

Statistics 2026		
Befehl	Beschreibung	Gruppe
set.seed(n)	Legt den Startwert für den Zufallsgenerator fest, damit Ergebnisse reproduzierbar sind.	Objects, Assignment, and Basic Arithmetic
a %/% b	quotient	
a %% b	remainder	
a^b	a to the power of b	
floor(x)	rundet das Ergebnis auf die nächstkleinere ganze Zahl ab.	
rm()	delete b	
seq(0,1,0.01)	erzeugt eine Zahlenfolge von 0 bis 1 in Schritten von 0,01.	
max()	gibt den größten Wert eines numerischen Vektors oder Objekts zurück (optional unter Ignorieren von NA).	
mean()	mean / Durchschnitt	
sqrt()	berechnet die Quadratwurzel der übergebenen Zahl(en).	
abs()	berechnet den Absolutbetrag einer Zahl bzw. eines numerischen Vektors. (Abstand zur 0)	
%>%	Pipeline	
as.Date(df\$MESS_DATUM)	wandelt die Spalte MESS_DATUM aus dem Data Frame df in ein Date-Objekt um.	
sum()	berechnet die Summe aller Elemente eines numerischen Vektors oder Objekts (optional ohne NA).	
count(wet)	zählt, wie oft jeder unterschiedliche Wert in der Variable wet vorkommt, und gibt die Häufigkeiten aus.	
mean(x%>%3==0 x%>%5==0)	Berechnet den Anteil (relative Häufigkeit) der Elemente in x, die durch 3 oder 5 teilbar sind da TRUE als 1 und FALSE als 0 gemittelt wird.	Vectors and Vectorized Operations
mean(A & C)	berechnet den Anteil der Elemente, bei denen sowohl A als auch C TRUE sind — also die gemeinsame Wahrscheinlichkeit (Schnittmenge) beider Bedingungen.	
var()	variation	
sd()	standard deviation / Standardabweichung	
iseven <- x %>% 2 == 0	logical Vector	
identical(x, y)	Prüft, ob zwei Objekte vollständig identisch sind (Werte, Typen, Länge etc.).	
list(poly = poly, stats = stats)	list of contents	
y[2:4]	Select elements 2 through 4 of a vector	
y[-1]	Exclude element 1 of a vector	
y + c(1, 2)	Add the vector c(1, 2) to y (gets repeated to the number of digits needed for sum)	
any()	prüft, ob mindestens ein Element in einem logischen Vektor TRUE ist, und gibt entsprechend TRUE oder FALSE zurück.	
duplicated(v)	gibt für jedes Element in v TRUE zurück, wenn es schon früher im Vektor vorkam, also ein Duplikat ist, sonst FALSE.	
sort()	ordnet die Elemente eines Vektors standardmäßig aufsteigend (optional auch absteigend).	
which.max()	liefert den Index des größten Werts im Vektor	
median()	berechnet den Median, also den mittleren Wert eines numerischen Vektors.	
factor(other)	converts vector in factors (same elements get the same number)	Factors and Tables
table(other)	frequency table	
prop.table(other)	proportion table	
matrix(1:9, nrow = 3, byrow = TRUE)	Creates a 3x3 matrix M filled by rows with the numbers 1 to 9.	Matrices and Matrix Algebra
t(M)	gegenmatrix	
M %*% t(M)	matrix multiplication	
cbind(o1,o2)	fügt die Objekte o1 und o2 spaltenweise zu einer Matrix oder einem Data Frame zusammen.	
rowSums(M)	berechnet die Summe der Werte jeder Zeile der Matrix oder des Data Frames M.	
cbind()	fügt mehrere Vektoren, Matrizen oder Data Frames spaltenweise zusammen und erstellt daraus eine Matrix oder einen Data Frame.	
M[, 2]	Extract the second column of matrix	
M[Row,Column]	Select one entry	
sweep()	wendet eine Operation (z. B. +, -, /) zeilen- oder spaltenweise auf ein Array/Matrix an , meist zum Addieren, Subtrahieren, Multiplizieren oder Dividieren mit einem Vektor.	
nrow()	gibt die Anzahl der Zeilen eines Data Frames oder einer Matrix zurück.	
rnorm(5)	5 normalverteilte zufällige Zahlen	Lists and Data Frames
dnorm(x, mean, sd)	Liefert die theoretische Dichtefunktion der Normalverteilung N(mean, sd).	
rep("MS", 5)	string 5 mal in vector speichern	
data.frame(x = x, initials = initials)	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.	
DF\$x	Select elements from data frame (selects x from DF)	
read.csv("path/to/your/file.csv", sep = ";")	Reading a CSV File with a Different Delimiter (standard is ",")	Reading/Writing
write.csv(df, "D:\\Bobdata.csv", row.names=FALSE)	Export a Data Frame to a CSV File	
fp <- tempfile(fileext = ".csv")	create a temporary file to write the contents to	
selectDWD()	filtert und wählt passende DWD-Datensätze bzw. Stationen anhand von Kriterien wie Parameter, Zeitraum oder Stations-ID aus.	
dataDWD(meta, read = FALSE)	lädt die zu den Metadaten meta gehörenden DWD-Dateien herunter, ohne sie direkt einzulesen, und gibt stattdessen Dateipfade bzw. Dateiinformationen zurück.	
readDWD(file)	liest eine DWD-Datendatei aus einer lokalen Datei ein und gibt die enthaltenen Wetterdaten als Data Frame/Tibble zurück.	Plotting
plot(DF2\$x, DF2\$y, pch = 19, main = "Scatter with LS line", xlab = "x", ylab = "y")	create a simple plot with descriptions	
hist(x, breaks, freq, main)	erstellt ein Histogramm, das die Häufigkeitsverteilung von numerischen Daten grafisch darstellt.	
abline(lm(y ~ x, data = DF2))	fitted least-squares line	
curve(expr, add = TRUE)	Zeichnet eine Funktion (z. B. Dichtekurve dnorm) in ein bestehendes Diagramm.	
par(mar = c(4,4,2,1))	legt die Plot-Ränder in R fest (unten, links, oben, rechts) in Zeilenbreite.	
matplot()	stellt mehrere Spalten einer Matrix gleichzeitig in einem Diagramm dar, typischerweise als Linien oder Punkte.	
ggplot(bwt, aes(x = bwt)) + geom_histogram(binwidth = 100) + ggtitle("Age with width = 100 and ggplot")	erstellt mit ggplot2 ein Histogramm der Variable bwt, verwendet Klassenbreite 100 und setzt den Titel des Diagramms auf „Age with width = 100 and ggplot“.	
ggplot(bwt, aes(x = bwt)) + stat_ecdf()	erstellt mit ggplot2 die empirische Verteilungsfunktion (ECDF) der Variable bwt, also den Anteil der Beobachtungen ≤ einem gegebenen Wert.	
ggplot(aes(x = t_mean, colour = season)) + stat_ecdf(na.rm = TRUE)	erstellt mit ggplot2 empirische Verteilungsfunktionen (ECDFs) der Variable t_mean, getrennt nach season, und ignoriert NA-Werte.	
ggplot(aes(x = t_mean, colour = season)) + geom_boxplot(na.rm = TRUE)	erstellt mit ggplot2 Boxplots der Variable t_mean, getrennt nach season, und ignoriert NA-Werte.	

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