Principal component analysis - Practice 1.1

José Luis Romero Béjar

2023-09-27

This brief guide is intended to familiarize the reader with the following:

- Loading and installing R packages.
- Loading data sets of different formats from R base installation and from local directories.
- Basic descriptive statistics.
- Graphical utils from *ggplot2* package.
- Deal with outliers: identification and making decisions.
- Principal component analysis: requirements, obtaining principal components, explained variance, appropriate number of principal components, graphical outputs, coordenates in the new reference system.

Loading packages and data sets

Loading and installing R packages

The following source code module is responsible for loading, if they are already installed, all the packages that will be used in this R session. While an R package can be loaded at any time when it is to be used, it is advisable to optimize its calls with this code chunk at the beginning.

Loading a package into an R session **requires it to be already installed**. If it is not, the first step is to run the sentence:

install.packages("name_of_the_library")

```
#########################################
# Loading necessary packages and reason #
#########################################

# This is an example of the first installation of a package
# Only runs once if the package is not installed
# Once it is installed this sentence has to be commented (not to run again)
# install.packages("summarytools")

# Package required to call 'freq' and 'descr' functions (descriptive statistics)
library(summarytools)

# Package required to call 'ggplot' function (graphical tools)
library(ggplot2)

# Package required to call 'ggarrange' function (graphical tools)
library(ggpubr)
```

```r
# Package required to call 'read.spss' function (loading '.spss' data format)
library(foreign)

# Package required to call 'read_xlsx' function (loading '.xlsx' data format)
library(readxl)

# Package required to load the data set 'RBGlass1'
library(archdata)

# Package required to call 'cortest.bartlett' function
library(psych)

# Package required to call 'fviz_pca_var, fviz_pca_ind and fviz_pca' functions
library(factoextra)

# Package required to call 'scatterplot3d' function
library(scatterplot3d)
```

Loading a data set from different formats

When loading a data set into an R session, it is recommended to save its structure in a data frame. This object (data.frame) is the best to work with data.

The next code chunk shows how to load different data set formats: *.sav (IBM SPSS)*, *.xlsx (Microsoft Excel)*, *.csv (comma separated values)* and *.txt or .dat (plain text)*. First, examples are performed with data sets available from the local working directory (it is relevant that these data sets are in the same directory of the current R working file). All these first examples work with the same data set with different format extension. Then, some examples of loading different data sets available in the base installation of R (running the sentence *data()* in the R console shows all available data sets in the package *dataset*) are performed.

The following code chunks show how to load these different formats.

IBM SPSS format *(.sav)*

```r
# Loading a .sav (IBM SPSS) file
# The output of this function is NOT a data.frame object
# Remember that package 'foreign' is required
data_spss<-read.spss("DB.sav", to.data.frame=TRUE)

# This sentence identifies the type of object that the identifier represents
class(data_spss)
```

```
## [1] "data.frame"
```

Excel format (*.xlsx*)

```r
# Loading a .xlsx (excel) file.
# The output of this function is already a data.frame object
# Remember that package 'readxl' is required
data_xlsx<-read_excel("DB.xlsx")

# This sentence identifies the type of object that the identifier represents
class(data_xlsx)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

Comma separated values (*.csv*)

```r
# Loading a .csv (comma separated values) file
# The output of this function is already a data.frame object
data_csv<-read.csv("DB.csv", header = TRUE,sep =";")

# This sentence identifies the type of object that the identifier represents
class(data_csv)
```

```
## [1] "data.frame"
```

Plain text (*.txt or .dat*)

```r
# Loading a .txt (plain text) file (for this example data are separated by tab)
# The output of this function is already a data.frame object
data_txt<-read.table("DB.dat", header = TRUE, sep="\t")

# This sentence identifies the type of object that the identifier represents
class(data_txt)
```

```
## [1] "data.frame"
```

Preloaded data sets

There are many data sets in different packages preloaded with the base installation of R. There are other packages (need to install) including interesting data sets.

Code chunk bellow shows a list of all the data sets in package *datasets* (preloaded with R.)

```r
# Data sets in a package
# This line has to be run directly in the R-console
data()
```

iris data set

```r
# Loading the data set 'iris'
data("iris")

# Loading 'iris' as data.frame in the new variable data_iris
data_iris<-iris

# This sentence identifies the type of object that the identifier represents
class(data_iris)
```

```
## [1] "data.frame"
```

RBGclass1 data set

The *RBGlass1* dataset is a database within the *archata* package that contains information on 11 chemical elements found in glass remains at two different locations (Mancetter and Leicester).

Hereafter, the problem referred by the data set *RBGlass1* is addressed along this guide as an example of pre-processing data and dimensionality reduction.

```r
# Loading the data set 'RBGclass1'
# Remember that package 'archdata' is required
data("RBGlass1")
```

```r
# Loading 'RBGlass1' as data.frame in the new variable data_RBGlass1
data_RBGlass1<-RBGlass1

# This sentence identifies the type of object that the identifier represents
class(data_RBGlass1)
```

```
## [1] "data.frame"
```

Basic descriptive statistics

In this section, a preliminary exploratory data analysis of the data set RBGlass1 is performed. For this purpose, if the variable is **quantitative**, the basic **numerical descriptive statistics** and a representation of its **histogram, density and boxplot** are shown. On the other hand, for the only **categorical** variable, *Site*, its **frequency table** and a **sector and bar diagram** are provided.

Exploring the data set

```r
# This line loads the variable names from this data.frame
# So that we can access by their name with no refer to the data.frame identifier
attach(data_RBGlass1)

# Retrieving the name of all variables
colnames(data_RBGlass1)
```

```
##  [1] "Site" "Al"   "Fe"   "Mg"   "Ca"   "Na"   "K"    "Ti"   "P"    "Mn"
## [11] "Sb"   "Pb"
```

```r
# Displaying a few records
head(data_RBGlass1, n=10)
```

```
##         Site   Al   Fe   Mg   Ca    Na    K   Ti    P   Mn   Sb   Pb
## 1  Mancetter 2.51 0.53 0.56 6.98 17.44 0.73 0.09 0.15 0.58 0.12 0.03
## 2  Mancetter 2.36 0.49 0.53 6.71 17.69 0.68 0.09 0.13 0.40 0.23 0.04
## 3  Mancetter 2.30 0.36 0.49 8.10 15.94 0.68 0.07 0.13 0.77 0.00 0.01
## 4  Mancetter 2.42 0.52 0.56 6.93 17.59 0.72 0.09 0.14 0.47 0.18 0.02
## 5  Mancetter 2.32 0.37 0.51 7.51 16.27 0.69 0.07 0.13 0.21 0.00 0.02
## 6  Mancetter 2.34 0.56 0.52 6.10 18.61 0.69 0.10 0.11 0.30 0.32 0.03
## 7  Mancetter 2.50 0.46 0.50 6.83 17.46 0.79 0.08 0.15 0.40 0.06 0.02
## 8  Mancetter 2.47 0.53 0.55 6.55 18.55 0.75 0.09 0.12 0.35 0.23 0.04
## 9  Mancetter 2.41 0.67 0.62 6.18 18.33 0.81 0.12 0.14 0.52 0.31 0.07
## 10 Mancetter 2.64 0.50 0.63 7.76 15.66 0.63 0.08 0.16 0.21 0.00 0.01
```

```r
# Displaying basic descriptives of variable 'Al'
summary(Al)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.160   2.320   2.420   2.417   2.500   2.770
```

```r
# Displaying frequency table of variable 'Site'
# Absolute
table(Site)
```

```
## Site
## Leicester Mancetter
##        59        46
```

```r
# Relative
round(prop.table(table(Site)),2)
```

```
## Site
## Leicester Mancetter
##      0.56      0.44
```

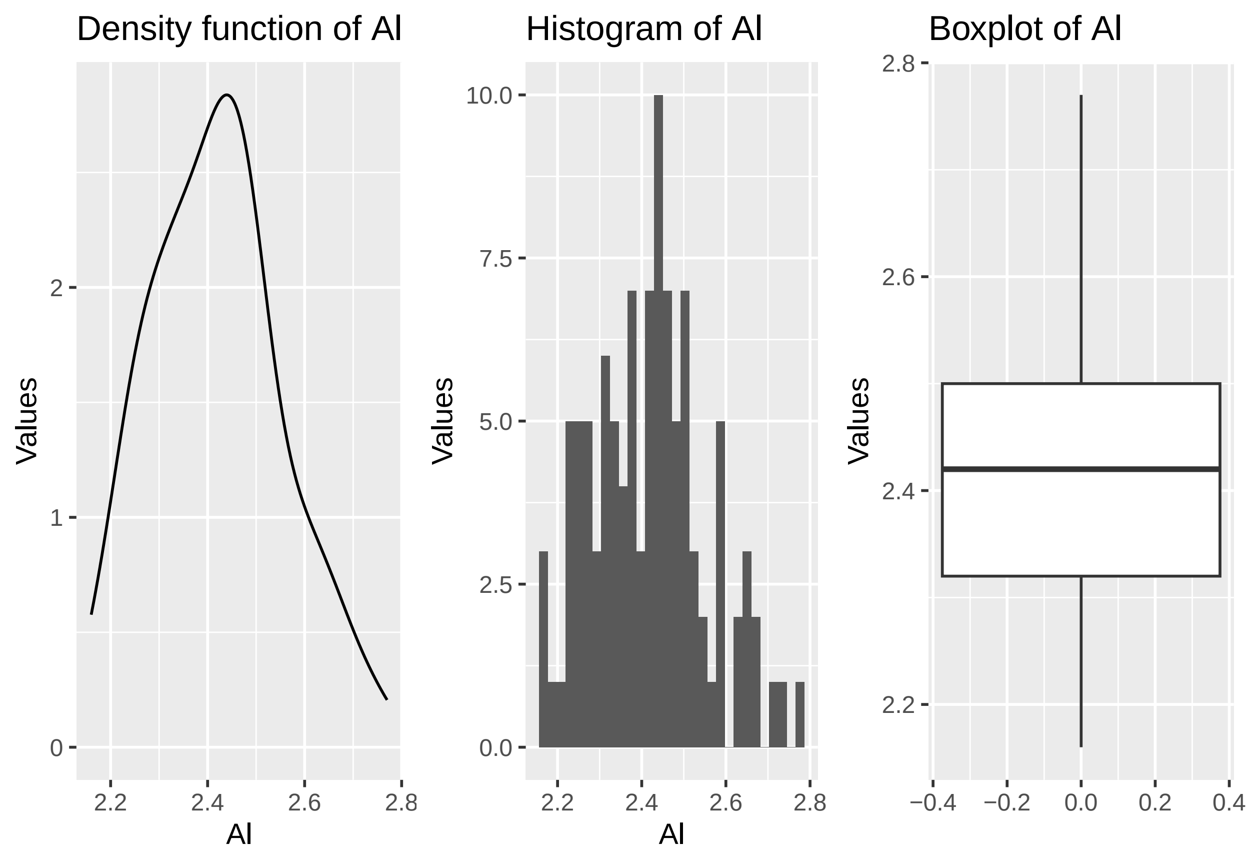
Basic descriptive statistics of quantitative variables

Al - Aluminum

```r
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Al)
```

```
## Descriptive Statistics
## Al
## N: 105
##
##                           Al
## ----------------- --------
##              Mean     2.42
##           Std.Dev     0.13
##               Min     2.16
##                Q1     2.32
##            Median     2.42
##                Q3     2.50
##               Max     2.77
##               MAD     0.13
##               IQR     0.18
##                CV     0.06
##          Skewness     0.30
##       SE.Skewness     0.24
##          Kurtosis    -0.39
##           N.Valid   105.00
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Al))+geom_density()+
  labs(title = "Density function of Al",x="Al",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Al))+geom_histogram()+
  labs(title = "Histogram of Al",x="Al",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Al))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Al",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



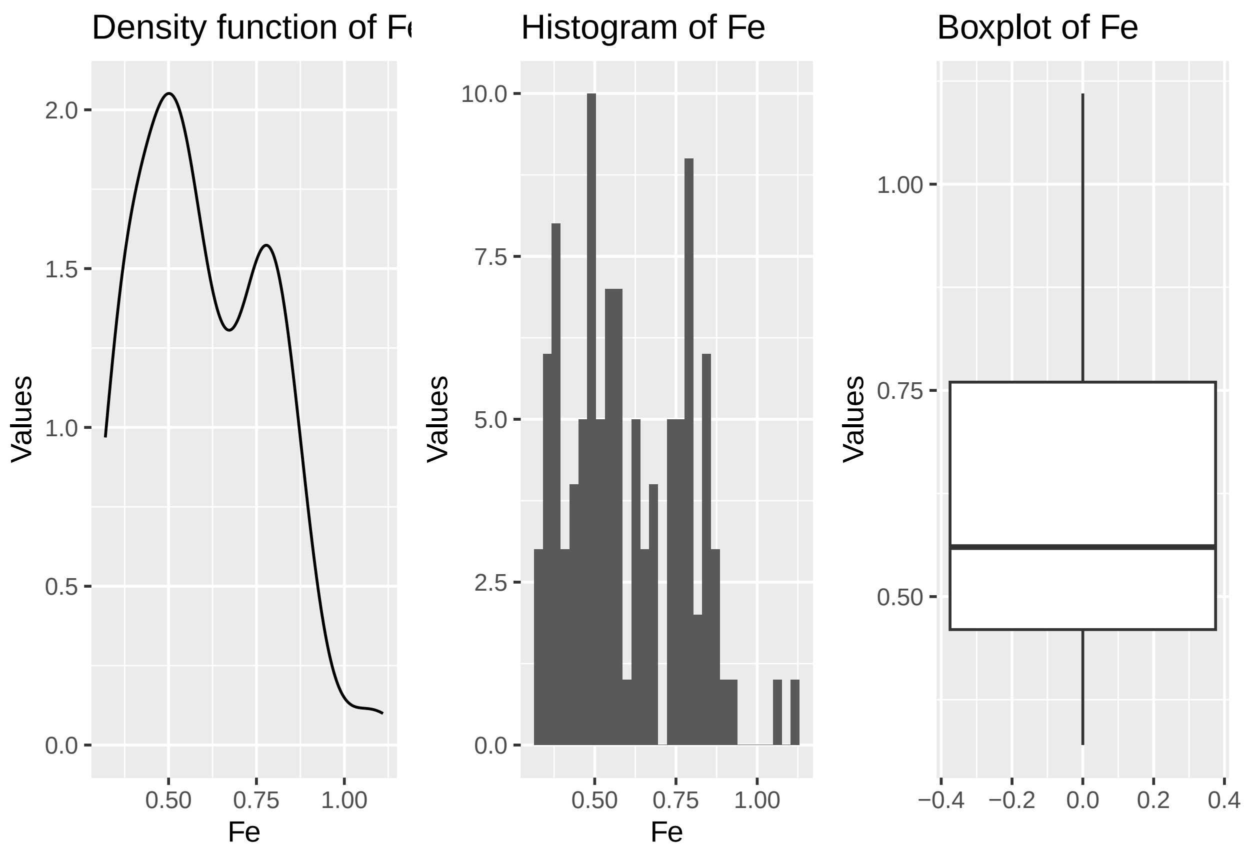
Fe - Iron

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Fe)
```

```
## Descriptive Statistics
## Fe
## N: 105
##
##                           Fe
## ----------------- --------
##              Mean     0.60
##           Std.Dev     0.18
##               Min     0.32
##                Q1     0.46
##            Median     0.56
##                Q3     0.76
##               Max     1.11
##               MAD     0.21
##               IQR     0.30
##                CV     0.30
##          Skewness     0.39
##       SE.Skewness     0.24
##          Kurtosis    -0.70
##           N.Valid   105.00
```

```
##          Pct.Valid   100.00
```

```
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Fe))+geom_density()+
  labs(title = "Density function of Fe",x="Fe",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Fe))+geom_histogram()+
  labs(title = "Histogram of Fe",x="Fe",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Fe))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Fe",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



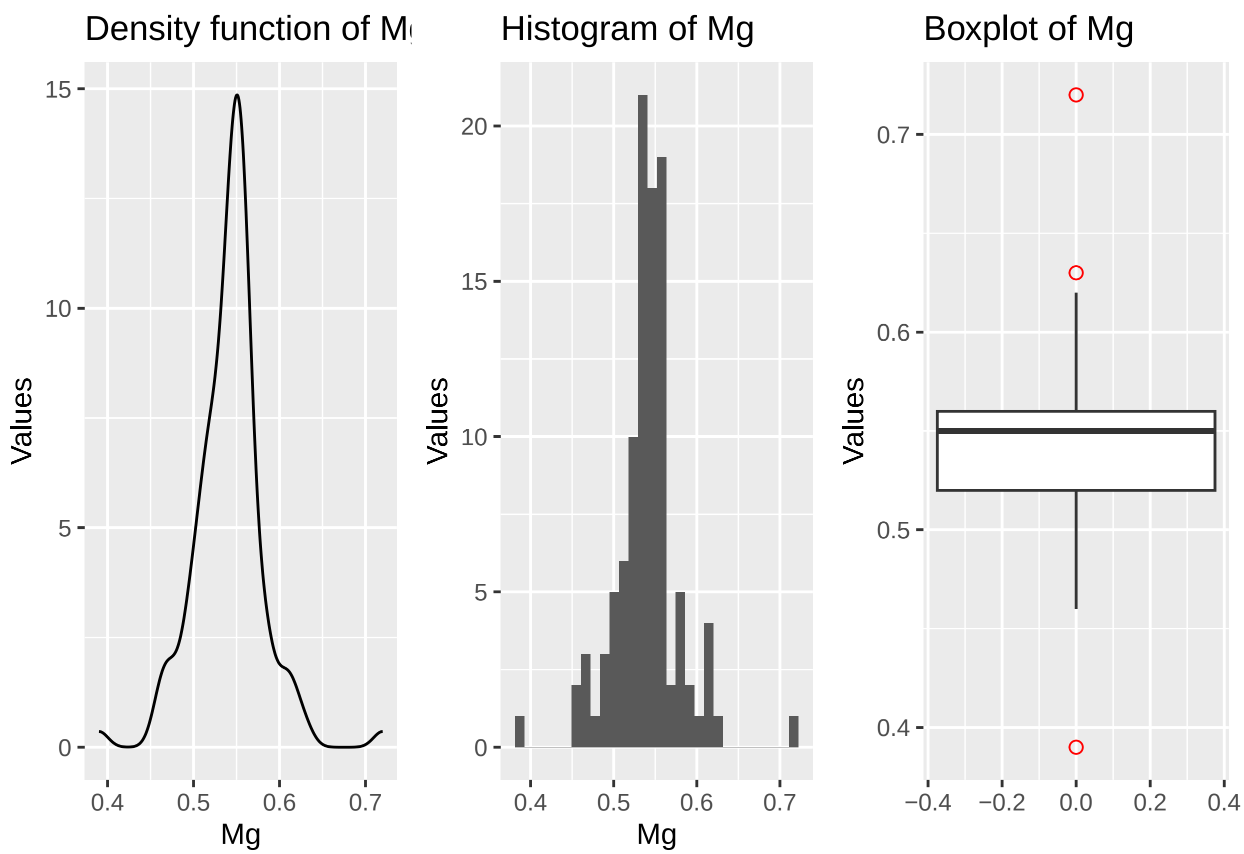
Mg - Magnesium

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Mg)
```

```
## Descriptive Statistics
## Mg
## N: 105
```

```
##
##                            Mg
## ----------------- --------
##              Mean      0.54
##           Std.Dev      0.04
##               Min      0.39
##                Q1      0.52
##            Median      0.55
##                Q3      0.56
##               Max      0.72
##               MAD      0.03
##               IQR      0.04
##                CV      0.07
##          Skewness      0.25
##       SE.Skewness      0.24
##          Kurtosis      4.17
##           N.Valid    105.00
##         Pct.Valid    100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Mg))+geom_density()+
  labs(title = "Density function of Mg",x="Mg",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Mg))+geom_histogram()+
  labs(title = "Histogram of Mg",x="Mg",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Mg))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mg",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



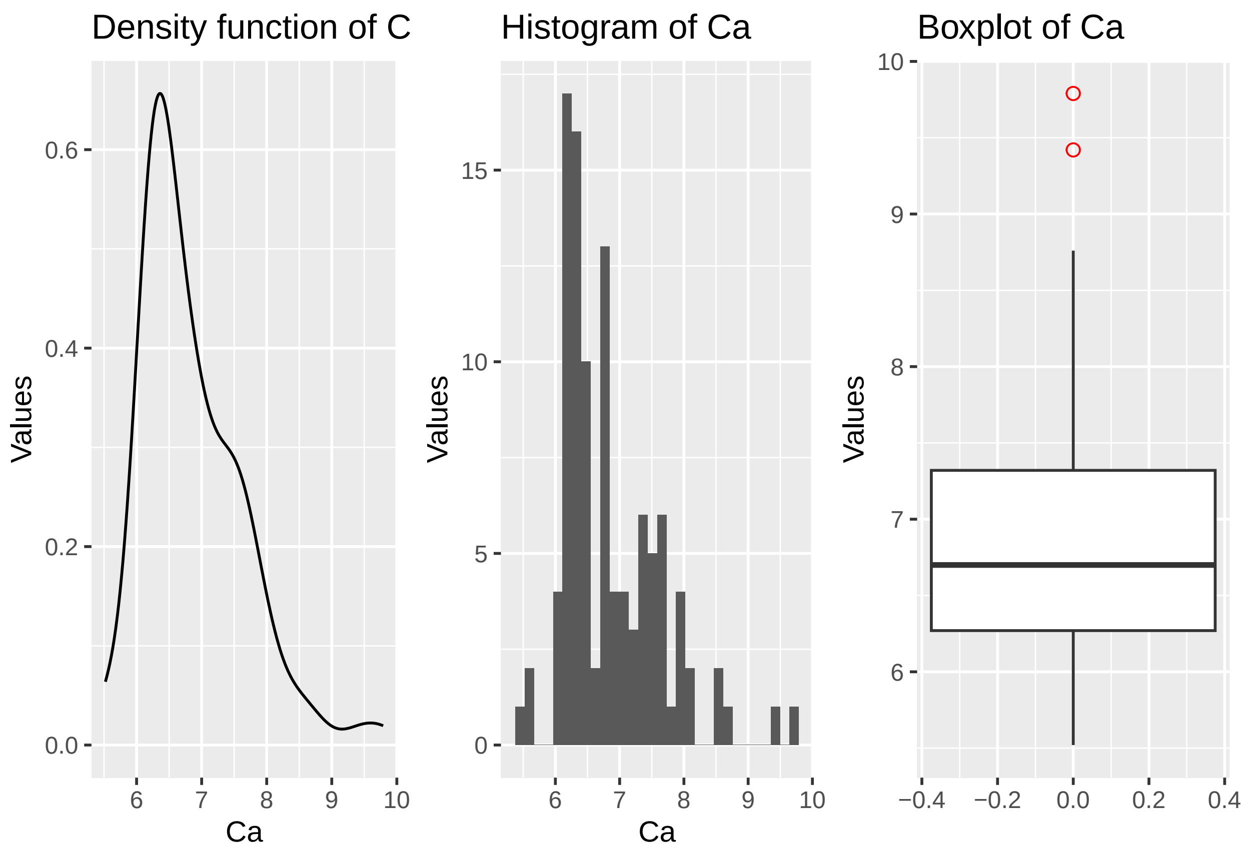
Ca - Calcium

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Ca)
```

```
## Descriptive Statistics
## Ca
## N: 105
##
##                                Ca
## ----------------- --------
##              Mean      6.85
##           Std.Dev      0.77
##               Min      5.52
##                Q1      6.27
##            Median      6.70
##                Q3      7.32
##               Max      9.79
##               MAD      0.70
##               IQR      1.05
##                CV      0.11
##          Skewness      1.19
##       SE.Skewness      0.24
##          Kurtosis      1.71
##           N.Valid    105.00
```

```
##         Pct.Valid   100.00
```

```
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Ca))+geom_density()+
  labs(title = "Density function of Ca",x="Ca",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Ca))+geom_histogram()+
  labs(title = "Histogram of Ca",x="Ca",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Ca))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ca",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



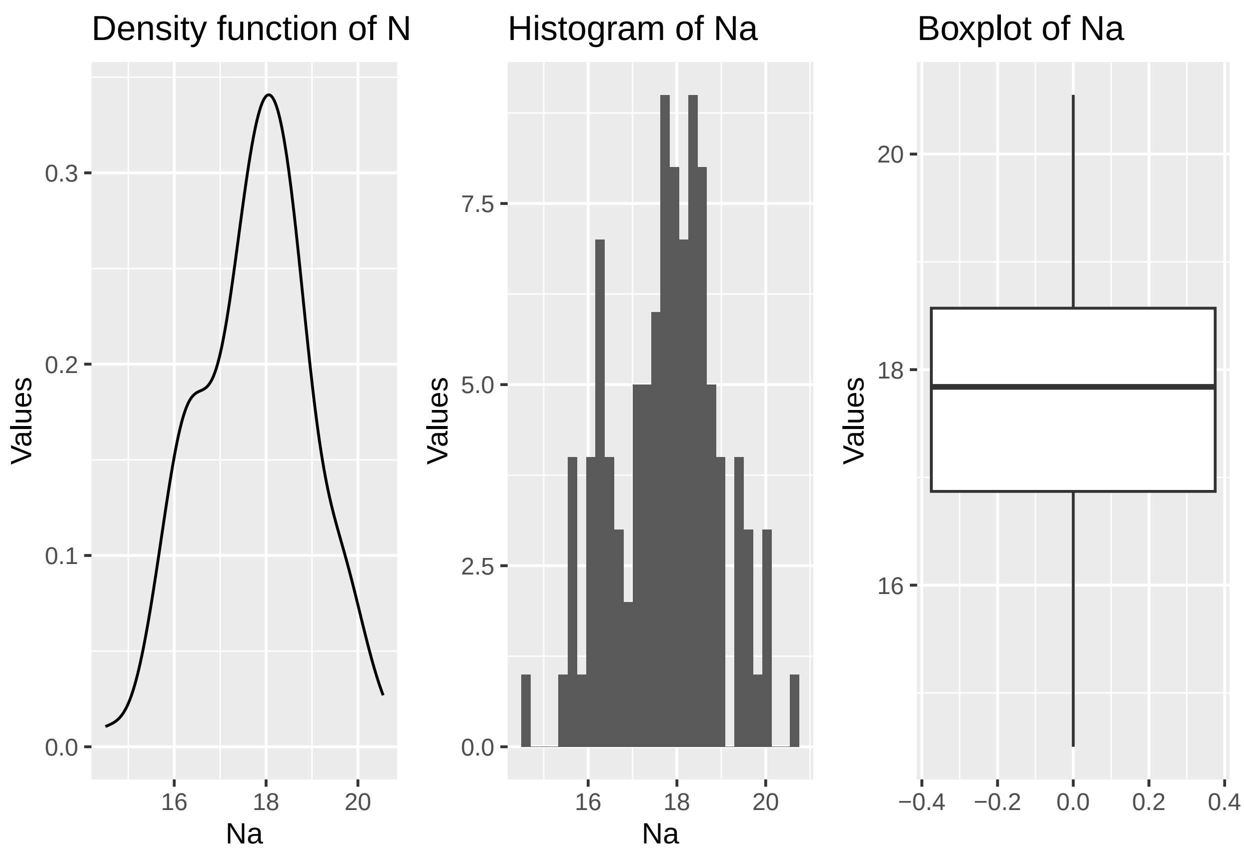
Na - Sodium

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Na)
```

```
## Descriptive Statistics
## Na
## N: 105
```

```
##
##                           Na
## ----------------- --------
##              Mean    17.76
##           Std.Dev     1.19
##               Min    14.50
##                Q1    16.87
##            Median    17.84
##                Q3    18.57
##               Max    20.55
##               MAD     1.11
##               IQR     1.70
##                CV     0.07
##          Skewness    -0.13
##       SE.Skewness     0.24
##          Kurtosis    -0.38
##           N.Valid   105.00
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Na))+geom_density()+
  labs(title = "Density function of Na",x="Na",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Na))+geom_histogram()+
  labs(title = "Histogram of Na",x="Na",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Na))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Na",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



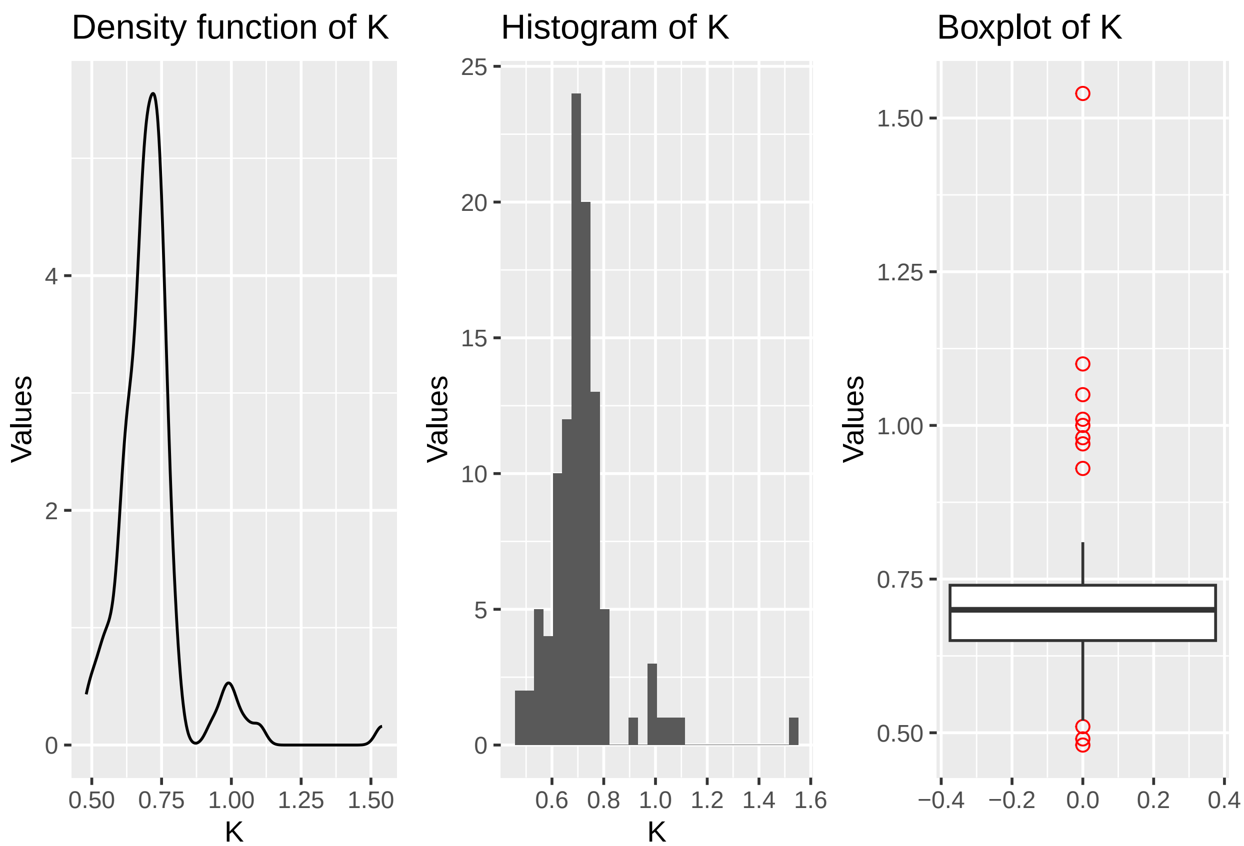
K - Potassium

```r
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(K)
```

```
## Descriptive Statistics
## K
## N: 105
##
##                               K
## ----------------- --------
##              Mean      0.71
##           Std.Dev      0.13
##               Min      0.48
##                Q1      0.65
##            Median      0.70
##                Q3      0.74
##               Max      1.54
##               MAD      0.07
##               IQR      0.09
##                CV      0.19
##          Skewness      2.70
##       SE.Skewness      0.24
##          Kurtosis     12.95
##           N.Valid    105.00
```

```
##          Pct.Valid   100.00
```

```
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=K))+geom_density()+
  labs(title = "Density function of K",x="K",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=K))+geom_histogram()+
  labs(title = "Histogram of K",x="K",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=K))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of K",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



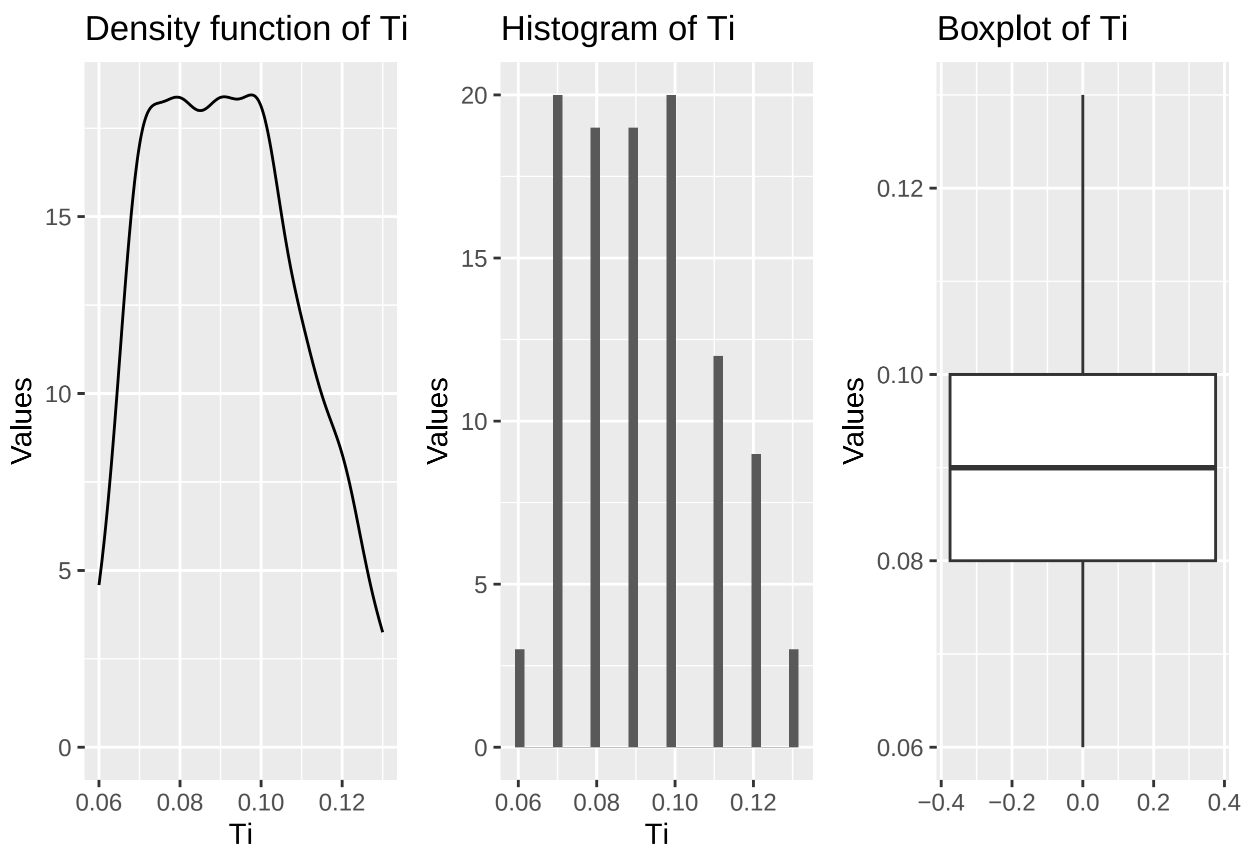
Ti - Titanium

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Ti)
```

```
## Descriptive Statistics
## Ti
## N: 105
```

```
##
##                          Ti
## ----------------- --------
##              Mean     0.09
##           Std.Dev     0.02
##               Min     0.06
##                Q1     0.08
##            Median     0.09
##                Q3     0.10
##               Max     0.13
##               MAD     0.01
##               IQR     0.02
##                CV     0.19
##          Skewness     0.27
##       SE.Skewness     0.24
##          Kurtosis    -0.83
##           N.Valid   105.00
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Ti))+geom_density()+
  labs(title = "Density function of Ti",x="Ti",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Ti))+geom_histogram()+
  labs(title = "Histogram of Ti",x="Ti",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Ti))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ti",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



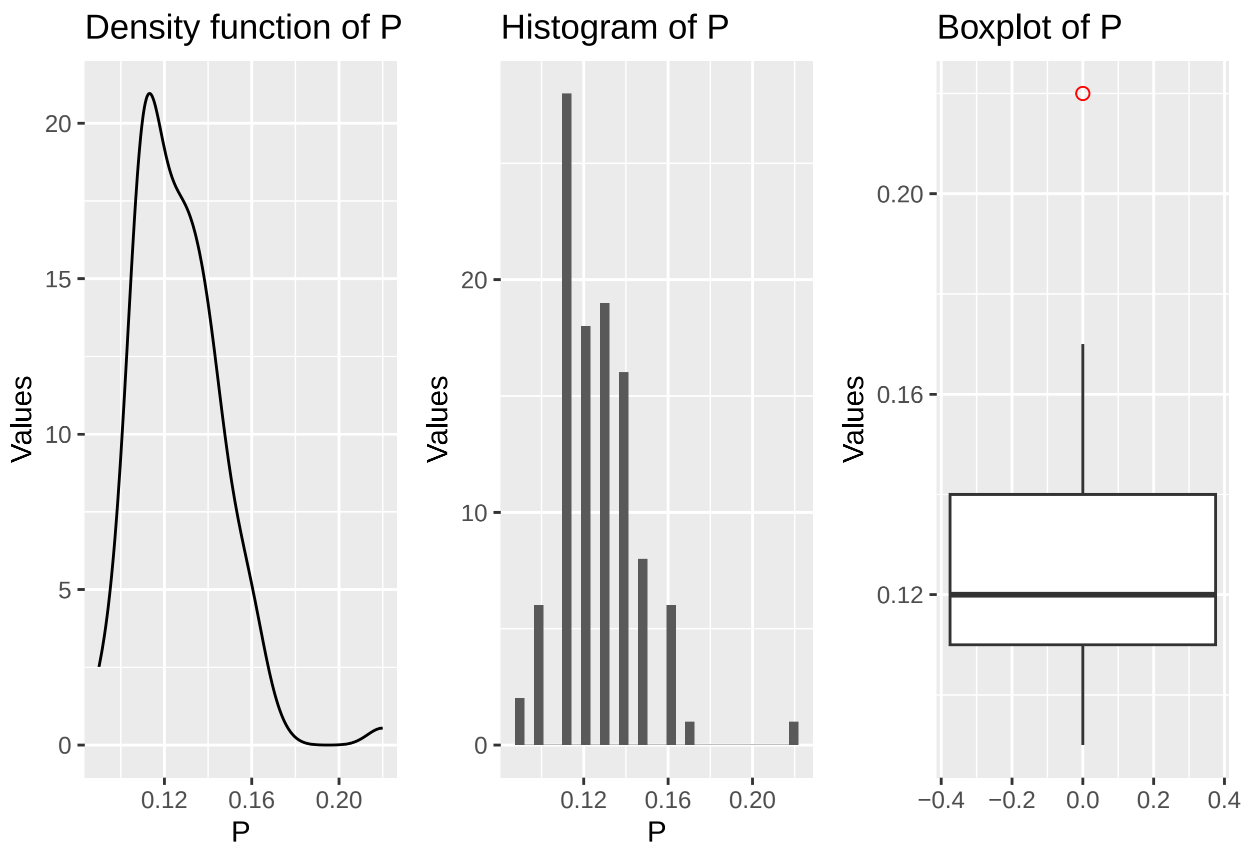
P - Phosphorus

```r
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(P)
```

```
## Descriptive Statistics
## P
## N: 105
##
##                                P
## ----------------- --------
##              Mean      0.13
##           Std.Dev      0.02
##               Min      0.09
##                Q1      0.11
##            Median      0.12
##                Q3      0.14
##               Max      0.22
##               MAD      0.01
##               IQR      0.03
##                CV      0.16
##          Skewness      1.17
##       SE.Skewness      0.24
##          Kurtosis      3.33
##           N.Valid    105.00
```

```
##          Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=P))+geom_density()+
  labs(title = "Density function of P",x="P",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=P))+geom_histogram()+
  labs(title = "Histogram of P",x="P",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=P))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of P",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



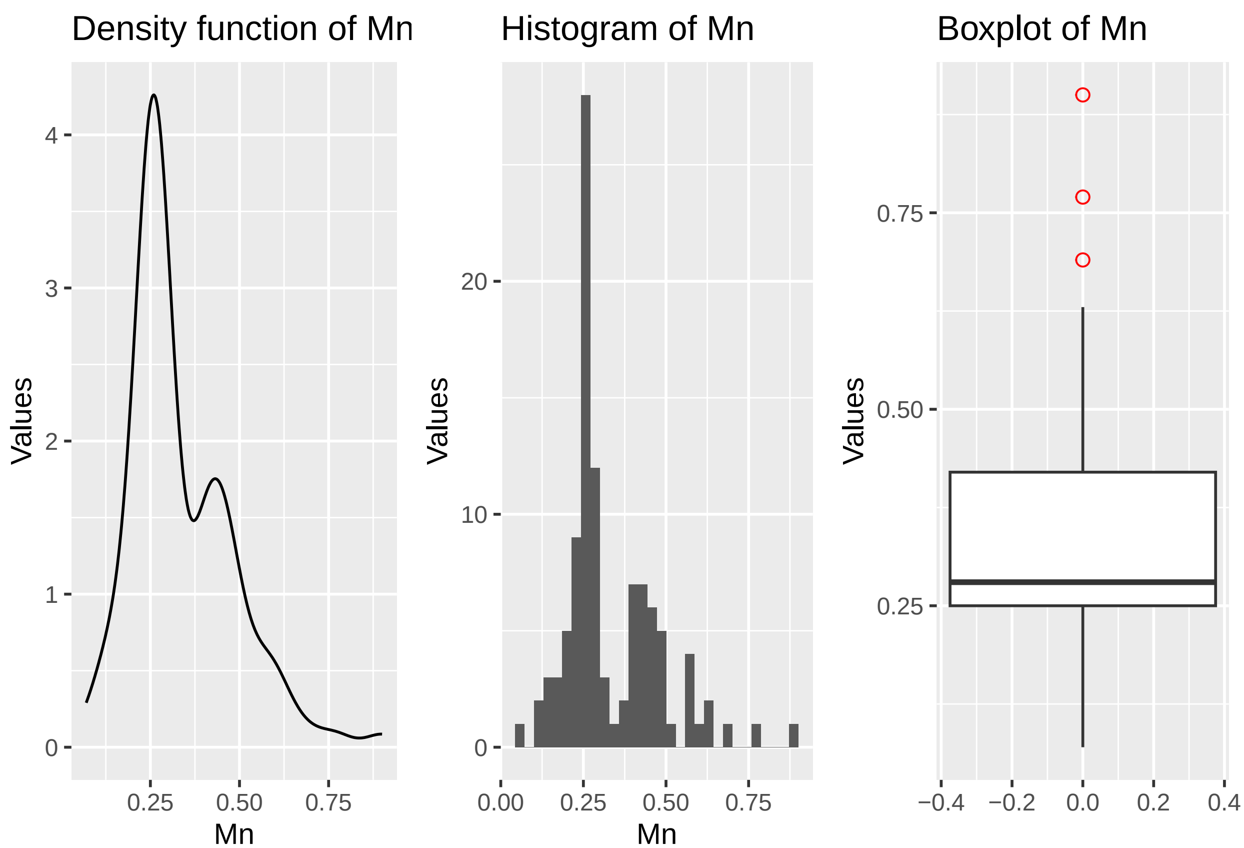
Mn - Manganese

```r
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Mn)
```

```
## Descriptive Statistics
## Mn
## N: 105
```

```
##
##                          Mn
## ----------------- --------
##              Mean     0.33
##           Std.Dev     0.14
##               Min     0.07
##                Q1     0.25
##            Median     0.28
##                Q3     0.42
##               Max     0.90
##               MAD     0.10
##               IQR     0.17
##                CV     0.43
##          Skewness     1.16
##       SE.Skewness     0.24
##          Kurtosis     1.65
##           N.Valid   105.00
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Mn))+geom_density()+
  labs(title = "Density function of Mn",x="Mn",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Mn))+geom_histogram()+
  labs(title = "Histogram of Mn",x="Mn",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Mn))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mn",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



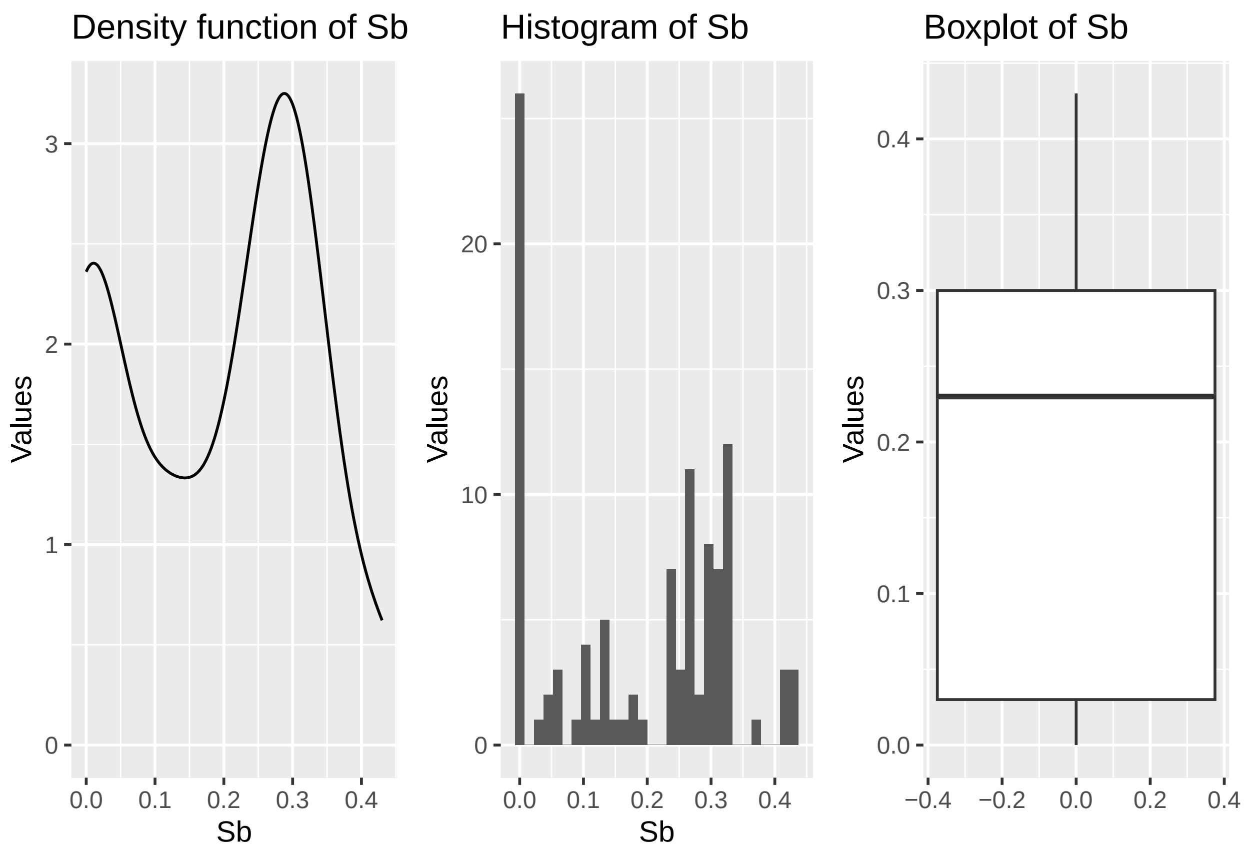
Sb - Antimony

```
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Sb)
```

```
## Descriptive Statistics
## Sb
## N: 105
##
##                           Sb
## ----------------- --------
##              Mean     0.19
##           Std.Dev     0.14
##               Min     0.00
##                Q1     0.03
##            Median     0.23
##                Q3     0.30
##               Max     0.43
##               MAD     0.13
##               IQR     0.27
##                CV     0.74
##          Skewness    -0.16
##       SE.Skewness     0.24
##          Kurtosis    -1.39
##           N.Valid   105.00
```

```
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Sb))+geom_density()+
  labs(title = "Density function of Sb",x="Sb",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Sb))+geom_histogram()+
  labs(title = "Histogram of Sb",x="Sb",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Sb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Sb",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



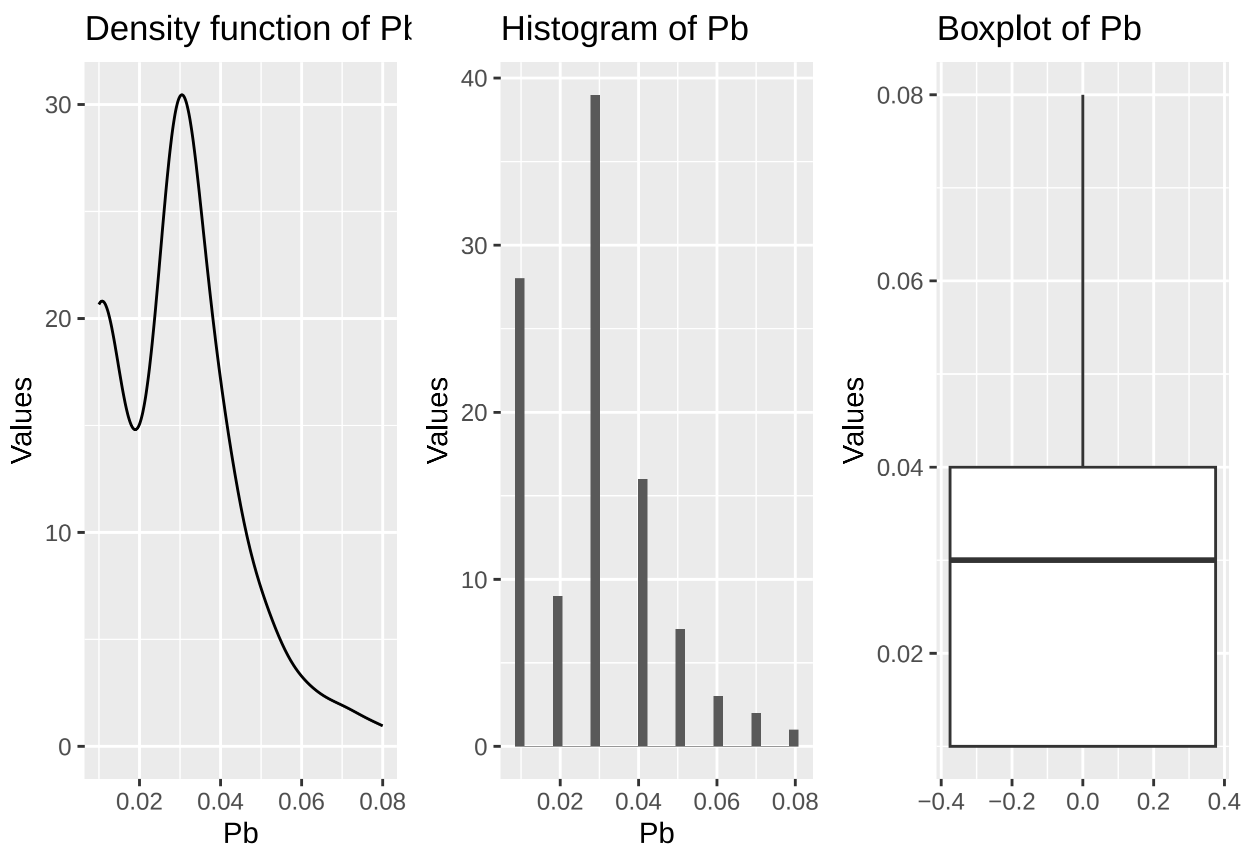
Pb - Plumb

```r
# Basic descriptive statistics
# Remember that package 'summarytools' is required
descr(Pb)
```

```
## Descriptive Statistics
## Pb
## N: 105
```

```
##
##                           Pb
## ----------------- --------
##              Mean     0.03
##           Std.Dev     0.02
##               Min     0.01
##                Q1     0.01
##            Median     0.03
##                Q3     0.04
##               Max     0.08
##               MAD     0.01
##               IQR     0.03
##                CV     0.54
##          Skewness     0.69
##       SE.Skewness     0.24
##          Kurtosis     0.49
##           N.Valid   105.00
##         Pct.Valid   100.00
```

```r
# Histogram, density and boxplot
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1,aes(x=Pb))+geom_density()+
  labs(title = "Density function of Pb",x="Pb",y="Values")

p2<-ggplot(data_RBGlass1,aes(x=Pb))+geom_histogram()+
  labs(title = "Histogram of Pb",x="Pb",y="Values")

p3<-ggplot(data_RBGlass1,aes(x=Pb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Pb",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3, nrow=1, common.legend = FALSE)
```



Basic descriptive statistics of categorical variables

The only categorical variable in this data set is *Site*.

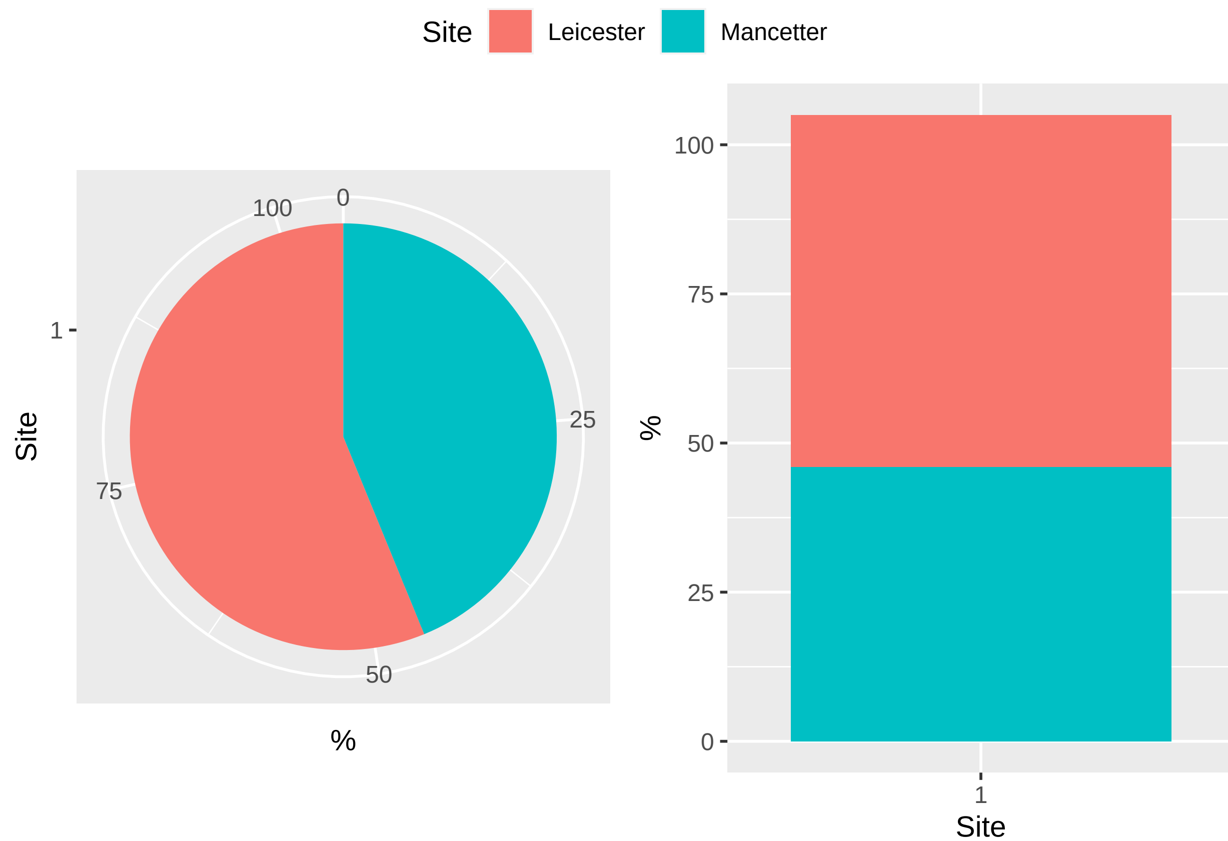
Site

```
# Frequency tables. Descriptive analysis
# Remember that package 'summarytools' is required
freq(Site)
```

```
## Frequencies
## Site
## Type: Factor
##
##                     Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## --------------- ------ --------- -------------- --------- --------------
##       Leicester     59     56.19          56.19     56.19          56.19
##       Mancetter     46     43.81         100.00     43.81         100.00
##            <NA>      0                               0.00         100.00
##           Total    105    100.00         100.00    100.00         100.00
```

```
# Pie chart and bar graph
p1<-ggplot(data_RBGlass1,aes(x=factor(1),fill=Site))+geom_bar()+
  coord_polar("y")+labs(x="Site",y="%")
p2<-ggplot(data_RBGlass1,aes(x=factor(1),fill=Site))+geom_bar()+
  labs(x="Site",y="%")
```

```r
# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,nrow = 1,ncol=2, common.legend = TRUE)
```

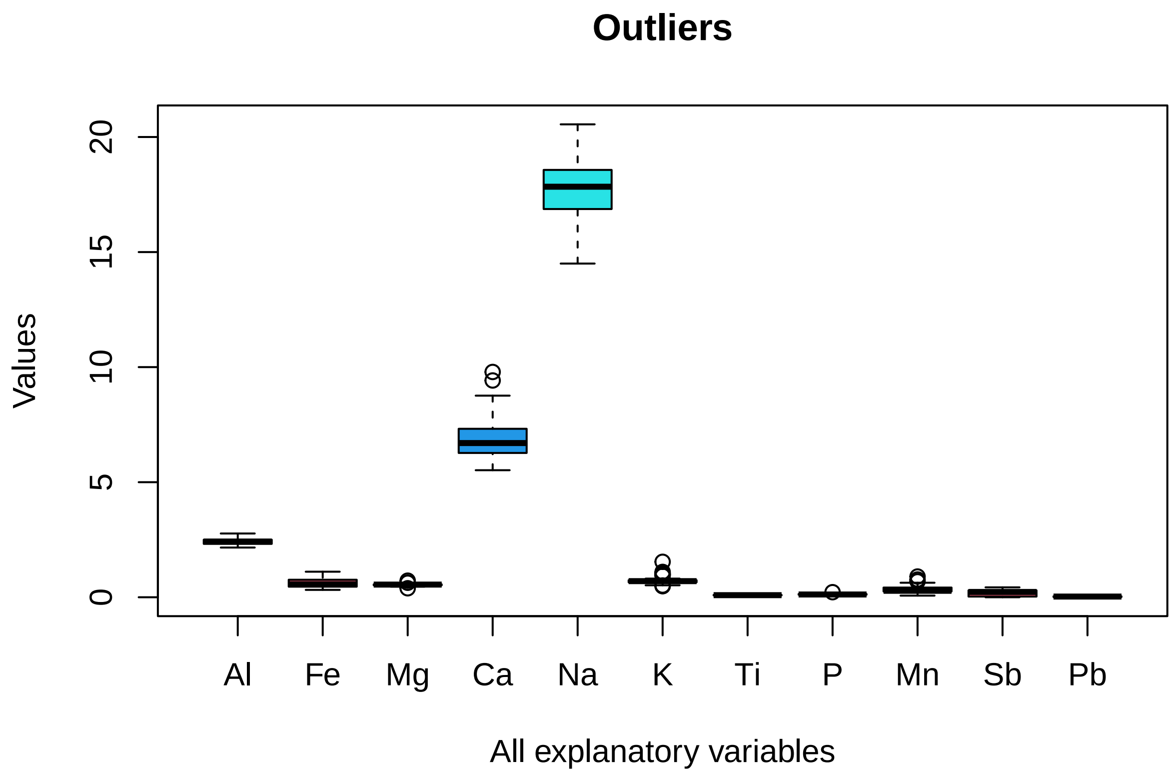


Outliers

Identification

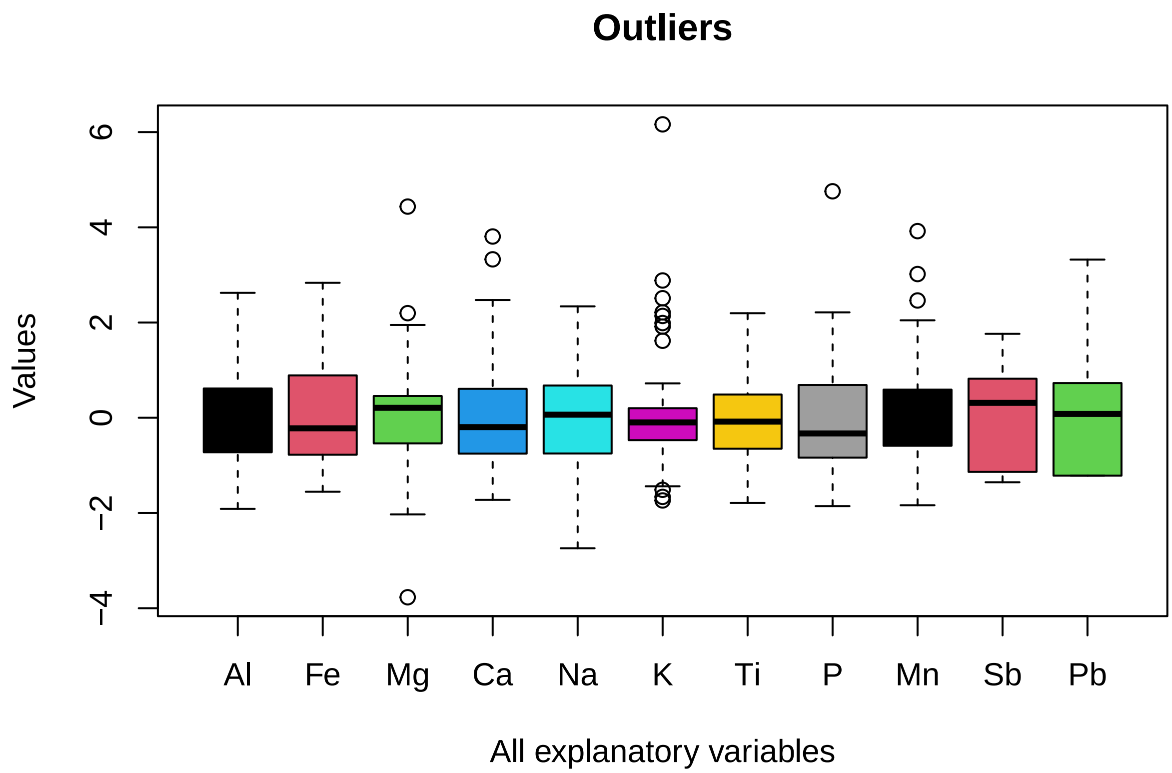
This graphical output shows together the boxplots of all the quantitative variables. This visualization is not the best due to the difference between the scales.

```r
# Boxplots of all variables together
# This visualization is not the best due to the difference between the scales
boxplot(data_RBGlass1[-1],main="Outliers",
        xlab="All explanatory variables",
        ylab="Values",
        col=c(1:11))
```



However, if the quantitative variables are standardized, the effect of scales differences is erased. This joint boxplots output is more informative.

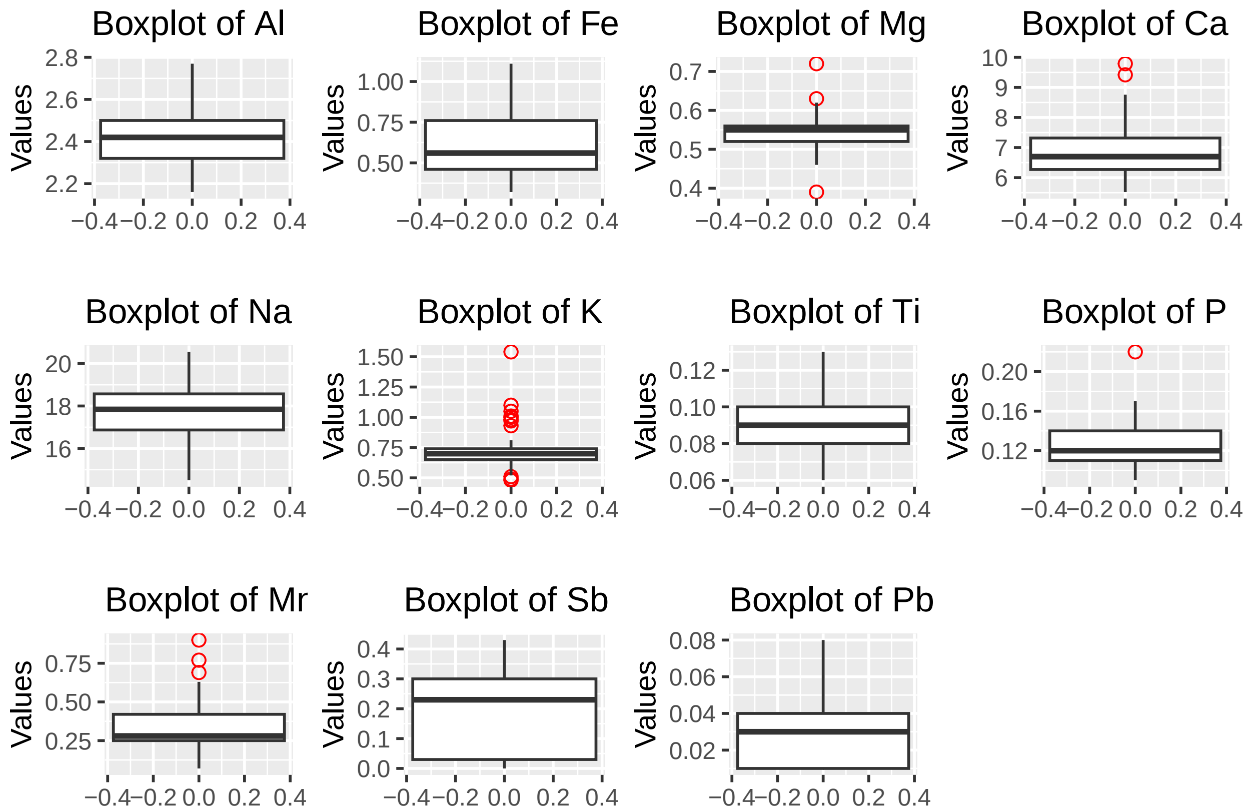
```
# Standardization
sca<-scale(data_RBGlass1[-1])

# Boxplots of all variables together
# This visualization is not affected by the difference between the scales
sca<-scale(data_RBGlass1[-1])
boxplot(sca,main="Outliers",
        xlab="All explanatory variables",
        ylab="Values",
        col=c(1:11))
```



This is another joint visualization of the boxplots without the effect of the difference in scales.

```
# Boxplots of all quantitative variables together
# Remember that package 'ggplot2' is required

p1<-ggplot(data_RBGlass1,aes(x=Al))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Al",x="Values",y="")

p2<-ggplot(data_RBGlass1,aes(x=Fe))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Fe",x="Values",y="")

p3<-ggplot(data_RBGlass1,aes(x=Mg))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mg",x="Values",y="")

p4<-ggplot(data_RBGlass1,aes(x=Ca))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ca",x="Values",y="")

p5<-ggplot(data_RBGlass1,aes(x=Na))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Na",x="Values",y="")

p6<-ggplot(data_RBGlass1,aes(x=K))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
```

```r
  coord_flip()+labs(title = "Boxplot of K",x="Values",y="")

p7<-ggplot(data_RBGlass1,aes(x=Ti))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ti",x="Values",y="")

p8<-ggplot(data_RBGlass1,aes(x=P))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of P",x="Values",y="")

p9<-ggplot(data_RBGlass1,aes(x=Mn))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mn",x="Values",y="")

p10<-ggplot(data_RBGlass1,aes(x=Sb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Sb",x="Values",y="")

p11<-ggplot(data_RBGlass1,aes(x=Pb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Pb",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11, nrow=3, ncol=4,
          commond.legend=FALSE)
```



Towards classification

On many occasions it is common to face **problems of classifying** data at the different levels of a response variable according to the different values of a set of explanatory variables. For this task, it is usually interesting to do a visual exploratory analysis that provides clues as to whether all the variables are really needed to build a good model or whether one or more of them is enough.

Unit 5 deals with the topic of classification but, at this stage, it is possible to anticipate which variable or combination of variables could provide good models.

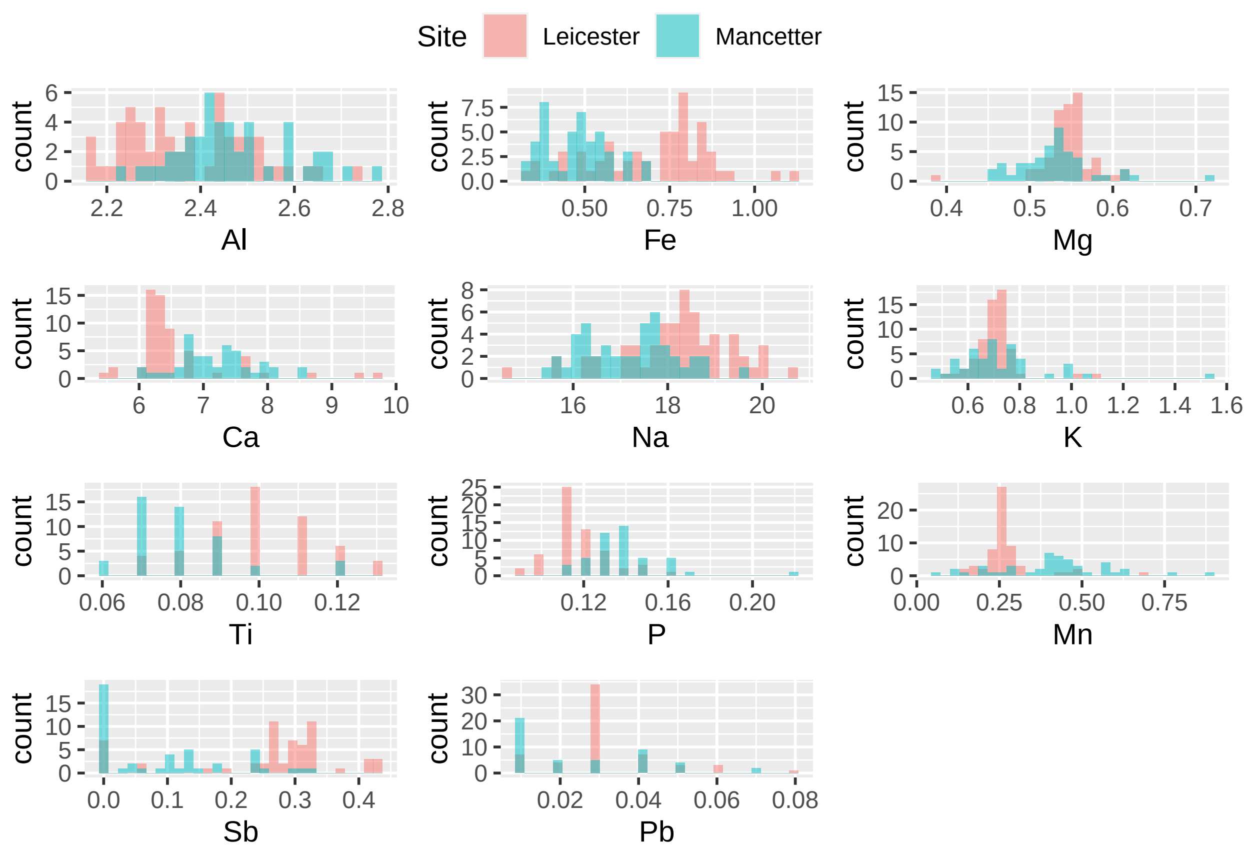
The next three sections illustrate a visual approach that searches for univariate, bivariate and trivariate classifiers.

Univariate visual exploratory analysis

Based on the graphical outputs below (overlapping histograms by *Site*), it appears that the variables *Fe*, *Mn* and *Sb* are candidates for building classification models with good performance.

```
# Boxplots of all quantitative variables together grouping by Site
# Remember that package 'ggplot2' is required
p1 <- ggplot(data = data_RBGlass1, aes(x = Al, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p2 <- ggplot(data = data_RBGlass1, aes(x = Fe, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p3 <- ggplot(data = data_RBGlass1, aes(x = Mg, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)

p4 <- ggplot(data = data_RBGlass1, aes(x = Ca, fill = Site)) +
```

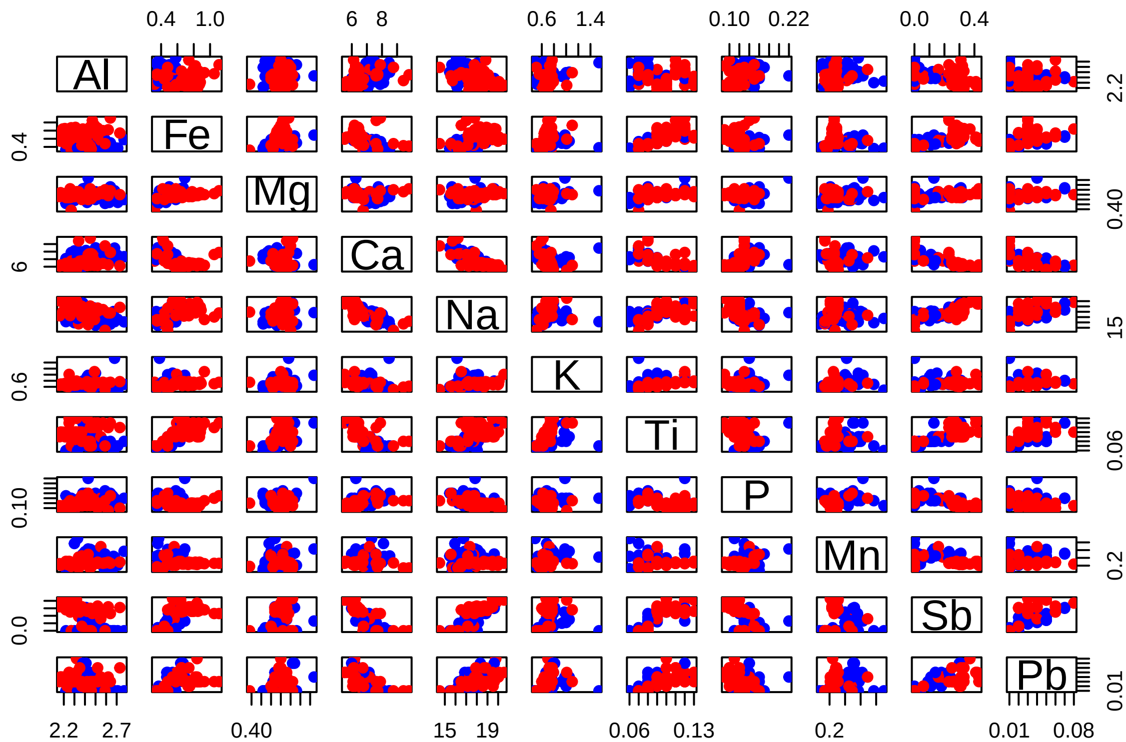
```
      geom_histogram(position = "identity", alpha = 0.5)
p5 <- ggplot(data = data_RBGlass1, aes(x = Na, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p6 <- ggplot(data = data_RBGlass1, aes(x = K, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)

p7 <- ggplot(data = data_RBGlass1, aes(x = Ti, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p8 <- ggplot(data = data_RBGlass1, aes(x = P, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p9 <- ggplot(data = data_RBGlass1, aes(x = Mn, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)

p10 <- ggplot(data = data_RBGlass1, aes(x = Sb, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)
p11 <- ggplot(data = data_RBGlass1, aes(x = Pb, fill = Site)) +
      geom_histogram(position = "identity", alpha = 0.5)

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2, p3,p4,p5,p6,p7,p8,p9,p10,p11, nrow = 4, ncol=3, common.legend = TRUE)
```



Bivariate visual exploratory analysis

As before, based on the following graphical output (scatter plots for each pair of variable combinations), the pairs *Al* - *Mn*, *Mn* - *Fe* and *Mn* - *Pb* could be consider to define a well-performing bivariate classification model, but it is not so clear.

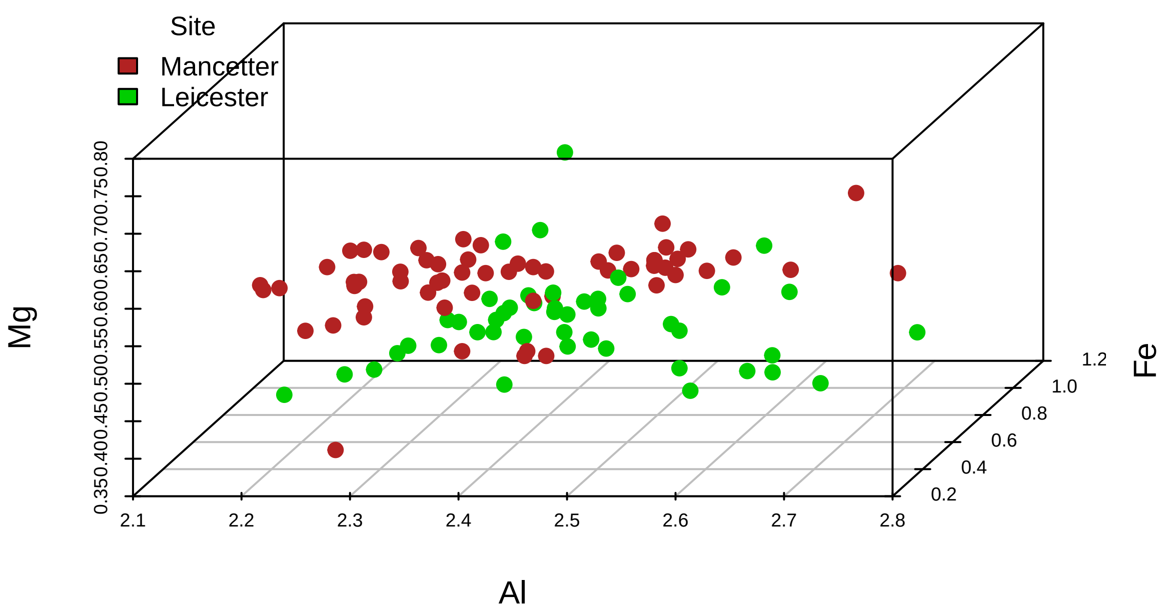
```r
# Scatter plots for each pair of variable combinations
pairs(x = data_RBGlass1[, c("Al","Fe","Mg","Ca","Na","K","Ti","P","Mn","Sb","Pb")],
      col = c("red","blue")[data_RBGlass1$Site], pch = 19)
```



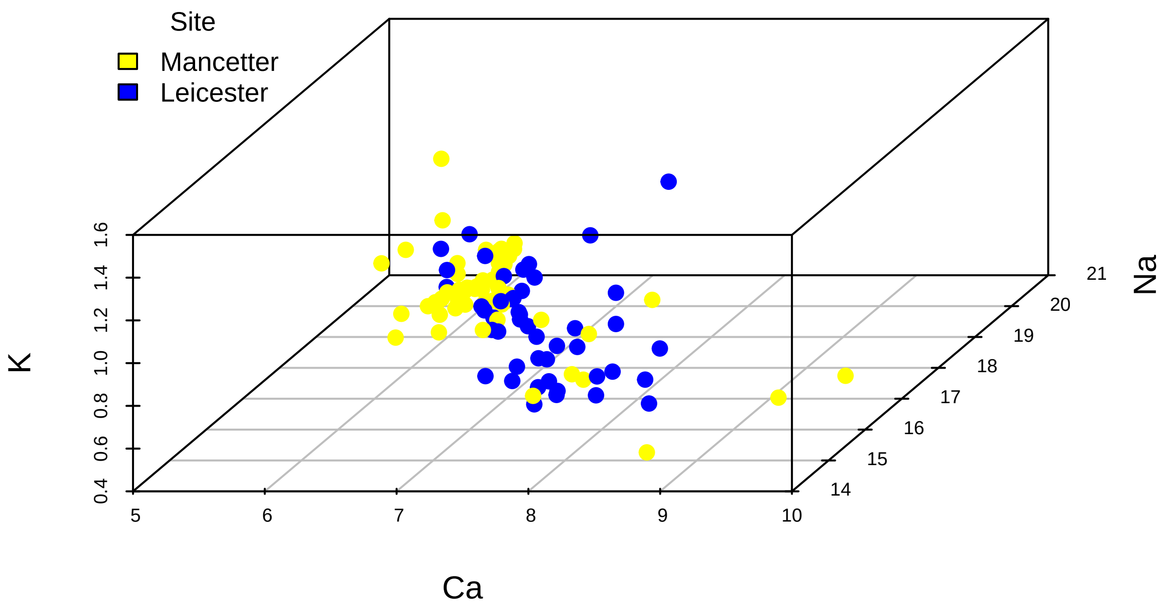
Trivariate visual exploratory analysis

Finally, these 3D-scatter plots analize wether combinations of three variables are able to separate the two sites, thinking again about the performance of classification models. For this illustration, three 3D-scatter plots are displayed combining different triplets of variables: Al-Fe-Mg; Ca-Na-K and Mg-P-Ca.

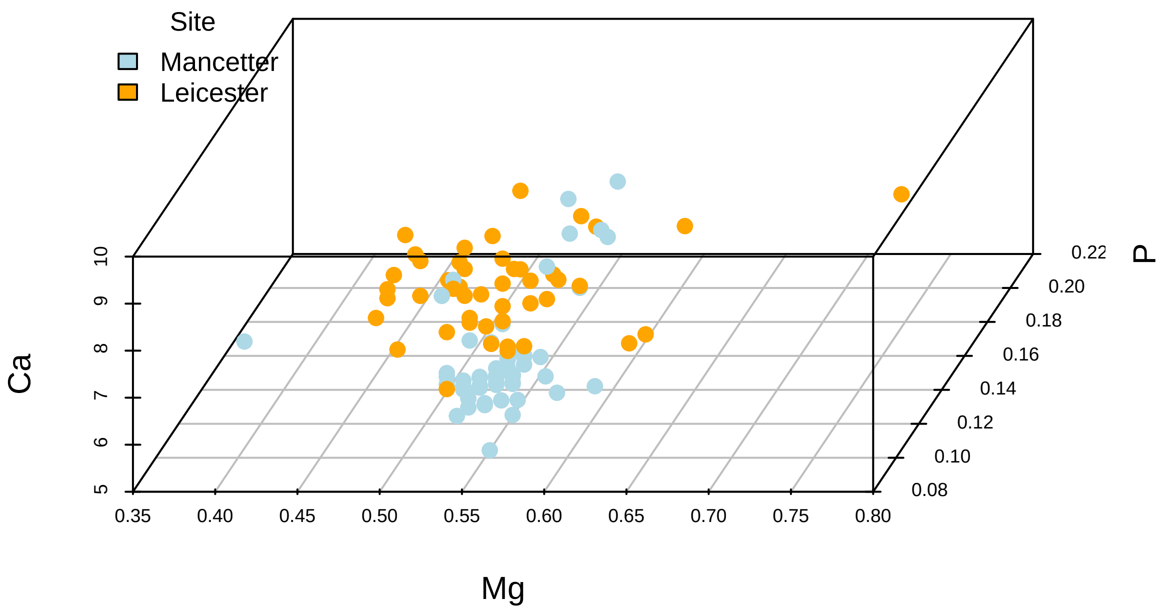
```r
# 3D scatter plot for the variables Al, Fe and Mg
# Remember that package 'scatterplot3d' is required
scatterplot3d(Al, Fe, Mg,
              color = c("firebrick", "green3")[data_RBGlass1$Site],pch = 19,
              grid = TRUE, xlab = "Al", ylab = "Fe",
              zlab = "Mg", angle = 65, cex.axis = 0.6)
legend("topleft",
       bty = "n", cex = 0.8,
       title = "Site",
       c("Mancetter", "Leicester"), fill = c("firebrick", "green3"))
```




```
# 3D scatter plot for the variables Ca, Na and K
# Remember that package 'scatterplot3d' is required
scatterplot3d(Ca, Na, K,
              color = c("yellow", "blue")[data_RBGlass1$Site],pch = 19,
              grid = TRUE, xlab = "Ca", ylab = "Na",
              zlab = "K", angle = 65, cex.axis = 0.6)
legend("topleft",
       bty = "n", cex = 0.8,
       title = "Site",
       c("Mancetter", "Leicester"), fill = c("yellow", "blue"))
```




```r
# 3D scatter plot for the variables Mg, P and Ca
# Remember that package 'scatterplot3d' is required
scatterplot3d(Mg, P, Ca,
              color = c("lightblue", "orange")[data_RBGlass1$Site],pch = 19,
              grid = TRUE, xlab = "Mg", ylab = "P",
              zlab = "Ca", angle = 65, cex.axis = 0.6)
legend("topleft",
       bty = "n", cex = 0.8,
       title = "Site",
       c("Mancetter", "Leicester"), fill = c("lightblue", "orange"))
```



Making decisions

From previous graphical outputs it is noticed the presence of outliers for the variables *Mg, Ca, K, P and Mn*. It is relevant to take into account these values since multivariate methods, such as principal component analisis (PCA), are sensitive to this fact.

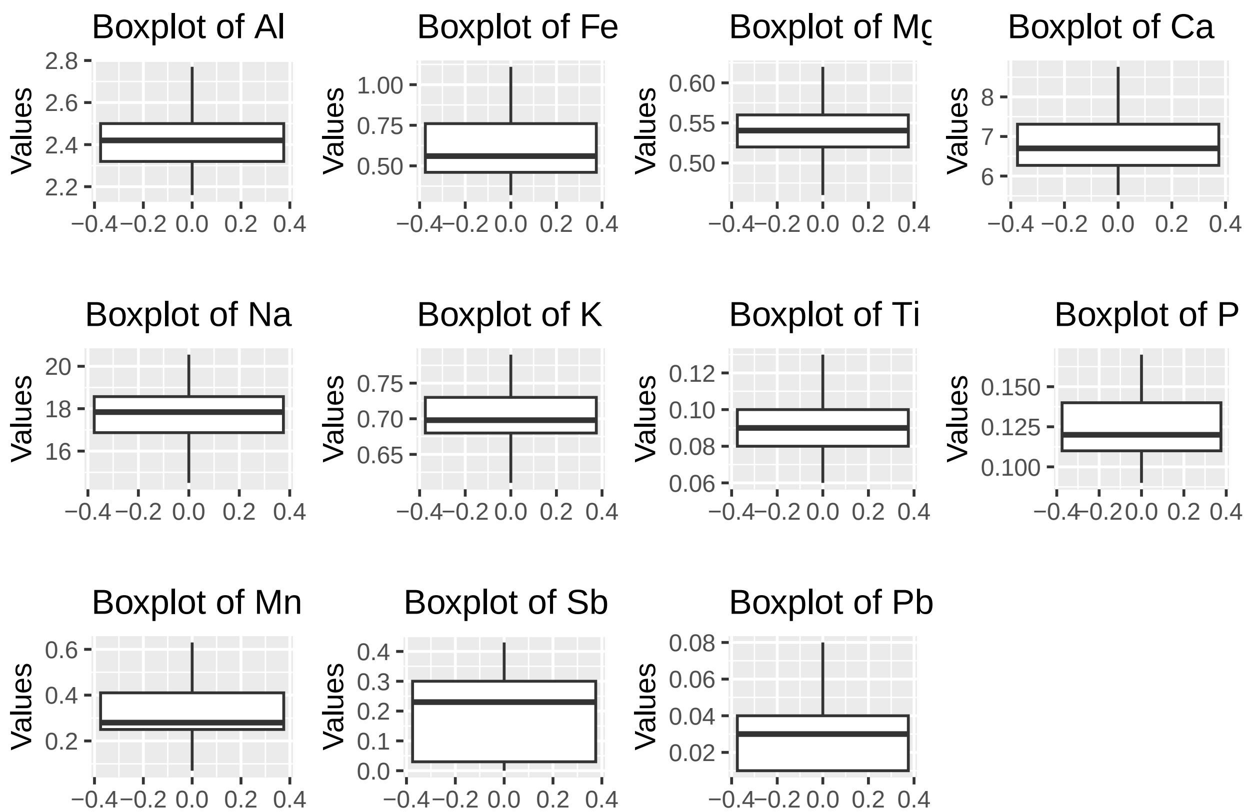
This is not a light topic and it should be analyzed outlier per outlier. However, since the objective of this guide is to introduce to the reader in this preliminary step of exploratory data analysis and data preparation, **the decision for outliers is to replace them by the mean of their variable**. Perhaps it is not the best option, it depends on the problem under analysis and the data recorded, but it is a way to introduce the reader to **how to define functions in R language**.

The following source code defines the function *outlier* whose utility is to deal with the univariate outliers.

```
# Recursive function that modifies outliers by the mean of their variable
outlier<-function(data,na.rm=T){

  H<-1.5*IQR(data)
  data[data<quantile(data,0.25,na.rm = T)-H]<-NA
  data[data>quantile(data,0.75, na.rm = T)+H]<-NA
  data[is.na(data)]<-mean(data, na.rm = T)
  H<-1.5*IQR(data)

  if (TRUE %in% (data<quantile(data,0.25,na.rm = T)-H) |
      TRUE %in% (data>quantile(data,0.75,na.rm = T)+H))
    outlier(data)
  else
```

```r
    return(data)

}

# This data.frame is to preserve original data once the outliers are modified
data_RBGlass1_aux<-data_RBGlass1

# Called to outlier function for each variable identified with outliers
data_RBGlass1_aux$Mg<-outlier(data_RBGlass1_aux$Mg)
data_RBGlass1_aux$Ca<-outlier(data_RBGlass1_aux$Ca)
data_RBGlass1_aux$K<-outlier(data_RBGlass1_aux$K)
data_RBGlass1_aux$P<-outlier(data_RBGlass1_aux$P)
data_RBGlass1_aux$Mn<-outlier(data_RBGlass1_aux$Mn)


# Boxplots of all quantitative variables together once outliers are dealt with
# Remember that package 'ggplot2' is required
p1<-ggplot(data_RBGlass1_aux,aes(x=Al))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Al",x="Values",y="")

p2<-ggplot(data_RBGlass1_aux,aes(x=Fe))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Fe",x="Values",y="")

p3<-ggplot(data_RBGlass1_aux,aes(x=Mg))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mg",x="Values",y="")

p4<-ggplot(data_RBGlass1_aux,aes(x=Ca))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ca",x="Values",y="")

p5<-ggplot(data_RBGlass1_aux,aes(x=Na))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Na",x="Values",y="")

p6<-ggplot(data_RBGlass1_aux,aes(x=K))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of K",x="Values",y="")

p7<-ggplot(data_RBGlass1_aux,aes(x=Ti))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Ti",x="Values",y="")

p8<-ggplot(data_RBGlass1_aux,aes(x=P))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of P",x="Values",y="")

p9<-ggplot(data_RBGlass1_aux,aes(x=Mn))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Mn",x="Values",y="")
```

```r
p10<-ggplot(data_RBGlass1_aux,aes(x=Sb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Sb",x="Values",y="")

p11<-ggplot(data_RBGlass1_aux,aes(x=Pb))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of Pb",x="Values",y="")

# This function controls the graphical output device
# Remember that package 'ggpubr' is required
ggarrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11, nrow=3, ncol=4,
          commond.legend=FALSE)
```



Principal component analysis

Requirements

Correlated variables

According to the numerical results below, it is observed that the data **are correlated** both **at the sample level** (see correlation matrix) and **at the populacion level** (Bartlett's sphericity test is significant).

```r
##################################
# Correlation at sample level #
##################################
```

```
# Are the variables correlated at sample level?
correlation_matrix<-cor(data_RBGlass1[-1])
correlation_matrix
```

```
##             Al         Fe          Mg          Ca         Na          K
## Al  1.00000000 -0.1150748  0.01023359  0.38513257 -0.4462544  0.1133173
## Fe -0.11507482  1.0000000  0.43329162 -0.55316158  0.5085758  0.1952836
## Mg  0.01023359  0.4332916  1.00000000 -0.10837026  0.2521078  0.3650340
## Ca  0.38513257 -0.5531616 -0.10837026  1.00000000 -0.7847872 -0.2024186
## Na -0.44625442  0.5085758  0.25210784 -0.78478722  1.0000000  0.1714821
## K   0.11331727  0.1952836  0.36503395 -0.20241858  0.1714821  1.0000000
## Ti -0.12992370  0.7734593  0.54250700 -0.67388920  0.6203305  0.3395985
## P   0.34573685 -0.3463501  0.12646087  0.48404469 -0.5576861  0.1025215
## Mn  0.15094053 -0.2471959  0.11770808  0.09957463 -0.1657524  0.1841405
## Sb -0.46757821  0.7270397  0.39033528 -0.81788919  0.8291989  0.2094871
## Pb -0.23445235  0.4511995  0.38614070 -0.63423737  0.6057673  0.2936122
##            Ti          P          Mn         Sb         Pb
## Al -0.1299237  0.3457368  0.15094053 -0.4675782 -0.2344524
## Fe  0.7734593 -0.3463501 -0.24719587  0.7270397  0.4511995
## Mg  0.5425070  0.1264609  0.11770808  0.3903353  0.3861407
## Ca -0.6738892  0.4840447  0.09957463 -0.8178892 -0.6342374
## Na  0.6203305 -0.5576861 -0.16575238  0.8291989  0.6057673
## K   0.3395985  0.1025215  0.18414054  0.2094871  0.2936122
## Ti  1.0000000 -0.3472705 -0.17213390  0.8129687  0.6348322
## P  -0.3472705  1.0000000  0.37082293 -0.6376719 -0.2745607
## Mn -0.1721339  0.3708229  1.00000000 -0.2465848  0.1242556
## Sb  0.8129687 -0.6376719 -0.24658476  1.0000000  0.7044968
## Pb  0.6348322 -0.2745607  0.12425560  0.7044968  1.0000000
```

```
det(correlation_matrix)
```

```
## [1] 0.0002359252
```

```
# It is noticed an important correlation between some variables
# For instance, sodium (NA) and antimony (Sb) or titanium (Ti) and iron (Fe)
cor(data_RBGlass1$Na,data_RBGlass1$Sb)
```

```
## [1] 0.8291989
```

```
cor(data_RBGlass1$Ti,data_RBGlass1$Fe)
```

```
## [1] 0.7734593
```

```
###################################
# Correlation at population level #
###################################

# Bartlett's sphericity test:
# This test checks whether the correlations are significantly different from 0
# The null hypothesis is H_0; det(R)=1 means the variables are uncorrelated
# R denotes the correlation matrix
# cortest.bartlett function in the package pysch performs this test
# This function works with standardized data.

# Standardization
data_RBGlass1_scale<-scale(data_RBGlass1[-1])
```

```r
# Bartlett's sphericity test
cortest.bartlett(cor(data_RBGlass1_scale))
```

```
## $chisq
## [1] 789.2636
##
## $p.value
## [1] 1.328886e-130
##
## $df
## [1] 55
```

Absence of outliers

Done in **Section 2.4.2** in the data.frame *data_RBGlass1_aux*.

Standardized data

It is not necessary, since the *prcomp* function that obtains the principal components standardizes the data on its own.

Principal components

Obtaining

```r
# The 'prcomp' function in the base R package performs this analysis
# Parameters 'scale' and 'center' are set to TRUE to consider standardized data
PCA<-prcomp(data_RBGlass1_aux[-1], scale=T, center = T)

# The field 'rotation' of the 'PCA' object is a matrix
# Its columns are the coefficients of the principal components
# Indicates the weight of each variable in the corresponding principal component
PCA$rotation
```

```
##            PC1         PC2         PC3          PC4         PC5         PC6
## Al -0.18179880 -0.40781189  0.50129477 -0.365381890  0.34996101 -0.41863473
## Fe  0.32110603 -0.06987227  0.44247680  0.053846877  0.22869275  0.47225421
## Mg  0.23535547 -0.27680947  0.26248103  0.670431813 -0.39665192 -0.38164570
## Ca -0.37619169 -0.11167957  0.15939482  0.155756331 -0.15827155 -0.06092607
## Na  0.36705551  0.06570094 -0.21871625 -0.128455787 -0.11981033 -0.16690952
## K   0.17688291 -0.46152967 -0.04648889 -0.530408439 -0.63510107  0.14312214
## Ti  0.36990846 -0.15716511  0.26118636  0.002883746  0.17167934  0.14128521
## P  -0.29819545 -0.34708754 -0.01939677  0.253346602 -0.06211663  0.61145092
## Mn -0.06071074 -0.55532244 -0.50560924  0.146567476  0.34260819 -0.09287739
## Sb  0.41137939  0.07383417 -0.01102711  0.059780185  0.08906259  0.02346636
## Pb  0.31974807 -0.25175413 -0.29027566  0.052680395  0.25785970 -0.03165677
##             PC7         PC8         PC9        PC10        PC11
## Al  0.104390369  0.24803088  0.11475639 -0.09793070  0.15144116
## Fe -0.346757129 -0.24975665  0.33224722 -0.31215314 -0.16598937
## Mg  0.003113559  0.05598816 -0.07376465 -0.19435784 -0.01620080
## Ca -0.038733828 -0.38246081  0.50786122  0.59939980 -0.05098185
## Na -0.210267495  0.57719454  0.53505415  0.16550305 -0.25171907
## K  -0.023610780 -0.19420099 -0.06540826 -0.04257265  0.06331666
## Ti  0.038304156  0.10752728 -0.47220042  0.64026574 -0.27360948
## P   0.235865139  0.50953064  0.09080024  0.01005202  0.15722187
```

```
## Mn -0.513174033 -0.10538043 -0.09709768  0.01965682  0.02312806
## Sb -0.012536323 -0.04358650  0.12459816  0.22447519  0.86323037
## Pb  0.708504010 -0.26764786  0.25757862 -0.07677539 -0.18438425
```

```r
# Standard deviations of each principal component
PCA$sdev
```

```
##  [1] 2.3442535 1.3223648 1.0155880 0.8596329 0.7474804 0.6614342 0.5733699
##  [8] 0.5025035 0.4749734 0.3536897 0.2392869
```

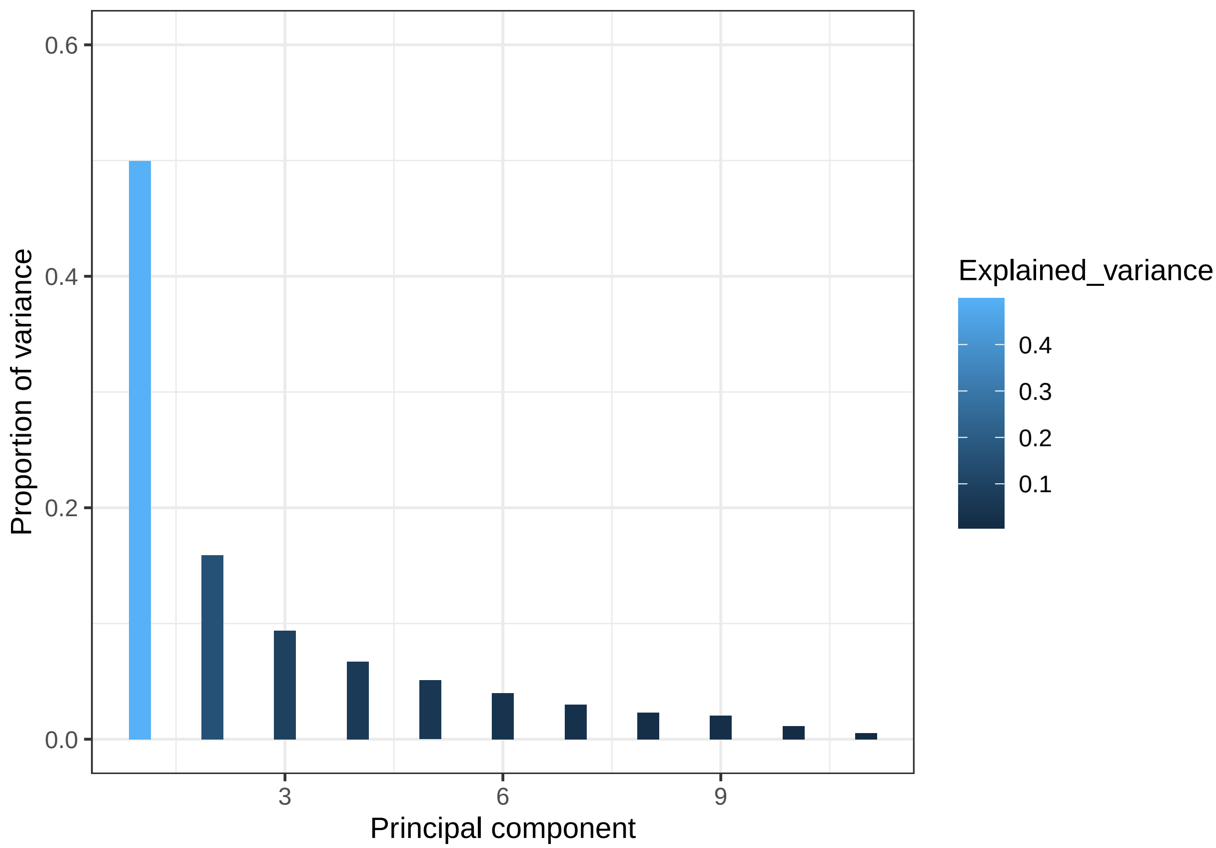
Each principal component is obtained in a simple way as a linear combination of all the variables with the coefficients indicated by the columns of the rotation matrix.

Explained variance rate

```r
# The function 'summary' applied to the 'PCA' object provides relevant information
# - Standard deviations of each principal component
# - Proportion of variance explained and cummulative variance
summary(PCA)
```

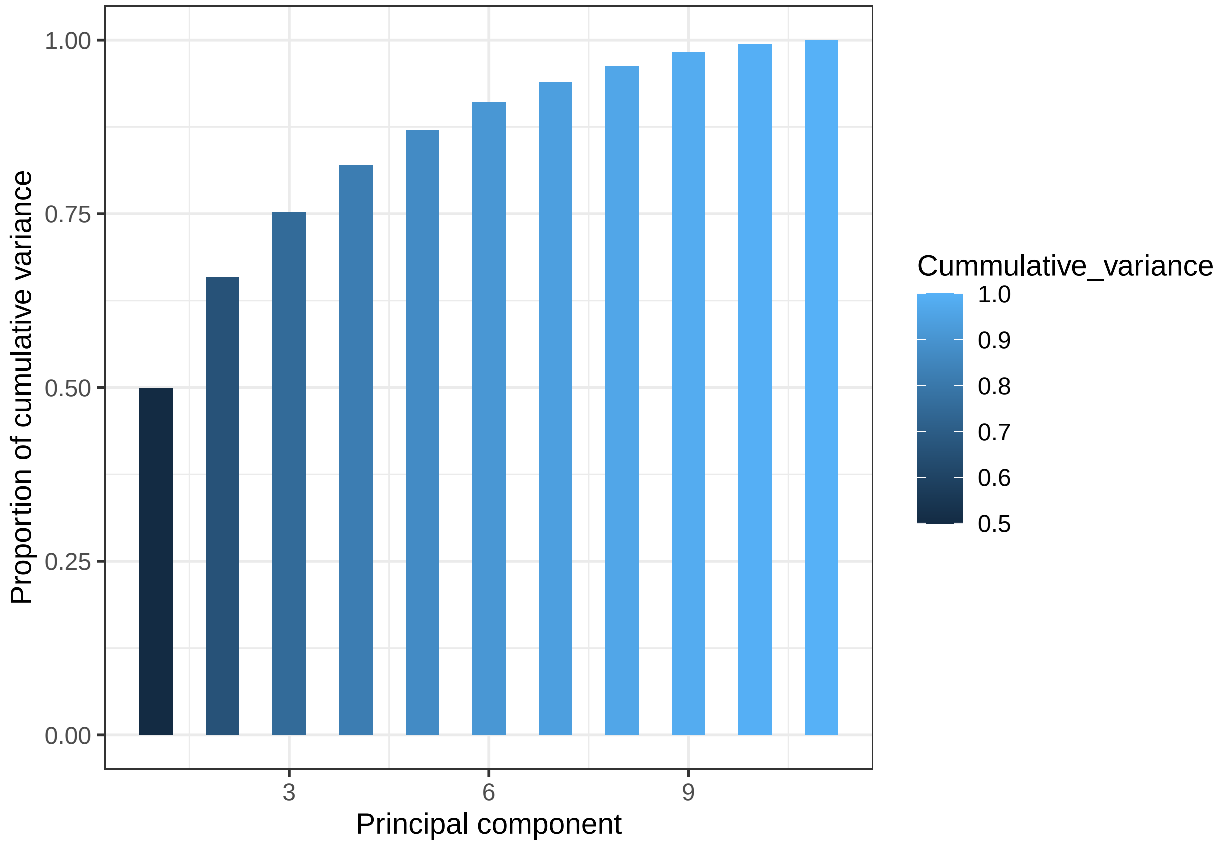
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.3443 1.3224 1.01559 0.85963 0.74748 0.66143 0.57337
## Proportion of Variance 0.4996 0.1590 0.09377 0.06718 0.05079 0.03977 0.02989
## Cumulative Proportion  0.4996 0.6586 0.75233 0.81951 0.87030 0.91007 0.93996
##                            PC8     PC9    PC10    PC11
## Standard deviation     0.50250 0.47497 0.35369 0.23929
## Proportion of Variance 0.02296 0.02051 0.01137 0.00521
## Cumulative Proportion  0.96291 0.98342 0.99479 1.00000
```

```r
# The following graph shows the proportion of explained variance
Explained_variance <- PCA$sdev^2 / sum(PCA$sdev^2)

p1<-ggplot(data = data.frame(Explained_variance, pc = 1:11),
  aes(x = pc, y = Explained_variance, fill=Explained_variance )) +
  geom_col(width = 0.3) + scale_y_continuous(limits = c(0,0.6)) + theme_bw() +
  labs(x = "Principal component", y= "Proportion of variance")

# The following graph shows the proportion of cumulative explained variance
Cummulative_variance<-cumsum(Explained_variance)

p2<-ggplot( data = data.frame(Cummulative_variance, pc = 1:11),
  aes(x = pc, y = Cummulative_variance ,fill=Cummulative_variance )) +
  geom_col(width = 0.5) +  scale_y_continuous(limits = c(0,1)) + theme_bw() +
  labs(x = "Principal component",
       y = "Proportion of cumulative variance")

p1
```



0.6
0.4
0.2
0.0
Proportion of variance
3
6
9
Principal component
Explained_variance
0.4
0.3
0.2
0.1


```
p2
```



Appropriate number of principal components

There are different methods:

- 1.- **Elbow method** (Cuadras, 2007).
- 2.- **At the discretion of the researcher** who chooses a minimum percentage of variance explained by the principal components (it is not reliable because it can give more than necessary).
- 3.- **Rule of Abdi et al.** (2010). The variances explained by the principal components are averaged and those whose proportion of explained variance exceeds the mean are selected.

For this illustration, applying the rule of Abdi et al., only **three principal components are considered**, as can be deduced from the following code chunk.

```
#######################
# Rule of Abdi et al. #
#######################

# Variances
PCA$sdev^2
```

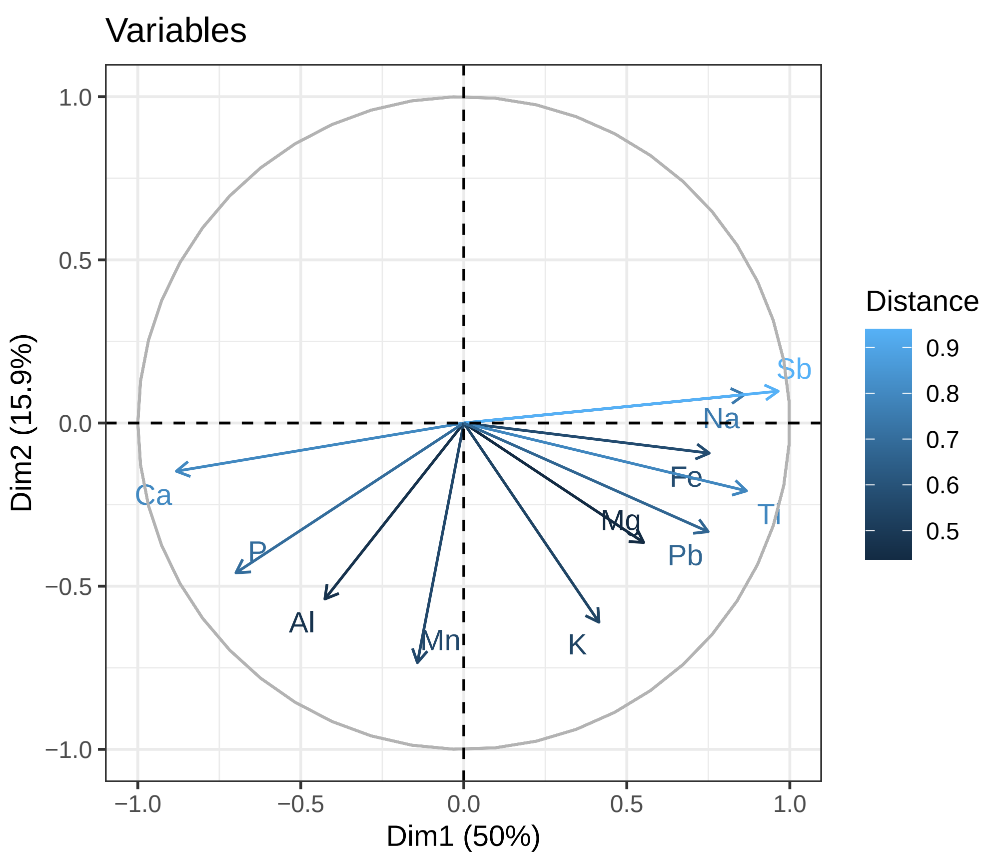
```
##  [1] 5.4955243 1.7486487 1.0314189 0.7389688 0.5587270 0.4374952 0.3287530
##  [8] 0.2525098 0.2255997 0.1250964 0.0572582
```

```
# Average of variances
mean(PCA$sdev^2)
```

```
## [1] 1
```

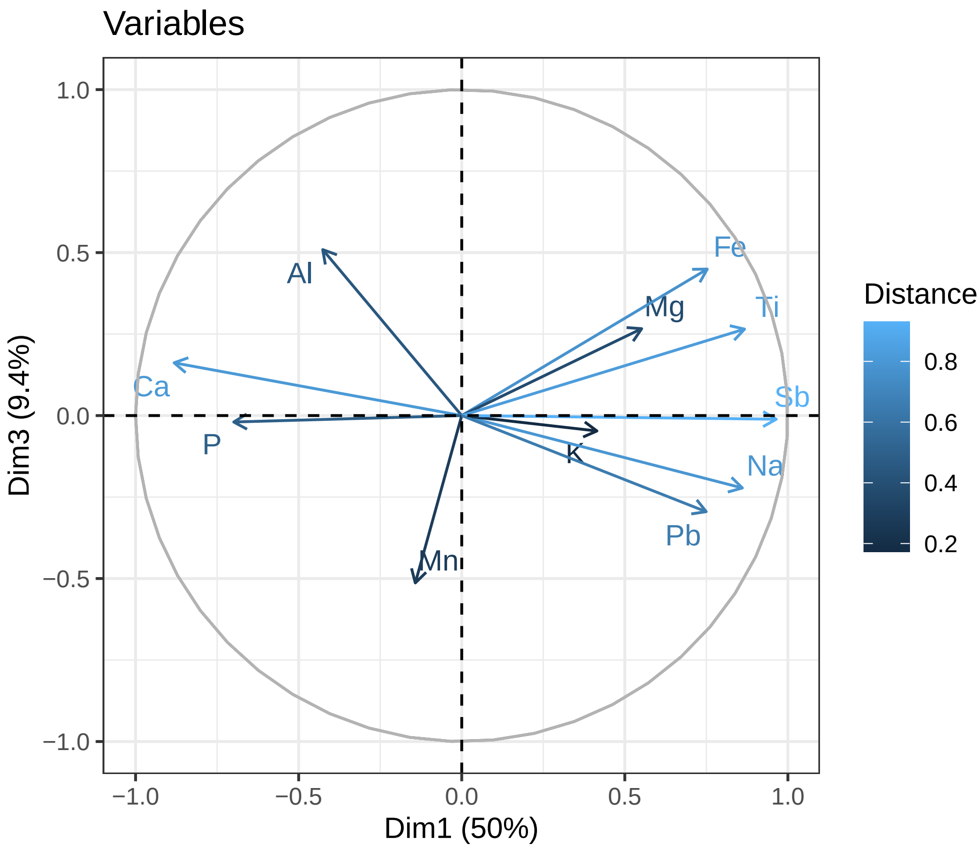
PCA graphical outputs of interest

```r
# These graphical outputs show the projection of the variables in two dimensions
# Display the weight of the variable in the direction of the principal component
p1<-fviz_pca_var(PCA,repel=TRUE,col.var="cos2",
                 legend.title="Distance", title="Variables")+theme_bw()

p2<-fviz_pca_var(PCA,axes=c(1,3),repel=TRUE,col.var="cos2",
                 legend.title="Distance", title="Variables")+theme_bw()

p3<-fviz_pca_var(PCA,axes=c(2,3),repel=TRUE,col.var="cos2",
                 legend.title="Distance", title="Variables")+theme_bw()

# Displaying graphics
p1
```

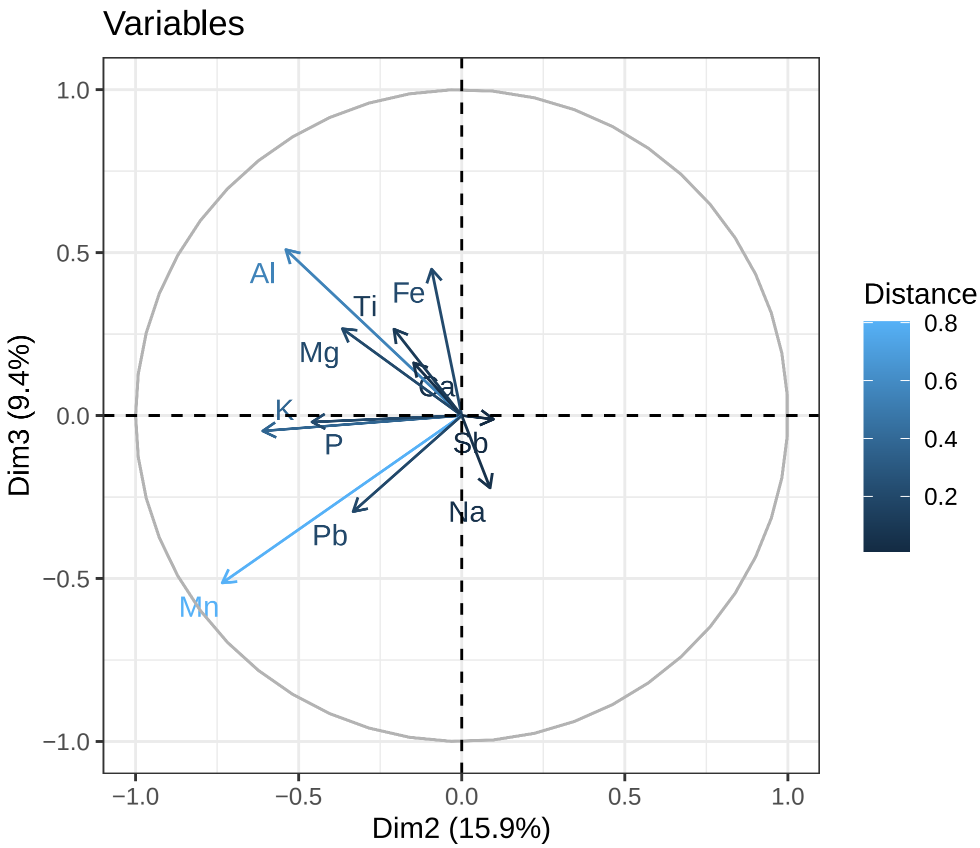


Distances

```r
p2
```

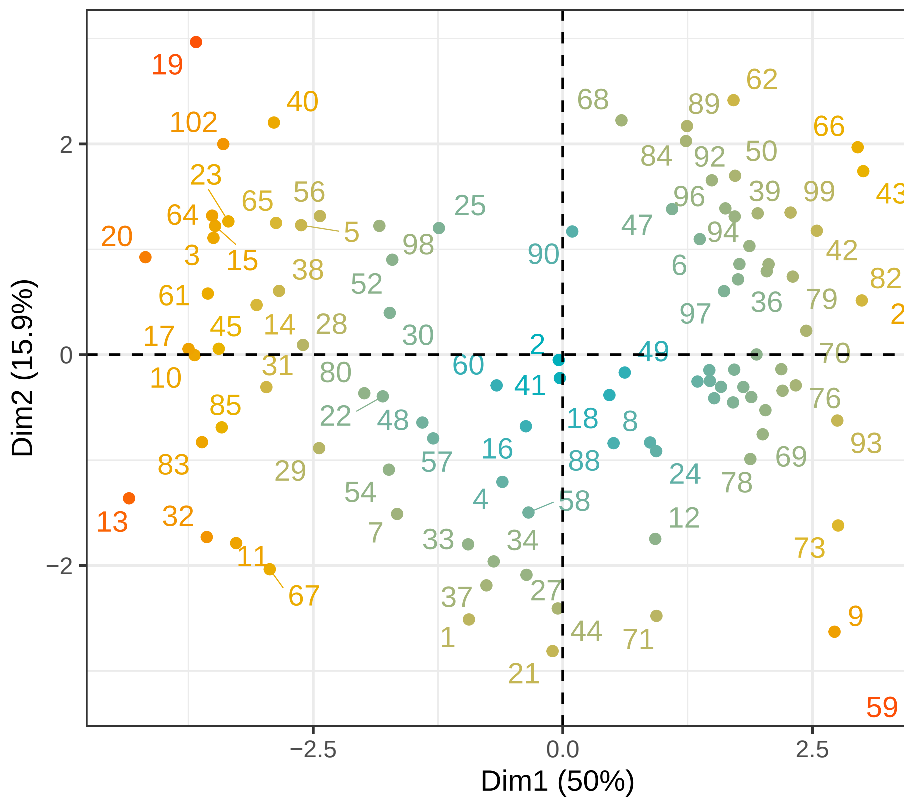

Variables
Dim1 (50%)
Dim3 (9.4%)
Distance
0.8
0.6
0.4
0.2
Fe
Mg
Ti
Al
Ca
Sb
P
K
Na
Pb
Mn
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

```
p3
```



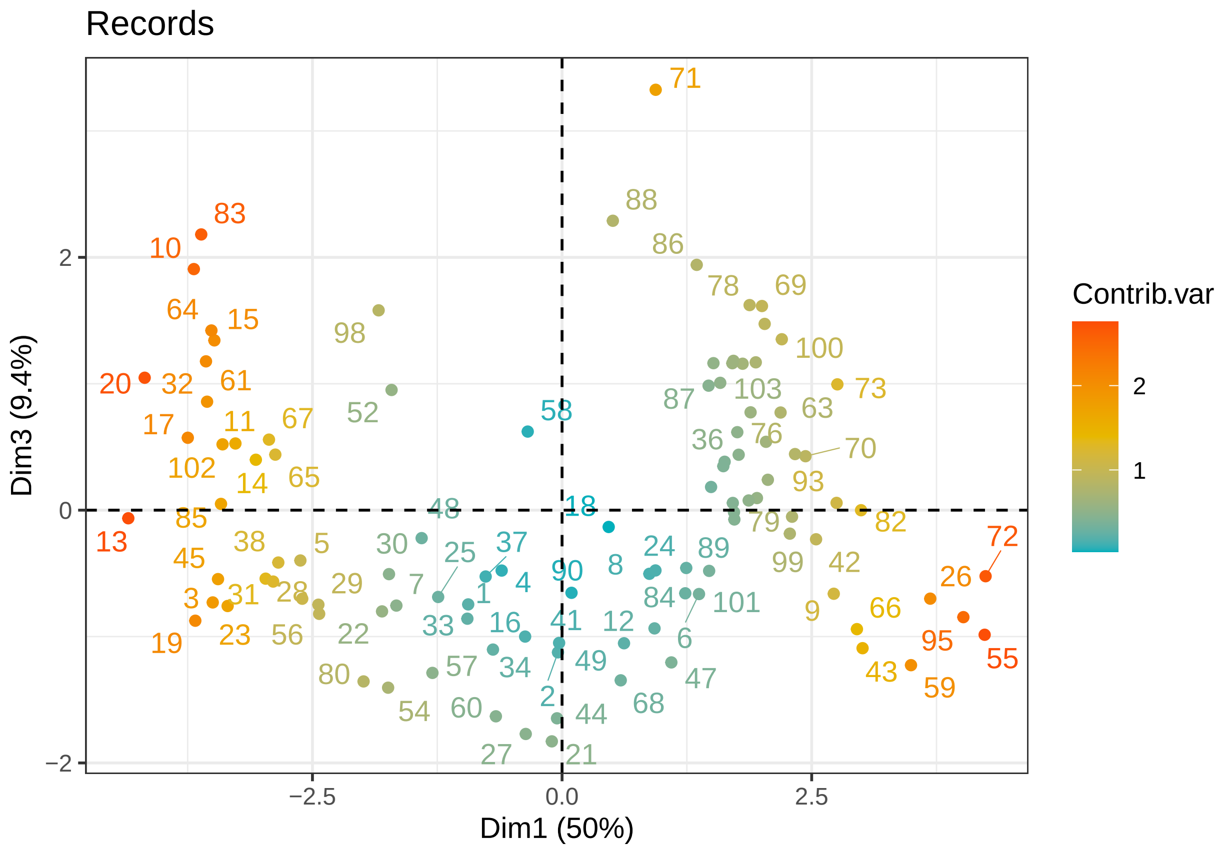

```
# It is also possible to represent the observations
# As well as identify with colors those observations that explain the greatest
# variance of the principal components
p1<-fviz_pca_ind(PCA,col.ind = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel=TRUE,legend.title="Contrib.var", title="Records")+theme_bw()

p2<-fviz_pca_ind(PCA,axes=c(1,3),col.ind = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel=TRUE,legend.title="Contrib.var", title="Records")+theme_bw()

p3<-fviz_pca_ind(PCA,axes=c(2,3),col.ind = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel=TRUE,legend.title="Contrib.var", title="Records")+theme_bw()

# Displaying graphics
p1
```

Records

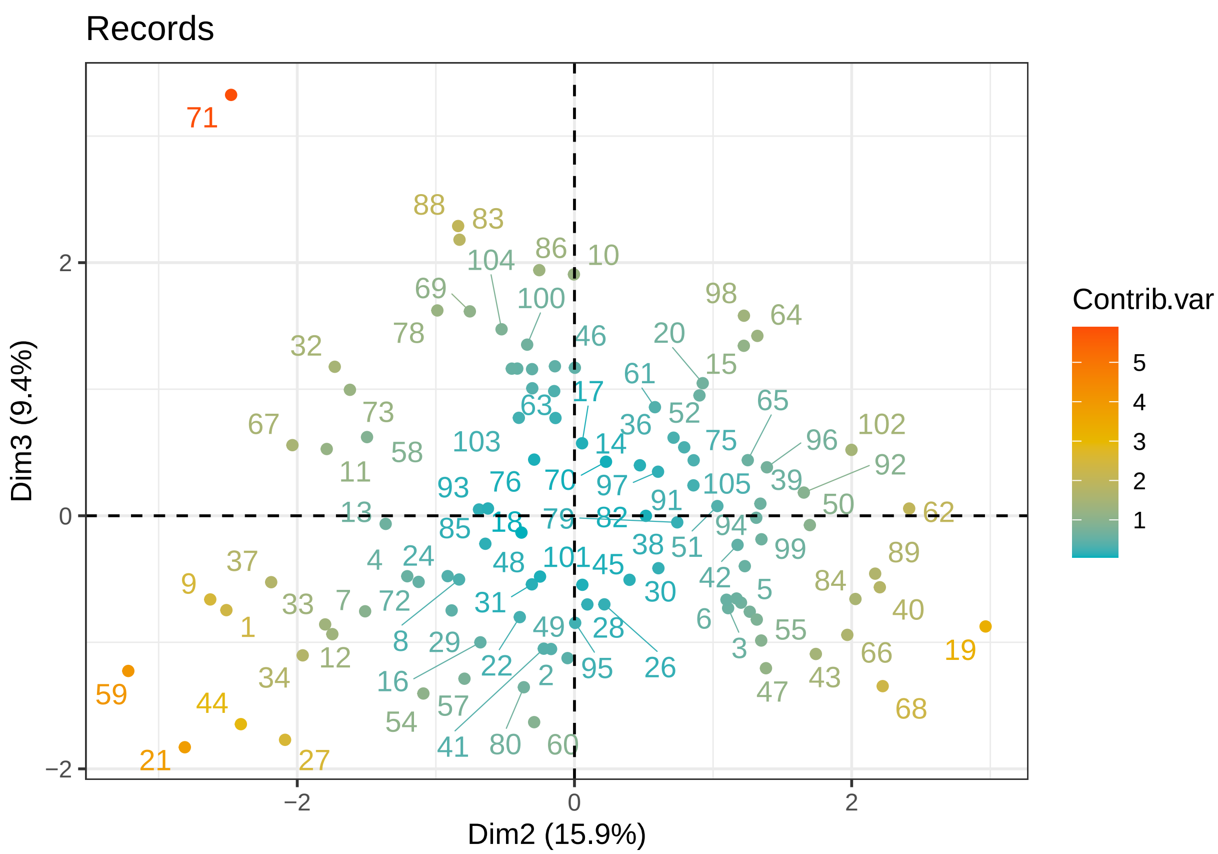


Observations and variance contribution

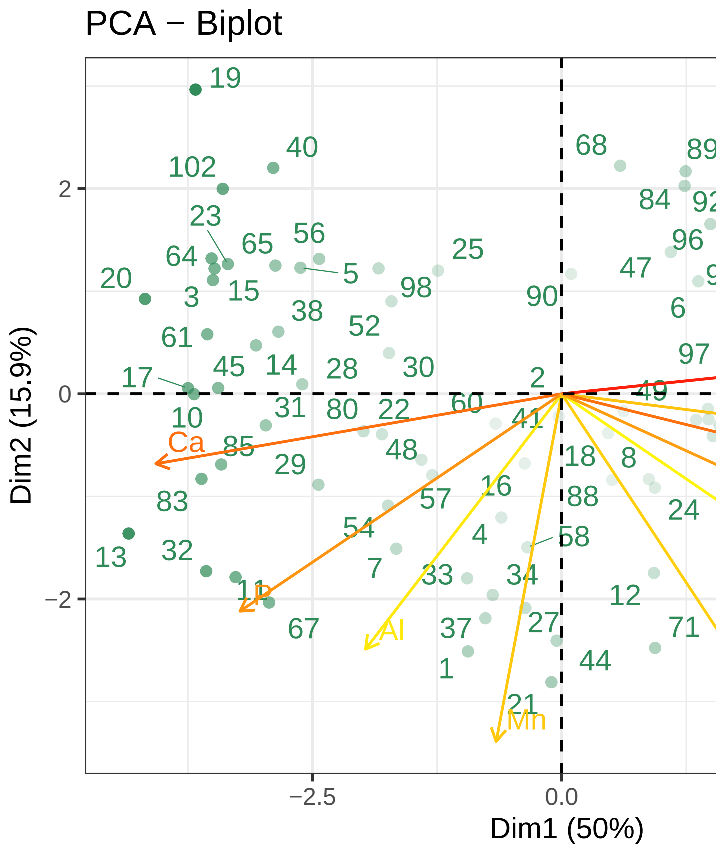
```
p2
```




```
p3
```

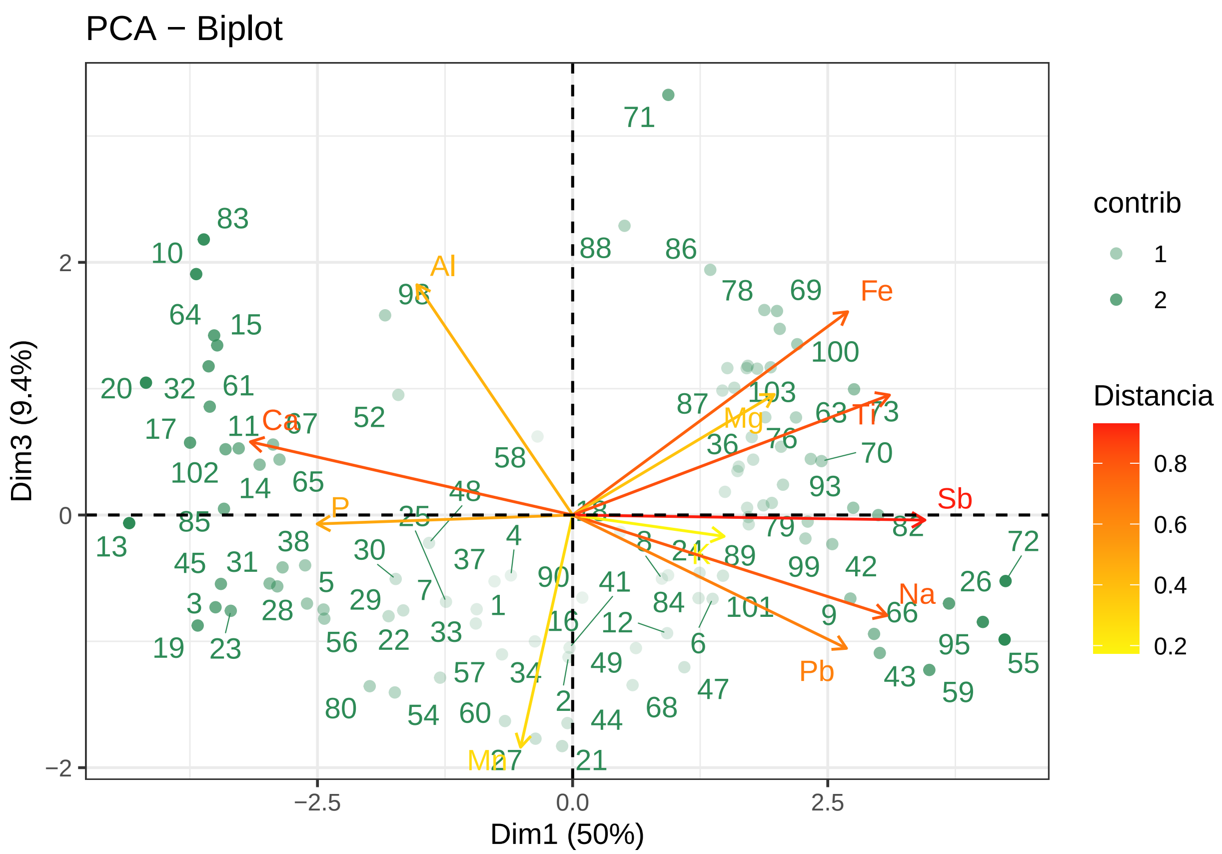



```
# Joint representation of variables and observations
# Relates the possible relationships between the contributions of the records
# to the variances of the components and the weight of the variables in each
# principal component

p1<-fviz_pca(PCA,alpha.ind ="contrib", col.var = "cos2",
         col.ind="seagreen",
         gradient.cols = c("#FDF50E", "#FD960E", "#FD1E0E"),
         repel=TRUE, legend.title="Distancia")+theme_bw()

p2<-fviz_pca(PCA,axes=c(1,3),alpha.ind ="contrib",
         col.var = "cos2",col.ind="seagreen",
         gradient.cols = c("#FDF50E", "#FD960E", "#FD1E0E"),
         repel=TRUE, legend.title="Distancia")+theme_bw()

p3<-fviz_pca(PCA,axes=c(2,3),alpha.ind ="contrib",
         col.var = "cos2",col.ind="seagreen",
         gradient.cols = c("#FDF50E", "#FD960E", "#FD1E0E"),
         repel=TRUE, legend.title="Distancia")+theme_bw()

# Displaying graphics
p1
```

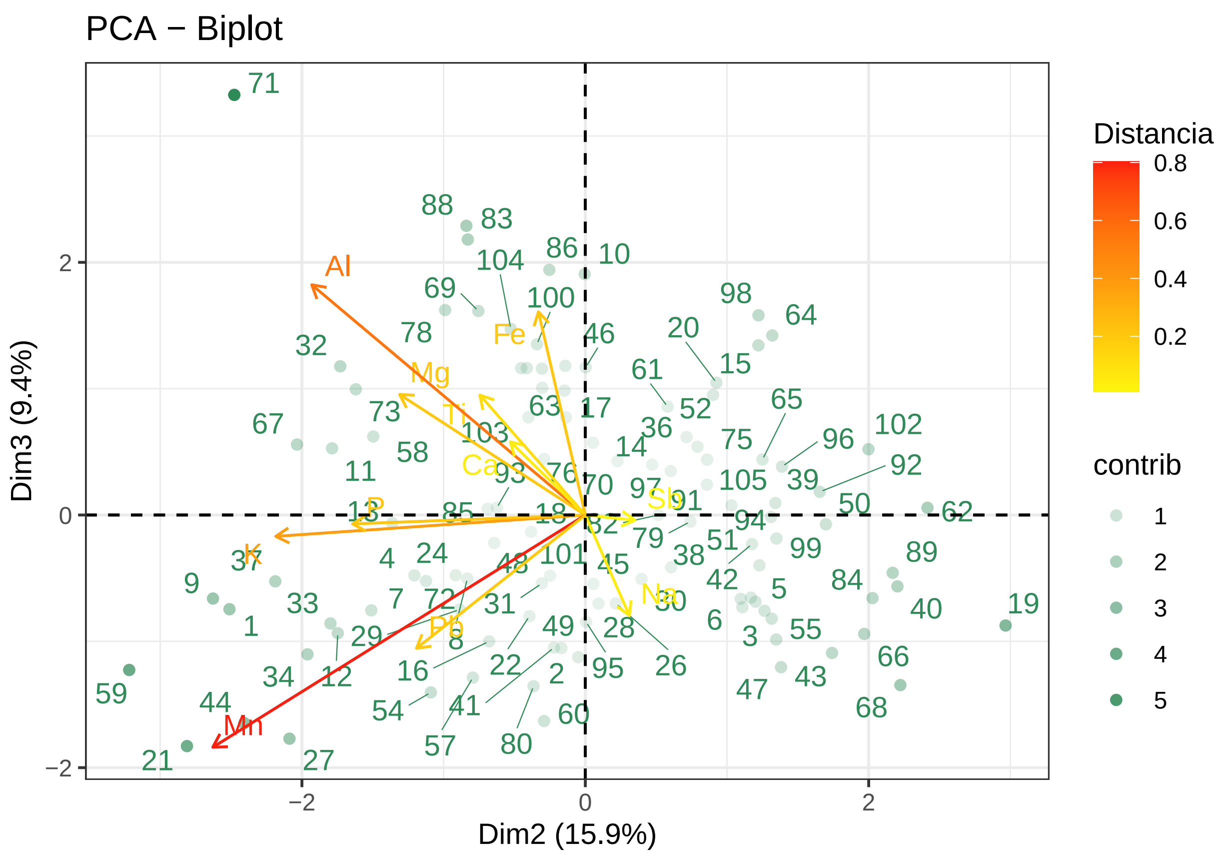


Observations and variables with variance contribution

```
p2
```


PCA − Biplot
Dim1 (50%)
Dim3 (9.4%)
contrib
1
2
Distancia
0.8
0.6
0.4
0.2
Al
Fe
Mg
Ti
Sb
Na
Pb
Ca
P
K
Mn

```
p3
```



Coordinates in the new reference system Finally, since the object of this study was to reduce the dimension of the data set, it is possible to obtain **the coordinates of the original data in the new reference system**. In fact, they are stored since we used the prcomp function to create the PCA variable.

```
head(PCA$x)
```

```
##               PC1         PC2        PC3        PC4         PC5         PC6
## [1,] -0.94001802 -2.51132893 -0.7456995  0.4974415  0.07504477  0.05650645
## [2,] -0.04072644 -0.05001002 -1.1250587  0.3697974  0.59460209  0.03148142
## [3,] -3.49863686  1.10849521 -0.7301384 -0.1361208 -0.45041316  0.39728340
## [4,] -0.60478971 -1.20618733 -0.4780341  0.5487647 -0.43070165  0.02375602
## [5,] -2.62074455  1.22902378 -0.3982816 -0.1666293 -0.82855657  0.22844532
## [6,]  1.37166061  1.09678306 -0.6636190 -0.5392607  0.43348070 -0.15302293
##             PC7        PC8         PC9        PC10       PC11
## [1,] -0.4709274  0.3893767 -0.22882181 -0.08716962  0.1166195
## [2,]  0.4308703 -0.0169307 -0.06987872  0.20569051  0.2457852
## [3,] -0.2073867 -1.1220984 -0.10609754  0.75593237 -0.2017568
## [4,] -0.6714851  0.3418526 -0.35566838  0.10547278  0.3649050
## [5,]  0.6959735 -0.6880391 -0.17556172  0.04048515 -0.3493687
## [6,] -0.1444326  0.2877034 -0.37880102  0.26333004  0.3574826
```