

Chapter 3 MCA

MCA analyzes rectangular qualitative data tables (0/1)

Rows are observations described by the columns (variables)

MCA creates new best variables called components or factor scores

A component is a mixture of the original variables (only the 1's count)

The amount in the mixture of a variable is called its loading/score

MCA makes two maps

One map for the observations: Scores give the coordinates

One map for the variables: Scores give the coordinates

Let us see if we get the same analysis for mca as pca on the same dataset.

LETS START BY CLEARING THE ENVIRONMENT.

```
rm(list = ls())  
graphics.off()  
knitr::opts_chunk$set(echo = TRUE)
```

3.1 DATASET

We will be using the same quantitative data set which we used in pca.

How to make quantitative data into qualitative data?

This is done by binning the variables into groups.

Histograms will help us understand how can we bin the data this is an experimental process which can be tried and tested for different values.

I have stuck to binning each variable into 3 groups.

```
library(ExPosition)
```

```
library(InPosition)
```

```
library(PTCA4CATA)
```

```
library(corrplot)
```

```
library(ggplot2)
```

```
library(data4PCCAR)
```

```
## Registered S3 method overwritten by 'data4PCCAR':
```

```
##   method                      from
```

```
##   print.str_colorsOfMusic PTCA4CATA
```

```
##
```

```
## Attaching package: 'data4PCCAR'
```

```
## The following objects are masked from 'package:PTCA4CATA':
```

```
##
```

```
##   addArrows, addCircleOfCor
```

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

```
a=data.pca
```

```
b=data.more
```

```
x=data.pca
```

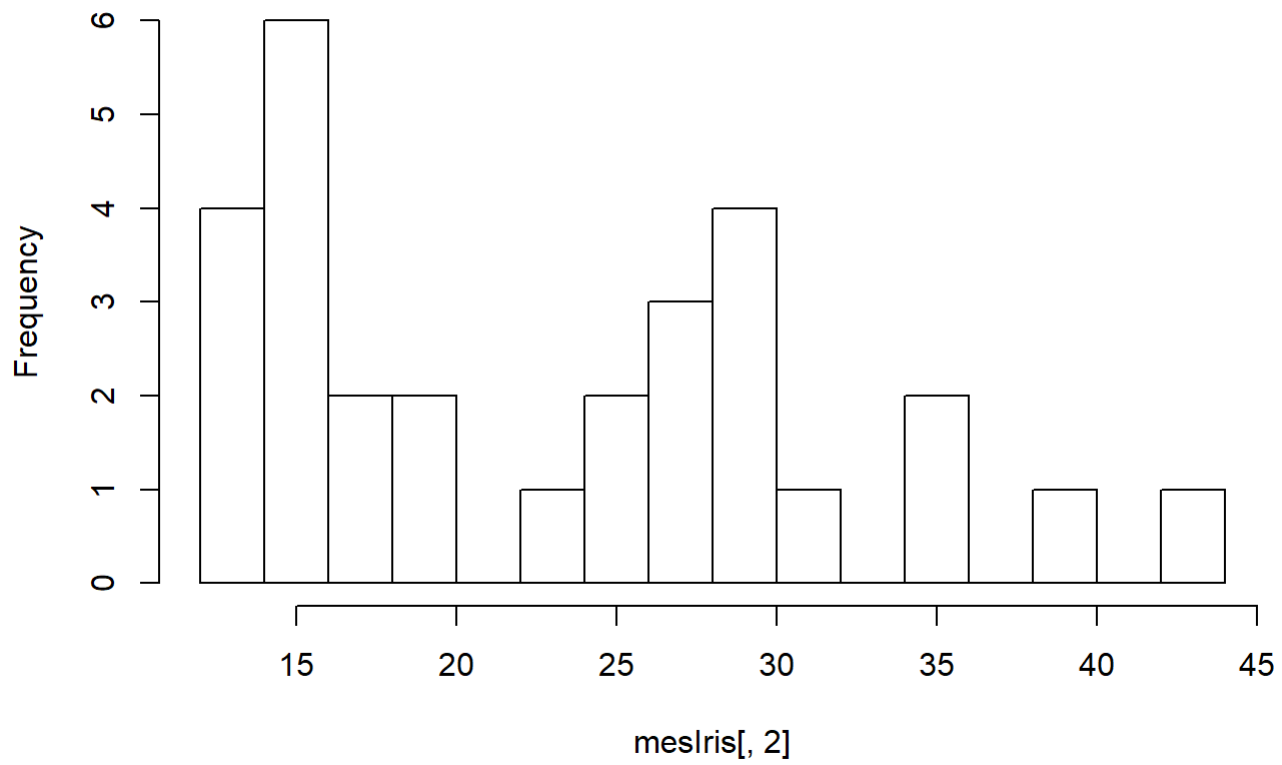
```
colnames(x)=c('pd','apps','appsmon','FEULP','FEUPM','FEULc','FEUMc')
```

```
mesIris <- x[,1:7]
```

```
grIris <- iris[,1]
```

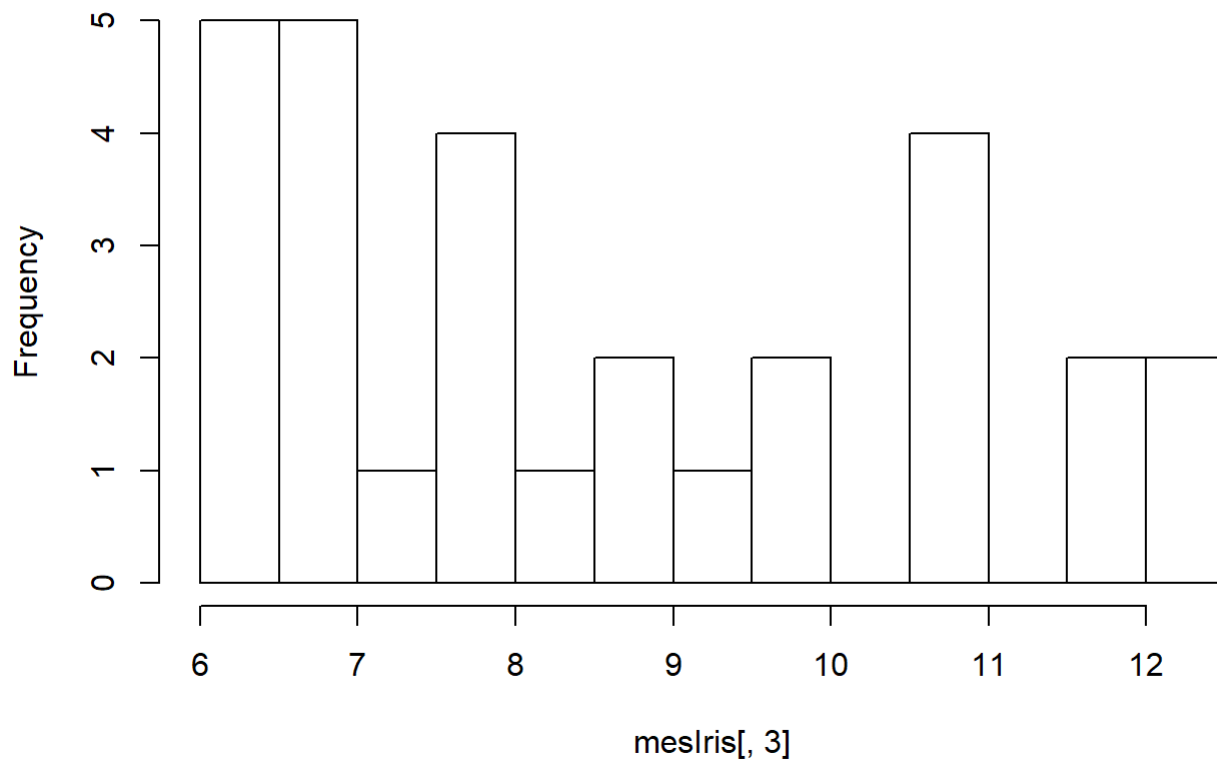
```
hist.SL <- hist(mesIris[,2], breaks = 20) # 4
```

Histogram of mesIris[, 2]



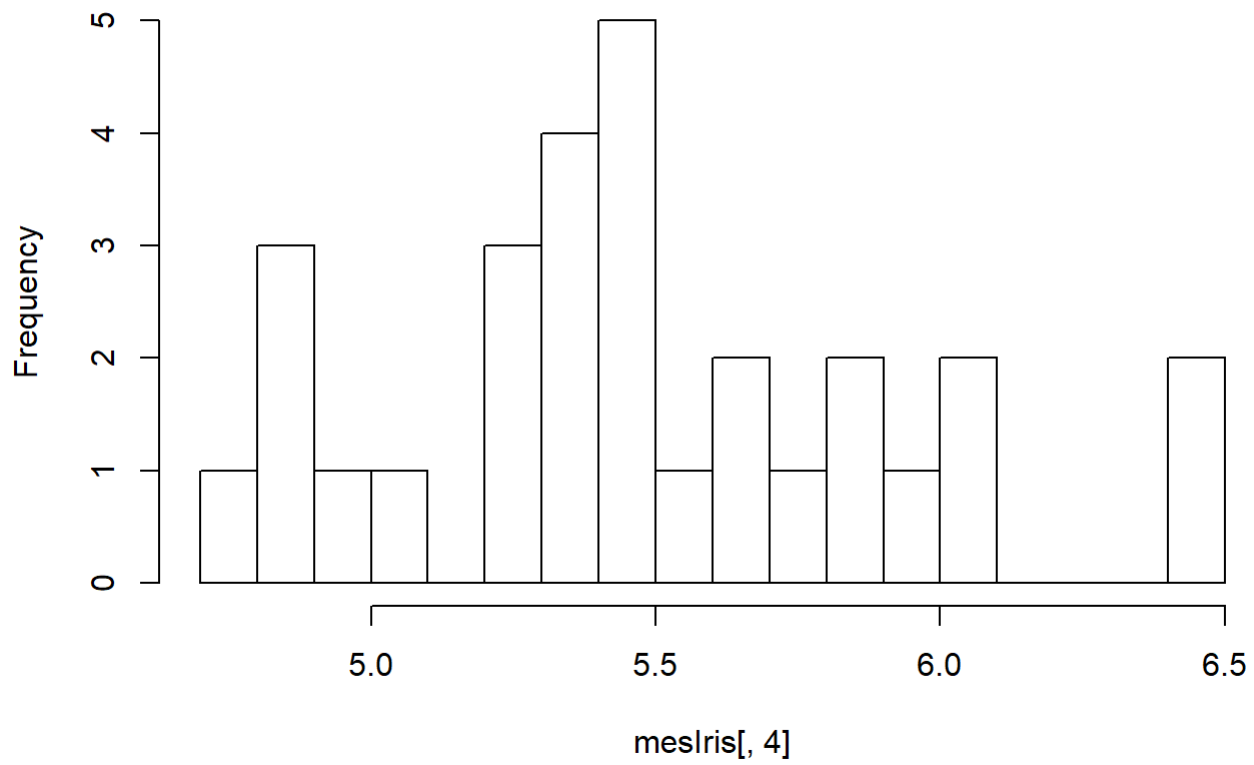
```
hist.SW <- hist(mesIris[,3], breaks = 20) # 3
```

Histogram of mesIris[, 3]



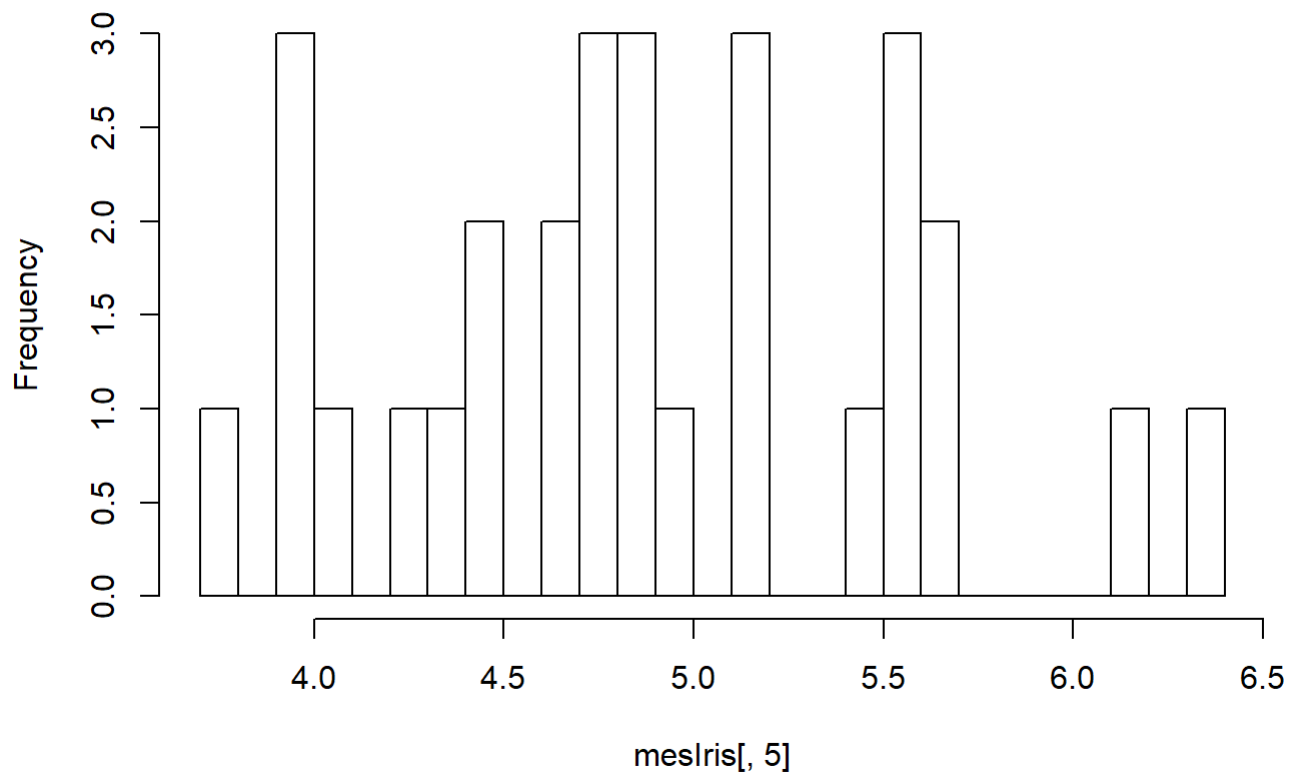
```
hist.PL <- hist(mesIris[,4], breaks = 20) # 3
```

Histogram of mesIris[, 4]



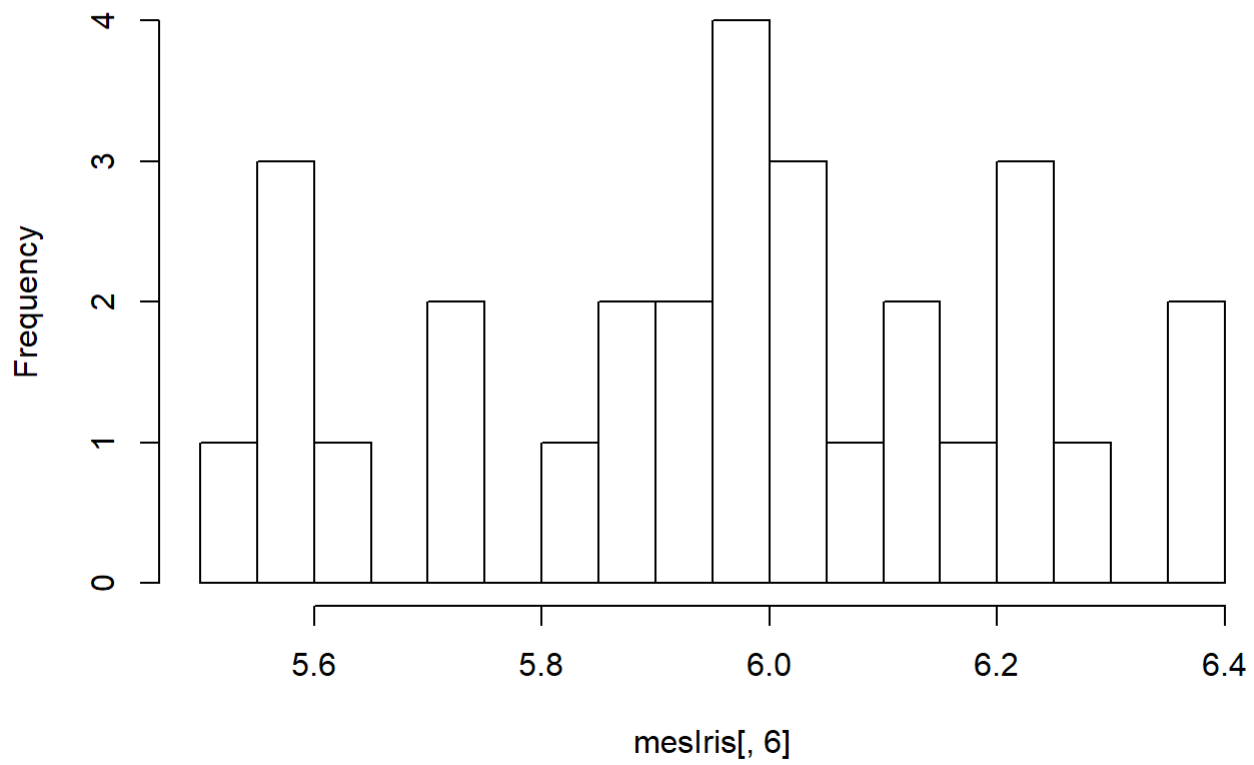
```
hist.PW <- hist(mesIris[,5], breaks = 20) # 3
```

Histogram of mesIris[, 5]



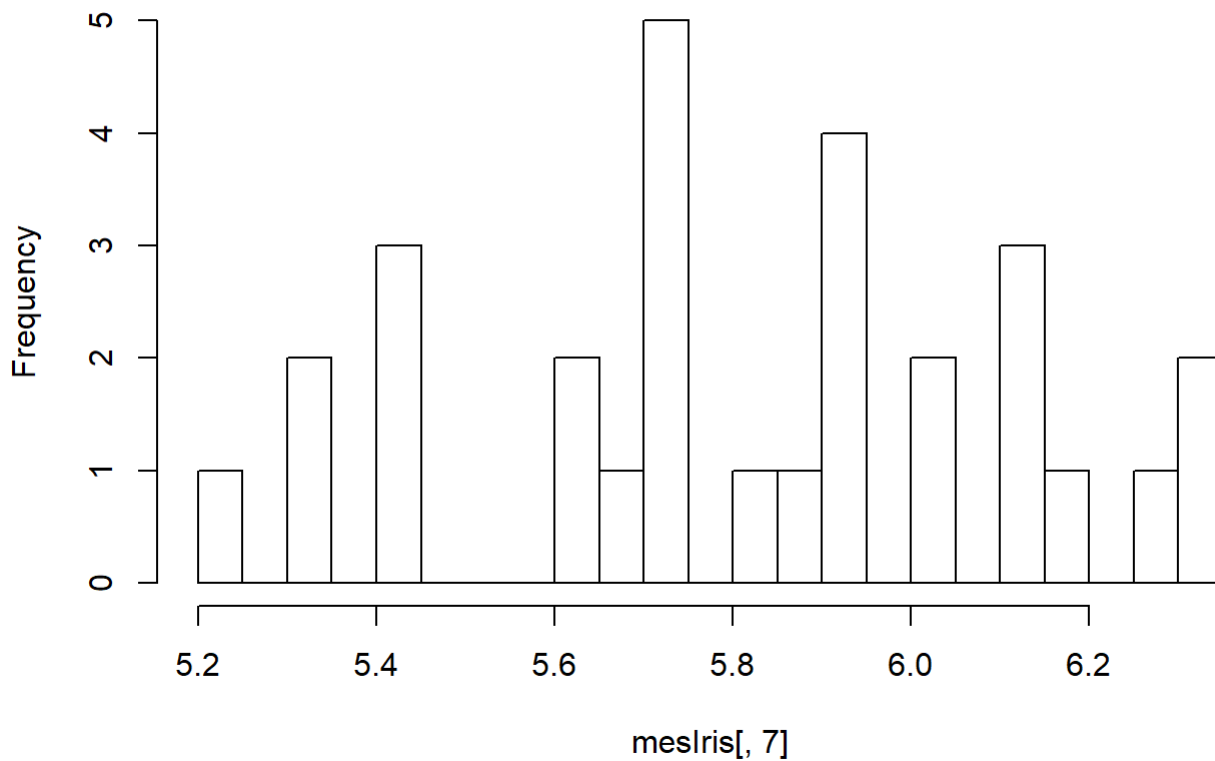
```
hist.So <- hist(mesIris[,6], breaks = 20) # 4
```

Histogram of mesIris[, 6]



```
hist.St <- hist(mesIris[,7], breaks = 20) # 3
```

Histogram of mesIris[, 7]



```
mesIrisRecoded <- data.frame(row.names = row.names(mesIris))
```

```
irec = which(colnames(mesIris)=='pd')
```

```
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(  
  mesIris[,irec], nClass = 3, stem = '')
```

```
cor(mesIris[,irec],as.numeric(mesIrisRecoded[,colnames(mesIris)[irec]]), method = "spearmanr")
```

```
## [1] 0.9429542
```

```
table(mesIrisRecoded[,colnames(mesIris)[irec]])
```



```
##
## 1 2 3
## 10 9 10

irec = which(colnames(mesIris)=='apps')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')
irec = which(colnames(mesIris)=='appsmon')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')
irec = which(colnames(mesIris)=='FEULP')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')

irec = which(colnames(mesIris)=='FEUPM')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')
irec = which(colnames(mesIris)=='FEULc')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')
irec = which(colnames(mesIris)=='FEUMc')
mesIrisRecoded[,colnames(mesIris)[irec]] <- BinQuant(
    mesIris[,irec], nClass = 3, stem = '')

mesIrisRecoded.dis <- makeNominalData(mesIrisRecoded)
```

3.2 COR PLOT

This plot will help us understand which of the variables are correlated to each other.

```
resMCA <- epMCA(DATA = mesIrisRecoded.dis,
  make_data_nominal = FALSE,
  DESIGN = x[,1],
  graphs = FALSE # TRUE first pass only
)
```

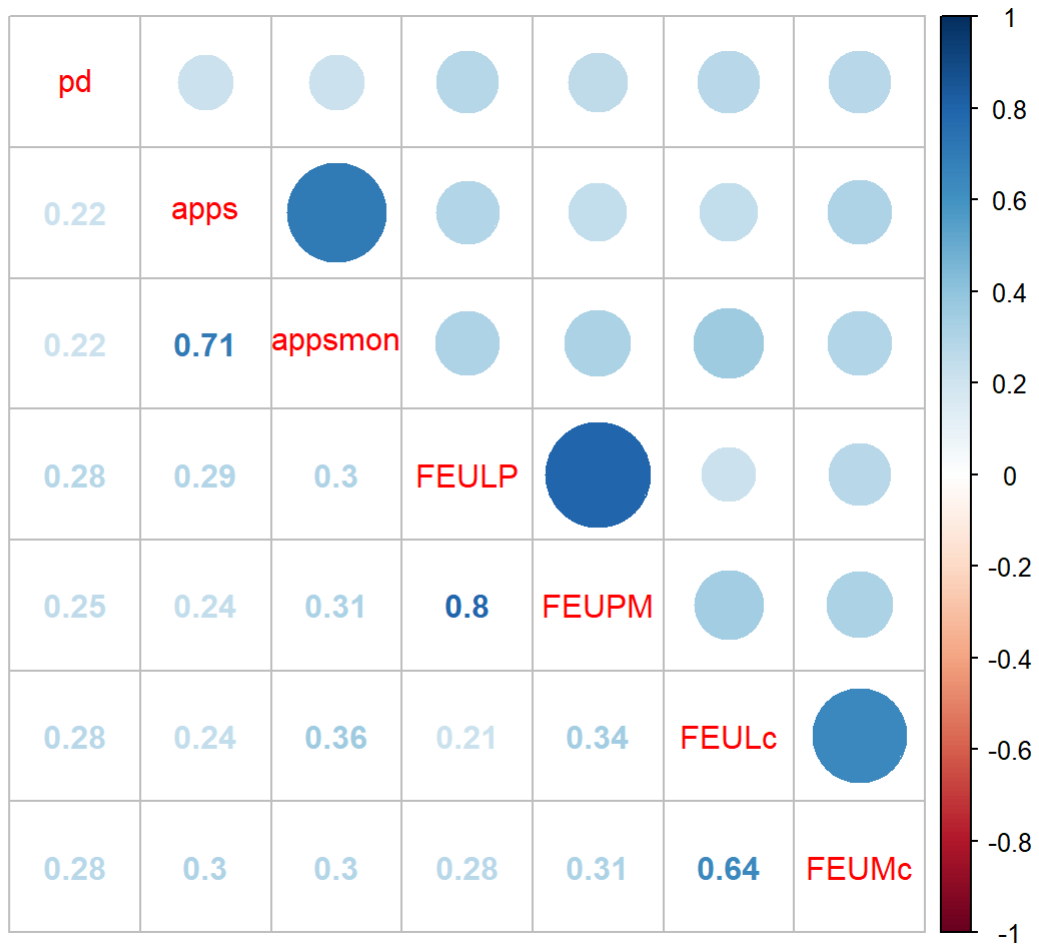
```
ctrK <- ctr4Variables(resMCA$ExPosition.Data$cj)
```

```
resMCA.inf <- InPosition::epMCA.inference.battery(DATA = mesIris,
  DESIGN = grIris,
  graphs = FALSE # TRUE first pass only
)
```

```
## [1] "Row dimensions do not match for X and Y. Creating default."
## [1] "It is estimated that your iterations will take 0 minutes."
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take
## =====
```



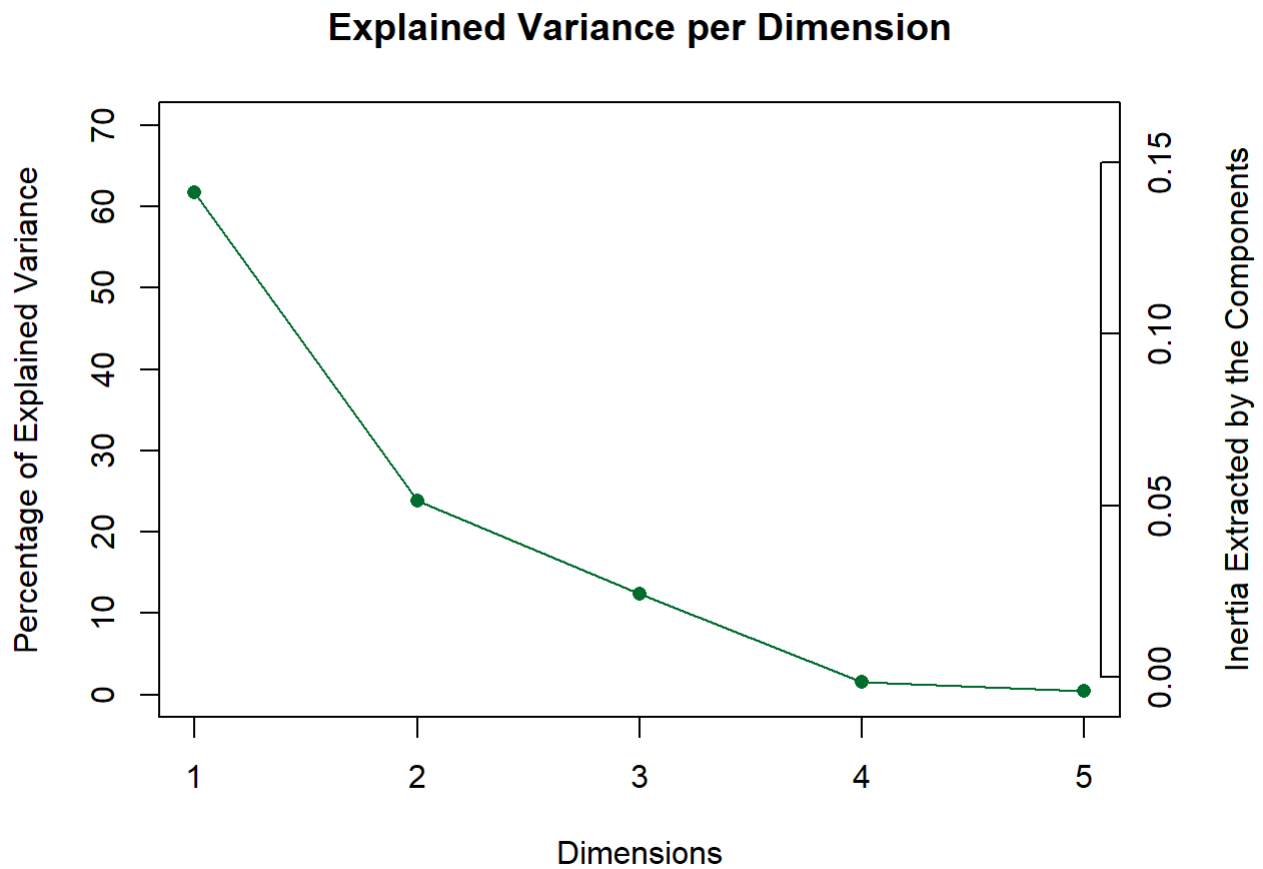
```
corrMatBurt.list <- phi2Mat4BurtTable(mesIrisRecoded)
corr4MCA.r <- corrplot.mixed(as.matrix(sqrt(corrMatBurt.list$phi2.mat),
  title = "Phi: Correlation Map for MCA"))
```



3.3 SCREE PLOT

According to the scree plot the first and the second dimensions explain the maximum variance.

```
PlotScree(ev = resMCA$ExPosition.Data$eigs)
```



```
a001a.screePlot <- recordPlot()
```

3.4 PROJECTION

```
iris.Imap <- PTCA4CATA::createFactorMap(
  title = 'MCA: Smartphones Data Set',
  resMCA$ExPosition.Data$fi,
  col.points = resMCA$Plotting.Data$fi.col,
  display.labels = TRUE,
  alpha.points = .5
)
iris.Imap
```

```
## -----  
## Basic Factor Maps (with ggplot2)  
## -----  
## $zeMap          A standard map with background, points, and Labels  
## $zeMap_background The background map  
## $zeMap_dots      The points map  
## $zeMap_text      The labels map  
## $factorScores    The factor scores (coordinates)  
## $constraints      map constraints (a list with minx miny maxx maxy)  
## -----
```

```

label4Map <- createxyLabels.gen(1,2,
                                lambda = resMCA$ExPosition.Data$eigs,
                                tau = resMCA$ExPosition.Data$t)

# _____

a002.Map.I <- iris.Imap$zeMap + label4Map


col4Var <- c('orange', 'orange4', 'red', 'red4', 'yellow', 'blue')
# Levels
col4Levels <- coloringLevels(rownames(resMCA$ExPosition.Data$fj),
                              col4Var)

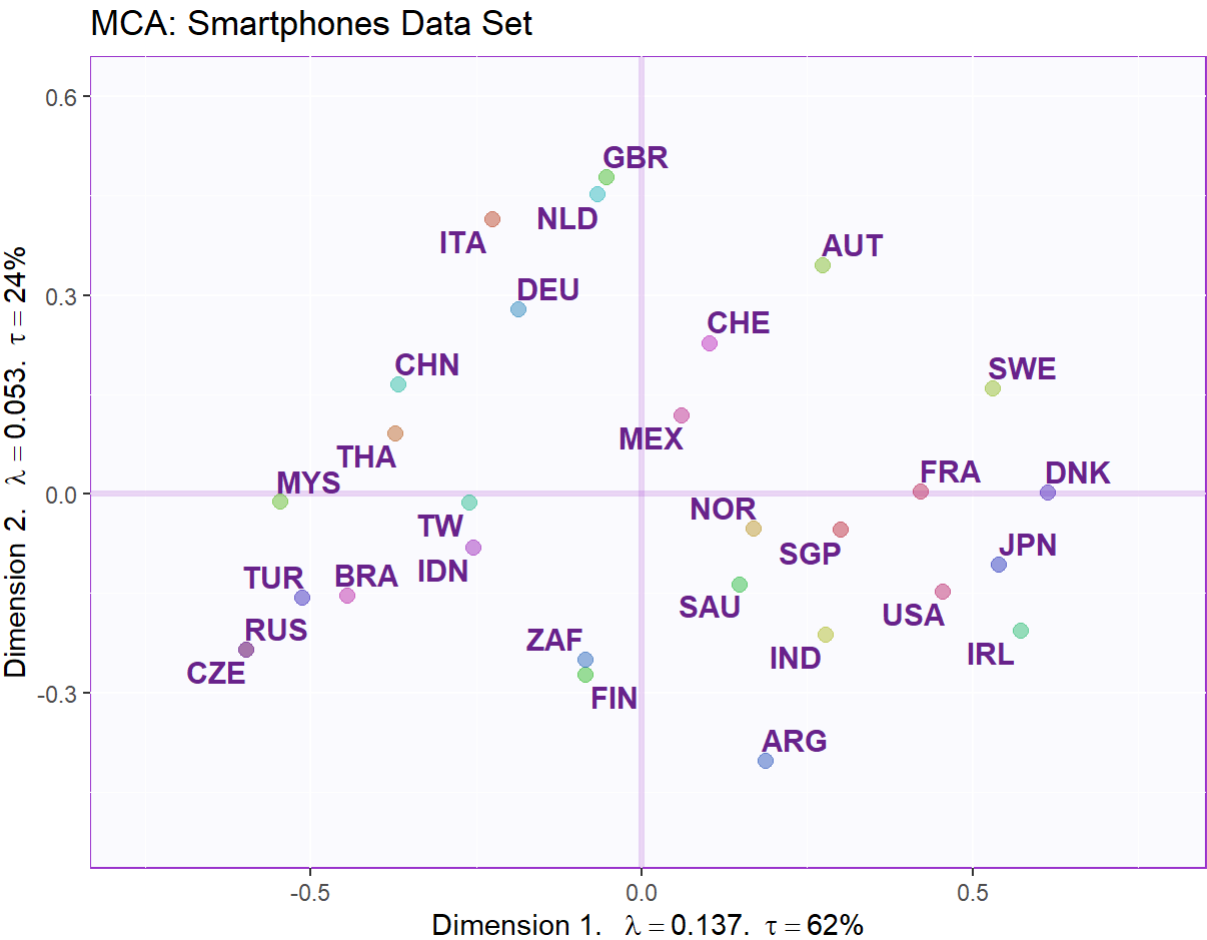
axis1 = 1
axis2 = 2
# to save typing
Fj <- resMCA$ExPosition.Data$fj
# _____

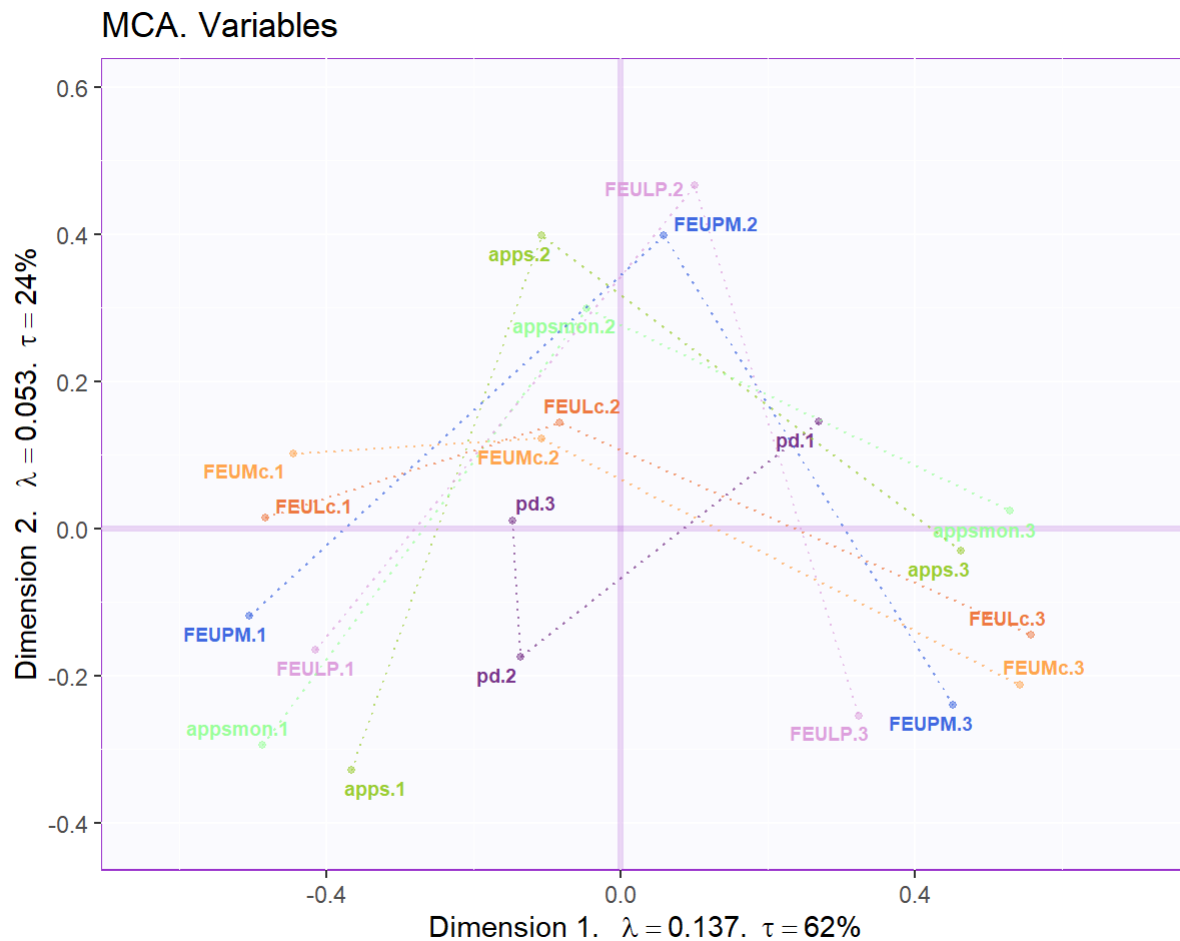
# generate the set of maps
BaseMap.Fj <- createFactorMap(X = Fj , # resMCA$ExPosition.Data$fj,
                              axis1 = axis1,
                              axis2 = axis2,
                              title = 'MCA. Variables',
                              col.points = col4Levels$color4Levels,
                              cex = 1,
                              col.labels = col4Levels$color4Levels,
                              text.cex = 2.5,
                              force = 2)


b001.BaseMap.Fj <- BaseMap.Fj$zeMap + label4Map
b002.BaseMapNoDot.Fj <- BaseMap.Fj$zeMap_background +
                        BaseMap.Fj$zeMap_text + label4Map

# add Lines ----
lines4J <- addLines4MCA(Fj, col4Var = col4Var)
b003.MapJ <- b001.BaseMap.Fj + lines4J

```





3.5 ANALYSIS/HYPOTHESIS

DEVELOPING ECONOMIES AND DEVELOPED ECONOMIES SHOW DIFFERENT TRENDS IN SMARTPHONES USAGE

PD1 IS EXPLAINED BY DIM1

PD2 IS EXPLAINED BY DIM1 AND DIM2

PD3 IS EXPLAINED BY DIM1

3.6 CONTRIBUTIONS

THE TWO COMPONENTS ARE MADE BY CONTRIBUTIONS OF THE VARIABLES AND OBSERVATIONS

CONTRIBUTION PLOTS WILL SHOW US HOW MUCH DOES EACH OBSERVATION AND EACH VARIABLE CONTRIBUTE FOR EACH COMPONENT

```

resMCA.sym <- epMCA(mesIrisRecoded.dis, symmetric = TRUE, graphs = FALSE)

signed.ctrI <- resMCA.sym$ExPosition.Data$ci * sign(resMCA.sym$ExPosition.Data$fi)
signed.ctrJ <- resMCA.sym$ExPosition.Data$cj * sign(resMCA.sym$ExPosition.Data$fj)

# plot contributions of rows for component 1
ctrI.1 <- PrettyBarPlot2(signed.ctrI[,1],
                        threshold = 1 / NROW(signed.ctrI),
                        font.size = 3,
                        color4bar = gplots::col2hex(resMCA.sym$Plotting.Data$fi.col), # we
                        ylab = 'Contributions',
                        ylim = c(1.2*min(signed.ctrI), 1.2*max(signed.ctrI))
) + ggtitle("Component 1", subtitle = 'rows')

# plot contributions of columns for component 1
ctrJ.1 <- PrettyBarPlot2(signed.ctrJ[,1],
                        threshold = 1 / NROW(signed.ctrJ),
                        font.size = 3,
                        color4bar = gplots::col2hex(resMCA.sym$Plotting.Data$fj.col), # we
                        ylab = 'Contributions',
                        ylim = c(1.2*min(signed.ctrJ), 1.2*max(signed.ctrJ))
) + ggtitle("", subtitle = 'columns')

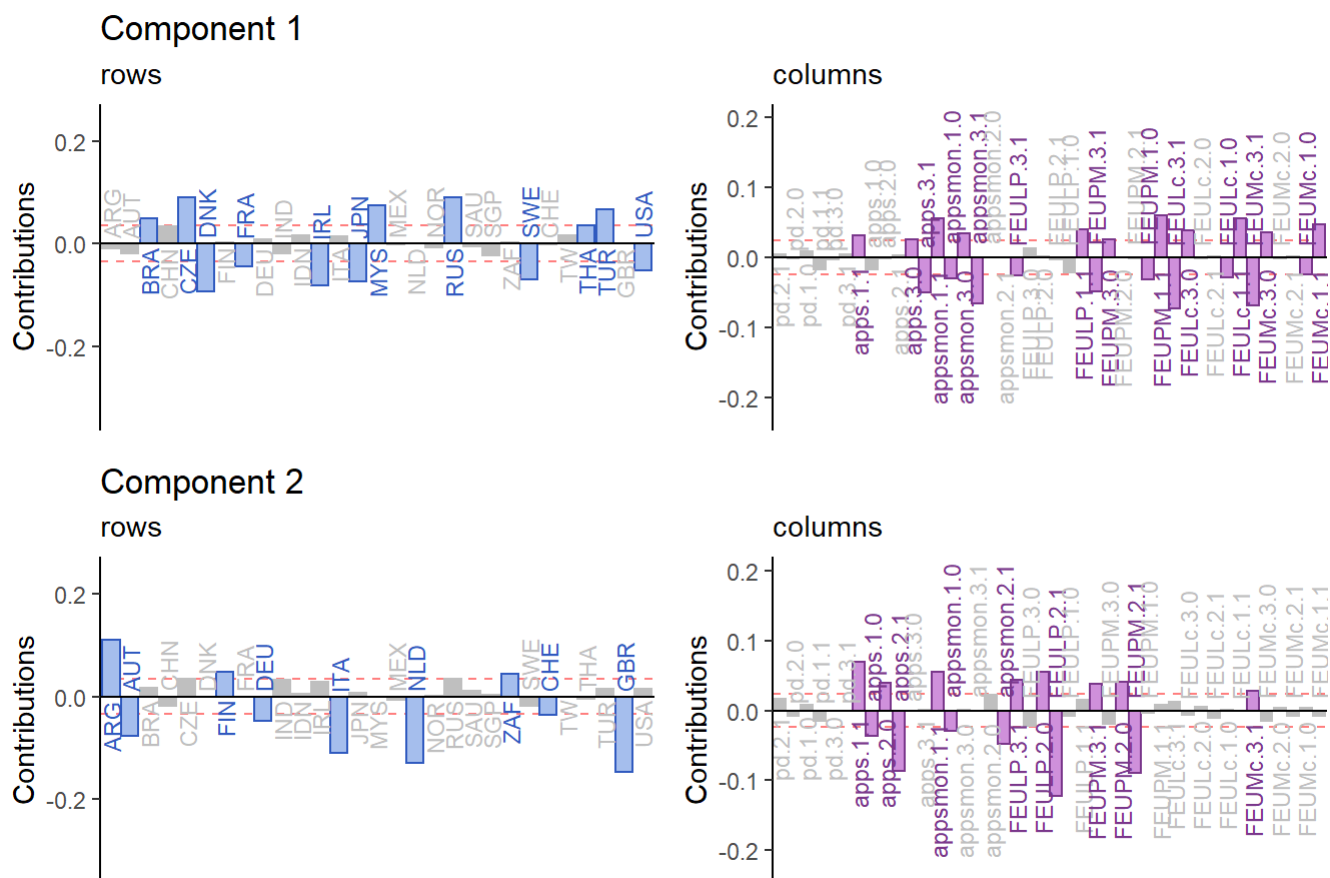
# plot contributions of rows for component 2
ctrI.2 <- PrettyBarPlot2(signed.ctrI[,2],
                        threshold = 1 / NROW(signed.ctrI),
                        font.size = 3,
                        color4bar = gplots::col2hex(resMCA.sym$Plotting.Data$fi.col), # we
                        ylab = 'Contributions',
                        ylim = c(1.2*min(signed.ctrI), 1.2*max(signed.ctrI))
) + ggtitle("Component 2", subtitle = 'rows')

library(gridExtra)
library(grid)
library(gridGraphics)
library(ggplotify)

```

```
# plot contributions of columns for component 2
ctrJ.2 <- PrettyBarPlot2(signed.ctrJ[,2],
                          threshold = 1 / NROW(signed.ctrJ),
                          font.size = 3,
                          color4bar = gplots::col2hex(resMCA.sym$Plotting.Data$fj.col), # we
                          ylab = 'Contributions',
                          ylim = c(1.2*min(signed.ctrJ), 1.2*max(signed.ctrJ))
) + ggtitle("", subtitle = 'columns')
grid.arrange(
  as.grob(ctrI.1), as.grob(ctrJ.1), as.grob(ctrI.2), as.grob(ctrJ.2),
  ncol = 2, nrow = 2,
  top = textGrob("Contributions", gp = gpar(fontsize = 18, font = 3))
)
```

Contributions



3.7 BOOTSTRAP TEST

FOR OUR HYPOTHESIS THAT DEVELOPING ECONOMIES AND DEVELOPED ECONOMIES SHOW DIFFERENT TRENDS IN SMARTPHONES PURCHASE WE NEED TO CHECK THE CONTRIBUTIONS WITH THE BOOTSTRAP METHOD.

FROM THE BOOTSTRAP MAP WE CAN SAY THAT CONTRIBUTIONS SHOWN BY ROWS AND COLUMNS ARE SIGNIFICANT HENCE WE CAN REJECT THE NULL HYPOTHESIS.

```
resMCAinf.sym4bootI <- epMCA.inference.battery(t(mesIrisRecoded.dis), symmetric = TRUE, gr
```

```
## [1] "It is estimated that your iterations will take 0.03 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take  
## =====
```

```
resMCAinf.sym4bootJ <- epMCA.inference.battery(mesIrisRecoded.dis, symmetric = TRUE, graph
```

```
## [1] "It is estimated that your iterations will take 0 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take  
## =====
```

```
# you need this line to be able to save them in the endBootstrap ratios
```

```
BR.I <- resMCAinf.sym4bootI$Inference.Data$fj.boots$tests$boot.ratios
```

```
BR.J <- resMCAinf.sym4bootJ$Inference.Data$fj.boots$tests$boot.ratios
```

```
laDim = 1
```

```
library(gplots)
```

```
##
```

```
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      lowess
```

```

# Plot the bootstrap ratios for Dimension 1
ba001.BR1.I <- PrettyBarPlot2(BR.I[,laDim],
                              threshold = 2,
                              font.size = 3,
                              color4bar = NULL, # we need hex code
                              ylab = 'Bootstrap ratios'
                              #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle(paste0('Component ', laDim), subtitle = 'rows')

ba002.BR1.J <- PrettyBarPlot2(BR.J[,laDim],
                              threshold = 2,
                              font.size = 3,
                              color4bar = NULL, # we need hex code
                              ylab = 'Bootstrap ratios'
                              #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("", subtitle = 'columns')

```

```

# Plot the bootstrap ratios for Dimension 2
laDim = 2
ba003.BR2.I <- PrettyBarPlot2(BR.I[,laDim],
                              threshold = 2,
                              font.size = 3,
                              color4bar = NULL, # we need hex code
                              ylab = 'Bootstrap ratios'
                              #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle(paste0('Component ', laDim), subtitle = 'rows')

```

```

ba004.BR2.J <- PrettyBarPlot2(BR.J[,laDim],
                               threshold = 2,
                               font.size = 3,
                               color4bar = NULL, # we need hex code
                               ylab = 'Bootstrap ratios'
                               #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim])))
) + ggtitle("", subtitle = 'columns')

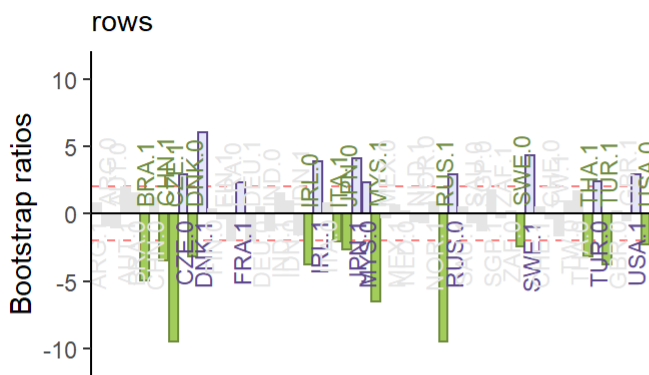
##We then use the next line of code to put two figures side to side:

grid.arrange(
  as.grob(ba001.BR1.I),as.grob(ba002.BR1.J),as.grob(ba003.BR2.I),as.grob(ba004.BR2.J),
  ncol = 2,nrow = 2,
  top = textGrob("Bootstrap ratios", gp = gpar(fontsize = 18, font = 3))
)

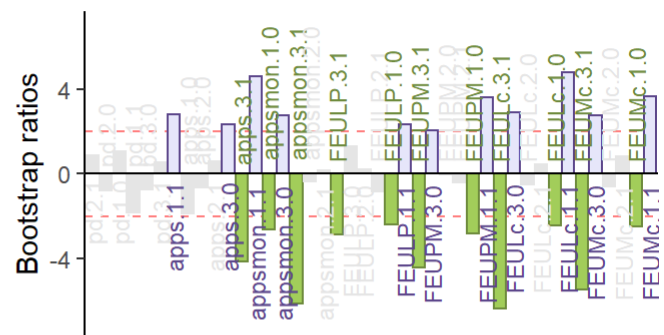
```

Bootstrap ratios

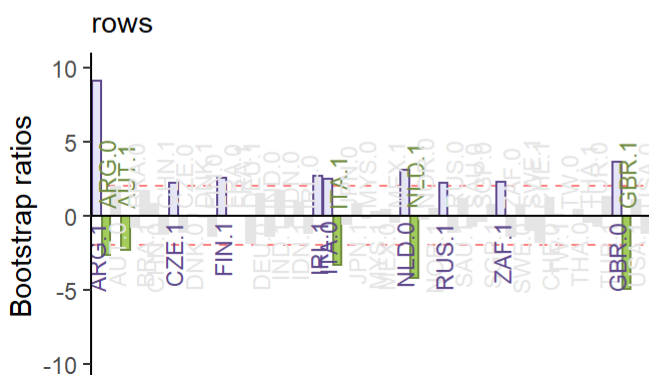
Component 1



columns



Component 2



columns

