Dimensionality Reduction

Anh Thu

10/18/2021

#load the library

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:plyr':
##
##     mutate
```

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.5     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x plyr::arrange()          masks dplyr::arrange()
## x lubridate::as.difftime() masks base::as.difftime()
## x purrr::compact()         masks plyr::compact()
## x plyr::count()            masks dplyr::count()
## x lubridate::date()        masks base::date()
## x plyr::failwith()         masks dplyr::failwith()
## x dplyr::filter()          masks stats::filter()
## x plyr::id()               masks dplyr::id()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x ggpubr::mutate()         masks plyr::mutate(), dplyr::mutate()
## x plyr::rename()           masks dplyr::rename()
## x lubridate::setdiff()     masks base::setdiff()
## x plyr::summarise()        masks dplyr::summarise()
## x plyr::summarize()        masks dplyr::summarize()
## x lubridate::union()       masks base::union()
```

```
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
```

```
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
```

```
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(factoextra)
```

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

```
library(xtable)
library(FactoMineR)
library(cluster.datasets)
```

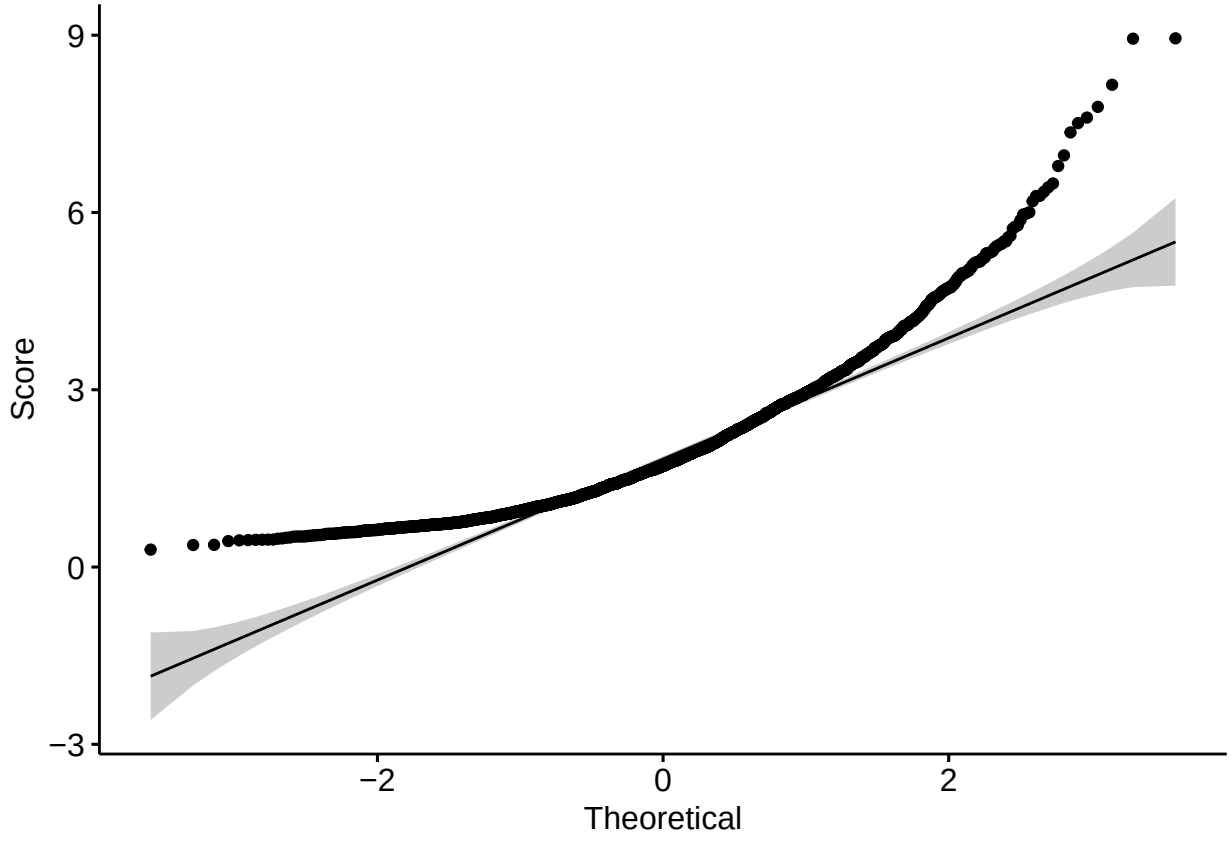
#Import the data set

```
df <- read.csv("users.db.csv")
colnames(df)
```

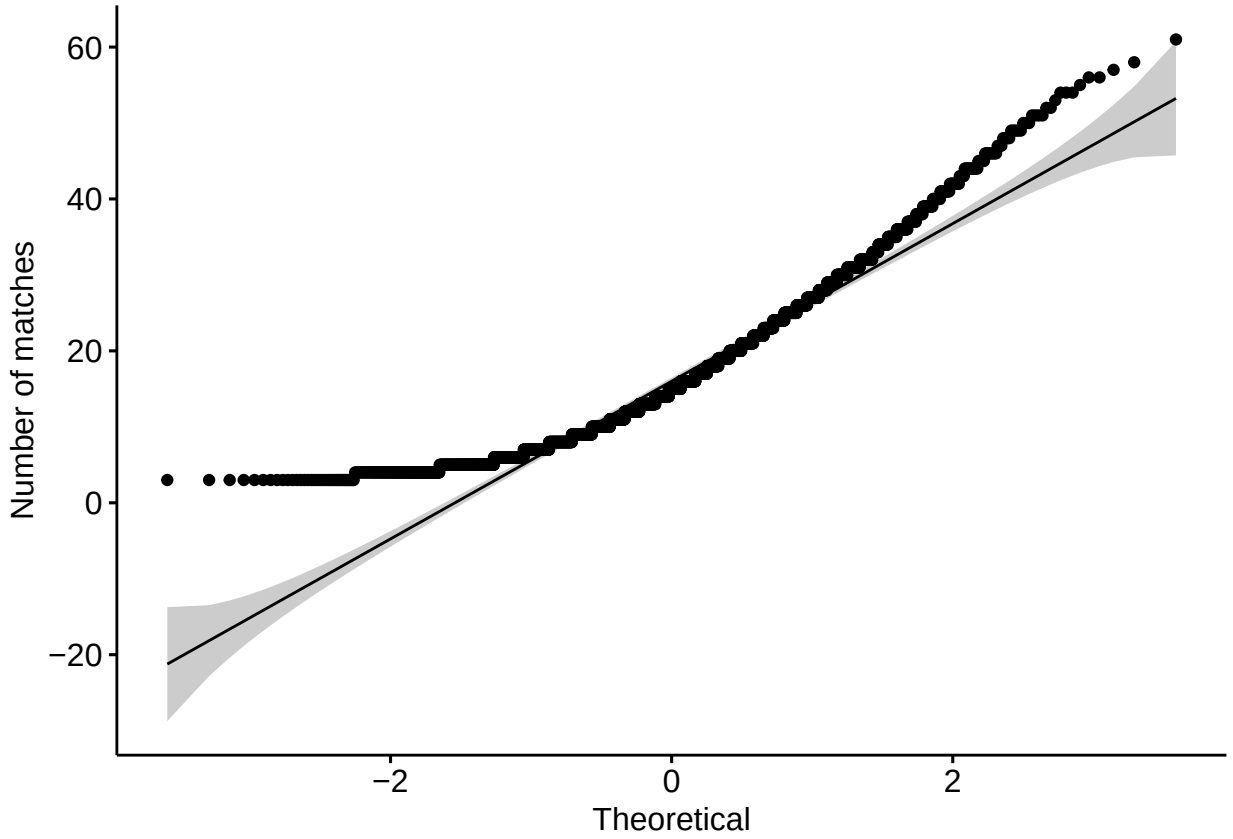
```
##  [1] "userid"          "date.crea"       "score"           "n.matches"
##  [5] "n.updates.photo" "n.photos"        "last.connex"     "last.up.photo"
##  [9] "last.pr.update"  "gender"          "sent.ana"        "length.prof"
## [13] "voyage"          "laugh"           "photo.keke"      "photo.beach"
```

#Identifying correlations in the variables

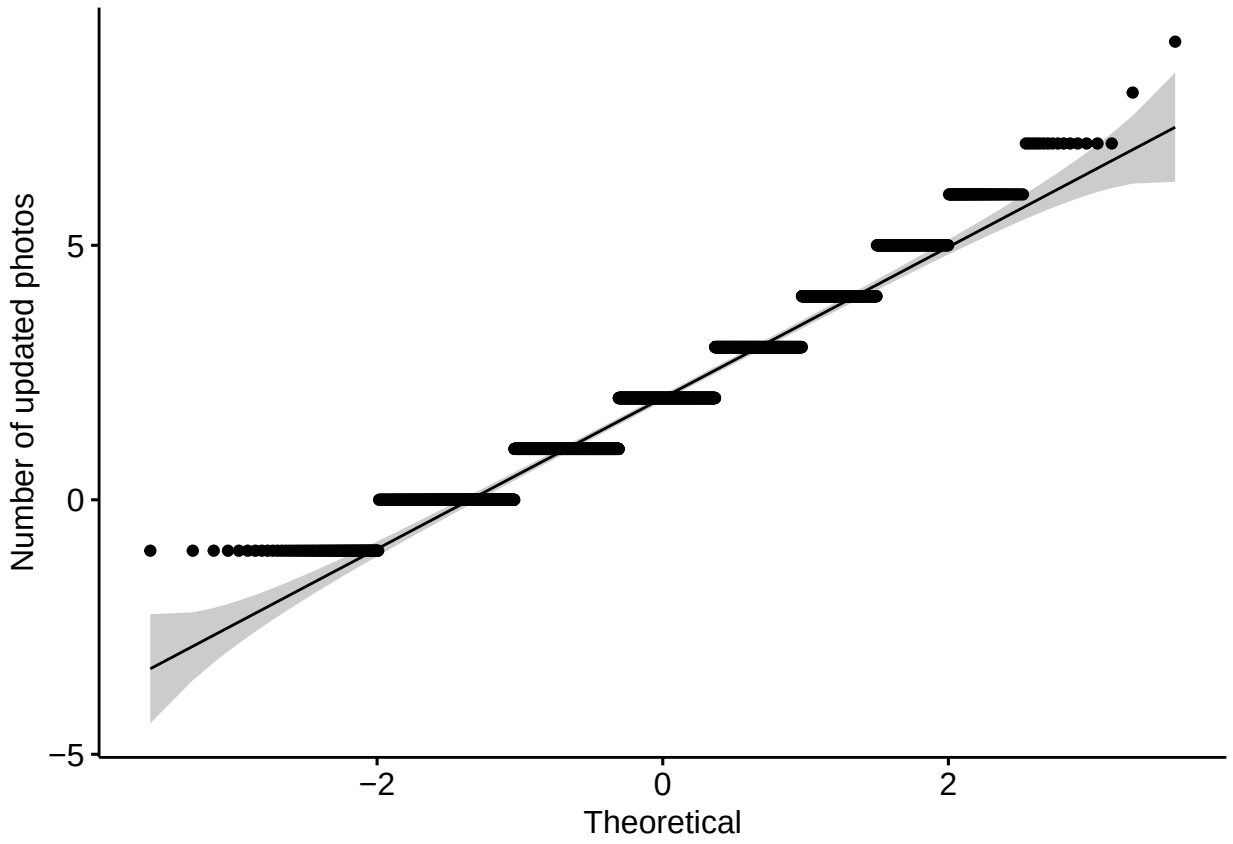
```
#Visual inspection of the data normality using Q-Q plots
ggqqplot(df$score, ylab = "Score")
```



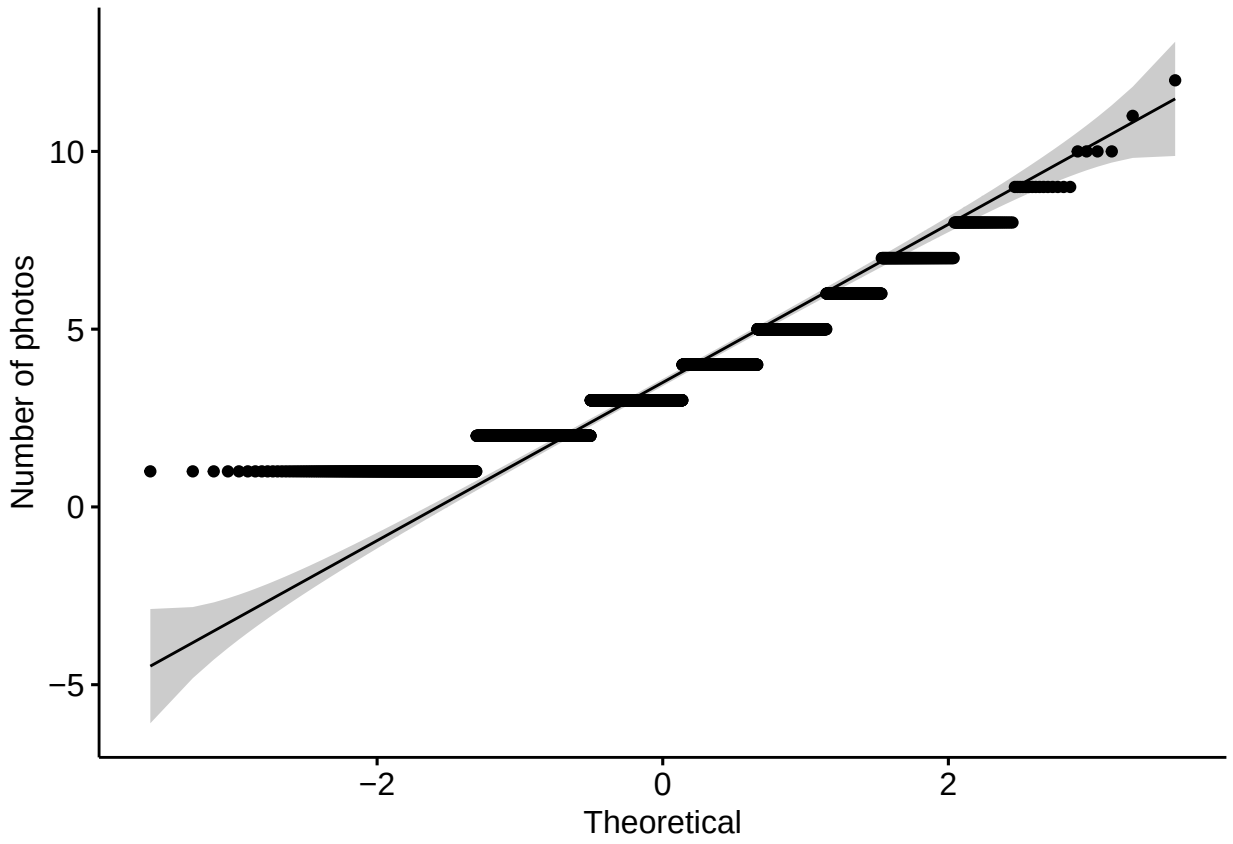
```
ggqqplot(df$n.matches, ylab = "Number of matches")
```




```
ggqqplot(df$n.updates.photo, ylab = "Number of updated photos")
```




```
ggqqplot(df$n.photos, ylab = "Number of photos")
```



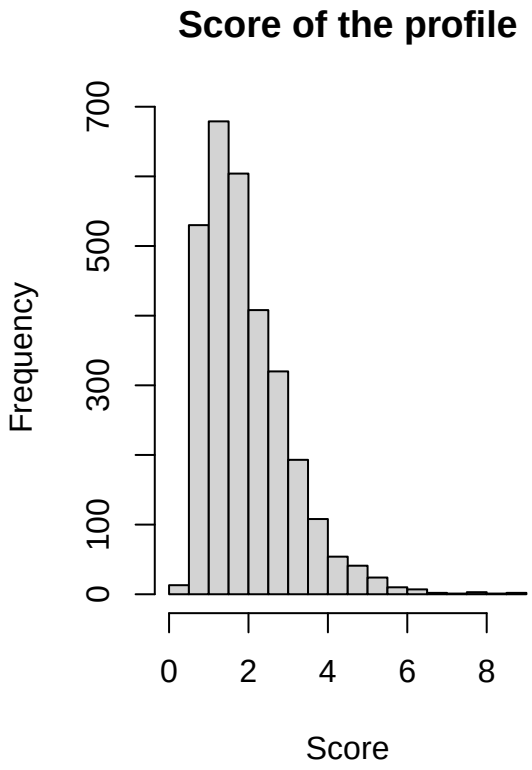
##Cor.test score & n.matches

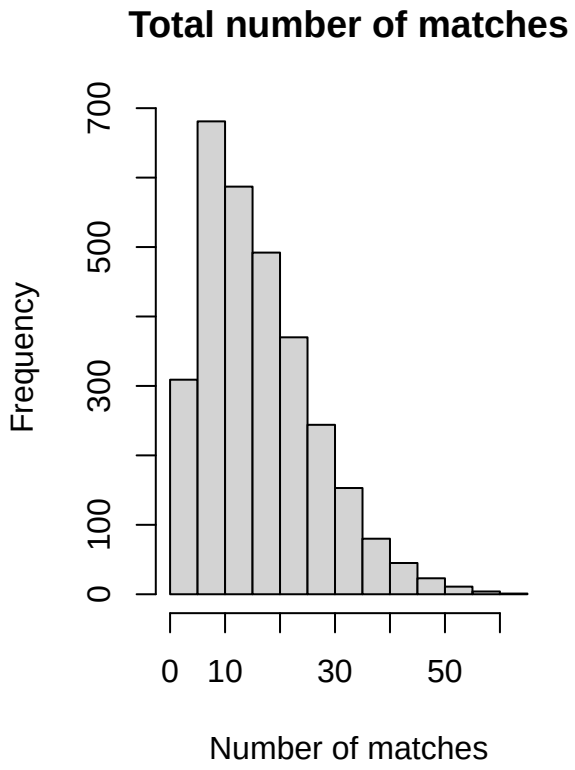
```
cor.test(df$score, df$n.matches, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$score and df$n.matches
## t = 114.44, df = 2998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8951737 0.9085205
## sample estimates:
##       cor
## 0.9020625
```

##Histograms of Score and Matches variable

```
par(mfrow=c(1,2))
hist(df$score,main='Score of the profile',xlab='Score')
hist(df$n.matches,main='Total number of matches',xlab='Number of matches')
```

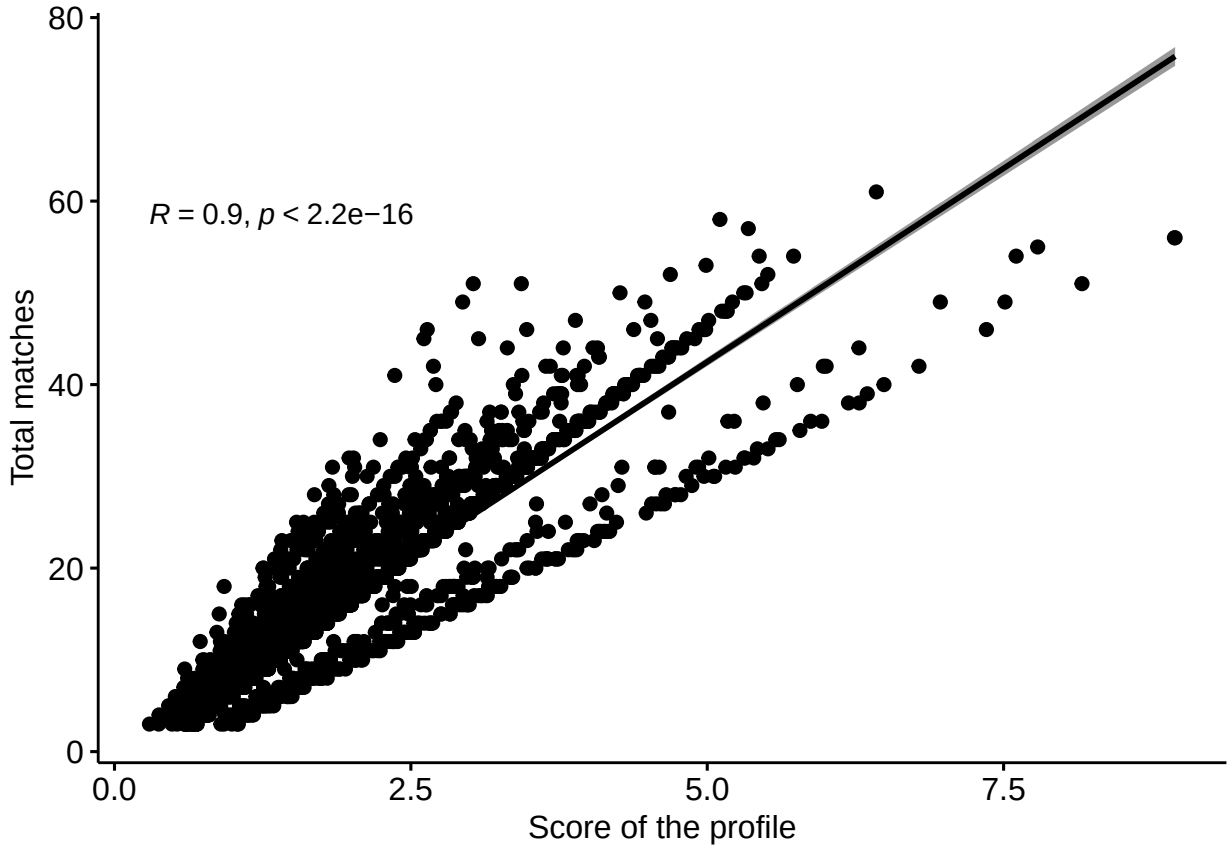




##Scatter plot for score & n.matches

```
ggscatter(df, x = "score", y = "n.matches",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Score of the profile", ylab = "Total matches")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Cor.test gender & n.photos

```
cor_score_keke <- cor.test(df$gender, df$n.photos, method = "pearson")
cor_score_keke
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$gender and df$n.photos
## t = 15.494, df = 2998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2388294 0.3051022
## sample estimates:
##       cor
## 0.2722887
```

Cor test gender & photo.keke

```
cor.test(df$gender, df$photo.keke, method ='spearman')
```

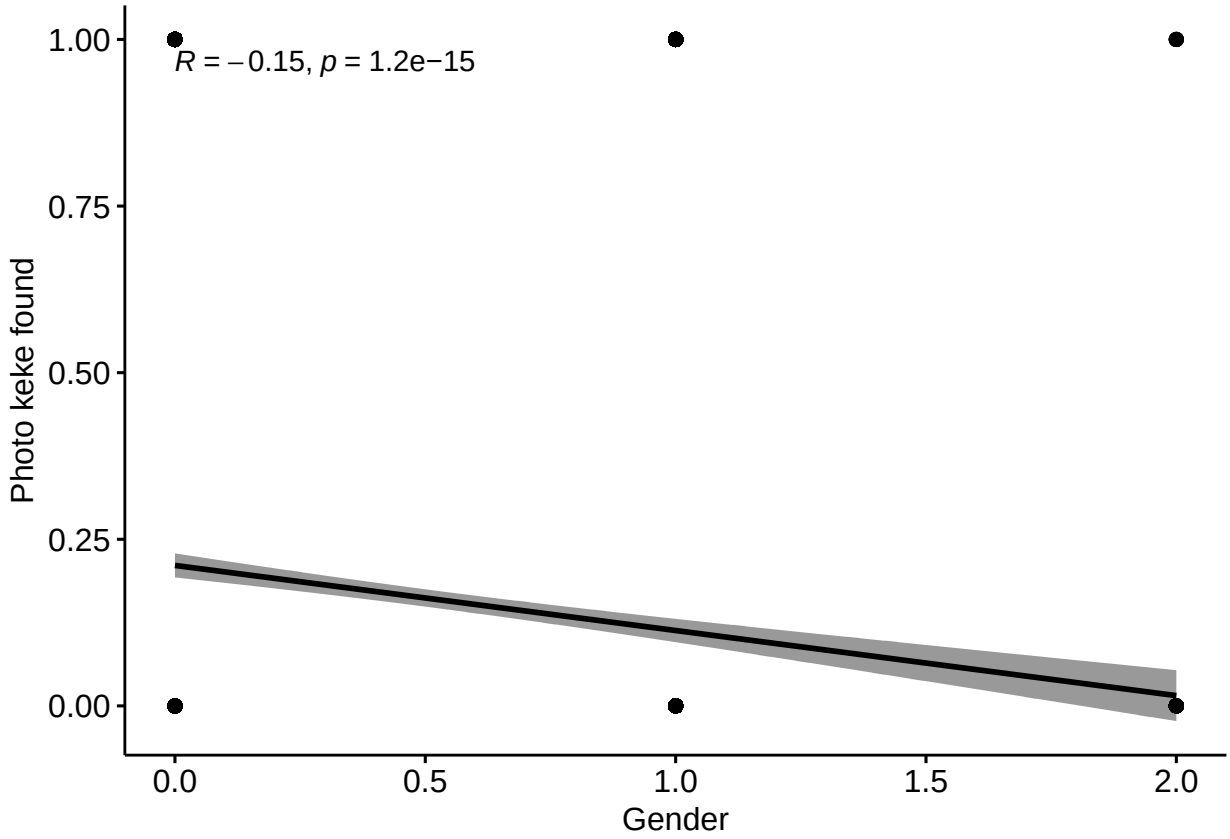
```
## Warning in cor.test.default(df$gender, df$photo.keke, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df$gender and df$photo.keke
## S = 5154699806, p-value = 1.165e-15
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.145489
```

Scatter plot for gender & keke

```
ggscatter(df, x = "gender", y = "photo.keke",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Gender", ylab = "Photo keke found")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



##The correlations between several variables

```
t1 <- round(cor(cbind(df$score,df$n.matches,df$n.updates.photo,df$n.photos)),2)
xtable(t1)
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
```

```
## % Fri Nov 19 10:36:06 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 1.00 & 0.90 & 0.29 & 0.05 \\
##   2 & 0.90 & 1.00 & 0.32 & -0.01 \\
##   3 & 0.29 & 0.32 & 1.00 & -0.02 \\
##   4 & 0.05 & -0.01 & -0.02 & 1.00 \\
##    \hline
## \end{tabular}
## \end{table}
```

#Dimensionality Reduction

```
simple.fit = lm(gender~score, df)
summary(simple.fit)
```

```
##
## Call:
## lm(formula = gender ~ score, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8709 -0.4882 -0.3987  0.4949  1.5813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.36195    0.02002   18.08   <2e-16 ***
## score        0.07920    0.00897    8.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5303 on 2998 degrees of freedom
## Multiple R-squared:  0.02535,    Adjusted R-squared:  0.02502
## F-statistic: 77.96 on 1 and 2998 DF,  p-value: < 2.2e-16
```

```
multi.fit = lm(gender ~ score + n.photos, df)
summary(multi.fit)
```

```
##
## Call:
## lm(formula = gender ~ score + n.photos, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9065 -0.4465 -0.2158  0.4427  1.7823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 0.082684   0.026637   3.104  0.00193 **
## score       0.072217   0.008656   8.343  < 2e-16 ***
## n.photos    0.083307   0.005479  15.206  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5111 on 2997 degrees of freedom
## Multiple R-squared:  0.09516,    Adjusted R-squared:  0.09455
## F-statistic: 157.6 on 2 and 2997 DF,  p-value: < 2.2e-16
```

```
df_active <- df[,c("score", "n.matches", "n.updates.photo", "n.photos","sent.ana","length.prof")]
head(df_active)
```

```
##      score n.matches n.updates.photo n.photos sent.ana length.prof
## 1 1.495834        11               5        6 6.490446     0.00000
## 2 8.946863        56               2        6 4.589125    20.72286
## 3 2.496199        13               3        4 6.473182    31.39928
## 4 2.823579        32               5        2 5.368982     0.00000
## 5 2.117433        21               1        4 5.573949    38.51022
## 6 1.700014        14               2        6 5.464667    23.11221
```

```
#df_active <- na.omit(df_active)
res.pca <- PCA(df_active, graph= FALSE)
```

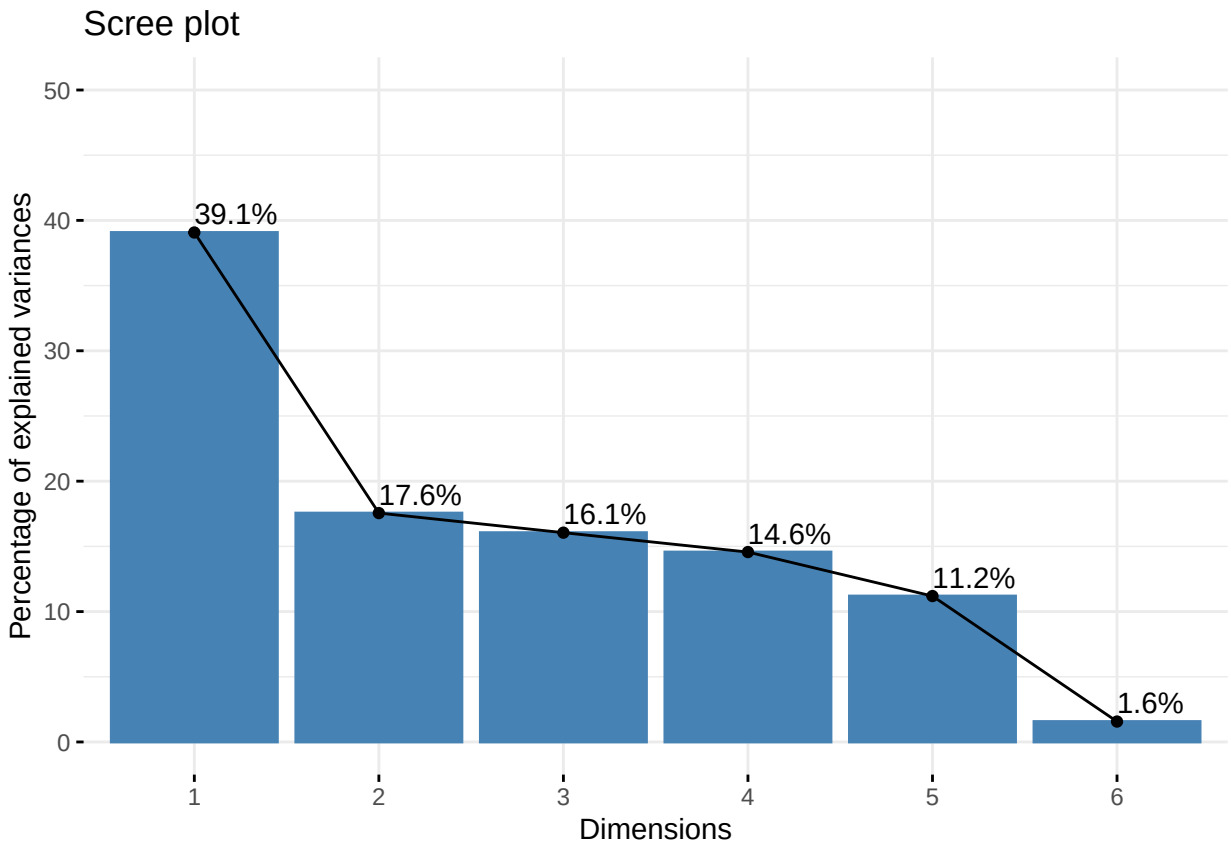
```
summary(res.pca)
```

```
##
## Call:
## PCA(X = df_active, graph = FALSE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
## Variance               2.344   1.053   0.963   0.874   0.671   0.094
## % of var.             39.072  17.554  16.052  14.565  11.190   1.567
## Cumulative % of var.  39.072  56.626  72.678  87.243  98.433 100.000
##
## Individuals (the 10 first)
##                     Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## 1                |  2.790 |  0.312  0.001  0.013 | -1.525  0.074  0.299 | -0.047
## 2                |  7.701 |  6.164  0.540  0.641 | -1.479  0.069  0.037 |  1.665
## 3                |  1.488 |  0.504  0.004  0.115 |  0.526  0.009  0.125 |  0.861
## 4                |  2.907 |  2.110  0.063  0.527 | -0.039  0.000  0.000 | -1.479
## 5                |  1.684 |  0.186  0.000  0.012 |  0.738  0.017  0.192 |  1.341
## 6                |  1.579 | -0.247  0.001  0.025 | -0.709  0.016  0.202 |  1.277
## 7                |  1.971 | -1.554  0.034  0.621 | -0.985  0.031  0.250 | -0.575
## 8                |  1.703 | -1.368  0.027  0.645 |  0.820  0.021  0.232 |  0.525
## 9                |  2.596 | -2.143  0.065  0.681 | -0.012  0.000  0.000 |  0.622
## 10               |  2.667 |  2.107  0.063  0.624 | -1.133  0.041  0.180 | -0.048
##                     ctr   cos2
## 1                 0.000  0.000 |
## 2                 0.096  0.047 |
```

```
## 3                0.026  0.335 |
## 4                0.076  0.259 |
## 5                0.062  0.633 |
## 6                0.056  0.654 |
## 7                0.011  0.085 |
## 8                0.010  0.095 |
## 9                0.013  0.057 |
## 10               0.000  0.000 |
##
## Variables
##                    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## score            |  0.917 35.872  0.841 | -0.082  0.646  0.007 |  0.058  0.355
## n.matches        |  0.936 37.405  0.877 | -0.015  0.021  0.000 |  0.025  0.067
## n.updates.photo |  0.486 10.075  0.236 |  0.023  0.049  0.001 | -0.117  1.416
## n.photos         |  0.005  0.001  0.000 | -0.746 52.858  0.557 |  0.649 43.679
## sent.ana         |  0.623 16.546  0.388 |  0.185  3.248  0.034 |  0.019  0.036
## length.prof      | -0.049  0.102  0.002 |  0.674 43.179  0.455 |  0.724 54.447
##                    cos2
## score            0.003 |
## n.matches        0.001 |
## n.updates.photo  0.014 |
## n.photos         0.421 |
## sent.ana         0.000 |
## length.prof      0.524 |
```

##Scree

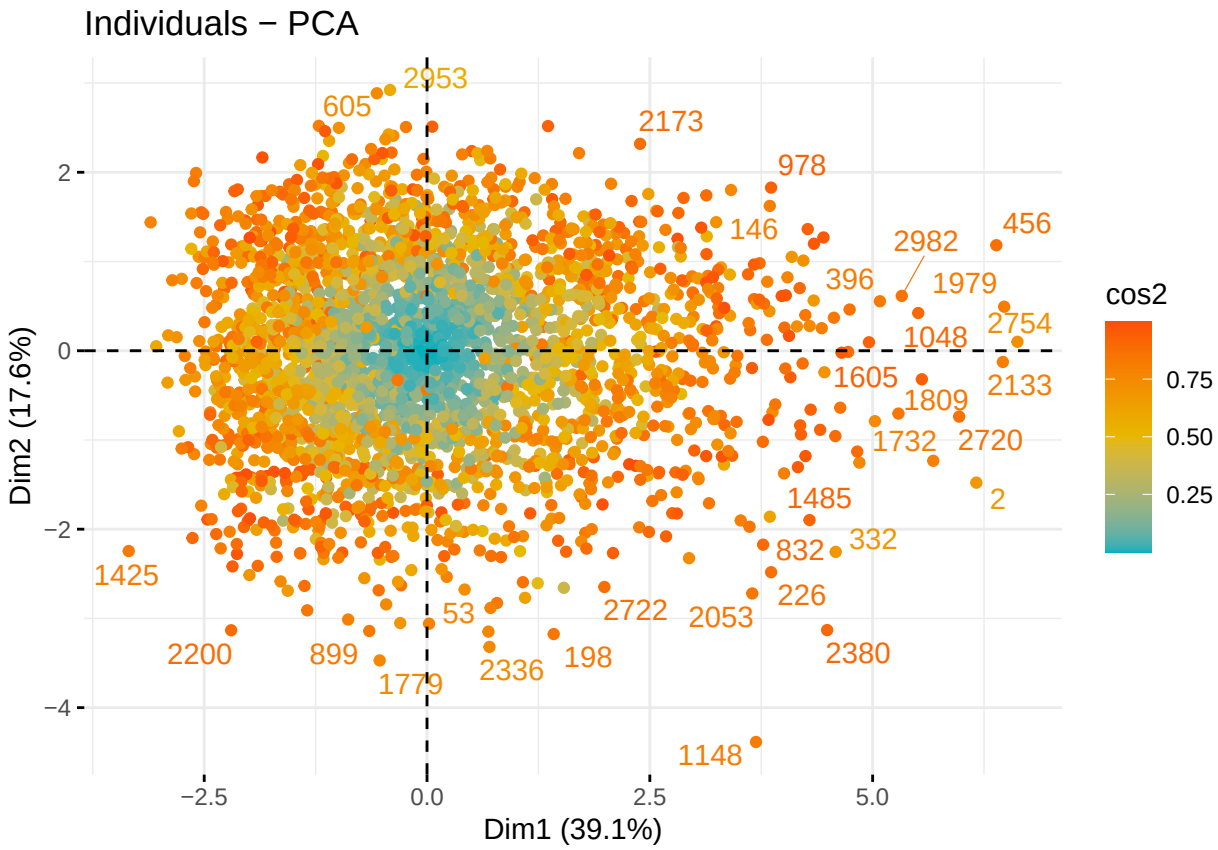
```
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 50))
```



##The Individuals factor map

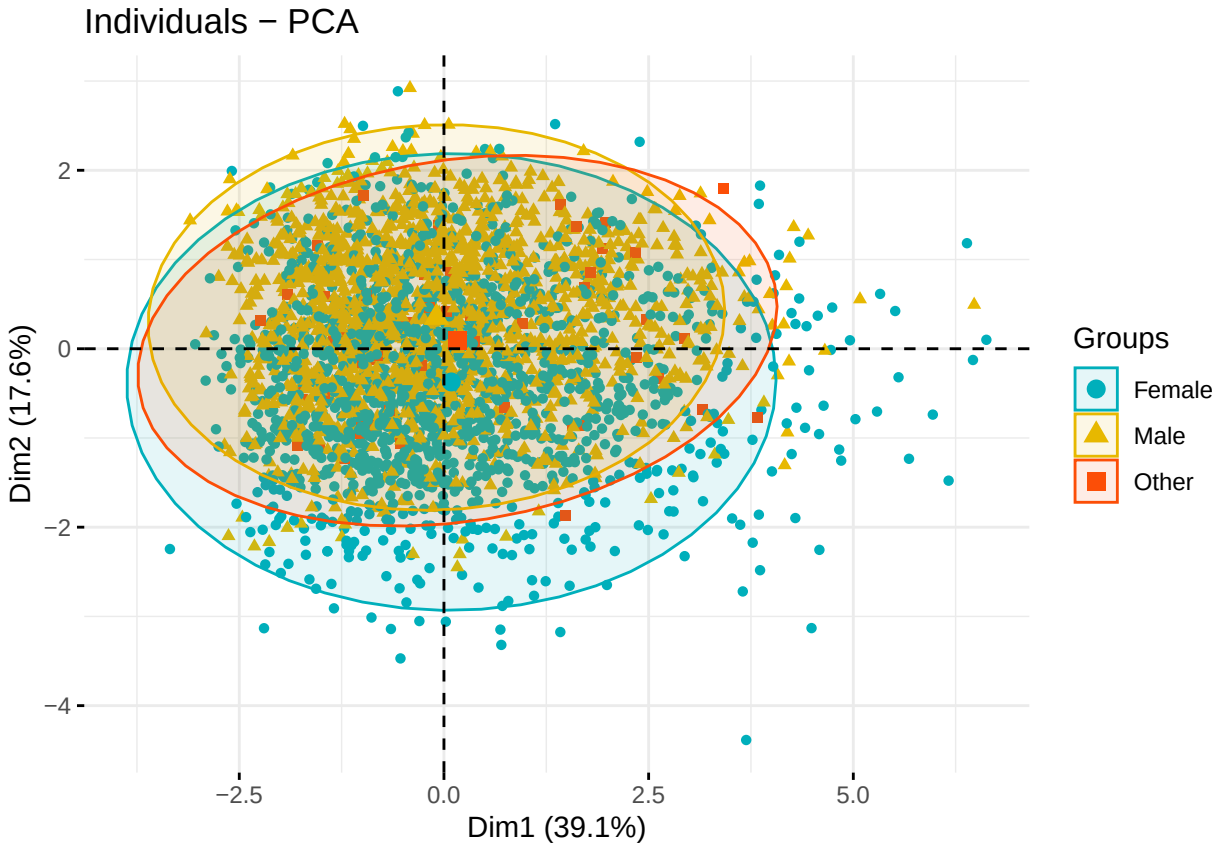
```
fviz_pca_ind(res.pca, col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```

```
## Warning: ggrepel: 2968 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



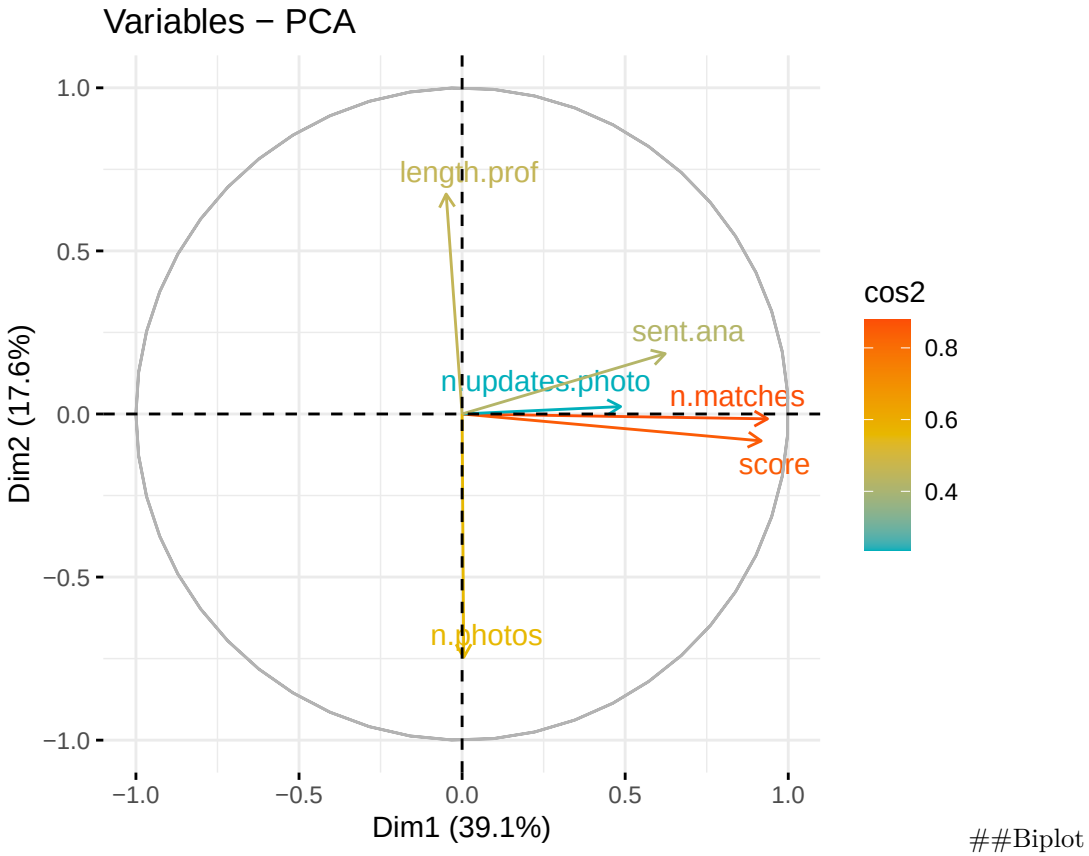

```
df <- df %>% mutate(Gender.c =case_when (gender == 0 ~ "Male",
                                         gender == 1 ~ "Female",
                                         gender == 2 ~ "Other"))
```

```
fviz_pca_ind(res.pca,
             geom.ind = "point", # show points only (nbut not "text")
             col.ind = df$Gender.c, # color by groups
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE, # Concentration ellipses
             legend.title = "Groups"
             )
```



##Variables factor map

```
fviz_pca_var(res.pca,
             col.var = "cos2", # Color by qualities of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



##Biplot

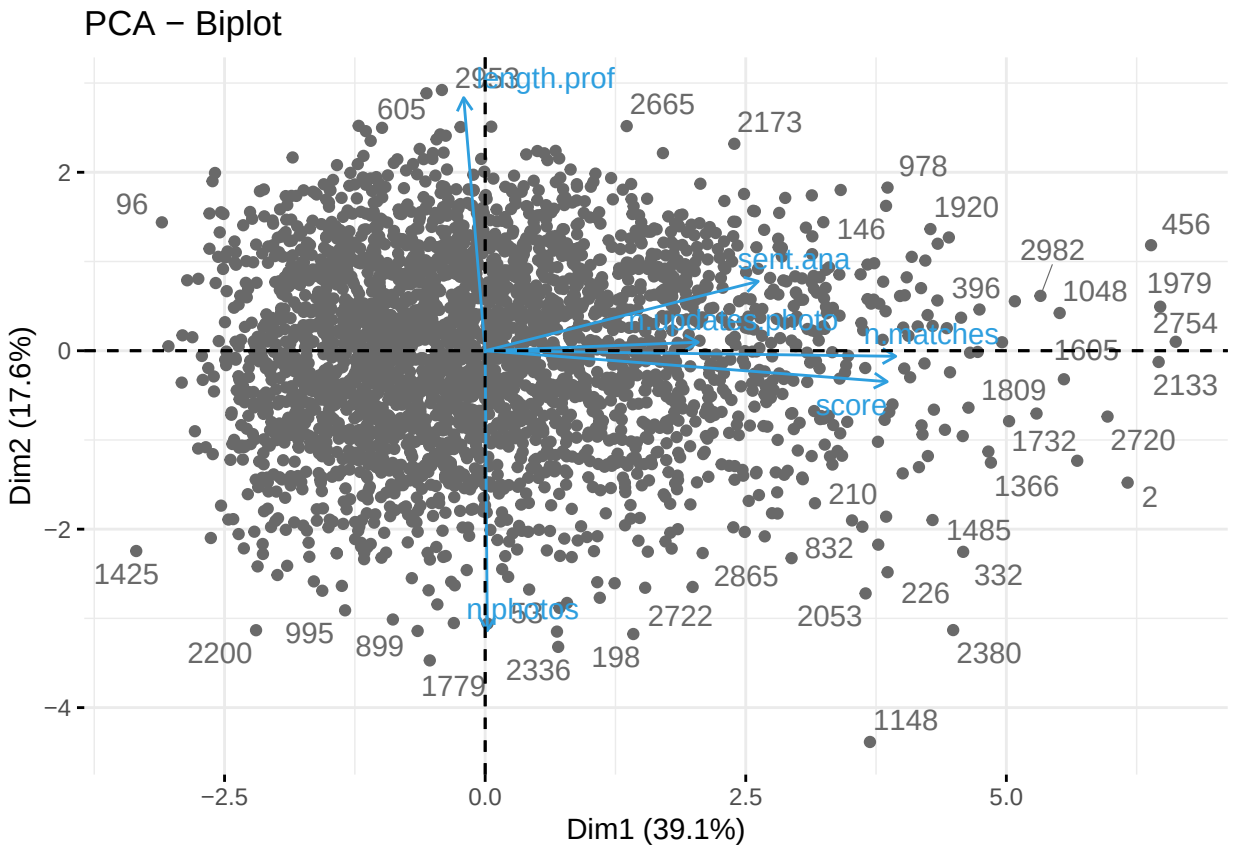
```
library(lares)
```

```
##
## Attaching package: 'lares'

## The following objects are masked from 'package:hrbrthemes':
##
##     scale_x_comma, scale_x_percent, scale_y_comma, scale_y_percent
```

```
fviz_pca_biplot(res.pca, repel = TRUE,
                col.var = "#2E9FDF", # Variables color
                col.ind = "#696969"  # Individuals color
                )
```

```
## Warning: ggrepel: 2961 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



##Table

```
# PCA with function prcomp
pca1 = prcomp(df_active, scale. = TRUE)

# sqrt of eigenvalues
pca1$sdev
```

```
## [1] 1.5311123 1.0262743 0.9814004 0.9348220 0.8193820 0.3066432
```

```
# loadings
xtable(pca1$rotation)
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Fri Nov 19 10:36:19 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
##   \hline
##  & PC1 & PC2 & PC3 & PC4 & PC5 & PC6 \\
##   \hline
## score & 0.60 & -0.08 & 0.06 & 0.08 & 0.38 & -0.69 \\
##   n.matches & 0.61 & -0.01 & 0.03 & 0.07 & 0.32 & 0.72 \\
##   n.updates.photo & 0.32 & 0.02 & -0.12 & -0.88 & -0.34 & -0.02 \\
##   n.photos & 0.00 & -0.73 & 0.66 & -0.04 & -0.18 & 0.04 \\
##   sent.ana & 0.41 & 0.18 & 0.02 & 0.45 & -0.77 & -0.04 \\
```

```
##   length.prof & -0.03 & 0.66 & 0.74 & -0.13 & 0.08 & -0.01 \\
##    \hline
## \end{tabular}
## \end{table}
```

##MCA

```
df_mca <- df[,c("Gender.c", "photo.keke", "photo.beach", "voyage","laugh")]
df_mca$Gender.c <- as.factor(df_mca$Gender.c)
df_mca$photo.keke <- as.factor(df_mca$photo.keke)
df_mca$photo.beach <- as.factor(df_mca$photo.beach)
df_mca$voyage <- as.factor(df_mca$voyage)
df_mca$laugh <- as.factor(df_mca$laugh)
head(df_mca)
```

```
##   Gender.c photo.keke photo.beach voyage laugh
## 1   Female          0           0      0     0
## 2   Female          0           1      0     0
## 3   Female          0           1      0     0
## 4     Male          0           1      0     0
## 5     Male          0           0      0     1
## 6   Female          0           0      0     0
```

```
res.mca <- MCA(df_mca, graph = FALSE)
res.mca
```

```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 3000 individuals, described by 5 variables
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$var"            "results for the variables"
## 3  "$var$coord"      "coord. of the categories"
## 4  "$var$cos2"       "cos2 for the categories"
## 5  "$var$contrib"    "contributions of the categories"
## 6  "$var$v.test"     "v-test for the categories"
## 7  "$ind"            "results for the individuals"
## 8  "$ind$coord"      "coord. for the individuals"
## 9  "$ind$cos2"       "cos2 for the individuals"
## 10 "$ind$contrib"    "contributions of the individuals"
## 11 "$call"           "intermediate results"
## 12 "$call$marge.col" "weights of columns"
## 13 "$call$marge.li"  "weights of rows"
```

###Eigenvalues / Variances

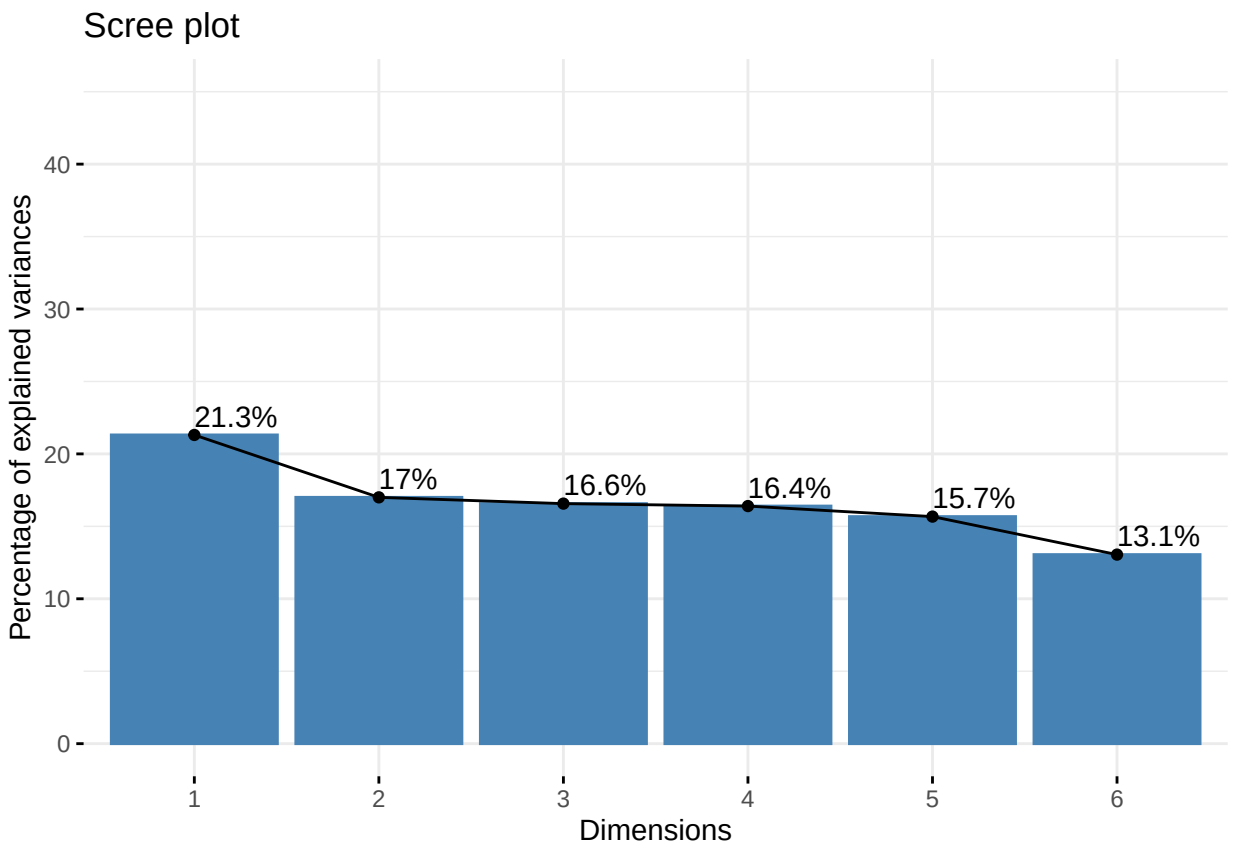
```
library("factoextra")
eig.val <- get_eigenvalue(res.mca)
head(eig.val)
```

```
##       eigenvalue variance.percent cumulative.variance.percent
```

```
## Dim.1   0.2556924          21.30770                    21.30770
## Dim.2   0.2040141          17.00117                    38.30887
## Dim.3   0.1988266          16.56888                    54.87775
## Dim.4   0.1967857          16.39880                    71.27655
## Dim.5   0.1880749          15.67291                    86.94946
## Dim.6   0.1566064          13.05054                   100.00000
```

###To visualize the percentages of inertia explained by each MCA dimensions

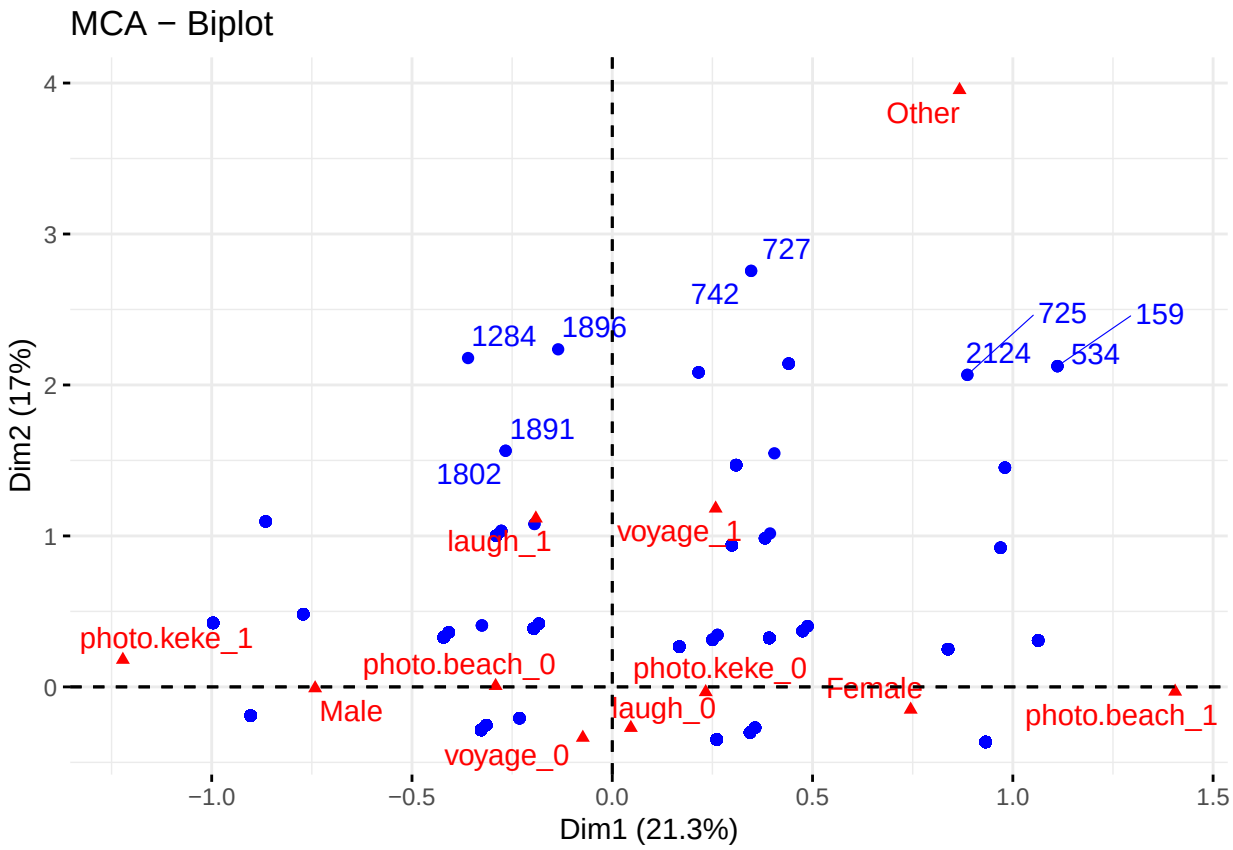
```
fviz_screeplot(res.mca, addlabels = TRUE, ylim = c(0, 45))
```



###Biplot

```
fviz_mca_biplot(res.mca, repel = TRUE,
                ggtheme = theme_minimal())
```

```
## Warning: ggrepel: 2990 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

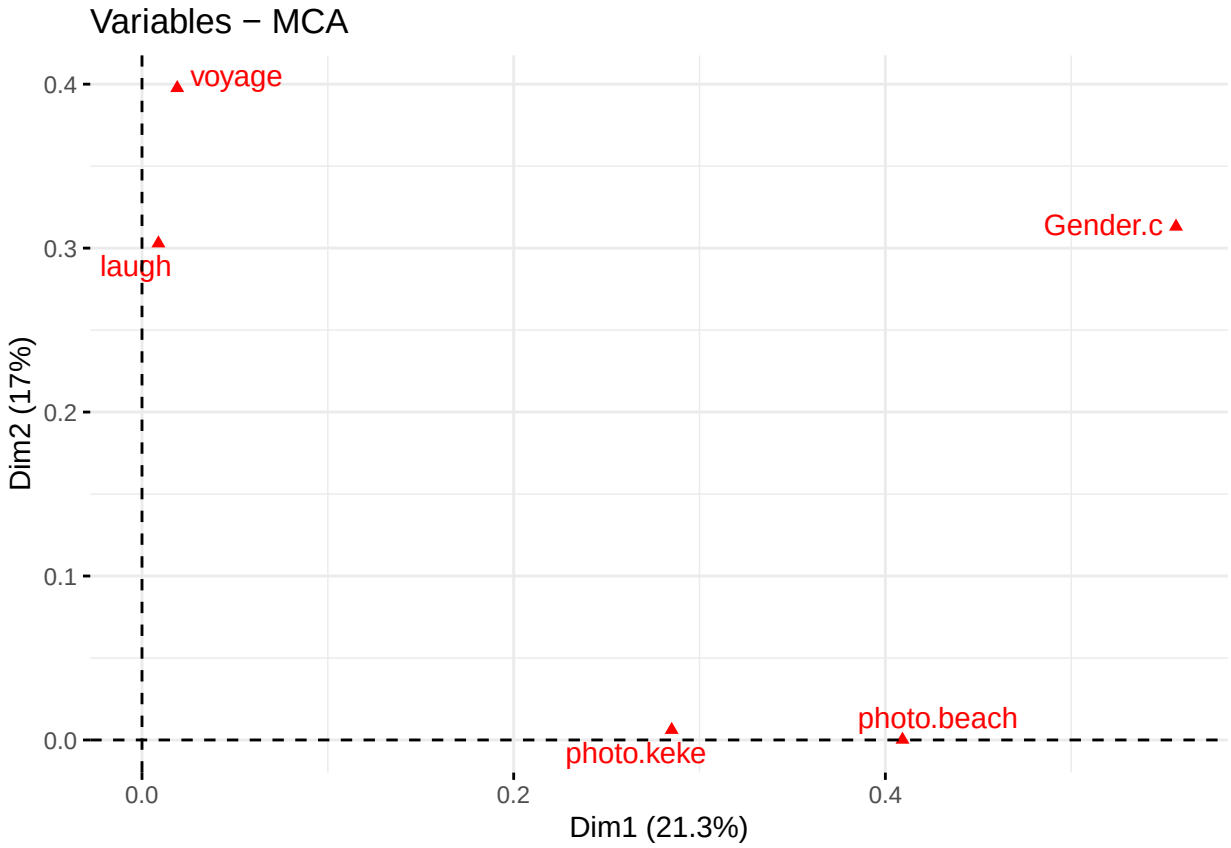


###Graph of variables

```
var <-get_mca_var(res.mca)
var
```

```
## Multiple Correspondence Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for categories"
## 2 "$cos2"    "Cos2 for categories"
## 3 "$contrib" "contributions of categories"
```

```
fviz_mca_var(res.mca, choice = "mca.cor",
             repel = TRUE, # Avoid text overlapping (slow)
             ggtheme = theme_minimal())
```

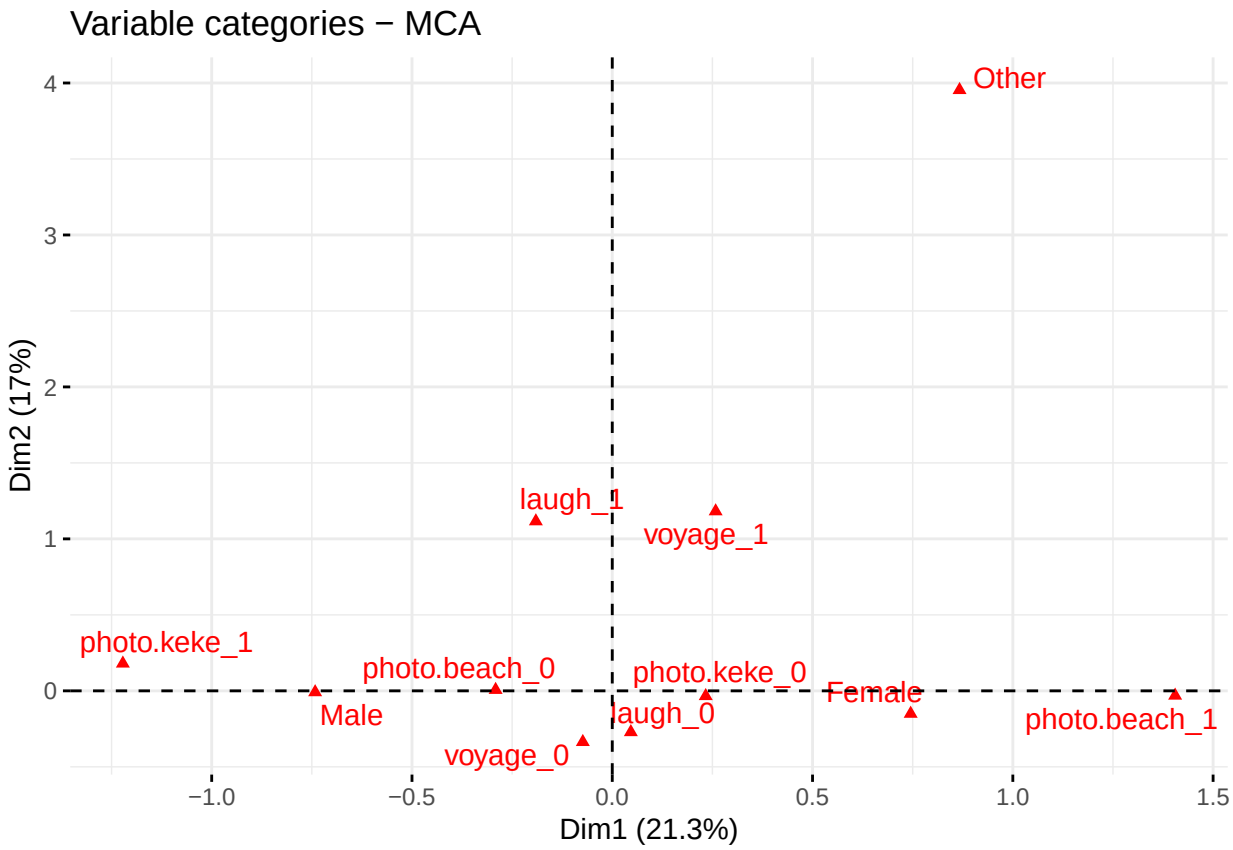



```
#Coordinates of variable categories
head(round(var$coord, 2), 4)
```

```
##              Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## Female        0.74 -0.15 -0.27 -0.05  0.02
## Male         -0.74 -0.01  0.06 -0.03 -0.06
## Other         0.87  3.95  5.35  1.97  1.14
## photo.keke_0  0.23 -0.03  0.06  0.08 -0.33
```

###Variable categories MCA

```
fviz_mca_var(res.mca,
             repel = TRUE, # Avoid text overlapping (slow)
             ggtheme = theme_minimal())
```

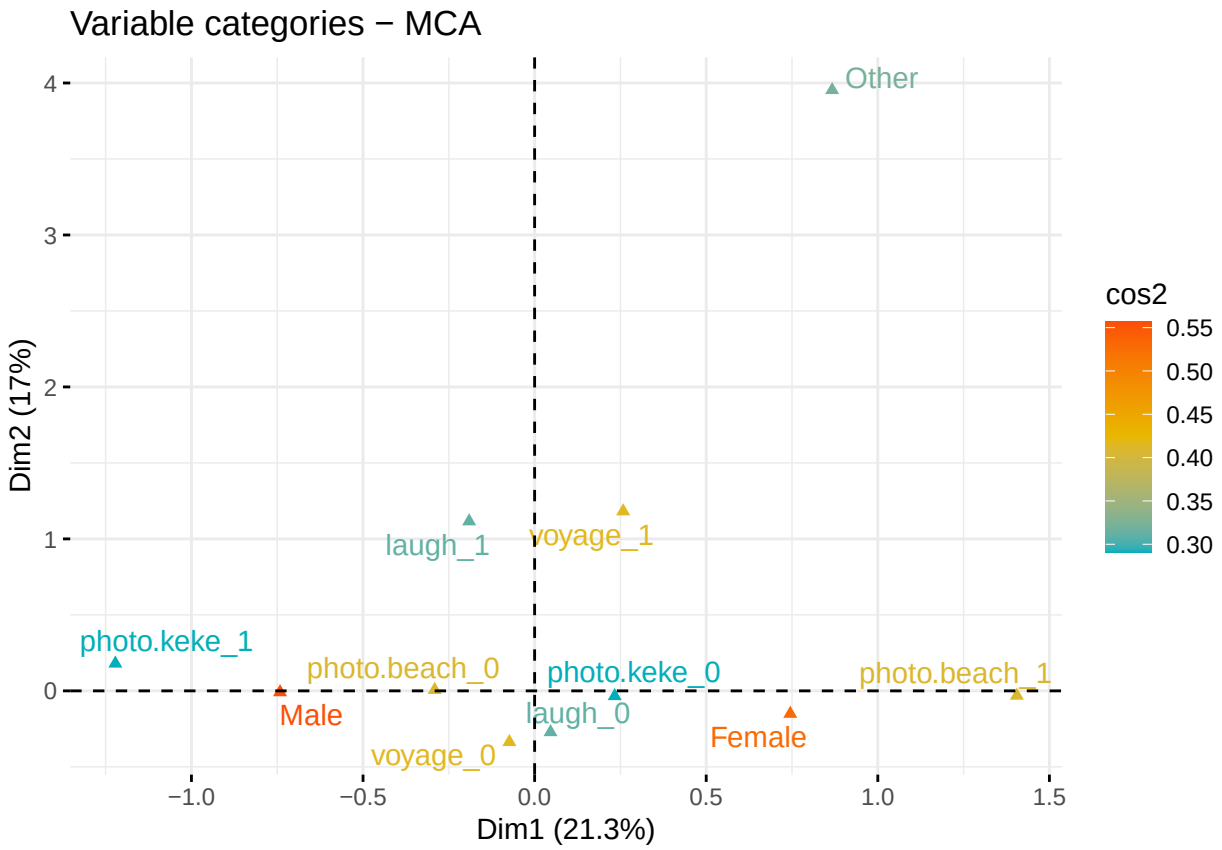


###Quality of representation of variable categories

```
head(var$cos2, 4)
```

```
##                  Dim 1        Dim 2       Dim 3        Dim 4        Dim 5
## Female       0.50822023 2.076478e-02 0.068945775 0.0024604233 0.0003244496
## Male         0.55612781 7.965194e-05 0.003079492 0.0007022995 0.0038326596
## Other        0.01481909 3.082064e-01 0.564070020 0.0762727439 0.0254266599
## photo.keke_0 0.28502604 6.173700e-03 0.019886892 0.0364299818 0.5583281400
```

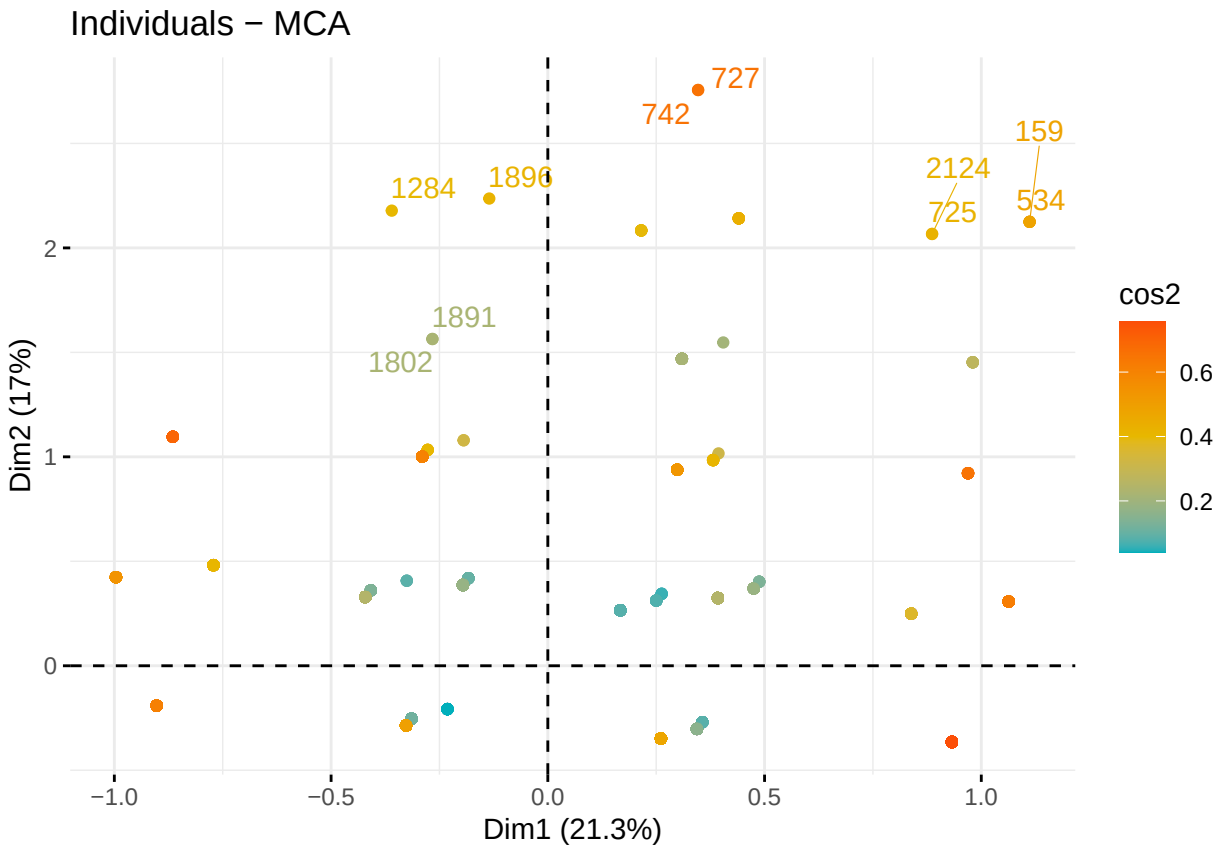
```
# Color by cos2 values: quality on the factor map
fviz_mca_var(res.mca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, # Avoid text overlapping
             ggtheme = theme_minimal())
```



Individuals MCA

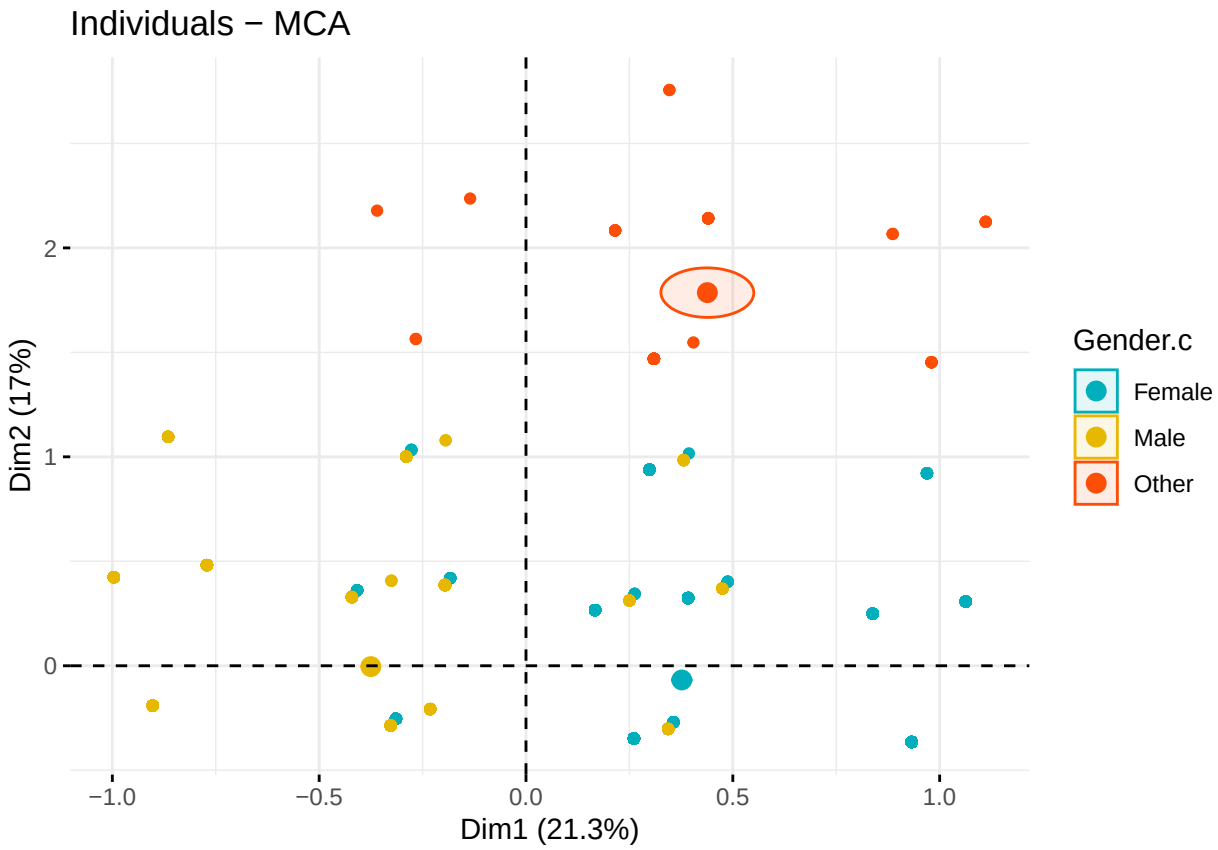
```
fviz_mca_ind(res.mca, col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, # Avoid text overlapping (slow if many points)
             ggtheme = theme_minimal())
```

```
## Warning: ggrepel: 2990 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



###Individuals by groups MCA

```
fviz_mca_ind(res.mca,
             label = "none", # hide individual labels
             habillage = "Gender.c", # color by groups
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE, ellipse.type = "confidence",
             ggtheme = theme_minimal())
```

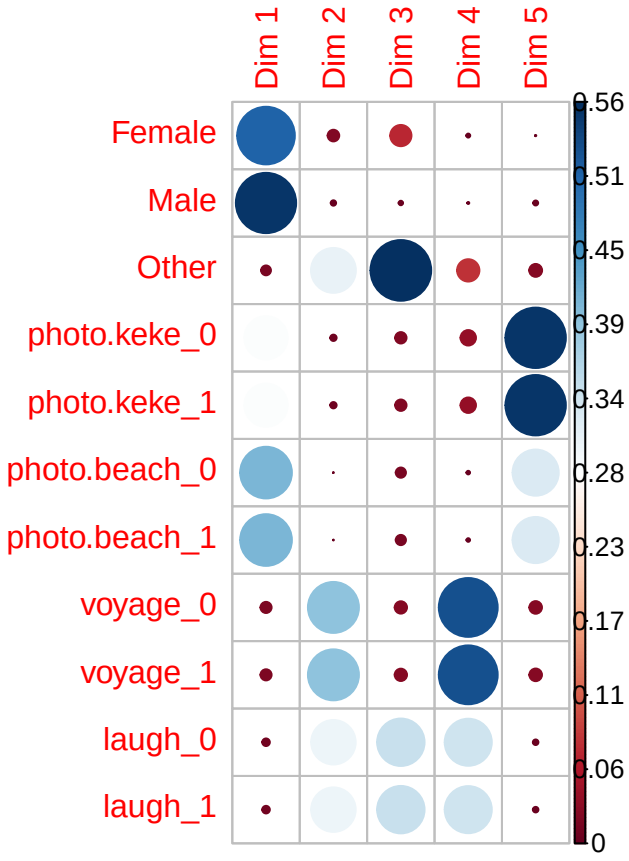


Visualize the cos2 of row categories

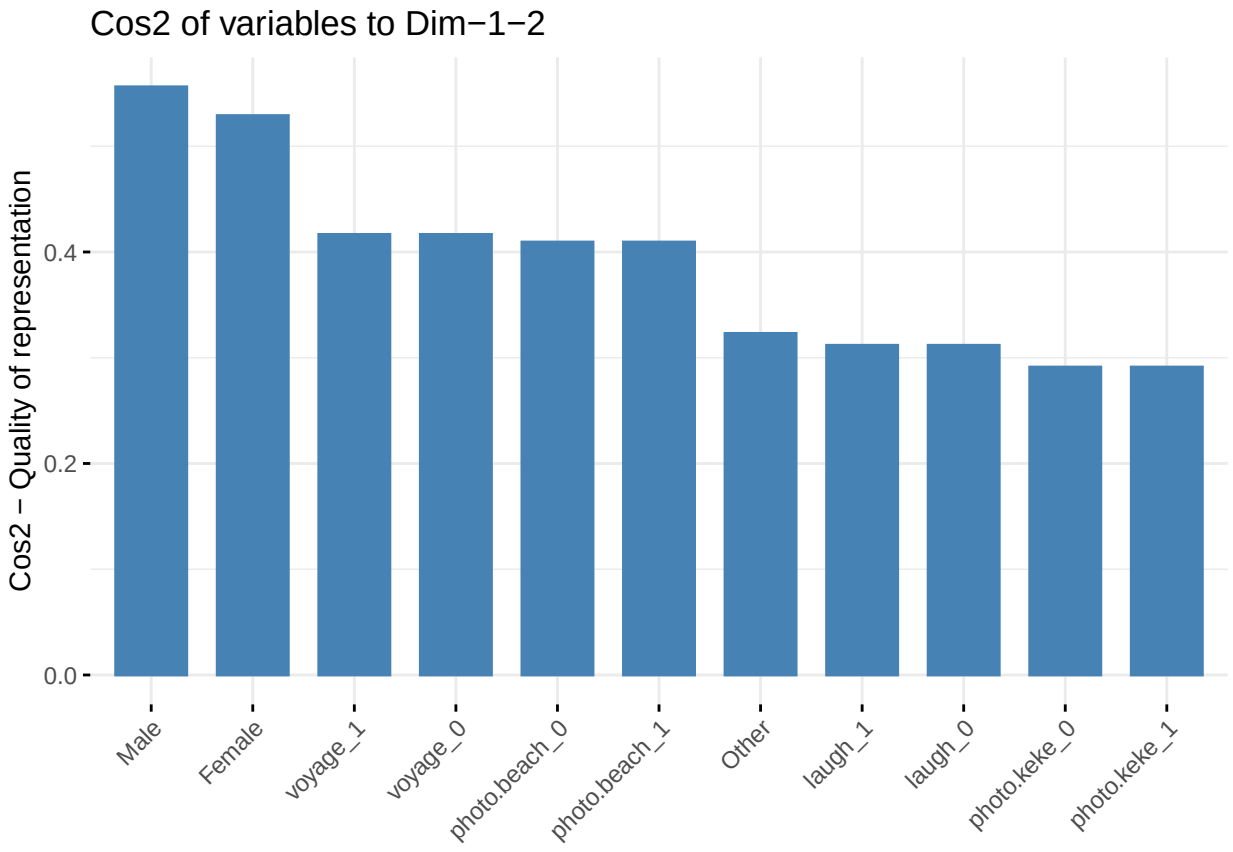
```
library("corrplot")
```

```
## corrplot 0.90 loaded
```

```
corrplot(var$cos2, is.corr=FALSE)
```



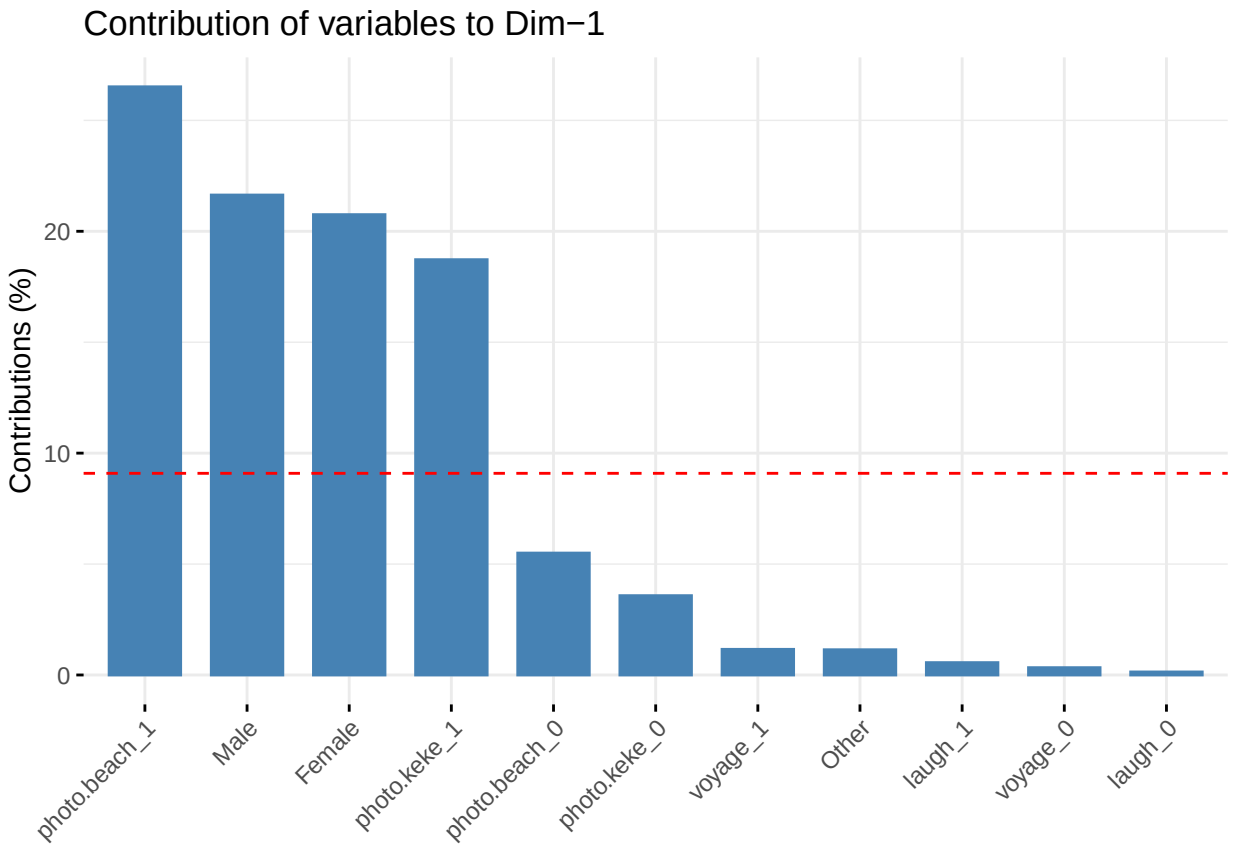

```
# Cos2 of variable categories on Dim.1 and Dim.2
fviz_cos2(res.mca, choice = "var", axes = 1:2)
```



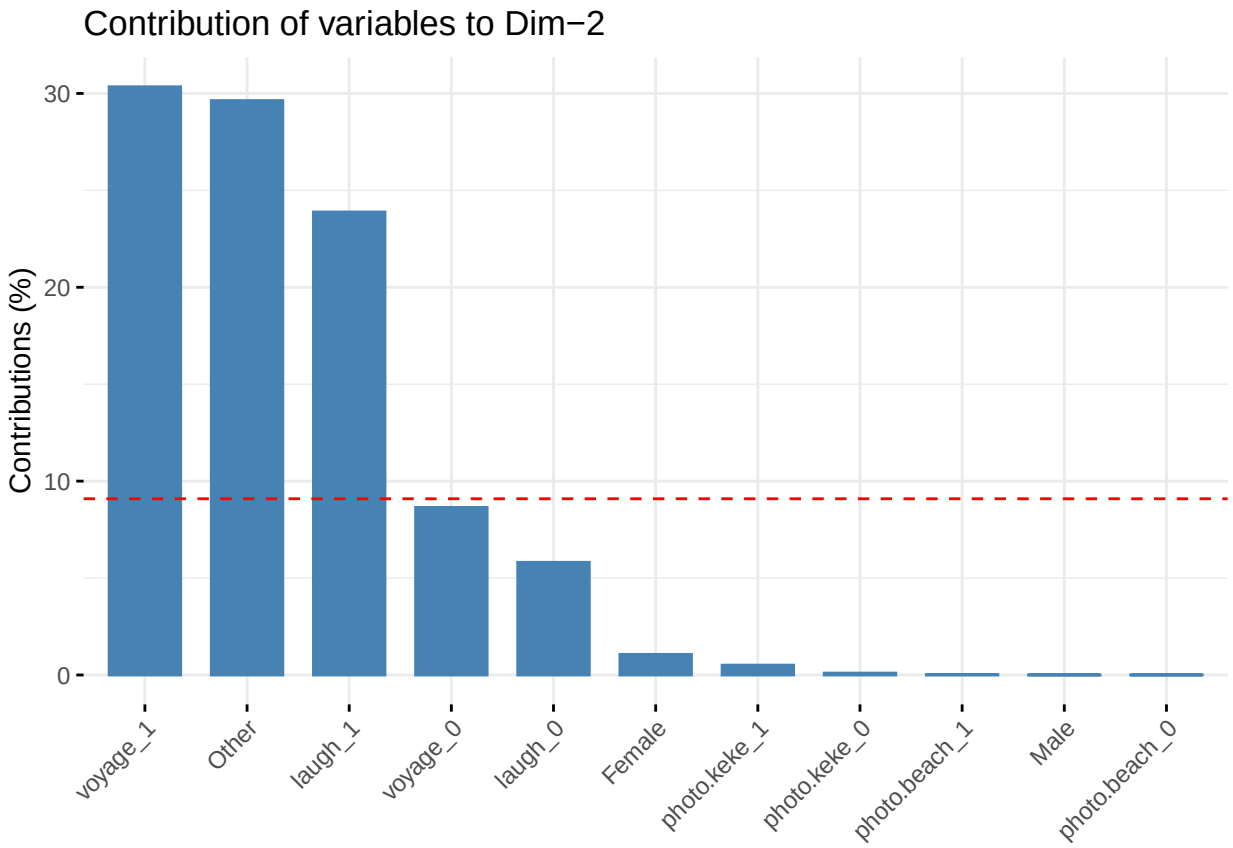

```
head(round(var$contrib,2), 4)
```

```
##              Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## Female       20.75  1.06  3.62  0.13  0.02
## Male         21.63  0.00  0.15  0.04  0.20
## Other         1.14 29.63 55.64  7.60  2.65
## photo.keke_0  3.57  0.10  0.32  0.59  9.52
```

```
# Contributions of rows to dimension 1
fviz_contrib(res.mca, choice = "var", axes = 1, top = 15)
```



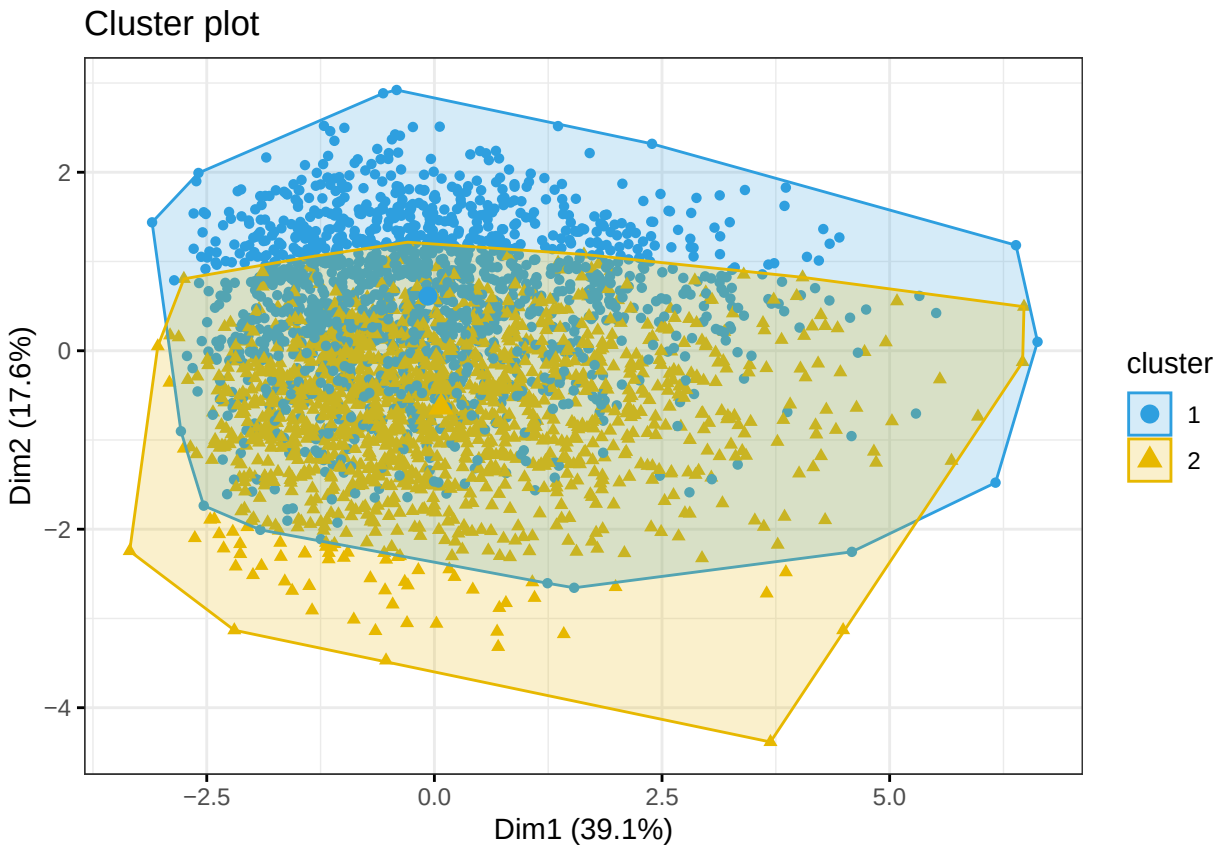

```
# Contributions of rows to dimension 2
fviz_contrib(res.mca, choice = "var", axes = 2, top = 15)
```



#K-means

```
km.res <- kmeans(df_active, 2)
```

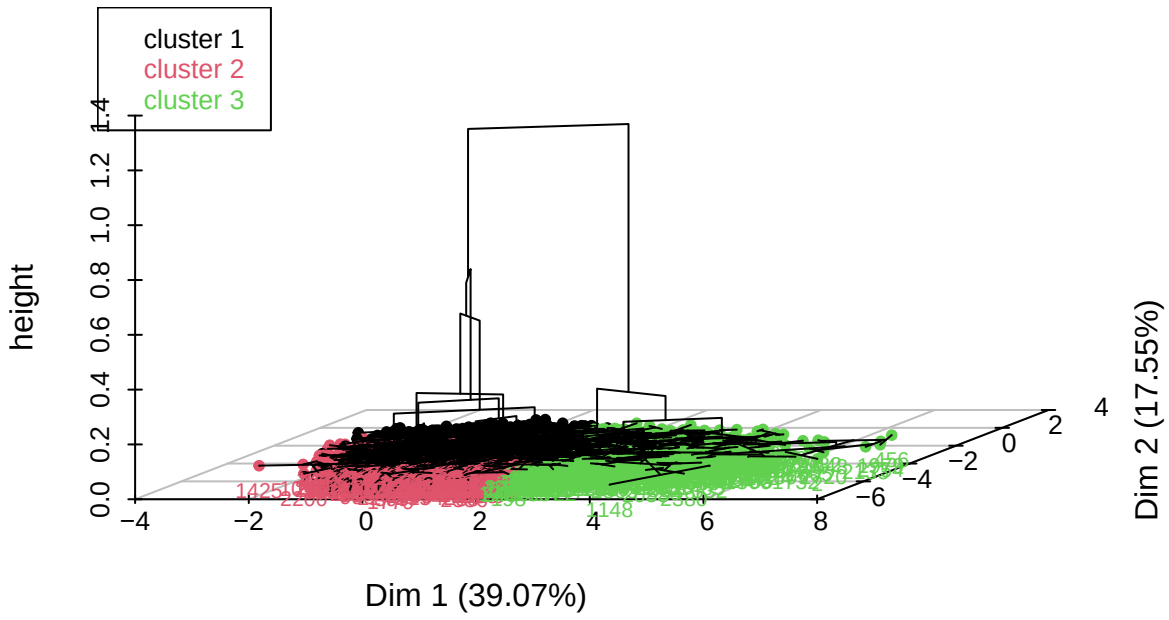
```
fviz_cluster(km.res, data = df_active,
             palette = c("#2E9FDF", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
             )
```



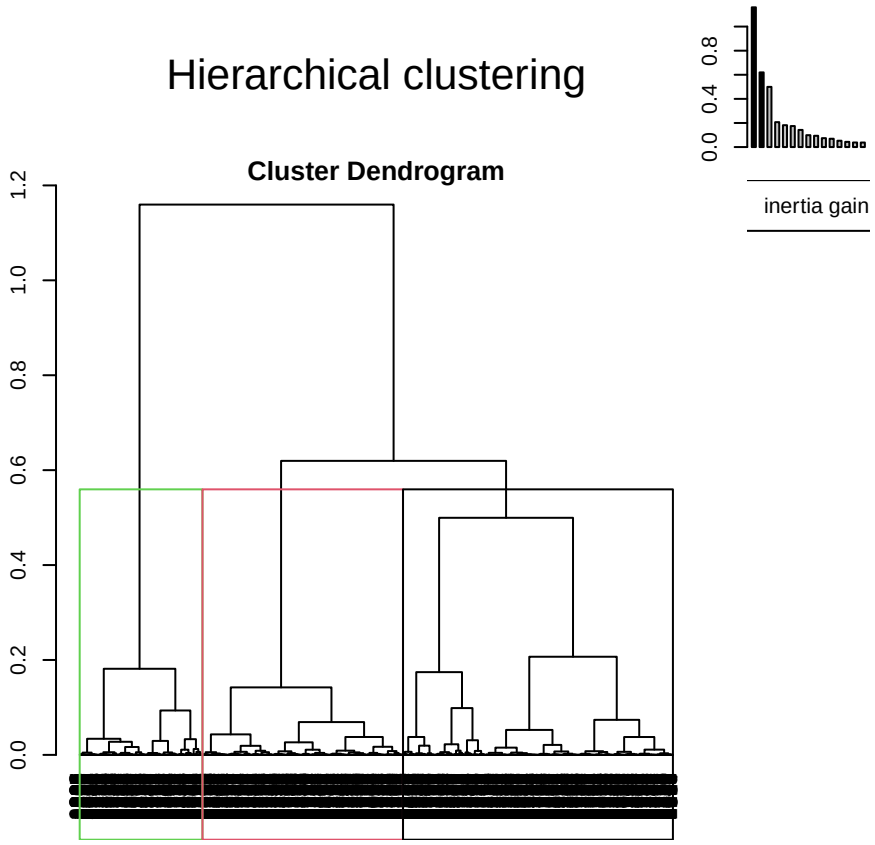
#Hierarchical Clustering

```
library(FactoMineR)
# Compute PCA with ncp = 3
res.pca <- PCA(df_active, ncp = 3, graph = FALSE)
# Compute hierarchical clustering on principal components
res.hcpc <- HCPC(res.pca, graph = FALSE)
```

```
plot(res.hcpc, choice = "3D.map")
```




```
plot(res.hcpc, choice="tree")
```



```
# Individuals facor map
fviz_cluster(res.hcpc, geom = "point", main = "Factor map")
```