

Universität
Basel

IDS

LEIBNIZ-INSTITUT FÜR
DEUTSCHE SPRACHE

Statistik mit R in der Linguistik

Eine Einführung

Sascha Wolfer

 @sascha_wolfer

Zu dieser Präsentation

- Dies ist das Folienmaterial für den Kurs "Statistik mit R für die Linguistik: Eine Einführung" gegeben im HS 2022 an der Universität Basel von Sascha Wolfer
- Alle Bilder sind, soweit nicht anders ausgewiesen, von der Stockphoto-Seite unsplash.com heruntergeladen.
- Wenn Sie Teile dieser Präsentation verwenden möchten, referenzieren Sie bitte entsprechend. Der DOI für diese Materialien lautet **10.5281/zenodo.7431504**

Ihre Lernziele

R verstehen

Sie verstehen die Grundlagen der Statistikumgebung R und die dazugehörige Entwicklungsumgebung RStudio.

Daten explorieren

Sie können Datensätze einlesen und explorieren.

Deskriptive Statistik

Sie können einfache Verfahren der statistischen Beschreibung von Stichproben verstehen und anwenden.

Inferenzstatistik

Sie können einfache Verfahren verstehen und anwenden, mit denen Sie von einer Stichprobe auf eine Grundgesamtheit schließen können.

Daten visualisieren

Sie können einfache Verfahren der Datenvisualisierung verstehen und anwenden.



Prolog: Grundbegriffe empirischer Forschung

Popper, K. R. (2016). *Freiheit und intellektuelle Verantwortung: Politische Vorträge und Aufsätze aus sechs Jahrzehnten*. Mohr Siebeck.



Was würden Sie sagen/denken, wenn ich Ihnen die folgende Aufgabe gäbe?

Gehen Sie auf den
Marktplatz und
beobachten Sie dort die
Menschen!

- Was soll ich beobachten?
- Worauf soll ich achten?
- Was will er denn überhaupt wissen?

„Ohne Problem keine Beobachtung.

- Karl Popper

Fragestellung, Theorie, Hypothese, Beobachtung

- Aus einer **Fragestellung** entsteht eine empirische **Theorie**.
 - Theorie: System logisch konsistenter Aussagen, das an dem Ausschnitt der Realität, den die Theorie zu erklären versucht, scheitern kann.
 - Aus der Theorie abgeleitete **Hypothesen** müssen sich also an der Welt messen lassen. Eine Theorie ohne Bezug zur Welt ist bedeutungslos.
- **Beobachtungen** verbinden Theorien mit der Welt.

Fragestellung, Theorie, Hypothese, Beobachtung

Fragestellung

Welchen Effekt hat
bilinguale Erziehung auf
den Spracherwerb des
Kindes?

Theorie

Interferenzen zwischen
den Sprachen

Hypothese

Bilingual erzogene Kinder
brauchen länger für den
Spracherwerb.

Beobachtung

Vergleichende Studie
zwischen ein- und
mehrsprachig erzogenen
Kindern.

Hypothesen

- Eine Hypothese im empirischen Sinn ist eine Aussage in Form einer **überprüfbaren** Behauptung.
- Wissenschaftliche Hypothesen nach Bortz & Döring (2016):
 - **Allgemeingültig**, über den Einzelfall oder ein singuläres Ereignis hinausgehende Behauptung
 - Kann in einen sinnvollen **Konditionalsatz** überführt werden ("wenn ... dann", "je ... desto ...")
 - Muss potentiell **falsifizierbar** sein. Es müssen Ereignisse denkbar sein, die dem Konditionalsatz widersprechen.

Hypothesen

gerichtete Hypothese

Rauchen kann gesundheitsschädlich sein.

Menschen, die rauchen, haben mehr schwere Krankheiten als Menschen, die nicht rauchen.

gerichtete Hypothese

Die Verwendung von Adjektiven unterscheidet sich zwischen Korpora.

In Korpus A ist die relative Frequenz von Adjektiven höher als in Korpus B.

ungerichtete Hypothese

Bestimmte Wörter werden in Wörterbüchern deutlich häufiger nachgeschlagen als andere.

Die Nachschlagehäufigkeit eines Worts hängt mit der Auftretensfrequenz des Wort in der Sprache zusammen.

Inhaltliche und statistische Hypothesen

- Inhaltliche Hypothesen müssen später in **statistische** Hypothesen umformuliert bzw. formalisiert werden.
 - Hypothese darüber, welchen Effekt ein bestimmter statistischer Test anzeigen sollte, wenn die inhaltliche Hypothese korrekt ist.

Die Nachschlagehäufigkeit eines Worts hängt mit der Auftretensfrequenz des Wort in der Sprache zusammen.

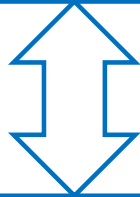
Sagt man die Nachschlagehäufigkeit mit einem linearen *Regressionsmodell* aus der Korpusfrequenz vorher, zeigt sich ein statistisch signifikanter Effekt.

Es lässt sich eine signifikante *Korrelation* zwischen Nachschlagehäufigkeit und Korpusfrequenz nachweisen.

Nullhypothese und Alternativhypothese



~~Nullhypothese: Es gibt keinen Zusammenhang oder Unterschied (= Effekt)~~



- Die statistischen Test, die Sie hier lernen, prüfen immer die **Nullhypothese**.
- Wenn wir diese mit ausreichender Wahrscheinlichkeit ablehnen können, nehmen wir an, dass die **Alternativhypothese** gilt.
 - Alternativhypothese = Forschungshypothese.

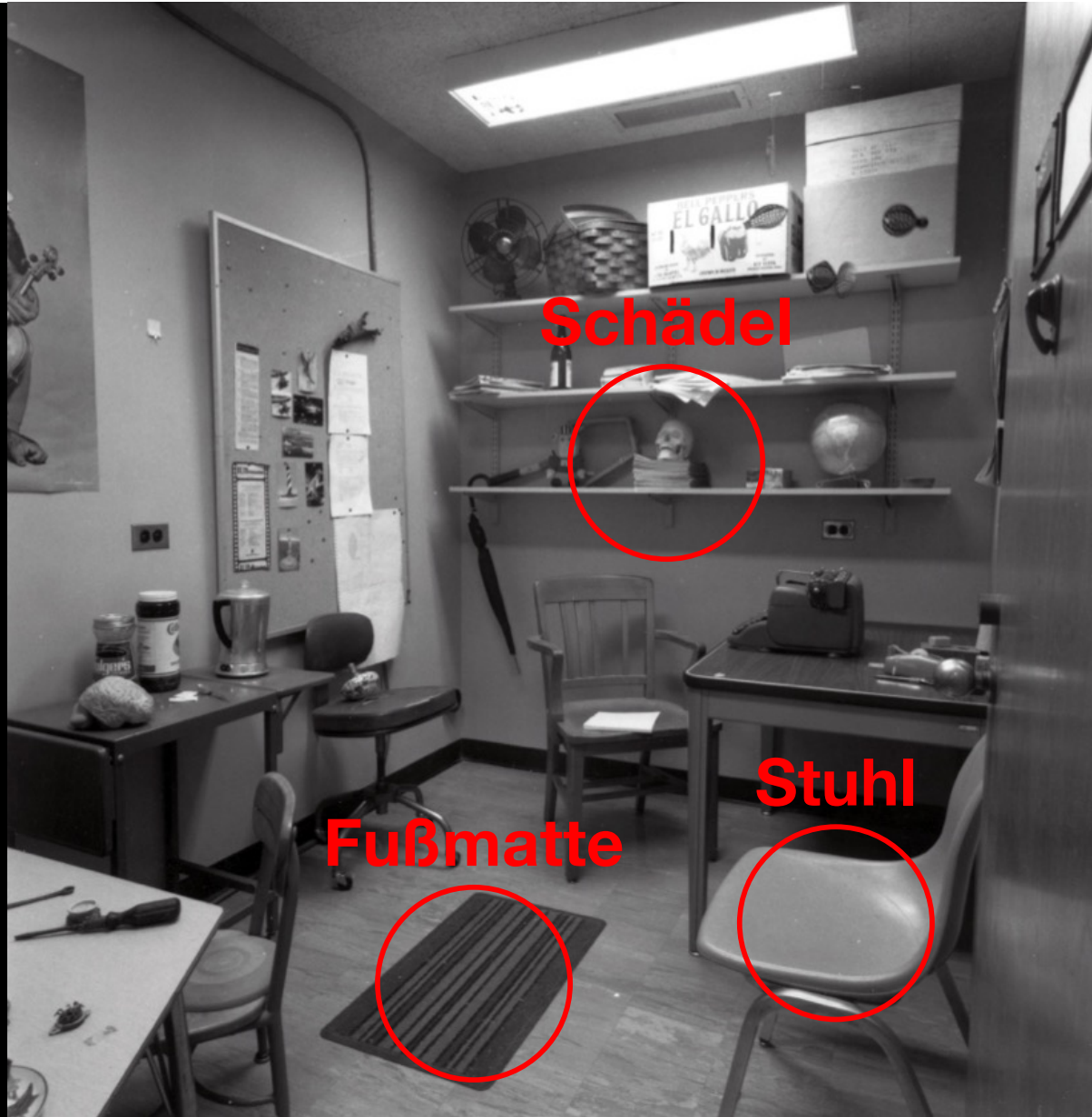


Alternativhypothese:
Es gibt einen Effekt.



Experiment

- Was haben Sie gesehen?
- Haben Sie einen Stuhl gesehen?
- Haben Sie ein Buch gesehen?
- Haben Sie einen Schädel gesehen?
- Haben Sie einen Computer gesehen?
- Haben Sie eine Fußmatte gesehen?



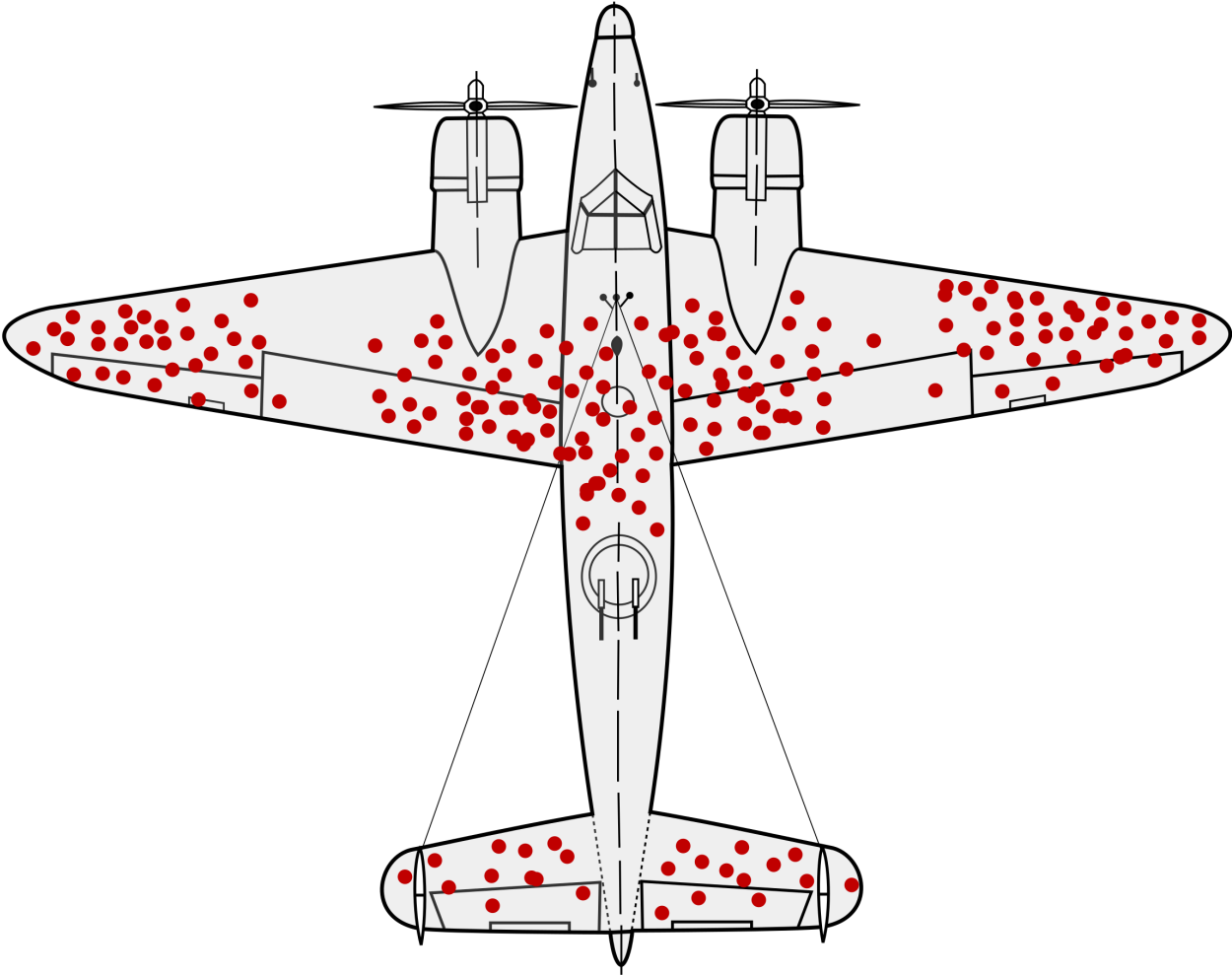
Schädel

Fußmatte

Stuhl

Verlässlichkeit von Beobachtungen

- Das Problem: Beobachtungen sind nicht immer verlässlich.
 - Ungenaue oder unangemessene Messinstrumente
 - Erinnerung rekonstruktiv und fehlerbehaftet
- Was bedeutet eine bestimmte Beobachtung?
- Welche Beobachtungen sind relevant?
 - Dominanz konkreter Ereignisse über abstrakte Grundraten
- *Biases* (z.B. *confirmation bias*, *survivorship bias*)





Gütekriterien empirischer Forschung

Objektivität

Die Testergebnisse sind unabhängig von der Person, die den Test durchführt.

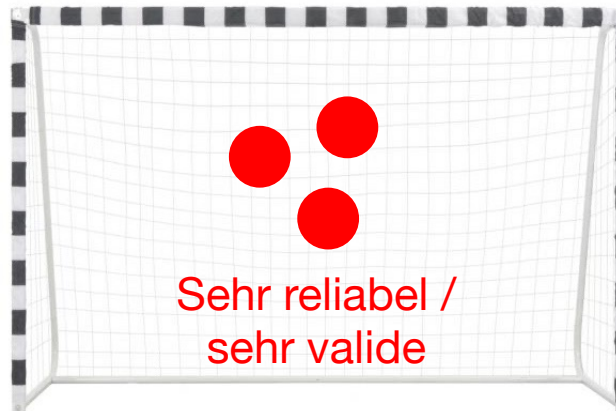


Nicht reliabel /
nicht valide



Reliabilität

Es wird möglichst genau gemessen.



Validität

Der Test misst das, was er zu messen vorgibt.



Sehr reliabel /
nicht valide

Objektivität

- **Durchführung**objektivität: Ergebnisse sind unabhängig von der forschenden Person.
 - Gegenbeispiel: Männer geben systematisch niedrigere Werte des Schmerzempfindens gegenüber weiblichen Testleiterinnen an.
- **Auswertung**objektivität: Gleiches Verhalten der Testpersonen wird gleich ausgewertet.
 - Nicht gegeben: Zwei Dozierende bewerten dieselbe Klausur. Dozent:in A benutzt vorgefertigte Stichwortlisten; Dozent:in B such nach inhaltlichen Argumentationsfehlern.
- **Interpretations**objektivität: Gleiche Ergebnisse werden gleich interpretiert.
 - 15 Punkte in einer Klausur bedeutet immer "sehr gut". Unter 5 Punkte bedeutet immer "nicht bestanden".

Objektivität

- **Klare und explizite Vorgaben** können die Objektivität erhöhen.
- Haben Sie eine Idee, wo in der Linguistik solche Vorgaben zum Einsatz kommen?

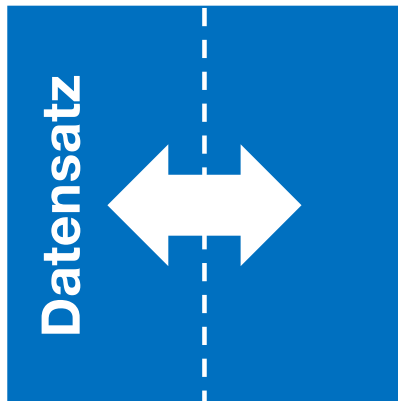
Reliabilität

- **Retest**-Reliabilität: Unter identischen Bedingungen muss bei wiederholten Messungen das gleiche Ergebnis herauskommen.
- **Interrater**-Reliabilität: Mehrere Personen müssen zum gleichen Ergebnis kommen (bspw. bei Annotationen oder Einschätzungen).
- **Split-half**-Reliabilität: Wenn ich den Datensatz in zwei Hälften teile, sollten die statistischen Kennwerte in beiden Hälften ähnlich sein.

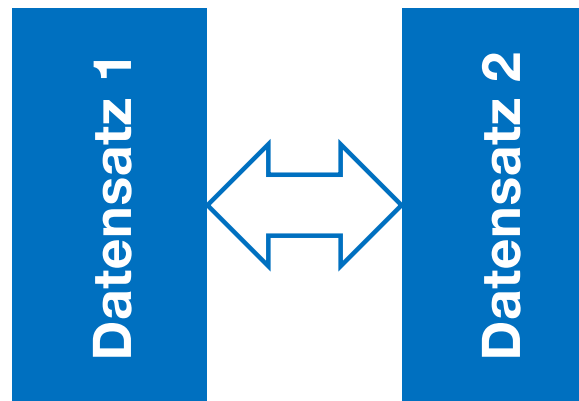
Reliabilität

After both raters finished the annotations, we identified all cases in which the raters disagreed. Mean Cohen's kappa for the initial annotation run (i.e. before any critical cases were discussed) was $k_{\text{mean}} = 0.819$. The mean kappa was 0.900 for Change/Modification and 0.740 for Improvement/Success. Carletta (1996: 252) referring to Krippendorff (1980), describes kappa scores above 0.8 as "good reliability".

Reliabilität kann man berechnen.
Hier: Interrater-Reliabilität.



Split-half-Reliabilität



Retest-Reliabilität

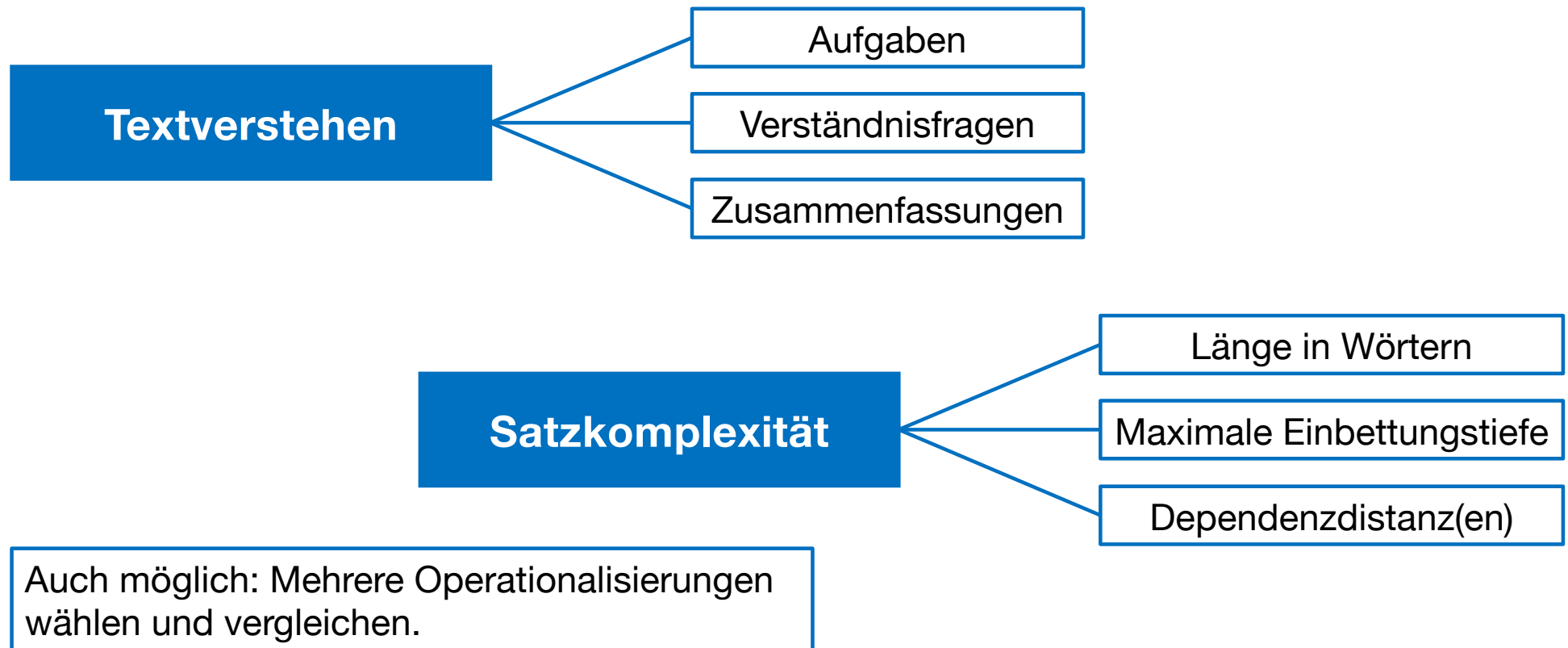
Validität

- **Interne Validität:** Wie gut misst der Test unter Laborbedingungen das, was er messen soll?
 - Laborbedingungen: konstant und kontrolliert
- **Externe (ökologische) Validität:** Sind die Ergebnisse des Tests auf die "wahre Welt" übertragbar?
 - Typischerweise: *Trade-off* zwischen externer und interner Validität
- **Konstruktvalidität:** Wie gut ist das zu messende Merkmal operationalisiert (= messbar gemacht)?

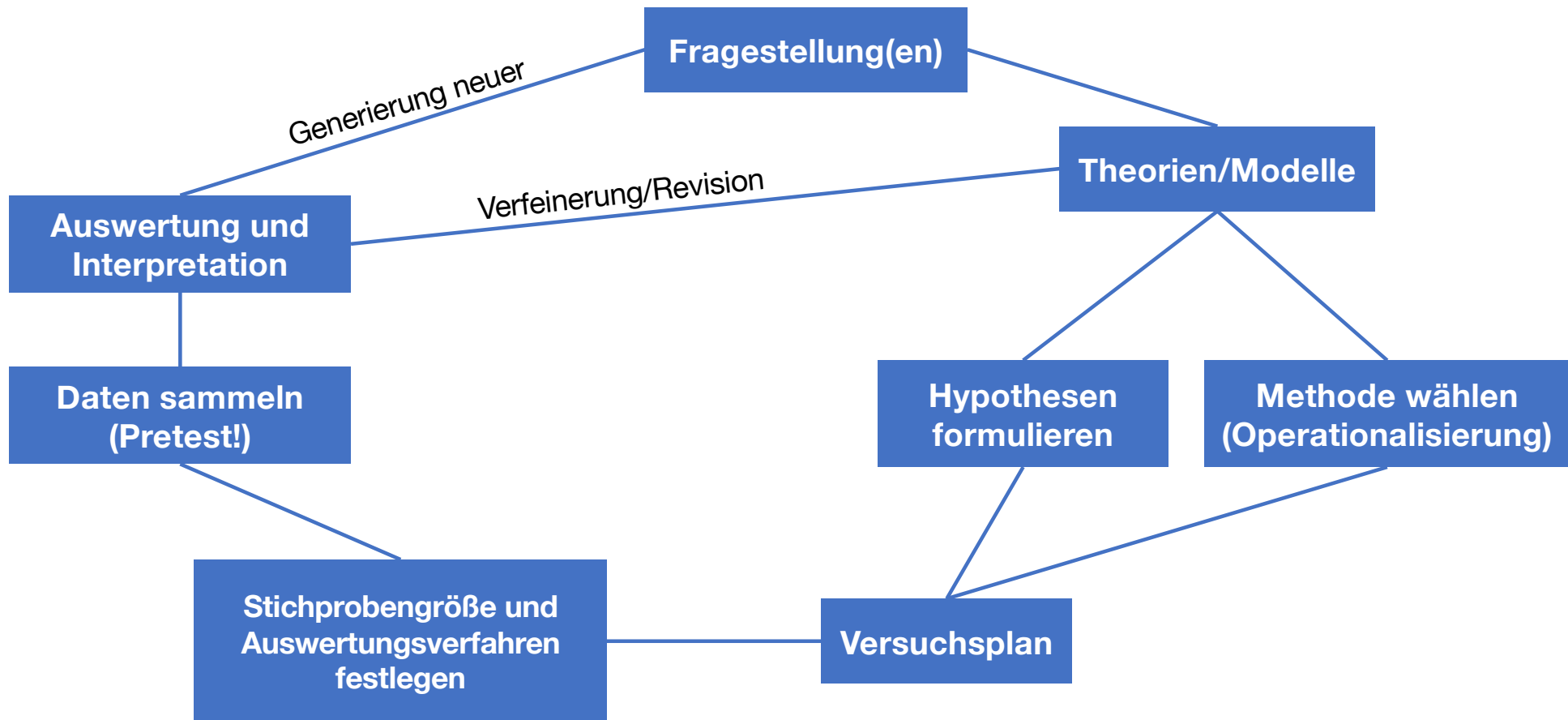
Operationalisierung

- Wie messen wir die Variable, die uns interessiert?
 - Ergebnis sollte ein konkretes Messergebnis sein (Zahl, Kodierung).
- Recherche notwendig:
 - Wie haben andere Forschungsteams dieselbe Variable gemessen?
 - Standardisierte Verfahren?
- Oft mehrere Operationalisierungen denkbar
- Viele wissenschaftliche Diskussionen drehen sich um die geeignete Form der Operationalisierung.

Operationalisierung: Beispiele



Kreislauf empirischer Forschung



Begriffe

Fragestellung

Nullhypothese

Konstruktvalidität

Theorie

Alternativhypothese

Operationalisierung

Hypothese

Objektivität

Falsifizierbarkeit

Reliabilität

**gerichtete /
ungerichtete
Hypothese**

**interne / externe
Validität**



Fragen?

Answers 1km →





R: Die Grundlagen

R: Die Grundlagen

- R und RStudio
- RStudio-Oberfläche
- Zusatzpakete
- Datentypen in R
- Indizierung
- Funktionen und Prädikate
- Daten einlesen
- Daten speichern



R und RStudio



- R: Programmier- / Scriptsprache
- RStudio: IDE (= integrated development environment)
 - erleichtert uns die Arbeit mit R
 - u.a. mit Scripteditor, Workspace-Übersicht, integrierter Hilfe, Speichern von Abbildungen, Syntax Highlighting usw.
- R ist verfügbar auf CRAN (= comprehensive R archive and network):
<https://cloud.r-project.org>
- RStudio: www.rstudio.com → Download

The image shows the RStudio interface with four panels highlighted by red boxes:

- Script Editor:** Contains R code:


```
1 norm <- rnorm(10000)
2 plot(density(norm))
3
```
- Environment / History:** Shows the workspace with a variable 'norm' of type 'double' and size 10000. The history shows the execution of the code in the script editor.
- Console:** Displays the R startup message and the execution of the code from the script editor:


```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R ist freie Software und kommt OHNE JEGliche GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()'
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

Welcome to R. This is .Rprofile V1
> plot(rnorm(10000))
> plot(density(rnorm(10000)))
> norm <- rnorm(10000)
> plot(density(norm))
>
```
- Plots / Hilfe:** Shows a density plot titled 'density.default(x = norm)'. The x-axis ranges from -4 to 4, and the y-axis (Density) ranges from 0.0 to 0.4. The plot shows a normal distribution curve centered at 0. Below the plot, it indicates 'N = 10000 Bandwidth = 0.1434'.

RStudio-Oberfläche

- Sammeln Sie Ihre Befehle im **Scripteditor!**
 - Speichern Sie Ihre Scripts mit der Endung `.R`
 - Schicken Sie Befehle an die Konsole mit `Cmd+Enter` / `Strg+Enter`
- **Kommentieren** Sie Ihren Code mit `#`
 - Alles hinter `#` wird nicht von R ausgewertet.
- In der **Konsole** finden Sie die Rückgabewerte von R.
 - Ergebnisse Ihrer Befehle
 - Nicht alle Befehle haben Rückgabewerte.
- Die **Hilfe** rufen Sie mit `?<Befehl>` auf.
 - Beispiel: `?mean`



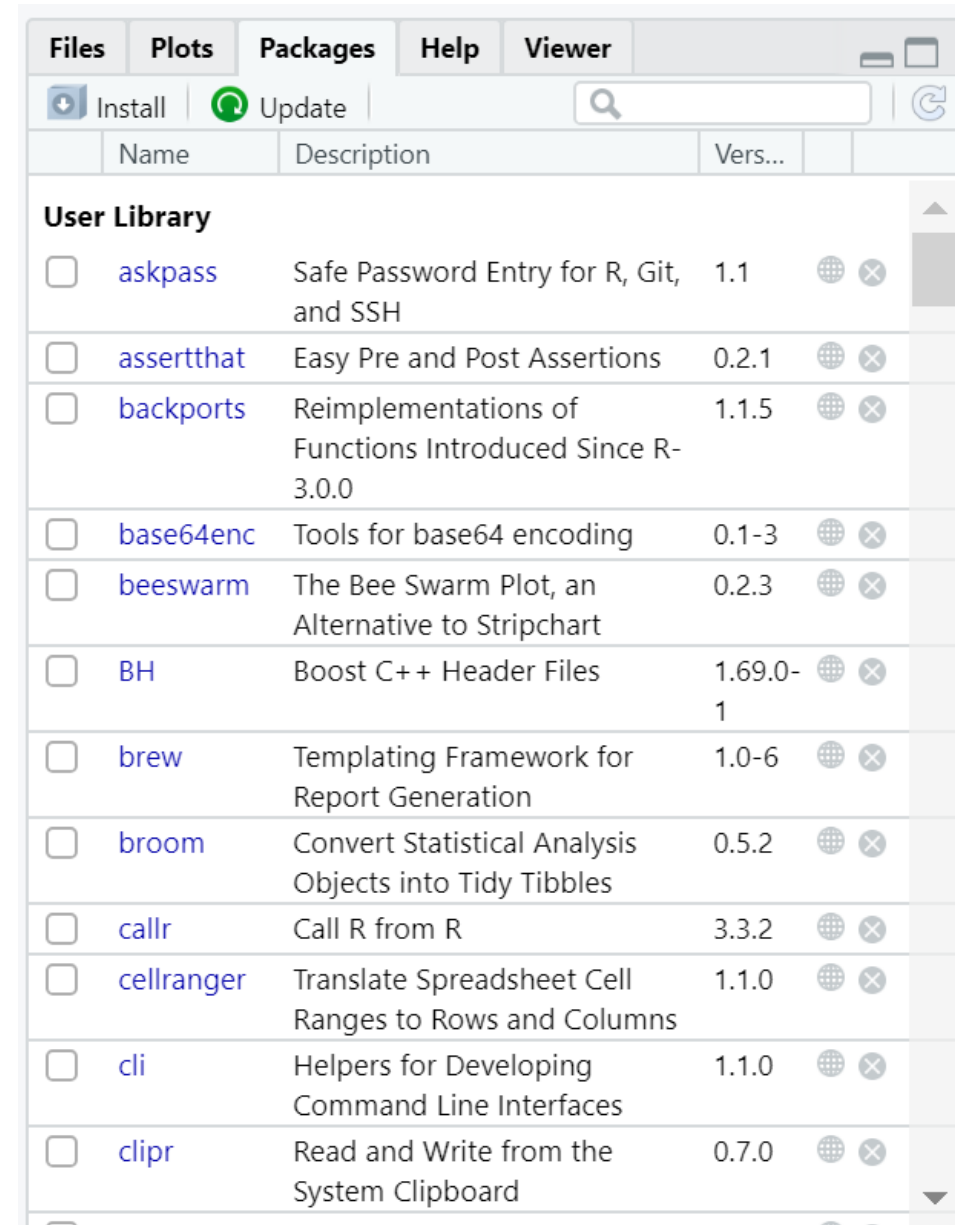
Übung



- Berechnen Sie in R das Ergebnis:
 - 1848 geteilt durch 44
 - 55,5 geteilt durch 12 (Dezimaltrennzeichen in R ist der Punkt)
 - Quadratwurzel aus 36 hoch 3 – Quadratwurzel: Funktion `sqrt(x)`
 - Mittelwert der Zahlen 20, 45 und 1, ohne die Funktion `mean(x)`

Zusatzpakete

- R kann mit (sehr vielen) Zusatzpaketen erweitert werden.
- Befehl:
`install.packages("<Paketname>")`
- "Packages" pane in RStudio
- Packages werden häufig mit geschweiften Klammern genannt:
{beeswarm}
- `data.table::fwrite()` bedeutet: Die Funktion `fwrite()` aus dem Paket {data.table}
- Installierte Pakete laden mit
`library(<Paketname>)`



The screenshot shows the RStudio interface with the 'Packages' pane open. The pane displays a list of packages under the 'User Library' section. Each row includes a checkbox, the package name, a description, the version number, and icons for refresh and delete. The packages listed are:

	Name	Description	Vers...		
<input type="checkbox"/>	askpass	Safe Password Entry for R, Git, and SSH	1.1	🌐	✕
<input type="checkbox"/>	assertthat	Easy Pre and Post Assertions	0.2.1	🌐	✕
<input type="checkbox"/>	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.5	🌐	✕
<input type="checkbox"/>	base64enc	Tools for base64 encoding	0.1-3	🌐	✕
<input type="checkbox"/>	beeswarm	The Bee Swarm Plot, an Alternative to Stripchart	0.2.3	🌐	✕
<input type="checkbox"/>	BH	Boost C++ Header Files	1.69.0-1	🌐	✕
<input type="checkbox"/>	brew	Templating Framework for Report Generation	1.0-6	🌐	✕
<input type="checkbox"/>	broom	Convert Statistical Analysis Objects into Tidy Tibbles	0.5.2	🌐	✕
<input type="checkbox"/>	callr	Call R from R	3.3.2	🌐	✕
<input type="checkbox"/>	cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0	🌐	✕
<input type="checkbox"/>	cli	Helpers for Developing Command Line Interfaces	1.1.0	🌐	✕
<input type="checkbox"/>	clipr	Read and Write from the System Clipboard	0.7.0	🌐	✕

Datentypen

- Programmiersprachen brauchen fest definierte Datentypen, damit jederzeit klar ist, was mit bestimmten Objekten gemacht werden kann und was nicht.

Elementare Datentypen

- Zahlen
- Zeichenketten
- Wahrheitswerte

Komplexe Datentypen

Zusammengesetzt aus anderen Elementen

- Vektoren
- Dataframes
- (Matrizen)
- (Listen)

Elementare Datentypen in R

Zahlen

numerical / integer

Zeichenketten

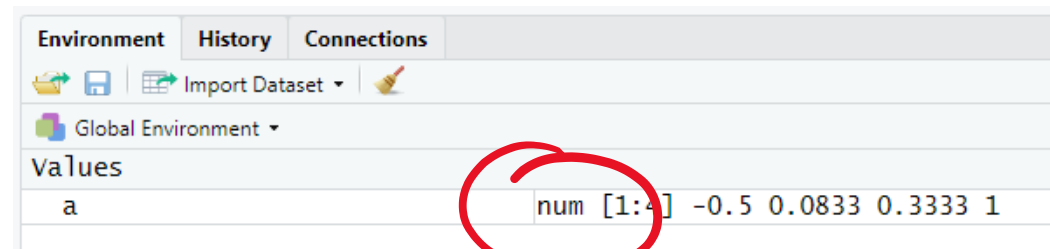
character / string

Wahrheitswerte

logical / boolean

Zahlen

- Alle numerischen Werte
- Spezialfälle (auch Zahlen!):
 - **NaN**: *not a number*
 - **Inf** / **-Inf**: (minus) unendlich
- Abkürzung: **num** für numerical
 - oder: **int** für integer → ganze Zahlen





Zeichenketten

- Alle Zeichen(folgen)
- Stehen in Anführungszeichen
 - "42" ist eine Zeichenkette, keine Zahl
 - 42 ist eine Zahl
 - "" ist eine leere Zeichenkette
 - " " ist eine Zeichenkette, die ein Leerzeichen enthält.
- Abkürzung: chr (*character*)

Wahrheitswerte

- Auch: *boolean / logical* (Abkürzung `logi`)
- **TRUE** (wahr) oder **FALSE** (falsch)
 - Können abgekürzt werden mit **T** und **F**.
- Repräsentieren "ja" und "nein" bzw. "wahr" und "falsch"
 - Werden oft von R als Antworten auf Fragen zurückgegeben.
 - **5 < 10** → T
- **"T"** ist eine Zeichenkette, **T** ist ein Wahrheitswert.

Variablen / Objekte

- In Variablen speichern wir Werte, damit wir sie wiederverwenden können.
- Alle Variablen erscheinen im Environment von RStudio.
 - Auch: *workspace*
- Zuweisungsoperator: `<-`
 - `a <- 3` speichert in der Variable `a` die Zahl 3.
 - `b <- "drei"` speichert in der Variable `b` die Zeichenkette "drei".
- Nennen Sie Variablen, wie Sie wollen, aber...
 - kein Variablenname darf mit einer Zahl beginnen.
 - keinen Bindestrich im Namen verwenden (= minus).



Variablen / Objekte

- Wenn Sie wissen möchten, was in einer Variable gespeichert ist, geben Sie einfach den Variablennamen auf der Konsole ein.
 - ... oder beobachten Sie Ihren Workspace.
- Variablenzuweisung hat keinen Rückgabewert auf der Konsole. Variable hat aber den neuen Wert.
 - Beobachten Sie Ihren Workspace.

Verändern von Variablen

- Achtung! Eine Variable verändert sich immer nur dann, wenn Sie das explizit schreiben (= "nicht-destruktive" Operationen).

```
a <- 2
```

```
a + 4
```

```
[1] 6
```

```
a
```

```
[1] 2
```

```
a <- a + 4
```

```
a
```

```
[1] 6
```

Umwandeln von Objekten

- Manche Objekte können verlustfrei in einen anderen Typ umgewandelt werden.
- Grundsätzlich gilt: Alles kann verlustfrei in eine Zeichenkette umgewandelt werden: `as.character(x)`
- In Zahl umwandeln: `as.numeric(x)`
- In Wahrheitswert: `as.logical(x)`

Übung



- Beobachten Sie bei jedem Schritt der Übung, was geschieht!
 - Speichern Sie in der Variable `test` die Zeichenkette "1".
 - Addieren Sie `test` mit 3.
 - Wandeln Sie `test` in eine Zahl um.
 - Addieren Sie `test` mit 3 und speichern Sie das Ergebnis wieder in `test`.
 - Ziehen Sie 4 von `test` ab (und speichern Sie das Ergebnis wieder in `test`).
 - Wandeln Sie `test` in einen Wahrheitswert um.

Begriffe



Scripteditor

Elementare D.~

Variablen / Objekte

Konsole

Komplexe D.~

Workspace

Zahlen

#

Zeichenketten

Datentypen

Wahrheitswerte

Zusammenfassung

- Schreiben Sie Ihre Scripts in den Scripteditor, kommentieren Sie (viel!) mit #.
- In der Konsole finden Sie die Rückgabewerte Ihrer Befehle.
- Elementare Datentypen: Zahlen, Zeichenketten, Wahrheitswerte.
- In Variablen können wir mit <- Werte speichern. Wenn Sie eine Variable verändern möchten, müssen Sie neu zuweisen.
- Sie können elementare Datentypen mit den Funktionen as. ... ineinander umwandeln.
 - Alles kann verlustfrei in Zeichenketten umgewandelt werden.

Fragen?



Komplexe Datentypen

Vektoren

Dataframes

Matrizen

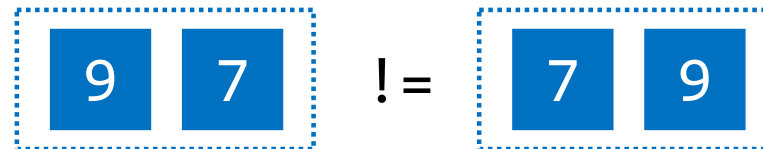
Listen



Vektoren



- Vektoren sind Reihen von Elementen **eines** Typs.
 - Numerische Vektoren
 - Character-Vektoren
 - Logische Vektoren
 - + Faktor-Vektoren
- Vektoren sind **geordnet**, die Reihenfolge ist wichtig.



Vektoren

- Erstellen Sie Vektoren mit `c()` – *construct*
 - `c("a", "b", 3) → [1] "a" "b" "3"`
- Andere Funktionen/Operatoren geben ebenfalls Vektoren zurück.
 - `5:9 → [1] 5 6 7 8 9`
 - `seq(5, 8, 0.5) → [1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0`
 - `rep(T, 5) → [1] TRUE TRUE TRUE TRUE TRUE`
- Sie können Vektoren miteinander verrechnen.
 - `c(3, 5) + c(7, 9) → [1] 10 14`



Recycling

$$\begin{array}{|c|} \hline 3 \\ \hline 5 \\ \hline \end{array} + \begin{array}{|c|} \hline 7 \\ \hline 9 \\ \hline \end{array} = \begin{array}{|c|} \hline 10 \\ \hline 14 \\ \hline \end{array}$$

$$c(3, 5) + c(7, 9)$$

Vektoren werden – falls nötig – *recyclet*. Sie werden also so lange wiederholt, wie es nötig ist.

$$c(3, 5, 9, 10) + c(7, 9)$$

$$\begin{array}{|c|} \hline 3 \\ \hline 5 \\ \hline 9 \\ \hline 10 \\ \hline \end{array} + \begin{array}{|c|} \hline 7 \\ \hline 9 \\ \hline 7 \\ \hline 9 \\ \hline \end{array} = \begin{array}{|c|} \hline 10 \\ \hline 14 \\ \hline 16 \\ \hline 19 \\ \hline \end{array}$$

Recycling: Achtung!

Das ist keine
Fehlermeldung!

```
> c(3,5,9,10) + c(7,9,10)
[1] 10 14 19 17
Warnmeldung:
In c(3, 5, 9, 10) + c(7, 9, 10) : Länge des längeren Objektes
ist kein Vielfaches der Länge des kürzeren Objektes
```

- R möchte Vektoren gerne komplett recyceln.
 - Hier kann der kürzere Vektor aber nur zum Teil recyclet werden.
- Wenn das geschieht, wird eine Warnmeldung ausgegeben.
 - Bei **Warnmeldungen** wird trotzdem ein Ergebnis zurückgegeben, bei **Fehlermeldungen** nicht.

Faktor-Vektoren

Um zu verstehen, was Faktor-Vektoren sind und wozu sie da sind, benötigen wir Kenntnisse über **Skalenniveaus**.

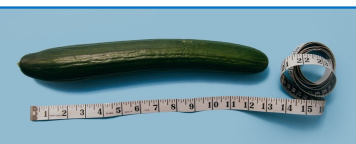


Skalenniveaus

Verhältnisskala

Intervallskala

Metrische Skala



Ordinalskala



Nominalskala



Nominalskala



- Notiert werden die **Gleich- und Verschiedenheit** von Elementen.
 - Keine Rangfolge
 - Keine Abstände
- Möglich: Auszählen von Häufigkeiten
- Beispiele: Wort-Type, Erstsprache, Korpusquelle
- Sonderfall: **Binäre** Skala
(richtig/falsch, vorhanden/nicht vorhanden)

Ordinalskala



- **Zusätzlich** ablesbar: Rangfolge von Elementen
 - Keine Abstände
- Wir können zusätzlich alle Analysen/Maßzahlen anwenden, die auf der Rangfolge beruhen (z.B. **Median**).
- Beispiel: Mohs'sche Härteskala von Mineralien

Intervallskala



- **Zusätzlich** können wir ablesen, wie weit Datenpunkte auf der Skala voneinander entfernt sind.
 - Kein absoluter Nullpunkt → keine Verhältnisse
- Wir können alle relevanten statistischen Verfahren anwenden.
- Beispiel: IQ

Verhältnisskala



- Einziger Unterschied zur Intervallskala: Absoluter Nullpunkt
- **Zusätzlich** können wir Aussagen über Verhältnisse machen.
 - "X ist doppelt so schwer wie Y."
- Unterschied in der Praxis (fast) nie relevant, daher Zusammenfassung von Intervall- und Verhältnisskala als **metrische Skala**
- Beispiel: Gewicht, Reaktionszeit

Nominalskala / Ordinalskala / Metrische Skala?

- Wortart (*parts-of-speech*) von Wörtern
- Länge eines Satzes in Wörtern
- Organ, in dem ein Wort veröffentlicht wurde
- Ist Deutsch eine Erstsprache von Teilnehmenden?
- Antworten auf Likert-Skalen
- Worthäufigkeiten
- Anzahl Texte, in denen ein Wort vorkommt
- Sprechrate (Silben pro Sekunde)

Jetzt aber: Faktor-Vektoren

```
> factor(c("A", "A", "B", "C", "C", "C"))  
[1] A A B C C C  
Levels: A B C
```

- Faktorvektoren werden in R dazu verwendet, **nominal-** und **ordinalskalierte** Daten zu notieren.
- Ordinal: Sonderfall *ordered factors* – erstellen mit `ordered(x)`
 - *Ordered factors* braucht man sehr selten.
- *Levels*: Alle Stufen eines Faktors



Faktor-Vektoren: Achtung!

- Weisen Sie einer Variable den folgenden numerischen Vektor zu:
 - `[42, -1, 8, 99]`
- Wandeln Sie die Variable in einen Faktor-Vektor um. Sie benötigen dazu die Funktion `as.factor()`. Geben Sie die Variable aus.
- Wandeln Sie die Variable mit `as.numeric()` wieder zurück in einen numerischen Vektor um.
- Geben Sie die Variable wieder aus. Was fällt Ihnen auf?

Begriffe



Vektoren

Skalenniveau

Faktor-Vektoren

Nominalskala

Recycling

Ordinalskala

Warnmeldung

Metrische Skala

Fehlermeldung

Binäre Skala

Vektoren: Zusammenfassung

- Vektoren sind ein komplexer Datentyp.
- Vektoren sind geordnete Reihen **eines** elementaren Datentyps.
- Vektoren werden Element für Element miteinander verrechnet.
 - u.U. werden Vektoren dabei recyclet.
 - Nebenbei: Wir müssen unterscheiden zwischen **Fehlermeldungen** und **Warnmeldungen**.
- Faktor-Vektoren kodieren nominal- und ordinalskalierte Variablen und haben Stufen (*levels*).

Fragen?



Dataframes

Dataframes kann man als nebeneinander gestapelte Vektoren begreifen.

- Datentabellen: Ein Fall pro Zeile
 - Fall: Versuchsdurchlauf, Wort, Zielkonstruktion, ...
- In den Spalten stehen Informationen zu jedem Fall.

Word	POS	WordLen	LogFreq	FuncWord
Die	ART	3	18.7	T
Biene	NN	5	8.97	F
summt	VVFIN	5	7.67	F
laut	ADV	4	13.9	F

Dataframes

- Dataframes sind für alle wichtigen Operationen der Datenanalyse der Ausgangspunkt.
 - Gruppenvergleiche
 - Zusammenhänge
 - Aggregationen
 - Visualisieren
 - ...
- Man kann Dataframes mit `data.frame()` erstellen, meist sind sie aber Ergebnisse von **Einleseoperationen**.
- Mit `head(<dataframe>)` werden die ersten 6 Zeilen eines Dataframes ausgegeben.

Matrizen

- Matrizen (Singular: Matrix) sind **mehrdimensionale Vektoren**.
 - Auch eine Matrix kann *nur einen Datentyp* enthalten!
- Zweidimensionale Matrizen haben auch Zeilen und Spalten.
 - Matrizen können aber beliebig viele Dimensionen haben.

```
> matrix(1:15, ncol = 3)      > matrix(1:15, ncol = 4)
      [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

Listen

- Listen sind geordnete Sequenzen von beliebigen Datentypen.
 - Im Gegensatz zu Vektoren können Listen auch komplexe Datentypen enthalten. Listen können also auch wiederum Listen enthalten.

```
> list(5:10, data.frame(A = 1:3, B = c("a", "b", "c")), rep(T, 4))
```

```
[[1]]
```

```
[1] 5 6 7 8 9 10
```

1. Element: Numerischer Vektor

```
[[2]]
```

```
  A B
```

```
1 1 a
```

```
2 2 b
```

```
3 3 c
```

2. Element: Zweispaltiger Dataframe

```
[[3]]
```

```
[1] TRUE TRUE TRUE TRUE
```

3. Element: Vektor mit Wahrheitswerten



Indizierung

Indizierung

- Indizierung = Zugriff; wir greifen auf komplexe Datentypen zu, um Teile von ihnen zu extrahieren oder zu verändern.
- Vektoren: `[n]`
 - `vec[2]`: Zweites Element des Vektors `vec`
 - `vec[3:5]`: Drittes bis fünftes Element von `vec`
- Dataframes: `[<Zeile n>, <Spalte m>]` oder `$<Spaltenname>`
 - `df[5,]`: Fünfte Zeile von Dataframe `df`
 - `df[, 4]`: Vierte Spalte von Dataframe `df`
 - `df[2, 3]`: Element in der zweiten Zeile in der dritten Spalte
 - `df$wort`: Komplette Spalte `wort` (= Vektor)

Indizierung

- Wir können auch über **Namen** zugreifen (wie in `df$wort`):
 - `df[4, "freq"]`: Der vierte Eintrag in der Spalte `freq`
- Wir können außerdem über **Wahrheitswerte** zugreifen (das wird später nochmal wichtig!):

```
> vec <- c("das", "ist", "ein", "vektor")
> vec[c(T, T, F, T)]
[1] "das"      "ist"      "vektor"
```


Sortieren = Indizierung

- `order(<Vektor>)` gibt die Rangfolge der Elemente in einem Vektor zurück.
 - Bei Zeichenketten: alphabetisch, bei Zahlen: von klein nach groß
 - `order(c("D", "B", "C", "A"))` → [1] 4 2 3 1
- Deshalb können wir Dataframes sortiert ausgeben, wenn wir `order()` verwenden, um auf Zeilen zuzugreifen.
 - df aufsteigend nach Spalte freq sortiert ausgeben:
`df[order(df$freq),]`
 - Absteigend sortieren: `order(..., decreasing = T)`

Zusammenfassung

- **Vektoren, Dataframes, Listen und Matrizen** sind die wichtigsten komplexen Datentypen.
 - Dataframes kann man sich vorstellen als nebeneinander gestellte Vektoren.
 - Matrizen sind mehrdimensionale Vektoren.
 - Listen können alle anderen Datentypen enthalten, auch Listen selbst.
- Wir können auf diese zugreifen (sie indizieren) über
 - Zahlen (= Stellen/Indizes): `vec[3]` oder `df[3,]`
 - Namen: `df$wort` oder `df[5, "wort"]`
 - Wahrheitswerte: `vec[c(T, F)]`

Fragen?



WHY
NOW



Übung

- Erstellen Sie fünf Vektoren:
 - Vektor `user` mit den Werten `["km", "smv", "sw", "al"]`
 - Vektor `tweets` `[18948, 11314, 2440, 14610]`
 - Vektor `followers` `[3584, 3609, 719, 2543]`
 - Vektor `follows` `[1374, 548, 877, 1059]`
 - Vektor `face.in.profile` `[T, F, T, T]`
- Kombinieren Sie die Vektoren in einem Dataframe.
- Extrahieren Sie die erste Zeile.
- Extrahieren Sie die zweite und dritte Spalte.
- Extrahieren Sie den Wert 877.
- Sortieren Sie den Dataframe nach der Spalte `followers`.

Funktionen

- Befehle an R
- **Argumente:**
 - Mit **was** soll etwas gemacht werden?
 - **Was** soll genau getan werden?
 - Manche Argumente haben "Default"-Werte
- Rückgabe(-Wert): Ergebnis der Funktion
 - Der Rückgabewert wird im Terminal ausgegeben.
 - Manche Funktionen (z.B. Plots) haben keinen Rückgabewert, sondern werden wegen ihres **Nebeneffekts** benutzt.



Funktionen

- Funktionen können ineinander **verschachtelt** werden. Sie werden dann **von innen nach außen** evaluiert/ausgewertet.
 - `sqrt(sum(c(5, 3, 1)))`
 - `max(nchar(c("HGW", "MA", "GSWG")))`
 - `as.character(min(c(9, 8, -1)) + max(1:5))`
 - `c(9, 3, 42)[sqrt(mean(c(7, 9, 11)))]`

data.frame {base} **Paket**

R Documentation

Data Frames

Description

The function `data.frame()` creates data frames, tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R's modeling software.

Usage

Default-Werte

```
data.frame(..., row.names = NULL, check.rows = FALSE,  
           check.names = TRUE, fix.empty.names = TRUE,  
           stringsAsFactors = default.stringsAsFactors())
```

```
default.stringsAsFactors()
```

genaue Beschreibung der Argumente

Arguments

<code>...</code>	these arguments are of either the form <code>value</code> or <code>tag = value</code> . Component names are created based on the tag (if present) or the deparsed argument itself.
<code>row.names</code>	<code>NULL</code> or a single integer or character string specifying a column to be used as row names, or a character or integer vector giving the row names for the data frame.
<code>check.rows</code>	if <code>TRUE</code> then the rows are checked for consistency of length and names.
<code>check.names</code>	logical. If <code>TRUE</code> then the names of the variables in the data frame are checked to ensure that they are syntactically valid variable names and are not duplicated. If necessary they are adjusted (by make.names) so that they are.
<code>fix.empty.names</code>	logical indicating if arguments which are "unnamed" (in the sense of not being formally called as <code>someName = arg</code>) get an automatically constructed name or rather name <code>" "</code> . Needs to be set to <code>FALSE</code> even when <code>check.names</code> is false if <code>" "</code> names should be kept.
<code>stringsAsFactors</code>	logical: should character vectors be converted to factors? The 'factory-fresh' default is <code>TRUE</code> , but this can be changed by setting options(stringsAsFactors = FALSE) .

Details

A data frame is a list of variables of the same number of rows with unique row names, given class `"data.frame"`. If no variables are included, the row names determine the number of rows.

Einige Funktionen

<code>min()</code> / <code>max()</code>	Minimal- / Maximalwert in Vektor
<code>mean()</code>	Mittelwert
<code>median()</code>	Median
<code>nchar()</code>	Anzahl Zeichen in Zeichenkette(n)
<code>length()</code>	Anzahl Elemente in Vektor
<code>ncol()</code> / <code>nrow()</code>	Anzahl Spalten / Zeilen in Dataframe
<code>unique()</code>	Gibt jedes Element / jede Zeile nur einmal zurück
<code>summary()</code>	Gibt eine Zusammenfassung des Arguments

number of characters R

Alle News Shopping Videos Bilder Mehr Suchfilter

Ungefähr 1.050.000.000 Ergebnisse (0,37 Sekunden)

<https://stat.ethz.ch/html/nchar/> Diese Seite übersetzen

Count the Number of Characters (or Bytes or Width) - R

Count the **Number of Characters** (or Bytes or Width). Description. nchar takes a **character** vector as an argument and returns a vector whose elements contain ...

delete repeated elements R

Alle Videos News Shopping Bilder Mehr Suchfilter

Ungefähr 16.700.000 Ergebnisse (0,51 Sekunden)

To remove duplicates in R, **Use duplicated() method**: It identifies the duplicate elements. Using unique() method: It extracts unique elements. dplyr package's distinct() function: Removing duplicate rows from a data frame. 11.09.2021

remove-duplicates-in-r-with-e...

Duplicates in R - R-Lang

number of rows dataframe r

Alle Bilder Videos News Shopping Mehr Suchfilter

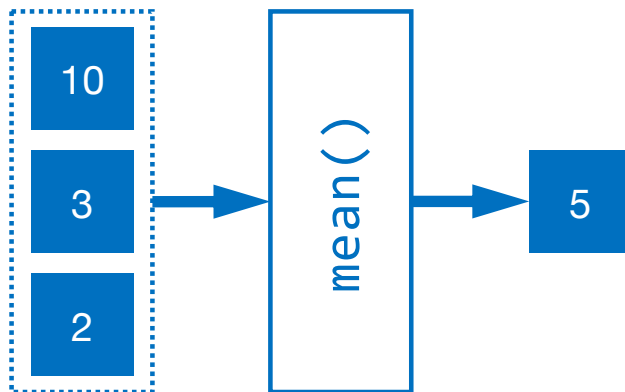
Ungefähr 11.200.000 Ergebnisse (0,62 Sekunden)

Meintest du: number of rows *data frame* r

To get number of rows in R Data Frame, **call the nrow() function and pass the data frame as argument to this function**. nrow() is a function in R base package.

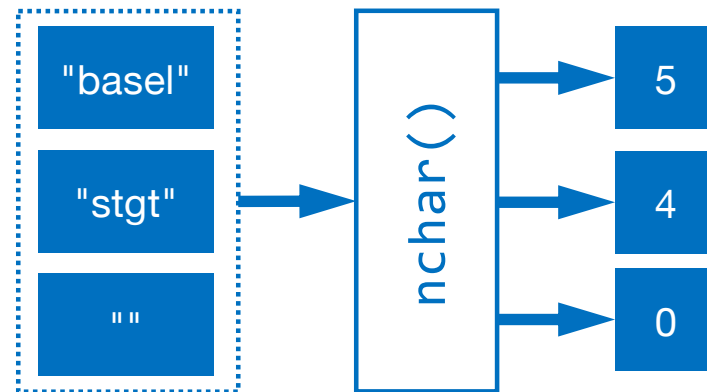
Rückgabewerte

```
mean(c(10, 3, 2))
```



Genau ein Rückgabewert

```
nchar(c("basel", "stgt", ""))
```



Ein Rückgabewert pro Element

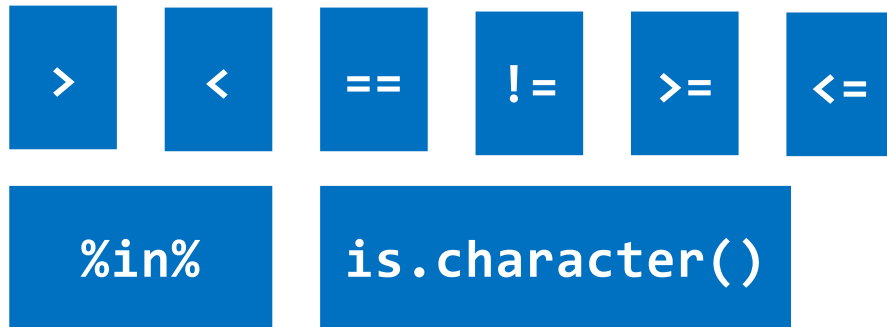
Zwischenfazit: Funktionen

Nebenbei: Man kann sich auch eigene Funktionen schreiben. Dazu aber später mehr.

- Funktionen sind Handlungsanweisungen mit **Argumenten**
 - Mit was wird etwas gemacht?
 - Was wird genau gemacht?
- Die meisten Funktionen haben **Rückgabewerte**.
 - Die Form der Rückgabe variiert je nach Funktion.
 - Die Rückgabewerte kann man als Argument für weitere Funktionen benutzen (Verschachteln von Funktionen).

Prädikate

- Prädikate sind ganz spezielle Funktionen, denn sie geben **immer** TRUE oder FALSE zurück.
- Prädikate sind also Fragen an R, ob etwas bestimmtes gilt.





Übung: Prädikate

- Ist 10 größer als 10?
- Ist 10 größer gleich 10?
- Befindet sich 10 in einem Vektor, der von 2 bis 11 geht?
- Befindet sich die Zeichenkette "a" im Character-Vektor ["uni", "base1"]?
- Ist 10 eine Zahl?
- Ist "10" eine Zahl?
- Ist *unendlich* eine Zahl?

Wozu braucht man Prädikate?

- Wir können außerdem über **Wahrheitswerte** zugreifen (das wird später nochmal wichtig!):

```
> vec <- c("das", "ist", "ein", "vektor")
> vec[c(T, T, F, T)]
[1] "das" "ist" "vektor"
```

Indizierung

- Wir können Fälle in Dataframes nach bestimmten Bedingungen **filtern**:

```
df[df$laenge >= 5,]
```

Gibt alle Fälle (Zeilen) zurück, bei denen die Spalte laenge größer 5 ist.

Fallunterscheidungen

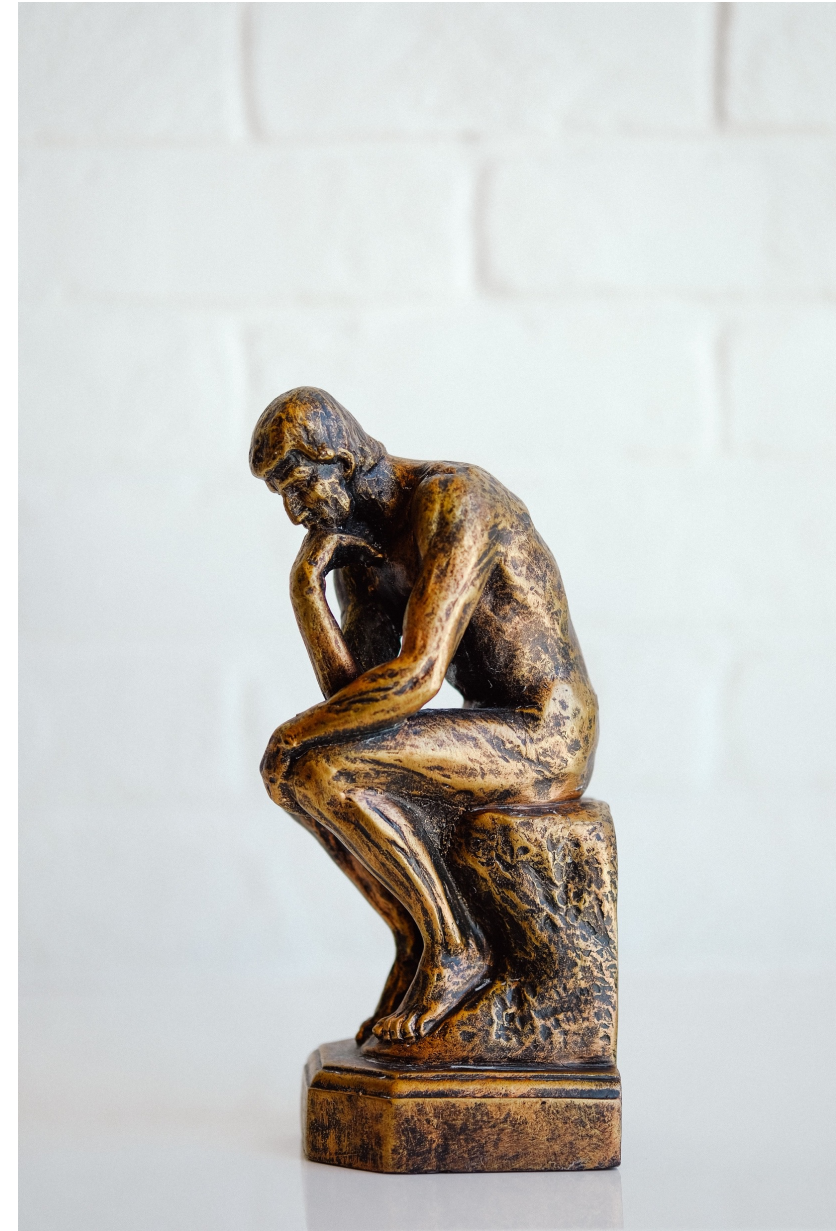
- Wir können Spalten in Abhängigkeit der Werte in anderen Spalten definieren:

```
df$wort_kat <- ifelse(df$laenge >= 5,
                      "lang", "kurz")
```

Wenn in Spalte laenge ein Wert größer gleich 5 steht, schreiben wir in Spalte wort_kat "lang", ansonsten "kurz".

Logische Operatoren

- Operatoren verknüpfen Wahrheitswerte
- Logisches UND: &
 - Beide Bedingungen müssen erfüllt sein.
- Logisches ODER: |
 - Mindestens eine Bedingung muss erfüllt sein.
- "Nicht"/Umkehren des Wahrheitswerts: !
 - Aus TRUE wird FALSE, aus FALSE wird TRUE.



Indizierung mit Prädikaten und Operatoren

- Welche Fälle werden ausgewählt?

```
df[df$pos == "NN",]
```

```
df[df$pos %in% c("NN", "NE") & df$laenge > 10,]
```

```
df[df$laenge %in% 5:10 & df$wort != "gehen",]
```

```
df[df$pos == "VVFIN" | df$laenge < 4,]
```

```
df[!(df$pos %in% c("NN", "NE")),]
```


Begriffe



Dataframes

Funktionen

Default-Werte

Matrizen

Argumente

Prädikate

Listen

Rückgabewert

Log. Operatoren

Indizierung

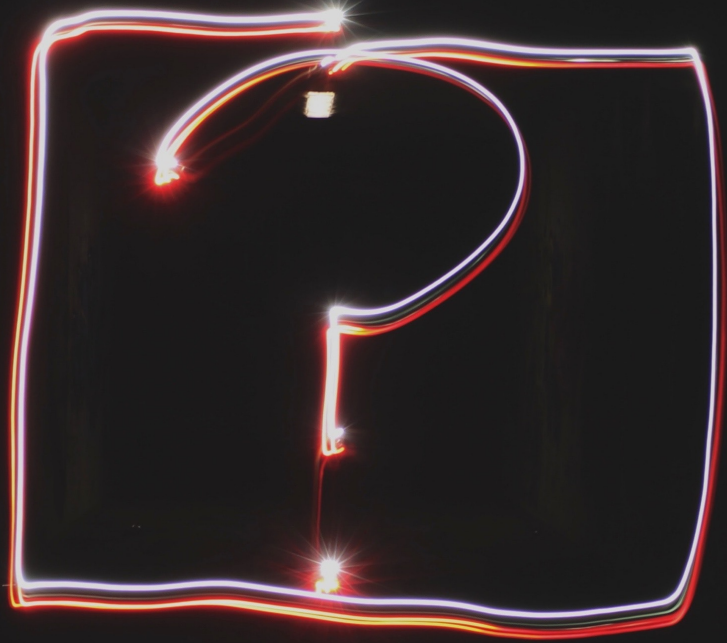
Nebeneffekt

& | !

[] [[]] [,] \$

Verschachteln

Fragen?





Übung

- Installieren Sie das Paket {openintro}.
- Schauen Sie sich die ersten 6 Zeilen des Dataframes `cia_factbook` an. *head()*
- Lassen Sie sich die Zeile für die Schweiz ausgeben.
- Lassen Sie sich die Zeilen für die Schweiz, Österreich und Deutschland ausgeben.
- Welches Land hat die größte Fläche im Datensatz?
- Berechnen Sie den Anteil von Internetbenutzer*innen für jedes Land, speichern Sie diesen Wert in der Spalte `www_percent`.



Daten einlesen

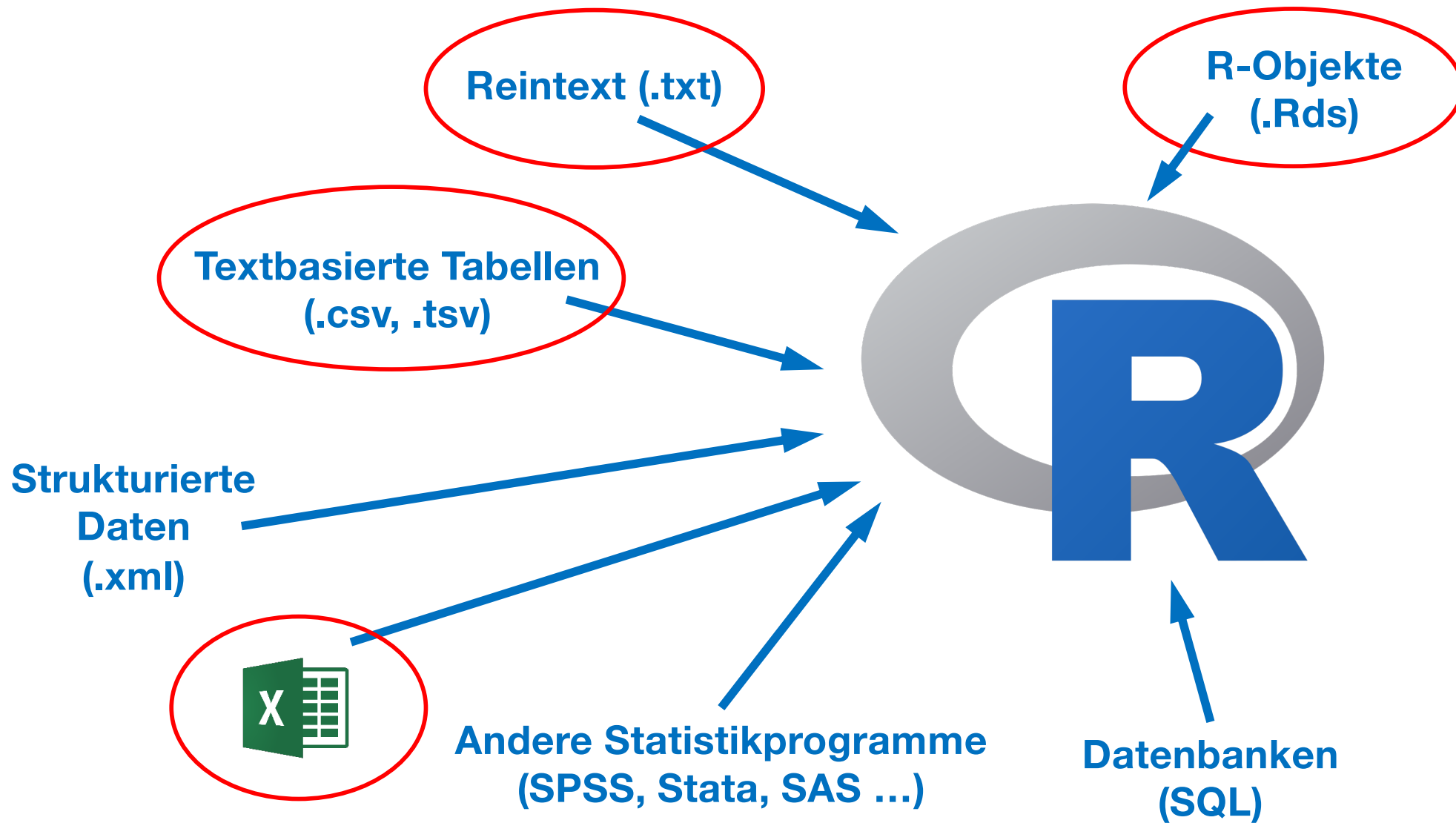
R aktualisieren

- Sollten Sie Probleme beim Installieren von Paketen haben, versuchen Sie bitte zunächst, R zu aktualisieren.
 - <https://cloud.r-project.org>
 - Nicht RStudio, sondern R!
- Die aktuelle R-Version (07.11.2022) ist **R 4.2.2** ("Innocent and Trusting")
- Ihre R-Version erscheint beim Startup. Sie können sie auch abrufen mit `R.Version()` – dort unter "version.string"



Bisher...

- ... haben wir mit `c()` oder `data.frame()` Datensätze selbst erstellt oder Datensätze aus Paketen verwendet.
- Meistens kommen Daten aber "von außen". Welche Daten können das sein?
 - "Rohe" Sprachdaten (z. B. Textdateien)
 - Ausgaben aus Korpusplattformen (z. B. Frequenzlisten)
 - Experimentaldaten (Reaktionszeiten, Antworten, ...)
 - Selbsterstellte Tabellen (z. B. händische Annotationen)
 - Webseiten ("Scraping")
 - ...



Reintext einlesen mit `scan()`

- Liest Dateien ein und speichert das Ergebnis in einem Vektor.
- Wir wollen **Text** einlesen, dann lautet die Syntax:
 - `scan(<Datei>, what = "character")`
- `scan()` kann auch Textdateien direkt aus dem Internet lesen:

```
udhr <- scan("http://research.ics.aalto.fi/cog/data/udhr/txt/ger.txt",  
            what = "character", fileEncoding = "UTF-8")
```


Exkurs: Häufigkeitstabellen mit `table()`

- `table()` erstellt aus einem Vektor eine Häufigkeitstabelle.

```
table(c("B", "A", "B"))
```

```
A B
```

```
1 2
```

```
udhr <- scan("http://research.ics.aalto.fi/cog/data/udhr/txt/ger.txt",  
            what = "character", fileEncoding = "UTF-8")
```

```
tab <- table(udhr)
```

```
tab
```

```
sort(tab, decreasing = T)[1:10]
```

Exkurs: Häufigkeitstabellen

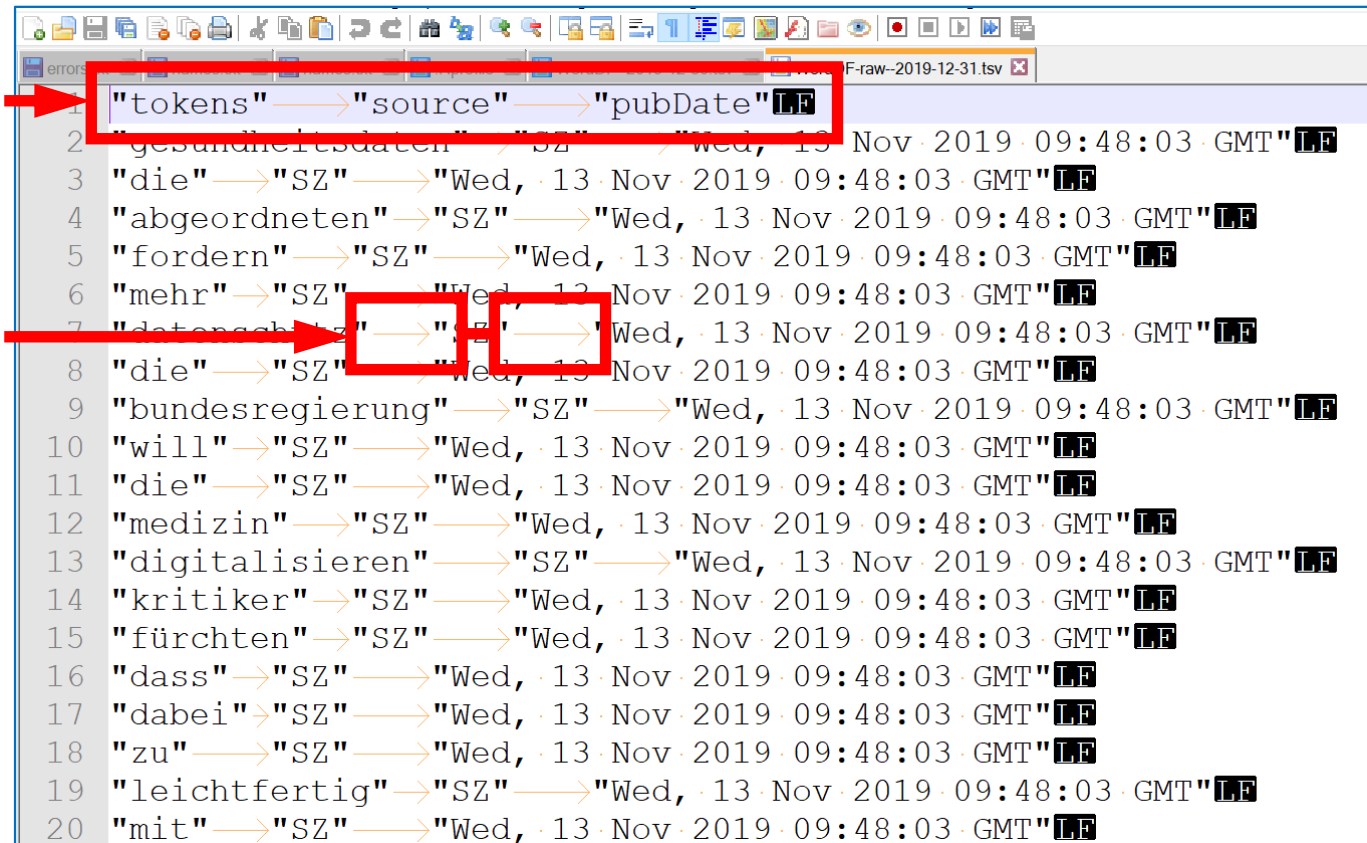
In korpuslinguistischen Untersuchungen werden Wörter oft in Kleinschreibung überführt.
In R: `tolower(<Vektor>)`

- Später werden wir noch lernen, wie wir die Häufigkeitstabelle für Wörter noch etwas "sauberer" hinbekommen.
 - z. B. keine Satzzeichen, die an Wörtern hängen ("ging," vs. "ging")
- Tables ≠ Dataframes
- Vektoren mit benannten Elementen
- Deshalb auch indizierbar über die Namen der Elemente
- Wie oft kommt das Wort "Generalversammlung" vor?
`tab["Generalversammlung"]`

Textbasierte tabellarische Daten

**Header/
Variablennamen**

**Separator/
Spaltentrenner**



The screenshot shows a text editor window titled 'F-raw-2019-12-31.tsv'. The content is a TSV file with 20 lines. The first line is a header: `"tokens" -> "source" -> "pubDate" [TF]`. The following lines are data rows, each starting with a word in quotes, followed by an arrow, a source code in quotes, another arrow, a date in quotes, and a frequency count in a box. Red annotations highlight the header line and the separator characters in the data rows.

```
1 "tokens" -> "source" -> "pubDate" [TF]
2 "gesundheitsdaten" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
3 "die" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
4 "abgeordneten" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
5 "fordern" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
6 "mehr" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
7 "datenschut" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
8 "die" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
9 "bundesregierung" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
10 "will" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
11 "die" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
12 "medizin" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
13 "digitalisieren" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
14 "kritiker" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
15 "fürchten" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
16 "dass" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
17 "dabei" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
18 "zu" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
19 "leichtfertig" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
20 "mit" -> "SZ" -> "Wed, 13 Nov 2019 09:48:03 GMT" [TF]
```



Textbasierte tabellarische Daten

- "comma-separated values" (CSV)
- Solche tabellarisch aufgebauten Daten lassen sich u.a. mit der Funktion `vroom()` aus dem Package `{vroom}` einlesen.

```
dat <- vroom(<Datei>)
```

- Die Funktion "rät" ziemlich gut selbst, wie das genaue Format der Datei ist (z. B. welcher Separator verwendet wird).
 - Falls es mal schief geht: `?vroom`
- `vroom()` gibt einen "Tibble" aus.
 - Tibbles sind (fast) wie Dataframes, die Ausgabe ist nur etwas komfortabler. Falls Sie lieber mit einem Dataframe arbeiten wollen:

```
dat <- as.data.frame(vroom(<Datei>))
```

Übung



- Laden Sie die Datei `dati`.`csv` herunter: <https://t1p.de/u2n63>
- Lesen Sie die Datei in die Variable `dat` ein. Schauen Sie sich die ersten 6 Zeilen an.
- Wie viele Zeilen hat die Datei insgesamt?
- Erstellen Sie eine sortierte Häufigkeitstabelle der Spalte `Verb`. Welches Verb kommt am häufigsten in `dati`.`csv` vor?



Excel-Dateien einlesen

- Paket: {readxl}, Funktion: `read_excel()` – Argumente:
 - `path`: (chr) Datei, die eingelesen werden soll (.xls/.xlsx)
 - `[sheet`: (chr/num) Arbeitsblatt, das eingelesen werden soll]
 - `[range`: (chr) Zellausschnitt, der gelesen werden soll]
 - Zellausschnitt wird angegeben wie in Excel (z. B. "B4:E20")
 - `[col_names`: (log) Erste Zeile Spaltenüberschriften/Variablennamen?]
 - weitere Argumente: `?read_excel`
- `read_excel()` gibt einen Tibble zurück, `as.data.frame()` um diesen in einen Dataframe umwandeln.

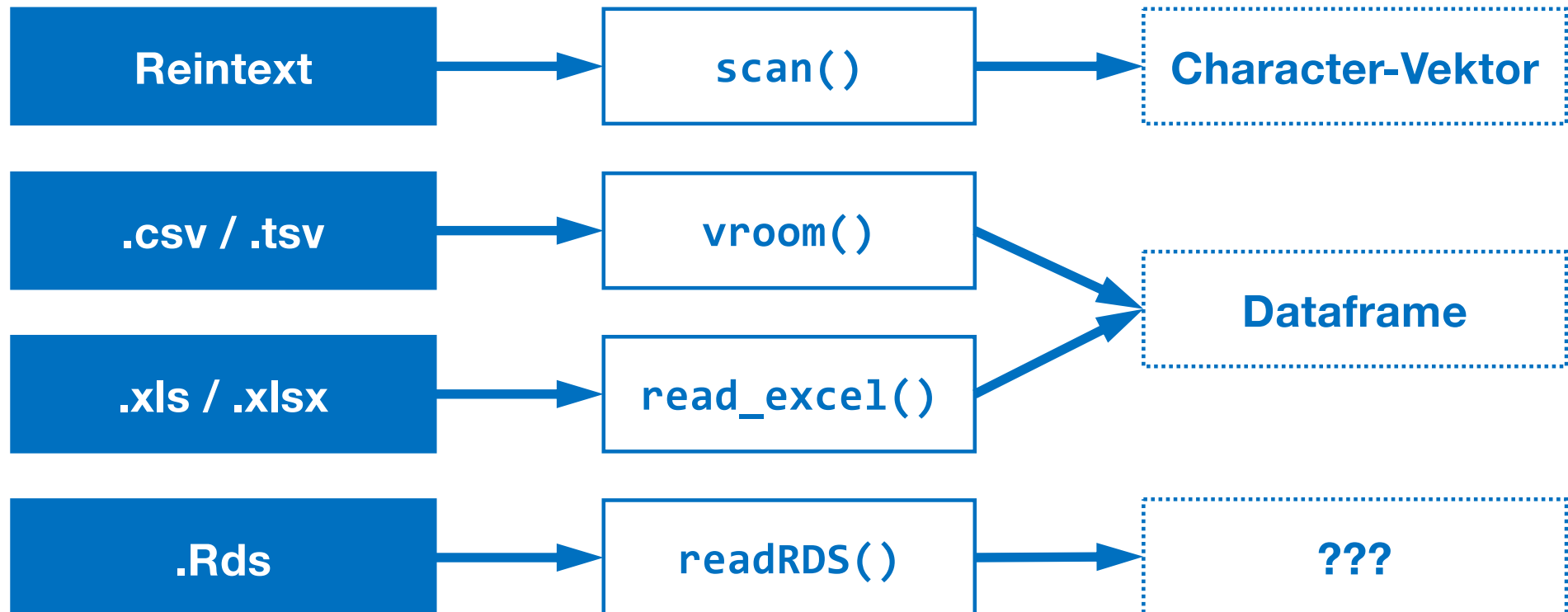
R-Objekte einlesen

- R-Objekte (Variablen) kann man in Dateien speichern. Diese Dateien haben die Endung `.Rds`.

```
dat <- readRDS(<Datei>)
```

- Diese Dateien können nur mit R eingelesen werden, eignen sich also nicht zum Austausch mit anderen Programmen.
- Dafür kann relativ wenig schiefgehen.
- Man kann alle Datentypen in Rds-Dateien abspeichern.
 - Nicht nur Dataframes, sondern auch Vektoren, Listen, Matrizen usw.

Einlesen: Zusammenfassung



A vintage white typewriter with a blue keyboard is shown from a high-angle perspective, resting on a white shag rug. The typewriter's carriage and typebars are visible through a cutout in the top cover. A blue semi-transparent overlay covers the bottom left portion of the image, containing the text "Daten ausgeben".

Daten ausgeben

Daten ausgeben: Funktionen

Reintext

`cat()`

`.csv / .tsv`

`vroom::vroom_write()`

`.xls / .xlsx`

`WriteXLS::WriteXLS()`

`.Rds`

`saveRDS()`

Reinen Text ausgeben mit `cat()`

- Es kommt nicht sonderlich oft vor, dass Sie aus R laufenden Text ausgeben müssen.
- Wenn, dann am einfachsten mit
- Ausgeben der Buchstaben a bis z in einzelnen Zeilen der Datei `buchstaben.txt`:

```
cat(<Vektor>, <Datei>, sep = <Separator>)
```

```
cat(letters, file = "buchstaben.txt", sep = "\n")
```

CSV-Dateien schreiben mit `vroom_write()`

- `vroom_write()` schreibt Dataframes in CSV-Dateien.
`vroom_write(<Dataframe>, <Dateiname>, delim = <Separator>)`
 - Weitere Argumente: `?vroom_write`

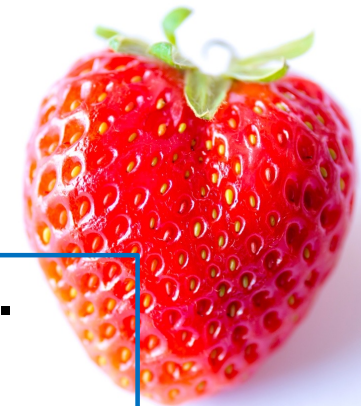
Excel-Dateien ausgeben mit `writeXLS()`

- Package: `{WriteXLS}`

```
writeXLS(<Dataframe>, <Dateiname>,  
        AdjWidth = T, BoldHeaderRow = T)
```

- Weitere Argumente: `?writeXLS`

`writeXLS()` benötigt eine funktionierende **Perl-Installation**.
Bei macOS wird diese mitgeliefert. Für Windows kann man
bspw. Strawberry Perl herunterladen: <https://strawberryperl.com>



R-Objekte schreiben mit `saveRDS()`

- Wenn Sie ein beliebiges R-Objekt in einer `.Rds`-Datei speichern möchten, benutzen Sie `saveRDS(<Variable>, <Datei>)`
- Diese Dateien können nur von R wieder eingelesen werden.
 - Erinnerung: Einlesen mit `readRDS()`



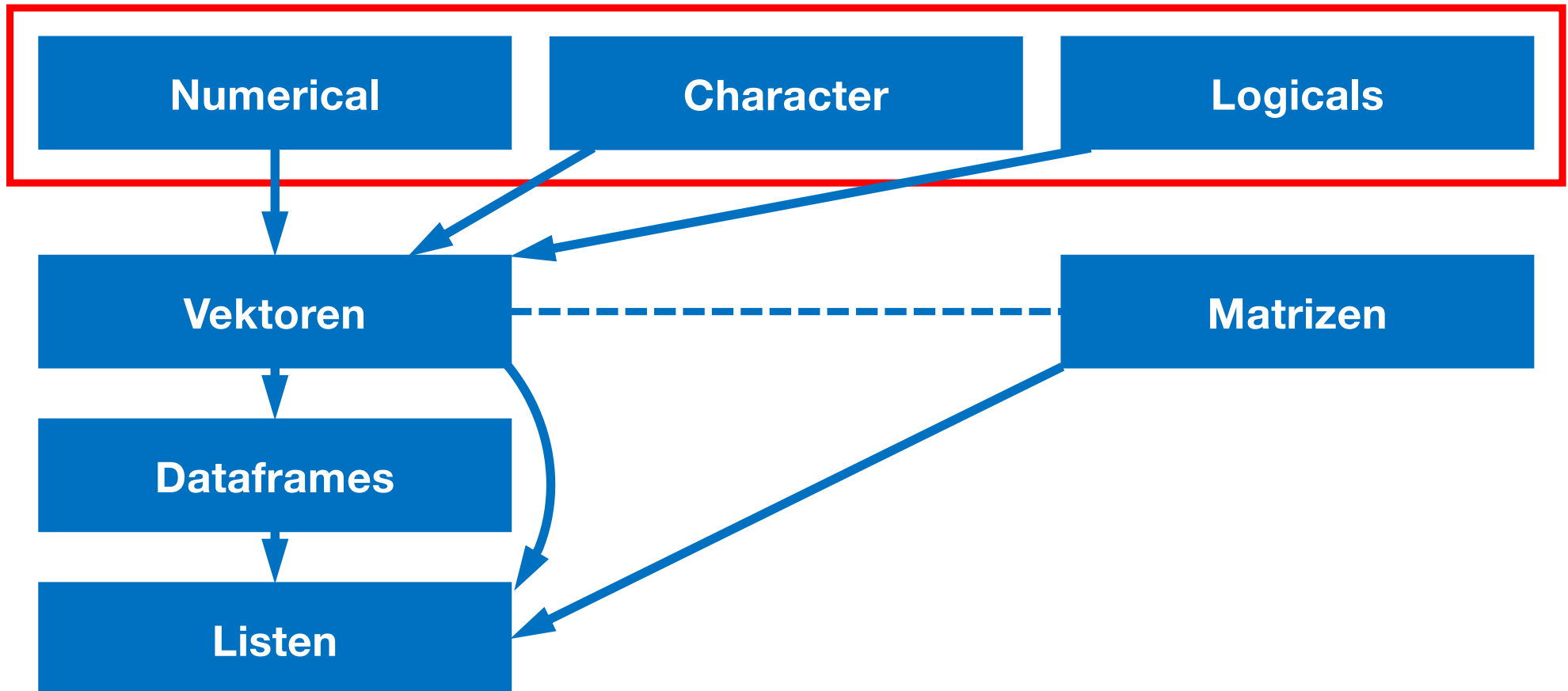
R-Einführung: Zusammenfassung

Oberfläche

- Befehle sammeln im Script-Editor, **jetzt** mit # kommentieren und sich später freuen!
- Pakete installieren mit `install.packages()`
- Hilfe abrufen mit `?<Funktion>` oder Hilfe durchsuchen mit `??<Suchterm>`

Datentypen

Elementare Datentypen



Vektoren ...

- ... bestehen immer aus *einem* elementaren Datentyp.
- ... werden automatisch **recyclet**.
- ... können **Faktor**vektoren sein, um nominal-/ordinalskalierte Daten zu repräsentieren.

Indizierung

- Zugreifen auf/Selektieren von Elemente(n) in komplexen Datentypen.

[]

[,]

\$

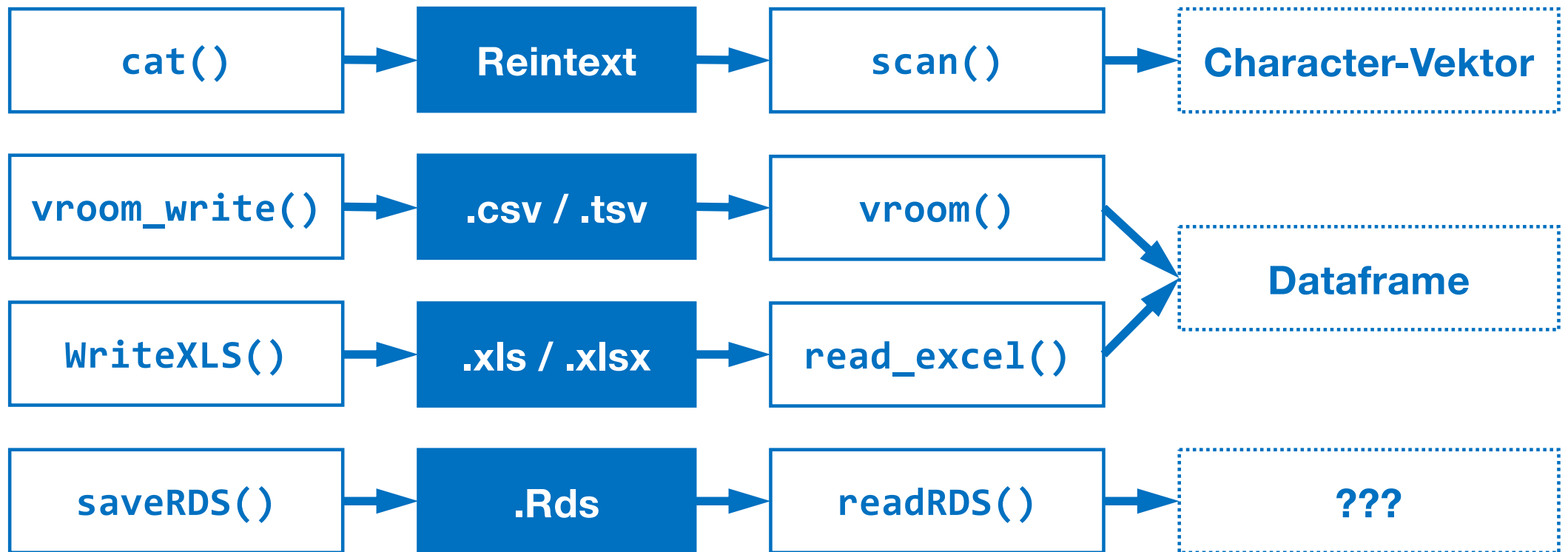
[[]]

Wir können über Zahlen, Namen und Wahrheitswerte indizieren.

Funktionen

- Funktionen haben **Argumente**.
- Funktionen können **verschachtelt** werden.
- **Prädikate** sind spezielle Funktionen, die immer einen Wahrheitswert zurückgeben.

Einlesen & Ausgeben





”

If you're not failing 90% of the time, then you're probably not working on sufficiently challenging problems.

- Alan Kay



Deskriptive Statistik

Deskriptive Statistik

- Ziel: **Beschreiben** einer Stichprobe
- Wir tun das für uns und andere!
 - Eindruck von den gesammelten Daten **bekommen**
 - Fehler entdecken
 - Muster entdecken
 - Überblick über die Daten **geben**
 - Offene Darstellung der eigenen Daten im wissenschaftlichen Prozess
- Bei der deskriptiven Statistik bleiben wir eng an der **Stichprobe**.
 - Keine Verallgemeinerung auf die Grundgesamtheit ("Population")

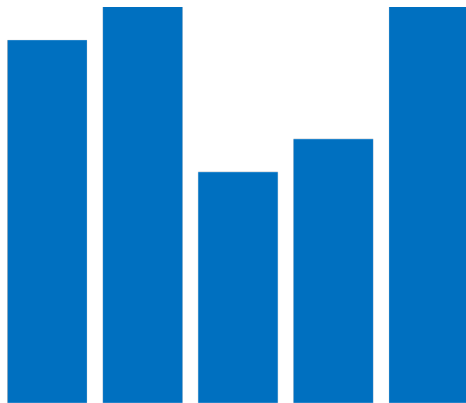
Grundgesamtheit

Stichprobe 

Deskriptive Statistik

- **Visualisierungen** sind ein Mittel der deskriptiven Statistik.
- Visualisierungen brauchen aber numerische Werte.
- Kennzahlen zu Charakteristika von **Verteilungen**.
 - Nominal-/Ordinalskalierte ("**diskrete**") Variablen: Häufigkeitsverteilung
 - Metrisch skalierte ("**kontinuierliche**") Variablen: Dichteverteilung

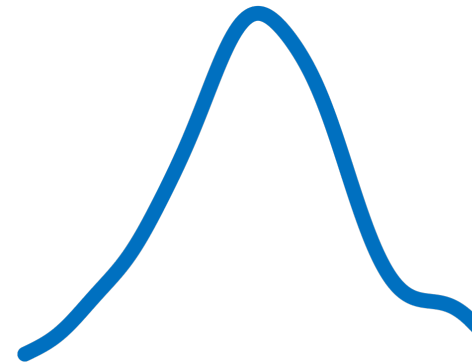
Verteilungen



Häufigkeitsverteilung

Gibt für jeden Wert einer Variable an, wie oft dieser Wert vorkommt.

Beispiel: Häufigkeit von Augenzahlen bei 100 Würfeln mit einem Würfel.



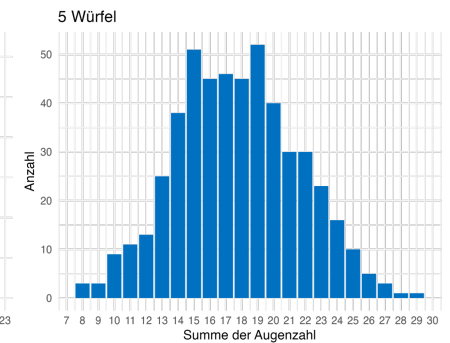
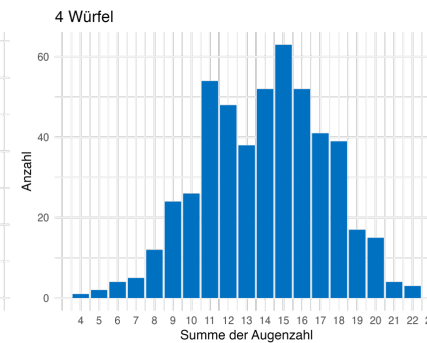
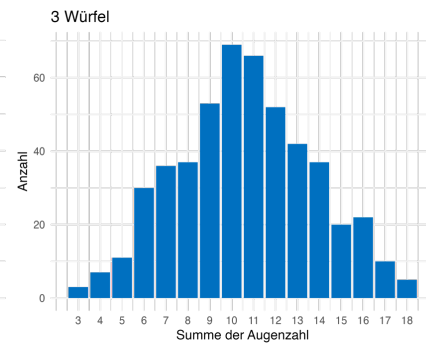
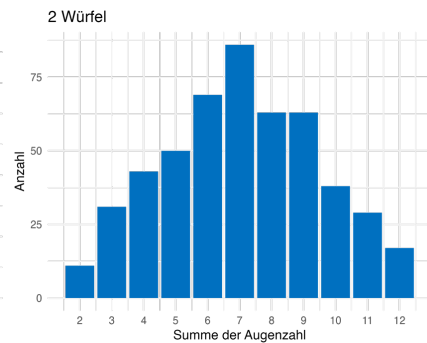
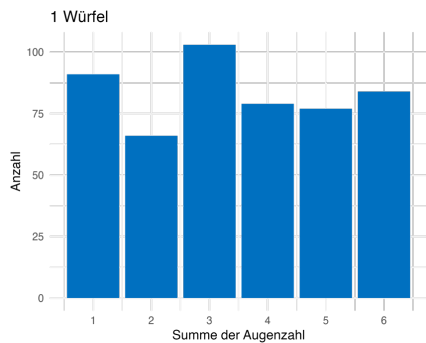
Dichteverteilung

Gibt an, wie viele Werte in einem bestimmten *Bereich* vorkommen.

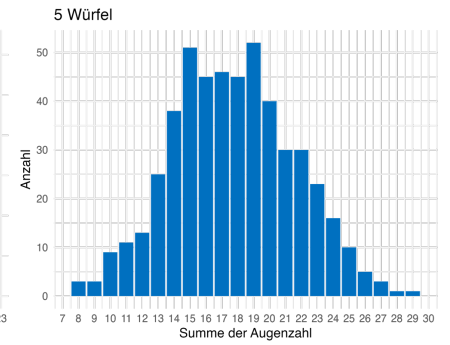
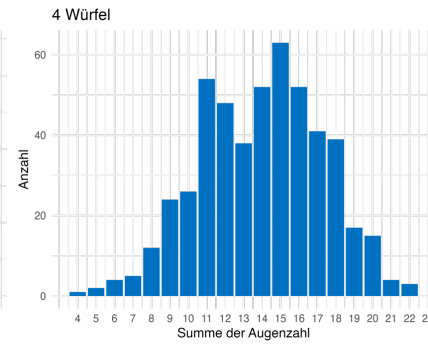
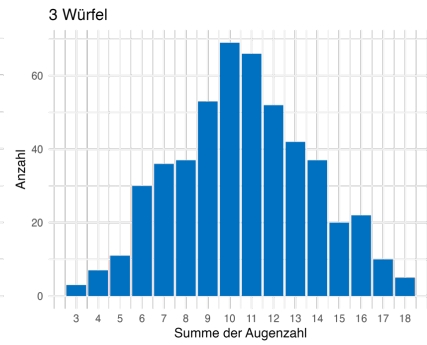
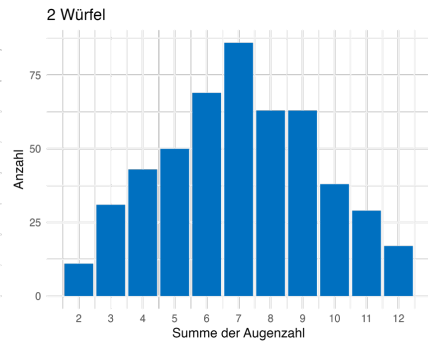
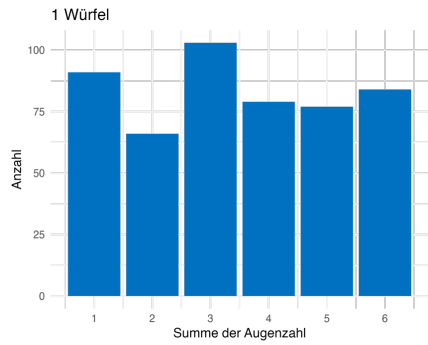
Beispiel: Anzahl von Reaktionszeiten zwischen 200 und 300 msec.

Nebenbemerkung

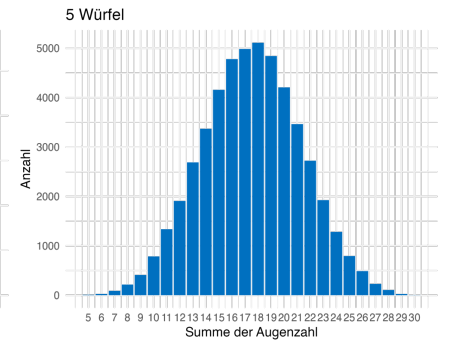
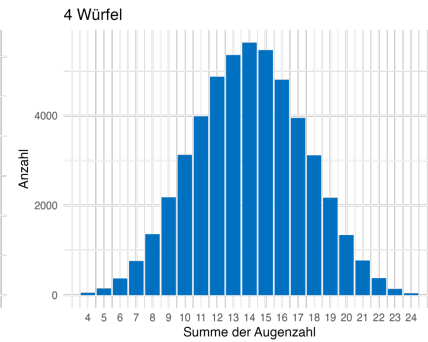
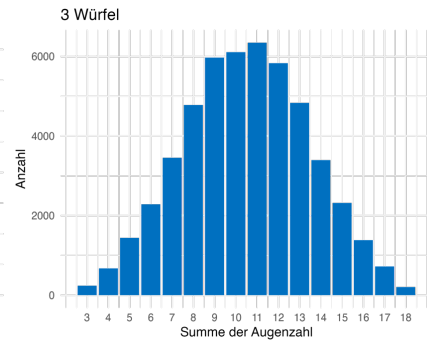
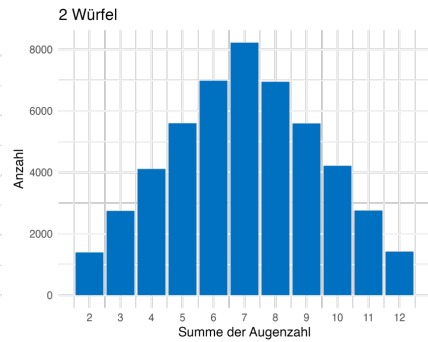
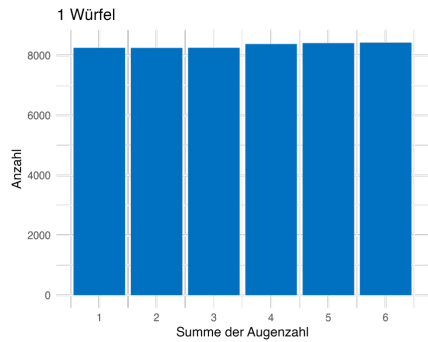
- So groß sind die Unterschiede gar nicht.
 - Eine Häufigkeitsverteilung kann in eine Dichteverteilung übergehen.
- Versuch: Wir würfeln 500x mit einer steigenden Anzahl von Würfeln (1 bis 5) und notieren uns für jeden Wurf die Summe der Augenzahl.

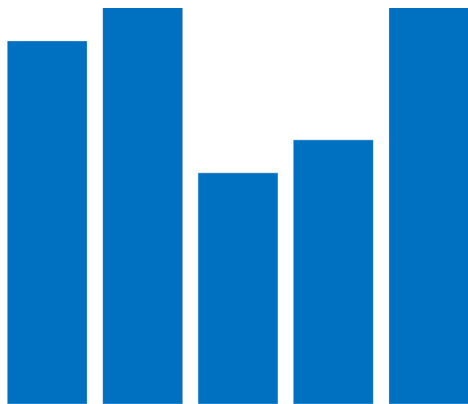


500 Mal

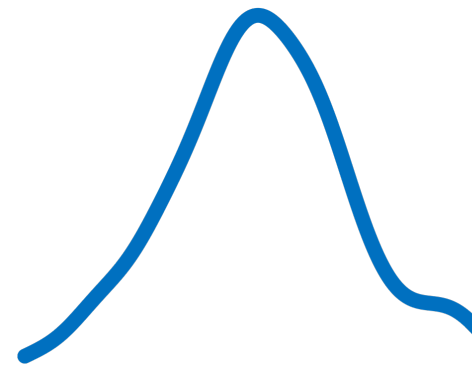


5000 Mal



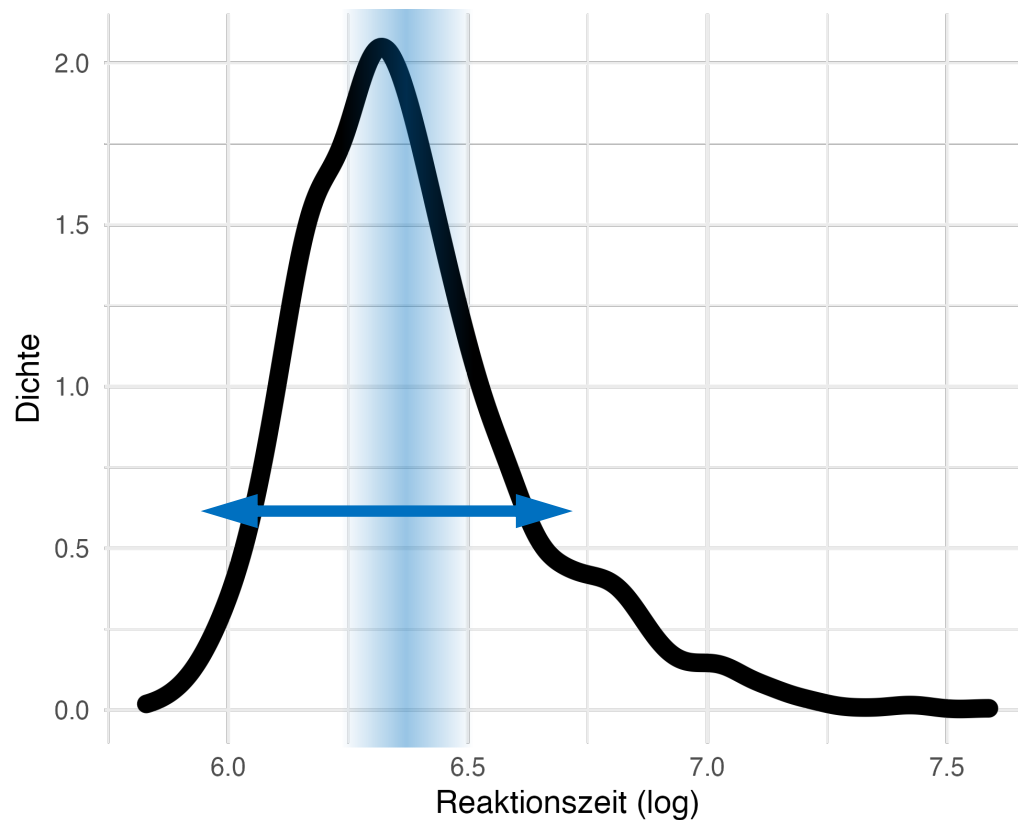


Häufigkeitsverteilung



Dichteverteilung

Kontinuierliche Variablen



- Wie können wir die Verteilung beschreiben?
- Wir wollen wissen:
 - Wo ist die "**Mitte**" der Verteilung?
 - Wie stark **streut** die Variable?

Masse der zentralen Tendenz

Streuungs-/Dispersionsmasse

Maße der zentralen Tendenz

Modus / Modalwert

Der am häufigsten vorkommende Wert

Median

Der Wert, der die Datenreihe in zwei Hälften teilt.

Arithmetisches Mittel

Der Mittelwert: Summe aller Werte geteilt durch die Anzahl

Aufsummieren aller Werte von 1 bis n

"x quer" \bar{x}

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

n : Anzahl Werte

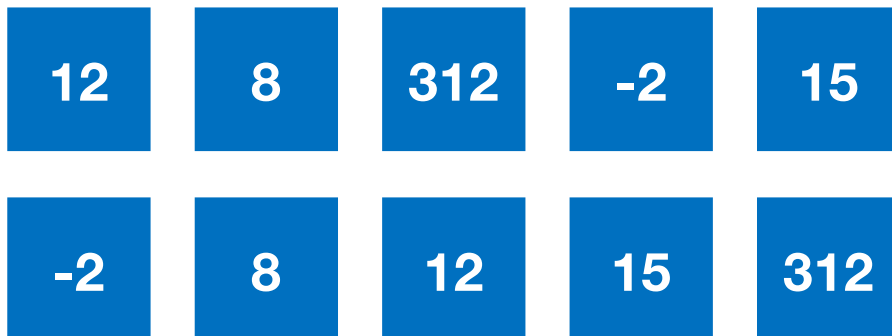
Modus / Modalwert

- Bei diskreten Variablen (→ Häufigkeitsverteilungen) kann nur der Modus sinnvoll berechnet werden.
 - Welcher Wert kommt am häufigsten vor?
- In R: `which.max(table(<Vektor>))`
- Bei kontinuierlichen Variablen ist der Modalwert hingegen meist sinnlos.
 - Hier wäre es angebrachter, ein bestimmtes **Intervall** anzugeben, in dem die meisten Werte vorkommen.

Median

Bei einer geraden Anzahl an Werten wird die Mitte der mittleren beiden Werte als Median angenommen.

- Der Median liegt in der Mitte aller Werte.
- Oder: Der Median ist der Wert, der alle vorkommenden Werte in zwei Hälften teilt.
 - Über und unter dem Median befinden sich 50% aller Werte.
- Kann ab der Ordinalskala berechnet werden.



```
median(c(12, 8, 312, -2, 15))
```

Mittelwert

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

- Summe aller Werte geteilt durch die Anzahl.
- Kann nur für metrisch skalierte Variablen berechnet werden.
 - Beispiel: Was sollte der Mittelwert aus den Erstsprachen Französisch, Englisch und Italienisch sein?

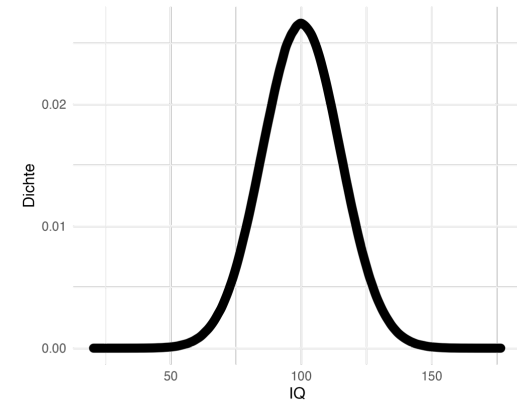
$$12 + 8 + 312 + -2 + 15 = 345$$

$$345 / 5 = 69$$

`mean(c(12, 8, 312, -2, 15))`

Median und Mittelwert

```
median(c(12, 8, 312, -2, 15)) → 12  
mean(c(12, 8, 312, -2, 15)) → 69
```



- Beides sind Maße der zentralen Tendenz, können aber zu unterschiedlichen Ergebnissen führen.
- Der Mittelwert ist deutlich anfälliger gegenüber **Ausreisserwerten!**
 - Oben: 312
- Identisch sind Median und Mittelwert nur bei exakt symmetrischen Verteilungen (z. B. Normalverteilung).

Beachten Sie aber auch das Argument `trim`. Siehe `?mean`

Fehlende Werte

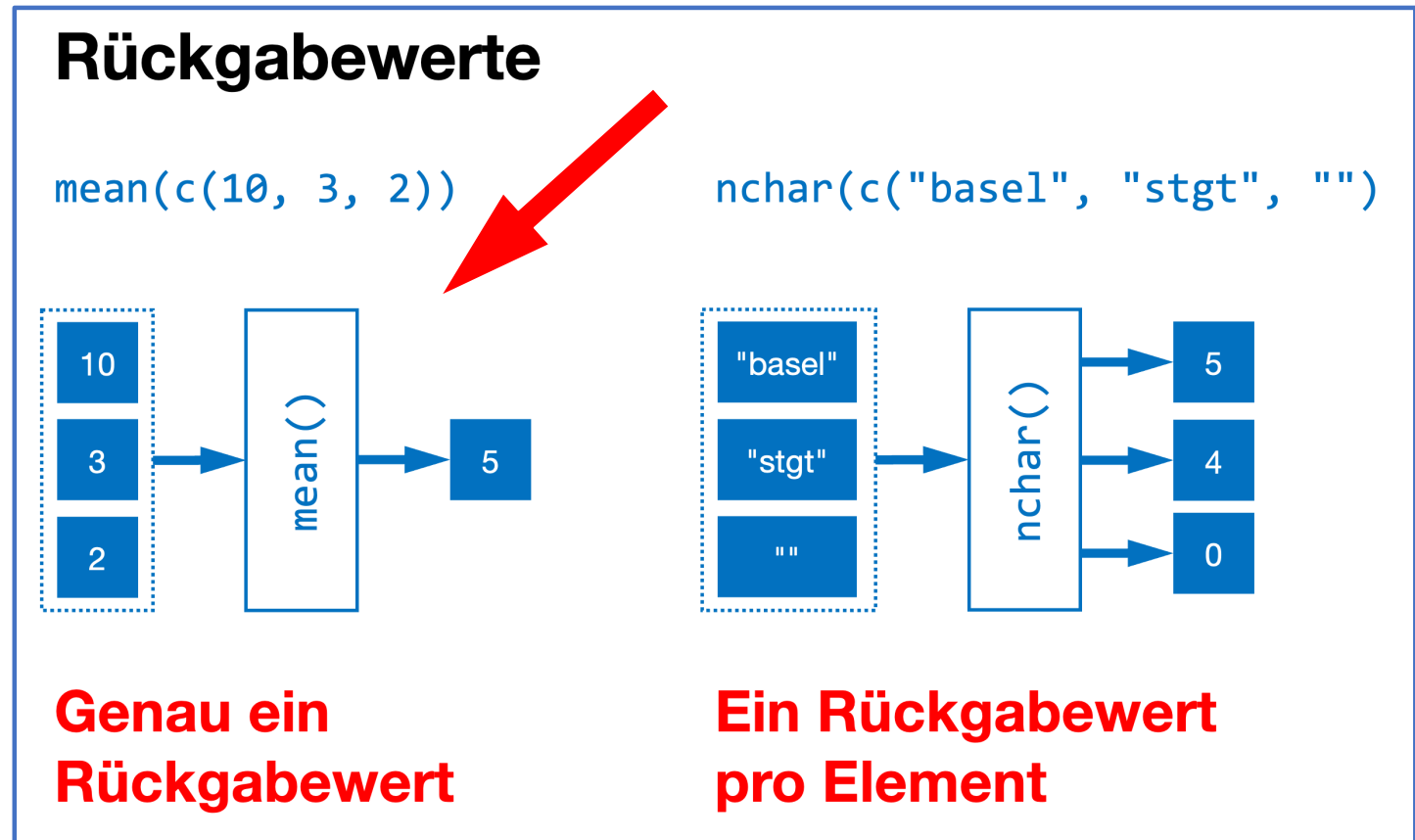
- Enthält eine Datenreihe einen fehlenden Wert, gibt R für `median()` und `mean()` ebenfalls NA zurück.
- Argument `na.rm = T` entfernt erst die fehlenden Werte und berechnet dann den Median/Mittelwert (`na.rm = NA remove`).
- Das gilt auch für die Funktionen für Standardabweichung und Varianz (s. später)!

Berechnung nach Gruppen

- Mit `tapply()` kann man Berechnungen gruppieren.
 - `tapply(<Vektor>, <Gruppierungsvektor>, <Funktion>)`
- Zum Beispiel: Mittelwert von Spalte `WLen` gruppiert nach `POS`
 - `tapply(data$WLen, data$POS, mean)`
- Nach `<Funktion>` können Argumente folgen, die `<Funktion>` übernimmt (bspw. `na.rm = T`).

Berechnung nach Gruppen

`tapply()` wird typischerweise nur mit Funktionen angewendet, die genau **einen** Rückgabewert haben!





Übung

- Laden Sie die Datei `Exp.csv` herunter und laden Sie sie in R:
<https://t1p.de/h0hct> / ADAM
- Berechnen Sie Mittelwert und Median für die Spalte `RT`.
 - `NA` ist keine gültige Lösung.
- Identifizieren Sie den Grund dafür, dass Median und Mittelwert so weit auseinanderliegen.
- Erstellen Sie ein einfaches Diagramm der Daten in der Spalte `RT`: `plot(<Spalte>)`
- Berechnen Sie den Mittelwert für jede Bedingung (Spalte `Bedingung`)

Streuungsmaße

Interquartilabstand

Der Abstand, der die mittleren 50% aller Werte umfasst

Varianz

Die aufsummierten quadrierten Abweichungen vom Mittelwert, geteilt durch die Anzahl der Werte

Standardabweichung

Die Quadratwurzel aus der Varianz

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

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

Interquartilabstand (IQR)

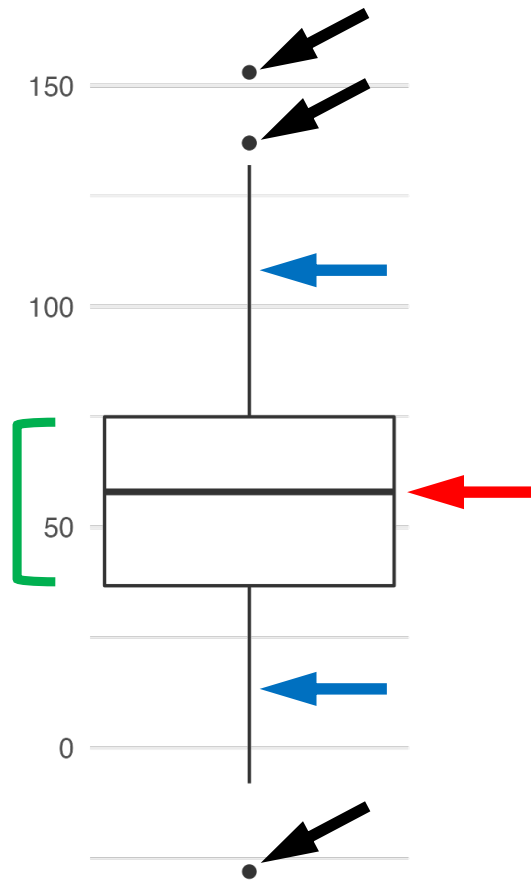
- Der **Median** befindet sich in der Mitte aller beobachteten Werte. Unter und über dem Median sind somit 50% aller Werte.
- Der **Interquartilabstand** umfasst die **mittleren 50%** aller Werte.
- Je weiter die Werte streuen, desto grösser muss dieser Abstand sein, um die mittleren 50% zu erfassen.



Interquartilabstand (IQR) und Spannweite

- In R: `IQR(<Vektor>)`
- IQR = interquartile range
- Man kann zusätzlich die **komplette** Spannweite / Range der Daten angeben, also Maximalwert minus Minimalwert.
 - `range()` gibt einen Vektor mit zwei Werten aus: Minimal- und Maximalwert.
 - Mit `diff(range(<Vektor>))` bekommen Sie den **Abstand** zwischen Minimal- und Maximalwert.

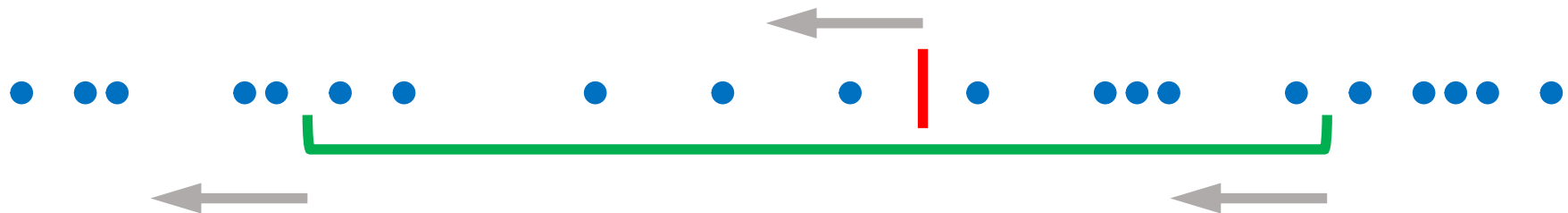
Boxplot



- **Median** und **Interquartilabstand** sind Bestandteil des Boxplots.
 - Median: Horizontale Linie in der Box
 - IQR: Höhe der Box
- "**Hinges**": $1,5 \cdot \text{IQR}$, bis zum letzten beobachteten Wert in diesem Abstand
- Outlier: Alle Punkte, die nicht innerhalb $\text{IQR} \pm 1,5 \cdot \text{IQR}$ liegen.
- Boxplot zeigt viele Informationen auf einmal und bleibt dabei einigermaßen übersichtlich.
- In R: `boxplot(x)`

Quartile und Perzentile

- Nochmal: **Median** teilt die Datenpunkte in zwei Hälften. Unter dem Median sind also 50% aller Daten.
- Unter der **unteren Grenze** des IQR sind 25% aller Daten.
- Unter der **oberen Grenze** sind 75% aller Daten.
- Man nennt die Grenzen deshalb auch erstes und drittes **Quartil**.
- Haben Sie eine Idee, wie man den Median noch nennt?



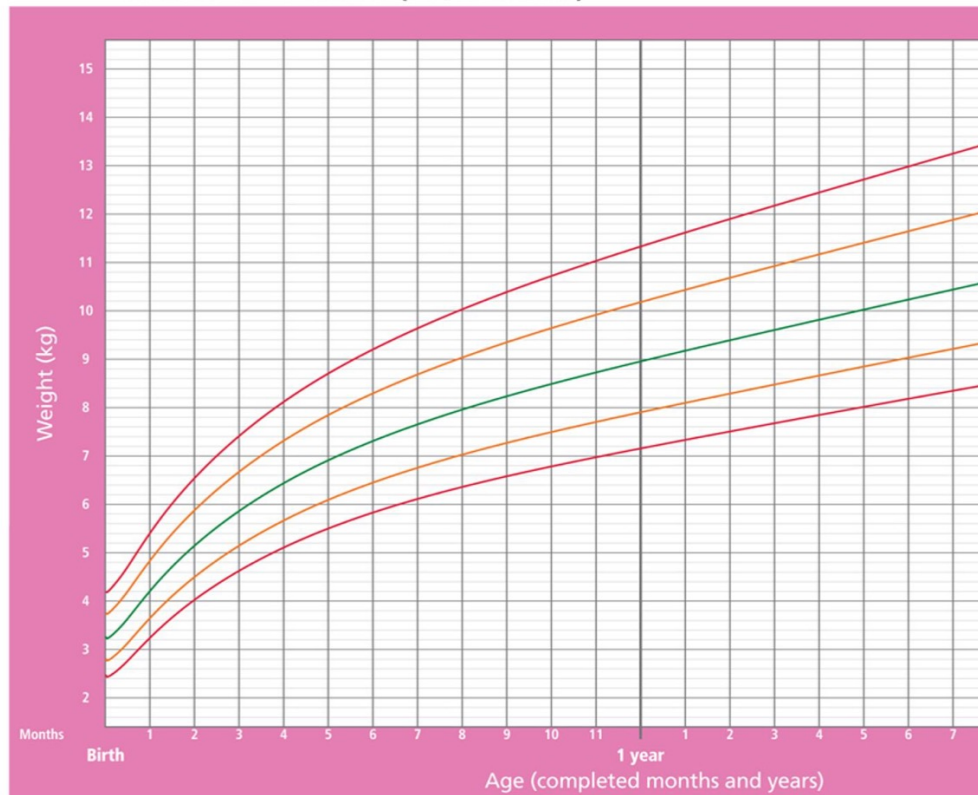
Quartile und Perzentile

- Verallgemeinerbar auf alle "Abschnitte" außer Quartilen.
 - Z. B.: Punkt, unter dem 42% aller Daten liegen.
- Name: **Perzentile** (auch: **Quantile**)
- Definition: Das x -te Perzentil ist jener Punkt, unter dem x Prozent aller Daten liegen.

Perzentile in der Welt

Gewicht nach Alter: MÄDCHEN

Geburt bis 2 Jahre (Perzentilen)



Statistik mit R für die Linguistik • Uni Basel • Sascha Wolfer

CollegeBoard

SAT

SAT Score Report

Jane Doe
123 Main St.
Parma, OH 44130

Your Total Score

1170 | 400–1600

77th ←

Nationally Representative
Sample Percentile

71st ←

SAT User Percentile

Essay Scores

4 | 2 to 8
Reading

4 | 2 to 8
Analysis

5 | 2 to 8
Writing

Varianz

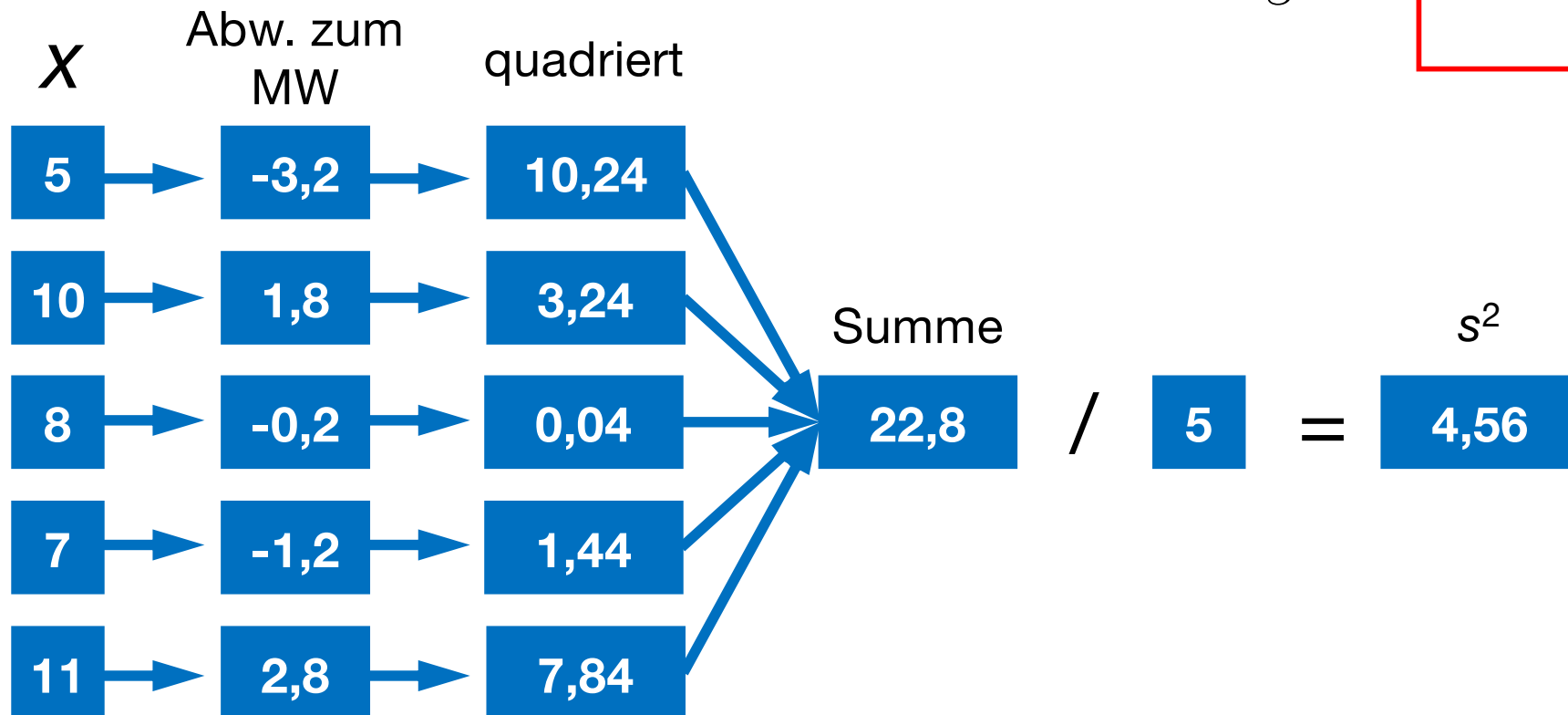
- Idee: Daten streuen mehr, wenn die einzelnen Werte weiter von ihrem Mittelwert entfernt sind.
- Wir können also ...
 1. die Abweichungen von jedem Wert zum Mittelwert berechnen,
 2. diese aufsummieren
 3. und dann durch die Anzahl der Werte teilen.
- Ergebnis: Durchschnittliche Abweichung zum Mittelwert
- Weil wir große Abweichungen mehr bestrafen möchten, quadrieren wir zusätzlich die Abweichungen in 1.

Varianz: Beispielrechnung

8,2

MW

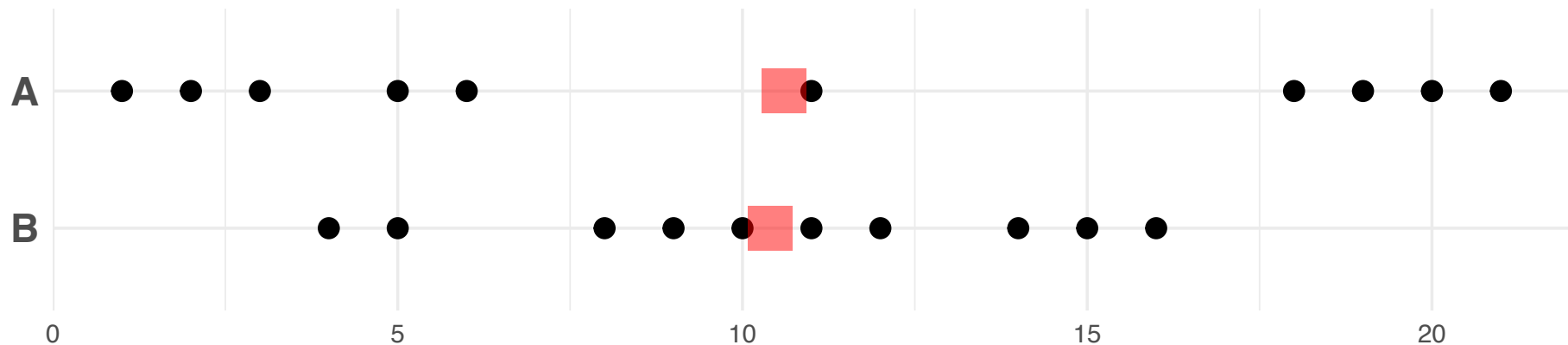
$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$



Varianz

Varianz für Datenreihe A: 66,5
Varianz für Datenreihe B: 16,3

Fällt Ihnen etwas auf, wenn Sie die Größenordnung der Varianz mit den Messwerten vergleichen (insb. in Datenreihe A)?



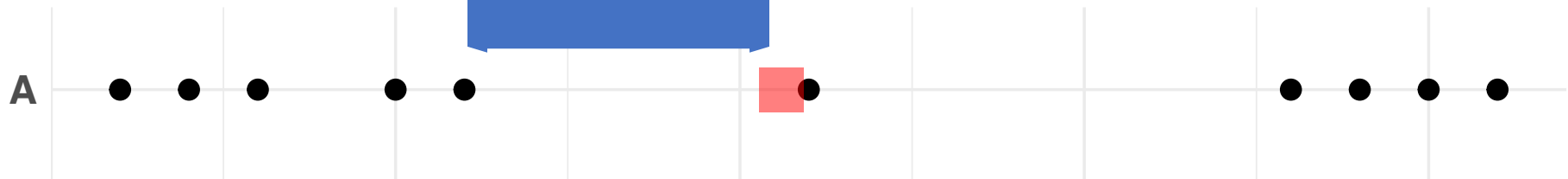
Varianz und Standardabweichung

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

- Problem: Durch die Quadrierung der Differenzen zum Mittelwert hat die Varianz eine andere Skalierung als die Datenpunkte.
- Lösung: Ziehen der Quadratwurzel → Standardabweichung s

Wichtig: Anderes Ergebnis als wenn wir einfach die Differenzen zum Mittelwert nicht quadrieren würden!

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



Dispersionsmaße in R

- Interquartilabstand: `IQR(x)`, Range: `range(x)`
- Perzentile: `quantile(x, <Wert>)` – Prozentwert zwischen 0 und 1
- `summary(x)` gibt 6 Werte aus:
 - Minimum & Maximum
 - 1. Quartil, 2. Quartil (Median), 3. Quartil
 - Mittelwert
- Varianz: `var(x)`
- Standardabweichung: `sd(x)` – *standard deviation*
- Achtung: `var(x)` und `sd(x)` werden mit $n - 1$ im Nenner berechnet (Schätzung von Populationsparametern).

Zusammenfassung

- In der deskriptiven Statistik beschreiben wir die **Stichprobe**, die wir gesammelt haben.
- Wir können unterscheiden zwischen Häufigkeitsverteilungen (**diskrete** Variablen) und Dichteverteilungen (**kontinuierliche** Variablen).
- Bei kontinuierlichen Variablen wollen wir wissen:
 - Wo ist die Mitte der Verteilung? → **Maße der zentralen Tendenz**
 - Wie stark streut die Variable? → **Streuungs-/Dispersionsmaße**

Zusammenfassung

- Maße der zentralen Tendenz:
 - **Modus / Modalwert**
 - **Median**
 - **Arithmetischer Mittelwert**
- Modalwert ist nur für diskrete Variablen sinnvoll.
- Mittelwert ist anfälliger für **Ausreisserwerte** als der Median.

Zusammenfassung

- Streuungsmaße:
 - **Interquartilabstand (IQR) / Spannweite**
 - **Varianz**
 - **Standardabweichung**
- **Quartile** teilen Daten in Viertel.
- Unter dem x ten **Perzentil** liegen x Prozent aller Datenpunkte.
 - Auch "**Quantil**" genannt.

Begriffe



Deskriptive Statistik

Dispersionsmaße

Spannweite / Range

Stichprobe

Modus / Modalwert

Quartil

Population

Median

Perzentil

Verteilungen

Mittelwert

Boxplot

**Maße der
zentralen Tendenz**

**Interquartilabstand
(IQR)**

Varianz

Standardabweich.



Übung

- Laden Sie die Datei `Exp.csv` herunter und laden Sie sie in R:
<https://t1p.de/h0hct> / ADAM
- Berechnen Sie IQR, Spannweite, Varianz und Standardabweichung für die Spalte `RT`.
- Lassen Sie sich eine Summary der Spalte geben.
- Erstellen Sie einen Boxplot der Spalte.
 - Versuchen Sie einen Boxplot unter Ausschluss des Ausreissers zu plotten.
- Bonusaufgabe: Berechnen Sie die Standardabweichung mit n statt mit $n - 1$ im Nenner (Lösung: 1257.265).



Korrelation

Korrelationen sind überall

- Speed-accuracy trade-off (Salthouse, 1979): Je schneller man etwas tut, desto ungenauer tut man es.
- Je mehr PS ein Auto hat, desto größer die Beschleunigung.
- Je häufiger ein Wort ist, desto kürzer ist es (Zipf, 1949).
- Je mehr Feuerwehrleute vor Ort sind, desto höher ist die Schadenssumme.
- Je mehr Störche in einem Gebiet leben, desto mehr Kinder werden dort geboren.

Korrelation

- Der **Korrelationskoeffizient** beschreibt den Zusammenhang zweier Variablen.
 - Pearson oder Spearman
- Der Korrelationskoeffizient kann Werte zwischen -1 und +1 annehmen.
 - +1: positiver Zusammenhang (je mehr x , desto mehr y)
 - -1: negativer Zusammenhang (je mehr x , desto weniger y)
 - 0: kein Zusammenhang

Korrelation

Akoglu, H. (2018). User's guide to correlation coefficients. Turkish Journal of Emergency Medicine, 18(3), 91–93. <https://doi.org/10.1016/j.tjem.2018.08.001>

Table 1

Interpretation of the Pearson's and Spearman's correlation coefficients.

Correlation Coefficient		Dancey & Reidy (Psychology)	Quinnipiac University (Politics)	Chan YH (Medicine)
+1	−1	Perfect	Perfect	Perfect
+0.9	−0.9	Strong	Very Strong	Very Strong
+0.8	−0.8	Strong	Very Strong	Very Strong
+0.7	−0.7	Strong	Very Strong	Moderate
+0.6	−0.6	Moderate	Strong	Moderate
+0.5	−0.5	Moderate	Strong	Fair
+0.4	−0.4	Moderate	Strong	Fair
+0.3	−0.3	Weak	Moderate	Fair
+0.2	−0.2	Weak	Weak	Poor
+0.1	−0.1	Weak	Negligible	Poor
0	0	Zero	None	None

Was als schwache, moderate und starke Korrelation gilt, variiert zwischen (und auch in) Disziplinen.

Taxifahren in London

- Datenlage: Anzahl der Dienstjahre von Personen, die in London Taxi fahren, korreliert positiv mit der Größe eines Teils des Hippocampus (zuständig u.a. für Orientierung).
- Untersucht wurden Personen mit unterschiedlicher Erfahrung.
- Folgerung: Taxifahren führt zur Vergrößerung des Hippocampus.



Interpretation von Korrelationen

- Nehmen wir an, zwei Variablen x und y korrelieren positiv (je mehr x , desto mehr y).
- Interpretation 1: Einseitige Steuerung x bewirkt y (oder andersherum)



Interpretation von Korrelationen

- Nehmen wir an, zwei Variablen x und y korrelieren positiv (je mehr x , desto mehr y).
- Interpretation 2: Gegenseitige Steuerung
 x wirkt auf y , y wirkt auf x zurück



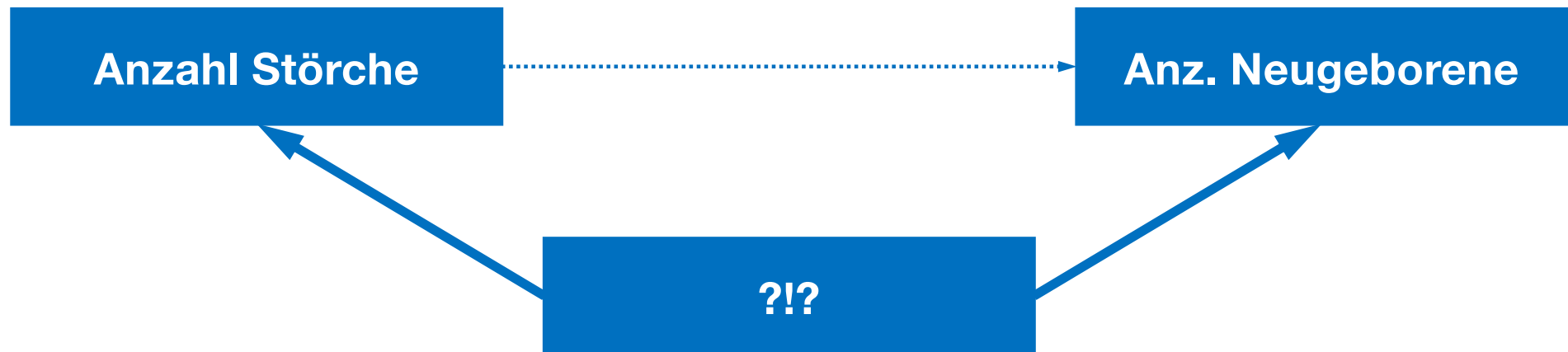
Interpretation von Korrelationen

- Nehmen wir an, zwei Variablen x und y korrelieren positiv (je mehr x , desto mehr y).
- Interpretation 3: Drittseitige Steuerung
 x und y hängen von einer dritten Variable z ab



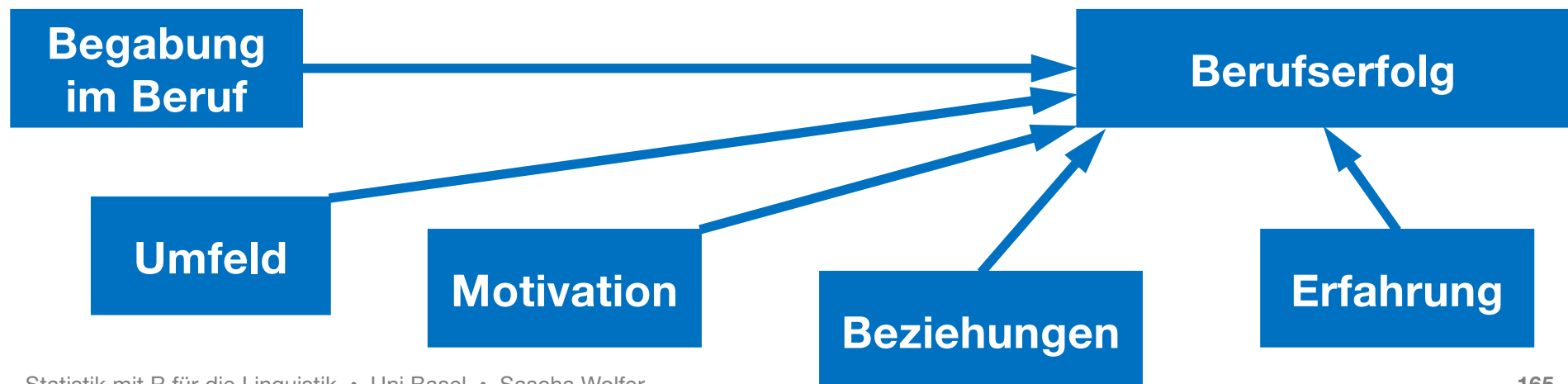
Interpretation von Korrelationen

- Nehmen wir an, zwei Variablen x und y korrelieren positiv (je mehr x , desto mehr y).
- Interpretation 3: Drittseitige Steuerung
 x und y hängen von einer dritten Variable z ab



Interpretation von Korrelationen

- Nehmen wir an, zwei Variablen x und y korrelieren positiv (je mehr x , desto mehr y).
- Interpretation 4: Komplexe Steuerung
Das Bedingungsgefüge ($a, b, c \dots x$) bewirkt y .

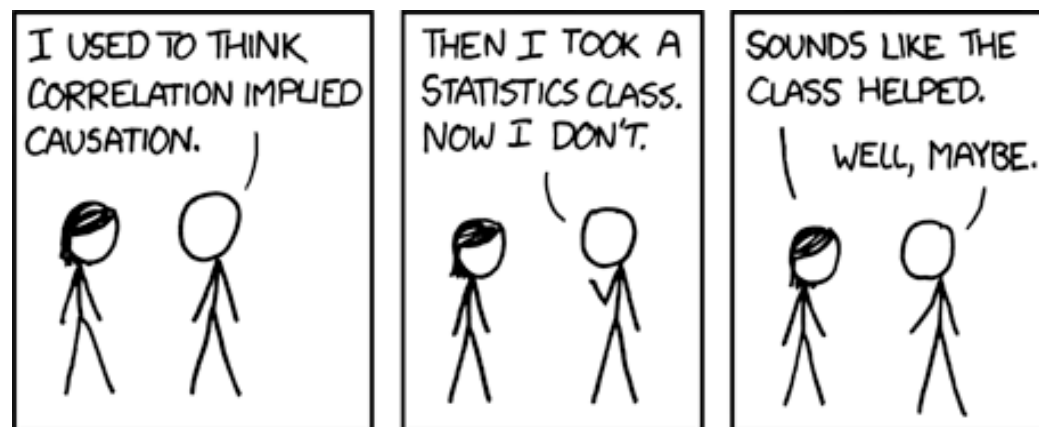


Interpretation von Korrelationen

- In der "echten Welt" haben wir es fast immer mit komplexen Steuerungen zu tun.
- Die Frage ist, wie wir mit den **Kovariaten** umgehen:
 - Konstant halten (wie im Labor)
 - Zufällig verteilen
 - Rechnerisch kontrollieren (also über statistische Modelle)
 - Vernachlässigen

Interpretation von Korrelationen

- Korrelation = gemeinsames Variieren von Variablen
- Vorsicht bei der **kausalen** Interpretation von Korrelationen!



Interpretation von Korrelationen

- Korrelationen sind zunächst einmal **Koinzidenzen**.
- Kausaler Zusammenhang kann – wenn überhaupt – nur angenommen werden, wenn eine Variable **systematisch variiert** wird.
 - Trainingszeit → Erfolg
 - Dosis → Wirkung
- Experimentelle Designs!
- Interpretation von Korrelationen bleibt immer **bidirektional**.

Pearson-Korrelation: Berechnung

Abweichungen jedes x-Werts von seinem Mittelwert

Abweichungen jedes y-Werts von seinem Mittelwert

Produkte werden aufsummiert

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{n \cdot s_x \cdot s_y}$$

Nenner: Produkt aus Stichprobengröße und den beiden Standardabweichungen

Abweichungen werden für jedes Wertepaar multipliziert.

Pearson-Korrelation: Berechnung

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{n \cdot s_x \cdot s_y}$$

- Was geschieht mit den Produkten im Zähler, wenn x_i und y_i beide nach oben von ihrem Mittelwert abweichen?
- Was geschieht mit den Produkten, wenn x_i nach unten und y_i nach oben vom Mittelwert abweicht?
- Was geschieht mit den Produkten, wenn x_i gleich seinem Mittelwert ist und y_i in eine Richtung abweicht?

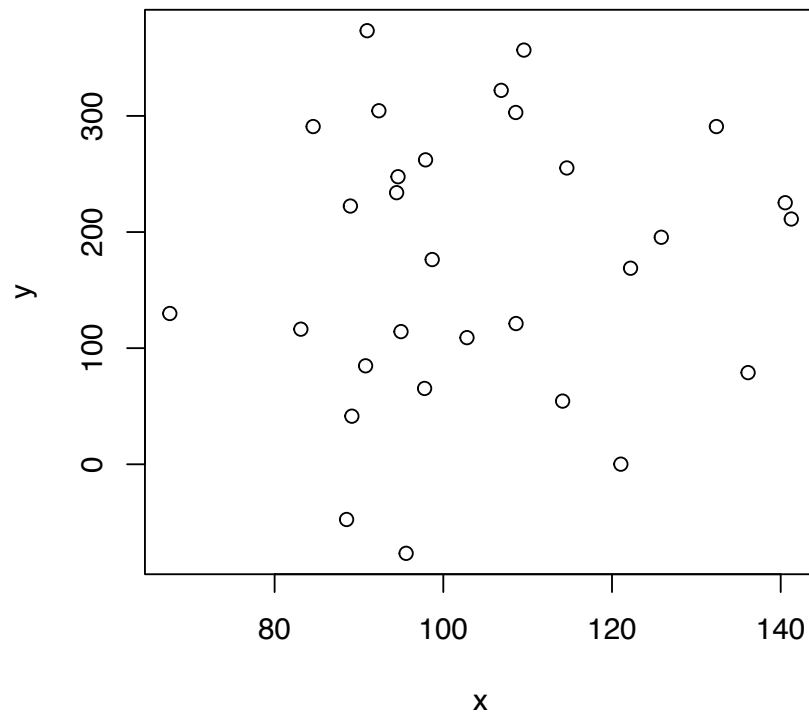
Pearson-Korrelation: Berechnung

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{n \cdot s_x \cdot s_y}$$

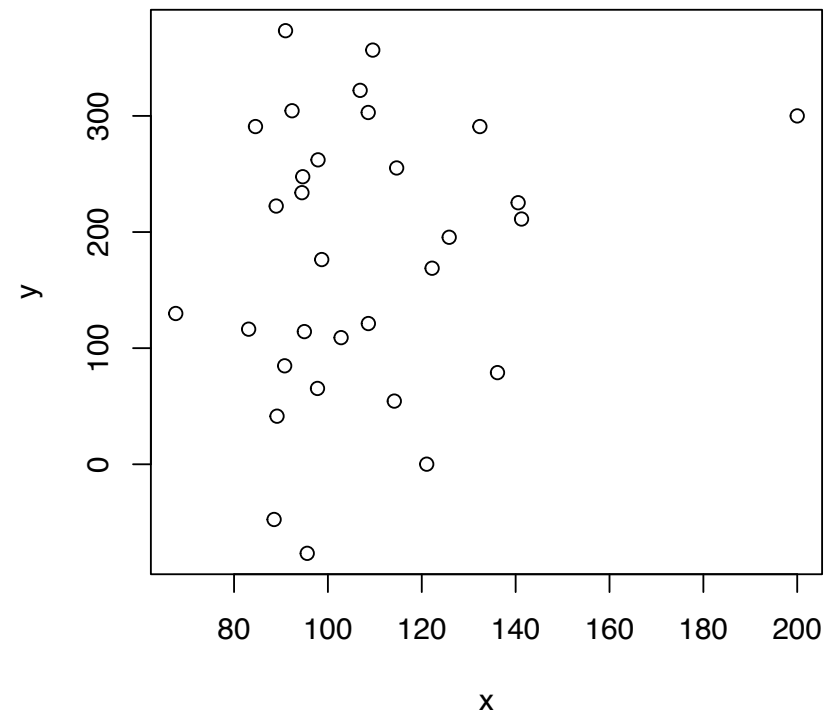
- Was geschieht mit r , wenn die Summe der Produkte sehr hoch positiv / negativ ist (der **Effekt**)?
- Was geschieht mit r , wenn eine der beiden Standardabweichungen s_x oder s_y sehr hoch ist (das **Rauschen**)?

Pearson-Korrelation: Ausreisser

$r = 0,099$



$r = 0,202$



Rangkorrelation nach Spearman

- Verwendung von Rangplätzen anstatt tatsächlicher Werte
- Einige Vorteile:
 - Testet nicht nur auf linearen Zusammenhang
 - Macht keine Vorannahmen zur zugrundeliegenden Verteilung
 - Weniger anfällig für Effekte von Ausreisserwerten
- Rangkorrelation gilt gemeinhin als etwas konservativer.

Zusammenhänge

- Datensatz zu mehreren Ländern der Erde mit den folgenden Variablen:
 - Bevölkerungsdichte (Menschen pro m²)
 - Urbanisierung (Anteil Menschen, der in Städten lebt, in %)
 - Alphabetisierung (Anteil Menschen, die lesen und schreiben können)
 - Ernährung (durchschnittliche tägliche Kalorienaufnahme)
 - Sterberate (wie viele Menschen von 1000 sterben pro Jahr)
- Wie können diese Variablen untereinander korrelativ zusammenhängen?
 - ++ / + / o / - / --

Zusammenhänge

	Bev.dichte	Urbanisierung	Alphabet.rate	Ernährung	Sterberate
Bev.dichte	1				
Urbanisierung		1			
Alphabet.rate			1		
Ernährung				1	
Sterberate					1

Zusammenhänge

	Bev.dichte	Urbanisierung	Alphabet.rate	Ernährung	Sterberate
Bev.dichte	1				
Urbanisierung	0,341	1			
Alphabet.rate			1		
Ernährung				1	
Sterberate					1

Korrelation in R

- Funktion: `cor(x, y, method)`
- Argument `method`:
 - Default: "pearson"
 - Rangkorrelation: "spearman"
- `cor(dat$wlen, dat$wfreq, method = "spearman")`
 - Korreliert die Spalten `wlen` und `wfreq` im Datensatz `dat`
 - Benutzt den Rangkorrelationskoeffizienten von Spearman
- Auch hier: Argument `na.rm`

Zusammenfassung

- Korrelationskoeffizient gibt an, wie stark zwei Variablen **kovariieren** (= gemeinsam von ihrem jeweiligen Mittelwert abweichen).
- Vorsicht bei:
 - Interpretation von **Kausalität**
 - Interpretation der **Effektrichtung** (Korrelation ist immer bidirektional!)
 - Ausreissern
- Bei der **Rangkorrelation** wird mit dem Rang der Werte und nicht mit den Werten selbst gerechnet.

Begriffe

Korrelationskoeffizient

Pearson

Spearman

Einseitige Steuerung

Gegenseitige Steuerung

Drittseitige Steuerung

Komplexe Steuerung

Kovariaten

Korrelation & Kausalität

Koinzidenz

bidirektional

Effekt & Rauschen



Übung

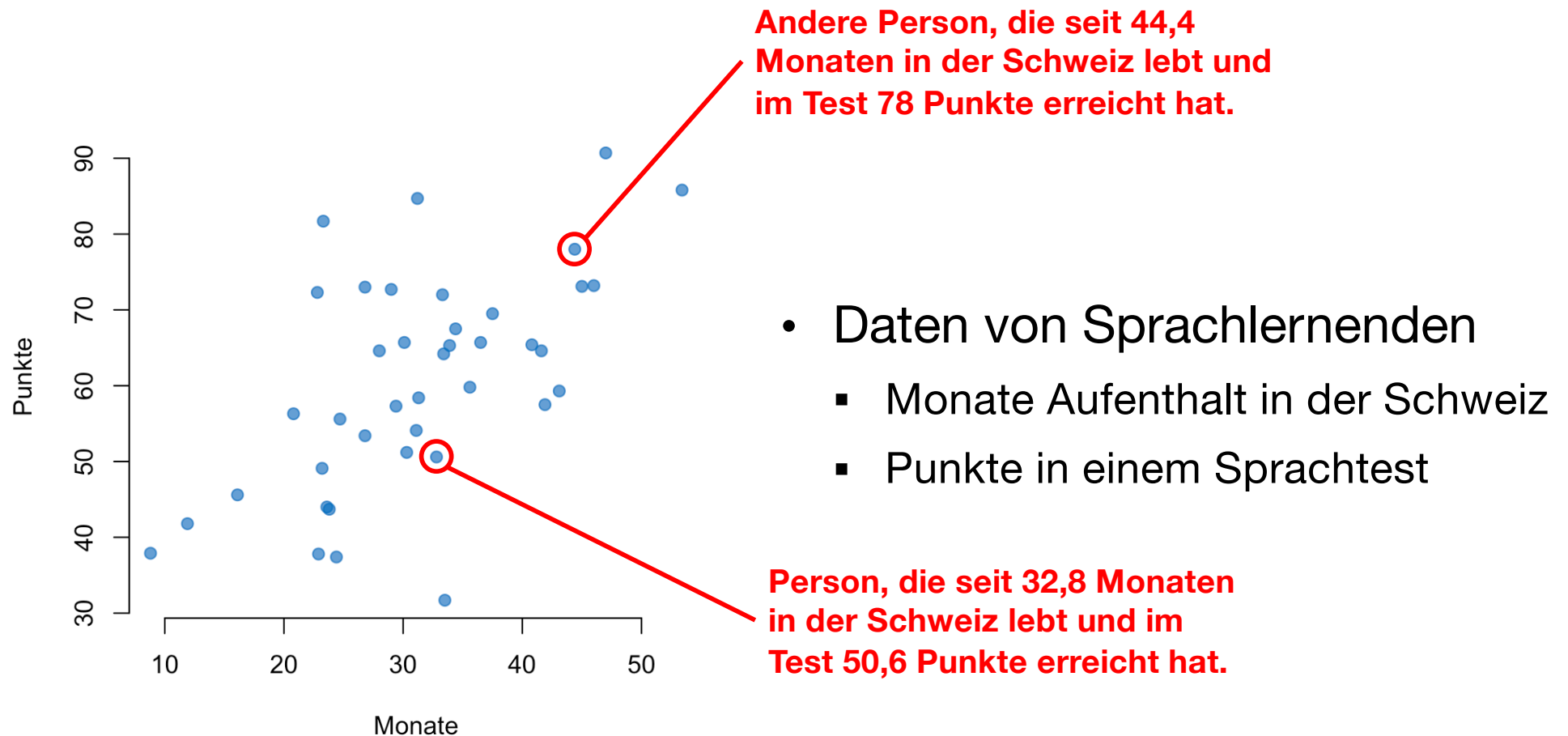


- Wie hoch ist die Spearman-Korrelation zwischen Hubraum (displacement, `disp`) und PS (horse power, `hp`) im Datensatz `mtcars`? Bilden Sie vor der Berechnung eine Hypothese zur Richtung der Korrelation heraus.

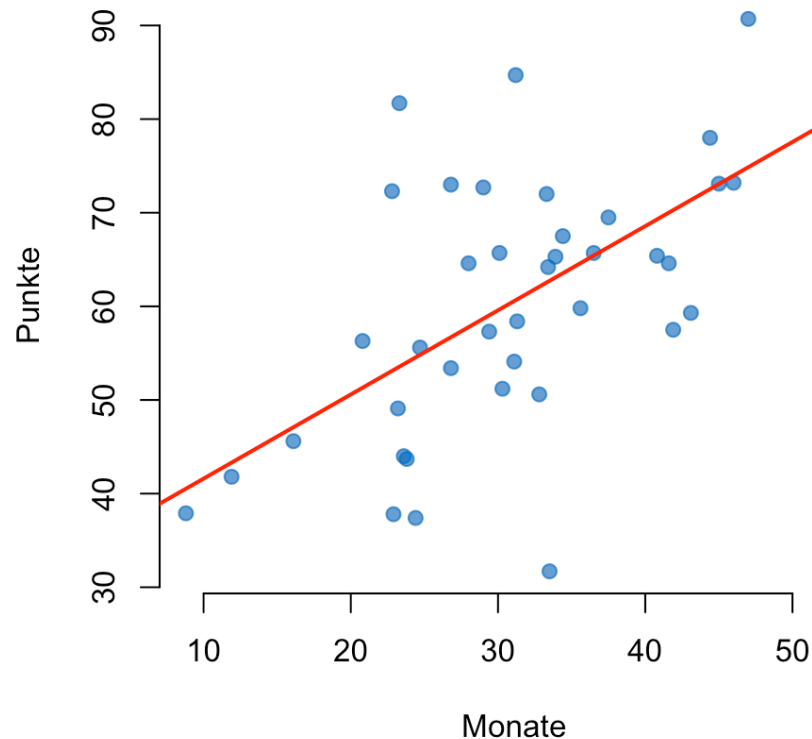


Regression

Daten



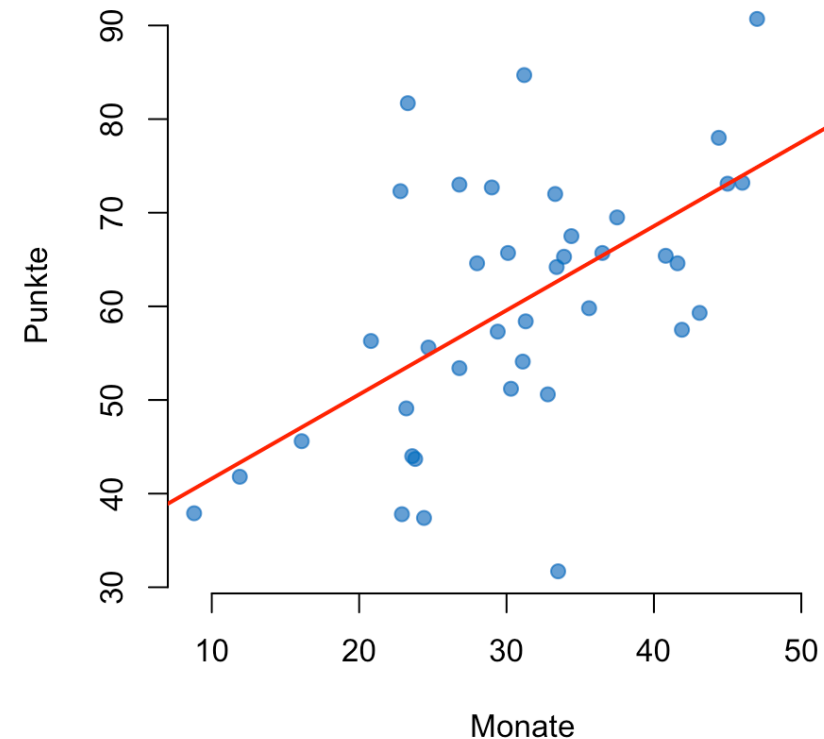
Lineare Regression



- Optimale Beschreibung einer Punktwolke durch eine Gerade
 - Modell wird angepasst oder *gefittet*.
- Zusammenhang zur Korrelation:
 - Positive Korrelation → Gerade steigt ("positive Steigung")
 - Negative Korrelation?
 - Korrelation gleich 0?

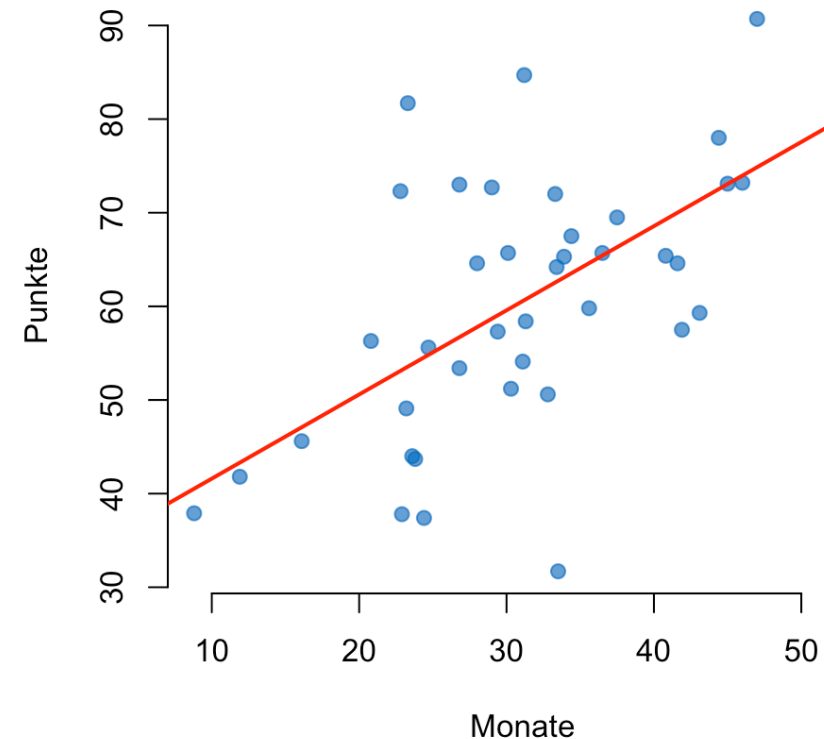
Lineare Regression

- Regressionsgeraden sind definiert durch zwei Parameter:
 - y -Achsenabschnitt / *Intercept* **a**
 - Steigung / *Slope* **b**
- y -Wert = Intercept + Slope \cdot x -Wert
- $y = a + b \cdot x$



Regressionsparameter

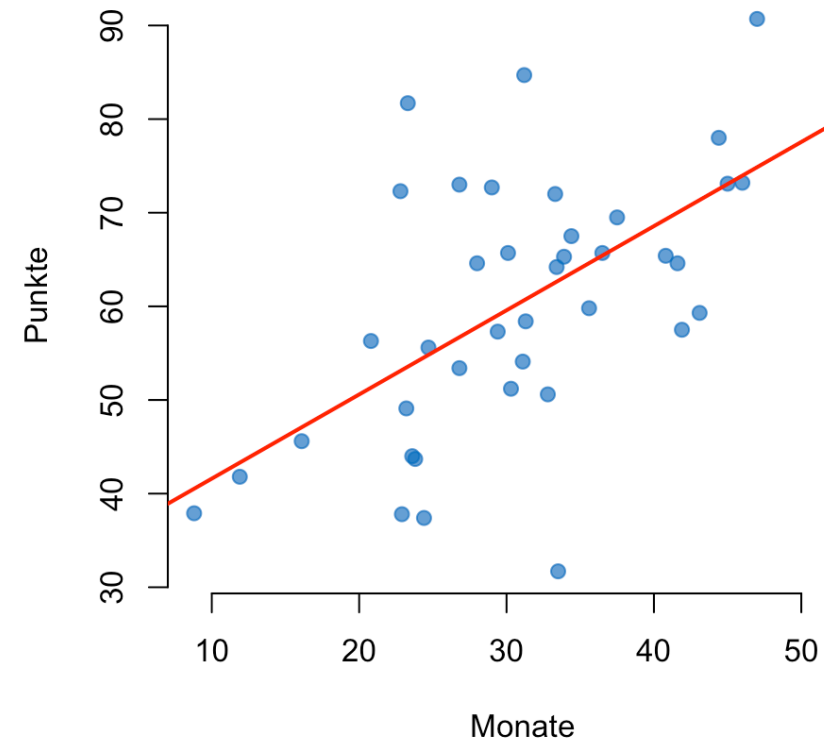
- Parameter sind interpretierbar.
 - Slope b : Um wie viel Punkte steigt das Testergebnis mit jedem Monat Aufenthalt in der Schweiz?
 - Intercept a : Wie viel Punkte erzielt man, wenn man noch nicht in der Schweiz war (Monate = 0)?
- Hier: Pro Monat ca. 1 Punkt mehr, 32,6 Punkte bei 0 Monaten.



$$\text{Punkte} = 32,6 + 0,9 \cdot \text{Monate}$$

Regression als Vorhersage

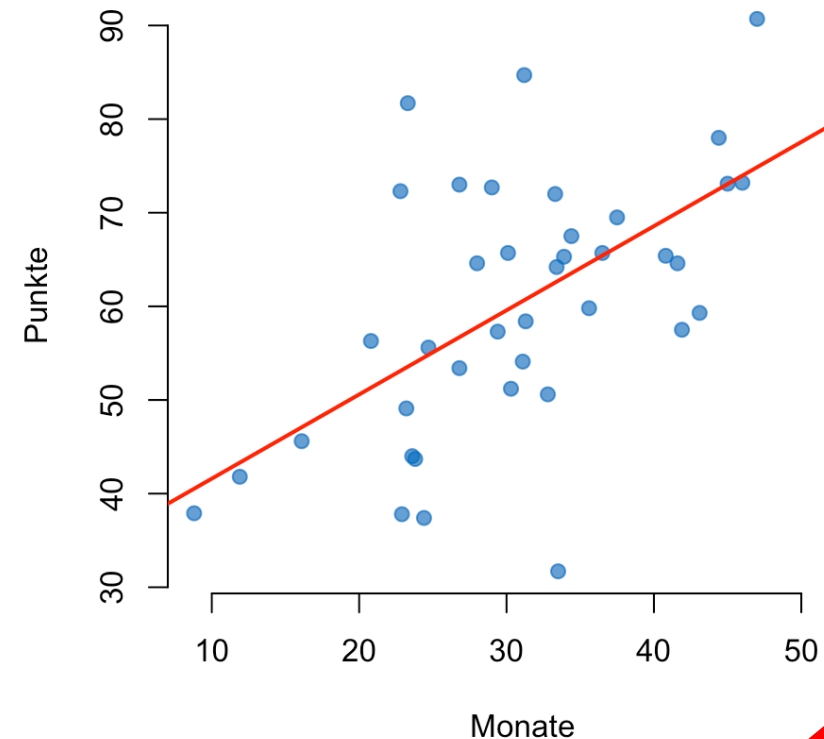
- Aufgrund der Gerade können wir bei neuen Werten von x vorhersagen, welchen y -Wert wir erwarten würden.



$$\text{Punkte} = 32,6 + 0,9 \cdot \text{Monate}$$

Regression als Vorhersage

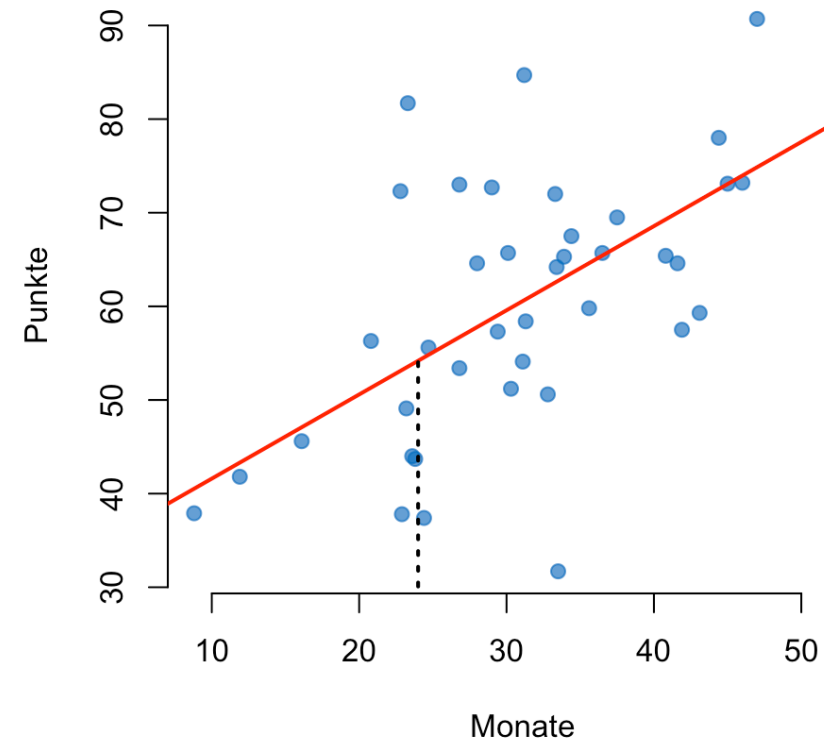
- Uns wird eine Sprachschülerin vorgestellt, die seit zwei Jahren in der Schweiz lebt.
- Wie viel Punkte wird sie wohl in dem Test erzielen?



$$\text{Punkte} = 32,6 + 0,9 \cdot 24$$

Regression als Vorhersage

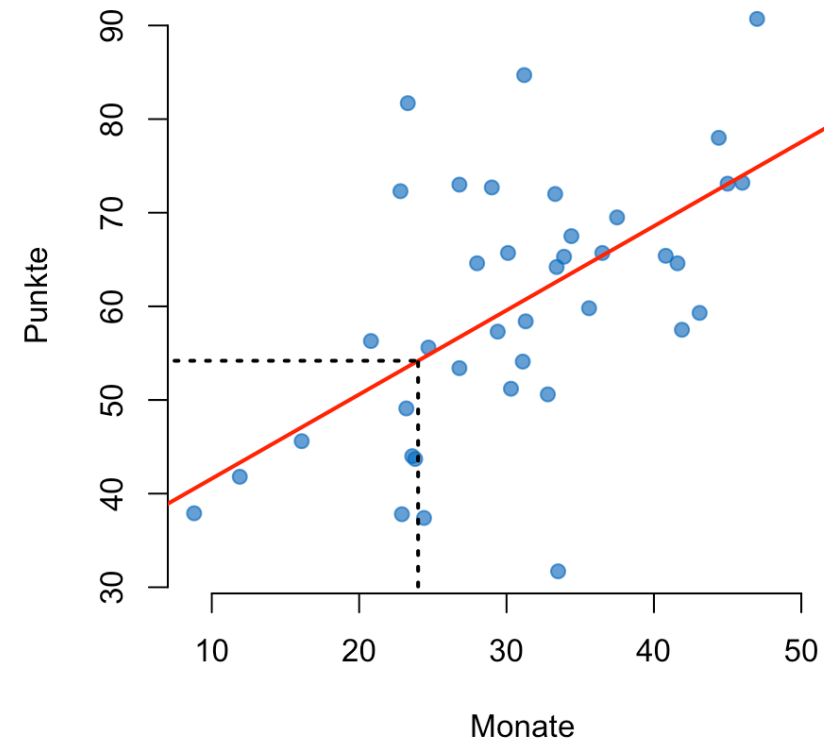
- Uns wird eine Sprachschülerin vorgestellt, die seit zwei Jahren in der Schweiz lebt.
- Wie viel Punkte wird sie wohl in dem Test erzielen?



$$\text{Punkte} = 32,6 + 0,9 \cdot 24$$

Regression als Vorhersage

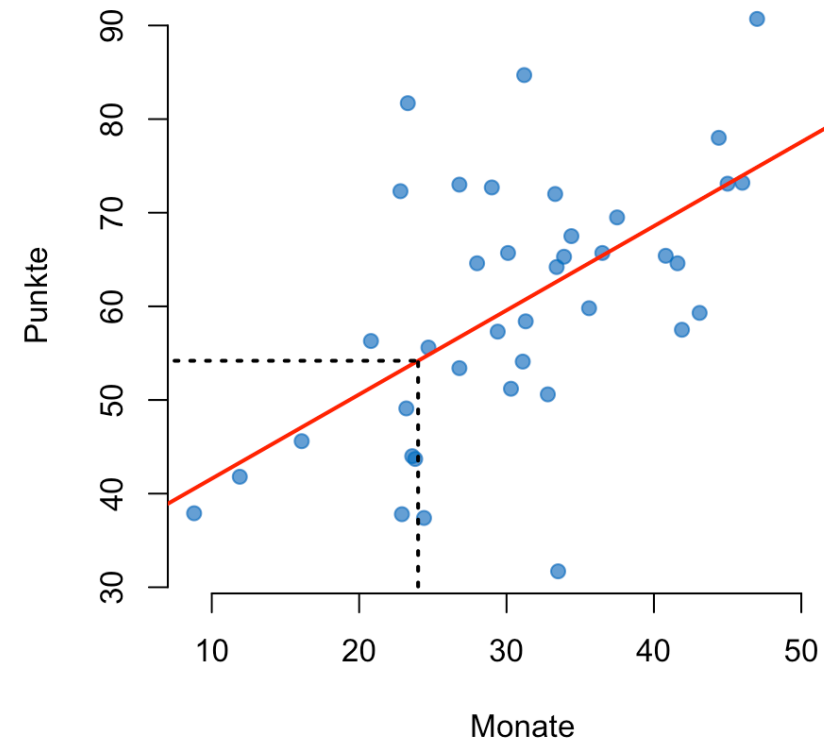
- Uns wird eine Sprachschülerin vorgestellt, die seit zwei Jahren in der Schweiz lebt.
- Wie viel Punkte wird sie wohl in dem Test erzielen?



$$54,2 = 32,6 + 0,9 \cdot 24$$

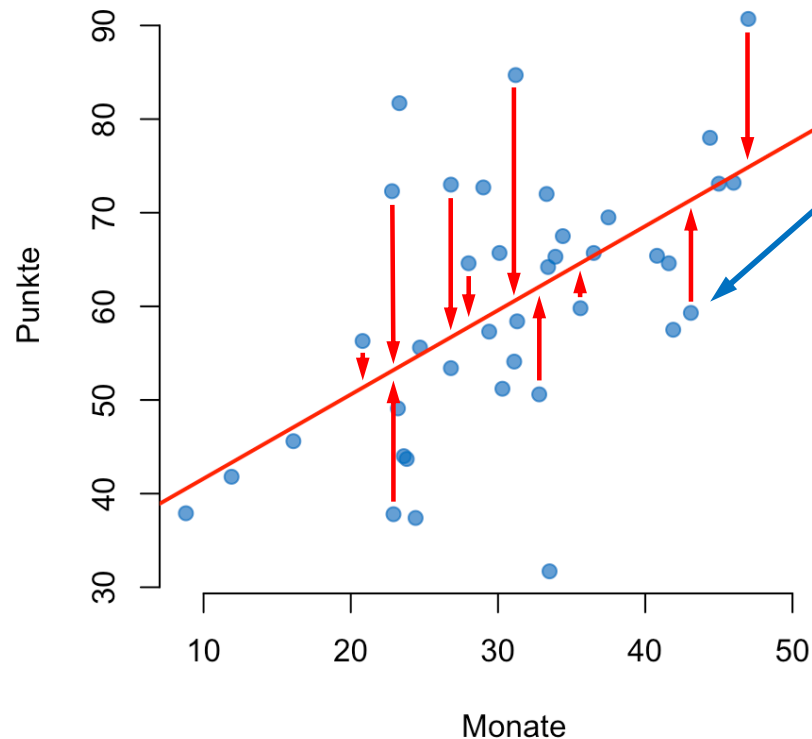
Regression als Vorhersage

- Natürlich können wir uns mit dieser Vorhersage irren.
- Sie ist aber unser "best guess" gegeben die Daten, die wir bisher gesammelt haben.



$$54,2 = 32,6 + 0,9 \cdot 24$$

Residuen (= Vorhersagefehler)



Hat diese Person "zu wenige" Punkte erreicht?

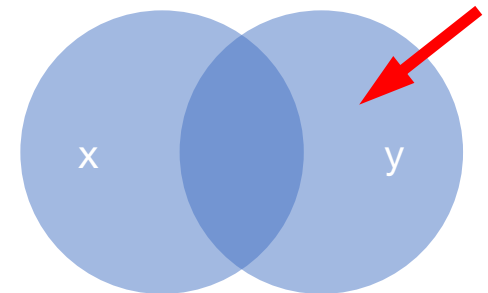
Oder ist gar unsere Regressionsgerade fehlerhaft?

- Die Person hat im Vergleich zu allen anderen und gegeben ihre Aufenthaltsdauer in der Schweiz zu wenige Punkte erreicht.
- Unsere Regressionsgerade beschreibt die Daten, die wir haben, optimal.

Das heisst: Die Gerade minimiert die Abweichung aller Punkte zur Geraden (= Summe der Residuen).

Residuen sind interpretierbar

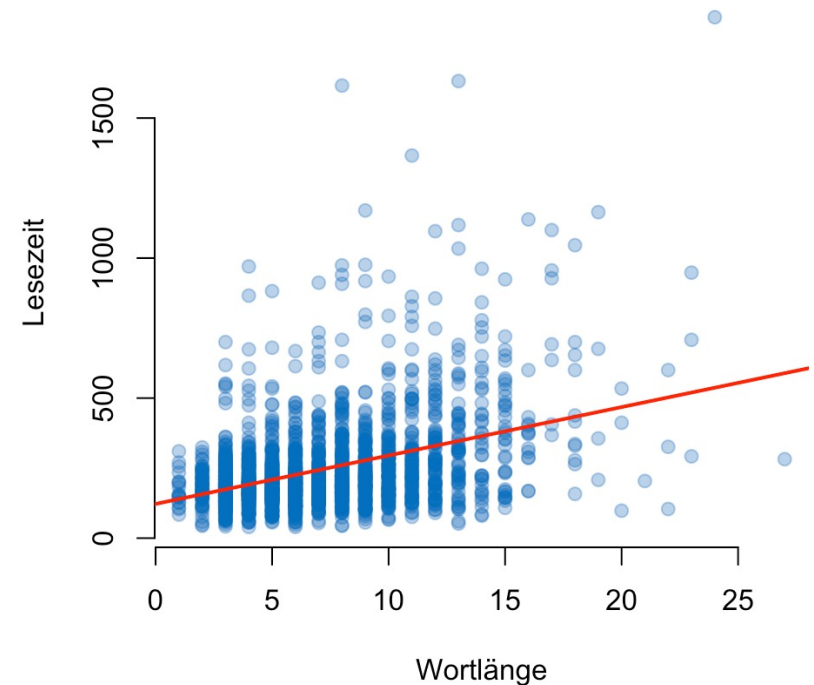
- Residuen sind das, was durch die Vorhersage von x auf y nicht erklärt werden kann.
 - Unaufgeklärte Varianz
- Was könnte das in unserem Beispiel sein?
 - Allgemeine Sprachfähigkeit
 - Kenntnis zusätzlicher Zweitsprachen
 - Kenntnis anderer germanischer Sprachen
 - Kontakthäufigkeit mit deutschsprachigen Personen in der Schweiz
- **Kovariaten**



Residuen sind manchmal nützlich

- Manchmal interessieren uns nur die Residuen einer Regressionsberechnung.
- Wortlänge (x) hat einen Einfluss auf die Lesezeit (y).
- Das ist aber einigermaßen trivial und interessiert uns eigentlich nicht.
- Wir können also die Regressionsgerade berechnen, wo wir Wortlänge durch die Lesezeit vorhersagen...
- ...und dann die Residuen extrahieren ("residual reading times").
- Damit rechnen wir dann weiter.

$$\text{Lesezeit} = 122,2 + 17,3 * \text{Wortlänge}$$

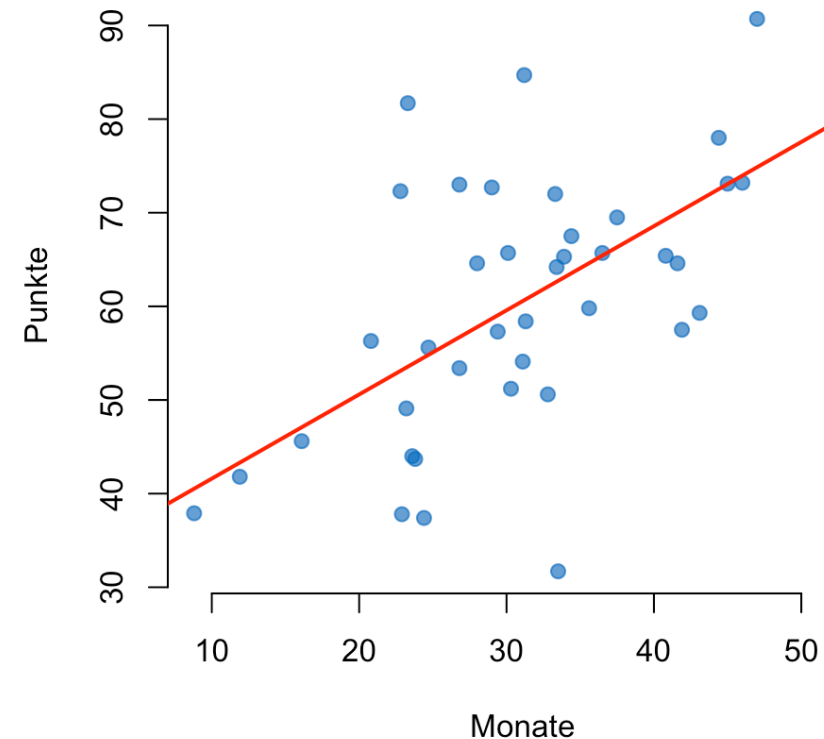


Residuen sind manchmal nützlich

- Wann wird phonetisch reduziert?
 - "going to" → "gonna"; "want to" → "wanna"
 - "natürlich" → "tüich"
- Interessante Faktoren: z. B. Vorhersagbarkeit aus dem Satzkontext, Korpusfrequenz usw.
- Vorher sollte jedoch dringend die Sprechgeschwindigkeit / *speech rate* "herausgerechnet" werden.

Korrelation und Regression

- Korrelation hier:
 - Pearson: $r = 0,60$
 - Spearman: $r = 0,56$
- Aber: Korrelationen sind **bidirektional**.
- Regressionen sind **gerichtet**: Vorhersage von y aus x .
 - y -Abweichungen (= Residuen) werden minimiert.
 - Bei Vorhersage von x aus y werden x -Abweichungen minimiert, dabei verändert sich die Regressionsgerade.



Multiple Regression

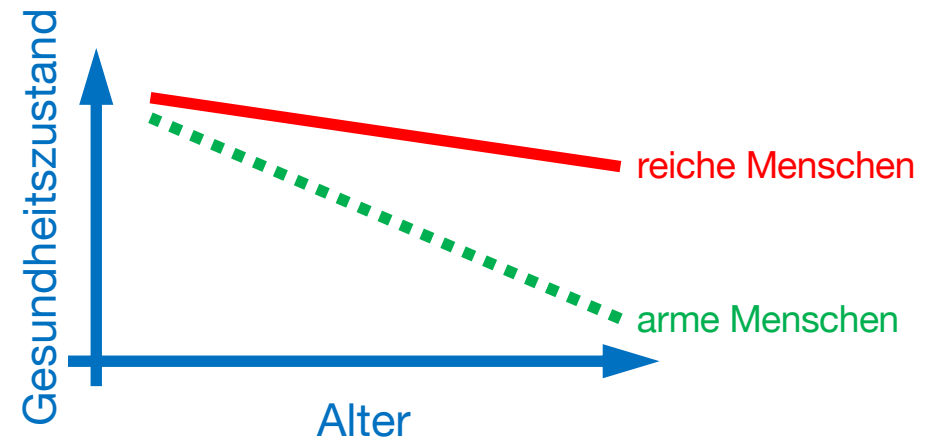
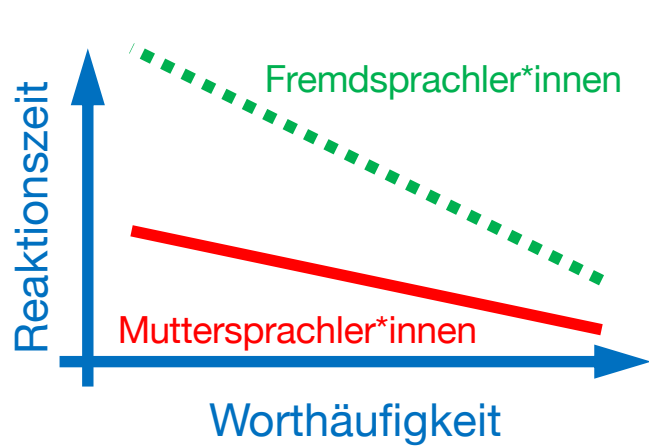
Prädiktoren werden auch **unabhängige Variablen** genannt. Die Kriteriumsvariable wird auch **abhängige Variable** genannt.

- Bisher haben wir eine y -Variable aus einer x -Variable vorhergesagt.
- Typischerweise benutzen wir mehrere **Prädiktoren**, um die **Kriteriumsvariable** vorherzusagen.
- Beispiele:
 - Korpusfrequenz + Wortart + Einbettungstiefe → Lesezeit
 - Ticketpreis + Wetter + Beliebtheit der Band → Anzahl Konzertbesucher
 - Fahrgastaufkommen + Wetter + Streckenzustand → Verspätungen
 - ...

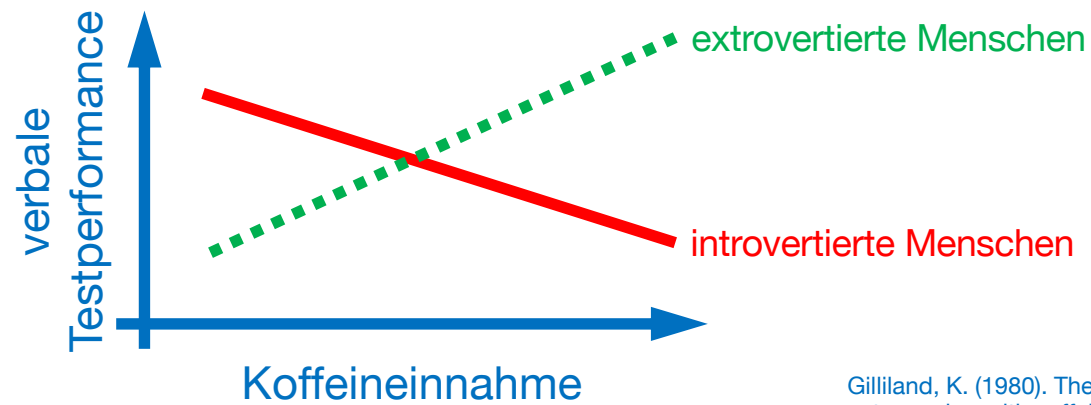
Multiple Regression

- Bei der multiplen Regression bekommt jeder Prädiktor seine eigene Steigung.
 - Auch: **(β -)Gewicht, Koeffizient, *coefficient, estimate***
- Neben den **Einzeleffekten** (*single/main effects*) sind auch **Interaktionen** möglich.
 - Interaktion: Das Zusammenwirken von zwei oder mehr Prädiktoren auf die Kriteriumsvariable.
 - Beispiele:
 - Das Wetter hat nur bei unbeliebteren Bands einen Einfluss auf die Anzahl der Gäste.
 - Je höher ein Wort eingebettet ist, desto mehr Einfluss hat die Worthäufigkeit auf die Lesezeit.

Interaktionen: Beispiele



Quelle: Datensatz `lexdec` aus `{languageR}`



Regression: Voraussetzungen

Linearität des Zusammenhangs

Das Kriterium kann als eine lineare Kombination der Prädiktoren ausgedrückt werden.

Varianzhomogenität der Residuen

Die Fehlervarianz ist überall ungefähr gleich.

Normalverteilung der Residuen

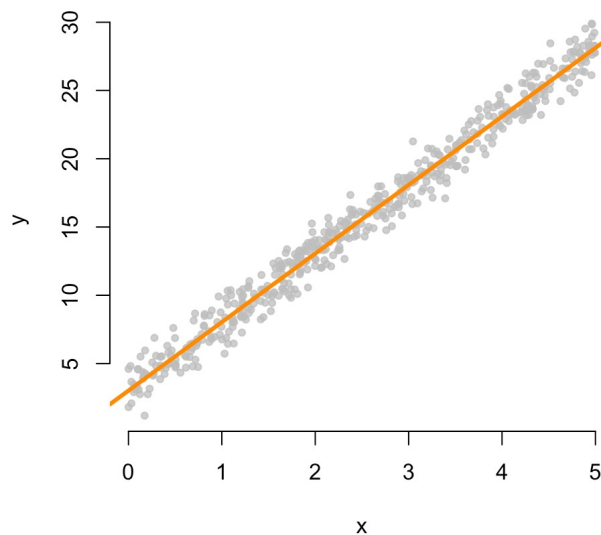
Die Residuen sind normalverteilt.

Voraussetzungen: Linearität

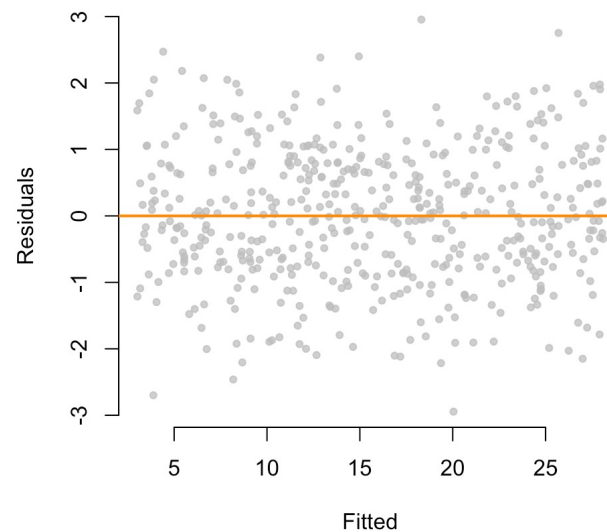
Fitted sind die geschätzten Werte, also die Werte auf der Regressionsgeraden.

- Keine große Überraschung: Lineare Regressionen können nur lineare Zusammenhänge erfassen.
- Nützlicher Diagnostik-Plot: *Fitted vs. residuals*

Modell 1 (Daten und Fit)



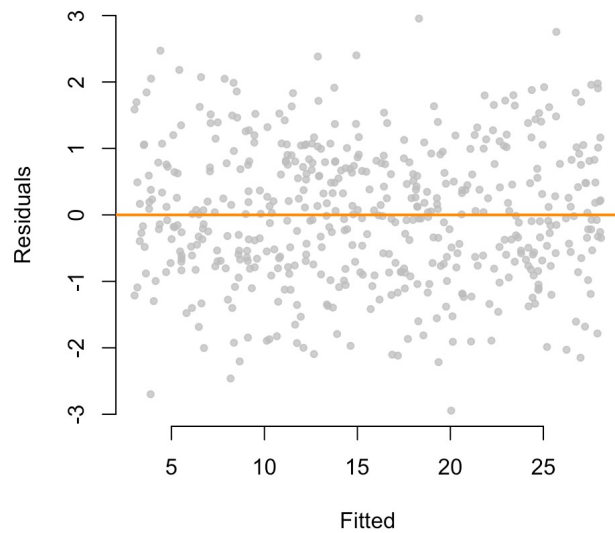
Modell 1 (Fitted vs. residuals)



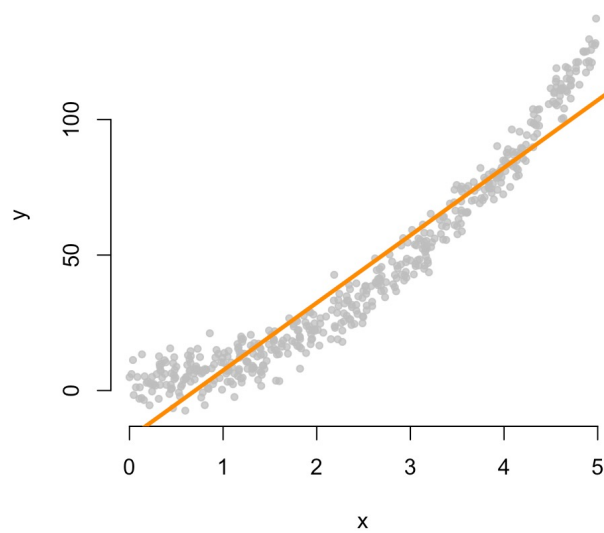
Linearität ist gegeben, wenn die Residuen sich gleichmäßig um den Fit verteilen und keine eindeutige Abweichung erkennbar ist.

Voraussetzungen: Linearität

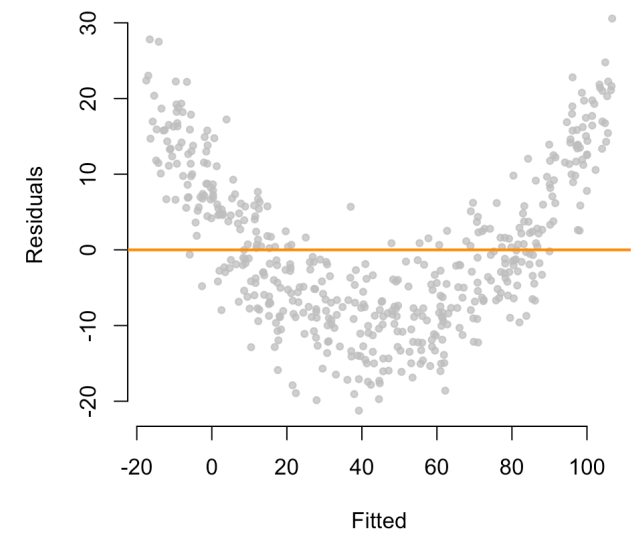
Modell 1 (Fitted vs. residuals)



Modell 3 (Daten und Fit)

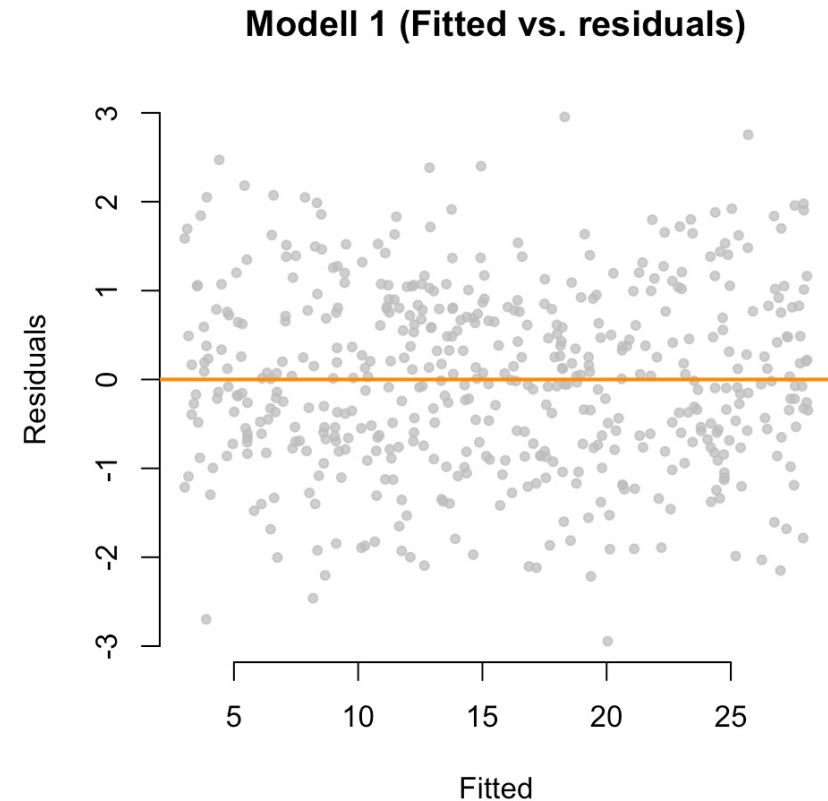


Modell 3 (Fitted vs. residuals)



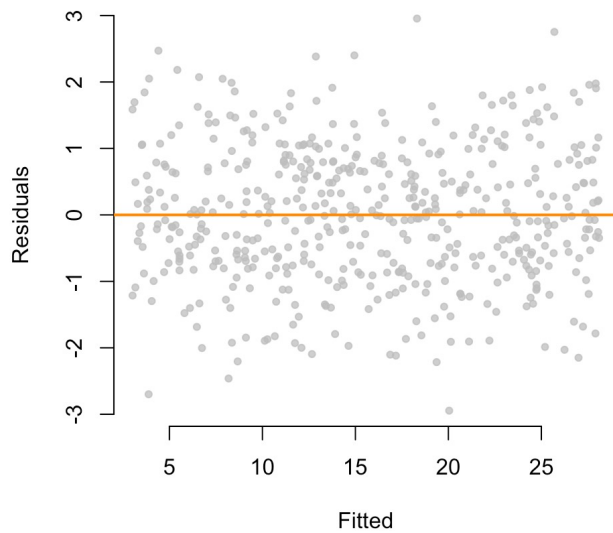
Voraussetzungen: Varianzhomogenität

- Die Varianz der Residuen muss für alle Abschnitte des Prädiktors (der Prädiktoren) ungefähr gleich sein.
- **Heteroskedastizität** ist die Verletzung dieses Prinzips.

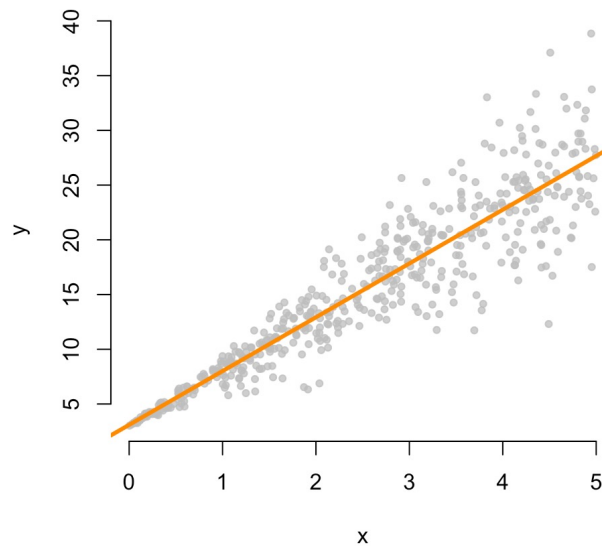


Voraussetzungen: Varianzhomogenität

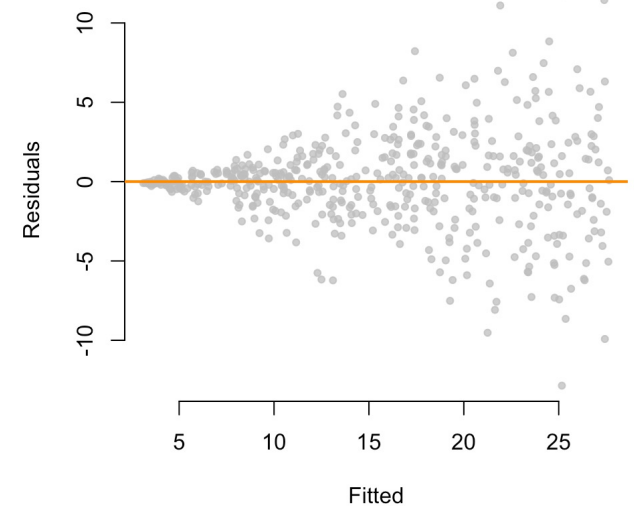
Modell 1 (Fitted vs. residuals)



Modell 2 (Daten und Fit)



Modell 2 (Fitted vs. residuals)

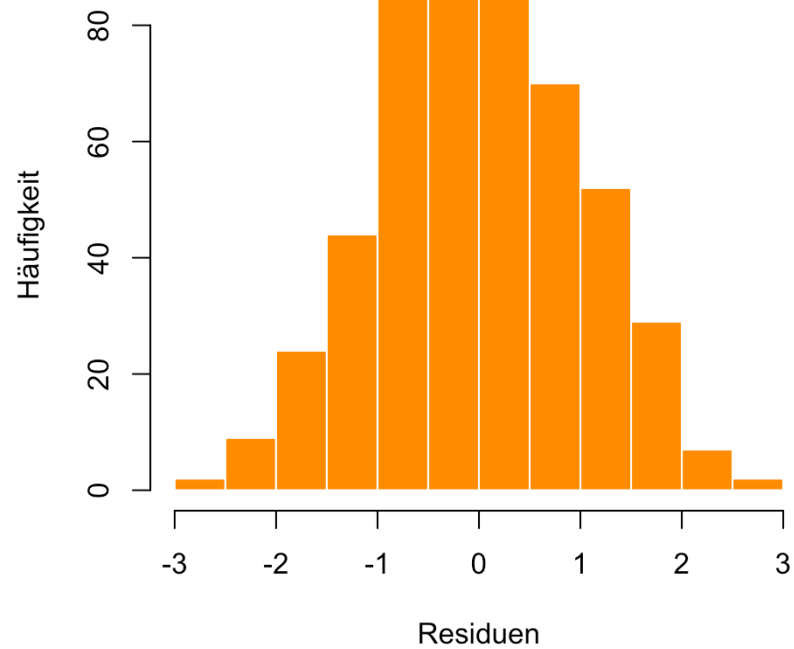


Voraussetzungen: Normalverteilung

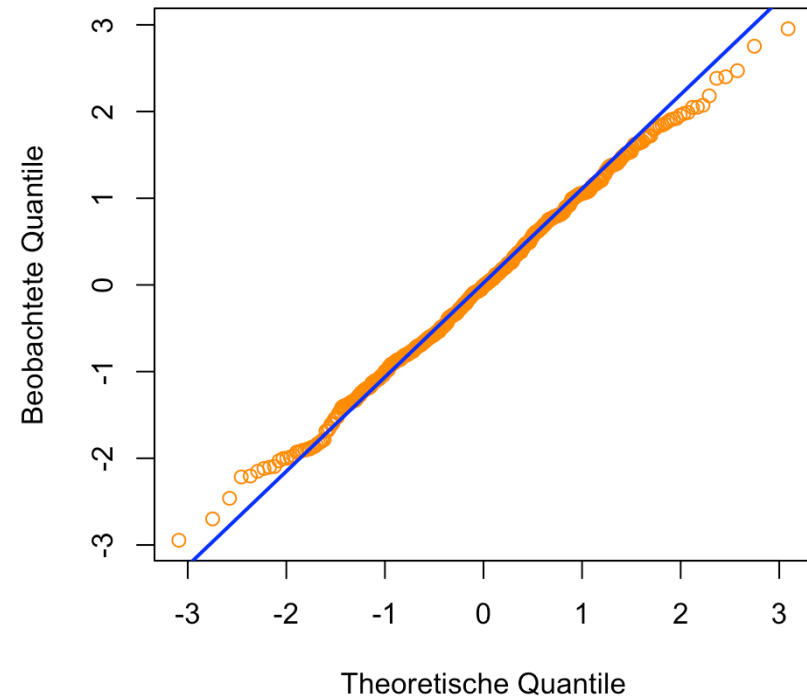
- Residuen müssen normalverteilt sein.
- Geeignete Diagnostik-Plots: Histogramm und QQ-Plot für die Residuen aus dem Regressionsmodell
- Histogramm zeigt die Anzahl an Datenpunkten in bestimmten Abschnitten (= Verteilung).
- QQ-Plot plottet theoretische Quantile (laut Normalverteilung) gegen beobachtete Quantile.

Voraussetzungen: Normalverteilung

Histogramm Residuen Modell 1

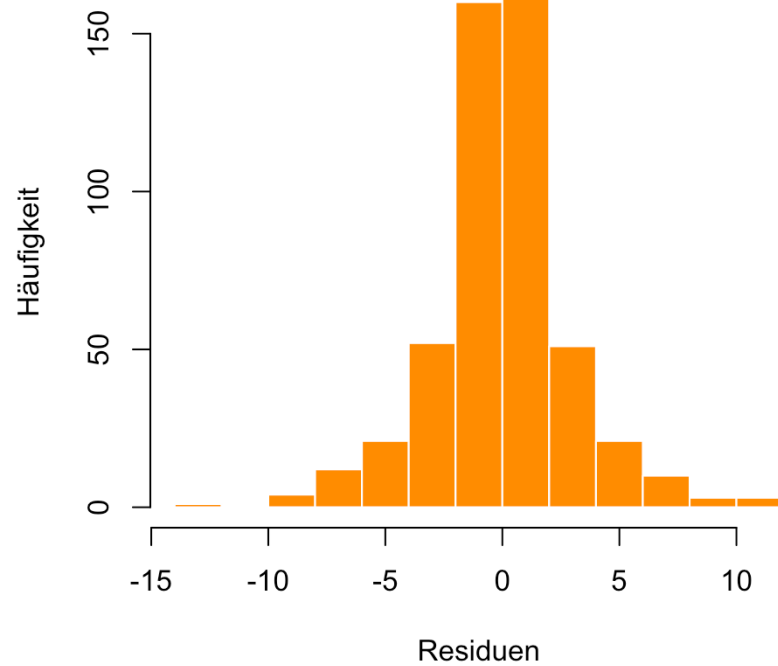


QQ-Plot Modell 1

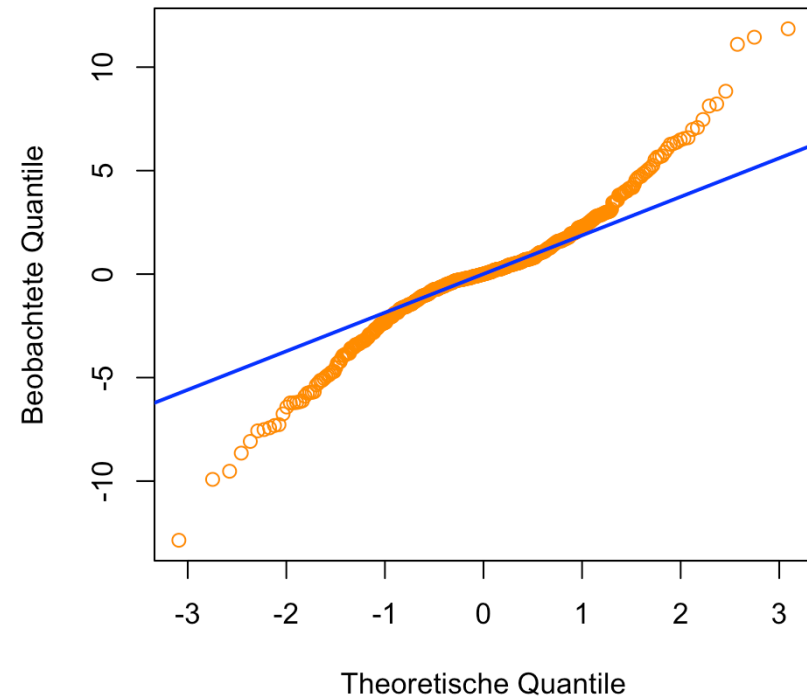


Voraussetzungen: Normalverteilung

Histogramm Residuen Modell 2

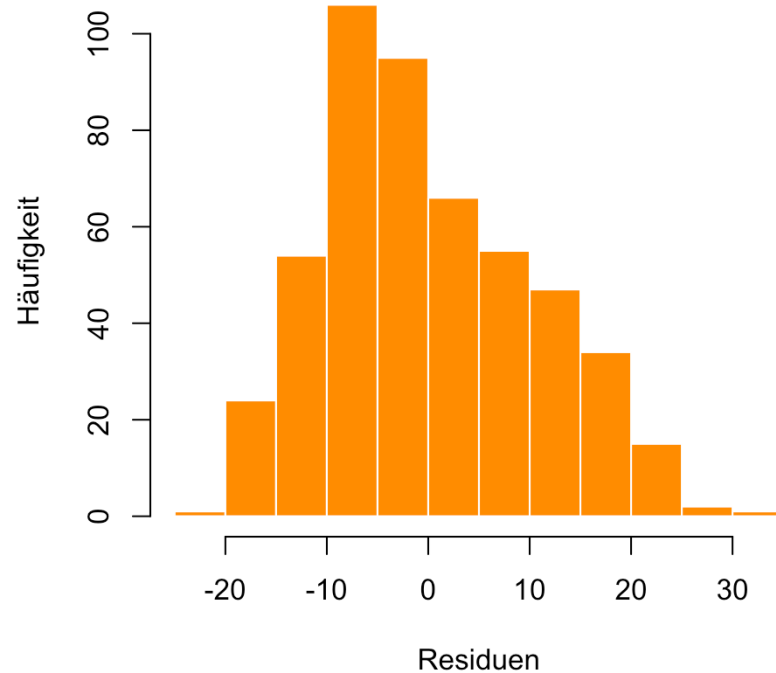


QQ-Plot Modell 2

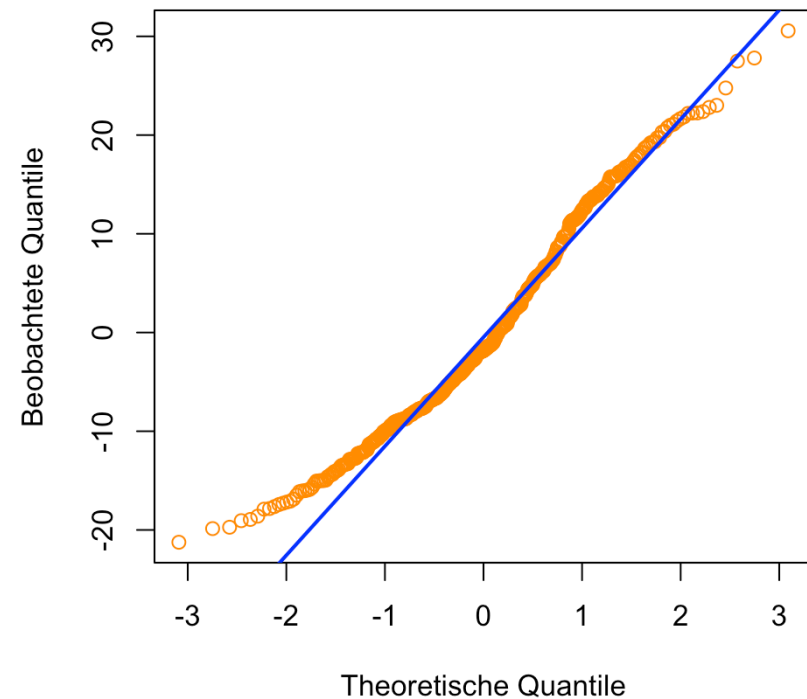


Voraussetzungen: Normalverteilung

Histogramm Residuen Modell 3



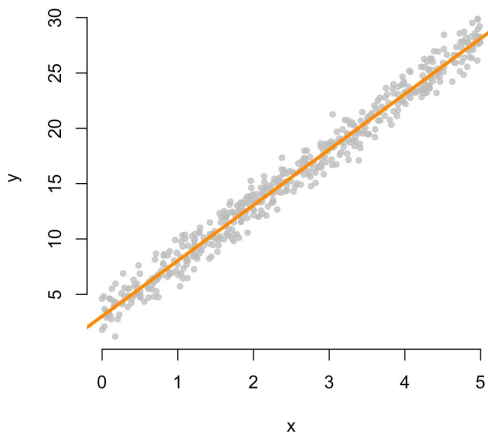
QQ-Plot Modell 3



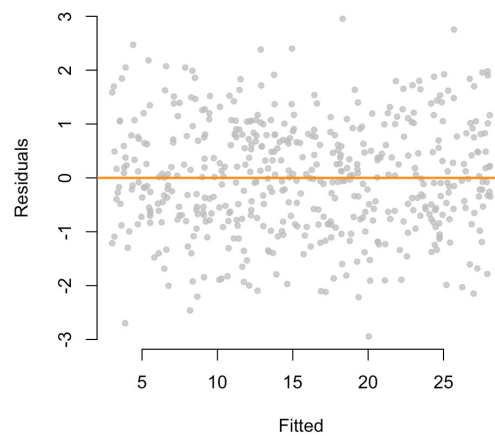
Voraussetzungen

- Von den vorherigen Modellen würde nur Modell 1 alle Voraussetzungen eindeutig erfüllen.

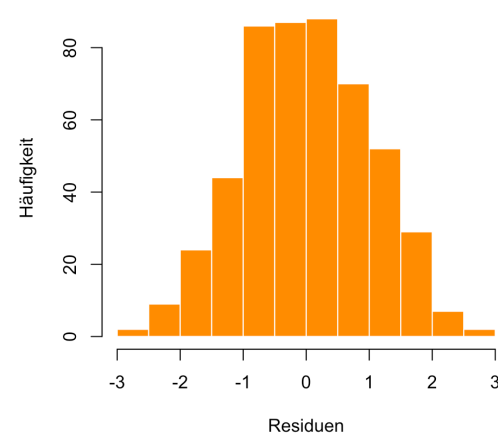
Modell 1 (Daten und Fit)



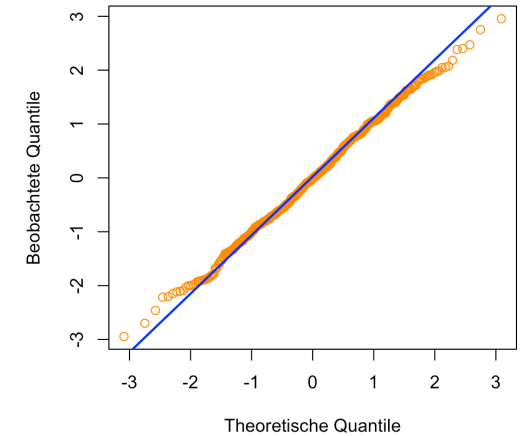
Modell 1 (Fitted vs. residuals)



Histogramm Residuen Modell 1



QQ-Plot Modell 1



Lineare Regression in R

- Vorhersage: Kontinuierliche (metrisch skalierte) Variable
- Prädiktoren: Kontinuierliche oder diskrete Variable(n)
- Funktion: `lm()` – *linear model*
- Syntax: `lm(<Formel>, data = <Datensatz>)`

`<Kriterium> ~ <Prädiktorstruktur>`

"predicted by"

Eine Regression mit einer zweistufigen Prädiktorvariable ersetzt den *t*-Test.

`RT ~ NativeLanguage + Frequency`: single effects

`RT ~ NativeLanguage * Frequency`: single effects und Interaktion

Lineare Regression in R

Aufruf

```
Call:
lm(formula = RT ~ NativeLanguage * Frequency, data = lexdec)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.67232 -0.14728 -0.03079  0.11713  1.06986
```

Verteilung der Residuen

Coefficients:

Effektschätzer & stat. Prüfgröße

p-Werte

- `mod <- lm(...)`
- `summary(mod)`

Prädiktoren

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.466060	0.027796	232.626	< 2e-16	***
NativeLanguageOther	0.286343	0.042459	6.744	2.12e-11	***
Frequency	-0.031098	0.005651	-5.504	4.31e-08	***
NativeLanguageOther:Frequency	-0.027472	0.008631	-3.183	0.00149	**

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

Residual standard error: 0.2218 on 1655 degrees of freedom

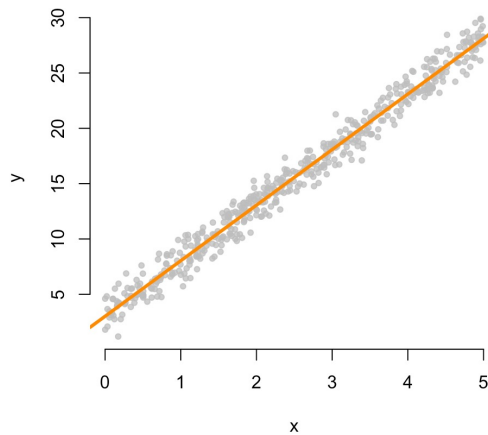
Varianzaufklärung Multiple R-squared: 0.1584, Adjusted R-squared: 0.1569

F-statistic: 103.8 on 3 and 1655 DF, p-value: < 2.2e-16

Regressionsdiagnostik

```
mod <- lm(y ~ x, data = df)
```

Modell 1 (Daten und Fit)



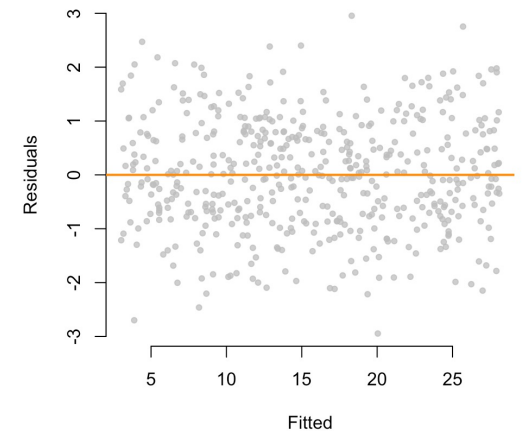
```
plot(df$x, df$y)  
lines(df$x, fitted(mod))
```

```
library(car)  
avPlots(mod)
```

*Für multiple
Regression*

```
plot(fitted(mod), resid(mod))  
abline(h = 0)
```

Modell 1 (Fitted vs. residuals)

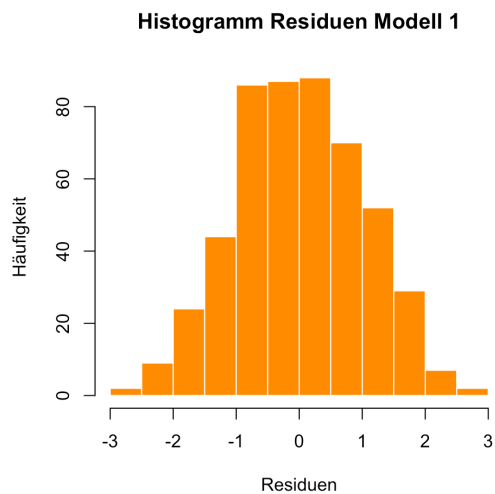


Regressionsdiagnostik

Die meisten Voraussetzungen können auch numerisch getestet werden. Hierzu z.B. <https://book.stat420.org/model-diagnostics.html>

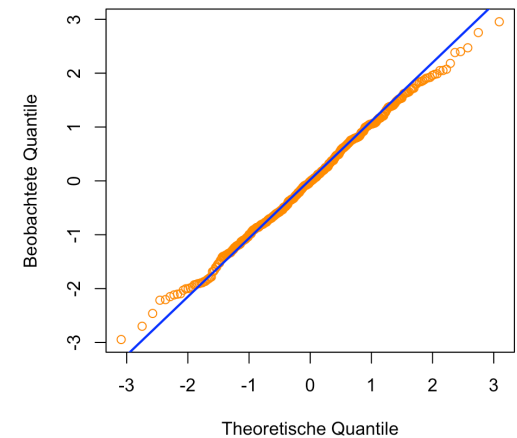
```
mod <- lm(y ~ x, data = df)
```

```
hist(resid(mod))
```



QQ-Plot Modell 1

```
qqnorm(resid(mod))  
qqline(resid(mod))
```



Binär-logistische Regression

- Bei der logistischen Regression wird eine **binäre** Kriteriumsvariable vorhergesagt.
 - korrekt/falsch; vorhanden/nicht vorhanden; eine von zwei Realisierungen; Erfolg/Misserfolg; ja/nein
- Es darf theoretisch keine dritte Möglichkeit denkbar sein! Geschätzt wird die Wahrscheinlichkeit des Eintretens (0 bis 1).
- `glm(<Formel>, data = <Daten>, family = "binomial")`

 ***generalized linear model***

Beispiel: Binär-logistische Regression

Mary gave [the book]_{Theme} to [the man]_{Recipient}.

Realisierung des
Rezipienten in **PP**

Mary gave [the man]_{Recipient} [the book]_{Theme}.

Realisierung des
Rezipienten in **NP**

Frage: Beeinflussen die folgenden Prädiktoren die Realisierung des Rezipienten?

- Modalität (spoken vs. written)
- Länge des **Themes** (in Wörtern)
- Animiertheit des **Rezipienten**

Beispiel: Binär-logistische Regression

Frage: Beeinflussen die folgenden Prädiktoren die Realisierung des Rezipienten?

- Modalität (spoken vs. written)
- Länge des **Themes** (in Wörtern)
- Animiertheit des **Rezipienten**

```
library(languageR)

dative$PP_real <- dative$RealizationOfRecipient == "PP"

mod <- glm(PP_real ~ Modality + LengthOfTheme + AnimacyOfRec,
           data = dative, family = "binomial")
summary(mod)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.72104	0.07104	-10.149	< 2e-16	***
Modalitywritten	1.38698	0.09842	14.092	< 2e-16	***
LengthOfTheme	-0.23628	0.01854	-12.745	< 2e-16	***
AnimacyOfRecinanimate	1.10800	0.14730	7.522	5.38e-14	***

log-odds, umformbar in Wahrscheinlichkeiten

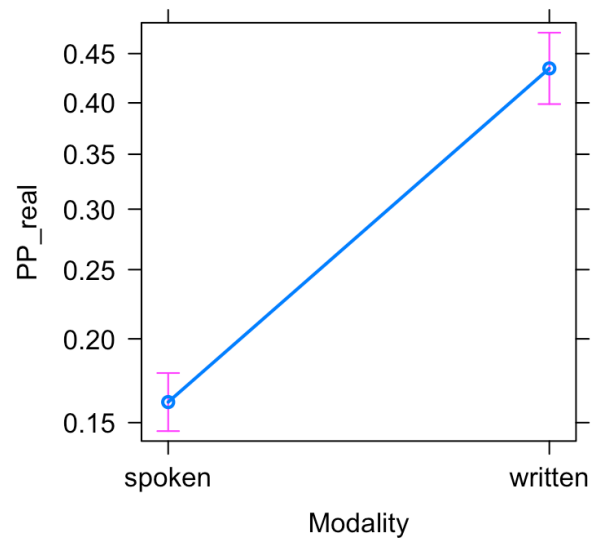
Extrahieren von Effekten: {effects}

```
mod <- glm(PP_real ~ Modality + LengthOfTheme + AnimacyOfRec,  
          data = dative, family = "binomial")  
library(effects)  
plot(allEffects(mod))
```

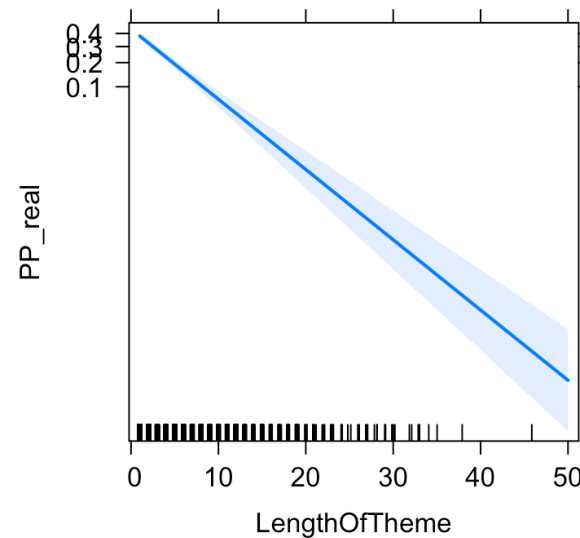
Coefficients:

	Estimate
(Intercept)	-0.72104
Modalitywritten	1.38698
LengthOfTheme	-0.23628
AnimacyOfRecinanimate	1.10800

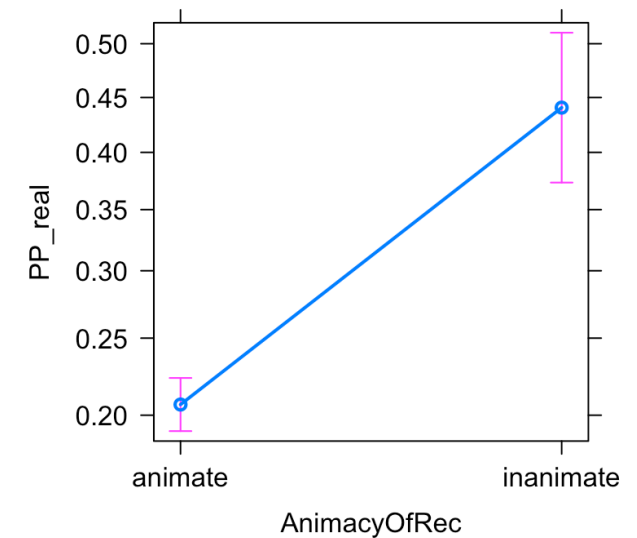
Modality effect plot



LengthOfTheme effect plot



AnimacyOfRec effect plot



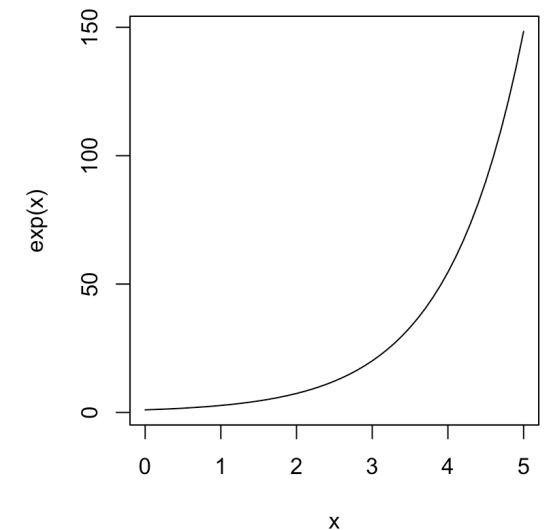
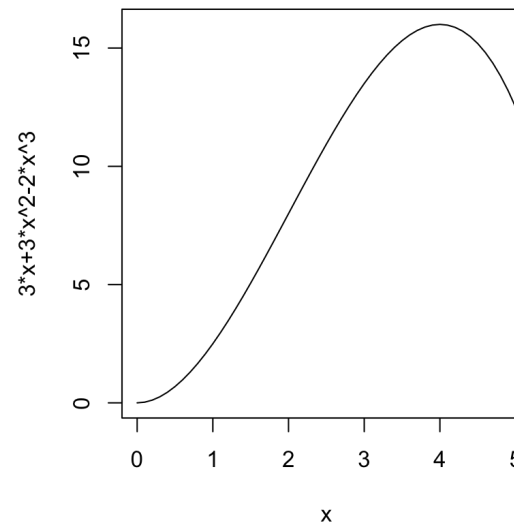
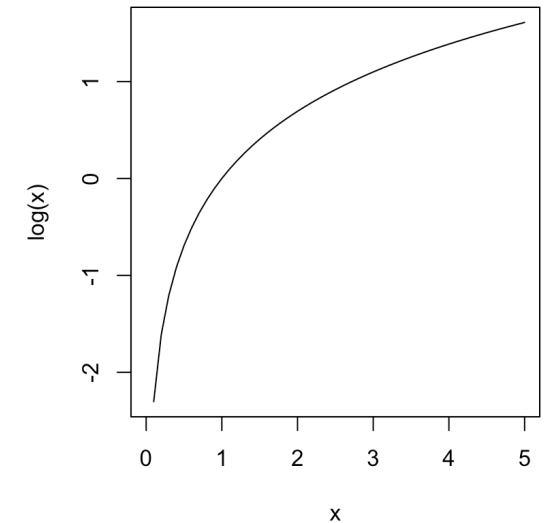
Nicht-lineare Regression

- Fitten eines nicht-linearen Zusammenhangs zwischen Prädiktoren und Kriterium

- Logarithmische Funktionen `log()`
- Exponentialfunktion `exp()`
- Polynome `poly()`
- ...

```
y ~ log(x)  
y ~ exp(x)  
y ~ poly(x, 3)
```

Anzahl der Terme, Vorsicht vor Overfitting!



Zusammenfassung

- Mit Regressionen beschreiben wir den (rechnerischen) Zusammenhang zwischen Variablen.
- Es wird immer **eine** Variable vorhergesagt.
 - Linear: Lineare Regression
 - Binär: Binär-logistische Regression
- Mehrere Prädiktoren möglich, ggf. auch Interaktionen
 - Zusammenwirken von Prädiktoren auf Kriterium
- Jeder Prädiktor/jede Interaktion bekommt einen Effektschätzer.
- Die Residuen sind die Vorhersagefehler.

Zusammenfassung

- Regressionen sind (im Gegensatz zu Korrelationen) **gerichtet**.
- Lineare Regressionsanalysen haben Voraussetzungen:
 - Linearität, Varianzhomogenität, Normalverteilung
- Die sog. Formel (*formula*) gibt, welche Zusammenhänge wir modellieren wollen.
 - ~ "predicted by"; + *single effect*; * *single effect* & Interaktion
- `lm()` / `glm(..., family = "binomial")`
- Nicht-lineare Regressionen: Modellierung nicht-linearer Zshge.

Übungen

Wolfer et al. (2016). The effectiveness of lexicographic tools for optimising written L1-texts. *International Journal of Lexicography* 31(1), 1-18



- Experimentelle Studie an der Uni Mannheim:
 - Effekt von lexikographischen Hilfsmitteln auf die Textüberarbeitung
 - Experimenteller Faktor 1: **Hilfsmittel** (nur Text/Markierung/Markierung+Hilfsmittel)
 - Experimenteller Faktor 2: **Thema des Texts** (Phraseologismen/Jugend)
- Laden Sie die Ergebnisdatei direkt aus dem Internet:
 - `readRDS(url("https://github.com/saschawo/Datasets/raw/main/wb_exp.Rds"))`
- Analysieren Sie den Einfluss der o.g. Faktoren (Bedingung, Text) darauf, ob ein Stolperstein verbessert wurde.
 - Beachten Sie auch die Interaktion der beiden Prädiktoren.
- Visualisieren Sie die Modellergebnisse und interpretieren Sie die Interaktion:
 - `install.packages("effects"); library(effects)`
 - `plot(allEffects(<Modell>), multiline = T, confint = list(style = "bars"))`

Begriffe



Intercept

Prädiktoren

Heteroskedastizität

Steigung / Slope

Kriteriumsvariable

Histogramm

Vorhersage

Interaktion

QQ-Plot

Residuen

Linearität

Binär-logistische R.

Kovariaten

Varianzhomogenität

Formel / *formula*

Multiple Regression

Normalverteilung

Nicht-lineare Regr.



Inferenzstatistik

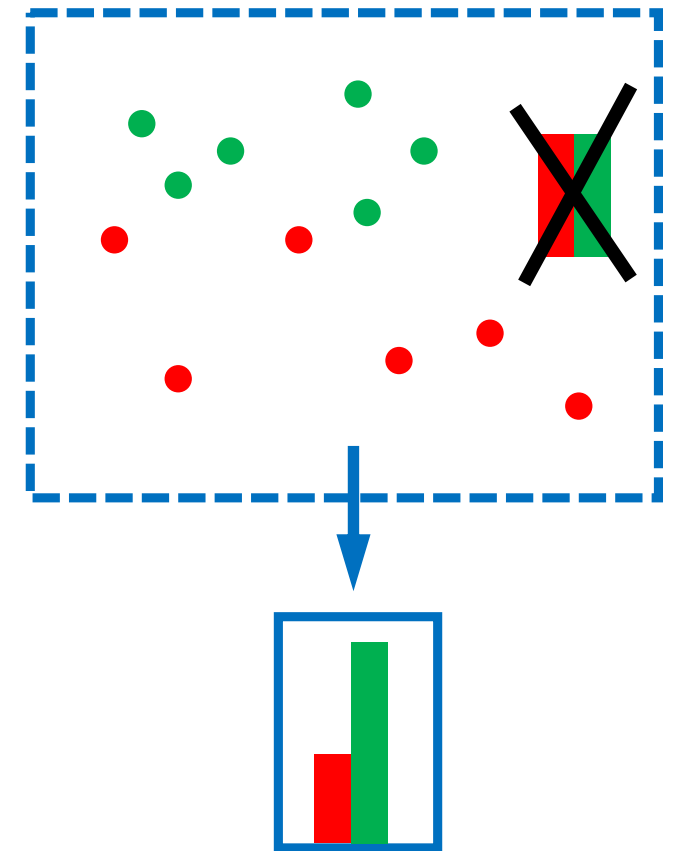
Inferenzstatistik

Man nennt dieses Vorgehen auch *null-hypothesis significance testing (NHST)*.

- Mit inferenzstatistischen Verfahren wollen wir von unserer **Stichprobe** auf die zugrundeliegende **Grundgesamtheit (Population)** verallgemeinern.
- Wir testen anhand unserer Stichprobe eine **Hypothese**, von der wir wissen möchten, ob sie in der Population gilt.
- Die Hypothese, die getestet wird, ist immer die **Nullhypothese**.
 - H_0 : Es gibt *keinen* Zusammenhang / Unterschied (= Effekt).
- Unsere Forschungshypothese ist stets die **Alternativhypothese**.
 - H_1 : Es gibt einen Zusammenhang / Unterschied.
- Wir wollen herausfinden, ob wir die Nullhypothese mit ausreichend großer Wahrscheinlichkeit **ablehnen** können.

Signifikanzniveau p

- Wir nehmen *hypothetisch* an, dass in der Grundgesamtheit die Nullhypothese gilt.
- Wir ziehen zufällig eine Stichprobe aus der Grundgesamtheit.
- p gibt uns an, wie wahrscheinlich die Ergebnisse in der Stichprobe sind, wenn in der Grundgesamtheit tatsächlich die Nullhypothese gelten sollte.
- Wenn p **klein genug** ist, lehnen wir die Nullhypothese ab und nehmen stattdessen die Alternativhypothese an.



Signifikanzniveau p

Permutationstests erlauben es, Effekte mit weniger Annahmen zu überprüfen. Siehe u.a. hier: <https://statologie.de/permutationstest-r/>

- "Klein genug": kleiner 5% ($p < 0,05$)
 - Andere Schwellenwerte: 0,01 (1%); 0,001 (0,1%)
- Unbedingte Voraussetzung: Zufällige Stichprobenziehung, ansonsten *sampling bias*
- p kann uns **nicht** dabei helfen, die Nullhypothese zu **bestätigen**.
 - Nicht-Effekte können also nicht interpretiert werden! Es ist somit in diesem Paradigma äußerst schwierig, Gleichheit oder Nicht-Zusammenhänge nachzuweisen.
- Alternativhypothese wird nie "bewiesen", wir können lediglich mit hoher Wahrscheinlichkeit die Nullhypothese ablehnen.

p-Werte in R & Effektgrößen

- Viele statistische Methoden geben *p*-Werte aus.
 - `summary()` von linearen Modellen
 - `cor.test()` statt `cor()`
- Statistische Signifikanz ist aber nicht alles. Auch (sehr) kleine Effekte können signifikant sein.
 - z. B. sehr kleine Unterschiede zwischen Gruppen, schwache Zusammenhänge
- Auch sehr große Effekte sind manchmal *nicht* signifikant ...
 - ... z. B., weil das assoziierte Rauschen sehr groß ist.
- 5% ist eine **arbiträre Grenze!**
 - Ist $p = 0,049$ wirklich veröffentlichungswert und $p = 0,051$ nicht?

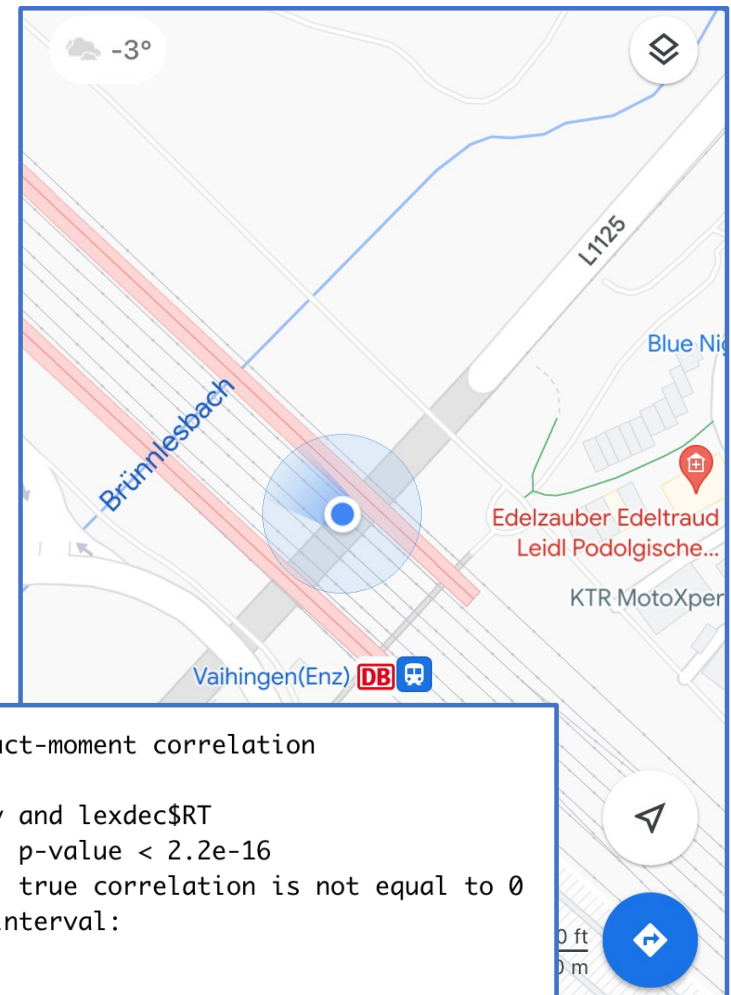
Konfidenzintervalle

- Wichtiges Konzept in der frequentistischen Inferenzstatistik
- Schätzen wir einen Parameter der Population mit einem bestimmten Verfahren, enthält ein bestimmter Anteil der Intervalle den wahren Wert.
 - 95%, 99%, 99,9%, ...
- Werfen wir wiederholt Ringe, um den echten Wert in der Population zu treffen, liegen 95%, 99%, ... der Ringe um den Wert.



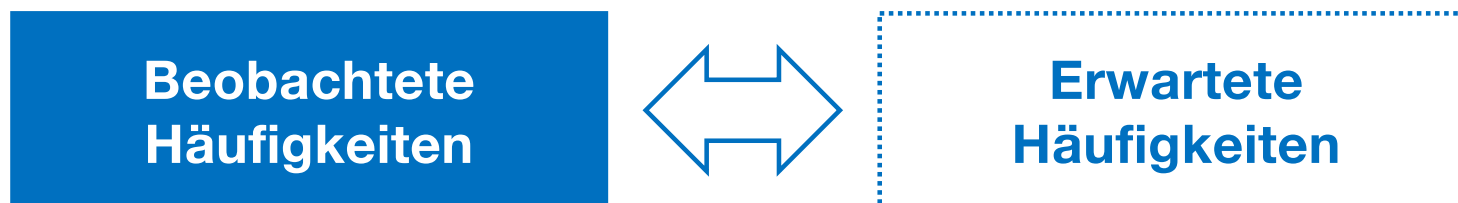
Konfidenzintervalle

- Konfidenzintervalle werden in R für eine Reihe von Tests ausgegeben.
- `confint(<Modell>)` gibt Konfidenzintervalle für die Effektschätzer aus.
 - `level = .95`



Chi-Quadrat-Test

- Methode zur Analyse von **häufigkeitsbasierten** Daten (Nominalskala).
- Grundidee: Wir **beobachten** bestimmte Häufigkeiten, und wir **erwarten** bestimmte Häufigkeiten unter der Nullhypothese.
- Wenn die beobachteten Häufigkeiten von den erwarteten Häufigkeiten abweichen, lehnen wir die Nullhypothese ab.



Chi-Quadrat-Test



- Beispiel: Eine Variable mit drei Ausprägungen
 - Wir wollen wissen, ob eine Farbe von Kugeln am zentralen Weihnachtsbaum der Uni Basel überrepräsentiert ist. Es gibt rote, blaue und grüne Kugeln. Insgesamt hängen **90 Kugeln** am Baum.
 - Was wären die **erwarteten Häufigkeiten** für jede der Farben unter der Nullhypothese?

30 rote Kugeln

30 blaue Kugeln

30 grüne Kugeln

41 rote Kugeln

31 blaue Kugeln

18 grüne Kugeln

Der Chi-Quadrat-Test testet, ob die beobachtete Verteilung der Kugeln von der erwarteten Verteilung abweicht.

Übung: Christbaumkugeln



- `chi <- chisq.test(c(41, 31, 18))` # erstellt Test
- `chi$expected` # gibt erwartete Hfgkten aus
- `chi` # gibt Testergebnis aus
- `residuals(chi)` # gibt Residuen aus

- Was schlussfolgern Sie aus den Ergebnissen?



Kreuztabellen

- Häufiger wird der Chi-Quadrat-Test bei **Kreuztabellen** eingesetzt.
 - Kreuztabellen setzen zwei nominalskalierte Variablen zueinander in Beziehung. In R: `table(<Vektor 1>, <Vektor 2>)`
- In Kreuztabellen ergeben sich die erwarteten Häufigkeiten über die Verrechnung der **Zeilen-** und **Spaltensummen** (Randsummen, *marginal sums*).
 - Berücksichtigung der sog. **Grundwahrscheinlichkeiten**.

$$f_{E(j,k)} = \frac{\text{Zeilensumme}(j) \cdot \text{Spaltensumme}(k)}{n}$$

Übung: Dativalternation



```
> library(languageR)
> tab <- table(dative$Modality, dative$RealizationOfRecipient)
> tab
> chi <- chisq.test(tab)
> chi$expected
> chi
> residuals(chi)
> mosaicplot(tab, shade = T)
```

Chi-Quadrat-Test: Achtung!

- Der Chi-Quadrat-Test reagiert extrem sensitiv auf die Größe der Stichprobe – auch wenn die Verhältnisse exakt gleich bleiben.
- Berechnen Sie daher auch die **Effektstärke** Φ / V :
 - > `library(psych)`
 - > `phi(tab)`
- Φ für 2-mal-2-Tabellen ("Vierfeldertabellen"), V für größere Tabellen

SA

Erkläre den Chi-Quadrat-Test in einfachen Worten.



Der Chi-Quadrat-Test ist ein statistischer Test, um zu prüfen, ob es einen signifikanten Unterschied zwischen den beobachteten Häufigkeiten in einer Zufallsvariablen gibt. Die beobachteten Häufigkeiten werden mit den erwarteten Häufigkeiten verglichen, die mit einem Chi-Quadrat-Verteilungstabelle berechnet werden. Ein signifikanter Unterschied zwischen den beiden signifikant sein kann, wenn die beobachteten Häufigkeiten von den erwarteten abweichen. Um Unterschiede in jeder Art von Zählungsdaten in Kategorien unterteilt werden können, kann der Chi-Quadrat-Test verwendet werden. Jedoch nicht automatisch, dass es eine bedeutende Abweichung gibt. Weitere Analysen sind erforderlich, um

SA

Okay, danke. Aber jetzt mach daraus eine süße Geschichte.



Einmal, in einem kleinen Dorf, gab es eine Gruppe von Bären, die wissen wollten, ob es einen Unterschied zwischen ihren Lieblingsfarben gab. Einer der Bären, der sehr gut im Rechnen war, hatte von einem magischen Test gehört, der ihnen helfen konnte, herauszufinden, ob es einen signifikanten Unterschied gab. Der Test hieß der Chi-Quadrat-Test.



Der Bär sammelte die Lieblingsfarben aller Bären im Dorf und berechnete die Summe der Quadrate der Abweichungen der tatsächlichen Häufigkeiten von den erwarteten Häufigkeiten. Dann verglich er die Zahl mit einer magischen Tabelle, die ihm sagte, ob der Unterschied zwischen den beiden signifikant war.

Zu ihrer Freude stellten die Bären fest, dass es tatsächlich einen signifikanten Unterschied zwischen ihren Lieblingsfarben gab! Sie feierten ihren Sieg mit einem großen Fest und lernten viel über die Macht des Chi-Quadrat-Tests. Sie beschlossen, ihn immer wieder zu verwenden, um herauszufinden, ob es Unterschiede in ihrem Dorf gab. Sie waren sehr glücklich und zufrieden und lebten fortan in Frieden und Harmonie.

chat.openai.com

SA

Der Beispieldatensatz `mtcars` ist in der Standardinstallation von R enthalten.

- Zu wie vielen Autos finden Sie Informationen in `mtcars`?
- Wie viele Variablen/Spalten sind in `mtcars` enthalten?
- Lassen Sie sich alle Autos aus `mtcars` ausgeben, die mehr als `hp = horse power` haben.
- Wie viele Autos in `mtcars` haben fünf Gänge (`gear`) und können Meilen pro Gallone Benzin (`mpg = miles per gallon`) fahren?
- Welches Auto in `mtcars` ist das sparsamste (im Sinne von Benzinverbrauch)?



- Es gibt Informationen zu 32 Autos in `mtcars`.
- `mtcars` enthält 11 Variablen/Spalten.
- Um alle Autos aus `mtcars` auszugeben, die mehr als 250 PS haben, können Sie den folgenden R-Code verwenden:

Copy code

```
mtcars[mtcars$hp > 250, ]
```

- Um herauszufinden, wie viele Autos in `mtcars` fünf Gänge haben und mehr als 30 Meilen pro Gallone fahren können, können Sie den folgenden R-Code verwenden:

Copy code

```
nrow(mtcars[mtcars$gear == 5 & mtcars$mpg > 30, ])
```

Dieser Code gibt die Anzahl der Zeilen/Autos zurück, die die angegebenen Kriterien erfüllen.

- Um das sparsamste Auto in `mtcars` zu finden, können Sie den folgenden R-Code verwenden:

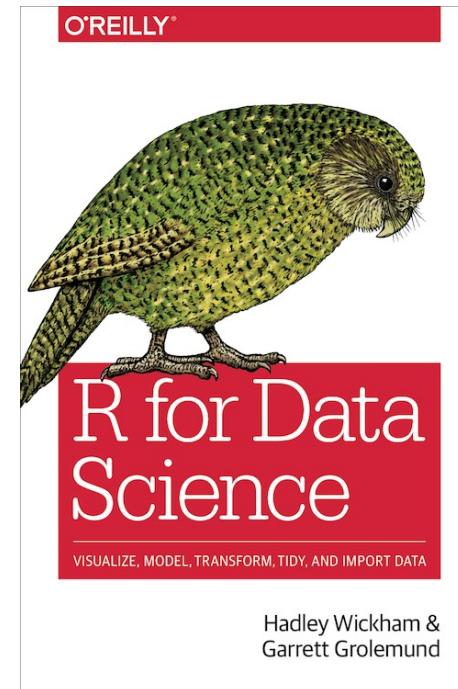
Copy code

```
mtcars[mtcars$mpg == min(mtcars$mpg), ]
```

Dieser Code gibt alle Autos aus, die den niedrigsten Benzinverbrauch haben. Wenn es mehrere Autos mit dem gleichen niedrigsten Benzinverbrauch gibt, werden alle diese Autos ausgegeben.

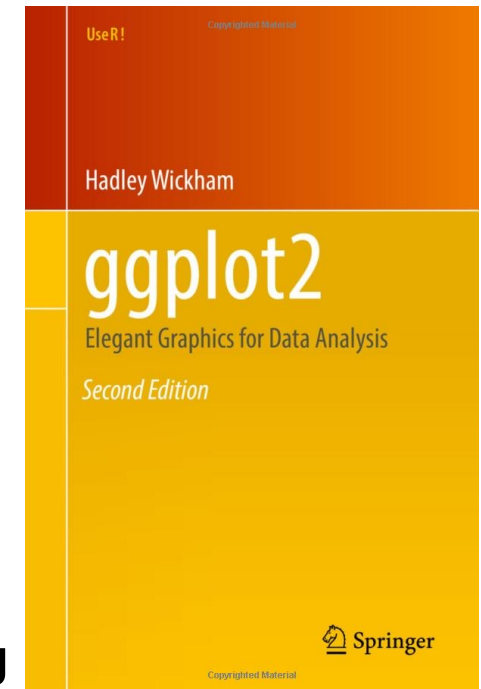
Hier geht's weiter...

- Fortgeschrittene Regressionsverfahren: Gemischte Modelle
- Explorative Verfahren: Clusteranalysen, Hauptkomponentenanalyse, Multidimensionale Skalierung
- Visualisierungen mit dem Paket `{ggplot2}`
- Datenmanipulation mit dem Paket `{dplyr}`
- ...



<https://ggplot2-book.org>

<https://r4ds.had.co.nz>



```
install.packages("christmas")  
library(christmas)  
xmassnowman(year = 2023,  
             language = "english")
```



 @sascha_wolfer

 @sascha_wolfer@fediscience.org