Laboratory 3: Exploring data with R

Felipe Bravo Oviedo
Universidad de Valladolid, Spain

In this Lab we will learn how to:

- Obtain descriptive statistics
- Generate basic plots to observe and compare the data
- Export a plot as image or pdf

We will use also the knowledge we acquire in our previous labs and the data set DATOS2.csv that we uploaded in the previous lab (see the document "Lab 01: An Introduction to R").
As usual, first we must define our working directory and load the data sets we will use.

```
# establishing the working directory and loading the data sets

setwd('C:/your_desired_working_directoryR')
data0<-read.csv('your_file.csv',  header=TRUE)
```

Obtain descriptive statistics

Descriptive statistics can help us to detect inconsistencies in our dataset and to explore quickly the data. Descriptive statistics can be classify as measures of central tendency (mode, mean, median…) or measures of spread (variance, range..) We can obtain the descriptive statistics (of a sample) one by one with the following code:

```
# establishing the working directory and loading the data sets

setwd('C:/datosR')
data2<-read.csv2('DATOS2.csv',  header=TRUE)

#  some measures of central tendency
# mean of variable Na
mean(data2$Na)
# median of variable Na
median(data2$Na)

# some measures of spread of a sample
# range of variable Na
range(data2$Na)
# sample variance
var(data2$Na)
# sample standard deviation
sd(data2$Na)
```

We should obtain the following:

```
> # mean of variable Na
> mean(data2$Na)
[1] 0.8432143
> # median of variable Na
> median(data2$Na)
[1] 0.33
>
> # some measures of spread of a sample
> # range of variable Na
> range(data2$Na)
[1] 0.05 4.95
> # sample variance
> var(data2$Na)
[1] 1.176267
> # sample standard deviation
> sd(data2$Na)
[1] 1.084558
```

Also it is possible to obtain the summary of the variable description by codig:

```
#mean, median, quartiles (25th  and 75th), min and max
summary(data2$Na)
```

```
> summary(data2$Na)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0500  0.1075  0.3300  0.8432  1.2250  4.9500 
```

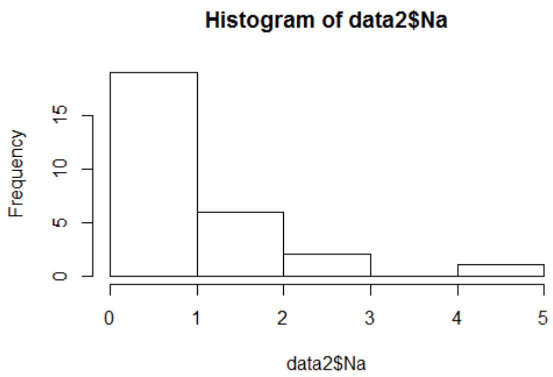
Generate basic plots to observe and compare the data

Histogram

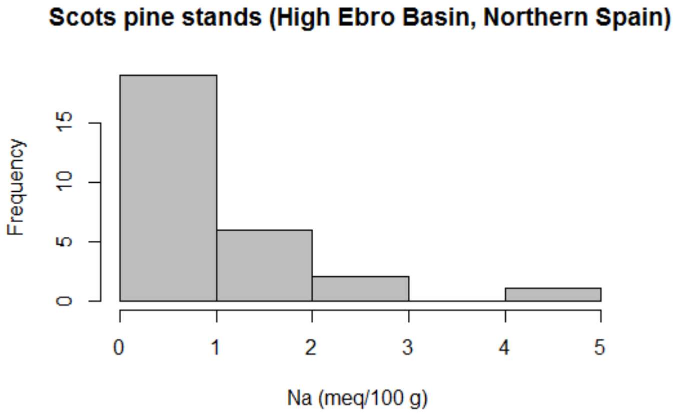
We can generate a histogram by using the hist( ) function indicating out variable of interest as follow:

```
# Creating a histogram for the variable Na in the dataset data2
hist(data2$Na)
```

We will obtain this histogram in the plot window of RStudio



We can improve the appearance of the histogram by adding some information with the different parameters within the hist() function (for a comprehensive information about the options to improve an histogram type ?hist into R and run it. For instance this histogram:



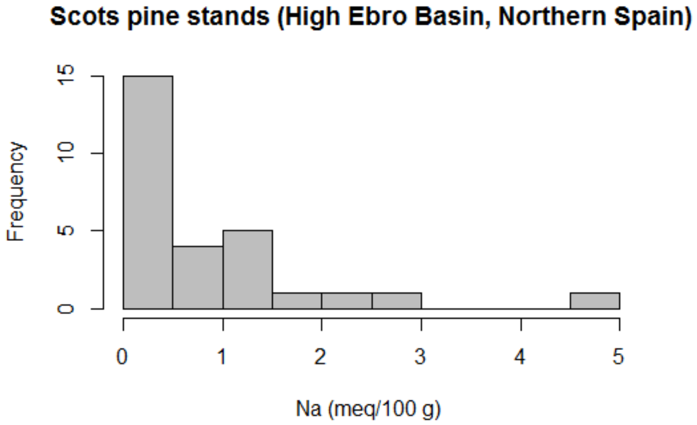
Can be obtained by specifying the colour the chart grey (col), create a title (main) and label the x-axis (xlab).

```
# Modify the previous histogram by using command that improve the visualisation
hist(data2$Na,  col= "grey", main="Scots pine stands (High Ebro Basin, Northern Spain)", xlab="Na (meq/100 g) ")
```

if we increase the number of classes in the x-axis we should include a new command (breaks) that will allow us to define the number of data breaks in the histogram.

```
# Modify the previous histogram by using command that improve the visualisation
hist(data2$Na, breaks=10, col= "grey", main="Scots pine stands (High Ebro Basin, Northern Spain)", xlab="Na (meq/100 g) ")
```

Then we will obtain this histogram:



Boxplot

Drawing a boxplot is a good option to show basic information from a set of variable (see the figure below for a boxtplot of a variable with normal distribution). Boxplot show the median, the IQR (interquartile range) defined as the difference between upper and lower quartiles (75th and 25th percentiles) also name as Q1 and Q3. In some cases, the boxplot shows the outliers of the distribution by plotting the observations with values smaller than Q1 − 1.5 IQR or greater than Q3 + 1.5 IQR.

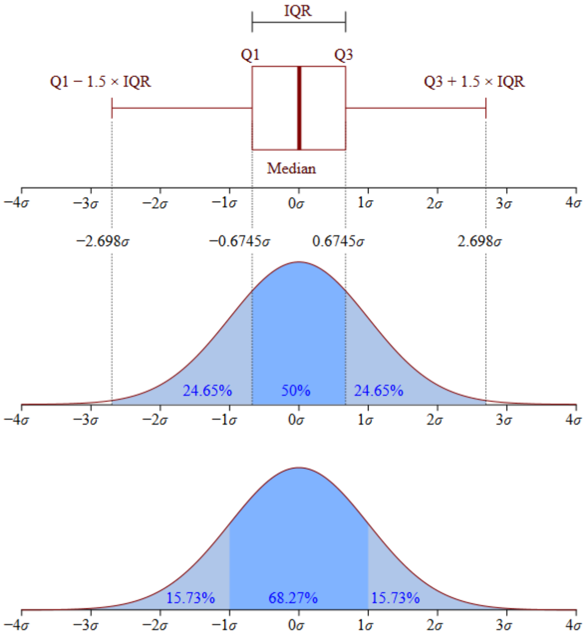
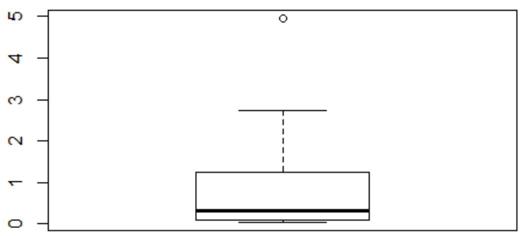


Figure obtained from Jhguch at en.wikipedia [CC BY-SA 2.5 (https://creativecommons.org/licenses/by-sa/2.5)], from Wikimedia Commons: https://commons.wikimedia.org/wiki/File%3ABoxplot_vs_PDF.svg

To generate a boxplot we must running the function *boxplot( )* for a variable or for a set of variables together.

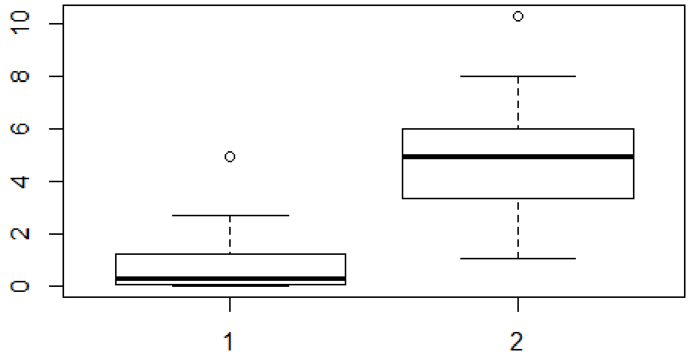
```
# boxplot of a variable
boxplot(data2$Na)
```

To obtain:



```
# boxplot of two variable
boxplot(data2$Na, data2$MO)
```

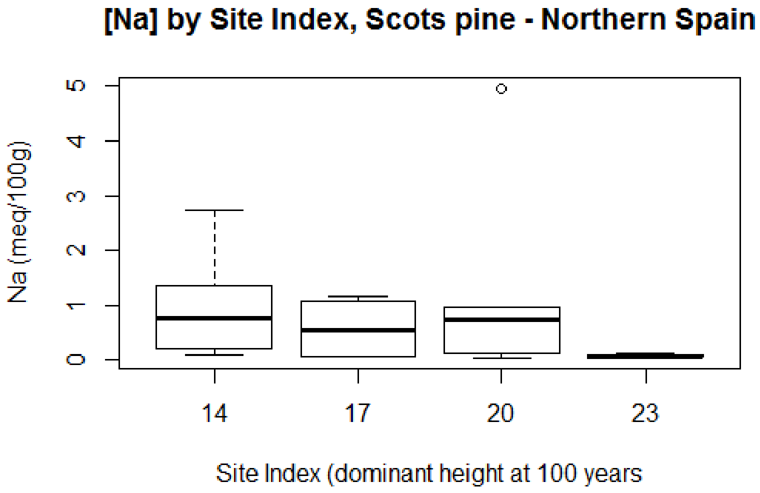
To obtain:



However, it can be more informative if we have the observations classified by a certain feature that we obtain the boxplot for each class. Now we will create the boxplot of our variable of interest (Na) for the different site index (SI) classes in the data sets. We should code as follow:

```
# boxplot by Site Index classes
boxplot(data2$Na ~ data2$SI, data = data2, xlab = "Site Index (dominant height at 100 years", ylab = "Na (meq/100g)", main = "[Na] by Site Index, Scots pine - Northern Spain")
```

To obtain the following boxplot:

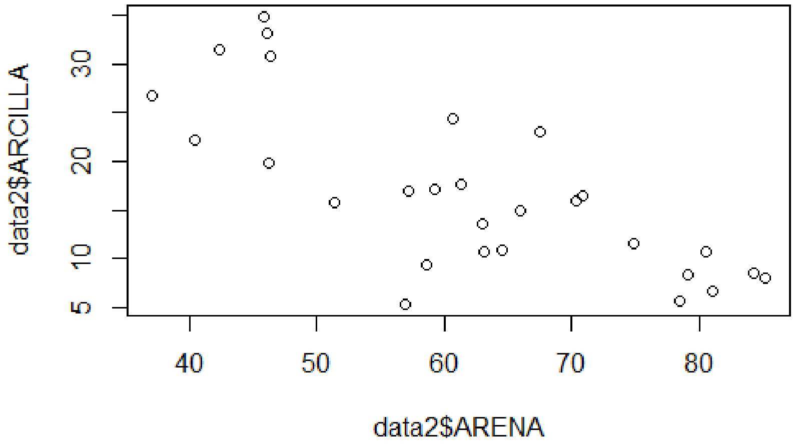


Scatter plot

Scatter plots can be generating by the plot ( ) function indicating the two variables of interest. Let's see how simple is by coding:

```
# Scatter plot witx the x-axis variable at the left of the comma
# and the y-axis variable at the right of the comma
plot(data2$ARENA,data2$ARCILLA)
```

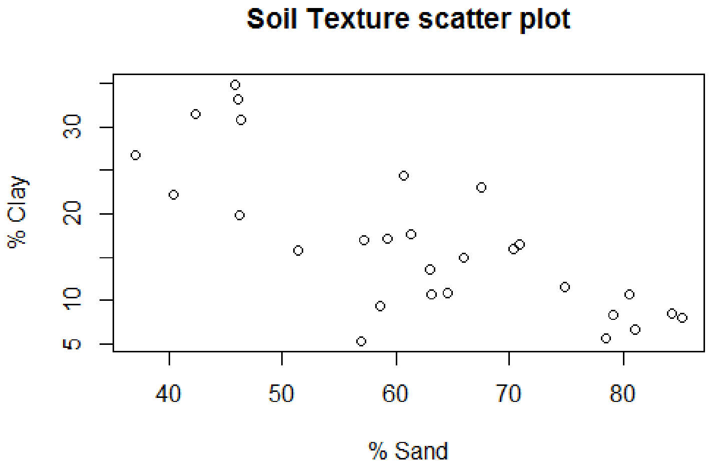
to obtain:



Also we can add title and axis labels:

```
# Improving the scatter plot
plot(data2$ARENA,data2$ARCILLA, main="Soil Texture scatter plot",
    xlab="% Sand", ylab="% Clay")
```

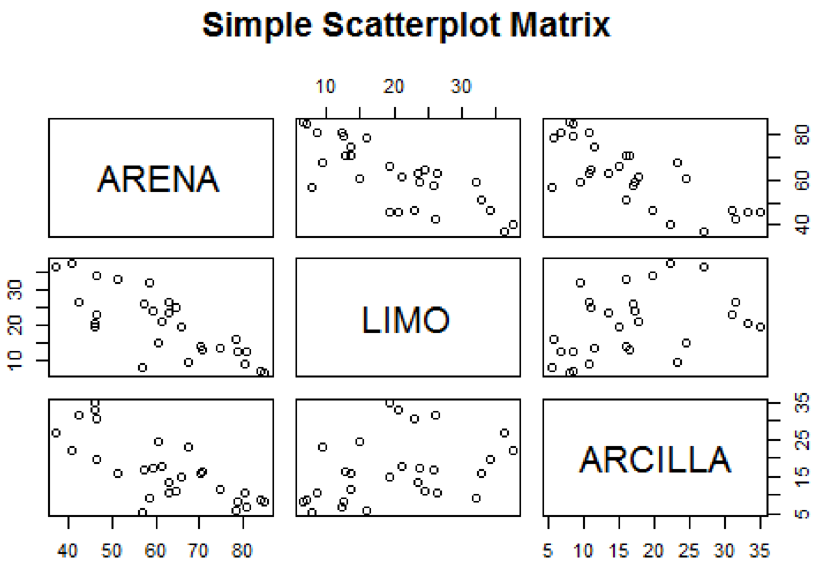
To obtain:



We can obtain a matrix of boxplot for a set of defined variables by using this code:

```
# Basic Scatterplot Matrix
pairs(~ARENA+LIMO+ARCILLA, data=data2, main="Simple Scatterplot Matrix")
```

To obtain:

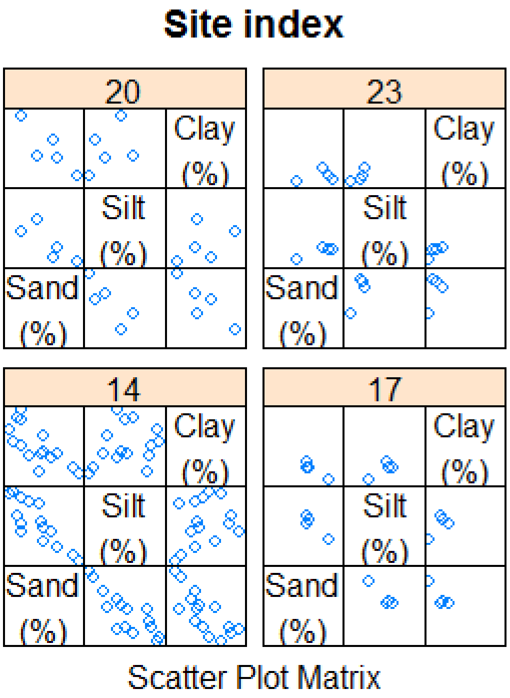


This scatterplot matrix can be improved by using the package lattice that will allow us to access to new features. Let's generate a matrix of scatterplots by site index classes.

```
# Scatterplot Matrices from the lattice Package
# Importing and loading the lattice package
if(!require(lattice))
   install.packages(lattice)
library(lattice)

# creating a matrix of scatterplots of texture variables by site index
splom(~data2[c(2, 3, 4)] | factor(data2$SI4),
         pscales = 0,
         varnames = c("Sand\n(%)", "Silt\n(%)", "Clay\n(%)"),
         main = "Site index",
         auto.key = list(columns = 1, title = "Site index"))
```

We will obtain the following scatterplot matrix:



Exporting images

In the plot window in RStudio you can choose to export the image as PDF (portable document format), as image file in different formats (PNG, JPEG, TIFF, BMP, Metafile, SVG or EPS) or copying to the clipboard as bitmap or metafile. The images can be obtained in a very high quality that can be used both in print documents or web pages.