Reduction de dimension avec l'Analyse en Composante Principal

BADJO Dibé

Exploration de données

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.1     v purrr   1.0.1
v tibble  3.1.8     v dplyr   1.1.0
v tidyr   1.3.0     v stringr 1.5.0
v readr   2.1.4     v forcats 1.0.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(FactoMineR)
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
Rows: 8 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): type
dbl (10): M1, M2, M3, M4, M5, M6, M7, M8, M9, M10

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

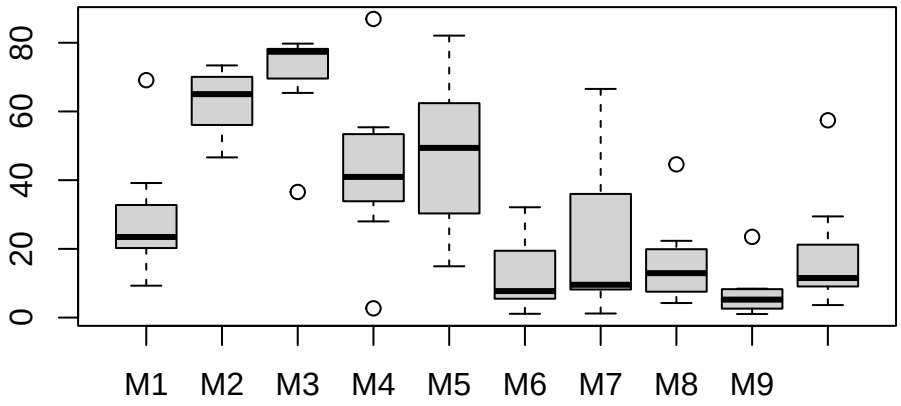
```r
head(data)
```

```
# A tibble: 6 x 11
  type       M1    M2    M3    M4    M5    M6    M7    M8    M9   M10
  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Cancer1 22.9   69.5  73.8  2.69  82.1  1.1   1.19  4.24  5.23  3.64
2 Cancer2 17.6   65.0  79.7 51.4   49.4 22.1   7.31  6.15  1.03  9.23
3 Cancer3 23.5   70.7  78.5 39.7   71.1 16.8   9.03  8.92  2.90 13.0
4 Cancer4  9.29  73.4  77.9 86.9   14.9  6.23 37.6  17.5   8.13  8.88
5 Cancer5 39.2   61.6  65.4 55.4   39.7  7.71 34.4  22.3   8.42 29.4
6 Cancer6 69.1   46.6  36.6 40.9   20.9 32.1  66.6  44.6  23.5  57.4
```

```r
x <- data[1:7,-1]
```

```r
summar <- summary(x)
view(summar)
```

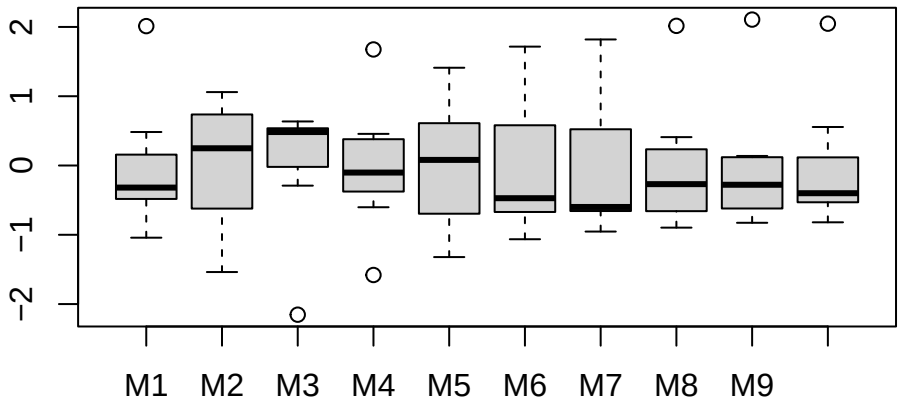
```r
boxplot(x)
```




```r
matCors <- cor(x)
view(matCors)
```

```r
Xsc <- scale(x, scale = T)
```

```r
boxplot(Xsc)
```



Pour parvenir a trouver les composantes principale, il faut tous d'abord calculer la matrix transpose(X)*X

```
sigma <- t(Xsc) %*% Xsc/nrow(Xsc)
view(sigma)
```

Utilisons cette matrix pour calculer ensuite les valeurs propres et les vecteurs propres

```
ACP <- eigen(sigma)
```

```
class(ACP)
```

```
[1] "eigen"
```

Les valeurs propres

```
values <- ACP$values
values
```

```
 [1]  5.899224e+00  1.661563e+00  5.142035e-01  3.981676e-01  8.649222e-02
 [6]  1.177830e-02  3.683852e-16  1.068940e-16 -4.803234e-17 -8.293099e-17
```

Les vecteurs propres

```
vec <- ACP$vectors
```

```r
view(vec)
```

Pour savoir le nombre de vecteur propre a selection il faut tracer le graph suivant, appellé le graph de coude, dans les resultats, le nombre de composante principale a sélectionner est 2

```r
inertie <- cumsum(values)/sum(values)
inertie
```

```
[1] 0.6882428 0.8820918 0.9420822 0.9885351 0.9986259 1.0000000 1.0000000
[8] 1.0000000 1.0000000 1.0000000
```

Une autre manière de le savoir est de voir combien de vecteur propre garde plus de 80% de l'information, dans les resulats, les deux premiers vecteur propres garde plus de 80% de l'information ce qui vient confirmé le nombre qu'on a selectionner qui est 2

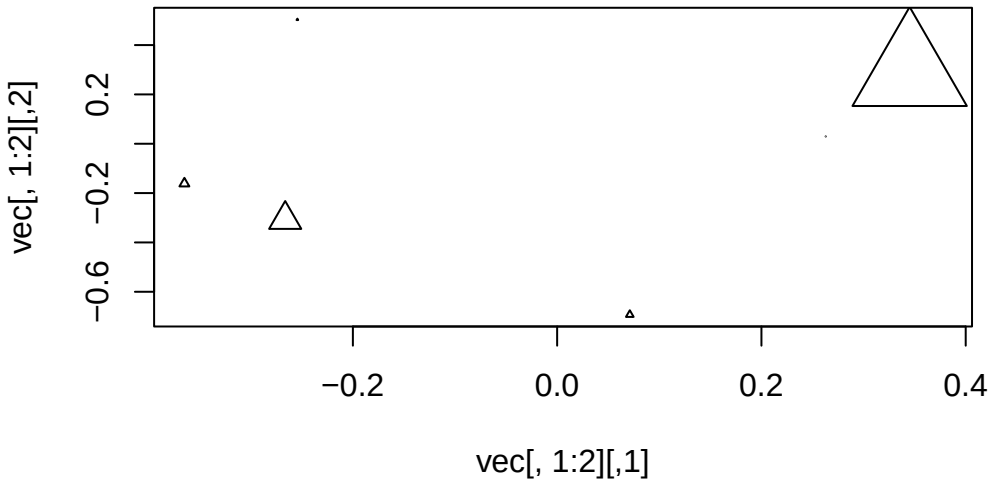
```r
pourcinertie <- inertie*100
pourcinertie
```

```
[1]  68.82428  88.20918  94.20822  98.85351  99.86259 100.00000 100.00000
[8] 100.00000 100.00000 100.00000
```

Plus de 80% de la variance sont expliquer par les 2 premiers composant principales, ce qui nous conduit a garder que ces deux composantes.

Dans le pourcentage expliquer par ces deux composant, le premier composant á lui seul explique approximativement 68% de la variance

```r
plot(vec[, 1:2], pch = 2, cex = values)
text(vec[, 1:2], labels = rownames(vec), pos = 3)
```



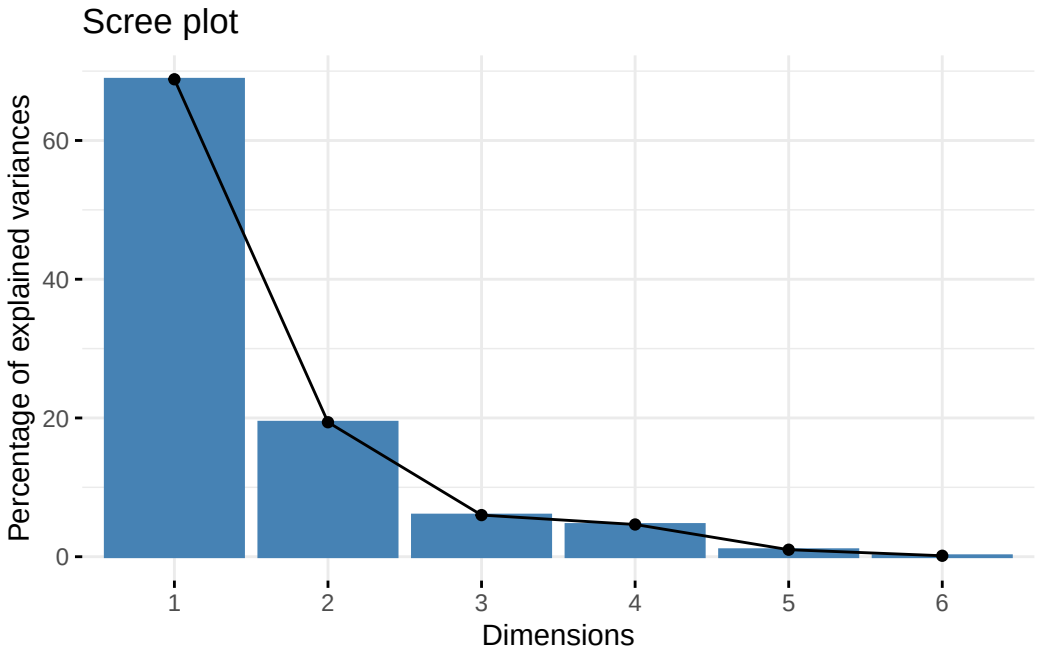

```
res.acp<-PCA(x, graph = FALSE)
```

```
view(res.acp$var$cos2)
```

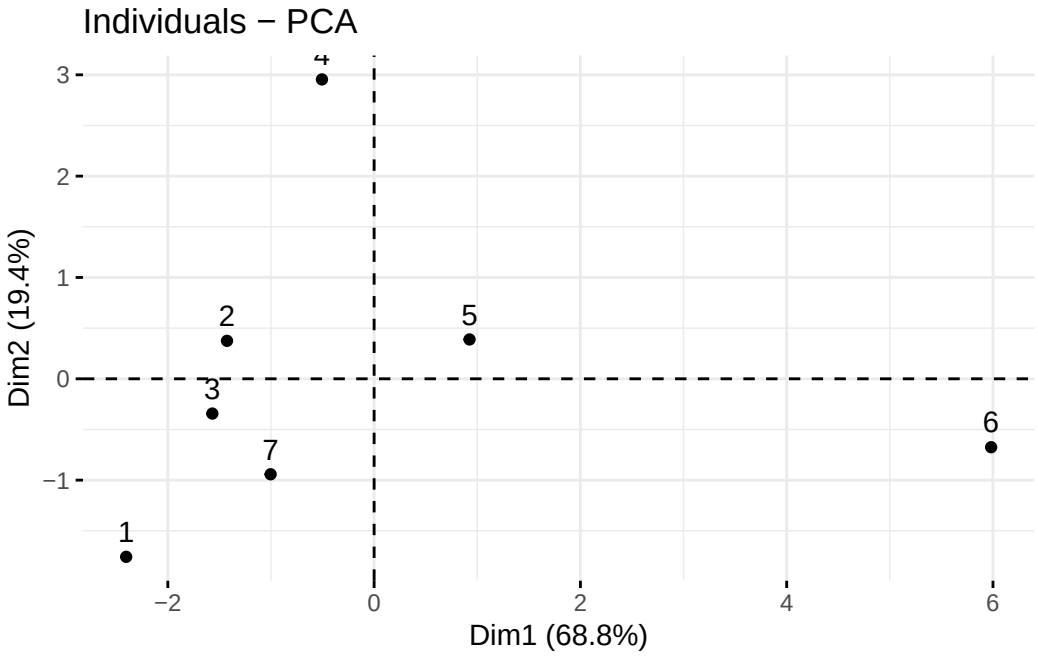
```
get_eigenvalue(res.acp)
```

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1 6.88242795       68.8242795                     68.82428
Dim.2 1.93849019       19.3849019                     88.20918
Dim.3 0.59990407        5.9990407                     94.20822
Dim.4 0.46452885        4.6452885                     98.85351
Dim.5 0.10090758        1.0090758                     99.86259
Dim.6 0.01374135        0.1374135                    100.00000
```

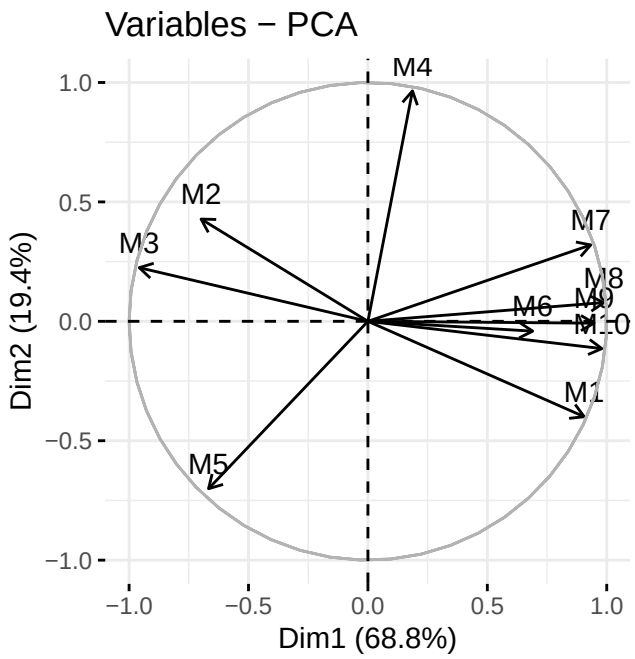
```
fviz_eig(res.acp)
```




```
fviz_pca_ind(res.acp)
```



```
fviz_pca_var(res.acp)
```




```
fviz_pca_biplot(res.acp)
```