

# Folien für das Modul ‘Praxis der Datenanalyse’

WS17



```
#library(mosaic)
library(tidyverse)
library(formatR)
library(knitr)
library(gridExtra)
```

# Vorwort

- Dieser Foliensatz dient zur Unterstützung des Unterrichts im Rahmen des Moduls 'Praxis der Datenanalyse' an der FOM Hochschule.
- Dieser Foliensatz ist *nicht* zur Vermittlung des Stoffes gedacht (auch nicht zum Nachbereiten der Stoffes); dazu gibt es ein **Skript**. Er soll zur Unterstützung der Vermittlung des Stoffes im Unterricht helfen.
- Autoren: Sebastian Sauer schrieb den Hauptteil dieses Buchs. Oliver Gansser schrieb das Kapitel zur Dimensionsreduktion. Karsten Lübke schrieb den Großteil des Kapitels zur Regression und zur Clusteranalyse sowie Teile des Kapitels 'Rahmen'. Matthias Gehrke schrieb den Großteil des Kapitels zur logistischen Regression. Norman Markgraf schrieb die Vorlage für Folien.

# Danke

- Norman Markgraf hat umfangreich Fehler gejagt und Verbesserungen angemahnt vorgenommen. Der Austausch mit den ifes-Mitgliedern hielt die Flamme am Köcheln. Eine Reihe weiterer Kollegen standen mit Rat und Tat zur Seite. Die Hochschulleitung sowie das Dekanat für Wirtschaftspsychologie hat dieses Projekt unterstützt. Die Abteilung Medienentwicklung der FOM hat bei Fragen rund um die Veröffentlichung geholfen. Last but not least: Viele Studierenden wiesen auf Inkonsistenzen, Fehler und Unklarheiten hin. Ihnen allen: Vielen Dank!

# Organisatorisches

# Modulziele

Die Studierenden können nach erfolgreichem Abschluss des Moduls:

- den Ablauf eines Projekts aus der Datenanalyse in wesentlichen Schritten nachvollziehen,
- Daten aufbereiten und ansprechend visualisieren,
- Inferenzstatistik anwenden und kritisch hinterfragen,
- klassische Vorhersagemethoden (Regression) anwenden,
- moderne Methoden der angewandten Datenanalyse anwenden (z.B. Textmining),
- betriebswirtschaftliche Fragestellungen mittels datengetriebener Vorhersagemodellen beantworten.

# Themen pro Termin (insgesamt 44UE Lehre)

Termin	Thema / Kapitel
1	Organisatorisches
1	Einführung
1	Rahmen
1	Daten einlesen
2	Datenjudo
3	Daten visualisieren
4	Fallstudie (z.B. zu 'movies')
5	Daten modellieren
5	Der p-Wert
6	Lineare Regression - metrisch
7	Lineare Regression - kategorial
8	Fallstudie (z.B. zu 'titanic' und 'affairs')
9	Vertiefung 1: Textmining oder Clusteranalyse
10	Vertiefung 2: Dimensionsreduktion
11	Wiederholung

# Prüfung - Allgemeine Hinweise

- Die Prüfung besteht aus zwei Teilen
  - einer Klausur (50% der Teilnote)
  - einer Datenanalyse (50% der Teilnote).

*Prüfungsrelevant* ist der gesamte Stoff aus dem Skript und dem Unterricht mit folgenden Ausnahmen:

- Inhalte/Abschnitte, die als "nicht klausurrelevant" gekennzeichnet sind,
- Inhalte/Abschnitte, die als "Vertiefung" gekennzeichnet sind,
- Fallstudien (nur für Klausuren nicht prüfungsrelevant),
- die Inhalte von Links,
- die Inhalte von Fußnoten,
- die Kapitel *Vorwort*, *Organisatorisches* und *Anhang*.

Alle Hinweise zur Prüfung gelten nur insoweit nicht anders vom Dozenten festgelegt.

# Klausur und Datenanalyse

## Klausur

- Hinweise zur Klausur finden Sie hier
- Im Unterricht findet eine Probeklausur statt.
- Lernaufgaben finden sich im Skript.

## Datenanalyse

- Hinweise zur Datenanalyse finden Sie hier.
- Die Datenanalyse wird (in fast jeder Stunde) praktisch eingeübt.
- Beispiele für gute Datenanalysen von Studierenden finden Sie hier.

# Rahmen

# Lernziele

- Einen Überblick über die fünf wesentliche Schritte der Datenanalyse gewinnen.
- R und RStudio installieren können.
- Einige häufige technische Probleme zu lösen wissen.
- R-Pakete installieren können.
- Einige grundlegende R-Funktionalitäten verstehen.
- Auf die Frage “Was ist Statistik?” eine Antwort geben können.

# Prozess der Datenanalyse - Überblick über das Modul

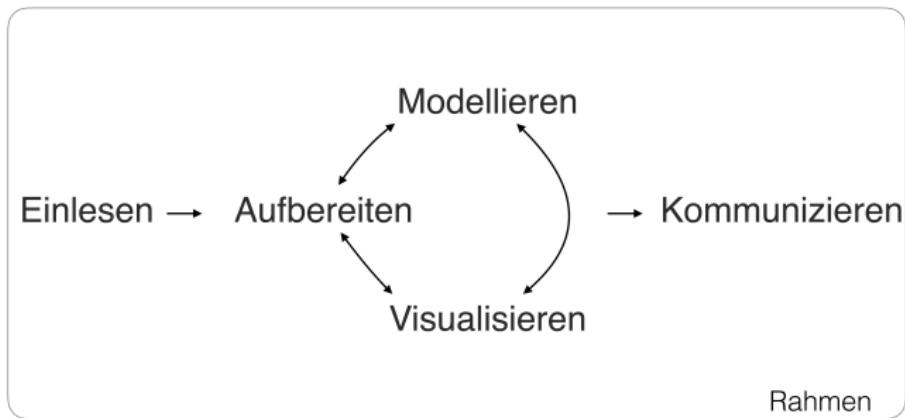


Abbildung 1: Der Prozess der Datenanalyse

# R und RStudio installieren

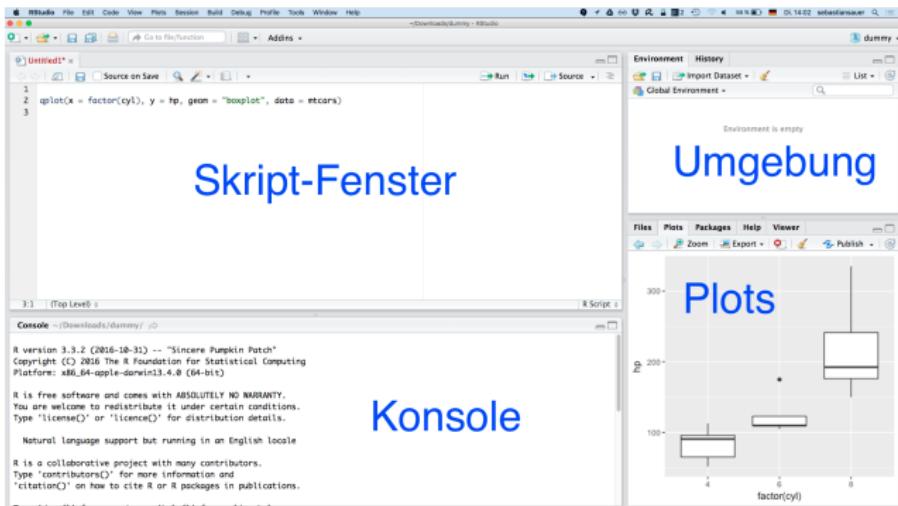


Abbildung 2: RStudio

# Hilfe! R!

Beliebte Fehler:

- `install.packages(dplyr)`
- `install.packages("dliar")`
- `install.packages("derpyler")`
- `install.packages("dplyr") # dependencies vergessen`
- Keine Internet-Verbindung
- `library(dplyr) # ohne vorher zu installieren`

# Pakete installieren

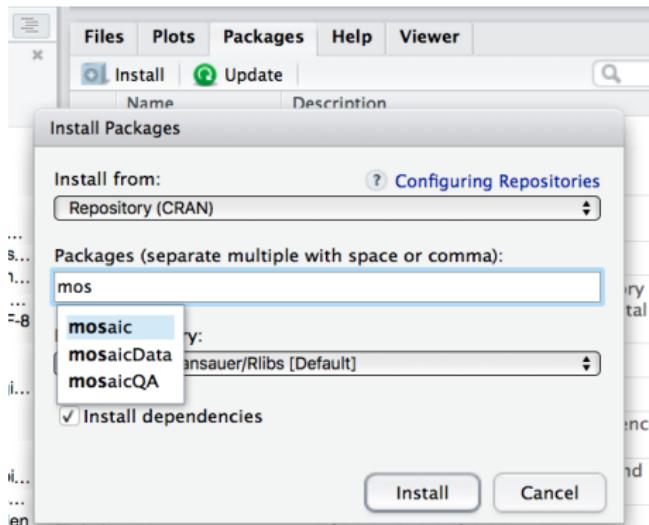


Abbildung 3: So installiert man Pakete in RStudio

# Aufgaben

1. Öffnen Sie das Cheatsheet für RStudio und machen Sie sich mit dem Cheatsheet vertraut.
2. Sichten Sie kurz die übrigen Cheatsheets; später werden die Ihnen vielleicht von Nutzen sein.

# Aufgaben

3. Führen Sie diese Syntax aus:

```
meine_coole_variable <- 10  
meine_coole_var1able
```

Woher röhrt der Fehler?

4. Korrigieren Sie die Syntax:

```
install.packages(dplyer)
```

# Aufgaben

```
y <- Hallo R!
```

```
Hallo R <- 1
```

```
Hallo_R <- 1
```

# Was ist Statistik?

Eine Antwort dazu ist, dass Statistik die Wissenschaft von Sammlung, Analyse, Interpretation und Kommunikation von Daten ist mithilfe mathematischer Verfahren ist und zur Entscheidungshilfe beitragen solle.

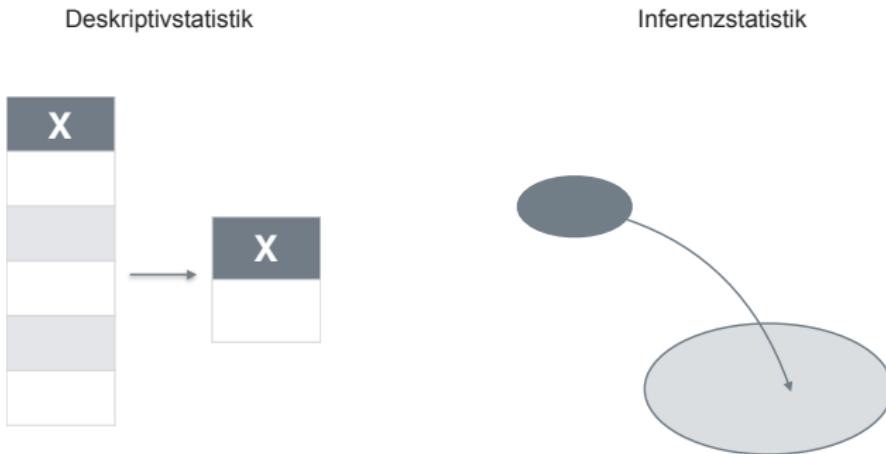


Abbildung 4: Sinnbild für die Deskriktiv- und die Inferenzstatistik

# Abduktion als klassische Denkfigur in der Statistik

Prämissen 1: Wenn Modell M wahr ist, dann sollten die Daten das

Prämissen 2: Die Daten weisen das Muster D auf.

---

Konklusion: Daher muss das Modell M wahr sein.

Die Konklusion ist *nicht* zwangsläufig richtig.

# Befehlsübersicht für das Kapitel ‘Rahmen’

Tabelle 2: Befehle des Kapitels ‘Rahmen’

Paket::Funktion	Beschreibung
install.packages("x")	Installiert Paket “x” (nicht: Paket “X”)
library	lädt ein Paket
<-	Weist einer Variablen einen Wert zu
c	erstellt eine Spalte/ einen Vektor

# Daten einlesen

## Lernziele

- Wissen, was eine CSV-Datei ist.
- Wissen, was UTF-8 bedeutet.
- Erläutern können, was R unter dem “working directory” versteht.
- Erkennen können, ob eine Tabelle in Normalform vorliegt.
- Daten aus R hinauskriegen (exportieren).

Dieses Kapitel beantwortet eine Frage: “Wie kriege ich Daten in vernünftiger Form in R hinein?“.

# Prozess der Datenanalyse – Einlesen

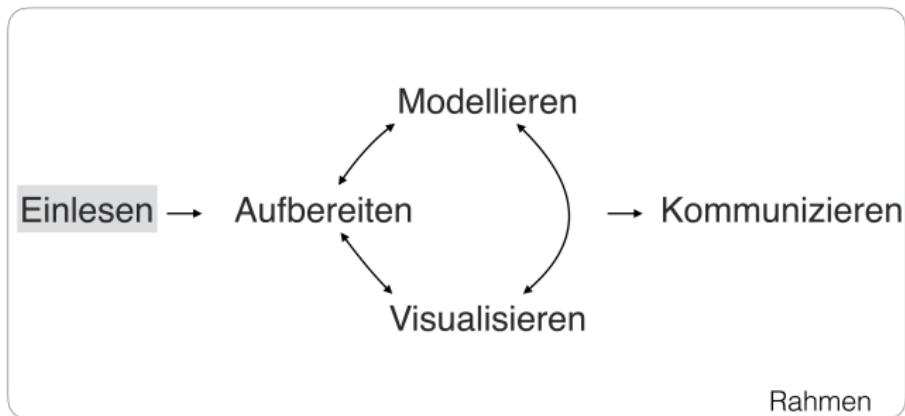


Abbildung 5: Daten sauber einlesen

# Daten (CSV, XLS,...) mit RStudio importieren

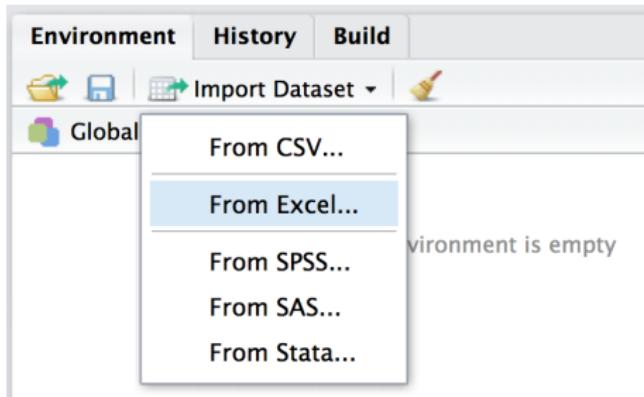


Abbildung 6: Daten einlesen (importieren) mit RStudio

# CSV-Dateien sind einer der wichtigsten Daten-Formate

```
row_number,date_time,study_time,self_eval,interest,score
1,05.01.2017 13:57:01,5,8,5,29
2,05.01.2017 21:07:56,3,7,3,29
3,05.01.2017 23:33:47,5,10,6,40
4,06.01.2017 09:58:05,2,3,2,18
5,06.01.2017 14:13:08,4,8,6,34
6,06.01.2017 14:21:18,NA,NA,NA,39
```

# Das Arbeitsverzeichnis mit RStudio wählen

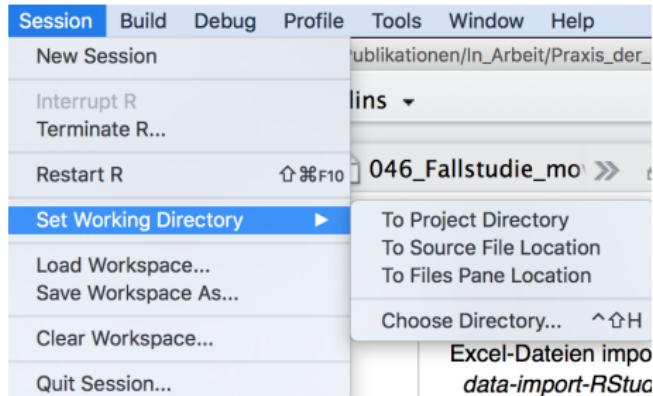


Abbildung 7: Das Arbeitsverzeichnis mit RStudio auswählen

# Normalform einer Tabelle

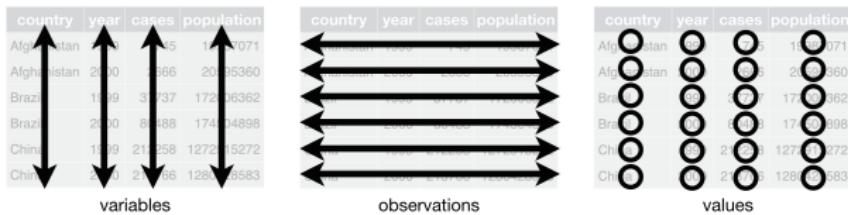


Abbildung 8: Schematische Darstellung eines Dataframes in Normalform

# Breit vs. Lang

Breit

ID	Q1	Q2	Q3	Q4
1	123	342	431	675
2	324	342	234	345
3	343	124	456	465
...				

Lang

ID	Quartal	Umsatz
1	Q1	342
2	Q2	342
3	...	124
...	Q1	342
	Q2	342
	Q3	124
	...	



Abbildung 9: Dieselben Daten - einmal breit, einmal lang

# Ein Dataframe in Normalform - Beispiel

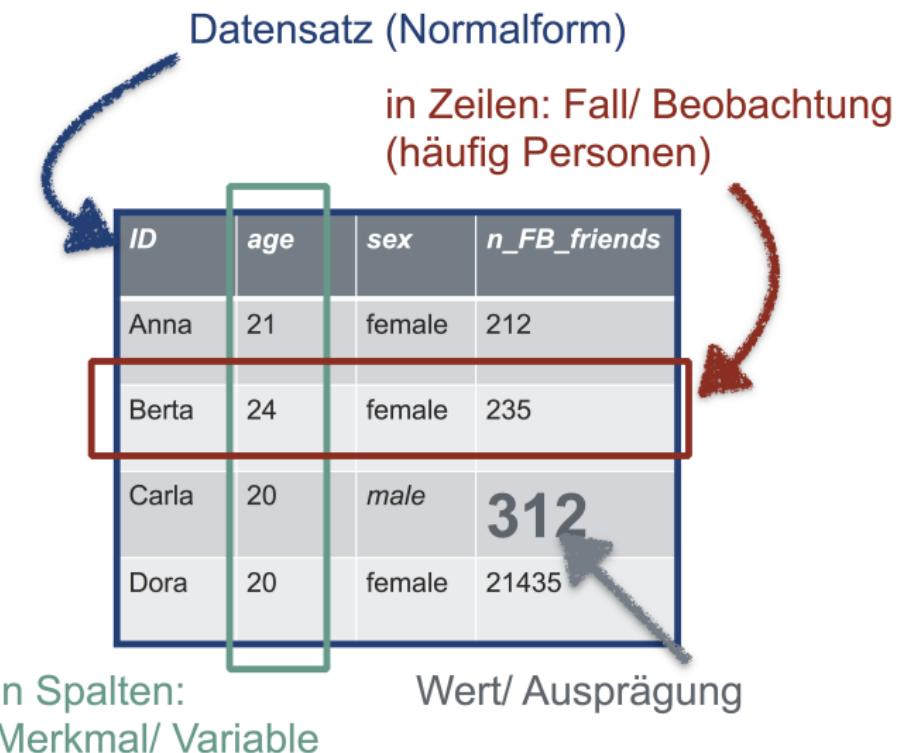


Abbildung 10: Folien für das Modul 'Praxis der Datenanalyse' WS17

# Tabelle in Normalform bringen

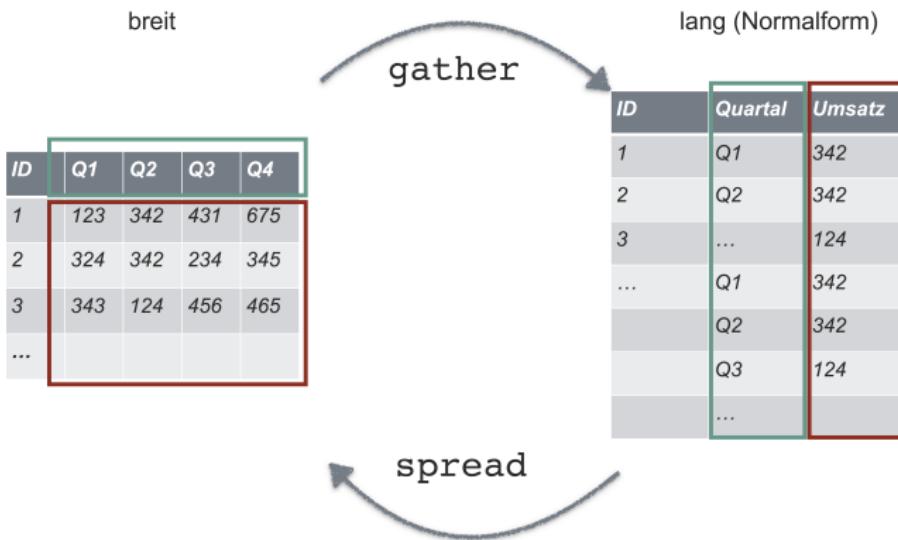


Abbildung 11: Mit 'gather' und 'spread' wechselt man von der breiten Form zur langen Form

# Beispiel für die Normalisierung einer Tabelle

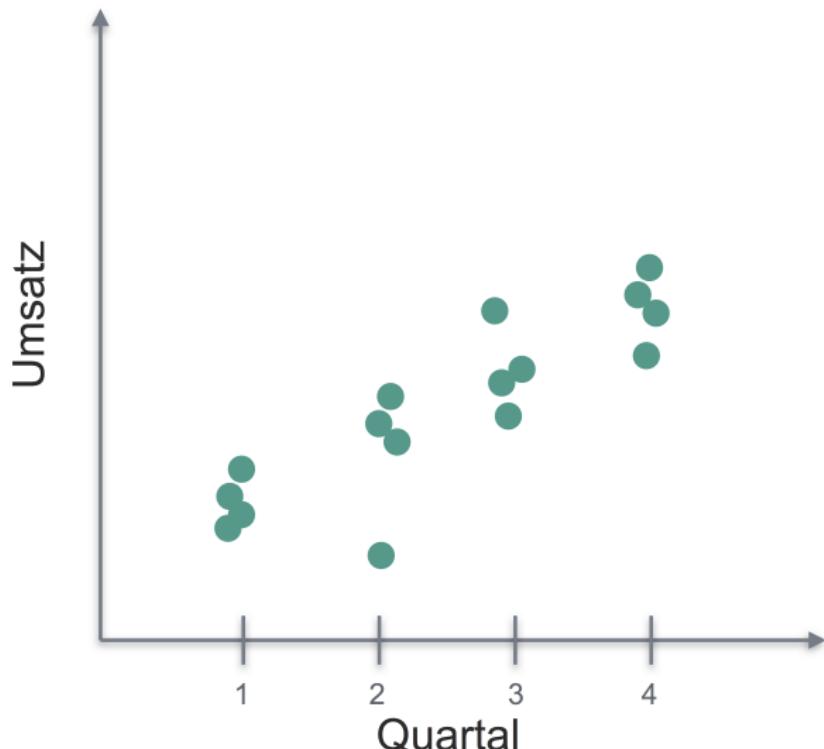


Abbildung 12: Ein Beispiel für eine Abbildung in einer Normalform Tabelle

## gather und spread

```
library(tidyr)

df_lang <- gather(df_breit, key = "Quartal", value = "Umsatz")

df_breit <- spread(df_lang, Quartal, Umsatz)

df_lang <- gather(df_breit, key = "Quartal", value = "Umsatz")
```

# Textkodierung und Daten exportieren

*Speichern Sie R-Textdateien wie Skripte stets mit UTF-8-Kodierung ab.*

```
write.csv(name_der_tabelle, "Dateiname.csv")
```

# Befehlsübersicht für das Kapitel ‘Daten einlesen’

Tabelle 3: Befehle des Kapitels ‘Daten einlesen’

Paket::Funktion	Beschreibung
read.csv	Liest eine CSV-Datei ein.
write.csv	Schreibt einen Dateframe in eine CSV-Datei.
tidyverse::gather	Macht aus einem “breiten” Dataframe einen “langen”.
tidyverse::separate	“Zieht” Spalten auseinander.

# Aufgaben<sup>1</sup>

1. In CSV-Dateien dürfen Spalten *nie* durch Komma getrennt sein.
2. RStudio bietet die Möglichkeit, CSV-Dateien per Klick zu importieren.
3. RStudio bietet *nicht* die Möglichkeit, CSV-Dateien per Klick zu importieren.
4. "Deutsche" CSV-Dateien verwenden als Spalten-Trennzeichen einen Strichpunkt.
5. In einer Tabelle in Normalform stehen in jeder Zeile eine Beobachtung.
6. In einer Tabelle in Normalform stehen in jeder Spalte eine Variable.
7. R stellt fehlende Werte mit einem Fragezeichen ? dar.
8. Um Excel-Dateien zu importieren, kann man den Befehl `read.csv` verwenden.

---

<sup>1</sup>F, R, F, R, R, R, F, F

# Datenjudo

# Lernziele für das Kapitel ‘Datenjudo’

- Die zentralen Ideen der Datenanalyse mit dplyr verstehen.
- Typische Probleme der Datenanalyse schildern können.
- Zentrale dplyr-Befehle anwenden können.
- dplyr-Befehle kombinieren können.
- Die Pfeife anwenden können.
- Werte umkodieren und “binnen” können.

# Prozess der Datenanalyse – Datenjudo

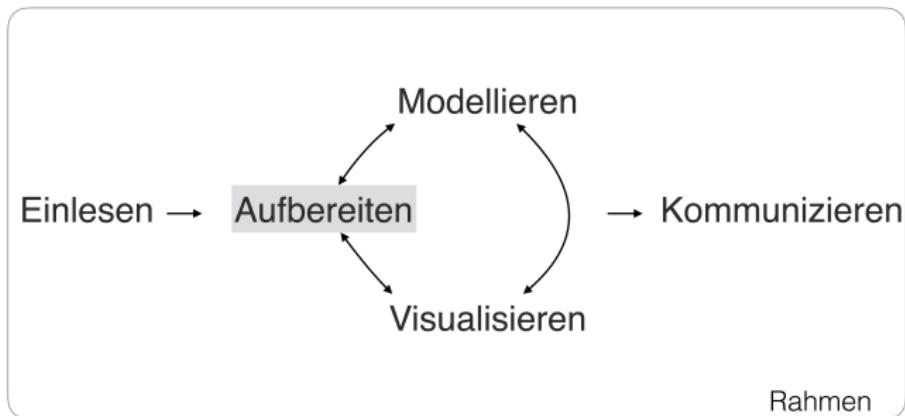


Abbildung 13: Daten aufbereiten

# Typische Probleme bei der Datenaufbereitung

Typische Probleme, die immer wieder auftreten, sind:

- *Fehlende Werte*
- *Unerwartete Daten*
- *Daten müssen umgeformt werden*
- *Neue Variablen (Spalten) berechnen:*
- ...

# Daten aufbereiten mit dplyr



Abbildung 14: Lego-Prinzip: Zerlege eine komplexe Struktur in einfache Bausteine

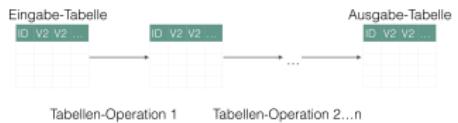


Abbildung 15: Durchpfeifen: Ein Dataframe wird von Operation zu Operation weitergereicht

## Zeilen filtern mit filter



ID	Name	Note1
1	Anna	1
2	Anna	1
3	Berta	2
4	Carla	2
5	Carla	2

ID	Name	Note1
1	Anna	1
2	Anna	1

Abbildung 16: Zeilen filtern

# Spalten wählen mit select

vorher					nachher		
ID	Name	N1	N2	N3	ID	Name	N1
1	Anna	1	2	3	1	Anna	1
2	Berta	1	1	1	2	Berta	1
3	Carla	2	3	4	3	Carla	2
...	...	...	...	...	...	...	...

Abbildung 17: Spalten auswählen

## Zeilen sortieren mit arrange

ID	Name	Note1
1	Anna	1
2	Anna	5
3	Berta	2
4	Carla	4
5	Carla	3

→

ID	Name	Note1
1	Anna	1
3	Berta	2
5	Carla	3
4	Carla	4
2	Anna	5

Gute Noten zuerst!

Abbildung 18: Spalten sortieren

# Datensatz gruppieren mit group\_by

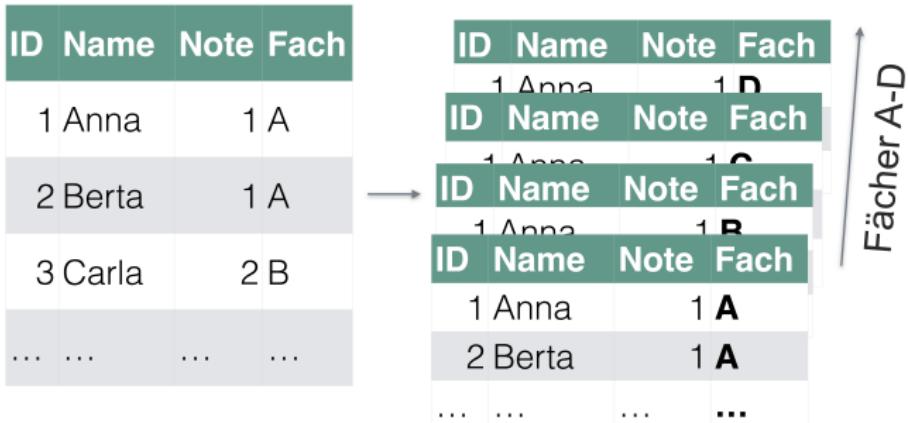


Abbildung 19: Datensätze nach Subgruppen aufteilen

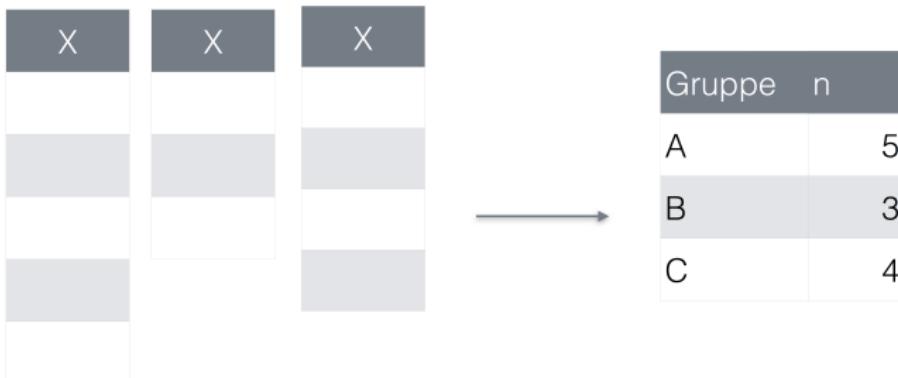
# Eine Spalte zusammenfassen mit summarise



Abbildung 20: Spalten zu einer Zahl zusammenfassen

# Zeilen zählen mit n und count

Gruppe A Gruppe B Gruppe C



5            3            4

Abbildung 21: Sinnbild für 'count'

# Die Pfeife



Abbildung 22: Das ist keine Pfeife

# Befehle hintereinander reihen mit der Pfeife

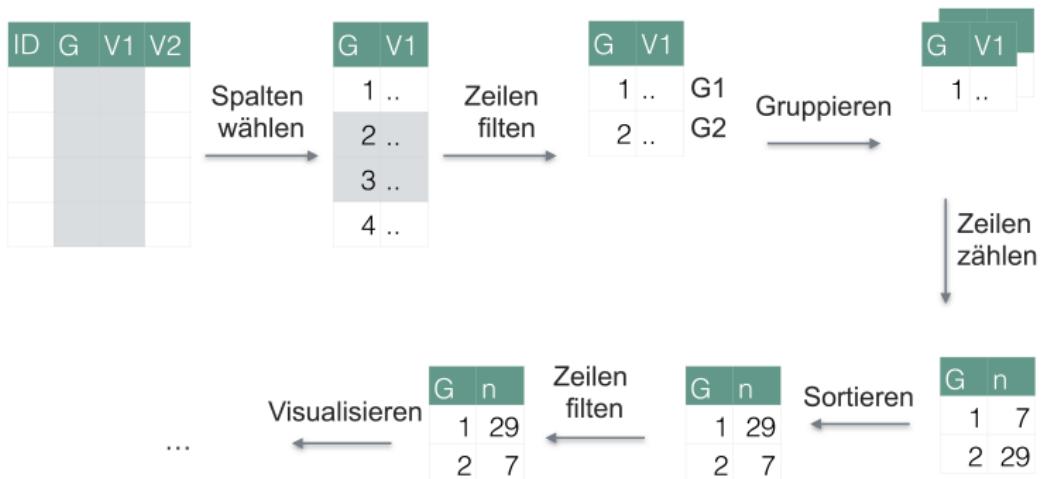


Abbildung 23: Das 'Durchpeifen'

# Introducing Pipe-Syntax

Vergleichen Sie mal diese Syntax

```
filter(summarise(group_by(filter(stats_test,
    !is.na(score)), interest), mw = mean(score)), mw > 30)
```

mit dieser

```
stats_test %>%
  filter(!is.na(score)) %>%
  group_by(interest) %>%
  summarise(mw = mean(score)) %>%
  filter(mw > 30)
```

# Pfeifen macht das Leben leichter

Tipp: In RStudio gibt es einen Shortcut für die Pfeife: Strg-Shift-M (auf allen Betriebssystemen).

Die Syntax von oben auf Deutsch:

- Nimm die Tabelle “stats\_test” UND DANN
- filtere alle nicht-fehlenden Werte UND DANN
- gruppiere die verbleibenden Werte nach “interest” UND DANN
- bilde den Mittelwert (pro Gruppe) für “score” UND DANN
- liefere nur die Werte größer als 30 zurück.

# Spalten berechnen mit `mutate`

## Sinnbild

The diagram illustrates the use of the `mutate` function to calculate a new column, `MW`, which represents the average of columns `N1`, `N2`, and `N3`. An arrow points from the original data frame on the left to the modified data frame on the right.

ID	N1	N2	N3	
1	1	2	3	
2	1	1	1	
3	2	3	4	
...	...	...	...	

→

ID	N1	N2	N3	MW
1	1	2	3	2
2	1	1	1	1
3	2	3	4	3
...	...	...	...	...

Will Durchschnittsnote pro Student wissen!

Abbildung 24: Sinnbild für `mutate`

# Beispiel für mutate

```
stats_test %>%  
  mutate(Streber = score > 38) %>%  
  head()
```

# Deskriptive Statistik mit dplyr

```
stats_test2 <- select(stats_test, -date_time)  
desctable(stats_test2)
```

# Befehlsübersicht

Tabelle @ref(tab:befehle-datenjudo) fasst die R-Funktionen dieses Kapitels zusammen.

Tabelle 4: Befehle des Kapitels 'Datenjudo'

Paket::Funktion	Beschreibung
dplyr::arrange	Sortiert Spalten
dplyr::filter	Filtert Zeilen
dplyr::select	Wählt Spalten
dplyr::group_by	gruppert einen Dataframe
dplyr::n	zählt Zeilen
dplyr::count	zählt Zeilen nach Untergruppen
%>% (dplyr)	verkettet Befehle
dplyr::mutate	erzeugt/berechnet Spalten
desctable::desctable	Liefert Tabelle mit deskriptiver Statistik zurück

# Daten visualisieren

# Lernziele für das Kapitel ‘Daten visualisieren’

- An einem Beispiel erläutern können, warum/ wann ein Bild mehr sagt, als 1000 Worte.
- Häufige Arten von Diagrammen erstellen können.
- Diagramme bestimmten Zwecken zuordnen können.

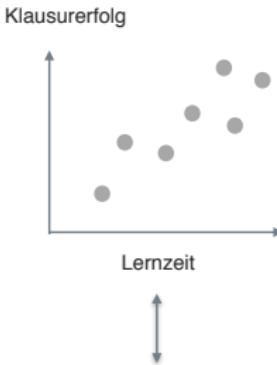
Statistik ist wie ein Bikini . . .

Dinosaurier-Video

# Die Anatomie eines Diagramms

Man kann folgende Bestandteile (“Anatomie”) eines Diagramms unterscheiden:

- Daten
- Abbildende Aspekte (Achsen, Farben, …)
- Geome (statistische Bilder wie Punkte, Linien, Boxplots, …)



# Beispiel für ein Diagramm mit ggplot2::qplot

```
qplot(x = year, y = budget, geom = "point", data = movies)
```

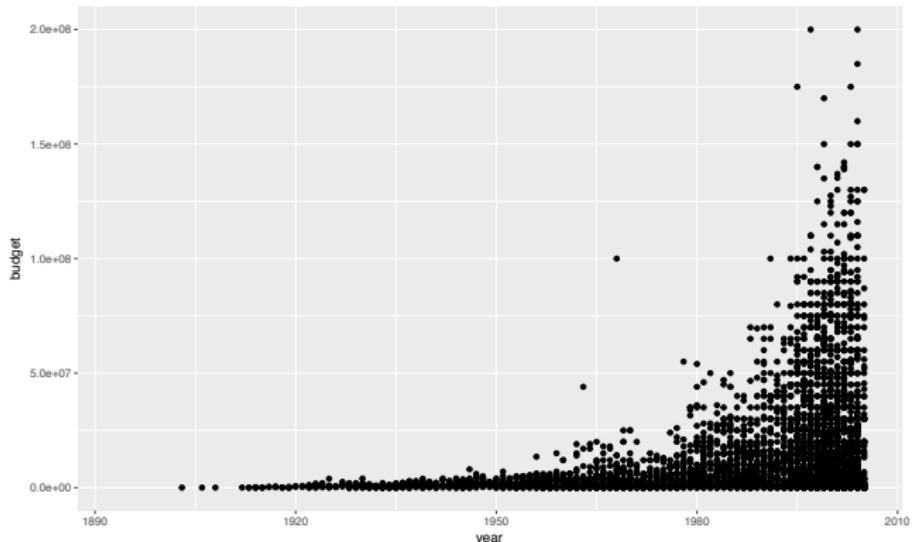


Abbildung 26: Mittleres Budget pro Jahr

# Anatomiestunde mit qplot

- qplot: Erstelle schnell (q wie quick in qplot) mal einen Plot (engl. “plot”: Diagramm).
- x: Der X-Achse soll die Variable “year” zugeordnet werden.
- y: Der Y-Achse soll die Variable “budget” zugeordnet werden.
- geom: (“geometrisches Objekt”) Gemalt werden sollen Punkte und zwar pro Beobachtung (hier: Film) ein Punkt; nicht etwa Linien oder Boxplots.
- data: Als Datensatz bitte `movies` verwenden.

# Syntax-Blaupause für qplot

Diese Syntax des letzten Beispiels ist recht einfach, nämlich:

```
qplot (x = X_Achse, y = Y_Achse, data = mein_dataframe, geom =
```

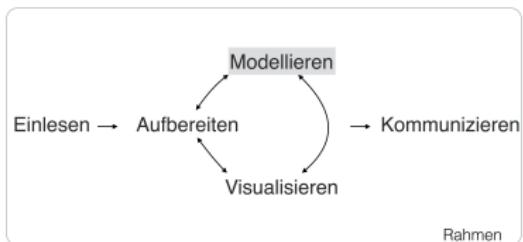
# Häufige Diagrammtypen

Tabelle 5: Häufige Diagramme

X-Achse	Y-Achse	Diagrammtyp
kontinuierliche Variable	-	Histogramm, Dichtediagramm
kontinuierliche Variable	kontinuierliche Variable	Punkte, Schachbrett-Diagramm
nominale Variable	-	Balkendiagramm
nominale Variable	nominale Variable	Mosaicplot (Fliesen-Diagramm)
nominale Variable	metrische Variable	Punktediagramm für Zusammensetzung
NA	NA	NA

# Grundlagen des Modellierens

# Prozess der Datenanalyse - Modellieren



# Was ist ein Modell



Abbildung 27: Modell eines VW-Käfers

# Die Beziehung von Gegenstandsbereich und Modell

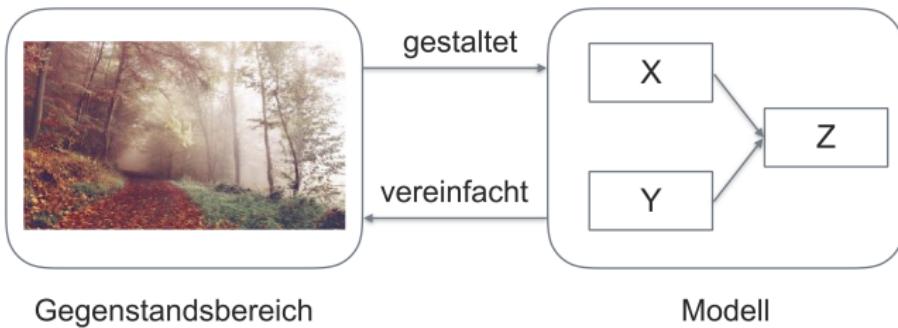


Abbildung 28: Modellieren

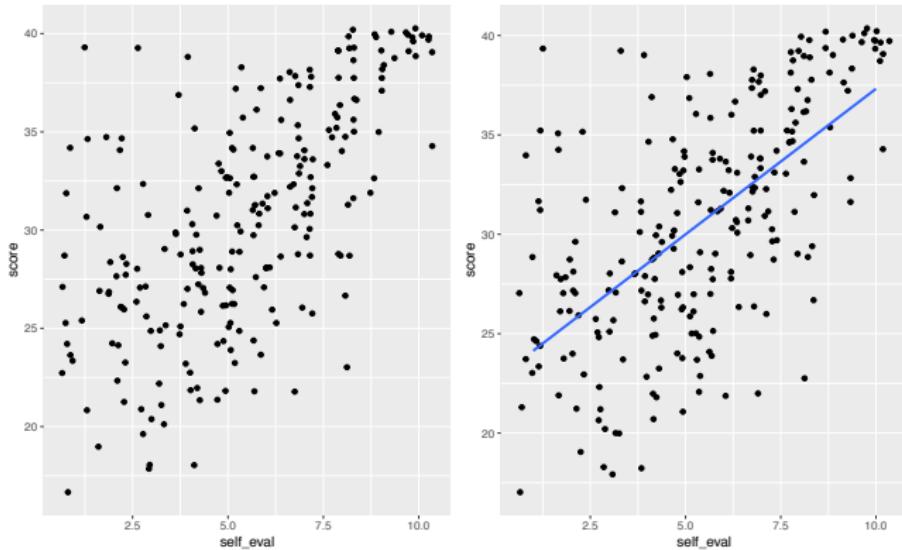
# Modelle spiegeln empirische Relationen in numerischen Relationen

*Modellieren bedeutet ein Verfahren zu erstellen, welches empirische Sachverhalte adäquat in numerische Sachverhalte umsetzt.*



Abbildung 29: Formaleres Modell des Modellierens

# Ein Beispiel zum Modellieren aus der Datenanalyse



Die blaue Gerade ist ein Modell für den Datensatz (sie versucht es zumindest).

# Modelle umfassen drei Aspekte

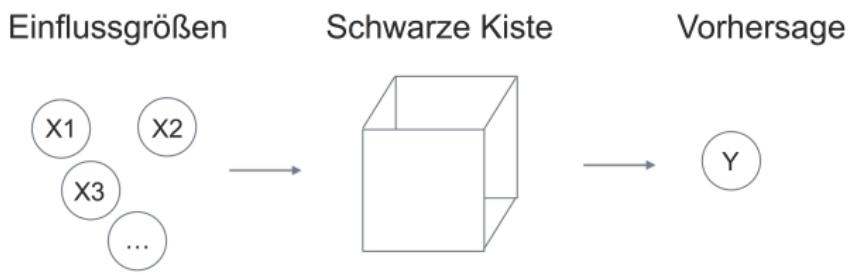


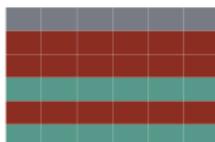
Abbildung 30: Modelle mit schwarzer Kiste

# Taxonomie der Ziele des Modellierens

- Geleitetes Modellieren
  - Prädiktives Modellieren
  - Explikatives Modellieren
- Ungeleitetes Modellieren
  - Dimensionsreduzierendes Modellieren
  - Fallreduzierendes Modellieren

# Veranschaulichung der beiden Arten des Modellierens

Fallreduzierendes  
Modellieren



Dimensionsreduzierendes  
Modellieren

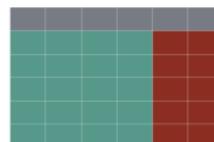


Abbildung 31: Die zwei Arten des ungeleiteten Modellierens

# Die vier Schritte des statistischen Modellierens

1. Man wählt eines der vier Ziele des Modellierens (z.B. ein prädiktives Modell).
2. Man wählt ein Modell aus (genauer: eine Modellfamilie), z.B. postuliert man, dass die Körpergröße einen linearen Einfluss auf die Schuhgröße habe.
3. Man bestimmt (berechnet) die Details des Modells anhand der Daten: Wie groß ist die Steigung der Geraden und wo ist der Achsenabschnitt? Man sagt auch, dass man die *Modellparameter* anhand der Daten schätzt ("Modellinstantiierung" oder "Modellanpassung", engl. "model fitting").
4. Dann prüft man, wie gut das Modell zu den Daten passt (Modellgüte, engl. "model fit"); wie gut lässt sich die Schuhgröße anhand der Körpergröße vorhersagen bzw. wie groß ist der Vorhersagefehler?

# Einfache vs. komplexe Modelle: Unter- vs. Überanpassung

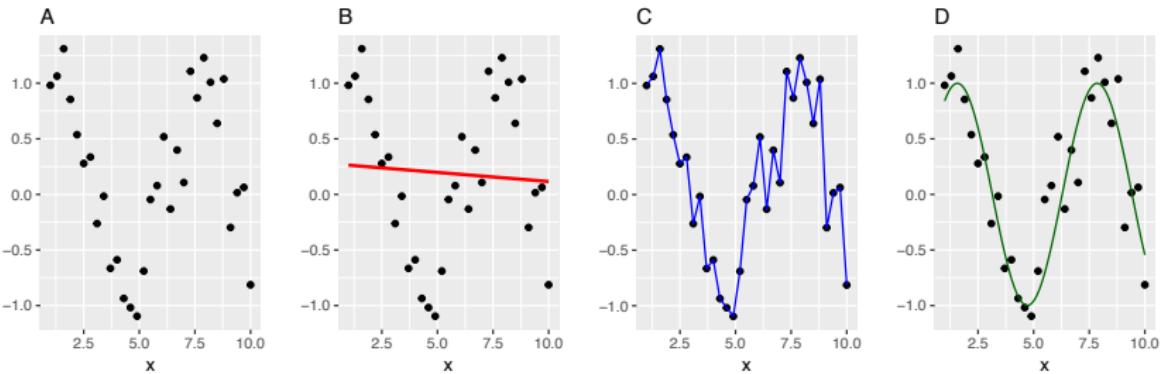


Abbildung 32: Welches Modell (Teil B-D; rot, grün, blau) passt am besten zu den Daten (Teil A) ?

# Vorhersagegüte der Trainings-Stichprobe vs. der Test-Stichprobe

Beschreibt ein Modell (wie das blaue Modell hier) eine Stichprobe sehr gut, heißt das noch *nicht*, dass es auch zukünftige (und vergleichbare) Stichproben gut beschreiben wird. Die Güte (Vorhersagegenauigkeit) eines Modells sollte sich daher stets auf eine neue Stichprobe beziehen (Test-Stichprobe), die nicht in der Stichprobe beim Anpassen des Modells (Trainings-Stichprobe) enthalten war.

# Overfitting

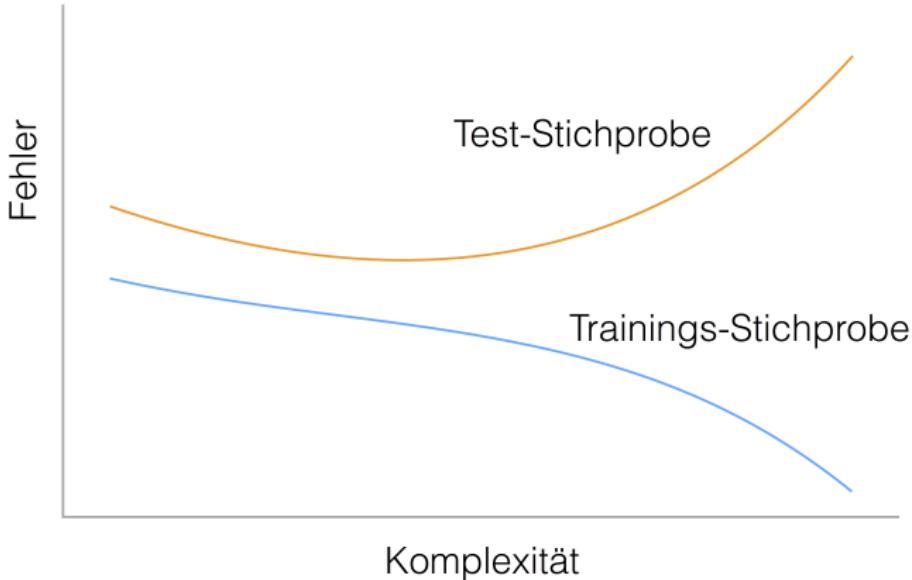


Abbildung 33: 'Mittlere' Komplexität hat die beste Vorhersagegenauigkeit (am wenigsten Fehler) in der Test-Stichprobe

# Bias-Varianz-Abwägung

*Einfache Modelle: Viel Bias, wenig Varianz. Komplexe Modelle: Wenig Bias, viel Varianz.*

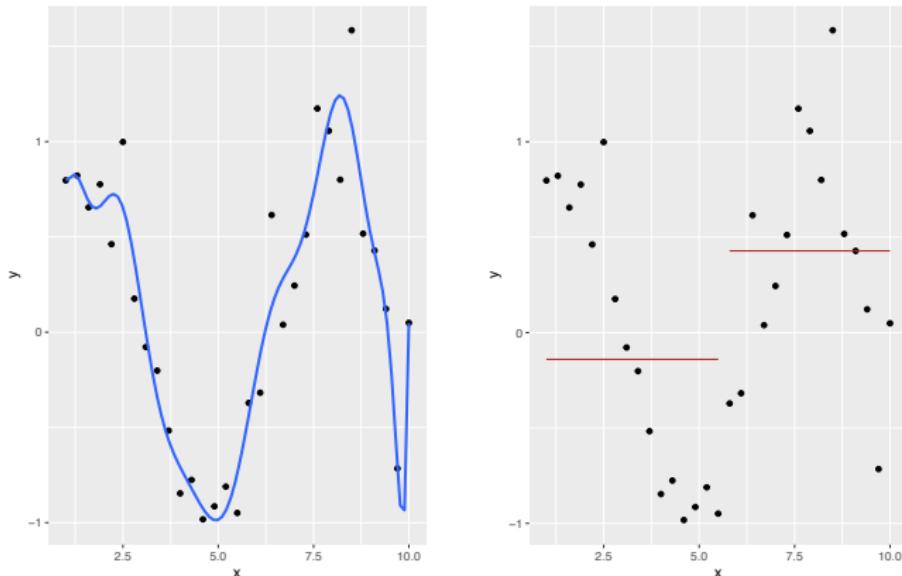


Abbildung 34: Der Spagat zwischen Verzerrung und Varianz

# Befehlsübersicht

Tabelle 6: Befehle des Kapitels ‘Modelli...

Paket::Funktion	Beschreibung
dplyr::sample_frac	Zielt eine Stichprobe von x% aus einem Dataframe
dplyr::anti_join	Behält alle Zeilen von df1, die *nicht in df2 vorkommen
dplyr::slice	Schneidet eine ‘Scheibe’ aus einem Datensatz (filter Zei...

# Der p-Wert

# Lernziele

- Den p-Wert erläutern können.
- Den p-Wert kritisieren können.
- Alternativen zum p-Wert kennen.
- Inferenzstatistische Verfahren für häufige Fragestellungen kennen.

# Sir Ronald Fisher, Erfinder des Nullhypothesen Testens

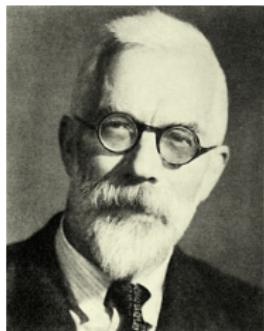


Abbildung 35: Der größte Statistiker des 20. Jahrhunderts ( $p < .05$ )

# Der p-Wert ist die heilige Kuh der Forscher

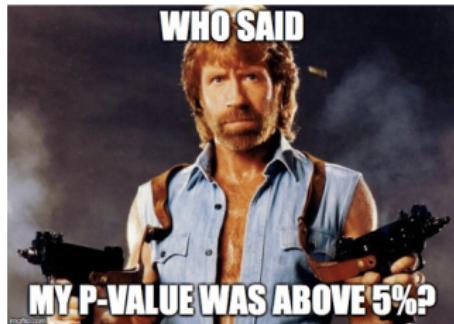


Abbildung 36: Der p-Wert wird oft als wichtig erachtet

*Der p-Wert sagt, wie gut die Daten zur Nullhypothese passen.*

# Von Männern und Päpsten

$$P(M|T) \neq P(T|M)$$

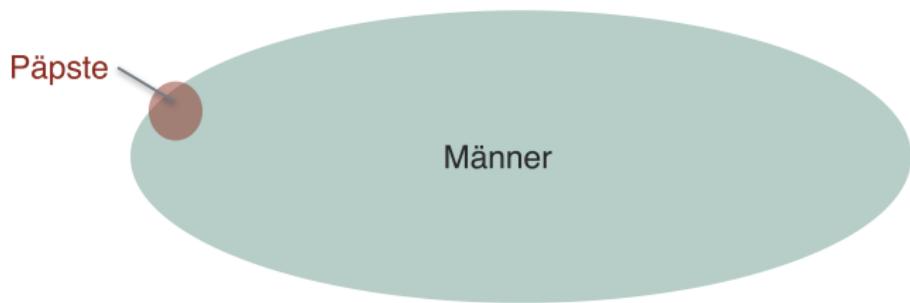


Abbildung 37: Moslem und Terrorist zu sein, ist nicht das gleiche.

# Der p-Wert ist eine Funktion der Stichprobengröße

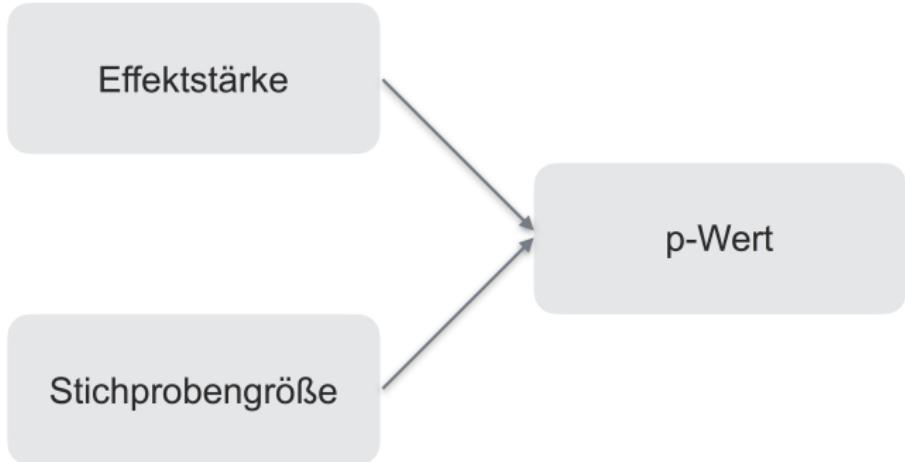


Abbildung 38: Zwei Haupteinflüsse auf den p-Wert

# Zur Philosophie des p-Werts: Frequentismus

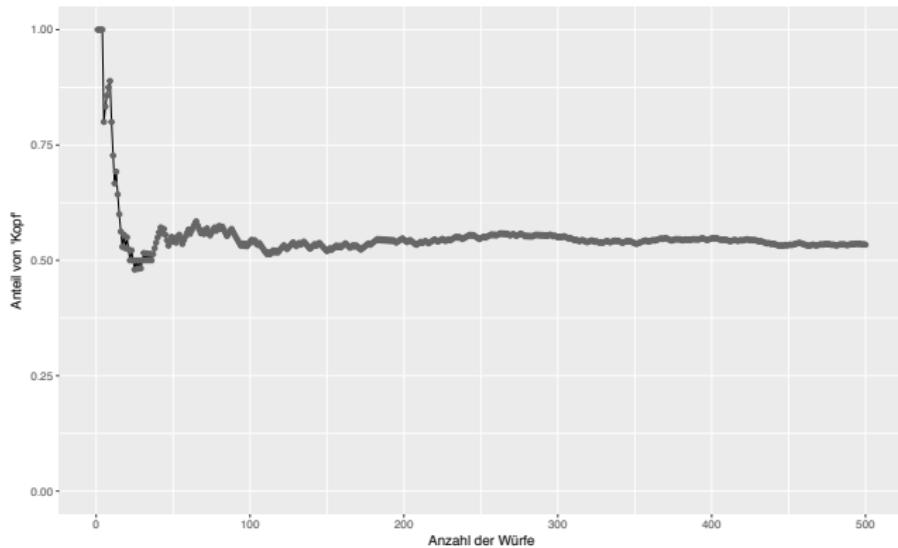


Abbildung 39: Anteil von 'Kopf' bei wiederholtem Münzwurf

# Alternativen zum p-Wert - Konfidenzintervalle

*Das 95%-Konfidenzintervall ist der Bereich, in dem der Parameter in 95% der Fälle fallen würde bei sehr häufiger Wiederholung des Versuchs.*

## Visualisierung zum Konfidenzintervall

# Alternativen zum p-Wert - Effektstärken

Tabelle 7: Überblick ü

Name	Test
Cohens d	Unterschied zwischen zwei Mittelwerten
Korrelationskoeffizient nach Pearson r	Zusammenhang zweier metrischer Größen
p	Unterschied in zwei Anteilen
$R^2$ , $\eta^2$	Anteil aufgeklärter Varianz (Varianzanteil)
$f^2$	Verhältnis von erklärter zu nicht erklärter Varianz
$\omega$	Häufigkeitsunterschiede

# Alternativen zum p-Wert - Bayes-Statistik

$$p(D|H)$$

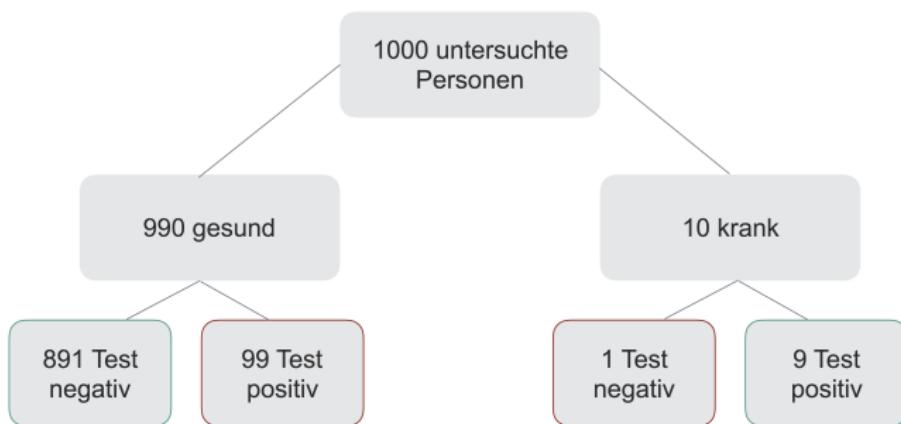


Abbildung 40: Die zwei Stufen der Bayes-Statistik in einem einfachen Beispiel

# Klassische lineare Regression

# Lernziel

## Lernziele:

- Wissen, was man unter Regression versteht.
- Die Annahmen der Regression überprüfen können.
- Regression mit kategorialen Prädiktoren durchführen können.
- Die Regression inferenzstatistisch absichern können.
- Die Modellgüte bei der Regression bestimmen können.
- Vertiefende Aspekte beherrschen, wie Modellwahl und Interaktionen.

# Beispiel für eine lineare Regression

`score = achsenabschnitt + steigung*study_time`

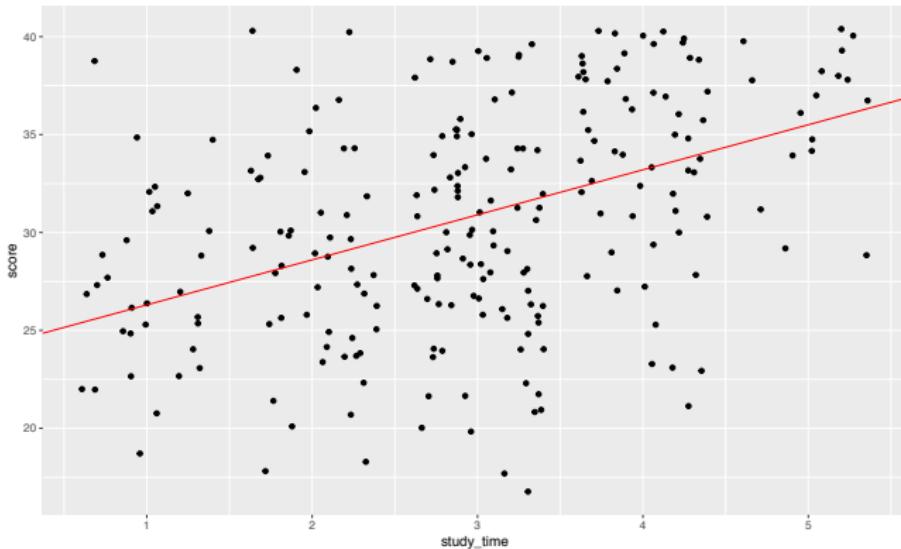


Abbildung 41: Beispiel für eine Regression

# Vorhersagegüte - Veranschaulichung

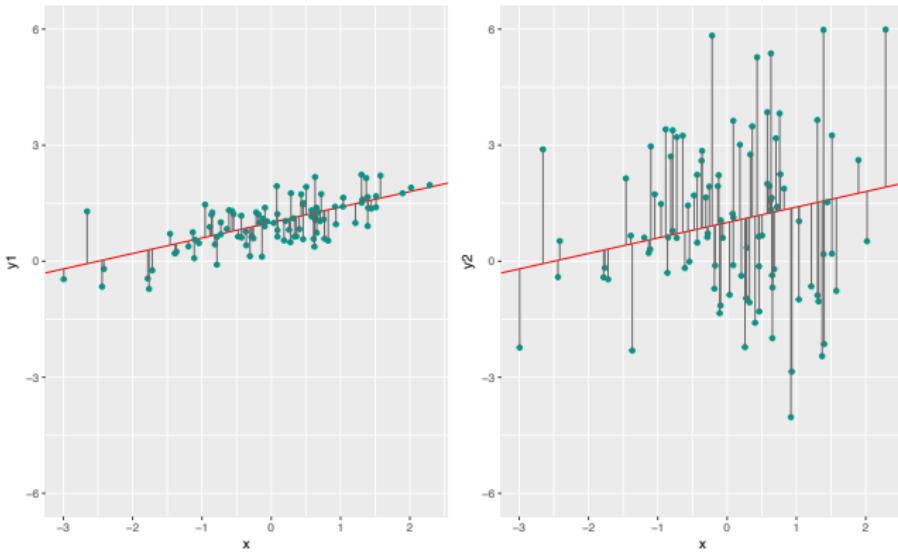


Abbildung 42: Geringer (links) vs. hoher (rechts) Vorhersagefehler

# Vorhersagegüte - MSE und $R^2$

$$MSE = \frac{1}{n} \sum (pred - obs)^2$$

$$R^2 = 1 - \left( \frac{SS_T - SS_M}{SS_T} \right)$$