

Zentrale Funktionen im Überblick

Allgemein

`install.packages("Paket")`: Paket installieren
`library(Paket)`: Paket laden
`objects()`: Inhalt des Workspaces
`remove(objekt)`: Entfernt ein Objekt aus dem Workspace
`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)`: Einzelne Vektoren werden als Spalten in einer Matrix zusammengefasst
`rbind(x, y)`: Einzelne Vektoren werden 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)`: Löscht alle Fälle mit fehlenden Werten
`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, ...) [11]`: Umkodieren

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
`str(x)`: Struktur des Objekts

Tabellen

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

Deskriptive Statistiken

`describe(x) [7]`: Umfangreiche deskriptive Statistiken
`min(x)`: Kleinster Wert in einem Objekt
`max(x)`: Größter Wert in einem Objekt
`which.max(tabelle)`: Modalwert
`median(x)`: Median
`mean(x)`: Mittelwert
`range(x)`: Wertebereich
`quantile(x)`: 5-Punkte-Zusammenfassung
`sd(x)`: Standardabweichung
`var(x)`: Varianz
`norm(x) [8]`: Schiefe und Exzess
`sum(x)`: Summe
`sum(is.na(x))`: Anzahl fehlender Werte
`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

`rcorr(x, y) [5]`: Korrelation, *N* und *p*-Werte
`partial.cor(matrix) [9]`: Partialkorrelation
`polycor(x, y) [6]`: Polychorische bzw. tetrachorische Korrelation
`polyserial(x, y) [6]`: Polyseriale bzw. biserial Korrelation
`use="complete"`: Argument bei fehlenden Werten

Graphiken

`plot(x, y)`: Flexible Graphikfunktion für Streudiagramme, Boxplots etc.
`barplot(table(x))`: Säulendiagramm
`boxplot(x)`: Einfacher Boxplot
`boxplot(y ~ x)`: Gruppierte Boxplots
`hist(x)`: Histogramm
`plotmeans(y ~ x) [4]`: Fehlerbalkendiagramm
`qq.plot(x) [11]`: Q-Q-Plot mit Konfidenzintervallen
`stem(x)`: Stamm-Blatt-Diagramm
`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)

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

Zentrale Funktionen im Überblick

Varianzanalyse

`aov(av ~ faktor)`: Einfaktorielle ANOVA

`aov(av ~ faktor1 * faktor2)`: Zweifaktorielle ANOVA

`interaction.plot(faktor1, faktor2, av)`: Interaktionsdiagramm

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

`ezANOVA(daten, dv, sid, within, between)` [3]: ANOVA mit Messwiederholung bzw. gemischte ANOVA

`ezPlot(daten, dv, sid, within, between)` [3]: Interaktionsdiagramm für ANOVA mit Messwiederholung

`ezStats(daten, dv, sid, within, between)` [3]: 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

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

Weitere Tests

`levene.test(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(...)` [11]: Varianzanalyse

`pwr.f2.test(...)` [11]: *R*²

`pwr.r.test(...)` [11]: Korrelation

`pwr.t.test(...)` [11]: *t*-Test

Konfidenzintervalle für Effektgrößen

`ci.sm(...)` [10]: Standardisierter Mittelwert

`ci.smd(...)` [10]: Standardisierte Mittelwertsdifferenz (Cohens *d*)

`ci.pvaf(...)` [10]: Partielles *eta*²

`ci.R2(...)` [10]: *R*²

Testkonstruktion

`reliability(daten)` [2]: Itemanalyse

`fa.parallel(daten)` [7]: Parallelanalyse

`fa(daten, faktoren, methode, rotation)` [7]: Faktorenanalyse

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

Sonstige Funktionen

`<-`: Zuweisungspfeil, erstellt ein neues Objekt

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

`scale(x)`: Variable standardisieren

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

Zusätzliche Pakete für die hier aufgeführten Funktionen

[1] `car`

[2] `CTT`

[3] `ez`

[4] `gplots`

[5] `Hmisc`

[6] `polycor`

[7] `psych`

[8] `QuantPsyc`

[9] `Rcmdr`

[10] `MBESS`

[11] `pwr`

Hinweis: Details zur Anwendung der einzelnen Funktionen und Paketen erhalten Sie über die `help`-Funktion.

Meine Funktionen und Notizen