

**Allgemein**

`install.packages("Paket")`: Paket installieren  
`library(Paket)`: Paket laden  
`objects()`: Inhalt des Workspaces  
`remove(objekt)`: Entfernt Objekt aus dem Workspace  
`setwd("...")`: Arbeitsverzeichnis ändern  
`load("...")`: Datei öffnen  
`save(objekt, file="...")`: Objekt speichern  
`save.image(file="...")`: Workspace speichern  
`attach(daten)`: Daten aktivieren  
`detach(daten)`: Daten deaktivieren  
`help(Funktion)`: Hilfe für bestimmte Funktion

**Datenmanagement**

`c(x, y)`: Erstellt einen Vektor  
`factor(x)`: Vektor in Faktor konvertieren  
`as.numeric(faktor)`: Faktor in Vektor konvertieren  
`cbind(x, y), rbind(x, y)`: Einzelne Vektoren werden als Spalten (`cbind`) bzw. als Zeilen in einer Matrix zusammengefasst  
`data.frame(x, y)`: Erstellt einen Data Frame  
`merge(data.frame.1, data.frame.2, by="x")`: Variablen aus zwei Data Frames zusammenfügen  
`na.omit(daten)`: Fälle mit fehlenden Werten entfernen  
`reshape(...)`: Transformiert die Daten vom Wide- ins Long-Format oder umgekehrt  
`fix(daten)`: Öffnet Datenmatrix im Dateneditor  
`subset(daten, select=x)`: Auswahl einer bestimmten Variablen  
`subset(daten, subset=(variable==x))`: Auswahl bestimmter Personen (Filtern)  
`recode(x, ...)` [1]: Umkodieren  
`ave(...)`: Funktionen getrennt für jede Untergruppe durchführen und zu Dataframe hinzufügen

**Objekteigenschaften**

`class(x)`: Art des Objekts  
`names(x)`: Namen der Komponenten des Objekts  
`length(x)`: Anzahl der Elemente eines Objekts  
`nrow(x)`: Anzahl Zeilen eines Objekts

**Tabellen**

`table(x)`: Häufigkeitstabelle  
`table(x, y)`: Kontingenztabelle  
`prop.table(tabelle)`: Relative Häufigkeiten

**Deskriptive Statistiken**

`describe(x)` [2]: Deskriptive Statistiken  
`describeBy(x)` [2]: Deskriptive Statistiken getrennt für bestimmte Gruppen  
`median(x)`: Median  
`mean(x)`: Mittelwert  
`range(x)`: Wertebereich  
`sd(x)`: Standardabweichung  
`norm(x)` [3]: Schiefe und Exzess  
`sum(x)`: Summe  
`tapply(av, uv, funktion)`: Wendet eine Funktion auf eine Variable `av` getrennt für die Ausprägungen der `uv` an  
`na.rm=TRUE`: Argument bei fehlenden Werten

**Zusammenhangsmaße**

`cor(x, y)`: Korrelationsmatrix  
`cor.test(x, y)`: Signifikanztest für Korrelation  
`corr.test(daten)` [2]: Korrelation,  $N$  und  $p$ -Werte  
`partial.r()` [2]: Partialkorrelation

**Graphiken**

`plot(x, y)`: Flexible Graphikfunktion für Streudia-

gramme, Boxplots etc.

`barplot(table(x))`: Säulendiagramm  
`boxplot(x)`: Einfacher Boxplot  
`boxplot(y ~ x)`: Gruppierte Boxplots  
`hist(x)`: Histogramm  
`plotmeans(y ~ x)` [4]: Fehlerbalkendiagramm  
`error.bars.by(y, x)` [2]: Fehlerbalkendiagramm  
`qqPlot(x)` [1]: Q-Q-Plot mit Konfidenzintervallen  
`curve(funktion(x), min, max)`: Funktionskurve  
`abline(funktion)`: Gerade hinzufügen  
`legend(position, namen, kennzeichen)`: Legende hinzufügen  
`text(position, "Text")`: Text hinzufügen  
`locator()`: Koordinaten bestimmen  
`identify(x, y)`: Identifizieren bestimmter Werte  
`par(...)`: Allgemeine Graphik-Parameter  
`par(mfrow=c(a, b))`: Anzahl der Diagramme pro Zeile ( $a$ ) und Spalte ( $b$ )  
`jpeg(...)`, `pdf(...)`: Export von Graphiken

**t-Tests**

`t.test(av, mu=x)`:  $t$ -Test für eine Stichprobe  
`t.test(av ~ uv)`:  $t$ -Test für unabhängige Stichproben  
`t.test(av1, av2, paired=TRUE)`:  $t$ -Test für abhängige Stichproben

**Varianzanalyse**

`aov(av ~ faktor)`: Einfaktorielle ANOVA  
`aov(av ~ faktor1 * faktor2)`: Zweifaktorielle ANOVA  
`Anova(modell, type = 3)` [1]: Typ-III-Quadratsummen für eine mehrfaktorielle Varianzanalyse  
`interaction.plot(faktor1, faktor2, av)`: Interaktionsdiagramm

`pairwise.t.test(av, faktor)`: Multiple Paarvergleiche

`TukeyHSD(modell)`: Tukey's HSD für ein Varianzanalyse-Modell

`ezANOVA(daten, dv, sid, within, between)`  
[5]: ANOVA mit Messwiederholung bzw. gemischte ANOVA

`ezStats(daten, dv, sid, within, between)`  
[5]: Deskriptive Statistiken für ANOVA mit Messwiederholung

### Regressionsanalyse

`lm(y ~ x)`: Einfache lineare Regression

`lm(y ~ x1 + x2)`: Multiple Regression

`lm(y ~ x1 * x2)`: Moderierte Regression

`lm(y ~ x + I(x^2))`: Regression mit quadratischem Zusammenhang

`anova(modell1, modell2)`: F-Test für den Vergleich zweier Regressionsmodelle

`contrasts(faktor)`: Zeigt die aktuelle Kodierung eines Faktors an

`plotSlopes()`: Diagramm der bedingten Regressionsgeraden in einer moderierten Regression

`testSlopes()`: Test der bedingten Regressionsgeraden in einer moderierten Regression

`glm(y ~ x, family=binomial)`: Logistische Regression

### Weitere Tests

`leveneTest(av ~ uv)` [1]: Levene-Test

`chisq.test(tabelle)`:  $\chi^2$ -Test

`wilcox.test(x, mu = y)`: Wilcoxon-Test

`kruskal.test(y ~ x)`: Kruskal-Wallis-Test

### Poweranalysen

`pwr.anova.test(...)` [6]: Varianzanalyse

`pwr.f2.test(...)` [6]:  $R^2$

`pwr.r.test(...)` [6]: Korrelation

`pwr.t.test(...)` [6]: t-Test

### Konfidenzintervalle für Effektgrößen

`ci.sm(...)` [7]: Standardisierter Mittelwert

`ci.smd(...)` [7]: Standardisierte Mittelwertsdifferenz (Cohens  $d$ )

`ci.pvaf(...)` [7]: Partielles  $\eta^2$

### Testkonstruktion

`alpha(daten)` [2]: Itemanalyse

`fa.parallel(daten)` [2]: Parallelanalyse

`fa(daten, faktoren, methode, rotation)` [2]: Faktorenanalyse

### Lineare Strukturgleichungsmodelle

'...' [8]: Spezifikation der Modellgleichungen

`sem(...)` [8]: Modell schätzen

`cfa(...)` [8]: konfirmatorische Faktorenanalyse

`parameterEstimates(...)` [8]: Modellparameter in Tabellenform, auch mit Konfidenzintervallen

`fitMeasures(...)` [8]: Modellgütemaße

`modindices(...)`: [8] Modification-Indices

`semPaths(...)` [9]: Pfaddiagramm erstellen

### Mehrebenenanalyse

`lmer(y ~ Intercept + (x|Gruppe))` [10]: Spezifikation des Modells

`VarCorr(...)` [10]: Varianzen, SDs & Korrelationen der Zufallseffekte

`dotplot(ranef(...))` [11]: Dotplot gruppenspezifischer Parameter

### Ergebnisdarstellung

`round(objekt, x)`: Rundet alle Werte eines Objekts auf  $x$  Nachkommastellen

`sort(x)`: Gibt die Elemente eines Objekts in aufsteigender Reihenfolge wieder

`summary(objekt)`: Fasst den Inhalt eines Objekts angemessen zusammen

`write.csv2(tabelle)`: Speichert eine Tabelle als csv-Datei

### Sonstige Funktionen

`<-`: Zuweisungspfeil, erstellt ein neues Objekt

`objekt$komponente`: Wählt eine Komponente aus einem Objekt aus

`object[a,b]`: Wählt bestimmte Werte aus einem Objekt aus

`scale(x)`: Variable standardisieren

`scale(x, scale=FALSE)`: Variable zentrieren

### Zusätzliche Pakete für die hier aufgeführten Funktionen

[1] car

[2] psych

[3] QuantPsyc

[4] gplots

[5] ez

[6] pwr

[7] MBESS

[8] lavaan

[9] semPlot

[10] lme4

[11] lattice