Datenanalyse I+II

WT+FT 2010

Dr. Detlef Steuer
Tel. 2819, steuer@hsuhh.de

22. Juni 2010

Struktur der Veranstaltung

- Die Vorlesung war komplett neu konzipiert für BAMA.
- Gegenüber dem Vorjahr findet von vornherein eine Straffung statt.
- Veranstaltung prinzipiell im EDV Labor.
- Sprechstunde ist im Prinzip jederzeit, für ausführliche Beratung bitte telefonisch oder per mail Termin ausmachen.
- Das Skript soll nach Möglichkeit jeweils am Dienstagnachmittag im Netz stehen.
- Tel 2819, steuer@hsuhh.de

Ziel der Veranstaltung

- Vermitteln einiger üblicher statistischer Analysemethoden in der Theorie.
- Vermitteln der Nutzung eines Werkzeugs zur zeitgemäßen Anwendung dieser Methoden.
- Vermitteln eines Eindrucks der Methoden und Probleme der praktischen Datenanalyse.

Methodische Inhalte (I + II)

- *Sehr* kurze Einführung (*crash course*) in das Programm R
- Datenvorbereitung
- Vertiefung Regression (diagnostische Plots, multiple Regression, p-Wert)
- Varianzanalyse (ANOVA)
- Clusterverfahren (Diskriminanzanalyse, Entscheidungsbäume etc.)
- Entdecken latenter Variablen (Faktoranalyse, Hauptkomponenten)
- Zeitreihenanalyse (Trend/Saisonmodelle, ARMA etc.)

Herangehensweise

- Mathematisch saubere Einführung von Verfahren, aber auch etliche Beispiele und grafische Verfahren
- Alle Verfahren werden auch im Rechner umgesetzt (R www.r-project.org)
- Die Veranstaltung findet im EDV-Labor statt
- Besonderes Augenmerk auf der Interpretation der Ergebnisse der Verfahren, nicht auf der einfachen (blinden) Anwendung
- Folien jeweils vorlesungsbegleitend als Skript (möglichst dienstags vor der Vorlesung)
- Klausur wird auf jeden Fall gut vorbereitet, Probeklausur unter realistischen Bedingungen zu Beginn des FT.

Literatur

- Dalgaard, Introductory statistics with R, Springer (elektronisch über die Bibliothek verfügbar)
- Faraway, Linear Models in R, Chapmann and Hall
- Ligges, Programmieren in R, Springer (elektronisch über die Bibliothek verfügbar)
- Literatur für den ersten Teil der Vorlesung, Beispiele sind dort zum Teil entnommen
- Reichhaltige Informationen im Netz!

Was ist Datenanalyse?

- Datenanalyse ist ein *Prozess*, der über das reine, mathematische Verfahren herausgeht!

- Schritte in diesem Prozess sind:

  1. Vertraut machen mit den Daten, d.h. Erläuterungen des Datenlieferanten verstehen. Woher kommen die Daten? Sind sie automatisch erfasst (gemessen) oder von Hand erfasst (Umfragen)?
  2. Daten reinigen, d.h. Ausreißer identifizieren, *missing values* eindeutig und einheitlich kodieren.
  3. Die eigentliche Analyse zerfällt in zwei Teile:
     – Die explorative Analyse (Histogramm, Boxplot etc.), Deskription,
     – und die Modellierung (Regression!) und Tests, schließende Statistik.
  4. Präsentation der Ergebnisse, d.h. sinnvolle Auswahl aus den Ergebnissen treffen, und stringent und punktgenau aufbereiten.

Aufgabe 1

Aufgabe: Installieren Sie R auf Ihrem Rechner oder machen Sie sich im EDV Labor mit dem Programm vertraut. Vollziehen Sie Beispiele der Vorlesung nach!

Alles weitere, z.B. Einlesen von Dateien, wenn es in der Vorlesung nötig wird.

Die Programmiersprache R

- R ist eine (Interpreter-)Sprache und eine Arbeitsumgebung für statistische Grafik und Analyse.

- R liefert in der Standardinstallation bereits eine große Zahl von statistischen und grafischen Verfahren der Datenanalyse und ist darüber hinaus entworfen, um leicht erweiterbar zu sein. Es gibt über 2000 Erweiterungspakete für alle Aspekte der Datananalyse.

- Evtl. die größte Stärke von R liegt in der leichten Anfertigung von veröffentlichungsfähigen Plots, inklusive mathematischer Annotationen.

- Das R-Core Team nennt R eine Umgebung für statistische Berechnugen und Grafik.

Die Programmiersprache R

- In dieser Vorlesung: Beschränkung auf die bereits implementierten Teile.
- R ist dann eine Art statistischer (Hochleistungs-)Taschenrechner.

Warum R?

- R ist *Freie Software* (kostenlos und open source).
- R ist plattformunabhängig, d.h. Sie nutzen weiter den Rechner und das Betriebssystem, das sie gewohnt sind, sei es Windows, MacOs oder Unix.
- Hervorragende Fähigkeiten: Immer mehr Firmen nutzen R, also bekommen Sie ein Werkzeug an die Hand, das Sie fast sicher im beruflichen Umfeld wieder sehen werden. R entwickelt sich im universitären Bereich zur Standardsoftware, ebenso, wenn auch verzögert im industriellen Bereich.
- Am 7.1.2009 sogar eine Titelseitengeschichte der NYT!

Warum R?

- Hervorragende eingebaute Hilfefunktion!
- Lokalisiert in etlichen Sprachen.
- Professioneller (oder besser) Support über Mailinglisten!
- Professionelle (oder besser) Qualitätskontrolle der Software ('make check'). Validierung der Software und der Rechenergebnisse während der ganzen Entwicklung.
- Sehr gute Handbücher werden mitinstalliert (Reference Manual $>$ 1800 Seiten).
- Für Bachelor-, Master- oder Doktorarbeiten: sehr gute Integration mit LaTeXund OpenOffice. (MS Office ist auch ok.)

Benutzerinterfaces für R

- Eingebaut ist nur ein CLI. R ist ein Interpreter mit *read-eval-loop*!
- Empfehlenswert: Interface zu einem externen Editor (emacs (!), winedt, etc.).
- Es gibt GUIs: gehören nicht zur Standardinstallation und werden in der Vorlesung nicht behandelt. Windows hat ein rudimentäres Mausinterface, aktuell scheint Tinn-R das einfachste zu sein.
- Batch mode (skriptgesteuert).
- etwas ausgefallener: R als Modul des Webservers.
- oder: R als shared library aus anderen Programmiersprachen aufrufen (python, perl).

Eine erste R-Sitzung

```
steuer@gaia:~> R
R version 2.10.1 Patched (2010-01-08 r50953)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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

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

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

Zuerst:

```
> Sys.setenv(http_proxy="http://backspace.unibw-hamburg.de:3128")

### Setzt den Uni-Proxy. Nicht nötig außerhalb des Uni-Netzes!

> contributors()

### Liste der Entwickler

> citation()

### Zitierung von R als Literaturstelle.
### R hat eine ISBN!
```

Erste Schritte: interaktive Nutzung von R

- R als Taschenrechner

```
> 3 + 4
[1] 7

> log(0)
[1] -Inf

> log(-1)
[1] NaN
Warning message:
NaNs were generated in: log(x)
```

Erste Schritte: interaktive Nutzung von R

```
> pi  #es kommt auf Groß- oder Kleinschreibunrg an
[1] 3.141593

> x <- .Last.value
# x = .Last.value geht "neuerdings" auch

> ls()
[1] x

> rm(x)
> q()
```

Externe Pakete

- In erheblichem Umfang zusätzliche Funktionalität in externen *packages* (oder *views*)
  `available.packages()` gibt eine Liste der aktuell vorhandenen Pakete

- Einfaches Einfügen in eine bestehende R Installation

```
install.packages("scatterplot3d")
```

- Laden in eine laufende R-Sitzung mit `library(scatterplot3d)` or `require(scatterplot3d)`

- Entfernen aus einer laufenden Sitzung `detach(package:scatterplot3d)`

Mathematische Operatoren

Symbol	Funktion
^ oder **	Potenz
*, /, +, -	Multiplikation, Division, Addition, Subtraktion
%/%, %%	ganzzahlige bzw. modulo Division
%*%	Matrixmultiplikation

Natürlich gibt es alle üblichen mathematischen Operationen: `round()`, `sin()`, `abs()`, `sqrt()` etc.

Wichtig für das Konzeptverständnis: Alle diese Operatoren sind gewöhnliche R-Funktionen:

```
> "+"(3,4)
[1] 7
```

Mathematische Operatoren

- Wichtig sind die Bezeichner für die speziellen Zahlen:
  - NaN : Not a Number,
  - Inf, -Inf : plus resp. minus unendlich,
  - NULL : nichts, leer,
  - TRUE, FALSE : Wahr oder falsch,
  - NA : not available, fehlender Wert, *missing value*.

- Achtung: `R` implementiert IEEE Arithmetik! Internationaler Standard.

```
> round(1.5) ; round(0.5)
[1] 2
[1] 0
```

- Achtung: `pi` ist nicht `PI`! `R` beachtet Groß- und Kleinschreibung!

Logische Operatoren

- == : beide Objekt sind **identisch**,
- all.equal() testet auf numerische Gleichheit bis auf eine festgelegte Abweichung,
- != : ungleich,
- <, > , <=, >= kleiner als, größer als (oder gleich),
- &, |, ! : (logisch) AND, OR, NOT .

Kleine Fallstricke

```
> a <- 3
> b <- 2.1/0.7
> a == b
[1] FALSE
```

Was passiert hier?

Kleine Fallstricke

```
> a <- 3
> b <- 2.1/0.7
> a == b
[1] FALSE
```

Was passiert hier?

Lösung in R: es gibt die Funktion `all.equal()`

```
> all.equal(a, b)
[1] TRUE
> ?all.equal
```

`all.equal()` überprüft die numerische Gleichheit bis auf ein ϵ
Standard: `sqrt( .Machine.double.eps )`

Kleine Fallstricke

Naiver Weise vermutet man, dass das Folgende funktioniert:

```
> a <- NA
> a == NA
```

oder

```
> a <- NaN
> a == NaN
```

Kleine Fallstricke

Naiver Weise vermutet man, dass das Folgende funktioniert:

```
> a <- NA
> a == NA

[1] NA
> a <- NaN
> a == NaN
[1] NA
```

Macht es aber nicht!
Für diese Fälle stellt R Folgendes zur Verfügung:

```
> a <- NA ; is.na(a)
> a <- NaN ; is.nan(a)
```

Elementare Statistik (Statistik I)

Vielzahl eingebauter Funktionen!

- `mean()`, `var()`, `sd()`, `cor()` etc.
- `runif()`, `rnorm()` etc. Zufallszahlenerzeugung
- `fivenum()`, `range()`, `summary()`, `stem()` Tukey's numbers, Spannweite, Stem-and-leaf plot
- `boxplot()`, `pie()`, `hist()` grundlegende grafische Darstellungen
- `lm()`, `t.test()` lineare Regression, t-Test

Umgang mit fehlenden Werten

- Die wichtige Option `'na.rm'` legt fest, wie NAs in Berechungen behandelt werden sollen.
  Insbesondere wichtig in der Form z.B. `mean(x, na.rm=TRUE)`, auch als globale Option `na.action`.

Kurzer Eindruck der Datenanalyse

```
> data(iris)
> names(iris)
> str(iris)
> ?iris
> summary(iris)
> attach(iris)
> species.n <- as.numeric(Species)
> plot(iris, col=species.n)
> hist(Petal.Length)
> op <- par(mfrow=c(2,2))
> for (i in 1:4){
      boxplot(iris[,i] ~ Species, main = colnames(iris[i]))}
> par(op)
```

Kurzer Eindruck der Datenanalyse

```
> library(rpart)
> (rpo <- rpart(Species ~ ., data=iris))
> plot(rpo, margin = 0.1, branch = 0.5)
> text(rpo)
> library(MASS)
> (ldao <- lda(Species ~ .,data=iris))
> plot(ldao, abbrev = TRUE, col = species.n)
> detach(iris)
> ls()
```

Eingebautes Hilfesystem

- http://cran.r-project.org/doc/contrib/Short-refcard.pdf
  Das cheat-sheet für `R`!

- `help` oder "?": äquivalent zu RTFM: versuchen Sie `help(plot)` oder `?plot`.

- Wenn man das genaue Kommando nicht weiß oder `help` nicht hilft, dann kann man `apropos()`, `find()` oder `help.search()` versuchen.

- Versteht man die Hilfeseite nicht, dann kann man mit `example(`*`command`*`)` oder `demo(`*`command`*`)` versuchen, den Befehl und seine Nutzung am Beispiel zu lernen.

Eingebautes Hilfesystem

- `help.start()` zeigt die Dokumentation im Standard-Webbrowser an.
- Die meisten von Nutzern hinzugefügten Pakete enthalten eine sog. Vignette, eine kurzes Handbuch im PDF Format. Mit dem Kommando `vignette()` kann man sich dieses anzeigen lassen.

Externe Hilfe

- Dokumentation auf CRAN: cran.r-project.org
  Sehr viel gut geschriebene Dokumentation! Installationshandbuch, Referenzhandbuch, Dokumentation für Datenaustausch, **FAQ** usw.

- Archive der Mailinglisten mit Suchinterface auf CRAN
  http://cran.r-project.org/search.html

Ultima ratio

- Selbst auf der Mailingliste r-help fragen. Unbedingt den posting guide beachten, sonst wird man ge'ripleyed'. Mehrere tausend Leser, mehr als 100 Mails am Tag. Es gibt praktisch auf jede vernünftig gestellte Frage ein fundierte Antwort.

- Bekommt man sein Problem gelöst, so sollte man sein Wissen teilen, in dem man es z.B. in das R-Wiki http://wiki.r-project.org/rwiki/doku.php einträgt.

Buchhaltung

Immer nur mit Kopien arbeiten! Nie mit Originalen!

Buchhaltung

- `ls()` zeigt in einer `R`-Sitzung alle aktuell exisitierenden Objekte an. Für etwas mehr Auskünfte über die Objekte kann man `ls.str` versuchen. Alle Objekte der aktuellen Sitzung werden in der Datei `.Rdata` im aktuellen Arbeitsverzeichnis gespeichert, wenn man mit 'y' auf `'q()'` antwortet.

- Das aktuelle Arbeitsverzeichis liest und schreibt man mit `getwd()` und `setwd()`.

- Pro: Automatische Datensicherung!

- Kontra: Es handelt sich um ein Binärformat! Man sollte dies nicht als Hauptsicherung wichtiger Daten nutzen!

- Siehe auch `save()` und `save.image()`.

Buchhaltung

- Eine natürliche Art der Arbeitsorganisation ist deshalb, pro Projekt ein Arbeitsverzeichnis zu verwenden.

- Den Verlauf der letzten eingegebenen Befehle findet man in der Datei `.Rhistory`.

Initiale Datenanalyse

- Ziel ist die Aufbereitung eines erhaltenen Datensatzes, so dass die "echte" Datenanalyse durchgeführt werden kann

- Beispieldaten aus Faraway, Linear Models with R

```
> install.packages("faraway")
> library(faraway)
> data(pima)
```

- Nutzung eines Datensatzes, der im Internet zur Verfügung gestellt wird.

- Studie des *National Institute of Diabetes and Digestive and Kidney Diseases* an 768 erwachsenen Frauen der Pima Indianer.

Hintergrundinformation

- Wegweisende Studie über den Zusammenhang von Diabetes mit genetischen Ursachen.
- Pima Indianer haben die weltweit höchste Diabetesrate.
- Sie sind in der Nähe von Phoenix beheimatet.

Der Datensatz

- Welche Daten enthält der Datensatz und wie sind diese kodiert?

```
> help(pima)
pima                  package:faraway                  R Documentation
Diabetes survey on Pima Indians
Description:
     The National Institute of Diabetes and Digestive and Kidney
     Diseases conducted a study on 768 adult female Pima Indians living
     near Phoenix.

Usage:
     data(pima)

Format:
```

```
     The dataset contains the following variables
     'pregnant' Number of times pregnant
     'glucose' Plasma glucose concentration at 2 hours in an oral
         glucose tolerance test
     'diastolic' Diastolic blood pressure (mm Hg)
     'triceps' Triceps skin fold thickness (mm)
     'insulin' 2-Hour serum insulin (mu U/ml)
     'bmi' Body mass index (weight in kg/(height in metres squared))
     'diabetes' Diabetes pedigree function
     'age' Age (years)
     'test' test whether the patient shows signs of diabetes (coded 0
         if negative, 1 if positive)

Source:
     The data may be obtained from  UCI Repository of machine learning
     databases at <URL:
     http://www.ics.uci.edu/\~mlearn/MLRepository.html>
```

Der erste Blick

```
> str(pima)
'data.frame': 768 obs. of  9 variables:
 $ pregnant : int  6 1 8 1 0 5 3 10 2 8 ...
 $ glucose  : int  148 85 183 89 137 116 78 115 197 125 ...
 $ diastolic: int  72 66 64 66 40 74 50 0 70 96 ...
 $ triceps  : int  35 29 0 23 35 0 32 0 45 0 ...
 $ insulin  : int  0 0 0 94 168 0 88 0 543 0 ...
 $ bmi      : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ diabetes : num  0.627 0.351 0.672 0.167 2.288 ...
 $ age      : int  50 31 32 21 33 30 26 29 53 54 ...
 $ test     : int  1 0 1 0 1 0 1 0 1 1 ...
```

- Man könnte 768 Beobachtungen noch einzeln durchgucken. Man kann es sich aber auch leichter machen! **Handarbeit ist schlecht!**

Einfache, datenbeschreibende Verfahren

Was fällt auf?

```
> summary(pima)
    pregnant         glucose        diastolic        triceps
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.0   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.0   1st Qu.: 0.00
 Median : 3.000   Median :117.0   Median : 72.0   Median :23.00
 Mean   : 3.845   Mean   :120.9   Mean   : 69.1   Mean   :20.54
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.0   3rd Qu.:32.00
 Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   :99.00
```

```
    insulin           bmi          diabetes          age
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
 Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
      test
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.3490
 3rd Qu.:1.0000
 Max.   :1.0000
```

Was fällt auf?

- 17 Schwangerschaften ist ungewöhnlich, aber nicht ausgeschlossen!
- Blutdruck 0 ist ungesund, ebenso BMI 0 ...

```
> pima$diastolic
....
```

Wie viele sind es nun genau?

ACHTUNG: wichtiger Trick!

```
> sum(pima$diastolic == 0)
[1] 35
```

- Vermutlich sind in der Studie fehlende Werte als 0 festgehalten worden.

Aufgabe 2

Aufgabe: Vollziehen Sie die bisherigen Schritte der Analyse des Datensatzes *pima* nach! Ersetzen Sie für alle Variablen die fehlenden Werte durch `NA`.

Daten "reparieren"

In der Arbeitskopie (!) fehlende Werte durch NA kodieren!

```
>pima$diastolic[ pima$diastolic == 0 ] <- NA
>pima$glucose[ pima$glucose == 0 ] <- NA
>pima$triceps [ pima$triceps == 0 ] <- NA
>pima$insulin [ pima$insulin == 0 ] <- NA
>pima$bmi [ pima$bmi == 0 ] <- NA ; summary(pima)
    pregnant         glucose        diastolic        triceps
 Min.   : 0.000   Min.   : 44.0   Min.   : 24.0   Min.   :  7.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.: 22.00
 Median : 3.000   Median :117.0   Median : 72.0   Median : 29.00
 Mean   : 3.845   Mean   :121.7   Mean   : 72.4   Mean   : 29.15
 3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.0   3rd Qu.: 36.00
 Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   : 99.00
                  NA's   :  5.0   NA's   : 35.0   NA's   :227.00
```

Weiterer Schwachpunkt der Daten:

- Die Variable `test` wird in der Zusammenfassung als numerischer Wert behandelt, obwohl es sich um eine kategorielle Variable handelt.

- In `R` werden solche Variablen `factor` genannt und können außerdem beschreibende Werte (Faktorstufen , *factor levels*) erhalten.

```
> pima$test <- factor(pima$test)
> levels(pima$test) <- c("negativ", "positiv")
> summary(pima$test)
negativ positiv
    500     268
```

- Allein an der Zusammenfassung der Daten kann man nunmehr keine Unregelmäßigkeiten mehr entdecken.

Weitere typische Probleme in Datensätzen

- Daten: 2009/19/01 oder 19.1.2009 oder 2009-01-19 etc.
  Unbedingt eine Vereinheitlichung durchführen!

- Zeiten: 16:15:35 Uhr oder 1615 Uhr oder 4:15 p.m. oder 4:15 pm
  Unbedingt dokumentieren lassen!

- Abkürzungen: m/w oder M/W, also Groß- und Kleinschreibung beachten!

- Spalten von Daten werden nicht als Zahlen erkannt, da der falsche Dezimaltrenner erwartet wird!

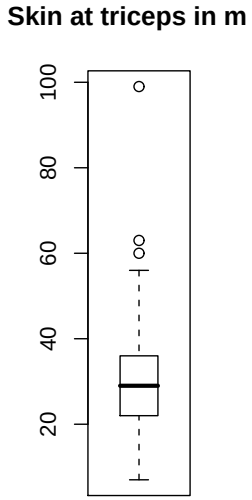
- Tausendertrenner werden verwendet.

- Zeichenketten dienen der Kodierung bestimmter Werte wie `NA`: nn, n.n., ung, etc.

Explorative Plots I

- Der *Boxplot*, auch *Box-and-Whisker plot*

- Beispiel: `boxplot(pima$triceps, main="Skin at triceps in mm")`



  - Der Boxplot wird neben (oder über) eine Skala der untersuchten Variablen gezeichnet
  - Dargestellt werden: Der Median (dicker Strich in der Box), das untere und das obere Quartil der Daten, diese bestimmen die Höhe (oder Breite) der Box, die Whiskers (gestrichelte Linien) mit der Länge max(1.5 Quartilsabstände, Abstand des Extrempunktes von der Box), sowie sogenannte Ausreißer, durch einzelne Symbole bezeichnete Datenpunkte, die außerhalb der Whiskers liegen.

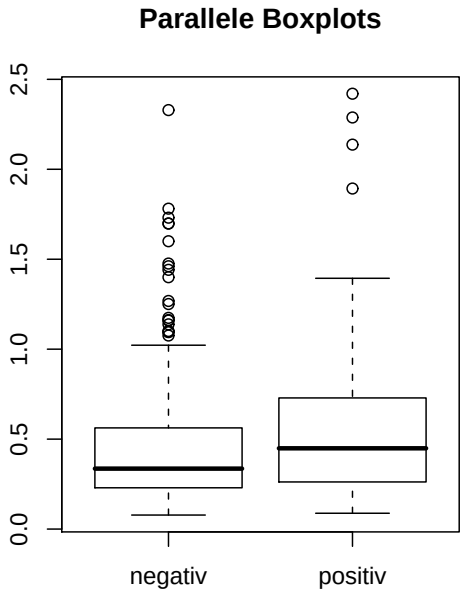
Interpretation des Boxplots

- Die Lage des Medians in der Box zeigt schön, ob die zugrundeliegende Verteilung symmetrisch oder schief ist.
- Unterstützt wird der Symmetrieeindruck durch die Länge der der Whisker.
- Die Größe der Box gibt einen Eindruck von Streuung der Daten.
- Da **alle** Datenpunkte eingezeichnet sind, erkennt man auch die Spannweite.
- Die Ausreißer kennzeichnen Datenpunkte, die evtl. noch mal auf Abnormitäten angesehen werden sollten.

Erweiterungen der Boxplotidee

- Die parallele Darstellung mehrerer Boxplots in einer Grafik ermöglicht den schnellen optischen Vergleich der Verteilungen von Untergruppen.

- Beispiel: `boxplot(diabetes ~ test , pima, main="Parallele Boxplots")`



  - Ein *notched boxplot* hat zusätzlich noch ein Konfidenzintervall für den Median eingezeichnet.
  - Manchmal gehen die *whiskers* auch bis zum 2.5% bzw. 97.5% Quantil.

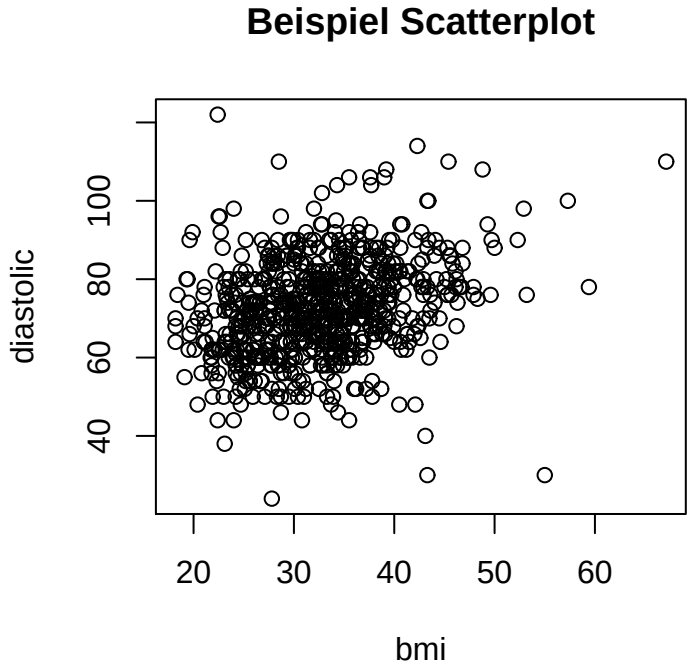
Exkurs: Verteilungsbezogene Funktionen in R

- Für viele Verteilungen gibt es Funktionen wie `pnorm, qnorm, dnorm, rnorm`, die das Ablesen aus Tabellen ersetzen bzw. der Verteilung entsprechende Zufallszahlen erzeugen. Die Verteilungsparameter können jeweils als Funktionsparameter angegeben werden. Hier am Beispiel für die Normalverteilung:

- `pnorm(q, mean=0, sd=1, ...)` Verteilungsfunktion (*probability*-),
- `dnorm(x, mean=0, sd=1, ...)` Dichtefunktion (*density*-),
- `qnorm(p, mean=0, sd=1, ...)` Quantilsfunktion (*quantile*-),
- `rnorm(n, mean=0, sd=1)` Zufallszahlenerzeugung (*randomnumber*-).
- Alle üblichen Verteilungen liegen in R bereits derart vor.

Explorative Plots II

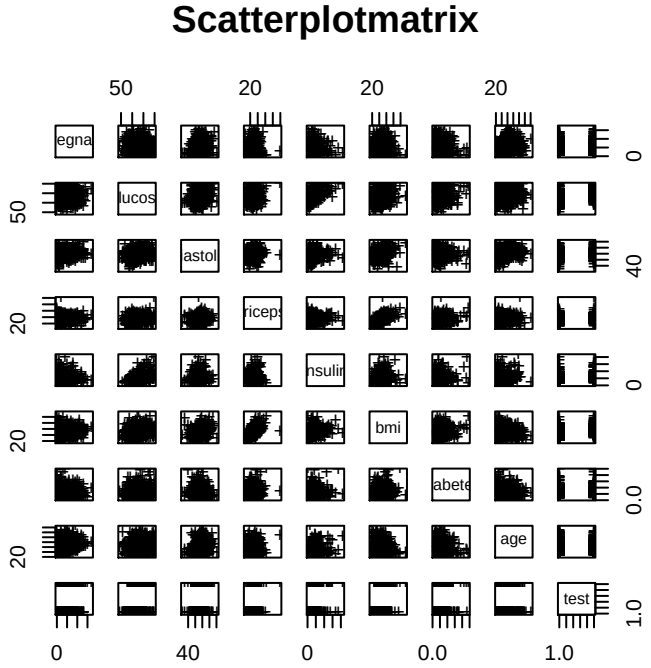
- Streudiagramm und -matrix
- Beispiel: `plot(diastolic ~ bmi, pima, main="Beispiel Scatterplot")`



- Abtragen zweier Variablen gegeneinander, als wären sie abhängige und unabhängige Größe
- Idee: leichtes Erkennen von funktionalen Zusammenhängen

Explorative Plots II

- Streudiagramm und -matrix
- Beispiel: `plot(pima, main="Beispiel Scatterplotmatrix")`



- Auf einen Blick alle Paare von Variablen!
- Natürlich keine Details, aber z.B. potentielle Ausreißer sind gut sichtbar.

Aufgabe 2

- Fertigen Sie mit R Boxplots für die anderen im Datensatz von Faraway enthaltenen Variablen an. Fällt Ihnen etwas auf?

- Fertigen Sie parallele Boxplots für die verschiedenen Variablen, wie im Beispiel getrennt nach den Untergruppen für Test positiv bzw. Test negativ, an. Fallen Unterschiede in den Gruppen auf?

- Angenommen, Sie haben zwei sehr große Stichproben, einmal aus der Standardnormalverteilung und einmal aus der Exponentialverteilung mit $\lambda = 10$. Welchen Anteil der Daten erwarten Sie jeweils außerhalb der Whiskers? Welche Werte erwarten Sie für Median, unteres und oberes Quartil und Interquartilsabstand.

Hinweis zu Aufgabe 2

- Die letzte Aufgabe können Sie entweder mit Mitteln aus Statistik II theoretisch lösen oder Sie schauen sich die Zufallszahlenerzeugung in `R` an (`rnorm` etc.) und lösen die Aufgabe empirisch.

Aufgabe 3

- Fertigen Sie die Scatterplotmatrix für den Datensatz `pima` wie im Beispiel gegeben an! Finden Sie beachtenswerte Punkte?

Explorative Plots II

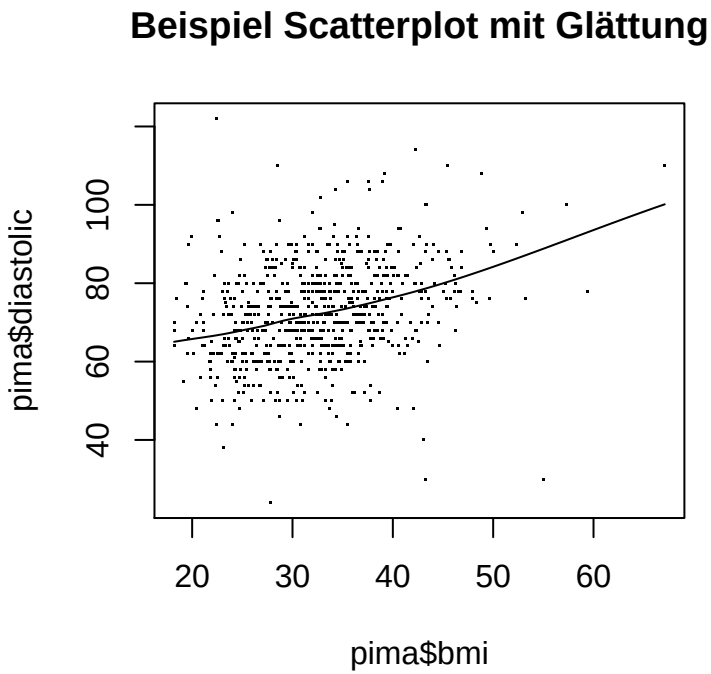
- Streudiagramme mit Glättungskurve

- Glättungskurven sind explorative Werkzeuge, die die Struktur eines Datensatzes ‘zeigen‘ wollen, ohne eine Modellannahme zu treffen, wie in der Regression nötig (‘nicht-parametrisches Verfahren‘)

- Beispiel: (`attach()` und `detach()` fügen einen Dataframe in den Suchpfad für R-Objekte ein, bzw. entfernen den Dataframe wieder.)

```
> attach(pima)
> scatter.smooth(bmi, diastolic,
      main="Beispiel Scatterplot mit Glättung", pch='.')
> detach("pima")
```

Explorative Plots II



- Abtragen zweier Variablen gegeneinander, als wären sie abhängige und unabhängige Größe, zusätzlich wird eine Glättungskurve eingezeichnet.
- Keine Modellannahme nötig!
- Verfahren heißen `loess()` bzw. `lowess()` und nutzen lokal angepasste polynomiale Modelle.

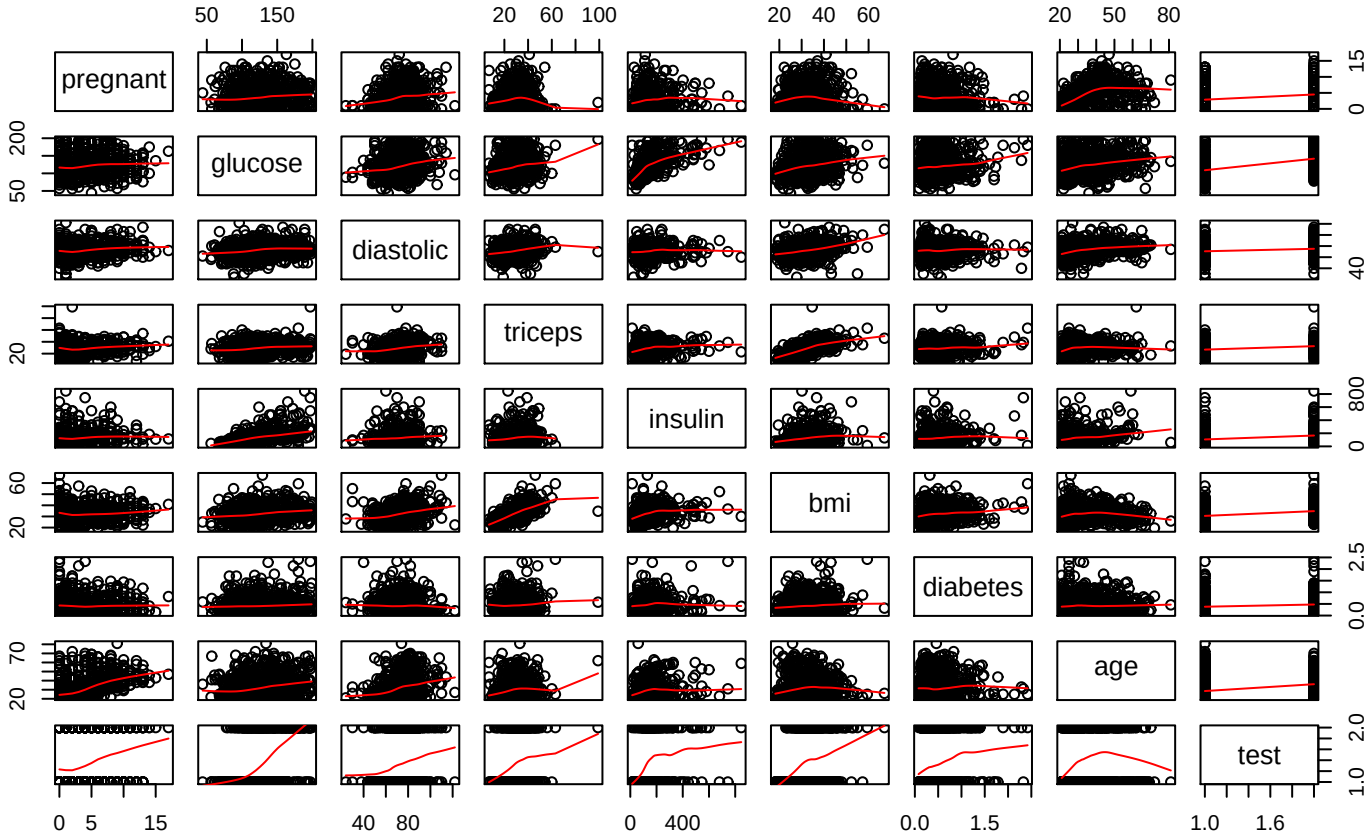
- Literatur: Cleveland, W. S., Grosse, E., Shyu, W. M. (1992): Local regression models.

Explorative Plots II

- Verfeinertes Beispiel für Scatterplot-Matrizen.
- Zusätzlich in jedem Plot noch eine Glättungskurve eingezeichnet.
- Die Glättungsfunktion kann selbst definiert werden.

```
> pairs( pima,
    panel= function( x, y) { panel.smooth( x, y, span= 2/3) })
```

Beispiel `pairs()`

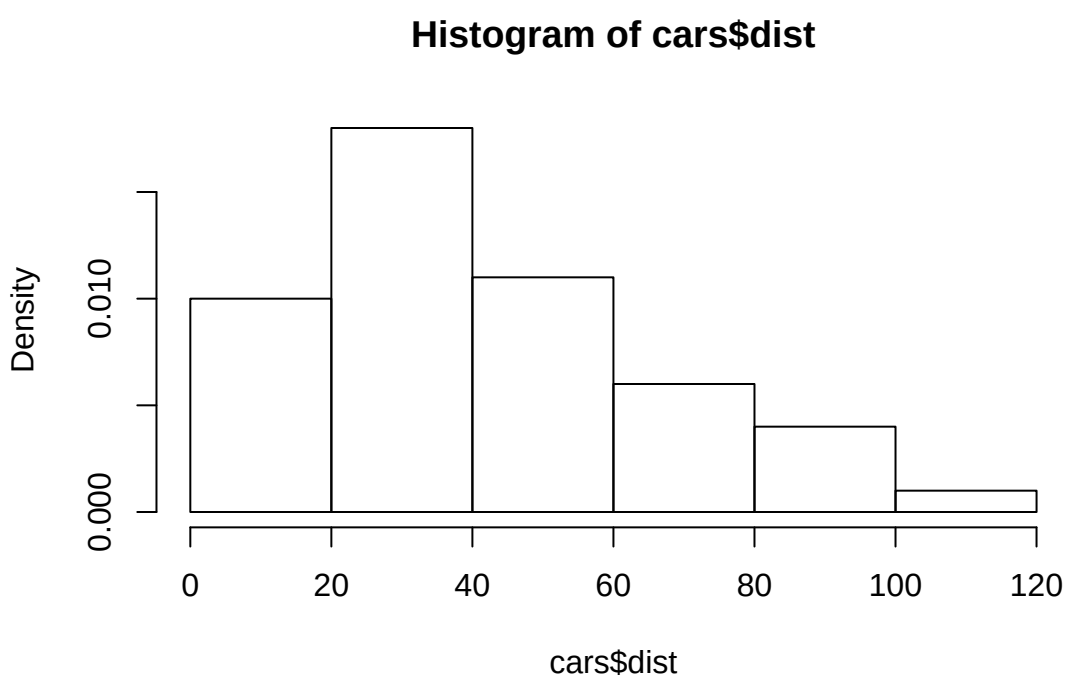


Aufgabe 4

Laden Sie den Datensatz `cars` und erzeugen Sie einen entsprechenden Scatterplot mit Glättungsfunktion! Was erkennen Sie an der Grafik?

Einige weitere explorative Plots (Statistik I)

- Histogramm, zahlreiche Optionen: `hist(cars$dist, freq=FALSE)`

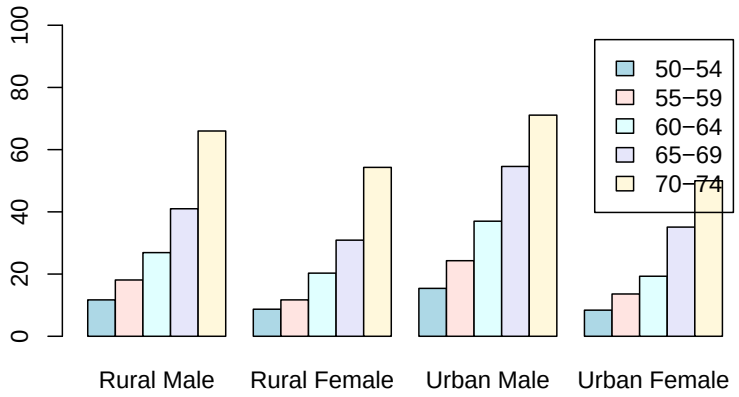


- Die wichtigsten: `freq` Häufigkeiten oder rel. Häufigkeiten?
- `breaks`: Wo sind die Klassengrenzen?
- **Nicht** der richtige Plot, um Anzahlen gegenüber zu stellen!

Barplot bzw. Balkendiagramm

- Als einfache Balken oder als gestapelte Balken oder nebeneinander gestellte Balken

```
barplot(VADeaths, beside = TRUE, col = c("lightblue",
        "mistyrose", "lightcyan", "lavender", "cornsilk"),
        legend = rownames(VADeaths), ylim = c(0, 100))
```



- Hier nach Klassen nebeneinander angeordnete Balken Daten (`?VADeaths`)
- Die Höhe der Balken ist proportional zur darzustellenden Zahl.

Stamm- und Blatt bzw. stem-and-leaf plot

- Halbgrafisches Verfahren ähnlich einem Histogramm, allerdings gehen die Werte in den Klassen nicht verloren

```
> stem(iris$Sepal.Length, scale=0.5)
  The decimal point is at the |
  4 | 3444
  4 | 56666778888999999
  5 | 00000000001111111122223444444
  5 | 55555556666667777777888888999
  6 | 00000011111122223333333334444444
  6 | 5555566777777778889999
  7 | 0122234
  7 | 677779
```

- Der Parameter `scale` steuert die Auflösung des Plots.

Stamm- und Blatt bzw. stem-and-leaf plot

- Konstruktion:
  - Festlegen, wieviele führende Stellen der Zahl den Stamm links vom „|“ bilden sollen.
  - Runden aller Ergebnisse auf die nächste Stelle.
  - Die gerundeten Ziffern hinter den zugehörigen Stamm eintragen.

- Interpretation:
  - Wie beim Histogramm identifiziert man die häufigen Klassen.
  - Zusätzliche Information gegenüber dem Histogramm, da bis auf Rundung der komplette Datensatz dargestellt wird.

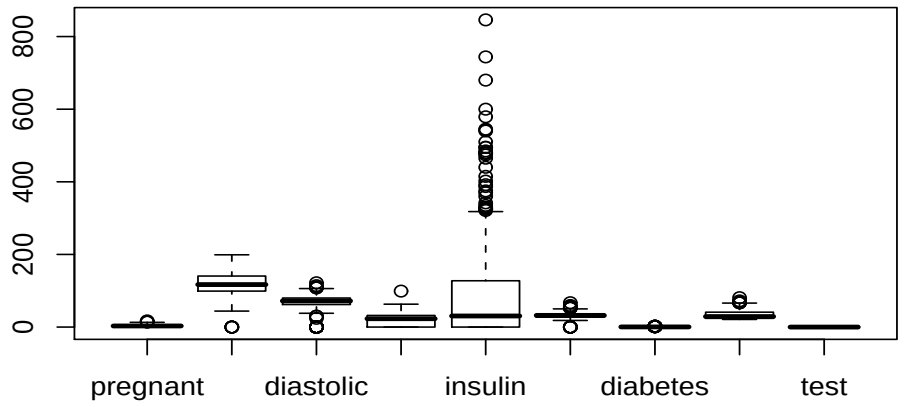
Weitere explorative Grafiktypen

- `dotchart()`: Art minimalistischer, auf der Seite liegender Balkenplot.
- `pie()`: Tortendiagramm, Häufigkeiten proportional zu den Sektorwinkeln.
- `mosaicplot()`: Ein Häufigkeitsplot für bivariate Daten. Die Interpretation Bedarf einiger Übung. Flächen sind Proportional zur relativen Häufigkeit der Zelle.
- `stars()`: Für multivariate Daten. Die Länge der Strahlen gibt die Koordianten an.
- und viele, viele mehr ...

Lösung Aufgabe 2

- Fertigen Sie mit R Boxplots für die anderen im Datensatz von Faraway enthaltenen Variablen an. Fällt Ihnen etwas auf?

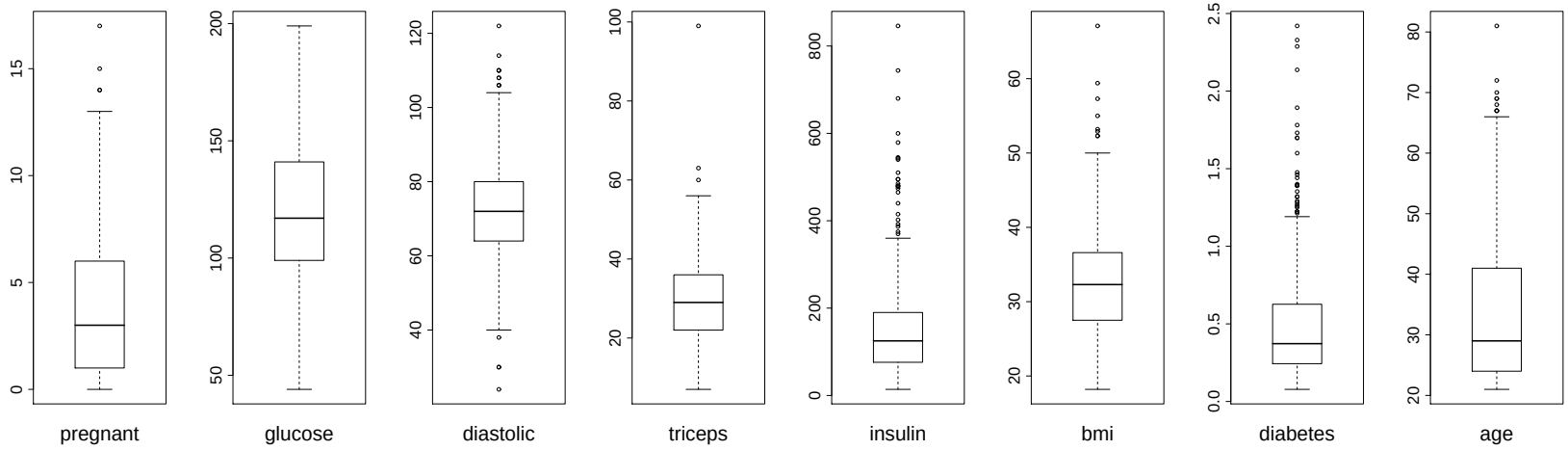
```
> boxplot(pima)
```



- unklare Darstellung!

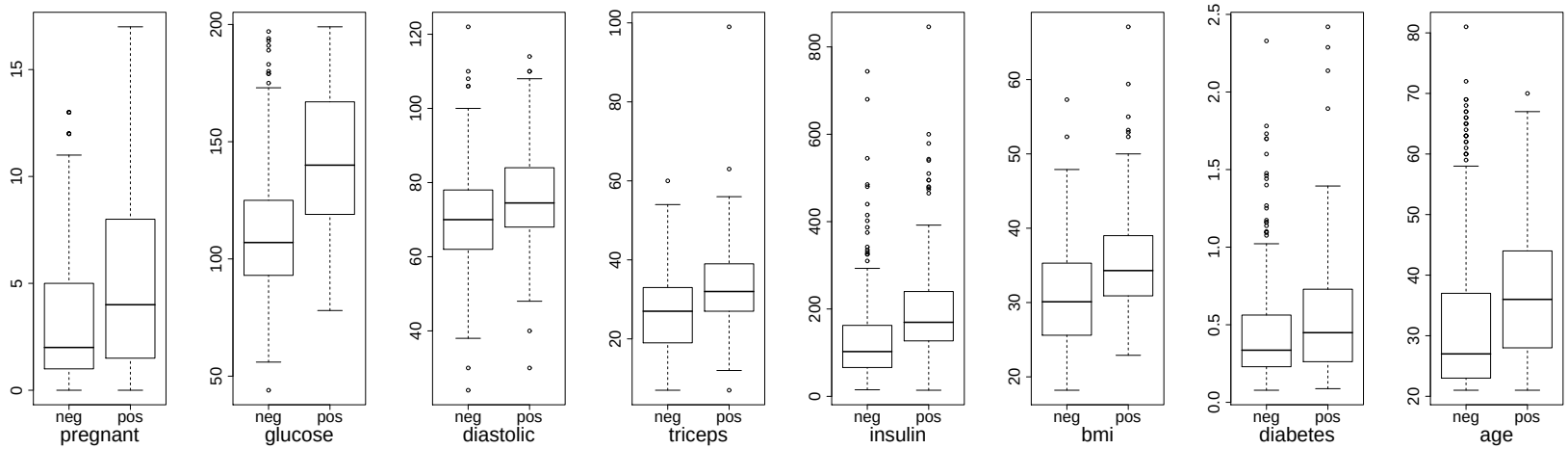
Lösung Aufgabe 2

```
for (variable in 1:8)
  {
     postscript(file=paste("boxplot-",names(pima)[variable],
          ".ps", sep=""), horizontal=FALSE, height=10, width=4)
     boxplot(pima[,variable], xlab=names(pima)[variable],
            cex.lab=2.5, cex.axis=2) ; dev.off()
  }
```



Lösung Aufgabe 2

- Fertigen Sie parallele Boxplots für die verschiedenen Variablen, wie im Beispiel getrennt nach den Untergruppen für Test positiv bzw. Test negativ, an. Fallen Unterschiede in den Gruppen auf?




```
boxplot(pima[,variable] ~ pima$test ,
        xlab=names(pima)[variable], cex.lab=2.5, cex.axis=2)
```

Lösung Aufgabe 2

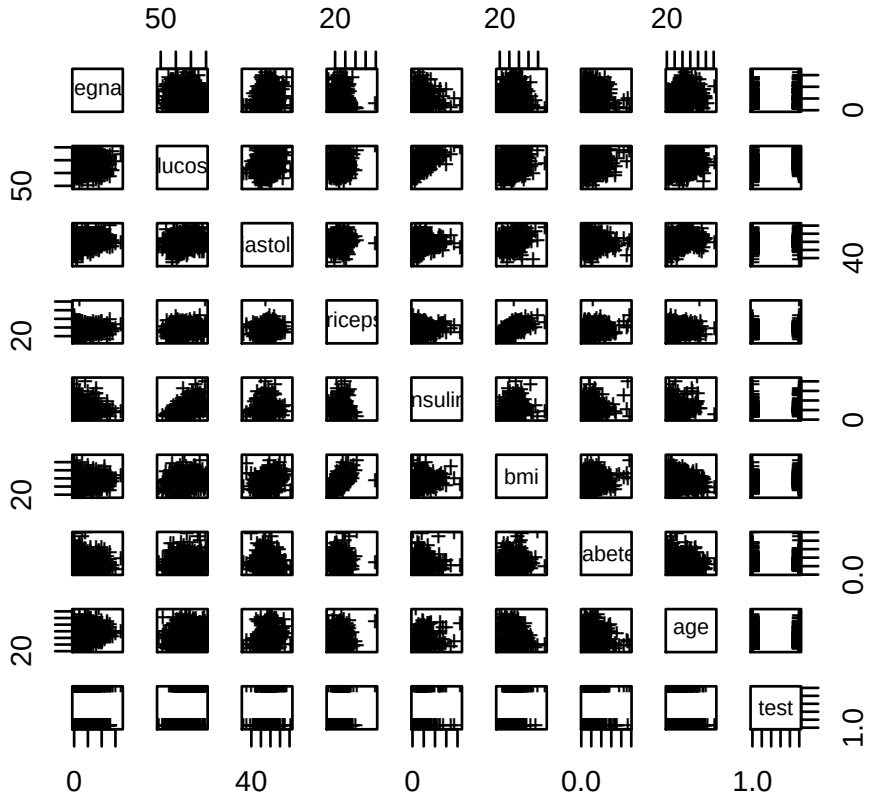
- Angenommen, Sie haben zwei sehr große Stichproben, einmal aus der Standardnormalverteilung und einmal aus der Exponentialverteilung mit $\lambda = 10$. Welchen Anteil der Daten erwarten Sie jeweils außerhalb der Whiskers? Welche Werte erwarten Sie für Median, unteres und oberes Quartil und Interquartilsabstand.

- Standardnormalverteilung
  - Median: `qnorm(0.5) = 0` , $q_{0.25} = \texttt{qnorm}(0.25) = -0.674, q_{0.75} = 0.674$, Interquartilsabstand $q_{0.75} - q_{0.25} = 1.349$
  - Ende unterer *whisker* $q_{0.25} - 1.5 \cdot 1.349 = -2.698$, oberer *whisker* 2.698, Anteil außerhalb der *whisker* `2*pnorm(-2.698) = 0.0069`, also ca. 0.7 %

- Exponentialverteilung, $\lambda = 10$
  - Median: `qexp(0.5, rate=10) = 0.069` , $q_{0.25} = \texttt{qexp}(0.25, \texttt{rate} = 10) = 0.029, q_{0.75} = 0.139$, Interquartilsabstand $q_{0.75} - q_{0.25} = 0.11$
  - Ende unterer *whisker* $q_{0.25} - 1.5 \cdot 0.11 = -0.136$, (!) oberer *whisker* 0.304, Anteil außerhalb der *whisker* `1-pexp(0.304, rate=10) = 0.048`, also ca. 4.8 %

Lösung Aufgabe 3

Scatterplotmatrix



Lösung Aufgabe 4

Laden Sie den Datensatz `cars` und erzeugen Sie einen entsprechenden Scatterplot mit Glättungsfunktion! Was fällt auf?

```
data(cars)
names(cars)
scatter.smooth(dist~ speed)
detach(cars)
```

Die Grafik legt nichtlinearen Zusammenhang nahe.

Der Begriff der Ordnungsstatistiken $x_{(i)}$

- Zu jeder Stichprobe paarweise verschiedener $x_i, i = 1, \ldots, n$ gehört die Folge der Ordnungsstatistiken $x_{(i)}, i = 1, \ldots, n$, die die aufsteigend sortierte Folge der Beobachtungen bezeichnet. Die erste Ordnungsstatistik $x_{(1)}$ ist gleich dem Minimum der Beobachtungen, $x_{(n)}$ gleich dem Maximum.

- Es gilt empirisch
$$F(x_{(i)}) = \frac{i}{n}.$$

- Es lassen sich natürlich entsprechende Zufallsvariablen $X_{(i)}$ für die Ordungsstatistiken definieren.

- Nach dem Satz von Gliwenko-Cantelli konvergiert die empirische Verteilungsfunktion der Stichprobe $x_i, i = 1, \ldots, n$ an jeder Stetigkeitsstelle

von F, der Verteilung der X_i, mit dem Stichprobenumfang n gegen die wahre Verteilung F.

- Damit gilt für hinreichend große n

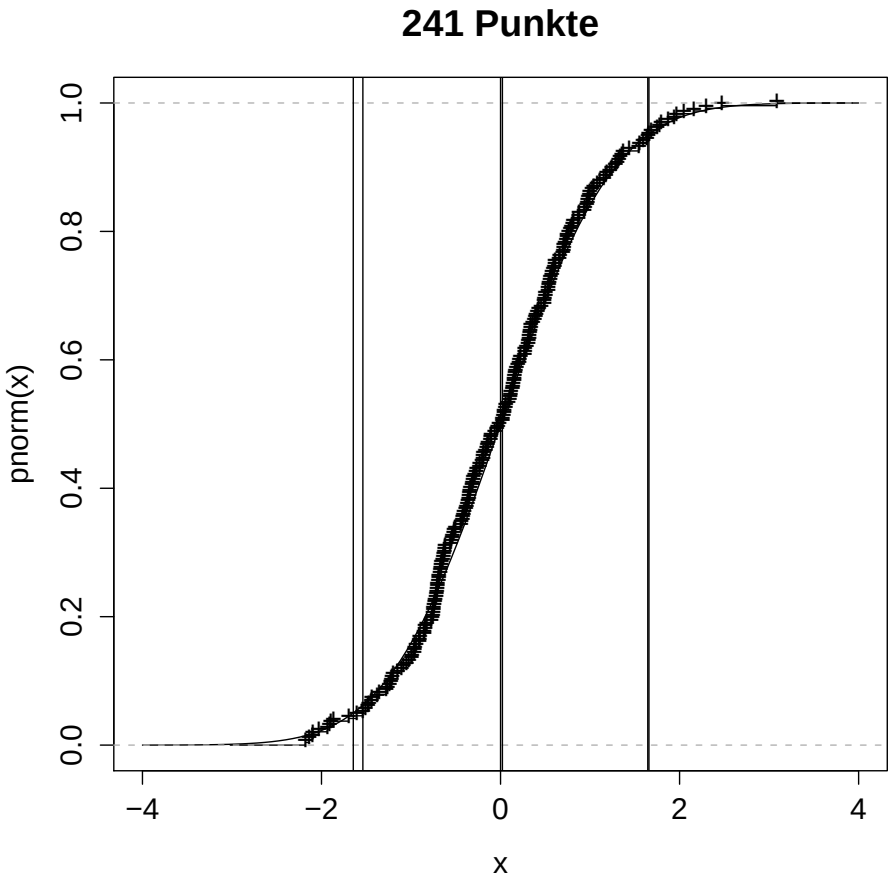
$$x_{(i)} \approx F^{-1}(\frac{i}{n}).$$

- Diese Eigenschaft wird ausgenutzt, um einen "grafischen Anpassungstest" zu entwickeln.

Demo der Approximation

```
par(ask=TRUE)
samplesize <- 1
while ( samplesize <250) {
  if (samplesize > 100) par(ask=FALSE)
  curve(pnorm(x), -4,4, main=paste(samplesize, "Punkte"))
  sample <- sort(rnorm(samplesize))
  lines(ecdf(sample), pch="+")
  abline(v=qnorm(c(0.05,0.5,0.95)))
  abline(v=c(sample[round(samplesize/20)],
           median(sample),
           sample[round(19*samplesize/20)], col = "red"  )  )
  samplesize <- samplesize +10
}
dev.copy2eps("approxdemo.eps")
```

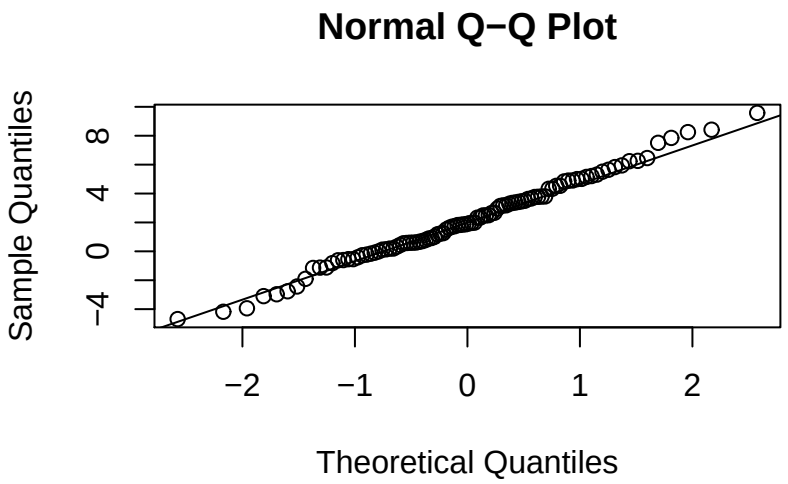
Demo der Approximation

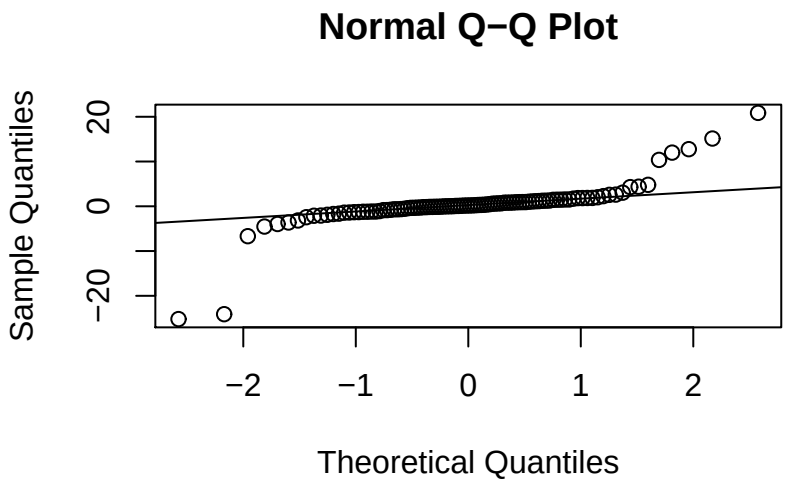


Der Q-Q Plot

- Beim Q-Q Plot werden die theoretischen Quantile einer Verteilung und die empirischen Quantile einer Stichprobe gegeneinander geplottet. Unter der Nullhypothese bildet dieser Graph eine Gerade.

- linker Plot: `x <- rnorm(100, mean=2, sd=3) ; qqnorm(x) ; qqline(x)`
  rechter Plot: `x <- rcauchy(100) ; qqnorm(x) ; qqline(x)`





Regression

- Von Regression spricht man immer, wenn man eine Zielgröße $Y \in \mathcal{R}$ (abhängige Variable, Antwortvariable, response, output, dependent) durch eine oder mehrere Einstellgrößen $X_1, X_2, \ldots, X_p$ (unabhängige Variable, Einstellgröße, erklärende Variable, predictor, input, independent) durch einen unterstellten funktionalen Zusammenhang $Y = f(X)$ erklären oder modellieren möchte. Bei $p = 1$ spricht man von *einfacher Regression*, bei $p > 1$ von *multipler Regression*. Gibt es mehr als eine Zielgröße Y, so spricht man von *multivariater Regression*.

- Sind X und Y reellwertig, so liegt eine einfache Regression, wie in Statistik I+II, vor.

- Ist ein X_i qualitativ, so gelangt man zur (Ko-)varianzanalyse (ANOVA).

Das lineare Modell

- Ganz allgemein wird ein funktionaler Zusammenhang $Y = f(X_1, \ldots, X_p) + \varepsilon$ postuliert.
  Normalerweise ist f nicht bekannt und folglich nicht schätzbar.

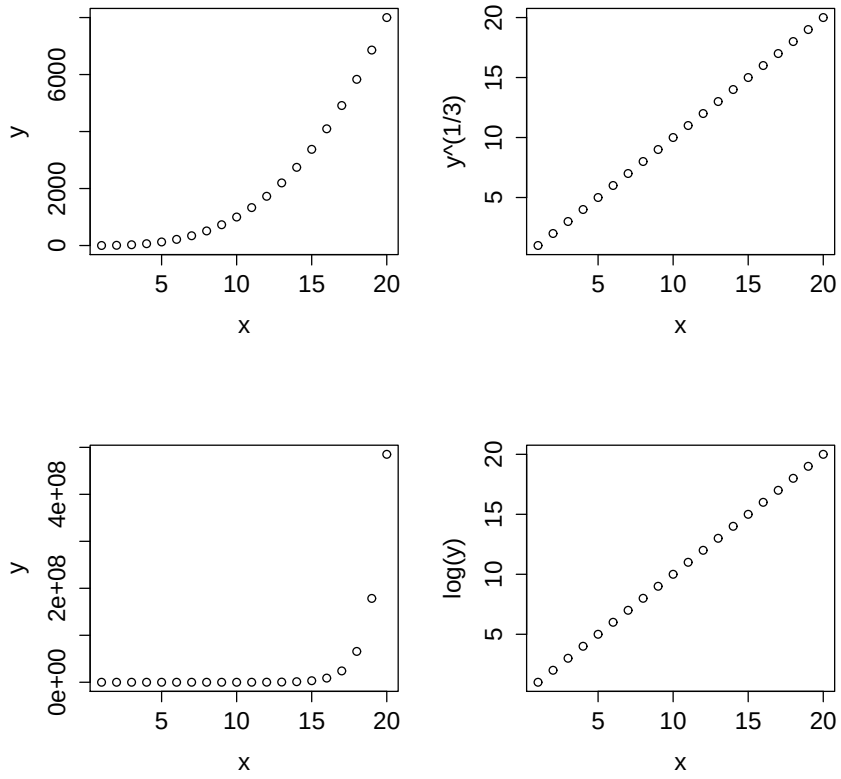
- Beschränkung auf lineare Modelle $Y = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p + \varepsilon$

- Linear bezieht sich darauf, dass der Einfluß der β_i linear ist, nicht auf die Einflußgrößen selbst. Z.B. ist $Y = \beta \log(X) + \varepsilon$ ein lineares Modell oder auch $Y = \beta X^2 + \varepsilon$.

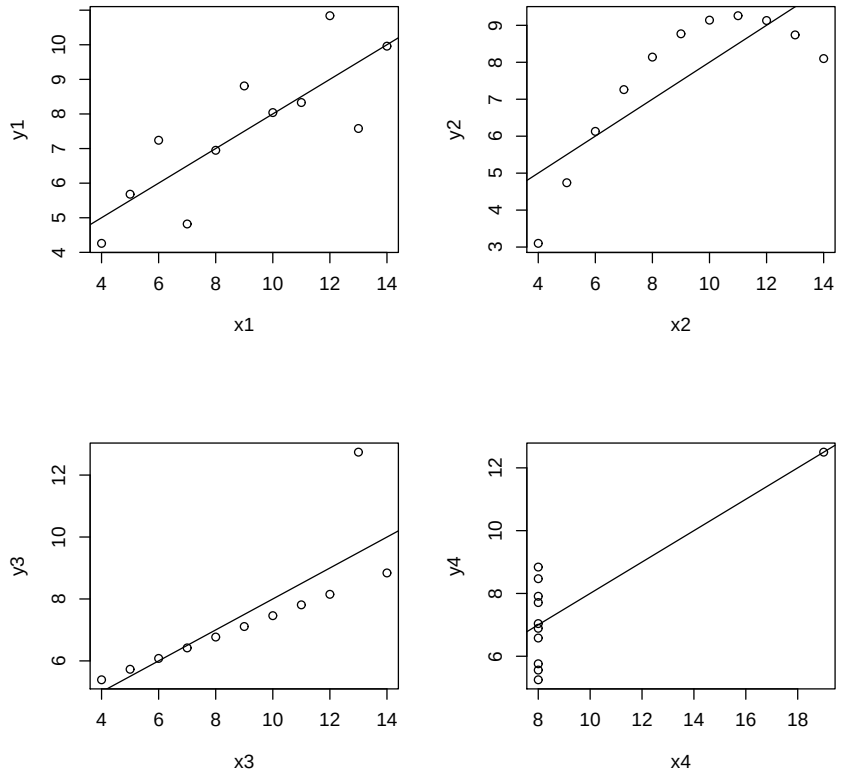
- Die Einschränkung auf lineare Modell ist in der Praxis nicht sehr streng. Manche Funktionen können in eine lineare Form transformiert werden und bei hinreichend glatten Funktionen ist die lineare Form oft eine gute Approximation (Taylor-Approximation).

Zielgrößentransformationen

- Übliche Transformationen sind $Y \rightarrow ln(Y)$ oder $Y \rightarrow Y^{\beta}$



Optischer Check: *Anscombe's quartet*



Alle Datensätze haben dieselben Werte für Mittel, Varianz und sogar dieselben Regressionsgeraden! (`data(anscombe)`)

Einfache lineare Regression

- Regression, also die Erklärung einer *Zielgröße*, auch abhängige Variable, durch eine *Einflußgröße* (auch Einstellgröße, Unabhängige) ist sicherlich **die** Methode der Statistik schlechthin.

- Generalvoraussetzung ab jetzt: $(x_1, y_1), \ldots, (x_n, y_n)$ sind eine gegebene Stichprobe vom Umfang n. Hierbei bezeichnet X die Einflußgröße und Y die Zielgröße, jeweils aus $\mathcal{R}$.

- Theorie bekannt aus Statistik, hier die Umsetzung in `R`.

- Beispiel `pima`-Daten. Aus der Scatterplotmatrix ist z.B. der Zusammenhang von `diastolic` und `bmi` interessant.

Lineare Regression in R

- Ziel: Schätzung von Parametern im linearen Modell
$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon.$$

- Ergebnis: Modell (*fit*)
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p.$$

- Residualvektor $\varepsilon = (y - \hat{y})_{i=1,\ldots,n}$ mit der Fehlervarianz $\hat{\sigma}_\varepsilon$.

- β_0 heißt Achsenabschnitt, die β_i heißen Regressionskoeffizienten, β der Koeffizientenvektor.

- X heißt *Designmatrix*.

- Das Ganze in R: `lm( Y ~ X [, dataframe])`

Streuungszerlegung im linearen Modell

- Seien $SQT = \sum_1^n (y_i - \bar{y})^2$ *sum of sqares total* oder Gesamtstreuung,
- $SQE = \sum_1^n (\hat{y}_i - \bar{y})^2$ *sum of squares explained* oder erklärte Streuung sowie
- $SQR = \sum_1^n (y_i - \hat{y}_i)^2$ *sum of squared residuals* oder Reststreuung.
- Dann gilt:

$$SQT = SQE + SQR!$$

(Aufgabe 5: bitte nachrechnen!)

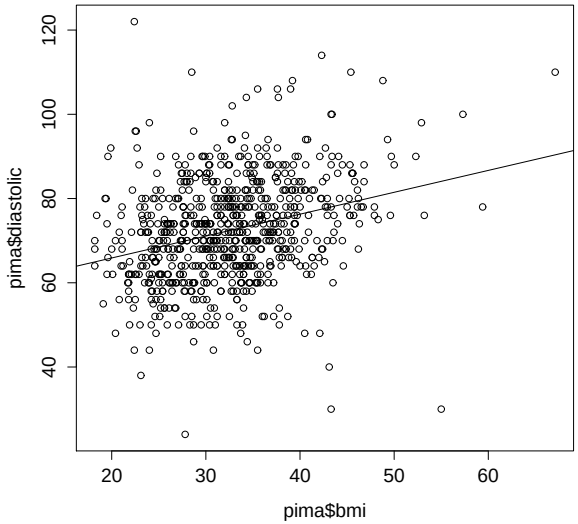
Einfache lineare Regression in R (p=1)

```
> lm(diastolic ~ bmi, pima)
Call:
lm(formula = diastolic ~ bmi, data = pima)
Coefficients:
(Intercept)          bmi
    55.4869       0.5199
```

- Kommando `lm()` (linear model)
- Liefert die bekannten Schätzer (und mehr)
- Die Anzeige ist **nicht** das Ergebnis der Regression in R, sondern die Methode `print()` angewendet auf ein Objekt vom Typ Regression.
- Das Ergebnis eines `lm()` Aufrufs ist ein *Objekt der Klasse* `lm`

```
> result <- lm(diastolic ~ bmi, pima)
> names(result)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "na.action"     "xlevels"       "call"          "terms"
[13] "model"
```

```
> plot(pima$bmi,
     pima$diastolic)
> abline(result)
```



```
> str(result)

List of 13
 $ coefficients : Named num [1:2] 55.49  0.52
  ..- attr(*, "names")= chr [1:2] "(Intercept)" "bmi"
 $ residuals    : Named num [1:729]  -0.955  -3.316  -3.600  -4.096
  ..- attr(*, "names")= chr [1:729] "1" "2" "3" "4" ...
 $ effects      : Named num [1:729] -1953.93    96.58    -3.41    -
  ..- attr(*, "names")= chr [1:729] "(Intercept)" "bmi" "" "" ...
 $ rank         : int 2
 $ fitted.values: Named num [1:729] 73.0 69.3 67.6 70.1 77.9 ...
  ..- attr(*, "names")= chr [1:729] "1" "2" "3" "4" ...
 $ assign       : int [1:2] 0 1
 $ qr           :List of 5
  ..$ qr   : num [1:729, 1:2] -27.000   0.037   0.037   0.037   0.0
```

```
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:729] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "(Intercept)" "bmi"
  .. ..- attr(*, "assign")= int [1:2] 0 1
  ..$ qraux: num [1:2] 1.04 1.03
  ..$ pivot: int [1:2] 1 2
  ..$ tol  : num 1e-07
  ..$ rank : int 2
  ..- attr(*, "class")= chr "qr"
 $ df.residual  : int 727
 $ na.action    :Class 'omit'  Named int [1:39] 8 10 16 50 61 79 82
  .. ..- attr(*, "names")= chr [1:39] "8" "10" "16" "50" ...
 $ xlevels      : list()
 $ call         : language lm(formula = diastolic ~ bmi, data = pir
 $ terms        :Classes 'terms', 'formula' length 3 diastolic ~ br
  .. ..- attr(*, "variables")= language list(diastolic, bmi)
```

```
  .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..$ : chr [1:2] "diastolic" "bmi"
  .. .. .. ..$ : chr "bmi"
  .. ..- attr(*, "term.labels")= chr "bmi"
  .. ..- attr(*, "order")= int 1
  .. ..- attr(*, "intercept")= int 1
  .. ..- attr(*, "response")= int 1
  .. ..- attr(*, ".Environment")=<R_GlobalEnv>
  .. ..- attr(*, "predvars")= language list(diastolic, bmi)
  .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric
  .. .. ..- attr(*, "names")= chr [1:2] "diastolic" "bmi"
 $ model         :'data.frame': 729 obs. of  2 variables:
  ..$ diastolic: int [1:729] 72 66 64 66 40 74 50 70 92 74 ...
  ..$ bmi      : num [1:729] 33.6 26.6 23.3 28.1 43.1 25.6 31 30.5
  ..- attr(*, "terms")=Classes 'terms', 'formula' length 3 diastoli
```

```
  .. .. ..- attr(*, "variables")= language list(diastolic, bmi)
  .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:2] "diastolic" "bmi"
  .. .. .. .. ..$ : chr "bmi"
  .. .. ..- attr(*, "term.labels")= chr "bmi"
  .. .. ..- attr(*, "order")= int 1
  .. .. ..- attr(*, "intercept")= int 1
  .. .. ..- attr(*, "response")= int 1
  .. .. ..- attr(*, ".Environment")=<R_GlobalEnv>
  .. .. ..- attr(*, "predvars")= language list(diastolic, bmi)
  .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "nume
  .. .. .. ..- attr(*, "names")= chr [1:2] "diastolic" "bmi"
  ..- attr(*, "na.action")=Class 'omit'  Named int [1:39] 8 10 16 5
  .. .. ..- attr(*, "names")= chr [1:39] "8" "10" "16" "50" ...
 - attr(*, "class")= chr "lm"
```

- Wichtigste Methode `summary()`

```
> summary(result)
Call:
lm(formula = diastolic ~ bmi, data = pima)
Residuals:
     Min       1Q   Median       3Q      Max
-54.0807  -7.6278  -0.3313   7.2619  54.8676
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.48694    2.11810  26.197  < 2e-16 ***
bmi          0.51989    0.06382   8.147 1.63e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 11.86 on 727 degrees of freedom
  (39 observations deleted due to missingness)
Multiple R-squared: 0.08365,Adjusted R-squared: 0.08239
F-statistic: 66.37 on 1 and 727 DF,  p-value: 1.630e-15
```

Interpretation der `summary`

- Zuerst steht die angewendete Modellgleichung.
- Dann die *five-number-summary* des Fehlervektors.
- Danach eine Tabelle mit je einer Zeile je geschätztem Parameter β_i.
- Für jeden Parameter steht in der Zeile der Variablenname, die Schätzung $\hat{\beta}_i$, die Standardabweichung dieses Schätzers, die Teststatistik, die sich daraus ergibt und der p-Wert unter der Nullhypothese $\beta_i = 0$.
- In der letzten Spalte finden sich die "Sternchen". Dort kann man für die üblichen Niveaus 10%, 5%, 1% und 0.1% direkt die Signifikanz eines entsprechenden Tests ablesen.

Interpretation der `summary`

- Weiter wird der Schätzer $\hat{\sigma}_\varepsilon$ mit den zugehörigen Freiheitsgraden angegeben und
- es wird auf die Anzahl von *missing values* hingewiesen.
- Abschließend sind noch das (multiple) Bestimmtheitsmaß R^2 bzw. R^2_{adj} und die F-Statisik zum sogenannte *Goodness-of-fit-test* angegeben.

Der p-Wert

- Die Spalte `Pr(> |t|)` gibt den sogenannte p-Wert zur Teststatistik an.

- Zur Erinnerung: Bei einem statistischen Test wird eine Hypothese $\mathcal{H}_0$ verworfen, wenn für eine realisierte Teststatistik T gilt, dass unter der der Nullhypothese die Wahrscheinlichkeit einer Realisierung in der gemessenen Größenordnung kleiner oder gleich dem festgelegten Niveau α ist. Dazu vergleicht man das zur Hypothese gehörende Quantil mit der beobachteten Teststatistik und entscheidet entsprechend.

- Dabei geht die Information verloren, wie nah die Realisierung an der kritischen Grenze beobachtet wurde.

- Der `p-Wert` gibt nun genau das Niveau eines Testes an, bei dem Teststatistik und kritischer Wert exakt zusammen fallen würden.

Das Bestimmtheitsmaß R^2

- In der einfachen Regression (eine Einflußgröße) ist das Bestimmtheitsmaß R^2 definiert als
$$R^2 = 1 - \frac{SSR}{SST}.$$
- Man kann zeigen: $R^2 = r_{XY}^2$, wobei r_{XY} den empirischen Korrelationskoeffizienten bezeichnet.
- Werte liegen zwischen 0 (Modell erklärt keinen Varianzanteil) und 1 (Modell erklärt die Varianz vollständig)
- Multiples und adjustiertes R^2 werden bei der multiplen Regression betrachtet.

Der *Goodness-of-fit*-Test

- Heißt auch der *Overall-F*-Test.
- Überprüft wird die Hypothese H_0

$$H_0 : \beta_i = 0 \text{ für alle } i \quad \text{gegen} \quad H_1 : \beta_j \neq 0 \text{ für mindestens ein } j.$$

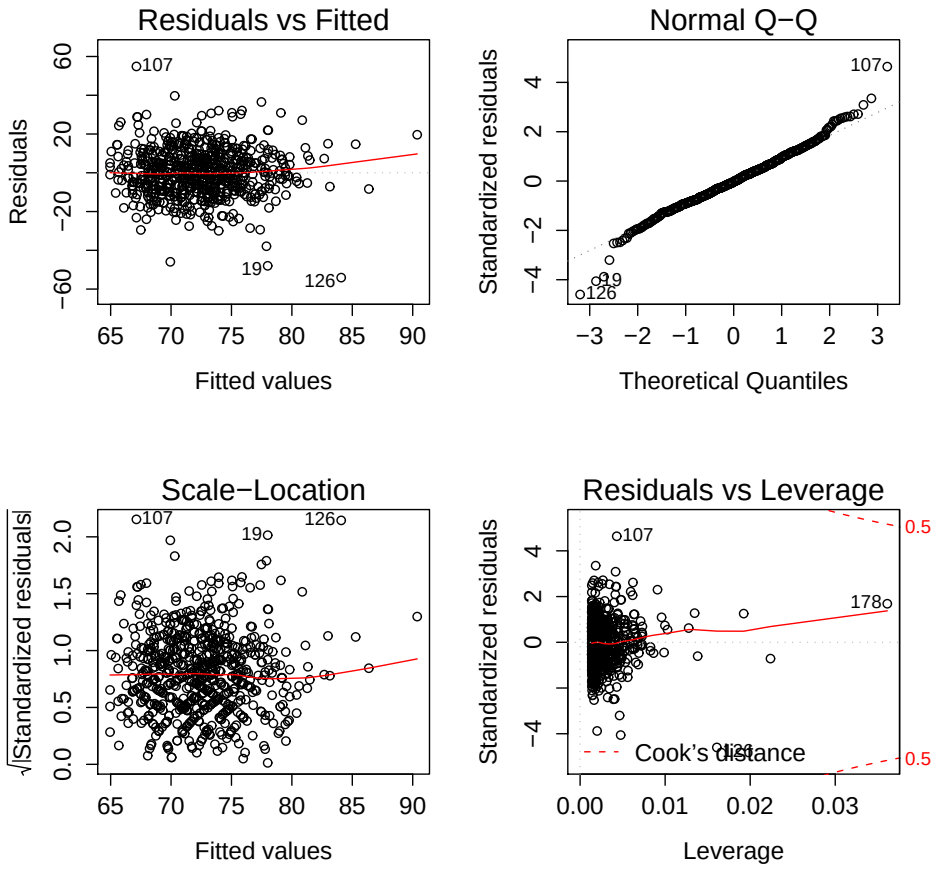
- Die Teststatistik ist in diesem Fall:

$$F = \frac{R^2}{1-R^2}\frac{n-p-1}{p} = \frac{SQE}{SQR}\frac{n-p-1}{p} \sim F(p, n-p-1) \text{ unter } H_0.$$

- Etwas irreführender Name, es wird getestet, ob irgendeiner der Regressoren signifikanten Einfluß hat.

Residualanalyse – diagnostische Plots

- Ergebnis von `plot(result)` sind *diagnostische Plots* zur optischen Beurteilung der Angemessenheit der Regression.
- In der linke Spalte sind Plots zur Beurteilung der Homoskedastizität (oben der sog. Tukey-Anscombe-Plot).
- Rechts oben der Q-Q Plot.
- Rechts unten ein *leverage* Plot (Einfluß einer Beobachtung).



Zusammenfassung: Wann erscheint die Regression angemessen?

- Die Regressionsgerade muss mitten durch die Punkte führen.
- Die Residualplots deuten nicht auf Heteroskedastizität hin.
- Der Q-Q Plot zeigt keine nennenswerten Abweichungen von der Normalverteilungsannahme.
- Dann können die signifikanten Faktoren interpretiert werden.

Literatur: Fahrmeier et al. Statistik, Springer

Aufgabe 6

Suchen Sie sich ein Variablenpaar in `pima`, bei dem Sie den linearen Zusammehang überprüfen wollen. Vollziehen sie die vorgestellten Schritte der einfachen linearen Regression nach ! Bringen Sie Analyseergebnisse und Graphen in die Textverarbeitung Ihrer Wahl.

Empfehlung für Textverarbeitungen:

1. TEXbzw. LATEX. Es gibt mittlerweile Lyx, ein komfortables Frontend.

2. OpenOffice

3. Word oder was auch immer

Prognosen und Prognosenfehler im linearen Modell

- Offensichtlich sind zwei Arten von Prognosen im Rahmen des einfachen linearen Modells von Interesse:

  a) Der Prognosefehler $(\hat{y_0} - y_0)$ für eine Stelle x_0 an der man beabsichtigt eine weitere Beobachtung vorzunehmen.
  b) Ein Konfidenzintervall für den Schätzer $\hat{y_0}$ für ein gegebenes x_o.

- Offensichtlich gilt für gegebenes x_0: $E(Y|X = x_0) = \beta_0 + \beta_1 x_0$.

- Als Schätzer aus der Modellfunktion liegt deshalb für gegebenes x_0 nahe:

$$\hat{y}|x_0 := \hat{\beta}_0 + \hat{\beta}_1 x_0.$$

- Bekannt sind die Verteilungen der Koeffizientschätzer $\hat{\beta}_i$. Wie ist die Verteilung von $\hat{y_0}|x_0$?

Der Prognosefehler $(\hat{Y}_0 - Y_0)$

- Die Verteilungen von $\hat{\beta}_0, \hat{\beta}_1$ sind bekannt, insbesondere sind sie standardisiert t-verteilt.

- Wegen der Unkorreliertheit der Fehler ε_i gilt:

$$Var(\hat{Y}_0 - Y_0) = \sigma^2 + Var(\hat{Y}_0)$$

- Damit ergibt sich für ein 1-α Prognoseintervall für eine zukünftige Beobachtung Y_0 an der Stelle x_0 die Form:

$$[\hat{Y}_0 - t_{n-2,1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_1^n x_i^2 - n\bar{x}^2}}; \hat{Y}_0 + t_{n-2,1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_1^n x_i^2 - n\bar{x}^2}}]$$

- Dieses Intervall heißt *Prognoseintervall, prediction interval*.

Konfidenzband für die Regressionsgerade

- Trägt man punktweise die Konfidenzintervalle für $\hat{Y}_0$ zu allen Stellen x_0 zur Regressionsgeraden ein, so bekommt man ein sogenanntes Konfidenzband zur Regressiongerade.

- Mit derselben Herleitung wie beim Prognosefehler ergibt sich das Konfidenzintervall für $\hat{Y}_0$ an der Stelle x_0 zum Niveau 1-α zu

$$[\hat{Y}_0 - t_{n-2,1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{\frac{1}{n}+\frac{(x_0-\bar{x})^2}{\sum_1^n x_i^2 - n\bar{x}^2}}; \hat{Y}_0 + t_{n-2,1-\frac{\alpha}{2}}\hat{\sigma}\sqrt{\frac{1}{n}+\frac{(x_0-\bar{x})^2}{\sum_1^n x_i^2 - n\bar{x}^2}}]$$

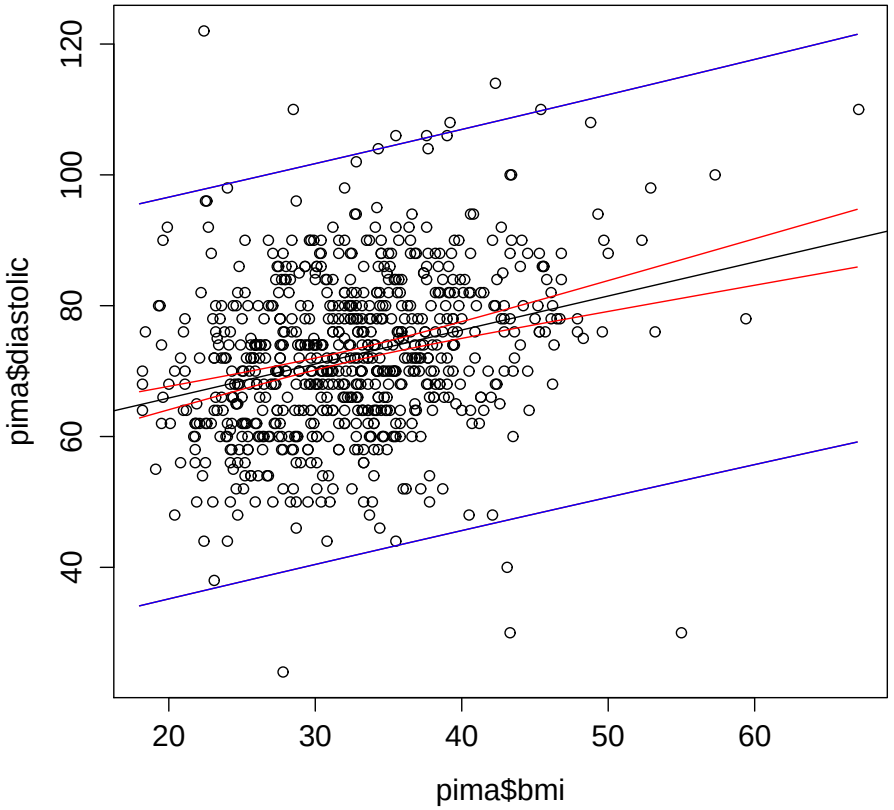
- Wichtigste Beobachtung: Beide Intervalle haben die minimale Weite für $x_0 = \bar{x}$!

Beispiel und Programm in R

- Datensatz pima, Zusammenhang zwischen bmi und diastolic, Einzeichnen des 99% Konfidenzintervalls und des 95% Prognoseintervalls in ein Streudiagramm mit Regressionsgerade.

- ```
  plot(pima$bmi, pima$diastolic)
  abline(result)
  xseq <- seq(round(min(pima$bmi, na.rm=TRUE)),
              round(max(pima$bmi, na.rm=TRUE)))
  cipred <- predict(result, data.frame(bmi=xseq),
                    interval="confidence")
  propred <- predict(result, data.frame(bmi=xseq),
                    interval="prediction", level=0.99)
  lines(xseq, cipred[,2], col="red");
  lines(xseq, cipred[,3], col="red")
  lines(xseq, propred[,2], col="blue")
  lines(xseq, propred[,3], col="blue")
  ```
  ```

Grafik: Prognose- und Konfidenzintervalle



- blau das 99% Prognoseintervall, rot das 95% Konfidenzintervall

Lösung Aufgabe 5: Streuungszerlegung im linearen Modell

Zu zeigen:

$$SQT = SQE + SQR!$$

Beweis:

$$\begin{aligned}
\sum_1^n (y_i - \bar{y})^2 &= \sum_1^n (\hat{y}_i - \bar{y})^2 + \sum_1^n (y_i - \hat{y}_i)^2 \\
\sum_1^n (y_i^2 - 2y_i\bar{y} + \bar{y}^2) &= \sum_1^n (\hat{y}_i^2 - 2\hat{y}_i\bar{y} + \bar{y}^2) + \sum_1^n (\hat{y}_i^2 - 2\hat{y}_i y_i + y_i^2) \\
-2\bar{y}\sum_1^n y_i &= 2\sum_1^n \hat{y}_i^2 - 2\bar{y}\sum_1^n \hat{y}_i - 2\sum_1^n \hat{y}_i y_i
\end{aligned}$$

Da $\sum_1^n y_i = \sum_1^n \hat{y}_i$ bleibt zu zeigen

$$\begin{aligned} 0 &= 2\sum_1^n \hat{y}_i^2 - 2\sum_1^n \hat{y}_i y_i \\ &= 2\sum_1^n \hat{y}_i(\hat{y}_i - y_i) = 2 < \hat{y}, \varepsilon > . \end{aligned}$$

Der letzte Term ist die bekannte Eigenschaft der Regressionsgeraden aus der KQ-Schätzung, dass Schätzvektor und Fehlervektor senkrecht aufeinander stehen.

(Falls nicht bekannt: Beweis z.B. in Draper/Smith , Applied Regression Analysis, Wiley and Sons. Man zeigt die Unkorreliertheit zwischen $\hat{y}$ und ε) $\square$.

Multiple lineare Regression

- Bisher **eine** Einflußgröße X_1 (und der Achsenabschnitt). Dagegen das Modell der multiplen Regression

$$Y = \beta_0 X_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

  mit p Einflußgrößen und dem Achsenabschnitt. Der Achsenabschnitt wird durch eine zusätzliche Variable $X_0 \equiv 1$ ins Modell eingefügt.

- Für die i. Beobachtung gilt also

$$y_i = \beta_0 x_{0,i} + \beta_1 x_{1,i} + \ldots + \beta_p x_{p,i} + \varepsilon_i$$

- Sei nun β der Vektor $(\beta_0, \ldots, \beta_p)$ der Koeffizienten, ε der Fehlervektor, $\mathbf{Y}$ der (n x 1)-Vektor der Beobachungen und $\mathbf{X}$ eine (n x (p+1))-Matrix,

die in der i. Spalte die Werte der i. Einflußgröße für alle n Beobachtungen enthält. Dann gilt

$$Y = X\beta + \varepsilon \text{ mit } E(\varepsilon) = \mathbf{0}.$$

$\mathbf{X}$ heißt Designmatrix des Modells.

- Praktisches Problem: Welche Variablen gehören in die Design-Matrix? (Variablenauswahl, *model selection*)

- Spezialfall einfache Regression: $\mathbf{Y} = (y_1, \ldots, y_n)^T$, $\beta = (\beta_0, \beta_1)^T$ und

$$\mathbf{X} = \begin{pmatrix} 1 & x_{1,1} \\ 1 & x_{1,2} \\ \vdots & \vdots \\ 1 & x_{1,n} \end{pmatrix}, \text{ dann gilt in Matrixschreibweise } Y = X\beta + \varepsilon.$$

Schätzung in der multiplen linearen Regression

- KQ-Schätzung nach demselben Prinzip wie in der einfachen Regression.
- In Matrixschreibweise wird die Minimierung der quadratischen Fehler zu

$$(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) \rightarrow \min_{\beta}.$$

- **Aufgabe 7:** Zeigen Sie, dass dies dem üblichen KQ-Problem entspricht!
- Ableiten und Nullsetzen der Ableitung liefert die *Normalengleichungen*

$$\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) = 0 \Longleftrightarrow \mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{Y}.$$

Schätzung in der multiplen linearen Regression

- Unter den Voraussetzungen , dass

  1. $n \geq p+1$, d.h. mehr Beobachtungen als Parameter zu schätzen und
  2. keine der Variablen $X_j, j = 0, \ldots, p$ mit $X_0 \equiv 1$ darf als Linearkombination der übrigen Variablen $X_k, k \neq j$ darstellbar sein, d.h. es darf für kein $j = 1, \ldots, p$ gelten

$$X_j = \sum_{k \neq j} a_k X_k + b,$$

  (Wäre diese Voraussetzung nicht erfüllt, so würden die Linearkombination und X_j dieselben Anteile von Y erklären.)

- ist die ((p+1) x (p+1))-Matrix $\mathbf{X}^T\mathbf{X}$ invertierbar und es gilt:

$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ ist KQ-Schätzer für den Paramtervektor β.

Eigenschaften der Schätzer in der multiplen Regression

- Der erwartungstreue Schätzer der Fehlervarianz ergibt sich zu

$$\hat{\sigma^2} = \frac{1}{n-p-1}\sum_1^n \hat{\varepsilon}_i^2 = \frac{1}{n-p-1}\sum_1^n (y_i - \hat{y}_i)^2,$$

  wobei $\hat{y} = \mathbf{X}\hat{\beta}$.

- Für die Verteilung der Schätzer $\hat{\beta}_j, j = 0, \ldots, p$ gilt unter der üblichen Normalverteilungsannahme für die Fehler

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}_j} \sim t(n-p-1),$$

  wobei $\hat{\sigma}_j$ die geschätzte Standardabweichung des Schätzer $\hat{\beta}_j$ bezeichnet.

Eigenschaften der Schätzer in der multiplen Regression

- Bezeichen die $v_j, j = 0, \ldots, p$ die Hauptdiagonalelemente von $(\mathbf{X^T X})^{-1}$, so ergeben sich die Schätzer für die Standardabweichung der $\hat{\beta}_j, j = 0, \ldots, p$, zu

$$\hat{\sigma}_j = \hat{\sigma}\sqrt{v_j}.$$

- Direkt folgen daraus die (1-α)-Konfidenzintervalle für β_j als

$$[\hat{\beta}_j - \hat{\sigma}_j t_{1-\frac{\alpha}{2}}(n-p-1);\ \ \hat{\beta}_j + \hat{\sigma}_j t_{1-\frac{\alpha}{2}}(n-p-1)].$$

Hypothesentests in der multiplen Regression

- Für die möglichen Hypothesentests über $\hat{\beta}_j$ gilt mit der Teststatistik $T_j = \frac{\hat{\beta}_j - \beta_{0j}}{\hat{\sigma}_j}$:
  - $H_0 : \beta_j = \beta_{0j}$ vs. $H_1 : \beta_j \neq \beta_{0j}$ ablehnen, wenn $|T_j| > t_{1-\frac{\alpha}{2}}(n-p-1)$,
  - $H_0 : \beta_j \geq \beta_{0j}$ vs. $H_1 : \beta_j < \beta_{0j}$ ablehnen, wenn $T_j < -t_{1-\alpha}(n-p-1)$,
  - $H_0 : \beta_j \leq \beta_{0j}$ vs. $H_1 : \beta_j > \beta_{0j}$ ablehnen, wenn $T_j > t_{1-\alpha}(n-p-1)$.

- Bei einer großen Zahl von Regressoren sollte man die p-Werte mit Vorsicht genießen, da die Problematik des *multiplen Testens* auftritt.

- Von besonderer Bedeutung ist der Fall $H_0 : \beta_j = 0$. Inhaltlich entscheidet dieser Test über die Aufnahme von X_j in die Menge der möglichen Einflußgrößen.

Das multiple Bestimmtheitsmaß R^2 und R_a^2

- Erinnerung: In der Regression (eine Einflußgröße) ist das Bestimmtheitsmaß R^2 definiert als
$$R^2 = 1 - \frac{SSR}{SST}.$$
- Das multiple $R^2 := 1 - \frac{SSR}{SST}$ genau, wie im Fall der einfachen linearen Regression.
- Das adjustiertes R_a^2 berücksichtigt die Anzahl der inkludierten Variablen. Dies ist sinnvoll, das jede zusätzliche Variable das R^2 erhöht. Es gilt:

$$R_a^2 = 1 - \frac{SSR/(n-p)}{SST/(n-1)} = 1 - (\frac{n-1}{n-p})(1-R^2).$$

Ausführliches Beispiel für die multiple Regression

Bitte installieren Sie das Paket DAAG auf ihrem Rechner.

Variablenauswahl (model selection)

- Da die Designmatrix $\mathbf{X}$ in der Regel nicht orthogonal ist, verändern sich Koeffizientenschätzer $\hat{\beta}_j$, wenn die Menge der Schätzer geändert wird.

- Gesucht ist der "beste" Satz von Einflußgrößen, um den Zusammenhang zwischen den X_i und Y zu beschreiben.

- Dieser Umstand ist auch immer bei der Interpretation von Koeffizientenschätzern zu berücksichtigen!

Variablenauswahl: (natürliche) Modellhierachie

- Angenommen ein Modell $Y = \ldots X_i + X_i^2 + \ldots + \epsilon$ sei gegeben. Es sollte vermieden werden, außer es gibt inhaltliche Evidenz, X_i aus der Menge der Regressoren zu entfernen und X_i^2 in dieser Menge zu belassen. Befolgt man diese Regel nicht, so wird die Koeffizientenschätzung abhängig von Skalenverschiebungen.

- Entsprechendes gilt für Modelle mit Wechselwirkungen. Aus einem Modell der Form $Y = \ldots X_i + X_j + X_i X_j + \ldots + \epsilon$ sollten nicht die sogenannten Haupteffekte X_i, X_j entfernt werden, wenn die Wechselwirkung im Modell belassen wird. Ausnahme hier ebenfalls, wenn es inhaltliche Gründe gibt, dies zu tun ("Zweikomponentenkleber")

Variablenauswahl: p-Wert basierte Methoden

- Varianten: backward selection, forward selection, stepwise regression
  - backward selection: Man beginnt mit einem Modell, dass alle Regressoren enthält und entfernt Schritt für Schritt jeweils die Variable, die den größten p-Wert oberhalb eines Schwellenwertes (5%, 10%) hat. Man hat das endgültige Modell gefunden, wenn es keinen solchen Prädiktor mehr gibt.
  - forward selection: Das gegenteilige Vorgehen, d.h. Beginnen mit einem leeren Modell, dann jeweils alle noch nicht enthaltenen Prädiktoren testweise hinzufügen und denjenigen neuen Prädiktor mit dem kleinsten p-Wert, also der höchsten Signifikanz, hinzufügen.
  - Stepwise regression: *freestyle* Kombination aus backward und forward selection, die die nachträglichen Änderungen von inkludierten bzw. exkludierten Einflußgrößen berücksichtigen.

Kriterienbasierte Verfahren der Variablenauswahl

- Üblich: das adjustierte R_a^2 wird anstelle des p-Wertes als Kriterium für die In- oder Exklusion eines Regressors benutzt. Auch hier sind Vorwärts- und Rückwärtsselektionen möglich.

- Es gibt zahlreiche weitere Kriterien, diese werden in dieser Vorlesung nicht behandelt. (AIC (Akaike Information Criterion), BIC (Baysean Information Criterion), Mellows C_p etc.)

Diskussion der Strategien der Variablenauswahl

- p-Wert basierte Methoden tendieren dazu, zu wenig Variablen für eine optimale Prognosefähigkeit aufzunehmen.

- Da jeweils nur eine Variable für den Ein- oder Ausschluss in Betracht gezogen wird, ist es möglich, die optimale Kombination zu übersehen.

- Die Untersuchung aller möglichen Kombinationen von Einflussgrößen ist in der Regel zu aufwendig (exponentiell wachsende Anzahl von Teilmengen).

Modellauswahl durch *backward selection*

- Beispiel für *backward selection* aus Faraway.
- Daten: `state.x77`.
- Ziel: Modell für die Lebenserwartung aus den anderen Variablen herleiten.
- Beginnend mit dem vollen Modell, wird in jedem Schritt der Einflußfaktor entfernt, der den höchsten p-Wert größer als 0.05 hat.
- In der Praxis würde man den letzten Schritt rückgängig machen, da das R_a^2 abnimmt und die gesetzte 5% Grenze von Einflußfaktor `Population` nur sehr knapp überschritten wird.

Die einzelnen Schritte in R

```
data(state)
?state
statedata <- data.frame(state.x77, row.names=state.abb)
tmpmodel <- lm(Life.Exp ~ . , data=statedata )
summary(tmpmodel)
### größter p-Wert: Area

tmpmodel <- update(tmpmodel, . ~ . - Area)
summary(tmpmodel)
### größter p-Wert: Illiteracy

tmpmodel <- update(tmpmodel, . ~ . - Illiteracy)
summary(tmpmodel)
### größter p-Wert: Income

tmpmodel <- update(tmpmodel, . ~ . - Income)
summary(tmpmodel)
```

```
### größter p-Wert Population

tmpmodel <- update(tmpmodel, . ~ . - Population)
summary(tmpmodel)

> summary(tmpmodel)
#### das finale Modell
Call:
lm(formula = Life.Exp ~ Murder + HS.Grad + Frost, data = statedata)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5015 -0.5391  0.1014  0.5921  1.2268

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.036379   0.983262  72.246  < 2e-16 ***
Murder      -0.283065   0.036731  -7.706 8.04e-10 ***
```

```
HS.Grad      0.049949   0.015201   3.286  0.00195 **
Frost       -0.006912   0.002447  -2.824  0.00699 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7427 on 46 degrees of freedom
Multiple R-squared: 0.7127,Adjusted R-squared: 0.6939
F-statistic: 38.03 on 3 and 46 DF,  p-value: 1.634e-12
```

Aufgabe 7: Das Beispiel nachvollziehen und forward selection durchspielen!

Exkurs: Nicht-lineare Modelle

- Das KQ Prinzip trägt auch für nicht-lineare Modellierung.
- Angenommen $f(x, \beta) = \beta_0 + x_1^{\beta_1} + x_2^{\beta_2}$.
- Das KQ Problem

$$\sum_{1}^{n} (y_i - f(x_i, \beta))^2 \rightarrow \min_{\beta}$$

  ist dann zwar nicht analytisch (explizit), jedoch numerisch lösbar und man bekommt ebenfalls einen Schätzvektor $\hat{\beta}$ mit dem man eine Schätzfunktion anpassen kann.

- Normalerweise ist für diesen Fall die Fehlerquadratsumme als Funktion zu definieren und dann auf einen der eingebauten Minimierungsalgorithmen der gewählten Programmiersprache zurückzugreifen.

- In `R` gibt es `optim`, `uniroot`, `nls` und `nlm` für diese numerische Schätzung von Parametern.

- Besonders `nls` (*nonlinear least squares*) ist extrem praktisch:

```
x <- -(1:100)/10 ; y <- 100 + 10 * exp(x / 2) + rnorm(x)/10
nlmod <- nls(y ~  Const + A * exp(B * x), trace=TRUE)
plot(x,y, main = "nls(*), data, true function and fit, n=100")
curve(100 + 10 * exp(x / 2), col=4, add = TRUE)
lines(x, predict(nlmod), col=2)
```

Lösung Aufgabe 7: Matrixschreibweise des KQ-Problems für die multiple lineare Regression

- Ganz allgemein steht das KQ-Prinzip für die Lösung der Minimierungsaufgabe:

$$\sum_{1}^{n}(y_i - \hat{y}_i)^2 \rightarrow \min$$

- Da wir im Moment parametrische Regressionen anschauen, kann man mit einem Parametervektor β auch schreiben

$$\sum_{1}^{n}(y_i - f(x_i, \beta))^2 \rightarrow \min_{\beta}$$

und die gesuchte Lösung dieses Minimierungsproblems heißt $\hat{\beta}$.

- In Matrixschreibweise wird die Minimierung der quadratischen Fehler sehen die einzelnen auftretenden Größen wie folgt aus:

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix};\ X = \begin{pmatrix} x_{0,1} & x_{1,1} & x_{2,1} & \cdots & x_{p,1} \\ \vdots & x_{1,2} & x_{2,2} & \cdots & x_{p,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{0,p} & x_{1,n} & x_{2,n} & \cdots & x_{p,n} \end{pmatrix}; \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

- Dabei ist zu beachten, dass alle $x_{0,i} \equiv 1$.

- Damit ergibt sich

$$X\beta = \begin{pmatrix} \sum_0^p x_{i,1}\beta_i \\ \vdots \\ \sum_0^p x_{i,n}\beta_i \end{pmatrix};\ Y - X\beta = \begin{pmatrix} y_1 - \sum_0^p x_{i,1}\beta_i \\ \vdots \\ y_n - \sum_0^p x_{i,n}\beta_i \end{pmatrix}$$

- Damit wiederum:

$$(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) =$$

$$\left(y_1 - \sum_0^p x_{i,1}\beta_i, \ldots, y_n - \sum x_{i,n}\beta_i\right) \begin{pmatrix} y_1 - \sum_0^p x_{i,1}\beta_i \\ \vdots \\ y_n - \sum x_{i,n}\beta_i \end{pmatrix} =$$

$$\sum_{i=1}^{n}(y_i - \underbrace{\sum_{j=1}^{p} x_{i,j}\beta_j}_{f(x_i,\beta)})^2 \qquad \square$$

Lösung Aufgabe 7: forward selection im Datensatz `state` durchspielen!

- `data(state)` Beispiel für *forward selection*
- Beginnend mit dem **leeren** Modell, wird in jedem Schritt der Einflußfaktor mit dem kleinsten p-Wert hinzugefügt, der noch kleiner als α, z.B. 0.05 ist. Als Kontrollgröße wird neben dem p-Wert R_a^2 benutzt.
- Hier werden wir zwar zum selben Ergebnis kommen, wie bei der *backward selection*, das ist aber nicht zwingend.
- Wenn man etwas programmiert, kann man sich hier eine Menge Handarbeit sparen!

Die einzelnen Schritte der *forward selection* in R

```
data(state)
?state
statedata <- data.frame(state.x77, row.names=state.abb)
options("show.signif.stars" = FALSE)
attach(statedata)
### entweder Schritt für Schritt von Hand
tmpmodel <- lm(Life.Exp ~ 1 , data=statedata )
summary(tmpmodel)
Call:
lm(formula = Life.Exp ~ 1)
Residuals:
    Min      1Q  Median      3Q     Max
-2.9186 -0.7611 -0.2036  1.0139  2.7214
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  70.8786     0.1898   373.4   <2e-16
Residual standard error: 1.342 on 49 degrees of freedom
```

```
### oder ein wenig schlauer
>summary(tmpmodel)$coef
             Estimate Std. Error  t value     Pr(>|t|)
(Intercept)  70.8786  0.1898431 373.3535 2.672210e-86
>dim(summary(tmpmodel)$coef)
[1] 1 4
### Uns interessieren erstmal nur die p-Werte.
### Diese stehen in der $coef Matrix der Summary des Linearen
### Modells, eine Zeile je Parameter und in der 4. Spalte der
### p-Wert. Für jeden neuen Parameter wird eine Zeile
### angehängt.

### Damit funktioniert etwas platzsparender:
> tmpmodel <- lm(Life.Exp ~ 1)
> summary(tmpmodel)$coef[1,4]
[1] 2.672210e-86
> tmpmodel <- lm(Life.Exp ~ Population-1)
```

```
> summary(tmpmodel)$coef[1,4]
[1] 1.891960e-08
> tmpmodel <- lm(Life.Exp ~ Income-1)
> summary(tmpmodel)$coef[1,4]
[1] 5.724488e-45
> tmpmodel <- lm(Life.Exp ~ Illiteracy-1)
> summary(tmpmodel)$coef[1,4]
[1] 9.188552e-18
> tmpmodel <- lm(Life.Exp ~ Murder-1)
> summary(tmpmodel)$coef[1,4]
[1] 2.761127e-18
> tmpmodel <- lm(Life.Exp ~ HS.Grad-1)
> summary(tmpmodel)$coef[1,4]
[1] 1.124834e-43
> tmpmodel <- lm(Life.Exp ~ Frost-1)
> summary(tmpmodel)$coef[1,4]
[1] 3.319432e-19
> tmpmodel <- lm(Life.Exp ~ Area-1)
```

```
> summary(tmpmodel)$coef[1,4]
[1] 4.179951e-07

### kleinster p-Wert bei der Variable X_0, dem Achsenabschnitt
### wird gesetzt als Einflussfaktor

### Welche Größe wird als nächste ins Modell genommen?
> tmpmodel <- lm(Life.Exp ~ Population)
> summary(tmpmodel)$coef[2,4]
[1] 0.6386594
> tmpmodel <- lm(Life.Exp ~ Income)
> summary(tmpmodel)$coef[2,4]
[1] 0.01561728
> tmpmodel <- lm(Life.Exp ~ Illiteracy)
> summary(tmpmodel)$coef[2,4]
[1] 6.96925e-06
> tmpmodel <- lm(Life.Exp ~ Murder)
> summary(tmpmodel)$coef[2,4]
```

```
[1] 2.260070e-11
> tmpmodel <- lm(Life.Exp ~ HS.Grad)
> summary(tmpmodel)$coef[2,4]
[1] 9.196096e-06
> tmpmodel <- lm(Life.Exp ~ Frost)
> summary(tmpmodel)$coef[2,4]
[1] 0.0659874
> tmpmodel <- lm(Life.Exp ~ Area)
> summary(tmpmodel)$coef[2,4]
[1] 0.4581464

### Kleinster p-Wert für die Variable Murder
> tmpmodel <- lm(Life.Exp ~ Murder)
> summary(tmpmodel)


Call:
lm(formula = Life.Exp ~ Murder)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.8169 -0.4814  0.0959  0.3977  2.3869
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 72.97356    0.26997  270.30  < 2e-16
Murder      -0.28395    0.03279   -8.66 2.26e-11
Residual standard error: 0.8473 on 48 degrees of freedom
Multiple R-squared: 0.6097,Adjusted R-squared: 0.6016
F-statistic: 74.99 on 1 and 48 DF,  p-value: 2.26e-11

### Welche Variable wird als dritte aufgenommen
> tmpmodel <- lm(Life.Exp ~ Murder + Population)
> summary(tmpmodel)$coef[3,4]
[1] 0.01636940
> tmpmodel <- lm(Life.Exp ~  Murder + Income)
> summary(tmpmodel)$coef[3,4]
[1] 0.06663619
```

```
> tmpmodel <- lm(Life.Exp ~  Murder + Illiteracy)
> summary(tmpmodel)$coef[3,4]
[1] 0.5429104
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad)
> summary(tmpmodel)$coef[3,4]
[1] 0.009088366
> tmpmodel <- lm(Life.Exp ~  Murder + Frost)
> summary(tmpmodel)$coef[3,4]
[1] 0.03520523
> tmpmodel <- lm(Life.Exp ~  Murder + Area)
> summary(tmpmodel)$coef[3,4]
[1] 0.4243751
}
### HS.Grad hat diesmal den kleinsten p-Wert

> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad)
> summary(tmpmodel)
Call:
```

```
lm(formula = Life.Exp ~ Murder + HS.Grad)
Residuals:
     Min       1Q   Median       3Q      Max
-1.66758 -0.41801  0.05602  0.55913  2.05625
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 70.29708    1.01567  69.213  < 2e-16
Murder      -0.23709    0.03529  -6.719 2.18e-08
HS.Grad      0.04389    0.01613   2.721  0.00909
Residual standard error: 0.7959 on 47 degrees of freedom
Multiple R-squared: 0.6628,Adjusted R-squared: 0.6485
F-statistic:  46.2 on 2 and 47 DF,  p-value: 8.016e-12

### und weiter ...

tmpmodel <- lm(Life.Exp ~ Murder + HS.Grad + Population)
> summary(tmpmodel)$coef[4,4]
[1] 0.01994926
```

```
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad + Income)
> summary(tmpmodel)$coef[4,4]
[1] 0.6924184
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad + Illiteracy)
> summary(tmpmodel)$coef[4,4]
[1] 0.4094209
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad + Frost)
> summary(tmpmodel)$coef[4,4]
[1] 0.006987727
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad + Area)
> summary(tmpmodel)$coef[4,4]
[1] 0.5138632
### Frost hat den niedrigsten p-Wert

> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad + Frost)
> summary(tmpmodel)
Call:
lm(formula = Life.Exp ~ Murder + HS.Grad + Frost)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.5015 -0.5391  0.1014  0.5921  1.2268
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.036379   0.983262  72.246  < 2e-16
Murder      -0.283065   0.036731  -7.706 8.04e-10
HS.Grad      0.049949   0.015201   3.286  0.00195
Frost       -0.006912   0.002447  -2.824  0.00699
Residual standard error: 0.7427 on 46 degrees of freedom
Multiple R-squared: 0.7127,Adjusted R-squared: 0.6939
F-statistic: 38.03 on 3 and 46 DF,  p-value: 1.634e-12

### und nochmal
tmpmodel <- lm(Life.Exp ~ Murder + HS.Grad + Frost + Population)
> summary(tmpmodel)$coef[5,4]
[1] 0.05200514
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Income)
```

```
> summary(tmpmodel)$coef[5,4]
[1] 0.571031
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Illiteracy)
> summary(tmpmodel)$coef[5,4]
[1] 0.5823608
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Area)
> summary(tmpmodel)$coef[5,4]
[1] 0.8317269

## kein Faktor mehr mit p-Wert unter 5%
## Population kann man versuchweise noch aufnehmen, da das
## R^2_a noch wächst

## letzter Durchgang:

tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Population + Income)
> summary(tmpmodel)$coef[6,4]
[1] 0.9153104
```

```
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Population + Illiteracy
> summary(tmpmodel)$coef[6,4]
[1] 0.9318143
> tmpmodel <- lm(Life.Exp ~  Murder + HS.Grad +  Frost + Population +  Area)
> summary(tmpmodel)$coef[6,4]
[1] 0.969369

### Alle weitern Größen insignifkant
### Damit ergibt sich als gewähltes Modell:

> tmpmodel <- lm(Life.Exp ~ Murder + HS.Grad + Frost + Population)
> summary(tmpmodel)
Call:
lm(formula = Life.Exp ~ Murder + HS.Grad + Frost + Population)
Residuals:
     Min       1Q   Median       3Q      Max
-1.47095 -0.53464 -0.03701  0.57621  1.50683
Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.103e+01  9.529e-01  74.542  < 2e-16
Murder      -3.001e-01  3.661e-02  -8.199 1.77e-10
HS.Grad      4.658e-02  1.483e-02   3.142  0.00297
Frost       -5.943e-03  2.421e-03  -2.455  0.01802
Population   5.014e-05  2.512e-05   1.996  0.05201
Residual standard error: 0.7197 on 45 degrees of freedom
Multiple R-squared: 0.736,Adjusted R-squared: 0.7126
F-statistic: 31.37 on 4 and 45 DF,  p-value: 1.696e-12
```

Ausblick: Variablenselektion

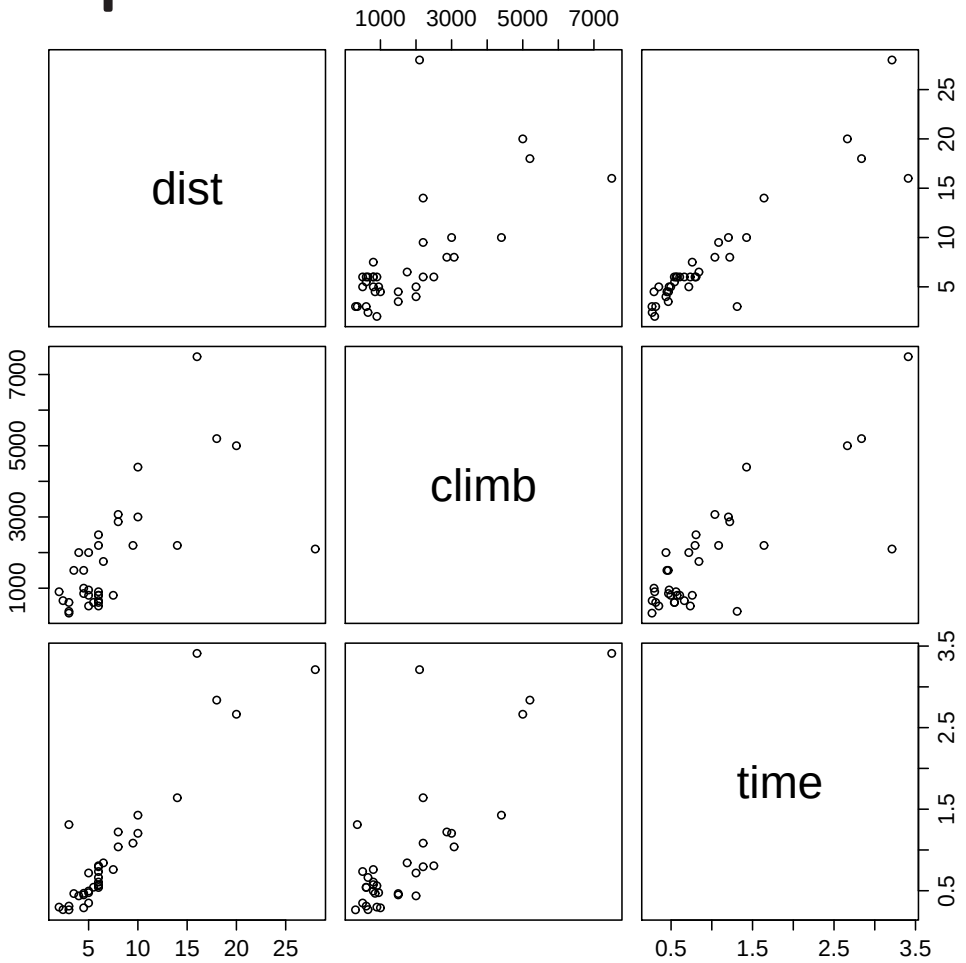
- Die hier vorgestellten Verfahren sind "einfach".
- Wie bereits bemerkt kann das schrittweise (*one-at-a-time*) Vorgehen dazu führen, dass die beste Teilmenge von Einflussgrößen nicht gefunden wird.
- Das Paket `leaps` stellt einige umfassendere Funktionen für ein allgemeineres Vorgehen zur Verfügung.
- Dazu gehört unter anderem die *erschöpfende Suche*, bei der **alle** möglichen Teilmengen von Größen untersucht werden.
- Zwar findet man so die beste Teilmenge von Größen, allerdings müssen 2^p mögliche Teilmengen von Einflußgrößen $X_1, \ldots, X_p$ untersucht werden!
- Auch heute noch nicht möglich für echte Probleme mit z.B. $p = 40$!

Abschließendes Bespiel zur Regression

- Datensatz `hills` aus dem Paket `DAAG`
- Rekordzeiten für diverse schottische Bergläufe, Stand 1984
- 35 Strecken im Datensatz, jeweils Streckenlänge in Meilen (`dist`), Höhenmeter in Fuß (`climb`) und Rekordzeit in Stunden (`time`)
- Vorbereiten der Analyse:

```
> library(DAAG) ; data(hills)
> help(hills)
> hi.a <- hills
> pairs(hi.a)
```

Scatterplotmatrix zum `hills` Datensatz

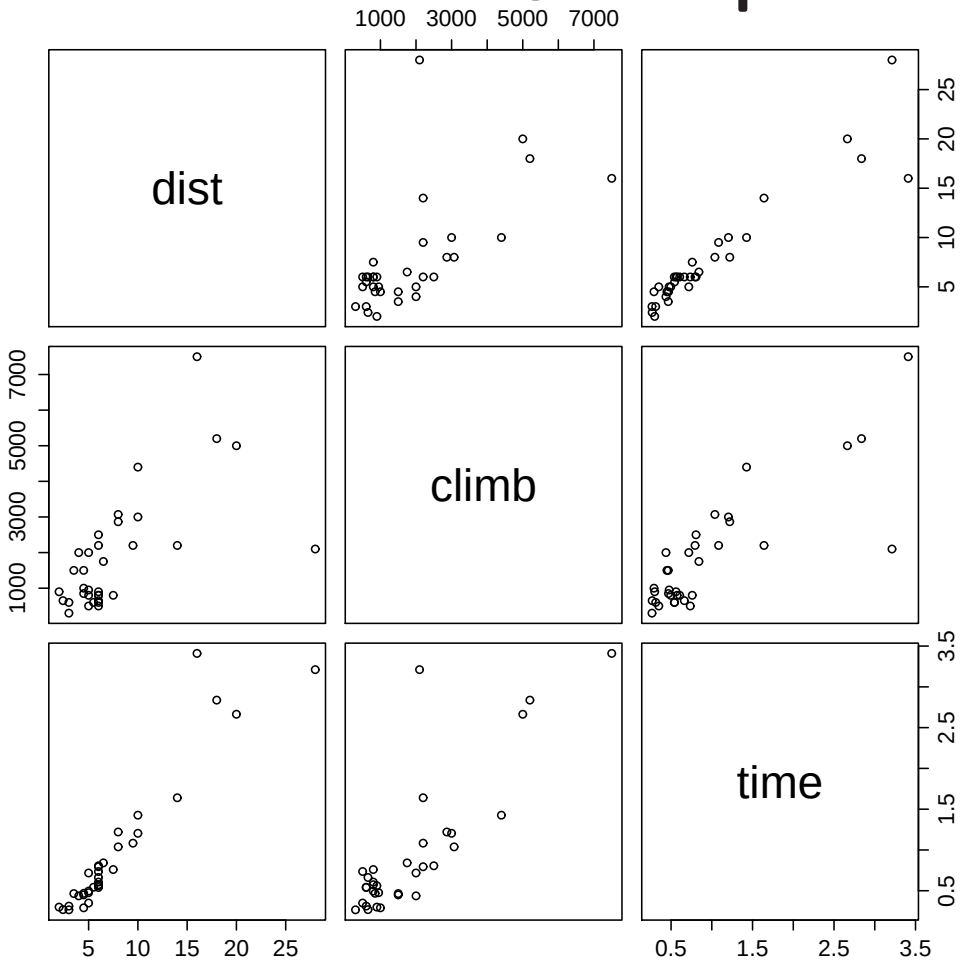


Analyse des Scatterplots

- Sind auffällige Punkte zu erkennen?
- Ja, die Beobachtung mit fast 1.5 h für 3 Meilen.
- Bebachtung finden und aus dem Analysedatensatz entfernen. (Nr. 18)

```
> hi.a
### in den Daten den Punkt suchen
> hi.a <- hi.a [-18,]
### und entfernen
> pairs(hi.a)
### Kontrolle!
```

Kontrolle des Scatterplots



Erster Modellansatz: Lineares Modell

- Inhaltliche Überlegung: Sowohl Länge als auch Höhenmeter sollten Einfluß auf die Gesamtzeit haben!

- Erste Idee: einfaches lineares Modell:

$$\text{time} = \beta_0 + \beta_1 * \text{dist} + \beta_2 * \text{climb}$$

- in R:

```
hi.a.lm <- lm(time~dist + climb , data=hi.a)
summary(hi.a.lm)
```

```
Call:
lm(formula = time ~ dist + climb, data = hi.a)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27838 -0.08837  0.01962  0.06253  0.45695

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.247e-01  4.420e-02  -5.083 1.69e-05
dist         1.060e-01  6.026e-03  17.592  < 2e-16
climb        1.976e-04  2.062e-05   9.584 8.76e-11

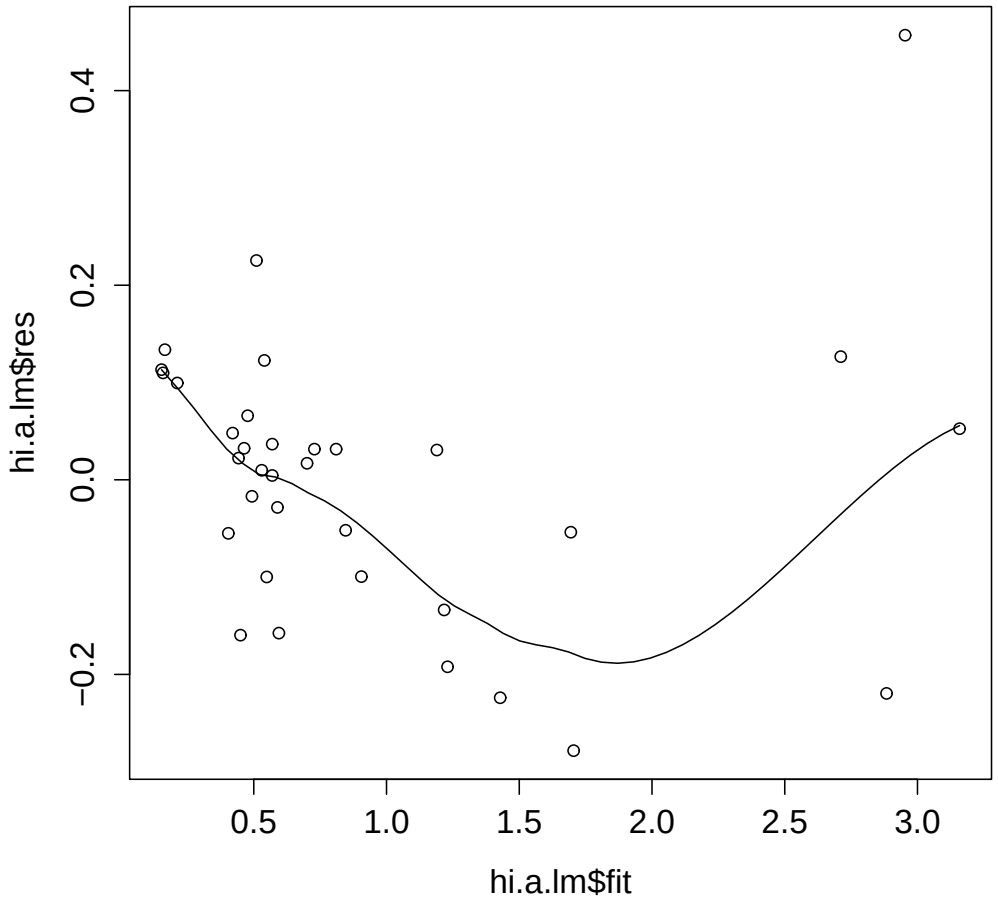
Residual standard error: 0.147 on 31 degrees of freedom
Multiple R-squared: 0.9715,Adjusted R-squared: 0.9697
F-statistic: 529.1 on 2 and 31 DF,  p-value: < 2.2e-16
```

Interpretation des multiplen linearen Modells

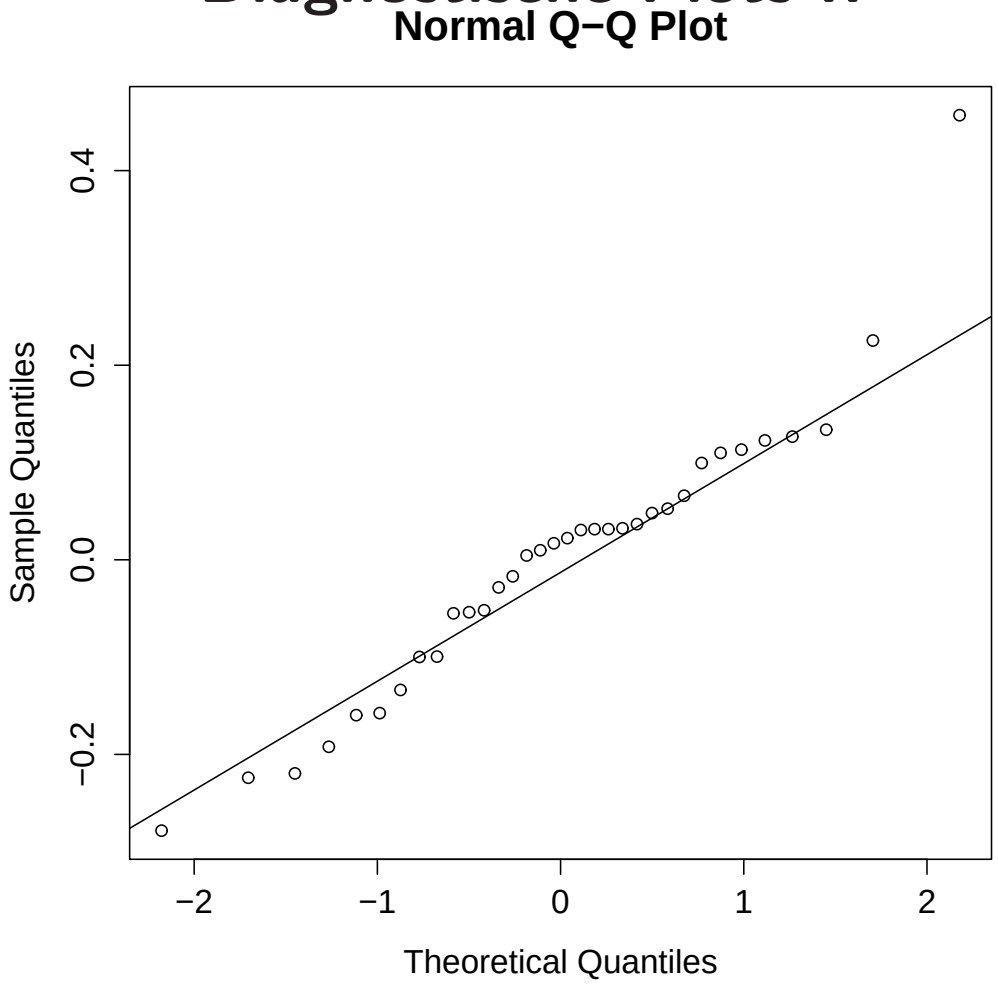
- Sehr hohes R_a^2. Dies spricht für das Modell.
- Allerdings: Die Grafik der angepassten Werte gegen die Modellfehler zeigt klar eine Struktur, genau wie der QQ-Plot der Fehler.
- in R:

```
scatter.smooth(hi.a.lm$fit , hi.a.lm$res)
qqnorm(hi.a.lm$res); qqline(hi.a.lm$res)
```

Diagnostische Plots I



Diagnostische Plots II



Modellverfeinerung

- Muss evtl eine Wechselwirkung zwischen `dist` und `climb` berücksichtigt werden?

- Neues Modell

$$\text{time} = \beta_0 + \beta_1 * \text{dist} + \beta_2 * \text{climb} + \beta_3 * \text{dist:climb}$$

- in `R`:

```
> hi.b.lm <- lm(time ~ dist + climb + dist:climb, data=hi.a)
> summary(hi.b.lm)
```

```
Call:
lm(formula = time ~ +dist + climb + dist:climb, data = hi.a)
Residuals:
     Min       1Q   Median       3Q      Max
-0.38684 -0.05109  0.01201  0.03721  0.31571

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.130e-02  6.946e-02  -0.163    0.872
dist         8.257e-02  8.207e-03  10.061 3.97e-11
climb        6.132e-05  4.125e-05   1.487    0.148
dist:climb   1.104e-05  3.028e-06   3.646    0.001

Residual standard error: 0.1244 on 30 degrees of freedom
Multiple R-squared: 0.9803,Adjusted R-squared: 0.9783
F-statistic:   497 on 3 and 30 DF,  p-value: < 2.2e-16
```

Modellverfeinerung II

- R^2 ist gewachsen und der Intercept ist nicht mehr signifikant. (Sinnvolle Modellannahme!)

- Also: Achsenabschnitt aus dem Modell entfernen!

- In R:

```
>hi.b.lm <- lm(time ~ -1 + dist + climb + dist:climb, data=hi.a)
>summary(hi.b.lm)
Call:
lm(formula = time ~ -1 + dist + climb + dist:climb, data = hi.a)

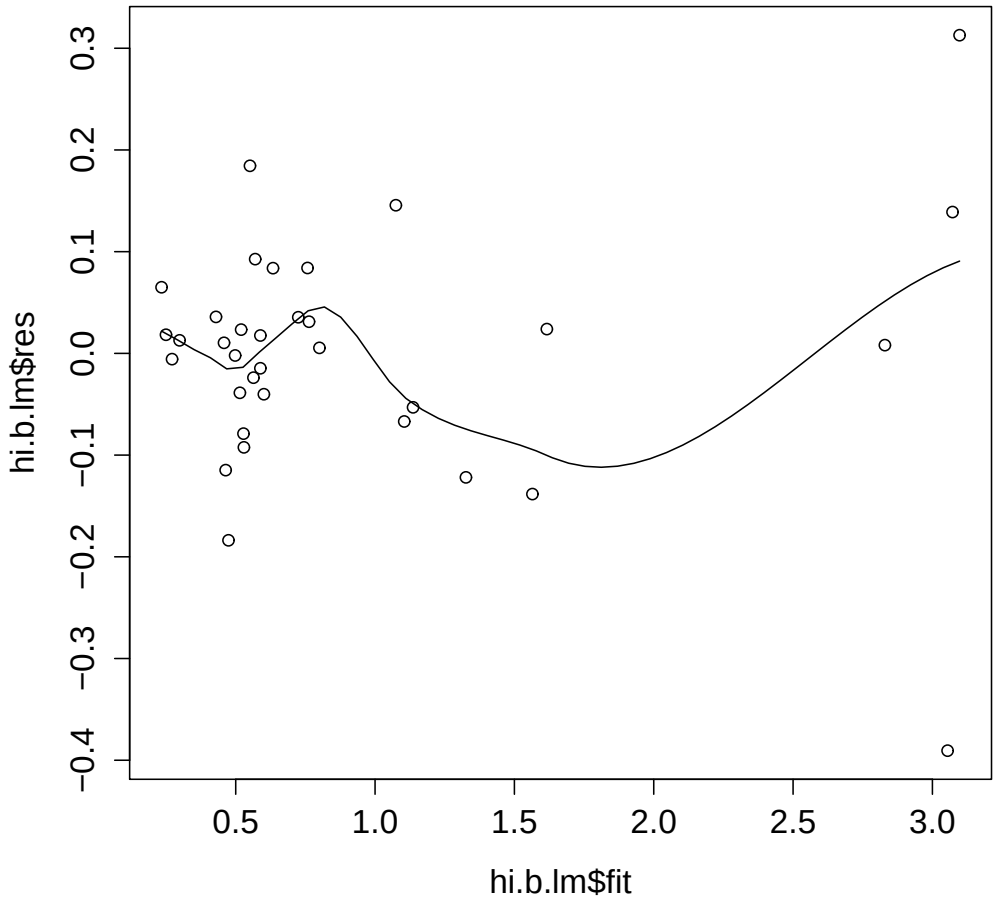
Residuals:
     Min       1Q   Median       3Q      Max
```

```
-0.39059 -0.04982  0.00924  0.03577  0.31281
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
dist        8.147e-02  4.592e-03  17.742  < 2e-16
climb       5.590e-05  2.394e-05   2.336   0.0262
dist:climb 1.146e-05  1.605e-06   7.137 5.07e-08

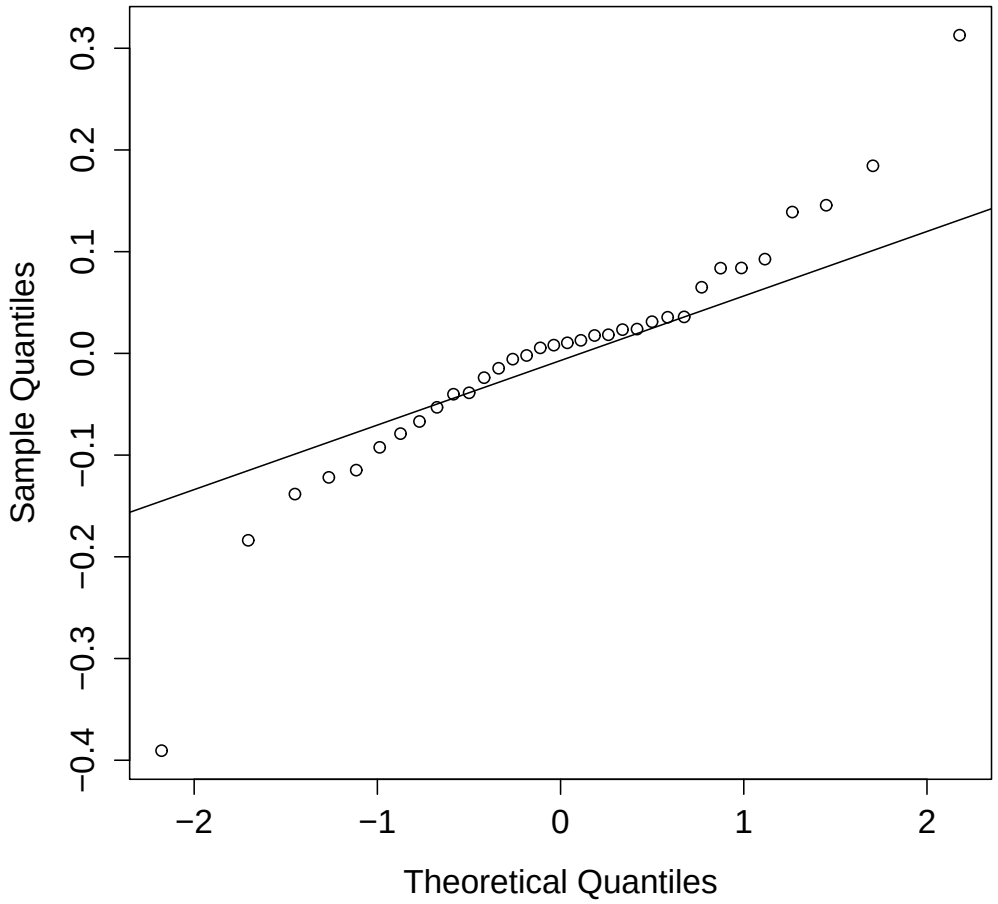
Residual standard error: 0.1224 on 31 degrees of freedom
Multiple R-squared: 0.9915,Adjusted R-squared: 0.9907
F-statistic:  1202 on 3 and 31 DF,  p-value: < 2.2e-16
```

- `summary(hi.b.lm)` zeigt R_a^2 von 0.991!

Diagnostische Plots III



Diagnostische Plots IV



Zwischenfazit

- Wer bis hierher kommt, kann schon viel mehr als die meisten. Jetzt noch die Kür!
- Der QQ-Plot der Residuen ist noch nicht optimal.
- Welche Beobachtungen sind die Abweichler im QQ-Plot?

```
> which.max(hi.b.lm$res); which.min(hi.b.lm$res); hi.a
```

- Es sind lange, steile Rennen!
- Evtl ist der Zusammenhang nicht rein linear?

Modellverfeinerung (Kür!)

- Annahme: Die Länge geht im Wesentliche linear in die Zeit ein, die Steigung hat aber überproportionalen Einfluß auf die Endzeit.

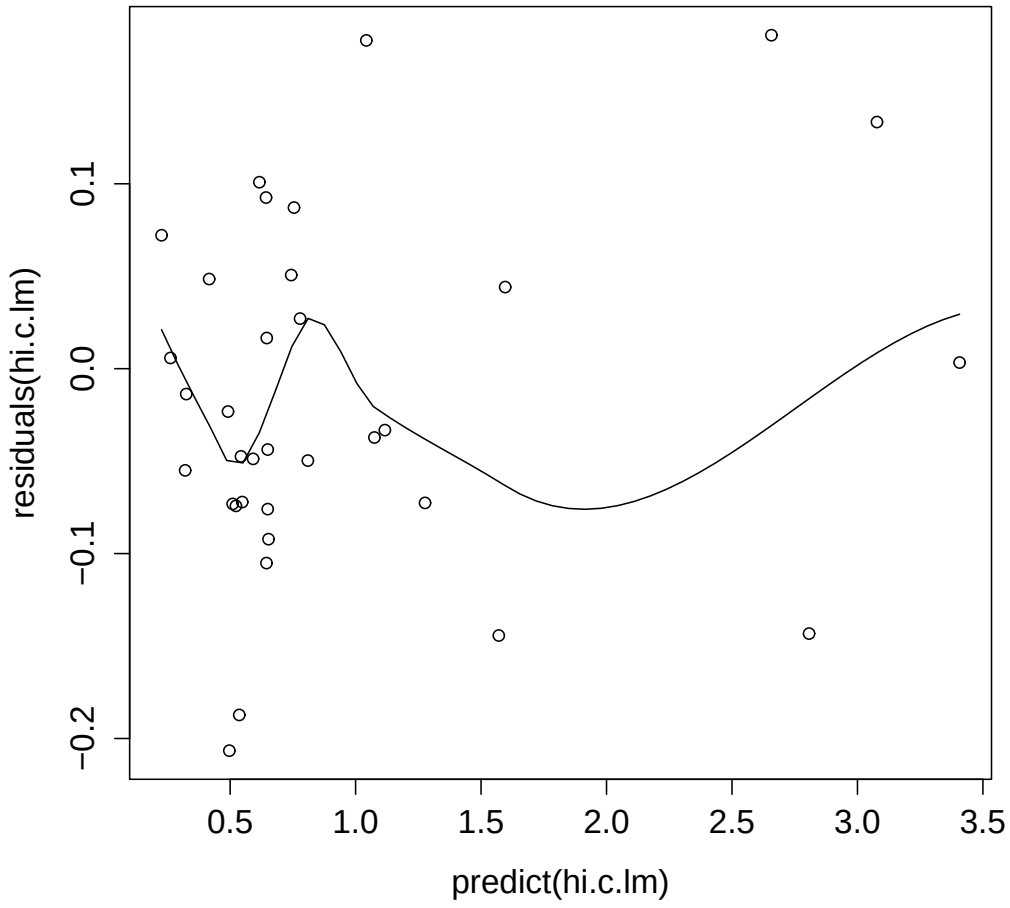
- Das Modell:

$$\text{time} = \beta \cdot \text{dist} + \gamma \cdot \text{climb}^{\delta}$$

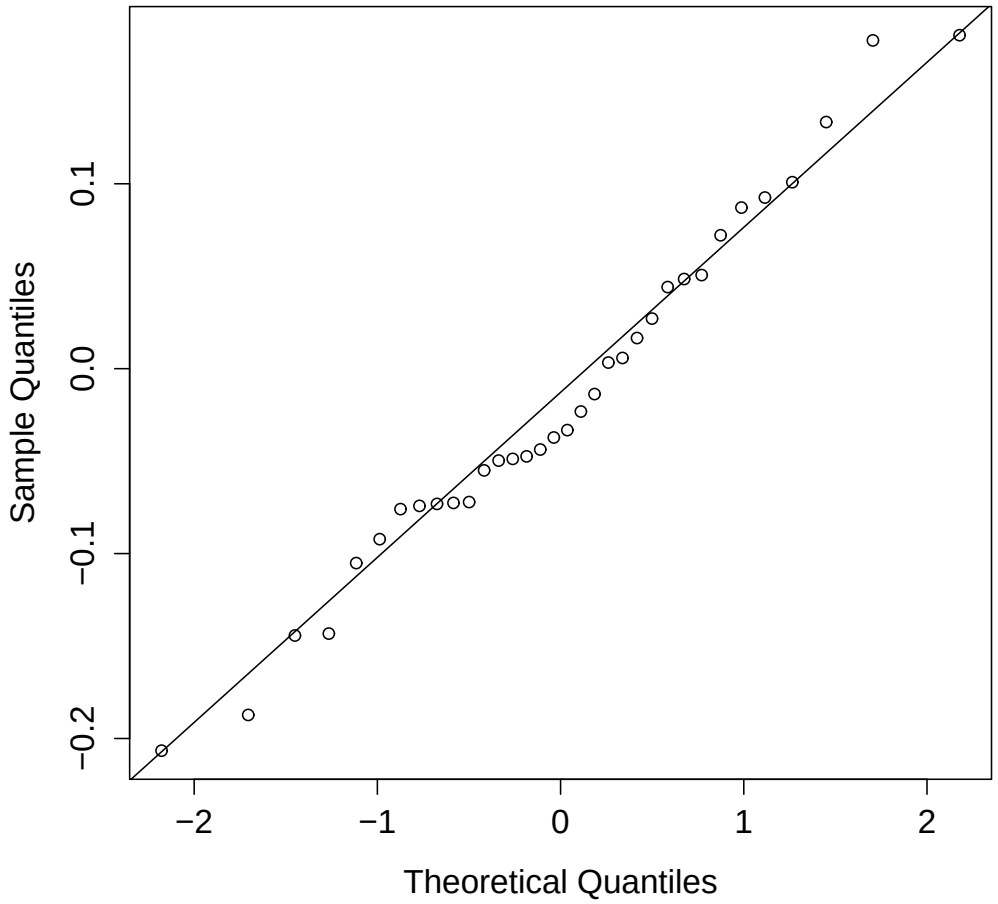
- In R:

```
hi.c <- hi.a ; hi.c$climb <- hi.c$climb/5280
### Damit X'X gut konditioniert ist!
hi.c.lm <- nls(time ~ (beta*dist) +
               gamma*(climb^delta) ,
               start= c(beta=1, gamma=1, delta=1), data=hi.c)
1 - var(residuals(hi.c.lm))/  var(hi.a$time) ### r.squared
[1] 0.98
```

Diagnostische Plots V



Diagnostische Plots VI



Was war mit Beobachtung 18?

- Das Modell scheint nunmehr den Annahmen zu entsprechen!
- Mit der schließlichen Modellanpassung ergibt sich, dass Strecke 18

```
> predict(hi.c.lm, data.frame(dist=hills[18,"dist"],
          climb=hills[18,"climb"]/5280))
[1] 0.3213
```

  Stunden gedauert haben sollte. Am Wahrscheinlichsten ist also eine Fehleingabe, bei der statt 0.3 Stunden 1.3 Stunden eingegeben wurden.

- Als Pedant könnte man die ganze Analyse an dieser Stelle mit den korrigierten Daten wiederholen.

Exkurs: Datenein- und Ausgabe mit R

- In der Regel ist Datenaustausch mit anderen Programmen im Rahmen des Datenanalyseprozesses notwendig.

- `R` hat viele Möglichkeiten der Datenein- und -ausgabe implementiert. Manche davon allerdings in externen Paketen.

- Über das Paket `foreign` können beispielsweise SPSS-, SAS- oder auch Stata-Files gelesen werden.

- Excel-Files sind sicher die häufigste Datenquelle. Man kann direkt mit ihnen arbeiten, aber es gibt immer Schwierigkeiten.

- Es gibt das Paket `xlsReadWrite`. Dieses ist aber nur unter Windows verfügbar und kein Opensource!

- Das Excel-Datenformat ist nicht klar definiert!

- Wenn es unbedingt sein muss, kann man auf ein Excel-Format vor Excel 2007 gehen, um die Interoperabilität mit anderen Programmen zu verbessern.

- Zugriff über RODBC ist eine sichere Variante. Dabei wird jedes Arbeitsblatt als Tabelle einer Datenbank betrachtet.

- Dasselbe Paket bietet zusammen mit DBI einen sehr komfortablen Zugang zu fast allen aktuellen Datenbanksystemen. Es wird ein Interface zur Datenbanksprache SQL (*structured query language*) implementiert.

- Entweder Datenbanken oder CSV (*comma separated values*, Textfiles(!)).

- Für unstrukturierte Dateneingaben gibt es `scan()` oder `readline()`.

Einlesen von CSV Dateien

- Ganz allgemein lassen sich Dateien, die eine Datenmatrix enthalten, mit dem Kommando `read.table()` einlesen. Das Ergebnis ist jeweils ein Dataframe.

- Es verbirgt sich eine ganze Familie von Funktionen hinter `read.table()`.

- ```
  read.table(file, header = FALSE, sep = "", quote = "\"'",
             dec = ".", row.names, col.names, as.is = FALSE,
             na.strings = "NA", colClasses = NA, nrows = -1,
      skip = 0, check.names = TRUE, fill = !blank.lines.skip,
             strip.white = FALSE, blank.lines.skip = TRUE,
             comment.char = "\#", allowEscapes = FALSE)
  ```
  ```

Komfortfunktionen für CSV Dateien

- `read.csv()` bzw. `read.csv2()` haben die Defaultparameter so voreingestellt, dass z.B. mit `read.csv2()` Dateien aus dem deutschsprachigen Raum korrekt eingelesen werden.

- Es handelt sich lediglich um Aliasse von `read.table()`!

- **Aufgabe 9:** Lesen Sie die Datei `sturmfluten.csv` von der Homepage der Veranstaltung mit `read.table()` ein!

Datenausgabe in CSV Dateien

- Wie bei der Eingabe beherrscht `R` auch bei der Ausgabe viele externe Dateiformate.

- Aus Gründen der Portabilität bevorzuge ich jedoch auch für die Ausgabe CSV Dateien! Alle Tabellenkalkulationen können diese lesen!

- Wenig überraschend lautet das Kommando zum sichern eines Dataframe in eine Datei `write.table()` (oder `write.csv()` bzw. `write.csv2()`).

- ```
  write.table(x, file = "", append = FALSE, quote = TRUE,
              sep = " ", eol = "\n", na = "NA", dec = ".",
              row.names = TRUE, col.names = TRUE,
              qmethod = c("escape", "double"))
  ```
  ```

Einfaktorielle Varianzanalyse (ANOVA)

- ANOVA: *Analysis of Variance*

- In der Regression wurde der Zusammenhang zwischen einer oder mehreren *metrischen* Einflußgrößen und einer ebenfalls *metrischen* Zielgröße modelliert.

- In der *einfaktoriellen ANOVA* wird untersuchte, ob es einen Einfluß der Ausprägung einer *kategoriellen* Einflußgröße auf eine *metrische* Zielgröße gibt.

- Kategoriell bedeutet in diesem Zusammenhang die Zugehörigkeit zu einer Gruppe innerhalb einer Einflußgröße. Beispielsweise die Einflußgröße "Geschlecht" und die Gruppen Männer und Frauen.

Einfaktorielle Varianzanalyse (ANOVA)

- Die Einflußgrößen in der Varianzanalyse heißen auch *Einflußfaktoren* oder kurz *Faktoren*. Die Ausprägungen der Faktoren heißen *Faktorstufen*. (*factor* und *factor level*)

- Erinnerung: Eine solche Variable kam bereits im `pima`-Datensatz vor, nämlich dort die Variable, ob bereits Symptome der Diabetes erkennbar sind.

- Da keine stetige x-Achse vorliegt, muss man sich auf den Einfluß der Gruppenzugehörigkeit auf den Stichprobenmittelwert beschränken.

- Ein Beispiel ist bereits bekannt aus Statistik II: Vergleich zweier Mittelwerte auf Gleichheit.

Einführendes Beispiel zur ANOVA

- Ein typisches Problem in der chemischen und pharmakologischen Industrie ist die Sicherstellung der Vergleichbarkeit von Analyseergebnissen von Untersuchungslabors. (ISO Zertifizierung!)

- Angenommen Sie haben zu beurteilen, ob drei Labore im Mittel identische Analyseergebnisse liefern.

- Jede präparierte Probe enthalte genau 4 mg eines Wirkstoffes.

- Die Labore erhalten die Aufgabe, den Wirkstoffgehalt zu messen.

Einführendes Beispiel zur ANOVA

- Es ergeben sich folgende Messreihen

Faktorstufe	Messungen					
Labor 1	4.13	4.07	4.04	4.07	4.05	4.04
Labor 2	3.86	3.85	4.08	4.11	3.83	4.01
Labor 3	4.00	4.02	4.01	4.01	4.04	3.99

- Eine solche Messreihe liefert Informationen über
  - die Schwankungen der Messungen innerhalb eines Labors und
  - die Konsistenz der Analysen der Labore.

- Offensichtlich sind in den Niveaus Unterschiede, aber sind diese statistisch signifikant?

Modell der Varianzanalyse I

- Um diese Frage zu beantworten benötigen wir ein statistisches Modell der zugrunde liegenden Datengenerierung.

- Schematisch lassen sich die Daten, die einer Varianzanalyse zugrunde liegen wie folgt darstellen:

	Zielgröße Y			Stichprobenumfang
Faktorstufe 1 (X_1)	y_{11}	$\ldots$	y_{1n_1}	n_1
Faktorstufe 2 (X_2)	y_{21}	$\ldots$	y_{2n_2}	n_2
$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\vdots$
Faktorstufe k (X_k)	y_{k1}	$\ldots$	y_{kn_k}	n_k

- Notation: y_{ij} ist die Beobachtung Nummer j bei der Faktorstufe i, $N = \sum_1^k n_i$ bezeichnet den Gesamtstichprobenumfang.

Modell der Varianzanalyse II

- Grundannahme: Die Varianz der Daten ist auf jeder Faktorstufe gleich.
- Verbal besagt das Modell der Varianzanalyse , dass sich der Wert der Zielgröße jeweils aus einem Mittelwert abhängig von der Faktorstufe und einem Fehler zusammensetzt.
- In Formeln

$$Y_{ij} = \mu_i + \varepsilon_{ij}, \;\; i = 1, \ldots, k, \;\; j = 1, \ldots, n_i \text{ und } \varepsilon_{ij} \sim N(0, \sigma^2)$$

- Besonders hervorzuheben ist hierbei, dass die Varianz der Störgröße ε_{ij} für alle Beobachtungen gleich sind.

Modell der Varianzanalyse III

- Eine sinnvolle Frage, die beantwortet werden soll, ist z.B. “Hat die Faktorausprägung einen Einfluß auf die Zielgröße¿‘

- Als statistische Test-Hypothese, die überprüft werden soll, wird das mit unserer Notation übersetzt in

$$H_0: \ \mu_1 = \mu_2 = \ldots = \mu_k \text{ vs. } H_1: \ \mu_i \neq \mu_j \ \text{ für ein Indexpaar } (i, j).$$

- Die Nullhypothese besagt, dass es keine Mittelwertunterschiede zwischen den Faktorstufen gibt, die Alternative, dass sich zumindest zwei Mittelwerte unterscheiden.

Modell der Varianzanalyse IV

- Es existiert ein äquivalentes Modell, bei dem jedoch eine andere Modellidee formuliert wird.

- Jeder Faktorstufe wird ein Effekt als Abweichung von einem allgemeinen Mittel zugeordnet. In unserer Notation

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad i = 1, \ldots, k \quad j = 1, \ldots, n_i \quad \varepsilon_{ij} \sim N(0, \sigma^2)$$

- Diese Darstellung heißt *Effektdarstellung* des Varianzanalysemodells.

- Hierbei heißt α_i mit $\mu = \frac{1}{N} \sum_i^k n_i \mu_i$ (dem allgemeinen Mittel) und $\alpha_i = \mu_i - \mu$ der Effekt der Faktorstufe i.

Modell der Varianzanalyse V

- Die Äquvalenz der beiden Modellformulierungen sieht man leicht:

$$Y_{ij} = \mu_i + \varepsilon_{ij} = \mu - (\mu_i - \mu) + \varepsilon_{ij} = \mu - \alpha_i + \varepsilon_{ij}$$

- Ebenso sieht man leicht $\sum_1^k n_i \alpha_i = 0$. Inhaltlich bedeutet dies, dass Abweichungen vom allgemeinen Mittel sich aufheben sollen. Ohne diese Bedingung wären die Parameter nicht eindeutig schätzbar.

- Die (äquivalente) Hypothese lautet dann

$$H_0 : \alpha_1 = \alpha_2 = \ldots = \alpha_k = 0 \; vs. \; H_1 : \text{mindestens zwei } \alpha_i \neq 0$$

Schätzung im Modell der ANOVA

- Zu schätzen sind $\mu, \alpha_i,\ i = 1, \ldots, k$ und die Fehlervarianz σ^2 innerhalb der Gruppen und für die Gesamtstichprobe.

- Ein geeigneter Schätzer $\hat{\mu}$ für das allgemeine Mittel μ ist

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{n_i} y_{ij} =: \bar{Y}_{..} \quad .$$

- Ein geeigneter Schätzer $\hat{\alpha}_i$ für den Effekt der Faktorstufe i auf das allgemeine Mittel μ ist

$$\hat{\alpha}_i = \bar{Y}_{i.} - \bar{Y}_{..} = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij} - \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{n_i} y_{ij} \quad .$$

Idee und Konstruktion der Testgröße I

- Die nahe liegende Idee zur Überprüfung der Hypothese H_0 ist die Ausnutzung der Wert $\bar{Y}_{\cdot} - \bar{Y}_{\cdot\cdot}$, also der Abweichungen der Gruppenmittelwerte vom allgemeinen Mittel.

- Nach dem KQ Prinzip und um gleichzeitig unterschiedliche Stichprobenumfänge in den Gruppen auszugleichen ist eine mögliche Teststatistik, analog zur Regression

$$\frac{1}{k-1}\sum_{i=1}^{k} n_i(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2 \quad .$$

- Nun ist noch die Standardisierung mit einem Schätzer für die Standardabweichung nötig.

Idee und Konstruktion der Testgröße II

- Für jede Gruppe i gilt, dass

$$\hat{\sigma}_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (y_{ij} - \bar{Y}_{i.})^2$$

  ein erwartungstreuer Schätzer für die Fehlervarianz σ^2 ist.

- Ebenso ist die Kombination dieser Gruppenschätzer zu einem Gesamtschätzer

$$\hat{\sigma}^2 = \frac{1}{N - k} \sum_{i=1}^{k} (n_i - 1) \hat{\sigma}_i^2$$

  ein erwartungstreuer Schätzer für σ^2.

Testgröße der Varianzanalyse

- In der Situation der Effektdarstellung der Varianzanalyse ist die Testgröße

$$F = \frac{\frac{1}{k-1}\sum_{i=1}^{k} n_i(\bar{Y}_{i.} - \bar{Y}_{..})^2}{\frac{1}{N-k}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij} - \bar{Y}_{i.})^2}$$

  unter H_0 F-verteilt mit k-1 und N-k Freiheitsgraden.

- Der kritische Wert zum Niveau α dieses Testes ist $F^{-1}_{k-1;N-k;1-\alpha}$.

Aufgaben

- Zeigen Sie Erwartungstreue der angeführten Schätzer für μ, α_i und σ^2!

Woher kommt der Name Varianzanalyse?

- Man kann zeigen, dass die Zerlegung gilt

Gesamtvarianz = Varianz zwischen den Gruppen + Varianz innerhalb der Gruppen

- Die F-Statistik setzt nun im Wesentlichen diese beiden Komponenten in Beziehung. Unter H_0 sollten die Varianzenkomponenten sich nicht unterscheiden und die Testgröße deshalb bei Eins liegen.

Die Varianzanalysetafel

- Die auftretenden Werte der Varianzanalyse werden gern in der **Varianzanalysetafel** zusammengefasst.

- Das Schema dieser Tafel ist wie folgt:

Streuungsursache	df	Quadratsumme	Mittlere Quadratsumme
Faktor 1	k-1	$SS(A) = \sum_{i=1}^{k} (\bar{Y}_{i.} - \bar{Y}_{..})^2$	$MS(A) = \frac{SS(A)}{k-1}$
zufälliger Fehler	N-k	$SS(E) = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{Y}_{i.})^2$	$MS(E) = \frac{SS(E)}{N-k}$
Gesamt	N-1	$SS(E) = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{Y}_{..})^2$	

- Die F-Statistik ergibt sich dann als $\frac{MS(A)}{MS(E)}$.

Fortführung des einführenden Beispiels

- Aber nicht von Hand, sondern in R!

```
lab1 <- c(4.13, 4.07, 4.04, 4.07, 4.05, 4.04)
lab2 <- c(3.86, 3.85, 4.08, 4.11, 3.83, 4.01)
lab3 <- c(4.00, 4.02, 4.01, 4.01, 4.04, 3.99)
ydata<-c(lab1, lab2, lab3)
xdata <- as.factor(c(rep("lab1", 6),rep("lab2", 6), rep("lab3", 6))
?anova
aov1 <- lm(ydata ~ xdata)
anova(aov1)
            Df   Sum Sq  Mean Sq F value  Pr(>F)
xdata        2 0.036300 0.018150  3.1823 0.07046
Residuals   15 0.085550 0.005703
```

Weiterführende Bemerkungen zur einfaktoriellen Varianzanalyse

- Die kritischen Annahmen der Varianzanalyse sind die Unabhängigkeit der Beobachtungen und die identische Normalverteilung der Fehler.

- Die Unabhängigkeit kann insbesondere bei Meßwiederholungen verletzt sein. Bei Vorliegen von Meßwiederholungen ist deshalb diese Eigenschaft besonders zu betrachten. (Varianz innerhalb der Gruppe) Die Annahme der Varianzgleichheit innerhalb der Gruppen sollte auch durch Kenntnisse der Fachwissenschaftler im jeweiligen Anwendungsgebiet unterfüttert werden.

- Wenn die Normalverteilungsannahme verletzt ist, kann man auf verteilungsfreie Tests ausweichen. Stichworte sind hier der Kruskal-Wallis-Test und der Wilcoxon-Rangsummen-Test.

Weiterführende Bemerkungen zur einfaktoriellen Varianzanalyse

- Bei unbalancierten Versuchsplänen, d.h. unterschiedlichen n_i ist die Voraussetzung der gleichen Varianzen in den Gruppen essentiell. Der Effekt ungleicher Varianzen bei unbalancierten Designs ist nicht kontrollierbar. (s. Schlittgen , Statistik , Oldenbourg, p. 350f)

Lösungen der Aufgaben

- Erwartungstreue von $\hat{\mu}$

$$
\begin{aligned}
E(\hat{\mu}) &= E\left(\frac{1}{N}\sum_{1}^{k}\sum_{1}^{n_i} y_{ij}\right) \\
&= E\left(\frac{1}{N}\sum_{1}^{k}\sum_{1}^{n_i} (\mu + \varepsilon_{ij})\right) \\
&= \underbrace{E\left(\frac{1}{N}\sum_{1}^{k}\sum_{1}^{n_i} \mu\right)}_{=\mu} + \underbrace{E\left(\frac{1}{N}\sum_{1}^{k}\sum_{1}^{n_i} \varepsilon_{ij}\right)}_{=0} \\
&= \mu
\end{aligned}
$$

□

Lösungen der Aufgaben

- Erwartungstreue von $\hat{\alpha}_i$

$$
\begin{aligned}
E(\hat{\alpha}_i) &= E(\bar{Y}_{i.} - \bar{Y}_{..}) \\
&= E\left(\frac{1}{n_i} \sum_{1}^{n_i} y_{ij} - \underbrace{\frac{1}{N} \sum_{1}^{k} \sum_{1}^{n_i} y_{ij}}_{=\mu} \right) \\
&= E\left(\underbrace{\frac{1}{n_i} \sum_{1}^{n_i} (\mu + \alpha_i + \varepsilon_{ij})}_{=\mu+\alpha_i} - \mu \right) \\
&= \mu + \alpha_i - \mu = \alpha_i
\end{aligned}
$$

Lösungen der Aufgaben

- Erwartungstreue von $\hat{\sigma}^2$

- bekannt aus Stat II: $E(\hat{\sigma}_i^2) = E\left(\frac{1}{n_i-1}\sum_1^{n_i}(y_{ij} - \bar{Y}_{i.})^2\right) = \sigma^2$. Damit:

$$\begin{aligned} E(\hat{\sigma}^2) &= E\left(\frac{1}{N-k}\sum_1^k (n_i - 1)\hat{\sigma}_i^2\right) \\ &= \frac{1}{N-k}\left[(n_1-1)\underbrace{E(\hat{\sigma}_1^2)}_{=\sigma^2} + (n_2-1)\underbrace{E(\hat{\sigma}_2^2)}_{=\sigma^2} + \ldots + (n_k-1)\underbrace{E(\hat{\sigma}_k^2)}_{=\sigma^2}\right] \\ &= \frac{N-k}{N-k}\sigma^2 = \sigma^2 \end{aligned}$$

Einlesen von `sturmfluten.csv`

- Es ist lediglich ein Aufruf von `read.csv2()` mit den korrekten Parametern nötig.

- Im Anschluss kann man noch die leeren Zellen entfernen!

```
floods <- read.csv2("1950-2005-Sturmfluten.csv",skip=2)
floods <- floods[1:211,1:4]
```

Welche(s) Paar(e) ist(sind) denn nun signifikant verschieden?

- Der F-Test der Varianzanalyse erlaubt lediglich eine Existenzaussage über ein Paar (i,j) mit $\mu_i \neq \mu_j$ zum einem Niveau α.

- In der Praxis interessiert evtl. auch, welches Paar die signifikanten Abweichungen zeitigt.

- Für den Vergleich zweier Mittelwerte (i,j) kennen wir den Zweistichproben-t-Test mit der Teststatistik

$$T_{ij} = (\bar{Y}_{i.} - \bar{Y}_{j.}) / \sqrt{\hat{\sigma}^2 \left(\frac{1}{n_i} + \frac{1}{n_j} \right)},$$

wobei $\hat{\sigma}^2 = \frac{1}{N-k}\sum_{i=1}^{k}(n_i - 1)\hat{\sigma}_i^2$ die Gesamtvarianzschätzung ist, die auf allen Beobachtungen beruht.

- Der Zweistichprobentest hält das Niveau α aber lediglich für einen einzelnen Test ein.

- Welches Niveau kann man garantieren, wenn man mehrere Tests simultan durchführt?

Problematik des multiplen Testens I

- Beispiel: Betrachte zwei Tests zum Niveau α mit den Ereignissen $A_i =$ "lehne Hypothese H_i ab" $i = 1, 2$. Welches simultane Niveau hält ein Test für beide Hypothesen gemeinsam ein?

$$\begin{aligned} P(\text{mind. eine Hypothese wird verworfen}|H_1, H_2) &= P(A_1 \cup A_2|H_1, H_2) \\ P(A_1|H_1, H_2) + P(A_2|H_1, H_2) - P(A_1 \cap A_2|H_1, H_2) &= \\ 2\alpha - P(A_1 \cap A_2|H_1, H_2) & \end{aligned}$$

Problematik des multiplen Testens II

- Ist nun aber $P(A_1 \cap A_2 | H_1, H_2) < \alpha$, so hält der simultane Test das Niveau nicht ein.

- Um aber das oder die signifikant unterschiedliche(n) Mittelwertpaar(e) im Modell der einfachen Varianzanalyse zu finden, sind bei k Faktorstufen $\binom{k}{2}$ paarweise Vergleiche durchzuführen.

- Lösung: Das Niveau der Einzeltests wird so angepasst, dass das simultane Niveau α garantiert werden kann!

- Aus der Wahrscheinlichkeitsrechnung ist die **Bonferroni**-Ungleichung bekannt.
  In der einfachsten Form besagt diese für Ereignisse $A_i, i = 1, \ldots, k$:

$$P(A_1 \cup \ldots \cup A_k) \leq P(A_1) + \ldots + P(A_k).$$

Problematik des multiplen Testens III

- Inhaltlich ist das sofort klar, wenn man überlegt, dass für zwei Ereignisse A_1, A_2 gilt: $P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2)$.

- Somit kann garantiert werden, dass ein simultaner Test aus m Einzeltests das multiple Niveau α einhält, wenn jeder Einzeltest das Niveau α/m einhält!

- Für eine nicht zu große Zahl k von Faktorstufen können also alle paarweisen Vergleiche im Modell der einfaktoriellen Varianzanalyse zum multiplen Niveau α durchgeführt werden, wenn als kritische Werte der Einzeltests die Quantile $t_{N-k;1-\alpha^*/2}$ mit $\alpha^* = \frac{\alpha}{\binom{k}{2}}$ gewählt werden.

Multiples Testen im Experiment

- Das multiplen Testen scheint also eine immer wiederkehrende Problematik zu sein. Welchen Fehler macht man denn, wenn man keine Niveauanpassung vornimmt?

- Bei mehreren simultanen Tests zu einem festgelegten Niveau α an der selben Datenbasis (!) wird nicht das simultane Niveau α für alle Tests gleichzeitig eingehalten!

- Vielmehr läßt sich die Niveauänderung, in diesem Fall Niveauverlust, für **unabhängige** Tests einfach berechnen.

- Für einen einzelnen Test gilt bekanntlich, dass die Wahrscheinlichkeit, korrekterweise die Hypothese abzulehen, 1- α beträgt.

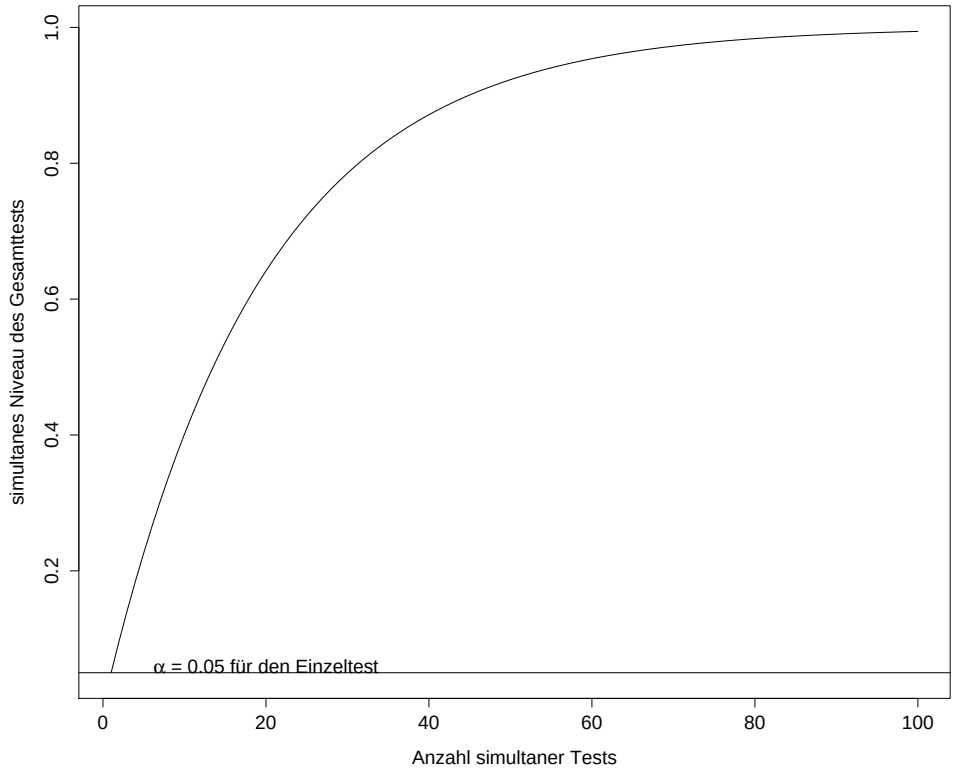
- Für n unabhängige Tests, die ja jeder für sich ein Zufallsexperiment darstellen, beträgt diese Wahrscheinlichkeit folglich $(1-\alpha)^n$.

- Damit beträgt das simultane Niveau für unabhängige Tests $1-(1-\alpha)^n$

- Um den Sachverhalt zu verdeutlichen wird heute ein Computerexperiment, zur "Erfahrbarmachung" dieses Sachverhalts durchgeführt!

- Damit ergibt sich einerseits eine Anschauung, andererseits eine Quantifizierung des Fehlers!

Zusammenhang zwischen simultanem Niveau und Anzahl unabhängiger Tests



Design des Experiments I

- Das Experiment soll simultan mehrere Tests durchführen und experimentell Hinweise auf die Höhe des Niveauverlusts durch simultane Tests geben.

- Als Anwendungsbeispiel werden die Signifikanztests für die Koeffizienten der linearen Regression gewählt.

- Dazu müssen Einflussfaktoren und Zielgrößen unter der Nullhypothese simuliert werden.

- Für die Zielgröße bedeutet dies, dass in der Simulation die Einflußfaktoren **keinen** Einfluß auf den Wert der Zielgröße haben (also alle $\beta_i = 0$).

Design des Experiments I

- Die Anzahl der (vermutlichen) Einflußfaktoren wird nach und nach erhöht und jeweils beobachtet, ob die Regressionsanalyse einen der Einflußfaktoren als signifikant einschätzt.

- Dieses Experiment wird mehrfach (10000-fach) wiederholt.

- Man erhält für jede Anzahl von vermutlichen Einflußgrößen einen empirischen Schätzer für das simultane Niveau des Tests.

Diskussion

- Großer Vorteil von Computerexperimenten gegenüber analytischen Ergebnissen ist die Flexibilität im Versuchsdesign. Auch Situationen, die analytisch nicht oder nicht leicht zu lösen sind, lassen sich numerisch lösen.

- Wichtiges Beispiel: abhängige Beobachtungen.

- Wichtigster Nachteil: oft sind die genauen Eigenschaften der Simulationsschätzer unbekannt.

Umsetzung in R und Durchführung

- Nachbildung der Modellannahmen im Computer. Es ist nötig geeignete Parameter für die Simulation zu wählen.

- Hier $n = 100, Var(\varepsilon) = 1, \alpha = 0.05$, Wahl dieser Parameter beeinflußt nicht das Ergebnis.

- Unter der Nullhypothese $Y = \varepsilon$ mit $\varepsilon \sim N(0, 1)$.

- In R: `yseq <- rnorm(100)`

- Einflußfaktoren werden einfach als Stichprobe aus der Gleichverteilung über [0,1] gezogen.

- In R: `xseq <- runif(100)`

- Signifikanzniveau (p-Wert) für eine Modellschätzung aus diese Daten anschauen.

- In R: `summary(lm(yseq ~ xseq - 1))$coef[1,4]`

- Damit ist **ein** Versuch beendet.

```
yseq <- rnorm(100)
xseq <- runif(100)
summary(lm(yseq ~ xseq - 1))$coef[1,4]
```

- Bei mehreren Faktoren wird die Nullhypothese abgelehnt, wenn für irgendeinen Koeffizienten das Signigfikanzniveau α unterschritten wird. Es reicht also, den kleinsten p-Wert anzuschauen und jeweils diesen mit α zu vergleichen. In R fällt die Entscheidung

über das Verwerfen der Hypothese durch Vergleich des Ausdrucks `min(summary(lm(yseq ~ xseq - 1))$coef[,4])` mit α.

- Wiederholung des Experiments in einer Schleife und Sammeln der Ergebnisse in einem Ergebnisvektor (Pseudocode!):

```
result <- rep(NA,10000)
for (counter in 1:10000){
     Experiment  ### symbolisch!
     result[counter]<- Ergebnis Experiment
}
```

- Die Anzahl verworfener Hypothesen zum Niveau $\alpha = 0.05$ ergibt sich damit als: `sum(result <= 0.05)`
  Diese Zahl ist ein Schätzer für das simultane Niveau der durchgeführten Tests.

- Warum 10000 Wiederholungen? Ein 95% KI für den Parameter p einer Binomialverteilung ergibt sich zu

$$[\hat{p} - 1.96 \cdot \frac{1}{n}\sqrt{n\hat{p}(1-\hat{p})} \quad ; \quad \hat{p} + 1.96 \cdot \frac{1}{n}\sqrt{n\hat{p}(1-\hat{p})}].$$

  Bei $n = 10000$ und $p \approx 0.05$ ist dieses Intervall ca. 0.01 lang!

- Das ganze wird für einige Anzahlen von Einflußfaktoren durchgeführt.

- Schließlich soll noch die Bonferronianpassung auf ihre Wirksamkeit überprüft werden. Auch hierzu wird ein entsprechendes Computerexperiment mit angepasstem Niveau α^* durchgeführt.

- Der hier vorgestellte Code ist leicht verallgemeinerbar, indem man eine Matrix von Daten als Einflußgrößen nutzt. Hier aus Gründen der Übersichtlichkeit als einzelne Variablen.

Durchführung

```
> result <- rep(NA,10000)
> ### 10000 Regressionen unter der Nullhypothese
> ### Ohne Achsenabschnitt
> for (counter in 1:10000)
+ {
+   xseq <- runif(100)
+   yseq <- rnorm(100)
+   result[counter] <- summary(lm(yseq ~ xseq - 1))$coef[1,4]
+ }
> sum(result <= 0.05)
[1] 496
```

- Das Niveau wird also eingehalten für einen einzelnen Signifikanztest.

Durchführung

```
> result <- rep(NA,10000)
> ### 10000 Regressionen unter der Nullhypothese
> ### Mit Achsenabschnitt
> for (counter in 1:10000)
+ {
+   xseq <- runif(100)
+   yseq <- rnorm(100)
+   result[counter] <- min(summary(lm(yseq ~ xseq ))$coef[,4])
+ }
> sum(result <= 0.05)
[1] 713
```

- Achsenabschnitt und Koeffizient sind nicht unabhängig
- Das Niveau sinkt in etwa auf 7%.

Durchführung

```
> result <- rep(NA,10000)
> ### 10000 Regressionen unter der Nullhypothese
> ### 2. Faktor
> for (counter in 1:10000)
+ {
+   xseq <- runif(100)
+   qseq <- runif(100)
+   yseq <- rnorm(100)
+   result[counter] <- min(summary(lm(yseq ~ xseq + qseq))$coef[,4]
+ }
> sum(result <= 0.05)
[1] 1164
```

- Niveau sinkt auf ca 11-12% bei Hinzunahme eines weiteren unabhängigen Einflußfaktors.

Durchführung

```
> result <- rep(NA,10000)
> ### 10000 Regressionen unter der Nullhypothese
> ### 3. Faktor
> for (counter in 1:10000)
+ {
+   xseq <- runif(100)
+   qseq <- runif(100)
+   wseq <- runif(100)
+   yseq <- rnorm(100)
+   result[counter] <-
+               min(summary(lm(yseq ~ xseq + qseq + wseq))$coef[,4]
> sum(result <= 0.05)
[1] 1590
```

- Niveau ca. 16% bei 3 Faktoren.

Durchführung der Bonferronikorrektur

```
> result <- rep(NA,10000)
> ### Bonferroni
> ### 10000 Regressionen unter der Nullhypothese
> for (counter in 1:10000){
+   xseq <- runif(100)
+   qseq <- runif(100)
+   wseq <- runif(100)
+   yseq <- rnorm(100)
+   result[counter] <-
+      min(summary(lm(yseq ~ xseq + qseq + wseq))$coef[,4]) }
> sum(result <= 0.05/4)
[1] 413
```

- Die Bonferronikorrektur erzwingt das simultane Niveau α für alle Tests, ist aber extrem konservativ!

Lösung zur Aufgabe mit dem Sturmflutendatensatz

- Zunächst einfach einlesen:

```
floods <- read.csv2("sturmfluten.csv",skip=2, as.is=TRUE)
floods <- floods[1:211,1:4]
```

- Jetzt sollten auch noch die drei Spalten mit Pegelständen kollabiert werden, sowie eine kategorielle Spalte eingeführt werden, die die Schwere der Flut enthält.

```
kategorie <- rep(NA,211)
for (zeile in 1:dim(floods)[1]) {
   if (!is.na(floods[zeile,2])) kategorie[zeile] <- "normal"
   if (!is.na(floods[zeile,3])) kategorie[zeile] <- "schwer"
   if (!is.na(floods[zeile,4])) kategorie[zeile] <- "sehr schwer"
   floods[zeile, 2] <- max(floods[zeile,2:4],na.rm=TRUE)}
```

- Reduktion auf die interessanten Spalten

```
floods <- cbind(floods[1:2],as.factor(kategorie))
names(floods) <- c("Datum", "Pegel in cm", "Kategorie")
rm(kategorie)
```

- Standardplot der Pegelstände:

```
plot(floods[,2],t="l")
```

- Leider ist der zeitliche Abstand zweischen den Sturmfluten nicht sichtbar!

- Wie kann ich diese Information aus dem Datensatz nutzen?

Exkurs: Datumsinformation in `R`

- Daten sind extrem wichtige Datentypen!
- Aber sehr schwierig im Rechner zu handhaben: Sommerzeit, Zeitzone, Rechnerzeit, Schaltjahre, Schaltsekunden etc.
- `?DateTimeClasses` implementiert POSIX konforme Daten- und Zeitklassen in `R`.
- Wichtigste Funktion: `strptime()` (string to posix time).
- Standardgenanuigkeit auf allen Rechner ist 1s. Auf den meisten Rechnern heute eine Auflösung im Bereich einer Mikrosekunde implementiert.

Exkurs: Datumsinformationen in R (Beispielsitzung)

```
dates <- c("12/15/92","12/20/95","12/25/97")
times <- c("10:01:00","06:00:00","02:30:00")
x <- paste(dates, times)
x
(z <- strptime(x, "%m/%d/%y %H:%M:%S"))
class(z)
z[2]-z[1]
as.Date(z)
ISOdate ( 2008, 10, 9, 10, 30)
format(Sys.time(), "%a %b %d %H:%M:%S %Y")
```

Verbesserter Plot

- Nutzen der Datumsinformation! Die x-Achse soll die zeitlichen Abstände der Sturmfluten wiederspiegeln.

- ```
  floods <- cbind(floods, strptime(floods[,1],"%d.%m.%Y" ))
  plot(floods[,4], floods[,2],t="l" ,
         main="Sturmflutpegel in HH", xlab="Datum",
         ylab="Pegel in cm", axes=FALSE)
  axis(2)
  axis(1,at=floods[,4],label=floods[,4] )
  points(floods[,4], floods[,2],  col=floods$Kategorie)
  ```
  ```

Exkurs: Zusammenhang üblicher Verteilungen

- **Die** Grundannahme der klassischen Statistik ist die unabhängig identische Normalverteilung der Fehler $\varepsilon \sim N(0, \sigma^2)$.

- Aus dieser Grundannahme ergeben sich einige Verteilungen natürlicherweise durch die Modellschätzung und die damit verbundenen Transformationen des zufälligen Anteils ε.

- Im Folgenden seien alle $X_i \sim N(0, 1)$.

- Die Verteilung von $Z = X_1/X_2$ heißt Cauchy-Verteilung. Diese Verteilung ist das Standardbeispiel für eine Verteilung, deren Momente nicht existieren.

- Die Verteilung von $Z = \sum_1^n X_i^2$ heißt χ^2-Verteilung mit n Freiheitsgraden (χ^2_n). Es gilt $E(Z) = n$ und $Var(Z) = 2n$.

- Die Verteilung von $T = \frac{X}{\sqrt{U/\nu}}$, bei der $U \sim \chi^2_\nu$ unabhängig von X heißt student's t-Verteilung mit ν Freiheitsgraden. Es gilt $E(T) = 0$ für $\nu > 1$ und $Var(T) = \frac{\nu}{\nu-2}$ für $\nu > 2$.

- Die Verteilung von $F = \frac{U_1/\nu_1}{U_2/\nu_2}$ heißt F-Verteilung mit ν_1 und ν_2 Freiheitsgraden, wenn die U_i unabhängig $\chi^2_{\nu_i}$ verteilt sind. Es gilt $E(F) = \frac{\nu_2}{\nu_2-2}$, $Var(F) = \frac{2\nu_2^2(\nu_1+\nu_2-2)}{\nu_1(\nu_2-2)^2(\nu_2-4)^2}$ für $\nu_2 > 2, \nu_2 > 4$ resp.

- Hinweis: Diese Verteilungen lassen sich auch in allgemeineren Kontext auf die sogenannten Gamma-Funktionen zurückführen.

Zweifaktorielle Varianzanalyse

- Bei der zweifaktoriellen Varianzanalyse wird der Einfluss zweier Faktoren A und B auf eine Zielgröße Y untersucht.

- Als zusätzliche Fragestellung taucht auf, ob die Faktorstufen der verschiedenen Faktoren sich gegenseitig beeinflußen. Eine solche Beeinflußung heißt *Wechselwirkung*.

- Der Formalismus ist analog zur einfachen Varianzanalyse, jedoch werden Beobachtungen nun dreifach indiziert gemäß den beteiligten Faktorstufen.

- Y_{ijk} bezeichnet die k-te Beobachung auf der i-ten Faktorstufe des ersten Faktors A und der j-ten Faktorstufe des zweiten Faktors B.

Modelldarstellung der zweifaktoriellen Varianzanalyse

- Wie im einfaktoriellen Fall gibt es zwei äquivalente Darstellungen des Modells der zweifaktoriellen VA.

- Zum einen die Modelldarstellung mit individuellem Niveau je Faktorstufenkombination (i, j):

$$Y_{ijk} = \mu_{ij} + \varepsilon_{ijk}, \quad \varepsilon \sim N(0, \sigma^2)$$

  mit $i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K$ und alle ε_{ijk} unabhängig.

- Im Unterschied zur einfachen ANOVA wird hier vom balancierten Design ausgegangen.

- In diesem Modell ist die Interpretation reduziert den einfaktoriellen Fall mit IxJ Faktorstufen, insbesondere können keine Aussagen über die Wechselwirkungen getroffen werden.

- Deshalb nutzt man zur besseren Interpretation auch hier das Effekt-Modell:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk},\ \varepsilon \sim N(0, \sigma^2)$$

  mit $i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K$ und alle ε_{ijk} unabhängig.

- Analog zum einfaktoriellen Modell ergibt sich die Eigenschaft der sich gegenseitig aufhebenden Effekte:

$$\sum_{1}^{I} \alpha_i = 0,\ \sum_{1}^{J} \beta_j = 0,\ \sum_{i=1}^{I} (\alpha\beta)_{ij} = 0,\ \sum_{j=1}^{J} (\alpha\beta)_{ij} = 0.$$

Haupteffekte und Wechselwirkungen

- Es bezeichne

$$\mu = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \mu_{ij}$$

  das allgemeine Mittel.

- Dann beschreibt $\alpha_i = \mu_{i.} - \mu$ mit $\mu_{i.} = \frac{1}{J} \sum_{j=1}^{J} \mu_{ij}$, dem Erwartungswert für Faktor A auf Stufe i ohne Betrachtung von Faktor B, den *Haupteffekt (Effekt) von Faktor A auf Stufe* i.

- Entsprechend bezeichnet $\beta_j = \mu_{.j} - \mu$ mit $\mu_{.j} = \frac{1}{I} \sum_{i=1}^{I} \mu_{ij}$, den *Effekt von Faktor B auf Stufe* j.

- $(\alpha\beta)_{ij} = \mu_{ij} - (\mu + \alpha_i + \beta_j)$ heißt die *Wechselwirkung der Stufe* i *von Faktor A mit Stufe* j *von Faktor B*.

Schätzung im zweifaktoriellen Modell

- Die Schätzer im zweifaktoriellen Modell ergeben sich direkt aus den Modellformulierungen, analog zum einfaktoriellen Fall.

- Das globale Mittel μ wird geschätzt durch

$$\hat{\mu} = \frac{1}{IJK} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} Y_{ijk} = \bar{Y}_{...} \quad .$$

- Die Mittel auf den jeweiligen Faktorstufen werden geschätzt durch

$$\hat{\mu}_{i.} = \bar{Y}_{i..} = \frac{1}{JK} \sum_{j=1}^{J} \sum_{k=1}^{K} Y_{ijk} \text{ und } \hat{\mu}_{.j} = \bar{Y}_{.j.} = \frac{1}{IK} \sum_{i=1}^{I} \sum_{k=1}^{K} Y_{ijk}$$

- Damit ergeben sich als Haupteffektschätzer

$$\hat{\alpha}_i = \bar{Y}_{i..} - \bar{Y}_{...} \text{ und } \hat{\beta}_j = \bar{Y}_{.j.} - \bar{Y}_{...}.$$

- Schließlich lässt sich der Wechselwirkungsschätzer $\widehat{(\alpha\beta)}_{ij}$ schreiben als

$$\widehat{(\alpha\beta)}_{ij} = \bar{Y}_{ij.} - (\hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j).$$

- Mit $\bar{Y}_{ij.} = \frac{1}{K}\sum_{k=1}^{K} Y_{ijk}$ ergibt sich auch

$$\widehat{(\alpha\beta)}_{ij} = \bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}$$

- Die Residuen berechnen sich zu

$$\hat{\varepsilon}_{ijk} = Y_{ijk} - (\hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j + \widehat{(\alpha\beta)}_{ij}).$$

Hypothesentests in der zweifaktoriellen ANOVA

- Für die Haupteffekte wird jeweils die Hypothese aus der einfaktoriellen ANOVA getestet, dass alle Haupteffekte Null sind gegen mindestens zwei Haupteffekte sind ungleich Null.

- Hinzu tritt die Hypothese für die Wechselwirkungen:

$$H_0^{AxB} : (\alpha\beta)_{ij} = 0 \text{ für alle Paare } (i,j) \text{ gegen}$$
$$H_1^{AxB} : \text{für mindestens zwei Paare } (i,j) \text{ gilt } (\alpha\beta)_{ij} \neq 0.$$

- Grundlage der Teststatistiken ist wie bereits im einfaktoriellen die Zerlegung der Varianz in die von den einzelnen Modellkomponenten erklärten Anteile und das in Beziehung setzen der verschiedenen Anteile.

- Im vorliegenden Modell gilt:

$$SQT = SQA + SQB + SQ(A\text{x}B) + SQR$$

wobei

$$
\begin{aligned}
SQT &= \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(Y_{ijk}-\bar{Y}_{...})^2, \\
SQA &= KJ\sum_{i=1}^{I}(\bar{Y}_{i..}-\bar{Y}_{...})^2, \\
SQB &= KI\sum_{j=1}^{J}(\bar{Y}_{.j.}-\bar{Y}_{...})^2, \\
SQ(A\times B) &= K\sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{\alpha\beta})_{ij}^2 \text{ und} \\
SQR &= \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(Y_{ijk}-\bar{Y}_{ij.})^2.
\end{aligned}
$$

Varianzanalysetafel im zweifaktoriellen Fall

Ursache	Streuung	df	MSE	Statistik
Faktor A	SQA	I-1	$MQA = \frac{SQA}{I-1}$	$F_A = \frac{MQA}{MQR}$
Faktor B	SQB	J-1	$MQB = \frac{SQB}{J-1}$	$F_B = \frac{MQB}{MQR}$
AxB (Wechselw.)	SQ(AxB)	(I-1)(J-1)	$MQ(A\text{x}B) = \frac{SQ(A\text{x}B)}{(I-1)(J-1)}$	$F_{A\text{x}B} = \frac{MQ(A\text{x}B)}{MQR}$
Residuen	SQR	IJ(K-1)	$MQR = \frac{SQR}{IJ(K-1)}$	
Gesamt	SQT	n-1		

- Die kritischen Werte sind jeweils der F-Verteilung mit den Freiheitsgraden der Zähler und des Nenners der Prüfgröße zu entnehmen.

- Beispielsweise ist $H_0^{A\text{x}B}$ zu verwerfen, wenn

$$F_{A\text{x}B} > F_{1-\alpha}((I-1)(J-1), IJ(K-1)) \quad .$$

Beispiel für zweifaktorielle ANOVA

- Hinzufügen eines zweiten Faktors zum Laborbeispiel. Zur Erinnerung, folgende Daten liegen vor:

Faktorstufe	Messungen					
Labor 1	4.13	4.07	4.04	4.07	4.05	4.04
Labor 2	3.86	3.85	4.08	4.11	3.83	4.01
Labor 3	4.00	4.02	4.01	4.01	4.04	3.99

- Ziel ist die Überprüfung des Vorhandenseins des sogenannten Laboranteneffektes. Angenommen, es gäbe zwei Laboranten, die reihum in den drei Labors arbeiten. Unterscheiden sich die Messergebnisse je nach messendem Laboranten?

- Angenommen in jedem Labor habe jeder der beiden Laboranten je 3 Versuche durchgeführt (Balanciertheit!).

- Umgesetzt in R ergibt sich ein neuer Datenvektor für den Faktor *Laborant*:

```
laboranten<- as.factor( rep(c(rep("laborant1",3),
                              rep("laborant2",3)),3))
```

- Durchführen der ANOVA in R und die resultierende Varianzanalysetafel:

```
# Achtung: Verbesserung der Schreibweise gegenüber summary(...)
anova(lm(ydata ~ xdata*laboranten ))
             Df   Sum Sq  Mean Sq F value Pr(>F)
xdata         2 0.036300 0.018150  2.7157 0.1064
laboranten    1 0.000450 0.000450  0.0673 0.7997
xdata:labor.  2 0.004900 0.002450  0.3666 0.7006
Residuals    12 0.080200 0.006683
```

- Die Interpretation wäre nun, dass evtl. das Labor einen Einfluss auf das Messergebniss hat, nicht jedoch der Laborant oder die Wechselwirkung zwischen Labor und Laborant.

Der Interaktionsplot

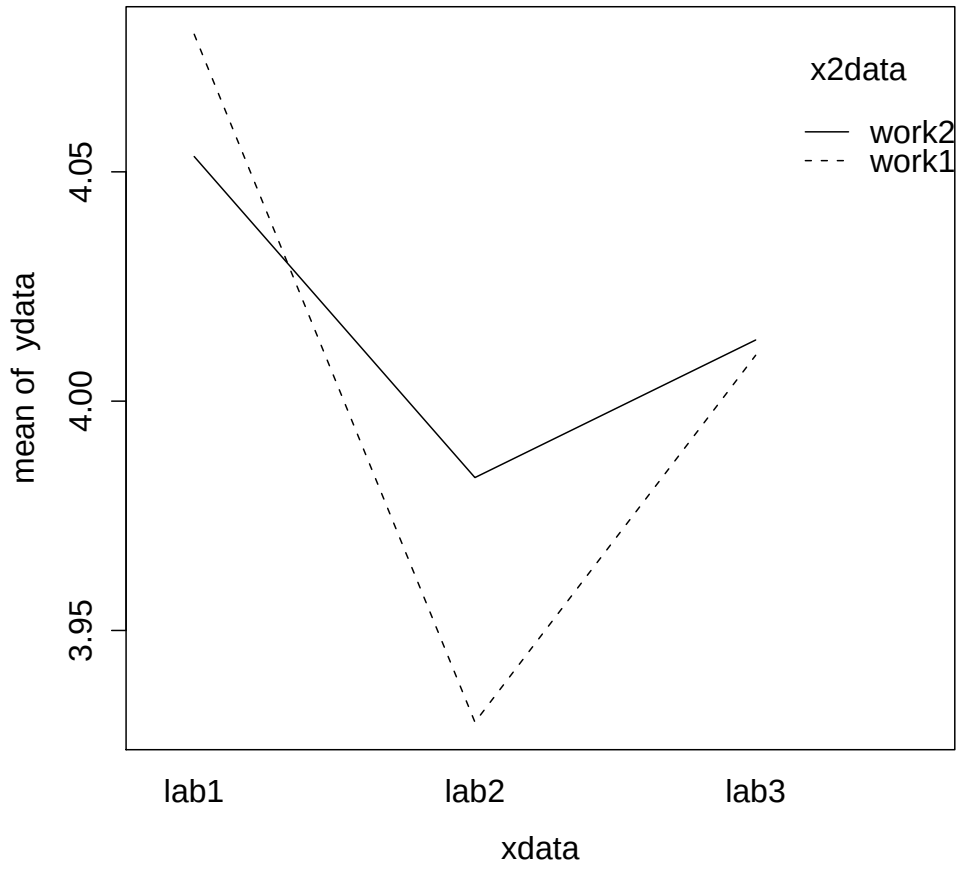
- Der Interaktionsplot (*interaction plot*) bietet eine grafische Möglichkeit, um auf einen Blick das Vorhandensein und die Richtung einer evtl. Interaktion zu beurteilen.

- Im Interaktionsplot wird auf der X-Achse einer der Faktoren, oBdA Faktor A, abgetragen, auf der Y-Achse die gemessene Zielgröße.

- Für jede Faktorstufe j des anderen Faktors B wird dann ein Linienzug eingezeichnet, der die Mittelwerte der Beobachtungen von Faktor A auf den den Faktorstufen i bei Faktorstufe j von B verbindet.

- Beispiel für einen Interaktionsplot mit den Beispieldaten für Labor und Praktikanten. In R: `interaction.plot(xdata,laboranten,ydata))`.

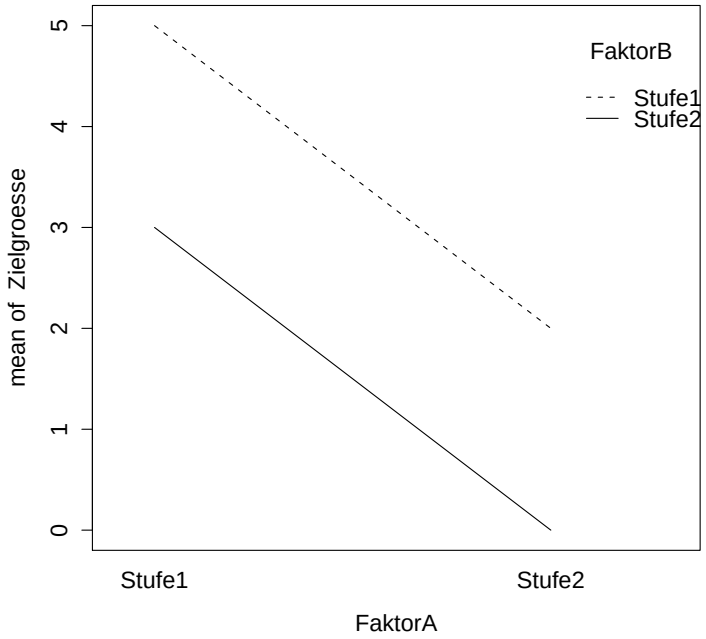
Beispiel für einen Interaktionsplot (Laborbeispiel)



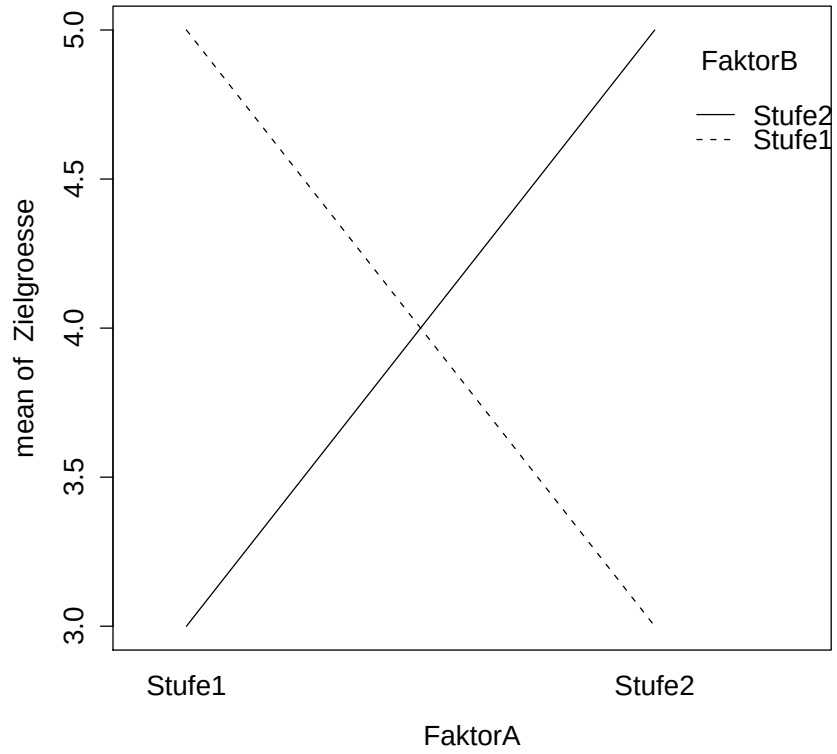
Interaktionsdiagnose anhand von Interaktionenplots

- Man unterscheidet drei Fälle:

  1. **Keine Wechselwirkung**. Liegt keine Wechselwirkung vor, dann liegen die Linienzüge parallel.



2. **Reine Wechselwirkung**. Bei einer reinen Wechselwirkung haben die Steigungen der Linienzüge umgekehrte Vorzeichen.

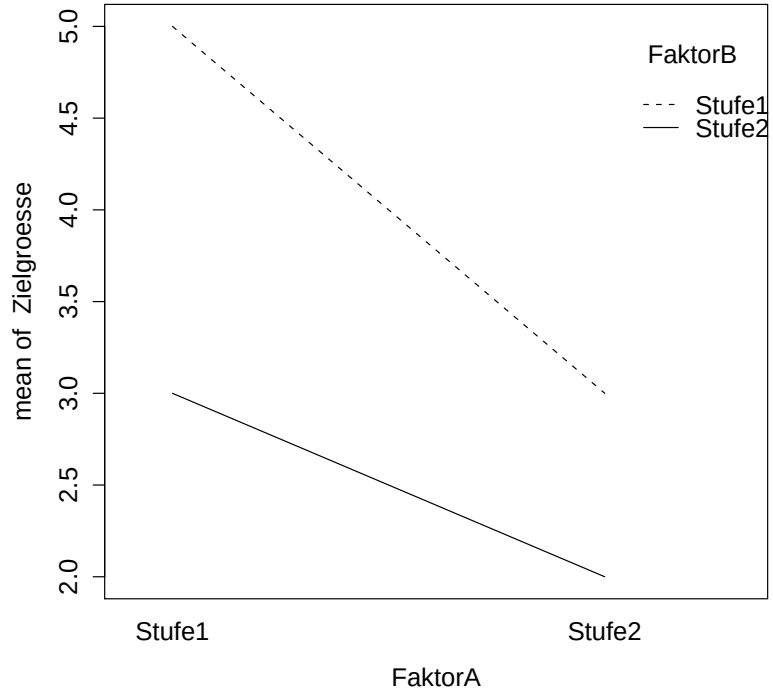


Der beobachtete Effekt wird von der Ausprägung der Kombination der beiden Faktorstufen dominiert. Kodiert man Stufe 1 für beide Faktorstufen mit -1 und Stufe 2 mit 1, dann liegen die hohen Beobachtungswerte jeweils vor, wenn

Faktorstufe von Faktor A x Faktorstufe von Faktor B $= 1$

ist. Entsprechend die niedrigen Beobachtungswerte, wenn dieses Produkt -1 ist. Man kann dann die Kombination (i, j) der Faktorstufen als einzelnen Faktor auffassen. Eine solche Bezeichnung der Faktorstufen mit $+1$ oder -1 ist in der Versuchsplanung üblich.

3. **Haupteffekte und Wechselwirkungen**. Liegen die Linienzüge nicht parallel, haben aber auch keine umgekehrten Vorzeichen, dann liegen Haupteffekte und Wechselwirkungen vor.



Modellkodierung für die Varianzanalyse

- Offensichtlich wird in R eine Regression gerechnet. Welche?
- Aus dem Effektmodell der Varianzanalyse

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk},\ \varepsilon \sim N(0, \sigma^2)$$

  kann man ablesen, wie die Kodierung für die Schätzung der Effekte der verschiedenen Faktorstufen zu geschehen hat.

- Analog zur Spalte für den Achsenabschnitt, welches bekanntermaßen als Spalte nur mit Einsen in der Desingmatrix auftaucht, werden die verschiedenen Faktorstufen durch sogenannte Dummy-Variablen in das Modell eingebracht.

- Für jede Faktorstufe i eines jeden Faktors A wird eine Dummy-Variable X_{iA} in das Modell eingeführt, welche den Wert 1 annimmt, wenn bei der Beobachtung der Faktor A auf Stufe i beobachtet wird und den Wert 0 sonst.

- Mit diesen Dummy-Variablen wird dann eine gewöhnliche Regression gerechnet und die Koeffizientenschätzer der Regression werden zu Effektschätzern.

- Wenn bei den Einflußfaktoren kategorielle und metrische Variablen gemeinsam auftreten, gelangt man in das Gebiet der Kovarianzanalyse.

- Aufgabe: Stellen Sie die Designmatrix X für das Laborbeispiel mit Laboranten auf!

Aufgabe Varianzanalyse

- Die mittlere Überlebenszeit von vier Tierarten unter der Gabe von drei Arten eines Rattengifts war:

	Tierart			
Gift	1	2	3	4
1	0.41	0.88	0.57	0.61
2	0.32	0.82	0.38	0.67
3	0.21	0.34	0.24	0.33

- Führen Sie die angemessene Varianzanalyse durch.

- Welchen Effekt hat die Erhöhung des Wertes 0.34 in Zelle (3,2) auf 0.51 auf die Wechselwirkung.

- Die Aufgabe ist mit `R` zu lösen.

Musterlösung der Aufgaben vom 24.3.

- Aufstellen der Designmatrix der zweifaktoriellen ANOVA für das Laborantenbeispiel.

- Die Faktoren heißen L (Labor) und A (Angestellter, Laborant), die Zielgröße Y (gem. Wirkstoffmenge). Ziel ist eine Darstellung der Art $Y = X\beta + \varepsilon,\ \varepsilon \sim N(0, \sigma^2)$ für das Effektmodell der Varianzanalyse mit Wechselwirkung $Y_{ijk} = \mu + \mathsf{L}_i + \mathsf{A}_j + (\mathsf{L} \cdot \mathsf{A})_{ij} + \varepsilon_{ijk},\ \varepsilon \sim N(0, \sigma^2)$ herzuleiten.

- Lösung: Mit X gegeben wie auf der folgenden Seite ergibt sich die gewünschte Darstellung!

$$X = \begin{pmatrix}
\mu & L_1 & L_2 & L_3 & A_1 & A_2 & L_1A_1 & L_1A_2 & L_2A_1 & L_2A_2 & L_3A_1 & L_3A_2 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Musterlösung zum 24.3.; Beispiel ANOVA

- Das Hauptproblem bei dieser Aufgabe ist, die Tabelle in einen *dataframe* zu verwandeln.

```
uezeit <- c(0.41, 0.88, 0.57, 0.61, 0.32, 0.82, 0.38, 0.67,
            0.21, 0.34, 0.24, 0.33)
gift <- as.factor(c(rep("gift1",4), rep("gift2",4),
                    rep("gift3",4)))
tier <- as.factor(rep(c("tier1", "tier2", "tier3", "tier4"),3))
datenmatrix <- data.frame(cbind(gift, tier,uezeit))
attach(datenmatrix)
```

- Die eigentliche Durchführung der ANOVA ist dann ohne weitere Fallstricke möglich:

```
summary(aov(lm(uezeit ~ gift + tier )))
            Df   Sum Sq  Mean Sq F value  Pr(>F)
gift         1 0.227813 0.227813  6.8452 0.02798 *
tier         1 0.022427 0.022427  0.6739 0.43291
Residuals    9 0.299527 0.033281

summary(aov(lm(uezeit ~ gift + tier + gift:tier)))
            Df   Sum Sq  Mean Sq F value  Pr(>F)
gift         1 0.227813 0.227813  6.0850 0.03890 *
tier         1 0.022427 0.022427  0.5990 0.46120
gift:tier    1 0.000022 0.000022  0.0006 0.98104
Residuals    8 0.299505 0.037438
```

- Auch nach Änderung der Zelle (3,2) ändert sich nichts Wesentliches:

```
interaction.plot (gift, tier, uezeit)
uezeit[8] <-0.51
### wichtig: datenmatrix ändert sich durch diese Zuweisung nicht
interaction.plot (gift, tier, uezeit)
interaction.plot (gift, tier, uezeit)
summary(aov(lm(uezeit ~ gift + tier, with = datenmatrix)))
summary(aov(lm(uezeit ~ gift + tier + gift:tier)))
```

Praktischer Versuch zur zweifaktoriellen ANOVA

- Auswerten einiger Daten, die jetzt erhoben werden.

- Fragestellung: Wie hängt die Studienmotivation der Besucher dieser Vorlesung von zwei Faktoren ab:

  1. Familiäre Situation, d.h. hier *zur Zeit Leben in einer Beziehung lebend* oder *zur Zeit nicht in einer Beziehung lebend*, und
  2. Subjektive Einschätzung des bisherigen Studienerfolgs, d.h. hier *bisher eher zufrieden* oder *eher unzufrieden*.

- Die Zielgröße ist die Begeisterung für das Studium auf einer Skala von 1 bis 10, 1 entspricht *total unmotiviert*, 10 entspricht *kann gar nicht genug davon bekommen*.

- Hypothesen: Der bisherige Studienverlauf ist wichtig für die aktuelle Motivationssituation, die Beziehungssituation kann sich nachteilig auswirken; evtl. gibt es aber eine verstärkende Wirkung von Studienerfolg und guter Beziehung.

- Darüberhinaus wird die Datenerfassung und das Einlesen der Daten in R gezeigt.

Umfrage

Bitte füllen Sie den Fragebogen aus!
Anonymität ist zugesichert.

Datenerfassung

- Sinnvollerweise geschieht die Datenerfassung nicht in R!
- Entweder legt man eine Textdatei an oder erfasst die Daten mit einer Tabellenkalkulation (Excel, OpenOffice).
- Bei umfangreichen Studien geschieht die Datenerfassung stets redundant über speziell programmierte Eingabemasken und die Daten werden in Datenbanken gespeichert.
- Nach der Datenerfassung kann man die Daten z.B. in das sogenannte CSV-Format exportiern (*comma separated values*) oder manchmal auch direkt das Format der Tabellenkalkulation einlesen.

Datenauswertung

- Die Daten wurden in einem Arbeitsblatt von Openoffice eingegeben und dann als CSV Daten nach `Umfrageergebnisse.csv` exportiert. (Download über die Vorlesungsseite)

- Einlesen und Varianzanalyse liefern dann folgende Zeilen `R` Code:

```
umfrage <- read.csv2(file="Umfrageergebnisse.csv")
### Achtung: das Arbeitsverzeichnis muss korrekt gesetzt sein!
### Rekodieren
umfrage[,1] <- as.factor(umfrage[,1])
umfrage[,2] <- as.factor(umfrage[,2])
```

```
levels(umfrage[,1]) <- c("ja", "nein")
levels(umfrage[,2]) <- c("eher erfolgreich", "eher nicht")

attach(umfrage)
interaction.plot(Beziehung, Erfolg, Motivation)

aov(Motivation  ~ Beziehung + Erfolg, data=umfrage)
summary(aov(Motivation  ~ Beziehung + Erfolg, data=umfrage))
summary(aov(Motivation  ~ Beziehung*Erfolg, data=umfrage))
```

```
                Df Sum Sq Mean Sq F value  Pr(>F)
Beziehung        1  5.115   5.115  1.2241 0.28050
Erfolg           1 24.906  24.906  5.9602 0.02314 *
Beziehung:Erfolg 1  5.160   5.160  1.2349 0.27846
Residuals       22 91.933   4.179
```

Bisherige Lernziele

- Bisherige Vorlesungsteile

  1. Datenvorbereitung
  2. Crash Kurs R
  3. Datenhandling
  4. Explorative Verfahren
  5. Regressionsanalyse uni- und multivariat
  6. ANOVA ein- und zweifaktoriell

Lernziele Datenvorbereitung

- Um was für Daten handelt sich?
- Umgang mit *missing values*
- Kodierung der Daten

Lernziele R Einführung

- Warum `R` ?
- Elementares Rechnen mit `R` , Spezielle Zahlen (NaN etc.)
- Vergleichsoperatoren, numerische Gleichheit
- Nutzung des Hilfesystems
- Elementare Statistikfunktionen
- Installation von Zusatzpaketen
- Verteilungsbezogenen Funktionen `d-, p-, q-, r-`Funktionen
- Subsetting in `R` : [...], boolsche Indizierung, which

Lernziele Datenhandling

- Nie auf den Originaldaten arbeiten
- Navigation in R
- Wie kommen die Daten in mein Programm?
- Datensicherung

Lernziele Exploration

- Verschiedene Plots; dazu jeweils:
  - Für welche Daten geeignet?
  - Wie wird der Plot in R erzeugt?
  - Was kann ich am Plot erkennen?
- Was sind Ordnungsstatistiken?
- Empirische Verteilungsfunktionen anfertigen?

Lernziele Regression

- Was ist Regression? Für welche Daten? Modellvoraussetzungen.
- Durchführen von Variablentransformationen.
- Durchführen und Interpretation der Ergebnisse einer Regression.
- Der p-Wert, (adjustiertes) Bestimmheitsmaß , Designmatrix
- Streuungszerlegung interpretieren.
- Q-Q Plot, anfertigen und interpretieren.
- Prognosen im Linearen Modell.
- Variablenauswahl, Durchführung in `R` , Idee, Wechselwirkungen

Lernziele ANOVA

- Modell(e) der ANOVA verstehen mit Voraussetzungen
- Idee, für welche Daten wird ANOVA verwendet? (*factor*)
- Die Schätzer in R berechnen können.
- Varianzanalysetafel lesen und interpretieren können.
- Problematik multiplen Testens erklären können, Bonferroni

Klausurvorbereitung

- Ab sofort gibt es jede Woche eine umfangreichere Aufgabe, die den Klausuraufgaben in etwa entsprechen soll.
- Nächste Woche Probeklausur!
- Bitte Feedback geben!

Probeklausur Musterlösung

Aufgabe 1 Auf der Homepage zur Veranstaltung finnden Sie unter dem Punkt Klausurdaten eine Datei `elastic.csv`. Laden Sie diese herunter und speichern Sie sie an einem geeigneten Ort ab.

a) Mit welchem Befehl lesen Sie die Datei in eine Variable `elastic` ein?

```
elastic <- read.table(
  "http://fawn.hsu-hh.de/~steuer/Klausur/elastic.csv",
                  header=TRUE,sep=",", strip.white=TRUE)
str(elastic)
```

Die Datei enthält Beobachungen eines Versuchs, bei dem für zwei verschiedene Materialien von Gummibändern gemessen wurde, bei welcher Auslängung des Bandes (in mm) dieses wie weit (in cm) fliegt.

b) Fertigen Sie einen Plot an, aus dem sie erkennen können, ob die Materialien sich ähnlich verhalten. Begründen Sie die Wahl des Plots und beschreiben Sie Ihre Beobachtungen! Notieren Sie die Befehle, die Sie zur Anfertigung des Plots genutzt haben.

```
plot(elastic[,1], elastic[,2], col=elastic[,3])
boxplot(elastic[,2] ~ elastic[,3])
```

c) Fällt eine Beobachung besonders auf? Wenn ja, entfernen Sie diese für die weitere Analyse aus den Daten. Mit welchem Befehl identifizieren Sie den auffälligen Punkt? Mit welchem Befehl entfernen Sie ihn aus den weiteren Analysen. Notieren Sie die nötigen Befehle.

```
which(elastic[,2] > 1000)
elastic <- elastic[-7,]
```

d) Betrachten Sie ein lineares Regressionsmodell für den bereinigten Datensatz. Mit welchem Befehl berechnen Sie in R das Modell? Interpretieren Sie das Ergebnis. Sehen Sie ein inhaltliches Problem mit dem Ergebnis der Regression?

```
summary(lm(Distanz ~ Streckung, elastic))
Call: lm(formula = Distanz ~ Streckung, data = elastic)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -69.8805    19.6089  -3.564  0.00311 **
### Und was heisst das inhaltlich?

Streckung     5.7467     0.4239  13.557 1.92e-09 ***
---
Residual standard error: 15.27 on 14 degrees of freedom
Multiple R-squared: 0.9292,Adjusted R-squared: 0.9242
F-statistic: 183.8 on 1 and 14 DF,  p-value: 1.925e-09
```

e) Führen Sie die einfache lineare Regressionen für jedes der beiden Materiale durch. Was sind die Hauptunterschiede in den Regressionsergebnissen?

```
weich <- which(elastic$Material == "weich")
lm(Distanz ~ Streckung , elastic[weich,])
lm(Distanz ~ Streckung , elastic[-weich,])
plot()
```

f) Zeichnen Sie mit R in einen Scatterplot, der die Regressionsgeraden für beide Fälle enthält. Notieren Sie die Befehle. Fällt Ihnen bei Betrachtung der Residuen eine Idee zur Modellverbesserung ins Auge?

```
plot(elastic$Streckung, elastic$Distanz, col=elastic$Material)
abline(lm(Distanz ~ Streckung , elastic[weich,]),col=2)
abline(lm(Distanz ~ Streckung , elastic[-weich,]),col=1)
plot(lm(Distanz ~ Streckung , elastic[weich,]))
plot(lm(Distanz ~ Streckung , elastic[-weich,]))
### evtl leicht nicht-linear
```

Aufgabe 2: Erzeugen sie 100 Datenpunkte aus der Normalverteilung mit $\mu = 11$ und $\sigma = 3$. Zeichnen Sie die empirische Verteilungsfunktion und die theoretische Verteilungsfunktion für diese Punkte in einen Plot. Notieren Sie die Befehle, die dazu nötig sind.

```
curve(pnorm(x, mean=11, sd=3), 4,17, main="Vergleich ecdf und Vtg")
lines(ecdf(rnorm(100, mean=11, sd=3)), pch="+")
```

Aufgabe 3: a) Welche Wertemenge bezeichnet man als die *five-number-summary* nach Tuckey?

```
?fivenum
Tukey Five-Number Summaries
Description:
Returns Tukey's five number summary (minimum, lower-hinge,
median, upper-hinge, maximum) for the input data.
```

b) In welcher grafischen Darstellung spielen diese Werte eine große Rolle und was möchte man über die Daten aus dieser Darstellung ablesen?
Lösung: Boxplot, Schiefe/Symmetrie, evtl Lagevergleich bei parallelen Boxplots

Aufgabe 4: Es sollen drei Waschmittel auf Unterschiede in ihrer Waschkraft untersucht werden. Gemessen wird die Waschkraft als Anteil des reflektierten Lichts in Prozent nach einer Wäsche an der Reflektion des Lichtes an einem reinweissen Referenzstoffs. Die Verschmutzung wurde durch gleichmässige Einfärbung simuliert. Die Daten zu diesem Experiment finden Sie in der Datei `waschkraft.csv` im selben Verzeichnis wie die Daten aus Aufgabe 1.

a) Lesen Sie die Datei ein und speichern Sie die Daten in einer Variablen waschen.

```
waschen<- read.table(
  "http://fawn.hsu-hh.de/~steuer/Klausur/waschkraft.csv",
              header=TRUE,sep=",", strip.white=TRUE, dec=".")
```

b) Welches statistische Verfahren beantwortet Fragestellungen wie die vorliegende?

Lösung: ANOVA

c) Führen Sie die entsprechende Analyse durch. Interpretieren Sie das Ergebnis für die vorliegende Fragestellung.

```
str(waschen)
'data.frame': 60 obs. of  2 variables:
 $ Mittel    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Reflektion: num  48.2 48.4 49.9 47.5 46.4 ...
waschen[,1] <- as.factor(waschen[,1])

> summary(aov(Reflektion ~ Mittel, waschen))
            Df Sum Sq Mean Sq F value   Pr(>F)
Mittel       2 13.336  6.6679  5.8143 0.005035 **
Residuals   57 65.368  1.1468
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

d) Erzeugen Sie parallele Boxplots für die Reflektion der verschiedenen Waschmittel. Welches Mittel unterscheidet sich augenscheinlich von den anderen? **Lösung:**

```
boxplot(waschen$Reflektion ~ waschen$Mittel)
### Mittel 1.
```

e) Wie viele Paarvergleiche wären nötig, um mittels statistischer Tests herauszufinden, welche Waschmittelpaare unterschiedlich sind?
Lösung: $\binom{3}{2} = 3$
f) Welche Problematik tritt bei zahlreichen Paarvergleichen auf und wie kann man dem Problem entgegenwirken?
Lösung: Problematik des multiplen Testens, einfachste Gegenmaßnahme: Bonferroni.

Klassifikation (und Dimensionsreduktion)

- Klassifikation ist eine *der* Aufgaben der Statistik. Dies wird auch deutlich, wenn man sich vor Augen führt, dass eine der großen Statistikgesellschaften im deutschsprachigen Raum "Gesellschaft für Klassifikation" heißt.

- Die Aufgabe der Klassifikation ist eng verbunden mit dem Problem der Dimensionsreduktion hochdimensionaler Daten (curse of dimensionality) und dem Problem der Prognose.

- Was bedeutet Klassifikation?

- Ein Individuum (Beobachtung) soll einer Klasse (Attribut) zugeordnet werden. Die Klassenzugehörigkeit ist ein *nominales* Merkmal, es gibt insbesondere keine Ordnung der Attribute. Die wahre Klassenzugehörigkeit des Individuums ist nicht bekannt. Sie soll aus messbaren Eigenschaften des Individuums abgeleitet werden.

Beispiele für Klassifikationsaufgaben

- Ein einfaches Beispiel ist das allgegenwärtige Kreditscoring. Der Kunde X wird beschrieben durch eine große Anzahl von Attributen, z.B. Alter, Familienstand, Einkommen, Telefonrechnung, durchschnittliches Einkommen der Nachbarn etc. Ein Kreditscoringverfahren ordnet ihn entsprechend in eine der beiden Gruppen "kreditwürdig" oder "nicht kreditwürdig" ein.

- Die Anwendung von Klassifikation hat auch überraschende Ergebnisse gebracht. Bei der routinemäßigen Anwendung von Clusteranalysen auf eine Stichprobe aus einer Population von Flußkrebsen, lieferten die Verfahren der Statistik die ersten Hinweise darauf, dass es sich keineswegs um Individuen einer homogenen Gruppe handelte. Vielmehr waren zwei klar getrennte Cluster von Individuen in den Daten auszumachen. Anschließende Genom-Analysen bestätigten die Vermutung, dass es sich um zwei Spezies handelte.

Prinzipielles Vorgehen in der Klassifikation

- Wie wird eine solche Klassifikation praktisch durchgeführt? Wie ordnet sich die Theorie in die Statistik ein?

- Es muß eine statistische Entscheidungsregel hergeleitet werden, die ein Individuum auf Grund der beobachteten Eigenschaften (Messungen) einer der möglichen Klassen zuordnet.

- Eine solche Regel heißt *Klassifikationsregel*. Sie ist eine Abbildung von $\mathcal{R}^p \rightarrow \{C_1, C_2, \ldots, C_k\}$, bei p messbaren Attributen und k Klassen.

- Um eine solche Entscheidungsregel herzuleiten, benutzt man in der Regel eine Stichprobe, für deren Individuen die Klassenzugehörigkeit bekannt ist (Trainingsdaten). Aus den Zusammenhängen zwischen den gemessenen

Eigenschaften und den bekannten Klassenzugehörigkeiten sollen eine Klassifikationsregel hergeleitet werden.

- Hier wird lediglich der einfachste Fall betrachtet, nämlich eine Klassifikation in eine von zwei Klassen.

- Es gibt auch Verfahren, die versuchen, zunächst automatisch die Anzahl von Klassen in den Daten zu bestimmen. Hier wird der Zusammenhang zwischen Clusteranalyse und Klassifikation klar. Solche Verfahren werden in dieser Vorlesung nicht behandelt.

Formalia für die Diskussion der Klassifikation

- Im Folgenden haben die n Beobachungen $X_i, i = 1, \ldots, n$ messbare Attribute $x_i, i = 1, \ldots, p$ und eine Beobachung X_i wird durch den den Vektor $X_i = (x_{i1}, \ldots, x_{ip})$ beschrieben. Die Zuordnung soll in eine von zwei Klassen C_1, C_2 erfolgen.

- Ein Klassifikationsverfahren ist in diesem Falle also eine Abbildung $K : \mathcal{R}^p \rightarrow \{C_1, C_2\}$ die einem Individuum X_i die zugehörige Klasse $C(X_i)$ zuweist.

Kurze historische Einordnung

- Die Fragestellung der Klassifikation ist in der Statistik schon sehr lange präsent, spätestens seit Sir Fisher 1936 die Lineare Diskriminanzanalyse eingeführt hat.

- In den 90er Jahren des 20. Jhds. hat die Statistik die Deutungshoheit über diese Fragen (vorübergehend) an die Informatik verloren.

- All die Schlagworte *neuronale Netze*, *machine learning*, *supervised and unsupervised learning*, *data mining* behandeln im Prinzip das alte Klassifikations-Problem.

- Das Problem der Statistik war, dass ihre alten Methoden nicht mit den immens steigenden Beobachtungszahlen skalierten. (Matrizenmultiplikation!)

- Die Informatik hatte als einzige Wissenschaft das Handwerkszeug, um mit den Daten umzugehen (Datenbanken), aber überhaupt keine Theorie zur Datenanalyse. Viele Dinge wurden deshalb "neu erfunden".

- Seit ca. 10-15 Jahren wird "miteinander geredet"! Statistiker lernen mit Datenbanken umzugehen und Informatiker lernen die statistische Theorie.

Anforderungen an ein Klassifikationsverfahren

- Ein Klassifikationsverfahren soll "korrekt" sein, jedes Individuum soll in seine "korrekte" Klasse einsortiert werden. Wie kann man sinnvoll die Korrektheit messen?

- Da keine vollständig korrekte Klassifikation zu erwarten ist, ist die sogenannte Fehlklassifikationsrate

$$P(C(X) = C_1 \mid X \in C_2 \cup C(X) = C_2 \mid X \in C_1)$$

  einer Klassifikationsregel C eine vernünftige Maßzahl.

- Bei einer perfekten Klassifikationsregel ist die Fehlklassifikationsrate 0.

- Bei jedem Klassifikationsverfahren ist es das Ziel, eine möglichst geringe Fehlerrate zu erreichen.

Bestimmung der Fehlklassifikationsrate

- In der Regel, da ja die wahren, zugrundeliegenden Verteilungen der Daten unbekannt sind, läßt sich die Fehlerrate nicht explizit berechnen.

- Folglich wird durch einfachen Abzählen auf den Trainingsdaten versuchen, die Fehlklassifikationrate zu optimieren. Allerdings sollte diese Rate, wenn möglich, nicht nur auf dem Trainingsset überprüft werden. Das eigentliche Problem ist ja die Prognose für in der Zukunft zu beobachtende Individuen.

- In der Regel teilt man deshalb den vorhandenen Datensatz in einen Trainings- und einen Testdatensatz auf. Die Fehlklassifikationsrate wird dann auf dem Testdatensatz bestimmt. Dabei dürfen keine Informationen aus dem Testdatensatz zur Konstruktion der Klassifikationsregel herangezogen werden.

Einschub: Multivariate Normalverteilung

- Bekannt ist die (Dichte der) Normalverteilung mit Parametern μ und σ^2.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2)}{2\sigma^2} \right\}.$$

- Die Dichte der multivariaten Normalverteilung in $\mathcal{R}^d$ lautet wie folgt:

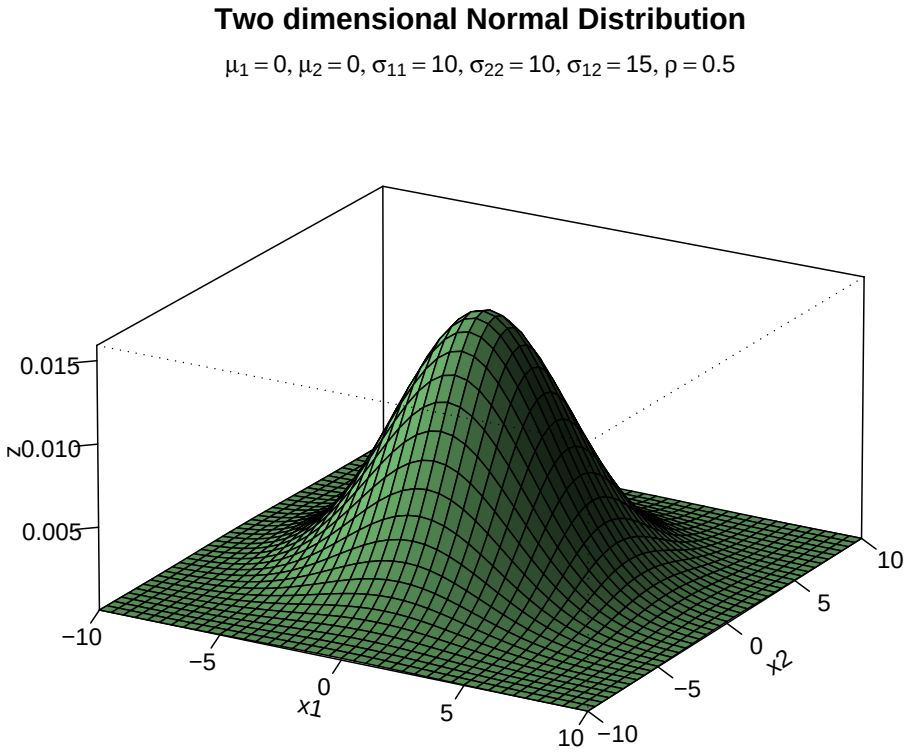
$$f(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \right\}.$$

- Dabei ist μ der d-dimensionale Mittelwert (Schwerpunkt) und Σ die dxd Kovarianzmatrix der Verteilung. Besitzt Σ nur identische Einträge auf der Hauptdiagonalen, so ist die resultierende Dichte rotationssymmetrisch, ansonsten haben die Isolinien der Dichtefunktion die Form eine Ellipse.

- Geschätzt werden μ und Σ durch das empirische Mittel $\bar{X}$ bzw. die empirische Kovarianzmatrix $Cov(X)$.

- In `R` haben Sie Zugriff auf die üblichen Funktionen für Verteilungsfunktionen, wenn Sie das Paket `mvtnorm` installieren. (`rmvnorm`, `dmvnorm`, `pmvnorm` etc.)

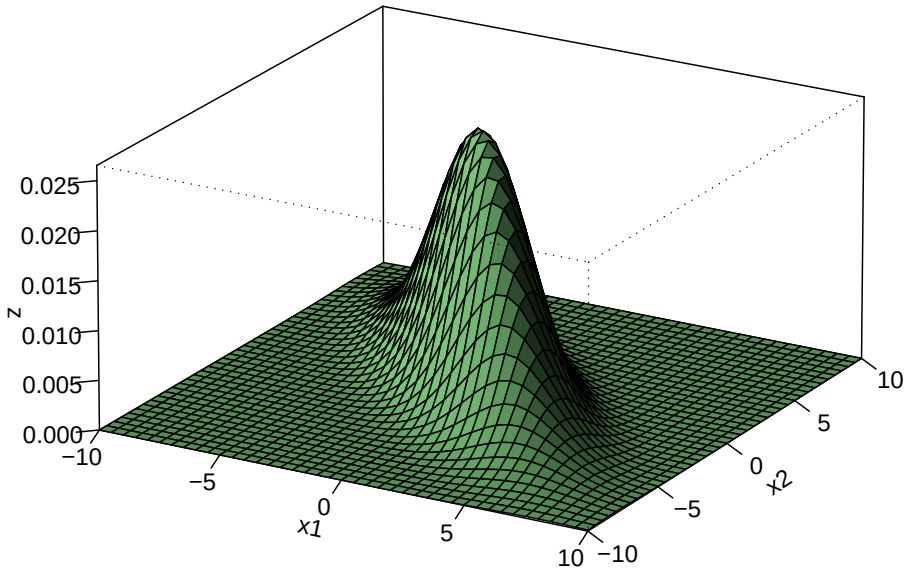
Unkorrelierter Fall



Korrelierter Fall

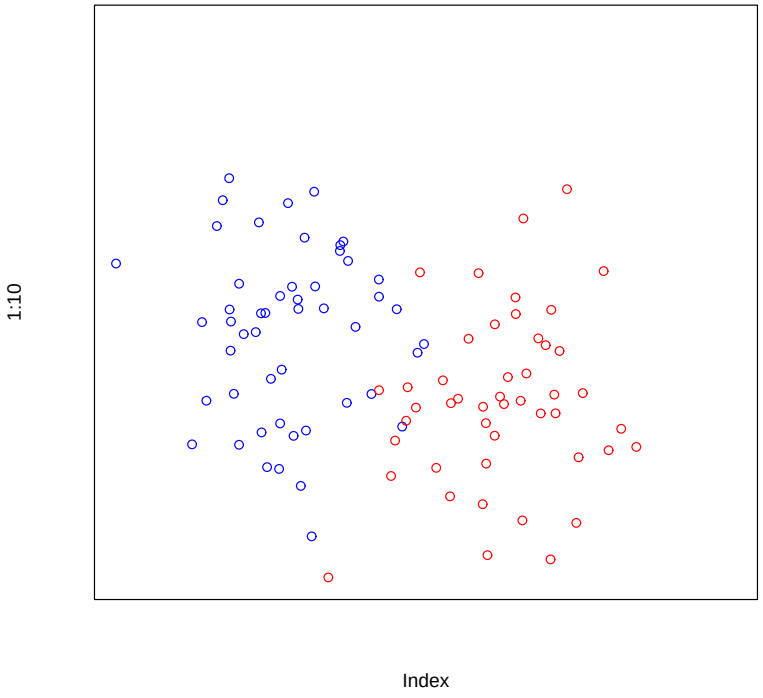
Two dimensional Normal Distribution

$\mu_1 = 0, \mu_2 = 0, \sigma_{11} = 10, \sigma_{22} = 10, \sigma_{12} = 15, \rho = 0.5$



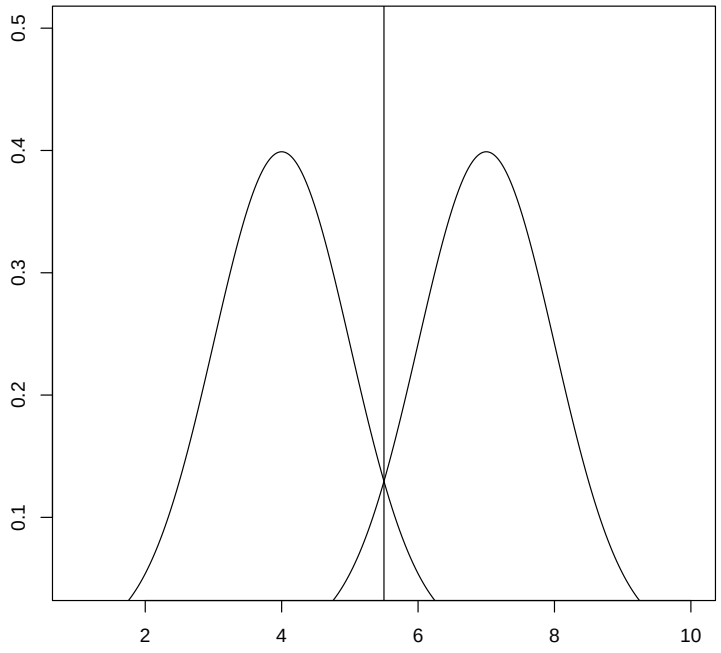
Lineare Diskriminanzanalyse: Ausgangssituation

- Gegeben ist ein Trainingsdatensatz von Individuen, deren Zugehörigkeit zu einer von zwei Gruppen jeweils bekannt ist.



Idee Diskriminanzanalyse (LDA) von Fisher

- Reduktion auf den univariaten Fall. Das Bild veranschaulicht die Situation für univariate Klassifikation.



Idee der Fischerschen Diskriminanzanalyse

- Hat man univariat beobachtete Individuen, also nur ein Merkmal pro Beobachtung, bei denen die Messungen des Merkmals innerhalb der verschiedenen Gruppen mit derselben Varianz σ^2 gestreut vorliegen, dann erhält man eine Klassifikationsregel mit minimaler Fehlerrate, wenn man links von $\frac{\mu_1+\mu_2}{2}$ die Beobachtungen der Klasse 1 und rechts davon der Klasse 2 zuschlägt.

- "Der nächste Mittelwert gewinnt".

Übertragung in den multivariaten Fall

- Suche ein Richtung im Raum, in der die Gruppen "maximal getrennt" sind.

- Fishers geniale Idee: Finde eine optimale Linearkombination $< W, X >$, mit $W \in \mathcal{R}^p \backslash \{0\}$, um das einfache Verfahren aus dem univariaten Fall anzuwenden.

- Wenn $\mathsf{E}(X) = \mu_i$ und $\mathsf{Cov}(X) = \Sigma$ für $X \in C_i,\ i = 1, 2$ gelten, dann gilt für eine Linearkombination $< W, X >$

$$\mathsf{E}(< W, X >) =< W, \mu_i > \text{ wobei i die Klasse von } X$$

und

$$\mathsf{Cov}(< W, X >) = W^T \Sigma W.$$

- Die Diskriminanzanalyse ist also auch ein Verfahren zur Dimensionsreduktion! Die Daten werden aus dem $\mathcal{R}^p$ in den Raum $\mathcal{R}$ herunterprojeziert.

- Die Modellannahme in der klassischen Diskriminanzanalyse ist, dass die Individuen der Klasse 1 mutivariat normalverteilt gemäß $N(\mu_1, \Sigma)$ und die Individuen der Klasse 2 gemäß $N(\mu_2, \Sigma)$ mit identischer Kovarianzmatrix Σ. (Homoskedastizität)

- Gibt man die Voraussetzung der linearen Kombination auf und erlaubt allgemeinere Ansätze zur Bestimmung einer trennenden Funktion, gelangt man beipielsweise zur QDA.

Optimalitätskriterien in der Diskriminanzanalyse

- Leider liefert jede Richtung im Raum $W \in \mathcal{R}^p$ eine Lösung, die dem eindimensionalen Fall entspricht.
- Wie kann man zwischen diesen Richtungen differenzieren?
- Fishers Idee: Wähle die Linearkombination so, dass die Klassen maximal getrennt sind.
- Formalisiert bedeutete dies: Minimiere die Varianz innerhalb der einzelnen Klassen und maximiere die Varianz zwischen den Klassen!

Formale Lösung der Diskriminanzanalyse

- Mit den Bezeichnungen

$$s^2_{between} := (< W, \mu_1 > - < W, \mu_2 >)^2 \text{ und}$$

$$s^2_{within} := 2W^T \Sigma W$$

soll der Quotient

$$S := \frac{s^2_{between}}{s^2_{within}}$$

über die Wahl des Vektors W maximiert werden. Da diese Aufgabe mit W auch für jedes λW gelöst wird, muss man noch die Nebenbedingung einführen, dass $||W|| = 1$.

- Dieses Optimierungsproblem ist analytisch lösbar und zwar löst

$$W_{max} = \frac{1}{2}\Sigma^{-1}(\mu_1 - \mu_0)$$

das Optimierungsproblem. W_{max} heißt erste *Diskriminante*.

- Insgesamt wird also das eindimensionale Klassifikationsproblem mit den Daten $\{< W_{max}, X_1 >, \ldots, < W_{max}, X_n >\}$ gelöst.

- Um eine neue Beobachtung X_{n+1} zu klassifizieren würde nunmehr die transformierte Größe $Y :=< W_{max}, X_{n+1} >$ betrachtet und geschaut, ob dieser Wert näher am transformierten Mittel der ersten oder der zweiten Klasse liegt.

- Die Diskriminanzanalyse liefert also sowohl eine Klassifikationsregel, als auch eine Dimensionsreduktion von Dimension p auf Dimension 1!

Diskussion der Vorraussetzungen der Diskriminanzanalyse

- Die strengen Voraussetzungen in der Herleitung sind nur der möglichst einfachen Vermittlung der Idee zu schulden.

- Diskriminanzanalyse ist auch im Falle von Heteroskedastizität zwischen den Klassen (verschiedene Σ_i) oder wenn im Trainingsdatensatz unterschiedlich starke Besetzungen der Klassen vorliegen gut anzuwenden.

- Die Diskriminanzanalyse wird auch für Probleme mit mehr als zwei Klassen angewendet und läßt sich analog formulieren.

Anwendung der LDA in R

- Die Funktion `lda` findet sich im Paket `MASS`, welches bei einer Standard-R-Installation vorhanden ist.

- Die Daten der Grafik, die die Ausgangssituation der LDA veranschaulichensollte, wurden z.B. wie folgt erzeugt:

```
> library(mvtrnorm)
> library(MASS)
> set1 <- rmvnorm(50, mean=c(3.5,4.5), sigma=diag(c(1,2)))
> set2 <- rmvnorm(50, mean=c(6.5,3.5), sigma=diag(c(1,2)))
> known <- c(rep("class1", 50), rep("class2",50))
> punkte <- rbind(set1, set2)
```

- Es liegen also jeweils 50 Beobachtungen jeder Klasse mit bekannter Klassifikation vor.

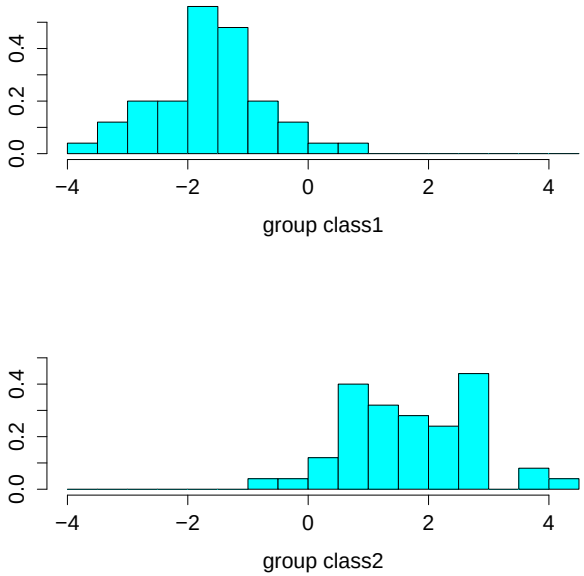
- Die Funktion `lda()` erwartet als erstes Argument einen *dataframe* oder eine Matrix mit den Beobachtungen und als zweites Argument den Vektor, der die bekannten Klassenzugehörigkeiten angibt.

```
> ?lda ; lda(punkte, known)
Prior probabilities of groups:
class1 class2
   0.5    0.5
Group means:
              1        2
class1 3.322418 4.285684
class2 6.401199 3.834064
Coefficients of linear discriminants:
             LD1
[1,] 1.070120078
[2,] 0.001494093
```

Der LDA-Plot

- Zu `lda()` existiert eine eigene Plotmethode, die sehr schön die ursprüngliche Idee von Fisher widerspiegelt:

```
> plot(lda(punkte, known))
```



Bestimmung der Fehlklassifikationsrate

- Die Funktion `predict` arbeitet auch für Objekte vom Typ `lda`.

```
> diskr <- lda(punkte, known)
> predict(diskr)
...
> > table(predict(diskr)$class, known)
        known
         class1 class2
  class1     48      2
  class2      2     48
```

- Also eine Fehlklassifikationsrate von 4%! (Allerdings auf dem Trainigsset)

Aufgabe zur LDA

- Führen Sie für den Iris-Datensatz für die verschiedenen Kombinationen von Spezies jeweils eine LDA durch. Bestimmen Sie die Diskriminanten und die Fehlklassifikationsraten.

- Wiederholen Sie die LDA indem Sie jeweils 80% der Daten in den Trainingsset nehmen und die Fehlklassifikation auf den übrigen 20% als Testset bestimmen!

Besprechung der Aufgabe zur LDA

- Führen Sie für den Iris-Datensatz für die verschiedenen Kombinationen von Spezies jeweils eine LDA durch. Bestimmen Sie die Diskriminanten und die Fehlklassifikationsraten.

- Wiederholen Sie die LDA, indem Sie jeweils 80% der Daten in einen Trainingsset nehmen und die Fehlklassifikation auf dem Testset der übrigen 20% bestimmen!

- In der Vorlesung war bisher nur die LDA zur Trennung von zwei Gruppen behandelt worden. Im Anschluss an die Übungsaufgabe wird die Theorie der LDA auf endlich viele Gruppen mit bekannten Gruppenzugehörigkeiten erweitert.

Vorüberlegungen

- Definition der Fehlklassifikationsrate F_D eines Datensatzes D :

$$F_D := \frac{\text{Anzahl falsch klassifizierter Datenpunkte aus } D}{\text{Anzahl der Beobachtungen in } D}.$$

- Welche LDA sind eigentlich durchzuführen?

- Zunächst die Vorarbeiten (Laden der Daten, laden der nötigen Bibliothek):

```
> data(iris)
> library(MASS)
> attach(iris)
```

Eigentliche Lösung

- Welche Kombinationen von `Species` sind anzuschauen?

```
levels(Species)
[1] "setosa"     "versicolor" "virginica"
```

- Es gibt also drei mögliche Kombinationen:
  - `setosa` vs. `versicolor`,
  - `setosa` vs. `virginica`,
  - `versicolor` vs. `virginica`.

- Wie wählt man die jeweils passende Teilmenge in `R` aus?

Passende Teilmengen in R auswählen

- Entweder spezifiziert man vollständig die Datenpunkte, die aufgenommen werden und legt diese in einer Variablen ab, um Tipparbeit zu sparen, etwa

```
included <- which( (Species == "setosa")  |
                       (Species == "versicolor"))
```

- Oder, weil es in diesem Fall weniger Tipparbeit ist, spezifiziert man die Datenpunkte, die jeweils nicht in die Analyse eingehen:

```
excluded <- which(Species=="virginica")
```

Berechnung der LDA

- In jedem Fall kann nun die erste LDA gerechnet werden. Wieder wird das Ergebnis in einer Variablen abgelegt:

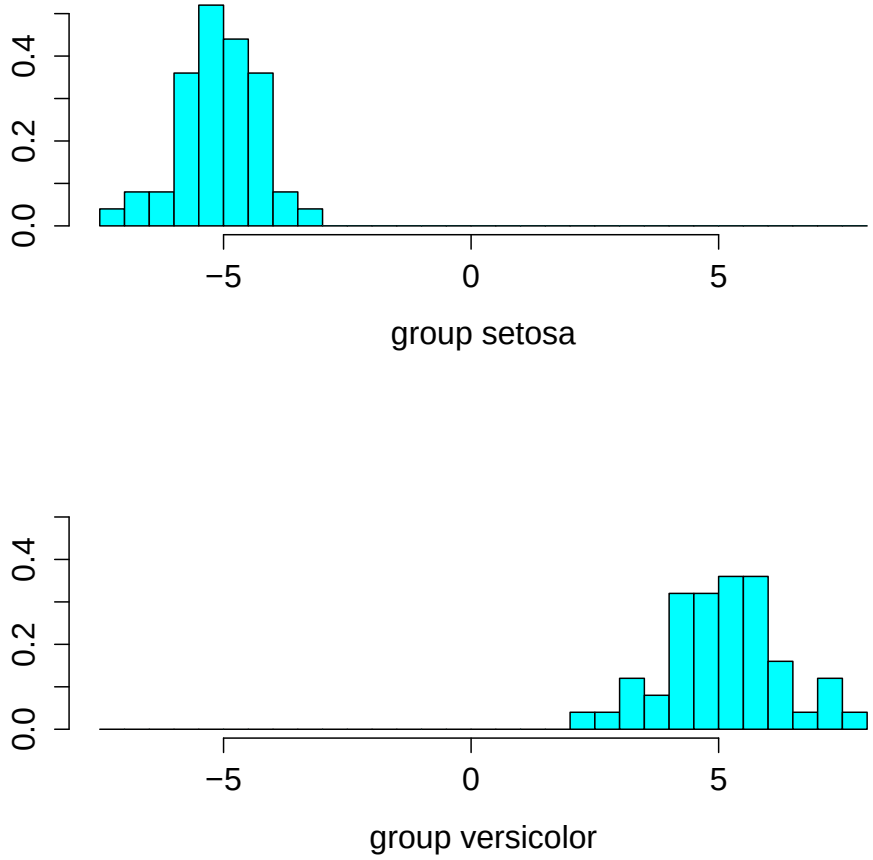
```
ld1 <- lda(iris[-excluded,1:4],iris[-excluded,5])
Warning message:
In lda.default(x, grouping, ...) : group virginica is empty
```

- Was bedeutet die Warnung? In `iris` hat die Variable `Species` 3 Faktorstufen, in den Daten, die in die Analyse eingehen, sind jedoch nur noch zwei davon vorhanden.

- Wie sieht das Ergebnis aus?

```
plot(ld1)
```

LDA Plot setosa und versicolor



Ablesen von Fehlklassifikationsrate und erster Diskriminante

- Die Fehlklassifikationsrate ist, wie in der Grafik leicht zu erkennen 0!
- Außerdem sollte die (erste) Diskriminante bestimmt werden.

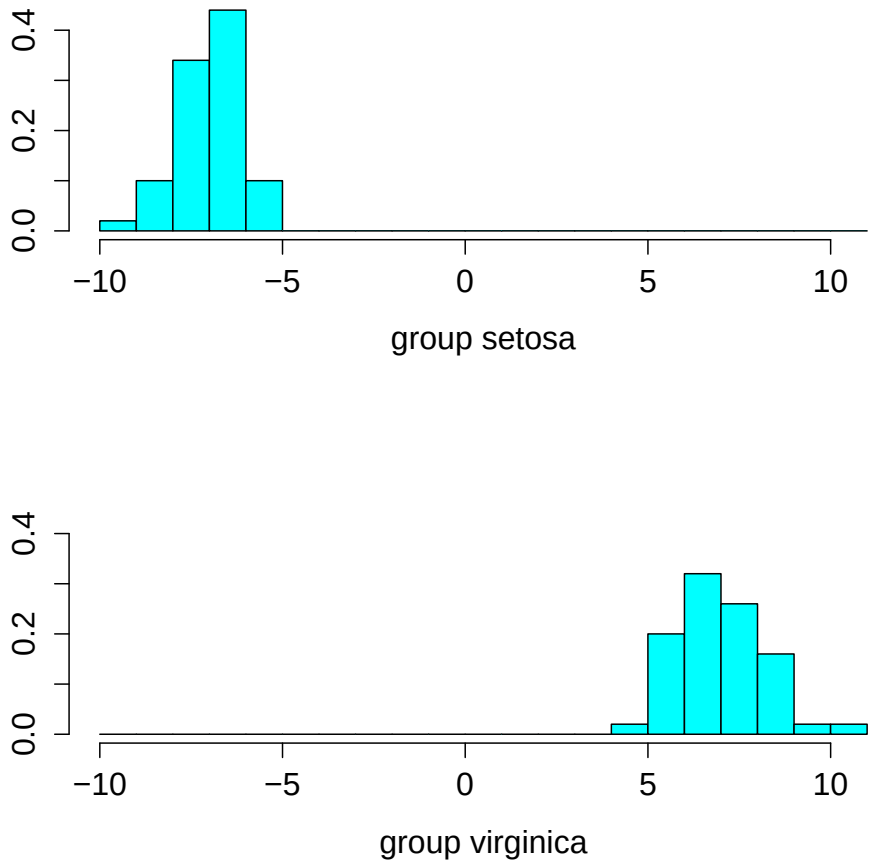
```
> ld1
...
Coefficients of linear discriminants:
                    LD1
Sepal.Length -0.3004580
Sepal.Width  -1.7738451
Petal.Length  2.1422596
Petal.Width   3.0357262
```

Die anderen Kombinationen

- Hat man eine LDA berechnet, so kann man für die anderen Fälle analog vorgehen.

```
> excluded<-which(Species=="versicolor")
> ld1 <- lda(iris[-excluded,1:4],iris[-excluded,5])
> plot(ld1)
```

LDA Plot für `setosa` vs. `virginica`



Fehlklassifikationsrate und Diskriminante

- Auch hier ist eine Fehlklassifikationsrate von 0 an der Grafik erkennbar.

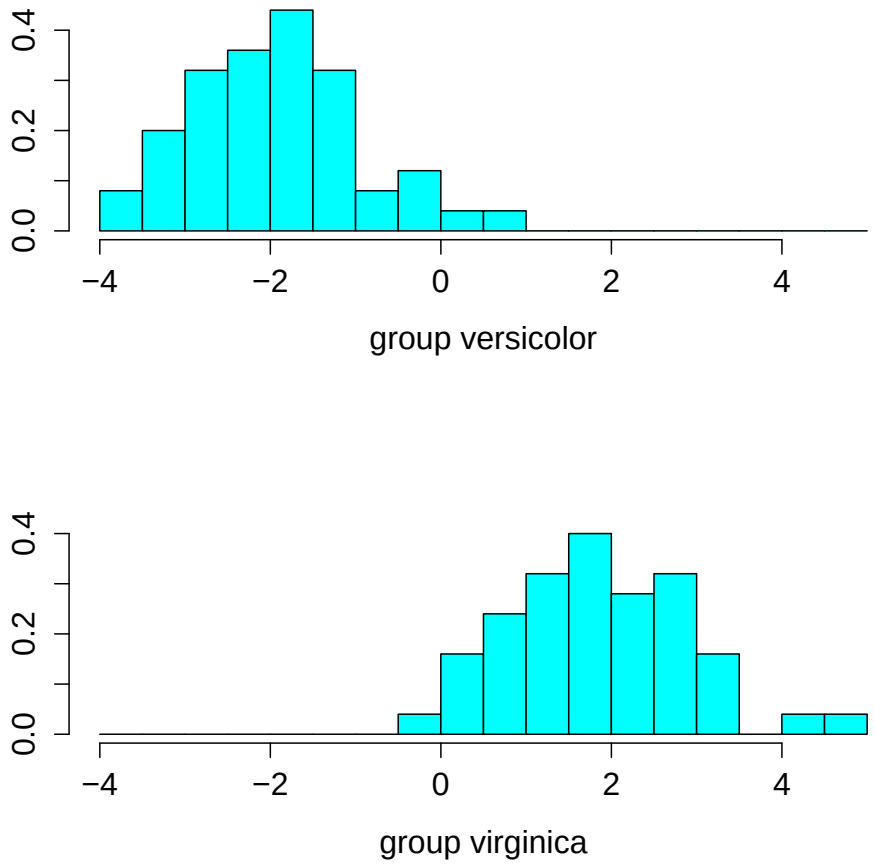
```
> ld1
...
Coefficients of linear discriminants:
                    LD1
Sepal.Length -1.1338828
Sepal.Width  -0.8603685
Petal.Length  2.6138926
Petal.Width   2.6310427
```

Kombination ohne `setosa`

- Vorgehen wieder entsprechend dem schon gesehenen.

```
> excluded<-which(Species=="setosa")
> ld1 <- lda(iris[-excluded,1:4],iris[-excluded,5])
> plot(ld1)
```

LDA Plot für `virginica` vs. `versicolor`



Fehlklassifikationsrate für die Kombination ohne `setosa`

- Hier ist die Fehlklassifikationsrate nicht 0!
- Wie kann man in `R` die Fehlklassifikation berechnen?
- Die Funktion `predict` kann auch für Ergebnisse der Diskriminanzanalyse Vorhersagen berechnen. Eine Fehlklassifikation ist äquivalent zu einer falschen Vorhersage der Klassenzugehörigkeit.
- Beinhaltet die Variable `ld1` das Ergebnis einer LDA, so erhält man die Prognosen der Klassenzugehörigkeiten der Beobachtungspunkte über den Befehl:

```
> predict(ld1)$class
```

Fehlklassifikationsrate für die Kombination ohne `setosa` II

- Die Anzahl, der falsch klassifizierten Beobachtungen liefert dann das folgende `R` Kommando:

```
> length( which( ! predict(ld1)$class == iris[-excluded,5] ) )
[1] 3
```

- Die Fehlklassifikationsrate liegt also bei 3%.

- Achtung! Die Fehlklassifikationsrate auf dem Trainigsdatensatz unterschätzt die in der Praxis zu erwartende Fehlklassifikation stets! Da die Schätzer optimal für die Daten im Trainingsdatensatz berechnet werden, ist die Fehlerrate auf einem Testdatensatz stets größer oder gleich der Fehlerrate auf den Trainingsdaten!

Die erste Diskriminante der Kombination ohne `setosa`

```
> ld1
....
Coefficients of linear discriminants:
                    LD1
Sepal.Length -0.9431178
Sepal.Width  -1.4794287
Petal.Length  1.8484510
Petal.Width   3.2847304
```

Aufteilung in Trainings- und Testdatensatz

- `R` stellt den Befehl `sample` zur Verfügung, um aus einem gegebenen Vektor eine Stichprobe einer bestimmten Länge zu ziehen.

- Nutzung sehr einfach:

```
> sample(x, size, replace = FALSE, prob = NULL)
```

- Jede unserer Faktorkombinationen hat 100 Beobachungen, die Aufteilung soll im Verhältnis 80:20 erfolgen.

```
> trainingsset <- sample(1:100,80)
> controlset <- seq(1:100)[-trainingsset]
```

- Welche Änderungen an den Ergebnissen kann man erwarten?

Kombination ohne `virginica`

```
> trainingsset <- sample(1:100,80)
> testset <- seq(1:100)[-trainingsset]
> excluded<-which(Species=="virginica")
> ld1 <- lda(iris[-excluded,1:4][trainingsset,],iris[-excluded,5][trainingsset])
#Warning message:
#In lda.default(x, grouping, ...) : group virginica is empty
> plot(ld1)
> length(which(!
  predict(ld1, iris[-excluded,1:4][trainingsset,] )$class ==
                          iris[-excluded,5][trainingsset] ))
> length(which(!
  predict(ld1, iris[-excluded,1:4][testset,] )$class ==
                            iris[-excluded,5][testset] ))
```

Kombination ohne `versicolor`

```
> trainingsset <- sample(1:100,80)
> testset <- seq(1:100)[-trainingsset]
> excluded<-which(Species=="versicolor")
> ld1 <- lda(iris[-excluded,1:4][trainingsset,],iris[-excluded,5][trainingsset])
#Warning message:
#In lda.default(x, grouping, ...) : group virginicaersicolor is empty
> plot(ld1)
> length(which(!
    predict(ld1, iris[-excluded,1:4][trainingsset,] )$class ==
                          iris[-excluded,5][trainingsset] ))
> length(which(!
    predict(ld1, iris[-excluded,1:4][testset,] )$class ==
                            iris[-excluded,5][testset] ))
```

Kombination ohne setosa

```
> trainingsset <- sample(1:100,80)
> testset <- seq(1:100)[-trainingsset]
> excluded<-which(Species=="setosa")
> ld1 <- lda(iris[-excluded,1:4][trainingsset,],iris[-excluded,5][trainingsset])
#Warning message:
#In lda.default(x, grouping, ...) : group setosa is empty
> plot(ld1)
> length(which(!
    predict(ld1, iris[-excluded,1:4][trainingsset,] )$class ==
                 iris[-excluded,5][trainingsset] ))
[1] 3

> length(which(!
    predict(ld1, iris[-excluded,1:4][testset,] )$class ==
                 iris[-excluded,5][testset] ))
[1] 0
```

Was sagt die beobachtete Fehlklassifikationsrate aus?

- Natürlich ist dies aber nur **eine** der möglichen Stichproben vom Umfang 80. Insgesamt gibt es etwa $\binom{100}{80}$ mögliche Stichproben mit vielen möglichen Fehlklassifikationsraten.

- Wie bekommt man nun einen Schätzer für die Klassifikationsleistung einer LDA bei einer zufälligen Aufteilung in Trainings- und Testdatensätze?

- Stichwort: Resampling! Man führt das Experiment im Rechner einfach sehr häufig durch, in dem man immer wieder neue Stichproben aus der vorhandenen Datenbasis zieht und ermittelt aus diesen Experimenten die durchschnittliche Fehlklassifikationsrate!

Resampling der Fehlklassifikationsrate in R

```
> tlt<-0 ; tlc<-0
> for ( i in 1:100){
 cat(i, "\n")
 trainingsset <- sample(1:100,80)
 testset <- seq(1:100)[-trainingsset]
 excluded<-which(Species=="setosa")
 ld1 <- lda(iris[-excluded,1:4][trainingsset,],iris[-excluded,5][trainingsset])
 tlt <- tlt +  length(which(!
     predict(ld1, iris[-excluded,1:4][trainingsset,] )$class ==
                  iris[-excluded,5][trainingsset] ))
  tlc <- tlc + length(which(!
     predict(ld1, iris[-excluded,1:4][testset,] )$class ==
                  iris[-excluded,5][testset] ))
  }
> tlt/8000 ; tlc/2000
[1] 0.026375 0.04
```

Bedeutung des Umfangs der Trainingsdatensatzes

- Was passiert, wenn der Umfang des Trainingsdatensatzes sinkt?
- Die Fehlklassifikationsrate sollte ansteigen, da weniger Informationen für das Training ("unsupervised learning") zur Verfügung stehen.
- Als Experiment werden im Beispiel die Daten diesmal im Verhältnis 20:80 aufgeteilt.

Umsetzung in R

```
> tlt<-0 ; tlc<-0
> for ( i in 1:100){
  cat(i, "\n")
  trainingsset <- sample(1:100,20)
  controlset <- seq(1:100)[-trainingsset]
  excluded<-which(Species=="setosa")
  ld1 <- lda(iris[-excluded,1:4][trainingsset,],
             iris[-excluded,5][trainingsset])
  tlt <- tlt + length(which(!
    predict(ld1, iris[-excluded,1:4][trainingsset,] )$class ==
                 iris[-excluded,5][trainingsset] ))
  tlc <- tlc + length(which(!
    predict(ld1, iris[-excluded,1:4][controlset,] )$class ==
                 iris[-excluded,5][controlset] ))
  }
> tlt/2000; tlc/8000
[1] 0.015 0.05925
```

LDA in der allgemeinen Formulierung

- Die Beschränkung auf zwei Klassen war nur nötig, um die Idee der LDA möglichst anschaulich zu machen. Diese Beschränkung hat keine Begründung im Verfahren selbst.

- Die Optimierungsaufgabe, den Quotienten

$$\max S := \frac{s^2_{between}}{s^2_{within}}$$

  zu maximieren, kann ebenso mit $N \geq 2$ Klassen gestellt und gelöst werden!

- Ist $N > 2$, so können ebenso, analog zum Vorgehen in der Hauptkomponentenanalyse, weitere Diskriminanten bestimmt werden.

- Gegeben eine erste Diskriminante W_1, muss dann die zweite Diskriminante W_2 die Optimierung unter Nebenbedingungen lösen

$$\max S := \frac{s^2_{between}}{s^2_{within}} \text{ unter } W_2' W_1 = 0.$$

- Analog wird für die höheren Diskriminanten $W_i, i = 3, \ldots, N-1$ vorgegangen, welche jeweils senkrecht auf allen vorhergehenden Diskriminanten stehen.

- Die Mathematik übernimmt hier die Software, so dass man sich auf die Interpretation konzentrieren kann, wenn man die Grundidee verstanden hat!

- Deshalb wird hier auf die mathematische Herleitung verzichtet!

Die allgemeine LDA in R

```
> lda(iris[,1:4], iris[,5])
Prior probabilities of groups:
...
Group means:
...

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

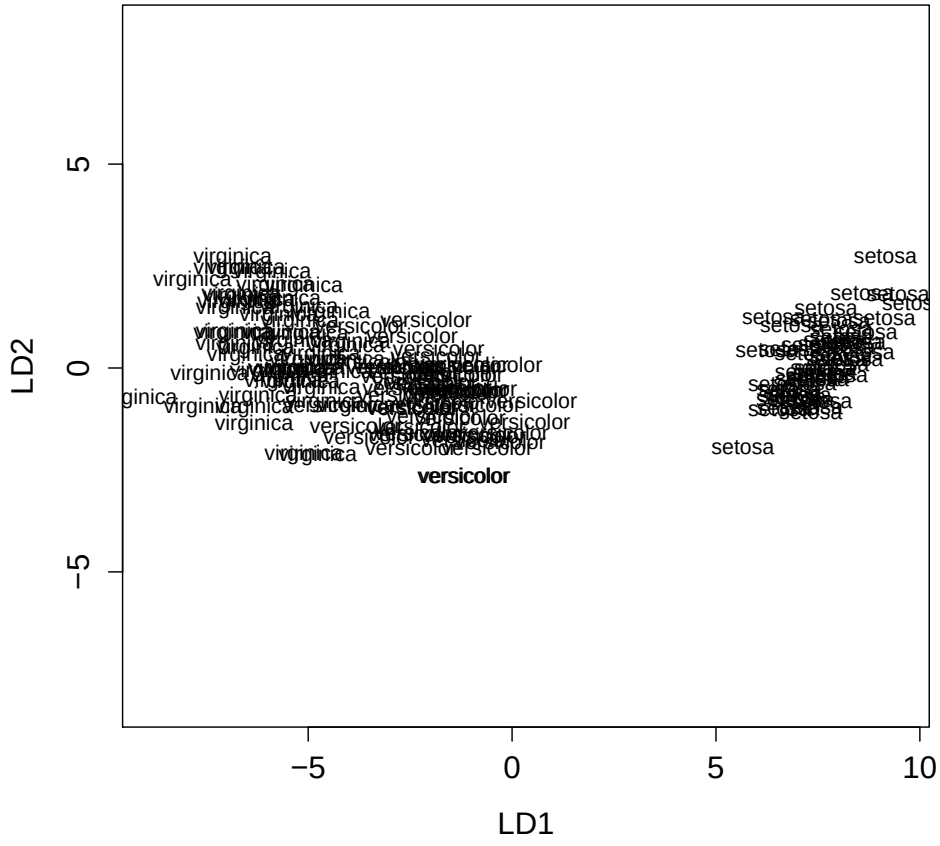
Interpretation der allg. LDA in R

- Der *proportion of trace* hat die Rolle, die der Varianzanteil bei der Hauptkomponentenanalyse spielt. Es wird der Anteil der erklärten Varianz angegeben.

- Auch für diesen Fall gibt es wieder einen Standardplot, der dem Biplot der Hauptkomponentenanalyse entspricht. Der Standardplot ist in der Methode für plot für Daten der Klasse LDA implementiert.

```
> plot(lda(iris[,1:4], iris[,5]))
```

Scatterplot der ersten beiden Diskriminanten



Aufgabe zur LDA

Sie finden auf der Webseite der Veranstaltung unter `Daten zur Vorlesung` einen Datensatz `possum.csv`.

a) Lesen Sie diesen Datensatz ein. Notieren den Befehl, um den Inhalt der Datei in einer Variablen abzulegen. Beachten Sie die Struktur der Datei mit der Kopfzeile und dem Leerzeichen als Trennzeichen!

b) Führen Sie mit den Daten eine LDA durch, die die Variablen `hdlngth`, `skullw`, `totlngth`, `taill`, `footlgth`, `earconch`, `eye`, `chest` und `belly` einbezieht, um die bekannte Klassenzugehörigkeit aus der Klasse `site` zu erklären. Notieren Sie den notwendigen Befehl.

c) Wie viele der erhaltenen Diskriminaten erscheinen Ihnen wichtig und wieso?

Hauptkomponentenanalyse - Motivation

- Oft wird das Verfahren einfach mit der englischen Abkürzung bezeichnet: PCA - *principal component analysis*.

- Literatur z. B. Andreas Handl, Multivariate Analysemethoden.

- Die ursprüngliche Aufgabenstellung der PCA war die Anordnung multivariater Daten. Oft werden pro Beobachtung mehrere (hier p) metrische Merkmale erhoben, beispielsweise Klausurnoten je Studierendem. Im folgenden liegen die Daten stets in einer (nxp) - Datenmatrix $\mathbf{X}$ vor.

- Wie vergleicht man nun sinnvoll mehrere Studierende?

- Erforderlich ist eine Dimensionsreduktion, denn der $\mathcal{R}^p$ kann nicht angeordnet werden, nur in $\mathcal{R}$ existiert eine totale Ordnung.

- Angenommen es werden je Studierendem p Noten $x_i, i = 1, \ldots, p$ erfasst, so kann man zum arithmetischen Mittel $\sum_1^p \frac{x_i}{p}$ übergehen (numerus clausus).

- Dieses Mittel ist auch als Linearkombination $a'X$ mit $a' = (\frac{1}{p}, \ldots, \frac{1}{p})$ darstellbar.

- Eine solche Linearkombination führt als Abbildung die gewünschte Dimensionsreduktion von $\mathcal{R}^p$ nach $\mathcal{R}$ durch. In $\mathcal{R}$ können die Beobachtungen nunmehr sinnvoll angeordnet werden.

- Weiterhin wird eine neue Variable eingeführt, die Durchschnittsnote, die an die Stelle der Originaldaten tritt, um die Beobachtung unidimensional zu charakterisieren. Eine solche Variable heißt auch latente Variable, da sie im Originaldatensatz nur verdeckt auftritt.

- Etwas Entsprechendendes wird nun von der Hauptkomponentenanalyse erreicht, nämlich eine Dimensionsreduktion auf $k < p$ Dimensionen durch Bildung bestimmter Linearkombinationen der Originalvariablen unter bestimmten Optimalitätsbedingungen.

Exkurs: Matrizenrechnung

- Um die PCA zu verstehen, benötigt man die Begriffe *Eigenwert* und *Eigenvektor*. Die Begriffe werden lediglich informell eingeführt.

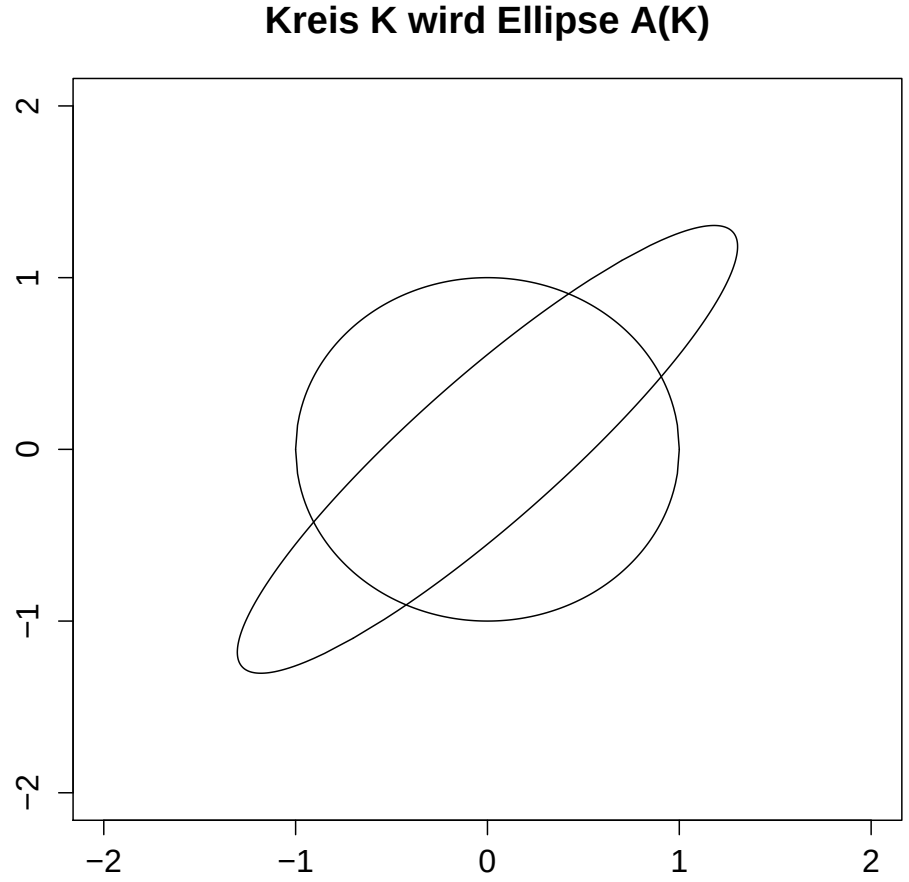
- Matrizen sind lineare Abbildungen.

- Besonders interessant sind die Abbildungen vom $\mathcal{R}^p$ in den $\mathcal{R}^p$, die repräsentiert werden durch die quadratischen pxp Matrizen, noch interessanter die symmetrischen Matrizen in dieser Gruppe (Kovarianzmatrizen!).

- Eine lineare Abbildung A bildet in diesem Fall beispielsweise einen Kreis K in eine Ellipse $A(K)$ ab.

- An diesem Übergang Kreis-Ellipse kann man die geometrische Bedeutung der Begriffe Eigenvektor und Eigenwert schön sehen.

Veranschaulichung Eigenwert/Eigenvektor



Exkurs: Eigenwerte und -vektoren

- Für symmetrische Abbildungen A geben die Eigenvektoren die Richtungen der Hauptachsen der Ellipse an, die Wurzeln der Eigenwerte die Streckungsfaktoren des Kreisradius in die Richtungen der Eigenvektoren.

- In R:

```
abbildung <- matrix(c(0.7, 1.1, 1.1, 0.7), nrow=2)
eigen(abbildung)
$values
[1]  1.8 -0.4
$vectors
           [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

Exkurs: Eigenwerte und -vektoren

- Sind nun $v_1, \ldots, v_p$ die Eigenvektoren einer Abbildung $A \in \mathcal{R}^{p \mathsf{X} p}$, und $\lambda_1, \ldots, \lambda_p$ die zugehörigen Eigenwerte, so gilt:

$$Av_i = \lambda_i v_i, \;\; i = 1, \ldots, p.$$

- Damit ist auch klar, dass sich die Eigenvektoren nur bis auf einen Proportionalitätsfaktor bestimmen lassen. Eigenvektoren werden deshalb normiert auf Länge 1 angegeben.

- Die (numerische) Mathematik zur Eigenwertberechnung ist nicht trivial und wird hier weiter nicht behandelt.

Idee der Hauptkomponentenanalyse

- Bei der Hauptkomponentenanalyse wird die Basis des $\mathcal{R}^p$, die aus den einzelnen Koordinatenachsen besteht, durch eine neue Orthogonalbasis ersetzt. Um inhaltlich interpretierbare Ergebnisse zu erhalten, müssen die Daten zunächst koordinatenweise mittelwertbereinigt werden. Es gilt also

$$\sum_{1}^{n} X_i = \mathbf{0} \text{ wobei } X_i \text{ die i-te Zeile der Datenmatrix ist.}$$

- Jeder neue Basisvektor der von dem Verfahren erzeugt wird ist eine Linearkombination der Originalkoordinaten und heißt auch Hauptkomponente.

Idee der Hauptkomponentenanalyse II

- Der erste Basisvektor, also die erste Hauptkomponente, wird so gewählt, dass sie unter allen Linearkombinationen der Originaldaten die maximale Varianz besitzt.

- Nachfolgend berechnete Basisvektoren stehen jeweils senkrecht auf allen bisher gewählten Vektoren und erklären den maximalen Anteil der verbliebenen Varianz.

- Eine Dimensionsreduktion wird nun immer dann möglich, wenn die ersten $k < p$ Hauptkomponenten bereits einen großen Teil der Gesamtvarianz erklären (z.B. mehr als 90%).

Idee der Hauptkomponentenanalyse III

- Entwickelt wurde die Hauptkomponentenanalyse bereits 1901 von K. Pearson. Heute wird sie meist als exploratives Tool genutzt, um entweder sogenannte latente Variablen zu entdecken oder um Vorhersagemodelle aufzustellen.

- Es lassen sich z.B. Daten auf die ersten beiden Hauptkomponenten projizieren und in einem Scatterplot darstellen (s. Beispiel *Biplot*).

- Oft sind die Hauptkomponenten inhaltlich interpretierbar.

Lösung zur LDA Aufgabe `possumcsv`

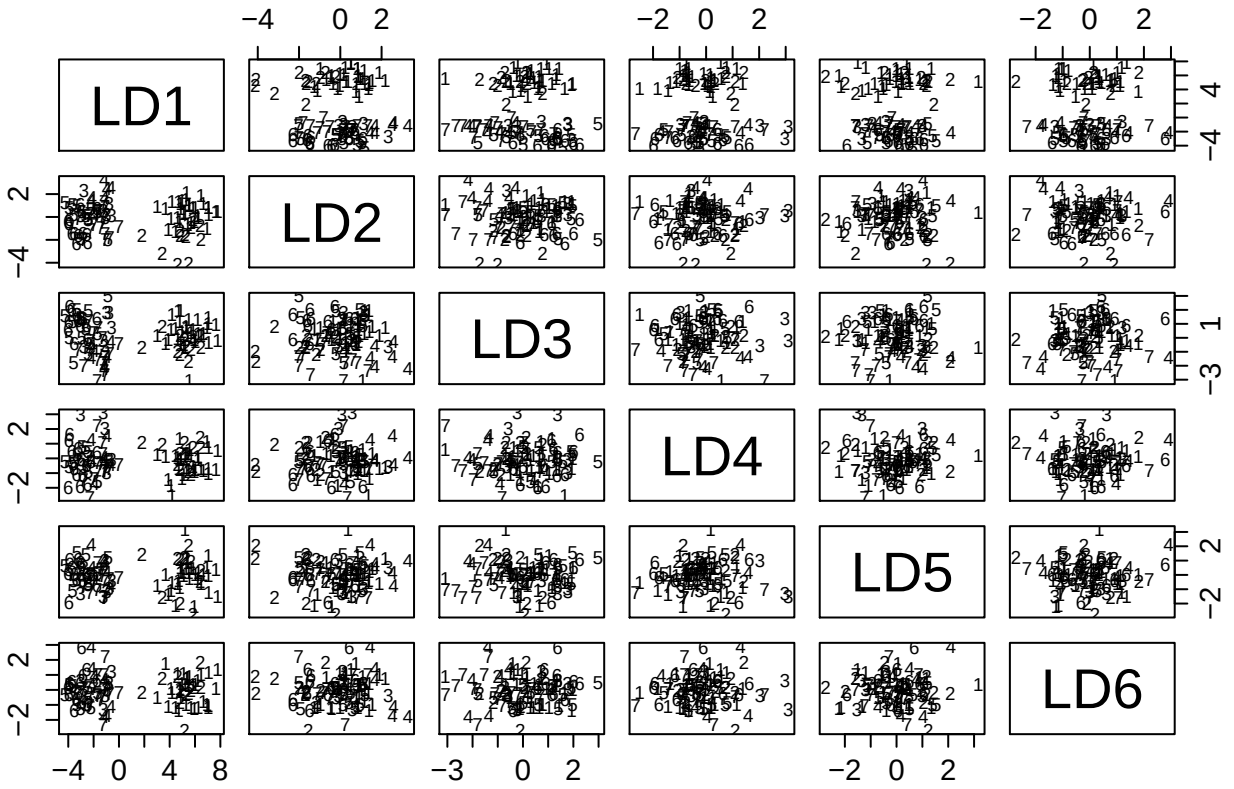
Sie finden auf der Webseite der Veranstaltung unter `Daten zur Vorlesung` einen Datensatz `possum.csv`.

a) Lesen Sie diesen Datensatz ein. Notieren den Befehl, um den Inhalt der Datei in einer Variablen abzulegen. Beachten Sie die Struktur der Datei mit der Kopfzeile und dem Leerzeichen als Trennzeichen!

```
possum <- read.table(
    file="http://fawn.hsu-hh.de/~steuer/Vorlesungsdaten/possum.csv"
    sep=" ", dec=".", header=TRUE)
```

b) Führen Sie mit den Daten eine LDA durch, die die Variablen `hdlngth`, `skullw`, `totlngth`, `taill`, `footlgth`, `earconch`, `eye`, `chest` und `belly` einbezieht, um die bekannte Klassenzugehörigkeit aus der Klasse `site` zu erklären. Notieren Sie den notwendigen Befehl.

```
attach(possum)
library(MASS)
ld1 <- lda(site ~  hdlngth + skullw + totlngth + taill +
           footlgth + earconch + eye + chest + belly)
ld1
plot(ld1)
```


LD1
LD2
LD3
LD4
LD5
LD6

Interpretation

c) Wie viele der erhaltenen Diskriminaten erscheinen Ihnen wichtig und wieso?

- Schaut man sich den Proportion of Trace an, so würde man eine oder maximal drei Diskriminanten wählen. Die erste Diskriminanten erklärt bereits 90 % des Kriteriums. Wenn man die zweite Diskriminante mit einbezieht, muss man auch die dritte mit einbeziehen, beide erklären ähnliche Anteile.

- Die höheren Diskriminaten sind uninteressant.

- Schaut man sich den LDA Standardplot an, würde man evtl. mit einer Diskriminanten zufrieden sein, da nur dort Struktur in den Daten zu erkennen ist.

Mathematische Herleitung der Hauptkomponenten

- Zunächst unter den Annahme, dass die Varianz-Kovarianz-Matrix $\mathbf{\Sigma}$ bekannt ist.

- Gegeben sei eine p-dimensionale ZV $\mathbf{X}$ mit $Var(\mathbf{X}) = \mathbf{\Sigma}$.

- Gesucht ist die Linearkombination $a_1'X$ mit größter Varianz unter allen Linearkombinationen mit der Nebenbedingung $a_1'a_1 = 1$.

- Es gilt

$$Var(a'X) = a'\Sigma a.$$

Mathematische Herleitung der Hauptkomponenten II

- Das Optimierungsproblem unter Nebenbedingung

$$\max_a a'\Sigma a \text{ unter der Bedingung } a'a = 1$$

führt zum Lagrangeansatz

$$L(a, \lambda) = a'\Sigma a - \lambda(a'a - 1).$$

- Die partiellen Ableitungen ergeben sich zu

$$\begin{aligned} \frac{\delta}{\delta a} L(a, \lambda) &= 2\Sigma a - 2\lambda a, \\ \frac{\delta}{\delta \lambda} L(a, \lambda) &= 1 - a'a. \end{aligned}$$

Mathematische Herleitung der Hauptkomponenten III

- Damit folgt aus der ersten Gleichung, dass eine notwendige Bedingung für a_1 in der Erfüllung der Eigenvektoreigenschaft

$$\Sigma a_1 = \lambda a_1$$

  besteht.

- Welcher der p Eigenwerte, die es in diesem Fall gibt, liefert nun die maximale Varianz?

- Unter den notwendigen Bedingungen gilt nun

$$Var(a_1' X) = a_1' \Sigma a_1 = a_1' \lambda a_1 = \lambda a_1' a_1 = \lambda.$$

Mathematische Herleitung der Hauptkomponenten IV

- Das heißt die Varianz von $a_1' X$ ist gleich dem Eigenwert, der zum Eigenvektor a_1 gehört.

- Damit ist klar, dass der Eigenvektor a_1, der zum größten Eigenwert λ_1 der Matrix Σ gehört, das Optimierungsproblem unter Nebenbedingungen löst!

Die weiteren Hauptkomponenten

- Analog ist für die zweite Hauptkomponente a_2 die Optimierung unter Nebenbedingungen

$$\max_{a_2} a_2' \Sigma a_2' \text{ unter } a_2' a_2 = 1 \text{ und } a_2' a_1 = 0$$

  zu lösen.

- Führt man hier ebenfalls eine Lagrange-Optimierung durch, sieht man nach einigem Rechnen, dass auch a_2 Eigenvektor von Σ sein muss, folglich zum zweitgrößten Eigenwert λ_2.

- Weitere Hauptkomponenten folgen nach dem gleichen Verfahren als die Eigenvektoren, die zu den der Größe nach geordneten Eigenwerten gehören.

Erste Bemerkungen zur PCA

- Wieviele der Hauptkomponenten soll man in die Betrachtung einbeziehen?

- Da die wahre Kovarianzmatrix in der Regel unbekannt ist, nimmt man schlicht die empirische Kovarianzmatrix der **zentrierten** Datenmatrix X als Schätzung von Σ.

- Wenn die Varianz der einzelnen Merkmale sich stark unterscheidet, benutzt man auch die Korrelationsmatrix anstelle der Kovarianzmatrix. Dies entspricht einer Normierung auf Varianz 1 der einzelnen Komponeneten.

- Achtung: Die PCA ist folglich offensichtlich nicht skalenunabhängig!

PCA in R

- In R exisitieren zwei Implementierungen der Hauptkomponentenanalyse, prcomp und princomp.

- princomp soll etwas numerisch stabiler sein, auf diese Implementierung beschränkt sich die Vorlesung. Das Beispiel nutzt den iris Datensatz.

```
> p1 <- princomp( ~ Sepal.Length + Sepal.Width
         + Petal.Length + Petal.Width, data = iris)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
2.0494032 0.4909714 0.2787259 0.1538707

 4  variables and  150 observations.
```

Die summary einer PCA in R

- ```
  > summary(p1)
  Importance of components:
                            Comp.1     Comp.2     Comp.3      Comp.4
  Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
  Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
  Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
  ```

- Die Komponenten einer PCA in R:

  ```
  > names(p1)
  [1] "sdev"     "loadings" "center"   "scale"    "n.obs"
   "scores"   "call"
  ```
  ```

Anzahl auszuwählender Hauptkomponenten

- Strebt man mit der Hauptkomponentenanalyse eine Dimensionsreduktion an, so muss festgelegt werden, welche Anzahl $k < p$ von Hauptkomponenten die neuen Koordinaten der Beobachtungen bestimmen sollen.

- Der Anteil der erklärten Varianz ist das einfachste und wohl auch gebräuchlichste Verfahren. Es wird ein Anteil an zu erklärender Varianz α, z.B. $\alpha = 0.9$ festgelegt und alle Hauptkomponenten betrachten, bis

$$\frac{\sum_1^k \lambda_i}{\sum_1^p \lambda_i} > \alpha$$

  .

- Zahlreiche Varianten dieses Kriteriums sind im Umlauf.

Interpretation der Hauptkomponentenanalyse

- Das Ergebnis der PCA ist im Wesentlichen eine Matrix, genannt die Ladungsmatrix (loadings), die in den Spalten die Hauptkomponenten enthält und in den Zeilen die Belegung ("Ladung") der Hauptkomponenten mit der jeweiligen Originalvariablen.

- `R` liefert diese Matrix in der Komponente `loadings`

```
> p1$loadings

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length  0.361 -0.657 -0.582  0.315
Sepal.Width         -0.730  0.598 -0.320
Petal.Length  0.857  0.173        -0.480
Petal.Width   0.358         0.546  0.754
```

Interpretation der Hauptkomponentenanalyse

- In dieser Ausgabe von `R` sind die Einträge freigehalten, deren Werte nahe bei 0 liegen. Die Standardeinstellung hierfür ist, Einträge kleiner als 0.1 zu unterdrücken.

- Da die Hauptkomponenten eine alternative Basis für den R^p darstellen, kann man mit den Beobachtungen eine Koordinatentransformation durchführen und eine Darstellung in Hauptkomponentenkoordinaten bekommen. Diese neuen Koordinaten werden in der PCA als *scores* bezeichnet.

- Diese Information liefert `R` in der gleichnamigen Komponente `scores`.

```
> p1$scores
          Comp.1      Comp.2       Comp.3       Comp.4
1   -2.684125626 -0.31939725 -0.027914828  0.0022624371
2   -2.714141687  0.17700123 -0.210464272  0.0990265503
3   -2.888990569  0.14494943  0.017900256  0.0199683897
4   -2.745342856  0.31829898  0.031559374 -0.0755758166
5   -2.728716537 -0.32675451  0.090079241 -0.0612585926
6   -2.280859633 -0.74133045  0.168677658 -0.0242008576
7   -2.820537751  0.08946138  0.257892158 -0.0481431065
8   -2.626144973 -0.16338496 -0.021879318 -0.0452978706
9   -2.886382732  0.57831175  0.020759570 -0.0267447358
...
```

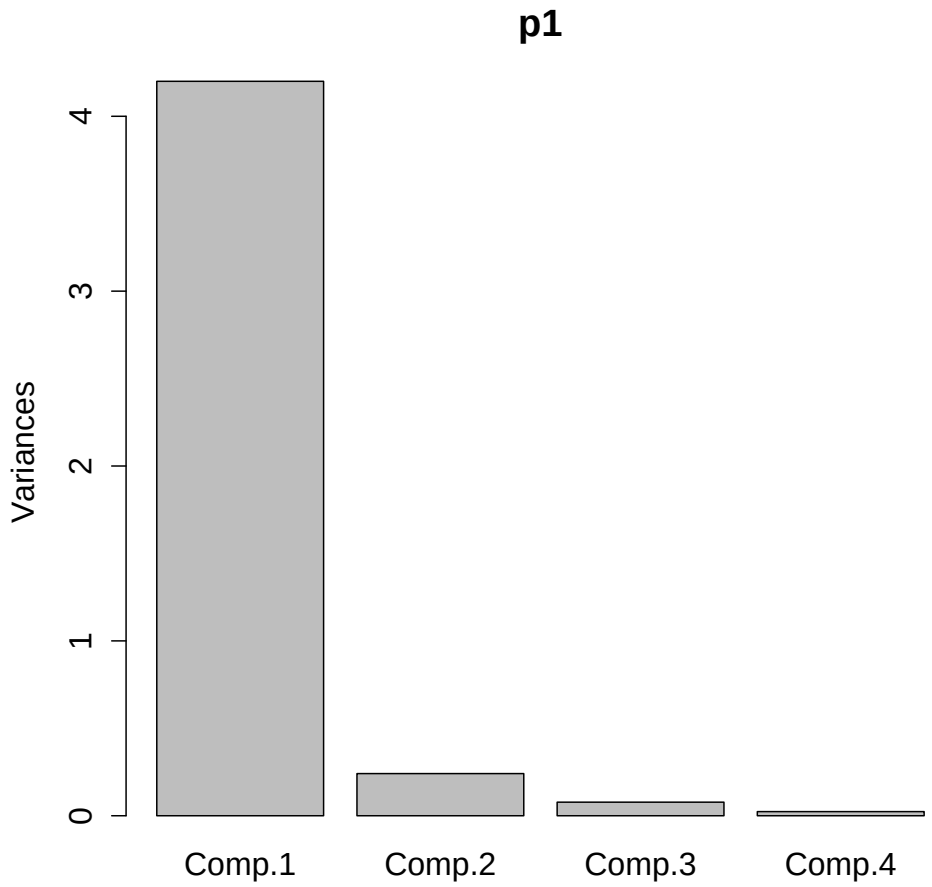
Grafische Verfahren der Hauptkomponentenanalyse

- Die Standardplotmethode für ein Objekt der Klasse Hauptkompontenanalyse ist der sogenannte Screeplot ("Abfallplot"). Er zeigt die Varianzen der Hauptkomponenten als Barplot.

  ```
  > plot(p1)
  ```

- Der Screeplot dient als ein Hilfsmittel, dass zusätzlich zur reinen Einhaltung eines Varianzerklärungsanteils noch eine grafische Komponente in die Auswahl der richtigen Anzahl von Hauptkomponenten mit aufnimmt. Die Idee ist, alle Hauptkomponenten aufzunehmen, bis man einen plötzlichen Abfall in der Varianz sieht, nach dem nur noch Abfall *scree* kommt.

Beispiel für einen Screeplot



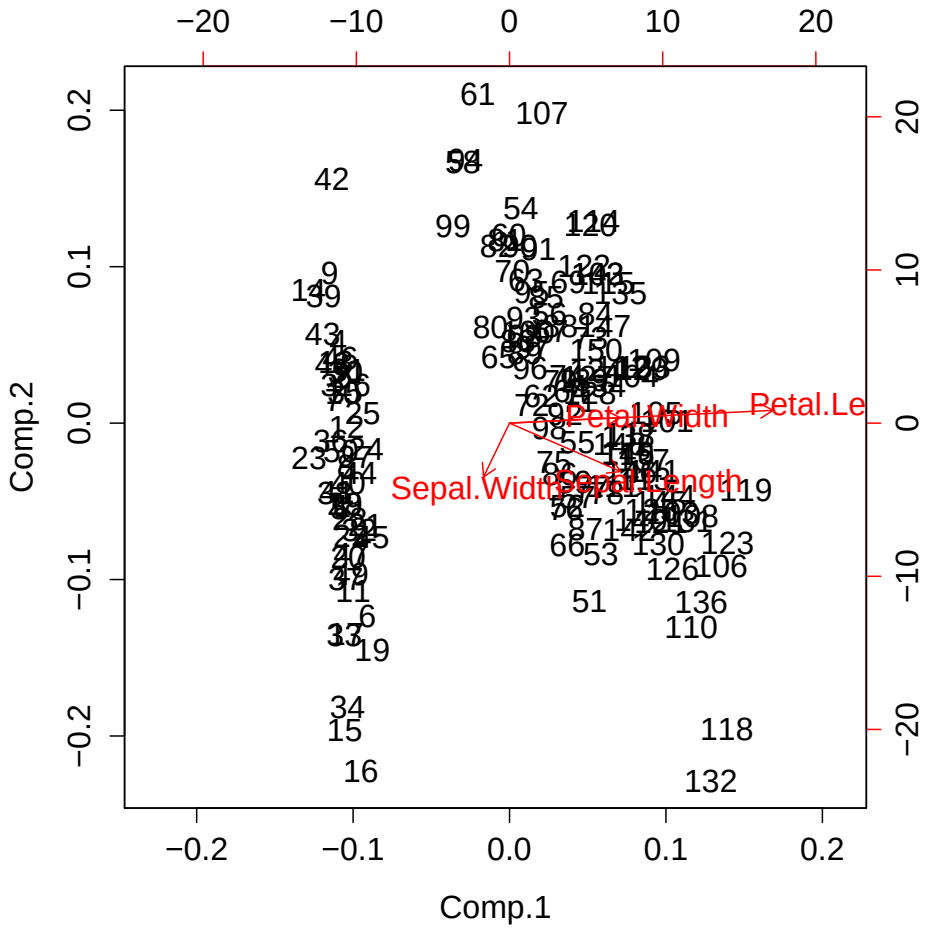
Grafische Verfahren der Hauptkomponentenanalyse II

- Eine zweite Plotmethode, welche im Rahmen der Hauptkomponetenanalyse Anwendung findet, der Biplot. Im Biplot werden die Beobachtungen in den Koordinaten der ersten beiden Hauptkomponenten als Scatterplot eingezeichnet. Zusätzlich werden dazu im Ursprung noch die transformierten Originalkoordinaten abgetragen.

  ```
  > biplot(p1)
  ```

- Am Biplot kann man oft Strukturen finden, die erst durch die Bildung der Scores der Hautpkomponenten aus den Originaldaten herausgearbeitet werden können. Am Beispiel wird ersichtlich, dass die Variable `Petal.Width` keine zusätzliche Information gegenüber `Petal.Length` enthält. Im großen Maßstab sind so evtl. Kosteneinsparung möglich, da Messungen eingespart werden können.

Beispiel für einen Biplot



Vergleich LDA und PCA

- Die Ideen hinter beiden Verfahren sind ähnlich.
- Die Diskriminanzanalyse möchte kategorielle Variablen vorhersagen. Eine bekannte Klasseneinteilung ist nötig.
- Die PCA ist zunächst rein explorativ und schaut sich lediglich die gemessenen Werte an.
- Die PCA liefert oft inhaltlich interpretierbare Ergebnisse.
- Beide Verfahren können zur Dimensionsreduktion eingesetzt werden!

Aufgabe zur PCA

- Führen Sie eine PCA mit dem Datansatz `USArrests` durch!

Übung zur Klausurvorbereitung

Im Paket DAAG finden Sie einen Datensatz `hills2000`, der die Daten der schottischen Bergläufe über 1984 hinaus fortschreibt.

- Führen Sie nach Geschlechtern getrennte Regressionen durch!
- Vergleichen Sie die Modelle, die Sie erhalten mit den Ergebnissen der Daten bis 1984!

Musterlösung PCA für USArrests

```
data(USArrests)
pca1 <- prcomp(~ ., data=USArrests)
plot(pca1) ## 1 optisch 1 Hauptkomponente.
summary(pca1)
Importance of components:
                          PC1     PC2    PC3     PC4
Standard deviation     83.732 14.2124 6.4894 2.48279
Proportion of Variance  0.966  0.0278 0.0058 0.00085
Cumulative Proportion   0.966  0.9933 0.9991 1.00000
### Proportion of Variance spricht für 2 HK.
biplot(pca1)
### keine besondere Struktur, aber wieder überragende 1. HK
```

Musterlösung Klausurbeispiel

Im Paket DAAG finden Sie einen Datensatz `hills2000`, der die Daten der schottischen Bergläufe über 1984 hinaus fortschreibt.

- Führen Sie nach Geschlechtern getrennte Regressionen durch!
- Klarer formuliert: Führen Sie nach Geschlechtern getrennte Regressionen durch, um Modelle für die Laufzeiten aus den Daten für Distanz und Höhenmeter abzuleiten!
- Im Folgenden eine kommentierte Muster-R-Sitzung zur Lösung dieser Aufgabe.

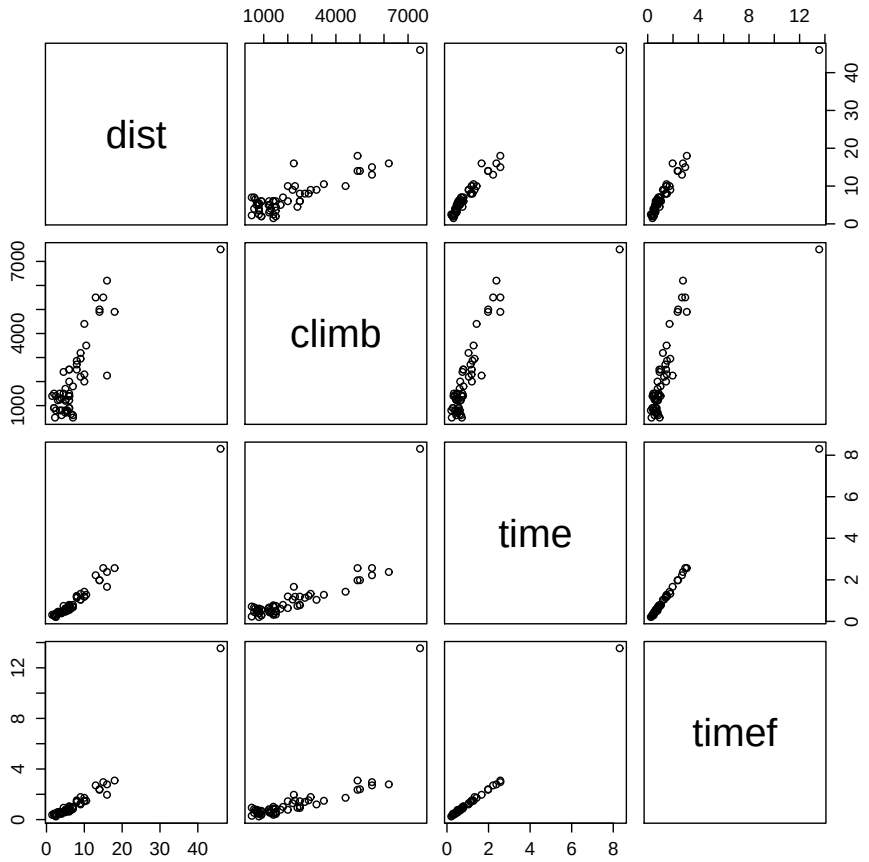
R-Mustersitzung

```
library(DAAG)
# Einbinden der Bibliothek, in der die Daten zu finden sind.

str(hills2000)
# Ansehen der Struktur des Datensatzes

?hills2000
# Erläuternungen zu den Variablen, insbsondere wichtig, um
# herauszufinden, welche Variablen die Zeiten für Frauen
# enthalten

attach(hills2000)
plot(hills2000)
#kurz schauen, ob die Daten eine Regression nahelegen
```


dist
climb
time
timef
1000 4000 7000
0 4 8 12
0 10 20 30 40
1000 4000 7000
0 2 4 6 8
0 4 8 12
0 10 20 30 40
0 2 4 6 8

```
which(dist > 30)
# Eine Beobachtung hat extreme Eigenschaften. Extreme Strecke und
# extreme Steigung.

f.model <- lm(timef ~ dist + climb -1)
m.model <- lm(time ~ dist + climb -1 )
# Berechnug der geforderten Regressionsmodelle
# jeweils ohne Achsenabschnitt aus inhaltlichen Gründen,
# erst einmal ohne Wechselwirkungen
```

```
summary(f.model)
...
Coefficients:
        Estimate Std. Error t value Pr(>|t|)
dist   2.786e-01  2.221e-02  12.541  < 2e-16 ***
climb -2.725e-04  8.382e-05  -3.252  0.00200 **
....
Residual standard error: 0.6385 on 53 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.9207,Adjusted R-squared: 0.9177
F-statistic: 307.6 on 2 and 53 DF,  p-value: < 2.2e-16
```

```
> summary(m.model)
...
Coefficients:
        Estimate Std. Error t value Pr(>|t|)
dist   1.589e-01  1.022e-02  15.543   <2e-16 ***
climb -3.321e-05  3.865e-05  -0.859    0.394
...
Residual standard error: 0.2952 on 54 degrees of freedom
Multiple R-squared: 0.9639,Adjusted R-squared: 0.9625
F-statistic:   720 on 2 and 54 DF,  p-value: < 2.2e-16
```

```
# Verfeinerung der Modelle, indem die Wechselwirkung mit
# aufgenommen wird.

f.model2 <- lm(timef ~ dist + climb + dist:climb-1)
m.model2 <- lm(time ~ dist + climb + dist:climb -1 )

> summary(f.model2)
...
             Estimate Std. Error t value Pr(>|t|)
dist         1.289e-01  1.290e-02   9.992 1.07e-13 ***
climb       -1.501e-04  3.503e-05  -4.285 7.92e-05 ***
dist:climb   2.405e-05  1.475e-06  16.308  < 2e-16 ***

Multiple R-squared: 0.987,Adjusted R-squared: 0.9863
...
```

```
> summary(m.model2)
...
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
dist       8.934e-02  5.228e-03   17.09   <2e-16 ***
climb      2.178e-05  1.433e-05    1.52    0.135
dist:climb 1.133e-05  5.999e-07   18.89   <2e-16 ***
...
Residual standard error: 0.1072 on 53 degrees of freedom
Multiple R-squared: 0.9953,Adjusted R-squared: 0.9951
F-statistic:  3762 on 3 and 53 DF,  p-value: < 2.2e-16
```

Interpretation?

- Was ist eine mögliche Erklärung für die contraintuitiven Ergebnisse bezüglich der Höhenmeter?

```
> summary(lm(climb ~ dist-1))
...
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
dist   243.69      13.87   17.57   <2e-16 ***
...
Residual standard error: 1030 on 55 degrees of freedom
Multiple R-squared: 0.8488,Adjusted R-squared: 0.8461
F-statistic: 308.8 on 1 and 55 DF,  p-value: < 2.2e-16

> cor(dist, climb)
[1] 0.8056136
```

- Es fehlen zum Vergleich Strecken, die sich in ihrer durchschnittlichen Steigung deutlich unterscheiden!

- Lässt man Beobachung 19 aus, ergibt sich ein einleuchtenderes Bild:

```
> summary(lm(timef[-19] ~ dist[-19] + climb[-19] +
             dist[-19]:climb[-19]  -1))
...
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
dist[-19]             9.966e-02  6.660e-03  14.964  < 2e-16 ***
climb[-19]            1.459e-04  2.841e-05   5.135 4.47e-06 ***
dist[-19]:climb[-19] 5.140e-06  1.620e-06   3.172  0.00256 **
```

```
...
Multiple R-squared: 0.9908,Adjusted R-squared: 0.9903
F-statistic:  1830 on 3 and 51 DF,  p-value: < 2.2e-16

> summary(lm(time[-19] ~ dist[-19] + climb[-19] +
              dist[-19]:climb[-19]  -1))
...
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
dist[-19]             8.120e-02  4.537e-03  17.897  < 2e-16 ***
climb[-19]            1.043e-04  1.957e-05   5.332 2.14e-06 ***
dist[-19]:climb[-19] 6.055e-06  1.116e-06   5.425 1.53e-06 ***
...
Multiple R-squared: 0.9935,Adjusted R-squared: 0.9931
F-statistic:  2653 on 3 and 52 DF,  p-value: < 2.2e-16
```

Vergleich mit den `hills` Daten

- Vergleichen Sie die Modelle, die Sie erhalten mit den Ergebnissen der Daten bis 1984!

- Vergleicht man die geschätzten Parameter, so fällt auf, dass die Hauptfaktoren einen größeren Einfluss bekommen haben, die Bedeutung der Wechselwirkung aber allem Anschein nach abgenommen hat. Allerdings ist die Interpretation nicht eindeutig, da die Einflussfaktoren stark korrelieren. Die Erklärungsgüte des Modells ist jedoch mit einem R_a^2 von ca. 0.99 in etwa konstant geblieben.

- Da `dist` und `climb` stark korreliert sind, kann man keinen Koeffizienten für eine der Variablen schätzen, der sich leicht interpretieren ließe. Dies wäre einfacher, wenn es z.B. Läufe mit sehr wenig Höhenmetern gäbe.

Zeitreihenanalyse - Literatur

- Time Series Analysis and its Applications: Shumway und Stoffer, Springer (Webseite mit Daten und R Programmen: http://www.stat.pitt.edu/stoffer/tsa2/)

- Zeitreihen: Schlittgen und Streitberg, Oldenbourg

- Der Weg zur Datenanalyse: Fahrmeir, Künstler, Pigeot und Tutz (online über die Bibliothek verfügbar)

Zeitreihen - Definition

- Von einer (univariaten) Zeitreihe spricht man, wenn lediglich eine Zielgröße Y zu verschiedenen Zeitpunkten $t_i, i \in \mathcal{G}$ beobachtet wird.

- Die Zeitreihe wird so durch eine Abbildung $Y : \mathcal{G} \rightarrow \mathcal{R}$ repräsentiert. Im Fall von zeitkontinuierlichen Beobachtungen ist $\mathcal{G} = \mathcal{R}$, bei diskreten Beobachtungszeitpunkten gilt $\mathcal{G} \subseteq \mathcal{Z}$.

- Modelle, die für Zeitreihen entwickelt werden, heißen
  - global, wenn sie alle Daten der Zeitreihe simultan zur Schätzung der Parameter nutzen und
  - lokal, wenn nicht.

Grundlegende Überlegungen

- Die zeitliche Struktur der Daten in einer Zeitreihe verletzt in der Regel die Annahme von unabhängig identisch verteilten Zufallsvariablen.

- Beispiel: angenommen Sie trainieren für eine Sportart. Es besteht die Hoffnung, dass dann in den Daten eine Verbesserung abzulesen ist, also ein Wert vom Vorgängerwert abhängt.

- Viele der Verfahren, die in der klassischen Statistik angewendet werden, sind deshalb nur mit Vorsicht oder überhaupt nicht anzuwenden.

- Als Beispiel sei das Anfertigen eines Histogramms der Daten einer Zeitreihe genannt. Als Dichteschätzung macht das Histogramm überhaupt keinen Sinn, wenn die Daten einer Zeitreihe entstammen. Ganz evtl. kann

es nützlich sein, um sich einen Überblick über die Streuung der Werte zu verschaffen.

- Ziel der Veranstaltung ist, ein elementares Verständnis von Zeitreihen und den damit verbundenen Fragestellungen vermitteln. Darüberhinaus soll die Fähigkeit zur Bearbeitung und Analyse von Zeitreihen in R geschult werden.

Beispiele für Zeitreihen

- Zeitreihe der Sonnenflecken `sunspots`: Die Anzahl der sichtbaren Sonnenflecken. Diese Zeitreihe wird seit 1749 kontinuierlich (monatlich) aufgezeichnet!

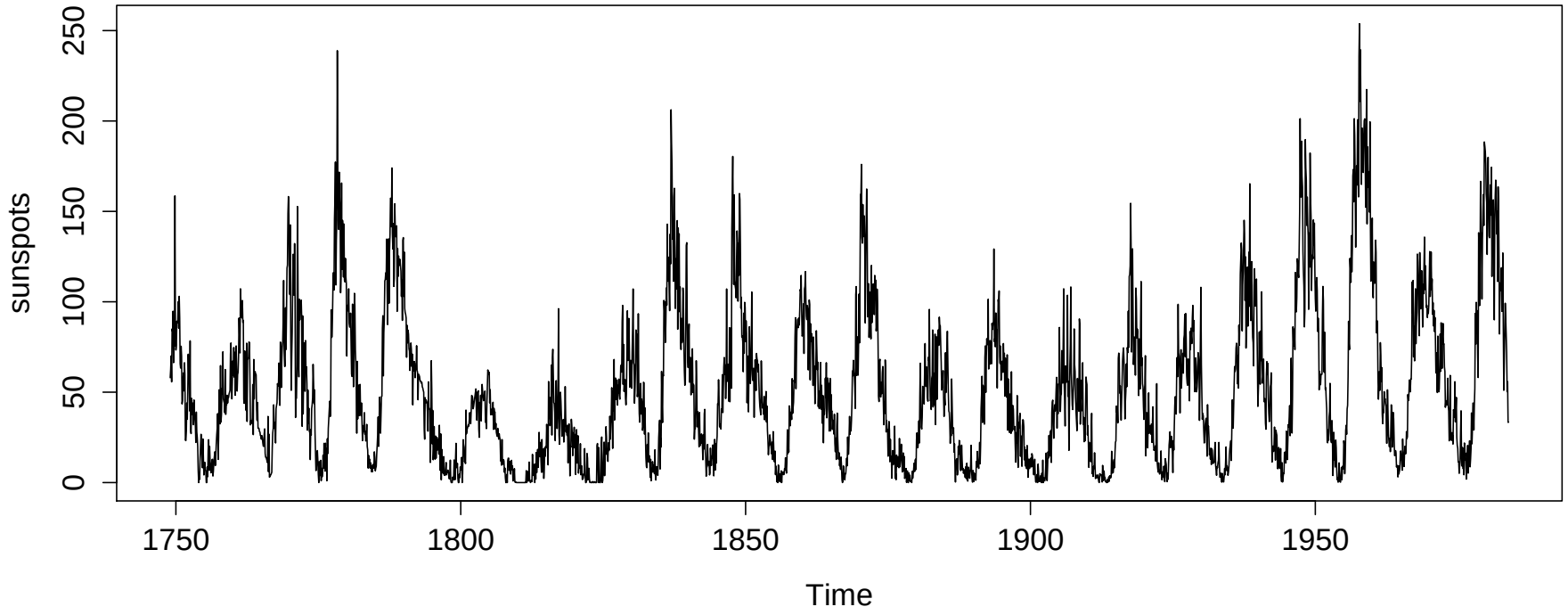
- Die Daten sind in `R` enthalten.

- Neuere Daten z.B. `http://sidc.oma.be/sunspot-data/`

- Es gibt einen speziellen Datentyp `ts` in `R`, um Zeitreihen zu repräsentieren.

```
data(sunspots)
plot(sunspots)
```

Zeitreihenplot `sunspots`



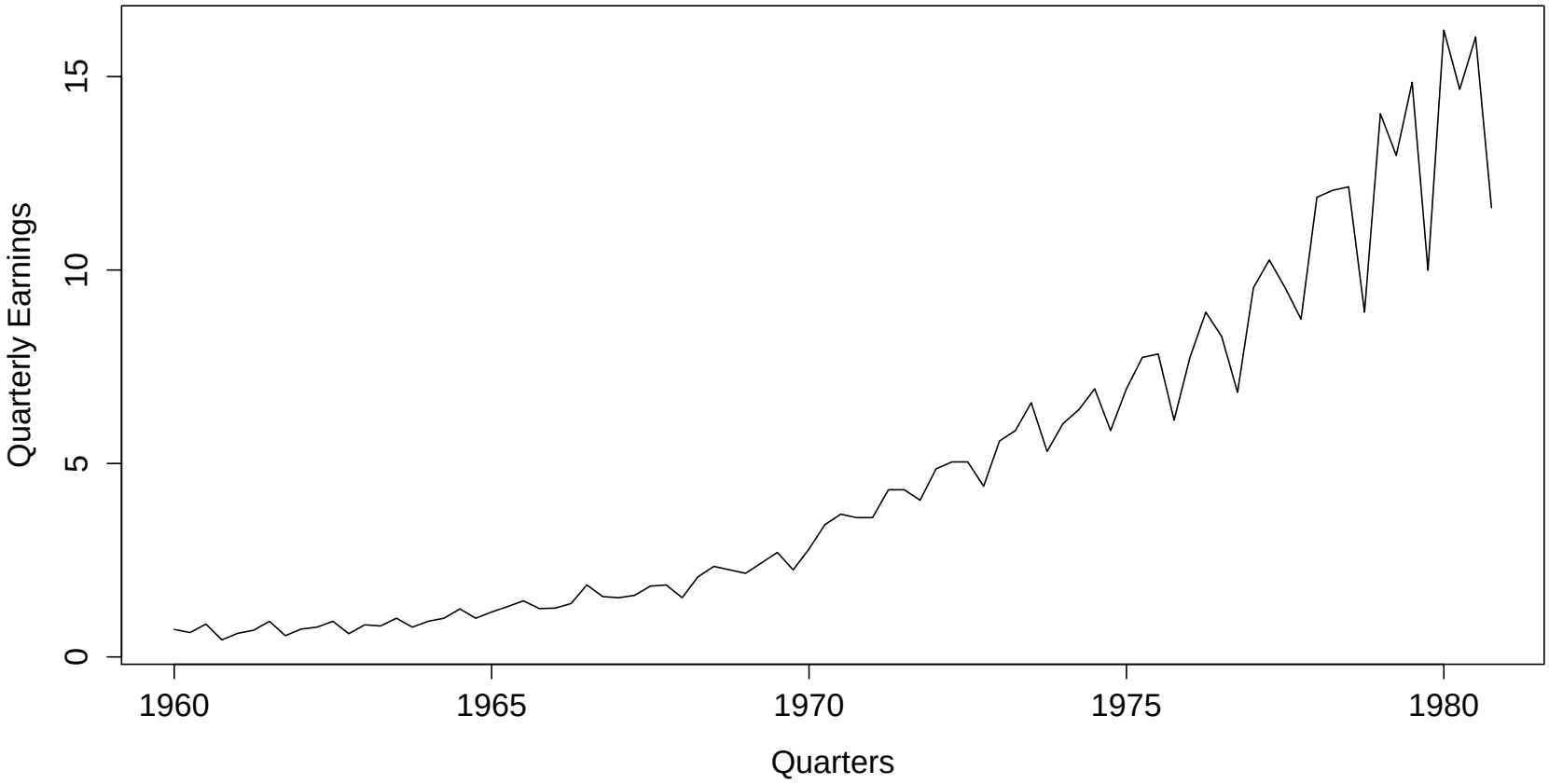
Der Zeitreihentyp in R

- Beispiel: Vierteljährliche Gewinne je Aktie, Johnson & Johnson

```
jj <- scan("TSA/data/jj.dat")
Read 84 items
jj <- ts(jj, start=1960, freq=4)
?ts
plot(jj, ylab="Quarterly Earnings",xlab="Quarters")
```

- Ein Zeitreihenobjekt besteht aus den Daten, einem Startzeitpunkt und einer Angabe darüber, wieviel Beobachtungen pro Zeiteinheit vorliegen.

Zeitreihenplot Johnson & Johnson



Zeitreihen-Komponenten

- Zeitreihen sollen in die Zukunft extrapoliert werden (Prognose).
- Idee: Eine Zeitreihe wird in verschiedene, deterministisch von der Zeit abhängende Komponenten und eine 'zufällige' Rest-Komponente zerlegt.
- Als derartige Komponenten haben sich etabliert: Trend, Konjunktur, Saison, Kalender und Rest.
- Konzeptionell kann man sich eine Zeitreihe dann als Summe ihrer Komponenten plus einem nicht erklärten Rest vorstellen.
- Im Folgenden wird von Daten ausgegangen, die über mehrere Jahre gemessen werden. Wenn die Grundeinheiten kürzer sind, müssen im Folgenden die Anpassungen sprachlich vorgenommen werden. Konzeptionell ändert sich nichts.

Zeitreihen-Komponenten: Trend

- Achtung: Trend ist kein Synonym für Trendgerade!
- Der Trend einer Zeitreihe ist eine dauerhafte Struktur, die die Gestalt der Zeitreihe längerfristig bestimmt.
- Ein Beispiel ist etwa der Trend im durchschnittlichen Bruttoeinkommen oder auch die Steigerung der Gewinne im Johnson & Johnson Plot.
- Der Trend ist diejenige Komponente, die systematische Niveauänderungen in der Zeitreihe beschreibt.

Zeitreihen-Komponenten: Konjunktur

- Die Konjunkturkomponente beschreibt langsame, mittelfristige Schwankungen um die Trendfigur.
- Die Trennung von Trend und Konjunktur ist oft nicht sinnvoll.
- Beide werden auch oft als *glatte Komponente* oder einfach Trend zusammengefasst, die dann auch weiterhin die systematische Niveauänderung einer Zeitreihe umfasst.

Zeitreihen-Komponenten: Saison

- Die Saisonkomponente umfasst jahreszeitliche Schwankungen. Insbesondere sind die Periodenlängen der in dieser Komponente betrachteten Schwankungen kleiner als ein Jahr (bzw. eine Grundzeiteinheit).

- In der Regel geht es hier um Zyklen, die über den Jahreslauf auftreten.

- Bei genügender Auflösung sind Schwankungen über den Wochen- bzw. Tagesverlauf ebenfalls in der Saisonfigur zu finden.

- Achtung: Die Saisonfigur beschreibt keine Niveauänderung!

Zeitreihen-Komponenten: Kalender

- Die Kalenderkomponente beschreibt Änderungen durch Feiertage und Effekte durch unterschiedlich lange Monate.

- Oft werden auch Kalender und Saison in der Saisonkomponente zusammengefasst.

Zeitreihen-Komponenten: Störungen

- Es werden zwei Arten von Störungen betrachtet:
  - Additive Ausreißer, bei denen Messfehler oder besondere einmalige Effekte auftreten.
  - Innovative Ausreißer (*Innovationen*), bei denen die Struktur der Zeitreihe sich nachhaltig durch eine plötzliche Niveauverschiebung ändert.
  - Entweder ist danach ein langsames Zurückkehren auf das alte Niveau zu beobachten, die Zeitreihe verbleibt auf dem neuen Niveau oder die Zeitreihe verliert ihre Struktur (*Crash).*

- Innovationen spielen eine besondere Rolle bei der Entdeckung von Strukturbrüchen.

Zeitreihen-Komponenten: Rest und Fazit

- Die Rest-Komponente $R(t)$ erfasst die nicht durch die anderen Komponenten erfassten Effekte.

- Es verbleiben die drei Komponenten Trend (T), Saison(S) und Rest (R).

- Bei metrischen Größen kann oft von einem additiven Zeitreihenmodell

$$Y(t) = T(t) + S(t) + R(t), \quad t \text{ Zeitindex}$$

ausgegegangen werden.

- Sind die Zielgrößen Verhältniszahlen, nutzt man gern ein multiplikatives Modell

$$Y(t) = T(t) \cdot S(t) \cdot R(t), \quad t \text{ Zeitindex}$$

- Multiplikative Modelle können duch Logarithmieren in additive Modelle umgewandelt werden.

- Die Wahl geeigneter Trend- bzw. Saisonfunktionen ist oft Gegenstand der Diskussion mit Anwendern. Insbesondere trifft dies bei der Anpassung lokaler Modelle zu.

Der Trend: Übliche Trendfunktionen

- Da die Komponenten nicht direkt aus den Graphen abzulesen sind, ist eine mathematische Methode nötig, um die einzelnen Komponenten zu schätzen.

- Da eine gewissen Hierachie bzgl. der Betrachtungshorizonte der Komponenten besteht, beginnen wir mit einem Modell, das zunächst nur eine globale Trendkomponente enthält:

$$Y(t) = T(t) + R(t) \quad t \,\text{Zeitindex.}$$

- Als Schätzmethode kommt in der Regel die Methode der Kleinsten Quadrate zum Tragen, wobei angenommen wird, dass die einzelnen

Komponenten von einem Parametervektor β abhängen, also

$T(t, \beta) = T(t)$ und als Minimierungsaufgabe $\min_{\beta \in \mathcal{R}^m} \sum_{i=1}^{n} (Y(t_i) - T(t_i, \beta))^2$.

- Einige typische Trendfunktionen sind:
  - Linearer Trend:
$$T(t, \beta) = \beta_0 + \beta_1 t,$$
  - Polynomialer Trend von Grad q:
$$T(t, \beta) = \sum_{i=0}^{q} \beta_i t^i,$$

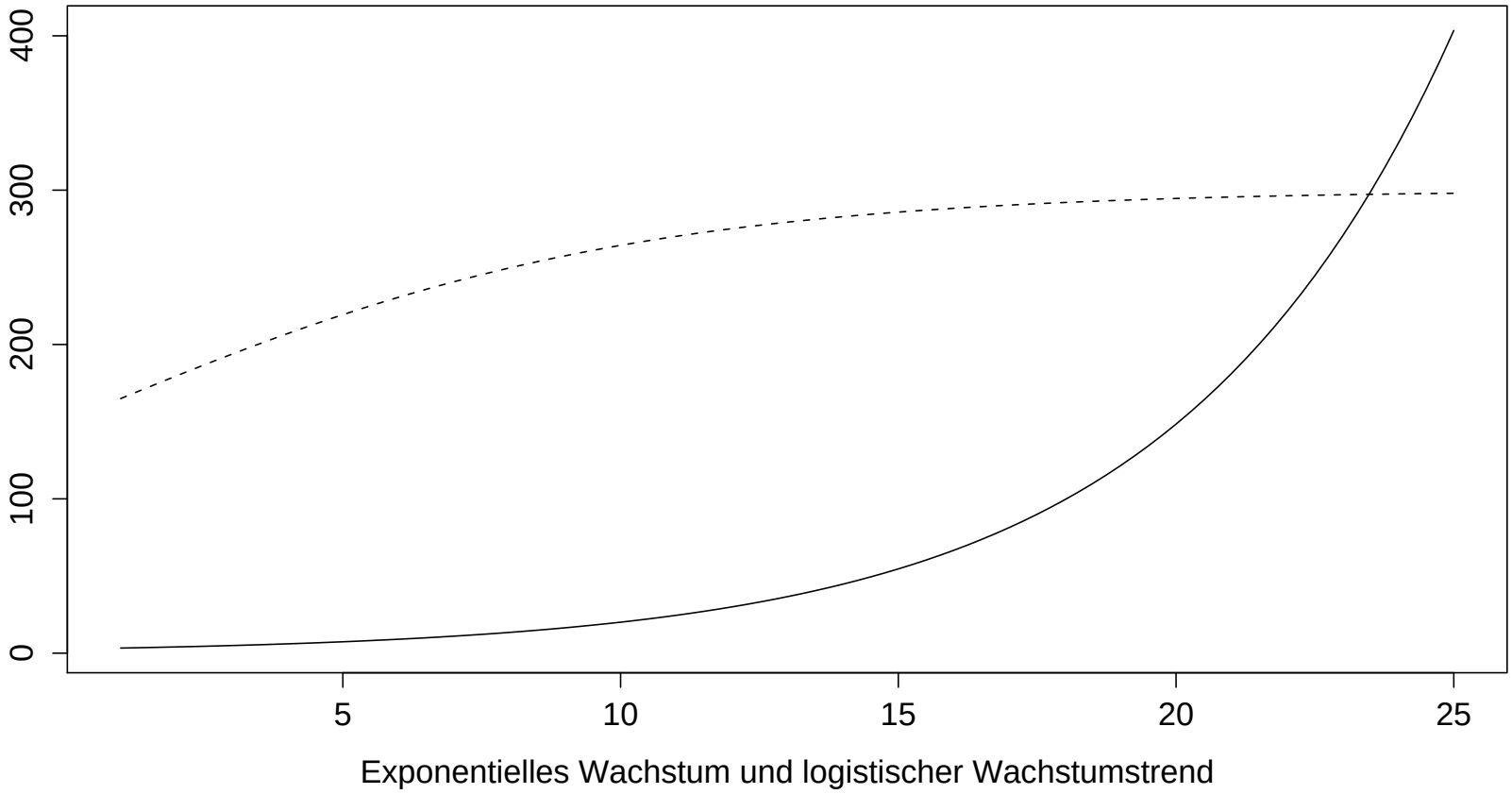
– Exponentielles Wachstum (sog. logarithmische Gerade)

$$T(t, \beta) = e^{\beta_0 + \beta_1 t} = ab^t \text{ mit } a = e^{\beta_0}, b = e^{\beta_1},$$

– Wachstum mit Sättigungsgrenze (logistische Sättigung)

$$T(t, \beta) = \frac{\beta_3}{\beta_2 + e^{\beta_1 t}}.$$

Exemplarische Graphen für einige Trendfunktionen



Saisonmodelle

- Die Saisonfigur schwingt um ein feststehendes Niveau.
- Es gibt zwei wichtige Ansätze zur Modellierung einer Saisonfigur.
- Zeitdiskret kann man ein Modell aufstellen, bei dem eine Regression mit einer Dummyvariablen je Saisonelement gerechnet wird.
- Angenommen man betrachtet Daten mit einer Grundperiode P mit den Abschnitten $j = 1, \ldots, P$ (Quartale, Monate etc.). Dann sei definiert

$$s_j(t) = \begin{cases} 1 & t = Pk + j, k \in \mathcal{Z} \\ 0 & sonst. \end{cases}$$

- Es ist zu beachten, dass entweder alle P Saisonelemente geschätzt werden können oder ein Absolutglied, da offenbar immer gilt

$$\sum_{1}^{P} s_i(t) = 1.$$

- Die Designmatrix für diesen Ansatz hat eine Gestalt, wie sie im Rahmen der ANOVA mit den dortigen Dummyvariablen vorgekommen sind.

- Der zweite, äquivalente Ansatz modelliert die saisonalen Schwingungen als trigonometrische Polynome.

- In diesem Fall wird die Saisonkomponenten S(t) als

$$S(t) = \sum_{j=1}^{(q)} (a_j \cos \lambda_j t + b_j \sin \lambda_j t), t \in \mathcal{R}, \mathcal{Z}$$

  modelliert, wobei $\lambda_j := \frac{2\pi}{P_j}$ für bekannte Perioden $P_j, j = 1, \ldots, q$.

- Der Summationsindex (q) deutet an, dass für den letzten Summanden der Sinusanteil zu streichen ist.

- Frequenzen größer als π können nicht beobachtet werden.

- Stichworte, die die Schwierigkeiten mit der Frequenzdarstellung erläutern sind Aliasing, Maskierung und Frequenzverfälschung.

- Mit geeigneten Designmatrizen lassen sich Modelle polynomialen Trends und die angegebenen Saisonfiguren in einem KQ Verfahren gemeinsam geschätzt werden. Eine gesonderte Trendbereiningung ist nicht erforderlich.

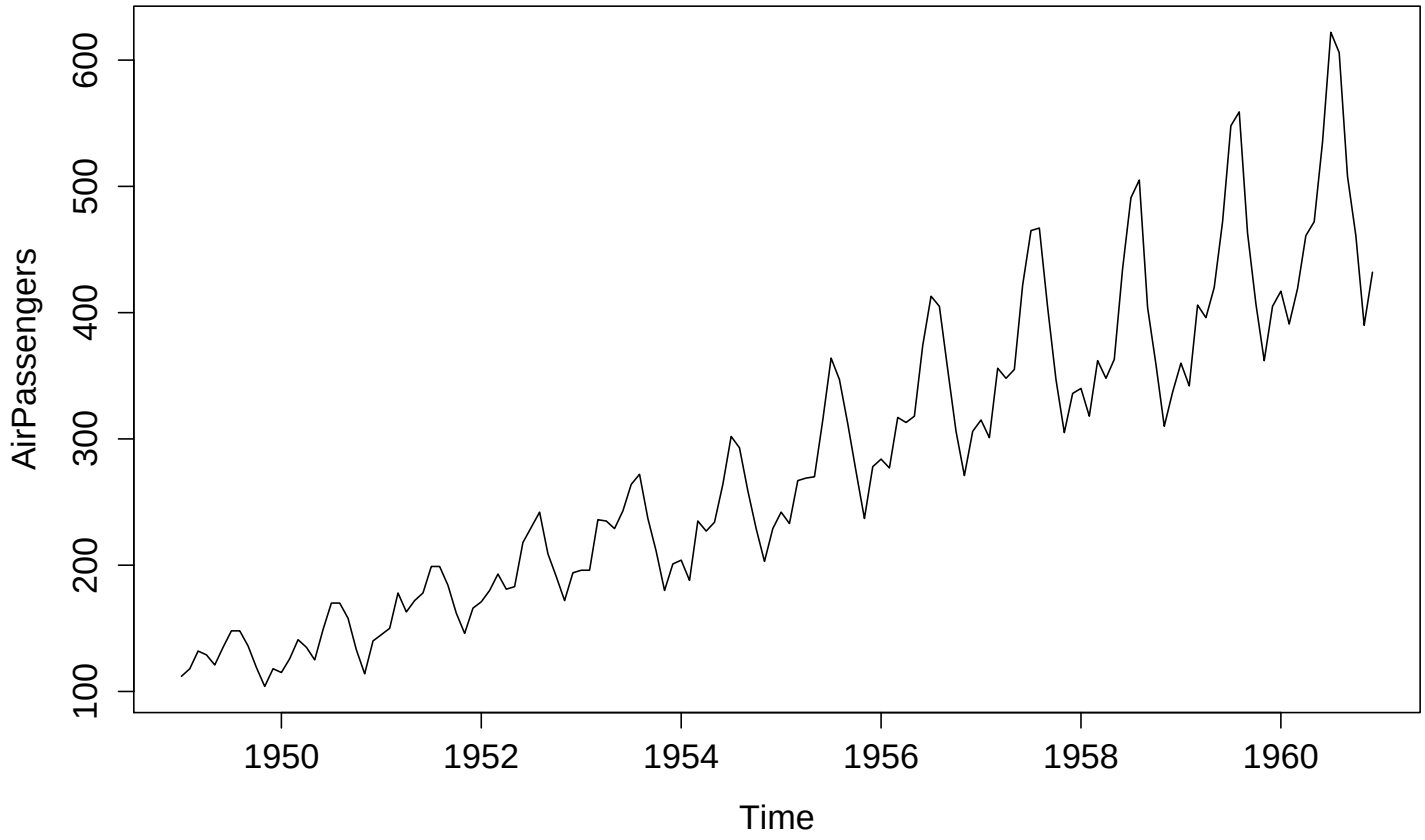
- Ein interessanter Effekt für die Interpretation ergibt sich, wenn ein Modell nach Eingang einer neuen Beobachtung aktualisiert wird. Es ergeben sich auch für die Vergangenheit (theoretisch) andere Prognosen, was zu Irritationen führen kann! Hier liegt einer der großen Vorteile der lokalen Modelle.

Beispiel: AirPassengers

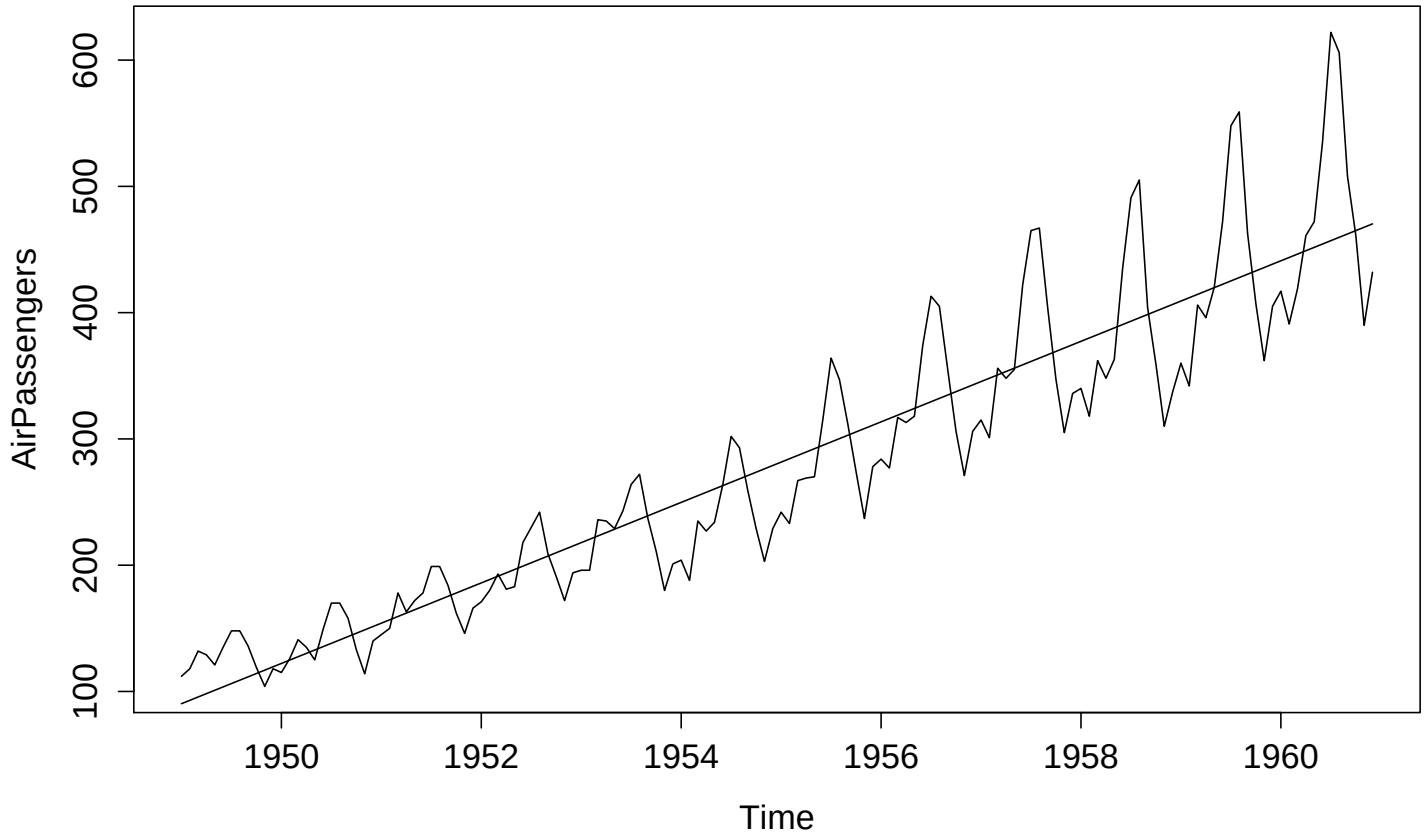
- Daten: Anzahl der monatlichen Flugpassagiere in den USA.
- Der Trend ergibt sich aus einem gewöhnlichen lineaen Modell, bei dem die Zeit der einzige Einflussfaktor ist.

```
data(AirPassengers)
plot(AirPassengers)
## Berechnung eines linearen Trends
trend <- lm(AirPassengers ~ seq(1,144))$fit
lines(ts(trend, start=1949, freq=12))
```

Originalzeitreihe



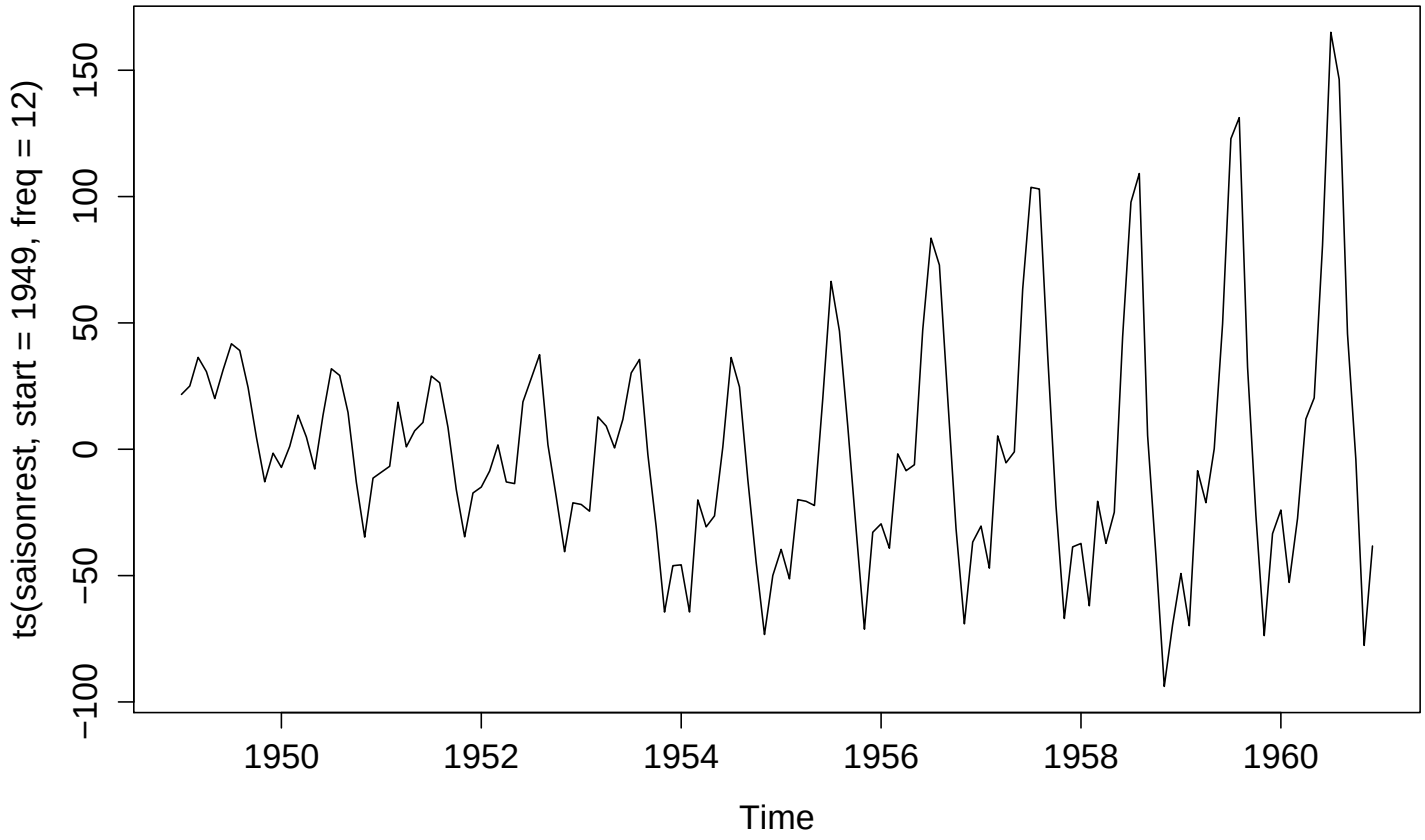
Zeitreihe und linearer Trend



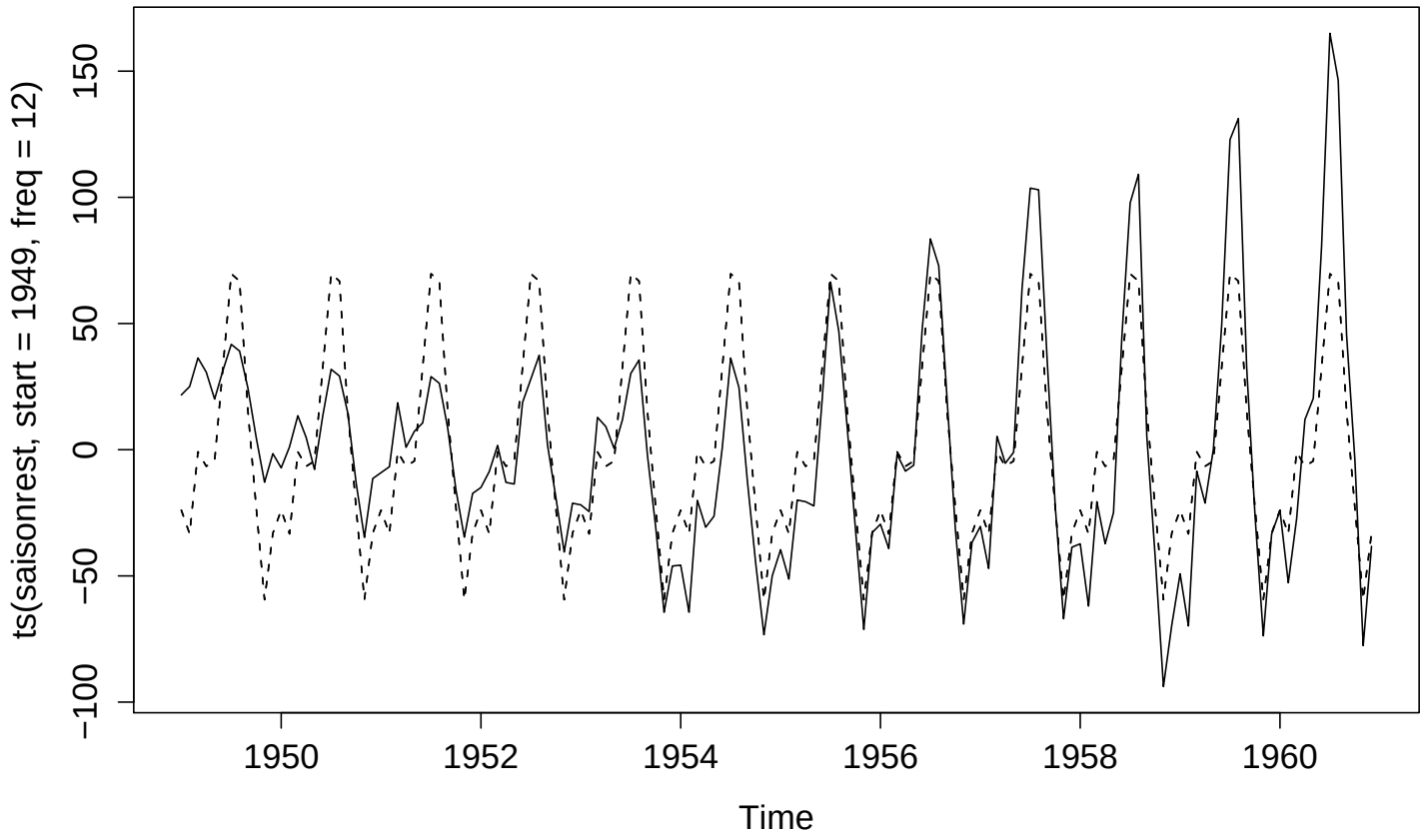
```
## Mit den Residuen kann jetzt die Saisonfigur
## berechnet werden
saisonrest <- lm(AirPassengers ~ seq(1,144))$res
plot(ts(saisonrest, start=1949, freq=12))

## Designmatrix für die Monatsfaktoren
Q <- factor(rep(1:12,12))
saisonfigur <- ts(lm(saisonrest ~ 0 + Q)$fit,
                  start=1949, freq=12 )
lines(saisonfigur, type="l", lty=2 )
plot(AirPassengers - saisonfigur - trend)
## Ganz interessant
model.matrix(lm(saisonrest ~ 0 + Q))
```

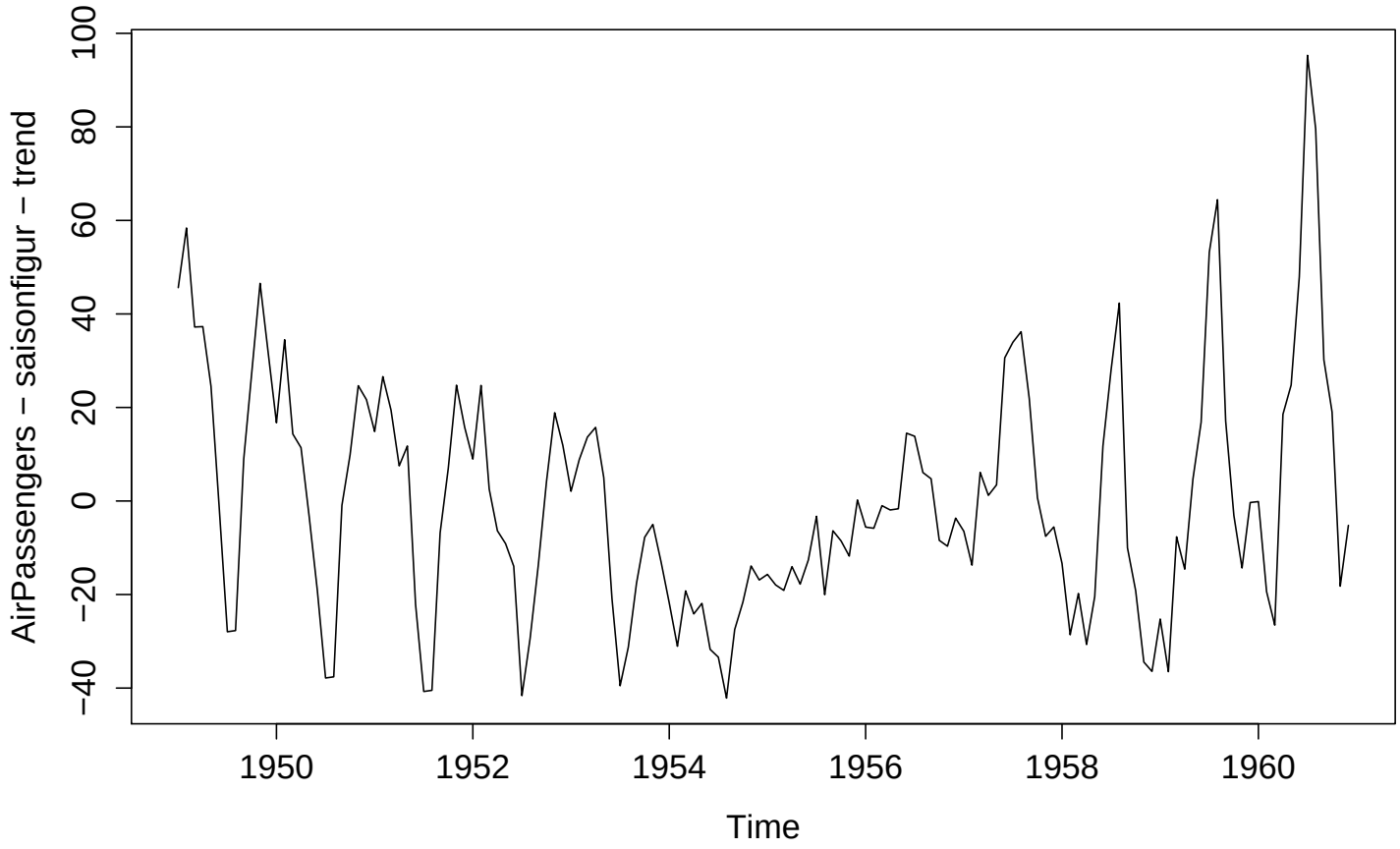
Trendbereinigte Zeitreihe



Trendbereinigte Zeitreihe und geschätzte Saison



Nicht erklärter Rest



Aufgabe zur Zeitreihenanalyse

- Besorgen Sie sich die Daten `jj.dat` von der angegebenen Webseite.
- Passen Sie entsprechend dem Beispiel einen linearen Trend und eine Saisonfigur an. Erzeugen Sie die entsprechenden Plots.

Diskussion der globalen Zeitreihenzerlegung

- Die Gültigkeit globaler Trendmodelle ist nicht sehr wahrscheinlich. Oft ist ein solches Modell zu starr, um kleine Änderungen zu modellieren, wie sie natürlicherweise in vielen Systemen auftreten.

- Wenn das globale Modell gilt, ist natürlich eine besonders einfache Interpretierbarkeit gegeben. Für Flexibilität benötigt man aber lokale Modelle.

- Im Folgenden wird das *Glätten* einer Zeitreihe mittels gleitender Mittel über sog. Fenster der Zeitreihe skizziert. Auf einige andere Verfahren wird hingewiesen.

- Die allgemeine Idee hinter diesem Vorgehen ist, den Trend nicht als feste Funktion, sondern lediglich als *glatte* Kurve durch die Zeitreihe zu beschreiben.

Das gleitende Mittel – Vorüberlegungen

- In einem lokalen Modell werden immer nur Daten in einem sogenannten *Fenster* (window) der Zeitreihe betrachtet, um eine Anpassung für einen bestimmten Zeitpunkt t durchzuführen. Für jeden Zeitpunkt t kommt ein anderes lokales Modell zum Tragen.

- Dieses Fenster wird durch seine Länge q, sprich die Anzahl der berücksichtigten Daten bestimmt.

- Wenn möglich sollte q ungerade gewählt werden.

- Für die lokale Zerlegung um den Zeitpunkt t werden dann die Beobachtungen $y_{t-(q-1)/2}, \ldots, y_t, \ldots, y_{t+(q-1)/2}$ betrachtet.

- Ein Problem gibt es natürlich am Anfang oder am Ende der Zeitreihe, da dort schlicht die Daten fehlen, um ein symmetrisches Fenster um den Beobachtungspunkt herum zu legen.

- Es gibt *optimale* Gewichte, um gleitende Durchschnitte bis zum Ende einer Zeitreihe fortzusetzen, auf die jedoch hier nicht weiter eingegangen werden soll. Im einfachsten Fall werden an diesen Stellen $t < (q-1)/2$ und $t > t_{max} - (q-1)/2$ fehlende Werte erzeugt.

- In Charts von Aktienkursen wird deshalb für den Zeitpunkt t einfach das arithmetischen Mittel der letzen n (typisch 200 oder 39) Tage eingezeichnet. Solche einseitigen gleitenden Mittel sind durchaus üblich, inhaltlich aber eigentlich unsinnig.

Das gleitende Mittel – Definition

- **Definition:** Eine Funktion $g(t)$ folgender Gestalt heißt *gewichtetes, gleitendes Mittel* der Länge q, q ungerade, zum Zeitpunkt t mit den Gewichten w_i:

$$g(t) = \sum_{i=t-(q-1/2)}^{t+(q-1)/2} w_i y(t), \sum w_i = 1.$$

- Der sogenannte einfache gleitende Durchschnitt belegt alle Beobachtungen mit dem gleichen Gewicht $w_i \equiv \frac{1}{q}$.

Das gleitende Mittel in `R`

- Die `R` Funktion für gleitende Mittel ist `filter()`.
- Wenn eine Zeitreihe in der Variablen `ts` vorliegt, gibt `filter(ts, weights)` eine mit dem gleitenden Mittel mit entsprechenden Gewichten geglättete Zeitreihe zurück.
- Am Anfang und am Ende werden *missing values* eingefügt.
- Die Wahl von q bestimmt die Glattheit der angepassten Kurve.

Aufgabe zur Zeitreihenanalyse

- Besorgen Sie sich die Daten `jj.dat` von der angegebenen Webseite.

- Passen Sie entsprechend dem Beispiel einen linearen Trend und eine Saisonfigur an. Erzeugen Sie die entsprechenden Plots.

```
jj <- scan("TSA/data/jj.dat")
### Einlesen der Daten, Speicherort ist natürlich der von Ihnen
### gewählte.
jj <- ts(jj, start=1960, freq=4)
### Umwandeln in Zeitreihe
plot(jj, ylab="Quarterly Earnings",xlab="Quarters")
trend <-  lm(jj ~ seq(1,84) )$fit
```

```
### Trend als globale Regression
lines(ts(trend, start=1960, freq=4))
saisonrest <- lm(jj ~ seq(1,84))$res
### Der trendfreie Rest ergibt sich als Residuen der Regression
plot(ts(saisonrest, start=1960, freq=4))
Q <- factor(rep(1:4,21))
### einfachste Methode die Designmatrix zu erzeugen!
lm(saisonrest ~ 0 + Q)
### Achsenabschnitt inhaltlich unsinnig
saisonfigur <- ts(lm(saisonrest ~ 0 + Q)$fit,
                      start=1960, freq=4 )
lines(saisonfigur, type="l", lty=2 )
plot(jj - saisonfigur - trend)
### Noch ein Blick auf die Reste. Das Modell ist den Daten
### nicht angemessen.
```

Beispiel: Anwendung eines gleitenden Filters

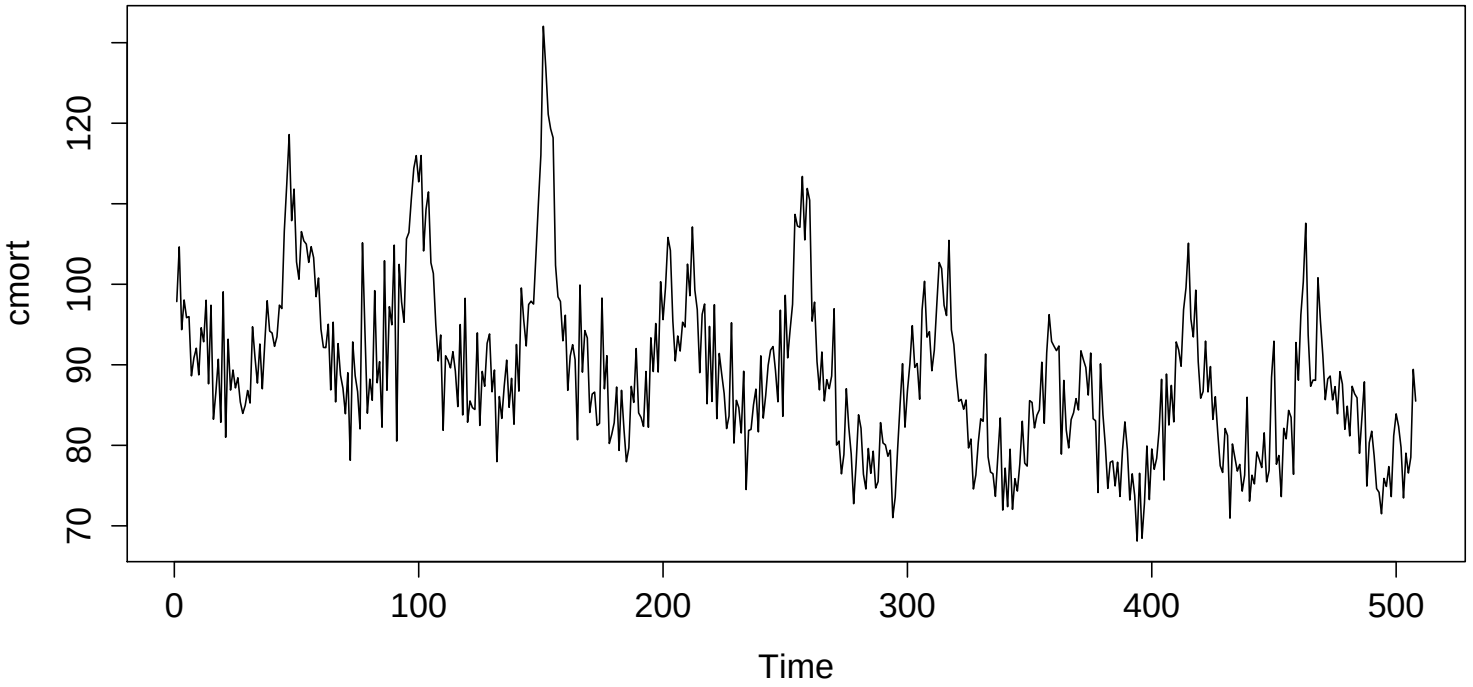
- Als Beispieldatensatz sollen die Mortalitätsdaten aus Los Angeles genutzt werden. (Quelle: Shumway und Stoffer)

- Es liegen Zeitreihen vor für die Temperatur, die Luftverschmutzung und die Todesfälle durch Herz- und Kreislaufprobleme in LA in den Jahren 1970-1979. Alle Daten liegen jeweils als Mittelwerte über 6 Tage vor. Insgesamt sind es 508 Messungen. Hier interessieren zunächst lediglich die Todesfallzahlen.

- Die Daten sollen eingelesen und dann über gleitende Mittel geglättet werden.

Zeitreihenglättung: Mortalität in LA

```
cmort <- ts(scan("TSA/data/cmort.dat")) ; plot(cmort)
```



Zeitreihenglättung: Mortalität in LA

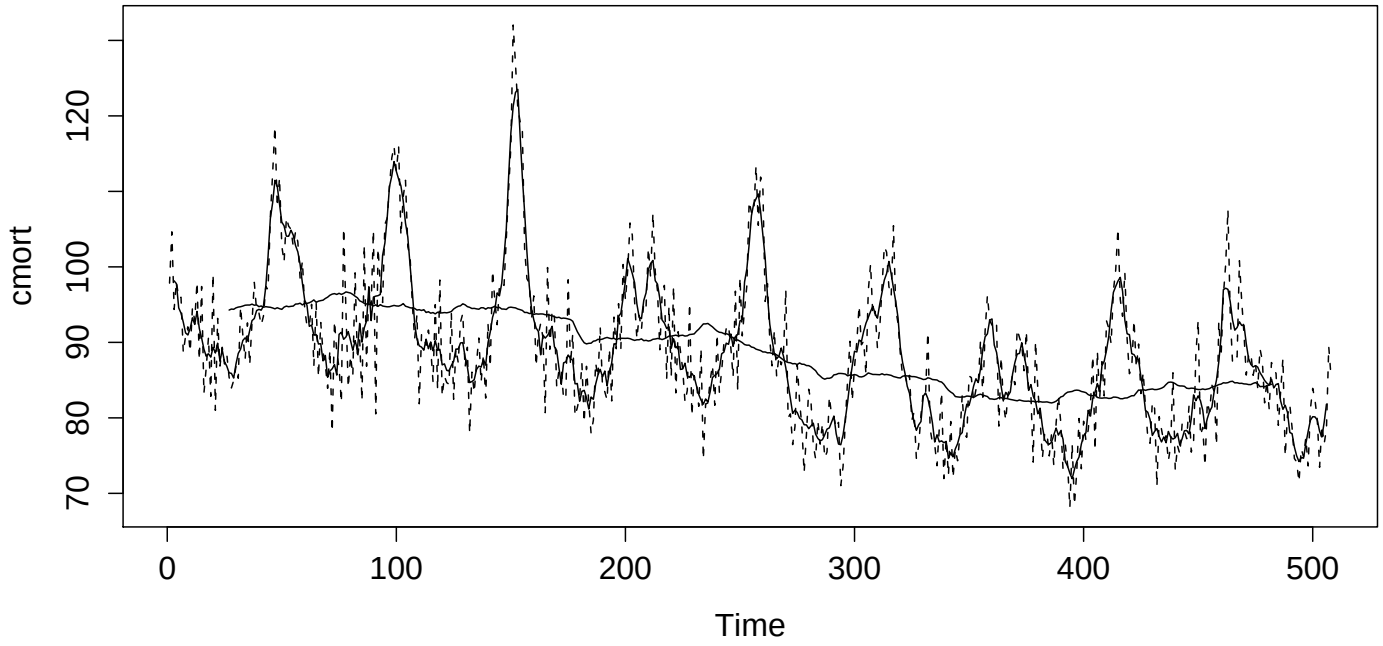
- Klar erkennbar ist ein langfristiger Trend zur Abnahme der Todesfälle wg. Kardioveskularerkrankungen.

- Ebenso klar erkennbar ist eine Saisonfigur, anscheinend jahreszeitabhängig.

- Zunächst finden eines Trendschätzers mittels einen gleitenden Mittels. Dazu werden einige Fensterbreiten ausprobiert, bis ein geeigneter Kompromiss zwischen Flexibilität und Glätte der Kurve gefunden sind.

Eine geeignete Anpassung:

```
lines(filter(cmort, rep(1/53,53))) ### Spannweite ein Jahr
lines(filter(cmort, rep(1/5,5)))   ### Auch Saisonfigur mögl.
```



Gleitende Mittel als lokale Regressionen

- Betrachtet man ein Fenster der Länge q und ersetzt den Wert $y(t)$ durch $\hat{y(t)}$, wobei $\hat{y(t)}$ als Prognose für den Wert zum Zeitpunkt t aus einer linearen Regression mit den Punkten $y(t-(q-1)/2), \ldots, y(t+(q-1)/2))$ berechnet wurde, so erhält man dieselbe Approximation wie bei einem einfachen gleitenden Mittel der Spannweite q. (Herleitung z.B. im Fahrmeir)

- Man kann entsprechend zeigen, dass bestimmte, anders gewichtete, gleitende Mittel zu entsprechenden lokalen, polynomialen Regressionen äquivalent sind.

Wann ist ein Trend glatt genug? Idee der Splineglättung.

- Glattheit ist kein klar definierter Begriff. Es gibt einige Willkür in der Wahl der Fensterbreite.

- Wie gesehen, kann z.B. der gleitende Durchschnitt der Kurve sehr nah folgen (q ist klein) oder immer mehr dem linearen Trend nahe kommen (q ist groß).

- Ein Ansatz ist, ein Maß für die Glattheit der Trendfunktion mit der Güte der Anpassung an die Kurve zu koppeln, genauer zu penalisieren.

- Seien nun $T(t), t = 1, \ldots, n$ der geschätzte Trend, so kann man nun eine Minimierungsaufgabe

$$\sum_{1}^{n}(y(t) - T(t))^2 + \lambda \sum_{2}^{n}(T(t) - T(t-1))^2 \rightarrow \min_{\{T(t)\}} \text{ lösen.}$$

Bedeutung der Parameter in der Minimierungsaufgabe

- Der Parameter λ bestimmt hier die Bedeutung der Glattheit des Trends.
- Im sog. Strafterm $\lambda \sum_2^n (T(t) - T(t-1))^2$ finden sich die ersten Differenzen der Trendanpassung.
- Kleine Werte von λ lassen die Lösung in Richtung der Interpolation der Ursprungsreihe treiben.
- Für große Werte von λ nähert sich die Lösung der Regressionsgerade.
- Ein solches Verfahren kommt z.B. im VBV (Verallgemeinertes Berliner Verfahren, Hebbel) zur Anwendung.
- Leider ist hier die Wahl von λ ebenso willkürlich, wie die Spannweite bei den gleitenden Mitteln.

Sonstige Ansätze zur Zeitreihenzerlegung

- Ganz allgemein können auch andere lokale Regressionsmethoden zur Glättung genutzt werden.

- Ein Beispiel einer solchen Methode war in `scatter.smooth` zu sehen. Übliche Verfahren sind `loess()` oder `lowess()`.

- Um den Einfluß von Ausreißern auf Glättung zu begrenzen, gibt es analoge Verfahren mit gleitenden Medianen, zensierten Mitteln etc.

- Die von den großen statistischen Ämtern angewandten Verfahren Census-X11 oder das Berliner Verfahren sind oft Hybridverfahren, die verschiedene Ansätze, auch iteriert, kombinieren.

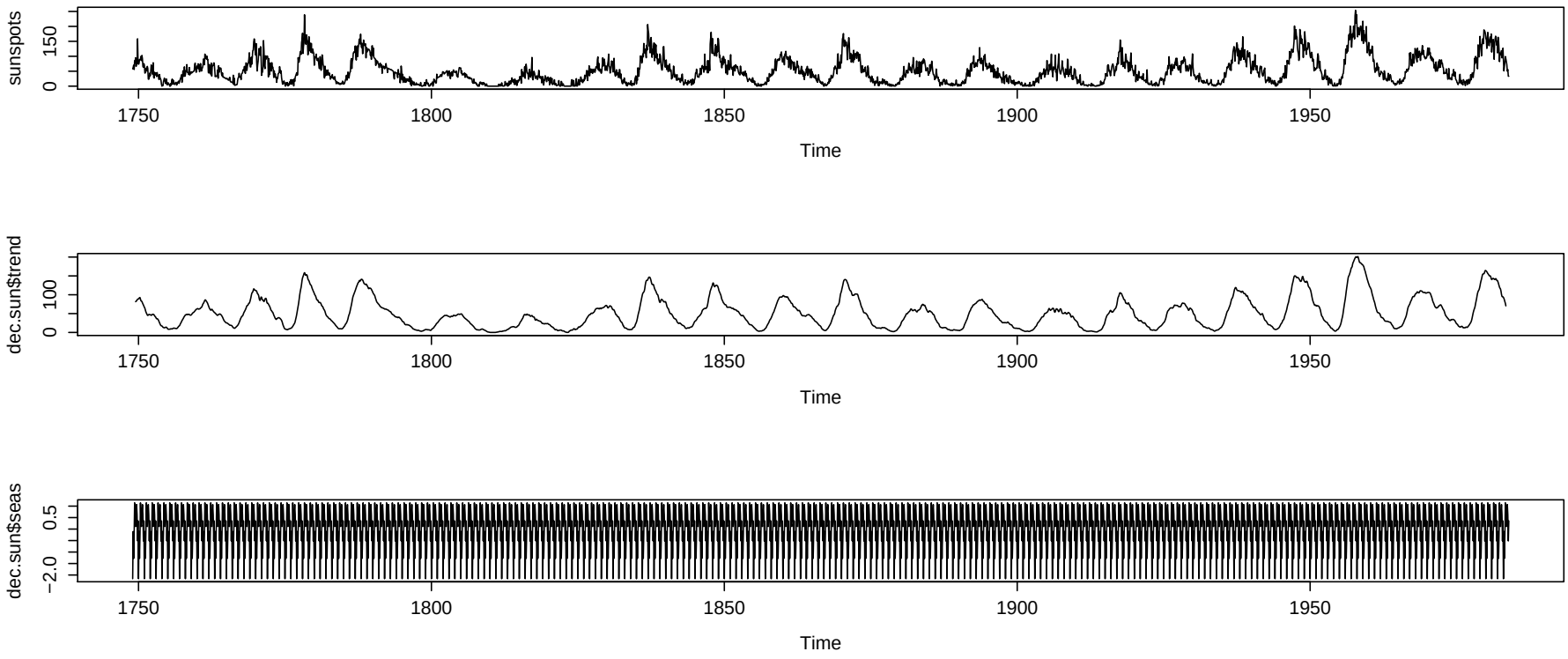
Zeitreihenzerlegung in R

- R hat einige Verfahren zur Zeitreihenzerlegung implementiert.
- Im Standardpaket finden sich `decompose()` und `stl()`.
- Die Argumente dieser Funktionen sind nicht standardisiert.
- Die Anpassung eines globalen Modells muss über die `lm()` Methode erfolgen.

Zeitreihenzerlegung in R: decompose()

- decompose zerlegt die Reihe über gleitende Mittel in ihre Komponenten.
- Beispiel Sonnenflecken: decompose(sunspots)

```
?decompose
dec.sun <- decompose(sunspots)
par(mfrow=c(3,1))
plot(sunspots)
plot(dec.sun$trend)
plot(dec.sun$seas)
par(mfrow=c(1,1))
```

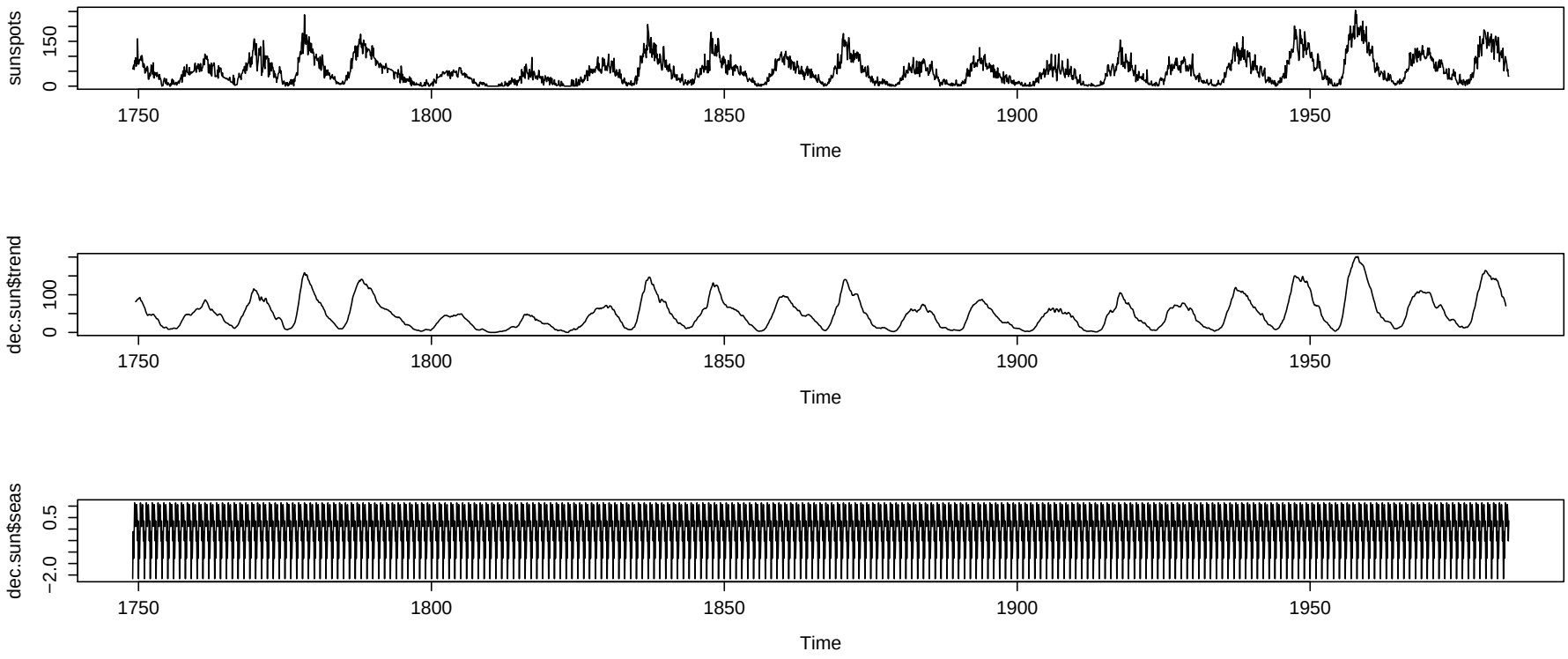

sunspots
0
150
1750
1800
1850
1900
1950
Time
dec.sun$trend
0
100
1750
1800
1850
1900
1950
Time
dec.sun$seas
−2.0
0.5
1750
1800
1850
1900
1950
Time

Zeitreihenzerlegung in R: stl

- stl() zerlegt die Zeitreihe mittel lokaler Regression, genauer LOESS, in ihre Komponenten.

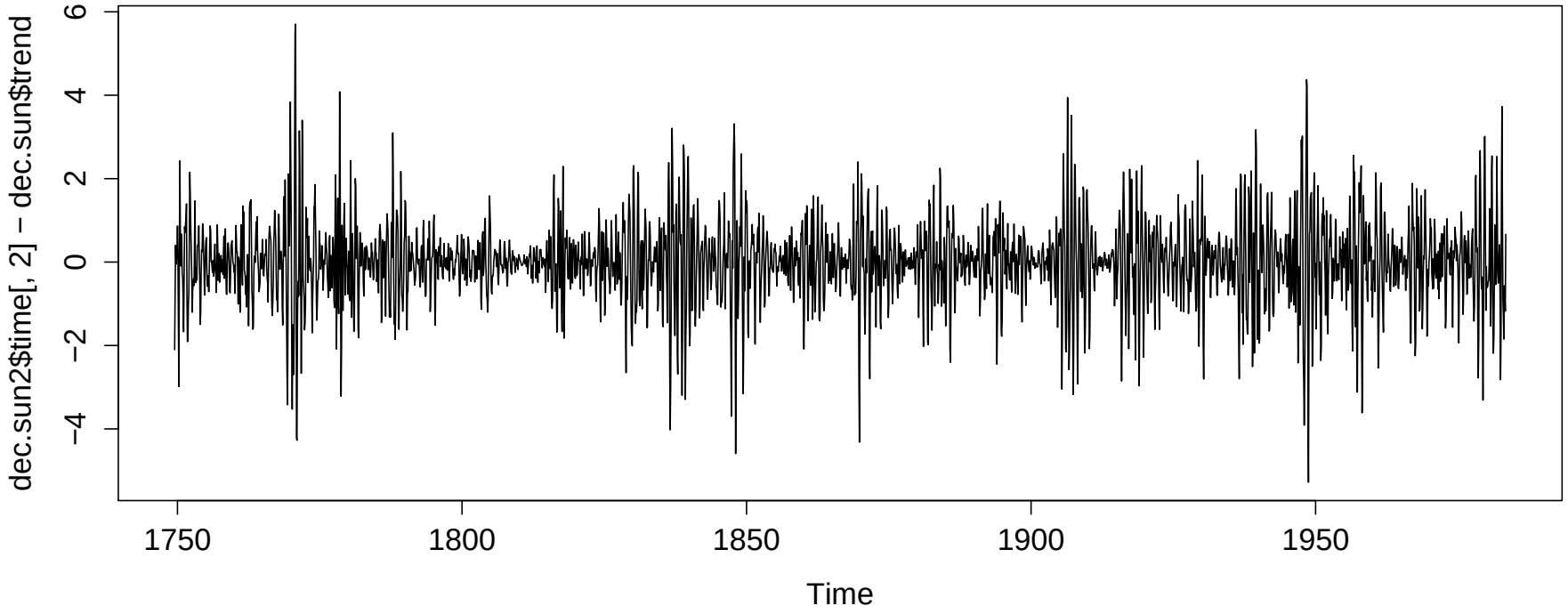
- Wieder die Sonnenflecken als Beispiel.

```
?stl
dec.sun2 <- stl(sunspots, s.window="periodic") ;str(dec.sun2)
par(mfrow=c(3,1))
plot(sunspots)
plot(dec.sun2$time[,2])
plot(dec.sun2$time[,1])
par(mfrow=c(1,1))
```


sunspots
0
150
1750
1800
1850
1900
1950
Time
dec.sun$trend
0
100
1750
1800
1850
1900
1950
Time
dec.sun$seas
−2.0
0.5
1750
1800
1850
1900
1950
Time

Die Figuren sind tatsächlich unterschiedlich!

```
plot(dec.sun2$time[,2] - dec.sun$trend)
```



Aufgabe zur Zeitreihenglättung

Sie finden auf der Webseite von Shumway und Stoffer den Datensatz `globtemp.dat`.

Passen Sie einen Ihrer Meinung nach passenden Trend bzw. Saisonfigur mittels gleitender Mittel an!

Passen Sie ein globales Modell mit linearem Trend an!

Vergleichen Sie die beiden Modelle!

MA-Prozesse, AR-Prozesse und ARMA-Prozesse

- Oft stellt sich der einfache Regressionsansatz als nicht hinreichend zur Erklärung des Verlaufs einer Zeitreihe heraus.

- Eine nahe liegende Idee ist nun, den Wert x_t einer Zeitreihe zum Zeitpunkt t nicht nur als abhängig vom Zeitpunkt, sondern auch abhängig von einer Teilmenge der vorhergehenden Beobachungen $x_{t_1}, x_{t_2}, \ldots, x_{t_k}$ zu modellieren.

- Historisch die erste Prozesse dieser Art waren:
  - AR - auto-regressive, MA - moving averages,
  - ARMA - eine Kombination aus beidem.

- Heute gibt es eine Vielzahl unterschiedlichster Ansätze: ARCH, GARCH, ARIMA etc.

Einige benötigte Begriffe

- Die Funktion $\gamma(s,t) = E((x_s - \mu_s)(x_t - \mu_t))$ heißt Autokovarianzfunktion der Zeitreihe $\{x_t\}$. Hierbei sind von der Zeit abhängige Mittelwerte im Prinzip erlaubt. Im Folgenden sind in der Regel alle $\mu_t = 0$.

- Eine Zeitreihe $\{x_t\}$ heißt streng stationär, wenn gilt

$$P(x_{t_1} \leq c_1, \ldots, x_{t_k} \leq c_k) = P(x_{t_1+h} \leq c_1, \ldots, x_{t_k+h} \leq c_k)$$

  für alle $k = 1, 2, \ldots$, alle Zeitpunkte $t_1, \ldots, t_k$, alle Zahlen $c_1, \ldots, c_k$ und alle Verschiebungen $h = 0, -1, 1, -2, 2, \ldots$.

- Aus strengen Stationarität folgt direkt, dass alle existierenden Momente der Verteilungen der x_t für alle t identisch sind.

- Eine etwas mildere Version dieser Bedingungen führt zur sogenannten schwachen Stationarität. Hier werden nur Bedingungen an die ersten und zweiten Momente gestellt.

- Eine Zeitreihe heißt schwach stationär (oder einfach stationär), wenn gilt

  1. Es gibt ein μ mit $E(X_t) = \mu$ für alle t und
  2. die Autokovarianzfunktion $\gamma(s,t)$ hängt nur über $|s-t|$ von s und t ab und man schreibt $\gamma(h) := \gamma(|s-t|)$ mit $h = |s-t|$.

- Die normierte Autokovarianzfunktion $\rho(h) = \gamma(h)/\gamma(0)$ heißt Autokorrelationsfunktion ACF der Zeitreihe $\{x_t\}$.

- Der Backshift-Operator B ist definiert als $Bx_t = x_{t-1}$. Analog ist $B^k x_t = x_{t-k}$.

- Der Differenzenoperator ∇ ist definert als $\nabla x_t = x_t - x_{t-1}$.

- ∇x_t bezeichnet die sogenannte erste Differenz der Zeitreihe oder Differenz zum lag 1.

- Damit kann man schreiben $\nabla x_t = (1 - B)x_t$.

- Die $k-$te Differenz einer Zeitreihe ist definert als $\nabla^k = (1 - B)^k x_t$.

- Rechen Sie bitte nach, dass gilt:

$$\nabla^2 x_t = x_t - 2x_{t-1} + x_{t-2}!$$

Grundlegende Prozesse: White Noise

- Eine Zeitreihe $\{x_t\}$ für die gilt

$$x_t = \epsilon_t \text{ mit} \epsilon_t \sim N(0, \sigma^2) \text{ für alle} t$$

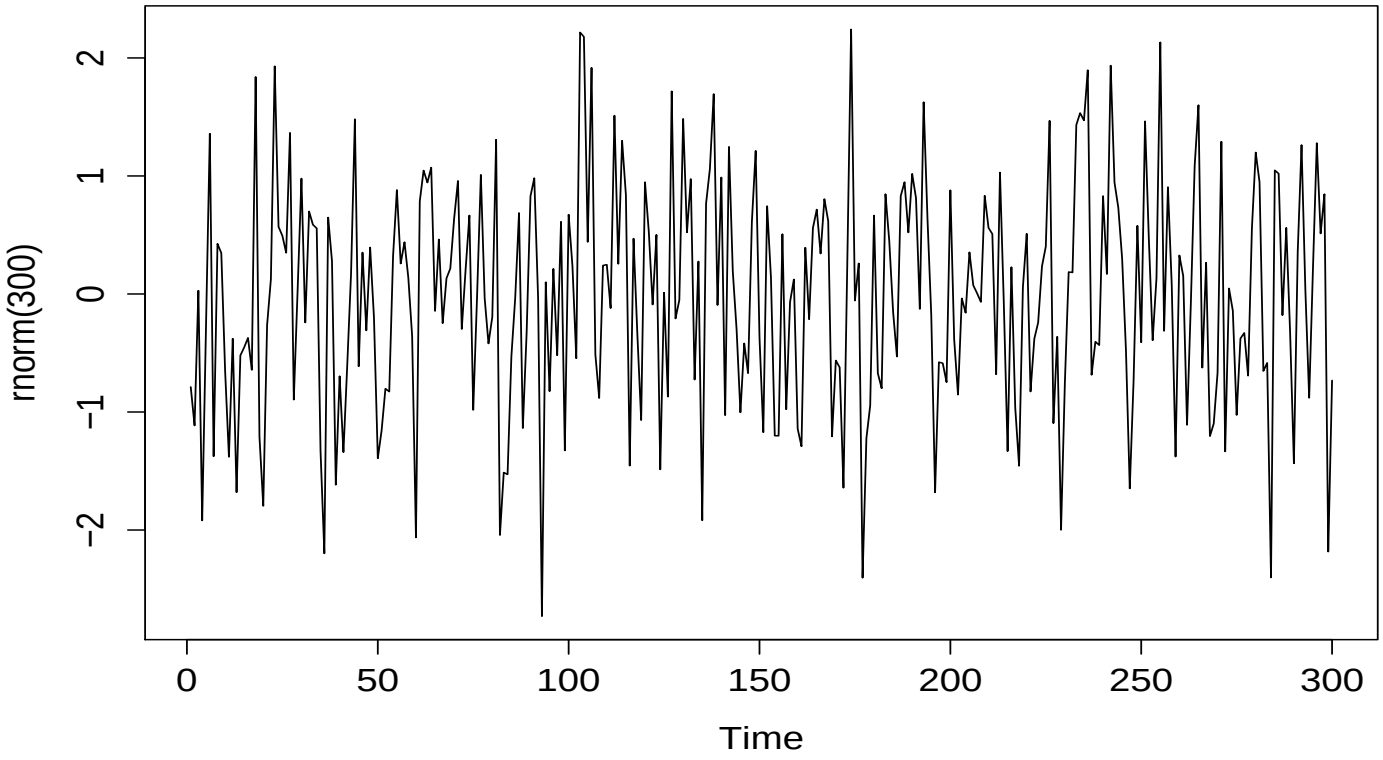
  heißt *weisses Rauschen* bzw. *white noise*.

- White Noise ist der einfachste Prozess und hat insbesondere in der Theorie Bedeutung.

- Ein White Noise Prozess sollte idealerweise nach Enfernung der modellierten Anteile übrig bleiben.

- Deshalb sind die Eigenschaften eines solchen Prozesses von Bedeutung.

White Noise



Grundlegende Prozesse: Der *Random Walk*

- Ein zweiter wichtiger Prozess ist der sog. *random walk*.
- Eine Zeitreihe $\{x_t\}$ für die gilt

$$x_t = \delta + x_{t-1} + \epsilon_t$$

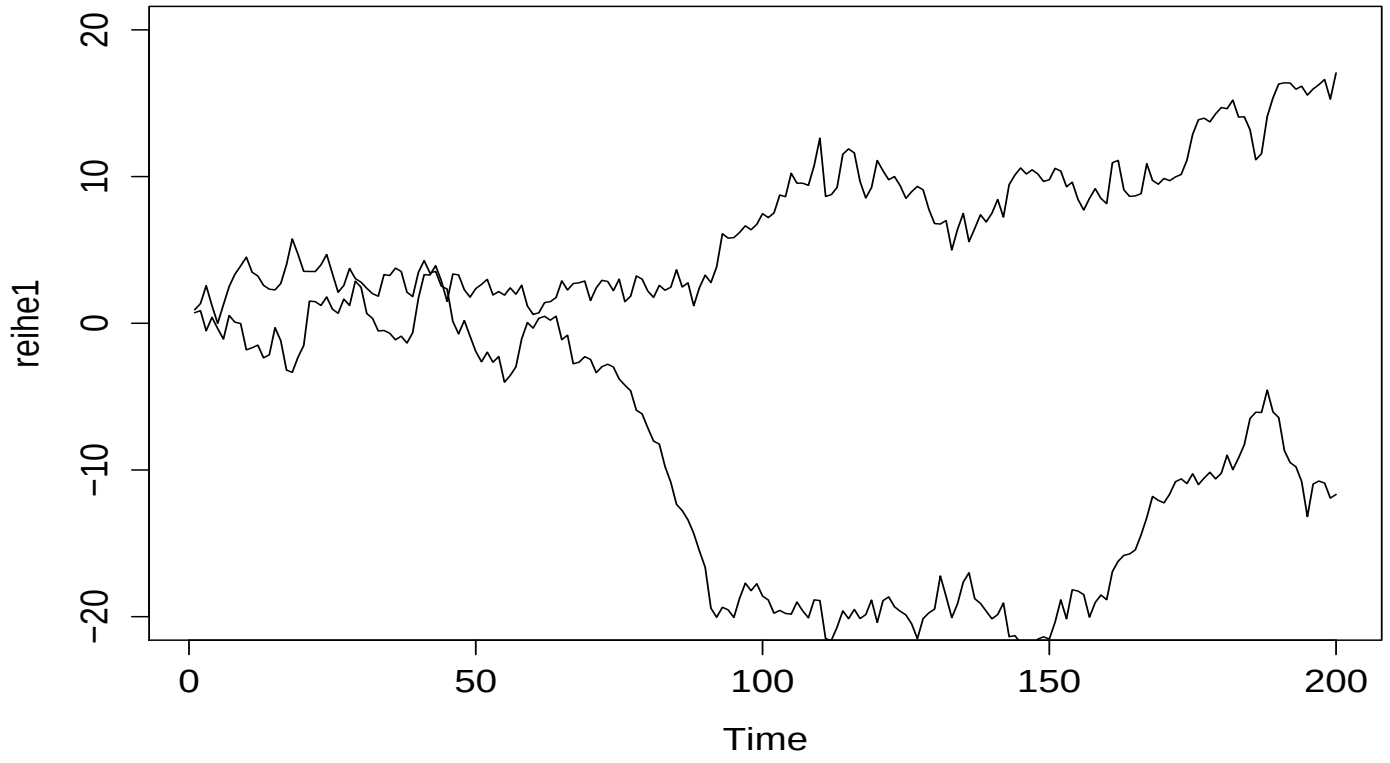
mit $x_0 = 0$ und $\epsilon_t \sim N(0, \sigma^2)$ für alle t heißt *random walk* mit *drift* δ. Ist $\delta = 0$ heißt $\{x_t\}$ einfach *random walk*.

- Ein *random walk* läßt sich als kumulierte Summe von weißem Rauschen darstellen.

$$x_t = \delta t + \sum_{1}^{t} \epsilon_i \text{ für } t = 1, 2, \ldots$$

Beispiele für den Random Walk

Der obere Verlauf ist mit $\delta = 0.1$, der untere mit $\delta = 0$.



AR und MA Prozesse

- Achtung: Im Folgenden sind die Prozesse stets als stationär mit Mittelwert 0 vorausgesetzt.

- Ein autoregressiver Prozess der Ordnung p, AR(p), besitzt die Form

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t,$$

  wobei $\phi_1, \ldots, \phi_p$ Konstanten sind und $\phi_p \neq 0$. Die ϵ_t sind i.i.d. $N(0, \sigma^2)$.

- Die formale Ähnlichkeit zum gewöhnlichen Regressionsmodell ist offenkundig. Mathematisch entstehen aber einige Schwierigkeiten aus der Zufälligkeit der Regressoren $x_{t-1}, \ldots, x_{t-p}$.

- Eine nützliche Schreibweise für den AR(p) ergibt die Umformulierung $(1 - \phi_1 B - \phi_1 B^2 - \cdots \phi_p B^p) x_t = \epsilon$.

- Etliche Eigenschaften eines $AR(p)$ Prozesses lassen sich aus den Eigenschaften dieses sogenannten autoregressiven Operators

$$\phi(B) := 1 - \phi_1 B - \phi_2 B^2 - \cdots \phi_p B^p$$

herleiten.

- Für einen AR(1) Prozess x_t ergibt sich durch iteriertes Einsetzen, dass

$$\begin{aligned} x_t &= \phi x_{t-1} + \epsilon_t \\ &= \phi(\phi x_{t-2} + \epsilon_{t-1}) + \epsilon_t \\ &\vdots \\ &= \phi^k x_{t-k} + \sum_{i=0}^{k-1} \phi^i \epsilon_{t-i}. \end{aligned}$$

- Ist $|\phi| < 1$ und $\{x_t\}$ stationär, so folgt

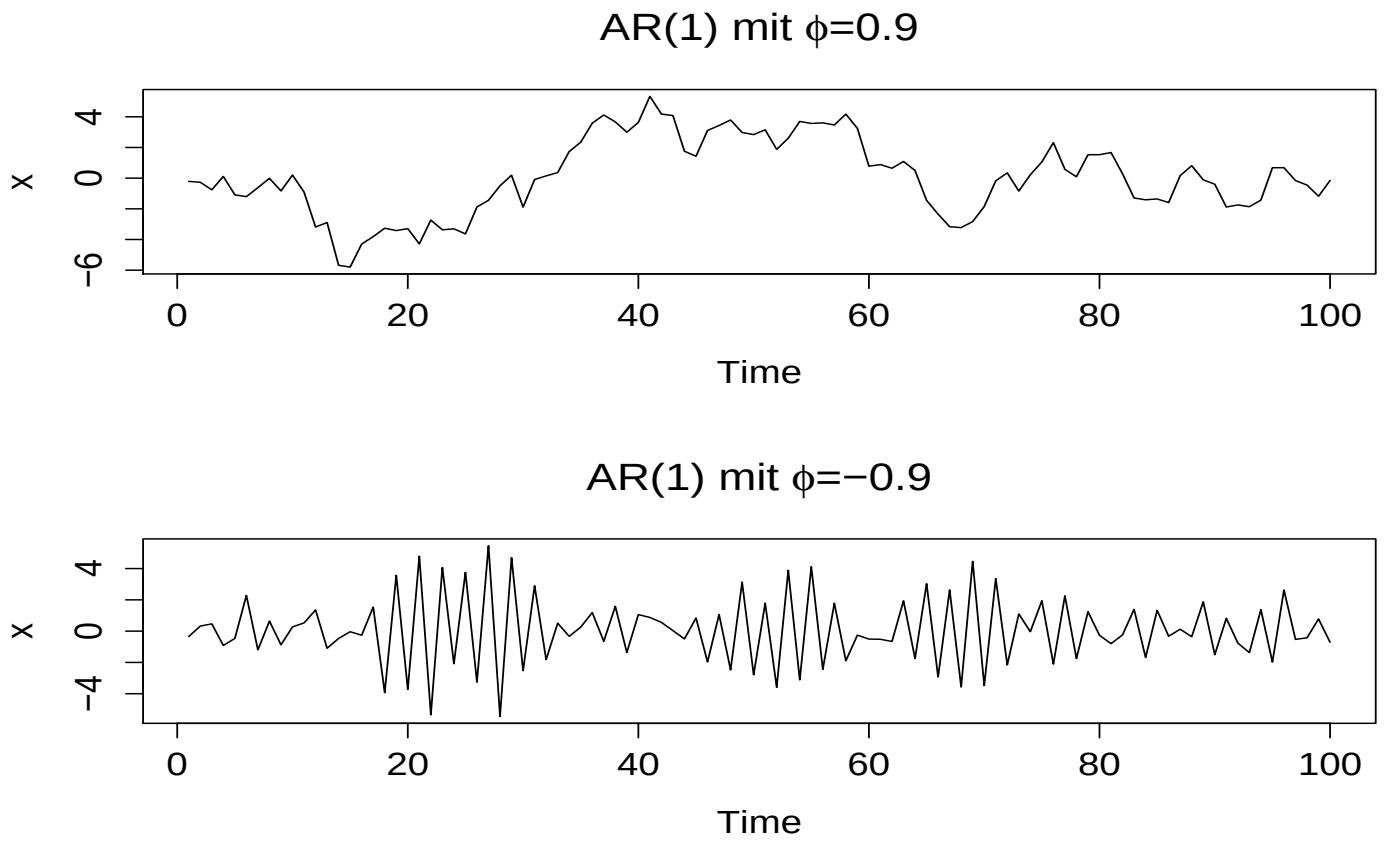
$$x_t = \sum_{i=0}^{\infty} \phi^i \epsilon_{t-i}.$$

- Ein solcher AR(1) Prozess ist folglich stationär mit $E(x_t) = 0$ und Autokovarianzfunktion $\gamma(h) = \frac{\sigma \phi^h}{1-\phi^2}$ für $h > 0$.

- Das Vorzeichen von ϕ hat große Bedeutung für die Gestalt eines solchen Prozesses.

- Die Idee von AR(p) Prozessen ist, durch eine Regression auf die Vergangenheit die Gegenwart zu modellieren.

Beispiele für AR(1) Prozesse



MA Prozesse

- Ein Moving Average Modell der Ordnung q, MA(q), ist definiert als

$$x_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q},$$

  wobei die $\theta_q \neq 0$ und alle $\epsilon_t \sim N(0, \sigma^2)$.

- Wenn $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$ den sogenannten MA-Operator bezeichnet, dann lässt sich der Prozess schreiben als

$$x_t = \theta(B)\epsilon_t.$$

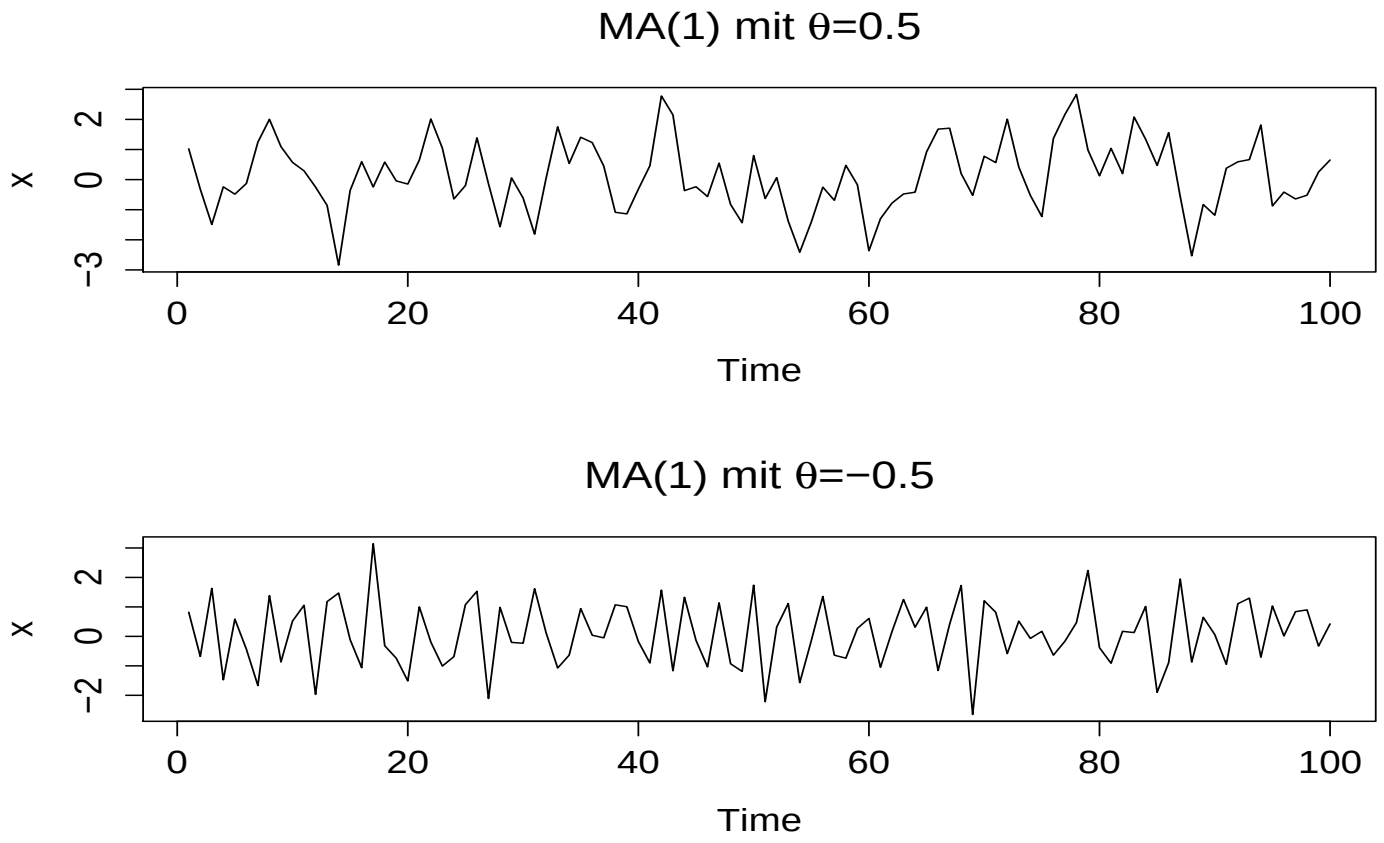
- MA-Prozesse sind für alle Parameterwahlen stationär.

- Für einen MA(1) Prozess $\{x_t\}$ gilt:

$$\gamma(h) = \begin{cases} (1+\theta^2)\sigma^2, & h = 0 \\ \theta\sigma^2, & h = 1 \\ 0, & h > 1 \end{cases}$$

- MA(q) Prozesse modellieren die Zeitreihe als Mittel der Störterme.

Beispiele für MA(1) Prozesse



Aufgabe zur Zeitreihenglättung

Sie finden auf der Webseite von Shumway und Stoffer den Datensatz `globtemp.dat`.

Passen Sie einen Ihrer Meinung nach passenden Trend bzw. Saisonfigur mittels gleitender Mittel an!

Passen Sie ein globales Modell mit linearem Trend an!

Vergleichen Sie die beiden Modelle!

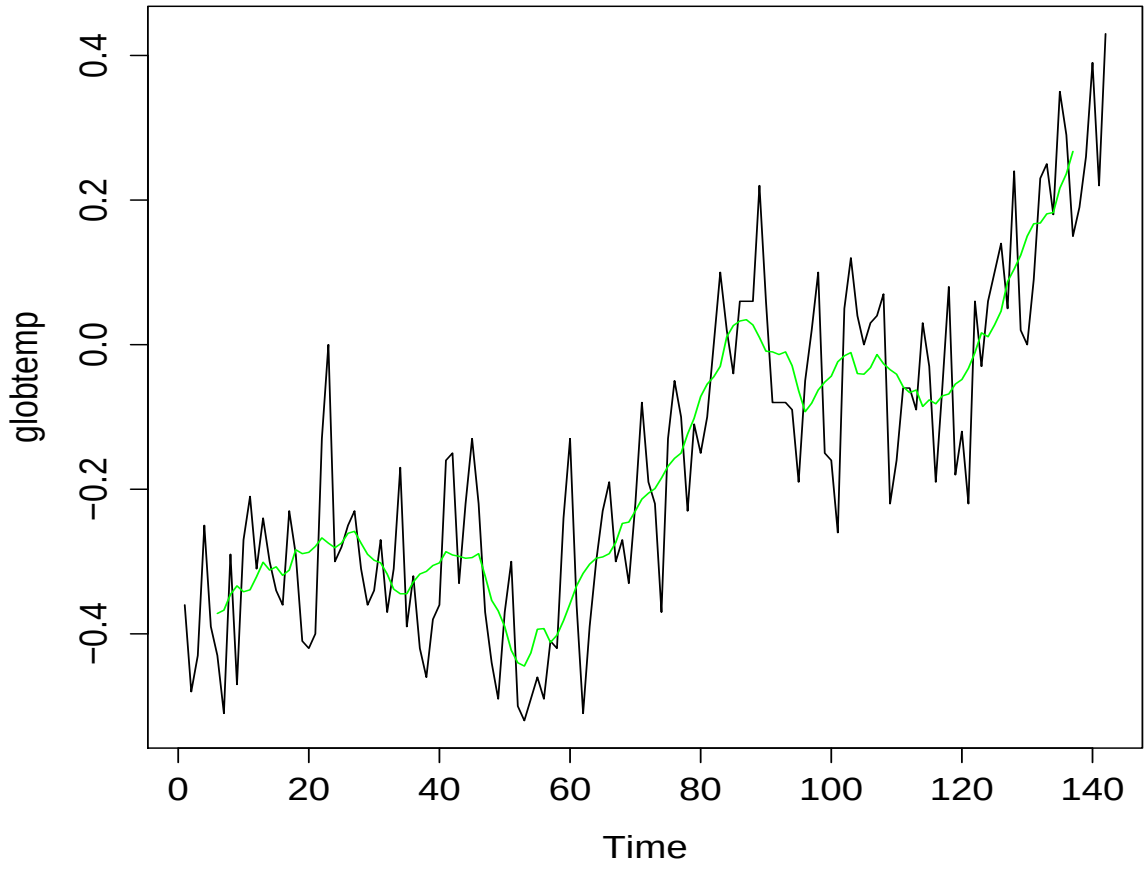
Lösung

- Der Datensatz gibt die Abweichungen in Grad Celsius von einem langjährigen Mittel an.
- Ein Trend soll die langfristige Tendenz in den Daten widerspiegeln.
- Die Modellierung soll über gleitende Mittel erfolgen.
- Zu beachten ist die ungerade Fensterbreite, an der man, ohne triftigen Grund abzuweichen, festhalten sollte.

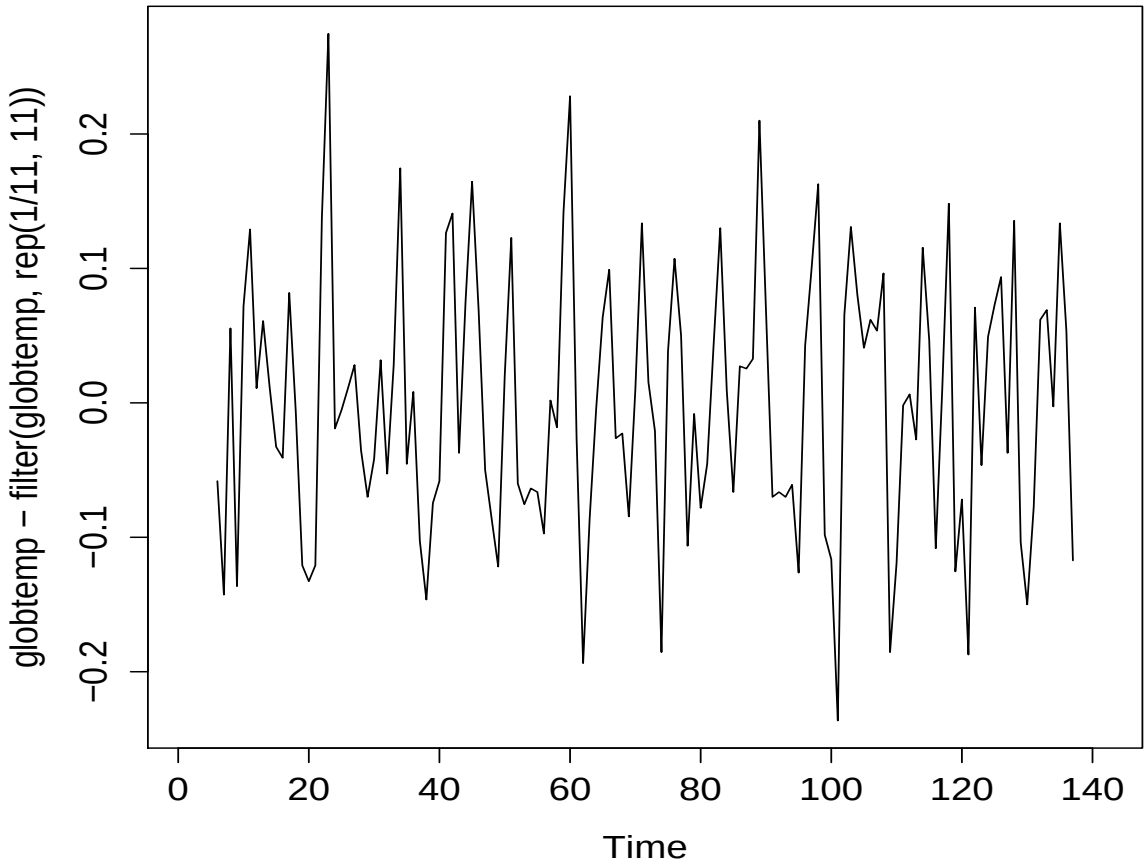
Lösung in R

```
globtemp <- ts(scan("TSA/data/globtemp.dat"))
plot(globtemp)
lines(filter(globtemp, rep(1/7,7)), col="red")
## etwas unruhig für einen Trend
lines(filter(globtemp, rep(1/11,11)), col="green")
## ganz ok
lines(filter(globtemp, rep(1/21,21)), col="blue")
## auch gut
plot(globtemp-filter(globtemp, rep(1/11,11)))
```

Trendfigur



Reste


globtemp – filter(globtemp, rep(1/11, 11))
−0.2
−0.1
0.0
0.1
0.2
0
20
40
60
80
100
120
140
Time

Linearer Trend

- Es sollte ein Vergleich mit einem globalen linearen Trend durchgeführt werden.

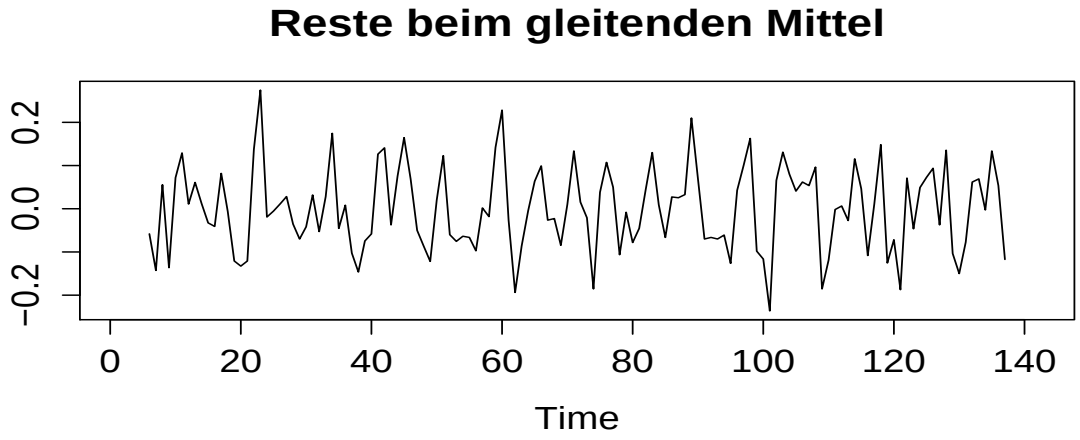
- `plot(globtemp)`
  `lines(ts(lm(globtemp ~ seq(1, length(globtemp) ) )$fit ) )`

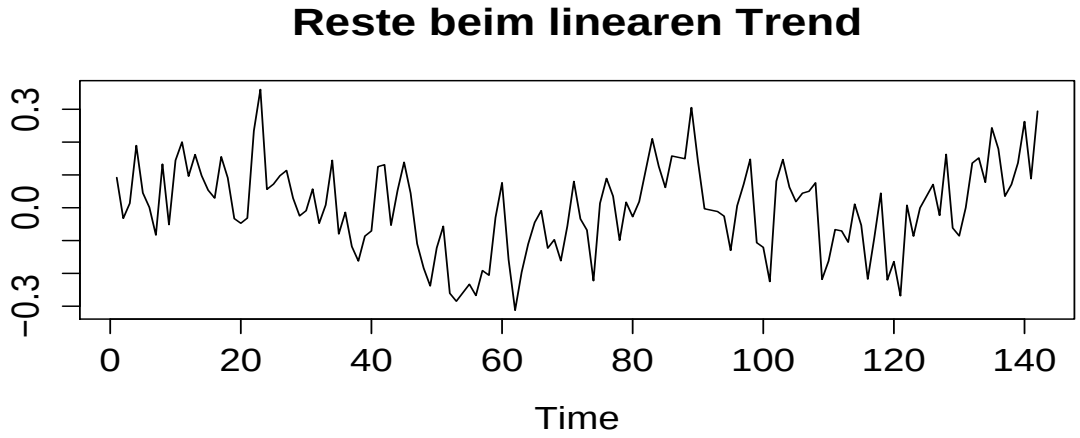
- Vergleicht man die Reste, so kann man die größere Flexibilität der gleitenden Mittel gut erkennen.

- Die modellbasierte lineare Trendanpassung kann die Schwankungen, denen die Temperaturkurve unterworfen ist nicht global gut anpassen.

- Bei den Resten aus dem linearen Trend sind ganz klar systematische Schwankungen in den Resten erkennbar.

Vergleich der Reste





ARMA(p,q) Prozesse

- ARMA(p, q) Prozesse kombinieren auf natürliche Weise AR und MA Prozesse.
- Eine Zeitreihe $\{x_t\}$ heißt ARMA(p, q), wenn sie stationär ist und gilt

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots \theta_q \epsilon_{t-q},$$

  wobei $\phi_p \neq 0, \theta_q \neq 0$ und $\epsilon_t \sim N(0, \sigma^2)$ für alle t.
- Man kann auch schreiben

$$\phi(B) x_t = \theta(B) \epsilon_t.$$

Nicht-Eindeutigkeit der Darstellung eines ARMA(p, q)

- Die Anzahl der Parameter in einer ARMA Darstellung ist nicht eindeutig. Beispielsweise ist

$$x_t = 0.5x_{t-1} - 0.5\epsilon_{t-1} + \epsilon_t$$

  nur eine Umformulierung des White Noise Prozesses, erscheint aber als ARMA(1,1) Prozess.

- Dies ist ein erhebliches Problem bei der Schätzung der Parameter eines ARMA Prozesses.

- AR und MA Prozesse ergeben sich als Spezialfälle von ARMA Prozessen, bei denen jeweils ein Operator gleich Null ist.

Bestimmen der Anzahl der Parameter eines ARMA(p, q)

- Zu einem gewissen Grad ist dies grafisch über ACF und PACF möglich.
- Die PACF (partial autocorrelation function) beseitigt die linearen Abhängigkeiten bis zu einem vorgebenen *lag*.
- Für die PACF ϕ_{hh} eines stationären Prozesses x_t zum *lag* h mit normalverteilten Fehlern ϵ_t gilt

$$\phi_{11} = \rho(1) \text{ und } \phi_{hh} = corr(x_t, x_{t-h}|x_{t-1}, \ldots, x_{t-(h-1)}).$$

- Die partielle Korrelation ϕ_{hh} gibt also den linearen Zusammenhang zwischen x_t und x_{t-h} wieder, nachdem die Einflüsse der dazwischenliegenden Zeitpunkte herausgerechnet wurden.

Verhalten von ACF und PACF

- Die Tabelle gibt die prinzipielle Gestalt von ACF und PACF für die verschiedenen Prozesse an.

	AR(p)	MA(q)	ARMA(p, q)
ACF	läuft aus	abgeschn. nach lag q	läuft aus
PACF	abgeschn. nach lag p	läuft aus	läuft aus

- In `R` kann die ACF mit der Funktion `acf()` berechnet werden, die PACF mit `pacf()`.

- Zur Zeitreihenanalyse gibt es in `R` sehr viel Funktionen. Darunter verschiedene Methoden der Parameterschätzung, der Simulation von Zeitreihen mit vorgegebenen Parametern etc.

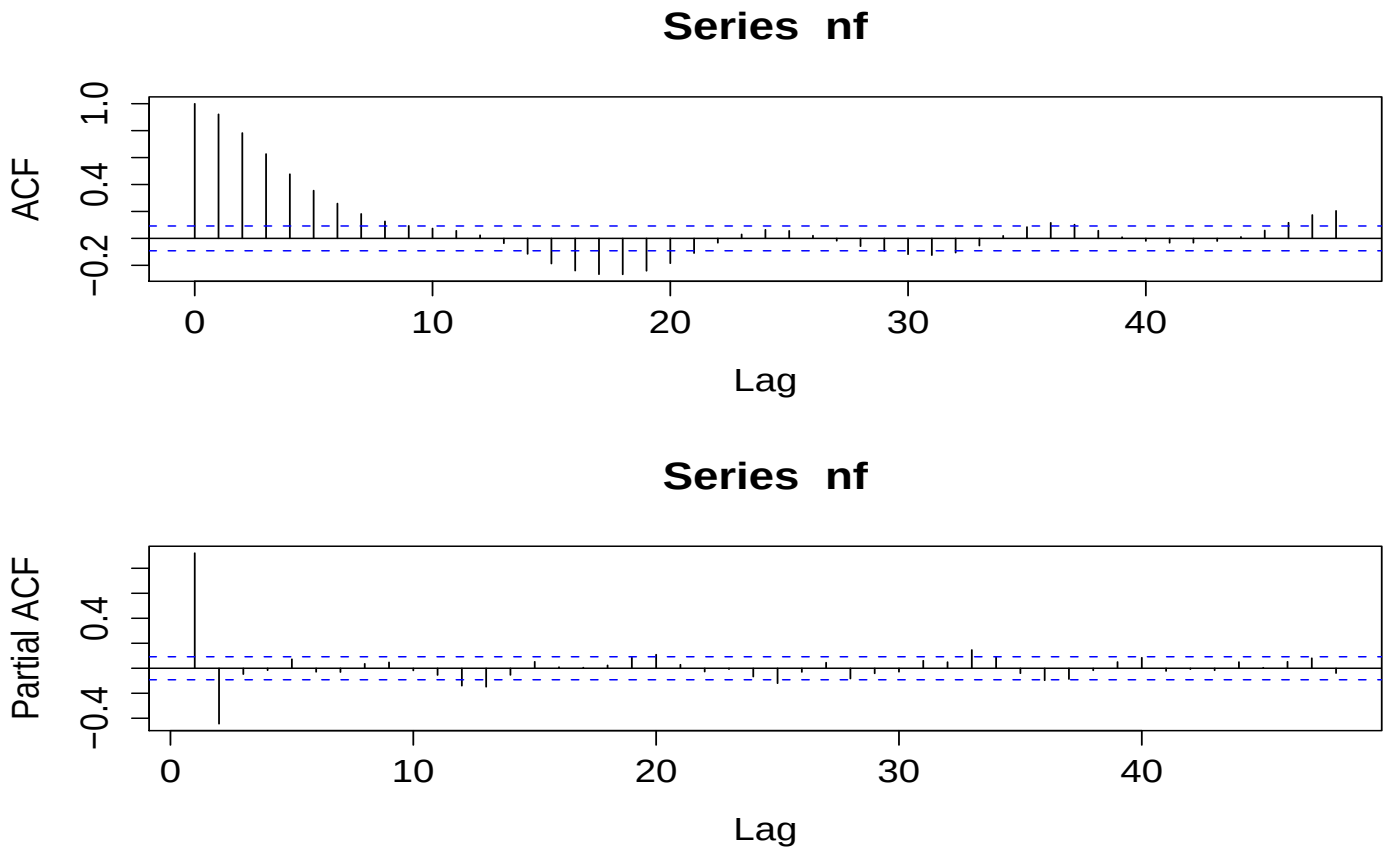
Beispiel zu ARMA

- Daten über 453 Monate von Anwerbungszahlen in `recruit.dat`.

- Die Anzahl der Parameter kann evtl aus ACF und PACF abgelesen werden.

```
nf <- scan("TSA/data/recruit.dat" )
par(mfrow=c(2,1))
acf(nf, 48)
pacf(nf, 48)
par(mfrow=c(1,1))
```

ACF und PACF für recruit.dat



Parameterschätzung in R

- Die Grafen sind kompatibel zu den Einträgen in der Tabelle für einen AR(2) Prozess.

- Schätzung der Parameter in R (z.B.) mit `ar.ols()`

```
ar.ols(nf, aic=FALSE, order.max=2, demean=TRUE, intercept=F)
Call:
ar.ols(x = nf, aic = FALSE, order.max = 2, demean = F,
      intercept = F)
Coefficients:
      1        2
 1.3971  -0.4157
Order selected 2  sigma^2 estimated as  97.04
```

- Der Prozess, der so angepasst würde, wäre also

$$x_t = 1.3971 x_{t-1} - 0.4157 x_{t-2}.$$

Ausblick

- Die Theorie zur Schätzung in diesen Modelle ist sehr ausgefeilt, hier kann nur das prinzipielle Vorgehen angeschnitten werden.

- Insbesondere wurden die Prozesse hier nur auf theoretischer Ebene diskutiert und der die Schätzung mittels realisierter Zeitreihen überhaupt nicht behandelt.

- Das Vorgehen und die prinzipielle Problematik sollten aber deutlich geworden sein.

Aufgabe zu ARMA Prozessen

- Bestimmen Sie die Parameter p, q eines angemessenen ARMA(p, q) Prozesses für die Mortalitätsdaten aus cmort.dat von Shumway und Stoffer.

- Zeichnen Sie dazu die ACF und die PACF der Zeitreihe.

- Schätzen Sie die Koeffizienten des Modells.

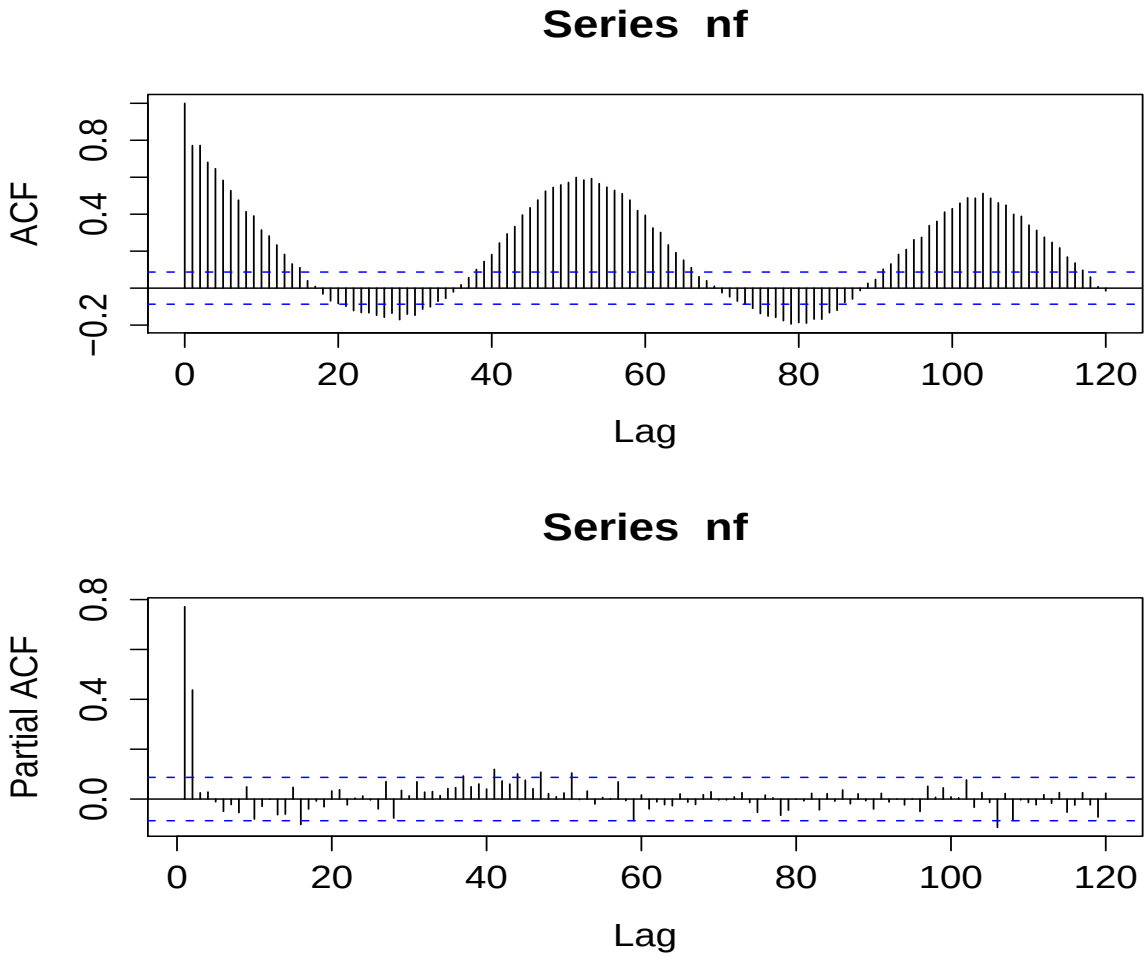
Lösung

```
nf <- scan("TSA/data/cmort.dat" )

par(mfrow=c(2,1))
acf(nf, 120)
pacf(nf, 120)
par(mfrow=c(1,1))

ar.ols(nf, order.max=2, demean=TRUE, intercept=FALSE)
```

Grafiken ACF / PACF



Koeffizientenschätzung

```
> ar.ols(nf, order.max=2, demean=TRUE, intercept=FALSE)
Call:
ar.ols(x = nf, order.max = 2, demean = TRUE, intercept = FALSE)
Coefficients:
     1       2
0.4286  0.4418
Order selected 2  sigma^2 estimated as  32.32
```

ANOVA Beispiel

- Im Datensatz `anovaexample.txt` finden Sie die Daten von 45 Opfern schwerer Verbrechen, die zufällig einer von vier psychologischen Behandlungsmethoden unterzogen wurden. (Das Original:Foa, E. B., Rothbaum, B. O., Riggs, D. S., & Murdock, T. B. Treatment of posttraumatic stress disorder in rape victims: A comparison between cognitive-behavioral procedures and counseling. Journal of Consulting and Clinical Psychology, 59, 715-723.)

- Als Zielgröße wird hier nur die Anzahl von bestimmten Symptomen betrachtet.

- Untersuchen sie diese Frage mit einer ANOVA.

- Mit welcher grafischen Darstellung könnten Sie überprüfen, welche Behandlungspaare evtl. signifikant verschieden sind? Welches Paar sind Ihrer Meinung nach der Hauptkandidat?

- Wie würden Sie die Signifikanz der Abweichungen testen?

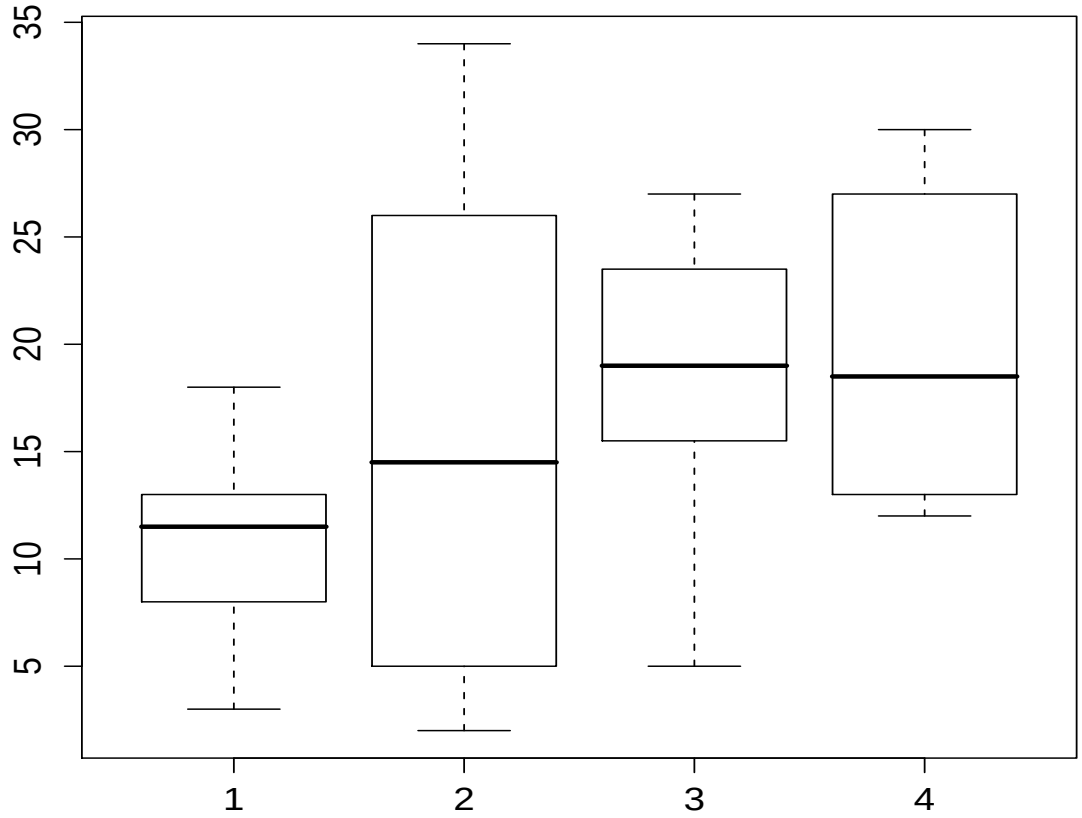
- Welche Modellverletzung für das Verfahren aus der vorhergehenden Teilaufgabe können Sie aus dem Plot ablesen?

Lösung R-code

```
anovadata <- read.table(file="anovaexample.txt",
                        sep=";", header=TRUE)
anovadata[,2] <- as.factor(anovadata[,2])

summary(aov(lm(anovadata[,3] ~ anovadata[,2])))
```

Boxplot



Inhaltliche Fragen

- Gruppe 1 und Gruppe 3 scheinen die geringste Überlappung zu haben.
- Die Hypothese könnte mit einem entsprechenden Zwei-Stichproben-t-Test überprüft werden.
- Die nicht erfüllte Voraussetzung ist offensichtlich die Homoskedastizität.

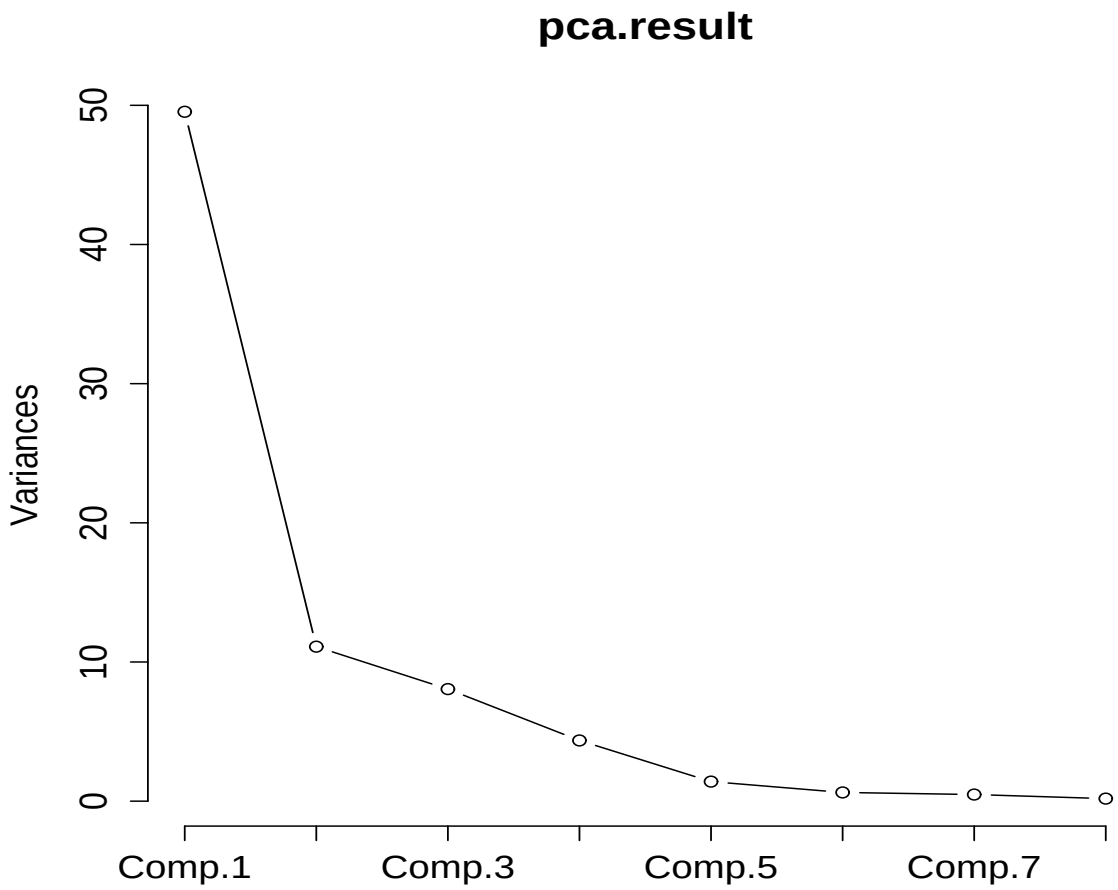
Beispiel Hauptkomponentenanalyse

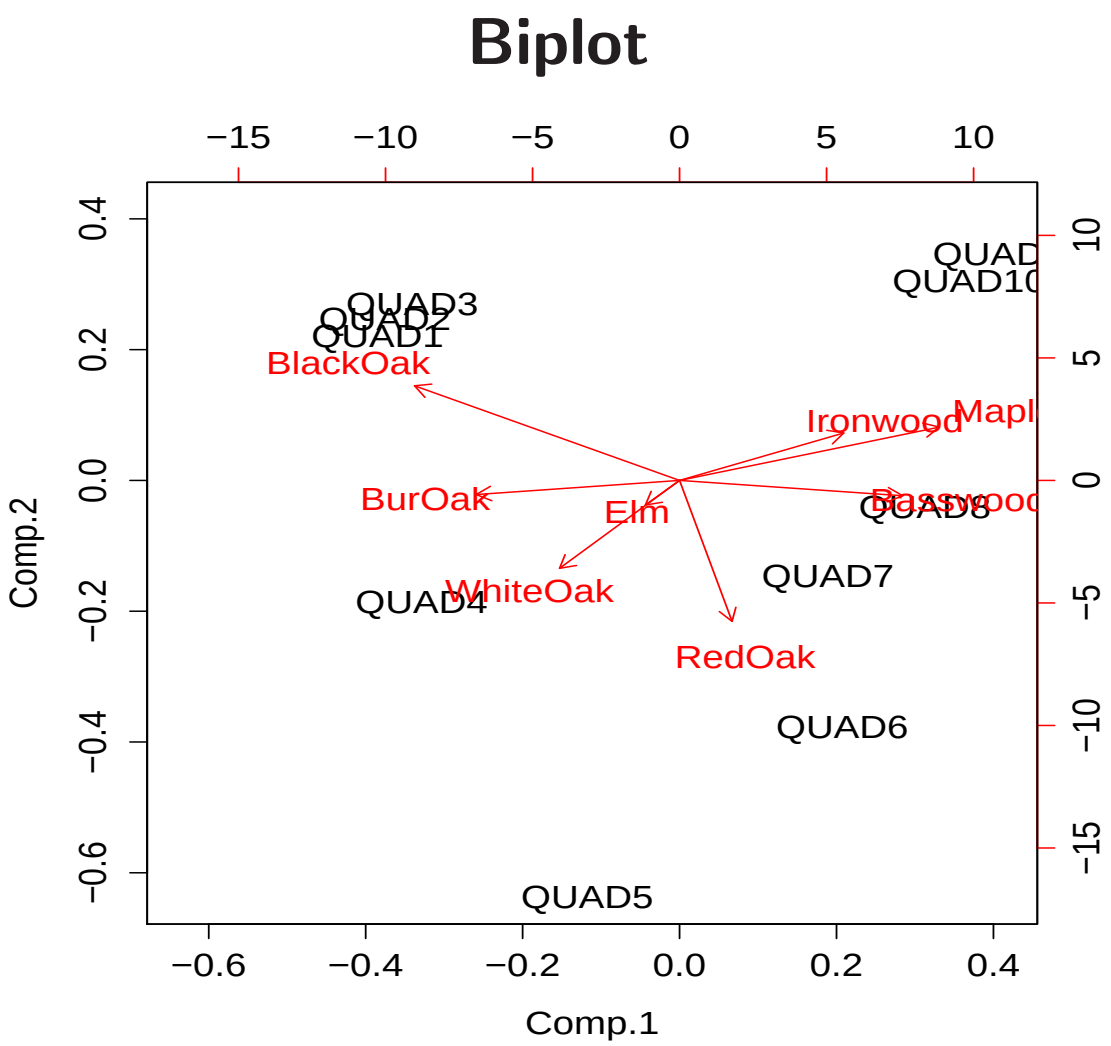
- Beispiel für Entwicklungsstufen von Waldböden.
- Gezählt wurden über einen sehr langen Zeitraum die Anzahl verschiedener Bäume in einem Waldgebiet. Zu jedem Zeitpunkt befand sich der Wald in einem vom Fachmann spezifizierten Entwicklungsstand.
- Die Daten finden sich in der Datei `pcaexample.txt`.
- Wie viele Hauptkomponenten sind Ihrer Meinung nach wichtig?
- Wenn man sich nun die Koordinaten der Entwicklungsstufen in den ersten beiden Hauptkomponenten anschaut, finden sich dort Gruppen von Stufen?

Lösung R-Code

```
pcadata <- read.table(file="pcaexample.txt",
                      sep=",", header=TRUE, row.names=1)
pca.result <-
  princomp(~Basswood+BlackOak+BurOak+Elm+Ironwood+
                    Maple+RedOak+WhiteOak, data=pcadata)
summary(pca.result)
plot(pca.result, type="lines")
pca.result$loadings
biplot(pca.result)
```

Screeplot




Biplot
−15
−10
−5
0
5
10
0.4
0.2
0.0
−0.2
−0.4
−0.6
10
5
0
−5
−10
−15
Comp.2
Comp.1
−0.6
−0.4
−0.2
0.0
0.2
0.4
QUAD
QUAD10
QUAD3
QUAD2
QUAD1
BlackOak
Ironwood
Mapl
BurOak
Elm
QUAD8
WhiteOak
QUAD4
QUAD7
RedOak
QUAD6
QUAD5

Inhaltliche Fragen

- Es sind höchstens 4 Hauptkomponenten nötig. Dies wird sowohl von dem kumulierten Varianzanteil, 96% bis zur 4. Komponente, als auch vom Screeplot gestützt. Um sich auf eine Komponente zu beschränken, erklärt diese einen zu geringen Varianzanteil.

- Im Biplot bilden QUAD 1-3 und Quad 9-10 jeweils einen Cluster. Es könnte also ausreichen sich auf 7 Entwicklungsstufen zurückzuziehen.