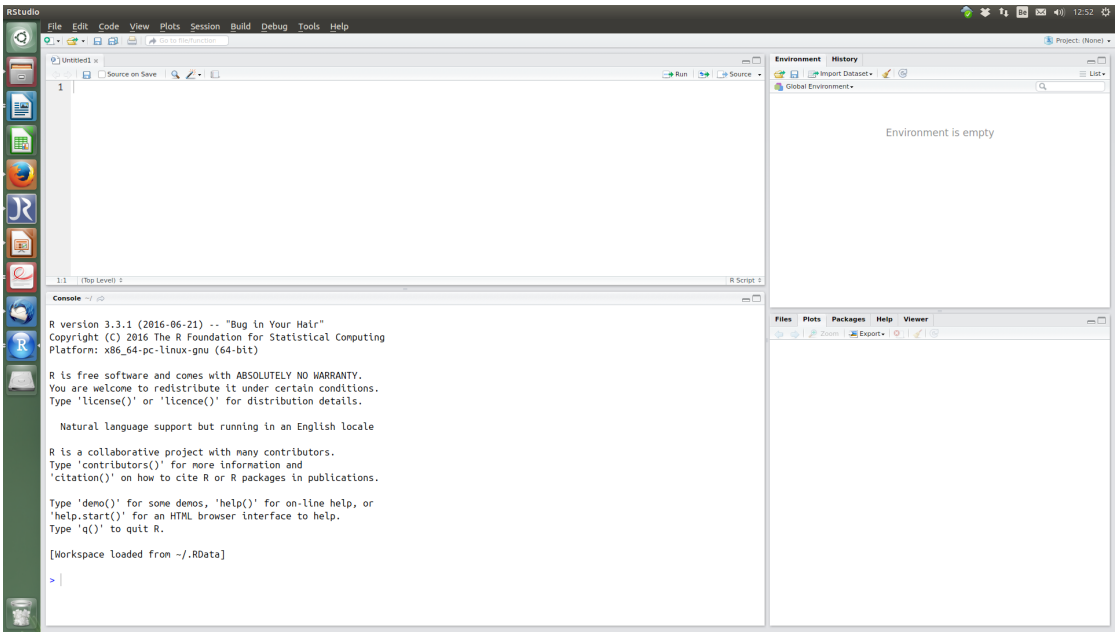


Abbildung 1.1: RStudio mit links unten der R-Konsole, links oben einem Texteditor, rechts oben dem Verzeichnis über die Arbeitsumgebung und rechts unten Hilfeseiten und Grafiken.

- Rechts oben werden alle Objekte in der R-Arbeitsumgebung aufgelistet. Da Sie noch keine Daten eingelesen haben, ist diese Umgebung momentan leer.
- Wenn Sie eine Grafik zeichnen, sehen Sie diese rechts unten. Wenn Sie eine Hilfeseite abfragen, erscheint diese ebenfalls hier.

Tippen Sie Befehle nicht direkt in die Konsole, sondern zuerst in den Texteditor ein!

1.3 Erweiterungspakete installieren

Es gibt für R jede Menge Erweiterungspakete, mit denen man z.B. informative Grafiken gestalten kann oder spezialisierte statistische Modelle rechnen kann. Mit dem folgenden Befehl installieren Sie das `tidyverse`-Bündel: eine Sammlung unterschiedlicher Pakete, die alle auf der gleichen Philosophie basieren und die das Arbeiten mit Datensätzen wesentlich erleichtern. Tragen Sie diesen Befehl in RStudio ins Fenster links *oben* ein. Selektieren Sie dann die Zeilen, klicken Sie auf `Code, Run Selected Line(s)` oder drücken Sie CTRL + ENTER (Mac: CMD + ENTER). Der Befehl wird nun der Konsole (links unten) weitergeleitet.

```
# Installiere Erweitung "tidyverse".
# Zeilen, denen ein '#' vorangestellt sind, gelten übrigens
# als Kommentare. Diese werden von R ignoriert - aber sie
```

```
# sind extrem praktisch, um Befehle zu dokumentieren, sodass
# man selber ein paar Wochen später noch nachvollziehen kann,
# was man genau gemacht hat!
install.packages("tidyverse")
```

Für die Erweitungspakete, die im `tidyverse`-Bündel zusammengepackt wurden, finden Sie unter `https://www.tidyverse.org/` Anleitungen und weitere Informationen.

Your closest collaborator is you six months ago, but you don't reply to email. Kommentieren Sie Ihren Code ausführlich, gerne auch in Vollsätzen! Beschränken Sie sich dabei nicht darauf, einfach die Kommentare aus diesem Skript zu übernehmen, sondern fügen Sie auch Ihre eigenen Kommentare und Gedächtnishilfen hinzu.

1.4 Den Arbeitsordner organisieren

Es ist sinnvoll, wenn die Skripts und Datensätze, die Sie für ein Forschungsprojekt brauchen, alle im gleichen Ordner ('Arbeitsordner') stehen. Diesen Ordner können Sie von Hand einstellen (Möglichkeit 1) oder Sie können in RStudio einen kreieren (Möglichkeit 2); ich empfehle letztere Möglichkeit.

1.4.1 Möglichkeit 1: Den Arbeitsordner von Hand einstellen

Kreieren Sie irgendwo auf Ihrem Computer einen Ordner, wo Sie die Skripts und Datensätze, die wir in diesem Kurs verwenden, speichern werden. Machen Sie in diesem Ordner einen zusätzlichen Ordner `Data`; hierin werden wir Datensätze speichern.

Öffnen Sie dann wieder RStudio. Klicken Sie auf `Session, Set Working Directory, Choose Directory` und wählen Sie den Ordner, den Sie gerade kreiert haben, aus. In der Konsole, also links unten, sehen Sie jetzt eine Zeile die etwa so aussieht:

```
> setwd("~Documents/StatistikKurs")
```

Kopieren Sie diese Zeile *ohne* das `>` und kleben Sie sie ins Skript (links oben) ein. Mit diesem Befehl stellen Sie den Arbeitsordner ein. Speichern Sie das Skript und lassen Sie diesen Befehl immer zuerst laufen, wenn Sie die Befehle in diesem Skript laufen lassen.[1]

[1] Die Verwendung der `setwd()`-Funktion ist erstaunlich umstritten, aber solange Sie sämtliche Dateien, die Sie für ein Projekt brauchen, im gleichen Ordner gespeichert haben, geht sie aus meiner Sicht in Ordnung. Es ist aber besser, Ihre Projekte mit Projektordnern zu verwalten (siehe Möglichkeit 2).

Den Arbeitsordner jedes Mal einstellen! Jedes Mal, wenn Sie R/RStudio neu öffnen und mit Datensätzen arbeiten möchten, müssen Sie den Arbeitsordner neu einstellen.

Alternativ können Sie auch direkt im Ordner, den Sie als Arbeitsordner verwenden möchten, eine neue leere Datei mit der Endung `.R` kreieren und diese Datei mit der rechten Maustaste in RStudio öffnen. Hierdurch wird der Arbeitsordner automatisch eingestellt.

1.4.2 Möglichkeit 2: Einen Projektordner kreieren (= besser)

Klicken Sie in RStudio auf `File > New Project... > New directory > Empty project` und geben Sie dem Projekt einen Namen (z.B. Statistikkurs).[2] Es wird jetzt ein Ordner auf Ihrer Festplatte kreiert. Wenn Sie zu diesem Ordner gehen, sehen Sie in ihm eine Datei mit der Endung `.Rproj`. Wenn Sie an diesem Projekt weiterarbeiten möchten, können Sie diese Datei öffnen; der Arbeitsordner wird hiermit automatisch eingestellt.

Damit Sie die Befehle aus diesem Skript direkt übernehmen können, sollten Sie in diesem Ordner noch einen Subordner namens `Data` kreieren.

1.5 Software zitieren

R und R-Pakete sind gratis und werden mehrheitlich von anderen Forschenden entwickelt. Wenn Sie bei Ihrer Arbeit sehr von R oder einem Erweiterungspaket profitiert haben, ziehen Sie es dann in Erwägung diesen Freiwilligen mit einer Referenz zu danken.

Eine Referenz für R erhalten Sie, wenn Sie in der Konsole den folgenden Befehl eintippen:

```
citation()
```

Wenn Sie ein bestimmtes Erweiterungspaket zitieren möchten, stellen Sie den Namen des Pakets zwischen Klammern und Anführungszeichen, etwa so:

```
citation("tidyverse")
```

Falls Sie genaue Softwareversionen brauchen, können Sie den folgenden Befehl verwenden:

```
sessionInfo()
```

[2] Die Optionen `Create a git repository` and `Use packrat with this project` brauchen Sie für dieses Projekt nicht einzuschalten; was diese Optionen bewirken, wird in diesem Skript folglich nicht behandelt.

Kapitel 2

Daten organisieren, einlesen und abfragen

Die erste Hürde, die es bei einer quantitativen Analyse zu überwinden gilt, ist, die Daten so zu organisieren, dass diese überhaupt analysierbar sind. Hat man dies geschafft, muss man die Daten noch in das Computerprogramm, mit dem man sie analysieren wird (hier: R), einlesen. Dieses Kapitel widmet sich diesen ersten Schritten.

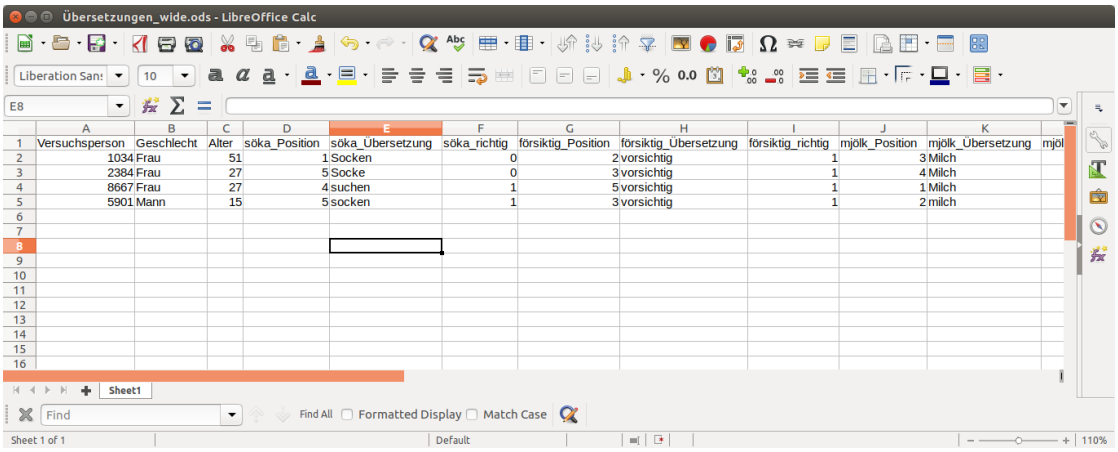
2.1 Daten organisieren

Stellen Sie sich die folgende Datenerhebung vor. Sie möchten untersuchen, wie sich die Fähigkeit, die Bedeutung von Wörtern in einer nicht beherrschten Sprache auf der Basis ihrer Ähnlichkeit zu Wörtern in beherrschten Sprachen zu erschliessen, im Laufe des Lebens verändert. Dazu legen Sie einer Reihe von deutschsprachigen Versuchspersonen unterschiedlichen Alters eine Anzahl geschriebener schwedischer Wörter vor und bitten Sie sie, diese ins Deutsche zu übersetzen. Der Übersichtlichkeit halber werden hier die Übersetzungen von vier Versuchspersonen für fünf Wörter gezeigt, und zwar für jede Versuchsperson in der Reihenfolge, in der die Wörter übersetzt wurden.

- Versuchsperson 1034. Frau, 51 Jahre.
  - Wort: *söka*. Übersetzung: *Socken* (falsch).
  - Wort: *försiktig*. Übersetzung: *vorsichtig* (richtig).
  - Wort: *mjölk*. Übersetzung: *Milch* (richtig).
  - Wort: *behärska*. Keine Übersetzung gegeben.
  - Wort: *fiende*. Übersetzung: *finden* (falsch).
- Versuchsperson 2384. Frau, 27 Jahre.
  - Wort: *fiende*. Keine Übersetzung gegeben.

  - Wort: *behärska*. Keine Übersetzung gegeben.
  - Wort: *försiktig*. Übersetzung: *vorsichtig* (richtig).
  - Wort: *mjölk*. Übersetzung: *Milch* (richtig).
  - Wort: *söka*. Übersetzung: *Socke* (falsch).
- Versuchsperson 8667. Frau, 27 Jahre.
  - Wort: *mjölk*. Übersetzung: *Milch* (richtig).
  - Wort: *behärska*. Keine Übersetzung gegeben.
  - Wort: *fiende*. Übersetzung: *finden* (falsch).
  - Wort: *söka*. Übersetzung: *suchen* (richtig).
  - Wort: *försiktig*. Übersetzung: *vorsichtig* (richtig).
- Versuchsperson 5901. Mann, 15 Jahre.
  - Wort: *behärska*. Übersetzung: *beherrschen* (richtig).
  - Wort: *mjölk*. Übersetzung: *milch* (sic.) (richtig).
  - Wort: *försiktig*. Übersetzung: *vorsichtig* (richtig).
  - Wort: *fiende*. Übersetzung: *feinde* (sic.) (richtig; eigentlich *Feind*).
  - Wort: *söka*. Übersetzung: *socken* (sic.) (falsch).

Aufgabe Öffnen Sie ein Spreadsheetprogramm (z.B. MS Excel oder LibreOffice.org Calc) und tragen Sie diese Angaben in ein Spreadsheet ein. (Blättern Sie noch nicht weiter zur nächsten Seite.)



	A	B	C	D	E	F	G	H	I	J	K
1	Versuchsperson	Geschlecht	Alter	söka_Position	söka_Übersetzung	söka_richtig	försiktig_Position	försiktig_Übersetzung	försiktig_richtig	mjölk_Position	mjölk_Übersetzung
2	1034	Frau	51	1	Socken	0	2	vorsichtig	1	3	Milch
3	2384	Frau	27	5	Socke	0	3	vorsichtig	1	4	Milch
4	8667	Frau	27	4	suchen	1	5	vorsichtig	1	1	Milch
5	5901	Mann	15	5	socken	1	3	vorsichtig	1	2	milch

Abbildung 2.1: Ein breiter Datensatz mit einer Zeile pro Versuchsperson.

Vergleichen Sie Ihr Spreadsheet mit denen Ihrer KollegInnen. Können jedem Spreadsheet alle oben aufgelisteten Informationen entnommen werden? Auch die Reihenfolge, in der die Wörter übersetzt wurden? Auch wenn Sie das Spreadsheet nach irgendeiner Spalte sortieren?

Ein kostenloses Spreadsheetprogramm. LibreOffice.org ist eine kostenlose Applikationssuite, die – wie Microsoft Office – aus einem Textbearbeitungsprogramm (Write), einem Spreadsheetprogramm (Calc), einem Präsentationsprogramm (Impress) usw., besteht. Selber finde ich LibreOffice Calc nützlicher als MS Excel, weil man beim Speichern von Spreadsheets gewisse Einstellungen viel einfacher ändern kann. Darauf werden wir später zurückkommen.

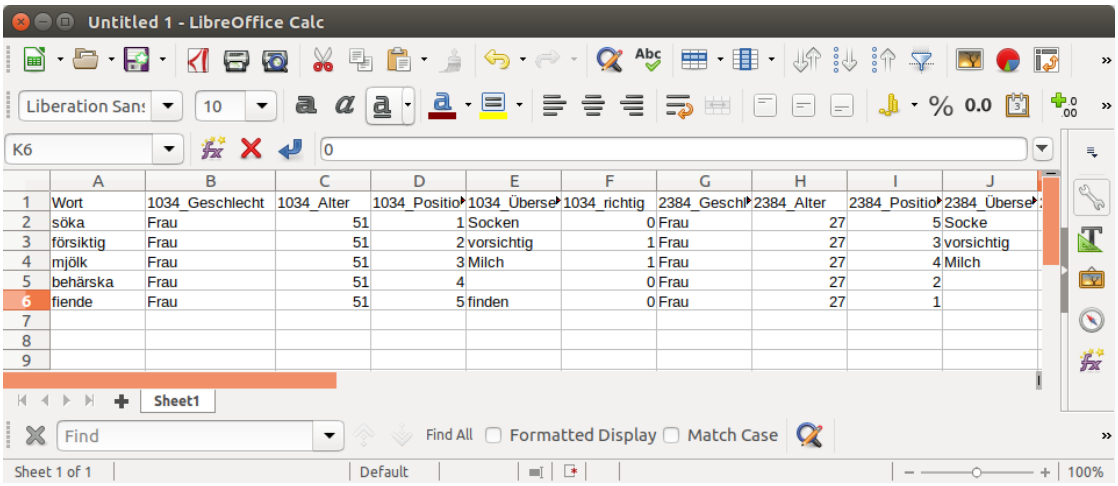
2.1.1 Lange Datensätze sind praktischer als breite

Die zwei üblichsten Formate, in denen Datensätze organisiert werden, sind das *breite* und das *lange* Format. Im breiten Format werden alle Angaben zu einer bestimmten **Erhebungseinheit** (z.B., zu einer Versuchsperson) in der gleichen Zeile eingetragen. In unserem Beispiel könnte ein breiter Datensatz wie in Abbildung 2.1 aussehen (7 Spalten werden nicht gezeigt).

Bemerken Sie, dass es für jedes Wort drei Spalten gibt: eine, in der steht, an welcher Stelle das Wort übersetzt wurde; eine, in der die Übersetzung steht; und eine, in der vermerkt wird, ob die Übersetzung richtig war.

Auch die Anordnung in Abbildung 2.2, in der es eine Zeile pro Wort gibt und alle Übersetzungen für dieses Wort auf der gleichen Zeile stehen, ist ein Beispiel eines breiten Datensatzes (10 Spalten nicht werden nicht gezeigt).

Im langen Format werden die Angaben zu einer **Beobachtungseinheit** in der gleichen Zeile arrangiert. Eine Definition von 'Erhebungseinheit' und 'Beobachtungseinheit' ist schwierig zu geben (siehe Wickham, 2014) und würde ausserdem wenig bringen. In



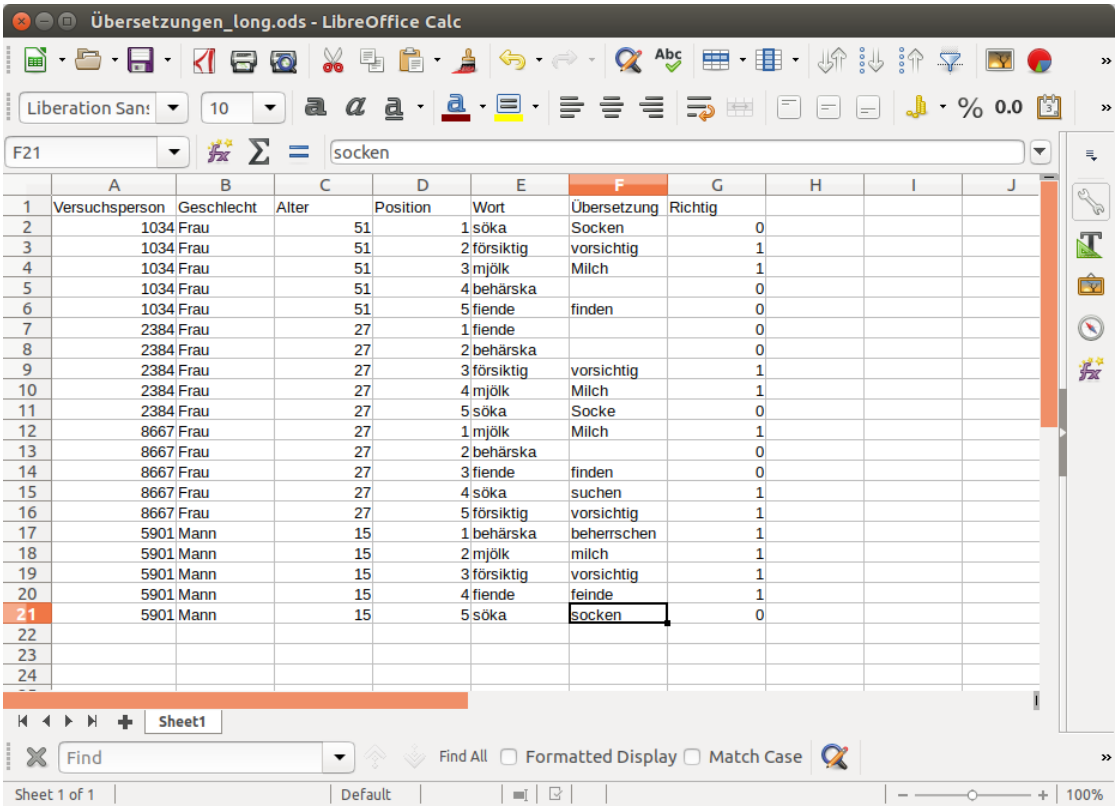
	A	B	C	D	E	F	G	H	I	J
1	Wort	1034_Geschlecht	1034_Alter	1034_Positio	1034_Übers	1034_richtig	2384_Geschl	2384_Alter	2384_Positio	2384_Übers
2	söka	Frau	51	1	Socken	0	Frau	27	5	Socke
3	försiktig	Frau	51	2	vorsichtig	1	Frau	27	3	vorsichtig
4	mjölk	Frau	51	3	Milch	1	Frau	27	4	Milch
5	behärska	Frau	51	4		0	Frau	27	2	
6	fiende	Frau	51	5	finden	0	Frau	27	1	
7										
8										
9										

Abbildung 2.2: Ein breiter Datensatz mit einer Zeile pro Stimulus.

diesem Beispiel wären die Beobachtungseinheiten aber die einzelnen Übersetzungen. Die gleichen Daten im langen Format könnten aussehen wie in Abbildung 2.3.

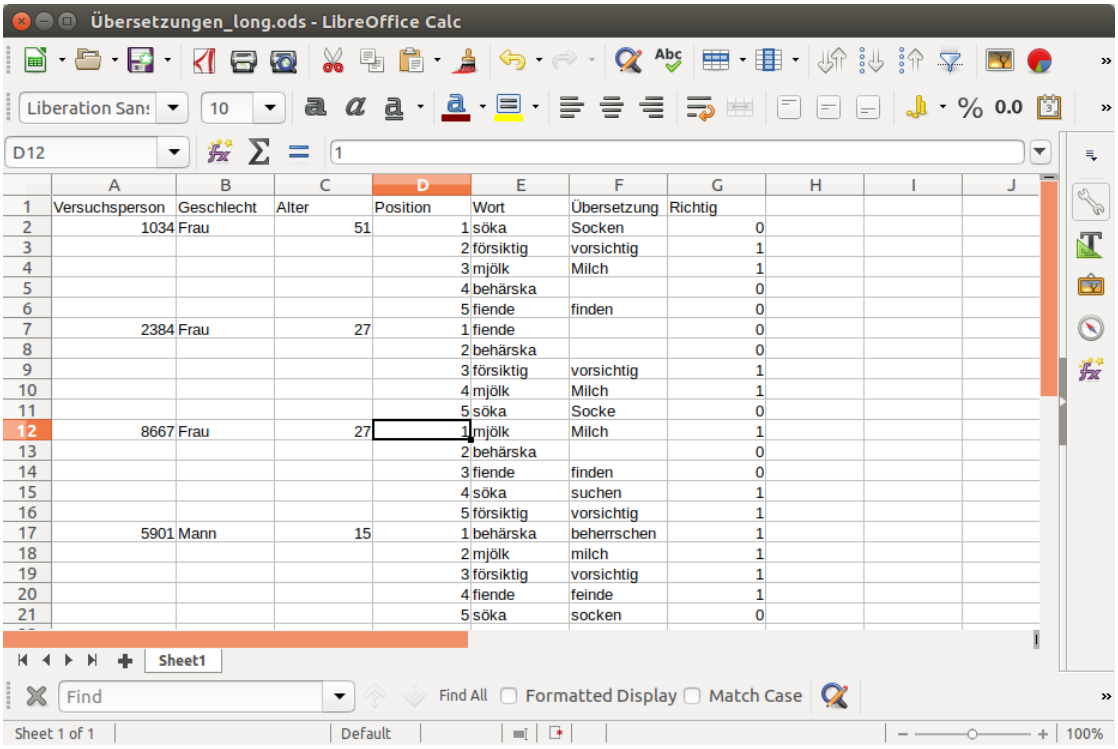
Um zu vermeiden, dass das absichtliche oder versehentliche Löschen einer Zeile dazu führt, dass die anderen Zeilen nicht mehr interpretiert werden können, werden die Angaben zu den Versuchspersonen in jeder Zeile wiederholt. Lassen Sie weder im langen noch im breiten Format Informationen weg, die man einer anderen Zeile entnehmen kann. Also nicht so wie in Abbildung 2.4!

Bemerken Sie auch, dass es eine Spalte gibt, welche die Reihenfolge, in der die Wörter übersetzt wurden, explizit macht. Im Prinzip könnte man diese Information aus der Struktur des Datensatzes ableiten. Indem man diese Information jedoch explizit hinzufügt, vermeidet man, dass sie verloren geht, wenn der Datensatz anders sortiert wird.



	A	B	C	D	E	F	G	H	I	J
1	Versuchsperson	Geschlecht	Alter	Position	Wort	Übersetzung	Richtig			
2	1034	Frau	51	1	söka	Socken	0			
3	1034	Frau	51	2	försiktig	vorsichtig	1			
4	1034	Frau	51	3	mjölk	Milch	1			
5	1034	Frau	51	4	behärska		0			
6	1034	Frau	51	5	fiende	finden	0			
7	2384	Frau	27	1	fiende		0			
8	2384	Frau	27	2	behärska		0			
9	2384	Frau	27	3	försiktig	vorsichtig	1			
10	2384	Frau	27	4	mjölk	Milch	1			
11	2384	Frau	27	5	söka	Socke	0			
12	8667	Frau	27	1	mjölk	Milch	1			
13	8667	Frau	27	2	behärska		0			
14	8667	Frau	27	3	fiende	finden	0			
15	8667	Frau	27	4	söka	suchen	1			
16	8667	Frau	27	5	försiktig	vorsichtig	1			
17	5901	Mann	15	1	behärska	beherrschen	1			
18	5901	Mann	15	2	mjölk	milch	1			
19	5901	Mann	15	3	försiktig	vorsichtig	1			
20	5901	Mann	15	4	fiende	feinde	1			
21	5901	Mann	15	5	söka	socken	0			
22										
23										
24										

Abbildung 2.3: Ein langer Datensatz mit einer Zeile pro Stimulus pro Versuchsperson. Lange Datensätze sind in der Regel einfacher zu verwalten und zu analysieren als breite.



	A	B	C	D	E	F	G
1	Versuchsperson	Geschlecht	Alter	Position	Wort	Übersetzung	Richtig
2	1034	Frau	51	1	söka	Socken	0
3				2	försiktig	vorsichtig	1
4				3	mjölk	Milch	1
5				4	behärska		0
6				5	fiende	finden	0
7	2384	Frau	27	1	fiende		0
8				2	behärska		0
9				3	försiktig	vorsichtig	1
10				4	mjölk	Milch	1
11				5	söka	Socke	0
12	8667	Frau	27	1	mjölk	Milch	1
13				2	behärska		0
14				3	fiende	finden	0
15				4	söka	suchen	1
16				5	försiktig	vorsichtig	1
17	5901	Mann	15	1	behärska	beherrschen	1
18				2	mjölk	milch	1
19				3	försiktig	vorsichtig	1
20				4	fiende	feinde	1
21				5	söka	socken	0

Abbildung 2.4: Nicht so! In diesem Datensatz wurden mehrere Zellen leer gelassen, da man deren Inhalt anderen Zellen entnehmen kann. Dies kann bei der Analyse aber zu Schwierigkeiten führen.

Sowohl breite als auch lange Datensätze sind **rechteckig**:

- Sie haben eine Anzahl Zeilen und Spalten. Alle Zeilen sind gleich lang. Dies gilt auch für die Spalten.
- Es gibt keine komplett leeren Zeilen und Spalten. Einzelne leere Zellen gibt es öfters schon, aber es ist nicht so, dass es Angaben in Spalten A–D gibt, überhaupt keine in Spalten E–F und dann wieder welche in Spalte G.
- In der Regel haben alle Spalten einen Namen. Manchmal kommt es zwar vor, dass keine einzige Spalte einen Namen hat, aber geben Sie nicht ein paar Spalten einen Namen und anderen nicht.
- Alle Spaltennamen stehen in *einer* Zeile. Die Spaltenbezeichnungen stellen sich also nicht aus mehreren Zellen zusammen.

Zum Vergleich: Das Spreadsheet in Abbildung 2.5 zeigt einen nicht-rechteckigen Datensatz. Die Zusammenfassungen unten haben in diesem Datensatz nichts zu suchen und würden beim Einlesen zu Problemen führen. *Rechnen Sie nicht im Datensatz selber herum!*

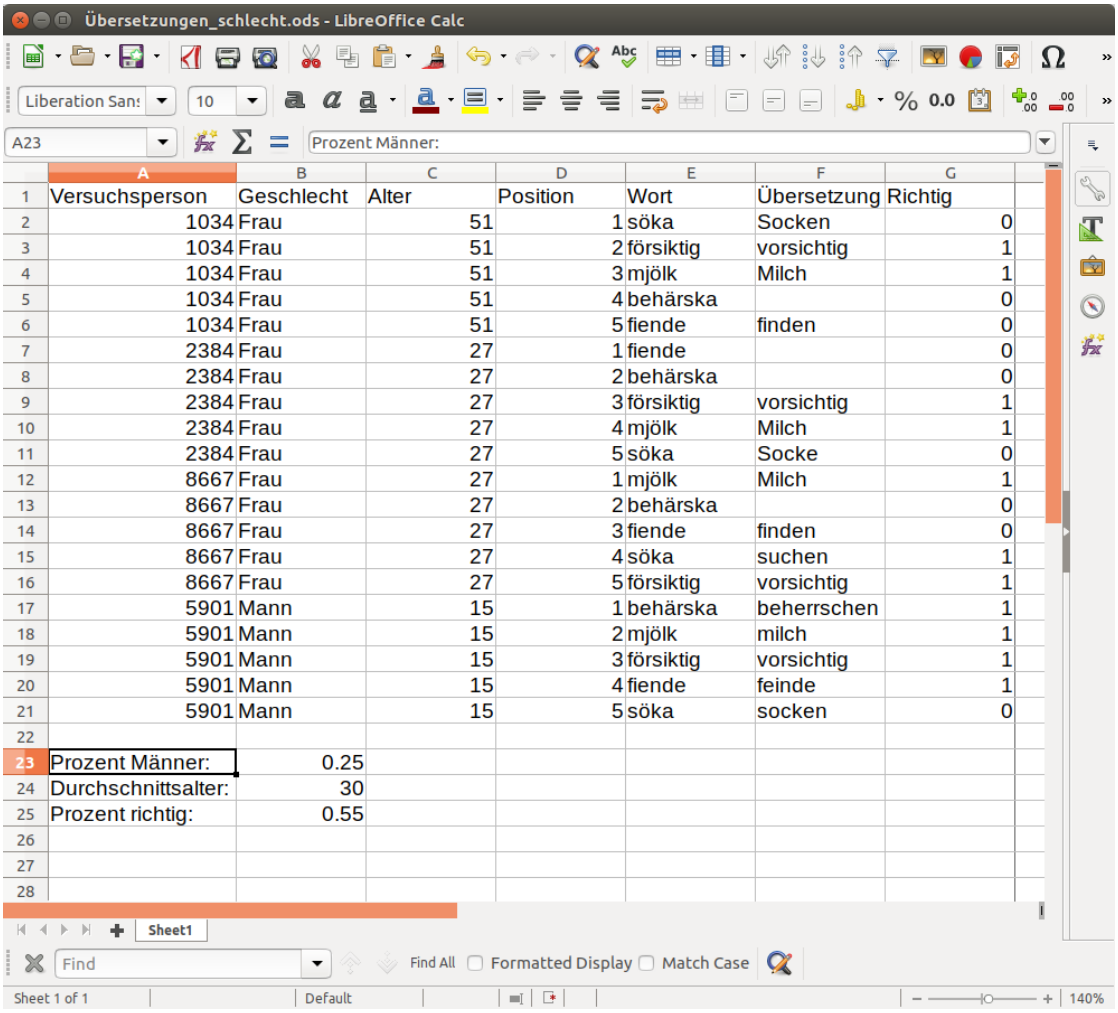
Was ist nun besser: breite oder lange Datensätze? Für manche Analysetechniken braucht man Daten im breiten Format, für andere im langen. Erfahrungsgemäss können Datensätze im langen Format jedoch einfacher ins breite Format konvertiert werden als andersherum. Wenn man die Wahl hat, sollte man sich also für das lange Format entscheiden.

2.1.2 Kurze, aber selbsterklärende Bezeichnungen verwenden

Machen Sie sich die spätere Analyse einfacher, indem Sie den Spalten und anderen Angaben in Ihren Datensätzen deutliche Namen geben. Dadurch vermeiden Sie, dass Sie während der Analyse ständig wieder nachschlagen müssen, was die Angaben überhaupt heissen. Dies verringert wiederum die Wahrscheinlichkeit, dass Sie Fehler machen.

Ein paar Beispiele:

- Sie werten einen Fragebogen aus. Machen Sie in jeder Spalte deutlich, worum es in den Fragen ging. Vermeiden Sie also Spaltennamen wie ‘Q3’ oder ‘Frage8’. Verwenden Sie stattdessen Spaltennamen wie ‘DiplomVater’ (wenn die Frage war, welchen Schulabschluss der Vater der Gewährsperson hat) oder ‘DialectUse’ (wenn die Frage war, wie oft die Gewährsperson Dialekt redet).
- Wenn Sie eine Spalte namens ‘Geschlecht’ haben, die mit Nullen und Einsen gefüllt ist, müssten Sie ständig nachschlagen, ob 0 jetzt für ‘Frau’ oder ‘Mann’ steht. Verwenden Sie stattdessen lieber direkt ‘Frau’ und ‘Mann’, oder sogar ‘f’ und ‘m’. Eine andere Möglichkeit ist, dass Sie die Spalte zu ‘Frau’ umbenennen, sodass es



	A	B	C	D	E	F	G
1	Versuchsperson	Geschlecht	Alter	Position	Wort	Übersetzung	Richtig
2	1034	Frau	51	1	söka	Socken	0
3	1034	Frau	51	2	försiktig	vorsichtig	1
4	1034	Frau	51	3	mjölk	Milch	1
5	1034	Frau	51	4	behärska		0
6	1034	Frau	51	5	fiende	finden	0
7	2384	Frau	27	1	fiende		0
8	2384	Frau	27	2	behärska		0
9	2384	Frau	27	3	försiktig	vorsichtig	1
10	2384	Frau	27	4	mjölk	Milch	1
11	2384	Frau	27	5	söka	Socke	0
12	8667	Frau	27	1	mjölk	Milch	1
13	8667	Frau	27	2	behärska		0
14	8667	Frau	27	3	fiende	finden	0
15	8667	Frau	27	4	söka	suchen	1
16	8667	Frau	27	5	försiktig	vorsichtig	1
17	5901	Mann	15	1	behärska	beherrschen	1
18	5901	Mann	15	2	mjölk	milch	1
19	5901	Mann	15	3	försiktig	vorsichtig	1
20	5901	Mann	15	4	fiende	feinde	1
21	5901	Mann	15	5	söka	socken	0
22							
23	Prozent Männer:	0.25					
24	Durchschnittsalter:	30					
25	Prozent richtig:	0.55					
26							
27							
28							

Abbildung 2.5: Nicht so! Dieser Datensatz ist nicht rechteckig—"Prozent Männer:" ist keine Versuchspersonnummer, und "0.25" ist kein Geschlecht. Ausserdem ist Zeile 22 komplett leer.

deutlich ist, dass eine 1 heisst, dass die Gewährsperson eine Frau war, und eine 0, dass es sich um einen Mann handelte.

- Verwenden Sie eher kurze Bezeichnungen. In der späteren Analyse werden Sie nämlich insbesondere die Spaltennamen mehrmals wieder eintippen müssen. Vermeiden Sie daher Spaltennamen wie 'wie_oft_sprechen_Sie_Hochdeutsch' und verwenden Sie stattdessen 'use_hochdeutsch' oder Ähnliches.

Am besten verwenden Sie übrigens keine Leertasten und Lesezeichen in den Spaltennamen.

2.1.3 Fehlende Angaben unzweideutig vermerken

Im Beispiel oben habe ich fehlende Übersetzungen einfach leer gelassen. Daraus kann ich ableiten, dass der Versuchsperson das Wort zwar vorgelegt wurde, aber sie dieses nicht übersetzt hat. Es wäre jedoch auch möglich gewesen, dass einigen Versuchspersonen bestimmte Wörter gar nie vorgelegt wurden, z.B., aufgrund eines Softwarefehlers. Es wäre wichtig, solche Fälle von den ersten zu unterscheiden, indem man diese Fälle mit einer Kürzel (z.B. 'NA' für 'not available' oder 'not applicable') vermerkt.

Gegebenenfalls kann man auch mehrere Kürzel verwenden, um unterschiedliche Gründe für das Nicht-Vorhanden-Seins voneinander zu unterscheiden. In der Regel ist es aber am einfachsten, sämtliche fehlende Daten mit 'NA' zu vermerken und den Grund hierfür in eine Kommentarspalte einzutragen.

Verwenden Sie keine Zahlen (wie -99 oder -9999), um fehlende Angaben zu vermerken. Der Grund ist, dass solche Zahlen manchmal zulässige Werte sind. Ausserdem können solche Angaben schwieriger auf den ersten Blick erkannt werden können, wenn man eine Zusammenfassung des Datensatzes generiert.

2.1.4 Mehrere kleinere Datensätze sind handlicher als ein riesiger

In den Spreadsheets oben wurden bestimmte Informationen mehrfach wiederholt. Zum Beispiel musste man bei den Übersetzungen von Versuchsperson 1034 fünf Mal eintragen, dass sie eine Frau im Alter von 51 Jahren war. In diesem Fall ist der Datensatz trotz wiederholten Informationen übersichtlich. Wenn man sich aber überlegt, dass in der Regel für jede Versuchsperson viel mehr Informationen vorliegen (z.B., Daten aus einem Hintergrundsfragebogen) und dass man öfters auch Informationen zu den verwendeten Stimuli (hier: den Wörtern) mit einbeziehen will, wird klar, dass man es schnell mit grossen Datensätzen zu tun hat, in denen viele Informationen mehrfach wiederholt werden.

Um die Übersicht zu bewahren und um sich eine Menge Tipp- oder Kopierarbeit zu sparen, lohnt es sich, statt eines grossen Datensatzes mehrere kleinere zu verwalten.

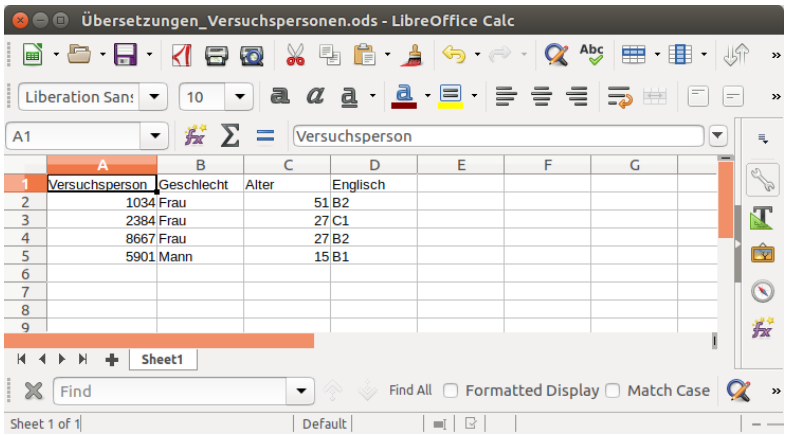


Abbildung 2.6: Ein erster Datensatz mit Informationen, die nur die Versuchspersonen betreffen.

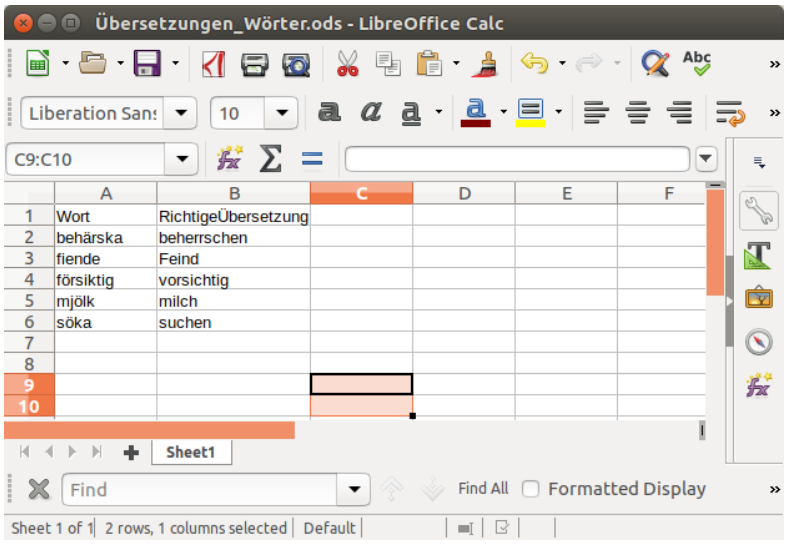
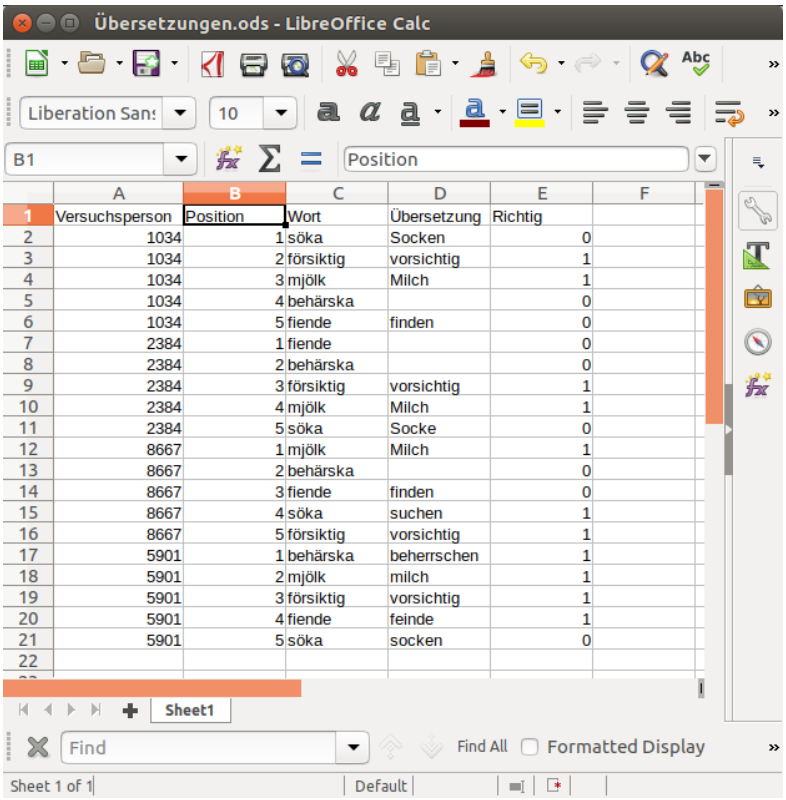


Abbildung 2.7: Ein zweiter Datensatz mit Informationen, die nur die Stimuli betreffen.

In unserem Beispiel würde man dann ein Spreadsheet mit Informationen zu den Versuchspersonen gestalten (Abbildung 2.6). Daneben kann man noch ein Spreadsheet mit Informationen zu den Wörtern anfertigen (Abbildung 2.7). In einem dritten Spreadsheet kann man dann die Antworten auf die Übersetzungsaufgabe aufführen (Abbildung 2.8).

Da im letzten Spreadsheet sowohl eine Spalte mit den Identifikationen der Versuchspersonen als auch mit den Bezeichnungen der Stimuli vorhanden ist, können ihm die Informationen aus den ersten zwei kleineren Datensätzen nachher problemlos hinzugefügt werden. Wie man dies in R machen kann, erfahren Sie später in diesem Kapitel.



	A	B	C	D	E	F
1	Versuchsperson	Position	Wort	Übersetzung	Richtig	
2	1034	1	söka	Socken	0	
3	1034	2	försiktig	vorsichtig	1	
4	1034	3	mjölk	Milch	1	
5	1034	4	behärska		0	
6	1034	5	fiende	finden	0	
7	2384	1	fiende		0	
8	2384	2	behärska		0	
9	2384	3	försiktig	vorsichtig	1	
10	2384	4	mjölk	Milch	1	
11	2384	5	söka	Socke	0	
12	8667	1	mjölk	Milch	1	
13	8667	2	behärska		0	
14	8667	3	fiende	finden	0	
15	8667	4	söka	suchen	1	
16	8667	5	försiktig	vorsichtig	1	
17	5901	1	behärska	beherrschen	1	
18	5901	2	mjölk	milch	1	
19	5901	3	försiktig	vorsichtig	1	
20	5901	4	fiende	feinde	1	
21	5901	5	söka	socken	0	
22						

Abbildung 2.8: Ein dritter Datensatz, in denen die Übersetzungen aufgeführt werden.

2.1.5 Gemischtes

- Beachten Sie Gross- und Kleinschreibung. Für manche Statistikprogramme ist ‘Frau‘ gleich ‘frau’, für andere (darunter R) nicht.
- Sonderzeichen, wie Umlaute, führen manchmal zu Problemen.
- Beachten Sie Leerzeichen. Für ein Computer ist ‘Mann’ nicht gleich ‘Mann ’ (mit Leerzeichen).
- Wenn Sie in Ihren Spreadsheets gerne mit Farben arbeiten: Diese gehen verloren, wenn Sie das Spreadsheet in R einlesen. Wenn die Farben Informationen kodieren, die nicht den Angaben im Spreadsheet entnommen werden können, fügen Sie diese Informationen also besser noch selbst hinzu.

2.2 Datensätze in R einlesen und zusammenfügen

Wenn die Daten in einem analysierbaren Format vorliegen, besteht die nächste Herausforderung darin, diese in R einzulesen. Hier behandeln wir nur zwei Fälle: das Einlesen von Excel-Spreadsheets im XLS(X)-Format und das Einlesen von Spreadsheets, die im CSV-Format gespeichert wurden. Ausserdem wird auch gezeigt, wie Datensätze zu-

sammengefügt werden können.

Zum Einlesen von SPSS-, Stata- und SAS-Dateien, siehe `http://haven.tidyverse.org/`.

2.2.1 Excel-Spreadsheets (XLS, XLSX)

Speichern Sie den Datensatz `uebersetzungen.xlsx` in den Ordner `Data` in Ihrem Arbeitsordner. Dieser Datensatz ist eine Exceldatei, die aus einem einzigen Spreadsheet besteht. Um ihn in R einzulesen, können Sie die folgenden Befehle ausführen.

```
# Funktionen aus dem Package "readxl" zugänglich machen
library(readxl)

# Datensatz "uebersetzungen.xlsx" einlesen
# und als 'translations' in R zugänglich machen.
# Der Datensatz steht im Ordner "Data"
translations <- read_excel("Data/uebersetzungen.xlsx")
```

Der Datensatz ist nun zugänglich in einem sog. *tibble* namens `translations`.[1]

<-, = und == Die Symbolenfolge `<-` ist der *assignment operator*. Sie kreiert ein neues Objekt im Arbeitsgedächtnis bzw. überschreibt ein bereits vorhandenes Objekt. Dieses Objekt trägt den Namen links von `<-`.

Kürzel in RStudio: ALT + -.

Oft wird auch das Ist-Gleich-Zeichen (=) als *assignment operator* verwendet. Es wird aber auch verwendet, um Parameter in Funktionen festzulegen (wie Sie bald sehen werden). Zwecks *one form, one function* werde ich daher ausschliesslich `<-` als *assignment operator* verwenden.

Zu guter Letzt gibt es noch die Symbolenfolge ==. Diese überprüft, ob zwei Werte identisch sind. Beispiel:

```
4 == 2 * 2

## [1] TRUE

8 == 2 * 3

## [1] FALSE
```

Fehlermeldungen Viele Fehlermeldungen in R sind notorisch unverständlich. Im Anhang finden Sie eine Liste mit den häufigsten Fehlermeldungen und möglichen Auslö-

[1] Rechteckige Datensätze heissen in R eigentlich *data frames*. *Tibbles* sind die Entsprechung von *data frames* in den *tidyverse*-Paketen, darunter auch das Paket `readxl`. Wer schon viel Erfahrung mit R hat, wird im Laufe des Skripts ein paar subtile Unterschiede zwischen *data frames* und *tibbles* feststellen, aber im Grossen und Ganzen sind sie klein.

sern und Lösungen. Wenn der Anhang Ihnen nicht weiterhilft, kleben Sie die Fehlermeldung am besten in Google ein. Dann hilft es, wenn die Fehlermeldungen auf Englisch und nicht auf Deutsch oder Französisch angezeigt werden. Leider ist dies ziemlich mühsam, um einzustellen. Siehe etwa `https://stackoverflow.com/q/12760491/1331521`.

Wenn Sie die folgende Fehlermeldung erhalten, heisst das, dass R die Datei am falschen Ort gesucht hat.

```
Error in read_fun(path = path, sheet = sheet, limits = limits, shim = shim,  :
  Evaluation error: zip file 'Data/uebersetzungen.xlsx' cannot be opened.
```

Haben Sie die Befehle richtig eingetippt? Haben Sie den Arbeitsordner richtig eingestellt? Haben Sie die Datei am richtigen Ort gespeichert?

Kontrollieren, ob die Daten richtig eingelesen wurden Auch wenn Sie keine Fehlermeldung erhalten, sollten Sie kontrollieren, ob der Datensatz richtig eingelesen wurde. Hierzu gibt es ein paar Möglichkeiten, darunter:

```
# Die ersten Zeilen anzeigen lassen:
head(translations)
```

```
## # A tibble: 6 x 5
##   Versuchsperson Position Wort      Übersetzung Richtig
##            <dbl>    <dbl> <chr>     <chr>         <dbl>
## 1           1034        1 söka      Socken            0
## 2           1034        2 försiktig vorsichtig        1
## 3           1034        3 mjölk     Milch             1
## 4           1034        4 behärska  <NA>              0
## 5           1034        5 fiende    finden            0
## 6           2384        1 fiende    <NA>              0
```

Alles scheint in Ordnung zu sein. Bemerken Sie aber, dass leere Zellen als `<NA>` vermerkt wurden.

```
# Die Anzahl Spalten abfragen:
ncol(translations)
```

```
## [1] 5
```

```
# Stimmt.

# Die Anzahl Zeilen abfragen:
nrow(translations)
```

```
## [1] 20
```

```
# Stimmt.
```

Um den ganzen Datensatz in RStudio anzuschauen, können Sie die `View()`-Funktion verwenden:

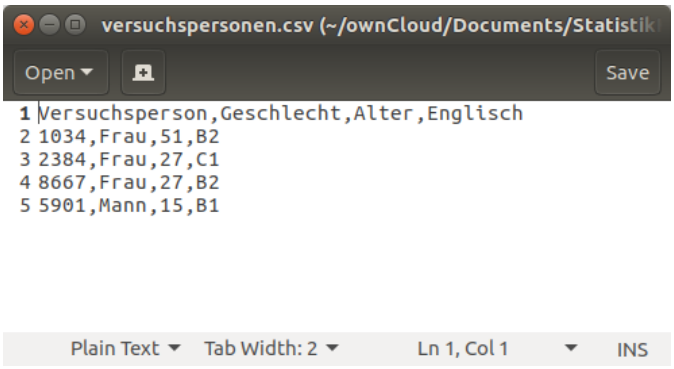


Abbildung 2.9: Ein Datensatz, der als *comma-separated values* gespeichert ist.

```
View(translations)
```

Wenn die Daten nicht richtig eingelesen wurden, kontrollieren Sie am besten nochmals, ob das Spreadsheet nach den Regeln der Kunst formattiert wurde.

Für mehr Details zur `read_excel()`-Funktion, siehe `http://readxl.tidyverse.org/`.

2.2.2 CSV-Dateien

Ein beliebtes Format, um Datensätze zu speichern und mit anderen zu teilen, ist das CSV-Format. CSV steht für *comma-separated values*: Die Zellen auf der gleichen Zeile werden durch Kommas voneinander getrennt; siehe Abbildung 2.9.

In Excel ist es blöderweise eher schwierig, Datensätze im CSV-Format zu speichern: Zwar gibt es diese Option, aber auf deutsch- oder französischsprachigen Systemen werden statt Kommas Semikolonen als Trennzeichen verwendet. In LibreOffice.org hingegen wird man jedes Mal gefragt, ob man Kommas oder Semikolonen verwenden will.

Manchmal werden Texteinträge auch noch zwischen Anführungszeichen gestellt, sodass Kommas auch in einem Textfeld vorkommen können. Die folgenden Funktionen erkennen dies in der Regel automatisch.[2]

```
library(tidyverse)
participants <- read_csv("Data/versuchspersonen.csv")
```

Sie werden beim Ausführen dieses Befehls ein paar Mitteilungen auf dem Bildschirm sehen. Lesen Sie nun noch den Datensatz mit den zu übersetzenden Wörtern ein.

```
items <- read_csv("Data/woerter.csv")
```

[2]Wenn Sie vorher schon mit R gearbeitet haben, ist die Wahrscheinlichkeit gross, dass Sie statt der `read_csv()`-Funktion (mit _) die `read.csv()`-Funktion (mit .) verwendet haben. `read.csv()` ist die Einlesefunktion von *base R*; `read_csv()` ist ihre Entsprechung aus dem *tidyverse*.

Unterschiedliche CSV-Formate. Wenn Sie auf einem französisch- oder deutschsprachigen Computersystem in Excel ein Spreadsheet im 'CSV-Format' speichern, werden die unterschiedlichen Zellen nicht mit Kommas sondern mit Semikolonen voneinander getrennt. Der Grund ist, dass das Komma in diesen Sprachen als Dezimaltrennzeichen dient und daher nicht mehr zur Trennung von Zellen verwendet werden kann. 'CSV'-Dateien, in denen Zellen durch Semikolonen getrennt werden, können Sie in R einlesen, indem Sie statt der Funktion `read_csv()` die Funktion `read_csv2()` verwenden.

In LibreOffice.org kann man für jede Datei selber einstellen, ob Kommas oder Semikolonen zur Trennung von Zellen verwendet werden sollten, und welches Symbol als Dezimaltrennzeichen dienen soll.

2.2.3 Datensätze zusammenfügen

Wir haben drei Datensätze eingelesen: einen mit den Antworten in der Übersetzungsaufgabe, einen mit Informationen zu den zu übersetzenden Wörtern, und einen mit Informationen zu den Versuchspersonen. Um die Daten auszuwerten, müssten diese Datensätze miteinander verknüpft werden. Zum Beispiel müssten wir dem Datensatz `translations` drei Spalten mit Informationen zu den jeweiligen Versuchspersonen hinzufügen: `Geschlecht`, `Alter`, `Englisch`. Für Zeilen in `translations`, für die `Versuchsperson` `1034` ist, sind die Einträge also `Frau`, `51` und `B2`; ist `Versuchsperson` = `5901`, sind die Einträge `Mann`, `15` und `B1`. Solange die gemeinsame Spalte in den beiden Datensätzen identisch heisst, ist dies ein Kinderspiel:

```
# Kombinierten Datensatz als "all_data" speichern.
all_data <- left_join(x = translations,
                      y = participants)

## Joining, by = "Versuchsperson"

head(all_data)

## # A tibble: 6 x 8
##   Versuchsperson Position Wort  Übersetzung Richtig
##            <dbl>    <dbl> <chr> <chr>         <dbl>
## 1           1034        1 söka  Socken            0
## 2           1034        2 förs~ vorsichtig        1
## 3           1034        3 mjölk Milch             1
## 4           1034        4 behä~ <NA>              0
## 5           1034        5 fien~ finden            0
## 6           2384        1 fien~ <NA>              0
## # ... with 3 more variables: Geschlecht <chr>, Alter <dbl>,
## #   Englisch <chr>
```

Die Funktion erkennt, dass es in beiden Datensätzen eine Spalte namens `Versuchsperson`

gibt und verwendet diese als 'Reissverschluss'.

Um auch noch Informationen zu den zu übersetzenden Wörtern hinzuzufügen, wiederholen wir den Befehl mit `y = items`.

```
all_data <- left_join(x = all_data,
                      y = items)

## Joining, by = "Wort"

head(all_data)

## # A tibble: 6 x 9
##   Versuchsperson Position Wort  Übersetzung Richtig
##            <dbl>    <dbl> <chr> <chr>         <dbl>
## 1           1034        1 söka  Socken            0
## 2           1034        2 förs~ vorsichtig        1
## 3           1034        3 mjölk Milch             1
## 4           1034        4 behä~ <NA>              0
## 5           1034        5 fien~ finden            0
## 6           2384        1 fien~ <NA>              0
## # ... with 4 more variables: Geschlecht <chr>, Alter <dbl>,
## #   Englisch <chr>, RichtigeÜbersetzung <chr>
```

Die `left_join()`-Funktion bewirkt, dass alle Einträge in Datensatz `x` bewahrt bleiben und diesem Datensatz die entsprechenden Informationen aus Datensatz `y` hinzugefügt werden, insofern welche vorhanden sind. Wenn es keine Entsprechung in `y` gibt, erscheint in den hinzugefügten Spalten `NA`. Weitere 'join'-Funktionen (siehe `http://dplyr.tidyverse.org/reference/join.html`):

- `right_join()`: Alle Einträge aus Datensatz `y` bleiben bewahrt; Entsprechungen aus `x` werden hinzugefügt, falls vorhanden.
- `full_join()`: Alle Einträge aus beiden Datensätzen bleiben bewahrt. `NA`, falls es im jeweils anderen Datensatz keine Entsprechung gibt.
- `inner_join()`: Nur Einträge aus Datensatz `x`, für die es eine Entsprechung in `y` gibt, bleiben bewahrt. Diese Entsprechungen werden hinzugefügt.
- `semi_join()`: Nur Einträge aus Datensatz `x`, für die es eine Entsprechung in `y` gibt, bleiben bewahrt. Diese Entsprechungen werden *nicht* hinzugefügt.
- `anti_join()`: Nur Einträge aus Datensatz `x`, für die es *keine* Entsprechung in `y` gibt, bleiben bewahrt.

In diesem Beispiel würden `left_join()`, `right_join()`, `full_join()` und `inner_join()` zum gleichen Resultat führen, aber dies ist nicht immer der Fall.

2.3 Daten in R anzeigen

2.3.1 Alle Daten anzeigen: `View()`

`View()` öffnet eine neue Registerkarte in RStudio, in der der ganze Datensatz aufgeführt wird.

```
View(all_data)
```

2.3.2 Zeilen nach Zeilennummer auswählen: `slice()`

Mit diesem Befehl zeigen wir die dritte Spalte des Datensatzes `all_data` an. Am Datensatz ändert sich hierdurch nichts—wir verlieren die anderen 19 Zeilen also nicht.

```
slice(all_data, 3)
```

```
## # A tibble: 1 x 9
##   Versuchsperson Position Wort  Übersetzung Richtig
##            <dbl>    <dbl> <chr> <chr>         <dbl>
## 1           1034        3 mjölk Milch             1
## # ... with 4 more variables: Geschlecht <chr>, Alter <dbl>,
## #   Englisch <chr>, RichtigeÜbersetzung <chr>
```

Eine alternative Schreibweise ist diese:

```
# Nehme "all_data", DANN
#  nehme dritte Zeile
all_data %>%
  slice(3)
```

```
## # A tibble: 1 x 9
##   Versuchsperson Position Wort  Übersetzung Richtig
##            <dbl>    <dbl> <chr> <chr>         <dbl>
## 1           1034        3 mjölk Milch             1
## # ... with 4 more variables: Geschlecht <chr>, Alter <dbl>,
## #   Englisch <chr>, RichtigeÜbersetzung <chr>
```

`%>%`? Die Symbolenfolge `%>%` wird *pipe* genannt und wird im *tidyverse* verwendet, um Befehle übersichtlicher zu organisieren. Es wird als *dann* (*then*) ausgesprochen. Für einfache Befehle wie diesen gibt es eigentlich keinen Mehrwert. Aber sobald wir mehrere Befehle kombinieren, ist die Notation mit *pipes* wesentlich einfacher zu lesen und zu verstehen.

Kürzel in RStudio: CTRL + SHIFT + M.

Im Folgenden werden die Ergebnisse der Befehle nicht mehr angezeigt. Probieren Sie die Befehle aber dennoch aus.

```
# Nehme "all_data", DANN
#  nehme Zeilen 5 und 7
all_data %>%
  slice(c(5, 7))

# Nehme "all_data", DANN
#  nehme Zeilen 5 BIS 7
all_data %>%
  slice(5:7)
```

```
# Zeilen 5 bis 7 vollständig anzeigen
all_data %>%
  slice(5:7) %>%
  View()
```

2.3.3 Zeilen nach bestimmten Werten auswählen: `filter()`

```
# Nehme "all_data", DANN
#  nehme jene Zeilen, bei denen 'Wort' 'fiende' ist.
all_data %>%
  filter(Wort == "fiende")

# Nehme "all_data", DANN
#  nehme jene Zeilen, bei denen 'Alter' über 30 ist.
all_data %>%
  filter(Alter > 30)

# Nehme "all_data", DANN
#  nehme jene Zeilen, bei denen 'Position' 1 ist, DANN
#  nehme jene Zeilen, bei denen 'Richtig' 0 ist.
all_data %>%
  filter(Position == 1) %>%
  filter(Richtig == 0)
```

Die Ergebnisse all dieser Auswählaktionen können auch in separaten Objekten gespeichert werden.

```
# Kreiere neues Objekt namens "nur_fiende":
# nehme "all_data", DANN
#  nehme jene Zeilen, bei denen 'Wort' 'fiende' ist.
nur_fiende <- all_data %>%
  filter(Wort == "fiende")

# neues Objekt anzeigen
```

```
nur_fiende
```

2.3.4 Spalten auswählen: `select()`

```
# Nehme "all_data", DANN
#  selektiere Spalten 'Wort', 'RichtigeÜbersetzung', 'Übersetzung'
all_data %>%
  select(Wort, RichtigeÜbersetzung, Übersetzung)
```

Es gibt auch ein paar Hilfefunktionen, mit denen man effizienter Spalten auswählen kann. Diese sind insbesondere bei grossen Datensätzen nützlich. Beispiele sind `contains()` und `starts_with()`. Die unten stehenden Beispiele zeigen Ihnen ausserdem, wie Sie unterschiedliche Befehle kombinieren können.

```
# Nehme "all_data", DANN
#  selektiere alle Spalten, in deren Namen "Übersetzung" vorkommt, DANN
#  nehme Zeilen 1 bis 3
all_data %>%
  select(contains("Übersetzung")) %>%
  slice(1:3)

# Nehme "all_data", DANN
#  selektiere alle Spalten, deren Namen mit "Richt" anfängt, DANN
#  nehme Zeilen 16 bis zur letzten Zeile.
#  Tipp: n() ergibt die Anzahl Zeilen im Objekt.
#  '16:n()' selektiert somit Zeilen 16 bis und mit der letzten Zeile.
all_data %>%
  select(starts_with("Richt")) %>%
  slice(16:n())
```

Für weitere Hilfefunktionen, siehe `http://dplyr.tidyverse.org/reference/select_helpers.html`.

2.3.5 Weitere Beispiele

```
# Nehme "all_data", DANN
#  behalte die Einträge, bei denen 'Wort' 'fiende' ist, DANN
#  selektiere nur Spalte 'Übersetzung'
all_data %>%
  filter(Wort == "fiende") %>%
  select(Übersetzung)
```

```
# Nehme "all_data", DANN
#  behalte die Einträge, bei denen 'Wort' 'behärska' ist, DANN
```

```
#  selektiere nur Spalte 'Übersetzung', DANN
#  behalte nur die *unterschiedlichen* Zeilen:
all_data %>%
  filter(Wort == "behärska") %>%
  select(Übersetzung) %>%
  distinct()
```

Im letzten Beispiel wird auch langsam klar, wieso es sich lohnt, das *pipe* (`%>%`) zu verwenden. Ohne sähe diese Befehlenkombination nämlich so aus:

```
distinct(select(filter(all_data, Wort == "behärska"), Übersetzung))
```

`filter()` ist der Befehl, der zuerst ausgeführt werden muss, wird in dieser Notation aber zuletzt geschrieben. Mit der pipe-Notation gibt es dieses Problem nicht.

2.4 Literaturempfehlungen

Zum Verwalten von Spreadsheets, siehe meinen Blogeintrag zu *Some tips on preparing your data for analysis* (18.6.2015). Broman & Woo (2017) haben weitere nützliche Hinweise.

2.5 Übung

Slavin et al. (2011) berichten die Ergebnisse einer mehrjährigen Evaluationsstudie, in der zwei Unterrichtsprogramme miteinander verglichen wurden. Schülern und Schülerinnen (SuS) in beiden Programmen wurden unter anderem ein spanischer und ein englischer Vokabeltest vorgelegt, und zwar in der 1., der 2., der 3. und der 4. Klasse. Die vier Tabellen auf der nächsten Seite zeigen einen Teil der Ergebnisse, die Slavin et al. (2011) berichten; es handelt sich dabei um die durchschnittlichen Vokabeltestergebnisse der getesteten SuS.

1. Tragen Sie diese Daten in ein Spreadsheet im *langen Format* ein. Jede Zeile soll das Ergebnis in einer einzigen Klasse, in einer einzigen Sprache und von einem einzigen Programm enthalten. Sie brauchen also 16 Zeilen mit Daten und eine Zeile mit passenden Spaltennamen.
2. Speichern Sie dieses Spreadsheet im CSV-Format.
3. Lesen Sie das Spreadsheet in R ein.
4. Kontrollieren Sie, ob das Spreadsheet richtig eingelesen wurde.
5. Zeigen Sie in R nun *nur* die Englischergebnisse im *transitional bilingual*-Programm an.

6. Zeigen Sie die Spanischergebnisse in der 1. und 2. Klasse im *English immersion*-Programm an.

Tabelle 2.1: Vokabeltestergebnisse 1. Klasse

	Transitional bilingual	English immersion
English	74.98	79.90
Spanish	99.85	90.19

Tabelle 2.2: Vokabeltestergebnisse 2. Klasse

	Transitional bilingual	English immersion
English	80.40	81.13
Spanish	92.94	87.54

Tabelle 2.3: Vokabeltestergebnisse 3. Klasse

	Transitional bilingual	English immersion
English	84.76	85.45
Spanish	92.86	85.64

Tabelle 2.4: Vokabeltestergebnisse 4. Klasse

	Transitional bilingual	English immersion
English	88.07	90.36
Spanish	91.00	86.27

Teil II

Populationen und Stichproben

Kapitel 3

Eine einzige numerische Variable beschreiben

Für meine Bachelorarbeit habe ich 23 Studierenden im zweiten Jahr im Fach schwedische Sprach- und Literaturwissenschaft an der Universität Gent (Belgien) vier Leseverstehensaufgaben vorgelegt: einen auf Schwedisch, einen auf auf Dänisch, einen auf bokmål-Norwegisch und einen auf nynorsk-Norwegisch. Die Ergebnisse aus den unterschiedlichen Lesetests sind nicht miteinander vergleichbar. (Im Datensatz sind die nynorsk-Daten nicht vorhanden.) Daneben gibt es Angaben zu den sonstigen Sprachkenntnissen der Teilnehmenden; diese ignorieren wir hier.

```
d <- read_csv("Data/jv_bachpap.csv")
d %>%
  slice(1:2)
```

```
## # A tibble: 2 x 9
##   LvlFrench LvlEnglish LvlGerman LvlSpanish NoLanguages
##       <dbl>      <dbl>     <dbl>      <dbl>       <dbl>
## 1         5          5         5          2           5
## 2         5          4         3          0           3
## # ... with 4 more variables: Swedish <dbl>, Danish <dbl>,
## #   Norwegian <dbl>, Participant <chr>
```

In diesem Kapitel widmen wir uns der grafischen und numerischen Beschreibung einer einzigen numerischen Variablen: den Ergebnissen beim norwegischen Lesetest. Vorübergehend gehen wir davon aus, dass wir uns ausschliesslich für die 23 Ergebnisse im Datensatz interessieren und keine allgemeineren Aussagen machen möchten—z.B. über das Leseverständnis im Norwegischen von Studierenden im zweiten Jahr im Fach schwedische Sprach- und Literaturwissenschaft, die nicht im Datensatz vorhanden sind. Wir betrachten die Ergebnisse, die uns zur Verfügung stehen, also als die ganze **Population**, für die wir uns interessieren, und nicht als bloss eine **Stichprobe**, d.h., einen Teil der Population, die von Interesse wäre.

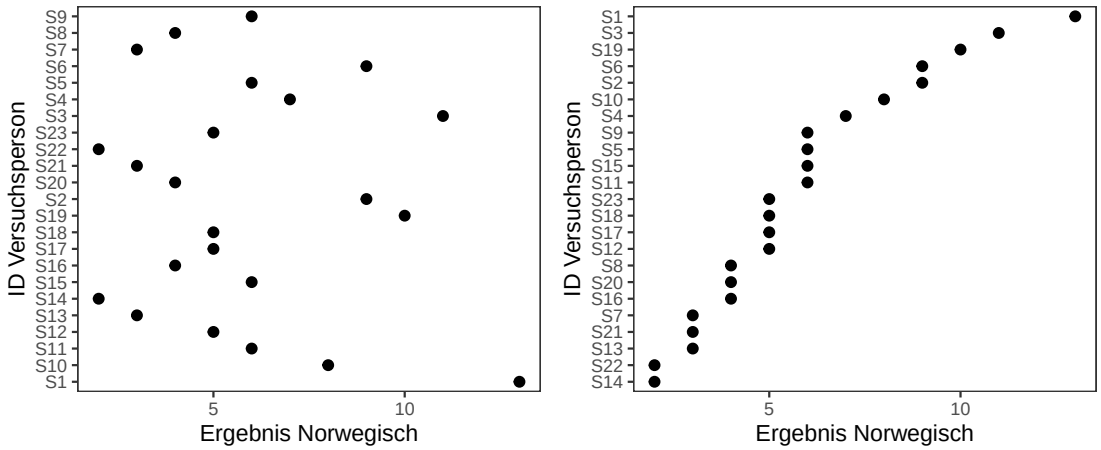


Abbildung 3.1: Ein Punktdiagramm sortiert nach den IDs der Versuchspersonen (links) und eins geordnet nach dem Ergebnis (rechts). Es gibt keine Werte, die weit von anderen liegen.

3.1 Das Punktdiagramm

Wenn wir über die Ergebnisse kommunizieren wollen und der Datensatz ganz klein ist, könnten wir ihn einfach direkt reproduzieren. Aber sogar für den relativ kleinen Datensatz hier—bloss 23 Beobachtungen—würde es sich lohnen, die Daten grafisch darzustellen und numerisch zusammenzufassen.

Eine erste grafische Darstellung ist das Punktdiagramm (*dot chart*), siehe Abbildung 3.1. Anstatt lediglich die Zahlen im Datensatz aufzulisten, werden die Beobachtungen als Punkte auf separaten Linien entlang der x-Achse dargestellt. Jede Linie wird mit einer ID entlang der y-Achse vermerkt.

Um die linke Grafik zu zeichnen, können Sie diesen Befehl verwenden. Achten Sie darauf, dass das *tidyverse*-Bündel geladen ist: Auch wenn Sie es installiert haben und in einer anderen Session verwendet haben, müssen Sie es in jeder Session erneut laden. Die Kommentare erläutern, wie die Grafik aufgebaut ist.

```
ggplot(data = d,                  # Datensatz mit den Variablen
       # aes() = aesthetics = welche Variable wie dargestellt werden soll
       aes(x = Norwegian,         # Variable auf x-Achse
           y = Participant)) +    # Variable auf y-Achse
  geom_point() +                  # Daten als Punkte darstellen
  xlab("Ergebnis Norwegisch") + # insb. bei Arbeiten/Vorträgen/Artikeln:
  ylab("ID Versuchsperson")       #     Achsen beschriften
```

Achten Sie insbesondere auf Gross- und Kleinschreibung, auf die Klammern und Kommas und auf die Pluszeichen. Mit Letzteren werden der Grafik zusätzliche Schichten hinzugefügt. Mit dem Befehl auf den ersten vier Zeilen (`ggplot(...)`) wird lediglich die 'Leinwand' der Grafik gezeichnet. Nach dem Pluszeichen folgt der Befehl

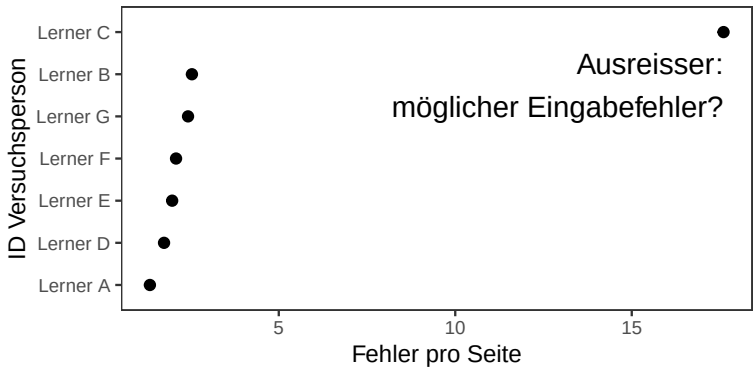


Abbildung 3.2: Beispiel von einem Ausreisser. Hier müsste man kontrollieren, ob der Datenpunkt (17.6) richtig eingetragen wurde und nicht etwa ein Tippfehler ist (statt 1.76).

`geom_point(...)`, der Punkte auf die Leinwand malt. Mit den Befehlen `xlab(...)` und `ylab(...)` werden Achsenbeschriftungen hinzugefügt bzw. überschrieben.

Um die rechte Grafik zu zeichnen, ersetzen Sie in der 4. Zeile `y = Participant` durch `y = reorder(Participant, Norwegian)` (Klammer nicht vergessen!).

In diesem Beispiel ist das Punktdiagramm insbesondere nützlich aufgrund von dem, was es eben *nicht* aufzeigt. Es scheint nämlich keine Datenpunkte, oder Grüppchen von Datenpunkten, zu geben, die ziemlich weit von den anderen entfernt liegen. Solche Datenpunkte, die man **Ausreisser** nennt, können das Ergebnis einer Analyse stark beeinflussen, sodass es wichtig ist, zu wissen, dass es sie gibt. Ausreisser *können* (nicht *müssen*) auch auf technische Fehler hinweisen und gilt es also nochmals zu kontrollieren. Abbildung 3.2 zeigt ein fiktives Beispiel, in dem die Anzahl morphologischer Fehler pro Textseite pro Lerner aufgeführt wird. Ein Datenpunkt liegt so weit von den anderen entfernt, dass man hier auf jeden Fall nochmals kontrollieren sollte, ob die Angabe tatsächlich stimmt, und nicht etwa auf einem Tippfehler bei der Dateneingabe beruht.

3.2 Das Histogramm

Eine zweite nützliche Grafik ist das Histogramm. Für ein Histogramm wird die Variable, die man darstellen möchte, in *bins* aufgeteilt und es wird gezählt, wie viele Beobachtungen es in jedem *bin* gibt. Diese Anzahlen werden in der Grafik als Bälkchen dargestellt, siehe Abbildung 3.3. Wie in diesem Beispiel sind die *bins* in den allermeisten Histogrammen gleich breit. Die Breite der *bins* muss man selber festlegen; wie die Befehle und Kommentare unten zeigen, ist dies eine Frage von Ausprobieren.

Verglichen mit dem Punktdiagramm ist ein Vorteil des Histogramms, dass auch sehr grosse Datensätze sinnvoll dargestellt werden können, während man in einem Punktdiagramm wohl schnell den Überblick verliert.

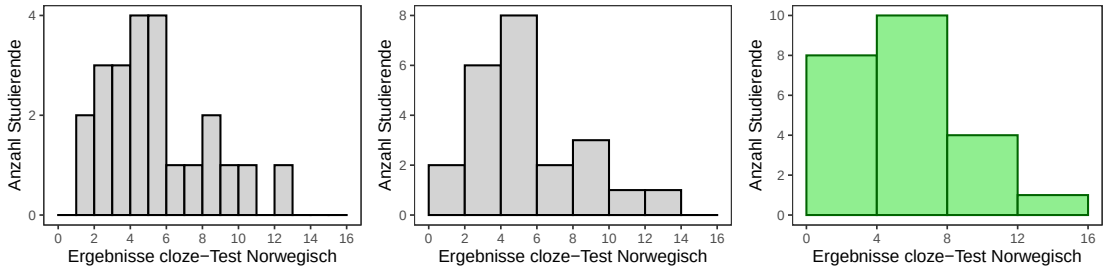


Abbildung 3.3: Drei Histogramme mit den `Norwegian`-Ergebnissen. Links: Die Grenzen zwischen den *bins* liegen bei 0, 1, 2, usw., 15, 16. Mitte: Grenzen bei 0, 2, 4, usw., 15, 16. Rechts: Grenzen bei 0, 4, 8, 12, 16. Sowohl die Grafik links als auch die in der Mitte halte ich hier für sinnvoll; die Grafik rechts ist nach meinem Geschmack ein bisschen zu grobe. Eine goldene Regel für die Wahl der Breite der *bins* gibt es nicht. Daher gilt: Mit den Einstellungen herumspielen und eine nützliche Grafik auswählen.

```
ggplot(data = d,
       aes(x = Norwegian)) +
  # Defaulteinstellungen fürs Histogramm
  geom_histogram()
```

```
ggplot(data = d,
       aes(x = Norwegian)) +
  # Anzahl 'bins' definieren
  geom_histogram(bins = 10)
```

```
ggplot(data = d,
       aes(x = Norwegian)) +
  # Binbreite definieren
  geom_histogram(binwidth = 3)
```

```
ggplot(data = d,
       aes(x = Norwegian)) +
  # Grenzen selbst festlegen, hier etwa bei 0, 4, 8, 12, 16.
  # Kürzel: seq(from = 0, to = 16, by = 4).
  # Die Farben kann man selbst auswählen.
  geom_histogram(breaks = seq(from = 0, to = 16, by = 4),
                 fill = "lightgreen",
                 colour = "darkgreen") +
  # Die x- und y-Achse muss man übrigens auch selbst einstellen.
  xlab("Ergebnisse cloze-Test Norwegisch") +
  ylab("Anzahl Studierende")
```

In den Histogrammen bisher stand die *Anzahl* Beobachtungen pro *bin* auf der y-Achse. Wenn man unterschiedliche Histogramme (z.B. von unterschiedlichen Gruppen) vergleichen möchte, kann es sinnvoller sein, diese Zahlen zu einer Art relative Frequenz

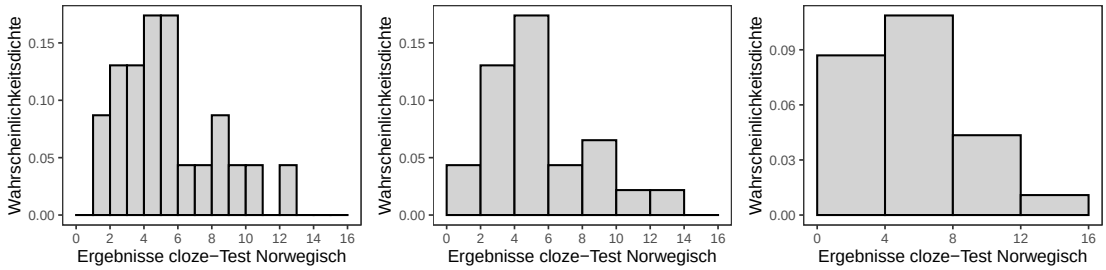


Abbildung 3.4: Drei Histogramme mit der `Norwegian`-Ergebnissen. Statt der absoluten Anzahl Beobachtungen pro *bin* stehen hier Wahrscheinlichkeitsdichten auf der y-Achse.

umzurechnen. Diese nennt man **Wahrscheinlichkeitsdichten**. Sie werden so berechnet, dass die Gesamtfläche, die das Histogramm einnimmt, 1 beträgt. Anders gesagt: Man nimmt die Höhe jedes *bins*, also die Wahrscheinlichkeitsdichte, und multipliziert diese mit seiner Breite, um die Oberfläche der Bälkchen zu berechnen. Die Höhe wird so festgelegt, dass wenn man alle Oberflächen addiert, die Summe 1 beträgt. Die Form des Histogramms ist aber gleich, egal ob man mit den Anzahlen oder den Wahrscheinlichkeitsdichten arbeitet.

Abbildung 3.4 zeigt ein Beispiel. Die Breite jedes *bins* in der rechten Grafik beträgt 4. Die Höhen sind etwa 0.09, etwa 0.105, etwa 0.045 und fast 0.015, sodass $(4 \times 0.09) + (4 \times 0.105) + (4 \times 0.045) + (4 \times 0.015) \approx 1$.

Um die Grafiken in Abbildung 3.4 selber zu zeichnen, ersetzen Sie in den Befehlen oben

```
ggplot(data = d,
       aes(x = Norwegian))
```

durch

```
ggplot(data = d,
       aes(x = Norwegian,
           y = ..density..))
```

3.3 Mittelwerte

Es ist unpraktisch, in einem Bericht alle einzelnen beobachteten Werte aufzulisten.[1] Wenn möglich probiert man diese Informationen daher zu komprimieren, indem man berichtet, welcher Wert typisch für diese Beobachtungen ist und wie sehr die einzelnen Beobachtungen von diesem typischen Wert abweichen. Die Frage nach dem typischen Wert betrifft die **zentrale Tendenz** der Beobachtungen und wird anhand von einem Mittelwert beantwortet. Die Frage nach den Abweichungen betrifft die **Streuung** der

[1] Aber es ist sehr sinnvoll, den Datensatz online verfügbar zu stellen und im Bericht auf ihn zu verweisen! Eine benutzerfreundliche Website, wo Sie dies machen können, ist `https://osf.io`. Siehe *Some advantages of sharing your data and code* (14.12.2015).

Beobachtungen und wird anhand von Streuungsmassen beantwortet. Zuerst widmen wir uns den Mittelwerten.

3.3.1 Das arithmetische Mittel

Oder einfach 'Mittel'. Wenn man vom 'Durchschnitt' oder 'Mittelwert' spricht, wird meistens das Mittel gemeint, aber eigentlich sind 'Durchschnitt' und 'Mittelwert' Hyponyme, denn es gibt ausser dem Mittel noch andere Durchschnittsmasse.

Um das Mittel zu berechnen, addiert man alle Werte und teilt man die Summe durch die Anzahl Werte. Das Populationsmittel kürzt man in Formeln meistens als μ ab; die Formel in Gross-Sigmanotation schaut dann so aus:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3.1}$$

Gross-Sigmanotation. Formeln wie diese mögen abschrecken, sind in Statistikhandbüchern jedoch gang und gäbe—und gar nicht so schwierig.

Reihen von Beobachtungen einer Variablen ('Vektoren') werden meistens mit römischen Buchstaben dargestellt. 'x' ist also die Reihe von Beobachtungen, deren Mittel wir berechnen wollen—hier also die 23 Ergebnisse beim norwegischen Lesetest.

Das kleine 'i' ist ein Index und wird verwendet um eine spezifische Beobachtung zu identifizieren. 'x_i' heisst lediglich 'die i. Beobachtung von x'. Ist i 5, dann heisst x_i eigentlich x_5, sprich die 5. Beobachtung von x.

'N' ist einfach die Anzahl Beobachtungen in der Reihe—bei uns also $N = 23$.

'$\sum$', zu guter Letzt, heisst lediglich 'Summe'. Die Werte, die wir addieren müssen, kommen nach dem Symbol. $\sum_{i=1}^{N} x_i$ heisst, dass wir die Werte in der Reihe x addieren müssen, anfangend mit dem ersten ($i = 1$) und endend beim letzten (N). $\sum_{i=3}^{5} x_i$ hiesse, dass wir nur den 3., 4. und 5. x-Wert addieren müssten.

Die Formel

$$\frac{1}{N} \sum_{i=1}^{N} x_i$$

kann also so umgeschrieben werden:

$$\frac{1}{N}(x_1 + x_2 + x_3 + \cdots + x_N)$$

In unserem Beispiel:

$$\frac{1}{23}(13+9+11+\cdots+5)$$

Oh ja: Multiplizieren mit $\frac{1}{23}$ ist dasselbe wie durch 23 teilen.

Das Mittel der Norwegischdaten können wir folgendermassen berechnen:

```
# Summe aller Norwegian-Werte
sum(d$Norwegian)

## [1] 136

# Anzahl Norwegian-Werte
length(d$Norwegian)

## [1] 23

# Summe geteilt durch Anzahl
136/23

## [1] 5.913

# Oder in einer Zeile
sum(d$Norwegian) / length(d$Norwegian)

## [1] 5.913
```

Einfacher geht es mit der `mean()`-Funktion:

```
mean(d$Norwegian)
```

Die `tidyverse`-Lösung brauchen wir hier im Prinzip nicht, aber sie zeigt, wie die `summarise()`-Funktion funktioniert:

```
d %>%
  summarise(Mittel_Norwegisch = mean(Norwegian))

## # A tibble: 1 x 1
##   Mittel_Norwegisch
##               <dbl>
## 1              5.91
```

Das Mittel hat ein paar nützliche mathematische Eigenschaften, denen es seine Omnipräsenz verdankt. (Eine betrifft den zentralen Grenzwertsatz; siehe Abschnitt 5.3.) Ein wesentlicher Nachteil des Mittels ist aber, dass er stark von Ausreissern beeinflusst wird. Nimmt man zum Beispiel die Daten aus Abbildung 3.2 und berechnet man ihr Mittel, dann ist das Ergebnis 4.25—ein ziemlich untypischer Wert für die Beobachtungen, sind doch 6 der 7 Beobachtungen unter 2.6 und 1 über 17.5. Lässt man den Ausreisser weg, ist das Mittel 2.02, was der Tendenz des Hauptanteils der Beobachtungen besser entspricht.

3.3.2 Der Median

Ein anderer beliebter Mittelwert ist der Median. Es handelt sich hier buchstäblich um den Wert in der Mitte: Zum Berechnen des Medians ordnet man die Daten von klein nach gross und nimmt den mittleren Wert. Gibt es eine gerade Anzahl Beobachtungen, gibt es zwei mittlere Werte. In solchen Fällen ist der Median das Mittel beider mittlerer Werte.

Zuerst die komplizierte Berechnungsmethode, um den Vorgang zu illustrieren: Mit `arrange()` die Daten von klein nach gross ordnen und dann den mittleren Wert nehmen. Da es 23 Beobachtungen gibt, ist der 12. Wert der mittlere (es gibt 11 kleinere und 11 grössere):

```
d %>%
  select(Norwegian) %>%
  arrange(Norwegian) %>%
  slice(12)
```

```
## # A tibble: 1 x 1
##   Norwegian
##       <dbl>
## 1         5
```

Oder kürzer mit `median()`:

```
median(d$Norwegian)
```

```
## [1] 5
```

Mit der `summarise()`-Funktion können wir eine Zusammenfassungstabelle mit mehreren Werten aufstellen:

```
d %>%
  summarise(Mittel_Norwegisch = mean(Norwegian),
            Median_Norwegisch = median(Norwegian))
```

```
## # A tibble: 1 x 2
##   Mittel_Norwegisch Median_Norwegisch
##               <dbl>             <dbl>
## 1              5.91                 5
```

Der Median ist weniger ausreisserempfindlich als das Mittel. Berechnet man für die Werte aus Abbildung 3.2 den Median mit dem Ausreisser, ist das Ergebnis 2.09; ohne ist es 2.04.

Grosse Unterschiede zwischen dem Mittel und dem Median sind in der Regel Ausreissern oder asymmetrischen Verteilungen (siehe Abschnitt 3.6) zuzuschreiben. So oder so gilt: **Keine Mittelwerte berechnen, ohne die Daten zuerst grafisch darzustellen!** Die Berechnung mag stimmen, aber unter Umständen ist sie nicht *sinnvoll*.

3.3.3 Der Modus

Den Modus trifft man weniger oft an, ist aber eine ganz einfache Art und Weise, um den 'typischen Wert' zu definieren: Es handelt sich schlicht und einfach um den Wert, der am häufigsten vorkommt. Eine Modusfunktion gibt es nicht, aber wir können für jeden Wert zählen, wie oft er vorkommt:

```
d %>%
  count(Norwegian)
```

```
## # A tibble: 11 x 2
##    Norwegian     n
##        <dbl> <int>
##  1         2     2
##  2         3     3
##  3         4     3
##  4         5     4
##  5         6     4
##  6         7     1
##  7         8     1
##  8         9     2
##  9        10     1
## 10        11     1
## 11        13     1
```

Zwei Werte kommen am häufigsten vor: 5 und 6 kommen beide 4 Mal vor. Die Modi der Norwegischdaten sind also 5 und 6.

```
d %>%
  count(Norwegian) %>%
  filter(n == max(n))
```

```
## # A tibble: 2 x 2
##   Norwegian     n
##       <dbl> <int>
## 1         5     4
## 2         6     4
```

Ein wesentlicher Nachteil des Modus ist, dass bei feinkörnigen Daten jede Beobachtung eh nur ein oder zwei Mal vorkommt, sodass es nicht sinnvoll ist, ihn zu berechnen.

3.3.4 Andere Mittelwerte

Diese Mittelwerte trifft man seltener an, aber sie seien hier der Vollständigkeit halber erwähnt.

Das getrimmte Mittel Man löscht die x% kleinsten und die x% grössten Werte und berechnet dann das Mittel. Fürs 25% getrimmte Mittel löscht man also zwei Mal ein Viertel der Daten: die 25% niedrigsten Beobachtungen und die 25% höchsten Beobachtungen. Das getrimmte Mittel wird verwendet, um den Einfluss von extremen Beobachtungen auf das Ergebnis zu reduzieren.

```
# Norwegischdaten sortiert von klein nach gross
norwegisch_sortiert <- sort(d$Norwegian)
norwegisch_sortiert
```

```
##  [1]  2  2  3  3  3  4  4  4  5  5  5  5  6  6  6  6  7  8
## [19]  9  9 10 11 13
```

```
# 25% getrimmte Mittel:
mean(norwegisch_sortiert, trim = 0.25)
```

```
## [1] 5.4615
```

```
# 1/4 von 23 ist 5.75; diese Zahl
# wird nach unten abgeründet, also
# werden die 5 kleinsten und 5 höchsten Werte gelöscht:
mean(norwegisch_sortiert[6:18])
```

```
## [1] 5.4615
```

Das winsorisierte Mittel Die x% kleinsten und die x% grössten Werte werden ersetzt und zwar durch den kleinsten bzw. grössten Wert, der nicht ersetzt wurde. Danach berechnet man das Mittel. Wenn man zum Beispiel 100 Beobachtungen hat und auf jeder Seite 10% der Beobachtungen winsorisiert, ersetzt man die 10 niedrigsten Beobachtungen alle durch den 11. niedrigsten Wert und die 10 höchsten Beobachtungen durch den 90. Wert.[2] Das winsorisierte Mittel wird ebenfalls verwendet, um den Einfluss von extremen Beobachtungen auf das Ergebnis zu reduzieren.

Es gibt noch andere Mittelwerte—etwa das geometrische und das harmonische Mittel. Diese werden hier nicht behandelt. Auch das gewichtete Mittel wird nicht an dieser Stelle behandelt.

[2] In diesem Beispiel wird durch die Winsorisierung eine Ganzzahl der Datenmenge ersetzt (also 10 Beobachtungen auf jeder Seite). Die genaue Berechnung ist schwieriger, wenn durch die Winsorisierung eigentlich eine Bruchzahl der Beobachtungen ersetzt werden soll. Zum Beispiel müsste man bei einer Datenreihe von 23 Beobachtungen 2.3 Beobachtungen winsorisieren, wenn man 10%-Winsorisierung durchführt. In solchen Fällen wird nicht der kleinste bzw. höchste nicht-ausgeschlossene Wert verwendet, sondern eine Intrapolation zwischen den ausgeschlossenen und nicht-ausgeschlossenen Werten. Siehe die Funktion `winsor.mean()` im `psych`-Paket.

3.4 Streuungsmasse

Wenn man nur einen oder ein paar Mittelwerte berichtet, bleibt die Frage unbeantwortet, wie stark die einzelnen Beobachtungen davon abweichen. Mit Streuungmassen versucht man diese Abweichnung numerisch auszudrücken.

3.4.1 Spannweite

Ein einfaches Streuungsmass ist die Spannweite. Man berechnet lediglich den niedrigsten und den höchsten Wert und berichtet diese oder den Unterschied zwischen ihnen:

```
min(d$Norwegian)

## [1] 2

max(d$Norwegian)

## [1] 13

range(d$Norwegian)

## [1]  2 13
```

Dieses Mass wird aus gutem Grund selten verwendet: Es ist extrem ausreisserempfindlich und basiert es auf nur zwei der Beobachtungen. Ausserdem unterschätzt die Spannweite einer Stichprobe systematisch die Spannweite der Population, aus der sie stammt. (Mit Stichproben beschäftigen wir uns in späteren Kapiteln.)

3.4.2 Summe der Quadrate

Wenn wir alle Beobachtungen ins Streuungsmass einfliessen lassen wollen, scheint es auf den ersten Blick sinnvoll, die Unterschiede zwischen den beobachteten Werten und dem Mittel zu berechnen und diese Unterschiede beieinander aufzuzählen: $(x_1 - \mu) + (x_2 - \mu) + \ldots$. Diese Summe ist aber immer 0:[3]

```
sum(d$Norwegian - mean(d$Norwegian))

## [1] 8.8818e-15
```

Gleitkommazahlen. Das Ergebnis der obigen Berechnung wird wiedergegeben als 8.8818e-15. Eigentlich heisst dies: 88818 mit 15 vorangestellten Nullen, also

[3] Beweis:

$$(x_1 - \mu) + (x_2 - \mu) + \cdots + (x_N - \mu) = (x_1 + x_2 + \cdots + x_N) - N\mu$$

μ wird berechnet als $\frac{1}{N}(x_1 + x_2 + \cdots + x_N)$, sodass $N\mu = (x_1 + x_2 + \cdots + x_N)$. Daher gilt:

$$(x_1 + x_2 + \cdots + x_N) - N\mu = (x_1 + x_2 + \cdots + x_N) - (x_1 + x_2 + \cdots + x_N) = 0.$$

0.0000000000000088818. Aber eigentlich sollte das Ergebnis *genau* 0 sein, nicht *fast* 0. Das Problem liegt bei der Genauigkeit, mit der ein Computer mit Zahlen umgeht. Wir werden auf dieses Problem hier nicht näher eingehen, da es für uns nicht von praktischer Bedeutung ist.

Um dieses Problem zu lösen, werden die Unterschiede zwischen den beobachteten Werten und dem Mittel quadriert, bevor sie beieinander aufgezählt werden. Dadurch werden sie alle positiv, sodass ihre Summe nicht länger 0 ist. Diese Summe der Quadrate wird in Formeln als d^2 oder SS (sum of squares) abgekürzt:

$$d^2 = \sum_{i=1}^{N} (x_i - \mu)^2 \tag{3.2}$$

```
sum((d$Norwegian - mean(d$Norwegian))^2)
```

```
## [1] 187.83
```

Achten Sie auf die Stelle der Klammern und der Quadrierung: Die Unterschiede müssen quadriert werden, nicht die Summe oder das Mittel.

```
# Falsch: Hier wird die Summe quadriert.
sum((d$Norwegian - mean(d$Norwegian)))^2
```

```
## [1] 7.8886e-29
```

```
# Falsch: Hier wird das Mittel quadriert.
sum((d$Norwegian - mean(d$Norwegian)^2))
```

```
## [1] -668.17
```

Vielleicht fragen Sie sich, wieso man hier mit quadrierten Unterschieden arbeitet. Wäre es nicht einfacher, mit den *absoluten* Unterschieden zu rechnen? Streuungsmasse, die auf den absoluten Unterschieden basieren, gibt es tatsächlich (*mean absolute deviation* und *median absolute deviation*), aber das Arbeiten mit quadrierten Unterschieden bietet mathematische Vorteile (z.B. zentralen Grenzwertsatz).

3.4.3 Varianz

Ein Problem mit d^2 ist, dass Datensätze unterschiedlicher Grösse nicht vergleichbar sind: Je mehr Beobachtungen es gibt, desto grösser ist d^2. d^2 drückt also sowohl die Grösse des Datensatzes als auch die Streuung der Beobachtungen aus, was unerwünscht ist. Die Lösung liegt auf der Hand: d^2 teilen durch die Anzahl Beobachtungen. Dies ergibt die Populationsvarianz (σ^2):

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{3.3}$$

```
sum((d$Norwegian - mean(d$Norwegian))^2) / length(d$Norwegian)
```

```
## [1] 8.1664
```

Populations- vs. Stichprobenvarianz. In der Regel müssen wir die Varianz einer Stichprobe, *nicht* jene einer Population berechnen. Diese wird leicht anders berechnet; siehe Kapitel 5.

Eigene Funktionen schreiben. Da wir uns meistens für die Varianz einer Stichprobe, nicht für jene einer Population interessieren, gibt es in R keine Funktion, um die Populationsvarianz zu berechnen. Wenn das Eintippen des obigen Befehls zu mühsam ist, z.B., weil Sie es immer wieder verwenden müssen, empfiehlt es sich, eine eigene Funktion zu schreiben. Diese könnte so aussehen:

```
pop_var <- function(x) {
  # Populationsvarianz berechnen
  sigma2 <- mean((x - mean(x))^2)

  # Ergebnis ausspucken
  sigma2
}
```

Wenn Sie diese Funktion übernehmen, können Sie die Populationsvarianz der Norwegischdaten einfach folgendermassen berechnen:

```
pop_var(d$Norwegian)
```

```
## [1] 8.1664
```

3.4.4 Standardabweichung

Varianzen sind nicht einfach zu interpretieren, da sie, aufgrund der Quadrierung in der Berechnung, in quadrierten Einheiten ausgedrückt werden (z.B. quadrierte Testergebnisse, quadrierte Sprecher per Sprache). Wir können aber die Wurzel der Varianz nehmen, was die Populationsstandardabweichung ergibt (σ):

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.4}$$

In R, anhand der selbst geschriebenen `pop_var()`-Funktion:

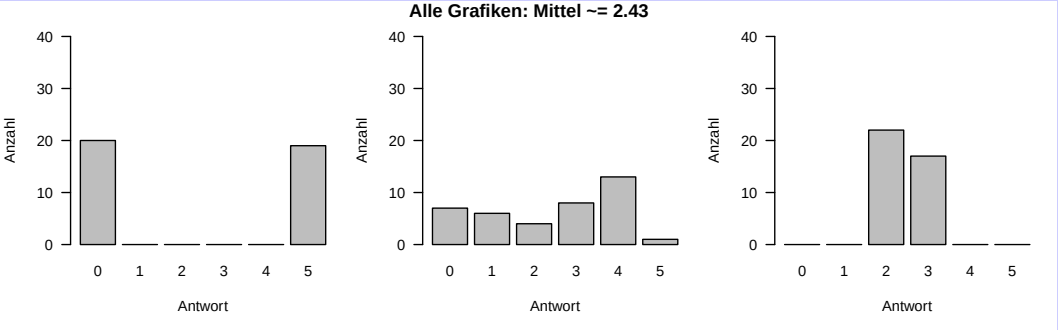
```
pop_var(d$Norwegian) %>%
  sqrt()
```

```
## [1] 2.8577
```

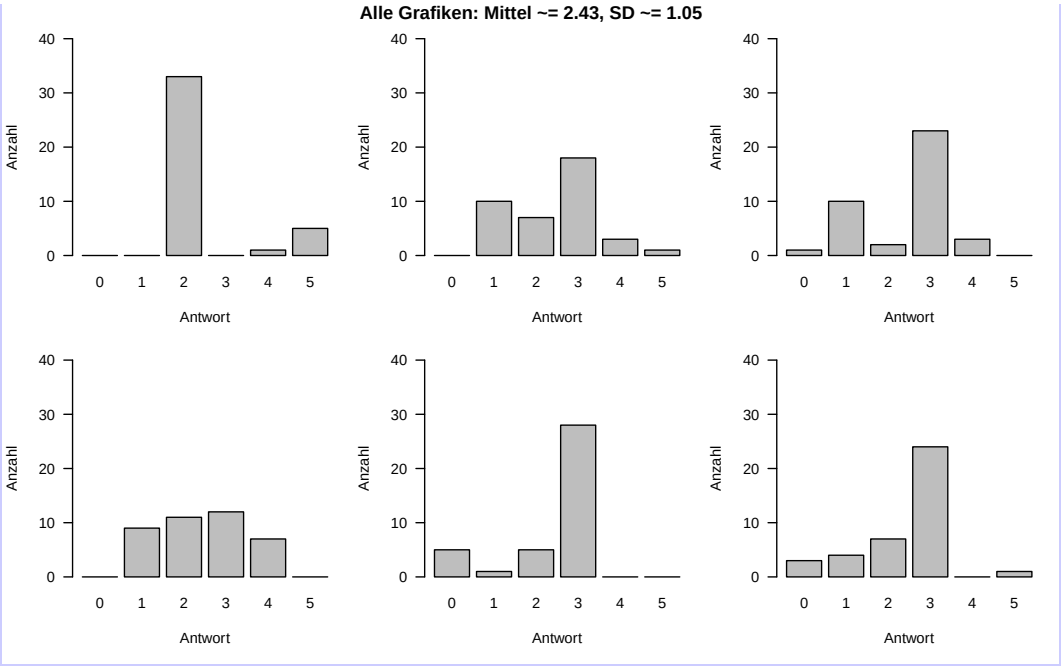
Standardabweichungen und Varianzen kann man nicht absolut interpretieren: Eine Standardabweichung von 0.4 ist je nach der Art von Daten klein, gross oder unauffällig, und dies gilt auch für Standardabweichungen von 8'000. So wäre etwa eine Standardabweichung von 30 unauffällig, wenn es sich um in Zentimetern gemessenen Körpergrössen von Menschen handelte; erstaunlich klein, wenn die Körpergrössen in Millimetern ausgedrückt wären; und erstaunlich gross, wenn sie in Zoll ausgedrückt wären.

Populations- vs. Stichprobenstandardabweichung. In der Regel müssen wir die Standardabweichung einer Stichprobe, *nicht* jene einer Population berechnen. Diese wird leicht anders berechnet; siehe Kapitel 5.

Daten nicht nur numerisch zusammenfassen. Stellen Sie sich vor, dass Sie in einer Studie lesen, dass 39 Versuchspersonen eine Frage auf einer 6er-Skala von 0 bis 5 beantwortet haben und das Mittel der Antworten 2.43 betrug. Vielleicht stellen Sie sich dann darunter vor, dass die meisten Versuchspersonen sich für '2' oder '3' entschieden. Dies muss aber nicht der Fall sein: Hinter diesem Mittelwert können sich viele andere Datenmuster verstecken, die zu anderen Schlussfolgerungen führen sollten. Vielleicht sind sich die Versuchspersonen einig in ihrer Gleichgültigkeit, weshalb sie alle Antworten in der Mitte der Skala wählen. Oder vielleicht handelt es sich um ein sehr kontroverses Thema mit überzeugten Gegnern und Befürwortern aber ohne eine moderate Mitte. Oder vielleicht sind alle Arten von Meinung etwas vertreten:



Wenn ein Streuuungsmass berichtet wird, schränkt sich Anzahl möglicher Muster zwar ein, aber trotzdem können sich hinter einem Mittel und einer Standardabweichung mehrere Verteilungen verstecken:[a]



[a]Diese Verteilungen wurden generiert anhand des R-Codes unter `http://bayesfactor.blogspot.ch/2016/03/how-to-check-likert-scale-summaries-for.html`. Für ähnlichen Ansätze, siehe Heathers et al. (2018) und Wilner et al. (2018).

3.5 Kerndichteschätzungen

In unserem Norwegischbeispiel gibt es es nur eine geringe Anzahl Beobachtungen und ausserdem ist die Variable nicht sehr feinkörnig. Eine sehr feinkörnige Variable wäre eine Variable mit sehr vielen möglichen Ergebnissen und höchstens einem Beleg pro möglichen Wert.

Was würde aber passieren, wenn wir eine grosse Anzahl Beobachtungen (z.B. 100'000) von einer sehr feinkörnigen Variablen erheben würden, diese Beobachtungen in einem Histogramm mit Wahrscheinlichkeitsdichten (siehe Abschnitt 3.2) darstellen würden und die Anzahl *bins* immer vergrössern würden? Wenn die Anzahl *bins* zu gross wird, können wir sie nicht mehr voneinander unterscheiden, wie Abbildung 3.5 zeigt. Ausserdem werden wir irgendwann nur noch *bins*, die entweder eine oder keine einzige Beobachtung beinhalten, haben.

In solchen Fällen arbeitet man stattdessen mit Kerndichtschätzungen. Die Berechnungsmethode braucht uns hier nicht zu interessieren; grundsätzlich handelt es sich um ein geglättetes Histogramm, bei dem die Wahrscheinlichkeitsdichten jedes Bälkchens mit einer Kurve verbunden werden und die Bälkchen selber nicht mehr dargestellt werden; siehe Abbildung 3.6.

Wahrscheinlichkeitsdichte ist nicht gleich Wahrscheinlichkeit! In Abbildung 3.6 ist die Wahrscheinlichkeit, dass ein Wert von 10 beobachtet wird, nicht etwa 13%,

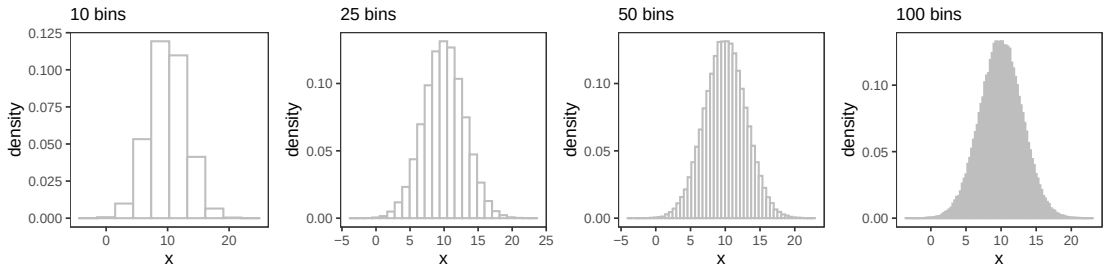


Abbildung 3.5: Vier Histogramme der gleichen feinkörnigen Variablen.

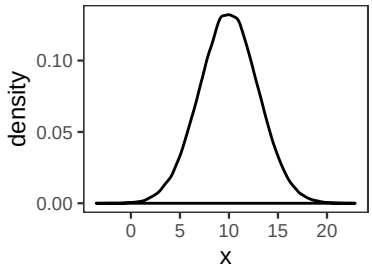


Abbildung 3.6: Eine Kerndichteschätzung der gleichen Variablen.

sondern verschwindend gering. Wenn man bloss genügend Dezimalstellen in Betracht zieht (z.B. 10,00000001 oder 9,9999999999), ist jeder einzelne Wert ja verschwindend unwahrscheinlich. Wir können deswegen keine sinnvollen Wahrscheinlichkeitsaussagen über spezifische Werte machen, sondern nur über Intervalle. Dies machen wir in den nächsten Kapiteln.

Eine Kerndichteschätzung können Sie mit dem Befehl `geom_density()` zeichnen. Siehe hierzu die Beispiele unter `http://ggplot2.tidyverse.org/reference/geom_density.html`.

3.6 Klassische (idealisierte) Verteilungen

Es lassen sich ein paar klassische Arten von Datenverteilungen unterscheiden. In ihrer reinen Form trifft man diese Verteilungen zwar selten an, aber viele Datenverteilungen können als Annäherungen dieser Idealisierungen betrachtet werden.

3.6.1 Gleichverteilung

Oder Uniformverteilung. In einer Gleichverteilung ist jeder mögliche Wert gleich wahrscheinlich. Ein typisches Beispiel ist das Würfeln eines fairen Würfels ('diskrete Gleichverteilung'; diskret, da es eine beschränkte Anzahl möglicher Ergebnisse gibt). Die Wahrscheinlichkeit, eine 6 zu würfeln, ist gleich gross wie jene, eine 1. usw. zu würfeln.

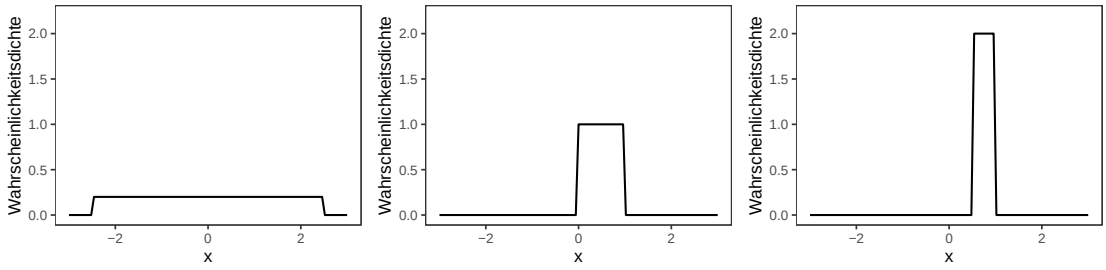


Abbildung 3.7: Wahrscheinlichkeitsdichten von drei kontinuierlichen Gleichverteilungen.

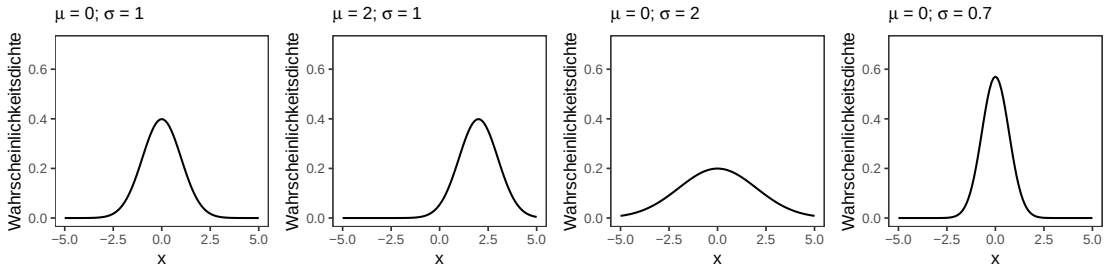


Abbildung 3.8: Die Form einer Normalverteilung ist von zwei Parametern abhängig: ihrem Mittel (μ) und ihrer Standardabweichung (σ).

Wenn die möglichen Ergebnisse feinkörniger sind, spricht man von einer 'kontinuierlichen Gleichverteilung'.

Aufgabe Abbildung 3.7 zeigt drei kontinuierliche Gleichverteilungen mit Bereichen [-2.5, 2.5], [0, 1] und [0.5, 1]. Erklären Sie, warum die Wahrscheinlichkeitsdichte höher als 1 sein kann. (Tipp: Berechnen Sie die Flächen unter den Kurven.)

3.6.2 Normalverteilung

Die Normalverteilung ist die typische 'Glockenkurve'. Ihre Wahrscheinlichkeitsdichte wird durch eine kompliziert aussehende Gleichung definiert, die für unsere Zwecke nicht so wichtig ist. Wichtig ist nur, dass die Form der Glockenkurve von zwei Faktoren bestimmt wird: dem Mittel der Datenverteilung (μ) und ihrer Standardabweichung (σ). μ bestimmt, um welchen Wert die Kurve zentriert ist; σ wie 'breit' und 'hoch' die Kurve ist. Siehe Abbildung 3.8.

Gleichverteilungen und Normalverteilungen sind Beispiele von **symmetrischen Verteilungen**: die linke Hälfte der Verteilung formt das Spiegelbild der rechten Hälfte. Bei Variablen, die symmetrisch verteilt sind, sind das Mittel und der Median einander (ungefähr) gleich.

Normalität überprüfen und Datentransformationen. Bei einer Normalverteilung sind Modus, Mittel und Median gleich, d.h., es gibt eine eindeutige zentrale Ten-

denz. Mit vielen statistischen Verfahren kann man Aussagen über das Mittel einer Population oder Stichprobe machen. Wenn Mittel, Median und Modus alle (mehr oder weniger) gleich sind – wie bei Normalverteilungen –, kann man mit diesen Verfahren die zentrale Tendenz also völlig erfassen. Wenn die Daten stark von einer Normalverteilung abweichen, können die Aussagen, die solche Verfahren übers Mittel machen, zwar noch stimmen. Sie sind aber eben weniger relevant fürs Erfassen der zentralen Tendenz: Das Mittel ist ja bloss ein Versuch, die zentrale Tendenz zu erfassen. Manchmal sind Daten zwar nicht-normalverteilt, können aber einfach zu annähernd normalverteilten Daten **transformiert** werden. Solche Datentransformationen werden in diesem Skript nur oberflächlich behandelt; siehe aber Gelman & Hill (2007, S. 59–68) und Baayen & Milin (2010).

Wie wir in den nächsten Kapiteln sehen werden, ist die Normalverteilung auch aus anderen Gründen in der Statistik von zentraler Bedeutung: Eine Reihe mathematischer Sätze, die die Analyse vereinfachen, gilt nur nachweisbar, wenn die Daten aus einer Normalverteilung stammen.

3.6.3 Bimodale Verteilung

Eine bimodale Verteilung ist eine Verteilung mit zwei 'Höckern'. Bei einer Befragung zu einem gesellschaftlichen Thema etwa würde eine solche Verteilung darauf hindeuten, dass die Bevölkerung stark zwischen Befürwortern und Gegnern polarisiert ist und dass relativ wenige Leute eine Zwischenposition vertreten.

Eine bimodale Verteilung kann auch darauf hindeuten, dass eigentlich zwei Populationen statt nur einer gemessen wurden. Zum Beispiel ist (in der akustischen Phonetik) die Verteilung der Grundfrequenz in der ganzen Population bimodal verteilt: Männerstimmen haben eine tiefere Grundfrequenz als Frauenstimmen, aber innerhalb jeder Gruppe sind die Werte ungefähr normalverteilt.

Manchmal trifft man auch **multimodale** Verteilungen, also Verteilungen mit mehreren Höckern, an.

Bi- und multimodale Verteilungen können zwar symmetrisch sein, und folglich können das Mittel und der Median einander recht ähnlich sein. Trotzdem zeigen diese Mittelwerte nicht, welche Werte typisch für solche Verteilungen sind. Wenn eine bimodale Verteilung in separate Populationen zerteilt werden kann (z.B. Männer und Frauen), ist es sinnvoller, die Mittelwerte innerhalb jeder Population zu berechnen. Wenn dies unmöglich ist, dürfte es besser sein, die Verteilung grafisch zu berichten (immer eine gute Idee!) oder sie in Vollsätzen zu beschreiben anstatt einen sinnlosen Mittelwert zu berechnen.

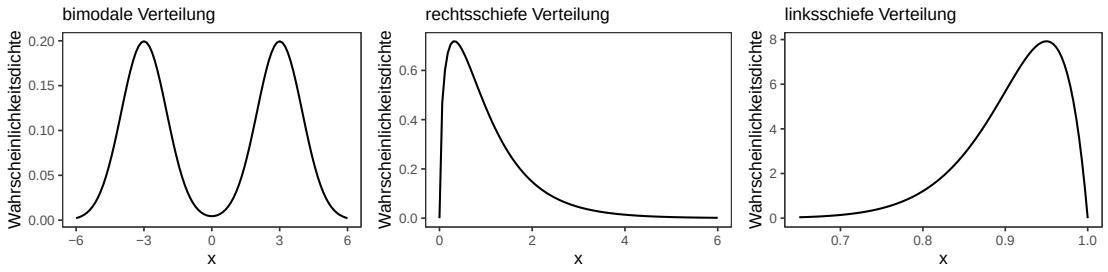


Abbildung 3.9: Eine bimodale und zwei schiefe Verteilungen.

3.6.4 Schiefe Verteilungen

Eine **rechtsschiefe Verteilung** (oder: Verteilung mit positiver Schiefe) ist eine nicht-symmetrische Verteilung, die nach rechts neigt. Etwa Reaktionszeiten, Wortfrequenzen und die Anzahl tip-of-the-tongue-Probleme pro Aufnahme sind oft rechtsschief verteilt: Die meisten Reaktionszeiten sind niedrig, aber einige werden hoch sein; die allermeisten Wörter kommen selten vor, aber eine Handvoll Wörter sehr häufig; in den meisten Aufnahmen wird es keine tip-of-the-tongue-Probleme geben, aber in ein paar schon einige.

Eine **linksschiefe Verteilung** (oder: Verteilung mit negativer Schiefe) ist nicht-symmetrisch und neigt nach links. Bei Testergebnissen könnte dies darauf hindeuten, dass der Test zu einfach war (**Deckeneffekt**): Personen mit dem gleichen hohen Testergebnis unterscheiden sich vermutlich noch voneinander in ihrer Fähigkeit, aber dies zeigt sich aufgrund des zu einfachen Tests nicht. Zu schwierige Tests führen hingegen zu rechtsschiefen Verteilungen (**Bodeneffekt**).

Bei schiefen Verteilungen können Mittel und Median weit auseinander liegen und es ist durchaus möglich, dass keiner der beiden Werte den typischen Wert der Verteilung wirklich erfasst.

Abbildung 3.9 zeigt eine bimodale, eine rechtsschiefe und eine linksschiefe Verteilung.

3.7 Weitere Literatur

Huff (1954) (*How to lie with statistics*) ist ein kurzes und sehr lesbares Büchlein. Es behandelt unter anderem die unterschiedlichen Mittelwerte und wie diese manipulativ eingesetzt werden. Johnson (2013) bietet eine Übersicht über weitere Möglichkeiten, Daten grafisch und numerisch zu beschreiben.

In diesem Skript werden zwar mehrere nützliche Arten von Grafiken vorgestellt, aber eine ausführlichere Behandlung finden Sie bei Healy (2019, ebenfalls mit `ggplot2`).

3.8 Übungen

1. Einkommensniveaus nach Land, Region oder Gemeinde werden üblicherweise in Medianen statt in Mitteln ausgedrückt. Warum?
2. 80 willkürlich ausgewählte Schweizer Staatsbürger werden gebeten, auf einer 10er-Skala anzudeuten, inwieweit sie mit der Aussage *Privater Waffenbesitz sollte verboten werden* einverstanden sind (1 = gar nicht einverstanden; 10 = völlig einverstanden). Würde diese Befragung annähernd normalverteilte Daten liefern? Wenn nicht, welcher Datenverteilung würden sie am ehesten entsprechen?
3. M&Ms können sechs Farben haben: blau, braun, gelb, grün, orange und rot. Wie schätzen Sie die relativen Frequenzen dieser Farben ein? Gibt es z.B. Ihrer Erfahrung nach eine ähnlich grosse Anzahl blaue wie rote M&Ms? Entspricht diese Verteilung einer der Verteilungen, die wir oben kennengelernt haben?
4. Die Datei `stocker2017.csv` enthält einen Teil der Daten aus einer on-line-Studie von Stocker (2017). 160 Versuchspersonen wurden gebeten, die Glaubwürdigkeit von Aussagen von SprecherInnen mit unterschiedlichen Akzenten (Englisch, Französisch, Deutsch und Italienisch) mithilfe eines *sliders* auf einer Skala von 0 bis 100 zu bewerten. Diese Daten stehen in der `score`-Spalte.
   (a) Lesen Sie diese Datei in R ein. Kontrollieren Sie, ob dies geklappt hat.
   (b) Berechnen Sie das Mittel und den Median der `score`-Daten. Sind sich diese Mittelwerte ähnlich?
   (c) Stellen Sie die `score`-Daten in einem Histogramm mit 10 *bins* dar. Welcher klassischen Verteilung entspricht diese am ehesten?
   (d) Zeichnen Sie ein Histogramm mit 100 bins. Beschreiben Sie dieses Histogramm. Sind das Mittel und der Median repräsentativ für diese Daten?
   (e) Welcher Wert ist der dritthäufigste? Warum, denken Sie?
   (f) Was ist bei den viert-, fünft-, sechst- usw. -häufigsten Werten auffällig?

Kapitel 4

Wahrscheinlichkeitsaussagen über zufällige Beobachtungen

Dieses Kapitel dient als Auffrischung der Wahrscheinlichkeitsrechnung. Konkret besprechen wir, wie wir Wahrscheinlichkeitsaussagen über **Zufallsvariablen** machen können, wenn wir schon wissen, aus welcher Verteilung diese Variable stammt. Was Zufallsvariablen sind, wird aus den Beispielen klar. Die Fähigkeit, Wahrscheinlichkeitsaussagen über Zufallsvariablen zu machen, ist an sich schon praktisch, aber zudem muss man die hinterliegende Logik kennen, wenn man Inferenzstatistik verstehen will.

4.1 Beispiel: kontinuierliche Gleichverteilung

Die Kreislinie eines Rads ist wie in Abbildung 4.1 mit Zahlen von 0 bis 360 vermerkt. Jedes Mal, wenn der Pfeil gedreht wird, bleibt er an einer zufälligen Stelle auf der Kreislinie stehen. Dies entspricht einer kontinuierlichen Gleichverteilung mit einem Bereich von 0 bis 360; siehe Abbildung 4.2. Da die Verteilung von 0 bis 360 geht und die Fläche zwischen der Wahrscheinlichkeitsdichte und der x-Achse 1 betragen muss, ist die Wahrscheinlichkeitsdichte überall $\frac{1}{360} \approx 0.0028$ (denn $(360 - 0) \times \frac{1}{360} = 1$).

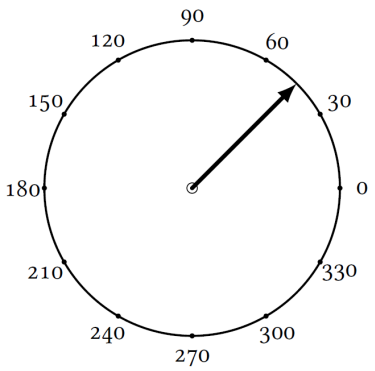


Abbildung 4.1

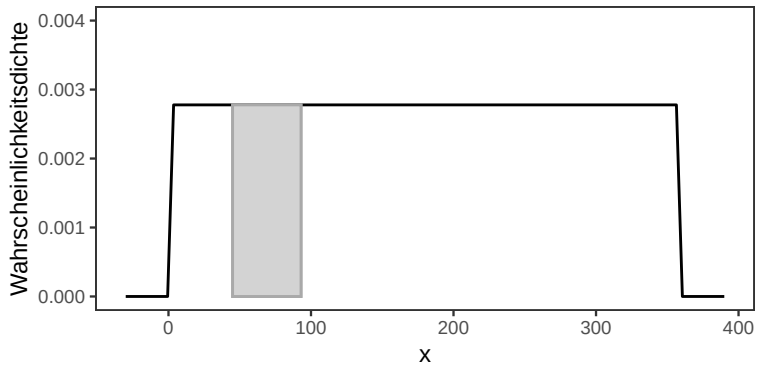


Abbildung 4.2: Wahrscheinlichkeitsdichte einer kontinuierlichen Gleichverteilung mit Bereich 0 bis 360.

4.1.1 Wahrscheinlichkeit = Fläche unter der Wahrscheinlichkeitsdichte

Wie wahrscheinlich ist es, dass wir den Pfeil drehen und er irgendwo zwischen 45 und 93 stehen bleibt? Zwischen den Werten 45 und 93 liegt etwa 13.3% der ganzen Wahrscheinlichkeitsverteilung: $93 - 45 = 48$ und $\frac{48}{360} = 0.133$. Die Wahrscheinlichkeit liegt also bei 13.3%.

Diese Berechnungsmethode lässt sich aber nur bei Gleichverteilungen anwenden – also bei Verteilungen, bei denen jeder Wert genauso wahrscheinlich ist. Eine Methode, die auch für andere Verteilungen gilt, besteht darin, **die Fläche unter der Wahrscheinlichkeitsdichte zwischen den beiden Werten** – das 'Integral' aus dem Gymnasium – zu berechnen. Diese Fläche wurde in der obigen Grafik grau eingefärbt. Bei einer Gleichverteilung ist dies ein Rechteck, dessen Fläche wir einfach berechnen können: Breite $\times$ Höhe $= (93 - 45) \times \frac{1}{360} = 0.133$.

4.1.2 Kumulative Verteilungsfunktion

Abbildung 4.3 zeigt, wie wahrscheinlich es ist, einen Wert kleiner als x zu beobachten. Diese Grafik nennt man eine **kumulative Verteilungsfunktion**; die kumulative Wahrscheinlichkeit variiert von 0 bis 1.

Mit `punif()` können wir die Wahrscheinlichkeit berechnen, dass wir einen Wert zwischen 45 und 93 beobachten. (*p* für *probability*, *unif* für *uniform distribution*.) Zuerst berechnen wir die Wahrscheinlichkeit, dass wir einen Wert kleiner als 93 beobachten. Diese Wahrscheinlichkeit entspricht dem Wert auf der y-Achse für die rote Linie in Abbildung 4.3 (Handgelenk mal Pi: etwa 25%). Mit `punif()` berechnen wir den genauen Wert:

```
# Wahrscheinlichkeit, dass man einen Wert niedriger als 93 antrifft,
# wenn die Verteilung eine Gleichverteilung zwischen 0 und 360 ist.
punif(93, min = 0, max = 360)
```

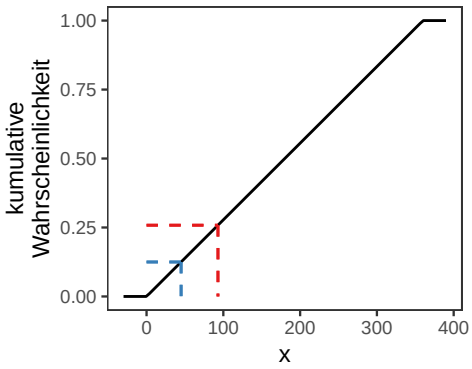


Abbildung 4.3: Kumulative Verteilungsfunktion einer kontinuierlichen Gleichverteilung mit Bereich 0 bis 360.

```
## [1] 0.25833
```

Ebenso können wir die Wahrscheinlichkeit berechnen, dass wir einen Wert kleiner als 45 beobachten (blau):

```
punif(45, min = 0, max = 360)
```

```
## [1] 0.125
```

Der Unterschied ist die Wahrscheinlichkeit, dass wir einen Wert zwischen 45 und 93 beobachten:

```
0.2583 - 0.125
```

```
## [1] 0.1333
```

```
# oder direkt:
punif(93, min = 0, max = 360) - punif(45, min = 0, max = 360)
```

```
## [1] 0.13333
```

4.2 Beispiel Normalverteilung

IQ-Werte sind normalverteilt mit – per Definition – Mittel 100 und Standardabweichung 15. Abbildung 4.4 zeigt die Wahrscheinlichkeitsdichte und die kumulative Wahrscheinlichkeit dieser Normalverteilung.

Wenn wir zufällig eine Person aus der Gesamtpopulation wählen, wie wahrscheinlich ist es dann, dass ihr IQ niedriger als 115 ist? Diese Wahrscheinlichkeit entspricht der Fläche unter der Wahrscheinlichkeitsdichte zwischen $-\infty$ (minus unendlich) und 115; diese Fläche wurde in der linken Grafik rötlich eingefärbt. Mit der `pnorm()`-Funktion können wir diesen Wert genau berechnen (roter Wert in der rechten Grafik; visuell geschätzt: 85%):

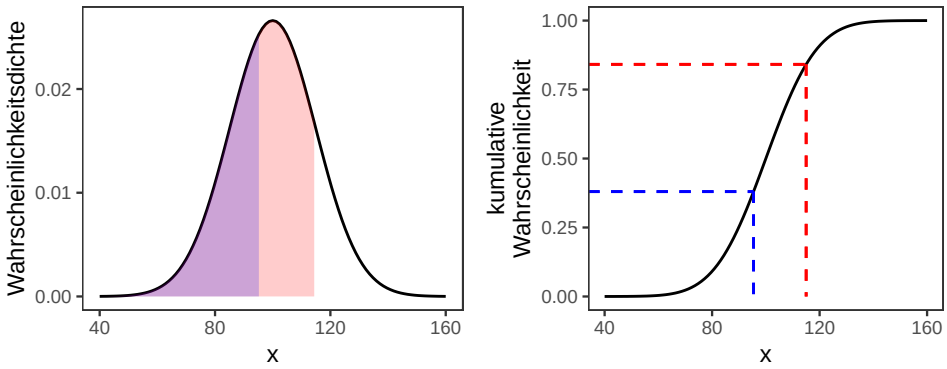


Abbildung 4.4: Wahrscheinlichkeitsdichte und kumulative Wahrscheinlichkeit einer Normalverteilung mit Mittel 100 und Standardabweichung 15.

```
pnorm(115, mean = 100, sd = 15)
```

```
## [1] 0.84134
```

Die Wahrscheinlichkeit, dass eine zufällig ausgewählte Person einen IQ von 115 oder niedriger hat liegt also bei 84%.

Mit der Option `lower.tail = FALSE` können wir das Komplement dieses Werts berechnen, d.h., die Wahrscheinlichkeit, einen Wert höher als 115 anzutreffen:

```
pnorm(115, mean = 100, sd = 15, lower.tail = FALSE)
```

```
## [1] 0.15866
```

```
# oder:
1 - pnorm(115, mean = 100, sd = 15)
```

```
## [1] 0.15866
```

Wir können die Frage auch andersherum stellen, z.B.: Für welchen IQ-Wert gilt, dass 38% der Population einen niedrigeren IQ haben? Hierzu verwenden wir die `qnorm()`-Funktion (*q* für *quantile*) (blauer x-Wert in der obigen Grafik):

```
qnorm(0.38, mean = 100, sd = 15)
```

```
## [1] 95.418
```

38% der Population hat also einen IQ niedriger als 95.4. Anders gesagt: Das 38. **Perzentil** der IQ-Verteilung (einer Normalverteilung mit Mittel 100 und einer Standardabweichung von 15) ist 95.4.

Eine andere Frage könnte sein: Zwischen welchen zwei Werten, die symmetrisch um das Mittel liegen, befinden sich 80% der IQ-Werte in der Population? Symmetrisch ums Mittel liegen 80% der Daten zwischen dem 10. und 90. Perzentil, daher:

```
qnorm(0.10, mean = 100, sd = 15)
## [1] 80.777
qnorm(0.90, mean = 100, sd = 15)
## [1] 119.22
```

Oder mithilfe der `c()`-Funktion (*combine*):

```
qnorm(c(0.10, 0.90), mean = 100, sd = 15)
## [1]  80.777 119.223
```

4.3 Wahrscheinlichkeiten kombinieren

Die folgenden Übungen dienen dazu, Sie mit dem Rechnen mit Wahrscheinlichkeiten vertrauter zu machen. Für diese Übungen brauchen Sie nicht mit R zu arbeiten.

1. M&Ms kommen in sechs Farben vor; in Tabelle 4.1 werden ihre relativen Frequenzen aufgelistet
   (a) Wie wahrscheinlich ist es, dass ein zufällig ausgewähltes M&M rot *oder* orange ist?
   (b) Wie wahrscheinlich ist es, dass zwei zufällig ausgewählte M&Ms *beide* rot oder orange (also zwei rote, zwei orange oder ein rotes und ein oranges) sind?
   (c) Wie wahrscheinlich ist es, dass von zwei zufällig ausgewählten M&Ms ein rotes *und* ein oranges dabei sind?
   (d) Wie wahrscheinlich ist es, dass wenn 5 M&Ms zufällig ausgewählt werden, *alle* blau sind?
   (e) Wie wahrscheinlich ist es, dass wenn 5 M&Ms zufällig ausgewählt werden, *kein einziges* blau ist?
   (Tipp: Wie wahrscheinlich ist es, dass Sie ein einziges M&M nehmen und es nicht blau ist?)

Tabelle 4.1: Relative Frequenzen von M&Ms nach Farbe.

Farbe	relative Frequenz
blau	23%
orange	23%
gelb	15%
grün	15%
braun	12%
rot	12%

2. (a) Wie wahrscheinlich ist es, dass eine zufällig ausgewählte Person einen IQ niedriger als 90 hat? (Siehe vorige Seiten für die IQ-Verteilung.)

   (b) Wie wahrscheinlich ist es, dass eine zufällig ausgewählte Person einen IQ grösser als 85 hat?

   (c) Wie wahrscheinlich ist es, dass eine zufällig ausgewählte Person einen IQ zwischen 110 und 120 hat?

   (d) Wie wahrscheinlich ist es, dass eine willkürlich ausgewählte Person einen IQ hat, der mehr als zwei Standardabweichungen vom Populationsmittel entfernt liegt?

   (e) Durchschnittliche Intelligenz ist definiert als der IQ der mittleren 45% der Bevölkerung. Zwischen welchen zwei Werten liegt er?

   (f) Die folgenden Übungen sind etwas schwieriger und haben als Ziel, Sie über kombinierte Wahrscheinlichkeiten nachdenken zu lassen. Wie wahrscheinlich ist es, dass, wenn zwei Personen zufällig ausgewählt werden, keine der beiden einen IQ niedriger als 105 hat?
   (Tipp: Wie wahrscheinlich ist es, dass eine einzige Person einen IQ höher als 105 hat?)

   (g) Wie wahrscheinlich ist es, dass, wenn drei Personen zufällig ausgewählt werden, *genau* eine Person einen IQ niedriger als 90 hat?
   (Tipp: Wie wahrscheinlich ist es, dass die erste Person einen IQ niedriger als 90 hat, die zweite und die dritte aber nicht? Was ist nun die Wahrscheinlichkeit, dass die zweite Person einen IQ niedriger als 90 hat, die erste und die dritte aber nicht? Und wie wahrscheinlich ist es, dass die dritte Person einen IQ niedriger als 90 hat, die ersten zwei aber nicht?)

   (h) Wie wahrscheinlich ist es, dass, wenn drei Personen zufällig ausgewählt werden, *mindestens* eine Person einen IQ niedriger als 90 hat?
   (Tipp: Wie wahrscheinlich ist es, dass keine einzige Person einen IQ niedriger als 90 hat?)

3. Wie wahrscheinlicht ist es, bei einer normalverteilten Variable (*egal welcher!*) einen zufällig ausgewählten Wert, der weniger als 1; 1,5; und 2 Standardabweichungen vom Mittel entfernt ist, anzutreffen?
   (Tipp: Zeichnen Sie ein paar Normalverteilungen mit anderen Mitteln und Standardabweichungen und beantworten Sie diese Frage für jede Verteilung separat.)

4. Ein Blatt Spielkarten zählt 52 Karten: 13 Werte (2, 3, . . . , Bube, Dame, König, Ass) in vier Farben (Kreuz, Pik, Herz und Karo).

   (a) Wie wahrscheinlich ist es, dass eine zufällig ausgewählte Karte ein Ass ist?

   (b) Sie ziehen zufällig eine Karte aus dem Blatt und legen diese auf die Seite. Dann ziehen Sie nochmals eine Karte aus dem gleichen Blatt. Wie wahrscheinlich ist es, dass Sie zwei Asse gezogen haben?

Kapitel 5

Zufallsstichproben

In Kapitel 3 sind wir davon ausgegangen, dass die Daten, die uns zur Verfügung standen, die ganze Population, für die wir uns interessierten, darstellten. Oft stellen die Daten, die gesammelt wurden, aber nur einen kleinen Teil der Population von Interesse da, d.h., sie bilden eine **Stichprobe**. Das Ziel ist es dann, anhand dieser Stichprobe Rückschlüsse über die Population, aus der die Stichprobe stammt, zu ziehen. Von Interesse sind also nicht sosehr die zentrale Tendenz und Streuung in der Stichprobe, sondern die zentrale Tendenz und Streuung in der Population.

Hier gehen wir davon aus, dass uns eine **Zufallsstichprobe** (*random sample*) aus der Population von Interesse zur Verfügung steht. Bei einer Zufallsstichprobe hat jedes *Element* aus der Population die gleiche Wahrscheinlichkeit, ausgewählt zu werden. Das heisst nicht, dass jeder *Wert* die gleiche Wahrscheinlichkeit hat, ausgewählt zu werden: Je nach Population kommen bestimmte Werte häufiger vor als andere. Dieses Kapitel widmet sich diesen beiden Fragen:

1. Wie können wir anhand einer Zufallsstichprobe am besten die zentrale Tendenz (insbesondere das Mittel) und die Streuung (insbesondere die Varianz und Standardabweichung) der Population **schätzen**?
2. Wenn wir unterschiedliche Zufallsstichproben aus der gleichen Population ziehen, wie stark unterscheiden sich diese Stichproben dann?

5.1 Stichprobenfehler

Zufallsstichproben widerspiegeln nicht perfekt jeden Aspekt der Population, aus der sie stammen. Um dies besser einzusehen, lohnt es sich, Zufallsstichproben aus Populationen zu ziehen, deren Eigenschaften wir kennen. So können wir sehen, wie stark diese von der Population und voneinander abweichen. Dies können wir tun, indem wir am Computer Stichproben **simulieren**.

Aufgabe Mit dem R-Code unten können Sie einschätzen, wie Stichproben aus einer normalverteilten Population aussehen. Zunächst habe ich hier die Stichprobengrösse auf 20 festgelegt, aber mit dieser Zahl sollten Sie selber herumspielen. Mit der Funktion `rnorm()` werden Zufallsstichproben einer bestimmten Grösse und mit bestimmten Parametern (Mittel, Standardabweichung) generiert (*r* für *random*). Die `hist()`-Funktion zeichnet ein einfaches Histogramm, ähnlich wie in Abschnitt 3.2. Führen Sie diese Befehle aus und zwar nicht ein Mal, sondern mehrmals. Bemerken Sie dabei, wie (un)ähnlich sich die Histogramme von Stichproben aus einer Normalverteilung sind.

```
# Stichprobengrösse definieren
groesse <- 20

# Stichprobe aus Normalverteilung ziehen: rnorm
# (hier: Mittel 3 und Standardabweichung 7)
x <- rnorm(n = groesse, mean = 3, sd = 7)

# Histogramm zeichnen
hist(x, col = "lightgrey")
```

Aufgabe Passen Sie den Code oben so an, dass er Stichproben aus einer Gleichverteilung mit Bereich [-5, 5] statt aus einer Normalverteilung generiert. Die Funktion, die Sie dazu brauchen, ist `runif()`. Neben dem `n`-Parameter hat diese Funktion einen `min`- und `max`-Parameter, mit denen der Bereich der Gleichverteilung eingestellt wird. Lassen Sie den angepassten Code dann mehrmals laufen. Sehen die einzelnen Histogramme aus wie Gleichverteilungen? Was ist, wenn Sie die Stichprobengrösse vergrössern?

Fazit Zufallsstichproben sind imperfekte Abbildungen der Population, aus der sie stammen. Diese Gegebenheit bezeichnet man als **Stichprobenfehler** (*sampling error*). Rückschlüsse über die Population verstehen sich also als **Schätzungen**. Sowohl die Schätzungen selbst als auch das Quantifizieren ihrer Genauigkeit sind das Ziel der **Inferenzstatistik**.

5.2 Die zentrale Tendenz und Streuung der Population anhand einer Stichprobe schätzen

In Kapitel 3 wurden das Mittel und die Varianz als Masse der zentralen Tendenz und der Varianz einer Population eingeführt. Hier erkunden wir, inwieweit diese Masse nützlich sind, um anhand einer Stichprobe die zentrale Tendenz und die Streuung einer Population zu erfassen. Wenn wir unterschiedliche Stichproben aus der gleichen Population ziehen und ihr Mittel und ihre Varianz ähnlich wie in Kapitel 3 berechnen, dann werden die Ergebnisse aufgrund des Stichprobenfehlers natürlich bei jeder Stichprobe

anders sein. Es wäre jedoch gut zu wissen, ob die Stichprobenmittel und -varianzen *im Durchschnitt* dem Populationsmittel bzw. der Populationsvarianz entsprechen.

Um dieser Frage nachzugehen, simulieren wir wieder Zufallsstichproben. Der folgende Code bewirkt Folgendes: Wir ziehen 10'000 Zufallsstichproben von je 5 Beobachtungen (`groesse`) aus einer Gleichverteilung mit Bereich [-5, 5]. Von jeder Stichprobe berechnen wir das Mittel mit der `mean()`-Funktion und die Varianz. Die Varianz wird mit der selbst geschriebenen Funktion `pop_var()` berechnet; siehe Seite 41. Um die 10'000 Stichproben zu generieren, wird hier ein *for-loop* verwendet (siehe Kommentare im Code).

```
# Stichprobengrösse festlegen - rumspielen!
groesse <- 5

# Anzahl Simulationen
n_sim <- 10000

# Insgesamt werden 10'000 Mittel und 10'000 Varianzen berechnet.
mittel <- vector(length = n_sim)
varianz <- vector(length = n_sim)

# 'for-loop': Die Schritte zwischen den geschwungenen
# Klammern werden 10'000 Mal ausgeführt.
for (i in 1:n_sim) {
  # Stichprobe aus einer Gleichverteilung
  # mit Bereich -5 bis 5 ziehen.
  x <- runif(n = groesse, min = -5, max = 5)

  # Mittel berechnen und speichern
  mittel[i] <- mean(x)

  # Varianz berechnen und speichern
  varianz[i] <- pop_var(x)
}
```

5.2.1 Das Stichprobenmittel

Das Mittel einer gleichverteilten Population mit Bereich $[a, b]$ liegt bei $\frac{b-a}{2}$. (Solche Infos findet man auf Wikipedia: `https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)`). In unserem Fall ist $\mu = \frac{5-(-5)}{2} = 0$. Das Mittel der Stichprobenmittel sollte also nahe bei 0 liegen:

```
mean(mittel)
```

```
## [1] -0.0085015
```

Aufgabe Dieses Ergebnis wird aufgrund des Stichprobenfehlers bei Ihnen etwas anders aussehen. Vergleichen Sie daher Ihr Ergebnis mit den Ergebnissen Ihrer KollegInnen (oder lassen Sie den Code mehrmals laufen). Wenn das Mittel einer Stichprobe im Schnitt dem Mittel der Population entspricht, sollte bei etwa der Hälfte Ihrer KollegInnen das Mittel der Stichprobenmittel grösser und bei der Hälfte kleiner als $\mu = 0$ sein.

Fazit Wenn wir das Mittel einer Zufallsstichprobe auf die gleiche Art und Weise berechnen wie das Mittel einer Stichprobe, erhalten wir *im Schnitt*, *über Tausende von Stichproben hinweg* das Populationsmittel. (Man sagt auch, dass der 'Erwartungswert' des Stichprobenmittels gleich dem Populationsmittel ist.) Das Stichprobenmittel (Kürzel: $\bar{x}$) ist also eine sog. 'unverzerrte' (*unbiased*) Schätzung des Populationsmittels (μ) und wird gleich berechnet:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{5.1}$$

5.2.2 Die Stichprobenvarianz

Der Wikipediaseite können wir entnehmen, dass die Varianz einer gleichverteilten Population mit Bereich $[a, b]$ $\sigma^2 = \frac{(b-a)^2}{12}$ ist. In unserem Fall also $\sigma^2 = \frac{(5-(-5))^2}{12} \approx 8.33$. Das Mittel der berechneten Varianzen sollte also nahe bei 8.33 liegen.

```
mean(varianz)
```

```
## [1] 6.6916
```

Aufgabe 1 Dieses Ergebnis wird aufgrund des Stichprobenfehlers bei Ihnen etwas anders aussehen. Vergleichen Sie daher Ihr Ergebnis mit den Ergebnissen Ihrer KollegInnen (oder lassen Sie den Code mehrmals laufen). Wenn die Varianz einer Stichprobe im Schnitt der Varianz der Population entspricht, sollte bei etwa der Hälfte Ihrer KollegInnen das Mittel dieser Varianz grösser und bei der Hälfte kleiner als $\sigma^2 = 8.33$ sein.

Aufgabe 2 Ändern Sie den Code oben, sodass jede Stichprobe aus bloss 2 Beobachtungen besteht. Lassen Sie den Code laufen und vergleichen Sie die durchschnittliche Varianz der Stichproben mit jener aus Aufgabe 1. Machen Sie dies auch für grössere Stichproben, z.B., mit 30 Beobachtungen.

Fazit Wenn wir die Varianz einer Stichprobe ähnlich berechnen wie die Varianz einer Population, erhalten wir im Schnitt einen Wert, der niedriger als die Populationsvarianz ist. Je kleiner die Stichprobe, desto mehr wird die Populationsvarianz unterschätzt.

Intuitiv lässt sich der Grund für diese Unterschätzung so verstehen: Wenn wir Stichproben von je nur einer Beobachtung aus einer Population ziehen, gibt es keine Streuung innerhalb jeder Stichprobe—die Beobachtung kann ja nicht von sich selbst abweichen. Bei der kleinst möglichen Stichprobe ist die Unterschätzung der Varianz in der Population also maximal. In grösseren Stichproben ist dieses Problem in stets geringerem Ausmass vorhanden.

Es stellt sich heraus, dass die Unterschätzung der Populationsvarianz vorhersagbar ist. Wenn die Stichprobe 5 Beobachtungen zählt, wird das Mittel der Varianzen der Stichproben $\frac{4}{5}$ Mal so gross sein als die Populationsvarianz. Zählt die Stichprobe 10 Beobachtungen, wird es $\frac{9}{10}$ Mal so gross sein, und so weiter ($\frac{n-1}{n}$). Um dies zu kompensieren, wird die Stichprobenvarianz (Kürzel: s^2) *nicht* wie die Populationsvarianz (σ^2), sondern folgendermassen berechnet:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{5.2}$$

Im Unterschied zur Populationsvarianz wird in dieser Formel das Stichprobenmittel verwendet (da das Populationsmittel in der Regel nicht bekannt ist) und wird die Summe der Quadrate durch $n - 1$ statt durch n geteilt. Letzteres kompensiert für die Überschätzung. Die `var()`-Funktion führt diese Berechnung aus.

5.2.3 Die Stichprobenstandardabweichung

Aus dem gleichen Grund, weshalb die Stichprobenvarianz nicht wie die Populationsvarianz berechnet wird, wird die Stichprobenstandardabweichung (s) nicht wie die Populationsstandardabweichung (σ) berechnet, sondern wie folgt:

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{5.3}$$

In R kann hierfür die `sd()`-Funktion verwendet werden.[1] Die Stichprobenvarianz- und standardabweichung der Norwegischergebnisse können also einfach so berechnet werden:

```
# Varianz
var(d$Norwegian)

## [1] 8.5375
```

[1] Ein kleines Detail: Während die Stichprobenvarianz ein unverzerrtes Mass der Populationsvarianz ist, unterschätzt die Stichprobenstandardabweichung die Populationsstandardabweichung noch immer leicht. Diese Unterschätzung zu korrigieren ist im besten Fall schwierig und meistens unmöglich. Sie ist aber ziemlich gering, sodass man diese Formel verwendet und die Unterschätzung in Kauf nimmt.

```r
# Standardabweichung
sd(d$Norwegian)
```

```
## [1] 2.9219
```

Es kommt eigentlich quasi nie vor, dass man für einen Datensatz die Populationsvarianz und -standardabweichung berechnet. Spricht man in diesem Kontext von der Varianz und Standardabweichung, meint man also die Stichprobenvarianz und die Stichprobenstandardabweichung. Bei grossen Populationen oder Stichproben ergeben beide Berechnungsmethoden ohnehin nahezu das Gleiche.

5.3 Die Verteilung von Stichprobenmitteln

Wie die Simulationen in Abschnitt 5.2.1 zeigen, haben sind die Mittel von Zufallsstichproben *im Schnitt* dem Populationsmittel gleich. Die einzelnen Mittel werden sich aber immer zumindest etwas von ihm unterscheiden. Aber wie stark weichen einzelne Stichprobenmittel nun vom Populationsmittel ab? Um diese Frage zu beantworten, simulieren wir wieder ein paar Szenarien.

5.3.1 Simulationen

Aufgabe 1: Stichproben aus einer Normalverteilung Für Aufgaben 1–3 werden wir Zufallsstichproben aus den drei Populationen in Abbildung 5.1 generieren und schauen, wie die Mittel dieser Stichproben verteilt sind.

Für die erste Aufgabe ziehen wir Stichproben aus einer Normalverteilung mit $\mu = 80$ und $\sigma^2 = 400$ (also $\sigma = 20$).

1. Kopieren Sie den Code auf Seite 57 und passen Sie ihn so an, dass er Stichproben aus dieser Normalverteilung generiert. (Den R-Befehl dafür finden Sie am Anfang dieses Kapitels.)
2. Verwenden Sie diesen Code, um 1'000 Stichproben mit je 2 Beobachtungen aus dieser Verteilung zu generieren und ihr Mittel zu berechnen.
3. Zeichnen Sie ein Histogramm der 1'000 Stichprobenmittel. Beschreiben Sie die Verteilung der Stichprobenmittel. Achten Sie dabei auch auf die Werte auf der x-Achse.
4. Berechnen Sie die Varianz der 1'000 Stichprobenmittel und notieren Sie das Ergebnis.
5. Wiederholen Sie Schritte 2–4 für Stichproben mit 5, 20 und 100 Beobachtungen. Was stellen Sie fest?

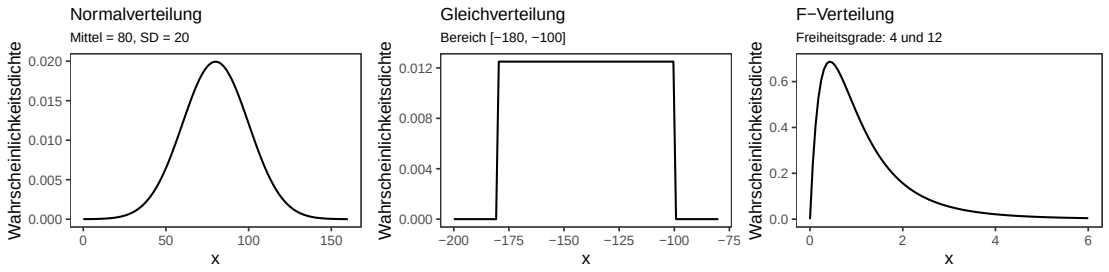


Abbildung 5.1: Drei Populationen, aus denen hier Stichproben generiert werden.

Aufgabe 2: Stichproben aus einer Gleichverteilung Für die zweite Aufgabe ziehen wir Stichproben aus einer Gleichverteilung mit Bereich [-180, -100]. Diese Verteilung hat $\mu = -140$ und $\sigma^2 \approx 533$.

1. Kopieren Sie den Code auf Seite 57 und passen Sie ihn so an, dass er Stichproben aus dieser Gleichverteilung generiert.
2. Verwenden Sie diesen Code, um 1'000 Stichproben mit je 2 Beobachtungen aus dieser Verteilung zu generieren und ihr Mittel zu berechnen.
3. Zeichnen Sie ein Histogramm der 1'000 Stichprobenmittel. Beschreiben Sie die Verteilung der Stichprobenmittel. Achten Sie dabei auch auf die Werte auf der x-Achse.
4. Berechnen Sie die Varianz der 1'000 Stichprobenmittel und notieren Sie das Ergebnis.
5. Wiederholen Sie Schritte 2–4 für Stichproben mit 5, 20 und 100 Beobachtungen. Was stellen Sie fest?

Aufgabe 3: Stichproben aus einer schiefen Verteilung Für die dritte Aufgabe ziehen wir Stichproben aus einer rechtsschiefen Verteilung. Es handelt sich hier um eine F-Verteilung mit den Freiheitsgraden (= Parametern) 4 und 12. Was eine F-Verteilung ist, ist im Moment nicht wichtig; wichtig ist nur, dass es sich um eine rechtsschiefe Verteilung handelt. Die $F(4, 12)$-Verteilung hat $\mu = 1.2$ und $\sigma^2 = 1.26$.

1. Kopieren Sie den Code auf Seite 57 und passen Sie ihn so an, dass er Stichproben aus einer $F(4, 12)$-Verteilung generiert. Statt des `runif(n = groesse, min = -5, max = 5)`-Befehls brauchen Sie `rf(n = groesse, df1 = 4, df2 = 12)`.
2. Verwenden Sie diesen Code, um 1'000 Stichproben mit je 2 Beobachtungen aus dieser Verteilung zu generieren und ihr Mittel zu berechnen.
3. Zeichnen Sie ein Histogramm der 1'000 Stichprobenmittel. Beschreiben Sie die Verteilung der Stichprobenmittel. Achten Sie dabei auch auf die Werte auf der x-Achse.

4. Berechnen Sie die Varianz der 1'000 Stichprobenmittel und notieren Sie das Ergebnis.
5. Wiederholen Sie Schritte 2–4 für Stichproben mit 5, 20 und 100 Beobachtungen. Was stellen Sie fest?

5.3.2 Fazit: Der zentrale Grenzwertsatz

Die Simulationen oben sollen den **zentralen Grenzwertzsatz** (*central limit theorem* (CLT)) illustrieren. Dieser Satz besagt Folgendes:

Wenn Zufallsstichproben mit n Beobachtungen aus einer Population mit Mittel μ und Varianz σ^2 gezogen werden,[2] sind die Stichprobenmittel *ungefähr normalverteilt*, wenn n gross genug ist. Dies gilt auch dann, *wenn die Population selber nicht normalverteilt ist.* Das Mittel der **Verteilung der Stichprobenmittel** ($\mu_{\bar{x}}$) ist gleich dem Populationsmittel (μ). Die Varianz der Stichprobenmittel ($\sigma^2_{\bar{x}}$) wird kleiner, je grösser die Stichproben sind:

$$\sigma^2_{\bar{x}} = \frac{\sigma^2}{n} \tag{5.4}$$

Die Standardabweichung der Verteilung der Stichprobenmittel, der **Standardfehler** (*standard error* (S.E.)), ist daher:

$$\sigma_{\bar{x}} = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}} \tag{5.5}$$

Diese Einsichten – dass die Mittel genügend grosser Stichproben annähernd normalverteilt sind und dass die Varianz dieser Normalverteilung proportional zur Stichprobengrösse abnimmt – sind von grösster Bedeutung für die Inferenzstatistik und für das Planen von Datenerhebungen, wie wir später sehen werden. Es stellt sich somit die Frage, was mit "wenn n gross genug ist" gemeint ist.

Die Antwort hängt von der Population, aus der die Stichproben stammen, ab. Die Simulationen sollen zeigen, dass bei Normalverteilungen die Stichprobenmittel bereits bei den kleinsten Stichprobengrössen normalverteilt sind. Auch bei *fast* normalverteilten Populationen und sogar bei gleichverteilten Populationen ist die Stichprobenmittelverteilung bereits bei ziemlich kleinen Stichproben annähernd normalverteilt. Bei schiefen Populationen, dahingegen, kann die Stichprobenmittelverteilung durchaus noch ein bisschen Schiefe aufweisen, auch wenn die Stichprobengrösse respektabel ist. (Probieren Sie mal Stichproben mit Grösse 50 aus der $F(4, 12)$-Verteilung zu ziehen.)

Impräzise an diesem Satz mag ausserdem erscheinen, dass er besagt, dass die Stichprobenmittel *annähernd* normalverteilt sein werden. Normalverteilungen reichen eigentlich von $-\infty$ bis ∞. Wenn man aber Stichproben aus, zum Beispiel, einer Gleichvertei-

[2] Es gibt ein paar Wahrscheinlichkeitsverteilungen, die kein Mittel oder keine Varianz haben. Für diese gilt der zentrale Grenzwertsatz nicht.

lung mit Bereich [-5, 20] generiert, dann werden die Stichprobenmittel natürlich immer zwischen -5 und 20 liegen. In diesem Sinne könnte die Verteilung der Stichprobenmittel aus dieser Population nie perfekt normalverteilt sein. Aber auch mit annähernd normalverteilten Stichprobenmitteln kommt man ein Stück weiter.

Beispiel 1 Die Verteilung der Mittel von Stichproben mit Grösse 36 aus einer Normalverteilung mit $\mu = 1.2$ und $\sigma^2 = 4.1$ hat ein Mittel von 1.2 und eine Standardabweichung von $\sqrt{\frac{4.1}{36}} \approx 0.34$ (= Standardfehler). Mit Stichprobengrössen von 50 bzw. 100 wäre der Standardfehler $\sqrt{\frac{4.1}{50}} \approx 0.29$ bzw. $\sqrt{\frac{4.1}{100}} \approx 0.20$. (Wer Lust hat, kann dies mit einer Simulation überprüfen.)

Beispiel 2 Wenn man mit einem fairen 6-seitigen Würfel würfelt sind die Werte 1, 2, 3, 4, 5 und 6 alle gleich wahrscheinlich. Anders als bei der kontinuierlichen Gleichverteilung aus Abschnitt 4.1 kann aber nicht jeder Wert im Bereich beobachtet werden: Man kann ja keine 1.72 würfeln. Eine solche Verteilung nennt man eine **diskrete Gleichverteilung**.

Der zentrale Grenzwertsatz trifft auch auf diskrete Gleichverteilungen zu. In diesem Fall hat die diskrete Gleichverteilung ein Mittel von 3.5 und eine Standardabweichung von 1.71 (oder eine Varianz von 2.92). Wenn man also mit 10 Würfeln mehrmals würfelt und jeweils das Mittel der Augen notiert, wird man feststellen, dass die Mittel ungefähr normalverteilt sind mit $\mu_{\bar{x}} = \mu = 3.5$ und $\sigma_{\bar{x}} = \frac{1.71}{\sqrt{10}} = 0.54$. Würfelt man mit 25 Würfeln, dann beträgt der Standardfehler $\frac{1.71}{\sqrt{25}} = 0.34$.

5.4 Übungen

1. Sie möchten wissen, wie viele Bücher im Schnitt in Schweizer Wohnzimmern vorhanden sind. Nach dem Zufallsprinzip wählen Sie acht Haushalte aus. Im Wohnzimmer jedes Haushalts zählen Sie die Anzahl Bücher pro Haushalt. Dies sind Ihre Ergebnisse:

   $$18, 10, 7, 142, 48, 27, 257, 14$$

   Tragen Sie diese Daten so in R ein:

   ```
   buecher <- c(18, 10, 7, 142, 48, 27, 257, 14)
   ```

   Sie können die Daten auch in ein Spreadsheet eintragen und dieses Spreadsheet in R einlesen.

   (a) Stellen Sie diese Daten grafisch dar und beschreiben Sie ihre Verteilung.

(b) Was ist Ihre beste Schätzung des Mittels der Anzahl Bücher pro Schweizer Haushalt?

(c) Was ist Ihre beste Schätzung der Varianz und der Standardabweichung der Anzahl Bücher pro Schweizer Haushalt?

(d) Erklären Sie, warum wir es hier mit Schätzungen zu tun haben. Warum sind wir uns nicht *sicher*, was das Mittel und die Streuung der Population betrifft?

2. Eine Gleichverteilung mit Bereich [-0.39, 20.39] hat $\mu = 10$ und $\sigma^2 = 36$.

(a) Wie wahrscheinlich ist es, dass eine Zufallsstichprobe mit 4 Beobachtungen ein Mittel von 5 oder weniger hat? Sie können davon ausgehen, dass der zentrale Grenzwertsatz zutrifft.

(b) Idem, aber für 10 Beobachtungen und für 50 Beobachtungen.

(c) Wie viel Prozent der Stichprobenmittel liegen mehr als 4 Einheiten von μ entfernt bei $n = 8$?

(d) Zwischen welchen zwei Werten liegen, symmetrisch um μ, 66.7% der Stichprobenmittel bei $n = 10$ und bei $n = 60$? Wie gross ist diese Entfernung zu μ, wenn man sie in Standardfehlern ausdrückt?

(e) Idem, aber für 90% und 95% der Stichprobenmittel.

5.5 Nicht-zufällige Stichproben

Wir haben uns in diesem Kapitel mit **Zufallsstichproben** (*random samples*) beschäftigt, also mit Stichproben, bei denen jedes Element in der Population die gleiche Wahrscheinlichkeit hat, ausgewählt zu werden, und bei denen die Auswahl eines Elements die Auswahl eines anderen Elementes nicht beeinflusst. Zwei grosse Vorteile von Zufallsstichproben sind, dass sie unverzerrte Schätzungen des Populationsmittels und der Populationsvarianz liefern und dass der zentrale Grenzwertsatz auf sie zutrifft.

In der Praxis ist es jedoch schwierig, eine Zufallsstichprobe aus einer einigermassen interessanten Population zu ziehen. Wenn wir etwa anhand einer Zufallsstichprobe die Englischkenntnisse bei Erwachsenen kosovarischer Herkunft im Kanton Sankt-Gallen charakterisieren möchten, brauchen wir zuerst eine vollständige Liste aller Erwachsenen kosovarischer Herkunft in SG. Dann müssten wir zufällig eine Stichprobe von ihnen auswählen und die Ausgewählten alle von einer Teilnahme an der Studie überzeugen: Sobald sich eine Person weigert, mitzumachen, haben wir keine Zufallsstichprobe mehr. Das Beispiel macht auch klar, was die Konsequenz hiervon ist: Während eine Zufallsstichprobe eine unverzerrte Schätzung des Populationsmittels liefert, wäre es bei einigen Weigerungen möglich, dass einige der ausgewählten Personen nicht zur Teilnahme bereit sind, gerade weil sie ihre Englischkenntnisse als ungenügend einschätzen oder weil sie sprachwissenschaftliche Forschung für uninteressant halten. Die Übrigen

dürften also tendenziell eher gut im Englischen sein oder sich eher für Sprachen interessieren; das Mittel dieser Stichprobe dürfte entsprechend das Populationsmittel eher über- als unterschätzen.[3]

Statt Zufallsstichproben werden daher in den Sozialwissenschaften meistens nicht-zufällige Stichproben verwendet. Die Konsequenz davon ist, dass man sich bei der Interpretation der Ergebnisse mehr Gedanken machen muss, wenn man Rückschlüsse über eine Population ziehen möchte, als wenn die Stichprobe zufällig ausgewählt worden wäre.

- Eine Meinungsumfrage auf Twitter erreicht tendenziell Menschen ähnlicher Meinung. Aber sogar die angesehensten Meinungsforschungsinstitute können keine Zufallsstichproben organisieren: Bei Telefonumfragen in den USA nehmen nur etwa 10% der Ausgewählten teil.
- Wer ohne Entgelt einen langen Fragebogen zu seinem mehrsprachigen Verhalten ausfüllt, findet Mehrsprachigkeit tendenziell wichtiger als jemand, der nach der dritten Frage das Browserfenster schliesst oder den Fragebogen nicht einmal erhalten hat (bei einer Erhebung nach dem Schneeballprinzip).
- Muster in einer gut ausgebildeten Stichprobe mit überdurchschnittlichem sozioökonomischem Status (z.B. Universitätsstudierende) dürften nicht auf Populationen mit niedrigerem Bildungsniveau oder sozioökonomischen Status generalisieren. Dies ist natürlich vor allem relevant, wenn Bildung und der sozioökonomische Status wichtig für den Forschungsgegenstand sind. Wenn man bereit ist, anzunehmen, dass diese Faktoren nur einen minimalen Effekt auf die Befunde haben, kann man zuversichtlicher generalisieren. Ob eine solche Annahme berechtigt ist, ist eine sachlogische – keine statistische – Frage.

5.6 Weitere Ressourcen

Die Website Khan Academy hat mehrere ausgezeichnete Videos, in denen statistische Konzepte erläutern werden, so auch ein Video zum zentralen Grenzwertsatz.

Unter `http://onlinestatbook.com/stat_sim/sampling_dist/` finden Sie eine Java-Applikation, die den zentralen Grenzwertsatz für belieblig verteilte Populationen illustriert.

[3] Eine etwas kompliziertere Art Stichprobe ist die geschichtete Zufallsstichprobe (*stratified random sample*). Hierzu teilt man die Population von Interesse (z.B. Studierende an der Universität Freiburg) in Gruppen auf (z.B. Studierende an der Philosophischen Fakultät, an der Theologischen Fakulät, an der Naturwissenschaftlichen Fakultät usw.). Dann zieht man zufällige Stichproben innerhalb jeder Gruppe. Somit hat man in der Stichprobe garantiert aus jeder Gruppe einige Beobachtungen (z.B. würde die Stichprobe sowieso Studierende an der Theologischen Fakultät beinhalten), aber ist es dennoch möglich, Aussagen über das Mittel und die Streuung in der Gesamtpopulation zu machen. Letzteres tut man grundsätzlich, indem man die Gruppenergebnisse nach der Gesamtgruppengrösse gewichtet. Geschichtete Zufallsstichproben werden wir in diesem Skript nicht behandeln.

Kapitel 6

Die Unsicherheit von Schätzungen einschätzen

Eine unumgängliche Gegebenheit beim Arbeiten mit Stichproben ist, dass wir Eigenschaften von Populationen ('Populationsparameter', z.B. Mittelwerte und Streuungsmasse, aber auch andere Parameter, denen wir in den nächsten Kapiteln begegen werden) nur *schätzen* können. Die Frage stellt sich, wie genau diese Schätzungen denn sind. Interessanterweise wissen wir dies in der Regel auch nicht genau, weshalb diese Unsicherheit *auch* anhand der Stichprobe geschätzt werden muss.

Das Ziel dieses Kapitels ist es, anhand eines Beispiels zu illustrieren, wie man mit einer Stichprobe die Unsicherheit in einer Parameterschätzung einschätzen kann. Dazu introduziert dieses Kapitel den sog. **Bootstrap**, ein flexibles, mechanistisches Verfahren, um Unsicherheit einzuschätzen. Danach wird gezeigt, wie man anhand des zentralen Grenzwertsatzes (Kapitel 5) das Gleiche machen kann.

6.1 Datensatz

DeKeyser et al. (2010) untersuchten, wie das Alter, in dem MigrantInnen angefangen haben, eine Zweitsprache zu lernen (*age of acquisition*, AOA), mit ihrer Leistung bei einer Grammatikaufgabe zusammenhängt (*grammaticality judgement task*, GJT). Die Teilnehmenden waren russische MigrantInnen in Israel und in Nordamerika. Die Grammatikaufgabe bestand aus 204 richtig/falsch-Items. In den nächsten Kapiteln werden wir uns mit dem Zusammenhang zwischen AOA und GJT befassen; hier verwenden wir den Datensatz von DeKeyser et al. (2010), um zu zeigen, wie man die Unsicherheit bei Stichprobenschätzungen quantifizieren kann.

Der Datensatz `dekeyser2010.csv` enthält die AOA- und GJT-Daten der russischen MigrantInnen in Nordamerika. Lesen Sie diesen Datensatz in R ein.

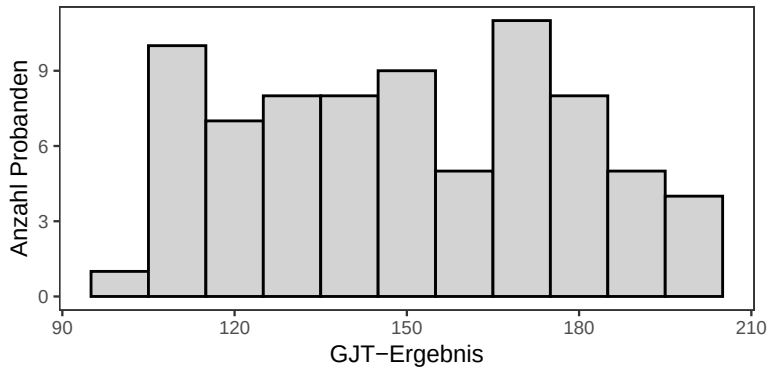


Abbildung 6.1: Histogramm der GJT-Daten aus der Nordamerika-Studie von DeKeyser et al. (2010)

```
library(tidyverse)
d <- read_csv("Data/dekeyser2010.csv")
```

Aufgabe 1 Zeichnen Sie die Grafik in Abbildung 6.1 selbst.

Aufgabe 2 Berechnen Sie das Mittel der GJT-Werte.

6.2 Stichprobenmittel variieren

Lasst uns davon ausgehen, dass die GJT-Daten in der ganzen Population so verteilt wären wie in Abbildung 6.1. Dies ist nur eine Annahme für didaktische Zwecke. Als Forschende haben wir keinen Zugriff zur ganzen Population, d.h., wir wissen eigentlich nicht, wie dieses Populationsverteilung aussieht. Stattdessen müssen wir uns mit Stichproben begnügen. Aber nehmen wir vorübergehend an, dass die Daten in der Population genau so verteilt wären wie in dieser Stichprobe.

Wie schon in Kapitel 5 besprochen, bilden die Mittel von Zufallsstichproben mit der gleichen Grösse eine Stichprobenmittelverteilung, deren Mittel gleich dem Populationsmittel ist ($\mu_{\bar{x}} = \mu$). Abbildung 6.2 zeigt exemplarisch fünf Stichproben mit Grösse 20 aus dieser GJT-Population und die Verteilung der Mittel von 20'000 Stichproben ($n = 20$) aus der Population. Die Standardabweichung der Stichprobenmittelverteilung, der Standardfehler (siehe Kapitel 5), beträgt 6.06 Punkte. 2.5% der Stichprobenmittel sind kleiner als 138.95; 2.5% sind grösser als 162.60. 95% aller Stichprobenmittel liegen also in einem Intervall von $162.60 - 138.95 = 23.65$ Punkten. Der Standardfehler oder die Breite eines solchen Intervalls wären sinnvolle Masse für die Genauigkeit, mit der man mit einer Stichprobe einen Populationsparameter schätzen kann.

Unser Problem ist aber, dass wir die Stichprobenmittelverteilung nur generieren können, wenn wir Zugriff zur ganzen Population haben. Wenn wir nur über eine Stichpro-

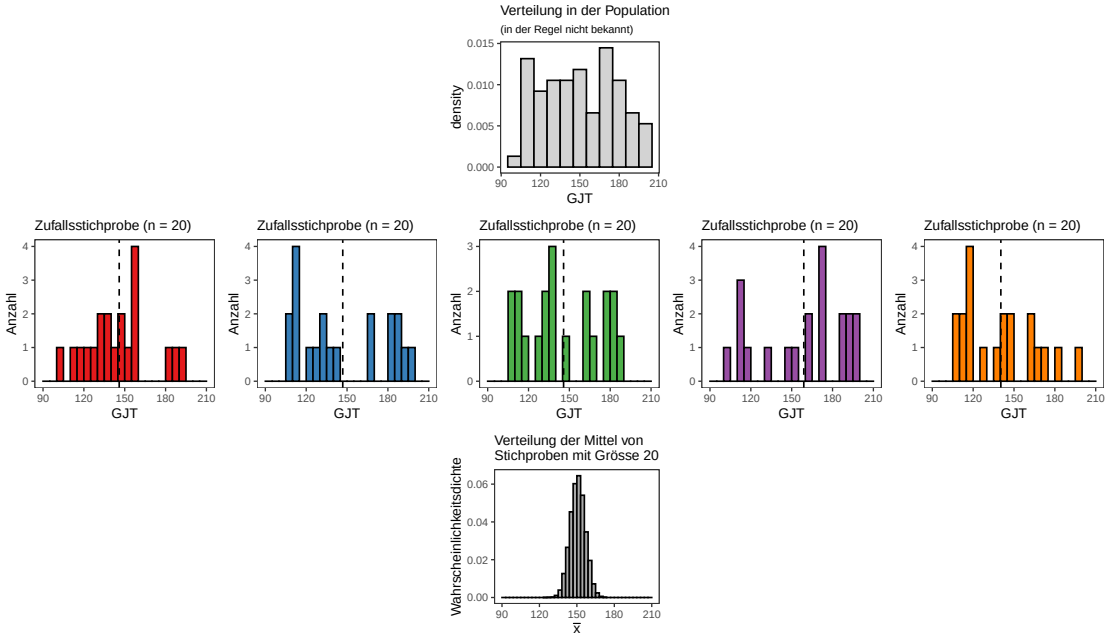


Abbildung 6.2: Wenn eine grosse Anzahl Stichproben mit der gleichen Grösse aus der Population gezogen werden und je ihr Mittel berechnet wird (senkrechte Linie), ergibt sich die Stichprobenmittelverteilung. In diesem Fall ist diese normalverteilt, aber dies ist nicht unbedingt der Fall. Exemplarisch werden fünf der Stichproben gezeigt.

be verfügen, müssen wir den Standardfehler bzw. die Breite solcher Intervalle anhand der Stichprobe schätzen.

6.3 Das *plug-in*-Prinzip und der *Bootstrap*

Enter the plug-in principle. Abbildung 6.2 zeigt zwar, dass jede einzelne Stichprobe die Population nur imperfekt widerspiegelt. Aber gleichzeitig ist diese Widerspieglung das Beste, was wir in der Praxis haben.[1] Um den Standardfehler bzw. die Form der Stichprobenmittelverteilung zu schätzen, können wir *die Stichprobe als Stellvertreter der Population* betrachten.[2]

6.3.1 Zwei Beispiele

Beispiel 1 Abbildung 6.3 zeigt das Vorgehen. Zur Verfügung steht uns die erste (rote) Stichprobe aus Abbildung 6.2. Wir tun nun, als ob die GJT-Population genau so wie diese Stichprobe verteilt wäre, denn wir haben keine besseren Anknüpfungspunkte. Um die Stichprobenmittelverteilung zu generieren, ziehen wir nun Zufallsstichproben

[1] Sogenannte bayessche Methoden erlauben es einem aber, auch Informationen, die man nicht aus den Daten selber ableiten kann, in der Analyse zu berücksichtigen.

[2] Dieser Teil wurde inspiriert von Hesterberg (2015).

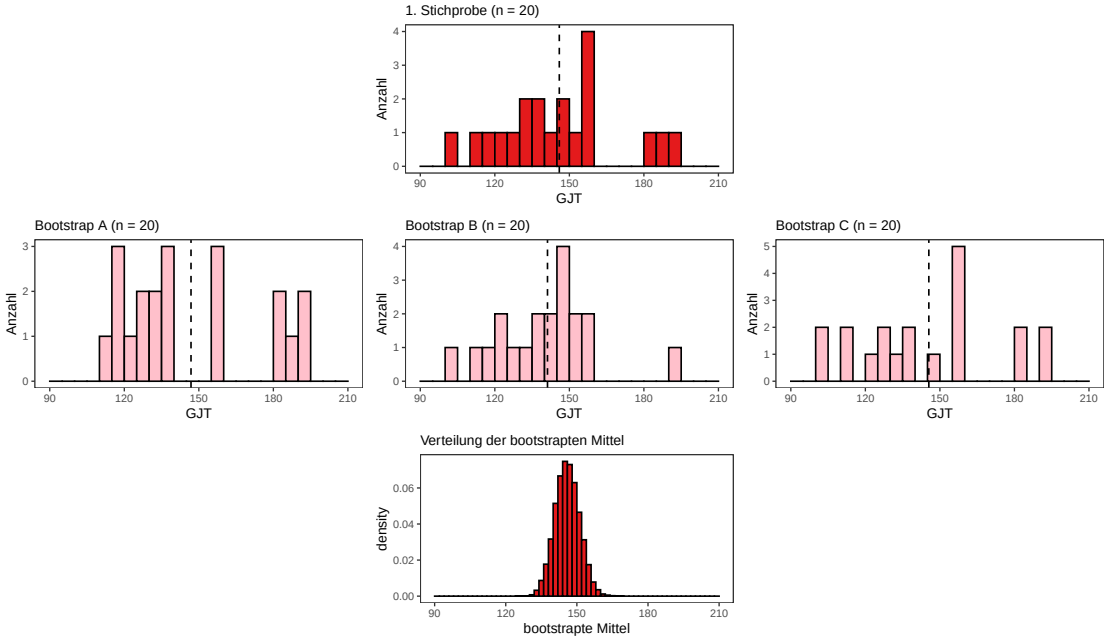


Abbildung 6.3: Die erste Stichprobe aus Abbildung 6.2 dient hier als Stellvertreter der GJT-Population. Exemplarisch werden drei Bootstrap-Stichproben mit Grösse 20 gezeigt. Wenn man 20'000 solche Bootstrap-Stichproben generiert, bilden ihre Mittel die Verteilung in der unteren Grafik. Diese hier schaut normalverteilt aus, aber dies ist nicht zwingend der Fall.

mit Grösse 20 aus dieser Stichprobe.[3] Diese Stichproben werden *Bootstrap*-Stichproben (oder *bootstrap replicates*) genannt. Abbildung 6.3 zeigt exemplarisch drei solche *Bootstrap*-Stichproben. Für jede Bootstrap-Stichprobe können wir das Mittel berechnen; die Verteilung 20'000 dieser Mittel steht in der unteren Grafik.

Das Mittel der bootstrapten Mittel ist gleich dem Mittel der Stichprobe (145.95). Ihre Standardabweichung beträgt etwa 5.20. 2.5% der bootstrapten Mittel sind kleiner als 135.95; 2.5% sind grösser als 156.20; die Breite dieses Intervalls ist also 20.25 Punkte.

Beispiel 2 Abbildung 6.4 zeigt das Verfahren noch einmal, diesmal mit der zweiten (blauen) Stichprobe aus Abbildung 6.2 als Ausgangspunkt. Das Mittel der bootstrapten Mittel ist gleich dem Mittel der Stichprobe (146.85). Ihre Standardabweichung beträgt etwa 7.03. 2.5% der bootstrapten Mittel sind kleiner als 133.55; 2.5% sind grösser als 161.00; die Breite dieses Intervalls ist also 27.45 Punkte.

6.3.2 Die Essenz des Bootstraps

Der Bootstrap ist eine ziemlich neue Technik (Efron, 1979; Efron & Tibshirani, 1993), um die Unsicherheit in Parameterschätzungen zu quantifizieren. Die tatsächliche Unsi-

[3] Ein Detail: Eine Beobachtung darf mehrmals in der gleichen Stichprobe vorkommen. Dies nennt man *sampling with replacement* und man macht es, weil man davon ausgeht, dass die Population (praktisch gesehen) unendlich gross ist.

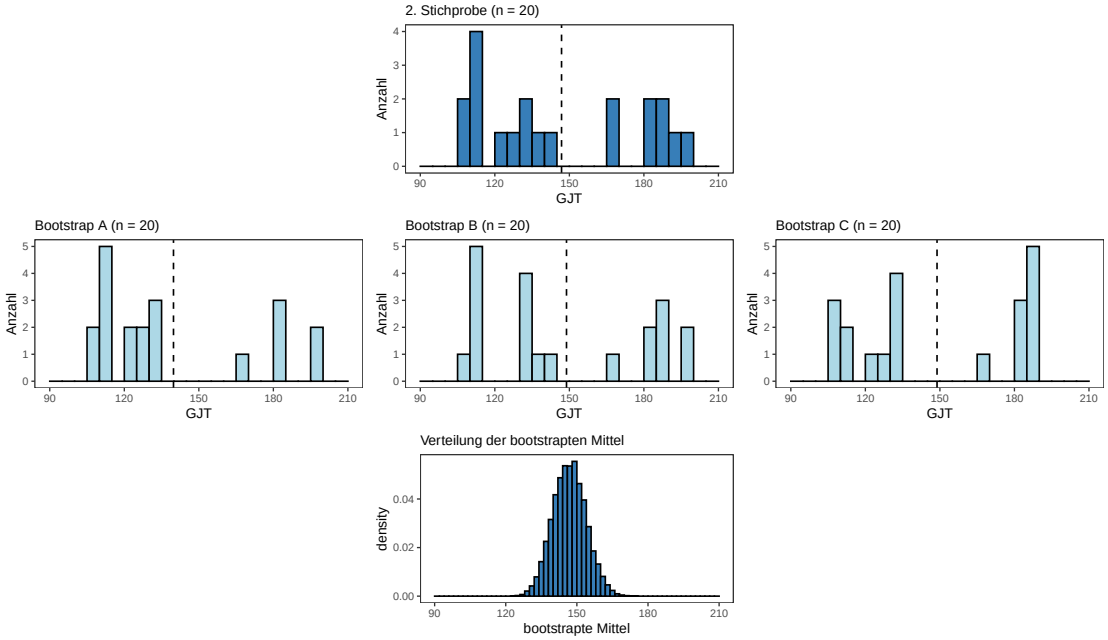


Abbildung 6.4: Die zweite Stichprobe aus Abbildung 6.2 dient hier als Stellvertreter der GJT-Population. Exemplarisch werden drei Bootstrap-Stichproben mit Grösse 20 gezeigt. Wenn man 20'000 solche Bootstrap-Stichproben generiert, bilden ihre Mittel die Verteilung in der unteren Grafik. Diese hier schaut normalverteilt aus, aber dies ist nicht zwingend der Fall.

cherheit in einer Parameterschätzung können wir nur berechnen, wenn wir eine grosse Anzahl Stichproben aus der gleichen Population ziehen und feststellen, wie die Schätzungen zwischen den Stichproben variieren:

- Population
  → Stichproben
  → Verteilung von Schätzungen in Stichproben
  → Variabilität in Schätzung berechnen

Mangels einer grossen Anzahl Stichproben aus der gleichen Population, verlässt man sich auf das *plug-in*-Prinzip: Die Stichprobe tritt stellvertretend für die Population auf, und geschaut wird, wie gut Stichproben einer bestimmten Grösse aus dieser Stichprobe den untersuchten Parameter schätzen können.

- Stichprobe
  → Bootstrap-Stichproben
  → Verteilung von Schätzungen in Bootstrap-Stichproben
  → Variabilität in Schätzung *schätzen*

Der Bootstrap ergibt eine *Schätzung* der Unsicherheit in der Parameterschätzung. Das wird klar, wenn man sich Abbildung 6.5 auf Seite 72 und Tabelle 6.1 anschaut. Abbildung 6.5 zeigt die Verteilung der bootstrapten Mittel für die fünf Stichproben; Tabelle 6.1 fasst ihre Standardabweichung, ihre 2.5. und 97.5. Perzentile, und den Unterschied

zwischen diesen Perzentilen zusammen. Die Standardabweichungen und die Breite der Intervalle zwischen dem 2.5. und dem 97.5. Perzentil sind in keinem der fünf Beispiele dem entsprechenden tatsächlichen, aber unbekannten Wert, gleich.

Aber im Schnitt sind sie ihm recht ähnlich. *Insbesondere wenn man über keine weiteren Anknüpfungspunkte verfügt* (z.B. vorherigen Studien, sachlogische Überlegungen), kann der Bootstrap also nützliche Informationen über die Unsicherheit einer Parameterschätzung liefern.

6.3.3 Vorteile des Bootstraps

- **Didaktisch wertvoll** (hoffe ich). Die mathematischen Anforderungen sind beim Bootstrappen gering. Dies erlaubt uns, wichtige Konzepte unabhängig von ihrer üblichen mathematischen Umsetzung zu besprechen.
- **Flexibilität.** Hier haben wir uns mit der Unsicherheit eines Stichprobenmittels befasst. Diese kann man auch mit einer relativ einfachen analytischen Methode ausdrücken (siehe unten). Der Bootstrap kann man aber auch verwenden, um die Unsicherheit vieler anderer Schätzungen auszudrücken, zum Beispiel eines getrimmten oder winsorisierten Mittels, eines Medians, einer Standardabweichung, eines bestimmten Perzentils, oder irgendwelcher anderen Masse. Ausserdem kann der Bootstrap auch bei komplexeren Modellen (z.B., wenn wir den Zusammenhang zwischen verschiedenen Variablen untersuchen) verwendet werden.
- **Minimale Annahmen.** Verglichen mit anderen Verfahren basiert der Bootstrap auf nur wenigen Annahmen. Dieser Punkt wird später in diesem Kapitel deutlicher werden, aber hier sei bereits darauf hingewiesen, dass wir in den obigen Beispielen nirgends davon ausgegangen sind, dass die Stichprobenmittelverteilung normalverteilt ist. (In den Beispielen sind die Verteilungen der bootstrapten Mittel zwar normalverteilt, aber hiervon sind wir nicht a priori *ausgegangen*.) Wir

Tabelle 6.1: Standardabweichung, Perzentile und der Unterschied zwischen den Perzentilen für die eigentliche Stichprobenmittelverteilung und die fünf Verteilungen der bootstrapten Mittel. (Die Perzentile und der Unterschied zwischen ihnen wurden gerundet.)

Stichprobenmittelverteilung	SD	2.5. Perzentil	97.5. Perzentil	Unterschied
Tatsächlich (= unbekannt)	6.1	139	163	24
Bootstrap Stichprobe 1	5.2	136	156	20
Bootstrap Stichprobe 2	7.0	134	161	27
Bootstrap Stichprobe 3	6.0	134	158	23
Bootstrap Stichprobe 4	6.9	145	172	27
Bootstrap Stichprobe 5	5.9	129	152	23

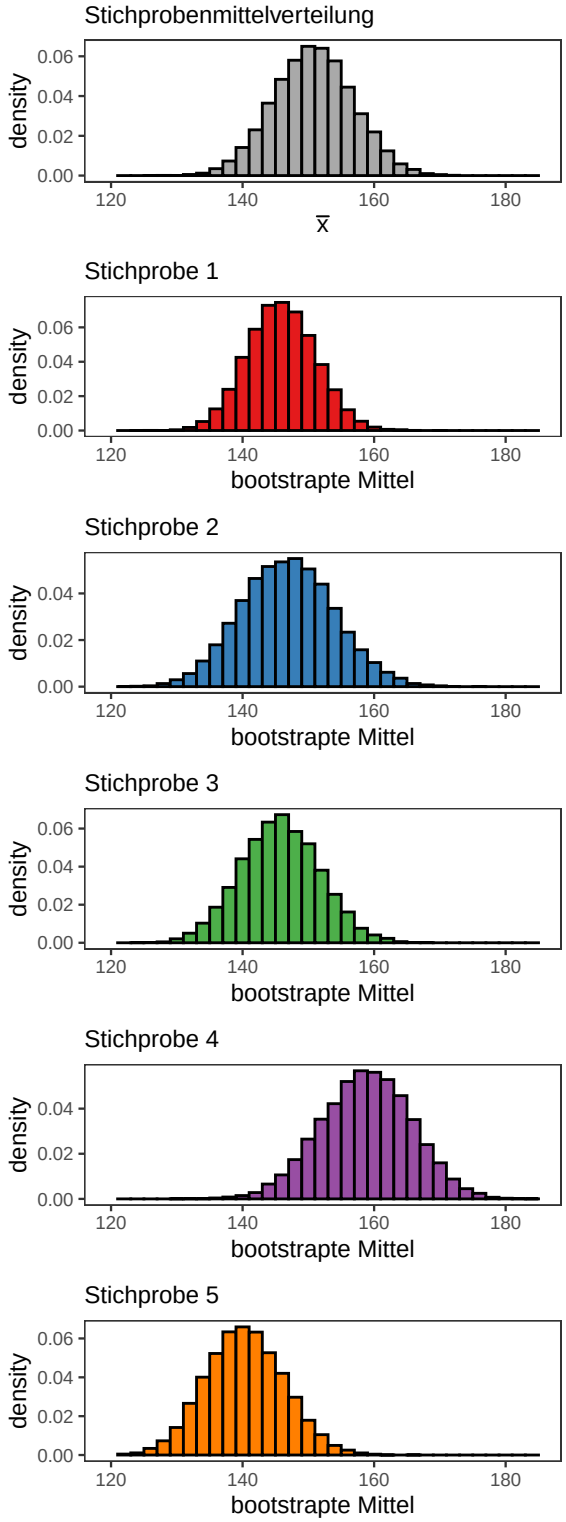


Abbildung 6.5: Die Verteilung der bootstrapten Mittel auf der Basis von fünf Stichproben.

sind auch nicht davon ausgegangen, dass die Population, aus der die Stichproben stammen, normalverteilt ist.

6.3.4 Nachteile des Bootstraps

- "Bootstrapping does not overcome the **weakness of small samples** as a basis for inference." (Hesterberg, 2015, S. 379) Einerseits ist die *tatsächliche* Unsicherheit einer Parameterschätzung bei einer kleinen Stichprobe natürlich grösser als bei einer grösseren (siehe auch den zentralen Grenzwertsatz). Aber andererseits ist unsere *Schätzung* dieser Unsicherheit bei kleineren Stichproben auch weniger genau als bei grösseren. Dies ist aber nicht sosehr ein Nachteil des Bootstraps, sondern von kleinen Stichproben im Allgemeinen: Andere Verfahren bieten hier keine bessere Lösung.[4]
- Die Implementierung des Bootstraps, die oben illustriert wurde, tendiert dazu, die Unsicherheit einer Parameterschätzung eher zu unter- als zu überschätzen. Dies ist umso mehr der Fall bei kleinen Stichproben. Der Grund dafür ist, dass eine Stichprobe die Streuung in der Population eher unter- als überschätzt; deswegen wird die Varianz einer Stichprobe ja leicht anders berechnet als jene einer Population (siehe Abschnitt 5.2.2 auf Seite 58). Beim Bootstrap tritt die Stichprobe aber stellvertretend für die Population auf. Insofern die Stichprobe die Streuung in der Population unterschätzt, unterschätzt der Bootstrap die Unsicherheit der Parameterschätzung. Es gibt ein paar Möglichkeiten, diese Verzerrung zu korrigieren (siehe Efron & Tibshirani, 1993), aber pädagogisch sind diese hier nicht so interessant.
- Da der Bootstrap so flexibel ist, ist es schwierig, eine allgemeine, benutzerfreundliche Funktion für ihn zu schreiben. Daher muss man den Bootstrap meistens selber programmieren.

6.3.5 Übungen

Die obigen Beispiele dienten nur einem pädagogischen Zweck: Wenn wir eine Stichprobe von 76 Versuchspersonen haben, ist es ja kaum sinnvoll, kleinere Stichprobe aus ihr zu ziehen. Stattdessen werden hier zuerst die Befehle gezeigt, mit denen Sie die Unsicherheit von DeKeyser et al.'s ursprunglichem Stichprobenmittel schätzen können. Dann folgen zwei Übungen, die die Flexibilität des Bootstrap illustrieren.

Übung 1: Mittel Der erste Codeabschnitt definiert, wie viele bootstrap replicates generiert werden sollen, generiert diese dann anhand eines *for-loops* und berechnet jeweils das Mittel. Zur Verwendung von *for-loops*, siehe Seite 57.

[4] Ausser sie machen striktere Annahmen oder sie berücksichtigen Informationen, die man nicht aus den Daten selber ableiten kann.

```
# Hesterberg (2015) empfiehlt, 20'000 bootstrap
# replicates zu generieren, sodass das Ergebnis nur
# minimal vom Zufallsfaktor im Bootstrap selber
# beeinflusst wird. Um eine grobe Idee zu erhalten,
# würden 1'000 oder so auch reichen, aber im Prinzip
# sollte es nicht sehr lange dauern.
n_bootstraps <- 20000
bootstraps <- vector(length = n_bootstraps)
```

```
for (i in 1:n_bootstraps) {
  # Sampling with replacement aus der beobachteten Stichprobe.
  # (Daher 'replace = TRUE'.)
  bootstrap_sample <- d %>%
    sample_frac(replace = TRUE)

  # Das Mittel des bootstrap replicates berechnen und speichern.
  bootstraps[i] <- mean(bootstrap_sample$GJT)
}
```

Ein schnelles Histogramm (ohne `ggplot2`) zeigt die Wirkung des zentralen Grenzwertsatzes.

```
# Histogramm der Verteilung der bootstrapten Mittel zeichnen.
# (Hier nicht gezeigt.)
hist(bootstraps)
```

Als Schätzung des Standardfehlers dient die Standardabweichung der bootstrapten Mittel.

```
# Standardfehler des Mittels schätzen.
sd(bootstraps)
```

```
## [1] 3.1194
```

Etwa 95% der bootstrapten Mittel liegen zwischen 145 und 157. Dieses Intervall nennt man übrigens ein **Konfidenzintervall**, aber darüber später mehr.

```
quantile(bootstraps, probs = c(0.025, 0.975))
```

```
##   2.5%  97.5%
## 144.72 156.88
```

Da die Verteilung der bootstrapten Mittel normalverteilt, können wir diese Perzentile auch mithilfe der Eigenschaften von Normalverteilungen berechnen. Das 2.5. Perzentil jeder Normalverteilung liegt etwa 1.96 Standardabweichungen unter dem Mittel:

```
qnorm(0.025)
```

```
## [1] -1.96
```

Und das 97.5. Perzentil liegt genauso weit über dem Mittel:

```
qnorm(0.975)
```

```
## [1] 1.96
```

Diese Berechnungsmethode ergibt daher grundsätzlich die gleiche Lösung:

```
mean(d$GJT) - 1.96 * sd(bootstraps)
```

```
## [1] 144.66
```

```
mean(d$GJT) + 1.96 * sd(bootstraps)
```

```
## [1] 156.89
```

Dies gilt natürlich nur, wenn die Verteilung der bootstrapten Mittel normalverteilt ist; die Perzentilmethode ist allgemeiner gültig.

Verglichen mit den Angaben in Tabelle 6.1 sind der geschätzte Standardfehler und die Breite des Intervalls kleiner. Können Sie sich erklären, wieso?

Übung 2: Standardabweichung Der Bootstrap ist nicht nur nützlich, um die Unsicherheit in der Schätzung eines Mittels zu quantifizieren. Berechnen Sie die Standardabweichung der GJT-Daten und verwenden Sie den Bootstrap, um die Unsicherheit in dieser Schätzung zu quantifizieren.

Übung 3: Median Wie Übung 2, aber mit dem Median statt der Standardabweichung. Was fällt Ihnen auf?[5]

6.4 Das *plug-in*-Prinzip und der zentrale Grenzwertsatz

Aus überwiegend historischen Gründen wird der Bootstrap eher selten verwendet, um die Unsicherheit eines Stichprobenmittel zu quantifizieren. Stattdessen verlässt man sich meistens auf den zentralen Grenzwertsatz (siehe Abschnitt 5.3 auf Seite 60).

Zur Erinnerung: Der zentrale Grenzwertsatz besagt, dass Stichprobenmittel ($\bar{x}$) zu einer Normalverteilung neigen, wenn die Stichproben gross genug sind. Das Mittel der Stichprobenmittelverteilung ist gleich dem Populationsmittel ($\mu_{\bar{x}} = \mu$); ihre Standardabweichung (der Standardfehler) beträgt:

$$\text{SE} = \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{6.1}$$

[5] Wenn Ihnen nichts auffällt, sollten Sie die Anzahl *bins* im Histogramm vergrössern: `hist(bootstraps, breaks = 100)`

Wenn wir (a) annehmen wollen, dass der zentrale Grenzwertsatz bereits bei unserer Stichprobengrösse zutrifft, und (b) die Standardabweichung der Population, aus der unsere Stichprobe stammt, kennen, können wir den Standardfehler also direkt berechnen.

Wenn die Standardabweichung der Population uns nicht bekannt ist, können wir wieder das *plug-in*-Prinzip anwenden: Die Stichprobenstandardabweichung s ist die beste Schätzung der Populationsstandardabweichung σ, die wir haben, weshalb wir diese Schätzung stellvertretend in die Formel eintragen:

$$\text{SE} \approx \widehat{\text{SE}} = \frac{s}{\sqrt{n}} \tag{6.2}$$

Bei den GJT-Daten beträgt die Stichprobenstandardabweichung etwa 27.23. Daher beträgt der geschätzte Standardfehler $\frac{27.32}{\sqrt{76}} = 3.13$. Dies ist nahezu die gleiche Antwort, die wir mit dem Bootstrap bekamen (3.12), was natürlich beruhigend ist.

Anhand des zentralen Grenzwertsatzes können wir auch ein 95%-Konfidenzintervall konstruieren:

```
mean(d$GJT) - 1.96 * sd(d$GJT)/sqrt(76)
```

```
## [1] 144.63
```

```
mean(d$GJT) + 1.96 * sd(d$GJT)/sqrt(76)
```

```
## [1] 156.92
```

Auch das Konfidenzintervall ist dem Konfidenzintervall, das mit dem Bootstrap konstruiert wurde, sehr ähnlich.

Bemerken Sie aber, dass wir diesmal davon *ausgegangen* sind, dass die Stichprobenmittel aus der Population normalverteilt sind; diese Annahme haben wir beim Bootstrap nicht gemacht. Bei sehr schiefen oder anderen asymmetrischen Verteilungen ist es durchaus möglich, dass der zentrale Grenzwertsatz auch bei Stichproben von 76 Beobachtungen noch nicht zutrifft. Wenn dieser Verdacht besteht, wäre der Bootstrap also geeigneter. Ausserdem gilt der zentrale Grenzwertsatz nur für das Mittel, nicht für andere Parameterschätzungen. Für ein paar Parameterschätzungen gibt es andere Formeln, aber der Bootstrap ist wesentlich flexibler.

6.5 Die t-Verteilungen

Die Stichprobenstandardabweichung (s) ist bloss eine Schätzung der Populationsstandardabweichung (σ). Insofern s σ unterschätzt, wird die Unsicherheit in der Parameterschätzung unterschätzt; überschätzt s σ, dann wird die Unsicherheit in der Parameterschätzung überschätzt. Damit könnte man sich abfinden, wenn Unter- und Überschätzungen gleich oft vorkämen. Versteckt in Fussnote 1 auf Seite 59 taucht aber ein

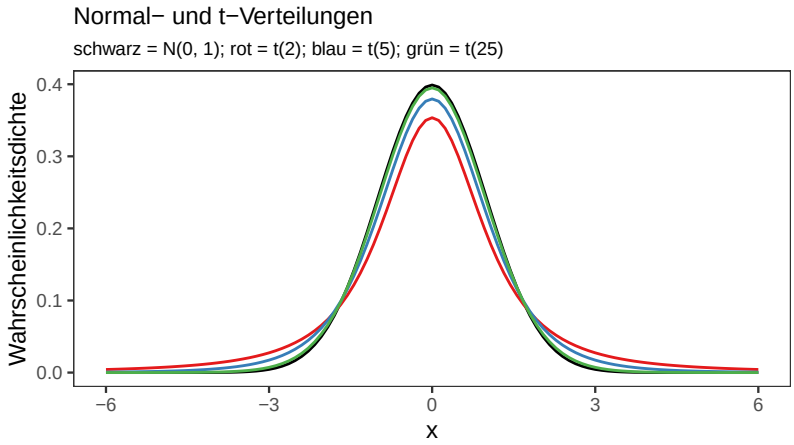


Abbildung 6.6: Eine Standardnormalverteilung und ein paar t-Verteilungen.

Problem auf: s tendiert dazu, σ zu unterschätzen, insbesondere bei kleinen Stichproben. Entsprechend ist die Stichprobenmittelverteilung eher breiter als schmaler als eine Normalverteilung mit $\frac{s}{\sqrt{n}}$ als Standardabweichung.

Meistens ist es unmöglich, s für diese Verzerrung zu korrigieren. Die Ausnahme ist, wenn die Stichprobe *aus einer normalverteilten Population* stammt. Statt die Standardabweichung und den geschätzten Standardfehler direkt zu korrigieren, wird in diesem Fall die geschätzte Stichprobenmittelverteilung angepasst, indem sie breiter gemacht wird (mehr für kleinere Stichproben). Diese breiteren Normalverteilungen nennt man t-Verteilungen; Abbildung 6.6 zeigt ein paar Beispiele.

t-Verteilungen haben einen Parameter, den man ihre Freiheitsgrade nennt. Beim Schätzen eines Mittels ist die Anzahl Freiheitsgrade einfach die Anzahl Beobachtungen minus 1: Gibt es 76 Beobachtungen, gibt es 75 Freiheitsgrade. Abbildung 6.6 zeigt t-Verteilungen mit 2, 5 und 25 Freiheitsgraden. Je mehr Freiheitsgrade, desto ähnlicher ist die Verteilung einer Standardnormalverteilung (d.h., eine Normalverteilung mit einer Standardabweichung von 1). Dies entspricht der Tatsache, dass grössere Stichproben σ tendenziell weniger unterschätzen als kleinere Stichproben.

Um das 95%-Konfidenzintervall um ein Stichprobenmittel mithilfe der t-Verteilungen zu finden, sucht man zuerst das 2.5. und das 97.5. Perzentil der t-Verteilung mit $n - 1$ Freiheitsgraden (hier: $76 - 1 = 75$). Dann multipliziert man den geschätzten Standardfehler mit diesen Perzentilen. Dies ist komplett analog zur Berechnung auf der Basis des zentralen Grenzwertsatzes, nur wird mit einer t- statt einer Normalverteilung gearbeitet.

```
qt(0.025, df = 75)
```

```
## [1] -1.9921
```

```
qt(0.975, df = 75)
```

```
## [1] 1.9921
```

```r
mean(d$GJT) - 1.99 * sd(d$GJT) / sqrt(76)
```

```
## [1] 144.54
```

```r
mean(d$GJT) + 1.99 * sd(d$GJT) / sqrt(76)
```

```
## [1] 157.01
```

In diesem Beispiel ergeben alle Berechungsmethoden ein recht ähnliches Ergebnis. Insbesondere bei kleinen Stichproben oder bei Stichproben, die den Verdacht nahelegen, dass die Population sehr schräg verteilt ist, ist dies aber nicht unbedingt der Fall.

Bemerken Sie, dass von den drei Methoden, die t-Methode die meisten Annahmen macht: Sie geht nicht nur davon aus, dass man anhand der Stichprobe sinnvolle Aussagen über die Unsicherheit machen kann und dass die Population irgendwelche Verteilung hat, für die den zentralen Grenzwertsatz bei dieser Stichprobengrösse zutrifft, sondern auch, dass die Population *selber* normalverteilt ist. Wenn all diese Annahmen aber stimmen, ist diese Methode auch die genauste. Der Bootstrap dahingegen ist sozusagen das Schweizer Sackmesser unter den Schätzungsmethoden: Er kann in vielen Situationen angewandt werden, aber je nach Situation gibt es spezialisierte Methoden, die schon besser funktionieren.

6.6 Konfidenzintervalle

Im Laufe dieses Kapitels haben wir ein paar Konfidenzintervalle konstruiert, sodass es nun die höchste Zeit ist, zu erklären, was diese überhaupt sind. Eine leider schwierige Definition ist die folgende:

> Ein α%-Konfidenzintervall um ein Stichprobenmittel besteht aus zwei Werten, die um dieses Stichprobenmittel liegen und die nach einem bestimmten Verfahren so gewählt wurden, dass das Intervall das wahre Populationsmittel (μ) in α% der Fälle enthält.

Zum Beispiel werden 95%-Konfidenzintervalle nach einem Verfahren konstruiert, das garantieren soll, dass wenn man eine grosse Anzahl Stichproben aus der Population zieht und für jede Stichprobe das Intervall berechnet, 95% der Intervalle μ enthalten.

Wie diese Definition zeigt, ist das Konzept schwieriger als was man auf den ersten Blick denken würde – auch für erfahrene Forschende (Hoekstra et al., 2014). Oft interpretiert man ein 95%-Konfidenzintervall als jene zwei Werte, zwischen denen der Populationsparameter (hier: μ) mit 95% Wahrscheinlichkeit liegt. Dies stimmt aber nicht (Morey et al., 2016). Nichtsdestoweniger schreibt Ehrenberg (1982) zur Interpretation von Konfidenzintervallen Folgendes:

> [T]he rough-and-ready interpretation of confidence limits ... will be close to the truth. The choice is between making a statement which is true but so complex that it is almost unactionable, and making one which is much

> simpler but not quite correct. Fortunately, the effective content of the two kinds of statement is generally similar. (S. 125)

Statt Konfidenzintervallen empfehlen Morey et al. (2016) den Gebrauch von 'Kredibilitätsintervallen'. Diese sind in der bayesschen Statistik angesiedelt und kommen momentan in unserer Forschungsliteratur kaum vor, weshalb ich sie hier nicht bespreche; siehe aber Abschnitt 18.4 auf Seite 277 für Literaturempfehlungen. Albers et al. (2018) bemerken, dass Konfidenz- und Kredibilitätsintervalle einander üblicherweise sehr ähnlich sind; Nalborczyk et al. (2018) ziehen diese Schlussfolgerung aber in Frage.

Dieser Bemerkung zum Trotz sind meines Erachtens insbesondere die folgenden Punkte wichtig:

- Konfidenzintervalle heben hervor, dass Schätzungen inhärent unsicher sind.
- Bei grossen Stichproben oder bei Stichproben aus Populationen, in denen es wenig Variation gibt, sind Konfidenzintervalle tendenziell schmaler.
- Rein durch Zufall kann eine Stichprobe die Streuung in der Population unterschätzen und daher kann das Konfidenzintervall die Unsicherheit in der Schätzung ebenso unterschätzen.
- Genauere Unsicherheitseinschätzungen erhält man mit grösseren Stichproben oder indem man weitere nützliche Annahmen über die Daten macht (wie in der bayesschen Statistik).

Unter `http://rpsychologist.com/d3/CI/` finden Sie eine lehrreiche App zu Konfidenzintervallen. Unter anderem zeigt die App, dass Konfidenzintervalle manchmal schmal sein können, aber die Schätzung trotzdem weit vom Populationswert entfernt liegen kann.

6.7 Denkaufgaben

Es kommt höchst selten vor, dass man das Mittel, den Median oder die Standardabweichung (usw.) einer Population schätzen muss. Stattdessen schätzt man in der Regel Unterschiede zwischen Gruppen oder Zusammenhänge zwischen Variablen. Für solche Fälle werden die gleichen Prinzipien wie jene in diesem Kapitel zutreffen, aber es scheint mir Beschäftigungstherapie zu sein, weitere praktische Aufgaben für dieses Kapitel zu erledigen. Stattdessen folgen hier ein paar Denkaufgaben.

1. Welche Faktoren bestimmen die tatsächliche Unsicherheit einer Parameterschätzung (z.B. eines Mittels)?
2. Wie könnte man als ForscherIn die Unsicherheit bei der Schätzung verringern?
3. Zwei Stichproben haben identische Stichprobenstandardabweichungen: $s_1 = s_2$. Stichprobe 1 besteht aus 16 Datenpunkten; Stichprobe 2 aus nur vier. Aus welchen

zwei Gründen wird das 95%-Konfidenzintervall bei Stichprobe 1 schmaler sein als bei Stichprobe 2, wenn Sie diese Intervalle mit t-Verteilungen konstruieren?

4. Beim Arbeiten mit t-Verteilungen geht man davon aus, dass die Population, aus der die Stichprobe stammt, normalverteilt ist. Warum?

5. Warum ist diese Normalitätsannahme weniger wichtig bei grösseren Stichproben?

Teil III

Das allgemeine lineare Modell

Kapitel 7

Ein anderer Blick aufs Mittel

In diesem und den folgenden Kapiteln behandeln wir das sog. **allgemeine lineare Modell** (*general linear model*).[1] Das allgemeine lineare Modell ist eine Methode, um zu ausdrücken, wie ein oder mehrere **Prädiktoren** mit dem ***outcome*** zusammenhängen. (Oft redet man statt von Prädiktoren und *outcomes* von unabhängigen bzw. abhängigen Variablen, aber ich finde die Begriffe Prädiktor und *outcome* deutlicher.)

In diesem Kapitel werden einige Schlüsselkonzepte des allgemeinen linearen Modells erläutert, indem wir das Mittel einer Population auf eine andere Art und Weise schätzen als wir es bisher gemacht haben. In den darauf folgenden Kapiteln werden die Modelle graduell komplexer, aber die Basisprinzipien aus diesem Kapitel werden noch immer zutreffen.

7.1 Ein Modell für die GJT-Daten

Lasst uns kurz alles über Mittelwerte vergessen. Wir erhalten Daten (hier: die GJT-Werte von DeKeyser et al. (2010), siehe letztes Kapitel) und müssen diese sinnvoll beschreiben. Sinnvoller als einfach alle Datenpunkte aufzulisten, wäre, die Datenpunkte in zwei Teile zu zerlegen: einen Teil, der die Gemeinsamkeiten zwischen allen Werten ausdrückt, und einen Teil, der die individuelle Unterschiede zwischen diesen Gemeinsamkeiten und den Werten ausdrückt:

$$\text{Wert einer Versuchsperson} = \text{Gemeinsamkeit} + \text{Abweichung} \tag{7.1}$$

Um die Notation übersichtlich zu halten, wird diese Gleichung meistens so geschrieben:

[1] Nicht zu verwechseln mit dem **verallgemeinerten linearen Modell** (*generalized linear model*). Dieses ist eine Erweiterung des allgemeinen linearen Modells, mit der wir uns in diesem Kurs nicht befassen. Siehe aber Kapitel 18.

$$y_i = \beta_0 + \varepsilon_i \tag{7.2}$$

y_i ist die i. Beobachtung im Datensatz, β_0 stellt die Gemeinsamkeit zwischen allen Werten in der Population dar, und ε_i drückt aus, wie stark die i. Beobachtung von diesem Populationswert abweicht. ε_i nennt man auch die **Residuen** oder den **Restfehler**. (Nachher werden wir die Gemeinsamkeit zwischen den y-Werten mithilfe mehrerer βs ausdrücken, daher die $_0$.)

In der Regel interessieren wir uns am stärksten für β_0. Da uns aber nicht die ganze Population zur Verfügung steht, müssen wir uns mit einer Schätzung von β_0 begnügen. In Gleichung 7.2 ist β_0 ein Parameter mit einem bestimmten, aber in der Regel unbekannten Wert; für Schätzungen dieses Parameters wird die Notation $\widehat{\beta_0}$ verwendet. Da $\widehat{\beta_0}$ bloss eine Schätzung ist, wird der Restfehler ebenfalls bloss geschätzt.

$$y_i = \widehat{\beta_0} + \widehat{\varepsilon_i} \tag{7.3}$$

Wie können wir nun $\widehat{\beta_0}$ berechnen (bzw. β_0 schätzen)? Im Prinzip geht die Gleichung für jeden β_0-Wert auf. Die ersten beiden GJT-Werte im Datensatz sind 151 und 182. Wenn wir für β_0 einen beliebigen Wert, z.B. 1823, wählen, gilt $\widehat{\varepsilon_1} = -1672$ und $\widehat{\varepsilon_2} = -1641$ und die Gleichung geht auf:

$$y_1 = 151 = 1823 - 1672$$

$$y_2 = 182 = 1823 - 1641$$

Wenn wir für β_0 den Wert 14 wählen, gilt $\widehat{\varepsilon_1} = 137$ und $\widehat{\varepsilon_2} = 168$ und die Gleichung geht wiederum auf:

$$y_1 = 151 = 14 + 137$$

$$y_2 = 182 = 14 + 168$$

Wir brauchen also eine prinzipielle Methode, um β_0 zu schätzen.

7.2 Die Methode der kleinsten Quadrate

Die Frage stellt sich, was die optimale Art und Weise ist, um $\widehat{\beta_0}$ zu berechnen. Die wenig überraschende Antwort lautet: Es hängt davon ab, was man unter 'optimal' versteht.

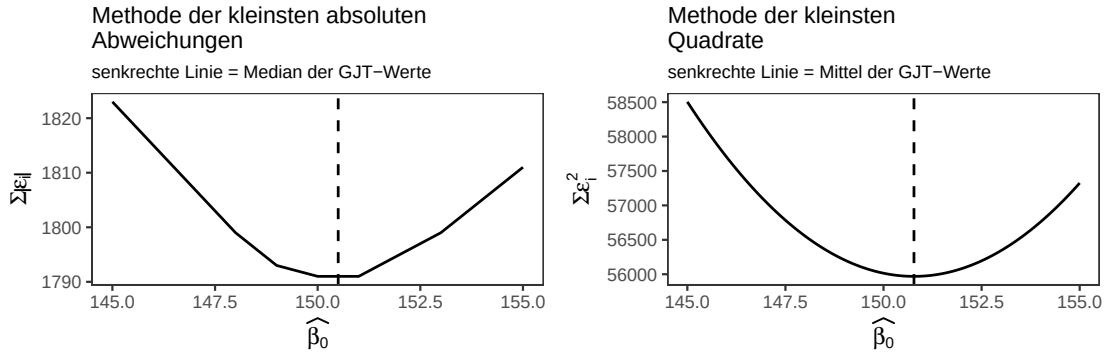


Abbildung 7.1: Links: Wenn der Parameter mit der Methode der kleinsten absoluten Abweichungen geschätzt wird, ist die Lösung gleich dem Median der Stichprobe. Rechts: Wenn der Parameter mit der Methode der kleinsten Quadrate geschätzt wird, ist die Lösung gleich dem Mittel der Stichprobe.

Eine sinnvolle Definition von 'optimal' ist, dass β_0 so geschätzt werden soll, dass die Summe der absoluten Restfehler ($\sum_{i=1}^{n} |\widehat{\varepsilon_i}|$) möglichst klein ist. Wenn wir für $\widehat{\beta_0}$ den Wert 135 wählen, beträgt die Summe der absoluten Restfehler 1993:

```
sum(abs(d$GJT - 135))
```

```
## [1] 1993
```

(Bemerken Sie, dass $y_i = \widehat{\beta_0} + \widehat{\varepsilon_i}$, also $\widehat{\varepsilon_i} = y_i - \widehat{\beta_0}$.)

Wenn wir stattdessen den Wert 148 wählen, beträgt die Summe der absoluten Restfehler 1799.

```
sum(abs(d$GJT - 148))
```

```
## [1] 1799
```

Wenn wir 'optimal' so definieren, ist 148 also die bessere Schätzung von β_0. Diese Übung können wir für jede Menge Kandidatwerte durchprobieren und dann den optimalen Wert wählen. Dies ist die **Methode der kleinsten absoluten Abweichungen**. Wie Abbildung 7.1 (links) zeigt, sind β_0-Schätzungen zwischen 150 und 151 in diesem Sinne optimal. Der Median der GJT-Werte ist nicht zufällig 150.5: Wenn man β_0 mit der Methode der kleinsten absoluten Abweichungen schätzt, ist das Ergebnis der Median der Stichprobe. Dass wir es hier mit einer *Schätzung* zu tun haben, wird klar, wenn man sich überlegt, dass diese Methode ein anderes Ergebnis liefern könnte, wenn man eine neue Stichprobe aus der gleichen Population zieht.

Eine andere sinnvolle Definition von 'optimal' ist, dass β_0 so geschätzt werden soll, dass die Summe der quadrierten Restfehler ($\sum_{i=1}^{n} \widehat{\varepsilon_i}^2$) möglichst klein ist. Dies ist die **Methode der kleinsten Quadrate**. Verglichen mit der Methode der kleinsten absoluten Abweichungen werden grosse Residuen mehr bestraft. Anders gesagt: Grosse Abweichungen (darunter auch Ausreisser) üben einen stärkeren Einfluss auf das Ergebnis aus. Wie Abbildung 7.1 (rechts) zeigt, ist die optimal β_0-Schätzung laut der Methode

der kleinsten Quadrate 150.78—nicht zufällig das Mittel der Stichprobe!

Während wir in Kapitel 3 die Summe der Quadrate als eine Funktion des Mittels betrachtet haben, kann man die Rollen auch umkehren: Das Mittel ist jene Wert, der die Summe der Quadrate minimiert. Ebenso ist der Median jene Wert, der die Summe der absoluten Abweichungen minimiert.[2]

Algebraisch ist die Methode der kleinsten Quadrate am einfachsten, und die Parameter in allgemeinen linearen Modellen werden daher meistens mit dieser Methode geschätzt (*ordinary least squares*, OLS). Aber dies ist keine Notwendigkeit. In bestimmten Bereichen trifft man ab und zu andere Optimierungskriterien an; in den Sprachwissenschaften ist dies aber eher selten der Fall. Im Prinzip kann man sogar selber beliebige Optimierungskriterien definieren, aber dies kommt noch weniger vor.

7.3 Lineare Modelle in R

Mit der `lm()`-Funktion können lineare Modelle aufgebaut werden. Ihre Parameter werden anhand der Methode der kleinsten Quadrate geschätzt. Innerhalb der Funktion braucht es eine Formel mit dem *outcome* vor und den Prädiktoren nach der Tilde. In diesem Fall gibt es keinen Prädiktor, stattdessen wird `1` verwendet.

```
mod.lm <- lm(GJT ~ 1, data = d)
```

Die geschätzten βs kann man abrufen, indem man den Namen des Modells (hier: `mod.lm`) eintippt.

```
mod.lm
```

```
##
## Call:
## lm(formula = GJT ~ 1, data = d)
##
## Coefficients:
## (Intercept)
##       150.8
```

Auch mit `coef()` erhält man die β-Schätzungen (hier nur β_0).

```
coef(mod.lm)
```

```
## (Intercept)
##    150.7763
```

Mit `predict()` erhält man einen Vektor mit n Werten (hier: $n = 76$), der die 'vorhergesagten' y-Werte enthält. (Ich mag den Begriff 'vorhergesagt' hier nicht.) Es handelt sich

[2] Der Modus ist übrigens der Wert, der die Summe der binären Abweichungen minimiert. Sind y_i und $\widehat{\beta_0}$ einander gleich, beträgt die binäre Abweichung 0, sonst 1.

um die y-Werte *abzüglich* der $\hat{\varepsilon}$-Werte: $\widehat{y_i} = y_i - \widehat{\varepsilon_i}$. In unserem Fall sind dies lediglich 76 Wiederholungen von $\widehat{\beta_0}$.[3]

```
predict(mod.lm)
```

```
##        1        2        3        4        5        6
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##        7        8        9       10       11       12
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       13       14       15       16       17       18
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       19       20       21       22       23       24
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       25       26       27       28       29       30
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       31       32       33       34       35       36
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       37       38       39       40       41       42
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       43       44       45       46       47       48
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       49       50       51       52       53       54
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       55       56       57       58       59       60
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       61       62       63       64       65       66
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       67       68       69       70       71       72
## 150.7763 150.7763 150.7763 150.7763 150.7763 150.7763
##       73       74       75       76
## 150.7763 150.7763 150.7763 150.7763
```

Die Residuen kann man mit der `resid()`-Funktion abfragen.

```
residuen <- resid(mod.lm)
```

7.4 Unsicherheit in einem allgemeinen linearen Modell quantifizieren

7.4.1 Der Bootstrap

Genauso wie im letzten Kapitel können wir den Bootstrap verwenden, um die Unsicherheit in der Schätzung von β_0 zu quantifizieren. Da in diesem Fall $\widehat{\beta_0} = \bar{x}$, ergibt

[3] $y_i = \widehat{\beta_0} + \widehat{\varepsilon_i}$, also $\widehat{\varepsilon_i} = y_i - \widehat{\beta_0}$, also $\widehat{y_i} = y_i - \widehat{\varepsilon_i} = y_i - (y_i - \widehat{\beta_0}) = \widehat{\beta_0}$.

dies natürlich die gleiche Lösung wie vorher. Aber es gibt mir die Gelegenheit, zu zeigen, wie man auch für komplexere lineare Modelle den Bootstrap verwenden kann.

Die Logik ist wiederum, dass wir die Stichprobe stellvertretend für die Population einsetzen. Aber anstatt Bootstrap-Stichproben aus der Stichprobe zu ziehen, ziehen wir nur Bootstrap-Stichproben aus den Residuen (ε). Diese kombinieren wir dann mit $\widehat{\beta_0}$, um die Bootstrap-Stichproben zu kombinieren. Wenn wir uns nur fürs Mittel interessieren, hat diese Methode überhaupt keinen Mehrwert, denn sie ist mathematisch der zuerst besprochenen Bootstrap-Methode gleich. Aber sie ist pädagogisch wertvoller (und manchmal auch statistisch besser), wenn wir später mehrere βs haben werden.

Konkret:

1. Man berechnet $\widehat{\beta_0}$ und erhält dazu auch noch $\hat{\varepsilon}$.
2. Man zieht eine Bootstrap-Stichprobe aus $\hat{\varepsilon}$ (*sampling with replacement*). Diese kann man als $\hat{\varepsilon}^*$ bezeichnen.
3. Man kombiniert $\widehat{\beta_0}$ und $\hat{\varepsilon}^*$. Dies ergibt eine neue Reihe von y-Werten: $y_i^* = \widehat{\beta_0} + \widehat{\varepsilon_i}^*$.
4. Man schätzt nun auf der Basis von y^* erneut den Parameter von Interesse ($\widehat{\beta_0}^*$).
5. Man führt Schritte 2–4 ein paar tausend Mal aus und erhält so die Verteilung der bootstrapten β_0-Schätzungen.

Keine Panik! Mathematische Notation verunsichert manchmal, aber so wichtig ist sie nicht. Sie ist vor allem nützlich, um Schritte wie jene oben möglichst eindeutig zu beschreiben. Die Schritte sind wichtig; wie man sie dann genau notiert nicht.

Jetzt in R-Code.

```
n_bootstrap <- 20000
bs_b0 <- vector(length = n_bootstrap)

for (i in 1:n_bootstrap) {
  # Residuen bootstrappen
  bs_residual <- sample(resid(mod.lm), replace = TRUE)

  # neuen Outcome kreieren:
  bs_outcome <- predict(mod.lm) + bs_residual

  # Modell neu berechnen mit diesem Outcome
  bs_mod <- lm(bs_outcome ~ 1)

  # Schätzung speichern
  bs_b0[i] <- coef(bs_mod)[1]
}
```

Wir können jetzt wieder die Verteilung der bootstrapten Parameterschätzungen visualisieren, und ihre Standardabweichung und 2.5. und 97.5. Perzentile berechnen. Die Ergebnisse sind natürlich identisch mit jenen aus Kapitel 6.

```
# Das Histogramm wird hier nicht gezeigt.
hist(bs_b0)
```

```
sd(bs_b0)
```

```
## [1] 3.124557
```

```
quantile(bs_b0, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 144.6974 156.9605
```

7.4.2 Ein anderer Bootstrap

Beim Bootstrap, den wir soeben besprochen haben, sind wir davon ausgegangen, dass die Residuen in der Stichprobe genau so verteilt sind wie die Residuen in der Population. Ein Nachteil dieser Annahme ist, dass wir dadurch die Feinkörnigkeit der Residuen in der Population wohl unterschätzen. Beispielsweise gibt es für das `mod.lm`-Modell ein Residuum von -14.78 und ein Residuum von -12.78, aber keines von -13.78. Ein Residuum von -14.78 entspricht einer Beobachtung von $150.78-14.78 = 136$; ein Residuum von -13.78 entspräche einer Beobachtung von $150.78 - 13.78 = 137$. Unserer Annahme zufolge gäbe es in der Population also keine Versuchspersonen mit einem GJT-Ergebnis von 137.

Dies ist natürlich eine etwas komische Annahme; in der Regel hat sie aber kaum einen Einfluss auf die Inferenzen. Aber eine Alternative wäre, dass wir davon ausgingen, dass die Residuen in der Population normalverteilt sind. Normalverteilungen sind unendlich feinkörnig, sodass diese Annahme sozusagen den Gegenpol der ersten Annahme darstellt. Das Mittel der Residuen beträgt 0, sodass wir nur die Standardabweichung der normalverteilten Restfehler in der Population finden müssen. Diese wird durch die Standardabweichung der Restfehler in der Stichprobe *geschätzt*. Dafür können wir hier zwei Funktionen verwenden:

```
sd(resid(mod.lm))
```

```
## [1] 27.31769
```

```
sigma(mod.lm)
```

```
## [1] 27.31769
```

In diesem Beispiel (lineares Modell ohne Prädiktoren) sind diese Werte identisch, aber sobald Prädiktoren im Spiel sind, liefert `sigma()` die bessere Schätzung der Standardabweichung der Residuen. Sie wird so berechnet (p ist die Anzahl geschätzten βs):

$$\widehat{\sigma_\varepsilon} = \sqrt{\frac{1}{n-p}\sum_{i=1}^{n}\widehat{\varepsilon_i}} \tag{7.4}$$

In diesem Fall ist $p = 1$, sodass die Gleichung die Stichprobenstandardabweichung ergibt. (Hier teilt man durch $n-p$ aus dem gleichen Grund, weshalb man bei der Standardabweichung der Beobachtungen in der einer Stichprobe durch $n-1$ teilt: Um eine systematische Unterschätzung zum grössten Teil entgegenzuwirken.)

Anstatt für jede Bootstrapstichprobe die Residuen durch *sampling with replacement* zu generieren, werden sie hier zufällig aus einer Normalverteilung mit $\mu = 0$ und $\sigma = \widehat{\sigma_\varepsilon}$ generiert:

```
n_bootstrap <- 20000
bs_b0 <- vector(length = n_bootstrap)

for (i in 1:n_bootstrap) {
  # Neue Residuen generieren, und zwar aus einer Normalverteilung mit
  # der geschätzten Standardabweichung der Residuen
  bs_residual <- rnorm(n = 76, sd = sigma(mod.lm))

  # neuen Outcome kreieren:
  bs_outcome <- predict(mod.lm) + bs_residual

  # Modell neu berechnen mit diesem Outcome
  bs_mod <- lm(bs_outcome ~ 1)

  # Schätzung speichern
  bs_b0[i] <- coef(bs_mod)[1]
}
```

```
# Histogramm: nicht gezeigt
hist(bs_b0)
```

```
# Geschätzter Standardfehler
sd(bs_b0)
```

```
## [1] 3.131636
```

```
# 95% Konfidenzintervall
quantile(bs_b0, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 144.6208 156.9098
```

Diese Art von Bootstrap – bei der wir davon ausgehen, dass die Residuen eine bestimmte Verteilung haben, und wir die relevanten Parameter dieser Verteilung anhand der Stichprobe schätzen – nennt man einen parametrischen Bootstrap. Der Bootstrap

aus dem letzten Abschnitt – bei der man Bootstrapstichproben der Modellresiduen mit den Modellvorhersagen kombiniert und die relevanten Parameter anhand dieser neuen Werte schätzt – nennt man einen semiparametrischen Bootstrap. Der Bootstrap aus dem letzten Kapitel – bei der man Bootstrapstichproben aus dem ursprunglichen Datensatz generiert – nennt man einen nichtparametrischen Bootstrap. Siehe auch *Some illustrations of bootstrapping* (20.12.2016).

7.4.3 Mit t-Verteilungen

Wenn man ohnehin davon ausgeht, dass die Residuen normalverteilt sind, können der geschätzte Standardfehler und das Konfidenzintervall auch algebraisch abgeleitet werden. Die Formeln, die man dazu braucht, werden hier nicht gezeigt, denn sie haben kaum einen didaktischen Mehrwert.

In R kann man die `summary()`-Funktion verwenden, um den geschätzten Standardfehler zu berechnen (`Std. Error`):

```
summary(mod.lm)
```

```
##
## Call:
## lm(formula = GJT ~ 1, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.776 -23.026  -0.276  23.224  47.224
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  150.776      3.134   48.12   <2e-16
##
## Residual standard error: 27.32 on 75 degrees of freedom
```

β_0 heisst hier `(Intercept)`. Was `t value` und `Pr(>|t|)` bedeuten, werden wir erst in einem späteren Kapitel besprechen.

Das 95%-Konfidenzintervall kann anhand von `confint()` berechnet werden.

```
confint(mod.lm)
```

```
##                2.5 %   97.5 %
## (Intercept) 144.534 157.0187
```

Natürlich kann man auch gerne andere Intervalle berechnen, z.B. 80%-Konfidenzintervalle:

```
confint(mod.lm, level = 0.80)
```

```
##                 10 %     90 %
```

```
## (Intercept) 146.7248 154.8278
```

Denkfrage Warum wird bei der `summary()`-Funktion der Median aber nicht das Mittel der Residuen gezeigt?

7.5 Fazit

- Datenpunkte kann man als eine Kombination einer Gemeinsamkeit aller Datenpunkte in der Population und einer spezifischen Abweichung verstehen.
- In der Regel ist die Gemeinsamkeit von Interesse; diese wird aber anhand der Abweichungen und eines Optimierungskriteriums geschätzt.
- Das am meisten verwendetete Optimierungskriterium ist die Methode der kleinste Quadrate, die im 'univariaten' Fall (= wenn man nur mit einer Variablen arbeitet) das Mittel ergibt, aber andere Methoden existieren.
- Die Unsicherheit in der Parameterschätzung kann mithilfe des Bootstraps oder anhand weiterer Annahmen geschätzt werden.

Kapitel 8

Einen Prädiktor hinzufügen

In diesem Kapitel beschäftigen wir uns mit der Frage, wie der Zusammenhang zwischen einem einzigen kontinuierlichen Prädiktor und einem einzigen kontinuierlichen Outcome erfasst werden kann. Eine **kontinuierliche** Variable ist eine eher feinkörnige Variable, deren Werte geordnet werden können (z.B. von klein nach gross oder von schlecht nach gut). Ausserdem können Unterschiede auf der Skala sinnvoll miteinander verglichen werden: Der Unterschied zwischen 15 und 20 °C ist gleich dem Unterschied zwischen −10 und −5 °C, und beide Unterschiede sind halb so gross wie jener zwischen 50 und 60 °C. Diese Eigenschaften haben sowohl die GJT- als auch die AOA-Variable im Datensatz von DeKeyser et al. (2010).

In den letzten zwei Kapiteln haben wir uns mit der zentralen Tendenz der GJT-Daten beschäftigt. Eigentlich interessierten sich DeKeyser et al. (2010) aber nicht sosehr für diese, sondern für den Zusammenhang zwischen GJT und AOA. Diesen Zusammenhang schauen wir uns hier an. Wie immer ist es eine gute Idee, die Daten zunächst grafisch darzustellen. Wenn man sich für den Zusammenhang zwischen zwei kontinuierlichen Variablen interessiert, bietet sich das **Streudiagramm** (*scatterplot*) an; siehe Abbildung 8.1. Beim Zeichnen eines Streudiagramms muss man spezifizieren, welche Variable entlang der x-Achse und welche entlang der y-Achse dargestellt wird. Wenn es naheliegender ist, dass Variable A Variable B beeinflusst als umgekehrt, empfiehlt es sich, Variable A entlang der x-Achse darzustellen und Variable B entlang der y-Achse. Hier ist es unmöglich, dass die Grammatikalitätsurteile das Erwerbsalter beeinflussen, aber sehr wohl, dass das Erwerbsalter die Grammatikalitätsurteile beeinflussen.

```
ggplot(data = d,
       aes(x = AOA,
           y = GJT)) +
  # 'shape = 1' zeichnet leere Kreischen, die
  # ich übersichtlicher finde als gefüllte Punkte
  geom_point(shape = 1) +
  xlab("Erwerbsalter (Jahre)") +
  ylab("Ergebnis\nGrammatikalitätsurteile")
```

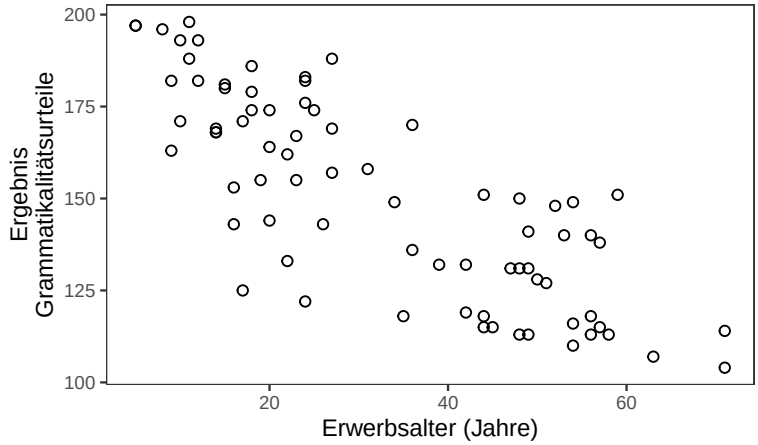


Abbildung 8.1: Zusammenhang zwischen AOA und GJT in der Studie von DeKeyser et al. (2010).

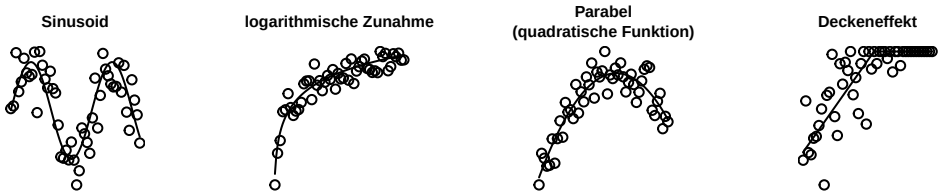


Abbildung 8.2: Beispiele von nicht-linearen Zusammenhängen.

Das Streudiagramm zeigt, dass die Leistung beim GJT mit steigendem Erwerbsalter allmählich abnimmt. Diese Senkung scheint auch ungefähr linear zu sein; zum Vergleich zeigt Abbildung 8.2 vier deutliche Beispiele von nicht-linearen Zusammenhängen. Ausserdem gibt es keine einzelnen Punkte, die sehr weit von der Punktwolke entfernt liegen, und alle Daten sind plausibel: Es gibt keine 207-Jährigen und in DeKeyser et al. (2010) kann man lesen, dass das GJT-Instrument aus 204 binären Items bestand. Das höchstmögliche Ergebnis von 204 wird hier nicht überschritten.

8.1 Zwei Fragen

Im Folgenden werden wir eine Antwort auf diese beiden Fragen geben:

1. *Wie stark* ist der Zusammenhang zwischen der GJT- und der AOA-Variable? Anders gesagt, wenn wir den Wert einer der beiden Variablen kennen, *wie gut* können wir dann den Wert der anderen Variablen schätzen? (**Korrelationsanalyse**)
2. *Was* ist der Zusammenhang zwischen den beiden Variablen? Anders gesagt, wenn wir den Wert einer Variablen kennen, *wie* können wir dann den Wert der anderen Variablen schätzen? (**Regressionsanalyse**)

Beide Fragen werden oft miteinander verwechselt, was manchmal zu Verwirrungen führt (siehe Vanhove, 2013). Zwei Beispiele, um den Unterschied klar zu machen:

- Wenn man die Temperatur in Grad Celsius kennt, kann man die Temperatur in Grad Fahrenheit perfekt 'schätzen': Die *Korrelation* ist also äusserst stark (Frage 1). Damit wissen wir aber noch nicht, *wie* wir die Temperatur in Grad Fahrenheit berechnen können, wenn wir die Temperatur in Grad Celsius kennen. Eine *Regression*sanalyse würde zeigen, dass wir dazu die folgende Formel anwenden müssten:

$$\text{Grad Fahrenheit} = 32 + \frac{9}{5} \times \text{Grad Celsius} \tag{8.1}$$

- Wenn man die Körpergrösse eines Menschen kennt, kann man sein Gewicht wesentlich besser schätzen, als wenn man die Körpergrösse nicht kennt. Die Schätzung ist aber nicht perfekt: Die Korrelation ist positiv, aber nicht so hoch wie im letzten Beispiel (Frage 1). Um zu wissen, wie man das Gewicht am besten anhand der Grösse schätzt (z.B. Gewicht in kg $=$ 0.6 $\times$ Grösse in cm $-$ 40kg für Frauen zwischen 145 und 185 cm), braucht es Regressionsanalyse.

Die zweite Frage ist m.E. in der Regel (nicht immer!) viel sinnvoller. In unserem Beispiel wäre das Ziel also, eine Gleichung wie Gleichung 8.1 zu finden, anhand derer man Unterschiede im GJT-Ergebnis mit Unterschieden im AOA verknüpfen kann. Um diese Gleichung zu finden, brauchen wir zuerst aber eine Antwort auf die erste Frage.

Nicht-lineare Zusammenhänge. Sowohl Korrelation- als auch Regressionsanalyse können sinnvoll sein, um lineare Zusammenhänge zu untersuchen. Ist der Zusammenhang zwischen den Variablen nicht *ungefähr* gerade, dann *kann* man die Berechnung zwar noch immer ausführen, dies ist aber kaum sinnvoll. Manchmal kann man Daten sinnvoll transformieren, sodass der Zusammenhang linear wird (Beispiele in etwa Baayen, 2008 und Gelman & Hill, 2007). Ist dies nicht möglich, dann sind komplexere Verfahren geeignet (siehe etwa Clark, 2018).

Fragen und Werkzeuge. Korrelations-, Regressions- und sonstige Analysen, Modelle und Tests sind lediglich Werkzeuge. Je nach Fragestellung sind diese Werkzeuge nützlich oder nutzlos. Anstatt sich etwa vorzunehmen, eine Korrelationsanalyse oder einen t-Test durchzuführen (vielleicht, weil dies in einer bestimmten Forschungsliteratur gang und gäbe ist), ist es sinnvoller, die Frage ohne ablenkenden technischen Wortschatz (z.B. *Korrelation*, *signifikant*) zu formulieren und sich dann zu überlegen, welches Werkzeug für deren Beantwortung am nützlichsten ist. Das Ziel einer Datenerhebung und einer Analyse ist es, eine Frage zu beantworten, nicht ein bestimmtes Werkzeug zu benutzen.

8.2 Antwort auf Frage 1: Kovarianz und Korrelation

8.2.1 Kovarianz

Um numerisch zu beschreiben, wie stark zwei Variable miteinander zusammenhängen, brauchen wir ein Mass, dessen absoluter Wert gross ist, wenn kleine Unterschiede in x mit kleinen Unterschieden in y zusammenhängen und grosse Unterschiede in x mit grossen Unterschieden in y, und dessen absoluter Wert klein ist, wenn grosse Unterschiede in der einen Variablen mit nur kleinen Unterschieden in der anderen Variablen zusammenhängen. Ein solches Mass ist die Kovarianz:

$$\text{Cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (\bar{x} - x_i)(\bar{y} - y_i) \tag{8.2}$$

Die Summe der Produkte ($\sum (\bar{x} - x_i)(\bar{y} - y_i)$) wird durch $n-1$ statt durch n geteilt aus dem gleichen Grund, weshalb dies bei der Varianzberechnung gemacht wird. Eine intuitivere Erklärung ist, dass man ja nicht über den Zusammenhang von zwei Variablen sprechen kann, wenn man nur eine Beobachtung pro Variable hat. (Das Streudiagramm würde dann nur einen Punkt zeigen.) Wenn $n = 1$ ist $n - 1 = 0$ und dann ergibt die Gleichung keine Antwort, denn durch 0 kann nicht geteilt werden.

In R:

```
# kompliziert:
sum((mean(d$AOA) - d$AOA) * (mean(d$GJT) - d$GJT)) / (nrow(d) - 1)
```

```
## [1] -394.9311
```

```
# einfacher:
cov(d$AOA, d$GJT)
```

```
## [1] -394.9311
```

Ist die Kovarianz positiv, dann besteht ein positiver Zusammenhang zwischen den beiden Variablen (je grösser x, desto grösser y); ist die Kovarianz negativ, dann gibt es einen negativen Zusammenhang (je grösser x, desto kleiner y). Abgesehen von diesen zwei Richtschnuren ist das Kovarianzmass schwierig zu interpretieren, weshalb Sie es in der Literatur nur selten antreffen werden. Aber Kovarianz ist ein wichtiges Konzept in der Mathe hinter komplexeren Verfahren, weshalb es sich trotzdem lohnt, zumindest zu wissen, dass es besteht.

Noch gut zu wissen: $\text{Cov}(x, y) = \text{Cov}(y, x)$:

```
cov(d$GJT, d$AOA)
```

```
## [1] -394.9311
```

Die Kovarianz einer Variablen mit sich selbst ist gleich ihrer Varianz:

```
cov(d$GJT, d$GJT)
```

```
## [1] 746.256
```

```
var(d$GJT)
```

```
## [1] 746.256
```

8.2.2 Korrelation

Da das Kovarianzmass nicht einfach zu interpretieren ist, wird meistens Pearsons **Produkt-Moment-Korrelationskoeffizient** (r) (oder einfach Pearsons Korrelation) verwendet. Diese Zahl drückt aus, wie gut der Zusammenhang durch eine gerade Linie beschrieben werden kann. Es wird ähnlich zum Kovarianzmass berechnet, aber die Variablen werden in Standardabweichungen zum Stichprobemittel ausgedrückt. Dies ergibt dann immer eine Zahl zwischen -1 und 1.

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{\bar{x} - x_i}{s_x} \frac{\bar{y} - y_i}{s_y} = \frac{\mathrm{Cov}(x, y)}{s_x s_y} \tag{8.3}$$

```
# kompliziert:
cov(d$AOA, d$GJT) / (sd(d$AOA) * sd(d$GJT))
```

```
## [1] -0.8028533
```

```
# einfach:
cor(d$AOA, d$GJT)
```

```
## [1] -0.8028533
```

Wenn ein paar Datenpunkte fehlen, ergibt diese Funktion das Ergebnis 'NA' (*not available*). Eine Möglichkeit ist dann, die fehlenden Daten zu ignorieren:

```
cor(d$AOA, d$GJT, use = "pairwise.complete.obs")
```

```
## [1] -0.8028533
```

Ist $r = 1$, dann liegen alle Datenpunkte perfekt auf einer geraden, steigenden Linie. Dies deutet fast ausnahmslos auf eine Tautologie hin. Zum Beispiel sind Körpergrössen in Zentimetern und in Zoll perfekt korreliert, aber dieser Zusammenhang ist nicht spektakulär sondern höchst langweilig. Ist $r = -1$, dann liegen alle Datenpunkte auf einer geraden, senkenden Linie. Dies deutet wohl darauf hin, dass die beiden Variablen perfekt komplementär sind. Zum Beispiel wird die Anzahl richtiger Antworten oft mit $r = -1$ mit der Anzahl falscher Antworten korrelieren; auch dies ist wenig spektakulär. Ist $r = 0$, dann ist die Linie perfekt senkrecht, d.h. es gibt überhaupt keinen linearen Zusammenhang zwischen den beiden Variablen.

Je grösser der absolute Wert von r, desto näher befinden sich die Datenpunkte bei der geraden Linie. Anders ausgedrückt: Je grösser der absolute r-Wert, desto präziser kann

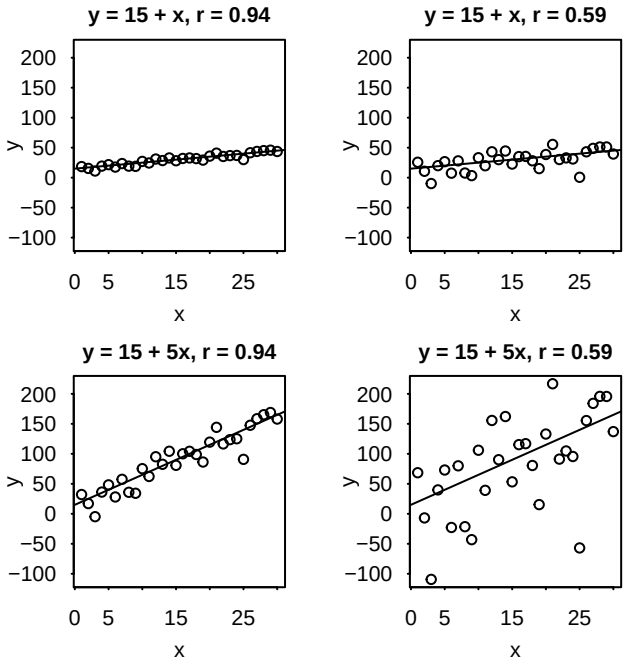


Abbildung 8.3: Korrelationskoeffizienten erzählen einem wenig über die Form eines Zusammenhangs.

man y bestimmen, wenn man x schon kennt (und umgekehrt) als wenn man x nicht gekannt hätte.

Die Korrelation zwischen x und y ist gleich der Korrelation zwischen y und x. Es macht also nichts aus, ob man `cor(dat$AOA, dat$GJT)` oder `cor(dat$GJT, dat$AOA)` eintippt.

Abbildung 8.3 zeigt vier Zusammenhänge, um die Bedeutung von Pearsons r zu illustrieren.

- *Oben links:* Es gibt wenig Streuung entlang der y-Achse. Die Streuung, die es gibt, wird grösstenteils von einer Geraden erfasst. r ist daher sehr hoch.
- *Oben rechts:* Es gibt nun mehr Streuung entlang der y-Achse; diese wird aber weniger gut von einer Geraden erfasst, daher der niedrigere Korrelationskoeffizient. Die Form der Geraden ist zwar gleich wie in der linken Grafik, der Korrelationskoeffizient jedoch nicht.
- *Unten links:* Es gibt zwar sehr viel Streuung entlang der y-Achse, aber diese wird grösstenteils von einer Geraden erfasst. r ist daher wiederum sehr hoch. Der Korrelationskoeffizient ist zwar gleich wie in der obigen Grafik, die Form der Geraden jedoch nicht.
- *Unten rechts:* Die gleiche Gerade erfasst die Streuung entlang der y-Achse weniger gut, daher ist die Form der Geraden zwar gleich, der Korrelationskoeffizient aber niedriger.

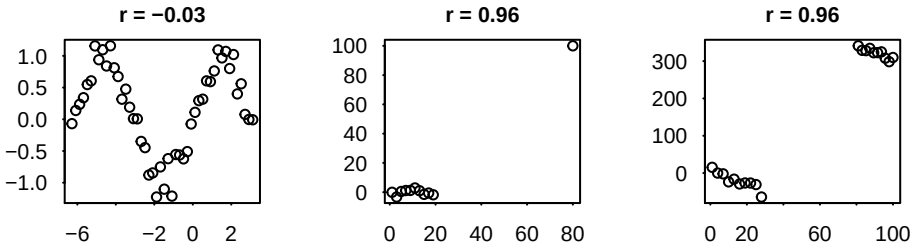


Abbildung 8.4: Ein Korrelationskoeffizient nahe 0 muss nicht heissen, dass es keinen Zusammenhang zwischen den Variablen gibt, und ein Korrelationskoeffizient nahe 1 muss nicht heissen, dass das Muster in den Daten am besten durch einen starken positiven Zusammenhang beschrieben wird.

Welche Frage beantwortet r (und welche nicht)?

Pearsons r drückt aus, wie nahe die Datenpunkte *auf einer geraden Linie* fallen; es gibt keine direkte Antwort auf die Frage, wie denn diese Linie ausschaut (ausser: steigend oder senkend); siehe die vier obigen Beispiele.

Ausserdem ist es möglich, dass es einen sehr starken (nicht-linearen) Zusammenhang zwischen zwei Variablen gibt, dieser aber in Pearsons r nicht zum Ausdruck kommt (Abbildung 8.4, links). Umgekehrt kann r den Eindruck geben, dass es sich um einen ziemlich starken linearen Zusammenhang handelt, während ein solcher Zusammenhang für die meisten Datenpunkte kaum vorliegt (mittel), oder während der Zusammenhang sogar eigentlich in die umgekehrte Richtung geht (rechts: Es gibt zwei Gruppen, in denen der Zusammenhang negativ ist; der Koeffizient ist jedoch positiv, wenn die beiden Gruppen gleichzeitig betrachtet werden).

Mit der `plot_r()`-Funktion im `cannonball`-Paket (`https://github.com/janhove/cannonball`) können Sie selber Streudiagramme zeichnen, die alle anders aussehen, aber den gleichen Korrelationskoeffizienten haben. Der Blogeintrag *What data patterns can lie behind a correlation coefficient?* (21.11.2016) beschreibt diese Funktion. Abbildung 8.5 zeigt 16 Zusammenhänge mit je 50 Beobachtungen, die alle eine Korrelation von $r = -0.72$ aufzeigen:

```
# cannonball installieren
devtools::install_github("janhove/cannonball")
```

```
library(cannonball)
plot_r(n = 50, r = -0.72)
```

Spielen Sie mit der `plot_r()`-Funktion herum, um besser zu verstehen, was ein Korrelationskoeffizient eben alles nicht bedeutet und was der Einfluss von nicht-linearen Zusammenhängen und Ausreissern sein kann:

```
plot_r(n = 15, r = 0.9)
plot_r(n = 80, r = 0.0)
```

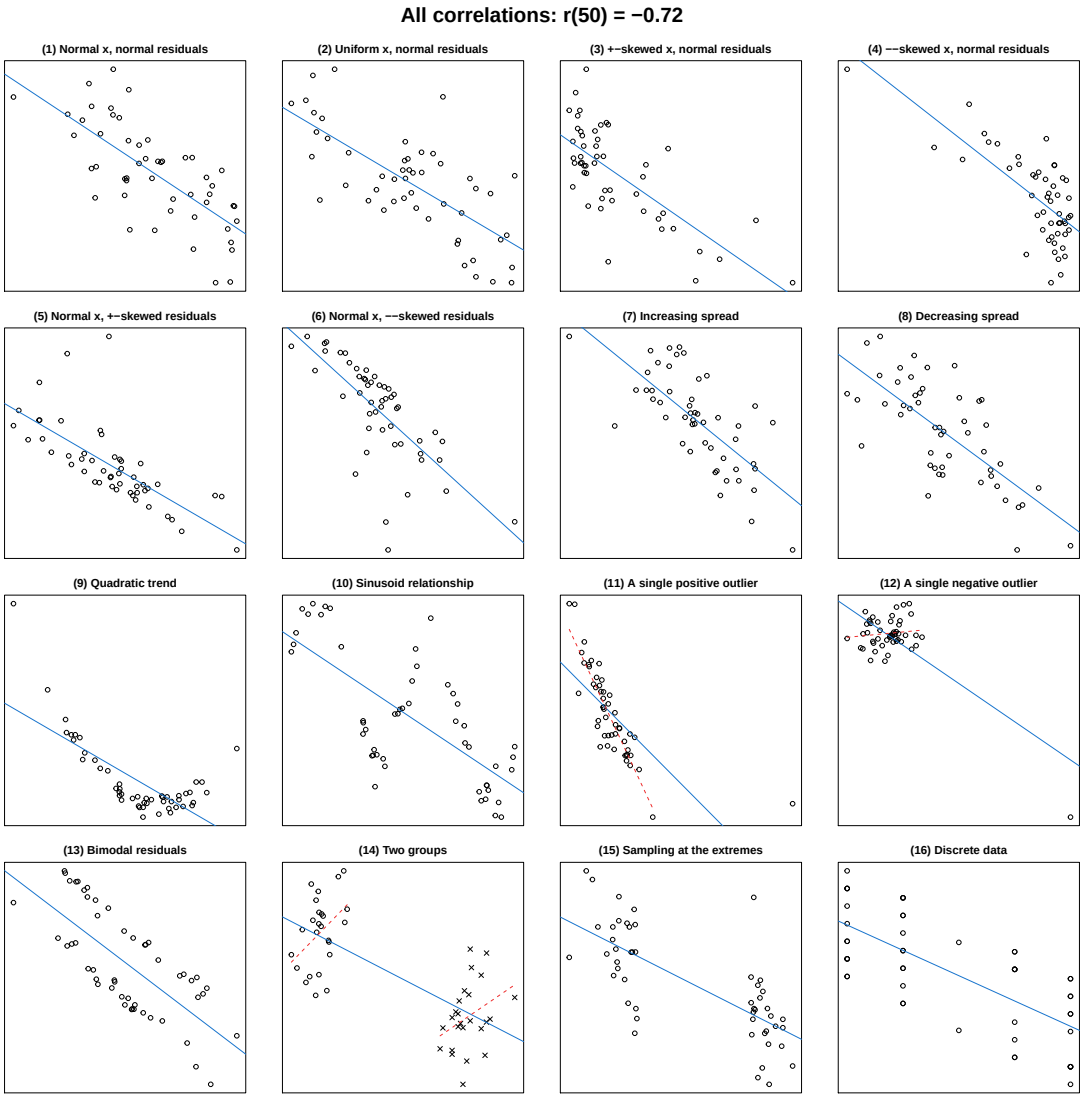


Abbildung 8.5: Alle sechzehn Zusammenhänge zeigen eine Korrelation von -0.72 auf, sehen jedoch zum Teil ganz unterschiedlich aus.

```
plot_r(n = 30, r = 0.4)

# Die Funktionsdokumentation können Sie wie folgt abrufen:
?plot_r
```

Ein Korrelationskoeffizient kann einer Vielzahl von Zusammenhängen entsprechen. Schauen Sie sich, bevor Sie Korrelationskoeffizienten berechnen, immer die Daten grafisch (Streudiagramm) an. Nehmen Sie diese Streudiagramme in Ihre Papers, Arbeiten und Vorträge auf.

Andere Korrelationsmasse

Ab und zu trifft man Spearmans ρ-Koeffizienten (oder manchmal: r_s) an. Hierfür drückt man die Daten in **Rängen** aus, d.h. man ordnet die Daten von klein nach gross und schaut, auf welchem Platz die einzelnen Datenpunkte stehen. Dann berechnet man einfach die Pearsonkorrelation für die Ränge statt für die Rohwerte:

```
cor(rank(d$AOA), rank(d$GJT))

## [1] -0.7887659

# einfacher:
cor(d$AOA, d$GJT, method = "spearman")

## [1] -0.7887659
```

Spearmans ρ kann nützlich sein, wenn der Zusammenhang zwischen zwei Variablen monoton aber nicht-linear ist (Monoton heisst: Tendenziell steigend oder tendenziell senkend; nicht etwa zuerst steigend und dann senkend.) oder wenn ein **Ausreisser** das Globalbild zerstört, aber man ihn aus irgendwelchem Grund nicht aus dem Datensatz entfernen kann. Wenn Spearmans $\rho = 1$, dann ist der Zusammenhang perfekt monoton steigend (höhere Werte in x entsprechen immer höheren Werten in y), wenn Spearmans $\rho = -1$, dann ist der Zusammenhang perfekt monoton senkend, und wenn $\rho = 0$, dann gibt es keinen monotonen Zusammenhang in den Daten. Bemerken Sie, dass man mit ρ eine andere Frage beantwortet als mit r: *Wie stark ist der monotone Zusammenhang?* vs. *Wie stark ist der lineare Zusammenhang?*

Ein anderes Mass ist Kendalls τ (`method = "kendall"`). Dieses wird aber nur höchst selten verwendet. Die Berechnung ist konzeptuell relativ einfach (siehe Noether, 1981), aber schaut in R-Code schwierig aus, weshalb ich sie hier nur in Worten zusammenfasse:

1. Vergleiche jeden x-Wert mit jedem anderen x-Wert und notiere, ob Ersterer grösser oder kleiner als Letzterer ist. Wenn die x-Werte zum Beispiel 5, 3, 8 und 7 sind, erhält man folgende Vergleiche:
    - 5 vs. 3: grösser

- 5 vs. 8: kleiner
- 5 vs. 7: kleiner
- 3 vs. 8: kleiner
- 3 vs. 7: kleiner
- 8 vs. 7: grösser

2. Vergleiche jeden y-Wert mit jedem anderen y-Wert. Für die y-Werte 8, -2, -4, -3 erhält man: grösser, grösser, grösser, grösser, grösser, kleiner
3. Zähle, wie viele Vergleiche in die gleiche Richtung gehen:
   - 1. Vergleich: grösser–grösser: gleich
   - 2. Vergleich: kleiner–grösser: anders
   - 3. Vergleich: kleiner–grösser: anders
   - 4. Vergleich: kleiner–grösser: anders
   - 5. Vergleich: kleiner–grösser: anders
   - 6. Vergleich: grösser–kleiner: anders

   Also 1 Vergleich, der in die gleiche Richtung geht ('konkordant'), und 5, die in die andere Richtung gehen ('diskordant').
4. Schätze jetzt Kendalls τ wie folgt:

$$\hat{\tau} = \frac{\text{Anzahl konkordant} - \text{Anzahl diskordant}}{\text{Anzahl Vergleiche}} \tag{8.4}$$

   Das Hütchen zeigt, dass wir es mit einer Schätzung auf der Basis einer Stichprobe zu tun haben. Also:

$$\hat{\tau} = \frac{1-5}{6} = -0.67 \tag{8.5}$$

```
# Für unser kleines Beispiel
x <- c(5, 3, 8, 7)
y <- c(8, -2, -4, -3)
cor(x, y, method = "kendall")
```

```
## [1] -0.6666667
```

```
# Für die AOA-GJT-Daten
cor(d$AOA, d$GJT, method = "kendall")
```

```
## [1] -0.6035606
```

Kendalls $\hat{\tau}$ schätzt den Unterschied zwischen der Proportion konkordanter Vergleiche und der Proportion diskordanter Vergleiche. Diese Interpretation finde ich selber

schwierig, aber es gibt eine einfachere Interpretation: Nehme zwei beliebige (x, y)-Paare (also (x_1, y_1) und (x_2, y_2)). Wenn x_2 grösser ist als x_1, dann ist es $\frac{1+\hat{\tau}}{1-\hat{\tau}}$ Mal wahrscheinlicher, dass auch y_2 grösser als y_1 ist als das er kleiner ist. Für die AOA–GJT-Daten: Wenn eine Person einen höheren AOA-Wert als eine andere hat, dann ist es $\frac{1+(-0.60)}{1-(-0.60)} = 0.25$ Mal wahrscheinlicher, dass sie auch einen höheren GJT-Wert als einen kleineren hat. Oder anders gesagt: Es ist 4 Mal wahrscheinlicher, dass sie einen kleineren GJT-Wert als einen grösseren hat.

In der Praxis ist die Anwendung von Spearmans ρ und Kendalls τ ist eher beschränkt. Statt *automatisch* auf ρ oder τ zurückzugreifen, wenn ein Zusammenhang nicht-linear ist oder wenn man einen Ausreisser vermutet, lohnt es sich m.E. eher, darüber nachzudenken, ob (a) man sich tatsächlich für Frage 1 (Stärke des Zusammenhangs) interessiert, (b) man eine oder beide Variablen nicht sinnvoll transformieren kann, sodass sich ein linearerer Zusammenhang ergibt, oder (c) der vermutete Ausreisser überhaupt ein legitimer Datenpunkt ist.

Starke und schwache Korrelationen

Korrelationskoeffizienten werden oft—ohne Berücksichtigung der Forschungsfrage oder des Kontextes—als klein, mittelgross oder gross eingestuft. Ich halte dies für weinig sinnvoll, weshalb ich diese Einstufungen hier nicht reproduziere. Wer mehr über sie erfahren möchte, kann sich bei Cohen (1992) und Plonsky & Oswald (2014) selber schlau machen. Dazu empfiehlt sich auch die Lektüre von Baguley (2009).

Selber finde ich, dass Korrelationskoeffizienten überverwendet werden. Blogeinträge zu diesem Thema:

- *Why I don't like standardised effect sizes* (5.2.2015)
- *More on why I don't like standardised effect sizes* (16.3.2015)
- *Abandoning standardised effect sizes and opening up other roads to power* (14.7.2017)

8.2.3 Die Ungenauigkeit eines Korrelationskoeffizienten einschätzen

Da sie auf der Basis von Stichproben berechnet werden, sind auch Korrelationskoeffizienten vom Stichprobenfehler betroffen: Andere Stichproben aus der gleichen Population werden Korrelationskoeffizienten ergeben, die mehr oder weniger voneinander abweichen. Die Ungenauigkeit bzw. die Variabilität eines auf einer Stichprobe basierenden Korrelationskoeffizienten kann mithilfe eines Konfidenzintervalls ausgedrückt werden. Besprochen werden hier eine Bootstrap-Methode und eine Methode, die auf t-Verteilungen basiert.

Mit dem Bootstrap

Das Vorgehen ist analog zum Bootstrap aus Kapitel 6: Aus der Stichprobe werden neue Bootstrap-Stichproben generiert und für jede Stichprobe wird die Statistik von Interesse (hier: die Korrelation zwischen AOA und GJT) berechnet. Die Streuung der Schätzungen in den Bootstrap-Stichproben gibt uns ein Indiz über die Variabilität des Korrelationskoeffizienten in Stichproben dieser Grösse.

```
# Anzahl Bootstraps definieren und Vektor für Ergebnisse freimachen.
# Für eine grobere Schätzung würden 200 oder so bootstrap samples auch
# reichen.
n_bootstraps <- 20000
bootstraps <- vector(length = n_bootstraps)

for (i in 1:n_bootstraps) {
  # Sampling with replacement aus der beobachteten Stichprobe.
  bootstrap_sample <- d %>%
    sample_frac(replace = TRUE)

  # Die Korrelation im bootstrap sample berechnen und speichern.
  bootstraps[i] <- cor(bootstrap_sample$GJT, bootstrap_sample$AOA)
}
```

```
# Histogramm mit den Bootstrap-Schätzungen
# (nicht gezeigt)
hist(bootstraps, breaks = 20)
```

Da das Histogramm nicht normalverteilt ist, verzichten wir hier auf die Berechnung eines Standardfehlers. (Die Verteilung von Stichproben-Korrelationskoeffizienten kann nicht normalverteilt sein, da Korrelationskoeffizienten zwischen -1 und 1 begrenzt sind.) Anhand der Perzentile der Verteilung können wir aber durchaus ein Konfidenzintervall konstruieren. Hier berechne ich ein 90%-Konfidenzintervall:

```
quantile(bootstraps, probs = c(0.05, 0.95))
```

```
##         5%        95%
## -0.8629308 -0.7304224
```

Mit t-Verteilungen

Die Formel, mit der man anhand von einer t-Verteilung ein Konfidenzintervall um einen Korrelationskoeffizienten konstruiert, werde ich hier nicht reproduzieren, da sie erstens abschreckt und zweitens keinen konzeptuellen Mehrwert bietet. Sie basiert auf der Annahme, dass die Population, aus der die beiden Variablen gezogen wurden, 'bivariat normal' ist. Grundsätzlich heisst dies, dass—insofern es einen Zusammenhang zwischen den Variablen gibt—dieser Zusammenhang linear ist und beide Variable nor-

malverteilt sind. Wenn diese Annahmen plausibel sind, kann das Konfidenzinterval (hier wiederum ein 90%-Konfidenzintervall) mit der `cor.test()`-Funktion berechnet werden:

```
cor.test(d$AOA, d$GJT, conf.level = 0.9)
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$AOA and d$GJT
## t = -11.584, df = 74, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 90 percent confidence interval:
##  -0.8614924 -0.7230816
## sample estimates:
##       cor
## -0.8028533
```

Mit dieser Methode erhalten wir ein 90%-Konfidenzintervall von $[-0.86, -0.72]$. Dieses unterscheidet sich nur minimal vom Konfidenzintervall, das wir mit dem Bootstrap berechnet haben. Beim Bootstrap sind wir jedoch nicht davon ausgegangen, dass die Population, aus der die Stichprobe stammt, bivariat normalverteilt war. Insbesondere bei kleineren Stichproben können die Ergebnisse der Bootstrap- und der t-Methode erheblich voneinander abweichen. Wenn ihre Annahmen stimmen, ist die t-Methode in solchen Fällen zweifellos besser, aber gerade bei kleinen Stichproben sind diese Annahmen schwierig zu überprüfen. Die Leistung der Bootstrapmethode kann noch etwas verbessert werden, indem die Konfidenzintervalle anders konstruiert werden (siehe hierzu DiCiccio & Efron, 1996), aber diese Konstruktionsmethoden sind schwieriger und weniger intuitiv. Für unsere Zwecke reicht es m.E., die Warnung von Hesterberg (2015) zu wiederholen: "Bootstrapping does not overcome the weakness of small samples as a basis for inference." (S. 379)

(Was 't $= -11.6$' und 'p-value $<$ 2.2e16' heissen, werden wir in einem späteren Kapitel besprechen.)

Gut zu wissen! Das Konfidenzintervall um einen Korrelationskoeffizienten hängt von nur drei Faktoren ab:

- ob das Konfidenzintervall ein 50%-, 80%-, 87%- usw.-Konfidenzintervall sein soll;
- dem Korrelationskoeffizienten selber;
- der Anzahl beobachteter Paare.

Wenn man weiss, was der Korrelationskoeffizient ist, und wie gross die Stichprobe war, kann man also selber das Konfidenzintervall berechnen. Selber finde ich dies nützlich, wenn das Konfidenzintervall um r in einer Studie nicht berichtet wurde.

Mit der Funktion `r.test()` aus dem `psych`-Package ist dies ein Kinderspiel, allerdings werden nur 95%-Konfidenzintervalle berechnet. (Gegebenenfalls müssen Sie das `psych`-Paket noch installieren.)

```
psych::r.test(r12 = -0.80, n = 76)
```

```
## Correlation tests
## Call:psych::r.test(n = 76, r12 = -0.8)
## Test of significance of a  correlation
##  t value -11.47    with probability < 0
##  and confidence interval  -0.87 -0.7
```

Das 95%-Konfidenzintervall eines Korrelationskoeffizienten von $r = -0.80$ in einer Stichprobe mit 76 Datenpunkten ist also $[-0.87, -0.70]$. Dabei gehen wir zwar davon aus, dass die Stichprobe aus einer bivariaten Normalverteilung stammt, aber bei 76 Beobachtungen würden andere Methoden wohl ein sehr ähnliches Ergebnis liefern.

Falls Sie lieber 80%- oder 90%-Konfidenzintervalle um Korrelationskoeffizienten berechnen, können Sie die unten stehende Funktion übernehmen. Auch die von ihr berechneten Konfidenzintervalle basieren auf der Annahme, dass die Daten aus einer bivariaten Normalverteilung stammen.

```
ci_r <- function(r, n, conf_level = 0.90) {
  # Datensatz mit erwünschten Merkmalen kreieren.
  # Dazu braucht's das cannonball-Paket.
  dat <- cannonball::plot_r(r = r, n = n, showdata = 1, plot = FALSE)
  # Konfidenzintervall berechnen
  ci <- cor.test(dat$x, dat$y, conf.level = conf_level)$conf.int[1:2]
  # Ausspucken
  return(ci)
}
# 95%-Konfidenzintervall
ci_r(r = -0.80, n = 76, conf_level = 0.95)
```

```
## [1] -0.8687618 -0.7009755
```

```
# 80%-Konfidenzintervall
ci_r(r = -0.80, n = 76, conf_level = 0.80)
```

```
## [1] -0.8478924 -0.7391568
```

```
# 50%-Konfidenzintervall
ci_r(r = -0.80, n = 76, conf_level = 0.50)
```

```
## [1] -0.8266792 -0.7697318
```

8.2.4 Übungen

1. Es gibt mittlerweile eine ausführliche Literatur zur Frage, inwieweit Zweisprachigkeit zu kognitiven Vorteilen führt. Ein kognitives **Konstrukt**, das oft in diesem Zusammenhang erwähnt wird, ist die kognitive Kontrolle. Dieses Konstrukt lässt sich nur *indirekt* messen, nämlich mithilfe von kognitiven Tests: Die Leistung beim Test *ist* nicht die kognitive Kontrolle einer Person, sondern lediglich ein imperfekter **Indikator** hierfür. Wenn unterschiedliche Indikatoren von kognitiver Kontrolle stark miteinander korrelieren, ist es aber wahrscheinlicher, dass Befunde, die auf dem einen Indikator basieren, auch zu anderen Indikatoren generalisieren.

   Die Datei `poarch2018.csv` enthält Angaben zu zwei kognitiven Tests, von denen angenommen wird, dass sie Indikatoren von kognitiver Kontrolle sind: dem Flanker-Test (Eriksen & Eriksen, 1974) und dem Simon-Test (Simon, 1969). In beiden Tests müssen die Versuchspersonen manchmal irrelevante Informationen ignorieren. Die Daten stammen aus einer kleinen Studie von Poarch et al. (2019), in der den Probanden beide Tests vorgelegt wurden. Die Ergebnisse stellen dar, wie viel schneller die Versuchspersonen reagierten, wenn die irrelevante Information 'kongruent' mit der relevanten Information ist als, wenn die irrelevante Information 'inkongruent' mir der relevanten Information ist. Ausgedrückt werden die Angaben in Stimuli pro Sekunde; ein Wert von 0.5 heisst also, dass die Versuchsperson in einer kongruenten Testsituation 5 Stimuli mehr bewältigen kann pro 10 Sekunden, als bei einer inkongruenten Testsituation.

   (a) Lesen Sie diesen Datensatz ein.

   (b) Stellen Sie den Zusammenhang zwischen den Variablen `Flanker` und `Simon` grafisch dar.

   (c) Berechnen Sie den Korrelationskoeffizienten, insofern Sie dies für sinnvoll halten.

   (d) Berechnen Sie gegebenfalls das 90%-Konfidenzintervall, und zwar sowohl anhand des Bootstraps als auch mit der t-Verteilung.

   (e) Fassen Sie Ihre Befunde schriftlich zusammen (höchstens drei Sätze).

2. Fakultativ: Kendalls $\hat{\tau}$ für den AOA–GJT-Zusammenhang beträgt -0.60. Für diese Schätzung berechnet die `cor.test()`-Funktion berechnet jedoch kein Konfidenzintervall.

   (a) Passen Sie den Bootstrap auf Seite 103 so an, dass dieser ein 90%-Konfidenzintervall für $\hat{\tau}$ berechnet.

   (b) $\hat{\tau} = -0.60$ heisst, dass es etwa 4 Mal wahrscheinlicher ist, dass eine belieblige Person mit einem höheren AOA-Wert als eine belieblige andere Person einen niedrigeren GJT-Wert als diese Person hat. Was ist das 90%-Konfidenzintervall um diese Zahl (also 4)?

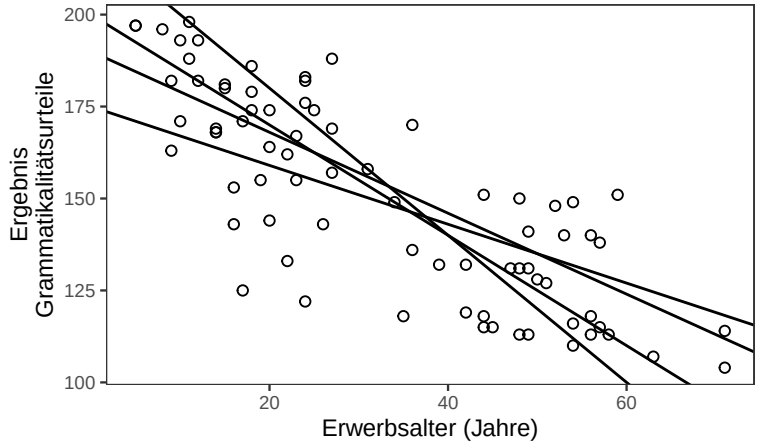


Abbildung 8.6: Wenn man von Hand eine gerade Linie durch die Punktwolke ziehen würde, zeichnet jeder die Linie wohl an einer anderen Stelle. Wir können aber Kriterien festlegen, sodass alle die gleiche Linie zeichnen.

8.3 Antwort auf Frage 2: Regression

Es ist klar, dass es im Datensatz `dekeyser2010.csv` einen Zusammenhang zwischen AOA und GJT gibt. Eine senkende gerade Linie erfasst die Tendenz in den GJT-Daten schon ziemlich gut. Aber wie schaut diese Linie genau aus? Wir könnten zwar von Hand eine Gerade durch die Punktwolke ziehen, aber jeder zieht die Linie wohl an einem etwas anderen Ort, siehe Abbildung 8.6. Eine prinzipiellere Herangehensweise wäre daher erwünscht.

8.3.1 Die einfache Regressionsgleichung

Ähnlich wie im letzten Kapitel können wir eine Gleichung aufschreiben, die die Datenpunkte in zwei Teile zerlegt: einen Teil, der die Gemeinsamkeiten zwischen allen Werten ausdrückt, und einen Teil, der die individuellen Unterschiede zwischen diesen Gemeinsamkeiten und den Werten ausdrückt. Diesmal können wir den Zusammenhang zwischen AOA und GJT als eine Gemeinsamkeit im Datensatz betrachten. Dieser Zusammenhang scheint linear, weshalb er als eine gerade Linie modelliert werden kann.

$$\text{Wert einer Vpn.} = \text{Gemeinsamkeit (inkl. AOA-Zusammenhang)} + \text{Abweichung} \quad (8.6)$$

Eine gerade Linie wird definiert durch einen Schnittpunkt (β_0; dies ist der y-Wert, wenn $x = 0$) und eine Steigung (β_1; diese sagt, um wie viele Punkte y steigt, wenn x um eine Einheit erhöht wird); siehe Abbildung 8.7.

Egal, wie wir β_0 und β_1 wählen: Die Linie $y_i = \beta_0 + \beta_1 x_i$ wird die Daten nicht perfekt beschreiben: Es wird noch einen Restfehler (ε) geben. Jeder y-Wert (y_1, y_2 etc.) kann

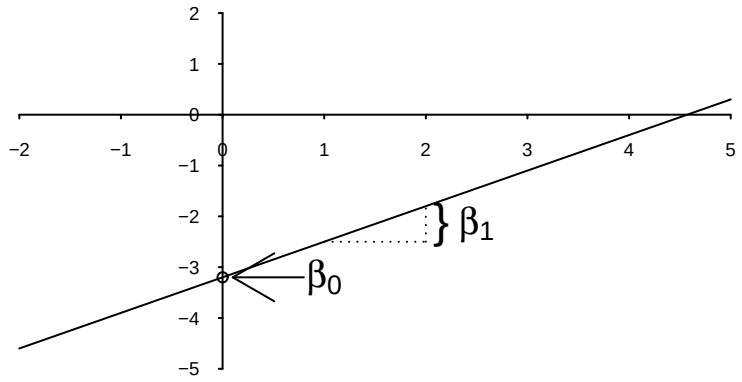


Abbildung 8.7: Schnittpunkt und Steigung einer Geraden.

also umschrieben werden als die Kombination eines systematischen Teils ($\beta_0 + \beta_1 x_i$) und eines Restfehlers:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{8.7}$$

Diese mathematische Beschreibung ist ein einfaches lineares Modell: 'einfach', weil y nur eine Funktion einer Variablen (x) ist, und 'linear', weil y als eine Summe (und nicht etwa ein Produkt oder etwas Komplexeres) verschiedener Terme modelliert wird.

8.3.2 Die Parameter schätzen

β_0 und β_1 sind die **Parameter** der einfachen Regressionsgleichung, und unsere nächste Aufgabe ist es, diese Parameter so gut wie möglich zu **schätzen**. Dass wir diese Parameter nur schätzen und nicht *wissen* können, zeigt sich in einem Gedankenexperiment: Wenn wir die gleiche Studie nochmals unter den gleichen Bedingungen durchführen würden, aber mit anderen Teilnehmenden, würde das Streudiagramm ja nicht identisch aussehen. Wir würden aber davon ausgehen, dass beide Studien uns Informationen über den 'wahren' Zusammenhang zwischen AOA und GJT in dieser Population liefern. Dieser 'wahre' Zusammenhang—so unsere Annahme—wird durch die obige Regressionsgleichung beschrieben, aber auf der Basis empirischer Forschung kann der Zusammenhang höchstens approximiert werden.

Im Prinzip gibt es unendlich viele Möglichkeiten, β_0 und β_1 auszuwählen, sodass die Gleichung aufgeht (wie Abbildung 8.6 illustriert), aber uns interessieren nur die β_0- und β_1-Werte der *optimalen* Geraden. Um diese zu schätzen, müssen wir definieren, was 'optimal' in diesem Kontext heisst. Wie im letzten Kapitel besprochen wurde, ist das am meisten verwendete Optimierungskriterion die Methode der kleinsten Quadrate, wonach die optimale Linie jene Gerade ist, die die Summe der quadrierten Restfehler minimiert (min $\sum_{i=1}^{n} \widehat{\varepsilon_i}^2$). Wir können diese Summe berechnen für unterschiedliche Kombinationen von $\widehat{\beta_0}$- und $\widehat{\beta_1}$-Werten.

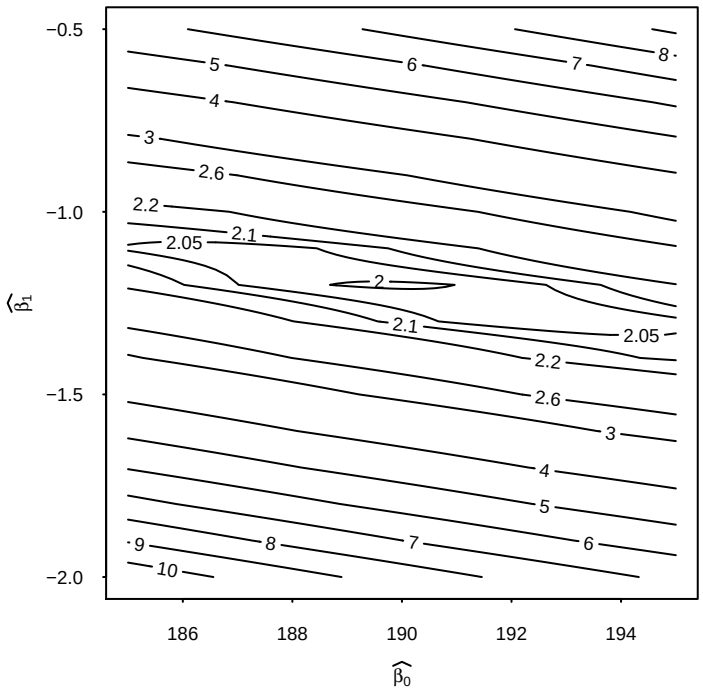


Abbildung 8.8: Die Summe der quadrierten Restfehler (geteilt durch 10'000) der GJT-Daten für unterschiedliche Parameterschätzungen. Diese Grafik kann wie eine topografische Karte gelesen werden; die Linien sind sozusagen Höhenlinien. Für Schnittpunkte nahe bei 190 und Steigungen nahe bei -1.2 wird diese Summe minimiert; bei diesen Koordinaten gibt es sozusagen einen Kessel.

- Sind $\widehat{\beta_0} = 185$ und $\widehat{\beta_1} = -2$, dann ist $\sum_{i=1}^{n} \widehat{\varepsilon_i}^2 = \sum_{i=1}^{n} (y_i - (185 - 2x_i))^2 = 107121$:

```
sum((d$GJT - (185 - 2*d$AOA))^2)

## [1] 107121
```

- Sind $\widehat{\beta_0} = 190$ und $\widehat{\beta_1} = -1.5$, dann ist $\sum_{i=1}^{n} \widehat{\varepsilon_i}^2 = \sum_{i=1}^{n} (y_i - (190 - 1.5x_i))^2 = 28810$:

```
sum((d$GJT - (190 - 1.5*d$AOA))^2)

## [1] 28810.25
```

$\widehat{\beta} = \{190, -1.5\}$ sind also optimaler als $\widehat{\beta} = \{180, -2\}$. Abbildung 8.8 zeigt die Summe der quadrierten Restfehler für unterschiedliche $\widehat{\beta_0}$– $\widehat{\beta_1}$-Kombinationen; Kombinationen nahe bei $\{190, -1.2\}$ haben die kleinste Summe der quadrierten Restfehler.

In der Praxis ackert man nicht zig Parameterkombinationen durch, um die optimale zu finden. Der Vorteil der Methode der kleinsten Quadrate ist, dass man die optimalen Parameterschätzungen schnell algebraisch finden kann, und zwar so:

$$\widehat{\beta_1} = r_{xy} \frac{s_y}{s_x} \tag{8.8}$$

$$\widehat{\beta_0} = \bar{y} - \widehat{\beta_1}\bar{x} \tag{8.9}$$

(Der Beweis dafür wird hier nicht reproduziert.)

Für die Daten von DeKeyser et al. (2010):

```
beta1_hat <- cor(d$AOA, d$GJT) * sd(d$GJT) / sd(d$AOA)
beta1_hat

## [1] -1.217977

beta0_hat <- mean(d$GJT) - beta1_hat * mean(d$AOA)
beta0_hat

## [1] 190.4086
```

Einfacher geht es mit der `lm()`-Funktion:

```
aoa.lm <- lm(GJT ~ AOA, data = d)
aoa.lm

##
## Call:
## lm(formula = GJT ~ AOA, data = d)
##
## Coefficients:
## (Intercept)          AOA
##     190.409       -1.218
```

`(Intercept)` ist hier $\widehat{\beta_0}$, also der Schnittpunkt der Regressionsgerade mit der y-Achse. `AOA` ist $\widehat{\beta_1}$ und zeigt, um wie viele Einheiten die Gerade steigt oder senkt, wenn `AOA` um eine Einheit erhöht wird.

8.4 Regressionsgeraden zeichnen

Das Ergebnis einer Regressionsanalyse kann auch grafisch dargestellt werden, indem man dem Streudiagramm die Regressionsgerade hinzufügt (Abbildung 8.9):

```
ggplot(data = d,
       aes(x = AOA,
           y = GJT)) +
  geom_point(shape = 1) +
  # Mit geom_abline() wird dem Streudiagramm eine Gerade hinzugefügt.
  geom_abline(intercept = 190.41, slope = -1.218)
```

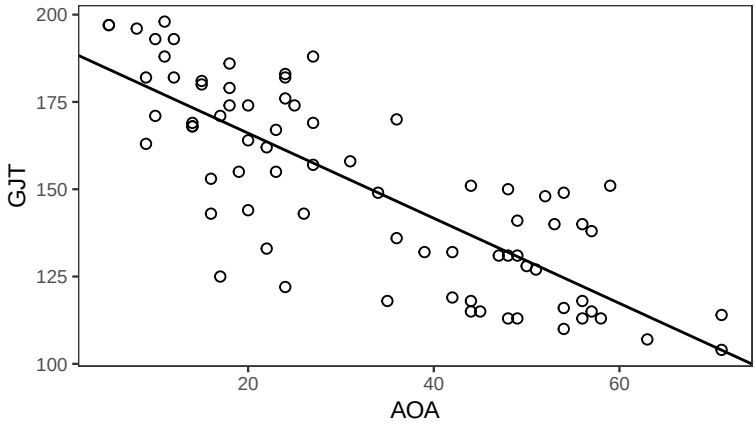


Abbildung 8.9: Ein Streudiagramm mit einer Regressionsgeraden. Die Regressionsgerade erfasst die modellierte zentrale Tendenz (genauer: das GJT-Mittel) für die unterschiedlichen AOA-Werte.

Eine alternative Methode ist die folgende. Wenn man bei `geom_smooth()` als Methode `"lm"` einstellt, wird das Regressionsmodell von `ggplot()` berechnet. Im Code unten habe ich den Parameter `se` auf `FALSE` gestellt; im nächsten Abschnitt wird klar warum.

```
# Nicht gezeichnet
ggplot(data = d,
       aes(x = AOA, y = GJT)) +
  geom_point(shape = 1) +
  geom_smooth(method = "lm", se = FALSE)
```

Aber was stellt diese Gerade genau dar? Mit dem Regressionsmodell versuchten wir die GJT-Werte (y_i) als eine Funktion der AOA-Werte (x_i) und eines Restfehlers (ε_i) zu modellieren:

$$y_i = \widehat{\beta_0} + \widehat{\beta_1} x_i + \widehat{\varepsilon_i} \tag{8.10}$$

Auf der Regressionsgeraden ($\widehat{\beta_0} + \widehat{\beta_1} x_i$) liegen die y_i-Werte *abzüglich* des Restfehlers, also

$$\widehat{y_i} = \widehat{\beta_0} + \widehat{\beta_1} x_i \tag{8.11}$$

Wie man diese $\widehat{y_i}$-Werte *konzeptuell* interpretieren kann, ist einfacher zu erklären, wenn wir uns zuerst anschauen, wie man die Unsicherheit in den Parameterschätzungen quantifizieren kann.

8.5 Unsicherheit in Parameterschätzungen schätzen

Die Parameterschätzungen $\widehat{\beta}$ werden aufgrund des Stichprobenfehlers von Stichprobe zu Stichprobe mehr oder weniger voneinander abweichen. Da $\widehat{\beta}$ aus Schätzungen besteht, ist auch die Regressionsgerade nur eine Schätzung. Um die Unsicherheit in den Parameterschätzungen und in der Regressionsgeraden zu quantifizieren, können wir uns wiederum auf den Bootstrap oder auf algebraische Methoden verlassen.

8.5.1 Mit dem Bootstrap

Das Bootstrappen eines linearen Modells mit einem Prädiktor verläuft analog zum Bootstrappen eines linearen Modells ohne Prädiktor. Zuerst wird hier die Methode aus Abschnitt 7.4.1 aufs lineare Modell mit einem Prädiktor angewandt:

1. Man berechnet $\widehat{\beta}$ (also $\widehat{\beta_0}$ und $\widehat{\beta_1}$) und erhält dazu auch noch $\widehat{\varepsilon}$.
2. Man zieht eine Bootstrap-Stichprobe aus $\widehat{\varepsilon}$ ($\widehat{\varepsilon}^*$).
3. Man kombiniert $\widehat{\beta_0}$, $\widehat{\beta_1}x_i$ und $\widehat{\varepsilon}^*$. Dies ergibt eine neue Reihe von y-Werten: $y_i^* = \widehat{\beta_0} + \widehat{\beta_1}x_i + \widehat{\varepsilon_i}^*$.
4. Auf der Basis von y_i^* wird $\widehat{\beta}$ erneut geschätzt.
5. Schritte 2–4 weren ein paar tausend Mal ausgeführt, sodass man die Verteilung der bootstrapten β-Schätzungen erhält.

In R-Code:

```r
# Wir werden 20'000 Bootstrapstichproben generieren.
runs <- 20000

# Und werden also 20'000 Bootstrapschätzungen pro Parameter haben.
# Diese speichern wir in einer Matrize mit 20'000 Zeilen und 2 Spalten.
bs_beta <- matrix(nrow = runs, ncol = 2)

for (i in 1:runs) {
  # Residuen des Modells bootstrappen (resampling with replacement)
  bs_residuals <- sample(resid(aoa.lm),
                         size = length(resid(aoa.lm)), # hier: 76
                         replace = TRUE)

  # Modellvorhersagen mit bootstrapten Residuen kombinieren
  bs_GJT <- predict(aoa.lm) + bs_residuals

  # Modell neu rechnen mit bootstrapten GJT-Daten.
  # d$AOA enthält die AOA-Daten des ursprunglichen Datensatzes
  # und zählt also 76 Beobachtungen.
```

```
  resampled.lm <- lm(bs_GJT ~ d$AOA)

  # Parameterschätzungen in neuer Zeile der Matrize speichern:
  bs_beta[i, ] <- coef(resampled.lm)
}
# Erste 6 Zeilen anzeigen:
head(bs_beta)
```

```
##          [,1]      [,2]
## [1,] 186.7644 -1.110948
## [2,] 188.9025 -1.198089
## [3,] 191.0417 -1.293858
## [4,] 191.2931 -1.193936
## [5,] 192.9736 -1.273488
## [6,] 200.2836 -1.490793
```

Die Verteilungen der Bootstrapschätzungen können wie gehabt mit Histogrammen gezeichnet werden; siehe Abbildung 8.10.

```
# Ergebnisse in dataframe giessen
bs_beta <- data.frame(bs_beta)
# und Spalten umbenennen
colnames(bs_beta) <- c("Schnittpunkt", "Steigung")

# Grafik zeichnen
bs_beta %>%
  # gather() stellt die Angaben, die jetzt in unterschiedlichen
  # Spalten stehen, in die gleiche Spalte. siehe ?gather
  gather(key = Parameter, value = Estimate) %>%
  ggplot(aes(x = Estimate)) +
  geom_histogram(fill = "lightgrey", col = "black", bins = 50) +
  # facet_wrap zeichnet separate Grafiken je nach (hier) Parameter
  facet_wrap(~ Parameter, scales = "free") +
  xlab("Bootstrapschätzung") +
  ylab("Anzahl")
```

Da diese Verteilungen normalverteilt aussehen, können die Konfidenzintervalle (hier: 90%) sowohl mit der `quantile()`- als auch mit der `qnorm()`-Funktion berechnet werden; die Ergebnisse sind einander nahezu identisch.

```
# Perzentilmethode
quantile(bs_beta$Schnittpunkt, probs = c(0.05, 0.95)) # Schnittpunkt
```

```
##       5%      95%
## 184.0250 196.6789
```

```
quantile(bs_beta$Steigung, probs = c(0.05, 0.95)) # Steigung
```

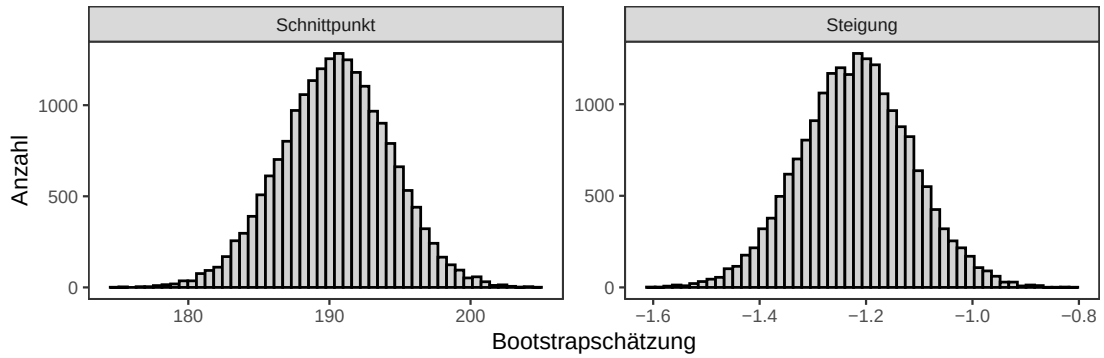


Abbildung 8.10: Verteilung der Bootstrap-Schätzungen der Parameter im Regressionsmodell `aoa.lm`.

```
##        5%       95%
## -1.391053 -1.045791

# Kürzer mit apply(). Die '2' heisst, dass die Funktion
# 'quantile' pro Spalte und nicht pro Zeile ausgeführt werden soll.
apply(bs_beta, 2, quantile, probs = c(0.05, 0.95))

##     Schnittpunkt  Steigung
## 5%      184.0250 -1.391053
## 95%     196.6789 -1.045791
```

Bemerken Sie, dass `coef(aoa.lm)` zwei Werte ausspuckt: die Schätzung des Schnittpunkts und die Schätzung der Steigung:

```
coef(aoa.lm)

## (Intercept)         AOA
##  190.408634   -1.217977
```

Um den ersten Wert zu selektieren, kann auch `coef(aoa.lm)[1]` verwendet werden, daher:

```
# qnorm
coef(aoa.lm)[1] + qnorm(c(0.05, 0.95)) * sd(bs_beta$Schnittpunkt)

## [1] 184.0356 196.7816

coef(aoa.lm)[2] + qnorm(c(0.05, 0.95)) * sd(bs_beta$Steigung)

## [1] -1.390275 -1.045678
```

Die Standardabweichungen der Bootstrapschätzungen schätzen den relevanten Standardfehler:

```
apply(bs_beta, 2, sd)

## Schnittpunkt     Steigung
```

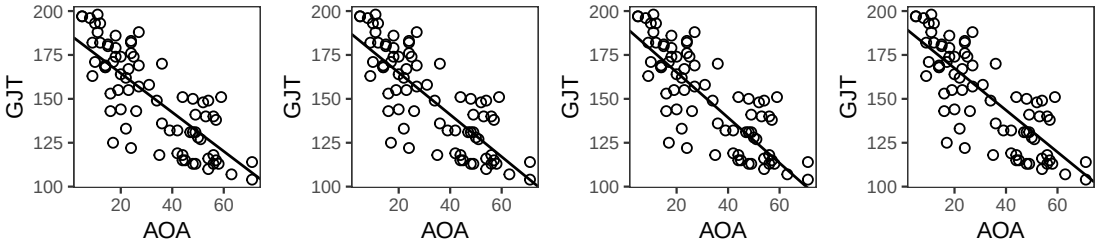


Abbildung 8.11: Regressionsgeraden, die anhand der 1., 2., 3. und 4. Bootstrapschätzungen des Schnittpunktes und der Steigung gezeichnet wurden. Die Grafiken sind einander recht ähnlich, aber nicht identisch.

```
##    3.8745063    0.1047499
```

Die Bootstrapschätzungen können auch verwendet werden, um die Unsicherheit in der Regressionsgeraden grafisch darzustellen. Wir können zum Beispiel die Bootstrapschätzung des Schnittpunktes und der Steigung abrufen und anhand derer Regressionsgeraden zeichnen. Der Übersichtlichkeit halber werden hier nur die ersten vier Bootstrapschätzungen gezeigt; die entsprechenden Regressionsgeraden sieht man in Abbildung 8.11:

```
# Die ersten 4 Bootstrapschätzungen für beta0 und beta1.
# Bei Ihnen werden diese Zahlen natürlich anders aussehen.
head(bs_beta, 4)
```

```
##   Schnittpunkt  Steigung
## 1     186.7644 -1.110948
## 2     188.9025 -1.198089
## 3     191.0417 -1.293858
## 4     191.2931 -1.193936
```

In Abbildung 8.12 mache ich nochmals das gleiche, aber diesmal mit 100 Schätzungen.[1] Wie Sie sehen können, kommen die Regressionsgeraden für durchschnittliche AOA-Werte einander ziemlich nahe, aber nahe den Minimum- und Maximumwerten fächern sie sich auf.

Anstatt das man diese Regressionsgeraden einzeln darstellt, färbt man in der Regel das Band, in das diese Geraden mehrheitlich fallen, ein. Analog zum Konfidenzintervall nennt man dieses dann ein **Konfidenzband**. Das Informationskästchen erklärt, wie man Konfidenzbänder mittels Bootstrapping konstruieren kann, aber dieses können Sie gerne überspringen.

Konfidenzbänder mit dem Bootstrap konstruieren. Die Idee ist die folgende. Man definiert eine Reihe von x-Werten, an denen man das Konfidenzband zeichnen möchte. In unserem Fall handelt es sich einfach um eine Handvoll Zahlen zwi-

[1] Man könnte es im Prinzip auch mit sämtlichen 20'000 Schätzungen machen.

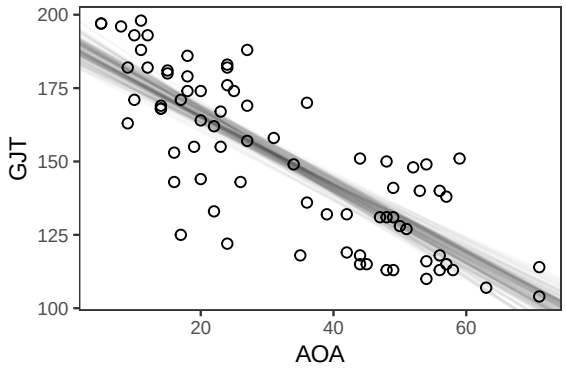


Abbildung 8.12: Regressionsgeraden, die auf der Basis von 100 Bootstrapschätzungen des Schnittpunktes und der Steigung gezeichnet wurden.

schen dem AOA-Minimum und dem AOA-Maximum. Es spielt hier keine grosse Rolle, wie viele Werte man festlegt.

```
neue_aoa <- seq(from = min(d$AOA), to = max(d$AOA), by = 1)
# also 5, 6, 7, etc., 69, 70, 71
```

Man nimmt die Bootstrapschätzungen des Schnittpunkts ($\widehat{\beta_0}^*$) und der Steigung ($\widehat{\beta_1}^*$) und man berechnet für jedes Paar von Schätzungen den $\hat{y}$-Wert für jeden x-Wert. Zum Beispiel ist (in meinem Fall) das erste Paar Bootstrapschätzungen:

```
bs_beta[1, ]
```

```
##   Schnittpunkt  Steigung
## 1     186.7644 -1.110948
```

Der Vektor von $\hat{y}$-Werten für dieses Paar von Bootstrapschätzungen ist daher:

```
# 186.76 + (-1.11) * 5, 186.76 + (-1.11) * 6, usw.
bs_beta[1, 1] + bs_beta[1, 2] * neue_aoa
```

```
##  [1] 181.2096 180.0987 178.9877 177.8768 176.7658 175.6549
##  [7] 174.5439 173.4330 172.3220 171.2111 170.1001 168.9892
## [13] 167.8782 166.7673 165.6563 164.5454 163.4345 162.3235
## [19] 161.2126 160.1016 158.9907 157.8797 156.7688 155.6578
## [25] 154.5469 153.4359 152.3250 151.2140 150.1031 148.9921
## [31] 147.8812 146.7702 145.6593 144.5483 143.4374 142.3264
## [37] 141.2155 140.1045 138.9936 137.8826 136.7717 135.6607
## [43] 134.5498 133.4389 132.3279 131.2170 130.1060 128.9951
## [49] 127.8841 126.7732 125.6622 124.5513 123.4403 122.3294
## [55] 121.2184 120.1075 118.9965 117.8856 116.7746 115.6637
## [61] 114.5527 113.4418 112.3308 111.2199 110.1089 108.9980
## [67] 107.8870
```

Diese Übung machen wir für alle Paare von Bootstrapschätzungen—in unserem

Fall also 20'000 Mal. Mit einem *for-loop* kann man dies übersichtlich tun. Da das Ergebnis jeder Iteration aber einen Vektor mit so vielen Elementen wie (hier) `neue_aoa` ist, ist es praktischer, diese Werte in einer Matrize zu speichern als in 20'000 Vektoren. (Diese Schritte kann man auch mit Matrizenalgebra ausführen, aber ich vermute, dass ein *for-loop* das Verfahren transparenter macht.)

```
# Matrize mit den Vorhersagen:
# wir brauchen 20'000 Zeilen (Anzahl Bootstraps)
# und so viele Spalten wie es vorherzusagende Werte pro Bootstrap gibt
bs_y_hat <- matrix(nrow = runs, # die Anzahl Bootstraps
                   ncol = length(neue_aoa)) # die Anzahl y-hat-Werte

# Für jedes Paar von Bootstrapschätzungen,
for (i in 1:runs) {
  # berechne die 'vorhergesagten' y-Werte für
  # jedes Element von neue_aoa und speichere diese
  # Werte in jeweils einer neuen Zeile der Matrize
  bs_y_hat[i, ] <- bs_beta[i, 1] + bs_beta[i, 2]*neue_aoa
}
```

Sie können diese Matrize mit etwa `head(bs_y_hat)` inspizieren. Um das 95%-Konfidenzband zu konstruieren, schlagen wir nun das 2.5. und das 97.5. Perzentil jeder Spalte nach. Die 2.5. Perzentile bilden die untere Grenze des Konfidenzbandes; die 97.5. die obere. Dazu verwende ich hier die `apply()`-Funktion, mit der man eine Funktion (hier `quantile()` mit dem Zusatzparameter `probs = 0.025`) bequem auf alle Spalten oder Zeilen einer Matrize (hier `bs_y_hat`) evaluieren kann. Die Zahl 2 spezifiziert, dass die Funktion pro Spalte evaluiert werden soll; `1` hiesse, dass sie pro Zeile zu evaluieren ist.

```
unten_95 <- apply(bs_y_hat, 2, quantile, probs = 0.025)
oben_95 <- apply(bs_y_hat, 2, quantile, probs = 0.975)
```

Das 80%-Konfidenzband würde man entsprechend so konstruieren:

```
unten_80 <- apply(bs_y_hat, 2, quantile, probs = 0.10)
oben_80 <- apply(bs_y_hat, 2, quantile, probs = 0.90)
```

Man kann auch noch das Mittel jeder Spalte berechnen—dies ergibt ungefähr die Regressionsgerade:

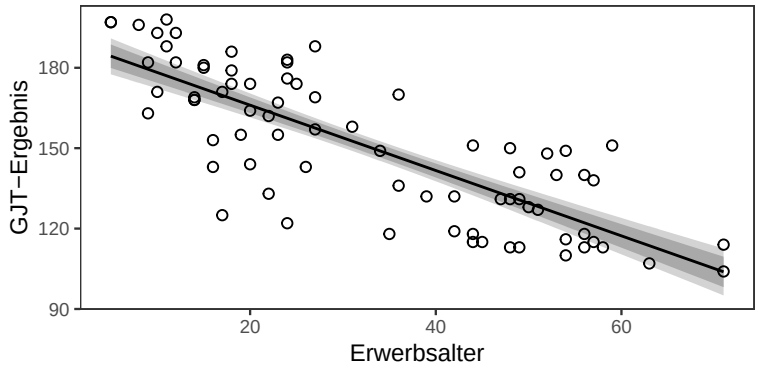
```
mittel <- apply(bs_y_hat, 2, mean)
```

Wenn man die Grafik mit `ggplot2()` zeichnen möchte, muss man diese Werte noch in ein data-frame giessen:

```
df_konfidenzband <- data.frame(neue_aoa,
                               mittel,
                               unten_95, oben_95,
                               unten_80, oben_80)
# head(df_konfidenzband)
```

Zeichnen kann man das Konfidenzband dann so:

```
ggplot(data = df_konfidenzband,
       aes(x = neue_aoa)) +
  # 95% Konfidenzband leicht
  geom_ribbon(aes(ymin = unten_95,
                  ymax = oben_95),
              fill = "lightgrey") +
  # 80% Konfidenzband dunkel
  geom_ribbon(aes(ymin = unten_80,
                  ymax = oben_80),
              fill = "darkgrey") +
  # Mittel (~ Regressionsgerade)
  geom_line(aes(y = mittel)) +
  # ev. auch noch die Rohdaten plotten.
  # Bemerken Sie, dass diese in einem
  # anderen data-frame stehen.
  geom_point(data = d,
             aes(x = AOA, y = GJT),
             shape = 1) +
  xlab("Erwerbsalter") +
  ylab("GJT-Ergebnis")
```



Das nächste Informationskästchen überspringen Sie aber besser nicht.

Identisch und unabhängig verteilte Restfehler. Beim Einschätzen der Unsicherheit in den Parameterschätzungen und in der Regressionsgeraden haben wir eine grundlegende Annahme gemacht, die bisher noch nicht diskutiert wurde. Beim

Bootstrappen haben wir auf der Basis der beobachteten Residuen zufällig neue Vektoren mit Residuen ($\widehat{\varepsilon}^*$) generiert und diese dann mit den $\widehat{y}$-Werten kombiniert. Dieser Schritt ist nur verteidigbar, wenn zwei Bedingungen gleichzeitig erfüllt sind:

1. Die Verteilung des Restfehler, inklusive ihre Streuung, ist für alle $\widehat{y}$-Werte gleich ('identisch verteilte Restfehler', 'Homoskedastizitätsannahme'). Zum Beispiel sollte es genauso plausibel sein, dass ein Restfehler von 25 auftaucht, wenn der $\widehat{y}$-Wert 120 ist als wenn er 180 ist. Sonst wäre es ja nicht sinnvoll gewesen, die Restfehler beim Bootstrappen komplett zufällig durcheinander zu werfen.
2. Die Restfehler bilden keine Klumpen. Anders gesagt, wenn wir den Restfehler einer bestimmten Beobachtung kennen, liefert uns dies nicht mehr Informationen über gewisse weitere Restfehler als über andere ('unabhängig verteilte Restfehler', 'Unabhängigkeitsannahme'). Wiederum wäre es sonst ja nicht sinnvoll gewesen, die Restfehler komplett zufällig durcheinander zu werfen, sondern hätten wir die Restfehler grüppchenweise neu zuordnen müssen.

Ein paar Beispiele, um diese Bedingungen anschaulicher zu machen:

- Es kann durchaus vorkommen, dass die Streuung um die Regressionsgerade systematisch zu- oder abnimmt für grössere $\widehat{y}$-Werte. Abbildung 8.13 auf der nächsten Seite zeigt zwei klare Beispiele.
- Jemand möchte die durchschnittliche Länge von [iː]-Produktionen von Bernern schätzen und wählt zufällig 25 Berner aus. (So weit, so gut.) Jeder Sprecher liest 50 Wörter mit einem [iː] vor. Die Vokallängen eines beliebigen Sprechers sind sich aber denkbar ähnlicher als die Vokallängen unterschiedlicher Sprecher: Manche Sprecher werden eher überdurchschnittlich lange Vokale produzieren, manche eher unterdurchschnittlich. Die Restfehler ('über-/unterdurchschnittlich') der einzelnen Produktionen sind also nicht unabhängig voneinander, sondern bilden pro Sprecher Klumpen.
- Ausserdem ist es wahrscheinlich, dass das [iː] in bestimmten phonologischen Kontexten unterschiedlich schnell ausgesprochen wird. Die Restfehler bilden also auch pro phonologischen Kontext (oder pro Wort) Klumpen.

Wenn die sogenannte 'i.d.d.'-Bedingung (*identically and independently distributed*) nicht erfüllt ist, dann ist es möglich, dass es effizientere Arten und Weisen gibt, um die Modellparameter zu schätzen. Damit ist Folgendes gemeint: Wenn Regressionsparameter mit der Methode der kleinsten Quadrate geschätzt werden, dann sind diese Schätzungen weder tendenziell Überschätzungen noch tendenziell Unterschätzungen – die Schätzungen sind also unverzerrt. Es gibt aber auch andere Methoden, um diese Parameter unverzerrt zu schätzen. Wenn die 'i.d.d.'-Bedingung erfüllt, ist es ausserdem so, dass die Methode der kleinste Quadrate

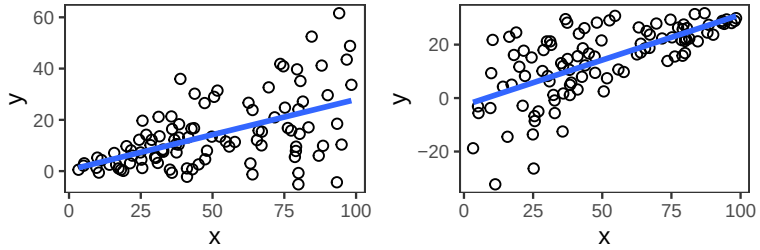


Abbildung 8.13: Die Streuung in beiden Streudiagrammen variiert erheblich je nach dem x- oder $\widehat{y}$-Wert.

Schätzungen liefert, die von Stichprobe zu Stichprobe am wenigsten voneinander abweichen. Ist die 'i.d.d.'-Bedingung nicht erfüllt, dann ist es möglich, dass eine andere unverzerrte Schätzungsmethode Schätzungen liefert, die von Stichprobe zu Stichprobe weniger variieren.

Wichtiger ist aber, dass die Schätzung der Unsicherheit betroffen ist. Insbesondere bei einer Verletzung der Unabhängigkeitsannahme wird die Unsicherheit in den Parameterschätzungen unterschätzt. Dies gilt nicht nur beim Bootstrappen, sondern auch beim Verwenden des zentralen Grenzwertzsatzes oder von t-Verteilungen. Im Beispiel mit den [i:] könnten wir also *nicht* den Standardfehler der durchschnittlichen Vokallänge berechnen, indem wir die Streuung in den Produktionen teilen durch die Wurzel von 1'250 (25×50).

Typische Verletzungen der Unabhängigkeitsannahme können mit sog. gemischten Modellen behoben werden; siehe Abschnitt 18.2 auf Seite 276.

Für Verletzungen der Homoskedastizitätsannahme bieten sich eine andere Erweiterung des linearen Modells an (siehe Zuur et al., 2009). Eine alternative Lösung ist, den Bootstrap anders durchzuführen (siehe Efron & Tibshirani, 1993, Abschnitt 9.5). Aber der wichtigste Grund, weshalb wir überhaupt den Bootstrap verwenden, ist, um die traditionellere Methoden besser zu verstehen, und diese angepassten Bootstraps sind für diesen Zweck weniger geeignet.

8.5.2 Bootstrappen unter der Normalitätsannahme

Wenn wir annehmen wollen, dass die Residuen nicht nur identisch und unabhängig, sondern auch noch normalverteilt sind, können wir die $\widehat{\varepsilon}^*$-Vektoren auch mit der `rnorm()`-Funktion generieren. Das Vorgehen ist komplett analog zu dem in Abschnitt 7.4.2 auf Seite 88 beschriebenen. Unsere Annahmen können expliziter gemacht werden:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i; \varepsilon_i \sim N(0, \sigma_\varepsilon^2) \tag{8.12}$$

Der neue Teil $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$ macht klar, dass wir annehmen, dass die Restfehler aus einer Normalverteilung mit Mittel 0 und Varianz σ_ε^2 stammen. σ_ε^2 ist ein einziger (obgleich

unbekannter) Wert, sodass klar ist, dass wir davon ausgehen, dass die Streuung der Residuen nicht von x (und somit $\hat{y}$ abhängt). Sowohl die Unabhängigkeits- als auch die Homoskedastizitätsannahme werden von dieser Annahme umfasst.

σ_ε ist zwar unbekannt, aber wird anhand der Stichprobe geschätzt; siehe Gleichung 7.4 auf Seite 89. Der folgende Code zeigt, wie die Bootstrapschätzungen des Schnittpunkts und der Steigung berechnet werden können. Abgesehen von der Konstruktion der $\hat{\varepsilon}^*$-Vektoren ist die Herangehensweise aber identisch zu jener des Bootstraps ohne die Normalitätsannahme aus dem letzten Abschnitt, sodass die anderen Schritte hier nicht mehr wiederholt werden.

```
# Wir werden 20'000 Bootstrapstichproben generieren.
runs <- 20000

# Und werden also 20'000 Bootstrapschätzungen für beide Parameter haben.
bs_beta <- matrix(nrow = runs, ncol = 2)

for (i in 1:runs) {
  # Residuen aus Normalverteilung generieren
  bs_residuals <- rnorm(n = length(resid(aoa.lm)), # hier: 76
                        mean = 0,
                        sd = sigma(aoa.lm))

  # Modellvorhersagen mit bootstrapten Residuen kombinieren
  bs_GJT <- predict(aoa.lm) + bs_residuals

  # Modell neu rechnen mit bootstrapten GJT-Daten.
  # d$AOA enthält die AOA-Daten des ursprunglichen Datensatzes
  # und zählt also 76 Beobachtungen.
  resampled.lm <- lm(bs_GJT ~ d$AOA)

  # Parameterschätzungen speichern
  bs_beta[i, ] <- coef(resampled.lm)
}
```

8.5.3 Mit t-Verteilungen

Wenn wir ohnehin davon ausgehen, dass die Residuen (i.d.d.) normalverteilt sind, können wir den Standardfehler, die Konfidenzintervalle und das Konfidenzband auch algebraisch anhand der t-Verteilungen berechnen. Dadurch wird auch die Unterschätzung von σ_ε durch $\widehat{\sigma_\varepsilon}$ mitberücksichtigt. Bei 76 Beobachtungen und bloss zwei Parameterschätzungen wird diese Unterschätzung aber kaum merkbar sein.

```
# Standardfehler der Parameterschätzungen nach t-Methode
summary(aoa.lm)
```

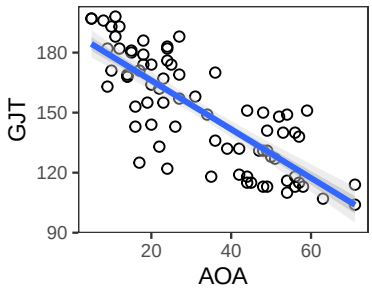


Abbildung 8.14: Regressionsgerade mit 67%- und 95%-Konfidenzband. Konfidenzbänder sind übrigens am schmalsten beim durchschnittlichen x-Wert.

```
##
## Call:
## lm(formula = GJT ~ AOA, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.703  -9.542  -0.260  13.021  32.452
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 190.4086     3.9040   48.77   <2e-16
## AOA          -1.2180     0.1051  -11.58   <2e-16
##
## Residual standard error: 16.4 on 74 degrees of freedom
## Multiple R-squared:  0.6446,Adjusted R-squared:  0.6398
## F-statistic: 134.2 on 1 and 74 DF,  p-value: < 2.2e-16
```

```
# 95%-Konfidenzintervalle nach t-Methode
confint(aoa.lm, level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 182.62969 198.187578
## AOA          -1.42747  -1.008484
```

Mit `geom_smooth()` können t-basierte Konfidenzbänder sofort gezeichnet werden, siehe Abbildung 8.14.

```
ggplot(data = d,
       aes(x = AOA,
           y = GJT)) +
  geom_point(shape = 1) +
  geom_smooth(method = "lm", level = 0.95, fill = "lightgrey") +
  geom_smooth(method = "lm", level = 0.67, fill = "darkgrey")
```

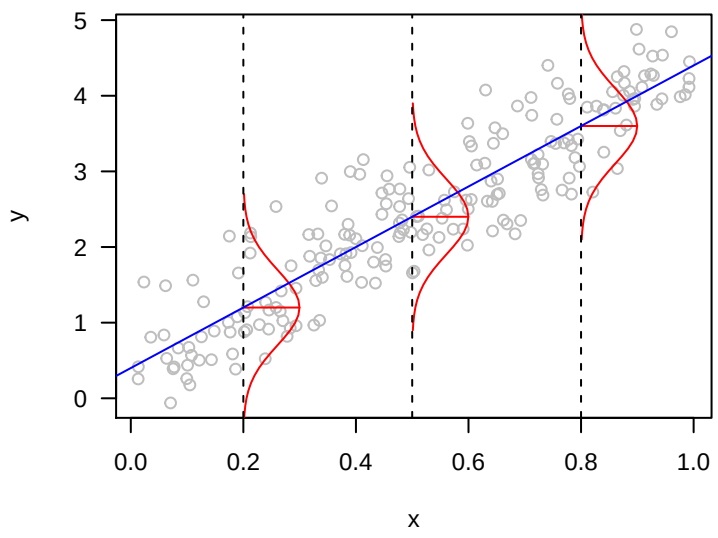


Abbildung 8.15: Wenn wir davon ausgehen, dass die Residuen i.d.d. verteilt sind, verbindet die Regressionsgerade die Mittel der y-Verteilungen für die unterschiedlichen x-Werte ('konditionelles Mittel'). In dieser Grafik sind die Residuen normalverteilt, aber dies ist keine Voraussetzung. Wenn die Residuen nicht normalverteilt sind, ist es jedoch möglich, dass das Mittel kein sehr relevantes Mass ist.

8.6 Regressionsgeraden interpretieren

Gleichung 8.12 auf Seite 120 ist nützlich, um die konzeptuelle Interpretation der Regressionsgeraden zu verstehen. Nach dieser Gleichung gehen wir davon aus, dass die Restfehler zufällig (und 'i.d.d.') aus einer Verteilung mit Mittel 0 stammen. Anders formuliert: Wir gehen davon aus, dass in der Population, die von Interesse ist, die y-Werte für jeden x-Wert normalverteilt sind. Folglich liegen auf der Geraden $\beta_0 + \beta_1 x_i$ die Mittel der y-Verteilung **konditionell** auf x. Abbildung 8.15 stellt dieses Konzept grafisch dar. Die geschätzte Regressionsgerade stellt also die auf Basis der Stichprobe geschätzten konditionellen Mittel von y dar, während der Querschnitt des 95%-Konfidenzbands an einem bestimmten x-Wert das 95%-Konfidenzintervall des y-Mittel für diesen x-Wert darstellt.

Beispiele:

- Laut dem `aoa.lm`-Modell sind die geschätzten β-Parameter 190.4 und −1.22. Laut dem Modell ist die beste Schätzung der durchschnittlichen (Mittel) GJT-Leistung von Versuchspersonen mit einem AOA von 15 also $190.4 - 1.22 \times 15 = 172.1$. Dieses Ergebnis erhält man auch mit `predict()`:

```
predict(aoa.lm,
        newdata = data.frame(AOA = 15))

##       1
## 172.139
```

Bemerken Sie aber, dass es in unserem Datensatz zwei Versuchspersonen mit einem AOA von 15 gibt und dass ihr Durchschnittsergebnis nicht 172.1 ist:

```
d %>% filter(AOA == 15)
```

```
## # A tibble: 2 x 2
##     AOA   GJT
##   <dbl> <dbl>
## 1    15   180
## 2    15   181
```

Inwiefern unsere modellbasierte Schätzung eine zuverlässigere Schätzung des konditionellen Mittels darstellt als das Mittel dieser beiden Werte, hängt von der Gültigkeit unserer Annahmen ab. Die Annahme eines linearen Zusammenhangs scheint hier doch auf jeden Fall nicht wahnsinnig daneben zu liegen. Konzeptuell erlaubt uns die Annahme, y-Mittel für bestimmte x-Werte besser zu schätzen, indem wir auch Information über den x–y-Zusammenhang, die wir aus den restlichen Daten ableiten, mit einbeziehen.

- Versuchspersonen mit einem AOA von 21 gibt es in der Stichprobe nicht. Nach der Regressionsgleichung wäre aber das Durchschnittsergebnis von Versuchspersonen mit diesem AOA in der Population etwa 165 Punkte.

```
predict(aoa.lm,
        newdata = data.frame(AOA = 21))
```

```
##        1
## 164.8311
```

Dies ist ein Beispiel von **Intrapolation**, denn es gibt sowohl Versuchspersonen mit niedrigeren als mit höheren AOA-Werten in der Stichprobe.

- Versuchspersonen mit einem AOA von 82 gibt es in der Stichprobe auch nicht. Nach der Regressionsgleichung wäre aber das Durchschnittsergebnis von Versuchspersonen mit diesem AOA in der Population etwa 91 Punkte.

```
predict(aoa.lm,
        newdata = data.frame(AOA = 82))
```

```
##        1
## 90.53455
```

Dies ist ein Beispiel von **Extrapolation**, denn das Maximumalter in der Stichprobe ist 71 Jahre.

- Das durchschnittliche (Mittel) AOA in der Stichprobe ist etwa 32.5 Jahre. Das geschätzte konditionelle GJT-Mittel für dieses Alter ist gleich dem Mittel der Stichprobe.

```
predict(aoa.lm,
        newdata = data.frame(AOA = mean(d$AOA)))
```

```
##        1
## 150.7763
```

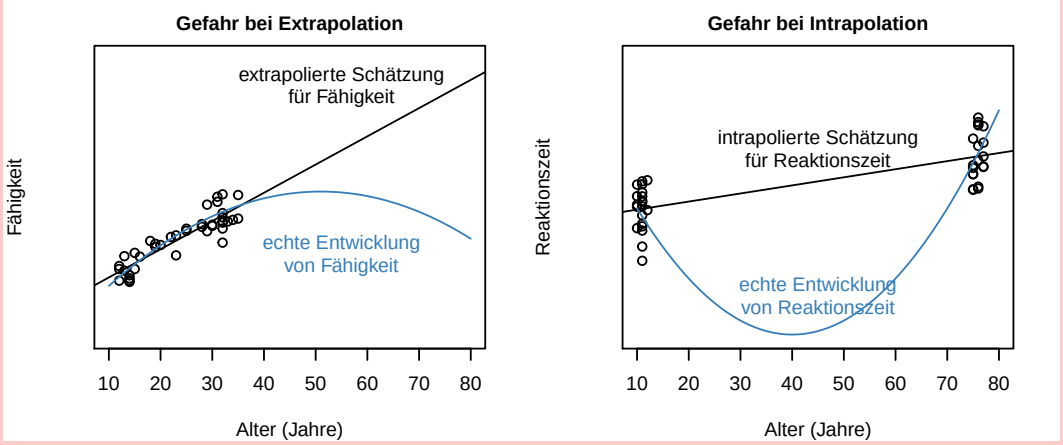
Dies ist natürlich kein Zufall, sondern ein allgemeines Phänomen.

- Konfidenzintervalle um konditionelle Mittel können mit den oben beschriebenen Bootstrapmethoden berechnet werden, indem man das 'Konfidenzband' für nur einen x-Wert berechnet. Man kann aber auch `predict()` verwenden; dann wird das Konfidenzintervall auf der Basis der geeigneten t-Verteilung konstruiert.

```
predict(aoa.lm,
        newdata = data.frame(AOA = 35),
        interval = "confidence",
        level = 0.80)
```

```
##        fit      lwr      upr
## 1 147.7795 145.3246 150.2343
```

Extra- und Intrapolation. Seien Sie vorsichtig mit Extrapolation: Wenn wir eine Stichprobe von Versuchspersonen zwischen 8 und 26 Jahren haben, ist es gefährlich, Aussagen über 5- oder 40-Jährige zu machen. Dies wird in der linken Abbildung illustriert: Eine Fähigkeit, die sich im Alter zwischen 10 und 35 entwickelt, hat nicht unbedingt die gleiche Entwicklung ausserhalb dieses Bereichs. Eine Extrapolierung auf der Basis der Regressionsgeraden ist hier irreführend. Auch bei **Intrapolation** ist Vorsicht geboten. Aus den Daten in der rechten Grafik könnte man zum Beispiel die Schlussfolgerung ziehen, dass sich Reaktionszeiten im Alter graduell verlängern. Auch diese Schlussfolgerung dürfte zu kurz greifen.



Den Schnittpunkt interpretierbarer machen. Der geschätzte Schnittpunkt hat nicht unbedingt eine nützliche Interpretation. In unserem Fall stellt er die geschätzte Durchschnittsleistung von Versuchspersonen mit AOA 0 dar. Solche gibt es in der

Stichprobe nicht, sodass diese Zahl eine Art Extrapolation darstellt. Sie können den Schnittpunkt interpretierbarer machen, indem Sie das Stichprobenmittel des Prädiktors von den Prädiktorwerten abziehen und mit diesen neuen Werten arbeiten. Diese Technik heisst **zentrieren** (*centring*). Ein Vorteil des Zentrierens ist, dass der geschätzte Schnittpunkt einem jetzt sofort auch sagt, was das Stichprobenmittel der y-Variable ist.

```
# AOA zentrieren
d$c.AOA <- d$AOA - mean(d$AOA)

# Modell neu fitten
aoa.lm <- lm(GJT ~ c.AOA, data = d)

# Parameterschätzungen
summary(aoa.lm)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 150.776316  1.8807316  80.16897 1.120538e-73
## c.AOA        -1.217977  0.1051385 -11.58450 2.728150e-18
```

Übrigens ist das 95%-Konfidenzintervall um das Stichprobenmittel dank der Berücksichtigung des Zusammenhangs zwischen AOA und GJT auch schmaler geworden: $[147.0, 154.5]$ verglichen mit $[144.5, 157.0]$ vorher.

```
confint(aoa.lm)
```

```
##                 2.5 %     97.5 %
## (Intercept) 147.02888 154.523755
## c.AOA        -1.42747  -1.008484
```

Achten Sie aber darauf, dass eine Versuchsperson mit einem AOA von 35 jetzt für das Modell eine Versuchsperson mit einem `c.AOA` von $35 - \bar{x}_{AOA} = 2.46$ ist:

```
predict(aoa.lm,
        newdata = data.frame(c.AOA = 35 - mean(d$AOA)),
        interval = "confidence",
        level = 0.80)
```

```
##        fit      lwr      upr
## 1 147.7795 145.3246 150.2343
```

8.7 Modellannahmen überprüfen

Die Modellresiduen sollten grafisch dargestellt werden, um die Modellannahmen zu überprüfen. Leitfragen dabei sind unter anderem:

- Gibt es noch einen erkennbaren Zusammenhang zwischen den Residuen und den $\hat{y}$-Werten? Ein solcher Zusammenhang deutet darauf hin, dass der Zusammen-

hang zwischen einem oder mehreren Prädiktoren und dem *outcome* nicht-linear ist.

- Variiert die Streuung der Residuen mit $\hat{y}$ oder mit den Prädiktoren? Systematische Unterschiede in der Streuung der Residuen deuten darauf hin, dass der Restfehler 'heteroskedastisch' ist.
- Sind die Residuen ungefähr normalverteilt? Nicht-normalverteilte Residuen lassen vermuten, dass die Annahme, dass der Restfehler aus einer Normalverteilung stammt, nicht stimmt. Dies hätte einerseits Konsequenzen für die auf t-Verteilungen basierten Konfidenzintervalle und Konfidenzbänder. Andererseits sind die konditionellen Mittel, die die Regressionslinie darstellt, eventuell weniger relevant.
- Gibt es einzelne Datenpunkte, die einen viel stärkeren Einfluss aufs Regressionsmodell ausüben als die meisten? Das Problem mit einflussreichen Datenpunkten ist, dass sie etwa dazu führen können, dass das Modell einen leichten positiven Zusammenhang zwischen den Variablen findet, während für die meisten Datenpunkte ein starker negativer Zusammenhang vorliegt.

Abbildung 8.16 zeigt ein paar nützliche Grafiken, die man einfach mit `plot(aoa.lm)` generieren kann. Auf riesige Probleme in den Modellannahmen deuten diese Grafiken m.E. nicht hin.

```
# 4 Grafiken in 2x2-Raster zeichnen.
# (Dies funktioniert nicht für ggplot!)
par(mfrow = c(2, 2))
# Modelldiagnosen darstellen
plot(aoa.lm)
# Ab jetzt wieder normal zeichnen.
par(mfrow = c(1, 1))
```

Aufgrund des Stichprobenfehlers wird man oft—rein durch Zufall—Zusammenhänge und nicht-normalverteilte Residuen finden, sodass man ein bisschen Erfahrung braucht, um unbedeutende Muster in den Residuen von potenziellen Problemen zu unterscheiden. Ausserdem sind Modellannahmen gerade bei kleineren Stichproben schwieriger zu überprüfen. Für mehr Informationen hierzu, siehe *Checking model assumptions without getting paranoid* (25.4.2018) und Vanhove (2018a). Siehe ausserdem *Before worrying about model assumptions, think about model relevance* (11.4.2019).

Das Thema Modellkritik wird weiter behandelt von unter anderem Baayen (2008), Cohen et al. (2003, Kapitel 4), Faraway (2005, Kapitel 4), Weisberg (2005, Kapitel 8–9) und Zuur et al. (2009, Kapitel 2).

Denkfrage. Obwohl die Modelldiagnose nicht auf grössere Probleme hindeutet, *können* die Modellannahmen eigentlich gar nicht stimmen. Erklären Sie.

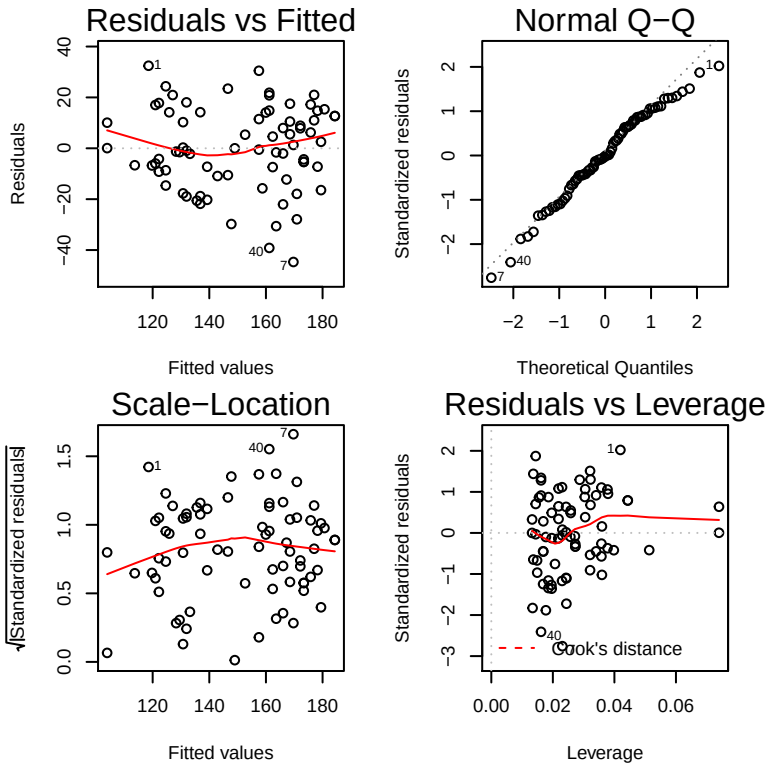


Abbildung 8.16: Grafische Modelldiagnose des `aoa.lm`-Modells mithilfe der `plot()`-Funktion. Links oben: Zusammenhang zwischen Residuen und $\hat{y}$. Die rote Trendlinie sollte ungefähr flach sein. Sonstige auffällige Muster wären auch unerwünscht. Rechts oben: Normalität der Residuen. Wenn die Residuen normalverteilt sind, liegen sie auf der gestrichelten Diagonale. Links unten: Streuung in den Residuen. Eine flache rote Trendlinie deutet auf Homoskedastizität hin. Rechts unten: Manchmal gibt es in dieser Grafik ein paar gestrichelte rote Linien. Datenpunkte, die jenseits dieser Linien liegen, dürften viel einflussreicher als andere Datenpunkte sein.

8.8 Übungen

1. Führen Sie folgende Analyse auf die dekeyser2010.csv-Daten aus:

```
plot(AOA ~ GJT, data = d)
mod2.lm <- lm(AOA ~  GJT, data = d)
summary(gjt.lm)
```

   (a) Erklären Sie, was Sie gerade berechnet haben. Was bedeuten die geschätzten Parameter? Wieso ist das Intercept so gross? Was bedeutet das Intercept?

   (b) Welches Modell finden Sie am sinnvollsten: `aoa.lm` oder `gjt.lm`? Warum?

2. Die Datei `vanhove2014_cognates.csv` enthält eine Zusammenfassung der Daten meiner Dissertation (Vanhove, 2014). 163 Deutschschweizer Versuchspersonen wurden gebeten, 45 geschriebene und 45 (andere) gesprochene schwedische Wörter ins Deutsche zu übersetzen. Die Anzahl richtiger Antworten steht in den Spalten `CorrectWritten` bzw. `CorrectSpoken`. Die Datei `vanhove2014_background.csv` enthält Angaben zur Leistung der Versuchspersonen bei weiteren Sprach- und kognitiven Tests. Fügen Sie die beiden Datensätze zusammen.

   Versuchen Sie, die folgenden Forschungsfragen zu beantworten.

   (a) `DS.Span` enthält die Leistung der Versuchspersonen bei einem Arbeitsgedächtnistest. Wie hängt die Leistung bei `DS.Span` mit `CorrectSpoken` zusammen?

   (b) Wie hängt die Leistung bei einem Englischtest (`English.Overall`) mit der Übersetzungsleistung in der geschriebenen Modalität zusammen?

   (c) Wie variiert die Übersetzungsleistung in den beiden Modalitäten mit dem Alter (`Age`) der Versuchspersonen?

Kapitel 9

Gruppenunterschiede

Im letzten Kapitel haben wir uns mit der Frage beschäftigt, wie man den Zusammenhang zwischen einem kontinuierlichen Prädiktor und einem kontinuierlichen *outcome* modellieren kann. In diesem Kapitel widmen wir uns der Frage, wie Zusammenhänge zwischen **kategorischen** Prädiktoren und einem kontinuierlichen *outcome* modelliert werden können. Das typische Beispiel eines kategorischen Prädiktors ist die Kondition in einem Experiment: Wie stark unterscheiden sich die Ergebnisse von Teilnehmenden in der Experimentalgruppe im Schnitt von jenen von Teilnehmenden in der Kontrollgruppe? Das Vorgehen ist nahezu identisch mit dem aus dem letzten Kapitel.

9.1 Unterschiede zwischen zwei Gruppen

Das Beispiel, dem wir uns hier widmen, stammt nicht aus der Sprachwissenschaft, sondern aus der Sozialpsychologie. Caruso et al. (2013, Experiment 1) berichteten, dass amerikanische Versuchspersonen Aussagen, die die US-amerikanische Sozialsystem rechtfertigen, stärker zustimmen, wenn man sie an Geld erinnert (sogenanntes *currency priming*). Ihr Design sah wie folgt aus. Es gab acht Aussagen im Stil von *Everyone has a fair shot at wealth and happiness*. Die Teilnehmenden deuteten am Bildschirm ihre Zustimmung zu diesen Aussagen auf einer 7-stufigen Likertskala an (1 = überhaupt nicht einverstanden, 7 = vollständig einverstanden). Pro Versuchsperson wurden die acht Zustimmungswerte gemittelt. Die Hälfte der Versuchspersonen sah im Hintergrund ein verblasstes aber ersichtliches Bild einer Banknote; bei der anderen Hälfte war dieses Bild verwischt. Die Versuchspersonen, welche die Banknote im Hintergrund sahen, stimmten den Aussagen stärker zu, als jene, bei denen das Bild verwischt war.

Klein et al. (2014) versuchten, dieses Ergebnis in 36 neuen Stichproben zu replizieren. In der Datei `Klein2014_money_abington.csv` finden Sie die Ergebnisse einer dieser Stichproben (84 Teilnehmende). Diese Daten werden wir analysieren.

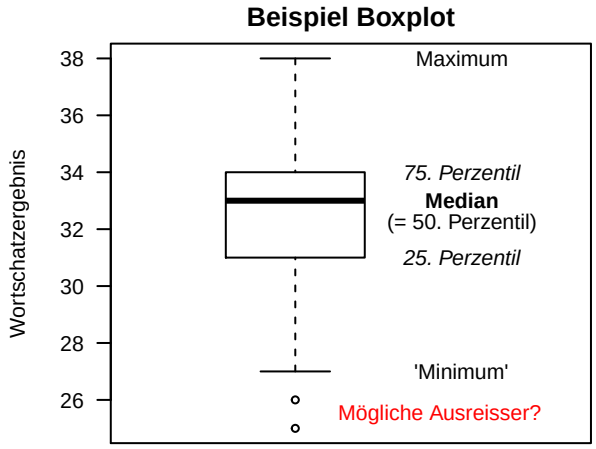


Abbildung 9.1: Erklärung Boxplot.

Aufgabe. Lesen Sie diese Datei in R ein. (Den Datensatz können Sie einfach `d` nennen.) Vergessen Sie nicht zu kontrollieren, ob das Einlesen geklappt hat.

9.1.1 Grafische Darstellung: Boxplots

Eine nützliche grafische Darstellung, um unterschiedliche Gruppen hinsichtlich einer mehr oder weniger kontinuierlichen *outcome* zu vergleichen, ist der Boxplot (zu Deutsch auch *Kastengrafik*). Abbildung 9.1 zeigt als Beispiel einen Boxplot der Ergebnisse bei einem Wortschatztest der 80 Versuchspersonen von Vanhove (2016). Die dickere Linie in der Mitte liegt beim Median und das Kästchen reicht vom 25. bis zum 75. Perzentil und umfasst somit die Hälfte der Datenpunkte. Manchmal gibt es (wie hier) auch Kreischen in einem Boxplot. Diese stellen Extremwerte dar, die mehr als 1.5 $\times$ die Distanz zwischen dem 25. und dem 75. Perzentil vom 25. oder 75. entfernt liegen (siehe `?boxplot` $\rightarrow$ `Arguments` $\rightarrow$ `range`). Diese Extremwerte sind *mögliche* (!) Ausreisser. Mit einem Dotplot kann man besser überprüfen, ob sie tatsächlich auch Ausreisser sind. (In diesem Beispiel liegen die zwei möglichen Ausreisser nicht sehr weit von anderen Datenpunkten entfernt, sodass sie nicht als Ausreisser gelten.)

Mit `ggplot()` können solche Boxplots wie folgt erzeugt werden; das Ergebnis steht in Abbildung 9.2.

```
p_boxplot <- ggplot(data = d,
                    aes(x = MoneyGroup,
                        y = Sysjust)) +
  geom_boxplot() +      # Daten als Boxplot darstellen
  xlab("Kondition") +
  ylab("Systemrechtfertigungsscore")
# Wenn die Grafik als Objekt gespeichert wurde (wie oben),
# kann man sie zeichnen, indem man einfach ihren Namen eintippt.
```

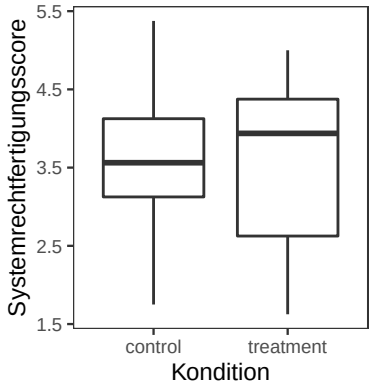


Abbildung 9.2: Vergleich der Systemrechtfertigungsscores in den beiden Konditionen in Klein et al.'s (2014) Replikation von Caruso et al. (2013, Experiment 1). Daten aus der Abington-Stichprobe.

```
p_boxplot
```

Ich finde es oft eine gute Idee, dem Boxplot auch noch die einzelnen Datenpunkte hinzuzufügen (siehe auch Weissgerber et al., 2015). Insbesondere bei eher kleinen Datensätzen beeinträchtigt dies die Interpretierbarkeit der Grafik nicht und hilft es den Lesenden, einzuschätzen, wie die Daten tatsächlich verteilt sind. (Boxplots können in dieser Hinsicht manchmal täuschen.) Der Code unten zeigt Ihnen, wie Sie dies machen können; siehe Abbildung 9.3 auf der nächsten Seite.

```
p_boxplotdeluxe <- ggplot(data = d,
                     aes(x = MoneyGroup,
                         y = Sysjust)) +
  # 'outlier.shape = NA' verhindert, dass allfällige
  # Extremwerte zwei Mal dargestellt werden: als Kreischen
  # beim Boxplot und als einzelner Datenpunkt
  geom_boxplot(outlier.shape = NA) +
  # 'shape = 1': Kreischen
  # 'position_jitter(width = x)': Die Datenpunkte horizontal
  #                               etwas verschieben.
  geom_point(shape = 1,
             position = position_jitter(width = 0.2, height = 0)) +
  xlab("Kondition") +
  # Die Konditionen verständlicher bezeichnen.
  scale_x_discrete(labels = c("ohne Banknote",
                              "mit Banknote")) +
  ylab("Systemrechtfertigungsscore")
p_boxplotdeluxe
```

Siehe auch meinen Blogeintrag *Some alternatives to bar plots* (7.1.2015).

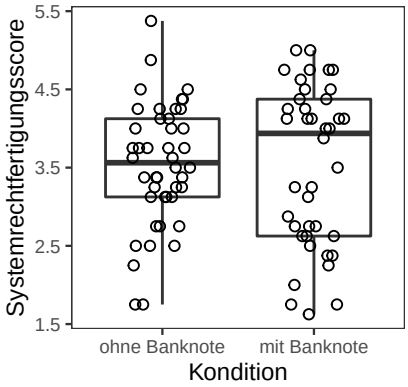


Abbildung 9.3: Nochmals die gleichen Daten, aber mit den einzelnen Datenpunkten.

9.1.2 Numerische Zusammenfassung

```
d %>%
  group_by(MoneyGroup) %>%
  summarise(AnzahlVpn = n(),
            Mittel = mean(Sysjust),
            Median = median(Sysjust),
            StdAbw = sd(Sysjust))
```

```
## # A tibble: 2 x 5
##   MoneyGroup AnzahlVpn Mittel Median StdAbw
##   <chr>          <int>  <dbl>  <dbl>  <dbl>
## 1 control           44   3.53   3.56  0.781
## 2 treatment         40   3.53   3.94  1.02
```

9.1.3 Modellierung

Dummy-Variable

Die grafische Darstellung zeigt, dass der Median der 'mit Banknote'-Kondition zwar höher ist als jener der 'ohne Banknote'-Kondition, aber das die Überlappung zwischen beiden Konditionen erheblich ist. Die numerische Zusammenfassung zeigt ausserdem, dass sich die Mittel kaum unterscheiden. Trotzdem können wir diese Daten – ähnlich wie im letzten Kapitel – in ein Modell giessen. Dieses Modell wird ebenfalls von Gleichung 8.7 auf Seite 108 beschrieben, die hier wiederholt wird:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{9.1}$$

y_i stellt nun den Systemrechtfertigungsscore der i. Versuchsperson dar; x_i stellt die Gruppenzugehörigkeit dieser Versuchsperson dar. Genau wie vorher müssen β_0 und β_1 (und als Konsequenz davon auch ε_i) geschätzt werden.

Gleichungen wie diese können natürlich schwer umgehen mit Wörtern wie 'control' und 'treatment', aber die Lösung ist erstaunlich einfach: Eine Gruppe bezeichnen wir als 0 und die andere als 1. Zum Beispiel können wir festlegen, dass $x_i = 0$, wenn die i. Versuchsperson zur 'control'-Kondition gehört, und dass $x_i = 1$, wenn sie zur 'treatment'-Kondition gehört.[1] Wenn wir dies gemacht haben, können wir die Reihe von Einsen und Nullen als Prädiktor in ein lineares Regressionsmodell aufnehmen. Wenn man kategorische Variable (hier: Gruppenzugehörigkeit) als Zahlenreihen umschreibt, spricht man von **Dummy-Variablen**.

```
# Dummy-Variable kreieren
d$n.Kondition <- ifelse(d$MoneyGroup == "treatment",
                        yes = 1, # n.Kondition wird 1, wenn MoneyGroup == "treatment"
                        no = 0)  # sonst wird n.Kondition 0
```

```
# Regressionsmodell mit Dummy-Variabler fitten
money.lm <- lm(Sysjust ~ n.Kondition, data = d)
```

Wie gehabt können die geschätzten Parameter abgerufen werden, indem man den Namen des Modells eintippt.

```
money.lm
```

```
##
## Call:
## lm(formula = Sysjust ~ n.Kondition, data = d)
##
## Coefficients:
## (Intercept)  n.Kondition
##    3.534091    -0.005966
```

Die zwei Parameterschätzungen sind $\widehat{\beta_0}$ bzw. $\widehat{\beta_1}$. Ihre Bedeutung kann aus der Regressionsgleichung abgeleitet werden:

$$y_i = 3.53 - 0.006x_i + \widehat{\varepsilon_i} \tag{9.2}$$

Für Versuchspersonen in der Kontrollgruppe ist $x_i = 0$. Daher wird die Gleichung zu $y_i = 3.53 - 0.006 \times 0 + \widehat{\varepsilon_i} = 3.53 + \widehat{\varepsilon_i}$, sodass $\widehat{y_i} = 3.53$. $\widehat{\beta_0}$ ist also das Gruppenmittel der Gruppe, die als 0 bezeichnet wurde.

Für Versuchspersonen in der Experimentalgruppe ist $x_i = 1$. Daher wird die Gleichung zu $y_i = 3.53 - 0.006 \times 1 + \widehat{\varepsilon_i}$, sodass $\widehat{y_i} = 3.53 - 0.006$. (Durch Ründungsfehler ergibt dies eigentlich auch 3.53.) $\widehat{\beta_1}$ ist also der *Unterschied* zwischen den Mitteln der beiden Gruppen. Ist dieser Wert negativ, dann hat die Gruppe, die als 1 bezeichnet wurde, ein niedrigeres Mittel als die Gruppe, die als 0 bezeichnet wurde.

[1] Wir könnten auch festlegen, dass $x_i = 1$ für Versuchspersonen in der Kontrollkondition und $x_i = 0$ für Versuchspersonen in der Experimentalkondition. Das macht eigentlich nichts aus.

Aufgabe Ändern Sie die Befehle oben, sodass nun die Kontrollgruppe als 1 bezeichnet wird und die Experimentalgruppe als 0. Was ändert sich im Output?

Unsicherheit in den Parameterschätzungen quantifizieren

Den minimalen Unterschied zwischen den zwei Gruppenmitteln hätten wir auch einfach von Hand berechnen können. Der Mehrwert des allgemeinen linearen Modells besteht aber darin, dass wir auch die Unsicherheit in den Parameterschätzungen schätzen können; dies machen wir hier. Ausserdem können dem allgemeinen linearen Modell mehrere Prädiktoren hinzugefügt werden; dies machen wir in einem nächsten Kapitel.

Bootstrappen ohne Normalitätsannahme. Die 'neuen' y-Werte (y^*) stellen sich aus den vom Modell 'vorhergesagten' y-Werten ($\hat{y}$) und einer Bootstrap-Stichprobe aus $\hat{\varepsilon}$ zusammen (*sampling with replacement*).[2]

```
# Bootstrapping ohne Normalitätsannahme
runs <- 20000
bs_beta <- matrix(nrow = runs, ncol = 2)

for (i in 1:runs) {
  neu_Sysjust <- predict(money.lm) + sample(resid(money.lm), replace = TRUE)
  bs_money.lm <- lm(neu_Sysjust ~ n.Kondition, data = d)
  bs_beta[i, ] <- coef(bs_money.lm)
}
```

Siehe Seite 113 für eine Erklärung:

```
bs_beta <- data.frame(bs_beta)
colnames(bs_beta) <- c("Schnittpunkt", "Unterschied")

bs_beta %>%
  gather(key = Parameter, value = Estimate) %>%
  ggplot(aes(x = Estimate)) +
  geom_histogram(fill = "lightgrey", col = "black", bins = 50) +
  facet_wrap(~ Parameter, scales = "free") +
  xlab("Bootstrapschätzung") +
  ylab("Anzahl")
```

[2] In diesem Beispiel können $\hat{\varepsilon}$-Werte aus der Kontrollkondition auch neu der Experimentkondition zugeordnet werden und umgekehrt. Dies entspricht der Homoskedastizitätsannahme (siehe Seite 119) des allgemeinen linearen Modells: Die Fehlervarianz ist überall gleich gross, sodass ein bestimmter Restfehler genau so gut in der anderen Kondition hätte vorkommen können. Man könnte den Bootstrap aber auch so programmieren, dass $\hat{\varepsilon}$-Werte aus der Kontrollkondition nur der Kontrollkondition zugewiesen werden können. Hiermit berücksichtigt man die Möglichkeit, dass die Fehlervarianz in der einen Gruppe abweichen könnte von jener in der anderen Gruppe.

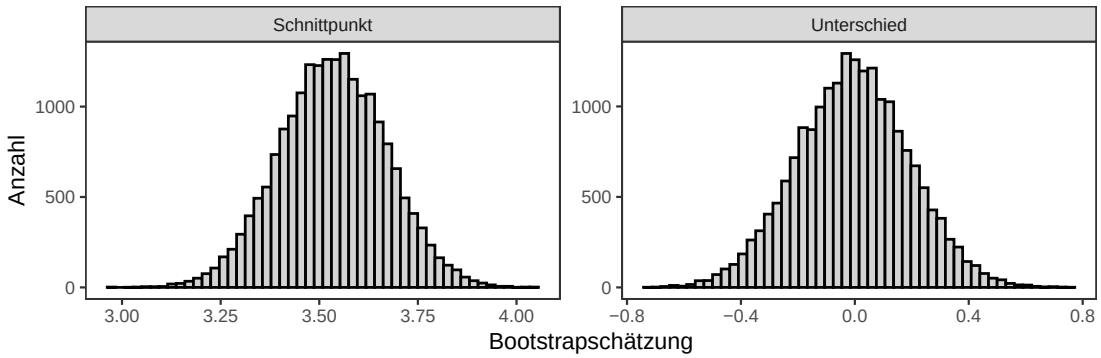


Abbildung 9.4: Verteilung der Bootstrap-Schätzungen der Parameter im Regressionsmodell `money.lm`.

Die Standardabweichungen der Bootstrapverteilungen von $\widehat{\beta}$ können wiederum als Schätzungen der Standardfehler dienen:

```
apply(bs_beta, 2, sd)
```

```
## Schnittpunkt  Unterschied
##    0.1348586    0.1972699
```

Ebenso können Konfidenzintervalle berechnet werden. In Fällen wie diesen interessiert man sich in der Regel hauptsächlich für den Standardfehler und das Konfidenzintervall um die Unterschiedsschätzung:

```
# 80% Konfidenzintervall
quantile(bs_beta$Unterschied, probs = c(0.1, 0.9))
```

```
##        10%        90%
## -0.2601069  0.2447533
```

Bootstrappen ohne Normalitätsannahme. Ähnlich wie in den letzten zwei Kapiteln kann man die Residuen auch aus einer Normalverteilung generieren.

```
# Bootstrapping mit Normalitätsannahme
runs <- 20000
bs_beta <- matrix(nrow = runs, ncol = 2)
for (i in 1:runs) {
  neu_Sysjust <- predict(money.lm) + rnorm(n = nrow(d), sd = sigma(money.lm))
  bs_money.lm <- lm(neu_Sysjust ~ n.Kondition, data = d)
  bs_beta[i, ] <- coef(bs_money.lm)
}
```

Die Histogramme werden nicht nochmals gezeichnet.

```
bs_beta <- data.frame(bs_beta)
colnames(bs_beta) <- c("Schnittpunkt", "Unterschied")
```

```
apply(bs_beta, 2, sd)
```

```
## Schnittpunkt  Unterschied
##    0.1358459    0.1967508
```

```
quantile(bs_beta$Unterschied, probs = c(0.1, 0.9))
```

```
##        10%        90%
## -0.2540017  0.2482426
```

Mit t-Verteilungen. Wenn man ohnehin davon ausgehen will, dass der Restfehler aus einer Normalverteilung stammt, kann man wiederum die `summary()`-Funktion verwenden, um die Standardfehler abzurufen:

```
summary(money.lm)
```

```
##
## Call:
## lm(formula = Sysjust ~ n.Kondition, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90312 -0.77812  0.09091  0.71591  1.84091
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.534091   0.136330   25.92   <2e-16
## n.Kondition -0.005966   0.197561   -0.03    0.976
##
## Residual standard error: 0.9043 on 82 degrees of freedom
## Multiple R-squared:  1.112e-05,Adjusted R-squared:  -0.01218
## F-statistic: 0.0009119 on 1 and 82 DF,  p-value: 0.976
```

Die Konfidenzintervalle um $\widehat{\beta}$ können mit `confint()` abgerufen werden:

```
confint(money.lm, level = 0.8)
```

```
##                 10 %      90 %
## (Intercept)  3.357958 3.7102240
## n.Kondition -0.261207 0.2492752
```

Der minimale Unterschied zwischen den Gruppenmitteln von bloss -0.006 Punkten auf einer 7er-Skala hat also ein 80%-Konfidenzintervall von [-0.26, 0.25] und könnte demnach fast genau so gut positiv als auch negativ sein. In dieser Stichprobe bestätigt sich also das Ergebnis eines positiven Unterschiedes von Caruso et al. (2013) nicht. Die Bootstrapmethoden liefern ein nahezu identisches Ergebnis.[3]

[3] An dieser Stelle sei darauf hingewiesen, dass der *outcome* in dieser Studie strikte genommen kei-

9.1.4 *Treatment coding* und *sum-coding*

Wenn man, wie oben, eine Gruppe als 0 und die andere als 1 bezeichnet, spricht man von ***treatment coding***. Das Intercept stellt dann das Mittel der 0-Gruppe dar und die Steigung den Unterschied zwischen den Gruppenmitteln.

Eine Alternative ist ***sum-coding***. Hierzu wird eine Gruppe als -0.5 und die andere als 0.5 bezeichnet:

```
d$n.Kondition <- ifelse(d$MoneyGroup == "treatment",
                        yes = 0.5,
                        no = -0.5)
money.lm <- lm(Sysjust ~ n.Kondition, data = d)
money.lm

##
## Call:
## lm(formula = Sysjust ~ n.Kondition, data = d)
##
## Coefficients:
## (Intercept)  n.Kondition
##    3.531108    -0.005966
```

$\widehat{\beta_1}$ stellt nach wie vor den Unterschied zwischen den beiden Gruppenmitteln dar, aber das Intercept ($\widehat{\beta_0}$) stellt nun den **Gesamtmittelwert** (*grand mean*) dar. Dies ist das Mittel der Gruppenmittel. Achtung: Dies ist nicht unbedingt das Mittel sämtlicher Daten!

Aufgabe Manche Forschende verwenden bei sum-coding lieber -1 und 1 als -0.5 und 0.5. Was würde sich im Output ändern, wenn man dies machen würde? Was bezeichnet der geschätzte Parameter für `n.Kondition` jetzt?

Alabama first. Eigentlich braucht man die Dummy-Variablen nicht selber zu kreieren: R macht dies automatisch, wenn Sie eine nicht-numerische Variable direkt dem Modell hinzufügen:

ne kontinuierliche Variable ist, sondern das Mittel ordinalskalierter Beobachtungen. Liddell & Kruschke (2018) und Bürkner & Vuorre (2018) argumentieren, dass das allgemeine lineare Modell gar nicht geeignet ist, um solche Daten auszuwerten. Sagen Sie mir daher bitte Bescheid, wenn Sie über einen Datensatz verfügen, in dem zwei Gruppen hinsichtlich ihres Mittels bei einer *echten* kontinuierlichen Variablen verglichen werden, ohne dass weitere Prädiktoren berücksichtigt werden müssen – denn ich habe keine solchen Datensatz, der als Beispiel dienen kann.

```
money.lm2 <- lm(Sysjust ~ MoneyGroup, data = d)
money.lm2

##
## Call:
## lm(formula = Sysjust ~ MoneyGroup, data = d)
##
## Coefficients:
##         (Intercept)  MoneyGrouptreatment
##            3.534091            -0.005966
```

Defaultmässig hantiert R *treatment coding*. Aber Achtung: Welche Gruppe als 0 bezeichnet wird und welche als 1, wird nach dem Alphabet festgelegt. 'treatment' kommt nach 'control', sodass 'control' die 0-Gruppe wird und 'treatment' die 1-Gruppe. Verlieren Sie dies bitte nicht aus dem Auge: Wenn Ihr Datensatz auf Deutsch zusammengestellt wurde, käme in einer Variablen namens `L1` 'Deutsch' vor 'Französisch'; auf Englisch käme aber 'German' nach 'French'.

9.1.5 Annahmen überprufen

Die wichtigsten Annahmen dieses Modells sind:

1. Die Datenpunkte sind unabhängig voneinander. Ein klassisches Beispiel von Datensatzen mit *abhängigen* Datenpunkte sind Erhebungen in unterschiedlichen Schulklassen: Kinder aus derselben Klasse sind sich ähnlicher (aufgrund vorheriger Selektion, gemeinsamer Lehrkräfte, usw.) als Kinder aus unterschiedlichen Klassen. Die Konsequenz davon ist, dass Kinder aus derselben Klasse dem Modell keine vollständig neue Information hinzufügen. Das Modell 'weiss' dies aber nicht und würde deswegen fälschlicherweise davon ausgehen, dass jeder Eintrag den gleichen Informationswert hat. Dadurch würde der Standardfehler unterschätzt. Für weitere Diskussion und Lösungen, siehe Vanhove (2015).

   Ein anderes Beispiel sind *within-subject*-Experimente, in denen Versuchspersonen in beiden/mehreren Konditionen getestet werden. *Within-subject*-Experimente bieten in der Regel mehr statistische Genauigkeit, sind aber schwieriger zu analysieren. Eine Option ist die Verwendung gemischter Modelle (siehe etwa Baayen et al., 2008; Baayen, 2008). Wenn alle Versuchspersonen in zwei Konditionen getestet werden und es nur zwei Konditionen gibt, ist eine einfache Option, den Wert jeder Versuchsperson in der einen Kondition von ihren Wert in der anderen abzuziehen und diese Unterschiede zu analysieren.

   Die Unabhängigkeitsannahme lässt sich meistens schwer überprüfen, sodass ihre Gültigkeit auf der Basis von Sachwissen eingeschätzt werden muss: Man muss eben *wissen*, ob die Datenpunkte Klümpchen bilden oder nicht.

2. Die Restfehler stammen aus einer Normalverteilung. In diesem Fall heisst dies,

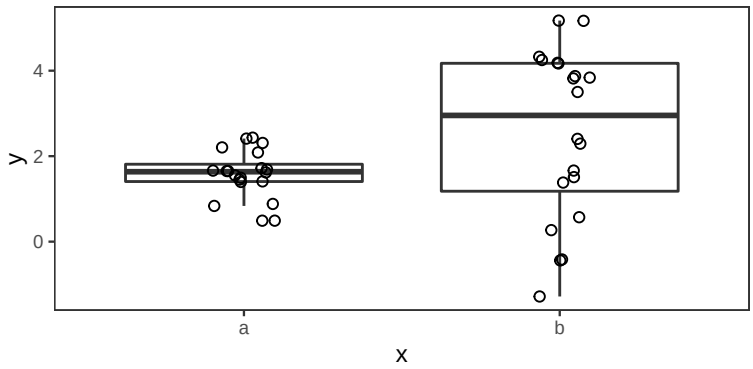


Abbildung 9.5: Beispiel von heteroskedastischen Daten.

dass die outcome-Werte *innerhalb* jeder Kondition etwa normalverteilt sind, aber in komplexeren Modellen müsste man hierzu die Verteilung der Restfehler selber anschauen. Die Konstruktion der Konfidenzintervalle anhand von t-Verteilungen setzt normalverteilte Restfehler voraus, aber wie das Beispiel oben zeigt, kann eine Bootstrapmethode, die eben keine normalverteilten Restfehler voraussetzt, recht ähnliche Ergebnisse liefern, insbesondere, wenn die Datenmenge ausreichend ist. M.E. wichtiger ist aber, dass Restfehler, die nicht ungefähr normalverteilt sind, darauf hinweisen, dass die Gruppenmittel, die man eben vergleicht, keine relevanten Masse der zentralen Tendenz sind. Siehe *Before worrying about model assumptions, think about model relevance* (11.4.2019).

3. Die Restfehler haben in jeder Gruppe die gleiche Streuung (Homoskedastizität). In etwa scheint dies in diesem Fall schon zu stimmen. Zum Vergleich: Die Boxplots in Abbildung 9.5 (mit zwei gleich grossen Gruppen, aber erheblichen Streuungsunterschieden) wären ein Grund, sich über diese Annahme Sorgen zu machen. Mögliche Lösungen finden sich bei Zuur et al. (2009, Kapitel 4). Eine andere Lösung wäre, dass man den Bootstrap so durchführt, dass die Residuen der einen Gruppe nicht der anderen Gruppe zugeordnet werden.

Zur Überprüfung dieser Annahmen, siehe Vanhove (2018a).

9.2 Unterschiede zwischen mehreren Gruppen

Vanhove (2017) untersuchte, inwiefern die Genuszuordnungen im Deutschen (Heisst es *der*, *die* oder *das Knie*?) bei flämischen DialektsprecherInnen von ihrem Dialekt beeinflusst werden. Zum Beispiel: Sagen InformantInnen, in deren Dialekt *knie* männlich ist, eher *der Knie*, verglichen mit InformantInnen, in deren Dialekt *knie* weiblich ist? Er kam zum Schluss, dass dies kaum der Fall ist, sodass sich die Frage stellte, woran dies liegt. Eine Vermutung war, dass den DialektsprecherInnen metalinguistische Kenntnisse über Genuszuordnungen in ihrem eigenen Dialekt fehlen, sodass sie nicht auf diese zurückgreifen können, wenn sie deutsche Nomen einem Genus zuordnen. Van-

hove (2018b) überprüfte diese Vermutung, indem er flämischen DialektsprecherInnen metalinguistische Informationen über ihren Dialekt verschaffte. Konkret gab es drei Konditionen:

- Versuchspersonen in der ersten Kondition wurde eine Strategie erklärt, mit der sie das Genus eines Wortes in ihrem Dialekt erschliessen können.
- Versuchspersonen in der zweiten Kondition wurde mitgeteilt, dass ihr Dialekt (nicht aber das Standardniederländische) die gleiche Anzahl Genera wie das Deutsche hat. Ihnen wurde aber nicht erklärt, wie sie das Genus eines Wortes erschliessen können.
- Versuchspersonen in der dritten Kondition erhielten Informationen über einen irrelevanten Aspekt ihres Dialektes. Diese Kondition dient als Kontrollkondition.

Dann wurde geschaut, ob die Genuszuordnungen sich zwischen den Konditionen unterschieden. Getestet wurden 29 deutsche Nomen mit niederländischen Kognaten und es wurde gezählt, wie viele Genuszuordnungen im Deutschen pro Versuchsperson kongruent mit dem Genus des Kognats in ihrem Dialekt waren. Erwartet wurde insbesondere, dass Versuchspersonen in der ersten Kondition (STRATEGY) sich nach den Instruktionen eher am Dialekt orientieren als Versuchspersonen in den beiden anderen Konditionen (INFORMATION und NO INFORMATION).

Lesen Sie die Daten ein. Die Spalte `ProportionCongruent` listet die Proportion kongruenter Genuszuordnungen pro Versuchsperson auf.[4]

```
d <- read_csv("Data/Vanhove2018.csv")
```

9.2.1 Grafische Darstellung

Ob zwei oder mehrere Gruppen, die Boxplots kann man mit dem gleichen Befehl zeichnen. Das Einzige, was ich hier anders mache, ist, dass ich mit `scale_x_discrete()` die Konditionen in einer Reihenfolge aufliste, die m.E. sinnvoller ist als die alphabetische.

```
ggplot(d,
       aes(x = Condition,
           y = ProportionCongruent)) +
  geom_boxplot(outlier.shape = NA) +
  geom_point(shape = 1,
             position = position_jitter(width = 0.2, height = 0)) +
  scale_x_discrete(limits = c("no information", "information", "strategy")) +
  xlab("Lernkondition") +
  ylab("Proportion L1-L2 kongruenter\nAntworten")
```

[4] Auch dieser *outcome* ist strikte genommen nicht kontinuierlich; siehe Fussnote 3.

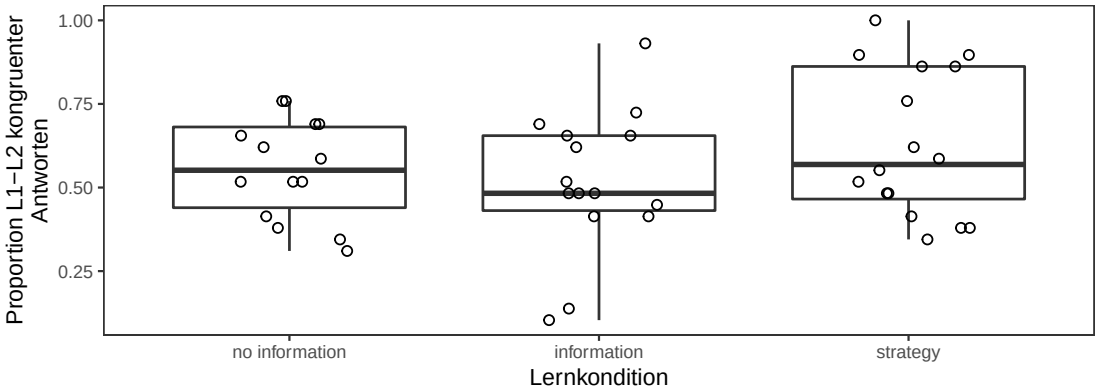


Abbildung 9.6: Boxplots der Daten von Vanhove (2018b).

9.2.2 Numerische Zusammenfassung

```
d %>%
  group_by(Condition) %>%
  summarise(AnzahlVpn = n(),
            Mittel = mean(ProportionCongruent),
            Median = median(ProportionCongruent),
            StdAbw = sd(ProportionCongruent))
```

```
## # A tibble: 3 x 5
##   Condition      AnzahlVpn Mittel Median StdAbw
##   <chr>              <int>  <dbl>  <dbl>  <dbl>
## 1 information           15  0.517  0.483  0.213
## 2 no information        14  0.554  0.552  0.150
## 3 strategy              16  0.627  0.569  0.219
```

9.2.3 Modellierung

Eine kategorische Variable mit zwei Ausprägungen konnten wir als eine binäre Dummy-Variable (0 vs. 1) umkodieren. Auch um eine kategorische Variable mit drei Ausprägungen in einem allgemeinen linearen Modell zu modellieren, brauchen wir Dummy-Variablen, aber wie viele? Man könnte die Zugehörigkeit zu jeder Kondition binär festlegen:

Kondition	Strategy?	Information?	(NoInformation?)
Strategy	1	0	(0)
Information	0	1	(0)
No information	0	0	(1)

Bemerken Sie aber, dass wir die letzte Spalte gar nicht brauchen: Wenn in der `Strategy?`- und in der `Information?`-Spalte eine Null steht, wissen wir schon, dass in der `NoInformation?`-

Spalte eine Eins folgt. Wir brauchen also nur zwei Dummy-Variablen.

```
# Dummy-Variablen Strategy und Information kreieren:
d$Strategy <- ifelse(d$Condition == "strategy", 1, 0)
d$Information <- ifelse(d$Condition == "information", 1, 0)

# Kontrolle:
head(d)
```

```
## # A tibble: 6 x 5
##   SubjectID  Condition ProportionCongr~ Strategy Information
##   <chr>      <chr>                <dbl>    <dbl>       <dbl>
## 1 59b197ec2~ strategy             0.586        1           0
## 2 59b3a2ae8~ no infor~            0.759        0           0
## 3 59b948264~ informat~            0.517        0           1
## 4 59b966ec6~ strategy             0.862        1           0
## 5 59b9b2065~ informat~            0.655        0           1
## 6 59bbff205~ no infor~            0.517        0           0
```

Das allgemeine lineare Modell kann problemlos mit mehreren Prädiktoren umgehen, sodass wir beide Prädiktoren dem Modell hinzufügen können. Die Modellgleichung schaut so aus:

$$y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + \varepsilon_i \tag{9.3}$$

$x1_i$ stellt den Wert der i. Versuchsperson bei der ersten Dummy-Variablen dar (z.B. `Information`: 0 oder 1); $x2_i$ stellt ihren Wert bei der zweiten Dummy-Variablen dar (z.B. `Strategy`: 0 oder 1).

```
mod.lm <- lm(ProportionCongruent ~ Information + Strategy, data = d)
mod.lm
```

```
##
## Call:
## lm(formula = ProportionCongruent ~ Information + Strategy, data = d)
##
## Coefficients:
## (Intercept)  Information     Strategy
##     0.55419     -0.03695      0.07297
```

Aufgabe 1 Vergleichen Sie die geschätzten Parameter mit den Werten in der numerischen Zusammenfassung oben. Was stellt die Parameterschätzung für `(Intercept)` (also $\widehat{\beta_0}$) dar? Was bedeuten die Parameterschätzungen für `Information` ($\widehat{\beta_1}$) und `Strategy` ($\widehat{\beta_2}$)?

Aufgabe 2 Dem Modell kann die kategorische Variable der Konditionzugehörigkeit auch direkt hinzufügt werden. Warum ergibt dies andere Parameterschätzungen?

```
mod.lm2 <- lm(ProportionCongruent ~ Condition, data = d)
mod.lm2
```

```
##
## Call:
## lm(formula = ProportionCongruent ~ Condition, data = d)
##
## Coefficients:
##             (Intercept)  Conditionno information
##                 0.51724                  0.03695
##       Conditionstrategy
##                 0.10991
```

9.2.4 Die Unsicherheit in den Parameterschätzungen quantifizieren

Die Bootstrapmethoden sollen mittlerweile ihrem didaktischen Zweck gedient haben, weshalb wir sie hier überspringen. Die Standardfehler der Parameterschätzungen können wie gehabt mit `summary()` abgerufen werden:

```
summary(mod.lm)
```

```
##
## Call:
## lm(formula = ProportionCongruent ~ Information + Strategy, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41379 -0.14039 -0.03448  0.13793  0.41379
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.55419    0.05301  10.455 2.92e-13
## Information -0.03695    0.07370  -0.501    0.619
## Strategy     0.07297    0.07258   1.005    0.320
##
## Residual standard error: 0.1983 on 42 degrees of freedom
## Multiple R-squared:  0.05562,Adjusted R-squared:  0.01065
## F-statistic: 1.237 on 2 and 42 DF,  p-value: 0.3006
```

Konfidenzintervalle lassen sich einfach mit `confint()` berechnen. Die Konfidenzintervalle um $\widehat{\beta_0}$ sind in der Regel weniger interessant, aber werden automatisch mitberechnet.

```
confint(mod.lm, level = 0.90)
```

```
##                      5 %       95 %
## (Intercept)  0.46503431 0.64334007
## Information -0.16090786 0.08701624
## Strategy    -0.04910963 0.19504559
```

Wir könnten hier schlussfolgern, dass der Einfluss der metalinguistischen Instruktion eher ungewiss bleibt. Zwar gaben Versuchspersonen in der STRATEGY-Kondition mehr kongruente Antworten als jene in der NO INFORMATION-kondition (7 Prozentpunkte mehr), aber die Unsicherheit in dieser Schätzung ist von der Grösse, dass auch negative Unterschiede oder Unterschiede nahe bei Null recht plausibel sind (vgl. das Konfidenzintervall).

9.2.5 Andere Kodierungssysteme

In dieser Analyse wurde *treatment coding* verwendet, sodass die Parameterschätzungen für β_1 und β_2 stets in Bezug zum Schnittpunkt (β_0) zu interpretieren sind. Dieser Schnittpunkt stellt den $\widehat{y}$-Wert für Versuchspersonen in der Gruppe, die als (0,0) kodiert wurde, dar.

Manchmal sind aber andere Kodierungssysteme nützlich. Zum Beispiel könnte es sinnvoll sein, die Parameter so schätzen zu lassen, dass $\widehat{\beta_2}$ den Unterschied zwischen der dritten und der *zweiten* Gruppe und nicht den Unterschied zwischen der dritten und der *ersten* Gruppe darstellt. Dies sei hier der Vollständigkeit halber erwähnt; mehr Informationen finden Sie unter auf der Website des Institute for Digital Research and Education der UCLA. Eine detaillierte Behandlung findet sich in Schad et al. (2018).

9.2.6 Annahmen überprüfen

Die Annahmen beim Vergleichen mehrerer Gruppen sind identisch mit den Annahmen beim Vergleichen von zwei Gruppen.

In Vanhove (2018b) wurden diese Daten übrigens anders analysiert, da der *outcome* eigentlich nicht kontinuierlich ist, sondern sich aus 29 binären Antworten pro Versuchsperson zusammensetzt. Diese Analyse führt aber zu den gleichen Schlussfolgerungen.

9.3 Übungen

1. Berthele (2012) spielte 155 angehenden Lehrpersonen eine Lautaufnahme vor, angeblich von einem Jungen, der Französisch als Fremdsprache spricht. Anschliessend wurden sie gebeten, das akademische Potenzial des Buben einzuschätzen

(Skala von 1–6). In etwa der Hälfte der Fälle enthielt die Lautaufnahme Codeswitches (also ein paar deutsche Wörter); in den anderen nicht. Ausserdem wurde der Hälfte der angehenden Lehrpersonen erzählt, der Junge hiesse Luca (ein gängiger Name in der Deutschschweiz); der anderen Hälfte wurde erzählt, er hiesse Dragan (ein Name, der man eher mit dem Balkan assoziiert). Die Frage war, ob dieses Labelling die Einschätzungen der angehenden Lehrpersonen beeinflusst, ggf. in Kombination mit den Codeswitches.

Hier fokussieren wir uns zunächst auf der Frage, wie gross der Unterschied in der durchschnittlichen Bewertung für Aufnahmen mit dem Dragan- und dem Luca-Label ist, wenn die Aufnahme Codeswitches enthält.

(a) Lesen Sie die Datei `berthele2011.csv` in R ein.

(b) Filtern Sie die Aufnahmen ohne Codeswitches heraus, sodass nur die Aufnahmen mit Codeswitches übrig bleiben.

(c) Analysieren Sie die Bewertungen hinsichtlich der Frage, ob diese vom 'Luca'- vs. 'Dragan'-Label beeinflusst werden. (Vergessen Sie nicht, die Daten grafisch darzustellen!)

(d) Fassen Sie Ihre Befunde in 2–3 Sätzen zusammen.

2. Ein anderer Befund, den Klein et al. (2014) zu replizieren versuchten, war der *gambler's fallacy*, der in einem Experiment von Oppenheimer & Monin (2009) belegt wurde. Klein et al. (2014) fassen dieses Experiment zusammen:

> Oppenheimer & Monin (2009) investigated whether the rarity of an independent, chance observation influenced beliefs about what occurred before that event. Participants imagined that they saw a man rolling dice in a casino. In one condition, participants imagined witnessing three dice being rolled and all came up 6's. In a second condition two came up 6's and one came up 3. In a third condition, two dice were rolled and both came up 6's. All participants then estimated, in an open-ended format, how many times the man had rolled the dice before they entered the room to watch him. Participants estimated that the man rolled dice more times when they had seen him roll three 6's than when they had seen him roll two 6's or two 6's and a 3. For the replication, the condition in which the man rolls two 6's was removed leaving two conditions.

Die Daten der Replikationsstudie finden Sie in der Datei `Klein2014_gambler.csv`. Analysieren Sie den Datensatz hinsichtlich der Forschungsfrage, aber beschränken Sie sich dabei auf die Stichprobe der University of Florida (`ufl` in der Spalte `Sample`). Fassen Sie Ihre Befunde in 2–3 Sätzen zusammen.

3. Wählen Sie im selben Datensatz eine beliebige andere Stichprobe aus und wiederholen Sie Ihre Analyse.

Kapitel 10

Interaktionen

Oft ist es nicht sosehr der Zusammenhang zwischen dem *outcome* und diesem oder jenem Prädiktor, der uns interessiert: Vielmehr sind wir am Zusammenspiel von zwei oder mehreren Prädiktoren interessiert. Zum Beispiel ist es nicht so interessant, ob man schneller auf hochfrequente als auf seltene Wörter reagiert – dieser Befund ist längst Gemeingut. Und es ist auch nicht so interessant, ob gute Lesende schneller auf bestehende Wörter reagieren als schlechte Lesende – auch das liegt auf der Hand. Interessanter wäre die Frage, ob der Effekt von Wortfrequenz unterschiedlich gross ist je nach der Lesefähigkeit der Versuchspersonen. Dies ist eine Frage nach der **Interaktion** zwischen Lesefähigkeit und Wortfrequenz (Beispiel inspiriert von der Forschungsfrage von Kuperman & Van Dyke, 2013).

In Abbildung 10.1 auf der nächsten Seite werden drei von vielen möglichen Interaktionsmustern aufgeführt. Ihr gemeinsames Merkmal ist, dass die gezeichneten Linien nicht parallel laufen; bei der Absenz einer Interaktion ist dies schon der Fall. In der Grafik links oben liegt aber keine Interaktion zwischen Lesefähigkeit und Wortfrequenz vor: Beide Variablen hängen zwar mit der Lesegeschwindigkeit zusammen, aber der Zusammenhang zwischen Lesefähigkeit und Lesegeschwindigkeit unterscheidet sich nicht je nach Wortfrequenz (oder umgekehrt).

Ein mit 'Interaktion' verwandter Begriff ist **Haupteffekt**. Ein Haupteffekt eines Prädiktors, z.B. Wortfrequenz, auf Lesegeschwindigkeit liegt dann vor, wenn es, gemittelt über die Ausprägungen des anderen Prädiktors, z.B. Lesefähigkeit, einen Effekt des ersten Prädiktors gibt. In der Grafik links oben gibt es also einen Haupteffekt von Lesefähigkeit: Gemittelt über die Ausprägungen von Wortfrequenz lesen beschlagene Lesende schneller als schlechtere Lesende (gut: $\frac{4.5+6.5}{2} = 5.5$; schlecht: $\frac{3+5}{2} = 4$). In dieser Grafik gibt es auch einen Haupteffekt von Wortfrequenz: Gemittelt über die Ausprägungen von Lesefähigkeit werden hochfrequente Wörter schneller gelesen als Wörter mit niedriger Frequenz (hoch: $\frac{5+6.5}{2} = 5.75$; niedrig: $\frac{3+4.5}{2} = 3.75$).

Auch in der Grafik rechts oben gibt es Haupteffekte von sowohl Lesefähigkeit (gut: $\frac{4.5+9}{2} = 6.75$; schlecht: $\frac{3+5}{2} = 4$) als auch von Wortfrequenz auf Lesegeschwindigkeit (hoch: $\frac{5+9}{2} = 7$; niedrig: $\frac{3+4.5}{2} = 3.75$). Dies gilt auch für die Grafik links unten (gut:

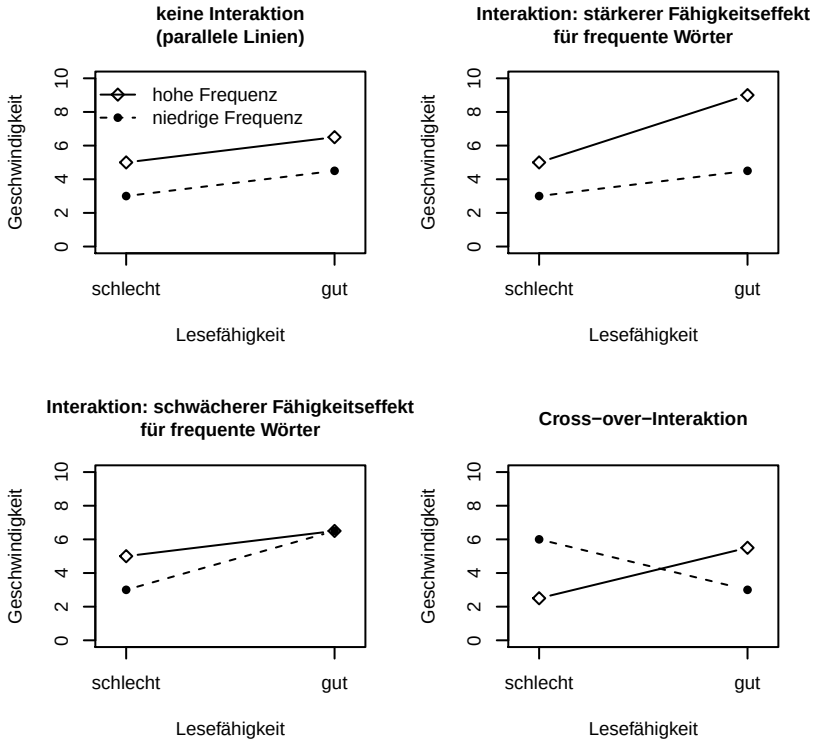


Abbildung 10.1: Wenn eine Interaktion zwischen Lesefähigkeit und Wortfrequenz auf Lesegeschwindigkeit vorliegt, dann unterscheidet sich der Effekt von Lesefähigkeit auf Lesegeschwindigkeit je nach Wortfrequenz. Umgekehrt gilt ebenfalls, dass sich der Effekt von Wortfrequenz auf Lesegeschwindigkeit je nach Lesefähigkeit unterscheidet. Dies zeigt sich in den nicht-parallelen Linien. (Die Einheiten entlang der y-Achse sind in diesem Beispiel arbiträr.)

$\frac{6.5+6.5}{2} = 6.5$; schlecht: $\frac{3+5}{2} = 4$; hoch: $\frac{5+6.5}{2} = 5.75$; niedrig: $\frac{3+6.5}{2} = 4.75$).

In der letzten Grafik liegt ebenfalls einen Haupteffekt von Wortfrequenz vor (hoch: $\frac{2.5+5.5}{2} = 4$; niedrig: $\frac{6+3}{2} = 4.5$). Es gibt aber keinen Haupteffekt von Lesefähigkeit: Wenn man über die beiden Ausprägungen von Wortfrequenz mittelt, zeigt sich ein Nulleffekt (gut: $\frac{3+5.5}{2} = 4.25$; schlecht: $\frac{2.5+6}{2} = 4.25$). Die letzte Grafik ist ebenfalls ein Beispiel einer ***cross-over***-Interaktion, denn die Effektslinien kreuzen sich.

10.1 Interaktionen zwischen zwei binären Prädiktoren

Eigentlich interessierte sich Berthele (2012) (Übungen letztes Kapitel) eher für die Interaktion zwischen dem Vorkommen (oder nicht) von Codeswitches und dem angeblichen Namen des Buben auf die Bewertungen seines akademischen Potenzials: Werden Codeswitches als gravierender betrachtet, wenn der Bub einen typischen Balkannamen hat als wenn er einen typischen schweizerischen Namen hat?[1]

```
d <- read_csv("Data/berthele2011.csv")
```

10.1.1 Grafische Darstellung (I)

Wir können wieder Boxplots zeichnen, um die Muster in den Daten zu veranschaulichen. Mit `facet_grid(. ~CS)` wird die Grafik horizontal in zwei Teile aufgespaltet: einen für die Fälle, in denen Codeswitching vorlag, und einen für Fälle, in denen dies nicht der Fall war. Wie Abbildung 10.2 aber zeigt, dürften diese Daten etwas zu grobkörning sein, um mit Boxplots dargestellt zu werden. Unten werden ein paar andere Grafiken gezeichnet, aber um diese zu zeichnen, müssen wir die Daten zuerst numerisch zusammenfassen.

```
ggplot(d,
       aes(x = Name,
           y = Potenzial)) +
  geom_boxplot(outlier.shape = NA) +
  geom_point(shape = 1,
             position = position_jitter(width = 0.2, height = 0)) +
  facet_grid(. ~ CS)
```

10.1.2 Numerische Zusammenfassung

Mit `group_by()` kann man problemlos den Datensatz nach mehreren Variablen gruppieren, um so die Daten innerhalb jeder **Zelle** des Designs zusammenzufassen.

[1] Siehe abermals Fussnote 3 auf Seite 138.

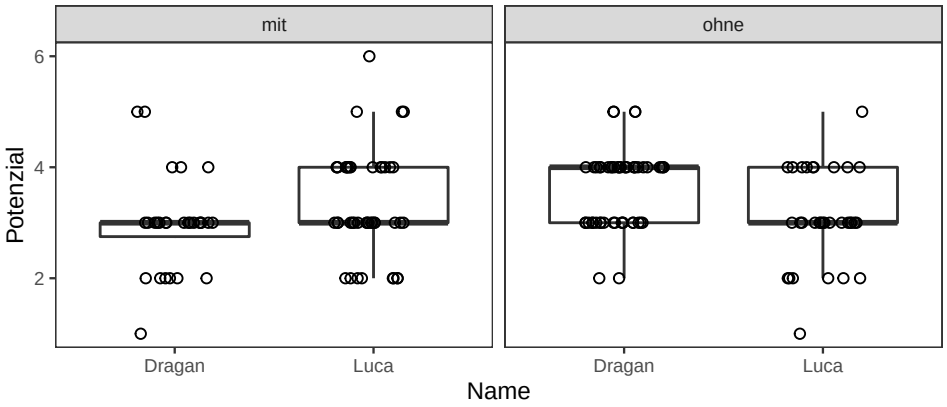


Abbildung 10.2: Boxplots der Bewertungen des akademischen Potenzials des Sprechers je nach Vorkommen von Codeswitches (mit vs. ohne) und seinem angeblichen Namen. Die Boxplots zeigen die Muster in den Daten hier nicht sehr deutlich, da die Daten nicht feinkörnig genug sind.

```
summary_berthele <- d %>%
  group_by(Name, CS) %>%
  summarise(n = n(),
            Mittel = mean(Potenzial),
            StdAb = sd(Potenzial))
summary_berthele
```

```
## # A tibble: 4 x 5
## # Groups:   Name [2]
##   Name   CS        n Mittel StdAb
##   <chr>  <chr> <int>  <dbl> <dbl>
## 1 Dragan mit      28   2.96 0.881
## 2 Dragan ohne     51   3.63 0.692
## 3 Luca   mit      44   3.36 0.942
## 4 Luca   ohne     32   3.09 0.856
```

10.1.3 Grafische Darstellung (II)

Abbildung 10.3 stellt die soeben berechneten Gruppenmittel in einem Liniendiagramm dar. Manche Forschende mögen es nicht, wenn für kategorische Prädiktoren Liniendiagramme gezeichnet werden, da diese implizieren würden, dass die Prädiktoren (hier etwa `Name`) kontinuierlich seien. Ich teile diese Meinung nicht: M.E. ist es klar, dass es zwischen `Dragan` und `Luca` keine Gradierung gibt. Die Zellenmittel selber werden ausserdem deutlich mit Punkten oder anderen Symbolen dargestellt, was dies nochmals betont. Die Linien, die diese Punkte verbinden, dienen nur der Deutlichkeit.

```
ggplot(summary_berthele,
       aes(x = Name,
```

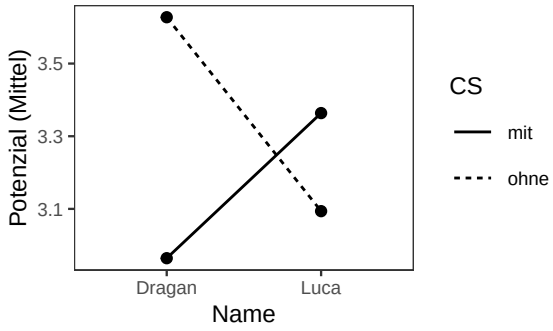


Abbildung 10.3: Liniendiagramm mit den Mitteln der Bewertungen des akademischen Potenzials des Sprechers je nach Vorkommen von Codeswitches (mit vs. ohne) und seinem angeblichen Namen. Die Muster sind hier deutlicher, aber dieser Grafik kann der Streuung der Daten nicht entnommen werden.

```
          y = Mittel,
          linetype = CS,  # unterschiedliche Linienarten je nach CS
          group = CS)) +  # Manchmal braucht ggplot ein bisschen Hilfe,
                          # um zu wissen, welche Punkte verknüpft werden
                          # sollten. 'group' verschafft diese Hilfe.
  geom_point() +
  geom_line() +
  ylab("Potenzial (Mittel)")
```

Es wäre jedoch nützlich, zusätzlich auch noch eine Idee über die Streuung der Daten innerhalb der Gruppen zu erhalten. Für Abbildung 10.4 wurden daher die Datenpunkte selber dargestellt und kombiniert mit den Mitteln aus dem Datensatz `summary_berthele`. Es ist also möglich, Angaben aus unterschiedlichen Datensätzen in einer Grafik zu kombinieren.

```
ggplot(d,
       aes(x = Name,
           y = Potenzial)) +
  geom_point(shape = 1, colour = "grey",
             position = position_jitter(width = 0.2, height = 0)) +
  geom_point(shape = 8, size = 3,
             data = summary_berthele,     # Daten aus anderem Datensatz
             aes(x = Name, y = Mittel)) +
  geom_line(data = summary_berthele,      # Daten aus anderem Datensatz
            aes(x = Name, y = Mittel, group = CS)) +
  facet_grid(. ~ CS)
```

Probieren Sie mehrere Grafiken aus. Für Gruppenvergleiche sind Boxplots oft eine gute Wahl, aber eben nicht immer. Wenn eine Grafik Ihrer Meinung nach Verbesserungspotenzial hat, dann sollten Sie nicht zögern, andere Varianten auszu-

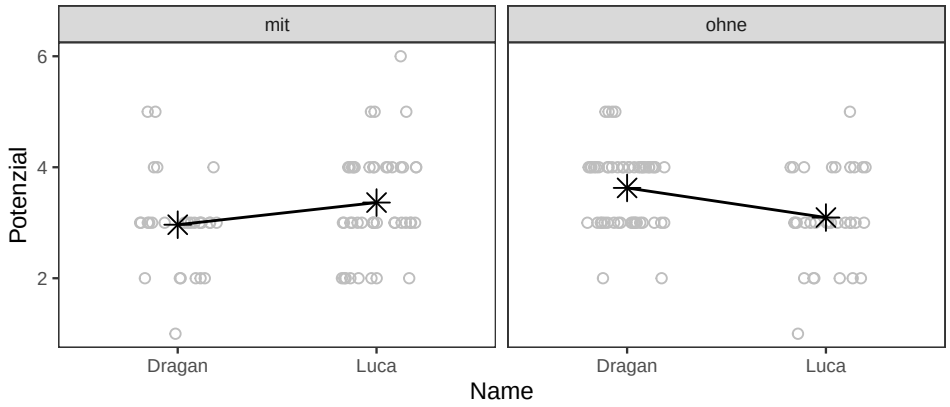


Abbildung 10.4: Bewertungen des akademischen Potenzials des Sprechers je nach Vorkommen von Codeswitches (mit vs. ohne) und seinem angeblichen Namen. Die Sternchen stellen die Gruppenmittel dar.

probieren.

Übrigens kostet es oft Zeit, eine gute grafische Darstellung zu konstruieren. In diesem Skript finden Sie das Endergebnis meiner Bemühungen, aber insbesondere für die letzte Grafik hat es eine Weile gedauert, bis ich herausgefunden habe, wie ich sie am besten zeichne. Die Entscheidung, die Datenpunkte grau zu färben, habe ich erst getroffen, nachdem ich feststellte, dass schwarze Datenpunkte die Zellenmittel nicht deutlich genug hervorhoben. Der Einstellung `size = 3` liegt eine ähnliche Beobachtung zu Grunde, usw.

10.1.4 Modellierung

Die Abbildungen zeigen, dass es innerhalb den Gruppen zwar ziemlich viel Variation gibt, aber dass der angebliche Name mit dem Vorkommen von Codeswitching interagieren dürfte: Liegen Codeswitches vor, dann wird das Potenzial von 'Luca' als besser eingestuft; liegen keine Codeswitches vor, dann scheinen die Lehrpersonen eher von der Leistung von Dragan als jene von Luca beeindruckt.

Mit der Modellierung möchten wir nicht nur die Fragen beantworten, welchen Zusammenhang der Name (Luca vs. Dragan) mit den Beurteilungen hat und welchen Zusammenhang Codeswitches mit ihnen haben, sondern auch inwiefern sich das Ausmass dieser Zusammenhänge je nach dem anderen Prädiktor unterscheidet. Um dies zu erreichen, brauchen wir vier β-Parameter:

1. β_0, der Schnittpunkt, erfasst den Baseline der Beurteilungen.
2. β_1 erfasst den Unterschied zwischen den Zellen je nach dem Namen (Dragan vs. Luca).
3. β_2 erfasst den Unterschied zwischen den Zellen je nach dem Vorkommen von Codeswitches.

4. β_3 passt β_1 und β_2 an: Wie viel grösser oder kleiner ist der Unterschied zwischen Dragan und Luca, wenn Codeswitches vorliegen als wenn keine vorliegen?

Mathematisch schaut die Modellgleichung so aus:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 (x_{1i} \times x_{2i}) + \varepsilon_i \tag{10.1}$$

- x_{1i} zeigt, ob der Versuchsperson erzählt wurde, dass der Bub Dragan (1) oder Luca (0) heisst.
- x_{2i} zeigt, ob der Versuchsperson der Text mit (1) oder ohne (0) Codeswitches vorgespielt wurde.[2]
- Entsprechend ist β_0 die Durchschnittsbeurteilung von Versuchspersonen, die angeblich Luca (0) ohne Codeswitches (0) gehört haben.

Der Faktor $(x_{1i} \times x_{2i})$ mag erstaunen: Tatsächlich wird die Anpassung (Interaktion) berechnet, indem eine neue Variable kreiert wird, die das Produkt der beiden Prädiktoren enthält. Für die vier Zellen im Design ergibt dies die folgenden Werte:

- Luca (0) ohne Codeswitches (0): $x_{1i} \times x_{2i} = 0 \times 0 = 0$
- Luca (0) mit Codeswitches (1): $x_{1i} \times x_{2i} = 0 \times 1 = 0$
- Dragan (1) ohne Codeswitches (0): $x_{1i} \times x_{2i} = 1 \times 0 = 0$
- Dragan (1) mit Codeswitches (1): $x_{1i} \times x_{2i} = 1 \times 1 = 1$

Mit diesen Befehlen werden die Dummy-Variablen kreiert und ihr Produkt berechnet.

```
d$Dragan <- ifelse(d$Name == "Dragan", 1, 0)
d$MitCS <- ifelse(d$CS == "mit", 1, 0)
d$DraganMitCS <- d$Dragan * d$MitCS

# Kontrolle:
d %>% sample_n(10)
```

```
## # A tibble: 10 x 6
##    CS    Name   Potenzial Dragan MitCS DraganMitCS
##    <chr> <chr>      <dbl>  <dbl> <dbl>       <dbl>
##  1 ohne  Luca           3      0     0           0
##  2 ohne  Dragan         4      1     0           0
##  3 mit   Dragan         4      1     1           1
##  4 ohne  Dragan         4      1     0           0
##  5 ohne  Dragan         4      1     0           0
##  6 ohne  Dragan         4      1     0           0
##  7 mit   Dragan         3      1     1           1
##  8 ohne  Luca           4      0     0           0
```

[2] Diese Notation geht hier von *treatment coding* aus. Andere Kodierungssysteme sind möglich, aber schwieriger zu erklären.

```
##  9 ohne  Dragan         4      1     0           0
## 10 ohne  Dragan         3      1     0           0
```

Diese Variablen können dem allgemeinen linearen Modell als Prädiktoren hinzugefügt werden:

```
potenzial.lm <- lm(Potenzial ~ Dragan + MitCS + DraganMitCS, data = d)
potenzial.lm
```

```
##
## Call:
## lm(formula = Potenzial ~ Dragan + MitCS + DraganMitCS, data = d)
##
## Coefficients:
## (Intercept)       Dragan        MitCS  DraganMitCS
##      3.0938       0.5337       0.2699      -0.9331
```

Mit diesen geschätzten Koeffizienten können die Gruppenmittel rekonstruiert werden.[3]

- Nicht Dragan (also Luca), ohne Codeswitching:

$$\hat{y} = 3.09 + (0.53 \times 0) + (0.27 \times 0) + (-0.93 \times 0) = 3.09$$

- Dragan, ohne Codeswitching:

$$\hat{y} = 3.09 + (0.53 \times 1) + (0.27 \times 0) + (-0.93 \times 0) = 3.63$$

- Nicht Dragan (also Luca), mit Codeswitching:

$$\hat{y} = 3.09 + (0.53 \times 0) + (0.27 \times 1) + (-0.93 \times 0) = 3.36$$

- Dragan, mit Codeswitching:

$$\hat{y} = 3.09 + (0.53 \times 1) + (0.27 \times 1) + (-0.93 \times 1) = 2.96$$

Die nicht-numerischen Variablen können auch direkt dem Modell hinzugefügt werden. Um die Interaktion zwischen ihnen zu modellieren, kann die `Name:CS`-Notation verwendet werden, aber dann müssen die beiden Variablen auch noch einzeln eingetragen werden. Eine alternative Schreibweise ist die `Name*CS`-Notation: Diese wird von R intern zu `Name + CS + Name:CS` konveriert.

[3] In den Summen wurden die Ründungsfehler korrigiert. Die Klammern sind überflüssig, aber verdeutlichen die Gleichungen.

```
potenzial.lm2 <- lm(Potenzial ~ Name + CS + Name:CS, data = d)
# Alternative Schreibweise
potenzial.lm2 <- lm(Potenzial ~ Name*CS, data = d)
potenzial.lm2
```

```
##
## Call:
## lm(formula = Potenzial ~ Name * CS, data = d)
##
## Coefficients:
##     (Intercept)         NameLuca           CSohne
##          2.9643           0.3994           0.6632
## NameLuca:CSohne
##         -0.9331
```

Denkaufgabe Warum haben sich die Parameterschätzungen geändert?

10.1.5 Unsicherheit einschätzen

Die Unsicherheit in den Parameterschätzungen kann wiederum mit `summary()` (für die Standardfehler) und mit `confint()` (für die Konfidenzintervalle) abgerufen werden. Die Konfidenzintervalle basieren wiederum auf t-Verteilungen und setzen also im Prinzip voraus, dass die Restfehler aus einer Normalverteilung stammen. In einem fakultativen Abschnitt werden wir diese Konfidenzintervalle nochmals berechnen mit einer Bootstrapmethode, die diese Voraussetzung nicht macht – die Ergebnisse unterscheiden sich dank der Datenmenge kaum.

```
summary(potenzial.lm)
```

```
##
## Call:
## lm(formula = Potenzial ~ Dragan + MitCS + DraganMitCS, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.09375 -0.62745  0.03571  0.37255  2.63636
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0938     0.1480  20.909  < 2e-16
## Dragan        0.5337     0.1888   2.827 0.005329
## MitCS         0.2699     0.1945   1.388 0.167221
## DraganMitCS  -0.9331     0.2767  -3.372 0.000948
##
```

```
## Residual standard error: 0.837 on 151 degrees of freedom
## Multiple R-squared:  0.08703,Adjusted R-squared:  0.06889
## F-statistic: 4.798 on 3 and 151 DF,  p-value: 0.003199
```

```
confint(potenzial.lm, level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept)  2.801406  3.3860940
## Dragan       0.160753  0.9066489
## MitCS       -0.114329  0.6541017
## DraganMitCS -1.479788 -0.3863151
```

Die Interpretation der Parameterschätzungen der Haupteffekte und ihre Unsicherheit ist etwas heikel: Wegen des *treatment codings* bezieht sich die Schätzung von 0.53 für `Dragan` eben *nur* auf Fälle, in denen kein Codeswitching vorliegt. Für die entsprechende Schätzung für Fälle mit Codeswitching muss man eben die Interaktion berücksichtigen: $0.53 - 0.93 = -0.40$.

Die Interpretation der Interaktion und ihre Unsicherheit ist einfacher: Der Effekt von Codeswitching ist wesentlich gravierender für Dragan als für Luca (nämlich 0.93 Punkte). (Oder, umgekehrt: Der Effekt der Absenz von Codeswitching ist 0.93 Punkte positiver für Dragan als für Luca.) Es gibt zwar ziemlich viel Unsicherheit über diesen Effekt, aber nach einer saloppen Interpretation des Konfidenzintervalls (siehe Abschnitt 6.6 auf Seite 78) scheint die Richtung dieses Effekts klar zu sein, liegt das Intervall doch komplett im negativen Bereich.

In Kapitel 8 haben wir die Ergebnisse der Modellierung grafisch dargestellt und die Unsicherheit mit einem Konfidenzband veranschaulicht. Dies können wir für diese Modellierung (und übrigens auch für die Modellierungen aus dem letzten Kapitel) machen. Das Kästchen zeigt, wie man diesen fakultativen Schritt ausführen kann. Überspringen Sie es bei der ersten Lektüre.

Modelle visualisieren. Die Idee ist, dass die vom Modell 'vorhergesagten' Werte ($\hat{y}$) und die Unsicherheit um diese Werte dargestellt werden. Es gibt zwar ein paar R-Funktionen, mit denen man dies automatisch machen kann (siehe Fox, 2003), aber da es ein Ziel dieses Kurses ist, Statistik zu demystifizieren, wird hier gezeigt, wie man solche Grafiken selber erzeugen kann.

Schritt 1: Wir kreieren einen neuen Datensatz, der die Kombinationen der Prädiktorwerte enthält, für die wir die $\hat{y}$-Werte zeichnen wollen. Die `expand.grid()`-Funktion ist hierfür besonders praktisch, denn sie kreiert einen Datensatz, in dem alle Kombinationen der Variablen vorkommen:

```
neue_daten <- expand.grid(Name = c("Dragan", "Luca"),
                          CS = c("mit", "ohne"))
neue_daten
```

```
##     Name   CS
## 1 Dragan  mit
## 2   Luca  mit
## 3 Dragan ohne
## 4   Luca ohne
```

Schritt 2: Im Modell arbeiten wir mit Dummy-Variablen. Diese müssen wir dem neuen Datensatz hinzufügen. Geben Sie dabei den Dummy-Variablen den gleichen Namen wie bei der Modellierung.

```
neue_daten$Dragan      <- ifelse(neue_daten$Name == "Dragan", 1, 0)
neue_daten$MitCS       <- ifelse(neue_daten$CS == "mit", 1, 0)
neue_daten$DraganMitCS <- neue_daten$Dragan * neue_daten$MitCS
neue_daten
```

```
##     Name   CS Dragan MitCS DraganMitCS
## 1 Dragan  mit      1     1           1
## 2   Luca  mit      0     1           0
## 3 Dragan ohne      1     0           0
## 4   Luca ohne      0     0           0
```

Schritt 3: Fügen Sie mit der `predict()`-Funktion die modellierten $\widehat{y}$-Werte hinzu. In diesem Fall sind diese $\widehat{y}$-Werte den Zellenmitteln gleich.

```
neue_daten$Mittel <- predict(potenzial.lm, newdata = neue_daten)
neue_daten
```

```
##     Name   CS Dragan MitCS DraganMitCS   Mittel
## 1 Dragan  mit      1     1           1 2.964286
## 2   Luca  mit      0     1           0 3.363636
## 3 Dragan ohne      1     0           0 3.627451
## 4   Luca ohne      0     0           0 3.093750
```

Schritt 4: Fügen Sie mithilfe der `predict()`-Funktion die Limiten des erwünschten Konfidenzintervalls hinzu. Dazu muss man spezifizieren, dass man sich ein Konfidenzintervall wünscht und das Konfidenzniveau spezifizieren. Wie Sie sehen, geniert diese Funktion eine Matrize, deren zweite Spalte (`lwr`) die untere Limite und deren dritte Spalte (`upr`) die obere Limite enthalten:

```
ki_limiten <- predict(potenzial.lm, newdata = neue_daten,
                      interval = "confidence", level = 0.95)
ki_limiten
```

```
##        fit      lwr      upr
## 1 2.964286 2.651757 3.276815
## 2 3.363636 3.114325 3.612948
## 3 3.627451 3.395880 3.859022
## 4 3.093750 2.801406 3.386094
```

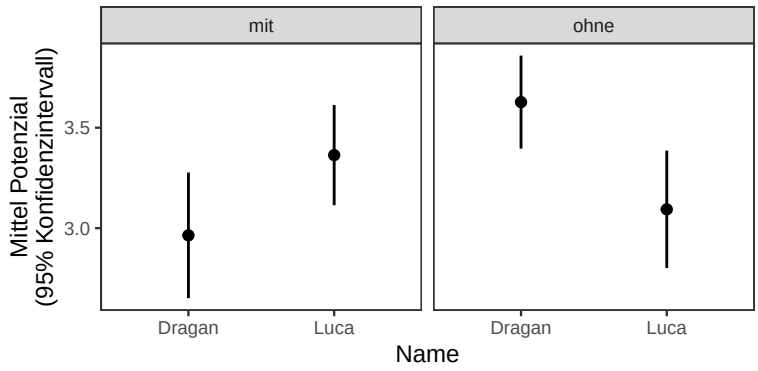
Diese Limiten fügen wir dann `dummy_data` hinzu:

```
neue_daten$KI_unten <- ki_limiten[, 2]
neue_daten$KI_oben  <- ki_limiten[, 3]
neue_daten
```

```
##     Name   CS Dragan MitCS DraganMitCS   Mittel KI_unten
## 1 Dragan  mit      1     1           1 2.964286 2.651757
## 2   Luca  mit      0     1           0 3.363636 3.114325
## 3 Dragan ohne      1     0           0 3.627451 3.395880
## 4   Luca ohne      0     0           0 3.093750 2.801406
##    KI_oben
## 1 3.276815
## 2 3.612948
## 3 3.859022
## 4 3.386094
```

Schritt 5: Stellen Sie die Werte in `Mittel` als Punkte dar und zeichnen Sie anhand der Konfidenzlimiten Intervalle um sie. Wenn Sie die vier #-Zeichen entfernen, werden auch noch die Rohdaten gezeichnet.

```
ggplot(neue_daten,
       aes(x = Name,
           y = Mittel)) +
  # geom_point(data = d,
  #            shape = 1, colour = "grey",
  #            aes(y = Potenzial),
  #            position = position_jitter(width = 0.2, height = 0)) +
  geom_point() +
  geom_linerange(aes(ymin = KI_unten,   # untere Limite
                     ymax = KI_oben)) + # obere Limite
  facet_grid(. ~ CS) +
  ylab("Mittel Potenzial\n(95% Konfidenzintervall)")
```



Wir hätten auch direkt auf der Basis der Zellenmittel und -standardabweichungen Konfidenzintervalle berechnen können, und zwar wie folgt:

```
summary_berthele <- summary_berthele %>%
  # Standardfehler berechnen
  mutate(SE = StdAb / sqrt(n)) %>%
  # untere Konfidenzlimite (t-basiert)
  mutate(KI_unten = Mittel + qt(0.025, df = n-1)*SE) %>%
  # obere Konfidenzlimite (t-basiert)
  mutate(KI_oben = Mittel + qt(0.975, df = n-1)*SE)
summary_berthele
```

```
## # A tibble: 4 x 8
## # Groups:   Name [2]
##   Name   CS        n Mittel StdAb     SE KI_unten KI_oben
##   <chr>  <chr> <int>  <dbl> <dbl>  <dbl>    <dbl>   <dbl>
## 1 Dragan mit      28   2.96 0.881 0.167      2.62    3.31
## 2 Dragan ohne     51   3.63 0.692 0.0969     3.43    3.82
## 3 Luca   mit      44   3.36 0.942 0.142      3.08    3.65
## 4 Luca   ohne     32   3.09 0.856 0.151      2.79    3.40
```

Der Grund, weshalb die Konfidenzintervalle in `neue_daten` und die in `summary_berthele` nicht identisch sind, ist, dass erstere von einem Modell abgeleitet wurden, das davon ausgeht, dass die Restfehler in jeder Zelle die gleiche Streuung hat. Letztere wurden direkt von den Standardabweichungen in den Zellen abgeleitet. Wenn der Restfehler in jeder Zelle tatsächlich (in der Population) die gleiche Streuung hat, liefert das Modell die besseren Konfidenzintervalle, da für die Schätzung der Streuung der Restfehler alle Datenpunkte zur Verfügung berücksichtigt wurden.

Weitere hoffentlich nützliche Links:

- Blogeintrag *Tutorial: Plotting regression models* (23.4.2017)
- Blogeintrag *Tutorial: Adding confidence bands to effect displays* (12.5.2017)
- `http://socviz.co/modeling.html#get-model-based-graphics-right`

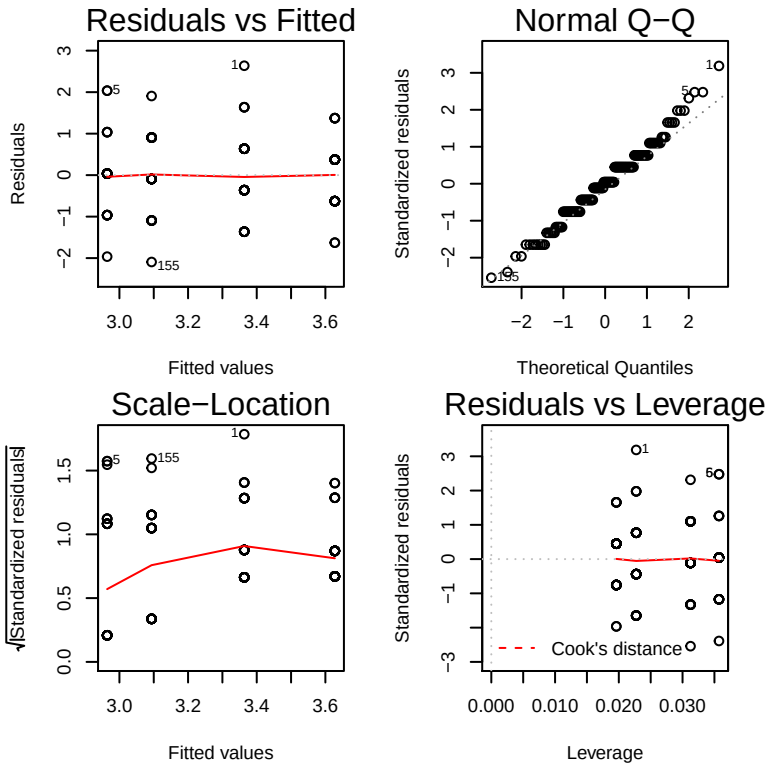


Abbildung 10.5: Diagnostische Plots für das `potenzial.lm`-Modell.

10.2 Annahmen überprüfen

Da auch dieses Modell ein Beispiel eines allgemeinen linearen Modells ist, hat es die gleichen Annahmen wie die Modelle aus den letzten Kapiteln: Unabhängigkeit, Normalität und Homoskedastizität. Die Beschreibung von Berthele (2012) lässt vermuten, dass Unabhängigkeit gegeben ist. Mit `plot(potenzial.lm)` können diagnostische Plots erzeugt werden, mit denen die Normalitäts- und Homoskedastizitätsannahmen eingeschätzt werden können, siehe Abbildung 10.5.

Die 'Treppchen' in der Grafik rechts oben zeigen, was wir schon wussten, nämlich dass die Potenzialbeurteilungen ziemlich grobkörnig sind. Normalverteilte Variablen sind im Prinzip unendlich feinkörnig und können ausserdem theoretisch von $-\infty$ bis ∞ reichen. Diese Daten sind aber theoretisch beschränkt zwischen 1 und 6. Erfahrungsgemäss weiss ich, dass solche Abweichungen von den theoretischen Annahmen wenig ausmachen, wenn die Datenmenge ausreichend ist, aber im Zweifelsfall können Sie die Konfidenzintervalle mit einer Methode überprüfen, die weniger Annahmen macht: dem Bootstrap. Das fakultative Kästchen zeigt wie.

Konfidenzintervalle mit dem Bootstrap überprüfen. Das Prinzip hinter dem Bootstrap wurde bereits mehrmals erklärt. Die letzten paar Male haben wir es erlaubt, dass Restfehler aus der einen Zelle/Kondition in der Bootstrapstichprobe in der an-

deren Zelle/Kondition auftauchen. Dies entsprach der Homoskedastizitätsannahme des allgemeinen linearen Modell. Aber wir können dies auch verhindern, indem wir die Bootstrapstichprobe so gestalten, dass Datenpunkte in einer bestimmten Zelle nur in der gleichen Zelle 'rezykliert' werden. Jede Zelle sollte gleich gross sein wie in der ursprunglichen Stichprobe.

```
# Zellengrössen in der eigentlichen Stichprobe
xtabs(~ Dragan + MitCS, data = d)
```

```
##       MitCS
## Dragan  0  1
##      0 32 44
##      1 51 28
```

```
# Innerhalb jede Zelle bootstrappen
bs_dat <- d %>%
  group_by(Dragan, MitCS) %>%
  sample_frac(replace = TRUE)

# Zellengrössen in der Bootstrapstichprobe
xtabs(~ Dragan + MitCS, data = bs_dat)
```

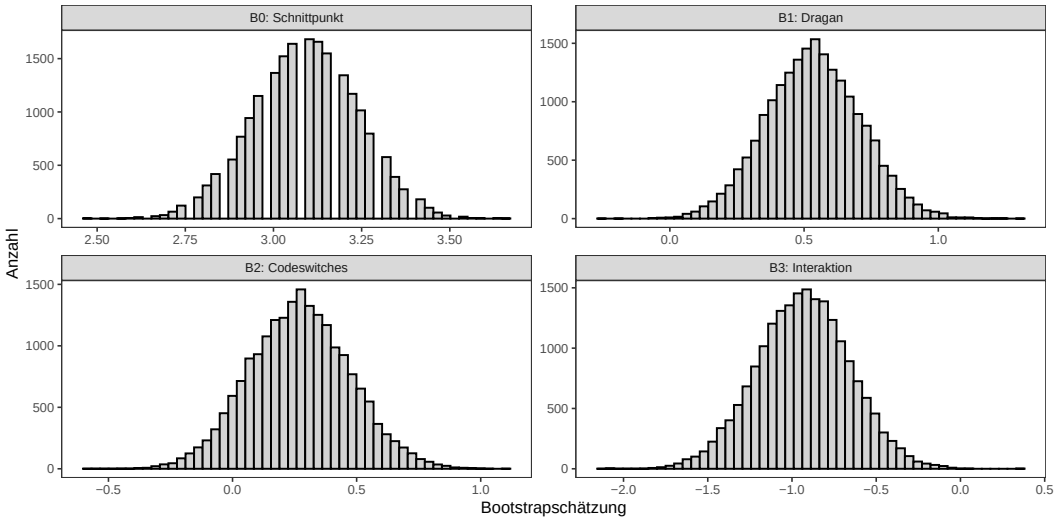
```
##       MitCS
## Dragan  0  1
##      0 32 44
##      1 51 28
```

Den Bootstrap können wir wie gehabt durchführen.

```
# Anzahl Bootstrapstichproben definieren
runs <- 20000

# Matrize für Parameterschätzungen
bs_beta <- matrix(nrow = runs, ncol = 4)

# For-loop für Bootstraps
for (i in 1:runs) {
  bs_dat <- d %>%
    group_by(Dragan, MitCS) %>%
    sample_frac(replace = TRUE)
  bs_mod <- lm(Potenzial ~ Dragan + MitCS + DraganMitCS,
               data = bs_dat)
  # Zeile der Matrize füllen.
  # 1. Spalte = intercept; 2. Spalte, Dragan;
  # 3. Spalte = MitCS; 4. Spalte: Interaktion
  bs_beta[i, ] <- coef(bs_mod)
}
```

Die Verteilungen der Bootstrapschätzungen können grafisch dargestellt werden; siehe Seite 113 für eine Erklärung der Befehle. Bemerken Sie, dass die Verteilung der Schätzungen für den Schnittpunkt Lücken aufzeigt: Der Schnittpunkt erfasst das Mittel der 32 Beurteilungen für Luca ohne Codeswitches. Da diese Beurteilungen aber grobkörnig sind, gibt es Werte, die gar kein Mittel dieser Beurteilungen sein können.

```
bs_beta <- data.frame(bs_beta)
# und Spalten umbenennen
colnames(bs_beta) <- c("B0: Schnittpunkt", "B1: Dragan",
                       "B2: Codeswitches", "B3: Interaktion")

# Grafik zeichnen
bs_beta %>%
  gather(key = Parameter, value = Estimate) %>%
  ggplot(aes(x = Estimate)) +
  geom_histogram(fill = "lightgrey", col = "black", bins = 50) +
  facet_wrap(~ Parameter, scales = "free", ncol = 2) +
  xlab("Bootstrapschätzung") +
  ylab("Anzahl")
```



Auch ohne die Annahme, dass die Restfehler aus einer Normalverteilung stammen und dass sie in jeder Zelle die gleiche Streuung haben, erhalten wir grundsätzlich die gleichen Konfidenzintervalle um die geschätzten Parameter.

```
apply(bs_beta, 2, quantile, probs = c(0.025, 0.975))
```

```
##       B0: Schnittpunkt B1: Dragan B2: Codeswitches
## 2.5%            2.8125  0.1936275       -0.1221591
## 97.5%           3.3750  0.8780637        0.6761364
##       B3: Interaktion
## 2.5%       -1.4777997
## 97.5%      -0.3940534
```

10.3 Interaktionen zwischen einem kontinuierlichen und einem binären Prädiktor

Manchmal stösst man auf Studien, in denen untersucht wird, ob der Zusammenhang eines kontinuierlichen Prädiktors mit dem *outcome* von Gruppe zu Gruppe variiert. Auch diese Art Fragestellung betrifft die Interaktion von zwei Variablen: einer kontinuierlichen und einer kategorischen.

Lieber ein einziges Modell als viele kleine. Um den Zusammenhang zwischen einem kontinuierlichen Prädiktor und dem *outcome* in unterschiedlichen Gruppen zu vergleichen, analysieren viele Forschende ihre Daten in separaten Modellen (einem Modell pro Gruppe). Wenn wir später über Signifikanztests sprechen, wird klar werden, wieso dies in der Regel eine schlechte Idee ist (siehe auch Gelman & Stern, 2006; Nieuwenhuis et al., 2011). Es ist besser die unterschiedlichen Prädiktoren und ihre Interaktionen in *einem* Modell zu analysieren, denn so erhält man Parameterschätzungen für die Interaktionen *und* Indizien über die Unsicherheit

dieser Schätzung. Wenn man die Gruppen in separaten Modellen analysiert, erhält man keine solchen Unsicherheitsmasse, und entsprechend wird die Unsicherheit über die Interaktionen in solchen Fällen fast ausnahmslos unterschätzt.

(Dieses Problem stelle ich in gefühlt einer von drei Studien, die ich lese, fest.)

Leider habe ich keinen Zugriff auf Datensätze, die eine solche Forschungsfrage behandeln *und* sinnvoll mit dem allgemeinen linearen Modell analysiert werden können. Um das Vorgehen zu illustrieren, versuche ich hier mit den Daten aus meiner Diss (siehe Übungen Kapitel 8) die etwas banale Frage, ob gute Englischkenntnisse beim Erkennen von gesprochenen Kognaten für Männer und Frauen unterschiedlich nützlich sind, zu beantworten.

```
# Daten einlesen und kombinieren; siehe Kapitel 8
cognates <- read_csv("Data/vanhove2014_cognates.csv")
background <- read_csv("Data/vanhove2014_background.csv")
d <- cognates %>%
  left_join(background)
head(d)
```

```
## # A tibble: 6 x 11
##   Subject Sex   CorrectWritten CorrectSpoken FirstBlock
##     <dbl> <chr>          <dbl>         <dbl> <chr>
## 1      64 fema~             25            23 Spoken
## 2     230 male              26            12 Spoken
## 3     527 male              15             9 Spoken
## 4     550 fema~             24            22 Spoken
## 5     552 fema~             18            17 Spoken
## 6     675 male              22            20 Spoken
## # ... with 6 more variables: Age <dbl>, NrLang <dbl>,
## #   DS.Span <dbl>, WST.Right <dbl>, Raven.Right <dbl>,
## #   English.Overall <dbl>
```

10.3.1 Grafische Darstellung

Abbildung 10.6 hebt hervor, dass man nie blind herumrechnen sollte.

```
ggplot(d,
       aes(x = English.Overall,
           y = CorrectSpoken)) +
  geom_point(shape = 1) +
  facet_grid(. ~ Sex)
```

Abbildung 10.7 zeigt, dass es sowohl für Männer als auch für Frauen einen positiven Zusammenhang zwischen der Kognatübersetzungsleistung und der Leistung beim Englischtest gibt.

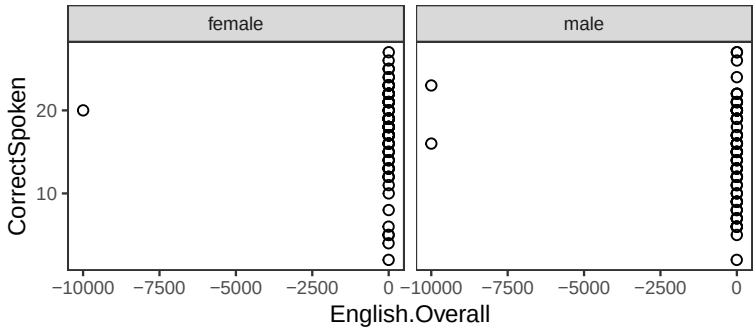


Abbildung 10.6: Ups.

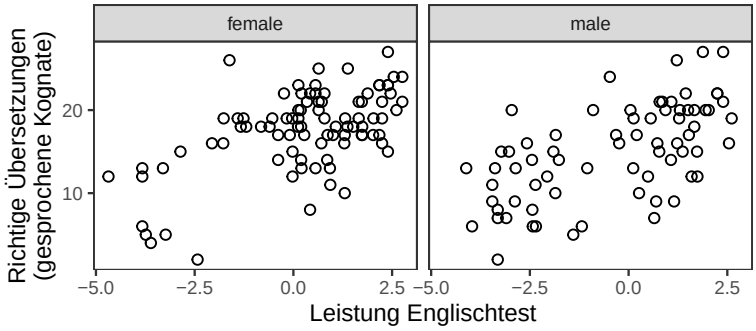


Abbildung 10.7: Für sowohl Männer als auch Frauen gibt es einen positiven Zusammenhang zwischen der Leistung beim Englischtest und bei der Kognatübersetzung.

```
# Fehlende Daten löschen:
d <- d %>%
  filter(English.Overall != -9999)

# Grafik zeichnen
ggplot(d,
       aes(x = English.Overall,
           y = CorrectSpoken)) +
  geom_point(shape = 1) +
  facet_grid(. ~ Sex) +
  xlab("Leistung Englischtest") +
  ylab("Richtige Übersetzungen\n(gesprochene Kognate)")
```

10.3.2 Modellierung

Die Modellierung verläuft analog zur vorherigen. Da es in der Stichprobe mehr Frauen als Männer gibt, kodiere ich Frauen als 0 und Männer als 1, aber das macht eigentlich nichts aus. Die Variable `English.Overall` ist bereits zentriert um 0, sodass wir dies nicht mehr machen müssen.

```
d$Mann <- ifelse(d$Sex == "male", 1, 0)
```

Auch in diesem Fall bezieht sich der geschätzte Parameter für die Interaktion auf das Produkt der beiden Prädiktoren. Aber R wird dies automatisch machen.

```
kognat.lm <- lm(CorrectSpoken ~ English.Overall * Mann,
                data = d)
kognat.lm
```

```
##
## Call:
## lm(formula = CorrectSpoken ~ English.Overall * Mann, data = d)
##
## Coefficients:
##          (Intercept)       English.Overall
##              17.1453                1.5920
##                 Mann  English.Overall:Mann
##              -1.4923                0.1281
```

Interpretation der Schätzungen:

- $\widehat{\beta_0}$: Die (modellierte) durchschnittliche Kognatübersetzungsleistung einer Frau mit einem Englischergebnis von 0 (hier: dem Stichprobenmittel). (Etwa 17 Punkte.)
- $\widehat{\beta_1}$: Wie viel besser schneidet (laut dem Modell) eine Frau im Schnitt ab, wenn sie ein Englischergebnis von 1 statt von 0 hat? (Etwa 1.6 Punkte besser.)
- $\widehat{\beta_2}$: Wie viel besser schneidet (laut dem Modell) ein Mann im Schnitt ab, wenn er ein Englischergebnis von 0 hat, verglichen mit einer Frau mit dem gleichen Englischergebnis? (Etwa 1.5 Punkte schlechter.)
- $\widehat{\beta_3}$: Wie viel 'nützlicher' ist ein Englischergebnis von 1 als eins von 0 für einen Mann als für eine Frau? (Etwa 0.1 Punkt beim Kognatsübersetzungstest nützlicher.)

10.3.3 Unsicherheit einschätzen

```
summary(kognat.lm)
```

```
##
## Call:
## lm(formula = CorrectSpoken ~ English.Overall * Mann, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2831  -2.8664   0.8951   2.6657  11.4299
```

```
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            17.1453     0.4581  37.429  < 2e-16
## English.Overall         1.5920     0.2540   6.268 3.42e-09
## Mann                   -1.4923     0.6888  -2.167   0.0318
## English.Overall:Mann    0.1281     0.3571   0.359   0.7204
##
## Residual standard error: 4.265 on 156 degrees of freedom
## Multiple R-squared:  0.3922,Adjusted R-squared:  0.3806
## F-statistic: 33.56 on 3 and 156 DF,  p-value: < 2.2e-16
```

```
confint(kognat.lm, level = 0.9)
```

```
##                              5 %       95 %
## (Intercept)           16.3873697 17.9033078
## English.Overall        1.1717316  2.0122668
## Mann                  -2.6319403 -0.3525918
## English.Overall:Mann -0.4628086  0.7189338
```

Hinsichtlich unserer banalen Frage zeigt das Konfidenzintervall für die Interaktion, dass Englischkenntnisse in dieser Stichprobe zwar nützlicher für Männer scheinen als für Frauen, aber die Unsicherheit ist so gross, dass das Muster auch mit einem negativen oder ungefähren Nullergebnis kompatibel ist.

10.3.4 Annahmen überprüfen

Die Annahmen werden auf die gleiche Art und Weise überprüft als in den letzten Abschnitten und Kapiteln. Das Vorgehen wird hier nicht gezeigt, da die Frage derartig banal ist und Vanhove (2014) die Daten ohnehin in einem angemesseren Modell analysiert hat (im Hinblick auf leicht weniger banale Fragen).

10.3.5 Modell visualisieren

Um das Modell zu visualisieren, kann die Technik, die im fakultativen Kästchen oben erklärt wurde, verwendet werden. Diese wird hier im Telegrammstil wiederholt. Das Ergebnis ist Abbildung 10.8.

```
# Datensatz mit Prädiktorkombinationen generieren.
# Für English.Overall nehme ich einfach die einzelnen
# Werte, die in der Stichprobe beobachtet wurden (unique()).
neue_daten <- expand.grid(Sex = c("male", "female"),
                          English.Overall = unique(d$English.Overall))
# Der Datensatz wird nicht angezeigt, um Papier zu sparen.
```

```
# neue_daten
```

```
# Dummy-Variable kodieren
neue_daten$Mann <- ifelse(neue_daten$Sex == "male", 1, 0)
```

```
# y-hat-Werte hinzufügen
neue_daten$yhat <- predict(kognat.lm, newdata = neue_daten)
```

```
# Konfidenzlimiten hinzufügen: hier sowohl 80% als auch 95%
limiten80 <- predict(kognat.lm, newdata = neue_daten,
                     interval = "confidence", level = 0.80)
limiten95 <- predict(kognat.lm, newdata = neue_daten,
                     interval = "confidence", level = 0.95)
neue_daten$ki_unten80 <- limiten80[, 2]
neue_daten$ki_oben80  <- limiten80[, 3]
neue_daten$ki_unten95 <- limiten95[, 2]
neue_daten$ki_oben95  <- limiten95[, 3]
# Ergebnis inspizieren (nicht gezeigt)
# head(neue_daten)
```

```
# Grafik zeichnen
ggplot(neue_daten,
       aes(x = English.Overall,
           y = yhat)) +
  geom_ribbon(aes(ymin = ki_unten95,
                  ymax = ki_oben95),
              fill = "lightgrey") +
  geom_ribbon(aes(ymin = ki_unten80,
                  ymax = ki_oben80),
              fill = "darkgrey") +
  geom_line() +
  ## Eventuell Datenpunkte hinzufügen
  # geom_point(data = d,
  #            shape = 1,
  #            aes(x = English.Overall,
  #                y = CorrectSpoken)) +
  facet_grid(. ~ Sex) +
  xlab("Ergebnis Englischtest") +
  ylab("Durchschnittliche Leistung\nKognatübersetzen")
```

10.4 Komplexere Interaktionen

Im Prinzip ist es auch möglich, Interaktionen zwischen zwei kontinuierlichen Prädiktoren dem Modell hinzuzufügen. Diese Möglichkeit wird in diesem Skript nicht bespro-

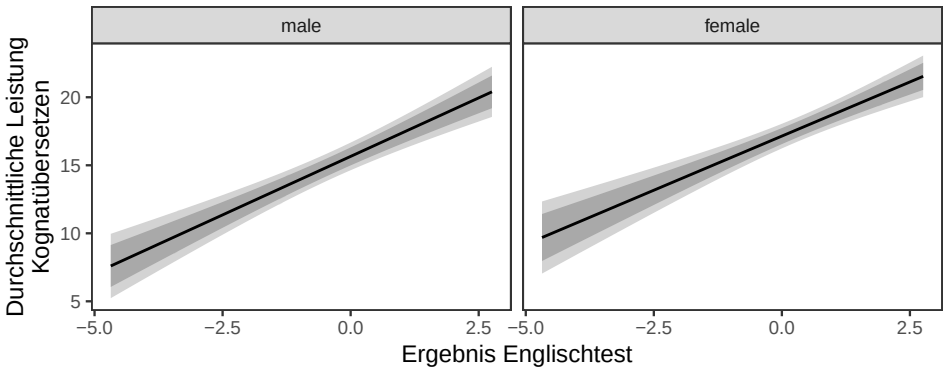


Abbildung 10.8: Visualisierung des `kognat.lm`-Modells mit 80%- und 95%-Konfidenzbändern.

chen; siehe aber meinen Blogeintrag *Interactions between continuous variables* (26.6.2017). Für ein ausführlicheres Beispiel, siehe Vanhove & Berthele (2017).

Auch Interaktionen zwischen drei oder mehr Prädiktoren können modelliert werden. Solche Fälle werden hier nicht behandelt, da ich erstens keinen Zugriff auf einen geeigneten Datensatz habe und da solche dreifache, vierfache usw. Interaktionen oft schwierig zu erklären sind, wenn man sich nicht bereits mit der Forschungsliteratur auskennt. Für ein Beispiel einer dreifachen Interaktion, siehe Bialystok et al. (2004) (Interaktion zwischen Alter (jung–alt), Kongruenz (kongruent–inkongruent) und Sprachgruppe (monolingual–bilingual) auf Reaktionsgeschwindigkeit).[4] Bialystok et al. analysierten ihre Daten übrigens nicht mit dem allgemeinen linearen Modell, da es Abhängigkeiten in den Daten gab: Die Versuchspersonen wurden sowohl in der kongruenten als auch in der inkongruenten Kondition getestet.

10.5 Zur Interpretation von Interaktionen und Haupteffekten

Es kann schwierig sein, Haupteffekte zu interpretieren, wenn eine Interaktion vorliegt; am besten basiert man sich hierbei auf einer Grafik. Bei etwa dem Datenmuster in Abbildung 10.9 wäre es vorschnell zu sagen, dass der *outcome* im Schnitt höher ist bei A als bei B (Haupteffekt von A vs. B) oder dass er höher ist bei Y als bei X (Haupteffekt von X vs. Y), auch wenn das Modell beide Haupteffekte belegen würde: Der Punkt ist ja dass es nur einen Unterschied zu geben scheint, wenn A und Y *gleichzeitig* vorkommen (Interaktion zwischen AB und XY).

Zur Interpretation von *non-cross-over interactions*, siehe Wagenmakers et al. (2012a). Zusammengefasst: Eine Interaktion in der gemessenen Variablen (z.B. Reaktionsge-

[4] Wie Abelson (1995) erklärt, können Interaktionen oft weggerechnet werden. Zum Beispiel hätten Bialystok et al. (2004) für jede Versuchsperson den Unterschied zwischen den kongruenten und inkongruenten Reaktionszeiten berechnen können und dann die zweifachen Interaktion zwischen Alter und Sprachgruppe untersuchen können. Das Ergebnis wäre genau gleich gewesen, aber die Analyse einfacher und wohl verständlicher.

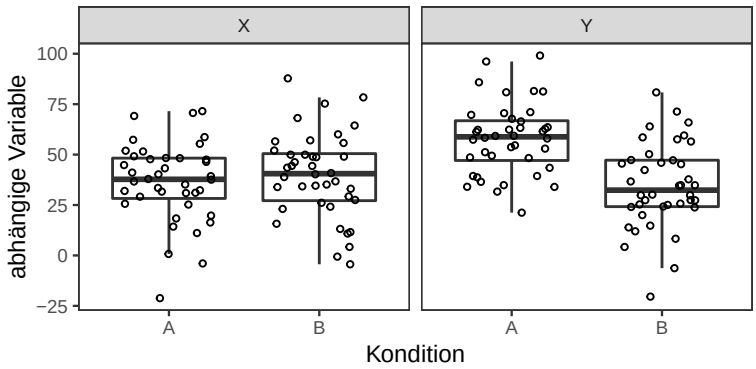


Abbildung 10.9: In diesem simulierten Datensatz gibt es zwar Haupteffekte von A vs. B und von X vs. Y, aber eigentlich schneidet nur die Gruppe mit A *und* Y wesentlich besser ab.

schwindigkeit) muss nicht zwingend darauf hindeuten, dass eine Interaktion im hinterliegenden Konstrukt (z.B. kognitiver Kontrolle) vorliegt.

Mangels geeigneter Datensätze gibt es in diesem Kapitel keine praktischen Aufgaben mit neuen Datensätzen.

Kapitel 11

Mehrere Prädiktoren in einem Modell

Wie wir es bereits in den letzten zwei Kapiteln gesehen haben, kann das allgemeine lineare Modell gut mit mehreren Prädiktoren umgehen. Von der Berechnung her ändert sich hierbei eigentlich nichts. In diesem Kapitel schauen wir uns zwei typische Situationen an, in denen es sich lohnt, mehrere Prädiktoren in einem Modell zu berücksichtigen: (a) wenn man den Einfluss mehrerer Prädiktoren gleichzeitig untersuchen möchte und dabei auch Zusammenhänge zwischen den Prädiktoren berücksichtigen möchte und (b) wenn man die Unsicherheit in der Parameterschätzung eines Prädiktors verringern möchte, indem man die Fehlervarianz anhand weiterer Prädiktoren reduziert. Beide Situationen überlappen in der Praxis übrigens öfters.[1]

11.1 Den Einfluss mehrerer Prädiktoren gleichzeitig untersuchen

Als Beispiel für diesen Verwendungszweck schauen wir uns wieder die Kognatsübersetzungsdaten von Vanhove (2014) an. Wir lesen zunächst sowohl die Übersetzungs- als auch die Hintergrundsdaten ein. Diese werden dann mit `left_join()` miteinander kombiniert. In der `English.Overall`-Spalte (Ergebnis bei einem Englischtest) steht der Wert −9999 für fehlende Angaben; diese werden hier gelöscht. Eine Versuchsperson hat in der `WST.Right`-Spalte (Ergebnis bei einem deutschen Wortschatztest) einen Wert von 0; auch dieser steht eigentlich für eine fehlende Angabe und wird daher gelöscht.[2]

```
cognates <- read_csv("Data/vanhove2014_cognates.csv")
background <- read_csv("Data/vanhove2014_background.csv")
```

[1] Für Verwendungszweck (a) wird oft den Begriff *mehrfache Regression* verwendet; für Zweck (b) ANCOVA (*analysis of covariance; Kovarianzanalyse*). Die verwendeten Modelle sind aber gleich.

[2] Es bestehen raffiniertere Methoden, um mit fehlenden Angaben umzugehen. Eine Einführung findet sich bei Graham (2009); siehe auch Honaker et al. (2011).

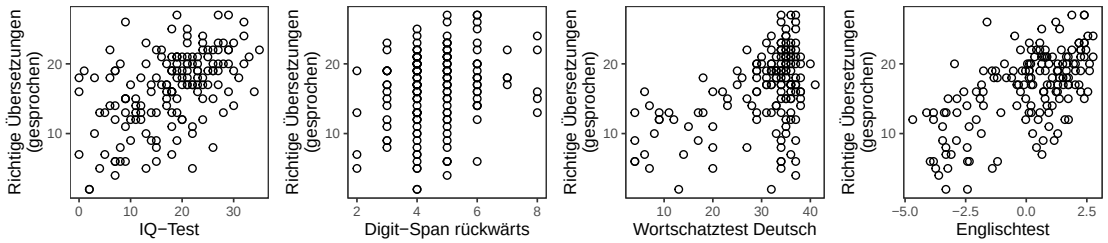


Abbildung 11.1: Zusammenhänge zwischen Anzahl richtiger Übersetzungsversuche und vier Prädiktoren.

```
d <- cognates %>%
  left_join(background) %>%
  filter(English.Overall != -9999) %>%
  filter(WST.Right != 0)
```

Abbildung 11.1 zeigt vier Streudiagramme für den Zusammenhang zwischen der Anzahl richtiger Übersetzungen in der gesprochenen Modalität einerseits und den Prädiktoren `Raven.Right` (Ergebnis bei einem Intelligenztest), `DS.Span` (Ergebnis bei einem Arbeitgedächtnistest), `WST.Right` und `English.Overall` andererseits. Die Befehle, um diese Streudiagramme zu zeichnen, werden hier nicht gezeigt.

Man könnte jetzt zwar vier unterschiedliche Modelle (eins für jeden Prädiktor) kreieren und sich diese getrennt anschauen. Aber das Problem mit diesem Ansatz ist, dass diese vier Modelle zum Teil die gleiche Information ausdrücken würden: Was Abbildung 11.1 nämlich nicht zeigt, ist, dass die Prädiktoren miteinander korrelieren. Dies wird deutlich aus der **Steudiagrammmatrize** in Abbildung 11.2, die mit dem `pairscor.fnc()`-Befehl aus dem `languageR`-Package gezeichnet wurde. Dieses Package müssen Sie ggf. noch installieren.

```
d %>%
  select(CorrectSpoken, WST.Right, English.Overall,
         Raven.Right, DS.Span) %>%
  languageR::pairscor.fnc()
```

Aus der Streudiagrammmatrize geht hervor, dass etwa die Prädiktoren `English.Overall` und `Raven.Right` miteinander korreliert sind ($r = 0.42$). Wenn wir also in dem einen Modell einen Zusammenhang zwischen `English.Overall` und dem outcome feststellen, dann ist bereits zu erwarten, dass wir auch im anderen Modell einen Zusammenhang zwischen `Raven.Right` und dem outcome feststellen werden, denn wer beim einen Prädiktor gut abschneidet, tendiert dazu, auch beim anderen gut abzuschneiden. Um besser einschätzen zu können, welchen Zusammenhang jeder Prädiktor unabhängig von den anderen Prädiktoren auf den outcome hat, kann man die unterschiedlichen Prädiktoren alle in einem Modell berücksichtigten.

Bevor ich die vier Prädiktoren in ein Modell giesse, **zentriere** ich diese – d.h., ich ziehe das jeweilige Stichprobenmittel von ihren Werten ab. Dadurch wird das `(Intercept)`

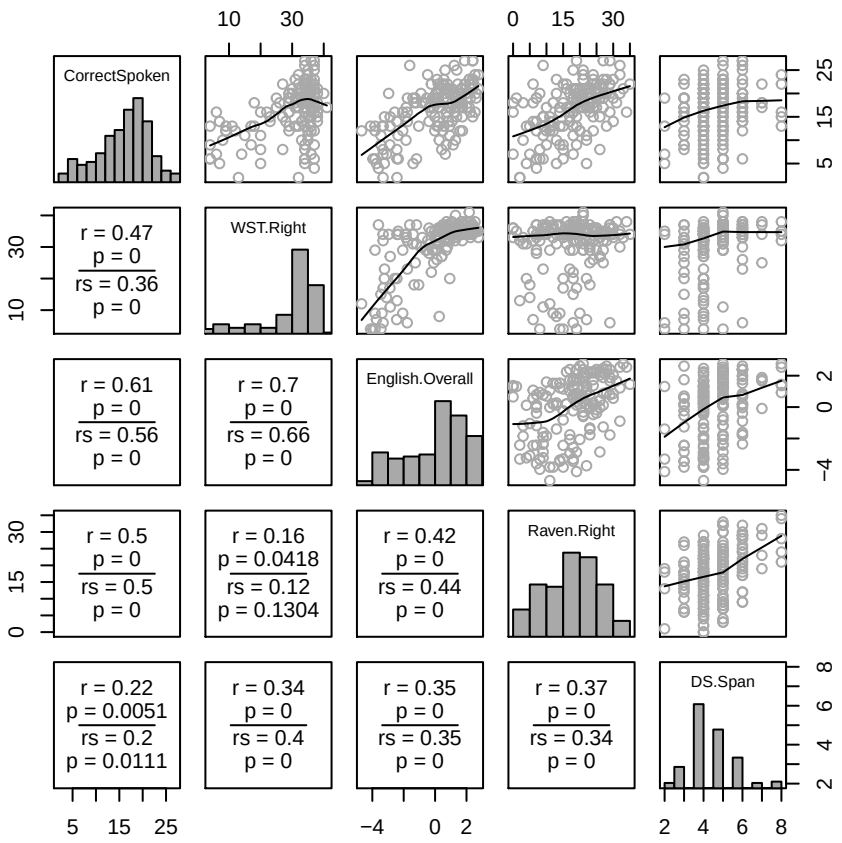


Abbildung 11.2: Eine Streudiagrammmatrize. Das Streudiagramm in der 3. Zeile, 5. Spalte, etwa, zeigt den Zusammenhang zwischen den Variablen `DS.Span` und `English.Overall`; die Zahlen in der 5. Zeile, 3. Spalte sind Pearsons Korrelationskoeffizient (r) und Spearmans Rangkorrelationskoeffizient (rs) für diesen Zusammenhang.

einfacher interpretierbar als die modellierte durchschnittliche Übersetzungsleistung von (imaginären) Teilnehmenden, die bei allen berücksichtigen Prädiktoren beim Stichprobenmittel abschneiden. Dieser Wert entspricht ebenfalls dem Stichprobenmittel des *outcomes*.

```
# Prädiktoren zentrieren
d$c.WST <- d$WST.Right - mean(d$WST.Right)
d$c.Englisch <- d$English.Overall - mean(d$English.Overall)
d$c.Raven <- d$Raven.Right - mean(d$Raven.Right)
d$c.DS <- d$DS.Span - mean(d$DS.Span)
```

Die Prädiktoren werden dem Modell hinzugefügt und die geschätzten Parameter und ihre Standardfehler können wie gehabt abgefragt werden:

```
alles.lm <- lm(CorrectSpoken ~ c.WST + c.Englisch + c.Raven + c.DS,
               data = d)
summary(alles.lm)
```

```
##
## Call:
## lm(formula = CorrectSpoken ~ c.WST + c.Englisch + c.Raven + c.DS,
##     data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7373  -2.4169   0.8803   2.4921  13.1768
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.47170    0.31817  51.770  < 2e-16
## c.WST        0.11615    0.05277   2.201 0.029218
## c.Englisch   1.03547    0.26144   3.961 0.000114
## c.Raven      0.23723    0.04645   5.107 9.54e-07
## c.DS        -0.46507    0.30353  -1.532 0.127524
##
## Residual standard error: 4.012 on 154 degrees of freedom
## Multiple R-squared:  0.4689,Adjusted R-squared:  0.4551
## F-statistic:    34 on 4 and 154 DF,  p-value: < 2.2e-16
```

Auch für dieses Modell können die Konfidenzintervalle um die geschätzten Parameter mit `confint()` berechnet werden:

```
# 80%-Konfidenzintervall, nur um zu zeigen,
#  dass die traditionellere Wahl für 95% rein arbiträr ist:
confint(alles.lm, level = 0.80)
```

```
##                    10 %       90 %
```

```
## (Intercept) 16.06218580 16.88121042
## c.WST        0.04822997  0.18406084
## c.Englisch   0.69897447  1.37196866
## c.Raven      0.17744509  0.29701175
## c.DS        -0.85573404 -0.07440382
```

11.1.1 Die Parameterschätzungen interpretieren

Die geschätzten Parameter des obigen Modells können eigentlich genauso wie die geschätzten Parameter der Modelle aus den letzten Kapiteln interpretiert werden:

1. Das geschätzte `(Intercept)` ist die vom Modell geschätzte durchschnittliche Anzahl richtiger Antworten in der gesprochenen Modalität für Teilnehmende, deren Werte bei den vier Prädiktoren 0 sind. Da wir die vier Prädiktoren zentriert haben, ist dies also die geschätzte Durchschnittsleistung von Teilnehmenden, die bei allen vier Prädiktoren beim jeweiligen Stichprobenmittel abschneiden.

2. Die Schätzung von etwa 0.12 für den Prädiktor `c.WST` zeigt, dass das Modell schätzt, dass Teilnehmende mit einem `c.WST`-Wert von 1 beim outcome im Schnitt 0.12 Punkte besser abschneiden als Teilnehmende mit einem `c.WST`-Wert von 0, *wenn ihre Werte bei den anderen Prädiktoren gleich sind*. Das lässt sich auch aus der Regressionsgleichung ableiten. Versuchspersonen, deren Prädiktorwerte überall 0 wären, hätten laut dem Modell im Schnitt einen outcome-Wert von:

$$\widehat{y} = 16.47 + (0.12 \times 0) + (1.04 \times 0) + (0.24 \times 0) + (-0.47 \times 0) = 16.47$$

   Für Versuchspersonen mit einem `c.WST`-Wert von 1 aber sonst identischen Prädiktorwerten wäre das Durchschnittsergebnis aber:

$$\widehat{y} = 16.47 + (0.12 \times 1) + (1.04 \times 0) + (0.24 \times 0) + (-0.47 \times 0) = 16.59$$

   Beachten Sie, dass sich das Modell nicht darüber äussert, ob dieser positive Zusammenhang kausal ist oder anderen Faktoren zugeschrieben werden kann. Vielleicht schneiden Teilnehmende mit höheren `c.WST`-Werten besser ab, etwa weil ihr Bildungsstand höher ist. Man kann Modelle natürlich kausal interpretieren, aber dies ist dann eben Interpretationssache—*zeigen*, dass es einen kausalen Zusammenhang gibt, tut das Modell keineswegs.

3. Die Schätzung von etwa 1.04 für den Prädiktor `c.Englisch` zeigt ebenfalls, dass das Modell schätzt, dass Teilnehmende mit einem `c.Englisch`-Wert von 1 beim outcome im Schnitt 1.04 Punkte besser abschneiden als Teilnehmende mit einem `c.Englisch`-Wert von 0, *wenn ihre Werte bei den anderen Prädiktoren gleich sind*.

   Aber Vorsicht: Die Zahl 1.04 ist natürlich zwar grösser als die Zahl 0.12, aber dies heisst nicht unbedingt, dass `c.Englisch` einen stärkeren Zusammenhang

mit dem outcome aufzeigt als `c.WST`. Die `c.WST`-Variable reicht von −26.2 bis 10.8; die `c.Englisch`-Variable nur von −4.7 bis 2.7; die Prädiktoren sind also nicht vergleichbar, und folglich auch die Parameterschätzungen nicht. Siehe noch den Blogeintrag *Which predictor is most important? Predictive utility vs. construct importance* (15.2.2017).

4. Auch die Schätzung für `c.Raven` lässt sich so interpretieren. *Unter sonst gleichen Bedingungen* würden Teilnehmende mit einem `c.Raven`-Wert von 1 im Schnitt 0.24 Punkte besser abscheiden als Teilnehmende mit einem `c.Raven`-Wert von 0. Ob die Bedingungen sonst überhaupt gleich sein können, darüber macht das Modell aber keine Aussagen.
5. Die Schätzung für `c.DS` ist negativ (−0.47). Das Modell vermutet also, dass Teilnehmende *unter sonst gleichen Bedingungen* beim Übersetzungstest im Schnitt schlechter abschneiden würden, wenn sie höhere Werte beim `DS.Span`-Test haben. Dabei gab es, wie es Abbildungen 11.1 und 11.2 zeigen, einen positiven Zusammenhang zwischen den Ergebnissen bei diesen beiden Tests. Dieser positive Zusammenhang dürfte aber durch die positiven Korrelationen zwischen `DS.Span` und den anderen Prädiktoren bedingt sein: Wenn man diese im Modell berücksichtigt, bleibt kein positiver Zusammenhang zwischen dem outcome und `DS.Span` mehr übrig.

Aufgabe 1 Was wäre laut dem Modell das durchschnittliche Übersetzungsergebnis von Teilnehmenden mit einem `c.Raven`-Wert von 12, deren sonstige Prädiktorwerte dem jeweiligen Stichprobenmittel entsprechen?

Aufgabe 2 Was wäre laut dem Modell das durchschnittliche Übersetzungsergebnis von Teilnehmenden mit einem `c.Raven`-Wert von 12 und einem `c.Englisch`-Wert von −3, deren sonstige Prädiktorwerte dem jeweiligen Stichprobenmittel entsprechen?

Aufgabe 3 Was wäre laut dem Modell das durchschnittliche Übersetzungsergebnis von Teilnehmenden mit einem `Raven.Right`-Wert (!) von 12, deren sonstige Prädiktorwerte dem jeweiligen Stichprobenmittel entsprechen?

Kollinearität und ihre Konsequenzen. Eine wichtige Phrase in diesem Abschnitt ist *unter sonst gleichen Bedingungen*. Insbesondere wenn die Prädiktoren stark miteinander korrelieren (also stark 'kollinear' sind), ist es möglich, dass die Bedingungen sonst nie gleich sein können. Wenn die Prädiktoren also ziemlich stark miteinander korrelieren, wird es schwieriger die Ergebnisse des Modells für bare Münze zu nehmen.

Diese Situation kann auch vorkommen, wenn unterschiedliche Prädiktoren (zum Teil) voneinander abgeleitet worden sind. Zum Beispiel könnte man Textqualitäts-

beurteilungen anhand von Texteigenschaften modellieren. Ein Prädiktor könnte dann die Textlänge (Anzahl *tokens*) sein, ein anderer die Anzahl unterschiedlicher Wörter (*types*) im Text und noch ein anderer die Ratio der beiden (*type–token ratio*). Das Modell würde problemlos die Koeffizienten für alle Prädiktoren schätzen, aber es ist natürlich nicht möglich, dass ein Text sich von einem anderen Text in der Anzahl *tokens* unterscheidet, aber nicht in der Anzahl *types* und in der *type–token ratio*: Unterscheidet sich die Anzahl *tokens* und ist die Anzahl *types* gleich, dann ist die *type–token ratio* natürlich auch eine andere!

Neben Interpretationsschwierigkeiten führt starke Kollinearität auch dazu, dass die Parameterschätzungen sich von Stichprobe zu Stichprobe stärker unterscheiden als wenn es keine Kollinearität gegeben hätte. Dies zeigt sich in grösseren Standardfehlern der Parameterschätzungen der von Kollinearität betroffenen Prädiktoren. (Wenn zwei Prädiktoren stark miteinander korreliert sind, nicht aber mit einem dritten Prädiktor, werden nur die Standardfehler der ersten beiden Parameterschätzungen vergrössern, nicht der Standardfehler der dritten Parameterschätzung.)

Das Ausmass an Kollinearität können Sie zum Teil bereits anhand einer Streudiagrammmatrize einschätzen. Eine weitere Möglichkeit besteht darin, sog. *variance inflation factors* (VIF) zu berechnen. Diese drücken aus, wie viel variabler jede Parameterschätzung ist verglichen mit dem hypothetischen Fall, wo es keine Kollinearität gegeben hätte. Ein VIF von 1 drückt aus, dass die Schätzung nicht von Kollinearität betroffen ist; ein VIF von 10 drückt aus, dass die Parameterschätzung 10 Mal variabler ist, als wenn es keine Kollinearität gegeben hätte.

```
# ggf. müssen Sie das "car"-Package noch installieren:
car::vif(alles.lm)
```

```
##      c.WST c.Englisch    c.Raven       c.DS
##   2.173397   2.456164   1.401556   1.279809
```

Manche sprechen von starker Kollinearität ab VIF-Werten von 10, andere ab VIF-Werten von 5; wie immer gibt es keinen nicht-arbiträren Schwellenwert. Mit einem VIF-Höchstwert von 2.46 sind wir hier aber komplett im grünen Bereich. Und auch wenn die VIF-Werte hoch gewesen wären, muss dies nicht zwangsläufig auf ein zu behebendes Problem hinweisen:

> Values of the VIF of 10, 20, 40, or even higher do not, by themselves, discount the results of regression analyses, call for the elimination of one or more independent variables from the analysis, suggest the use of ridge regression, or require combining of independent variable[s] into a single index. (O'Brien, 2007, Abstract)

Einleuchtend finde ich die folgende Behandlung:

> [C]ollinearity is not per se a statistical problem, but rather a problem

> for statistical analysis stemming from the state of the real world (i.e. in the real world, it is rare for variables to be orthogonal [= unkorreliert]). *Since the problem, then, is not statistical in origin, it does not have a strictly statistical solution.* Collinearity is at base a problem about information. If two factors are highly correlated, researchers do not have ready access to much information about conditions of the dependent variables when only one of the factors actually varies and the other does not. If we are faced with this problem, there are really only three fundamental solutions: (1) find or create (e.g. via an experimental design) circumstances where there is reduced collinearity; (2) get more data (i.e. increase the N size), so that there is a greater quantity of information about rare instances where there is some divergence between the collinear variables; or (3) add a variable or variables to the model, with some degree of independence from the other independent variables, that explain(s) more of the variance of *Y*, so that there is more information about that which is being modeled.
>
> (...)
>
> Statistical "solutions," such as residualization that are often used to address collinearity problems do not, in fact, address the fundamental issue, a limited quantity of information, but rather *serve to obfuscate it*. It is perhaps obvious to point out, but nonetheless important in light of the widespread confusion on the matter, that no statistical procedure can actually produce more information than exists in the data. (York, 2012, S. 1384, meine Hervorhebung)

Siehe auch Wurm & Fisicaro (2014).

11.1.2 Was heissen *multiple R-squared* und *adjusted R-squared*?

Eine der Zeilen im `summary()`-Output, die wir bisher ignoriert haben, ist die Zeile mit den Infos `Multiple R-squared` (R^2) und `Adjusted R-squared` (R^2_{adj}).

Die erste Zahl, R^2, drückt aus, welchen Anteil der Varianz im outcome *in dieser Stichprobe* von der Regressionsgleichung erfasst wird.[3] Die Varianz des outcomes beträgt etwa 29.5:

```
var(d$CorrectSpoken)

## [1] 29.54192
```

Die Varianz der Modellresiduen beträgt noch etwa 15.7:

[3] Oft spricht man dabei dann von 'erklärter Varianz', aber das Modell erklärt ja eigentlich nichts (das ist den Forschenden überlassen), sondern beschreibt nur.

```
var(resid(alles.lm))
```

```
## [1] 15.68875
```

Von der Varianz in der outcome-Variablen bleibt also noch $\frac{15.7}{29.5} = 53\%$ übrig, wenn man die linearen Zusammenhänge mit den vier Prädiktoren berücksichtigt. Mit anderen Worten beschreibt das Modell 47% der Varianz im outcome:[4]

```
1 - var(resid(alles.lm)) / var(d$CorrectSpoken)
```

```
## [1] 0.4689326
```

Ein Problem mit R^2 ist, dass es nur grösser werden kann, wenn man dem Modell mehr und mehr Prädiktoren hinzufügt. Dies auch dann wenn diese Prädiktoren eigentlich nicht mit dem outcome zusammenhängen: Rein durch Zufall wird der geschätzte Regressionskoeffizient für den Zusammenhang zwischen einem zusätzlichen, irrelevanten Prädiktor und dem outcome in der Stichprobe nie ganz genau 0 sein. In der *Stichprobe* beschreibt ein irrelevanter Prädiktor also immer noch ein bisschen Varianz im outcome, auch wenn er dies in der *Population* nicht tut. Um dieses Problem vorzubeugen, wird der R^2-Wert manchmal nach unten korrigiert, und zwar mit dieser Formel:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}, \tag{11.1}$$

wo n die Anzahl Datenpunkte ist und p die Anzahl Prädiktoren. In unserem Fall ergibt dies also:

```
1 - (1 - 0.4689)*(159 - 1)/(159 - 4 - 1)
```

```
## [1] 0.4551052
```

Selber bin ich kein grosser Fan von R^2 oder R^2_{adj}; siehe *Why reported* R^2 *values are often too high* (22.4.2016).

Was schätzt R^2 genau? Viele Forschende scheinen zu denken, dass R^2 ihnen sagt, wie viel Varianz das geschätzte Regressionsmodell vermutlich in einer neuen Stichprobe erfassen wird. Das stimmt aber nicht: R^2_{adj} schätzt, wie viel Varianz ein *ähnliches* Modell *aber mit neuen Parameterschätzungen* in einer neuen Stichprobe erfassen wird.

Die Vorhersagekraft eines Modells mit Kreuzvalidierung schätzen. R^2 sagt uns nicht, wie gut das geschätzte Modell neue Daten vorhersagen kann, ohne das die Parameterschätzungen geändert werden. Aber es gibt ein paar Möglichkeiten, um die Vorhersagekraft eines Modells einzuschätzen; siehe Kuhn & Johnson (2013, Kapitel 4) und Yarkoni & Westfall (2017). Hier wird eine solche Methode besprochen:

[4] Es gibt noch andere Berechnungsmethoden, aber diese ergeben beim allgemeinen linearen Modell alle die gleiche Lösung; siehe Kvålseth (1985).

leave-on-out cross-validation (LOOCV).

In Kreuzvalidierung allgemein lässt man einen Teil des Datensatzes ausser Betracht und schätzt man das Modell erneut. Dann verwendet man dieses neue Modell, um die ausser Betracht gelassenen Daten vorherzusagen. Dann schaut man, wie stark die Modellvorhersagen von den tatsächlichen Beobachtungen abweichen. Dieses Prozedere wiederholt man mehrmals. Die durchschnittliche Vorhersagekraft all dieser Modelle gibt einem dann ein Indiz darüber, wie gut das Modell, das auf der Basis aller Daten geschätzt wurde, neue Daten vorhersagen wird. *leave-on-out cross-validation* (LOOCV) ist eine Form von Kreuzvalidierung, in der man jeweils einen Datenpunkt ausser Betracht lässt.

Der Code unten bewirkt Folgendes:

1. Das Modell `alles.lm` wird mehrmals erneut berechnet, in dem jeweils einen anderen Datenpunkt ausser Betracht gelassen wird. Es gibt 159 Datenpunkte im Datensatz, also wird das Modell 159 Mal neu berechnet.
2. Jeder Datenpunkt wird ein Mal auf der Basis der Prädiktoren vorhersagt, und zwar von dem Modell, in dem es nicht berücksichtigt wurde.
3. Der Unterschied zwischen den 159 Vorhersagen und tatsächlichen Beobachtungen wird berechnet.
4. Anhand dieser Unterschiede werden die folgenden Masse der Vorhersagekraft geschätzt:
    - der durchschnittliche (Mittel, Median) absolute Unterschied zwischen den Vorhersagen und den Beobachtungen;
    - der *root mean square error* (RMSE), d.h., die Wurzel des Mittels der quadrierten Unterschiede;
    - die Varianz der Unterschiede;
    - und davon abgeleitet: die von den Vorhersagen erfasste Varianz in den Beobachtungen (vgl. R^2).

```r
# Neue Spalte für die Modellvorhersagen kreieren
# (im Moment leer)
d$loocv_Vorhersage <- NA

# 159 Mal 1 Datenpunkt auslassen, Modell fitten und Datenpunkt vorhersagen
for (i in 1:nrow(d)) {
  # Modell ohne Datenpunkt i
  modell_ohne <- lm(CorrectSpoken ~ c.WST + c.Englisch + c.Raven + c.DS,
                    data = d[-i, ])
  # Vorhersage des i. Datenpunkts
  d$loocv_Vorhersage[i] <- predict(modell_ohne, newdata = d[i, ])
}

# Fehler berechnen
d$loocv_Fehler <- d$CorrectSpoken - d$loocv_Vorhersage

# Schätzung des durchschnittlichen absoluten Fehlers für neue Daten
mean(abs(d$loocv_Fehler))   # Mittel der absoluten Fehler
```

```
## [1] 3.24237
```

```r
median(abs(d$loocv_Fehler)) # Median der absoluten Fehler
```

```
## [1] 2.572746
```

```r
# Schätzung des RMSE
sqrt(mean(d$loocv_Fehler^2))
```

```
## [1] 4.094376
```

```r
# Varianz der Fehler
var(d$loocv_Fehler)
```

```
## [1] 16.86993
```

```r
# In den Beobachtungen war die Varianz etwa 29.5.
# (29.5-16.9)/29.5 = 0.43 ist also die von den Vorhersagen erfasste Varianz:
1 - var(d$loocv_Fehler)/var(d$CorrectSpoken)
```

```
## [1] 0.4289494
```

Es ist also zu vermuten, dass die Modellgleichung des ursprunglichen Modells (`alles.lm`) neue Daten mit einem durchschnittlichen Fehler von etwa 3.2 Punkten vorhersagen kann (Mittel der Fehler) und etwa 43% der Varianz in neuen Daten erfassen kann. Diese letzte Zahl ist niedriger als die R^2- und R^2_{adj}-Werte, die das Modell ausspuckt, aber beantwortet auch eine andere Frage (Wie gut kann *dieses* Modell neue Daten *vorhersagen*? vs. Wie gut kann ein *neu geschätztes* Modell neue Daten *erfassen*?)

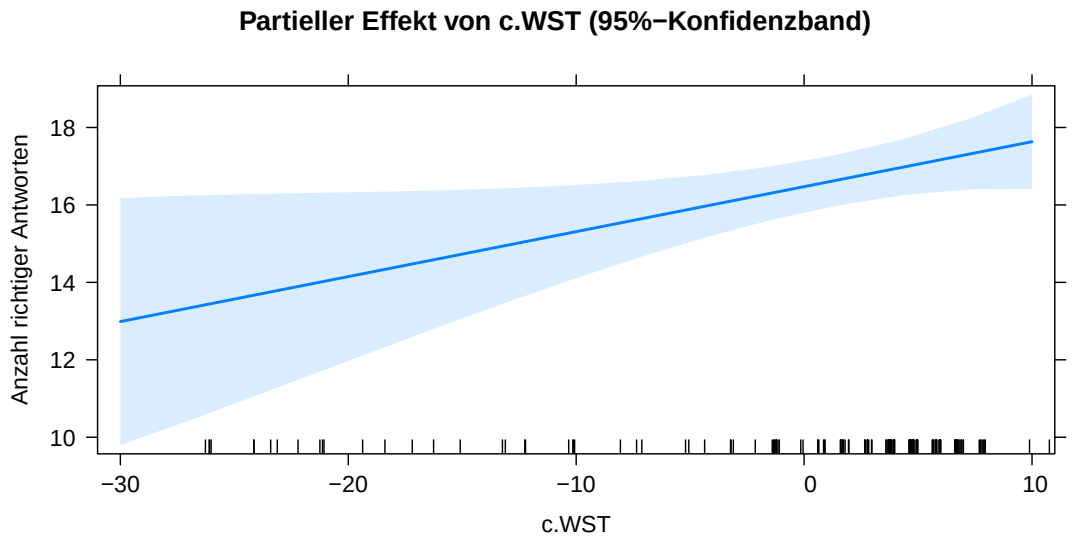


Abbildung 11.3: Der 'partielle' (*ceteris paribus*) Effekt der `c.WST`-Variablen im Modell mit seinem 95%-Konfidenzband. Defaultmässig fixiert das `effects`-Package für solche Grafiken die anderen Prädiktoren bei ihrem Stichprobenmittel. Diese Grafik zeigt also, dass laut dem Modell die durchschnittliche Anzahl richtiger Antworten von Versuchspersonen mit einem `c.WST`-Wert von -20 und `c.Englisch`, `c.Raven` und `c.DS`-Werten von 0 (dem jeweiligen Stichprobenmittel für diese Prädiktoren) bei knapp über 14 liegt.

Für mehr Informationen zum Einschätzen der Vorhersagekraft von Modellen, siehe Harrell (2001, Kapitel 5), Kuhn & Johnson (2013) und Yarkoni & Westfall (2017). Wichtig ist noch, dass dieses Verfahren davon ausgeht, dass die Datenpunkte alle unabhängig voneinander sind. Wenn dies nicht der Fall ist, lohnt es sich einen Blick auf Roberts et al. (2017) zu werfen.

11.1.3 Die Ergebnisse visualisieren

Die Ergebnisse der Modellierung kann man grafisch darstellen. In den letzten Kapiteln wurde bereits in optionalen Kästchen erklärt, wie man dies für einfache (statt mehrfacher) Regressionsmodelle und für Modelle mit Interaktionen von Hand machen kann. Für mehrfache Regressionsmodelle gelten die gleichen Prinzipien. Man kann solche Grafiken auch mit spezialisierten Funktionen zeichnen lassen—aber ich finde es trotzdem sinnvoll, dass Sie wissen, wie diese hinter den Kulissen funktionieren.

Ein Package, das solche spezialisierten Funktionen liefert, ist das `effects`-Package (siehe Fox, 2003). Abbildung 11.3 zeigt das Ergebnis der folgenden Befehle und erklärt, wie solche Grafiken zu interpretieren sind.

```
library(effects)
plot(effect("c.WST", mod = alles.lm),
     ylab = "Anzahl richtiger Antworten",
     main = "Partieller Effekt von c.WST (95%-Konfidenzband)")
```

McElreath (2016) nennt solche Grafiken *counterfactual plots*: Sie zeigen, wie sich (laut dem Modell) der outcome ändert, wenn man die Werte eines Prädiktors variiert, ohne dabei die Werte der anderen Prädiktoren zu ändern. Dies ist in der Realität nicht unbedingt möglich.

Ich finde das `effects`-Package ziemlich nützlich, um schnell die Ergebnisse einer Modellierung zu visualisieren. Wenn man ein paar Details (inkl. ästhetische) selber einstellen möchte, finde ich es aber einfacher, sie von Hand zu zeichnen:

```
# Neuer Dataframe, in dem c.WST variiert,
# die anderen Prädiktoren aber nicht
# (fixiert beim Stichprobenmittel):
wst_df <- expand.grid(c.WST = seq(from = min(d$c.WST),
                                  to = max(d$c.WST),
                                  length.out = 100),
                      c.Englisch = mean(d$c.Englisch),
                      c.Raven = mean(d$c.Raven),
                      c.DS = mean(d$c.DS))
# wst_df anzeigen (hier nicht gemacht)
# wst_df

# Modellvorhersagen + Konfidenzintervalle (hier 90%)
wst_predict <- predict(alles.lm, newdata = wst_df,
                       interval = "confidence",
                       level = 0.90)

# cbind = Spalten (Columns) kombinieren
wst_df <- cbind(wst_df, wst_predict)

# wst_df anzeigen (hier nicht gemacht)
# wst_df

ggplot(wst_df,
       aes(x = c.WST,
           y = fit,
           ymin = lwr,
           ymax = upr)) +
  geom_line() +
  geom_ribbon(fill = "lightblue",
              alpha = 0.5) + # je kleiner 'alpha', desto mehr Transparenz
  xlab("WST-Ergebnis\n(zentriert ums Stichprobenmittel)") +
  ylab("Anzahl richtiger Antworten (gesprochen)") +
  ggtitle("Modellierter Zusammenhang zwischen WST
          und Anzahl richtiger Übersetzungsversuche",
          subtitle = "Trendlinie mit 90%-Konfidenzband")
```

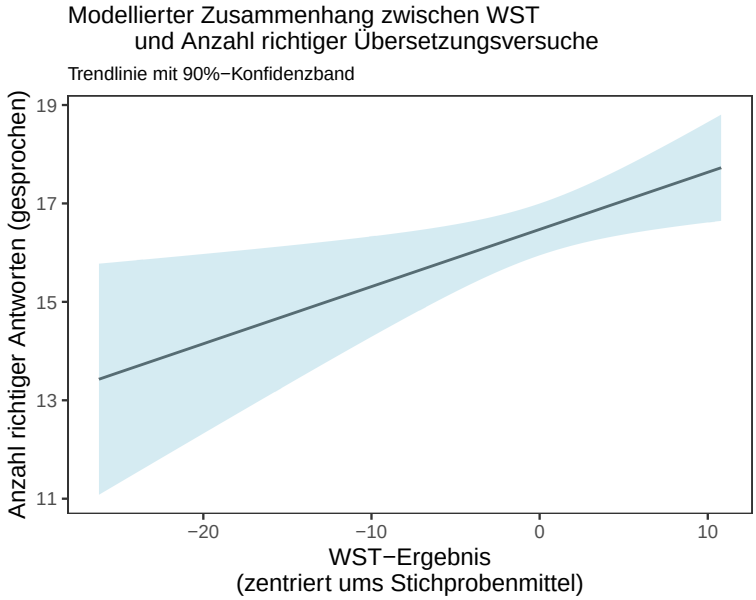


Abbildung 11.4: Der von Hand gezeichnete partielle Effekt für `c.WST` mit einem 90%-Konfidenzband.

Die Prädiktoren im Modell wurden alle um ihr Stichprobenmittel zentriert, aber eventuell will man in der Grafik die Werte der ursprunglichen Variablen (nicht der zentrierten) darstellen. Hierzu kann man dem neu kreierten Datensatz `wst_df` die ursprunglichen Pendants der zentrierten Werte hinzufügen. Um die Werte zu zentrieren, haben wir das Stichprobenmittel von ihnen abgezogen; um die Werte zu dezentrieren, zählen wir das Stichprobenmittel wieder bei ihnen auf. Dann kann die Grafik gezeichnet werden mit der dezentrierten Variablen auf der x-Achse. (Ergebnis hier nicht gezeigt.)

```
wst_df$WST <- wst_df$c.WST + mean(d$WST.Right)

ggplot(wst_df,
       aes(x = WST,
           y = fit,
           ymin = lwr,
           ymax = upr)) +
  geom_line() +
  geom_ribbon(fill = "lightblue",
              alpha = 0.5) + # je kleiner 'alpha', desto mehr Transparenz
  xlab("WST-Ergebnis") +
  ylab("Anzahl richtiger Antworten (gesprochen)") +
  ggtitle("Modellierter Zusammenhang zwischen WST und
          Anzahl richtiger Übersetzungsversuche",
          subtitle = "Trendlinie mit 90%-Konfidenzband")
```

Für mehr Informationen, siehe etwa *Tutorial: Adding confidence bands to effect displays* (12.5.2017).

Aufgabe 1 Zeichnen Sie solche Grafiken für einen anderen Prädiktor, und zwar mit beiden Methoden.

Aufgabe 2 Fügen Sie dem Modell den kategorischen Prädiktor `Sex` hinzu. Wie interpretieren Sie den geschätzten Koeffizienten für diesen zusätzlichen Prädiktor?

Aufgabe 3 Zeichnen Sie auf der Basis des neuen Modells einen *counterfactual plot* für den Prädiktor `DS.Span` (oder seine zentrierte Version `c.DS`) und erklären Sie, was die Trendlinie in dieser Grafik darstellt. (Ihre Antwort wird davon abhängen, wie Sie im vorigen Schritt genau vorgegangen sind.)

11.1.4 Übung: Direktere und indirektere Zusammenhänge

Diese Aufgabe basiert auf einem leider fiktiven Datensatz, der zu Demonstrationszwecken von Strobl et al. (2009) zusammengestellt wurde.

1. Lesen Sie den Datensatz `readingSkills.csv` ein. Die Spalte `score` enthält (fiktive) Ergebnisse bei einem Lesetest von 200 Kindern; die Spalte `shoeSize` enthält ihre Schuhgrösse.
2. Stellen Sie den Zusammenhang zwischen den Schuhgrössen und den Ergebnissen beim Lesetest grafisch dar (Lesetestergebnisse als outcome).
3. Modellieren Sie diesen Zusammenhang in einem einfachen Regressionsmodell. Wie interpretieren Sie die Ergebnisse dieser Analyse?
4. Fügen Sie dem Modell jetzt die `age`-Variable als Prädiktor hinzu. Wie ändert sich der Parameter für `shoeSize`? Wie erklären Sie sich diese Veränderung? (Tipp: Stellen Sie auch einmal den Zusammenhang zwischen `age` und `shoeSize` grafisch dar.)

Mehrfache Regression erlaubt es (oft), 'direktere' von 'indirekteren' Effekten zu trennen. `shoeSize` und `score` variieren zwar zusammen, aber der Zusammenhang mit `shoeSize` lässt sich durch ihren gemeinsamen Zusammenhang mit `age` erklären: Ältere Kinder haben grössere Füsse und schneiden besser beim Lesetest ab. Zugegebenermassen sind nicht alle Fälle so eindeutig wie dieser; siehe auch *Controlling for confounding variables in correlational research: Four caveats* (24.8.2015). Ausführliche Referenzen hierzu sind etwa Christenfeld et al. (2004) und Westfall & Yarkoni (2016).

11.2 Mit Kontrollvariablen die Fehlervarianz senken

Die Genauigkeit, mit der ein Modellparameter geschätzt wird, hängt von der Datenmenge und der Fehlervarianz ab. Dies zeigte sich bereits in der Formel zur Berechnung des Standardfehlers eines Mittels ($\widehat{SE} = \frac{s}{\sqrt{n}}$). Die Genauigkeit kann man also erhöhen, indem man mehr Daten sammelt oder indem man die Fehlervarianz senkt. Letzteres kann man wiederum auf ein paar Arten und Weisen bewirken:

- indem man zuverlässigere Instrumente, die das Konstrukt, für das man sich interessiert, genauer messen, verwendet;
- indem man den Einfluss von Störfaktoren auf die outcome-Variable im Design der Studie reduziert. Wenn Alter ein möglicher Störfaktor wäre, könnte man zum Beispiel nur Teilnehmende in einer bestimmten schmalen Altersspanne rekrutieren. Hier muss man natürlich eine Abwägung zwischen der Genauigkeit der Ergebnisse und ihrer Generalisierbarkeit machen.
- indem man den Einfluss von Störfaktoren, die man nicht von der Studie hat ausschliessen können, statistisch berücksichtigt.

Dieser Abschnitt ist der letzten Möglichkeit gewidmet. Zur Illustration dient der fiktive `readingSkills`-Datensatz von Strobl et al. (2009) aus dem letzten Abschnitt. Diesmal interessieren wir uns aber für den Unterschied in Lesekenntnissen zwischen L1- und L2-SprecherInnen (`nativeSpeaker`).

1. Stellen Sie den Zusammenhang zwischen `score` und der Variablen `nativeSpeaker` grafisch dar. Beschreiben Sie, was der Grafik entnommen werden kann.
2. Modellieren Sie jetzt dieselben Variablen in einem Regressionsmodell mit `score` als outcome und konstruieren Sie ein 95%-Konfidenzintervall für den geschätzten Koeffizienten von `nativeSpeaker`.
3. Fügen Sie dem Regressionsmodell nun die Altersvariable hinzu und konstruieren Sie das 95%-Konfidenzintervall für den geschätzten Koeffizienten von `nativeSpeaker` erneut. Was stellen Sie im Vergleich zum Konfidenzintervall aus dem letzten Schritt fest? Wie erklären Sie sich dies?

Mehrfache Regression erlaubt uns, Variable zu berücksichtigen, die uns vielleicht zwar nicht stark interessieren (in diesem Beispiel: `age`), die aber dennoch mit dem outcome zusammenhängen. Hierdurch wird der Restfehler kleiner, was wiederum die Standardfehler der übrigen Parameterschätzungen verkleinert. Auch wenn Ihnen der Einfluss irgendeiner Variablen egal ist, kann es sich daher lohnen, diese Variable trotzdem mitzuerheben, wenn sie den Restfehler eingreifend reduzieren kann, insbesondere wenn die zusätzliche Variable leicht zu erheben ist.

Dieses Prinzip sollte man auch nicht übertreiben: Beschränken Sie sich lieber auf ein paar zusätzliche Variable, von denen Sie ziemlich sicher wissen, dass sie die

Fehlervarianz reduzieren, statt jede Menge Variabler zu erheben, die je nach Wetterlage und Mondposition eventuell einen Einfluss haben könnten.

Mehrfache Regression/ANCOVA hat übrigens diesen Vorteil, *auch wenn (sogar: besonders wenn!) die Gruppen, die man vergleichen möchte, sich hinsichtlich der zusätzlichen Variablen nicht unterscheiden*; siehe hierzu noch Vanhove (2015, Abschnitt 2, für eine kurze Erklärung) und Mutz et al. (2017, für eine ausführlichere).

Post-treatment variables. Manchmal ist es möglich, dass das *treatment* (z.B. Instruktionen in experimenteller vs. Kontrollkondition) nicht nur Unterschiede im outcome herbeiführen, sondern auch (direkt oder indirekt) in den zusätzlichen Variablen, die man hätte mitberücksichtigen wollen, um die Fehlervarianz zu senken. Wenn man diese Variablen erst *nach* dem *treatment* misst, sollte man sich auf jeden Fall nochmals genauer überlegen, ob es sinnvoll ist, diese Variablen dem Modell hinzuzufügen. Eine Gefahr ist nämlich, dass man einen bestehenden Effekt des *treatments* 'kaputtkontrolliert'. Die Interpretation des Modells ist auf jeden Fall einfacher, wenn solche Variablen *vor* dem *treatment* erhoben werden oder wenn es implausibel ist, dass das *treatment* die zusätzlichen Variablen beeinflussen kann.

Für mehr Informationen hierzu, siehe Huitema (2011, S. 209–213) und Rohrer (2018).

11.3 Annahmen überprüfen

Die Annahmen von mehrfacher Regression/ANCOVA sind genau die gleichen wie sonst beim allgemeinen linearen Modell. Um die Annahmen der Linearität, Homoskedastizität und Normalität zu überprüfen, kann man etwa die Methode im Blogeintrag *Checking model assumptions without getting paranoid* (25.4.2018; nach Buja et al., 2009 und Majumber et al., 2013; siehe auch Vanhove, 2018a) verwenden.

Dazu noch: Wenn etwa die Annahme der Normalität nicht ganz genau zutrifft, heisst das nicht, dass die Ergebnisse des Modells nicht zuverlässig sind. Mit der Erfahrung bekommt mann ein gewisses Gefühl dafür, ab wann Verletzungen bestimmter Annahmen problematisch werden. Im Zweifelsfall können Sie die Ergebnisse mit einer Bootstrapmethode überprüfen, die weniger Annahmen macht. Oft kommt dabei dann mehr oder weniger das Gleiche heraus, was dann heisst, dass Sie auf beiden Ohren schlafen können. Wichtig ist vor allem, dass das Modell sinnvoll und relevant ist. Bei stark schiefverteilten oder bimodalen Residuen mag die Regressionslinie die konditionellen Mittel zwar noch immer erfassen, aber diese sind dann weniger relevant.

Teil IV

Signifikanztests

Kapitel 12

Die Logik des Signifikanztests

Für viele Forschende scheint der Sinn und Zweck einer statistischen Analyse das Produzieren eines möglichst 'signifikanten' p-Wertes zu sein. Meines Erachtens lässt sich dies dadurch erklären, dass solche Forschende die Bedeutung von p-Werten falsch verstehen. Um derartige Missverständnisse vorzubeugen, introduziert dieses Kapitel p-Werte zunächst rein konzeptuell, ohne zusätzliche Mathe zu verwenden.

12.1 Randomisierung als Inferenzbasis

12.1.1 Ein einfaches Experiment

Stellen Sie sich folgendes Experiment vor (inspiriert von Guiora et al., 1972, aber siehe auch Renner et al., 2018). Um den Effekt von Alkohol auf die Sprechgeschwindigkeit zu untersuchen, werden sechs Germanistikstudierende zu einem Experiment eingeladen. (Sechs Teilnehmende ist natürlich eine zu geringe Anzahl, aber diese Erklärung bleibt dadurch übersichtlich.) Nach dem Zufallsprinzip wird die Hälfte der Studierenden der Experimentalgruppe und die andere Hälfte der Kontrollgruppe zugeteilt. Die Versuchspersonen in der Experimentalgruppe müssen ein Videofragment beschreiben, nachdem sie zuerst 5 Deziliter alkoholhaltiges Bier getrunken haben. Die Versuchspersonen in der Kontrollgruppe erledigen dieselbe Aufgabe, trinken statt alkoholhaltigem aber 5 Deziliter alkoholfreies Bier. Die Versuchspersonen wissen nicht, ob das Bier, das sie trinken, alkoholfrei oder alkoholhaltig ist. Gemessen wird die Sprechgeschwindigkeit in Silben pro Sekunde. Auch die Mitarbeitenden, die die Silben zählen, wissen nicht, welche Versuchspersonen welcher Kondition zugeteilt wurden (*double-blind experiment*).

Denkaufgabe Wieso sollten im Idealfall weder die Versuchspersonen noch die Mitarbeitenden wissen, welche Versuchspersonen welcher Kondition zugeordnet wurden?

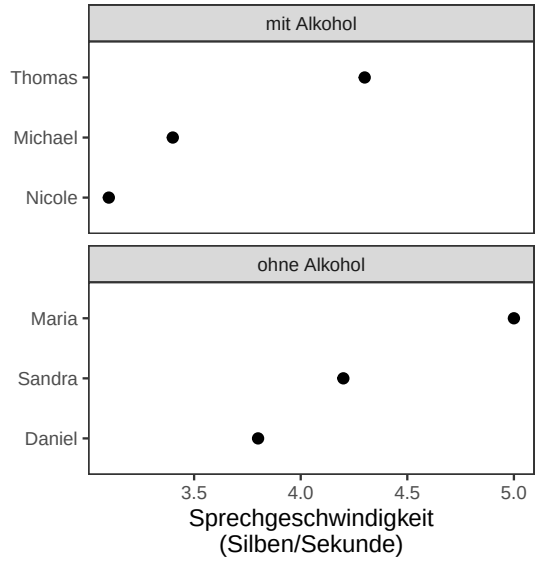


Abbildung 12.1: Ergebnisse eines fiktiven Experiments.

Von den sechs Studierenden wurden Sandra, Daniel und Maria nach dem Zufallsprinzip der Kontrollgruppe zugeteilt, während Nicole, Michael und Thomas der Experimentalgruppe zugeteilt wurden. Die Versuchspersonen in der Kontrollgruppe äusserten beim Beschreiben des Videofragments 4.2, 3.8 und 5.0 Silben pro Sekunde; diejenigen in der Experimentalgruppe 3.1, 3.4 und 4.3 Silben pro Sekunde; siehe Abbildung 12.1. Es ist klar, dass die Versuchspersonen in der Kontrollgruppe eine höhere durchschnittliche Sprechgeschwindigkeit haben als jene in der Experimentalgruppe: Der Unterschied zwischen den Gruppenmitteln beträgt etwa 0.73 Silben pro Sekunde. Können wir daraus schliessen, dass das Trinken von alkoholhaltigem vs. alkoholfreiem Bier diesen Unterschied mitverursacht hat, oder beruht er auf reinem Zufall?

12.1.2 Warum randomisieren?

Die Versuchspersonen wurden nach dem Zufallsprinzip einer der Gruppen zugeordnet. So wurde sichergestellt, dass die Ergebnisse nicht *systematisch* verzerrt wurden. Zum Beispiel gibt es in der Kontrollgruppe zwei Frauen und in der Experimentalgruppe nur eine. Aber dieser Unterschied ist rein zufällig. Das Ziel von Randomisierung ist eben *nicht*, perfekt äquivalente Gruppen zu generieren, sondern eine systematische Verzerrung vorzubeugen – sowohl was bekannte als was unbekannte Störvariablen betrifft. (Siehe Vanhove, 2015, zu diesem Missverständnis.)

Ausserdem handelt es sich in diesem Fall um ein *double-blind experiment*: Weder die Versuchspersonen selber noch die auswertenden Mitarbeitenden wussten, wer welcher Kondition zugeteilt wurde. Dies beugt eine Verzerrung der Ergebnisse aufgrund von **Erwartungseffekten** vonseiten der Versuchspersonen (*subject-expectancy effect*, vgl. den Placebo-Effekt) oder vonseiten der Forschenden (*observer-expectancy effect*) vor.

Sicher hätten wir dieses Design verfeinern können, etwa indem wir die Herkunft der Versuchspersonen in den beiden Gruppen fixiert hätten (z.B. eine Bünderin, ein Zürcher und eine Bernerin in jeder Gruppe; wer sich für solche raffiniertere Designs interessiert, kann sich ausgewälte Kapitel aus Oehlert, 2010, anschauen) oder indem wir die Sprachgeschwindigkeit der Versuchspersonen auch vor dem Experiment gemessen hätten ('Vortest'), sodass wir diese in der Analyse hätten mitberücksichtigen können. Aber auch ohne solche Raffinesse erlaubt dieses Design dank der Randomisierung und der Blindierung gültige Aussagen.

12.1.3 Die Nullhypothese und Re-Randomisierung

Der Unterschied zwischen den Mitteln der Gruppen beträgt 0.73 Silben pro Sekunde. Da wir ein randomisiertes Experiment ausgeführt haben und somit eine systematische Verzerrung der Ergebnisse vorgebeugt haben, könnten wir daraus sogar schliessen, dass dieser Unterschied z.T. von unserer experimentellen Manipulation *verursacht* wurde: Der Konsum von 5 Deziliter alkoholhaltigem Bier senkt die Sprechgeschwindigkeit.

Bevor wir eine solche kausale Aussage machen, müssen wir uns mit einer trivialeren Erklärung beschäftigen: Vielleicht beruht der Unterschied auf reinem Zufall. Dies ist unsere **Nullhypothese** (H_0), die wir mit einer (ziemlich vagen) **Alternativhypothese** (H_A) kontrasieren können:

- H_0: Der Unterschied zwischen beiden Mitteln ist *nur* dem Zufallsfaktor zuzuschreiben.
- H_A: Der Unterschied ist *auch teilweise* der experimentellen Manipulation zuzuschreiben.

In der 'frequentistischen' Tradition des Nullhypothesen Testens berechnet man, wie wahrscheinlich es ist, den beobachteten Effekt (hier: Unterschied zwischen den Gruppenmitteln) oder noch extremere Effekte anzutreffen, wenn die Nullhypothese denn tatsächlich stimmen würde. Diese Wahrscheinlichkeit bezeichnet man als den p-Wert (p für *probability*). Ist diese Wahrscheinlichkeit gering, dann zieht man daraus die Schlussfolgerung, dass die Annahme, dass die Nullhypothese stimmt, wohl nicht berechtigt ist, und dass auch ein systematischer Effekt im Spiel ist. In der Regel hantiert man dabei eine arbiträre Schwelle (z.B. 5%), unter der der p-Wert als zu klein gilt.

Bevor wir uns einigen konzeptuellen Problemen mit diesem Vorgehen widmen und einige Komplikationen besprechen, schauen wir uns eine Methode an, um den p-Wert zu berechnen. Wenn wir davon ausgehen, dass die Nullhypothese stimmt, dann ist der Unterschied zwischen den Gruppen lediglich das Ergebnis der Randomisierung, also des Zufalls: Unter dieser Annahme hätte Michael auch in der Kontrollgruppe 3.4 Silben pro Sekunde geäussert; ebenso hätte Sandra in der Experimentalgruppe 4.2 Silben pro Sekunde geäussert. Wenn das Zufallsverfahren also statt Michael Sandra der Experimentalgruppe zugeteilt hätte und Alkoholkonsum die Sprechgeschwindigkeit nicht beeinflusst, dann wäre das Mittel der Experimentalgruppe 3.87 gewesen und das der

Kontrollgruppe 4.07 – und hätten wir also eine um 0.20 Silben pro Sekunde höhere Sprechgeschwindigkeit in der Experimentalgruppe festgestellt.

Insgesamt gibt es 20 Möglichkeiten, wie die Experimental- und Kontrollgruppe hätten aussehen können:[1]

```
# Wie viele Möglichkeiten gibt es, um
# 6 Teilnehmende zwei Gruppen von 3 zuzuordnen?
choose(n = 6, k = 3)
```

```
## [1] 20
```

Jede dieser Möglichkeiten ist in Abbildung 12.2 dargestellt. Für jede können wir berechnen, wie gross der Gruppenunterschied ist. (Der R-Code ist dabei nicht wichtig, nur die Logik.)

Abbildung 12.3 stellt die 20 möglichen Gruppenunterschiede dar, die man hätte antreffen können, wenn die Nullhypothese tatsächlich stimmen würde. Im Schnitt betrüge der Gruppenunterschied unter Annahme der Nullhypothese 0, aber je nachdem, welche Versuchsperson welcher Kondition zugeordnet worden wäre, hätte man kleinere aber durchaus auch grössere Unterschiede feststellen können. Dieser Grafik kann man entnehmen, wie ungewöhnlich nun Gruppenunterschiede, die mindestens so gross sind wie der Gruppenunterschied, den wir tatsächlich festgestellt haben, unter Annahme der Nullhypothese wären. Die roten Strichellinien zeigen, dass in 6 von 20 Fällen die Gruppenunterschiede um 0.73 Silben pro Sekunde oder noch mehr voneinander abweichen. Auch wenn die Nullhypothese in unserem Beispiel tatsächlich stimmen würde, hätten wir in 6 der 20 möglichen Stichproben (also in 30% der Fälle) einen Gruppenunterschied von 0.73 Silben pro Sekunde oder sogar noch mehr feststellt. Dies ist unser p-Wert. Da eine Wahrscheinlichkeit von 30% doch beträchtlich ist, würde wohl kaum jemand schlussfolgern, dass wir ausschlaggebende Evidenz gegen die Nullhypothese gesammelt haben. Dies heisst aber *nicht*, dass wir die Nullhypothese 'bestätigt' haben, sondern nur, dass nur wenig statistische Evidenz vorliegt, dass sie abgelehnt werden sollte. (Absenz von Evidenz $\neq$ Evidenz für Absenz!)

12.1.4 Bemerkungen

Verknüpfung zum Forschungsdesign

Der Hypothesentest, den wir soeben durchgeführt haben, ist ein **Randomisierungstest** (manchmal auch **Permutationstest** genannt). Sein Gebrauch wird durch das Forschungsdesign, genauer gesagt: durch die uneingeschränkte Randomisierung und der Blindierung, legitimiert:

[1] Für grössere Gruppen wird diese Zahl schnell zu gross, um das Vorgehen nachvollziehbar darzustellen. So gibt es 137'846'528'820 Möglichkeiten, um eine Gruppe von 40 Teilnehmenden in zwei gleich grossen Gruppen zu verteilen.

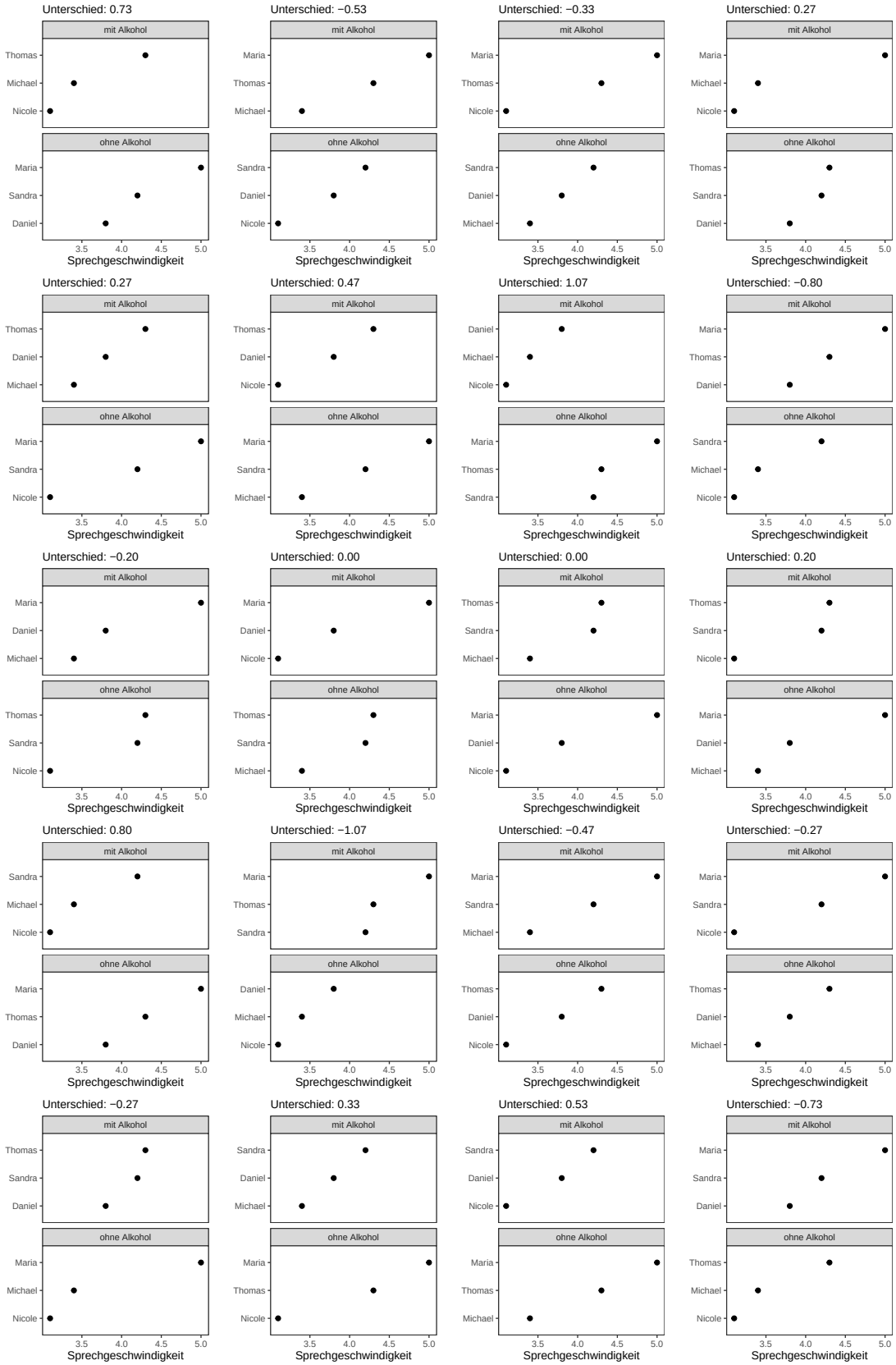


Abbildung 12.2: Die 20 möglichen Ergebnisse laut der Annahme, dass die Unterschiede in der Stichprobe nur der Randomisierung zuzuschreiben sind.

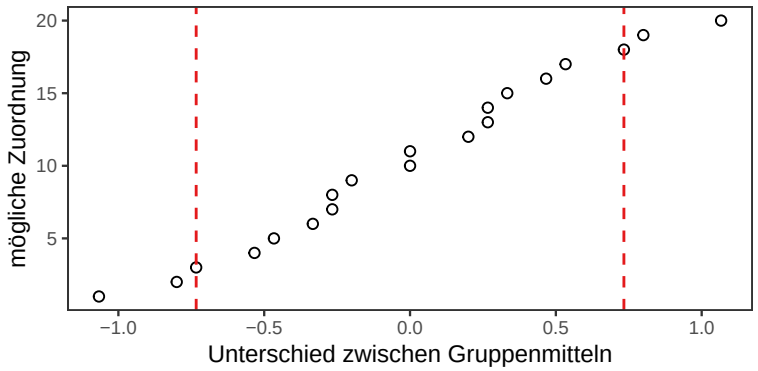


Abbildung 12.3: Die Unterschiede zwischen den Gruppenmitteln in allen 20 möglichen Konstellationen. Die roten senkrechten Strichellinien stellen den in der tatsächlichen beobachteten Unterschied dar (-0.73) und seine Gegenzahl (0.73) dar.

- Wenn es zum Beispiel so gewesen wäre, dass wir Sandra etwa aus medizinischen Gründen nie der 'mit Alkohol'-Gruppe zugewiesen hätten, dann hätte es nicht 20 mögliche Ergebnisse unter der Nullhypothese gegeben, sondern nur 10: Alle Permutationen mit Sandra in der 'mit Alkohol'-Gruppe hätten nicht vorkommen können. Dies hätte man dann in der Analyse berücksichtigen müssen.
- Wenn es so gewesen wäre, dass Michael und Nicole unbedingt in der gleichen Kondition getestet werden wollten, dann hätte es auch keine 20 möglichen Ergebnisse unter der Nullhypothese gegeben, sondern wiederum nur 10: Alle Permutationen mit Nicole und Michael in anderen Konditionen hätten nicht vorkommen können. Auch dies hätte man dann berücksichtigen müssen.
- Wenn es so gewesen wäre, dass wir zwecks Ausgleichs der beiden Gruppen mindestens eine Frau und einen Mann jeder Kondition hätten zuordnen wollen, dann hätten die zwei Permutationen mit lediglich Männern oder Frauen in einer Kondition nicht vorkommen können. Ein solches Design mag unter Umständen zwar sinnvoll sein, aber diese Einschränkung hätte man berücksichtigen müssen.

Züfallige Auswahl vs. zufällige Zuordnung

Dieses Experiment und seine Auswertung illustrieren weiter den Unterschied zwischen **zufällige Zuordnung** und **zufällige Auswahl**: Wir haben unsere Versuchspersonen zufällig den experimentellen Konditionen zugeordnet, aber wir haben sie nicht zufällig aus irgendeiner Population gewählt. Wenn wir überzeugendere Evidenz für eine Wirkung der experimentellen Manipulation gefunden hätten, dann hätten wir folglich daraus immer noch nicht ohne Weiteres schliessen können, dass die experimentelle Manipulation einen Effekt in einer bestimmten *Population* hätte. Dazu hätten wir sowohl die Versuchsperson zufällig aus dieser Population wählen müssen (*random sampling*) und diese dann zufällig den Kondition zuweisen müssen (*random assignment*). Ohne eine zufällige Auswahl beruht eine solche Schlussfolgerung auf einer (oft impliziten)

sachlogischen Argumentation—nicht auf einer statistischen Gegebenheit. (Das muss nicht heissen, dass eine solche Schlussfolgerung dann falsch ist. Aber es ist wichtig zu erkennen, dass es keine statistische Frage ist.) Diese Nuance entspricht dem Unterschied zwischen **interner Validität** (Ist der Unterschied oder der Effekt, der wir in dieser Stichprobe beobachtet haben, der experimentellen Manipulation zuzuschreiben?) und **externer Validität** (Lässt sich dieser Befund über die Stichprobe hinaus generalisieren?)[2]

Statistische und wissenschaftliche Hypothesen

Die für den Test formulierten Null- und Alternativhypothesen sind statistische Hypothesen. Diese haben einen erstaunlich geringen wissenschaftlichen Inhalt: Die Nullhypothese besagt lediglich, dass die beobachteten Muster rein auf Zufall basieren; die Alternativhypothese, dass sie nicht rein auf Zufall basieren. Worauf die Muster dann—neben Zufall—schon zurückzuführen wären, darüber macht die Alternativhypothese keine Aussage. In unserem Beispiel wären ein paar mögliche Auslöser die folgenden:

- Alkohol senkt die Hemmungen beim Sprechen, was dann wiederum die Sprechgeschwindigkeit erhöht.
- Leute mit einem halben Liter alkoholhaltigem Bier intus drücken sich eher in einfachen Strukturen und Floskeln aus, die sie schneller aussprechen als schwierigere Strukturen und neue Phrasen.
- Die auswertenden Mitarbeitenden haben doch irgendwie mitbekommen, wer welcher Gruppe zugeordnet wurde, und haben sich bewusst oder unbewusst bei ihren Auswertungen von diesem Wissen leiten lassen.

Lange Rede, kurzer Sinn. Auch wenn ein Muster unter der Nullhypothese sehr implausibel ist, sagt Ihnen der Signifikanztest noch nicht, aus welchem Grund das Muster möglicherweise zu Stande gekommen ist.

Weiter ist zu bemerken, dass die Alternativhypothese auch statistisch vage ist. Zum Beispiel besagt sie nicht, wie gross eine allfällige, von Alkohol ausgelöste Änderung der Sprechgeschwindigkeit sein könnte. Sie besagt nicht einmal, ob diese Änderung eine Beschleunigung oder eine Verlangsamung wäre.

Ein- und zweiseitige Tests

Im obigen Beispiel wurde ein zweiseitiger Signifikanztest verwendet: Es wurde nicht nur berechnet, wie wahrscheinlich es unter Annahme der Nullhypothese wäre, einen

[2] Wer sich für die Effizienz didaktischer Methoden interessiert ist, muss wohl die externe Validität der Untersuchung berücksichtigen. Aber für etwa experimentelle Psychologen ist externe Validität nicht unbedingt so wichtig (Mook, 1983): Für sie *kann* es wichtiger sein, zu zeigen, dass eine Manipulation überhaupt einen Effekt erzeugen *kann*, ohne dass die Grenzen dieses Befunds schon erprobt werden müssen.

Unterschied von mindestens 0.73 Silben pro Sekunde zugunsten der Kontrollgruppe (dem beobachteten Ergebnis) anzutreffen, sondern auch, wie wahrscheinlich es unter Annahme der Nullhypothese wäre, einen Unterschied von mindestens 0.73 Silben pro Sekunde zugunsten der Experimentalgruppe anzutreffen. Der Grund hierfür ist, dass die Alternativhypothese vage war und wir lediglich die Vermutung aufgestellt haben, dass Alkoholkonsum *irgendeine* Änderung der Sprechgeschwindigkeit herbeiführen dürfte.

In der Literatur trifft man ab und zu auch einseitige Tests an. Bei solchen Tests schaut man sich nur eine der beiden Wahrscheinlichkeiten ('grösser' oder 'kleiner') an. Die p-Werte von einseitigen Tests sind kleiner als jene von zweiseitigen Tests. Einseitige Tests können sinnvoll sein, wenn man *im Vorhinein* klar spezifiziert hat, dass nur ein Unterschied in einer bestimmten Richtung mit der wissenschaftlichen Hypothese, die hinter der Arbeit steckt, kompatibel ist. Für eine kurze Übersicht, siehe `http://daniellakens.blogspot.ch/2016/03/one-sided-tests-efficient-and-underused.html`.

Vorsicht. Man sollte sich nicht zuerst die Daten anschauen, und dann entscheiden, dass man einen einseitigen Test verwenden möchte—etwa, weil der zweiseitige Test ein nicht-signifikantes Ergebnis produziert.

Im Zweifelsfall. Wenn Sie auch den geringsten Zweifel haben, ob ein ein- oder zweiseitiger Test angebracht ist, berechnen Sie dann den zweiseitigen Test:

> Two sided tests should be used unless there is a very good reason for doing otherwise. If one sided tests are to be used the direction of the test must be specified in advance. One sided tests should never be used simply as a device to make a conventionally non-significant difference significant. (Bland & Altman, 1994)

12.2 Zur Bedeutung des p-Wertes

Wie das Beispiel zeigt, ist p die Wahrscheinlichkeit, dass man das beobachtete Muster (hier: einen Gruppenunterschied von 0.73 Silben pro Sekunde) *oder ein noch extremeres Muster* feststellen würde, *wenn die Nullhypothese tatsächlich stimmen würde*. Die alternativen Gruppenunterschiede haben wir ja unter der Annahme, dass der beobachtete Gruppenunterschied nur durch Zufall zu Stande gekommen ist, generiert.

Sämtliche andere Definitionen und Interpretationen des p-Wertes sind schlicht und einfach falsch. Zur Vorbeugung einiger häufiger Missverständnisse:

- Der p-Wert ist *nicht* die Wahrscheinlichkeit, dass die Nullhypothese stimmt. Wir können also nicht schlussfolgern, dass es eine Wahrscheinlichkeit von 30% gibt, dass H_0 stimmt.

- Der p-Wert ist auch nicht das Komplement der Wahrscheinlichkeit, dass die Alternativhypothese stimmt. Wir können in diesem Beispiel also *nicht* schlussfolgern, dass H_A mit $1 - 0.3 = 70\%$ Wahrscheinlichkeit zutrifft.
- Der p-Wert ist nicht das Komplement der Wahrscheinlichkeit, dass sich das beobachtete Ergebnis in einer Replikationsstudie bestätigen würde.[3]

Solche—und andere—Missverständnisse trifft man geläufig an, sogar manchmal in Statistikeinführungen, die sich an Psychologie- und Linguistikstudierende richten! Für weitere Missverständnisse, siehe Goodman (2008) und Greenland et al. (2016).

Viele Forschende unterscheiden zwischen 'statistisch signifikanten' und 'statistisch nichtsignifikanten' p-Werten. Dabei gelten p-Werte unter einer bestimmten Schwelle für sie als statistisch signifikant. Bei einem statistisch signifikanten p-Wert schlussfolgern diese Forschenden dann, dass die Daten nach der Nullhypothese zu unwahrscheinlich sind, sodass sie diese Hypothese zugunsten der Alternativhypothese ablehnen. Die Signifikanzschwelle (als α bezeichnet) kann im Prinzip von den Forschenden arbiträr festgelegt werden; in den Sozial- und Geisteswissenschaften liegt sie in der Regel aber fast ausnahmslos bei $\alpha = 0.05$—dies grundsätzlich aus keinem anderen Grund, als dass eine Hand fünf Finger zählt.[4]

Schreibtipp. In der Statistik ist 'Signifikanz' ein technischer Begriff, der nicht mit dem alltäglicheren Begriff von praktischer oder theoretischer Signifikanz oder Bedeutung verwechselt werden soll. Versuchen Sie in Ihren eigenen Arbeiten, diese Zweideutigkeit zu vermeiden.

12.3 Fehlentscheide

Auch wenn man es sich wohl gerne anders wünschte, bieten Signifikanztests keine Sicherheit. Traditionellerweise spricht man beim Signifikanztesten (engl.: *null hypothesis significance testing* oder *NHST*) von zwei Arten von Fehlentscheiden, die man treffen kann.

Die erste Art von Fehlentscheid ist, dass man eine tatsächlich zutreffende Nullhypothese ablehnt. Wer sich der traditionellen α-Schwelle von 5% bedient, wird tatsächlich zutreffende Nullhypothesen mit einer Wahrscheinlichkeit von jeweils 5% ablehnen—vorausgesetzt, dass die Daten richtig analysiert werden. Diese Art von Fehlentscheid nennt man einen **Fehler der ersten Art** (engl.: *Type-I error*; auch: falsch positiv, also etwas finden, was nicht da ist). Da man α selber definieren kann bzw. da α *de facto* bereits

[3] Ich habe keine blasse Ahnung, wo dieses Missverständnis herkommt, aber diesen Kommentar habe ich einmal in einem Gutachten erhalten.

[4] Neuerdings gibt es, insbesondere auf den sozialen Medien, eine rege Diskussion darüber, ob man die Signifikanzschwelle nicht umdefinieren sollte oder abschaffen sollte. Siehe hierzu Benjamin et al. (2018), Lakens et al. (2018), McShane et al. (2017) und Amrhein et al. (2017).

auf 0.05 vordefiniert wurde, ist die Häufigkeit von Fehlern der ersten Art im Prinzip festgelegt: Nur 5% aller statistischen Nullhypothesen würde man fälschlicherweise ablehnen, wenn man die Daten richtig analysiert. In der Praxis tauchen hier jedoch einige Komplikationen auf, denen wir uns zu einem späteren Zeitpunkt widmen; siehe Kapitel und 13 und 16.

Wenn nun H_0 nicht zutrifft (d.h., das Muster lässt sich nicht nur durch Zufall erklären), dann besteht trotzdem die Gefahr, dass man einen p-Wert über der α-Schwelle findet. In solchen Fällen würde man die Nullhypothese nicht ablehnen, obwohl dies eigentlich schon erwünscht wäre. Diese Art von Fehlentscheid nennt man einen **Fehler der zweiten Art** (engl.: *Type-II error*; auch: falsch negativ, also etwas nicht finden, was schon da ist). Die Wahrscheinlichkeit eines Fehlers der zweiten Art wird als β bezeichnet (nicht zu verwirren mit den βs aus den Regressionsmodellen); das Komplement von β, $1 - \beta$, nennt man die statistische **power** eines Tests. β (und somit die power) können nicht ohne Weiteres festgelegt werden, denn diese Wahrscheinlichkeit hängt von vier Faktoren ab:

1. wie stark das eigentliche Muster denn wäre, wenn die Nullhypothese nicht zutrifft (z.B. wie stark moderater Alkoholkonsum die Sprechgeschwindigkeit beeinflusst): Je stärker das Muster (z.B. je grösser der Unterschied), desto höher die power;
2. wie gross die Fehlervarianz ist (z.B. wie stark die Teilnehmenden innerhalb jeder Gruppe voneinander abweichen): Je grösser die Fehlervarianz, desto niedriger die power;
3. wie gross die Datenmenge ist: Je mehr Daten, desto grösser die power;
4. welcher Test verwendet wird und ob ihre Annahmen ungefähr erfüllt sind.

Mittels Powerberechnung (siehe Kapitel 14) kann man die power eines Signifikanztests einschätzen. Siehe hierzu Cohen (1977, 1992). Für ein verwandter Ansatz, siehe Kelley et al. (2003). Hier sei bereits darauf hingewiesen, dass man für sinnvolle und genaue Powerberechnung schon *vor* der Untersuchung über viele Informationen über den Forschungsgegenstand verfügen muss: Was für Effektgrösse erwartet man bzw. welche Effektgrösse würde man als interessant betrachten? Wie wird man die Daten analysieren? Und wie gross wird die Variabilität der Daten sein?

	H_0 stimmt	H_0 stimmt nicht
$p < \alpha$	Fehler der 1. Art (α)	OK ($1 - \beta$)
$p > \alpha$	OK ($1 - \alpha$)	Fehler der 2. Art (β)
Total	$\alpha + (1 - \alpha) = 100\%$	$(1 - \beta) + \beta = 100\%$

Die Interpretation von nicht-signifikanten p-Werten. Aufgrund des Fehlers der zweiten Art kann man bei einem nicht-signifikanten Ergebnis weder schlussfol-

gern, dass es einen Unterschied gibt, noch dass es *keinen* gibt: Es ist immer möglich, dass man den Unterschied lediglich nicht gefunden hat. Wenn Sie irgendwo lesen, dass A und B sich nicht signifikant voneinander unterschied und daher einander gleich (oder 'grundsätzlich gleich') sind, ist dies in der Regel nur bequeme Rhetorik: **Absenz von Evidenz ist nicht gleich Evidenz für Absenz**. Schmidt (1996) nennt diesen Fehlschluss übrigens "the most devastating of all to the research enterprise" (S. 126).

Manchmal trifft man auch den Begriff **Fehler der dritten Art** (*Type-III error*) an. Dieser wird unterschiedlich definiert. Für manche ist ein Fehler der dritten Art, wenn man die richtige Antwort auf eine falsche Frage gibt; andere verwenden ihn für Situationen, in denen Forschende die Nullhypothese zwar zu Recht ablehnen, aber fälschlicherweise schlussfolgern, dass der Unterschied positiv ist, während er eigentlich negativ ist (oder umgekehrt).

Viele MethodikerInnen und StatistikerInnen (aber längst nicht alle!) halten das Paradigma des Signifikanztests und insbesondere der Fehler der ersten und zweiten Art für überholt. Auf seinem Blog bringt Andrew Gelman dies auf den Punkt (`http://andrewgelman.com/2004/12/29/type_1_type_2_t/`):

> **Never a Type 1 or Type 2 error**
>
> I've never in my professional life made a Type I error or a Type II error. But I've made lots of errors. How can this be?
>
> A Type 1 error occurs only if the null hypothesis is true (typically if a certain parameter, or difference in parameters, equals zero). In the applications I've worked on, in social science and public health, I've never come across a null hypothesis that could actually be true, or a parameter that could actually be zero.
>
> A Type 2 error occurs only if I claim that the null hypothesis is true, and I would certainly not do that, given my statement above!

In unserem Beispiel, etwa, wäre es kaum vorstellbar, dass moderater Alkoholkonsum nicht den geringsten Effekt auf die Sprechgeschwindigkeit hätte. (Die Nullhypothese besagt ja, dass dieser Effekt gleich 0 ist; aber 0 heisst eben buchstäblich 0, also 0,000…) Die Fehler, die er dann aber sehr wohl gemacht hat, bezeichnet Gelman als *Type-S*- und *Type-M*-Fehler (Gelman & Carlin, 2014). Das S steht für *sign* (Vorzeichen); Typ-S-Fehler macht man, wenn man behauptet, ein Zusammenhang sei positiv, während er eigentlich negativ ist (oder umgekehrt.) Das M steht dann wieder für *magnitude* (Grössenordnung); Typ-M-Fehler macht man, wenn man behauptet, ein Zusammenhang sei klein, während er eigentlich gross ist (oder umgekehrt).

12.4 Randomisierungs- und Permutationstests in R

12.4.1 Gruppenunterschiede

Randomisierungstests trifft man in der Literatur zwar nur selten an, aber sie haben den Vorteil, dass sie wenig Annahmen machen. Angenommen haben wir bei den Berechnungen oben eigentlich nur, dass die Versuchspersonen nach dem Zufallsprinzip den Konditionen zugeteilt wurden. Wir haben dabei die Gruppenmittel verglichen, aber wir hätten auch zum Beispiel die Mediane miteinander vergleichen können.

In R können solche Tests mithilfe der `independence_test()`-Funktion aus dem `coin`-Package durchgeführt werden. Mit den folgenden Befehlen werden die Daten zunächst in ein dataframe gegossen und dann einem Permutationstest unterzogen:

```
Rates <- c(4.2, 3.8, 5.0,
           3.1, 3.4, 4.3)
Condition <- factor(c(rep("ohne Alkohol", 3),
                      rep("mit Alkohol", 3)))
df <- data.frame(Rates, Condition)
df
```

```
##   Rates    Condition
## 1   4.2 ohne Alkohol
## 2   3.8 ohne Alkohol
## 3   5.0 ohne Alkohol
## 4   3.1  mit Alkohol
## 5   3.4  mit Alkohol
## 6   4.3  mit Alkohol
```

```
library(coin)
independence_test(Rates ~ Condition, data = df,
                  distribution = exact())
```

```
##
##  Exact General Independence Test
##
## data:  Rates by
##   Condition (mit Alkohol, ohne Alkohol)
## Z = -1.3148, p-value = 0.3
## alternative hypothesis: two.sided
```

Mit der Parametereinstellung `exact()` für `distribution` wird eingestellt, dass ein exakter Permutationstest durchgeführt werden soll. Mit einem solchen Test wird das beobachtete Ergebnis mit jeder möglichen Permutation abgeglichen. Für grössere Datensätze ist die Anzahl möglicher Permutationen aber riesig, sodass diese kaum mehr zu berechnen sind. In solchen Fällen kann man `distribution = approximate()`; dann werden 'nur' 10'000 zufällige Permutationen generiert. Hier ein Beispiel mit den Daten

von Klein et al. (2014), die wir bereits in Kapitel 9 analysiert haben:

```
klein <- read_csv("Data/Klein2014_money_abington.csv")
independence_test(Sysjust ~ MoneyGroup, data = klein,
                  distribution = approximate())
```

```
## Error in xtrafo(object@x): data class "character" is not supported
```

Laut der Fehlermeldung wird die Datenklasse "character" nicht unterstützt. Die Lösung besteht darin, die Variable `MoneyGroup` entweder als Zahl oder als Faktor umzukodieren:

```
# Als Zahl:
klein$n.MoneyGroup <- ifelse(klein$MoneyGroup == "control", 0, 1)
independence_test(Sysjust ~ n.MoneyGroup, data = klein,
                  distribution = approximate())
```

```
##
##  Approximative General Independence Test
##
## data:  Sysjust by n.MoneyGroup
## Z = -0.030381, p-value = 0.9886
## alternative hypothesis: two.sided
```

```
# Als Faktor
klein$f.MoneyGroup <- as.factor(klein$MoneyGroup)
independence_test(Sysjust ~ f.MoneyGroup, data = klein,
                  distribution = approximate())
```

```
##
##  Approximative General Independence Test
##
## data:  Sysjust by
##   f.MoneyGroup (control, treatment)
## Z = 0.030381, p-value = 0.9866
## alternative hypothesis: two.sided
```

Manchmal unterscheiden sich die p-Werte leicht voneinander. Das liegt dann daran, dass eben jeweils 'nur' 10'000 Permutationen generiert wurden: Je nachdem, welche Permutationen generiert werden, ist der p-Wert ein anderer. An den Schlussfolgerungen ändert dies jedoch selten etwas.

Um die Mediane miteinander zu vergleichen, kann man entweder einige Parameter bei `independence_test()` einstellen (spezifischer: die y-Variable zu Rängen konvertieren) oder die Funktion `wilcox_test()` verwenden:

```
# Alkoholbeispiel
independence_test(Rates ~ Condition, data = df,
                  distribution = exact(), ytrafo = rank)
```

```
##
##  Exact General Independence Test
##
## data:  Rates by
##   Condition (mit Alkohol, ohne Alkohol)
## Z = -1.0911, p-value = 0.4
## alternative hypothesis: two.sided
```

```
wilcox_test(Rates ~ Condition, data = df, distribution = exact())
```

```
##
##  Exact Wilcoxon-Mann-Whitney Test
##
## data:  Rates by
##   Condition (mit Alkohol, ohne Alkohol)
## Z = -1.0911, p-value = 0.4
## alternative hypothesis: true mu is not equal to 0
```

```
# Money-Priming
independence_test(Sysjust ~ f.MoneyGroup, data = klein,
                  distribution = approximate(), ytrafo = rank)
```

```
##
##  Approximative General Independence Test
##
## data:  Sysjust by
##   f.MoneyGroup (control, treatment)
## Z = -0.27355, p-value = 0.7863
## alternative hypothesis: two.sided
```

```
wilcox_test(Sysjust ~ f.MoneyGroup, data = klein, distribution = approximate())
```

```
##
##  Approximative Wilcoxon-Mann-Whitney Test
##
## data:  Sysjust by
##   f.MoneyGroup (control, treatment)
## Z = -0.27355, p-value = 0.7819
## alternative hypothesis: true mu is not equal to 0
```

12.4.2 Andere Muster (z.B. Korrelationen)

Permutationstests kann man übrigens auch einsetzen, um p-Werte für andere Muster als Gruppenunterschiede zu berechnen. Auf Seite 106 haben Sie den Korrelationskoeffizienten für den Zusammenhang zwischen zwei Indikatoren für kognitive Kontrolle in einer Stichprobe von 34 Teilnehmenden berechnet (Daten von Poarch et al., 2019).

```
poarch <- read_csv("Data/poarch2018.csv")
cor_poarch <- cor(poarch$Flanker, poarch$Simon)
cor_poarch
```

```
## [1] 0.4623553
```

Das Vorgehen beim Berechnen von p-Werten ist immer gleich. Der p-Wert drückt aus, wie wahrscheinlich es denn wäre ein Muster zu beobachten, dass mindestens so extrem wäre wie das tatsächlich beobachtete Muster, *wenn die Nullhypothese stimmen würde*. Beim Berechnen eines p-Wertes für einen Korrelationskoeffizienten ist die Aufgabe also, die Wahrscheinlichkeit zu berechnen, den beobachteten Korrelationskoeffizienten oder einen noch stärkeren Korrelationskoeffizienten in der Stichprobe anzutreffen, wenn es gar keinen Zusammenhang zwischen den zwei Variablen gäbe.

Um diese Wahrscheinlichkeit zu berechnen, müssen wir zuerst die Verteilung der Korrelationskoeffizienten generieren, die wir in dieser Stichprobe feststellen würde, wenn die zwei Variablen (Flanker und Simon) unabhängig voneinander wären. Eine Möglichkeit, dies zu machen, besteht darin, die beobachteten Werte einer der Variablen zu permutieren (= durcheinander zu schmeissen), ohne die beobachteten Werte der anderen Variablen mitzupermutieren; welche Variable permutiert wird, macht übrigens nichts aus. Hierdurch wird der systematische Zusammenhang zwischen den zwei Variablen gebrochen.[5] Abbildung 12.4 zeigt den beobachteten Zusammenhang und zwei Zusammenhänge, die man antreffen könnte, wenn man die Simon-Variable zufällig durcheinander schmeisst. Der R-Code unten zeigt Ihnen, wie Sie diese Abbildung selber zeichnen können.

```
p1 <- ggplot(poarch,
             aes(x = Flanker,
                 y = Simon)) +
  geom_point(shape = 1) +
  ggtitle("Eigentlicher Datensatz")

# Wenn man bei sample() keine weiteren Parameter einstellt,
# wird der Input zufällig permutiert.
p2 <- ggplot(poarch,
             aes(x = Flanker,
                 y = sample(Simon))) +
  geom_point(shape = 1) +
  ggtitle("Eine mögliche Permutation")
```

[5] Noch ein kleines Beispiel: Stellen Sie sich vor, dass Sie drei Paare von Beobachtungen haben: $(1, 100)$, $(2, 200)$ und $(3, 300)$. Zwischen den zwei Werten pro Beobachtung gibt es eine perfekte Korrelation. Wenn nun die Werte der ersten Beobachtung permutiert werden, ohne die Werte der anderen Beobachtung mitzupermutieren, könnte man etwa diese Paare feststellen: $(2, 100)$, $(3, 200)$, $(1, 300)$. Oder auch: $(3, 100)$, $(2, 200)$, $(1, 300)$. Die Stärke der Korrelation bei jeder Permutation ist nun rein zufallsbedingt. Zum Beispiel ist es genau so wahrscheinlich, eine negative Korrelation anzutreffen als eine positive.

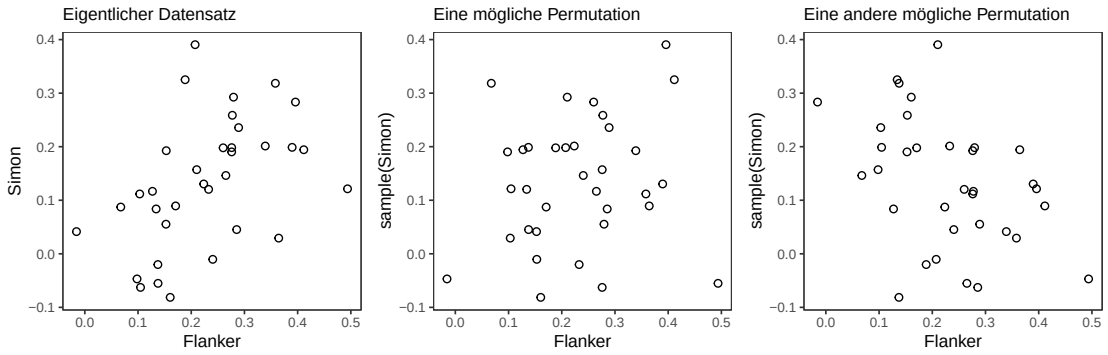


Abbildung 12.4: Links: Der festgestellte Zusammenhang zwischen den Variablen Flanker und Simon in der Studie von Poarch et al. (2018). Mitte und rechts: Indem die eine Variable (hier: Simon) unabhängig von der anderen permutiert wird, wird der systematische Zusammenhang zwischen beiden Variablen gebrochen. Dies entspricht der Nullhypothese. Um zu wissen, wie die Korrelationskoeffizienten laut der Nullhypothese verteilt sind, kann man diese Permutierung ein paar tausend Mal vornehmen und jeweils die Korrelation zwischen den Variablen berechnen.

```
p3 <- ggplot(poarch,
             aes(x = Flanker,
                 y = sample(Simon))) +
  geom_point(shape = 1) +
  ggtitle("Eine andere mögliche Permutation")

# Mit grid.arrange() aus dem Package gridExtra,
# können Sie mehrere Grafiken in einer Abbildung zeichnen.
gridExtra::grid.arrange(p1, p2, p3, ncol = 3)
```

Bei 34 Beobachtungen gibt es fast 300 Sextillionen (eine 3 mit 38 Nullen) mögliche Permutationen der Simon-Variablen.[6] Es ist unmöglich, für all diese den Korrelationskoeffizienten zu berechnen. Daher begnügen wir uns hier mit 20'000 Permutationen:

```
n_runs <- 20000
r_h0 <- vector(length = n_runs)

for (i in 1:n_runs) {
  r_h0[i] <- cor(poarch$Flanker, sample(poarch$Simon))
}
```

Die Verteilung der Korrelationskoeffizienten unter der Nullhypothese können Sie in einem Histogramm darstellen; Abbildung 12.5 zeigt eine etwas ausführlichere Variante.

```
df_r_h0 <- data.frame(r_h0)
ggplot(df_r_h0,
       aes(r_h0)) +
```

[6] $34 \times 33 \times 32 \times \cdots \times 3 \times 2 \times 1$

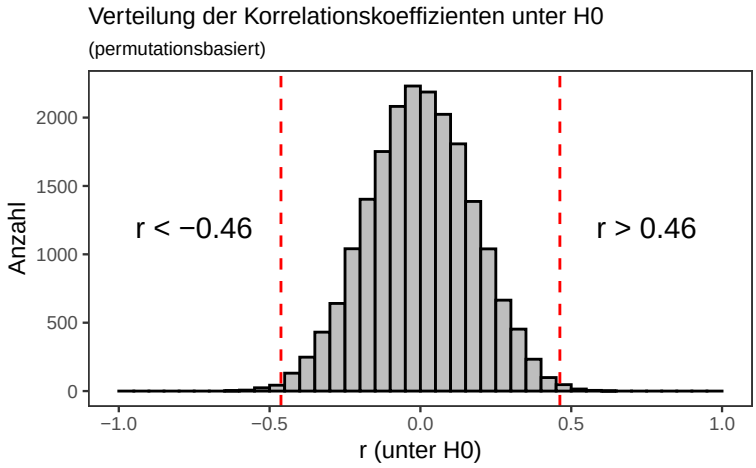


Abbildung 12.5: Die Verteilung der Korrelationskoeffizienten für diese Stichprobe mit 34 Beobachtungen, wenn man eine Variable zufällig permutiert.

```
  geom_histogram(fill = "grey", colour = "black",
                 breaks = seq(-1, 1, 0.05)) +
  geom_vline(xintercept = -cor_poarch,
             linetype = 2, colour = "red") +
  geom_vline(xintercept = cor_poarch,
             linetype = 2, colour = "red") +
  ggtitle("Verteilung der Korrelationskoeffizienten unter H0",
          subtitle = "(permutationsbasiert)") +
  xlab("r (unter H0)") +
  ylab("Anzahl") +
  annotate("text", x = -0.75, y = 1200, label = "r < -0.46") +
  annotate("text", x = 0.75, y = 1200, label = "r > 0.46")
```

Aus dieser Verteilung kann man ablesen, wie ungewöhnlich Korrelationskoeffizienten von $r = 0.46$ oder noch stärkere Korrelationskoeffizienten wären, wenn es keinen systematischen Zusammenhang zwischen den zwei Variablen gibt. Für den einseitigen Test (H_A : Die Korrelation ist positiv.) schaut man sich an, welche Proportion der unter der H_0 generierten Korrelationskoeffizienten gleich 0.46 oder grösser sind:

```
# Einseitiger p-Wert
mean(r_h0 >= cor_poarch)
```

```
## [1] 0.00245
```

Also 0.25% ($p = 0.0025$). Wenn Sie diese Berechnungen nochmals selber ausführen, werden Sie ein leicht anderes Ergebnis feststellen. Das liegt daran, dass die 20'000 Permutationen bei Ihnen dann eben nicht die gleichen wie bei mir sind.

Für den zweiseitigen Test (H_A : Die Korrelation ist nicht gleich 0, aber könnte sowohl positiv als auch negativ sein.) muss man sich zudem auch noch anschauen, welche

Proportion der unter der H_0 generierten Korrelationskoeffizienten gleich -0.46 oder kleiner ist:

```
# Zweiseitiger p-Wert
mean(r_h0 >= cor_poarch) + mean(r_h0 <= -cor_poarch)
```

```
## [1] 0.0056
```

Also 0.56% ($p = 0.0056$). Die Wahrscheinlichkeit, eine solche starke Korrelation oder eine noch stärkere anzutreffen, wenn die Nullhypothese tatsächlich stimmen würde, ist also recht klein.

Mit der `independence_test()`-Funktion wird dieses Vorgehen erleichert (aber wichtiger ist eben vor allem die Logik hinter dem Vorgehen):

```
independence_test(Simon ~ Flanker, data = poarch,
                  distribution = approximate())
```

```
##
##  Approximative General Independence Test
##
## data:  Simon by Flanker
## Z = 2.656, p-value = 0.0072
## alternative hypothesis: two.sided
```

Das Ergebnis fällt hier leicht anders aus als bei unseren manuellen Berechnungen, aber das liegt lediglich an der Zufallsauswahl der Permutationen. An den Schlussfolgerungen ("Dieser Zusammenhang wäre unter der Nullhypothese sehr unwahrscheinlich.") ändert sich ja nichts.

12.5 Geläufige Signifikanztests als mathematische Kürzel: der t-Test

Obwohl Permutationstests gültige Signifikanztests sind, die ausserdem auf nur wenigen Annahmen basieren, trifft man sie in der Praxis selten an. Dies ist zum Teil historischen Gründen zuzuschreiben: Im Jahr Schnee war es zu aufwändig, ein paar tausend Permutationen der Daten durchzurechnen, um die Verteilung eines Gruppenunterschieds oder eines Korrelationskoeffizienten unter der Nullhypothese zu generieren. Stattdessen wurden Signifikanztests entwickelt, deren mathematische Herleitung zwar komplizierter ist, die aber schneller ausgeführt werden können. Beispiele solcher Tests sind der t-Test und der F-Test. Dass diese so oft verwendet werden, verdanken sie der Tatsache, dass ihre Ergebnisse oft mit jenen der Permutationstests übereinstimmen:

> the statistician does not carry out this very simple and very tedious process [= Daten permutieren], but his conclusions have no justification beyond the fact that they agree with those which could have been arrived at by this

elementary method. (Fisher, 1936)

Der meist verbreitete dieser mathematischen Tricks, der t-Test, wird hier vorgestellt, sodass wir ihn in den Aufgaben verwenden können.

12.5.1 Der t-Wert

Mit den folgenden Befehlen werden zwei Stichproben (Gruppe A und Gruppe B) mit 4 bzw. 2 Beobachtungen aus Normalverteilungen generiert. Die Normalverteilungen sind einander gleich, denn sie haben das gleiche Mittel und die gleiche Standardabweichung.

```
gruppe <- rep(c("Gruppe A", "Gruppe B"), times = c(4, 2))
ergebnis <- c(rnorm(n = 4, mean = 3, sd = 1),
              rnorm(n = 2, mean = 3, sd = 1))
```

Diese Daten können wir wie gehabt in einem linearen Modell modellieren. Hier ist das zwar überflüssig, da wir wissen, dass es eigentlich keinen systematischen Unterschied zwischen den Gruppen gibt (sie stammen ja aus der gleichen Verteilung), aber so kann das Vorgehen besser illustriert werden.

```
# Modellieren
mod.lm <- lm(ergebnis ~ gruppe)

# Geschätzte Koeffizienten usw. anzeigen
summary(mod.lm)$coefficients
```

```
##                 Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)     3.8419135  0.2942221 13.057867 0.00019855
## gruppeGruppe B -0.5151547  0.5096077 -1.010885 0.36925320
```

Aufgrund des Zufalls wird der Unterschied zwischen den zwei Stichproben natürlich nie genau gleich 0 sein. Hier beträgt er -0.52; bei Ihnen wird er anders aussehen (zufällig generierte Daten). Die erste Information in diesem Output, die wir bisher immer ignoriert haben, sind die Werte in der Spalte `t value`. Es handelt sich hierbei lediglich um das Verhältnis des Unterschieds zwischen der Parameterschätzung und der Parameter laut der Nullhypothese (meistens 0) und des geschätzten Standardfehlers dieses Unterschieds:

$$t = \frac{\text{Parameterschätzung} - \text{Parameter laut } H_0}{\text{geschätzter Standardfehler}} \tag{12.1}$$

Der Output der `summary()`-Funktion zeigt stets den t-Wert unter der Annahme, dass der Parameter laut der Nullhypothese 0 beträgt.

```
# 2. Zeile, 1. Spalte: Estimate
summary(mod.lm)$coefficients[2, 1]
```

```
## [1] -0.5151547

# 2. Zeile, 2. Spalte: Std. Error
summary(mod.lm)$coefficients[2, 2]

## [1] 0.5096077

# Ratio der beiden
summary(mod.lm)$coefficients[2, 1] / summary(mod.lm)$coefficients[2, 2]

## [1] -1.010885

# = t-value (2. Zeile, 3. Spalte)
summary(mod.lm)$coefficients[2, 3]

## [1] -1.010885
```

12.5.2 Die t-Verteilungen

Wieso berechnet man solche t-Werte überhaupt? Der Grund ist, dass—unter bestimmten Annahmen—diese t-Werte in eine bestimmte Verteilung (eine t-Verteilung) fallen, wenn die Nullhypothese stimmt. Diese Verteilung hängt nicht von der Skala und Variabilität der Daten ab. In der Prä-Computerära (und jetzt noch immer) vereinfachte dies die Berechnungen erheblich.

Dass t-Werte unter der Nullhypothese einer t-Verteilung folgen, kann mit einer Simulation gezeigt werden. Mit den folgenden Befehlen werden—ähnlich wie oben—Daten für zwei Gruppen (mit 4 bzw. 2 Beobachtungen) aus der gleichen Normalverteilung generiert, und zwar 20'000 Mal. Jedes Mal wird der t-Wert für den Gruppenunterschied gespeichert.

```
n_runs <- 20000
n_gruppeA <- 4
n_gruppeB <- 2
t_werte <- vector(length = n_runs)

for (i in 1:n_runs) {
  sim_gruppe <- rep(c("Gruppe A", "Gruppe B"),
                    times = c(n_gruppeA, n_gruppeB))
  sim_ergebnis <- c(rnorm(n = n_gruppeA, mean = 3, sd = 1),
                    rnorm(n = n_gruppeB, mean = 3, sd = 1))
  sim_mod.lm <- lm(sim_ergebnis ~ sim_gruppe)
  t_werte[i] <- summary(sim_mod.lm)$coefficients[2, 3]
}
```

Wenn wir die 20'000 t-Werte grafisch darstellen, sehen wir, dass die Verteilung dieser Werte einer t-Verteilung mit 5 Freiheitsgraden entspricht; siehe Abbildung 12.6. (Die

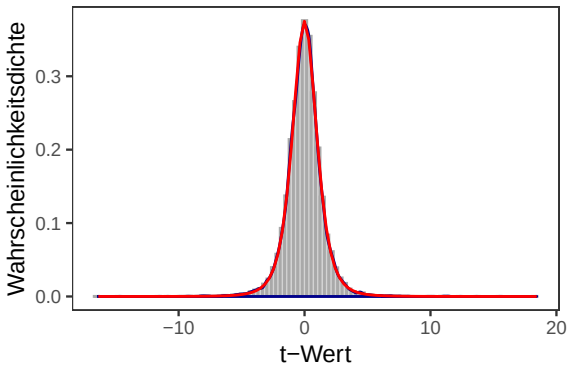


Abbildung 12.6: Die Verteilung von 20'000 t-Werten aus einer Simulation. Die blaue Linie (kaum sichtbar, da die rote mit ihr überlappt) stellt die auf der Basis der Simulationen geschätzte Wahrscheinlichkeitsdichte dar; die rote Linie stellt eine t-Verteilung mit 4 Freiheitsgraden dar.

Anzahl Freiheitsgrade ist, bei unabhängigen Daten, die Anzahl Beobachtungen minus die Anzahl geschätzter Parameter. Es gab 6 Beobachtungen und zwei geschätzte Parameter (`(Intercept)` und `gruppeGruppe B`), sodass die Anzahl Freiheitsgrade hier 4 beträgt.)

```
df_t <- data.frame(t_werte)
ggplot(df_t, aes(t_werte)) +
  # Histogramm der t-Werte
  geom_histogram(aes(y = ..density..), # Wahrscheinlichkeitsdichte zeichnen
                 bins = 100, fill = "lightgrey", colour = "darkgrey") +
  # geschätzte Wahrscheinlichkeitsdichte
  geom_density(colour = "darkblue") +
  # theoretische t-Verteilung mit 4 Freiheitsgraden
  stat_function(fun = dt, args = list(df = 4), colour = "red") +
  xlab("t-Wert") +
  ylab("Wahrscheinlichkeitsdichte")
```

Aufgabe: Ungleiche Varianzen Erhöhen Sie im Simulationscode die Standardabweichung einer Gruppe von 1 auf 10. Zeichnen Sie das Histogramm und die Wahrscheinlichkeitsdichten erneut; dazu können Sie die genau gleichen Befehle verwenden. Entspricht die Verteilung dieser t-Werte noch immer einer t-Verteilung mit 4 Freiheitsgraden?

12.5.3 Der t-Test selbst

Um nun die Frage zu beantworten, wie wahrscheinlich der beobachtete Gruppenunterschied oder noch extremere Gruppenunterschiede laut der Nullhypothese wären,

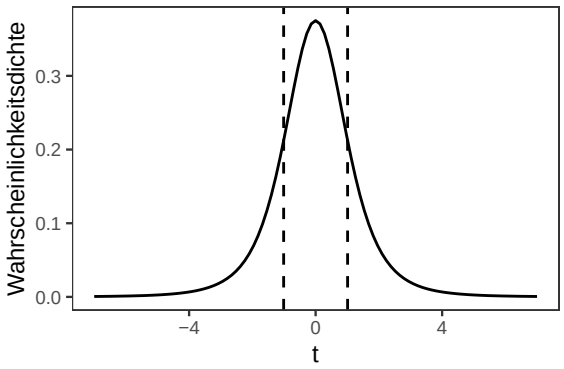


Abbildung 12.7: Die t-Verteilung mit 4 Freiheitsgraden. Die Strichellinien stellen den beobachteten Gruppenunterschied (positiv und negativ) dar.

können wir den beobachteten t-Wert mit der geeigneten t-Verteilung abgleichen (Abbildung 12.7).

```
ggplot(data = data.frame(t = c(-7, 7)),
       aes(x = t)) +
  stat_function(fun = dt, args = list(df = 4)) +
  geom_vline(xintercept = 1.010885, linetype = 2) +
  geom_vline(xintercept = -1.010885, linetype = 2) +
  ylab("Wahrscheinlichkeitsdichte")
```

Da es sich um eine mathematisch festgelegte Verteilung handelt, ist es für den Computer ein Kinderspiel, die Wahrscheinlichkeit von extremeren als dem beobachteten t-Wert zu berechnen (die Flächen unter der Kurve jenseits der Strichellinien):

```
pt(-1.010885, df = 4 + 2 - 2) +
  pt(1.010885, df = 4 + 2 - 2, lower.tail = FALSE)
```

```
## [1] 0.3692532
```

Dies ist auch die Wahrscheinlichkeit, die im Modelloutput der letzten Spalte zu entnehmen ist: $p = 0.37$.

Statt die Daten mit der `lm()`-Funktion zu analysieren, kann man sie auch der `t.test()`-Funktion füttern. Der Parameter `var.equal` wird hier auf `TRUE` gestellt, was soviel heisst, dass bei der Berechnung davon ausgegangen werden soll, dass die Gruppen laut der Nullhypothese aus Populationen mit der gleichen Varianz stammen; dies entspricht der Homoskedastizitätsannahme des allgemeinen linearen Modells:[7]

```
t.test(ergebnis ~ gruppe, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
```

[7] Wenn Sie mit eigenen Datensätzen arbeiten, müssen Sie noch den `data`-Parameter einstellen.

```
## data:  ergebnis by gruppe
## t = 1.0109, df = 4, p-value = 0.3693
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8997431  1.9300525
## sample estimates:
## mean in group Gruppe A mean in group Gruppe B
##               3.841913               3.326759
```

Die Ergebnisse sind denen des linearen Modells gleich, und alle Informationen in diesem Output kann man auch dem linearen Modell entnehmen. Zusammenfassen würde man das Ergebnis des Testes etwa so: "$t(4) = 1.01$, $p = 0.37$".

Ein t-Test ist ein lineares Modell. Und zwar eins, in dem nur ein Gruppenunterschied modelliert wird.

Wenn man nicht davon ausgehen will, dass die Varianzen in den beiden Gruppen gleich sind, kann man `var.equal = FALSE` (die Defaulteinstellung) einstellen. Dann wird der sog. t-Test nach Welch durchgeführt, in dem der t-Wert und die Anzahl Freiheitsgrade leicht anders berechnet werden:

```
t.test(ergebnis ~ gruppe, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  ergebnis by gruppe
## t = 1.346, df = 3.9996, p-value = 0.2495
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.547529  1.577838
## sample estimates:
## mean in group Gruppe A mean in group Gruppe B
##               3.841913               3.326759
```

Ruxton (2006) empfiehlt, dass der t-Test nach Welch immer dann durchgeführt werden soll, wenn man einen t-Test durchführen möchte.

12.5.4 Annahmen

Die Annahmen beim t-Test sind die gleichen wie die Annahmen, die man macht, wenn man für die Parameterschätzungen in einem allgemeinen linearen Modell Konfidenzintervalle anhand von t-Verteilungen konstruieren will. Beim t-Test nach Welch wird lediglich die Homoskedastizitätsannahme aufgehoben. Im Vergleich dazu wird einem

Randomisierungstest davon ausgegangen, dass die Datenpunkte unabhängig voneinander sind (eine Annahme, die der Test mit dem allgemeinen linearen Modell teilt) und dass die Datenpunkte unter der Nullhypothese genauso gut in der jeweils anderen Gruppe hätten vorkommen können. Letzteres ist eine weniger strikte Annahme als die Normalitätsannahme.

12.5.5 t-Tests für andere Parameterschätzungen

Die t-Verteilungen können auch eingesetzt werden, um die Nullhypothese bei Parameterschätzungen, die sich nicht auf Gruppenmittel beziehen, zu testen. In den `summary()`-Outputs in den Kapiteln in Teil III finden Sie hierfür viele Beispiele:

- Seite 121: Die Nullhypothesen, die überprüft werden, besagen, dass die Parameter für `(Intercept)` und `AOA` eigentlich gleich 0 sind. Bemerken Sie, dass diese erste Nullhypothese für niemanden von Interesse ist, da man sich kaum fürs Intercept interessiert und da niemand behaupten würde, er dürfte gleich 0 sein. Auch die zweite Nullhypothese kann kaum stimmen, denn sie besagt, dass das `AOA` *überhaupt keinen* Zusammenhang zum `GJT` aufweist.
- Seite 137: Die überprüften Nullhypothesen besagen wiederum, dass die Parameter für `(Intercept)` und `n.Kondition` eigentlich gleich 0 sind. Die t- und p-Werte für `n.Kondition` könnte man auch mit der `t.test()`-Funktion berechnen.
- Seite 144: Wiederum besagen die überprüften Nullhypothesen, dass die Parameter eigentlich gleich 0 sind.
- Seite 155: Idem. Von Interesse wäre hier allenfalls der Signifikanztest für den Interaktionsparameter (`DraganMitCS`).
- Seite 166.
- Seite 174.

Zwar handelt es sich in all diesen Beispielen eigentlich um t-Tests, aber den Begriff t-Test reserviert man in der Regel für Gruppenunterschiede, die man dann eben auch mit der `t.test()`-Funktion überprüfen kann.

12.6 Aufgaben

1. Schauen Sie sich die Grafiken in Abbildung 12.8 an. Wenn Sie eine Interventionsstudie durchführen, in der beide Gruppen identisch behandelt werden (eine Interventionsstudie ohne Intervention), muss die Nullhypothese stimmen. Wenn Sie die Daten trotzdem analysieren würden, aus welcher der vier Verteilungen wurde Ihr p-Wert dann stammen?

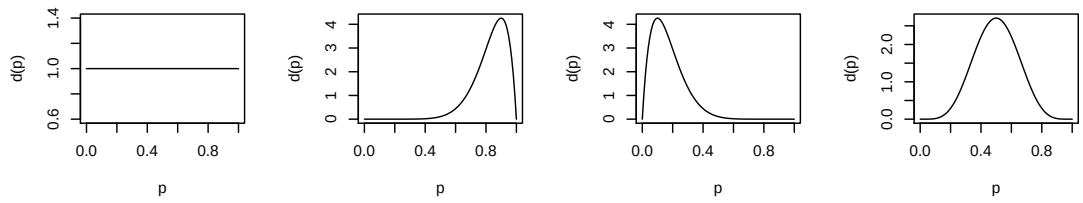


Abbildung 12.8: Aus welcher Verteilung stammt der p-Wert, wenn die Nullhypothese tatsächlich stimmt? *Links:* Gleichverteilung – alle p-Werte zwischen 0 und 1 wären gleich wahrscheinlich. *Mitte links:* Linksschiefe Verteilung – hohe p-Werte wären wahrscheinlicher als kleine p-Werte. *Mitte rechts:* Rechtsschiefe Verteilung – kleine p-Werte wären wahrscheinlicher als hohe p-Werte. *Rechts:* Glockenkurve – p-Werte um 0.5 herum wären am wahrscheinlichsten.

2. Ändern Sie den Simulationscode aus dem letzten Abschnitt (Seite 208), um Ihre Antwort zu überprüfen.
3. Aus welcher Art von Verteilung würde der p-Wert stammen, wenn die Nullhypothese nicht stimmt?
4. Ändern Sie den Simulationscode aus dem letzten Abschnitt, um Ihre Antwort zu überprüfen.

Kapitel 13

Varianzanalyse

13.1 Mehrere Gruppen vergleichen: Das Problem

Wenn wir statt zwei *mehrere* Gruppen mithilfe von Signifikanztests hinsichtlich eines outcomes vergleichen möchten, läge es auf der Hand, mehrere t-Tests einzusetzen. Wenn man zum Beispiel drei Gruppen vergleichen möchte, könnte man zuerst Gruppen A und B vergleichen, dann Gruppen A und C und zu guter Letzt Gruppen B und C. Das Problem mit diesem Vorgehen ist, dass die Wahrscheinlichkeit, *irgendeinen* signifikanten Unterschied zu finden, wenn die Nullhypothese tatsächlich stimmt, grösser als α (in der Regel: $\alpha = 0.05$) ist. Die Simulationen in diesem Abschnitt stellen dieses Problem unter Beweis; danach folgen mögliche Lösungen.

Generieren wir zuerst einen Datensatz mit Zufallsdaten, sodass wir wissen, dass die Nullhypothese buchstäblich stimmt. Mit dem folgenden Befehl wird ein Datensatz namens `df` kreiert. Dieser enthält 60 Beobachtungen von je zwei Variablen: `gruppe` (drei Ausprägungen mit je 20 Beobachtungen) und `ergebnis`. Letztere Variable wurde aus einer Normalverteilung generiert, deren Eigenschaften unabhängig von `gruppe` sind. Die Nullhypothese stimmt also. Abbildung 13.1 stellt diese Daten grafisch dar.

```
df <- data.frame(
  gruppe = rep(c("A", "B", "C"), times = 20),
  ergebnis = rnorm(n = 3*20, mean = 10, sd = 3)
)
```

Wir könnten nun diesen Datensatz drei Mal aufspalten: Ein Mal behalten wir nur die Daten aus den Gruppen A und B, ein Mal behalten wir nur die Daten aus Gruppen A und C, und ein Mal nur die Daten aus Gruppen B und C. Wir führen dann jedes Mal einen t-Test aus. (Hier wird der t-Test nach Welch verwendet, aber dies spielt keine Rolle.) Bei Ihnen werden der Output anders aussehen, da die Daten zufällig generiert wurden.

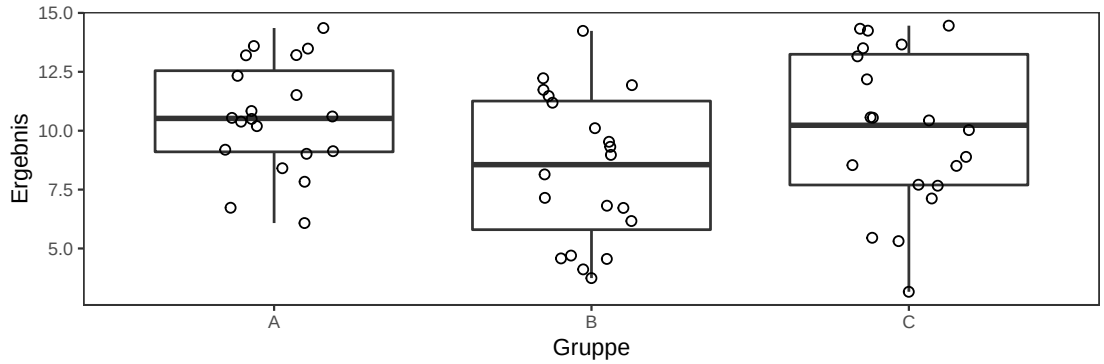


Abbildung 13.1: Zufällig generierte Daten. Bei Ihnen werden diese Daten natürlich anders aussehen.

```
# A vs. B
df_ab <- df %>%
  filter(gruppe %in% c("A", "B"))
t.test(ergebnis ~ gruppe, data = df_ab)

##
##  Welch Two Sample t-test
##
## data:  ergebnis by gruppe
## t = 2.4881, df = 34.972, p-value = 0.01775
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.402551 3.971529
## sample estimates:
## mean in group A mean in group B
##       10.557221        8.370181
```

```
# A vs. C
df_ac <- df %>%
  filter(gruppe %in% c("A", "C"))
t.test(ergebnis ~ gruppe, data = df_ac)

##
##  Welch Two Sample t-test
##
## data:  ergebnis by gruppe
## t = 0.64088, df = 33.998, p-value = 0.5259
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.267666  2.435479
## sample estimates:
## mean in group A mean in group C
```

```
##       10.557221        9.973315
```

```
# B vs. C
df_bc <- df %>%
  filter(gruppe %in% c("B", "C"))
t.test(ergebnis ~ gruppe, data = df_bc)
```

```
##
##  Welch Two Sample t-test
##
## data:  ergebnis by gruppe
## t = -1.559, df = 37.888, p-value = 0.1273
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.6850506  0.4787827
## sample estimates:
## mean in group B mean in group C
##        8.370181        9.973315
```

Wenn man sich in die Logik des Signifikanztestens einschreibt, dann müsste man aufgrund dieser Ergebnisse die Nullhypothese, dass die Daten in jeder Gruppe aus der gleichen Verteilung stammen, ablehnen, denn für den Vergleich zwischen Gruppen A und B findet man ja einen p-Wert, der kleiner als $\alpha = 0.05$ ist ($p = 0.018$). Dies obwohl die Nullhypothese in dieser Simulation buchstäblich stimmt. Dass man Nullhypothesen, die tatsächlich stimmen, ab und zu zu Unrecht ablehnt, ist klar—und das ist hier auch nicht das Problem. Vielmehr ist das Problem, dass diese Fehlerquote (Fehler der ersten Art) angeblich bei $\alpha = 0.05$ festgelegt wurde, aber eigentlich wesentlich höher ist.

Diese Tatsache können wir mit einer Simulation illustrieren. Mit den folgenden Befehlen wird das gleiche Szenario wie oben (3 Gruppen mit je 20 Beobachtungen; Nullhypothese stimmt; 3 t-Tests) 5'000 durchlaufen. Jedes Mal wird der p-Wert der einzelnen t-Tests gespeichert sowie auch der kleinste der jeweils drei p-Werte.

```
n_runs <- 5000
pval_ab <- vector(length = n_runs)
pval_ac <- vector(length = n_runs)
pval_bc <- vector(length = n_runs)
min_pval <- vector(length = n_runs)

for (i in 1:n_runs) {
  sim_df <- data.frame(
    gruppe = rep(c("A", "B", "C"), times = 20),
    ergebnis = rnorm(n = 3*20, mean = 10, sd = 3)
  )
```

```
  # A vs. B
  # Datensatz aufspalten
  sim_df_ab <- sim_df %>%
    filter(gruppe %in% c("A", "B"))
  # t-Test ausführen und p-Wert speichern
  pval_ab[i] <- t.test(ergebnis ~ gruppe, data = sim_df_ab)$p.value

  # A vs. C
  sim_df_ac <- sim_df %>%
    filter(gruppe %in% c("A", "C"))
  pval_ac[i] <- t.test(ergebnis ~ gruppe, data = sim_df_ac)$p.value

  # B vs. C
  sim_df_bc <- sim_df %>%
    filter(gruppe %in% c("B", "C"))
  pval_bc[i] <- t.test(ergebnis ~ gruppe, data = sim_df_bc)$p.value

  # Kleinsten der 3 p-Werte speichern
  min_pval[i] <- min(c(pval_ab[i], pval_ac[i], pval_bc[i]))
}
```

Wenn die Nullhypothese stimmt, müssten die p-Werte gleichverteilt sein (siehe Aufgaben letztes Kapitel). Wenn Sie Histogramme der p-Werte der einzelnen t-Tests zeichnen, werden Sie feststellen, dass dies tatsächlich der Fall ist. (Die Abweichungen, die Sie feststellen werden, sind zufallsbedingt; wenn Sie die Anzahl Simulationen erhöhen, werden diese Abweichungen kleiner.) Wenn Sie nun aber die Verteilung der jeweils kleinsten der drei p-Werte (`min_pval`) zeichnen, werden Sie feststellen, dass diese *nicht* gleichverteilt sind; siehe Abbildung 13.2 auf Seite 219.

Wenn wir uns bei unseren Inferenzen darüber, ob sich die drei Gruppen voneinander unterscheiden, nach dem kleinsten der drei p-Werte richten, würden wir nicht in bloss $\alpha = 5\%$ der Fälle die Nullhypothese zu Unrecht ablehnen, sondern in mehr als 10% der Fälle, wenn drei Gruppen verglichen werden:

```
mean(min_pval < 0.05)
```

```
## [1] 0.1244
```

Familywise Type-I error rate. Wenn man mehrere Signifikanztests verwendet, um eine Nullhypothese zu überprüfen, steigt die Wahrscheinlichkeit, dass mindestens einer von ihnen einen signifikanten p-Wert produziert—auch wenn die Nullhypothese stimmt und jeder Test korrekt ausgeführt wurde. Die Wahrscheinlichkeit, dass man mindestens einen signifikanten p-Wert antrifft, wenn die Nullhypothese stimmt, nennt man die *familywise Type-I error rate*; siehe Abbildung 13.3 auf der nächsten Seite.

In Fällen wie im obigen Beispiel ist die Tatsache, dass man eine Nullhypothese mit mehreren Tests überprüft hat (*multiple comparisons*), allen klar. Ausserdem kann das Problem in solchen Fällen relativ leicht behoben werden. In anderen Fällen sind die *multiple comparisons* weniger einfach aufzudecken bzw. schwieriger bei der Interpretation der Ergebnisse zu berücksichtigen – zum Beispiel, weil im Forschungsbericht nur eine Auswahl der ausgeführten Signifikanztests beschrieben wird.

13.2 Erste Lösung: Permutationstest

Die einfachste Art und Weise, das *multiple comparisons*-Problem zu lösen, ist, die Nullhypothese mit *einem einzigen* Test zu überprüfen statt mit mehreren Tests. Am häufigsten wird hierzu Varianzanalyse (ANOVA: *analysis of variance*) bzw. der F-Test eingesetzt (siehe nächsten Abschnitt). Aber das Gleiche kann man bewirken mit einem Permutationstest, der m.E. einfacher nachvollziehbar ist.

Die zu überprüfende Nullhypothese besagt, dass die Beobachtungen in den drei Gruppen allen aus der gleichen Verteilung stammen und somit eigentlich alle das gleiche Mittel haben. In einer Stichprobe werden wir natürlich immer feststellen, dass sich diese Mittel zumindest etwas voneinander unterscheiden:

```
df %>%
  group_by(gruppe) %>%
  summarise(mittel = mean(ergebnis))
```

```
## # A tibble: 3 x 2
##   gruppe mittel
##   <fct>   <dbl>
## 1 A       10.6
## 2 B        8.37
## 3 C        9.97
```

Die entscheidende Frage ist daher: Wie wahrscheinlich wäre es, dass sich die Mittel so stark oder noch stärker voneinander unterscheiden würden, wenn die Nullhypothese tatsächlich stimmt? Um diese Frage zu beantworten, müssen wir zuerst in einer Zahl ausdrücken, wie stark sich diese drei Mittel voneinander unterscheiden. Die Varianz der Gruppenmittel bietet sich hierfür an.

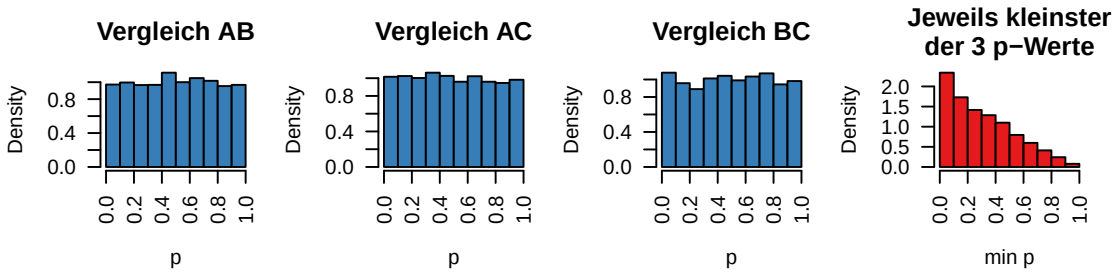


Abbildung 13.2: Wenn die Nullhypothese stimmt, sollten die p-Werte gleichverteilt sein. Für die p-Werte aus den einzelnen t-Tests ist dies auch der Fall: Die Wahrscheinlichkeit, dass man einen p-Wert kleiner als 0.05 beobachtet, liegt tatsächlich bei 5%, wenn die Nullhypothese stimmt. Wenn man aber auf der Basis des jeweils kleinsten der drei p-Werte schliesst, ob sich die Gruppen unterscheiden, dann sind die relevanten p-Werte nicht gleich- sondern rechtsschief verteilt: Die Wahrscheinlichkeit, dass man *irgendeinen* p-Wert kleiner als 0.05 beobachtet, kann erheblich höher als 5% sein, auch wenn die Nullhypothese stimmt.

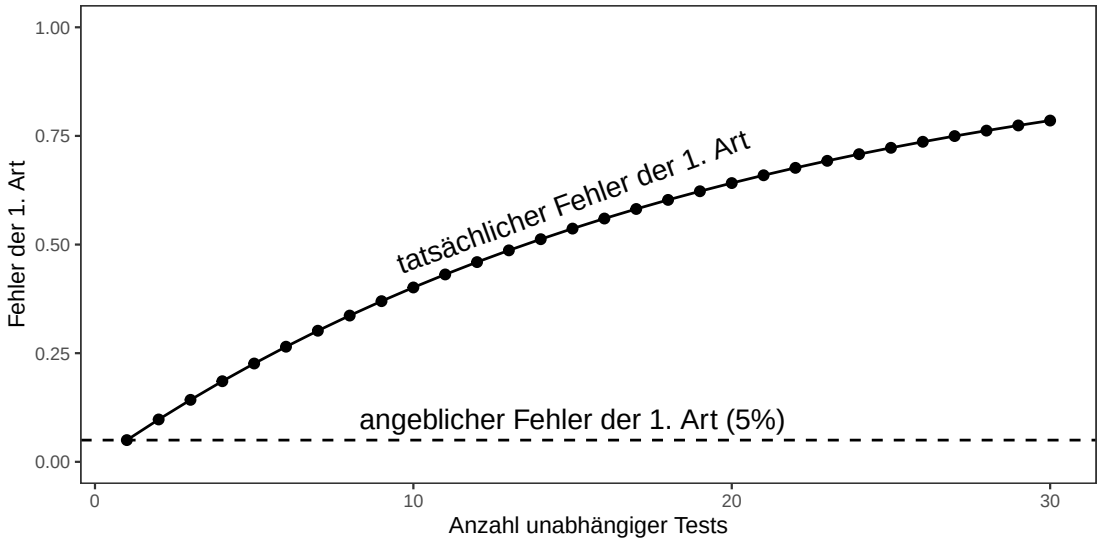


Abbildung 13.3: Wenn alle Nullhypothesen stimmen und mehrere unabhängige Signifikanztests durchgeführt werden, dann wird die Wahrscheinlichkeit, dass mindestens ein Test ein signifikantes Ergebnis produziert immer grösser ($y = 1 - (1 - 0.05)^{\text{Anzahl Tests}}$). Diese Wahrscheinlichkeit ist der *familywise Type-I error rate*. Zwei Tests sind unabhängig voneinander, wenn das Ergebnis des einen Tests überhaupt keine Indizien darüber gibt, was das Ergebnis des anderen Tests sein wird—z.B., weil komplett andere Daten benutzt werden. Wenn die Tests nicht unabhängig voneinander sind (z.B., weil man die gleichen Daten mit mehreren Tests auswertet), wird dieses Problem zwar immer noch vorhanden sein, aber es wird weniger ausgeprägt sein.

```
var_mittel <- df %>%
  group_by(gruppe) %>%
  summarise(mittel = mean(ergebnis)) %>%
  select(mittel) %>%
  var()
var_mittel
```

```
##          mittel
## mittel 1.282355
```

Die Varianz der Stichprobenmittel beträgt also 1.282. Bei Ihnen wird das Ergebnis aufgrund der zufällig generierten Daten anders aussehen. Um die Verteilung der Varianzen der Stichprobenmittel unter Annahme der Nullhypothese zu generieren, können wir die Beobachtungen ein paar tausend Mal zufällig den Gruppenbezeichnungen zuordnen und jedes Mal die Varianz der Stichprobenmittel der permutierten Daten berechnen. Die Logik ist identisch mit jener der Permutationstests aus dem letzten Kapitel, nur schauen wir uns die Varianz der Gruppenmittel statt des Unterschieds zwischen den zwei Gruppenmitteln an.

```
# Sie können diese Anzahl verkleinern,
# sodass die Berechnung schneller geht.
n_runs <- 10000
var_mittel_H0 <- vector(length = n_runs)

# Der Datensatz df wird hier als df_H0 kopiert,
# sodass er nachher nicht überschrieben wird.
df_H0 <- df

for (i in 1:n_runs) {
  df_H0$gruppe <- sample(df_H0$gruppe)

  var_mittel_H0[i] <- df_H0 %>%
    group_by(gruppe) %>%
    summarise(mittel = mean(ergebnis)) %>%
    select(mittel) %>%
    var()
}
```

Die Verteilung dieser Varianzen (`var_mittel_H0`) wird in Abbildung 13.4 dargestellt. Die Proportion der Varianzen, die mindestens so gross wie die beobachtete Varianz sind, stellt ein gültiger p-Wert dar.

```
mean(var_mittel_H0 > 1.282355)
```

```
## [1] 0.0638
```

In diesem Fall also $p = 0.068$. Von Durchführung zu Durchführung wird sich dieser

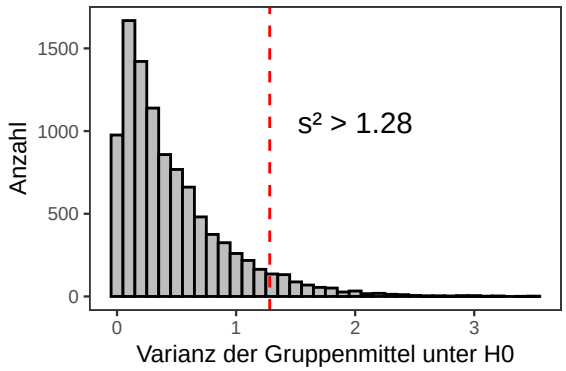


Abbildung 13.4: Die Verteilung der Varianzen der Stichprobenmittel, wenn die Beobachtungen zufällig den Gruppen zugeordnet werden. Die Proportion der Varianzen, die mindestens so gross wie die beobachtete Varianz sind, stellt ein gültiger p-Wert dar.

Wert leicht ändern, da er auf 'nur' 10'000 der fast 578 Quadrillionen möglicher Permutationen basiert. Aber die Unterschiede werden minimal sein.

Aufgabe 1: Standardabweichung Wir haben die Unterschiede zwischen den Gruppenmitteln mit dem Varianzmass erfasst. Würde sich am Ergebnis etwas ändern, wenn wir stattdessen die Standardabweichung benutzt hätten?

Aufgabe 2: Echter Datensatz In Abschnitt 9.2 auf Seite 140 wurde ein echter Datensatz mit drei Gruppen vorgestellt. Überprüfen Sie die Nullhypothese, dass sich die Durchschnittsleistung zwischen den 'no information'-, 'information'- und 'strategy'-Konditionen nicht unterscheidet und zwar mit einem Permutationstest.

Permutationstests wie diese braucht man nicht selber zu programmieren. Mit der Funktion `independence_test()` aus dem `coin`-Package können sie schnell und einfach durchgeführt werden. Am Ergebnis ändert sich nichts—wenn Sie diese Funktion mehrmals laufen lassen, werden Sie feststellen, dass die kleineren Abweichungen dem Zufallsverfahren hinter dem Test zuzuweisen sind.

```
coin::independence_test(ergebnis ~ gruppe, data = df,
                        distribution = "approximate")

##
##  Approximative General Independence Test
##
## data:  ergebnis by gruppe (A, B, C)
## maxT = 2.2527, p-value = 0.0613
## alternative hypothesis: two.sided
```

13.3 Zweite Lösung: Varianzanalyse und F-Tests

Permutationtests waren im Jahr Schnee zu schwierig, um durchzuführen. Wenn man aber annehmen will, dass die Daten aus einer Normalverteilung stammen, dann kann man ein paar mathematische Tricks anwenden, um die Berechnung zu beschleunigen. Diese Tricks heissen Varianzanalyse (ANOVA: *analysis of variance*) und der F-Test. Da Forschungsartikel oft voller ANOVAs und F-Tests stehen, werden diese Tricks hier kurz vorgestellt. Ich finde den Permutationtest aber wesentlich nachvollziehbarer.

13.3.1 Streuungszerlegung

Der erste Schritt in einer ANOVA besteht darin, dass man die Streuung im outcome in Teilen zerlegt: Streuung *zwischen* den Gruppen und Streuung *innerhalb* der Gruppen. Dazu berechnet man zuerst die Gesamtquadratsumme (vgl. Quadratsumme auf Seite 39). Bei Ihnen werden diese Zahlen anders aussehen, da die Daten zufällig generiert wurden.

```
# Gesamtquadratsumme
sq.total <- sum((df$ergebnis - mean(df$ergebnis))^2)
sq.total
```

```
## [1] 556.7215
```

Um die Streuung innerhalb der Gruppen, die Residuenquadratsumme, zu berechnen, kann man von allen Beobachtungen ihr Gruppenmittel abziehen. Oder man modelliert die Daten in einem allgemeinen linearen Modell und berechnet die Quadratsumme der Residuen:

```
df.lm <- lm(ergebnis ~ gruppe, data = df)

# Residuenquadratsumme
sq.rest <- sum((resid(df.lm))^2)
sq.rest
```

```
## [1] 505.4273
```

Die vom Modell 'erklärte' (lies: beschriebene) Quadratsumme beträgt also:

```
sq.gruppe <- sq.total - sq.rest
sq.gruppe
```

```
## [1] 51.2942
```

Wir haben, zusätzlich zum Intercept, zwei Parameter gebraucht, um den Einfluss des Gruppenfaktors zu modellieren. (Dies können Sie kontrollieren, indem Sie das Modell `df.lm` inspizieren.) Im Schnitt beschreibt jeder Parameter also eine Quadratsumme von 25.6:

```
meanSq.gruppe <- sq.gruppe / 2
meanSq.gruppe
```

```
## [1] 25.6471
```

Es gibt 60 unabhängige Datenpunkte und das Modell schätzt bereits 3 Parameter (Intercept + 2 Parameter für die Gruppenunterschiede). Daher könnten wir im Prinzip noch 57 Parameter schätzen lassen. (Dies wäre zwar nicht sinnvoll, aber es wäre möglich. Mehr Parameter zu schätzen, ginge nicht: In einem allgemeinen linearen Modell kann man nur so viele Parameter schätzen als es Beobachtungen gibt.) Im Schnitt würden diese 57 Parameter eine Quadratsumme von 8.87 beschreiben:

```
meanSq.rest <- sq.rest / (60 - 2 - 1)
meanSq.rest
```

```
## [1] 8.867146
```

Die kritische Einsicht ist nun, dass wenn die Nullhypothese stimmt, die zwei Gruppenparameter im Schnitt nur die gleiche Quadratsumme wie die 57 nicht-modellierte Parameter beschreiben können. Also: Wenn H_0 zutrifft, sollten `meanSq.gruppe` und `meanSq.rest` einander gleich sein. Eine andere Art und Weise, um dies auszudrücken, ist, dass das Verhältnis von `meanSq.gruppe` und `meanSq.rest` laut der Nullhypothese im Schnitt 1 ('E') sein soll:

$$H_0 : E\left(\frac{\texttt{meanSq.gruppe}}{\texttt{meanSq.rest}}\right) = 1$$

Dieses Verhältnis bezeichnet man als F. In unserem Fall beträgt F 2.89.

```
f.wert <- meanSq.gruppe / meanSq.rest
f.wert
```

```
## [1] 2.892374
```

13.3.2 Der F-Test

Aufgrund des Stichprobenfehlers wird F nie ganz genau 1 sein. Die nächste Frage ist daher: Wie ungewöhnlich ein F-Wert von 2.89 oder grösser wäre, wenn die Nullhypothese stimmt? Wenn die Daten in den unterschiedlichen Gruppen aus Normalverteilungen mit dem gleichen Mittel und der gleichen Varianz stammen, dann fällt der F-Wert in eine bestimmte Verteilung, nämlich eine F-Verteilung. F-Verteilungen haben zwei Parameter ('Freiheitsgrade'): die Anzahl Parameter, die man brauchte, um den Effekt zu modellieren (hier: 2), und die Anzahl Beobachtungen, die man nicht aufgebraucht hat (hier: 57). Abbildung 13.5 zeigt eine $F(2, 57)$-Verteilung.[1]

[1] Achtung: 2, 57 ist keine Kommazahl; es handelt sich um zwei separate Zahlen, die Freiheitsgrade.

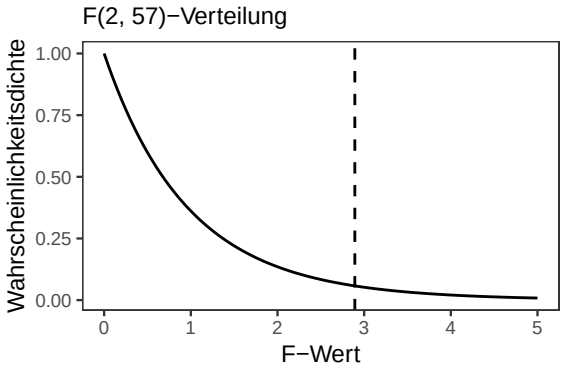


Abbildung 13.5: Die F-Verteilung mit 2 und 57 Freiheitsgraden. (F-Verteilungen haben zwei Freiheitsgrade.) Die Strichellinie stellt den beobachteten F-Wert dar.

Die Fläche unter der Kurve rechts von der Strichellinie ist die Wahrscheinlichkeit, dass man einen F-Wert von 2.89 oder mehr beobachtet, wenn die Nullhypothese tatsächlich stimmt. Sie kann so berechnet werden:

```
pf(f.wert, df = 2, df2 = 60 - 2 - 1, lower.tail = FALSE)
```

```
## [1] 0.06361948
```

Dies ist fast das gleiche Ergebnis wie beim Permutationstest.

13.3.3 Varianzanalyse in R

Glücklicherweise muss man all diese Schritte nicht von Hand ausführen. Es reicht, die Daten in einem allgemeinen linearen Modell zu modellieren (was oben bereits gemacht wurde) und die `anova()`-Funktion aufs Modell anzuwenden:

```
anova(df.lm)
```

```
## Analysis of Variance Table
##
## Response: ergebnis
##           Df Sum Sq Mean Sq F value  Pr(>F)
## gruppe     2  51.29 25.6471  2.8924 0.06362
## Residuals 57 505.43  8.8671
```

Berichten würde man dieses Ergebnis wie folgt: $F(2, 57) = 2.89$, $p = 0.064$.

Aufgabe 1: Echter Datensatz In Abschnitt 9.2 auf Seite 140 wurde ein echter Datensatz mit drei Gruppen vorgestellt. Überprüfen Sie die Nullhypothese, dass sich die Durchschnittsleistung zwischen den 'no information'-, 'information'- und 'strategy'-Konditionen nicht unterscheidet und zwar mit einer ANOVA.

Aufgabe 2: Zwei Gruppen In Abschnitt 9.1.3 auf Seite 133 wurde ein echter Datensatz mit zwei Gruppen vorgestellt. Diesen haben Sie im letzten Kapitel mit einem t-Test analysiert. Analysieren Sie ihn hier nochmals mit einem normalen t-Test (also nicht nach Welch) und mit einer ANOVA. Welchen Merksatz ziehen Sie?[2]

13.4 Varianzanalyse mit mehreren Prädiktoren

Varianzanalyse kann man auch anwenden, wenn es mehrere Prädiktoren in einem Modell gibt. Die Daten aus der Dragan/Luca-Studie kann man wie folgt in einer ANOVA auswerten:

```
b11 <- read_csv("Data/berthele2011.csv")
b11.lm1 <- lm(Potenzial ~ CS * Name, data = b11)
anova(b11.lm1)
```

```
## Analysis of Variance Table
##
## Response: Potenzial
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## CS          1   1.755  1.7550  2.5051 0.1155705
## Name        1   0.364  0.3644  0.5201 0.4718954
## CS:Name     1   7.965  7.9651 11.3695 0.0009484
## Residuals 151 105.786  0.7006
```

Jede Zeile in der Tabelle zeigt, wie viele Parameter für einen bestimmten Prädiktor modelliert werden mussten (`Df`) und was die assoziierte Quadratsumme und sein F- und p-Wert ist. Aber Vorsicht: Wenn wir die Reihenfolge der Prädiktoren ändern, ändern sich ein paar Zahlen:

```
b11.lm2 <- lm(Potenzial ~ Name * CS, data = b11)
anova(b11.lm2)
```

```
## Analysis of Variance Table
##
## Response: Potenzial
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Name        1   0.786  0.7855  1.1213 0.2913379
## CS          1   1.334  1.3339  1.9040 0.1696729
## Name:CS     1   7.965  7.9651 11.3695 0.0009484
## Residuals 151 105.786  0.7006
```

Der Grund hierfür ist, dass bei der Berechnung von `Sum Sq` sequenziell vorgegangen wird. Wenn wir die Varianzanalyse von Hand ausführen würden, würden wir:

1. die Gesamtquadratsumme berechnen;

[2] Tipp: Quadrieren Sie den t-Wert, den Sie beim t-Test erhalten.

```
(ss.gesamt <- sum((b11$Potenzial - mean(b11$Potenzial))^2))
```

```
## [1] 115.871
```

2. den Effekt der ersten Variablen rausrechnen und berechnen, wie gross die Quadratsumme ist, die diese beschreiben kann;

```
cs.lm <- lm(Potenzial ~ CS, data = b11)
(ss.cs <- ss.gesamt - sum(resid(cs.lm)^2))
```

```
## [1] 1.755004
```

3. den Effekt der zweiten Variablen rausrechnen und berechnen, wie gross die Quadratsumme ist, die diese beschreiben kann;

```
name.lm <- lm(Potenzial ~ CS + Name, data = b11)
(ss.name <- ss.gesamt - ss.cs - sum(resid(name.lm)^2))
```

```
## [1] 0.3643985
```

4. den Interaktionseffekt rausrechnen und berechnen, wie gross die Quadratsumme ist, die dieser beschreiben kann;

```
interaktion.lm <- lm(Potenzial ~ CS + Name + CS:Name, data = b11)
(ss.interaktion <- ss.gesamt - ss.cs - ss.name - sum(resid(interaktion.lm)^2))
```

```
## [1] 7.965143
```

5. die restliche Quadratsumme berechnen;

```
(ss.rest <- ss.gesamt - ss.cs - ss.name - ss.interaktion)
```

```
## [1] 105.7864
```

6. F-Werte für alle Effekte berechnen.

```
ss.cs / (ss.rest/151)
```

```
## [1] 2.5051
```

```
ss.name / (ss.rest/151)
```

```
## [1] 0.520144
```

```
ss.interaktion / (ss.rest/151)
```

```
## [1] 11.36948
```

Je nachdem, ob wir `CS` oder `Name` als erste Variable ins Modell eintragen, ändert sich die Quadratsumme, die dieser Variablen zugeschrieben wird. Der Grund dafür ist, dass die Variablen `CS` und `Name` in diesem Datensatz etwas miteinander korreliert sind: Es gibt mehr Datenpunkte für Dragan mit Codeswitches als ohne und mehr Datenpunkte für Luca ohne Codeswitches als ohne:

```
xtabs(~ CS + Name, data = b11)
```

```
##       Name
## CS     Dragan Luca
##   mit      28   44
##   ohne     51   32
```

Ein Teil der Streuung in den `Potenzial`-Werten kann daher nicht eindeutig dem einen oder dem anderen Prädiktor zugewiesen werden. Wird `CS` als erste Variable dem Modell hinzugefügt, dann wird all die Streuung, für die es nicht ganz klar ist, ob sie `CS` oder `Name` zu verdanken ist, der Variablen `CS` zugeschrieben; wird `Name` als erste Variable dem Modell hinzugefügt, wird all diese Streuung ihr zugeschrieben. Daher ändern sich die Ergebnisse für diese Variablen, je nachdem, welche Variable zuerst hinzugefügt wurde. Am Ergebnis für den letzten Effekt (die Interaktion) ändert sich jedoch nichts.

Type-I, Type-II, Type-III sum of squares. Wenn die Quadratsummen und die von ihnen abgeleiteten F- und p-Werte sequenziell berechnet werden (wie oben), spricht man von *Type-I sum of squares*. Diese Berechnungsart ist insbesondere dann sinnvoll, wenn man sich für den als letzten modellierten Effekt interessiert, aber zuerst noch ein paar andere Variablen berücksichtigt.

Eine alternative Berechnungsart wird in gehabter kreativer Weise *Type-II sum of squares* genannt. Hierzu werden die Effekte in allen möglichen Reihenfolgen ins Modell eingetragen (aber Interaktionen kommen immer nach Haupteffekten) und gilt als der F- und p-Wert eines Effektes jene F- und p-Werte, die man antrifft, wenn der Effekt als letzter eingetragen wurde. In unserem Beispiel sähe das Ergebnis wie folgt aus:

- `CS`: Man übernimmt die Angaben für `CS` aus der ANOVA-Tabelle für Modell `b11.lm2`, da hier `CS` als letzter Haupteffekt eingetragen wurde: $F(1, 151) = 1.9$, $p = 0.17$.
- `Name`: Man übernimmt die Angaben für `Name` aus der ANOVA-Tabelle für Modell `b11.lm1`, da hier `Name` als letzter Haupteffekt eingetragen wurde: $F(1, 151) = 0.52$, $p = 0.47$.
- Die Interaktion kommt immer nach den Haupteffekten, weshalb ihr Ergebnis in beiden Modellen gleich ist: $F(1, 151) = 11.4$, $p < 0.001$.

Diese Ergebnisse kann man automatisch mit der `Anova()`-Funktion (mit grossem A) aus dem `car`-Package berechnen:

```
car::Anova(b11.lm1)

## Anova Table (Type II tests)
##
## Response: Potenzial
##            Sum Sq  Df F value    Pr(>F)
## CS          1.334   1  1.9040 0.1696729
## Name        0.364   1  0.5201 0.4718954
## CS:Name     7.965   1 11.3695 0.0009484
## Residuals 105.786 151
```

Es gibt auch noch die Berechnungsart *Type-III sum of squares*, aber von dieser wird meistens abgeraten. *Type-I* und *Type-II sum of squares* können beide je nach der Situation nützlich sein, und wenn man sich nur für die Interaktion interessiert oder wenn man es genau die gleiche Anzahl Beobachtungen pro 'Zelle' gibt, macht es eh nichts aus.

Keine Daten wegschmeissen! Wenn jede Zelle im Design die gleiche Anzahl Beobachtungen hat, dann ergibt die drei Berechnungsarten für die Quadratsummen (Type-I, Type-II und Type-III) alle die gleiche Lösung. Schmeissen Sie aber bitte keine Daten weg, um so gleich grosse Zellen zu erhalten und nicht zwischen diesen Berechnungsarten wählen zu müssen!

Varianzanalyse kann man auch verwenden, wenn die Prädiktoren nicht kategorisch, sondern kontinuierlich sind. An der Vorgehensweise ändert sich nichts:

```
rs <- read_csv("Data/readingSkills.csv")
rs.lm <- lm(score ~ age + nativeSpeaker, data = rs)
anova(rs.lm)

## Analysis of Variance Table
##
## Response: score
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## age             1 12677.9 12677.9 46109.4 < 2.2e-16
## nativeSpeaker   1  1753.5  1753.5  6377.4 < 2.2e-16
## Residuals     197    54.2     0.3
```

13.5 Abkürzungen

One-way ANOVA (einfaktorielle Varianzanalyse) Eine Varianzanalyse mit einem einzigen Prädiktor. Meistens ist dies eine Gruppenvariable mit zwei oder mehreren Ausprägungen.

Two-way ANOVA (zweifaktorielle Varianzanalyse) Eine Varianzanalyse mit zwei Prädiktoren. Oft ist hierbei die Interaktion zwischen den Prädiktoren von Interesse. Wenn die eine Variable zwei Ausprägungen hat und die andere vier, spricht man oft von einer 2×4 *two-way* ANOVA. *Three-way*, *four-way*, usw., ANOVA gibt es auch: Man fügt dem Modell einfach mehr Prädiktoren hinzu.

ANCOVA Steht für *analysis of covariance*. Es handelt sich lediglich um eine Varianzanalyse, in der mindestens eine kontinuierliche Kontrollvariable berücksichtigt wird. Die Varianzanalyse des Modells `rs.lm` ist also eine ANCOVA. Für weiterführende Literatur, siehe Huitema (2011).

RM-ANOVA Steht für *repeated-measures analysis of variance*. Dieses Vorgehen kann man verwenden, wenn etwa mehrere Datenpunkte der gleichen Variablen pro Versuchsperson vorliegen. RM-ANOVA wird hier nicht besprochen, da es mittlerweile flexiblere Werkzeuge gibt, um mit Abhängigkeitsstrukturen in den Daten umzugehen (gemischte Modelle).

MANOVA Steht für *multivariate analysis of variance*. Es ist eine Erweiterung von ANOVA auf mehrere *outcomes*. Zu MANOVA schreiben Everitt & Hothorn (2011) in der Einführung ihres Buches *An introduction to applied multivariate analysis with R* Folgendes:

> But there is an area of multivariate statistics that we have omitted from this book, and that is multivariate analysis of variance (MANOVA) and related techniques (...). There are a variety of reasons for this omission. First, we are not convinced that MANOVA is now of much more than historical interest; researchers may occasionally pay lip service to using the technique, but in most cases it really is no more than this. They quickly move on to looking at the results for individual variables. And MANOVA for repeated measures has been largely superseded by the models that we shall describe [in ihrem Buch]. (S. vii-viii)

13.6 Geplante Vergleiche und Post-hoc-Tests

Mit einfaktorieller Varianzanalyse versucht man die Frage zu beantworten, ob sich die Gruppenmittel (*irgendwelche* Gruppenmittel) in der Population voneinander unterscheiden.[3] Findet man einen kleinen p-Wert, dann tendiert man dazu, davon auszugehen, dass irgendwelche Gruppenmittel sich tatsächlich voneinander unterscheiden. Die Varianzanalyse bietet aber keine Antwort auf die naheliegende Folgefrage: Welche Gruppen unterscheiden sich eigentlich genau voneinander? Unterscheiden sich Gruppen A und B, oder eher Gruppen A und C oder B und C?

[3] Oft ist diese Frage ziemlich trivial: In vielen Situationen kann man kaum davon ausgehen, dass die Gruppenmittel in der Population einander *genau* gleich sind.

Um solche Fragen zu beantworten, bedienen sich Forschende oft auf die Varianzanalyse folgender Signifikanztests. Wenn sich die gestellten Fragen erst *nach* der Datenerhebung ergeben und nicht im Vorhinein aus der Theorie abgeleitet wurden (exploratorische Analyse), spricht man von **Post-hoc-Tests**. Wenn die Fragen schon *vor* der Untersuchung vorlagen (konfirmatorische Analyse), spricht man von **geplanten Vergleichen**.

Häufig verwendete Verfahren für solche nachfolgende Tests tragen Namen wie 't-Tests mit Bonferroni-Korrektur', 't-Tests mit Holm–Bonferroni-Korrektur', 'Fishers *least significant difference*-Test', 'Scheffé-Test' usw. Die Idee ist, dass das aufgrund der mehrfachen Tests gestiegene globale Risiko, einen Fehler der ersten Art zu begehen, kontrolliert werden muss.

Zusätzliche Tests sind jedoch nicht immer nötig oder erwünscht. Entscheidend sind meines Erachtens die Theorie und die Hypothesen, die der Studie zu Grunde lagen, und welche Datenmuster man als Belege für dese Theorie und Hypothesen betrachten würde:

- Sagt die Theorie voraus, dass es *irgendwelche* Gruppenunterschiede (egal welche) geben wird, dann reicht eine Varianzanalyse aus. Man kann zusätzlich eventuell interessante Gruppenunterschiede deskriptiv berichten, etwa indem man das lineare Modell, für das man die Varianzanalyse ausgeführt hat, grafisch darstellt. Diese möglichen Unterschiede überlässt man dann einer neuen, konfirmatorischen Studie (siehe Bender & Lange, 2001, S. 344). Falls die Varianzanalyse keine Signifikanz ergibt, sollte man in diesem Fall auch auf zusätzliche Tests verzichten, sodass man den *familywise Type-I error rate* nicht erhöht.
- Sagt die Theorie jedoch einen *spezifischen* Gruppenunterschied voraus, oder werden mehrere separate Theorien überprüft, die sich auf unterschiedliche Gruppenmittel beziehen (z.B. A vs. B und C vs. D), dann braucht man eigentlich die Varianzanalyse nicht auszuführen und reichen t-Tests. Allfällige interessante aber nicht vorhergesagte Gruppenunterschiede werden deskriptiv (nicht inferenzstatistisch) berichtet und man überlasst sie wiederum einer neuen, konfirmatorischen Studie.
- Sagt die Theorie voraus, dass sich ein bestimmter Unterschied *oder* ein bestimmter anderer Unterschied zeigen wird, dann sollte man sich über die oben angesprochenen Methoden schlau machen. Dies gilt auch wenn die Theorie komplexere Gruppenunterschiede vorhersagt, etwa 'Das Gesamtmittel von Gruppen A und B ist niedriger als das Gesamtmittel von Gruppen B, C und D'. Zu diesen Verfahren kann ich Ihnen leider keine detaillierten Ratschläge machen, da ich sie selber noch nie eingesetzt habe.
- Sagt die Theorie voraus, dass sich ein bestimmter Unterschied *und* ein bestimmter anderer Unterschied zeigen wird, dann reichen m.E. wiederum zwei t-Tests.

Eine kurze Einführung mit vielen Referenzen ist Bender & Lange (2001); Ruxton & Beauchamp (2008) geben konkrete Ratschläge, denen jedoch wohl schwierig zu folgen ist, wenn man noch keine konkrete Erfahrung mit derartigen Analysen hat. Ein Blogeintrage zum Thema ist *On correcting for multiple comparisons: Five scenarios* (1.4.2016).

Seien Sie vorsichtig und sparsam mit Post-Hoc-Erklärungen. Im Nachhinein gelingt es einem oft, gewisse Muster in den Daten theoretisch zu deuten. Dabei ist es durchaus möglich, dass diese Muster rein zufallsbedingt sind und sich bei einer neuen Studie nicht mehr ergeben.

Es ist übrigens erstaunlich einfach, sich selbst im *Nachhinein* weiszumachen, dass man ein bestimmtes Muster in den Daten *vorher*gesagt hat. Ein wichtiger Grund hierfür ist, dass wissenschaftliche Theorien und Hypothesen oft mehreren statistischen Hypothesen entsprechen. Um zu vermeiden, dass man zuerst sich selbst und nachher seinen Lesenden vortäuscht, dass man die Muster in den Daten so vorhergesagt hat, wie sie sich ergeben haben, kann man seine wissenschaftlichen und statistischen Hypothesen präregistrieren: Man spezifiziert vor der Datenerhebung, welche Muster man erwartet und wie man die Daten analysieren wird. Für mehr Informationen zu Präregistration, siehe etwa `http://datacolada.org/64` und Wagenmakers et al. (2012b).

13.7 Zwischenfazit Signifikanztests

- p-Werte drücken die Wahrscheinlichkeit aus, mit der man das beobachtete Muster (z.B. einen Gruppenunterschied oder eine andere Parameterschätzung) oder noch extremere Muster antreffen würde, wenn die Nullhypothese tatsächlich stimmen würde.
- Die Nullhypothese ist fast ausnahmlos, dass das Muster nur aufgrund von Zufall zu Stande gekommen ist. Anders ausgedrückt: Sie besagt, dass der eigentliche Gruppenunterschied bzw. der Parameter, den man zu schätzen versucht hat, gleich 0 ist.
- $0 = 0.00000000$.
- Auch wenn die Nullhypothese nicht stimmt, muss das nicht heissen, dass dafür Ihre wissenschaftliche Hypothese stimmt: Vielleicht gibt es noch andere theoretische Ansätze, die mit den Befunden kompatibel ist, oder vielleicht verzerrt Ihre Datenerhebung die Ergebnisse.
- Oft versucht man, die Wahrscheinlichkeit, dass man eine tatsächlich zutreffende Nullhypothese zu Unrecht ablehnt, auf 5% zu reduzieren, indem man nicht schlussfolgert, dass die Nullhypothese nicht stimmt, wenn $p > 0.05$.
- $p > 0.05$ heisst aber *nicht*, dass die Nullhypothese stimmt.
- Permutationstests, t-Tests und F-Tests haben alle den gleichen Zweck. t-Tests verwendet man dabei, um die Nullhypothese für Parameterschätzungen (z.B. Gruppenunterschiede oder Regressionskoeffizienten) zu testen; F-Tests, um zu testen, ob ein Prädiktor (oft mit mehr als zwei Ausprägungen) mehr Streuung in den Daten beschreibt als man rein durch Zufall erwarten könnte.

- t- und F-Tests können vom allgemeinen linearen Modell abgeleitet werden. Ihre Annahmen sind den Annahmen dieses Modells gleich.
- Es gibt Signifikanztests, denen wir noch nicht begegnet sind. Von der Berechnung her unterscheiden diese sich von den t- und F-Tests, aber ihr Ziel ist nach wie vor gleich: Die Wahrscheinlichkeit berechnen, mit der man ein beobachtetes oder ein noch extremeres Muster antreffen würde, wenn dieses Muster nur Zufall zuzuschreiben wäre.

Kapitel 14

Powerberechnungen

Auch wenn es tatsächlich einen Unterschied zwischen zwei Gruppen oder einen Zusammenhang zwischen einigen Variablen auf der Populationsebene gibt, ist es möglich, dass man in einer Stichprobe diesen Unterschied oder diesen Zusammenhang nicht mit einem Signifikanztest belegen kann. Diese Tatsache wird in Abbildung 14.1 auf der nächsten Seite illustriert. Die Wahrscheinlichkeit, zu der ein Signifikanztest ein tatsächlich vorhandenes Muster belegt, nennt man seine **power**. Fürs Planen von Studien kann es nützlich sein, diese *power* in Erwägung zu ziehen. In diesem Kapitel wird daher gezeigt, wovon die *power* eines Tests abhängt und wie man diese berechnen kann.[1] Eine kurze Einführung in die Powerberechnung ist Cohen (1992); ein weiterer lesenswerter Artikel ist Kelley et al. (2003).

14.1 Power mit Simulationen berechnen

Um die *power* des in Abbildung 14.1 illustrierten Szenarios zu berechnen, können wir eine Simulation schreiben. Der Simulationscode unten bewirkt das folgende:

1. Wir definieren die Standardabweichung der zwei Populationen. In dieser Simulationen ist die Standardabweichung $\sigma = 2$ für beide Populationen. Übrigens sind beide Verteilungen normalverteilt, da dies eine Annahme des t-Tests ist, die für die Analyse verwendet wird.
2. Wir definieren den Unterschied zwischen den Mitteln der beiden Normalverteilung. Hier $\mu_B - \mu_A = 0.7$.
3. Wir legen die Anzahl Beobachtungen pro Stichprobe fest. Hier $n_A = n_B = 32$.

[1] Der Ehrlichkeit halber füge ich dem hinzu, dass ich selber selten Powerberechnungen ausführe. Der erste Grund hierfür ist, dass – wie Sie in diesem Kapitel sehen werden – eine sinnvolle Powerberechnung voraussetzt, dass man bereits über viele Informationen über das Problem verfügt. Der zweite Grund ist, dass Powerberechnungen vor allem dann nützlich sind, wenn man seine Schlussfolgerungen hauptsächlich auf p-Werten basiert—was ich immer mehr zu vermeiden versuche.

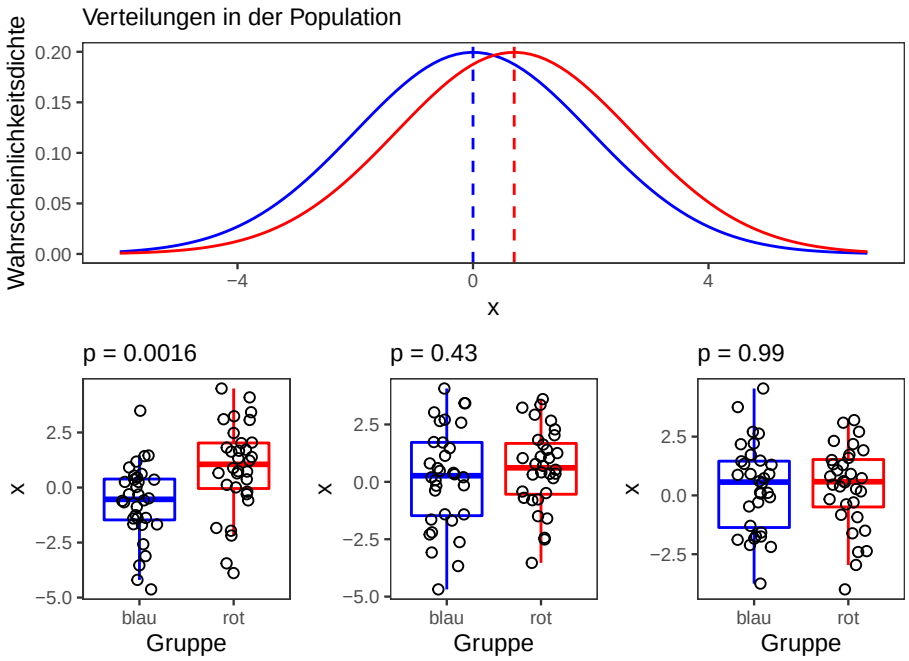


Abbildung 14.1: *Obere Zeile:* Zwei Normalverteilungen mit $\sigma = 2$. Das Mittel der blauen Normalverteilung liegt bei 0; das Mittel der roten bei 0.7. *Untere Zeile:* Aus diesen Normalverteilungen wurden drei Mal jeweils Zufallsstichproben mit 32 Beobachtungen gezogen. Obwohl es in der Population einen Unterschied zwischen den Mitteln dieser Verteilungen gibt, kann es durchaus vorkommen, dass man keinen signifikanten Unterschied zwischen den Stichprobenmitteln feststellt. Die Wahrscheinlichkeit, zu der man einen signifikanten Unterschied feststellt, wenn es in der Population tatsächlich einen Unterschied geben, nennt man die *power* eines Signifikanztests.

4. Wir ziehen 10'000 Mal eine Stichprobe aus jeder Population.
5. Diese Stichproben werden in einem allgemeinen linearen Modell modelliert und der p-Wert des Unterschieds (basierend auf einer t-Verteilung) wird gespeichert. (Für diesen letzten Schritt könnte man auch die `t.test()`-Funktion verwenden, aber diesen Code kann man einfacher anpassen, um komplizierteren Designs gerecht zu werden.)
6. Die Verteilung der p-Werte wird dargestellt (Abbildung 14.2).
7. Die Proportion der signifikanten p-Werte ($p < 0.05$) wird berechnet.

```
# Merkmale der Population; rumspielen!
sd_gruppe <- 2
unterschied <- 0.7

# Stichprobengrössen; rumspielen!
n_gruppeA <- 32
n_gruppeB <- 32

# Simulationseinstellungen
n_runs <- 10000
p_werte <- vector(length = n_runs)

for (i in 1:n_runs) {
  # Datensatz mit Gruppenvariabler (A vs. B) und outcome
  df <- data.frame(
    gruppe = rep(c("A", "B"), times = c(n_gruppeA, n_gruppeB)),
    outcome = c(rnorm(n = n_gruppeA, mean = 0, sd = sd_gruppe),
                rnorm(n = n_gruppeB, mean = unterschied, sd = sd_gruppe))
  )

  # Modell
  mod.lm <- lm(outcome ~ gruppe, data = df)

  # p-Wert auslesen
  p_werte[i] <- summary(mod.lm)$coefficients[2, 4]
}
```

```
# Proportion signifikanter p-Werte
mean(p_werte < 0.05)
```

```
## [1] 0.2832
```

Schlussfolgerung: Wenn die Populationen normalverteilt mit $\sigma = 2$ sind und es einen Unterschied von 0.7 zwischen ihren Mitteln gibt, dann gibt es eine Wahrscheinlichkeit von (nur!) 28%, dass man einen signifikanten Unterschied feststellt, wenn man Zufalls-

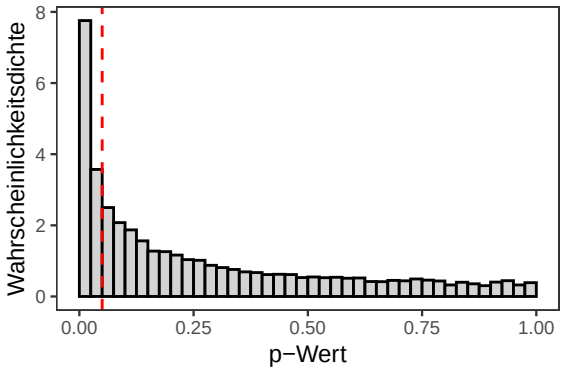


Abbildung 14.2: Die Verteilung der p-Werte in der Powersimulation.

stichproben von je 32 Beobachtungen zieht. (Der Signifikanztest hier ist ein zweiseitiger t-Test unter Annahme von gleichen Varianzen.)

Aufgabe 1 Wie ändert sich die *power* in diesem Beispiel, wenn die Normalverteilungen Standardabweichungen von $\sigma = 3$ statt von $\sigma = 2$ hätten?

Aufgabe 2 Wie ändert sich die *power* in diesem Beispiel, wenn wir statt 32 Beobachtungen in jeder Stichprobe, 50 Beobachtungen in einer Stichprobe und 14 in den anderen hätten?

Aufgabe 3 Wie viel *power* hätte man, wenn der Unterschied zwischen den zwei Populationen infinitesimal ist? Überprüfen Sie Ihre Antwort mit einer Simulation.

14.2 Algebraische Powerberechnung

Für ein paar Signifikanztests, darunter t-Tests, kann man die *power* auch algebraisch berechnen. Die Formeln dazu werden hier nicht gezeigt, aber sie sind in der `power.t.test()`-Funktion implementiert worden. Für einen Unterschied von 0.7 Einheiten zwischen Normalverteilungen mit $\sigma = 2$ und Stichproben mit je 32 Beobachtungen, kann man die *power* so berechnen:

```
power.t.test(n = 32, delta = 0.7, sd = 2)
```

```
##
##      Two-sample t test power calculation
##
##               n = 32
##           delta = 0.7
##              sd = 2
```

```
##       sig.level = 0.05
##           power = 0.2804138
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Der Vorteil dieser Funktion ist, dass man die erwünschte *power* einstellen kann, dafür die Anzahl Beobachtungen leer lässt, um zu erfahren, wie viele Beobachtungen man pro Stichprobe bräuchte, um die erwünschte *power* zu erreichen:

```
power.t.test(delta = 0.7, sd = 2, power = 0.90)
```

```
##
##      Two-sample t test power calculation
##
##               n = 172.5158
##           delta = 0.7
##              sd = 2
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

173 Beobachtungen pro Stichprobe bräuchte man also, um zu einer Wahrscheinlichkeit von 90% einen signifikanten Unterschied zu finden, auch wenn die Populationen um 0.7 Einheiten voneinander abweichen und eine Standardabweichung von 2 haben.

Man kann auch die anderen Parameter leer lassen:

```
power.t.test(n = 40, sd = 2, power = 0.75)
```

```
##
##      Two-sample t test power calculation
##
##               n = 40
##           delta = 1.192908
##              sd = 2
##       sig.level = 0.05
##           power = 0.75
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Wenn man 75% *power* haben möchte, 40 Beobachtungen pro Gruppe hat und davon ausgeht, dass die Populationen eine Standardabweichung von 2 haben, muss man also hoffen, dass der Unterschied zwischen den Populationsmitteln mindestens 1.2 Einhei-

ten beträgt.

Um eine Powerberechnung durchzuführen, muss man über mindestens drei der vier folgenden Informationen verfügen:

- Die Anzahl Beobachtungen.
- Die Effektgrösse in der Population (hier: Unterschied zwischen den Populationsmitteln). Dies kann die erwartete oder verhoffte Effektgrösse sein, aber auch die kleinste Effektgrösse, die man selber für relevant halten würde.
- Die Variabilität in der Population (hier: Standardabweichung der Verteilungen).[2]
- Die erwünschte *power*.

Abbildung 14.3 illustriert, wie diese vier Faktoren zusammenhängen. Meines Erachtens ist es übrigens nicht sehr sinnvoll, über *die* power eines Tests zu sprechen: Sinnvoller wäre es, zu sagen, dass ein Test unter bestimmten Annahmen (betreffend den eigentlichen Unterschied, die Variabilität innerhalb der Gruppen und die Verteilung der Residuen) eine gewisse power hätte und unter anderen Annahmen eine andere. Abbildung 14.3 bringt dies m.E. auf den Punkt.

```
# Kombinationen von Stichprobengrössen, Unterschieden
# und Standardabweichungen generieren, für die man die power
# berechnen will.
power.df <- expand.grid(
  Stichprobengroesse = seq(from = 2, to = 100, by = 1),
  Delta = seq(0.3, 1.5, by = 0.3),
  SD = c(0.5, 1, 1.5, 2)
)

# Power berechnen und speichern
power.df$Power <- power.t.test(n = power.df$Stichprobengroesse,
                               delta = power.df$Delta,
                               sd = power.df$SD)$power

# Text hinzufügen
power.df$SD <- paste("Standardabweichung Restfehler:\n", power.df$SD)

# Grafik zeichnen
ggplot(power.df,
       aes(x = Stichprobengroesse,
           y = Power,
           colour = factor(Delta))) +
  geom_line() +
```

[2] Im Prinzip ist nur das *Verhältnis* zwischen der Effektgrösse und der Variabilität relevant; siehe Cohen (1977, 1992).

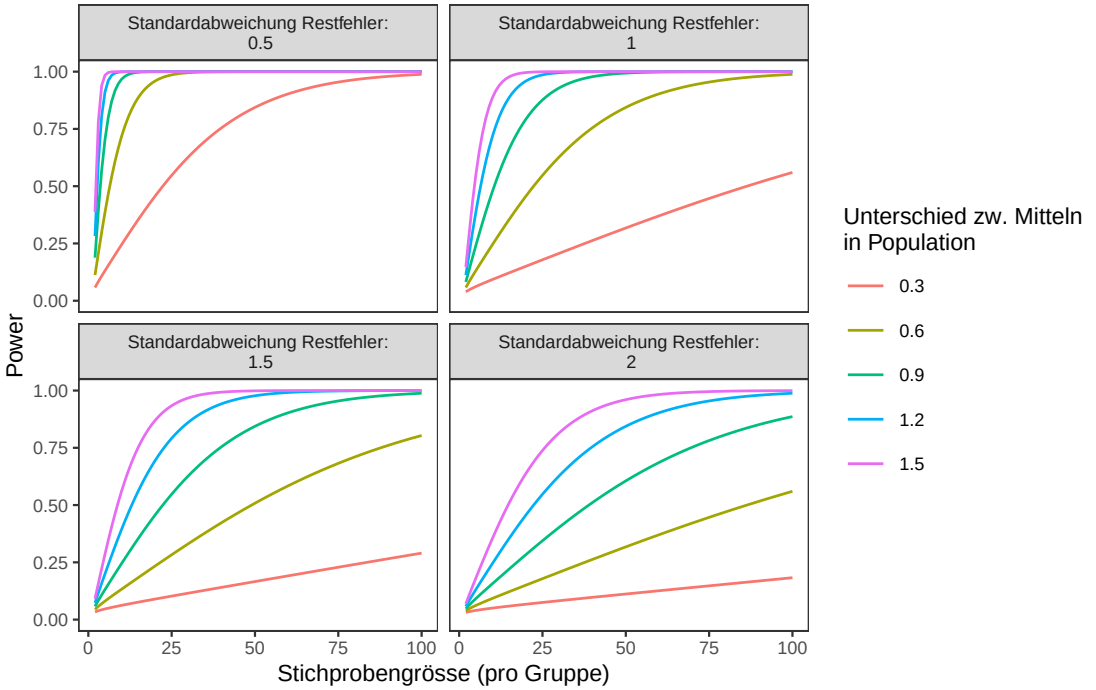


Abbildung 14.3: Die power eines Tests hängt von der Effektgrösse (hier: dem Unterschied zwischen den Mitteln in der Population), der Fehlervarianz (hier gezeigt als die Standardabweichung des Restfehlers) und die Datenmenge ab.

```
  ylim(0, 1) +
  facet_wrap(~ SD) +
  xlab("Stichprobengrösse (pro Gruppe)") +
  scale_colour_discrete(name = "Unterschied zw. Mitteln\nin Population")
```

14.3 Weitere Ressourcen

Das `pwr`-Package (Champely, 2018) bietet weitere Funktionen zur Powerberechnung.

Artikel und Blogeinträge:

- Sedlmeier & Gigerenzer (1989), *Do studies of statistical power have an effect on the power of studies?*
- `http://jakewestfall.org/blog/index.php/2015/06/16/dont-fight-the-power-analysis/`
- Westfall et al. (2014), *Statistical power and optimal design in experiments in which samples of participants respond to samples of stimuli* (eher für Fortgeschrittene, aber relevant für Experimente in den Sprachwissenschaften)

Kapitel 15

Überflüssige Signifikanztests

Wann und ob Signifikanztests sinnvoll sind, darüber scheiden sich die Geister. Was weniger umstritten ist, ist, dass Signifikanztests oft in Kontexten eingesetzt werden, in denen sie überflüssig sind (sog. *silly tests*, Abelson, 1995). Überflüssige Tests führen dazu, dass Forschungsberichte schwerer zugänglich werden, da Lesende noch mehr als sonst die relevanten Informationen von den irrelevanten trennen müssen. Im Folgenden werden daher fünf Arten überflüssiger Signifikanztests besprochen, sodass Sie diese in Ihren eigenen Arbeiten nicht verwenden.

15.1 RM-ANOVA für Prätest/Posttest-Designs

Ein nützliches Forschungsdesign ist das Prätest/Posttest-Design mit einer Experimental- und einer Kontrollgruppe. Alle Teilnehmenden erledigen zuerst eine Aufgabe (Prätest), werden dann nach dem Zufallsprinzip der Experimental- oder Kontrollgruppe zugewiesen und erledigen nach Ablauf der Intervention die gleiche oder eine ähnliche Aufgabe (Posttest). Solche Daten werden oft mit einer Varianzanalyse mit Messwiederholungen (RM-ANOVA) ausgewertet. Die Ergebnisse könnten dann etwa wie folgt berichtet werden:

> A repeated-measures ANOVA yielded a nonsignificant main effect of Condition ($F(1, 48) < 1$) but a significant main effect of Time ($F(1, 48) = 154.6$, $p < 0.001$). In addition, the Condition $\times$ Time interaction was significant ($F(1, 48) = 6.2$, $p = 0.016$). [Nach einem fiktiven Beispiel von Vanhove, 2015.]

Das Problem mit dieser Analyse ist, dass drei Tests berichtet werden ('main effect of Condition', 'main effect of Time' und 'Condition $\times$ Time interaction'), aber nur die Interaktion von Interesse ist: Interessant ist nur, ob sich die Leistung in der Experimentalgruppe mehr verbessert hat als jene in der Kontrollgruppe.

- Dass die Posttestleistung über die beiden Gruppen hinweg besser ist als die Prä-

testleistung ('main effect of Time'), ist uninteressant: Vielleicht handelt es sich um einen Übungseffekt, vielleicht ist der Posttest einfacher als der Prätest, vielleicht haben die Teilnehmenden in beiden Gruppen etwas aus dem Experiment gelernt.

- Dass sich die Leistung der beiden Gruppen über die beiden Messpunkte hinweg unterscheidet oder nicht ('main effect of Condition'), ist auch nicht relevant für unsere Frage: Vielleicht gibt es im Schnitt tatsächlich kaum einen Unterschied, aber haben die Teilnehmenden in der Experimentalgruppe trotzdem mehr gelernt (z.B. wenn ihre Leistung beim Prätest niedriger war als jene der Kontrollgruppe, aber dafür beim Posttest höher).
- Nur der Interaktionseffekt ist hier interessant: Ist der Effekt von 'Time' unterschiedlich gross je nach 'Condition'?

Diese Analyse kann wesentlich vereinfacht werden, indem nicht die Rohdaten, sondern die *Unterschiede* zwischen der Posttest- und dem Prätestleistung jeder Versuchsperson mit einem t-Test ausgewertet werden. Am Ergebnis würde sich nichts ändern, aber die Analyse ist jetzt besser nachvollziehbar und berichtet wird nur ein einziger (relevanter) Test.

Eine noch bessere Alternative besteht darin, ein lineares Modell zu konstruieren, in dem die Posttestergebnisse als *outcome* gelten und sowohl die Gruppenzugehörigkeit als auch die Prätestergebnisse als Prädiktoren mitmodelliert werden (= ANCOVA). Dies führt zu einer grosseren *power* bzw. zu mehr Präzision und hat den Vorteil, dass Prä- und Posttest nicht einmal auf der gleichen Skala ausgedrückt werden müssen. Siehe Vanhove (2015) für mehr Informationen.

15.2 *Balance tests*

Auch wenn sie die Teilnehmenden nach dem Zufallsprinzip den unterschiedlichen Gruppen zugeordnet haben, beschreiben Forschende oft, wie sich die Gruppen voneinander unterscheiden, was etwa das Alter, die Fremdsprachenkenntnisse oder den IQ der Versuchspersonen betrifft. Hierzu werden dann oft Signifikanztests eingesetzt. In Vanhove (2015) bespreche ich drei Gründe, weshalb solche Signifikanztests überflüssig sind; diese sind die zwei wichtigsten:

- Solche Signifikanztests überprüfen die Nullhypothese, dass sich (etwa) das Durchschnittsalter in den unterschiedlichen Gruppen nur aufgrund von Zufall unterscheidet. Da die Versuchspersonen den Gruppen aber nach dem Zufallsprinzip zugewiesen wurden, *wissen* wir, dass diese Nullhypothese tatsächlich stimmt. Entweder sagt der Test dann, dass es keinen Unterschied zwischen den Gruppen hinsichtlich dieser Hintergrundsvariablen gibt – was wir schon wussten – oder er sagt, dass es schon einen Unterschied gibt – was falsch wäre (Fehler der ersten Art). Der Test kann uns also nichts sagen, was stimmt *und* was wir noch nicht wussten.

- Die Verwendung solcher *balance tests* lässt vermuten, dass Forschende denken, dass das Ziel von Randomisierung ist, perfekt balanzierte Gruppen zu generieren. Wie wir in Kapitel 12 bereits gesehen haben, ist dies eben nicht das Ziel. Vielmehr ist das Ziel von Randomisierung, eine *systematische* Verzerrung vorzubeugen. In den p-Werten und Konfidenzintervallen sind Gruppenunterschiede, die aufgrund von Zufall zu Stande kommen, bereits berücksichtigt. Es ist also nicht nötig, dass Gruppen hinsichtlich der Hintergrundsvariablen perfekt balanziert sind.[1]

15.3 Tautologische Tests

> The 70 participants were divided into three proficiency groups according to their performance on a 20-point French-language test. The high-proficiency group consisted of participants with a score of 16 or higher ($n = 20$); the mid-proficiency group of participants with a score between 10 and 15 ($n = 37$); and the low-proficiency group of participants with a score of 9 or lower ($n = 13$). An ANOVA showed significant differences in the test scores between the high-, mid- and low-proficiency groups ($F(2, 67) = 133.5$, $p < 0.001$). [Fiktives Beispiel aus dem Blogeintrag *Silly significance tests: Tautological tests* (15.10.2014)]

Das Problem mit diesem Signifikanztest ist, dass er vollkommen tautologisch ist. Ähnlich wie *balance tests* können solche tautologischen Tests uns nichts sagen, was wir noch nicht wussten und was gleichzeitig auch stimmt: Wir haben die Gruppen absichtlich so konstruiert, dass sie sich hinsichtlich einer bestimmten Variablen nicht überlappen. Selbstverständlich stimmt die Nullhypothese ('Jeglicher Unterschied beruht rein auf Zufall.') hier nicht.

Tautologische Tests sind m.E. symptomatisch für ein wichtigeres Problem: die Vorstellung, dass Prädiktoren unbedingt kategorisch sein sollten bzw. dass man immer Grüppchen bilden sollte. Indem man aus kontinuierlichen Daten Gruppen bildet, verliert man Information und dadurch auch statistische Genauigkeit. Siehe hierzu weiter *The problem with cutting up continuous variables and what to do when things aren't linear* (16.10.2015).

15.4 Tests, die nichts mit der Forschungsfrage zu tun haben

Erstaunlich häufig sind Signifikanztests, die eigentlich gar nichts mit der Forschungsfrage zu tun haben. Ein fiktives Beispiel hierfür ist, wenn man eine Interventionsstudie durchführt und nicht nur die Leistung der Kontroll- und Interventionsgruppe ver-

[1] Wichtige Hintergrundsvariable können natürlich in der Analyse berücksichtigt werden, aber das macht man besser, indem man sie dem Modell als Prädiktoren hinzufügt. Gegebenfalls kann man *zusätzlich* die Randomisierung so einschränken, dass die Gruppen hinsichtlich solcher Variablen vergleichbarer sind (*blocking*, siehe Imai et al., 2008, und Vanhove, 2015).

gleicht, sondern auch noch die Leistung der Jungen und der Mädchen, oder die Leistung bei unterschiedlichen Sozialschichten usw. Faktoren wie Geschlecht und Sozialschicht mögen die Leistung beeinflüssen, aber nur deswegen müssen diese Faktoren nicht nochmals mit einem Signifikanztest überprüft werden. Wenn man davon ausgeht, dass solche Faktoren wichtig sind, ist es sinnvoller, wenn man sie sofort im Modell mitberücksichtigt anstatt separate Analysen mit ihnen durchzuführen.

Für weitere Empfehlungen, siehe den Blogeintrag (siehe *Silly significance tests: Tests unrelated to the genuine research questions*, 8.6.2015).

15.5 Tests für Haupteffekte, während man sich für die Interaktion interessiert

Oft interessiert man sich für die Interaktion zwischen zwei oder mehreren Prädiktoren. Wenn man diese Interaktion in einer Varianzanalyse testet, erhält man aber auch Signifikanztests für die Haupteffekte. Aber wichtig sind diese letzteren Tests nicht. M.E. tragen sie im Gegenteil dazu bei, dass Forschungsberichte schwer verdaulich werden. Wenn es aus irgendeinem Grund unbedingt nötig sein sollte, dass Signifikanztests für uninteressante Haupteffekte berichtet werden, sollten diese m.E. in einer Tabelle stehen, deren Beschriftung dann deutlich macht, dass eigentlich nur die Interaktion relevant ist.

15.6 Fazit

Nur weil man einen Signifikanztest durchführen könnte oder weil die Software einen ausspuckt, heisst das noch nicht, dass man ihn auch berichten sollte. Fragen Sie sich immer, wie relevant jeder Signifikanztest wäre – und zwar für die Fragen, die Sie beantworten wollen.

Missverstehen Sie den letzten Absatz bitte nicht: Ich schlage *nicht* vor, dass man nur jene Signifikanztests berichten soll, die einen signifikanten p-Wert ergeben. Ausserdem denke ich auch nicht, dass jede Analyse mit einem p-Wert belegt werden sollte.

Kapitel 16

Signifikanztests und fragwürdige Forschungspraktiken

Wenn man mit Signifikanztests umgeht – sei es, weil man sie selber verwendet oder weil man sie beim Lesen der Forschungsliteratur oft antrifft –, sollte man sich einiger häufiger Fehlanwendungen bewusst sein, die dazu führen, dass p-Werte nicht länger ihre angebliche Bedeutung haben. Das übergreifende Problem hinter all diesen fragwürdigen Forschungspraktiken ist grundsätzlich, dass Forschende und Herausgeber bestimmte Ergebnisse (lies: signifikante Ergebnisse) bevorzugen und dass der Forschungs- und Publikationsprozess daher auf das Produzieren und Publizieren von solchen Ergebnissen ausgerichtet ist. Wenn man bei der Datenerhebung, Analyse und Veröffentlichung aber flexibel genug vorgeht, ist es äusserst einfach, signifikante Muster zu finden und zu berichten – ohne dass die 'Befunde' einigermassen der Realität entsprechen.

Damit Sie sich selbst über fragwürdige, mit Signifikanztests verknüpfte Forschungspraktiken informieren können, werden hier einige einschlägige Artikel vorgestellt. Eine gute Übersicht über diese Probleme in Buchform ist Chambers (2017, *The seven deadly sins of psychology*).

16.1 Sterling et al. (1995) zu *publication bias*

Sterling et al. (1995, 4 Seiten Text) zählten, wie oft Signifikanztests in medizinischen und psychologischen Fachzeitschriften einen signifikanten p-Wert ergaben. In den medizinischen Zeitschriften wurde die Nullhypothese in 85% der Fälle abgelehnt, in den Psychologiezeitschriften sogar in satten 96%. Wie Sterling et al. erklären, zeigen diese Zahlen, dass Forschende und Herausgeber gegen nicht-signifikante Befunde diskriminieren, denn sogar wenn die Nullhypothese nie stimmen würde, wäre es unmöglich, dass so viele Tests signifikante Ergebnisse aufweisen. Ihre Vermutung ist, dass Forschende dazu neigen, Artikel zu signifikanten (statt zu nicht-signifikanten) Befunden

zu schreiben, und dass Herausgeber dazu neigen, Artikel mit nicht-signifikanten Befunden abzulehnen. Dieses *publication bias* kann dazu führen, dass die Literatur die Evidenz und die Stärke eines bestimmten Effekts (etwa 'Zweisprachigkeit führt zu kognitiven Vorteilen.') überschätzt (siehe auch de Bruin et al., 2015).

16.2 Kerr (1998) zu *hypothesizing after the results are known*

Kerr (1998, 20 Seiten Text) bespricht Gründe und Nachteile einer häufig vorkommenden Praxis beim Schreiben wissenschaftlicher Artikel: Die Forschenden suchen in ihren Daten nach interessanten Mustern, aber anstatt dass sie ihre Befunde darstellen als das Ergebnis einer exploratorischen Analyse, wird der Bericht geschrieben, als ob sie das beobachtete Muster vorhergesagt hätten. Diese Praxis bezeichnet Kerr als HARKing – *hypothesizing after the results are known*. Als Kosten von HARKing sieht er unter anderen die folgenden (S. 211):

- "Translating Type I errors into hard-to-eradicate theory."
- "Disguising post hoc explanations as a priori explanations (when the former tend also be more ad hoc, and consequently, less useful)."
- "Not communicating valuable information about what did not work."
- "Encouraging retention of too-broad, disconfirmable old theory."
- "Inhibiting identification of plausible alternative hypotheses."
- "Implicitly violating basic ethical principles."

HARKing ist eng verknüpft mit Rosinenpickerei (*cherry picking*). Hierbei schaut man sich (explizit aber auch implizit!) unterschiedliche Muster an (Muster könnten hier etwa Gruppenunterschiede oder Korrelationen sein) und berichtet dann nur die stärksten oder anderswie auffälligsten. Das Problem hiermit wird im xkcd-Cartoon in Abbildung 16.1 auf den Punkt gebracht.

Diese und verwandte Probleme werden auch auf verständliche Art und Weise von de Groot (2014, ursprunglich veröffentlicht auf Niederländisch im Jahr 1956; 3.5 Seiten Text + 3 Seiten Kommentar) besprochen. Siehe auch Berthele (2019) zu ähnlichen Problemen in der angewandten Linguistik.

16.3 Simmons et al. (2011) zu intransparenter Flexibilität beim Erheben und Analysieren von Daten

Simmons et al. (2011, 7 Seiten Text) zeigen mit einem eigenen echten Beispiel und anhand von Simulationen, dass die Verwendung einiger geläufiger Praktiken dazu füh-

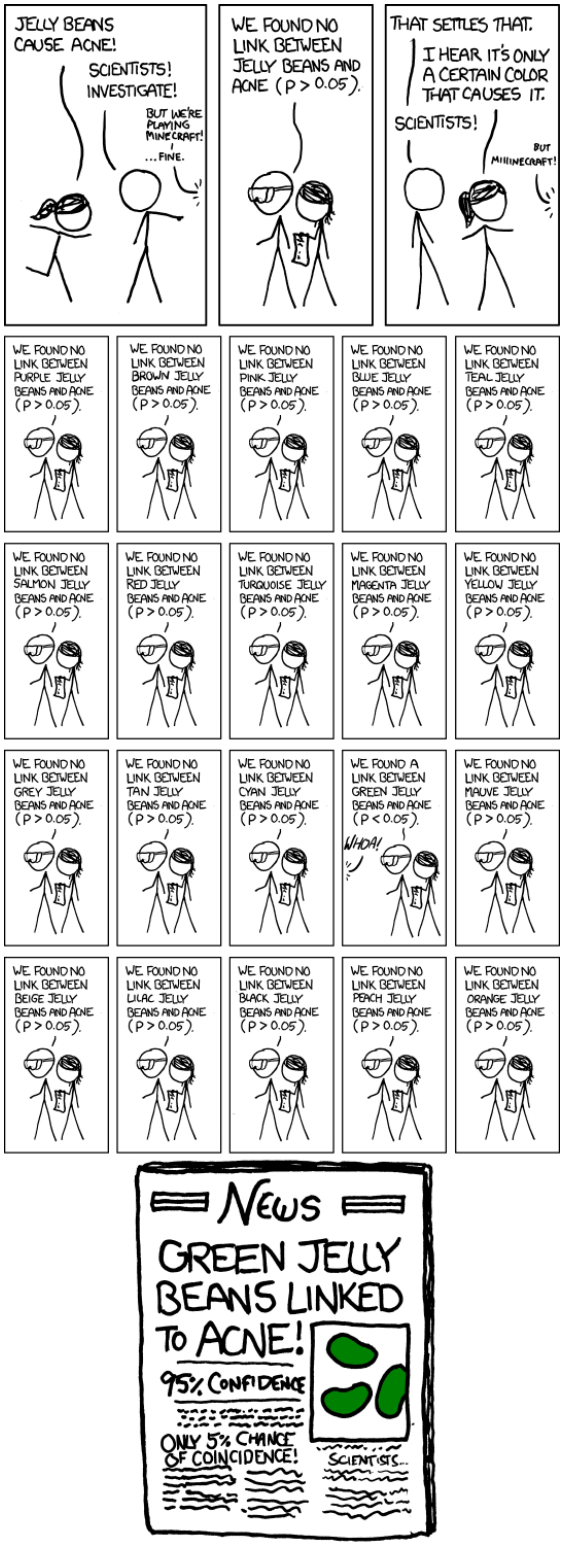


Abbildung 16.1: Quelle: `https://xkcd.com/882/`

ren kann, dass man die Nullhypothese zu einer sehr hohen Wahrscheinlichkeit ablehnt, auch wenn diese stimmt. Die untersuchten Praktiken sind:

- mehrere *outcomes* analysieren und dann eben nur denjenigen, der für die 'besten' Ergebnisse sorgt, berichten;
- nach der ersten Welle der Datenerhebung schauen, ob es schon ein signifikantes Ergebnis gibt; wenn nicht, weitere Daten erheben;[1]
- Kontrollvariablen erst in der Analyse berücksichtigen, wenn das Ergebnis ohne sie nicht-signifikant ist;[2]
- im Design mehrere Konditionen haben, aber einige davon weglassen, wenn dies zu 'besseren' Ergebnissen führt.

Selbstverständlich gibt es noch andere Praktiken, die die Fehlerquote noch erhöhen können – zum Beispiel einen lockeren Umgang mit Ausreissern. Das übergreifende Problem in all diesen Fällen ist, dass man signifikante Ergebnisse als 'gut' oder 'informativ' und nicht-signifikante Ergebnisse als 'schlecht' oder 'nicht informativ' betrachtet. Daher betrachtet man dann wiederum Entscheide in der Datenerhebung oder Analyse, die zu signifikanten Ergebnissen führen (z.B. das Weglassen bzw. das Hinzufügen einer Kontrollvariablen oder das Weglassen bzw. Behalten einiger Ausreisser), als verteidigbar oder logisch, während man dies vielleicht nicht gemacht hätte, wenn diese Entscheide zu nicht-signifikanten Ergebnissen geführt hätten.

Es sei hier noch darauf hingewiesen, dass die Autoren mittlerweile nicht mehr hinter ihrer zweiten Empfehlung stehen (*Authors must collect at least 20 observations per cell or else provide a compelling cost-of-data-collection justification*); siehe Simmons et al. (2018).

Komplimentär zu diesem Artikel sind die Artikel von Gelman & Loken (2013) und Steegen et al. (2016), in denen gezeigt wird, wie viel Flexibilität Forschende bei der Auswertung überhaupt haben (siehe auch Poarch et al., 2019). Wissenschaftliche Theorien können meistens mehreren statistischen Hypothesen entsprechen. Diese Flexibilität kann dazu führen, dass sich Forschende (auch unbewusst) auf jene Muster in den Daten fokussieren, die sie als kompatibel mit ihrer Theorie betrachten. Dabei können sie dann aber aus dem Auge verlieren, dass es andere Auswertungsmöglichkeiten geben dürfte, die zu Ergebnissen führen, die nicht mit der Theorie kompatibel sind. Ähnliches wird auch in einer ethnografischen Studie von Peterson (2016) aufgezeigt.

[1] Wenn man dies selber vorhat, sollte man sich bei etwa Lakens (2014) darüber schlau machen, wie man die Daten zu analysieren hat. Die relevanten Entscheidungen müssen übrigens *vor* der Datenerhebung getroffen werden, nicht erst nachher!

[2] Das Problem hierbei ist *nicht*, dass Kontrollvariablen in der Analyse berücksichtigt werden, sondern dass man sie nur dann verwendet, wenn sie zu einem signifikanten Ergebnis führen. Wichtige Kontrollvariablen in der Analyse zu berücksichtigen, ist eine gute Idee, nur soll die Entscheidung, welche Kontrollvariablen man berücksichtigt, nicht von den Ergebnissen abhängen.

Teil V

Übers allgemeine lineare Modell hinaus

Kapitel 17

Binäre Daten auswerten: Logistische Regression

Das allgemeine lineare Modell – das Werkzeug, das wir in den vorangehenden Kapiteln immer wieder verwendet haben – dient zur Auswertung kontinuierlicher *outcomes*. Oft haben wir es aber mit kategorischen *outcomes* zu tun, etwa mit binären Daten. Beispiele solcher Daten sind:

- Korrektheit, etwa ob eine Übersetzung als richtig oder als falsch gilt;
- die Realisierung des /r/-Phonems als [R] oder [r];
- die An- bzw. Abwesendheit eines Morphems;
- ob nach *wegen* eine Dativ- oder Genitivform kommt, usw.

Analysiert man Daten dieser Art in einem allgemeinen linearen Modell (z.B., indem man eine Ausprägung der Variablen als 0 bezeichnet und die andere als 1), dann ergibt sich eine Reihe von Komplikationen (siehe Jaeger, 2008). Ein besonders auffälliges Problem ist, dass ein lineares Modell mit kontinuierlichen Prädiktoren unmögliche Daten vorhersagen kann, nämlich Werte unter 0 oder grösser als 1. Des Weiteren basiert die defaultmässige Inferenzstatistik (p-Werte, Konfidenzintervalle) auf den Annahmen der Normalität und Homoskedastizität der Residuen; beide Annahmen sind *garantiert* verletzt, wenn die abhängige Variable binär ist:

- Die Residuen für Beobachtungen mit dem gleichen vorhergesagten Wert sind binär verteilt: Wird der Wert $\widehat{y_i}$ (z.B. 0.43) vorhergesagt, ist der Restfehler entweder $1 - \widehat{y_i}$ (z.B. 0.57) oder $0 - \widehat{y_i}$ (z.B. -0.43). Die Annahme, dass die y-Werte normalverteilt um jeden $\widehat{y}$-Wert liegen (siehe Abbildung 8.15 auf Seite 123), ist also unbegründet.
- Wenn eine Reihe binärer Beobachtungen, die als 0 oder 1 kodiert wurden, ein Mittel von 0 hat, dann gibt es keine Streuung in der Gruppe. Auch wenn sie ein Mittel von 1 hat, gibt es keine Streuung in der Gruppe. Die Streuung nimmt aber stetig zu, je näher das Gruppenmittel bei 0.50 liegt:

```
var(c(rep(0, 0), rep(1, 10))) # 0*0, 10*1
## [1] 0
var(c(rep(0, 2), rep(1, 8)))  # 2*0, 8*1, etc.
## [1] 0.1777778
var(c(rep(0, 3), rep(1, 7)))
## [1] 0.2333333
var(c(rep(0, 5), rep(1, 5)))
## [1] 0.2777778
var(c(rep(0, 7), rep(1, 3)))
## [1] 0.2333333
var(c(rep(0, 8), rep(1, 2)))
## [1] 0.1777778
var(c(rep(0, 10), rep(1, 0)))
## [1] 0
```

Wenn die Mittel der Gruppen unterschiedlich sind, ist die Variabilität also auch unterschiedlich. Die Annahme der Homoskedastizität ist also unbegründet.

Die beiden letzten Probleme könnte man allenfalls noch in einem allgemeinen linearen Modell zu lösen versuchen, indem man sich einer Bootstrapmethode bedient, aber eine allgemeiner gültige Lösung besteht darin, dass allgemeine lineare Modell so anzupassen, dass es auch mit nicht-kontinuierlichen *outcomes* umgehen kann. Dies ergibt das **verallgemeinerte lineare Modell** (*generalized linear model* – was leicht verwechselbare Begriffe betrifft, kennen Generativisten in Statistikern ihresgleichen).

Das verallgemeinerte lineare Modell hat mehrere Erscheinungsformen, die wir hier nicht alle behandeln werden (siehe Faraway, 2006). Für binäre Daten ist die wohl gängigste Erscheinungsform das **logistische Regressionsmodell**, dem die Einführung in diesem Kapitel gewidmet ist. Referenzen zu weiterführender Literatur finden Sie im nächsten Kapitel.[1]

[1] Den χ^2-Test behandeln wir hier nicht, da logistische Regression wesentlich flexibler ist. Ausserdem halte ich es für sinnvoller, dass man sich über das *Modellieren* von Daten Gedanken macht als dass man eine Reihe von Signifikanztests lernt.

17.1 Ein kategorischer Prädiktor

17.1.1 Daten

Tversky & Kahneman (1981) legten ihren Versuchspersonen ein hypothetisches Szenario vor und baten sie, zwischen zwei möglichen Aktionen zu wählen:

> Imagine that the U.S. is preparing for the outbreak of a an unusual Asian disease, which is expected to kill 600 people. Two alternative programs to combat the disease have been proposed. Assume that the exact scientific estimate of the consequences of the programs are as follows:

Bei etwa der Hälfte der Versuchspersonen wurden die Konsequenzen der Programme wie folgt formuliert (*gain framing*):

> If Program A is adopted, 200 people will be saved.
>
> If Program B is adopted, there is a 1/3 probability that 600 people will be saved, and a 2/3 probability that no people will be saved.

Bei den anderen Versuchspersonen war die Formulierung wie folgt (*loss framing*):

> If Program A is adopted, 400 people will die.
>
> If Program B is adopted, there is a 1/3 probability that nobody will die, and a 2/3 probability that 600 people will die.

Die Konsequenz von Programm A ist, egal wie sie formuliert wird, identisch, und das gleiche gilt für Programm B. Ausserdem sind die Erwartungswerte von Programmen A und B einander gleich: Im Schnitt sterben jeweils 400 Leute.

Die Ergebnisse waren wie folgt:

```
d <- data.frame(
  Framing = c("gain", "loss"),
  ProgrammA = c(109, 34),
  ProgrammB = c(43, 121)
)
d
```

```
##   Framing ProgrammA ProgrammB
## 1    gain       109        43
## 2    loss        34       121
```

Die Frage ist nun, ob sich die Präferenzen der Versuchspersonen für Programm A oder B ändern, wenn diese Konsequenzen dieser Programme anders (aber logisch identisch) beschrieben werden.

17.1.2 Proportionen und Wahrscheinlichkeiten

Berechnen wir für die beiden Konditionen die Proportion der Entscheide für das Programm, das Sicherheit bietet (Programm A):

```
d <- d %>%
  mutate(prop_sicher = ProgrammA / (ProgrammA + ProgrammB))
d

##   Framing ProgrammA ProgrammB prop_sicher
## 1    gain       109        43   0.7171053
## 2    loss        34       121   0.2193548
```

Diese Proportionen können wir natürlich auch grafisch darstellen:

```
# Diese Grafik wird im Skript nicht dargestellt.
ggplot(d, aes(x = Framing, y = prop_sicher)) +
  geom_point() +
  ylim(0, 1) +
  ylab("Proportion Wahl für sicheres Ergebnis") +
  coord_flip()
```

Wenn wir über keine weiteren Informationen verfügen, sind diese Proportionen unsere besten Schätzungen der Wahrscheinlichkeit, zu der die *gain*- bzw. die *loss*-Formulierung eine Wahl für das Programm A auslöst. Es sind aber lediglich Schätzungen: Eine andere Stichprobe würde andere Proportionen ergeben.

17.1.3 Chancen und Chancenverhältnisse

Statt in Proportionen kann man die Daten auch in Chancen (*odds*) ausdrücken. Eine Chance sagt, wie viel wahrscheinlicher ein Ergebnis als ein anderes ist. Auf der Basis dieser Stichprobe können wir schätzen, dass eine Wahl für Programm A beim *gain*-Framing etwa 2.5 Mal so wahrscheinlich ist als eine Wahl für Programm B: $\frac{109}{43} = 2.53$. Anders gesagt: für jede Wahl für Programm B gibt es 2.5 Wahlen für Programm A. Beim *loss*-Framing ist eine Wahl für A etwa 0.3 Mal so wahrscheinlich als eine Wahl für B: $\frac{34}{121} = 0.28$. Anders gesagt, für jede Wahl für B gibt es nur etwa 0.3 Wahlen für A.

```
d <- d %>%
  mutate(odds = prop_sicher / (1 - prop_sicher))
d

##   Framing ProgrammA ProgrammB prop_sicher      odds
## 1    gain       109        43   0.7171053 2.5348837
## 2    loss        34       121   0.2193548 0.2809917
```

Die Chance, dass man A statt B wählt, ist beim *gain*-Framing also etwa 9 Mal so gross als beim *loss*-Framing ($\frac{2.53}{0.28} = 9.0$). Diese Zahl nennt man das Chancenverhältnis (*odds*

ratio). Umgekehrt gilt, dass die Chance, dass man A statt B wählt, beim *loss*-Framing etwa 0.11 Mal so gross ist als beim *gain*-Framing: ($\frac{0.28}{2.53} = 0.11$).

In diesem Beispiel kann man all diese Zahlen leicht von Hand berechnen. Bei etwas komplizierteren Datensätzen (z.B. in Datensätzen mit kontinuierlichen Prädiktoren) ist dies aber nicht mehr der Fall; dazu braucht man dann ein statistisches Modell. Darum schauen wir jetzt, wie wir mit einem logistischen Regressionsmodell diese Proportionen, Chancen und Chancenverhältnisse berechnen können.

17.1.4 Modellierung: Möglichkeit 1

Für einfache Datensätze kann man einen Datensatz konstruieren, der die Antwortmuster pro Kondition zusammenfasst. Dies haben wir oben schon gemacht. Diese Daten können wie folgt in ein logistisches Regressionsmodell gegossen werden:

```
tversky.glm <- glm(cbind(ProgrammA, ProgrammB) ~ Framing,
                   data = d, family = binomial(link = "logit"))
```

Statt `lm()` wird `glm()` (*generalized linear model*) verwendet. Vor der Tilde kommen die Namen der Spalten, die die Anzahl Beobachtungen der beiden Ausprägungen enthalten; diese werden mit `cbind()` kombiniert. Die Prädiktoren kommen wie üblich nach der Tilde.

Neu ist, dass ein `family`-Parameter spezifiziert werden muss. Wir gehen hier davon aus, dass die Daten aus einer Binomialverteilung stammen. Das Kästchen erklärt, was das bedeutet. Was die Einstellung `link = "logit"` bedeutet wird in Kürze erklärt.

Binomialverteilungen. Wenn die Wahrscheinlichkeit, zu der sich eine Versuchsperson in der Studie von Tversky & Kahneman (1981) für Programm A entscheidet, bei $p = 0.4$ läge, wie wahrscheinlich wäre es dann, dass von vier Versuchspersonen drei sich für Programm A entscheiden?

Es gibt vier Möglichkeiten, wie sich dieses Szenario ereignen kann. Jede dieser vier Möglichkeiten ist gleich wahrscheinlich:

- Person 1: A, Person 2: A, Person 3: A, Person 4: B: $0.4 \times 0.4 \times 0.4 \times (1-0.4) = 0.4^3 \times (1-0.4)^1 = 0.0384$
- Person 1: A, Person 2: A, Person 3: B, Person 4: A: $0.4 \times 0.4 \times (1-0.4) \times 0.4 = 0.4^3 \times (1-0.4)^1 = 0.0384$
- Person 1: A, Person 2: B, Person 3: A, Person 4: A: $0.4 \times (1-0.4) \times 0.4 \times 0.4 = 0.4^3 \times (1-0.4)^1 = 0.0384$
- Person 1: B, Person 2: A, Person 3: A, Person 4: A: $(1-0.4) \times 0.4 \times 0.4 \times 0.4 = 0.4^3 \times (1-0.4)^1 = 0.0384$

Die Wahrscheinlichkeit, zu der irgendeine dieser Möglichkeiten zutrifft, liegt also

bei $4 \times 0.4^3 \times (1 - 0.4)^1 = 0.1536$.

Machen wir die gleiche Übung für ein anderes Szenario: Wenn $p = 0.7$, wie wahrscheinlich wäre es dann, dass 3 von 5 Personen sich für Programm A entscheiden? Eine Möglichkeit, wie sich dieses Szenario ereignen kann, ist diese:

- Person 1: A, Person 2: A, Person 3: A, Person 4: B, Person 5: $0.7 \times 0.7 \times 0.7 \times (1 - 0.7) \times (1 - 0.7) = 0.7^3 \times (1 - 0.7)^2 \approx 0.031$

Jede andere Möglichkeit ist gleich wahrscheinlich, sodass wir dieses Ergebnis ($0.7^3 \times (1-0.7)^2$) nur mit der Anzahl Möglichkeiten zu multiplizieren haben. Diese Anzahl lässt sich wie folgt berechnen:

```
choose(n = 5, k = 3)
```

```
## [1] 10
```

Die `choose()`-Funktion berechnet, wie viele Möglichkeiten es gibt, k unterschiedliche Elemente aus n Elementen zu ziehen. Die Berechnung, die hinter ihr liegt, ist diese:

$$\frac{n!}{k!(n-k)!} = \frac{n \times (n-1) \times (n-2) \times \cdots \times 1}{(k \times (k-1) \cdots \times 1)((n-k) \times (n-k-1) \times \cdots \times 1)} \tag{17.1}$$

Für $n = 5$ und $k = 3$ ergibt dies $\frac{5 \times 4 \times 3 \times 2 \times 1}{(3 \times 2 \times 1)(2 \times 1)} = 10$

Insgesamt liegt die Wahrscheinlichkeit, zu der drei aus fünf Personen Programm A wählen (wenn $p = 0.7$) also bei $10 \times 0.7^3 \times (1 - 0.7)^2 = 0.3087$.

Aus diesen Beispielen können wir eine allgemeine Regel distillieren: Um die Wahrscheinlichkeit ('Pr'), dass ein Ereignis genau k von n Mal zutrifft, wenn jedes Ereignis eine Wahrscheinlichkeit von p hat, zu erhalten, multipliziert man die Anzahl Möglichkeiten, dass dies passieren kann, mit der Wahrscheinlichkeit dieser Möglichkeiten:

$$Pr(k; n, p) = \frac{n!}{k!(n-k)!} \times p^k \times (1-p)^{(n-k)} \tag{17.2}$$

Die Binomialverteilung drückt für alle Möglichkeiten für k die Wahrscheinlichkeit Pr(k; n, p) aus. Mit der R-Funktion `dbinom()` können wir diese schnell berechnen:

```
dbinom(x = 3, size = 4, prob = 0.4)
```

```
## [1] 0.1536
```

```
dbinom(x = 3, size = 5, prob = 0.7)
```

```
## [1] 0.3087
```

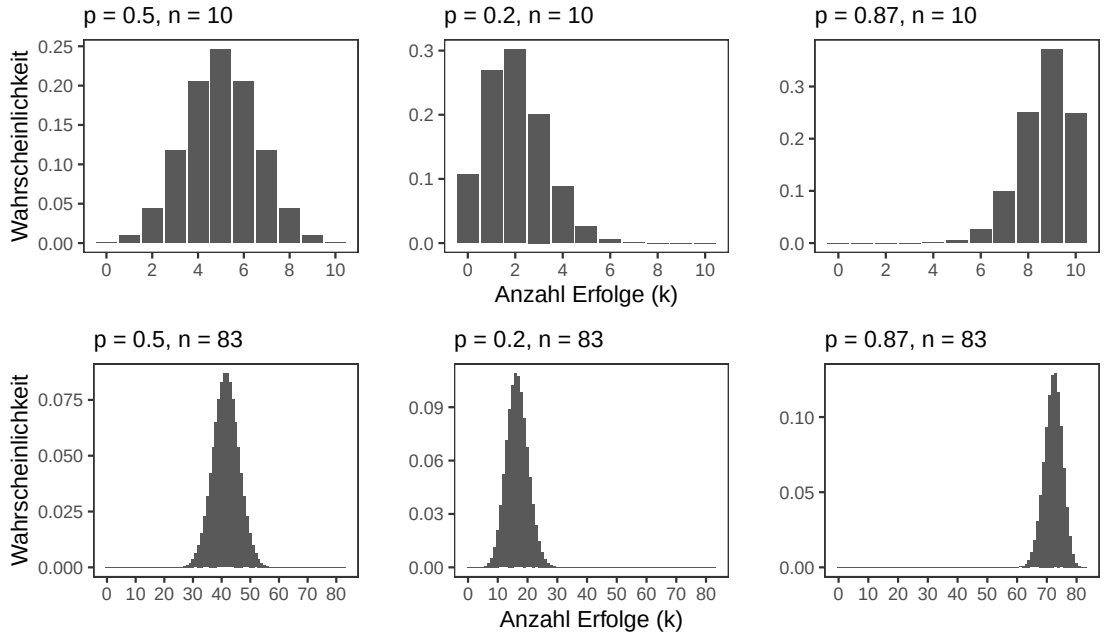


Abbildung 17.1: *Obere Zeile:* Drei Binomialverteilungen mit $n = 10$. *Untere Zeile:* Drei Binomialverteilungen mit $n = 83$.

Abbildung 17.1 zeigt sechs Beispiele von Binomialverteilungen (mit unterschiedlichen p- und n-Werten).

Bemerken Sie, dass wir in diesen Beispielen davon ausgehen, dass p konstant ist und die Ereignisse voneinander unabhängig sind. Wenn es etwa so gewesen wäre, dass die Wahl von Person 2 von der Wahl von Person 1 abhängt, dann wären beide Ereignisse nicht länger unabhängig voneinander.

Die Parameterschätzung des Modells und ihre geschätzten Standardfehler können wie gehabt abgerufen werden:

```
summary(tversky.glm)

##
## Call:
## glm(formula = cbind(ProgrammA, ProgrammB) ~ Framing, family = binomial(link = "logi
##     data = d)
##
## Deviance Residuals:
## [1]  0  0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9301     0.1801   5.165  2.4e-07
## Framingloss  -2.1996     0.2648  -8.307  < 2e-16
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7.9984e+01  on 1  degrees of freedom
## Residual deviance: 1.5099e-14  on 0  degrees of freedom
## AIC: 14.393
##
## Number of Fisher Scoring iterations: 3
```

Diesmal werden keine t-Werte, sondern z-Werte gezeigt. Diese werden aber identisch berechnet; der feine Unterschied ist lediglich, dass die p-Werte in der letzten Spalte nicht auf t-Verteilungen, sondern auf der Standardnormalverteilung (einer Normalverteilung mit $\mu = 0$ und $\sigma^2 = 1$) basieren. Insbesondere bei kleinen Stichproben verstehen sich diese p-Werte als Annäherungen.[2]

Auf dem ersten Blick haben die geschätzten Parameter aber nichts mit den Zahlen, die wir oben berechnet haben, zu tun. Der Grund dafür ist, dass diese Parameter in **log-odds** ausgedrückt werden. Log-odds sind logarithmisch transformierte Wahrscheinlichkeitsquoten:[3]

$$\text{log-odds} = \ln \text{odds} = \ln \frac{\text{Proportion A}}{\text{Proportion B}} = \ln \frac{\text{Proportion A}}{1 - \text{Proportion A}} \tag{17.3}$$

Diese Funktion – die Proportionen zu log-odds transformiert – heisst die logit-Funktion: $\text{logit}(x) = \ln \frac{x}{1-x}$, was das `link = "logit"` im `glm()`-Befehl erklärt.

Log-odds vereinfachen das Rechnen hinter den Kulissen erheblich, sind aber schwieriger zu interpretieren. Abbildung 17.2 zeigt die Zusammenhänge zwischen Wahrscheinlichkeiten, Odds und log-odds. Diese Grafiken können wir verwenden, um herauszufinden, was die Parameterschätzungen genau bedeuten.

- Die Parameterschätzung des Intercepts beträgt 0.93 log-odds. Der Grafik links unten in Abbildung 17.2 können wir entnehmen, dass dies odds von etwa 2.5 entspricht. Der genaue Wert kann mit `exp()` berechnen werden; dies ist die Umkehrfunktion der logarithmen Funktion:

  ```
  exp(0.93)

  ## [1] 2.534509
  ```

[2] Je komplexer die Modelle, desto approximativer die Inferenzen:

> [I]t is striking that as model flexibility increases, so that the models become better able to describe the reality that we believe generated a set of data, so the methods for inference become less well founded. The linear model class is quite restricted, but within it, hypothesis testing and interval estimation are exact, while estimation is unbiased. For the larger class of GLMs this exactness is generally lost in favour of ... large sample approximations ..., while estimators themselves are consistent, but not necessarily unbiased. (Wood, 2006, xvi–xvii)

[3] Wenn $x^y = z$, ist $\log_x z = y$. Beispiele: $10^2 = 100$, also $\log_{10} 100 = 2$; $2^5 = 32$, also $\log_2 32 = 5$. ln ist die Kürzel für den sog. natürlichen Logarithmus, d.h., $\log_e$, wo $e \approx 2.71828$.

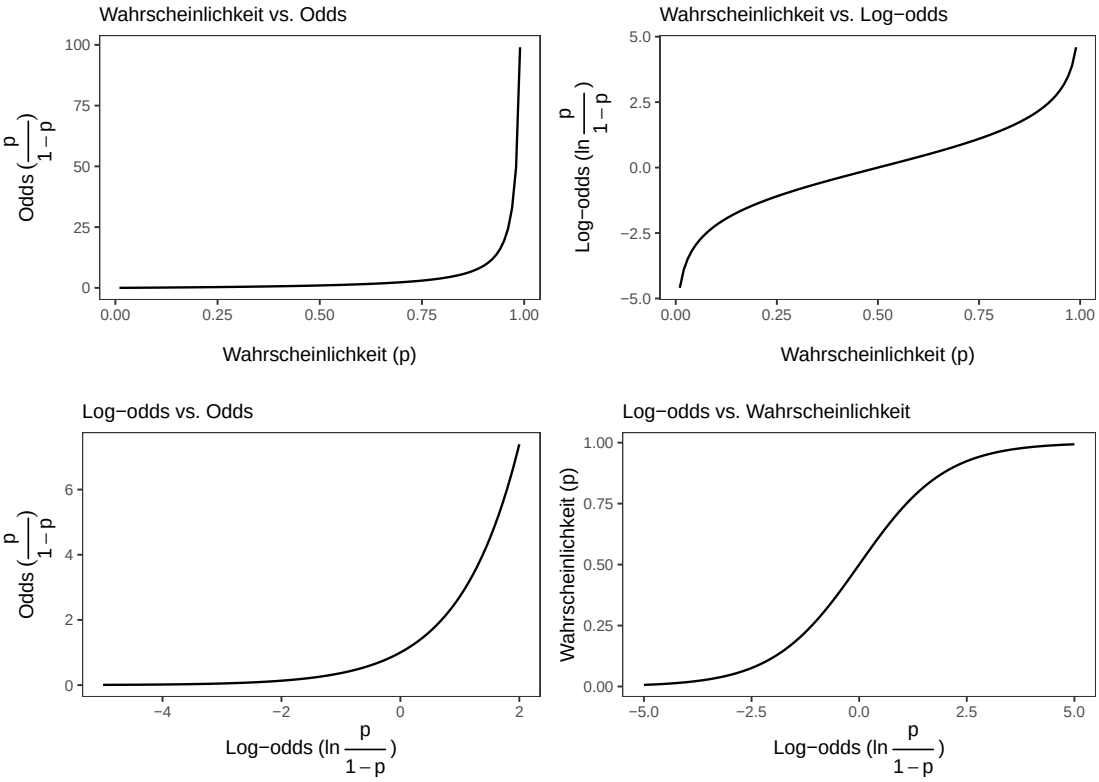


Abbildung 17.2: Odds, log-odds und Wahrscheinlichkeiten zueinander konvertieren.

Oben hatten wir schon berechnet, dass es bei der *gain*-Framing 2.5 Wahlen für Programm A für jede Wahl für Programm B gab. Diese Parameterschätzung betrifft also die Präferenzen in der *gain*-Framing-Kondition. Das Intercept betrifft die *gain*-Framing-Kondition, da diese alphabetisch vor der *loss*-Framing-Kondition kommt (siehe Seite 139), und die Zahl betrifft jeweils die Wahl für Programm A statt für Programm B, da wir zuerst die Spalte `ProgrammA` ins Modell eingetragen haben.

- Eine Chance von 2.5-zu-1 bzw. log-odds von 0.93 entsprechen eine Wahrscheinlichkeit von etwa 0.72. Diese Zahl kann man mit der `plogis()`-Funktion berechnen:

```
plogis(0.93)

## [1] 0.7170753
```

Dies ist die oben berechnete geschätzte Wahrscheinlichkeit, dass eine Versuchsperson in der *gain*-Framing-Kondition für Programm A entscheidet.

- Die Parameterschätzung für `Framingloss` beträgt −2.2 log-odds. Konvertiert zu odds ergibt dies 0.11:

```
exp(-2.2)

## [1] 0.1108032
```

Dies ist das Chancenverhältnis, das wir oben berechnet haben: Das Modell schätzt, dass die Chance, dass sich jemand in der *loss*-Kondition für Programm A entscheidet, 0.11 so gross ist als die Chance, dass sich jemand in der *gain*-Kondition für Programm A entscheidet.

- Eine Chance kann man zwar zu einer Wahrscheinlichkeit konvertieren, ein Chancen*verhältnis* jedoch nicht.
- In log-odds ausgedrückt beträgt die 'Wahrscheinlichkeit', dass sich jemand in der *loss*-Kondition für Programm A entscheidet $0.93 + (-2.2) = -1.27$. Hierzu muss man nur die Parameterschätzungen beieinander aufzählen, genauso wie wir bei linearen Modellen gemacht haben.
- In odds ausgedrückt ergibt dies 0.28. Dies ist der gleiche Wert, den wir schon von Hand berechnet haben:

```
exp(0.93 - 2.2)
```

```
## [1] 0.2808316
```

- Dies entspricht einer modellierten Wahrscheinlichkeit von 0.22; das hatten wir auch schon von Hand berechnet.

```
plogis(0.93 - 2.2)
```

```
## [1] 0.2192573
```

Der Mehrwert von logistischer Regression ist vorübergehend minimal, denn all diese Schätzungen können auch von Hand berechnet werden. Was nützlich ist, ist, dass das Modell auch geschätzte Standardfehler ausspuckt und das Berechnen von Konfidenzintervallen ermöglicht:[4]

```
tversky_ki <- confint(tversky.glm)
```

```
## Waiting for profiling to be done...
```

```
tversky_ki
```

```
##                 2.5 %    97.5 %
## (Intercept)  0.5852679  1.293136
## Framingloss -2.7308428 -1.691284
```

Verlieren Sie aber nicht aus dem Auge, dass auch diese Standardfehler und Konfidenzintervalle in *log-odds* ausgedrückt werden. In Tat und Wahrheit kann kaum jemand solche log-odds direkt interpretieren. **Am besten berichten Sie die Er-**

[4] Wenn diese Zahlen bei Ihnen leicht anders aussehen, müssen Sie noch das `MASS`-Paket installieren. Sie brauchen es aber nicht zu laden.

gebnisse eines logistischen Regressionsmodell daher auch in odds und/oder in Wahrscheinlichkeiten!

Das 95%-Konfidenzintervall für das Chancenverhältnis von 0.11 können wir leicht berechnen, indem wir das betreffende Konfidenzintervall zu odds konvertieren:

```
# steht in der zweiten Zeile
exp(tversky_ki[2, ])
```

```
##      2.5 %     97.5 %
## 0.06516435 0.18428282
```

Wenn sich die Antwortpräferenz nicht zwischen den Konditionen unterscheiden sollte (was meistens der Nullhypothese entspricht), wäre dieses Chancenverhältnis etwa 1. Diese Daten deuten aber sehr stark darauf hin, dass sich das Chancenverhältnis von 1 unterscheidet.

Das Chancenverhältnis ist wohl die wichtige Erkenntniss der Studie, aber wenn wir wollen, können wir auch das 95%-Konfidenzintervall für die Chance von 2.5 (Präferenz der *gain*-Gruppe) berechnen:

```
exp(tversky_ki[1, ])
```

```
##    2.5 %   97.5 %
## 1.795472 3.644196
```

Mit `plogis()` kann auch das Konfidenzintervall für die geschätzte Wahrscheinlichkeit von 0.72 berechnet werden:

```
plogis(tversky_ki[1, ])
```

```
##     2.5 %    97.5 %
## 0.6422786 0.7846775
```

Aufgabe Das Konfidenzintervall um die geschätzten Präferenzen der Versuchspersonen in der *loss*-Kondition (sowohl für die Chance als auch für die Wahrscheinlichkeit) können nicht direkt aus diesem Modell abgeleitet werden. Dazu müssten Sie dafür sorgen, dass die *loss*- statt der *gain*-Kondition vom Intercept erfasst wird. Versuchen Sie, dies hinzukriegen.

17.1.5 Modellierung: Möglichkeit 2

Wenn man die Daten nicht in einem Datensatz wie `d` zusammengefasst hat oder sie nicht so zusammenfassen kann (z.B. weil kontinuierliche Prädiktoren mit im Spiel sind), kann man die logistische Regression oberflächlich leicht anders berechnen. Der Datensatz `d_anders`, den wir unten kreieren, enthält die genau gleichen Informationen wie Datensatz `d`, nur listet er die individuellen Wahlen auf:

```
d_anders <- data.frame(
  Framing = c(rep("gain", 109 + 43),
              rep("loss", 34 + 121)),
  Wahl = c(rep("ProgrammA", 109), rep("ProgrammB", 43),
           rep("ProgrammA", 34), rep("ProgrammB", 121))
)
# 1, wenn Programm A; 0, wenn Programm B
d_anders$sicher <- ifelse(d_anders$Wahl == "ProgrammA", 1, 0)
# Zufällige Auswahl anzeigen
d_anders %>%
  sample_n(12)
```

```
##    Framing      Wahl sicher
## 1     loss ProgrammA      1
## 2     loss ProgrammB      0
## 3     gain ProgrammA      1
## 4     gain ProgrammB      0
## 5     loss ProgrammB      0
## 6     gain ProgrammA      1
## 7     loss ProgrammB      0
## 8     loss ProgrammA      1
## 9     loss ProgrammB      0
## 10    loss ProgrammB      0
## 11    loss ProgrammB      0
## 12    loss ProgrammB      0
```

Statt `glm()` zwei Spalten mit Anzahlen zu füttern, reicht es jetzt, einfach die Spalte mit Nullen und Einsen als *outcome* zu definieren:

```
tversky.glm <- glm(sicher ~ Framing,
                   data = d_anders, family = binomial(link = "logit"))
summary(tversky.glm)
```

```
##
## Call:
## glm(formula = sicher ~ Framing, family = binomial(link = "logit"),
##     data = d_anders)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5891  -0.7037  -0.7037   0.8155   1.7419
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9301     0.1801   5.165  2.4e-07
```

```
## Framingloss  -2.1996     0.2648  -8.307  < 2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 424.15  on 306  degrees of freedom
## Residual deviance: 344.17  on 305  degrees of freedom
## AIC: 348.17
##
## Number of Fisher Scoring iterations: 4
```

Was die Parameterschätzungen betrifft, ist dieses Modell dem Modell aus dem letzten Abschnitt gleich.

17.2 Mehrere kategorische Prädiktoren

Keysar et al. (2012, Experiment 1a) verwendeten die gleiche Aufgabe wie Tversky & Kahneman (1981), legten aber etwa der Hälfte der Versuchspersonen das Dilemma in ihrer Muttersprache (Englisch) und der anderen Hälfte in der Fremdsprache Japanisch vor. Sie interessierten sich dafür, ob der (logisch irrelevante) Unterschied in der Formulierung zwischen den *gain*- und *loss*-Konditionen einen kleineren Einfluss auf die Wahl der Versuchspersonen ausübt, wenn das Dilemma in einer Fremdsprache vorliegt. Wir lesen den Datensatz ein und lassen 12 willkürliche Zeilen anzeigen:

```
d <- read_csv("Data/Keysar2012_Exp1a.csv")
d %>%
  sample_n(12)
```

```
## # A tibble: 12 x 3
##    Sprache   Formulierung Wahl
##    <chr>     <chr>        <chr>
##  1 Englisch  Verlust      sicher
##  2 Japanisch Verlust      sicher
##  3 Englisch  Verlust      unsicher
##  4 Japanisch Verlust      unsicher
##  5 Japanisch Verlust      unsicher
##  6 Englisch  Gewinn       sicher
##  7 Englisch  Verlust      unsicher
##  8 Japanisch Verlust      sicher
##  9 Japanisch Gewinn       sicher
## 10 Englisch  Verlust      unsicher
## 11 Japanisch Verlust      unsicher
## 12 Englisch  Verlust      unsicher
```

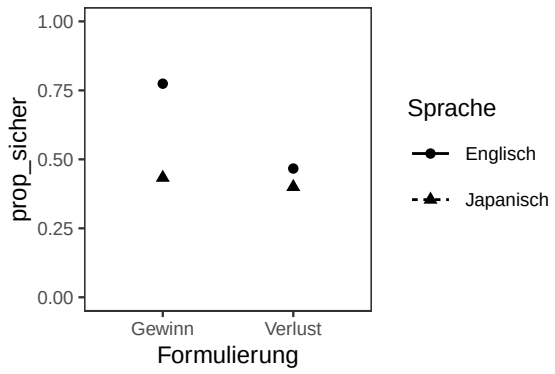
17.2.1 Numerische Zusammenfassung und Grafik

```
# Anzahlen und Proportionen pro Zelle
d_summary <- d %>%
  group_by(Sprache, Formulierung) %>%
  summarise(n = n(),
            n_sicher = sum(Wahl == "sicher"),
            n_unsicher = sum(Wahl == "unsicher"),
            prop_sicher = mean(Wahl == "sicher"))
d_summary
```

```
## # A tibble: 4 x 6
## # Groups:   Sprache [2]
##   Sprache Formulierung     n n_sicher n_unsicher prop_sicher
##   <chr>   <chr>        <int>    <int>      <int>       <dbl>
## 1 Englis~ Gewinn          31       24          7       0.774
## 2 Englis~ Verlust         30       14         16       0.467
## 3 Japani~ Gewinn          30       13         17       0.433
## 4 Japani~ Verlust         30       12         18       0.4
```

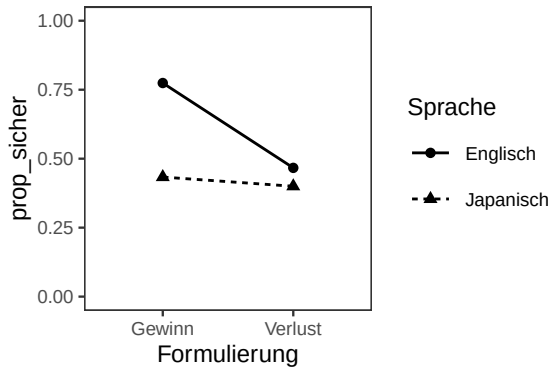
```
# Grafik: 1. Versuch
ggplot(d_summary,
       aes(x = Formulierung, y = prop_sicher,
           linetype = Sprache, shape = Sprache)) +
  geom_point() +
  geom_line() +
  ylim(0, 1)
```

```
## geom_path: Each group consists of only one observation.
## Do you need to adjust the group aesthetic?
```



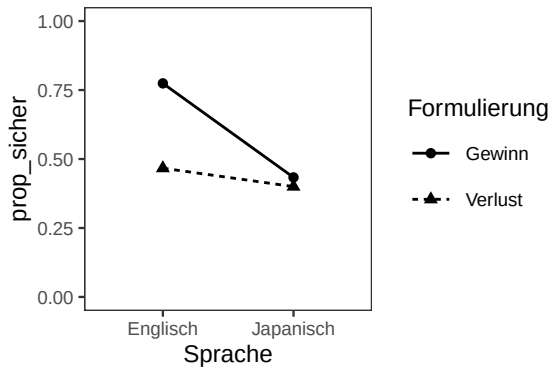
Die Warnung erklärt, wieso keine Linie zwischen den Punkten gezeichnet wurde: Die Funktion versteht nicht, zwischen welchen Punkten genau eine Linie gezeichnet werden soll, und braucht etwas Hilfe in der Form des `group`-Parameters:

```
# Grafik: 2. Versuch
ggplot(d_summary,
       aes(x = Formulierung, y = prop_sicher,
           linetype = Sprache, shape = Sprache,
           group = Sprache)) +
  geom_point() +
  geom_line() +
  ylim(0, 1)
```



Man kann natürlich auch die Faktoren Sprache und Formulierung in der Grafik umwechseln:

```
ggplot(d_summary,
       aes(x = Sprache, y = prop_sicher,
           linetype = Formulierung, shape = Formulierung,
           group = Formulierung)) +
  geom_point() +
  geom_line() +
  ylim(0, 1)
```



17.2.2 Was man *nicht* machen soll

Keysar et al. (2012) analysierten diese Daten mit zwei Signifikanztests. Der eine Test

zeigte, dass es bei den Versuchspersonen, die die Aufgabe auf Englisch erledigten, einen signifikanten Unterschied zwischen den *gain*- und *loss*-Konditionen gab, bei denen, die den Test auf Japanisch erledigten, jedoch nicht. Es ist aber eine äusserst schlechte Idee, Daten so zu analysieren, denn, wie Gelman & Stern (2006) es auf den Punkt bringen, "The difference between 'significant' and 'not significant' is not itself statistically significant." Siehe auch *Assessing differences of significant* (28.10.2014). Die Lösung besteht darin, sämtliche Daten in *einem* Modell zu analysieren.

17.2.3 Modellierung: Erster Versuch

Wir fügen zuerst dem Datensatz eine Variable hinzu, die 1 ist, wenn die Versuchsperson sich für die sichere Option entschieden hat, und 0, wenn nicht. Die beiden Prädiktoren werden dann im gleichen Modell berücksichtigt.

```
d$Sicher <- ifelse(d$Wahl == "sicher", 1, 0)
keysar.glm1 <- glm(Sicher ~ 1 + Formulierung + Sprache,
                   data = d, family = binomial(link = "logit"))
summary(keysar.glm1)
```

```
##
## Call:
## glm(formula = Sicher ~ 1 + Formulierung + Sprache, family = binomial(link = "logit"
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5645  -1.1840   0.8346   1.1147   1.4902
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.8756     0.3366   2.601  0.00929
## FormulierungVerlust  -0.7264     0.3787  -1.918  0.05511
## SpracheJapanisch     -0.8600     0.3787  -2.271  0.02316
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 167.53  on 120  degrees of freedom
## Residual deviance: 158.59  on 118  degrees of freedom
## AIC: 164.59
##
## Number of Fisher Scoring iterations: 4
```

```
confint(keysar.glm1)
```

```
## Waiting for profiling to be done...
```

```
##                            2.5 %       97.5 %
## (Intercept)           0.2343907  1.562385014
## FormulierungVerlust -1.4803837  0.009110024
## SpracheJapanisch    -1.6152795 -0.125899308
```

Die Parameterschätzungen sind wie folgt zu interpretieren:

- Das Intercept erfasst die Präferenzen der Versuchspersonen, die die Referenzgruppe ausmachen. Hier handelt es sich um die Versuchspersonen in der *gain framing*-Kondition, die die Aufgabe auf Englisch erledigten. Dem Modell zufolge ist es 2.41 Mal so wahrscheinlich, dass diese sich für die sichere Option entscheiden als dass sie sich für die unsichere Option entscheiden. Bei dieser Zahl handelt es sich also nicht um ein Chancenverhältnis, sondern um eine Chance.

  ```
  exp(0.88)
  ```

  ```
  ## [1] 2.4109
  ```

- Der geschätzte Koeffizient für `SpracheJapanisch` erfasst den Unterschied zwischen den Präferenzen der Versuchspersonen, die die Aufgabe auf Japanisch erledigten, und denen der Versuchspersonen, die die Aufgabe auf Englisch erledigten. Der Koeffizient (-0.86) wird in log-odds ausgedrückt. Exponiert man diese Zahl, erhält man das entsprechende Chancenverhältnis:

  ```
  exp(-0.86)
  ```

  ```
  ## [1] 0.4231621
  ```

  Dem Modell zufolge ist es also 0.42 Mal so wahrscheinlich, dass jemand in der Japanischgruppe die sichere Option wählt als dass jemand in der Englischgruppe die sichere Option wählt. Oder umgekehrt: Es ist $\exp(0.86) = 2.36$ Mal so wahrscheinlich, dass jemand in der Englischgruppe die sichere Option wählt als dass jemand der Japanischgruppe die sichere Option wählt.

- Der geschätzte Koeffizient für `FormulierungVerlust` erfasst den Unterschied zwischen den Präferenzen der Versuchspersonen, in der *loss framing*-Kondition und denen der Versuchspersonen in der *gain framing*-Kondition. Das Chancenverhältnis beträgt:

  ```
  exp(-0.73)
  ```

  ```
  ## [1] 0.481909
  ```

  Dem Modell zufolge ist es also 0.48 Mal so wahrscheinlich, dass jemand in der *loss framing*-Kondition die sichere Option wählt als dass jemand in der *gain framing*-Kondition die sichere Option wählt.

Das Modell `keysar.glm1` hilft uns bei der Beantwortung unserer Forschungsfrage leider keinen Schritt weiter: Wir wollten wissen, ob sich der Einfluss von *gain*- vs. *loss*-Framing ändert, wenn die Aufgabe in einer Fremdsprache statt in der Erstsprache vor-

liegt. Wir interessieren uns also für die Interaktion zwischen Sprache und Framing, sodass wir diese mitmodellieren müssen.

17.2.4 Modellierung: Zweiter Versuch

Wir fügen dem Modell die Interaktion zwischen Sprache und Formulierung hinzu.

```
keysar.glm2 <- glm(Sicher ~ 1 + Formulierung + Sprache +
                     Formulierung:Sprache,
                   data = d, family = binomial(link = "logit"))
summary(keysar.glm2)
```

```
##
## Call:
## glm(formula = Sicher ~ 1 + Formulierung + Sprache + Formulierung:Sprache,
##     family = binomial(link = "logit"), data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7251  -1.0658   0.7155   1.2346   1.3537
##
## Coefficients:
##                                     Estimate Std. Error
## (Intercept)                           1.2321     0.4296
## FormulierungVerlust                  -1.3657     0.5643
## SpracheJapanisch                     -1.5004     0.5659
## FormulierungVerlust:SpracheJapanisch   1.2285     0.7701
##                                     z value Pr(>|z|)
## (Intercept)                           2.868  0.00413
## FormulierungVerlust                  -2.420  0.01552
## SpracheJapanisch                     -2.651  0.00802
## FormulierungVerlust:SpracheJapanisch   1.595  0.11067
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 167.53  on 120  degrees of freedom
## Residual deviance: 156.01  on 117  degrees of freedom
## AIC: 164.01
##
## Number of Fisher Scoring iterations: 4
```

```
confint(keysar.glm2)
```

```
## Waiting for profiling to be done...
```

```
##                                         2.5 %     97.5 %
```

```
## (Intercept)                              0.4424004  2.1538204
## FormulierungVerlust                     -2.5222681 -0.2905951
## SpracheJapanisch                        -2.6621249 -0.4247117
## FormulierungVerlust:SpracheJapanisch -0.2670673  2.7646464
```

Die Parameterschätzungen sind wie folgt zu interpretieren:

- Das Intercept erfasst die Präferenzen der Versuchspersonen, die die Referenzgruppe ausmachen. Hier handelt es sich um die Versuchspersonen in der *gain framing*-Kondition, die die Aufgabe auf Englisch erledigten. Dem Modell zufolge ist es 3.42 Mal so wahrscheinlich, dass diese sich für die sichere Option entscheiden als dass sie sich für die unsichere Option entscheiden. Bei dieser Zahl handelt es sich also nicht um ein Chancenverhältnis, sondern um eine Chance.

```
exp(1.2321)
```

```
## [1] 3.428422
```

- Der geschätzte Koeffizient für `FormulierungVerlust` vergleicht die Referenzgruppe ('Englisch und Gewinn') mit der Gruppe 'Englisch und Verlust'. Das Chancenverhältnis beträgt:

```
exp(-1.3657)
```

```
## [1] 0.255202
```

  Dem Modell zufolge ist es also 0.26 Mal so wahrscheinlich, dass jemand in der Gruppe 'Englisch und Verlust' die sichere Option wählt als dass jemand in der Gruppe 'Englisch und Gewinn' die sichere Option wählt. Die gleiche Zahl erhalten wir, wenn wir uns Tabelle `d_summary` anschauen:

$$\frac{14/16}{24/7} = 0.26$$

- Der geschätzte Koeffizient für `SpracheJapanisch` vergleicht die Referenzgruppe ('Englisch und Gewinn') mit der Gruppe 'Japanisch und Verlust'. Das Chancenverhältnis beträgt:

```
exp(-1.5004)
```

```
## [1] 0.2230409
```

  Dem Modell zufolge ist es also 0.22 Mal so wahrscheinlich, dass jemand in der Gruppe 'Japanisch und Gewinn' die sichere Option wählt als dass jemand in der Gruppe 'Englisch und Gewinn' die sichere Option wählt. Die gleiche Zahl erhalten wir, wenn wir uns Tabelle `d_summary` anschauen:

$$\frac{13/17}{24/7} = 0.22$$

- Der geschätzte Interaktionskoeffizient drückt aus, wie stark sich der Einfluss des Framings je nach der Sprache unterscheidet. (Oder, anders ausgedrückt, aber mathematisch identisch: wie stark sich der Einfluss der Sprache je nach dem Framing unterscheidet.) In log-odds ausgedrückt ist dieser Koeffizient schwierig zu interpretieren, sodass wir auch diese Zahl exponieren:

```
exp(1.2285)

## [1] 3.416102
```

Was heisst nun diese 3.42? Den Einfluss des Framings bei den Versuchspersonen, die die Aufgabe auf Englisch erledigten, können wir in einem Chancenverhältnis (odds ratio; OR) ausdrücken:

$$\text{OR für Englisch} = \frac{14/16}{24/7} = 0.2552083$$

Ich ründe dieses Zwischenergebnis hier ab nicht, sodass das Endergebnis nicht vom Ründungsfehler betroffen wird.

Ebenso können wir den Einfluss des Framings bei den Versuchspersonen, die die Aufgabe auf Englisch erledigten, in einem Chancenverhältnis ausdrücken:

$$\text{OR für Japanisch} = \frac{12/18}{13/17} = 0.8717949$$

Das Verhältnis dieser Verhältnisse sagt uns, wie viel stärker der Einfluss des Framings auf Japanisch als auf Englisch ist:

$$\frac{\text{OR für Japanisch}}{\text{OR für Englisch}} = \frac{0.8717949}{0.2552083} = 3.42$$

Dies ist die Zahl, die wir erhalten, wenn wir den Interaktionskoeffizienten exponieren. Wenn sich das Chancenverhältnis nicht je nach Sprache unterscheidet, soll das Verhältnis dieser Verhältnisse bei 1 liegen. Auf der Basis des 95%-Konfidenzintervalls um diese 3.42 herum ([0.77, 15.9]) können wir noch nicht schlussfolgern, dass die Daten nicht mit einem identischen Chancenverhältnis für beide Sprachen kompatibel sind.

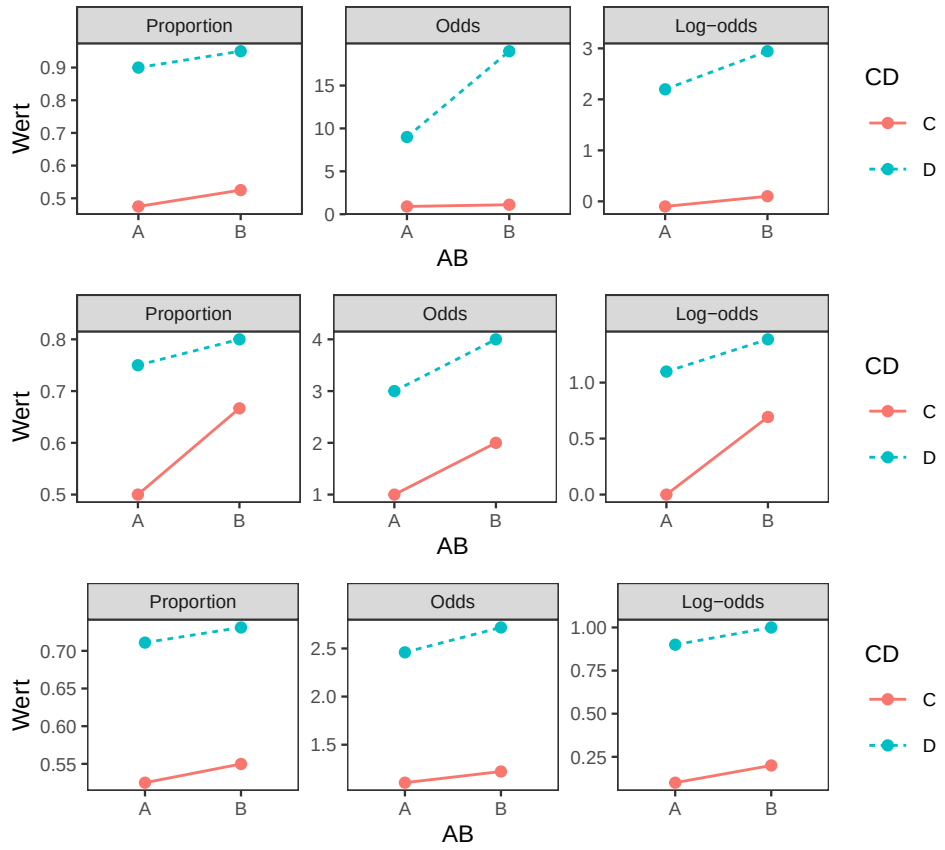


Abbildung 17.3: Ob man schlussfolgert, dass eine Interaktion vorliegt, hängt in diesen drei Beispielen davon ab, wie man die Daten ausdrückt: als Proportionen, als Odds (Chancen) oder als Log-Odds.

```
exp(confint(keysar.glm2)[4, ])

## Waiting for profiling to be done...

##      2.5 %     97.5 %
##  0.7656215 15.8734254
```

Die Interpretation der geschätzten Regressionskoeffizienten gestaltet sich bei logistischen Regressionsmodellen also deutlich schwieriger als beim allgemeinen linearen Modell, insbesondere wenn auch noch Interaktionen mit im Spiel sind. Für kommentierte Beispiele mit dreifachen Interaktionen und Interaktionen mit kontinuierlichen Prädiktoren, siehe Jaccard (2001).

Interaktionen in generalisierten linearen Modellen. Interaktionen *theoretisch* zu interpretieren ist nicht so einfach, wie man manchmal vermutet (siehe Wagenmakers et al., 2012a). Bei logistischen Regressionsmodellen zeigt sich, wieso dies der Fall ist.

Die drei Grafiken auf der oberen Zeile von Abbildung 17.3 auf der vorherigen Seite zeigt drei Mal das gleiche Datenmuster, aber jeweils anders ausgedrückt. Ausgedrückt in Proportionen sieht man zwei parallele Linien (rot: 0.475–0.525, blau: 900–950), die zeigen, dass der Unterschied zwischen A und B in beiden Fällen 5 Prozentpunkte beträgt. Ausgedrückt in Proportionen liegt also keine Interaktion vor. Drückt man diese Proportionen aber in Chancen (odds) aus, dann ergeben sich nicht-parallele Linien, die auf die Existenz einer Interaktion hindeuten. Auch ausgedrückt als Log-Odds ergibt sich eine Interaktion.

Die drei Grafiken auf der mittleren Reihe sowie auch die drei Grafiken auf der unteren Reihe zeigen, dass es auch die umgekehrte Situationen geben kann: Auf der mittleren Reihe gibt es keine Interaktion, was die Chancen betrifft, aber schon was die Proportionen und Log-Odds betrifft; auf der unteren Reihe gibt es keine Interaktion, was die Log-Odds betrifft, aber schon was die Proportionen und Chancen betrifft.

Siehe *Interactions in logistic regression models* (7.8.2019).

17.3 Kontinuierliche Prädiktoren

Auch kontinuierliche Prädiktoren können in einem logistischen Modell berücksichtigt werden. Für dieses Beispiel verwenden wir die Daten einer einzigen Versuchspersonen aus der Studie von Vanhove & Berthele (2013). Sie versuchte 181 Wörter aus den germanischen Sprachen Niederländisch, Friesisch, Dänisch und Schwedisch ins Deutsche zu übersetzen. Die Spalte `Korrekt` zeigt für jedes Wort, ob ihr dies gelungen ist; die Spalte `OrthOverlap` zeigt, was die orthografische Überlappung zwischen dem zu übersetzenden Wort und seiner Übersetzung ist (0 = keine Überlappung; 10 = vollständige Überlappung). Wir möchten wissen, wie stark die Erfolgsquote beim Übersetzen von der orthografischen Überlappung abhängt.

```
# Daten einlesen
d <- read_csv("Data/VanhoveBerthele2013_eineVpn.csv")

## Parsed with column specification:
## cols(
##   Item = col_character(),
##   RichtigeUebersetzung = col_character(),
##   Zielsprache = col_character(),
##   Korrekt = col_character(),
##   OrthOverlap = col_double()
## )

# Neue Variable mit 1 und 0
d$Richtig <- ifelse(d$Korrekt == "richtig", 1, 0)
```

Um eine erste Idee über den Zusammenhang zwischen dem Prädiktor und der Richtig-

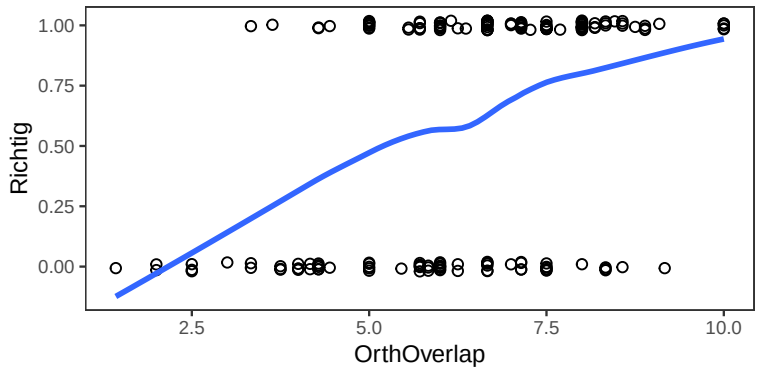


Abbildung 17.4: Ein Streudiagramm mit einer Trendlinie, die davon ausgeht, dass die abhängige Variable kontinuierlich (statt binär) ist. Immerhin zeigt sie, dass einen *monotonen* Zusammenhang zwischen den beiden Variablen gibt.

keit der Übersetzung zu erhalten, können wir ein Streudiagramm mit einer Trendlinie zeichnen (Abbildung 17.4). Die Trendlinie geht davon aus, dass die abhängige Variable kontinuierlich ist, was hier natürlich nicht der Fall ist. Eine Konsequenz dieser Annahme ist, dass die Trendlinie vermuten lässt, dass die durchschnittliche Proportion richtiger Antworten bei Wörtern mit niedriger orthografischer Überlappung *negativ* ist.

```
ggplot(data = d,
       aes(x = OrthOverlap,
           y = Richtig)) +
  geom_point(shape = 1,
             # Die Punkte vertikal leicht verschieben, um sie
             # besser sichtbar zu machen.
             position = position_jitter(width = 0, height = 0.02)) +
  # se = FALSE schaltet das Konfidenzband aus
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Die Grafik hat trotzdem ihren Nutzen: Sie deutet darauf hin, dass der Zusammenhang zwischen dem Prädiktor und der abhängigen Variablen *monoton* ist: Wenn die orthografische Überlappung grösser wird, steigt die Erfolgsquote. Es ist zum Beispiel nicht so, dass die Erfolgsquote zuerst ansteigt und dann konstant bleibt oder senkt.

Der kontinuierliche Prädiktor kann wie gehabt dem Modell hinzugefügt werden:

```
vanhove.glm <- glm(Richtig ~ OrthOverlap,
                   data = d, family = binomial(link = "logit"))
summary(vanhove.glm)
```

```
##
## Call:
## glm(formula = Richtig ~ OrthOverlap, family = binomial(link = "logit"),
```

```
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1201  -1.0364   0.6203   0.9167   1.7698
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3132     0.7340  -4.514 6.37e-06
## OrthOverlap   0.5944     0.1144   5.198 2.01e-07
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 242.45  on 180  degrees of freedom
## Residual deviance: 207.54  on 179  degrees of freedom
## AIC: 211.54
##
## Number of Fisher Scoring iterations: 3
```

Zur Interpretation der geschätzten Parameter:

- Das Intercept zeigt die modellierte Erfolgsquote (in log-odds!) für Fälle, in denen alle Prädiktoren den Wert 0 haben. In unserem Fall hiesse das also die modellierte Erfolgsquote bei Wörtern mit einer orthografischen Überlappung von 0. (Diese gibt es in diesem Datensatz übrigens nicht.) Konvertiert zu einer Chance:

  ```
  exp(-3.31)

  ## [1] 0.03651617
  ```

  Laut dem Modell ist eine richtige Antwort bei Wörtern ohne orthografische Überlappung also 0.037 Mal so wahrscheinlich wie eine falsche Antwort. (Oder umgekehrt: Eine falsche ist 27.4 Mal so wahrscheinlich wie eine richtige.)

  In Wahrscheinlichkeiten: Die Wahrscheinlichkeit, dass ein Wort ohne orthografische Überlappung richtig übersetzt wird, liegt bei 3.5%.

  ```
  plogis(-3.31)

  ## [1] 0.03522972
  ```

- Der geschätzte Parameter für `OrthOverlap` drückt aus, wie sich die Erfolgsquote (in log-odds!) laut dem Modell ändert, wenn `OrthOverlap` um eine Einheit ansteigt. Konvertiert zu einem Chancenverhältnis zeigt sich, dass es bei einem Wort mit einer orthografischen Überlappung von a 1.8 Mal so wahrscheinlich ist, eine richtige Antwort zu geben, als bei einem Wort mit einer orthografischen Überlappung von $a - 1$:

```
exp(0.594)
```

```
## [1] 1.811219
```

- Ein konkreteres Beispiel: Die modellierte Erfolgsquote bei einem Wort mit einer orthografischen Überlappung von 5.4 beträgt, ausgedrückt in einem Chancenverhältnis, 0.90.

```
exp(-3.31 + 0.594 * 5.4)
```

```
## [1] 0.9026684
```

Ausgedrückt in einer Wahrscheinlichkeit beträgt sie 47.4%.

```
plogis(-3.31 + 0.594 * 5.4)
```

```
## [1] 0.4744223
```

Bei einem Wort mit einer orthografischen Überlappung von 6.4 erhält man die folgenden Werte:

```
exp(-3.31 + 0.594 * 6.4)
```

```
## [1] 1.63493
```

($\frac{1.63}{0.90} = 1.8$.)

```
plogis(-3.31 + 0.594 * 6.4)
```

```
## [1] 0.6204833
```

Um die Ergebnisse des Modells zu berichten, ist es sinnvoll, diese grafisch darzustellen. In der Regel stellt man hierbei dar, wie sich die modellierte Wahrscheinlichkeit mit den kontinuierlichen Prädiktoren ändert (siehe Abbildung 17.5):

```
# Datensatz, in dem der Prädiktor
# zwisch ihrem Mindest- und Höchstwert variiert
df_pred <- data.frame(OrthOverlap = seq(from = min(d$OrthOverlap),
                                        to = max(d$OrthOverlap),
                                        length.out = 100))

# Modellierte Wahrscheinlichkeiten ("response") hinzufügen
df_pred$Wahrscheinlichkeit <- predict(vanhove.glm, newdata = df_pred,
                                      type = "response")

# Zeichnen
ggplot(df_pred,
       aes(x = OrthOverlap,
           y = Wahrscheinlichkeit)) +
  geom_line() +
  ylim(0, 1) +
```

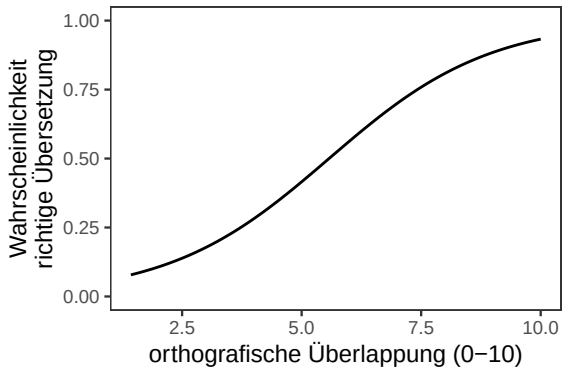


Abbildung 17.5: Der modellierte Zusammenhang zwischen der orthografischen Überlappung und der Wahrscheinlichkeit, zu der ein Wort richtig übersetzt wurde.

```
  xlab("orthografische Überlappung (0-10)") +
  ylab("Wahrscheinlichkeit\nrichtige Übersetzung")
```

Für mehr Info siehe *Tutorial: Adding confidence bands to effect displays* (12.5.2017) und `https://data.library.virginia.edu/visualizing-the-effects-of-logistic-regression/`.

Nicht-Linearitäten nicht überinterpretieren! Abbildung 17.5 zeigt, dass es *laut dem Modell* einen nicht-linearen Zusammenhang zwischen dem Grad an orthografischer Überlappung und der Erfolgswahrscheinlichkeit gibt. Dies ist nicht erstaunlich: Wir haben diesen Zusammenhang zwar linear, aber in log-odds modelliert. Wenn wir ihn dann in Wahrscheinlichkeiten ausdrücken, ergibt sich eine nicht-lineare Kurve statt einer Geraden. Wir hätten gar keinen linearen Zusammenhang zwischen dem Prädiktor und der Erfolgswahrscheinlichkeit finden *können*.

Kapitel 18

Weiterbildung

In einem Kurs wie diesem kann natürlich nicht das ganze Spektrum von der Berechnung eines Mittelwerts hin bis zur Modellierung komplizierter Datensätze abgedeckt werden. Über welche statistischen Methoden Sie sich weiter schlau machen sollten, hängt davon ab, welche Art von Datensätzen Sie in Ihren eigenen Projekten auswerten werden. Im Folgenden erhalten Sie eine kurze Übersicht über die Methoden, die für die meisten unter Ihnen nützlich sein dürften.

18.1 Mit kategorischen *outcomes* arbeiten

Oft sind die *outcomes*, mit denen wir es zu tun haben, nicht kontinuierlich sondern kategorisch. Zum Beispiel kann man Übersetzungsversuche als 'gelungen' oder 'nicht gelungen' betrachten anstatt diese auf einer kontinuierlichen Skala einzustufen. Manchmal hat man sogar outcome-Variablen mit mehr als zwei Ausprägungen: Etwa kann man die Sprache eines Dokumentes als 'Französisch', 'Italienisch', 'Deutsch', 'Englisch' oder 'andere' kodieren.

Outcomes dieser Art – ob sie nun zwei oder mehrere Ausprägungen haben – kann man nicht in einem allgemeinen linearen Modell auswerten; dieses Modell ist nämlich nur für kontinuierliche outcomes gedacht. Es gibt aber Modelle, die die Prinzipien des allgemeinen linearen Modells auf nicht-kontinuierliche outcomes verallgemeinern. Für binäre outcomes bietet sich so das **(binomiale) logistische Regressionsmodell** an. Literaturempfehlungen:

- Jaeger (2008, insbesondere S. 434–442) erklärt, wieso man binäre outcomes nicht einfach als 0 vs. 1 kodieren soll und in einem allgemeinen linearen Modell analysieren soll. Ausserdem erklärt er das Prinzip hinter logistischer Regression und illustriert dieses Verfahren mit einem linguistischen Beispiel.
- Johnson (2008, Kapitel 5) erklärt logistische Regression (sowie auch χ^2-Tests) anhand von Beispielen aus der variationistischen Soziolinguistik.

- Baayen (2008, S. 195–208) bespricht ein weiteres Beispiel.

Wenn man es mit outcomes mit mehr als zwei Ausprägungen zu tun hat, sollte man sich wohl über das **multinomiale logistische Regressionsmodell** oder eine Alternative schlau machen:

- Auch wenn sich bei Faraway (2006, Kapitel 5) die Mathe nicht vermeiden lässt, findet man dort eine kurze Anleitung.
- McElreath (2016, Abschnitt 10.3) bespricht ebenfalls multinomiale Regression, aber aus bayesscher Perspektive.

18.2 Abhängigkeitsstrukturen berücksichtigen

Zwei Beispiele von Datensätzen mit Abhängigkeitsstrukturen sind:

- Die Versuchspersonen reagieren auf mehrere Stimuli. Die Reaktionen einer bestimmten Versuchsperson werden einander ähnlicher sein als die Reaktionen unterschiedlicher Versuchspersonen. Ausserdem werden die Reaktionen auf einen bestimmten Stimulus einander ähnlicher sein als die Reaktionen auf unterschiedliche Stimuli.
- Die Versuchspersonen bilden 'Klümpchen', zum Beispiel weil einige unter ihnen von der gleichen Lehrperson unterrichtet werden, in die gleiche Schule gehen oder aus dem gleichen Haushalt kommen. Versuchspersonen aus dem gleichen Klümpchen werden einander tendenziell ähnlicher sein als Versuchspersonen aus unterschiedlichen Klümpchen.

Diese Arten von Abhängigkeiten zwischen Datenpunkten sollten bei der Analyse berücksichtigt werden. Eine Möglichkeit dazu ist, die Daten in sog. 'gemischten' Modellen zu analysieren. Literaturempfehlungen:

- Auf Bodo Winters Website finden Sie eine ausgezeichnete konzeptuelle Erklärung über gemischte Modelle: `http://www.bodowinter.com/tutorials.html`.
- Baayen et al. (2008) und Baayen (2008, Kapitel 7) besprechen weitere Beispiele. Bestimmte von ihnen verwendete Funktionen funktionieren aber mittlerweile nicht mehr (insb. die `pvals.fnc()`-Funktion).
- Jaeger (2008, insbesondere S. 442–445) und Quené & van den Bergh (2008) besprechen *logistische* gemischte Modelle. Diese sind nützlich, wenn man sowohl binäre outcomes als auch Abhängigkeitsstrukturen in den Daten hat.

18.3 Nicht-lineare Zusammenhänge modellieren

Wenn Ihre Daten deutliche nicht-lineare Zusammenhänge aufweisen, können Sie diese in generalisierten additiven Modellen analysieren. Eine nützliche Einführung findet sich unter `https://m-clark.github.io/generalized-additive-models/`. Auch die Erklärung in Zuur et al. (2009, Kapitel 3) finde ich gelungen. Man kann generalisierte additive Modelle auch mit gemischten Modellen kombinieren und sie können auch mit binären outcomes umgehen. Für ein Beispiel, siehe Vanhove (2014).

18.4 Vorherige Information übers Problem berücksichtigen

Bayessche Statistik stellt eine Alternative zum sog. 'frequentistischen' Ansatz, dem dieser Kurs gewidmet war. BefürworterInnen von bayesscher Statistik sehen viele Vorteile gegenüber frequentistischer Statistik (siehe etwa Dienes, 2011; Goodman, 2008). Selber sehe ich als grösste Vorteile bayesscher Ansätze, dass man einfacher Modelle, die andere Annahmen machen, konstruieren kann und dass diese es besser ermöglichen, vorheriges Wissen über die Daten – auch vages Wissen – in der Analyse zu berücksichtigen. Oft weiss man nämlich wesentlich mehr über das Problem, zu dem man forscht, als der Computer:

- Vielleicht weiss man nicht genau, wie viel schneller Zweisprachige auf Kognatwörter als auf andere Wörter reagieren – aber man weiss schon, dass es sich um allerhöchstens 1.5–2 Sekunden handelt.
- Vielleicht weiss man nicht, wie viel effizienter eine etablierte Lernmethode als eine andere ist, aber es ist wesentlich wahrscheinlicher, dass der Unterschied ziemlich klein ist (vielleicht 5%) als dass er riesig (z.B. 80%) ist: Sonst hätte man dies wohl auch ohne Forschung bemerkt.

Wenn man dem statistischen Modell diese Infos nicht 'mitteilt', 'weiss' es nicht, dass im ersten Beispiel Unterschiede von 10 Sekunden und im zweiten Beispiel von 90% implausibel sind. Insbesondere wenn die Datenmenge gering oder die Variabilität in den Daten gross ist, können Modelle sinnvollere Inferenzen liefern, wenn solche *a priori* Informationen in die Modellierung einfliessen.

Eine gut gelungene Einführung in bayesscher Statistik, in der insbesondere den Mehrwert vorherigen Wissens betont wird, ist McElreath (2016). Auch wenn man nicht vorhat, selber bayessche Modelle zu verwenden, lohnt sich die Lektüre dieses Buchs, denn man lernt auch viel über andere Konzepte. Kürzere Einführungen, die auch mehr auf LinguistInnen und PsychologInnen zugeschnitten sind, bieten Nicenboim & Vasishth (2016a,b) und Sorensen et al. (2016).

18.5 Abschliessende Tipps

- *Use it or lose it.* Bauen Sie R in Ihren beruflichen Alltag ein. Wenn Sie noch nicht über eigene Daten verfügen, können Sie trotzdem R und RStudio verwenden und so Ihre Fähigkeiten pflegen:
  - Vorträge gestalten, Artikel schreiben und eine Website kreieren, siehe `https://support.rstudio.com/hc/en-us/articles/200486468-Authoring-R-Presentations` und `https://www.emilyzabor.com/tutorials/rmarkdown_websites_tutorial.html`.
  - Grafiken zeichnen. Wenn Sie einen Artikel voller komplizierter Tabellen lesen, können Sie probieren, die Zahlen in ein Spreadsheet einzutragen und dann grafisch darzustellen. Ich finde es oft einfacher, Grafiken als Tabellen zu interpretieren. Es ist auch möglich, Datenpunkte aus publizierten Grafiken zu extrahieren, um diese besser darzustellen. Siehe `https://automeris.io/WebPlotDigitizer/` und `http://socviz.co/`.
  - Man kann in R auch Texte analysieren: `https://www.tidytextmining.com/` und `https://cran.r-project.org/web/packages/koRpus/vignettes/koRpus_vignette.html`.
- Planen Sie regelmässig Zeit fürs Selbststudium ein. Auch wenn es nur jede zweite Woche zwei Stunden am Freitagnachmittag sind, hilft Ihnen das mehr, als wenn Sie erst in den letzte Monaten Ihres Doktorats volle Pulle Statistik lernen. Ein paar Literaturempfehlungen (in Reihenfolge): Broman & Woo (2017), Gelman & Stern (2006), Goodman (2008), Simmons et al. (2011), Rohrer (2018), `http://www.bodowinter.com/tutorials.html`, Christenfeld et al. (2004), Vanhove (2015), Westfall & Yarkoni (2016).
- Lernen Sie, was Sie brauchen. Niemand ist mit allen Verfahren vertraut. Versuchen Sie, zu antizipieren, was Sie selber brauchen werden, und versuchen Sie sich, diese Verfahren anzueignen. Wenn Ihre *outcomes* kategorisch sein werden, sollten Sie sich zum Beispiel wohl über logistische Regression schlau machen (siehe Baayen, 2008; Johnson, 2008); wenn Ihre Daten eine Abhängigkeitsstruktur haben werden (z.B. mehrere Antworten pro Versuchsperson oder SchülerInnen in Klassen), dürften gemischte Modelle nützlich sein (siehe Baayen, 2008, `http://www.bodowinter.com/tutorials.html` und `http://jakewestfall.org/blog/index.php/2015/06/20/reading-list-introduction-to-linear-mixed-models-for-cognitive-sc`
  Es ist sowieso eine gute Idee, sich über die Analysemethode zu informieren, bevor man anfängt, die Daten zu sammeln.
- Gehen Sie nicht davon aus, dass publizierte Analysen Sinn ergeben. In vielen Artikeln ist dies nämlich nicht der Fall. Ausserdem werden viele Analysen schlecht berichtet. Oft wird zu viel irrelevante und daher ablenkende Information im Text berichtet und werden zu viele Signifikanztests berichtet.

- Verwenden Sie reichlich Grafiken und nicht (nur) Tabellen, sowohl beim Analysieren als auch in Ihren Berichten.
- Zeigen Sie möglichst die Variabilität der Daten (Streudiagramme, Boxplots) und die Unsicherheit Ihrer Schätzungen (z.B. Konfidenzintervalle).
- Machen Sie nach Möglichkeit Ihre Daten und R-Code online verfügbar. Einerseits brauchen Sie so Details, die Sie für irrelevant halten, nicht in den Bericht aufzunehmen: Interessierte Lesende können die Details selber im Anhang nachschlagen. Andererseits ist es durchaus möglich, dass in den nächsten Jahren statistische Verfahren entwickelt werden, mit denen man Ihre Daten besser auswerten kann. Wenn Ihre Daten frei zugänglich sind, können Sie entsprechend reanalysiert werden und vermeiden Sie, dass Ihre Studie irrelevant wird.

Teil VI

Anhang

Anhang A

Häufige Fehlermeldungen in R

Probleme immer der Reihe nach lösen! Wenn Ihr ganzer Bildschirm voller roter Fehlermeldungen steht, fangen Sie dann bei der allerersten Fehlermeldung an. Öfters sind die anderen Fehlermeldungen Konsequenzen der ersten.

Debugging. Wenn Sie jemanden um Hilfe bitten, sollten Sie ein reproduzierbares Beispiel, das das Problem illustriert, schreiben. Diese Beispiele sollten möglichst *kurz* sein, siehe `https://stackoverflow.com/q/5963269/1331521`. Oft findet man beim Schreiben eines möglichst kurzen reproduzierbaren Beispiels selber die Lösung. Eine weitere nützliche Technik, um Fehler in Computercode aufzudecken, ist *rubber duck debugging*, siehe `https://rubberduckdebugging.com/`

A.1 object 'x' not found

Sie versuchen ein Objekt abzurufen, das im Arbeitsgedächtnis nicht vorhanden ist. Tippfehler sind hierfür ein typischer Auslöser. Es kann auch vorkommen, dass Sie ein Objekt nicht eingelesen haben oder dass das Objekt anders heisst, als Sie denken.

Beispiel:

```
x <- c(1, 5, 4)
mean(X)

## Error in mean(X): object 'X' not found
```

In der ersten Zeile wird ein Objekt namens `x` kreiert, aber in der zweiten wird versucht, das Mittel eines Objektes namens `X` zu berechnen. Gross- und Kleinschreibung spielen eine Rolle!

Wenn Sie sich nicht mehr sicher sind, welche Objekte alles im Arbeitsgedächtnis vorhanden sind, können Sie folgende Funktion verwenden; diese listet alle vorhandenen Objekte auf.

```
ls()
```

A.2 could not find function ‘x’

Sie wollen eine Funktion verwenden, die nicht besteht oder nicht zugänglich ist. Kontrollieren Sie zuerst, ob Sie den Namen der Funktion richtig eingetippt haben. Wenn es sich um eine Funktion aus einem Erweiterungspaket handelt, sollten Sie ausserdem dieses Paket geladen haben.

Beispiel:

```
ggplot(dat,
       aes(x = predictor,
           y = outcome)) +
  geom_point()
```

```
## Error in ggplot(dat, aes(x = predictor, y = outcome)): could not find
function "ggplot"
```

Die `ggplot()`-Funktion ist Teil eines Erweiterungspaket. Verwenden Sie den Befehl `library(ggplot2)` (oder `library(tidyverse)`), um sie verwenden zu können.

A.3 Error in library(x) : there is no package called ‘x’

Sie versuchen ein Erweiterungspaket zu laden, aber haben dieses noch nicht installiert.

Beispiel:

```
library(ggjoy)
```

```
## Error in library(ggjoy): there is no package called 'ggjoy'
```

Lösung: Paket installieren:

```
install.packages("ggjoy")
```

A.4 unexpected symbol

Häufige Auslöser für diese Fehlermeldung sind vergessene Kommas.

Beispiel:

```
library(ggplot2)
ggplot(data = iris
       aes(x = Sepal.Width,
```

```
           y = Sepal.Length)) +
  geom_point()
```

```
## Error: <text>:3:8: unexpected symbol
## 2: ggplot(data = iris
## 3:        aes
##          ^
```

Nach `data = iris` fehlt eine Komma.

Beispiel:

```
ggplot(data = iris)
       aes(x = Sepal.Width,
           y = Sepal.Length)) +
  geom_point()
```

```
## Error: <text>:3:29: unexpected ')'
## 2:        aes(x = Sepal.Width,
## 3:            y = Sepal.Length))
##                                ^
```

Die Klammer nach der ersten Zeile soll eine Komma sein.

Beispiel:

```
ggplot(data = iris,
       aes(x = Sepal.Width,
           y = Sepal.Length))) +
  geom_point()
```

```
## Error: <text>:3:30: unexpected ')'
## 2:        aes(x = Sepal.Width,
## 3:            y = Sepal.Length)))
##                                ^
```

Auf der dritten Zeile steht eine Klammer zu viel.

A.5 Es funktioniert nicht und es gibt nicht mal eine Fehlermeldung!

Vermutlich fehlt irgendwo eine Klammer oder etwas Ähnliches.

Beispiel: Dieser Befehl ergibt keine Fehlermeldung, aber produziert auch keine Grafik.

```
> ggplot(data = iris,
+        aes(x = Sepal.Width,
+            y = Sepal.Length) +
```

```
+   geom_point()
+
```

Wie Sie sehen, gibt es nach dem Befehl eine neue Zeile, die mit + anfängt. Das heisst, dass der Befehl noch nicht abgeschlossen ist: Nach abgeschlossenen Befehlen folgen Zeilen, die mit > anfangen. In diesem Fall ist der Auslöser eine fehlende Klammer auf der 3. Zeile: Die letzte Klammer schliesst den Befehl `aes(` ab, aber es fehlt noch eine Klammer, um den Befehl `ggplot(` abzuschliessen. So funktioniert es schon:

```
> ggplot(data = iris,
+        aes(x = Sepal.Width,
+            y = Sepal.Length)) +
+   geom_point()
>
```

(Bemerken Sie, dass es nun eine neue Zeile mit > gibt: Der letzte Befehl wurde also ausgeführt.)

Anderes Beispiel: Dieser Befehl ergibt auch keine Fehlermeldung, aber produziert auch keine Grafik.

```
> ggplot(data = iris,
+        aes(x = Sepal.Width,
+            y = Sepal.Length))
> geom_point()
geom_point: na.rm = FALSE
stat_identity: na.rm = FALSE
position_identity
```

Problem: Das Plus-Zeichen nach der 3. Zeile fehlt.

A.6 Das Ergebnis einer Berechnung ist 'NA'

Sie wollen einen Mittelwert o.Ä. berechnen, aber die Datenreihe enthält `NA`-Angaben (*not available*). R betrachtet `NA`-Angaben als unbekannte Zahlen; enthält eine Datenreihe eine unbekannte Zahl, dass ist ihr Mittel oder ihre Standardabweichung (usw.) auch unbekannt (`NA`).

```
x <- c(5, 4, 18, NA, 3)
mean(x)

## [1] NA
```

Wenn Sie es für sinnvoll halten, können Sie das Mittel berechnen, ohne die unbekannte Zahl zu berücksichtigen:

```
mean(x, na.rm = TRUE)
```

```
## [1] 7.5
```

Für mehr Informationen zum Umgang mit fehlenden Daten, siehe Graham (2009).

Literaturverzeichnis

Abelson, Robert P. 1995. *Statistics as principled argument*. New York, NY: Psychology Press.

Albers, Casper J., Henk A. L. Kiers & Don van Ravenzwaaij. 2018. Credible confidence: A pragmatic view on the frequentist vs Bayesian debate. doi:10.17605/OSF.IO/WE9H6.

Amrhein, Valentin, Fränzi Korner-Nievergelt & Tobias Roth. 2017. The earth is flat ($p > 0.05$): significance thresholds and the crisis of unreplicable research. *PeerJ* 5. e3544. doi:10.7717/peerj.3544.

Baayen, R. H., D. J. Davidson & D. M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language* 59(4). 390–412. doi:10.1016/j.jml.2007.12.005.

Baayen, R. Harald. 2008. *Analyzing linguistic data: A practical introduction to statistics using R*. Cambridge: Cambridge University Press.

Baayen, R. Harald & Petar Milin. 2010. Analyzing reaction times. *International Journal of Psychological Research* 3.2. 12–28.

Baguley, Thom. 2009. Standardized or simple effect size: What should be reported? *British Journal of Psychology* 100(3). 603–617. doi:10.1348/000712608X377117.

Bender, Ralf & Stefan Lange. 2001. Adjusting for multiple testing: when and how? *Journal of Clinical Epidemiology* 54(4). 343–349. doi:10.1016/S0895-4356(00)00314-0.

Benjamin, Daniel J, James O Berger, Magnus Johannesson, Brian A Nosek, E-J Wagenmakers, Richard Berk, Kenneth A Bollen, Björn Brembs, Lawrence Brown, Colin Camerer et al. 2018. Redefine statistical significance. *Nature Human Behaviour* 2(1). 6. doi:10.1038/s41562-017-0189-z.

Berthele, Raphael. 2012. The influence of code-mixing and speaker information on perception and assessment of foreign language proficiency: An experimental study. *International Journal of Bilingualism* 16(4). 453–466. doi:10.1177/1367006911429514.

Berthele, Raphael. 2019. Policy recommendations for language learning: Linguists' contributions between scholarly debats and pseudoscience. *Journal of the European Second Language Association* 3(1). 1–11. doi:10.22599/jesla.50.

Bialystok, Ellen, Fergus I. M. Craik, Raymond Klein & Mythili Viswanathan. 2004. Bilingualism, aging, and cognitive control: Evidence from the Simon task. *Psychology and Aging* 19(2). 290–303. doi:10.1037/0882-7974.19.2.290.

Bland, J. Martin & Douglas G. Altman. 1994. One and two sided tests of significance. *BMJ* 309. 248.

Broman, Karl W. & Kara H. Woo. 2017. Data organization in spreadsheets. *The American Statistician* doi:10.1080/00031305.2017.1375989.

Buja, Andreas, Dianne Cook, Heike Hofmann, Michael Lawrence, Eun Kyung Lee, Deborah F. Swayne & Hadley Wickham. 2009. Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A* 367(1906). doi:10.1098/rsta.2009.0120.

Bürkner, Paul-Christian & Matti Vuorre. 2018. Ordinal regression models in psychology: A tutorial. doi:10.31234/osf.io/x8swp.

Caruso, Eugene M., Kathleen D Vohs, Brittani Baxter & Adam Waytz. 2013. Mere exposure to money increases endorsement of free-market systems and social inequality. *Journal of Experimental Psychology: General* 142(2). 301–306. doi:10.1037/a0029288.

Chambers, Chris. 2017. *The seven deadly sins of psychology: A manifesto for reforming the culture of scientific practice*. Princeton, NJ: Princeton University Press.

Champely, Stephane. 2018. pwr: Basic functions for power analysis. package, version 1.2-2. `http://cran.r-project.org/package=pwr`.

Christenfeld, Nicholas J. S., Richard P. Sloan, Douglas Carroll & Sander Greenland. 2004. Risk factors, confounding, and the illusion of statistical control. *Psychosomatic Medicine* 66. 868–875. doi:10.1097/01.psy.0000140008.70959.41.

Clark, Michael. 2018. Generalized additive models. `https://m-clark.github.io/generalized-additive-models/`.

Cohen, Jacob. 1977. *Statistical power analysis for the behavioral sciences*. New York, NY: Academic Press revised edition edn.

Cohen, Jacob. 1992. A power primer. *Psychological Bulletin* 112(1). 155–159.

Cohen, Jacob, Patricia Cohen, Stephen G. West & Leona S. Aiken. 2003. *Applied multiple regression/correlation analysis for the behavioral sciences*. Mahwah, NJ: Erlbaum.

de Bruin, Angela, Barbara Treccani & Sergio Della Sala. 2015. Cognitive advantage in bilingualism: An example of publication bias? *Psychological Science* 26(1). 99–107. doi:10.1177/0956797614557866.

de Groot, Adrianus Dingeman. 2014. The meaning of 'significance' for different types of research [translated and annotated by Eric-Jan Wagenmakers, Denny Borsboom, Josine Verhagen, Rogier Kievit, Marjan Bakker, Angelique Cramer, Dora Matzke, Don Mellenbergh, and Han L. J. van der Maas]. *Acta Psychologica* 148. 188–194. doi:10.1016/j.actpsy.2014.02.001.

DeKeyser, Robert, Iris Alfi-Shabtay & Dorit Ravid. 2010. Cross-linguistic evidence for the nature of age effects in second language acquisition. *Applied Psycholinguistics* 31. 413–438. doi:10.1017/S0142716410000056.

DiCiccio, Thomas J. & Bradley Efron. 1996. Bootstrap confidence intervals. *Statistical Science* 11(3). 189–212. `http://www.jstor.org/stable/2246110`.

Dienes, Zoltan. 2011. Bayesian versus orthodox statistics: Which side are you on? *Perspectives on Psychological Science* 6(3). 274–290. doi:10.1177/1745691611406920.

Efron, Bradley. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7(1). 1–26. doi:10.1214/aos/1176344552.

Efron, Bradley & Robert J. Tibshirani. 1993. *An introduction to the bootstrap*. Boca Raton, FL: Chapman & Hall/CRC.

Ehrenberg, Andrew S. C. 1982. *A primer in data reduction: An introductory statistics textbook*. Chichester: Wiley.

Eriksen, Barbara A. & Charles W. Eriksen. 1974. Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & Psychophysics* 16. 143–149. doi:10.3758/BF03203267.

Everitt, Brian & Torsten Hothorn. 2011. *An introduction to applied multivariate analysis with R*. New York: Springer.

Faraway, Julian J. 2005. *Linear models with R*. Boca Raton, FL: Chapman & Hall/CRC.

Faraway, Julian J. 2006. *Extending the linear model with r: Generalized linear, mixed effects and nonparametric regression models*. Boca Raton, FL: Chapman & Hall/CRC.

Fisher, Ronald Aylmer. 1936. "The coefficient of racial likeness" and the future of craniometry. *Journal of the Royal Anthropological Institute of Great Britain and Ireland* 66. 57–63.

Fox, John. 2003. Effect displays in R for generalised linear models. *Journal of Statistical Software* 8. 1–27. doi:10.18637/jss.v008.i15.

Gelman, Andrew & John Carlin. 2014. Beyond power calculations: Assessing Type S (sign) and Type M (magnitude) errors. *Perspectives on Psychological Science* 9(6). 641–651. doi:10.1177/1745691614551642.

Gelman, Andrew & Jennifer Hill. 2007. *Data analysis using regression and multilevel/hierarchical models*. New York: Cambridge University Press.

Gelman, Andrew & Eric Loken. 2013. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no 'fishing expedition' or 'p-hacking' and the research hypothesis was posited ahead of time. `http://www.stat.columbia.edu/~gelman/research/unpublished/p_hacking.pdf`.

Gelman, Andrew & Hal Stern. 2006. The difference between "significant" and "not significant" is not itself statistically significant. *The American Statistician* 60(4). 328–331. doi:10.1198/000313006X152649.

Goodman, Steven. 2008. A dirty dozen: Twelve *p*-value misconceptions. *Seminars in Hematology* 45. 135–140. doi:10.1053/j.seminhematol.2008.04.003.

Graham, John W. 2009. Missing data analyis: Making it work in the real world. *Annual Review of Psychology* 60. 549–576. doi:10.1146/annurev.psych.58.110405.085530.

Greenland, Sander, Stephen J. Senn, Kenneth J. Rothman, John B. Carlin, Charles Poole, Steven N. Goodman & Douglas G. Altman. 2016. Statistical tests, p values, confidence intervals, and power: A guide to misinterpretations. *European Journal of Epidemiology* 31. 337–350. doi:10.1007/s10654-016-0149-3.

Guiora, Alexander Z., Benjamin Beit-Hallahmi, Robert C. L. Brannon, Cecelia Y. Dull & Thomas Scovel. 1972. The effects of experimentally induced changes in ego state on pronunciation ability in a second language: An exploratory study. *Comprehensive Psychiatry* 13(5). 421–428.

Harrell, Frank E., Jr. 2001. *Regression modeling strategies with application to linear models, logistic regression, and survival analysis*. New York, NY: Springer. doi:10.1007/978-1-4757-3462-1.

Healy, Kieran. 2019. *Data visualization: A practical introduction*. Princeton, NJ: Princeton University Press.

Heathers, James A., Jordan Anaya, Tim van der Zee & Nicholas J. L. Brown. 2018. Recovering data from summary statistics: Sample Parameter Reconstruction via Iterative TEchniques (SPRITE). *PeerJ Preprints* 6. e26968v1. doi:https://doi.org/10.7287/peerj.preprints.26968v1.

Hesterberg, Tim C. 2015. What teachers should know about the bootstrap: Resampling in the undergraduate statistics curriculum. *The American Statistician* 69(4). 371–386. doi:10.1080/00031305.2015.1089789.

Hoekstra, Rink, Richard D. Morey, Jeffrey N. Rouder & Eric-Jan Wagenmakers. 2014. Robust misinterpretation of confidence intervals. *Psychonomic Bulletin & Review* 21(5). 1157–1164. doi:10.3758/s13423-013-0572-3.

Honaker, James, Gary King & Matthew Blackwell. 2011. Amelia II: A program for missing data. *Journal of Statistical Software* 45. doi:10.18637/jss.v045.i07.

Huff, Darrell. 1954. *How to lie with statistics*. New York: Norton.

Huitema, Bradley E. 2011. *The analysis of covariance and alternatives: Statistical methods for experiments, quasi-experiments, and single-case studies*. Hoboken, NJ: Wiley.

Imai, Kosuke, Gary King & Elizabeth A. Stuart. 2008. Misunderstandings between experimentalists and observationalists about causal inference. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 171. 481–502. doi:10.1111/j.1467-985X.2007.00527.x.

Jaccard, James. 2001. *Interaction effects in logistic regression*. Thousand Oaks, CA: Sage.

Jaeger, T. Florian. 2008. Categorical data analysis: Away from ANOVAs (transformation or not) and towards logit mixed models. *Journal of Memory and Language* 59(4). 434–446. doi:10.1016/j.jml.2007.11.007.

Johnson, Daniel Ezra. 2013. Descriptive statistics. In Robert J. Podesva & Devyani Sharma (eds.), *Research methods in linguistics*, 288–315. Cambridge: Cambridge University Press.

Johnson, Keith. 2008. *Quantitative methods in linguistics*. Malden, MA: Blackwell.

Kelley, Ken, Scott E. Maxwell & Joseph R. Rausch. 2003. Obtaining power or obtaining precision. *Evaluation & the Health Professions* 26(3). 258–287. doi:10.1177/0163278703255242.

Kerr, Norbert L. 1998. HARKing: Hypothesizing after the results are known. *Personality and Social Psychology Review* 2(3). 196–217. doi:10.1207/s15327957pspr0203_4.

Keysar, Boas, Sayuri L. Hayakawa & Sun Gyu An. 2012. The foreign-language effect: Thinking in a foreign tongue reduces decision biases. *Psychological Science* 23(6). 661–668.

Klein, Richard A., Kate A. Ratliff, Michelangelo Vianello, Reginald B. Adams Jr., Štěpán Bahník, Michael J. Bernstein, Konrad Bocian et al. 2014. Investigating variation in replicability: A "many labs" replication project. *Social Psychology* 45(3). 142–152. doi:10.1027/1864-9335/a000178.

Kuhn, Max & Kjell Johnson. 2013. *Applied predictive modeling*. New York: Springer. doi:10.1007/978-1-4614-6849-3.

Kuperman, Victor & Julie A. Van Dyke. 2013. Reassessing word frequency as a determinant of word recognition for skilled and unskilled readers. *Journal of Experimental Psychology: Human Perception and Performance* 39. 802–823. doi:10.1037/a0030859.

Kvålseth, Tarald O. 1985. Cautionary note about R^2. *The American Statistician* 4(1). doi:10.2307/2683704.

Lakens, Daniel, Federico G Adolfi, Casper J Albers, Farid Anvari, Matthew AJ Apps, Shlomo E Argamon et al. 2018. Justify your alpha. *Nature Human Behaviour* 2(3). 168. doi:10.1038/s41562-018-0311-x.

Lakens, Daniël. 2014. Performing high-powered studies efficiently with sequential analyses. *European Journal of Social Psychology* 44. 701–710. doi:10.1002/ejsp.2023.

Liddell, Torrin M. & John K. Kruschke. 2018. Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology* 79. 328–348. doi:10.1016/j.jesp.2018.08.009.

Majumber, Mahbubul, Heike Hofmann & Dianne Cook. 2013. Validation of visual statistical inference, applied to linear models. *Journal of the American Statistical Association* 108(503). 942–956. doi:10.1080/01621459.2013.808157.

McElreath, Richard. 2016. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Boca Raton, FL: CRC Press.

McShane, Blakeley B, David Gal, Andrew Gelman, Christian Robert & Jennifer L Tackett. 2017. Abandon statistical significance. *arXiv preprint arXiv:1709.07588* `https://arxiv.org/abs/1709.07588`.

Mook, Douglas G. 1983. In defense of external invalidity. *American Psychologist* 38. 379–387.

Morey, Richard D., Rink Hoekstra, Jeffrey N. Rouder, Michael D. Lee & Eric-Jan Wagenmakers. 2016. The fallacy of placing confidence in confidence intervals. *Psychonomic Bulletin & Review* 23(1). 103–123. doi:10.3758/s13423-015-0947-8.

Mutz, Diana C., Robin Pemantle & Philip Pham. 2017. The perils of balance testing in experimental design: Messy analyses of clean data. *The American Statistician* doi:10.1080/00031305.2017.1322143.

Nalborczyk, Ladislas, Paul-Christian Bürkner & Donald R. Williams. 2018. Pragmatism should not be a substitute for statistical literacy: A commentary on Albers, Kiers, and van Ravenzwaaij (2018). `https://psyarxiv.com/fmtkj/`.

Nicenboim, Bruno & Shravan Vasishth. 2016a. Statistical methods for linguistic research: Foundational ideas—Part I. *Language and Linguistics Compass* 10(8). 349–369. doi:10.1111/lnc3.12201.

Nicenboim, Bruno & Shravan Vasishth. 2016b. Statistical methods for linguistic research: Foundational ideas—Part II. *Language and Linguistics Compass* 10(11). 591–613. doi:10.1111/lnc3.12207.

Nieuwenhuis, Sander, Birte U. Forstmann & Eric-Jan Wagenmakers. 2011. Erroneous analyses of interactions in neuroscience: A problem of significance. *Nature Neuroscience* 14. 1105–1107. doi:10.1038/nn.2886.

Noether, G. E. 1981. Why Kendall tau? *Teaching Statistics* 3(2). 41–43. doi:10.1111/j.1467-9639.1981.tb00422.x.

O'Brien, Robert M. 2007. A caution regarding rules of thumb of variance inflation factors. *Quality & Quantity* 41. 673–690. doi:10.1007/s11135-006-9018-6.

Oehlert, Gary W. 2010. *A first course in the design and analysis of experiments*. `http://users.stat.umn.edu/~gary/book/fcdae.pdf`.

Oppenheimer, Daniel M & Benoît Monin. 2009. The retrospective gambler's fallacy: Unlikely events, constructing the past, and multiple universes. *Judgment and Decision Making* 4(5). 326–334.

Peterson, David. 2016. The baby factory: Difficult research objects, disciplinary standards, and the production of statistical significance. *Socius* 2. 1–10. doi:10.1177/2378023115625071.

Plonsky, Luke & Frederick L. Oswald. 2014. How big is "big"? Interpreting effect sizes in L2 research. *Language Learning* 64(4). 878–912. doi:10.1111/lang.12079.

Poarch, Gregory J., Jan Vanhove & Raphael Berthele. 2019. The effect of bidialectalism on executive function. *International Journal of Bilingualism* 23(2). 612–628. doi:10.1177/1367006918763132.

Quené, Hugo & Huub van den Bergh. 2008. Examples of mixed-effects modeling with crossed random effects and with binomial data. *Journal of Memory and Language* 59. 413–425. doi:10.1016/j.jml.2008.02.002.

Renner, Fritz, Inge Kersbergen, Matt Field & Jessica Werthmann. 2018. Dutch courage? Effects of acute alcohol consumption on self-ratings and observer ratings of foreign language skills. *Journal of Psychopharmacology* 32(1). 116–122. doi:10.1177/0269881117735687.

Roberts, David R., Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig & Carsten F. Dormann. 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* 40(8). 913–929. doi:10.1111/ecog.02881.

Rohrer, Julia M. 2018. Thinking clearly about correlations and causation: Graphical causal models for observational data. *Advances in Methods and Practices in Psychological Science* 1(1). 27–42. doi:10.1177/2515245917745629.

Ruxton, Graeme D. 2006. The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney u test. *Behavioral Ecology* 17. 688–690. doi:10.1093/beheco/ark016.

Ruxton, Graeme D. & Guy Beauchamp. 2008. Time for some a priori thinking about post hoc testing. *Behavioral Ecology* 19(3). 690–693. doi:10.1093/beheco/arn020.

Schad, Daniel J., Sven Hohenstein, Shravan Vasishth & Reinhold Kliegl. 2018. How to capitalize on a priori contrasts in linear (mixed) models: A tutorial. `https://arxiv.org/abs/1807.10451`.

Schmidt, Frank L. 1996. Statistical significance testing and cumulative knowledge in psychology: Implications for training of researchers. *Psychological Methods* 1. 115–129. doi:10.1037/1082-989X.1.2.115.

Sedlmeier, Peter & Gerd Gigerenzer. 1989. Do studies of statistical power have an effect on the power of studies? *Psychological Bulletin* 105. 309–316.

Simmons, Joseph P., Leif D. Nelson & Uri Simonsohn. 2011. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science* 22. 1359–1366. doi:10.1177/0956797611417632.

Simmons, Joseph P., Leif D. Nelson & Uri Simonsohn. 2018. False-positive citations. *Perspectives on Psychological Science* 13(2). 255–259. doi:10.1177/1745691617698146.

Simon, J. Richard. 1969. Reactions toward the source of stimulation. *Journal of Experimental Psychology* 81. 174–176. doi:10.1037/h0027448.

Slavin, Robert E., Nancy Madden, Malgarita Calderón, Anne Chamberlain & Megan Hennessy. 2011. Reading and language outcomes of a multiyear randomized evaluation of transitional bilingual education. *Educational Evaluation and Policy Analysis* 33(1). 47–58. doi:10.3102/0162373711398127.

Sorensen, Tanner, Sven Hohenstein & Shravan Vasishth. 2016. Bayesian linear mixed models using Stan: A tutorial for psychologists, linguists, and cognitive scientists. *Quantitative Methods for Psychology* 12(3). 175–200. doi:10.20982/tqmp.12.3.p175.

Steegen, Sara, Francis Tuerlinckx, Andrew Gelman & Wolf Vanpaemel. 2016. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science* 11(5). 702–712. doi:10.1177/1745691616658637.

Sterling, Theodore D., W. L. Rosenbaum & J. J. Weinkam. 1995. Publication decision revisited: The effect of the outcome of statistical tests on the decision to publish and vice versa. *The American Statistician* 49. 108–112. `http://www.jstor.org/stable/2684823`.

Stocker, Ladina. 2017. The impact of foreign accent on credibility: An analysis of cognitive statement ratings in a Swiss context. *Journal of Psycholinguistic Research* 46(3). 617–628. doi:10.1007/s10936-016-9455-x.

Strobl, Carolin, Torsten Hothorn & Achim Zeileis. 2009. Party on! A new, conditional variable-importance measure for random forests available in the party package. *R Journal* 1. 14–17.

Tversky, Amos & Daniel Kahneman. 1981. The framing of decisions and the psychology of choice. *Science* 211(4481). 453–458. doi:10.1126/science.7455683.

Vanhove, Jan. 2013. The critical period hypothesis in second language acquisition: A statistical critique and a reanalysis. *PLOS ONE* 8. e69172. doi:10.1371/journal.pone.0069172.

Vanhove, Jan. 2014. *Receptive multilingualism across the lifespan: Cognitive and linguistic factors in cognate guessing*: University of Fribourg dissertation. `http://doc.rero.ch/record/210293`.

Vanhove, Jan. 2015. Analyzing randomized controlled interventions: Three notes for applied linguists. *Studies in Second Language Learning and Teaching* 5. 135–152. doi: 10.14746/ssllt.2015.5.1.7.

Vanhove, Jan. 2016. The early learning of interlingual correspondence rules in receptive multilingualism. *International Journal of Bilingualism* 20(5). 580–593. doi: 10.1177/1367006915573338.

Vanhove, Jan. 2017. The influence of standard and substandard Dutch on gender assignment in second language German. *Language Learning* 67(2). 431–460. doi: 10.1111/lang.12230.

Vanhove, Jan. 2018a. Checking the assumptions of your statistical method without getting paranoid. doi:10.31234/osf.io/zvawb.

Vanhove, Jan. 2018b. Metalinguistic knowledge about the native language and language transfer in gender assignment.

Vanhove, Jan & Raphael Berthele. 2013. Factoren bij het herkennen van cognaten in onbekende talen: algemeen of taalspecifiek? *Taal & Tongval* 65. 171–210.

Vanhove, Jan & Raphael Berthele. 2017. Interactions between formal distance and participant-related variables in receptive multilingualism. *International Review of Applied Linguistics in Language Teaching* 55(1). 23–40. doi:10.1515/iral-2017-0007.

Wagenmakers, Eric-Jan, Angelos-Miltiadis Krypotos, Amy H. Criss & Geoff Iverson. 2012a. On the interpretation of removable interactions: A survey of the field 33 years after Loftus. *Memory & Cognition* 40(2). 145–160. doi:10.3758/s13421-011-0158-0.

Wagenmakers, Eric-Jan, Ruud Wetzels, Denny Borsboom, Han L. J. van der Maas & Rogier A. Kievit. 2012b. An agenda for purely confirmatory research. *Perspectives on Psychological Science* 7(6). doi:10.1177/1745691612463078.

Weisberg, Sanford. 2005. *Applied linear regression*. Hoboken, NJ: Wiley.

Weissgerber, Tracey L., Natasa M. Milic, Stacey J. Winham & Vesna D. Garovic. 2015. Beyond bar and line graphs: Time for a new data presentation paradigm. *PLOS Biology* 13(4). e1002128. doi:10.1371/journal.pbio.1002128.

Westfall, Jacob, David A. Kenny & Charles M. Judd. 2014. Statistical power and optimal design in experiments in which samples of participants respond to samples of stimuli. *Journal of Experimental Psychology: General* 143. 2020–2045. doi:10.1037/xge0000014.

Westfall, Jacob & Tal Yarkoni. 2016. Statistically controlling for confounding constructs is harder than you think. *PLOS ONE* 11(3). e0152719. doi:10.1371/journal.pone.0152719.

Wickham, Hadley. 2014. Tidy data. *Journal of Statistical Software* 59. doi:10.18637/jss.v059.i10.

Wilner, Sean, Katherine Wood & Daniel Simons. 2018. Complete recovery of values in Diophantine systems (CORVIDS). doi:10.31234/osf.io/7shr8. PsyArXiv preprint.

Wood, Simon N. 2006. *Generalized additive models: An introduction with r*. Boca Raton, FL: Chapman & Hall/CRC.

Wurm, Lee H. & Sebastiano A. Fisicaro. 2014. What residualizing predictors in regression analyses does (and what it does *not* do). *Journal of Memory and Language* 72. 37–48. doi:10.1016/j.jml.2013.12.003.

Yarkoni, Tal & Jacob Westfall. 2017. Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives in Psychological Science* 12(6). 1100–1122. doi:10.1177/1745691617693393.

York, Richard. 2012. Residualization is not the answer: Rethinking how to address multicollinearity. *Social Science Research* 41. 1379–1386. doi:10.1016/j.ssresearch.2012.05.014.

Zuur, Alain F., Elena N. Ieno, Neil J. Walker, Anatoly A. Saveliev & Graham M. Smith. 2009. *Mixed effects models and extensions in ecology with R*. New York: Springer.