

Datenstrukturen

```
list(x=w , y=z)
data = data.frame(data=cbind(c(),c())) #2 gleich lange Vektoren zu Tabelle basteln
names(data)<-c("x","y") #benenne Tabellenspalten mit x und y
read.table(file="clipboard", header=TRUE,dec=",")
read.csv(file = "") # .
read.csv2(file = "") # ,
```

Regression

```
lm(y~l(x)+l(x^2), data) #data ist data.frame
- residuals #ist regression mit Regressoren x und x^2
pars = - $coefficients #Vector von (my-huts abzügl y)^2
plot(data) #betas
#plotte daten um die Näherung zu sehen
#füge my-hut kurve hinzu
curve(pars[1]+x*pars[2]+x^2*pars[3],min(a2data[,1]),max(a2data[,2]),add=TRUE)
summary(linmod.a4)
anova(linmod.a4) #Signifikanz-Codes??
```

Variante:

```
attach(a3data)
lm(Preis~Alter,a3data)
```

Gleichungssysteme und Matrizen

```
matrix(c(27,13,3,7),nrow=2) #Matrix mit n Zeilen // ncol Spalten
c() #Vector
rep(1,n) #Vector mit n mal der 1
solve(matrix(c(),nrow,ncol),c()) #löse Gleichungssystem
solve(matrix(c(),nrow,ncol)) #invertiere Matrix
eigen() #Eigenwerte und Vektoren einer Matrix
floor() #abrunden
ceiling(x) #aufrunden
NROW(data) #Anzahl Zeilen einer Matrix
```

Clustern

```
dmat=dist(data,method="euclidean") #erzeuge Diszanzmatrix
hclust(dmat, method="average") #erstelle Clusterung
-$height #Homogenitätsgrade aufsteigend
cutree(hclust,h=40, k=Anzahl Cluster) #Vektor mit zugewiesenen Clusterzahlen
data$clust=cutree #Vektor als Spalte in data einfügen
partdata=data[data$clust==3,] #data mit allen Zeilen, für die gilt,...

plot(rev(hg),type="l") #Homogenitätsgrade Plotten für Knick erkennen
rug(hg,side=2,col="red") #Homos an die linke Seite
```

Chi-Quadrat-Statistik

```
zusammenhang = function(table){
n=sum(table)
d=dim(table)
cs=chisq.test(table,correct=FALSE)$statistic[[1]] #erzeuge Kontingenztabelle, dann:
ckk=sqrt(cs/(cs+n)) #ist chi-quadrat
ckkk=ckk*sqrt(min(d)/(min(d)-1))
cv=sqrt(cs/(n*(min(d)-1)))
list() #das fehlt, siehe Datenstrukturen
```

```
//
kont3=read.table(file="clipboard",header=TRUE)
chisquare.prod=chisq.test(kont3$Losqual,kont3$Prod,correct=FALSE)
chisquare.prod$statistic
```

Q-Plot (eigener)

```
rgamma(1000,1,1)           #1000 Zufallszahlen nach Gamma(1,1)
fractions <- 1:n/(n+1)     #Vektor mit 1/(n+1) bis n/(n+1)
qgamma(fractions,a,b)     #frage Quantile zu obigen fractions ab
sort(data)                #für eigenen qq-Plot, sortiere aufsteigend
abline(0,1,col="")        #45°-Linie in Farbe ""
```

eigene Funktionen

```
"%hoch%" = function() {} #jetzt Aufruf durch A%hoch%b möglich
```

Lage,Skala, Beziehung

```
summary(data)             #min,max,quartile,median,mittel
sd(data)                  #Standardabweichung
var(data)                 #Varianz
fivenum(data)            # Vektor aus (minimum, lower-hinge, median, upper-hinge,
                           maximum)
cor(data) oder cor(c,c)   #Korrelationskoeffizient (pearson)
cov(data)                 #Kovarianz
mean(data)                #arithm. Mittel
quantile()
ecdf(data)                #empirische Verteilungsfunktion
```

Barplot

```
barplot(data,col=c("", ""),names.arg=c(),main=" ")
```

#Stabdiagramm

Boxplot

```
boxplot(data$Preis,horizontal=TRUE)
```

Histogramm

```
hist(data$Alter,freq=FALSE)
```

```
#freq = FALSE: relative statt absolute Häufigkeiten
```

```
pie()
```

```
#Tortendiagramm
```

```
qqplot(x,y,...)
```

```
#QQ-Plot
```

Grafikelemente hinzufügen

```
par(mfrow=c(,))
```

```
#Änderung Grafikausgabe
```

```
rug(data$Preis,col=c("red"), side=1)
```

```
#füge Striche(Daten) an der Seite eines Plots etc. an, 2 links
```

```
lines(density(Alter),col="red")
```

```
#beliebige Funktion hinzufügen, hier: Kernsichteschätzer
```

```
points(c(), col="")
```

```
#Punkte gemäß Vektor hinzufügen
```

```
abline(0,1,col="")
```

```
#45°-Linie in Farbe ""
```

sonstiges

```
attach(data)
```

```
detach(data)
```

```
rev(hg)
```

```
#Reihenfolge im Vektor umdrehen (z.B. Homo mein Clust)
```

```
proc.time()
```

```
#Aufruf der Systemzeit zum Vergleich (öfter)
```

```
choose(n,k)
```

```
#n über k
```

seq(Startwert, Endwert, Feinheit in .001)

Zugriffe in ein Feld

Matrizen

[,1] erste Spalte

[1,] erste Zeile

Listen

[[1]] erstes Bestandteil einer Liste

\$name wenn listenElemente benannt

data[data\$clust==3,] #data mit allen Zeilen, für die gilt,...

Vektoren

[3:5] in Vector

Quantilfunktionen

qbinom(0.25,20,0.4)

Zufallszahlen

rbinom(100,20,0.4)

plot(ecdf(rbinom(100,20,0.5)))

EinzelWS, Dichte

dbinom(5,10,0.5)

VF

pbinom(5,10,0.5)