Dimensionality reduction

Aya benh hriz

05/11/2021

Importing libraries

```
library('plyr')
library('dplyr')
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Hmisc)
```

```
## Le chargement a nécessité le package : lattice
```

```
## Le chargement a nécessité le package : survival
```

```
## Le chargement a nécessité le package : Formula
```

```
## Le chargement a nécessité le package : ggplot2
```

```
##
## Attachement du package : 'Hmisc'
```

```
## Les objets suivants sont masqués depuis 'package:dplyr':
##
##     src, summarize
```

```
## Les objets suivants sont masqués depuis 'package:plyr':
##
##     is.discrete, summarize
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     format.pval, units
```

```
library("viridis")
```

```
## Le chargement a nécessité le package : viridisLite
```

```
library('naniar')
library(xtable)
```

```
##
## Attachement du package : 'xtable'
```

```
## Les objets suivants sont masqués depuis 'package:Hmisc':
##
##     label, label<-
```

```
library(schoRsch)
library(gmodels)
library(readr)
library(ggpubr)
```

```
##
## Attachement du package : 'ggpubr'
```

```
## L'objet suivant est masqué depuis 'package:plyr':
##
##     mutate
```

```
library(magrittr)
library(dplyr)
library("FactoMineR")
library("factoextra")
```

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

```
library(RColorBrewer)
library(scales)
```

```
##
## Attachement du package : 'scales'
```

```
## L'objet suivant est masqué depuis 'package:readr':
##
##     col_factor
```

```
## L'objet suivant est masqué depuis 'package:viridis':
##
##     viridis_pal
```

~~~~
## Importing our data set

```r
~~~~

```
df  <- read.csv('users.db.csv')
head(df)
```

```
##   userid  date.crea    score n.matches n.updates.photo n.photos
last.connex
## 1      1 2011-09-17 1.495834        11               5        6  2011-10-
07
## 2      2 2017-01-17 8.946863        56               2        6  2017-01-
31
## 3      3 2019-05-14 2.496199        13               3        4  2019-06-
17
## 4      4 2015-11-27 2.823579        32               5        2  2016-01-
15
## 5      5 2014-11-28 2.117433        21               1        4  2015-01-
15
## 6      6 2017-06-05 1.700014        14               2        6  2017-07-
03
##   last.up.photo last.pr.update gender sent.ana length.prof voyage laugh
## 1    2011-10-02             NA      1 6.490446     0.00000      0     0
## 2    2017-02-03             NA      1 4.589125    20.72286      0     0
## 3    2019-06-19             NA      1 6.473182    31.39928      0     0
## 4    2015-12-09             NA      0 5.368982     0.00000      0     0
## 5    2015-01-02             NA      0 5.573949    38.51022      0     1
## 6    2017-06-25             NA      1 5.464667    23.11221      0     0
##   photo.keke photo.beach
## 1          0           0
## 2          0           1
## 3          0           1
## 4          0           1
## 5          0           0
## 6          0           0
```

```
df$gender=factor(df$gender)
```

```
# correlation for all variables
corr=df[,c(3,4,5,6,11,12,13,14,15,16)]
tab=round(cor(corr),
  digits = 2 # rounded to 2 decimals
)
tab
```

```
##                   score n.matches n.updates.photo n.photos sent.ana
length.prof
## score            1.00      0.90            0.29     0.05     0.39       -
0.05
## n.matches        0.90      1.00            0.32    -0.01     0.44       -
0.03
## n.updates.photo  0.29      0.32            1.00    -0.02     0.14       -
0.01
## n.photos         0.05     -0.01           -0.02     1.00    -0.05       -
0.04
```

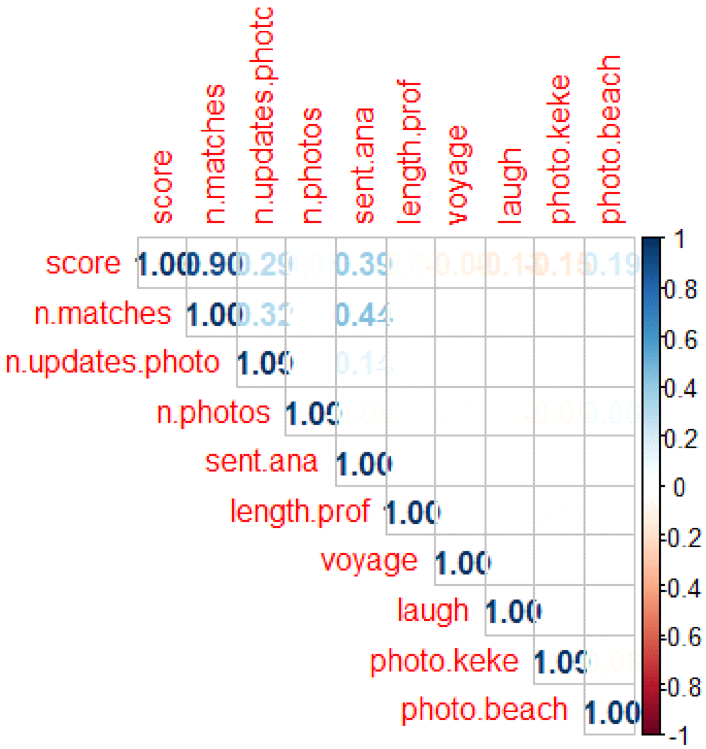
```
## sent.ana         0.39      0.44               0.14     -0.05      1.00
0.02
## length.prof     -0.05     -0.03              -0.01     -0.04      0.02
1.00
## voyage          -0.09     -0.01              -0.02      0.03     -0.01
0.00
## laugh           -0.13     -0.03              -0.01     -0.03      0.00          -
0.02
## photo.keke      -0.15      0.02               0.02     -0.05      0.01
0.04
## photo.beach      0.19     -0.01               0.00      0.06     -0.03          -
0.02
##                 voyage laugh photo.keke photo.beach
## score            -0.09 -0.13      -0.15        0.19
## n.matches        -0.01 -0.03       0.02       -0.01
## n.updates.photo  -0.02 -0.01       0.02        0.00
## n.photos          0.03 -0.03      -0.05        0.06
## sent.ana         -0.01  0.00       0.01       -0.03
## length.prof       0.00 -0.02       0.04       -0.02
## voyage            1.00  0.01       0.00        0.01
## laugh             0.01  1.00       0.02       -0.01
## photo.keke        0.00  0.02       1.00       -0.06
## photo.beach       0.01 -0.01      -0.06        1.00
```

```
#This correlation matrix gives an overview of the correlations for all
combinations of two variables.
```

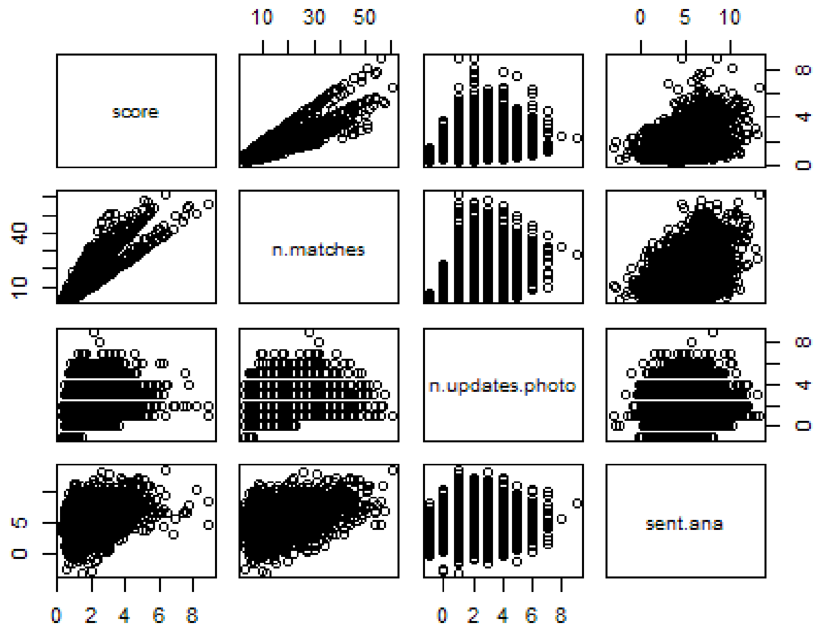
```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
corrplot(cor(corr),
  method = "number",
  type = "upper" # show only upper side
)
```



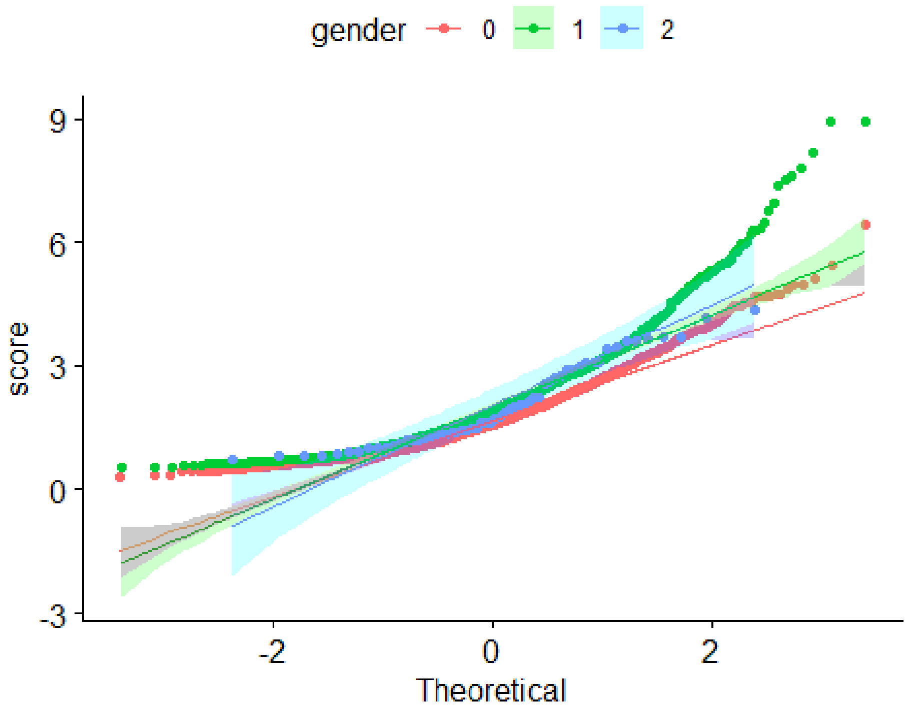
```
# multiple scatterplots
pairs(df[, c("score", "n.matches", "n.updates.photo","sent.ana")])
```



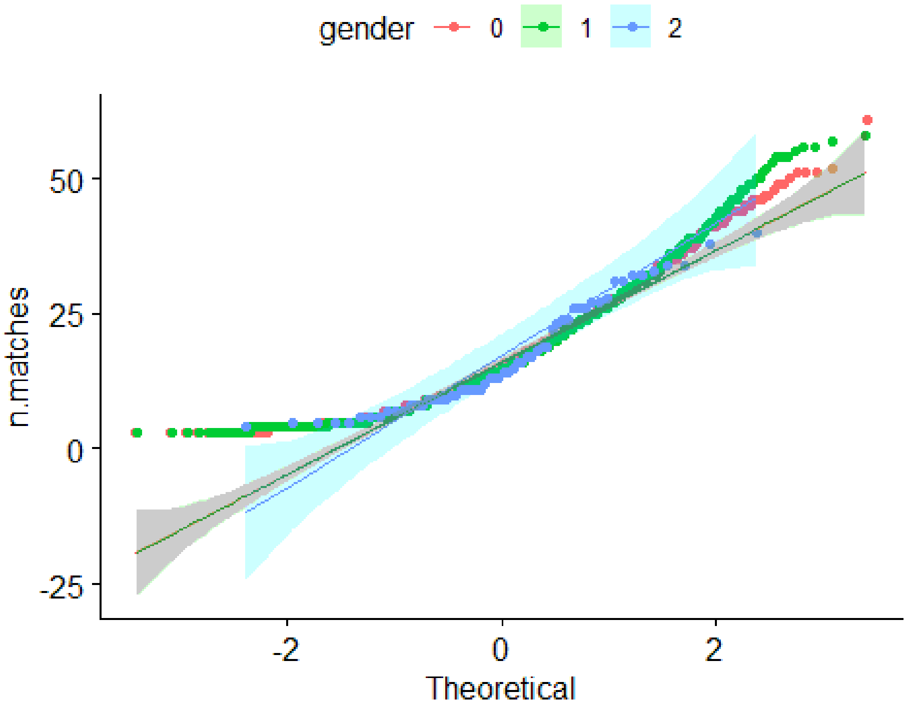
#The figure indicates that score is positively correlated with n.matches, n.updates.photo and sent.ana.

##dentifying correlations in the variables

```
ggqqplot(df,x="score",color="gender",ylab = "score")
```



```
ggqqplot(df,x='n.matches',color="gender",ylab = "n.matches")
```

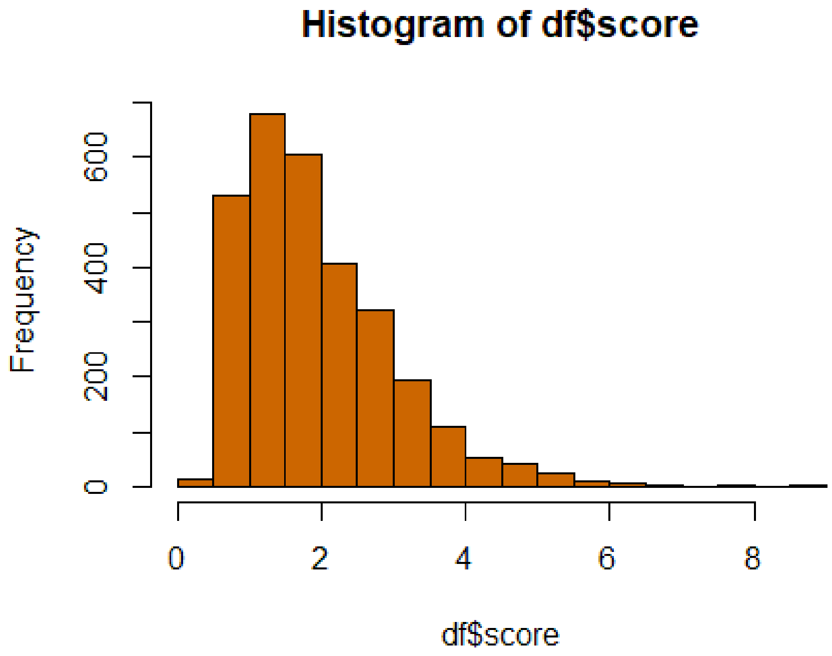


```
#check if it's normal it doesn't follow a straight line so it's not normal
and we use  then spearman's
```

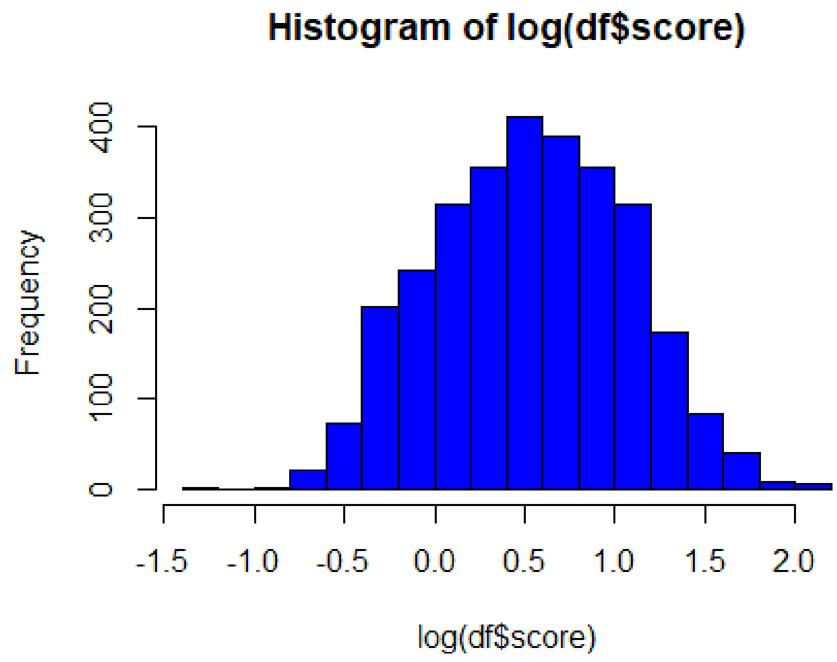
```
tab=cor.test(df$score,df$n.matches, method = 'spearman')
```

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

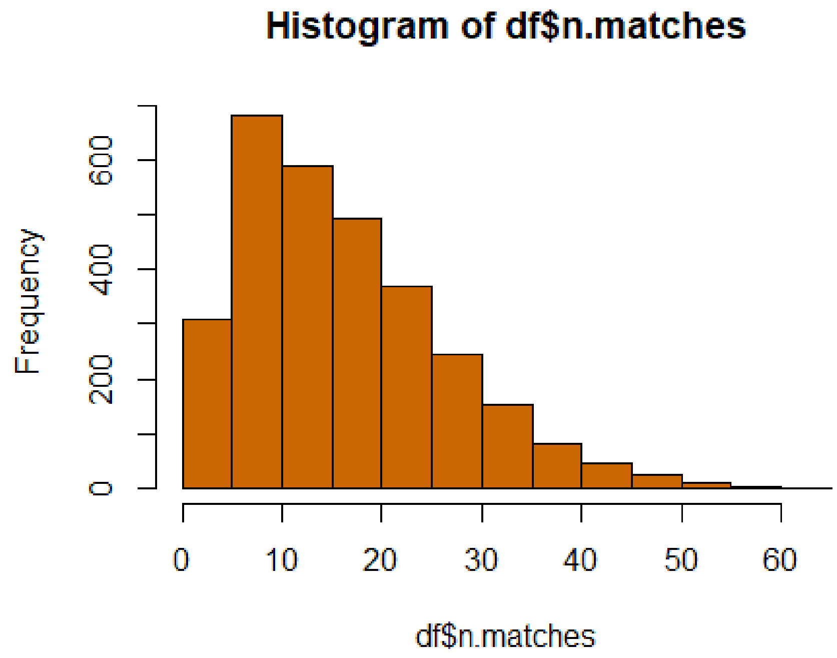
```
hist(df$score,col="chocolate")
```



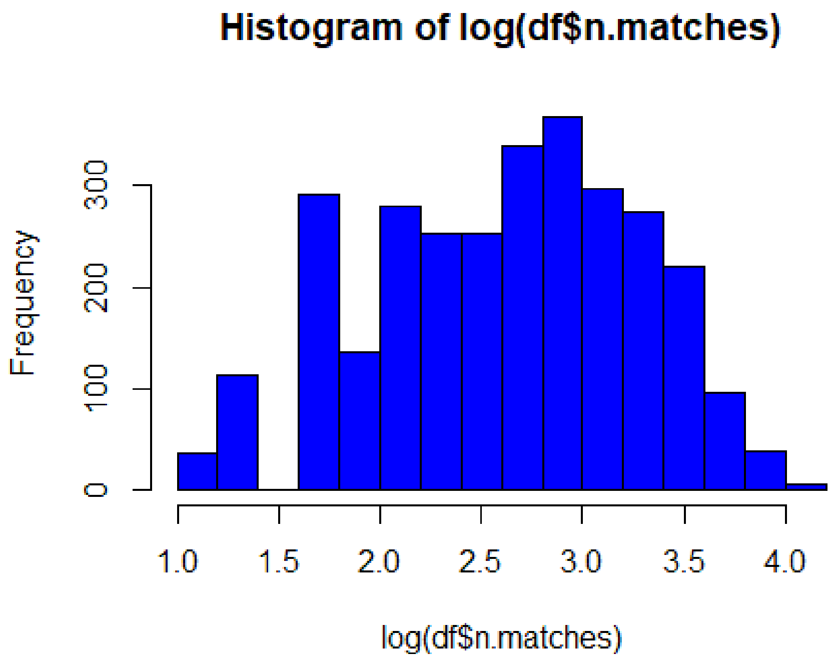
```
hist(log(df$score),col="blue")
```



```
hist(df$n.matches,col="chocolate")
```



```
hist(log(df$n.matches),col="blue")
```



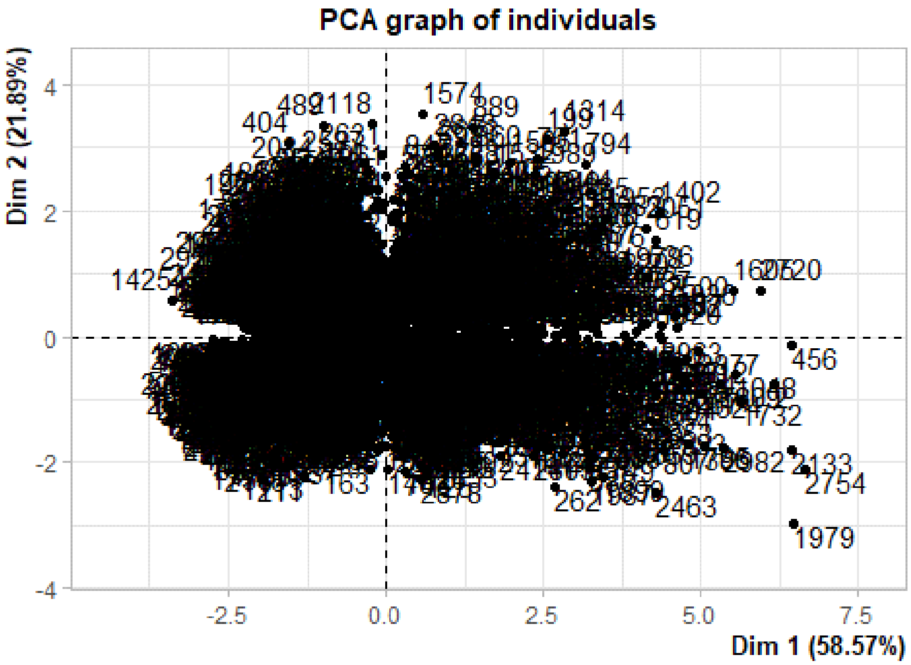
```
#when it's not normally distributed we have to use log so that we can use lm
mod <- lm(log(n.matches) ~
log(df$score)+gender+photo.keke+gender*photo.beach, data = df)
summary(mod)

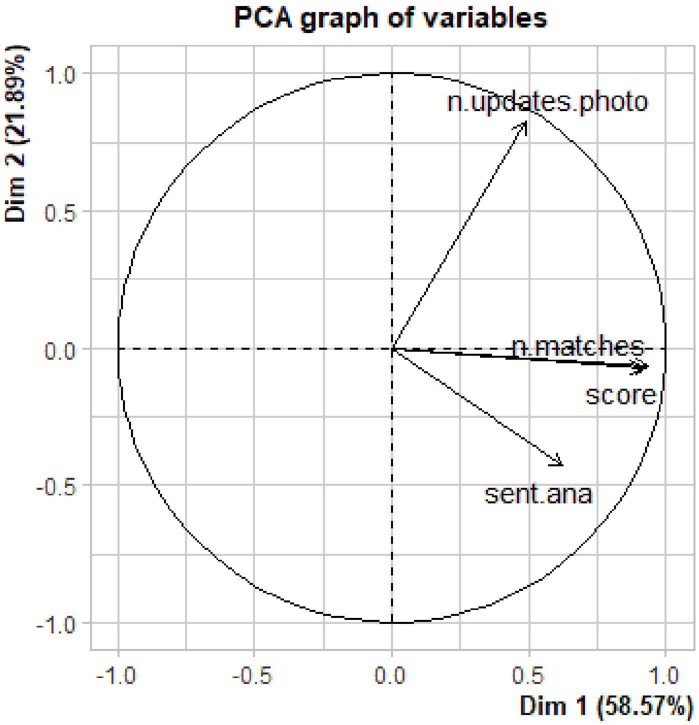
##
## Call:
## lm(formula = log(n.matches) ~ log(df$score) + gender + photo.keke +
##     gender * photo.beach, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63095 -0.07556 -0.01281  0.08869  0.44190
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.999728   0.004560 438.511  < 2e-16 ***
## log(df$score)       1.241509   0.004761 260.776  < 2e-16 ***
## gender1            -0.056613   0.005413 -10.459  < 2e-16 ***
## gender2            -0.119253   0.020198  -5.904 3.94e-09 ***
## photo.keke          0.305489   0.006714  45.503  < 2e-16 ***
## photo.beach         0.269180   0.011386  23.641  < 2e-16 ***
## gender1:photo.beach -0.775623   0.014222 -54.539  < 2e-16 ***
## gender2:photo.beach -0.246712   0.041960  -5.880 4.57e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1317 on 2992 degrees of freedom
## Multiple R-squared:  0.9579, Adjusted R-squared:  0.9578
## F-statistic:  9727 on 7 and 2992 DF,  p-value: < 2.2e-16
```

Dimensionality Reduction

PCA

```
#keep only the nuemrical variables for scaling
df2=df[,c("score","n.matches","n.updates.photo","sent.ana")]
#inertia drops when you add more variables
#when scaling we only keep the numeric variables
res.pca=PCA(df2, scale.unit = TRUE, graph = FALSE)
PCA(df2, scale.unit = TRUE, graph = TRUE)
```





```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 3000 individuals, described by 4 variables
## *The results are available in the following objects:
##
##    name                 description
## 1  "$eig"               "eigenvalues"
## 2  "$var"              "results for the variables"
## 3  "$var$coord"        "coord. for the variables"
## 4  "$var$cor"          "correlations variables - dimensions"
## 5  "$var$cos2"         "cos2 for the variables"
## 6  "$var$contrib"      "contributions of the variables"
## 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"             "summary statistics"
## 12 "$call$centre"      "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"       "weights for the individuals"
## 15 "$call$col.w"       "weights for the variables"
```

```
mt.pca <- prcomp(df2, center = TRUE,scale. = TRUE)
```

```
summary(mt.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     1.5307 0.9358 0.8280 0.30937
## Proportion of Variance 0.5857 0.2189 0.1714 0.02393
## Cumulative Proportion  0.5857 0.8047 0.9761 1.00000
```

```
library(devtools)
```

```
## Le chargement a nécessité le package : usethis
```

```
install_github("vqv/ggbiplot")
```

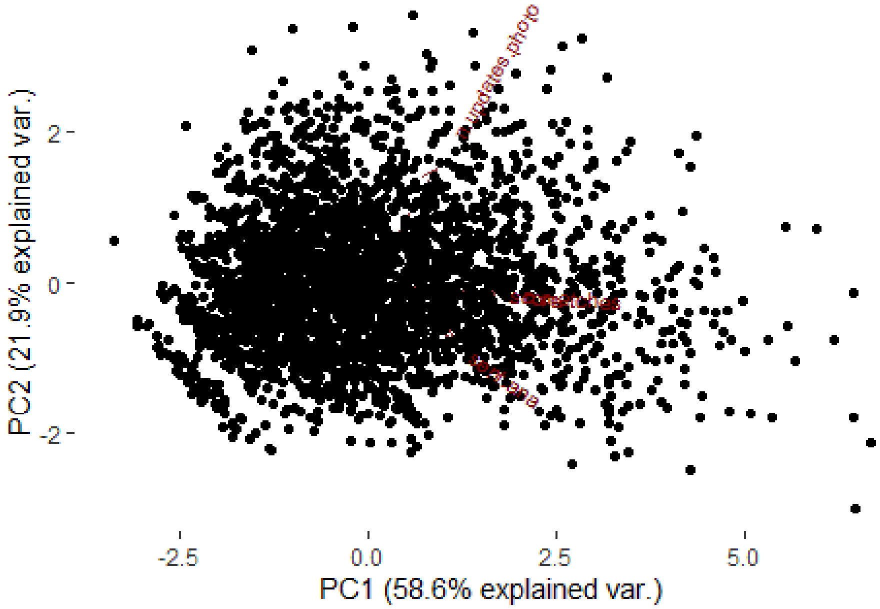
```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools 4.0 from https://cran.r-project.org/bin/windows/Rtools/.
```

```
## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last install.
##   Use `force = TRUE` to force installation
```

```
library(ggbiplot)
```

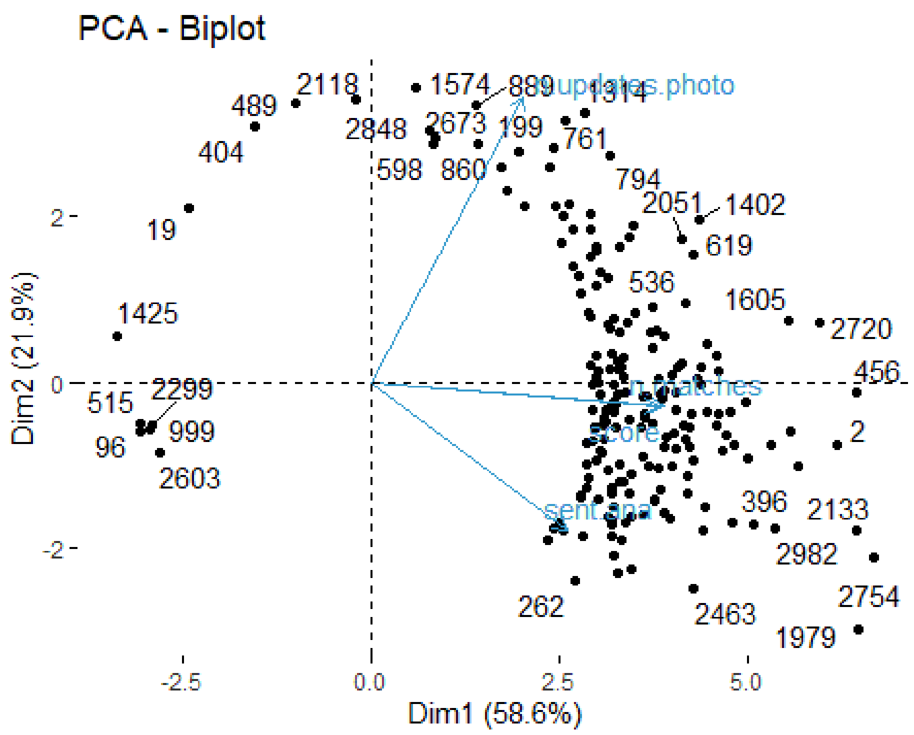
```
## Le chargement a nécessité le package : grid
```

```
ggbiplot(mt.pca,obs.scale = 1, var.scale = 1)
```



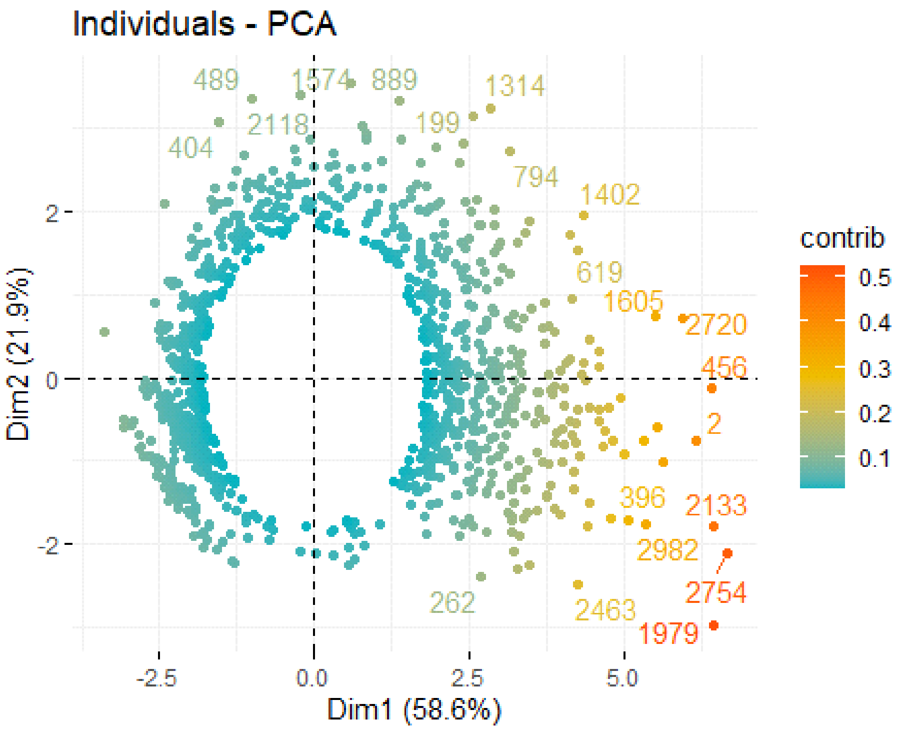
```
fviz_pca_biplot(res.pca,
                repel = TRUE, # Avoid text overlapping (slow if many point)
                ggtheme = theme_minimal(),select.ind=list(contrib=200))
```

```
## Warning: ggrepel: 165 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



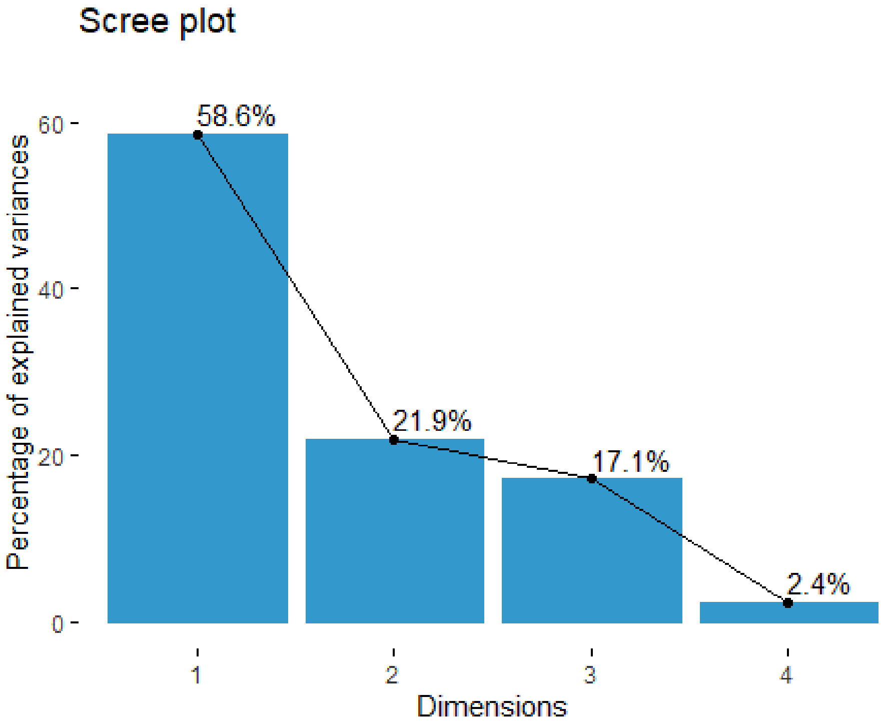
```
fviz_pca_ind(res.pca,
                repel = TRUE, # Avoid text overlapping (slow if many point)
                ggtheme = theme_minimal(),select.ind=list(contrib=1000),
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),col.ind =
"contrib")
```

```
## Warning: ggrepel: 979 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
#the further the obs is far from the origin for the x axis the highest the contirb for the 1st pr comp
```

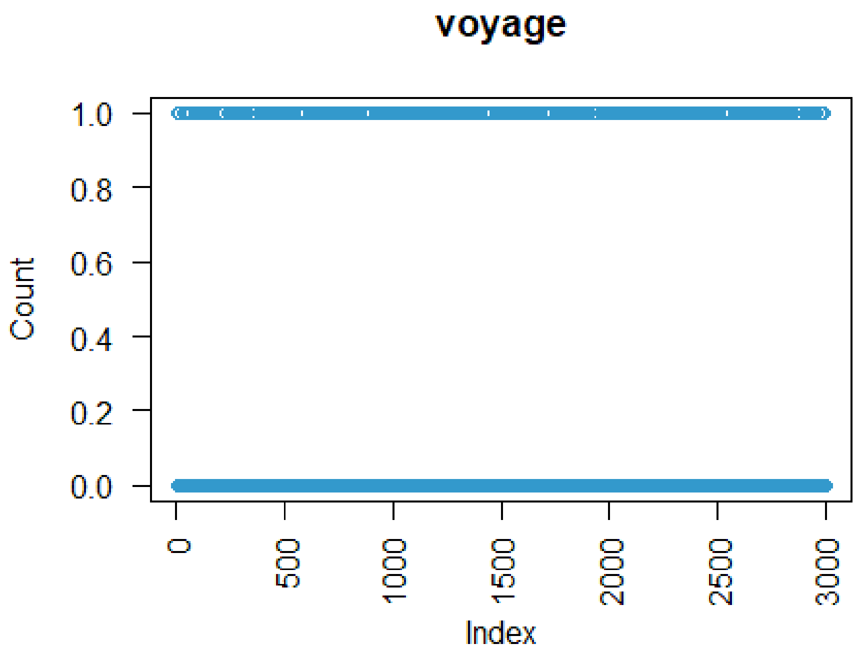
```
fviz_screeplot(res.pca, addlabels = TRUE, ylim = c(0, 65))
```



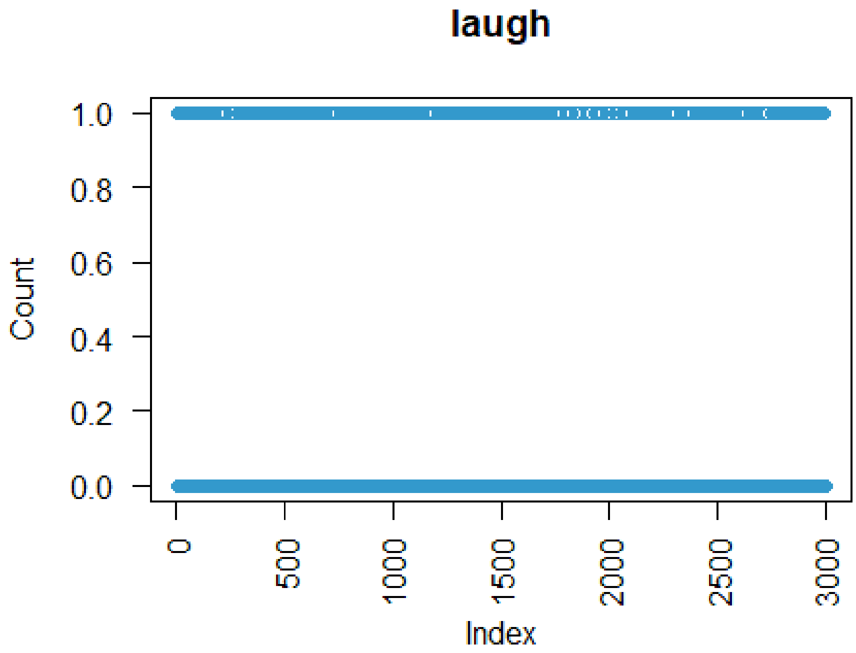
Apply pca and kmean

using MCA for discrete variables

```
df3=df[,c(13,14,15,16,10)]
for (i in 1:5) {
  plot(df3[,i], main=colnames(df3)[i],
       ylab = "Count", col="steelblue", las = 2)
}
```



voyage



laugh

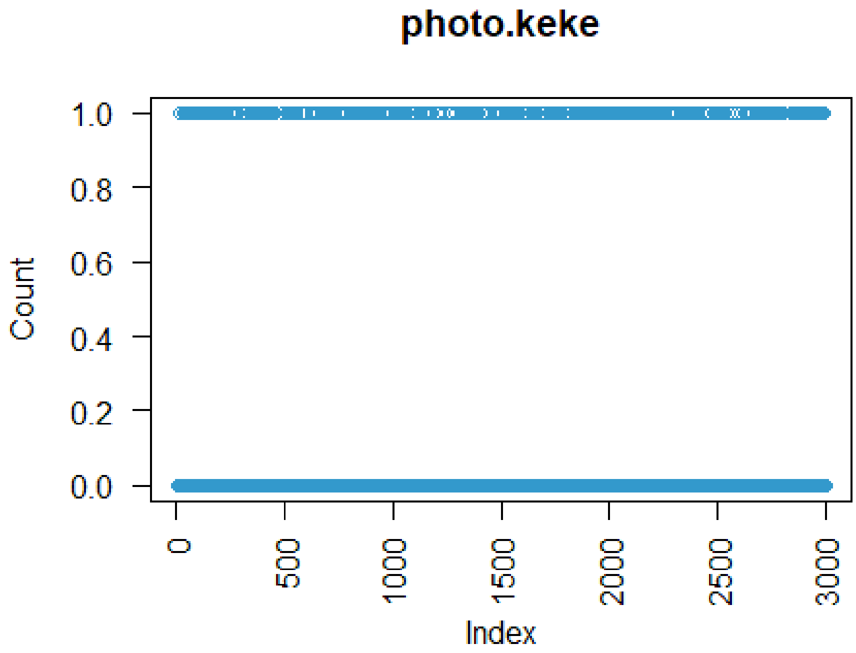


photo.keke
Count
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
Index

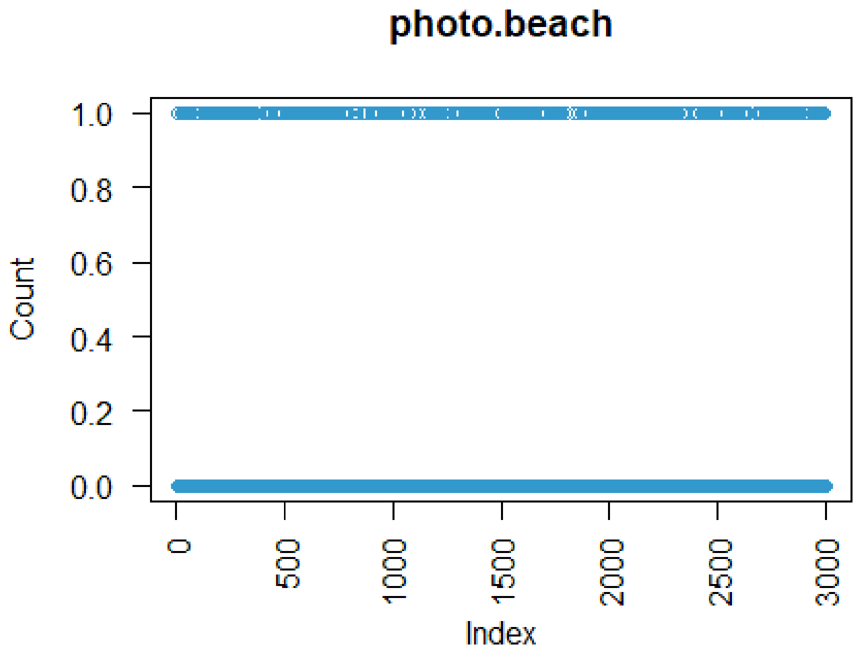
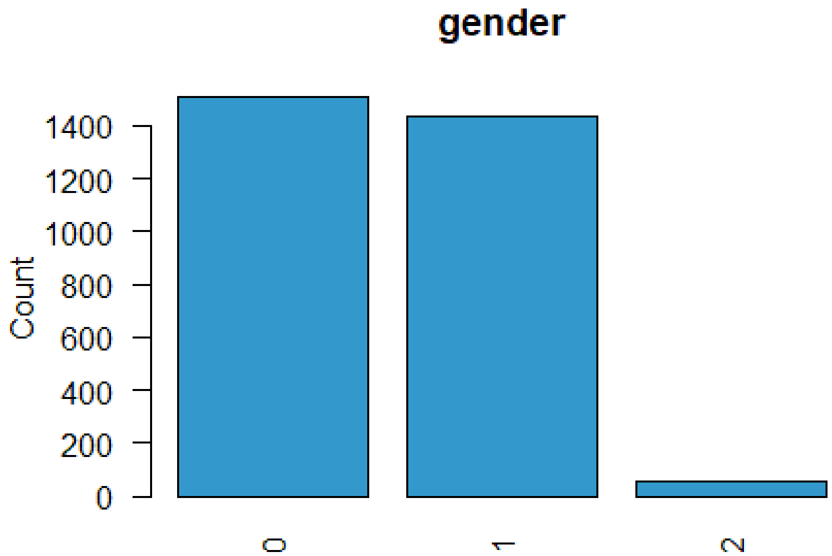
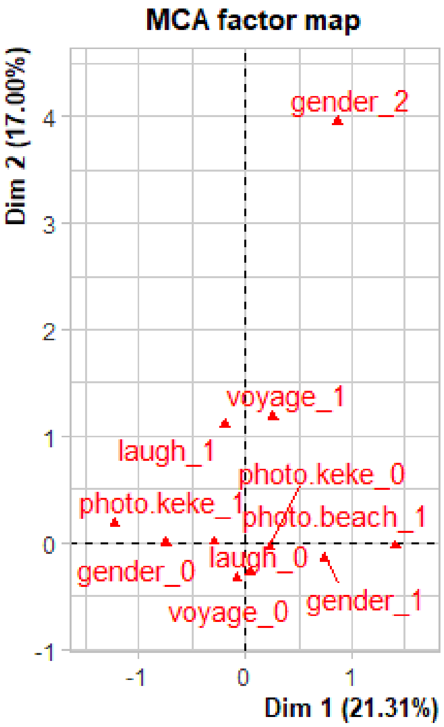


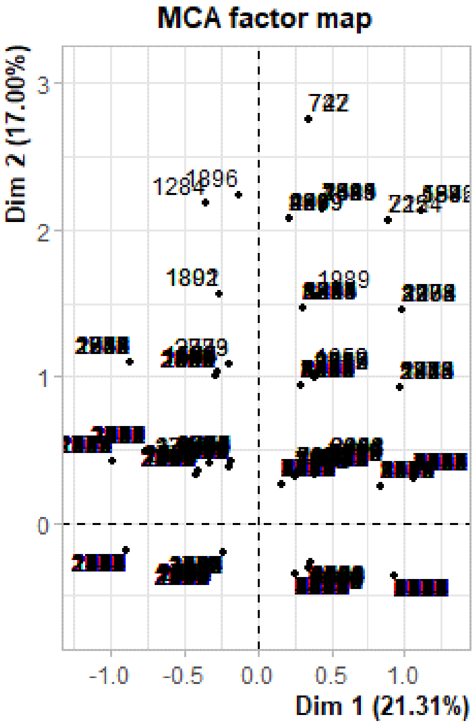
photo.beach
Count
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
Index

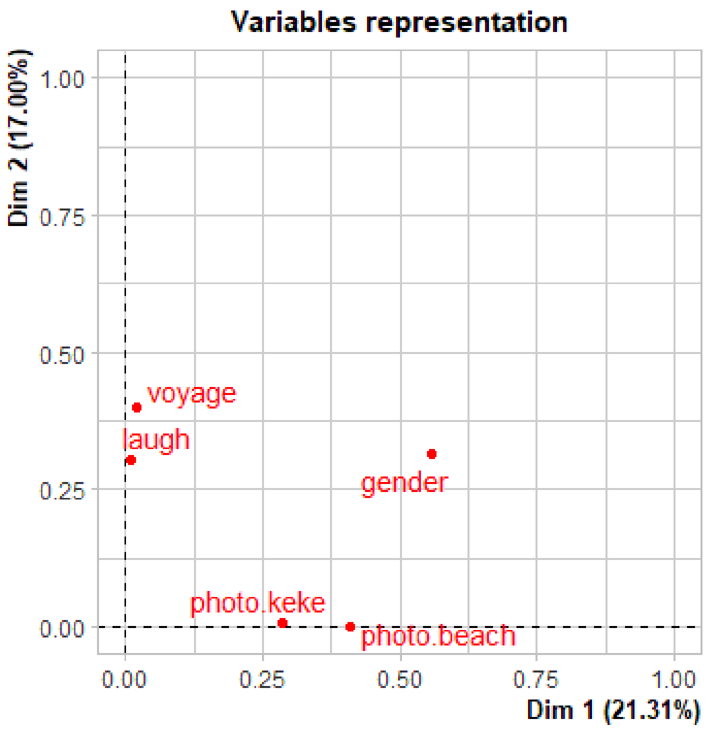


```
df3[sapply(df3, is.numeric)] <- lapply(df3[sapply(df3, is.numeric)],
                                       as.character)
df3[sapply(df3, is.character)] <- lapply(df3[sapply(df3, is.character)],
                                       as.factor)
MCA(df3)
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


MCA factor map
Dim 2 (17.00%)
Dim 1 (21.31%)
4
3
2
1
0
-1
-1
0
1
gender_2
voyage_1
laugh_1
photo.keke_0
photo.keke_1
photo.beach_1
laugh_0
gender_0
gender_1
voyage_0


MCA factor map
Dim 2 (17.00%)
Dim 1 (21.31%)
3
2
1
0
-1.0
-0.5
0.0
0.5
1.0
722
1284
1896
2224
1892
1989
2779

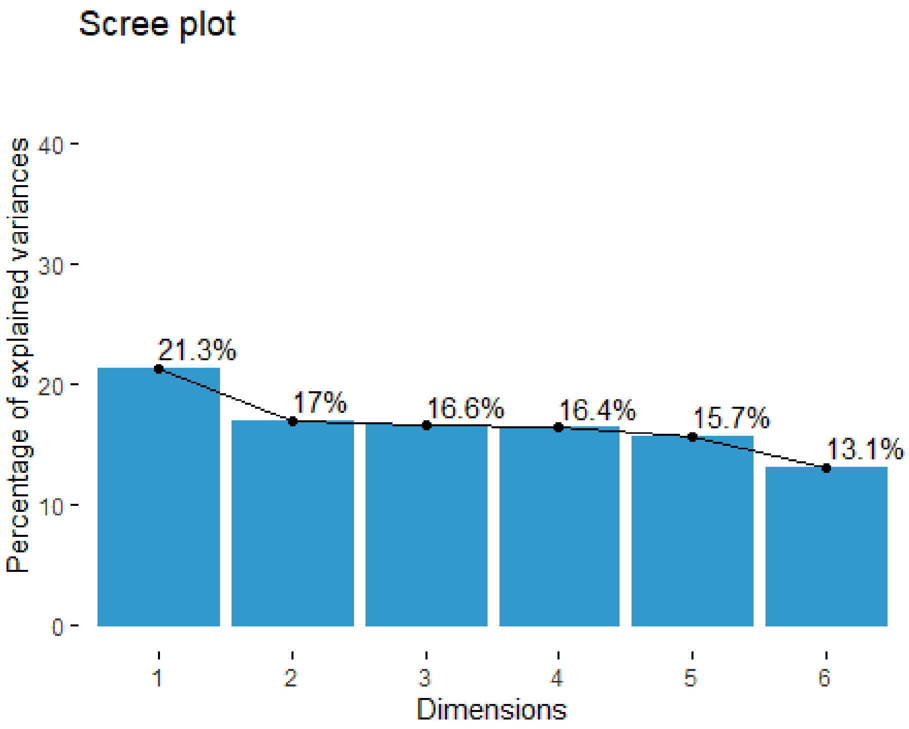


```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 3000 individuals, described by 5 variables
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$var"            "results 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"
```

```
res.mca <- MCA(df3, graph = FALSE)
print(res.mca)
```

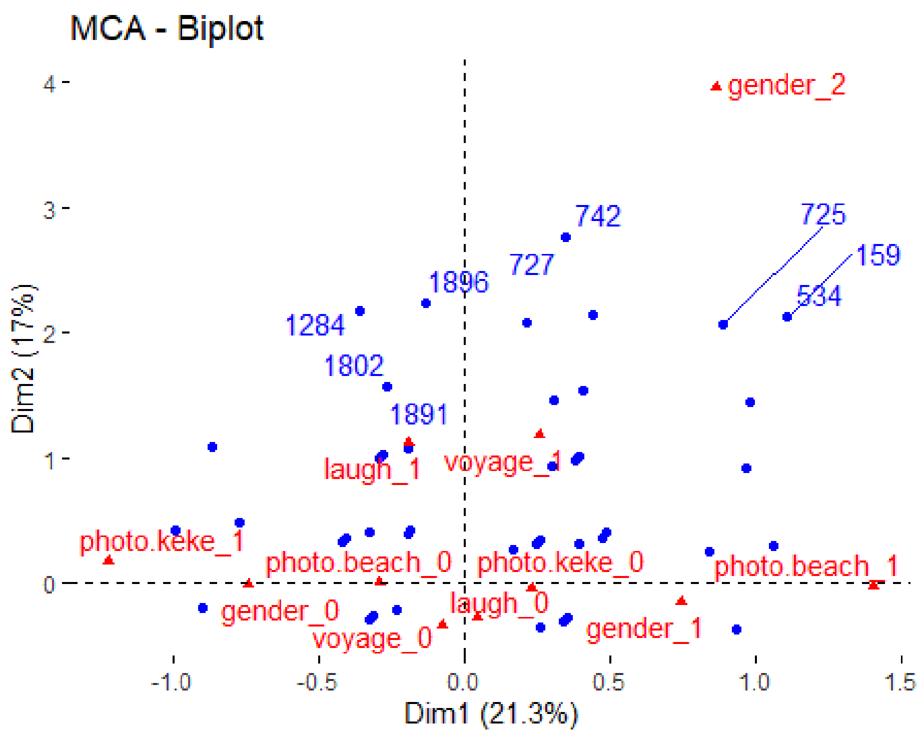
```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 3000 individuals, described by 5 variables
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$var"             "results 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"
```

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

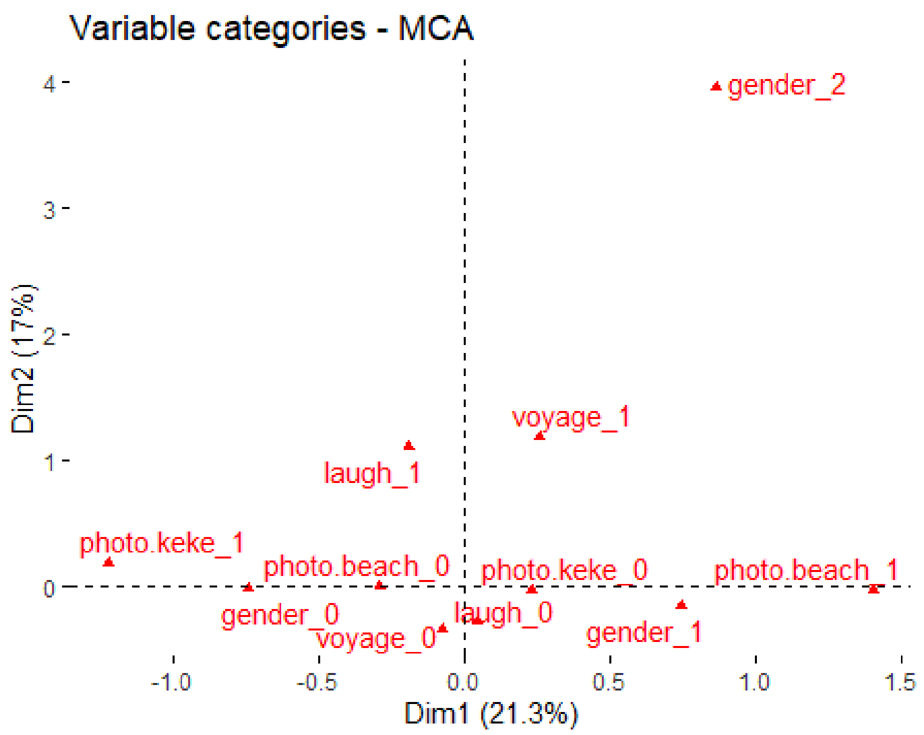


```
fviz_mca_biplot(res.mca,
                repel = TRUE, # Avoid text overlapping (slow if many point)
                ggtheme = theme_minimal())
```

```
## Warning: ggrepel: 2991 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

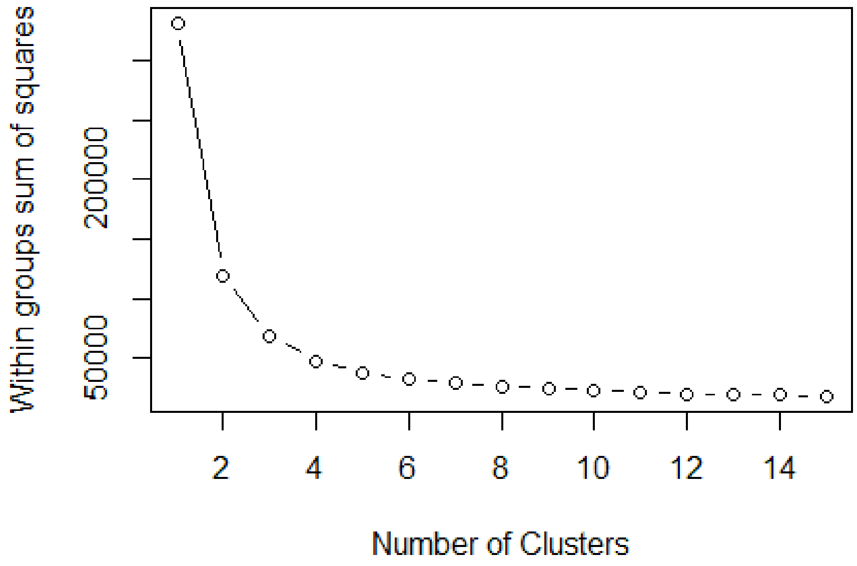


```
#individuals_MCA
fviz_mca_var(res.mca, col.ind = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, # Avoid text overlapping (slow if many points)
             ggtheme = theme_minimal())
```



```
wss <- (nrow(df2)-1)*sum(apply(df2,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(df2,
                                     centers=i)$withinss)
```

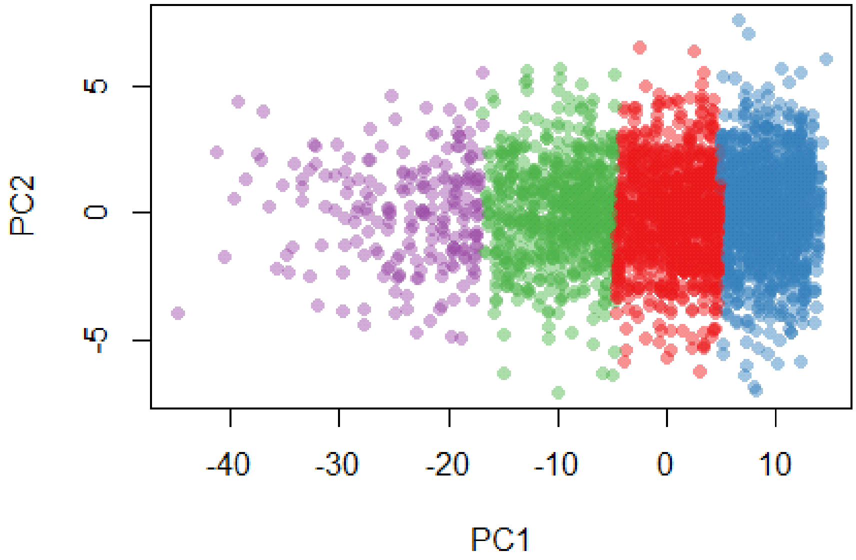
```
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



```
#So here we can see that the "elbow" in the scree plot is at k=4, so we apply
the k-means clustering function with k = 4 and plot.
```

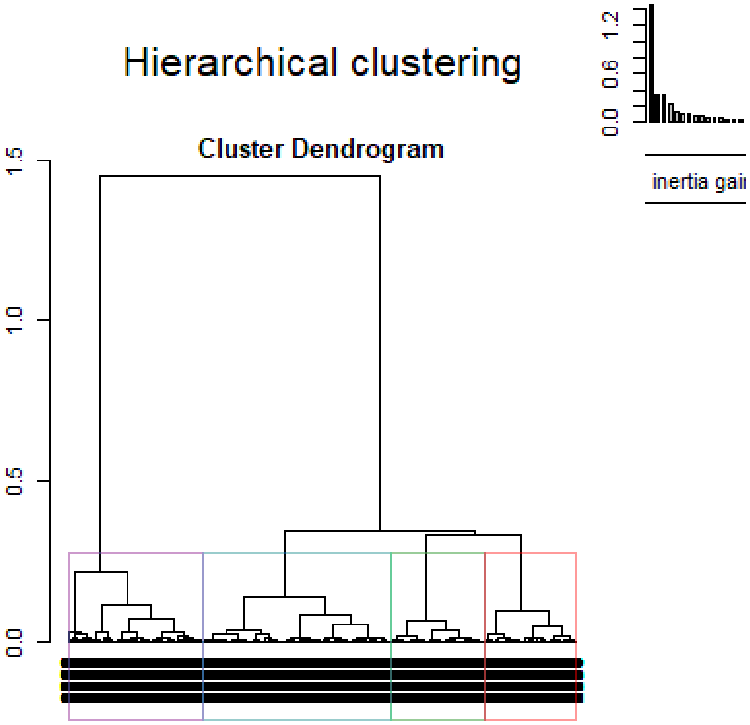
```
# From scree plot elbow occurs at k = 4
# Apply k-means with k=4
pc <- prcomp(df2)
comp <- data.frame(pc$x[,1:2])
k <- kmeans(comp, 4, nstart=25, iter.max=1000)
```

```
palette(alpha(brewer.pal(9,'Set1'), 0.5))
plot(comp, col=k$clust, pch=16)
```

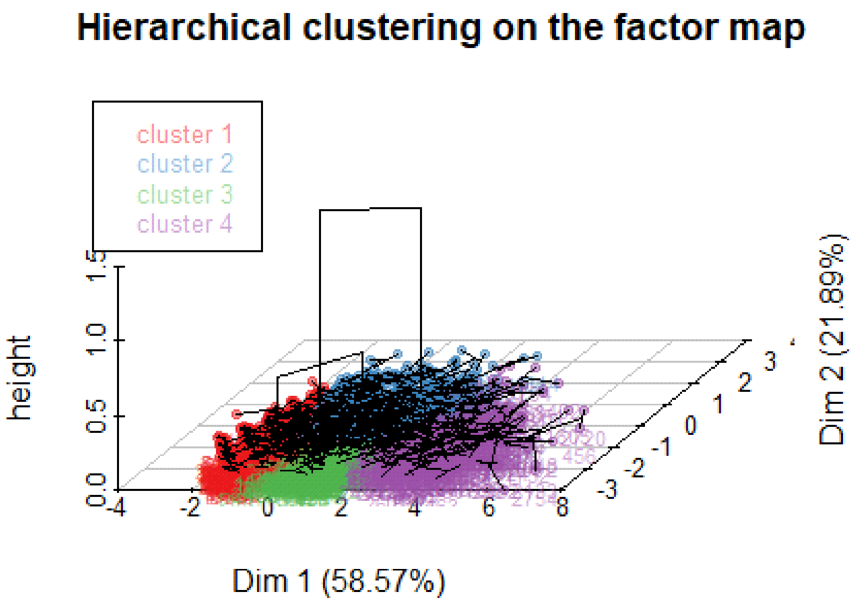


## HCPC

```
res.pca <- PCA(df2, graph = FALSE)
res.hcpc <- HCPC(res.pca, graph = FALSE)
plot(res.hcpc,choice='tree')
```



```
plot(res.hcpc)
```



Hierarchical clustering on the factor map