Chapter 7 MFA

Multiple factor analysis (MFA, also called multiple factorial analysis) is an extension of principal component analysis (PCA) tailored to handle multiple data tables that measure sets of variables collected on the same observations, or, alternatively, (in dual-MFA) multiple data tables where the same variables are measured on different sets of observations. MFA proceeds in two steps:First it computes a PCA of each data table and 'normalizes' each data table by dividing all its elements by the first singular value obtained from its PCA. Second, all the normalized data tables are aggregated into a grand data table that is analyzed via a (non-normalized) PCA that gives a set of factor scores for the observations and loadings for the variables. In addition, MFA provides for each data table a set of partial factor scores for the observations that reflects the specific 'view-point' of this data table. Interestingly, the common factor scores could be obtained by replacing the original normalized data tables by the normalized factor scores obtained from the PCA of each of the set tables.In this article, we present MFA, review recent extensions, and illustrate it with a detailed example

```
rm(list = ls())
graphics.off()

knitr::opts_chunk$set(echo = TRUE)
```

7.1 DATASET

SmartphoneUsage dataset has 2 parts.

we have further divided the second part into 2 parts.

so now we have 3 datasets one is for usage(data3) one is for GDP(data1) and one is for populations (data2)

Rows are countries for all the 3 datasets

NOTE** (We will combine north america and south america as one continent while grouping our countries and we have excluded african continent as it had just one country.)

```
load("C:/Users/jeevan/Desktop/RM2/R-M/SmartphoneUsage (1).RData")
b=data.more

b["C"]=c("Am","E","Am","A","E","E","E","E","E","A","A","E","E","A","A","Am","E","E","A","A'

b=b[c(-22),]


data.pca=data.pca[c(-22),]

data1 <- b[,4:12]

data2 <- b[,14:23]

newdata=cbind(data1,data2,data.pca)

data.design <- c(1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3)


x=colnames(newdata)

data<- newdata

design=data.frame(t(data.design))

colnames(design)=x
```

```r
library(MExPosition)
library(dplyr)
library(tidyverse)
library(TExPosition)
library(TInPosition)
library(prettyGraphs)
library(ExPosition)
library(PTCA4CATA)
library(data4PCCAR)
demo.mfa.2007 <- mpMFA(data,design)
```

```
## [1] "Preprocessed the Rows of the data matrix using:  None"
## [1] "Preprocessed the Columns of the data matrix using:  Center_1Norm"
## [1] "Preprocessed the Tables of the data matrix using:  MFA_Normalization"
## [1] "Preprocessing Completed"
## [1] "Optimizing using:  None"
## [1] "Processing Complete"
```

```r
col4row <- b$C
col4row <- recode(col4row,
                  "Am" = 'red',
                  "E" = 'yellow',
                  "A" = 'blue'
                   )
```

7.2 COR PLOT/HEATMAP

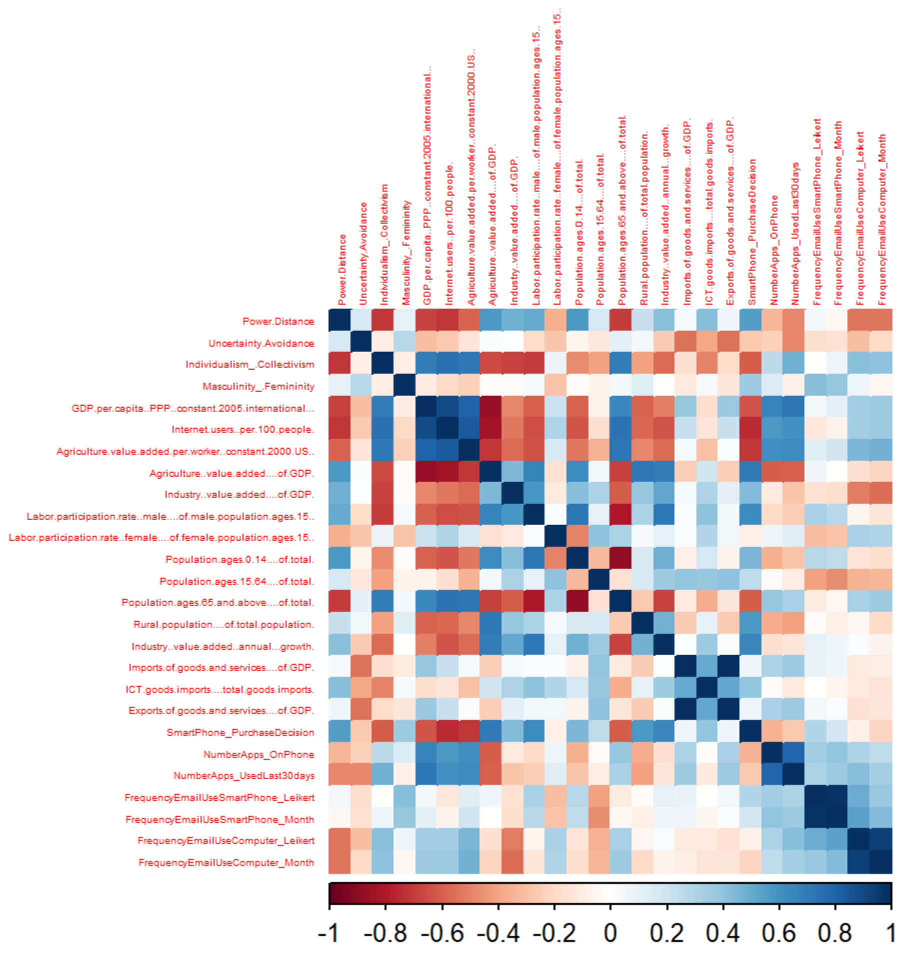
Usage data3 is correlated to data1 more than data2.

Data 2 and data 1 are correlated.

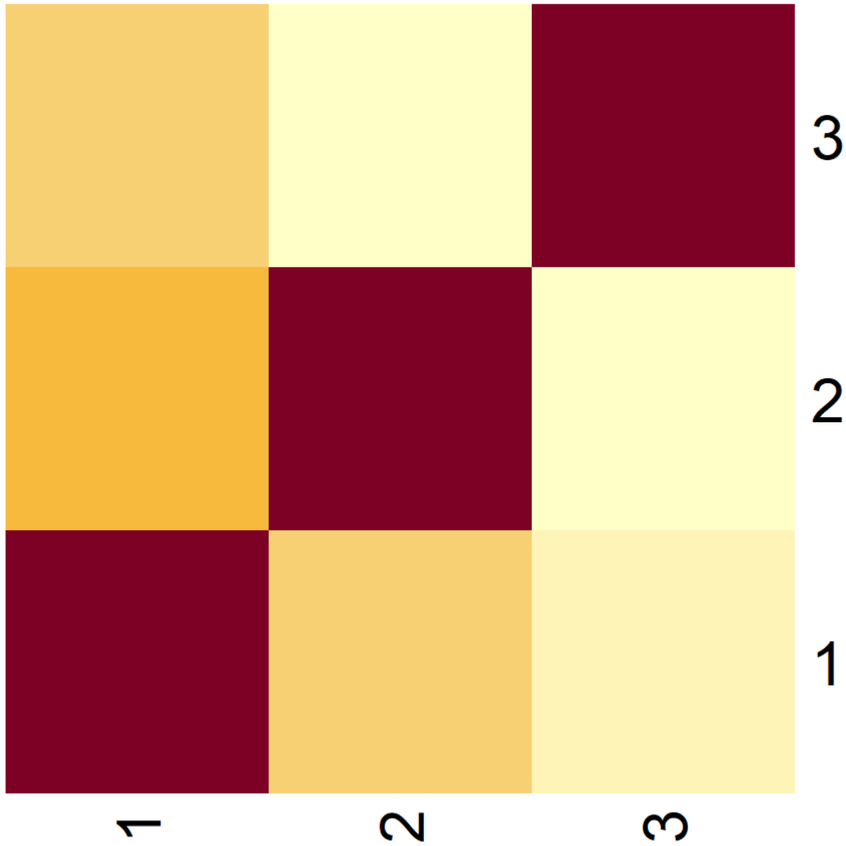
```
rv=demo.mfa.2007$mexPosition.Data$InnerProduct$RVMatrix
```

```
library(corrplot)
XYZ.cor <- cor(data)
```

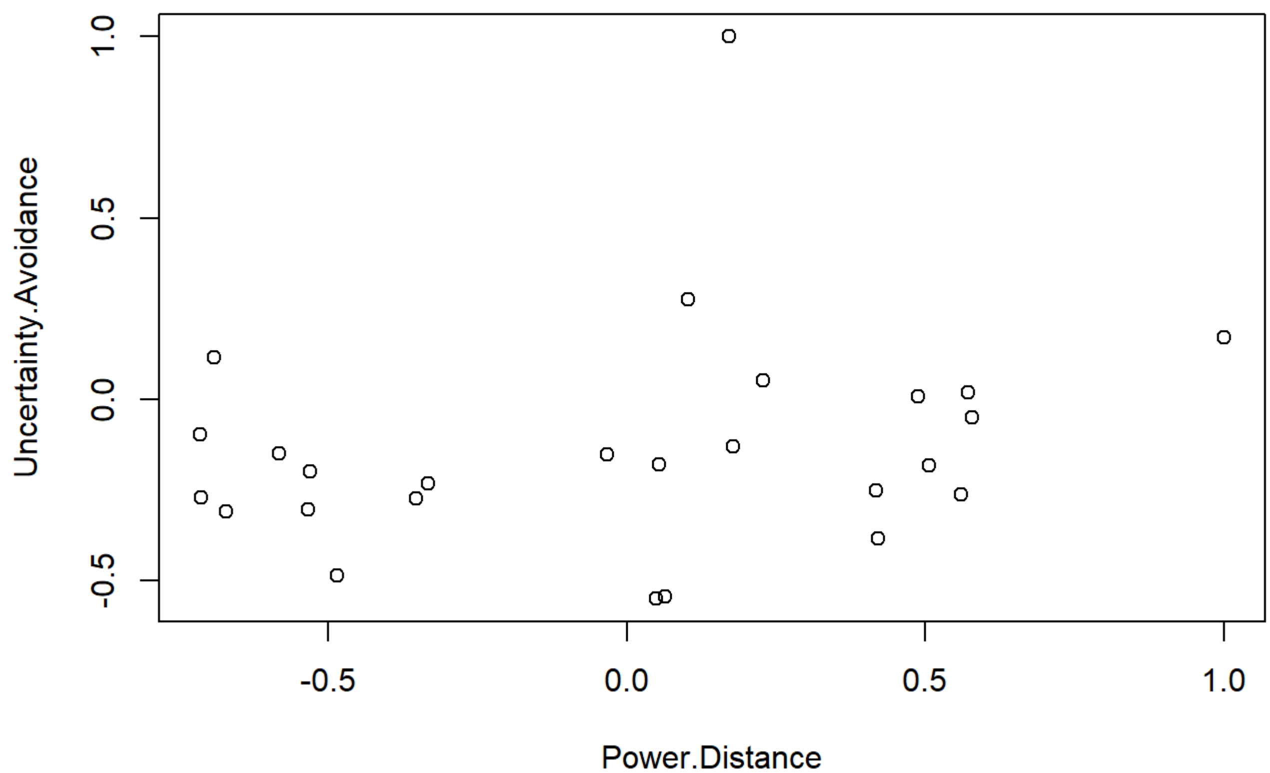
```
corr.plot1<-corrplot(XYZ.cor, method = "color",tl.cex = .3, cl.pos ='b'
         #col = colorRampPalette(c("darkred", "white","midnightblue"))(20)
)
```




```
heatmap(rv,Rowv = NA,Colv = NA)
```




```r
plot(corr.plot1)
```




```
#demo.mfa.2007$mexPosition.Data$Compromise
```

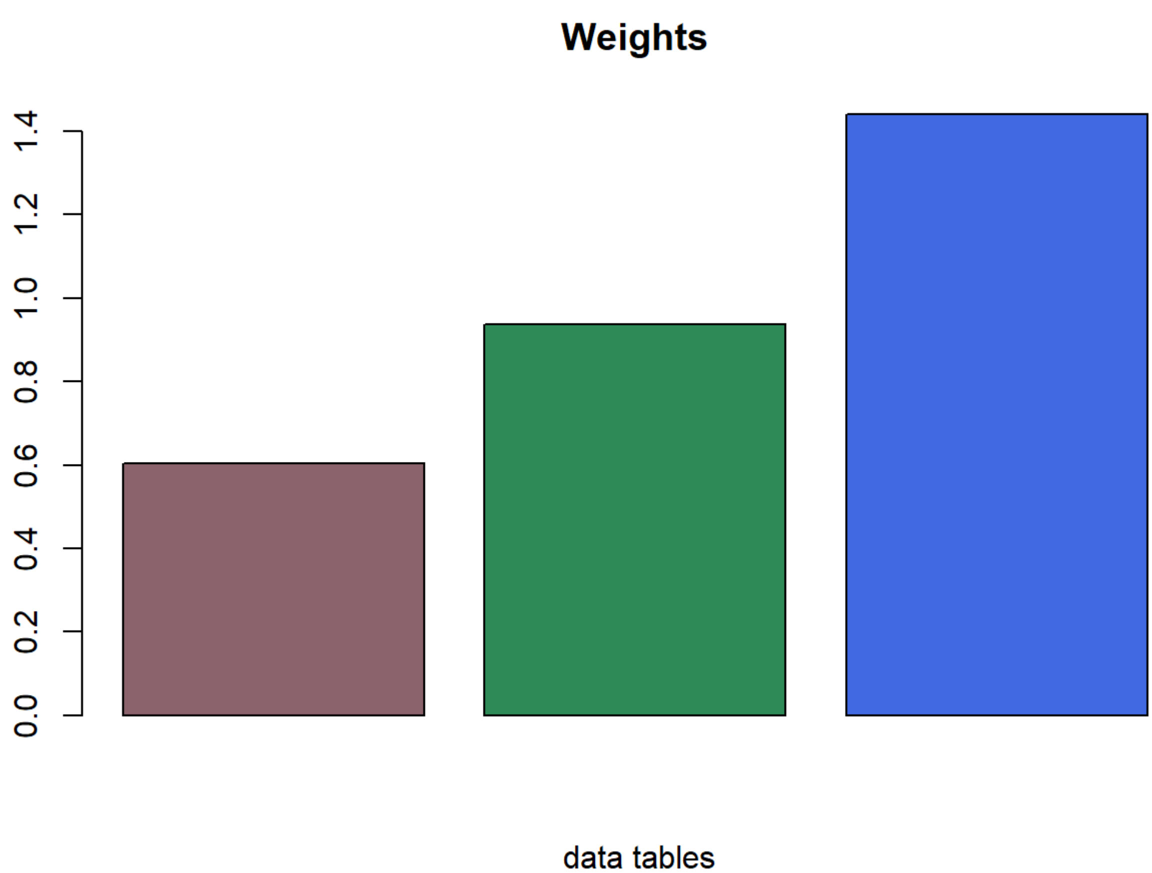
```
#demo.mfa.2007$mexPosition.Data$Table
```

7.3 WEIGHTS

As we don’t want any data to dominate our analysis.

We assign weights to datasets so that we get an equal contribution from all the datasets for our analysis.

```
Eig.tab <- demo.mfa.2007$mexPosition.Data$Compromise$compromise.eigs
Alpha <- 1/sqrt(Eig.tab)
weight <- Alpha

plot.weights <-barplot(weight, main= "Weights",
        col = as.vector(c("pink4","seagreen","royalblue") ),
        xlab= " data tables")
```




```
plot.weights
```

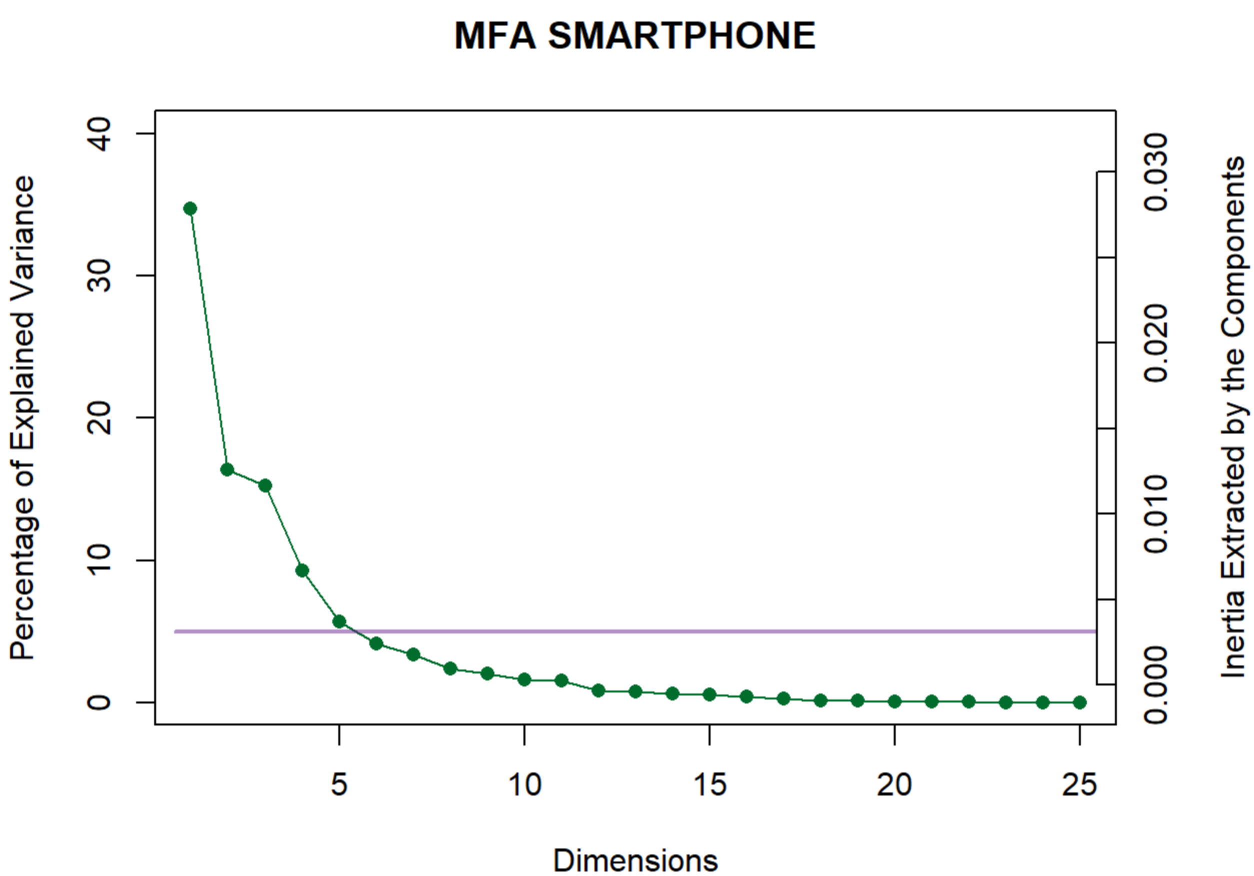
```
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
```

7.4 SCREE PLOT

We have 4 components showing significant varaince.

But just for the convenience of our analysis we will consider first two components.

```
Eig4scree <- demo.mfa.2007$mexPosition.Data$Table$eigs
Tau4scree <- demo.mfa.2007$mexPosition.Data$Compromise$compromise.t
PlotScree(ev=Eig4scree,title = 'MFA SMARTPHONE',plotKaiser = TRUE,color4Kaiser = ggplot2::a
```



##PROJECTIONS

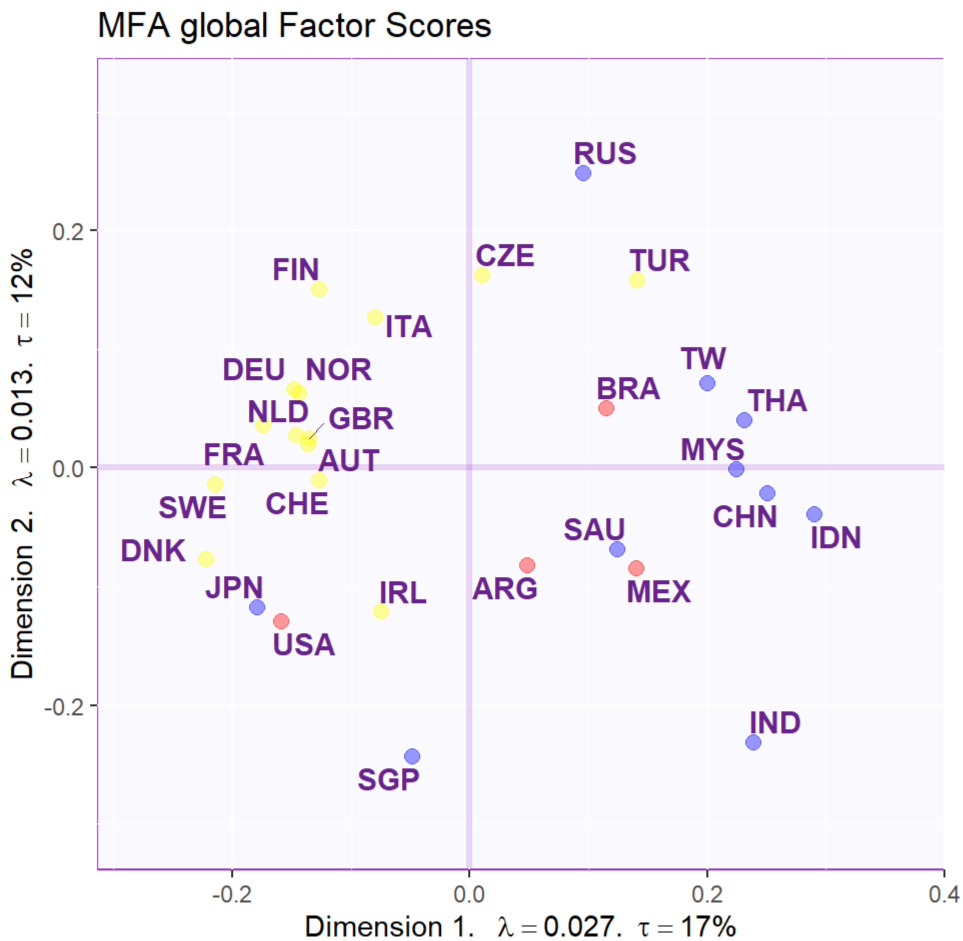
MFA global Factor Scores:

This plot is howing how countries are projected onto the two components and these countries are grouped into 3 groups as per continents with mean of each group marked as continent.

Global Row Factor Scores with Confidence Intervals:

This plot shows the confidence intervals for means

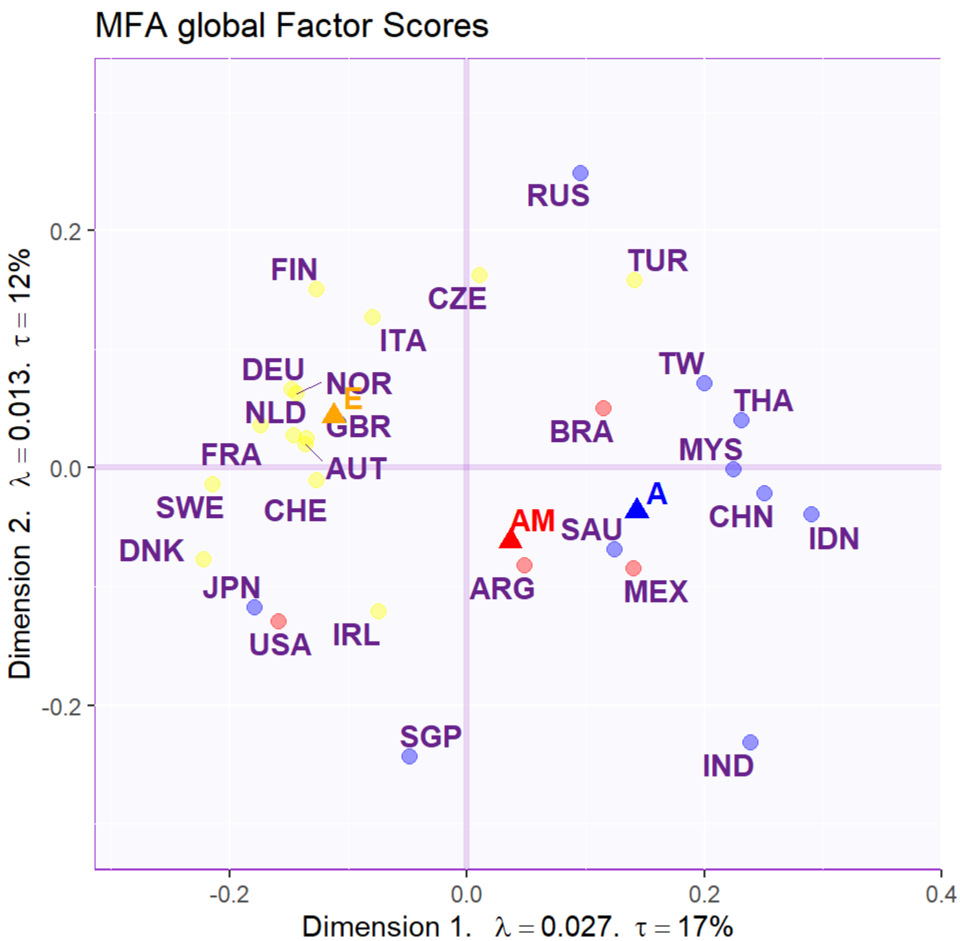
```
#demo.mfa.2007$mexPosition.Data$Table$fi



Fi <-demo.mfa.2007$mexPosition.Data$Table$fi



col4means<- gplots::col2hex(c( 'blue', 'red','orange'))

# Labels for Inertia
label4Map.mfa <- createxyLabels.gen(1,2,
                                    lambda =demo.mfa.2007$mexPosition.Data$Table$eigs,
                                    tau = demo.mfa.2007$mexPosition.Data$Table$t)

#MFA I-set map Dimension(1 and 2)
baseMap.i <- PTCA4CATA::createFactorMap( Fi,
                                         title = "MFA global Factor Scores",
                                         col.points = (col4row),
                                         display.labels = TRUE,
                                         alpha.points = .4
                                         )

aggMap.i <- baseMap.i$zeMap_background + baseMap.i$zeMap_dots + label4Map.mfa+baseMap.i$zeM

aggMap.i
```




```r
#grp.ind <- order(data1_active$memoryGroups)[!duplicated(sort(data1_active$memoryGroups))]
#grp.col <- col4row
#grp.name <- data1_active$memoryGroups[grp.ind] # get the corresponding groups
#names(grp.col) <- grp.name



group.mean <- aggregate(Fi,
                        by = list((b$C)),
                        mean
                        )
rownames(group.mean)=c('A','AM','E')
#rownames(group.mean) <- group.mean[,1] # Use the first column as row names
fi.mean <- group.mean[,-1] # Exclude the first column
fi.mean
```

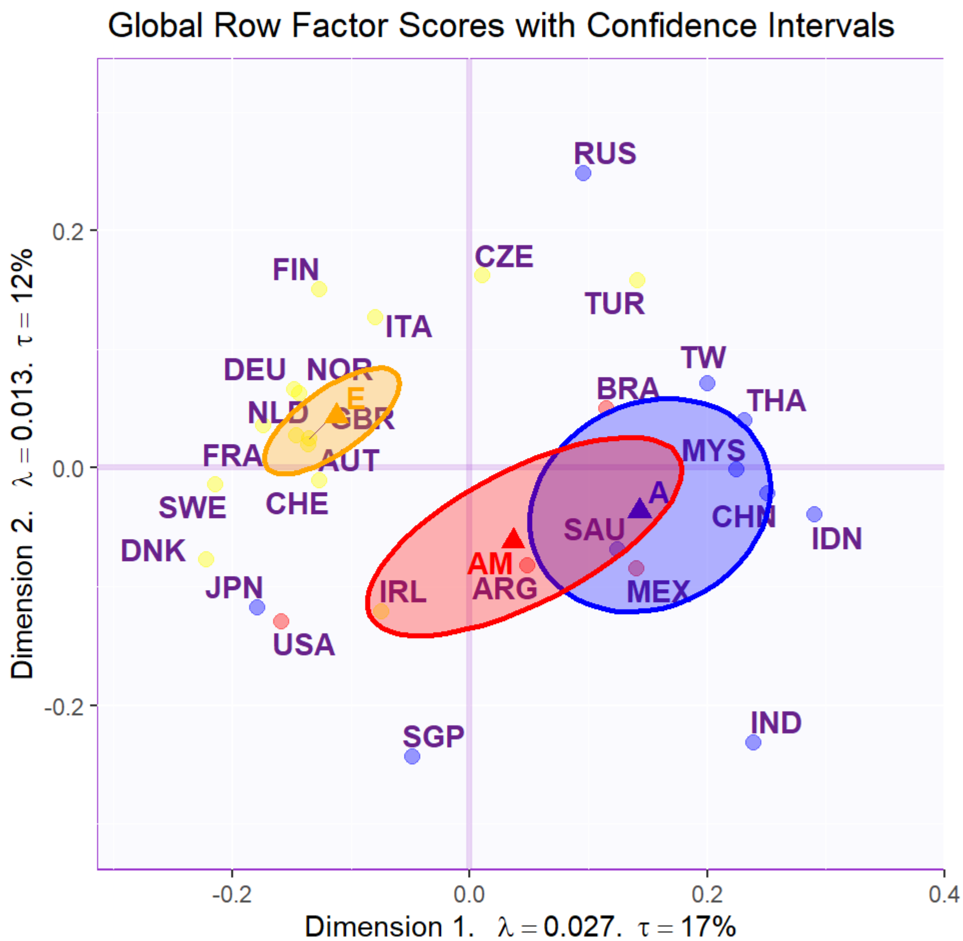
```
##             V1          V2           V3            V4           V5
## A   0.14296259 -0.03639493  0.036115211 -9.862963e-05  0.025656958
## AM  0.03645612 -0.06158369 -0.096895595  6.418108e-03 -0.037587302
## E  -0.11253217  0.04359172  0.001887876 -1.763295e-03 -0.007587169
##               V6           V7            V8           V9          V10
## A   0.0147615813  0.004581035 -0.0115666616  0.015769157  0.002298790
## AM -0.0398348477  0.069072550  0.0294076994 -0.028962108 -0.006532320
## E   0.0008373984 -0.023007182 -0.0001402986 -0.002988796  0.000224384
##              V11          V12          V13          V14          V15
## A  -0.0104355095  0.003814352  0.005183009  0.002150099 -0.003788054
## AM -0.0001883341 -0.005942079 -0.005811317 -0.015525866  0.002426317
## E   0.0075077451 -0.001026800 -0.002041773  0.002900177  0.002012519
##              V16           V17           V18           V19           V20
## A  -0.0017885356  0.0022202779 -0.0003340310  0.0003476294 -0.0008738233
## AM  0.0002972322 -0.0052212947 -0.0004729770 -0.0022692476 -0.0009855975
## E   0.0011926019 -0.0000941143  0.0003737299  0.0004000497  0.0009057588
##              V21           V22           V23           V24           V25
## A   0.0000589198 -0.0013571476 -3.687586e-05 -0.0001760425 -5.471856e-05
## AM  0.0014043421  0.0025239962 -4.895062e-04 -0.0001056741 -7.163063e-05
## E  -0.0004433262  0.0002482493  1.661988e-04  0.0001559372  5.955058e-05
```

```
fi.mean.plot <- createFactorMap(fi.mean,
                                alpha.points = 1,
                                display.labels = TRUE,
                                col.points = col4means,
                                col.labels = col4means,
                                pch = 17,
                                cex = 3,
                                text.cex = 4
                                )

aggMap.i.withMeans <- aggMap.i + label4Map.mfa +fi.mean.plot$zeMap_dots + fi.mean.plot$zeMa
aggMap.i.withMeans
```



MFA global Factor Scores
RUS
TUR
FIN
CZE
ITA
TW
DEU
NOR
E
THA
NLD
GBR
BRA
FRA
MYS
AUT
A
SWE
CHE
CHN
IDN
AM
SAU
DNK
ARG
MEX
JPN
USA
IRL
SGP
IND
0.2
0.0
-0.2
-0.2
0.0
0.2
0.4
Dimension 1. λ = 0.027. τ = 17%
Dimension 2. λ = 0.013. τ = 12%

```r
# MFA Confidence Intervals - Bootstrap for CI

BootCube <- PTCA4CATA::Boot4Mean(Fi,
                                 design = (b$C),
                                 niter = 100,
                                 suppressProgressBar = TRUE)



# Create the ellipses using function MakeCIEllipses from package PTCA4CATA
GraphElli <- PTCA4CATA::MakeCIEllipses(BootCube$BootCube[,1:2,],
                                       names.of.factors = c("Dimension 1","Dimension 2"),
                                       col = col4means,
                                       p.level = .95
)

# create the I-map with Observations, means and confidence intervals
Map.I.withCI <-  aggMap.i.withMeans +  GraphElli + label4Map.mfa + ggtitle(" Global Row Fac

Map.I.withCI
```




```
Map.I.withCI <- recordPlot()
```

```
aggMap.i.withMeans<- recordPlot()
```

Means of our groups as per each dataset.

The plot below shows how far are our continents from each other when we consider each dataset i.e data 1 data 2 and data 3

```r
#demo.mfa.2007$mexPosition.Data$Table$partial.fi.array
col.design.col<-c(rep("red",9),rep("blue",10),rep("green",7))

design.col<-design

F_j<- demo.mfa.2007$mexPosition.Data$Table$partial.fi.array

demo.mfa.2007$mexPosition.Data$InnerProduct$alphaWeights
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.3333333 0.3333333 0.3333333
```

```
##            [,1]      [,2]      [,3]
## [1,] 0.3333333 0.3333333 0.3333333
Eig.tab <- demo.mfa.2007$mexPosition.Data$Compromise$compromise.eigs
alpha_j <- 1/sqrt(Eig.tab)

data_tables<- as.numeric(design.col)
code4Groups<- unique(data_tables)
nK<- length(code4Groups)



F_k <- array(0, dim = c(dim(F_j)[[1]], dim(F_j)[[2]],nK))
dimnames(F_k) <- list(dimnames(F_j)[[1]], dimnames(F_j)[[2]], code4Groups)

alpha_k <- rep(0, nK)
names(alpha_k) <- code4Groups
Fa_j <- F_j

# A horrible loop
for (j in 1:dim(F_j)[[3]]){ Fa_j[,,j] <- F_j[,,j] * alpha_j[j] }

# Another horrible loop
for (k in 1:nK){
  #lindex <- data_tables == code4Groups[k]
  alpha_k[k] <- alpha_j[k]
  F_k[,,k] <- (1/alpha_k[k])*apply(Fa_j[,,k],c(1,2),sum)

}




#group.mean <- apply(aggregate(F_k,
#                       by = list(design.row),
#                      # must be a list
#                       mean
#                       ))
```

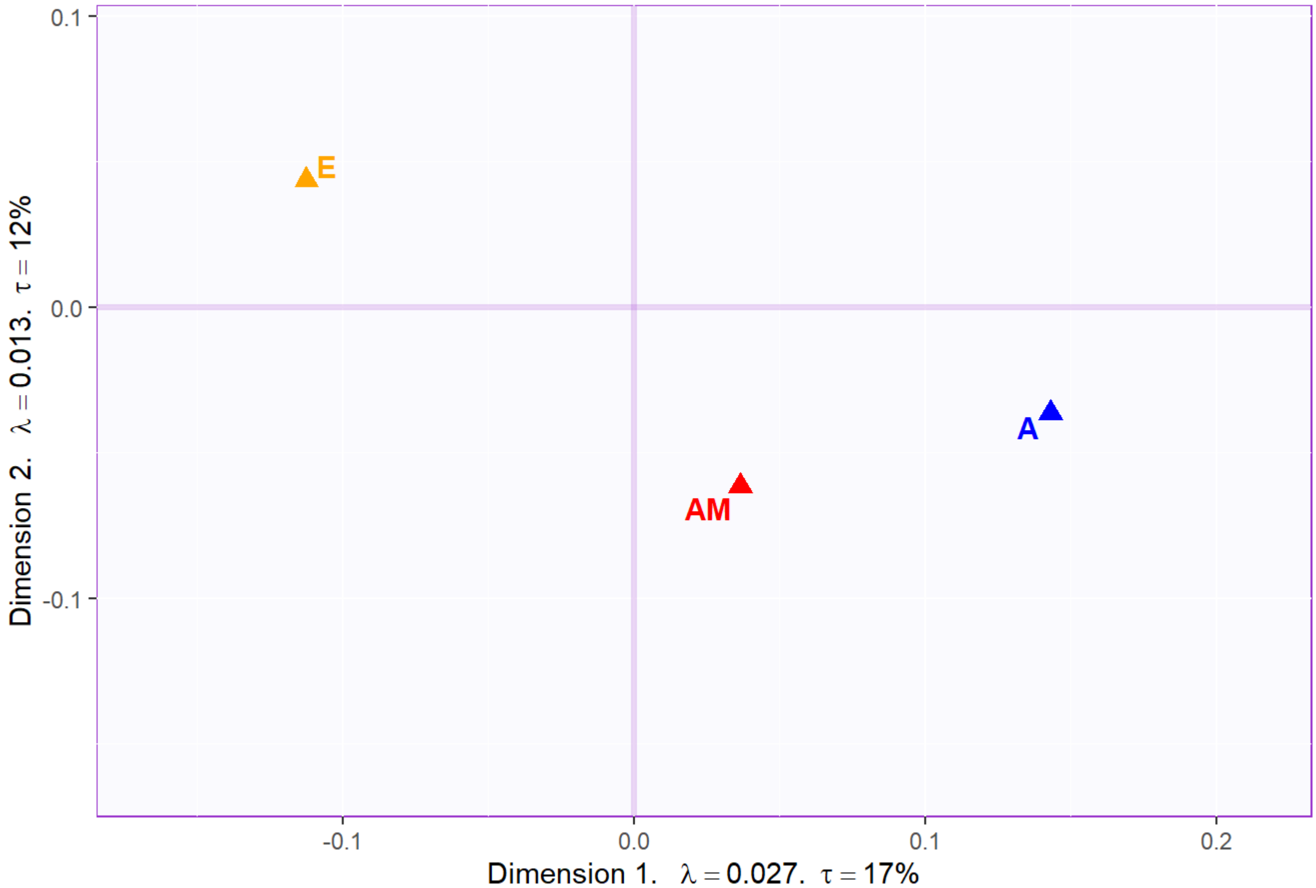
```r
meanfk <-
  apply(F_k, c(2,3), FUN = function(x){
  aggregate(x, by = list(as.factor(b$C)), mean)$x
  })
dim(meanfk)
```

```
## [1]  3 25  3
```

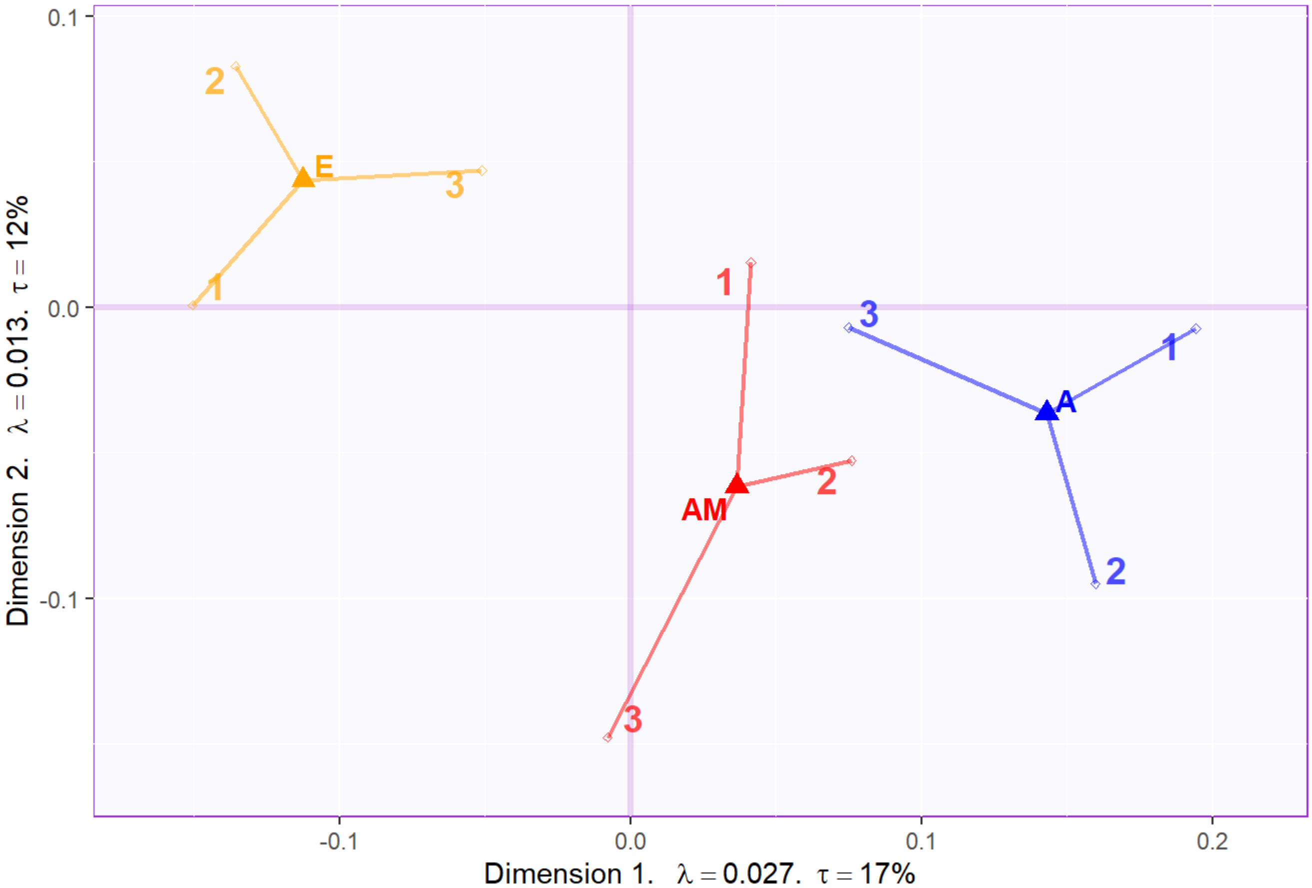
```
## [1]  3 31  3
```

```r
group.mean <-meanfk
```

```r
 mean.plot <- createFactorMap(fi.mean,
                                constraints = minmaxHelper4Partial(fi.mean, meanfk, axis1 =
                              alpha.points = 1,
                                display.labels = TRUE,
                                col.points = col4means,
                                col.labels = col4means,
                                pch = 17,
                                cex = 3,
                                text.cex = 4
                                )

Fi.meanonly.plot<- mean.plot$zeMap_background+mean.plot$zeMap_dots + mean.plot$zeMap_text+

Fi.meanonly.plot
```




```
pf.means <- createPartialFactorScoresMap(
                                        factorScores = fi.mean,
                                        partialFactorScores = meanfk,
                                        axis1 = 1, axis2 = 2,
                                        colors4Items = as.vector(col4means),
#colors4Blocks = c("lightblue", "skyblue","midnightblue"),
                                        names4Partial = dimnames(meanfk)[[3]], #
                                        font.labels = 'bold',
                                        size.labels = 5,
)

plot.pFi.mean <- Fi.meanonly.plot + label4Map.mfa + pf.means$mapColByItems

plot.pFi.mean
```



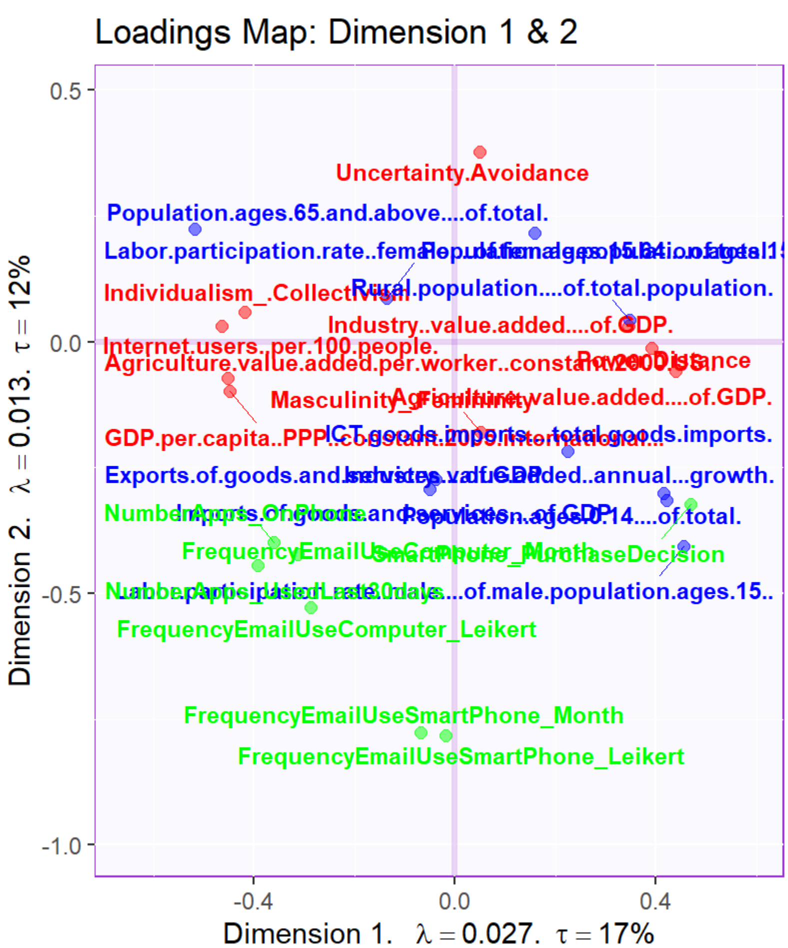
7.5 LOADINGS

Loadings will help us understand how much does each variable from our combined dataset contribute towards the components 1 and 2.

Also we could get the relations between our variables.

```
Fi<- demo.mfa.2007$mexPosition.Data$Table$fi
pFi <- F_k

colnames(Fi)<- c(paste0('Dimension ',1:25))
colnames(pFi)<-c(paste0('Dimension ',1:25))


map4PFS <- createPartialFactorScoresMap(
                                        factorScores =Fi[,1:2],
                                        partialFactorScores = pFi[,1:2,],
                                        axis1 = 1, axis2 = 2,
                                        #colors4Items = as.vector(col4row),
#colors4Blocks = c("lightblue", "skyblue","midnightblue"),

                                        font.labels = 'bold',
                                        size.labels = 5)
#plot.pFi1 <- baseMap.i$zeMap + label4Map.mfa+
#map4PFS$mapColByItems

#plot.pFi1

#plot.pFi1 <- recordPlot()
#plot.pFi2 <- baseMap.i$zeMap + label4Map.mfa+
#map4PFS$mapColByBlocks

#plot.pFi2

#plot.pFi2 <- recordPlot()
Q <- demo.mfa.2007$mexPosition.Data$Table$Q


label4Map.mfa <- createxyLabels.gen(1,2,
                                    lambda = demo.mfa.2007$mexPosition.Data$Table$eigs,
                                    tau = demo.mfa.2007$mexPosition.Data$Table$t)
```

```r
baseMap.j <- createFactorMap(Q, #constraints = constraints.sym,
                             col.points = col.design.col,
                             col.labels = col.design.col,
                             display.labels = TRUE,
                             display.points = TRUE,
                             text.cex = 3,
                             force=2,
                             cex =2,
                             title = "Loadings Map: Dimension 1 & 2")


#lines4J <- addLines4MCA(Fj, col4Var = col4Levels.imp$color4Variables, size = 1)

Loadings_12 <- baseMap.j$zeMap+ label4Map.mfa

Loadings_12
```



```
Loadings_12<- recordPlot()



baseMap.j.2 <- PTCA4CATA::createFactorMap(Q, axis1 = 3, axis2 = 2,
                                          col.points   =  gplots::col2hex(col.design.col),
                                          alpha.points =  .8,
                                          alpha.labels = .8,
                                          col.labels   =  gplots::col2hex(col.design.col),
                                          force = 5,
                                          cex = 2,
                                          text.cex = 3,
                                          title = "Loadings Map: Dimension 2 and 3")

# A graph for the J-set 2 and 3
Loadings_23 <- baseMap.j.2$zeMap + label4Map.mfa

#Loadings_23



#Loadings_23<- recordPlot()
```

7.6 ANALYSIS/HYPOTHESIS

INTERPRETATIONS

GDP factors and population factors and Purchase decision have shown significant contributions for dim 1.

Variables from data3 have shown significant contributions for dim 2.

So European countries are different from other continents in terms of GDP factors and population factors and Purchase decision. American and asian countries are same according to dim 1

USA is again closer to European countries according to dim 1

European countries are different from other continents in terms of Smartphones usage also. American and asian countries are same according to dim 2

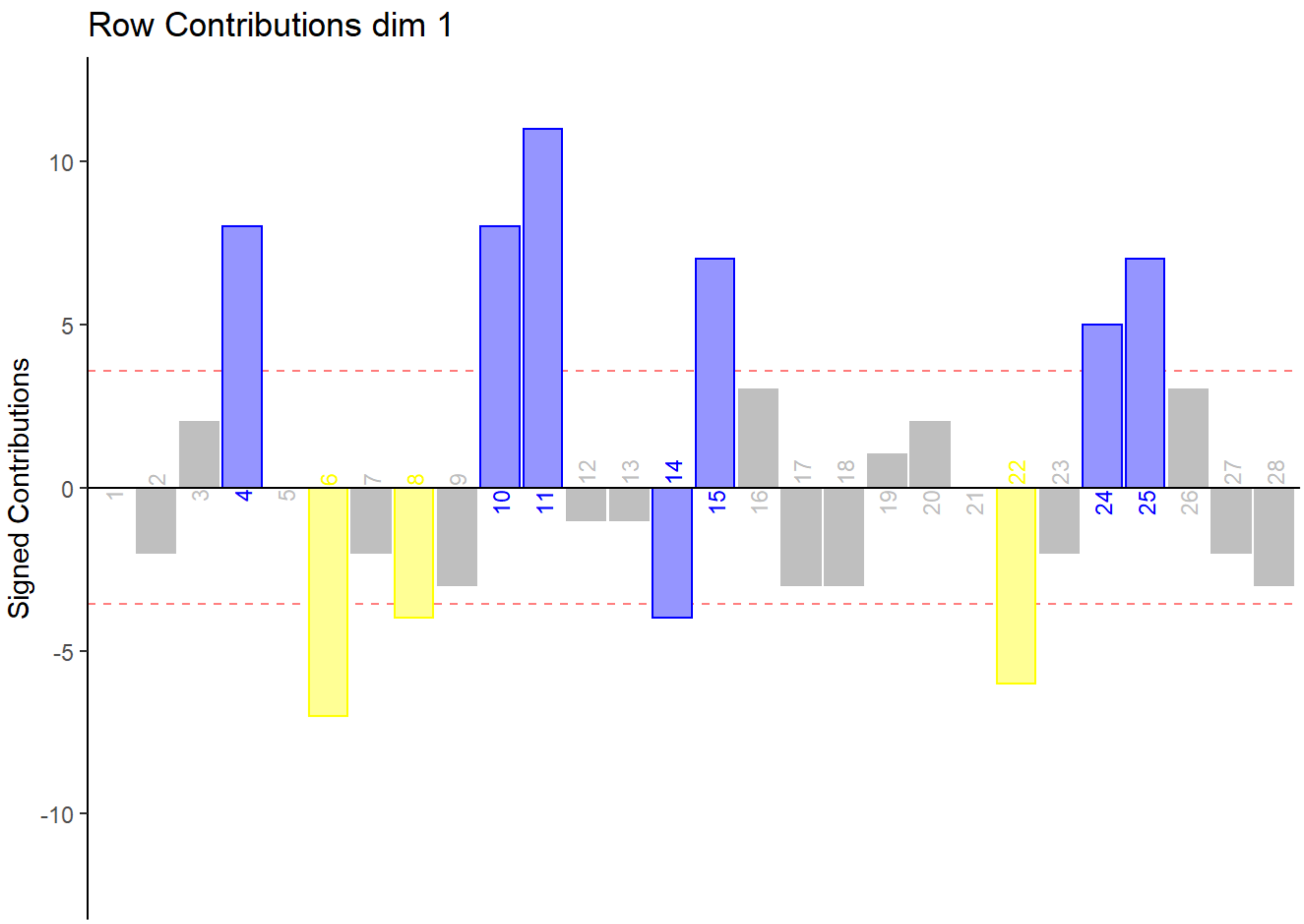
USA is again closer to European countries according to dim 2

Purchase Decsion is mainly related to GDP and population factors

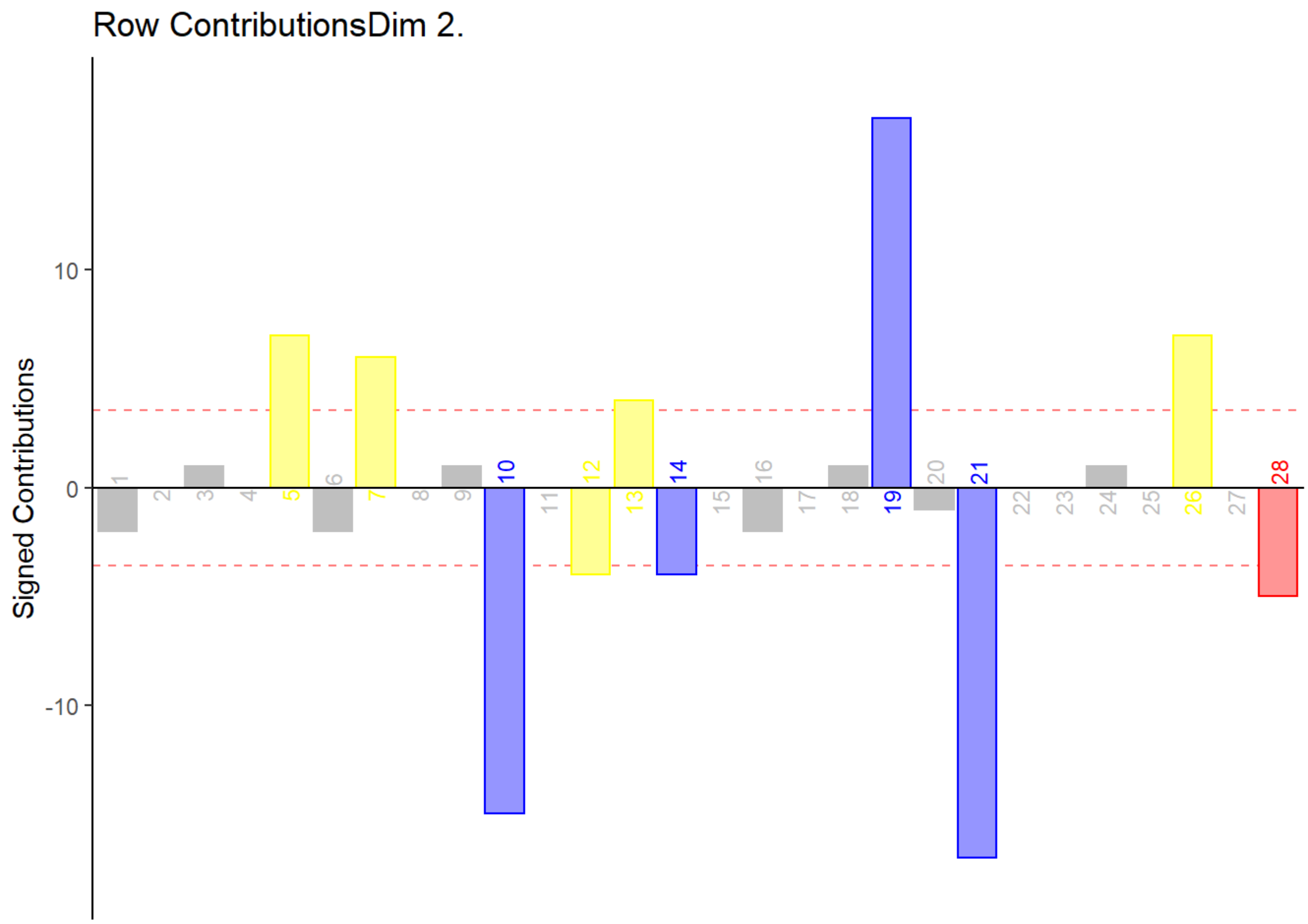
7.7 BOOTSTRAP

According to the test reject the null hypothesis.

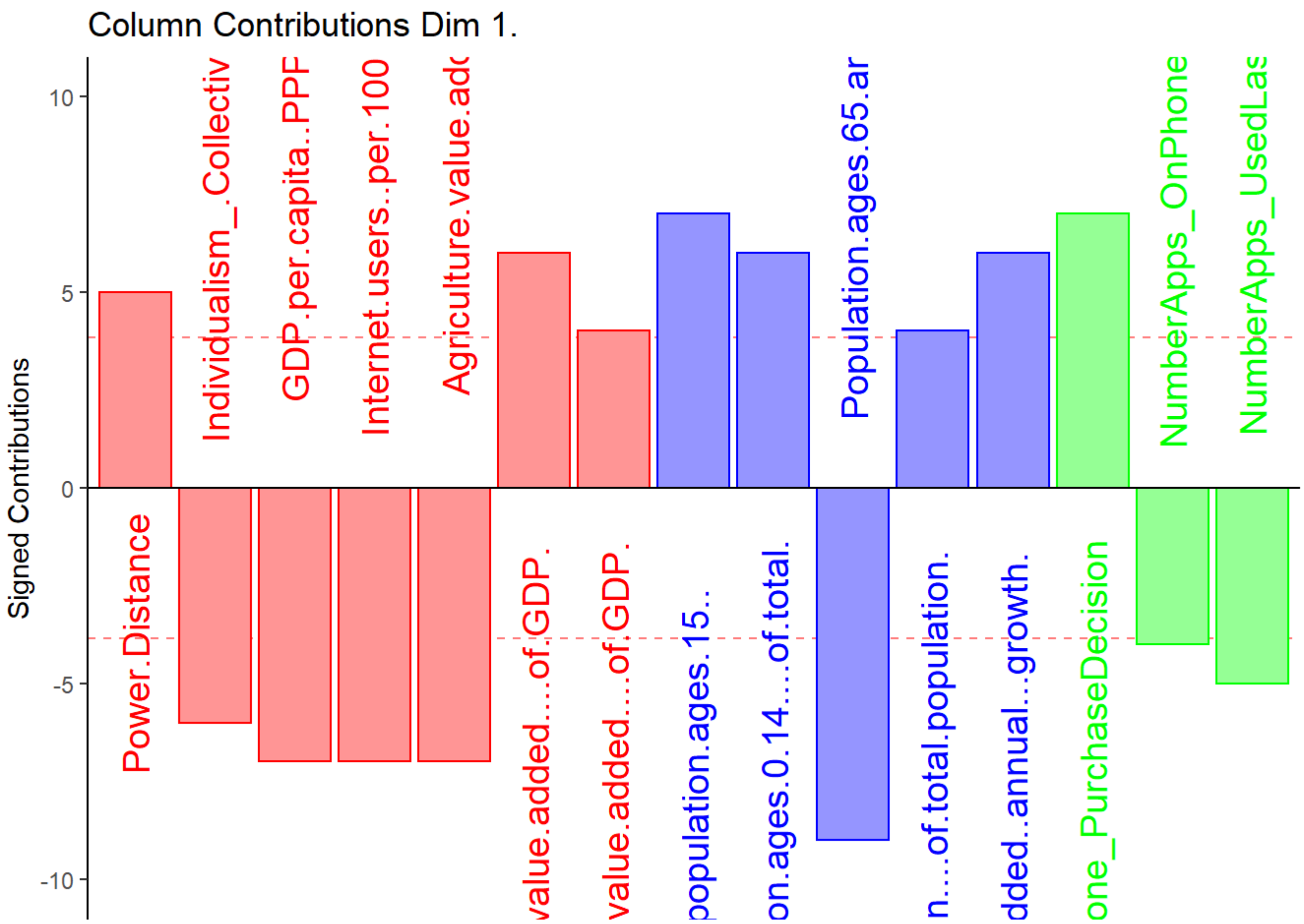
```
ctrI<- demo.mfa.2007$mexPosition.Data$Table$ci

signed.ctrI <- as.matrix( ctrI * sign(Fi))

colnames(signed.ctrI)<- c(paste0('Dimension ',1:25))
rownames(signed.ctrI)<-c(paste0('', 1:28))



plotctrI.1 <- PrettyBarPlot2(bootratio = round(100*signed.ctrI[,1]),
                      threshold = 100 / nrow(signed.ctrI),
                      ylim = NULL,
                      color4bar = gplots::col2hex(as.matrix(col4row)),
                      color4ns = "gray75",
                      plotnames = TRUE,
                      main = 'Row Contributions dim 1',
                      ylab = "Signed Contributions")

plotctrI.1
```



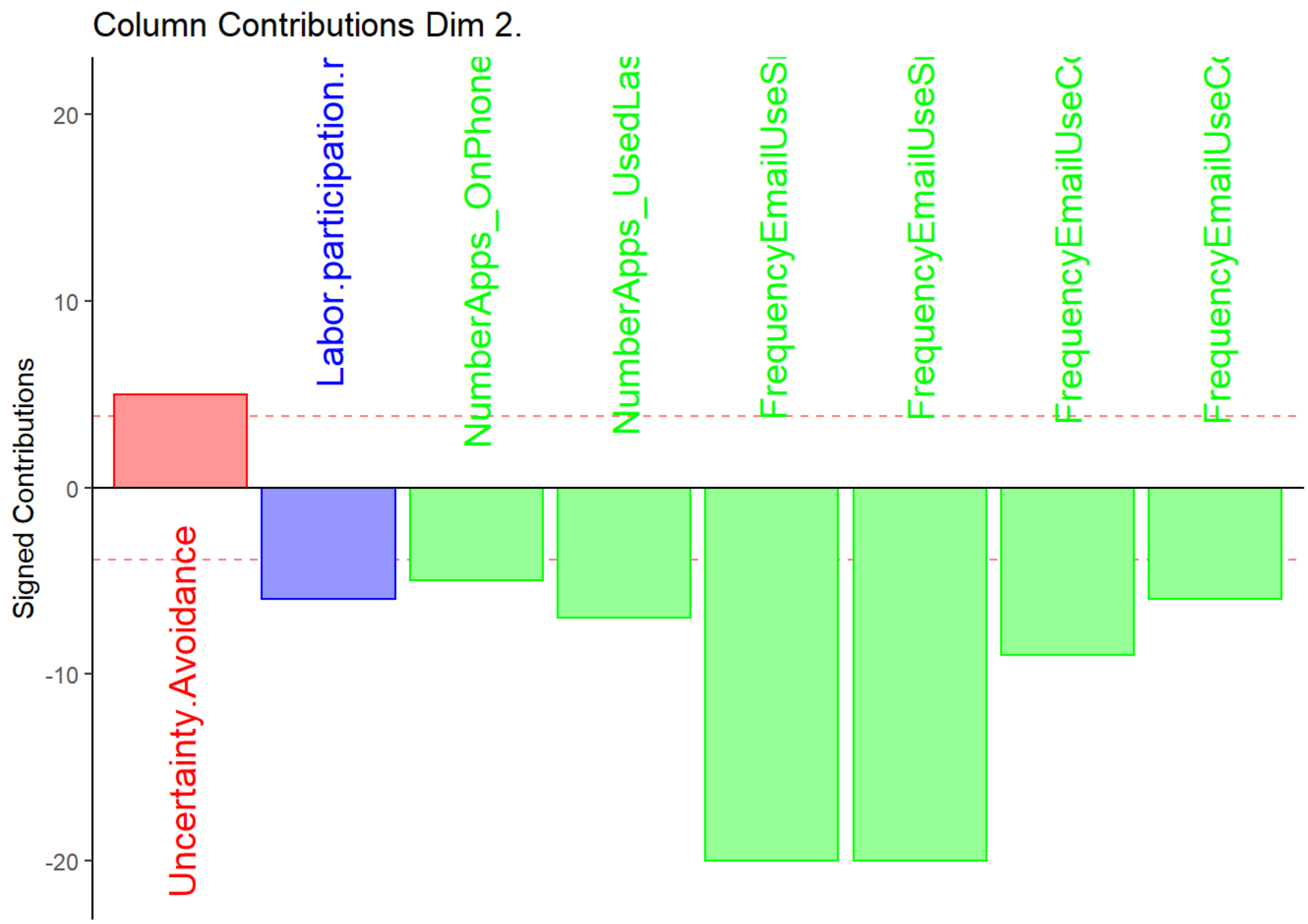

```
plotctrI.2 <- PrettyBarPlot2(
                   bootratio = round(100*signed.ctrI[,2]),
                   threshold = 100 / nrow(signed.ctrI),
                   ylim = NULL,
                   color4bar = gplots::col2hex(as.matrix(col4row)),
                   color4ns = "gray75",
                   plotnames = TRUE,
                   main = 'Row ContributionsDim 2.',
                   ylab = "Signed Contributions")
plotctrI.2
```


Row ContributionsDim 2.
Signed Contributions
10
0
-10
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```r
Fj<- demo.mfa.2007$mexPosition.Data$Table$Q
ctrJ<-demo.mfa.2007$mexPosition.Data$Table$cj
signed.ctrJ <- ctrJ * sign(Fj)

plotctrJ.1 <- PrettyBarPlot2(
                       bootratio = round(100*signed.ctrJ[,1]),
                       threshold = 100 / nrow(signed.ctrJ),
                       ylim = NULL,
                       color4bar = gplots::col2hex(col.design.col),
                       color4ns = "gray75",
                       plotnames = TRUE,
                       main = 'Column Contributions Dim 1.',
                       ylab = "Signed Contributions",
                       signifOnly = TRUE,
                       font.size = 5)

plotctrJ.1
```




```
plotctrJ.2 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctrJ[,2]),
                        threshold = 100 / nrow(signed.ctrJ),
                        ylim = NULL,
                        color4bar = gplots::col2hex(col.design.col),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Column Contributions Dim 2.',
                        ylab = "Signed Contributions",
                        signifOnly = TRUE,
                        font.size = 5)


plotctrJ.2
```




```r
plotctrJ.3 <- PrettyBarPlot2(
                       bootratio = round(100*signed.ctrJ[,3]),
                       threshold = 100 / nrow(signed.ctrJ),
                       ylim = NULL,
                       color4bar = gplots::col2hex(col.design.col),
                       color4ns = "gray75",
                       plotnames = TRUE,
                       main = 'Column Contributions Dim 3.',
                       ylab = "Signed Contributions",
                      signifOnly = TRUE,
                      font.size = 5)

#plotctrJ.3
```

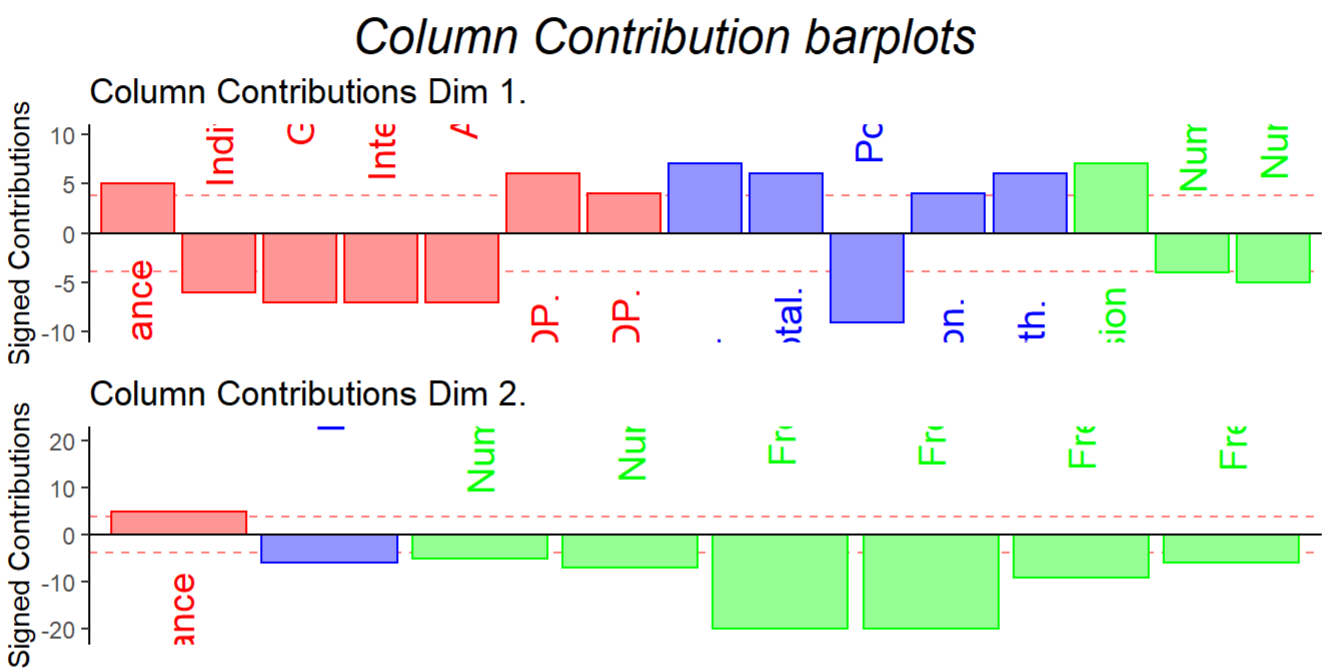
Important Factor Scores

```
#demo.mfa.2007$mexPosition.Data$Table$Q
```

```r
ctrK <- demo.mfa.2007$mexPosition.Data$Table$Q



var12 <- data4PCCAR::getImportantCtr(ctr = ctrK,
                                     eig = demo.mfa.2007$mexPosition$Table$eigs,
                                     axis1 = 1,
                                     axis2 = 2
                                     )
importantVar <- var12$importantCtr.1or2
col4ImportantVar <- col.design.col
col4NS <- 'gray90'
col4ImportantVar[!importantVar] <- col4NS
ctrV12.imp <- PTCA4CATA::createFactorMap(X = ctrK,
                                         title = "Important Variables: Contributions",
                                         # col.points = col4ImportantVar,
                                         # col.labels = col4ImportantVar,
                                         alpha.points = 0.5,
                                         cex = 2.5,
                                         alpha.labels = 1,
                                         text.cex = 4,
                                         font.face = "plain",
                                         font.family = "sans")
a0008.Var.ctr12.imp <- ctrV12.imp$zeMap + label4Map.mfa
#a0008.Var.ctr12.imp

#a0008.Var.ctr12.imp <- recordPlot()

label4Map23.mfa <- createxyLabels.gen(2,3,
                                    lambda = demo.mfa.2007$mexPosition.Data$Table$eigs,
                                    tau = demo.mfa.2007$mexPosition.Data$Table$t)
```

```
#Variable contribution map with Dimensions 2 & 3

var32 <- data4PCCAR::getImportantCtr(ctr = ctrK,
                                     eig = demo.mfa.2007$mexPosition.Data$Table$eigs,
                                     axis1 = 2,
                                     axis2 = 3
                                     )
importantVar32 <- var32$importantCtr.1or2
col4ImportantVar32 <- col.design.col
col4NS <- 'gray90'
col4ImportantVar32[!importantVar32] <- col4NS



ctrV32.imp <- PTCA4CATA::createFactorMap(X = ctrK,
                                         axis1 = 2, axis2 = 3,
                                         title = "Important Variables: Contributions 3 2",
                                         col.points = col4ImportantVar32,
                                         col.labels = col4ImportantVar32,
                                         alpha.points = 0.5,
                                         cex = 2.5,
                                         alpha.labels = 1,
                                         text.cex = 4,
                                         font.face = "plain",
                                         font.family = "sans")


a0009.Var.ctr32.imp <- ctrV32.imp$zeMap + label4Map23.mfa
#a0009.Var.ctr32.imp


#a0009.Var.ctr32.imp <- recordPlot()

library(grid)
library(gridExtra)
library(gridGraphics)
```

```r
library(ggplot2)
library(ggplotify)
grid.barplot1 <-gridExtra::grid.arrange(as.grob(plotctrJ.1 ),
                        as.grob(plotctrJ.2 ),
                         #as.grob(plotctrJ.3 ),
                        ncol = 1,nrow = 3 ,
                        top = textGrob("Column Contribution barplots",
                        gp=gpar(fontsize=18,font=3))
                       )
```




```r
grid.barplot1
```

```
## TableGrob (4 x 1) "arrange": 3 grobs
##   z     cells    name                 grob
## 1 1 (2-2,1-1) arrange       gtable[layout]
## 2 2 (3-3,1-1) arrange       gtable[layout]
## 3 3 (1-1,1-1) arrange text[GRID.text.4154]
```