```r
# 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?
sum(d > 500)

# Sind alle Dauern unter 700?
all(d < 700)

# Gibt es mindestens eine Dauer, die weniger ist als 40?
any(d < 40)
```

```r
# '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"?
d[words != "wedding"]

# Welche Wörter haben eine Dauer von mehr als 400 ms?
words[d > 400]

# Welche Wörter haben eine Dauer größer als der Dauer-Mittelwert?
words[d > mean(d)]

# Welches Wort hat die größte Dauer?
words[d == max(d)]

# Alle Wörter, die eine Dauer haben, die größer ist, als die Dauer von "black"
words[d > d[words == "black"]]


# '1.4 Boolean & |'
# & bedeudet `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 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())
```

```
length(typ)

# Bestätigen Sie, dass 'typ' und 'words' von derselben Länge sind
length(typ) == length(words)

# Zeigen Sie welche Wörter Inhaltswörter sind
words[typ == "I"]

# Zeigen Sie die Dauer der Funktionswörter
d[typ == "I"]

# Ist der Dauermittelwert (Funktion mean())  der Inhaltswörter größer als die
# der Dauermittelwert der Funktionswörter?
mean(d[typ == "I"]) > mean(d[typ =="F"])
#
# oder (da wir nur 2 Kategorien haben:
mean(d[typ == "I"]) > mean(d[typ != "I"])

# Zeigen Sie die Funktionswörter mit einer Dauer größer als 100 ms
words[typ == "F" & d > 100]

# Gibt es irgenwelche Inhaltswörter mit einer Dauer zwischen 300 und 500 ms?
words[typ == "I" & d > 300 & d < 500]

# '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]
# Die Mitarbeiter, die mehr als 40 Min. brauchen, um in die Arbeit zu kommen

# Wer braucht weniger als 20 Minuten?
mit[zeit < 20]

# Wer braucht am längsten? (Hier muss auch die max() Funktion verwendet werden).
mit[zeit == max(zeit)]

# Wer braucht zwischen 25 und 45 Minuten?
mit[zeit > 25 & zeit < 45]

# Wieviele Mitarbeiter brauchen weniger als 40 Minuten?
sum(zeit < 40)

# Gibt es irgendwelche Mitarbeiter, die mehr als 45 Minuten brauchen?
any(zeit > 45)
```

```
# '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
vok[vok[,3] == "S20",]
# oder
vok[vok$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 Beobachtung (Reihe) 1: 9 Autos am Montag, und 1 Auto
# am Dienstag fuhren über die Brücke zwischen 0-1 h.

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

```r
a.df$Zeit[a.df$Mi > a.df$Di]
# oder
a.df[a.df[,3] > a.df[,2],1]

# Zu welchen Zeiten fuhren weniger als 60 Autos am Dienstag über die Brücke?
a.df$Zeit[a.df$Di < 60]

# Was sind die Zeiten, zu denen weniger als 5 Autos
#  entweder am Dienstag oder am Mittwoch über die Brücke?
a.df$Zeit[a.df$Di < 5 | a.df$Mi < 5]

# Was sind die Zeiten, zu denen zwischen 60 und 100 Autos am Dienstag über die Brücke
fuhren?
a.df$Zeit[a.df$Di > 60 & a.df$Di < 100]

# Zu welchen Zeiten, gab es mehr als 30 Autos am Dienstag als am Mittwoch?
a.df$Zeit[a.df$Di - a.df$Mi > 30]

# Wie viele Zeiten gibt es, zu denen der Unterschied zwischen Dienstag und Mittwoch
# weniger als 30 war?
sum(a.df$Di - a.df$Mi < 30)

# Gibt es irgendwelche Zeiten, zu denen die Anzahl am Mittwoch größer als 60 ist?
any(a.df$Mi > 60)

# Ist zu jeder Zeit die Anzahl der Autos am Mittwoch kleiner als am Montag?
all(a.df$Mi > a.df$Mo)

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

# Wie viele Beobachtungen gibt es?
nrow(asp.df)

# Was sind die Namen der Variablen (Spalten)?
names(asp.df)

# Zeigen Sie die ersten paar Reihen auf dem Bildschirm
head(asp.df)

# Erstellen Sie eine Tabelle (function table()) der Konsonanten (Variable oder Spalte
4);
# der Konsonanten in einer betonten (Variable 5) Silbe


table(asp.df$Kons)
#
oder
table(asp.df[,4])

table(asp.df$Kons[asp.df$Bet == "be"])
```

```r
# Zeigen Sie Beobachtungen (Reihen) 100 bis 110 der Dauer (Variable 1) und der Wörter
(Variable 2)
# auf dem Bildschirm
asp.df[100:110,1:2]

# Zeigen Sie Beobachtungen 10, 19, 21 von Variablen 3 und 5
asp.df[c(10, 19, 21), c(3, 5)]

# Zeigen Sie alle Beobachtungen von Sprecher k01
asp.df[asp.df$Vpn == "k01",]

# Zeigen Sie alle Beobachtungen der unbetonten /t/s von Sprecher k01
asp.df[asp.df$Vpn == "k01" & asp.df$Bet == "un",]

# Wieviele Beobachtungen gibt es von betonten /k/s?
sum(asp.df$Bet == "be" & asp.df$Kons == "k")

# Zeigen Sie die Wörter der unbetonten /k/s von Sprecher k69
asp.df$Wort[asp.df$Kons=="k" & asp.df$Vpn == "k69"]

# Ist der Dauermittelwert der betonten /k/s größer
# als der Dauermittelwert der betonten /t/s?
mean(asp.df$d[asp.df$Kons == "k" & asp.df$Bet == "be"]) > mean(asp.df$d[asp.df$Kons =
= "t" & asp.df$Bet == "be"])

# Berechnen Sie den Dauermittelwert  in 'Mutter'
mean(asp.df$d[asp.df$Wort == "Mutter"])

# Zeigen Sie alle Beobachtungen von unbetonten /t/s, für die
# die Dauer zwischen 100 und 120 ms liegt.
asp.df[asp.df$Bet == "un" & asp.df$d > 100 & asp.df$d < 120,]

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

# Wieviele veschiedene Versuchspersonen gibt es?
length(unique(dip.df$Vpn))

# Tabellieren Sie die Diphthonge
table(dip.df$V)

# Was ist der Dauermittelwert aller /aU/ Diphthonge?
mean(dip.df$d[dip.df$V == "aU"])

# Wieviele /aU/ Diphthonge gibt es mit einer Dauer größer
# als der Dauermittelwert der /aI/ Diphthonge?
sum(dip.df$d[dip.df$V == "aU"] > mean(dip.df$d[dip.df$V == "aI"]))

# Lesen Sie den folgenden Data-Frame von veschiedenen Vokalen ein:
v.df = read.table(file.path(pfadu, "vlax.txt"))

# Wieviele Beobachtungen gibt es? Und wieviele Variablen?
```

```r
dim(v.df)
# oder Beobachtungen:
nrow(v.df)
# Variablen:
ncol(v.df)

# Zeigen Sie alle Beobachtungen mit einem /I/ (Variable V)
v.df[v.df$V == "I",]

# Wieviele /I/ oder /E/ Vokalen haben einen F2 von weniger als 1800 Hz?
sum((v.df$V == "I" | v.df$V == "E") & v.df$F2 < 1800)

# es gibt übrigens eine andere Syntax für
# x == "a" | x == "b" | x ==  "c" |  x == "d"
# und die ist:
# x %in% c("a", "b", "c", "d")
# daher auch
sum(v.df$V %in% c("I", "E") & v.df$F2 < 1800)

# Tabellieren Sie die Vokale (Funktion table()) für die F1 größer als 400 Hz
# und F2 weniger als 1600 Hz ist.
table(v.df$V[v.df$F1 > 400 & v.df$F2 < 1600])

# Zeigen Sie die Wörter, für die F3 - F2 von /I/ Vokalen größer ist
# als 900 Hz
v.df$Wort[v.df$F3 - v.df$F2 > 900]
```