

## Elementare Befehle

### Kommandos

<code>ls()</code> or <code>objects()</code>	Auflistung der Objekte im Workspace
<code>rm(object)</code>	<i>object</i> wird aus dem Workspace entfernt

### 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>	$10 \times "$
<code>logical(10)</code>	$10 \times \text{FALSE}$
<code>seq(0,1,0.1)</code>	generiert den Vektor <code>0 0.1 ... 0.9 1</code>
<code>1:10</code>	wie <code>seq(1,10,1)</code>
<code>c(1,2,3)</code>	generiert den Vektor <code>1 2 3</code>
<code>rep(c(2,7),2)</code>	erzeugt den Vektor <code>2 7 2 7</code>

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

## Numerische Funktionen

### Elementare Funktionen

<code>abs(x)</code>	Betrag von $x$
<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 $x$
<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 $x_1, \dots, x_n$
<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 <code>pmin</code>
<code>length(x)</code>	Anzahl der Komponenten des Vektors $x$
<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 kumul. Summen der Komponenten von $x$
<code>cumprod(x)</code>	Vektor der kumul. Produkte der Komp. des Vektors $x$

## Funktionen aus der Statistik

<code>mean(x)</code>	Durchschnitt von $x$ : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
<code>sd(x)</code>	Standardabweichung, wie <code>sqrt(var(x))</code>
<code>var(x)</code>	Varianz von $x$ : $\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
<code>cor(x,y)</code>	Korrelationskoeffizientenmatrix von $x$ und $y$
<code>median(x)</code>	Median von $x$ , wie <code>quantile(x,0.5)</code>
<code>quantile(x,p)</code>	$p$ -Quantil von $x$
<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,k)</code>	bedingte Häufigkeiten gegeben die $k$ te Spalte
<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. $x$

## Indizierung und Komponentensteuerung

### 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>x[sex="male"]</code>	Selektion anhand eines qualitativen Merkmals

### Matrizen und Datentabellen

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

## 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., <code>qnorm(p)</code> 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>pbeta(x, a, b)</code>	$\beta(a, b)$ -Verteilung
<code>pbinom(x, n, p)</code>	Binomialvtlg. mit Parametern $n$ und $p$
<code>pchisq(x, df)</code>	$\chi^2$ -Verteilung
<code>pexp(x, rate)</code>	Exponentialvtlg.
<code>pf</code>	$F$ -Verteilung
<code>pgamma(x, a, b)</code>	$\Gamma(a, b)$ -Verteilung
<code>plnorm(x, mean, sd)</code>	Lognormalvtlg.
<code>pnorm(x, mean, sd)</code>	Normalvtlg.
<code>ppois(x, lambda)</code>	Poisson( $\lambda$ )-Vtlg.
<code>psignrank(x)</code>	Vtlg. der Wilcoxon'schen Vorzeichen-Statistik
<code>pt(x, df)</code>	Student'sche $t$ -Vtlg. ( $df$ für <i>degrees of freedom</i> )
<code>punif(x, a, b)</code>	Gleichverteilung auf $(a, b)$

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

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

## Graphiken

### Standard-Plots

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

### Zusätzliche Plot-Elemente

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

### Graphikparameter

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

## Programmierung

<code>if(cond){commands}</code>	Ausführung der Befehle <code>commands</code> , wenn der Wert der Formel <code>cond</code> <b>TRUE</b> 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 $x$
<code>while(cond){commands}</code>	Wiederholung der Befehle <code>commands</code> solange der Wert der Formel <code>cond</code> <b>TRUE</b> 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> <b>TRUE</b> ist, wird die aktuelle <code>repeat</code> -Schleife abgebrochen