More about R: 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.

x == y	x equals y
x != y	x is not equal to y
x < y	x is less than y
x <= y	x is less than or equal to y
x > y	x is greater than y
x >= y	x is greater than or equal to y
x %in% y	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)

# Dauern von "to" und "at"
temp = words %in% c("to", "at")
d[temp]

# Dauern aller anderen Wörter (nicht "to", nicht "at"
d[!temp]

# Dauern 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, deren Dauern größer sind 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 Segmentiste der "a" Vokale)

# Die Dauern von vowlax
d = dur(vowlax)
```

Die durchschnittliche Dauer der "I" Vokale

Eine Segmentliste der Vokale, deren Dauern
größer sind als die durchschnittliche Dauer 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)	Wiviele 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. zB

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 Wieviele 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 der Dauern von allen "aI" und "aU" Segmenten.

3.4. Die Maximal-Dauer der ersten 20 "aI" Segmente.

3.5. Die Proportion der "OY" Diphthonge deren Dauern zwischen 100 und 150 ms liegt

4. Ein Data-Frame kann aus numerischen Vektoren und Schriftzeichen-Vektoren zusammengestellt werden. Sie ist vor allem in der statistischen Auswertung nützlich. So könnte ein Data-Frame zusammengestellt werden:

```
d = dur(vowlax[1:5,])
vowlab = vowlax.l[1:5]
sprecher = vowlax.spkr[1:5]
v.df = data.frame(Dauer=d, Etik=vowlab, Sprecher=sprecher)
```

Der Data-Frame lässt sich genau wie eine Matrix behandeln. zB Reihen 1:3

```
v.df[1:3,]
```

usw. Mit attach(data.frame) sind auch die Spalten vom Data-Frame durch deren Namen zugänglich (Der Data-Frame erscheint dann in der Liste mit search()). zB

```
attach(v.df)
Dauer
[1] 89.813 43.810 53.130 39.750 51.190
```

```
mean(Dauer)
[1] 55.5386

Sprecher[1:3]
[1] 67 67 67
```
usw.

Die Namen vom Data-Frame sind zugänglich mit
```
names(v.df)
[1] "Dauer"    "Etik"     "Sprecher"
```

Wenn Sie mit dem Data-Frame fertig sind, dann:
```
detach(v.df)
```

In Emu-R gibt es einen Data.frame vowlax.df. Geben Sie ein:
```
attach(vowlax.df)
```

Schauen Sie sich die ersten 3 Reihen an:

```
vowlax.df[1:3,]
  F1   F2   F3     f0      rms     dur   phonetic   word   speaker
1 562 1768 2379 148.995 81.5328 89.813         E   Geld        67
2 648 1463 2523   0.000 70.9245 43.810         a allein        67
3 703 1280 2504 127.266 75.1751 53.130         a  macht        67
```

Die Spalten enthalten Informationen zu F1, F2, F3, f0 (Grundfrequenz), Intensität (rms), phonetische Etikettierung (phonetic), Word-Etikettierung (word) und Sprecher-Etikettierung (speaker). Diese Variablen

```
names(vowlax.df)
```

sind nach attach(vowlax.df) **als Vektoren zugänglich** - siehe Beispiel v.df oben oder geben Sie z.B. ein:

```
F1
[1]  562  648  703  361  397  551  370  497  562  649  299...

phonetic
[1] "E" "a" "a" "I" "I" "E" "I" "O" "O" "a"...

word
[1] Geld           allein         macht          nicht...
```

usw.
Daher wenn man zugreifen will, auf F1 und F3 der Wörter *Geld* und *macht*:

entweder:
```
vowlax.df[word %in% c("Geld", "macht"), c(1,3)]
```

oder

```
mat = cbind(F1, F3)
mat[word %in% c("Geld", "macht"),]
```

oder in einer Zeile:

```
cbind(F1, F3)[word %in% c("Geld", "macht"),]
```

Schreiben Sie R- Befehle für:

4.1 Die Anzahl der Reihen im Data-Frame

4.2 Die Anzahl der Spalten vom Data-Frame.

4.3 Die ersten 5 Reihen vom Data-Frame.

4.4 Alle Reihen vom Data-Frame die den Vokal "I" enthalten

4.5 Die Anzahl der "I" und "E" Vokale (zusammen) die einen F2 von weniger als 1800 Hz haben.

4.6 (Die Funktion bark(x) konvertiert einen Vektor x von Hertz-Werten in Bark). Alle Reihen vom Data-Frame die "I" Vokale enthalten, fur die F3 Bark - F2 Bark weniger ist als 1 Bark.

4.7 Die F2 und F3 Werte vom Vokal mit der größten Dauer.

4.8 Die f0 und RMS-Werte der Vokale, die eine Dauer haben von mehr als 100 ms und einen f0 von weniger als 90 Hz.

4.9. Eine Tabellierung der Vokal-Etikettierungen (table() verwenden) für die F1 größer ist als 400 Hz und F2 weniger als 1600 Hz.

4.10 Die Wörter, für die F3 - F2 von "I" Vokalen größer ist als 900 Hz.

Geben Sie detach(vowlax.df) ein wenn Sie fertig sind.