

```
# 1. 'Logische Vektoren'
# '1.1 Einführung'

# Ein logischer Vektor besteht aus True und False Elementen und ist u.a.
# für Indizierung und Auswahl unentbehrlich.
# Mit logischen Vektoren stellt man Fragen:
# x == y          x gleicht y?
# x != y          x gleicht nicht y?
# x < y           x ist weniger als y? (x <= y; ... weniger oder gleich)
# x > y           x ist größer als y? (x >= y; ... größer oder gleich)

# In den meisten Fällen ist x ein Vektor und y hat die Länge 1
x = c(10, 20, 30)

# Welche davon sind 20?
x == 20

labs <- c("I", "E", "A", "I", "I", "A", "E")
# Welche davon sind "I"?
labs == "E"

# '1.2 sum(), any(), all()'
lvec = c(T, F, T, T, F, T)

# Eigenschaften von 'lvec'
class(lvec)

# Wieviele T?
sum(lvec)

# Wieviele F?
sum(!lvec)

# Kommt mindestens ein T vor?
any(lvec)

# Sind sind sie alle T?
all(lvec)

# Hier sind einige Dauern.
d <- c( 177, 551, 655, 138, 194, 486, 76, 44, 412)

# Wieviele Dauern sind über 500?

# Sind alle Dauern unter 700?

# Gibt es mindestens eine Dauer, die weniger ist als 40?

# '1.3 Die Indizierung'
# Ein Vektor einiger Wörter
words <- c("it's", "entirely", "inappropriate", "to", "wear", "black", "at",
"a", "wedding")
```

```
# Deren Dauern (d.h. die Dauer von 'entirely' ist 551 ms usw.
d <- c( 177, 551, 655, 138, 194, 486, 76, 44, 412)

# Die Dauer von 'to'
temp = words == "to"
d[temp]

# oder
d[words == "to"]

# 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?

# Alle Wörter, die eine Dauer haben, die größer ist, als die Dauer von "black"

# '1.4 Boolean & |'
# & bedeutet `und`; | bedeutet `oder`

d <- c( 177, 551, 655, 138, 194, 486, 76, 44, 412)
# Welche Dauer sind zwischen 300 und 500?
d > 300 & d < 500
temp = d > 300 & d < 500
d[temp]

# Daher: welche Welche Wörter haben eine Dauer zwischen 300 und 500?
words[temp]

# Welche Wörter haben eine Dauer von entweder weniger als 250 oder eine Dauer von
mehr als 610 ms?
temp = d < 250 | d > 610
words[temp]

# Hier ist ein Schriftzeichen-Vektor Vektor ob das Wort ein Funktions- oder
Inhaltswort ist
typ = c("F", "I", "I", "F", "I", "I", "F", "F", "I")

# Was ist die Länge von (wieviele Elemente in) 'typ'? (Funktion length())

# Bestätigen Sie, dass 'typ' und 'words' von derselben Länge sind

# Zeigen Sie welche Wörter Inhaltswörter sind

# Zeigen Sie die Dauer der Funktionswörter

# Ist der Dauermittelwert (Funktion mean()) der Inhaltswörter größer als die
# der Dauermittelwert der Funktionswörter?
```

```
# Zeigen Sie die Funktionswörter mit einer Dauer größer als 100 ms

# Gibt es irgendetwelche Inhaltswörter mit einer Dauer zwischen 300 und 500 ms?

# '1.5 Einige Fragen'
# 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)

# Was bedeutet:
mit[zeit > 40]

# Wer braucht weniger als 20 Minuten?

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

# Wer braucht zwischen 25 und 45 Minuten?

# Wieviele Mitarbeiter brauchen weniger als 40 Minuten?

# Gibt es irgendwelche Mitarbeiter, die mehr als 45 Minuten brauchen?

# '1.6 Die Indizierung von Data-Frames und Matrizen'
# Hier ist ein Data-Frame von F2 Werten von [I, E] produziert von 20 Vpn.
vok = read.table(file.path(pfadu, "vokal.txt"))
head(vok)
dim(vok)    # Anzahl der Beobachtungen (Reihen) und Variablen (Spalten)
class(vok)
nrow(vok)   # Anzahl der Beobachtungen
names(vok)  # Namen der Variablen

# Die F2-Werte sind in Spalte 1, wie man sieht. Welche F2 Werte sind über 1800 (Hz)?
vok[,1] > 1800

temp = vok[,1] > 1800
# Welche Reihen vom Data-Frame sind das?
# Alles was vor dem Komma kommt, bezieht sich auf die Reihen; danach auf die Spalten
vok[temp,]

# oder
vok[vok[,1] > 1800,]

# Welche Versuchspersonen haben F2 Werte unter 1500 Hz?
temp = vok[,1] < 1500
vok[temp,3]

# Welche Vokale haben einen F2-Wert über 1900 Hz?
```

```
temp = vok[,1] > 1900
vok[temp,2]

# Die Beobachtungen für Vpn S20

# Wieviele Beobachtungen gibt es mit F2 über 2000 Hz?
sum(vok[,1] > 2000)

# Gibt es irgendwelche Beobachtungen mit Versuchsperson S18?
any(vok[,3] == "S18")

# Sind die Reihen (Beobachtungen) 17-20 alle "E" Vokale?
all(vok[17:20,2] == "E")

# 1.7 Fragen
# Hier ist ein Data-Frame der Anzahl der Autos am Montag, Dienstag, und Mittwoch
# die über eine Brücke fahren zu gewissen Zeiten
a.df = read.table(file.path(pfadu, "zeit.txt"))

# z.B. bedeutet Reihe 1: 9 Autos am Montag, und 1 Auto
# am Dienstag fahren über die Brücke zwischen 0-1 h.

# Welche sind die Zeiten, zu denen mehr Autos
# am Mittwoch als am Dienstag über die Brücke fahren?

# Zu welchen Zeiten fahren weniger als 60 Autos am Dienstag über die Brücke?

# Was sind die Zeiten, zu denen weniger als 5 Autos
# entweder am Dienstag oder am Mittwoch über die Brücke?

# Was sind die Zeiten, zu denen zwischen 60 und 100 Autos am Dienstag über die Brücke
# fahren?

# Zu welchen Zeiten, gab es mehr als 30 Autos am Dienstag als am Mittwoch?

# Wie viele Zeiten gibt es, zu denen der Unterschied zwischen Dienstag und Mittwoch
# weniger als 30 war?

# Gibt es irgendwelche Zeiten, zu denen die Anzahl am Mittwoch größer als 60 ist?

# Ist zu jeder Zeit die Anzahl der Autos am Mittwoch kleiner als am Montag?

# Lesen Sie diesen Data-Frame ein
asp.df = read.table(file.path(pfadu, "asp.txt"))

# Wie viele Beobachtungen gibt es?

# Was sind die Namen der Variablen (Spalten)?

# Zeigen Sie die ersten paar Reihen auf dem Bildschirm

# Erstellen Sie eine Tabelle (function table()) der Konsonanten (Variable oder Spalte
```

```
4);  
# der Konsonanten in einer betonten (Variable 5) Silbe  
  
# Zeigen Sie Beobachtungen (Reihen) 100 bis 110 der Dauer (Variable 1) und der Wörter  
(Variable 2)  
# auf dem Bildschirm  
  
# Zeigen Sie Beobachtungen 10, 19, 21 von Variablen 3 und 5  
  
# Zeigen Sie alle Beobachtungen von Sprecher k01  
  
# Zeigen Sie alle Beobachtungen der unbetonten /t/s von Sprecher k01  
  
# Wieviele Beobachtungen gibt es von betonten /k/s?  
  
# Zeigen Sie die Wörter der unbetonten /k/s von Sprecher k69  
  
# Ist der Dauermittelwert der betonten /k/s größer  
# als der Dauermittelwert der betonten /t/s?  
  
# Berechnen Sie den Dauermittelwert in 'Mutter'  
  
# Zeigen Sie alle Beobachtungen von unbetonten /t/s, für die  
# die Dauer zwischen 100 und 120 ms liegt.  
  
# Lesen Sie diesen Data-Frame von Diphthongen ein  
dip.df = read.table(file.path(pfadu, "dip.txt"))  
  
# Wieviele verschiedene Versuchspersonen gibt es?  
# (Hier length() und unique() verwenden)  
  
# Tabellieren Sie die Diphthonge  
  
# Was ist der Dauermittelwert aller /aU/ Diphthonge?  
  
# Wieviele /aU/ Diphthonge gibt es mit einer Dauer größer  
# als der Dauermittelwert der /aI/ Diphthonge?  
  
# Lesen Sie den folgenden Data-Frame von verschiedenen Vokalen ein:  
v.df = read.table(file.path(pfadu, "vlax.txt"))  
  
# Wieviele Beobachtungen gibt es? Und wieviele Variablen?  
  
# Zeigen Sie alle Beobachtungen mit einem /I/ (Variable V)  
  
# Wieviele /I/ und /E/ Vokalen haben einen F2 von weniger als 1800 Hz?  
  
# Tabellieren Sie die Vokale (Funktion table()) für die F1 größer als 400 Hz  
# und F2 weniger als 1600 Hz ist.  
  
# Zeigen Sie die Wörter, für die F3 - F2 von /I/ Vokalen größer ist  
# als 900 Hz
```

