Principal component analysis - Practice 1.2

José Luis Romero Béjar

2023-11-19

In this guide, a second example of dimensionality reduction in a dataset is performed using the R language. To carry out this practice you must download the following files available on the PRADO platform of the course:

- *PCA_1.2.Rmd*
- *DB_PCA_1.2.sav*

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

- Exploratory data analisys to identify outliers and not available data (NA).
- Dealing with outliers: identification and decision making.
- Dealing with not available data (NA): identification and decision making.
- 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 set

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)
```

```r
library(ggplot2)

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

# 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)
```

Description of the data set *DB_PCA_1.2.sav*

The following code chunk shows how to load the dataset with IBMS SPSS format (.sav).

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_PCA_1.2.sav",to.data.frame=TRUE,reencode="latin1")

# Only if the computers current locale encoding causes problems with read.spss
# library(haven)
# data_spss<-read_sav("DB_PCA_1.2.sav")

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

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

The file *DB_PCA_1.2.sav* contains, among others, the variables *znac_def, zmortinf, zfertil, zinc_pob, rate_na, zurbana, zalfabet, zcaloría, zlog_pib, zpib_cap, zpoblac, zdensida*, which are the standardized variables of the original ones with the same label but without the initial *z*, and which respectively are the values for each country in the world of:

- Births/Deaths Rate (nac_def)
- Infant mortality: deaths per 1000 live births (mortinf)
- Fertility: average number of children (fertile)
- Population increase in annual % (inc_pop)
- Birth rate per 1,000 inhabitants (tasa_na)
- Inhabitants in cities in % (urban)

- Literate People in % (alfabet)
- Daily calorie intake (calories)
- Log(10) of GDP_CAP (log_pib)
- Gross domestic product per capita (gdp_cap)
- Population in thousands (poblac)
- Inhabitants per km2 (density)

Basic descriptive statistics

In this section, a preliminary exploratory data analysis of the data set 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 **categorical** variables their **frequency table** and a **sector and bar diagram** are provided.

Exploring the data set

```
# 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_spss)

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

```
##  [1] "país"      "poblac"    "densidad"  "urbana"    "relig"     "espvidaf"
##  [7] "espvidam"  "alfabet"   "inc_pob"   "mortinf"   "pib_cap"   "región"
## [13] "calorías"  "sida"      "tasa_nat"  "tasa_mor"  "tasasida"  "log_pib"
## [19] "logtsida"  "nac_def"   "fertilid"  "log_pob"   "cregrano"  "alfabmas"
## [25] "alfabfem"  "clima"     "region2"   "uso7"      "logden"    "patalfaf"
## [31] "pat_cal"   "znac_def"  "zmortinf"  "zfertil"   "zinc_pob"  "ztasa_na"
## [37] "zurbana"   "zespvida"  "zalfabet"  "zcaloría"  "zlog_pib"  "zpib_cap"
## [43] "zpoblac"   "zdensida"  "zlog_pob"  "zlogden"   "pib_CATEG" "PIB_Grupo"
```

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

```
##                país poblac densidad urbana    relig espvidaf espvidam alfabet
## 1  Acerbaján          7400     86.0     54 Musulma.       75       67      98
## 2  Afganistán        20500     25.0     18 Musulma.       44       45      29
## 3  Alemania          81200    227.0     85 Protest.       79       73      99
## 4  Arabia Saudí      18000      7.7     77 Musulma.       70       66      62
## 5  Argentina         33900     12.0     86 Católica       75       68      95
## 6  Armenia            3700    126.0     68 Ortodoxa       75       68      98
## 7  Australia         17800      2.3     85 Protest.       80       74     100
## 8  Austria            8000     94.0     58 Católica       79       73      99
## 9  Bahrein             600    828.0     83 Musulma.       74       71      77
## 10 Bangladesh       125000    800.0     16 Musulma.       53       53      35
##    inc_pob mortinf pib_cap          región calorías  sida tasa_nat tasa_mor
## 1     1.40    35.0    3000   Oriente Medio       NA    NA       23        7
## 2     2.80   168.0     205 Asia / Pacífico       NA     0       53       22
## 3     0.36     6.5   17539            OCDE     3443 11179       11       11
## 4     3.20    52.0    6651   Oriente Medio     2874    61       38        6
## 5     1.30    25.6    3408  América Latina     3113  3904       20        9
## 6     1.40    27.0    5000   Oriente Medio       NA     2       23        6
## 7     1.38     7.3   16848            OCDE     3216  4727       15        8
```

```
## 8     0.20     6.7   18396            OCDE     3495  1150       12       11
## 9     2.40    25.0    7875   Oriente Medio       NA    13       29        4
## 10    2.40   106.0     202 Asia / Pacífico     2021     1       35       11
##        tasasida  log_pib  logtsida  nac_def fertilid  log_pob cregrano alfabmas
## 1            NA 3.477121        NA 3.285714     2.80 3.869232       18      100
## 2  0.000000e+00 2.311754 0.0000000 2.409091     6.90 4.311754       12       44
## 3  1.376724e+01 4.244005 1.6895435 1.000000     1.47 4.909556       34       NA
## 4  3.388889e-01 3.822887 0.8053997 6.333333     6.67 4.255273        1       73
## 5  1.151622e+01 3.532500 1.6302793 2.222222     2.80 4.530200        9       96
## 6  5.405405e-02 3.698970 0.5579119 3.833333     3.19 3.568202       17      100
## 7  2.655618e+01 4.226548 1.9267844 1.875000     1.90 4.250420        6      100
## 8  1.437500e+01 4.264723 1.7042040 1.090909     1.50 3.903090       17       NA
## 9  2.166667e+00 3.896251 1.1672353 7.250000     3.96 2.778151        2       55
## 10 8.576329e-04 2.305351 0.2435905 3.181818     4.70 5.096910       67       47
##    alfabfem    clima         region2 uso7    logden patalfaf  pat_cal
## 1       100    árido   Oriente Medio    0 1.9344985 presente  ausente
## 2        14    árido Asia / Pacífico    1 1.3979400 presente  ausente
## 3        NA templado          Europa    1 2.3560259  ausente presente
## 4        48 desierto   Oriente Medio    1 0.8864907 presente presente
## 5        95 templado  América Latina    0 1.0791812 presente presente
## 6       100     <NA>   Oriente Medio    0 2.1003705 presente  ausente
## 7       100    árido Asia / Pacífico    1 0.3617278 presente presente
## 8        NA templado          Europa    0 1.9731279  ausente presente
## 9        55    árido   Oriente Medio    0 2.9180303 presente  ausente
## 10       22 tropical Asia / Pacífico    0 2.9030900 presente presente
##       znac_def   zmortinf    zfertil   zinc_pob   ztasa_na     zurbana
## 1   0.03868813 -0.1920453 -0.4011015 -0.2358004 -0.2364663 -0.10443910
## 2  -0.37384556  3.3006728  1.7539901  0.9332426  2.1905419 -1.59183548
## 3  -1.03695529 -0.9404849 -1.1001922 -1.1042324 -1.2072696  1.17637445
## 4   1.47287936  0.2543924  1.6330948  1.2672549  0.9770378  0.84584192
## 5  -0.46178485 -0.4388991 -0.4011015 -0.3193035 -0.4791671  1.21769102
## 6   0.29639437 -0.4021336 -0.1961050 -0.2358004 -0.2364663  0.47399283
## 7  -0.62518554 -0.9194761 -0.8741704 -0.2525010 -0.8836685  1.17637445
## 8  -0.99417401 -0.9352327 -1.0844233 -1.2378373 -1.1263693  0.06082717
## 9   1.90425719 -0.4546557  0.2086317  0.5992303  0.2489353  1.09374132
## 10 -0.01020475  1.6724884  0.5975995  0.5992303  0.7343370 -1.67446861
##      zespvida    zalfabet   zcaloría    zlog_pib   zpib_cap     zpoblac
## 1   0.4582045  0.85930109         NA  0.08919672 -0.4413664 -0.27482369
## 2  -2.4741310 -2.15601258         NA -1.79026124 -0.8727045 -0.18554185
## 3   0.8365704  0.90300129  1.2137014  1.32599590  1.8023633  0.22815340
## 4  -0.0147528 -0.71390604  0.2116370  0.64683381  0.1220738 -0.20258037
## 5   0.4582045  0.72820050  0.6325393  0.17850868 -0.3784018 -0.09421538
## 6   0.4582045  0.85930109         NA  0.44698551 -0.1327166 -0.30004070
## 7   0.9311619  0.94670149  0.8139323  1.29784279  1.6957248 -0.20394345
## 8   0.8365704  0.90300129  1.3052785  1.35940998  1.9346197 -0.27073445
## 9   0.3636131 -0.05840307         NA  0.76515171  0.3109675 -0.32116846
## 10 -1.6228078 -1.89381139 -1.2905792 -1.80058693 -0.8731674  0.52666826
##       zdensida   zlog_pob    zlogden     pib_CATEG PIB_Grupo
## 1  -0.17376613 -0.3741906  0.2412855                       0
## 2  -0.26404219  0.3022811 -0.6190089                       0
## 3   0.03490475  1.2161251  0.9171441 _duplicated_2         1
## 4  -0.28964507  0.2159396 -1.4390443                       0
## 5  -0.28328135  0.6362133 -1.1300927                       0
## 6  -0.11456872 -0.8343670  0.5072375                       0
```

```
## 7  -0.29763672  0.2085217 -2.2804261 _duplicated_2         1
## 8  -0.16192665 -0.3224323  0.3032221 _duplicated_2         1
## 9   0.92434589 -2.0420958  1.8182374                      0
## 10  0.88290770  1.5025281  1.7942827                      0
```

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.28043 -0.51566  0.03555  0.00000  0.50724  3.13597
```

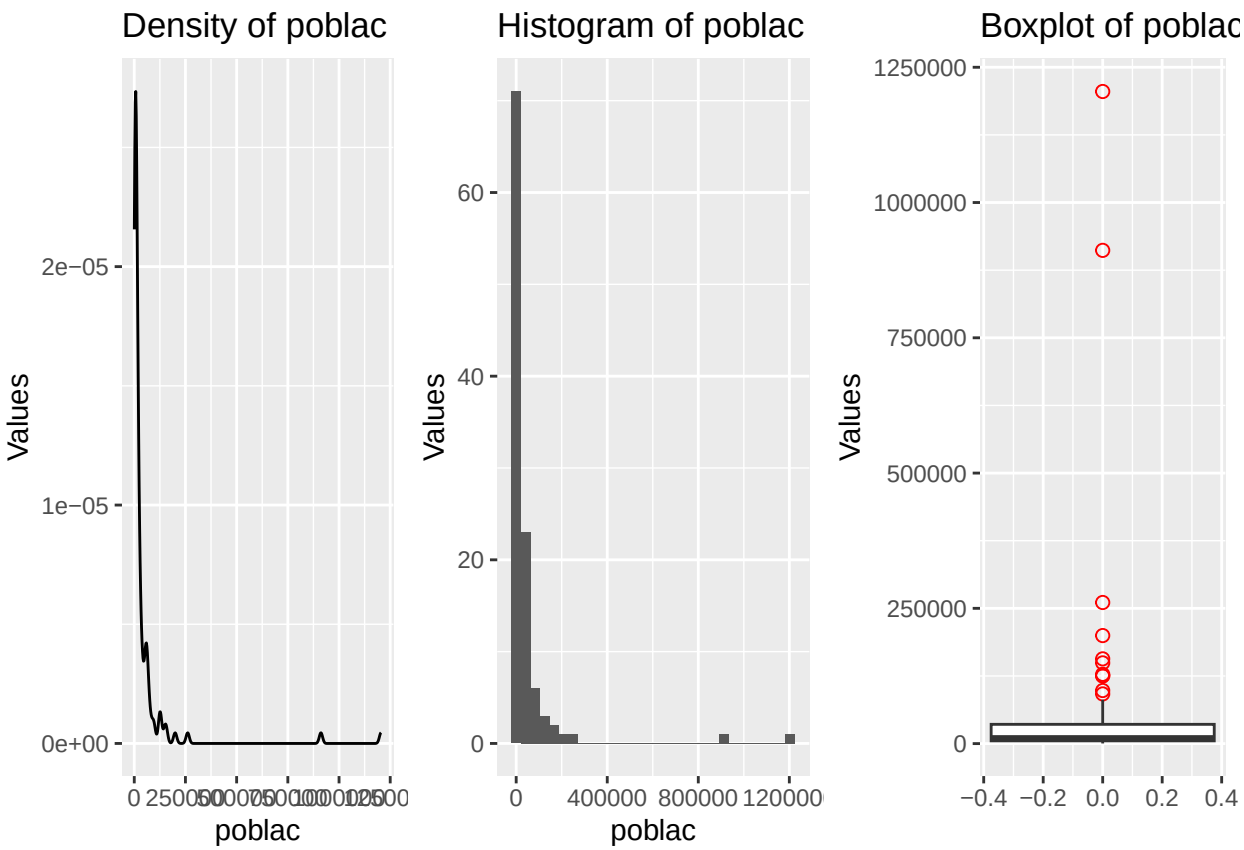
Descriptive analysis (numerical and graphical)

poblac - Population

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

```
## Descriptive Statistics
## poblac
## N: 109
##
##                        poblac
## ----------------- ------------
##              Mean     47723.88
##           Std.Dev    146726.36
##               Min       256.00
##                Q1      5100.00
##            Median     10400.00
##                Q3     35600.00
##               Max   1205200.00
##               MAD     11267.76
##               IQR     30500.00
##                CV         3.07
##          Skewness         6.41
##       SE.Skewness         0.23
##          Kurtosis        43.62
##           N.Valid       109.00
##         Pct.Valid       100.00
```

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

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

p3<-ggplot(data_spss,aes(x=poblac))+
  geom_boxplot(outlier.colour="red", outlier.shape=1,outlier.size=2)+
  coord_flip()+labs(title = "Boxplot of poblac",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)
```



Assignment

Replicates the previous output for the varaible *poblac* for variables *densidad, urbana, relig, espvidaf, espvidam, alfabet, inc_pob, mortinf, pib_cap, región, calorías, sida, tasa_nat, tasa_mor, tasasida, nac_def* and *fertilid*. It must be taken into account that variables *relig* and *región* are categorical. It implies that the previous output is not adequate (see the variable *site* in the file *PCA_1.1.Rmd* to proceed in the same way with these two variables).

Variables selection

The data frame *data_spss* has 48 variables. For this illustration we are interested in the standardized variables whose labels begin with *z*. These variables are in the columns 32 to 46.

Next code chunk defines a new data frame object with the only fifteen variables of interest for this illustration.

```
# The first 31 variables are eliminated.
data_pca<-data_spss[,-(1:31)]

# The last two variables seem to be duplicates or with irrelevant information.
data_pca<-data_pca[,-(16:17)]


# The first three records in the database are displayed
head(data_pca,n=3)
```

```
##      znac_def   zmortinf    zfertil   zinc_pob   ztasa_na    zurbana   zespvida
```

```
## 1  0.03868813 -0.1920453 -0.4011015 -0.2358004 -0.2364663 -0.1044391  0.4582045
## 2 -0.37384556  3.3006728  1.7539901  0.9332426  2.1905419 -1.5918355 -2.4741310
## 3 -1.03695529 -0.9404849 -1.1001922 -1.1042324 -1.2072696  1.1763745  0.8365704
##     zalfabet zcaloría    zlog_pib   zpib_cap    zpoblac    zdensida   zlog_pob
## 1  0.8593011       NA  0.08919672 -0.4413664 -0.2748237 -0.17376613 -0.3741906
## 2 -2.1560126       NA -1.79026124 -0.8727045 -0.1855418 -0.26404219  0.3022811
## 3  0.9030013 1.213701  1.32599590  1.8023633  0.2281534  0.03490475  1.2161251
##      zlogden
## 1  0.2412855
## 2 -0.6190089
## 3  0.9171441
```

Not available data (NA)

Identification and treatment

The decision for not available data is to replace them by the mean of their variable. This decision has been made assuming that the behavior of the *NA* is totally random (this would have to be analyzed in depth to confirm this decision made). 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 *not_available* whose utility is to deal with not available data.

```
# Construction of the function that handles missing values.
not_available<-function(data,na.rm=F){
  data[is.na(data)]<-mean(data,na.rm=T)
  data
}

# We call the not_available function for each variable in the database
data_pca$znac_def<-not_available(data_pca$znac_def)
data_pca$zmortinf<-not_available(data_pca$zmortinf)
data_pca$zfertil<-not_available(data_pca$zfertil)
data_pca$zinc_pob<-not_available(data_pca$zinc_pob)
data_pca$ztasa_na<-not_available(data_pca$ztasa_na)
data_pca$zurbana<-not_available(data_pca$zurbana)
data_pca$zespvida<-not_available(data_pca$zespvida)
data_pca$zalfabet<-not_available(data_pca$zalfabet)
data_pca$zcaloría<-not_available(data_pca$zcaloría)
data_pca$zlog_pib<-not_available(data_pca$zlog_pib)
data_pca$zpib_cap<-not_available(data_pca$zpib_cap)
data_pca$zpoblac<-not_available(data_pca$zpoblac)
data_pca$zdensida<-not_available(data_pca$zdensida)
data_pca$zlog_pob<-not_available(data_pca$zlog_pob)
data_pca$zlogden<-not_available(data_pca$zlogden)

# We view the data again
head(data_pca,n=3)
```

```
##      znac_def   zmortinf    zfertil   zinc_pob   ztasa_na    zurbana   zespvida
## 1  0.03868813 -0.1920453 -0.4011015 -0.2358004 -0.2364663 -0.1044391  0.4582045
## 2 -0.37384556  3.3006728  1.7539901  0.9332426  2.1905419 -1.5918355 -2.4741310
## 3 -1.03695529 -0.9404849 -1.1001922 -1.1042324 -1.2072696  1.1763745  0.8365704
##     zalfabet     zcaloría    zlog_pib   zpib_cap    zpoblac    zdensida
## 1  0.8593011 5.128681e-16  0.08919672 -0.4413664 -0.2748237 -0.17376613
```

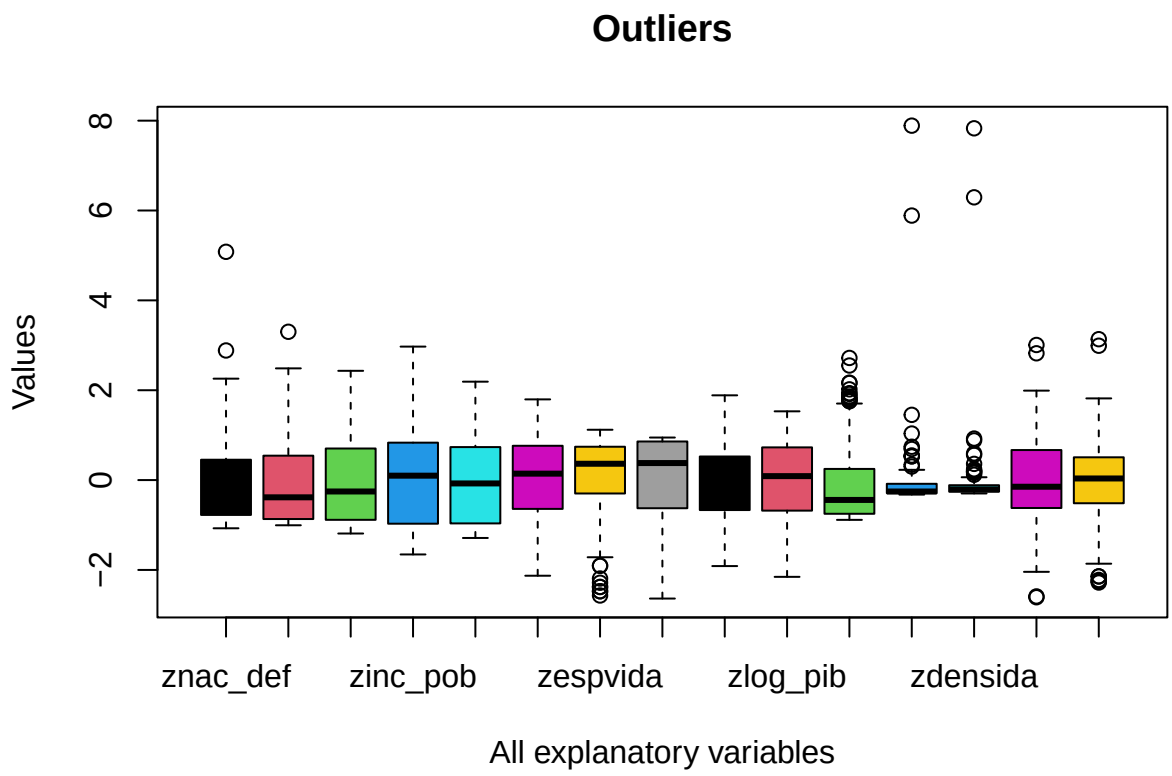
```
## 2 -2.1560126 5.128681e-16 -1.79026124 -0.8727045 -0.1855418 -0.26404219
## 3  0.9030013 1.213701e+00  1.32599590  1.8023633  0.2281534  0.03490475
##     zlog_pob    zlogden
## 1 -0.3741906  0.2412855
## 2  0.3022811 -0.6190089
## 3  1.2161251  0.9171441
```

Outliers

Identification

This graphical output shows together the boxplots of all the quantitative variables. Since all the variables are standardized there is no problem with the scales.

```
# Boxplots of all variables together
# This visualization is not the best due to the difference between the scales
boxplot(data_pca,main="Outliers",
        xlab="All explanatory variables",
        ylab="Values",
        col=c(1:15))
```



Making decisions

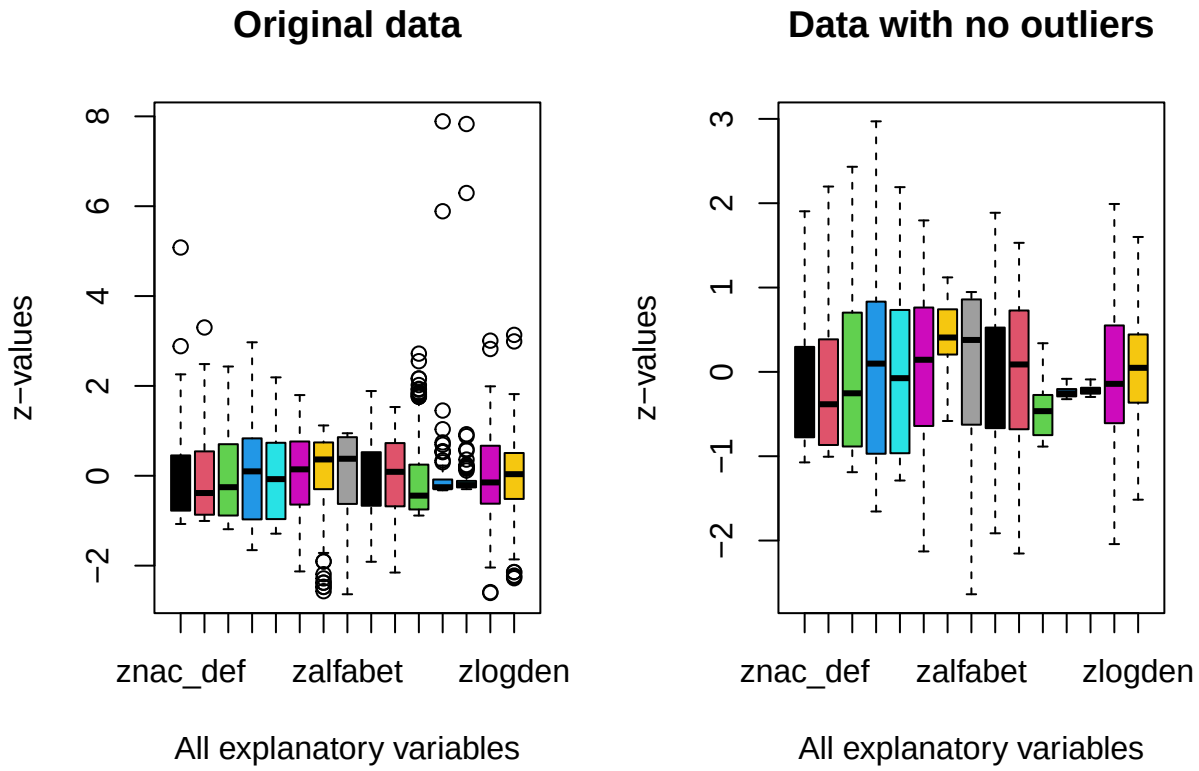
From previous graphical outputs it is noticed the presence of outliers for several variables. It is relevant to take into account these values since multivariate methods, such as principal component analisis (PCA), are sensitive to this fact.

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

    return(data)

}

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

# Called to outlier function for each variable identified with outliers
data_pca_aux$znac_def<-outlier(data_pca_aux$znac_def)
data_pca_aux$zmortinf<-outlier(data_pca_aux$zmortinf)
data_pca_aux$zespvida<-outlier(data_pca_aux$zespvida)
data_pca_aux$zpib_cap<-outlier(data_pca_aux$zpib_cap)
data_pca_aux$zpoblac<-outlier(data_pca_aux$zpoblac)
data_pca_aux$zdensida<-outlier(data_pca_aux$zdensida)
data_pca_aux$zlog_pob<-outlier(data_pca_aux$zlog_pob)
data_pca_aux$zlogden<-outlier(data_pca_aux$zlogden)

# We compare the original data and the fixed ones with respective boxplots
par(mfrow=c(1,2))
# Boxplot original data
boxplot(data_pca,main="Original data",
        xlab="All explanatory variables",
        ylab="z-values",
        col=c(1:15))
# Boxplot fixed data
boxplot(data_pca_aux,main="Data with no outliers",
        xlab="All explanatory variables",
        ylab="z-values",
        col=c(1:15))
```



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).

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

# Are the variables correlated at sample level?
correlation_matrix<-cor(data_pca_aux)
correlation_matrix
```

```
##             znac_def    zmortinf     zfertil    zinc_pob    ztasa_na
## znac_def  1.00000000  0.31249772  0.46816617  0.67292672  0.54238244
## zmortinf  0.31249772  1.00000000  0.81789418  0.61089102  0.83350821
## zfertil   0.46816617  0.81789418  1.00000000  0.83540114  0.96752460
## zinc_pob  0.67292672  0.61089102  0.83540114  1.00000000  0.86125623
## ztasa_na  0.54238244  0.83350821  0.96752460  0.86125623  1.00000000
## zurbana  -0.13486029 -0.71715045 -0.61312846 -0.37214616 -0.62583339
## zespvida -0.54851563 -0.58958055 -0.57573016 -0.58697985 -0.63729277
## zalfabet -0.33899217 -0.84810568 -0.85098793 -0.68712950 -0.85919685
## zcaloría -0.33647562 -0.67485805 -0.59385908 -0.48103785 -0.63584336
```

```
## zlog_pib -0.38018624 -0.80729451 -0.69164346 -0.55706858 -0.76852216
## zpib_cap -0.26741107 -0.60564349 -0.48348001 -0.41612342 -0.58019460
## zpoblac   0.11204240  0.11521228  0.02693397  0.02268286  0.07322864
## zdensida -0.16677102 -0.10863878 -0.23872659 -0.27597225 -0.24856606
## zlog_pob -0.04416102  0.06478885 -0.10456755 -0.13077994 -0.07181053
## zlogden  -0.16831258 -0.12990515 -0.23556917 -0.27730398 -0.24114966
##              zurbana   zespvida    zalfabet     zcaloría    zlog_pib
## znac_def -0.13486029 -0.5485156 -0.33899217 -0.336475616 -0.38018624
## zmortinf -0.71715045 -0.5895806 -0.84810568 -0.674858053 -0.80729451
## zfertil  -0.61312846 -0.5757302 -0.85098793 -0.593859084 -0.69164346
## zinc_pob -0.37214616 -0.5869799 -0.68712950 -0.481037846 -0.55706858
## ztasa_na -0.62583339 -0.6372928 -0.85919685 -0.635843360 -0.76852216
## zurbana   1.00000000  0.4754899  0.63862105  0.584511574  0.75235758
## zespvida  0.47548992  1.0000000  0.48508492  0.542916764  0.64184985
## zalfabet  0.63862105  0.4850849  1.00000000  0.564406375  0.72874071
## zcaloría  0.58451157  0.5429168  0.56440637  1.000000000  0.75109373
## zlog_pib  0.75235758  0.6418498  0.72874071  0.751093729  1.00000000
## zpib_cap  0.52568861  0.4039970  0.55709083  0.559931186  0.75745427
## zpoblac  -0.07004911 -0.1241143 -0.05156648 -0.006842524 -0.21303114
## zdensida -0.11565426  0.1426419  0.09853886  0.177167508 -0.01676850
## zlog_pob -0.05506122 -0.0611491  0.03095859  0.085208098 -0.10582778
## zlogden  -0.02476024  0.1814206  0.13371010  0.087478756  0.04973048
##             zpib_cap      zpoblac    zdensida    zlog_pob     zlogden
## znac_def -0.26741107  0.112042403 -0.16677102 -0.04416102 -0.16831258
## zmortinf -0.60564349  0.115212276 -0.10863878  0.06478885 -0.12990515
## zfertil  -0.48348001  0.026933971 -0.23872659 -0.10456755 -0.23556917
## zinc_pob -0.41612342  0.022682859 -0.27597225 -0.13077994 -0.27730398
## ztasa_na -0.58019460  0.073228641 -0.24856606 -0.07181053 -0.24114966
## zurbana   0.52568861 -0.070049106 -0.11565426 -0.05506122 -0.02476024
## zespvida  0.40399697 -0.124114262  0.14264185 -0.06114910  0.18142062
## zalfabet  0.55709083 -0.051566485  0.09853886  0.03095859  0.13371010
## zcaloría  0.55993119 -0.006842524  0.17716751  0.08520810  0.08747876
## zlog_pib  0.75745427 -0.213031137 -0.01676850 -0.10582778  0.04973048
## zpib_cap  1.00000000 -0.235672820  0.05431088 -0.18279177  0.04762462
## zpoblac  -0.23567282  1.000000000 -0.01831188  0.68117954 -0.06047899
## zdensida  0.05431088 -0.018311881  1.00000000  0.14324163  0.64506688
## zlog_pob -0.18279177  0.681179536  0.14324163  1.00000000  0.11575103
## zlogden   0.04762462 -0.060478987  0.64506688  0.11575103  1.00000000
```

```r
det(correlation_matrix)
```

```
## [1] 3.577126e-07
```

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

```
## [1] -0.8591968
```

```r
####################################
# 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
# R denotes the correlation matrix
# cortest.bartlett function in the package pysch performs this test
# This function works with standardized data.

# Standardization
data_pca_aux_scale<-scale(data_pca_aux)

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

```
## $chisq
## [1] 1382.923
##
## $p.value
## [1] 8.632594e-222
##
## $df
## [1] 105
```

Absence of outliers

Done in **Section 2.4.2** in the data.frame *data_acp_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_pca_aux, 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
## znac_def -0.2033782257 -0.17091077  0.173946280  0.66238262  0.20076981
## zmortinf -0.3328825609  0.09031952 -0.066900498 -0.16816166 -0.14852363
## zfertil  -0.3382183478 -0.11514469 -0.047823141  0.01693219 -0.34300033
## zinc_pob -0.3005657583 -0.21949445  0.057004542  0.30890083 -0.22493343
## ztasa_na -0.3543220078 -0.09083238 -0.009741712  0.04987721 -0.22123594
## zurbana   0.2689803646 -0.22707409  0.209852968  0.24005478 -0.00246764
## zespvida  0.2701965422  0.01262584 -0.093520748 -0.26117405 -0.37357056
## zalfabet  0.3224791887 -0.01661489  0.105956181  0.09405666  0.38714570
## zcaloría  0.2817928106 -0.02495913  0.115320783  0.15865219 -0.50585498
## zlog_pib  0.3281002535 -0.20314224  0.052014542  0.08109587 -0.22230823
## zpib_cap  0.2570822655 -0.21986463 -0.049651245  0.18136614 -0.28159144
## zpoblac  -0.0498228826  0.36351991  0.584806041  0.03028937 -0.10451731
## zdensida  0.0757733906  0.46049512 -0.381359069  0.33796438 -0.12780237
```

```
## zlog_pob -0.0008063132  0.47613897  0.478727210  0.01857923 -0.13695471
## zlogden   0.0826360669  0.42403556 -0.400677181  0.34636488 -0.01954228
##                   PC6          PC7         PC8          PC9         PC10
## znac_def  0.14510162 -0.105867242  0.21911857 -0.253129128  0.28639460
## zmortinf -0.10694040  0.117574727 -0.10009777 -0.113986846 -0.19994086
## zfertil  -0.02123533  0.123025638  0.01995051  0.045984208  0.02625726
## zinc_pob  0.13991751  0.004920553 -0.03811899  0.004512657  0.02161508
## ztasa_na  0.05188330  0.046413033 -0.01297816  0.072444135  0.03481406
## zurbana   0.36605554  0.179322730 -0.25699186 -0.073386802 -0.69508245
## zespvida  0.57068820  0.041117920  0.42639149 -0.341954108  0.21328810
## zalfabet -0.07704512 -0.013580617  0.05923839  0.070846137  0.23020638
## zcaloría -0.04914836 -0.463938996 -0.31834928  0.375436774  0.21377980
## zlog_pib -0.03693063  0.142301698 -0.10010738 -0.040777000  0.10599173
## zpib_cap -0.63193928  0.379538246  0.32394744 -0.146324613 -0.06835236
## zpoblac   0.02752821  0.064608905  0.51428956  0.430071891 -0.19253749
## zdensida -0.12000097 -0.449438894  0.17805337 -0.250035585 -0.35597884
## zlog_pob -0.11236030  0.165164159 -0.39294897 -0.514932907  0.19346801
## zlogden   0.21968695  0.557346919 -0.14720371  0.333991344  0.17657046
##                   PC11         PC12        PC13         PC14         PC15
## znac_def -0.426579523  0.035562136  0.01443274  0.089700749 -0.047428195
## zmortinf -0.432466937  0.634684295  0.18644227 -0.336375472  0.014433813
## zfertil   0.209550552  0.213758077 -0.07365997  0.536597324 -0.591922616
## zinc_pob  0.543762827 -0.055845217  0.12497759 -0.611172971 -0.052228984
## ztasa_na  0.142204839  0.120539192 -0.17780408  0.339182927  0.788537574
## zurbana  -0.033186799  0.095432883 -0.18705191  0.091605869 -0.010201130
## zespvida -0.019529283  0.113799961 -0.13215031 -0.087446268  0.016407984
## zalfabet  0.398360848  0.695736059 -0.12056666  0.013515956  0.025946135
## zcaloría -0.227272380  0.110459790 -0.21639843 -0.091211536 -0.030878512
## zlog_pib  0.026753046  0.042761912  0.82498879  0.219209931  0.129013546
## zpib_cap -0.052596729 -0.053511765 -0.27035785 -0.123578132  0.049993362
## zpoblac  -0.008847773  0.006756976  0.14107180 -0.006471395  0.002467309
## zdensida  0.188510064  0.075278920  0.12275400  0.105714038  0.037819094
## zlog_pob  0.121583762 -0.061220302 -0.09259136  0.031510374  0.003769110
## zlogden  -0.074058496 -0.010478361 -0.03622072 -0.047070923 -0.007489877
```

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

```
##  [1] 2.6924769 1.4274971 1.2708317 1.0183232 0.8509704 0.7350785 0.6415794
##  [8] 0.5803036 0.5134328 0.5080576 0.4474333 0.3431384 0.3154899 0.3004071
## [15] 0.1360262
```

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.

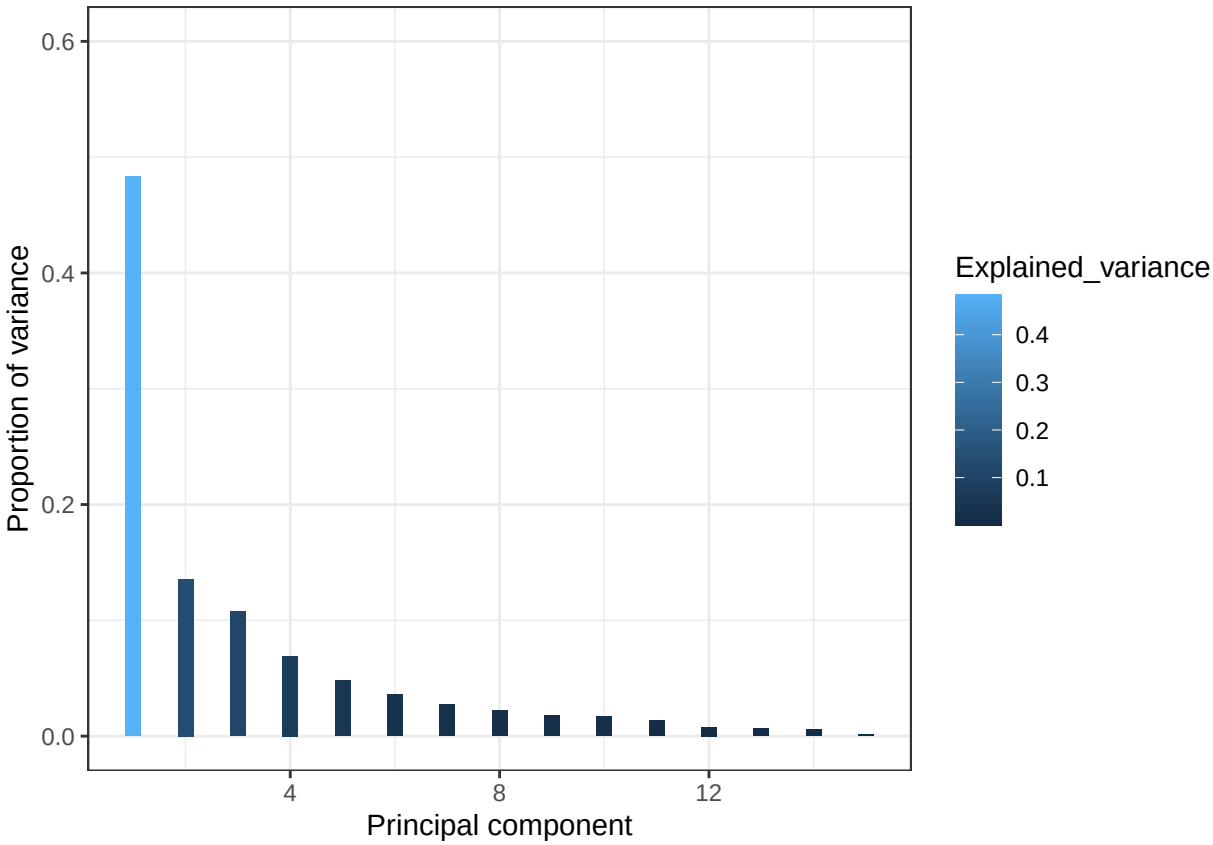
Explained variance rate

```
# 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)
```

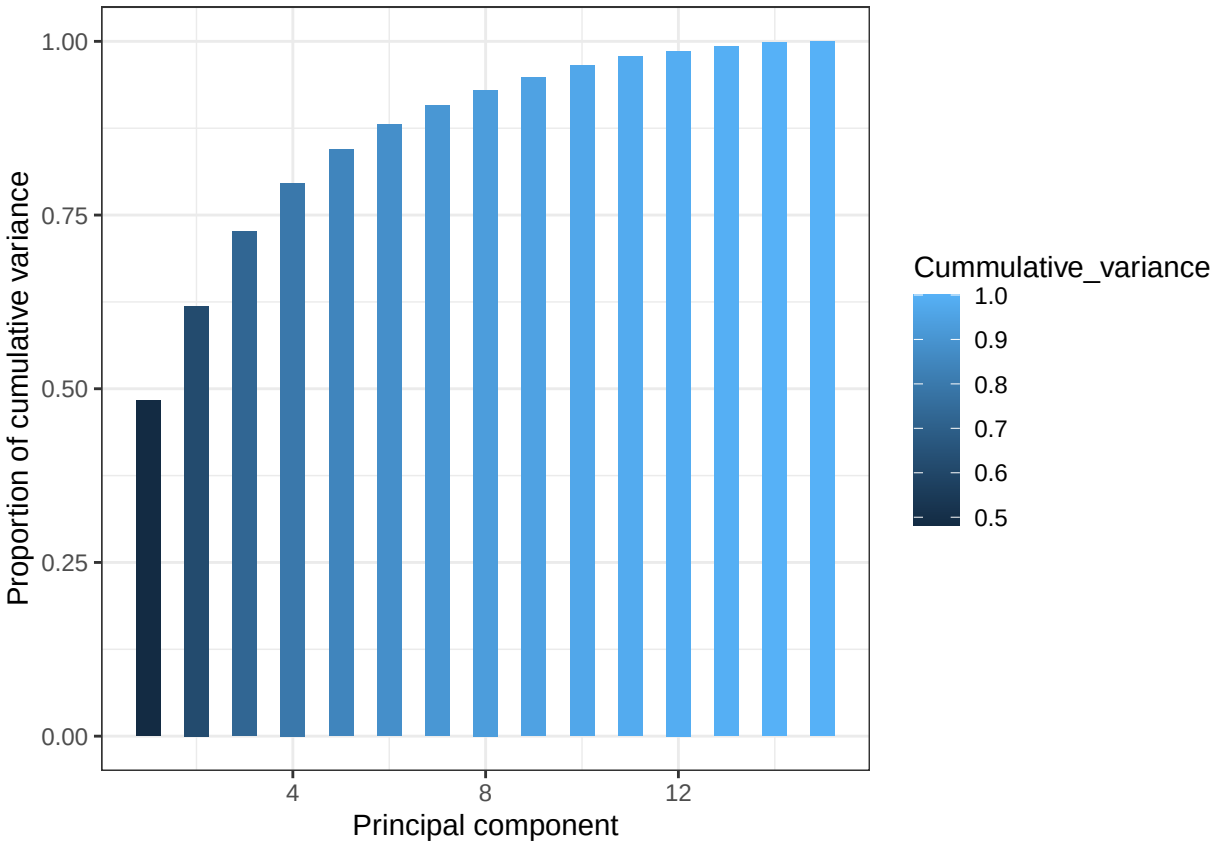
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.6925 1.4275 1.2708 1.01832 0.85097 0.73508 0.64158
```

```
## Proportion of Variance 0.4833 0.1358 0.1077 0.06913 0.04828 0.03602 0.02744
## Cumulative Proportion  0.4833 0.6191 0.7268 0.79594 0.84422 0.88024 0.90769
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.58030 0.51343 0.50806 0.44743 0.34314 0.31549 0.30041
## Proportion of Variance 0.02245 0.01757 0.01721 0.01335 0.00785 0.00664 0.00602
## Cumulative Proportion  0.93014 0.94771 0.96492 0.97826 0.98611 0.99275 0.99877
##                           PC15
## Standard deviation     0.13603
## Proportion of Variance 0.00123
## Cumulative Proportion  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 = 1:15),
  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 = 1:15),
  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
4
8
12
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 **four principal components are considered**, as can be deduced from the following code chunk.

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

# Variances
PCA$sdev^2
```

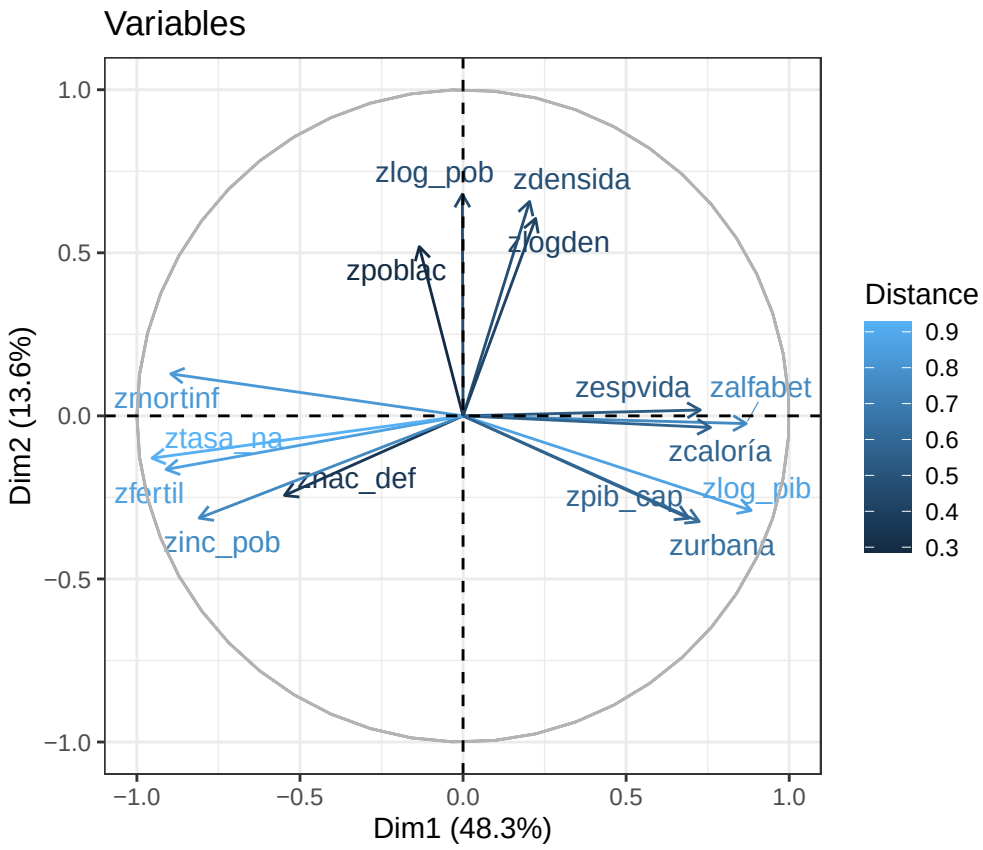
```
##  [1] 7.24943169 2.03774784 1.61501318 1.03698219 0.72415059 0.54034042
##  [7] 0.41162407 0.33675226 0.26361324 0.25812257 0.20019656 0.11774396
## [13] 0.09953386 0.09024445 0.01850312
```

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

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

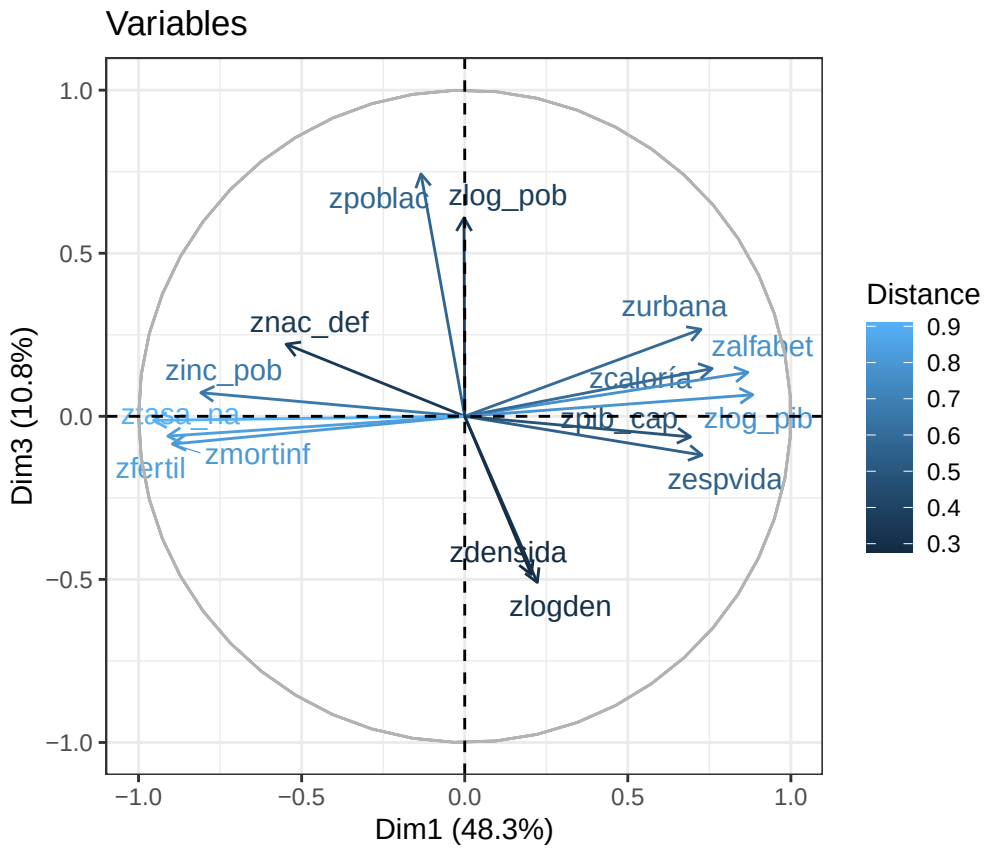
PCA graphical outputs of interest

```
# 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
```

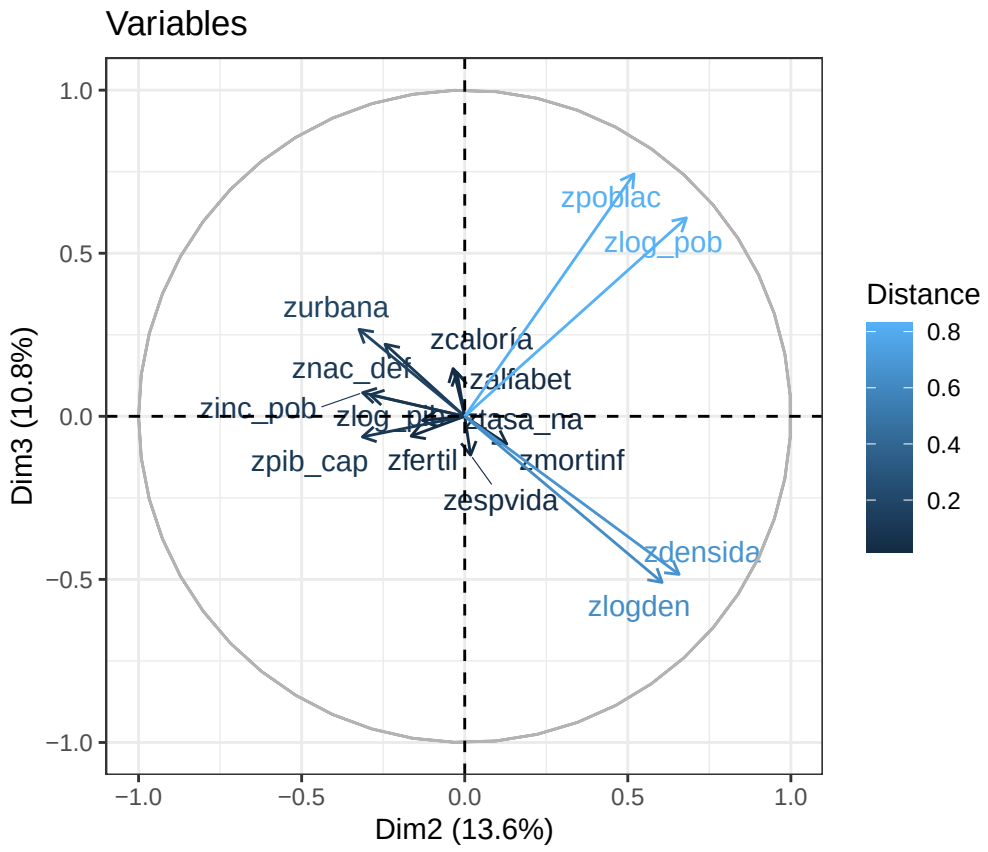


Distances

```
p2
```

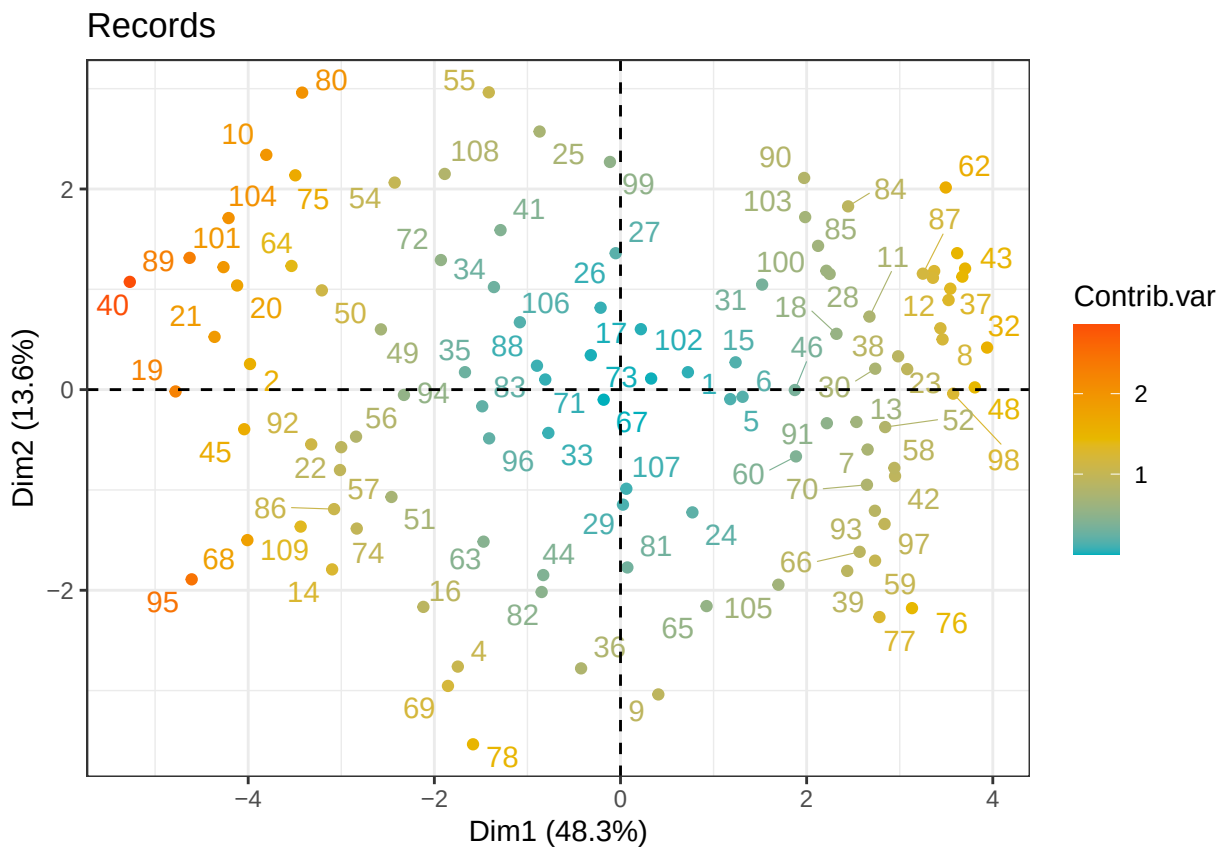

Variables
zpoblac
zlog_pob
znac_def
zurbana
zalfabet
zinc_pob
zcaloría
zesp_na
zpib_cap
zlog_pib
zfertil
zmortinf
zespvida
zdensida
zlogden
Distance
0.9
0.8
0.7
0.6
0.5
0.4
0.3
1.0
0.5
0.0
−0.5
−1.0
Dim3 (10.8%)
−1.0
−0.5
0.0
0.5
1.0
Dim1 (48.3%)

```
p3
```

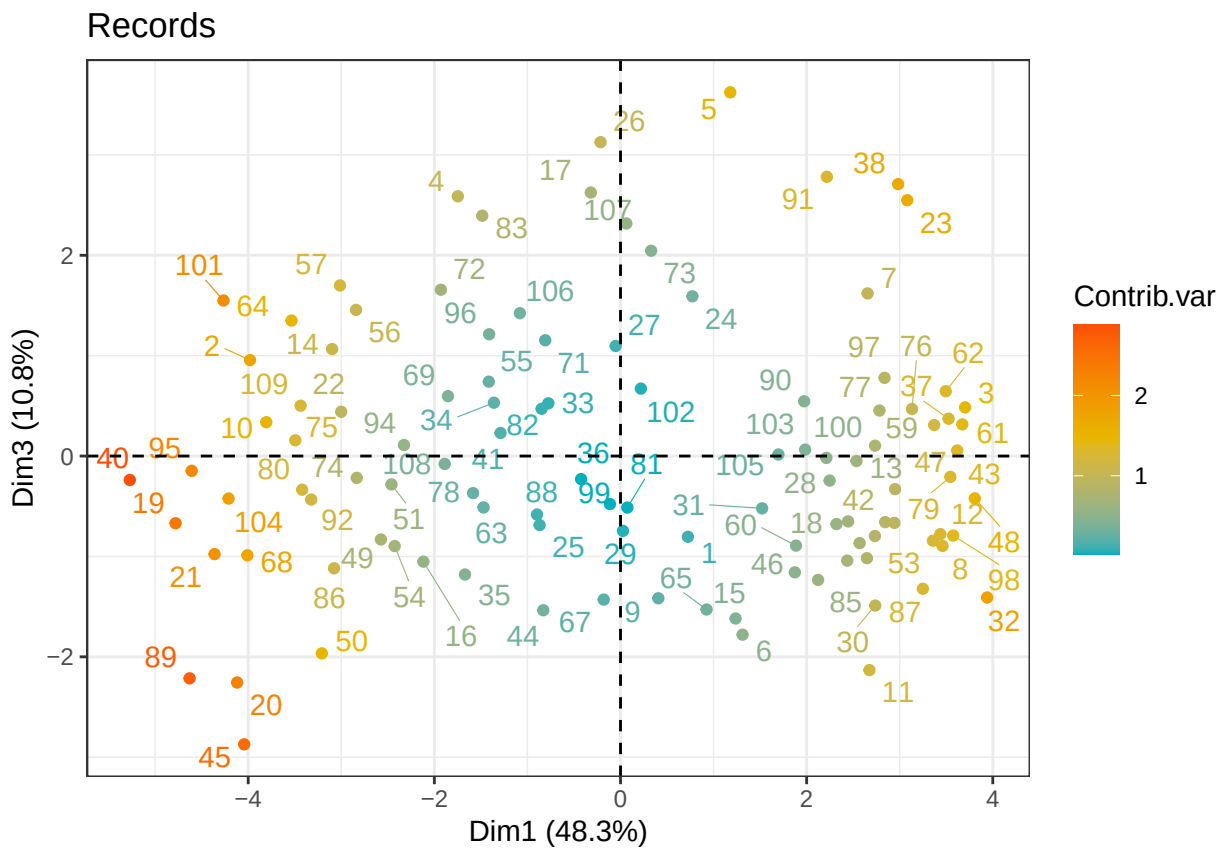


Observations and variance contribution These graphical outputs show the observations with their variance contribution.

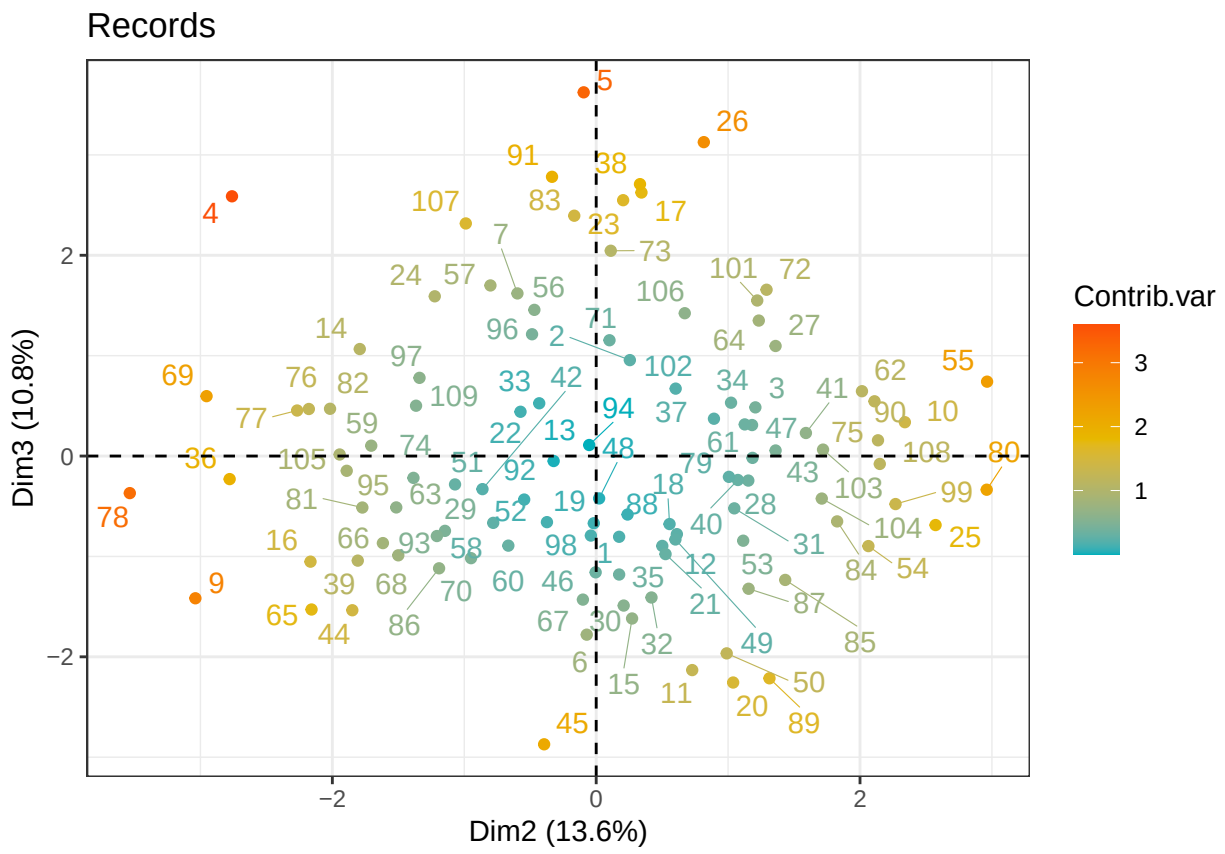
```
# It is also possible to represent the observations
# As well as identify with colors those observations that explain the greatest
# variance of the principal components
p1<-fviz_pca_ind(PCA,col.ind = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel=TRUE,legend.title="Contrib.var",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




```
p2
```

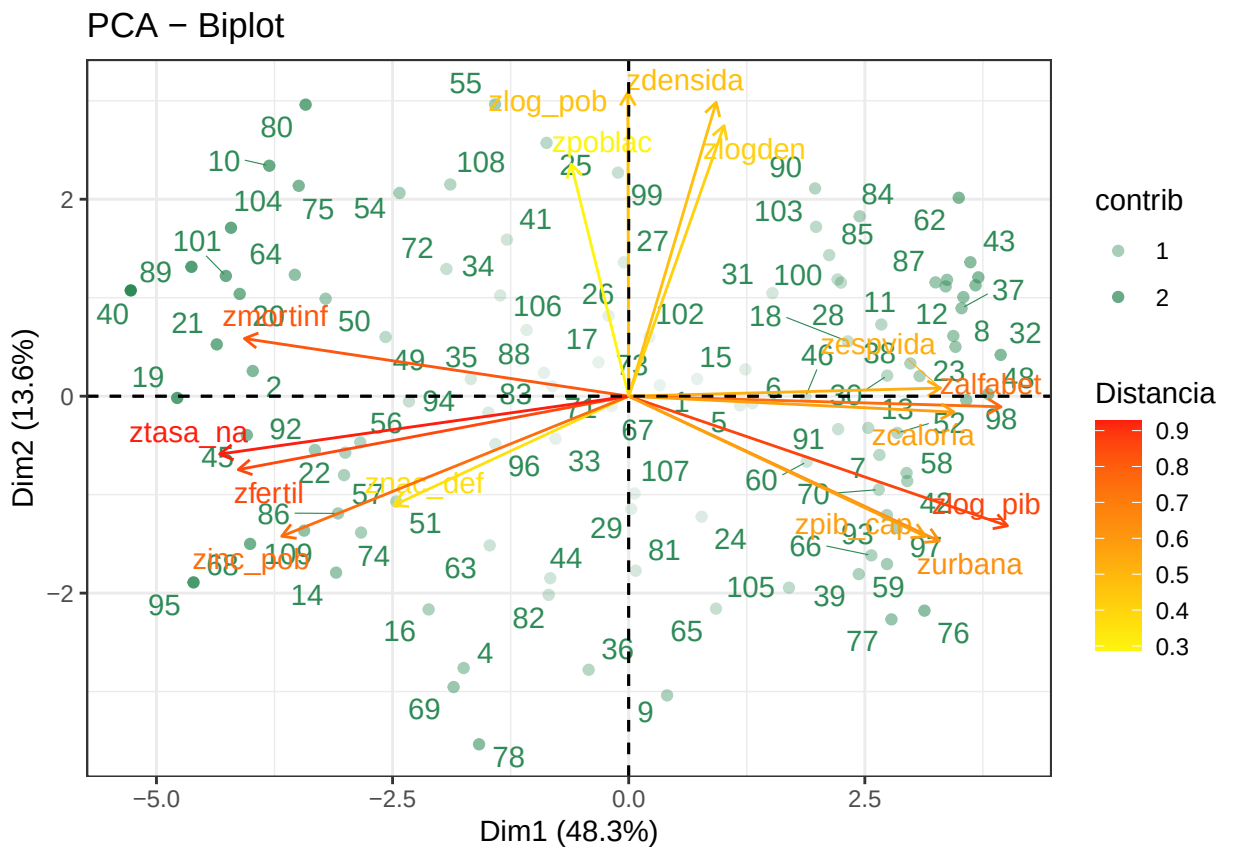



```
p3
```

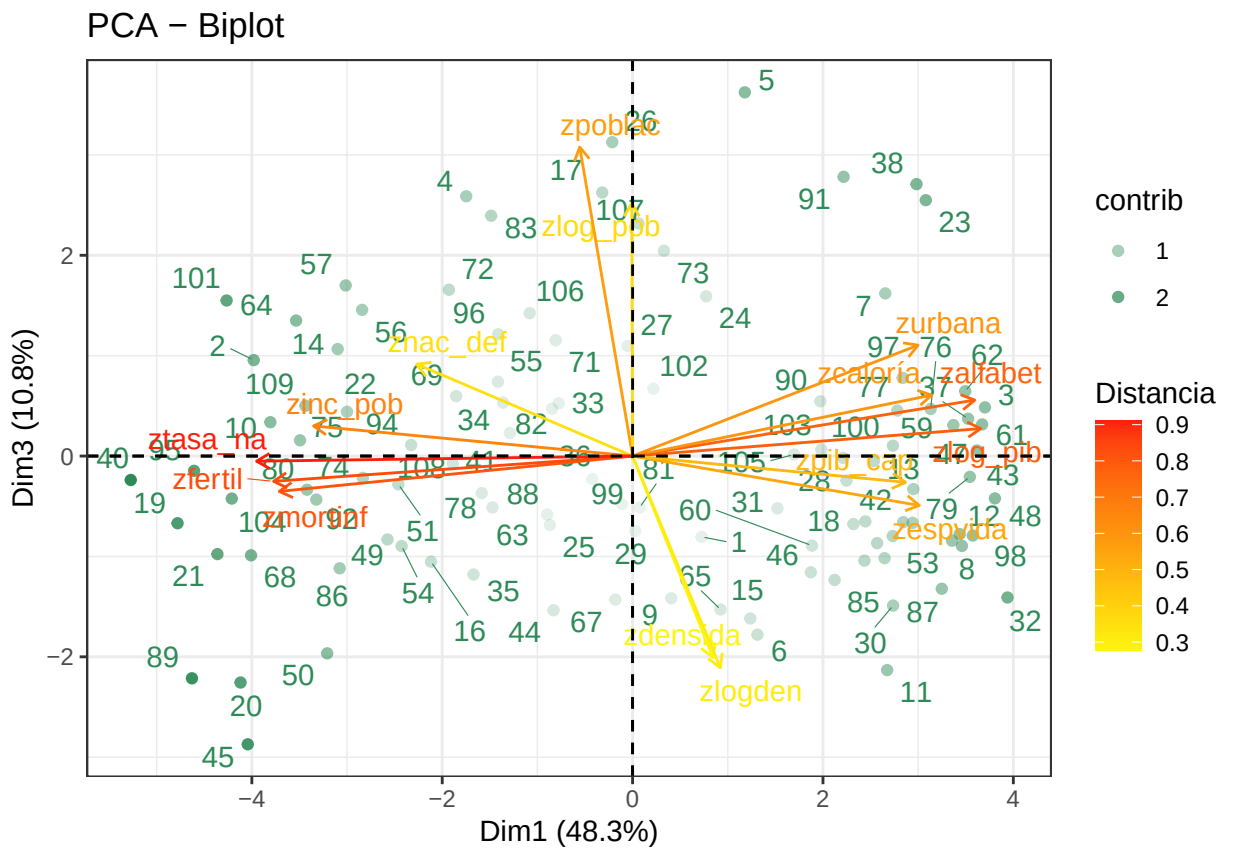


Observations and variables with variance contribution These graphical outputs show observations and the variables with their variance contribution.

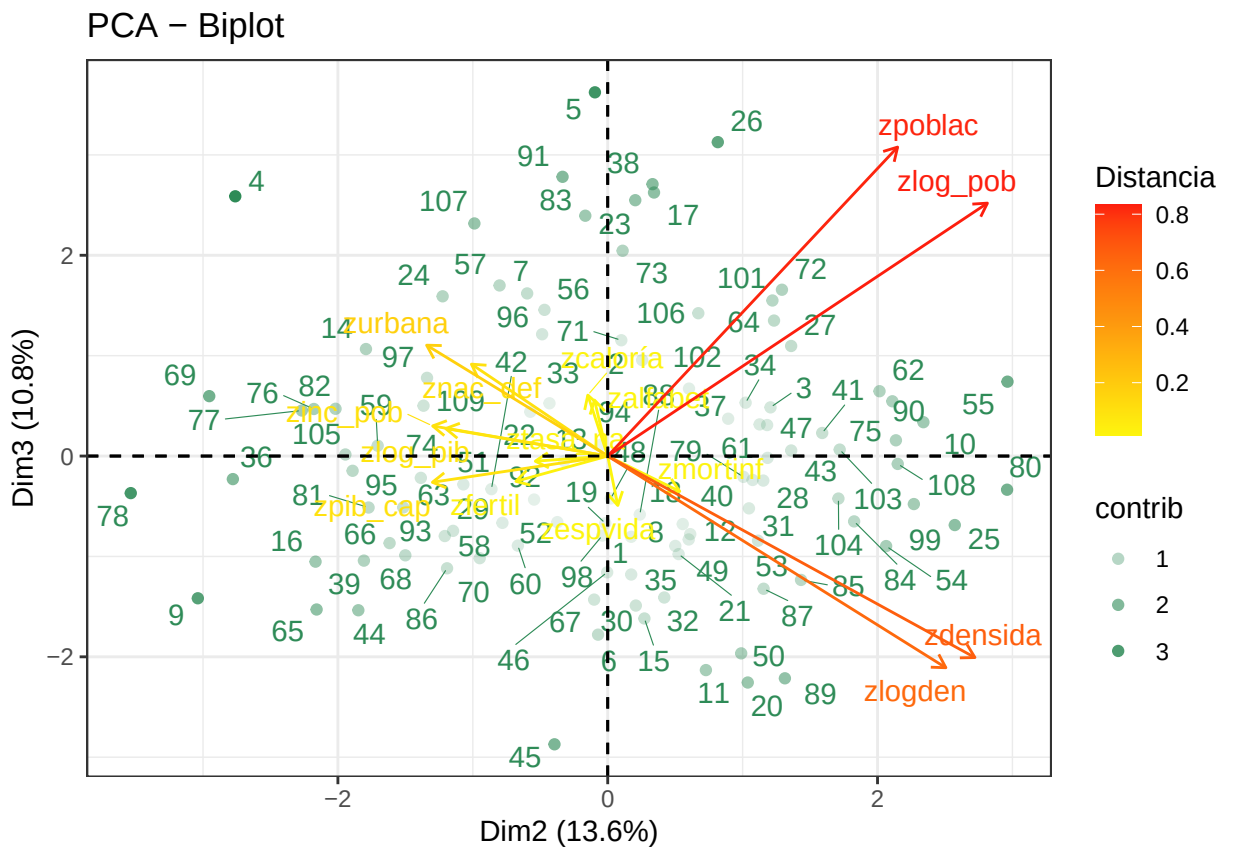
```
# Joint representation of variables and observations
# Relates the possible relationships between the contributions of the records
# to the variances of the components and the weight of the variables in each
# principal component

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
```


PCA − Biplot
zdensida
zlog_pob
zpoblac
zlogden
zmortinf
ztasa_na
zfertil
znac_def
zalfabet
zcalofla
zespvida
zlog_pib
zpib_cap
zurbana
contrib
1
2
Distancia
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Dim2 (13.6%)
Dim1 (48.3%)
−5.0
−2.5
0.0
2.5
2
0
−2

```
p2
```


PCA − Biplot
Dim3 (10.8%)
Dim1 (48.3%)
contrib
1
2
Distancia
0.9
0.8
0.7
0.6
0.5
0.4
0.3
zpobpc
zlog_pob
znac_def
zurbana
zescolaridad
zalfabet
zinc_pob
ztasa_na
zfertil
zmorinf
zdensidad
zlogden
zpib
zagr
zespervida

```
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,]  0.7243719  0.17384337 -0.8043097  0.53765412  0.5743451 -0.08773443
## [2,] -3.9802641  0.25536881  0.9558049 -1.20408733 -1.3162738  0.17462620
## [3,]  3.7020613  1.20674102  0.4850883 -0.03300283 -0.6822311  0.28003937
## [4,] -1.7475142 -2.76086772  2.5876897  1.57824408 -1.2164138 -1.18873591
## [5,]  1.1791692 -0.09470827  3.6232733 -0.75216178 -0.1755295 -0.06676258
## [6,]  1.3116565 -0.07152538 -1.7792326  1.66347951  0.2375102 -0.40621317
##             PC7        PC8          PC9        PC10        PC11        PC12
## [1,] -0.4371808  0.2533790 -0.176991056  0.1293083  0.14162067  0.50298068
## [2,] -0.6283591  0.2089123  0.945020902  0.4344030  0.45804011 -1.21305781
## [3,]  0.6722553 -1.0720675  0.003271464  0.1978932  0.03307633 -0.02222497
## [4,]  0.3432371  0.5850235 -0.540208374 -0.2687585 -0.47528130  0.02492051
## [5,] -0.1513129  0.7839306  0.897802841 -0.8323610  0.05902138  0.20274012
## [6,] -0.4036208  0.5949921 -0.515990927 -0.5048333  0.19904873  0.55010132
##             PC13          PC14         PC15
## [1,]  0.04435935 -0.0046767339  0.118644799
## [2,] -0.86148820  0.6660416716  0.282590448
## [3,] -0.03668565  0.0004020828 -0.007550196
## [4,]  0.03283394  0.5513469693 -0.292062936
## [5,] -0.18107552 -0.0827253048 -0.112049315
```

```
## [6,]  0.04110586  0.3255252656  0.099928444
```