

# Kurzeinführung in R

## im Rahmen der Vorlesung „Einführung in die Statistik“

---

### Nützliche R-Befehle

help.start()	Öffnet ein Browserfenster mit verschiedenen R-Dokumentationen
help( <i>Befehl</i> ), ? <i>Befehl</i>	Zeigt Hilfe zu <i>Befehl</i> an
q()	R beenden
ls(), objects()	Listet die definierten Objekte auf
rm()	Entfernen von Objekten
library()	Bibliothek mit weiteren R-Funktionen
data()	Beispiel-Datensätze in R
read.table()	Funktion um eigene Daten zu importieren
mean()	Mittelwert eines Datenvektors
var()	Stichprobenvarianz
cor()	Korrelation
length()	Länge
sum()	Summe der Komponenten
prod()	Produkt der Komponenten
sqrt()	Quadratwurzel der einzelnen Komponenten
max()	Maximum
min()	Minimum
sort()	Vektor aufsteigend sortieren
abs()	Absolutbeträge der Komponenten
plot()	allgemeine Graphikfunktion
hist()	Histogramm
lines()	Verbindungsline zwischen Datenpunkten
abline()	gerade Linie
dunif(), punif(), runif()	Dichte, Verteilungsfunktion und simulierte Zufallsvariable der Uniformverteilung
dbinom(), pbinom(), rbinom()	Binomialverteilung
dhyper(), phyper(), rhyper()	Hypergeometrische Verteilung
dgeom(), pgeom(), rgeom()	Geometrische Verteilung
dnbinom(), pnbinom(), rnbinom()	Binomialverteilung
dpois(), ppois(), rpois()	Poisson-Verteilung
dexp(), pexp(), rexp()	Negativ-Exponentialverteilung
dgamma(), pgamma(), rgamma()	Gamma-Verteilung
dnorm(), pnorm(), rnorm()	Normalverteilung
dcauchy(), pcauchy(), rcauchy()	Cauchy-Verteilung

## Mein erstes R-Programm

```
x <- „Hello, World!“  
x
```

## Zuweisen, Anzeigen und Manipulieren von Vektoren in R

```
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)  
x  
sortiert <- sort(x)  
sortiert  
1/x  
x[4]  
x[2:4]  
x[lenght(x)]  
x[2] <- 9.9  
x
```

```
y <- c(4 ,9, 2:5)  
y  
sum(y)  
max(y)  
sqrt(y)  
quadrat <- y^2  
y; quadrat; y^3
```

```
z <- 31  
z  
z*y  
y-z  
x+y  
y+c(-1,2)  
x*y  
y*c(-1,2)  
z <- c(y, x, y)  
z  
w <- rep(x, 3)  
w  
ls()  
rm(sortiert, quadrat, w, x, y, z)  
ls()
```

```
sequenz1 <- 1:100
sequenz1
n <- 50
sequenz1 <- 1:n
sequenz1
sequenz2 <- seq(10, 20, 0.01)
sequenz2

pos.und.neg <- c(1, 4, -5, -3, 0, 7, -4, 7, -1, 2, 2, 2, -2)
neg <- pos.und.neg[pos.und.neg < 0]
nicht.neg <- pos.und.neg[pos.und.neg >= 0]
neg
nicht.neg
absolut <- abs(pos.und.neg)
rm(sequenz1, sequenz2, pos.und.neg, neg, nicht.neg)
```

## Simulation von Zufallsvariablen und ihre graphische Darstellung

### Histogramme standardnormalverteilter Stichproben

```
n1 <- rnorm(100)
n1
mean(n1); var(n1)
summary(n1)
hist(n1, main="Histogramm einer N(0,1)-verteilten Stichprobe")
hist(n1, main="Histogramm einer N(0,1)-verteilten Stichprobe", ylab="Häufigkeit")
hist(n1, main="Histogramm einer N(0,1)-verteilten Stichprobe", ylab="relativeHäufigkeit",
     probability=TRUE)
stem(n1)
n2 <- rnorm(1000)
hist(n2)
n3 <- rnorm(10000)
hist(n3)
hist(n3, nclass=100)
hist(n3, nclass=100, probability=TRUE)
```

### Dichte der Negativ-Exponentialverteilung graphisch darstellen

```
a <- seq(0, 10, 0.1)
plot(a, dexp(a))
werte <- locator()
werte
plot(a, dexp(a), type="l")
```

## Verteilungen von X und F(X)

```
stichprobe1 <- rcauchy(1000)
plot(stichprobe1)
stichprobe1 <- pcauchy(stichprobe1)
plot(stichprobe1)
stichprobe2 <- rgamma(1000, shape=2)
plot(stichprobe2)
stichprobe1 <- pgamma(stichprobe2, shape=2)
plot(stichprobe2)
stichprobe3 <- runif(1000)
plot(stichprobe3)

rm(n1, n2, n3, a, werte, stichprobe1, stichprobe2, stichprobe3)
```

## Definieren und Abrufen eigener Funktionen

### Standardabweichung aus der Stichprobenvarianz eines Vektors

```
standabw <- function(x) { sqrt(var(x)) }

datensatz <- c(4, 6.8, 7.005, 3, 0, 0, -1, 3, 3, 4.5, 0.5)
standabw(datensatz)
```

### Wahrscheinlichkeiten berechnen

```
wahrsch <- function(oben, unten, t) {
  pexp(oben, rate=t) - pexp(unten, rate=t)
}

wahrsch(7, 6, 0.5)

g <- function(n, p) {
  if (n<0) print("Als erstes Argument bitte nicht-negative Zahl eingeben!")
  else {
    if (p<0 || p>1) print("Als zweites Argument bitte Wahrscheinlichkeit eingeben!")
    else dgeom(n, prob=p)
  }
}

g(-7,0.5)
g(10,0.7)
```

### Pfad einer Poisson-verteiltern Zufallsvariablen

```
poisson0 <- function(anzahl, EW) {
  x <- rpois(n=anzahl, lambda=EW)
  plot(x, xlab="n", ylab="X", type="l")
  abline(h=EW, col=2, lty=3)
}
```

```
poisson0(300, 20)
```

### Simulation des SGGZ für Poisson-vertelte Zufallsvariablen (ein Pfad)

```
poisson1 <- function(anzahl, EW) {
  x <- rpois(n=anzahl, lambda=EW)
  S <- rep(0, anzahl)
  for(i in 1:anzahl){
    S[i] <- sum(x[1:i])/i
  }
  plot(S, main="SGGZ für unabhängige Po-Variablen", xlab="n", ylab="Sn/n", type="l")
  abline(h=EW, col=2, lty=3)
}
```

```
poisson1(100, 5)
poisson1(10000, 8)
```

### Simulation des SGGZ für Poisson-vertelte Zufallsvariablen (mehrere Pfade)

```
poisson2 <- function(anzahl, EW, pfade) {
  plot(c(1:anzahl), c(EW-sqrt(EW), EW+sqrt(EW), rep(EW,anzahl-2)), main="SGGZ für
  unabhängige Po-Variablen", xlab="n", ylab="Sn/n", type="n")
  abline(h=EW, lty=3)
  for(j in 1:pfade){
    x <- rpois(n=anzahl, lambda=EW)
    S <- rep(1, anzahl)
    for(i in 1:anzahl){
      S[i]<-sum(x[1:i])/i
    }
    lines(S, col=j+1)
  }
}
```

```
poisson2(1000, 50, 7)
```

## Beispiel-Datensätze in R

Das folgende Beispiel wird in ähnlicher Form auch in „An Introduction to R“ (mit `help.start()` aufrufbar) betrachtet. Die Datei *faithful* enthält 262 Beobachtungswerte zur Eruptionlänge und Wartezeit auf den nächsten Ausbruch des Old Faithful Geysirs im Yellowstone Nationalpark in den USA.

```
data()
data(faithful)
faithful

plot(faithful, main="Vergleich zwischen Eruptionlänge und Wartezeit auf nächsten
      Ausbruch")
cor(faithful)

hist(faithful$eruptions, seq(1.6, 5.2, 0.2), probability=TRUE, main="Histogramm der relativen
      Häufigkeiten der Eruptionen des Old Faithful", xlab="Dauer in Minuten",
      ylab="Relative Häufigkeit")
lines(density(faithful$eruptions), col=2, lty=2)
lines(density(faithful$eruptions, bw=0.1), col=3, lty=2)

library(stepfun)
plot(ecdf(faithful$eruptions))
plot(ecdf(faithful$eruptions), do.points=FALSE, verticals=TRUE, main="Empirische
      Verteilungsfunktion der Eruptionen des Old Faithful")

lange.eruptionen <- faithful$eruptions[faithful$eruptions > 3]
plot(ecdf(lange.eruptionen), do.points=FALSE, verticals=TRUE, main="Eruptionen, länger als
      3 Minuten")
x <- seq(3, 5.5, 0.01)
lines(x, pnorm(x, mean=mean(lange.eruptionen), sd=sqrt(var(lange.eruptionen))), lty=3,
      col=2)

q()
```