```
#Aufgabe:
#Machen Sie sich mit folgenden Datenstrukturen in R vertraut
#a) Vektoren
#b) Arrays
#c) Matrizen
#d) Listen
#e) Data Frames
#f) Faktoren

#Lösung

#Vectors
a <- c(1,2,5.3,6,-2,4) # numeric vector
b <- c("eins","zwei","drei") # character vector
c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) #logical vector

#Indexing vectors to extract elements
a[c(2)]
b[c(1:2)]

#Matrix
#General definiton
#mymatrix <- matrix(vector, nrow=r, ncol=c, byrow=FALSE,
#dimnames=list(char_vector_rownames, char_vector_colnames))

#1. Example
erste_matrix<-matrix(1:10,nrow=2,ncol=5)
erste_matrix

#2. Example
cells <- c(1,26,24,68)
rnames <- c("R1", "R2")
cnames <- c("C1", "C2")
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,
  dimnames=list(rnames, cnames))

#3. Example
x <- matrix(1:20,nrow=4,ncol=5) # Generate a 4 by 5 Matrix.
x

#Extract elements X[1,3], X[2,2] and X[3,1] as a vector structure, and
#Replace these entries in the array X by zeroes.

i <- matrix(c(1:3,3:1),nrow=3,ncol=2)
i

x[i] # Extract those elements

x[i] <- 0 # Replace those elements by zeros.
x



#Arrays
#Arrays are similar to matrices but can have more than two dimensions.
#See help(array) for details.
```

```
#Listen
#An ordered collection of objects (components).
#A list allows you to gather a variety of (possibly unrelated) objects
under one name.

# example of a list with 4 components -
# a string, a numeric vector, a matrix, and a scaler
w <- list(name="Fred", mynumbers=a, mymatrix=x, age=5.3)
w

#Identify elements of a list using the [[]] convention.

w[[2]] # 2nd component of the list



#Data Frames
A dataframe is more general than a matrix,
#in that different columns can have different modes
#(numeric, character, factor, etc.).
#This is similar to SAS and SPSS datasets.

d <- c(1,2,3,4)
e <- c("red", "white", "red", NA)
f <- c(TRUE,TRUE,TRUE,FALSE)
mydata <- data.frame(d,e,f)
names(mydata) <- c("ID","Color","Passed") # variable names

#There are a variety of ways to identify the elements of a dataframe .

mydata[1:2] # columns 1,2 of dataframe
mydata[c("ID","Color")] # columns ID and Color from dataframe
mydata$ID # variable ID in the dataframe



#Factors
#Tell R that a variable is nominal  by making it a factor.
#The factor stores the nominal values as a vector of integers in the range
[1...k]
#(where k is the number of unique values in the nominal variable),
#and an internal vector of character strings (the original values) mapped
to these integers.

# variable gender with 20 "male" entries and
# 30 "female" entries
gender <- c(rep("male",20), rep("female", 30))
gender <- factor(gender)
gender
# stores gender as 20 1s and 30 2s and associates
# 1=female, 2=male internally (alphabetically)
# R now treats gender as a nominal variable
summary(gender)

An ordered factor is used to represent an ordinal variable.

# variable rating coded as "large", "medium", "small'
rating<-c("large", "medium", "small')
rating <- ordered(rating)
```

```
# recodes rating to 1,2,3 and associates
# 1=large, 2=medium, 3=small internally
# R now treats rating as ordinal

#R will treat factors as nominal variables and ordered factors as ordinal
variables
#in statistical proceedures and graphical analyses.
#You can use options in the factor( ) and ordered( ) functions
#to control the mapping of integers to strings (overiding the alphabetical
ordering).
```