

# Statistik mit R

**1**



# Statistik mit R

|                                                    |    |
|----------------------------------------------------|----|
| R.1 Einführung in R                                | 3  |
| R.2 Einige grundlegende Dinge                      | 6  |
| R.3 R als Taschenrechner                           | 10 |
| R.4 Datenstrukturen: Vektoren, Matrizen und Listen | 13 |
| R.5 Arbeiten mit logischen Operatoren              | 31 |
| R.6 Deskriptive Statistik mit R                    | 36 |

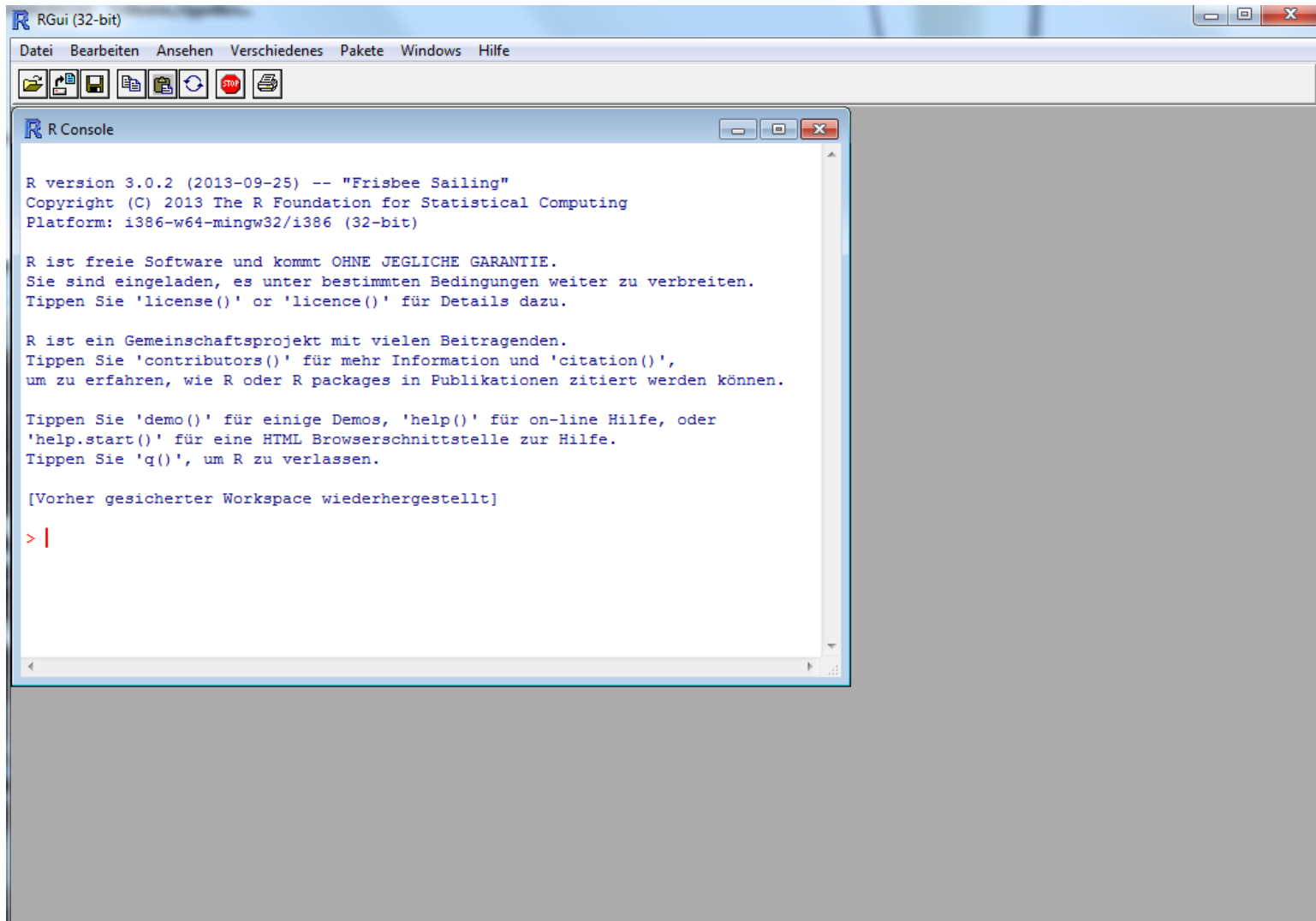
---

## Statistik 2

|                                           |     |
|-------------------------------------------|-----|
| R.7 Wahrscheinlichkeitsverteilungen mit R | 105 |
| R.8 Testen mit R                          | 120 |
| R.9 Regression mit R                      | 135 |

Beachten Sie auch das Skript „Statistik mit R“

# R.1 Einführung in R



```
RGui (32-bit)
Datei Bearbeiten Ansehen Verschiedenes Pakete Windows Hilfe

R Console

R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

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

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

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

[Vorher gesicherter Workspace wiederhergestellt]

> |
```

## R.1 Einführung in R



**John Chambers**

**S**



**S-Plus**



**R**

## R.1 Einführung in R



**R**obert Gentleman



**R**oss Ihaka

## R.2 Einige grundlegende Dinge

# Bereitschafts- und Fortsetzungszeichen

> 1+1                                   > als Prompt (Eingabeaufforderung)

[1] 2

> 1+2+3

[1] 6

> 1+2+3+

+ 4                                   + als Fortsetzungszeichen

[1] 10

> 2\*(2+3

+ )                                   + als Fortsetzungszeichen

[1] 10

# Zuweisungen

> x=0.5                               speichere 0.5 unter dem Namen x

> x                                   Abruf des Wertes von x

[1] 0.5

> x+1

[1] 1.5

> x=1           Überschreiben des alten Werts 0.5 ohne Warnung

## R.2 Einige grundlegende Dinge

```
> x
[1] 1           Ausgabe des neuen Wertes
> x+1
[1] 2
> y=2
> y
[1] 2
> y+3
[1] 5
> x+y
[1] 3
> 1=x        Zuweisungsrichtung: Von rechts nach links!
Fehler in 1 = x : ungültige (do_set) linke Seite in
Zuweisung
> x=y        Zuweisung des Wertes von y nach x
> x
[1] 2
```

## R.2 Einige grundlegende Dinge

# Sichern und Löschen von Objekten

```
> ls()           Listen aller kreierten Objekte  
[1] "x" "y"         im Workspace (Global Environment)  
> z=c(1,2,3)     Speichern mehrerer Werte in einem  
> z             Vektor
```

```
[1] 1 2 3  
> ls()  
[1] "x" "y" "z"
```

```
> rm(z)          Löschen eines Objekts
```

```
> ls()  
[1] "x" "y"  
> z
```

```
Fehler: Objekt 'z' nicht gefunden
```

# Hilfesystem und Literaturhinweise

```
> sum(c(1,2,3))  
[1] 6
```

```
> ?sum
```

```
starting httpd help server ... done
```



## R.2 Einige grundlegende Dinge

```
> help(sum)
> options(help_type="text")      Hilfetext als Textdatei
> ?sum
> x=c(3,2,5,NA,4)                Fehlender Wert (Not Available)
> sum(x)
[1] NA
> ?sum
> sum(x,na.rm=TRUE)              Fehlende Werte werden ignoriert
[1] 14
> ?boxplot
> boxplot(count ~ spray, data = InsectSprays, col = "lightgray")
```

## R.3 R als Taschenrechner

```
> 1.8+2
```

```
[1] 3.8
```

```
> 1.8-2
```

```
[1] -0.2
```

```
> 1.8/3
```

```
[1] 0.6
```

```
> 2+2*3
```

```
[1] 8
```

```
> (2+2)*3
```

```
[1] 12
```

```
> 2^3
```

$2^3$

```
[1] 8
```

```
> 2**3
```

alternativ auch so

```
[1] 8
```

```
> 8^(1/3)
```

```
[1] 2
```

```
> 9^0.5
```

```
[1] 3
```

## R.3 R als Taschenrechner

```
> sqrt(9)           square root  
[1] 3
```

```
> exp(1)  
[1] 2.718282
```

```
> exp(2)  
[1] 7.389056
```

```
> log(7.389056)  
[1] 2
```

```
> log(exp(3))  
[1] 3
```

Basis von log per Default ist Zahl e

```
> ?log
```

```
> log(100,10)       Verwendung der Basis 10  
[1] 2
```

```
> abs(-1.2)  
[1] 1.2
```

## R.3 R als Taschenrechner

```
> sqrt(2)
[1] 1.414214
> round(1.414,2)      Runden auf 2 Nachkommastellen
[1] 1.41
> round(1.414,1)
[1] 1.4
> round(sqrt(2),4)
[1] 1.4142

> pi
[1] 3.141593
> sin(pi/2)
[1] 1
> cos(pi)
[1] -1

> pi/10^15             Kreiszahl dividiert durch 1015
[1] 3.141593e-15        steht für 3.141593 × 10-15
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
##### Vorbemerkung zu Datentypen #####  
> x=2  
> x+1  
[1] 3          numeric... eine Zahl  
> x="2"  
> x  
[1] "2"        character... eine Zeichenkette  
> x+2          damit kann man nicht rechnen!  
Fehler in x + 2 : nicht-numerisches Argument für binären  
Operator  
> x=neu  
Fehler: Objekt 'neu' nicht gefunden  
> x="neu"      Speicherung von Zeichenketten nur in  
> x            Anführungszeichen  
[1] "neu"
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
##### Vektoren #####  
# Konstruktion von Vektoren  
> v1=c(0.8,2.4,-3.4,4,5)  
> v1  
[1] 0.8 2.4 -3.4 4.0 5.0  
> v2=c(1,2,3,v1)  
> v2  
[1] 1.0 2.0 3.0 0.8 2.4 -3.4 4.0 5.0  
> v3=c("a","b","c")  
> v3  
[1] "a" "b" "c"  
  
# Spezielle Befehle zur Erzeugung von Vektoren  
> v=1:5  
> v  
[1] 1 2 3 4 5  
> 6:2  
[1] 6 5 4 3 2  
> 1:12  
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> seq(1,5,by=0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> seq(1,5,le=8)
[1] 1.000000 1.571429 2.142857 2.714286
[5] 3.285714 3.857143 4.428571 5.000000
> rep(1,10)
[1] 1 1 1 1 1 1 1 1 1 1
> rep(c(0,1),4)
[1] 0 1 0 1 0 1 0 1
> rep(1:5,2)
[1] 1 2 3 4 5 1 2 3 4 5

# Zugriff auf Vektorkomponenten
> v1
[1] 0.8 2.4 -3.4 4.0 5.0
> v1[1]
[1] 0.8
> v1[2]
[1] 2.4
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> v1[5]           5. Komponente von v1
[1] 5
> v1[c(2,5,3)]    2., 5. und 3. Komponente von v1
[1] 2.4 5.0 -3.4    (in dieser Reihenfolge!)
> v1[1:4]
[1] 0.8 2.4 -3.4 4.0
> v1[-1]          alle Komponenten außer die 1.
[1] 2.4 -3.4 4.0 5.0
> v1[-c(5,1)]     alle Komponenten außer die 5. und 1.
[1] 2.4 -3.4 4.0
```

# Rechnen mit Vektoren

```
> a=1:5
> b=5:1
> a
[1] 1 2 3 4 5
> b
[1] 5 4 3 2 1
> a+1           Vektor plus Skalar (das geht!)
[1] 2 3 4 5 6
```



## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> a+b
```

Vektor plus Vektor

```
[1] 6 6 6 6 6
```

```
> a*b
```

Vektor mal Vektor (das geht!)

```
[1] 5 8 9 8 5
```

```
> a/2
```

```
[1] 0.5 1.0 1.5 2.0 2.5
```

```
##### Matrizen #####
```

```
# Konstruktion von Matrizen
```

```
> v=1:12
```

```
> v
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

```
> matrix(v,3,4)
```

Matrix mit 3 Zeilen, 4 Spalten

```
      [,1] [,2] [,3] [,4]
```

```
[1,] 1 4 7 10
```

```
[2,] 2 5 8 11
```

```
[3,] 3 6 9 12
```

Zuerst Befüllen der Spalten als Default

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> matrix(v,3,4,byrow=TRUE)
```

Zuerst Befüllen der Zeilen  
als wählbare Option

```
      [,1] [,2] [,3] [,4]  
[1,]    1    2    3    4  
[2,]    5    6    7    8  
[3,]    9   10   11   12
```

# Zugriff auf Matrixkomponenten

```
> m1=matrix(v,3,4,byrow=TRUE)
```

```
> m1
```

```
      [,1] [,2] [,3] [,4]  
[1,]    1    2    3    4  
[2,]    5    6    7    8  
[3,]    9   10   11   12
```

```
> m1[1]      1. Komponente = Zeile 1, Spalte 1
```

```
[1] 1
```

```
> m1[2]      2. Komponente = Zeile 2, Spalte 1
```

```
[1] 5
```

```
> m1[3]      3. Komponente = Zeile 3, Spalte 1
```

```
[1] 9
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> m1[5]           5. Komponente = Zeile 2, Spalte 2
[1] 6
> m1[11]          11. Komponente = Zeile 2, Spalte 4
[1] 8
> m1[1,2]         Zeile 1, Spalte 2
[1] 2
> m1[2,3]         Zeile 2, Spalte 3
[1] 7
> m1[1,]          Zeile 1
[1] 1 2 3 4
> m1[2,]          Zeile 2
[1] 5 6 7 8
> m1[,1]          Spalte 1
[1] 1 5 9
> m1[,4]          Spalte 4
[1] 4 8 12
> m1[1:2,2:3]     Zeilen 1-2, Spalten 2-3
      [,1] [,2]
[1,]    2    3
[2,]    6    7
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> m1[1:2,2:4]
```

Zeilen 1-2, Spalten 2-4

```
      [,1] [,2] [,3]
[1,]     2     3     4
[2,]     6     7     8
```

```
> m1[-1,]
```

Alle Zeilen außer 1

```
      [,1] [,2] [,3] [,4]
[1,]     5     6     7     8
[2,]     9    10    11    12
```

# Rechnen mit Matrizen

```
> m1
```

```
      [,1] [,2] [,3] [,4]
[1,]     1     2     3     4
[2,]     5     6     7     8
[3,]     9    10    11    12
```

```
> m2=matrix(v,3,4)
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> m2
```

|      | [,1] | [,2] | [,3] | [,4] |
|------|------|------|------|------|
| [1,] | 1    | 4    | 7    | 10   |
| [2,] | 2    | 5    | 8    | 11   |
| [3,] | 3    | 6    | 9    | 12   |

```
> m1+m2
```

|      | [,1] | [,2] | [,3] | [,4] |
|------|------|------|------|------|
| [1,] | 2    | 6    | 10   | 14   |
| [2,] | 7    | 11   | 15   | 19   |
| [3,] | 12   | 16   | 20   | 24   |

```
> m1*m2      Keine Matrixmultiplikation im üblichen Sinne!
```

|      | [,1] | [,2] | [,3] | [,4] |
|------|------|------|------|------|
| [1,] | 1    | 8    | 21   | 40   |
| [2,] | 10   | 30   | 56   | 88   |
| [3,] | 27   | 60   | 99   | 144  |

```
> m1%*%m2     Matrixmultiplikation: (3x4)- mal (3x4)-Matrix  
Fehler in m1 %*% m2 : nicht passende Argumente  
Dimensionen passen jedoch nicht zueinander
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> t(m2)
```

Transponierte von m2

```
      [,1] [,2] [,3]
[1,]     1     2     3
[2,]     4     5     6
[3,]     7     8     9
[4,]    10    11    12
```

```
> m1%*%t(m2)
```

Jetzt: (3x4) - mal (4x3) -Matrix

```
      [,1] [,2] [,3]
[1,]    70    80    90
[2,]   158   184   210
[3,]   246   288   330
```

```
> 2*m1
```

Skalar mal Matrix

```
      [,1] [,2] [,3] [,4]
[1,]     2     4     6     8
[2,]    10    12    14    16
[3,]    18    20    22    24
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> m1+1      Matrix plus Skalar (das geht!)
```

```
      [,1] [,2] [,3] [,4]
[1,]    2    3    4    5
[2,]    6    7    8    9
[3,]   10   11   12   13
```

```
##### Listen #####
```

```
# Konstruktion von Listen
```

```
> x=1
```

```
> v=1:5
```

```
> Begrüßung=c("Guten Tag","Hallo","Hi")
```

```
> x
```

```
[1] 1
```

```
> v
```

```
[1] 1 2 3 4 5
```

```
> Begrüßung
```

```
[1] "Guten Tag" "Hallo"      "Hi"
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> Liste=list(x,v,Begrüßung)      Liste aus 3 Objekten
> Liste
[[1]]
[1] 1

[[2]]
[1] 1 2 3 4 5

[[3]]
[1] "Guten Tag" "Hallo"      "Hi"

# Zugriff auf Listenkomponenten
# 1. Möglichkeit
> Liste[[1]]
[1] 1
> Liste[[2]]
[1] 1 2 3 4 5
> Liste[[3]]
[1] "Hallo"      "Guten Tag" "Hi"
```



## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
# 2. Möglichkeit
> Liste=list(Skalar=x,Vektor=v,Gruß=Begrüßung)
> Liste
$Skalar
[1] 1
$Vektor
[1] 1 2 3 4 5
$Gruß
[1] "Hallo"      "Guten Tag"  "Hi"
> Liste$Skalar
[1] 1
> Liste$Vektor
[1] 1 2 3 4 5
> Liste$Gruß
[1] "Hallo"      "Guten Tag"  "Hi"
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
# 3. Möglichkeit
```

```
> attach(Liste)      Temporäres Bekanntmachen der  
> Skalar             Komponentennamen einer Liste
```

```
[1] 1
```

```
> Vektor
```

```
[1] 1 2 3 4 5
```

```
> Gruß
```

```
[1] "Hallo"      "Guten Tag" "Hi,,
```

```
# Achtung! mit attach()
```

```
> Vektor="Störung"   Kreation eines namensgleichen Objekts
```

```
> Skalar
```

```
[1] 1
```

```
> Vektor
```

```
[1] "Störung"      Workspace geht vor attachment!
```

```
> Gruß
```

```
[1] "Hallo"      "Guten Tag" "Hi"
```

```
> rm(Vektor)         Entfernen des Workspace-Objekts
```

```
> Vektor
```

```
[1] 1 2 3 4 5         Listenkomponente rückt dann nach
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> detach(Liste)    Aufheben eines attachments
```

```
> Vektor
```

```
Fehler: Objekt 'Vektor' nicht gefunden
```

```
> Gruß
```

```
Fehler: Objekt 'Gruß' nicht gefunden
```

```
> Vektor="Störung"
```

```
> attach(Liste)
```

```
Das folgende Objekt ist maskiert _by_ .GlobalEnv:
```

```
      Vektor      Hinweis auf namensgleiches Workspace-Objekt
```

```
> Vektor
```

```
[1] "Störung"    Workspace-Objekt hat Vorrang
```

```
##### Data Frames (Spezialformen von Listen) #####
```

```
# Konstruktion von Data Frames
```

```
> rm(list=ls())      Löschen aller Objekte (Klarschiff)
```

```
> Name=c("Anton", "Berta", "Cäsar", "Dagmar", "Emil")
```

```
> Geschlecht=c(0,1,0,1,0)
```

```
> Größe=c(175,170,180,174,162)
```

```
> Gewicht=c(78,72,84,71,56)
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> Studenten=data.frame(Name,Geschlecht,Größe,Gewicht)
```

```
> Studenten
```

|   | Name   | Geschlecht | Größe | Gewicht |
|---|--------|------------|-------|---------|
| 1 | Anton  | 0          | 175   | 78      |
| 2 | Berta  | 1          | 170   | 72      |
| 3 | Cäsar  | 0          | 180   | 84      |
| 4 | Dagmar | 1          | 174   | 71      |
| 5 | Emil   | 0          | 162   | 56      |

```
> dim(Studenten)
```

```
[1] 5 4 # 5 Zeilen (Beobachtungen), 4 Spalten (Variablen)
```

```
> names(Studenten) Namen der Listekomponenten (Variablen)
```

```
[1] "Name" "Geschlecht" "Größe" "Gewicht"
```

```
> summary(Studenten) # Statistische Zusammenfassungen
```

|          | Name     | Geschlecht | Größe    | Gewicht |
|----------|----------|------------|----------|---------|
| Anton :1 | Min.     | :0.0       | Min.     | :162.0  |
| Berta :1 | 1st Qu.: | 0.0        | 1st Qu.: | 170.0   |
| Cäsar :1 | Median   | :0.0       | Median   | :174.0  |
| Dagmar:1 | Mean     | :0.4       | Mean     | :172.2  |
| Emil :1  | 3rd Qu.: | 1.0        | 3rd Qu.: | 175.0   |
|          | Max.     | :1.0       | Max.     | :180.0  |

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
# Zugriff auf Komponenten über Listensyntax
> rm(Name, Geschlecht, Größe, Gewicht)
> Name
Fehler: Objekt 'Name' nicht gefunden
> attach(Studenten)
> Name
[1] Anton  Berta  Cäsar  Dagmar  Emil    Datentyp factor
Levels: Anton Berta Cäsar Dagmar Emil
Vorkommende Faktorstufen (Ausprägungsmöglichkeiten)
> Geschlecht
[1] 0 1 0 1 0
> Größe
[1] 175 170 180 174 162
> Gewicht
[1] 78 72 84 71 56
> detach(Studenten)
> Name
Fehler: Objekt 'Name' nicht gefunden
```

## R.4 Datenstrukturen: Vektoren, Matrizen und Listen

```
> Studenten[[3]]
[1] 175 170 180 174 162
> Studenten$Größe
[1] 175 170 180 174 162

# Zugriff über Matrizensyntax
> Studenten
      Name Geschlecht Größe Gewicht
1  Anton           0   175       78
2  Berta           1   170       72
3  Cäsar           0   180       84
4 Dagmar           1   174       71
5   Emil           0   162       56
> Studenten[,3]
[1] 175 170 180 174 162
> Studenten[2:4,1:3]
      Name Geschlecht Größe
2  Berta           1   170
3  Cäsar           0   180
4 Dagmar           1   174
```

## R.5 Arbeiten mit logischen Operatoren

```
# TRUE wie 1, FALSE wie 0
```

```
> TRUE+TRUE
```

```
[1] 2
```

```
> TRUE+TRUE+TRUE
```

```
[1] 3
```

```
> TRUE+TRUE+TRUE+FALSE
```

```
[1] 3
```

```
> FALSE+FALSE
```

```
[1] 0
```

```
# Vergleichsoperationen und Abzählen mittels sum()
```

```
> x=c(3,1,5,3,6,-1,0)
```

```
> x
```

```
[1] 3 1 5 3 6 -1 0
```

```
> sum(x)
```

```
[1] 17
```

```
> x>0
```

```
[1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE
```

```
> sum(x>0)
```

```
[1] 5
```

## R.5 Arbeiten mit logischen Operatoren

```
> x>=0
[1] TRUE TRUE TRUE TRUE TRUE FALSE TRUE
> sum(x>=0)
[1] 6
> x<0
[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE
> x==0      Prüfung einer Gleichheit (keine Zuweisung!)
[1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE
> x!=0      Prüfung einer Ungleichheit
[1] TRUE TRUE TRUE TRUE TRUE TRUE FALSE

# UND und ODER
> x>=3&x<=5
[1] TRUE FALSE TRUE TRUE FALSE FALSE FALSE
> x==3|x==5
[1] TRUE FALSE TRUE TRUE FALSE FALSE FALSE
```



## R.5 Arbeiten mit logischen Operatoren

```
# Bedingter Zugriff auf Komponenten
> x
[1] 3 1 5 3 6 -1 0
> x[c(TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,FALSE)]
[1] 3 1 3 6
> x>2
[1] TRUE FALSE TRUE TRUE TRUE FALSE FALSE
> x[x>2]      reflexiver Zugriff
[1] 3 5 3 6
> y=c(0,0,0,0,1,1,1)
> y==1
[1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE
> x[y==1]     projektiver Zugriff
[1] 6 -1 0
```

## R.5 Arbeiten mit logischen Operatoren

```
> Studenten
```

|   | Name   | Geschlecht | Größe | Gewicht |
|---|--------|------------|-------|---------|
| 1 | Anton  | 0          | 175   | 78      |
| 2 | Berta  | 1          | 170   | 72      |
| 3 | Cäsar  | 0          | 180   | 84      |
| 4 | Dagmar | 1          | 174   | 71      |
| 5 | Emil   | 0          | 162   | 56      |

```
> attach(Studenten)
```

```
> Name[Geschlecht==1]
```

```
[1] Berta Dagmar
```

Namen der Frauen

```
Levels: Anton Berta Cäsar Dagmar Emil
```

```
> Größe[Geschlecht==0]
```

```
[1] 175 180 162
```

Größen der Männer

```
> Größe[Geschlecht==1]
```

```
[1] 170 174
```

Größen der Frauen

```
# Bedingte Analysen z. B. mit mean() und table()
```

```
> mean(Größe[Geschlecht==0])
```

```
[1] 172.3333
```

Durchschnittsgröße der Männer

## R.5 Arbeiten mit logischen Operatoren

```
> mean (Größe[Geschlecht==1])  
[1] 172                                Durchschnittsgröße der Frauen  
  
> table (Geschlecht)  
Geschlecht  
0 1  
3 2                                Absolute Häufigkeitsverteilung  
  
> prop.table (table (Geschlecht))  
Geschlecht  
0 1  
0.6 0.4                            Relative Häufigkeitsverteilung  
  
> Geschlecht[Größe>170]    Geschlecht der über 170cm Großen  
[1] 0 0 1  
  
> abv=table (Geschlecht[Größe>170])  
> abv                                Absolute bedingte Verteilung  
0 1  
2 1  
  
> rbv=prop.table (abv)  
> rbv                                Relative bedingte Verteilung  
0 1  
0.6666667 0.3333333
```

## R.6 Deskriptive Statistik mit R

```
# length(), sum(), mean(), table()
> x=c(1,1,5,2,1,5,2,3,10,2)
> length(x)      Länge des Vektors (Anzahl der Komponenten)
[1] 10
> sum(x)          Summe
[1] 32
> mean(x)         Mittelwert
[1] 3.2
> table(x)        Eindimensionale absolute
x                Häufigkeitsverteilung
 1  2  3  5 10
 3  3  1  2  1
> prop.table(x)   das ist i.A. nicht sinnvoll
[1] 0.03125 0.03125 0.15625 0.06250 0.03125
[6] 0.15625 0.06250 0.09375 0.31250 0.06250
> prop.table(table(x)) Relative Häufigkeitsverteilung
x
 1  2  3  5 10
0.3 0.3 0.1 0.2 0.1
```

## R.6 Deskriptive Statistik mit R

```
# Arithmetisches Mittel, Median und Quantile
> x=c(1,1,5,2,1,5,2,3,10,2)
> mean(x)                      Arithmetisches Mittel
[1] 3.2
> median(x)                    Median
[1] 2
> sort(x)
[1] 1 1 1 2 2 2 3 5 5 10
> quantile(x,type=2)           type=2 entspricht unserer Definition
 0%   25%   50%   75%  100%
  1     1     2     5   10
> quantile(x,0.3,type=2)       0.3-Quantil
30%
1.5
> quantile(x,0.35,type=2)      0.35-Quantil
35%
2
> quantile(x,probs=seq(0.1,0.9,by=0.1),type=2)
10% 20% 30% 40% 50% 60% 70% 80% 90%
1.0 1.0 2.0 2.0 2.0 2.5 5.0 5.0 7.5
```

## R.6 Deskriptive Statistik mit R

```
# Varianz und Standardabweichung
> x
[1] 1 1 5 2 1 5 2 3 10 2
> x^2
[1] 1 1 25 4 1 25 4 9 100 4
> x-mean(x)
[1] -2.2 -2.2 1.8 -1.2 -2.2 1.8 -1.2 -0.2 6.8 -1.2
> (x-mean(x))^2
[1] 4.84 4.84 3.24 1.44 4.84 3.24 1.44 0.04 46.24 1.44
> mean((x-mean(x))^2)  Varianz von Hand (originäre Formel)
[1] 7.16
> mean(x^2)-mean(x)^2  von Hand via Verschiebungsformel
[1] 7.16
> var(x)  Varianz mittels var()
[1] 7.955556
> 9/10*var(x)  ... verwendet Divisor n-1
[1] 7.16
```

## R.6 Deskriptive Statistik mit R

```
> sqrt(mean(x^2) - mean(x)^2)  Standardabweichung von Hand
[1] 2.675818
> sd(x)                        Standardabweichung mittels sd()
[1] 2.820559                    (Standard Deviation)
> sqrt(var(x))                 ... verwendet Divisor n-1
[1] 2.820559

# Mittlere absolute Abweichungen
> x
[1] 1 1 5 2 1 5 2 3 10 2
> median(x)
[1] 2
> x - median(x)
[1] -1 -1 3 0 -1 3 0 1 8 0
> abs(x - median(x))
[1] 1 1 3 0 1 3 0 1 8 0
> mean(abs(x - median(x)))     MAA vom Median
[1] 1.8
> mean(abs(x - mean(x)))      MAA vom arithmetischen Mittel
[1] 2.08
```

## R.6 Deskriptive Statistik mit R

```
# MAD (Median Absolute Deviation)
> median(abs(x - median(x))) MAD von Hand
[1] 1
> mad(x)                                MAD mittels mad()
[1] 1.4826
> ?mad
> mad(x, constant=1)    Tatsächlicher MAD wird mit 1.4826
[1] 1                    multipliziert (theoretische Gründe)
                        deshalb muss const=1 gewählt werden
                        um originären MAD zu erhalten
```



## R.6 Deskriptive Statistik mit R

```
# Analyse von Kontingenztabellen
```

```
# Beispiel 1: Kontingenztafel wird mit table() erzeugt
```

```
> Geschlecht=c(0,0,0,0,1,1,1,1,0,0,1,1,1,1,0,0)
```

```
> Raucherstatus=c(0,1,2,0,1,2,0,0,1,1,2,0,1,0,2,0)
```

```
> table(Geschlecht,Raucherstatus)
```

|            | Raucherstatus |   |   | Zweidimensionale absolute<br>Häufigkeitsverteilung |
|------------|---------------|---|---|----------------------------------------------------|
| Geschlecht | 0             | 1 | 2 |                                                    |
| 0          | 3             | 3 | 2 |                                                    |
| 1          | 4             | 2 | 2 |                                                    |

```
> Tabelle=table(Geschlecht,Raucherstatus)
```

```
> prop.table(Tabelle)
```

|            | Raucherstatus |        |        | Zweidimensionale relative<br>Häufigkeitsverteilung |
|------------|---------------|--------|--------|----------------------------------------------------|
| Geschlecht | 0             | 1      | 2      |                                                    |
| 0          | 0.1875        | 0.1875 | 0.1250 |                                                    |
| 1          | 0.2500        | 0.1250 | 0.1250 |                                                    |

```
> prop.table(Geschlecht,Raucherstatus)
```

```
Fehler in margin.table(x, margin) : 'x' ist kein Array
```

```
das funktioniert übrigens nicht! Für prop.table()
```

```
muss table- oder matrix-Objekt vorliegen
```

## R.6 Deskriptive Statistik mit R

```
> Tabelle
```

```
      Raucherstatus  
Geschlecht 0 1 2  
      0 3 3 2  
      1 4 2 2
```

```
> prop.table(Tabelle,1)      Bedingte Verteilungen  
                                in den Zeilen
```

```
      Raucherstatus  
Geschlecht      0      1      2  
      0 0.375 0.375 0.250  
      1 0.500 0.250 0.250
```

```
> prop.table(Tabelle,2)      Bedingte Verteilungen  
                                in den Spalten
```

```
      Raucherstatus  
Geschlecht      0      1      2  
      0 0.4285714 0.6000000 0.5000000  
      1 0.5714286 0.4000000 0.5000000
```

## R.6 Deskriptive Statistik mit R

# Beispiel 2: Kontingenztafel wird direkt eingegeben

*Tabelle 5.1.9: Tatsächliche und zu erwartende Häufigkeiten  
mit jeweiligen Abweichungen*

|          | Raucher         | Gelegenheits-<br>raucher | Nichtraucher      | Summe |
|----------|-----------------|--------------------------|-------------------|-------|
| weiblich | 4 (6.4)<br>-2.4 | 8 (8)<br>0               | 28 (25.6)<br>2.4  | 40    |
| männlich | 12 (9.6)<br>2.4 | 12 (12)<br>0             | 36 (38.4)<br>-2.4 | 60    |
| Summe    | 16              | 20                       | 64                | 100   |

$$\chi^2 = 1.875 \quad \Rightarrow \quad C = \sqrt{\frac{\chi^2}{\chi^2 + n}} = \sqrt{\frac{1.875}{1.875 + 100}} \approx 0.1357$$

$$M = \min(2,3) = 2 \quad \Rightarrow \quad C_K = 0.1357 \times \sqrt{2/1} \approx 0.1919$$

## R.6 Deskriptive Statistik mit R

```
> m=c(4,12,8,12,28,36)
> m=matrix(m,2,3)
> m
```

|      | [,1] | [,2] | [,3] |
|------|------|------|------|
| [1,] | 4    | 8    | 28   |
| [2,] | 12   | 12   | 36   |

```
> chisq.test(m) Theoretische Hintergründe hierzu
                dann in Statistik 2
                Pearson's Chi-squared test

data:  m
X-squared = 1.875, df = 2, p-value = 0.3916
> res=chisq.test(m)      res ist jetzt eine Liste
> names(res)             Namen der Listekomponenten
[1] "statistic" "parameter" "p.value"  "method"    "data.name" "observed"
[7] "expected"  "residuals" "stdres"
> attach(res)           Zugriff mittels attach
> statistic
X-squared              Das ist der Chiquadrat-Koeffizient!
1.875
```

## R.6 Deskriptive Statistik mit R

```
> observed                                beobachtete Häufigkeiten
      [,1] [,2] [,3]
[1,]     4     8    28
[2,]    12    12    36

> expected                                zu erwartende Häufigkeiten
      [,1] [,2] [,3]
[1,]   6.4     8 25.6
[2,]   9.6    12 38.4

> n=sum(m)
> n
> C=sqrt(statistic/(statistic+n))      Kontingenzkoeffizient
> C
X-squared      nicht irritieren lassen durch dieses Attribut
0.1356647
> CK=sqrt(2)*C      Korrigierter Kontingenzkoeffizient
> CK
X-squared      nicht irritieren lassen durch dieses Attribut
0.1918588
```

## R.6 Deskriptive Statistik mit R

### # Beispiel 3: Sonderfall 2x2-Tabellen

|          | Raucher | Gelegenheits-<br>raucher |
|----------|---------|--------------------------|
| weiblich | 4       | 8                        |
| männlich | 12      | 12                       |

$$\chi^2 = \frac{36 \times (4 \times 12 - 12 \times 8)^2}{16 \times 20 \times 12 \times 24} = 0.9$$

```
> m=c(4,12,8,12)
> m=matrix(m,2,2)
> m
```

```
      [,1] [,2]
[1,]    4    8
[2,]   12   12
```

## R.6 Deskriptive Statistik mit R

```
> chisq.test(m)
```

```
Pearson's Chi-squared test with Yates'
continuity correction
```

```
data: m
```

```
X-squared = 0.3516, df = 1, p-value = 0.5532
```

```
> chisq.test(m)$statistic
```

```
X-squared
```

```
0.3515625      Wert entspricht nicht Formel
                (tiefergehende theoretische Gründe)
```

```
> chisq.test(m,correct=FALSE)  setze deshalb correct=FALSE
```

```
Pearson's Chi-squared test
```

```
data: m
```

```
X-squared = 0.9, df = 1, p-value = 0.3428
```

## R.6 Deskriptive Statistik mit R

### # Korrelation nach Pearson

Gegeben: (2, 1), (4, 2), (4, 3), (5, 2), (6, 4), (8, 5), (9, 6), (10, 4), (4, 5), (7, 3)

$$r_{XY} = \frac{\tilde{s}_{XY}}{\tilde{s}_X \tilde{s}_Y} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}}$$

*Tabelle 5.2.2: Arbeitstabelle zur Berechnung der Korrelation nach Pearson*

| $i$   | $x_i$ | $y_i$ | $x_i^2$ | $y_i^2$ | $x_i y_i$ |
|-------|-------|-------|---------|---------|-----------|
| 1     | 2     | 1     | 4       | 1       | 2         |
| 2     | 4     | 2     | 16      | 4       | 8         |
| 3     | 4     | 3     | 16      | 9       | 12        |
| 4     | 5     | 2     | 25      | 4       | 10        |
| 5     | 6     | 4     | 36      | 16      | 24        |
| 6     | 8     | 5     | 64      | 25      | 40        |
| 7     | 9     | 6     | 81      | 36      | 54        |
| 8     | 10    | 4     | 100     | 16      | 40        |
| 9     | 4     | 5     | 16      | 25      | 20        |
| 10    | 7     | 3     | 49      | 9       | 21        |
| Summe | 59    | 35    | 407     | 145     | 231       |



## R.6 Deskriptive Statistik mit R

```
> x=c(2,4,4,5,6,8,9,10,4,7)
> y=c(1,2,3,2,4,5,6,4,5,3)
> x
[1] 2 4 4 5 6 8 9 10 4 7
> y
[1] 1 2 3 2 4 5 6 4 5 3
> mean(x)
[1] 5.9
> mean(y)
[1] 3.5
> x*y
[1] 2 8 12 10 24 40 54 40 20 21
> mean(x*y) - mean(x) * mean(y)      Kovarianz von Hand
[1] 2.45                             via Verschiebungsformel

> cov(x,y)                           Kovarianz mit cov()
[1] 2.722222                          verwendet Divisor n-1
> 9/10*cov(x,y)
[1] 2.45
```

## R.6 Deskriptive Statistik mit R

```
> covxy=mean(x*y) -mean(x) *mean(y)
> sx2=mean(x^2) -mean(x) ^2          Varianz von Hand
> sy2=mean(y^2) -mean(y) ^2          Varianz von Hand
> covxy/sqrt(sx2*sy2)                Korrelation von Hand
[1] 0.6730033

> cor(x,y)                           Korrelation mit cor()
[1] 0.6730033

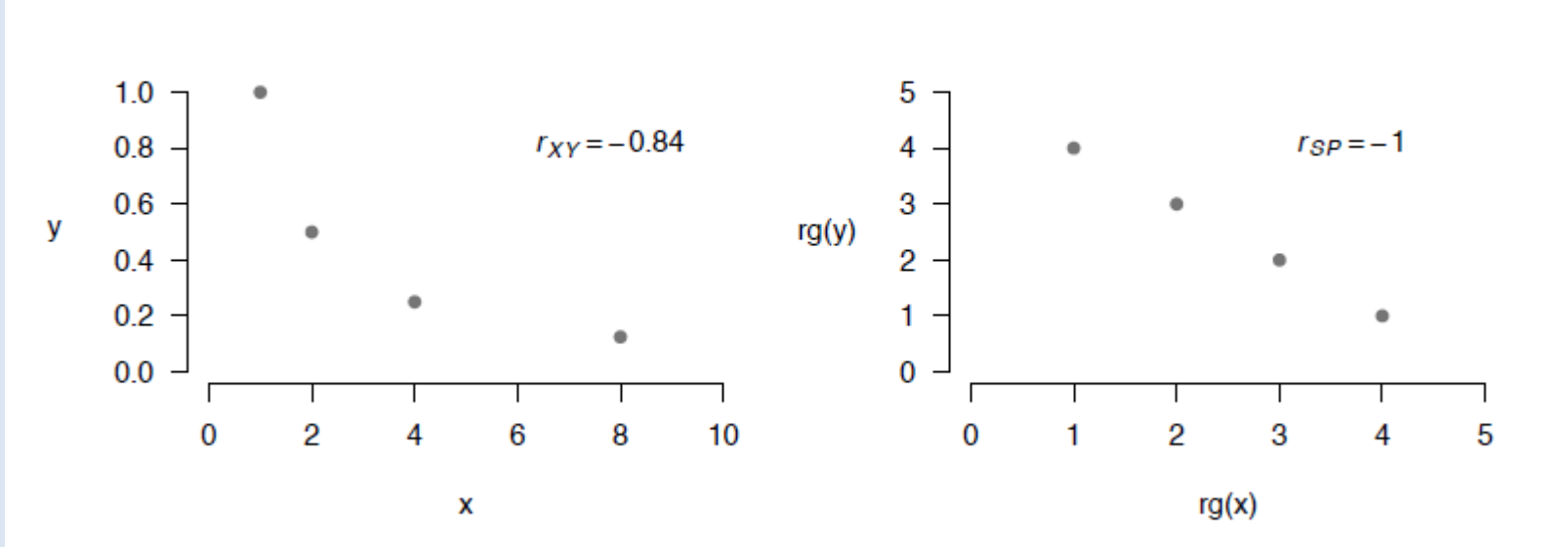
> cov(x,y)/sqrt(var(x)*var(y))       Korrelation mit cov() und
[1] 0.6730033                        var()

> cov(x,y)/(sd(x)*sd(y))             Korrelation mit cov() und
[1] 0.6730033                        sd()
```

## R.6 Deskriptive Statistik mit R

### # Korrelation nach Spearman

Abb. 5.2.11: Streudiagramm von Ursprungswerten und zugehörigen Rangwerten



```
> x=c(1,2,4,8)
```

```
> y=c(1,0.5,0.25,0.125)
```

```
> cor(x,y)
```

Korrelation nach Pearson

```
[1] -0.8434783
```

## R.6 Deskriptive Statistik mit R

```
> rank(x)
[1] 1 2 3 4
> rank(y)
[1] 4 3 2 1
> cor(rank(x), rank(y))      Korrelation nach Spearman
[1] -1

> x=c(1,2,3,2,4,5,6,4,5,3)   hier liegen Bindungen vor
> sort(x)
[1] 1 2 2 3 3 4 4 5 5 6       so sieht man es genauer
> rank(x)
[1] 1.0 2.5 4.5 2.5 6.5 8.5 10.0 6.5 8.5 4.5
Bildung von Durchschnittsrängen
```

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Kovarianz- und Korrelationsmatrizen
```

```
> v1=c(5,1,4,NA,5,6,3)      Variable 1: 1 fehlender Wert
```

```
> v2=c(5,NA,2,NA,NA,8,10)   Variable 2: 3 fehlende Werte
```

```
> v3=c(NA,3,2,3,1,3,6)     Variable 3: 1 fehlender Wert
```

```
> v123=data.frame(v1,v2,v3)
```

```
> v123
```

|   | v1 | v2 | v3 |
|---|----|----|----|
| 1 | 5  | 5  | NA |
| 2 | 1  | NA | 3  |
| 3 | 4  | 2  | 2  |
| 4 | NA | NA | 3  |
| 5 | 5  | NA | 1  |
| 6 | 6  | 8  | 3  |
| 7 | 3  | 10 | 6  |

insgesamt nur 3 komplett vollständige  
3-dimensionale Beobachtungswerte

```
> cov(v123)      Kovarianzmatrix
```

|    | v1 | v2 | v3 |
|----|----|----|----|
| v1 | NA | NA | NA |
| v2 | NA | NA | NA |
| v3 | NA | NA | NA |

fehlende Werte erzeugen fehlende Werte!

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
> cov(v123,use="complete.obs")
```

Kovarianzmatrix

|    | v1         | v2         | v3         |
|----|------------|------------|------------|
| v1 | 2.3333333  | -0.3333333 | -1.8333333 |
| v2 | -0.3333333 | 17.3333333 | 7.3333333  |
| v3 | -1.8333333 | 7.3333333  | 4.3333333  |

beruhend auf nur komplett vollständigen Beobachtungen

```
> cov(v123,use="pairwise.complete.obs")
```

Kovarianzmatrix

|    | v1    | v2        | v3        |
|----|-------|-----------|-----------|
| v1 | 3.20  | -0.500000 | -1.250000 |
| v2 | -0.50 | 12.250000 | 7.333333  |
| v3 | -1.25 | 7.333333  | 2.800000  |

beruhend auf paarweise vollständigen Beobachtungen

```
> cor(v123,use="pairwise.complete.obs")
```

Korrelationsmatrix

|    | v1         | v2         | v3         |
|----|------------|------------|------------|
| v1 | 1.0000000  | -0.1106567 | -0.3473563 |
| v2 | -0.1106567 | 1.0000000  | 0.8461538  |
| v3 | -0.3473563 | 0.8461538  | 1.0000000  |

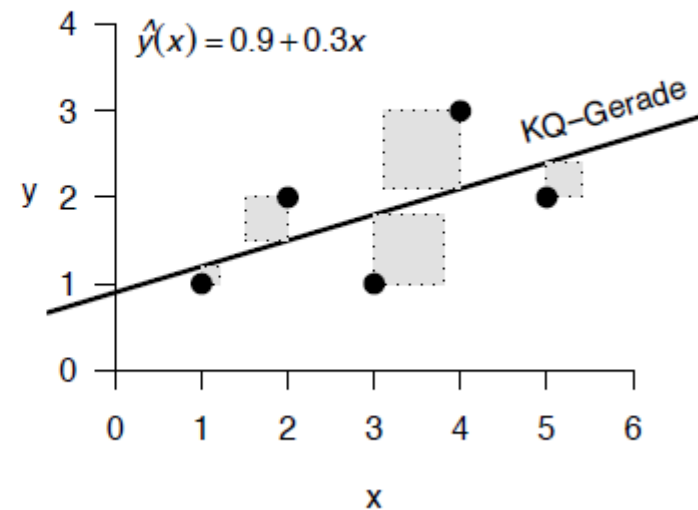
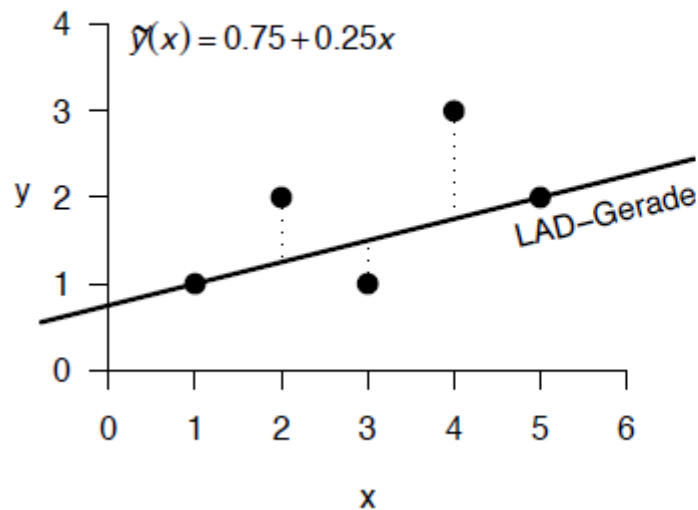
beruhend auf paarweise vollständigen Beobachtungen

## R.6 Deskriptive Statistik mit R

### # KQ-Regression und LAD-Regression

Beobachtungswerte: (1, 1), (2, 2), (3, 1), (4, 3), (5, 2)

Abb. 5.2.15: Minimierung absoluter oder quadratischer Abweichungen



## R.6 Deskriptive Statistik mit R

Tabelle 5.2.4: Arbeitstabelle zur Bestimmung einer KQ-Geraden

| $i$   | $x_i$ | $y_i$ | $x_i y_i$ | $x_i^2$ | $\hat{y}_i$ | $\hat{u}_i$ | $ \hat{u}_i $ | $\hat{u}_i^2$ |
|-------|-------|-------|-----------|---------|-------------|-------------|---------------|---------------|
| 1     | 1     | 1     | 1         | 1       | 1.2         | -0.2        | 0.2           | 0.04          |
| 2     | 2     | 2     | 4         | 4       | 1.5         | 0.5         | 0.5           | 0.25          |
| 3     | 3     | 1     | 3         | 9       | 1.8         | -0.8        | 0.8           | 0.64          |
| 4     | 4     | 3     | 12        | 16      | 2.1         | 0.9         | 0.9           | 0.81          |
| 5     | 5     | 2     | 10        | 25      | 2.4         | -0.4        | 0.4           | 0.16          |
| Summe | 15    | 9     | 30        | 55      | 9           | 0           | 2.8           | 1.9           |

$$\bar{x} = \frac{15}{5} = 3$$

$$\bar{y} = \frac{9}{5} = 1.8$$

$$\hat{b}_1 = \frac{\tilde{s}_{XY}}{\tilde{s}_X^2} = \frac{\frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y}}{\frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2} = \frac{30/5 - 3 \times 1.8}{55/5 - 3^2} = 0.3$$

$$\hat{b}_0 = \bar{y} - \hat{b}_1 \bar{x} = 1.8 - 0.3 \times 3 = 0.9$$

$$\Rightarrow \hat{y}(x) = \hat{b}_0 + \hat{b}_1 x = 0.9 + 0.3x$$



## R.6 Deskriptive Statistik mit R

```
# Berechnung der KQ-Geraden
```

```
> x=c(1,2,3,4,5)
```

```
> y=c(1,2,1,3,2)
```

```
> b1dach=cov(x,y)/var(x)
```

Steigung von Hand

```
> b0dach=mean(y)-b1dach*mean(x)
```

Achsenabschnitt von Hand

```
> b0dach
```

```
[1] 0.9
```

```
> b1dach
```

```
[1] 0.3
```

```
> lm(y~x)
```

KQ-Regression mit lm()

```
Call:
```

Standard-Output

```
lm(formula = y ~ x)
```

```
Coefficients:
```

| (Intercept) | x   |
|-------------|-----|
| 0.9         | 0.3 |

## R.6 Deskriptive Statistik mit R

```
> 0.9+0.3*x  
[1] 1.2 1.5 1.8 2.1 2.4
```

```
> ydach=0.9+0.3*x
```

KQ-gefittete Werte von Hand

```
> ydach
```

```
[1] 1.2 1.5 1.8 2.1 2.4
```

```
> udach=y-ydach
```

KQ-Residuen von Hand

```
> udach
```

```
[1] -0.2  0.5 -0.8  0.9 -0.4
```

```
> reslm=lm(y~x)
```

reslm ist jetzt eine Liste

```
> names(reslm)
```

Namen der Listenkomponenten

```
[1] "coefficients" "residuals" "effects" "rank"  
[5] "fitted.values" "assign" "qr" "df.residual"  
[9] "xlevels" "call" "terms" "model"
```

```
> attach(reslm)
```

Zugriff via attach

```
> coefficients
```

KQ-Koeffizienten

```
(Intercept)      x  
          0.9      0.3
```

## R.6 Deskriptive Statistik mit R

```
> fitted.values      KQ-gefittete Werte
```

```
  1    2    3    4    5  
1.2 1.5 1.8 2.1 2.4
```

```
> residuals          KQ-Residuen
```

```
  1    2    3    4    5  
-0.2  0.5 -0.8  0.9 -0.4
```

Eigenschaften der KQ-Regression (vgl. Folie 224)

```
> sum(fitted.values)
```

```
[1] 9
```

```
> sum(y)
```

```
[1] 9
```

$$\sum_{i=1}^n \hat{y}_i = \sum_{i=1}^n y_i$$

```
> sum(residuals)
```

```
[1] 8.326673e-17
```

muss exakt 0 sein!

$$\sum_{i=1}^n \hat{u}_i = 0$$

Musterbeispiel eines numerisches Artefakts

## R.6 Deskriptive Statistik mit R

```
# Berechnung der LAD-Geraden
```

```
> library(quantreg)           Installieren und Laden des Pakets  
                                quantreg (quantile regression)
```

```
> rq(y~x)                     LAD-Regression mit rq()
```

```
Call:                          Standard-Output  
rq(formula = y ~ x)
```

```
Coefficients:
```

```
(Intercept)          x  
      0.75          0.25
```

```
Degrees of freedom: 5 total; 3 residual
```

```
> yschlange=0.75+0.25*x
```

```
> yschlange                   LAD-gefittete Werte von Hand
```

```
[1] 1.00 1.25 1.50 1.75 2.00
```

## R.6 Deskriptive Statistik mit R

```
> uschlange=y-yschlange      LAD-Residuen von Hand
> uschlange
[1]  0.00  0.75 -0.50  1.25  0.00
> resrq=rq(y~x)      resrq ist jetzt eine Liste
> names(resrq)      Namen der Listenkomponenten
[1] "coefficients"  "x"          "y"          "residuals"  "dual"
[6] "fitted.values" "formula"    "terms"      "xlevels"    "call"
[11] "tau"          "rho"       "method"     "model"
> attach(resrq)      Zugriff via attach
Die folgenden Objekte sind maskiert _by_ .GlobalEnv:

      x, y
Die folgenden Objekte sind maskiert from res:

      method, residuals      Warnung da reslm noch attached
                               war (vgl. Folie 58)
> coefficients
(Intercept)          x
      0.75          0.25
```

## R.6 Deskriptive Statistik mit R

```
> fitted.values
```

LAD-gefittete Werte

```
  1    2    3    4    5  
1.00 1.25 1.50 1.75 2.00
```

```
> residuals
```

LAD-Residuen

```
  1    2    3    4    5  
0.00 0.75 -0.50 1.25 0.00
```

Eigenschaften der KQ-Regression nicht zwingend erfüllt!

```
> sum(fitted.values)
```

```
[1] 7.5
```

```
> sum(y)
```

```
[1] 9
```

$$\sum_{i=1}^n \tilde{y}_i \neq \sum_{i=1}^n y_i$$

```
> sum(residuals)
```

```
[1] 1.5
```

$$\sum_{i=1}^n \tilde{u}_i \neq 0$$

## R.6 Deskriptive Statistik mit R

```
##### Grafiken mit R #####  
# High-Level und Low-Level-Grafikfunktionen und ihre  
# Parameter am Beispiel eines Streudiagramms
```

1: Unmodifizierter High-Level-Grafikbefehl `plot()`

```
> x=c(1,2,3,4,5)  
> y=c(1,2,1,3,2)  
> plot(x,y)
```

2: Modifizierter High-Level-Grafikbefehl `plot()`

```
> plot(x,y,  
+ pch=16,           Ausgefüllte Punkte  
+ cex=2,            Doppelte Größe der Punkte  
+ xlim=c(0,6),ylim=c(0,4), Wertebereiche der Achsen  
+ xlab="x-Werte",ylab="y-Werte", Achsenbeschriftungen  
+ las=1,            las=1: Beschriftung horizontal  
+ cex.axis=1.5,cex.lab=1.5,cex.main=1.5, Vergrößerungen  
+ main="Streudiagramm") (Haupt-)Überschrift
```

## R.6 Deskriptive Statistik mit R

### 3: Modifizierter High-Level-Befehl mit diversen Low-Level-Zusätzen

Modifizierter High-Level-Grafikbefehl `plot()` und ...

```
> plot(x,y,  
+ pch=16,  
+ cex=2,  
+ xlim=c(0,6),ylim=c(0,4),  
+ xlab="",ylab="",           keine Achsenbeschriftungen  
+ axes=FALSE)               Unterdrückung der Achsen
```

... Low-Level-Grafikbefehle

```
> axis(1,cex.axis=1.5,pos=0)
```

Achse (1: Achse unten)

Bündigkeit an der Stelle `y=0`

```
> axis(2,cex.axis=1.5,las=1,pos=0)
```

Achse (2: Achse links)

Bündigkeit an der Stelle `x=0`



## R.6 Deskriptive Statistik mit R

```
> abline(a=0.9,b=0.3,lwd=3)
```

Gerade mit Achsenabschnitt a und Steigung b

lwd=3: dreifache Linienstärke

```
> text(6.5,0,"x",cex=1.5,xpd=TRUE)
```

Text an der Stelle (6.5, 0)

xpd=TRUE: Position außerhalb des Achsenbereichs

```
> text(0,4.5,"y",cex=1.5,xpd=TRUE)
```

```
> points(3,1.8,pch=8,cex=1.5)
```

Punkt an der Stelle (3,1.8)

pch=8: Sternsymbol

```
> points(c(1,2,3,4,5),c(1.2,1.5,1.8,2.1,2.4),pch=4,cex=1.5)
```

Punkte mit x- und y-Koordinaten in separaten Vektoren

```
> segments(0,1.8,3,1.8,lty=2,lwd=1.5)
```

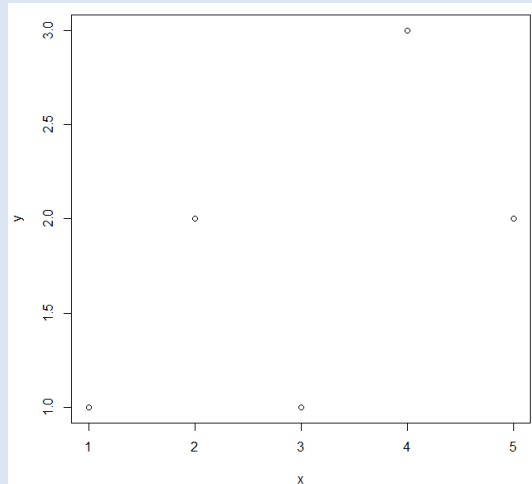
Linie von (0,1.8) bis (3,1.8)

lty=2: Linientyp gestrichelt

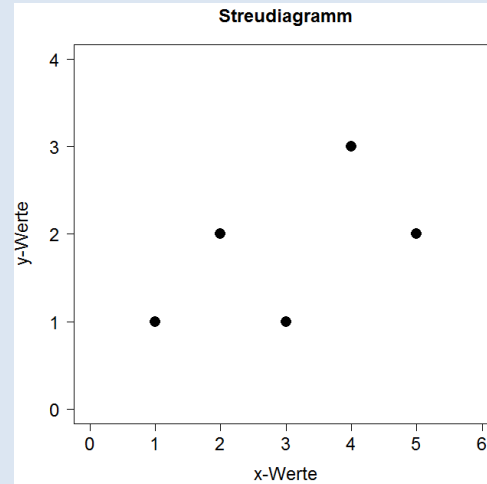
```
> segments(3,0,3,1.8,lty=2,lwd=1.5)
```

Linie von (3,0) bis (3,1.8)

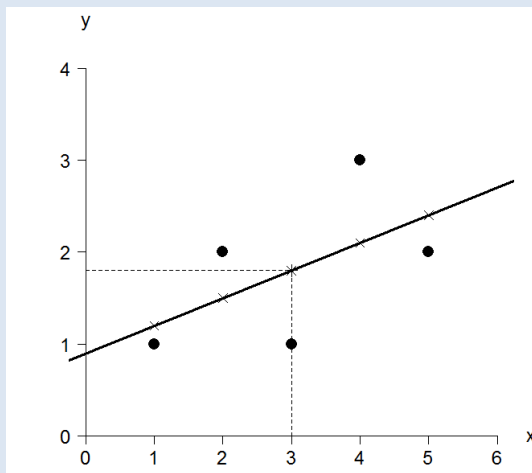
## R.6 Deskriptive Statistik mit R



1: ohne Modifikationen



2: modifiziert



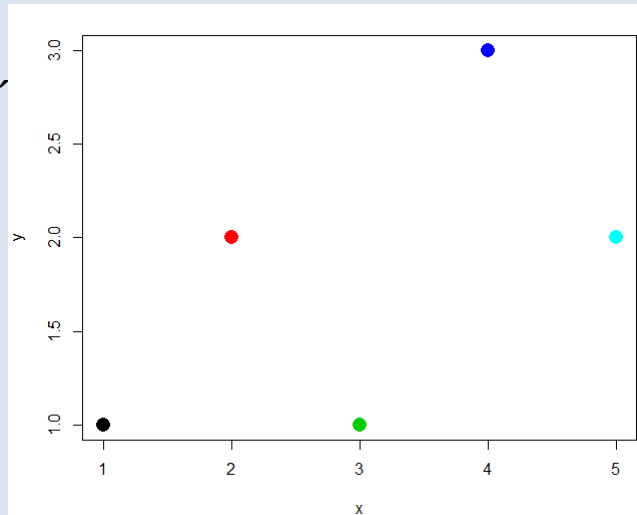
3: modifiziert und mit  
Low-Level-Zusätzen

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

# Farben

```
> x=c(1,2,3,4,5)
> y=c(1,2,1,3,2)
> plot(x,y,pch=16,cex=2,col=1:5)
```



Standardfarben



Recyclingprinzip bei Farben



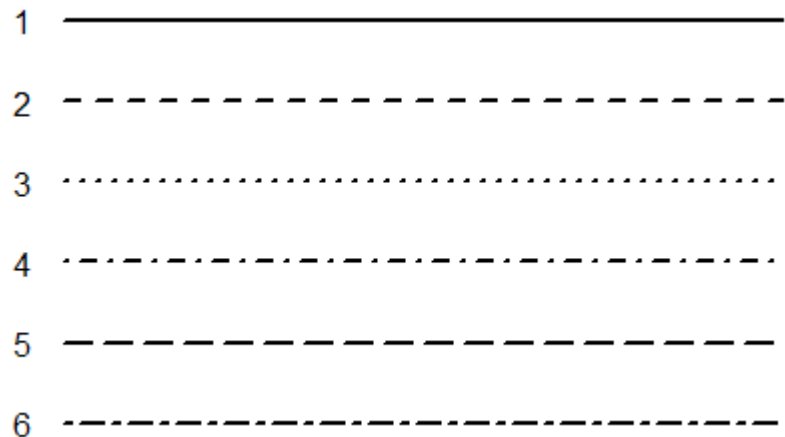
## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Linientypen
```

```
> plot(x,y,type="n",xlim=c(0,1),ylim=c(0,6),xlab="",  
+ ylab="",axes=FALSE)  
> abline(h=6:1,lty=1:6,lwd=2)  
h=6:1: horizontale Geraden auf den Höhen 6,5,...,1  
> text(-0.1,6:1,1:6,xpd=TRUE)
```

Standardlinientypen



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

### # Symbolarten

```
> plot(x,y,  
+ type="n",  
+ xlim=c(1,11),ylim=c(0,2.5),xlab="",ylab="",axes=FALSE)  
> points(1:11,rep(1.6,11),pch=0:10,cex=3)  
> points(1:10,rep(0.6,10),pch=11:20,cex=3)  
> text(1:11,2,0:10)  
> text(1:10,1,11:20)
```

### Symbolarten für Beobachtungswerte

|                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                       |                                                                                       |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 0                                                                                   | 1                                                                                   | 2                                                                                   | 3                                                                                   | 4                                                                                   | 5                                                                                   | 6                                                                                   | 7                                                                                   | 8                                                                                   | 9                                                                                     | 10                                                                                    |
|    |    |    |    |    |    |    |    |    |    |    |
| 11                                                                                  | 12                                                                                  | 13                                                                                  | 14                                                                                  | 15                                                                                  | 16                                                                                  | 17                                                                                  | 18                                                                                  | 19                                                                                  | 20                                                                                    |                                                                                       |
|  |  |  |  |  |  |  |  |  |  |  |

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Globale Grafikparameter
```

```
> par()
```

```
$xlog
```

```
[1] FALSE
```

```
$ylog
```

```
[1] FALSE
```

```
...      kein vollständiger Output
```

```
> names(par())  par() erzeugt Liste! Namen der Komponenten
```

|      |            |             |            |           |            |             |
|------|------------|-------------|------------|-----------|------------|-------------|
| [1]  | "xlog"     | "ylog"      | "adj"      | "ann"     | "ask"      | "bg"        |
| [7]  | "bty"      | "cex"       | "cex.axis" | "cex.lab" | "cex.main" | "cex.sub"   |
| [13] | "cin"      | "col"       | "col.axis" | "col.lab" | "col.main" | "col.sub"   |
| [19] | "cra"      | "crt"       | "csi"      | "cxy"     | "din"      | "err"       |
| [25] | "family"   | "fg"        | "fig"      | "fin"     | "font"     | "font.axis" |
| [31] | "font.lab" | "font.main" | "font.sub" | "lab"     | "las"      | "lend"      |
| [37] | "lheight"  | "ljoin"     | "lmitre"   | "lty"     | "lwd"      | "mai"       |
| [43] | "mar"      | "mex"       | "mfcoll"   | "mfg"     | "mfrow"    | "mgp"       |
| [49] | "mkh"      | "new"       | "oma"      | "omd"     | "omi"      | "page"      |
| [55] | "pch"      | "pin"       | "plt"      | "ps"      | "pty"      | "smo"       |
| [61] | "srt"      | "tck"       | "tcl"      | "usr"     | "xaxp"     | "xaxs"      |
| [67] | "xaxt"     | "xpd"       | "yaxp"     | "yaxs"    | "yaxt"     | "ylbias"    |

```
> ?par()      zur Bedeutung konsultiere man Hilfetexte
```

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Beispiel: Änderung von Default-Werten

```
> par()$las
```

```
[1] 0
```

Default für Achsenlabels

```
> par()$pch
```

```
[1] 1
```

Default für Symbolart

```
> plot(0,0)
```

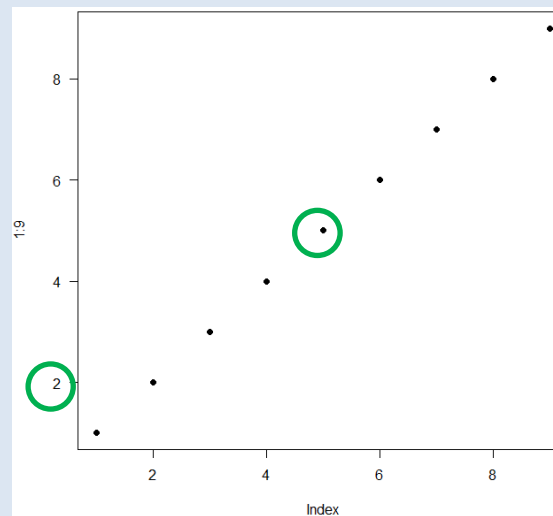
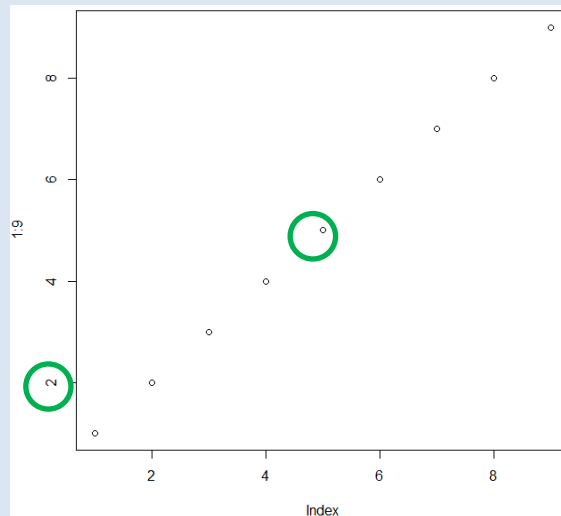
```
> par(las=1,pch=16)
```

Änderungen

```
> plot(0,0)
```

```
> par(las=0,pch=1)
```

Zurücksetzen auf Defaultwerte



## R.6 Deskriptive Statistik mit R

### # Kreisdiagramm

#### Unmodifiziert

```
> Anteile=c(41.5,25.7,4.8,8.6,8.4,10.9)
```

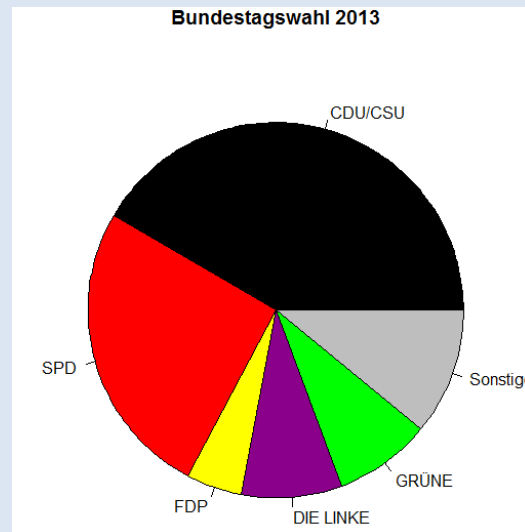
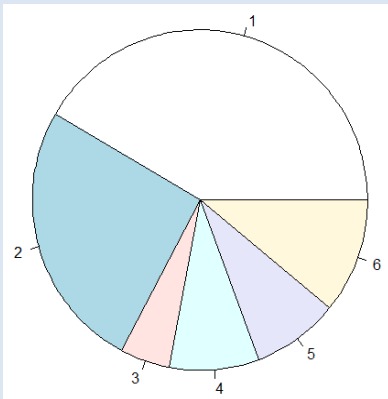
```
> pie(Anteile)
```

#### Modifiziert

```
> Parteien=c("CDU/CSU","SPD","FDP","DIE LINKE","GRÜNE","Sonstige")
```

```
> Buntfarben=c("black","red","yellow","darkmagenta","green","grey")
```

```
> pie(Anteile,labels=Parteien,col=Buntfarben,main="Bundestagswahl 2013")
```





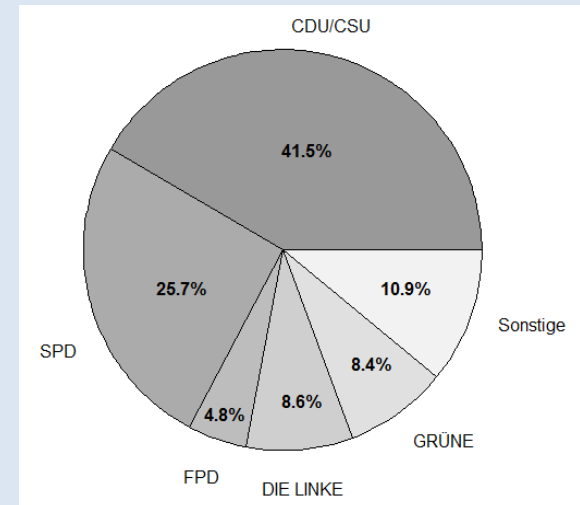
## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Modifiziert und mit Low-Level-Zusätzen

```
> Graufarben=gray(seq(0.6,0.95,le=6))
> pie(Anteile,labels="",col=Graufarben)
> text(0.1,0.4,"41.5%",font=2)
> text(0.2,0.9,"CDU/CSU")
> text(-0.4,-0.15,"25.7%",font=2)
> text(-0.9,-0.4,"SPD")
> text(-0.23,-0.65,"4.8%",font=2)
> text(-0.32,-0.9,"FPD")
> text(0.08,-0.6,"8.6%",font=2)
> text(0.085,-0.95,"DIE LINKE")
> text(0.36,-0.45,"8.4%",font=2)
> text(0.65,-0.75,"GRÜNE")
> text(0.5,-0.15,"10.9%",font=2)
> text(1,-0.3,"Sonstige")
```

font=2: **fett**



## R.6 Deskriptive Statistik mit R

Kreisdiagramm auf Basis von `table()` bzw. `prop.table()`

```
> x=c(0,0,0,0,0,1,1,2,2,2)
```

```
> nj=table(x)
```

```
> nj
```

```
x
```

```
0 1 2
```

```
5 2 3
```

```
> fj=prop.table(nj) angewendet auf table(x) (nicht auf x!)
```

```
> fj
```

```
x
```

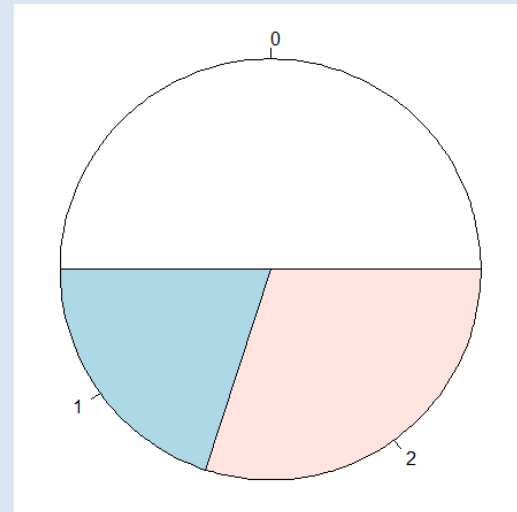
```
0 1 2
```

```
0.5 0.2 0.3
```

```
> pie(nj)
```

```
> pie(fj)
```

jeweils gleiches Resultat



## R.6 Deskriptive Statistik mit R

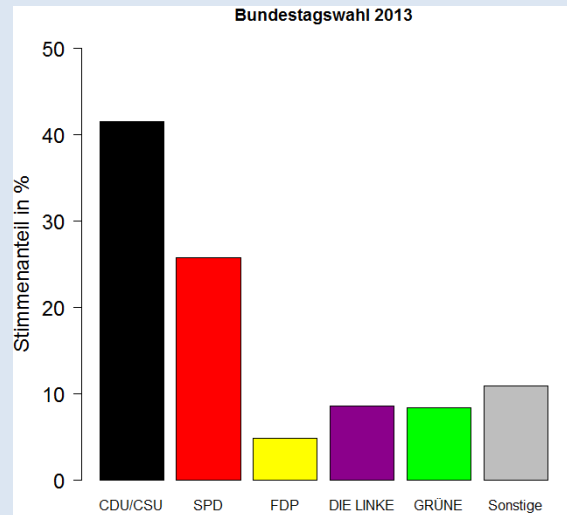
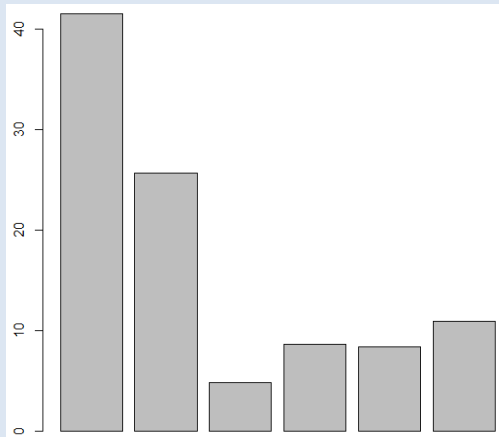
```
# Säulendiagramme (Daten von Folie 67)
```

```
Unmodifiziert
```

```
> barplot(Anteile)
```

```
Modifiziert
```

```
> barplot(Anteile, names.arg=Parteien,  
+ ylim=c(0,50), ylab="Stimmenanteil in %", las=1,  
+ cex.axis=1.5, cex.lab=1.5,  
+ col=Buntfarben, main="Bundestagswahl 2013")
```

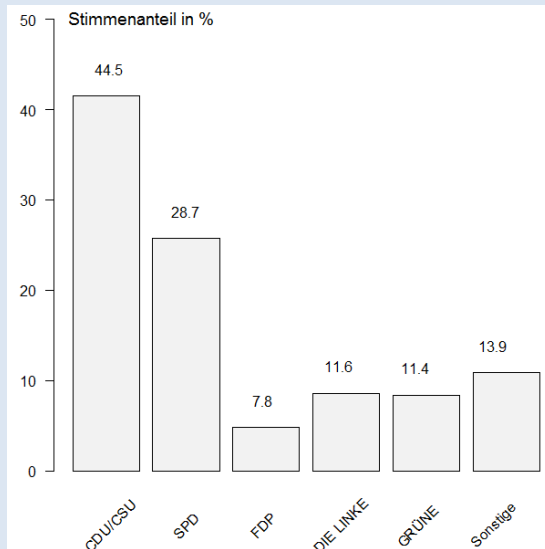


## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Modifiziert und mit Low-Level-Zusätzen

```
> barplot(Anteile, names.arg="", ylim=c(0, 50), ylab="",  
+ las=2, col=gray(0.95))  
> text(0, 50, "Stimmenanteil in %", xpd=TRUE, pos=4, cex=1.2)  
> text(seq(0.75, 6.5, by=1.15), -6, Parteien, srt=45, xpd=TRUE)  
> text(seq(0.75, 6.5, by=1.15), Anteile+3, Anteile+3)
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

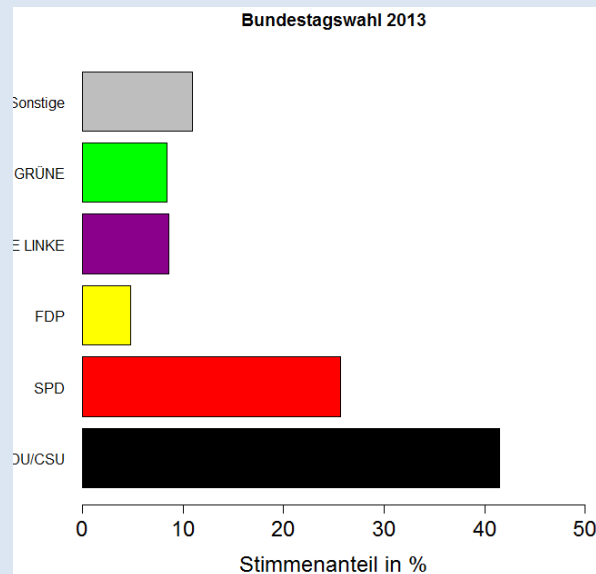
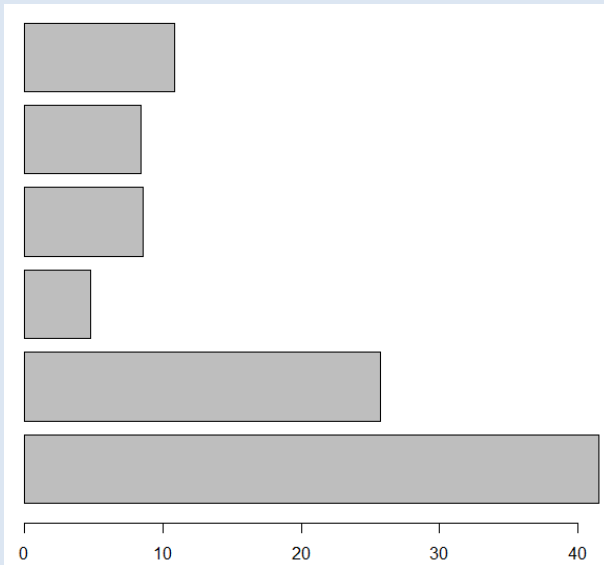
```
# Balkendiagramme (Daten von Folie 67)
```

Unmodifiziert

```
> barplot(Anteile,horiz=TRUE)
```

Modifiziert mit Beschriftung links

```
> barplot(Anteile,names.arg=Parteien,xlim=c(0,50),  
+ xlab="Stimmenanteil in %",cex.axis=1.5,cex.lab=1.5,  
+ las=1,col=Buntfarben,  
+ horiz=TRUE,main="Bundestagswahl 2013")
```

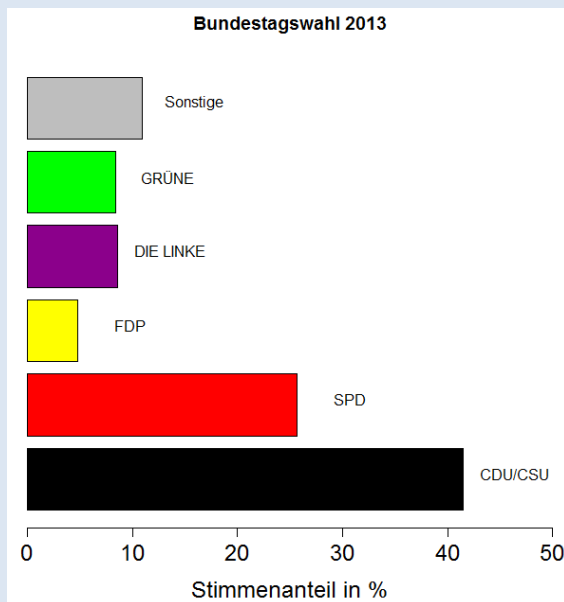


## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Modifiziert mit Low-Level-Beschriftung rechts

```
> barplot(Anteile, names.arg="", xlim=c(0, 50),  
+ xlab="Stimmenanteil in%", cex.axis=1.5, cex.lab=1.5,  
+ las=1, col=Buntfarben,  
+ horiz=TRUE, main="Bundestagswahl 2013")  
> text(Anteile+5, seq(0.8, 6.8, by=1.2), Parteien)
```

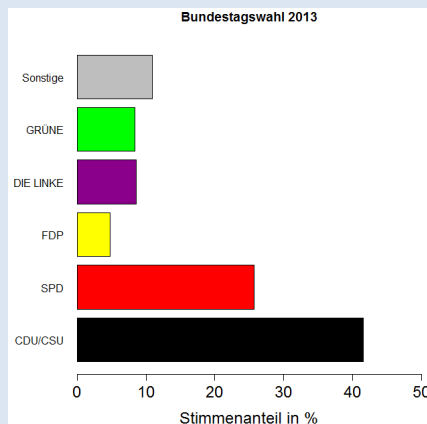


## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Modifiziert mit mehr Platz für Beschriftung links

```
> par()$mar
[1] 5.1 4.1 4.1 2.1      Default-Werte für Grafikränder
> marStandard=par()$mar  Abspeichern der Default-Werte
> par(mar=marStandard+c(0,2,0,0))  Vergrößerung des
> par()$mar                  Platzes am linken Rand
[1] 5.1 6.1 4.1 2.1
> barplot(Anteile,names.arg=Parteien,xlim=c(0,50),
+ xlab="Stimmenanteil in %",cex.axis=1.5,cex.lab=1.5,
+ las=1,col=Buntfarben,
+ horiz=TRUE,main="Bundestagswahl 2013")
> par(mar=marStandard)      Zurücksetzen auf alte Werte
```



## R.6 Deskriptive Statistik mit R

# Gestapelte Säulendiagramme

# (vgl. Folie 43)

Unmodifiziert, gestapelt über Y

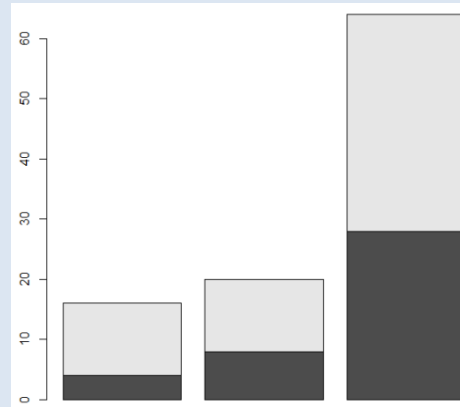
```
> m=c(4,12,8,12,28,36)
```

```
> m=matrix(m,2,3)
```

```
> m
```

```
      [,1] [,2] [,3]  
[1,]    4    8   28  
[2,]   12   12   36
```

```
> barplot(m)
```

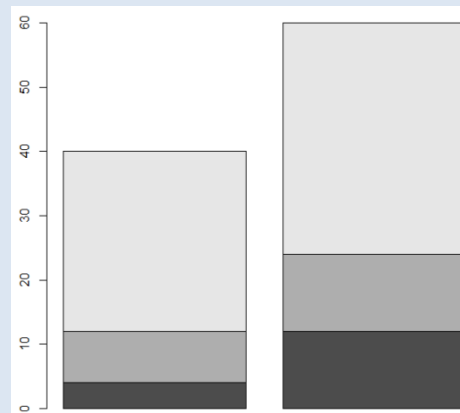


Unmodifiziert, gestapelt über X

```
> t(m)
```

```
      [,1] [,2]  
[1,]    4   12  
[2,]    8   12  
[3,]   28   36
```

```
> barplot(t(m))
```

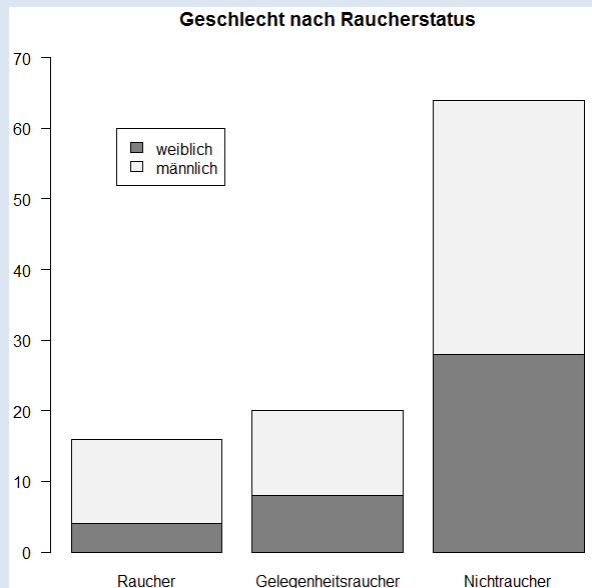




## R.6 Deskriptive Statistik mit R

Modifiziert mit Low-Level-Legende

```
> barplot(m,  
+ names.arg=c("Raucher","Gelegenheitsraucher","Nichtraucher"),  
+ ylim=c(0,70),las=1,col=c("grey50","grey95"),  
+ main="Geschlecht nach Raucherstatus")  
> legend(0.5,60,legend=c("weiblich","männlich"),  
+ fill=c("grey50","grey95"))
```



## R.6 Deskriptive Statistik mit R

# Gruppierte Säulendiagramme

Unmodifiziert, gruppiert nach Y

```
> barplot(m,beside=TRUE)
```

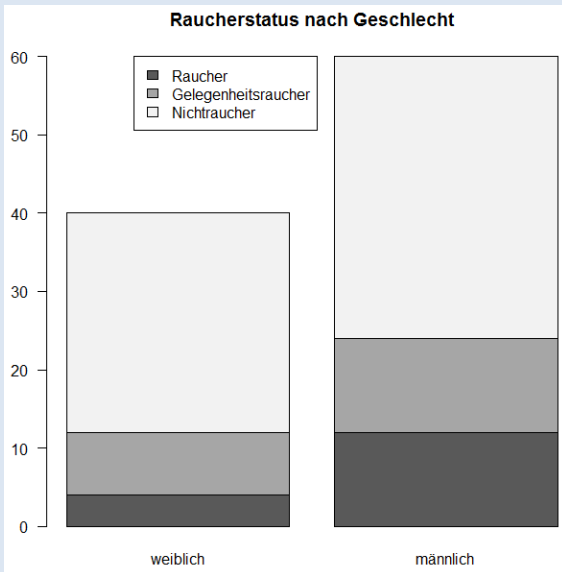
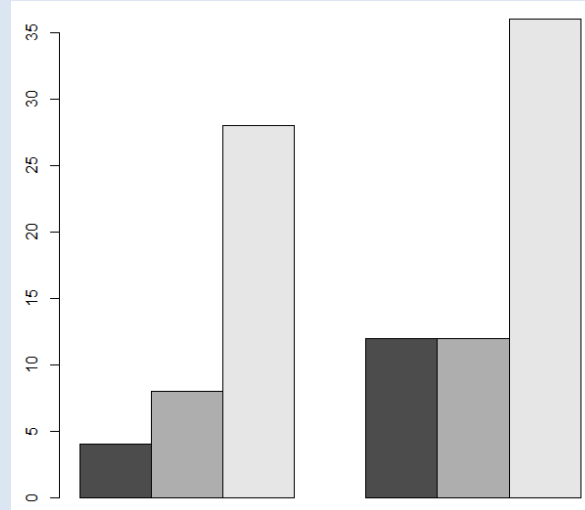
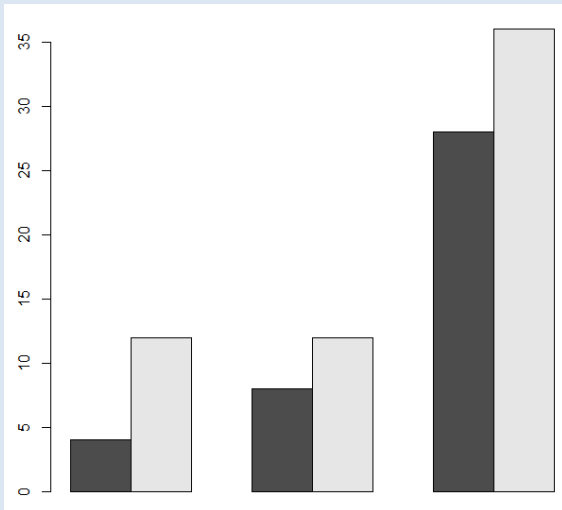
Unmodifiziert, gruppiert nach X

```
> barplot(t(m),beside=TRUE)
```

Modifiziert mit Low-Level-Legende

```
> barplot(t(m),  
+ names.arg=c("weiblich","männlich"),  
+ las=1,col=c("grey35","grey65","grey95"),  
+ main="Raucherstatus nach Geschlecht")  
> legend(0.5,60,  
+ legend=c("Raucher","Gelegenheitsraucher","Nichtraucher"),  
+ fill=c("grey35","grey65","grey95"))
```

## R.6 Deskriptive Statistik mit R



## R.6 Deskriptive Statistik mit R

```
# Segmentierte Säulen- und Balkendiagramme
```

```
# (vgl. Folie 43)
```

```
1: Unmodifiziert, bedingt auf Y (Säulen)
```

```
> m
```

```
      [,1] [,2] [,3]  
[1,]     4     8    28  
[2,]    12    12    36
```

```
> prop.table(m,2)
```

```
      [,1] [,2] [,3]  
[1,] 0.25  0.4 0.4375  
[2,] 0.75  0.6 0.5625
```

```
> 100*prop.table(m,2)
```

```
      [,1] [,2] [,3]  
[1,]    25    40 43.75  
[2,]    75    60 56.25
```

```
> barplot(100*prop.table(m,2))
```

```
2: Unmodifiziert, bedingt auf Y (Balken)
```

```
> barplot(100*prop.table(m,2),horiz=TRUE)
```

## R.6 Deskriptive Statistik mit R

3: Unmodifiziert, bedingt auf X (Säulen)

```
> prop.table(m,1)
```

```
      [,1] [,2] [,3]  
[1,]  0.1  0.2  0.7  
[2,]  0.2  0.2  0.6
```

```
> t(prop.table(m,1))
```

```
      [,1] [,2]  
[1,]  0.1  0.2  
[2,]  0.2  0.2  
[3,]  0.7  0.6
```

```
> 100*t(prop.table(m,1))
```

```
      [,1] [,2]  
[1,]   10   20  
[2,]   20   20  
[3,]   70   60
```

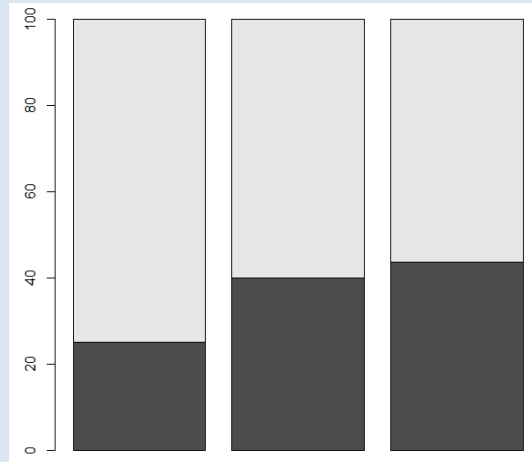
```
> barplot(100*t(prop.table(m,1)))
```

4: Unmodifiziert, bedingt auf X (Balken)

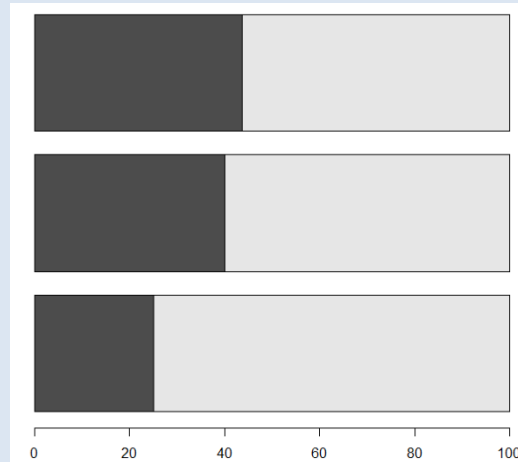
```
> barplot(100*t(prop.table(m,1)),horiz=TRUE)
```

## R.6 Deskriptive Statistik mit R

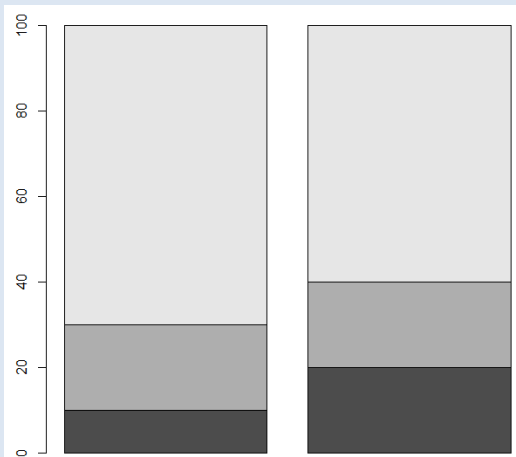
1: bedingt auf Y



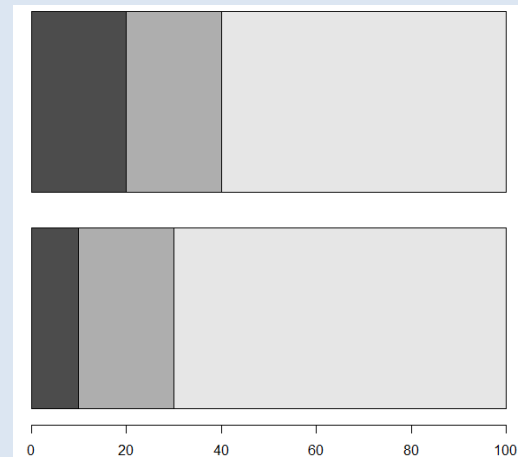
2: bedingt auf Y



3: bedingt auf X



4: bedingt auf X



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Assoziationsplot
```

```
Unmodifiziert
```

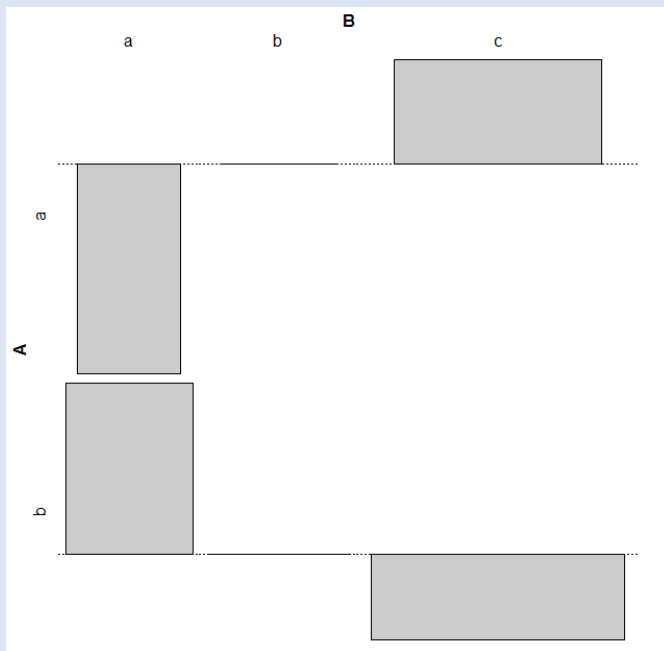
```
> m
```

```
      [,1] [,2] [,3]  
[1,]     4     8    28  
[2,]    12    12    36
```

```
> library(vcd)
```

Laden des Paketes „vcd“

```
> assoc(m)
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

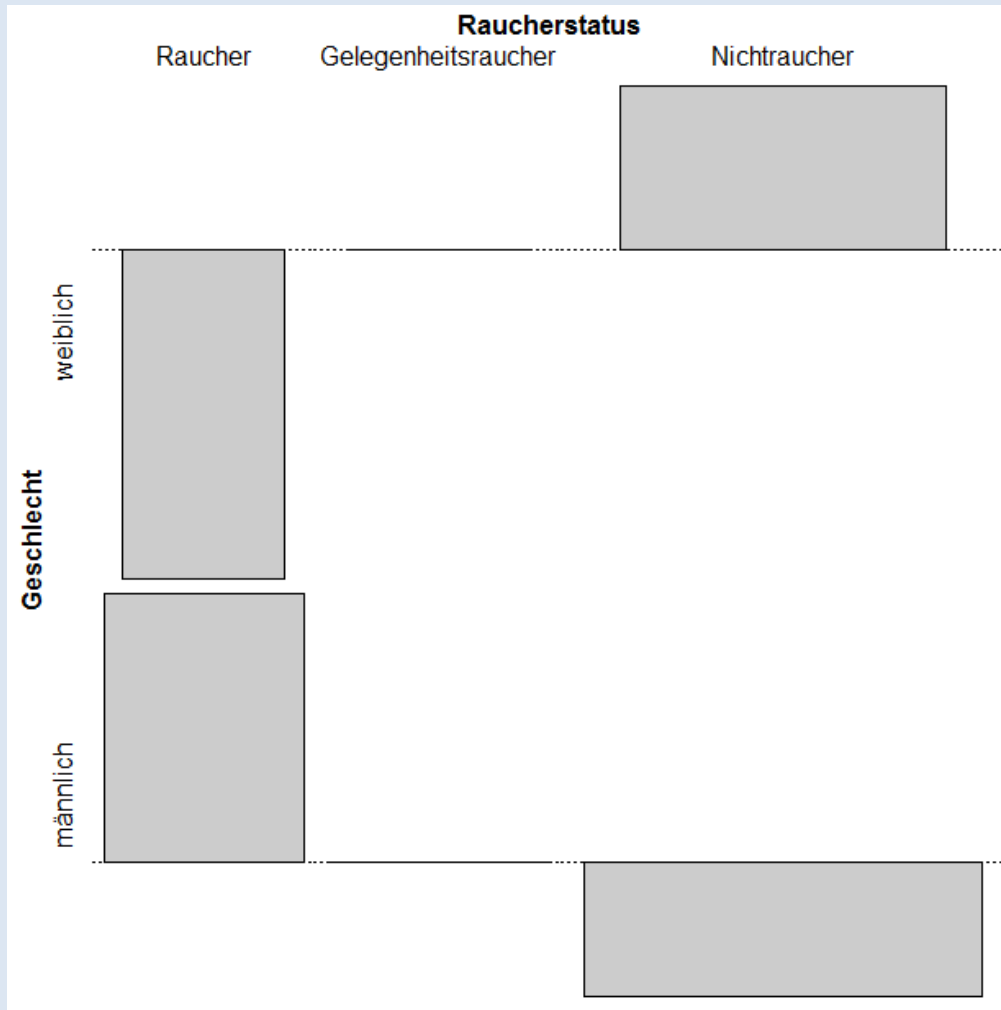
Modifiziert (mit sinnvollen Beschriftungen)

```
> dim(m)
[1] 2 3
> dimnames(m)
NULL
> dimnames(m)=list(
+ Geschlecht=c("weiblich","männlich"),
+ Raucherstatus=c("Raucher","Gelegenheitsraucher","Nichtraucher"))
> dimnames(m)
$Geschlecht
[1] "weiblich" "männlich"
$Raucherstatus
[1] "Raucher"          "Gelegenheitsraucher"
   "Nichtraucher"
> assoc(m)
```



# R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Mosaikplot
```

```
Unmodifiziert
```

```
> library(vcd)
```

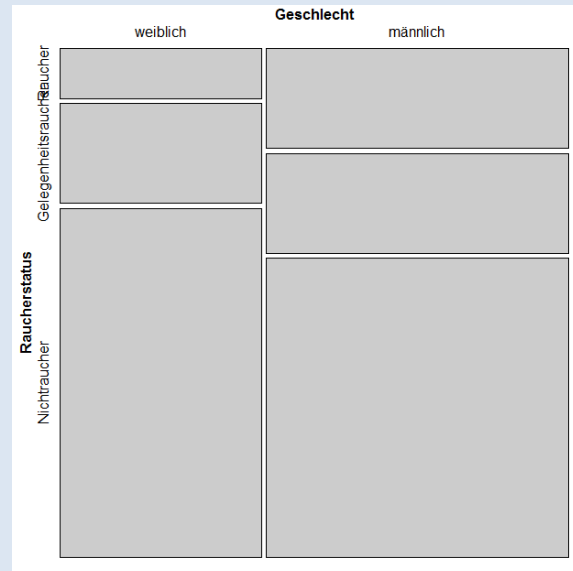
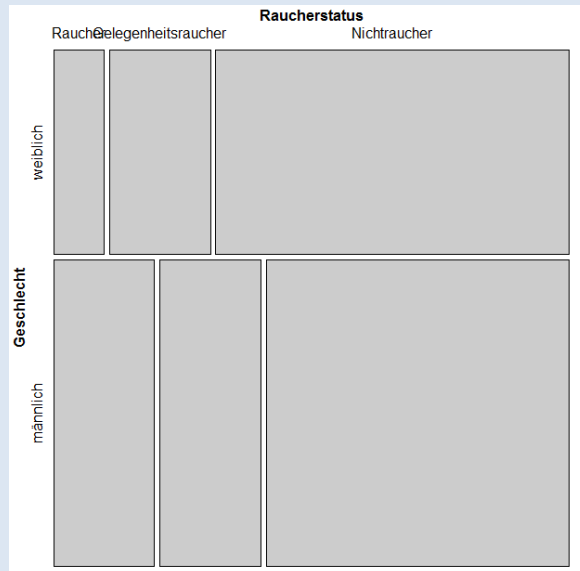
```
> mosaic(m)
```

```
> mosaic(m,direction="v")
```

Laden des Paketes „vcd“

Balkenvariante

Säulenvariante



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Spineplot
```

```
> dim(Sternzeichen)
```

Datensatz vorher einlesen

```
[1] 206    2
```

```
> names(Sternzeichen)
```

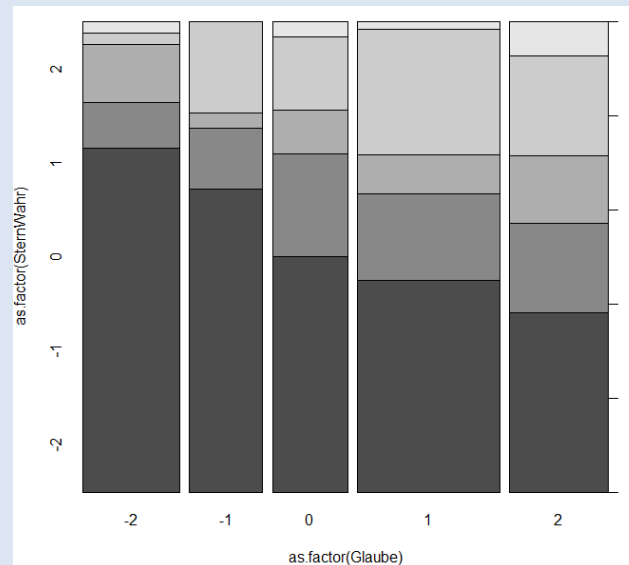
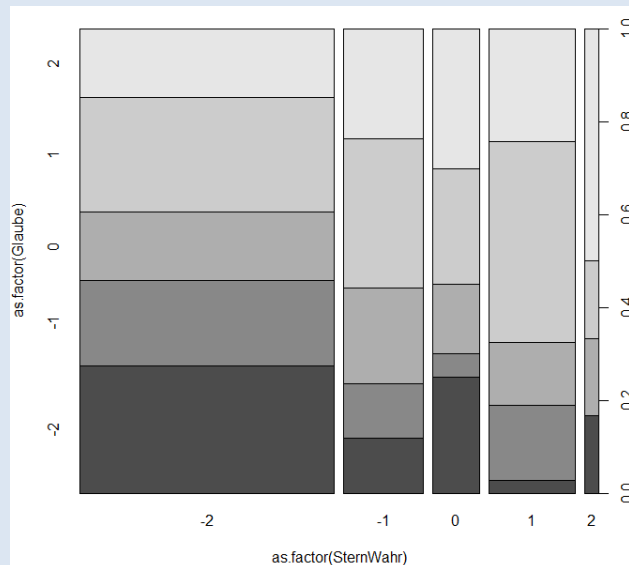
```
[1] "Glaube"    "SternWahr,,
```

```
> attach(Sternzeichen)
```

Unmodifiziert

```
> plot(as.factor(Glaube) ~ as.factor(SternWahr)) Variante 1
```

```
> plot(as.factor(SternWahr) ~ as.factor(Glaube)) Variante 2
```



# R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Stamm-Blatt-Diagramm
```

```
> set.seed(1)      Zufallsgenerator auf Anfangswert setzen
```

```
> z=runif(10)      10 Zufallszahlen aus (0,1) ziehen
```

```
> z
```

```
[1] 0.26550866 0.37212390 0.57285336 0.90820779
```

```
[5] 0.20168193 0.89838968 0.94467527 0.66079779
```

```
[9] 0.62911404 0.06178627
```

```
> stem(z)
```

The decimal point is 1 digit(s) to the left of the |

```
0 | 6          ist eine 0.06
```

```
2 | 077
```

```
4 | 7
```

```
6 | 36
```

```
8 | 014
```

```
> dim(Deutschland) Datensatz vorher einlesen
```

```
[1] 396 4
```

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
> names(Deutschland)      Datensatz vorher einlesen
[1] "Bevölkerungsdichte" "Wanderungssaldo"
[3] "Insolvenzen"         "Arbeitslosigkeit"
> stem(Deutschland$Arbeitslosigkeit)
```

The decimal point is at the |

```
1 | 4
2 | 111112233445666777777888888888899999999
3 | 000011111222222222233333345555555666666777778888889999
4 | 0000000011111111122223333333444445555566677777888999999999
5 | 000000000111123334444445555556667778888888999999
6 | 0000001111222222333344444555556666677888999
7 | 0000011122233344455556666777888899999
8 | 00001112222244555666
9 | 001222233444445566677899
10 | 001222333344456666677779
11 | 00001255668
12 | 00111224457889
13 | 0011223677
14 | 12358
15 | 1
16 | 37
```

# R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

## # Histogramm

### Unmodifiziert

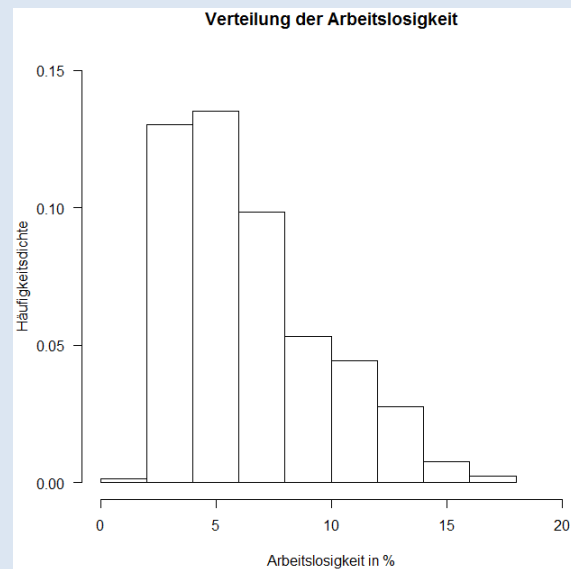
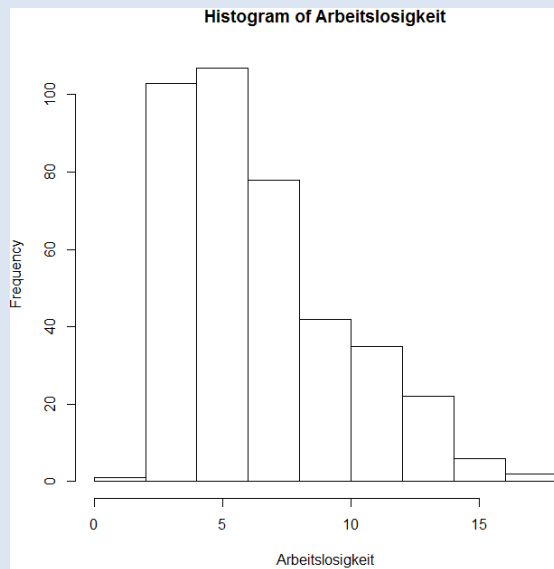
```
> attach(Deutschland)
```

(Datensatz vorher einlesen)

```
> hist(Arbeitslosigkeit)
```

### Modifiziert

```
> hist(Arbeitslosigkeit, prob=TRUE,      prob=TRUE: Dichtewert  
+ xlab="Arbeitslosigkeit in %", ylab="Häufigkeitsdichte",  
+ xlim=c(0,20), ylim=c(0,0.15), las=1,  
+ main="Verteilung der Arbeitslosigkeit")
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

`hist()` erzeugt eine Liste

```
> reshist=hist(Arbeitslosigkeit,plot=FALSE)
```

```
> names(reshist)
```

```
[1] "breaks"    "counts"    "density"   "mids"      "xname"     "equidist"
```

```
> attach(reshist)
```

```
> breaks
```

```
[1]  0  2  4  6  8 10 12 14 16 18
```

```
> counts
```

```
[1]  1 103 107  78  42  35  22  6  2
```

```
> density
```

```
[1] 0.001262626 0.130050505 0.135101010 0.098484848
```

```
[5] 0.053030303 0.044191919 0.027777778 0.007575758
```

```
[9] 0.002525253
```

```
> mids
```

```
[1]  1  3  5  7  9 11 13 15 17
```

# R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

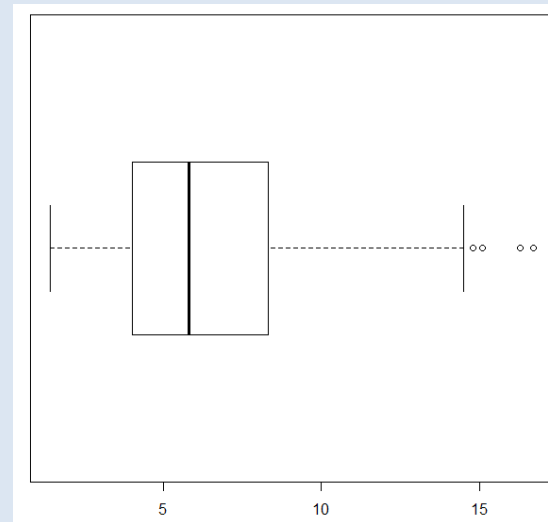
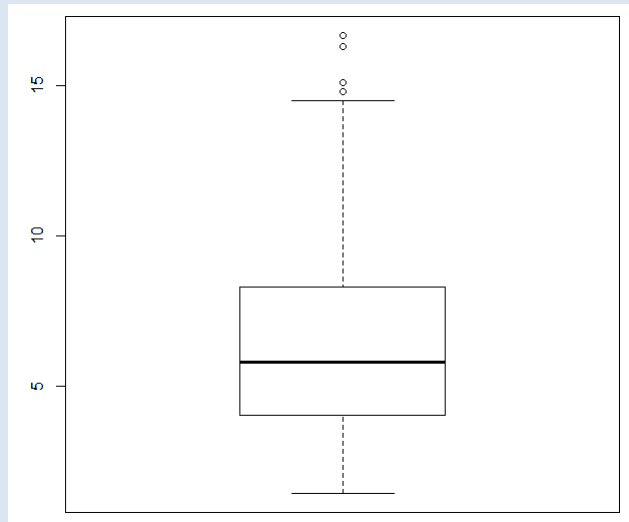
## # Boxplot

Unmodifiziert (einzeln)

> attach(Deutschland) (Datensatz vorher einlesen)

> boxplot(Arbeitslosigkeit) Variante 1

> boxplot(Arbeitslosigkeit, horizontal=TRUE) Variante 2





## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Unmodifiziert (mehrere)

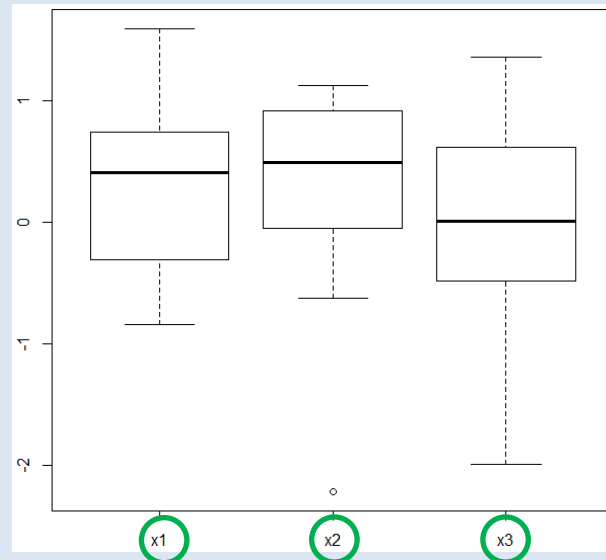
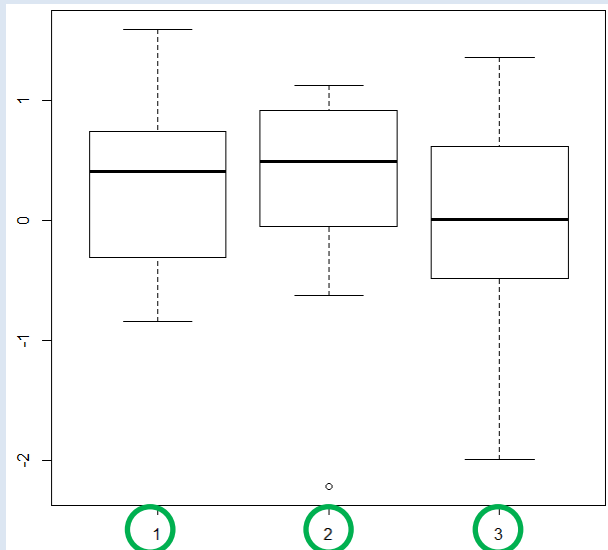
```
> set.seed(1)           Zufallsgenerator auf Anfangswert setzen  
> x1=rnorm(10)          10 Zufallszahlen  
> x2=rnorm(10)          ...  
> x3=rnorm(10)          ...
```

entweder so

```
> boxplot(x1,x2,x3)
```

oder so

```
> boxplot(data.frame(x1,x2,x3))
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Modifiziert, metrisches vs. kategoriales Merkmal

```
> dim(Studenten2)           Datensatz vorher einlesen
```

```
[1] 214    3
```

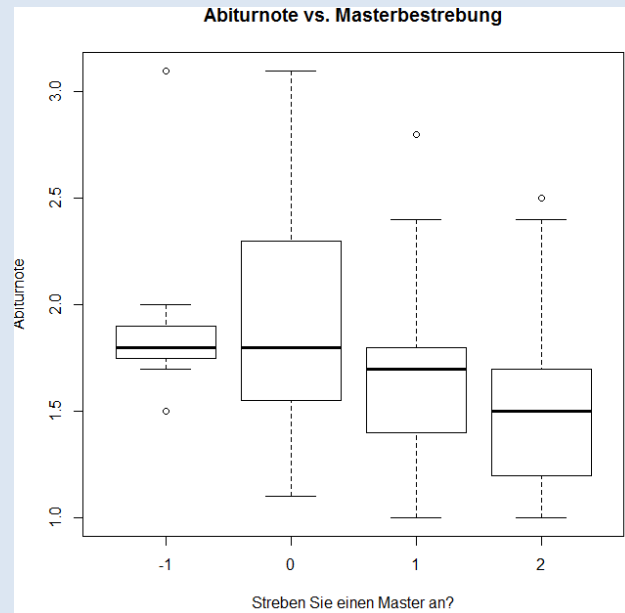
```
> names(Studenten2)
```

```
[1] "Abiturnote" "Alter"      "Master"
```

```
> attach(Studenten2)
```

Modellvariante mit  $y \sim x$

```
> boxplot(Abiturnote~Master, xlab="Streben Sie einen Master an?",  
+ main="Abiturnote vs. Masterbestrebung")
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

### # Spinogramm

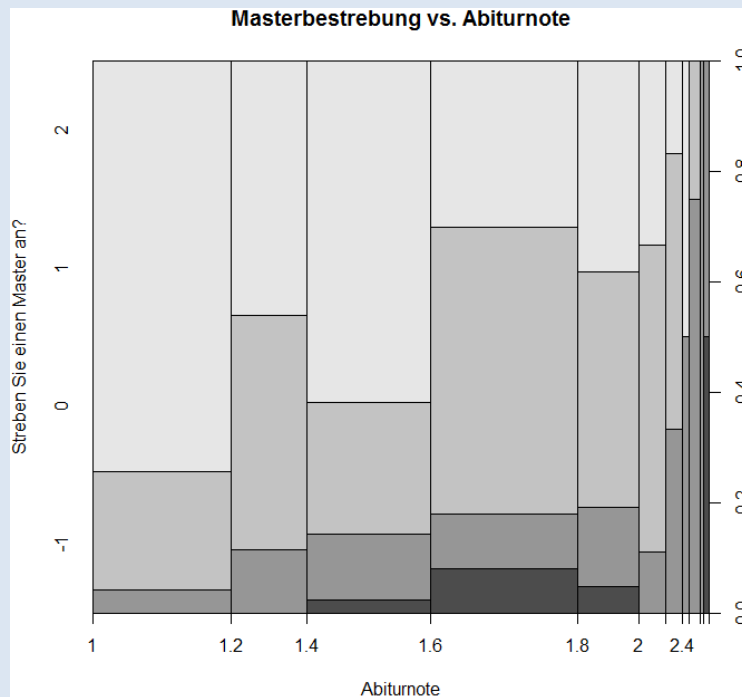
Modifiziert, kategoriales vs. metrisches Merkmal

```
> attach(Studenten2) (Datensatz vorher einlesen)
```

```
> plot(as.factor(Master)~Abiturnote,
```

```
+ ylab="Streben Sie einen Master an?",
```

```
+ main="Masterbestrebung vs. Abiturnote")
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
# Streudiagramme  
# Allgemeines: Vgl. Folie 63-66  
# Jetzt: Einige Spezialfälle
```

Beispiel 1: Symbolart gemäß kategorialer Variable

```
> dim(Studenten1) Datensatz vorher einlesen
```

```
[1] 201 5
```

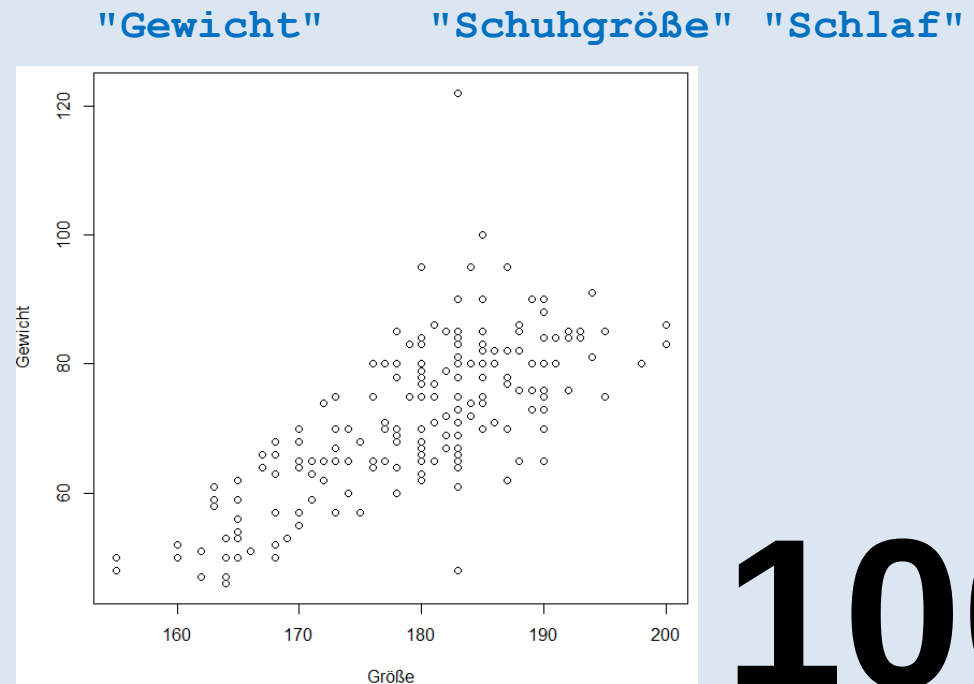
```
> names(Studenten1)
```

```
[1] "Geschlecht" "Größe"
```

```
> attach(Studenten1)
```

Unmodifiziert

```
> plot(Größe, Gewicht)
```



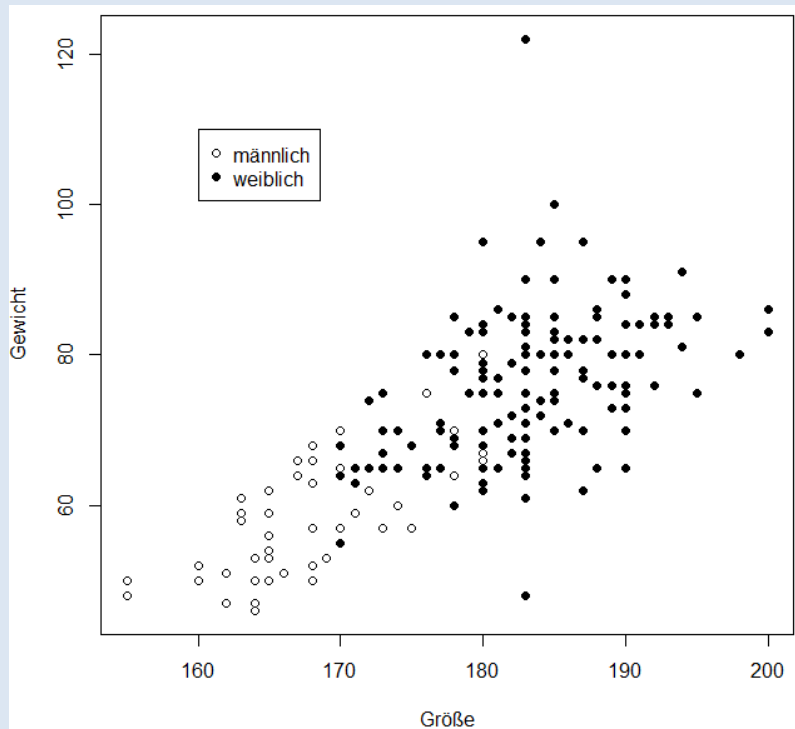
100

## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

```
> ifelse (Geschlecht==0,16,1)
[1] 16 16  1 16  1 16  1 16 16 16 16 16 16 16 16
[16] 16 ... kein vollständiger Output

> pchtyp=ifelse (Geschlecht==0,16,1)      Korrektur! (Ergänzung)
> plot (Größe, Gewicht, pch=pchtyp)
> legend (160,110,legend=c ("männlich", "weiblich"), pch=c (1,16))
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

Beispiel 2: Symbolart gemäß kategorialer Variable

```
> dim(Wohnen)           Datensatz vorher einlesen
```

```
[1] 32  3
```

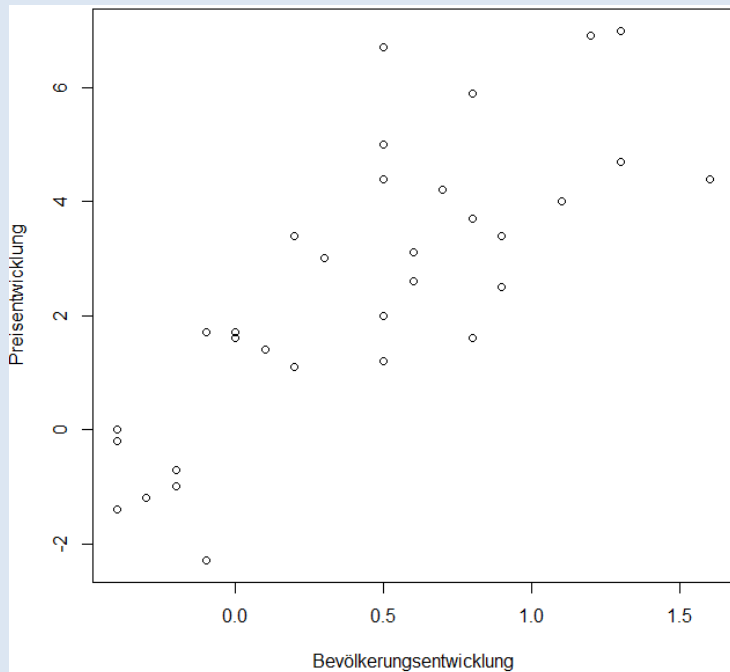
```
> names(Wohnen)
```

```
[1] "Stadt"    "Preisentwicklung"    "Bevölkerungsentwicklung"
```

```
> attach(Wohnen)
```

Unmodifiziert

```
> plot(Bevölkerungsentwicklung, Preisentwicklung)
```



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

> Stadt

```
[1] Hamburg      Hannover      Bremen       Düsseldorf
```

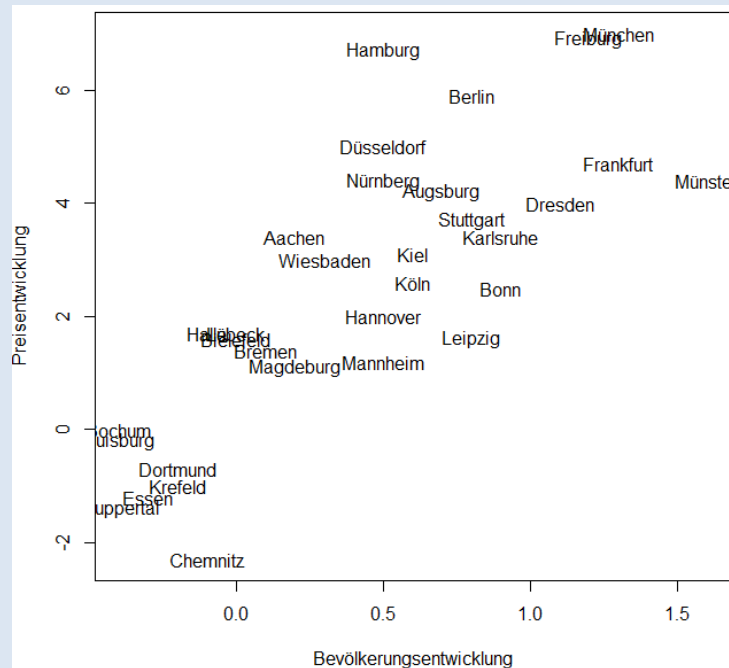
```
[5] Duisburg      ... kein vollständiger Output
```

> plot(Bevölkerungsentwicklung, Preisentwicklung, type="n")

type="n": Unterdrückung eines Plotsymbols (keine Punkte)

> text(Bevölkerungsentwicklung, Preisentwicklung, Stadt)

Einfügung der Städtenamen an der Stelle der Punkte



## R.6 Deskriptive Statistik mit R

Nicht prüfungsrelevant

### Beispiel 3: Streudiagramm-Matrix

```
> names(Deutschland)      Datensatz vorher einlesen
```

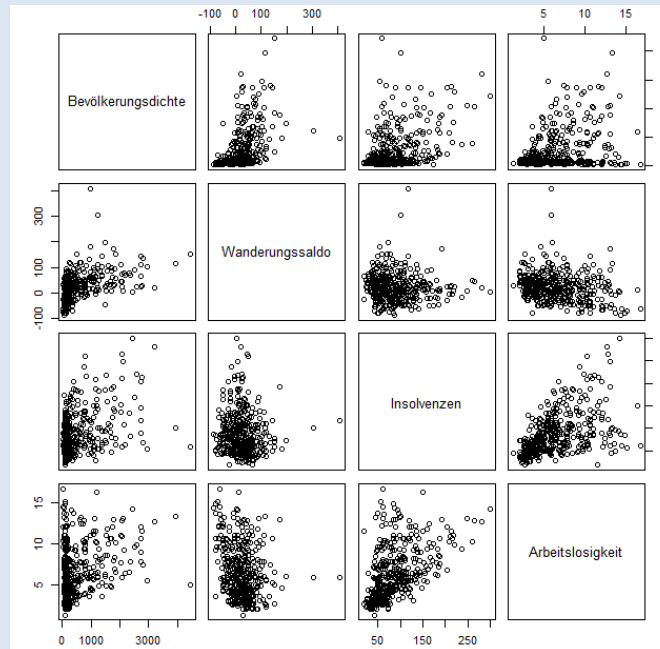
```
[1] "Bevölkerungsdichte" "Wanderungssaldo"
```

```
[3] "Insolvenzen"         "Arbeitslosigkeit"
```

```
> round(cor(Deutschland), 2)
```

|                    | Bevölkerungsdichte | Wanderungssaldo | Insolvenzen | Arbeitslosigkeit |
|--------------------|--------------------|-----------------|-------------|------------------|
| Bevölkerungsdichte | 1.00               | 0.46            | 0.42        | 0.33             |
| Wanderungssaldo    | 0.46               | 1.00            | -0.06       | -0.21            |
| Insolvenzen        | 0.42               | -0.06           | 1.00        | 0.50             |
| Arbeitslosigkeit   | 0.33               | -0.21           | 0.50        | 1.00             |

```
> pairs(Deutschland)
```





## R.7 Wahrscheinlichkeitsverteilungen mit R

##### Binomialverteilung #####

Abb. 7.3.3:  $B(10, \pi)$ -Verteilung für unterschiedliche Werte von  $\pi$

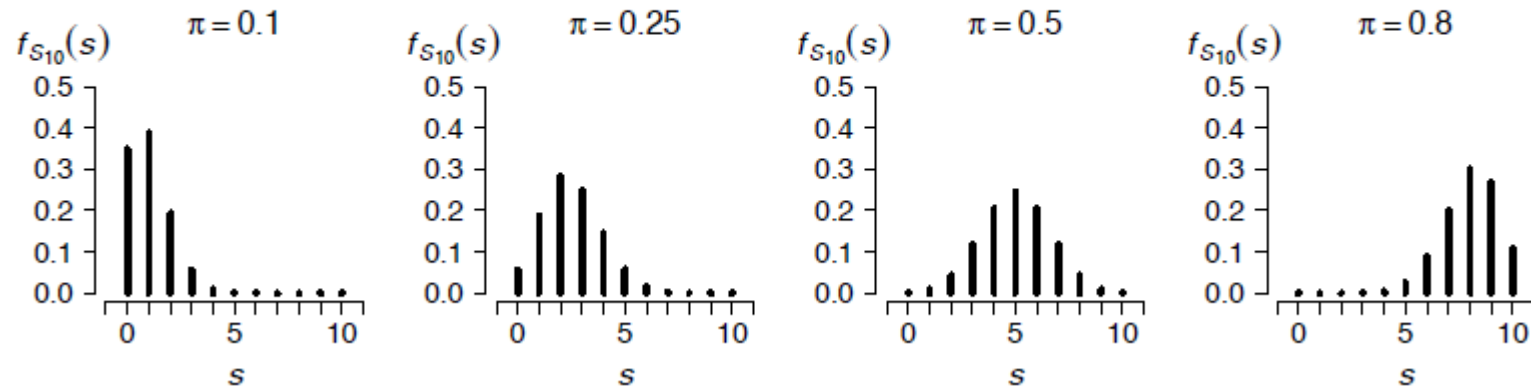
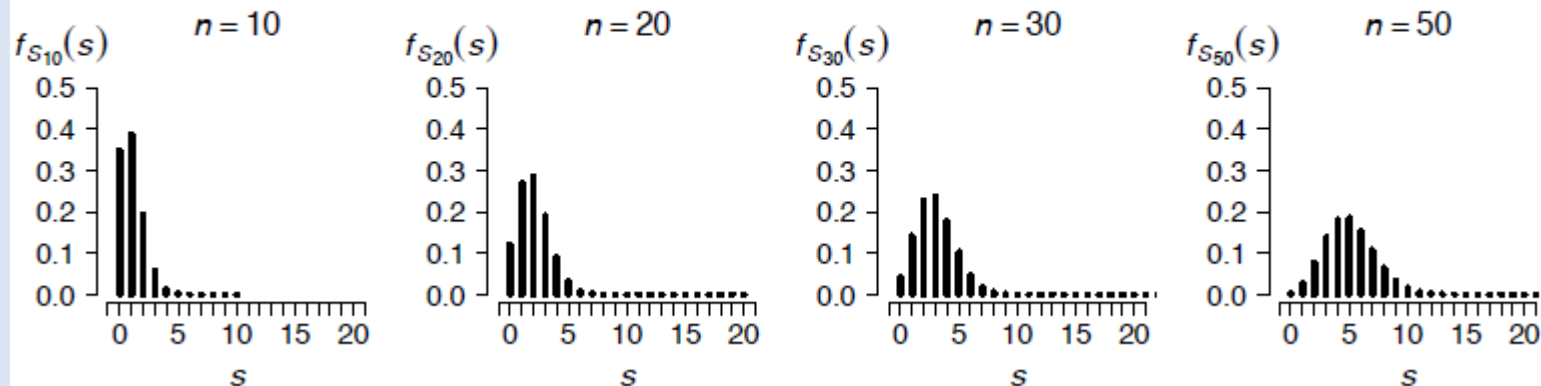


Abb. 7.3.4:  $B(n, 0.1)$ -Verteilung für unterschiedliche Werte von  $n$



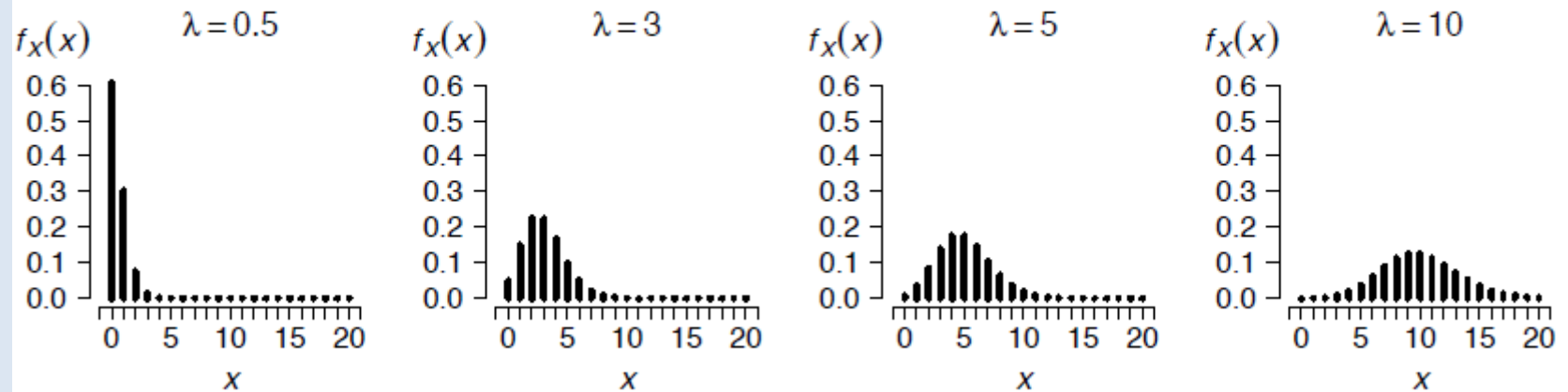
## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: B(10,0.1) und B(10,0.25)
> dbinom(0,10,0.1)
[1] 0.3486784
> dbinom(1,10,0.1)
[1] 0.3874205
> dbinom(2,10,0.1)
[1] 0.1937102
> pbinom(2,10,0.1)
[1] 0.9298092
> x=0:10
> dbinom(x,10,0.1)
[1] 0.3486784401 0.3874204890 0.1937102445 0.0573956280
[5] 0.0111602610 0.0014880348 0.0001377810 0.0000087480
[9] 0.0000003645 0.0000000090 0.0000000001
> round(dbinom(x,10,0.1),2)
[1] 0.35 0.39 0.19 0.06 0.01 0.00
[7] 0.00 0.00 0.00 0.00 0.00
> y=dbinom(x,10,0.1)
> plot(x,y,type="h")
> plot(x,dbinom(x,10,0.25),type="h")
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

##### Poisson-Verteilung #####

*Abb. 7.3.5:  $Po(\lambda)$ -Verteilung für unterschiedliche Werte von  $\lambda$*



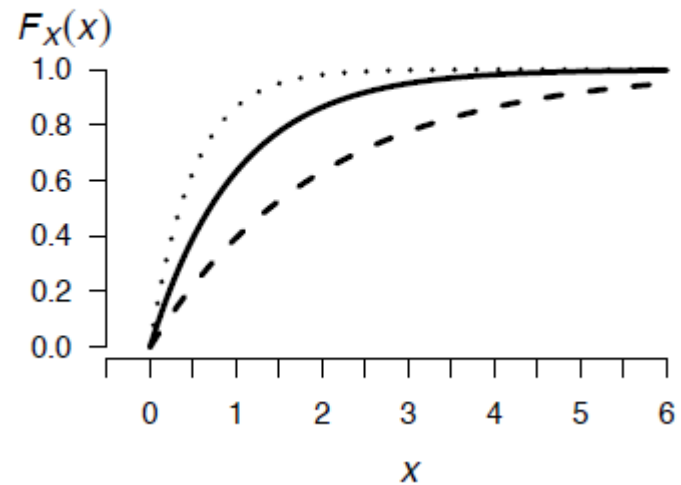
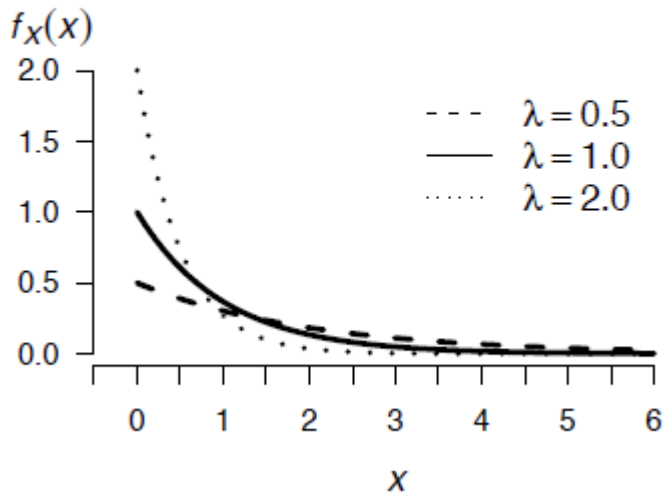
## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: Po(10) und Po(20)
> dpois(0,10)
[1] 4.539993e-05
> dpois(1,10)
[1] 0.0004539993
> dpois(2,10)
[1] 0.002269996
> ppois(2,10)
[1] 0.002769396
> x=0:20
> round(dpois(x,20),2)
[1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00
[7] 0.00 0.00 0.00 0.00 0.01 0.01
[13] 0.02 0.03 0.04 0.05 0.06 0.08
[19] 0.08 0.09 0.09
> y=dpois(x,10)
> plot(x,y,type="h")
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

##### Exponentialverteilung #####

*Abb. 7.3.9:  $\text{Exp}(\lambda)$ -Verteilung für unterschiedliche Werte von  $\lambda$*



## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: Exp(0.5) und Exp(1)
> dexp(0,0.5)
[1] 0.5
> dexp(1,0.5)
[1] 0.3032653
> pexp(1,0.5)
[1] 0.3934693
> dexp(0,1)
[1] 1
> dexp(1,1)
[1] 0.3678794
> pexp(1,1)
[1] 0.6321206
> xgitter=seq(0,6,le=100)
> y=dexp(xgitter,1)
> plot(xgitter,y,type="l",ylim=c(0,2))
> lines(xgitter,dexp(xgitter,0.5),lty=2)
> plot(xgitter,pexp(xgitter,1),type="l")
> lines(xgitter,pexp(xgitter,0.5),lty=2)
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

```
> qexp(0.5, 0.5)
[1] 1.386294
> qexp(0.05, 0.5)
[1] 0.1025866
> qexp(0.95, 0.5)
[1] 5.991465
> qexp(0.5, 1)
[1] 0.6931472
> qexp(0.05, 1)
[1] 0.05129329
> qexp(0.95, 1)
[1] 2.995732
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

##### Normalverteilung #####

Abb. 7.3.10:  $N(\mu, 1)$ -Verteilung für unterschiedliche Werte von  $\mu$

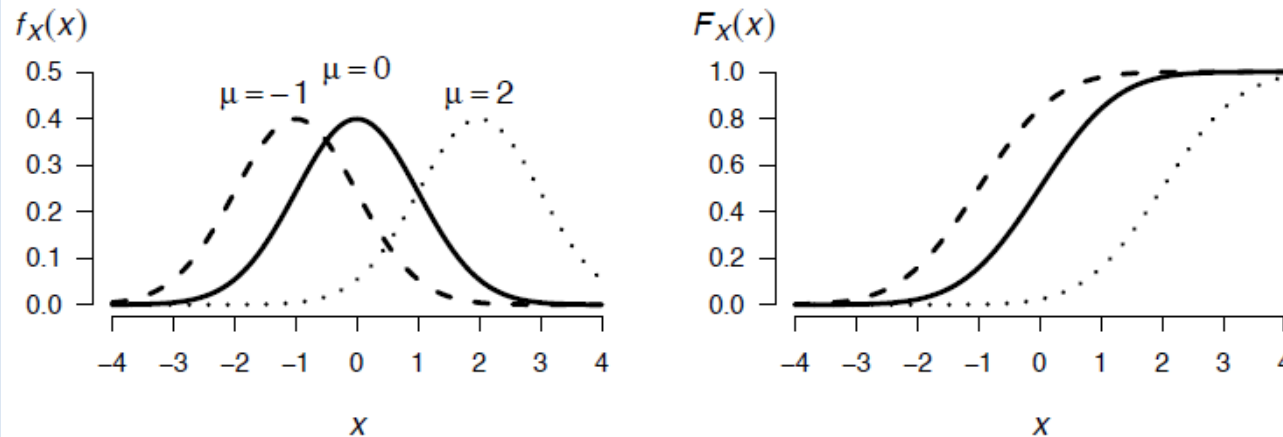
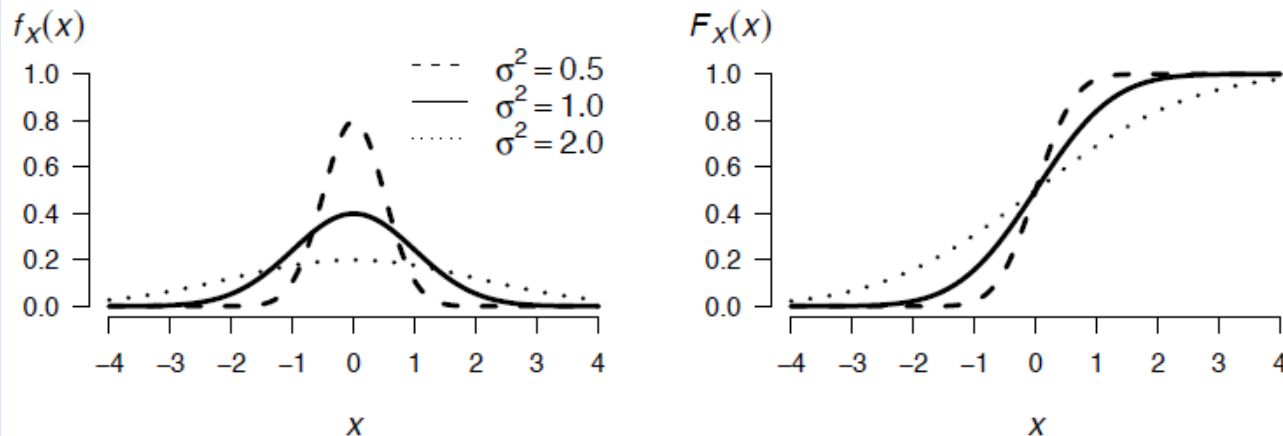


Abb. 7.3.11:  $N(0, \sigma^2)$ -Verteilung für unterschiedliche Werte von  $\sigma^2$





## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: N(0,1) und N(0,sqrt(0.5))
> dnorm(0,0,1)
[1] 0.3989423
> dnorm(1,0,1)
[1] 0.2419707
> pnorm(1,0,1)
[1] 0.8413447
> pnorm(1.64,0,1)
[1] 0.9494974
> dnorm(0,0,sqrt(0.5))
[1] 0.5641896
> pnorm(1,0,sqrt(0.5))
[1] 0.9213504
> xgitter=seq(-4,4,le=100)
> y=dnorm(xgitter,0,1)
> plot(xgitter,y,type="l",ylim=c(0,1))
> lines(xgitter,dnorm(xgitter,0,sqrt(0.5)),lty=2)
> plot(xgitter,pnorm(xgitter,0,1),type="l")
> lines(xgitter,pnorm(xgitter,0,sqrt(0.5)),lty=2)
```

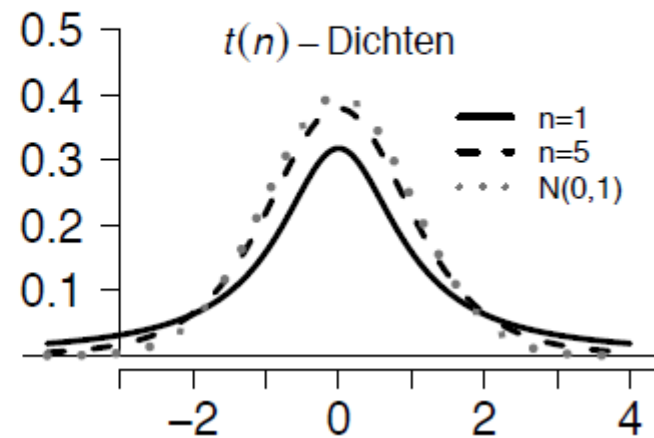
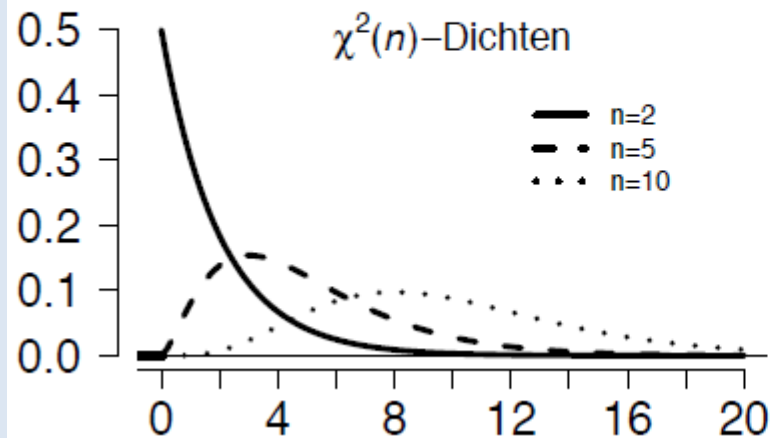
## R.7 Wahrscheinlichkeitsverteilungen mit R

```
> qnorm(0.5,0,1)
[1] 0
> qnorm(0.05,0,1)
[1] -1.644854
> qnorm(0.95,0,1)
[1] 1.644854
> qnorm(0.975,0,1)
[1] 1.959964
> qnorm(0.5,0,sqrt(0.5))
[1] 0
> qnorm(0.95,0,sqrt(0.5))
[1] 1.163087
> qnorm(0.05,0,sqrt(0.5))
[1] -1.163087
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

##### t-Verteilung und Chiquadrat-Verteilung #####

Abb. 8.1.1: Dichten der  $\chi^2$ -Verteilung und  $t$ -Verteilung



## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: t(1), t(5) und N(0,1)
```

```
> dt(0,1)
```

```
[1] 0.3183099
```

```
> dt(0,5)
```

```
[1] 0.3796067
```

```
> dnorm(0,0,1)
```

```
[1] 0.3989423
```

```
> dt(2.5,1)
```

```
[1] 0.04390481
```

```
> dt(2.5,5)
```

```
[1] 0.03332624
```

```
> dnorm(2.5,0,1)
```

```
[1] 0.0175283
```

```
> pt(2,1)
```

```
[1] 0.8524164
```

```
> pt(2,5)
```

```
[1] 0.9490303
```

```
> pnorm(2,0,1)
```

```
[1] 0.9772499
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

```
> qt(0.975,1)
[1] 12.7062
> qt(0.975,5)
[1] 2.570582
> qnorm(0.975,0,1)
[1] 1.959964

> xgitter=seq(-4,4,le=100)
> plot(xgitter,dt(xgitter,1),type="l",ylim=c(0,0.5))
> lines(xgitter,dt(xgitter,5),lty=2)
> lines(xgitter,dnorm(xgitter,0,1),lty=3)
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

```
# Beispiel: Chi(2)-, Chi(5)- und Chi(10)-Verteilung
> dchisq(4,2)
[1] 0.06766764
> dchisq(4,5)
[1] 0.1439759
> dchisq(4,10)
[1] 0.04511176
> pchisq(4,2)
[1] 0.8646647
> pchisq(4,5)
[1] 0.450584
> pchisq(4,10)
[1] 0.05265302
> qchisq(0.95,2)
[1] 5.991465
> qchisq(0.95,5)
[1] 11.0705
> qchisq(0.95,10)
[1] 18.30704
```

## R.7 Wahrscheinlichkeitsverteilungen mit R

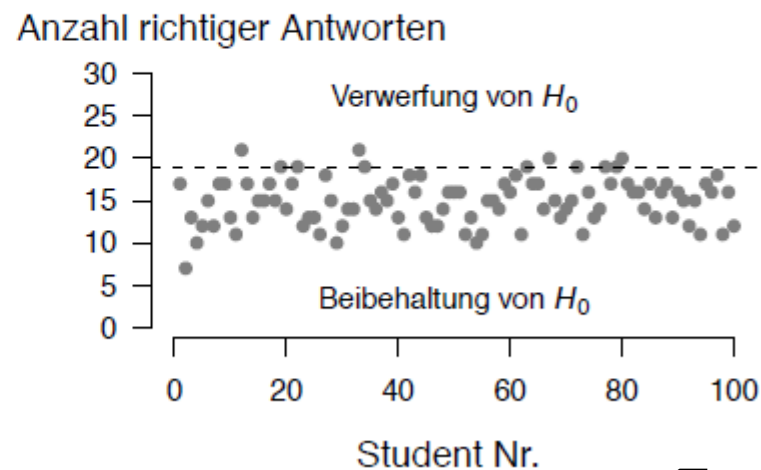
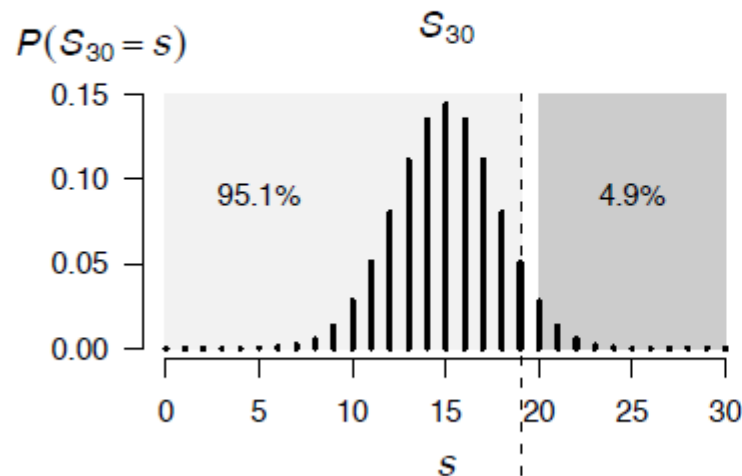
```
> xgitter=seq(0,20,le=100)
> plot(xgitter,dchisq(xgitter,2),type="l",ylim=c(0,0.5))
> lines(xgitter,dchisq(xgitter,5),lty=2)
> lines(xgitter,dchisq(xgitter,20),lty=3)
```

## R.8 Testen mit R

### ##### Exakter Binomialtest #####

- **Beispiel 11.1.1: Raten vs. Wissen** • Angenommen, eine Multiple-Choice-Klausur besteht aus 30 Aussagen, die entweder richtig oder falsch sind. Falls nun ein Student 19 dieser Fragen korrekt und 11 Fragen falsch beantwortet, wie ist dann eine solche Leistung einzustufen?

Abb. 11.1.1: Einseitiger oberer Binomialtest – Testverteilung und Simulation





## R.8 Testen mit R

```
> round(dbinom(x,30,0.5),2)
```

```
[1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.01  
[11] 0.03 0.05 0.08 0.11 0.14 0.14 0.14 0.11 0.08 0.05  
[21] 0.03 0.01 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
[31] 0.00
```

```
> round(pbinom(x,30,0.5),2)
```

```
[1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.02  
[11] 0.05 0.10 0.18 0.29 0.43 0.57 0.71 0.82 0.90 0.95  
[21] 0.98 0.99 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  
[31] 1.00
```

```
> pWert=1-pbinom(18,30,0.5)
```

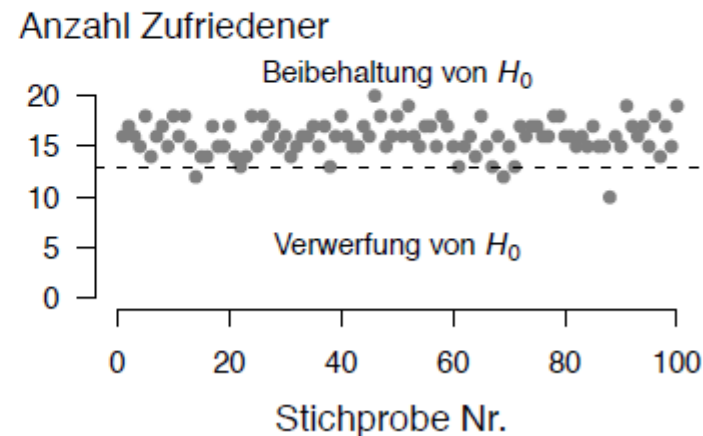
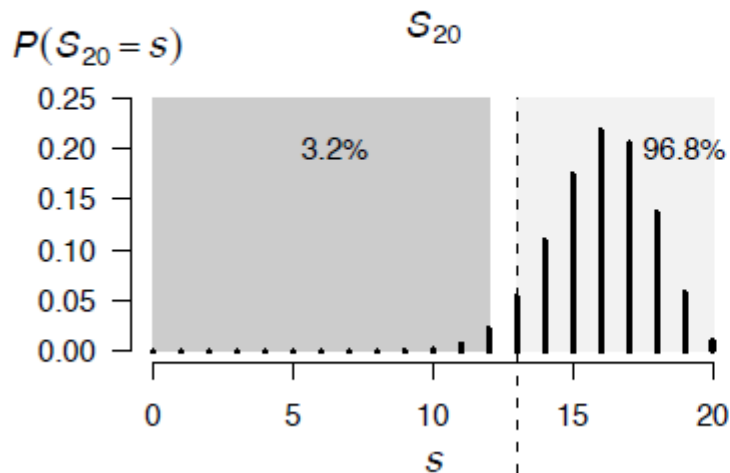
```
> pWert
```

```
[1] 0.1002442
```

## R.8 Testen mit R

• **Beispiel 11.1.2: Kein Rückgang vs. Rückgang** • Gegeben sei eine ähnliche Situation wie in Beispiel 1.2.1. Studierende werden zu ihrer Zufriedenheit mit der Mensa befragt. Angenommen, die Zufriedenheitsquote lag in einem vorhergehenden Jahr bei 80%. In einer Blitzumfrage äußerten sich aktuell nun 15 von 20 Befragten zufrieden. Dies entspricht einem aktuellen Anteil von nur noch 75%. Ist damit die Zufriedenheit unter allen Studierenden in der Grundgesamtheit nun tatsächlich zurückgegangen?

Abb. 11.1.2: Einseitiger unterer Binomialtest – Testverteilung und Simulation



```
> pWert=pbinom(15,20,0.8)
> pWert
[1] 0.3703517
```

## R.8 Testen mit R

### ##### Tests über Erwartungswerte #####

• **Beispiel 11.1.4: Keine Wirkung vs. Wirkung** • Gegeben sei das Pflanzenexperiment aus Beispiel 9.1.3 (fortgesetzt mit Beispiel 10.2.3). Wie lässt sich hier statistisch prüfen und entscheiden, ob ein gemessener Unterschied zwischen den beiden Behandlungsgruppen signifikant oder möglicherweise nur Zufall ist? Ab wann kann man von einer systematischen Wirkung der Düngung ausgehen?

Gruppe 0 (ohne Düngung): 14, 25, 36,  
Gruppe 1 (mit Düngung): 64, 46, 55.

$$Z_D = \frac{\bar{Y}_1 - \bar{Y}_0 - \delta_0}{\sqrt{\sigma_0^2/n_0 + \sigma_1^2/n_1}} \sim N(0, 1) \quad \text{für } \mu_1 - \mu_0 = \delta_0$$

$$\tilde{T}_D = \frac{\bar{Y}_1 - \bar{Y}_0 - \delta_0}{\sqrt{S_p^2/n_0 + S_p^2/n_1}} \sim t(n_0 + n_1 - 2) \quad \text{für } \mu_1 - \mu_0 = \delta_0$$

$$S_p^2 = \frac{1}{n_0 + n_1 - 2} (n_0 \tilde{S}_0^2 + n_1 \tilde{S}_1^2)$$

$$\hat{T}_D = \frac{\bar{Y}_1 - \bar{Y}_0 - \delta_0}{\sqrt{\tilde{S}_0^2/n_0 + \tilde{S}_1^2/n_1}} \stackrel{a}{\sim} N(0, 1) \quad \text{für } \mu_1 - \mu_0 = \delta_0$$

# 123

## R.8 Testen mit R

```
# Beispiel: t-Test für Erwartungswertdifferenz
> y=c(14,25,36,64,46,55)
> x=c(0,0,0,1,1,1)
> n0=3
> n1=3
> y0quer=mean(y[x==0])
> y1quer=mean(y[x==1])
> s02=var(y[x==0])
> s12=var(y[x==1])
> sp2=1/(n0+n1-2)*((n0-1)*s02+(n1-1)*s12)
> TDschlange=(y1quer-y0quer)/sqrt(sp2/n0+sp2/n1)
> TDschlange
[1] 3.656
> kritWert=qt(0.975,4)
> kritWert
[1] 2.776445
> pWert=2*(1-pt(TDschlange,n0+n1-2))
> pWert
[1] 0.02165653
```

## R.8 Testen mit R

##### Nichtparametrische Chiquadrat-Tests #####  
 # Beispiel 1: Anpassungstest

*Tabelle 10.2.1: Umfrageergebnis und Wahlergebnis (in %) der Bundestagswahl 2013*

| Partei                  | Union         | SPD           | FDP          | Linke        | Grüne        | AfD          | Sonst.       |
|-------------------------|---------------|---------------|--------------|--------------|--------------|--------------|--------------|
| Umfrage und Genauigkeit | 40.0<br>±2.6% | 27.0<br>±2.4% | 5.5<br>±1.2% | 8.5<br>±1.5% | 9.0<br>±1.5% | 4.0<br>±1.0% | 6.0<br>±1.3% |
| Endergebnis             | 41.5          | 25.7          | 4.8          | 8.6          | 8.4          | 4.7          | 6.3          |

*Daten: Der Bundeswahlleiter [2016], Forschungsgruppe Wahlen [2016]*

*Tabelle 11.3.1: Beobachtete vs. erwartete Häufigkeiten in Beispiel 11.3.3*

| Ausprägung $i$      | 1     | 2     | 3     | 4     | 5     | 6     | 7     |
|---------------------|-------|-------|-------|-------|-------|-------|-------|
| $f_i$               | 0.400 | 0.270 | 0.055 | 0.085 | 0.09  | 0.040 | 0.060 |
| $n_i = n \cdot f_i$ | 548   | 370   | 75    | 116   | 123   | 55    | 82    |
| $\pi_i$             | 0.415 | 0.257 | 0.048 | 0.086 | 0.084 | 0.047 | 0.063 |
| $n\pi_i$            | 568.1 | 351.8 | 65.7  | 117.7 | 115.0 | 64.3  | 86.2  |

$$\chi_A^2 = \sum_{i=1}^n \frac{(N_i - n\pi_i)^2}{n\pi_i} \stackrel{a}{\sim} \chi^2(k-1)$$

## R.8 Testen mit R

```
> ni=c(548,370,75,116,123,55,82)
> pi=c(0.415,0.257,0.048,0.086,0.084,0.047,0.063)
> chisq.test(x=ni,p=pi)
```

Chi-squared test for given probabilities

```
data:  ni
X-squared = 5.1129, df = 6, p-value = 0.5294
```

```
> res=chisq.test(x=ni,p=pi)
> names(res)
[1] "statistic" "parameter" "p.value"
[4] "method"    "data.name"  "observed"
[7] "expected"  "residuals" "stdres"
> attach(res)
> statistic
X-squared
 5.112886
> p.value
[1] 0.5294184
```

## R.8 Testen mit R

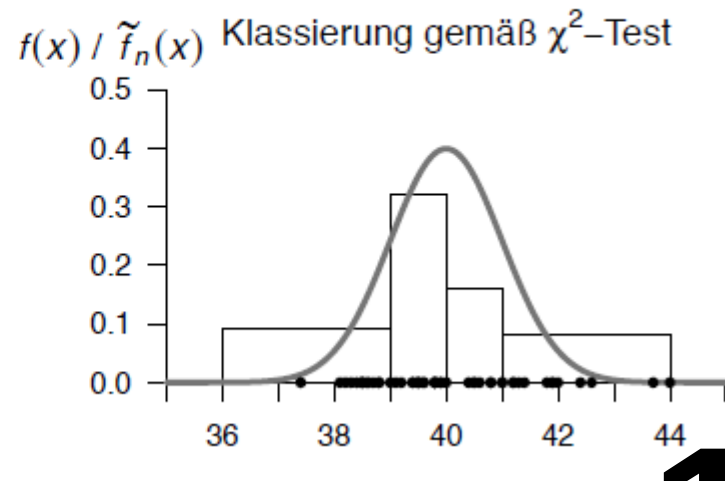
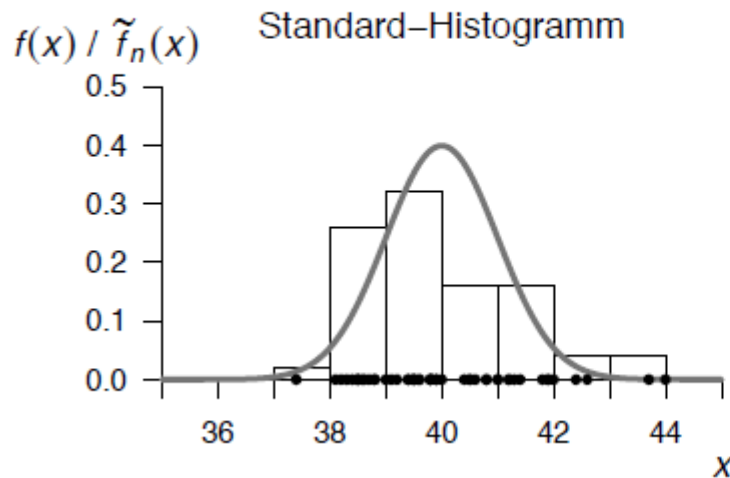
```
> observed
[1] 548 370 75 116 123 55 82
> expected
[1] 568.135 351.833 65.712 117.734
[5] 114.996 64.343 86.247
> kritWert=qchisq(0.95,6)
> kritWert
[1] 12.59159
> pWert=1-pchisq(statistic,6)
> pWert
X-squared
0.5294184
```

## R.8 Testen mit R

### # Beispiel 2: Anpassungstest (Test auf Verteilung)

37.4, 38.1, 38.2, 38.3, 38.4, 38.5, 38.5, 38.5, 38.6, 38.6,  
38.7, 38.8, 38.8, 39.0, 39.1, 39.2, 39.4, 39.4, 39.5, 39.5,  
39.5, 39.6, 39.6, 39.8, 39.8, 39.8, 39.8, 39.9, 39.9, 40.0,  
40.4, 40.5, 40.5, 40.6, 40.8, 40.8, 41.0, 41.0, 41.2, 41.2,  
41.3, 41.4, 41.8, 41.9, 41.9, 42.0, 42.4, 42.6, 43.7, 44.0.

Abb. 11.3.1: Empirische vs. hypothetische Verteilung in Beispiel 11.3.4





## R.8 Testen mit R

$$Y \sim N(40, 1)$$

$$\pi_1 = P(Y \in [36, 39]) \approx 0.16,$$

$$\pi_2 = P(Y \in (39, 40]) \approx 0.34,$$

$$\pi_3 = P(Y \in (40, 41]) \approx 0.34,$$

$$\pi_4 = P(Y \in (41, 44]) \approx 0.16,$$

$$X = \begin{cases} 1, & \text{für } Y \in [36, 39], \\ 2, & \text{für } Y \in (39, 40], \\ 3, & \text{für } Y \in (40, 41], \\ 4, & \text{für } Y \in (41, 44]. \end{cases}$$

Tabelle 11.3.2: Beobachtete vs. erwartete Häufigkeiten in Beispiel 11.3.4 (gerundet)

| Ausprägung i        | 1     | 2     | 3     | 4     |
|---------------------|-------|-------|-------|-------|
| $f_i$               | 0.280 | 0.320 | 0.160 | 0.240 |
| $n_i = n \cdot f_i$ | 14    | 16    | 8     | 12    |
| $\pi_i$             | 0.16  | 0.34  | 0.34  | 0.16  |
| $n\pi_i$            | 8     | 17    | 17    | 8     |

## R.8 Testen mit R

```
> x=c(37.4,38.1,38.2,38.3,38.4,38.5,38.5,38.5,38.6,38.6,  
+     38.7,38.8,38.8,39.0,39.1,39.2,39.4,39.4,39.5,39.5,  
+     39.5,39.6,39.6,39.8,39.8,39.8,39.8,39.9,39.9,40.0,  
+     40.4,40.5,40.5,40.6,40.8,40.8,41.0,41.0,41.2,41.2,  
+     41.3,41.4,41.8,41.9,42.0,42.0,42.4,42.6,43.7,44.0)  
> hist(x,breaks=c(36,39,40,41,44),prob=TRUE,ylim=c(0,0.5))  
> xgitter=seq(37,44,le=100)  
> y=dnorm(xgitter,40,1)  
> lines(xgitter,y)  
> sum(x>=36&x<=39)  
[1] 14  
> sum(x>39&x<=40)  
[1] 16  
> sum(x>40&x<=41)  
[1] 8  
> sum(x>41&x<=44)  
[1] 12
```

## R.8 Testen mit R

```
> pnorm(39,40,1)-pnorm(36,40,1)
[1] 0.1586236
> pnorm(40,40,1)-pnorm(39,40,1)
[1] 0.3413447
> pnorm(41,40,1)-pnorm(40,40,1)
[1] 0.3413447
> pnorm(44,40,1)-pnorm(41,40,1)
[1] 0.1586236
> ni=c(14,16,8,12)
> pi=c(0.16,0.34,0.34,0.16)
> chisq.test(x=ni,p=pi)
```

Chi-squared test for given probabilities

```
data:  ni
X-squared = 11.3235, df = 3, p-value = 0.0101
```

## R.8 Testen mit R

```
# Beispiel 3: Unabhängigkeitstest
```

```
> x=c(1,1,1,1,1,0,0,0,0,0)
```

```
> y=c(1,0,1,0,1,0,0,1,1,0)
```

```
> table(x,y)
```

|   | y |   |
|---|---|---|
| x | 0 | 1 |
| 0 | 3 | 2 |
| 1 | 2 | 3 |

$$\chi^2 = \sum_{i=1}^k \sum_{j=1}^l \frac{\left(N_{ij} - \frac{N_{i\bullet} N_{\bullet j}}{n}\right)^2}{\frac{N_{i\bullet} N_{\bullet j}}{n}} \stackrel{a}{\sim} \chi^2((k-1)(l-1)) \text{ unter } H_0$$

$$\chi^2 = \frac{n(N_{11}N_{22} - N_{12}N_{21})}{N_{\bullet 1}N_{\bullet 2}N_{1\bullet}N_{2\bullet}}$$

```
> chisq.test(x,y,correct=FALSE)
```

Pearson's Chi-squared test

data: x and y

X-squared = 0.4, df = 1, p-value = 0.5271

Warnmeldung:

In chisq.test(x, y, correct = FALSE) :

Chi-Quadrat-Approximation kann inkorrekt sein

## R.8 Testen mit R

### # Beispiel 4: Unabhängigkeitstest

*Tabelle 5.1.9: Tatsächliche und zu erwartende Häufigkeiten  
mit jeweiligen Abweichungen*

|          | Raucher         | Gelegenheits-<br>raucher | Nichtraucher      | Summe |
|----------|-----------------|--------------------------|-------------------|-------|
| weiblich | 4 (6.4)<br>-2.4 | 8 (8)<br>0               | 28 (25.6)<br>2.4  | 40    |
| männlich | 12 (9.6)<br>2.4 | 12 (12)<br>0             | 36 (38.4)<br>-2.4 | 60    |
| Summe    | 16              | 20                       | 64                | 100   |

```
> m=c(4,12,8,12,28,36)
> m=matrix(m,2,3)
> chisq.test(m)
```

Pearson's Chi-squared test

```
data: m
```

```
X-squared = 1.875, df = 2, p-value = 0.3916
```

## R.8 Testen mit R

```
> res=chisq.test(m)
> attach(res)
> statistic
X-squared
  1.875
> p.value
[1] 0.3916056
> observed
      [,1] [,2] [,3]
[1,]    4    8   28
[2,]   12   12   36
> expected
      [,1] [,2] [,3]
[1,]   6.4    8 25.6
[2,]   9.6   12 38.4
```

## R.9 Regression mit R

##### Einfache lineare Regression #####

*Tabelle 5.2.4: Arbeitstabelle zur Bestimmung einer KQ-Geraden*

| $i$   | $x_i$ | $y_i$ | $x_i y_i$ | $x_i^2$ | $\hat{y}_i$ | $\hat{u}_i$ | $ \hat{u}_i $ | $\hat{u}_i^2$ |
|-------|-------|-------|-----------|---------|-------------|-------------|---------------|---------------|
| 1     | 1     | 1     | 1         | 1       | 1.2         | -0.2        | 0.2           | 0.04          |
| 2     | 2     | 2     | 4         | 4       | 1.5         | 0.5         | 0.5           | 0.25          |
| 3     | 3     | 1     | 3         | 9       | 1.8         | -0.8        | 0.8           | 0.64          |
| 4     | 4     | 3     | 12        | 16      | 2.1         | 0.9         | 0.9           | 0.81          |
| 5     | 5     | 2     | 10        | 25      | 2.4         | -0.4        | 0.4           | 0.16          |
| Summe | 15    | 9     | 30        | 55      | 9           | 0           | 2.8           | 1.9           |

```
> x=c(1,2,3,4,5)
> y=c(1,2,1,3,2)
> lm(y~x)
```

Call:

```
lm(formula = y ~ x)
```

Coefficients:

```
(Intercept)          x
          0.9         0.3
```

## R.9 Regression mit R

```
> beta1dach=cov(x,y)/var(x)
> beta0dach=mean(y)-beta1dach*mean(x)
> beta0dach
[1] 0.9
> beta1dach
[1] 0.3
> reslm=lm(y~x)
> names(reslm)
[1] "coefficients" "residuals"      "effects"
[4] "rank"          "fitted.values"  "assign"
[7] "qr"           "df.residual"    "xlevels"
[10] "call"         "terms"          "model" "model"
> attach(reslm)
> coefficients
(Intercept)          x
          0.9          0.3
> ydach=fitted.values
> udach=residuals
```



## R.9 Regression mit R

```
> ydach
```

```
  1    2    3    4    5  
1.2 1.5 1.8 2.1 2.4
```

```
> udach
```

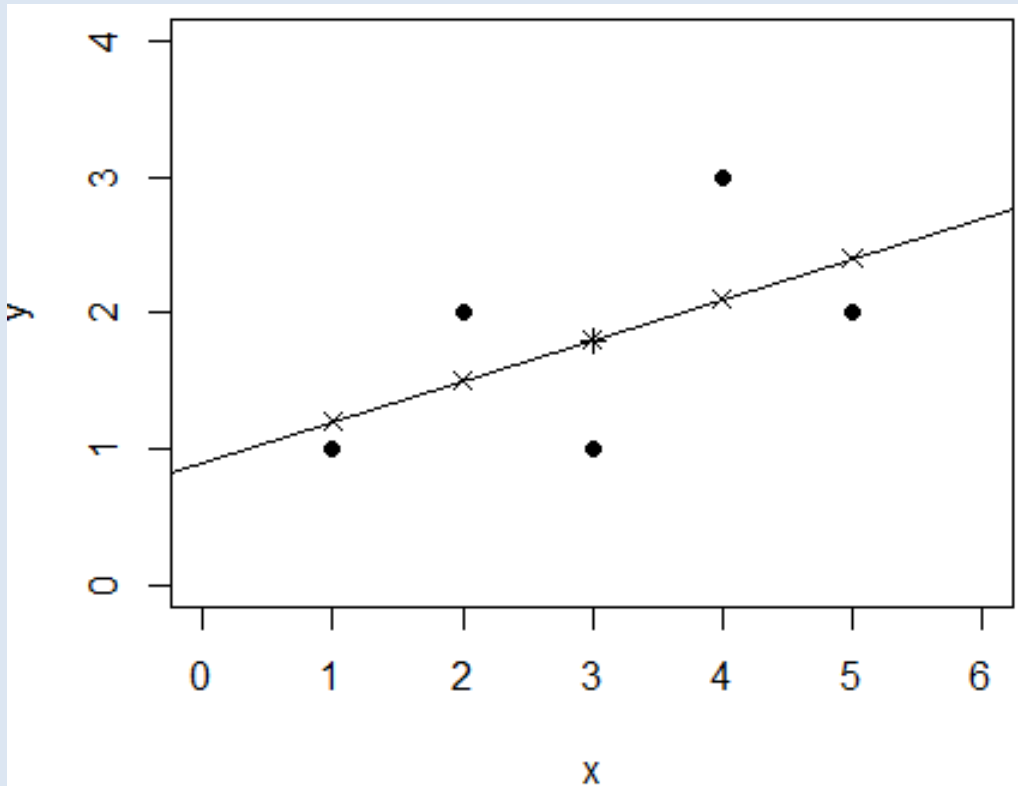
```
  1    2    3    4    5  
-0.2 0.5 -0.8 0.9 -0.4
```

### Satz 12.1.1: Eigenschaften der empirischen KQ-Regression

1. Die KQ-Gerade geht durch den Schwerpunkt  $(\bar{x}, \bar{y})$ .
2. Die Summe der gefitteten Werte ist gleich der Summe der  $y$ -Werte.
3. Die Summe der KQ-Residuen ist gleich 0.
4. KQ-Residuen und  $x$ -Werte sind unkorreliert.
5. Gefittete Werte und KQ-Residuen sind unkorreliert.
6. Es gilt die Streuungszerlegungsformel der KQ-Regression.

## R.9 Regression mit R

```
> plot(x,y,xlim=c(0,6),ylim=c(0,4),pch=16)  
> abline(reslm)  
> points(x,fitted.values,pch=4)  
> points(mean(x),mean(y),pch=8)
```



## R.9 Regression mit R

```
> sum(ydach)
[1] 9
> sum(y)
[1] 9
> sum(udach)
[1] 8.326673e-17
> cor(x,udach)
[1] -2.545783e-17
> cor(ydach,udach)
[1] 1.783105e-16
> var(y)
[1] 0.7
> var(ydach)+var(udach)
[1] 0.7
```

## R.9 Regression mit R

# Fallbeispiel 1: Modell KN (gleiche Daten wie zuvor)

$$\widehat{Wuchshöhe} = \begin{matrix} 0.9 \\ (0.83) \end{matrix} + 0.3 \cdot \begin{matrix} Wassermenge, \\ (0.25) \end{matrix} \quad R^2 \approx 0.32, \quad SER \approx 0.8.$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \quad \text{und} \quad \hat{\beta}_1 = \frac{\tilde{S}_{XY}}{\tilde{S}_X^2} \quad \text{mit} \quad \tilde{S}_X^2 > 0$$

$$\tilde{\sigma}_{\hat{\beta}_0}^2 = \frac{\frac{1}{n} \sum_{i=1}^n X_i^2 \hat{\sigma}_U^2}{n \tilde{S}_X^2}$$

$$\tilde{\sigma}_{\hat{\beta}_1}^2 = \frac{\hat{\sigma}_U^2}{n \tilde{S}_X^2},$$

$$\hat{\sigma}_U^2 = \frac{1}{n-2} \sum_{i=1}^n \hat{U}_i^2$$

$$\tilde{T}_{\beta_j} = \frac{\hat{\beta}_j - \beta_{j,0}}{\tilde{\sigma}_{\hat{\beta}_j}} \sim t(n-2) \quad \text{für} \quad \beta_j = \beta_{j,0} \quad \text{und} \quad j = 0, 1$$

$$[\hat{\beta}_j - t_{n-2, 1-\alpha/2} \cdot \tilde{\sigma}_{\hat{\beta}_j}, \hat{\beta}_j + t_{n-2, 1-\alpha/2} \cdot \tilde{\sigma}_{\hat{\beta}_j}]$$

## R.9 Regression mit R

```
# Geschätzte Varianzen und Standardfehler der KQ-Schätzer
# Bestimmtheitsmaß und SER
> varUdach=sum(udach^2) / 3
> var1dach=varUdach/ (4*var(x) )
> var0dach=mean(x^2) *varUdach/ (4*var(x) )
> SE0=sqrt(var0dach)
> SE1=sqrt(var1dach)
> SE0
[1] 0.8346656
> SE1
[1] 0.2516611
> R2=var(ydach) /var(y)
> R2
[1] 0.3214286
> 1-var(udach) /var(y)
[1] 0.3214286
> SER=sqrt(sum(udach^2) / 3)
> SER
[1] 0.7958224
```

## R.9 Regression mit R

```
# Zweiseitiger Test in Bezug auf Bewässerungsmenge
# Variante 1: Von Hand
> Tbeta1schlange=(beta1dach-0)/SE1           # Teststatistik
> Tbeta1schlange
[1] 1.192079
> kritWert=qt(0.975,3)
> kritWert
[1] 3.182446
> pWert=2*(1-pt(Tbeta1schlange,3))
> pWert
[1] 0.3189318
```

## R.9 Regression mit R

```
# Zweiseitiger Test in Bezug auf Bewässerungsmenge  
# Variante 2: Summary von lm()  
> summary(reslm)
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

| 1    | 2   | 3    | 4   | 5    |
|------|-----|------|-----|------|
| -0.2 | 0.5 | -0.8 | 0.9 | -0.4 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t ) |
|-------------|----------|------------|---------|----------|
| (Intercept) | 0.9000   | 0.8347     | 1.078   | 0.360    |
| x           | 0.3000   | 0.2517     | 1.192   | 0.319    |

Residual standard error: 0.7958 on 3 degrees of freedom

Multiple R-squared: 0.3214, Adjusted R-squared: 0.09524

F-statistic: 1.421 on 1 and 3 DF, p-value: 0.3189

## R.9 Regression mit R

### # Fallbeispiel 2: Modell BH

Abb. 12.1.15: Empirische Verteilung einzelner Variablen

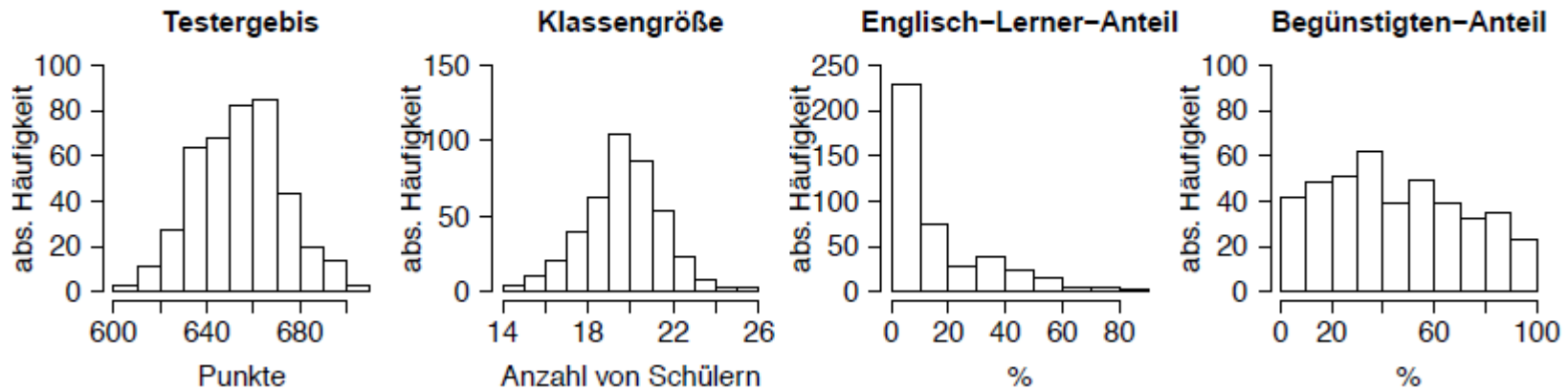
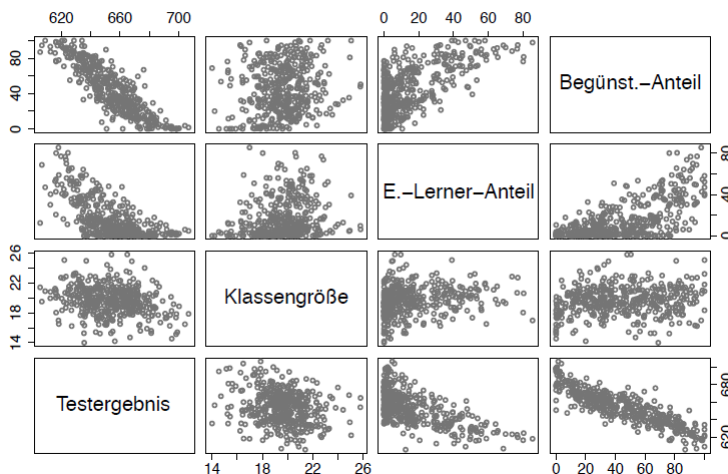


Abb. 12.1.16: Paarweise empirische Abhängigkeit der Variablen



|         | testscr | str     | elpct   | mealpct |
|---------|---------|---------|---------|---------|
| testscr | 1.0000  | -0.2264 | -0.6441 | -0.8688 |
| str     | -0.2264 | 1.0000  | 0.1876  | 0.1352  |
| elpct   | -0.6441 | 0.1876  | 1.0000  | 0.6531  |
| mealpct | -0.8688 | 0.1352  | 0.6531  | 1.0000  |



## R.9 Regression mit R

```
> library(Ecdat)      # Datensatz aus Paket „Ecdat“
> dim(Caschool)
[1] 420  15
> names(Caschool)
[1] "distcod"  "grspan"   "enrltot"  "teachers" "calwpct"
[6] "mealpct"  "computer" "testscr"  "compstu"  "expnstu"
[11] "str"      "avginc"   "elpct"    "readscr"  "mathscr"
> attach(Caschool)
> hist(testscr)
> hist(str)
> hist(elpct)
> hist(mealpct)
> X=cbind(testscr, str, elpct, mealpct)
> pairs(X, rowlattice=FALSE)
> round(cor(X), 4)
```

|         | testscr | str     | elpct   | mealpct |
|---------|---------|---------|---------|---------|
| testscr | 1.0000  | -0.2264 | -0.6441 | -0.8688 |
| str     | -0.2264 | 1.0000  | 0.1876  | 0.1352  |
| elpct   | -0.6441 | 0.1876  | 1.0000  | 0.6531  |
| mealpct | -0.8688 | 0.1352  | 0.6531  | 1.0000  |

## R.9 Regression mit R

$$\widehat{\text{Testergebnis}} = \begin{matrix} 693.93 \\ (10.34) \end{matrix} \quad \begin{matrix} -2.28 \cdot \text{Klassengröße}, \\ (0.52) \end{matrix} \quad R^2 \approx 0.05, \quad SER \approx 18.58$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \quad \text{und} \quad \hat{\beta}_1 = \frac{\tilde{S}_{XY}}{\tilde{S}_X^2} \quad \text{mit} \quad \tilde{S}_X^2 > 0$$

$$\hat{\sigma}_{\hat{\beta}_0}^2 = \frac{\frac{1}{n} \sum_{i=1}^n \hat{H}_i^2 \hat{U}_i^2}{n \left( \frac{1}{n} \sum_{i=1}^n \hat{H}_i^2 \right)^2}$$

$$\hat{\sigma}_{\hat{\beta}_1}^2 = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \hat{U}_i^2}{n (\tilde{S}_X^2)^2}$$

$$\hat{T}_{\beta_j} = \frac{\hat{\beta}_j - \beta_{j,0}}{\hat{\sigma}_{\hat{\beta}_j}} \stackrel{a}{\sim} N(0, 1) \quad \text{für} \quad \beta_j = \beta_{j,0} \quad \text{und} \quad j = 0, 1$$

$$[\hat{\beta}_j - z_{1-\alpha/2} \cdot \hat{\sigma}_{\hat{\beta}_j}, \hat{\beta}_j + z_{1-\alpha/2} \cdot \hat{\sigma}_{\hat{\beta}_j}]$$

## R.9 Regression mit R

```
# Inferenz im Modell BH mit Hilfe von vcovHC()
> reslm=lm(testscr~str)
> betadach=reslm$coefficients
> betadach
(Intercept)          str
 698.932952    -2.279808
> ydach=reslm$fitted.values
> udach=reslm$residuals
> R2=var(ydach)/var(testscr)
> SER=sqrt(sum(udach^2)/418)
> R2
[1] 0.0512401
> SER
[1] 18.58097
> library(sandwich)
> covdach=vcovHC(reslm,type="HC0")
> covdach      # Geschätzte Kovarianzmatrix
              (Intercept)          str
(Intercept) 106.908432 -5.338367
str         -5.338367  0.268584
```

## R.9 Regression mit R

```
> diag(covdach) # Geschätzte Varianzen auf der Diagonalen
(Intercept)      str
106.908432      0.268584
> SE=sqrt(diag(covdach)) # Standardfehler (SE's)
> SE
(Intercept)      str
10.3396534      0.5182509

# Zweiseitiger Test in Bezug auf Klassengröße
> Tbetaldach=betadach[2]/SE[2] # Teststatistik
> Tbetaldach
      str
-4.399044
> kritWerte=qnorm(c(0.025,0.975))
> kritWerte
[1] -1.959964  1.959964
> pWert=2*pnorm(Tbetaldach)
> pWert
      str
1.08729e-05
```

## R.9 Regression mit R

```
##### Multiple lineare Regression #####  
# Partielle Regression
```

*Tabelle 12.2.1: Nachhilfeunterricht an einer Sprachschule*

| Schüler | Stundenzahl | Punkteanzahl im<br>Eingangstest | Punkteanzahl im<br>Sprachtest |
|---------|-------------|---------------------------------|-------------------------------|
| 1       | 4           | 190                             | 52                            |
| 2       | 4           | 280                             | 66                            |
| 3       | 3           | 300                             | 58                            |
| 4       | 3           | 410                             | 77                            |
| 5       | 2           | 420                             | 70                            |
| 6       | 2           | 490                             | 91                            |
| 7       | 1           | 510                             | 81                            |
| 8       | 1           | 630                             | 95                            |

|    | ST      | S       | ET      |
|----|---------|---------|---------|
| ST | 1.0000  | -0.7887 | 0.9444  |
| S  | -0.7887 | 1.0000  | -0.9273 |
| ET | 0.9444  | -0.9273 | 1.0000  |

## R.9 Regression mit R

Abb. 12.2.2: Lineares Herausrechnen von ET aus ST und S

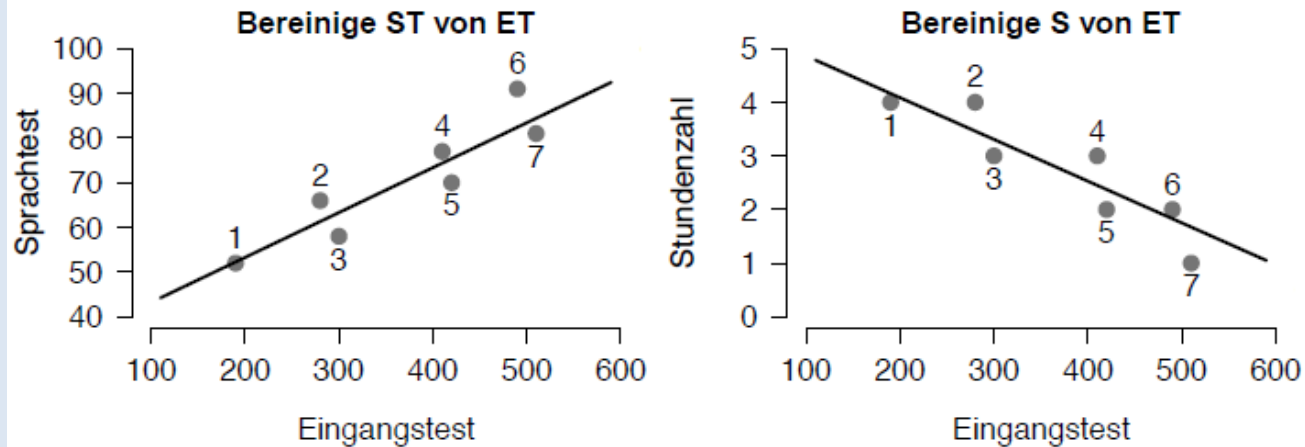
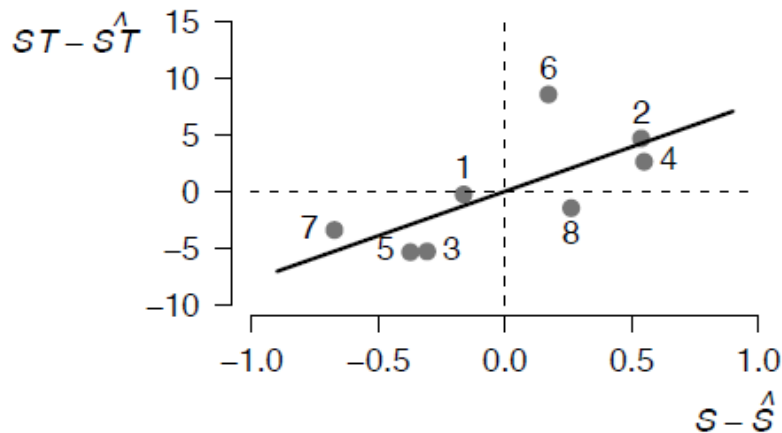


Abb. 12.2.3: Partieller Regressionsplot von S und ST unter ET



## R.9 Regression mit R

```
> S=c(4,4,3,3,2,2,1,1)
> ET=c(190,280,300,410,420,490,510,630)
> ST=c(52,66,58,77,70,91,81,95)
> X=cbind(ST,S,ET)
> round(cor(X),4)
```

|    | ST      | S       | ET      |
|----|---------|---------|---------|
| ST | 1.0000  | -0.7887 | 0.9444  |
| S  | -0.7887 | 1.0000  | -0.9273 |
| ET | 0.9444  | -0.9273 | 1.0000  |

```
> reslm1=lm(ST~ET)
> plot(ET,ST,xlab="Eingangstest",ylab="Sprachtest")
> abline(reslm1)
> text(ET,ST,1:8,pos=4)
```

```
> reslm2=lm(S~ET)
> plot(ET,S,xlab="Eingangstest",ylab="Stundenzahl")
> abline(reslm2)
> text(ET,S,1:8,pos=4)
```

## R.9 Regression mit R

```
> udach1=reslm1$residuals
> udach2=reslm2$residuals
> plot(udach2,udach1)
> cor(udach1,udach2)
[1] 0.7072275 # partielle Korrelation
> reslmresiduals=lm(udach1~udach2)
> reslmresiduals

Call:
lm(formula = udach1 ~ udach2)

Coefficients:
(Intercept)      udach2
  1.463e-16    7.879e+00 # partieller Effekt
                        # ggf. = direkter Effekt

> abline(reslmresiduals)
> text(udach2,udach1,1:8,pos=4)
```



## R.9 Regression mit R

# Äquivalenz partieller und multipler Regression

Die KQ-Regressionskoeffizienten einer multiplen Regression stimmen mit den Koeffizienten einer entsprechenden partiellen Regression überein.

```
> lm(ST~S+ET)
```

Call:

```
lm(formula = ST ~ S + ET)
```

Coefficients:

| (Intercept) | S      | ET     |
|-------------|--------|--------|
| -11.2777    | 7.8795 | 0.1618 |

## R.9 Regression mit R

# Direkte und indirekte Effekte im 3-Variablen-Fall

$$\beta_{Y \sim X} = \beta_X + \beta_Z \beta_{Z \sim X}$$

Gesamteffekt = Direkter Effekt + Indirekter Effekt.

```
> lm(ST~S)$coefficients
```

```
(Intercept)          S  
    98.75         -10.00
```

```
> lm(ST~S+ET)$coefficients
```

```
(Intercept)          S          ET  
-11.2776590    7.8794946    0.1618054
```

```
> lm(ET~S)$coefficients
```

```
(Intercept)          S  
    680.0         -110.5
```

```
> 7.8794946+0.1618054*(-110.5)
```

```
[1] -10
```

## R.9 Regression mit R

```
# Inferenz: Fallbeispiel 2 fortgesetzt
# Modell BH in verschiedenen Varianten
```

*Tabelle 12.2.2: Geschätzte Varianten für Modell BH*

| Modell<br>Regressor | 1                 | 2                | 3                | 4                |
|---------------------|-------------------|------------------|------------------|------------------|
| Konstante           | 698.93<br>(10.34) | 686.03<br>(8.70) | 702.91<br>(5.50) | 700.15<br>(5.54) |
| Klassengröße        | -2.28<br>(0.52)   | -1.10<br>(0.43)  | -1.12<br>(0.27)  | -1.00<br>(0.27)  |
| E.-Anteil           |                   | -0.65<br>(0.03)  |                  | -0.12<br>(0.03)  |
| B.-Anteil           |                   |                  | -0.60<br>(0.02)  | -0.55<br>(0.02)  |
| $R^2$               | 0.0512            | 0.4264           | 0.7668           | 0.7745           |
| SER                 | 18.58             | 14.46            | 9.22             | 9.08             |

## R.9 Regression mit R

```
> attach(Caschool)
> reslm_M1=lm(testscr~str)
> reslm_M2=lm(testscr~str+elpct)
> reslm_M4=lm(testscr~str+elpct+mealpct)

> betadach_M1=reslm_M1$coefficients
> betadach_M2=reslm_M2$coefficients
> betadach_M4=reslm_M4$coefficients

> library(sandwich)
> covdach_M1=vcovHC(reslm_M1, type="HC0")
> covdach_M2=vcovHC(reslm_M2, type="HC0")
> covdach_M4=vcovHC(reslm_M4, type="HC0")
> SE_M1=sqrt(diag(covdach_M1))
> SE_M2=sqrt(diag(covdach_M2))
> SE_M4=sqrt(diag(covdach_M4))
```

## R.9 Regression mit R

```
> betadach_M1
```

```
(Intercept)          str  
698.932952    -2.279808
```

```
> SE_M1
```

```
(Intercept)          str  
10.3396534    0.5182509
```

```
> betadach_M2
```

```
(Intercept)          str          elpct  
686.0322487    -1.1012959    -0.6497768
```

```
> SE_M2
```

```
(Intercept)          str          elpct  
8.69699568    0.43129850    0.03092074
```

```
> betadach_M4
```

```
(Intercept)          str          elpct          mealpct  
700.1499647    -0.9983092    -0.1215733    -0.5473456
```

```
> SE_M4
```

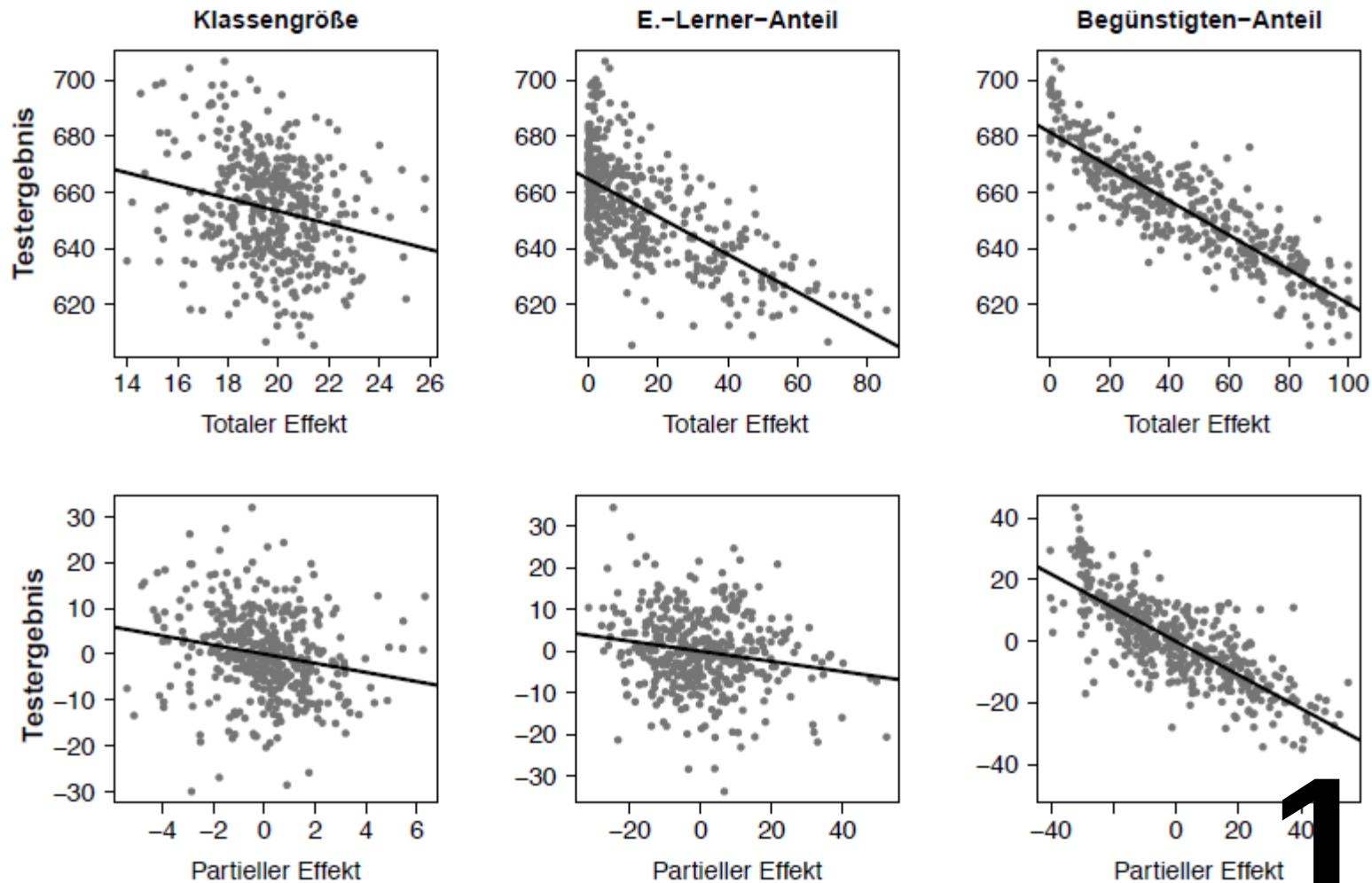
```
(Intercept)          str          elpct          mealpct  
5.54186975    0.26879069    0.03267497    0.02399218
```

## R.9 Regression mit R

```
> ydach_M1=reslm_M1$fitted.values
> ydach_M2=reslm_M2$fitted.values
> ydach_M4=reslm_M4$fitted.values
> udach_M1=reslm_M1$residuals
> udach_M2=reslm_M2$residuals
> udach_M4=reslm_M4$residuals
> R2_M1=var(ydach_M1)/var(testscr)
> R2_M2=var(ydach_M2)/var(testscr)
> R2_M4=var(ydach_M4)/var(testscr)
> SER_M1=sqrt(sum(udach_M1^2)/418)
> SER_M2=sqrt(sum(udach_M2^2)/417)
> SER_M4=sqrt(sum(udach_M4^2)/416)
> c(R2_M1,SER_M1)
[1] 0.0512401 18.5809675
> c(R2_M2,SER_M2)
[1] 0.4264314 14.4644850
> c(R2_M4,SER_M4)
[1] 0.7745159 9.0800791
```

## R.9 Regression mit R

*Abb. 12.2.6: Totale und direkte Effekte der Einflussvariablen*



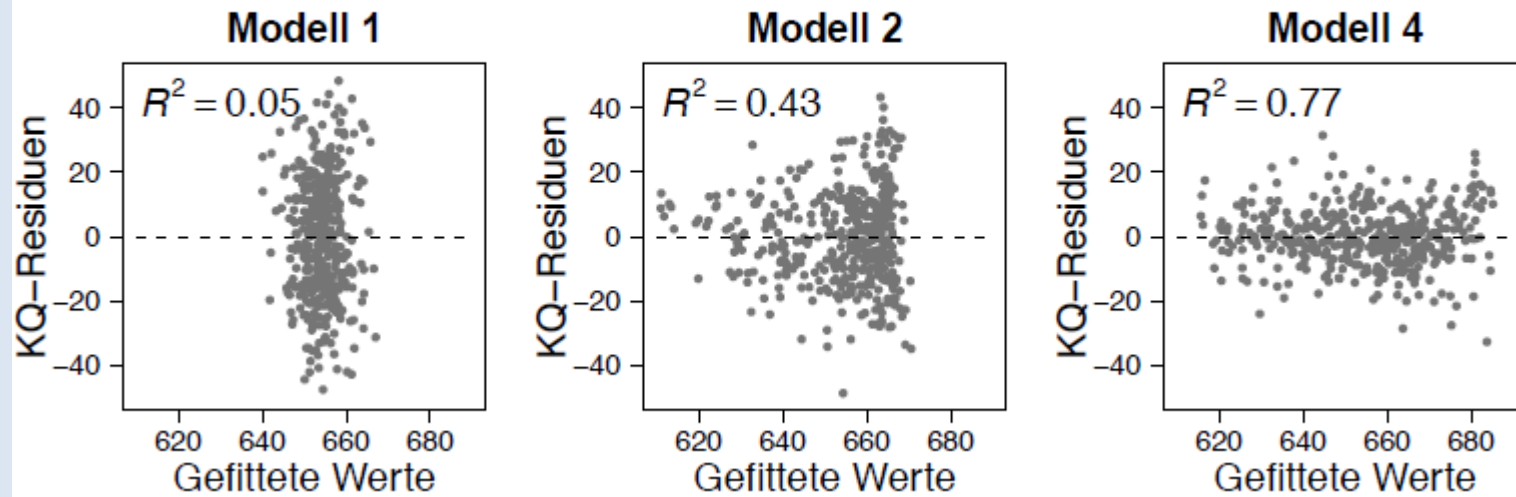
## R.9 Regression mit R

```
> reslm_M1$coefficients
(Intercept)          str
  698.932952    -2.279808          # Totaler Effekt
> plot(str, testscr)
> abline(reslm_M1)
> udach1=lm(testscr~elpct+mealpct)$residuals
> udach2=lm(str~elpct+mealpct)$residuals
> reslmresiduals=lm(udach1~udach2)
> reslmresiduals$coefficients
(Intercept)          udach2
 3.922878e-17 -9.983092e-01          # Direkter Effekt
> plot(udach2, udach1)          # in Modell 4
> abline(reslmresiduals)
```



## R.9 Regression mit R

Abb. 12.2.7: Residualplots der Modelle 1, 2 und 4



```
> plot(ydach_M1,udach_M1,xlim=c(620,680),ylim=c(-40,40))  
> plot(ydach_M2,udach_M2,xlim=c(620,680),ylim=c(-40,40))  
> plot(ydach_M2,udach_M4,xlim=c(620,680),ylim=c(-40,40))
```