Vorlesung Statistik A
Explorative Datenanalyse

Florian Wickelmaier

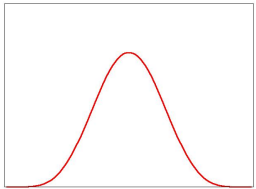
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

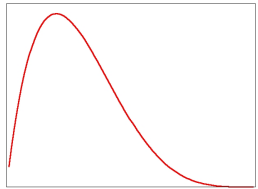


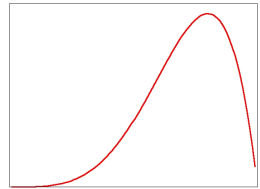
Wintersemester 2026/2027

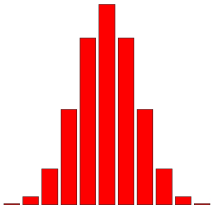
Formen von Häufigkeitsverteilungen

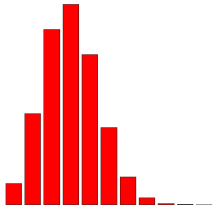
Unimodale Verteilungen

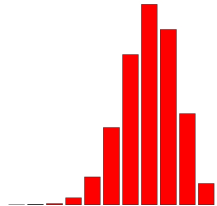












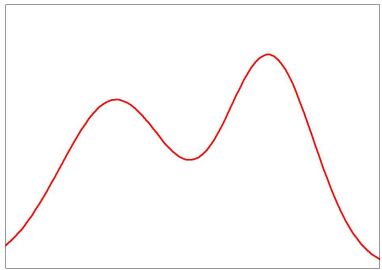
symmetrisch

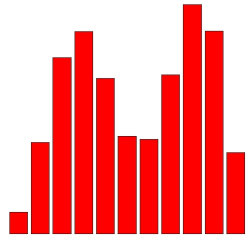
rechtsschief
linkssteil

linksschief
rechtssteil

Formen von Häufigkeitsverteilungen

Bimodale Verteilung





Datenstrukturen in R

Matrizen und Arrays

Generieren einer (3×4)-Matrix, also einer Matrix mit drei Zeilen und vier Spalten

```
A <- matrix(1:12, nrow = 3, ncol = 4, byrow = TRUE)
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

Beschriften der Matrix

```
rownames(A) <- c("a1", "a2", "a3")  # analog: colnames()
```

Matrizen können auch spalten- oder zeilenweise generiert werden

```
cbind(a1 = 1:4, a2 = 5:8, a3 = 9:12)  # "column bind"
rbind(a1 = 1:4, a2 = 5:8, a3 = 9:12)  # "row bind"
```

Arrays

Arrays sind Datenstrukturen mit mehr als zwei Dimensionen

```
array(c(A, 2*A), c(3, 4, 2))  # 3 x 4 x 2 Array
, , 1
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
, , 2
     [,1] [,2] [,3] [,4]
[1,]    2    4    6    8
[2,]   10   12   14   16
[3,]   18   20   22   24
```

Listen

Verschiedene Arten von R-Objekten können in einer Datenstruktur gespeichert werden, die als Listen bezeichnet werden

```
list1 <- list(w = gewicht, h = groesse, s = stat2, A = A)
```

Die Komponenten von Listen werden mit $ und dem Namen der Komponente extrahiert

```
list1$h
[1] 1.75 1.80 1.65 1.90 1.74 1.91

list1$A
     [,1] [,2] [,3] [,4]
a1    1    2    3    4
a2    5    6    7    8
a3    9   10   11   12
```

Dataframes: "fundamentale" Datenstruktur in R

Dataframes sind Listen, die aus Vektoren und Faktoren gleicher Länge bestehen

Die Zeilen im Dataframe beziehen sich auf eine Einheit (Beobachtung oder Person)

```
id  <- factor(paste0("s", 1:6))
dat <- data.frame(id, gewicht, groesse)
```

```
  id gewicht groesse
1 s1      60    1.75
2 s2      72    1.80
3 s3      57    1.65
4 s4      90    1.90
5 s5      95    1.74
6 s6      72    1.91
```

Indizieren von Variablen

(siehe ?Extract)

Die Variablen (Spalten) im Dataframe werden mit $ extrahiert

```
dat$id
[1] s1 s2 s3 s4 s5 s6
Levels: s1 s2 s3 s4 s5 s6
```

Die einzelnen Elemente einer Variable sind durch [] zugänglich

```
gewicht[4]            # 4. Element
gewicht[4] <- 92      # 4. Element veraendern
gewicht[c(1, 2, 6)]   # Elemente 1, 2, 6
gewicht[1:5]          # Elemente 1 bis 5
gewicht[-3]           # ohne Element 3
```

Indizieren von Variablen

Indizes können auch logisch sein

```
gewicht[gewicht > 60]                  # Werte ueber 60
gewicht[gewicht > 60 & gewicht < 80]   # zw. 60 und 80
groesse[gewicht > 60 & gewicht < 80]
```

Logische Bedingungen können mit UND `&`, ODER `|`, GLEICH `==`, UNGLEICH `!=`, bzw. `<`, `<=`, `>`, `>=`, `%in%` gebildet werden

Indizieren von Dataframes

Dataframes haben Zeilen- und Spaltenindizes

Weglassen eines Indexes bedeutet alle Zeilen bzw. Spalten werden ausgewählt

```
dat[3, 2]   # 3. Zeile, 2. Spalte
dat[1:4, ]  # Zeilen 1 bis 4, alle Spalten
dat[, 3]    # alle Zeilen, 3. Spalte
```

Mit logischen Indizes werden Zeilen extrahiert, die bestimmte Bedingungen erfüllen

```
dat[dat$id == "s2", ]      # alle Beob. von s2
dat[dat$gewicht > 60, ]    # alle Beob. ueber 60kg
```

Auch Matrizen und Arrays haben zwei oder mehr Indizes

```
array(c(A, 2*A), c(3, 4, 2))[, , 2]  # 2. "Scheibe"
```

Daten einlesen in R

Textdateien lesen

Textdateien in Tabellen- oder Spreadsheetform (eine Zeile pro Beobachtung, eine Spalte pro Variable) werden mit `read.table()` eingelesen

```
dat <- read.table("D:/experiment/data.txt", header = TRUE)
```

Es wird ein Dataframe erzeugt, numerische Variablen werden zu numerischen Vektoren, nicht-numerische zu Zeichen-Vektoren

Auch unter Windows verwendet man Schrägstriche `/`, um den Dateipfad anzugeben

Häufige Optionen

- `header`: Variablenname in der ersten Zeile?
- `sep`: Welches Trennzeichen zwischen Variablen (white space, Tabulator `\t`, Komma, etc.)?
- `dec`: Dezimalpunkt oder Komma?

Textdateien lesen

Für die Datei muss entweder der komplette Pfad angegeben werden (s. o.) oder das aktuelle Verzeichnis mit `setwd()` gesetzt werden

```
getwd()                 # aktuelles Arbeitsverzeichnis
setwd("C:/Users/...")   # Verzeichnis wechseln
```

Kleinere Datensätze kann man auch direkt in ein R-Skript einfügen

```
dat <- read.table(header = TRUE, text = "
  id gewicht groesse
  s1      60    1.75
  s2      72    1.80
  s3      57    1.65
  s4      90    1.90
  s5      95    1.74
  s6      72    1.91
")
```

Dateien anderer Programme lesen

Daten in anderem Programm als Textdatei abspeichern und dann eine der folgenden Funktionen verwenden

```
read.table()
read.csv()
read.csv2()
read.delim()
read.delim2()
```

Lesen von SPSS-Dateien

```
dat <- foreign::read.spss("file.sav",
                          to.data.frame = TRUE)
```

Weitere Informationen (z. B. bzgl. Excel-Dateien, Datenbanken) im "R Data Import/Export"-Manual

Session Management in R

Arbeitsspeicher

Alle während einer Sitzung erzeugten Objekte werden im Arbeitsspeicher behalten

Nicht mehr verwendete Objekte sollten gelöscht werden

```
ls()              # anzeigen der Objekte
rm(A)             # loescht A
rm(list = ls())   # loescht alles
```

Tipp für einen übersichtlichen Arbeitsspeicher

- Beim Beenden der R-Sitzung mit `q()` die Objekte *nicht* speichern
- Stattdessen die Funktionen, die die Objekte erzeugen, in ein R-Skript schreiben *und dann* die R-Sitzung mit `q("no")` beenden

Arbeitsverzeichnis

Beim Start von R wird das Arbeitsverzeichnis gesetzt

```
getwd()     # get working directory
[1] "D:/"
dir()       # Inhalt des Arbeitsverzeichnisses
```

Tipp: Das Arbeitsverzeichnis auf das aktuelle Verzeichnis setzen; so bleiben die Dateipfade kurz und die Gefahr Daten zu überschreiben wird minimiert

```
setwd("D:/projects/experiment3/analysis")  # set working
                                           #   directory
```

Zusatzpakete

Ein Großteil der Funktionalität von R wird durch Zusatzpakete zur Verfügung gestellt

Einige Pakete werden gemeinsam mit R installiert

```
library()                # Welche Pakete sind installiert?
library(nlme)            # Paket nlme laden
search()                 # Welche Pakete sind geladen?
detach("package:nlme")   # Paket aus Speicher loeschen
```

Pakete werden mit `install.packages()` installiert, sofern eine Verbindung zum Internet besteht