Meike Steinhilber
Maria Kapsali
Patricia Heise

Stiftung Innovation in der Hochschullehre

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Online Tutorial Tool zum Erlernen von R

Grundlagen

otter vermittelt die Grundlagen der R-Programmierung und ist damit für absolute Neulinge geeignet. Es setzt die Basis dafür, eigenständig mit R zu arbeiten und Methoden anwenden zu können.

Didaktischer Einsatz

Begleitend, zusätzlich oder unahängig zur Veranstaltung – otter ist mit inhaltlichen Bausteinen aufgebaut und so flexibel in der Lehre einsetzbar.

Praktische Übungen

In Otter werden die Themen einfach erklärt. Zur Veranschaulichung gibt es ausführbare Beispielcodes sowie praktische Übungsaufgaben.

Barrierefreiheit

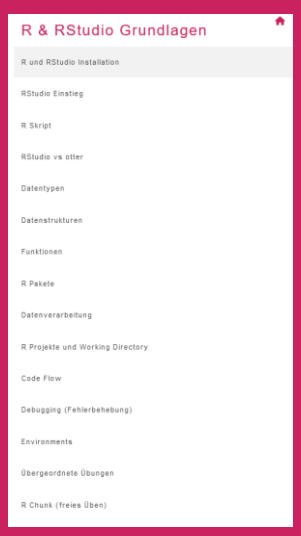
otter wird screenreader-freundlich geschrieben. Außerdem werden wir Einstellungen zum Anpassen der Schriftfarbe und -größe einbauen.

Feedback

Übungsaufgaben können automatisiert auf Richtigkeit geprüft werden. Wer bei einer Übung nicht weiterkommt, kann sich außerdem Lösungshinweise anzeigen lassen.

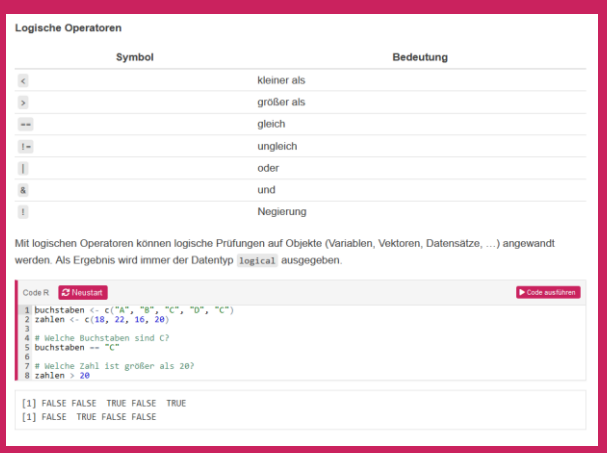
otter.uni-frankfurt.de

Bausteine



Bsp. Unterkapitel

Beispiele mit Code und Übungen



Logische Operatoren

Symbol	Bedeutung
<	kleiner als
>	größer als
==	gleich
!=	ungleich
|	oder
&	und
!	Negierung

Mit logischen Operatoren können logische Prüfungen auf Objekte (Variablen, Vektoren, Datensätze, ...) angewandt werden. Als Ergebnis wird immer der Datentyp `logical` ausgegeben.

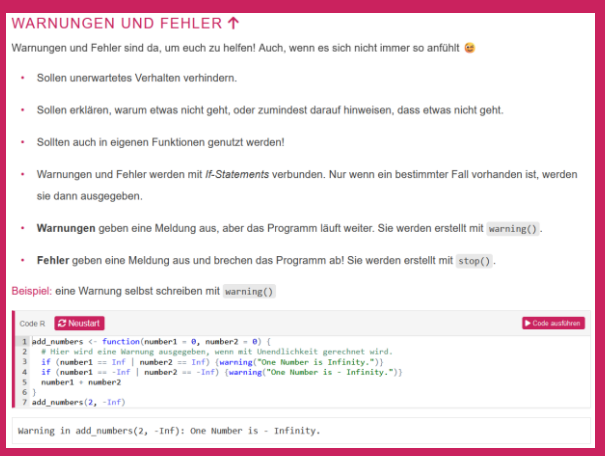
Code R | Neustart | Code ausführen

```
buchstaben <- c("A", "B", "C", "D", "C")
zahlen <- c(18, 22, 16, 20)

# Welche Buchstaben sind C?
buchstaben == "C"

# Welche Zahl ist größer als 20?
zahlen > 20
```

```
[1] FALSE FALSE  TRUE FALSE  TRUE
[1] FALSE  TRUE FALSE FALSE
```



WARNUNGEN UND FEHLER ↑

Warnungen und Fehler sind da, um euch zu helfen! Auch, wenn es sich nicht immer so anfühlt 😅

- Sollen unerwartetes Verhalten verhindern.
- Sollen erklären, warum etwas nicht geht, oder zumindest darauf hinweisen, dass etwas nicht geht.
- Sollten auch in eigenen Funktionen genutzt werden!
- Warnungen und Fehler werden mit *If-Statements* verbunden. Nur wenn ein bestimmter Fall vorhanden ist, werden sie dann ausgegeben.
- **Warnungen** geben eine Meldung aus, aber das Programm läuft weiter. Sie werden erstellt mit `warning()`.
- **Fehler** geben eine Meldung aus und brechen das Programm ab! Sie werden erstellt mit `stop()`.

Beispiel: eine Warnung selbst schreiben mit `warning()`

Code R | Neustart | Code ausführen

```
add_numbers <- function(number1 = 0, number2 = 0) {
  # Hier wird eine Warnung ausgegeben, wenn mit Unendlichkeit gerechnet wird.
  if (number1 == Inf | number2 == Inf) {warning("One Number is Infinity.")}
  if (number1 == -Inf | number2 == -Inf) {warning("One Number is - Infinity.")}
  number1 + number2
}
add_numbers(2, -Inf)
```

```
Warning in add_numbers(2, -Inf): One Number is - Infinity.
```

Beispiel für Übungsaufgaben

ÜBUNGEN

Vorbereitung:

```
namen <- c("Lisa Müller", "Peter Bauer", "Hannah Schmidt")
angemeldet <- c(TRUE, FALSE, TRUE)
alter <- c(18, 22, 16)
```

Aufgabe: *Beantworte die gestellte Frage mit R Code.*

Code R | Neustart | Tipps | Code ausführen | Antwort einreichen

```
# Wer von den Personen heißt NICHT Lisa Müller?


```

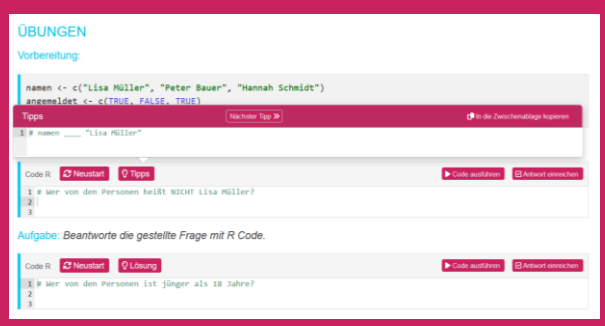
Aufgabe: *Beantworte die gestellte Frage mit R Code.*

Code R | Neustart | Lösung | Code ausführen | Antwort einreichen

```
# Wer von den Personen ist jünger als 18 Jahre?


```

Beispiel für Lösungshinweise



ÜBUNGEN

Vorbereitung:

```
namen <- c("Lisa Müller", "Peter Bauer", "Hannah Schmidt")
angemeldet <- c(TRUE, FALSE, TRUE)
```

Tipps | Nächster Tipp | In die Zwischenablage kopieren

```
# namen ___ "Lisa Müller"
```

Code R | Neustart | Tipps | Code ausführen | Antwort einreichen

```
# Wer von den Personen heißt NICHT Lisa Müller?


```

Aufgabe: *Beantworte die gestellte Frage mit R Code.*

Code R | Neustart | Lösung | Code ausführen | Antwort einreichen

```
# Wer von den Personen ist jünger als 18 Jahre?


```

Beispiel für Feedback

Aufgabe: *Beantworte die gestellte Frage mit R Code.*

Code R | Neustart | Tipps | Code ausführen | Antwort einreichen

```
# Wer von den Personen heißt NICHT Lisa Müller?

namen != "Lisa Müller"
```

```
[1] FALSE  TRUE  TRUE
```

Richtig!