

# Elementare Befehle

## Kommandos

<code>ls()</code> oder <code>objects()</code>	Auflistung der Objekte im Workspace
<code>rm(object)</code>	<code>object</code> wird aus dem Workspace entfernt
<code>help(funktion)</code> oder <code>?funktion</code>	Zeigt die Hilfeseite einer Funktion
<code>example(funktion)</code>	Zeigt Beispieleaufrufe einer Funktion

## Variablennamen

Variablen werden aus Buchstaben (Groß- und Kleinschreibung beachten!), Ziffern und dem Punkt „.“ gebildet. Sie dürfen nicht mit Ziffern beginnen und sollten nicht mit dem Punkt beginnen.

## Zuweisungen

<code>&lt;-</code>	Übergabe eines Werts an eine Variable
<code>-&gt;</code>	Übergabe eines Werts an eine Variable nach rechts

## Operatoren

### Arithmetische Operationen

<code>+</code>	Addition
<code>-</code>	Subtraktion
<code>*</code>	Multiplikation
<code>/</code>	Division
<code>^</code>	Potenzierung
<code>%%</code>	Division mit Rest
<code>%%</code>	Rest einer Division

### Logische Operationen

<code>==</code>	gleich
<code>!=</code>	ungleich
<code>&lt;</code>	kleiner
<code>&gt;</code>	größer
<code>&lt;=</code>	kleiner gleich
<code>&gt;=</code>	größer gleich
<code>&amp;</code>	logisches UND
<code> </code>	logisches ODER
<code>!</code>	logisches NICHT

## Vektoren und Datenstrukturen

### Generierung von Vektoren

<code>numeric(10)</code>	ein Vektor mit 10 Nullen
<code>character(10)</code>	10 × "
<code>logical(10)</code>	10 × FALSE
<code>seq(0,1,0.1)</code>	generiert den Vektor 0 0.1 ... 0.9 1
<code>1:10</code>	wie <code>seq(1,10,1)</code>
<code>c(1,2,3)</code>	generiert den Vektor 1 2 3
<code>rep(c(2,7),2)</code>	erzeugt den Vektor 2 7 2 7

## Umwandlung

<code>as.character(x)</code>	wandelt in Zeichenketten-Variable um
<code>as.numeric(x)</code>	wandelt in numerische Variable um
<code>factor(x)</code>	konvertiert in einen nominalskalierten Vektor

## Datentabellen

<code>data.frame(x,y)</code>	erzeugt Datentabelle mit den Spalten x und y
<code>data.frame(age=x, height=y)</code>	erzeugt Datensatz mit zwei benannten Vektoren x und y
<code>attach(data)</code>	data wird in den Suchpfad kopiert
<code>detach(data)</code>	data wird wieder aus dem Suchpfad gelöscht

## Einlesen von Datensätzen

<code>read.table("Pfadname")</code>	liest externen Datensatz ein
-------------------------------------	------------------------------

## Gebräuchliche Argumente der `read.table`-Funktion

<code>header=TRUE</code>	die erste Zeile wird als Zeile der Variablenamen aufgefasst
<code>sep=","</code>	Daten werden durch Kommata getrennt
<code>dec=","</code>	Komma ist Dezimalkomma (üblich: Punkt)
<code>na.strings="."</code>	Punkte für fehlende Einträge

## Varianten von `read.table`

<code>read.csv("Pfadname")</code>	liest durch Kommata getrennte Spalten
<code>read.csv2("Pfadname")</code>	liest durch Semikola getrennte Spalten, Komma als Dezimalkomma
<code>read.delim("Pfadname")</code>	liest Tab-getrennte Spalten
<code>read.delim2("Pfadname")</code>	liest Tab-getrennte Spalten, Komma als Dezimalkomma

## Indizierung und Komponentenansteuerung

### Vektoren

<code>x[i]</code>	gibt die i-te Komponente des Vektors x aus
<code>x[1:5]</code>	gibt die ersten 5 Komponenten von x aus
<code>x[c(2,3,5)]</code>	gibt die 2., 3. und 5. Komponente des Vektors x aus
<code>x[y&lt;=30]</code>	gibt den Vektor derjenigen Komponenten $x_i$ aus, für die $y_i \leq 30$ ist
<code>which[y&lt;=30]</code>	gibt die Positionen derjenigen Komponenten $y_i$ aus, für die $y_i \leq 30$ ist
<code>x[sex=="male"]</code>	Selektion anhand eines qualitativen Merkmals

## Matrizen und Datentabellen

<code>dfr\$body</code>	gibt den Vektor <code>body</code> der Datentabelle <code>dfr</code> aus
<code>dfr[dfr\$body&lt;=70]</code>	gibt alle Zeilen der Datentabelle <code>dfr</code> aus, die in der Spalte <code>body</code> einen Wert $\leq 70$ haben
<code>subset(dfr, body&lt;=70)</code>	wie oben, in vielen Situationen einfacher
<code>m[4,]</code>	gibt die 4. Zeile der Matrix/Datentabelle <code>m</code> aus
<code>m[,3]</code>	gibt die 3. Spalte von <code>m</code> aus

# Numerische und logische Funktionen

## Elementare Funktionen

<code>abs(x)</code>	Betrag von x
<code>choose(n,k)</code>	Binomialkoeffizient $\binom{n}{k}$
<code>factorial(n)</code>	Fakultät von n
<code>log(x)</code>	natürlicher Logarithmus von x
<code>log10(x)</code>	10er Logarithmus von x
<code>exp(x)</code>	$e^x$
<code>sin(x)</code>	Sinus von x
<code>cos(x)</code>	Kosinus von x
<code>tan(x)</code>	Tangens von x
<code>asin(x)</code>	Arkussinus von x
<code>acos(x)</code>	Arkuskosinus von x
<code>atan(x)</code>	Arkustangens von x

## Funktionen für Vektoren

<code>sort(x)</code>	Aufsteigende Sortierung des Vektors x
<code>length(x)</code>	Anzahl der Komponenten des Vektors x
<code>any(x,cond)</code>	Gibt TRUE zurück, wenn eine Komponente die Bedingung cond erfüllt
<code>all(x,cond)</code>	Gibt TRUE zurück, wenn alle Komponenten die Bedingung cond erfüllen
<code>min(x)</code>	Minimum der Komponenten des Vektors x
<code>min(x<sub>1</sub>,...,x<sub>n</sub>)</code>	Min. der Komponenten der Vektoren x <sub>1</sub> ,...,x <sub>n</sub>
<code>max(x)</code>	Maximum der Komponenten des Vektors x
<code>range(x)</code>	wie <code>c(min(x),max(x))</code>
<code>pmin(x<sub>1</sub>,...,x<sub>n</sub>)</code>	wie <code>c(min(c(x<sub>1</sub>[1],...,x<sub>n</sub>[1])),...)</code>
<code>pmax(x)</code>	analog zu pmin
<code>sum(x)</code>	Summe der Komponenten des Vektors x
<code>prod(x)</code>	Produkt der Komponenten des Vektors x
<code>cumsum(x)</code>	Vektor der kumulierten Summen der Komponenten von x
<code>cumprod(x)</code>	Vektor der kumulierten Produkte der Komponenten von x

## Funktionen für Matrizen

<code>t(m)</code>	Transponiert die Matrix m
<code>m %*% n</code>	Matrizenmultiplikation von m und n
<code>det(m)</code>	Determinante von m
<code>colSums(m)</code>	Spaltensummen von m
<code>rowSums(m)</code>	Zeilensummen von m
<code>colMeans(m)</code>	Mittelwerte der Spalten von m
<code>rowMeans(m)</code>	Mittelwerte der Zeilen von m

## Funktionen aus der Statistik

<code>mean(x)</code>	Durchschnitt von $x: \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
<code>var(x)</code>	Stichprobenvarianz von $x: \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
<code>sd(x)</code>	Standardabweichung, wie <code>sqrt(var(x))</code>
<code>cor(x,y)</code>	Korrelationskoeffizientenmatrix von $x$ und $y$
<code>quantile(x,p)</code>	$p$ -Quantil von $x$
<code>median(x)</code>	Median von $x$ , wie <code>quantile(x,0.5)</code>
<code>summary(x)</code>	Zusammenstellung wichtiger Statistiken von $x$
<code>cut(x,breaks)</code>	teilt den Wertebereich von $x$ in Intervalle mit den in <code>breaks</code> angegebenen Intervallgrenzen und kodiert die Werte von $x$ entsprechend der Klasse, in die sie fallen
<code>table(x)</code>	Kontingenztabelle von $x$
<code>prop.table(x,margin)</code>	bedingte Häufigkeiten gegeben die Zeilen, falls <code>margin=1</code> bedingen unter Spalten mit <code>margin=2</code>
<code>order(x)</code>	Ordnungsstatistik angewandt auf $x$
<code>rank(x)</code>	gibt die Rangwerte des numerischen Vektors $x$ wieder; bei Bindungen oder Ties werden mittlere Ränge berechnet
<code>lm(y~x)</code>	bestimmt Regressionsparameter der lin. Regression der abhängigen Variablen $y$ und der unabh. $x$

## Wahrscheinlichkeitsverteilungen

### Normalverteilungen

<code>dnorm(x)</code>	Dichte der Standardnormalverteilung in $x$
<code>pnorm(x)</code>	Verteilungsfunktion der Standardnormalverteilung in $x$
<code>qnorm(p)</code>	$p$ -Quantil der Standardnormalverteilung, d. h., $qnorm(p)$ ist das $x$ mit $\mathcal{N}(0, 1)((-\infty, x]) = p$
<code>rnorm(n)</code>	erzeugt $n$ (Pseudo-) Zufallszahlen gemäß der $\mathcal{N}(0, 1)$ -Vtlg.

### Andere Verteilungen

<code>pbinom(x,n,p)</code>	Binomialverteilung mit Parametern $n$ und $p$
<code>pgeom(x,p)</code>	Geometrische-Verteilung mit Parameter $p$
<code>phyper(x,m,n,k)</code>	Hypergeometrische-Verteilung
<code>pnbinom(x,size,prob)</code>	negative Binomialverteilung
<code>ppois(x,lambda)</code>	Poisson( $\lambda$ )-Verteilung
<code>punif(x,a,b)</code>	Gleichverteilung auf $(a, b)$
<code>pexp(x,rate)</code>	Exponentialverteilung
<code>pbeta(x,a,b)</code>	$\beta(a, b)$ -Verteilung
<code>pchisq(x,df)</code>	$\chi^2$ -Verteilung
<code>pf(x,df1,df2)</code>	$F$ -Verteilung
<code>pgamma(x,a,b)</code>	$\Gamma(a, b)$ -Verteilung
<code>pweibull(x,a,b)</code>	Weibull-Verteilung mit Parametern $a$ und $b$
<code>pcauchy(x,a,b)</code>	Cauchy-Verteilung mit Parametern $a$ und $b$
<code>plnorm(x,mean,sd)</code>	Lognormalverteilung
<code>plogis(x,location,scale)</code>	Logistische Verteilung
<code>pnorm(x,mean,sd)</code>	Normalverteilung
<code>psignrank(x,n)</code>	Verteilung der Wilcoxon'schen Vorzeichen-Statistik
<code>pt(x,df)</code>	Student'sche $t$ -Vtlg. ( $df$ für <i>degrees of freedom</i> )
<code>pwilcox(x,m,n)</code>	Vtlg. der Wilcoxon'schen Rangsummenstatistik für Samples der Größe $m$ und $n$

Ersetzt man das **p** in den Befehlen durch **d** oder **q**, so erhält man die Dichte oder ein Quantil der entsprechenden Verteilung. Ersetzt man das **p** durch ein **r**, so erhält man eine Randomisierung (auch: *Sample*) gemäß der jeweiligen Verteilung.

## Tests

### Testen in R

<b>binom.test</b>	Exakter Binomialtest
<b>chisq.test</b>	Chiquaret-Anpassungstest
<b>ks.test</b>	Kolmogoroff-Smirnoff-Test
<b>shapiro.test</b>	Shapiro-Wilk-Normalverteilungstest
<b>t.test</b>	Student'scher <i>t</i> -Test und Welch'scher <i>t</i> -Test
<b>wilcox.test</b>	Wilcoxon-Vorzeichen-Rangtest und Wilcoxon-Rangsummentest

## Graphiken

### Standard-Plots

<b>barplot()</b>	Säulendiagramm
<b>boxplot()</b>	Boxplot
<b>curve(f(x),a,b)</b>	zeichnet den Graphen der (stetigen) Funktion $f$ auf $[a,b]$
<b>hist()</b>	Histogramm
<b>interaction.plot()</b>	Profildiagramm
<b>mosaicplot()</b>	Mosaikplot
<b>plot()</b>	Standardplot
<b>plot(ecdf(x))</b>	Empirische Verteilungsfunktion des Vektors $x$
<b>pie()</b>	Kreisdiagramm
<b>qqplot()</b>	QQ-Plot
<b>qqnorm()</b>	NQ-Plot

### Zusätzliche Plot-Elemente

<b>abline(a,b)</b>	fügt Gerade mit Achsenabschnitt <b>a</b> und Steigung <b>b</b> in bestehenden Plot ein
<b>abline(h=u)</b>	wie oben; jedoch Horizontale in <b>u</b>
<b>abline(v=u)</b>	wie oben; jedoch Vertikale in <b>u</b>
<b>box()</b>	erstellt einen Rahmen um die Graphik
<b>legend()</b>	Legende
<b>lines()</b>	zusätzliche Linienelemente für einen Plot
<b>points()</b>	zusätzliche Punkte für einen Plot
<b>title()</b>	erstellt Titel (und Untertitel) für den letzten Plot

### Graphikparameter

<b>expression</b>	mit <b>xlab</b> und <b>ylab</b> , erlaubt das Einsetzen von mathematischen Termen (z. B. mit Index)
<b>lty, lwd</b>	Linientyp bzw. Linienstärke
<b>mfrow, mfcoll</b>	mehrere Graphiken in einem Bild ( <b>multiframe</b> )
<b>pch</b>	Einstellung des <b>plotting characters</b>
<b>xlim=c(0,9)</b>	die Graphik zeigt den x-Achsenabschnitt von 0 bis 9
<b>ylim=c(0,9)</b>	wie <b>xlim</b> für die y-Achse

## Speichern von Graphiken

<code>pdf("Pfadname")</code>	Speichert die Graphik des folgenden <code>plot</code> -Befehls in eine <code>pdf</code> Datei
<code>jpeg("Pfadname")</code>	Speichert die Graphik des folgenden <code>plot</code> -Befehls in eine <code>jpg</code> Datei
<code>png("Pfadname")</code>	Speichert die Graphik des folgenden <code>plot</code> -Befehls in eine <code>png</code> Datei
<code>bmp("Pfadname")</code>	Speichert die Graphik des folgenden <code>plot</code> -Befehls in eine <code>bmp</code> Datei
<code>tiff("Pfadname")</code>	Speichert die Graphik des folgenden <code>plot</code> -Befehls in eine <code>tiff</code> Datei

Im Anschluss des `plot`-Befehls muss `dev.off()` aufgerufen werden.

Beispiel: `pdf("Pfadname"); plot(ecdf(rnorm(15))); dev.off();`

## Programmierung

<code>if(cond){commands}</code>	Ausführung der Befehle <code>commands</code> , wenn der Wert der Formel <code>cond</code> TRUE ist
<code>else{commands}</code>	alternative Arbeitsanweisungen (im Anschluss an einen <code>if</code> -Befehl)
<code>for(i in x){commands}</code>	Schleife über die Werte des Vektors <code>x</code>
<code>while(cond){commands}</code>	Wiederholung der Befehle <code>commands</code> solange der Wert der Formel <code>cond</code> TRUE ist
<code>repeat {commands}</code>	(erstmal) unbegrenzte Wiederholung der Befehle <code>commands</code>
<code>if(cond) break</code>	wenn der Wert der Formel <code>cond</code> TRUE ist, wird die aktuelle <code>repeat</code> -Schleife abgebrochen