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

# HELP #

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

?read.table

??read.table

help.search("data input")

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

# PACKAGES #

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

# which packages are installed?

library()

#installing packages

install.packages("akima")

#activating package

library(akima)

#help concerning individual package including list of commands

library(help=nlme)

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

# BASIC OPERATIONS

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

#how to create vectors

a<-1

a

a<-c(1,2,3)

a

b<-rep(1,10)

b

b<-rep(a,4)

b

b<-rep(a,each=4)

b

a<-seq(1,10)

a

a<-1:10

a

a<-seq(1,10,by=2)

a

a<-seq(1,10,length=20)

a

a<-seq(10,1)

a

seq(1:10)

a

b<-c(a,100)

b

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

# DATA IMPORT AND BASIC MANIPULATION WITH DATA SETS

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

#import data file into R

data<-read.table("http://user.mendelu.cz/drapela/Forest\_Biometry/data/plots.txt",sep='\t',header=TRUE)

#preview of first rows of the data table

head(data)

#column names

names(data)

#function summary - basic statististical properties of data frame (different fot quantitative and qualitative variables)

summary(data)

#excel-like table suitable for quick editing of data

fix(data)

## MANIPULATION WITH DATA

#data in row 63

data[63,]

#data in 4th column

data[,4]

#the same operation with column name

data$d13

#value in row 63 and column 8

data[63,8]

#values in rows 63 and 70

data[c(63,70),]

#values in rows 63 and 70 and colums 4 and 5

data[c(63,70),c(4,5)]

#values in rows 60-70 and colums 4-8

data[60:70,4:8]

# data type of dataframe "data" and variables

class(data)

class(data$d13)

class(data$species)

#ordering of dataframe according to selected variable

data[order(data$d13),]

data[order(data$d13,decreasing=T),]

#ordering of dataframe according to selected 2 variables

data[order(data$d13,data$h),]

#ordering of dataframe according to selected 2 variables and only for columns 4,5 and 8

data[order(data$d13,data$h),c(4,5,8)]

#saving the same ordered table into dataframe xx

xx<-data[order(data$d13,data$h),c(4,5,8)]

#adding 5 years to age of 63th tree but the result was not saved, only displayed

data[63,8] + 5

data[63,8]

#adding 5 years to age of 63th tree, result was saved

data[63,8] <- data[63,8] + 5

data[63,8]

#subtracting 5 years from age of 63th tree, result was saved

data[63,8] <- data[63,8] - 5

data[63,8]

#fitering data only for pine

data[data$species=='PINE']

#fitering data only for pine on plot 1424

data[data$species=='PINE'&data$id\_plot=='1424', ]

#saving values of age into new variable x

x <- data$age

x

#length of vector x

length(x)

#variables in workspace

ls()

#to make the variables accessible by name within the R session

attach(data)

# now we can access to variables directly without name of dataframe

age

h

d13

#detaching of variables

detach(data)

#variables cannot be called directly

age

h

d13

## BASIC STATISTICAL FUNCTIONS

mean(data$d13)

# mean for d13 and age

mean\_d\_age<-apply(data[,c(4,8)],2,mean)

mean\_d\_age

#mean of d13 only for spruce

mean(data[data$species=='SPRUCE',]$d13)

#add unique indentificator of individual tree to dataframe

data$id<-paste(data$id\_plot, data$id\_tree, sep='')

head(data)

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

# BASIC GRAPHS

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

#barplot

x<-seq(1,20,by=2) # data

labels<-letters[1:10] # labels

barplot(x,names.arg=labels, main="Bar plot", col=sample(colors(),10))

windows()

barplot(x,names.arg=labels, main="Bar plot", col="red")

#1000 random values from standardized normal distribution in two variables "x" and "y" (1000 values for each variable)

x<-rnorm(1000)

y<-rnorm(1000)

#scatter plot

plot(x,y, main="Scatter plot", xlab="x", ylab="y", xlim=c(-4,4), ylim=c(-4,9))

#add red line with intercept 1 and slope 0

abline(1,0, col="red")

#add blue vertical line at point x=1

abline(v = 1, col="blue")

#add green line with intercept 1 and slope 2

abline(1, 2, col="green")

#add additional points to graph

x1<-seq(-3, 3, by=0.1)

y1<-x1^2 - 3

points(x1,y1, pch=21, bg="blueviolet")

#histogram for x

hist(x, main = "Histogram - variable x" , xlab="x ", probability=TRUE, col="lightblue")

#add empirical density for x

lines(density(x), col="red")

# histogram for x and y in one plot

windows()

par(mfrow=c(1,2))

hist(x, main = "Histogram x", xlab="x ", probability=TRUE, col="lightblue")

lines(density(x), col="red")

hist(y, main = "Histogram y", xlab="y ", probability=TRUE, col="lightgreen")

lines(density(x), col="red")

#saving graphs to file with extension \*.png

png("histogram\_xy.png",width=800, height=600)