Datenanalyse

Vorlesungsskriptum

Univ.-Prof. Dipl.-Ing. Dr.techn.
Peter Filzmoser

Institut für Stochastik und Wirtschaftsmathematik

Technische Universität Wien

Wien, im März 2021

Vorwort

Ein erster Schritt in der Datenanalyse sollte ein explorativer Zugang sein, in dem man versucht, auf eher informale Weise Strukturen in den Daten zu erkennen. In erster Linie wird dieser Zugang durch entsprechende grafische Aufbereitung der Daten unterstützt. Grafische Verfahren haben eine lange Tradition in der beschreibenden Statistik. Histogramme z.B. zählen zu den ältesten in der angewandten Statistik verwendeten Verfahren. Welche grafischen Verfahren für eine bestimmte Analyse am besten geeignet sind, hängt auch von den zu analysierenden Daten ab. Umfang der Stichprobe und Dimension der Daten spielen dabei eine entscheidende Rolle.

Probleme bereitet oft die Dimension der Daten. Zwar kann man univariate Verfahren auf jede einzelne Variable anwenden und gewisse Teilinformation gewinnen, dabei gehen jedoch Zusammenhänge verloren, die man erst durch eine multivariate Betrachtensweise erhält. Während grafische Verfahren zur Darstellung ein- und zweidimensionaler Daten praktisch zu den Standardmethoden zählen, sind grafischen Verfahren für multivariate Daten methodisch wesentlich komplexer. Ein tieferes Verständnis für solche Methoden ist essentiell um die Resultate verstehen und interpretieren zu können.

Statistische Entscheidungen basieren auf Beobachtungen oder werden aufgrund spezieller Voraussetzungen (Zufälligkeit, Unabhängigkeit, Normalverteilung, usw.) getroffen. Gerade diese Voraussetzungen sind es, unter denen einerseits die klassischen Methoden funktionieren und andererseits die mathematisch einfache Handhabung ermöglicht wird. Geringfügige Abweichungen führen schon meist zu Fehlschlüssen. In der Praxis sind aber diese idealen Voraussetzungen eher selten erfüllt.

Ein Ziel der Datenanalyse ist auch, Prozeduren zu finden, die gegenüber solchen Abweichungen resistent sind. Es sollen also kleine Abweichungen vom Modell auch nur geringe Auswirkungen haben. Es gibt nun verschiedene Methoden dies zu verwirklichen. Einige davon werden in dieser Vorlesung behandelt, sodass ein Überblick über die gebräuchlichsten Methoden vermittelt wird.

Einer der Wegbereiter der explorativen Datenanalyse war John W. Tukey, der es verstanden hat, auf einfache Weise effiziente Darstellungen und Verfahren zur Analyse statistischer Daten zu entwickeln. Obwohl der Computer erst nach diesen Entwicklungen verstärkt für die Datenanalyse eingesetzt wurde, sind viele dieser Verfahren nach wie vor sehr beliebt, wie z.B. die Boxplots. Die entsprechende Aufbereitung von Daten ist ganz wensentlich: *"It is important to understand what you can do before you learn to measure how well you seem to have done it."* (Tukey, 1977)

Die heute auftretenden Unmengen von Daten haben die Welt der Statistik stark beeinflusst. Für viele Probleme gibt es keine geschlossene mathematische Lösung, wodurch numerische Algorithmen zur Lösungsfindung immer bedeutender werden. Generell hat der Einsatz von effizienten Algorithmen stark an Bedeutung gewonnen, und die Informatik hat daher die

gleiche Wichtigkeit für moderne Statistik erreicht wie die Mathematik. Viele sogenannte Statistiker machen heute die gleiche Arbeit wie Informatiker: sie entwickeln Programmsysteme zur Analyse von großen Datenmengen. Für solche Systeme, und vor allem für Leute, die damit umgehen können und die "Kunst der Datenanalyse" verstehen, besteht mehr und mehr Bedarf in der Wirtschaft. In einem Artikel vom 6. August 2009 in der New York Times sagt der Chief Economist Hal Varian von Google: *"I keep saying that the sexy job in the next 10 years will be statisticians. And I'm not kidding."* Heute wissen wir, dass der Beruf "Data Scientist" enorm gefragt ist, und dass viele Unternehmen Schwierigkeiten haben, geeignete Personen zu finden.

In dieser Vorlesung wird auch auf die Umsetzbarkeit der erlernten Methoden gelegt, und mit Umsetzbarkeit ist natürlich der Einsatz des Computers gemeint. Als Statistik-Software wird hier *R* eingesetzt, siehe *http://www.R-project.org*, und zwar deshalb, weil heute die große Mehrheit der neu entwickelten statistischen Verfahren in *R* implementiert wird, und weil *R* weit über die Grenzen der Statistikwelt hinaus eine zentrale Bedeutung erlangt hat. Die meisten Grafiken im Skriptum wurden mit *R* erzeugt. In der Legende zu den Abbildungen stehen die entsprechenden *R*-Befehle `in gesperrter Schrift`.

Literatur zur Vorlesung

J.M. Chambers, W.S. Cleveland, B. Kleiner, and P.A. Tukey (1983). *Graphical Methods for Data Analysis*, Chapman and Hall, New York.

W.S. Cleveland (1987). *The Collected Works of John W. Tukey*, Volume V, Graphics 1965-1985, Chapman and Hall, New York.

W.S. Cleveland (1993). *Visualizing Data*, Hobart Press, Summit, New Jersey.

S.H.C. du Toit, A.G.W. Steyn, and R.H. Stumpf (1986). *Graphical Exploratory Data Analysis*, Springer, New York.

M. Friendly (2000). *Visualizing Categorical Data*, SAS Press, Cary, NC.

J.R. Gessler (1993). *Statistische Graphik*, Birkhäuser, Basel.

D.C. Hoaglin, F.M. Mosteller, and J.W. Tukey (1983). *Understanding Robust and Exploratory Data Analysis*, Wiley, New York.

R. Maronna, D. Martin, and V. Yohai (2006). *Robust Statistics. Theory and Methods*, John Wiley & Sons Canada Ltd., Toronto, ON.

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter (2008). *Statistical Data Analysis Explained. Applied Environmental Statistics with R*, John Wiley & Sons, Chichester.

J.W. Tukey (1977). *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachussets.

K. Varmuza and P. Filzmoser (2009). *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, Boca Raton, FL.

E.J. Wegman E.J. and D.J. DePriest (1986). *Statistical Image Processing and Graphics*, Marcel Dekker, New York.

Inhaltsverzeichnis

Kapitel 1

Stichprobendesign: vom Problem zur statistischen Lösung

1.1 Einleitung

"Sammle alle Information die du kriegen kannst – wir denken später darüber nach, was wir damit machen."

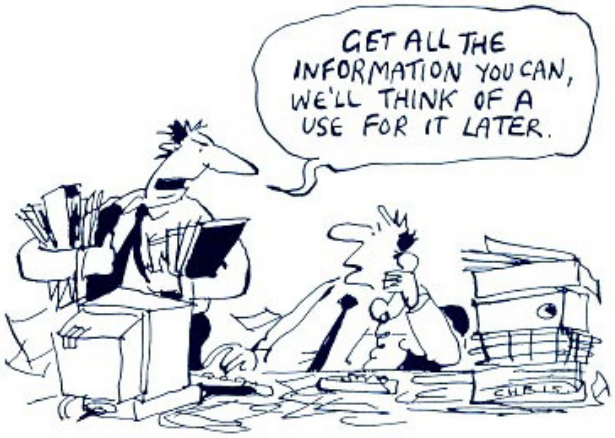


Abbildung 1.1: Daten, Daten, Daten, Daten, ...

Dieses Motto wäre nicht zielführend für sinnvolle statistische Aussagen, weil:

- Information sammeln kann kosten- und zeitintensiv sein. Viel Information sammeln kann noch kosten- und zeitintensiver werden.
- Bereits beim Sammeln der Information soll man die Problemstellung und die mögliche statistische Methodik im Hinterkopf haben (ergebnisorientiert sammeln).
- Auch hier gilt: Qualität vor Quantität
- Das ist genau das Ziel von Statistik, dass man auch mit wenig Information allgemeinere Aussagen treffen kann (Prognose).
- Daten müssen vergleichbar sein, um eine gemeinsame Analyse damit machen zu können. Im Laufe der Zeit könnten sich die Rahmenbedingungen verändern, womit die Information nicht mehr vergleichbar ist.

- Statistik hat nichts zu tun mit Detektivarbeit, wo man *irgendwelche* Anhaltspunkte sucht, sondern man möchte aufgrund von repräsentativen Beobachtungen zu allgemeingültigen Schlussfolgerungen kommen.

1.2 Grundbegriffe

Grundgesamtheit: Die Grundgesamtheit oder **Population** ist die Gesamtmenge aller statistischen Einheiten mit übereinstimmenden Identifikationskriterien. Beispiele für statistische Einheiten sind Personen, Haushalte, oder Ereignisse. Beispiele für Populationen sind alle Personen, die an einem Stichtag den Hauptwohnsitz in Österreich hatten.

Stichprobe: Eine Stichprobe ist eine Teilmenge einer Grundgesamtheit bei einer statistischen Untersuchung. Die Stichprobe soll so zusammengestellt werden, um bestimmte Eigenschaften der Gesamtpopulation zu untersuchen. Ist man an mittleren Haushaltseinkommen in Österreich interessiert, wird die Stichprobe aus einer Untermenge aller möglichen österreichischen Haushalte bestehen.

Mit Hilfe einer Stichprobe möchte man auf die Grundgesamtheit schließen. Diese Vorgangsweise wird deshalb gemacht, weil die Grundgesamtheit üblicherweise nicht beobachtet werden kann, oder weil es zu teuer oder zu aufwändig ist, alle Elemente der Grundgesamtheit zu "befragen" oder zu vermessen.

Stichprobendesign: Das Stichprobendesign beschreibt das Auswahlverfahren, wie (nach welchem Schema) die Stichprobe aus der Grundgesamtheit entnommen wird. Oberstes Prinzip ist, dass die Stichprobe *repräsentativ* sein muss, um auch gültige Schlussfolgerungen auf die Grundgesamtheit machen zu können.

Repräsentative Stichprobe: Eine repräsentative Stichprobe soll die Komplexität und Zusammensetzung der Grundgesamtheit wiederspiegeln. Ist man an einer Schätzung des mittleren Haushaltseinkommen interessiert, so muss die Stichprobe nach bestimmten Merkmalen (Alter, Geschlecht, Staatsbürgerschaft, Einkommenskomponente, etc.) die Struktur der Grundgesamtheit reflektieren. Abbildung 1.2 zeigt dies schematisch mit *Mosaikplots* (die Anzahl der Beobachtungen in jeder Kategorie ist proportional zur dargestellten Fläche): Die Struktur der Population (links) und jene der Stichprobe (rechts) stimmen sehr gut überein.

Zufalls- bzw. Quoten-Stichprobe: Darunter versteht man zwei Strategien (Stichprobendesigns), um eine Stichprobe zu erhalten. Bei der Zufallsstichprobe werden die Beobachtungen rein zufällig ausgewählt; bei der Quotenstichprobe wird nach einem bestimmten Schema vorgegangen, um bewusst gewisse Zielgruppen in der Stichprobe vertreten zu haben. Die Auswahl entsprechend Abbildung 1.2 kann nur eine Quotenstichprobe gewesen sein, weil mit rein zufälliger Selektion die verschiedenen Kategorien nicht so präzise dem Verhältnis der Grundgesamtheit entsprechen wird. Innerhalb einer Quotenstichprobe werden aber die Beobachtungen (entsprechend der Quoten) wiederum zufällig ausgewählt.

1.3 Planung der statistischen Datenerhebung

Zuerst müssen Problemstellung und Zielsetzung festgelegt werden. Zur Beantwortung der Forschungsfragen muss entschieden werden:

- wie die *Grundgesamtheit* definiert wird,

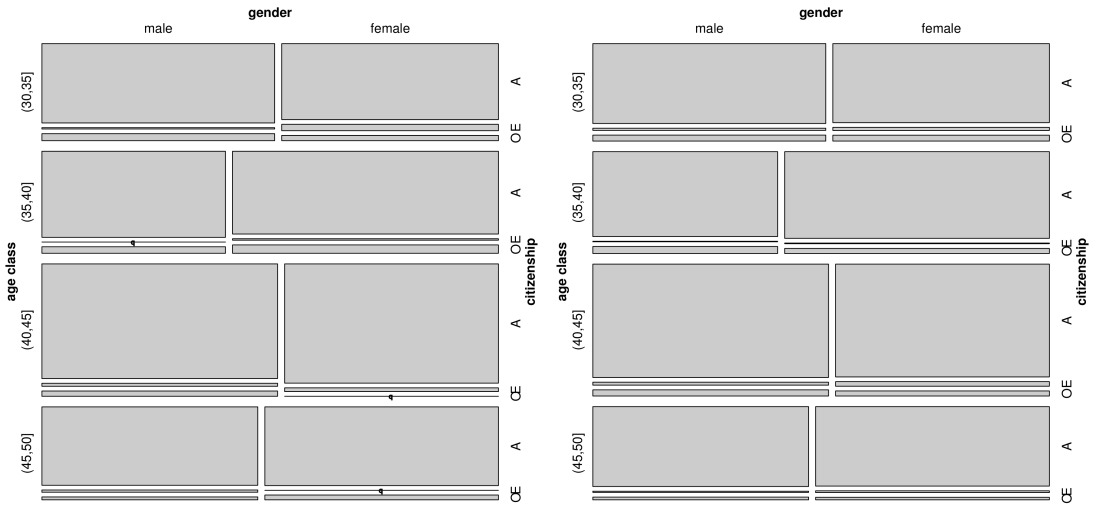


Abbildung 1.2: Population links, Stichprobe rechts.

- an welchen *statistischen Einheiten* gemessen werden soll,
- welche *Variablen* erhoben werden,
- wie die Information *gemessen* wird (Anzahl, Index),
- über die *Art* und den *Umfang* (meist gleichbedeutend mit Kosten) der Erhebung.

Beispiel: Eine online-Handelsfirma möchte die Werbewirksamkeit ihrer Produkte verbessern. Die Grundgesamtheit besteht aus allen Computern (IP-Adressen), von denen jemals auf ihre Produkte zugegriffen wurde. Die statistischen Einheiten an denen gemessen wird könnten alle Zugriffe von IP-Adressen sein, die in einem bestimmten Zeitintervall auf selektierte Produkte erfolgt sind. Die zu messenden Variablen könnten für diese Produkte die Anzahl der Verkäufe pro Zeitintervall und die Anzahl der Zugriffe auf die Produkte pro Zeitintervall sein. Falls Peronendaten zu den IP-Adressen bekannt sind, können weitere Variablen Geschlecht, Alter, Wohnbezirk, etc., sein. Als Werte für die Variablen erhält man hier direkt Zähldaten bzw. Kategorien, Zahlen und Kennzahlen für Geschlecht, Alter und Wohnbezirk, die direkt verarbeitet werden können. Die Art der Erhebung ist internet-basiert (und nicht etwa Befragungen), und der Umfang wird durch das gewählte Zeitintervall und die selektierten Produkte bestimmt.

1.4 Statistische Datenerhebung

Man unterscheidet drei Arten:

Primär-statistische Erhebung: Wir erheben die Daten selbst. Dies kann geschenen durch Umfragen, durch eigenes Beobachten und Messen, bzw. durch eigenes Aufzeichnen. Es sind hier alle früher erwähnten Punkte zu beachten, von der richtigen Planung bis zur Auswahl einer repräsentativen Stichprobe. Speziell bei Umfragen sind wichtige Dinge zu beachten, wie Gewährleistung der Anonymität (sonst Verzerrung möglich) oder Vermeidung von Beeinflussung durch den Interviewer.

Sekundär-statistische Erhebung: Wir verwenden existierende Datenbanken, wie z.B. Datenbasen von Statistik Austria oder EuroStat, Datenbanken von Banken, Versicherungen, Firmen, etc. Nachteile solcher Datenbanken sind, dass sie nicht genau im Kontext

zur Forschungsfrage stehen müssen, dass sie veraltet sein können, oder dass sie schlechte Datenqualität haben.

Tertiär-statistische Erhebung: Wir verwenden existierende aggregierte Daten. Z.B. sind Umsätze von Unternehmen meist nicht als Einzeldaten verfügbar, sondern nur in aggregierter (zusammengefasster) Form, wobei die Aggregation z.B. nach räumlichen Aspekten oder nach Branchen erfolgt sein kann.

1.5 Statistische Datenaufbereitung

Nachdem die Daten vorliegen, ist meist noch ein langer Weg bis zur Datenanalyse zu beschreiten, weil die Daten zuerst für eine sinnvolle Analyse vorbereitet werden müssen. Zu den möglichen Problemen gehören:

Kodierung: Es liegen nur kodierte Daten, aber keine Zahlenwerte vor, die verglichen werden können. Nicht immer ist eine Umwandlung in Zahlenwerte möglich oder sinnvoll; man würde dann eben mit "Faktorvariablen" arbeiten müssen.

Datenbereinigung: Die Daten können unsinnige Werte beinhalten, müssen daher auf *Plausibilität* geprüft und gegebenenfalls korrigiert werden. Daten können *Ausreißer* beinhalten, die entweder korrigiert werden können, oder deren Einfluss mit geeigneten *robusten statistischen Methoden* reduziert wird. *Fehlende Werte* sind problematisch für viele statistische Berechnungen. Wenn das Ergänzen dieser Werte nicht möglich ist (auch mittels statistischer Verfahren), müssen entsprechende Methoden verwendet werden, die mit *missings* umgehen können. *Transformationen* von Daten (Variablen) könnten nötig sein, bevor statistisch analysiert wird.

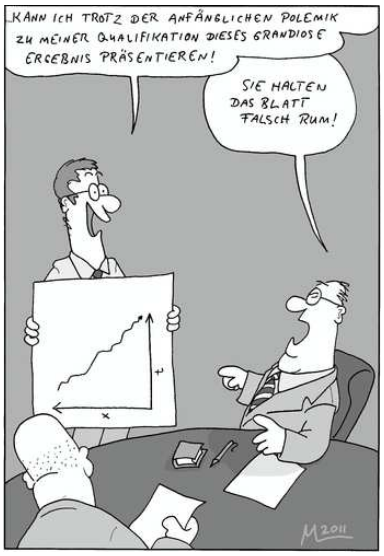
1.6 Statistische Datenanalyse

Diese muss sinnvollerweise mit geeigneter statistischer Software durchgeführt werden. Möchte man eine sehr umfangreiche Software die gratis zur Verfügung steht, so wird die Wahl auf `R` fallen (`http://cran.r-project.org`). Andere kommerzielle Produkte sind z.B. SAS, Statistica, SPSS, Stata, Eviews, Minitab.

Nur mit der Installation einer geeigneten Software ist die statistische Analyse noch nicht gemacht. Auch die Fähigkeit, auf "entsprechende Knöpfe drücken" zu können, muss noch nicht bedeuten, dass die Analyse sinnvoll und zweckmäßig ist. Ein tieferes Verständnis für die anzuwendenden Methoden ist unumgänglich!

1.7 Interpretation der Ergebnisse

Auch die Interpretation erfordert ein tieferes Verständnis der statistischen Methode, die zu diesem Ergebnis geführt hat. Insbesondere ist die Validität der Ergebnisse abzustimmen mit den Voraussetzungen und Einschränkungen der statistischen Methoden. Rigorose Interpretationen sollten immer von Fachleuten gemacht werden, die auch inhaltlich mit der Thematik der Problemstellung vertraut sind. Die Interpretation soll natürlich in Hinblick auf die ursprüngliche Fragestellung gemacht werden. Statistik gerät heute oft in Verruf, weil Ergebnisse gezielt in eine Richtung gelenkt werden, die der gewünschten Interpretation am besten entsprechen. Diese Vorgehensweise ist unwissenschaftlich und frei von jeder Objektivität. Eine (selbst-)kritische Haltung kann bei der Interpretation nicht schaden.



Kapitel 2

Darstellung eindimensionaler Daten

Wir gehen hier davon aus, dass n Datenwerte $x_1, x_2, \ldots, x_n$ gegeben sind.

2.1 Eindimensionale Streudiagramme (Scatterplots)

Bei eindimensionalen Streudiagrammen werden die Daten mit einem gewählten Symbol einfach ihrem Wert entsprechend auf einer Geraden aufgetragen.

Die Gerade muss natürlich nicht in horizontaler Richtung liegen, sondern kann im Prinzip in beliebiger Richtung verlaufen. Um die Größe der Datenwerte erkennen zu können, sollte auch eine Skalierung angebracht werden.

Schwierigkeiten in der Darstellung gibt es dann, wenn gleiche Beobachtungswerte (Mehrfachpunkte) auftreten oder Beobachtungen sehr nahe beieinander liegen, aber auch dann, wenn extreme Ausreißer vorhanden sind.

2.1.1 Mehrfachpunkte

Sie können durch folgende Variationen des Streuungsdiagramms angezeigt werden:

Variationen in der Symbolart, Symbolgröße, Symbolfarbe. Diese Variationen sind nicht empfehlenswert, da sie manchmal einen falschen Eindruck des Datensatzes vermitteln.

Horizontales oder vertikales Versetzen der Symbole. Eng zusammenliegende Datenwerte können Probleme bereiten, da sich Symbole dann meistens überschneiden.

Markierung durch vertikale Striche statt durch Markersymbole. Die Strichlänge wird proportional zur Anzahl gleicher Datenwerte gewählt. Falls Datenpunkte zu eng beisammen liegen, können einige Striche zu einer unregelmäßigen Fläche verschmelzen.

Zufällige Wahl der vertikalen Position (jittering): Statt auf einer Geraden die Postitionen $x_1, \ldots, x_n$ zu markieren, werden Punkte (x_i, y_i) gezeichnet, wobei $(y_1, \ldots, y_n)$ Realisierungen einer gleichverteilten Zufallsgröße sind. Der Plot sollte als schmales

Rechteck dargestellt werden (y-Richtung kurz), weil die wesentliche Information in x-Richtung liegt.

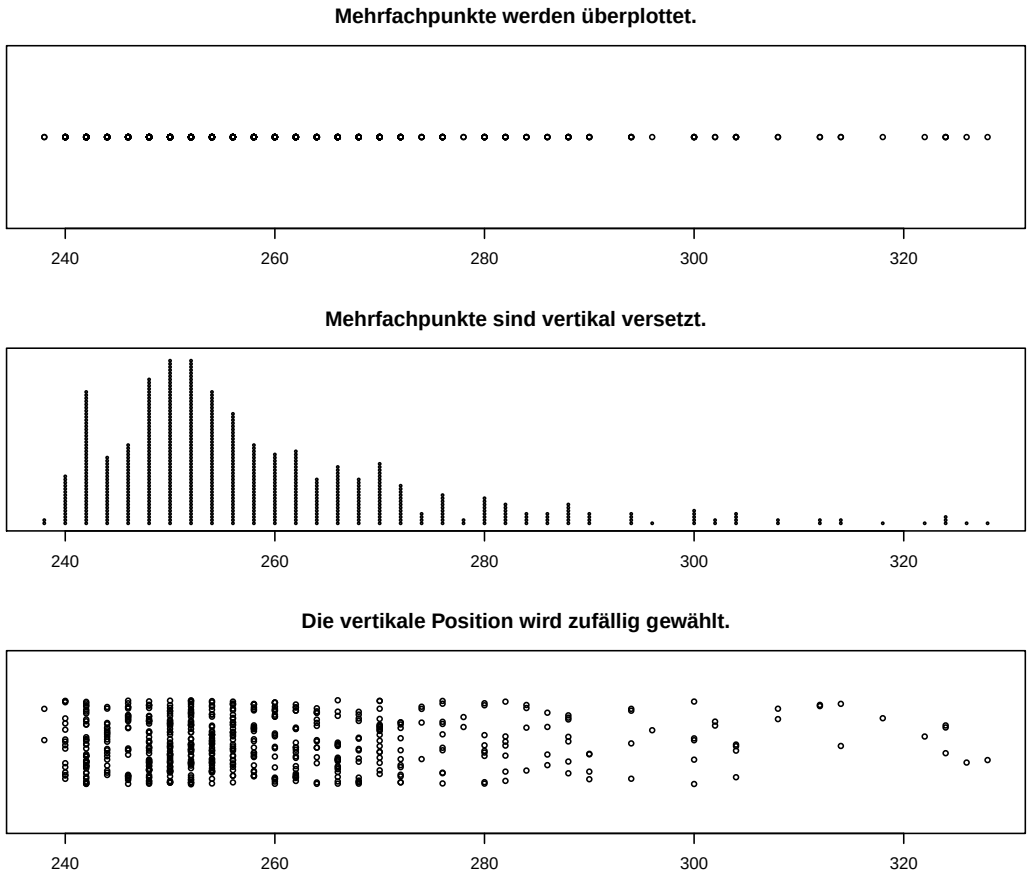


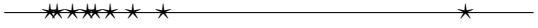
Abbildung 2.1: Durchschnittliche monatliche Ozon-Konzentration im Dezember 2000 auf einem Gitter von 24×24 Messpunkten in Zentralamerika. (`stripchart`)

Die Grafiken in Abbildung 2.1 können folgendermaßen erzeugt werden:

```
data(ozone,package="plyr")              # gesamte Daten laden
oz72 <- ozone[,,72]                     # Dezember 2000 selektieren
stripchart(oz72,method="overplot")
stripchart(oz72,method="stack")
stripchart(oz72,method="jitter")
```

2.1.2 Ausreißer

Extreme Ausreißer bewirken, dass der Rest der Beobachtungen in einem Streudiagramm sehr eng beisammen liegt.



Das Weglassen von Ausreißern lässt mehr Details vom Hauptteil der Daten erkennen. In diesem Fall sollte jedoch immer auf das Vorhandensein der weggelassenen Ausreißer hingewiesen werden.

2.2 Histogramme

$x_1, \ldots, x_n$	...	Stichprobe
n	...	Umfang der Stichprobe
k	...	Anzahl der Histogramm-Balken
$t_1, \ldots, t_k$	...	Intervallgrenzen
n_i	...	Anzahl der Daten im Intervall $[t_i, t_{i+1})$

Die Histogramm-Funktion ist definiert als

$$H(x) := \sum_{i=1}^{k-1} \frac{1}{t_{i+1} - t_i} \frac{n_i}{n} I_{[t_i, t_{i+1})}(x)$$

mit der Indikatorfunktion

$$I_{[t_i, t_{i+1})}(x) := \begin{cases} 1 & \text{für } x \in [t_i, t_{i+1}) \\ 0 & \text{sonst.} \end{cases}$$

Die Histogramm-Funktion ist hier mit relativen Häufigkeiten definiert, kann aber auch mit absoluten Häufigkeiten dargestellt werden. Die Division durch die Intervallbreite macht vor allem Sinn bei nicht äquidistanten Intervallgrenzen, weil dann die einzelnen Balken des Histogramms flächenmäßig richtig dargestellt werden (siehe Abbildung 2.2).

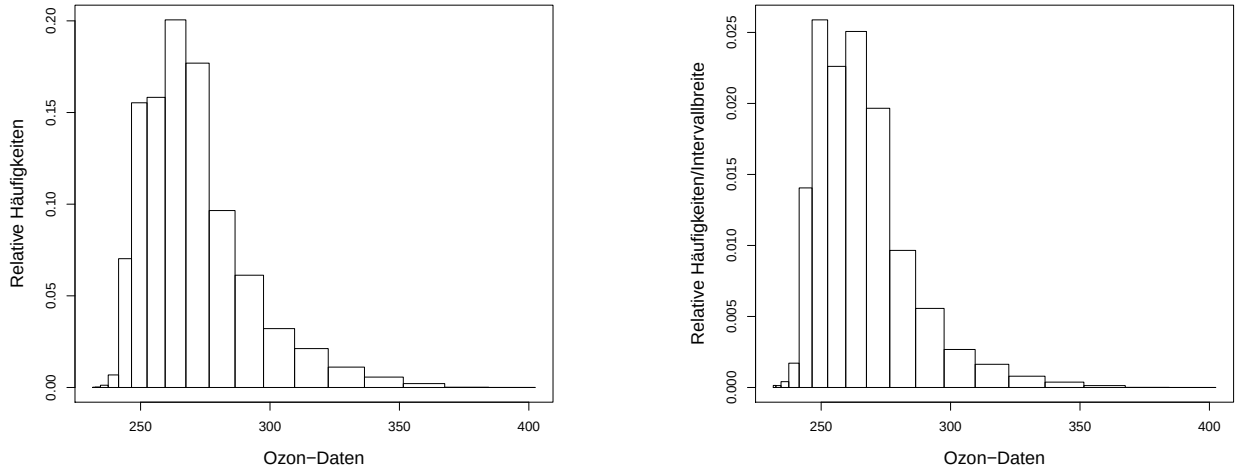


Abbildung 2.2: Darstellung der Histogramm-Funktion anhand der Ozon-Daten. LINKS: Nur die relativen Häufigkeiten werden aufgetragen. RECHTS: Die relativen Häufigkeiten werden dividiert durch die Intervallbreite. (`hist`)

2.2.1 Wahl der Intervalllänge

Die Erscheinungsform eines Histogramms hängt von der Intervalllänge ab, aber auch vom Start-(End-)Punkt, bei dem die Intervalleinteilung beginnt (aufhört). Wir gehen nachfolgend davon aus, dass die Intervallgrenzen t_i äquidistant sind. Somit ergibt sich einer Intervallänge von $h_n = t_{i+1} - t_i$, für alle $i = 1, \ldots, k-1$.

Intervalllänge nach Sturges: Die optimale Anzahl k von Histogramm-Balken wird gewählt als

$$k = \lceil \log_2(n) + 1 \rceil$$

wobei $\lceil \cdot \rceil$ Aufrunden zur nächsten ganzen Zahl bedeutet. Somit wäre die optimale Intervallänge $h_n = (t_k - t_1)/k$. Diese Wahl ist gedacht für Daten, die aus einer normalverteilten Grundgesamtheit kommen. Für genauere Diskussion, siehe `https://robjhyndman.com/papers/sturges.pdf`.

Intervalllänge nach Scott: Unter den Voraussetzungen:

f Dichtefunktion der Daten
f stetig
f', f'' stetig und beschränkt

ist die optimale Intervalllänge

$$h_n = t_{i+1} - t_i = \left(\frac{6}{\int_{-\infty}^{\infty} f'(x)^2 dx}\right)^{\frac{1}{3}} n^{-\frac{1}{3}}$$

Sie ist optimal in Bezug auf den MSE (Mean Squared Error)

$$\mathrm{MSE}(x) = \mathrm{E}(H(x) - f(x))^2 \qquad \text{für festes } x$$

Ein Maß für die Abweichungen über den gesamten Bereich ist der IMSE (Integrated Mean Squared Error):

$$\mathrm{IMSE} = \int \mathrm{MSE}(x) dx$$

Da f unbekannt ist, kann obige Formel nicht ausgewertet werden. Unter Normalverteilung gilt

$$h_n = \frac{3.5s}{\sqrt[3]{n}} \qquad s \text{ empirische Standardabweichung}$$

Intervalllänge nach Freedman und Diaconis: Unter ähnlichen Voraussetzungen wie bei Scott ist die optimale Intervalllänge

$$h_n = \frac{2 \cdot \mathrm{IQR}}{\sqrt[3]{n}}$$

wobei $\mathrm{IQR} = q_{0.75} - q_{0.25}$ ist, also die Differenz der Quantile 0.75 und 0.25. IQR ist eine robuste Variante zur Schätzung der Standardabweichung. Diese Regel sollte daher von Ausreißern weniger beeinflusst sein.

Abbildung 2.3 zeigt einen Vergleich von Histogrammen berechnet nach Scott bzw. Freedman-Diaconis. Die Regel von Scott ist empfindlich gegenüber Ausreißern.

Abbildung 2.4 zeigt einen numerischen und grafischen Vergleich der Intervalllängen zwischen Scott und Freedman-Diaconis unter der Annahme, dass die Daten normalverteilt sind. Ein Vergleich der Anzahl der Klassen (unter Normalverteilung) ist in Tabelle 2.1 zu finden. $\mathrm{E}(R_n)$ ist dabei der Erwartungswert für die Spannweite R_n, definiert als Differenz von Maximum und Minimum.

2.3 Dichteschätzung

Wir versuchen nun direkt die Dichtefunktion der zugrunde liegenden Zufallsgröße zu schätzen. Eine Dichteschätzung bewirkt durch eine lokale Vorgangsweise ein glatteres Aussehen. Die Dichte der Daten im Punkt x wird aus den Daten berechnet, die in einem Fenster

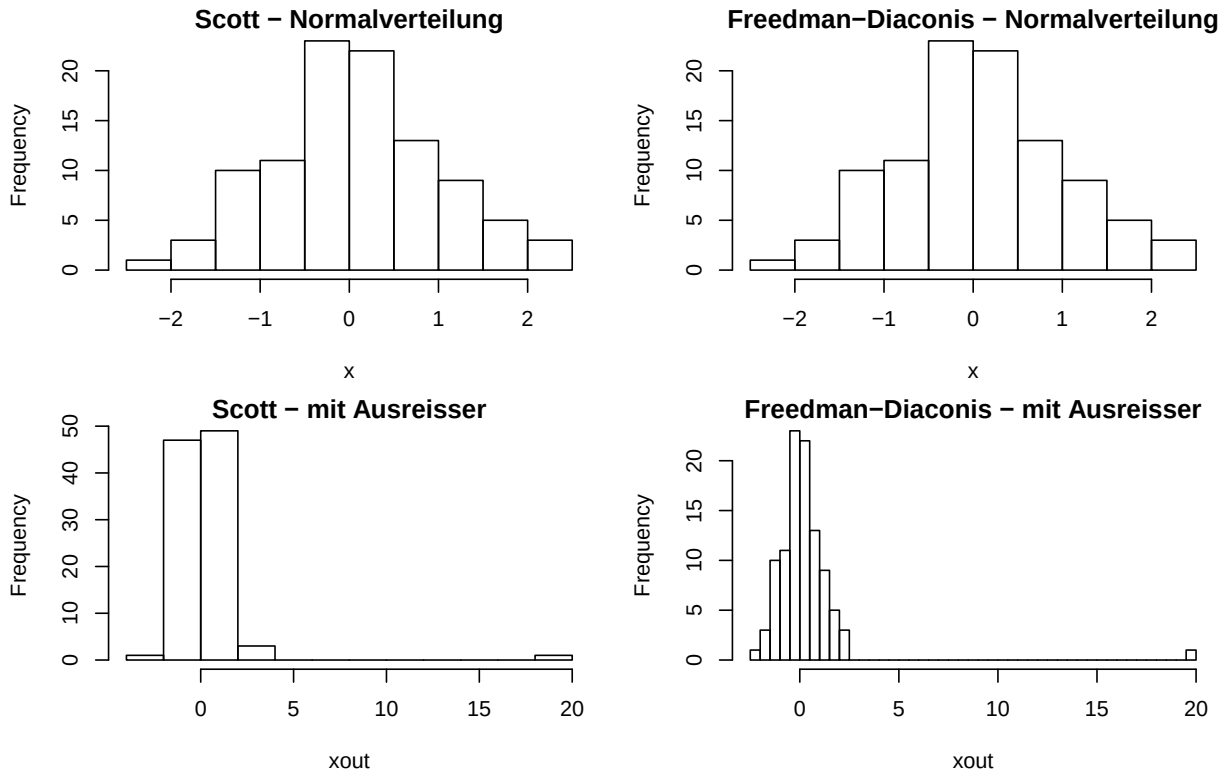


Abbildung 2.3: Vergleich von Histogrammen nach Scott bzw. Freedman-Diaconis bei normalverteilten Daten ohne und mit Ausreißern

n	Scott	Freedman-Diaconis
10	1.620	1.252
20	1.286	0.994
30	1.123	0.868
40	1.020	0.789
50	0.947	0.743
75	0.828	0.640
100	0.752	0.582
150	0.657	0.508
200	0.597	0.461
300	0.521	0.403

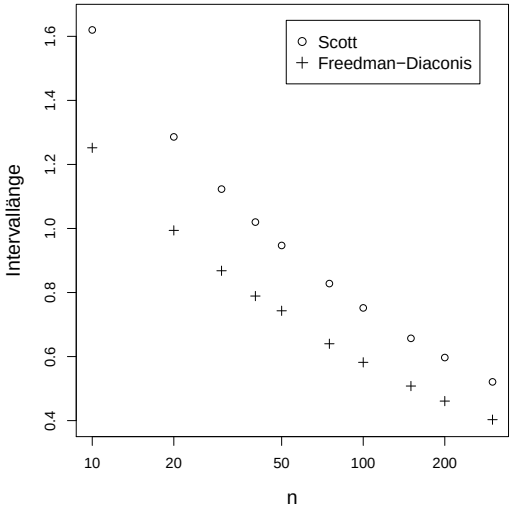


Abbildung 2.4: Vergleich von Intervalllängen (in Einheiten der geschätzten Streuung s) optimaler Histogramme unter der Annahme, dass die Daten normalverteilt sind

n	$\mathrm{E}(R_n)$	$\frac{\mathrm{E}(R_n)}{h_n}$ Scott	$\frac{\mathrm{E}(R_n)}{h_n}$ Freedman-Diaconis
10	3.078	1.90	2.46
20	3.735	2.91	3.76
30	4.086	3.64	4.71
40	4.322	4.23	5.48
50	4.498	4.95	6.14
75	4.806	5.81	7.51
100	5.015	6.67	8.63
150	5.298	8.87	10.43
200	5.492	9.20	11.90
300	5.756	11.04	14.28

Tabelle 2.1: Vergleich der Klassenanzahl optimaler Histogramme unter der Annahme, dass die Daten normalverteilt sind (EW_n gemessen in Einheiten der geschätzten Streuung s)

$[x - \frac{h}{2}, x + \frac{h}{2}]$ liegen (lokale Dichte!).

h ... Intervalllänge (= Fensterbreite)

$W(t)$ ... Gewichtsfunktion $\int_{-\infty}^{\infty} W(t) = 1$

$$\hat{f}(x) := \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} W(\frac{x - x_i}{h})$$

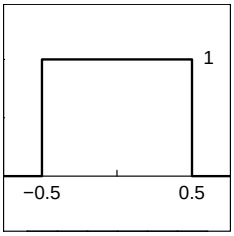
Daraus folgt mit der Substitution $t = \frac{x-x_i}{h}$ $(dt = \frac{1}{h}dx)$:

$$\int_{-\infty}^{\infty} \hat{f}(x)dx := \frac{1}{n} \sum_{i=1}^{n} \int_{-\infty}^{\infty} \frac{1}{h} W(\frac{x - x_i}{h})dx = \frac{1}{n} \sum_{i=1}^{n} \int_{-\infty}^{\infty} \frac{1}{h} hW(t)dt = 1$$

a) Rechteck Gewichtsfunktion (boxcar function):

$$W(t) = \begin{cases} 1 & \mid t \mid \leq \frac{1}{2} \\ 0 & \text{sonst} \end{cases}$$

d.h. $W(\frac{x-x_i}{h}) = 1$ für $|x - x_i| \leq \frac{h}{2}$



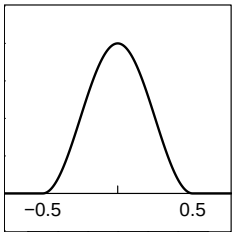
Als Ergebnis erhält man eine Treppenfunktion. Meistens wird jedoch $\hat{f}(x)$ an äquidistanten x-Werten ermittelt und diese Werte linear verbunden. Dann gilt jedoch nicht mehr $\int \hat{f}(x)dx = 1$.

b) Cosinus Gewichtsfunktion (cosine function):

$$W(t) = \begin{cases} 1 + \cos 2\pi t & \mid t \mid < \frac{1}{2} \\ 0 & \text{sonst} \end{cases}$$

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} (1 + \cos 2\pi t)dt = 1 + \frac{1}{2\pi} \sin 2\pi t \mid_{-\frac{1}{2}}^{\frac{1}{2}} =$$

$$1 + 0 - 0 = 1$$



Mit dieser Gewichtsfunktion ergibt sich $\hat{f}(x)$ als stetige Funktion.

Abbildung 2.5 zeigt die Auswirkung der Wahl der Gewichtsfunktion auf die Gestalt der Dichteschätzung.

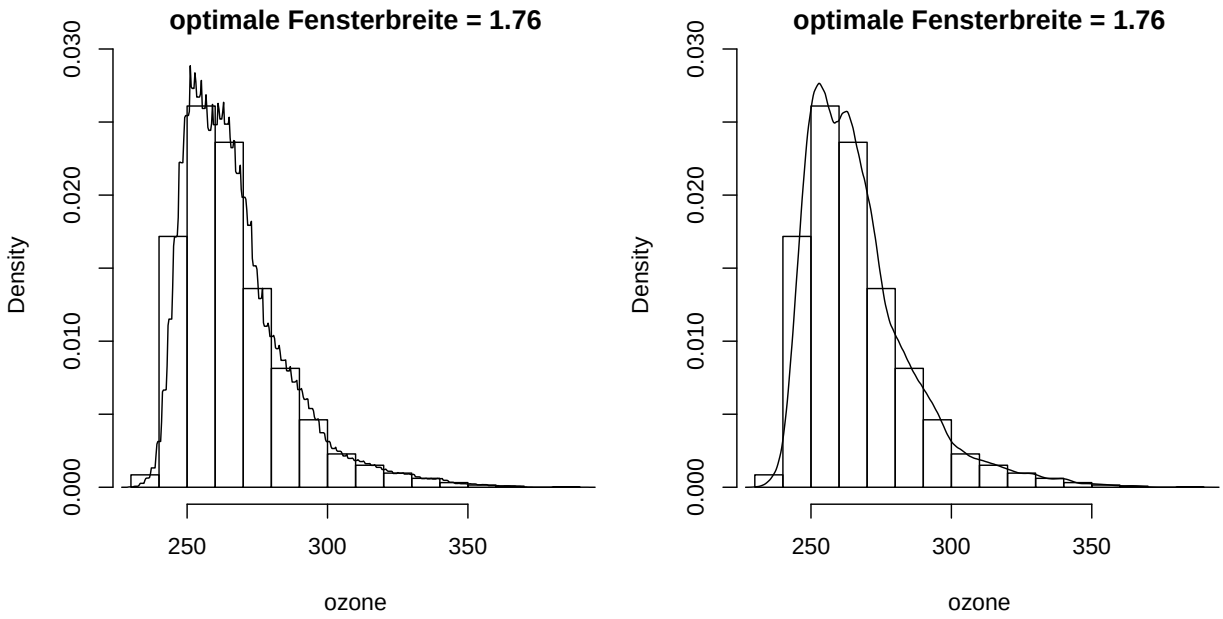


Abbildung 2.5: Dichteschätzung für die Ozondaten aus Abb. 2.1 mit der Rechteck-Gewichtsfunktion (links) und der Cosinus-Gewichtsfunktion (rechts). (`density`)

2.4 Auswahl eines diskreten Wahrscheinlichkeitsmodells

Aus folgenden Gründen ist es sinnvoll, Verteilungsannahmen über die Daten zu treffen:

- Kompakte Beschreibung der Daten als Stichprobe einer theoretischen Verteilung.
- Annahmen über die Verteilung führen zu "besseren" statistischen Verfahren.

Diskrete Verteilungen, die grafisch leicht identifiziert werden können:

- Binomialverteilung:
  $p_x = P(X = x) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \qquad x = 0, 1, \ldots, n \qquad 0 < \theta < 1$
- Negative Binomialverteilung:
  $p_x = \binom{x+m-1}{m-1}\theta^m(1-\theta)^x \qquad x = 0, 1, \ldots \qquad 0 < \theta < 1, \quad m > 1$
- Poissonverteilung:
  $p_x = \frac{\lambda^x e^{-\lambda}}{x!} \qquad x = 0, 1, \ldots \qquad \lambda > 0$
- Logarithmische Verteilung
  $p_x = -\frac{\theta^x}{x \ln(1-\theta)} \qquad x = 1, 2, \ldots \qquad 0 < \theta < 1$

Für die beiden nächsten Verfahren gelten folgende Voraussetzungen:

$x_1, \ldots, x_n$	Stichprobe
n_x	Anzahl der x_j, für die $x_j = x$
$\hat{p}_x$	$\frac{n_x}{n}$

2.4.1 Verfahren von Ord

Für eine große Anzahl von Stichproben eignet sich für eine grafische Darstellung folgendes:

$$\hat{U}_x := \frac{x\hat{p}_x}{\hat{p}_{x-1}}$$

Zeichnen aller Punkte $(x, \hat{U}_x)$, für die $\hat{p}_{x-1} > 5\%$. Falls $\hat{U}_x \approx a + bx$ (d.h. linear), dann ist je nach Verlauf der Geraden eine Entscheidung gemäß Abbildung 2.6 zu treffen.

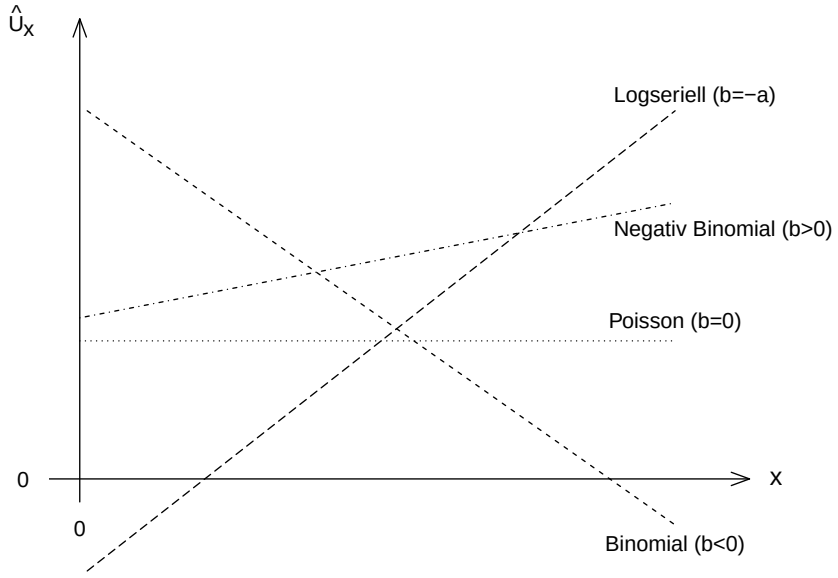


Abbildung 2.6: Auswahl eines diskreten Wahrscheinlichkeitsmodells nach dem Verfahren von Ord

2.4.2 Verfahren von Hoaglin 1960

Für kleine Stichproben. Entscheidung zwischen Poissonverteilung und Logarithmischer Verteilung. Zeichnen aller Punkte $(x, \hat{Y}_x)$ und $(x, \hat{V}_x)$ mit

$$\hat{Y}_x := \ln(x!\hat{p}_x)$$

und

$$\hat{V}_x := \ln(x\hat{p}_x).$$

$(x, \hat{Y}_x)$ linear $\rightarrow$ Poissonverteilung
$(x, \hat{V}_x)$ linear $\rightarrow$ Logarithmische Verteilung

2.5 Empirische Verteilungsfunktion und Wahrscheinlichkeitsnetz

2.5.1 Empirische Verteilungsfunktion

Es seien Stichproben $x_1, \ldots, x_n$ gegeben. Wir wollen nun nicht mehr wie früher die *Dichtefunktion* f characterisieren, sondern die *Verteilungsfunktion* F. Für kontinuierliche Größen ist die (theoretische) Verteilungsfunktion gegeben durch

$$F(x) = \int_{-\infty}^{x} f(t)\mathrm{d}t$$

für alle x aus dem Definitionsbereich. Die empirische Verteilungsfunktion F_n stellt eine Schätzung der theoretischen Verteilungsfunktion F dar, und sie ist definiert durch:

$$F_n(x) := \frac{1}{n}\sum_{i=1}^{n} I_{[x_i,\infty)}(x)$$

Beim Vergleich zwischen empirischer und theoretischer Verteilungsfunktion kann es zu Schwierigkeiten kommen, falls die theoretische Verteilungsfunktion einen Definitionsbereich besitzt, der $-\infty$ oder ∞ enthält (z.B. Normalverteilung). Besser ist daher folgende Defninition:

$$F_n(x) := \begin{cases} \frac{i-0.5}{n} & x = x_i \\ \text{An den Rändern und zwischen den } x_i \text{ konsistent mit der Definition} \\ \text{einer Verteilungsfunktion festgelegt, z.B. linear interpoliert, } \ldots \end{cases}$$

Anmerkung: Es werden meistens nur die Werte $F_n(x_i)$ gezeichnet.

In Abbildung 2.7 werden die empirischen Verteilungsfunktionen von zufällig erzeugten standardnormalverteilten Werten gezeigt. Links wurden 30 Werte generiert, rechts 1000. Die jeweils punktierte Linie ist die theoretische Verteilungsfunktion der Standard-Normalverteilung, und man sieht dass bei wenig Werten die Abstände zur theoretischen Verteilungsfunktion doch groß sein können, während bei höherer Stichprobenanzahl die empirische kaum mehr von der theoretischen Verteilungsfunktion unterscheidbar ist. Man kann auch zeigen, dass für $n \longrightarrow \infty$ **die empirische gegen die theoretischen Verteilungsfunktion konvergiert**.

Abbildung 2.8 zeigt Daten aus der Geochemie. Auf der Halbinsel Kola im Grenzgebiet Norwegen–Finnland–Russland wurden jeweils etwa 600 Stichproben in verschiedenen Tiefenlagen des Bodens genommen. Das Erdmaterial wurde anschließend im Labor nach der Konzentration verschiedenster chemischer Elemente untersucht. In der linken Grafik werden die Werte von Scandium (Sc) im C-Horizont mittels empirischer Verteilungsfunktion dargestellt. Offenbar wurden bei höheren Konzentrationen die Werte im Labor gerundet, was durch höhere Sprünge ersichtlich ist. Die rechte Grafik zeigt die Werte von Nickel (Ni) im O-Horizont in eine log-Skala. Hohe Werte von Ni im O-Horizont weisen auf starke Bodenverschmutzung. Die Verteilung scheint nicht symmetrisch zu sein.

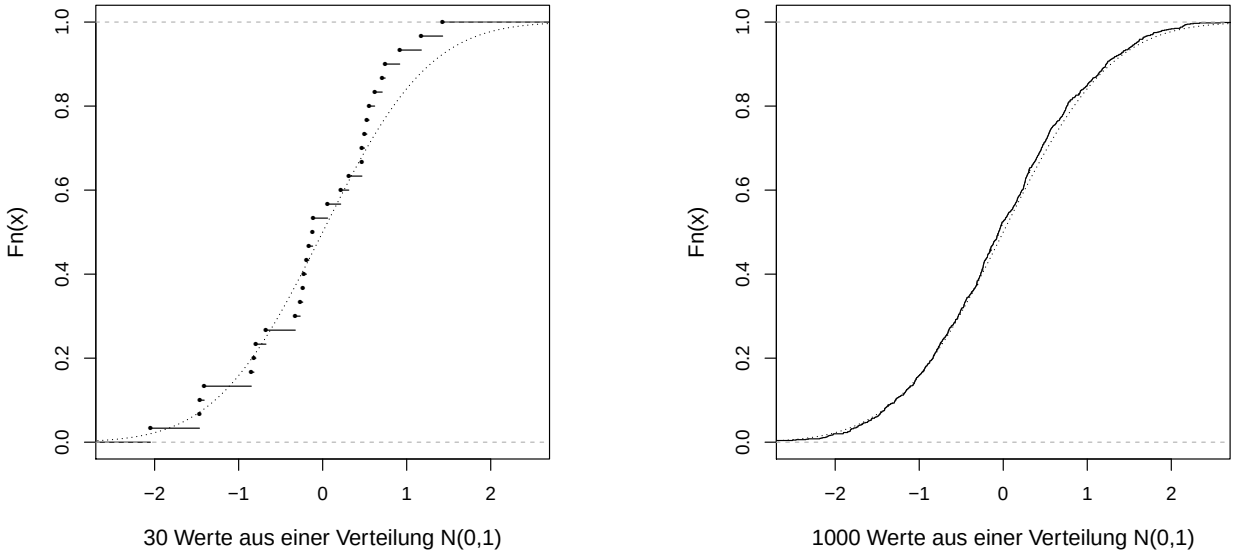


Abbildung 2.7: Vergleich der empirischen Verteilungsfunktion mit der theoretischen Verteilungsfunktion (punktiert). (`ecdf`)

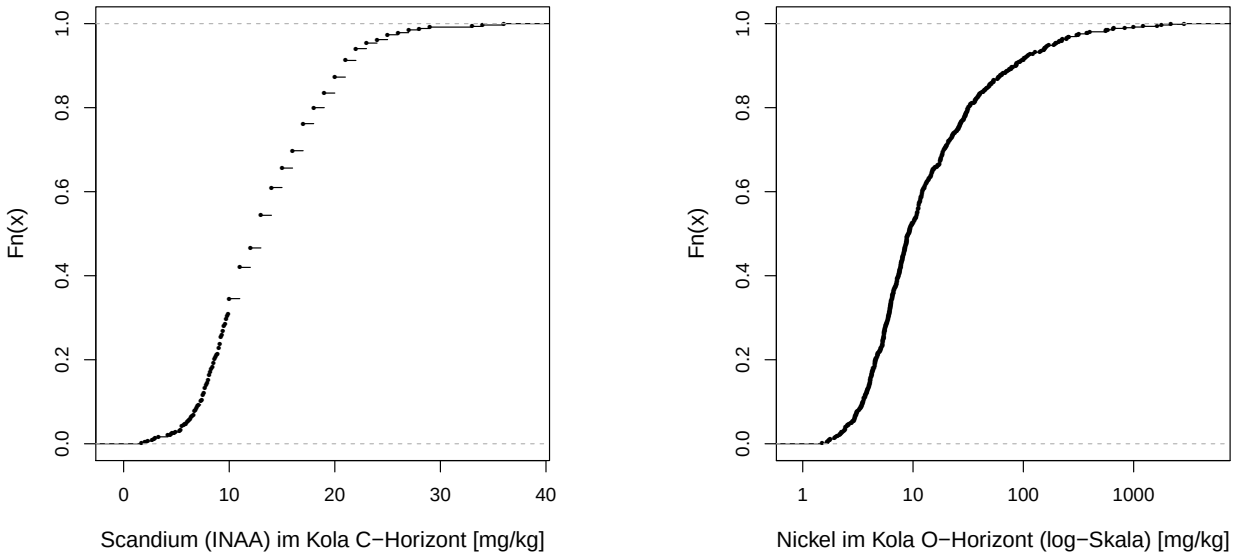


Abbildung 2.8: Empirische Verteilungsfunktionen. LINKS: Sc im C-Horizont, RECHTS: Ni (log) im O-Horizont

2.5.2 Wahrscheinlichkeitsnetz

Beim Plot der empirischen Verteilungsfunktion ist oft schwer nachvollziehbar, ob die Daten aus einer speziellen Verteilung (z.B. Normalverteilung) kommen. Man könnte aber zu diesem Zweck die vertikale Achse verzerren, indem man die Skalierung ändert. Im Wahrscheinlichkeitsnetz wird die vertikale Achse zwischen 0 und 1 nicht in gleich große Intervalle geteilt, sondern es werden die Abstände proportional zu Φ^{-1} aufgetragen, wobei Φ^{-1} die Inverse der Verteilungsfunktion der Standard-Normalverteilung ist. Bei Verwendung dieser Skalierung wird dann nicht $F_n(x)$ über x grafisch dargestellt, sondern $\Phi^{-1}(F_n(x))$ über x. Wenn nun die Daten ungefähr normalverteilt sind, so wird $F_n(x) \sim F(x)$ (also $N(\mu, \sigma^2)$) sein und

$$\Phi^{-1}(F_n(x)) \sim \Phi^{-1}(F(x)) = \Phi^{-1}\left(\Phi\left(\frac{x-\mu}{\sigma}\right)\right) = \frac{x-\mu}{\sigma},$$

sodass die Punkte ungefähr auf einer Geraden zu liegen kommen. Man liest dann bei der Wahrscheinlichkeit 50% eine Schätzung für μ ab und bei 84% eine Schätzung für $\mu + \sigma$. Im Falle von Normalverteilung sollten die Abweichungen von der Geraden "rein zufällig" sein und nicht systematisch, aber diese hängen stark von der Anzahl der Beobachtungen ab. Je mehr Beobachtungen vorhanden sind, desto weniger Abweichungen sollten auftreten.

Für die Kola Daten aus Abbildung 2.8 sind die Wahrscheinlichkeitsnetze in Abbildung 2.9 dargestellt. Links sieht man Sc im C-Horizont. Die Abweichungen von der Geraden scheinen doch beträchtlich zu sein. Bemerkenswert sind wieder die gerundeten Werte, die hier besser ersichtlich sind als in Abbildung 2.8, weil jeder einzelne Wert eingetragen ist. Aufgrund der Skalierung der horizontalen Achse kann man auch die Rückschlüsse auf die originalen Datenwerte machen.

In der rechten Grafik von Abbildung 2.9 ist Ni im O-Horizont dargestellt (log-Skala). Man überprüft somit nicht auf Normalverteilung sondern auf logarithmische Normalverteilung. Auch hier sind starke Abweichungen von der Geraden ersichtlich. Man erkennt außerdem einen "Knick" in der Funktion (etwa beim Wert 10) und kann daher darauf schließen, dass es sich hier um die Zusammensetzung zweier Verteilungen handelt. Nachdem Nickel im O-Horizont typisch für Verunreinigungen ist, könnte ein Anteil die nicht verunreinigten Gebiete beschreiben, und der zweite Anteil die verschmutzten Gebiete.

Bemerkung: Nachdem die vertikale Achse eigentlich den Quantilen der Standard-Normalverteilung entspricht, könnte man anstelle der Skalierung mit den Wahrscheinlichkeiten auch eine Skalierung mit den Quantilen vornehmen. Die Struktur des Plots würde sich nicht verändern, bloß die Skalierung der vertikalen Achse. Dies wird im folgenden Abschnitt als Q-Q Plot eingeführt.

2.6 Quantile-Quantile Plots

Mit Quantile-Quantile Plots kann man zwei Verteilungen unmittelbar miteinander vergleichen. Meist ist eine dieser Verteilungen eine hypothetische Verteilung und die andere die Verteilung vorhandener Daten. Als hypothetische Verteilung kann z.B. die Normalverteilung angenommen werden, und man vergleicht somit die Datenverteilung mit der Normalverteilung. Die Vergleichsbasis sind dabei die Quantile der Verteilungen.

In Abbildung 2.10 werden zwei Verteilungsfunktionen F_x und F_y dargestellt. Für eine konkrete Wahrscheinlichkeit p können nun die Quantile $q_x(p)$ und $q_y(p)$ ermittelt werden.

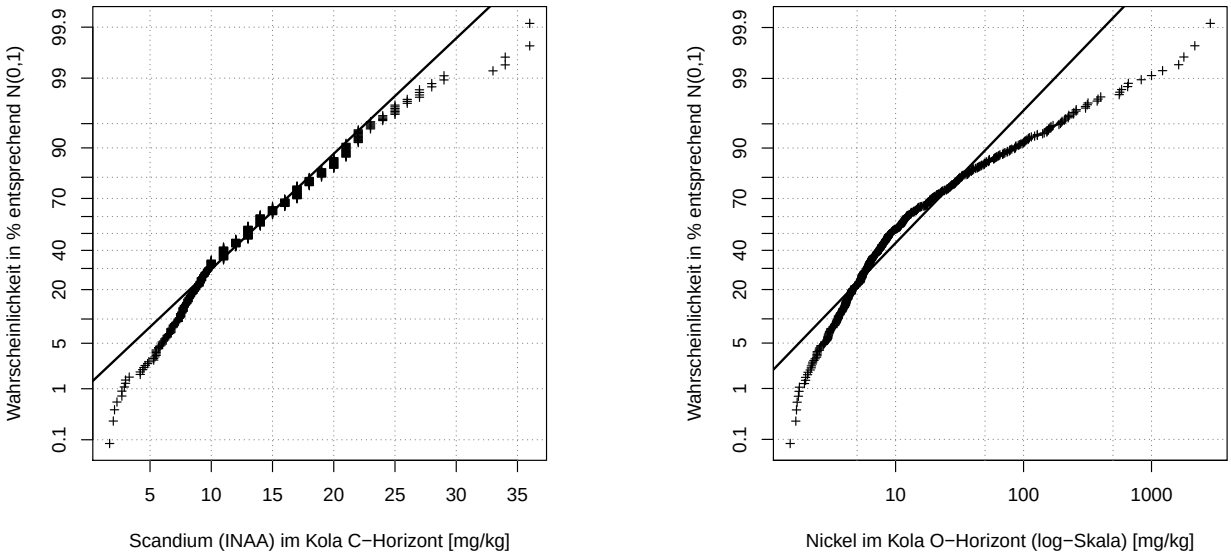


Abbildung 2.9: Wahrscheinlichkeitsnetze für die Kola Daten. LINKS: Sc im C-Horizont, RECHTS: Ni (log) im O-Horizont

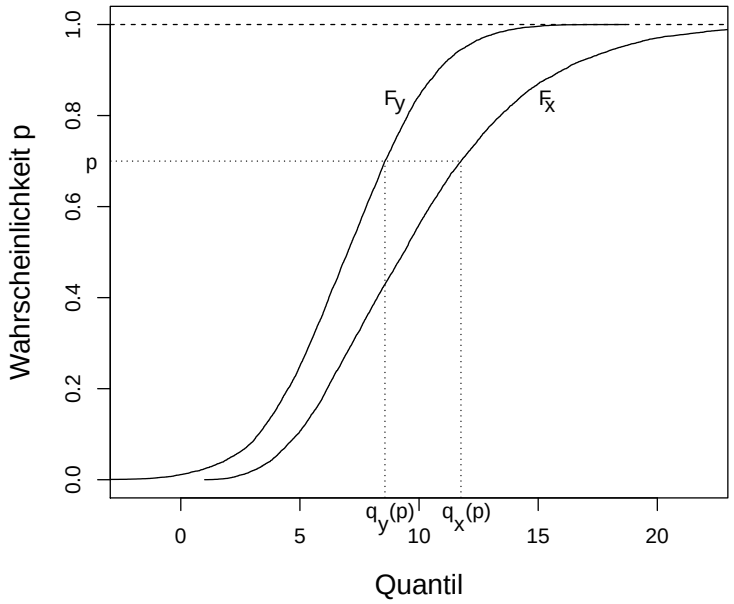


Abbildung 2.10: Darstellung von zwei Verteilungsfunktionen.

Seien also F_x bzw. F_y empirische oder theoretische Verteilungsfunktionen. Dann sind die Quantile bestimmt durch die Umkehrfunktionen:

$$F_x(t) = P(X \leq t), \quad q_x(p) = F_x^{-1}(p)$$

$$F_y(t) = P(Y \leq t), \quad q_y(p) = F_y^{-1}(p)$$

Quantile-Quantile Plot (Q-Q Plot): Zeichnen der Punkte $(q_x(p_i), q_y(p_i))$ für ausgewählte p_i, z.B. $p_i := \frac{i-\alpha}{n-2\alpha+1}$ für $0 \leq \alpha < 1$.

Wenn nun die beiden Verteilungen gleich sind, müssen auch die Quantile gleich sein und die Punkte auf einer Geraden liegen, also:

$$F_x = F_y \Leftrightarrow (q_x(p_i), q_y(p_i)) \text{ liegen auf der Geraden } y = x.$$

Man wird aber i.A. nicht auf exakte Gleichheit der Verteilungen überprüfen wollen, sondern – ähnlich wie im Wahrscheinlichkeitsnetz – darauf, ob die beiden Verteilungen aus der gleichen Verteilungsfamilie stammen. Z.B. möchte man überprüfen, ob die vorliegenden Daten normalverteilt sind mit Parametern μ und σ. Da aber die Parameter unbekannt sind, kann der Vergleich mit einer standardisierten Form derselben Verteilungsfamilie durchgeführt werden, hier also mit der Standard-Normalverteilung. Die Frage ist nun, ob und wie im Q-Q Plot dieser Vergleich durchgeführt werden kann.

Nehmen wir an, die Zufallsgröße Y gehe aus einer Transformation aus der Zufallsgröße X hervor, also

$$Y = \mu + \sigma X.$$

Dann gilt:

$$F_y(t) = P(Y \leq t) = P(\mu + \sigma X \leq t) = P\left(X \leq \frac{t-\mu}{\sigma}\right) = F_x\left(\frac{t-\mu}{\sigma}\right)$$

$$q_y(p) = F_y^{-1}(p) \qquad | \quad F_y(p)$$
$$F_y(q_y(p)) = F_y(F_y^{-1}(p)) = p$$

Mit $t = q_y(p)$ gilt somit:

$$F_y(q_y(p)) = F_x\left(\frac{q_y(p)-\mu}{\sigma}\right) = p$$

$$q_x(p) = F_x^{-1}(p) = F_x^{-1}\left(F_x\left(\frac{q_y(p)-\mu}{\sigma}\right)\right) = \frac{q_y(p)-\mu}{\sigma}$$

$$\Rightarrow \qquad q_y(p) = \mu + \sigma q_x(p)$$

Das heißt also, für Verteilungsfamilien (z.B. stetige Gleichverteilung, Exponentialverteilung, Normalverteilung,...), bei denen sich die Verteilungen der Familie nur durch Lage μ und Streuung σ unterscheiden, liegen die Punkte des Q-Q Plots zwischen 2 Verteilungen der Familie auf einer Geraden. Der Intercept-Term gibt eine Schätzung für μ, die Steigung (*slope*) eine Schätzung für σ.

Diese Vorgangsweise wird in Abbildung 2.11 für jährliche Schneefall-Daten aus Buffalo von 1910-1973 illustriert. In der linken Grafik werden die Parameter μ und σ der Normalverteilung ermittelt. Die rechte Grafik zeigt das Histogramm mit der geschätzten Dichtefunktion, und darübergelegt die theoretische Normalverteilung mit den ermittelten Parametern. Offenbar hat die Schätzung der Parameter zu einem sehr guten Ergebnis geführt.

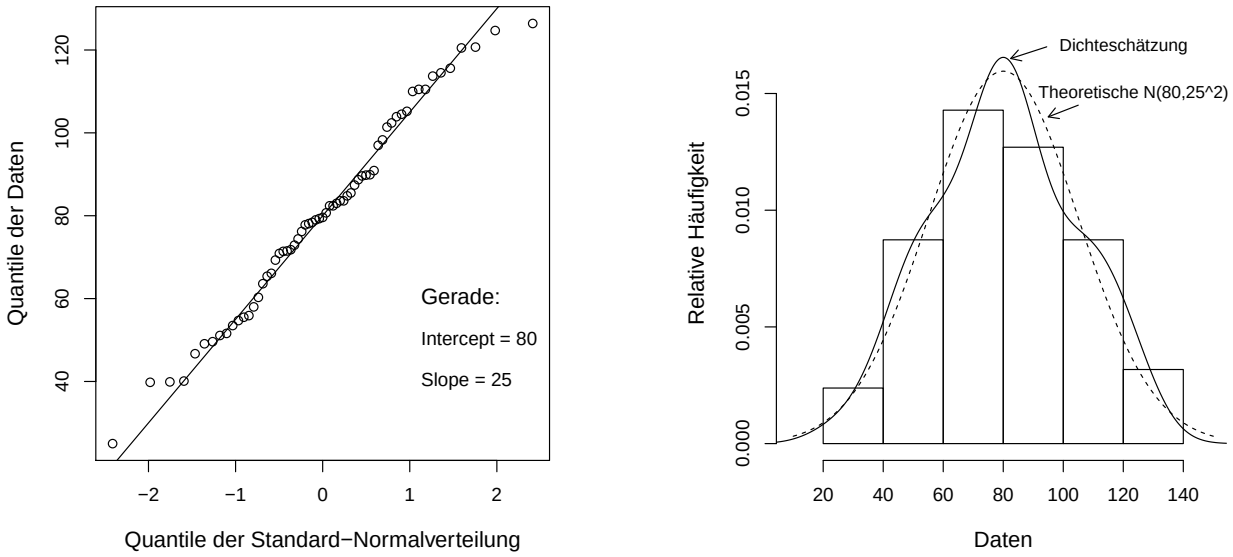
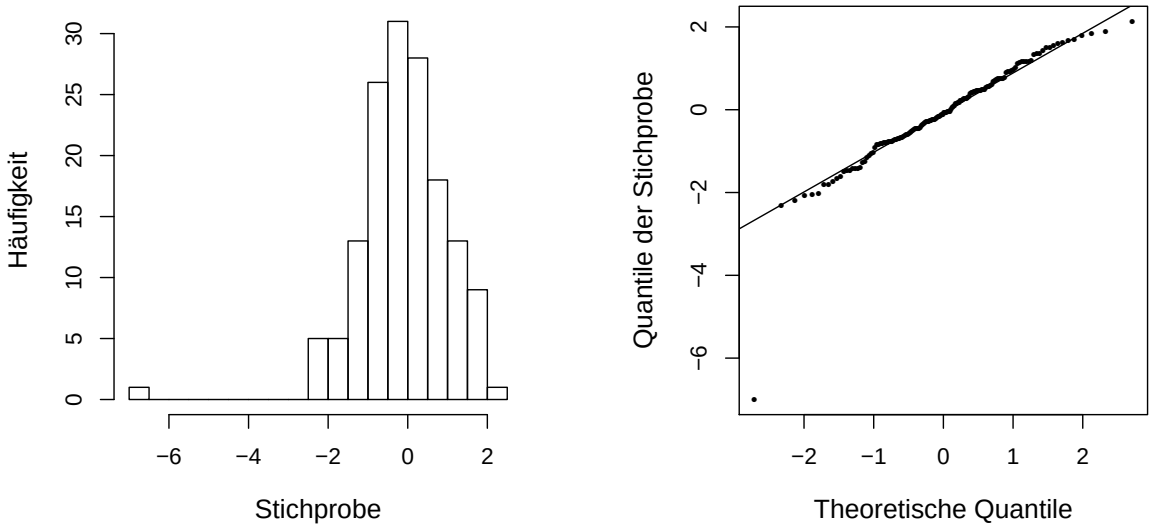


Abbildung 2.11: LINKS: Q-Q Plot der Schneefall-Daten von Buffalo (1910-1973) mit den geschätzten Parametern der Normalverteilung; RECHTS: Histogramm mit geschätzter Dichtefunktion der Originaldaten, sowie theoretische Normalverteilung mit den ermittelten Parametern.

2.6.1 Abweichungen von der Geradenform

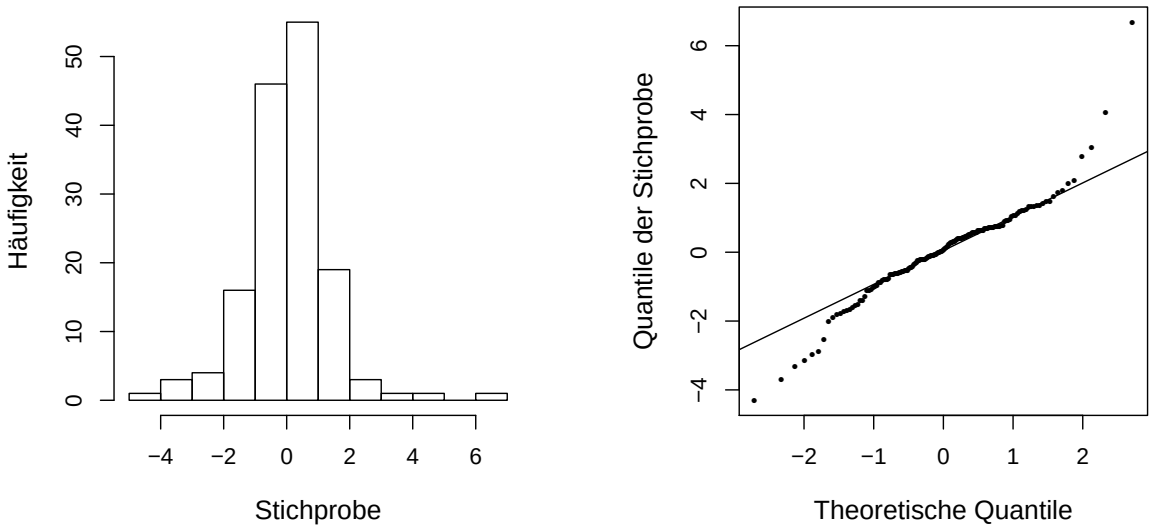
Wenn sich im Q-Q Plot Abweichungen von der Geradenform zeigen, sind mehr oder weniger große Unterschiede zwischen den beiden Verteilungsfunktionen vorhanden. Die Unterschiede können zufälliger Natur sein, z.B. bedingt durch zufällige Stichproben mit sehr geringem Stichprobenumfang. In diesem Fall kann man noch nicht darauf schließen, dass X und Y tatsächlich aus verschiedenen Verteilungen kommen. Allerdings können sich systematische Unterschiede zeigen, die auf folgende Ursachen zurückzuführen sind:

1. Ausreißer: führen zu einzelnen isolierten Punkten im unteren oder oberen Bereich

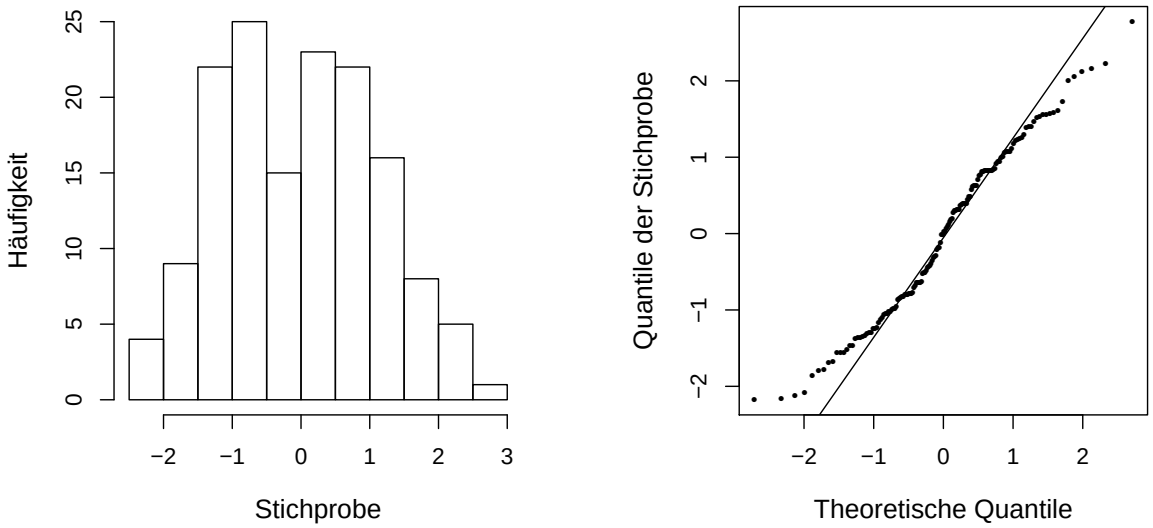


2. Krümmung an den Enden: Y (empirische VF) hat mehr Masse in den Schwänzen der

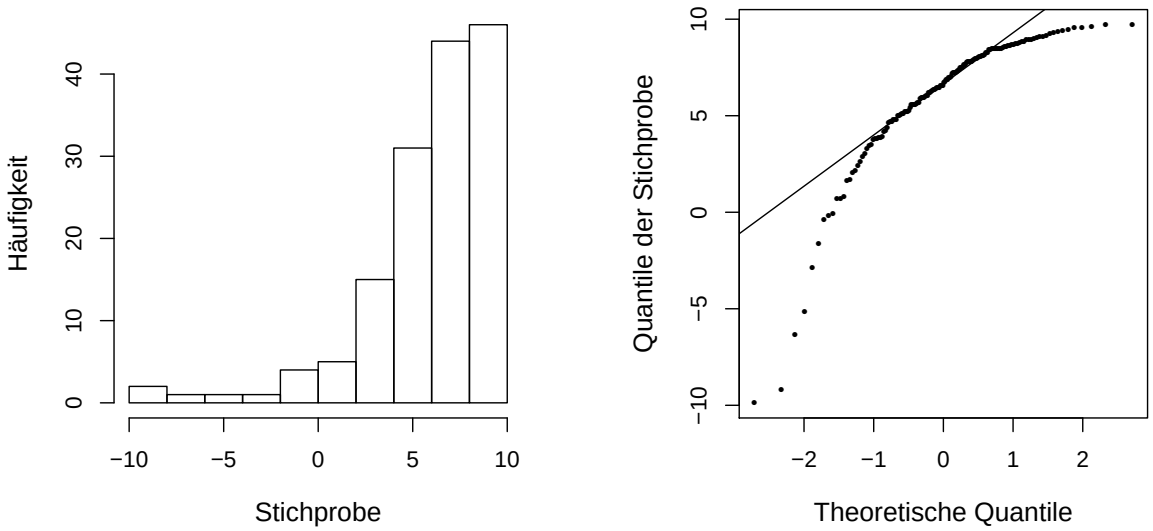
Verteilung als X (theoret. VF).



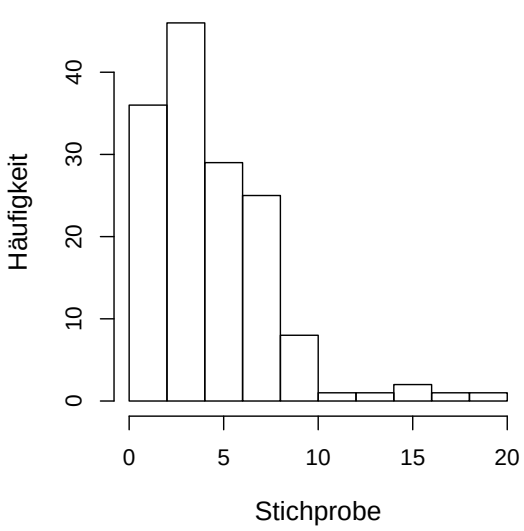
3. Krümmung an den Enden: Y (empirische VF) hat weniger Masse in den Schwänzen der Verteilung als X (theoret. VF).

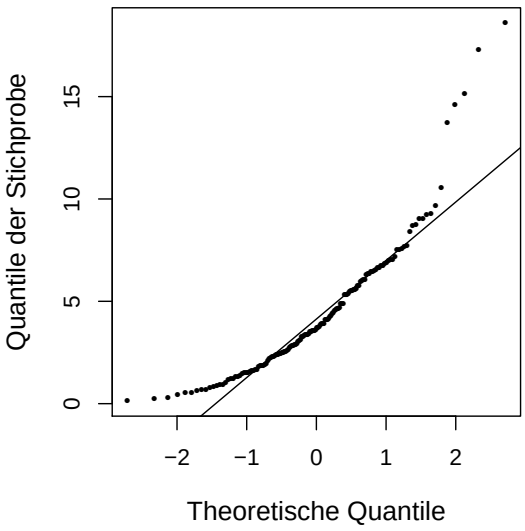


4. Konvexe oder konkave Gestalt: Falls X symmetrisch verteilt ist: Y ist linksschief, wenn die unteren Quantile weiter vom Median entfernt sind als die oberen Quantile

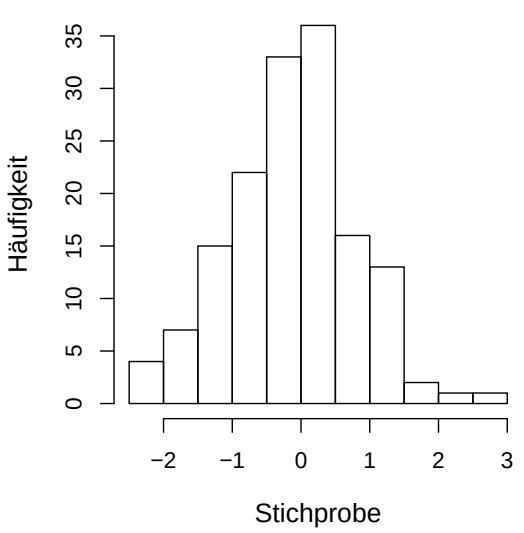


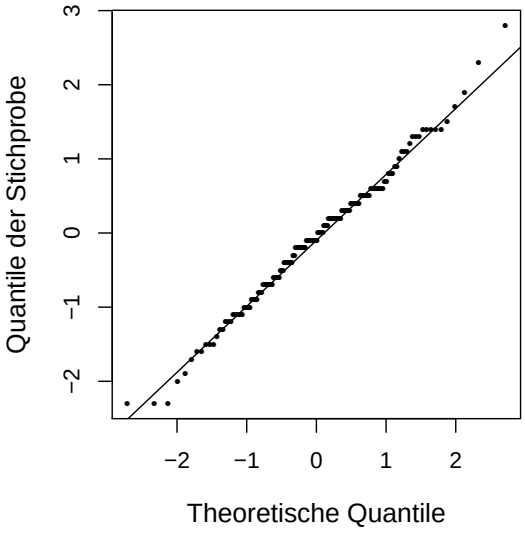
5. Konvexe oder konkave Gestalt: Falls X symmetrisch verteilt ist: Y ist rechtsschief, wenn die oberen Quantile weiter vom Median entfernt sind als die unteren Quantile



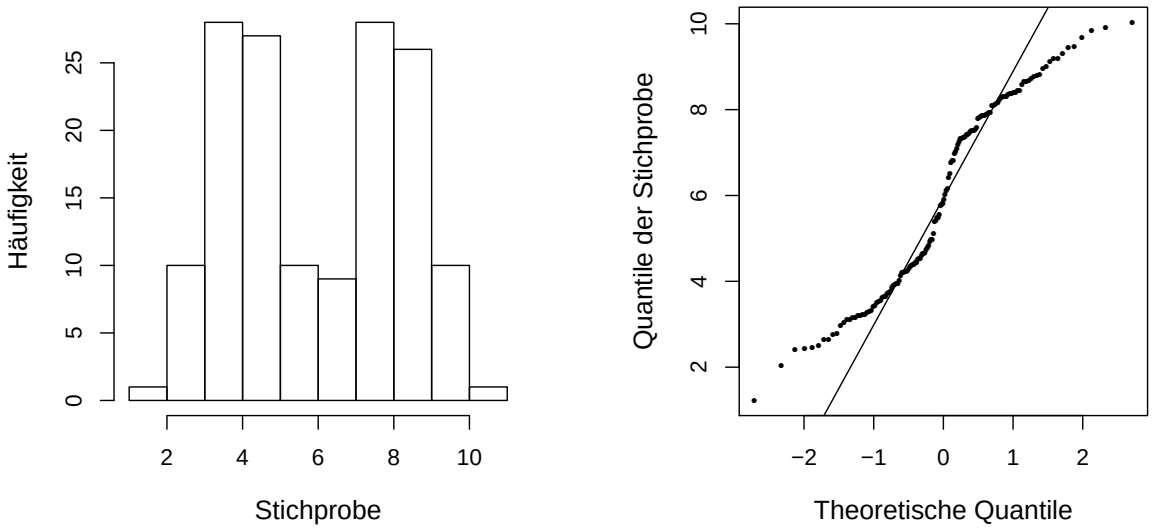


6. Horizontale Segmente: Gerundete Werte oder diskrete Verteilung





7. Plateaus: Zusammensetzung von 2 oder mehreren Verteilungen, oder Cluster in den Daten



Anmerkung:

1. Q-Q Plots reagieren empfindlich auf (auch zufällige) Abweichungen nahe den Rändern bei Verteilungen, deren Definitionsbereich bis $-\infty$ oder $+\infty$ geht (z.B. Normalverteilung).
2. Um eine schöne lineare Gestalt zu sehen, benötigt man eine größere Anzahl von Datenwerten.

2.6.2 Streuung in Q-Q Plots

Aufgrund des asymptotischen Verhaltens von Ordnungsstatistiken ist die Abweichung von Punkten in Q-Q Plots verteilt nach $N(Q(p), \frac{p(1-p)}{nf(Q(p))^2})$, wobei $Q(p)$ das Quantil zur Wahrscheinlichkeit p ist, und f die Dichte der Verteilung bezeichnen. Wir bezeichnen mit z_i die empirischen Quantile des Q-Q Plots. Ein Schätzwert s_{z_i} für den Standardfehler der empirischen Quantile eines Q-Q Plots ist

$$s_{z_i} := \frac{\hat{\delta}}{g(q_i)} \sqrt{\frac{p_i(1-p_i)}{n}}$$

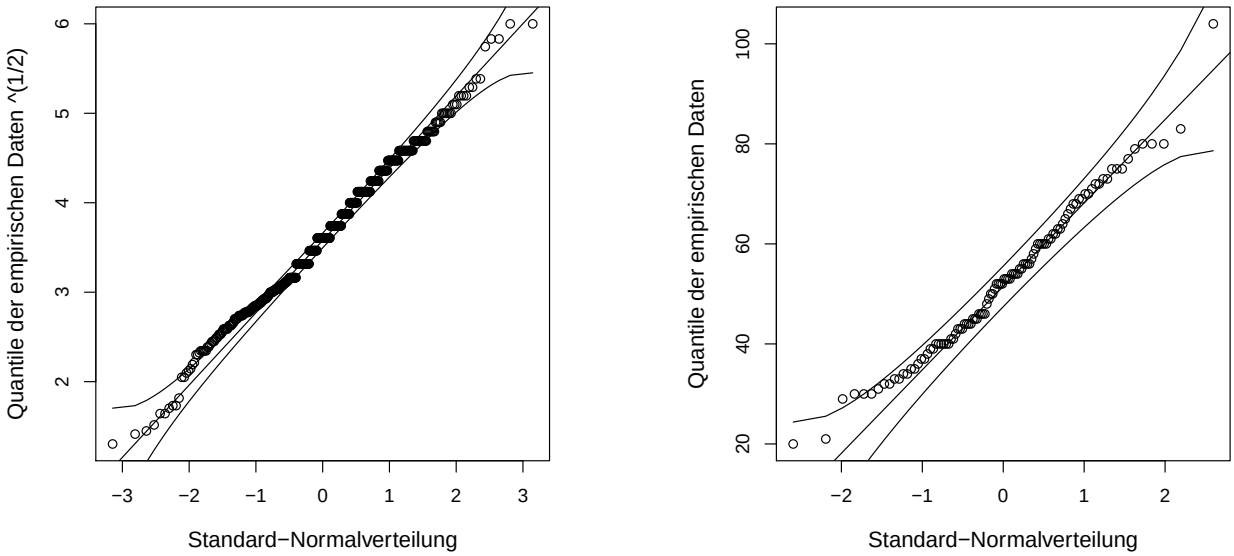
mit

$\hat{\delta}$	Schätzwert für den Anstieg des linear verlaufenden Q-Q Plots, z.B. $\frac{Q_{0.75}-Q_{0.25}}{q_{0.75}-q_{0.25}}$ mit den theoretischen Quantilen $q_{0.75}$ und $q_{0.25}$ und den empirischen Quantilen $Q_{0.75}$ und $Q_{0.25}$.
$g(x)$	Dichte der standardisierten Verteilung (z.B. Standard-Normalverteilung).
q_i	theoretische Quantile des Q-Q Plots.

Abbildung 2.12 zeigt Q-Q Plots mit zusätzlicher Streuungsinformation. Dabei wurde für jedes theoretische Quantil q_i (horizontal) der Standardfehler ermittelt und vom Schnittpunkt mit der Geraden zwei mal nach oben und unten aufgetragen (Konfidenzintervall). Die Punkte wurden dann mit Linien verbunden, um den optischen Eindruck zu erleichtern. Punkte, die somit außerhalb dieser Linien liegen, würden eine signifikante Abweichung von der theoretischen Verteilung haben.

Man bemerke, dass für steigenden Stichprobenumfang n die Standardfehler kleiner werden und somit die beiden Linien enger beisammenliegen.

Aus Abbildung 2.12(a) ist ersichtlich, dass diese (transformierten) Daten relativ gut der Normalverteilung entsprechen. Die Niederschlagsdaten in Abbildung 2.12(b) zeigen allerdings signifikante Abweichung im unteren Bereich.



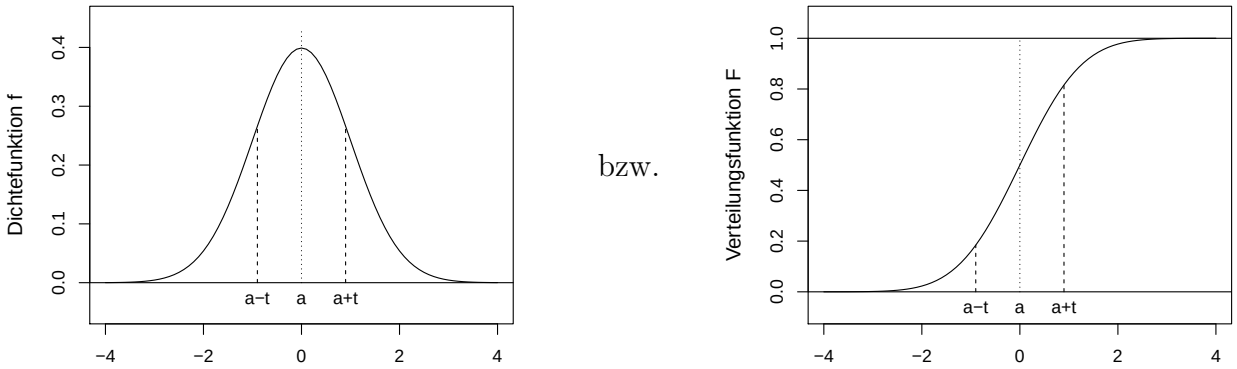
(a) Sc in C-Horizont (b) Schneefall-Daten von Buffalo

Abbildung 2.12: Q-Q Plots mit zusätzlicher Streuungsinformation, erzeugt z.B. mit der Funktion `qq.plot` von der `library(car)`.

2.6.3 Prüfung auf Symmetrie einer Verteilung durch Q-Q Plots

Für eine um $x = a$ symmetrische Dichtefunktion $f(x)$ gilt:

$$f(a-t) = f(a+t) \qquad F(a-t) = 1 - F(a+t)$$



Mit $p := F(a-t)$ ist somit $F(a+t) = 1-p$. Betrachten wir nun die Quantile (für $p \leq 0.5$):

$$F^{-1}(p) = F^{-1}(F(a-t)) = a - t \quad \text{bzw.} \quad F^{-1}(1-p) = F^{-1}(F(a+t)) = a + t$$

Somit sind auch die Quantile $F^{-1}(p)$ und $F^{-1}(1-p)$ (für $p \leq 0.5$) symmetrisch um a. Diese Idee kann nun verwendet werden, um auf Symmetrie zu prüfen.

Wir betrachten die geordneten Stichprobenwerte $x_{(1)}, \ldots, x_{(n)}$ sowie den Median x_M der Stichprobe. Diese Werte werden als Quantile der Verteilung verstanden und können im Q-Q Plot verwendet werden.

- Symmetrie um den Median:
  Wir bilden die Punktepaare

  $$(x_M - x_{(1)}, x_{(n)} - x_M), (x_M - x_{(2)}, x_{(n-1)} - x_M), \ldots,$$

  die bei Symmetrie auf einer 45-Grad Geraden liegen müssten.

- Symmetrie aller Quantilspaare:
  Wir bilden die Punktepaare

  $$(x_{(n)} - x_{(1)}, x_{(n)} + x_{(1)}), (x_{(n-1)} - x_{(2)}, x_{(n-1)} + x_{(2)}), \ldots.$$

  Bei Symmetrie müssten die Punkte auf einer horizontalen Geraden liegen (etwa 2 mal Median).

Beispiel: Abbildung 2.13 zeigt beide Varianten der Q-Q Plots für die Schneefall-Daten von Buffalo.

In der linken Grafik erkennt man, dass die Punkte im Zentrum (links unten) recht gut um die Gerade liegen, was auf Symmetrie um den Median schießen lässt. Je weiter man vom Zentrum weggeht, umso asymmetrischer wird die Verteilung. Nachdem die Werte oberhalb der Geraden liegen, haben die oberen Quantile größere Abstände zum Median als die unteren Quantile, was auf eine rechtsschiefe Verteilung schließen lässt. Das Minimum ist offenbar viel weiter weg vom Median als das Maximum.

Ähnliche Aussagen erhält man mit der rechten Grafik von Abbildung 2.13. Die eingezeichnete Gerade entspricht zwei mal dem Median der Werte. Man erkennt, dass die "inneren" Punkte noch eine ähnliche horizontale Position haben, die allerdings leicht oberhalb des Medians verlaufen. Danach ist ein Trend nach oben erkennbar, was bedeutet, dass die Differenzen größer werden und somit eine rechtsschiefe Verteilung vorliegt. Das Minimum ist wiederum als Sonderfall erkennbar.

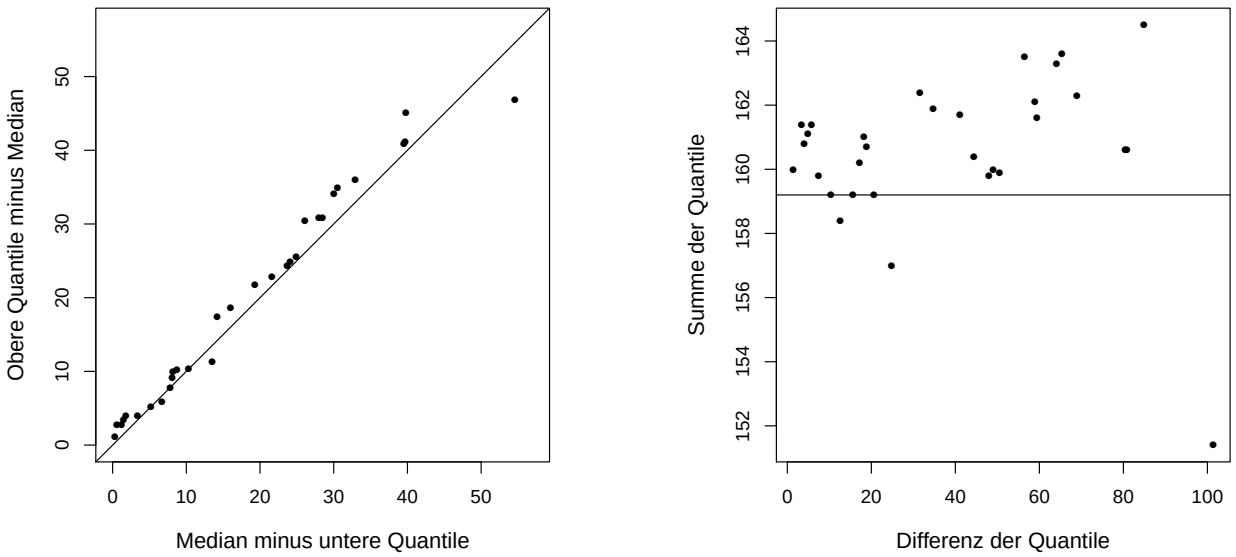


Abbildung 2.13: Q-Q Plots der Schneefall-Daten von Buffalo. LINKS: Überprüfung auf Symmetrie mit Normierung um den Median; RECHTS: Überprüfung auf Symmetrie ohne Normierung.

2.7 Boxplots

Zweck: Darstellung wichtiger Kenndaten von Verteilungen eindimensionaler Zufallsgrößen.
Vergleich mehrerer Stichproben, von Datengruppen, usw.

Stichprobe: $x_1, \ldots, x_n$
$x_M := median(x_1, \ldots, x_n)$
$Q_{0.25}$ und $Q_{0.75}$ sind die Quantile 0.25 bzw. 0.75
$\mathrm{IQR} = Q_{0.75} - Q_{0.25}$ ist der Interquartil-Abstand

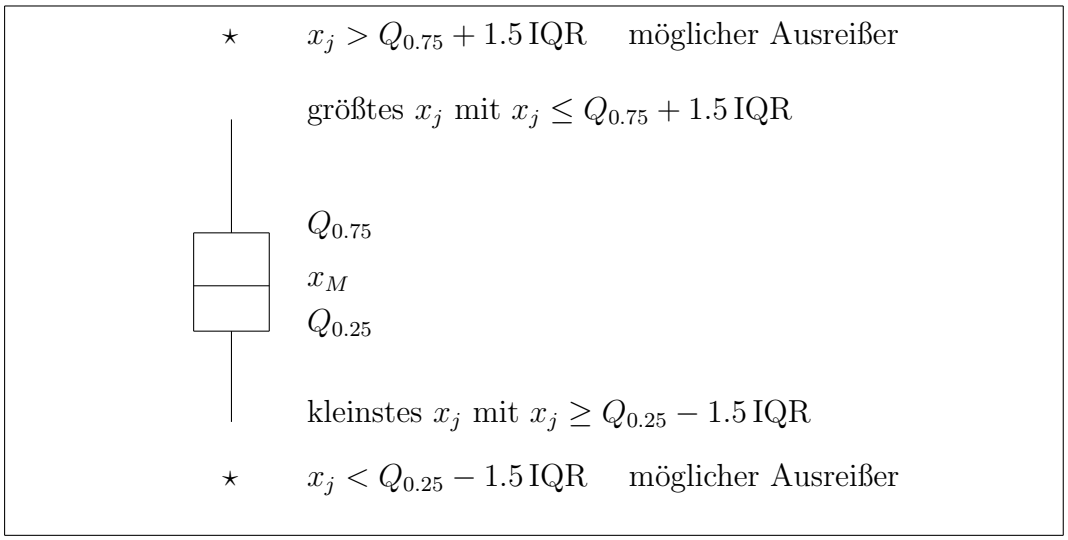


Abbildung 2.14: Definition eines Boxplots (`boxplot`)

Boxplots eignen sich als gute Ergänzung zu den früher vorgestellten Grafiken, wie Abbildung 2.15 für die Buffalo-Schneefalldaten (links) und für die logarithmierten Arsen-Daten des Kola O-Horizonts (rechts) zeigt.

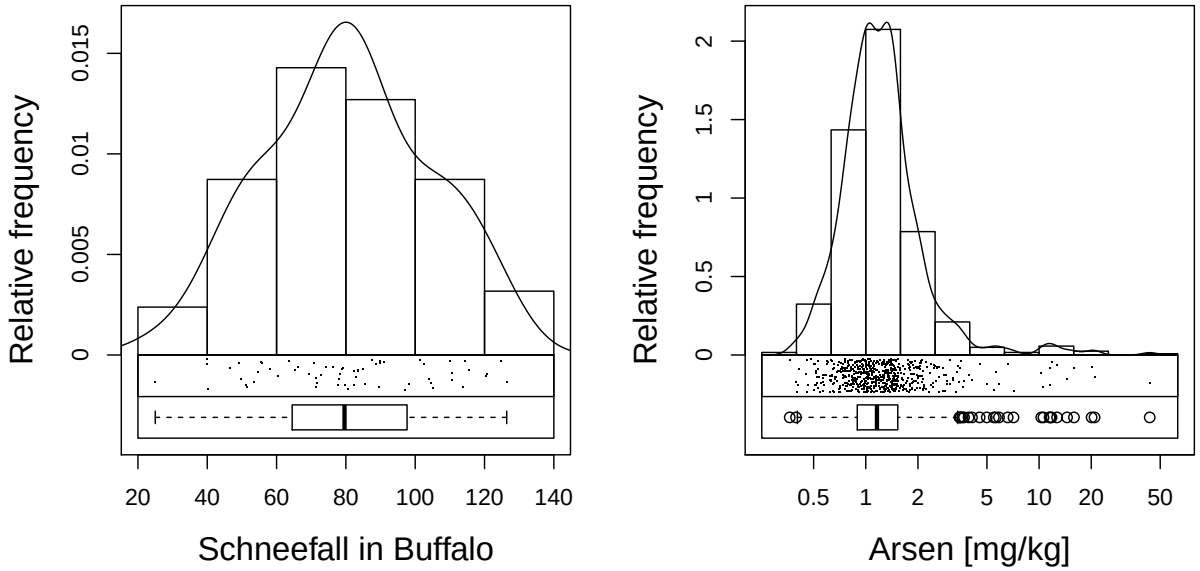


Abbildung 2.15: Kombination von Histogramm, Dichteschätzung, eindimensionaler Scatterplot und Boxplot für die Schneefalldaten von Buffalo (links) und für Arsen vom Kola O-Horizont (rechts). (`edaplot` bzw. `edaplotlog` aus `library(StatDA)`)

Beispiel 1: Boxplots sind besonders wertvoll für den visuellen Vergleich zweier oder mehrerer Datenreihen. Abbildung 2.16 zeigt einen Vergleich von Unfällen auf schwedischen Autobahnen im Jahr 1961. In diesem Jahr wurden dort Geschwindigkeitsbeschränkungen eingeführt. Ein Boxplot mit Daten vor Einführung der Beschränkung und ein Boxplot mit den Datan danach lässt gut erkennen, dass das Mittel zwar gleich bleibt, aber die Streuung sich wesentlich erhöht.

Unfälle mit Beschränkung		Unfälle ohne Beschränkung	
9	19	9	20
11	19	9	21
12	21	11	22
12	21	11	24
13	22	13	24
14	22	15	26
15	23	15	28
15	27	17	29
16	29	18	31
18	41	28	32
19	42	19	40

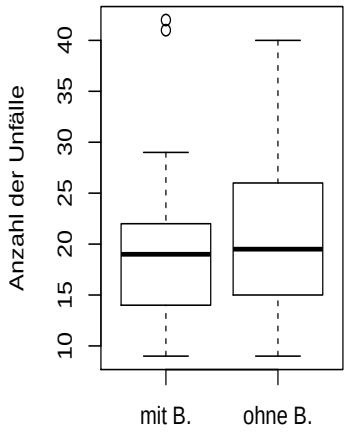


Abbildung 2.16: Anzahl der Unfälle auf schwedischen Autobahnen im Jahr 1961 an 22 hintereinanderliegenden Tagen mit bzw. ohne Geschwindigkeitsbeschränkung.

Boxplots sind durch ihre Konstruktion sehr informativ:

- *Robustheit* durch die Verwendung von Median und Quartilen.
- *Schiefe* zu beurteilen nach Lage des Medians zu den Quartilen und nach dem Auftreten von Ausreißern (d.h. wenn sie nur auf einer Seite bzw. auf einer Seite gehäuft auftreten).
- Beurteilung der *Wölbung* und des *Ausreißeranteils* (siehe Tabelle 2.2).

Verteilung	Median	oberes Quartil	Ausreißer Grenzen	Prozent außerhalb	Wert von $1.96\sigma^\dagger$	% außerhalb $\mu \pm 1.96\sigma$
symmetrische Verteilungen						
U(-1,1)	0	0.500	$\pm$2.000	0.00	1.132	0.00
N(0,1)	0	0.674	$\pm$2.698	0.70	1.960	5.00
t_{20}	0	0.687	$\pm$2.748	1.24	2.066	5.20
t_{10}	0	0.700	$\pm$2.800	1.88	2.191	5.32
t_5	0	0.727	$\pm$2.908	3.35	2.530	5.25
t_1	0	1.000	$\pm$4.000	15.59	-	-
schiefe Verteilungen						
χ^2_1	0.45	0.102 1.323	- 1.730 3.155	7.58	- 1.772 3.772	5.22
χ^2_5	4.35	2.675 6.626	- 3.252 12.552	2.80	- 1.198 11.198	4.78
χ^2_{20}	19.34	15.452 23.828	2.888 36.392	1.39	7.604 32.396	4.53

† 95% der Daten bei Normalverteilung

Tabelle 2.2: Verhalten verschiedener Verteilungen bei der Darstellung in Boxplots

Modifikationen von Boxplots:

1. Breite proportional zu $\sqrt{n}$
2. Kerben als Konfidenzintervalle.
   Kerben: $median \pm 1.57\frac{IQR}{\sqrt{n}}$ ($\alpha = 0.05$ für Tests).
   Ob man die Boxplots für Tests verwenden kann, hängt auch davon ab, ob die Varianzen der Plots mehr oder weniger stark differieren.

Beispiel 1 fortgesetzt: Abbildung 2.17 zeigt für die Unfalldaten den Vergleich mit gekerbten (*notched*) Boxplots. Nachdem sich die Kerben überlappen, kann man nicht von einem signifikanten Unterschied der Mediane ausgehen. Allerdings sollte diese Aussage eher mit Vorsicht gemacht werden, weil die Varianzen sehr unterschiedlich sind.

Beispiel 3: Im R Paket `ISLR` findet man die Daten `Carseats`. Hier wurden von 400 Geschäften die Umsatzzahlen (*Sales*) von Auto-Kindersitzen aufgezeichnet, sowie weiters u.a. die Qualität der Unterbringung der Sitze in den Regalen (*ShelveLoc*) with Ausprägung Bad/Median/Good, und No/Yes Information *Urban* (Stadt/Land), und *US* für die Lokation des Geschäftes. Diese kategoriellen Variablen können dafür verwendet werden, um die Sales-Daten zu gruppieren, und die einzelnen Datengruppen mit Boxplots zu vergleichen. Dies wurde zunächst in Abbildung 2.18 gemacht. Man kann hier die Formel-Schreibweise in R verwenden:

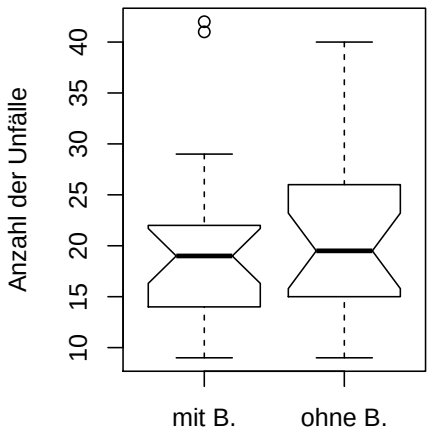


Abbildung 2.17: Boxplots mit Kerben der Unfalldaten von Abbildung 2.16.

```
library(ISLR)
data(Carseats)
attach(Carseats)
boxplot(Sales~Urban,notch=TRUE)
boxplot(Sales~US,notch=TRUE)
boxplot(Sales~ShelveLoc,notch=TRUE)
```

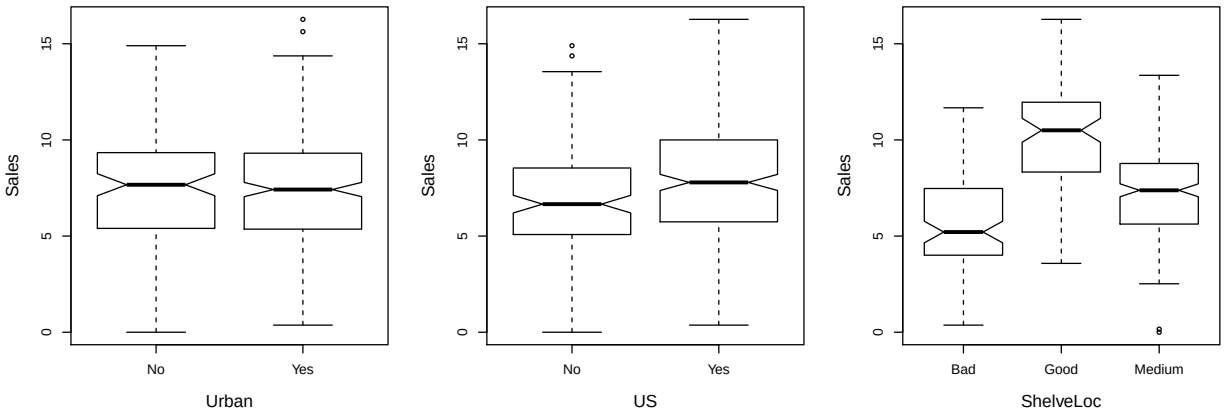


Abbildung 2.18: Vergleich der Sales-Daten von den Auto-Kindersitzen nach verschiedenen Kategorien von zusätzlichen Variablen.

Abbildung 2.18 zeigt, dass die mittleren (Median) Umsätze bei Geschäften in der Stadt bzw. am Land nicht signifikant unterschiedlich sind (links), dass man aber bzgl. Lokation US und Positionierung im Regal sehr wohl signifikante Unterschiede sieht.

Man kann auch gleichzeitig nach mehreren kategoriellen Variablen unterteilen, wie folgender Code und Abbildung 2.19 zeigen:

```
boxplot(Sales~US:ShelveLoc,notch=TRUE)
boxplot(Sales~US:Urban:ShelveLoc,notch=TRUE)
```

In Abbildung 2.19 (oben) erkennt man, dass die besten (mittleren) Umsätze für Geschäfte erzielt werden mit Standort USA, und wo die Kindersitze im Geschäft gut sichtbar plaziert sind. Die untere Grafik geht noch eine Stufe weiter: zusätzlich zur vorigen Erkenntnis hat man im städtischen Bereich bessere Umsätze als bei Geschäften am Land.

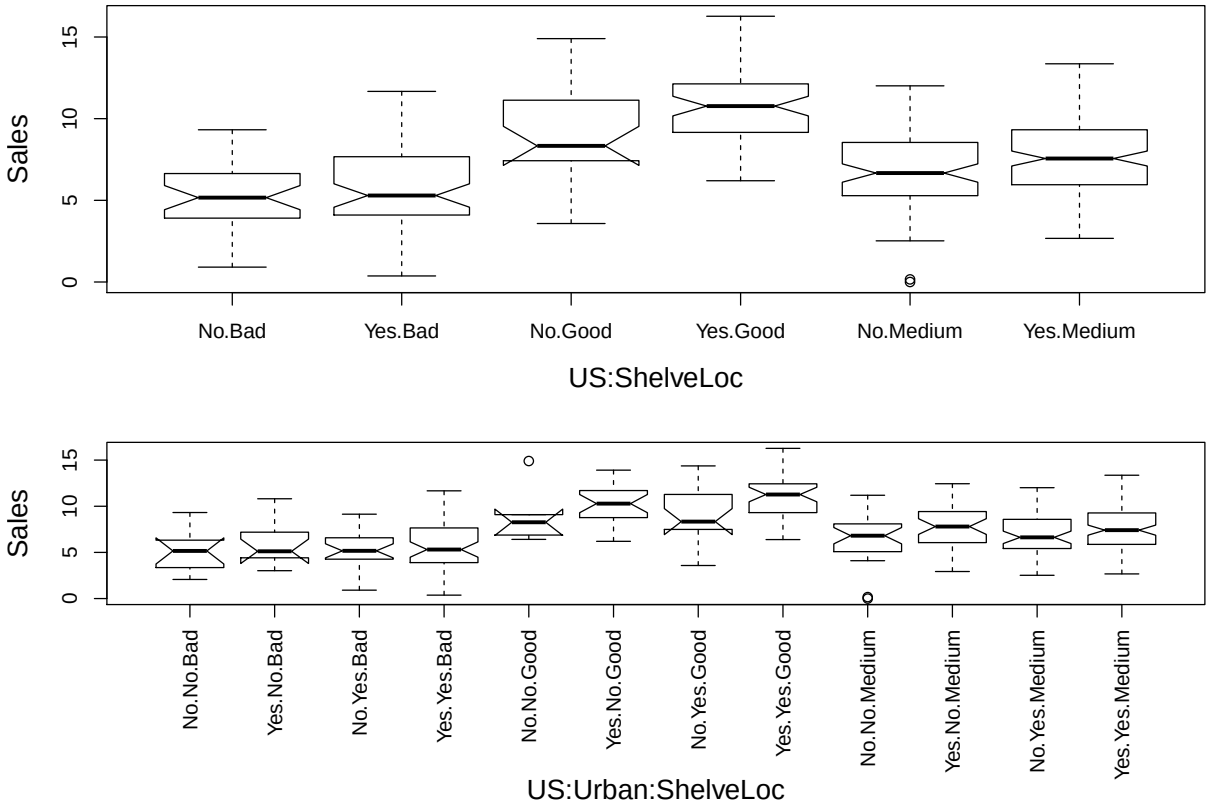


Abbildung 2.19: Vergleich der Sales-Daten von den Auto-Kindersitzen gleichzeitig nach mehreren Kategorien von zusätzlichen Variablen.

Kapitel 3

Robuste univariate Schätzer

In den vorigen Kapiteln wurde klar, dass die Lokation und die Streuung (oder Varianz) eine sehr zentrale Rolle spielen. Zum Beispiel ist es bei Daten, $x_1, \ldots, x_n$, die aus einer Normalverteilung $N(\mu, \sigma^2)$ kommen, sehr wesentlich, wie Lokation μ und Streuung σ geschätzt werden. Die Schätzung dieser Parameter hängt stark von der Datenqualität ab. Bei guter Qualität (keine Ausreißer, keine gröberen Rundungseffekte, etc.) bietet sich an, das arithmetische Mittel

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

als Schätzer für μ, und die empirische Standardabweichung

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

als Schätzer für σ zu verwenden. Wenn aber die Datenqualität schlecht ist, weil diverse Ausreißer existieren, die stark vom Hauptteil der Daten abweichen, werden diese Schätzer sehr verzerrte Ergebnisse liefern. In diesem Fall ist es besser, robuste Schätzer zu verwenden, die sich auf den (homogenen) Hauptteil der Daten konzentrieren.

Univariat heißt hier nichts anderes als *eindimensional*, dass also von einer Variable Messungen vorliegen. Im Gegensatz dazu werden wir später *multivariate* Schätzer kennenlernen, wenn also gleichzeitig mehrere Größen beobachtet werden.

3.1 Robuste Schätzung von Lokation und Streuung

Sei $x_1, \ldots, x_n$ die Stichprobe und $x_{(1)}, \ldots, x_{(n)}$ die vom kleinsten zum größten Wert sortierte Stichprobe.

Ein sehr bekannter robuster Lokationsschätzer ist das **gestutzte Mittel** (α-trimmed mean), das für $0 \leq \alpha < 0.5$ und $g = \lfloor n\alpha \rfloor$ definiert ist als:

$$m(\alpha) = \frac{1}{n - 2g}(x_{(g+1)} + \ldots + x_{(n-g)})$$

Hier bedeutet $\lfloor k \rfloor$, dass die größte ganze Zahl $\leq k$ genommen wird (abrunden).

Eine weitere Möglichkeit zur robusten Lokationsschätzung ist der **Median**:

$$median(x_1, \ldots, x_n) = \begin{cases} x_{(\frac{n+1}{2})} & \text{n ungerade} \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2} & \text{n gerade} \end{cases}$$

Über die Quartile

$$\begin{array}{lcll} Q_{0.25} & := & median(x_{(1)}, \ldots, x_{([\frac{n+1}{2}])}) & \ldots \text{ Schätzer für unteres Quartil (Quantil 0.25)} \\ Q_{0.75} & := & median(x_{([\frac{n}{2}]+1)}, \ldots, x_{(n)}) & \ldots \text{ Schätzer für oberes Quartil (Quantil 0.75)} \end{array}$$

erhält man den Interquartilabstand als robustes Streuungsmaß:

$$\text{IQR} = Q_{0.75} - Q_{0.25} \quad \ldots \quad \textbf{I}\text{nter}\textbf{q}\text{uartile } \textbf{R}\text{ange}$$

Analog zum gestutzten Mittel kann man auch eine **gestutzte Streuung** definieren:
α-trimmed standard deviation, für $0 \leq \alpha < 0.5, \quad g := \lfloor n\alpha \rfloor$

$$S(\alpha) = \sqrt{\frac{1}{n-2g-1} \sum_{i=g+1}^{n-g} (x_{(i)} - m(\alpha))^2}$$

Ein weiteres robustes Streuungmaß ist der MAD (medmed):

$$\text{MAD} = \underset{1 \leq i \leq n}{median} \; (\mid x_i - \underset{1 \leq j \leq n}{median} \; x_j \mid) \quad \ldots \quad \textbf{M}\text{edian } \textbf{A}\text{bsolute } \textbf{D}\text{eviation}$$

Der MAD ist zwar sehr robust, aber er hat eine geringe statistische Effizienz. Ein sehr robuster und effizienter Streuungsschätzer ist der Qn:

$Qn = \{|x_i - x_j|; i < j\}_{(k)}$

wobei (k) den k-ten Wert der aufsteigend sortierten Reihe darstellt, mit $k = \binom{h}{2} \approx \binom{n}{2}/4$ und $h = \lfloor n/2 \rfloor + 1$.

Weder IQR, noch $S(\alpha)$ MAD oder Qn sind dafür geeignet, den Parameter σ einer Normalverteilung zu schätzen, weil dies *keine konsistenten Schätzer* sind (die "Schätzfunktion" konvergiert nicht gegen den Parameter σ). Möchte man Konsistenz erreichen, so muss man mit Faktoren korrigieren. Konsistente Schätzer für die Standardabweichung σ sind:

1. $s_{\text{IQR}} = \frac{\text{IQR}}{1.35}$
2. $s(\alpha) = \frac{S(\alpha)}{c_\alpha}$
   Die Konstante c_α ist abhängig von α. Für $\alpha = 0.1$ ist $c_{0.10} = 0.66$.
3. $s_{\text{MAD}} = \frac{\text{MAD}}{0.675} = 1.483 \cdot \text{MAD}$
4. $s_{Qn} = 2.219 \cdot Qn$

Anmerkung: Als Schätzfunktion für die Varianz wird das Quadrat der angeführten Schätzfunktionen genommen.

Beispiel:

Stichprobe:	2.1	3.7	2.6	5.8	1.6	1.1	32.7	4.7	3.1	4.8
geordnet:	1.1	1.6	2.1	2.6	3.1	3.7	4.7	4.8	5.8	32.7

Für $\alpha = 0.1$: $g = \lfloor 10 \cdot 0.1 \rfloor = 1 \Rightarrow m(0.1) = \frac{1}{10-2}(x_{(2)} + \ldots + x_{(9)}) = \frac{1}{8}28.4 = 3.55$
$median = \frac{(3.1+3.7)}{2} = 3.4$
$Q_{0.25} = 2.1$

$Q_{0.75} = 4.8$

$s = \sqrt{\frac{1}{10-1}\sum_{i=1}^{10}(x_i - \bar{x})^2} = 9.4$

$s_{\text{IQR}} = \frac{4.8-2.1}{1.35} = 2.0$

$s(0.1) = \frac{1}{0.66}\sqrt{\frac{1}{8-1}\sum_{i=2}^{9}(x_i - m(\alpha))^2} = 2.2$

MAD:

$x_i - \underset{1 \le j \le 10}{median}\ x_j$:	-2.3	-1.8	-1.3	-0.8	-0.3	0.3	1.3	1.4	2.4	29.3
$\mid x_i - \underset{1 \le j \le 10}{median}\ x_j \mid_{(i)}$:	0.3	0.3	0.8	1.3	1.3	1.4	1.8	2.3	2.4	29.3

$\text{MAD} = \frac{1.3+1.4}{2} = 1.35$

$s_{\text{MAD}} = \frac{\text{MAD}}{0.675} = 2.0$

Qn:
absolute Differenzen $|2.1-3.7|, |2.1-2.6|, \ldots, |2.1-4.8|, |3.7-2.6|, \ldots, |3.1-4.8|$ sortieren, und den k-größten Wert nehmen, mit $k = \binom{6}{2} = 15$
Der 15. größte Wert ist $Qn = 1.5$. Somit ist $s_{Qn} = 3.33$
Die R Funktion `Qn` im Paket `robustbase` liefert den Wert 2.397, weil dort noch eine Korrektur für kleine Stichprobengröße gemacht wird.

3.2 Eindimensionale Ausreißererkennung

Boxplots eignen sich unmittelbar dazu, Ausreißer in einer Messreihe zu erkennen. Wir können aber noch weitere Regeln zur Identifikation von univariaten Ausreißern herleiten, nämlich solche, die auf robuster Lokations- und Streuungschätzung basieren. In Anlehnung an die Normalverteilung, wo im Bereich Mittel ± 2 mal Standardabweichung etwa die inneren 95% der Werte liegen, können entsprechende Regeln für den robusten Fall hergeleitet werden. Abbildung 3.1 zeigt den Vergleich solcher Regeln für simulierte normalverteilte (links) bzw. log-normalverteilte (rechts) Daten. Man bemerke, dass der Anteil der identifizerten "Ausreißer", die hier wohl eher die Extreme der Verteilungen darstellen, vom Datenumfang abhängt.

Interessant ist aber nun die Funktionsweise der Ausreißerregeln, wenn tatsächlich Ausreißer in den Daten vorhanden sind. Für Abbildung 3.2 wurden simulierte Daten mit entsprechend vielen Beobachtungen von Normalverteilung (A) bzw. log-Normalverteilung (B) genommen, bei denen aber ein variierenden Anteil aus einer anderen Verteilung kommt. Dieser Anteil kann als tatsächlicher Ausreißeranteil betrachtet werden. Mit den Ausreißerregeln kann nun verglichen werden, wie viele der simulierten Ausreißer tatsächlich erkannt werden, bzw. wie hoch der Anteil der Ausreißer sein darf, bis die Regel "zusammenbricht".

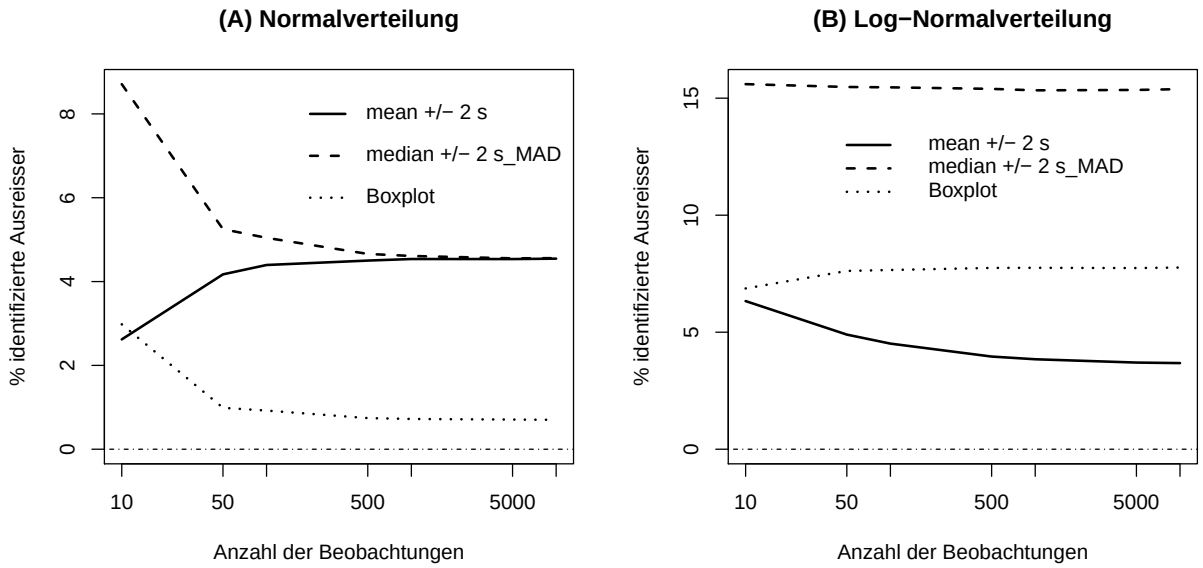


Abbildung 3.1: Anteil der identifizierten "Ausreißer" bei normalverteilten (A) und log-normalverteilten (B) Daten.

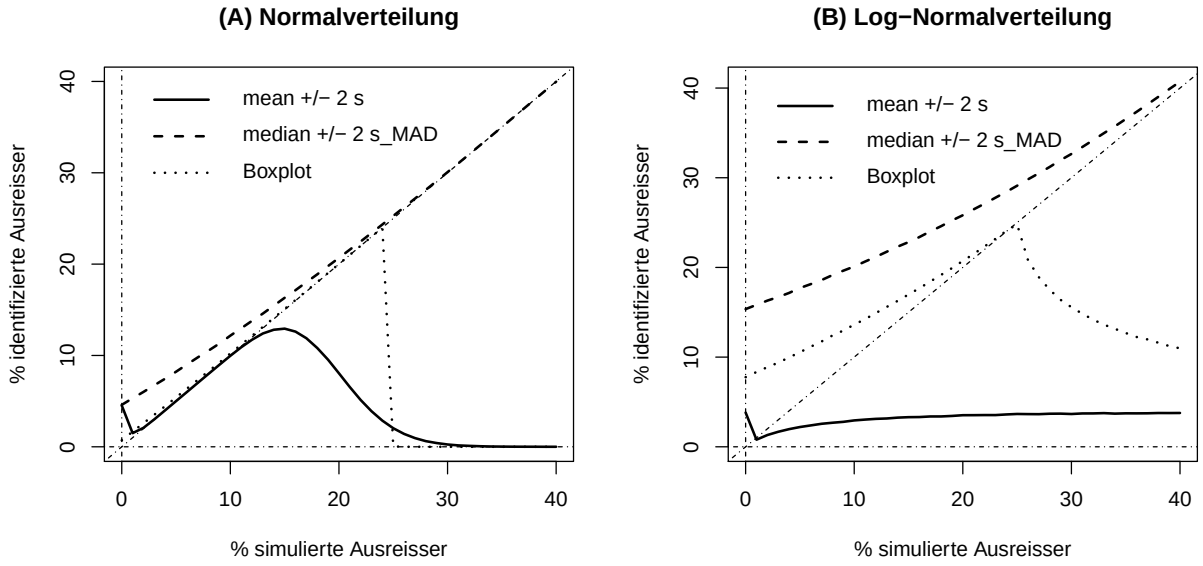


Abbildung 3.2: Anteil der identifizierten "Ausreißer" bei normalverteilten (A) und log-normalverteilten (B) Daten mit echten simulierten Ausreißern.

Kapitel 4

Darstellung zweidimensionaler Daten

4.1 Streuungsdiagramme (Scatterplots)

Es liegen Beobachtungen der beiden Variablen X und Y vor. Dabei können sowohl X als auch Y Zufallsvariable sein. Die einfachste Form eines Streuungsdiagramms ist das Zeichnen der Punktepaare (x_i, y_i) für $i = 1, \ldots, n$. Mehrfachpunkte werden auf diese Art jedoch nicht gekennzeichnet. Sie können folgendermaßen sichtbar gemacht werden:

- durch zufälliges Stören der Datenpunkte (*jittering*)

  $\tilde{x}_i := x_i + \theta_x u_i$
  $\tilde{y}_i := y_i + \theta_y v_i$

  mit u_i, v_i unabhängig stetig in (-1,1) gleichverteilte Zufallszahlen und θ_x, θ_y fest (z.B. $\theta_x = 0.02(x_{max} - x_{min})$ und $\theta_y = 0.02(y_{max} - y_{min})$).
- durch Sunflowers
  Verschlüsselung von Mehrfachpunkten durch folgende Symbole:
  1 Wert · 2 Werte | 3 Werte  4 Werte  5 Werte  usw.
- durch Einteilung in Zellen und Sunflowers
  Der Datenbereich wird in Zellen eingeteilt und die Zellenhäufigkeiten nach Transformationsvorschrift der Sunflowers dargestellt.

Beispiele: In Tabelle 4.1 sind Altersdaten von Managern, die bei *Bell Laboratories* beschäftigt sind, gegeben. Es handelt sich um zweidimensionale Daten, da das Alter im Jahr 1982 und die seit dem Studienabschluss verstrichenen Jahre gegeben sind.

In Abbildung 4.1 (links) werden die Originaldaten gezeigt. Allerdings gibt es Mehrfachpunkte, die im Plot nicht ersichtlich sind. Es wird daher eine Einteilung in Zellen vorgenommen. Die rechte Grafik zeigt die einzelnen Zellenhäufigkeiten mit Hilfe von Sunflowers dargestellt.

4.2 Streifen-Boxplots

Abgesehen vom Problem mit Mehrfachpunkten ist es oft schwierig, aus einer 2-dimensionalen Punktwolke Trends abzulesen. Als Abhilfe kann man die Daten unterteilen (z.B. in horizontaler Richtung) und mittels Boxplots darstellen. Dies ist auch ein sinnvoller Ansatz, wenn

Tabelle 4.1: Daten von Managern bei *Bell Labs*: Alter im Jahr 1982 (A) und Anzahl der Jahre (J), die seit dem Studienabschluß verstrichen sind.

A	J	A	J	A	J	A	J	A	J	A	J	A	J	A	J
35	12	42	16	44	18	47	21	50	28	54	28	57	32	59	31
36	10	42	19	44	19	47	21	51	22	54	29	57	37	59	32
36	12	43	15	44	19	47	21	51	27	54	29	58	23	59	33
36	14	43	17	44	20	47	21	52	19	54	30	58	27	59	34
37	10	43	17	45	13	47	23	52	25	55	25	58	28	59	35
37	12	43	17	45	15	47	25	52	25	55	27	58	31	60	27
38	10	43	17	45	20	47	26	52	26	55	29	58	32	60	28
38	14	43	20	45	20	48	18	53	22	55	29	58	33	60	33
39	10	43	21	45	21	48	21	53	23	55	30	58	33	61	34
39	14	44	9	45	21	48	23	53	24	55	31	58	33	61	40
39	15	44	12	46	18	48	26	53	27	55	31	58	34	62	43
40	12	44	14	46	18	49	20	53	30	55	33	58	34	62	35
40	14	44	16	46	19	49	22	53	31	55	33	59	25	63	30
41	10	44	16	46	20	50	17	53	32	56	27	59	28	63	41
41	17	44	17	46	21	50	21	54	21	56	28	59	29	64	40
42	8	44	17	47	18	50	23	54	27	56	28	59	30	64	41
42	12	44	18	47	21	50	24	54	28	56	30	59	30	66	43

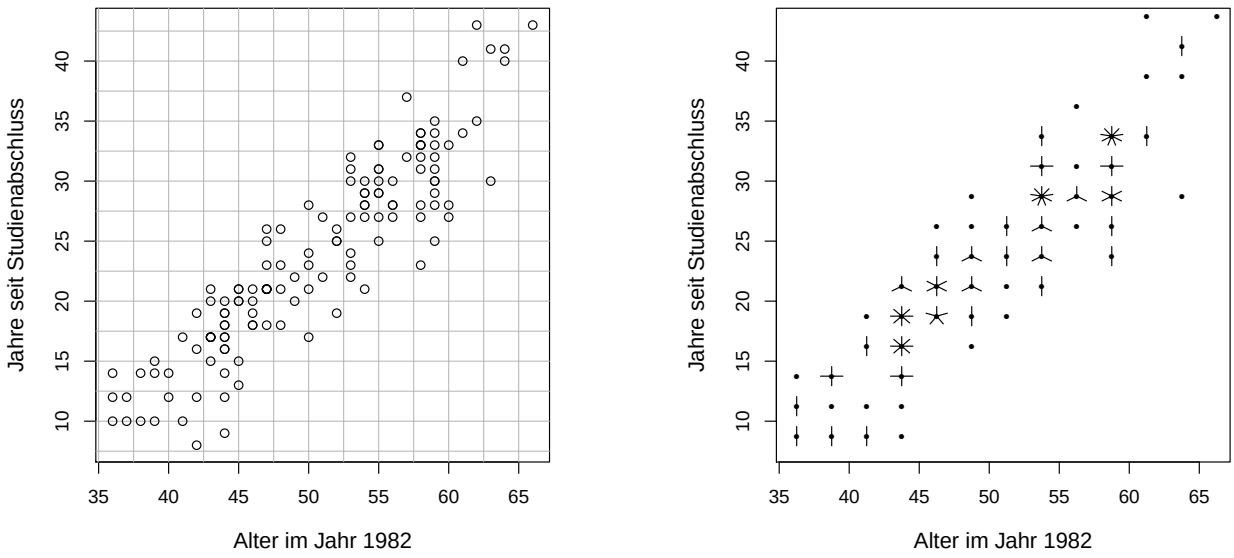


Abbildung 4.1: Scatterplot der Manager-Daten mit Zelleneinteilung (links) und und daraus resultierende Sunflowers (rechts). (`sunflowerplot` aus Paket `graphics`)

extrem viele Messungen vorliegen, sodass der Scatterplot nur mehr einen "großen schwarzen Fleck" geben würde.

Betrachten wir die Daten aus Tabelle 4.2, in der von Hamstern das erreichte Alter und der Anteil des Winterschlafs in ihrem Leben gemessen wurden.

Tabelle 4.2: Winterschlafdaten von 144 Hamstern: Alter von Hamstern zum Todeszeitpunkt (A) und Prozentanteil des Winterschlafs in ihrem Leben.

%	A	%	A	%	A	%	A	%	A	%	A
0	116	4	959	12	1124	15	1107	19	1008	23	1025
0	612	4	810	12	876	15	843	19	1174	23	760
0	711	4	678	12	843	15	760	19	1438	24	1587
0	744	6	397	12	826	15	711	20	1256	24	1504
0	760	6	496	12	793	15	512	20	1174	24	1289
0	579	6	727	12	545	16	1388	20	1140	24	1041
0	562	6	1008	12	364	16	826	20	1074	25	909
0	545	7	1058	12	264	16	810	20	810	25	1190
0	496	8	876	13	1289	16	777	20	711	25	1207
0	364	8	975	13	1140	16	331	21	1223	25	1256
0	314	9	975	13	678	17	760	21	1140	25	1587
1	826	9	810	13	446	17	893	21	992	26	760
1	975	9	711	14	1289	17	909	21	942	26	1091
1	893	9	678	14	1273	17	1289	21	860	27	1372
1	826	9	446	14	1157	18	1289	21	843	28	264
1	727	10	579	14	1132	18	1207	22	645	28	1107
1	678	10	694	14	1124	18	1124	22	727	28	1124
1	579	10	810	14	1107	18	1058	22	1107	29	760
1	430	11	1107	14	893	18	1041	23	1421	29	1107
2	826	11	826	14	884	18	1008	23	1306	29	1273
2	860	11	760	14	876	18	909	23	1273	29	1620
2	1074	11	744	14	860	18	860	23	1256	30	760
3	760	11	727	14	760	18	562	23	1174	32	1355
3	975	12	1207	15	1223	19	545	23	1074	33	1074

Die Beobachtungen werden in der linken Grafik von Abbildung 4.2 dargestellt. Im rechten Plot wurden die Daten in senkrechte Streifen eingeteilt (man könnte auch waagrechte Streifen wählen), wobei jeder Streifen annähernd gleich viele Datenpunkte enthalten soll.

Die Beobachtungen jedes Streifens sollten nun mit Boxplots dargestellt werden. In der linken Grafik von Abbildung 4.3 werden nun für die Beobachtungen eines Streifens die Mediane in y-Richtung berechnet und mit Linien dargestellt. Die Länge dieser Linien in x-Richtung entspricht dabei der Länge der durch die Streifen getrennten Bereiche, und die Position der Linien in y-Richtung den Medianen. Man erkennt auch punktierte vertikale Linien im Plot: Die horizontale Position dieser Linien entspricht dem Median (in x-Richtung) der einzelnen Bereiche (die Länge in y-Richtung hat hier keine Bedeutung).

In der rechten Grafik von Abbildung 4.3 werden nun für die einzelnen durch die Streifen getrennten Bereiche die Boxplots dargestellt. Die Boxplots sind in horizontaler Richtung zentriert an den in der linken Grafik dargestellten punktierten Linien.

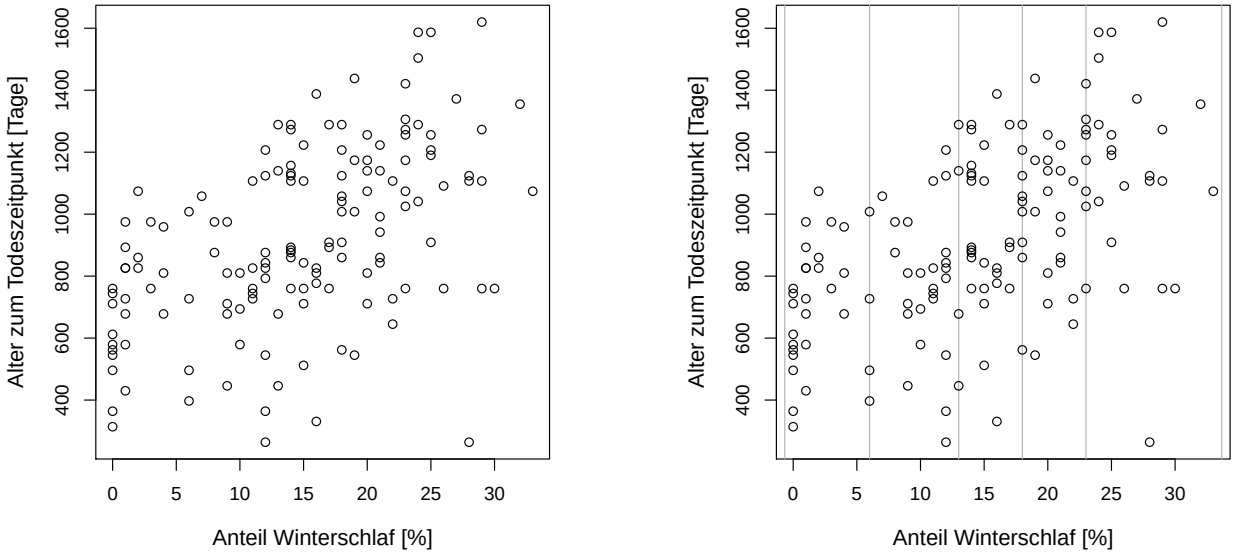


Abbildung 4.2: Einteilung der Hamsterdaten aus Tabelle 4.2 in Streifen. Die Bereiche sollten annähernd gleiche Anzahl von Datenpunkten enthalten.

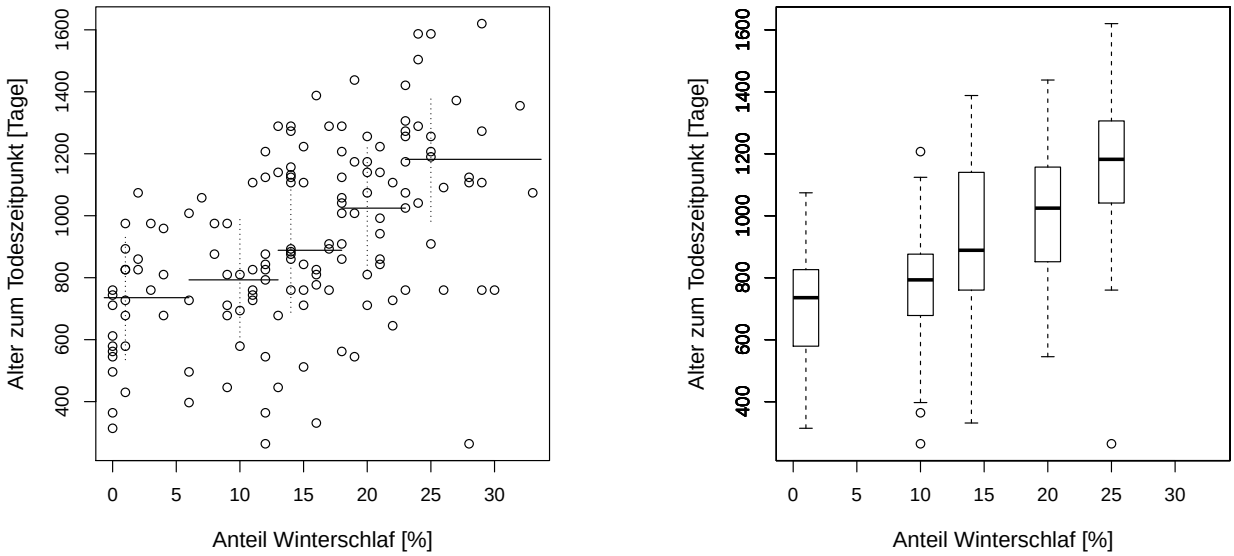


Abbildung 4.3: Scatterplot der Hamsterdaten aus Tabelle 4.2 mit Streifenmedianen (links) und Darstellung der Daten als Streifen-Boxplots (rechts).

Mit den Streifen-Boxplots in Abbildung 4.3 kann man gut Trends erkennen. Man sieht deutlich, dass mit zunehmendem Anteil von Winterschlaf auch das Alter der Hamster zunimmt. Man bemerke, dass auch bereits die linke Grafik in Abbildung 4.3 diesen Trend gut erkennen lässt, und dass zusätzlich auch die Daten dargestellt werden können.

Für eine objektivere Darstellung könnte die Anzahl der Streifen variiert werden.

4.3 Dichteschätzung in zwei Dimensionen

Ähnlich wie im eindimensionalen Fall kann auch im zweidimensionalen Fall eine Dichtefunktion geschätzt werden. Das Grundprinzip ist wiederum, einen Teilbereich auszuwählen (quadratisch, kreisförmig), in dem eine vorgegebene Gewichtsfunktion evaluiert wird. Die Gewichtsfunktion kann die Gestalt einer *Boxcar Funktion* haben:

$$W(u,v) = \begin{cases} \frac{1}{\pi} & u^2+v^2 \leq 1 \\ 0 & \text{sonst} \end{cases}$$

Es gilt wieder $\int W(u,v)dudv = 1$. Die *lokale Dichte* hat dann die Gestalt:

$$\hat{f}(x,y) = \frac{1}{h^2 n}\sum_{i=1}^{n} W(\frac{x-x_i}{h}, \frac{y-y_i}{h}) \quad \text{mit } h = \text{Fensterbreite bzw. Fensterhöhe}$$

Glattere Dichten können mit der cosinus-Gewichtsfunktion

$$W(u,v) = \begin{cases} \frac{1+\cos(\pi\sqrt{u^2+v^2})}{\pi} & u^2+v^2 \leq 1 \\ 0 & \text{sonst} \end{cases}$$

erzielt werden.
Darstellung von $\hat{f}(x,y)$ durch

1. Graustufen oder Farbabstufungen
2. Isolinien
3. Gebirge
4. ...

Als Beispiel betrachten wir die *Old Faithful Geyser* Daten, die die Aktivität heißer Quellen im Yellowstone National Park im Zeitraum von 1.-15. August 1985 beschreiben. Es liegen 299 Beobachtungen für die Eruptionszeit (in Minuten) und die Wartezeit bis zur nächsten Eruption vor. (Die Daten sind in `R` in der *library(MASS)* unter *data(geyser)* verfügbar.) In Abbildung 4.4 werden die Originaldaten dargestellt. Für die nachfolgenden Darstellungen wurde als Gewichtsfunktion die bivariate Dichtefunktion der Normalverteilung gewählt. Fensterbreite und Fensterhöhe wurden nach einem speziellen Verfahren, das hier nicht weiter erwähnt wird, gewählt. Die zugehörige R Funktion heißt `kde2d` vom Paket `MASS`.

Abbildung 4.5 zeigt die Darstellung der zweidimensionalen Dichtefunktion mittels Graustufen (links) und Isolinien (rechts). In Abbildung 4.6 wird die Dichtefunktion räumlich durch Gebirge dargestellt.

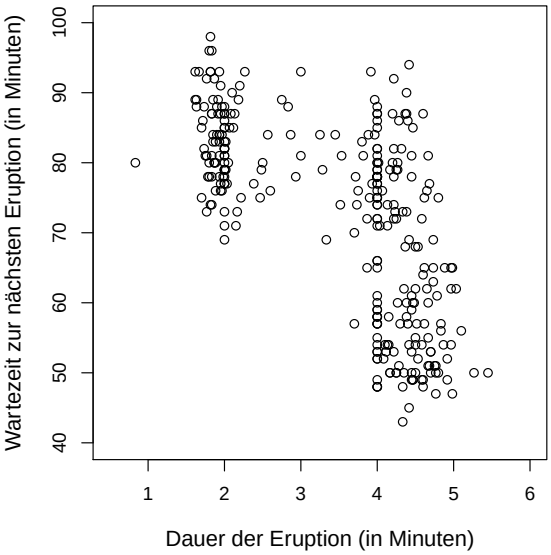


Abbildung 4.4: *Old Faithful Geyser* Daten

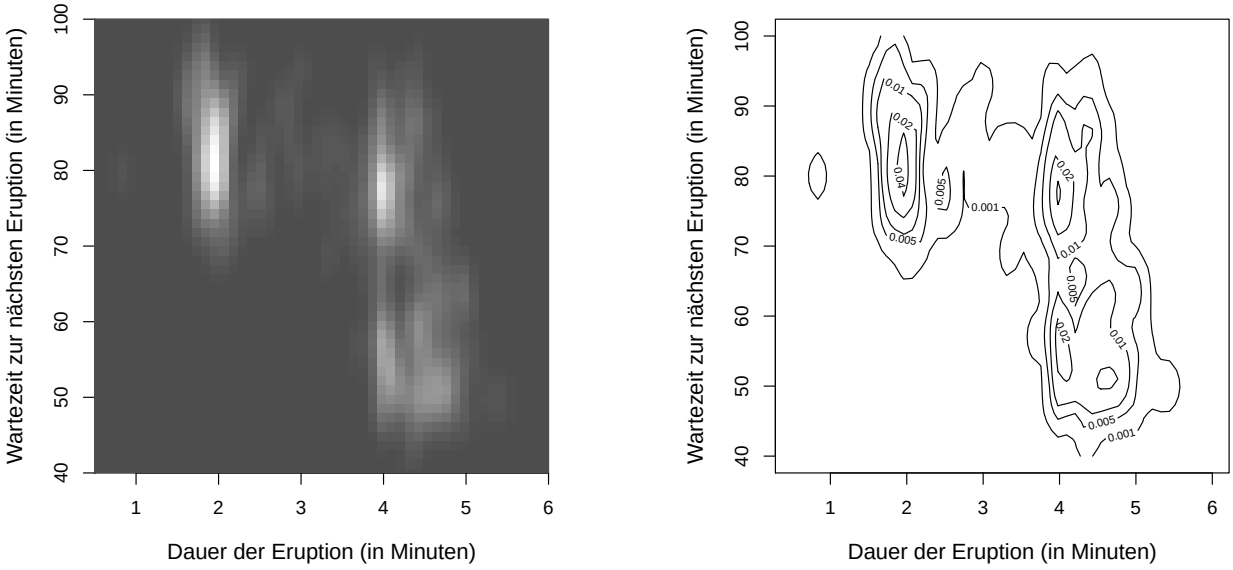


Abbildung 4.5: Darstellung der zweidimensionalen Dichtefunktion mittels Graustufen (links) und Isolinien (rechts). (`image` bzw. `contour`)

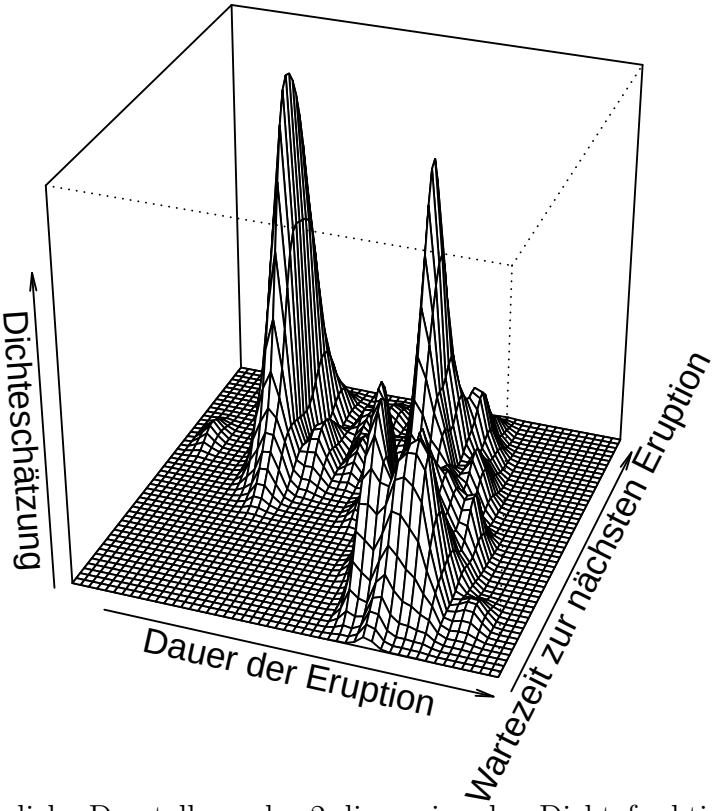


Abbildung 4.6: Räumliche Darstellung der 2-dimensionalen Dichtefunktion. (`persp`)

Kapitel 5

Robuste Schätzung linearer Trends

Dieses Problem ist in der Literatur als lineare Regression bekannt. Es soll aufgrund einer *Input*-Größe x (unabhängige Variable) eine *Output*-Größe y (abhängige Variable) vorhergesagt werden, mit der linearen Funktion

$$y = \alpha + \beta x + \varepsilon \ ,$$

wobei ε einen zufälligen Fehler darstellt. Der Abszissenabstand α und die Steigung β sind dabei die Parameter einer Geraden.

Liegt nun eine Stichprobe von Umfang n vor, also $(x_1, y_1), \ldots, (x_n, y_n)$, so sollen die Parameter der Geraden geschätzt werden, also Koeffizienten $\hat{\alpha}$ und $\hat{\beta}$, sodass die resultierende Gerade "bestmöglich" durch die Punktepaare geht:

$$y_i \approx \hat{\alpha} + \hat{\beta} x_i \qquad i = 1, \ldots, n$$

Mit den geschätzten Parametern erhält man auch die prognostizierten y-Werte

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i \qquad i = 1, \ldots, n,$$

die jetzt natürlich genau auf einer Gerade liegen. Die bei der Prognose entstehenden Fehler nennt man Residuen

$$r_i = y_i - \hat{y}_i = y_i - \hat{\alpha} - \hat{\beta} x_i \qquad i = 1, \ldots, n.$$

Zur Schätzung der Parameter α und β wird meist das kleinste Quadrate oder Least Squares (LS) Kriterium verwendet:

$$(\hat{\alpha}_{LS}, \hat{\beta}_{LS}) = \text{argmin}_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 = \text{argmin}_{\alpha,\beta} \sum_{i=1}^{n} r_i^2 \ .$$

Als Lösung für die Regressionsparameter ergibt sich:

$$\hat{\beta}_{LS} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \text{und} \quad \hat{\alpha}_{LS} = \bar{y} - \hat{\beta} \bar{x}$$

Obwohl die LS-Schätzer unter bestimmten Voraussetzungen gute statistische Eigenschaften aufweisen, gibt es doch den gravierenden Nachteil, dass sie empfindlich gegenüber Ausreißern sind. Ausreißer in y-Richtung, speziell aber Ausreißer in x-Richtung können massiven Einfluss auf die Parameterschätzung haben. Der Grund liegt am Kriterium. Abweichende Werte haben großes Residuenquadrat, und das Minimum von obigem Kriterium wird kleiner, wenn die Residuen "gleichmäßiger" aufgeteilt werden (siehe Abbildung 5.1).

In den nachfolgenden Kapiteln werden daher Regressionsmethoden besprochen, die robuster gegenüber Ausreißern sind.

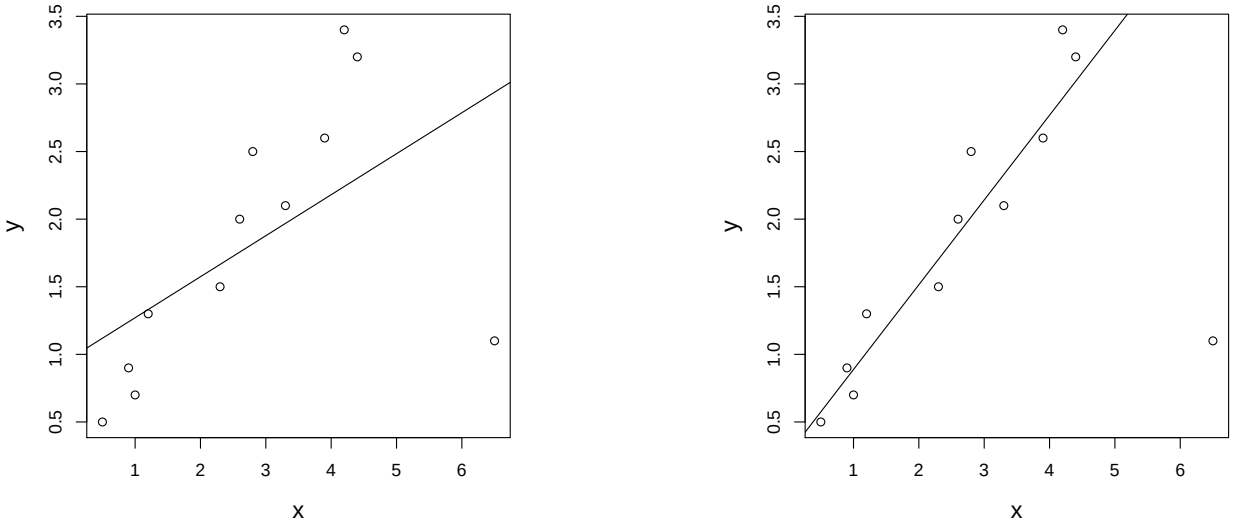


Abbildung 5.1: Die übliche kleinste Quadrate Regressionsgerade (links) ist empfindlich gegenüber Ausreißern. Die hypothetische Gerade im rechten Plot ist robust. (`lm`)

5.1 Robuste Gerade nach Tukey

Es handelt sich dabei um ein iteratives Verfahren, das von Tukey (1970) entwickelt wurde, und das folgendermaßen definiert ist:

1. Sortieren der Datenpaare nach den x-Werten.
2. Einteilung der Datenpaare in 3 Gruppen.
   Gruppe L (links) Paare (x_i, y_i) mit den n_L kleinsten x-Werten
   Gruppe M (Mitte) Paare (x_i, y_i) mit den n_M mittleren x-Werten
   Gruppe R (rechts) Paare (x_i, y_i) mit den n_R größten x-Werten

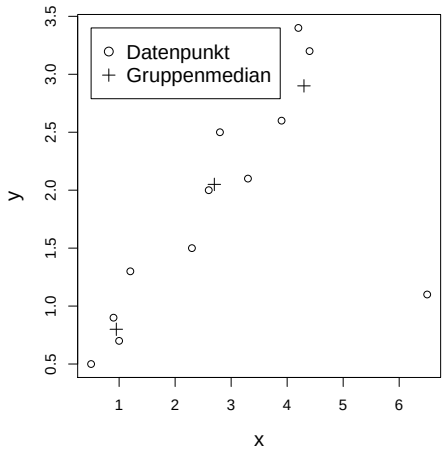
   $n_L + n_M + n_R = n$.

   Richtlinie:

	$n = 3k$	$n = 3k+1$	$n = 3k+2$
n_L	k	k	$k+1$
n_M	k	$k+1$	k
n_R	k	k	$k+1$

   Paare mit gleichen x-Werten sollten derselben Gruppe zugeordnet werden.

3. Berechnung der Mediane für x-Werte und y-Werte in den einzelnen Gruppen:
   (x_L, y_L), (x_M, y_M), (x_R, y_R) mit
   $x_L = \underset{(x_i,y_i)\epsilon L}{median}\ x_i$, $y_L = \underset{(x_i,y_i)\epsilon L}{median}\ y_i$, usw.



4. $\hat{\beta}_0 := \frac{y_R - y_L}{x_R - x_L}$ erster Schätzwert für β

$$y = \alpha + \beta(x - x_M) \quad \Rightarrow \quad \alpha = y - \beta(x - x_M)$$

$$\begin{aligned}\hat{\alpha}_0^{(*)} &= \frac{1}{3}[(y_L - \hat{\beta}_0(x_L - x_M)) + y_M + (y_R - \hat{\beta}_0(x_R - x_M))] \\ &= \left(\frac{1}{3}(y_L + y_M + y_R) - \hat{\beta}_0\frac{1}{3}(x_L + x_M + x_R)\right) + \hat{\beta}_0 x_M := \hat{\alpha}_0 + \hat{\beta}_0 x_M\end{aligned}$$

5. Residuen
   $r_i^{(0)} := y_i - (\hat{\alpha}_0^{(*)} + \hat{\beta}_0(x_i - x_M)) \qquad$ für $i = 1, \ldots, n$
6. Schritte 3-5 mit den Datenpaaren $(x_i, r_i^{(0)}) \quad i = 1, \ldots, n$ ergibt die Gerade $r_i^{(0)} = \hat{\alpha}_1 + \hat{\beta}_1(x_i - x_M)$ und die Residuen $r_i^{(1)} := r_i^{(0)} - (\hat{\alpha}_1 + \hat{\beta}_1(x_i - x_M))$
7. Iteration: alle weiteren Schritte wie Schritt 6, d.h.
   $r_i^{(j)} \to \hat{\alpha}_{j+1}, \hat{\beta}_{j+1} \to r_i^{(j+1)}$ bis ein geeignetes Abbruchkriterium erfüllt wird.
8. $\hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 + \cdots$
   $\hat{\alpha} = \hat{\alpha}_0 + \hat{\alpha}_1 + \hat{\alpha}_2 + \cdots$

Anmerkung: Schwachstellen des Verfahrens

- eventuell sehr langsame Konvergenz bei "schlechter" Konfiguration der Daten.
- manchmal keine Konvergenz $\longrightarrow$ kann aber durch geringfügige Modifikation des Algorithmus repariert werden.

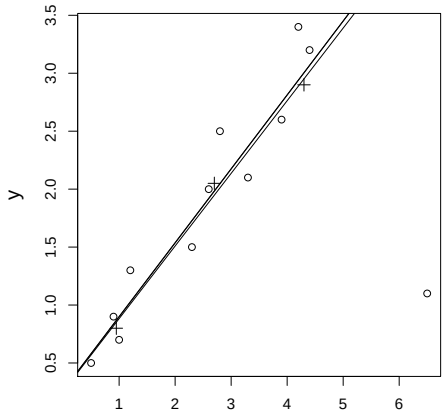


Abbildung 5.2: Robuste Gerade nach Tukey nach 3 Iterationsschritten. (`line`)

5.2 Robuste Gerade nach Theil

Theil (1950) berücksichtigt zur Berechnung der Geraden die Mediane paarweiser Anstiege. Voraussetzung für das Verfahren ist, dass alle x_i verschieden sind.
$\hat{\beta}_{ij} := \frac{y_j - y_i}{x_j - x_i} \qquad 1 \le i < j \le n \qquad$ d.h. $\binom{n}{2} = \frac{n(n-1)}{2} \quad$ Anstiege.
$\hat{\beta}_T := \underset{1 \le i < j \le n}{median} \ (\hat{\beta}_{ij})$

Der "Grad der Robustheit" von verschiedenen Methoden kann verglichen werden. Man kann sogar i.A. für jeden Schätzer (nicht nur für die Parameter bei Regression) die Robustheit angeben, also die Empfindlichkeit gegenüber einen Anteil an Ausreißern. Dieses Maß nennt man den **Bruchpunkt** für einen Schätzer.

n ... Stichprobenumfang

k ... maximale Anzahl von Stichprobenelementen, die durch beliebige Werte ersetzt werden können, ohne dass dadurch der Schätzwert unbeschränkt wird.

Der Bruchpunkt eines Schätzers ist definiert als jener *minimale* Anteil $\frac{k}{n}$ der Daten, der durch beliebige Werte ersetzt werden kann, sodass dadurch der Schätzer unsinnige Resultate liefert.

Bruchpunkte für verschiedene Verfahren:

Stichprobenmittel	0.00
oberes Quartil	0.25
unteres Quartil	0.25
Median	0.50
IQR	0.25
robuste Gerade nach TUKEY	$\frac{1}{6}$
robuste Gerade nach THEIL	0.29
robuste Gerade nach SIEGEL	0.50

Wenn der Ausreißeranteil kleiner als dieser Wert ist, wird der Schätzer sich zwar verändern, aber es werden noch sinnvolle Ergebnisse zu erwarten sein. Im Spezialfall kann jedoch ein weitaus größerer Ausreißeranteil auftreten ohne massiven Einfluss auf den Schätzwert zu nehmen, z.B. bei IQR kann sogar ein Ausreißeranteil bis zu 50% mit beschränktem Einfluss bleiben, wenn sich die "schlechten" Datenwerte gleichmäßig auf beide Enden der geordneten Stichprobe aufteilen. Ber der Definition des Bruchpunktes meint man aber eine Kontamination der Daten, die so angelegt ist, dass sie möglichst viel "Schaden" bewirken kann (z.B. nur auf einer Seite des Wertebereiches).

Bruchpunkt für die robuste Gerade nach Theil:

$\binom{k}{2} + k(n-k) = \frac{\binom{n}{2}}{2}$

$\binom{k}{2}$ ... Anzahl der Anstiege mit beiden Endpunkten unbrauchbar.

$k(n-k)$ ... Anzahl der Anstiege mit nur 1 Endpunkt unbrauchbar.

$\binom{n}{2}$ ... Anzahl der Geraden, Bruchpunkt für den Median = 0.5

$$\begin{aligned}
\frac{k(k-1)}{2} + k(n-k) &= \frac{n(n-1)}{4} \\
k^2 - k + 2nk - 2k^2 &= \frac{n(n-1)}{2} \\
-k^2 + 2nk - k &= \frac{n^2(1-\frac{1}{n})}{2} \\
\left(\frac{k}{n}\right)^2 - 2\frac{k}{n} + \frac{k}{n^2} &= -\frac{1}{2} + \frac{1}{2n} \\
\left(\frac{k}{n}\right)^2 - 2\frac{k}{n} + \frac{1}{2} &\approx 0 \qquad \text{für große } n \\
\frac{k}{n} &= 1 \pm \sqrt{1 - \frac{1}{2}} \\
\frac{k}{n} &= 1 - 0.71 \quad = \quad 0.29
\end{aligned}$$

5.3 Robuste Gerade nach Siegel (Repeated Median Line)

Der Schätzer von Siegel (1982) basiert ebenfalls auf Medianen von paarweisen Anstiegen, diese werden aber wiederholt berechnet:

$\hat{\beta}_{RM} := \underset{1 \le i \le n}{median} \left(\underset{\substack{1 \le j \le n \\ j \ne i}}{median} (\hat{\beta}_{ij}) \right)$ d.h. bei der Medianberechnung für $\hat{\beta}_{RM}$ scheint die Steigung zwischen (x_i, y_i) und (x_j, y_j) 2-mal auf.

$\hat{\alpha}_{RM} := \underset{1 \le i \le n}{median} (y_i - \hat{\beta}_{RM} x_i)$

Bruchpunkt: 0.5

Beweis: für $n = 2k$

Sind $k-1$ Elemente Ausreißer, so ist $\underset{j \ne i}{median} (\hat{\beta}_{ij})$ für $k-1$ Indizes $i_1, \ldots, i_{k-1}$ Ausreißer. Ist (x_i, y_i) ein "guter" Datenpunkt, so ist auch $\underset{j \ne i}{median} (\hat{\beta}_{ij})$ "gut", da nur $k-1$ $(< \frac{n}{2})$ $\hat{\beta}_{ij}$ Ausreißer sind.

$\Rightarrow$ in $\underset{1 \le i \le n}{median} \left(\underset{\substack{1 \le i \le n \\ j \ne i}}{median} (\hat{\beta}_{ij}) \right)$ sind nur $k-1$ $(< \frac{n}{2})$ Werte Ausreißer.

$\lim_{n \to \infty} \frac{k-1}{n} = \frac{k-1}{2k} = \frac{1}{2}$

Ein analoger Beweis gilt für $n = 2k+1$.

Anmerkung: Größere Robustheit (= Resistenz) wird durch größeren Rechenaufwand erkauft.

5.4 Least Median of Squares (LMS) Regression

Während bei LS-Regression die *Summe* der quadrierten Residuen minimiert wird, ist das LMS Kriterium (Rousseeuw, 1984) definiert als

$$(\hat{\alpha}_{LMS}, \hat{\beta}_{LMS}) = \text{argmin}_{\alpha,\beta} \, \text{median}_i r_i^2 \; .$$

Man ersetzt also die Summe durch den Median, wodurch eine starke Robustheit erzielt wird (Bruchpunkt 50%).

Die Lösung für $\hat{\alpha}_{LMS}$ und $\hat{\beta}_{LMS}$ wird mit einem "subsampling" Algorithmus gefunden, siehe unten.

5.5 Least Trimmed Squares (LTS) Regression

Bei LTS Regression (Rousseeuw, 1984) wird nun versucht, nur die Summe der kleinsten quadrierten Residuen zu minimieren. Seien $r^2_{(1)} \le r^2_{(2)} \le \ldots \le r^2_{(n)}$ die der Größe nach geordneten quadrierten Residuen. Dann lautet das LTS Kriterium

$$(\hat{\alpha}_{LTS}, \hat{\beta}_{LTS}) = \text{argmin}_{\alpha,\beta} \sum_{i=1}^{h} r^2_{(i)}$$

mit $\frac{n}{2} < h < n$. Je nach Wahl von h erhält man eine Methode mit Bruchpunkt zwischen 0 und 50%. Mit geringerem Bruchpunkt leidet zwar die Robustheit, aber die Effizienz (Präzision des Schätzers) wird erhöht.

Anmerkung: Für ein einfaches Regressionsproblem (x gegen y) scheint der Vorteil von robuster Regression oft nicht unmittelbar ersichtlich, weil Ausreißer im zweidimensionalen

Raum noch ohneweiters grafisch erkannt werden können. Alle oben erwähnten Methoden funktionieren aber auch im Fall von mehreren x-Variablen (multiple Regression), und dann ist die grafische Ausreißererkennung i.A. nicht mehr möglich.

Algorithmus: Sowohl für LMS als auch für LTS Regression wurde ein Algorithmus entwickelt, der auf "Subsampling" basiert. Wir nehmen im folgenden an, dass $p \geq 1$ x-Variablen vorliegen, die zur Erklärung von y verwendet werden können:

1. Wähle zufällig $p + 1$ Beobachtungen aus. Durch diese wird die Regressionsgerade (Regressionshyperebene) exakt bestimmt.
2. Berechne alle Residuen und daraus das LMS- bzw. LTS-Kriterium.
3. Iteriere 1. und 2. "sehr oft", und wähle als endgültige (approximative) Lösung jene, die den kleinsten Wert der Zielfunktion ergibt.

Der Rechenaufwand steigt vor allem mit größerem p.

Fast-LTS Algorithmus: Für LTS-Regression wurde ein schnellerer Algorithmus entwickelt:

1. Wähle zufällig h Beobachtungen.
2. Schätze mit den h Beobachtungen die Regressionsparameter mittels LS-Regression (Least-Squares – nicht robust!).
3. Berechne die Residuen von *allen* Beobachtungen und ordne deren Absolutbeträge der Größe nach.
4. Nehme jene h Beobachtungen, die von den kleinsten (absoluten) Residuen aus Punkt 3. kommen.
5. Iteriere 2.-4. bis Konvergenz. Diese ist gesichert, weil die Summe der h gewählten quadrierten Residuen nie größer wird.
6. Führe Schritte 1.-5. mehrmals durch. Die approximative Lösung ist jene, die den kleinsten Wert der Zielfunktion ergibt.

R-code: In der `library(rrcov)` ist der Fast-LTS Algorithmus in der Funktion `ltsReg` implementiert. Die `library(robustbase)` enthält die Funktion `lmrob`, die von der Input/Output-Struktur der Funktion `lm` für klassische LS-Schätzung nachgebaut ist. `lmrob` führt robuste MM-Regression durch, ein Verfahren, das bestmögliche Robustheit bei optimaler Präzision erzielt.

5.6 Mehrere x-Variablen

Im allgemeineren Fall möchte man eine *Output*-Größe y nicht nur durch eine *Input*-Größe x erklären, sondern es liegen mehrere *Input*-Größen $x_1, x_2, \ldots, x_p$ vor. Diese p *Inputs* könnten p verschiedene Variablen sein, die in einem Produktionsprozess gemessen werden, um schließlich die Produktqualität y vorherzusagen. Nimmt man wiederum eine lineare Vorhersagefunktion der *Inputs*, dann hat das Regressionsmodell folgende Form:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

Hier ist α wieder der Abszissenabstand, die βs sind die Steigungsparameter für die einzelnen *Inputs*, und ε ist der zufällige Fehler.

Für jede dieser Größen liegen dann n Beobachtungen vor, und man hat somit die Stichprobe $(x_{i1}, x_{i2}, \ldots, x_{ip}, y_i)$ für $i = 1, \ldots, n$. Mit den geschätzten Paramtern $\hat{\alpha}$, $\hat{\beta}_1, \ldots, \hat{\beta}_p$ erhält man wieder die prognostizierten Werte

$$\hat{y}_i = \hat{\alpha} + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip}$$

und die Residuen

$$r_i = y_i - \hat{y}_i \quad \text{für} \quad \text{i=1,...,n.}$$

Wie bereits im einfachen linearen Regressionsfall (eine x-Variable) kann man auch hier unterschiedliche Kriterien heranziehen, um zu Schätzungen für die Parameter zu kommen. Das LS-Kriterium wäre naheliegend, wiederum aber mit dem Nachteil, dass es empfindlich gegenüber Ausreißern ist. Man bemerke hier auch, dass Ausreißer in x-Richtung (sogenannte Hebelpunkte) nun Aureißer in p Dimensionen sein können, also in den Variablen $x_1, \ldots, x_p$, und daher durch grafische Mittel schwer oder gar nicht zu finden sind.

Nicht alle der in den vorangehenden Kapiteln behandelten Methoden können hier als robuste Variante herangezogen werden, aber LMS- bzw. LTS-Regression, sowie auch die erwähnte MM-Regression bieten sich bestens an. Die Kriterien sind exakt die gleichen, nur die Algorithmen zur Schätzung der Parameter werden etwas komplexer.

Beispiel: Wir betrachten die Daten *rice* vom Paket *rrcov* mit subjektiven Evaluierungen von 105 verschiedenen Reissorten. Die y-Variable stellt die gesamte Evaluierung (overall evaluation) dar, die x-Variablen sind Aroma (flavor), Erscheinung (appearance), Geschmack (taste), Klebrigkeit (stickiness) und Zähheit (toughness). Man möchte nun ein Modell erhalten, um die Qualität aufgrund der x-Variablen prognostizieren zu können.

Wir betrachten zunächst nur einfache lineare Regression, mit der x-Variable Aroma. Als Methoden werden LS-Regression sowie die robuste MM-Regression verglichen. Die resultierenden Regressionsgeraden werden in Abbildung 5.3 links dargestellt. Offenbar sind die geschätzten Koeffizienten der beiden Varianten einander sehr ähnlich. Es scheint keine gröberen Ausreißer zu geben, die LS-Regression stärker beeinflussen würden.

```
library(robustbase)
library(rrcov)
data(rice)
attach(rice)

plot(Favor,Overall_evaluation)
r1 <- lm(Overall_evaluation~Favor,data=rice)
abline(r1,col="red")
r2 <- lmrob(Overall_evaluation~Favor,data=rice)
abline(r2,col="blue")
legend("topleft",legend=c("LS-Regression","MM-Regression"),
       lty=c(1,1),col=c("red","blue"))
```

Abbildung 5.3 rechts zeigt das Ergebnis der klassischen und robusten Regression für zwei erklärende Variablen. Auch hier ist kein großer Unterschied zu erkennen, wie man an den geschätzten Regressionskoeffizienten sieht:

```
mod <- lm(Overall_evaluation~Favor+Appearance, data=rice)
coef(mod)
(Intercept)       Favor  Appearance
 -0.1385352   0.5041728   0.7237637
```

```
mod2 <- lmrob(Overall_evaluation~Favor+Appearance, data=rice)
coef(mod2)
(Intercept)       Favor  Appearance
 -0.1582099   0.4895092   0.7668813
```

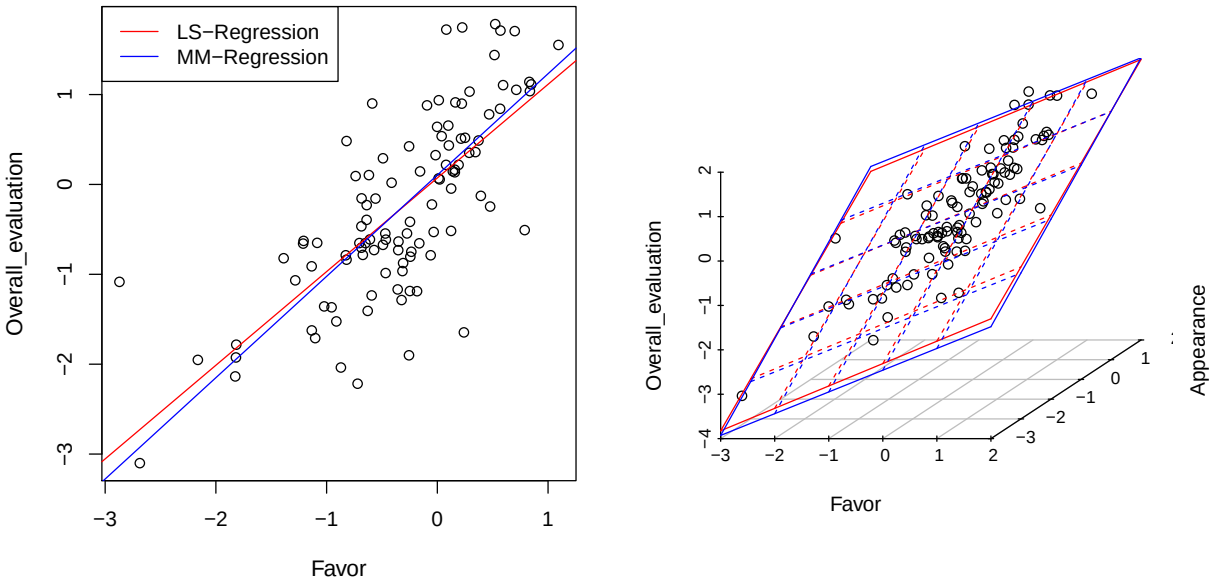


Abbildung 5.3: Lineare Regression (LS und MM) für die Reis-Daten, mit einer (links) bzw. zwei (rechts) erklärenden Variablen.

Schließlich sollten alle 5 erklärenden Variablen für die Prognose verwendet werden. Eine Visualisierung des Problems ist nun nicht mehr möglich. Dafür sehen wir uns aber die sogenannte Inferenzstatistik genauer an:

```
re1 <- lm(Overall_evaluation~.,data=rice)
summary(re1)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.13026    0.03507  -3.715 0.000337 ***
Flavor       0.19359    0.05398   3.586 0.000523 ***
Appearance   0.10829    0.05993   1.807 0.073805 .
Taste        0.53905    0.08163   6.604 2.02e-09 ***
Stickiness   0.40599    0.07146   5.682 1.34e-07 ***
Toughness    0.03513    0.05733   0.613 0.541460
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2745 on 99 degrees of freedom
Multiple R-squared:  0.9306,Adjusted R-squared:  0.927
F-statistic: 265.3 on 5 and 99 DF,  p-value: < 2.2e-16
```

Beim Modell für LS-Regression tragen die Variablen *Flavor, Taste* und *Stickiness* signifikant zur Erklärung bei. Die geschätzten Regressionskoeffizienten sind unter *Estimate* abzulesen. *Appearance* ist an der Grenze zur Signifikanz. Das Modell erklärt die *Output*-Variable sehr gut, zu 93% (*multiple R-squared*).

Für MM-Regression erhält man folgende Resultate:

```
re2 <- lmrob(Overall_evaluation~.,data=rice)
summary(re2)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.09841    0.03729  -2.639  0.00966 **
Flavor       0.21569    0.07926   2.721  0.00769 **
Appearance   0.02917    0.07986   0.365  0.71572
Taste        0.60889    0.09639   6.317 7.64e-09 ***
Stickiness   0.36465    0.07954   4.584 1.33e-05 ***
Toughness    0.01428    0.04812   0.297  0.76726
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Robust residual standard error: 0.2131
Multiple R-squared:  0.954,Adjusted R-squared:  0.9517
Convergence in 18 IRWLS iterations

Robustness weights:
 observation 75 is an outlier with |weight| = 0 ( < 0.00095);
 6 weights are ~= 1. The remaining 98 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0154  0.8960  0.9506  0.8880  0.9795  0.9989
```

Der Koeffizient für *Appearance* ist sehr verschieden vom LS Resultat, und auch weit weg von Signifikanz. Beobachtung 75 ist ein Ausreißer. Abbildung 5.4 zeigt Diagnostik-Plots für MM-Regression, die man mit `plot(re2)` erhält. Die linke Grafik zeigt die geschätzten Werte $\hat{y}$ gegenüber y. Tatsächlich erkennt man ein paar abweichende Beobachtungen, für die das Modell nicht so gut prognostiziert. Im Allgemeinen erhält man aber sehr gute Schätzungen. Die rechte Grafik zeigt robuste Mahalanobis-Distanzen (kommt später im Skriptum) gegen die standardisierten Residuen, also Residuen dividiert durch die Standardabweichung der Residuen, die im Bereich der strichlierten Linien liegen sollten. Tatsächlich findet man hier ein paar Ausreißer in y-Richtung.

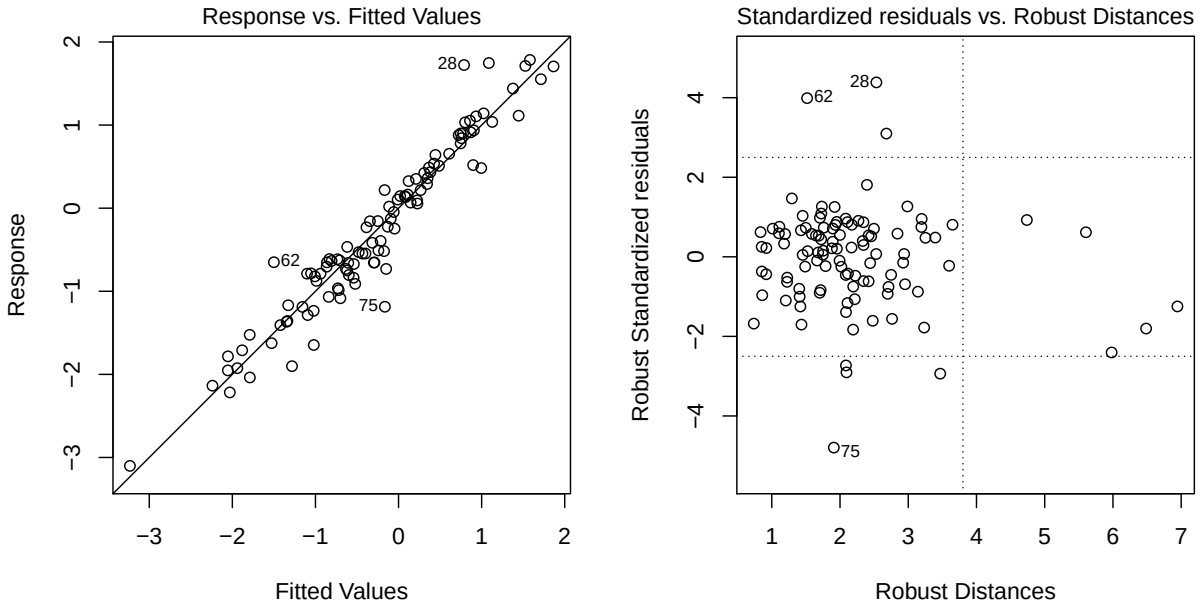


Abbildung 5.4: Diagnostik-Plots für robuste MM-Regression für die Reis-Daten.

Kapitel 6

Glättung und Schätzung nichtlinearer Trends

Wir gehen davon aus, dass Paare von Beobachtungen $(x_1, y_1), \ldots, (x_n, y_n)$ vorliegen. Hier könnten die x-Werte auch Zeitpunkte sein, und die y-Werte die beobachteten Werte einer Variable über der Zeit.

Die folgenden beiden Abschnitte behandeln das Glätten eines 2-dimensionalen Signals, und damit das Schätzen von nichtlinearen zweidimensionalen Trends. Die Abschnitte unterscheiden sich dadurch, dass zunächst von äquidistanten y-Werten ausgegangen wird (z.B. regelmäßige Zeitpunkte), und dann auch nicht äquidistanten y-Werte möglich sind (z.B. Messungen an einem zweiten kontinuierlichen Merkmal).

6.1 Nichtlineare Glätter für äquidistante (Zeit-)Punkte

Wir nehmen an, dass eine Zeitreihe in der Form von Messpunkten x_t, $t = \ldots$, -2, -1, 0, 1, 2, ... mit konstanter Differenz zwischen den Messzeitpunkten vorliegt. Unser Ziel ist es, mit einfachen explorativen Methoden eine Glättung der Zeitreihe zu erreichen, die auch zu einem gewissen Grad robust ist gegenüber Störungen im Signal (peaks, jumps). Mit der Glättung sollen Muster in der Zeitreihe erkannt werden.

Zeitreihe	=	glatte Kurve	+	Residuen
x_t	=	Gx_t	+	r_t
x_t	=	z_t	+	r_t

G ... Glättungsalgorithmus
$r_t := x_t - Gx_t$

Lineare Filter:

$$z_t = \sum_{i=-l_1}^{l_2} \alpha_i x_{t+i} \quad \text{gewichtete Summe der } x_i \text{ in einer Umgebung von } t$$

mit $0 \leq l_1$, $0 \leq l_2$, $\alpha_{-l_1} \neq 0$, $\alpha_{+l_2} \neq 0$.
Anmerkung: Lineare Filter sind sehr empfindlich gegenüber Ausreißern.

Robuste Verfahren:

- Meist mit Hilfe des Medians gebildet.
- Mathematisch eher schwer zu analysieren.
- Anwendungsbereich: EDA, robuste Bereinigung saisonaler Schwankungen, Signal- und Bildverarbeitung.

Spannweite eines Glättungsalgorithmus: (größter Index - kleinster Index + 1) bezüglich der x_t, aus denen z_t berechnet wird.

Medianglättungsalgorithmen mit ungerader Spannweite $2s+1$**:**

$$z_t = Gx_t := \text{median}(x_{t-s}, x_{t-s+1}, \ldots, x_t, x_{t+1}, \ldots, x_{t+s})$$

Z.B. $s = 1$: $z_t = \text{median}(x_{t-1}, x_t, x_{t+1})$
$\Longrightarrow$ erhalte Zeitreihe $\ldots, z_{t-1}, z_t, z_{t+1}, \ldots$

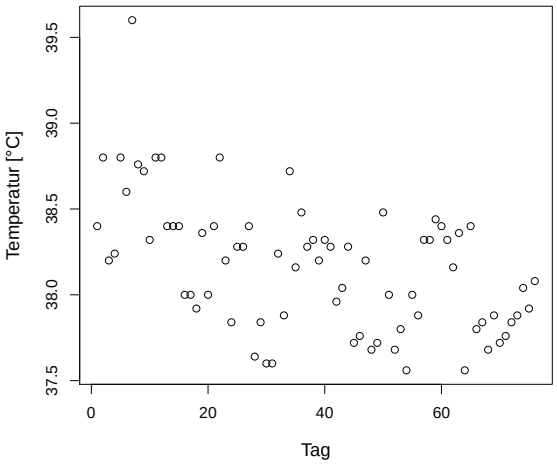
Medianglättungsalgorithmen mit gerader Spannweite $2s$**:**

$$z_{t+\frac{1}{2}} = Gx_t := \text{median}(x_{t-s+1}, \ldots, x_t, x_{t+1}, \ldots, x_{t+s})$$

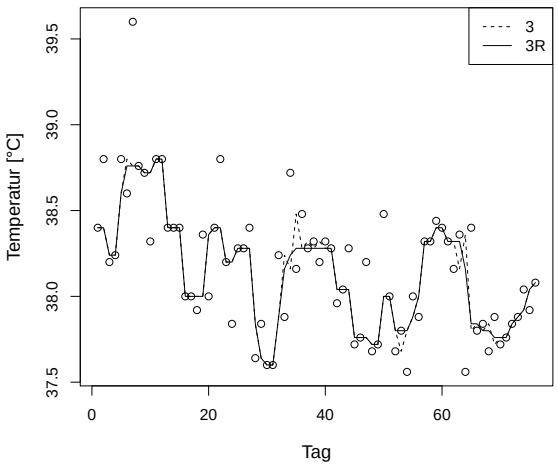
Z.B. $s = 1$: $z_{t+\frac{1}{2}} = \text{median}(x_t, x_{t+1})$
$\Longrightarrow$ erhalte Zeitreihe $\ldots, z_{t-1-\frac{1}{2}}, z_{t-\frac{1}{2}}, z_{t+\frac{1}{2}}, z_{t+1+\frac{1}{2}}, \ldots$

Ein ganzzahliger Index t wird durch einen zweiten geraden Medianglättungsalgorithmus erzielt. Meistens nimmt man $s = 1$. $\Rightarrow z_t = G_2 G_1 x_t$

Abbildung 6.1(a) zeigt eine Zeitreihe von Körpertemperaturen einer Kuh, die täglich an 75 aufeinanderfolgenden Tagen um 6:30 Uhr gemessen wurde. In Abbildung 6.1(b) werden die Resultate verschiedener Glättungsalgorithmen dargestellt.



(a) Originaldaten: Morgendliche Körpertemperaturen gemessen bei einer Kuh an 75 aufeinanderfolgenden Tagen.



(b) "3" Glättung: einmalige Anwendung des Medianalgorithmus, und "3R" Glättung: wiederholte Anwendung des Medianalgorithmus mit der Spannweite 3 (bis Konvergenz).

Abbildung 6.1: Auswirkungen verschiedener Glätter auf die Zeitreihe der Kuh-Daten.

6.2 Robustes Filtern mit Repeated Median

Glätten und Filtern sind äquivalente Aufgaben. Bei Glättung ist man eher am Signal interessiert, bei Filtern ebenso am geglätteten Signal, aber auch an den Residuen, insbesondere an Residuen die Ausreißer sind. Solche "signifikanten" Abweichungen können wichtigen Aufschluss über den dahinterliegenden datengenerierenden Prozess liefern. Voraussetzung für eine zuverlässige Ausreißeridentifikation ist aber, dass die Glättung robust ist, also von den Ausreißern selbst möglichst wenig beeinflusst wird.

Im R Paket `robfilter` sind einige solche Filter-Algorithmen implementiert, die auch Ausreißer-Diagnostik liefern. Es wird dort der Repeated Median (siehe Abschnitt 5.3) verwendet, weil er sehr robust und schnell berechenbar ist, und somit auch für online-Filtern geeignet ist (diese Algorithmen wurden im Zusammenhang für online-Überwachung in der Intensiv-Medizin entwickelt).

Wie im vorigen Abschnitt gehen wir von einer Zeitreihe mit Messwerten x_t aus, die an den (äquidistanten) Zeitpunkten $t = 1, \ldots, T$ beobachtet wurde. Der Filter soll wieder lokal arbeiten, also Information verwenden, die innerhalb eines bestimmten Zeitfensters liegt, wobei das Zeitfenster kontinuierlich weitergeschoben wird. Dazu nehmen wir die Spannweite als $2s+1$ an, mit $s > 0$. Wollen wir also eine Glättung für x_t (filtern bei Zeitpunkt t), so werden dafür die Werte $\{x_{t-s}, x_{t-s+1}, \ldots, x_t, x_{t+1}, \ldots, x_{t+s}\}$ berücksichtigt.

Wir bezeichnen das zugrundeliegende Signal als μ_t, für $t = 1, \ldots, T$, das allerdings nicht beobachtbar ist und geschätzt werden soll. Die Zeitreihe kommt dann zustande durch $x_t = \mu_t + \epsilon_t$, wobei ϵ_t einen Fehler darstellt, der aus einer Mischung von Normalverteilung und einer "heavy-tailed"-Verteilung (erzeugt Ausreißer) charakterisiert ist. Wir nehmen an, dass μ_t lokal (innerhalb eines Zeitfensters mit Spannweite $2s+1$) approximiert werden kann durch eine lineare Funktion:

$$\mu_{t+i} \approx \mu_t + \beta_t \cdot i \quad \text{für } i = -s, -s+1, \ldots, s.$$

Die Parameter dieser linearen Funktion sind der "level" μ_t und die Steigung β_t, die geschätzt werden müssen. Dies kann im Prinzip mit einer beliebigen (robusten) Regressionsmethode durchgeführt werden, wobei die Autoren den Repeated Median propagieren:

$$\hat{\beta}_t = \underset{-s \le i \le s}{median} \left(\underset{\substack{-s \le j \le s \\ j \ne i}}{median} \left(\frac{x_{t+i} - x_{t+j}}{i-j} \right) \right)$$

$$\hat{\mu}_t = \underset{-s \le i \le s}{median} \left(x_{t+i} - \hat{\beta}_t \cdot i \right)$$

Für Ausreißer-Diagnostik wird eine Schätzung der Standardabweichung der Residuen σ_t zu jedem Zeitpunkt t benötigt. Diese kann man mit obigem Prinzip der lokalen linearen Approximation erhalten. Lokal um den Zeitpunkt t erhält man die Residuen

$$r_t(t+i) = x_{t+i} - \hat{\mu}_t - \hat{\beta}_t \cdot i \quad \text{für } i = -s, -s+1, \ldots, s.$$

Man kann nun einen beliebigen (robusten) Streuungsschätzer verwenden um σ_t zu schätzen, wie z.B. den Qn, siehe Abschnitt 3.1:

$$\hat{\sigma}_t = 2.219 \cdot \{|r_t(t+i) - r_t(t+j)|; i < j\}_{(k)} \text{ mit } k = \binom{h}{2} \text{ und } h = \lfloor (2s+1)/2 \rfloor + 1.$$

Mit dieser Streuungsschätzung kann man nun die übliche robuste Ausreißer-Diagnostik machen: ein Ausreißer wird dann diagnostiziert, falls $|\hat{r}_t| > 2 \cdot \hat{\sigma}_t$ ist, wobei $\hat{r}_t = x_t - \hat{\mu}_t$ das geschätzte Residuum zum Zeitpunkt t ist.

Beispiel: Wir betrachten die Kuh-Daten vom letzten Abschnitt. Folgender R Code zeigt eine Anwendung des oben beschriebenen Algorithmus, wobei hier die Spannweite $2s+1 = 7$

gewählt wird (muss eine ungerade Zahl sein). Das Ergebnis wird in Abbildung 6.2 links gezeigt. Die rechte Grafik zeigt das Resultat für Spannweite $2s + 1 = 19$. Man sieht also, dass das Ergebnis stark von diesem Parameter abhängt. Im Paket `robfilter` ist auch ein Algorithmus (`scarm.filter()`) implementiert, der die Spannweite automatisch aufgrund statistischer Tests wählt. Allerdings werden dafür "ausreichend viele" Daten benötigt.

```
library(robfilter)
res <- robust.filter(kuh,width=7) # trend by RM, scale by Qn
plot(res)
res
 # $level
 # [1] 38.4 38.48667 38.57333 38.66 38.47733     ...
 # $slope
 # [1] 0.086667 0.086667 0.086667 0.086667 0.122667     ...
 # $sigma
 # [1] 0.830133 0.830133 0.830133 0.830133 0.833152     ...

# indicate outliers:
ind <- which(res$ol!=0)
points(ind,kuh[ind])
```

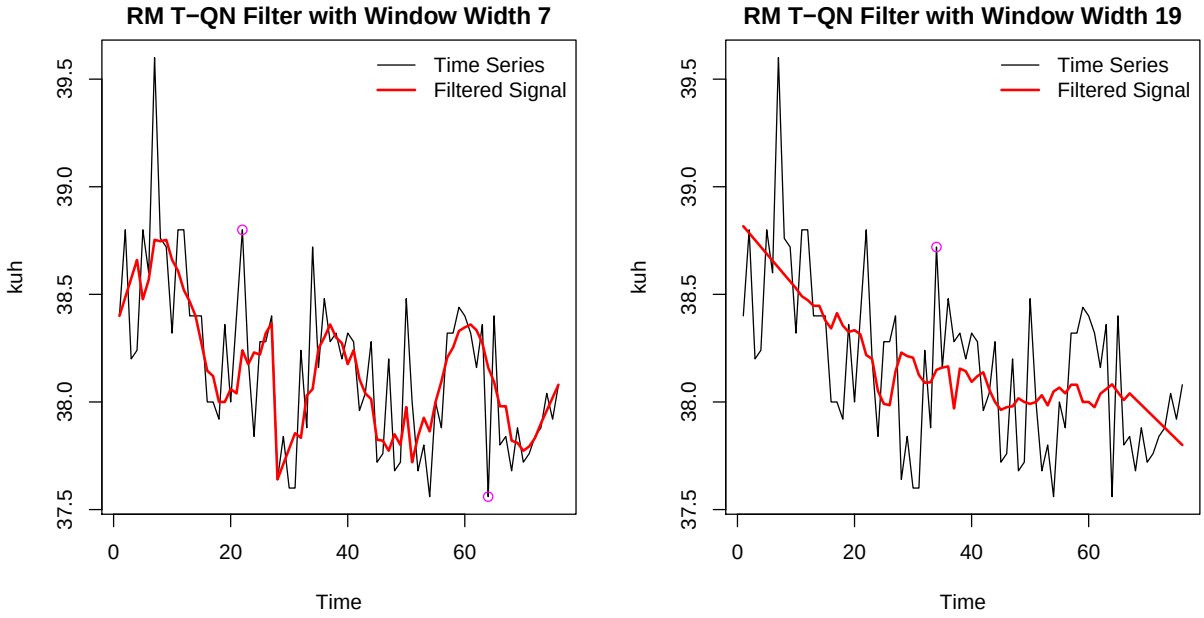


Abbildung 6.2: Anwendung eines robusten Filters auf die Kuh-Daten, mit Fensterbreite 7 (links) und 19 (rechts). Erkannte Ausreißer sind gekennzeichnet.

6.3 LOWESS

LOWESS steht für *LOcally WEighted regression Scatter plot Smoothing* und ist ein auf Scatterplots anwendbares Glättungsverfahren, das Trends in den Daten erkennen lässt. Konzeptionell arbeitet es wie ein nichtlineares Regressionsverfahren: Es werden x-Daten vorgegeben, und die entsprechenden y-Daten geglättet. Man könnte LOWESS auch im Kontext von Zeitreihen anwenden.

Über die n 2-dimensionalen Daten (x_i, y_i), $i = 1, \ldots, n$, wird nichts vorausgesetzt. LOWESS wird durch folgenden Algorithmus beschrieben:

1. f ... Anteil der Punkte, die bei der Glättung verwendet werden sollen.
   $q := \lfloor nf + 0.5 \rfloor$
2. Wähle die q am nächsten zu (x_i, y_i) gelegenen Punkte (x_k, y_k) ((x_i, y_i) ist in diesen q Punkten enthalten), für $i = 1, \ldots, n$.
   $d_{ik} :=\mid x_i - x_k \mid$ Distanz zwischen (x_i, y_i) und (x_k, y_k) (nur die x-Richtung ist relevant!).
   $d_i :=\mid x_i - x_{i_{max}} \mid$ mit i_{max} = Index des am weitesten von (x_i, y_i) entfernten Punktes der ausgewählten q Punkte.
   Trikubische Gewichtsfunktion:
   $$T(t) := \begin{cases} (1 - \mid t \mid^3)^3 & \mid t \mid < 1 \\ 0 & \text{sonst} \end{cases}$$
3. Gewicht von (x_k, y_k) bezüglich (x_i, y_i): $t_i(x_k)$
   $$t_i(x_k) := \begin{cases} T(\frac{\lvert x_i - x_k \rvert}{d_i}) \;=\; T(\frac{d_{ik}}{d_i}) & d_i \neq 0 \\ 1 & d_i = 0 \end{cases}$$
   Anmerkung: $d_i = 0$ bedeutet, dass alle q Punkte die gleiche x-Koordinate haben.
4. Gewichtete Regression für alle i: d.h.
   $$\min \sum_{k=1}^{n} t_i(x_k)(y_k - a - bx_k)^2 \to \hat{a}^{(i)}, \hat{b}^{(i)} \quad \to \quad \hat{y}_i := \hat{a}^{(i)} + \hat{b}^{(i)} x_i \qquad i = 1, \ldots, n$$
   $d_i = 0 \to \hat{b}^{(i)}$ nicht schätzbar $\to \hat{y}_i := \hat{a}^{(i)}$ ($\hat{y}_i$ wird auf eine Konstante gesetzt, z.B. auf den Median der q betrachteten y_k).
5. Berechne Residuen $r_i := y_i - \hat{y}_i$, für $i = 1, \ldots, n$.
6. Berechne neue robuste Gewichte $w(x_k)$ mit der Gewichtsfunktion $B(t)$ (Biweight)
   $$B(t) := \begin{cases} (1 - t^2)^2 & \mid t \mid < 1 \\ 0 & \text{sonst} \end{cases}$$
   $m := \underset{1 \leq k \leq n}{median} \mid r_k \mid$
   (bei Normalverteilung ist m ein Schätzwert für $\approx \frac{2}{3}\sigma \Rightarrow 3m \approx 2\sigma$)

   $$w(x_k) := B(\tfrac{r_k}{3m})$$
   Es werden somit für jeden Punkt Gewichte vergeben, die von der Größe der Residuen abhängen. Punkte mit sehr großen Residuen (außerhalb des 2σ-Bereiches) werden ganz niedergewichtet.
7. Gewichtete Regression wie bei Punkt 4, jedoch mit Gewichten $w(x_k)t_i(x_k)$.
   Schritte 4 - 7 mehrmals wiederholen.
8. Schritte 1 - 7 für jeden Datenpunkt (bzw. für jeden verschiedenen x-Wert) liefert eine Sequenz von Punkten, die linear interpoliert werden.

Anmerkung:

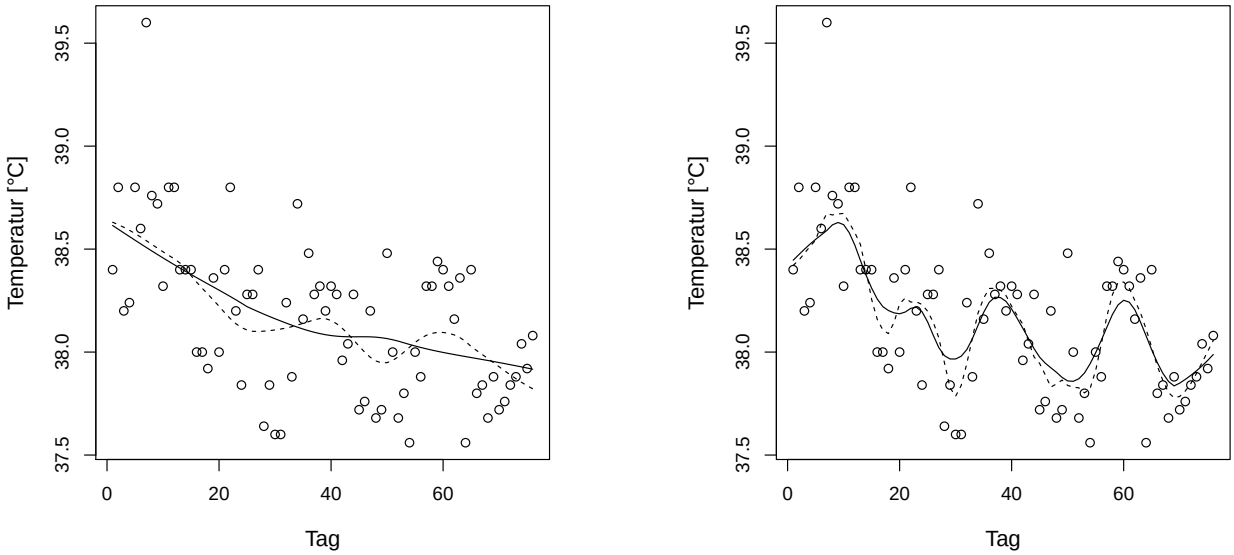
1. f zwischen $\frac{1}{3}$ und $\frac{2}{3}$ liefert gute Ergebnisse.
2. Bei der gewichteten Regression in Punkt 4 gibt es auch die Version, dass nicht lineare Regression, sonder Regression höherer Ordnung verwendet wird. Lineare Regression würde das Modell $y = a + bx$+Fehler betrachten, quadratische Regression betrachtet $y = a + bx + cx^2$+Fehler. Diese Erweiterung ist in der `R`-Funktion `loess` zu finden.

3. Um Veränderungen der Streuung in Abhängigkeit von den x_i zu sehen, kann man das LOWESS Verfahren auf die absoluten Residuen anwenden. Dieses Verfahren wird auch *Spread Smoothing* genannt.

$$(x_i, y_i) \to \begin{cases} \hat{y}_i & \dots \text{ geglättete Werte} \\ r_i & := \; y_i - \hat{y}_i \end{cases}$$

LOWESS Smoothing des Streuungsdiagramms $(x_i, \mid r_i \mid)$.

In Abbildung 6.3 wird der LOWESS Algorithmus auf die Kuh-Daten von Abbildung 6.1(a) angewandt. Die Auswirkungen verschiedener Wahlen des Parameters f (Anteil der bei der Glättung verwendeten Punkte) werden dargestellt.



(a) Glättung mit $f = \frac{2}{3}$ (durchgezogene Linie) und $f = \frac{1}{3}$ (strichlierte Linie).

(b) Glättung mit $f = \frac{1}{5}$ (durchgezogene Linie) und $f = \frac{1}{8}$ (strichlierte Linie).

Abbildung 6.3: LOWESS Smoothing: Auswirkungen der Wahl des Parameters f auf die geglättete Kurve.

Abbildung 6.4 zeigt die LOWESS Glättung angewandt auf die Hamster-Daten aus Tabelle 4.2. In der rechten Grafik wird Spread Smoothing durchgeführt.

6.3.1 Upper and Lower Smoothing

Zusätzlich zu den geglätteten Werten bietet Upper and Lower Smoothing auch eine Streuungsinformation. Man geht folgendermaßen vor:

1. LOWESS Glättung des Streuungsdiagrammes (x_i, y_i). Die geglätteten Werte sind $\hat{y}_i$.
2. Aufteilung der Residuen $r_i := y_i - \hat{y}_i$ in positive und negative Residuen:

r_i^+	positive Residuen	r_i^-	negative Residuen
x_i^+	Abszissen zu den r_i^+	x_i^-	Abszissen zu den r_i^-
$\hat{y}_i^+$	geglättete Werte zu den r_i^+	$\hat{y}_i^-$	geglättete Werte zu den r_i^-

3. Glättung von (x_i^+, r_i^+) und (x_i^-, r_i^-)

$$(x_i^+, r_i^+) \to \hat{r}_i^+; \; (x_i^-, r_i^-) \to \hat{r}_i^-$$

Zeichnen der Kurven $(x_i, \hat{y}_i)$, $(x_i^+, \hat{y}_i^+ + \hat{r}_i^+)$ und $(x_i^-, \hat{y}_i^- + \hat{r}_i^-)$.

Abbildung 6.5 zeigt ein Beispiel deses Verfahrens mit den Hamsterdaten aus Tabelle 4.2.

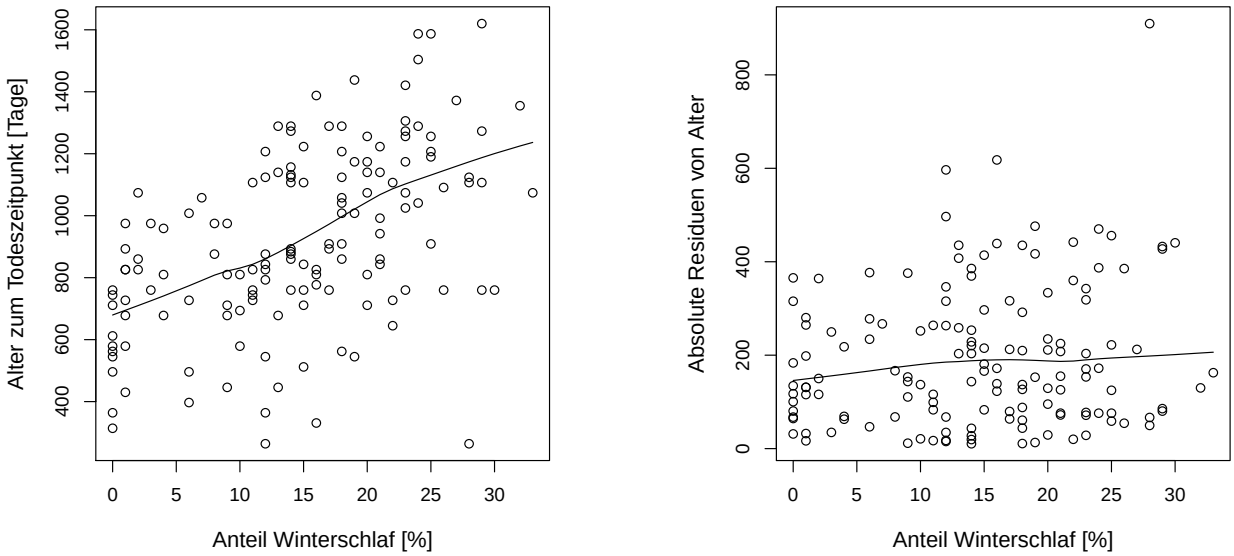


Abbildung 6.4: LOWESS Smoothing der Hamsterdaten aus Tab. 4.2 (links) und Spread Smoothing der gleichen Daten (rechts), das eine leichte Zunahme der Streuung der Residuen mit steigenden x-Werten anzeigt

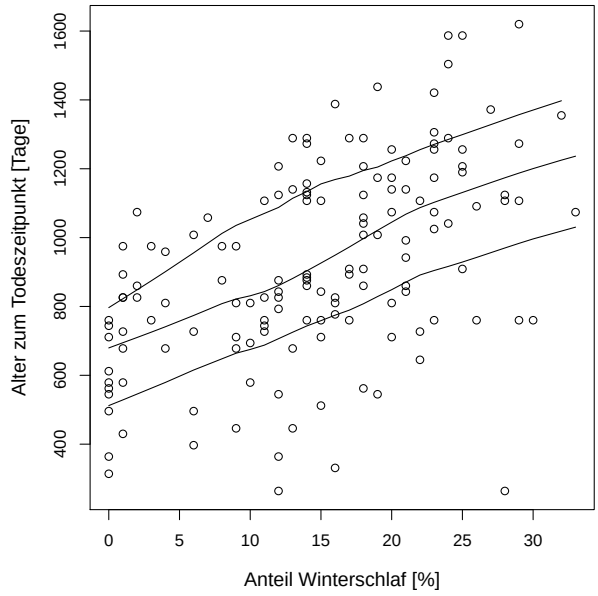


Abbildung 6.5: Upper and Lower Smoothing der Hamsterdaten.

6.3.2 Pairs of Middle Smoothing

Nachdem es nicht egal ist, ob LOWESS auf Daten (x_i, y_i) oder auf (y_i, x_i), für $i = 1, \ldots, n$ angewendet wird, könnten auch beide Varianten versucht werden. In Abbildung 6.6 wird dies für 4-dimensionale Daten gemacht, wobei immer Paare gegenübergestellt werden. Diese Darstellung wird im folgenden Kapitel als Scatterplot-Matrix oder *Draftman's Display* eingeführt. Für jedes Paar erhält man somit die Kurven $(x_i, \hat{y}_i)$ und $(\hat{x}_i, y_i)$. Die Daten sind Messungen der Luftqualität in New York, die sind als `environmental` im R-Paket `library(lattice)` zu finden.

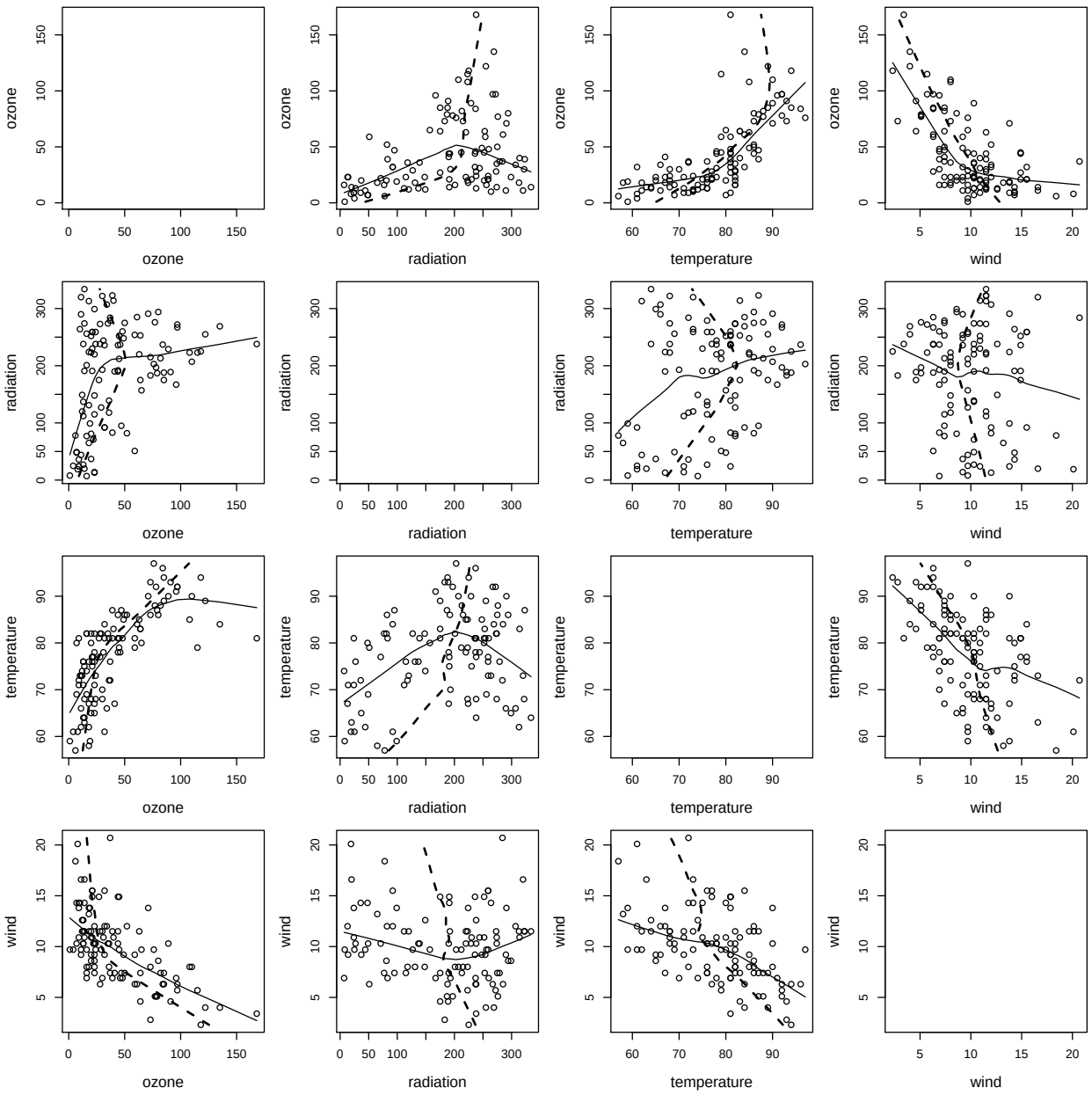


Abbildung 6.6: Pairs of Middle Smoothing der Daten `environmental` aus der `library(lattice)`. Zur besseren Unterscheidbarkeit wurden die Kurven $(x_i, \hat{y}_i)$ mit durchgezogenen Linien, und die Kurven $(\hat{x}_i, y_i)$ mit strichlierten Linien gezeichnet.

Kapitel 7

Zeitreihenanalyse – eine Einführung

Ziel dieses Kapitels ist es, einen Überblick über die Vorgehensweisen mit Zeitreihen zu erhalten, und das Wissen zu vermitteln, wie Basis-Analysen praktisch durchgeführt werden. Der Name *Zeitreihenanalyse* sagt auch schon, worauf man hinaus will, nämlich Zeitreihen nicht nur visuell darstellen sondern auch analysieren. Wichtige Fragestellungen in diesem Kontext sind:

- Gibt es Trends oder saisonale Schwankungen? Wenn ja, kann man diese quantifizieren?
- Welche Struktur haben die Residuen, die nach Abzug des Trends und der saisonalen Schwankung verbleiben?
- Folgt die Zeitreihe einem bestimmten Muster, das man modellieren kann?
- Kann man in die Zukunft prognostizieren?
- Gibt es Strukturbrüche in der Zeitreihe oder Ausreißer?

In einige dieser Fragestellungen wird hier ein Einblick vermittelt.

Wir beschäftigen uns hier nur mit *univariaten Zeitreihen*, also mit Werten x_t, für $t = 1, \ldots, T$.

Als Beispiel so einer Zeitreihe betrachten wir die monatlich produzierte Biermenge von Australien, im Zeitraum Jänner 1956 bis August 1995. Die Daten sind erhältlich auf der Seite `http://134.76.173.220/beer.zip`. Man kann sie in `R` einlesen, als Zeitreihenobjekt umwandeln, und grafisch darstellen mit:

```
beer <- read.csv2("beer.csv")
beer <- ts(beer[,1],start=1956,freq=12)
plot(beer)
```

Abbildung 7.1 (oben) zeigt die resultierende Zeitreihe. Man erkennt einen klaren Trend, aber auch klare saisonale Schwankungen. Abgesehen davon gibt es aber auch immer wieder größere lokale Abweichungen (Ausreißer). Wenn man die Zeitreihe in Abbildung 7.1 (oben) genau betrachtet, dann bemerkt man, dass die Abweichungen nach oben größer sind als Abweichungen nach unten. Dies könnte Nachteile bringen bei der späteren Schätzung. Als einfachen Ausweg kann man mit den *logarithmierten Daten* arbeiten, die in Abbildung 7.1 (unten) dargestellt sind. Die lokalen Abweichungen der logarithmierten Daten scheinen jetzt etwas symmetrischer zu sein. Differenzen von den logarithmierten benachbarten Werten sind:

$$\ln(x_t) - \ln(x_{t-1}) = \ln\left(\frac{x_t}{x_{t-1}}\right) \approx \frac{x_t - x_{t-1}}{x_{t-1}}$$

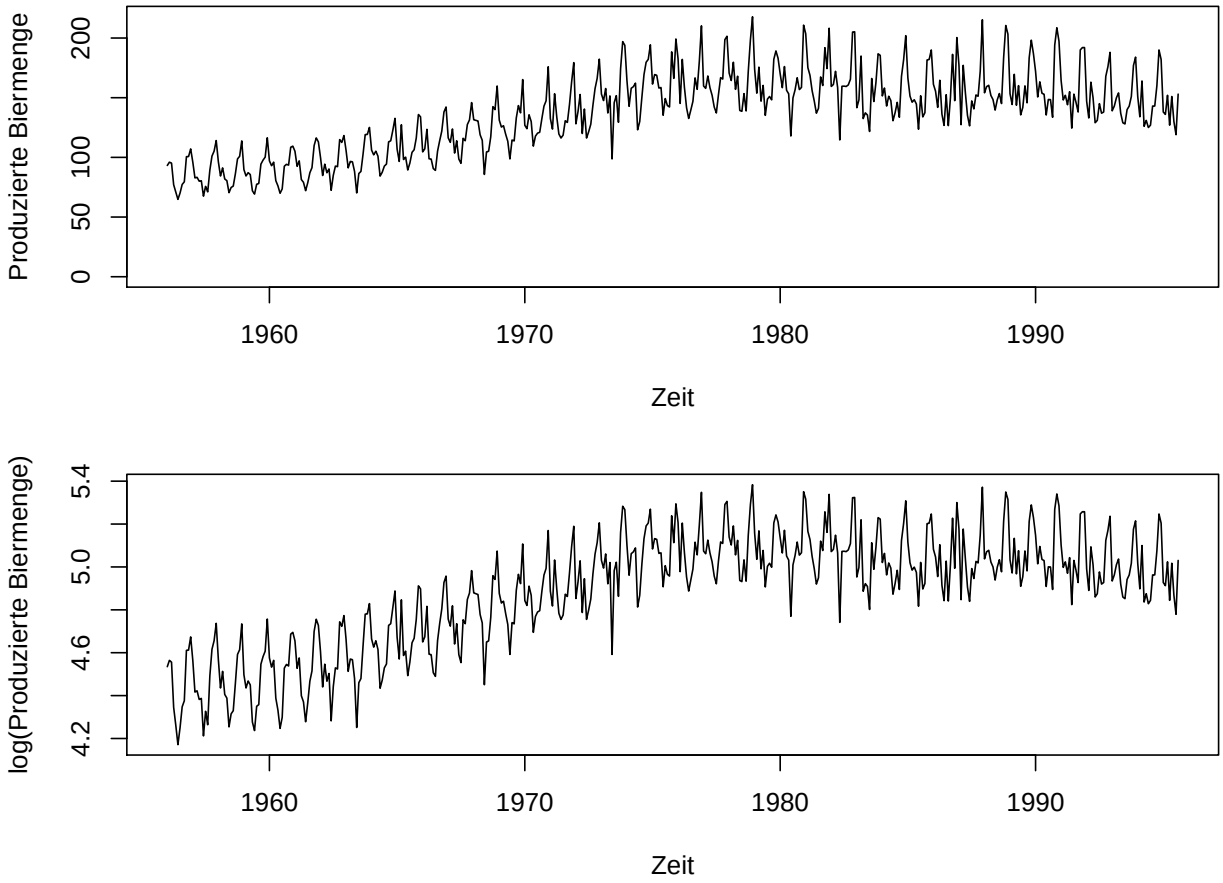


Abbildung 7.1: Originale (oben) und logarithmierte (unten) Zeitreihe der Bier-Daten.

Letztere Beziehung gilt, weil die log-Differenzen von Nachbarn i.A. kleine Zahlen sind. Mit den logarithmierten Werten ist man also nur an *relativen Unterschieden* interessiert, und nicht an den absoluten. Eine log-Darstellung wird auch oft bei Finanzzeitreihen gemacht; das Fachvokabel in diesem Zusammenhang heißt *log-returns*. Wir werden nun in weiterer Folge mit diesen log-transformierten Werten arbeiten.

7.1 Zerlegung der Zeitreihe in Komponenten

Unter dem *Trend* einer Zeitreihe versteht man, dass die Zeitreihe über einen längeren Zeitraum tendenziell steigt oder fällt. Saisonalität bedeutet, dass bestimmte Schwankungen vorliegen, wie z.B. jahreszeitliche Schwankungen. Oft tritt beides gemeinsam auf, man möchte aber dennoch jede Komponente einzeln schätzen. Wir möchten also die Zeitreihe zerlegen in folgende Komponenten:

$$x_t = \tau_t + \delta_t + e_t \quad \text{für } t = 1, \ldots, T,$$

mit der *Trendkomponente* τ_t, der *Saisonkomponente* δ_t, und der *Restkomponente* e_t. Wir wissen zusätzlich, dass die Zeitreihe eine Periodizität von P hat. Bei der Bier-Zeitreihe ist $P = 12$, weil monatliche Werte vorliegen. Somit ergeben sich $C = \lfloor T/P \rfloor$ Zyklen.

In Kapitel 6 wurden bereits Methoden behandelt, mit denen ein Signal geglättet werden kann, bzw. mit denen nichtlineare Trends geschätzt wurden. Die `R` Funktion `stl` verwendet solche Methoden, insbesondere die Funktion `loess`, siehe Abschnitt 6.3. `stl` geht nach folgendem

iterativen Schema vor:
In der k-ten Iteration, $k = 1, 2, \ldots$, seien die Schätzungen der Komponenten $\tau_t^{(k)}$ und $\delta_t^{(k)}$. Die Iteration $(k+1)$ hat die Gestalt:

(1) Trendbereinigung (*detrending*): $x_t - \tau_t^{(k)}$

(2) `loess` angewandt auf (1) für die Zeitpunkte $t_i = t + i \cdot P$, mit $i = 0, \ldots, C-1$, in Folge für alle $t \in \{1, \ldots, P-1\}$. Z.B. werden alle Jänner-Werte geglättet, dann alle Februar-Werte, usw.

(3) Anwendung eines (linearen) Filters auf die Werte von (2), siehe Kapitel 6, ergibt $\delta_t^{(k+1)}$.

(4) Saisonbereinigung: $x_t - \delta_t^{(k+1)}$

(5) `loess` angewandt auf (4) ergibt $\tau_t^{(k+1)}$

Ähnlich wie beim LOWESS Algorithmus, siehe Abschnitt 6.3, werden von der Restkomponente $e_t^{(k+1)} = x_t - \tau_t^{(k+1)} - \delta_t^{(k+1)}$ (Residuen) Gewichte berechnet, die Ausreißer in den Iterationen der obigen Prozedur niedergewichten sollen.

Abbildung 7.2 zeigt das Resultat von `stl` angewandt auf die logarithmierte Bier-Zeitreihe, erzeugt mit:

```
plot(stl(log(beer),s.window="periodic"))
```

Falls die Restkomponente (*remainder*) keine Struktur mehr aufweist, würde man das als "weißes Rauschen" bezeichnen. Wenn das nicht der Fall ist, enthalten sie noch immer wichtige Information, die modelliert werden soll. In den Ergebnissen in Abbildung 7.2 enthalten die Residuen zwar keine auffällige Struktur, aber unterschiedlich hohe "peaks", die auch von Interesse sein können.

7.2 Regressionsmodelle für Zeitreihen

Ein wichtiges Ziel bei der Modellierung von Zeitreihen ist die Progose für zukünftige Werte. Man könnte nun mit Regression versuchen, in die Zukunft zu prognostizieren. Wir werden hier ein paar einfache Modelle für diesen Zweck verwenden.

7.2.1 Lineares Modell

In Kapitel 5 wurden Methoden zur (robusten) Schätzung linearer Trends vorgestellt. Für logarithmierte Zeitreihen hat das lineare Modell die Form

$$\ln(x_t) = \beta_0 + \beta_1 t + e_t$$

mit den Koeffizienten β_0 und β_1, und dem Fehlerterm e_t. Die Koeffizienten könnten mit den Methoden aus Kapitel 5 geschätzt werden. In Folge verwenden wir LTS-Regression.

Für die Bier-Daten würden wir folgendermaßen vorgehen:

```
t <- (1:length(beer))/12+1956   # Zeitachse entsprechend erzeugen
logbeer <- log(beer)            # modelliere logarithmierte Werte
library(rrcov)                  # fuer LTS-Regression
plot(logbeer)
res <- ltsReg(logbeer~t)        # LTS-Regression mit linearem Modell
lines(t,res$fit)
```

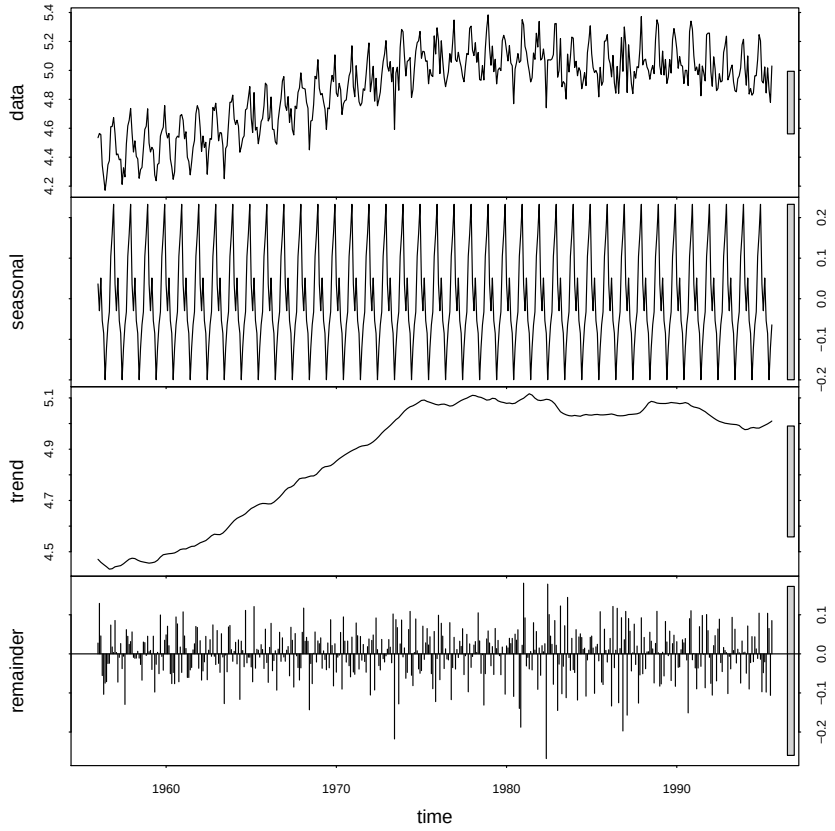


Abbildung 7.2: Zerlegung der Zeitreihe in die einzelnen Komponenten. (`stl`)

Abbildung 7.3 (links) zeigt das Resultat. Das lineare Modell ist natürlich zu einfach, und wir sollten zu einem komplexeren Modell übergehen.

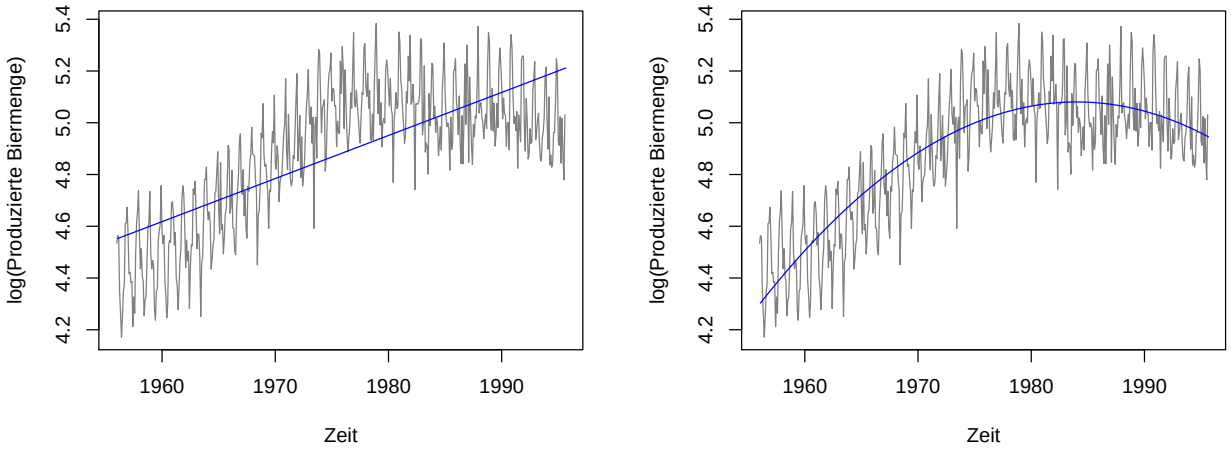


Abbildung 7.3: Lineares Modell (links) und quadratisches Modell (rechts) für die Bier-Zeitreihe.

7.2.2 Regression mit quadratischem Term

Obiges lineares Modell kann sehr einfach erweitert werden mit einem quadratischen Term:

$$\ln(x_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + e_t$$

Für unser Beispiel sieht das so aus:

```
t2 <- t^2                          # quadratischer Term
plot(logbeer)
res <- ltsReg(logbeer~t+t2)        # LTS-Regression fuer neues Modell
lines(t,res$fit)
```

Das Ergebnis zeigt Abbildung 7.3 (rechts). Man erhält somit eine (glatte) Schätzung des Trends. Zukünftige Werte könnten nun sehr einfach prognostiziert werden. Wenn $\hat{\beta}_0$, $\hat{\beta}_1$, und $\hat{\beta}_2$ die geschätzten Regressionsparameter sind, so ist

$$\hat{x}_t = \exp\left(\hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 t^2\right)$$

die Schätzung für zukünftige Zeitpunkte $t = T+1, \ldots, N$. Nachdem das Modell aber relativ einfach ist, wird die Prognose nur bedingt sinnvoll, und bestenfalls nur für die nahe Zukunft anwendbar sein.

7.2.3 Regression mit Fourier Koeffizienten

Man könnte ein Zeitsignal $f(t)$ mit Periodenlänge P auch mit (endlichen) Fourierreihen darstellen:

$$f(t) = a_0 + \sum_{j=1}^{J}\Big(a_j \cos(\omega_j t) + b_j \sin(\omega_j t)\Big), \qquad \text{mit } \omega_j = \frac{2\pi j}{P}.$$

Die Summanden in der Reihe entsprechen den Frequenzen mit höher werdender Schwingung, beginnend bei der Grundschwingung mit Periodenlänge P. Würden wir den ersten Summanden in unser Modell zusätzlich dazugeben, so könnte etwa die Saisonalität mitmodelliert werden. Wir haben somit das Modell:

$$\ln(x_t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 \cos\left(\frac{2\pi}{P} \cdot t\right) + \beta_4 \sin\left(\frac{2\pi}{P} \cdot t\right) + e_t.$$

Für unser Beispiel sieht das so aus (Achtung, wir haben `t` bereits früher durch $P = 12$ dividiert!):

```
cos.t <- cos(2*pi*t)
sin.t <- sin(2*pi*t)
plot(logbeer)
res <- ltsReg(lbeer~t+t2+cos.t+sin.t) # LTS-Regression fuer neues Modell
lines(t,res$fit)
```

Abbildung 7.4 zeigt das Resultat, das natürlich noch immer nicht "perfekt" prognostizieren wird. Man könnte noch Terme höherer Ordnung einschliessen, um die Anpassung zu verbessern.

`R` liefert mit `summary(res)` auch eine *Inferenztabelle* für das Modell:

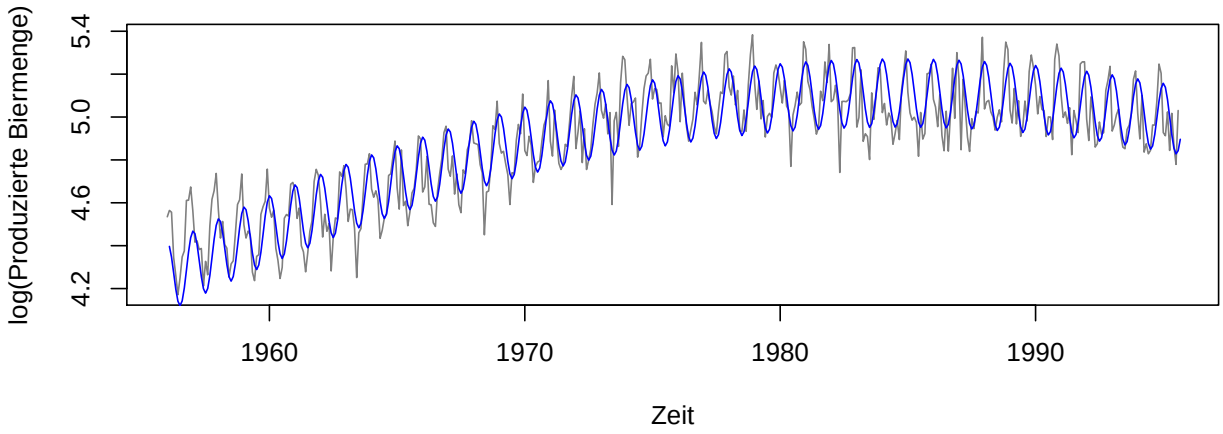


Abbildung 7.4: Modell mit Fourerdarstellung für die Bier-Daten.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
Intercept -4.161e+03  1.411e+02 -29.480   <2e-16 ***
t          4.198e+00  1.429e-01  29.387   <2e-16 ***
t2        -1.058e-03  3.615e-05 -29.260   <2e-16 ***
cos.t      1.582e-01  6.014e-03  26.308   <2e-16 ***
sin.t      7.136e-03  5.966e-03   1.196    0.232
```

Demnach sind alle Koeffizienten außer β_4 signifikant verschieden von 0. Allerdings würde man für eine Fourier Darstellung trotzdem diesen Term inkludieren, weil Kosinus und Sinus nur als Paar auftreten sollten.

7.3 Exponentielles Glätten (exponential smoothing)

Die Prognose von zukünftigen Werten könnte so gemacht werden, dass man die beobachteten Werte gewichtet nach ihrer Aktualität berücksichtigt. Wenn die Zeitreihe x_t für $t = 1, \ldots, T$ beobachtet wird, werden die geglätteten Werte $\tilde{x}_t$ erhalten durch

$$\tilde{x}_t = \alpha x_t + (1 - \alpha)\tilde{x}_{t-1},$$

mit dem Glättungsfaktor $0 < \alpha < 1$. Je kleiner α genommen wird, desto weniger werden die aktuellsten Werte berücksichtigt, und daraus folgt, dass die Sequenz der $\tilde{x}_t$ glatter wird. Den Startwert $\tilde{x}_m$ kann man wählen als arithmetisches Mittel der ersten m Werte von x_t. α kann praktisch so bestimmt werden, dass für die vorhandenen Daten die Summe der quadrierten Residuen (oder ein robusteres Kriterium) minimiert wird:

$$\sum_{t=m}^{T} (x_t - \tilde{x}_t)^2 \longrightarrow \min$$

Es ist leicht zu sehen, dass diese Art der Gewichtung einer rekursiven Berechnung entspricht:

$$\tilde{x}_t = \alpha x_t + \alpha(1 - \alpha)x_{t-1} + \alpha(1 - \alpha)^2 x_{t-2} + \alpha(1 - \alpha)^3 x_{t-3} + \ldots$$

Die Gewichte für vergangene Werte fallen also *exponentiell* ab, was auch den Namen des Verfahrens rechtfertigt.

Möchte man zum Zeitpunkt t nun für h Zeitschritte vorausschätzen, bezeichnet mit $\hat{x}_{t+h|t}$, dann kann man dafür den aktuellen geglätteten Wert verwenden, also

$$\hat{x}_{t+h|t} = \tilde{x}_t.$$

Die Prognose aller künftigen Werte hängt hier also nicht vom Horizont h ab, bis zu dem man vorhersagen möchte. Das ist sinnvoll für *stationäre Zeitreihen*, die keinen Trend und keine saisonalen Abhängigkeiten aufweisen, und die immer relativ rasch zu ihrem Mittelwert zurückkehren, nicht aber für Zeitreihen mit Trend. Bei der **Glättung nach Holt-Winters** wird eine Trendvariable b_t mitberücksichtigt:

$$\tilde{x}_t = \alpha x_t + (1-\alpha)(\tilde{x}_{t-1} + b_{t-1})$$
$$b_t = \beta(\tilde{x}_t - \tilde{x}_{t-1}) + (1-\beta)b_{t-1}$$

An jedem Zeitpunkt t erfasst also b_t den lokalen Anstieg der Zeitreihe. α und β liegen wieder in $(0, 1)$ und regulieren den Grad der Glättung. Die Prognose mit Holt-Winters um h Zeitschritte nach vorne ist dann

$$\hat{x}_{t+h|t} = \tilde{x}_t + hb_t.$$

Die Prognose wird somit nach einer Geradengleichung ermittelt, mit dem aktuellsten Anstieg b_t, ausgehend von $\tilde{x}_t$.

Auf ähnliche Weise kann auch noch eine saisonale Komponente modelliert werden.

Für die Bier-Zeitreihe wird die Holt-Winters Methode in `R` folgendermaßen verwendet:

```
plot(beer)
beer.hw <- HoltWinters(beer)        # Holt-Winters
lines(beer.hw$fitted[,1])           # geglaettete Linie
```

Es wird sowohl eine Trend als auch eine saisonale Komponente berücksichtigt. Das Ergebnis der Glättung ist in Abbildung 7.5 (oben). Die geschätzten Parameter sind:

```
Smoothing parameters:
 alpha:  0.07532444
 beta :  0.07434971              # Parameter fuer den Trend
 gamma:  0.143887                # Parameter fuer die Saison
```

Mit diesem Modell kann man nun in die Zukunft prognostizieren, hier für die nächsten 48 Monate:

```
plot(beer,xlim=c(1956,1999))
lines(predict(beer.hw,n.ahead=48))
```

Abbildung 7.5 (unten) zeigt das Resultat.

7.4 Modellierung von Zeitreihen

7.4.1 Kenngrößen

Zeitreihenmodelle berücksichtigen Abhängigkeiten, die bei einer systematischen Verschiebung der Zeitreihe um k Schritte gegeben sind. Wir betrachten also die Werte x_t und die Werte x_{t-k}, wobei t variiert. Man spricht in diesem Zusammenhang auch von einem *lag* k. Abhängigkeiten werden mit der **Autokovarianz** analysiert. Die Autokovarianz der *Ordnung*

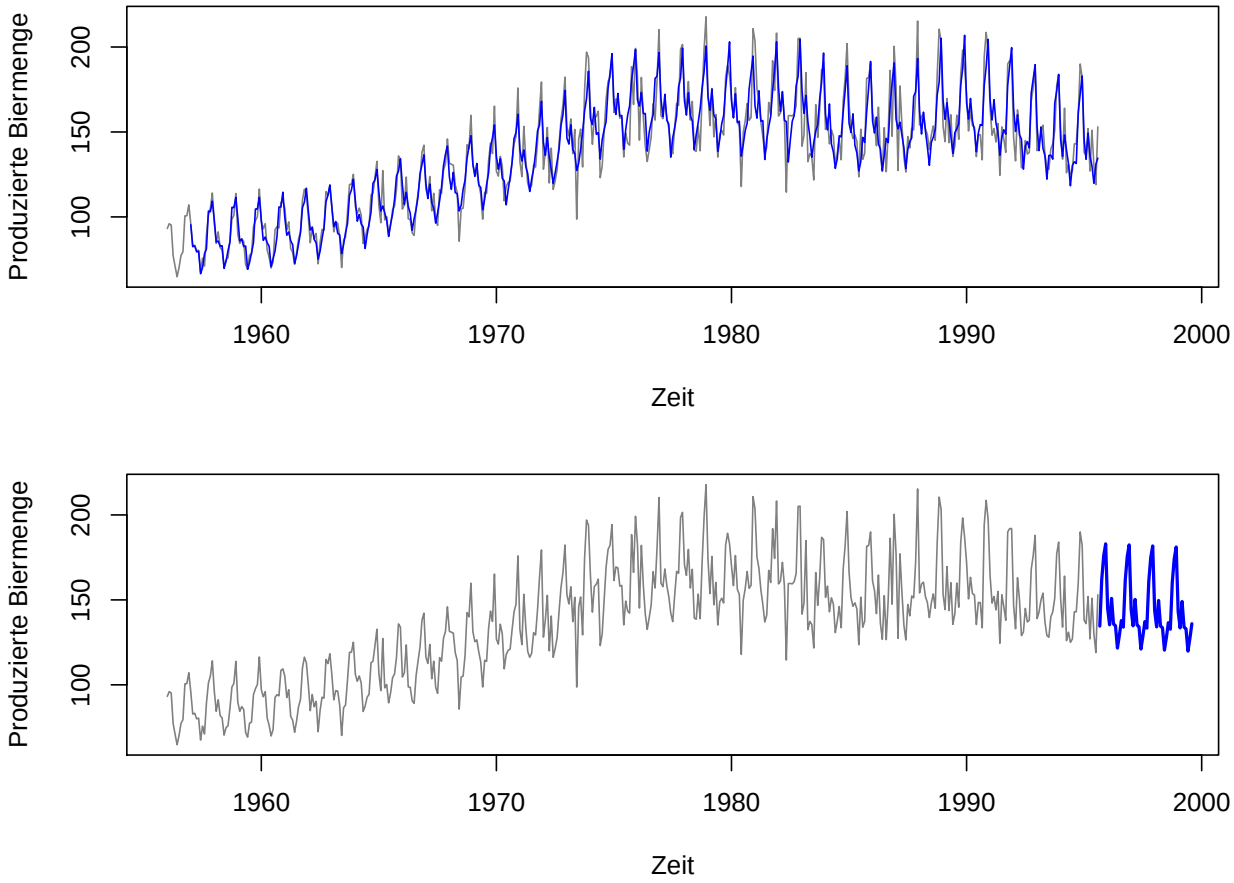


Abbildung 7.5: Exponentielles Glätten nach Holt-Winters (oben) und Prognose (unten).

k ist definiert als $\text{Cov}(x_t, x_{t-k})$, und sie kann geschätzt werden mit

$$c_k = \frac{1}{T} \sum_{t=k+1}^{T} (x_t - \bar{x})(x_{t-k} - \bar{x}),$$

mit dem arithmetischen Mittel $\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t$. Die Autokovarianz für *lag* 0, also c_0, ist die Varianz von x_t. Somit kann man die **Autokorrelation der Ordnung** k definieren als

$$\rho_k = \text{Corr}(x_t, x_{t-k}) = \text{Cov}(x_t, x_{t-k})/\text{Var}(x_t),$$

und schätzen mit $r_k = c_k/c_0$.

Oben haben wir von *stationären Zeitreihen* gesprochen. Diese sind dadurch charakterisiert, dass sie gleichen Erwartungswert (Mittel) und gleiche Varianz für alle t haben, und dass ihre Autokovarianz gleich für alle t und jedes $k > 0$ ist. Stationäre Zeitreihen mit Autokorrelation Null für $k > 0$ werden als *white noise* (weißes Rauschen) bezeichnet.

Man kann auf die Eigenschaft *white noise* testen mit der Q-Statistik, auch **Ljung-Box Statistik** genannt. Dabei wird die Hypothese $H_0 : \rho_1 = \rho_2 = \ldots = \rho_{kmax} = 0$ getestet, mit einem festgelegten Wert $kmax$.

In Abbildung 7.6 ist der Wasserstand (Jahresmittel) vom Huron-See im Zeitraum 1875-1972 (in *feet*) gegeben. In Abbildung 7.7 (links) sind die Autokorrelationen für meherer Werte von k gezeigt. Diese Darstellung nennt man auch **Korrelogramm**. Die strichlierten horizontalen Linien entsprechen den Grenzen für Signifikanz, und sie werden durch unkorrelierte

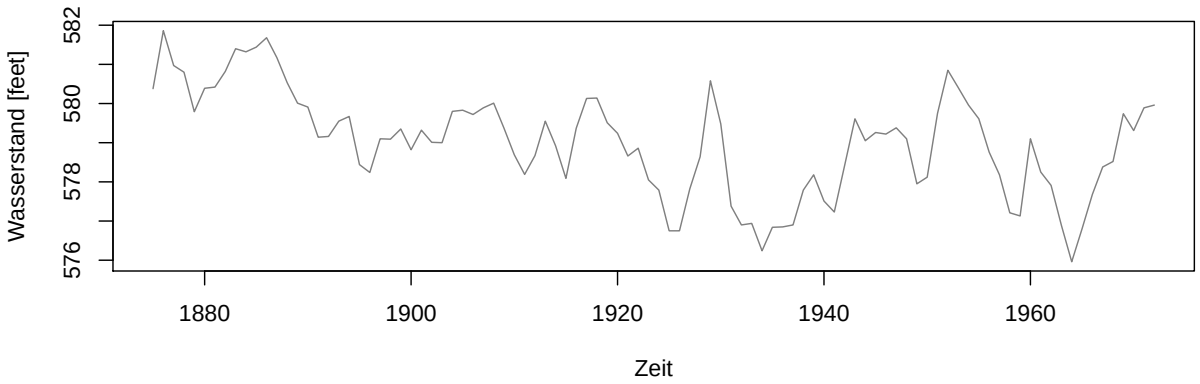


Abbildung 7.6: Jährlicher Wasserstand vom Huron-See.

Zeitreihen ermittelt. Bei unkorrelierten sowie bei stationären Zeitreihen müssten alle Autokorrelationen für $k > 0$ innerhalb dieser Grenzen sein. Bei diesem Beispiel erkennt man ausgeprägte Korrelationen, die erst nach einem großen *lag* abklingen.

Korrelationen zwischen benachbarten Zeitpunkten können transferiert werden, also von x_t auf x_{t-1}, von x_{t-1} auf x_{t-2}, usw. Die resultierende Korrelation zwischen x_t und x_{t-k} ist daher beeinflusst durch die dazwischenliegenden Beobachtungen. Um diesen Effekt herauszurechnen wird daher eine weitere Gößе zur Beurteilung der Abhängigkeit definiert, nämlich die **Partielle Autokorrelation** der Ordnung k:

$$\text{Corr}(x_t, x_{t-k}|x_{t-1}, \ldots, x_{t-k+1}) \quad \text{für} \quad k = 0, 1, 2, \ldots$$

Das entspricht der Autokorrelation zwischen den Residuen einer Regression von x_t auf $x_{t-1}, \ldots, x_{t-k+1}$ und den Residuen einer Regression von x_{t-k} auf die gleichen Variablen $x_{t-1}, \ldots, x_{t-k+1}$. Abbildung 7.7 (rechts) zeigt die partiellen Autokorrelationen für die Daten vom Huron-See. Bis *lag* 2 erhält man also noch ausgeprägte partielle Korrelationen. Diese Darstellung nennt man *partielles Korrelogramm*.

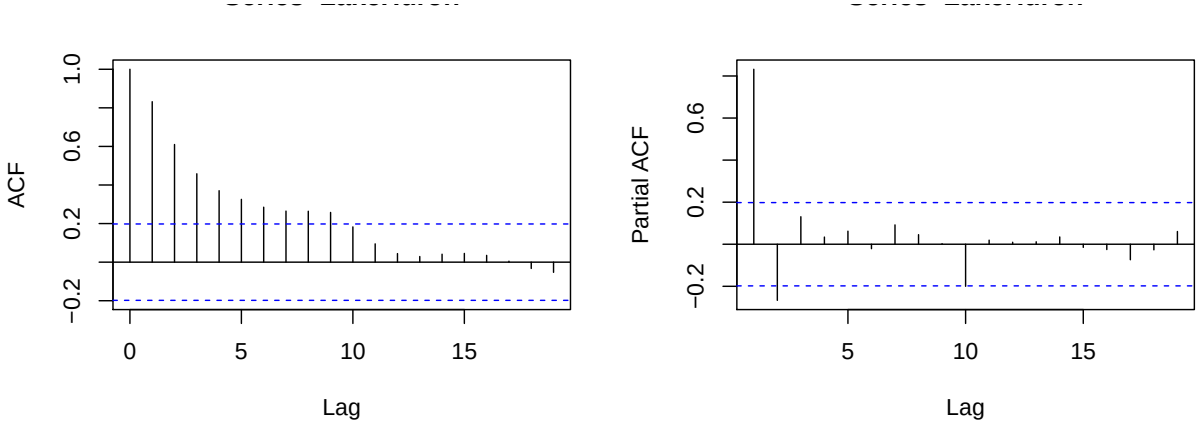


Abbildung 7.7: Korrelogramm (links) und partielles Korrelogramm (rechts) der Wasserstandsdaten vom Huron-See.

7.4.2 Grundlegende Zeitreihenmodelle

Moving Average (MA) Modell

Eine *stationäre* Zeitreihe folgt einem *Moving Average* Prozess der Ordnung 1, kurz MA(1), wenn

$$x_t = a + u_t - \theta u_{t-1},$$

mit den unbekannten Parametern a und θ. Hier steht u_t für *white noise*. Analog kann man *Moving Average* der Ordnung q, also MA(q), definieren als

$$x_t = a + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - \ldots - \theta_q u_{t-q},$$

mit den unbekannten Parametern a und $\theta_1, \ldots, \theta_q$.

Die Autokorrelationen von MA(q) sind 0 für *lags* größer als q. Wenn im Korrelogramm die Korrelationen stark abfallen, und nach *lag* q nicht mehr signifikant sind, so ist das ein Indiz für einen MA(q) Prozess. Ein Beispiel von MA(2) ist in Abbildung 7.8 (oben) zu sehen.

Autoregressives (AR) Modell

Eine *stationäre* Zeitreihe folgt einem *autoregressiven* Prozess der Ordnung 1, kurz AR(1), wenn

$$x_t = a + \phi x_{t-1} + u_t,$$

mit den unbekannten Parametern a und ϕ. Analog kann man einen *autoregressiven* Prozess der Ordnung p, also AR(p), definieren als

$$x_t = a + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + u_t,$$

mit den unbekannten Parametern a und $\phi_1, \ldots, \phi_p$.

Die partiellen Autokorrelationen von AR(p) sind 0 für *lags* größer als p. Die Autokorrelationen gehen langsamer gegen 0, und manchmal weisen sie eine sinus-förmige Struktur auf. Ein Beispiel von AR(2) ist in Abbildung 7.8 (unten) zu sehen.

ARMA (Autoregressive-Moving-Average) Modell

Wenn weder im Korrelogramm noch im partiellen Korrelogramm eine "Implosion" auf 0 ab einem bestimmten *lag* sichtbar wird, dann kann eine Mischung aus AR(p) und MA(q) sinnvoll sein. Diese wird bezeichnet als ARMA(p, q) und ist definiert als

$$x_t = a + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - \ldots - \theta_q u_{t-q}.$$

Einfachen Modellen sollte der Vorzug gegeben werden, also p und q sollten klein gewählt werden.

ARIMA Modell

Das ARMA Modell setzt eine stationäre Zeitreihe voraus. Liegt ein Trend vor, so spricht man von einer nicht-stationären oder *integrierten*, also einer trendbehafteten Zeitreihe (saisonal

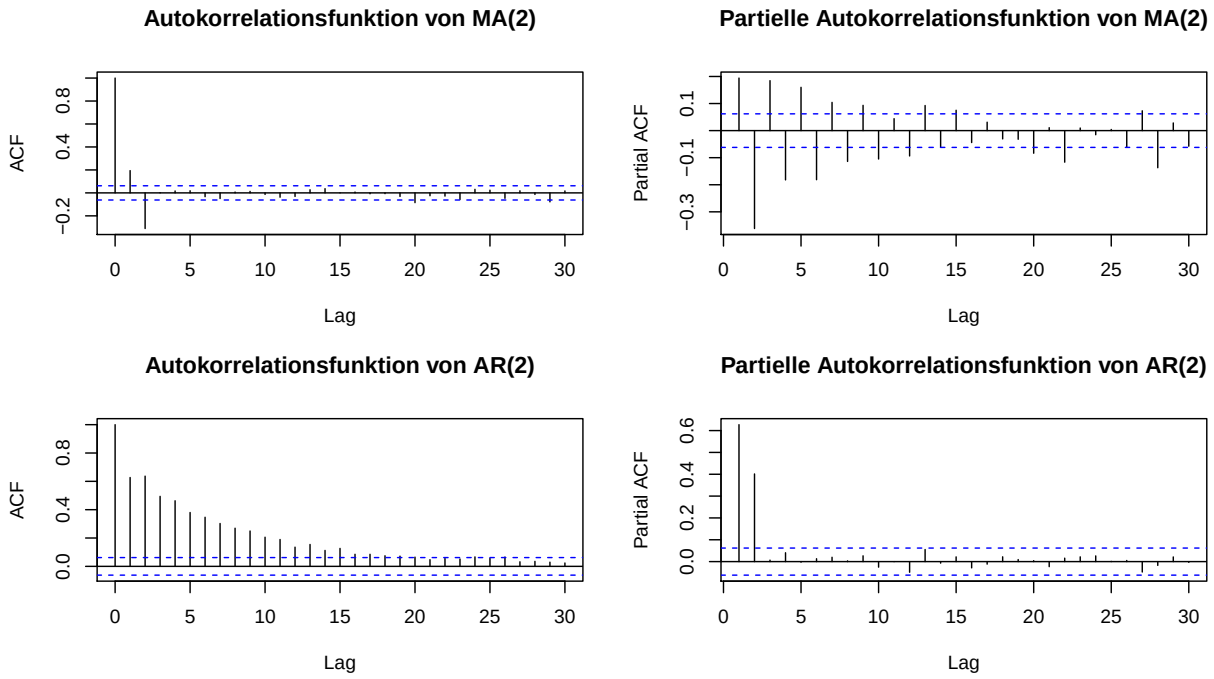


Abbildung 7.8: Typische Struktur von MA(2) bzw. AR(2) Prozessen.

inkludiert). Trends können durch Bildung von *Differenzen* eliminiert werden. Dazu definieren wir den "Differenzen-Operator" Δ als

$$\Delta x_t = x_t - x_{t-1}.$$

Lineare Trends können durch einmalige Anwendung von Δ eliminiert werden, quadratische Trends durch zweimalige Anwendung von Δ, also

$$\Delta^2 x_t := \Delta(\Delta x_t) = \Delta(x_t - x_{t-1}) = x_t - 2x_{t-1} + x_{t-2}.$$

Definition: Eine Zeitreihe x_t, $t = 1, \ldots, T$, folgt einem ARIMA(p, d, q) Modell, wenn $\Delta^d x_t$ einem ARMA(p, q) Modell folgt.

Wegen der bevorzugten Einfachheit von Modellen wählt man d meist als 0 oder 1.

Beispiel: Das Modell ARIMA(1,1,0) hat die Gestalt

$$\Delta x_t = a + \phi \Delta x_{t-1} + u_t,$$

also

$$x_t = x_{t-1} + a + \phi(x_{t-1} - x_{t-2}) + u_t.$$

7.4.3 Schätzung der Parameter

Die Parameterschätzung für die oben beschriebenen Modelle erfolgt nach dem Kleinste-Quadrate Prinzip, also durch Minimierung von $\sum_{t=1}^{T}(x_t - \hat{x}_t)^2$. Wir wollen hier allerdings nicht auf die technischen Details dieser Schätzung eingehen. In `R` können die Parameter geschätzt werden mit

```
arima(data,order=c(p,d,q))
```

für die Daten `data`, mit den entsprechenden Ordnungen des ARIMA(p, d, q) Modells.

Für die Daten vom Huron-See versuchen wir in Anlehnung an Abbildung 7.7 folgendes Modell:

```
data(LakeHuron)
fit <- arima(LakeHuron,order=c(1,0,1))
```

Das ist also ein ARMA(1,1) Modell von der Gestalt

$$x_t = a + \phi x_{t-1} + u_t - \theta u_{t-1}.$$

Die Schätzungen der Parameter ϕ, θ und a sind:

```
> fit$coef
        ar1         ma1   intercept
  0.7448993   0.3205891 579.0554556
```

7.4.4 Diagnostik von Zeitreihenmodellen

Ein ganz wichtiger Schritt bei der Modellsuche ist die Diagnostik. Das ARMA(1,1) Modell für die Daten vom Huron-See gibt zwar Schätzungen der Koeffizienten zurück, aber wir wissen nicht, ob das Modell überhaupt passend ist. Eine Diagnostik kann gemacht werden mit:

```
tsdiag(fit)
```

Das Resultat ist in Abbildung 7.9 gezeigt. Man sieht oben die (standardisierten) Residuen, die bei einem korrekt spezifizierten Modell *white noise* sein sollten, in der Mitte das Korrelogramm der Residuen, wo keine Struktur mehr ersichtlich sein sollte, und unten die Ljung-Box Statistik bis *lag* 10, wo alle p-Werte jenseits der Schranke 0.05 sein sollten, also nicht signifikant. Unser Modell scheint also gut spezifiziert zu sein.

7.4.5 Prognose

Nachdem wir ein sinnvoll spezifiziertes Modell haben, können wir zum interessantesten Punkt, nämlich zur Prognose gehen. Ein sinnvoll spezifiziertes Modell muss aber nicht notwendigerweise ein gutes Prognosemodell sein. Das Resultat `fit` gibt auch Information über die Modellgüte zurück, wie z.B. die Varianz der Schätzungen. Diese wird dazu verwendet, um zusätzlich zur Prognose auch ein Konfidenzintervall zu erhalten.

```
plot(LakeHuron,xlim=c(1875,1980))
LH.pred <- predict(fit,n.ahead=8)  # Prognose fuer die nachsten 8 Jahre
lines(LH.pred$pred,col="blue")
lines(LH.pred$pred+2*LH.pred$se,col="blue",lty=2)
lines(LH.pred$pred-2*LH.pred$se,col="blue",lty=2)
```

Abbildung 7.10 zeigt die Progose für die folgenden 8 Jahre, gemeinsam mit einem 95%-Konfidenzintervall. An der Größe dieses Intervalls kann man gut erkennen, dass unser Modell – obwohl inhaltlich sinnvoll – als Prognosemodell nicht sehr wertvoll ist.

Man kann auch im Internet auf

```
http://www.glerl.noaa.gov/data/now/wlevels/lowlevels/
       plot/data/Michigan-Huron-1860-.csv
```

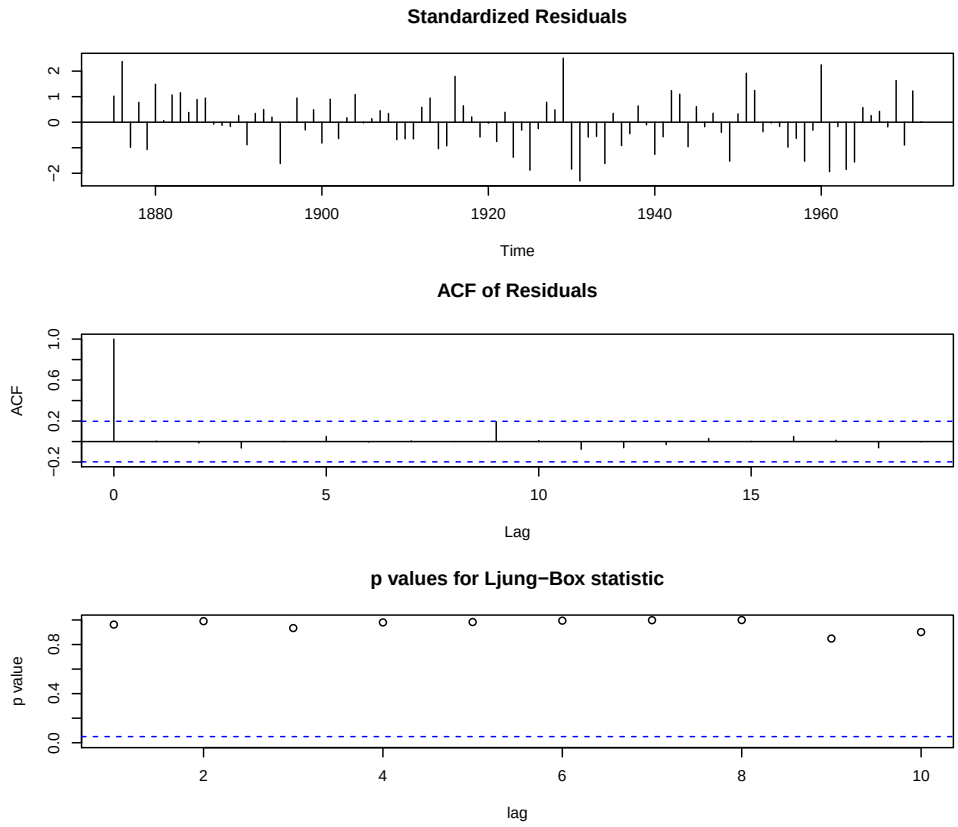


Abbildung 7.9: Diagnostik für das ARMA(1,1) Modell der Daten von Huron-See.

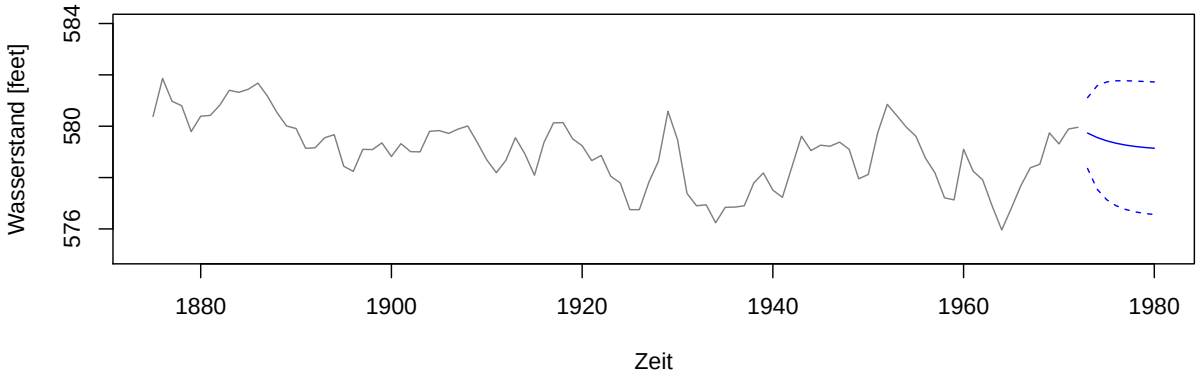


Abbildung 7.10: Prognose des Wasserstandes vom Huron-See für die nächsten 8 Jahre, samt Konfidenzintervall.

historische und aktuelle Daten vom Wasserstand des Lake Huron finden. Abbildung 7.11 zeigt diese Daten als rote Linie, zusätzlich zur Information von Abbildung 7.10. Die Daten aus `R` stimmen nicht exakt mit den Daten vom Internet überein, was daran liegen kann, dass an einer anderen Stelle des Sees gemessen wurde. Hier wurde nun die Prognose bis zum Jahr 2011 fortgesetzt, nur um den Vergleich zu den aktuellen Daten zu haben. Das Konfidenzintervall umspannt praktisch den ganzen Datenbereich, und die Prognose pendelt sich rasch auf den Mittelwert ein. Die Prognose ist, wenn überhaupt, nur sehr kurzfristig brauchbar. Unser Modell ist generell entweder zu einfach, oder es kann eben kein besseres Modell gefunden werden.

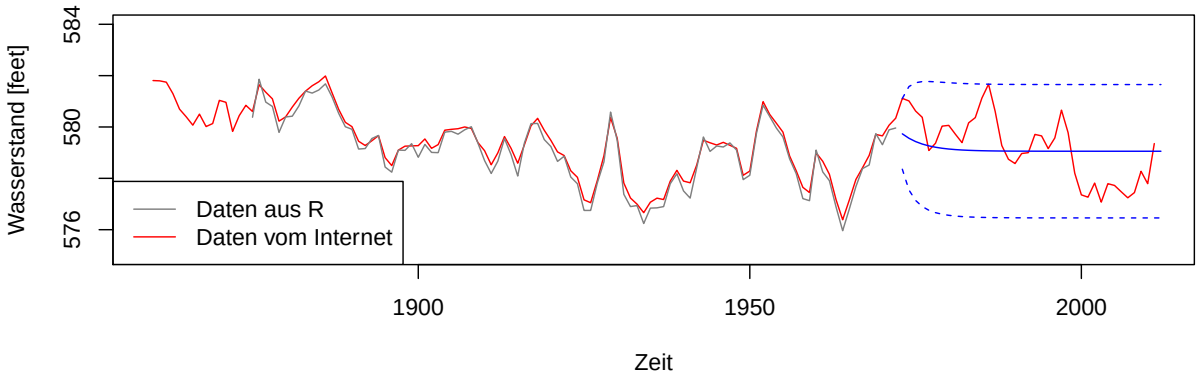


Abbildung 7.11: Vergleich mit aktuellen Daten vom Huron-See aus dem Internet.

Kapitel 8

Multivariate Grafiken

Multivariate Daten werden meist in einer rechteckigen Tabelle (Matrix), bestehend aus n Zeilen und p Spalten dargestellt, wobei jede Zelle einen numerischen Wert enthält. Jede Zeile enthält Information eines *Objektes*, und jede Spalte Information einer *Variablen*. Die Matrix wird nachfolgend mit $\boldsymbol{X}$ bezeichnet. Die Zeilen oder Stichproben werden notiert mit $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Die i-te Stichprobe ist somit $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ (Spaltenvektor!).

8.1 Streudiagramme

In Streudiagrammen niedrigdimensionaler Daten sind die Daten noch durch **Variation des Symbols** (Markers), durch das die Datenwerte repräsentiert werden, darzustellen. Als Beispiel dafür sind in Abbildung 8.1 die 4-dimensionalen Iris-Daten angeführt. Die ersten beiden Dimensionen *Sepal Length* und *Sepal Width* bilden horizontale und vertikale Achse im Plot. Die Variable *Petal Length* ist durch die Größe der Symbole repräsentiert. Die Variable *Petal Width* ist im Plot durch verschiedene Graustufen dargestellt.

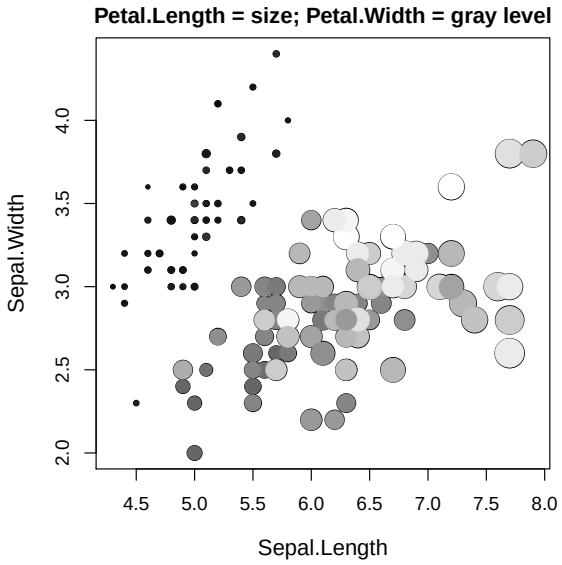


Abbildung 8.1: Iris-Daten: Mittels Symbolgröße und Graustufen werden 4 Dimensionen dargestellt.

Eine andere Möglichkeit ist die Darstellung aller $\binom{p}{2}$ 2-dimensionaler Ansichten der p Variablen (**Draftsman's Display**), wie dies mit den 4-dimensionalen Iris-Daten in Abbildung 8.2 gemacht wurde. Es wurde hier zusätzlich die Information der Gruppenzugehörigkeiten der 3 Typen von Schwertlilien durch verschiedene Wahl von Graustufen visualisiert.

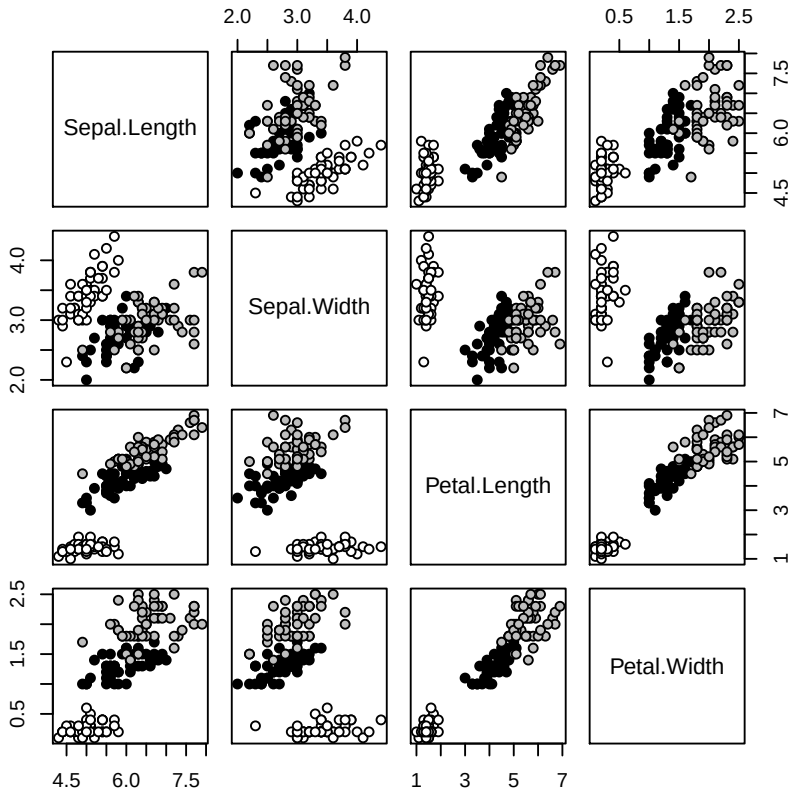


Abbildung 8.2: Iris-Daten: Draftman's Display mit zusätzlicher Gruppeninformation.

Diese Darstellung aller Paare von 2-dimensionalen Ansichten ist sogar für Daten höherer Dimension brauchbar, sofern die Daten markante Strukturen aufweisen. Man muss sich aber dessen immer bewusst sein, dass man "nur" 2-dimensionale Projektionen der p-dimensionalen Daten sieht.

Natürlich könnte man mehrdimensionale Daten auch in 3-dimensionalen Streudiagrammen darstellen (für die restlichen Dimensionen müssten verschiedene Symbole, Farben, etc. gewählt werden). Dies erscheint jedoch nur sinnvoll mit einer entsprechenden interaktiven Visualisierung.

Eine andere Variation sind die *Casement Displays* – eine schichtenweise Darstellung der Daten, sowie die *Multiwindow Plots*, bei denen eine oder zwei der p Variablen für eine Zerlegung der Stichprobe herangezogen und die resultierenden Teilmengen in Streuungsdiagrammen dargestellt werden.

8.2 Profile, Sterne, Segmente, Chernoff Faces

8.2.1 Profile

Jeder Datenwert $\boldsymbol{x_i}$ wird durch p Balken oder Striche dargestellt. Die Länge des j-ten Balkens (Striches) des k-ten Punktes ist proportional zu x_{kj}.

Als Beispiel betrachten wir die Daten in Tabelle 8.1 mit Todesraten in Virginia. Die "Variablen" sind verschiedene Bevölkerungsgruppen (Spalten), und die "Objekte" verschiedene Altersgruppen (Zeilen). Abbildung 8.3 zeigt Profile für die einzelnen Objekte (links), aber

Tabelle 8.1: Todesraten (in %) in Virginia im Jahr 1940. Die Daten sind in Altersgruppen (Zeilen) und Bevölkerungsgruppen (Spalten) untergliedert.

Age group	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

auch die äquivalente Darstellung der Profile bezogen auf die einzelnen Variablen (rechts).

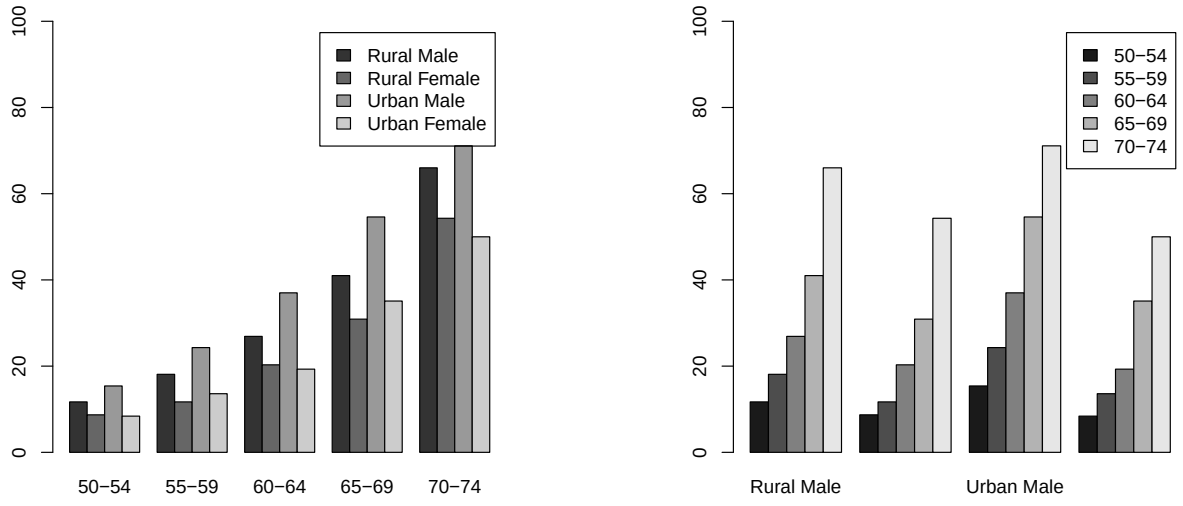


Abbildung 8.3: Darstellung der Daten aus Tabelle 8.1 mit Profilen für Objekte (links) und Variablen (rechts).

Die Darstellung mit Profilen eignet sich für Daten mit relativ hoher Dimension (etwa 20), allerdings gibt es zweifellos eine Einschränkung hinsichtlich der Anzahl der Objekte. Die Profile können zwar ohne Koordinatensystem, sowie nebeneinander und untereinander dargestellt werden, aber schon für eine dreistellige Anzahl von Objekten wird die Darstellung nicht mehr übersichtlich.

Wenn Variablen mit sehr unterschiedlicher Skalierung betrachtet werden, sollten die Daten vorher normiert werden, weil sonst die Übersichtlichkeit verloren gehen kann. Als Normierung sind viele Möglichkeiten denkbar, wie z.B. Normierung jeder Variable auf

- das Intervall $[0, 1]$ (oder ein anderes Intervall),
- (robuste) Varianz 1,
- Norm 1, d.h. $\sum_{i=1}^{n} x_{ij}^2 = 1$ für alle j.

Alternativ (oder zusätzlich) können die Variablen auch transformiert werden (z.B. mit Logarithmus).

8.2.2 Sterne

Je nach Variationsart werden sie auch "Webs", "Polygons" und "Circular Plots" genannt. Die Datenpunkte werden "normiert", z.B.

$$\tilde{\boldsymbol{x}}_i := \left(\frac{x_{i1}-\bar{x}_{.1}}{s_1}, \frac{x_{i2}-\bar{x}_{.2}}{s_2}, \ldots, \frac{x_{ip}-\bar{x}_{.p}}{s_p}\right)^T$$

mit

$$\begin{array}{rcl} \bar{x}_{.j} & := & \text{Mittelwert der } n \text{ Werte } x_{kj} \text{ für Variable } j = 1, \ldots, p \\ s_j & := & \text{Streuung der } n \text{ Werte } x_{kj} \text{ für Variable } j = 1, \ldots, p \end{array}$$

und danach die $\tilde{x}_{ki}$ radial in die p Richtungen $\frac{2\pi}{p}k$ aufgetragen ($k = 0, \ldots, p-1$).

Als Beispiel betrachten wir die Fahrzeug-Daten aus Tabelle 8.2. Diese verschiedenen Characteristika von 32 Fahrzeugen sind aus amerikanischen Motor-Journalen des Jahres 1974 entnommen. (Die gesamten Daten sind in `R` unter *data(mtcars)* verfügbar.)

In Abbildung 8.4 werden diese 7-dimensionalen Daten mit Sternen dargestellt, wobei jedes der 32 Fahrzeuge einen Stern repräsentiert. Aufgrund der Ähnlichkeit von manchen Sternen kann man die Fahrzeuge visuell in Gruppen (Cluster) einteilen.

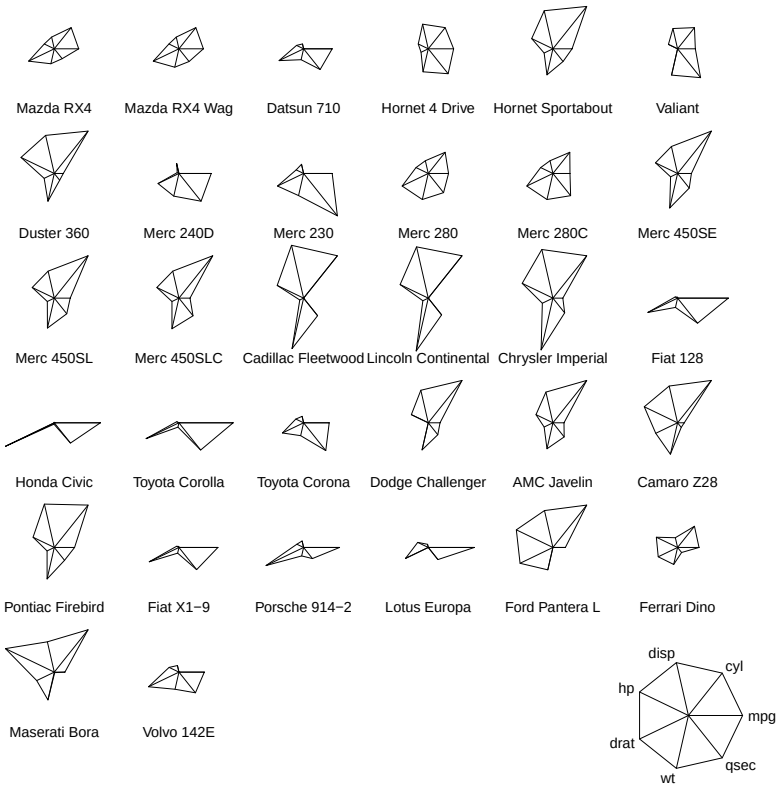


Abbildung 8.4: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Sternen. (`stars`)

Tabelle 8.2: Fahrzeug-Daten: Von 32 Fahrzeugen wurden verschiedene Charakteristika erhoben. Die Daten stammen aus amerikanischen Motor-Journalen des Jahres 1974.

Car type	Miles per gallon	No. of cylinders	Displacement	Horsepower	Rear axle ratio	Weight (lb/1000)	Time for 1/4 mile
Abbreviation	*mpg*	*cyl*	*disp*	*hp*	*drat*	*wt*	*qsec*
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02
Valiant	18.1	6	225.0	105	2.76	3.460	20.22
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60

8.2.3 Segmente

Die Darstellung mit Segmenten ist sehr ähnlich zu jener mit Sternen. Die Daten werden zuerst entsprechend normiert. Dann wird der gesamte Winkel von 2π regelmäßig in p Teile untergliedert. Nun wird der Wert jeder Variable eines Objektes in ein Segment eingetragen, wobei die Fläche des Kreissegmentes dem Wert auf der Variable entspricht.

In Abbildung 8.5 werden obige Fahrzeug-Daten mit Segmenten dargestellt.

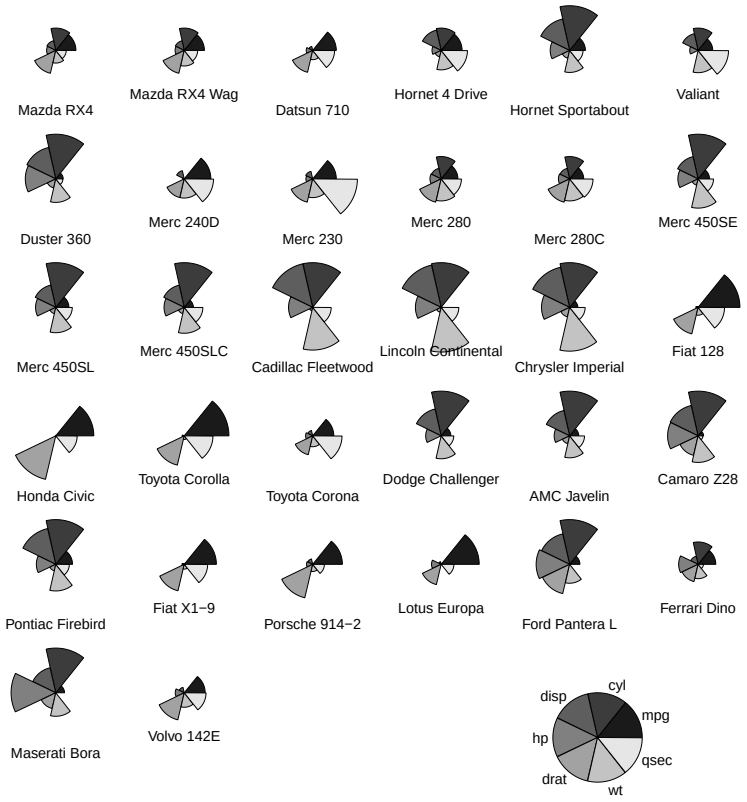


Abbildung 8.5: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Segmenten. (`stars`)

8.2.4 Chernoff Faces

Jede Variable wird durch die Größe, Richtung oder Krümmung eines Gesichtsteiles dargestellt. Hier sind kaum mehr Rückschlüsse von der Darstellung eines Datenpunktes auf die Werte seiner Komponenten (d.h. auf die x_{ij}) möglich. Außerdem ist bei dieser Darstellung immer eine gewisse Subjektivität hinterlegt, weil der Analyst auf manche Gesichtsteile mehr Augenmerk legt und manche weniger wichtig erscheinen. Abbildung 8.6 zeigt diese Darstellung für die Fahrzeug-Daten.

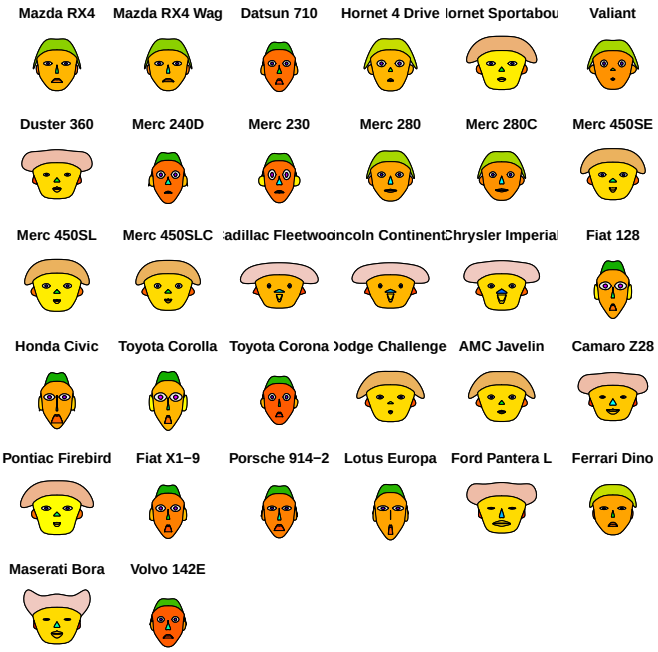


Abbildung 8.6: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Chernoff faces. (`faces` aus `library(aplpack)`)

8.2.5 Quader (Boxes)

Die Variablen werden mittels Clusteranalyse in 3 Gruppen unterteilt. Ausschlaggebend dafür ist die Ähnlichkeit zwischen den Variablen, die mittels der Korrelationsmatrix ermittelt werden kann. Dann wird jede Variablengruppe als eigene Seite in einem Quader repräsentiert. Die relativen Anteile jeder Beobachtung an den Variablen bestimmt schließlich die Größe der Quader. Abbildung 8.7 zeigt für die obigen Fahrzeug-Daten die Darstellung mit Quadern.

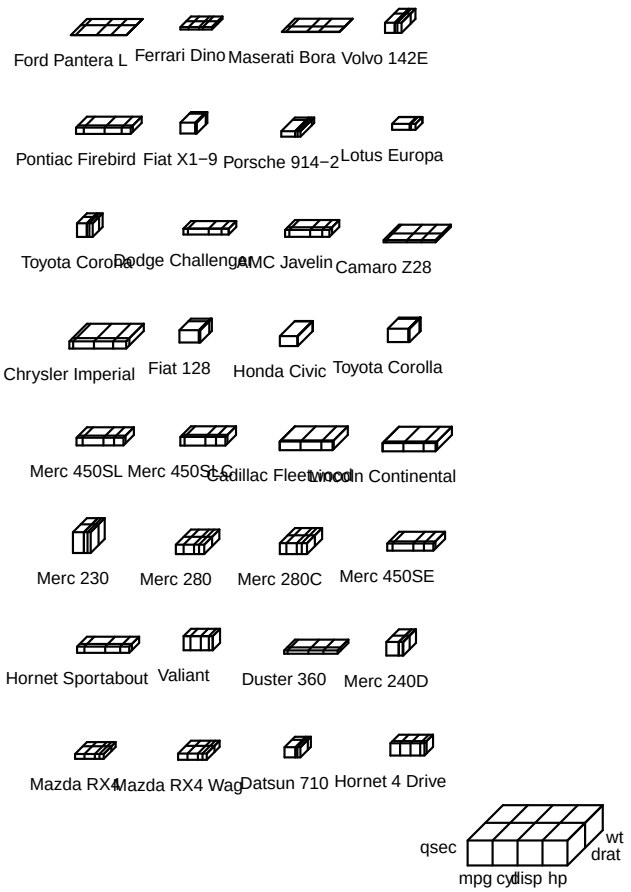


Abbildung 8.7: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Quadern. (`boxes` aus `library(StatDA)`)

8.3 Bäume (Trees)

Es handelt sich hier um eine Darstellung der Daten in der Form eines Baumgerüstes. Es wird dabei versucht, die Korrelation zwischen den Variablen auch in der Darstellung der Daten zum Ausdruck zu bringen. Diese ist Ausschaggebend für die Sequenz der Verzweigungen.

Der nachfolgende Algorithmus beschreibt die Konstruktion eines Hartigan Baumes. Es geht dabei um die Dicke und Länge der Äste, um die Winkel zwischen den Ästen und zwischen dem Stamm und den Ästen:

1. Die Dicke eines Astes ist proportional zur Anzahl der Äste oberhalb des Astes (der Cluster Tree gibt an, was oberhalb ist).
2. Winkel zwischen 2 Ästen: Es werden ein minimaler Winkel und ein maximaler Winkel vorgegeben. Die Spannweite (Range) der Winkel erhöht sich mit der Anzahl der Variablen.
3. Die Richtung der Äste sollte so erfolgen, dass keine Überschneidungen stattfinden.
4. Winkel:
   (a) Der Winkel eines Astes mit der Vertikalen ist proportional zur Dicke des Astes.
   (b) Der Winkel des Stammes mit der Vertikalen ist invers proportional zur Dicke.
   (c) Die Summe der beiden Winkel zwischen den Ästen einer Verzweigung und der Vertikalen ist durch Punkt 2) gegeben.
5. Die Länge eines Astes ist proportional zur mittleren Länge aller Variablen über der Verzweigung.

Abbildung 8.8 zeigt die Darstellung der Fahrzeugdaten mittels Bäumen.

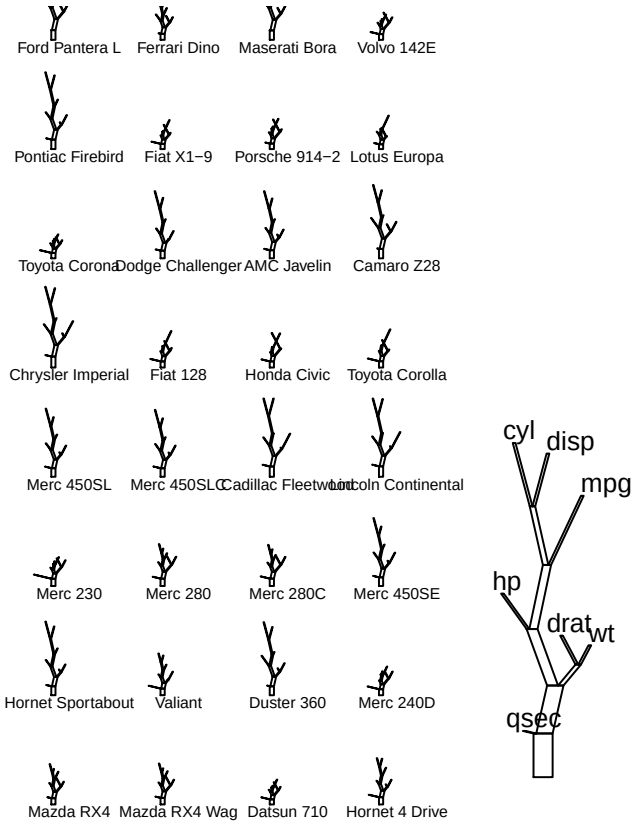


Abbildung 8.8: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Bäumen. (`tree` aus `library(StatDA)`)

8.4 Burgen (Castles)

Als Alternative zu den Bäumen kann auch die Darstellungsform “Burgen” gewählt werden. Das Prinzip der Konstruktion ist analog zu Bäumen, nur dass hier die Winkel von den Verzweigungen gleich Null sind. Abbildung 8.9 zeigt die Darstellung der Fahrzeugdaten mittels Burgen.

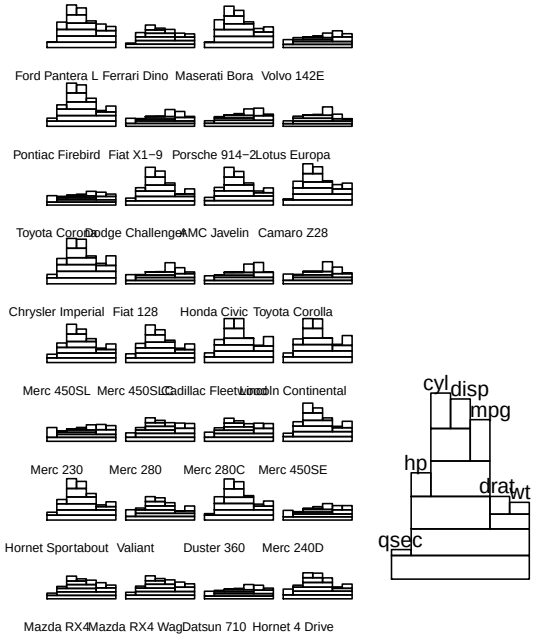


Abbildung 8.9: Darstellung der Fahrzeugdaten aus Tabelle 8.2 mit Burgen. (`tree` aus `library(StatDA)`)

8.5 Plot mit parallelen Koordinaten

Die Idee ist hier, dass die einzelnen Variablen als Koordinatenachsen nebeneinandergestellt werden. Die Werte der einzelnen Variablen werden auf gleichen Bereich gebracht,

$$x_{ij}^* := \frac{x_{ij} - \min_i(x_{ij})}{\max_i(x_{ij}) - \min_i(x_{ij})} \qquad \text{für} \qquad j = 1, \ldots, p.$$

Dann werden Objekte in Form von Linien entsprechend deren Koordinaten aufgetragen. Man kann duch verschiedene Farben oder Graustufen, sowie durch verschiedene Strichstärken oder Linientypen weitere Information im Plot integrieren.

In Abbildung 8.10 werden die Iris-Daten mit parallelen Koordinaten dargestellt. Die Gruppeninformation wird durch unterschiedliche Farben visualisiert.

Als weiteres Beispiel für einen Plot mit parallelen Koordinaten werden die Daten `glass` aus der `library(chemometrics)` in Abbildung 8.11 dargestellt. Dabei sind Konzentrationen

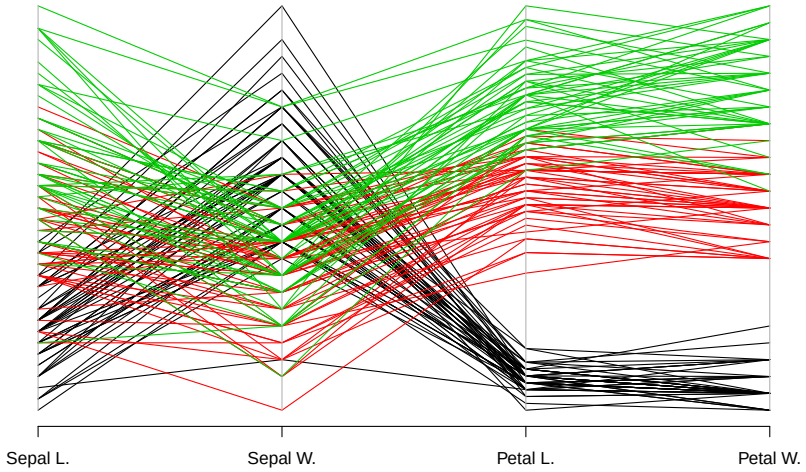


Abbildung 8.10: Darstellung der Iris-Daten mit parallelen Koordinaten. (`parcoord`)

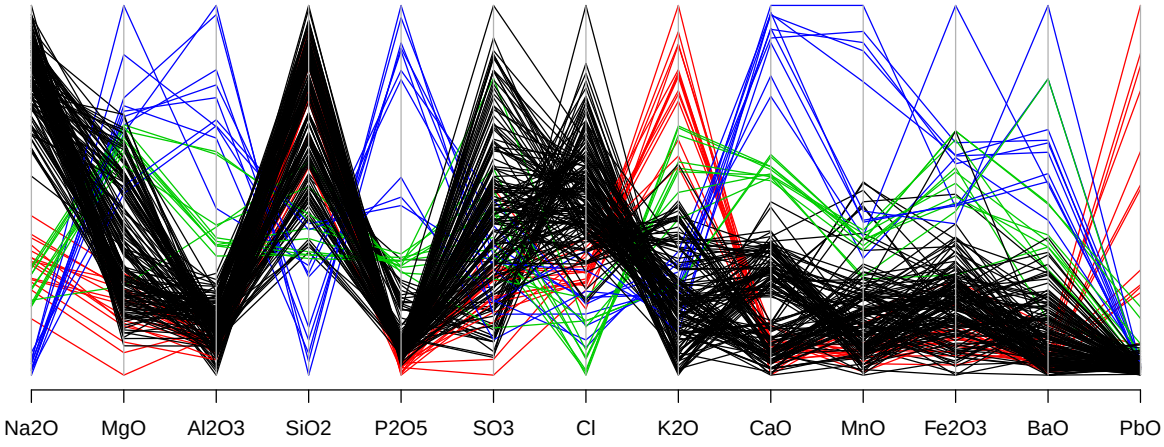


Abbildung 8.11: Darstellung der Glas-Daten mit parallelen Koordinaten.

verschiedener Oxide gemessen worden. Die Farben entsprechen verschiedenen Typen von Glas.

Bemerkungen: Diese Visualisierungsart hat u.a. folgende Eigenschaften:

- Die Datenqualität (Rundungseffekte, extreme Ausreißer) wird mit einem Blick erkannt. Kategorielle Variablen verleiten allerdings oft zu einer subjektiven Clusterung.
- Die Reihenfolge der Variablen (Koordinaten) in der Darstellung kann entscheidend sein für die Übersichtlichkeit.
- Dieser Plot "verträgt" ohneweiters eine sehr hohe Anzahl von Objkten, aber auch viele Dimensionen können dargestellt werden.
- Trends oder Cluster werden relativ leicht erkannt, ebenso Werte, die völlig von der Datenstruktur abweichen (diese haben einen ganz anderen Linienverlauf).

Kapitel 9

Parameterschätzung im Mehrdimensionalen

In Kapitel 3 wurden bereits wichtige Schätzer für univariate Daten aufgelistet. Hier möchten wir eine Erweiterung auf den multivariaten Fall machen. Wir nehmen dazu an, dass Messungen von p Variablen (Merkmalen) $x_1, \ldots, x_p$ vorliegen, und nicht nur *eine* univariate Größe x. Natürlich werden wir später wieder für jede dieser Variablen n Beobachtungen benötigen, nämlich die konkreten multivariaten Daten.

9.1 Kovarianz und Korrelation

Die Kovarianz und die Korrelation stellen ein Maß für den Zusammenhang dieser Variablen dar. Meist werden nur Maße für den *linearen* Zusammenhang angegeben. Die Korrelation kann besser als die Kovarianz interpretiert werden, weil sie eine standardisierte Größe ist.

Wir gehen zunächst davon aus, dass $x_1, \ldots, x_p$ Zufallsvariablen sind, deren Realisierungen später die $n \times p$ Datenmatrix $\boldsymbol{X}$ bilden. Die **Kovarianz** zwischen dem Paar x_j und x_k $(j, k \in \{1, \ldots, p\})$ ist definiert als:

$$\sigma_{jk} = \mathrm{E}[(x_j - \mathrm{E}(x_j))(x_k - \mathrm{E}(x_k))]$$

Das Symbol "E" beschreibt den üblichen Erwartungswert, der für eine konkrete Stichprobe meist durch das arithmetische Mittel geschätzt wird. Für $j = k$ erhält man demnach $\sigma_{jj} = \sigma_{kk}$, die *Varianz*.

Hat man nun konkrete Daten gegeben, also die Beobachtungen wurden simultan an jeder Variable gemessen, dann gibt es insbesondere für x_j die Werte $x_{1j}, \ldots, x_{nj}$ und für x_k die Werte $x_{1k}, \ldots, x_{nk}$. Die *klassische Schätzung der Kovarianz* σ_{jk} wird als *Stichprobenkovarianz* bezeichnet, und sie ist definiert als

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k),$$

mit den arithmetischen Mitteln

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad \text{und} \quad \bar{x}_k = \frac{1}{n} \sum_{i=1}^{n} x_{ik}.$$

Setzt man nun wieder $j = k$, so erhält man die *Stichprobenvarianz*.

Der **Korrelationskoeffizient** zwischen den Zufallsgrößen x_j und x_k ist definiert als

$$\rho_{jk} = \frac{\sigma_{jk}}{\sqrt{\sigma_{jj}\sigma_{kk}}}$$

und ist ein dimensionsloses Maß für den linearen Zusammenhang dieser beiden Merkmale. Diese Größe nimmt immer einen Wert im Intervall $[-1, 1]$. Bei einem Wert von 1 bzw. -1 enthalten x_j und x_k die gleiche Information, bei -1 besteht ein umgekehrter Zusammenhang. Bei einem Wert von 0 besteht kein linearer Zusammenhang zwischen x_j und x_k.

Die *klassische Schätzung der Korrelation* ρ_{jk} wird als *Stichprobenkorrelation* bezeichnet, und sie ist definiert als

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}s_{kk}}}.$$

Dieses Korrelationsmaß misst die *lineare Beziehung* zwischen x_j und x_k. Ist also $r_{jk} = 0$, so besteht kein *linearer* Zusammenhang zwischen x_j und x_k, es könnte aber sehr wohl einen nichtlinearen Zusammenhang geben.

Wenn nun die Kovarianz bzw. die Korrelation für alle Paare von Variablen ermittelt wird, so erhält man Matrizen der Ordnung $p \times p$. Die (theoretische) *Kovarianzmatrix* wird mit $\boldsymbol{\Sigma}$ bezeichnet, sie enthält die Elemente σ_{jk}. Schätzt man diese Elemente mit der Stichprobenkovarianz s_{jk} so erhält man damit die $p \times p$ Stichproben-Kovarianzmatrix $\boldsymbol{S}$. Analog, wenn man die Korrelationen ρ_{jk} mit den Stichprobenkorrelationen r_{jk} schätzt, dann ergibt sich die Matrix $\boldsymbol{R}$.

```
R:   S <- cov(X)     # Stichproben-Kovarianzmatrix
R:   R <- cor(X)     # Stichproben-Korrelationsmatrix
```

Abbildung 9.1 sollte ein Gefühl dafür vermitteln, wie eine theoretisch definierte Kovarianzmatrix praktisch aussehen könnte. Die unter den Abbildungen stehenden theoretischen Kovarianzmatrizen resultieren in den theoretischen Korrelationen 0.8, 0, und -0.8, und die gezeigte Punktwolke stammt von 200 zufällig erzeugten Werten mit dieser Kovarianzmatrix. Würde man von den Punkten nun die Stichprobenkorrelationen ermitteln, so müsste der Wert sehr nahe dem theoretischen Wert sein.

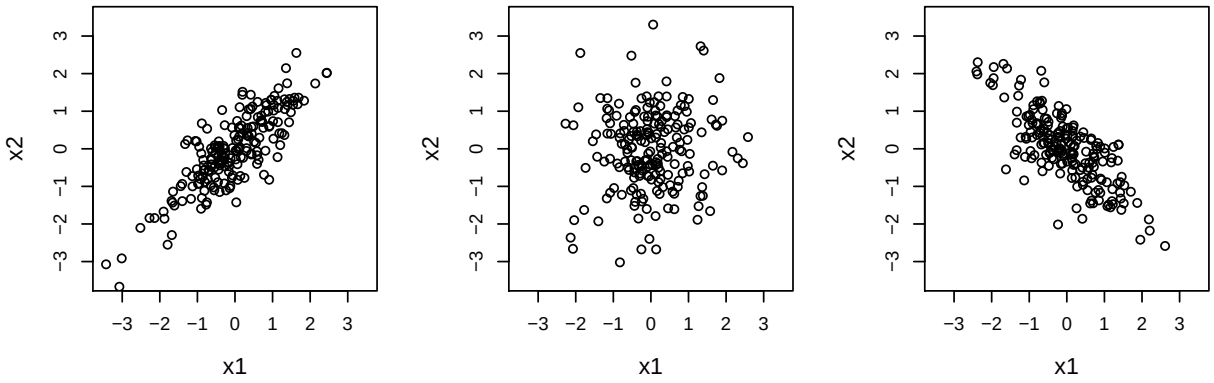


Abbildung 9.1: Punktwolken mit Realisierungen zu den theoretischen Kovarianzmatrizen, entsprechend den Korrelationen 0.8, 0, und -0.8.

9.1.1 Robustere Schätzung der Kovarianz und Korrelation

Wie schon in früheren Kapiteln festgestellt, sind arithmetisches Mittel und Stichprobenvarianz sehr empfindlich gegenüber Ausreißern. Robustere Schätzer sind zum Beispiel Median bzw. MAD. Analog ist die Stichprobenkovarianz empfindlich gegenüber Ausreißern, weil eben eine (beliebig) weit weg liegende Beobachtung sehr großen Einfluss auf das Ergebnis haben kann.

Robustere Schätzungen der Kovarianz bzw. Korrelation kann man erhalten, indem Beobachtungen, die vom "Hauptteil" der gemeinsamen Datenstruktur abweichen, weniger Gewicht verliehen wird. Im Extremfall könnte das Gewicht sogar 0 sein, womit die Beobachtung gar nicht mehr direkt eingeht.

Ein Ansatz zur Robustifizierung ist die **Spearman Rang-Korrelation**, wo nicht die Datenwerte selbst, sondern nur die Ränge der Werte von den einzelnen Variablen genommen werden. Anschließend wird von diesen Rängen die Stichprobenkorrelation berechnet. Der Rang ist immer eine Zahl aus $\{1, \ldots, n\}$ (bei n Beobachtungen), und es spielt keine Rolle, wie extrem die Werte von Ausreißern sind. Ränge können auch Nichtlinearitäten gut abbilden, womit das resultierende Maß ein Maß nicht nur für den linearen sondern auch für den *nichtlinearen* Zusammenhang ist.

```
R:  cor(X,method="spearman")
```

Eine robustere Variante erhält man z.B. mit dem **MCD** (*Minimum Covariance Determinant*) **Schätzer**. Wie bereits der Name verrät, wird dabei die Determinante der Kovarianzmatrix minimiert. Es werden dabei aber nicht alle n, sondern nur $h < n$ Beobachtungen genommen, von denen die Stichproben-Kovarianzmatrix berechnet wird. Deren Determinante definiert dann das Zielkriterium, und mit einem Algorithmus sucht man nach jenen h Beobachtungen, die zur kleinsten Determinante führen. Für h empfiehlt sich, etwa die Hälfte oder 3/4 der Beobachtungen zu nehmen. Die robuste Kovarianzschätzung ist somit die empirische Kovarianzmatrix dieser h Beobachtungen (multipliziert mit einem Faktor für Konsistenz bei Normalverteilung). Man erhält als "Nebenprodukt" auch eine robuste Lokationsschätzung mit dem arithmetischen Mittel der h Beobachtungen.

```
R:  library(robustbase)
R:  covMcd(X)
```

Die robuste Kovarianzschätzung mittels MCD kann unmittelbar dazu verwendet werden, um eine robuste Schätzung der Korrelationsmatrix zu erhalten. Sei c_{jk} das Element (j,k) der MCD-Kovarianzmatrix, für $j, k = 1, \ldots, p$. Die robusten Varianzen sind dann gegeben durch c_{jj}. Die Korrelation ist definiert als Kovarianz durch die Wurzeln aus den Varianzen, also ist

$$\frac{c_{jk}}{\sqrt{c_{jj}}\sqrt{c_{kk}}}$$

das Element (j,k) der robusten Korrelationsmatrix.

9.2 Distanz und Ähnlichkeit

Wärend im letzten Abschnitt die Beziehungen zwischen den Variablen im Vordergrund waren, wird nun die Beziehungen zwischen den Objekten untersucht. Wir betrachten wiederum Beobachtungen im p-dimensionalen Raum, insbesondere die Beobachtungen $\boldsymbol{x}_A = (x_{A1}, \ldots, x_{Ap})^T$ und $\boldsymbol{x}_B = (x_{B1}, \ldots, x_{Bp})^T$.

Die **Euklidische Distanz** zwischen $\boldsymbol{x}_A$ und $\boldsymbol{x}_B$ ist definiert als

$$d_E(\boldsymbol{x}_A, \boldsymbol{x}_B) = \left(\sum_{j=1}^{p} (x_{Bj} - x_{Aj})^2 \right)^{1/2} = [(\boldsymbol{x}_B - \boldsymbol{x}_A)^T (\boldsymbol{x}_B - \boldsymbol{x}_A)]^{1/2} = \|\boldsymbol{x}_B - \boldsymbol{x}_A\|.$$

`R: dist(X,method="euclidean")`

Die **Manhattan Distanz** (auch *city block* Distanz) nimmt hingegen die absoluten koordinatenweisen Abstände:

$$d_M(\boldsymbol{x}_A, \boldsymbol{x}_B) = \sum_{j=1}^{p} |x_{Bj} - x_{Aj}|$$

`R: dist(X,method="manhattan")`

Verallgemeinert werden obige zwei Distanzmaße mit der **Minkowski Distanz**, definiert als

$$d_{Mink}(\boldsymbol{x}_A, \boldsymbol{x}_B) = \left(\sum_{j=1}^{p} (x_{Bj} - x_{Aj})^m \right)^{1/m}$$

`R: dist(X,method="minkowski",p=m)`

Der **Kosinus des Winkels** α zwischen den Objekt-Vektoren ist ein Ähnlichkeitsmaß. Es ist von der Länge der Vektoren unabhängig und berücksichtigt daher nur die relativen Werte der Variablen:

$$\cos\alpha = \frac{\boldsymbol{x}_A^T \boldsymbol{x}_B}{\sqrt{(\boldsymbol{x}_A^T \boldsymbol{x}_A)(\boldsymbol{x}_B^T \boldsymbol{x}_B)}} = \frac{\boldsymbol{x}_A^T \boldsymbol{x}_B}{\|\boldsymbol{x}_A\| \cdot \|\boldsymbol{x}_B\|}$$

Allgemein kann man ein Distanzmaß d leicht in ein Ähnlichkeitsmaß umwandeln, indem man z.B. $1 - d/d_{max}$ rechnet, wobei d_{max} die maximale Distanz zwischen allen Objektpaaren ist.

Abbildung 9.2 zeigt diese verschiedenen Distanzmaße für Beobachtungen im Zweidimensionalen.

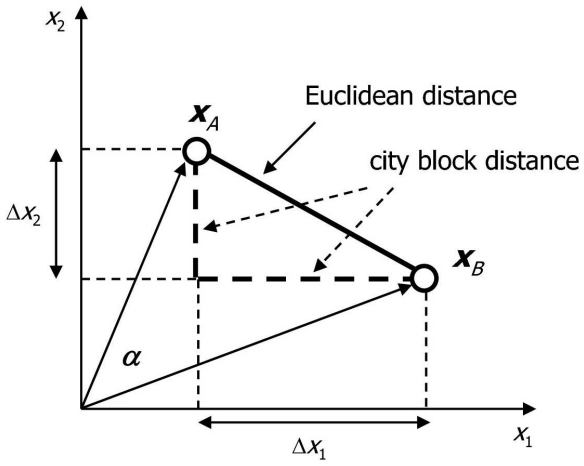


Abbildung 9.2: Verschiedene Distanzmaße, im Zweidimensionalen visualisiert.

Die **Mahalanobis Distanz** ist ein besonders wichtiges statistisches Distanzmaß, weil es die Kovarianzstruktur mitberücksichtigt. Sie hängt nicht von der Skalierung der Variablen ab, und ist definiert als

$$d_{Mahal}(\boldsymbol{x}_A, \boldsymbol{x}_B) = [(\boldsymbol{x}_B - \boldsymbol{x}_A)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_B - \boldsymbol{x}_A)]^{1/2}.$$

Würde hier die inverse Kovarianzmatrix gleich der Einheitsmatrix $\boldsymbol{I}$ gesetzt werden, so würde man die euklidische Distanz erhalten. Interessant ist hier oft die Distanz einer Beobachtung zum Zentrum $\boldsymbol{\mu}$ der Verteilung. Man erhält dann die Formel

$$d_{Mahal}(\boldsymbol{x}_i, \boldsymbol{\mu}) = [(\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})]^{1/2} \qquad \text{für } i = 1, \ldots, n.$$

Abbildung 9.3 (rechts) zeigt verschiedene Niveaus der Mahalanobis Distanz zum Zentrum der Verteilung in Form von Ellipsen dargestellt. Im Zentrum hat man also kleine Distanzen, die gegen den Rand der Punktwolke wachsen. Im Vergleich dazu ist links die euklidische Distanz visualisiert, die sich mit $\boldsymbol{\Sigma} = \boldsymbol{I}$ ergibt. Dieses Distanzmaß kann klarerweise nicht beschreiben, wie weit man an den Rand der Punktwolke kommt – außer man hat eine kugelsymmetrische Verteilung.

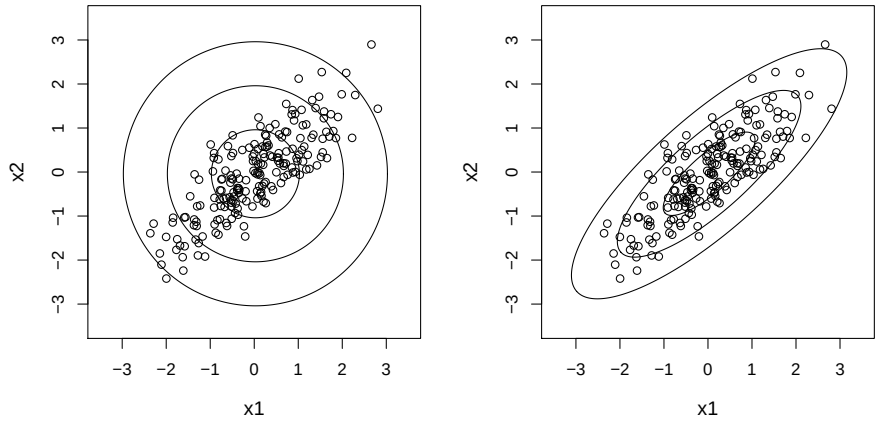


Abbildung 9.3: Euklidische Distanz (links) und Mahalanobis Distanz (rechts) zum Zentrum der Verteilung, in Form von Ellipsen dargestellt.

9.3 Multivariate Ausreißererkennung

Das Konzept der Mahalanobis Distanz kann gut verwendet werden, um Ausreißer in multivariaten Daten zu identifizieren. Ausreißer können also solche Beobachtungen sein, die extrem in einer Dimension sind (hier wären sie einfach univariat zu erkennen) und solche, die in verschiedenen Dimensionen verborgen sind. Beide Typen sollen mit folgender Methodik erkannt werden.

Die Mahalanobis Distanz einer Beobachtung zum Zentrum der Verteilung,

$$d_{Mahal}(\boldsymbol{x}_i, \boldsymbol{\mu}) = [(\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})]^{1/2} \qquad \text{für } i = 1, \ldots, n,$$

zeigt an, wie weit die Beobachtung von diesem Zentrum entfernt ist, relativ zur zugrunde liegenden Kovarianzstruktur. Beobachtungen mit grosser Mahalanobis Distanz wären somit Kandidaten als Ausreißer. Was aber heißt "groß"? Unter bestimmten Voraussetzungen (u.A. multivariate Normalverteilung) kann man zeigen, dass die quadrierten Mahalanobis Distanzen einer Chiquadrat-Verteilung mit p Freiheitsgraden, χ^2_p, folgen. Definiert man nun ein Quantil dieser Verteilung als Schranke, z.B. $\chi^2_{p;0.975}$, so kann eine Ausreißerregel erstellt werden: Beobachtungen, deren quadrierte Mahalanobis Distanz größer als diese Schranke sind, werden als potentielle Ausreißer deklariert.

Nachdem bei der Formel für die Mahalanobis Distanz sowohl das Zentrum $\boldsymbol{\mu}$ als auch die Kovarianzmatrix $\boldsymbol{\Sigma}$ eingehen, müssen beide Größen aus den Daten geschätzt werden. Würde man hierfür die klassischen Schätzer, arithmetisches Mittel und Stichprobenkovarianz, nehmen, dann hätte man ein schlechtes Werkzeug für Ausreißererkennung, weil ja diese Schätzer genau von den Ausreißern beeinflusst werden. Man benötigt also *robuste Schätzer* für Lokation und Kovarianz. Beides liefert der MCD Schätzer.

```
R:  library(robustbase)
R:  plot(covMcd(X))
```

Abbildung 9.4 zeigt den Unterschied, der sich bei der Berechnung der Mahalanobis Distanz ergibt, wenn klassische bzw. robuste Schätzer verwendet werden. Um dies gut visualisieren zu können, wurden nur 2-dimensionale Daten verwendet, und zwar zwei Variablen aus den Daten `glass` der `library(chemometrics)`. Die Schranke für Ausreißer entspricht $\sqrt{\chi^2_{2;0.975}} = 2.72$, und diese ist als Ellipse in den Plots erkennbar. In der linken Grafik wurden die klassischen Schätzer verwendet, während bei der rechten Grafik der MCD genommen wurde. Mit robuster Schätzung erhält man eine ganze Gruppe von Ausreißern, die hier einem bestimmten Glastyp (Symbol) entspricht. Abbildung 9.5 zeigt nun auch direkt die erhaltenen Werte der

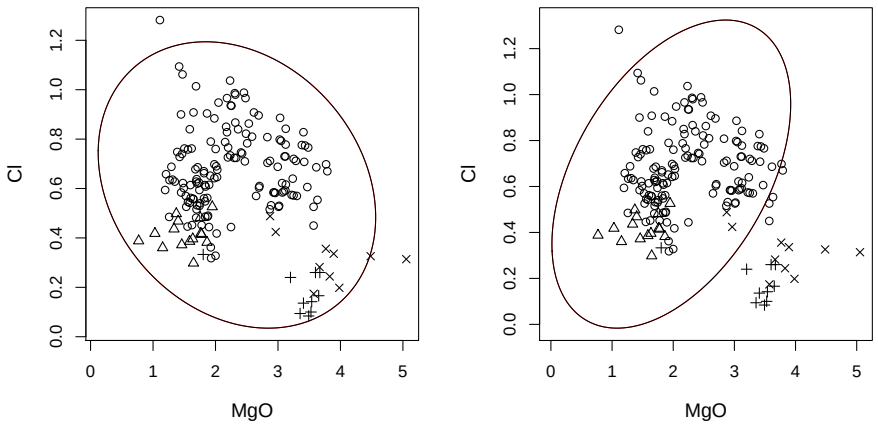


Abbildung 9.4: Schranken für Ausreißer visialisiert mit einer Ellipse, links mit klassischen und rechts mit robusten Schätzern ermittelt.

Mahalanobis Distanzen für die klassische (links) und robuste (rechts) Schätzung.

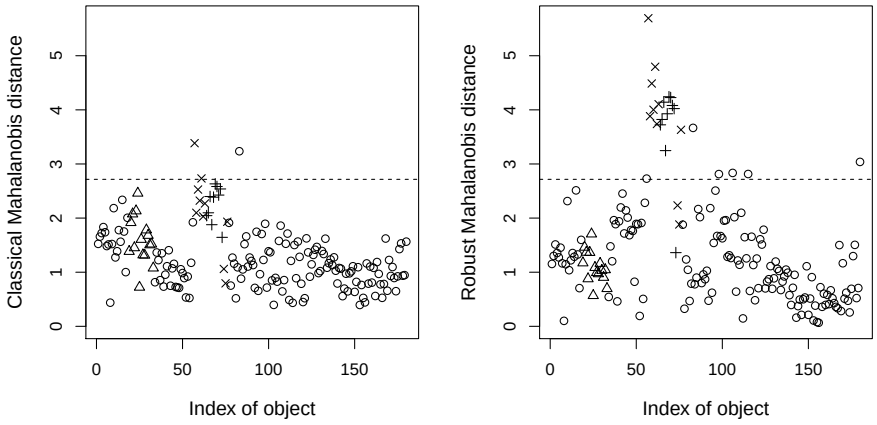


Abbildung 9.5: Mahalanobis Distanzen basierend auf klassischer (links) und robuster (rechts) Schätzung.

Kapitel 10

Projektionen mehrdimensionaler Daten

In diesem Kapitel geht es um die Reduktion der Dimension multivariater Daten durch Projektionen auf Unterräume. Es wird versucht, Projektionen der Daten zu finden, die noch immer bestimmte Eigenschaften der Daten zum Ausdruck bringen und damit in Bezug auf diese Eigenschaften möglichst wenig Informationsverlust bringen. Ziel ist eine 2-D oder 3-D Darstellung der Daten mit Hilfe dieser Transformationen.

10.1 Linearkombinationen von Variablen

Wir gehen wieder davon aus, dass p Variablen $x_1, \ldots, x_p$ vorliegen. Anstatt alle p Dimensionen separat zu betrachten, könnte man durch Linearkombination eine einzige Variable erhalten. So eine Linearkombination hat die Form

$$u = b_1 x_1 + b_2 x_2 + \ldots + b_p x_p,$$

mit Koeffizienten $b_1, b_2, \ldots, b_p$. Es werden also alle Variablen linear verknüpft und zu einer neuen Variable u zusammengefasst. Wäre z.B. $b_1 = 1$ und $b_2 = \ldots = b_p = 0$, so würde $u = x_1$ sein. Für $b_1 = \ldots = b_p = 1$ liefert jede Variable den gleichen Beitrag zu u. Die Koeffizienten können somit als *Gewicht* verstanden werden, mit dem die entsprechende Variable zu u beiträgt.

Linearkombinationen können auch *geometrisch interpretiert* werden. Nachdem die Variablen $x_1, \ldots, x_p$ die Dimensionen (in einer Grafik die Achsen) ausmachen, werden mit den Koeffizienten $b_1, \ldots, b_p$ diese Dimensionen gewichtet und zusammengefasst zu einer neuen Richtung im p-dimensionalen Raum. Diese Richtung entspricht einer linearen Projektion. Wenn für $x_1, \ldots, x_p$ schließlich n konkrete Beobachtungen vorliegen, besteht auch u aus n Beobachtungen, die vom p-dimensionalen Raum auf diese Richtung projiziert wurden.

Die Koeffizienten $b_1, \ldots, b_p$ werden als *Ladungen* (*loadings*) bezeichnet, und u wird als *score* bezeichnet.

Mit der Schreibweise $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ und $\boldsymbol{b} = (b_1, \ldots, b_p)^T$ erhält man u auch durch

$$u = \boldsymbol{x}^T \boldsymbol{b}.$$

Die Länge des Richtungsvektors $\boldsymbol{b}$ wird meist auf 1 normiert, es gilt also $\boldsymbol{b}^T \boldsymbol{b} = 1$.

Im Allgemeinen wird man nicht nur an einer Projektionsrichtung (Linearkombination) interessiert sein, sondern an mehreren. Dies kann einfach erreicht werden, indem eine Matrix $\boldsymbol{B}$ von Koeffizienten definiert wird, bestehend aus den Spalten $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$. Jeder dieser k Vektoren enthält p Koeffizienten, die die k Richtungen definieren. Man erhält somit k Linearkombinationen

$$u_j = \boldsymbol{x}^T \boldsymbol{b}_j \qquad \text{für } j = 1, \ldots, k,$$

die jeweils verschiedene "Einblicke" in den p-dimensionalen Raum geben. Man fordert oft, dass man möglichst unterschiedliche "Einblicke" erhalten möchte, dass also die Richtungen zueinander normal stehen. Dies wird mit *orthogonal* bezeichnet, und drückt sich mathematisch aus durch $\boldsymbol{b}_i^T \boldsymbol{b}_j = 0$, für $i, k = 1, \ldots, p$, $i \neq j$.

Bisher wurden keine Bedingungen an die Projektionsrichtungen gestellt. Mit solchen Bedingungen kann man aber erreichen, dass die Koeffizienten eindeutig bestimmbar werden. Man könnte z.B. an einer Projektionsrichtung interessiert sein, die zwei vorhandene Gruppen in den Daten möglichst gut unterscheiden lässt. Die erhaltene Richtung ist damit eindeutig festgelegt. Ein anderes Kriterium könnte lauten, dass die Varianz der projizierten Punkte maximal wird (die Richtung also möglichst informativ wird). Dies führt zur ersten Hauptkompomente, was Inhalt des nächsten Abschnittes ist.

10.2 Hauptkomponenten

Die Hauptkomponentenanalyse (Principal Component Analysis – PCA) zählt zu den wichtigsten Methoden der multivariaten Statistik. Hauptkomponenten werden häufig als explorativ Werkzeug eingesetzt, das einen Überblick über die Daten liefert.

10.2.1 Definition der Hauptkomponenten

Wir gehen wieder von einer $n \times p$ Datenmatrix $\boldsymbol{X}$ aus. Die Hauptkomponenten werden über Linearkombinationen folgendermaßen definiert:

$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{B}$$

Die $p \times p$ Matrix $\boldsymbol{B}$ enthält Koeffizienten, die p neue Richtungen definieren. $\boldsymbol{U}$ ist die *score* Matrix der Dimension $n \times p$ (also gleiche Dimension wie $\boldsymbol{X}$, die die projizierten Datenwerte enthält. Die einzelnen Spalten $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ der *score* Matrix werden als *Hauptkomponenten* bezeichnet.

Die einzelnen Spalten $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p$ der *Ladungsmatrix* $\boldsymbol{B}$ werden mit folgenden Kriterien *eindeutig bestimmt*:

- Die Koeffizienten $\boldsymbol{b}_1$ zur Bestimmung der ersten Hauptkomponente $\boldsymbol{u}_1$ werden so gewählt, dass die Varianz von $\boldsymbol{u}_1$ maximal ist. Außerdem wird $\boldsymbol{b}_1^T \boldsymbol{b}_1 = 1$ gefordert.
- Die Koeffizienten $\boldsymbol{b}_2$ zur Bestimmung der zweiten Hauptkomponente $\boldsymbol{u}_2$ werden so gewählt, dass die Varianz von $\boldsymbol{u}_2$ maximal ist. Außerdem wird $\boldsymbol{b}_1^T \boldsymbol{b}_2 = 0$ und $\boldsymbol{b}_2^T \boldsymbol{b}_2 = 1$ gefordert.
- Die Koeffizienten $\boldsymbol{b}_j$ zur Bestimmung der j-ten Hauptkomponente $\boldsymbol{u}_j$ $(2 < j \leq p)$ werden so gewählt, dass die Varianz von $\boldsymbol{u}_j$ maximal ist. Außerdem wird $\boldsymbol{b}_j^T \boldsymbol{b}_l = 0$ (für $1 \geq l < j$) und $\boldsymbol{b}_j^T \boldsymbol{b}_j = 1$ gefordert.

Abbildung 10.1 zeigt für 10 Datenpunkte im Zweidimensionalen, wie die erste Hauptkomponente aussehen könnte. Die auf diese Komponente projizierten Punkte haben also maximale Varianz. Man kann somit keine andere Richtung mit höherer Varianz finden.

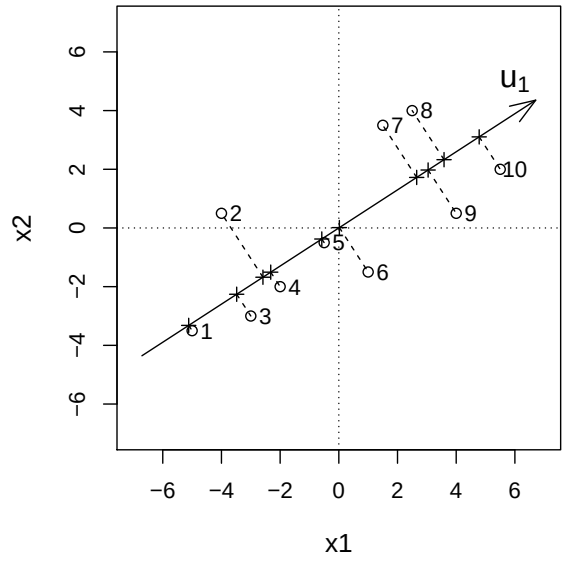


Abbildung 10.1: Ermittlung der ersten Hauptkomponente mit maximaler Varianz.

Geometrisch betrachtet erhält man mit $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{B}$ nichts anderes als eine Darstellung der Daten in einem neuen Koordinatensystem, das sogar orthogonal ist (die Spalten von $\boldsymbol{B}$ sind orthogonal). Somit enthält $\boldsymbol{U}$ exakt die selbe Information wie $\boldsymbol{X}$, nur in einer anderen Darstellung. Die Darstellung ist aber speziell, weil laut Konstruktion der Informationsgehalt (Varianz) der ersten Hauptkomponente am größten ist, und mit steigender Nummer der Hauptkomponenten dieser Informationsgehalt abnimmt. Man kann also davon ausgehen, dass die letzten Hauptkomponenten uninformativ sind, und somit weggelassen werden können. Damit würde man eine Dimensionsreduktion erreichen, weil die ersten paar Hauptkomponenten "ausreichend" sind. Genauere Kriterien für die relevante Anzahl der Hauptkomponenten kommen in Abschnitt 10.2.3.

10.2.2 Algorithmus zur Bestimmung der Hauptkomponenten

Mathematisch gesehen ist die Bestimmung der Hauptkomponenten ein Maximierungsproblem unter Nebenbedingungen. Wir formulieren dieses Problem mit Zufallsvariablen $x_1, \ldots, x_p$. Der Grund dafür ist, weil es um Varianzmaximierung geht, und wir somit mit der theoretischen Varianz "Var" arbeiten können. Würde man seinen "Lieblingsschätzer" für die Varianz nehmen, so würden die Eigenschaften der resultierenden Hauptkomponenten ganz wesentlich von den Eigenschaften dieses Schätzers abhängen.

Nach obigen Definitionen ist die j-te Hauptkomponente die Linearkombonation

$$u_j = x_1 b_{1j} + \ldots + x_p b_{pj},$$

für $1 \leq j \leq p$. Der Koeffizientenvektor $\boldsymbol{b}_j = (b_{1j}, \ldots, b_{pj})^T$ ist normiert mit $\boldsymbol{b}_j^T \boldsymbol{b}_j = 1$, und ist orthogonal zu "früheren" Richtungen $\boldsymbol{b}_j^T \boldsymbol{b}_l = 0$ (für $j > l$, falls $j \geq 2$). Das Ziel ist die

Varianzmaximierung, also $\mathrm{Var}(u_j)$ soll maximiert werden. Diese ist aber

$$\mathrm{Var}(u_j) = \mathrm{Var}(x_1 b_{1j} + \ldots + x_p b_{pj}) = \boldsymbol{b}_j^T \mathrm{Cov}(x_1, \ldots, x_p)\boldsymbol{b}_j = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j.$$

Die Matrix $\boldsymbol{\Sigma}$ ist die theoretische Kovarianzmatrix bezogen auf die Population. Eine Maximierung dieses Ausdrucks unter Nebenbedingungen wird als Lagrange Problem formuliert:

$$\phi_j = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j - \lambda_j (\boldsymbol{b}_j^T \boldsymbol{b}_j - 1) \qquad \text{für } j = 1, \ldots, p,$$

mit den Langrange Parametern λ_j. Die partiellen Ableitungen nach den unbekannten Parametern $\boldsymbol{b}_j$ werden Null gesetzt, also

$$\frac{\partial \phi_j}{\partial \boldsymbol{b}_j} = 2\boldsymbol{\Sigma}\boldsymbol{b}_j - 2\lambda_j \boldsymbol{b}_j = \boldsymbol{0}.$$

Dies ist gleichbedeutend mit

$$\boldsymbol{\Sigma}\boldsymbol{b}_j = \lambda_j \boldsymbol{b}_j \qquad \text{für } j = 1, \ldots, p.$$

Man erkennt daraus, dass man zu einem *Eigenwertproblemen* kommt: $\boldsymbol{b}_j$ sind die *Eigenvektoren* von $\boldsymbol{\Sigma}$ zu den *Eigenwerten* λ_j.

Die Rolle der Eigenwerte λ_j ist auch von Bedeutung:

$$\mathrm{Var}(u_j) = \boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_j = \boldsymbol{b}_j^T \lambda_j \boldsymbol{b}_j = \lambda_j \boldsymbol{b}_j^T \boldsymbol{b}_j = \lambda_j$$

Die Eigenwerte sind somit die Varianzen der Hauptkomponenten, die maximiert wurden. Die Eigenvektoren (Spalten von $\boldsymbol{B}$) und Eigenwerte werden so geordnet, dass gilt: $\lambda_1 \geq \ldots \geq \lambda_p$.

Hinweis: Die Richtungen der Hauptkomponenten sind die Eigenvektoren der Kovarianzmatrix $\boldsymbol{\Sigma}$. Man stelle sich vor, dass die Varianz von x_1 tausend mal so hoch ist wie die Varianz der restlichen Variablen. Wird jetzt nach einer Richtung gesucht, die die Varianz maximiert, so wird diese Richtung mehr oder weniger exakt mit x_1 übereinstimmen. Ist dieser Effekt nicht erwünscht, möchte man also, unabhängig von der Varianz der einzelnen Variablen, dass jede Variable gleichberechtigt zu den Hauptkomponenten beitragen kann, so muss zuerst skaliert werden. Im einfachsten Fall sollten die Varianzen der skalierten Variablen gleich 1 sein. Dies bedeutet aber, dass man nicht Eigenvektoren von der Kovarianzmatrix, sondern von der Korrelationsmatrix nimmt. Verwendet man klassische Schätzer, so ermittelt man also nicht Eigenvektoren (und Eigenwerte) von $\boldsymbol{S}$, sondern von der empirischen Korrelationsmatrix $\boldsymbol{R}$.

R: `X.pca <- princomp(X,cor=TRUE)`

10.2.3 Anzahl der relevanten Hauptkomponenten

Das Ziel der Hauptkomponentenanalyse ist eine Dimensionsreduktion bei möglichst wenig Informationsverlust. Nachdem die Hauptkomponenten nach absteigender Varianz $\lambda_1 \geq \ldots \geq \lambda_p$ geordnet sind, sucht man also eine Anzahl k, mit $1 \leq k \leq p$, von Hauptkomponenten, die die "wesentliche" Information der Daten abbilden.

Nachdem die Gesamtvarianz der Daten ausgedrückt werden kann als $\lambda_1 + \ldots + \lambda_p$, entspricht der Informationsgehalt der ersten k Hauptkomponenten $\lambda_1 + \ldots + \lambda_k$. Als Kriterium eignet

sich daher der Anteil an der Gesamtvarianz, der durch die ersten k Hauptkomponenten ausgedrückt wird:

$$\frac{\lambda_1 + \ldots + \lambda_k}{\lambda_1 + \ldots + \lambda_p}$$

Dieser Anteil wird oft visualisiert. Der resultierende *scree plot* ("Geröll-Plot") gibt dann einen Hinweis auf das "optimale" k: Abbildung 10.2 (links) zeigt schematisch so einen *scree plot*. Man möchte k dort festlegen, wo der Linienzug "ausflacht". Alle Punkte, die in etwa auf einer Geraden liegen, entsprechen Hauptkomponenten mit irrelevantem Informationsgehalt, und daher können diese weggelassen werden. In der Grafik würde man sich daher für $k = 2$ entscheiden. Abbildung 10.2 (rechts) zeigt die kumulierten Varianzen der Hauptkomponenten.

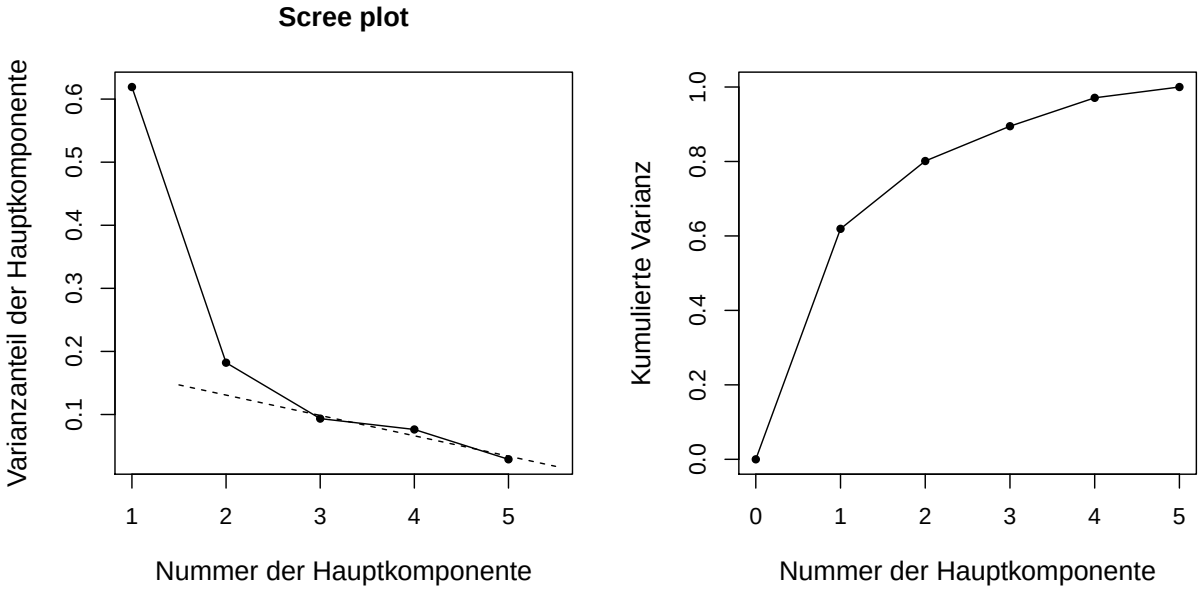


Abbildung 10.2: Scree plot (links) und Anteil an der erklärten Gesamtvarianz (rechts). Die Daten erhält man mit `data(scor,package="bootstrap")`.

Als Faustregel könnte man k so wählen, dass zumindest 80% der Gesamtvarianz erklärt sind. Dieser Anteil hängt allerdings stark davon ab, was man mit den ersten k Hauptkomponenten nun machen möchte (Visualisierung oder Modellierung).

10.2.4 Zentrieren und Skalieren der Daten

Wie bereits oben diskutiert, wird man i.A. unterschiedliche Hauptkomponenten erhalten, wenn man die unskalierten oder die skalierten Daten nimmt. Aber auch die Zentrierung der Daten wird eine Rolle spielen.

Zentrierung bezieht sich auf die Mittel der originalen Variablen. Bei üblichen Messungen wird der Mittelwert der einzelnen Variablen nicht 0 sein, die Daten sind also *unzentriert*. Würde man diese Mittelwerte auf 0 bringen, so spricht man von *zentrierten* Daten. Eine Zentrierung erreicht man, indem von der jeweiligen Spalte der Matrix das Spaltenmittel (arithmetisches Mittel, Median, etc.) abgezogen wird. Abbildung 10.3 zeigt den Effekt der Zentrierung im Zusammenhang mit der Berechnung der Hauptkomponenten. Links wurden unzentrierte Daten verwendet, rechts zentrierte. Die Richtung der ersten Hauptkomponente ist unverändert.

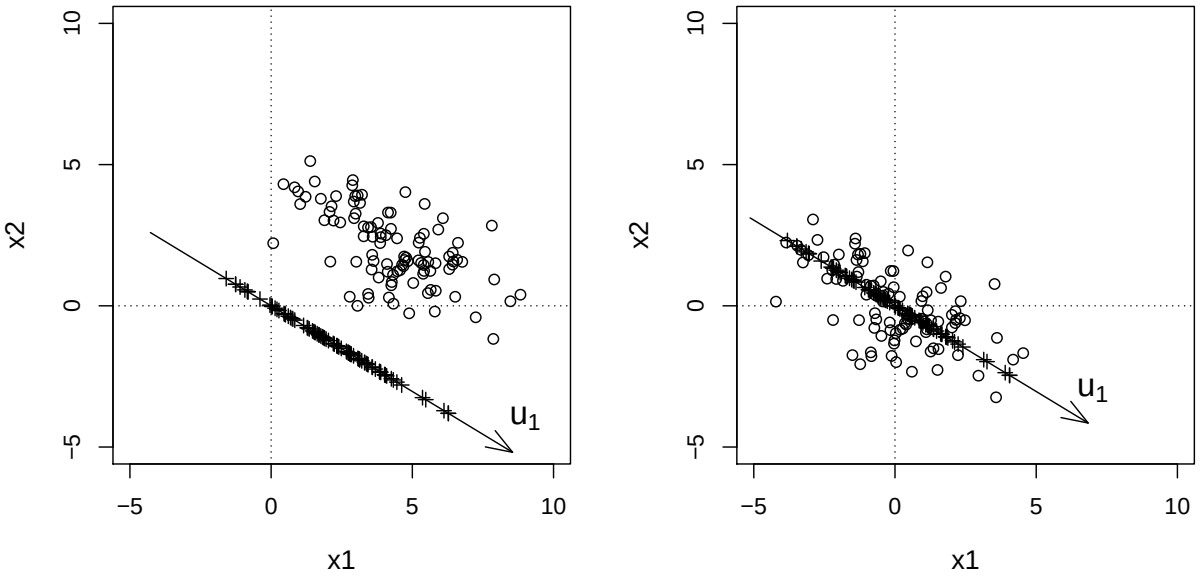


Abbildung 10.3: Effekt der Zentrierung der Datenmatrix auf Spaltenmittel Null bei der Berechnung der Hauptkomponenten: links unzentriert, rechts zentriert.

Auch das Verhältnis der projizierten Punkte (*scores*) zueinander ist unverändert. Allerdings sind im unzentrierten Fall die *scores* auch unzentriert, im zentrierten Fall sind sie zentriert.

Man sollte allgemein immer die zentrierten Daten nehmen für die Ermittlung der Hauptkomponenten, weil man i.A. nur an der Rekonstruktion der Struktur der Daten interessiert ist, nicht aber an der Rekonstruktion der Lokation.

Skalierung bezieht sich auf die Varianzen der originalen Variablen. Bei üblichen Messungen wird die Varianz der einzelnen Variablen nicht 1 sein, die Daten sind also *unskaliert*. Würde man diese Varianzen auf 1 bringen, so spricht man von *skalierten* Daten. Eine Skalierung erreicht man, indem die Werte der jeweiligen Spalte durch die Standardabweichung der Spaltenwerte (Wurzel der Stichprobenvarianz, MAD, etc.) dividiert wird. Abbildung 10.4 zeigt den Effekt der Skalierung im Zusammenhang mit der Berechnung der Hauptkomponenten. Offensichtlich ist die Varianz der Variable x_1 wesentlich höher als die von x_2. Links wurden die (zentrierten) unskalierten Daten verwendet, rechts skalierte. Die Richtung der ersten Hauptkomponente ändert sich, und somit ändern sich auch die *scores* der ersten Hauptkomponente. In der linken Grafik erkennt man, dass x_1 wesentlich mehr zur Bestimmung der Hauptkomponentenrichtung beiträgt als x_2. Möchte man diesen Effekt nicht haben, dass Variablen mit höherer Varianz automatisch höheren Beitrag leisten, so muss zuerst skaliert werden.

10.2.5 Normalverteilung und Ausreißer

Bei der Definition der Hauptkomponenten wurde nicht von irgendeiner Verteilungsvoraussetzung gesprochen. Benötigt man also multivariate Normalverteilung? Die mathematische Herleitung kommt ohne diese Voraussetzung aus.

Abbildung 10.5 (links) zeigt bivariate Daten, wo speziell die Variable x_2 sehr schief ist (rechtsschief). Die Daten wurden hier zentriert und skaliert, und die Richtung der ersten

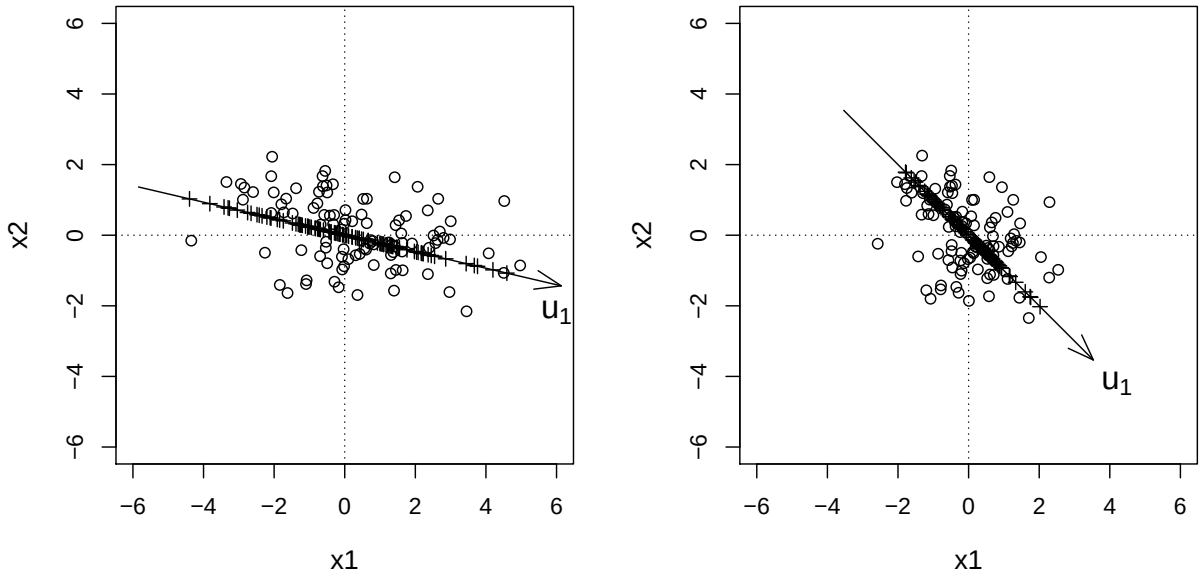


Abbildung 10.4: Effekt der Skalierung der Datenmatrix auf Spaltenvarianz 1 bei der Berechnung der Hauptkomponenten: links unskaliert, rechts skaliert.

Hauptkomponente scheint sinnvoll zu sein. Man wird aber mit einer Hauptkomponente nicht auskommen, um die Struktur in den Daten ausreichend zu beschreiben, sondern benötigt dazu auch die zweite Hauptkomponente. Wir haben also *keine* Dimensionsreduktion erreicht. In

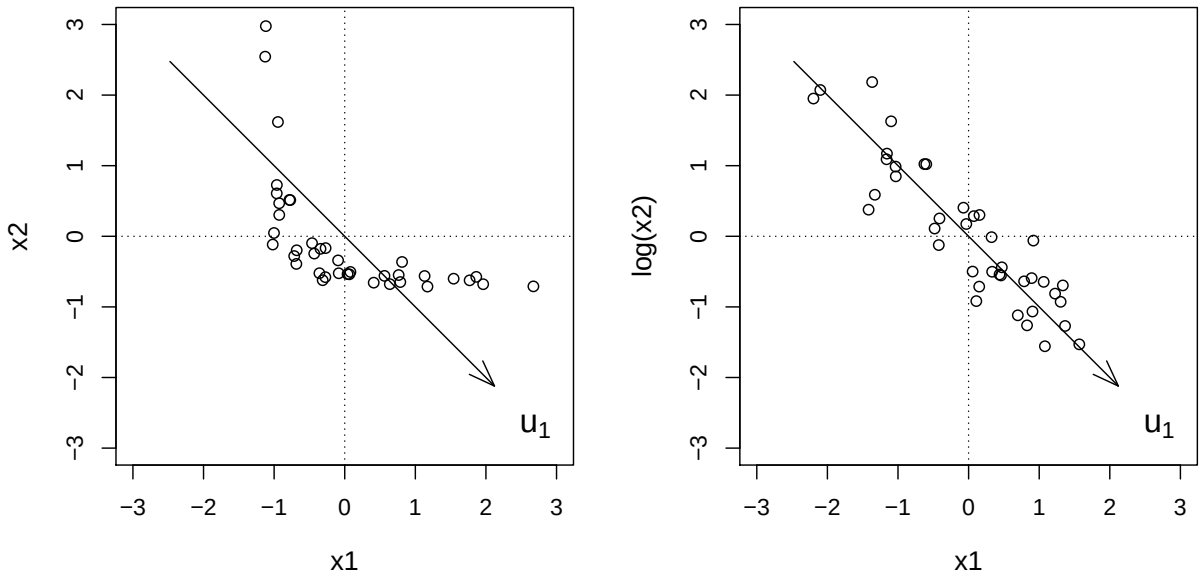


Abbildung 10.5: Asymmetrische Verteilung (links) und Symmetrisierung durch Transformation (rechts.

Abbildung 10.5 (rechts) wurde statt x_2 die Variable $\log(x_2)$ genommen, und somit symmetrie erreicht – möglicherweise sogar bivariate Normalverteilung. x_1 und $\log(x_2)$ stehen in starker linearer Beziehung, die durch die Richtung der ersten Hauptkomponente ausgedrückt wird. Eine zweite Komponente wird hier nicht nötig sein, und somit konnte die Dimensionsreduktion erfolgreich eingesetzt werden. Elliptische Symmetrie der Daten scheint somit wesentlich

für eine effiziente Dimensionsreduktion zu sein.

Eine andere Frage ist, ob *Ausreißer* die Richtung der Hauptkomponenten beeinflussen können. In Abbildung 10.6 sind zwei Gruppen von Ausreißern in den Daten sichtbar. In der linken Grafik wurde die "klassische" Schätzung der ersten Hauptkomponente durchgeführt, in der rechten Grafik eine robuste Variante. Die Richtung ändert sich offensichtlich. Klassische Schätzung heißt hier, dass die Richtungen den Eigenvektoren der "klassischen" empirischen Korrelationsmatrix $\boldsymbol{R}$ (bzw. der empirischen Kovarianzmatrix) entsprechen. Im robusten Fall werden Eigenvektoren von der robusten Schätzung der Kovarianz- oder Korrelationsmatrix genommen. So eine robuste Schätzung erhält man z.B. über den MCD-Schätzer, siehe Abschnitt 9.1.1.

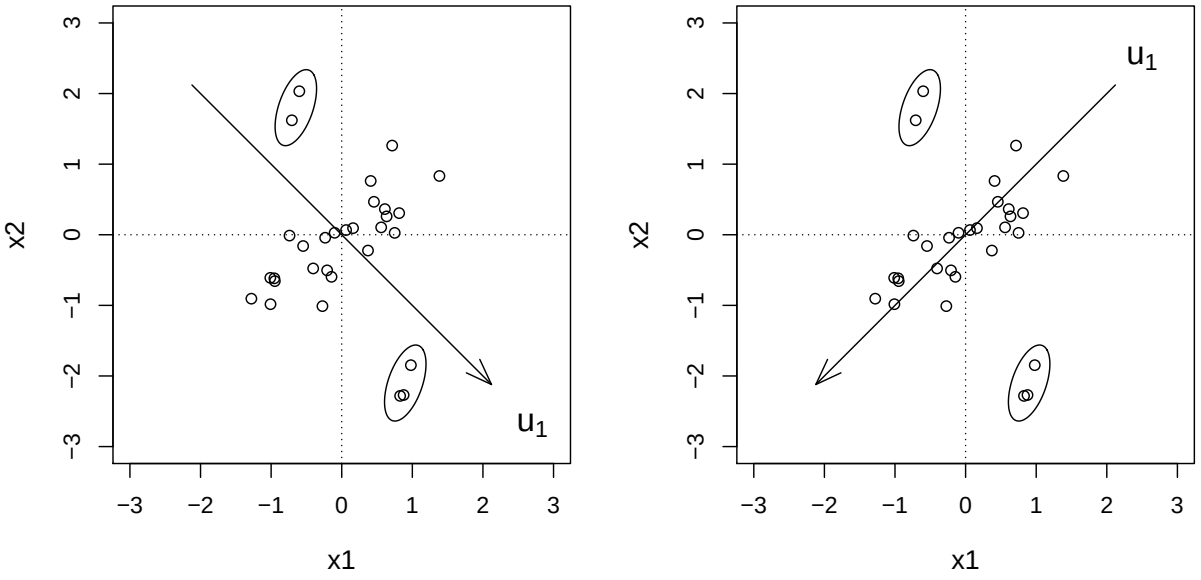


Abbildung 10.6: Effekt von Ausreißern: links die Lösung von klassischer, rechts von robuster Hauptkomponentenanalyse.

```
R:   library(robustbase)
R:   X.mcd <- covMcd(X,cor=TRUE)    # liefert auch robuste Korrelationsmatrix
R:   X.pca <- princomp(X,covmat=X.mcd,cor=TRUE)    # robuste Analyse
```

10.2.6 Darstellung der Ergebnisse, Biplot

Neben einem *scree plot* sind auch noch andere Ergebnisse der Hauptkomponentenanalyse wesentlich für eine Visualisierung, nämlich die erhaltenen *scores* $\boldsymbol{U}$ und die Ladungen (*loadings*) $\boldsymbol{B}$. Man interessiert sich dabei vor allem für die ersten k Spalten dieser Matrizen, die die relevante Information tragen. Die Matrix der *scores* ist somit von Dimension $n \times k$, und die der *loadings* hat Dimension $p \times k$. In den *scores* steckt also die Information über die Objekte, in den *loadings* die Information über die Variablen.

Beispiel: Von 88 Studenten wurden die Prüfungsergebnisse in den Fächern Mechanik, Analytische Geometrie, Lineare Algebra, Analysis und Elementare Statistik aufgezeichnet. Von

Tabelle 10.1: Prüfungsergebnisse von 88 Studenten in den Fächern Mechanik (ME), Analytische Geometrie (AG), Lineare Algebra (LA), Analysis (AN) und Elementare Statistik (ES); 100 Punkte waren erreichbar.

Student	ME	AG	LA	AN	ES	Student	ME	AG	LA	AN	ES
1	77	82	67	67	81	45	46	61	46	38	41
2	63	78	80	70	81	46	40	57	51	52	31
3	75	73	71	66	81	47	49	49	45	48	39
4	55	72	63	70	68	48	22	58	53	56	41
5	63	63	65	70	63	49	35	60	47	54	33
6	53	61	72	64	73	50	48	56	49	42	32
7	51	67	65	65	68	51	31	57	50	54	34
8	59	70	68	62	56	52	17	53	57	43	51
9	62	60	58	62	70	53	49	57	47	39	26
10	64	72	60	62	45	54	59	50	47	15	46
11	52	64	60	63	54	55	37	56	49	28	45
12	55	67	59	62	44	56	40	43	48	21	61
13	50	50	64	55	63	57	35	35	41	51	50
14	65	63	58	56	37	58	38	44	54	47	24
15	31	55	60	57	73	59	43	43	38	34	49
16	60	64	56	54	40	60	39	46	46	32	43
17	44	69	53	53	53	61	62	44	36	22	42
18	42	69	61	55	45	62	48	38	41	44	33
19	62	46	61	57	45	63	34	42	50	47	29
20	31	49	62	63	62	64	18	51	40	56	30
21	44	61	52	62	46	65	35	36	46	48	29
22	49	41	61	49	64	66	59	53	37	22	19
23	12	58	61	63	67	67	41	41	43	30	33
24	49	53	49	62	47	68	31	52	37	27	40
25	54	49	56	47	53	69	17	51	52	35	31
26	54	53	46	59	44	70	34	30	50	47	36
27	44	56	55	61	36	71	46	40	47	29	17
28	18	44	50	57	81	72	10	46	36	47	39
29	46	52	65	50	35	73	46	37	45	15	30
30	32	45	49	57	64	74	30	34	43	46	18
31	30	69	50	52	45	75	13	51	50	25	31
32	46	49	53	59	37	76	49	50	38	23	9
33	40	27	54	61	61	77	18	32	31	45	40
34	31	42	48	54	68	78	8	42	48	26	40
35	36	59	51	45	51	79	23	38	36	48	15
36	56	40	56	54	35	80	30	24	43	33	25
37	46	56	57	49	32	81	3	9	51	47	40
38	45	42	55	56	40	82	7	51	43	17	22
39	42	60	54	49	33	83	15	40	43	23	18
40	40	63	53	54	25	84	15	38	39	28	17
41	23	55	59	53	44	85	5	30	44	36	18
42	48	48	49	51	37	86	12	30	32	35	21
43	41	63	49	46	34	87	5	26	15	20	20
44	46	52	53	41	40	88	0	40	21	9	14

den 100 erreichbaren Punkten gab es die in Tabelle 10.1 angeführten Ergebnisse. Die Daten sind verfügbar mit `data(scor,package="bootstrap")`.

Wir möchten nun diese 5-dimensionalen Daten mittels Hauptkomponenten darstellen. In Abbildung 10.2 wurden bereits die gleichen Daten für den *scree plot* und den Plot der kumulierten Varianzanteile genommen. Man konnte daraus erkennen, dass $k = 2$ sinnvoll ist,

und dass mit 2 Hauptkomponenten etwa 80% der gesamten Variabilität wiedergegeben sind. Abbildung 10.7 zeigt nun auch die *scores* (links) und *loadings* (rechts) dieser ersten beiden Hauptkomponenten. Man erkennt, dass bei den *scores* in Richtung der 1. Hauptkomponente eine gewisse Ordnung in den Zahlen (Nummer der Studenten) vorhanden ist. Links sind die "Top-Studenten" zu finden, rechts jene mit den schlechteren Ergebnissen. Sieht man sich den *loadings* Plot an, so kann man auch diese Richtung der 1. Hauptkomponente interpretieren: Die Ladungen der 5 Variablen sind etwa gleich, und somit entspricht die Richtung der 1. Hauptkomponente einer "Durchschnittsleistung". Je kleiner der *score*, desto besser die Durchschnittsleistung. Die 2. Hauptkomponente differenziert zwischen den eher Geometrielastigen Gegenständen (positiv) und den eher analytisch-lastigen Gegenstaenden (negativ). Dies drückt sich auch durch die Position der Studenten bei den *scores* der 2. Hauptkomponente aus.

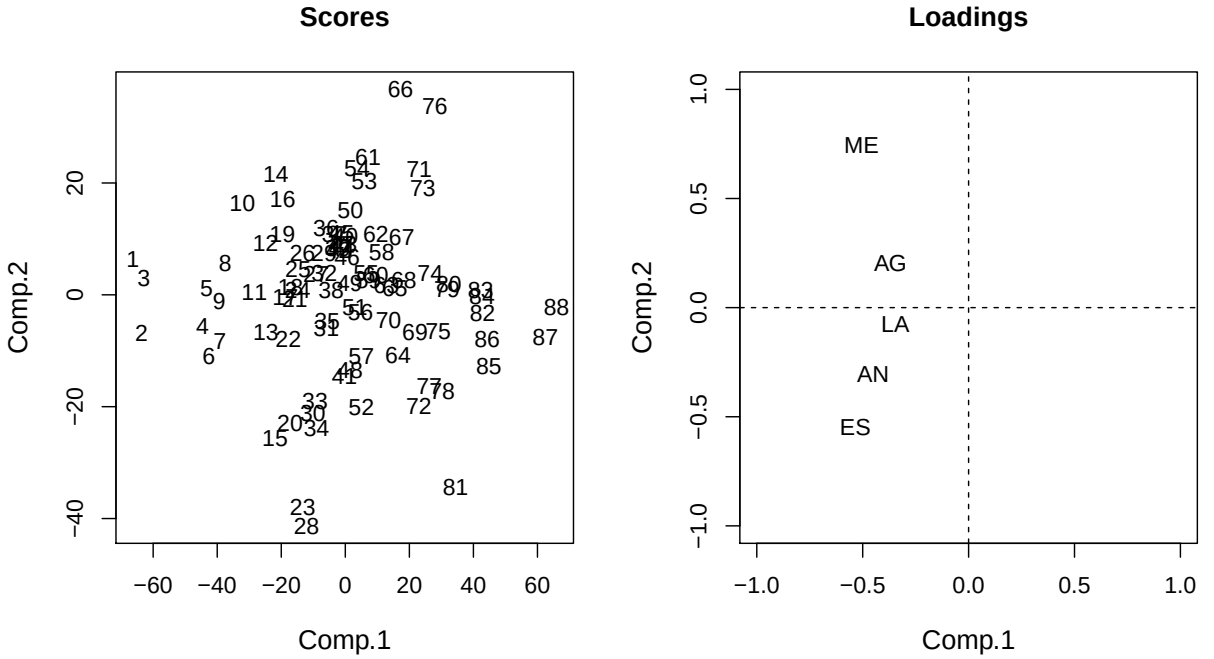


Abbildung 10.7: *Scores* (links) und *loadings* (rechts) der ersten beiden Hauptkomponenten. Die Daten erhält man mit `data(scor,package="bootstrap")`, siehe Tabelle 10.1.

Die beiden Abbildungen in 10.7 für *scores* und *loadings* können auch in einer Grafik vereint werden. Das Resultat ist der **Biplot**, der in Abbildung 10.8 gezeigt wird. Das "Bi" bezieht sich dabei nicht auf 2-dimensional, sondern darauf, dass beide, *scores* und *loadings*, gleichzeitig dargestellt werden. Man erkennt, dass die Skalierungen hier anders gewählt wurden als in Abbildung 10.7. Der Grund dafür liegt bei der Interpretation des Zusammenhangs zwischen *scores* und *loadings* in diesem Plot:

- Die orthogonalen Projektionen der Beobachtungen auf die Variablen(pfeile) approximieren die originalen (zentrierten) Datenwerte.
- Der Kosinus des Winkels zwischen den Variablen(pfeilen) approximiert die Korrelation zwischen den Variablen.
- Die euklidischen Distanzen zwischen den Beobachtungen approximieren die Mahalanobis Distanzen diesen Beobachtungen.

Wenn hier von “Approximation” gesprochen wird, so bezieht sich das auf den Informationsgehalt dieser beiden Hauptkomponenten, die hier 80% der Information wiedergeben.

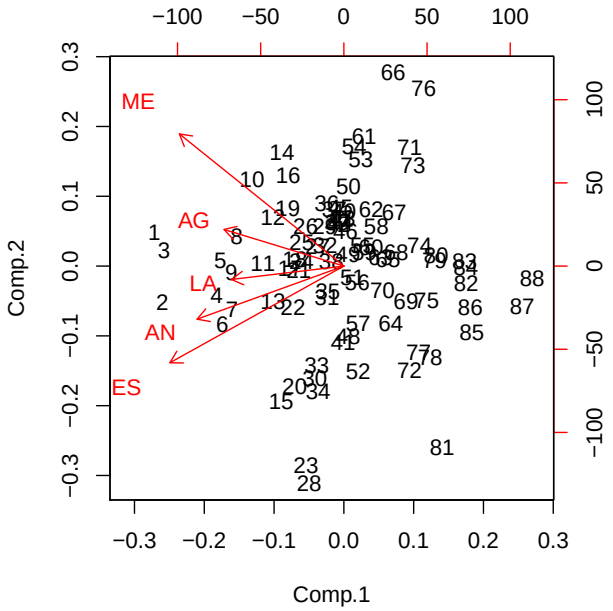


Abbildung 10.8: Biplot der ersten beiden Hauptkomponenten. Die Daten erhält man mit `data(scor,package="bootstrap")`, siehe Tabelle 10.1.

```
R:   data(scor,package="bootstrap")
R:   biplot(princomp(scor))
```

10.3 Projection Pursuit

Dieses Verfahren zielt darauf ab, Projektionen der p-dimensionalen Beobachtungen $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ auf niedrigere Dimensionen zu finden, sodass durch diese Projektionen interessante Strukturen (Nichtlinearitäten, Cluster,...) zum Ausdruck kommen. *Pursue* drückt also aus, dass interessante Projektionen “verfolgt” werden. Dies geschieht durch die Wahl eines Projektionsindex, bei dessen Extremwerten die Struktur am deutlichsten erkennbar ist.

Der Projektionsindex sollte *affin invariant* sein: Er soll die gleiche Projektion anzeigen, egal, ob die Originaldaten $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ oder transformierte Daten

$$\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{b}$$

verwendet werden. Hier ist $\boldsymbol{A}$ eine reguläre $p \times p$ Matrix, die orthonormal ist (d.h. $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$), und $\boldsymbol{b}$ ein p-dimensionalen Vektor.

Projection Pursuit Lösungen (d.h. die gefundenen Projektionsrichtungen, Projektionsebenen, ...) sind selten eindeutig, da der Projektionsindex meistens viele lokale Extremwerte besitzt. Da jede Projektion zum Erkennen der Struktur beitragen kann, sollte man den Projection Pursuit Algorithmus mit möglichst vielen Anfangswerten iterieren lassen, um zu verschiedenen lokalen Optima zu gelangen.

10.3.1 Projektionsindex

Er sollte so beschaffen sein, dass er Strukturen in den Daten erkennen lässt, die nicht durch die Korrelationsmatrix beschrieben werden. Das wird durch einen Projektionsindex erreicht, der invariant gegenüber allen nichtsingulären affinen Transformationen in $\mathbb{R}^p$ ist.

Der Index soll groß sein, wenn die Projektion interessant ist, sonst klein. Wann ist eine Projektion interessant? Man möchte Gruppen oder nichtlineares Verhalten erkennen können. Somit werden Projektionen, die einer Normalverteilung folgen, als uninteressant angesehen, weil hier weder Gruppen noch Asymmetrien sichtbar sind.

Es folgt daraus, dass der Projektionsindex so konstruiert wird, dass mit wachsenden Abweichungen von normalverteilten Projektionen auch der Wert des Index zunimmt.

10.3.2 Berechnung des Projektionsindex

Um der Forderung von affiner Invarianz nachzukommen, werden im ersten Schritt die Daten in Form von standardisierten Hauptkomponenten dargestellt. Wenn wir wieder die Notation von Zufallsvariablen nehmen, dann werden die originalen Variablen $x_1, \ldots, x_p$ als Hauptkomponenten $u_1, \ldots, u_p$ dargestellt, siehe Abschnitt 10.2.2. Die Varianzen dieser Hauptkomponenten sind $\lambda_1, \ldots, \lambda_p$. Definiert man also

$$z_j = \frac{u_j}{\sqrt{\lambda_j}} \qquad \text{für } j = 1, \ldots, p,$$

dann ist $\mathrm{Var}(z_j) = 1$, und wir haben standardisierten Hauptkomponenten.

Bemerkung: Sollten die letzten λ_j null sein, dann würden diese Komponenten einfach weggelassen werden, und wir hätten dadurch die ganze Information in einem Unterraum dargestellt.

Wir suchen zunächst nach **univariaten Projektionsrichtungen** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^T$ im p-dimensionalen Raum. Um Eindeutigkeit der Richtung zu gewährleisten, wird $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$ verlangt. Dadurch hat die Projektion

$$Y = \alpha_1 z_1 + \ldots + \alpha_p z_p$$

ebenso Varianz 1,

$$\mathrm{Var}(Y) = \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1.$$

Der Projektionsindex sollte nun einen hohen Wert erhalten, wenn die Dichte von Y stark strukturiert ist. Diese Dichte bezeichnen wir mit $p_\alpha(Y)$. Der Algorithmus für den Projektionsindex hat folgende Gestalt:

1. Transformation von Y in das Intervall $(-1, 1)$ durch

   $$R = 2\Phi(Y) - 1 \quad \text{mit } \Phi \text{ Verteilungsfunktion der Standardnormalverteilung.}$$

   Falls Y normalverteilt ist, ist R stetig gleichverteilt in $(-1, 1)$, und der Wert der Dichtefunktion in diesem Intervall ist $\frac{1}{2}$.

2. Als Maß für die Abweichung der Dichte $p_R(r)$ von R von der Gleichverteilung wird der Projektionsindex $I(\boldsymbol{\alpha})$ definiert als

   $$I(\boldsymbol{\alpha}) = \int_{-1}^{1} \left(p_R(R) - \frac{1}{2} \right)^2 dR = \int_{-1}^{1} p_R^2(R) dR - \frac{1}{2}.$$

Für die praktische Berechnung wird dieses Integral durch Polynome approximiert.

In Abbildung 10.9 ist anhand von Beispielen die Funktionsweise dieses Algorithmus illustriert. Die oberen fünf Bilder gehen davon aus, dass die Dichtefunktion der Projektion standard-normalverteilt ist. Die nächste Zeile von Abbildungen basiert auf der χ^2-Verteilung mit 2 Freiheitsgraden, und die unteren fünf Bilder zeigen, wie der Algorithmus eine Bimodalverteilung (Komposition zweier Normalverteilungen) transformiert.

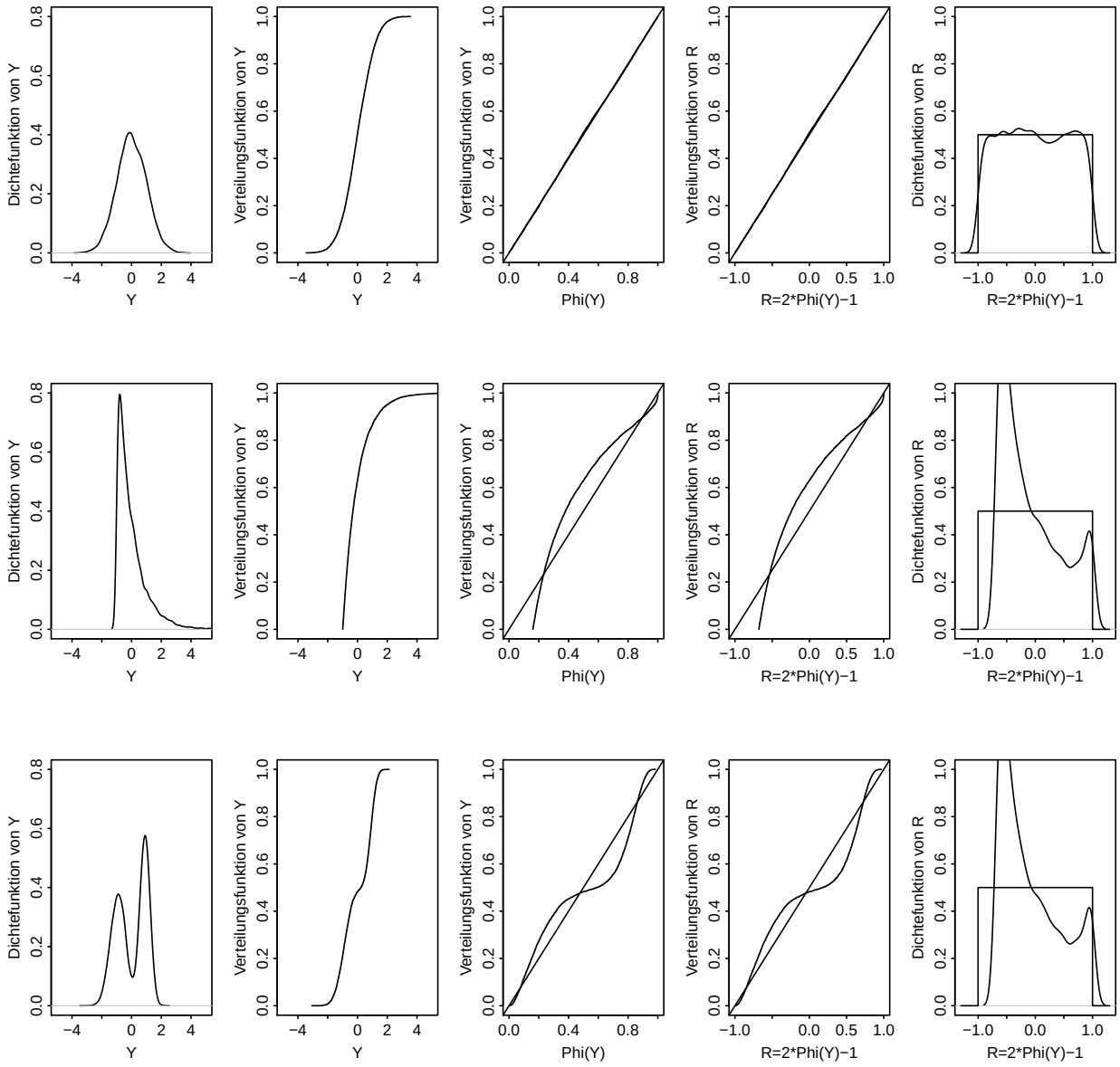


Abbildung 10.9: Illustration zur Funktionsweise des Projection Pursuit Algorithmus.

Für **2-dimensionale Projektionen** werden 2 Richtungen $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^T$ und $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ gesucht, sodass die gemeinsame Dichte von

$$Y_1 = \alpha_1 z_1 + \ldots + \alpha_p z_p$$
$$Y_2 = \beta_1 z_1 + \ldots + \beta_p z_p$$

stark strukturiert ist. Y_1 und Y_2 müssen unkorreliert sein, was gleichbedeutend mit $\boldsymbol{\alpha}^T\boldsymbol{\beta} = 0$ ist. Weiters muss $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = \boldsymbol{\beta}^T\boldsymbol{\beta} = 1$ gelten. Eine zum Vorhergehenden analoge Vorgangsweise liefert dann einen Projektionsindex $I(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Beispiel: Wir betrachten die Iris-Daten, die bekanntlich relativ starke Struktur aufweisen, da 3 Gruppen in den Daten vorhanden sind. Abbildung 10.10 vergleicht bei 1-dimensionalen Projektionen die Ergebnisse bei Verwendung von Hauptkomponenten (links) und bei Anwendung des Projection Pursuit Verfahrens (rechts). Die Lösung mit Projection Pursuit ist wesentlich besser strukturiert, die Gruppen überlappen sich weniger.

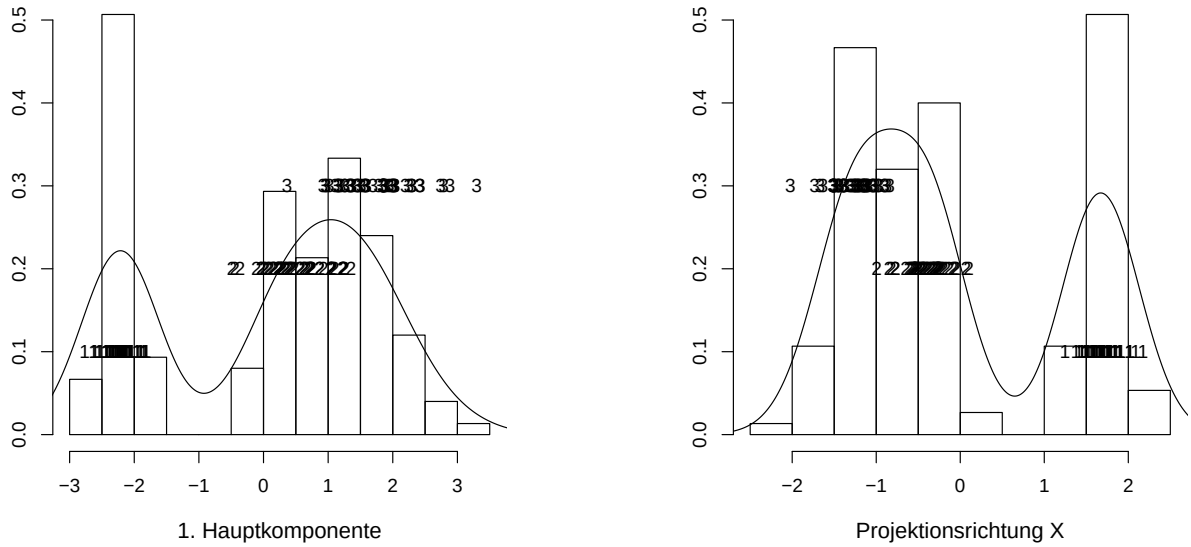


Abbildung 10.10: Iris-Daten: 1-dimensionale Projektion auf die erste Hauptkomponente (LINKS) und Projection Pursuit für eine 1-dimensionale Projektionsrichtung (RECHTS).

In Abbildung 10.11 ist die Lösung von Projection Pursuit für eine 2-dimensionale Projektion dargestellt. Die linke Grafik ist die Projektion der Daten auf die gesuchten Projektionsrichtungen, in der rechten Grafik sieht man eine Dichteschätzung der Lösung.

10.3.3 Strukturelimination

Hat man eine interessante Projektionsrichtung gefunden, kann man die Suche nach weiteren Projektionsrichtungen fortsetzen. Um jedoch zu vermeiden, dass der Projektionsindex durch die bereits gefundene Richtung beeinflusst wird, muss die Struktur, die in dieser gefundenen Richtung ja am deutlichsten zum Ausdruck kommt, eliminiert werden. Da der Projektionsindex Abweichungen von der Normalverteilung am stärksten bewertet, Normalverteilung jedoch einen Indexwert 0 bewirkt, muss versucht werden, eine Transformation zu finden, die in der bereits gefundenen Projektionsrichtung eine Normalverteilung erzeugt.

Wurde bei vorhergehender Anwendung des Projection Pursuit Verfahrens (eindimensionale Version) die Projektionsrichtung $\boldsymbol{\alpha}$ gefunden, wird die Strukturelimination folgendermaßen durchgeführt:

1. Transformiere $\boldsymbol{z} = (z_1, \ldots, z_p)^T$ durch eine orthonormale Transformation so, dass die erste Koordinate Y der transformierten Variable die Projektion in der Richtung $\boldsymbol{\alpha}$ ist (d.h. $Y = \boldsymbol{\alpha}^T\boldsymbol{z}$).

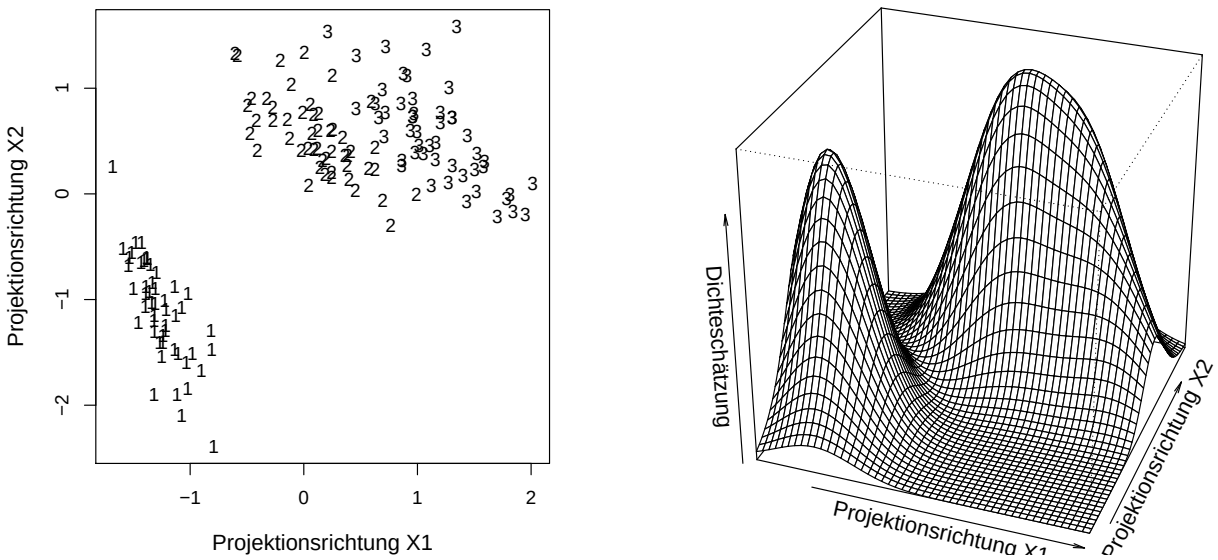


Abbildung 10.11: Iris-Daten: 2-dimensionale Projektion mit Projection Pursuit.

2. Transformiere diese erste Koordinate auf Normalverteilung durch

   $$Y' = \Phi^{-1}(F_{\boldsymbol{\alpha}}(Y)) \quad \text{mit } F_{\boldsymbol{\alpha}} \text{ Verteilungsfunktion von } Y$$

   Da man ja $F_{\boldsymbol{\alpha}}$ nicht kennt, nimmt man statt der Verteilungsfunktion die empirische Verteilungsfunktion.

3. Mache die orthonormale Transformation rückgängig. Auf die so erhaltene Variable $\boldsymbol{z}'$ wende den Projection Pursuit Algorithmus an.

Strukturelimination im 2-dimensionalen Fall ist etwas schwieriger, da man dazu eine Transformation benötigt, die eine beliebige 2-dimensionale Verteilung auf eine 2-dimensionale Normalverteilung transformiert. Theoretische Lösungen sind rechnerisch nicht durchzuführen. Es gibt jedoch eine approximative Lösung, auf die hier nicht eingegangen wird.

Kapitel 11

Weitere multivariate statistische Methoden – ein Überblick

11.1 Clusteranalyse

Der Ausdruck "Cluster" hat die Bedeutung von "konzentrierte" Gruppe. Ziel von Clusteranalyse ist es, Beobachtungen in homogene Gruppen zu unterteilen. Dabei sollen Beobachtungen mit großer Ähnlichkeit in der gleichen Gruppe (im gleichen Cluster) sein, und Beobachtungen, die einander unähnlich sind, in unterschiedliche Cluster eingeteilt werden. In speziellen Fällen ist nicht am Clustern von Beobachtungen, sondern an einer Clusterung von Variablen interessiert (siehe Boxes oder Trees bei Multivariaten Grafiken, Abschnitt 8).

Ähnlichkeiten von Beobachtungen werden über Distanzen bestimmt. Hier können alle Distanzmaße von Abschnitt 9.2 herangezogen werden. *Achtung:* Nachdem Distanzen von der Skalierung der Variablen abhängen, wird es bei den meisten Clusterverfahren wichtig sein, zuerst die Variablen auf Mittel 0 und Varianz 1 zu standardisieren, um eine Vergleichbarkeit der Variablen zu gewährleisten.

Die "wirkliche" Anzahl k von Clustern in multivariaten Daten ist normalerweise unbekannt, und sie muss durch später zu wählende Kriterien geschätzt werden. Meist liegen auch keine klar getrennten Gruppen in den Daten vor, was die Schätzung der Anzahl von Clustern sowie die Zuordnung von Beobachtungen zu den Clustern zusätzlich erschwert. Grundsätzlich gibt es mehrere Vorgangsweisen für die Konstruktion von Clustern:

- **Partitionierungsmethoden:** Jede Beobachtung wird genau einem Cluster zugeordnet. Man erhält somit eine disjunkte Gruppierung der Beobachtungen in eine Anzahl k von Clustern.
- **Hierarchische Clustermethoden:** Es wird eine Hierarchie von Partitionierungen konstruiert, in der die Anzahl der Cluster von 1 bis n (= Anzahl der Beobachtungen) variiert. Dies erlaubt eine Übersicht über verschiedene Anzahlen von Gruppierungen, und diese Übersicht wird mittels *Dendrogramm* visualisiert. Hierarchien können agglomerativ (von einer n-Cluster bis zu einer 1-Cluster Partitionierung) oder divisiv (ein Cluster wird schrittweise aufgespalten, bis eine n-Cluster Lösung entsteht) konstruiert werden, wobei agglomerativ am gebräuchlichsten ist.
- **Fuzzy Clustering:** Jede Beobachtung wird jedem Cluster über einen Zugehörigkeitskoeffizienten zugeteilt. Die Koeffizienten sind aus dem Intervall $[0, 1]$, und die Summe der Koeffizienten für eine Beobachtung über alle Cluster ergibt 1.

- **Modellbasierte Clusterung:** Hier handelt es sich um eine Partitionierungsmethode, wobei aber die Form eines Clusters durch eine Modellverteilung parametrisiert wird. Die typische Modellverteilung ist eine multivariate Normalverteilung mit einem bestimmten Erwartungswert und einer gewissen Kovarianz.

Wir gehen wieder von einer $n \times p$ Datenmatrix $\boldsymbol{X}$ aus, und möchten die Beobachtungen $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ clustern.

11.1.1 Partitionierungsmethoden

Sei k die Anzahl von Clustern, in die die Beobachtungen disjunkt unterteilt werden sollen. Sei weiters I_j eine Indexmenge, die die Indizes der Beobachtungen des j-ten Clusters enthält, und n_j die Anzahl der Elemente in I_j ($j = 1, \ldots, k$). Die Indexmenge hat also die Form $I_j = \{i_1, i_2, \ldots, i_j\}$, und demnach enthält das j-te Cluster die Beobachtungen $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \ldots, \boldsymbol{x}_{i_j}$. Es gilt bei Partitionierungen $n_1 + n_2 + \ldots + n_k = n$.

Der bekannteste Algorithmus zur Konstruktion einer Partitionierung ist der **k-means** Algorithmus. Als Input-Parameter benötigt er die gewünschte Anzahl k von Clustern. k-means verwendet sogenannte Zentroide oder Cluster-Zentren, die einfach als arithmetische Mittel der Beobachtungen eines Clusters genommen werden können:

$$\bar{\boldsymbol{x}}_j = \frac{1}{n_j} \sum_{i \in I_j} \boldsymbol{x}_i \quad \text{für} \quad j = 1, \ldots, k \tag{11.1}$$

Die zu minimierende Zielfunktion bei k-means ist

$$\sum_{j=1}^{k} n_j \sum_{i \in I_j} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j\|^2 \longrightarrow \min . \tag{11.2}$$

Man verwendet zur Minimierung einen iterativen Algorithmus, wobei sich in jedem Schritt die Indexmengen I_j mehr oder weniger stark verändern werden. Nach Konvergenz erhält man die endgültigen Indexmegen $I_1^*, \ldots, I_k^*$, und damit die endgültige Zuteilung der n Beobachtungen zu den erwünschten k Clustern. Der iterative Algorithmus wird meist initialisiert über eine zufällige Auswahl von k Beobachtungen, die anfangs die k Cluster-Zentren bilden. Je nach Zufallsauswahl kann man daher zu unterschiedlichen Cluster-Lösungen kommen!

```
# R code k-means
Xs <- scale(X)          # X ist die Datenmatrix, die hier standardisiert wird
res <- kmeans(Xs,3)     # k=3 ist die Anzahl von Clustern
str(res)                # zeigt die Inhalte des Output-Objektes
res$cluster             # zeigt die Zuteilung der Objekte zu den Clustern
```

Ein sehr ähnlicher Algorithmus zu k-means ist **PAM** (Partitioning Around Medoids), nur werden hier die Cluster-Zentroide robust über Mediane bestimmt.

```
# R code PAM
Xs <- scale(X)          # X ist die Datenmatrix, die hier standardisiert wird
library(cluster)
res <- pam(Xs,3)        # k=3 ist die Anzahl von Clustern
str(res)                # zeigt die Inhalte des Output-Objektes
res$clustering          # zeigt die Zuteilung der Objekte zu den Clustern
plot(res)               # Diagnostik-Plots
```

Eine Entscheidung über eine "sinnvolle" Anzahl k von Clustern kann eigentlich erst mit einem Gütemaß getroffen werden, das später definiert wird. Man kann k-means oder PAM dann mit unterschiedlichen Werten von k ausführen, und aufgrund des Gütemaßes die beste Lösung wählen. Dies ist auch hilfreich bei unterschiedlichen Lösungen für gleiches k.

11.1.2 Hierarchische Clustermethoden

Wir beschreiben hier die agglomerative Vorgangsweise zur Konstruktion einer Hierarchie von Partitionen. Zu Beginn bildet jede der n Beobachtungen ein eigenes Cluster - solche Cluster werden als *Singletons* bezeichnet. Dann werden jene Singletons mit kleinster Distanz in ein Cluster vereint, und man erhält $n-1$ Cluster. Sollen dann nicht mehr Singletons sondern größere Cluster vereint werden, dann benötigt man eine Definition der Distanz zwischen Clustern. Seien zwei unterschiedliche Cluster gegeben durch die Indexmengen I_j und $I_{j'}$. Es sind folgende Definitionen gebräuchlich, die auch den Namen der entsprechenden Algorithmen bezeichnen:

- **Complete linkage:** $\max_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Single linkage:** $\min_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Average linkage:** $\text{average}_{i\in I_j, i'\in I_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
- **Centroid Methode:** $d(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_{j'})$
- **Ward Methode:** $d(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_{j'}) \frac{\sqrt{2n_j n_{j'}}}{\sqrt{n_j + n_{j'}}}$

Als Distanz $d(\cdot,\cdot)$ kann hier ein Distanzmaß von Abschnitt 9.2 genommen werden, z.B. die euklidische Distanz. Die unterschiedlichen Algorithmen bewirken auch meist eine unterschiedliche Clusterung. Z.B. werden mit *single linkage* typischerweise kettenförmige Cluster gebildet.

Abbildung 11.1 illustriert für zwei Cluster (durch Ellipsen eingezeichnet) die Distanzen *complete* und *single linkage*. Es werden also in einem Schritt des hierarchischen Clusterns jene beiden Cluster mit kleinster Distanz zueinander verbunden.

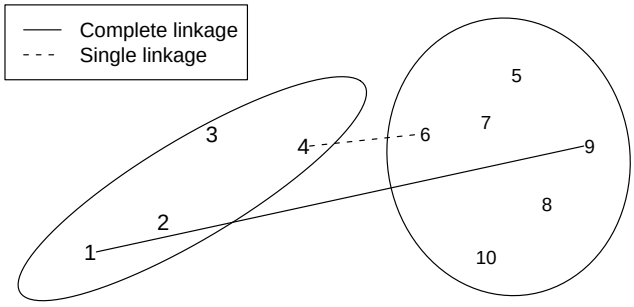


Abbildung 11.1: Illustration der *complete* und *single linkage* Distanz (hier basierend auf euklidischer Distanz) für zwei Cluster.

Für die Beispieldaten in Abbildung 11.1 ist nun in Abbildung 11.2 die ganze Hierarchie der Cluster dargestellt, wie sie mittels *complete linkage* erzeugt wird. Im Gegensatz zu k-means gibt es hier keine Zufälligkeit – das Ergebnis ist immer eindeutig aufgrund des gegebenen Minimierungsproblems.

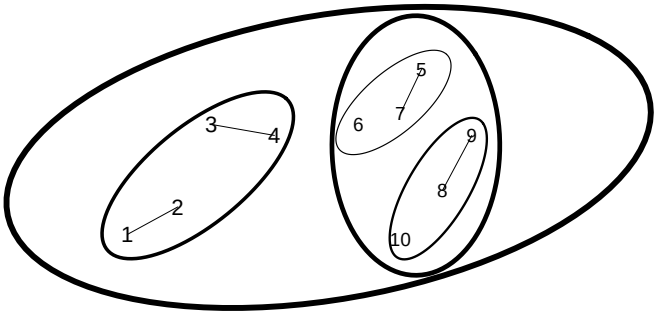


Abbildung 11.2: Ergebnis von *complete linkage* auf die Beispieldaten von Abbildung 11.1. Zunächst bildet jede Beobachtung ein eigenes Cluster. Diese werden dann schrittweise vereint, bis zuletzt alle Beobachtungen in einem einzigen Cluster liegen.

Das Ergebnis in Abbildung 11.2 kann besser dargestellt werden mit dem **Dendrogramm**. Dabei ist in vertikaler Richtung die Distanz aufgetragen (z.B. *complete linkage*), die schrittweise wächst. Horizontale Linien bedeuten, dass bei der entsprechenden Distanz die jeweiligen Cluster verbunden werden. Die Beobachtungen werden dabei so angeordnet, dass keine Überschneidungen bei den Linien entstehen. Das Dendrogramm ist also von unten nach oben zu lesen.

Abbildung 11.3 zeigt links das Dendrogramm von *complete linkage*, und rechts jenes von *single linkage*. Während *complete linkage* sehr klar zwei Cluster zeigt (diese würden erst bei sehr großer Distanz vereint werden), ist das bei *single linkage* nicht so offensichtlich. Diese Darstellung kann also dazu genützt werden, eine "sinnvolle" Anzahl k von Clustern in den Daten zu identifizieren. Mit einem horizontalen Schnitt bei der entsprechenden Distanz kann man die Zuordnung der Beobachtungen zu den k Clustern erhalten.

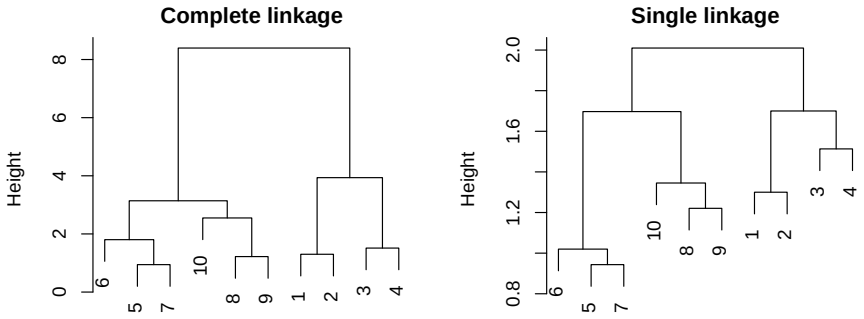


Abbildung 11.3: Dendrogramm für die Beispieldaten aus Abbildung 11.1. Links die Lösung für *complete linkage*, siehe auch Abbildung 11.2, rechts jene für *single linkage*.

```
# R code hierarchische Clusterung
Xs <- scale(X)           # X ist die Datenmatrix, die hier standardisiert wird
res <- hclust(dist(Xs))  # default ist complete linkage und euklidische Distanz
plot(res)                # zeigt das Dendrogramm
cl <- cutree(res,3)      # liefert die Zuteilung der Objekte zu 3 Clustern
```

11.1.3 Fuzzy Clustering

Anstelle von "harter" Zuteilung der Beobachtungen zu den Clustern wird hier eine "weiche" (fuzzy) Zuteilung gemacht. Bei der Partitionierung wird somit jede der n Beobachtungen jedem der k Cluster zugeteilt. Dies geschieht über einen Zugehörigkeitskoeffizienten u_{ij}, für $i = 1, \ldots, n$ und $j = 1, \ldots, k$, der im Intervall $[0, 1]$ liegt. Man verlangt außerdem, dass $\sum_{j=1}^{k} u_{ij} = 1$ ist, für alle i, und kann somit die Koeffizienten als anteilsmäßige Aufteilung einer Beobachtung auf die Cluster interpretieren.

Der Clusteralgorithmus muss somit die Matrix mit den Koeffizienten u_{ij} schätzen. Die Anzahl der Cluster k wird analog zu k-means vom Anwender vorgegeben. Man kann vom Ergebnis von fuzzy clustering immer eine "harte" Zuteilung machen, indem eine Beobachtung jenem Cluster zugeteilt wird, für das der Zugehörigkeitskoeffizient am höchsten ist.

Der gebräuchlichste Algorithmus ist der *fuzzy c-means* Algorithmus. Die Zielfunktion ist ähnlich zu k-means:

$$\sum_{j=1}^{k} \sum_{i=1}^{n_j} u_{ij}^2 \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_j\|^2 \longrightarrow \min, \tag{11.3}$$

mit den Cluster-Zentroiden

$$\tilde{\boldsymbol{x}}_j = \frac{\sum_{i=1}^{n_j} u_{ij}^2 \boldsymbol{x}_i}{\sum_{i=1}^{n_j} u_{ij}^2} \quad \text{für} \quad j = 1, \ldots, k. \tag{11.4}$$

Wiederum wird das Problem über einen iterativen Algorithmus gelöst, wobei in jedem Schritt die Koeffizienten u_{ij} neu geschätzt werden. Ähnlich zu k-means wir der Algorithmus zufällig initialisiert, was zu unterschiedlichen Ergebnissen führen kann.

```
# R code fuzzy clustering
Xs <- scale(X)           # X ist die Datenmatrix, die hier standardisiert wird
library(e1071)
res <- cmeans(Xs,3)      # fuzzy c-means mit 3 Clustern
str(res)                 # zeigt desn Inhalt des Output-Objektes
res$cluster              # harte Cluster-Zuteilung
res$membership           # Matrix der Zugehoerigkeitskoeffizienten
```

11.1.4 Modellbasierte Clusterung

Die Form der Cluster wird hier "modelliert". Meist nimmt man als Modell an, dass das j-te Cluster aus einer p-dimensionalen Normalverteilung mit Mittel $\boldsymbol{\mu}_j$ und Kovarianz $\boldsymbol{\Sigma}_j$ kommt. Der Cluster-Algorithmus muss dann nicht nur die Zugehörigkeiten der Beobachtungen zu den Clustern (Partitionierung) ermitteln, sondern auch die Parameter $\boldsymbol{\mu}_j$ und $\boldsymbol{\Sigma}_j$ schätzen. Speziell die Schätzung der $p \times p$ Matrizen $\boldsymbol{\Sigma}_j$, für $j = 1, \ldots, k$, ist aufwändig und erfordert viel Information (Daten). Man ist daher interessiert an einer Vereinfachung, indem man z.B. annimmt, dass die Kovarianzen der Cluster alle gleich sind und vielleicht sogar noch sehr einfache Struktur haben.

Die einfachste Variante wäre $\boldsymbol{\Sigma}_j = \sigma^2 \boldsymbol{I}$ für $j = 1, \ldots, k$. Hier ist $\boldsymbol{I}$ die Einheitsmatrix und σ^2 ein Parameter für die Varianz. Somit wären alle Cluster kreisförmig, mit gleichem Radius in allen Dimensionen. Eine weniger strenge Annahme wäre $\boldsymbol{\Sigma}_j = \sigma_j^2 \boldsymbol{I}$ für $j = 1, \ldots, k$. Die Cluster wären nach wie vor kreisförmig, hätten aber unterschiedliche Größe, entsprechend der Varianz σ_j^2. Abbildung 11.4 illustriert unterschiedliche Varianten.

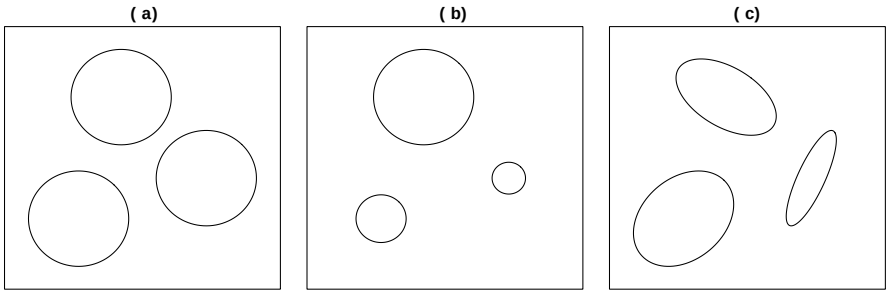


Abbildung 11.4: Unterschiedliche Kovarianzen für die drei Cluster: (a) $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}_3 = \sigma^2\boldsymbol{I}$; (b) $\mathbf{\Sigma}_j = \sigma_j^2\boldsymbol{I}$, für $j = 1, 2, 3$; (c) alle $\mathbf{\Sigma}_j$ unterschiedlich und von keiner speziellen Struktur.

Ein Vorteil von modellbasierter Clusteranalyse ist, dass die Daten nicht vorher skaliert werden müssen (weil eben im Modell unterschiedliche Varianzen verwendet werden können).

Im folgenden Beispiel werden mit dem Algorithmus `Mclust()` vom package `mclust` die Iris-Daten geclustert. Bei diesem Algorithmus kann man für die Anzahl von Cluster einen Bereich angeben, hier von 3 bis 9:

```
# R code modellbasierte Clusterung
data(iris)                      # iris Daten, nur Spalten 1-4 nehmen!
library(mclust)
res <- Mclust(iris[,1:4],3:9)  # Loesungen fuer 3 bis 9 Cluster
plot(res)                       # diverse Diagnostik Plots
str(res)                        # Inhalte des Output-Objektes
res$classification              # Cluster-Zuteilung nach bester Loesung
```

Ergebnisse werden in Abbildung 11.5 gezeigt. In der linken Abbildung sind die BIC (Bayesian Information Criterion) Werte für die Cluster-Lösungen mit unterschiedlichen Anzahlen von Clustern (Number of Components) gezeigt. Die Linien entsprechen verschiedenen Cluster-Modellen, siehe Legende. Modell "EII" ist das einfachste Modell, mit Kovarianzen $\mathbf{\Sigma}_j = \sigma^2\boldsymbol{I}$. Die Modelle werden dann immer komplexer, bis hin zu "VVV", was individuellen Kovarianzen $\mathbf{\Sigma}_j$ für die einzelnen Cluster entspricht. Das beste Modell und die beste Anzahl von Clustern ist dann erreicht bei maximalem BIC Wert, in diesem Fall für $k = 3$ Cluster und Modell "VEV".

Die rechte Abbildung in 11.5 zeigt einen Scatter-Plot der Daten mit den Zuordnungen der Beobachtungen zu den $k = 3$ Clustern, sowie mit Ellipsen eingezeichnet die Struktur der Kovarianzen.

11.1.5 Gütemaße

Ein Gütemaß soll die Entscheidung für eine gute Wahl der Anzahl k von Clustern unterstützen, es kann aber auch für die Auswahl eines geeigneten Cluster-Algorithmus herangezogen werden. Leider gibt es aber mindestens so viele unterschiedliche Gütemaße wie Cluster-Algorithmen. Das BIC Kriterium von modellbasierter Clusterung ist eine Möglichkeit.

Ziel einer Clusteranalyse ist es, dass Beobachtungen innerhalb eines Clusters ähnlich zueinander sind (homogen), und Beobachtungen verschiedener Cluster möglichst unterschiedlich sind (heterogen). Maße für *Homogenität* können basieren auf der maximalen, minimalen, oder durchschnittlichen Distanz aller Beobachtungen eines Clusters, oder an der Streuung

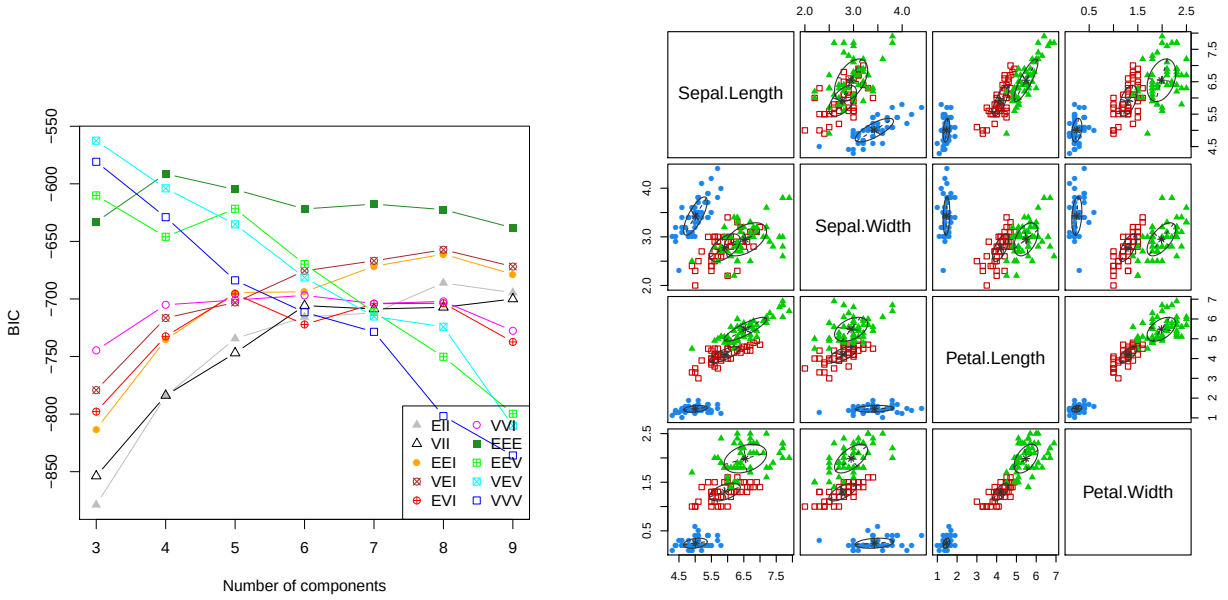


Abbildung 11.5: Ergebnisse von modellbasierter Clusterung mit dem Algorithmus `Mclust()`: links für die gewünschten Anzahlen von Clustern (horizontale Achse) die BIC-Werte für die unterschiedlichen Cluster-Modelle; rechts die Lösung für das laut BIC optimale Modell.

(Varianz) der Beobachtungen eines Clusters. Letzteres wird z.B. durch die *Within-Cluster Sum-of-Squares* ausgedrückt,

$$W_k = \sum_{j=1}^{k} \sum_{i \in I_j} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j\|^2, \tag{11.5}$$

siehe auch Formeln (11.1) und (11.2). Der Wert von W_k sollte möglichst klein werden, aber dies hängt natürlich auch von k ab; je höher k gewählt wird, desto kleiner kann die Streuung der Punkte innerhalb eines Clusters werden.

Andererseits kann man die *Heterogenität* über die Distanzmaße von *complete linkage*, *single linkage*, etc., definieren, oder durch die *Between-Cluster Sum-of-Squares*

$$B_k = \sum_{j=1}^{k} \|\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}\|^2, \quad \text{mit} \quad \bar{\boldsymbol{x}} = \frac{1}{k} \sum_{j=1}^{k} \bar{\boldsymbol{x}}_j \tag{11.6}$$

ausdrücken. Der Wert von B_k sollte möglichst groß werden, was aber wiederum von k abhängt.

Der **Calinski-Harabasz-Index** ist ein normalisiertes Verhältnis der beiden Größen,

$$\mathrm{CH}_k = \frac{B_k/(k-1)}{W_k/(n-k)}.$$

Der **Hartigan-Index** ist definiert als

$$\mathrm{H}_k = \log \frac{B_k}{W_k}.$$

Die optimale Anzahl von Clustern wird für den kleinsten Wert eines dieser Indizes über alle betrachteten k gewählt.

11.2 Diskriminanzanalyse

Diese multivariate Methode hat – wie Clusteranalyse – das Ziel, Daten zu gruppieren. Es gibt aber einen essentiellen Unterschied zu Clusteranalyse: Bei Clusteranalyse ist nicht bekannt, welche Beobachtungen zu welchen Gruppen gehören; es ist nicht einmal klar, wie viele Gruppen in den Daten existieren, und ob überhaupt eine Gruppenstruktur vorliegt. Bei Diskriminanzanalyse hingegen kennt man die Klassenzugehörigkeiten der Beobachtungen, zumindest jener Beobachtungen von einem Trainingsdatensatz. Man weißz.B., dass gewisse Messungen von erkrankten und von gesunden Personen kommen, und kennt somit sowohl die Anzahl der Gruppen, als auch die Gruppenzugehörigkeit. Nun möchte man eine Regel lernen, die es erlaubt, aufgrund der gleichen gemessenen Merkmale (Variablen) neue Beobachtungen den Klassen zuzuordnen. Diese neuen Beobachtungen bilden den sogenannten Testdatensatz.

Gegeben seien die Trainingsdaten $\boldsymbol{X}$, eine $n \times p$ Datenmatrix mit den Beobachtungen $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$. Die Beobachtungen stammen aus k Gruppen, die Gruppenzugehörigkeiten sind bekannt, und die Anzahlen der Beobachtungen in den Gruppen sind $n_1, \ldots, n_k$, mit $n_1 + \ldots + n_k = n$. Die Variable G beschreibt die prognostizierte Gruppenzugehörigkeit, und kann somit einen Wert aus der Menge $\{1, \ldots, k\}$ annehmen.

Weiters wird angenommen, dass die zugrunde liegende Verteilung der Gruppen bekannt ist. Hier nehmen wir multivariate Normalverteilung mit Parametern $\boldsymbol{\mu}_j$ und $\boldsymbol{\Sigma}_j$ an, für $j = 1, \ldots, k$; die Dichtefunktion ist

$$\boldsymbol{\phi}_j(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma}_j)}} \exp\left\{ -\frac{(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)}{2} \right\}. \tag{11.7}$$

Schließlich nehmen wir auch noch eine *a-priori* Wahrscheinlichkeit p_j für jede Gruppe an, wobei $p_1 + \ldots + p_k = 1$ ist. Beschreibt z.B. Gruppe 1 erkrankte Personen und Gruppe 2 gesunde, so ist typischerweise $p_1 < p_2$, weil es (hoffentlich) weniger wahrscheinlich ist, diese Erkrankung in der Population anzutreffen.

Mit diesen Angaben kann nun die bedingte (*a-posteriori*) Wahrscheinlichkeit laut Bayes'schem Theorem ermittelt werden: Bei gegebener Beobachtung $\boldsymbol{x}$ ist die bedingte Wahrscheinlichkeit, dass die Variable G den Wert j annimmt, gegeben durch

$$\mathrm{P}(G = j|\boldsymbol{x}) = \frac{\boldsymbol{\phi}_j(\boldsymbol{x}) p_j}{\sum_{l=1}^k \boldsymbol{\phi}_l(\boldsymbol{x}) p_l}. \tag{11.8}$$

Der Nenner in Gleichung (11.8) ist für jede Gruppe gleich, und daher kann man direkt die *a-posteriori* Wahrscheinlichkeiten für zwei Gruppen vergleichen. Ist $\mathrm{P}(G = j|\boldsymbol{x}) > \mathrm{P}(G = l|\boldsymbol{x})$, so würde $\boldsymbol{x}$ der j-ten Gruppe zugeteilt werden. Anders ausgedrückt: $\boldsymbol{x}$ wird der j-ten Gruppe (und nicht der l-ten Gruppe) zugeteilt, wenn gilt

$$\log \frac{\mathrm{P}(G = j|\boldsymbol{x})}{\mathrm{P}(G = l|\boldsymbol{x})} = \log \frac{\boldsymbol{\phi}_j(\boldsymbol{x}) p_j}{\boldsymbol{\phi}_l(\boldsymbol{x}) p_l} = \log \frac{\boldsymbol{\phi}_j(\boldsymbol{x})}{\boldsymbol{\phi}_l(\boldsymbol{x})} + \log \frac{p_j}{p_l} > 0. \tag{11.9}$$

11.2.1 Lineare Diskriminanzanalyse (LDA)

Wir treffen eine weitere Annahme, nämlich dass $\boldsymbol{\Sigma}_1 = \ldots = \boldsymbol{\Sigma}_k$ gilt. Die Kovarianzen aller Gruppen werden somit als gleich angenommen, und wir bezeichnen diese Kovarianz jetzt mit $\boldsymbol{\Sigma}$. Man kann nun Gleichung (11.7) einsetzen in (11.9) und erhält schließlich die **lineare**

Diskriminanz-Regel:
$\boldsymbol{x}$ wird der j-ten Gruppe (und nicht der l-ten Gruppe) zugeteilt, wenn $\delta_j(\boldsymbol{x}) > \delta_l(\boldsymbol{x})$ ist, wobei

$$\delta_j(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j + \log p_j \tag{11.10}$$

die sogenannte *linearen Diskriminanzfunktion* der j-ten Gruppe darstellt.

Sind nun $k > 2$ Gruppen gegeben, dann berechnet man für eine neue Testbeobachtung $\boldsymbol{x}$ die lineare Diskriminanzfunktion für jede Gruppe, und $\boldsymbol{x}$ wird dann jener Gruppe zugeteilt, für die der Wert der linearen Diskriminanzfunktion am größten ist.

Man erkennt gleich, dass die resultierenden Entscheidungsregeln *linear* sind, weil die Diskriminanzfunktion in (11.10) linear in $\boldsymbol{x}$ ist.

Praktisch müssen die Parameter in Gleichung (11.10) zuerst aus den Trainingsdaten geschätzt werden. Als Schätzung für p_j kann man n_j/n nehmen, sofern die Trainingsdaten die Grundgesamtheit widerspiegeln. Sei I_j die Indexmenge für die Beobachtungen der j-ten Gruppe der Trainingsdaten (vgl. Clusteranalyse). Als Schätzung für $\boldsymbol{\mu}_j$ bietet sich das arithmetische Mittel (Vektor!) der Trainingsdaten der j-ten Gruppe an,

$$\hat{\boldsymbol{\mu}}_j = \bar{\boldsymbol{x}}_j = \frac{1}{n_j} \sum_{i \in I_j} \boldsymbol{x}_i \quad \text{für } j = 1, \ldots, k, \tag{11.11}$$

siehe auch Gleichung (11.1). Die gemeinsame Kovarianzmatrix kann durch eine "gepoolte" Kovarianz geschätzt werden,

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{S}_{pooled} = \frac{1}{n-k} \sum_{j=1}^{k} \sum_{i \in I_j} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T. \tag{11.12}$$

11.2.2 Quadratische Diskriminanzanalyse (QDA)

Wenn man nicht die Gleichheit der Kovarianzen voraussetzen kann (möchte), dann wird beim Einsetzen von Gleichung (11.7) in (11.9) der resultierende Ausdruck komplizierter. Man erhält dann die sogenannten *quadratischen Diskriminanzfunktionen*

$$\delta_j^{(q)}(\boldsymbol{x}) = -\frac{1}{2} \log(\det(\boldsymbol{\Sigma}_j)) - \frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) + \log p_j \tag{11.13}$$

für $j = 1, \ldots, k$, die quadratisch in $\boldsymbol{x}$ sind. Die neue Beobachtung $\boldsymbol{x}$ wird dann jener Gruppe zugeteilt, für die der Wert der quadratischen Diskriminanzfunktion am größten ist.

Wiederum müssen für die praktische Anwendung der Regel zuerst die Parameter aus den Trainingsdaten geschätzt werden. Im Gegensatz zur LDA braucht man jetzt keine gemeinsame Schätzung der Kovarianz, sondern individuelle Schätzungen, z.B. durch die Stichproben-Kovarianzen

$$\hat{\boldsymbol{\Sigma}}_j = \boldsymbol{S}_j = \frac{1}{n_j - 1} \sum_{i \in I_j} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T, \tag{11.14}$$

aber auch robuste Schätzungen mittels MCD-Schätzer wären denkbar. Man bemerke, dass man im Vergleich zu LDA hier wesentlich mehr Parameter schätzen muss, was eine "Überanpassung" an die Trainingsdaten bewirken und zu einer schlechteren Prognose der Gruppenzugehörigekeit führen kann.

Als Beispiel betrachten wir wieder die Iris-Daten. Im Gegensatz zur Clusteranalyse wird diesmal die Gruppeninformation für die Trainingsdaten verwendet zum Erstellen der Diskriminanzfunktionen, und anschließend zur Evaluierung der Methode anhand der Testdaten herangezogen. LDA kann folgendermaßen durchgeführt werden:

```
# R code LDA (QDA)
data(iris)                          # iris Daten
X <- iris[,1:4]                     # Messungen
grp <- iris[,5]                     # Gruppe
grpn <- as.numeric(grp)             # Gruppennummern
set.seed(123)                       # bewirkt immer gleiche Zufallsauswahl
n <- nrow(iris)                     # Anzahl der Beobachtungen
train <- sample(n,round(n*2/3)) # Indizes Trainingsdaten
test <- (1:n)[-train]               # Indizes Testdaten
library(MASS)
res <- lda(X[train,],grp[train])   # LDA auf Trainingsdaten
### fuer QDA einfach den Befehl qda() nehmen
res.pred <- predict(res,X[test,])  # Prognose fuer Testdaten
table(grp[test],res.pred$class)    # Vergleich mit Wirklichkeit
#              setosa versicolor virginica
#  setosa          14          0         0
#  versicolor       0         17         2
#  virginica        0          0        17
#
plot(res,abbrev=2,col=grpn[train])       # lineare Diskriminanzfunktionen
points(res.pred$x,col=grpn[test],pch=as.numeric(res.pred$class)) # Testdaten
```

Abbildung 11.6 zeigt eine Projektion der Daten in den Raum der linearen Diskriminanzfunktionen. Die Symbole mit den Texten sind die Trainingsdaten, und die Farbe entspricht deren Gruppenzugehörigkeit. Die Testdaten werden durch nicht-Textsymbole dargestellt; die Farbe ist die wirkliche Gruppe, das Symbol die prognostizierte Gruppe. Man erkennt bei zwei roten Beobachtungen ein falsches Symbol.

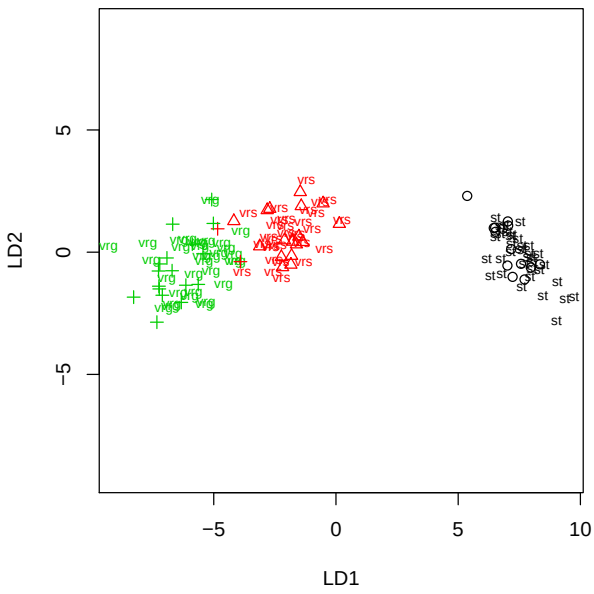


Abbildung 11.6: Ergebnisse von LDA auf die Iris-Daten.