## Section 2 :Data Import and Export

##############################################################################

## Task 1 ##

##############################################################################

# a) Prepare the Excel file „people.xlsx“ for import in R.

# in Excel: open "people.xlsx", save as csv --> "people.csv", ignore and

# cofirm error messages and warnings.Fehlermeldungen ignorieren

# Start R. Change to the directory you want to work in. Then:

data = read.delim("../Data\_2/people.csv",sep=";",dec=".",header=TRUE,

row.names = 1)

# b) b) Import the file to R and make sure that everything is ok. # Check the data:

data

str(data)

# c) c) Name the rows with ascending numbers.

rownames(data) = seq(from=1,to=10,by=1)

# display to check:

data

# d) d) Change the name of the column „size“ to „SIZE“.

colnames(data)[1] = "SIZE" # "size" ist der erste Eintrag!

# display to check:

data

# e) e) Optional: Sort the whole data.frame in R by decreasing “SIZE”. There is a useful command for doing this – find it with google. Recommendation: in your later work this may be easier directly in Excel.

# The command to be used is "order". It gives indices of sorting:

sorted\_positions = order(data$SIZE,decreasing = TRUE);

data\_resorted = data[sorted\_positions,]

data\_resorted

# Think about: Why can we not directly use "sort"?

# f) f) Export the sorted dataset to „people\_sorted.txt“.

write.table(data\_resorted,file="people\_sorted.txt",

sep=";",dec="." , quote=FALSE , row.names=FALSE)

##############################################################################

## Task 2 ##

##############################################################################

# 2. a) Import the file „colors.csv“ to R. Convert the data.frame into a matrix (as.matrix). Now the entries should not be factors – check this.

# Check the file with a text editor to understand its structure!

cols = read.delim("../Data\_2/colors.csv",sep=";",dec=".",header=FALSE)

#Careful: the variablename "colors" would not work, since this name

# is already occupied by an R command.

cols

str(cols)

# Transformation to matrix format:

cols\_matrix = as.matrix(cols)

str(cols\_matrix) # now characters, not factors.

# b) b) How many different colors are present in the dataset? Hint: Use the matrix version of the data and remember the commands for vectors to avoid double entries and for determining the length of a vector.

cols\_vec = as.vector(cols\_matrix)

unique( cols\_vec ) # streicht Mehrfachnennungen

length( unique(cols\_vec) )

# c) At what position do you find the color „pink“? Try the command “which” in combination with “arr.ind=TRUE”. What is the difference to “arr.ind=FALSE”?

which(cols\_matrix=="pink",arr.ind=TRUE)

which(cols\_matrix=="pink") #considers the matrix a 1-dimensional vector.

# pink appears on the 32 position in this long vector.

which(cols\_vec=="pink")# the same as above with the vector from b).

##############################################################################

## Task 3 ##

##############################################################################

# 3. a) Import the dataset „drinks.txt” to R. Check for correctness.

drinks = read.delim("../Data\_2/drinks.txt",sep=" ",dec=",",header=TRUE)

drinks

str(drinks)

# b) How many different values do you find with Rivella, how many with Cola?

length( unique(drinks$rivella))

length( unique(drinks$cola) )

# c) Name the first 10 rows „day\_1“, the others „day\_2“. Try this first with the command “rownames”. This will not work. Why? What could be alternatives?

name1 = rep("day\_1",10) # the first 10 rows

name2=rep("day\_2",length(drinks$rivella)-10) # the rest of the rows

name=c(name1,name2) # combine the names.

name #check

rownames(drinks) = name # and assign them to the rownames of "drinks".

# This did not work. R needs to have UNIQUE rownames.

# alternatively: the days are built into the data set

# as an additional column:

drinks$test\_day = as.factor(name)

drinks

str(drinks)

##############################################################################

## Task 4 ##

##############################################################################

# 4. a) Import the dataset „dry\_fruits.txt“ to R and check for correctness.

dfruits = read.delim("../Data\_2/dry\_fruits.txt",sep=";",dec=".",header=TRUE)

dfruits

str(dfruits)

# b) The first column carries a typo: change “drie” to “dry” in R.

colnames(dfruits)[1] = "dry"

dfruits

# c) Add a column with name „dummy“, consisting only of values „5“.

dfruits$dummy = rep(5,length(dfruits$peach))

# also this would work: dfruits$dummy = rep(5,length(dfruits$dry))

dfruits

# d) Export the data from c) to „dry\_fruits\_new.txt“.

write.table(dfruits,file="dry\_fruits\_new.txt",

sep=";",dec="." , quote=FALSE , row.names=FALSE)

##############################################################################

## Task 5 ##

##############################################################################

# 5. a) Import the dataset „pets.txt“ to R and check for correctness.

pets = read.delim("../Data\_2/pets.txt",sep=":",dec=",",header=TRUE)

pets

str(pets)

head(pets)

# b) How many animals are black? (assess this question with programming)

length( pets$color[pets$color=="black"] )

# or more simply:

sum(pets$color=="black")

# this would not work:

length(pets$color=="black")

# because the argument is a list of TRUE and FALSE with length of the

# full dataset:

pets$color=="black"

c) Add a row with values „brown“, 3.2, 5,3 to the end of the data set. What happens if you try to add a row with just values „brown“, 3.2? Why would this not work properly? pets[length(pets$color)+1,] = c("brown" , 3.2 , 5 , 3)

pets

# this behaves erroneous since every variable in the data.frame needs

# to have a value. In other words: all rows must have the same length.

pets[length(pets$color)+1,] = c("brown" , 3.2)

pets

## 3. Section: Datatypes, Packages and Functions

#############################################################################

## "my\_lib.R

## contains some functions for Exercice sheet 3.

#############################################################################

## gives back absolute values of x:

abs\_val = function(x){ return(abs(x)) }

## returns sign flipped x:

neg\_val = function(x){ return(-x) }

## returns x, except for elements of value == 1

all\_except\_1 = function(x){ return( x[x!=1] ) }

source("my\_lib.R")

ls() # what is in my workspace?

x = c(-1,1,2,3,-4,-3,2,0.5,0)

abs\_val(x)

neg\_val(x)

all\_except\_1(x)

##############################################################################

## Task 2 ##

#############################################################################

data = read.delim("../Data\_3/tomatoes.csv",sep=";",dec=".",header=TRUE)

str(data) # color should be a factor:

data$color = as.factor(data$color)

str(data)

# a) choice = data[data$temperature > 42,]

choice

# b) sum( data$water=="wet" )

# or (more complicated):

length(data$water[data$water=="wet"])

# or (see also later):

table(data$water)

# c)s selection = data$water=="humid" & data$fertilizer=="middle" & (data$color==2 | data$color==3)

sum(selection)

## 4. Section

##############################################################################

## Task 1 ##

##############################################################################

x = c(1,2,3,2,3,4,1,5,4,3,4,5,6,2,3,4,5,6,1000000,1000000,100)

str(x)

mean(x) #huge

median(x) # small

sd(x) # huge

range(x) # huge

quantile(x,probs=seq(0,1,0.25),type=3)

quantile(x,probs=seq(0,1,0.1),type=3)

hist(x)

##############################################################################

## Task 2 ##

##############################################################################

tomatoes = read.delim("../Files/tomatoes.csv",sep=";",dec=".",header=TRUE)

str(tomatoes)

#a)) What is the average weight of the tomatoes separated in colors?

tapply(tomatoes$weight,tomatoes$color,mean)

tomatoes$weight

tomatoes$color

?tapply

#b)What is the standard deviation of the diameters separated in water levels?

tapply(tomatoes$diameter,tomatoes$water,sd)

#c)What is the average weight separated in terms of colors and water levels? Use ftable to

# the NA.

tomatoes$weight

tomatoes$water

tomatoes$color

tapply(tomatoes$weight, list(tomatoes$water, tomatoes$color), mean)

ftable(tomatoes$water,tomatoes$color) # no occurencies in certain combinations...

#d)Change the factors in the dataset to numbers (careful: ordinal values may need setting of

#levels, see chapters 1 or 3). Where do you find correlations in this data set?

str(tomatoes)

t\_new = tomatoes

t\_new

t\_new$fertilizer # needs releveling.

t\_new$fertilizer = factor(tomatoes$fertilizer,levels=c("low","middle","high"))

t\_new$fertilizer = as.numeric(t\_new$fertilizer)

t\_new

t\_new$water # leveling ok.

t\_new$water = factor(tomatoes$water,levels=c("dry","humid","wet"))

t\_new$water = as.numeric(t\_new$water)

str(t\_new)

str(t\_new)

correlations = cor(t\_new,method="spearman")

correlations

library(ellipse)

plotcorr(correlations)

# diameter and weight seem to be clearly correlated...

#e)Are weight and diameter linearly correlated?

plot(tomatoes$diameter,tomatoes$weight)

regr = lm(tomatoes$weight~tomatoes$diameter) # careful order!!

regr # intercept -116.68, diameter 62.77

# careful: negative diameters (intercept) are not reasonable --> model limitations.

abline(regr,col=2,lwd=2)

cor(tomatoes$diameter,tomatoes$weight,method="pearson")

#high correlation, good quality of regression line.

################################################################

## Task 3 ##

##############################################################################

data = read.delim("../Data\_4/experiment.csv",sep=";",dec=".",header=TRUE)

str(data)

boxplot(data$measured~data$point)

title("measurements of experiment")

# make sure that the error\_bars.R script is in the current working directory:

source("error\_bars.R")

means = tapply(data$measured,data$point,mean)

std = tapply(data$measured,data$point,sd)

b = barplot(means,ylim=c(0,20))

error.bars(b,means,upper=std)

title("measurements with error bars of +/- 1 sd")

# in scatterplot

x = seq(1,9)

x

plot(x,means,ylim=c(0,20))

title("measurements with error bars of +/- 1 sd")

error.bars(x,means,upper=std)

##############################################################################

## Task 4 ##

##############################################################################

people = read.delim("../Data\_4/even\_more\_people.csv",sep=";",dec=".",header=TRUE)

str(people)

# a)

p\_rel = people[people$year<1995,]

str(p\_rel)

# b)

xtabs(~p\_rel$year)

barplot(xtabs(~p\_rel$year))

100\*prop.table( tabs(~p\_rel$year))

barplot(100\*prop.table(xtabs(~p\_rel$year)))

# c)

xtabs(~people$year + people$nationality)

barplot(xtabs(~people$year + people$nationality),legend.text=TRUE) # wrong way round...

xtabs(~people$nationality + people$year)

barplot(xtabs(~people$nationality + people$year),legend.text=TRUE) # wrong way round...

?barplot

# d)

prop.table(xtabs(~people$nationality + people$year),margin=2)

barplot(prop.table(xtabs(~people$nationality + people$year),margin=2),legend.text=TRUE)

## Mock Exam

# Solutions to the R exam on 29-10-2015

View(swiss)

# Question/Task 1: how many cases

nrow(swiss) # ==> 47 towns

# Question/Task 2: standardabweichung

sd(swiss$Fertility) # ==> 12.5

# Question/Task 3: make histogram and boxplot of values agriculture

hist(swiss$Agriculture)

boxplot(swiss$Agriculture)

# Question/Task 4: fraction of the catholic majority

Mayo=swiss$Catholic[swiss$Catholic>=50]

A=length(Mayo)

B=length(swiss$Catholic)

A/B # ==> 38.3%

# Question/Task 5: 5 middlesized values of infant mortality

sorted=sort(swiss$Infant.Mortality)

middlesized=sorted[22:26]

# Question/Task 6:

which(is.na(swiss$Examination))

# Question/Task 7:

#a)

data = read.delim("tomatoes.csv",sep=";",dec=".",header=TRUE) #use correct path

str(data) # color should be a factor:

data$color = as.factor(data$color)

str(data)

# b)

choice = data[data$temperature < 42,]

choice

#c)

mean(choice$diameter) #=> 6.12

# d)

sum( data$water=="dry" ) #=>6

# or (more complicated):

length(data$water[data$water=="dry"])

# or table(data$water)

# e)

selection = data$water=="humid" & data$fertilizer=="high" & (data$color==1 | data$color==3)

sum(selection) #=>2