

# DDM – Statistik

## R Cheat Sheet

### Hilfe aufrufen

#### ?funktionsname

Hilfe für eine bestimmte Funktion aufrufen.

#### help.search("Suchbegriffe")

Die Hilfe nach einem oder mehreren Begriffen durchsuchen.

#### help(package = "Bibliotheksname")

Hilfe zu einer Bibliothek finden.

### Bibliotheken verwenden

#### install.packages("Bibliotheksname")

Ein Paket herunterladen und installieren.

#### library("Bibliotheksname")

Bibliothek laden und die Funktionen verfügbar machen.

### Arbeitsverzeichnis

#### getwd()

Den Pfad zum aktuellen Arbeitsverzeichnis ausfindig machen.

#### setwd("Ordnerpfad")

Festlegen des Arbeitsverzeichnisses.

#### setwd(choose.dir())

Festlegen des Arbeitsverzeichnisses. Zur Wahl des Ordnerpfades ein Fenster öffnen.

### Vektoren

#### Vektoren erstellen

##### meinvektor <- c(1, 3, 5)

Werte kombinieren und der Variablen mit **meinvektor** zuweisen

##### names(meinvektor) <- c("A", "B")

Vektorelemente benennen

#### Vektorfunktionen

##### sort(vektor)

Vektor sortieren.

##### rev(vektor)

Reihenfolge umkehren.

##### table(vektor)

Häufigkeiten der enthaltenen Werte ermitteln

##### unique(vektor)

Unterschiedliche Werte zurückgeben

#### Vektorelemente auswählen

##### vektor[zahl]

Element an Position **zahl**

##### vektor[-zahl]

alle Elemente bis auf Index **zahl**

##### vektor[2:5]

Elemente 2 bis 5

##### vektor[c(1,4)]

Elemente 1 & 4

**vektor[vektor == 5]** alle Elemente, die dem Wert 5 entsprechen

##### vektor["name"]

Element mit der Benennung **name**

### R Umgebung

#### ls()

Variablen im Workspace auflisten

#### rm(list=ls())

alle Variablen im Workspace löschen

### Daten einlesen/speichern

#### read.csv2("Dateipfad")

csv-Datei mit deutscher Formatierung einlesen, Rückgabeformat: Dataframe

#### read.csv2(file.choose())

csv-Datei mit deutscher Formatierung einlesen, Dateipfad über Fenster auswählen, Rückgabeformat: Dataframe

#### write.csv2(daten, "Dateipfad.csv")

Daten in eine csv-Datei mit deutscher Formatierung speichern

#### saveRDS(daten, "Dateipfad.RDS")

Speichern der Daten im R-eigenen Format RDS (Kodierung bleibt erhalten)

### Faktoren

#### factor(variable)

Erzeugen eines Faktors, Levels in alphabetischer Reihenfolge

#### factor(variable, ordered=TRUE)

Erzeugen eines Faktors mit Rangfolge

#### levels(variable)

Faktor-Level abrufen

#### factor(variable, levels = c("level\_1", "level\_2"))

Faktorvariable erzeugen mit angegebenen Levels (Achtung: Nicht enthaltene Werte werden zu **NA**)

#### revalue(variable, c("alt1" = "neu1", "alt2" = "neu2"))

Faktorlevel umbenennen, ermöglicht auch Zusammenführung von Faktorleveln (erfordert Bibliothek **plyr**)

### DataFrame

#### daten\$**v1**

Zugriff auf Spalte **v1**

#### daten[zeile, spalte]

Zugriff über Zeilen-/Spaltenangabe

#### daten\$**v1** <- NULL

Entfernen von **v1** aus dem DataFrame

#### daten[daten\$**v1** == 5, ]

Alle Zeilen auswählen, für die in der Spalte **v1** der Wert 5 steht

#### daten[daten\$**v1** == 5, "**v2**"]

Werte von **v2** für Zeilen mit **v1** == 5 auswählen

#### daten\$**v1**[daten\$**v1** == 5]

Einträge in Spalte **v1** auswählen, die den Wert 5 aufweisen

### Zusammenführen

#### merge(d1, d2, by="spaltenname")

Zusammenführen der DataFrames **d1** und **d2** anhand des Schlüssels in Spalte „spaltenname“.

#### cbind(d1, d2)

DataFrame **d1** und **d2** horizontal zusammenführen (gleiche Zeilenanzahl)

#### rbind(d1, d2)

DataFrame **d1** und **d2** vertikal zusammenführen (benötigt gleiche Spaltennamen)

### Objektinformationen

#### str(daten)

Struktur der Daten abrufen

#### summary(daten)

Zusammenfassung der Daten abrufen

## Deskriptive Statistik

### **table(variable)**

Absolute Häufigkeiten ermitteln

### **table(variable, variable, ...)**

Kontingenztafel mit absoluten Häufigkeiten erstellen

### **prop.table(variable)**

Kontingenztafel mit relativen Häufigkeiten erstellen

### **describe(variable)**

Ermitteln deskriptiver Basis Kennzahlen (z.B. Mittelwerte, Schiefe, Wölbung)

### **median(variable)**

Median ermitteln

### **quantile(variable)**

Quantile (25%, 50%, 75%) ermitteln

### **rank(variable)**

Ränge der Werte ermitteln

### **mean(variable)**

arithm. Mittelwert ermitteln

### **var(variable)**

Varianz ermitteln

### **sd(variable)**

Standardabweichung ermitteln

### **IQR(variable)**

Interquartilbereich ermitteln

### **cov(variable1, variable2)**

Kovarianz ermitteln

### **cor(variable1, variable2)**

Korrelation ermitteln

### **cohen.kappa(dataframe)**

Berechnung der Inter-Rater-Agreements

## Schließende Statistik

### **shapiro.test(variable)**

Shapiro-Wilk Test auf Normalverteilung

### **leveneTest(Zielvariable, gruppenbildende\_Variable)**

Levene-Test zur Prüfung von Varianzhomogenität (aus: Bibliothek `car`)

### **cor.test(variable1, variable2)**

Signifikanztest für Korrelation

### **lm(variable1~variable2, data=daten)**

Regressionsmodell berechnen

### **summary(regressionsmodell)**

Informationen zum Regressionsmodell abrufen (Modellparameter, Angaben zur Signifikanz, Effektgröße)

### **confint(regressionsmodell)**

Konfidenzintervalle für Regressionsmodellparameter abrufen

### **t.test(variable1~variable2, data=daten, paired=FALSE)**

t-Test für unabhängige Stichproben durchführen (**paired=TRUE** für abhängige Stichproben)

## Mathem. Funktionen

### **sum(variable)**

Summe der Elemente bilden

### **sqrt(variable)**

Wurzel ziehen

### **max(variable)**

größten Wert ermitteln

### **round(variable, nachkommastelle)**

Werte auf die angegebene Anzahl Nachkommastellen runden.

## Graphiken (Basispaket)

### **plot(daten)**

Graphiken für alle bivariaten Variablenkombinationen darstellen

### **plot(variable1, variable2)**

Numerische Variablen in einem Koordinatensystem darstellen (x,y)

### **hist(variable)**

Histogramm erstellen

### **barplot(variable)**

Balkendiagramm erstellen

### **pie(variable)**

Tortendiagramm erstellen

weitere Parameter zur Gestaltung über die Hilfe abrufbar

## Graphiken (ggplot2)

### **p <- ggplot(daten, aes(x))**

Basisgraphikobjekt mit Variable für x-Achse erzeugen und in p abspeichern

### **p + geom\_bar()**

Balkendiagramm erzeugen (x:kategorial)

### **p + geom\_hist()**

Histogramm erzeugen (x:metrisch)

### **p <- ggplot(daten, aes(x,y))**

Basisgraphikobjekt erzeugen mit x- und y-Variable

### **p + geom\_point()**

Scatterplot erzeugen

### **p + geom\_boxplot()**

Boxplot erstellen (x:kategorial, y:metrisch), auch nur mit y-Variable

Ausführliches Cheat Sheet zu ggplot2 unter:  
<https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.1.pdf>