

R Vorkurs

Teil 2

Martin Arnold & Jens Klenke

28.10 - 30.10.2020



Überischt

1. Logische Operatoren
2. Dataframes
3. Listen
4. Bedingte Anweisungen
5. Schleifen
6. Funktionen
7. R Pakete

Logische Operatoren

Einführung

Die Klasse `logical` habt ihr im ersten Teil schon mal gesehen (`is.na()`)

Logische Vergleiche werden für das Programmieren bedingter Anweisungen benötigt. Sie kommen aber auch häufig bei Schleifen und beim Subsetzen von Daten zum Einsatz.

Das Ergebnis logischer Vergleiche ist immer ein *boolscher Wert*, also TRUE / FALSE bzw. T / F.

Die wichtigsten Operatoren im Überblick:

```
== # "ist gleich"
!= # "ist ungleich"
<  # "ist kleiner"
<= # "ist kleiner oder gleich"
>  # "ist größer"
>= # "ist größer gleich"
&  # "logisches 'Und'"
|  # "logisches 'Oder'"
!  # einen boolschen Wert "negieren"
```

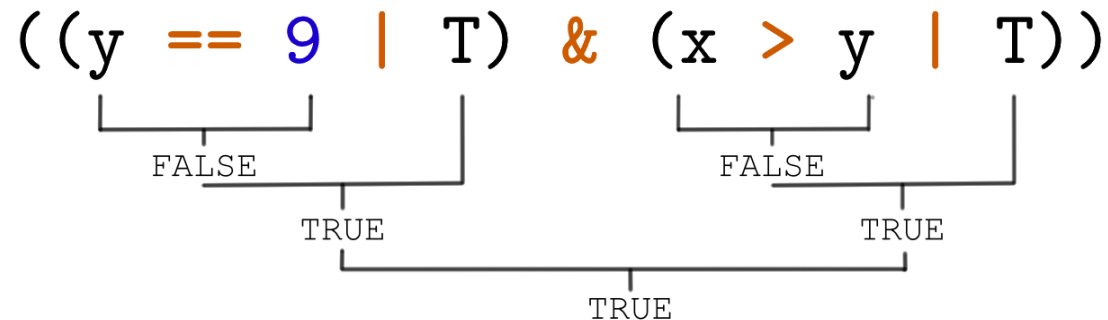
Logische Operatoren

Beispiele

Hier ein paar Beispiele:

```
x <- 5
y <- 10

x == y
x > y | y == 10
x > y | x == 10
!(x == y | y < x)
((y == 9 | T) & (x > y | T))
!T & !F
```



Logische Operatoren

Übungsaufgaben --- 1

1. Sind die folgenden Ausdrücke TRUE oder FALSE?

- $5 \geq 5$
- $5 > 5$
- $T = 5$
- $T \wedge F \vee F \wedge T$
- $F \wedge F \wedge F \vee T$
- $(\neg(5 > 3) \vee A = B)$
- $\neg(((T > F) > T) \wedge \neg T)$

Logische Operatoren

Übungsaufgaben --- 2

2. Es sei $z <- c(1, 2, NA, 4)$. Überprüfen Sie die folgenden Aussagen mittels einer Logikabfrage in *R*.

- Die Länge des Vektors z ist ungleich 2.
- Die Länge der logischen Überprüfungen, ob die einzelnen Elementen gleich 2 sind, ist 4.
- Der Vektor z hat die Klasse `numeric`.
- Einige Elemente des Vektors z sind `NA`.
- Das Zweite Element des Vektors z ist `numeric`
- Das Minimum und das Maximum sind ungleich.

Logische Operatoren

Übungsaufgaben --- 3

3. Es sei $M <- matrix(1:9, ncol = 3)$. Was ergeben folgende Ausdrücke:

- `sum$(M[, 1]) == 6$`
- `max$(M[, 2]) <= 5$`
- `M[2, 2] != 4 & M[2, 2] > 6`

Dataframes

Über Dataframes

Ein dataframe ist eine Sammlung von Variablen, ähnlich einer Matrix.

Am Beispiel des Datensatzes `iris`:

```
iris[1:10, ]
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1           5.1           3.5           1.4           0.2
## 2           4.9           3.0           1.4           0.2
## 3           4.7           3.2           1.3           0.2
## 4           4.6           3.1           1.5           0.2
## 5           5.0           3.6           1.4           0.2
## 6           5.4           3.9           1.7           0.4
## 7           4.6           3.4           1.4           0.3
## 8           5.0           3.4           1.5           0.2
## 9           4.4           2.9           1.4           0.2
## 10          4.9           3.1           1.5           0.1
##      Species
## 1      setosa
## 2      setosa
```


Dataframes

Über Dataframes

Die Funktion `str()` liefert Informationen über die Struktur eines Objekts:

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

Wir sehen:

- das Objekt `iris` hat die Klasse `data.frame`
- `iris` hat 150 Zeilen (Beobachtungen) und 5 Spalten (Variablen).
- Vier der Variablen gehören zur Klasse `numeric`, 1 Variable zur Klasse `Factor`

Dataframes

Erstellen von dataframes

Ein dataframe wird mit der Funktion `data.frame()` erstellt. Hierzu übergeben wir einfach Vektoren, welche als Spalten gruppiert werden sollen. Spalten können benannt werden.

Anders als bei Matrizen müssen die einzelnen Spalten (wie oben gesehen) nicht derselben Klasse angehören!

```
df <- data.frame(Letters = c("A", "B", "C", "D"),  
                 Numbers = 1:4,  
                 Logicals = c(T, F, FALSE, TRUE),  
                 z)
```

df

##	Letters	Numbers	Logicals	z
## 1	A	1	TRUE	1
## 2	B	2	FALSE	2
## 3	C	3	FALSE	NA
## 4	D	4	TRUE	4

Zugriff auf Spalten und Elemente

Auf die einzelnen Zellen in einem dataframe kann man wie bei Matrizen durch Indexierung mit `[]` zugreifen. Zugriff auf einzelne Variablen/Spalten erhält man mit `$`:

```
df[ , 1]      # Spalte 1
```

```
## [1] "A" "B" "C" "D"
```

```
df[1, ]      # Zeile 1
```

```
##   Letters Numbers Logicals z  
## 1      A      1     TRUE 1
```

```
df[1, 1]     # Zeile 1, Spalte 1
```

```
## [1] "A"
```

```
df[ , 1:2]   # Spalten 1 und 2
```

```
##   Letters Numbers  
## 1      A      1  
## 2      B      2  
## 3      C      3  
## 4      D      4
```

Dataframes

Zugriff auf Spalten und Elemente

```
df$Numbers      # Spalte/Variable "Numbers"
```

```
## [1] 1 2 3 4
```

Subsetting:

```
df[df$Letters == "B", ]
```

```
##   Letters Numbers Logicals z
## 2      B        2    FALSE 2
```

```
df[df$Numbers > 2, ]
```

```
##   Letters Numbers Logicals z
## 3      C        3    FALSE NA
## 4      D        4     TRUE  4
```

Dataframes

Bearbeiten von dataframes

Ein dataframe ist nach seiner Erstellung nicht unveränderbar. Man kann Spalten und Zeilen hinzufügen oder entfernen. Das gilt auch für einzelne Beobachtungen.

```
df$Greeks <- c("alpha", "beta", "gamma", "delta") # Hinzufügen einer Spalte "Greeks"
df        <- df[-2, ]                             # Entfernen der zweiten Zeile
df[2, 2]  <- NA                                     # Beobachtung an Stelle 3x2 auf NA setzen
df
```

##	Letters	Numbers	Logicals	z	Greeks
## 1	A	1	TRUE	1	alpha
## 3	C	NA	FALSE	NA	gamma
## 4	D	4	TRUE	4	delta

Dataframes

Übungsaufgaben

4. Verschaffen Sie sich einen Überblick über den Datensatz *mtcars* (dieser ist in base R bereits geladen)
 - Aus wieviele Variablen besteht er? Welche Klasse haben die einzelnen Variablen?
5. Lassen Sie sich folgende Subsets von *mtcars* ausgeben:
 - nur die Variable *mpg*
 - nur die ersten drei Zeilen
 - nur die ersten drei Variablen
 - nur die ersten beiden Beobachtungen der Variablen *cyl* und *hp*
 - alle Beobachtungen deren Ausprägung der Variablen *hp* größer ist als 200

Dataframes

Übungsaufgaben

6. Erstellen Sie einen dataframe *persons* mit den Variablen Name (character), Height (cm, numeric) und Weight (kg, numeric) von 5 fiktiven Personen.
- Lassen Sie sich das Körpergewicht der 3. Person anzeigen.
 - Lassen Sie sich nun die Körpergröße aller Personen anzeigen.
 - Fügen Sie die Variable "Augenfarbe" hinzu. Die Ausprägungen sollten vom Typ character sein. Schauen Sie sich den veränderten dataframe an.

Listen

Listen erzeugen

Listen werden mit der Funktion `list()` erzeugt. Ein \hil{Vorteil} von Listen ist, dass die einzelnen Elemente von unterschiedlicher Größe und Typ sein können.

Der Zugriff auf Listenelemente erfolgt ebenfalls mit `$`:

```
my.list <- list(A = 1:5, B = mtcars, C = list(letters, LETTERS))
```

Viele Funktionen in R geben Ergebnisse als Listen zurück.

```
# Regressionsmodell
model <- lm(mpg ~ hp, data = mtcars)
str(model)
```

```
## List of 12
## $ coefficients : Named num [1:2] 30.0989 -0.0682
```

(Mehr dazu in Teil 3)