Chapter 8 CA

wE WILL NOT BE USING SmartphoneUsage DATASET FOR CA

CA transforms a data table into two sets of new variables called factor scores (obtained as linear combinations of, respectively, the rows and columns): one set for the rows and one set for the columns. These factor scores give the best representation of the similarity structure of, respectively,the rows and the columns of the table. In addition, the factor scores can be plotted as maps that optimally display the information in the original table. In these maps, rows and columns are represented as points whose coordinates are the factor scores and where the dimensions are also called factors, components(by analogy with PCA), or simply dimensions. Interestingly, the factor scores of the rows and the columns have the same variance and, therefore, the rows and columns can be conveniently represented in one single map.

In correspondence analysis, the total variance (often called inertia) of the factor scores is proportional to the independence chi-square statistic of this table and, therefore, the factor scores in CA decompose this chi-square into orthogonal components.

Always have a fresh start.

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

8.1 DATA SET

The data is about alchol consumption and the observations are countries.

The variables are ‘beer’ ‘wine’ ‘spirit’ and ‘others’.

For CA we would be dividing countries into two parts active and supplementary respectively.

Variables ‘beer’ ‘wine’ and ‘spirit’ would be active and ‘others’ would be supplementary.

Below is the code for data manipulation.

```
library(readxl)
AlcoholEU_2_ <- read_excel("AlcoholEU(2).xlsx")


## New names:
## * `` -> ...1


View(AlcoholEU_2_)


keeps <- c("ObsTyps", "Beer","Wine","Spirit","Other")
mydata<- AlcoholEU_2_[ keeps]
myactive=mydata[mydata$ObsTyps %in% c("ACTIVE"),]
keepss=c("Beer","Wine","Spirit")
myactive1<-myactive[keepss]
X=as.matrix(myactive1)
rownames(X) <- c('Belarus','Belgium','Bulgaria','Croatia','CzechRepublic','Estonia','France
                 'Lithuania','Luxembourg','Netherlands','Poland','Portugal','Romania','Slov
myactive3=mydata[mydata$ObsTyps %in% c("SUPP"),]
MYACTIVE4=myactive3[keepss]
HApunct=as.matrix(MYACTIVE4)
k=c("Other")
other=mydata[mydata$ObsTyps %in% c("ACTIVE"),]
otherPunct=as.matrix(other[k])
```

8.2 COR PLOT

Here we get the correlation plot between supplementary countries and beer wine and spirt.

HYPOTHESES

Beer shows significant negative correlation with France.

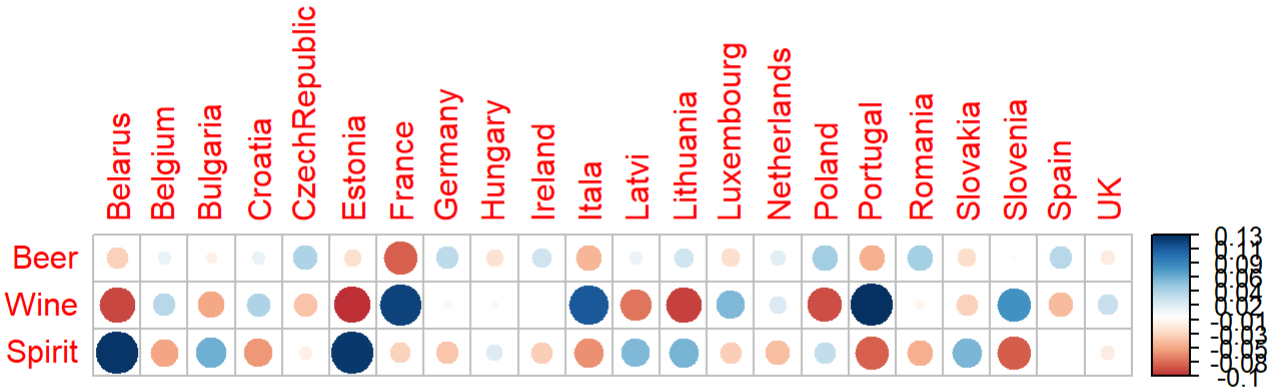
Wine shows negative correlation with Belarus Estonia Lithuania Poland and shows positive correlation with France and Portugal.

Spirit shows positive correlation with Belarus Estonia.

```
chi2    <-  chisq.test(X)
```

```
## Warning in chisq.test(X): Chi-squared approximation may be incorrect
```

```
# Components of chi2: the chi-squares for each cell before we add them up to compute the ch
Inertia.cells <- chi2$residuals / sqrt(sum(X))
# To be Plotted

# You can always compute it directly from the data
Z <- X / sum(X) # observed
r <- as.matrix(rowSums(Z)) # expected for each row
c <- as.matrix(colSums(Z)) # expected for each column
# Inertia.cells
test.Inertia.cells <- diag( as.vector(r^(-1/2)) ) %*%
                    (Z - r%*%t(c) ) %*% diag(as.vector(c^(-1/2)))


# Components of chi2: the chi-squares for each cell before we add them up to compute the ch
Inertia.cells <- chi2$residuals / sqrt(sum(X))
# To be Plotted
library(corrplot)
corrplot(t(Inertia.cells), is.cor = FALSE)
```




```
a0.residuals <- recordPlot()
```

8.3 SCREE PLOT

It is evident that we have two components showings maximum variance.

So we will reduce the dimensions to 2 for the analysis.

```
library(InPosition)
library(ExPosition)
library(prettyGraphs)
resCA.sym  <- epCA(X, symmetric = TRUE, graphs = FALSE)
resCAinf.sym4bootJ  <- epCA.inference.battery(X, symmetric = TRUE, graphs = FALSE)
```

```
## [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
## ================================================================================
```

```
resCAinf.sym4bootI  <- epCA.inference.battery(t(X), symmetric = TRUE, graphs = FALSE)
```

```
## [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
## ================================================================================
```

```
resCA.asym <- epCA(X, symmetric = FALSE, graphs = FALSE)

HA.sup <- supplementaryRows(SUP.DATA = HApunct, res = resCA.sym)

punct.sup <- supplementaryCols(SUP.DATA = otherPunct, res = resCA.sym)

library(PTCA4CATA)
#my.scree <- PlotScree(ev = resCA.sym$ExPosition.Data$eigs,
                      #p.ev = resCAinf.sym4bootI$Inference.Data$components$p.vals)
#my.scree
```

8.4 PROJECTIONS

From the scree plot we get two dimensions on which we will project all our obserations and variables (by calculating the factor scores).

Asymmetric:

From these two plots we can say that Beer is mainly explained by dim 2 and wine and spirit are explained by both the dimensions.

For wine and spirit dim 1 shows max variance.

Symmetric:

NOTE The points 1 2 3 on the graph are Andorra Russia Ukraine which are the supplementary observations.

All the active countries and active variables are projected onto a two dimension and if compare these projections with the actual data set we see the same relations.

Now when we project supplememtarty observations and column (other) which were not used in calculating our two components.

The variance of (Other) is explained by both the components.

1 is closer to wine than beer

1 is closer to beer than spirit.

1 is closer to spirit than other.

2 is closer to spirit than beer

2 is closer to beer than other.

2 is closer to other than wine.

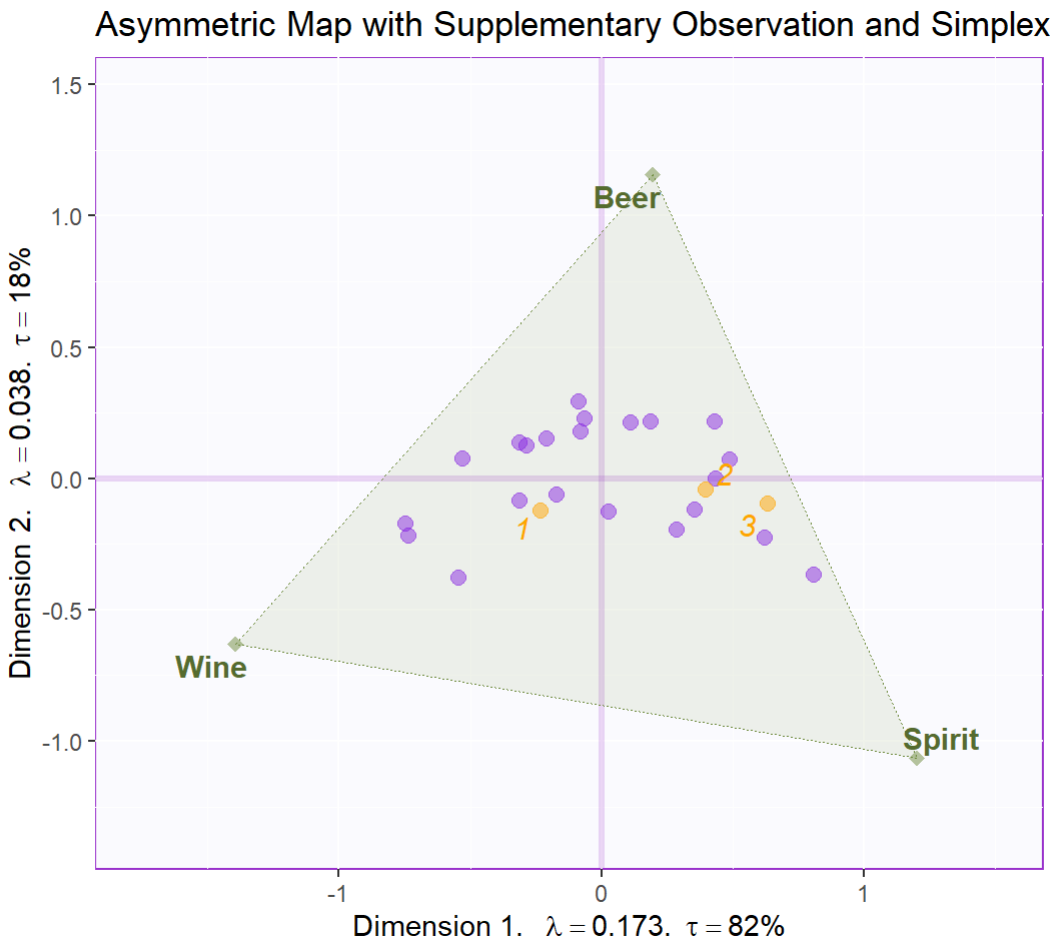
3 is closer to spirit than beer

3 is closer to beer than wine.

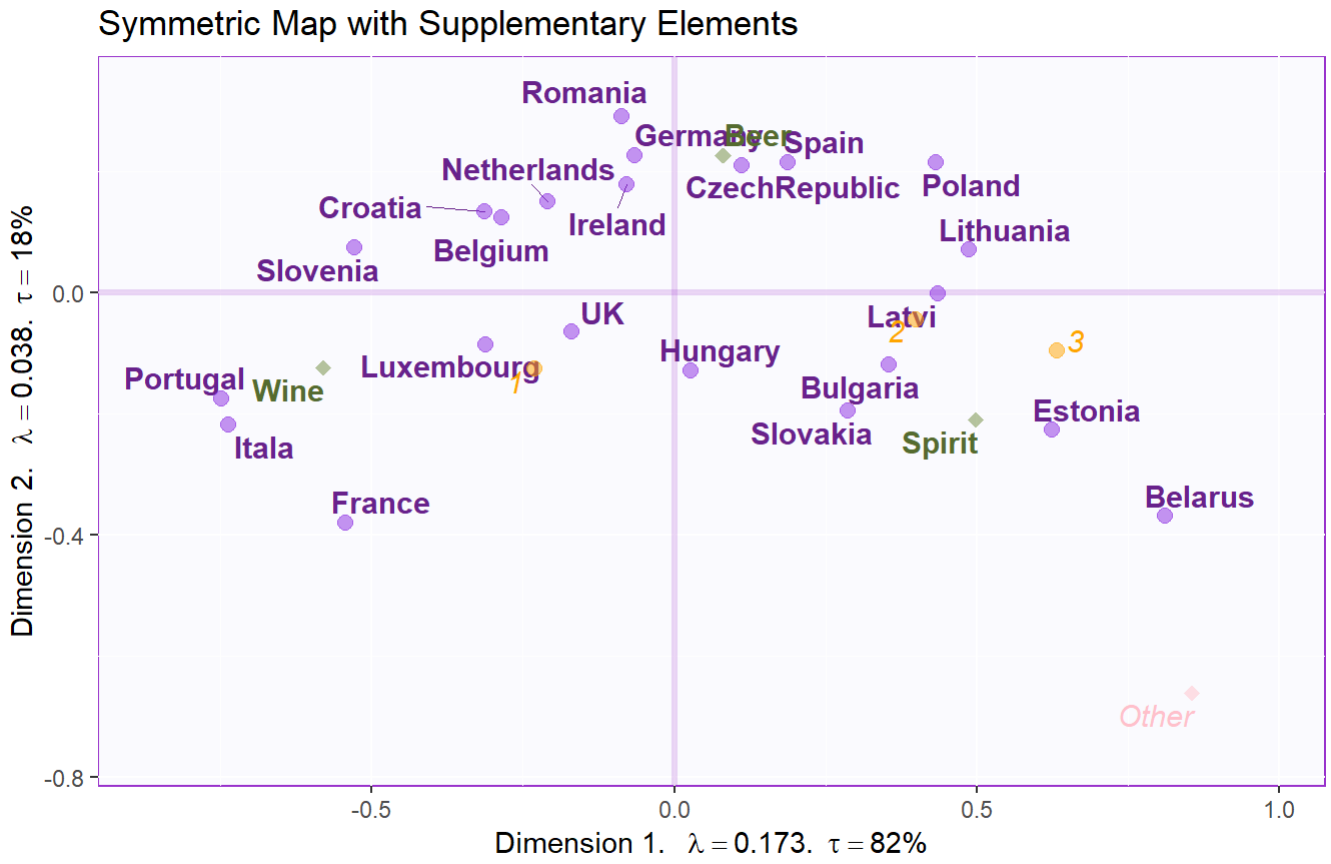
3 is closer to wine than other.

```r
Fj.a <- resCA.asym$ExPosition.Data$fj
Fi   <- resCA.sym$ExPosition.Data$fi
Fj   <- resCA.sym$ExPosition.Data$fj

# constraints -----
# first get the constraints correct
constraints.sym  <- minmaxHelper(mat1 = Fi, mat2  = Fj)
constraints.asym <- minmaxHelper(mat1 = Fi, mat2  = Fj.a)
constraints.sup  <- minmaxHelper(mat1 = rbind(Fi, HA.sup$fii),
                                 mat2  = rbind(Fj, punct.sup$fjj) )

# Get some colors ----
color4Authors <-prettyGraphsColorSelection(n.colors = nrow(Fi))
# baseMaps ----
colnames(Fi) <- paste("Dimension ", 1:ncol(Fi))
colnames(Fj) <- paste("Dimension ", 1:ncol(Fj))
colnames(Fj.a) <- paste("Dimension ", 1:ncol(Fj.a))
asymMap  <- createFactorMapIJ(Fi,Fj.a)
mapSup <- createFactorMapIJ(as.data.frame(HA.sup$fii),
                            as.data.frame(punct.sup$fjj)  ,
                            col.points.i = "Orange",
                            col.labels.i = 'Orange' ,
                            font.face.i = 'italic',
                            alpha.labels.i = .8,
                            alpha.points.i = .8,
                            col.points.j = 'Pink',
                            col.labels.j = 'Pink',
                            alpha.labels.j = .9,
                            font.face.j = 'italic',
                            alpha.points.j = .8,
                            constraints = constraints.sup
)

# Make the simplex visible
zePoly.J <-  PTCA4CATA::ggdrawPolygon(Fj.a,
```

```
                                    color = 'darkolivegreen4',
                                    size = .2,
                                    fill =  'darkolivegreen4',
                                    alpha = .1)
# Labels
labels4CA <- createxyLabels(resCA = resCA.asym)
library(ggplot2)
library(ggplotify)

# Combine all elements you want to include in this plot
map.I.sup.asym <- asymMap$baseMap + zePoly.J +
                          asymMap$I_points +
                          asymMap$J_labels + asymMap$J_points +
                          mapSup$I_labels + mapSup$I_points +
                          labels4CA +
ggtitle('Asymmetric Map with Supplementary Observation and Simplex')
map.I.sup.asym
```


Asymmetric Map with Supplementary Observation and Simplex
Beer
Wine
Spirit
1
2
3
1.5
1.0
0.5
0.0
-0.5
-1.0
-1
0
1
Dimension 2. λ = 0.038. τ = 18%
Dimension 1. λ = 0.173. τ = 82%

```
symMap  <- createFactorMapIJ(Fi,Fj)
# supplementary elements
mapSup <- createFactorMapIJ(as.data.frame(HA.sup$fii),
                            as.data.frame(punct.sup$fjj)  ,
                            col.points.i = "Orange",
                            col.labels.i = 'Orange' ,
                            font.face.i = 'italic',
                            alpha.labels.i = .8,
                            alpha.points.i = .8,
                            col.points.j = 'Pink',
                            col.labels.j = 'Pink',
                            alpha.labels.j = .9,
                            font.face.j = 'italic',
                            alpha.points.j = .8,
                            constraints = constraints.sup
)

map.IJ.sup.sym <- mapSup$baseMap + # the baseMap needs to come from your mapSup to be corre
  symMap$I_labels + symMap$I_points +
  symMap$J_labels + symMap$J_points +
  mapSup$I_labels + mapSup$I_points +
  mapSup$J_labels + mapSup$J_points +
  ggtitle('Symmetric Map with Supplementary Elements') +
  labels4CA
map.IJ.sup.sym
```



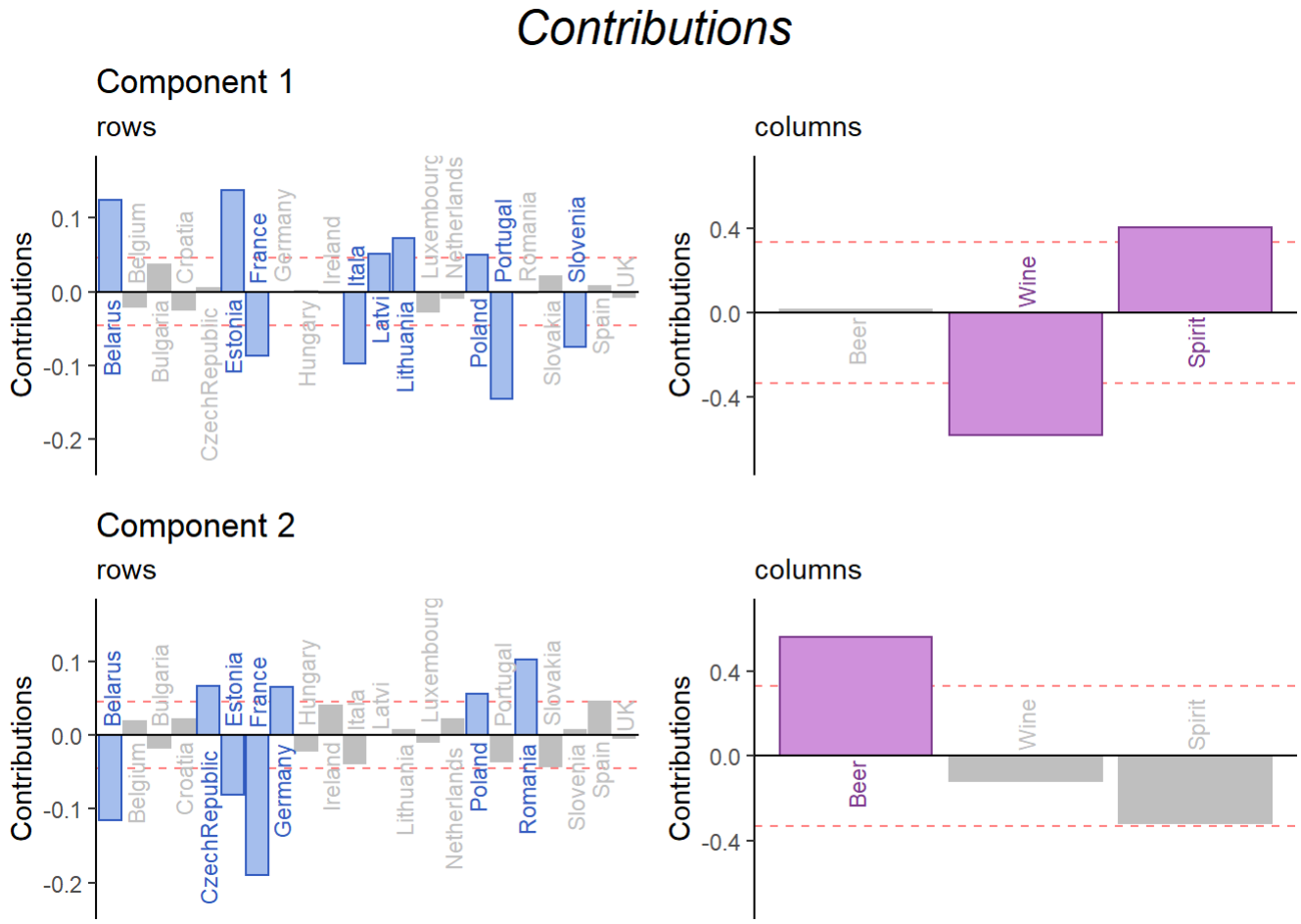
8.5 CONTRIBUTIONS

Here we get to see which active observation is explained by which component.

And which active variable is exaplained by which componenet. Note (you will find assymetric interpretations same as this.)

```
map.sepI.sup.sym <- mapSup$baseMap  +
  symMap$I_labels + symMap$I_points +
  mapSup$I_labels + mapSup$I_points+
  ggtitle('Symmetric: Row') +
  labels4CA

# plot the columns factor scores with confidence intervals
map.sepJ.sup.sym <- mapSup$baseMap +
  symMap$J_labels + symMap$J_points +
  mapSup$J_labels + mapSup$J_points +
  ggtitle('Symmetric: Column') +
  labels4CA
signed.ctrI <- resCA.sym$ExPosition.Data$ci * sign(resCA.sym$ExPosition.Data$fi)
signed.ctrJ <- resCA.sym$ExPosition.Data$cj * sign(resCA.sym$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(resCA.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(resCA.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(resCA.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')

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

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

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))
  )
```



8.6 BOOTSTRAP RATOS.

For deciding how reliable our interpretations and our hypotheses are we will need the bootstrap ratio.

If we compare our contribution plot with the bootstrap plots for columns we can see it is pretty reliable and for rows we can say the first dimension is reliable.

```r
 # you need this line to be able to save them in the endBootstrap ratios
BR.I <- resCAinf.sym4bootI$Inference.Data$fj.boots$tests$boot.ratios
BR.J <- resCAinf.sym4bootJ$Inference.Data$fj.boots$tests$boot.ratios

laDim = 1

# Plot the bootstrap ratios for Dimension 1
ba001.BR1.I <- PrettyBarPlot2(BR.I[,laDim],
                        threshold = 2,
                        font.size = 3,
                     color4bar = gplots::col2hex(resCA.sym$Plotting.Data$fi.col), # we need h
                    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 = gplots::col2hex(resCA.sym$Plotting.Data$fj.col), # we need h
                    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 = gplots::col2hex(resCA.sym$Plotting.Data$fi.col), # we need h
                    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 = gplots::col2hex(resCA.sym$Plotting.Data$fj.col), # we need h
                    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))
  )
```