

| Statistics 2026  |   |   |
|--|---|---|
| Befehl   | Beschreibung  | Gruppe                                    |
| <code>set.seed(n)</code>   | Legt den Startwert für den Zufallsgenerator fest, damit Ergebnisse <b>reproduzierbar</b> sind.  | Objects, Assignment, and Basic Arithmetic |
| <code>a %% b</code>  | quotient  |   |
| <code>a %b</code>  | remainder   |   |
| <code>a^b</code>   | a to the power of b   |   |
| <code>floor(x)</code>  | rundet das Ergebnis auf die nächstkleinere ganze Zahl ab.   |   |
| <code>rm()</code>  | delete b  |   |
| <code>seq(0,1,0.01)</code>   | erzeugt eine Zahlenfolge von 0 bis 1 in Schritten von 0,01.   |   |
| <code>max()</code>   | gibt den größten Wert eines numerischen Vektors oder Objekts zurück (optional unter Ignorieren von NA).   |   |
| <code>mean()</code>  | mean / Durchschnitt   |   |
| <code>sqrt()</code>  | berechnet die Quadratwurzel der übergebenen Zahl(en).   |   |
| <code>abs()</code>   | berechnet den Absolutbetrag einer Zahl bzw. eines numerischen Vektors. ( <b>Abstand zur 0</b> )   |   |
| <code>%&gt;%</code>  | Pipeline  |   |
| <code>as.Date(df\$MESS_DATUM)</code>   | wandelt die Spalte MESS_DATUM aus dem Data Frame df in ein Date-Objekt um.  |   |
| <code>sum()</code>   | berechnet die Summe aller Elemente eines numerischen Vektors oder Objekts (optional ohne NA).   |   |
| <code>count(wet)</code>  | zählt, wie oft jeder unterschiedliche Wert in der Variable wet vorkommt, und gibt die Häufigkeiten aus.   |   |
| <code>mean(x%%3==0   x%%5==0)</code>   | Berechnet den Anteil (relative Häufigkeit) der Elemente in x, die durch <b>3 oder 5 teilbar</b> sind da TRUE als 1 und FALSE als 0 gemittelt wird.                                      | Vectors and Vectorized Operations         |
| <code>mean(A &amp; C)</code>   | berechnet den Anteil der Elemente, bei denen sowohl A als auch C TRUE sind — also die gemeinsame Wahrscheinlichkeit (Schnittmenge) beider Bedingungen.                                  |   |
| <code>var()</code>   | variation   |   |
| <code>sd()</code>  | standard deviation / Standardabweichung   |   |
| <code>iseven &lt;- x %% 2 == 0</code>  | logical Vector  |   |
| <code>identical(x, y)</code>   | Prüft, ob zwei Objekte <b>vollständig identisch</b> sind (Werte, Typen, Länge etc.).  |   |
| <code>list(poly = poly, stats = stats)</code>  | list of contents  |   |
| <code>y[2:4]</code>  | Select elements 2 through 4 of a vector   |   |
| <code>y[-1]</code>   | Exclude element 1 of a vector   |   |
| <code>y + c(1, 2)</code>   | Add the vector c(1,2) to y (gets repeated to the number of digits needed for sum)   |   |
| <code>any()</code>   | prüft, ob mindestens ein Element in einem logischen Vektor TRUE ist, und gibt entsprechend TRUE oder FALSE zurück.  | Factors and Tables                        |
| <code>duplicated(v)</code>   | gibt für jedes Element in v TRUE zurück, wenn es schon früher im Vektor vorkam, also ein Duplikat ist, sonst FALSE.   |   |
| <code>sort()</code>  | ordnet die Elemente eines Vektors standardmäßig aufsteigend (optional auch absteigend).   |   |
| <code>which.max()</code>   | <b>Liefert den Index des größten Werts</b> im Vektor  |   |
| <code>median()</code>  | berechnet den Median, also den <b>mittleren Wert</b> eines numerischen Vektors.   |   |
| <code>factor(other)</code>   | converts vector in factors (same elements get the same number)  |   |
| <code>table(other)</code>  | frequency table   |   |
| <code>prop.table(other)</code>   | proportion table  |   |
| <code>matrix(1:9, nrow = 3, byrow = TRUE)</code>   | Creates a 3x3 matrix M filled by rows with the numbers 1 to 9.  | Matrices and Matrix Algebra               |
| <code>t(M)</code>  | gegenmatrix   |   |
| <code>M %*% t(M)</code>  | matrix multiplication   |   |
| <code>cbind(o1,o2)</code>  | fügt die Objekte o1 und o2 spaltenweise zu einer Matrix oder einem Data Frame zusammen.   |   |
| <code>rowSums(M)</code>  | berechnet die Summe der Werte jeder Zeile der Matrix oder des Data Frames M.  |   |
| <code>cbind()</code>   | fügt mehrere Vektoren, Matrizen oder Data Frames spaltenweise zusammen und erstellt daraus eine Matrix oder einen Data Frame.   |   |
| <code>M[, 2]</code>  | Extract the second column of matrix   |   |
| <code>M[Row, Column]</code>  | Select one entry  |   |
| <code>sweep()</code>   | wendet eine Operation (z. B. +, -, *, /) zeilen- oder spaltenweise auf ein Array/Matrix an , meist zum <b>Addieren, Subtrahieren, Multiplizieren oder Dividieren mit einem Vektor</b> . |   |
| <code>nrow()</code>  | gibt die Anzahl der Zeilen eines Data Frames oder einer Matrix zurück.  |   |
| <code>rnorm(5)</code>  | 5 normalverteilte zufällige Zahlen  | Lists and Data Frames                     |
| <code>dnorm(x, mean, sd)</code>  | Liefert die theoretische <b>Dichtefunktion</b> der Normalverteilung N(mean, sd).  |   |
| <code>rep("MS", 5)</code>  | string 5 mal in vector speichern  |   |
| <code>data.frame(x = x, initials = initials)</code>  | create a data frame (normale Tabelle) data.frame() requires columns (vectors) of the same length; matrices with different dimensions cannot be combined directly without reshaping.     |   |
| <code>DF\$x</code>   | Select elements from data frame (selects x from DF)   |   |
| <code>read.csv("path/to/your/file.csv", sep = ",")</code>  | Reading a CSV File with a Different Delimiter (standard is ",")   | Reading/Writing                           |
| <code>write.csv(df, "D:\\Bobdata.csv", row.names=FALSE)</code>   | Export a Data Frame to a CSV File   |   |
| <code>fp &lt;- tempfile(fileext = ".csv")</code>   | create a temporary file to write the contents to  |   |
| <code>selectDWD()</code>   | filtert und wählt passende DWD-Datensätze bzw. Stationen anhand von Kriterien wie Parameter, Zeitraum oder Stations-ID aus.   |   |
| <code>dataDWD(meta, read = FALSE)</code>   | Lädt die zu den Metadaten meta gehörenden DWD-Dateien herunter, ohne sie direkt einzulesen, und gibt stattdessen Dateipfade bzw. Dateiinformationen zurück.                             |   |
| <code>readDWD(file)</code>   | liest eine DWD-Datendatei aus einer lokalen Datei ein und gibt die enthaltenen Wetterdaten als Data Frame/Tibble zurück.  | Plotting                                  |
| <code>plot(DF2\$x, DF2\$y, pch = 19, main = "Scatter with LS line", xlab = "x", ylab = "y")</code>                   | create a simple plot with descriptions  |   |
| <code>hist(x, breaks, freq, main)</code>   | erstellt ein Histogramm, das die <b>Häufigkeitsverteilung</b> von numerischen Daten grafisch darstellt.   |   |
| <code>abline(lm(y ~ x, data = DF2))</code>   | fitted least-squares line   |   |
| <code>curve(expr, add = TRUE)</code>   | Zeichnet eine Funktion (z. B. Dichtekurve dnorm) in ein bestehendes Diagramm.   |   |
| <code>par(mar = c(4,4,2,1))</code>   | legt die Plot-Ränder in R fest (unten, links, oben, rechts) in Zeilenbreite.  |   |
| <code>mapplot()</code>   | stellt mehrere Spalten einer Matrix gleichzeitig in einem Diagramm dar, typischerweise als Linien oder Punkte.  |   |
| <code>ggplot(bwt, aes(x = bwt)) + geom_histogram(binwidth = 100) + ggtitle("Age with width = 100 and ggplot")</code> | erstellt mit ggplot2 ein Histogramm der Variable bwt, verwendet Klassenbreite 100 und setzt den Titel des Diagramms auf „Age with width = 100 and ggplot“.                              |   |
| <code>ggplot(bwt, aes(x = bwt)) + stat_ecdf()</code>   | erstellt mit ggplot2 die empirische Verteilungsfunktion (ECDF) der Variable bwt, also den <b>Anteil der Beobachtungen s einem gegebenen Wert</b> .                                      |   |
| <code>ggplot(aes(x = t_mean, colour = season)) + stat_ecdf(na.rm = TRUE)</code>                                      | erstellt mit ggplot2 empirische Verteilungsfunktionen (ECDFs) der Variable t_mean, getrennt nach season, und ignoriert NA-Werte.  |   |
| <code>ggplot(aes(x = t_mean, colour = season)) + geom_boxplot(na.rm = TRUE)</code>                                   | erstellt mit ggplot2 Boxplots der Variable t_mean, getrennt nach season, und ignoriert NA-Werte.  |   |

|  |  |                                      |
|--|--|--------------------------------------|
| <code>ggplot(aes(x = wet, colour = season)) + geom_bar()</code>  | erstellt mit ggplot2 ein Balkendiagramm, das die Häufigkeiten der Werte von wet darstellt, farblich getrennt nach season.  | Plotting                             |
| <code>ggplot(diag_tbl, aes(x = .fitted, y = .resid)) + geom_point() + geom_hline(yintercept = 0, linetype = "dashed") + labs(title = "Residuals vs Fitted", x = "Fitted values", y = "Residuals") )</code> | erstellt ein Residuen-gegen-Vorhersage-Diagramm, um Muster, Heteroskedastizität oder Modellfehler im linearen Regressionsmodell zu erkennen.   |                                      |
| <code>z_score &lt;- function(x) { CODE }</code>  | funktion mit x als attribut  | Control Flow and Functions           |
| <code>replicate(n, expr)</code>  | führt den Ausdruck <b>expr</b> n-mal aus und speichert die Ergebnisse (meist in einem Vektor oder einer Matrix)  |                                      |
| <code>which(pm &gt; 0.5)[1]</code>   | liefert den Index des ersten Elements im Vektor pm, das größer als 0.5 ist.  |                                      |
| <code>length()</code>  | gibt die Anzahl der Elemente eines Vektors, einer Liste oder eines ähnlichen Objekts zurück.   |                                      |
| <code>tail(pmf_hat, 1)</code>  | gibt das letzte Element des Vektors oder Data Frames pmf_hat zurück.   |                                      |
| <code>seq_along(grid)</code>   | erzeugt eine Zahlenfolge von 1 bis zur Länge von grid, also die passenden Indizes für grid   |                                      |
| <code>select(-e0r)</code>  | entfernt die Spalte e0r aus einem Data Frame (typischerweise mit dplyr::select).   |                                      |
| <code>na.omit()</code>   | entfernt alle Zeilen, die mindestens einen NA-Wert enthalten, aus einem Objekt (z. B. Data Frame oder Vektor).   |                                      |
| <code>mutate()</code>  | fügt neue Spalten zu einem Data Frame hinzu oder verändert bestehende Spalten, meist auf Basis bereits vorhandener Variablen (aus dplyr).  |                                      |
| <code>select()</code>  | wählt gezielt Spalten aus einem Data Frame aus oder schließt bestimmte Spalten aus (typischerweise aus dem Paket dplyr).   |                                      |
| <code>filter()</code>  | wählt Zeilen aus einem Data Frame aus, die eine oder mehrere logische Bedingungen erfüllen (typischerweise aus dem Paket dplyr).   | Pipeline Functions                   |
| <code>filter(!is.na(wet))</code>   | entfernt alle Zeilen, in denen die Variable wet den Wert NA hat, und behält nur vollständige Beobachtungen für wet.  |                                      |
| <code>case_when()</code>   | erstellt neue Variablen anhand mehrerer Wenn-Dann-Bedingungen und weist je nach erfüllter Bedingung entsprechende Werte zu (aus dplyr).  |                                      |
| <code>as_tibble()</code>   | wandelt ein Objekt (z. B. Data Frame oder Matrix) in ein Tibble um, eine moderne, übersichtlichere Form eines Data Frames (aus tibble).  |                                      |
| <code>summarise(across(everything(), ~sum(is.na(.))))</code>   | zählt für jede Spalte eines Data Frames die Anzahl der NA-Werte und gibt das Ergebnis als eine Zeile zurück.   |                                      |
| <code>apply()</code>   | Wendet eine Funktion auf Zeilen oder Spalten einer Matrix oder eines Dataframes an.  |                                      |
| <code>lapply()</code>  | Wendet eine Funktion auf jedes Listenelement an und gibt eine Liste zurück.  | The apply Family                     |
| <code>sapply()</code>  | Wie lapply(), versucht aber das Ergebnis in einen Vektor oder eine Matrix zu vereinfachen.   |                                      |
| <code>vapply()</code>  | Wie sapply(), aber mit <b>vordefiniertem Rückgabetyp</b> (sicherer).   |                                      |
| <code>tapply()</code>  | Wendet eine Funktion auf Gruppen eines Vektors an, die durch einen Faktor definiert sind.  |                                      |
| <code>mapply()</code>  | Wendet eine Funktion auf mehrere Vektoren oder Listen gleichzeitig an (mehrere Argumente).   |                                      |
| <code>rapply()</code>  | Rekursive Variante von lapply() für verschachtelte Listen.   |                                      |
| <code>eapply()</code>  | Wendet eine Funktion auf alle Objekte in einem Environment an.   |                                      |
| <code>fit &lt;- lm(y ~ x)</code>   | Erstellt ein lineares Regressionsmodell (y abhängig von x).  | Simple Linear Models and Diagnostics |
| <code>summary(fit)</code>  | Zeigt detaillierte Modellzusammenfassung (Koeffizienten, R <sup>2</sup> , Standardfehler etc.).  |                                      |
| <code>tidy(fit)</code>   | wandelt die Ergebnisse eines Modellobjekts (z. B. aus lm()) in eine übersichtliche Tabellenform um, mit Koeffizienten,   |                                      |
| <code>glance(fit)</code>   | liefert eine kompakte Zusammenfassung des Modells (z. B. R <sup>2</sup> , adj. R <sup>2</sup> , AIC, BIC, Residualstandardfehler) in einer einzigen ergänzt die ursprünglichen Daten um modellbezogene Werte, z. B. Vorhersagen, Residuen und Hebelwerte, und gibt sie als |                                      |
| <code>p &lt;- c("1" = 1/2, "2" = 1/3, "3" = 1/6)</code>  | setzt die Wahrscheinlichkeiten für die Ereignisse {1}, {2} und {3}.  | Probabilities                        |
| <code>P_12 = sum(p[c("1", "2")])</code>  | wählt die Wahrscheinlichkeiten für die Ereignisse {1} und {2} aus.   Ergibt also c(1/2, 1/3).  |                                      |
| <code>choose(deck, cards)</code>   | die Anzahl der möglichen Kombinationen, wenn man cards Karten ohne Zurücklegen und ohne Reihenfolge aus einem deck von 52 Karten zieht.  |                                      |
| <code>dhyper(x, m=8, n=12, k=5)</code>   | berechnet die Wahrscheinlichkeitsdichte der hypergeometrischen Verteilung – also die Wahrscheinlichkeit, genau k Erfolge zu ziehen, wenn man 5 Objekte ohne Zurücklegen aus einer Gesamtmenge von 20 (8 Erfolge, 12 Misserfolge) zieht.                                    |                                      |
| <code>pnorm(2)</code>  | gibt die Wahrscheinlichkeit an, dass eine standardnormalverteilte Zufallsvariable kleiner oder gleich 2 ist (also P(X ≤ 2)P(X ≥ 2)).   |                                      |
| <code>pnorm(6) - pnorm(2)</code>   | Zwischen 2 und 7   |                                      |
| <code>ecdf(x)</code>   | erstellt die empirische Verteilungsfunktion der Daten in x, also eine Funktion, die für jeden Wert den Anteil der Beobachtungen ≤ diesem Wert angibt.  | Random Numbers                       |
| <code>sample(1:100,N, replace = TRUE)</code>   | wählt zufällig N verschiedene Zahlen aus dem Bereich 1 bis 100 aus   |                                      |
| <code>sample(c("H", "T"), 3*N, TRUE)</code>  | erzeugt eine Zufallsauswahl aus "H" und "T" (z. B. Kopf oder Zahl) mit insgesamt 3 × N Ziehungen, mit Zurücklegen.   |                                      |
| <code>rgeom(n, prob = p)</code>  | erzeugt n Zufallswerte aus der geometrischen Verteilung mit Erfolgswahrscheinlichkeit p (Anzahl der Fehlversuche vor dem ersten Erfolg).   |                                      |
| <code>rbinom()</code>  | erzeugt Zufallszahlen aus der Binomialverteilung, also die Anzahl der Erfolge in einer festen Anzahl von Versuchen mit gegebener Erfolgswahrscheinlichkeit.  |                                      |
| <code>runif(N)</code>  | erzeugt N gleichverteilte Zufallszahlen im Intervall [0,1]   |                                      |
| <code>rexp(n, lambda)</code>   | erzeugt n Zufallswerte aus der Exponentialverteilung mit Rateparameter lambda.   |                                      |
| <code>summary(bwt\$bwt)</code>   | gibt eine statistische Zusammenfassung der Variable bwt aus, also Minimum, 1. Quartil, Median, Mittelwert, 3. Quartil und Maximum.   | Data distribution                    |
| <code>quantile(bwt\$lwt, 0.75)</code>  | berechnet das 75%-Quantil (3. Quartil) der Variable lwt  |                                      |
| <code>quantile(bwt\$lwt, 0.25)</code>  | berechnet das 25%-Quantil (1. Quartil) der Variable lwt  |                                      |
| <code>(quantile(bwt\$lwt, 0.75) - quantile(bwt\$lwt, 0.25))</code>   | Interquartile Range (IQR), also die Spannweite zwischen dem 3. und 1. Quartil.   |                                      |