

Folien für das Modul ‘Praxis der Datenanalyse’

WS17



Grobgliederung

- Vorwort
- Rahmen
- Daten einlesen
- Datenjudo
- Daten visualisieren
- Grundlagen des Modellierens
- Der p-Wert
- Lineare Regression
- Klassifizierende (logistische) Regression
- Clusteranalyse
- Dimensionsreduktion
- Textmining
- Anhang

Vorwort

Hinweise

- Diese Folien vermitteln *nicht* den Stoff.
- Sie visualisieren nur einige zentrale Ideen.
- Der Stoff wird vom Skript vermittelt.
- Nutzen Sie das Skript zum eigentlichen Arbeiten.

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

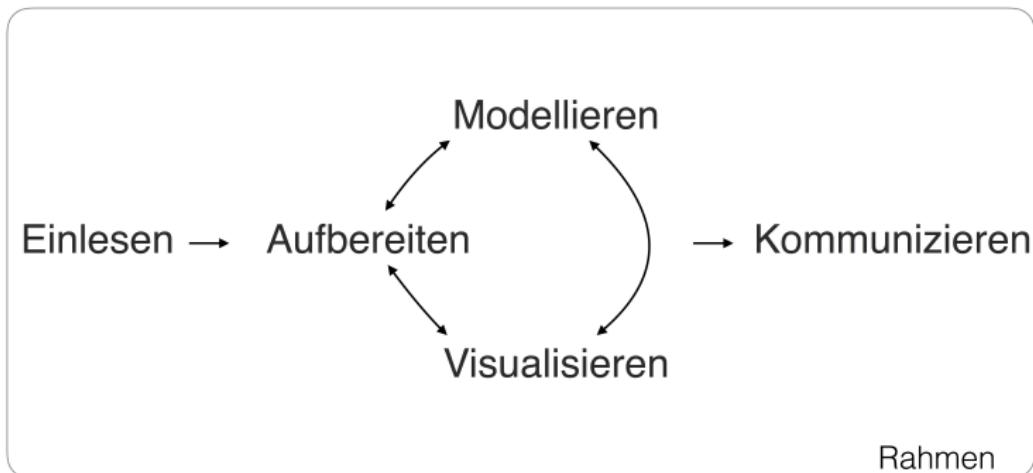
- Hinweise zur Datenanalyse finden Sie hier.
- Die Datenanalyse wird (in fast jeder Stunde) praktisch eingeübt.
- Beispiele für gute Datenanalysen von Studierenden finden Sie hier.

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



Rahmen

Abbildung 1: Der Prozess der Datenanalyse

R und RStudio installieren



Skript-Fenster

Umgebung

Konsole

Plots

The screenshot displays the RStudio desktop application. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The main window is divided into several panes:

- Script Editor:** Shows the R code `qplot(x = factor(cyl), y = hp, geom = "boxplot", data = mtcars)`.
- Environment:** Displays the message "Environment is empty".
- Plots:** Shows a box plot for the `hp` variable grouped by `cyl` (4, 6, 8).
- Console:** Displays the R startup message and information about the R version and platform.

Abbildung 2: RStudio

Folien: [http://tiny.cc/meyarw](#) | Praxis der Datenanalyse

WS17

13 / 140

Hilfe! R!

Beliebte Fehler beim Installieren von Paketen:

- `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 leichtgemacht

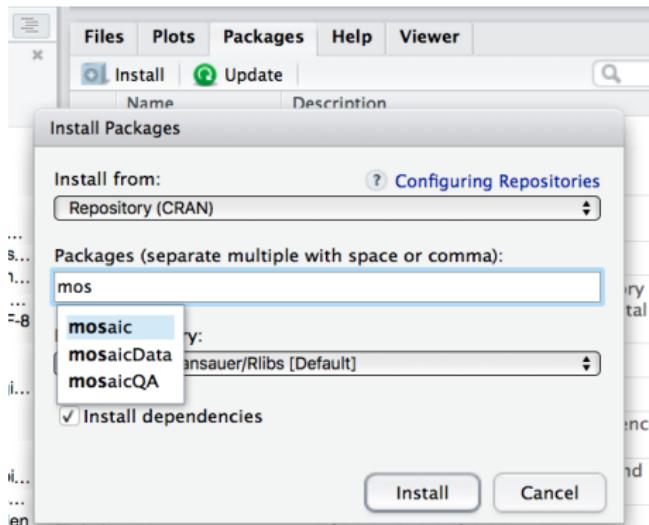


Abbildung 3: So installiert man Pakete in RStudio

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 soll.

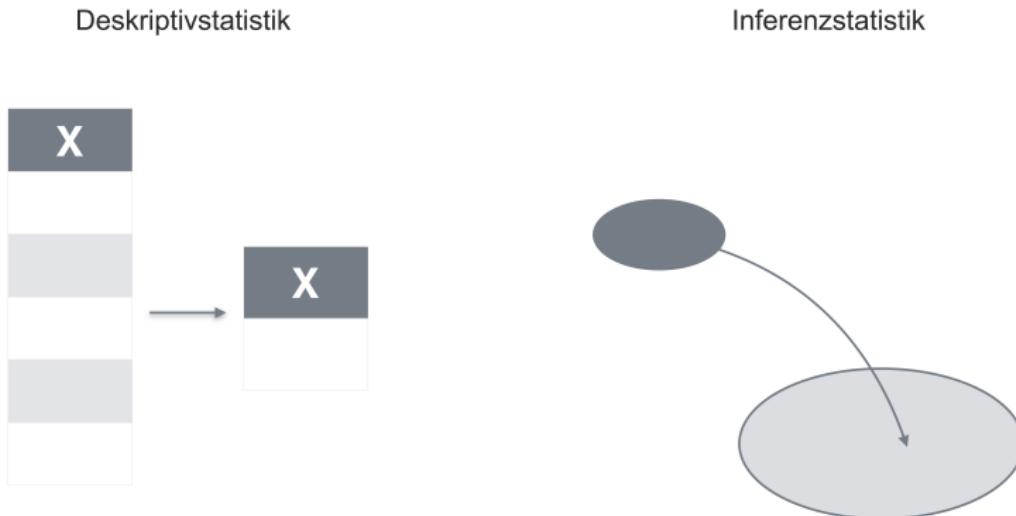


Abbildung 4: Sinnbild für die Deskriktiv- und die Inferenzstatistik

Abduktion als klassische Denkfigur in der Statistik

“ Prämisse 1: Wenn Modell M wahr ist,
dann sollten die Daten das Muster D aufweisen. Prämisse 2: Die Daten weisen das Muster
D auf. — Konklusion: Daher muss das Modell M wahr sein.

\normalsize

Die Konklusion ist ***nicht*** zwangsläufig richtig.

```
# Daten einlesen {#tidy}
```

```
## 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).

Das Arbeitsverzeichnis mit RStudio wählen

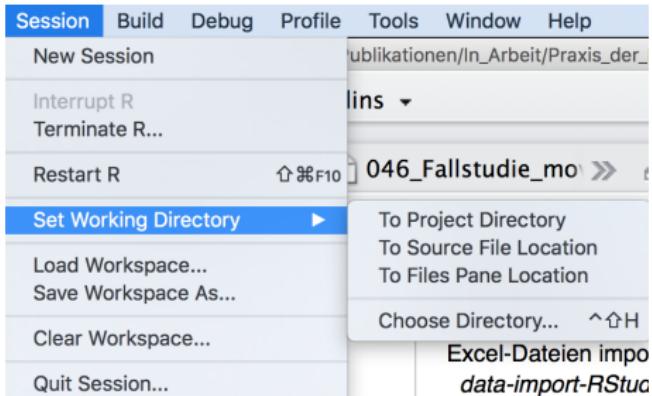


Abbildung 5: Das Arbeitsverzeichnis mit RStudio auswählen

Normalform einer Tabelle

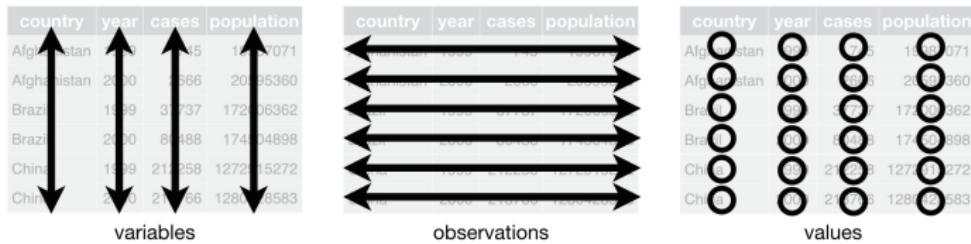


Abbildung 6: Schematische Darstellung eines Dataframes in Normalform

Breit vs. Lang

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



Abbildung 7: Dieselben Daten - einmal breit, einmal lang

Ein Dataframe in Normalform - Beispiel

Datensatz (Normalform)

in Zeilen: Fall/ Beobachtung
(häufig Personen)

ID	age	sex	n_FB_friends
Anna	21	female	212
Berla	24	female	235
Carla	20	male	312
Dora	20	female	21435

in Spalten:
Merkmal/ Variable Wert/ Ausprägung

The diagram illustrates a data frame in normal form. It shows a table with four columns: ID, age, sex, and n_FB_friends. The rows represent individual observations (falls/beobachtungen) of people. The columns represent variables (Merkmale). The value '312' is highlighted in the sex column for the observation 'Carla'. Arrows point from descriptive text to specific parts of the table to explain the structure.

Abbildung 8: Illustration eines Datensatzes in Normalform

Tabelle in Normalform bringen

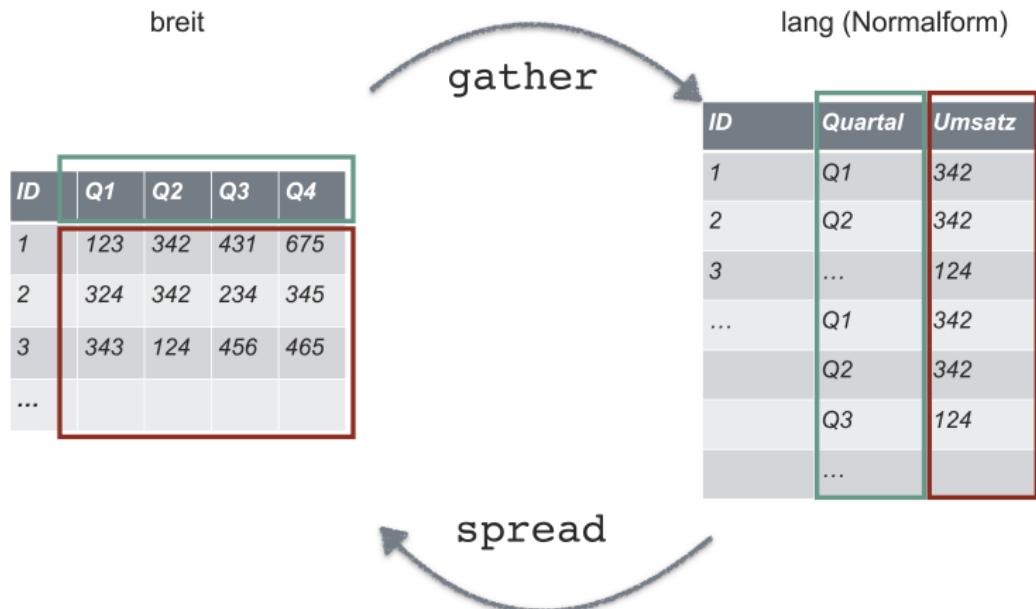


Abbildung 9: Mit 'gather' und 'spread' wechselt man von der breiten Form zur langen Form

Beispiel für die Normalisierung einer Tabelle

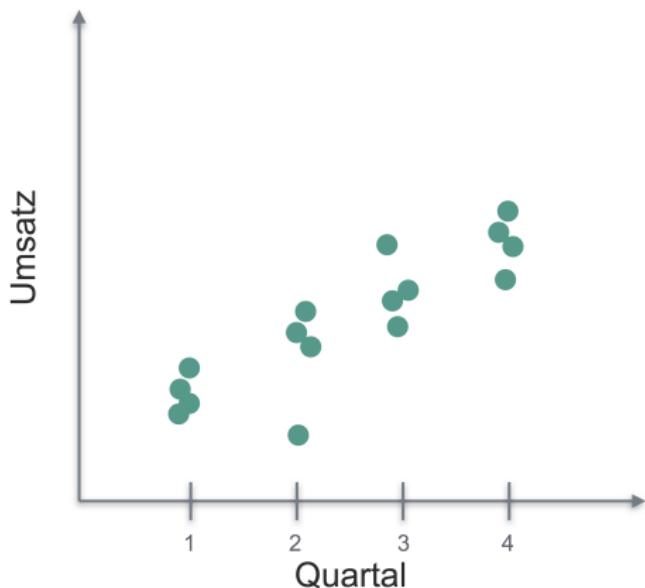


Abbildung 10: Ein Beispiel für eine Abbildung zu einer Normalform-Tabelle

gather und spread

```
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")
```

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

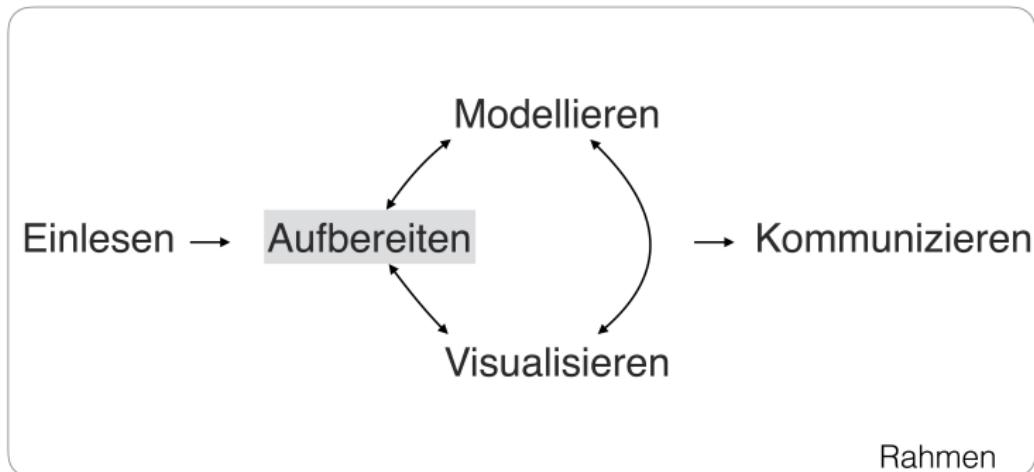


Abbildung 11: Daten aufbereiten

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 12: Lego-Prinzip: Zerlege eine komplexe Struktur in einfache Bausteine

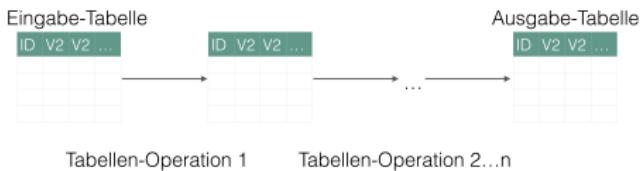


Abbildung 13: Durchpfeifen: Ein Dataframe wird von Operation zu Operation weitergereicht

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 14: 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 15: Spalten auswählen

Zeilen sortieren mit arrange

The diagram illustrates the use of the `arrange` function for sorting rows. It shows two tables side-by-side, connected by a horizontal arrow pointing from left to right.

Left Table (Initial Data):

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

Text: Gute Noten zuerst!

Right Table (Sorted Data):

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

Abbildung 16: Spalten sortieren

Datensatz gruppieren mit group_by

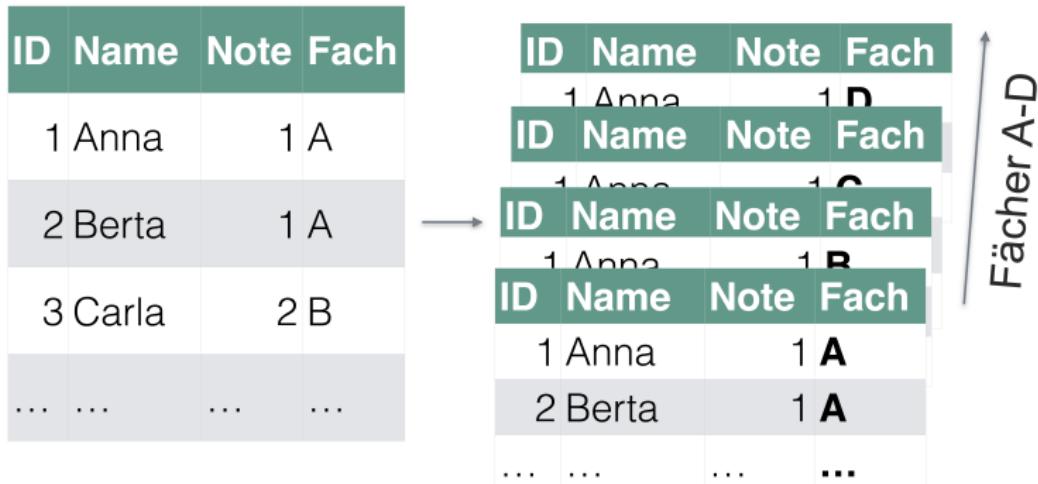


Abbildung 17: Datensätze nach Subgruppen aufteilen

Eine Spalte zusammenfassen mit summarise



Abbildung 18: Spalten zu einer Zahl zusammenfassen

Zeilen zählen mit n und count

Gruppe A Gruppe B Gruppe C



5 3 4

Abbildung 19: Sinnbild für 'count'

Die Pfeife



Abbildung 20: Das ist keine Pfeife

Befehle hintereinander reihen mit der Pfeife

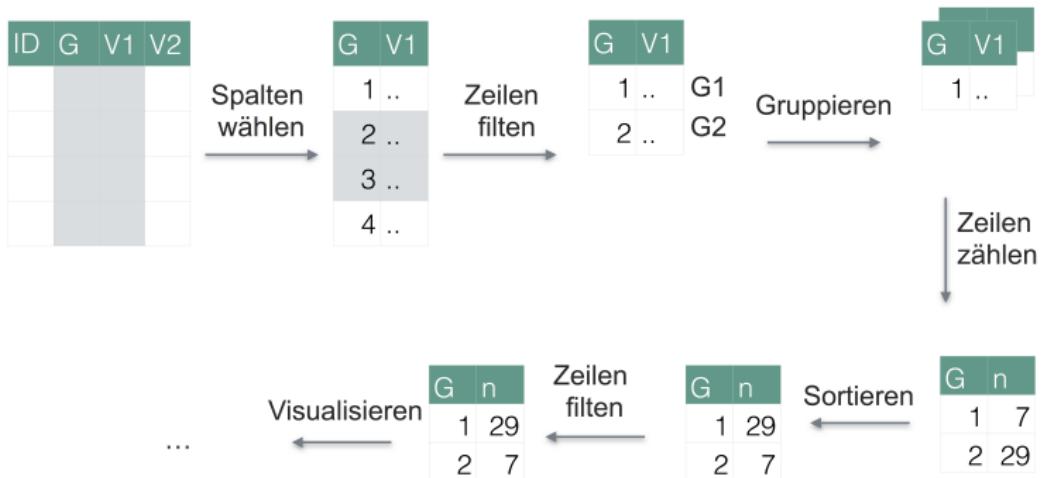


Abbildung 21: Das 'Durchpeifen'

Introducing Pipe-Syntax

Vergleichen Sie mal diese Syntax

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

mit dieser

```
stats_test %>% filter(!is.na(score)) %>% group_by(interest) %>%  
    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

Will Durchschnittsnote pro Student wissen!

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

Abbildung 22: 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)
```

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 . . .

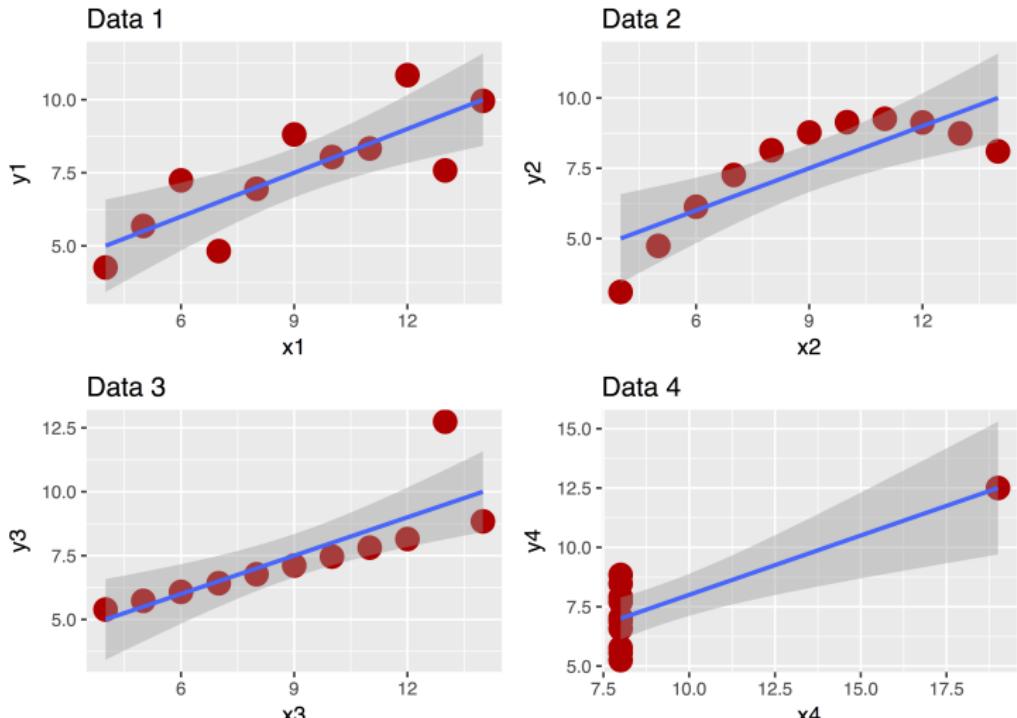
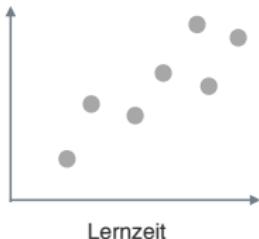


Abbildung 23: Das Anscombe Quartett

Die Anatomie eines Diagramms

Klausurerfolg



Lernzeit



Klausurerfolg

Lernzeit

	Lernzeit	Punkte in Klausur
Anna	10	30
Berta	30	60
Carla	20	90
Dora	50	120

Daten

Achsen

Geom

Beispiel für ein Diagramm mit ggplot2::qplot

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

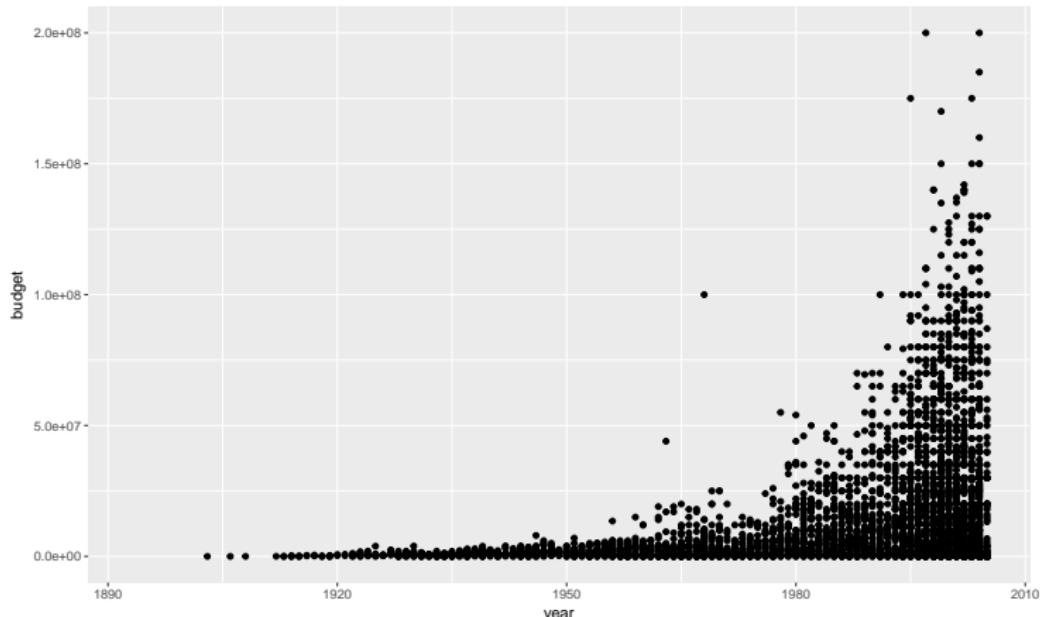


Abbildung 25: Mittleres Budget pro Jahr

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:

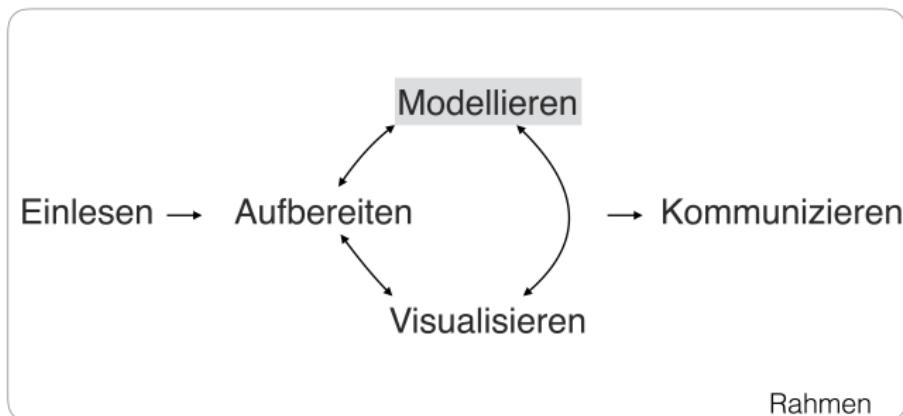
```
qplot(x = X_Achse, y = Y_Achse, data = mein_dataframe, geom =
```

Häufige Diagrammtypen

- Histogramm, Dichtediagramm
- Punkte, Schachbrett-Diagramme
- Balkendiagramm
- Mosaicplot (Fliesen-Diagramm)
- Punktediagramm für Zusammenfassungen
- Boxplots

Grundlagen des Modellierens

Prozess der Datenanalyse - Modellieren



Was ist ein Modell



Abbildung 26: Modell eines VW-Käfers

Die Beziehung von Gegenstandsbereich und Modell

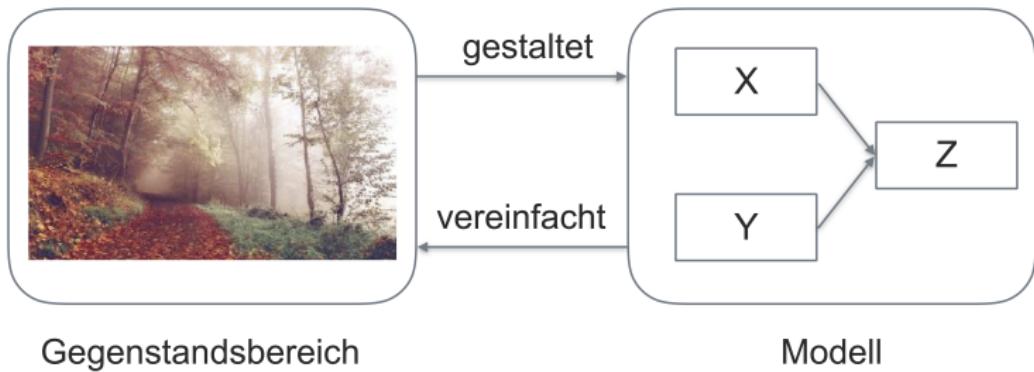


Abbildung 27: Modellieren

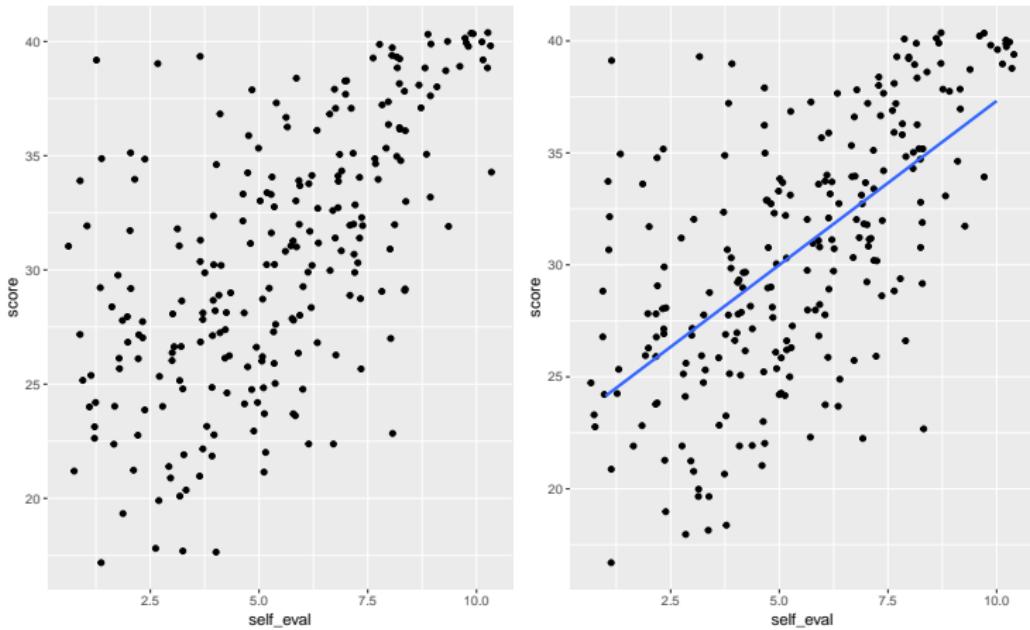
Modelle spiegeln empirische Relationen in numerischen Relationen

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



Abbildung 28: 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

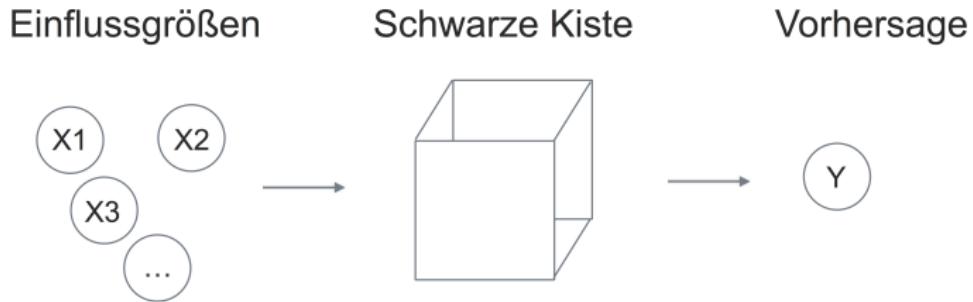


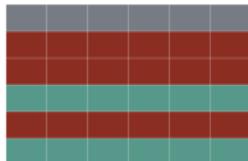
Abbildung 29: Modelle mit schwarzer Kiste

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

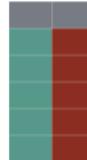
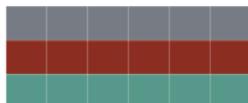
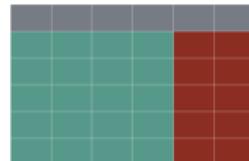


Abbildung 30: Die zwei Arten des ungeleiteten Modellierens

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 Modelfamilie), 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

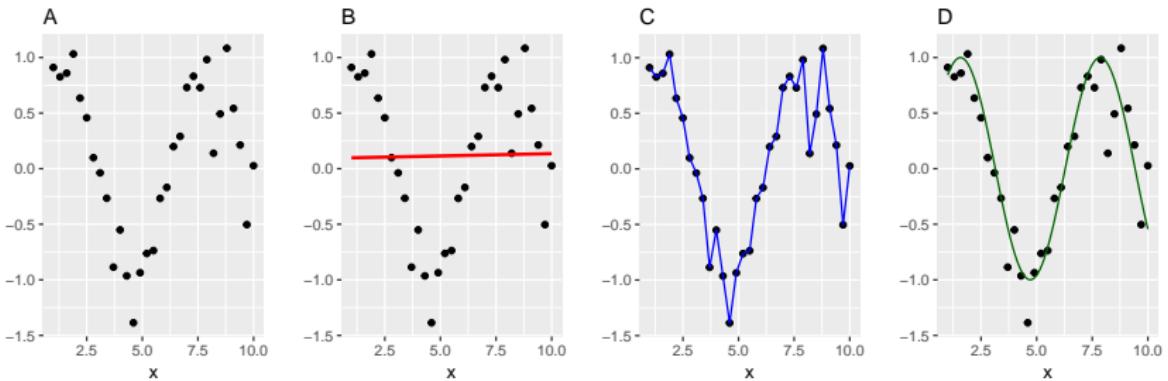


Abbildung 31: Welches Modell (Teil B-D; rot, grün, blau) passt am besten zu den Daten (Teil A) ?

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

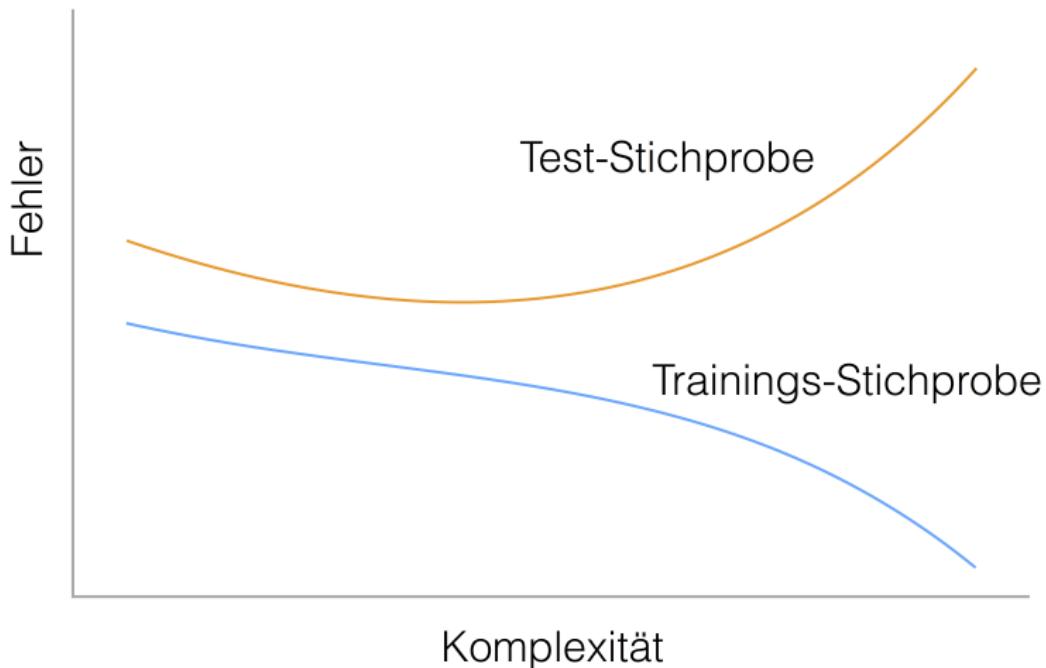


Abbildung 32: 'Mittlere' Komplexität hat die beste Vorhersagegenauigkeit (am wenigsten Fehler) in der Test-Stichprobe

Bias-Varianz-Abwägung

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

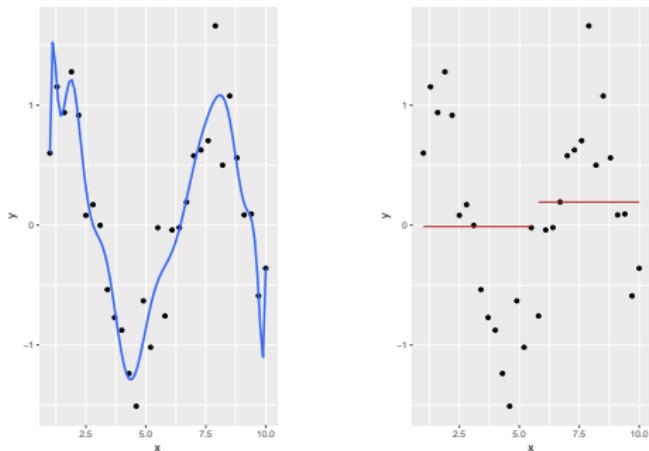


Abbildung 33: Der Spagat zwischen Verzerrung und Varianz

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 Nullhypotesen Testens

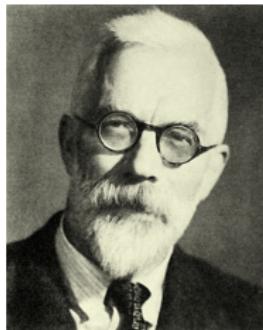


Abbildung 34: Der größte Statistiker des 20. Jahrhunderts ($p < .05$)

Der p-Wert ist die heilige Kuh der Forscher

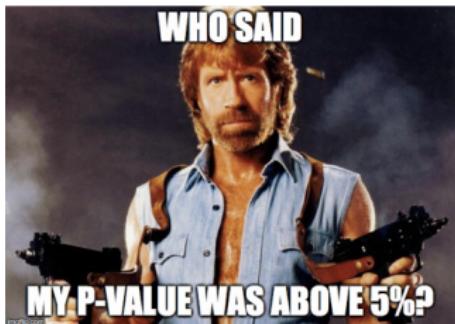


Abbildung 35: Der p-Wert wird oft als wichtig erachtet

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

Von Männern und Päpsten

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

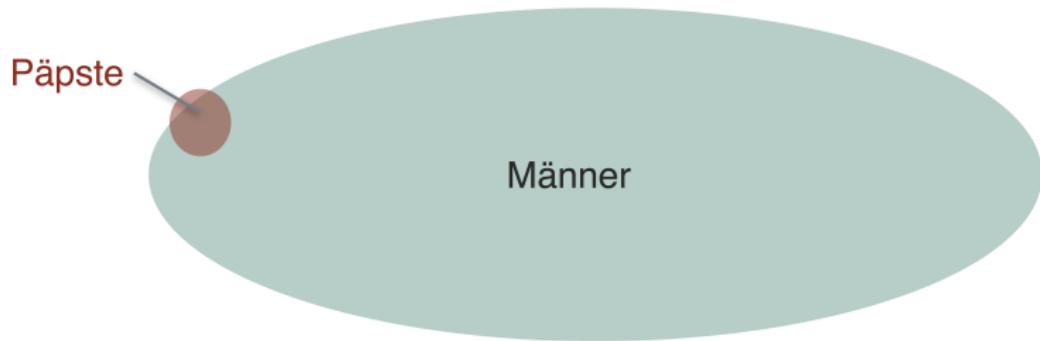


Abbildung 36: Mann und Papst zu sein, ist nicht das gleiche.

Der p-Wert ist eine Funktion der Stichprobengröße

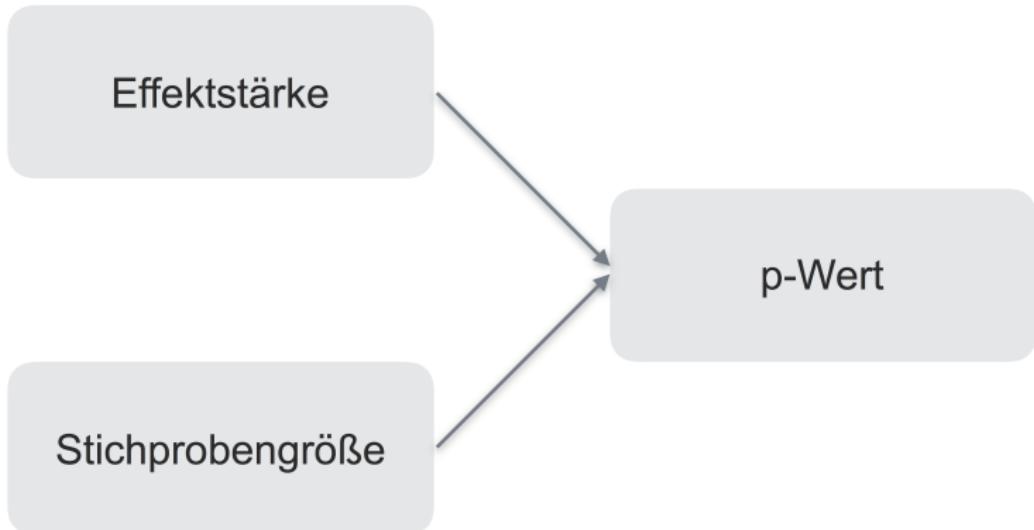


Abbildung 37: Zwei Haupteinflüsse auf den p-Wert

Zur Philosophie des p-Werts: Frequentismus

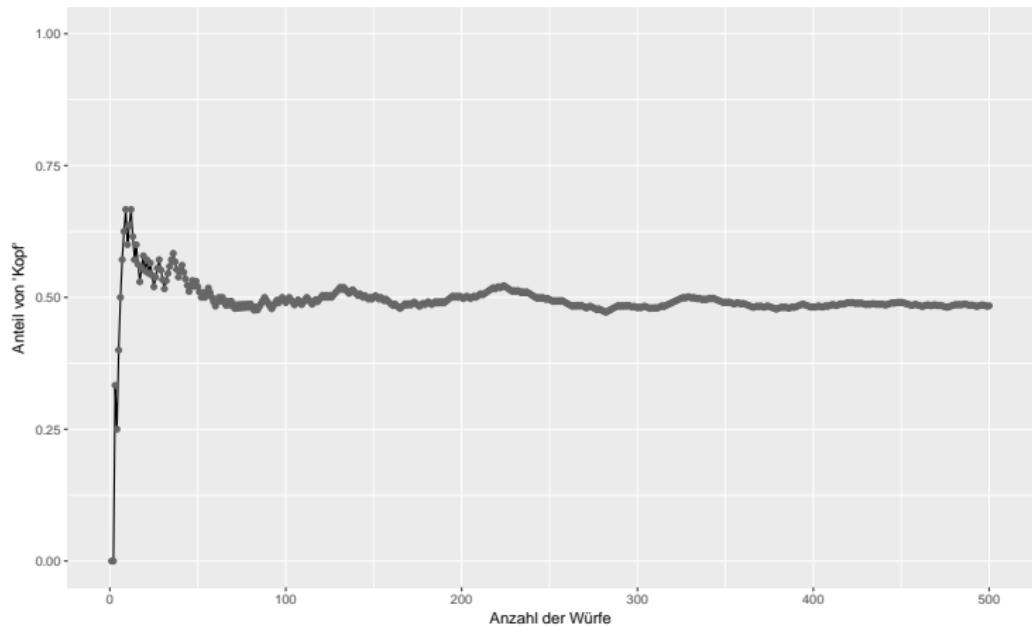


Abbildung 38: Anteil von 'Kopf' bei wiederholtem Münzwurf

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 2: Überblick über gängige Effektstärkemaße (continued below)

Name	Test	kleiner Effekt
Cohens d	Unterschied zwischen zwei Mittelwerten	.2-.5
r	Zusammenhang zweier metrischer Größen	0.1
p	Unterschied in zwei Anteilen	NA
R^2, η^2	Anteil aufgeklärter Varianz (Varianzanalyse, Regressionsanalyse)	0.01
f^2	Verhältnis von erklärter zu nicht erklärter Varianz (signal-to-noise ratio)	0.02
ω	Häufigkeitsunterschiede	0.1

Alternativen zum p-Wert - Bayes-Statistik

$$p(D|H)$$

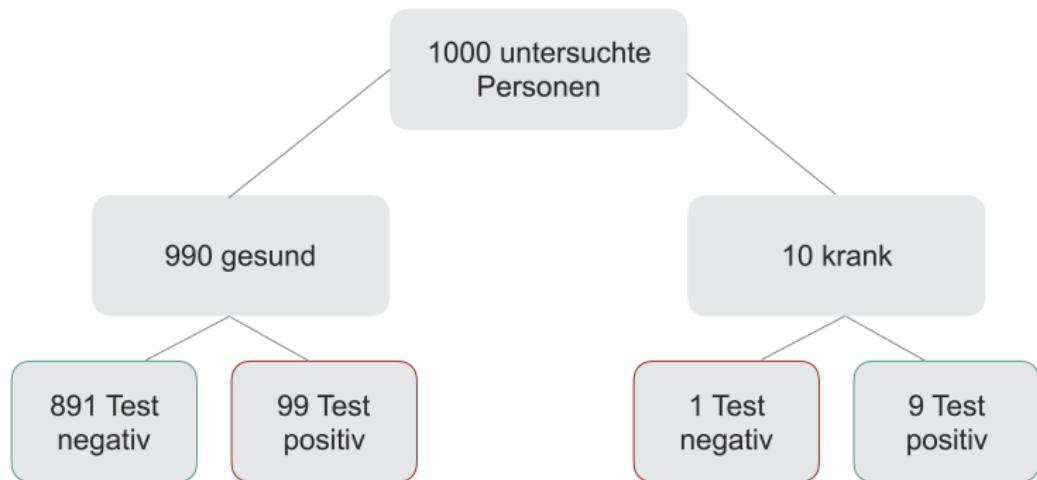


Abbildung 39: Die zwei Stufen der Bayes-Statistik in einem einfachen Beispiel

Lineare Regression

Lernziele

- Wissen, was man unter Regression versteht.
- Die Annahmen der Regression überprüfen können.
- Regression mit kategorialen Prädiktoren durchführen können.
- Die Modellgüte bei der Regression bestimmen können.
- Interaktionen erkennen und ihre Stärke einschätzen können.

Beispiel für eine lineare Regression

score = achsenabschnitt + steigung*study_time

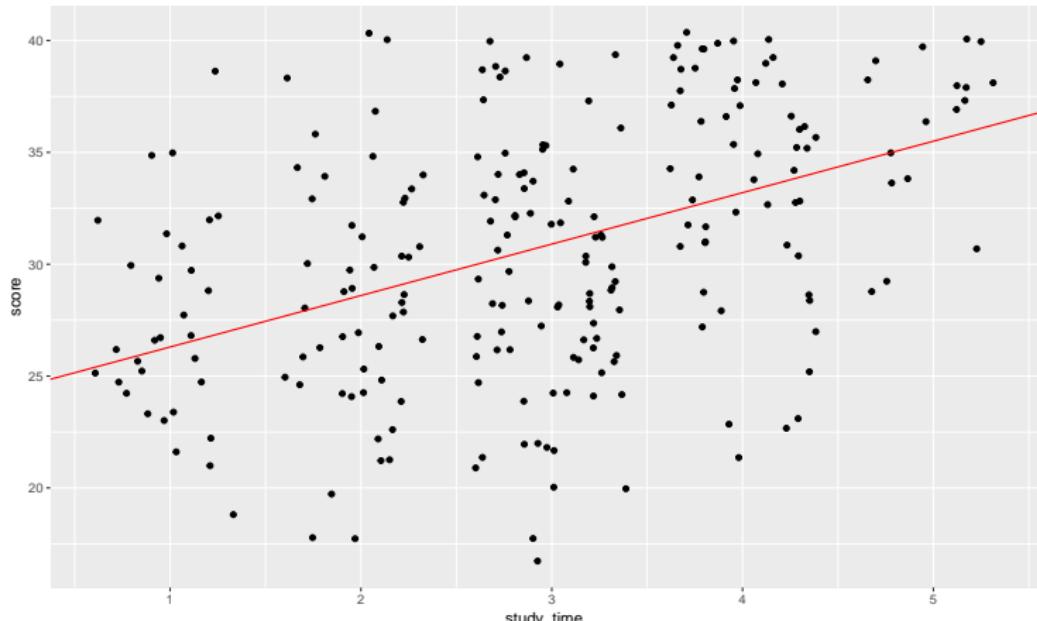


Abbildung 40: Beispiel für eine Regression

Die Formel einer einfachen Regression

```
score = achsenabschnitt + steigung*study_time
```

Vorhersagegüte - Veranschaulichung

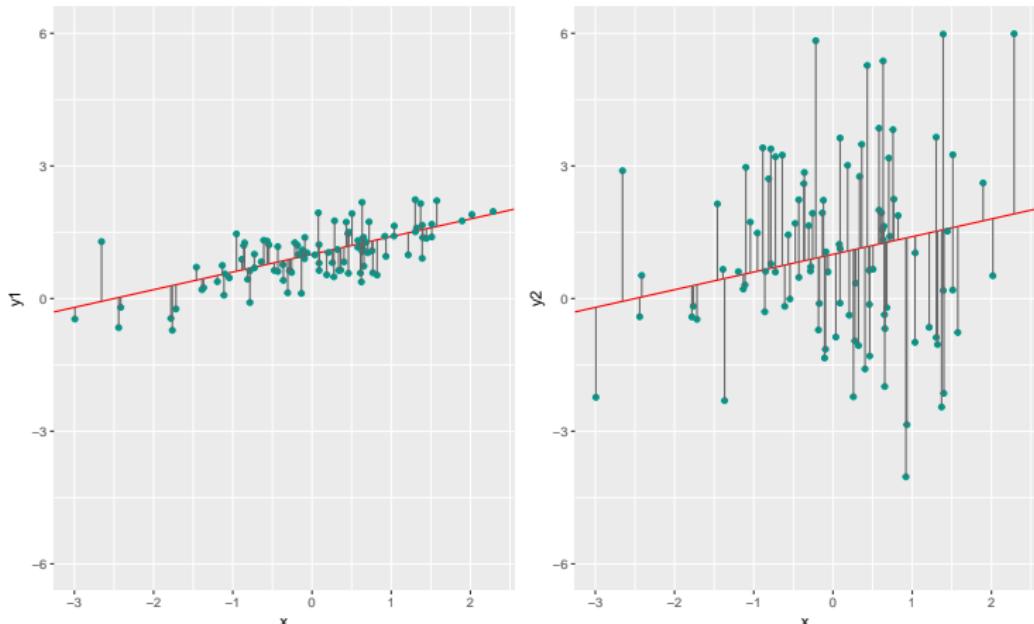


Abbildung 41: Geringer (links) vs. hoher (rechts) Vorhersagefehler

Vorhersagegüte - MSE und R^2

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

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

Überprüfung der Annahmen der linearen Regression

- Linearität des Zusammenhangs
- Normalverteilung der Residuen
- Konstante Varianz
- Extreme Ausreißer
- Unabhängigkeit der Beobachtungen

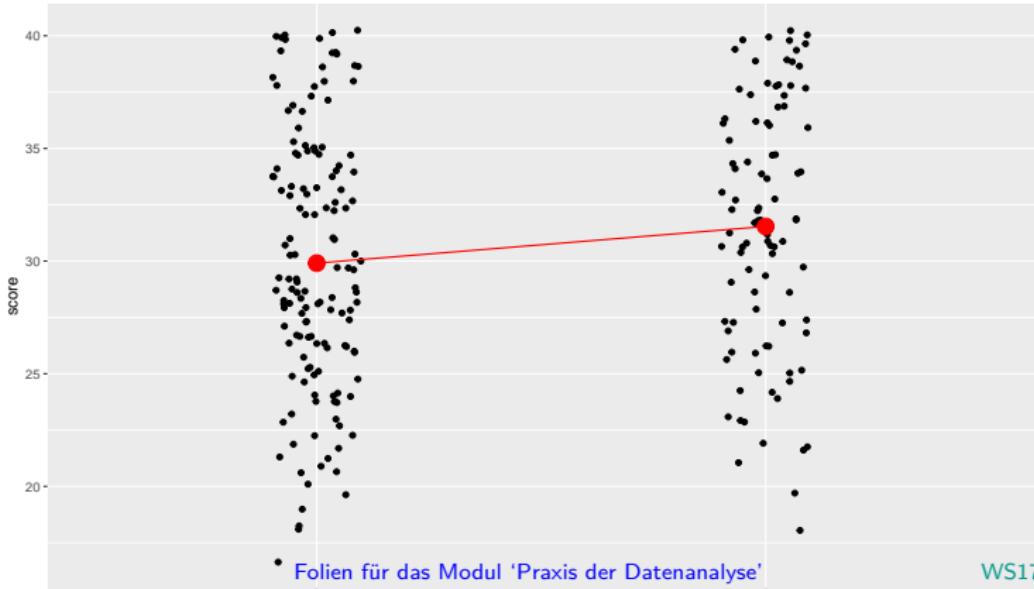
Unterscheiden sich Interessierten von Nicht-l. im Klausurerfolg?

```
stats_test$interessiert <- stats_test$interest > 3
score_interesse <- stats_test %>% group_by(interessiert) %>%
score_interesse

## # A tibble: 3 x 2
##   interessiert     score
##       <lgl>     <dbl>
## 1 FALSE      29.90909
## 2 TRUE       31.53684
## 3 NA         33.08824
```

Kategoriale Prädiktoren

```
stats_test %>% na.omit %>% ggplot() + aes(x = interessiert, y  
    geom_point(data = score_interesse, color = "red", size = 5  
group = 1, color = "red")
```



Multiple Regression

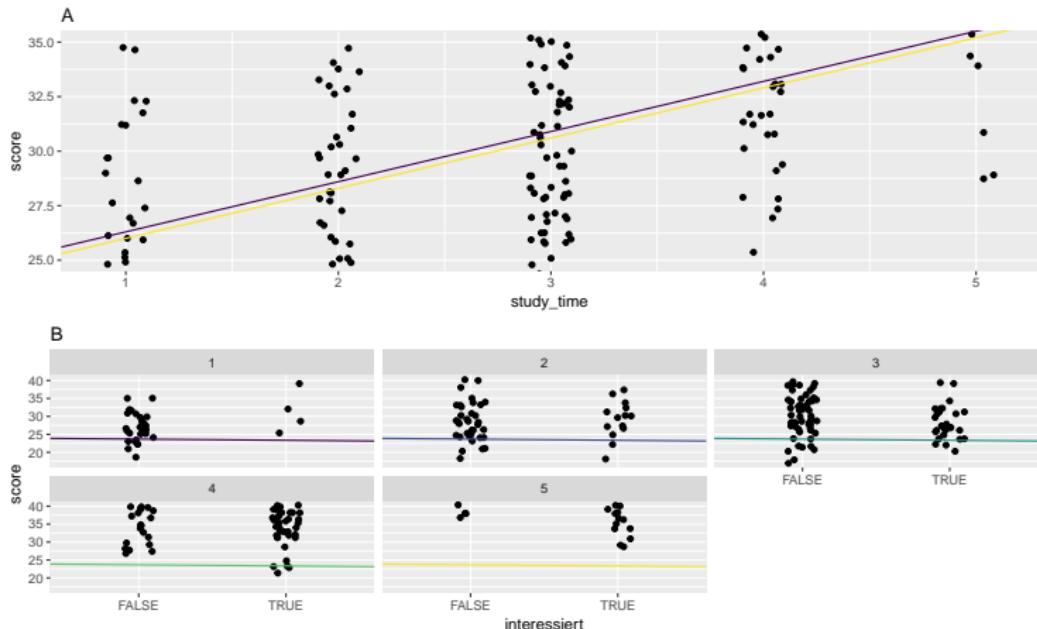


Abbildung 42: Eine multivariate Analyse fördert Einsichten zu Tage, die bei einfacheren Analysen verborgen bleiben

Multivariate Analysen sind cool

Die multivariate Analyse zeigt ein anderes Bild, ein genaueres Bild als die einfache Analyse. Ein Sachverhalt, der für den ganzen Datensatz gilt, kann in Subgruppen anders sein.

Erlaubt man der Regression, dass die Regressionsgeraden nicht parallel sein müssen, spricht man von einer *Interaktion*.

Ein Beispiel für einen Interaktionseffekt

Die Linien sind *nicht* (ganz) parallel: ein kleiner Interaktionseffekt.

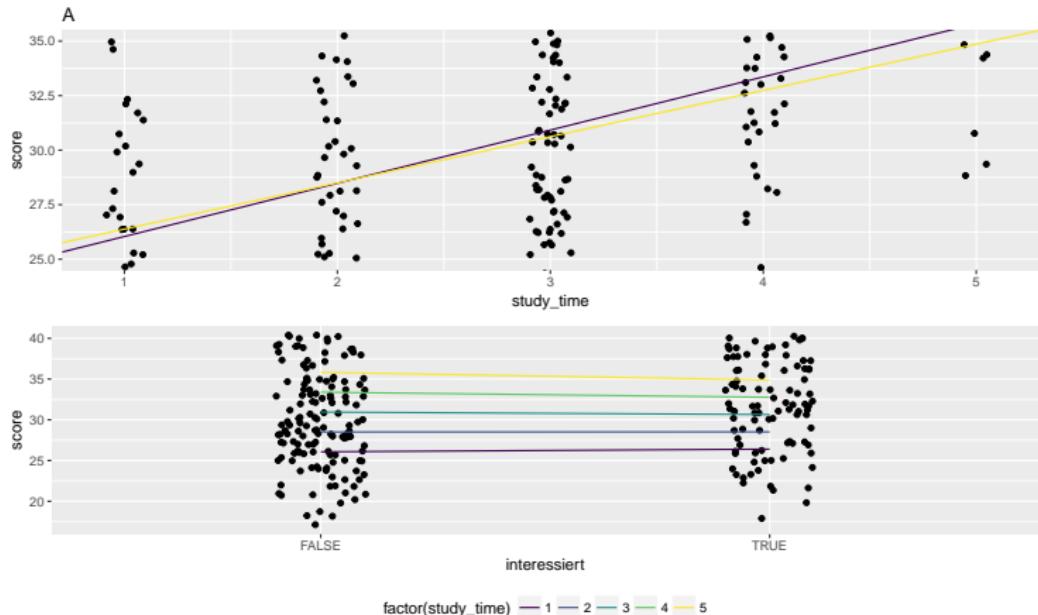


Abbildung 43: Eine Regressionsanalyse mit Interaktionseffekten

Fallstudie zu Overfitting

```
caret::postResample(pred = lm2_predict, obs = test$score)

##      RMSE Rsquared
## 4.433257 0.271658
```

Die Modellgüte im in der Test-Stichprobe ist meist schlechter als in der Trainings-Stichprobe. Das warnt uns vor Befunden, die naiv nur die Werte aus der Trainings-Stichprobe berichten.

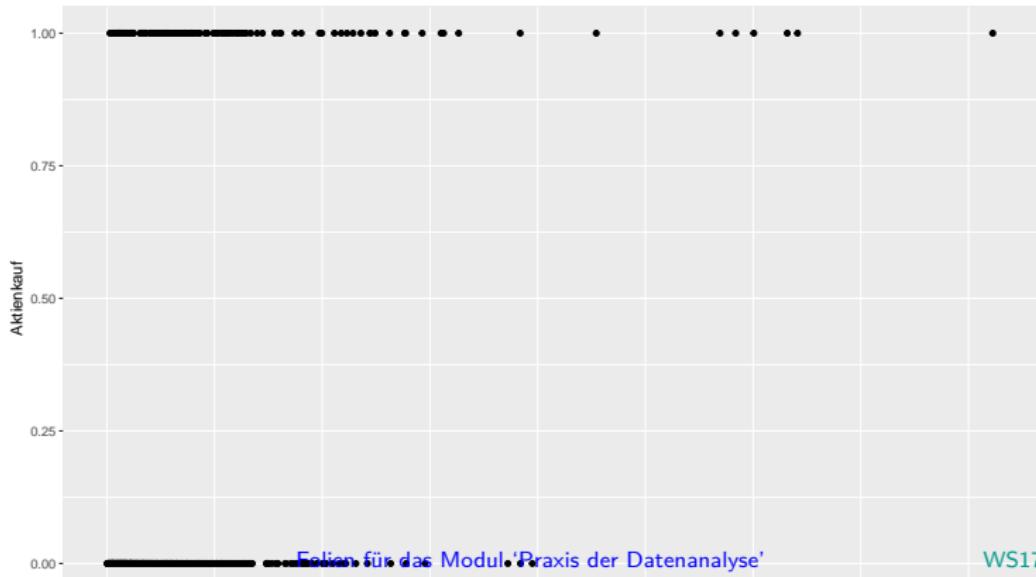
Klassifizierende (logistische) Regression

Lernziele

- Die Idee der logistischen Regression verstehen.
- Die Koeffizienten der logistischen Regression interpretieren können.
- Die Modellgüte einer logistischen Regression einschätzen können.
- Klassifikatorische Kennzahlen kennen und beurteilen können.

Problemstellung

```
p1 <- ggplot(aes(y = Aktienkauf, x = Risikobereitschaft), data = daten)
  geom_point()
```



R-Befehl

Die Funktion `glm` führt die logistische Regression durch.

```
glm1 <- glm(Aktienkauf ~ Risikobereitschaft, family = binomial)
```

Visualisierung der logistischen Regression

```
p1 + geom_abline(intercept = 0.18, slope = 0.05, color = "red")
```

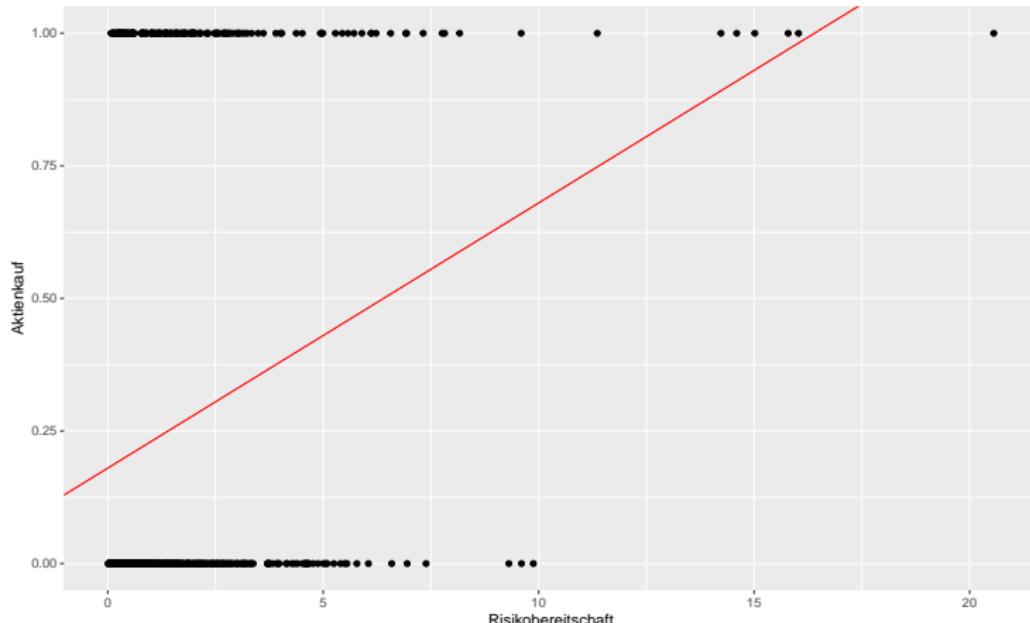


Abbildung 45: Regressionsgerade für Aktien-Modell

Formel der logistischen Regression

$$p(y = 1) = \frac{e^x}{1+e^x}$$

Die e-Funktion: $p(y=1) = \frac{e^x}{1+e^x}$

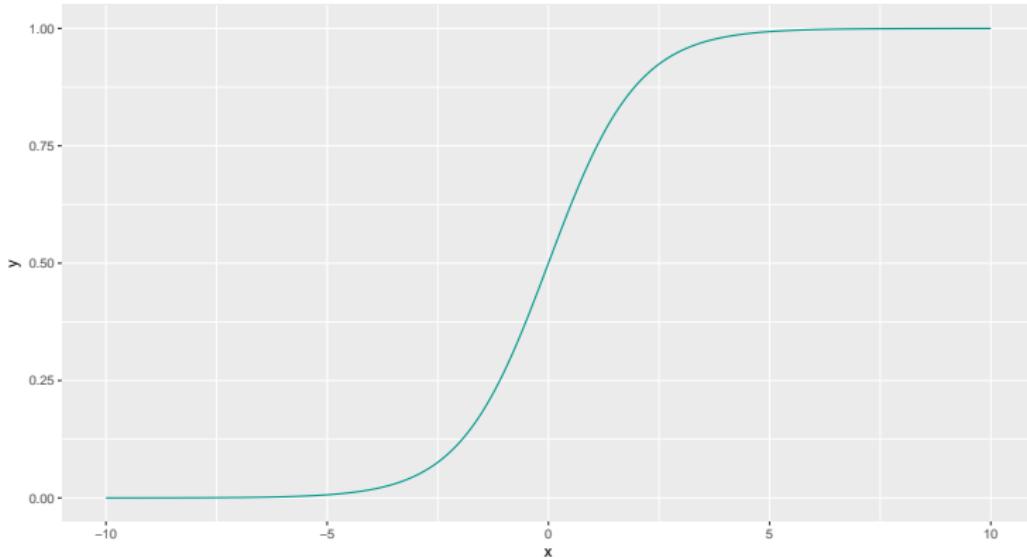


Abbildung 46: Die logistische Regression beschreibt eine 's-förmige' Kurve

Die logistische Regression für die Aktien-Daten

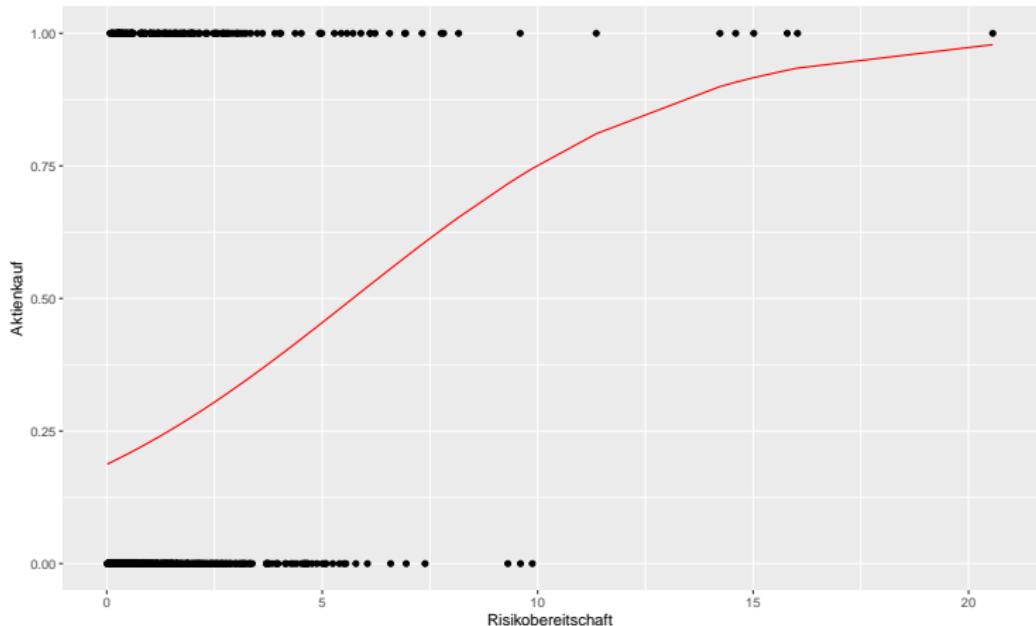


Abbildung 47: Modelldiagramm für den Aktien-Datensatz

Interpretation der Koeffizienten

Ist ein Logit \mathfrak{L} größer als 0, so ist die zugehörige Wahrscheinlichkeit größer als 50% (und umgekehrt.)

Logits \mathfrak{L} :

$$\mathfrak{L} = \ln\left(\frac{p}{1-p}\right)$$

y = intercept + 3*Risikobereitschaft, also

```
(y <- -1.469 + 3 * 0.257)
```

```
## [1] -0.698
```

Also $y = -0.698$ *Logits* (\mathfrak{L}).

Vorhersage individueller Wahrscheinlichkeiten

```
predict(glm1, data.frame(Risikobereitschaft = 1), type = "resp")  
  
##           1  
## 0.2294028
```

Kategoriale Prädiktoren

```
str(stats_test$bestanden)
stats_test$bestanden <- factor(stats_test$bestanden, levels =
log_stats <- glm(bestanden ~ interessiert, family = binomial('
summary(log_stats)
```

Vier Arten von Ergebnisse von Klassifikationen

Tabelle 4: Vier Arten von Ergebnisse von Klassifikationen (continued below)

Wahrheit	Als negativ (-) vorhergesagt
In Wahrheit negativ (-)	Richtig negativ (RN)
In Wahrheit positiv (+)	Falsch negativ (FN)
Summe	N*

Als positiv (+) vorhergesagt	Summe
Falsch positiv (FP)	N
Richtig positiv (RN)	P
P*	N+P

Konfusionsmatrix

```
(cm <- SDMTools::confusion.matrix(Aktien$Aktienkauf, glm1$fitt
```

```
##      obs
## pred   0   1
##      0 509 163
##      1   8  20
## attr(,"class")
## [1] "confusion.matrix"
```

```
sensitivity(cm)
```

```
## [1] 0.1092896
```

```
specificity(cm)
```

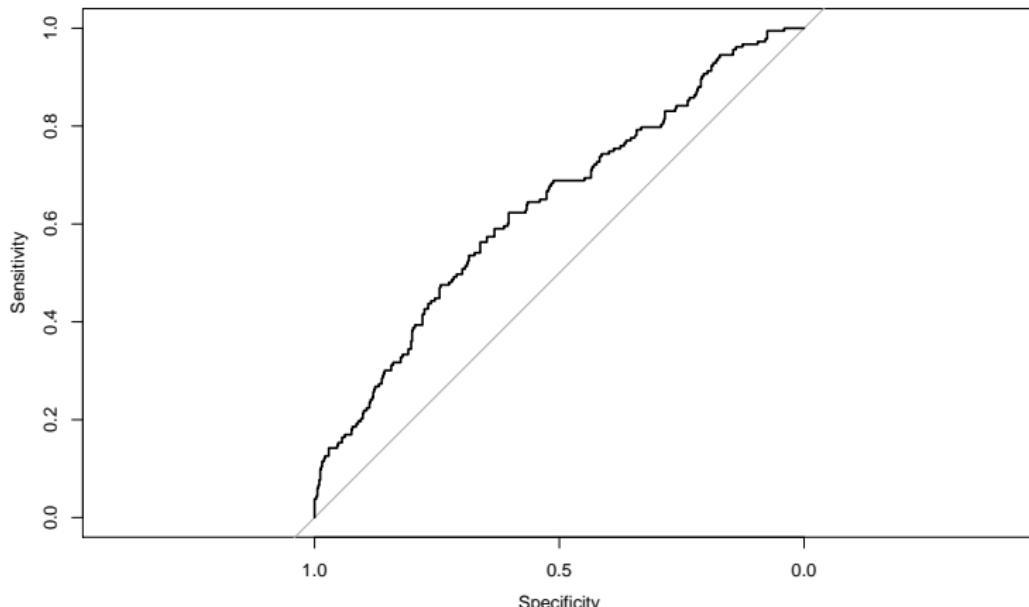
Vier Arten von Ergebnissen einer Klassifikation

Tabelle 6: Geläufige Kennwerte der Klassifikation

Name	Definition
Falsch-Positiv-Rate (FP-Rate)	FP/N
Richtig-Positiv-Rate (RP-Rate)	RP/N
Falsch-Negativ-Rate (FN-Rate)	FN/N
Richtig-Negativ-Rate (RN-Rate)	RN/N
Positiver Vorhersagewert	RP/P*
Negativer Vorhersagewert	RN/N*
Gesamtgenauigkeitsrate	(RP+RN) / (N+P)

ROC-Kurven

```
lets_roc <- roc(Aktien$Aktienkauf, glm1$fitted.values)  
plot(lets_roc)
```



Beispiele für ROC-Kurven

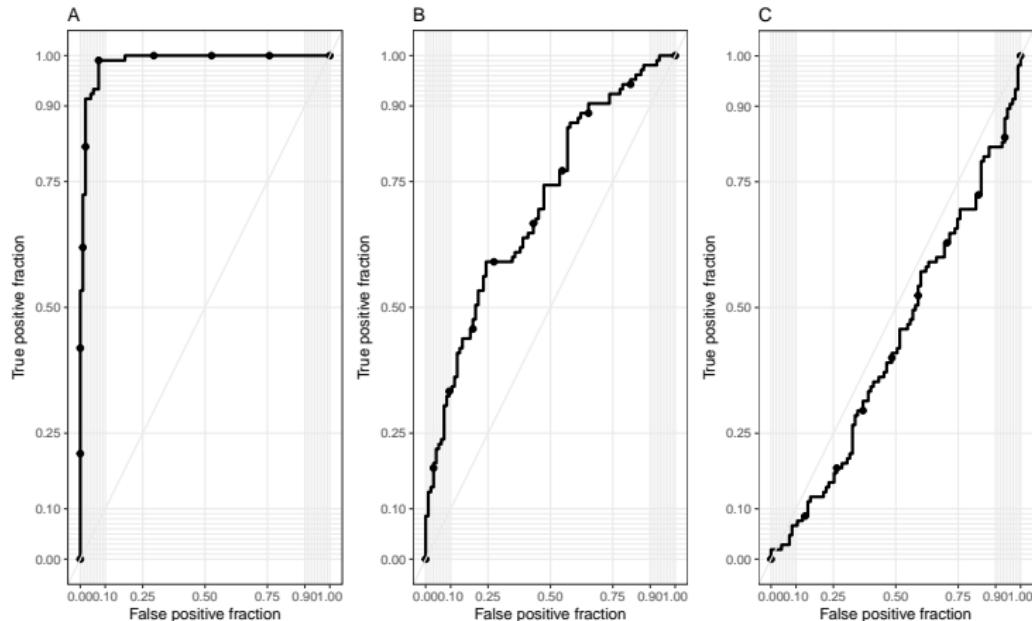


Abbildung 48: Beispiel für eine sehr gute (A), gute (B) und schlechte (C) Klassifikation

Clusteranalyse

Lernziele

- Das Ziel einer Clusteranalyse erläutern können.
- Das Konzept der euklidischen Abstände verstehen.
- Eine k-Means-Clusteranalyse berechnen und interpretieren können.

Clustern Sie diesen Datensatz!

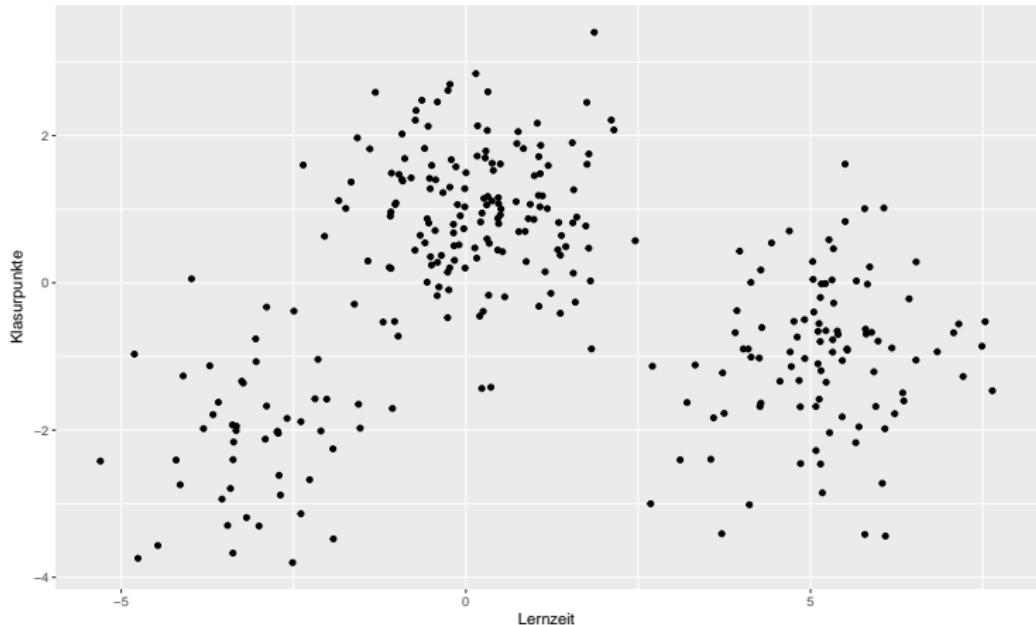


Abbildung 49: Ein Streudiagramm - sehen Sie Gruppen (Cluster) ?

Intuitive Darstellung der Clusteranalyse

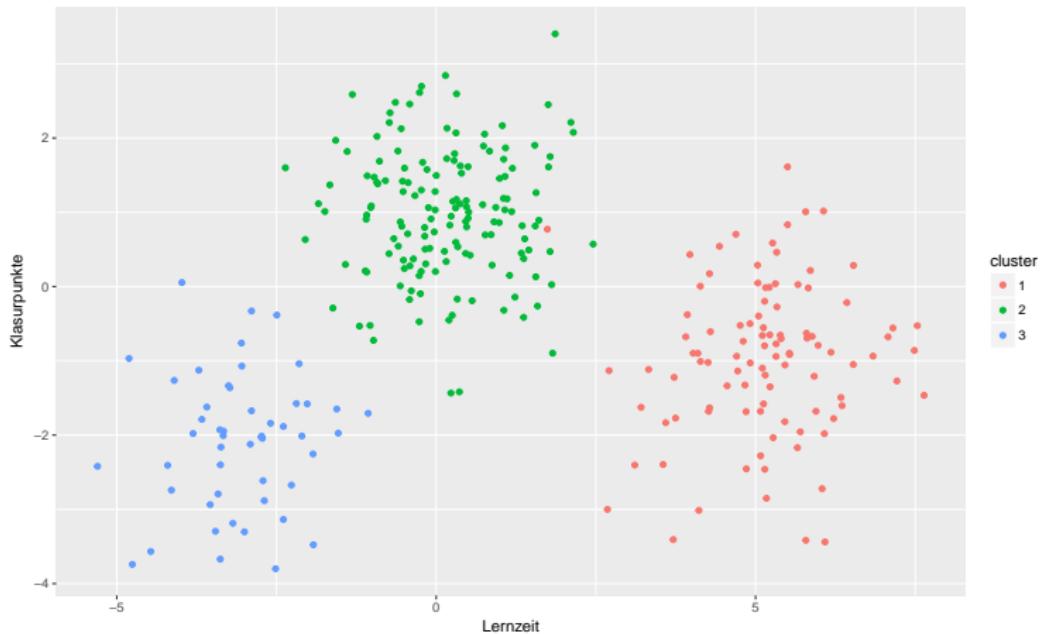


Abbildung 50: Ein Streudiagramm - mit drei Clustern

Unterschiedliche Anzahlen von Clustern im Vergleich

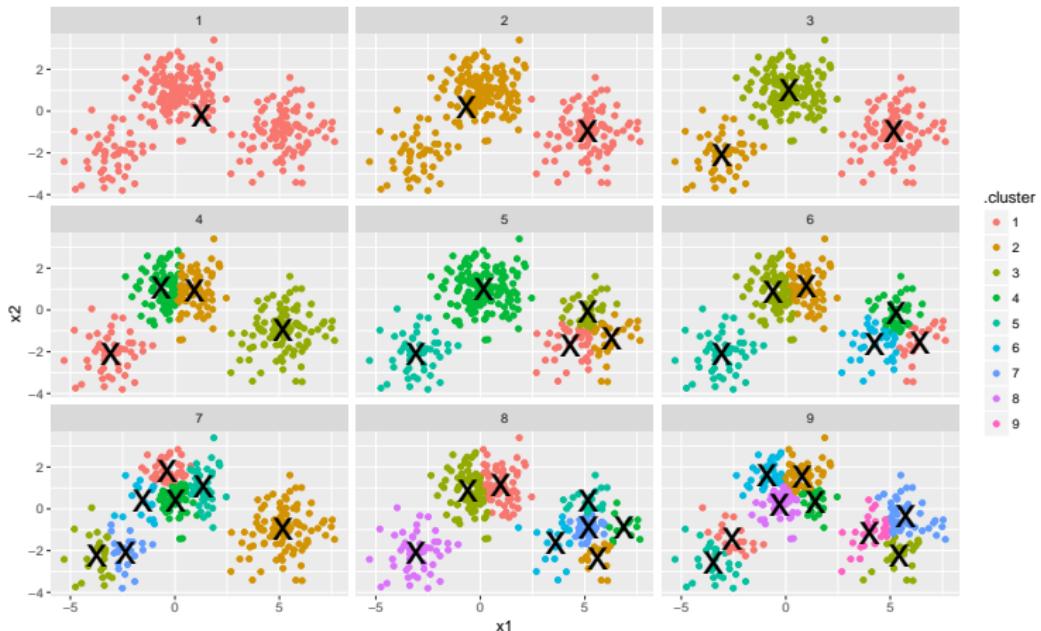


Abbildung 51: Unterschiedliche Anzahlen von Clustern im Vergleich

Wie groß ist der "Abstand" zwischen Anna und Berta?

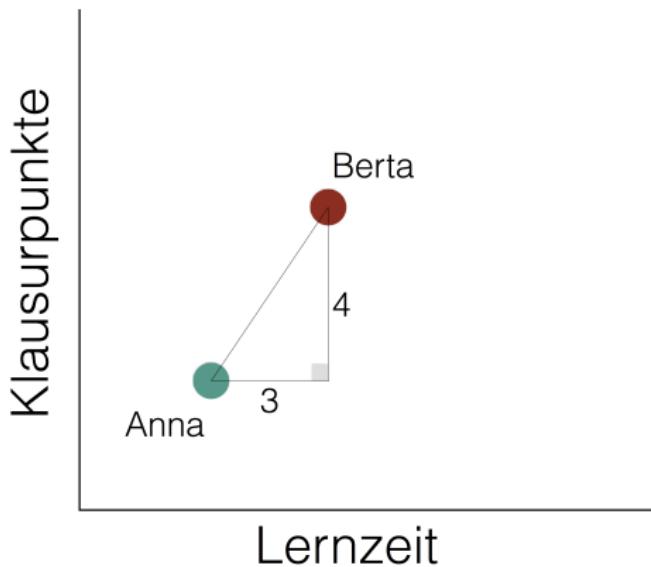


Abbildung 52: Distanz zwischen zwei Punkten in der Ebene

Pythagoras strikes back

$$c^2 = a^2 + b^2$$

In unserem Beispiel heißt das $c^2 = 3^2 + 4^2 = 25$. Folglich ist $\sqrt{c^2} = \sqrt{25} = 5$. Der Abstand oder der Unterschied zwischen Anna und Berta beträgt also 5 - diese Art von "Abstand" nennt man den *euklidischen Abstand*.

Pythagoras in 3D

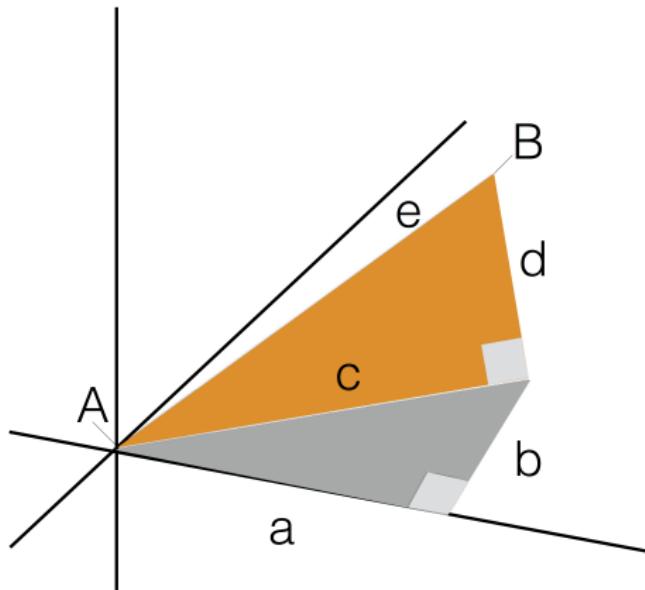


Abbildung 53: Pythagoras in 3D

Pythagoras in Reihe geschaltet

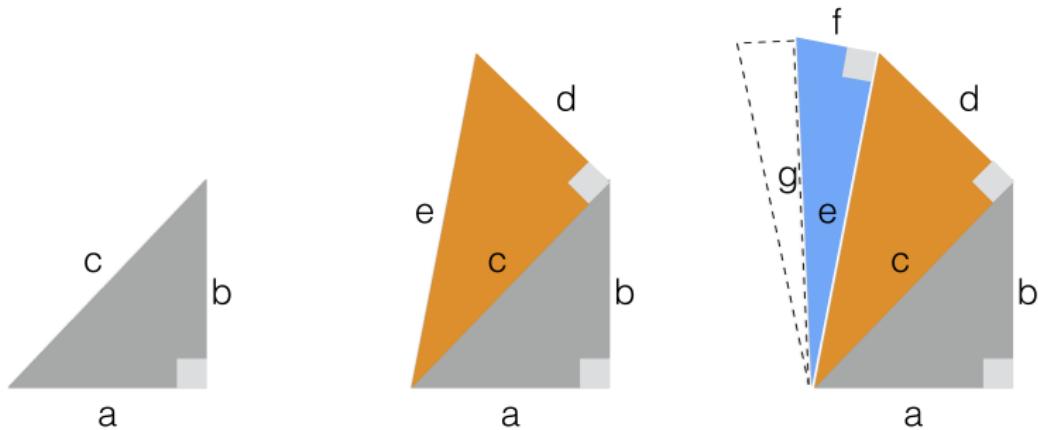
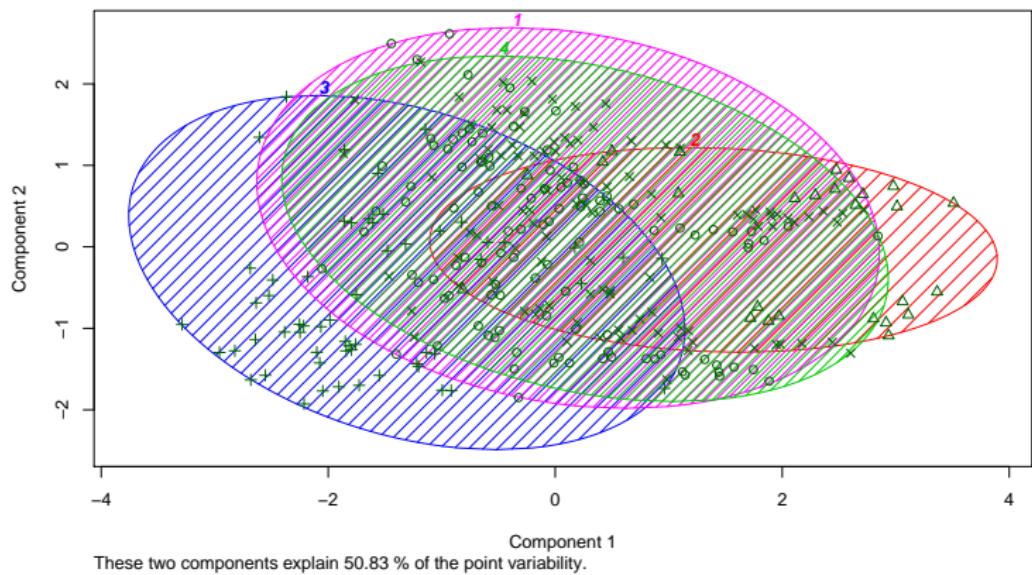


Abbildung 54: Pythagoras in Reihe geschaltet

k-Means Clusteranalyse

CLUSPLOT(segment.num)



Dimensionsreduktion

Lernziele

- Den Unterschied zwischen einer Hauptkomponentenanalyse und einer Exploratorische Faktorenanalyse kennen
- Methoden kennen, um die Anzahl von Dimensionen zu bestimmen
- Methoden der Visualisierung anwenden können
- Umsetzungsmethoden in R anwenden können
- Ergebnisse interpretieren können.

PCA vs. EFA

- Die *Hauptkomponentenanalyse* (engl. principal component analysis, PCA)
 - reduziert Daten
 - erklärt die Gesamtvarianz
- Die *Exploratorische Faktorenanalyse (EFA)*
 - führt manifeste Variablen (Items) auf latente Faktoren zurück
 - erklärt nicht die komplette Varianz, sondern nur die Varianz, die durch die vorhandenen Variablen erklärt wird

Nutzen der Dimensionsreduktion

- *Dimensionen reduzieren*
- *Unsicherheit verringern*
- *Aufwand verringern*

Intuition zur Dimensionsreduktion

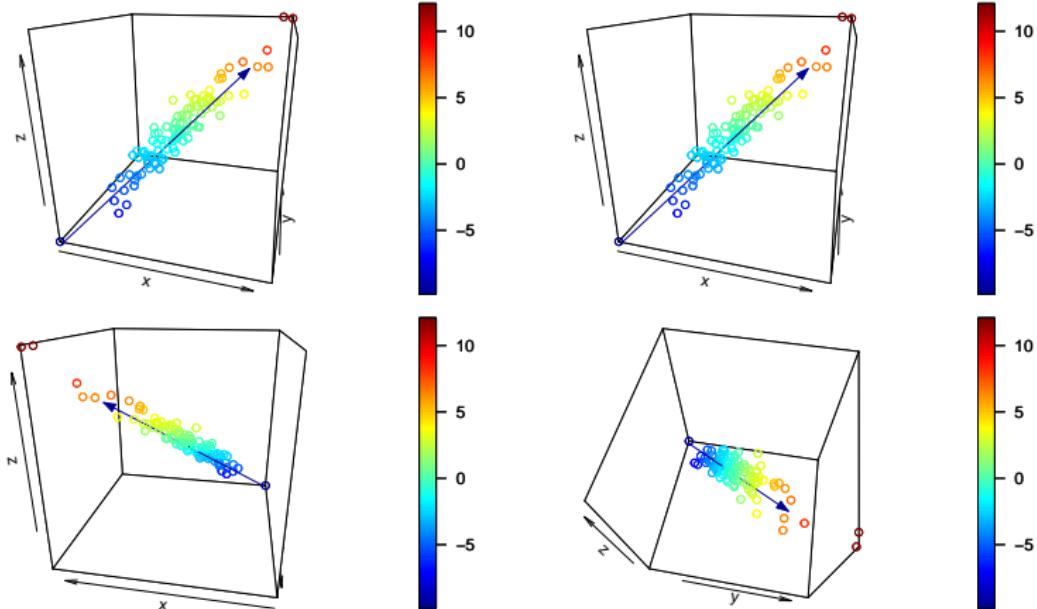


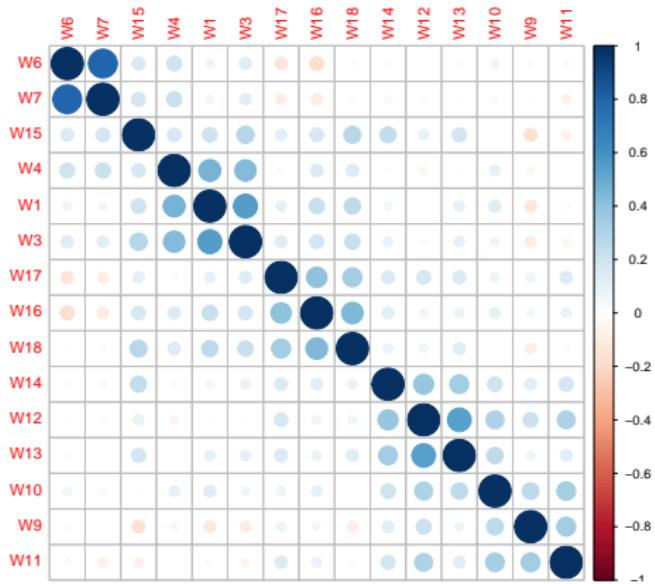
Abbildung 55: Der Pfeil ist eindimensional; reduziert also die drei Dimensionen auf eine

Datensatz Werte – z-transformiert

W1	W3	W4	W6	W7
0.5149008	0.5961663	1.6629148	1.3261145	1.2935031
-1.4990528	1.2759455	-0.6949935	-1.4505072	-1.5356813
-2.1703707	1.2759455	-0.6949935	1.3261145	1.2935031
1.1862187	1.2759455	1.6629148	0.2154658	-0.4040075
-1.4990528	-0.7633920	-0.6949935	0.2154658	0.1618293

Korrelationsplot

```
corrplot::corrplot(cor(Werte.sc), order = "hclust")
```

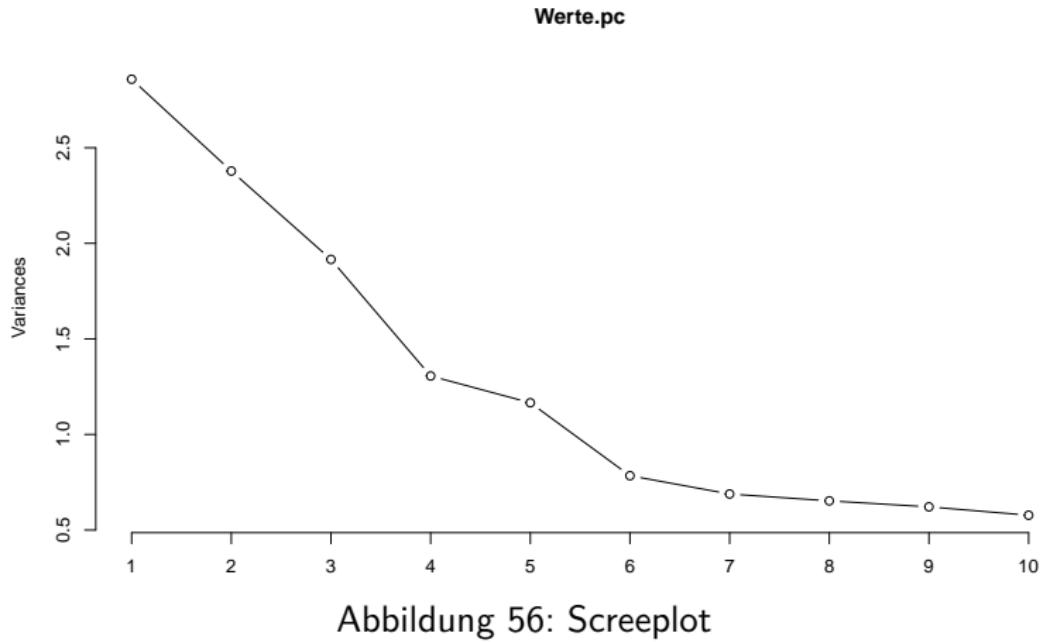


PCA berechnen

```
Werte.pc <- prcomp(Werte.sc) # Principal Components berechnen  
summary(Werte.pc)  
  
Gesamtvarianz <- sum(Werte.pc$sdev^2)  
  
# Varianzanteil der ersten Hauptkomponente  
Werte.pc$sdev[1]^2/Gesamtvarianz
```

Scree-Plot

```
plot(Werte.pc, type = "l")
```



Eigenwert-Kriterium

Der *Eigenwert* ist eine Metrik für den Anteil der erklärten Varianz pro Hauptkomponente.

```
eigen(cor(Werte))
```

Laut dem Eigenwert-Kriterium sollen nur Faktoren mit einem *Eigenwert größer 1* extrahiert werden.

Screeplot

VSS.scree(Werte)

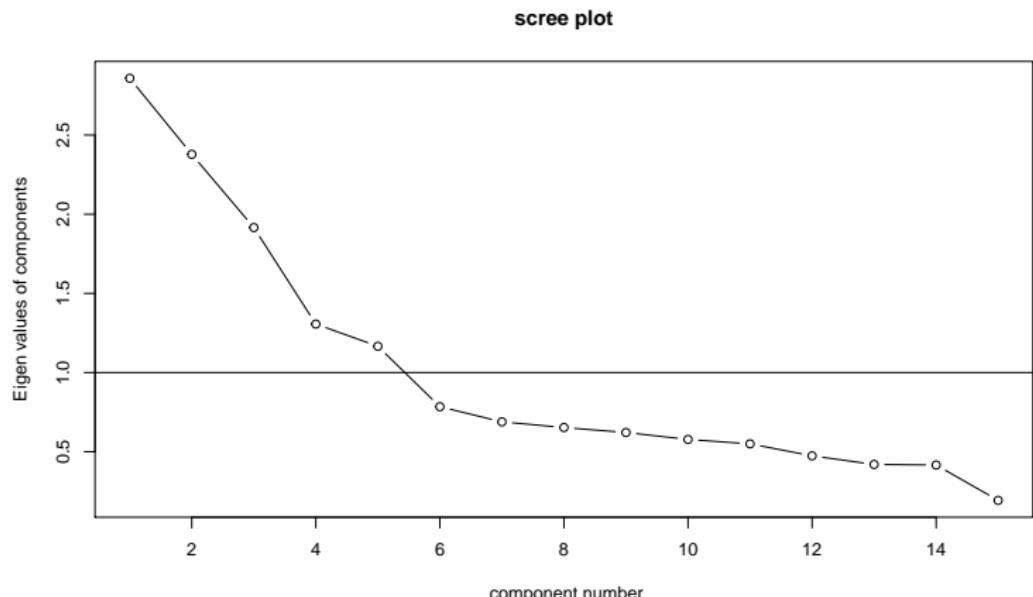
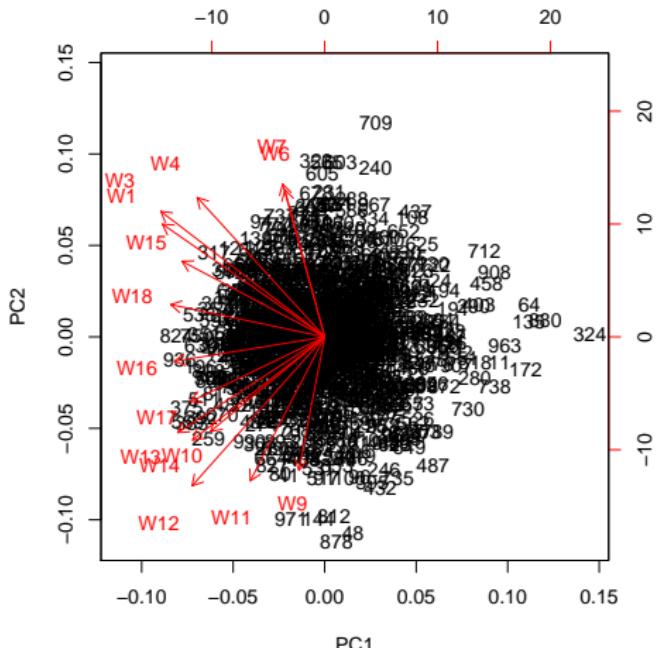


Abbildung 57: VSS-Screepplot
Folien für das Modul 'Praxis der Datendanalyse'

Biplot

```
biplot(Werte.pc)
```



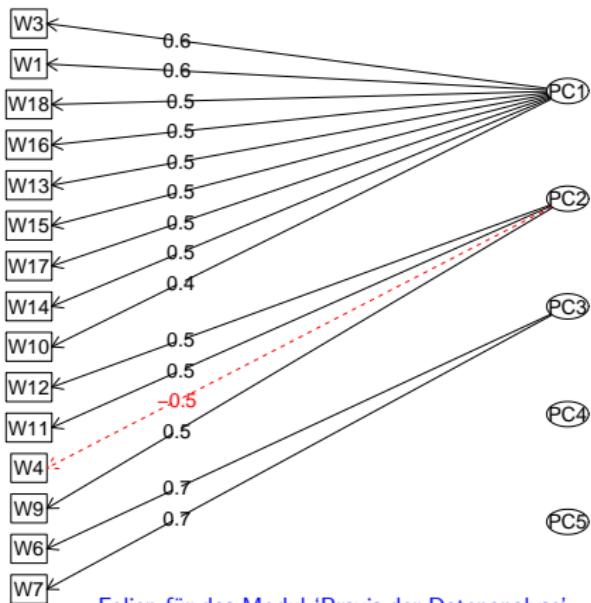
Ladungen der Items auf die Hauptkomponenten

```
Werte.pca <- principal(Werte, nfactors = 5, rotate = "none")
```

Pfaddiagramm der Ladungen auf die Hauptkomponenten

```
fa.diagram(Werte.pca)
```

Factor Analysis



Rotation

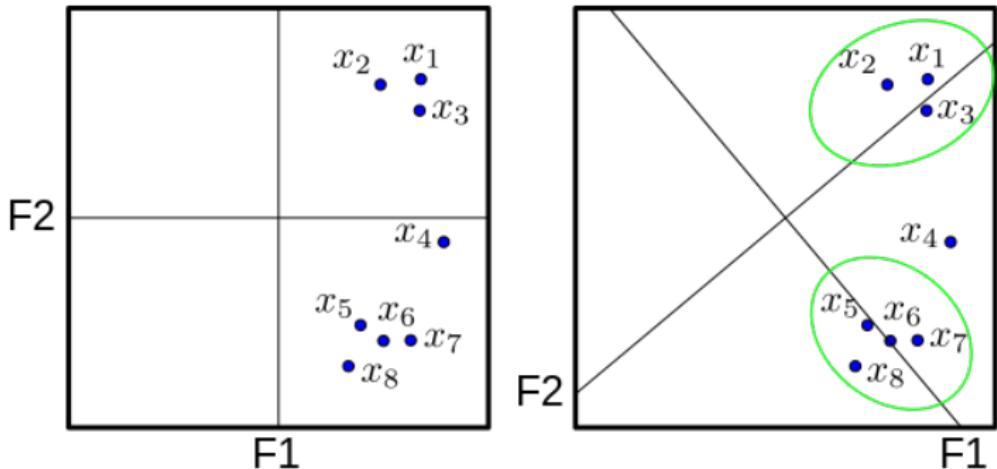


Abbildung 58: Beispiel für eine rechtwinklige Rotation

Heatmap

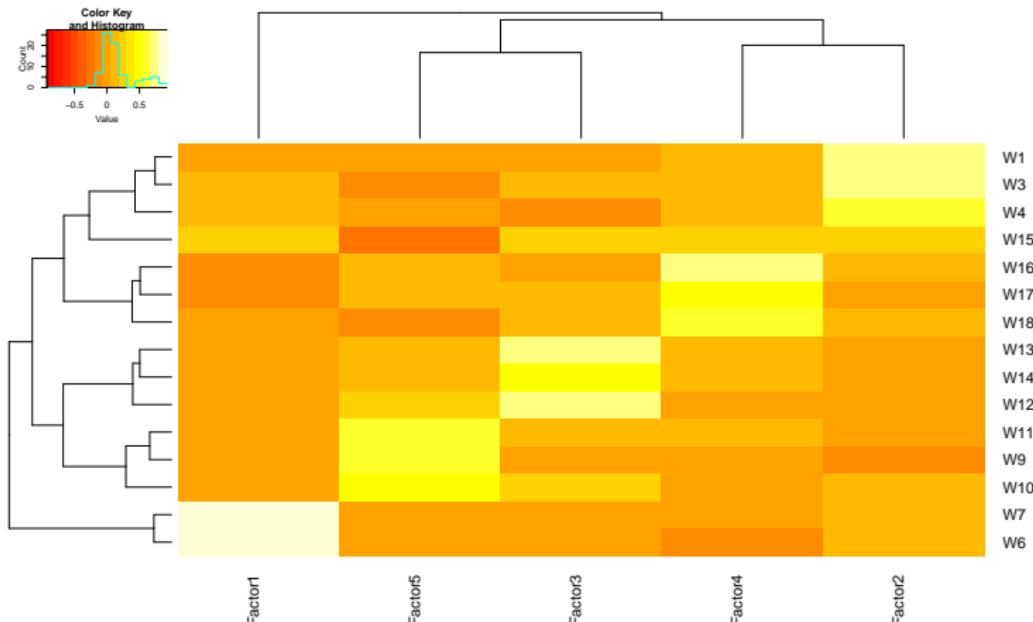


Abbildung 59: Heatmap einer EFA

Faktorwerte

```
Werte.ob <- factanal(Werte, factors = 5, scores = "Bartlett")
```

Interne Konsistenz der Skalen

Inhaltlich ist Alpha eine Art mittlere Korrelation, die sich ergibt wenn man alle Items (paarweise) miteinander korriert: I1-I2, I1-I3,...

```
alpha(Werte[, c("W12", "W13", "W14", "W15")], check.keys = TRUE)
```

Faustregeln zur Höhe von Cronbachs Alpha

Alpha	Bedeutung
größer 0,9	exzellent
größer 0,8	gut
größer 0,7	akzeptabel
größer 0,6	fragwürdig
größer 0,5	schlecht

Textmining

Lernziele

- Sie kennen zentrale Ziele und Begriffe des Textminings.
- Sie wissen, was ein 'tidy text dataframe' ist.
- Sie können Worthäufigkeiten auszählen.
- Sie können Worthäufigkeiten anhand einer Wordcloud visualisieren.

Zentrale Begriffe

- Ein *Corpus* bezeichnet die Menge der zu analysierenden Dokumente-
- Ein *Token (Term)* ist ein elementarer Baustein eines Texts, die kleinste Analyseeinheit, häufig ein Wort.
- Unter *tidy text* versteht man einen Dataframe, in dem pro Zeile nur ein Term steht.

Tidytext – Input

```
text <- c("Wir haben die Frauen zu Bett gebracht,", "als die M  
"Wir hatten uns das viel schöner gedacht.", "Wir waren nur  
text_df <- data_frame(Zeile = 1:4, text = text)
```

Tidytext – Output

```
tidytext_df <- text_df %>% unnest_tokens(output = wort, input  
  
tidytext_df %>% head  
  
## # A tibble: 6 x 2  
##   Zeile     wort  
##   <int>    <chr>  
## 1       1     wir  
## 2       1     haben  
## 3       1     die  
## 4       1     frauen  
## 5       1     zu  
## 6       1    bett
```

In einem 'tidy text Dataframe' steht in jeder Zeile ein Wort

Folien für das Modul 'Praxis der Datenanalyse'

WS17

137 / 140

Worthäufigkeiten auszählen

```
afd_df %>%  
  na.omit() %>%  # fehlende Werte löschen  
  count(token, sort = TRUE) %>%  
  head
```

```
## # A tibble: 6 x 2  
##   token     n  
##   <chr> <int>  
## 1 die    1151  
## 2 und    1147  
## 3 der    870  
## 4 zu     435  
## 5 für    392  
## 6 in     392
```

Stopwörter entfernen

token	n
deutschland	190
afd	171
programm	80
wollen	67
bürger	57

Anhang