

Übersicht R-Befehle

Workspace-Befehle

Befehl	Bedeutung
ls(), objects()	Listet alle aktuellen Objekte des Arbeitsspeichers auf
rm(Obj1,...,Objn)	Löscht die angegebenen Objekte Obj1,...,Objn aus dem Arbeitsspeicher
edit(file="Filename.R")	Erstellen eines Files, welches R-Objekte enthält
source("Filename.R")	Einlesen der R-Objekte, die in einem File (definiert werden) stehen
sink("Filename.lis")	Umlenken aller Ausgaben von R in einen File
sink()	Rücklenken der R-Ausgabe auf den Bildschirm

Datentypen und Operatoren in R

Befehl	Bedeutung
x<-2	Zuweisungen
c(1,2,3,4)	Vektor der Länge 4
date()	Datum
pi	Die Zahl Pi
list(O1,...,On)	Liste von R-Objekten verschiedener Typen
„a“	Charakter (in „“)
a + bi	Komplexe Zahlen
z<-scan(what="character")	Schnelle Eingabe eines Charakter-Vektors über den Bildschirm . Default: numeric.
Built-in-Funktionen	sin(), cos(), tan(), cot(), exp(), sqrt(), log() , log2(), log10(), logb()
round()	Runden
trunc()	Abschneiden
Arithmetische Op.	+, -, *, /, ^, %*%
Boolsche Op.	! negation , & && and, or
Vergleichsoperatoren	== , <=, >=, !=, <, >
class(objekt)	Gibt den Objekt-Typ zurück
is.integer(), is.vector(), is.data.frame() usw.	Erfragt den Objekt-Typ
as.integer(), as.vector(), as.data.frame() usw.	Wandelt ein R-Objekt in den entsprechenden Typ um

Definition von Sequenzen

Befehl	Bedeutung
c(a:b)	Vektor der Zahlen von a bis b (Schrittweite s=1), für a > b ist die Schrittweite s=-1
c(2*1:15)	is the vector c(2, 4, ..., 28, 30)
c(2*1:15-3)	is the vector c(-1, 1, ..., 25, 27)
c(1,1,3:1,1:4,3)	is the vector c(1,1,3,2,1,1,2,3,4,3)
seq(from= -5, to= 5, by=0.2)	is the vector c(-5, -4.8,...,4.8,5)
seq(length=51, from=-5,by=.2)	is the same vector
rep(x, times=5)	put five copies of x end-to-end in a vector
rep(x, each=5)	repeats each element of x five times before moving on to the next.
X<-c(1:4);rep(x,x)	is the vector c(1,2,2,3,3,3,4,4,4,4)

Übersicht R-Befehle

Zeichenketten – Befehle

Befehl	Bedeutung
<code>\\</code>	Darstellung von <code>\</code> in einer Zeichenkette
<code>\“</code>	Darstellung von <code>“</code> in einer Zeichenkette
<code>\n</code>	Darstellung von newline in einer Zeichenkette
<code>\t</code>	Darstellung von tab in einer Zeichenkette
<code>\b</code>	Darstellung von backspace in einer Zeichenkette
<code>paste("a","b",...)</code>	Fügt alle String-Objekte a, b,... zu einer Zeichenkette "a b" zusammen Ergebnis ist der Vektor <code>c("1","2","3","4","5","6")</code>
<code>paste(c("X"), 1:5, sep="")</code>	Ist der Vektor <code>"X1" "X2" "X3" "X4" "X5"</code>
<code>paste(c("X","Y"), 1:4, sep="~")</code>	Ist der Vektor <code>"X~1", "Y~2", "X~3", "Y~4"</code>
<code>cat(Obj1,...,Objn , sep = " ")</code>	'cat' converts its arguments to character strings, concatenates them, separating them by the given 'sep=' string, and then prints them. No linefeeds are printed unless explicitly requested by <code>\n</code> .
<code>x<-c(1:5);y<-c(1:10); cat(5,":",sum(x),"\n",10,":",sum(y),"\n")</code>	Ausgabe von <code>5 : 15</code> <code>10 : 55</code>

Befehle für Matrizen

Befehl	Bedeutung												
<code>dim(vec) <-c(k,n)</code>	Erstellen einer Matrix mit k Zeilen und n Spalten aus einem Vektor der Länge k·n.												
<code>matrix(vec,k,n)</code>													
<code>matrix(vec,k,n,byrow=FALSE)</code>	byrow: logical. If 'FALSE' (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.												
<code>matrix(c(1,2,3, 11,12,13),2,3, dimnames = list(c("row1", "row2"), c("C.1", "C.2", "C.3")))</code>	dimnames(): Namen für Zeilen und Spalten Liefert die Matrix <table style="margin-left: 40px;"> <tr> <td></td> <td>C.1</td> <td>C.2</td> <td>C.3</td> </tr> <tr> <td>row1</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>row2</td> <td>11</td> <td>12</td> <td>13</td> </tr> </table>		C.1	C.2	C.3	row1	1	2	3	row2	11	12	13
	C.1	C.2	C.3										
row1	1	2	3										
row2	11	12	13										
<code>diag(a1,...,ak)</code>	Diagonalmatrix mit den Diagonalelementen a1,...,ak												
<code>A %**% B</code>	Matrizenmultiplikation. Es gibt kein Transponiert. Die Typen der Matrizen müssen für die Multiplikation passen, sonst gibt es eine Fehlermeldung)												
<code>x %**% A %**% x</code>	= $x^T A x$ falls x ein Vektor ist												
<code>Solve(A)</code>	ist die Inverse von A.												
<code>Solve(A,b)</code>	löst das Gleichungssystem $A \cdot x = b$.												

Übersicht R-Befehle

Auswahl von Elementen und Teilmatrizen aus Matrizen, Tabellen und data.frames Befehle für Matrizen

Befehl	Bedeutung
B[i,j]	(i,j)-tes Element
B[i,]	i.-te Zeile
B[,j] (j.te Spalte),	j.te Spalte
B[i1:i2,j1:j2]	Auswahl einer Teilmatrix: Zeilen i1 bis i2 und Spalten j1 bis j2
nrow(), ncol()	return the number of rows or columns present in 'x'. x: a vector, array or data frame
row.names(), col.names()	Abfragen und Zuweisen von Zeilen- und Spalten-Namen einer Matrix
Solve(A)	ist die Inverse von A.
Solve(A,b)	löst das Gleichungssystem $A \cdot x = b$.

Steuer-Befehle

Befehl	Bedeutung
if (cond) expr	expr wird solange ausgeführt, wie cond = TRUE ist.
if (cond) expr1 else expr2	
while (cond) expr	
repeat expr	∞ Wiederholung von expr bis break erreicht wird
for (var in list) expr	
break	innerhalb der bodys (expr) von repeat, for, while unmittelbarer exit aus den loop
next	innerhalb der bodys (expr) von repeat, for, while Sprung zum Beginn der nächsten Iteration des loops

Funktionen

Definition: fname <- function(arg1,arg2,..., name1=Wert1, name2=Wert2,...) {statements}
Aufruf: fname(val1,val2,..., valname1, valname2,...) oder fname(val1,val2,...)
Rückgabewert der Funktion : ist der Wert des letzten Statements (gewöhnlich einfach ein R Objekt), welches am Ende der Gruppe der Statements steht. Die Funktion kann an jeder Stelle terminiert werden durch den Befehl return()
Bsp: summe<- function(x,y,z=0.5,w=3) <pre> { if (length(x) == 0) return("Fehler in x"); if (length(y) == 0) return("Fehler in y"); s<-x+y+z+w; s} </pre>
Aufruf: summe(1,2,0.5,3) oder nur summe(1,2) , Rückgabe: s = 6.5

Übersicht R-Befehle

Befehle für das Erstellen von data.frames

Befehl	Bedeutung
edit(data.frame(z))	Editieren des Dataframes z mit dem R-Editor
row.names(z)	Zeilenamen des Dataframes z
names(z)	Namen der Spalten (Vektoren) des Dataframes z
data.frame(u)	Erstellen eines Dataframes aus der Matrix u
data.frame(x,y,z,...)	Erstellen eines Dataframes aus den Vektoren x,y,z,....
data.frame(V1=x,V2=y,V3=z,...)	Erstellen eines Dataframes aus den Vektoren x,y,z,.... mit den Spaltennamen: V1,V2,V3,...
read.table("filename")	Einlesen eines Dataframes, der unter „filename“ abgespeichert ist
read.table("meinwork/abc.txt", row.names=c("nr1","nr2"), header=T)	Einlesen eines Dataframes, die Zeilen werden benannt und die erste Zeile sollen als Header(Spaltennamen) interpretiert werden.
write.table(df,file="filename")	Abspeichern eines Dataframes df im File „filename“ .

Befehle für die Auswertung von data.frames

Befehl	Bedeutung
split(x, f, ...)	x: vector or data frame containing values to be divided into groups. f: a "factor" in the sense that 'as.factor(f)' defines the grouping, or a list of such factors in which case their interaction is used for the grouping.
str(z)	Anzeige von Informationen über den data.frame z
attach(z)	Im folgenden beziehen sich alle Variablennamen auf Variablen aus dem data.frame z
detach(z)	Rückgängig machen von attach()
table(X), table(z\$X)	Abs.Häufigkeiten der Variable X , im data.frame z
tapply(X,statistik)	Erzeugt eine Tabelle von statistiken für X als statistik kommt in Frage: mean, summary, max,...(siehe u.stehende Tabelle)
tapply(X,Y,statistik)	Erzeugt eine Tabelle von statistiken für X pro Werten für Y Bsp. tapply(Lebensdauer..Jahre.,Hersteller,summary)
lapply(list(X,Y,...),statistik)	Wendet die Funktion statistik(z.B. mean, summary usw.) auf alle Variablen der Liste an!
sapply(split(X,Y),statistik)	Erzeugt eine Tabelle von statistiken für X pro Werten für Y
df[order(X,decreasing=T),]	Im df werden die Zeilen nach Variable X in absteigender Reihenfolge sortiert
df[order(X,Y,decreasing=T),]	Innerhalb der nach X sortierten Gruppen gleicher Werte wird Y in absteigender Reihenfolge sortiert
sample(x,k)	Zufallsauswahl einer Stichprobe von k Zahlen aus einem Vektor x der Länge n
match(x, nr, nomatch = 0)	nr= Vektor 1:n. mach erzeugt einen Vektor der Länge n, der an allen Positionen des Vektors eine 0 einsetzt, die nicht in nr enthalten sind.

Übersicht R-Befehle

Statistik-Befehle auf Vektoren z – statistische Maßzahlen für eine Stichprobe (Urliste) z

Befehl	Bedeutung										
table(z)	Tabelle der absoluten Häufigkeiten der Elemente des Vektors z										
cut(c(1,1,2,2,2,3,4),br=0.5*0:8)	Erzeugen der Klassen: (0.5,1] (0.5,1] (1.5,2] (1.5,2] (1.5,2] (2.5,3] (3.5,4]										
table(cut(x, breaks = c(-3,-2,-1,1,2,3)));	Erstellen einer absol. Klassen-Häufigkeitstabelle für x, die Klassen werden durch den breaks-Parameter definiert., z.B. <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="padding-right: 20px;">(-3, -2]</td> <td style="padding-right: 20px;">(-2, -1]</td> <td style="padding-right: 20px;">(-1, 1]</td> <td style="padding-right: 20px;">(1, 2]</td> <td>(2, 3]</td> </tr> <tr> <td style="padding-right: 20px; text-align: center;">0</td> <td style="padding-right: 20px; text-align: center;">19</td> <td style="padding-right: 20px; text-align: center;">61</td> <td style="padding-right: 20px; text-align: center;">17</td> <td style="text-align: center;">3</td> </tr> </table>	(-3, -2]	(-2, -1]	(-1, 1]	(1, 2]	(2, 3]	0	19	61	17	3
(-3, -2]	(-2, -1]	(-1, 1]	(1, 2]	(2, 3]							
0	19	61	17	3							
hist(x,breaks =...,plot=FALSE)	Erzeugung einer Klassenhäufigkeitstabelle mit verschiedenen Angaben, z.B. Klassenmitte										
mean(z)	Arith. Mittel eines Vektors z										
min()	Minimum aller Elemente von z										
max()	Maximum ""										
sum()	Summe ""										
prod()	Produkt""										
range()	Differenz zwischen größten und kleinsten Wert des Vektors z										
length()	Länge des Objekts z (Vektor, data.frame)										
var(z)	Streuung der Elemente von z										
cumsum(z)	Kummulierte Summe der Elemente von z										
sort(z)	Sortiert die Elemente von z										
sample(x,k)	Zufallsauswahl einer Stichprobe von k Zahlen aus einem Vektor x der Länge n										
match(x, nr, nomatch = 0)	nr= Vektor 1:n. mach erzeugt einen Vektor der Länge n, der an allen Positionen des Vektors eine 0 einsetzt, die nicht in nr enthalten sind.										
rnorm(n)	Erzeugung von n Standardnormalverteilten Zufallszahlen										
rank(x)	Rangplatzwerte des Vektors x										
quantile(x,type=n)	Berechnet 0%, 25%,50%,75% und 100%-Quantil n = integer von 1 bis 7 (Default = 7). n = Typ des Quantils. n = 1: der x-Wert für den p*100% aller Daten ≤ x sind ! n= 5 : entspricht der Definition in der Vorlesung: 50% ist z.B. der Wert x, der die geordnete Stichprobe halbiert. Liegt er zwischen 2 Werten, wird gemittelt.										

Übersicht R-Befehle

Statistik-Grafiken

Boxplots

```
boxplot(x,y,...,type=5,col=c("red","yellow",...), weiter Farb- und Beschriftungsparameter);
```

x,y,...: Liste von Vektoren, für die ein Boxplot in einer Grafik gezeichnet werden soll.

type: 1-7 wie bei Quantilen (Type=1 oder Type= 5 verwenden!)

col usw.: Farb und Beschriftungsparameter der Grafik

Balkendiagramme

```
barplot(y,col=rainbow(10),main="Balkendiagramm", weitere Farb- und  
Beschriftungsparameter );
```

y: Vektor von Häufigkeiten, für den ein Balkendiagramm erstellt werden soll.

Tortendiagramme

```
pie(x, labels=names, col=farbe, weitere Farb- und Beschriftungsparameter)
```

x : Vektor von relativen Häufigkeiten, für den ein Tortendiagramm erstellt werden soll

labels: name der einzelnen Tortenstückchen

col usw.: Farb und Beschriftungsparameter der Grafik

Histogramme

```
hist(x, breaks = c(...), [nclass = NULL, include.lowest = TRUE,]  
freq = NULL, right = TRUE, . Farb- und Beschriftungsparameter)
```

x: a vector of values for which the histogram is desired.

breaks: a vector giving the breakpoints between histogram cells,

nclass: Wenn breaks nicht vorhanden: Anzahl der Klassen

include.lowest: logical; if 'TRUE', an 'x[i]' equal to the 'breaks' value will be included in the first (or last, for 'right = FALSE') bar. This will be ignored (with a warning) unless 'breaks' is a vector.

freq: logical; if 'TRUE', the histogram graphic is a representation of frequencies, the 'counts' component of the result; if 'FALSE', probability densities, component 'density', are plotted (so that the histogram has a total area of one).

right: logical; if 'TRUE', the histograms cells are right-closed (left open) intervals.

Defaults: Freq = TRUE, ohne breaks und nclass: Eigene Klasseneinteilung, äquidistant.
Am besten: breaks verwenden! (nclass und include.lowest sind nicht ganz klar)

Erstellen mehrerer Grafiken in einem Bild mit dem par()-Parameter

Mittels par() –Befehl können die Parameter von Grafik-Befehlen gesetzt und abgefragt werden. Für weitere Informationen dazu: siehe help(par).

Man kann aber auch mit Hilfe von par() mehrere Grafiken in ein Bild zeichnen:

```
par(mfrow=c(n,m));
```

Das Gesamtbild wird in eine Matrix mit n Zeilen und m Spalten aufgeteilt. In jedes Kästchen Wird eine Grafik gezeichnet. Rückgängig: par(mfrow=c(1,1)).

Übersicht R-Befehle

Parameter für Farben und Beschriftungen der Grafik

Einbeziehung von Farben und Schraffuren, sowie die Achsenbeschriftungen, Über- und Unterschriften, Legenden, Grafik horizontal drehen, satt vertikal usw..

density: the density of shading lines, in lines per inch. The default value of 'NULL' means that no shading lines are drawn. Non-positive values of 'density' also inhibit the drawing of shading lines.

angle: the slope of shading lines, given as an angle in degrees (counter-clockwise).

col: a colour to be used to fill the bars. The default of 'NULL' yields unfilled bars.

border: the color of the border around the bars. The default is to use the standard foreground color.

main, sub, xlab, ylab: these arguments to 'title' have useful defaults here.

font.main, font.sub: Schriftsatz und **col.main, col.sub:** Schriftfarbe

xlim, ylim: the range of x and y values with sensible defaults. Note that 'xlim' is not used to define the histogram (breaks), but only for plotting (when 'plot = TRUE').

axes: logical. If 'TRUE' (default), axes are drawn if the plot is drawn.

plot: logical. If 'TRUE' (default), a histogram is plotted, otherwise a list of breaks and counts is returned.

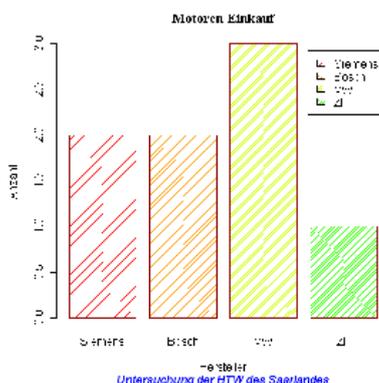
labels: logical or character. Additionally draw labels on top of bars, if not 'FALSE'; see 'plot.histogram'.

...: further graphical parameters to 'title' and 'axis'.

horiz: logical. Für Balkendiagramme. horiz=TRUE: Balken liegen horizontal. Default: FALSE.

Beispiel: `y<-c(2 2 3 1);`

```
barplot(y,names.arg=c("Siemens","Bosch","VW","ZF"),col=rainbow(10),
density=c(10,20,30,40),ylab="Anzahl",xlab="Hersteller",main="Motoren-Einkauf",
font.main=7,sub="Untersuchung der HTW des Saarlandes",col.sub="blue",font.sub=4,
border="dark red",legend=c("Siemens","Bosch","VW","ZF"))
```



Übersicht R-Befehle

Legenden mittels legend()-Befehl

Alle Far b- und Beschriftungsparameter kann man anschließend an den Grafik-Befehl auch mit einem eigenständigen legend-Befehl definieren:

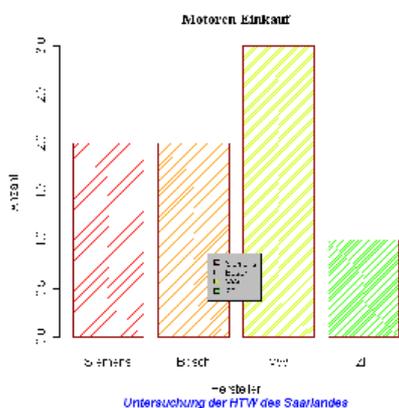
```
legend(x,y,Farb-und Beschriftungsparameter)  
legend(locator(1), Farb-und Beschriftungsparameter)
```

x,y: Position der Legende

locator(1): R wartet, bis der Cursor in der Grafik an die Stelle platziert wurde

Beispiel:

```
legend(locator(1),cex=0.7,bg="grey",legend= c("Siemens","Bosch","VW","ZF"),  
fill=rainbow(10), density=c(10,20,30,40));
```



cex=0.7: Verkleinerung der Schrift der Legende

locator(1): R wartet, bis der Cursor in der Grafik an die Stelle platziert wurde, wo die Legende hin soll (linke obere Ecke).

Darstellung von Funktionen im R³

Definition der Funktion:

```
x<-seq(0,10,by=0.5) #x von 0 bis 10, Schrittweite 0.5
```

```
y<-x # y wie x
```

```
f<-function(x,y){(x^0.5)*(y^0.5)}
```

```
z<-outer(x,y,f) # z ist die Tabelle der Funktionswerte (Matrix)
```

```
persp(x, y, z, theta = 30, phi = 10, expand = 0.5, col = "lightgreen") # Zeichnen der Fkt.
```

theta: Drehung der x-Achse in Uhrzeigerrichtung , **phi:** Kippung nach vorne

