PCA of mixed data with R

- The **gironde** dataset.
- Three functions to perform PCA of mixed data
- Comparison of the three functions
  - Principal Components
  - Eigenvalues
  - Variables
  - Plots of the observations
  - Correlation circle
  - Plot of the levels
- Comparison with MCA
  - Eigenvalues and principal components
  - Levels coordinates

The **gironde** dataset.

The dataset **gironde** is available in the R package **PCAmixdata**. This dataset is a list of 4 datatables and **housing** is one of them.

```
library(PCAmixdata)
data(gironde)
housing <- gironde$housing
head(housing)
```

```
##                      density primaryres   houses owners council
## ABZAC                    132         89  inf 90%     64  sup 5%
## AILLAS                    21         88  sup 90%     77  inf 5%
## AMBARES-ET-LAGRAVE       532         95  inf 90%     66  sup 5%
## AMBES                    101         94  sup 90%     67  sup 5%
## ANDERNOS-LES-BAINS       552         62  inf 90%     72  inf 5%
## ANGLADE                   64         81  sup 90%     81  inf 5%
```

This dataset has:

- $p = 5$ variables with $p_1 = 3$ numerical variables (`density` , `primaryres` , `owners`) and $p_2 = 2$ categorical variables (`houses` and `council`),
- $m = 4$ levels (`inf 90%` and `sup 90%` for the variable `houses` and `inf 5%` and `sup 90%` for the variable `council`),
- $n = 542$ observations.

Three functions to perform PCA of mixed data

Principal Component Analysis of mixed data is available in the following three functions :

- `PCAmix` of the R package **PCAmixdata**
- `FAMD` of the R package **FactoMineR**
- `dudi.mix` of the R package **ade4**

```r
library(PCAmixdata)
library(FactoMineR)
library(ade4)
```

The functions `PCAmix`, `FADM` and `dudi.mix` are used to perform PCA of the mixed dataset `housing`.

```r
# PCAmix (PCAmixdata)
split <- splitmix(housing)
pcamix <- PCAmix(X.quanti=split$X.quanti,
                 X.quali=split$X.quali,
                 rename.level=TRUE,
                 graph=FALSE, ndim=2)
# FAMD (FactoMineR)
famd <- FAMD(housing,
             graph = FALSE, ncp = 2)

# dudi.mix (ade4)
dudimix <- dudi.mix(housing,
                 scannf = FALSE, nf = 2)
```

Comparison of the three functions

Principal Components

Principal components are the coordinates of the projection of the n observations (also called individuals) on the factor maps.

All the three functions give the same principal component scores.

```r
head(pcamix$scores)
```

```
##                      dim 1  dim 2
## ABZAC                 2.36  0.024
## AILLAS               -0.88  0.123
## AMBARES-ET-LAGRAVE    2.62  0.800
## AMBES                 0.93  0.919
## ANDERNOS-LES-BAINS    1.18 -2.481
## ANGLADE              -1.01 -0.424
```

```r
head(famd$ind$coord)
```

```
##                      Dim.1  Dim.2
## ABZAC                 2.36  0.024
## AILLAS               -0.88  0.123
## AMBARES-ET-LAGRAVE    2.62  0.800
## AMBES                 0.93  0.919
## ANDERNOS-LES-BAINS    1.18 -2.481
## ANGLADE              -1.01 -0.424
```

```
head(dudimix$li)
```

```
##   Axis1  Axis2
## 1 -2.36  0.024
## 2  0.88  0.123
## 3 -2.62  0.800
## 4 -0.93  0.919
## 5 -1.18 -2.481
## 6  1.01 -0.424
```

Eigenvalues

The eigenvalues are the variances of the principal components. Because principal components are identical, all three functions give then same eigenvalues.

```
pcamix$eig[1:2,1]
```

```
## dim 1 dim 2
##   2.5   1.1
```

```
famd$eig[,1]
```

```
## comp 1 comp 2
##    2.5    1.1
```

```
dudimix$eig[1:2]
```

```
## [1] 2.5 1.1
```

Moreover, the **total inertia** is by definition equal to $p_1 + m - p_2 = 3 + 4 - 2 = 5$ and this total inertia is the sum of all the eigenvalues.

```
sum(dudimix$eig)
```

```
## [1] 5
```

Variables

Squared loadings

Squared loadings are :

- squared correlations with the principal components when the variables are numerical (`density` , `primaryres` , `owners`),
- correlation ratios with the principal components when the variables are categorical (`houses` and `council`).

Because principal components are identical, all the three functions give the same squared loadings.

```
pcamix$sqload
```

```
##              dim 1 dim 2
## density    0.49550 0.061
## primaryres 0.00035 0.946
## owners     0.73651 0.017
## houses     0.68226 0.030
## council    0.61226 0.016
```

```
famd$var$coord
```

```
##              Dim.1 Dim.2
## density    0.49550 0.061
## primaryres 0.00035 0.946
## owners     0.73651 0.017
## houses     0.68226 0.030
## council    0.61226 0.016
```

```
dudimix$cr
```

```
##                RS1   RS2
## density    0.49550 0.061
## primaryres 0.00035 0.946
## houses     0.68226 0.030
## owners     0.73651 0.017
## council    0.61226 0.016
```

Levels coordinates

The coordinates of the projections on the levels on the factor maps are obtained with the three functions. The functions `PCAmix` and `dudi.mix` give the same results.

```
pcamix$levels$coord
```

```
##                  dim 1  dim 2
## houses= inf 90%  1.63 -0.339
## houses= sup 90% -0.42  0.087
## council= inf 5% -0.40 -0.065
## council= sup 5%  1.52  0.245
```

```
dudimix$co[-c(1,2,5),]
```

```
##                  Comp1  Comp2
## house..inf.90. -1.63 -0.339
## house..sup.90.  0.42  0.087
## counc..inf.5.   0.40 -0.065
## counc..sup.5.  -1.52  0.245
```

The function `FAMD` gives the same results up to a factor of $\sqrt{\lambda_\alpha}$ in each dimension (where λ_α is the αth eigenvalue).

```
famd$quali.var$coord %*%diag(1/sqrt(pcamix$eig[1:2,1]))
```

```
##           [,1]   [,2]
##  inf 90%  1.63 -0.339
##  sup 90% -0.42  0.087
##  inf 5%  -0.40 -0.065
##  sup 5%   1.52  0.245
```

In other words, the level coordinates obtained with the functions `PCAmix` and `dudi.mix` verify the so-called **quasi_barycentric** property. This property says that a level is represented at the barycenter of the observations that have this level, up to a factor of $\frac{1}{\sqrt{\lambda_\alpha}}$ in each dimension.

```
barycenter <- apply(pcamix$scores[which(housing$houses==" inf 90%"),],2,mean)
quasi_barycenter <- barycenter/sqrt(pcamix$eig[1:2,1])
# PCAmix coordinates of the level 'inf 90%'
pcamix$levels$coord[1,, drop=FALSE]
```

```
##                  dim 1 dim 2
## houses= inf 90%   1.6 -0.34
```

```
quasi_barycenter
```

```
## dim 1 dim 2
##  1.63 -0.34
```

The level coordinates of the `FAMD` on their part verify the **barycentric** property.

```
barycenter <- apply(famd$ind$coord[which(housing$houses==" inf 90%"),],2,mean)
# FAMD coordinates of the level 'inf 90%'
famd$quali.var$coord[1,, drop=FALSE]
```

```
##          Dim.1 Dim.2
##  inf 90%   2.6 -0.35
```

```
barycenter
```

```
## Dim.1 Dim.2
##  2.59 -0.35
```

Numerical variables coordinates (correlations)

The coordinates of the projections of the numerical variables interprets as correlations with the principal components. All the three functions give the same results.

```
pcamix$quanti$coord
```

```
##              dim 1 dim 2
## density     0.704  0.25
## primaryres -0.019  0.97
## owners     -0.858  0.13
```

```
famd$quanti.var$coord
```

```
##              Dim.1 Dim.2
## density     0.704  0.25
## primaryres -0.019  0.97
## owners     -0.858  0.13
```

```
dudimix$co[c(1,2,5),]
```

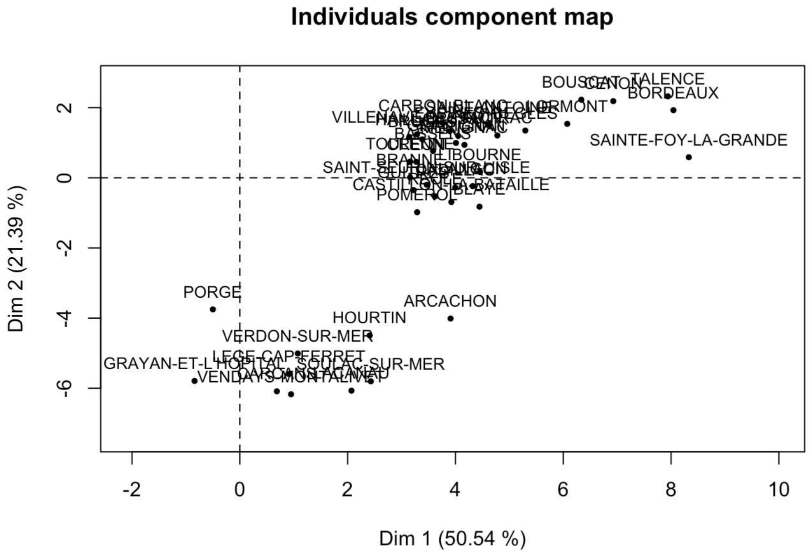
```
##              Comp1 Comp2
## density    -0.704  0.25
## primaryres  0.019  0.97
## owners      0.858  0.13
```

Plots of the observations

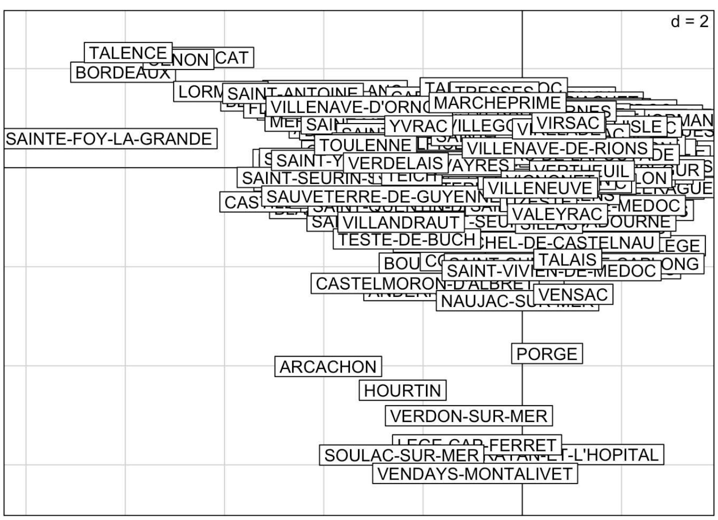
```
n <- nrow(housing)
100/n # mean contribution
```

```
## [1] 0.18
```

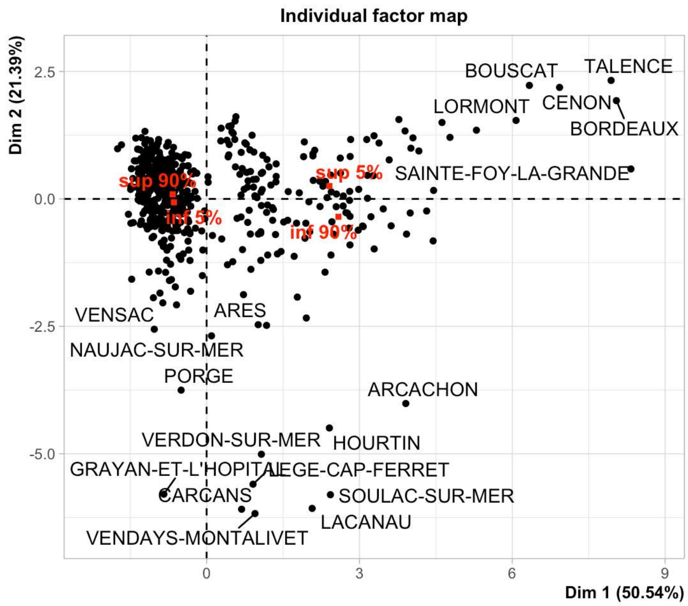
```
plot(pcamix,choice="ind", lim.contrib.plot = 0.5, cex=0.8)
```




```
s.label(dudimix$li, label = rownames(gironde$housing))
```



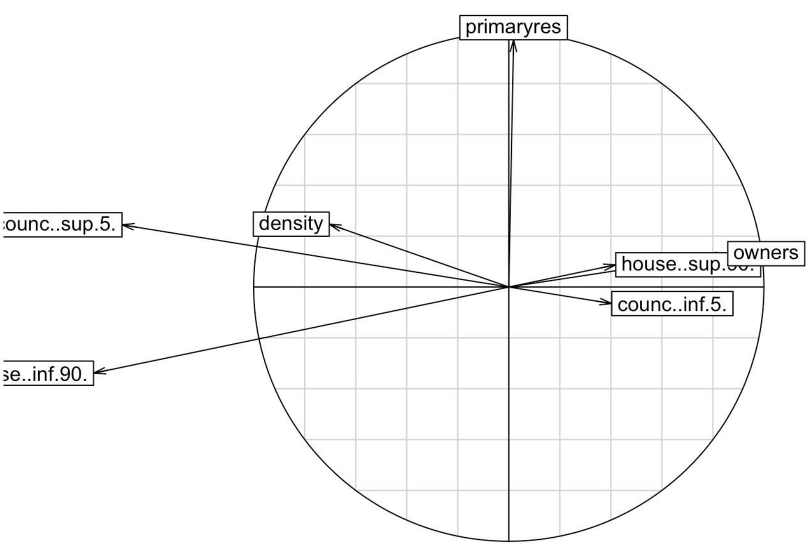

```
plot(famd, choix="ind")
```



Correlation circle

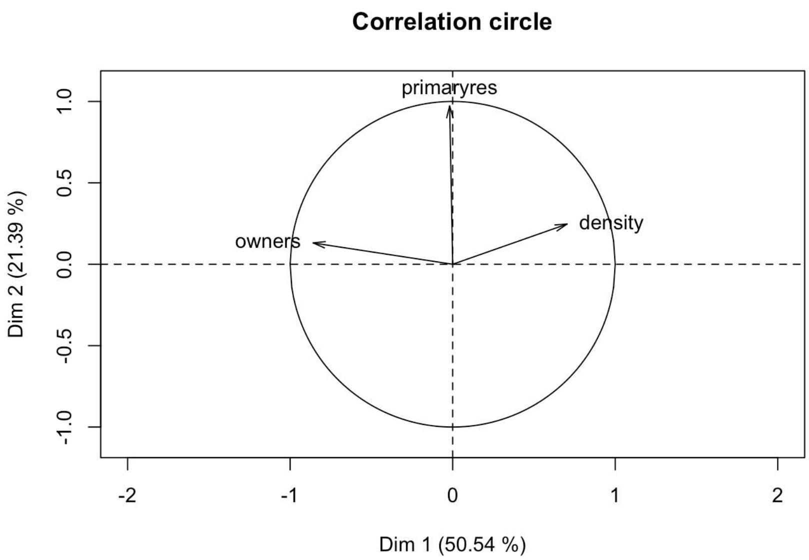
With `ade4` the representation of the numerical variables and the representation of the levels are necesseraly on the same plot.

```
s.corcircle(dudimix$co)
```

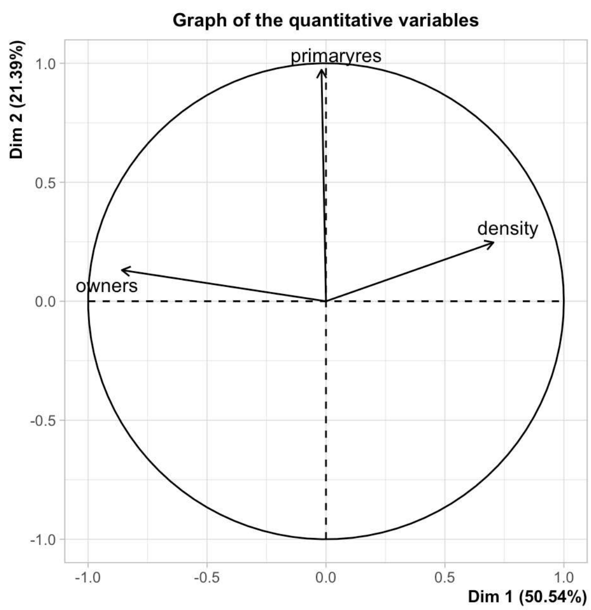


With `PCAmixdata` and `FactoMineR` the correlation circle is obtained separately.

```
plot(pcamix, choice = "cor")
```



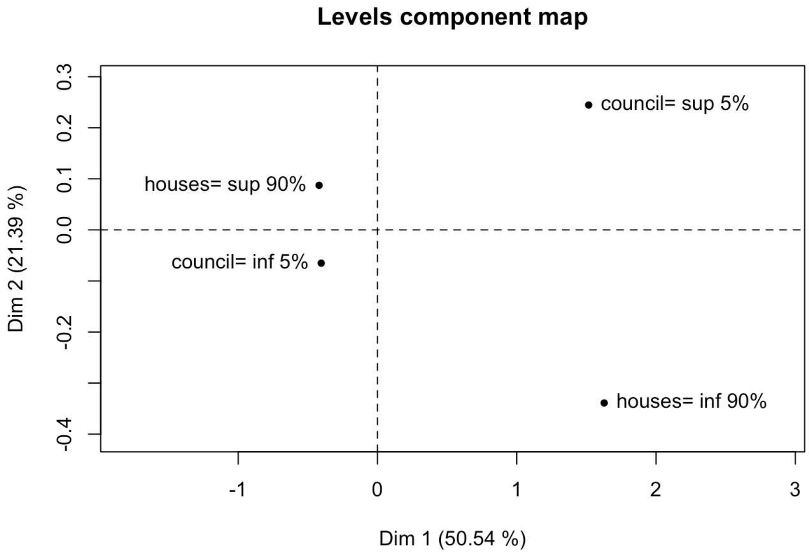

```
plot(famd, choix = "quanti")
```



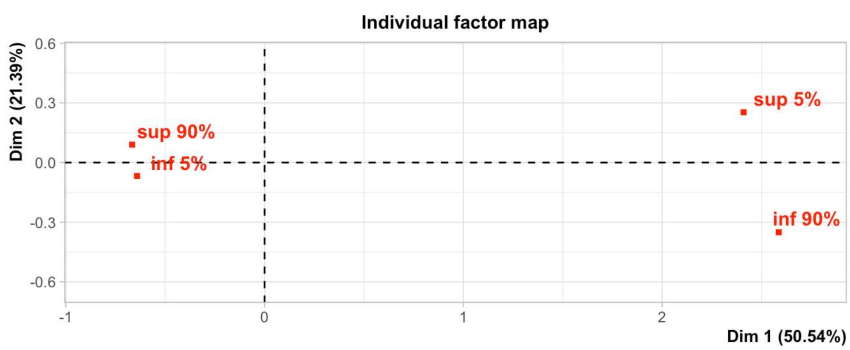
Plot of the levels

We have seen that with **ade4** the levels are ploted on the “correlation circle”. With the two other packages a specific plot can be drawn.

```
plot(pcamix, choice = "levels", xlim=c(-1.5,2.4))
```




```
plot(famd, choix = "ind", invisible = "ind")
```



Comparison with MCA

The datatable **services** is a dataset with $p = 9$ categorical variables and the same $n = 542$ observations (cities).

```
data(gironde)
services <- gironde$services
head(services)
```

```
##                      butcher  baker postoffice dentist grocery nursery doctor
## ABZAC                      0 2 or +     1 or +       0       0       0      0
## AILLAS                     0      0          0       0  1 or +       0 3 or +
## AMBARES-ET-LAGRAVE         1 2 or +     1 or +  3 or +  1 or +  1 or + 3 or +
## AMBES                      0      1     1 or +  1 to 2  1 or +       0 3 or +
## ANDERNOS-LES-BAINS  2 or + 2 or +     1 or +  3 or +  1 or +       0 3 or +
## ANGLADE                    0      1          0       0  1 or +       0      0
##                      chemist restaurant
## ABZAC                      1          1
## AILLAS                     0          1
## AMBARES-ET-LAGRAVE  2 or +     3 or +
## AMBES                      1     3 or +
## ANDERNOS-LES-BAINS  2 or +     3 or +
## ANGLADE                    0          2
```

When the data are categorical, the three functions `PCAmix` , `FADM` and `dudi.mix` perform simple multiple correspondance analysis (MCA).

```
# MCA with PCAmix (PCAmixdata)
mca.pcamix <- PCAmix(X.quali=services,
                 rename.level=TRUE,
                 graph=FALSE, ndim=2)
# MCA with FAMD (FactoMineR)
mca.famd <- FAMD(services,
             graph = FALSE, ncp = 2)

# MCA with dudi.mix (ade4)
mca.dudimix <- dudi.mix(services,
                 scannf = FALSE, nf = 2)
```

It is also possible to use the functions- `dudi.acm` of the package **ade4** and the function `MCA` of the package **FactoMineR**.

```
# function MCA (FactoMineR)
mca <- MCA(services,
             graph = FALSE, ncp = 2)

# function dudi.acm (ade4)
mca.dudi <- dudi.acm(services,
                 scannf = FALSE, nf = 2)
```

Eigenvalues and principal components

The principal component scores obtained with `PCAmix` , `FADM` and `dudi.mix` are identical (as stated above).

However, they are slightly different when the functions `MCA` and `dudi.acm` are used.

```
mca.pcamix$eig[1:2,1] # PCAmix, dudi.mix, FADM
```

```
## dim 1 dim 2
##   5.8   2.6
```

```
mca.dudi$eig[1:2] # MCA with ade4
```

```
## [1] 0.64 0.29
```

```
mca$eig[1:2,1] # MCA with FactoMineR
```

```
## dim 1 dim 2
##  0.64  0.29
```

The principal component of the functions `MCA` and `dudi.acm` must be multiplied by $\sqrt{p}$ where p is the number of categorical variables. In other words, the eigenvalues should be multiplied by p to get identical results.

```
p <- ncol(services)
mca$eig[1:2,1]*p
```

```
## dim 1 dim 2
##   5.8   2.6
```

Levels coordinates

The levels coordinates obtained with `PCAmix` and `dudi.mix` are identical but differs from that obtained with `FADM` from a factor $\sqrt{\lambda_\alpha}$. As stated above, the levels coordinates obtained with `PCAmix` and `dudi.mix` are quasi-barycenters whereas they are barycenters with `FADM` .

When the functions `MCA` and `dudi.mca` are used, the levels coordinates are identical to those obtained with `PCAmix` and `dudi.mix` .

```
head(mca.famd$quali.var$coord)
```

```
##         Dim.1  Dim.2
## 0       -1.18 -0.043
## 1        1.21  1.101
## 2 or +   4.23 -1.167
## 0       -1.66 -0.511
## 1        0.27  1.733
## 2 or +   3.65 -0.594
```

```
head(mca.pcamix$levels$coord)
```

```
##                 dim 1  dim 2
## butcher=0      -0.49 -0.027
## butcher=1       0.50  0.686
## butcher=2 or +  1.76 -0.727
## baker=0        -0.69 -0.318
## baker=1         0.11  1.080
## baker=2 or +    1.52 -0.370
```

```
head(mca$var$coord)
```

```
##                 Dim 1  Dim 2
## butcher_0      -0.49 -0.027
## butcher_1       0.50  0.686
## butcher_2 or +  1.76 -0.727
## baker_0        -0.69 -0.318
## baker_1         0.11  1.080
## baker_2 or +    1.52 -0.370
```

```
head(mca.dudi$co)
```

```
##                 Comp1  Comp2
## butcher.0       0.49  0.027
## butcher.1      -0.50 -0.686
## butcher.2.or.. -1.76  0.727
## baker.0         0.69  0.318
## baker.1        -0.11 -1.080
## baker.2.or..   -1.52  0.370
```