

Folien für das Modul ‘Praxis der Datenanalyse’

FOM – WS17

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

Vorwort

- ▶ 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

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.

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

- ▶ 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 **einigen Ausnahmen**

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

Klausur

- ▶ Hinweise zur Klausur finden Sie [hier](#).
- ▶ Im Skript finden Sie eine [Probeklausur](#).
- ▶ Lernaufgaben finden sich im Skript am Ende jedes Kapitels.

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](#) (im OC).

Rahmen

- ▶ 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.

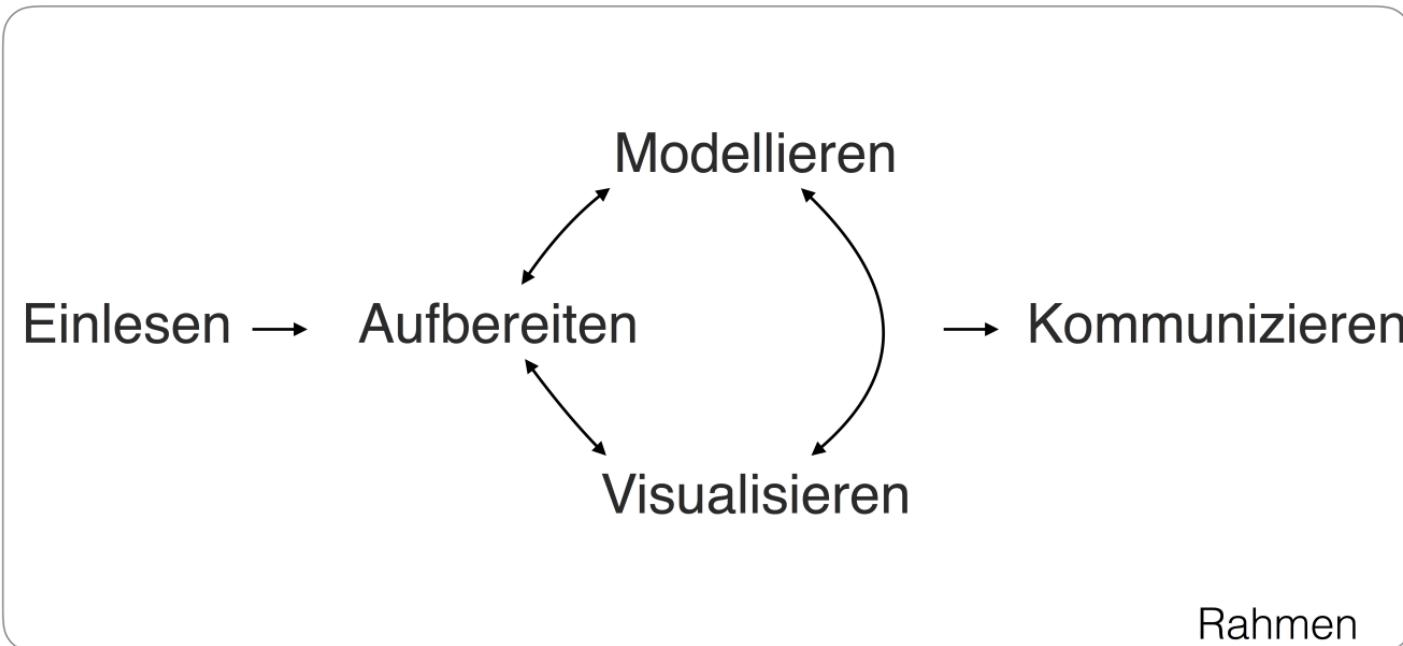


Abbildung 1: Der Prozess der Datenanalyse

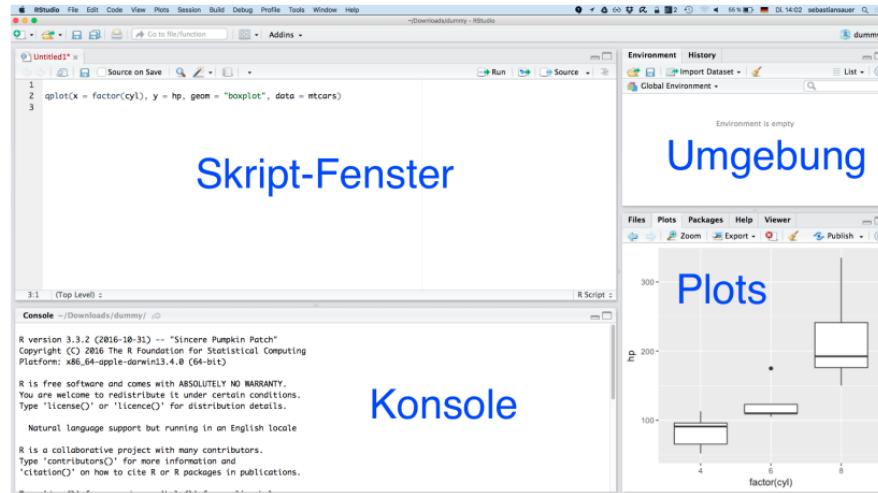


Abbildung 2: RStudio

Bitte laden Sie sich diesen Ordner [Github-Repositorium](#) herunter. Dazu klicken Sie auf den grünen Button "Clone or Download", wählen Sie dann "Download Zip". Daraufhin wird dieser Ordner heruntergeladen.

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`

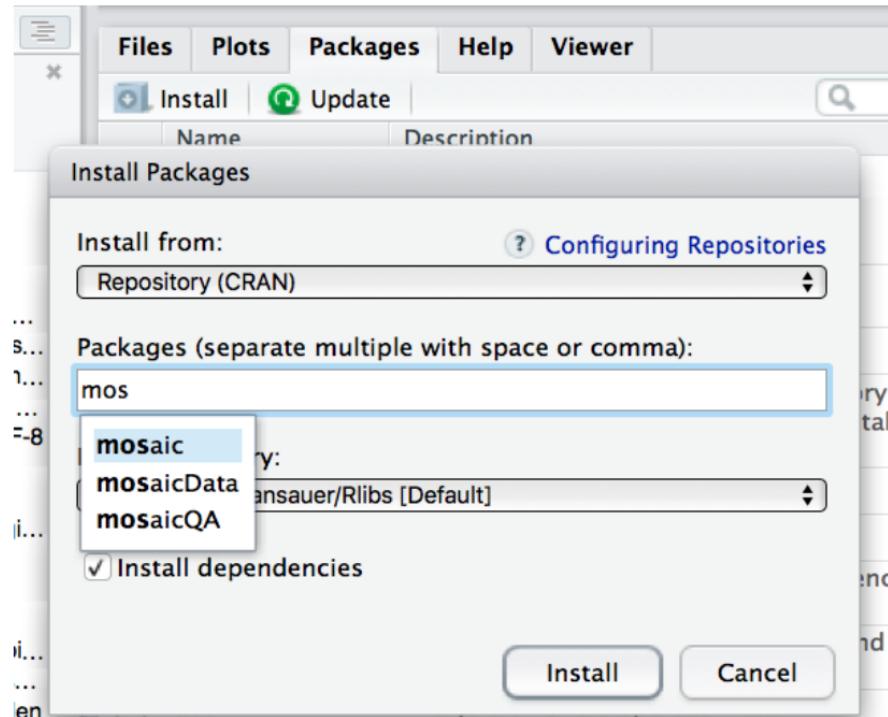
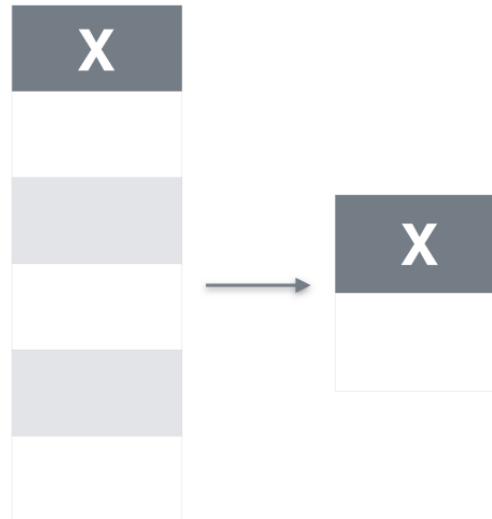


Abbildung 3: So installiert man Pakete in RStudio

Alle Datensätze liegen im Ordner `data/`, den Sie vom [Github-Repositorium](#) herunterladen können.

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.

Deskriptivstatistik



Inferenzstatistik

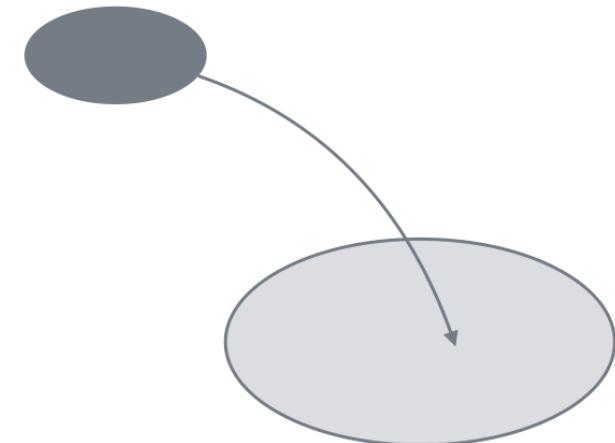


Abbildung 4: Sinnbild für die Deskriptiv- und die Inferenzstatistik

Prämissse 1: Wenn Modell M wahr ist,
dann sollten die Daten das Muster D aufweisen.
Prämissse 2: Die Daten weisen das Muster D auf.

Konklusion: Daher muss das Modell M wahr sein.

Die Konklusion ist *nicht* zwangsläufig richtig.

Daten einlesen

4. 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?”.

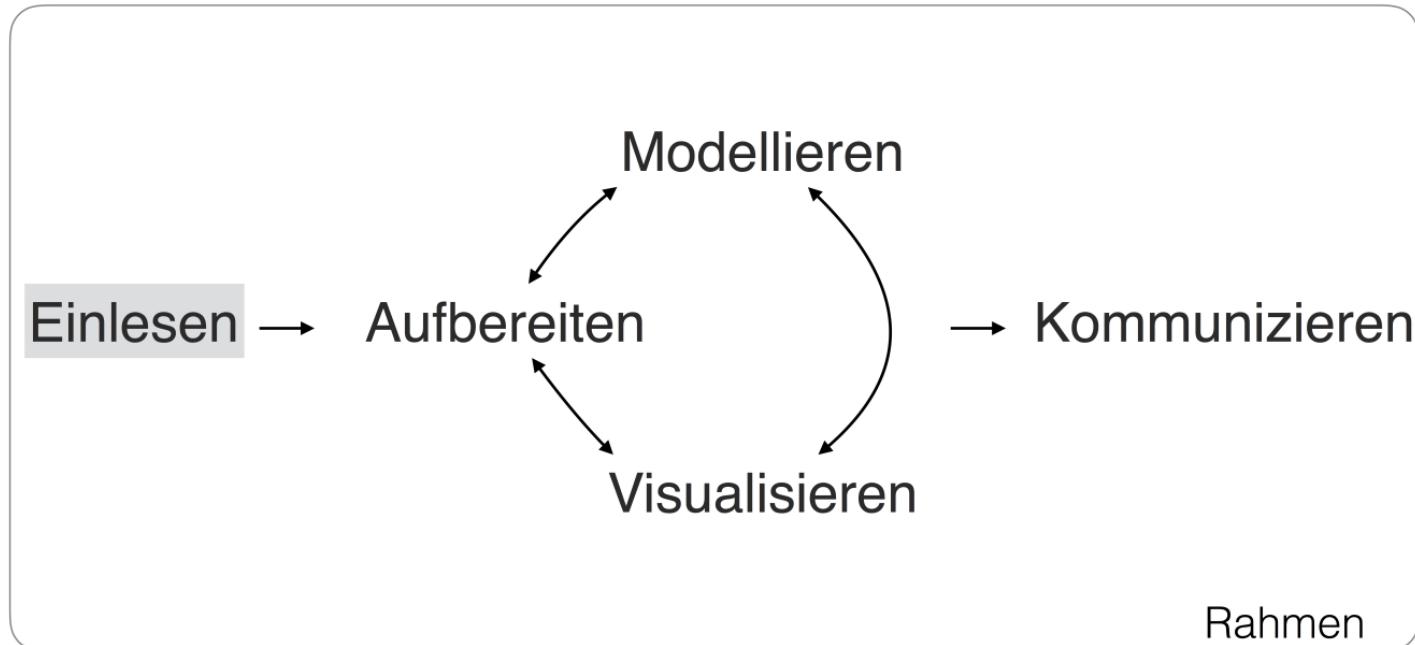


Abbildung 5: Daten sauber einlesen

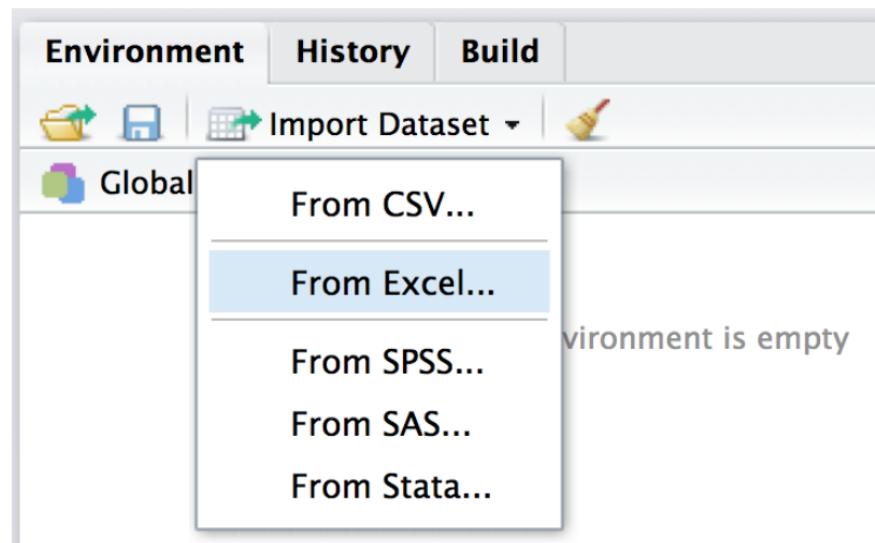


Abbildung 6: Daten einlesen (importieren) mit RStudio

```
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
```

4. Daten einlesen

Das Arbeitsverzeichnis mit RStudio wählen

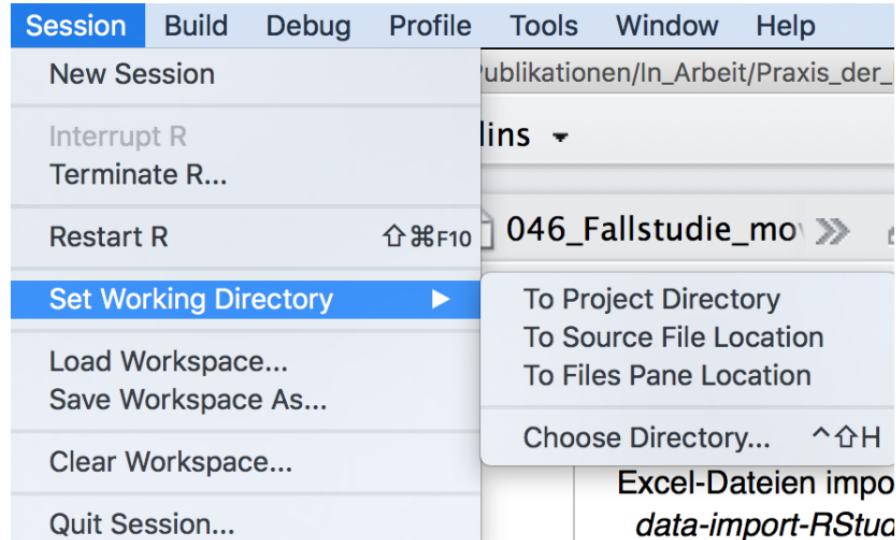


Abbildung 7: Das Arbeitsverzeichnis mit RStudio auswählen

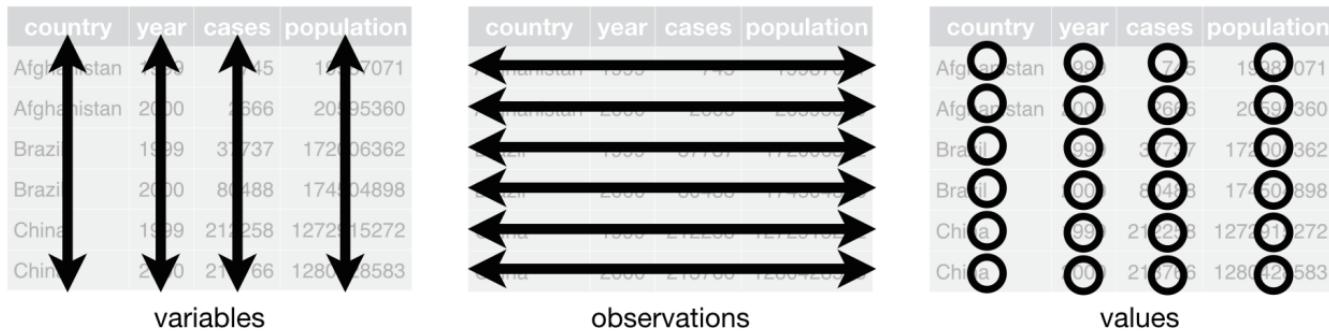


Abbildung 8: Schematische Darstellung eines Dataframes in Normalform

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

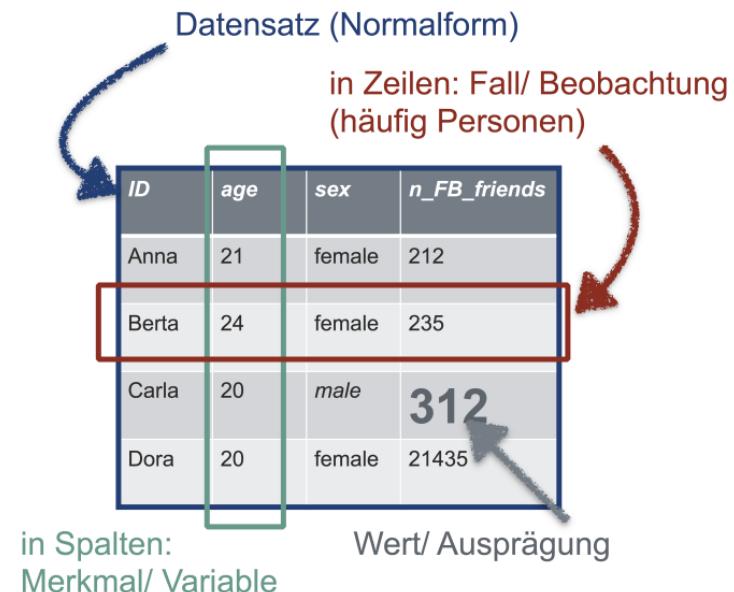


Abbildung 10: Illustration eines Datensatzes in Normalform

4. Daten einlesen

Tabelle in Normalform bringen

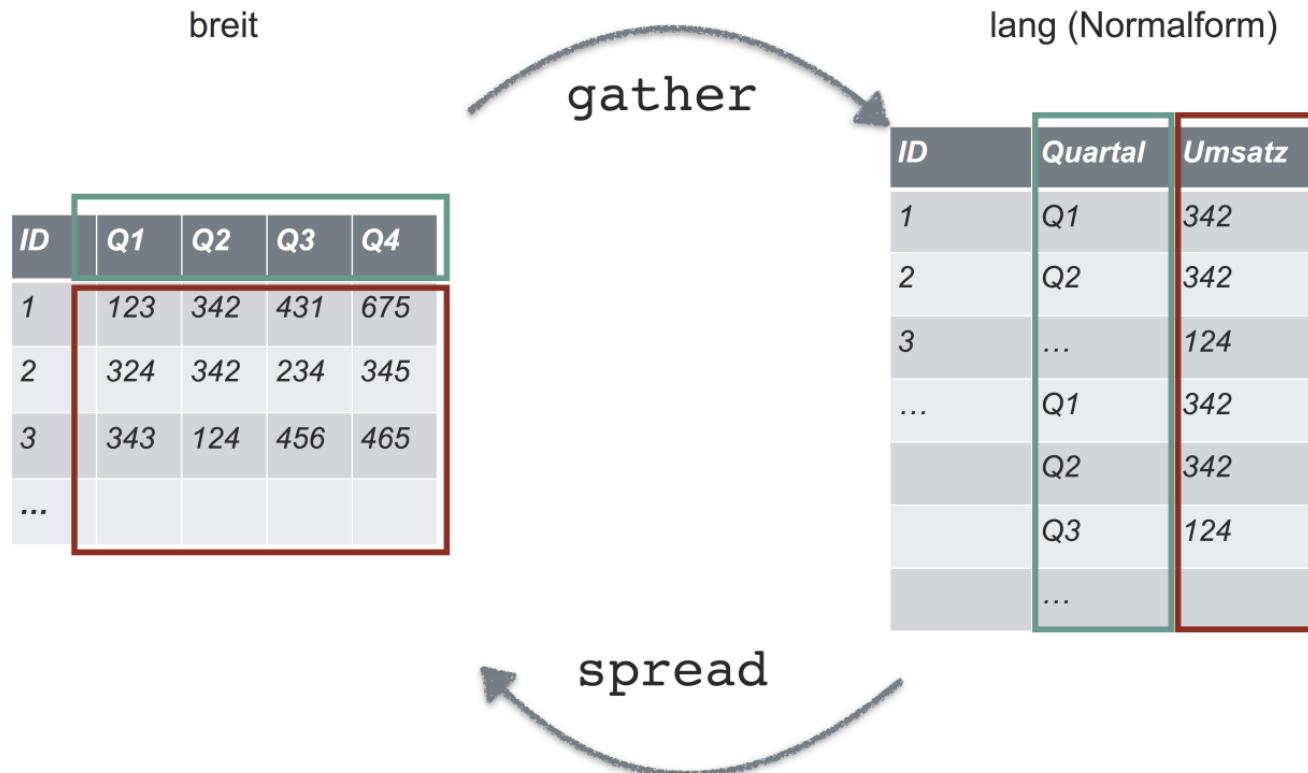


Abbildung 11: Mit 'gather' und 'spread' wechselt man von der breiten Form zur langen Form

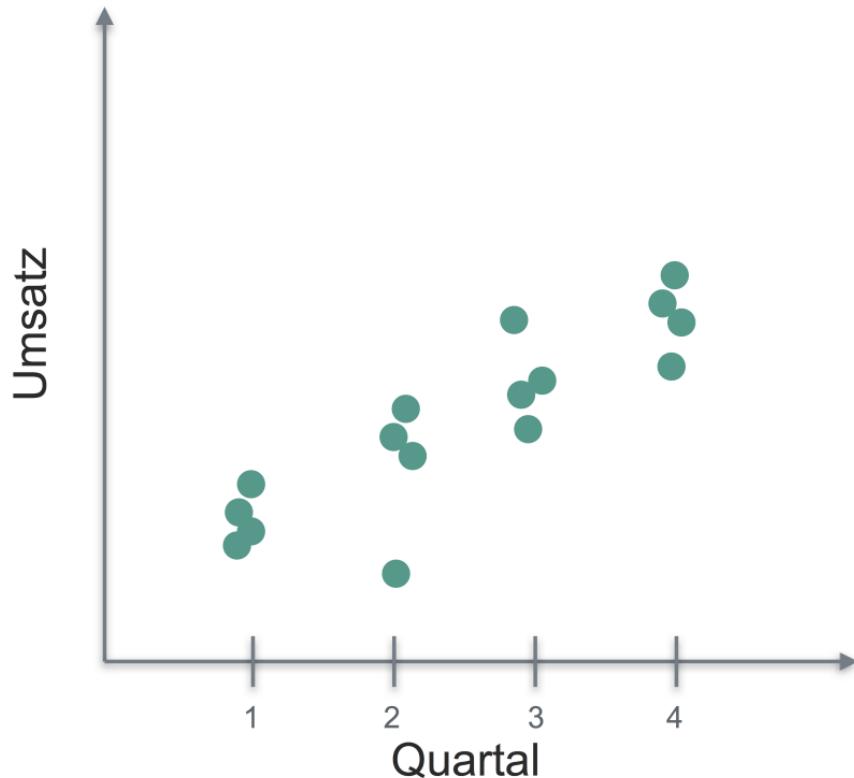


Abbildung 12: Ein Beispiel für eine Abbildung zu einer Normalform-Tabelle

4. Daten einlesen

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", -ID)
```

4. Daten einlesen

Textkodierung und Daten exportieren

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

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

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.

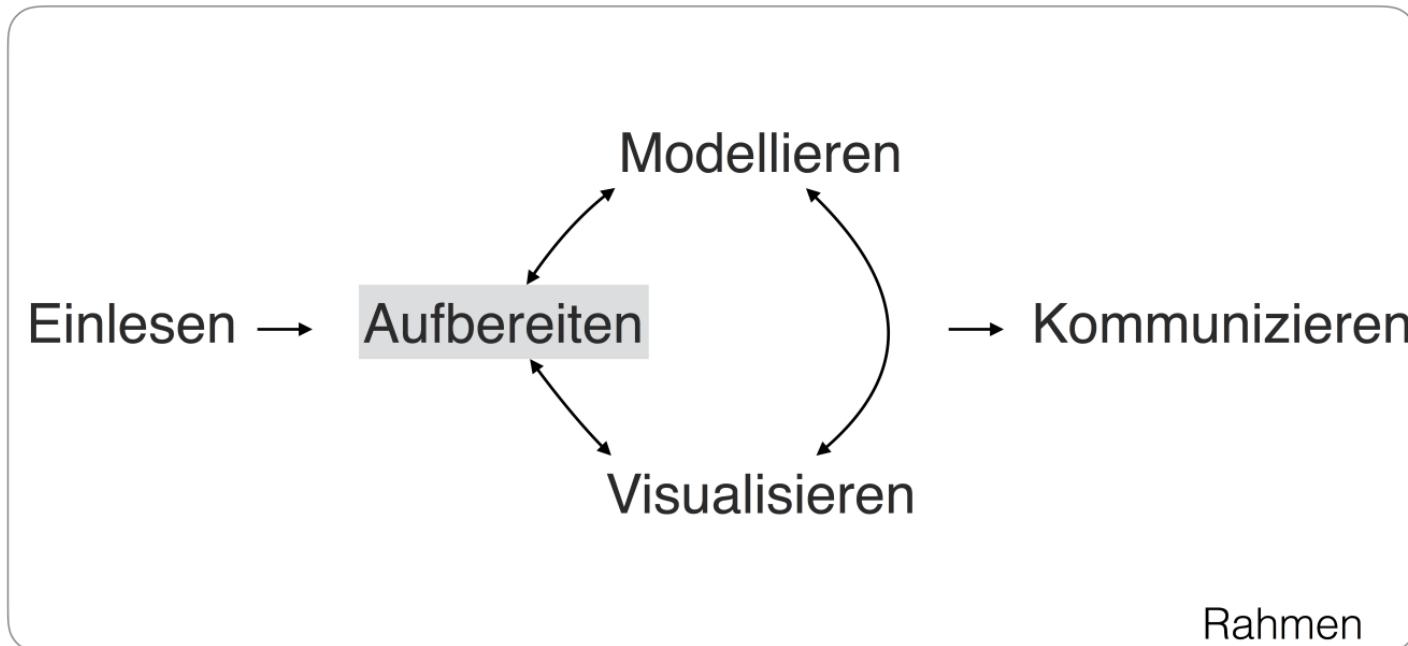


Abbildung 13: Daten aufbereiten

Typische Probleme, die immer wieder auftreten, sind:

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



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

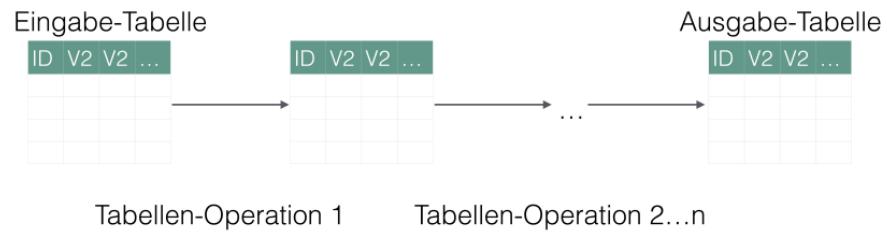


Abbildung 15: Durchpfeifen: Ein Dataframe wird von Operation zu Operation weitergereicht

The diagram illustrates the process of filtering rows from a full dataset to a subset. On the left, a full table contains five rows of data with columns ID, Name, and Note1. The rows are numbered 1 through 5. Rows 1 and 2 have Note1 values of 1, while rows 3, 4, and 5 have Note1 values of 2. An arrow points from the full table to a smaller table on the right, which only contains the two rows where Note1 is 1. These two rows correspond to the first two rows of the original table.

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

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

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

Right Table (Sorted Data):

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

Abbildung 18: Spalten sortieren

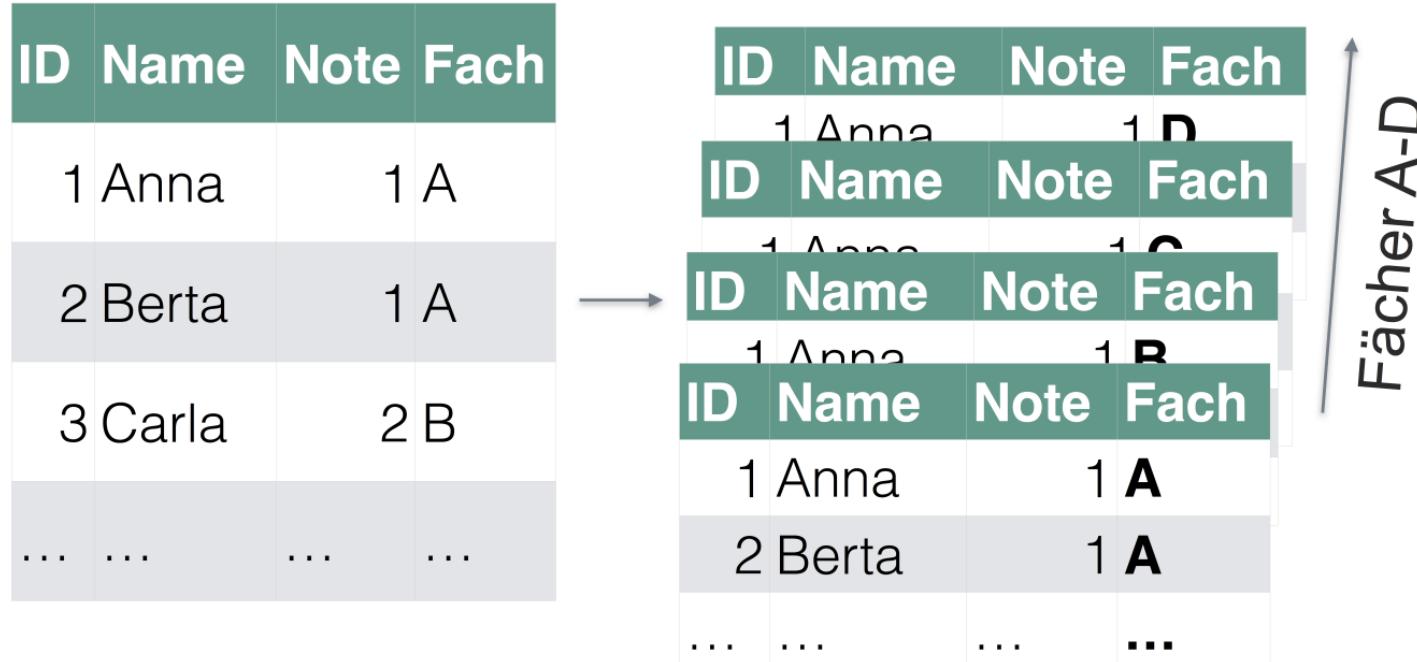


Abbildung 19: Datensätze nach Subgruppen aufteilen

Note
1
1
2
4
...

Mittelwert



Note
2

Gruppe A Gruppe B Gruppe C

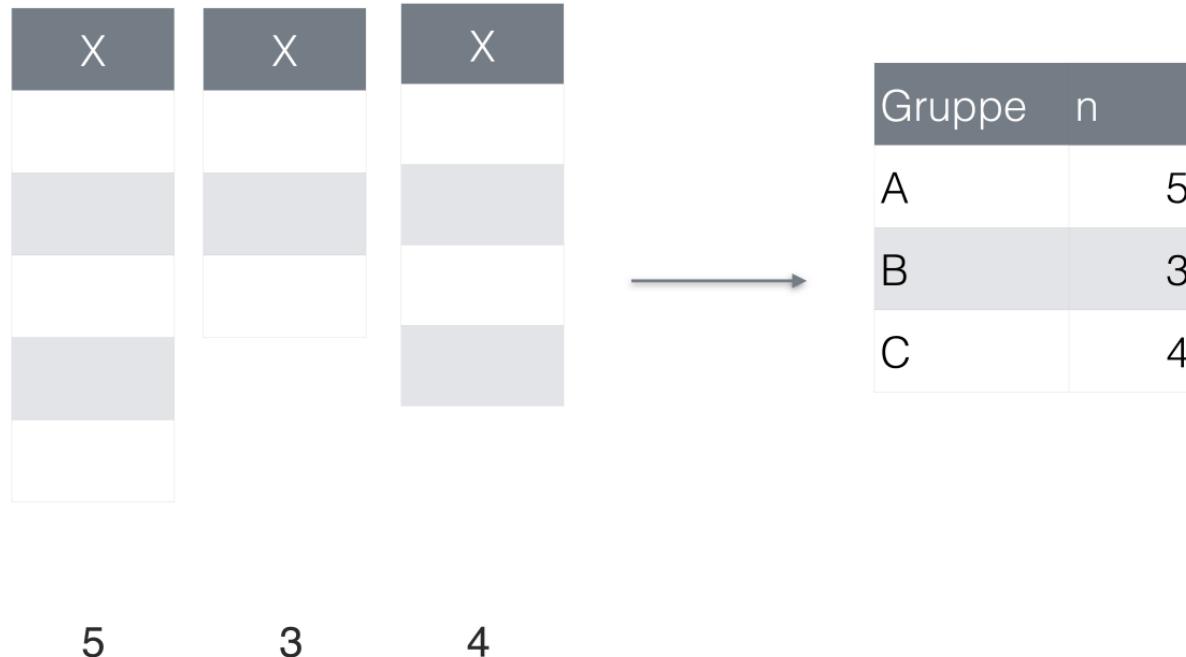


Abbildung 21: Sinnbild für 'count'



Abbildung 22: Das ist keine Pfeife

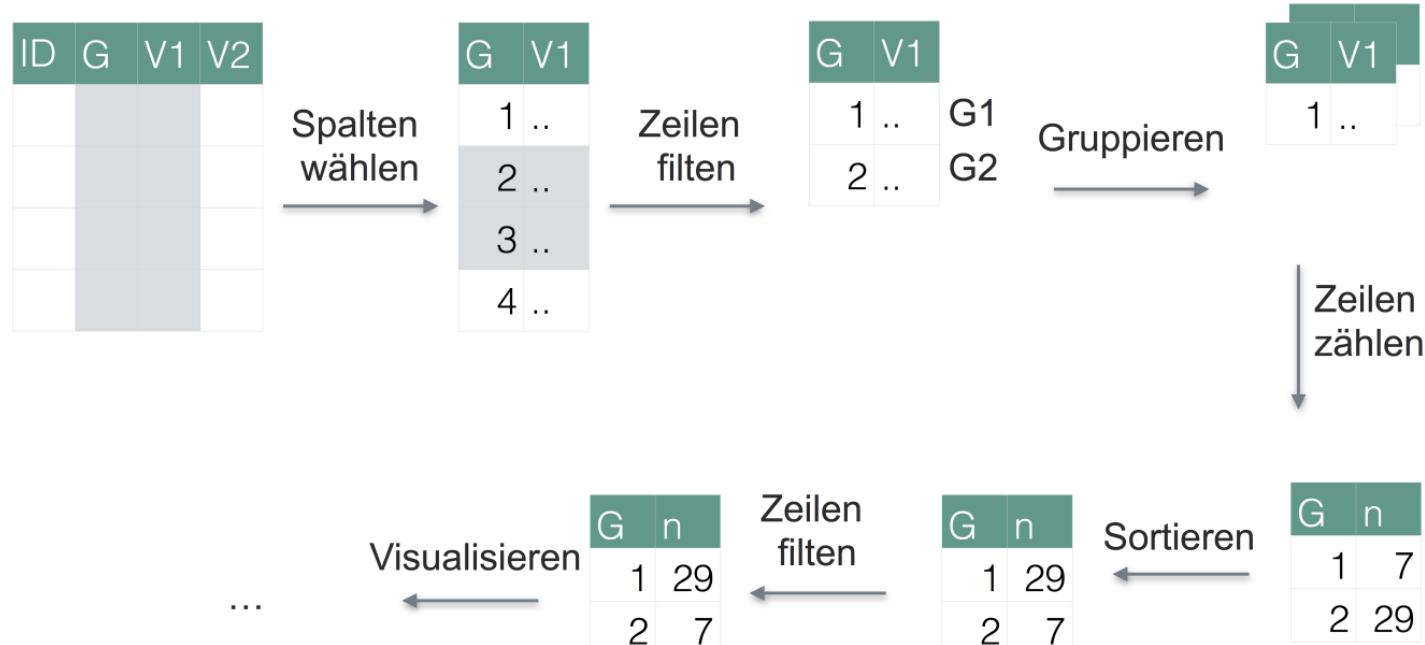


Abbildung 23: Das 'Durchpeifen'

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)
```

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.

Sinnbild

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



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 24: Sinnbild für mutate

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

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

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.

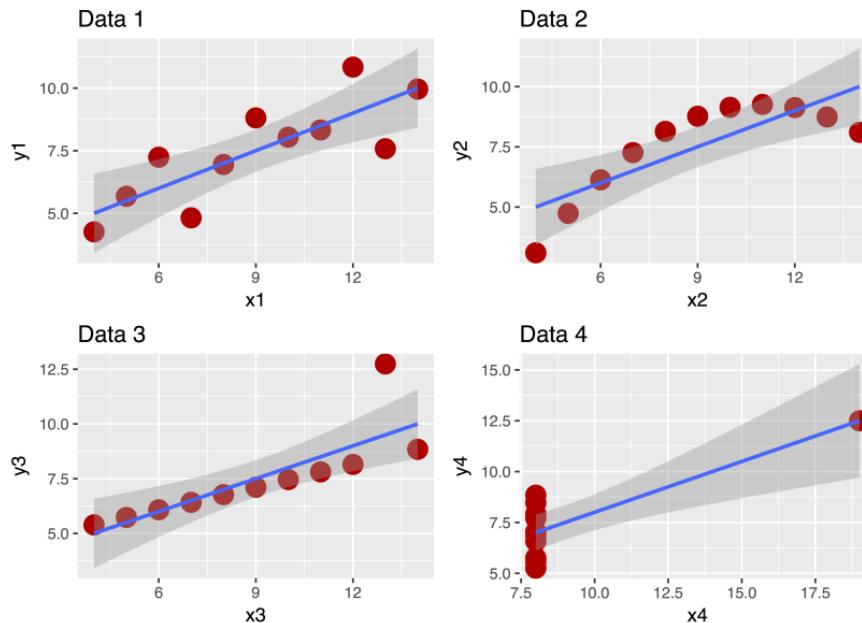


Abbildung 25: Das Anscombe-Quartett

Dinosaurier-Video

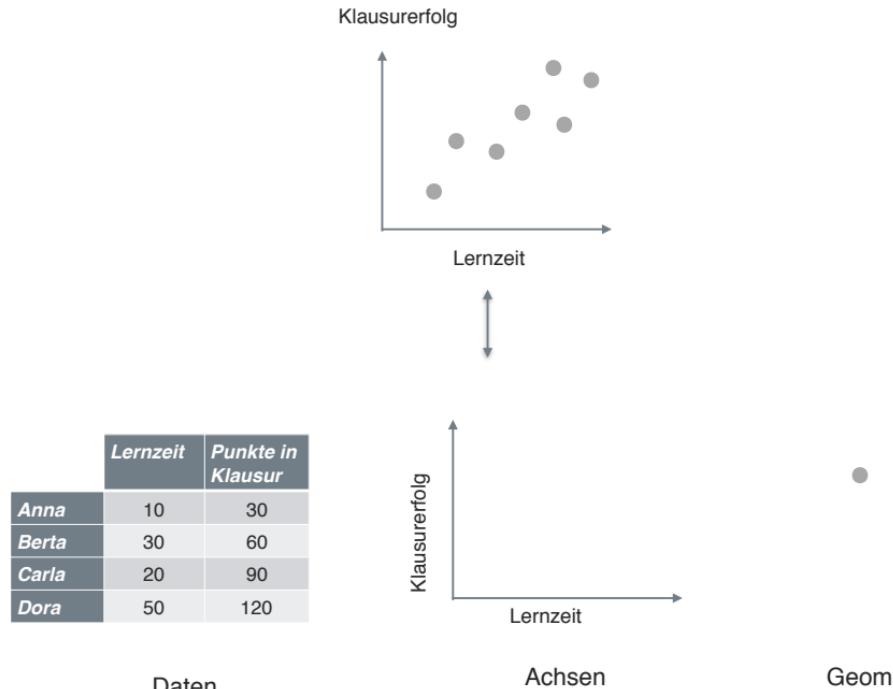


Abbildung 26: Anatomie eines Diagramms

6. Daten visualisieren

Beispiel für ein Diagramm mit `ggplot2::qplot`

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

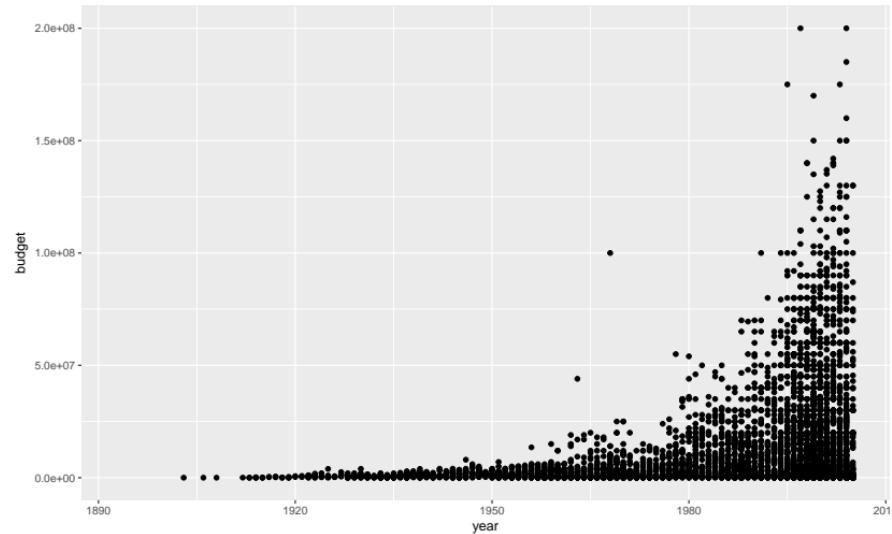


Abbildung 27: Mittleres Budget pro Jahr

- ▶ 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.

Diese Syntax des letzten Beispiels ist recht einfach, nämlich:

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

s. Skript

- ▶ Histogramm, Dichtediagramm
- ▶ Punkte, Schachbrett-Diagramme
- ▶ Balkendiagramm
- ▶ Mosaicplot (Fliesenbild)
- ▶ Punktediagramm für Zusammenfassungen
- ▶ Boxplots

Grundlagen des Modellierens

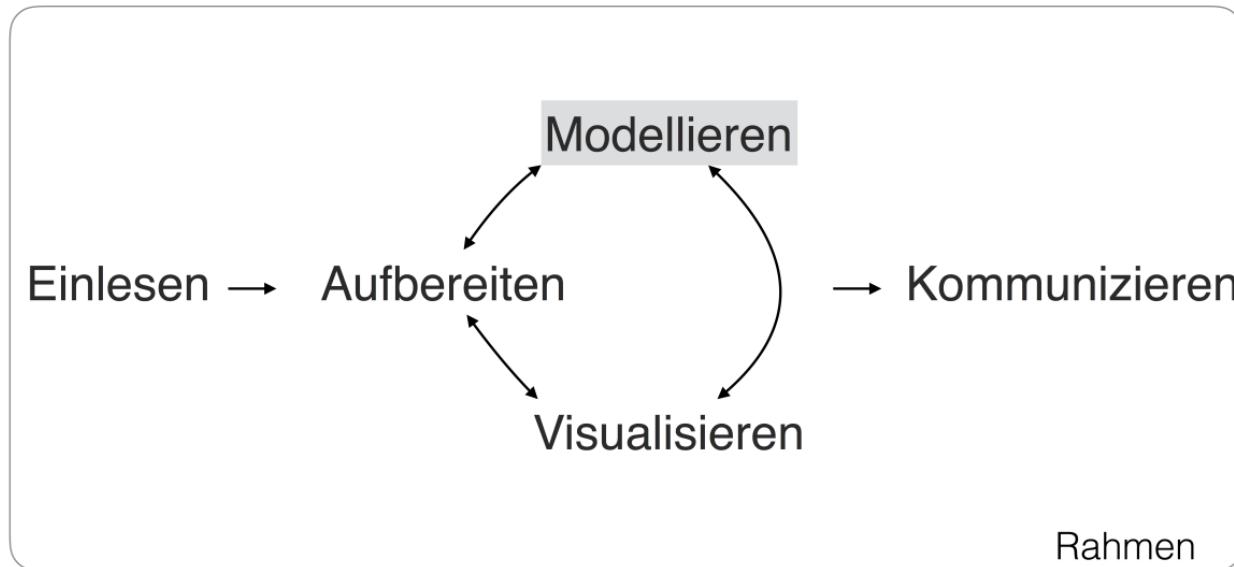




Abbildung 28: Modell eines VW-Käfers

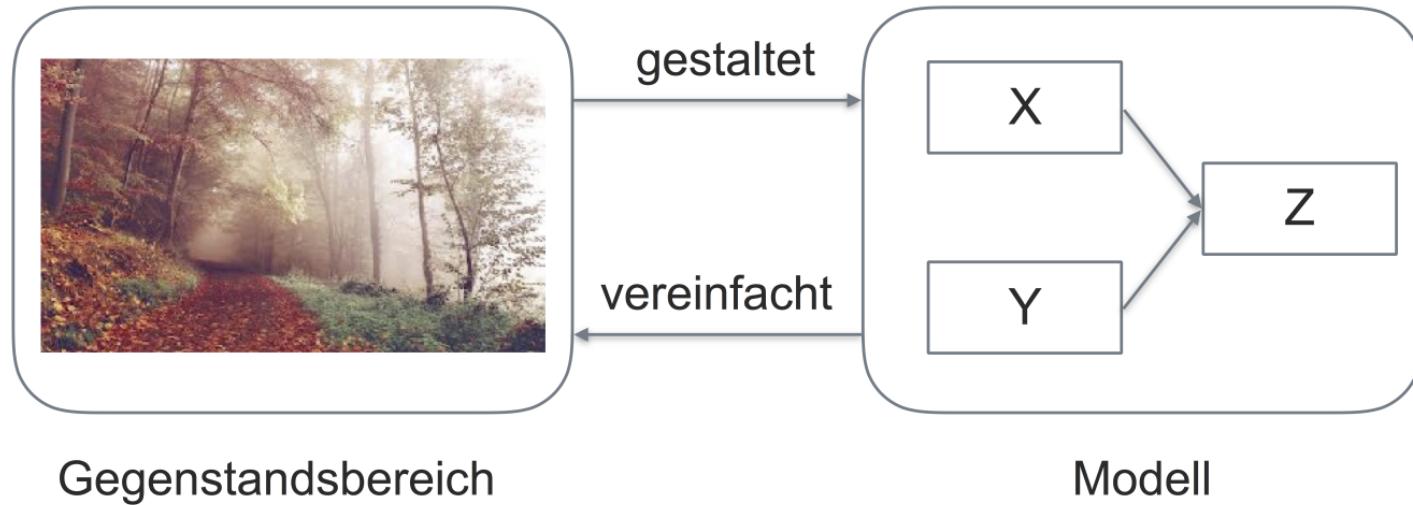
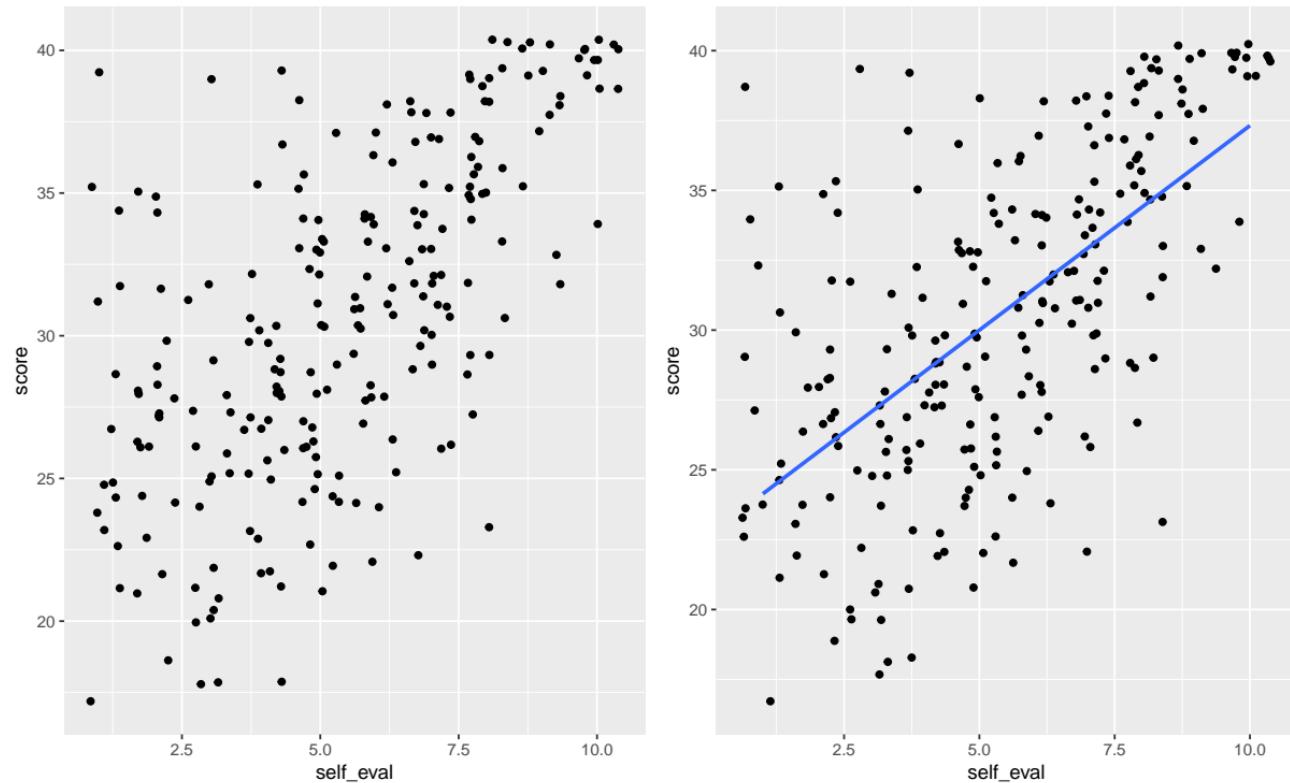


Abbildung 29: Modellieren

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

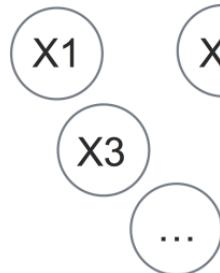


Abbildung 30: Formaleres Modell des Modellierens

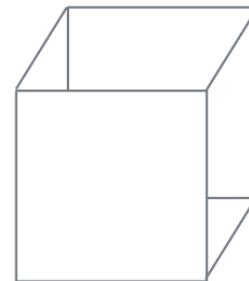


Die blaue Gerade ist ein Modell für den Datensatz (sie versucht es zumindest).

Einflussgrößen



Schwarze Kiste



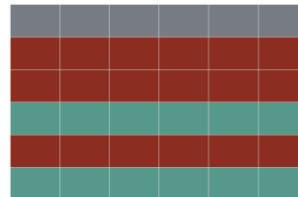
Vorhersage



Abbildung 31: Modelle mit schwarzer Kiste

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

Fallreduzierendes Modellieren



Dimensionsreduzierendes Modellieren

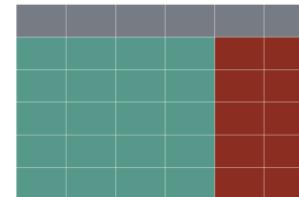


Abbildung 32: Die zwei Arten des ungeleiteten Modellierens

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?

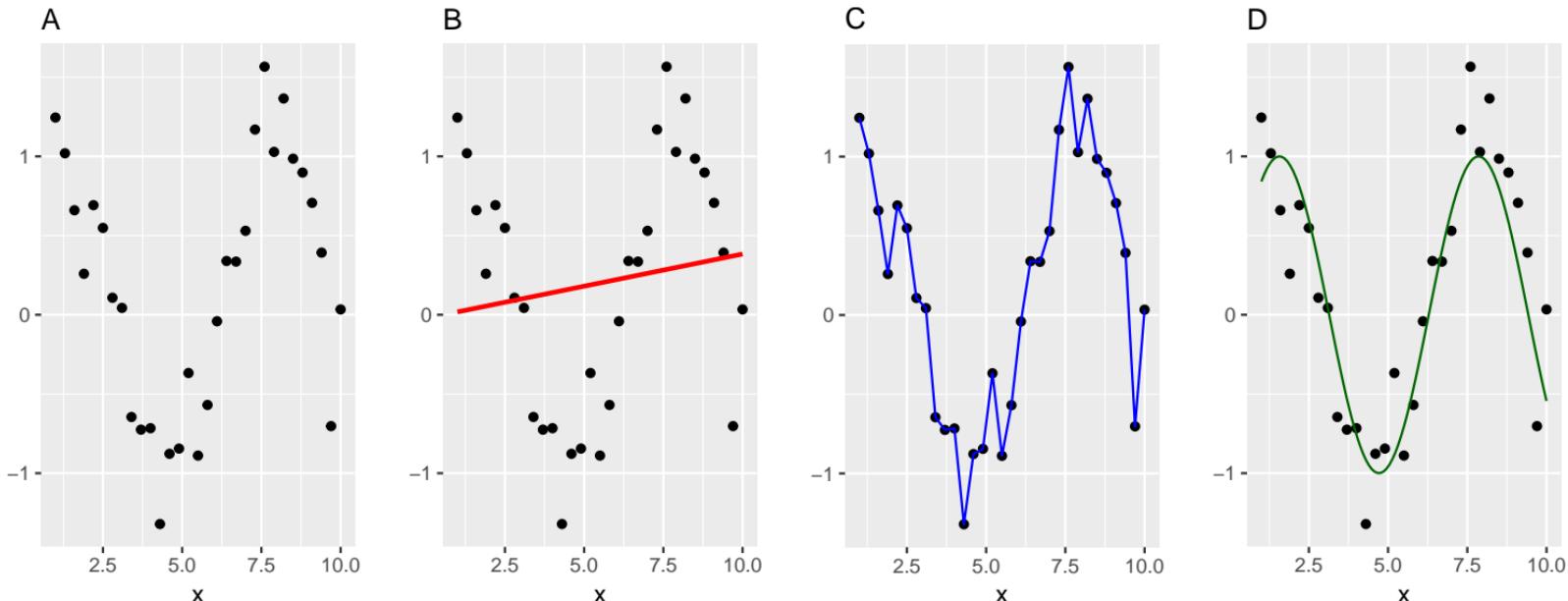


Abbildung 33: Welches Modell (Teil B-D; rot, grün, blau) passt am besten zu den Daten (Teil A) ?

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.

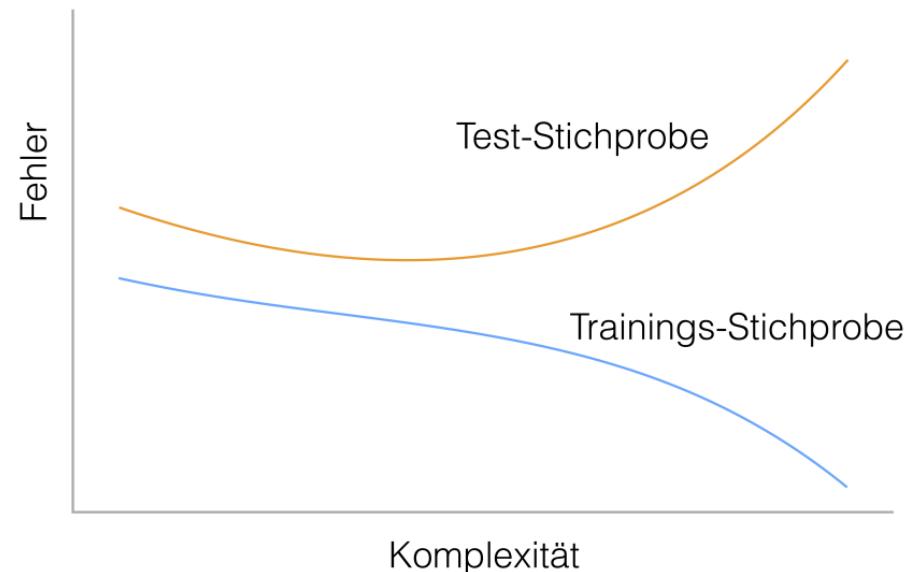


Abbildung 34: 'Mittlere' Komplexität hat die beste Vorhersagegenauigkeit (am wenigsten Fehler) in der Test-Stichprobe

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

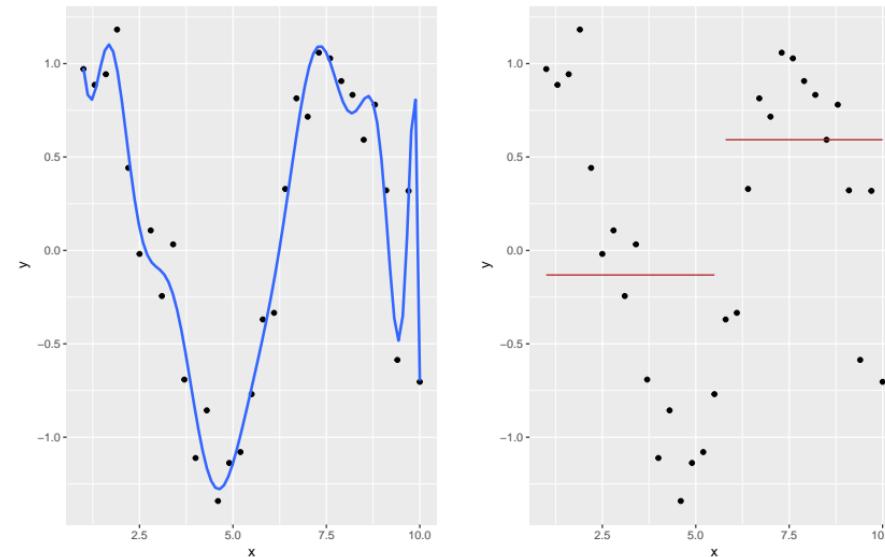


Abbildung 35: Der Spagat zwischen Verzerrung und Varianz

Der p-Wert

8. 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.

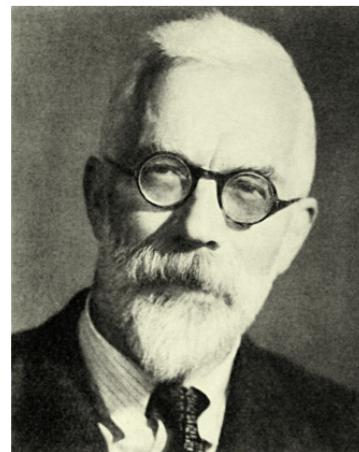


Abbildung 36: Der größte Statistiker des 20. Jahrhunderts ($p < .05$)

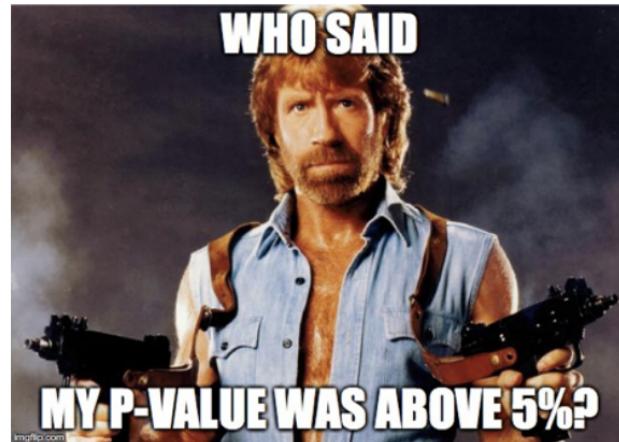


Abbildung 37: Der p-Wert wird oft als wichtig erachtet

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

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

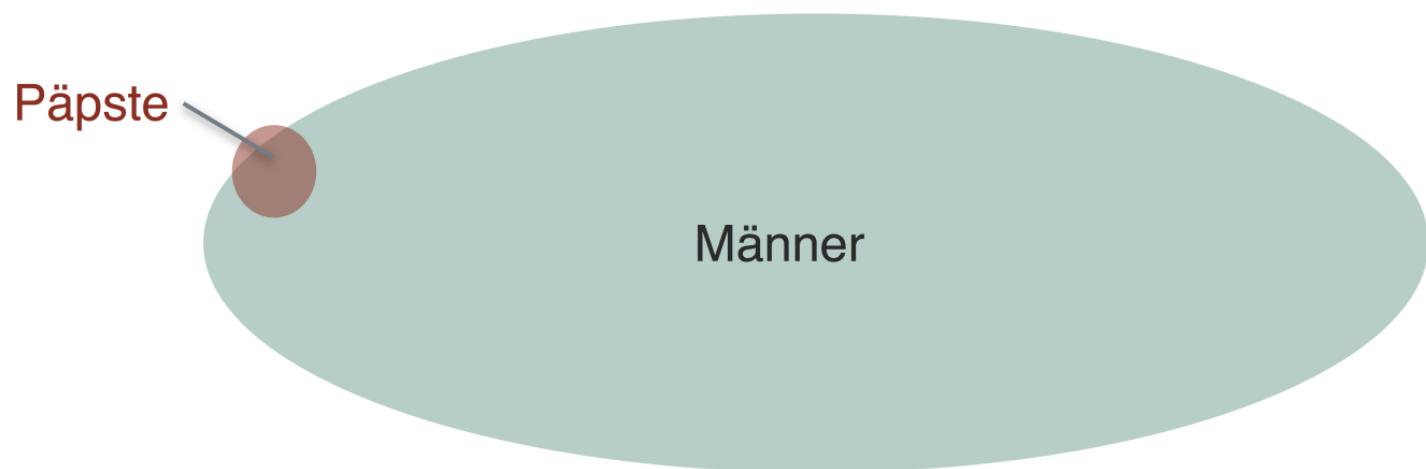


Abbildung 38: Mann und Papst zu sein, ist nicht das gleiche.

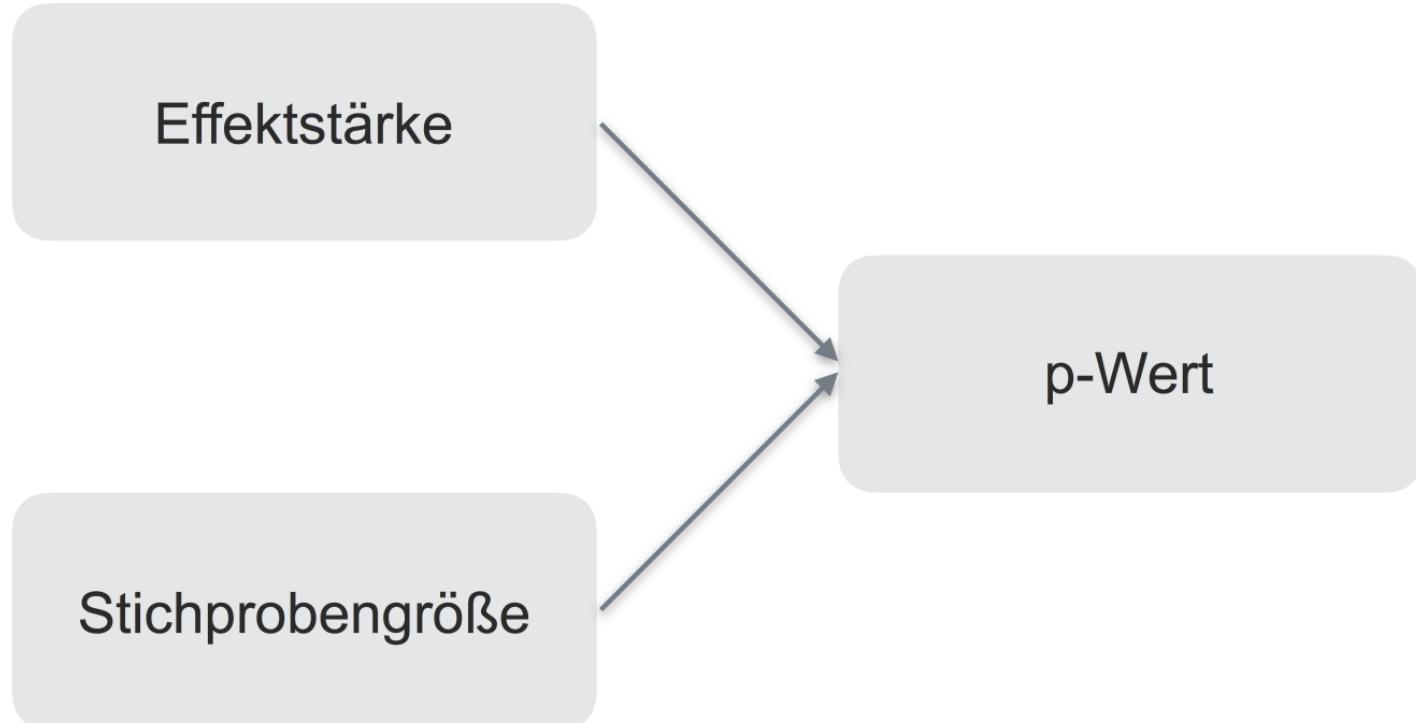


Abbildung 39: Zwei Haupteinflüsse auf den p-Wert

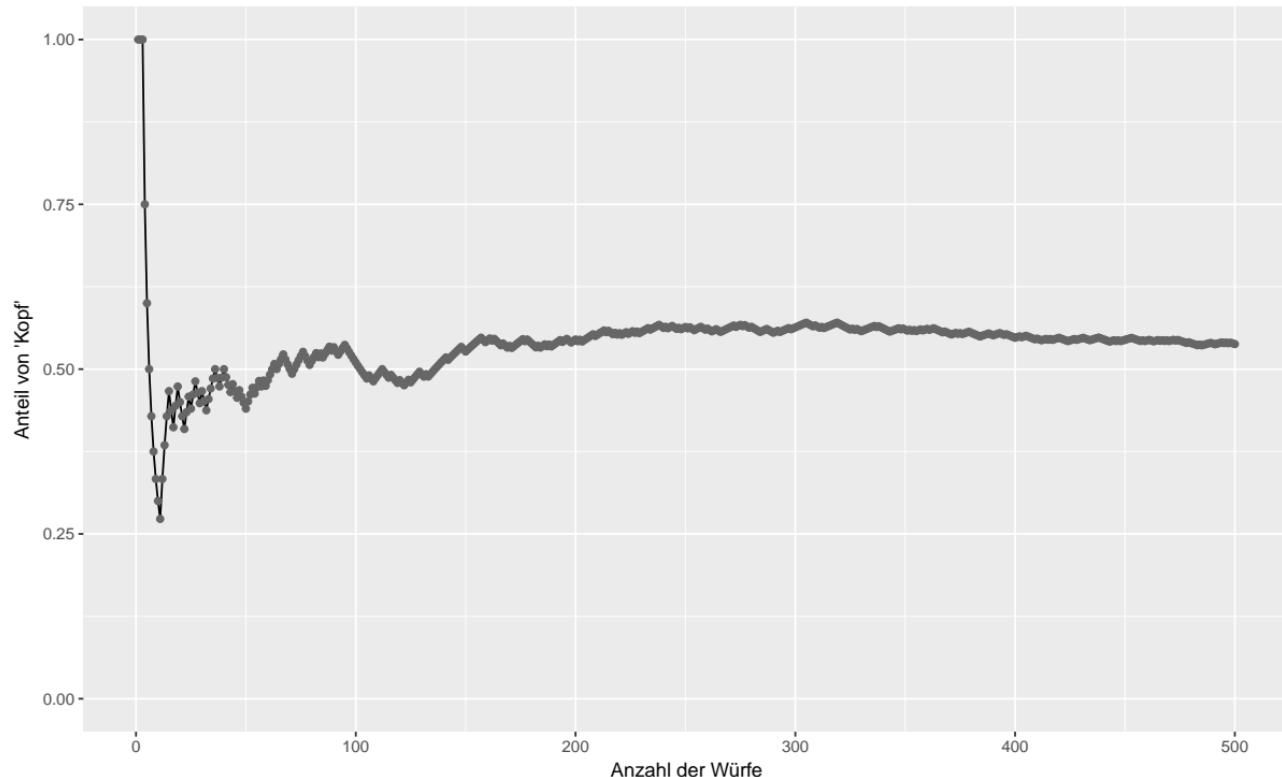


Abbildung 40: Anteil von 'Kopf' bei wiederholtem Münzwurf

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](#)

8. Der p-Wert

Alternativen zum p-Wert - Effektstärken

19. Tabelle im Skript

Bayes liefert $p(D|H)$.

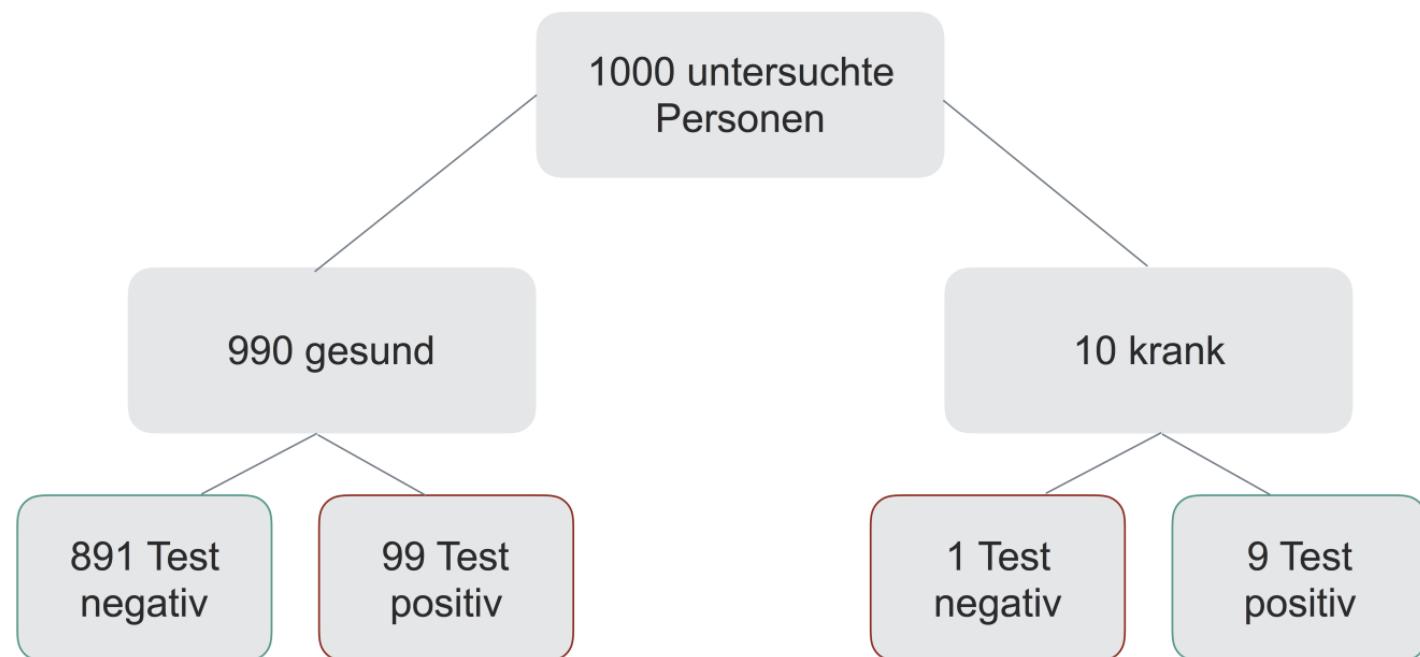


Abbildung 41: Die zwei Stufen der Bayes-Statistik in einem einfachen Beispiel

Lineare Regression

- ▶ 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.

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

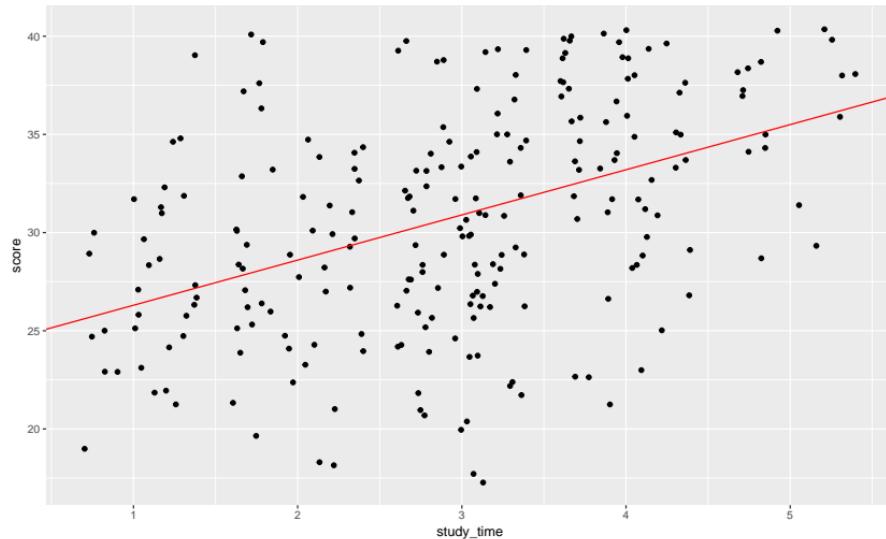


Abbildung 42: Beispiel für eine Regression

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

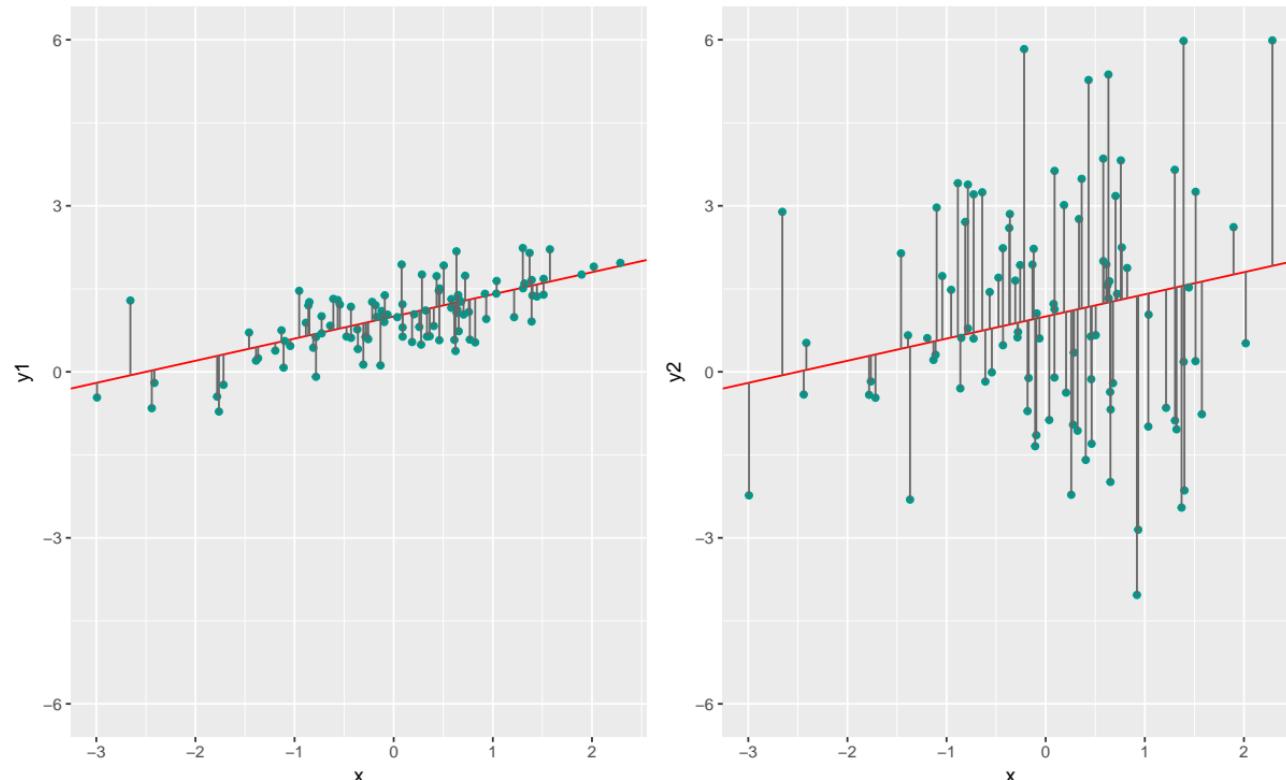


Abbildung 43: Geringer (links) vs. hoher (rechts) Vorhersagefehler

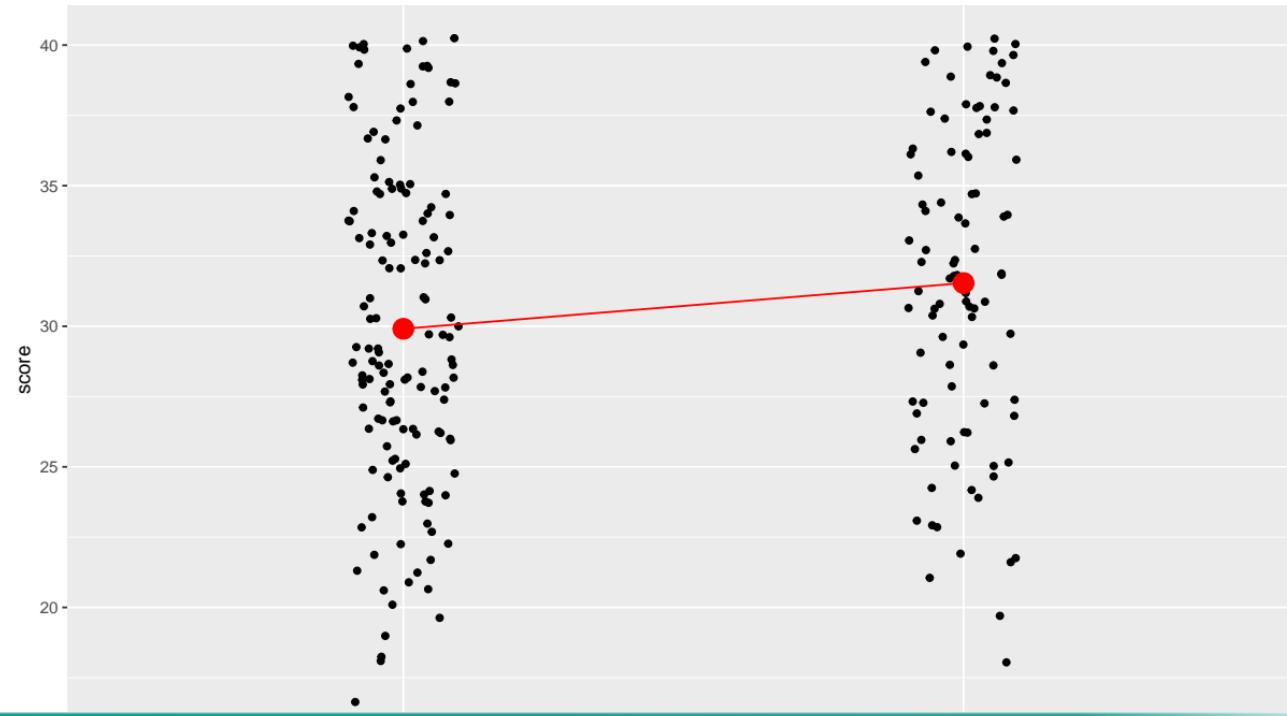
$$\text{MSE} = \frac{1}{n} \sum (\text{pred} - \text{obs})^2$$

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

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

interessiert	score
FALSE	29.90909
TRUE	31.53684
NA	33.08824

```
stats_test %>% na.omit %>% ggplot() + aes(x = interessiert, y = score) + geom_jitter(width = 0.1) +  
  geom_point(data = score_interesse, color = "red", size = 5) + geom_line(data = score_interesse,  
  group = 1, color = "red")
```



9. Lineare Regression Multiple Regression

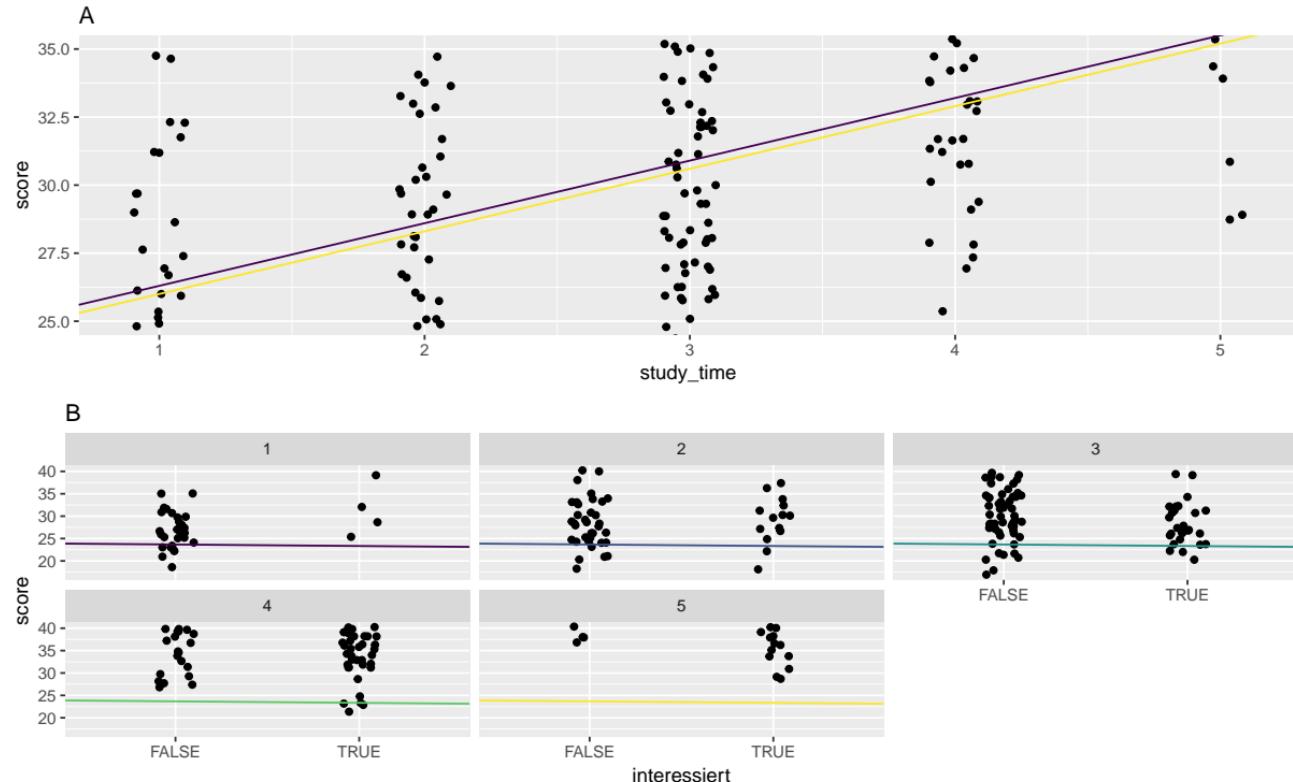


Abbildung 44: Eine multivariate Analyse fördert Einsichten zu Tage, die bei einfacheren Analysen verborgen bleiben

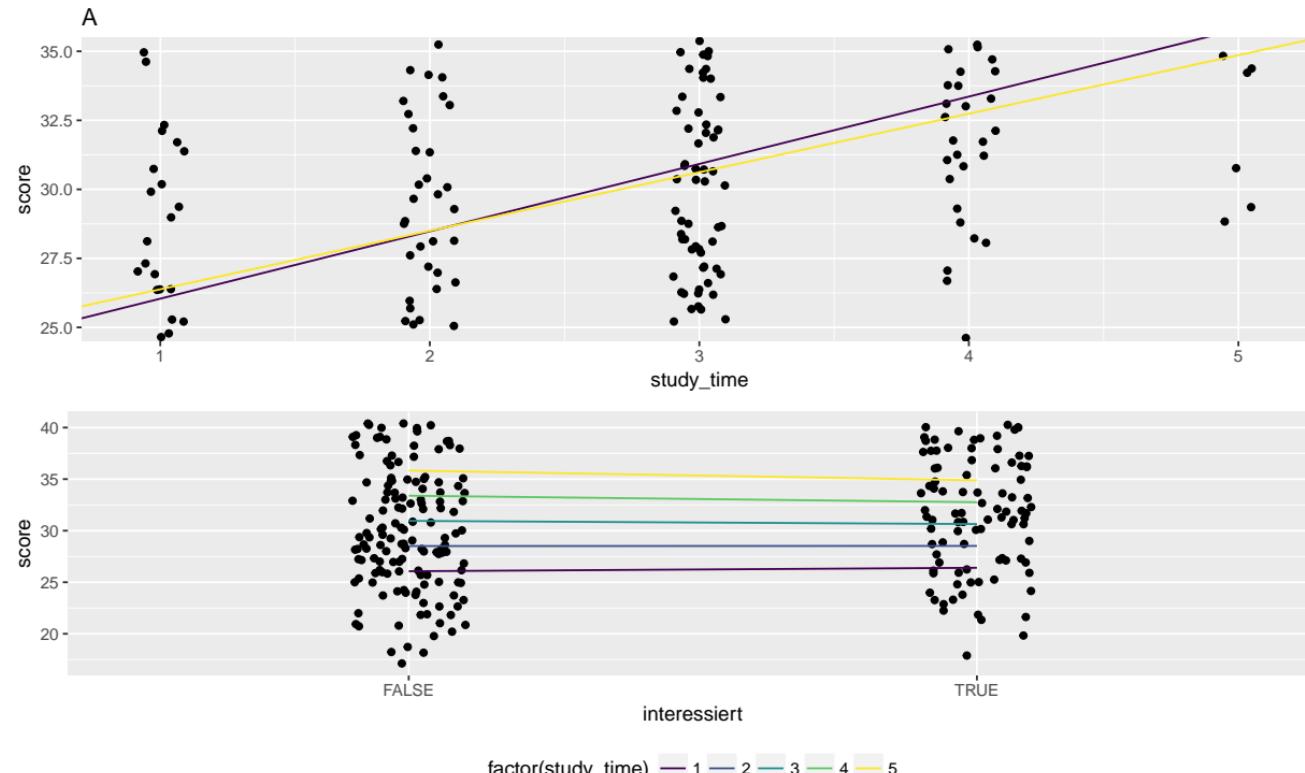
Die multivariate Analyse zeigt ein anderes Bild, ein genaueres Bild als die einfache Analyse. Ein Sachverhalt, der für den ganzen Datensatz gilt, kann in Subgruppen anders sein.

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

9. Lineare Regression

Ein Beispiel für einen Interaktionseffekt

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



```
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

- ▶ 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.

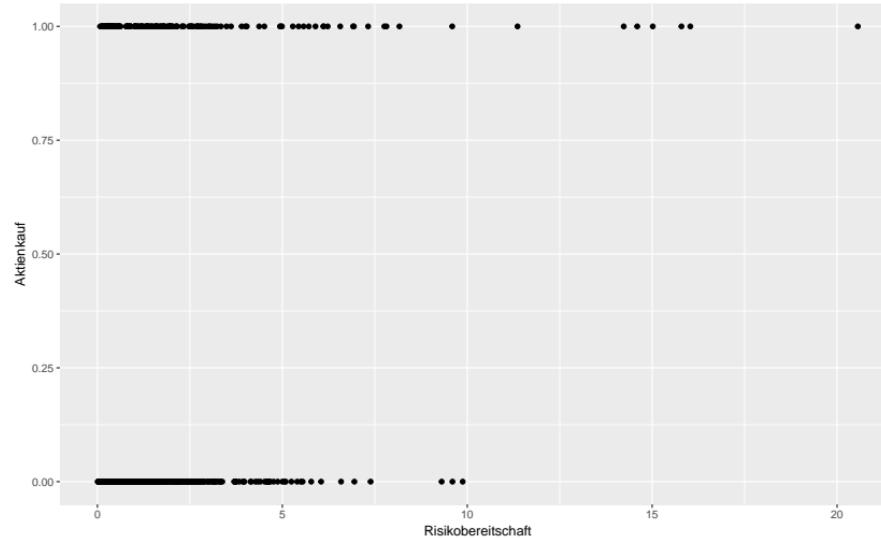


Abbildung 46: Streudiagramm von Risikobereitschaft und Aktienkauf

Die Funktion `glm` führt die logistische Regression durch.

```
glm1 <- glm(Aktienkauf ~ Risikobereitschaft, family = binomial("logit"), data = Aktien)
```

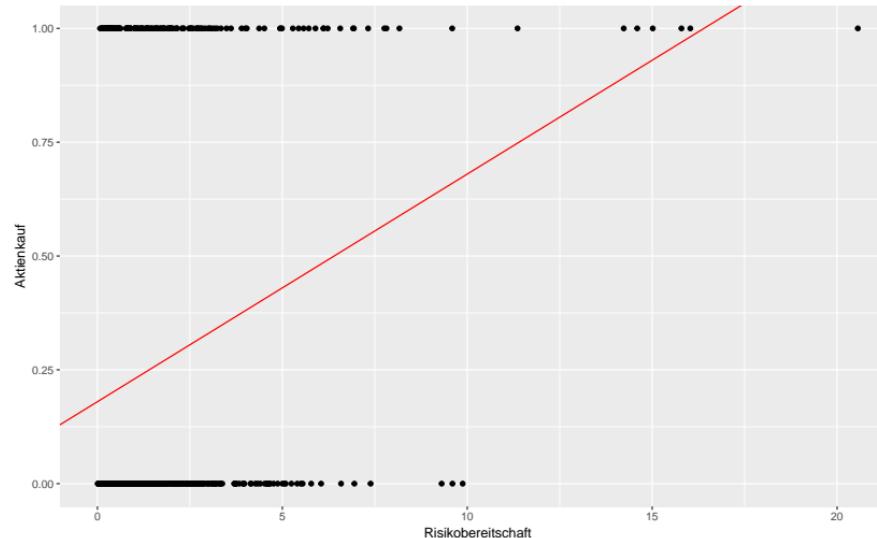


Abbildung 47: Regressionsgerade für Aktien-Modell

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

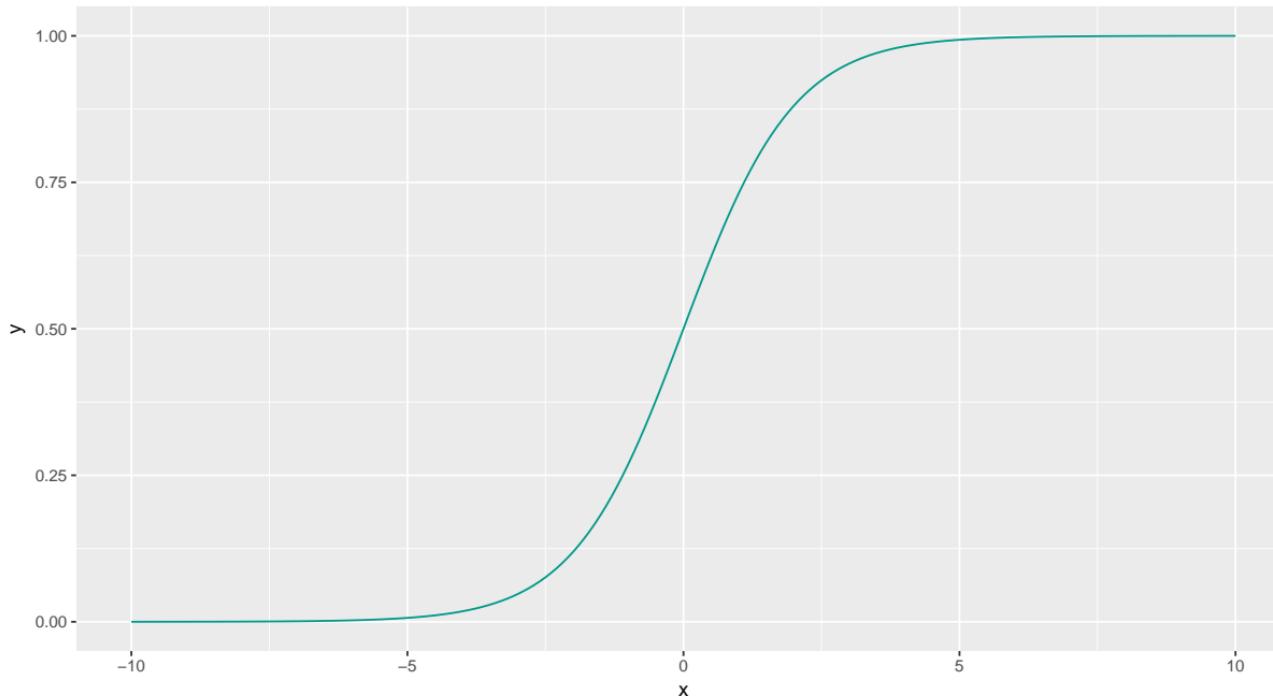


Abbildung 48: Die logistische Regression beschreibt eine 's-förmige' Kurve

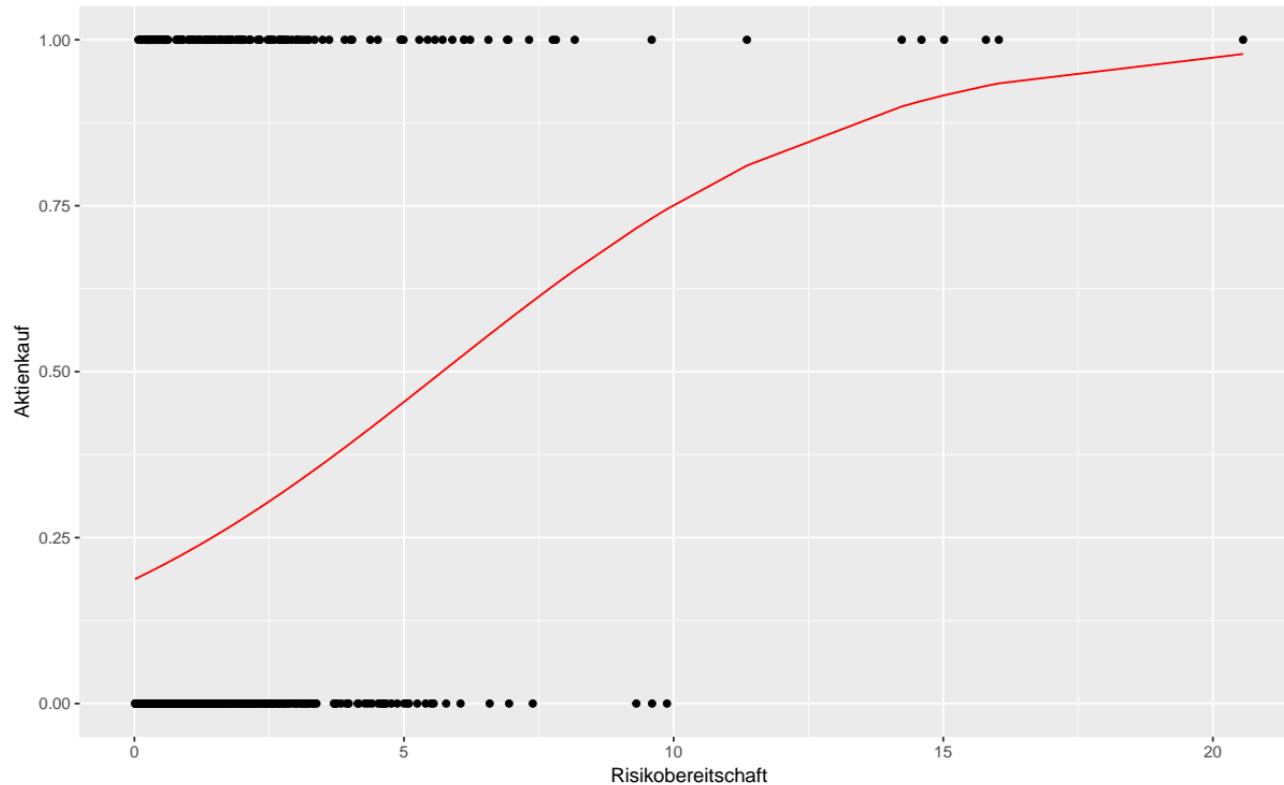


Abbildung 49: Modelldiagramm für den Aktien-Datensatz

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

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

y = intercept + 3*Risikobereitschaft, also

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

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

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

```
predict(glm1, data.frame(Risikobereitschaft = 1), type = "response")
```

```
##          1
## 0.2294028
```

```
str(stats_test$bestanden)
stats_test$bestanden <- factor(stats_test$bestanden, levels = c("nein", "ja"))
log_stats <- glm(bestanden ~ interessiert, family = binomial("logit"), data = stats_test)
summary(log_stats)
```

Tabelle 3: 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

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

## [1] 0.1092896

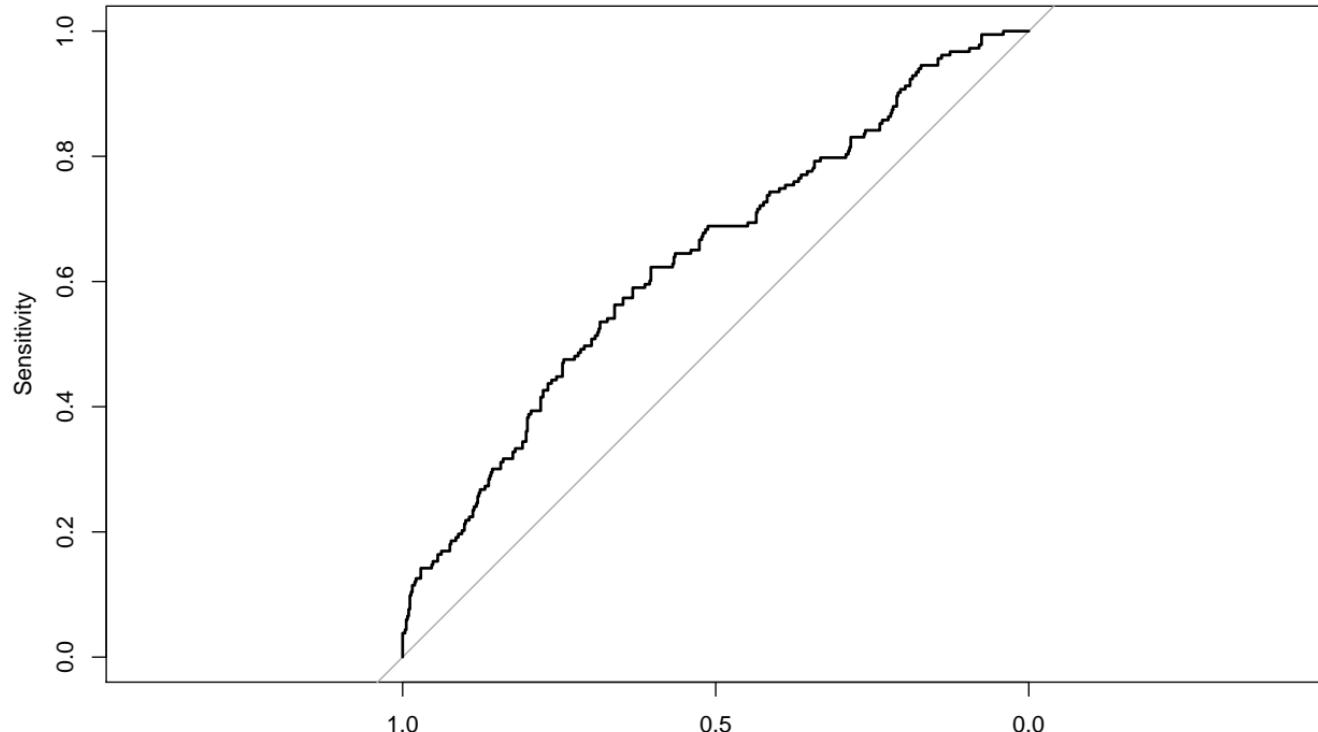
## [1] 0.9845261
```

Tabelle 5: 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)

10. Klassifizierende (logistische) Regression ROC-Kurven

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



10. Klassifizierende (logistische) Regression Beispiele für ROC-Kurven

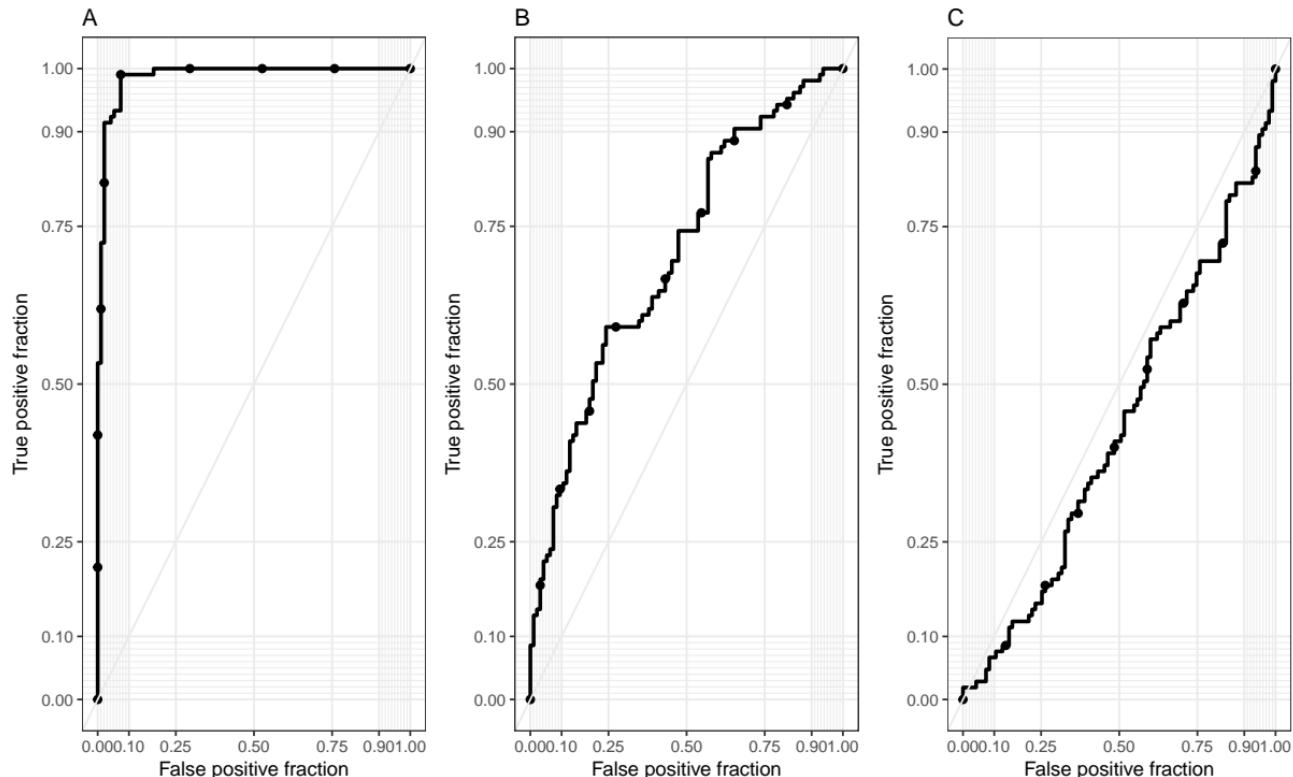


Abbildung 50: Beispiel für eine sehr gute (A), gute (B) und schlechte (C) Klassifikation

Clusteranalyse

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

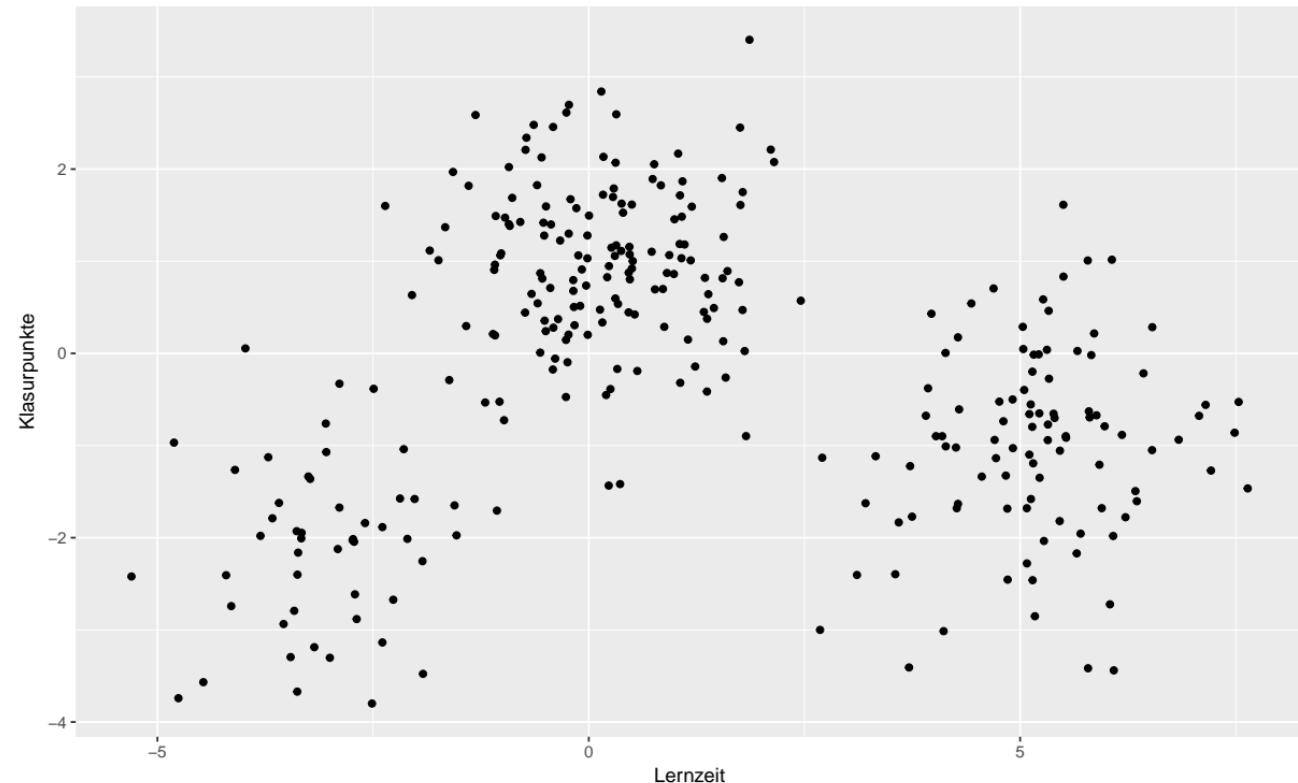


Abbildung 51: Ein Streudiagramm - sehen Sie Gruppen (Cluster) ?

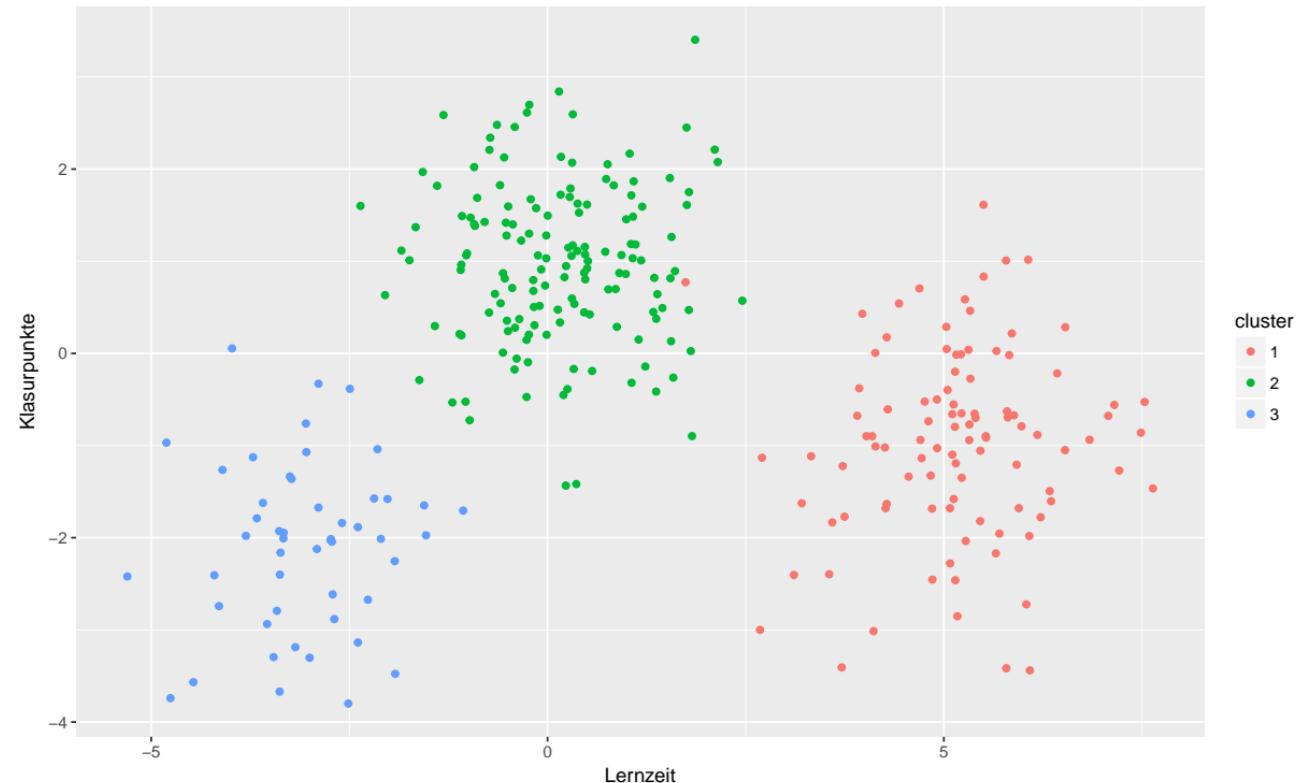


Abbildung 52: Ein Streudiagramm - mit drei Clustern

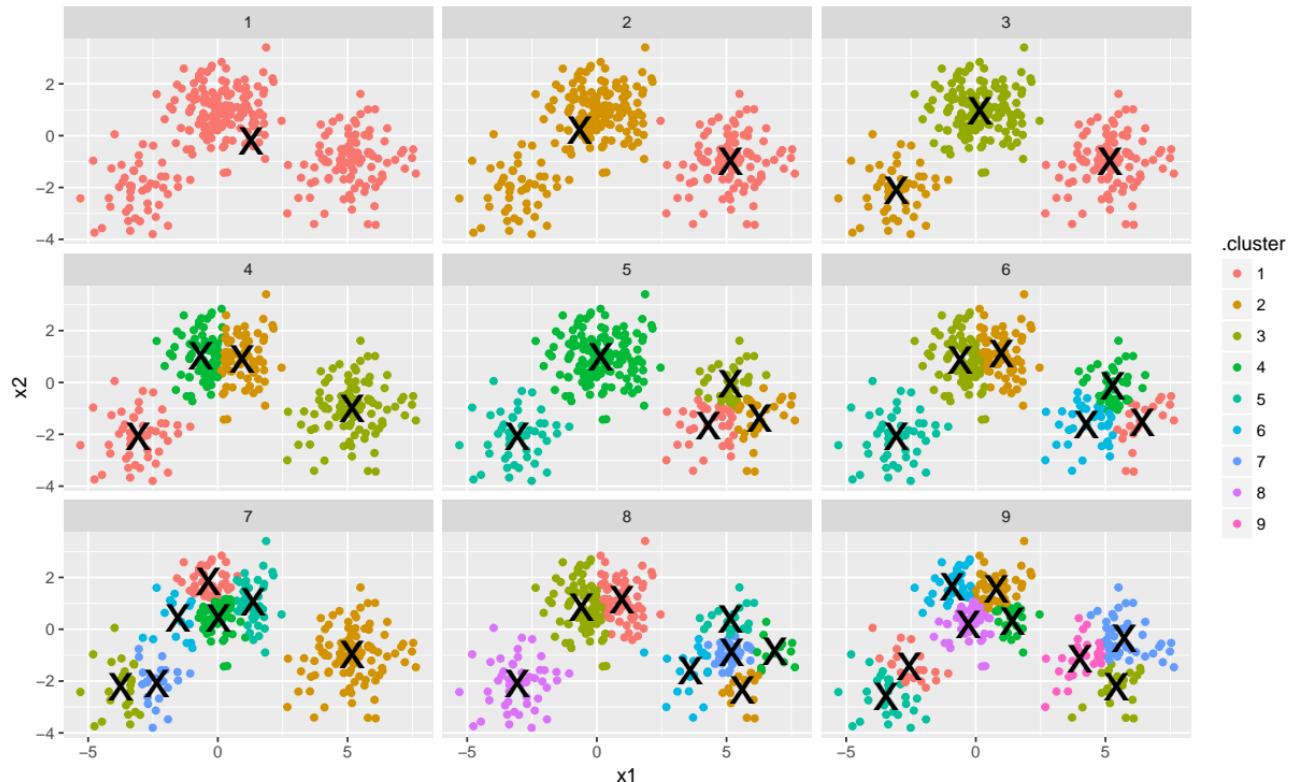


Abbildung 53: Unterschiedliche Anzahlen von Clustern im Vergleich

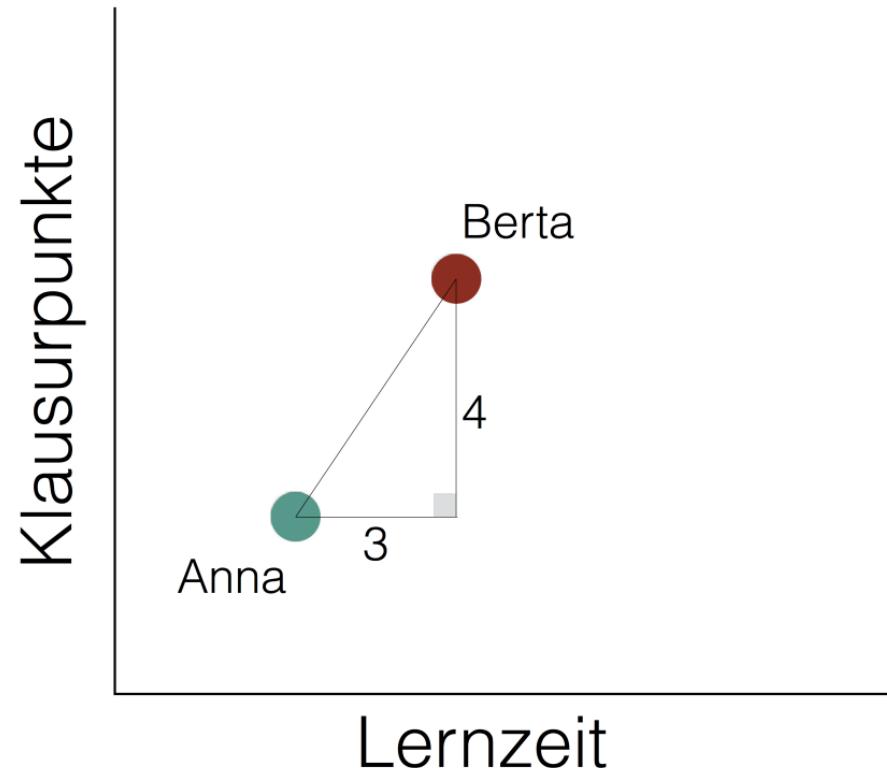


Abbildung 54: Distanz zwischen zwei Punkten in der Ebene

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

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*.

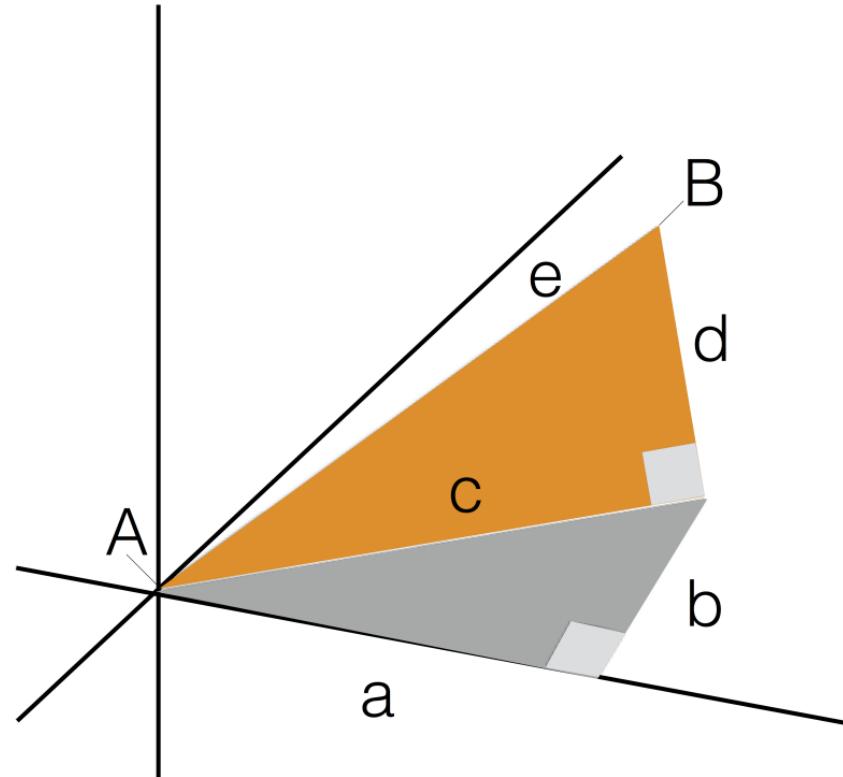


Abbildung 55: Pythagoras in 3D

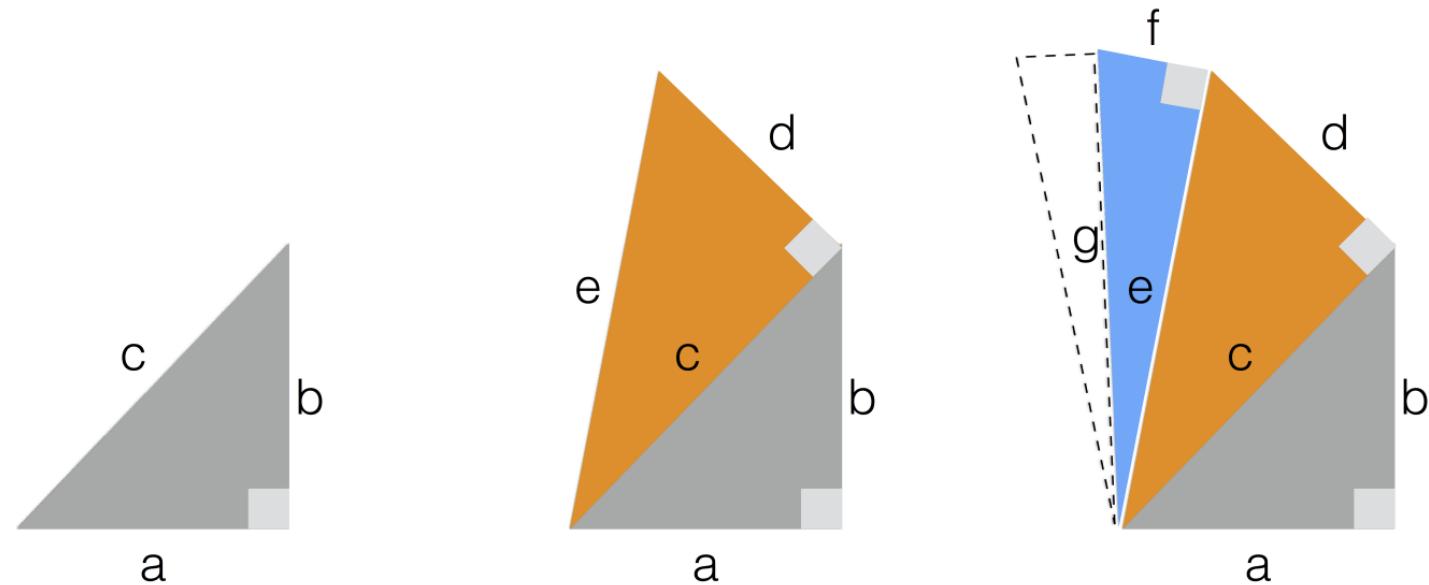


Abbildung 56: Pythagoras in Reihe geschaltet

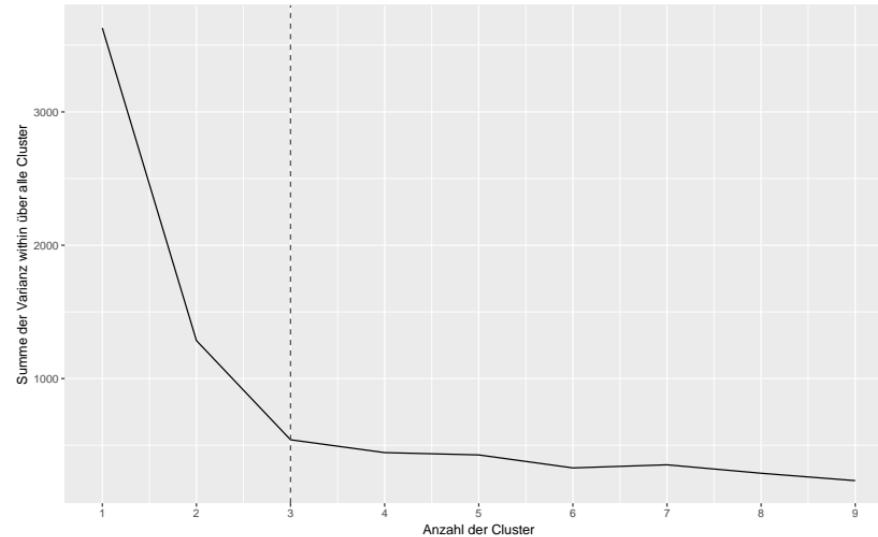
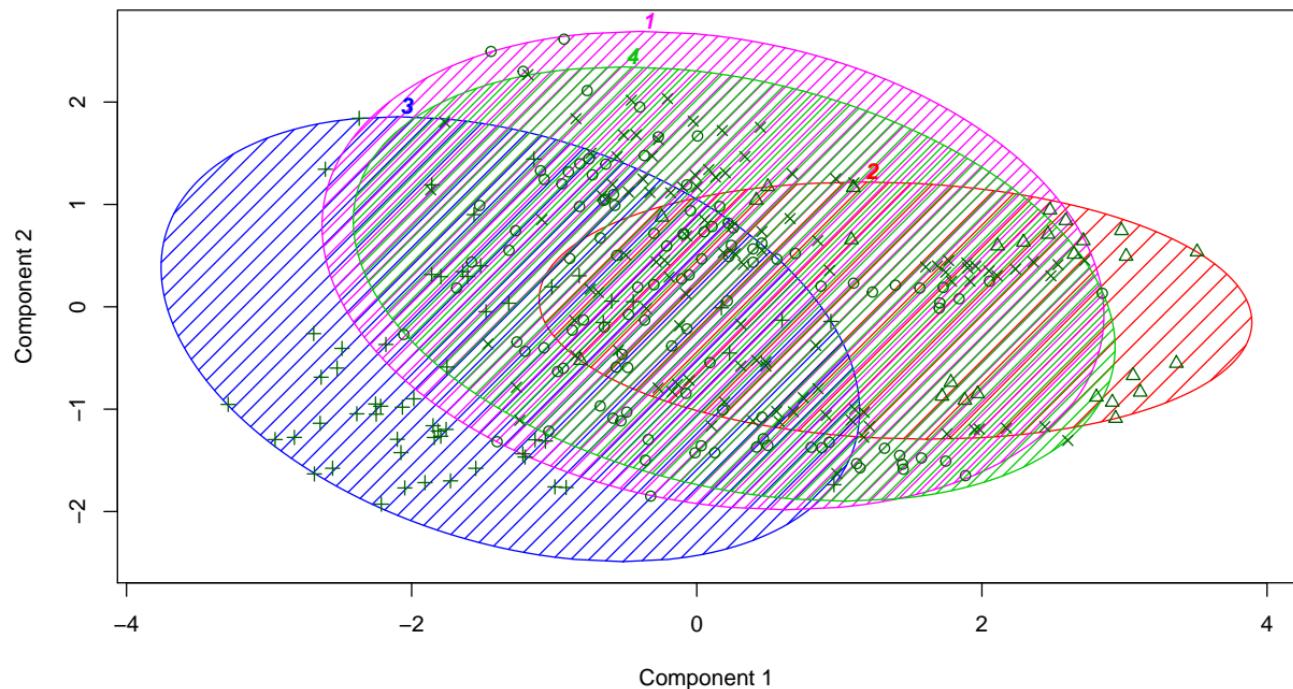


Abbildung 57: Die Summe der Varianz within in Abhängigkeit von der Anzahl von Clustern. Ein Screeplot.

CLUSPLOT(segment.num)



These two components explain 50.83 % of the point variability.

Dimensionsreduktion

- ▶ 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.

- ▶ 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

- ▶ Dimensionen reduzieren
- ▶ Unsicherheit verringern
- ▶ Aufwand verringern

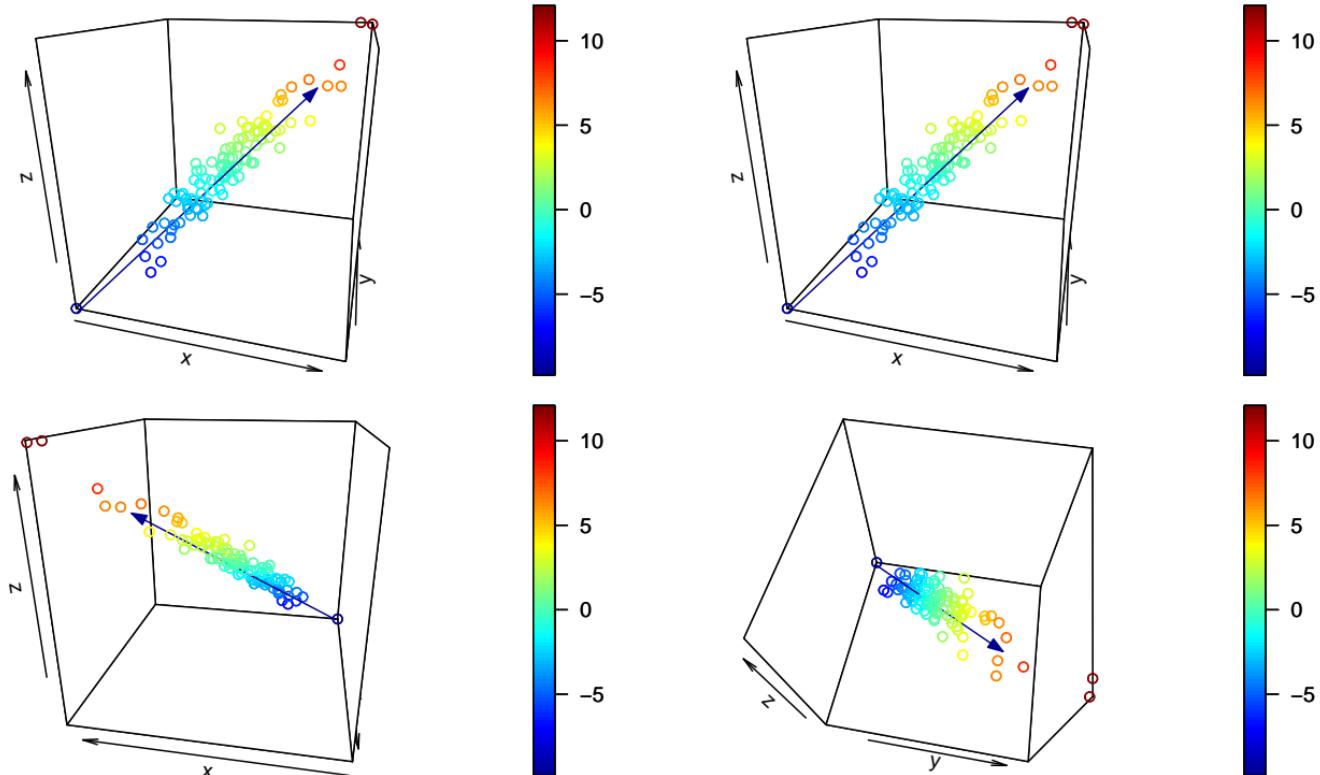
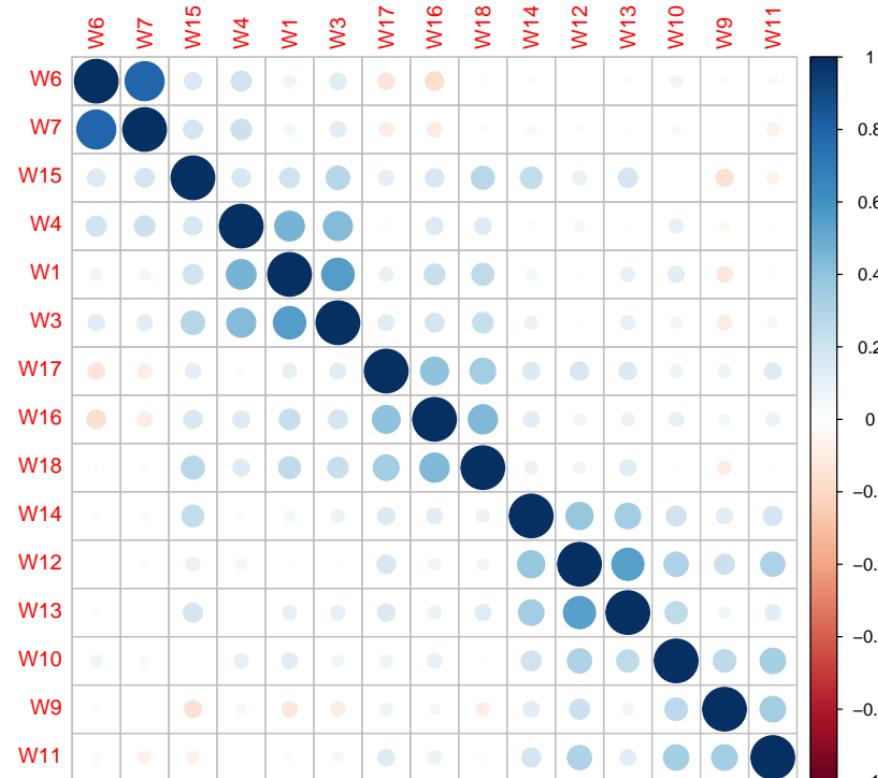


Abbildung 58: Der Pfeil ist eindimensional; reduziert also die drei Dimensionen auf eine

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

12. Dimensionsreduktion Korrelationsplot

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



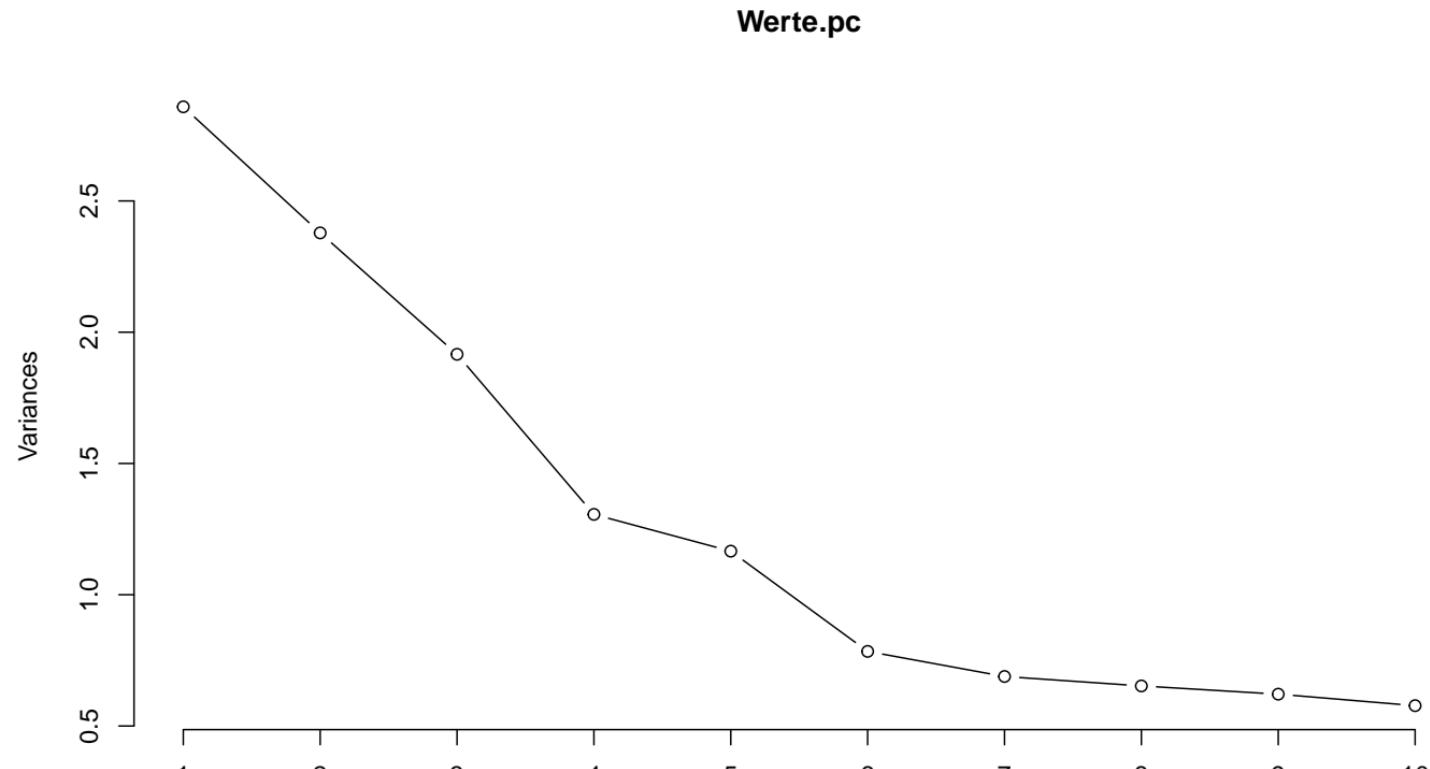
```
Werte.pc <- prcomp(Werte.sc) # Principal Components berechnen
summary(Werte.pc)

Gesamtvarianz <- sum(Werte.pc$sdev^2)

# Varianzanteil der ersten Hauptkomponente
Werte.pc$sdev[1]^2/Gesamtvarianz
```

12. Dimensionsreduktion Scree-Plot

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

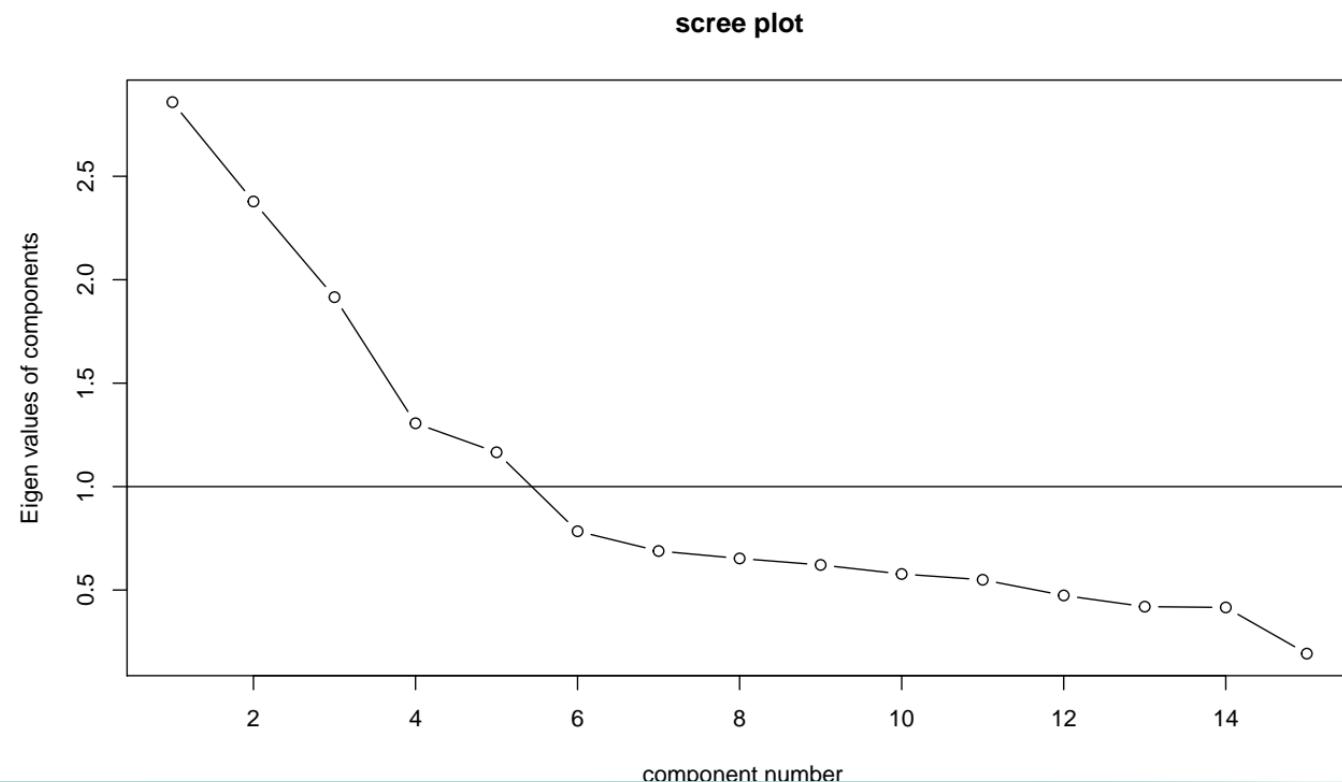


Der *Eigenwert* ist eine Metrik für den Anteil der erklärten Varianz pro Hauptkomponente.

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

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

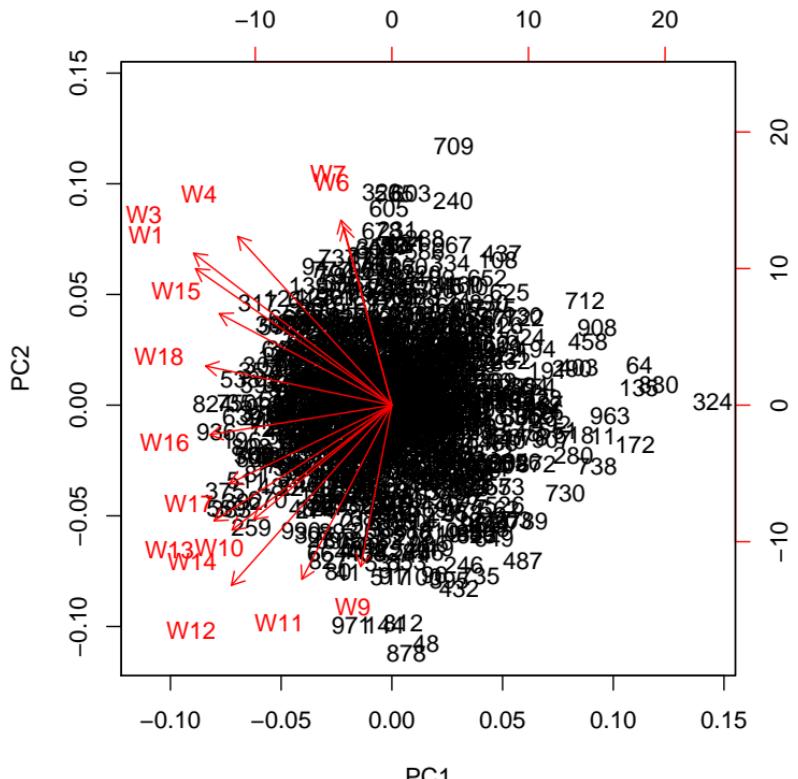
VSS.scree(Werte)



12. Dimensionsreduktion

Biplot

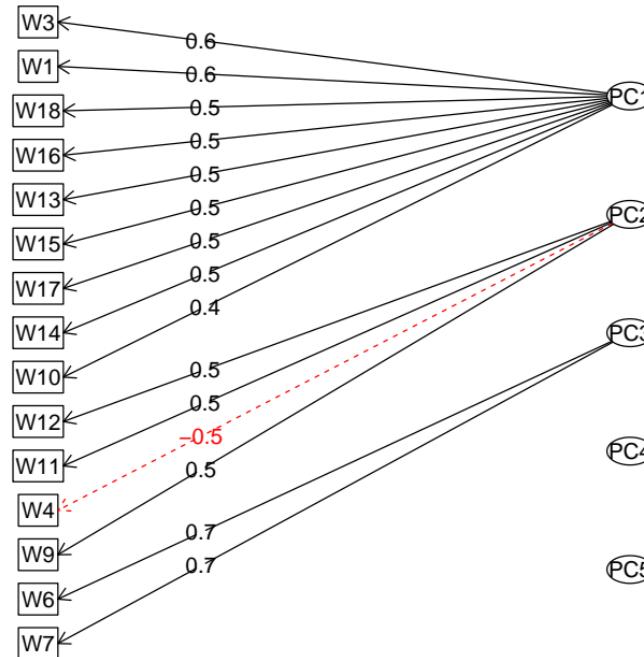
```
biplot(Werte.pc)
```



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

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

Factor Analysis



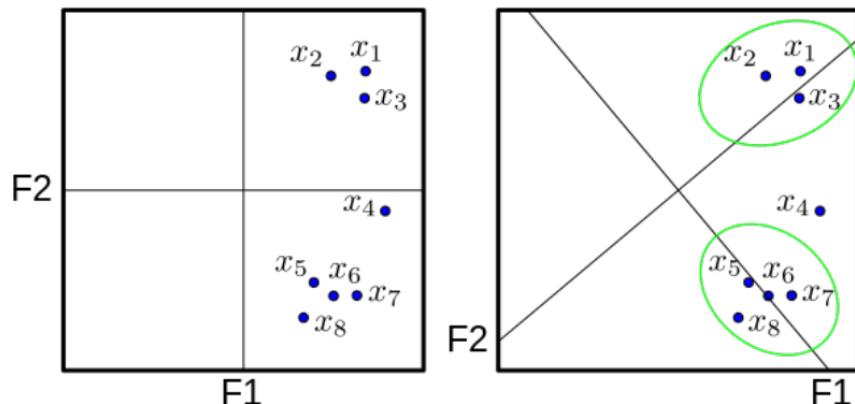


Abbildung 61: Beispiel für eine rechtwinklige Rotation

12. Dimensionsreduktion Heatmap

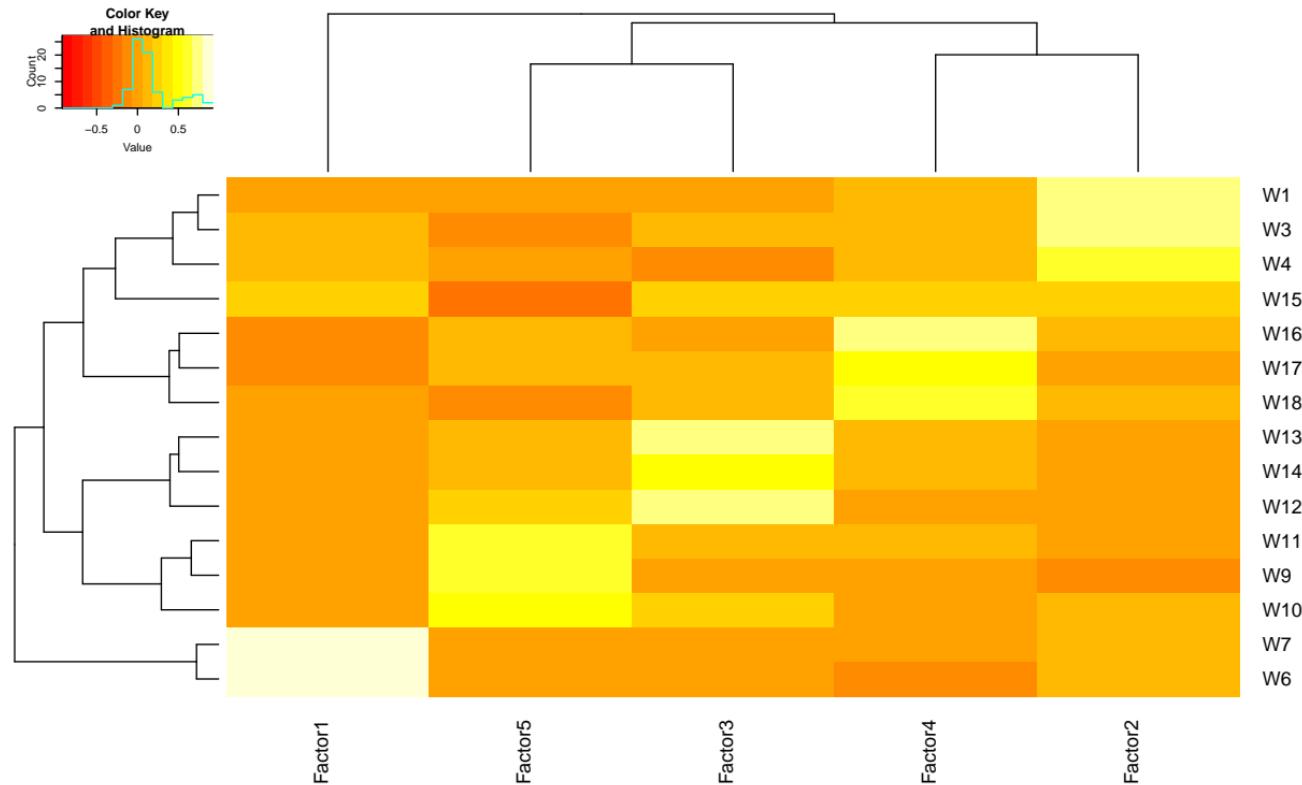


Abbildung 62: Heatmap einer EFA

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

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

```
df <- Werte %>% select(W12, W13, W14, W15)  
  
psych::alpha(df, check.keys = TRUE)
```

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

Textmining

- ▶ 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.

- ▶ 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.

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

Zeile	wort
1	wir
1	haben
1	die
1	frauen
1	zu
1	bett

In einem 'tidy text Dataframe' steht in jeder Zeile ein Wort (token) und die Häufigkeit des Worts im Dokument.

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 26396
```

```
afd_df %>%  
  na.omit() %>%  # fehlende Werte löschen  
  dplyr::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
```


13. Textmining

word

ab
aber
abgesehen alle
allein
aller

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

Tabelle 10: Die häufigsten Wörter im AfD-Parteiprogramm mit 'stemming'

token_stem	n
deutschland	219
afd	171
deutsch	119

```
wordcloud(words = afd_count$token_stem, freq = afd_count$n, max.words = 100,  
          scale = c(2, 0.5), colors = brewer.pal(6, "Dark2"))
```



Anhang