Kapitel 2

Statistische Graphik und deskriptive Statistik

2.1 Vorbereitung

Datentafeln und Datenmatrizen sind sehr unübersichtlich. Wir wollen unsere Daten zunächst graphisch darstellen. Dazu müssen wir das Folgende lernen:

- Welche statistischen Graphiken gibt es?
- Welche Graphik eignet sich für welche Daten?
- Welche Graphik eignet sich für welche Fragestellung?
- Wie erzeugt man die Graphik mit "R"?
- Wie wird die Information in der Graphik dargestellt?
- Welche Artefakte produziert die Graphik?
- Welche Einstellungsmöglichkeiten gibt es und wie sind sie zu wählen?
- Wie interpretiert man die Graphik richtig?

Dazu verwenden wir zunächst einen sehr einfachen Datensatz.

2.1.0.0.2 R: [Laden der Beispieldaten]

Quelle: Statlib

Story Names Acorn Size Oak Distribution

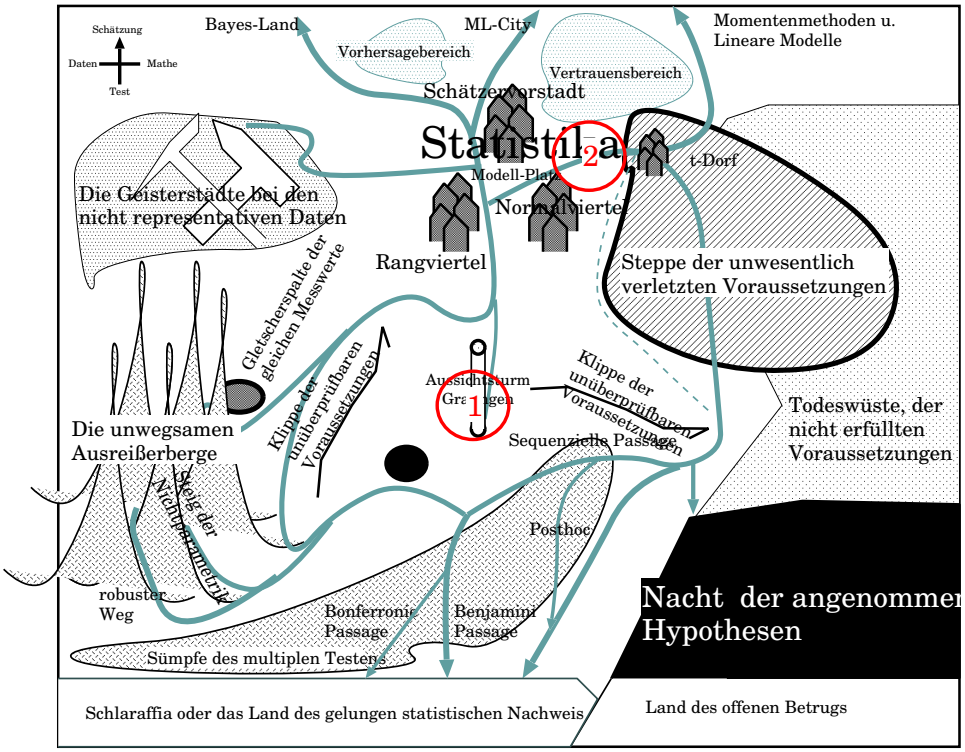
Reference Aizen and Patterson. (1990). *Journal of Biogeography*, volume 17, p. 327-332. <b>

Authorization contact authors

Description Interest lies is the relationship between the size of the acorn and the geographic range of the oak tree species. Note that the *Quercus tomentella Engelm* species in the California region grows only on the Channel Islands (total area 1014 sq. km) and the island of Guadalupe (total area 265 sq. km). All other species grow on the Continental United States.

Number of cases 39

Variable Names
1. Species: Latin name of the species
2. Region: Atlantic or California region
3. Range: The geographic area covered by the species in $km^2 \times 100$



Kapitel 2 setzt sich mit den beschreibenden, noch modellfreien Methoden der Statistik auseinander.

1 Die statistische Graphik erlaubt es sich einen Überblick über die Daten und das weitere Vorgehen zu verschaffen. Deshalb werden Sie durch einen Aussichtsturm symbolisiert. Sie beschreiben die qualitativen Merkmale der Daten.

2 Deskriptive Statistiken (also aus den Daten ausgerechneten Kennzahlen) fassen die quantitativen Merkmale der Daten zusammen. Sie sind das wichtigste Instrument für den normalverteilungsbasierten Teil der Statistik, der auf der nach rechts gehenden Hauptstraße dargestellt wird.

Abbildung 2.1: Übersichtskarte Kapitel 2
Die Einordnung von Kapitel 2 in die Übersichtskarte.

4. Acorn.size: Acorn size (cm^3)
5. Tree.height: Tree Height (m)

Data

```
Species                    Region         Range Acorn.size Tree.Height
Quercus alba L.            Atlantic       24196   1.4     27
Quercus bicolor Willd.     Atlantic       7900    3.4     21
Quercus macrocarpa Michx.  Atlantic       23038   9.1     25
Quercus prinoides Willd.   Atlantic       17042   1.6     3
Quercus Prinus L.          Atlantic       7646    10.5    24
Quercus stellata Wang.     Atlantic       19938   2.5     17
Quercus virginiana Mill    Atlantic       7985    0.9     15
Quercus Michauxii Nutt.    Atlantic       8897    6.8     .30
Quercus lyrata Walt.       Atlantic       8982    1.8     24
Quercus Laceyi Small.      Atlantic       233     0.3     11
Quercus Chapmanii Sarg.    Atlantic       1598    0.9     15
Quercus Durandii Buckl.    Atlantic       1745    0.8     23
Quercus Muehlenbergii Engelm Atlantic     17042   2.0     24
Quercus ilicifolia Wang.   Atlantic       4082    1.1     3
Quercus incana Bartr.      Atlantic       3775    0.6     13
Quercus falcata Michx.     Atlantic       13688   1.8     30
Quercus laevis Walt.       Atlantic       3978    4.8     9
Quercus laurifolia Michx.  Atlantic       5328    1.1     27
Quercus marilandica Muenchh. Atlantic     18480   3.6     9
Quercus nigra L.           Atlantic       10161   1.1     24
Quercus palustris Muenchh. Atlantic       8643    1.1     23
Quercus Phellos L.         Atlantic       9920    3.6     27
Quercus rubra L.           Atlantic       28389   8.1     24
Quercus velutina Lam.      Atlantic       21067   3.6     23
Quercus imbricaria Michx.  Atlantic       14870   1.8     18
Quercus myrtifolia Willd.  Atlantic       2540    0.4     9
Quercus texana Buckl.      Atlantic       829     1.1     9
Quercus coccinea Muenchh.  Atlantic       8992    1.2     4
Quercus Douglasii Hook.Arn California     559     4.1     18
Quercus dumosa Nutt.       California     433     1.6     6
Quercus Engelmannii Greene California     259     2.0     17
Quercus Garryana Hook.     California     1061    5.5     20
Quercus lobata Nee         California     870     5.9     30
Quercus agrifolia Nee.     California     803     2.6     23
Quercus Kelloggii Newb.    California     826     6.0     26
Quercus Wislizenii A. DC.  California     699     1.0     21
Quercus chrysolepis Liebm. California     690     17.1    15
Quercus vaccinifolia Engelm. California     223     0.4     1
Quercus tomentella Engelm  California     13      7.1     18
```

Lesen wir die Daten zunächst ein:

```
> Acorn = read.table("AcornData.txt", header = T, sep = "\t")
> Acorn
```

```
                            Species     Region Range Acorn.size
1                  Quercus alba L.   Atlantic 24196        1.4
2           Quercus bicolor Willd.   Atlantic  7900        3.4
3          Quercus macrocarpa Michx. Atlantic 23038        9.1
4           Quercus prinoides Willd. Atlantic 17042        1.6
5                  Quercus Prinus L. Atlantic  7646       10.5
6             Quercus stellata Wang. Atlantic 19938        2.5
7            Quercus virginiana Mill Atlantic  7985        0.9
8            Quercus Michauxii Nutt. Atlantic  8897        6.8
9               Quercus lyrata Walt. Atlantic  8982        1.8
10            Quercus Laceyi Small.  Atlantic   233        0.3
11           Quercus Chapmanii Sarg. Atlantic  1598        0.9
12           Quercus Durandii Buckl. Atlantic  1745        0.8
13      Quercus Muehlenbergii Engelm Atlantic 17042        2.0
14          Quercus ilicifolia Wang. Atlantic  4082        1.1
15             Quercus incana Bartr. Atlantic  3775        0.6
16            Quercus falcata Michx. Atlantic 13688        1.8
17              Quercus laevis Walt. Atlantic  3978        4.8
18         Quercus laurifolia Michx. Atlantic  5328        1.1
19      Quercus marilandica Muenchh. Atlantic 18480        3.6
```

```
20                  Quercus nigra L.   Atlantic 10161        1.1
21         Quercus palustris Muenchh.   Atlantic  8643        1.1
22                 Quercus Phellos L.   Atlantic  9920        3.6
23                   Quercus rubra L.   Atlantic 28389        8.1
24              Quercus velutina Lam.   Atlantic 21067        3.6
25          Quercus imbricaria Michx.   Atlantic 14870        1.8
26          Quercus myrtifolia Willd.   Atlantic  2540        0.4
27              Quercus texana Buckl.   Atlantic   829        1.1
28          Quercus coccinea Muenchh.   Atlantic  8992        1.2
29 Quercus Douglasii Hook. &amp;amp; Arn California   559        4.1
30               Quercus dumosa Nutt. California   433        1.6
31         Quercus Engelmannii Greene California   259        2.0
32             Quercus Garryana Hook. California  1061        5.5
33                 Quercus lobata Nee California   870        5.9
34             Quercus agrifolia Nee. California   803        2.6
35            Quercus Kelloggii Newb. California   826        6.0
36          Quercus Wislizenii A. DC. California   699        1.0
37         Quercus chrysolepis Liebm. California   690       17.1
38       Quercus vaccinifolia Engelm. California   223        0.4
39          Quercus tomentella Engelm California    13        7.1
   Tree.Height
1         27.0
2         21.0
3         25.0
4          3.0
5         24.0
6         17.0
7         15.0
8          0.3
9         24.0
10        11.0
11        15.0
12        23.0
13        24.0
14         3.0
15        13.0
16        30.0
17         9.0
18        27.0
19         9.0
20        24.0
21        23.0
22        27.0
23        24.0
24        23.0
25        18.0
26         9.0
27         9.0
28         4.0
29        18.0
30         6.0
31        17.0
32        20.0
33        30.0
34        23.0
35        26.0
36        21.0
37        15.0
```

```
38         1.0
39        18.0
```

```
> lapply(Acorn, class)
```

```
$Species
[1] "factor"

$Region
[1] "factor"

$Range
[1] "integer"

$Acorn.size
[1] "numeric"

$Tree.Height
[1] "numeric"
```

```
> attach(Acorn)
```

Der vorletzte Befehl prüft den Datentyp der einzelnen geladenen Variablen, der letzte Befehl (`attach`) stellt die Variablen des Datensatzes direkt zur Verfügung, so dass wir uns das `Acorn$` davor sparen können. Attach muss mittels `detach` aufgehoben werden, wenn wir mit dem Datensatz fertig sind.

2.1.1 Das Streudiagramm

Bevor wir uns mit der Theorie beschäftigen, werfen wir zunächst einen Blick auf die einfachste statistische Graphik: das **Streudiagramm** (auch **Scatterplot** genannt). Das Streudiagramm erzeugen wir mit dem Befehl `plot` (Abb. 2.2):

```
> plot(Tree.Height, Acorn.size)
```

Das Streudiagramm stellt für jedes statistische Individuum einen Punkt dar, der zwei reelle Merkmale graphisch kodiert. Die Merkmalswerte werden durch die Lage des Punktes dargestellt. Dazu wird ein kartesisches Koordinatensystem genutzt. Der erste Merkmalswert x_{i1} bestimmt den Rechtswert des i-ten Punktes im Koordinatensystem und der zweite Merkmalswert x_{i2} den Hochwert des Punktes. Auf diese Weise werden die Merkmalsinformationen vollständig dargestellt.

Das kartesische Koordinatensystem ist jedem aus der Schule hinlänglich bekannt, so dass es keine Mühe macht, die Bedeutung der Graphik zu verstehen. Sie folgt einem einfachen graphischen Grundprinzip:

> Reelle Merkmalswerte werden durch die geometrische Lage dargestellt.

Die Graphik erscheint also völlig trivial.

Dennoch bleiben die meisten praktisch relevanten Fragen mit der Definition erst einmal offen:

- Wofür eignet sich ein Streudiagramm und wofür nicht?
  (z.B. für welche Skalen)
- In welchen Situationen setzt man ein Streudiagramm ein?
  (z.B. für einen ersten Überblick)
- Welche Aussagen über die Grundgesamtheit können wir einem Streudiagramm entnehmen? Und wie macht man das?
  (überraschend wenige)

```
> plot(Tree.Height, Acorn.size)
```

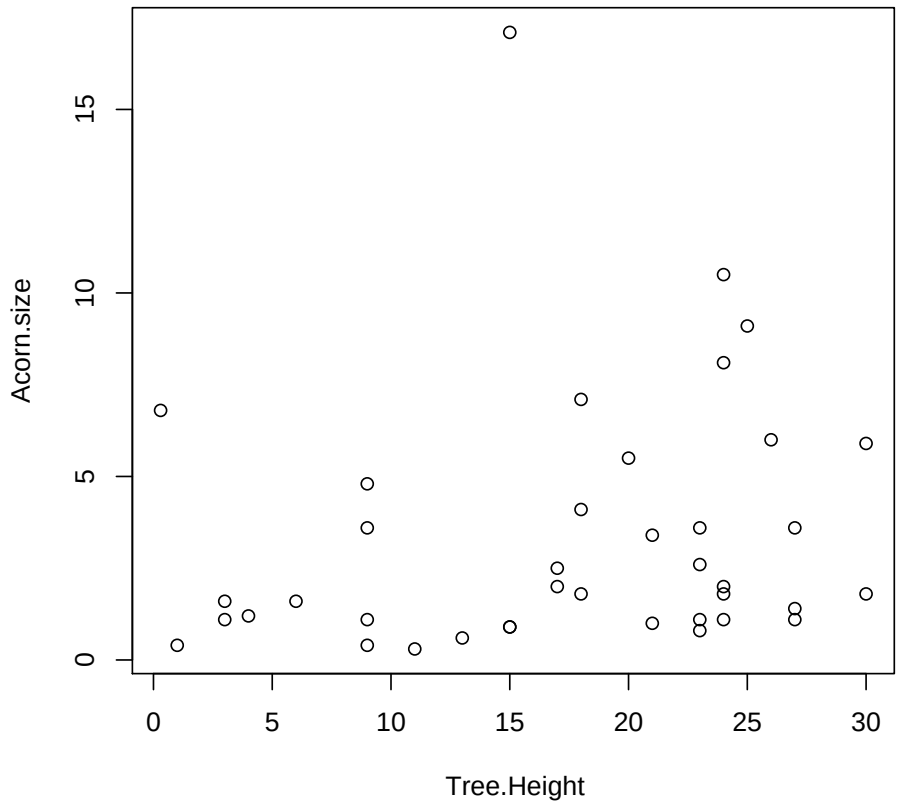


Abbildung 2.2: Einfaches Streudiagramm
Ein einfaches Streudiagramm. Jedes statistische Individuum wird dabei als ein Punkt dargestellt. Die kartesischen Koordinaten des Punktes ergeben sich aus den Werten bei zwei metrischen Merkmalen. Der Zahlenwert wird also in eine geometrische Information umgesetzt.

- Welche Aussagen über die Grundgesamtheit kann das menschliche Auge einem Streudiagramm nicht oder nur schwer entnehmen?
  (z.B. den Mittelwert)
- Welche Informationen werden in einem Streudiagramm völlig unsichtbar?
  (z.B. die Reihenfolge der Beobachtungen)
- Wie kann man ein Streudiagramm modifizieren, um es besser interpretieren zu können?
  (z.B. mit farbigen Punkten oder einem besser angepassten Koordinatensystem)

2.1.2 Systematik der deskriptiven Methoden

Aus dieser Perspektive betrachtet erscheint das Streudiagramm als eine relativ komplizierte und auch spezialisierte Graphik. Wir werden daher die grundlegenden Graphiken, nach Einsatzbereich systematisiert, nach all diesen Fragen durchsprechen.

Dabei ergibt sich eine Einteilung nach der Anzahl darzustellender Merkmale und nach der Skala der darzustellenden Merkmale:

- Univariat (Betrachtung eines einzelnen Merkmals)
  - reelle Daten
    - Datenzentriert: Punktdiagramm
    - Verteilungszentriert: Histogramm, Kerndichteschätzer
    - Gesamtheitszentriert: Boxplot
    - Kenngrößen
      - Lage: arithmetischer Mittelwert, Median, Quantile
      - Streuung: Varianz, Standardabweichung, IQR
  - positive Daten
    - Datenzentriert: Punktdiagramm in log-Skala
    - Verteilungszentriert: Histogramm der log-Daten
    - Gesamtheitszentriert: Boxplot der log-Daten
    - Kenngrößen
      - Lage: geometrischer Mittelwert, Median, Quantile
      - Streuung: Variationskoeffizient, metrische Varianz, geometrische Standardabweichung, geometrischer IQR
  - kategorielle Daten
    - Mengenzentriert: Balkendiagramm
    - Anteilzentriert: Kuchendiagramm
    - Kenngrößen: Anteil, Wahrscheinlichkeit
- Bivariat (Betrachtung zweier Merkmale)
  - reell – reell (oder positiv)
    - Streudiagramm
    - Kenngrößen: Kovarianz, Pearson Korrelation, Rangkorrelation
  - positiv – positiv
    - Log-Log-Diagramm
    - Kenngrößen: metrische Kovarianz, Rangkorrelation

  - kategoriell – kategoriell
    - bedingend: gestapelte Balkendiagramme
    - vergleichend: gruppierte Balkendiagramme
    - Kenngrößen: bedingte Wahrscheinlichkeit, Odds, Logodds
  - kategoriell – reell
    - Datenzentriert: parallele Punktdiagramme
    - Gesamtheitszentriert: parallele Boxplots
    - Schließend: gekerbte parallele Boxplots
    - Kenngrößen: R^2

- Multivariat (Betrachtung mehrere Merkmale)
  - mehrere reelle Daten
    - Streudiagrammmatrix
    - 3D-Streudiagramme
    - nD-Streudiagramme/Projektion Persuite
    - Kenngrößen: Kovarianzmatrix, Korrelationsmatrix, verallgemeinerte Varianz, verallgemeinerte Kovarianz
    - Biplot
  - mehrere kategorielle Daten
    - Mosaikplot
    - Verbundene interaktive Balkendiagramme
    - ...
  - mehrere reelle Daten und eine kategorielle Größe
    - Trellis
    - Streudiagramme mit Symbolen
    - ...
  - Zusammensetzungsdaten
    - Ternäre Diagramme
    - ...

2.2 Univariate Graphik und Beschreibung für stetige Daten

Die folgenden Graphiken dienen der Darstellung von Daten mit stetiger Skala:

- Punktdiagramm (normal, verzittert, gestapelt)
- Histogramm
- Boxplot oder Kastendiagramm

```
> opar <- par(mfrow = c(3, 1), pch = 20)
> stripchart(Acorn.size, main = "Punktdiagramm")
> stripchart(Acorn.size, method = "stack", main = "gestapeltes Punktdiagramm")
> stripchart(Acorn.size, method = "jitter", main = "verzittertes Punktdiagramm")
> par(opar)
```

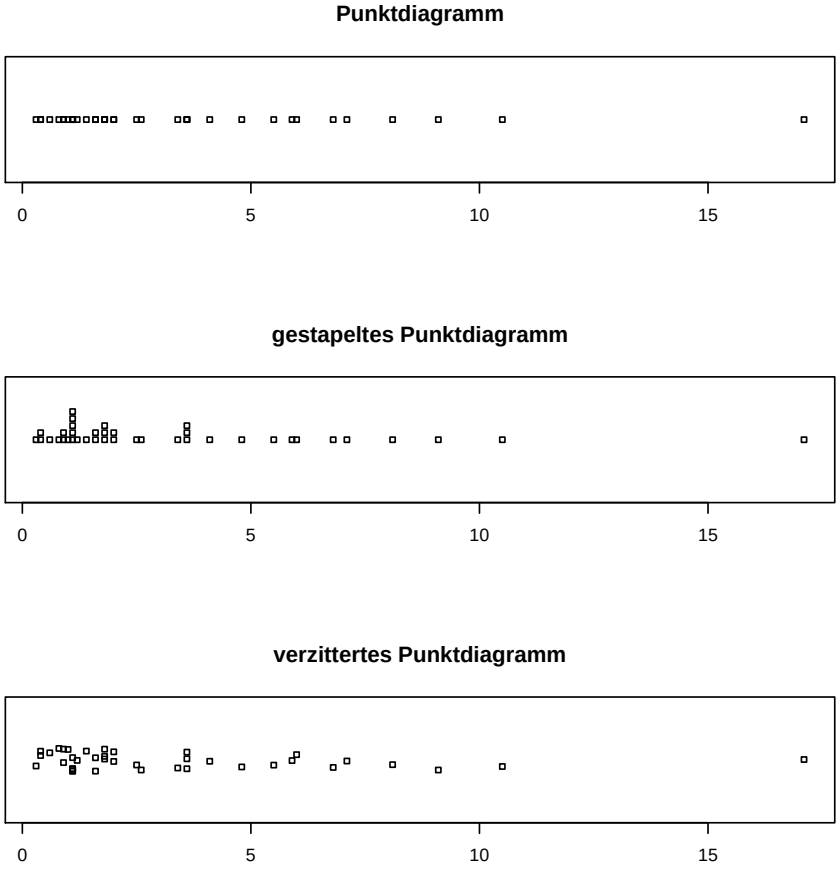


Abbildung 2.3: Punktdiagramm
Drei Punktdiagramme von `Acorn.size`

2.2.1 Punktdiagramm

Die folgenden Befehle erzeugen die Punktdiagramme aus Abbildung 2.3.

```
> opar <- par(mfrow = c(3, 1), pch = 20)
> stripchart(Acorn.size, main = "Punktdiagramm")
> stripchart(Acorn.size, method = "stack", main = "gestapeltes Punktdiagramm")
> stripchart(Acorn.size, method = "jitter", main = "verzittertes Punktdiagramm")
> par(opar)
```

Das Punktdiagramm folgt dem Prinzip, den Wert des Merkmals durch den Ort darzustellen, an dem ein Punkt für das statistische Individuum erscheint. Leider können sich zwei Punkte mit ähnlichen oder gleichen Werten dadurch leicht überdecken, so dass gerade in den Bereichen mit besonders vielen Daten Punkte durch "**Überdeckung**" verloren gehen. Dagegen hilft ein leichtes Abändern der Punktposition in der ungenutzten Richtung durch zufälliges "**Verzittern**", oder gezieltes **Stapeln** der Punkte mit dem gleichen Merkmalswert.

2.2.1.1 Diskussion

- Die Information wird (bis auf das Problem der Überdeckung) vollständig dargestellt.
- Überdeckungen können diese Graphik stark verfälschen, insbesondere, wenn die dargestellten Werte stark gerundet sind. Diesem Problem kann man durch verzittern oder stapeln entgegentreten.
- Ein verzittertes Punktdiagramm sieht nach jedem Neuzeichnen anders aus.
- Ein gestapeltes Punktdiagramm gaukelt dem Auge Muster vor, wo nur Zufall ist.
- Das Auge kann die Verteilung der Punkte nur sehr schwer erfassen. Man sieht sozusagen den Wald vor lauter Bäumen nicht.

2.2.2 Histogramm

```
> hist(Acorn.size)
```

Das Histogramm (von gr. histaemi stellen) versucht die Dichte der Punkte möglichst gut darzustellen. Es zählt dazu die Punkte, die in einen Bereich der x-Achse fallen und stellt anstelle der Punkte einen Balken dar, dessen Fläche proportional zur Anzahl der darin enthaltenen Punkte ist. Sind alle Balken gleich breit, so ist damit natürlich auch die Höhe des Balkens proportional zur Anzahl der Punkte. In jedem Fall ist aber die Höhe des Balkens proportional zum Anteil Punkte pro Längenabschnitt der x-Achse. Diesen Anteil pro Fläche nennt man auch Dichte.

```
> hist(Acorn$Acorn.size, sub = "als Dichte", freq = FALSE)
```

Viele theoretische Verteilungen werden beschrieben, indem ihre Dichte angegeben wird. Z.B. können wir die letzte Graphik mit der Dichte, der sogenannten Lognormalverteilung, mit den passenden Parametern vergleichen (ohne, das wir Befehl oder Konzept schon verstehen):

```
> x <- seq(0, 10, by = 0.1)
> lines(x, dlnorm(x, mean = mean(log(Acorn.size)), sd = sd(log(Acorn.size))))
```

```
> detach(Acorn)
```

```
> opar <- par(mfrow = c(1, 3), pch = 20)
> hist(Acorn.size)
> hist(Acorn.size, sub = "mit Erklaerung")
> stripchart(Acorn.size, method = "stack", add = T)
> hist(Acorn.size, sub = "als Dichte", freq = FALSE)
> x <- seq(0, 10, by = 0.1)
> lines(x, dlnorm(x, mean = mean(log(Acorn.size)), sd = sd(log(Acorn.size))))
> par(opar)
```

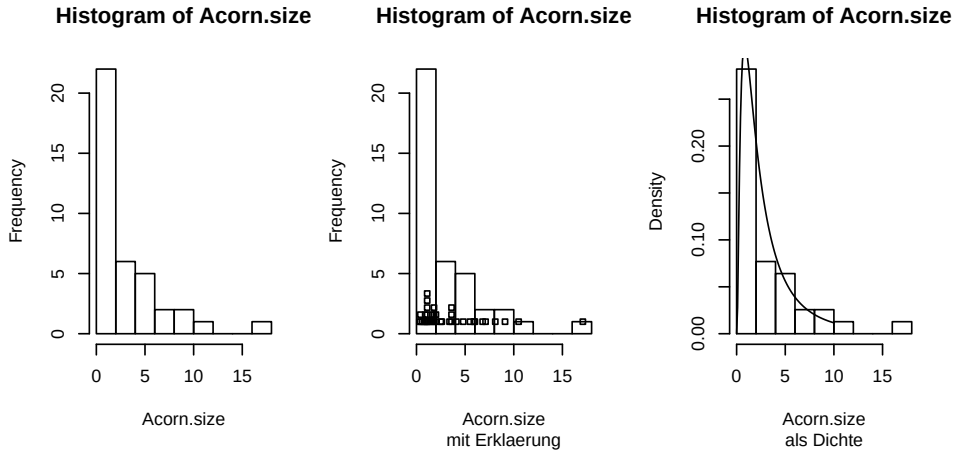


Abbildung 2.4: Histogramm

2.2.2.0.1 Konfidenz: Genauigkeit des Histogramms

```
> lpois <- function(x, p, lower.tail = TRUE, log.p = FALSE) {
+     if (length(p) > 1)
+         return(sapply(p, function(p) lpois(x, p, lower.tail = lower.tail,
+             log.p = log.p)))
+     if (length(x) > 1)
+         return(sapply(x, function(x) lpois(x, p, lower.tail = lower.tail,
+             log.p = log.p)))
+     obj <- function(ll, y = x, pr = p) ppois(q = y, lambda = 1/(1 -
+         ll) - 1, lower.tail = lower.tail, log.p = log.p) -
+         pr
+     r <- uniroot(obj, c(1e-04, 0.999))
+     1/(1 - r$root) - 1
+ }
> CI.hist <- function(x, ..., freq = TRUE, probability = !freq,
+     alpha = 0.05) {
+     h <- hist(x, ..., freq = freq, probability = probability)
+     lb <- lpois(h$counts, alpha/2)
+     ub <- lpois(h$counts, 1 - alpha/2)
+     cc <- h$counts
+     if (probability) {
+         segments(h$mids, h$density * lb/cc, h$mids, h$density *
+             ub/cc)
+     }
+     else {
+         segments(h$mids, lb, h$mids, ub)
+     }
+     h$lb <- lb
```

```
> opar <- par(mfrow = c(4, 3))
> for (x in c(10, 7.5, 5, 2.5, 2, 1, 0.8, 0.6, 0.5, 0.25,
+     0.2, 0.1)) {
+     hist(Acorn.size, breaks = c(seq(0, 20, by = x), 21),
+         prob = TRUE)
+     x <- seq(0, 20, by = 0.1)
+     lines(x, dlnorm(x, mean = mean(log(Acorn.size)),
+         sd = sd(log(Acorn.size))), col = "red")
+ }
> par(opar)
```

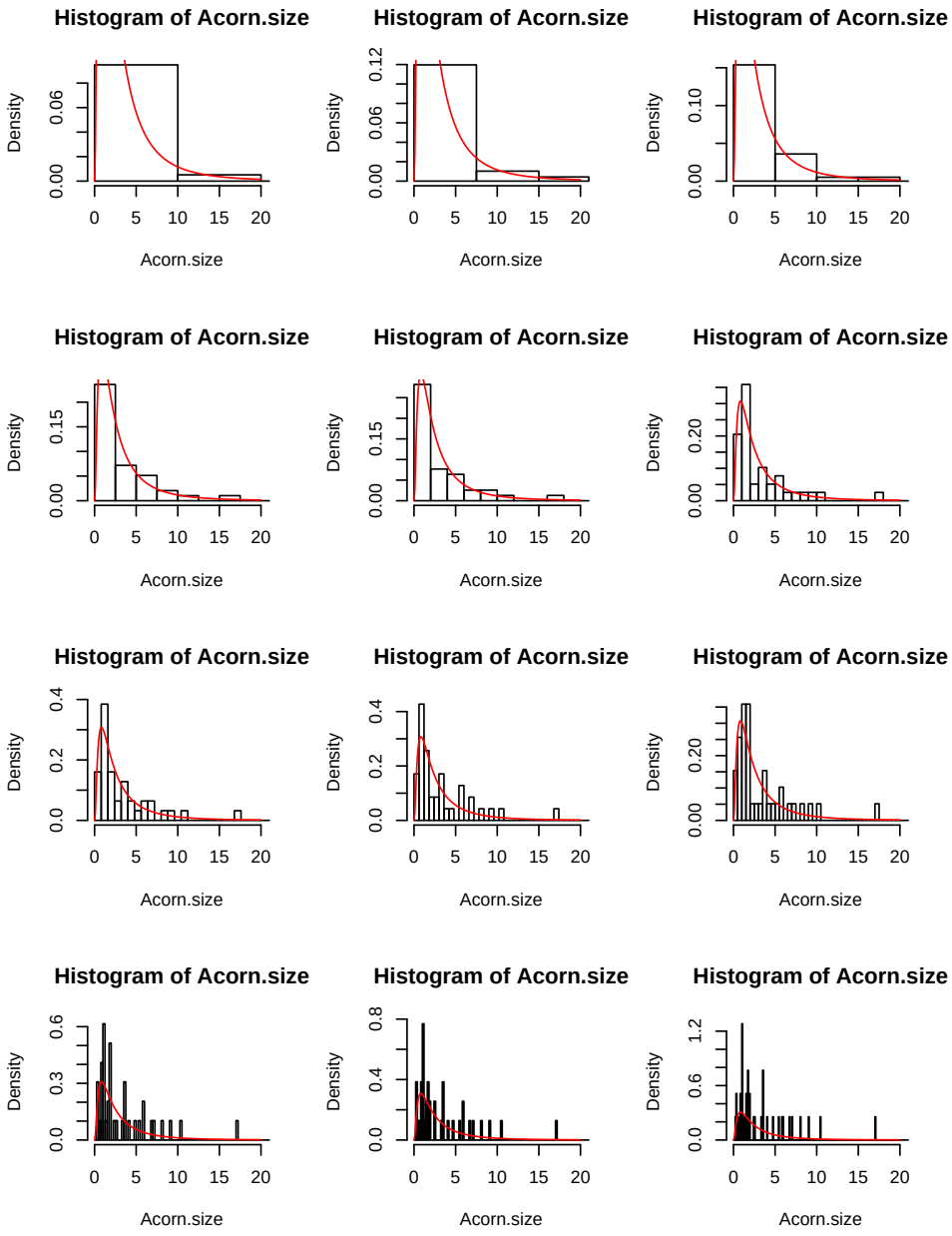


Abbildung 2.5: Einfluss der Balkenbreite auf das Histogramm

```
> opar <- par(mfrow = c(3, 2))
> for (x in c(5, 4, 3, 2, 1, 0)) {
+     hist(Acorn.size, breaks = c(seq(0 - x, 20 - x, by = 2.5),
+         21))
+     x <- seq(0, 20, by = 0.1)
+     lines(x, dlnorm(x, mean = mean(log(Acorn.size)),
+         sd = sd(log(Acorn.size))), col = "red")
+ }
> par(opar)
```

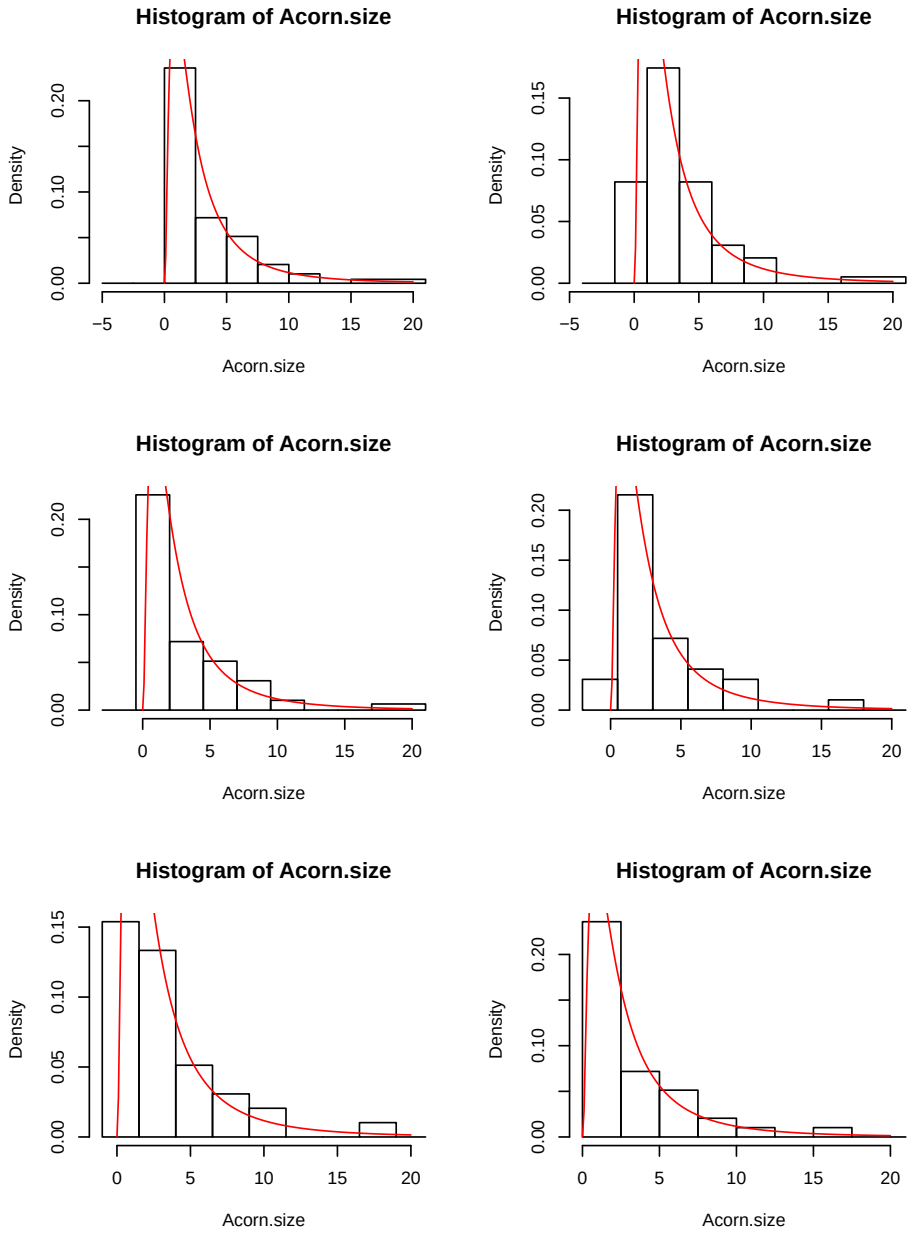


Abbildung 2.6: Einfluss der Balkenanfangsposition auf das Histogramm

```
> opar <- par(mfrow = c(3, 3))
> hist(rnorm(10000))
> hist(rnorm(1000))
> hist(rnorm(1000))
> hist(rnorm(100))
> hist(rnorm(100))
> hist(rnorm(100))
> hist(rnorm(20))
> hist(rnorm(20))
> hist(rnorm(20))
> par(opar)
```

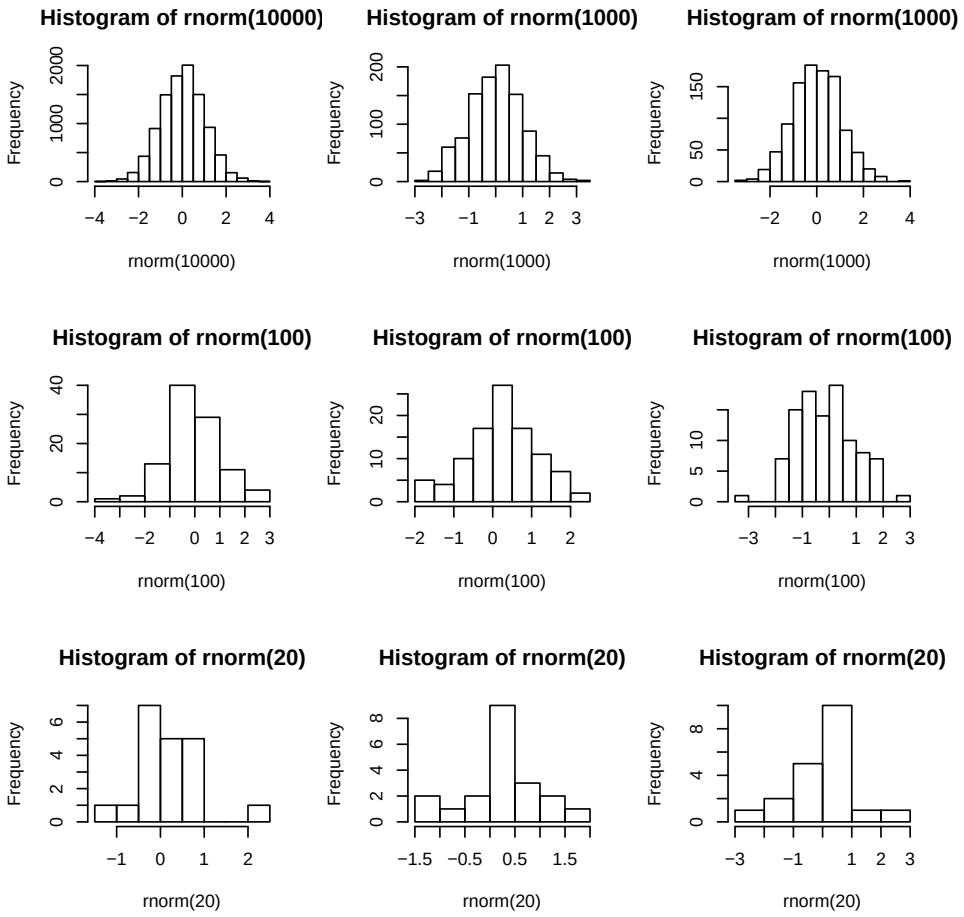


Abbildung 2.7: Histogramm de Normalverteilung
Einige Histogramme normalverteilter Daten zur Eichung des Auges. Der Befehl `rnorm` (von Random NORMal) simuliert die angegebene Anzahl normalverteilter Zufallswerte mit Mittelwert 0 und Standardabweichung 1.

```
> hist(rnorm(100), prob = TRUE, xlim = c(-5, 5), ylim = c(0,
+     0.5), col = "gray", main = "Histogramm und Dichte\n einer Normalverteilung")
> x <- seq(-5, 5, by = 0.01)
> lines(x, dnorm(x), lwd = 2)
> text(-3, 0.35, expression(f(x) == frac(1, sqrt(2 * pi *
+     sigma^2)) * e^-frac((x - mu)^2, 2 * sigma^2)), cex = 1.5)
```

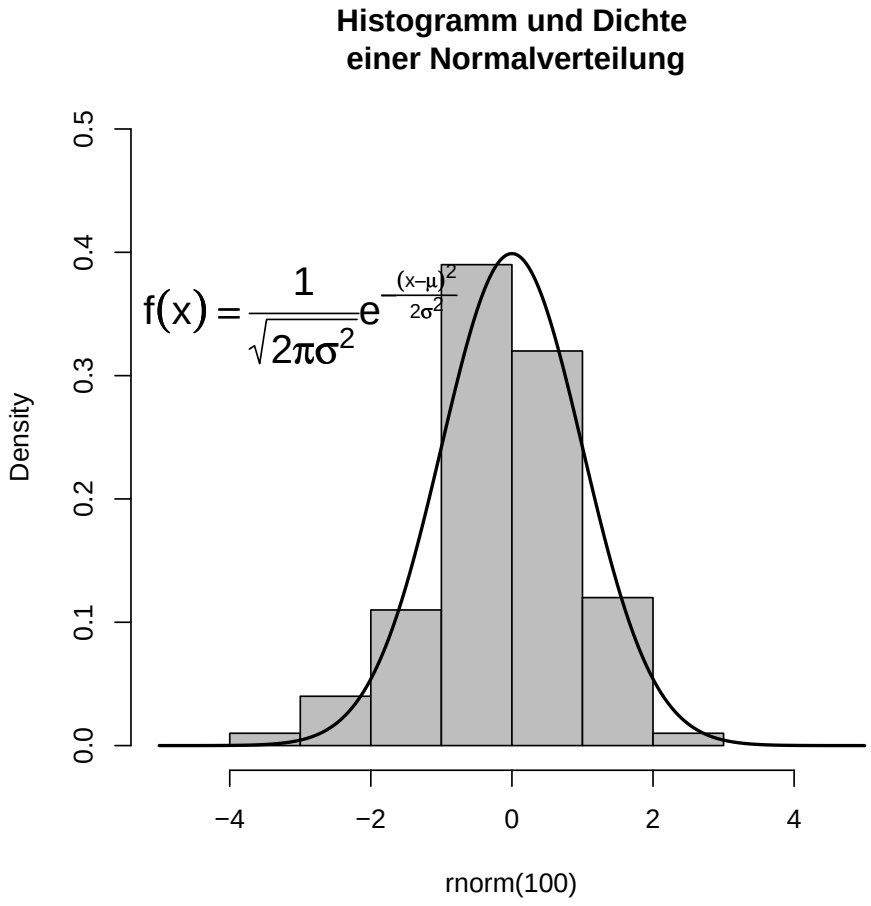


Abbildung 2.8: Dichte der Normalverteilung
Die Graphik zeigt ein Histogramm eines normalverteilten Datensatzes und die zugehörige Dichtefunktion.

```
+     h$ub <- ub
+     invisible(h)
+ }
> CI.hist(iris$Petal.Length)
```

2.2.3 Normalverteilung als Referenzverteilung

2.2.3.1 Die Normalverteilung

Die wichtigste Referenzverteilung der Statistik ist die **Normalverteilung**. Die Abbildung 2.7 enthält Histogramme einiger solcher Datensätze, die mittels einer idealen Normalverteilung simuliert wurden. Abbildung 2.8 enthält die Dichtefunktion einer Normalverteilung mit Parameter $\mu = 0$ und $\sigma^2 = 1$. Diese Parameter geben in Abschnitt 2.2.4 beschriebenen Eigenschaften von Erwartungswert und Varianz der Grundgesamtheit an. Die Normalverteilung wurde von Carl Friedrich Gauss eingeführt und scheint aus verschiedene Gründen an vielen Stellen in der Natur (zumindest ungefähr) aufzutreten:

- Nach dem **zentralen Grenzwertsatz** entsteht, wenn sich viele verschiedene Fehler und Variationen additiv überlagern, ungefähr eine Normalverteilung. Das betrifft insbesondere Größen, die in irgendeiner Weise durch Mittelwertbildung zustandekommen.
- Die Normalverteilung tritt in natürlicher Weise als Verteilung in vielen physikalischen und wirtschaftlichen Prozessen, z.B. Diffusion, Brownsche Molekularbewegung oder Börsenhandel auf.
- Viele andere physikalisch interpretierbare Verteilungen sind einer Normalverteilung sehr ähnlich.
- Viele Fragestellungen können nur für die Normalverteilung exakt gelöst werden. Die dafür entwickelten Methoden können aber oft auch unverändert angewendet werden, wenn die Verteilung der Normalverteilung ähnlich ist, was sehr häufig vorkommt.

Da außerdem die Normalverteilung mathematisch relativ leicht zu behandeln ist und auf einfache Ergebnisse führt, die auch noch schnell berechnet werden können, hat die Normalverteilung in der Statistik eine zentrale Rolle:

- Sie gilt als Referenzverteilung, mit der man die tatsächliche Verteilung der Daten vergleicht.
- Es gibt einen unübersehbaren Vorrat an fertigen und weit bekannten Methoden für die Normalverteilung.
- Weiterreichende Konzepte lassen sich oft anhand von Methoden für die Normalverteilung am leichtesten erklären.

Die Verteilungsdichte der Normalverteilung ist gegeben durch:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

2.2.3.2 Verteilungseigenschaften

Die Normalverteilung ist auf den ganzen reellen Zahlen definiert und eignet sich daher nur zur Modellierung reell skalierter Merkmale und erst einmal nur näherungsweise für die Modellierung relativ skalierter Merkmale. Für relativ skalierte

Merkmale verwendet man daher besser die der Normalverteilung eng verwandte Lognormalverteilung. Ist der Wertebereich einer Verteilung nicht die ganzen reellen Zahlen, spricht man daher von einem eingeschränkten Wertebereich.

Die Normalverteilung ist symmetrisch um den Wert μ, d.h. im gleichen Abstand von diesem Wert nach rechts und nach links trifft man die gleichen Dichtewerte. Verteilungen mit dieser Eigenschaft heißen **symmetrisch**. Ist eine Verteilung nicht symmetrisch um einen zentralen Wert, so heißt sie **asymmetrisch**.

Bei der Normalverteilung liegen am Symmetriezentrum die Werte am dichtesten und die Dichte fällt nach beiden Seiten etwa gleich schnell ab. Die Stelle höchster Dichte heißt Modus der Verteilung. Verteilungen mit mehreren Stellen höchster Dichte heißen **multimodal** oder **mehrgipflig**. Liegt nur eine Stelle höchster Dichte vor, so spricht man von **unimodalen** oder **eingipfligen** Verteilungen. Detaillierter wird das in Abschnitt 2.2.5.7 besprochen. Unimodale asymmetrische Verteilungen heißen **schief**. Diese Verteilungen heißen **steil** an der Seite, an der die Dichte vom Modalwert aus gesehen schneller abfällt. Also z.B. **linkssteil**, wenn die Dichte zu kleineren Werten hin schnell abfällt und **rechtssteil**, wenn die Dichte zu den größeren Werten hin schnell abfällt. Die andere Seite, zu der hin die Dichte langsamer abfällt, heißt **schief**. Eine Verteilung heißt also **linksschief**, wenn die Dichte zu den kleinen Werten hin langsam abfällt und **rechtsschief**, wenn die Dichte zu den großen Werten hin langsamer abfällt. Konventionell beschreibt man die Schiefe der Verteilung, indem man die schiefe Seite nennt (und nicht die steile). Ist die Schiefe der Verteilung so stark ausgeprägt, dass die Dichte auf der steilen Seite praktisch sofort auf 0 fällt, so heißt sie auch **monoton**. Eine extrem rechtsschiefe Verteilung hieße somit **monoton fallend**, weil die Dichte abfällt. Monotone Verteilungen deuten normalerweise auf einen eingeschränkten Bereich hin.

Rechtsschiefe Verteilungen findet man insbesondere bei solchen Merkmalen häufig, die besser auf einer relativen Skala zu bearbeiten wären, wie z.B. geochemische Daten. Monoton fallende Verteilungen kenne ich insbesondere von Überlebenszeiten.

2.2.4 Kenngrößen und Parameter

Statistische Graphiken vermitteln ein Gefühl für die Form einer Verteilung. Statistische Kenngrößen fassen einen Datensatz quantitativ zusammen. Diese Kenngrößen fassen einen Datensatz oder eine Grundgesamtheit mit wenigen Zahlen zusammen. Man spricht daher auch von **deskriptiven** Statistiken (d.h. beschreibenden Statistiken). Es gibt:

1. **Lageparameter**
   Lageparameter geben einen Eindruck, in welchem Wertebereich die Daten liegen bzw. mit welchem durchschnittlichen Wert man rechnen muss, wenn man viele Individuen (z.B. Kunden, Feriengäste, Abbaublöcke, Lose) aus der Grundgesamtheit zufällig auswählt.

2. **Streuparameter**
   Streuparameter geben einen Eindruck davon, wie ähnlich sich die verschiedenen Individuen der Verteilung untereinander sind, also “wie weit sie im Punktdiagramm verstreut sind”.

3. **Formparameter**
   Formparameter versuchen die Form einer Verteilung (wie man sie im Histogramm sieht) quantitativ zu beschreiben. Ihre wichtigsten Vertreter sind Schiefe (Skewness) und Kurtosis (Wölbung). Die Interpretation von Formparametern ist jedoch relativ schwierig und daher empfehle ich immer eher eine Graphik zu zeigen als solche Parameter zu bemühen.

```
> opar = par(mfrow = c(3, 3))
> set.seed(13456)
> hist(rnorm(1000, mean = 3), main = "symmetrisch eingipflig",
+     col = "gray")
> hist(rlnorm(1000, mean = log(3), sd = 0.3), main = "rechtsschief",
+     col = "gray")
> hist(c(rnorm(1000, mean = 3, sd = 0.4), rnorm(500, mean = 5,
+     sd = 0.4)), main = "zweigipflig/bimodal", col = "gray")
> hist(c(rnorm(1000, 3, 0.3), rnorm(500, 5, 0.3), rnorm(1000,
+     1, 0.3)), main = "multimodal", col = "gray")
> hist(rbeta(1000, 10, 2), main = "linksschief, eingeschraenkt",
+     col = "gray")
> hist(rbeta(10000, 1, 1), main = "Gleichverteilung auf [0,1]",
+     col = "gray")
> hist(rcauchy(1000), main = "Schwere Verteilungsschwaenze",
+     breaks = 200, col = "gray", xlim = c(-100, 100))
> hist(c(rnorm(100, mean = 3), 20), main = "Ausreisser",
+     breaks = 50, col = "gray")
> hist(rexp(300), main = "rechtsschief\nmonoton fallend\n nach unten beschraenkt",
+     breaks = 12, col = "gray")
> par(opar)
```

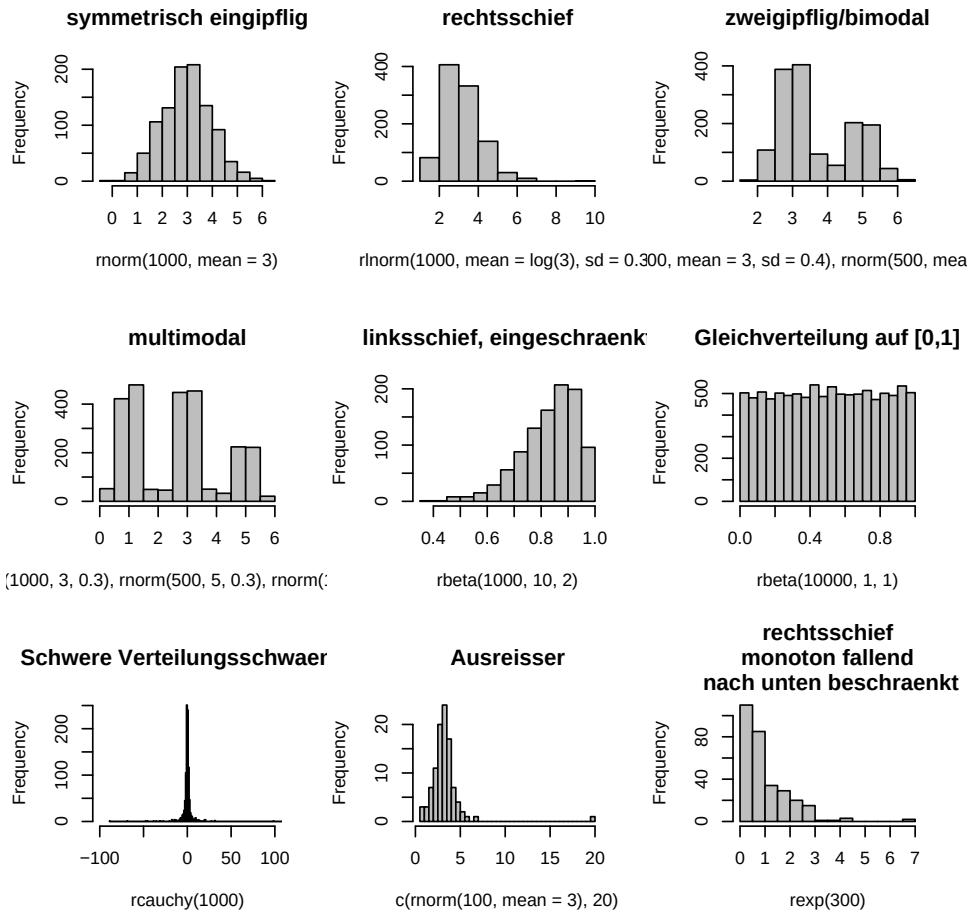


Abbildung 2.9: Klassische Verteilungseigenschaften
Die Bilder zeigen jeweils sehr klare Beispiele der jeweiligen Eigenschaften.

4. **Verteilungsparameter**
   Verteilungsparameter sind eigentlich Parameter eines gedachten mathematischen Modells für die Grundgesamtheit. Sie haben manchmal eine inhaltliche Interpretation, wie z.B. als Gesamtzahl von vorhandenen Individuen oder als Rate des Auftretens von Unfällen, und sind oft eng verwandt mit speziellen Lage oder Streuparametern. Der Wert eines Verteilungsparameters kann oft aus dem Datensatz durch spezielle mathematische Prozeduren ungefähr geschätzt werden.

Man muss diese Parameter insbesondere aus zwei Gründen kennen:

- Um Verteilungen beschreiben zu können. z.B. im Rahmen von Forschungsberichten oder Diplomarbeiten.
- Um die Beschreibungen anderer Leute (insbesondere in wissenschaftlichen Veröffentlichungen) richtig deuten zu können.

2.2.5 Lageparameter

Lageparameter beschreiben, wie groß die beobachteten Werte ungefähr sind. Die physikalische Einheit der Lageparameter entspricht grundsätzlich der physikalischen Einheit der Daten selbst.

2.2.5.1 Arithmetischer Mittelwert

Der arithmetische Mittelwert ist wohl die berühmteste statistische Kenngröße, auch wenn er wegen seiner Anfälligkeit gegenüber extremen Werten (sogenannten Ausreißern) in letzter Zeit in Verruf gekommen ist.

Der arithmetische **Mittelwert** (engl. arithmetic mean) $\bar{x}$ eines Datensatzes mit Werten $x_i, i = 1, \ldots, n$ berechnet sich gemäß der Formel:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + x_2 + \ldots + x_n)$$

Der arithmetische Mittelwert beschreibt die durchschnittliche Größe der Werte und eignet sich insbesondere für reell skalierte Größen und solche, für welche die Addition eine wichtige Rolle spielt (z.B. Gewinne, Kosten, Kohlevorrat, ...). Der Mittelwert der Stichprobe schätzt darüber hinaus einen Verteilungsparameter, den Erwartungswert, der seinerseits als Mittelwert über die Grundgesamtheit definiert ist.

Der Mittelwert bildet den Schwerpunkt der Punktwolke im Punktdiagramm und die x-Koordinate des Schwerpunktes der Fläche unter der (dem Histogramm verwandten) Verteilungsdichte. Ein einzelner, riesiger Wert kann den Mittelwert jedoch beliebig stark verändern, so dass sich Datenfehler oder seltene Extremwerte (Ausreißer) stark auf den Mittelwert auswirken können. Man sagt daher: "Der Mittelwert ist nicht robust".

Definition 21 (Robustheit) *Ein statistisches Verfahren heißt* **robust** *mit* **Bruchpunkt** *p, wenn eine beliebige Abänderung von weniger als dem Anteil p die Ergebnisse nicht beliebig stark beeinflussen können.*

Der Mittelwert einer reellen Variable kann in R mit dem Befehl `mean` berechnet werden:

```
> mean(iris$Sepal.Length)

[1] 5.843333
```

```
> opar = par(mfrow = c(2, 2))
> attach(iris)
> hist(Sepal.Length, xlim = c(0, 10))
> abline(v = mean(Sepal.Length), col = "red")
> hist(Sepal.Width, xlim = c(0, 10))
> abline(v = mean(Sepal.Width), col = "red")
> hist(Petal.Length, xlim = c(0, 10))
> abline(v = mean(Petal.Length), col = "red")
> hist(Petal.Width, xlim = c(0, 10))
> abline(v = mean(Petal.Width), col = "red")
> detach(iris)
> par(opar)
```

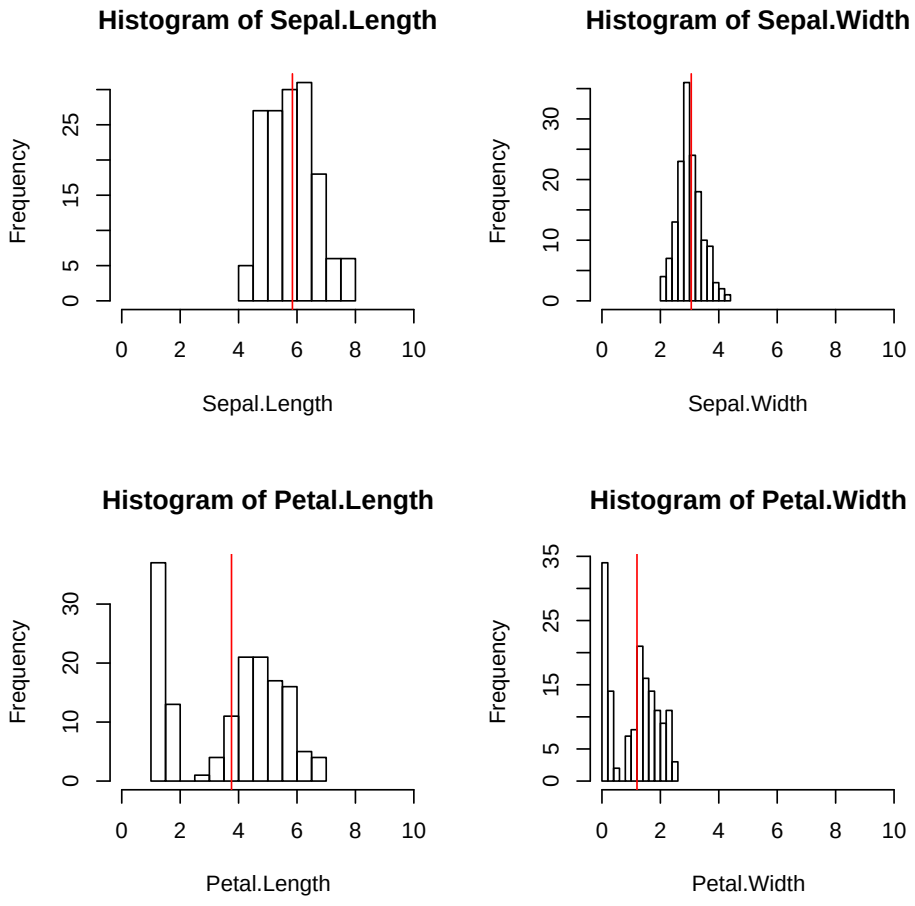


Abbildung 2.10: Mittelwert als Lageparameter
Der Mittelwert, hier jeweils als senkrechte Linie eingezeichnet, zeigt den ungefähren Wert der Daten. Ohne ein Gefühl für die Streuung der Daten ist der Mittelwertes jedoch schwer zu interpretieren. Insbesondere müssen im Bereich des Mittelwertes nicht unbedingt viele Daten liegen.

```
> opar = par(mfrow = c(2, 2))
> attach(iris)
> hist(Sepal.Length, xlim = c(0, 10))
> abline(v = mean(Sepal.Length), col = "red")
> hist(Sepal.Width, xlim = c(0, 10))
> abline(v = mean(Sepal.Width), col = "red")
> hist(Petal.Length, xlim = c(0, 10))
> abline(v = mean(Petal.Length), col = "red")
> hist(Petal.Width, xlim = c(0, 10))
> abline(v = mean(Petal.Width), col = "red")
> detach(iris)
> par(opar)
```

Eine robuste Version des Mittelwertes zum Bruchpunkt p, das sogenannte **getrimmte** Mittel, kann durch Angabe des Bruchpunktes p als Trimmung (`trim=`p) berechnet werden. Es sind Bruchpunkte im Bereich $0 \leq p \leq 0.5$ zulässig. Beachten Sie bitte, dass R immer Dezimalpunkte statt Dezimalkommata verwendet:

```
> mean(iris$Sepal.Length, trim = 0.3)

[1] 5.815
```

Beim getrimmten Mittel werden ein Anteil `trim` der Daten oben und unten weggelassen und der Mittelwert aus den verbleibenden Daten bestimmt. Dadurch wird die Bestimmung des Mittelwertes natürlich auch ungenauer.

Die Anwendung von `mean` auf einen Datensatz berechnet den Mittelwert für jede enthaltene Variable.

```
> mean(iris)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    5.843333     3.057333     3.758000     1.199333           NA
```

Der Mittelwert ist für kategorielle Größen nicht definiert. Daher gibt R für die kategorielle Größe den Wert `NA` (NA=Not Available, also keine Angabe) und eine entsprechende Warnung aus.

Ein Vergleich der berechneten Mittelwerte weißt darauf hin, dass sowohl in Länge als auch in der Breite das Kelchblatt deutlich größer als das Blütenblatt ist. Beide Blätter sind offenbar länglich. Vor einer direkten Interpretation von Mittelwerten in dieser Weise, ohne eine weitere statistische Absicherung muss jedoch gewarnt werden, da auch bei durchschnittlich gleich großen Blättern immer ein Stichprobenmittelwert größer als der andere sein wird, da beide ja vom Zufall der Stichprobenauswahl beeinflusst werden. Die Absicherung, dass solche Unterschiede nicht vom Zufall herrühren können, ist eine wesentliche Aufgabe der Statistik.

Andererseits ist es natürlich sehr interessant, ob sich die drei Grundgesamtheiten von Irisblüten in den Blattabmessungen unterscheiden. Wir berechnen daher den Mittelwert, als Schätzung für den Erwartungswert, in jeder der drei Grundgesamtheit einzeln. Dazu verwenden wir den folgenden Befehl:

```
> lapply(split(iris, iris$Species), mean)

$setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
       5.006        3.428        1.462        0.246           NA

$versicolor
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
       5.936        2.770        4.260        1.326           NA

$virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
       6.588        2.974        5.552        2.026           NA
```

```
> lapply(split(iris, iris$Species), mean, trim = 0.2)
```

```
$setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
        5.00         3.41         1.46         0.22           NA

$versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    5.910000     2.796667     4.306667     1.340000           NA

$virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    6.546667     2.963333     5.493333     2.023333           NA
```

Der Befehl berechnet hier die Mittelwerte für alle drei Arten getrennt. Eine naive direkte Interpretation der Ausgabe bringt uns zu der Interpretation, dass Iris setosa wohl, bis auf die enorme Breite ihres Kelchblatts die deutlich kleinste unter den Irisblüten ist und Iris virginica in allen Richtungen knapp größer, als Iris versicolor. Außerdem scheinen die generellen Beobachtungen, die wir bereits aus den letzten Mittelwerten gefolgert haben, auch für jede Art einzeln richtig zu sein.

In dieser Interpretation passiert ein wichtiger Abstraktionsschritt: Wir interpretieren die Mittelwerte als generelle Tendenz und Aussagen über eine Irisart an sich und nicht mehr als Aussage über den Datensatz. Dazu muss man sich zumindest über die folgenden Punkte im Klaren sein:

- Unterschiede in Mittelwerten sind immer da und können auch durch den reinen Zufall entstanden sein. Sie brauchen nicht auf eine generelle Tendenz im Datensatz hinzudeuten, es sei denn, wir können mit später zu erlernenden Methoden beweisen, dass die Unterschiede nicht durch Zufall entstanden sein können.
- Die beobachteten Tendenzen können wir grundsätzlich nur für die Grundgesamtheit folgern, für die unsere Stichprobe repräsentativ ist. Man denke sich etwa eine nichtrepräsentative Stichprobe der Menschheit (z.B. die Studenten in diesem Raum) und versuchen über eine generelle Tendenz bezüglich eines der folgenden Merkmale auf die Menschheit zurückzuschließen:
  - Hautfarbe
  - Intelligenz
  - Einkommen
  - Familienstand
  - Gesundheitszustand
  - Wasserverbrauch
  - Musikgeschmack
- Die generelle Tendenz in einer Population bedeutet noch nichts für den Einzelnen: Es gibt z.B. viele Iris Setosa Blüten, die längere Kelchblätter haben als einige Iris virginica.

2.2.5.1.1 R: Verwendung von lapply für die Berechnung mit Teildatensätzen

In der letzten Befehlsfolge spaltet der Befehl `split` gemäß der Konvention:

$$\texttt{split}(\textit{Daten}, \textit{Faktor})$$

den Datensatz zunächst nach der Art in drei Teildatensätze auf:

```
> split(iris, iris$Species)
```

```
$setosa
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
11          5.4         3.7          1.5         0.2  setosa
12          4.8         3.4          1.6         0.2  setosa
13          4.8         3.0          1.4         0.1  setosa
14          4.3         3.0          1.1         0.1  setosa
15          5.8         4.0          1.2         0.2  setosa
16          5.7         4.4          1.5         0.4  setosa
17          5.4         3.9          1.3         0.4  setosa
18          5.1         3.5          1.4         0.3  setosa
19          5.7         3.8          1.7         0.3  setosa
20          5.1         3.8          1.5         0.3  setosa
21          5.4         3.4          1.7         0.2  setosa
22          5.1         3.7          1.5         0.4  setosa
23          4.6         3.6          1.0         0.2  setosa
24          5.1         3.3          1.7         0.5  setosa
25          4.8         3.4          1.9         0.2  setosa
26          5.0         3.0          1.6         0.2  setosa
27          5.0         3.4          1.6         0.4  setosa
28          5.2         3.5          1.5         0.2  setosa
29          5.2         3.4          1.4         0.2  setosa
30          4.7         3.2          1.6         0.2  setosa
31          4.8         3.1          1.6         0.2  setosa
32          5.4         3.4          1.5         0.4  setosa
33          5.2         4.1          1.5         0.1  setosa
34          5.5         4.2          1.4         0.2  setosa
35          4.9         3.1          1.5         0.2  setosa
36          5.0         3.2          1.2         0.2  setosa
37          5.5         3.5          1.3         0.2  setosa
38          4.9         3.6          1.4         0.1  setosa
39          4.4         3.0          1.3         0.2  setosa
40          5.1         3.4          1.5         0.2  setosa
41          5.0         3.5          1.3         0.3  setosa
42          4.5         2.3          1.3         0.3  setosa
43          4.4         3.2          1.3         0.2  setosa
44          5.0         3.5          1.6         0.6  setosa
45          5.1         3.8          1.9         0.4  setosa
46          4.8         3.0          1.4         0.3  setosa
47          5.1         3.8          1.6         0.2  setosa
48          4.6         3.2          1.4         0.2  setosa
49          5.3         3.7          1.5         0.2  setosa
```

```
50          5.0         3.3          1.4         0.2  setosa

$versicolor
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
51           7.0         3.2          4.7         1.4 versicolor
52           6.4         3.2          4.5         1.5 versicolor
53           6.9         3.1          4.9         1.5 versicolor
54           5.5         2.3          4.0         1.3 versicolor
55           6.5         2.8          4.6         1.5 versicolor
56           5.7         2.8          4.5         1.3 versicolor
57           6.3         3.3          4.7         1.6 versicolor
58           4.9         2.4          3.3         1.0 versicolor
59           6.6         2.9          4.6         1.3 versicolor
60           5.2         2.7          3.9         1.4 versicolor
61           5.0         2.0          3.5         1.0 versicolor
62           5.9         3.0          4.2         1.5 versicolor
63           6.0         2.2          4.0         1.0 versicolor
64           6.1         2.9          4.7         1.4 versicolor
65           5.6         2.9          3.6         1.3 versicolor
66           6.7         3.1          4.4         1.4 versicolor
67           5.6         3.0          4.5         1.5 versicolor
68           5.8         2.7          4.1         1.0 versicolor
69           6.2         2.2          4.5         1.5 versicolor
70           5.6         2.5          3.9         1.1 versicolor
71           5.9         3.2          4.8         1.8 versicolor
72           6.1         2.8          4.0         1.3 versicolor
73           6.3         2.5          4.9         1.5 versicolor
74           6.1         2.8          4.7         1.2 versicolor
75           6.4         2.9          4.3         1.3 versicolor
76           6.6         3.0          4.4         1.4 versicolor
77           6.8         2.8          4.8         1.4 versicolor
78           6.7         3.0          5.0         1.7 versicolor
79           6.0         2.9          4.5         1.5 versicolor
80           5.7         2.6          3.5         1.0 versicolor
81           5.5         2.4          3.8         1.1 versicolor
82           5.5         2.4          3.7         1.0 versicolor
83           5.8         2.7          3.9         1.2 versicolor
84           6.0         2.7          5.1         1.6 versicolor
85           5.4         3.0          4.5         1.5 versicolor
86           6.0         3.4          4.5         1.6 versicolor
87           6.7         3.1          4.7         1.5 versicolor
88           6.3         2.3          4.4         1.3 versicolor
89           5.6         3.0          4.1         1.3 versicolor
90           5.5         2.5          4.0         1.3 versicolor
91           5.5         2.6          4.4         1.2 versicolor
92           6.1         3.0          4.6         1.4 versicolor
93           5.8         2.6          4.0         1.2 versicolor
94           5.0         2.3          3.3         1.0 versicolor
95           5.6         2.7          4.2         1.3 versicolor
96           5.7         3.0          4.2         1.2 versicolor
97           5.7         2.9          4.2         1.3 versicolor
98           6.2         2.9          4.3         1.3 versicolor
99           5.1         2.5          3.0         1.1 versicolor
100          5.7         2.8          4.1         1.3 versicolor

$virginica
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
101          6.3         3.3          6.0         2.5 virginica
```

```
102          5.8         2.7          5.1         1.9 virginica
103          7.1         3.0          5.9         2.1 virginica
104          6.3         2.9          5.6         1.8 virginica
105          6.5         3.0          5.8         2.2 virginica
106          7.6         3.0          6.6         2.1 virginica
107          4.9         2.5          4.5         1.7 virginica
108          7.3         2.9          6.3         1.8 virginica
109          6.7         2.5          5.8         1.8 virginica
110          7.2         3.6          6.1         2.5 virginica
111          6.5         3.2          5.1         2.0 virginica
112          6.4         2.7          5.3         1.9 virginica
113          6.8         3.0          5.5         2.1 virginica
114          5.7         2.5          5.0         2.0 virginica
115          5.8         2.8          5.1         2.4 virginica
116          6.4         3.2          5.3         2.3 virginica
117          6.5         3.0          5.5         1.8 virginica
118          7.7         3.8          6.7         2.2 virginica
119          7.7         2.6          6.9         2.3 virginica
120          6.0         2.2          5.0         1.5 virginica
121          6.9         3.2          5.7         2.3 virginica
122          5.6         2.8          4.9         2.0 virginica
123          7.7         2.8          6.7         2.0 virginica
124          6.3         2.7          4.9         1.8 virginica
125          6.7         3.3          5.7         2.1 virginica
126          7.2         3.2          6.0         1.8 virginica
127          6.2         2.8          4.8         1.8 virginica
128          6.1         3.0          4.9         1.8 virginica
129          6.4         2.8          5.6         2.1 virginica
130          7.2         3.0          5.8         1.6 virginica
131          7.4         2.8          6.1         1.9 virginica
132          7.9         3.8          6.4         2.0 virginica
133          6.4         2.8          5.6         2.2 virginica
134          6.3         2.8          5.1         1.5 virginica
135          6.1         2.6          5.6         1.4 virginica
136          7.7         3.0          6.1         2.3 virginica
137          6.3         3.4          5.6         2.4 virginica
138          6.4         3.1          5.5         1.8 virginica
139          6.0         3.0          4.8         1.8 virginica
140          6.9         3.1          5.4         2.1 virginica
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

und `lapply` (für "List APPLY") gemäß dem Aufruf:

`lapply`(*Liste*, *Funktion*)

wendet die Funktion `mean` auf jeden Eintrag unsere Liste von Datensätzen an, wie in:

```
> lapply(list(1, "Nashorn", TRUE), class)

[[1]]
[1] "numeric"
```

```
[[2]]
[1] "character"

[[3]]
[1] "logical"
```

`class` den Datentyp jedes Listeneintrags feststellt und zurückgibt.

2.2.5.2 Theoretische Kenngrößen, Schätzung, Schätzfehler und Vertrauensbereiche

Zu den meisten Kenngrößen einer Stichprobe lassen sich entsprechende Verteilungsparameter angeben, die meist als die entsprechende Maßzahl angewandt auf die Grundgesamtheit verstanden werden kann. Man spricht dann von der "**theoretischen**" Kenngröße. Im Gegensatz dazu heißen die Kenngrößen der Stichprobe "**empirische**".

Beispiel 22 (Erwartungswert) *In unserem Fall heißt die empirische Kenngröße Mittelwert und wird mit $\bar{x}$ abgekürzt. Die entsprechende theoretische Kenngröße (also der Mittelwert der Merkmalswerte in der Grundgesamtheit) heißt Erwartungswert und wird mit μ_1 abgekürzt. Allgemeiner bezeichnet auch das Symbol:*

$$E[X] = \textit{Mittelwert von } X \textit{ über die Grundgesamtheit.}$$

den Erwartungswert von X

Während die empirischen Größen sich leicht aus den Daten berechnen lassen, sind ihre theoretischen Entsprechung meist unbekannt und können nur ungefähr geschätzt werden. Dennoch sind die theoretischen Kenngrössen die für die Wissenschaft eigentlich interessanten, da nur sie eine Aussage über die hinter den Daten stehenden Gesetzmäßigkeiten erlauben und nicht bei der bloßen Beschreibung des Datensatzes stehenbleiben, sondern zu einer Beschreibung der Grundgesamtheit weitergehen können.

Beispiel 23 *Es spielt einfach keine Rolle, was das Durchschnittseinkommen der befragten Gruppe von Touristen (Stichprobe) ist, die Rügen besucht haben. Um das Fremdenverkehrsangebot zu planen, kommt es auf das Durchschnittseinkommen der Touristen an, die Rügen besuchten (Grundgesamtheit). Die Studie macht also nur Sinn, wenn man aus der Stichprobe auf die Grundgesamtheit zurückschließen kann.*

Oft kann eine theoretischen Kenngrößen k aus den Merkmalswerten $x_1, \ldots, x_n$ der Stichprobe durch sogenannte "Schätzer" ungefähr ermittelt werden. Ein "**Schätzer**" $\hat{k}$ für den Parameter k ist eine auf der Stichprobe definierte Funktion oder Rechenvorschrift, welche einen ungefähren Wert für einen Verteilungsparameter berechnet. Den berechneten Wert bezeichnet man als "**Schätzwert**" für den Parameter. Der Schätzer und der Schätzwert für eine Kenngröße k werden oft durch das gleiche Symbol wie der Parameter, mit einem Hut gekennzeichnet: $\hat{k}$. Typischerweise ist der Schätzwert $\hat{k} = \hat{k}(x_1, \ldots, x_n)$ nicht gleich dem wahren Wert des Parameter, sondern nur ähnlich groß. Oft kann der theoretische Wert durch den empirischen geschätzt werden

Beispiel 24 *In unserem Fall*

- *$\bar{x}$: Mittelwert der Stichprobe*
- *μ_0: Erwartungswert oder Mittelwert der Grundgesamtheit*
- *$\hat{\mu}_0$: Allgemeine Bezeichung für einen Schätzer für μ_0*

- *Ein gute Wahl ist $\hat{\mu}_0 = \bar{X}$. Der Erwartungswert wird also durch den Mittelwert der Beobachtungen geschätzt.*

Den (zufälligen und unbekannten) Unterschied $\Delta_k = \hat{k}(x_1, \ldots, x_n) - k$ zwischen Schätzwert $\hat{k}$ und dem wahren Wert des Parameters k bezeichnet man als **Schätzfehler**.

Definition 25 *Wenn man als Grundgesamtheit die Menge der möglichen Stichproben annimmt, so definieren wir:*

- *Den Erwartungswert bias $:= E[\Delta_k] = E[\hat{k}(x_1, \ldots, x_n) - k]$ von Δ_k als die* **Verzerrung** *oder den* **Bias** *des Schätzers $\hat{k}$. Ist $b = 0$ so heißt der Schätzer* **unverzerrt** *oder* **erwartungstreu**.
- *Den Erwartungswert $MSE := E[\Delta_k^2]$ als den* **mittlere quadratischen Schätzfehler** *oder engl. "Mean Squared Error".*
- *Die Wurzel $RMSE := \sqrt{MSE}$ (Root Mean Square Error) des quadratischen Schätzfehlers heißt auch Standardschätzfehler.*

Für Parameter, die nur positive Werte haben können, sind außerdem die sich auf die Logarithmen beziehenden relativen Größen relevant:

- $\Delta_k^* = \ln \hat{k}(x_1, \ldots, x_n) - \ln k$
- *Den Erwartungswert bias* $:= E[\Delta_k^*] = E[\ln \hat{k}(x_1, \ldots, x_n) - \ln k]$ von Δ_k^* als die multiplikative oder geometrische* **Verzerrung** *oder den* **Bias** *des Schätzers $\hat{k}$. Ist $b = 0$, so heißt der Schätzer dann* **geometrisch unverzerrt** *oder* **multiplikativ erwartungstreu** *(engl. geometrically unbiased) .*
- *Den Erwartungswert $MSLE := E[\Delta_k^{*^2}]$ als den* **mittleren quadratischen logarithmischen Schätzfehler** *oder engl. "Mean Squared Logarithmic Error".*
- *Die Wurzel $RMSLE := \sqrt{MSLE}$ (Root Mean Square Logarithmic Error) des quadratischen Schätzfehlers heißt auch logarithmischer Standardschätzfehler.*

Die Schätzfehler sind eine nur zu oft ignorierte, wichtige technische Information im tatsächlichen statistischen Einsatz. Ihre Interpretation entspricht etwa der in Abschnitt 2.2.6 für Varianz und Standardabweichung Beschriebenen und wird später im Detail besprochen. Dennoch ziehe ich es vor, die Information nicht aus dem Kontext zu reißen und füge sie in Abschnitten, wie dem Folgenden ein, die beim ersten Lesen getrost übergangen werden können.

2.2.5.2.1 Schätzfehler: Mittelwert Der Mittelwert $\bar{x}$ ist ein unverzerrter Schätzer für den Erwartungswert. Für eine repräsentative Stichprobe $X_1, \ldots, X_n$ gilt:

$$b = 0$$

$$MSE(\bar{x}, \hat{\mu}_0) = \frac{1}{n}\mathrm{var}(X)$$

$$RMSE(\bar{x}, \hat{\mu}_0) = \frac{1}{n}\mathrm{var}(X) = \frac{1}{\sqrt{n}}\mathrm{sd}(X)$$

wobei $\mathrm{var}(X)$ und $\mathrm{sd}(X)$ die sogenannten Streuparameter der Grundgesamtheit sind. Die Proportionalität von $RMSE$ und $\frac{1}{\sqrt{n}}$ ist mehr oder weniger genau bei fast allen Schätzern zu finden und ist allgemein unter dem Namen **Wurzel-n-Gesetz** bekannt.

Beide Größen können geschätzt werden, so dass man eine entsprechende Schätzung erhält:

$$\widehat{MSE}(\bar{x}, \hat{\mu}_0) = \frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$\widehat{RMSE}(\bar{x}, \hat{\mu}_0) = \sqrt{\widehat{MSE}}$$

```
> "%=%" <- function(x, y) y
> x <- iris$Sepal.Length
> Mittelwert %=% mean(x)

[1] 5.843333

> MSE %=% var(x)/length(x)

[1] 0.00457129

> RMSE %=% sqrt(var(x)/length(x))

[1] 0.06761132
```

Dabei sind die "$Parametername$%=%" nur da, um anzugeben, was berechnet wird. Tatsächlich würde man also schreiben:

```
> x <- iris$Sepal.Length
> mean(x)

[1] 5.843333

> var(x)/length(x)

[1] 0.00457129

> sqrt(var(x)/length(x))

[1] 0.06761132
```

2.2.5.3 Vertrauensbereiche/Konfidenzintervalle

Oft kann man aus der Stichprobe zufällige Intervalle $[u(X_1, \ldots, X_n), o(X_1, \ldots, X_n)]$, sogenannte **Konfidenzintervalle** oder **Vertrauensbereiche** berechnen, welche die wahren Verteilungsparameter unter gewissen Voraussetzungen mit einer Wahrscheinlichkeit von $1 - \alpha$ enthalten. Dabei ist $\alpha > 0$ eine kleine positive Zahl, die man entsprechend der gewünschten Sicherheit festlegen kann. Die entsprechenden Intervalle heißen dann $1 - \alpha$-Konfidenzintervalle für den jeweiligen Parameter.

Mit den Mitteln dieses Kapitels ist es noch nicht möglich, diese Intervalle näher zu besprechen. Dennoch ziehe ich es vor, die entsprechenden für das praktische Arbeiten wichtigen Informationen nicht aus dem Kontext zu reißen. Sie werden daher in speziell gekennzeichneten Paragraphen, wie dem Folgenden eingeführt, die man beim ersten Lesen gefahrlos übergehen kann.

2.2.5.3.1 Konfidenz: Konfidenzintervall des arithmetischen Mittelwertes Unter der Voraussetzung der Normalverteilung lässt sich ein Konfidenzintervall für den Erwartungswert (=Mittelwert der Grundgesamtheit) durch die folgenden Funktionen berechnen:

```
> CInorm.mean = function(x, ...) UseMethod("CInorm.mean",
+     x)
> CInorm.mean.data.frame = function(x, ...) sapply(x, CInorm.mean,
+     ...)
> CInorm.mean.list = function(x, ...) lapply(x, CInorm.mean,
+     ...)
> CInorm.mean.default = function(x, ...) c(u = NA, o = NA)
> CInorm.mean.numeric = function(x, trim = 0, ..., alpha = 0.05) {
+     if (trim != 0)
+         warning("Trimmed mean not supported")
+     x = x[!is.na(x)]
+     m = mean(x, trim = trim)
+     s = sd(x)
```

```
+     df = length(x) - 1
+     q <- qt(1 - alpha/2, df = df)
+     c(u = m - q * s/sqrt(df), o = m + q * s/sqrt(df))
+ }
> CInorm.mean(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.709285    2.986775     3.472231    1.075941      NA
o     5.977382    3.127892     4.043769    1.322725      NA
```

```
> CInorm.mean(split(iris, iris$Species))
```

```
$setosa
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     4.904806    3.319177     1.412144   0.2157457      NA
o     5.107194    3.536823     1.511856   0.2762543      NA

$versicolor
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.787816    2.679914     4.125097    1.269229      NA
o     6.084184    2.860086     4.394903    1.382771      NA

$virginica
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u      6.40545    2.881417     5.393561    1.947153      NA
o      6.77055    3.066583     5.710439    2.104847      NA
```

für allgemeinere Verteilungen mag ein Bootstrap basiertes Intervall angemessener sein:

```
> CIboot.mean = function(x, ...) UseMethod("CIboot.mean",
+     x)
> CIboot.mean.data.frame = function(x, ...) sapply(x, CIboot.mean,
+     ...)
> CIboot.mean.list = function(x, ...) lapply(x, CIboot.mean,
+     ...)
> CIboot.mean.default = function(x, ...) c(u = NA, o = NA)
> CIboot.mean.numeric = function(x, ..., alpha = 0.05) {
+     if (alpha < 0.01)
+         warning("CIboot did not use enough samples")
+     ms <- sapply(1:10000, function(i, ...) mean(sample(x,
+         length(x), TRUE), ...), ...)
+     m <- mean(x, ...)
+     q1 <- unname(quantile(ms, alpha/2))
+     q2 <- unname(quantile(ms, 1 - alpha/2))
+     c(u = m - (q2 - m), o = m + (m - q1))
+ }
> CIboot.mean(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.713333    2.986667     3.485317    1.078667      NA
o     5.977333    3.126683     4.041350    1.320000      NA
```

2.2.5.4 Geometrischer Mittelwert

Der geometrische Mittelwert hat die gleichen Aufgaben und Probleme wie der arithmetische, eignet sich für relativ skalierte Größen und ist insbesondere anzuraten, wenn

- die Addition von Werten in der Untersuchung keine Rolle spielt.
- die Verteilung stark rechtsschief ist.

- der Unterschied zwischen 1 und 2 ungleich größer und bedeutender erscheint, als der zwischen 1000 und 1001.

Der geometrische Mittelwert ist definiert als

$$\bar{x}^* = \sqrt[n]{\prod_{i=1}^{n} x_i} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln x_i\right)$$

und hat in R (ohne das Erweiterungspaket "compositions" für relative Skalen) zunächst keinen eigenen Befehl, was allerdings leicht zu beheben ist:

```
> gmean = function(x, ...) UseMethod("gmean")
> gmean.default = function(x, ...) NA
> gmean.numeric = function(x, ...) {
+     exp(mean(log(x), ...))
+ }
> gmean.data.frame = function(x, ...) sapply(x, gmean,
+     ...)
> gmean(iris$Sepal.Length)

[1] 5.78572

> mean(iris$Sepal.Length, trim = 0.3)

[1] 5.815

> gmean(iris)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
   5.7857204    3.0265978    3.2382668    0.8417075           NA

> lapply(split(iris, iris$Species), gmean)

$setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
   4.9938411    3.4070803    1.4517340    0.2265819           NA

$versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    5.913979     2.751874     4.233081     1.311187           NA

$virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    6.557795     2.957014     5.525789     2.007214           NA

> lapply(split(iris, iris$Species), gmean, trim = 0.2)

$setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
   4.9979264    3.4058413    1.4582718    0.2168944           NA

$versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    5.904537     2.792120     4.300748     1.336649           NA

$virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
    6.540957     2.960233     5.486452     2.016675           NA
```

2.2.5.4.1 R: Definition eigener Funktionen in R eigentlich ist die Definition eigener Funktionen (wie hier des geometrischen Mittelwertes) ganz einfach und folgt dem Muster:

Funktionsname = `function(`*Argumentliste*`)` *Ausdruck*

Dabei kann der Ausdruck selbst wieder ein kompliziertes R-Programm sein und die in der Argumentliste als übergebenen Parameter als Werte verwenden. Ein Beispiel:

```
> f = function(x) x^2
> f(0)

[1] 0

> f(1)

[1] 1

> f(2)

[1] 4
```

Die oben definierte neue Funktion für den geometrischen Mittelwert ist da schon etwas komplizierter. Sie ist nämlich eine sogenannte generische Funktion, die, je nachdem was für ein Parameter übergeben wurde, eine andere Berechnung durchführt. Wird eine numerische Variable übergeben, so wird in `gmean.numeric` gemäß der weiter oben gegebenen Formel der geometrische Mittelwert bestimmt. Dabei sorgt `...` für die eventuelle Weitergabe des `trim` Parameters oder anderer noch nicht besprochender Parameter. Wird ein Datensatz übergeben, so wird mittels `sapply` (Simple APPLY) in `gmean.data.frame`, das so ähnlich funktioniert wie `lapply`, die `gmean` Funktion einfach auf jede Variable einzeln angewendet und das Ergebnis zurückgegeben. Für andere Eingaben wie z.B. Faktoren wissen wir nicht, was wir machen sollen und geben deshalb in `gmean.default` (default=Standard) einfach NA (Not Available) für unbekannt zurück. Das `UseMethod("gmean")` in der Definition von `gmean` sorgt dafür, das jeder Aufruf von `gmean` an die richtige Methode (`gmean.numeric`, `gmean.data.frame`, `gmean.default`) von `gmean` weitergeleitet wird, je nachdem welche `class` `x` hat. Diese Technik nennt man **S3** generische Funktionen.

2.2.5.4.2 Konfidenz: Konfidenzintervall des geometrischen Mittelwertes Unter der Voraussetzung der Lognormalverteilung lässt sich ein Konfidenzintervall für den geometrischen Erwartungswert (=geometrischer Mittelwert der Grundgesamtheit) durch die folgenden Funktionen berechnen:

```
> CIlnorm.gmean = function(x, ...) UseMethod("CIlnorm.gmean",
+     x)
> CIlnorm.gmean.data.frame = function(x, ...) sapply(x,
+     CIlnorm.gmean, ...)
> CIlnorm.gmean.list = function(x, ...) lapply(x, CIlnorm.gmean,
+     ...)
> CIlnorm.gmean.default = function(x, ...) c(u = NA, o = NA)
> CIlnorm.gmean.numeric = function(x, trim = 0, ..., alpha = 0.05) {
+     if (trim != 0)
+         warning("Trimmed mean not supported")
+     x = log(x[!is.na(x)])
+     m = mean(x, trim = trim, ...)
+     s = sd(x)
+     df = length(x) - 1
+     q <- qt(1 - alpha/2, df = df)
+     exp(c(u = m - q * s/sqrt(df), o = m + q * s/sqrt(df)))
+ }
> CIlnorm.gmean(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.654982    2.957321     2.943231   0.7178807      NA
o     5.919481    3.097498     3.562877   0.9868933      NA
```

```
> CIlnorm.gmean(split(iris, iris$Species))
```

```
$setosa
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     4.893789    3.298629     1.402236   0.2014981      NA
o     5.095939    3.519098     1.502979   0.2547882      NA

$versicolor
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.767754    2.660875     4.094848    1.254906      NA
o     6.063912    2.845985     4.375981    1.369993      NA

$virginica
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     6.377465    2.866610     5.372827    1.928543      NA
o     6.743224    3.050269     5.683106    2.089095      NA
```

für allgemeinere Verteilungen mag ein Bootstrap basiertes Intervall angemessener sein:

```
> CIboot.gmean = function(x, ...) UseMethod("CIboot.gmean",
+     x)
> CIboot.gmean.data.frame = function(x, ...) sapply(x,
+     CIboot.gmean, ...)
> CIboot.gmean.list = function(x, ...) lapply(x, CIboot.gmean,
+     ...)
> CIboot.gmean.default = function(x, ...) c(u = NA, o = NA)
> CIboot.gmean.numeric = function(x, ..., alpha = 0.05) {
+     if (alpha < 0.01)
+         warning("CIboot did not use enough samples")
+     ms <- log(sapply(1:10000, function(i, ...) gmean(sample(x,
+         length(x), TRUE), ...), ...))
+     m <- log(gmean(x, ...))
+     q1 <- unname(quantile(ms, alpha/2))
+     q2 <- unname(quantile(ms, 1 - alpha/2))
+     exp(c(u = m - (q2 - m), o = m + (m - q1)))
+ }
> CIboot.gmean(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     5.656899    2.958250     2.948970   0.7244135      NA
o     5.918343    3.096928     3.568902   0.9850394      NA
```

2.2.5.5 Median

Der **Median** ist der mittlere Wert der Daten und ein Schätzer für den Verteilungsparameter "Median der Grundgesamtheit". Seine Berühmtheit verdankt er der großen Robustheit zum Bruchpunkt $p = 0.5$ und seiner einfachen Interpretation als Mitte der Daten. Die Regeln zur Ermittlung des Median sind einfach:

Bei einem Datensatz (z.B. -0.36, -0.7, -0.31, 0.73, 0.65) mit ungerader Länge (hier n=5) sortieren wir zunächst, die Zahlen der Größe nach: -0.7, -0.36, -0.31, 0.65, 0.73 und zählen dann den mittleren Wert Nr. $\frac{n+1}{2} = 3$ ab, also hier $median = -0.31$.

Bei einem Datensatz (z.B. -0.07, -0.61, 2.08, -0.73, 0.09, -1.01) mit gerader Länge (hier n=6) sortieren wir zunächst wieder, die Zahlen der Größe nach: -1.01, -0.73, -0.61, -0.07, 0.09, 2.08 und bilden dann den Mittelwert $\frac{1}{2}(-0.61 + -0.07) = -0.34$ der beiden mittleren Werte -0.61 und -0.07 an Nr. $\frac{n}{2}$ und Nr. $\frac{n}{2} + 1$.

```
> median(iris$Sepal.Length)
```

```
[1] 5.8
```

```
> sapply(iris[, 1:4], median)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
        5.80         3.00         4.35         1.30
```

Der Median entspricht dem getrimmten Mittelwert zu $p = 0.5$.

```
> sapply(iris[, 1:4], mean, trim = 0.5)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
        5.80         3.00         4.35         1.30
```

```
> sapply(iris[, 1:4], gmean, trim = 0.5)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
    5.800000     3.000000     4.349713     1.300000
```

Man erkauft sich also die extrem robuste Schätzung mit etwas Genauigkeit:

```
> n <- 20
> sd(sapply(1:10000, function(i) mean(rnorm(n)))) * sqrt(n)
```

```
[1] 1.002436
```

```
> sd(sapply(1:10000, function(i) median(rnorm(n)))) * sqrt(n)
```

```
[1] 1.193717
```

2.2.5.5.1 Konfidenz: Konfidenzintervall für den Median Für eine repräsentative Stichprobe lässt sich ein Konfidenzintervall für den theoretischen Median (=Median der Grundgesamtheit) folgendermaßen berechnen:

```
> qlimits <- function(x, probs, alpha = 0.05) {
+     perm <- order(x)
+     xperm <- x[perm]
+     n <- length(x)
+     lo <- qbinom(alpha/2, n, probs, lower.tail = TRUE)
+     hi <- (n + 1) - qbinom(alpha/2, n, 1 - probs, lower.tail = TRUE)
+     mid <- round((n + 1) * probs)
+     xac <- function(x, i) c(-Inf, x, Inf)[ifelse(i <
+         1, 1, ifelse(i > length(x), length(x) + 2, i +
+         1))]
+     erg <- rbind(u = xac(xperm, lo), o = xac(xperm, hi))
+     colnames(erg) <- format(probs)
+     erg
+ }
> CI.median = function(x, ...) UseMethod("CI.median", x)
> CI.median.data.frame = function(x, ...) sapply(x, CI.median,
+     ...)
> CI.median.list = function(x, ...) lapply(x, CI.median,
+     ...)
> CI.median.default = function(x, ...) c(u = NA, o = NA)
> CI.median.numeric = function(x, ..., alpha = 0.05) {
+     structure(c(qlimits(x, probs = 0.5, alpha = alpha)),
+         names = c("u", "o"))
+ }
> CI.median(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u          5.6         3.0          4.0         1.2      NA
o          6.0         3.1          4.6         1.5      NA
```

```
> CI.median(split(iris, iris$Species))
```

```
$setosa
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u          4.9         3.3          1.4         0.2      NA
o          5.1         3.5          1.5         0.2      NA

$versicolor
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u          5.7         2.7          4.1         1.3      NA
o          6.1         2.9          4.5         1.4      NA

$virginica
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u          6.3         2.8          5.2         1.9      NA
o          6.7         3.0          5.7         2.1      NA
```

2.2.5.6 Quantile

Der Median ist der Wert, so dass der Anteil $p = 0.5$ der Daten kleiner gleich diesem Wert ist. Verwendet man statt von $p = 0.5$ einen beliebigen Anteil p, so spricht man von einem p-Quantil.

Definition 26 Quantile

- *Jeder Wert, zu dem der Anteil p der Daten kleiner gleich und der Anteil* $1-p$ *größer ist, heißt p-***Quantil** *der Stichprobe. Wenn es viele solcher Werte gibt, so wird einer ausgewählt. Dafür gibt es viele verschiedene Regeln.*
- *Das p-Quantil wird mit* q_p *bezeichnet.*
- *Das* $\frac{1}{2}$*-Quantil* $q_{0.5}$ *heißt auch* **Median**.
- *Das* $\frac{1}{4}$*-Quantil* $q_{0.25}$*heißt auch unteres* **Quartil** *oder 1. Quartil*
- *Das* $\frac{3}{4}$*-Quantil* $q_{0.75}$ *heißt auch oberes* **Quartil** *oder 3. Quartil* [1]
- *Die* $\frac{n}{10}$*-Quantile* $q_{0.n}$ *heißen auch n-tes* **Dezentil**.
- *Das größte* 0*-Quantil* $\min(x)$ *entspricht dem kleinsten Datenwert und heißt auch* **Minimum**.
- *Das kleinste* 1*-Quantil* $\max(x)$ *entspricht dem größten Datenwert und heißt auch* **Maximum**.
- *Die Quantile (Median, Quartile, ...) der Strichprobe heißen auch* **empirische** *Quantile (Median, Quartile, . . .). Wie diese Zuordnung genau stattfindet, hängt davon ab in welches Buch man sieht oder welches Computerprogramm man benutzt. Eine häufige Wahl ist bei einer Stichprobe vom Umfang n den i-ten kleinsten Wert mit dem* $\frac{i-0.5}{n}$*-Quantil zu identifizieren.*
- *Die Quantile (Median, Quartile, ...) der Grundgesamtheit heißen auch* **theoretische** *Quantile (Median, Quartile, . . .)*

```
> stripchart(iris$Sepal.Width, method = "stack")
> abline(v = quantile(iris$Sepal.Width, seq(0, 1, 0.1)),
+     col = "red")
```

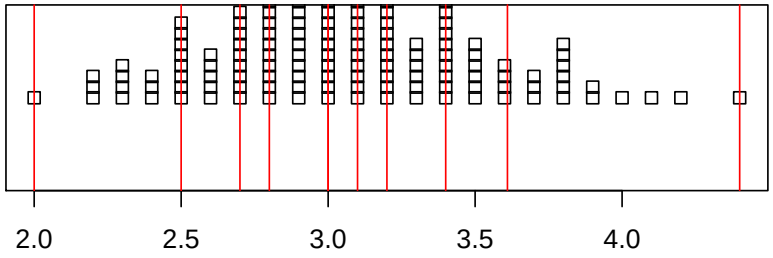


Abbildung 2.11: Darstellung einer Verteilung durch ihre Dezentile. Es ist z.B. leicht zu erkennen, dass 80% der Daten im Bereich 2.5 bis 3.6 liegen.

Die Angabe der Quantile erlaubt meist eine recht gute Einschätzung der Verteilung der Daten.

```
> quantile(iris$Sepal.Length)

  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9

> quantile(iris$Sepal.Length, c(0.25, 0.75))

25% 75%
5.1 6.4

> quantile(iris$Sepal.Length, seq(0, 1, 0.1))

  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
4.30 4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90 7.90

> sapply(iris[1:4], quantile, seq(0, 1, 0.25))

     Sepal.Length Sepal.Width Petal.Length Petal.Width
0%            4.3         2.0         1.00         0.1
25%           5.1         2.8         1.60         0.3
50%           5.8         3.0         4.35         1.3
75%           6.4         3.3         5.10         1.8
100%          7.9         4.4         6.90         2.5

> summary(iris)

  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
```

[1] das $\frac{2}{4}$-Quantil, also der Median bildet somit das 2.Quartil

```
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

```
> stripchart(iris$Sepal.Width, method = "stack")
> abline(v = quantile(iris$Sepal.Width, seq(0, 1, 0.1)),
+     col = "red")
```

dabei ist `seq` ein kleiner Befehl um eine Folge von Werten in einem festen Abstand zu erzeugen:

```
> seq(0, 100, 5)

 [1]   0   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75
[17]  80  85  90  95 100

> seq(0, 100, length = 5)

[1]   0  25  50  75 100
```

2.2.5.6.1 Konfidenz: Konfidenzintervall für die Quantile Für eine repräsentative Stichprobe lässt sich ein Konfidenzintervall für die theoretischen Quantile angeben:

```
> qlimits <- function(x, probs, alpha = 0.05) {
+     perm <- order(x)
+     xperm <- x[perm]
+     n <- length(x)
+     lo <- qbinom(alpha/2, n, probs, lower.tail = TRUE)
+     hi <- (n + 1) - qbinom(alpha/2, n, 1 - probs, lower.tail = TRUE)
+     mid <- round((n + 1) * probs)
+     xac <- function(x, i) c(-Inf, x, Inf)[ifelse(i <
+         1, 1, ifelse(i > length(x), length(x) + 2, i +
+         1))]
+     erg <- rbind(u = xac(xperm, lo), o = xac(xperm, hi))
+     colnames(erg) <- format(probs)
+     erg
+ }
> CI.quantile = function(x, ...) UseMethod("CI.quantile",
+     x)
> CI.quantile.data.frame = function(x, ...) lapply(x, CI.quantile,
+     ...)
> CI.quantile.list = function(x, ...) lapply(x, CI.quantile,
+     ...)
> CI.quantile.default = function(x, ...) c(u = NA, o = NA)
> CI.quantile.numeric = function(x, probs = seq(0, 1, 0.25),
+     ..., alpha = 0.05) {
+     qlimits(x, probs = probs, alpha = alpha)
+ }
> CI.quantile(iris)

$Sepal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  5.0  5.6  6.3  7.9
```

```
o  4.3  5.4  6.0  6.7  Inf

$Sepal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  2.7  3.0  3.2  4.4
o    2  2.9  3.1  3.4  Inf

$Petal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  1.5  4.0  4.9  6.9
o    1  1.9  4.6  5.5  Inf

$Petal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  0.2  1.2  1.6  2.5
o  0.1  0.5  1.5  2.0  Inf

$Species
 u  o
NA NA
```

```
> CI.quantile(split(iris, iris$Species))
```

```
$setosa
$setosa$Sepal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  4.6  4.9  5.1  5.8
o  4.3  4.9  5.1  5.4  Inf

$setosa$Sepal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  3.0  3.3  3.5  4.4
o  2.3  3.4  3.5  3.8  Inf

$setosa$Petal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  1.3  1.4  1.5  1.9
o    1  1.4  1.5  1.6  Inf

$setosa$Petal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  0.2  0.2  0.2  0.6
o  0.1  0.2  0.2  0.4  Inf

$setosa$Species
 u  o
NA NA


$versicolor
$versicolor$Sepal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  5.5  5.7  6.1    7
o  4.9  5.7  6.1  6.6  Inf

$versicolor$Sepal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  2.4  2.7  2.9  3.4
o    2  2.7  2.9  3.1  Inf
```

```
$versicolor$Petal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  3.7  4.1  4.5  5.1
o    3  4.2  4.5  4.7  Inf

$versicolor$Petal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  1.0  1.3  1.4  1.8
o    1  1.3  1.4  1.5  Inf

$versicolor$Species
 u  o
NA NA


$virginica
$virginica$Sepal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  5.9  6.3  6.7  7.9
o  4.9  6.4  6.7  7.4  Inf

$virginica$Sepal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  2.6  2.8  3.0  3.8
o  2.2  2.9  3.0  3.3  Inf

$virginica$Petal.Length
  0.00 0.25 0.50 0.75 1.00
u -Inf  5.0  5.2  5.6  6.9
o  4.5  5.3  5.7  6.1  Inf

$virginica$Petal.Width
  0.00 0.25 0.50 0.75 1.00
u -Inf  1.8  1.9  2.1  2.5
o  1.4  1.9  2.1  2.3  Inf

$virginica$Species
 u  o
NA NA
```

2.2.5.7 Modalwert

Alle bisherigen Lageparameter haben ein gemeinsames Problem: an der angegeben Stelle muss kein oder nur ein Wert liegen. Dabei erscheint es vielen Menschen logisch die Lage der Daten durch diejenige Stelle zu beschreiben, an der die Daten am dichtesten liegen. Dieser Wert heißt dann **Modus** oder **Modalwert**. Leider kann der Modalwert zumindest bei stetig verteilten Merkmalen oft nicht eindeutig oder nicht exakt bestimmt werden. Er kann daher nicht von einem Computerprogramm berechnet werden, sondern muss im Histogramm abgelesen werden.

Definition 27

- *Ein Wert, zu dem nirgends in der Umgebung die Werte dichter liegen, heißt ein* **Modus** *oder* **Modalwert** *der Verteilung.*
- *Hat eine Stichprobe oder Verteilung mehrere klare Modi vergleichbarer Dichte, so heißt sie* **multimodal**. *Sind es genau zwei Modi vergleichbarer Dichte, so heißt sie auch* **bimodal**.
- *Hat eine Stichprobe oder Verteilung nur einen Modus, so heißt sie* **unimodal** *oder* **eingipflig**.

Abbildung 2.12 erklärt das Ablesen der Modalwerte am Beispiel.

```
> opar <- par(mfrow = c(1, 2))
> hist(iris$Sepal.Length)
> points(6.25, 32, col = "red", pch = 20)
> hist(iris$Petal.Length)
> points(1.25, 37, col = "red", pch = 20)
> points(4.5, 22, col = "red", pch = 20)
> par(opar)
```

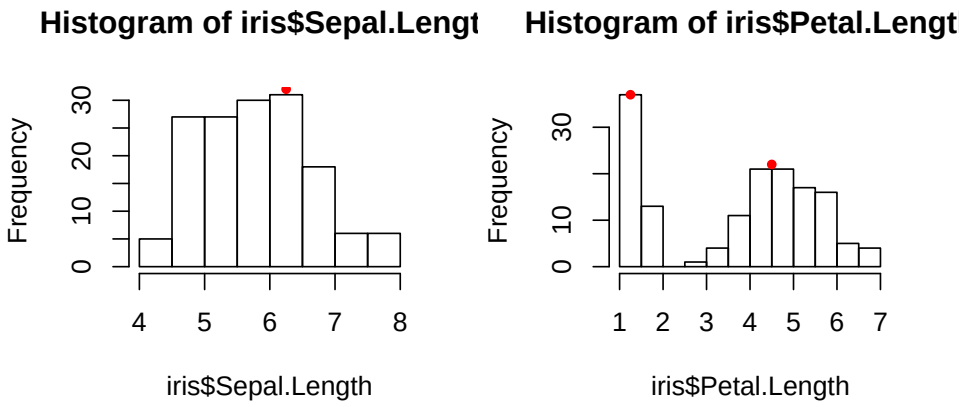


Abbildung 2.12: Modalwerte
Bestimmung von Modalwerten. Modalwerte sind nicht eindeutig definiert und werden meist im Histogramm abgelesen. Im linken Bild sieht man eine klare unimodale Verteilung mit einem Modus im Bereich 6.0-6.5. Im rechten Bild eine bimodale Verteilung mit einem dominanten Modus im Bereich 1.0-1.5 und einem sekundären Modus im Bereich 4.0-5.0. Die Modi haben wir durch die `points` Befehle per Hand mit roten Punkten markiert. Wir würden also sagen: Die Kelchblattlänge ist unimodal und hat einen Modus bei 6.25. Die Blütenblattlänge hat ist bimodal und hat einen schärferen Modus bei 1.25 und einen breiteren Modus bei 4.5. Es sei jedoch gewarnt, dass man mit einem anderen Histogramm möglichwerweise auch zu anderen Schlüssen gekommen wäre. Bimodale und multimodale Verteilungen deuten oft darauf hin, dass es noch wesentlich verschiedene Untergruppen in den Daten gibt. Hier sind es die verschiedenen Arten der Irisblüten.

2.2.6 Streuparameter

Im Gegensatz zu den Lageparametern beschreiben die Streuparameter nicht die absoluten Werte, sondern die Abweichung der Werte untereinander, also gewissermaßen den Grad der Zufälligkeit. Man unterscheidet absolute und relative Streuparameter. Die absoluten Streuparameter haben eine von der physikalischen Einheit des Datensatzes abgeleitete Einheit. Die relativen Streuparameter haben die Einheit 1.

2.2.6.1 Interquartilsabstand

Der **Interquartilsabstand IQR** (Inter Quartile Range) ist ein robuster absoluter Streuparameter und beschreibt die Streuung im mittleren Bereich der Daten. Er ist

definiert durch die Differenz des oberen und unteren Quartils:

$$\text{IQR} := q_{0.75} - q_{0.25}$$

Der Bruchpunkt ist $p = 0.25$.

```
> IQR(iris$Sepal.Length)

[1] 1.3

> summary(iris)

  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50

> sapply(iris[1:4], IQR)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
         1.3          0.5          3.5          1.5
```

Das Problem des IQR ist, dass er keinerlei Aussagen über die 50% der Werte trifft, die weiter von der Mitte des Datensatzes entfernt liegen.

2.2.6.1.1 Konfidenz: IQR

```
> CI.IQR = function(x, ...) UseMethod("CI.IQR", x)
> CI.IQR.data.frame = function(x, ...) sapply(x, CI.IQR,
+     ...)
> CI.IQR.list = function(x, ...) lapply(x, CI.IQR, ...)
> CI.IQR.default = function(x, ...) c(u = NA, o = NA)
> CI.IQR.numeric = function(x, ..., alpha = 0.05) {
+     O <- qlimits(x, probs = 0.75, alpha = alpha)
+     U <- qlimits(x, probs = 0.25, alpha = alpha)
+     structure(c(max(O[1] - U[2], 0), O[2] - U[1]), names = c("u",
+         "o"))
+ }
> CI.IQR(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u          0.9         0.3            3         1.1      NA
o          1.7         0.7            4         1.8      NA
```

2.2.6.2 Varianz und Standardabweichung

Die am weitesten verbreiteten absoluten Streuungsmaße sind die Varianz und ihre Wurzel die Standardabweichung. Die Varianz ist zunächst eine theoretische Größe, die als die mittlere quadratische Abweichung der Werte vom Erwartungswert definiert ist:

$$\sigma^2 := \text{var}(X) := E[(X - \mu_1)^2]$$

```
> opar = par(mfrow = c(2, 2))
> for (name in names(iris)[1:4]) {
+     x = iris[[name]]
+     hist(x, main = name)
+     abline(v = quantile(x, c(0.25, 0.75)), col = "red")
+ }
> par(opar)
```

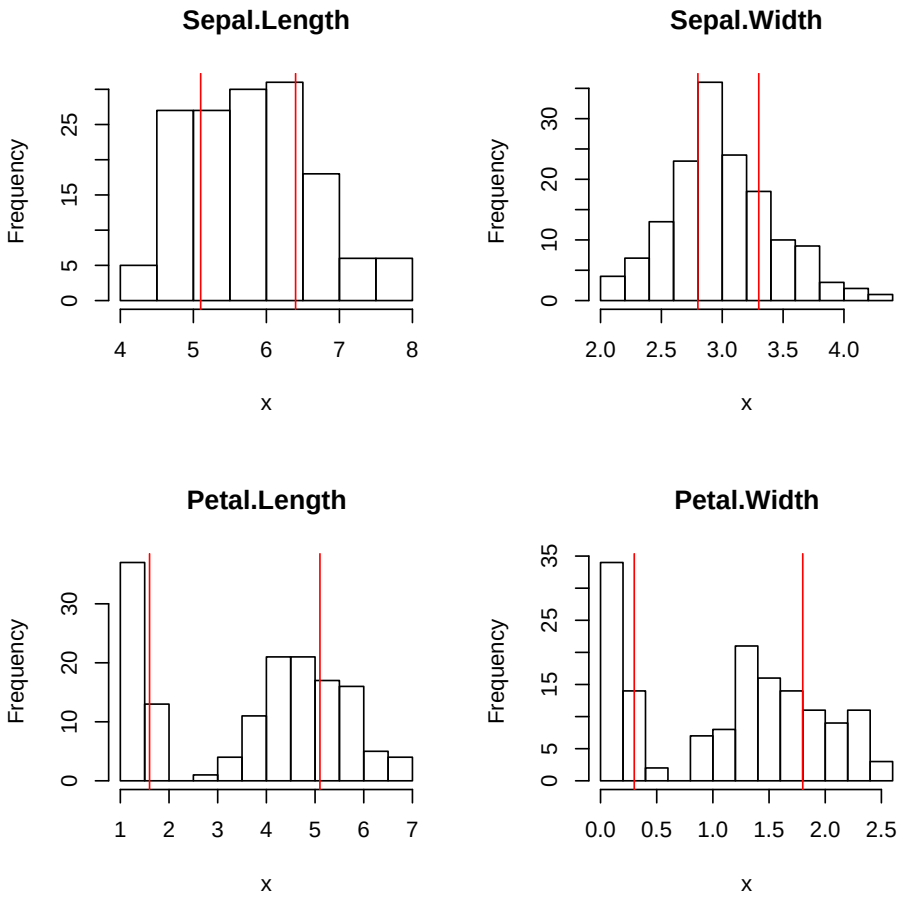


Abbildung 2.13: Interquartilsabstand
Der Interquartilsabstand gibt die Länge des Bereich an, den die mittleren 50% der Daten einnehmen.

Die Varianz kann aus einer Stichprobe erwartungstreu durch die folgende Formel geschätzt werden:

$$\hat{\sigma}^2 := \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Die Varianz ist von großer theoretischer Bedeutung. Die Varianz trägt als Einheit das Quadrat der Einheit der zugrundeliegenden Messwerte und ist für sich allein genommen schwer zu interpretieren. Deshalb verwendet man oft ihre Wurzeln, die sogenannte **Standardabweichung** σ (engl. standard deviation):

$$\sigma = sd(X) = \sqrt{\mathrm{var}(X)}$$

die durch

$$\hat{\sigma} = \sqrt{\hat{\sigma}^2}$$

geschätzt wird. Die Einheit der Standardabweichung ist dieselbe wie die der Daten.

```
> var(iris$Sepal.Length)

[1] 0.6856935

> sapply(iris[1:4], var)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.6856935    0.1899794    3.1162779    0.5810063

> sd(iris$Sepal.Length)

[1] 0.8280661

> sapply(iris[1:4], sd)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.8280661    0.4358663    1.7652982    0.7622377
```

Nach der Ungleichung von **Tschebyscheff** ist der Anteil der Grundgesamtheit, der mehr als $csd(X)$ vom Erwartungswert entfernt liegt kleiner gleich $\frac{1}{c^2}$. Also liegen z.B. mindestens $75\% = 1 - \frac{1}{2^2}$ der Grundgesamtheit im Bereich $[E[X] - 2\sigma, E[X] + 2\sigma]$. Dieser sogenannte 2σ Bereich, wird in Abbildung 2.14 dargestellt.

Gleicht die Verteilung der Daten außerdem einer Normalverteilung (Abschnitt 2.2.3), so liegen nach der 2σ**-Regel** sogar 95% der Grundgesamtheit in diesem Bereich. Es ist allerdings zu beachten (!!!), dass dies so nur für die theoretischen Werte bzw. für Schätzungen ab 62 Beobachtungen gilt.

Obwohl immer noch die klassischen Streuungsmaße, sind Standardabweichung und Varianz aus verschiedenen Gründen etwas in Verruf geraten:

- Eine einfache Interpretation ist nur im Zusammenhang mit der Normalverteilung möglich.
- Die klassischen Schätzer sind nicht robust. Robuste Schätzer lassen sich nur wieder in Verbindung mit der Annahme der Normalverteilung erstellen.

Eine robuste Schätzung kann z.B. mittels der Funktion `cov.rob` aus dem R-Paket `MASS` erzielt werden.

```
> library(MASS)
> x <- iris$Petal.Length
> cov.rob(x)
```

```
> opar = par(mfrow = c(2, 2))
> nothing = lapply(iris[1:4], function(x) {
+     m = mean(x)
+     s = sd(x)
+     hist(x, xlim = range(data.matrix(iris)), main = paste("mean=",
+         round(m, 2), " sd=", round(s, 2)))
+     abline(v = mean(x), col = "red")
+     abline(v = c(m - s, m + s), col = "green")
+     segments(m - 2 * s, 1, m + 2 * s, 1, lwd = 4, col = "blue")
+ })
> par(opar)
```

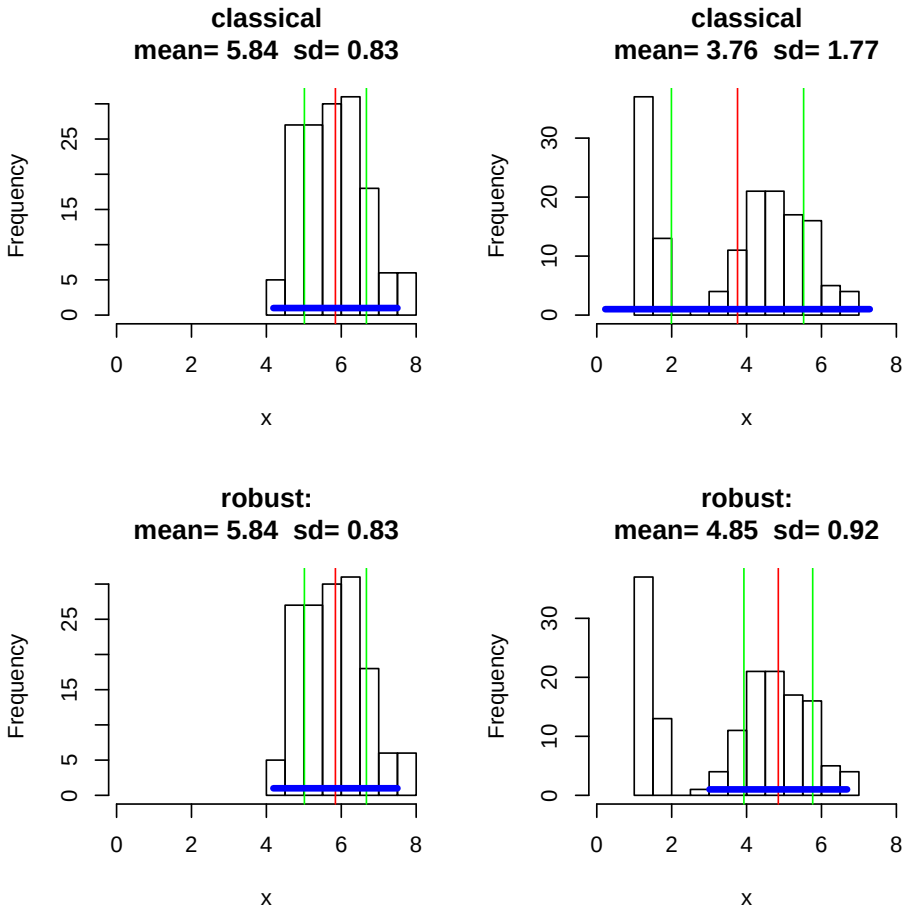


Abbildung 2.14: Kenngrößen
Mittelwert (mittlere senkrechte rote Linie), Standardabweichung (Abstand der äußeren senkrechten grünen Linien vom Mittelwert) und 2σ-Bereich (blaue horizontale Strecke).

```
$center
[1] 4.847059

$cov
          [,1]
[1,] 0.8435061

$msg
[1] "28 singular samples of size 2 out of 1000"

$crit
[1] -0.2107210

$best
 [1]  51  52  53  54  55  56  57  59  62  63  64  66  67  68  69  71
[17]  72  73  74  75  76  77  78  79  84  85  86  87  88  89  90  91
[33]  92  93  95  96  97  98 100 102 104 105 107 109 111 112 113 114
[49] 115 116 117 120 121 122 124 125 127 128 129 130 133 134 135 137
[65] 138 139 140 141 142 143 145 146 147 148 149 150

$n.obs
[1] 150
```

```
> Varianzschaetzung %=% cov.rob(x)$cov

          [,1]
[1,] 0.8435061
```

```
> var(x)

[1] 3.116278
```

```
> RobusteMittelwertschaetzung %=% cov.rob(x)$center

[1] 4.847059
```

```
> mean(x)

[1] 3.758
```

```
> Standardabweichung %=% sqrt(cov.rob(x)$cov)

          [,1]
[1,] 0.9184259
```

Diese Funktion liefert gleich noch eine robuste Mittelwertsschätzung. Der Bruchpunkt ist standardmäßig etwas kleiner als $\frac{1}{2}$. Für eine genauere Beschreibung, z.B. was welche Ausgabe bedeutete oder wie man den Bruchpunkt festlegen kann, konsultieren Sie bitte die Anleitung mit dem Befehl `? cov.rob`

Eine robuste Schätzung bedeutet letztlich, dass man nicht den ganzen Datensatz beschreibt, sondern nur seinen "Hauptteil". Das hat gewisse Vorteile:

- Die Ergebnisse werden nicht von Datenfehlern oder singulären Spezialfällen (z.B. Landsitz des Multimillionärs im Bergdorf) beeinflusst.

  ```
  > x <- rnorm(30)
  > x
  ```

```
> opar = par(mfcol = c(2, 2))
> lapply(iris[c(1, 3)], function(x) {
+     m = mean(x)
+     s = sd(x)
+     hist(x, xlim = range(data.matrix(iris)), main = paste("classical\nmean=",
+         round(m, 2), " sd=", round(s, 2)))
+     abline(v = m, col = "red")
+     abline(v = c(m - s, m + s), col = "green")
+     segments(m - 2 * s, 1, m + 2 * s, 1, lwd = 4, col = "blue")
+     robust = cov.rob(x)
+     m = robust$cente
+     s = sqrt(robust$cov)
+     hist(x, xlim = range(data.matrix(iris)), main = paste("robust:\nmean=",
+         round(m, 2), " sd=", round(s, 2)))
+     abline(v = m, col = "red")
+     abline(v = c(m - s, m + s), col = "green")
+     segments(m - 2 * s, 1, m + 2 * s, 1, lwd = 4, col = "blue")
+ })

$Sepal.Length
NULL

$Petal.Length
NULL

> par(opar)
```

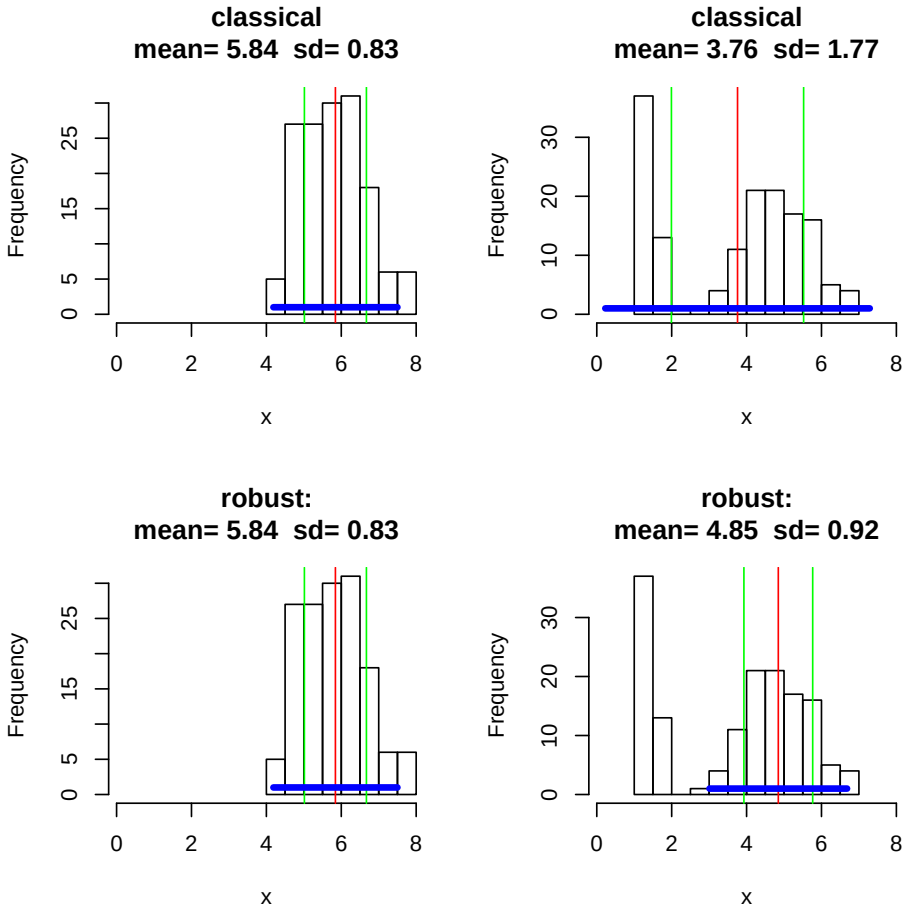


Abbildung 2.15: Eigenschaften von Kenngrößen
Mittelwert, Standardabweichung und 2σ-Bereich für Sepal.Length (erste Spalte) und Petal.Length (zweite) Spalte und zwar einmal mit klassischer Schätzung (erste Zeile) und einmal mit robuster Schätzung (zweite Zeile). Während sich

```
 [1]  0.13675044 -1.04147838  1.11204366  1.32561645 -0.60355933
 [6] -0.51080941 -0.57466079 -1.08236091  0.19032709 -1.15046392
[11] -1.18484601  0.28110761 -0.93531876  0.28357949 -0.78629560
[16] -0.56080590 -0.57901107 -1.35482181 -0.09000389  1.52268984
[21] -0.69630877  0.50634126 -2.26617873  1.01396437  0.67198659
[26]  0.97350015 -2.22254026 -1.28729222  0.20026056 -0.94101993

> var(x)

[1] 0.9715575

> cov.rob(x, method = "mcd")$cov

          [,1]
[1,] 0.9715575

> x[1] <- 10000
> var(x)

[1] 3333559

> cov.rob(x, method = "mcd")$cov

          [,1]
[1,] 0.9984924

> myVarRob <- function(x, ...) UseMethod("myVarRob")
> myVarRob.list <- function(x, ...) lapply(x, myVarRob,
+     ...)
> myVarRob.data.frame <- function(x, ...) sapply(x, myVarRob,
+     ...)
> myVarRob.default <- function(x, ...) NA
> myVarRob.numeric <- function(x, p = 0.5, nsamp = 1000,
+     ...) {
+     n <- length(x)
+     m <- floor(n * (1 - p))
+     if (p < 0.5 || p < 0.2)
+         warning("Bruchpunkt sollte zwischen 0.2 und 0.5 liegen in myVarRob")
+     fci <- function(x) {
+         x <- sort(x)
+         min(x[(m + 1):n] - x[1:(n - m)])
+     }
+     fci(x)/exp(mean(sapply(1:nsamp, function(i) log(fci(rnorm(n))))))
+ }
> myVarRob(x)

[1] 1.126765
```

- Spezialfälle können leichter sichtbar gemacht werden, da Sie sich um deutlich mehr als die geschätzte Standardabweichung vom robusten Mittelwert entfernt auffinden lassen.

```
> s <- sqrt(myVarRob(x))
> m <- median(x)
> x[abs(x - m)/s > 2]

[1] 10000
```

aber auch Nachteile:

- Robuste Schätzungen liefern unter Umständen unrealistisch kleine Streuungen.
- Robuste Schätzungen sind ungenauer, wenn die Datenqualität gut ist.
- Eine robuste Schätzung beschreibt nicht die tatsächliche Varianz in den Daten, sondern nur die Varianz in dem "gutartigen" "schönen" Teil der Daten. Aber die extremen "Ausreißerwerte" können natürlich nicht nur in der Stichprobe, sondern auch in der Grundgesamtheit selbst vorkommen.

2.2.6.2.1 Konfidenz: Konfidenzintervall für die Varianz und Standardabweichung Ein Konfidenzintervall basierend auf robusten Schätzungen ist hier nicht sinnvoll, da dann ja nicht klar wird, was "richtige" Daten und was "falsche" Daten sind. Für die hier angegebenen Konfidenzintervalle muss allgemein Normalverteilung vorausgesetzt werden:

```
> CInorm.var = function(x, ...) UseMethod("CInorm.var",
+     x)
> CInorm.var.data.frame = function(x, ...) sapply(x, CInorm.var,
+     ...)
> CInorm.var.list = function(x, ...) lapply(x, CInorm.var,
+     ...)
> CInorm.var.default = function(x, ...) c(u = NA, o = NA)
> CInorm.var.numeric = function(x, ..., alpha = 0.05) {
+     x = x[!is.na(x)]
+     n <- length(x)
+     v <- var(x, ...)
+     c(u = v * qchisq(alpha/2, n - 1), o = v * qchisq(1 -
+         alpha/2, n - 1))
+ }
> CInorm.var(split(iris, iris$Species))
```

```
$setosa
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     3.920666    4.534120    0.9516705   0.3504528      NA
o     8.725063   10.090244    2.1178507   0.7798987      NA

$versicolor
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u      8.40726    3.107193     6.967841    1.233990      NA
o     18.70954    6.914758    15.506255    2.746126      NA

$virginica
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     12.75901    3.281840     9.611241    2.380271      NA
o     28.39393    7.303418    21.388887    5.297063      NA
```

```
> CInorm.sd = function(x, ...) UseMethod("CInorm.sd", x)
> CInorm.sd.data.frame = function(x, ...) sapply(x, CInorm.sd,
+     ...)
> CInorm.sd.list = function(x, ...) lapply(x, CInorm.sd,
+     ...)
> CInorm.sd.default = function(x, ...) c(u = NA, o = NA)
> CInorm.sd.numeric = function(x, ..., alpha = 0.05) sqrt(CInorm.var.numeric(x,
+     ..., alpha = alpha))
> CInorm.sd(split(iris, iris$Species))
```

```
$setosa
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     1.980067    2.129347     0.975536   0.5919905      NA
```

```
o     2.953822    3.176514     1.455284   0.8831187      NA

$versicolor
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     2.899528    1.762723     2.639667    1.110851      NA
o     4.325453    2.629593     3.937798    1.657144      NA

$virginica
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     3.571975    1.811585     3.100200    1.542813      NA
o     5.328596    2.702484     4.624812    2.301535      NA
```

```
> CInorm.myVarRob <- function(x, ...) UseMethod("CInorm.myVarRob")
> CInorm.myVarRob.data.frame <- function(x, ...) sapply(x,
+     CInorm.myVarRob, ...)
> CInorm.myVarRob.list <- function(x, ...) lapply(x, CInorm.myVarRob,
+     ...)
> CInorm.myVarRob.default <- function(x, ...) c(NA, NA)
> CInorm.myVarRob.numeric <- function(x, p = 0.5, nsamp = ceiling(100 *
+     alpha), ..., alpha = 0.05) {
+     n <- length(x)
+     m <- floor(n * p)
+     fci <- function(x) {
+         x <- sort(x)
+         min(x[(m + 1):n] - x[1:(n - m)])
+     }
+     a <- sapply(1:nsamp, function(i) log(fci(rnorm(n))))
+     structure(c(fci(x)/quantile(a, c(1 - alpha/2, alpha/2))),
+         names = c("u", "o"))
+ }
> CInorm.myVarRob(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     3.972027    1.090731     4.091402    2.845597      NA
o    25.774663    3.557792    11.848115   23.177170      NA
```

```
> CInorm.mySdRob <- function(x, ...) UseMethod("CInorm.mySdRob")
> CInorm.mySdRob.data.frame <- function(x, ...) sapply(x,
+     CInorm.mySdRob, ...)
> CInorm.mySdRob.list <- function(x, ...) lapply(x, CInorm.mySdRob,
+     ...)
> CInorm.mySdRob.default <- function(x, ...) c(NA, NA)
> CInorm.mySdRob.numeric <- function(x, ...) sqrt(CInorm.myVarRob.numeric(x,
+     ...))
> CInorm.mySdRob(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
u     1.997886    1.392743     2.509865    1.839720      NA
o     2.756179    2.418847     3.249949    2.548234      NA
```

2.2.6.3 Bereich

Ein rein beschreibendes Merkmal ist der **Bereich** (engl. **range**) der Daten, der gegeben ist durch den minimalen und maximalen beobachteten Wert:

```
> range(iris$Sepal.Length)
```

```
[1] 4.3 7.9
```

```
> sapply(iris[1:4], range)
```

```
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]          4.3         2.0          1.0         0.1
[2,]          7.9         4.4          6.9         2.5
```

Obwohl der Bereich eine, im ersten Moment, sehr ansprechend erscheinende Größe ist, da er eine exakte Information über die Daten liefert, erlaubt der Bereich fast keine Rückschlüsse auf die Grundgesamtheit. Das kann man leicht an einem Simulationsbeispiel zeigen, das die starke Streuung des Bereichs verdeutlicht und aufzeigt, dass der Bereich von der Stichprobengröße abhängt:

```
> replicate(20, range(rnorm(10, mean = 3, sd = 1)))

         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
[1,] 1.234964 1.092057 1.547835 1.554017 2.325640 1.359551 1.944572
[2,] 5.390186 4.332009 4.929457 4.068817 5.042938 3.799693 4.114209
         [,8]      [,9]   [,10]    [,11]    [,12]    [,13]    [,14]
[1,] 2.158255 0.3090875 1.82497 1.581791 1.736060 2.364467 1.251481
[2,] 3.286070 3.4838273 5.46488 4.070146 4.708294 4.162400 4.110101
         [,15]    [,16]    [,17]    [,18]    [,19]    [,20]
[1,] 0.8063462 1.548887 0.868764 2.594959 1.356853 1.081543
[2,] 3.4509251 4.619134 4.266619 4.520601 4.329364 3.772671

> replicate(20, range(rnorm(100, mean = 3, sd = 1)))

           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] -0.1185282 0.3689196 0.8286576 0.4330661 0.5356555 0.4162561
[2,]  5.1820795 5.6978401 5.1179679 5.3781013 5.1550217 5.8636356
          [,7]        [,8]       [,9]     [,10]      [,11]     [,12]
[1,] 0.1459199 -0.04524303 -0.1848459 0.3250903 -0.9142264 0.6949834
[2,] 4.6852213  6.31646917  5.4584060 4.8810193  5.1387524 6.6585720
          [,13]       [,14]     [,15]     [,16]     [,17]      [,18]
[1,] -0.6702272 0.002385191 0.5858787 0.1346648 0.4444836 -0.1025590
[2,]  6.0722694 5.485726629 5.7049779 4.9794228 5.6805101  5.6531850
          [,19]      [,20]
[1,] 0.02194517 -0.5709292
[2,] 5.64655913  5.5572263
```

Insbesondere kann ein einzelner Ausreißer das Ergebnis stark verfälschen.

2.2.6.4 Relative Streuparameter

Relative Streuparameter braucht man eigentlich immer dann, wenn man mit relativen Skalen arbeitet und sie sind im Prinzip auch nur für diese definiert. Sie treffen eine Aussage darüber, wie stark ein Wert im Verhältnis schwankt. So würde beispielsweise die Information, dass eine Waage eine Messgenauigkeit von 100g hat, erst einmal den Eindruck erwecken, die Waage wäre ungenau. Erst wenn man erfährt, dass die Waage für LKW's bestimmt ist, würde man seine Meinung ändern. Hätte man von Anfang an gesagt: "Die Waage wiegt auf ein hundertstel Promill genau", so hätte man sofort gewusst, dass diese Waage phänomenal genau ist. Relative Streuparameter sind solche einheitenlosen Parameter, die ohne Kenntnis der absoluten Zahlenwerte sofort interpretierbar sind.

Leider wird der relativen Skala meist sehr wenig Beachtung geschenkt, so dass außer dem Variationskoeffizienten praktisch keines dieser Maße einen, quer durch die Wissenschaftslandschaft, einheitlichen Namen besitzt. Man muss daher immer dazusagen, wie man den relativen Streuparameter genau berechnet hat.

Im Prinzip gibt es zwei Konstruktionsprinzipien für relative Streuparameter:

1. Man teilt einen absoluten Streuparameter einfach durch den einen Lageparameter.

   - z.B. der Variationskoeffizient:

     $$v(x) := \frac{sd(x)}{\bar{x}}$$

   - mediannormierter Inter-Quartile-Range

     $$\frac{IQR(x_1, \ldots, x_n)}{median(x_1, \ldots, x_n)}$$

   Dieser Ansatz führt nur selten zu interpretierbaren Streuparametern. Er ist trotzdem weit verbreitet.

2. Man wendet einen absoluten Streuparameter einfach auf den logarithmierten Datensatz an. Als einheitliche Notation markiert man solche Parameter mit einem hochgestellten $*$ und bezeichnet sie als Parameter, weil sie die Streuung in der logarithmischen Metrik der relativen Skala messen.

   - metrische Standardabweichung

     $$sd^*(X) = sd(log(X))$$

     Eine metrische Standardabweichung interpretiert sich am leichtesten, wenn man $fsd(X) := exp(2 * sd^*(X))$ berechnet, denn dies ist der Faktor, durch den man aus dem geometrischen Erwartungswert durch Multiplikation und dividieren ein "relatives" 2σ-Intervall bekommt, das dann ähnlich zu interpretieren ist, wie bei der gewöhnlichen Standardabweichung. Die folgende Tabelle mag da eine Orientierungshilfe sein:

   - metrische Varianz

     $$\text{var}^*(X) = \text{var}(log(X))$$

     Wie die gewöhnliche Varianz ist auch die metrische Varianz eher von theoretischem Interesse.

   - metrischer IQR

     $$IQR^*(X) = IQR(log(X))$$

     Für die Interpretation des IQR^* bietet sich ähnlich, wie bei der Standardabweichung an, zu $fIQR = exp(IQR^*)$ überzugehen, denn das ist der Faktor, der das 1.Quartil der Daten vom 3.ten Quartil unterschiedet.

   Die Einheit der metrischen Streuparameter ist 1.

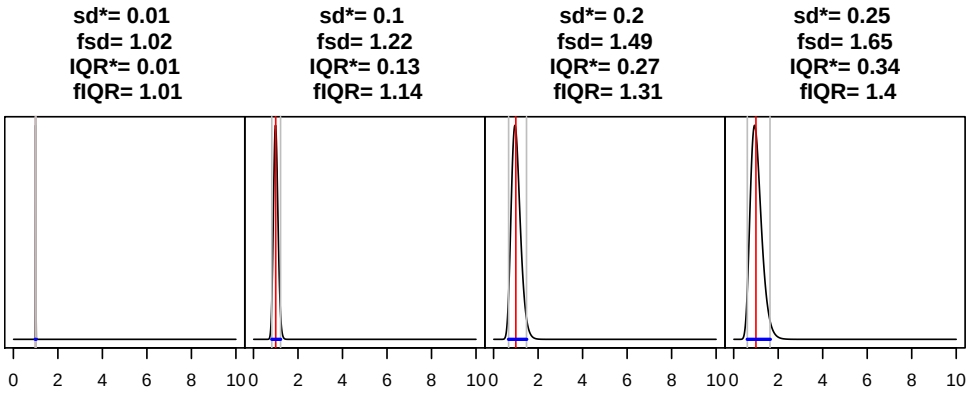
2.2.7 Boxplot oder Kastendiagramm

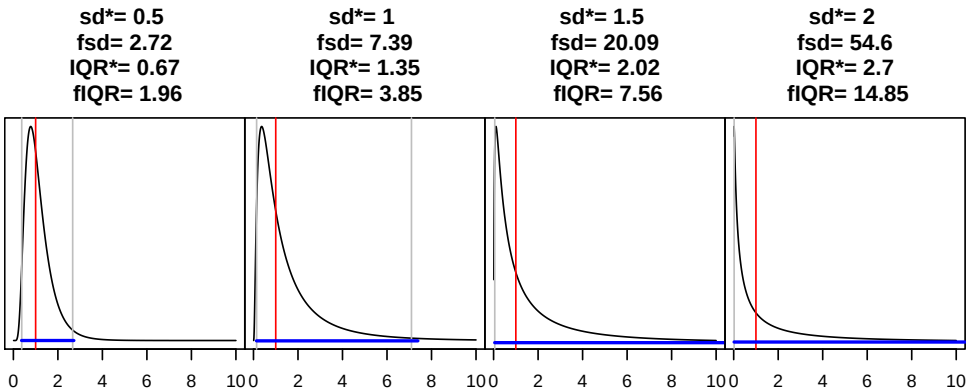
Ein Zwischending zwischen der rein datenzentrierten Darstellung im Punktdiagramm und der rein quantitativen Darstellung durch einen Parameter bildet der **Boxplot** (auch **Kastendiagramm** genannt).

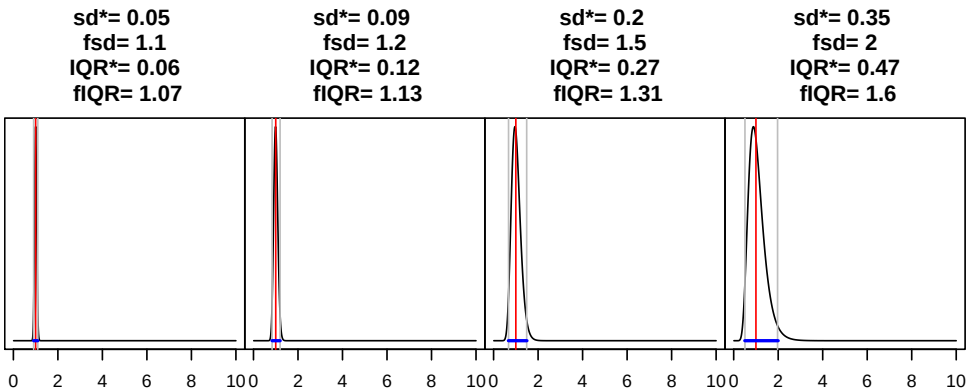
2.2.7.1 Aufbau eines Boxplot

Der genaue Aufbau des Boxplot wird in Abbildung 2.17 erläutert.

```
> opar = par(mfrow = c(4, 4), mar = c(2, 0, 6, 0), yaxt = "n")
> demostar <- function(s) {
+     x <- seq(0.01, 10, by = 0.01)
+     f <- dlnorm(x, 0, s)
+     d <- log(qlnorm(c(0.25, 0.75), 0, s))
+     d <- d[2] - d[1]
+     plot(x, f, type = "l", main = paste("sd*=", round(s,
+         2), "\n", "fsd=", round(exp(2 * s), 2), "\n",
+         "IQR*=", round(d, 2), "\n", "fIQR=", round(exp(d),
+             2)))
+     abline(v = 1, col = "red")
+     abline(v = qlnorm(c(0.025, 0.975), 0, s), col = "gray")
+     segments(1/exp(2 * s), 0, exp(2 * s), 0, col = "blue",
+         lwd = 2)
+     s
+ }
> invisible(sapply(c(sort(c(0.01, 0.1, 0.2, 0.25, 0.5,
+     1, 1.5, 2)), sort(log(c(1.1, 1.2, 1.5, 2, 3, 5, 10,
+     20))/2)), demostar))
> par(opar)
```







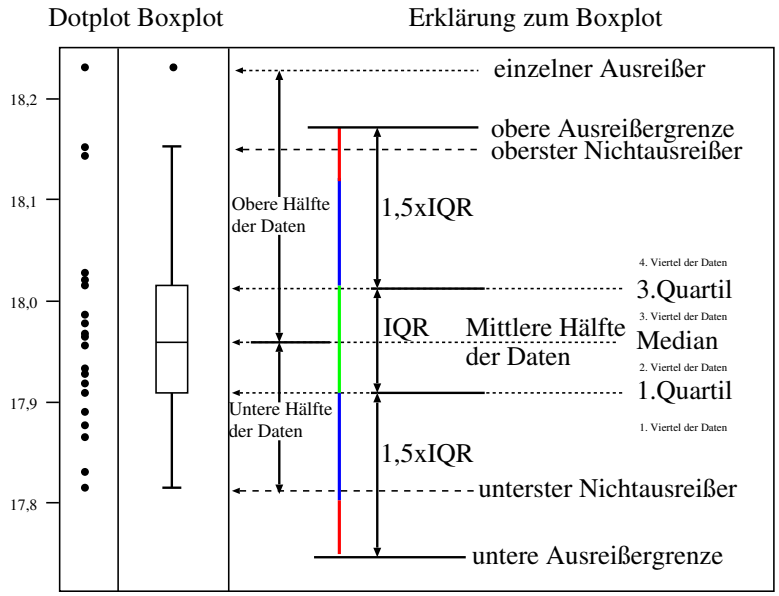
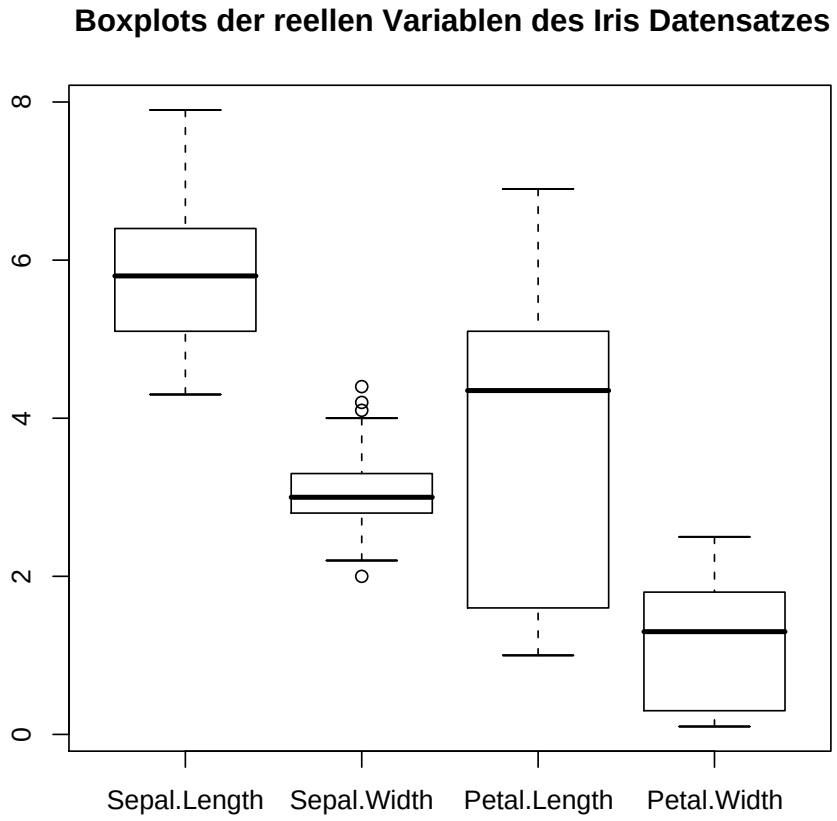


Abbildung 2.17: Definition des Boxplot

2.2.7.1.1 R: Boxplots erzeugen In R erzeugt man einen Boxplot einfach durch Übergabe der Daten an den Boxplotbefehl:

```
> boxplot(iris[1:4], main = "Boxplots der reellen Variablen des Iris Datensatzes")
```



Der Boxplot versucht die wichtigsten Informationen über die Verteilung so kompakt wie möglich darzustellen und dabei robust gegen Ausreißer zu bleiben:

- Lage (durch den Median)
- Streuung (durch den IQR (Inter Quartile Range))
- Symmetrie (durch die Symmetrie der Box)
- Besondere Werte (durch eine gesonderte Darstellung von Ausreißern)
- Bereich (durch die Zäune)

Der Boxplot besteht aus einem Rechteck – der **Box** – das vom unteren bis zum oberen Quartil des Datensatzes reicht, es umschließt also die mittlere Hälfte der Daten. Ein Strich auf der Höhe des Stichprobenmedian teilt die Box in einen oberen und unteren Bereich, die jeweils ein Viertel der Daten enthalten.

Die Höhe der Box entspricht also dem IQR, der ja auch als robuste Kennzahl für die Streuung der Daten verwendet wird. Basierend auf dieser Streuungsschätzung und einer Normalverteilungsannahme als Referenzmodell sollen nun Ausreißer erkannt werden. Dabei werden all diejenigen Beobachtungen, die mehr als 1,5 IQR von der Box entfernt liegen als **Ausreißer** betrachtet. Die Ausreißer werden wie im Punktdiagramm einzeln dargestellt. Dabei werden zwei verschiedene Symbole verwendet: Ein einfaches Symbol für näher liegende Ausreißer, und ein auffälligeres

Symbol für die sogenannten **fernen Ausreißer** die mehr als 2,5 IQR von der Box entfernt liegen.

Anschließend wird der kleinste und größte nicht als Ausreißer markierte Wert als Begrenzung des Bereichs der "guten" Daten mit kurzen waagrechten Strichen, den sogenannten **Zäunen**, angezeichnet. Die Zäune werden mit der Box durch eine Strecke verbunden. Werden also keine Ausreißer erkannt, so markiert der Bereich vom unteren bis zum oberen Zaun den Bereich der Daten.

2.2.7.2 Interpretation des Boxplot

Bei jeder Interpretation des Boxplots sollte man bedenken, dass der Boxplot bei kleinen Stichprobengrößen noch relativ ungenau ist.

- *Ausreißer*
  Ausreißer werden im Boxplot direkt dargestellt.

- *Stichprobenlage / Median*
  Mit dem Median wird ein robuster Lageparameter der Stichprobe dargestellt.

- *Ablesen der Stichprobenstreuung*
  Die Höhe der Box entspricht dem IQR (Interquartilsabstand), der einen robusten empirischen Streuungsparameter darstellt.

- *Symmetrie und Schiefe der Verteilung*
  Bei einer symmetrischen Verteilung würde man erwarten, dass der Median die Box in zwei ähnlich große Bereiche teilt. Eine schiefe Verteilung kann man also oft daran erkennen, dass diese beiden Bereiche verschieden groß sind. Entsprechendes gilt für die Zäune. Allerdings sind die Zäune als Schätzwerte für den Bereich deutlich stärker gestreut und nicht quantitativ interpretierbar. Ist die obere Box größer, so deutet das auf eine rechtsschiefe Verteilung hin. Ist die untere Box größer, so deutet das auf eine linksschiefe Verteilung hin.

- *extreme Werthäufungen*
  Tritt ein einzelner Wert sehr häufig (z.B. bei der Hälfte aller Wert) auf, so fallen häufig einige Quantile des Boxplot zusammen. Für kleinere Gruppen von Wertwiederholungen oder Häufungen ist der Boxplot jedoch blind. Diese können leichter in gestapelten oder verzitterten Punktdiagrammen erkannt werden.

- *Vergleich der Stichprobenlage*
  Da der Boxplot den Lageparameter Median direkt darstellt, können mit nebeneinander gezeichneten Boxplots (sogenannte parallele Boxplots) die Lage mehrerer Stichproben verglichen werden. Um jedoch eine quantitative Aussage treffen zu können, setzt man dann **gekerbte Boxplots** ein, die um den Median herum eine Einkerbung haben, die einen Konfidenzbereich für den Median darstellt. Der parallele und der gekerbte Boxplot wird im Detail im Abschnitt 2.6.2 besprochen.

Weitere Details z.B. zu

- genauer Verteilungsform
- Anzahl der Beobachtungen
- Werthäufungen
- Bindungen

```
> opar <- par(mfcol = c(4, 4), pch = 20, mar = c(2, 2,
+     2, 2))
> for (n in c(2, 4, 8, 16, 32, 64, 128, 256)) {
+     x <- rnorm(n)
+     boxplot(x, points = TRUE, main = paste(n, " Datenpunkte"),
+         ylim = c(-4, 4))
+     hist(x, main = "", xlim = c(-4, 4))
+ }
> par(opar)
```

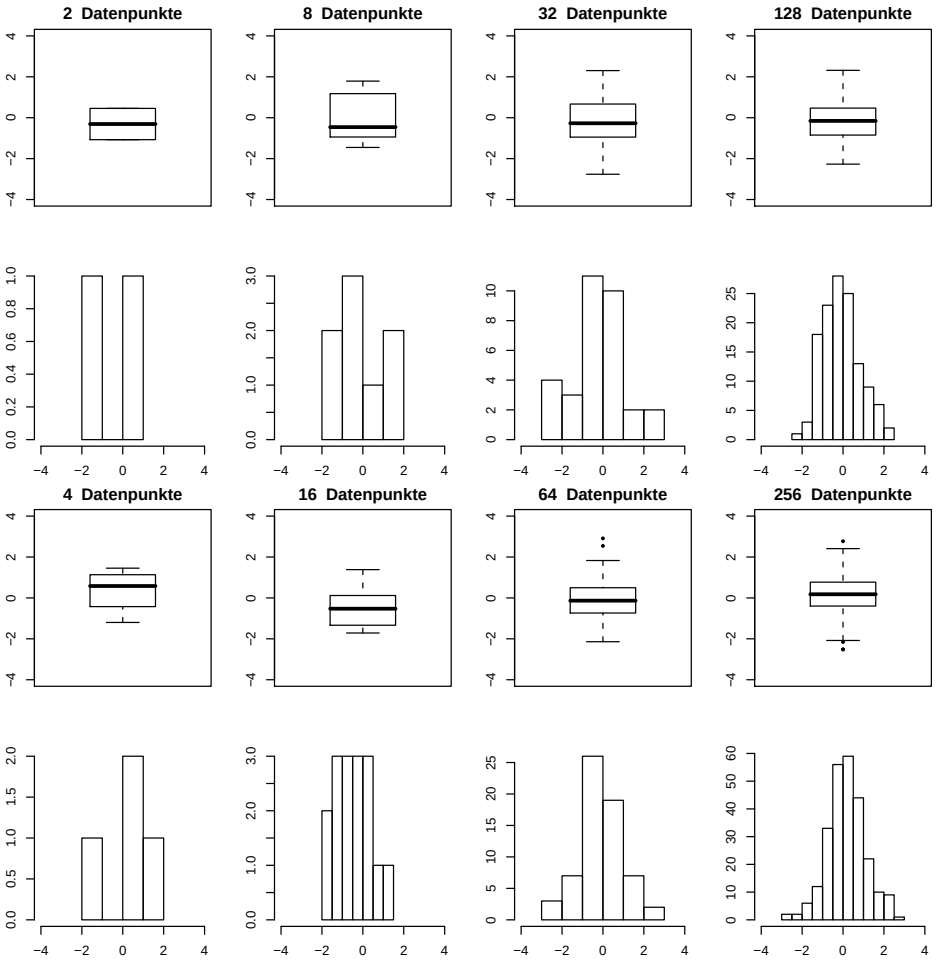


Abbildung 2.18: Vergleich Boxplot vs. Histogramm
Aussagekraft von Boxplots und Histogrammen im Vergleich. Der Boxplot hat insbesondere im Bereich kleinerer Stichproben eine deutlich höhere Aussagekraft als das Histogramm, da sich das Histogramm erst für größere Stichproben stabilisiert. Bei sehr kleinen Stichproben ist aber, wie man sieht auch der Boxplot sehr instabil.

- Anzahl und Lage der Modalwerte
- Mittelwert
- Varianz

können im Boxplot nicht! abgelesen werden. Überlegen Sie, mit welchen Methoden diese Werte zu erkennen sind.

2.2.8 QQ-Plot

Zum direkten Vergleich von Beobachtungswerten mit einem Verteilungsmodell oder der Verteilungen eines zweiten Datensatzes eignet sich der **Quantils-Quantils-Plot** oder **QQ-Plot**.

Beim Vergleich mit einer theoretischen Verteilung werden die Datenwerte, die ja empirischen Quantilen zu Wahrscheinlichkeiten $p = \frac{i-0.5}{n}$ entsprechen, gegen die entsprechenden theoretischen Quantile der Verteilung in ein Streudiagramm aufgetragen. Die theoretischen Quantile werden dabei auf der x-Achse dargestellt. Es werden also die theoretischen Quantile der Verteilung auf der x-Achse gegen die empirischen Quantile auf der y-Achse aufgetragen. Daher heißt der Plot ein Quantils-Quantils-Plot.

Beim Vergleich zweier empirischer Verteilungen werden statt der theoretischen Quantile die entsprechenden empirischen Quantile der zweiten Verteilung verwendet.

Da sowohl bei den empirischen als auch bei den theoretischen Quantilen zu größeren p auch jeweils größere Quantile gehören, ergibt sich immer eine monoton steigende Punktfolge. Das ist ein Hinweis, dass man kein gewöhnliches Streudiagramm sieht, sondern einen QQ-Plot.

2.2.8.0.1 R: QQ-Plots Der QQ-Plot für die Normalverteilung ist in R direkt implementiert (`qqnorm`) (Abb. 2.19):

```
> opar <- par(mfrow = c(2, 2))
> for (Var in names(iris[1:4])) qqnorm(iris[[Var]], main = Var)
> par(opar)
```

Es fehlt allerdings die Implementierung für eine allgemeine Lokations-Skalen Familie. Daher ist mit `QQplot` eine einfache Implementierung angegeben (Abb. 2.20):

```
> x <- iris$Petal.Width[iris$Species == "virginica"]
> QQplot <- function(x, q, ..., xlab = substitute(q)) qqplot(q(ppoints(length(x))),
+     x, ..., xlab = xlab)
> opar <- par(mfrow = c(2, 2))
> QQplot(x, qnorm)
> QQplot(x, qlnorm)
> QQplot(x, qunif)
> QQplot(x, qcauchy)
> par(opar)
```

Die QQ-Plot für zwei Datensätze (`qqplot()`) ist wieder direkt vorhanden (Abb. 2.21):

```
> qqplot(iris$Sepal.Length, iris$Petal.Length)
```

Bei einer guten Übereinstimmung der theoretischen Verteilung mit der empirischen sollten die beiden Quantilreihen sich in etwa entsprechen und die Punkte somit in der Nähe der durch $x = y$ gegebenen Geraden liegen. Offenbar ist das bei keiner der obigen Graphiken der Fall. Allerdings eignet sich der QQ-Plot nicht nur zum Vergleich mit einer einzelnen Verteilung, sondern mit einer ganzen sogenannten Lokations-Skalen-Familie von Verteilungen.

```
> opar <- par(mfrow = c(2, 2))
> for (Var in names(iris[1:4])) qqnorm(iris[[Var]], main = Var)
> par(opar)
```

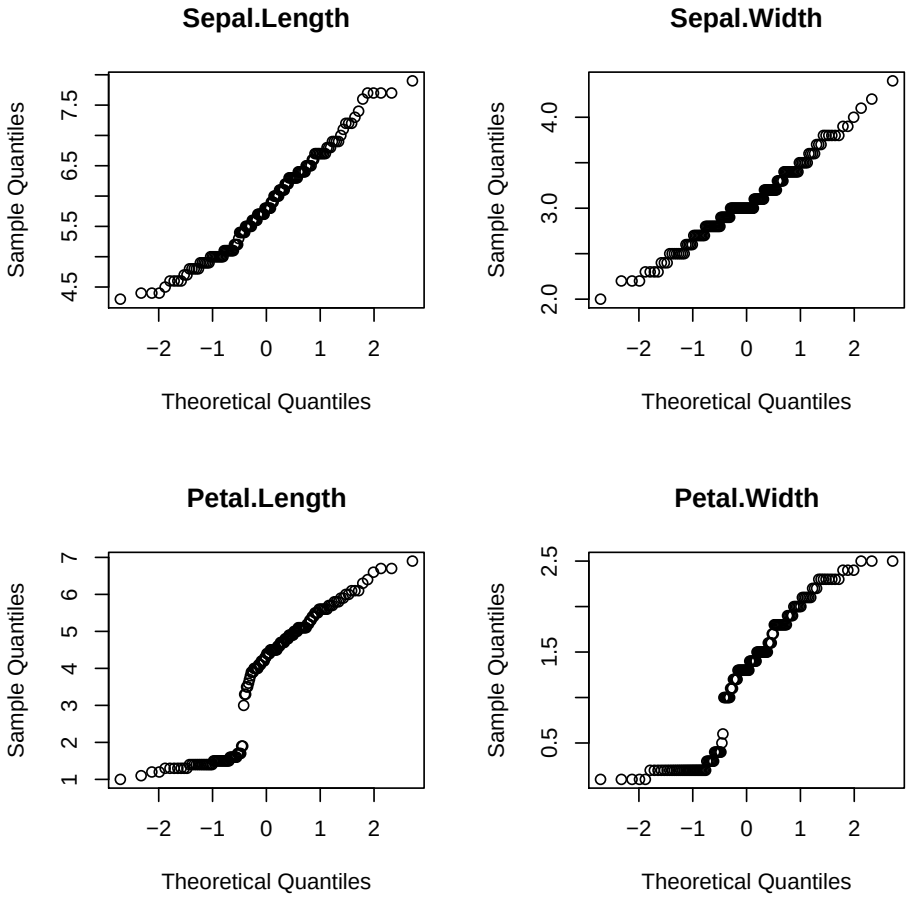


Abbildung 2.19: Normal-QQ-Plot
QQ-Plot zum Vergleich mit der Normalverteilung

```
> x <- iris$Petal.Width[iris$Species == "virginica"]
> QQplot <- function(x, q, ..., xlab = substitute(q)) qqplot(q(ppoints(length(x))),
+     x, ..., xlab = xlab)
> opar <- par(mfrow = c(2, 2))
> QQplot(x, qnorm)
> QQplot(x, qlnorm)
> QQplot(x, qunif)
> QQplot(x, qcauchy)
> par(opar)
```

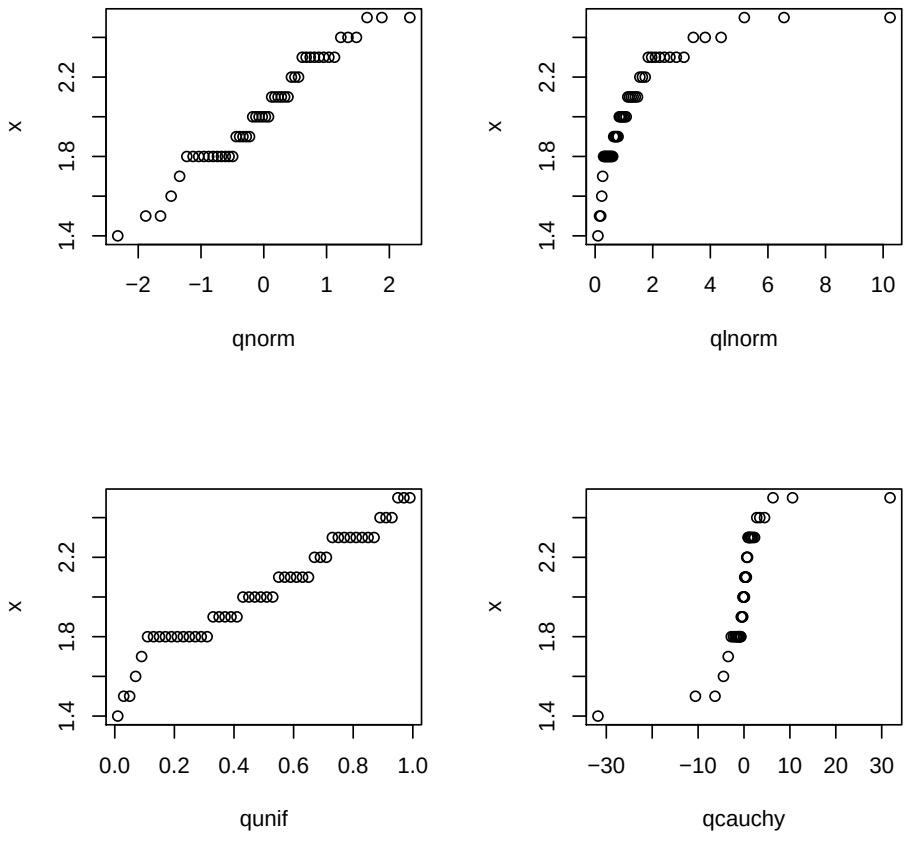


Abbildung 2.20: Allgemeiner QQ-Plot
QQ-Plot zum Vergleich mit verschiedenen Verteilungsfamilien.

```
> qqplot(iris$Sepal.Length, iris$Petal.Length)
```

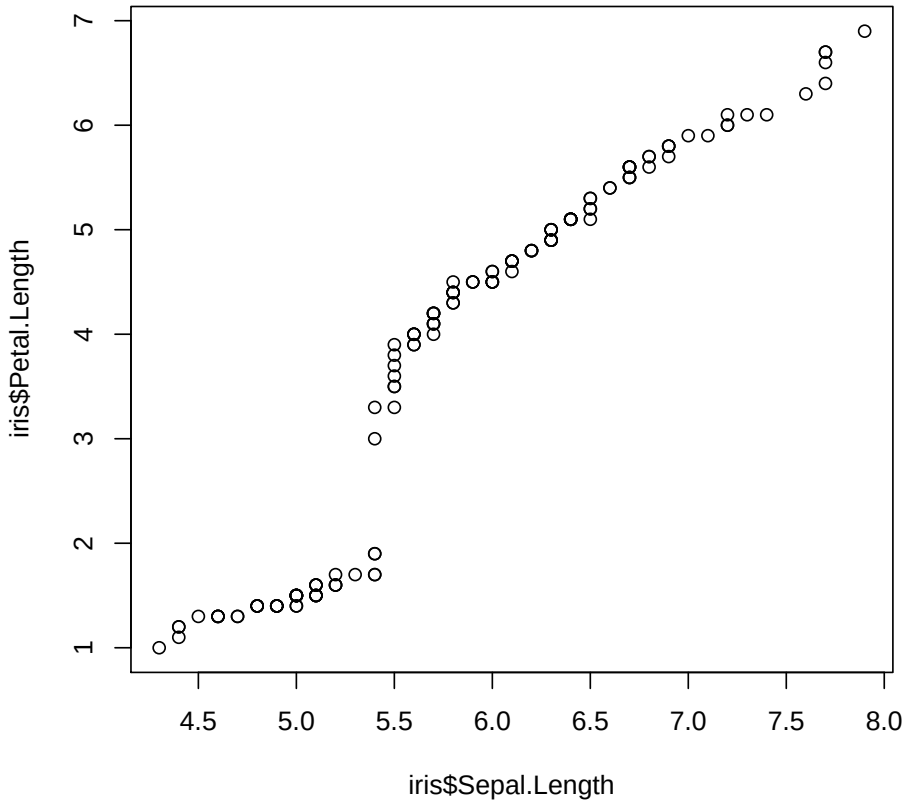


Abbildung 2.21: QQ-Plot zum Vergleich zweier Stichproben

Definition 28 *Die zu einer Verteilungsfunktion $F(x)$ gegebene und durch die Parameter $\mu \in \mathbb{R}$ und $s \in \mathbb{R}^+$ indizierte Familie und durch die Verteilungsfunktion*

$$F_X(x) = F\left(\frac{x-\mu}{s}\right)$$

gegebenen Verteilungen $P^X_{\mu,\sigma}$, heißt die Lokations-Skalen-Familie von F. Dabei heißt μ der Lageparameter und s der Skalenparameter der Familie.

Praktisch ist eine Lokations-Skalen-Familie ein Verteilungsmodell, in dem alle verschobenen und skalierten Versionen einer Standardverteilung liegen.

Beispiel 29

- *Die Normalverteilung $N(\mu, \sigma^2)$ ist die Lokations-Skalen-Familie zu der Verteilungsfunktion der Standardnormalverteilung $N(0,1)$ und dem Lokationsparameter μ und dem Skalenparameter σ.*
- *Die Gleichverteilung $Unif(a,b)$ auf dem Intervall $[a,b]$ ist die Lokations-Skalen-Familie zu der Verteilungsfunktion der Gleichverteilung $U(0,1)$ auf dem Einheitsintervall mit dem Lokationsparameter $\mu = \frac{a+b}{2}$ und dem Skalenparameter $s = b - a$.*
- *Die Cauchyverteilung $C(l,s)$ mit Lageparameter l und Skalenparameter s ist eine Lokations-Skalen-Familie zu der Verteilungsfunktion der Cauchyverteilung $C(0,1)$.*

Beim Vergleich mit einer Lokations-Skalen-Familie verwendet man die Referenzverteilung $P_{0,1}$.

2.2.8.1 Interpretation von QQ-Plots

```
> opar <- par(mfrow = c(3, 3))
> qqnorm(rnorm(10), main = "Genaue Verteilung", sub = "a")
> abline(0, 1)
> qqnorm(rnorm(100), main = "Genaue Verteilung", sub = "b")
> abline(0, 1)
> qqnorm(rnorm(1000), main = "Genaue Verteilung", sub = "c")
> abline(0, 1)
> qqnorm(rnorm(100, 0.5, 0.5), main = "Lokationsskalenfamilie",
+     sub = "d")
> abline(0, 1)
> abline(0.5, 0.5)
> qqnorm(c(rnorm(99), 6), main = "Ausreisser", sub = "e")
> abline(0, 1)
> qqnorm(rcauchy(100), main = "Schwere Schwaenze", sub = "f")
> abline(0, 1)
> qqnorm(runif(100, -1, 1), main = "Leichte Schwaenze",
+     sub = "g")
> abline(0, 0.7)
> qqnorm(rlnorm(100, 0, 0.6), main = "Schiefe", sub = "h")
> abline(1, 1)
> qqnorm(round(rnorm(100, 4, 3)), main = "Rundung und Bindungen",
+     sub = "i")
> abline(4, 3)
> par(opar)
```

```
> opar <- par(mfrow = c(2, 2))
> x <- seq(-2, 8, by = 0.01)
> plot(x, dnorm(x, mean = 0, sd = 1), ylim = c(0, 0.5),
+     type = "l")
> plot(x, dnorm(x, mean = 2, sd = 1), ylim = c(0, 0.5),
+     type = "l")
> plot(x, dnorm(x, mean = 4, sd = 2), ylim = c(0, 0.5),
+     type = "l")
> plot(x, dnorm(x, mean = 2, sd = 2), ylim = c(0, 0.5),
+     type = "l")
> par(opar)
```

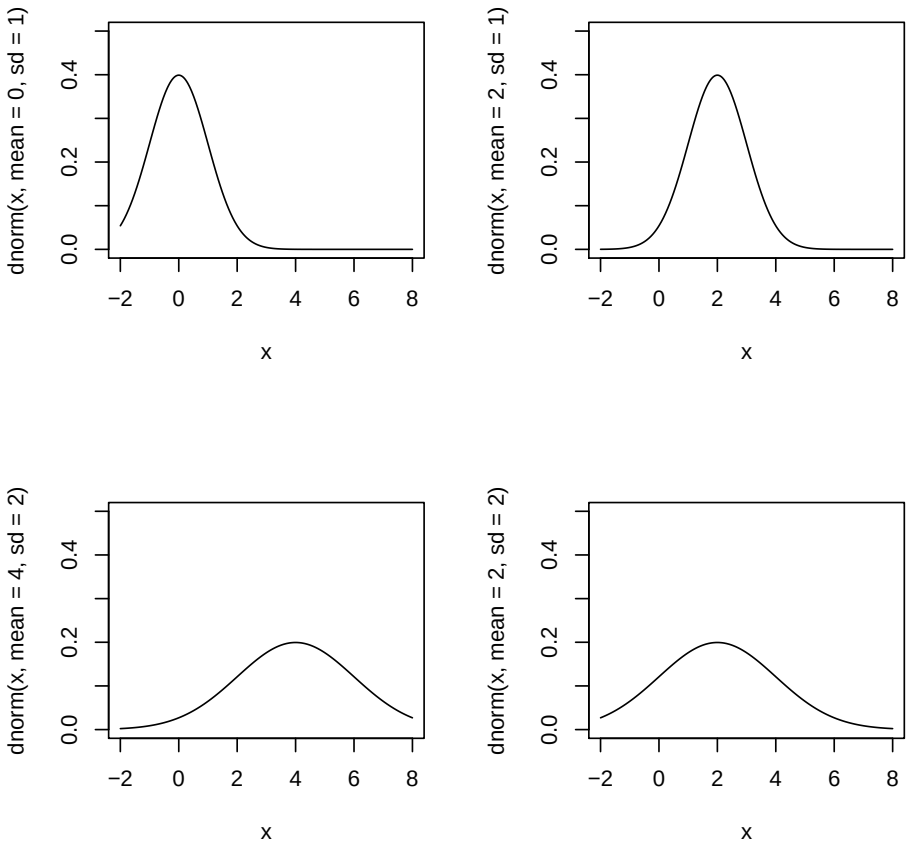


Abbildung 2.22: QQ-Plot und Lokations-Skalen-Familien
Illustration der Lokations-Skalen-Familie der Normalverteilung anhand der Dichtefunktionen zu verschiedenen Parameterkombinationen.

```
> opar <- par(mfrow = c(3, 3))
> qqnorm(rnorm(10), main = "Genaue Verteilung", sub = "a")
> abline(0, 1)
> qqnorm(rnorm(100), main = "Genaue Verteilung", sub = "b")
> abline(0, 1)
> qqnorm(rnorm(1000), main = "Genaue Verteilung", sub = "c")
> abline(0, 1)
> qqnorm(rnorm(100, 0.5, 0.5), main = "Lokationsskalenfamilie",
+     sub = "d")
> abline(0, 1)
> abline(0.5, 0.5)
> qqnorm(c(rnorm(99), 6), main = "Ausreisser", sub = "e")
> abline(0, 1)
> qqnorm(rcauchy(100), main = "Schwere Schwaenze", sub = "f")
> abline(0, 1)
> qqnorm(runif(100, -1, 1), main = "Leichte Schwaenze",
+     sub = "g")
> abline(0, 0.7)
> qqnorm(rlnorm(100, 0, 0.6), main = "Schiefe", sub = "h")
> abline(1, 1)
> qqnorm(round(rnorm(100, 4, 3)), main = "Rundung und Bindungen",
+     sub = "i")
> abline(4, 3)
> par(opar)
```

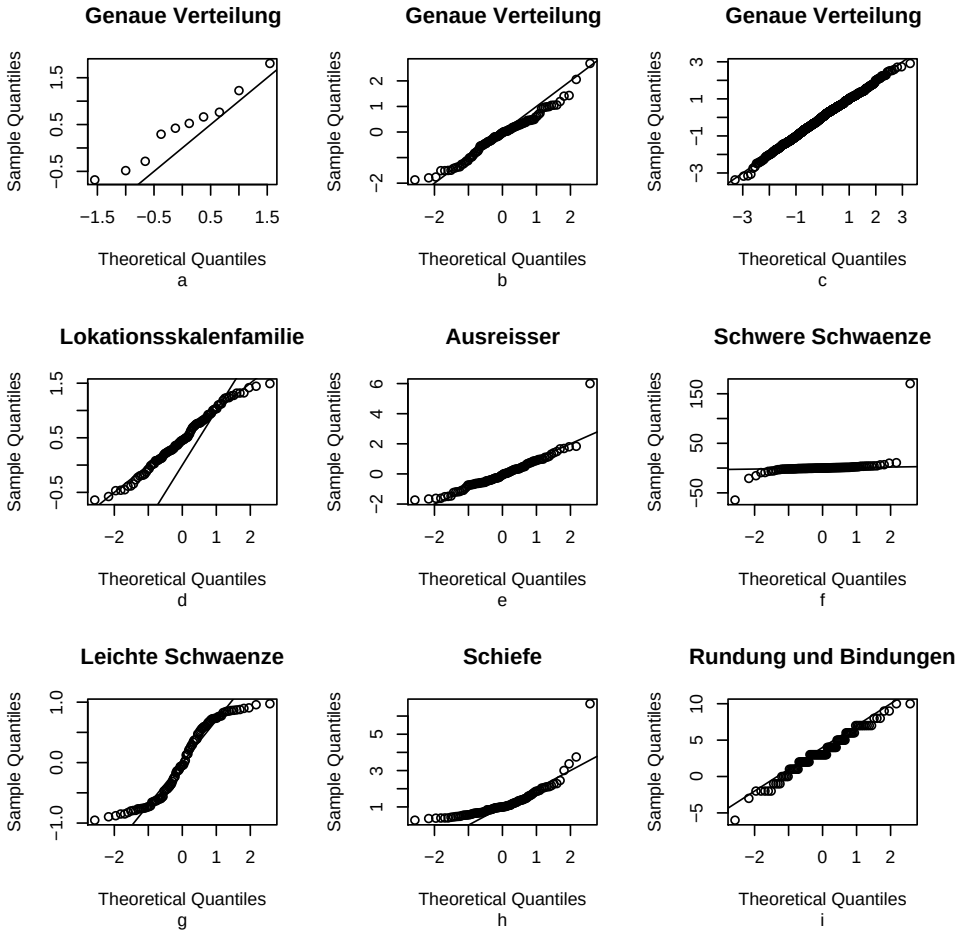


Abbildung 2.23: Interpretation von QQ-Plots

- *Winkelhalbierende*
  Liegen die Werte in etwa um die Gerade $y = x$, so ist die Verteilung der Daten in etwa mit der Vergleichsverteilung bzw. der Verteilung des Vergleichsdatensatzes kompatibel. Andernfalls nicht. Da das Auge den Einfluss des Stichprobenzufalls nicht quantifizieren kann, ist diese Interpretation jedoch nicht definitiv. Siehe Abbildung 2.23a-c.

- *Gerade*
  Liegen die Werte um eine Gerade $y = sx + \mu$ so ist die Verteilung der Daten in etwa mit der Verteilung $P_{\mu,s}$ aus der von der Vergleichsverteilung erzeugten Lokations-Skalen-Familie kompatibel. Anderfalls ist sie nicht mit dieser Lokations-Skalen-Familie kompatibel. Siehe Abbildung 2.23d.

- *Übersteilungen an den Enden*
  Ist der Anstieg der Kurve im QQ-Plot an einem oder beiden Enden im Vergleich zum Anstieg im Mittelbereich wesentlich steiler, so sind die extremen Werte extremer als nach dem Verteilungsmodell erwartet. Diese Beobachtung könnte auf Ausreißer oder eine höhere Wahrscheinlichkeit für extreme Werte, also auf sogenannte "**schwere Schwänze**", hindeuten. Siehe Abbildung 2.23e und f.

- *Verflachung an den Enden*
  Ist der Anstieg der Kurve im QQ-Plot an einem oder beiden Enden im Vergleich zum Anstieg im Mittelbereich wesentlich flacher, so sind die extremen Werte weniger extrem als nach dem Verteilungsmodell erwartet. Diese Beobachtung deutet oft auf begrenzte Wertebereiche oder Verteilungen mit besonders wenig Wahrscheinlichkeit für extreme Wert hin. Siehe Abbildung 2.23g.

- *Einfach Bogenform*
  Klare Bogenformen weisen auf schiefe Verteilungen hin. Typischerweise treten diese Bogenformen in Verbindung mit einem schweren Verteilungschwanz auf einer Seite auf. Siehe Abbildung 2.23h. Die gezeigte Durchbiegung deutet auf eine rechtsschiefe Verteilung hin, da der steile Teil rechts ist.

- *Plateaus*
  Liegen mehrer aufeinanderfolgende Punkte auf einer waagrechten Linie zusammen, so haben sie den gleichen Beobachtungswert. Das Mehrfachauftreten gleicher Messwerte wird auch als **Bindungen** bezeichnet. Bindungen können hindeuten auf:

  - gerundete Beobachtungswerte
  - Teilgruppen mit Spezialeigenschaften: z.B. Einkommen genau HARTZ IV.
  - Ungenau Angaben, z.B. wenn viele Befragte nur ungefähre Angaben machen (man spricht dann auch von **Jubiläumsstatistiken**, wenn das Alter des Betriebs nur ungefähr nach dem nächsten Jubiläum angegeben wird). Dieses Phänomen tritt auch bei Ausgaben von automatisierten Messapperaturen häufig auf.
  - geschönte Werte (wenn z.B. irgendwelche Grenzen eingehalten werden sollen, tendieren manche Leute, die knapp jenseits der Grenze liegen einen Wert knapp diesseits zu behaupten)
  - fehlende Angaben, sogenannte **missings**, die durch einen Standardwert dargestellt werden (z.B. durch 0 oder 9999)

Für parametrische statistischen Verfahren sind Bindungen, die durch Rundung zustande kommen, unproblematisch. *Für nichtparametrische Verfahren, wie Anpassungstests und rangbasierte Verfahren stellt diese Abweichung vom Modell, ein teils erhebliches Problem dar, für das nicht immer geeignete Korrekturmethoden bereitstehen.* Siehe Abbildung 2.23i.

2.2.9 Konzept für relative Skala: Darstellung auf einer Log-Skala

Für die Darstellung von Daten, die auf einer relativen Skala liegen und ein große relative Streuung (z.B. Einen Variationskoeffizienten über 0.5) haben, eignen sich die bisher vorgestellten Methoden oft nur bedingt. In diesem Fall lohnt es sich oft anstelle der eigentlichen Daten ihren Logarithmus auszuwerten (Abb. 2.24).
Unser Beispiel ist ein Datensatz, der Körpergewicht [kg] und Gehirngewicht [g] einer Auswahl von Säugetierarten angibt.

```
> data(mammals)
> mammals
```

	body	brain
Artic fox	3.385	44.50
Owl monkey	0.480	15.50
Mountian beaver	1.350	8.10
Cow	465.000	423.00
Grey wolf	36.330	119.50
Goat	27.660	115.00
Roe deer	14.830	98.20
Guinea pig	1.040	5.50
Verbet	4.190	58.00
Chinchilla	0.425	6.40
Ground squirrel	0.101	4.00
Artic ground squirrel	0.920	5.70
African giant pouched rat	1.000	6.60
Lesser short-tailed shrew	0.005	0.14
Star-nosed mole	0.060	1.00
Nine-banded armadillo	3.500	10.80
Tree hyrax	2.000	12.30
N.A. opossum	1.700	6.30
Asian elephant	2547.000	4603.00
Big brown bat	0.023	0.30
Donkey	187.100	419.00
Horse	521.000	655.00
European hedgehog	0.785	3.50
Patas monkey	10.000	115.00
Cat	3.300	25.60
Galago	0.200	5.00
Genet	1.410	17.50
Giraffe	529.000	680.00
Gorilla	207.000	406.00
Grey seal	85.000	325.00
Rock hyrax-a	0.750	12.30
Human	62.000	1320.00
African elephant	6654.000	5712.00
Water opossum	3.500	3.90

```
Rhesus monkey               6.800  179.00
Kangaroo                   35.000   56.00
Yellow-bellied marmot       4.050   17.00
Golden hamster              0.120    1.00
Mouse                       0.023    0.40
Little brown bat            0.010    0.25
Slow loris                  1.400   12.50
Okapi                     250.000  490.00
Rabbit                      2.500   12.10
Sheep                      55.500  175.00
Jaguar                    100.000  157.00
Chimpanzee                 52.160  440.00
Baboon                     10.550  179.50
Desert hedgehog             0.550    2.40
Giant armadillo            60.000   81.00
Rock hyrax-b                3.600   21.00
Raccoon                     4.288   39.20
Rat                         0.280    1.90
E. American mole            0.075    1.20
Mole rat                    0.122    3.00
Musk shrew                  0.048    0.33
Pig                       192.000  180.00
Echidna                     3.000   25.00
Brazilian tapir           160.000  169.00
Tenrec                      0.900    2.60
Phalanger                   1.620   11.40
Tree shrew                  0.104    2.50
Red fox                     4.235   50.40
```

```
> opar <- par(mfrow = c(3, 3), pch = 20)
> sapply(mammals, function(x) sd(x)/mean(x))

    body    brain
4.523156 3.285647

> sapply(mammals, mean)

    body    brain
198.7900 283.1342

> sapply(mammals, function(x) exp(mean(log(x))))

     body     brain
 3.809656 23.108440

> sapply(mammals, function(x) mean(log(x)))

    body    brain
1.337539 3.140198

> plot(mammals, main = "Plot on real scale", sub = "Alles in einer Ecke")
> plot(mammals, log = "xy", main = "Plot on log scale")
> plot(sapply(mammals, log), main = "Plot of logs", sub = "Unverstaendliche Skala")
> boxplot(mammals, main = "Boxplot on real scale", sub = "Extreme Ausreisser")
> boxplot(log(mammals), main = "Boxplot of Logs")
```

```
> opar <- par(mfrow = c(3, 3), pch = 20)
> sapply(mammals, function(x) sd(x)/mean(x))
> sapply(mammals, mean)
> sapply(mammals, function(x) exp(mean(log(x))))
> sapply(mammals, function(x) mean(log(x)))
> plot(mammals, main = "Plot on real scale", sub = "Alles in einer Ecke")
> plot(mammals, log = "xy", main = "Plot on log scale")
> plot(sapply(mammals, log), main = "Plot of logs", sub = "Unverstaendliche Skala")
> boxplot(mammals, main = "Boxplot on real scale", sub = "Extreme Ausreisser")
> boxplot(log(mammals), main = "Boxplot of Logs")
> boxplot(mammals, log = "y", main = "Boxplot on log scale",
+     sub = "Unsinnige Ausreisser")
> qqnorm(mammals$body, main = "Missfitting: qqnorm of body",
+     sub = "Falsche Verteilung")
> qqnorm(log(mammals$body), main = "qqnorm of log(body)")
> QQplot(mammals$body, qlnorm, main = "QQplot with lognormal",
+     sub = "Keine Lokationsskalenfamilie")
> par(opar)
```

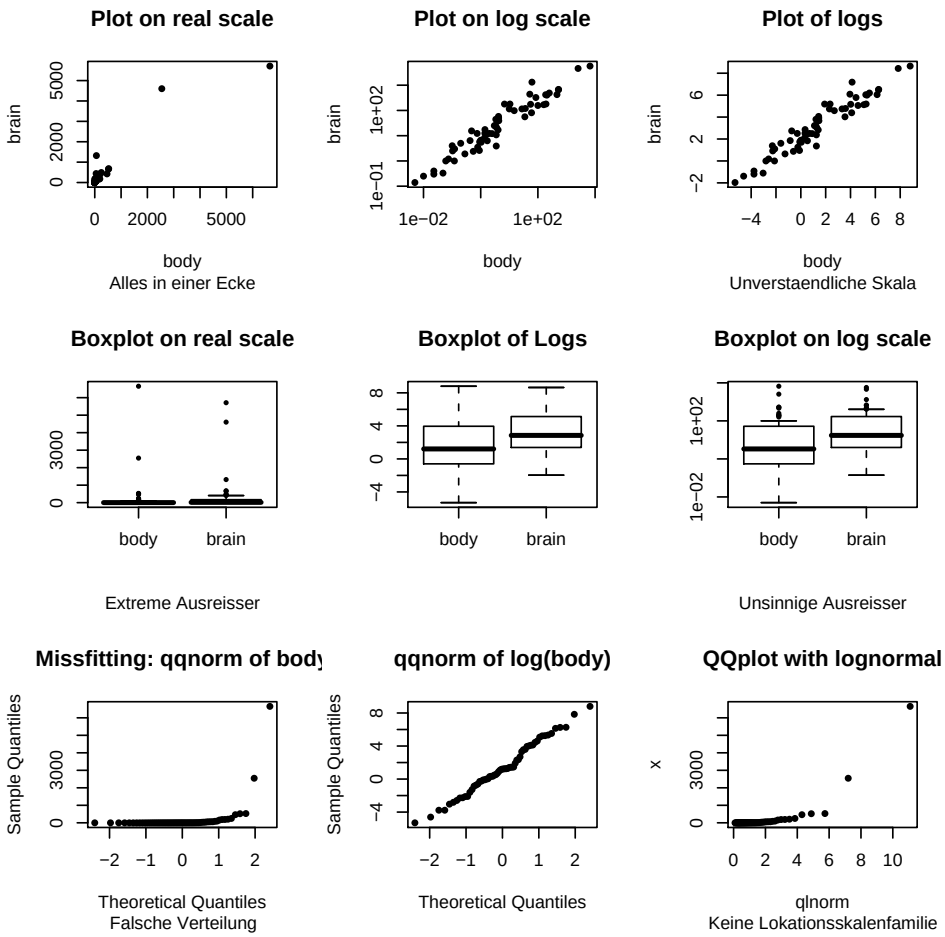


Abbildung 2.24: Effekte und Arten der Logarithmustransformation
Darstellung der Auswertung in Logarithmen. Die Linke Spalte zeigt die klassische Graphik und die dabei entstehenden Artefakte. Die Mittlere Spalte zeigt die richtig Graphik zur Auswertung in log-Skala. Die Rechte Spalte zeigt eine ungeschickte Graphik in Log-Skala mit den entstehenden Artefakten.

```
> boxplot(mammals, log = "y", main = "Boxplot on log scale",
+     sub = "Unsinnige Ausreisser")
> qqnorm(mammals$body, main = "Missfitting: qqnorm of body",
+     sub = "Falsche Verteilung")
> qqnorm(log(mammals$body), main = "qqnorm of log(body)")
> QQplot(mammals$body, qlnorm, main = "QQplot with lognormal",
+     sub = "Keine Lokationsskalenfamilie")
> par(opar)
```

Interessant ist dabei z.B. die Angabe des Durchschnittsgewichts, dass beim gewöhnlichen Mittelwert auf knapp 200kg angesetzt wird, wobei der geometrische Mittelwert auf einer relativen Skala knapp 4 kg ansetzt und damit dem Gorilla sein Recht als großes Säugetier lässt.

Bedenken Sie aber auch, dass dieser Datensatz aller Wahrscheinlichkeit nach keine repräsentative Stichprobe aller lebenden Säugerarten darstellt. Wir dürfen also die hier gefundenen Ergebnisse keinesfalls auf alle Säugerarten verallgemeinern.

2.3 Univariate Daten

Bei kategoriellen Daten gibt es nur wenige Wert, die dann oft auftreten. Es steht also nicht die Darstellung des Wertes selbst, sondern die Darstellung des Anteils, wie oft jeder Wert auftritt.

2.3.0.0.1 R: Erzeugung einer Datentafel Bei kategoriellen Daten können die Informationen als Datenmatrix oder als Datentafel vorliegen. Bei rein kategoriellen Fragestellungen bevorzugt man oft die Darstellung als Datentafel. Eine Datentafel kann man aus einer Datenmatrix mittel des `table`-Befehls erzeugen:

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
```

Manchmal liegen die Daten aber auch bereits als (unter Umständen hochdimensionale) Datentafel vor. So einen Datentafel kann mit dem Befehl `ftable` (**f**lat **table**) übersichtlich angezeigt werden:

```
> help(Titanic)

Titanic               package:datasets               R
Documentation

Survival of passengers
on the Titanic

Description:

     This data set provides information on the fate of
     passengers on the fatal maiden voyage of the ocean liner
     'Titanic', summarized according to economic status (class),
     sex, age and survival.

Usage:

     Titanic

Format:

     A 4-dimensional array resulting from cross-tabulating
```

```
     2201 observations on 4 variables.  The variables and
     their levels are as follows:

       No  Name      Levels
        1  Class     1st, 2nd, 3rd, Crew 2  Sex       Male,
        Female 3  Age       Child, Adult 4  Survived  No, Yes

Details:

     The sinking of the Titanic is a famous event, and new
     books are still being published about it.  Many well-known
     facts-from the proportions of first-class passengers to
     the "women and children first" policy, and the fact that
     that policy was not entirely successful in saving the
     women and children in the third class-are reflected in
     the survival rates for various classes of passenger.

     These data were originally collected by the British
     Board of Trade in their investigation of the sinking.
     Note that there is not complete agreement among primary
     sources as to the exact numbers on board, rescued, or lost.

     Due in particular to the very successful film
     'Titanic', the last years saw a rise in public
     interest in the Titanic.  Very detailed data about
     the passengers is now available on the Internet,
     at sites such as _Encyclopedia Titanica_ (<URL:
     http://www.rmplc.co.uk/eduweb/sites/phind>).

Source:

     Dawson, Robert J. MacG. (1995), The 'Unusual Episode' Data
     Revisited. _Journal of Statistics Education_, *3*. <URL:
     http://www.amstat.org/publications/jse/v3n3/datasets.dawson.html>

     The source provides a data set recording class, sex,
     age, and survival status for each person on board of the
     Titanic, and is based on data originally collected by
     the British Board of Trade and reprinted in:

     British Board of Trade (1990), _Report on the Loss of the
     'Titanic' (S.S.)_. British Board of Trade Inquiry Report
     (reprint). Gloucester, UK: Allan Sutton Publishing.

Examples:

     require(graphics) mosaicplot(Titanic, main = "Survival
     on the Titanic") ## Higher survival rates in children?
     apply(Titanic, c(3, 4), sum) ## Higher survival rates
     in females?  apply(Titanic, c(2, 4), sum) ## Use loglm()
     in package 'MASS' for further analysis ...
```

```
> data(Titanic)
> ftable(Titanic)
                    Survived  No Yes
Class Sex    Age
1st   Male   Child            0   5
             Adult          118  57
      Female Child            0   1
```

```
             Adult           4 140
2nd   Male   Child           0  11
             Adult         154  14
      Female Child           0  13
             Adult          13  80
3rd   Male   Child          35  13
             Adult         387  75
      Female Child          17  14
             Adult          89  76
Crew  Male   Child           0   0
             Adult         670 192
      Female Child           0   0
             Adult           3  20
```

Um daraus eine einzelne Information herauszugreifen kann man den folgenden Befehl verwenden:

```
> margin <- function(x, ...) apply(x, pmatch(c(...), names(dimnames(x))),
+     sum)
> margin(Titanic, "Sex", "Survived")
```

```
        Survived
Sex        No Yes
  Male   1364 367
  Female  126 344
```

2.3.1 Balkendiagramm/Barchart

Häufigkeiten und Anteile können graphisch durch

- Flächen
- Längen (z.B. Höhen oder Breiten)
- Winkelanteile

dargestellt werden. Allgemein gilt das eine flächenproportionale Darstellung den richtigen groben Eindruck vermittelt, während sich nur eine längenproportionale Darstellung für objektive Vergleiche eignet. Beide Ansätze kombiniert treten im **Balkendiagramm** (engl. **barchart**) auf:

```
> opar <- par(mfrow = c(1, 3))
> barplot(margin(Titanic, "Survived"), main = "Survived")
> barplot(margin(Titanic, "Sex"), main = "Geschlecht")
> barplot(margin(Titanic, "Class"), main = "Klasse")
> par(opar)
```

- Im Balkendiagramm werden Häufigkeiten flächenproportional dargestellt.
- Da alle Balken gleich breit sind, werden die Häufigkeiten auch höhenproportional dargestellt.
- In vielen "populären" Graphiken (z.B. in Zeitungen) werden die Balken fast nie werteproportional dargestellt.
- Höhen von Balken mit 3D-Effekt sind schwer einzuschätzen.
- Das Balkendiagramm wird häufig mit dem Histogramm verwechselt. Die Unterschiede sind jedoch leicht erkennbar:

```
> opar <- par(mfrow = c(1, 3))
> barplot(margin(Titanic, "Survived"), main = "Survived")
> barplot(margin(Titanic, "Sex"), main = "Geschlecht")
> barplot(margin(Titanic, "Class"), main = "Klasse")
> par(opar)
```


```
> opar <- par(mfrow = c(1, 3))
> pie(margin(Titanic, "Survived"), main = "Survived")
> pie(margin(Titanic, "Sex"), main = "Geschlecht")
> pie(margin(Titanic, "Class"), main = "Klasse")
> par(opar)
```

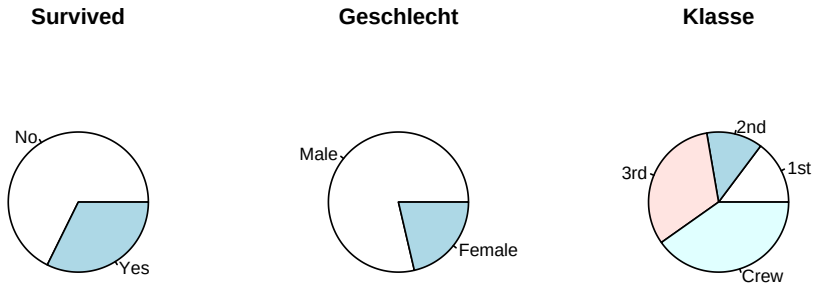


Abbildung 2.25: Darstellung univariater kategorieller Daten

  - Balkendiagramme stellen Häufigkeiten von Daten einer diskreten Skala dar. Histogramme stellen Dichten von Daten einer stetigen Skala dar.
  - Die Balken des Balkendiagramms haben einen Abstand, um den diskreten Charakter zu betonen. Die Balken des Histogramms berühren sich direkt, da sie auf derselben Intervallgrenze liegen.

- Balkendiagramm und Histogramm haben aber auch Gemeinsamkeiten:
  - Beide Graphiken stellen Anteile oder Häufigkeiten von Daten mit bestimmten Werten flächenproportional dar.

- Bei der Anzeige **ordinaler Daten** sollte die Reihenfolge der Balken der natürlichen Ordnung der Daten entsprechen. Das kann man oft nur durch Tricks, wie z.B. die Vergabe alphabetisch richtig sortierender Namen erreichen.

2.3.2 Kenngrößen für die diskrete Daten

Die Wichtigste Kenngröße für diskrete Daten ist die durch den Anteil $\hat{p}_k$ geschätzte Wahrscheinlichkeit p_k, dass Kategorie k auftritt:

```
> margin(Titanic, "Survived")/sum(Titanic)

      No      Yes
0.676965 0.323035

> margin(Titanic, "Sex")/sum(Titanic)

     Male    Female
0.7864607 0.2135393

> margin(Titanic, "Class")/sum(Titanic)

      1st       2nd       3rd      Crew
0.1476602 0.1294866 0.3207633 0.4020900
```

2.3.3 Kuchendiagramm/Piechart

Das **Kuchendiagramm** oder **Kreisdiagramm** (engl. **barchart**)erfreut sich, insbesondere im Medienbereich großer Beliebtheit.

```
> opar <- par(mfrow = c(1, 3))
> pie(margin(Titanic, "Survived"), main = "Survived")
> pie(margin(Titanic, "Sex"), main = "Geschlecht")
> pie(margin(Titanic, "Class"), main = "Klasse")
> par(opar)
```

2.4 Multivariate Graphik für stetig Daten

Für die Darstellung der Abhängigkeiten zwischen Merkmalen benötigt man multivariate Graphiken, in denen die Informationen mehrerer Variablen kombiniert dargestellt werden. Die multivariaten Graphiken verwenden dieselben Darstellungsmöglichkeiten, wie die univariaten Graphiken, kombinieren diese allerdings zu einem größeren Ganzen.

Die Graphiken Einteilung der multivariaten Graphiken erfolgt wieder nach den Skalen. Allerdings spielen dabei mehrere Skalen eine Rolle, so dass immer darauf geachtet werden muss, welche Skalen miteinander kombiniert werden.

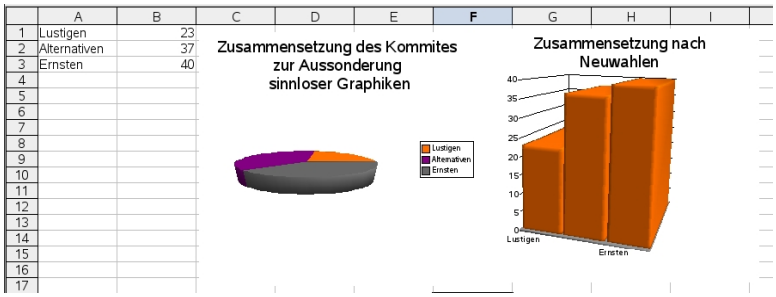


Abbildung 2.26: Der Unsinn der perspektivischer "Profigraphik"
3D Graphiken kann man leicht mit jeder Tabellenkalkulation herstellen und sie sehen auch beeindruckend aus. Eine tatsächliche Information kann das Auge der Graphik jedoch kaum entnehmen.

2.4.1 Streudiagramm

Das **Streudiagramm** (engl. **Scatterplot**) ist der Klassiker unter den statistischen Graphiken.

```
> attach(iris)
> plot(Sepal.Length, Sepal.Width, col = c("red", "green",
+      "blue")[Species], pch = 20, main = "Kelchblatt")
> detach(iris)
```

Es eignet sich zur simultanen Darstellung zweier stetiger Merkmale.

- Wie das Punktdiagramm ist auch das Streudiagramm anfällig gegenüber Bindungen, die durch Überlagerung unsichtbar werden. Dadurch können sogar typische Datenkombinationen unwichtig erscheinen.

  Wie beim Punktdiagramm können die Punkte verzittert werden. Da allerdings keine kanonische Verzitterungsrichtung mehr zur Verfügung steht, muss die Verzitterung per Hand dosiert werden.

- Fehlende Werte werden in Streudiagrammen typischerweise nicht eingezeichnet.

- Kenngrößen kann man im Streudiagramm praktisch nicht mit dem Auge schätzen.

- Die Beurteilung der Qualität Ausreißer zu sein, fällt im Streudiagramm schwer.

- Die Lage des Nullpunktes und der Grenzen (die z.B. durch Ausreißer bestimmt werden) können den Eindruck, den ein Streudiagramm macht, maßgeblich beeinflussen.

2.4.2 Kenngrößen für die stetige Abhängigkeit

Zum Beschreiben eines Zusammenhangs zweier reeller Merkmale gibt es eine Reihe von Kenngrößen:

- *Die empirische Kovarianz*
  Die **empirische Kovarianz** ist die durch

$$\hat{\mathrm{cov}}(X,Y) := \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

```
> attach(iris)
> plot(Sepal.Length, Sepal.Width, col = c("red", "green",
+     "blue")[Species], pch = 20, main = "Kelchblatt")
> detach(iris)
```

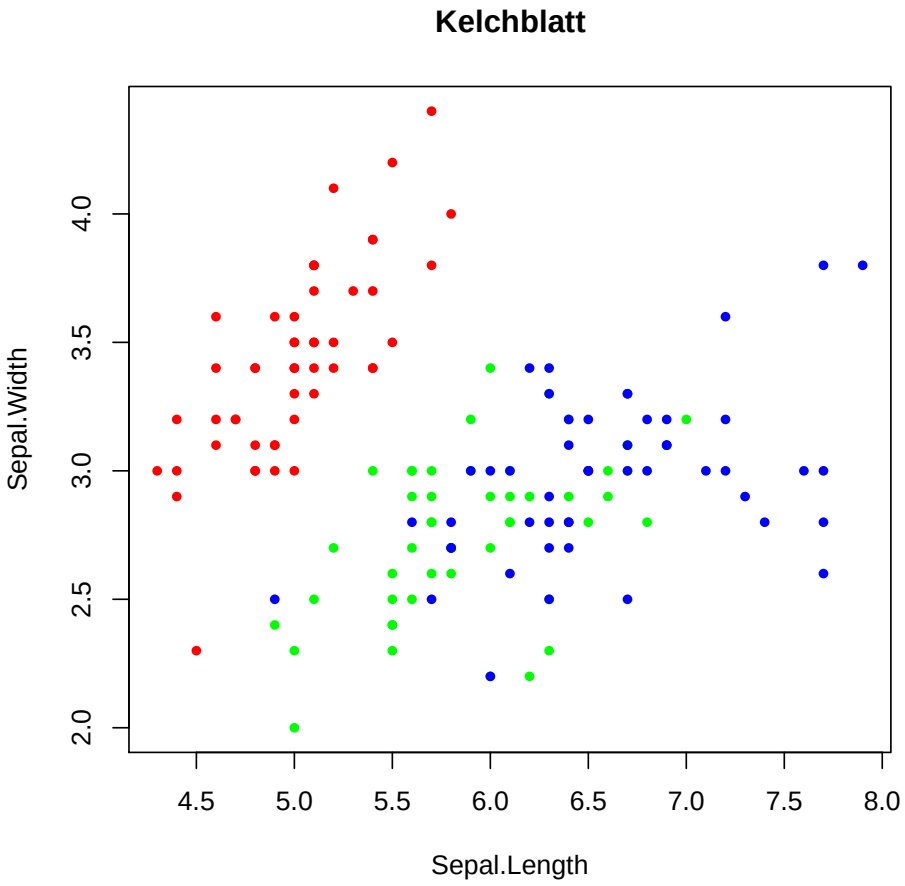


Abbildung 2.27: Das Streudiagramm

```
> attach(iris)
> plot(Sepal.Length, Sepal.Width, col = c("red", "green",
+     "blue")[Species], pch = 20, main = "Kelchblatt",
+     xlim = c(0, max(Sepal.Length)), ylim = c(0, max(Sepal.Width)))
> detach(iris)
```

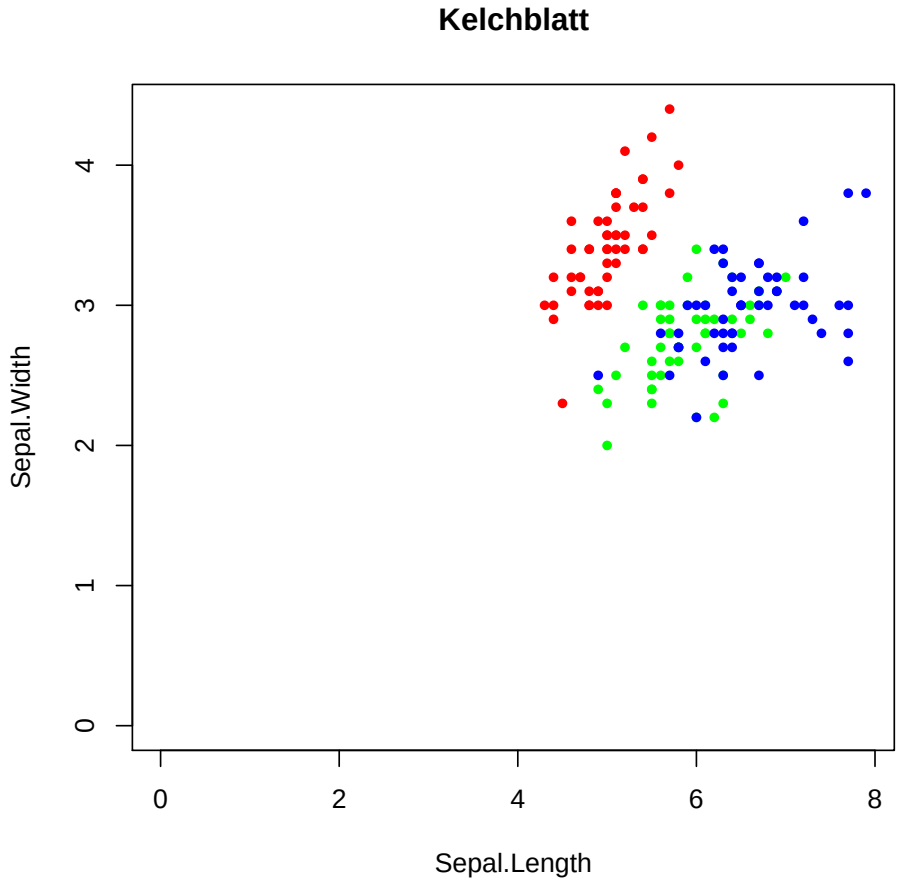


Abbildung 2.28: Optischer Effekt des Nullpunkts

```
> attach(iris)
> plot(Sepal.Length + rnorm(150, sd = 0.04), Sepal.Width +
+     rnorm(150, sd = 0.01), col = c("red", "green", "blue")[Species],
+     pch = 1, main = "Kelchblatt")
> detach(iris)
```

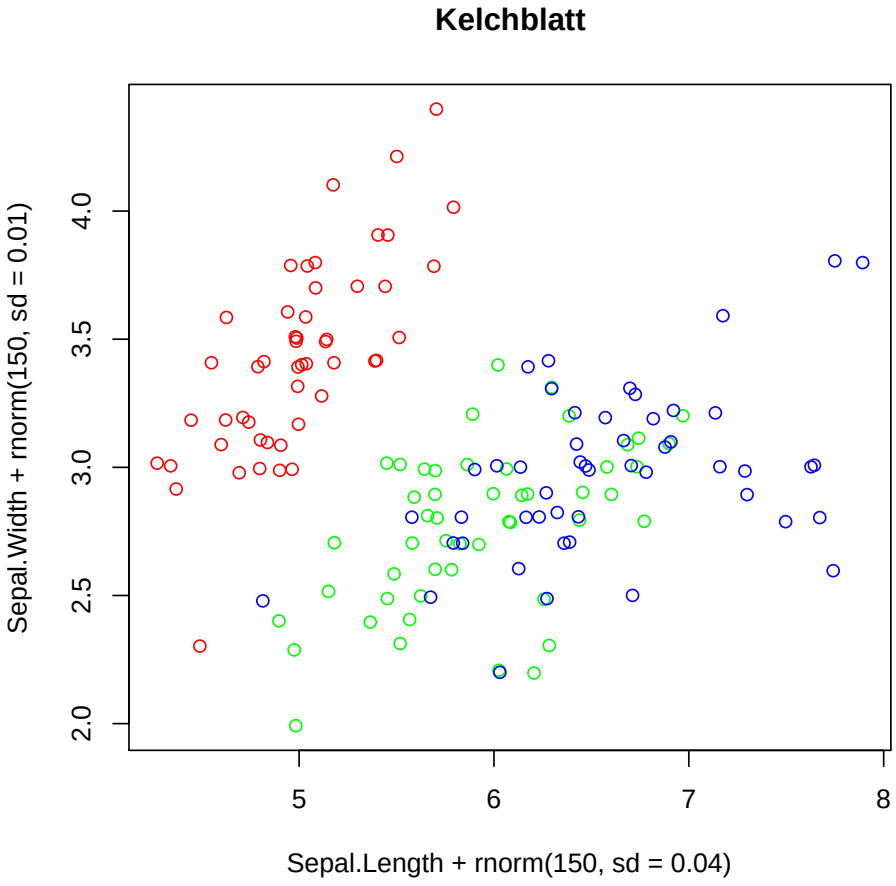


Abbildung 2.29: Auflösen von Überlagerung durch Verzittern

gegeben und stellt eine absolute Kenngröße des Zusammenhangs dar. Für normalverteilte Daten kann sie über die einfache Interpretation, dass es unabhängige Größen Z_x, Z_y und Z_{xy} gibt, verstanden werden, so dass:

$X = Z_x + Z_{xy}$, $Y = Z_y + Z_{xy}$, $\mathrm{var}(Z_x y) = \mathrm{cov}(X, Y)$, $\mathrm{var}(X) = \mathrm{var}(Z_x) + \mathrm{var}(Z_{xy})$, $\mathrm{var}(Y) = \mathrm{var}(Z_y) + \mathrm{var}(Z_{xy})$

Die Kovarianz entspricht also "der Varianz, die beide gemeinsam" haben.

Der Nachteil der Kovarianz ist, dass sie nur als Anteil an den Einzelvarianzen wirklich interpretierbar ist.

- *(Pearsons) Korrelation*
  Die **empirische Korrelation** ist durch
  $$\hat{\mathrm{cor}}(X, Y) := \frac{\hat{\mathrm{cov}}(X, Y)}{\sqrt{\hat{\mathrm{var}}(X)\hat{\mathrm{var}}(Y)}}$$
  gegeben und stellt eine relative Kenngröße des Zusammenhangs dar. Der Wert der Korrelation liegt im Intervall $[-1, 1]$ und misst den linearen Zusammenhang zwischen den Merkmalen. Je größer der Betrag ist, desto mehr nähert sich die Punktwolke einer Geraden an. Ist der Betrag positiv, so deutet das auf einen direkt proportionalen Zusammenhang hin. Ist der Betrag negativ, so auf einen antiproportionalen.

- *Spearmans Rangkorrelation*
  Spearmans Rangkorrelation ersetzt die Daten durch Ihre Rangziffern und berechnet dann die gewöhnliche Pearsonsche Korrelation. Auf diese Weise wird die Berechnung gegen Ausreißer stabil und es werden auch nichtlineare aber steigende Zusammenhänge mit großen Korrelationen wiedergegeben. Übersteigt also die Spearman Korrelation die Pearson Korrelation deutlich, so liegt vermutlich ein nichtlinearer Zusammenhang oder ein Ausreißer vor.

  Übersteigt hingegen die Pearson Korrelation die Spearman Korrellation Betragsmäßig deutlich, so liegen vermutlich Ausreißer vor. Eine Aufklärung kann nur ein Streudiagramm bringen.

2.4.3 log-Skala

Bei Werten in relativer Skala kann wieder ein logarithmische Skala angebracht sein.

2.4.4 Streudiagrammmatrix

2.4.5 Parallele Plots mit gleichem Koordinatensystem

Ein einfaches Prinzip um mehrere Informationen gleichzeitig zu zeigen, ist sie einfach nebeneinader zu malen. Dabei kommt als Informationsgewinn allerdings bei einer Abstimmung der Koordinatensystem eine wertemäßige Vergleichsmöglichkeit hinzu.

2.5 Parallele Koordinaten

Ganz anders kann man dieselbe Idee des Nebeneinandersetzen ausnutzen, indem man anstelle Verbindung über die gleichen Koordinaten, die Verbindung über zusammengehörige Individuen herstellt. Die einzelnen Werte werden nun auf unverbundenen parallelen Koordinatenachsen aufgetragen. Die Verbindung zwischen den

```
> opar <- par(mfrow = c(3, 3))
> require(MASS)

[1] TRUE

> corPlot <- function(rho) {
+     plot(mvrnorm(100, c(0, 0), cbind(c(1, rho), c(rho,
+         1))), main = paste("cor(X,Y)=", rho), xlab = "X",
+         ylab = "y")
+ }
> for (rho in seq(-1, 1, len = 9)) corPlot(rho)
> par(opar)
```

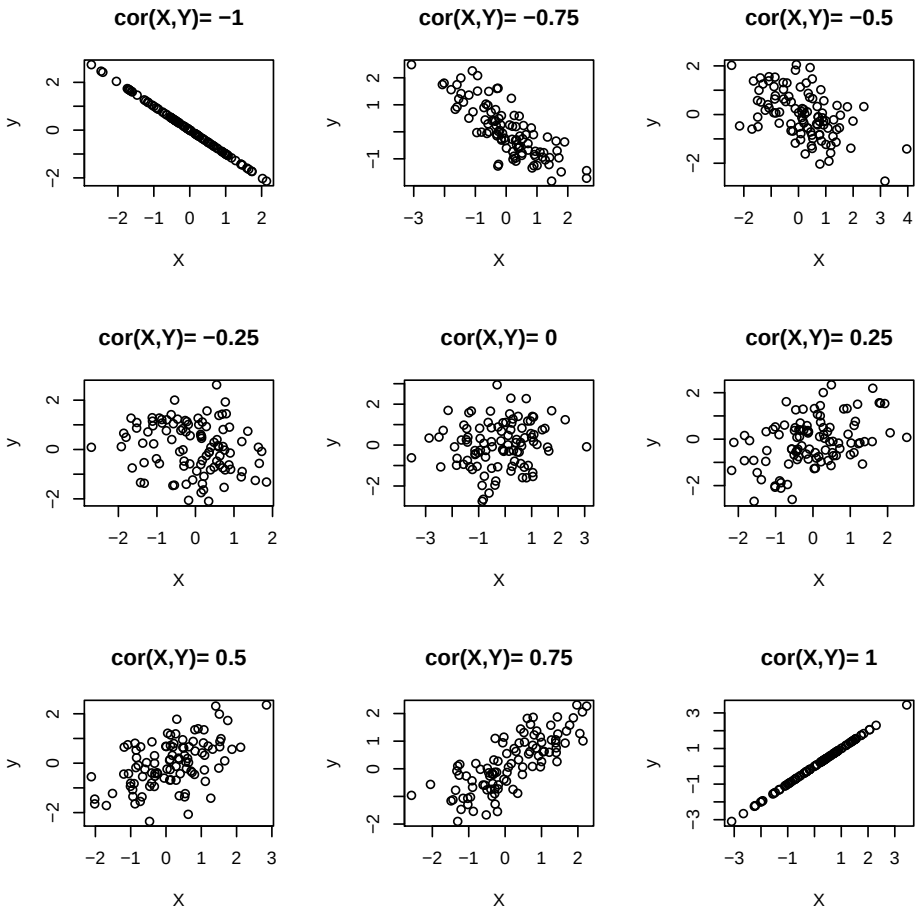


Abbildung 2.30: Pearson Korrelation

```
> attach(iris)
> opar <- par(mfrow = c(1, 2))
> plot(Petal.Length, Petal.Width, col = c("red", "green",
+     "blue")[Species], pch = 20, main = "Bluetenblatt",
+     log = "")
> plot(Petal.Length, Petal.Width, col = c("red", "green",
+     "blue")[Species], pch = 20, main = "Bluetenblatt",
+     log = "xy")
> par(opar)
> detach(iris)
```

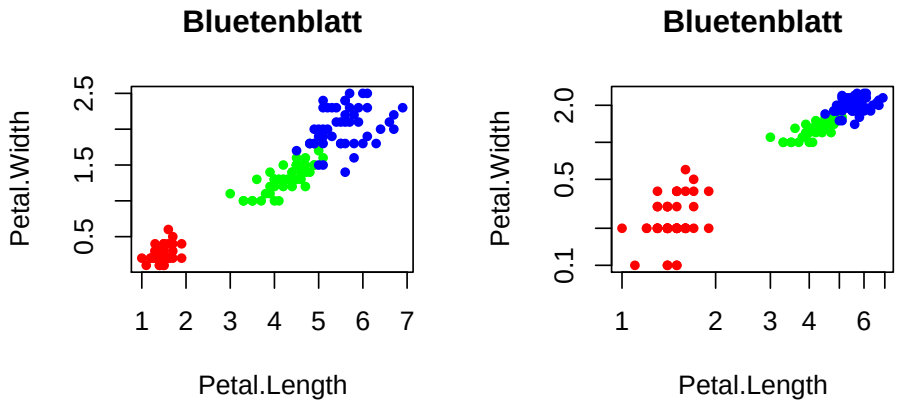


Abbildung 2.31: Streudiagramm in logarithmischer Skala
In der linken Unlogarithmierten Graphik erhält man den Eindruck, dass die Blütenblätter von Iris versicolor und Iris virginica eine deutlich größere Streuung aufweisen, als die von Iris setosa. Die Darstellung in log-Skala auf der rechten Seite zeigt jedoch klar, dass die relative Streuung eher bei Iris setosa deutlich größer ist.

```
> pairs(iris)
```

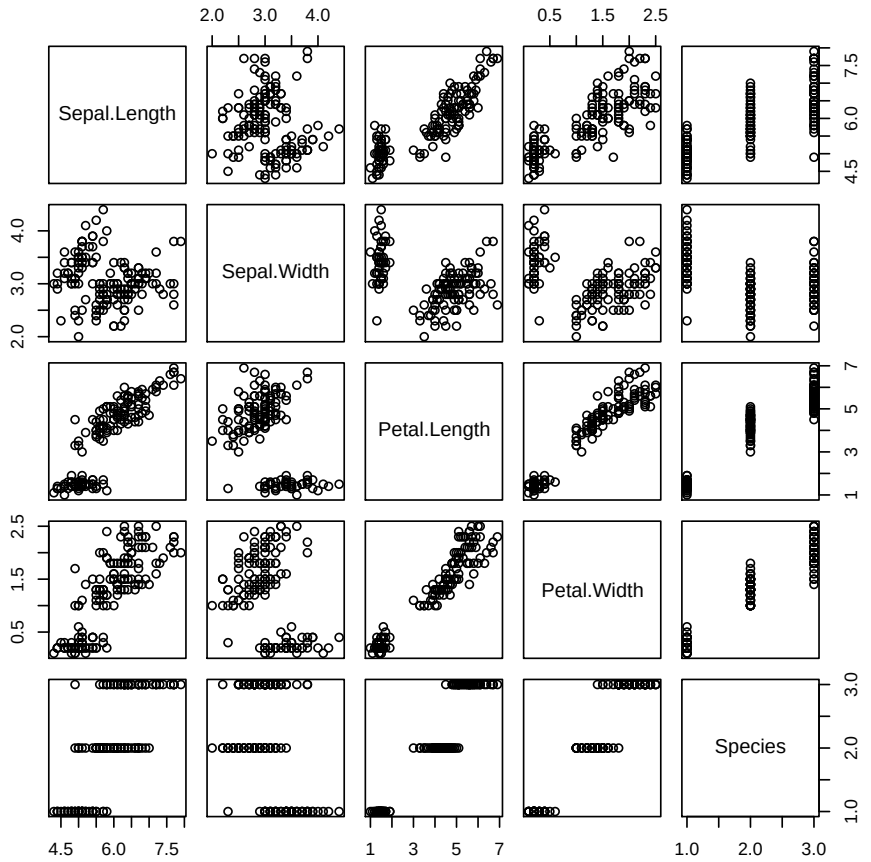


Abbildung 2.32: Streudiagrammmatrix

```
> stripchart(iris, method = "jitter")
```

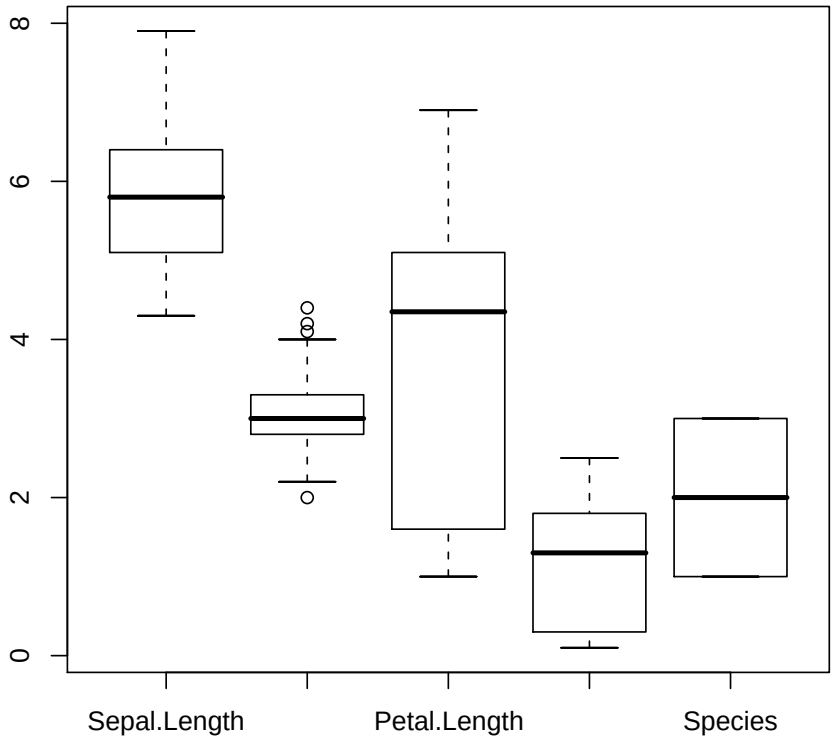


Abbildung 2.33: Parallele Punktdiagramme für mehrere Variablen

```
> boxplot(iris)
```

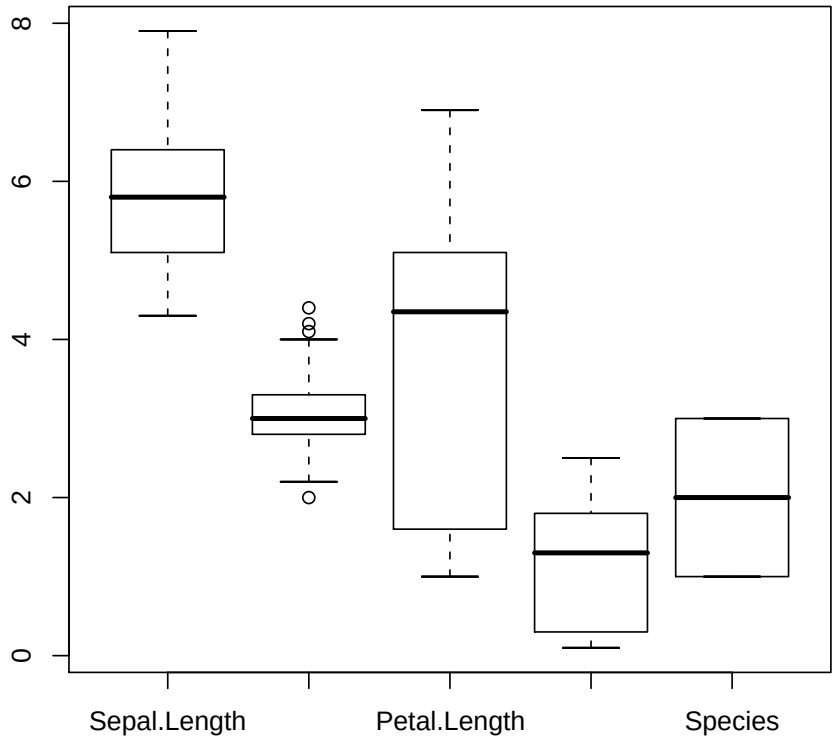


Abbildung 2.34: Parallel Boxplots für mehrere Variablen

Merkmalsausprägungen des gleichen Individuums kann dann über eine Verbindungslinie hergestellt werden.

Punkte im Merkmalsraum werden in parallelen Koordinatenplots also als Linien dargestellt. Auf der anderen Seite deuten gemeinsame Schnittpunkte der Linien zwischen zwei Koordinatenachsen auf eine gemeinsame Gerade hin, auf der die Punkte im Merkmalsraum liegen.

Der große Vorteil dieser Graphik ist, dass man auf diese Weise praktisch unbegrenzt viele stetige Informationen in einer Graphik darstellen kann. Allerdings werden gerade die ferneren Abhängigkeiten schnell unübersichtlich.

Da immer nur zwei Koordinatenachsen unmittelbar verbunden sind, spielt die Reihenfolge der Achsen für den Gesamteindruck eine wichtige Rolle.

```
> require(MASS)

[1] TRUE

> parcoord(iris[1:4], col = c("red", "green", "blue")[iris$Species])
```

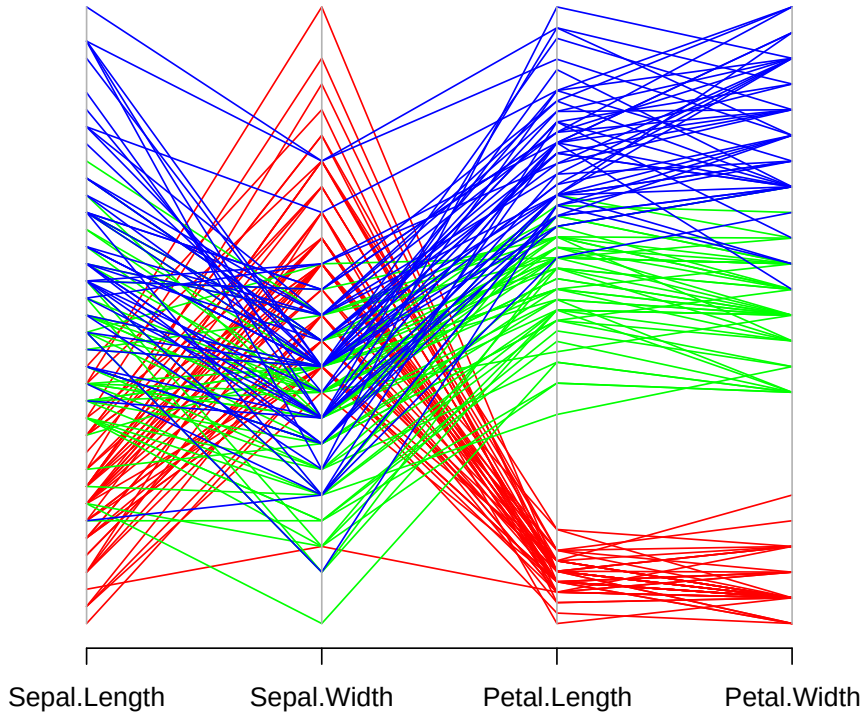


Abbildung 2.35: Parallele Koordinaten

2.6 Graphik für gemischte Daten

Das einfachste Grundprinzip zur Darstellung von Abhängigkeiten zwischen diskreten und stetigen Größen ist, die stetige Graphik einfach aufgegliedert nach den Stufen des stetigen Faktors mehrfach zu widerholen. Dieses Grundprinzip der Aufgespalteten Darstellung eines Datensatzes nach einem oder mehreren Merkmalen bezeichnet man manchmal auch als **trellis** (engl. für Gitter).

Das Prinzip der parallelen Darstellung, welches einen wertemäßigen direkten Vergleich erlaubt, wird hier wiederverwendet. Nur wird nun die Zugehörigkeit zu einer Teilgesamtheit, anstelle der Zugehörigkeit zu einem Merkmal als Aufteilungskriterium verwendet.

2.6.1 Gesplittete Punktdiagramme

```
> stripchart(split(iris$Sepal.Length, iris$Species), method = "jitter",
+     main = "Sepal.Length")
```

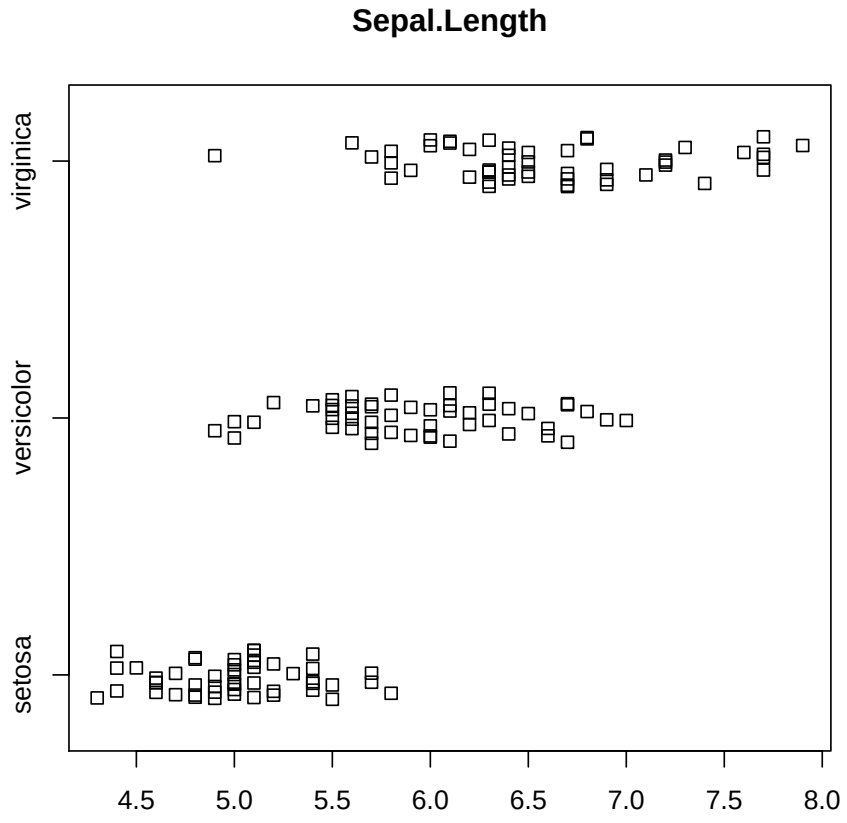


Abbildung 2.36: Gesplittete Punktdiagramme

2.6.2 Gesplittete Boxplots

```
> boxplot(split(iris$Petal.Width, iris$Species), main = "Petal.Width")
```

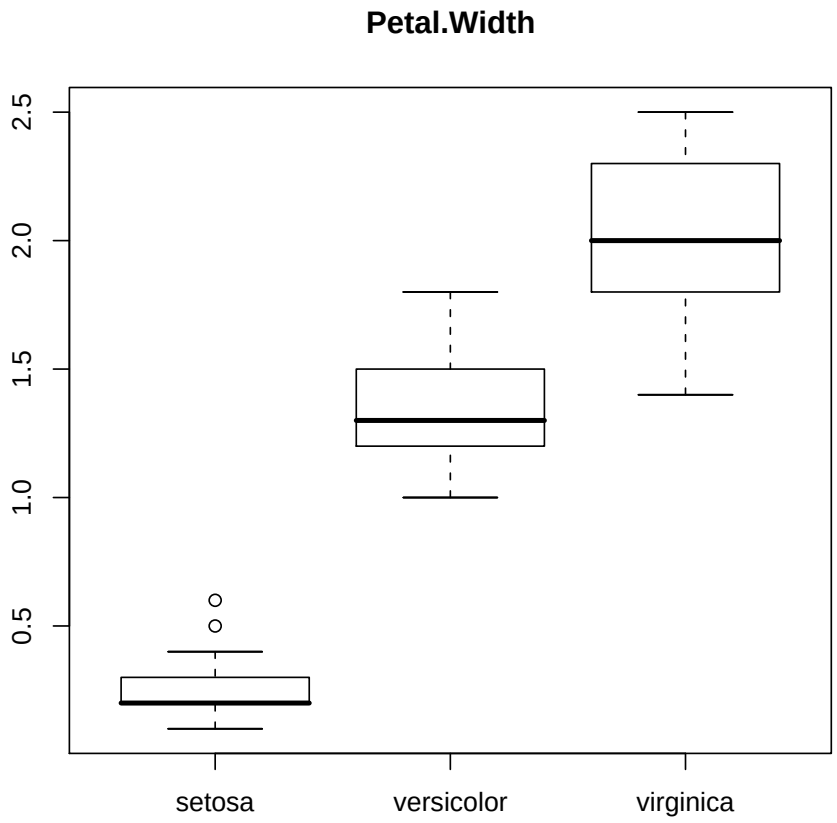


Abbildung 2.37: Gesplittete Boxplots

2.6.3 Gekerbte Boxplots

Um schnell graphisch zu überprüfen, ob sich Gruppen wirklich unterscheiden, oder ob die Abweichungen innerhalb der Zufallsschwankungen durch die Stichprobenwahl liegt, eignen sich **gekerbte Boxplots**. Die Kerben basieren auf einer Normalverteilungsannahme und erlauben, wenn diese in etwa gegeben ist die folgende Interpretation:
Die Kerben zweier Boxen zu Grundgesamtheiten mit gleichem Median überschneiden sich mit einer Wahrscheinlichkeit von 95%. Überlappen sich die Kerben zweier Boxen nicht, so kann man also davon ausgehen, dass die Mediane der Grundgesamtheiten vermutlich unterschiedlich sind. Ein objektiverer Vergleich wird durch statische Tests ermöglicht.

```
> boxplot(split(iris$Sepal.Width, iris$Species), main = "Sepal.Width",
+     notch = TRUE)
```

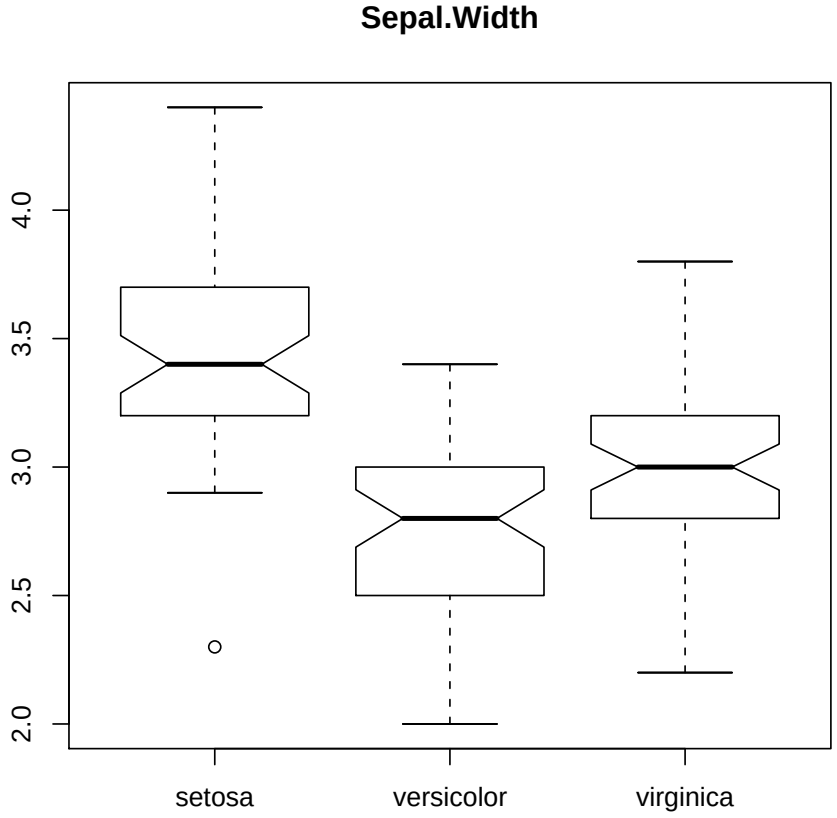


Abbildung 2.38: Gesplittete und gekerbte Boxplots

2.6.4 Labels und Farben

Farben, Symbole und Beschriftungen eignen sich, um eine oder zwei kategorielle Informationen zusätzlich jede stetige Graphik hinzuführen, welche die Fälle durch separate Punkte darstellt. Übereinanderplotten von Symbolen ist hier allerdings

wieder eine großes Problem, da so objektiv Informationen verloren gehen können. Im optischen Gesamteindruck tritt das Plotsymbol hinter der Farbe zurück und kann daher nicht wirklich Verteilungsinformation transportieren. Farben müssen (beispielsweise durch eine Legende) immer erst mit einer Bedeutung versehen werden, ehe sie vom Nutzer verstanden werden.

```
> pairs(iris, col = c("red", "green", "blue")[iris$Species],
+     pch = c("s", "v", "c")[iris$Species])
```

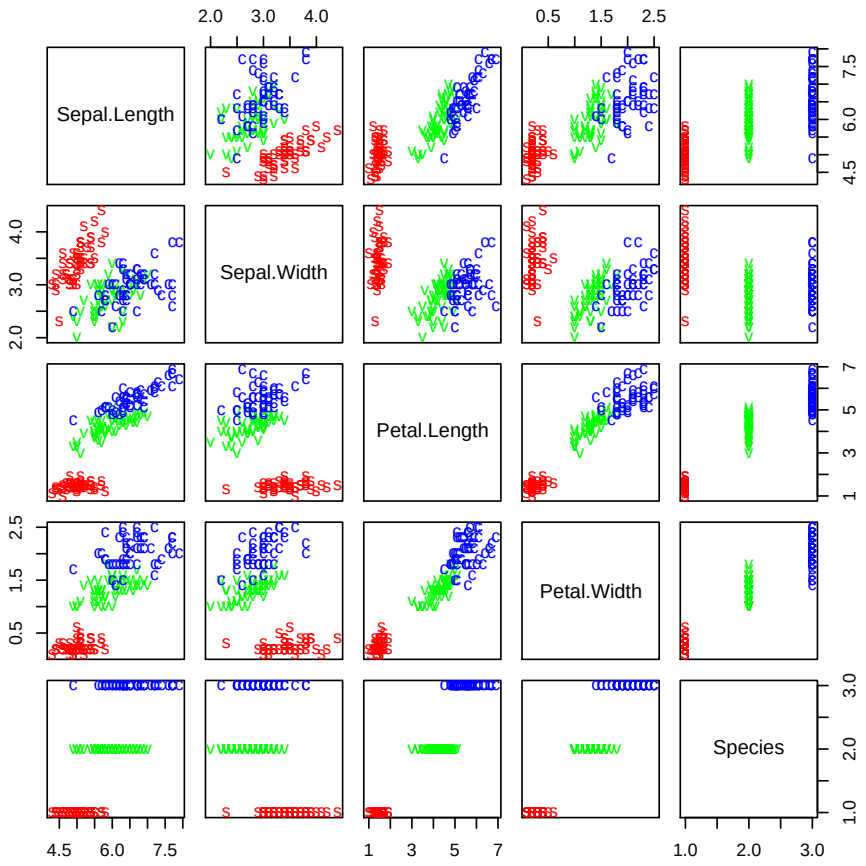


Abbildung 2.39: Gesplittete und gekerbte Boxplots

```
> try(detach(Acorn))
```

2.7 Multivariate Graphik für diskrete Daten

```
> data(Titanic)
> X <- apply(Titanic, c(2, 3), sum)
> X
```

```
        Age
Sex      Child Adult
  Male      64  1667
  Female    45   425
```

2.7.1 Gestapelte Balken

Gestapelte Balken erlauben die kombinierte Darstellung zweier kategorieller Merkmale.

- Es können sowohl absolute Häufigkeiten, als auch die bedingte Verteilung der zweiten gegeben die erste Variable wahrgenommen werden. Beide können aber vom Auge nicht direkt quantitativ verglichen werden.
- Die Reihenfolge ist daher für die Darstellung wesentlich, da ja immer nur eine bedingte Verteilung direkt erkennbar wird.
- Bei ordinalen Daten sind die Kategorien entsprechend geordnet darzustellen.

```
> barplot(X, main = "Passagiere der Titanic")
```

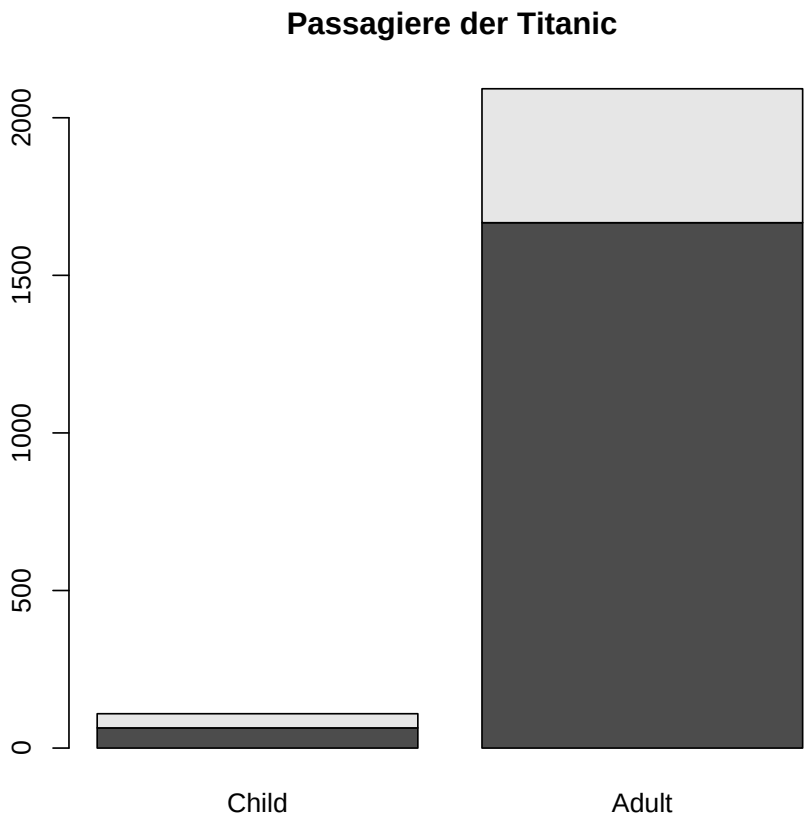


Abbildung 2.40: Gestapelte Balken (erste Möglichkeit)

2.7.2 Parallele Balken

Parallele Balkendiagramme erlauben den direkten Vergleich der absoluten Häufigkeiten in den verschiedenen Untergruppierungen und eine qualitative Beurteilung von odds in den Untergruppierungen. Die bedingten Verteilungen entziehen sich hier der Beobachtung.

```
> barplot(t(X), main = "Passagiere der Titanic")
```

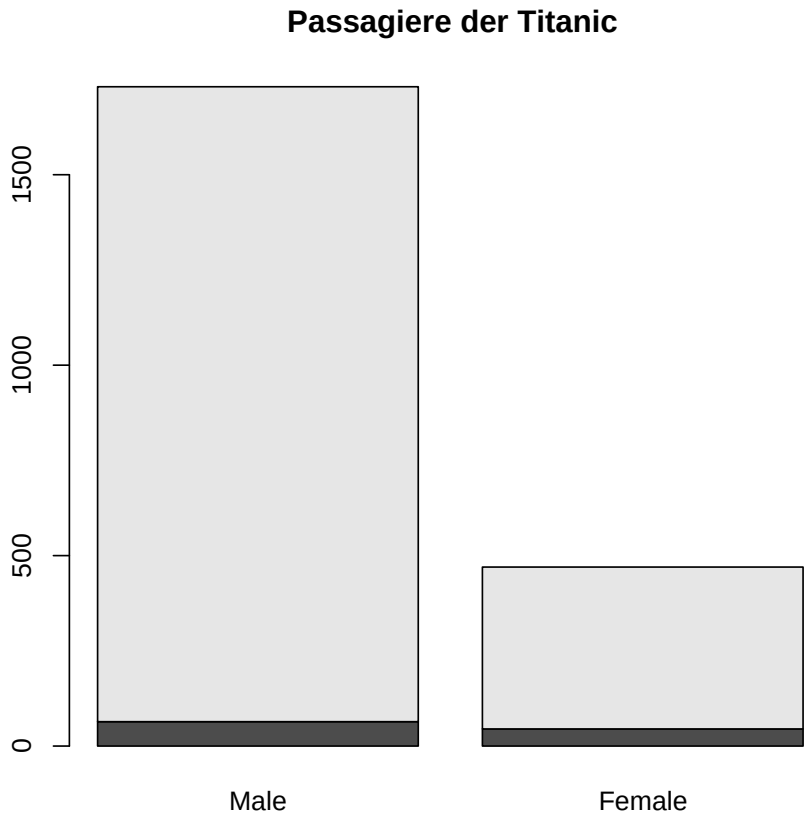


Abbildung 2.41: Gestapelte Balken (zweite Möglichkeit)

```
> barplot(X, beside = TRUE, main = "Passagiere der Titanic")
```

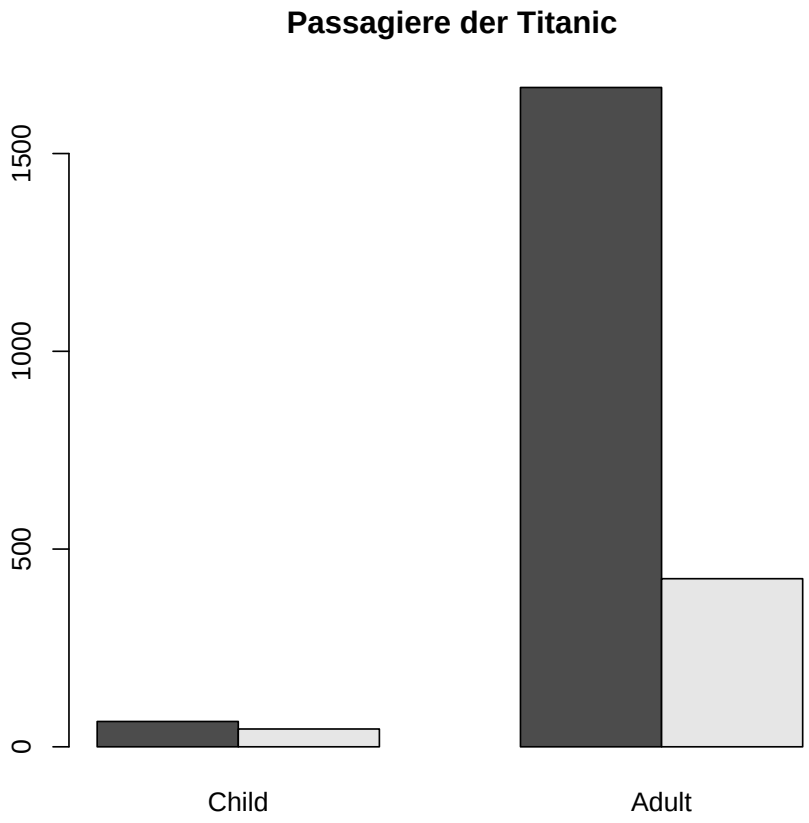


Abbildung 2.42: Parallele Balken

2.7.3 Mosaikplot

Der Mosaikplot eignet sich zur gleichzeitigen Darstellung mehrere kategorieller Variablen. Besonders gut zu erkennen ist jeweils die bedingte Verteilung der späteren Variablen gegeben die früheren.

```
> mosaicplot(X, main = "Passagiere der Titanic")
```

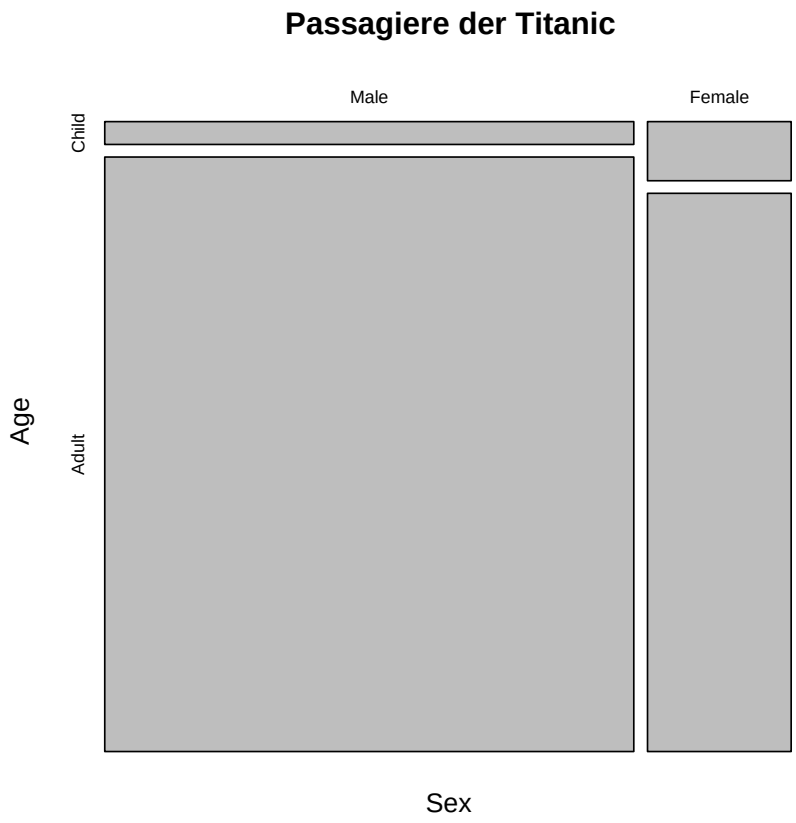


Abbildung 2.43: Mosaikplot (bedingte Balken)

```
> mosaicplot(t(X), main = "Passagiere der Titanic")
```

Abbildung 2.44: Mosaikplot (andere Bedingung)

```
> mosaicplot(Titanic, main = "Passagiere der Titanic")
```

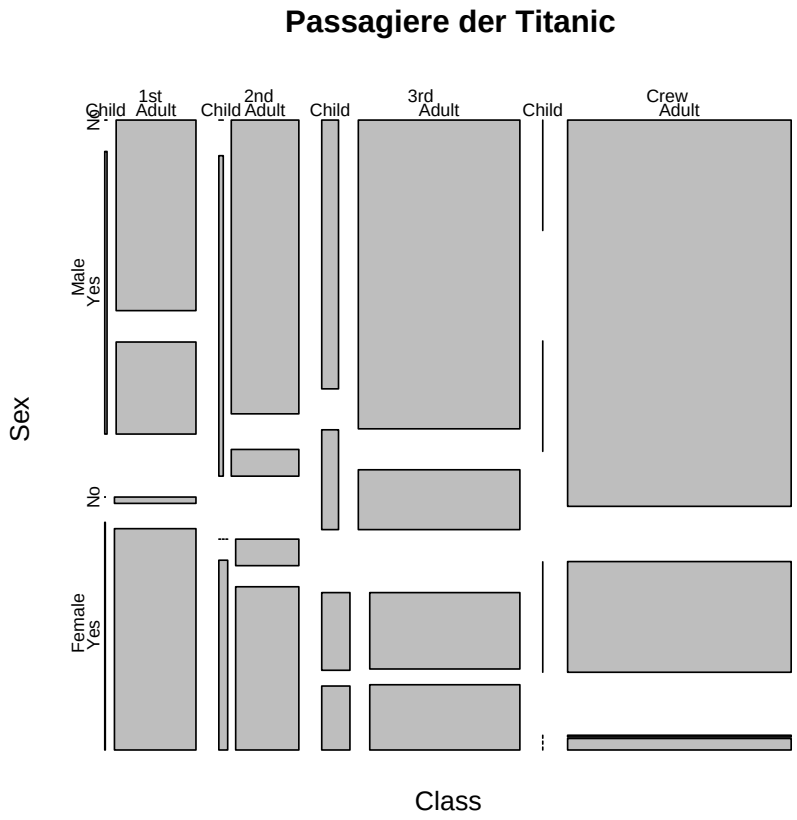


Abbildung 2.45: Mosaikplot (multivariater)