Logische Vektoren

7. Logische Vektoren

Ein logischer Vektor besteht aus True und False Elementen und ist u.a. für die Indizierung sehr nützlich. Im Allgemeinen sind logische Vektoren für fast alle Aufgaben in R unentbehrlich.

7.1 Wann kommen logische Vektoren vor?

Logische Vektoren entstehen wenn **comparison operators** angewandt werden. Hier sind **x** und **y** zwei Vektoren - angenommen Fall 1 oder Fall 2 unten.

<code>x == y</code>	x equals y
<code>x != y</code>	x is not equal to y
<code>x < y</code>	x is less than y
<code>x <= y</code>	x is less than or equal to y
<code>x > y</code>	x is greater than y
<code>x >= y</code>	x is greater than or equal to y
<code>x %in% y</code>	Siehe Fall 3 unten

Fall 1: **x** und **y** sind Vektoren und **y** hat die Länge 1

```
x = c(10, 20, 30)
```

```
# bedeutet: welche davon sind 20?
```

```
x == 20
```

```
labs <- c("I", "E", "A", "I", "I", "A", "E")
```

```
# Welche Etikettierung(en) ist "I"?
```

```
# Welche sind nicht "E"?
```

Fall 2: **x** and **y** sind Vektoren derselben Länge.

```
x = c(10, 20, 30)
```

```
y = c(10, 19, 30)
```

```
# Haben x[n] und y[n] denselben Wert?
```

```
x == y
```

```
# Ist Elemente x[n] größer als y[n]?
```

```
x > y
```

Fall 3: x und y haben nicht dieselbe Länge (und y hat nicht die Länge 1)

Hier kommen logische Vektoren nur in diesem einzigen Fall vor:

```
x <- c(10, 20, 30)
```

```
# Besteht x aus entweder 20 oder 30?
```

```
(x == 20) | (x == 30)
```

```
# Der %in% Operator ist eine Abkürzung dafür
```

```
x %in% c(20, 30)
```

```
# oder
```

```
y = c(20, 30)
```

```
x %in% y
```

7.2 Indizierung: Vektoren

```
# Etikettierungen
```

```
labs = c("I", "E", "A", "I", "I", "A", "E")
```

```
# Entsprechende Dauer (labs und d sind parallel zueinander)
```

```
d = c(80, 90, 110, 85, 100, 150, 105)
```

```
# Die Dauern der "I" Vokale
```

```
temp = labs == "I"
```

```
d[temp]
```

```
# das gleiche
```

```
d[labs=="I"]
```

```
# Die durchschnittliche Dauer aller Vokale außer "E"
```

```
temp = labs != "E"
```

```
mean(d[temp])
```

```
words = c("it's", "entirely", "inappropriate", "to", "wear", "black", "at", "a",  
"wedding")
```

```
d = c(177, 551, 655, 138, 194, 486, 76, 44, 412)
```

```
# Dauer von "to" und "at"
```

```
temp = words %in% c("to", "at")
```

```
d[temp]
```

```
# Die Dauer aller anderen Wörter (nicht "to", nicht "at"
```

```
d[!temp]
```

```
# Dauer aller Wörter außer "wedding"?
```

```
# Welche Wörter haben eine Dauer von mehr als 400 ms?
```

```
# Welche Wörter haben eine Dauer größer als die durchschnittliche Dauer?
```

```
# Welches Wort hat die größte Dauer?
```

```
# Die Wörter mit einer größeren Dauer als die Dauer von "black"
```

7.3 Funktion für logische Vektoren: `any()`, `sum()`

```
labs <- c("I", "E", "A", "I", "I", "A", "E")
```

```
# Kommt "I" überhaupt vor (tritt "I" mindestens einmal auf?)
```

```
temp = labs=="I"
```

```
any(temp)
```

```
# oder
```

```
any(labs == "I")
```

```
# Wieviele "I"s kommen vor?
```

```
sum(labs == "I")
```

7.4 Boolean `&` und Boolean `|`

Sowohl `True & True` als auch `True | True` sind `True`

Sowohl `False & False` als auch `False | False` sind `False`

Jedoch

```
T & F
```

```
F
```

```
T | F
```

```
T
```

Anwendung. Hier sind einige Etikettierungen und F1-Werte

```
vowlab = c("I", "E", "A", "I", "O")
```

```
werte = c(280, 420, 960, 320, 550)
```

Welche Vokale liegen im Bereich 300-700 Hz (Für welche Vokale ist `werte > 300`

und `werte < 700` `True`?)

```
temp = werte > 300 & werte < 700
```

```
vowlab[temp]
```

Welche Vokale liegen außerhalb von diesem Bereich. (Für welche Vokale ist `werte < 300` oder `werte > 700` `True`?)

```
temp = werte < 300 | werte > 700
```

```
vowlab[temp]
```

7.5 Logische Vektoren, Indizierung und Matrizen/Segmentlisten

Was vor dem Komma erscheint, bezieht sich auf die Reihen; was nach dem Komma erscheint, auf die Spalten

```

duration = c(33, 56, 37, 50, 49, 21)
F1 = c(979, 592, 224, 597, 281, 737)
F2 = c(1680, 1940, 2451, 2050, 2624, 1544)
# Eine Matrix
daten = cbind(duration, F1, F2)
labs = c("a", "e", "i", "e", "i", "a")

# Die Reihen der Matrix, die den "a" Vokalen entsprechen
temp = labs == "a"
daten[temp,]

# Das gleiche aber nur die ersten 2 Spalten
daten[temp,1:2]

# Die Reihen der Matrix für "a" und "i" Vokale
temp = labs %in% c("a", "i")
daten[temp,]

# Das gleiche aber nur F2 (die 3e Spalte)
daten[temp,3]

# Die Reihen der Matrix, für die F2 - F1 größer als 1500 Hz ist?

# Das gleiche aber nur die Dauern (ich will die Dauern sehen, wenn F1 - F2 > 1500)

# Eine zweispaltige Matrix der Dauern und F2 für "e" Vokale

# F1 und F2 (eine 2 spaltige Matrix) wenn die Dauern weniger als 40 ms sind.

# Die Vokale, für die F2 über 2000 Hz liegt

vowlax ist eine Segmente. Die Etikettierungen davon
labs = label(vowlax)

# Eine Tabelle davon
table(labs)

# Die Reihen von vowlax der "a" Vokale (also eine Segmentliste der "a" Vokale)

# Die Dauer der Segmente von vowlax
d = dur(vowlax)

```

Der Dauermittelwert der "I" Vokale

Eine Segmentliste der Vokale mit einer Dauer größer als
der Dauermittelwert der "I" Vokale

Gibt es irgendwelche Vokale mit einer Dauer (a) über 93 ms
(b) zwischen 90 und 100 ms? (Für (b) braucht man den logischen &)

Wieviele "I" Vokale gibt es mit einer Dauer von weniger als 30 ms?
Und welche Vokale sind das? (Also die Segmentliste davon).

Wieviele "I" Vokale haben eine Dauer, die größer ist, als die durchschnittliche
Dauer der "a" Vokale?

7.6 Zusammenfassung der wichtigsten Befehle aus 7.

x und y sind Vektoren derselben Länge, oder y ist von der Länge 1 (eins)

$x == y$	$x[n]$ gleicht $y[n]$
$x != y$	$x[n]$ gleicht nicht $y[n]$
$x < y$	$x[n]$ ist weniger als $y[n]$
$x <= y$	$x[n]$ ist weniger als oder gleicht $y[n]$
$x > y$	$x[n]$ ist größer als $y[n]$
$x >= y$	$x[n]$ ist größer als oder gleicht $y[n]$

x und y sind Vektoren unterschiedlicher Längen

$x \%in\% y$ Kommen die Elemente von y in x vor?

$temp$ ist ein logischer Vektor. zB

```
temp = c(T, F)
is.logical(temp)
[1] TRUE
```

```
class(temp)
[1] "logical"
```

x ist ein Vektor, $temp$ ist ein logischer Vektor. x und $temp$ haben dieselbe Länge (also $length(x)$ und $sum(temp)$ geben denselben Wert zurück)

```
x[temp]           # alle Elemente in x für die temp True ist
x[!temp]         # alle Elemente in x für die temp False ist
```

x ist eine Matrix oder Segmentliste oder Data-Frame, $temp$ ist ein logischer Vektor. Die Reihenanzahl von x und die Länge vom logischen Vektor geben denselben Wert zurück (also $nrow(x)$ und $sum(temp)$ haben denselben Wert)

```
x[temp,]         # alle Reihen in x für die temp True ist
x[!temp,]       # alle Reihen in x für die temp False ist
```

x ist eine Matrix oder Segmentliste oder Data-Frame, $temp$ ist ein logischer Vektor. Die Spaltenanzahl von x und die Länge vom logischen Vektor geben denselben Wert zurück (also $ncol(x)$ und $sum(temp)$ haben denselben Wert)

```
x[,temp]        # alle Spalten in x für die temp True ist
x[,!temp]       # alle Spalten in x für die temp False ist
```

$temp$ ist ein logischer Vektor

$any(temp)$	Kommt mindestens ein True in $temp$ vor?
$any(!temp)$	Kommt mindestens ein False in $temp$ vor?
$sum(temp)$	Wieviele True gibt es in $temp$?
$sum(!temp)$	Wieviele False gibt es?

$tempa$ und $tempb$ sind logische Vektoren derselben Länge. Dann:

$tempa \& tempb$	True wenn $tempa[n]$ und $tempb[n]$ True sind, sonst False
$tempa tempb$	True wenn mindestens einer von $tempa[n]$ oder $tempb[n]$ True ist, sonst False

Kopieren Sie den R-Befehl (nicht was R zurückgibt) unter jeder Frage. z.B.

Median Wert von 1 bis 10?

`median(1:10)`

7.7 Fragen

1. Hier ist ein Vektor von Mitarbeitern einer Firma:

`mit = c("Paul", "Karin", "Elke", "Georg", "Peter")`

und hier ist noch ein Vektor für die Zeit, die sie jeweils benötigen, um in die Arbeit zu kommen:

`zeit = c(50, 11, 35, 41, 12)`

1.1. Erklären Sie die Bedeutung von:

`mit[zeit > 40]`

Setzen Sie die Fragen in R-Befehle um:

1.2. Beispiel. Wer braucht genau 35 Minuten um in die Arbeit zu kommen?

`mit[zeit==35]`

1.3 Wer braucht weniger als 20 Minuten?

1.4 Wer braucht am längsten? (Hier muss auch die `max()` Funktion verwendet werden).

1.5 Wer braucht zwischen 25 und 45 Minuten?

1.6 Wie viele Mitarbeiter brauchen weniger als 40 Minuten?

1.7 Gibt es Mitarbeiter, die mehr als 45 Minuten brauchen?

2. `bridge` ist eine vorhandene Matrix in `library(emu)`. Sie fasst zusammen, die Anzahl der Autos, die zwischen 0h und 12h am Montag, Dienstag, und Mittwoch über eine Brücke fahren. Die Werte für Montag, Dienstag, und Mittwoch sind in Spalten 1-3 (geben Sie `bridge` ein, um dies zu sehen).

Hier sind die Uhrzeiten:

`zeit = as.numeric(rownames(bridge))`

Schreiben Sie R-Befehle für:

2.1 Die Uhrzeiten, zu denen mehr Autos am Mittwoch (Spalte 3) als am Dienstag (Spalte 2) über die Brücke fahren

2.2 Die Uhrzeiten, zu denen weniger als 60 Autos am Dienstag (Spalte 2) über die Brücke fahren.

2.3 Die Uhrzeiten, zu denen weniger als 5 Autos entweder am Dienstag oder am Mittwoch über die Brücke fahren.

2.4 Die Anzahl der Stunden, zu denen mehr Autos am Mittwoch als am Montag über die Brücke fahren.

2.5 Die Stunden, zu denen zwischen 60 und 100 Autos am Dienstag über die Brücke fahren.

2.6 Die Stunden, zu denen der Unterschied von Dienstag auf Mittwoch in der Anzahl der Autos, die über die Brücke fahren, weniger als 30 ist.

3. Für die vorhandene Segmentliste `dip` und deren Etikettierung, `dip.l` erstellen Sie einen Vektoren der Dauern:

`d = dur(dip)`

Schreiben Sie R-Befehle für

3.1 Die Standardabweichung der Dauer aller "aU" Segmente.

3.2. Die Anzahl der "aU" Segmente deren Dauer größer ist, als die Median-Dauer aller Segmente.

3.3. Der Bereich (range) der Dauer aller "aI" und "aU" Segmente.

3.4. Der Dauer-Höchstwert (max) der ersten 20 "aI" Segmente.

3.5. Die Proportion der "OY" Diphthonge, die eine Dauer zwischen 100 und 150 ms haben.

Antworten auf Frage 2 und 3 auf der nächsten Seite

```

zeit = as.numeric(rownames(bridge))

# 2.1
zeit[bridge[,3] > bridge[,2]]

# 2.2
zeit[bridge[,2] < 60]

# 2.3
zeit[bridge[,2] < 5 | bridge[,3] < 5]

# 2.4
sum(bridge[,3] > bridge[,1])

# 2.5
zeit[bridge[,2] > 60 & bridge[,2] < 100]

# 2.6
zeit[ bridge[,2] - bridge[,3] < 30 ]

# 3.
# 3.1
sd(d[dip.l == "aU"])

# 3.2
sum(d[dip.l == "aU"] > median(d) )

# 3.3
range(d[dip.l %in% c("aI", "aU")])

# 3.4
max(d[dip.l == "aI"][1:20])

# 3.5
sum(d[dip.l == "OY"] > 100 & d[dip.l == "OY"] < 150) / sum(dip.l == "OY")
# oder in Schritten
# Dauerwerte der "OY" Diphthonge
temp = dip.l == "OY"
doi = d[temp]
# wieviele dieser Werte liegen zwischen 100 ms und 150 ms?
sum(doi > 100 & doi < 150)
# wieviele "OY" Diphthonge gibt es?
sum(temp)
# Proportion
sum(doi > 100 & doi < 150) / sum(temp)

```