# Chapter 2 PCA

Principal componentt analysis (PCA) is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables. Its goal is to extract the important information from the table, to represent it as a set of new orthogonal variables called principal components, and to display the pattern of similarity of the observations and of the variables as points in maps. The quality of the PCA model can be evaluated using cross-validation techniques such as the bootstrap and the jackknife. PCA can be generalized as correspondence analysis (CA) in order to handle qualitative variables and as multiple factor analysis (MFA) in order to handle heterogeneous sets of variables. Mathematically, PCA depends upon the eigen-decomposition of positive semidefinite matrices and upon the singular value decomposition (SVD) of rectangular matrices.

LETS START BY CLEARING THE ENVIRONMENT.

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

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

## 2.1 DATASET

The dataset used is SmartphonesUsage.

COLUMNS/VARIABLES:

(1)Smartphone purchase decision

(2)Number of apps on phone

(3)Number of apps used in last 30 days

(4)Frequency of email use on smartphone_liekert

(5)Frequency of email use on smartphone_Month

(6)Frequency of email use on Computer_liekert

(7)Frequency of email use on Computer_Month

ROWS/OBSERVATION:

29 countries out of which 28 belong to 4 different continents Asia, North America, South America, EUROPE. We have just one country from the continent of Africa.

Load the dataset

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

Import Packages for PCA

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(ExPosition)
```

```
## Loading required package: prettyGraphs
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)
library(PTCA4CATA)
```

```
## Warning: replacing previous import 'coin::confint' by 'stats::confint' when
## loading 'PTCA4CATA'
```

```
library(InPosition)
```

```
##
## Attaching package: 'InPosition'
```

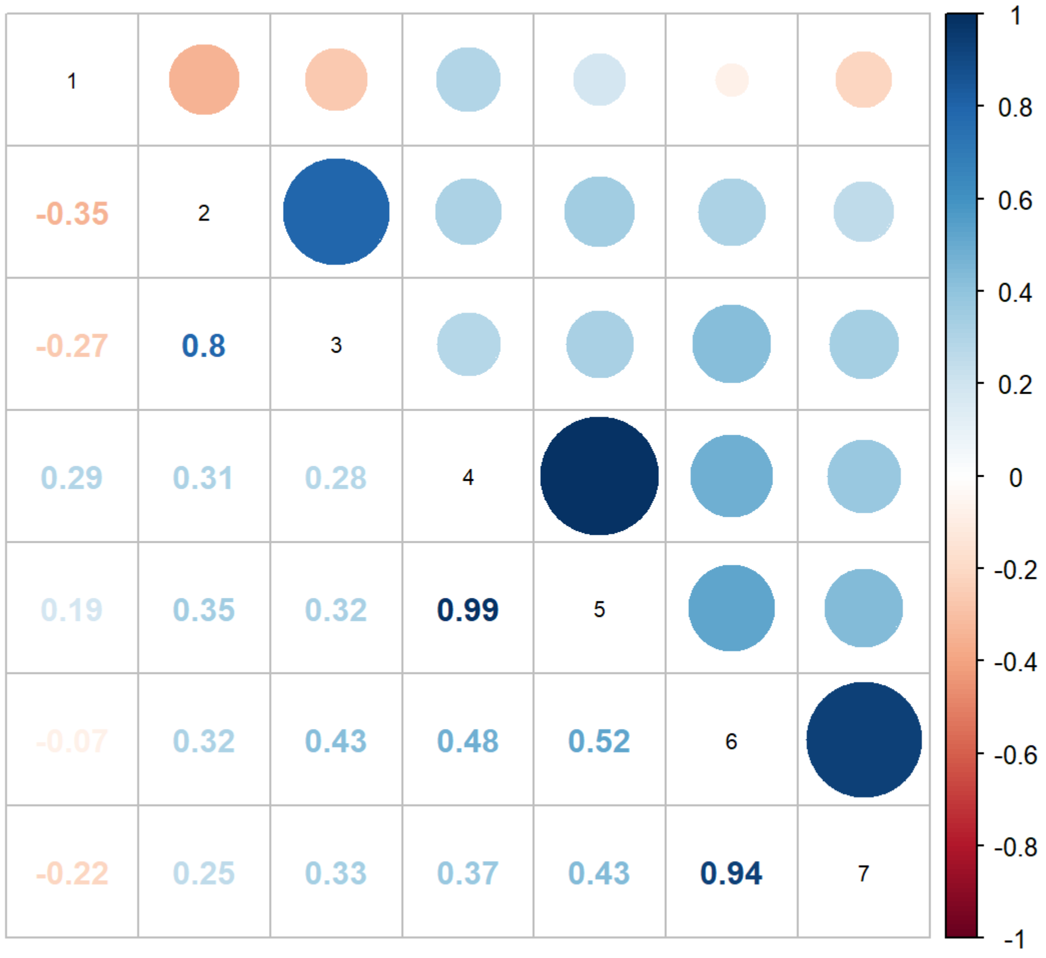
```
## The following object is masked from 'package:PTCA4CATA':
##
##     boot.ratio.test
```

```
library(gridExtra)
library(ggplotify)
library(grid)
```

## 2.2 COR PLOT:

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

```
colnames(a)=c('1','2','3','4','5','6','7')
cor.res <- cor(a)
corrplot.mixed(cor.res, tl.cex = 0.7, tl.col = "black")
```



```
cor.plot <- recordPlot()
```

PCA CALCULATIONS

```
b=data.more
res_pcaInf <- epPCA.inference.battery(a, center = TRUE, scale = "SS1", DESIGN = b$Continent
```

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

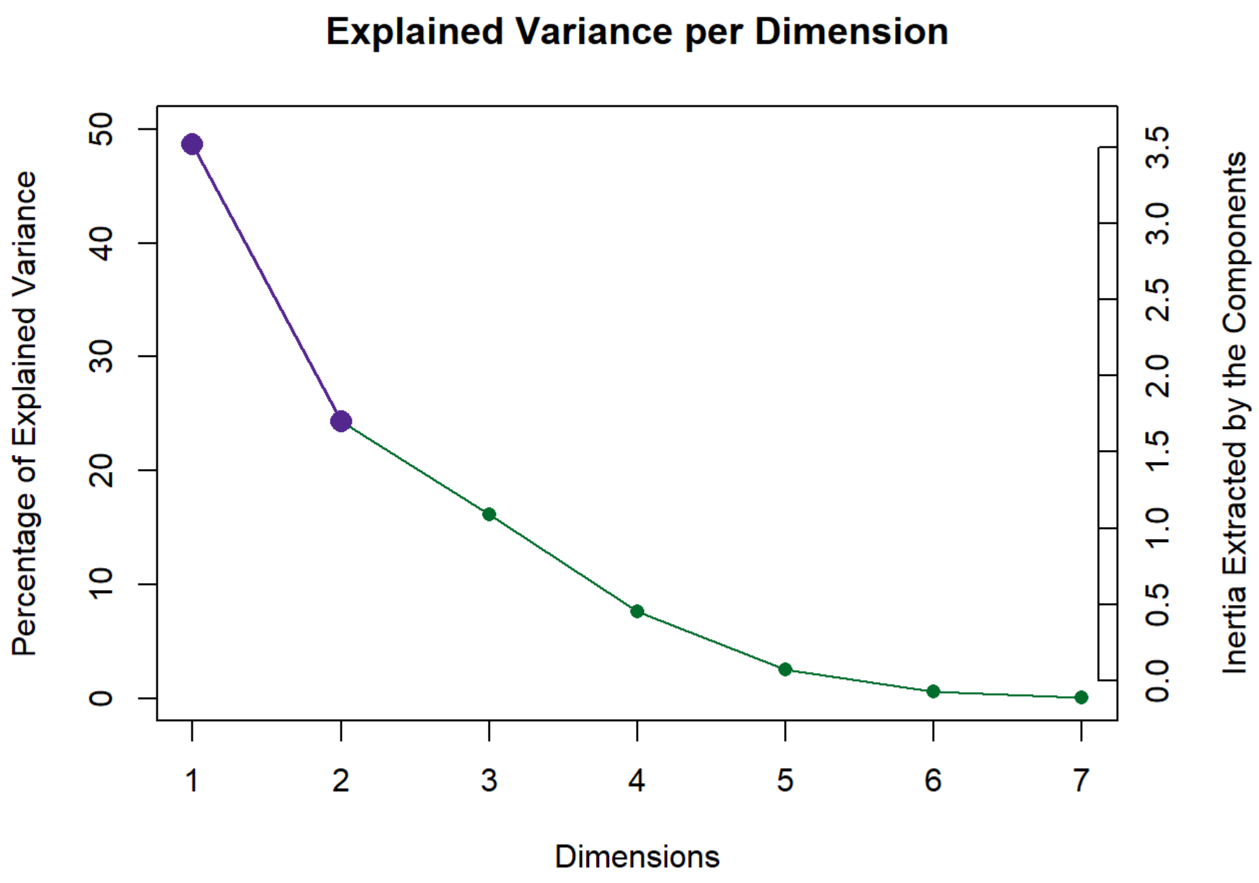
## 2.3 SCREE PLOT

The dimensions are the eigenvectors. The most significant eigenvectors are the ones which explain max variance.

ANALYSIS:

In this case we have dimension 1 and dimension 2 as significant. That means we can reduce our 7 variables to 2.

```
my.scree <- PlotScree(ev = res_pcaInf$Fixed.Data$ExPosition.Data$eigs,
                      p.ev = res_pcaInf$Inference.Data$components$p.vals)
```



```
my.scree <- recordPlot() # you need this line to be able to save them in the end
```

## 2.4 PROJECTIONS:

ROW FACTOR SCORES:

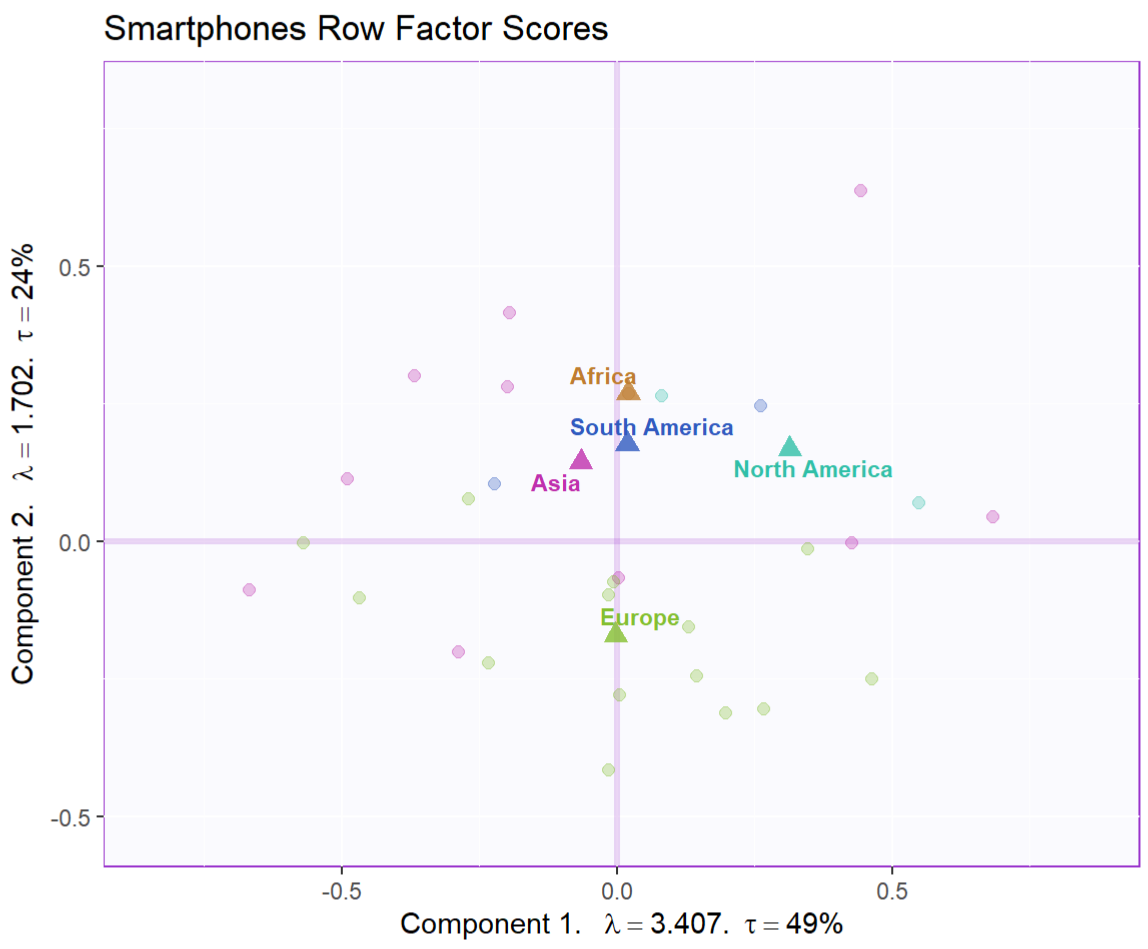
1.Grouping them as per continents

2.Taking the mean of each group

```
my.fi.plot <- createFactorMap(res_pcaInf$Fixed.Data$ExPosition.Data$fi, # data
                            title = "Smartphones Row Factor Scores", # title of the plot
                            axis1 = 1, axis2 = 2, # which component for x and y axes
                            pch = 19, # the shape of the dots (google `pch`)
                            cex = 2, # the size of the dots
                            text.cex = 2.5, # the size of the text
                            alpha.points = 0.3,
                            col.points = res_pcaInf$Fixed.Data$Plotting.Data$fi.col,

                            col.labels = res_pcaInf$Fixed.Data$Plotting.Data$fi.col, # col
                            )


fi.labels <- createxyLabels.gen(1,2,
                             lambda = res_pcaInf$Fixed.Data$ExPosition.Data$eigs,
                             tau = round(res_pcaInf$Fixed.Data$ExPosition.Data$t),
                             axisName = "Component "
                             )
grp.ind <- order(b$Continent)[!duplicated(sort(b$Continent))]
grp.col <- res_pcaInf$Fixed.Data$Plotting.Data$fi.col[grp.ind]

grp.name <- b$Continent[grp.ind] # get the corresponding groups
names(grp.col) <- grp.name
group.mean <- aggregate(res_pcaInf$Fixed.Data$ExPosition.Data$fi,
                     by = list(b$Continent), # must be a list
                     mean)
group.mean
```

```
##         Group.1          V1         V2          V3           V4
## 1        Africa  0.02039262  0.2685105 -0.03366757  0.185166803
## 2          Asia -0.06562383  0.1437742  0.11805739 -0.037917347
## 3        Europe -0.00222700 -0.1709847 -0.03735378 -0.004028255
## 4 North America  0.31432601  0.1678262 -0.02380424  0.055626910
## 5 South America  0.01918582  0.1759409 -0.28817249  0.069574210
##            V5           V6            V7
## 1  0.04351230 -0.012535548 -1.064505e-02
## 2 -0.05336604 -0.003959464  5.786329e-05
## 3  0.02774789  0.006138895  8.618425e-04
## 4  0.07052637  0.009204862  3.538517e-03
## 5 -0.01968751 -0.026112036 -4.538206e-03
```

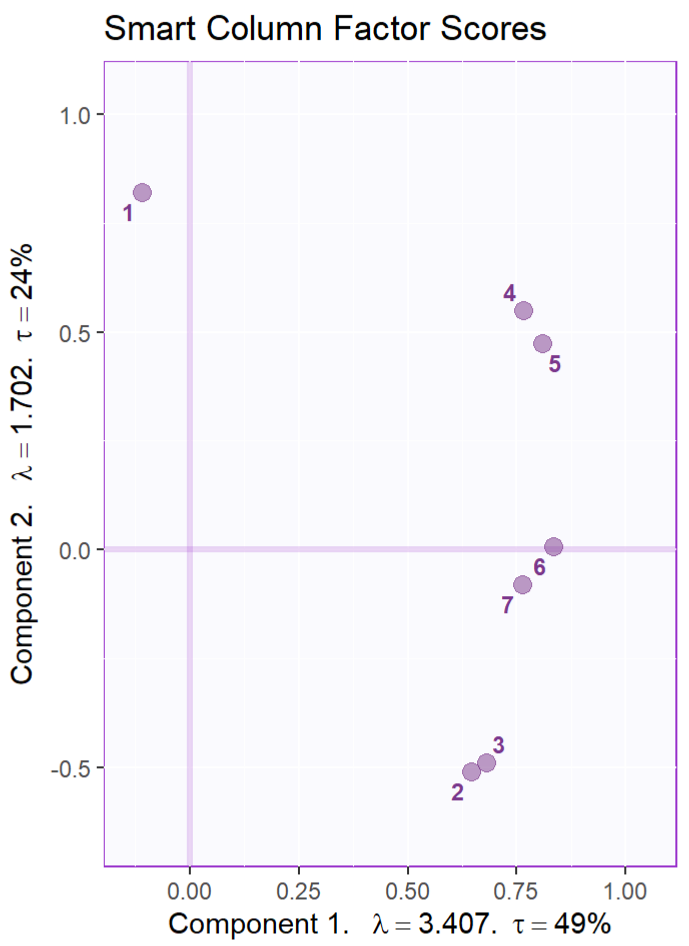
```
# need to format the results from `aggregate` correctly
rownames(group.mean) <- group.mean[,1] # Use the first column as row names
fi.mean <- group.mean[,-1] # Exclude the first column
fi.mean.plot <- createFactorMap(fi.mean,
                                alpha.points = 0.8,
                                col.points = grp.col[rownames(fi.mean)],
                                col.labels = grp.col[rownames(fi.mean)],
                                pch = 17,
                                cex = 3,
                                text.cex = 3)
fi.WithMean <- my.fi.plot$zeMap_background + my.fi.plot$zeMap_dots + fi.mean.plot$zeMap_do
fi.WithMean
```



COLUMNS FACTOR SCORE

Here we will see how the variables are projected on our two dimensions.

```
my.fj.plot <- createFactorMap(res_pcaInf$Fixed.Data$ExPosition.Data$fj, # data
                            title = "Smart Column Factor Scores", # title of the plot
                            axis1 = 1, axis2 = 2, # which component for x and y axes
                            pch = 19, # the shape of the dots (google `pch`)
                            cex = 3, # the size of the dots
                            text.cex = 3, # the size of the text
                            col.points = res_pcaInf$Fixed.Data$Plotting.Data$fj.col, # colo
                            col.labels = res_pcaInf$Fixed.Data$Plotting.Data$fj.col, # colo
                            )

fj.plot <- my.fj.plot$zeMap + fi.labels # you need this line to be able to save them in the
fj.plot
```



## 2.5 LOADINGS

THIS PLOT TELLS US WHICH VARIABLE CONTRIBUTES TO WHICH COMPONENT.

THIS PLOT WILL SHOW THE CORRELATION ANGLES BETWEEN VARIABLES

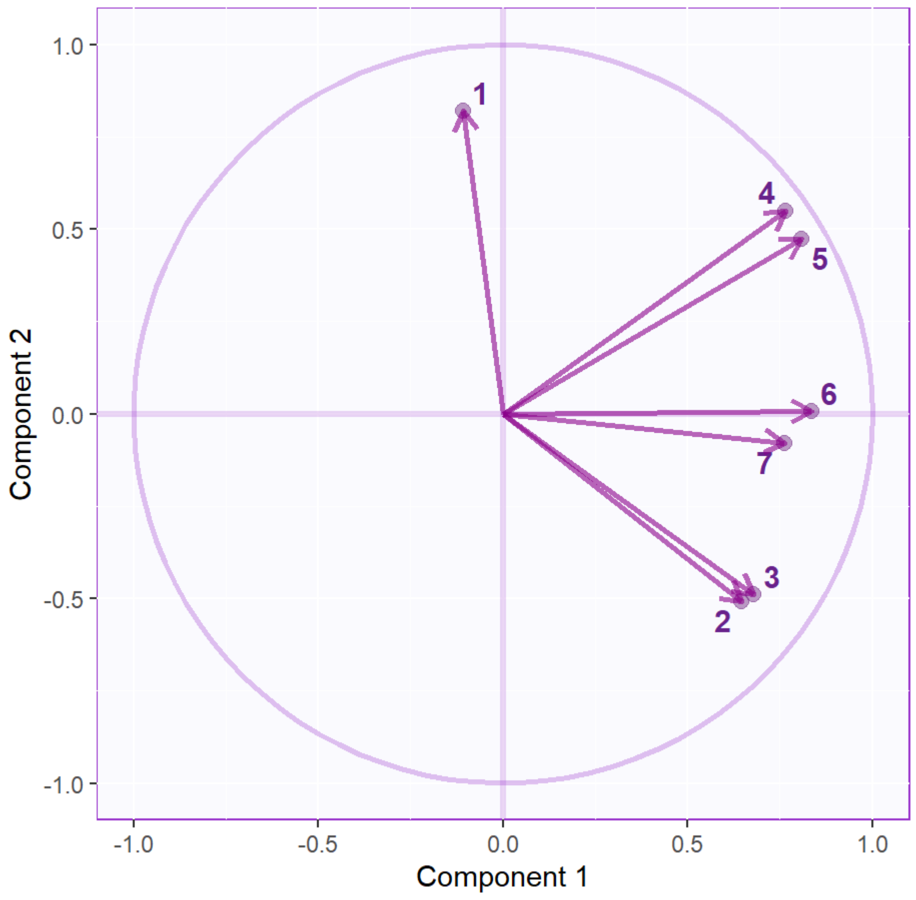
COSINE OF THE ANGLES BETWEEN THE VARIABLES WILL SHOW US THE CORRELATION

THE MORE CLOSER THE ARROWS TO THE CRICLE THE MORE CORRECT ARE OUR INTERPRETATION

AS THEY WOULD CONTRIBUTE MORE TOWARDS THE COMPONENTS

```
library(InPosition)
res_pcaInf <- epPCA.inference.battery(a, center = TRUE, scale = "SS1", DESIGN = b$Continent
## [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
## ===========================================================================
```

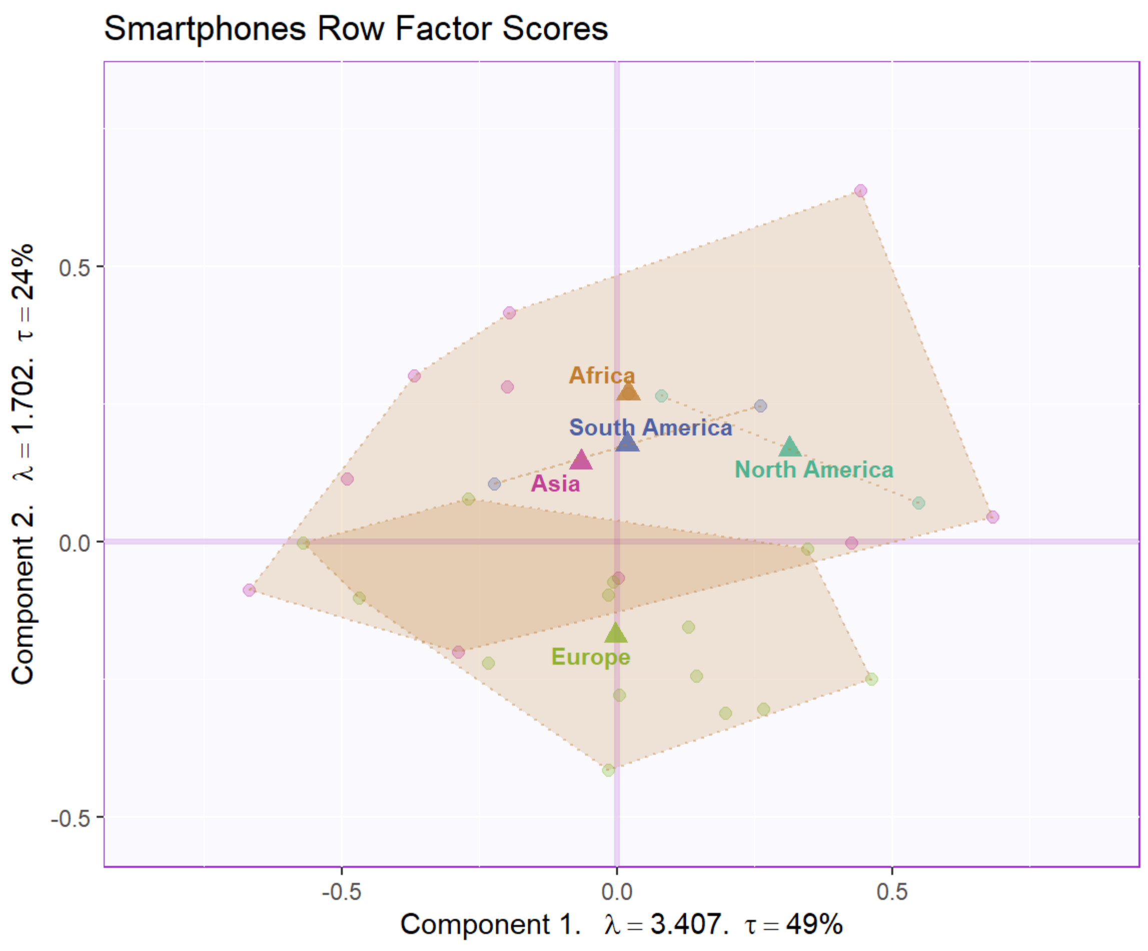
```
cor.loading <- cor(a, res_pcaInf$Fixed.Data$ExPosition.Data$fi)
rownames(cor.loading) <- rownames(cor.loading)

loading.plot <- createFactorMap(cor.loading,
                                constraints = list(minx = -1, miny = -1,
                                                   maxx = 1, maxy = 1),
                                col.points = res_pcaInf$Fixed.Data$Plotting.Data$fj.col)
LoadingMapWithCircles <- loading.plot$zeMap +
  addArrows(cor.loading, color = "magenta4") +
  addCircleOfCor() + xlab("Component 1") + ylab("Component 2")

LoadingMapWithCircles
```



## 2.6 TOLERANCE INTERVALS

The countries which overlap are shown by tolerance intervals

```
TIplot <- MakeToleranceIntervals(res_pcaInf$Fixed.Data$ExPosition.Data$fi[which(b$Continent
                            design = as.factor(b$Continent)
                            [which(b$Continent!="Africa")],
                            # line below is needed
                            names.of.factors =  c("Dim1","Dim2"), # needed
                            col = grp.col[rownames(fi.mean)[which(b$Continent!="Africa")]],
                            line.size = .50,
                            line.type = 3,
                            alpha.ellipse = .2,
                            alpha.line    = .4,
                            p.level       = .95)

fi.WithMeanTI <- my.fi.plot$zeMap_background + my.fi.plot$zeMap_dots + fi.mean.plot$zeMap_

fi.WithMeanTI
```



## 2.7 BOOTSTRAP MEANS TEST

By using the bootstrap methods we can find the confidence intervals for the means

```
fi.boot <- Boot4Mean(res_pcaInf$Fixed.Data$ExPosition.Data$fi[which(b$Continent!="Africa"),
                     design = b$Continent[which(b$Continent!="Africa")],
                     niter = 1000)
# Check what you have
fi.boot
```

```
## ----------------------------------------------------------------------------
## Bootstrapped Means K groups, J Variables, L iterations
## ----------------------------------------------------------------------------
## $BootCube                 An K*J*L brick of Bootstrapped means
## $GroupMeans               The K*J table of means
## $BootstrappedGroupMeans   The K*J table of means of BootCube
## ----------------------------------------------------------------------------
```

```
# What is the cube? Check the first 4 tables
fi.boot$BootCube[,,1:4]
```

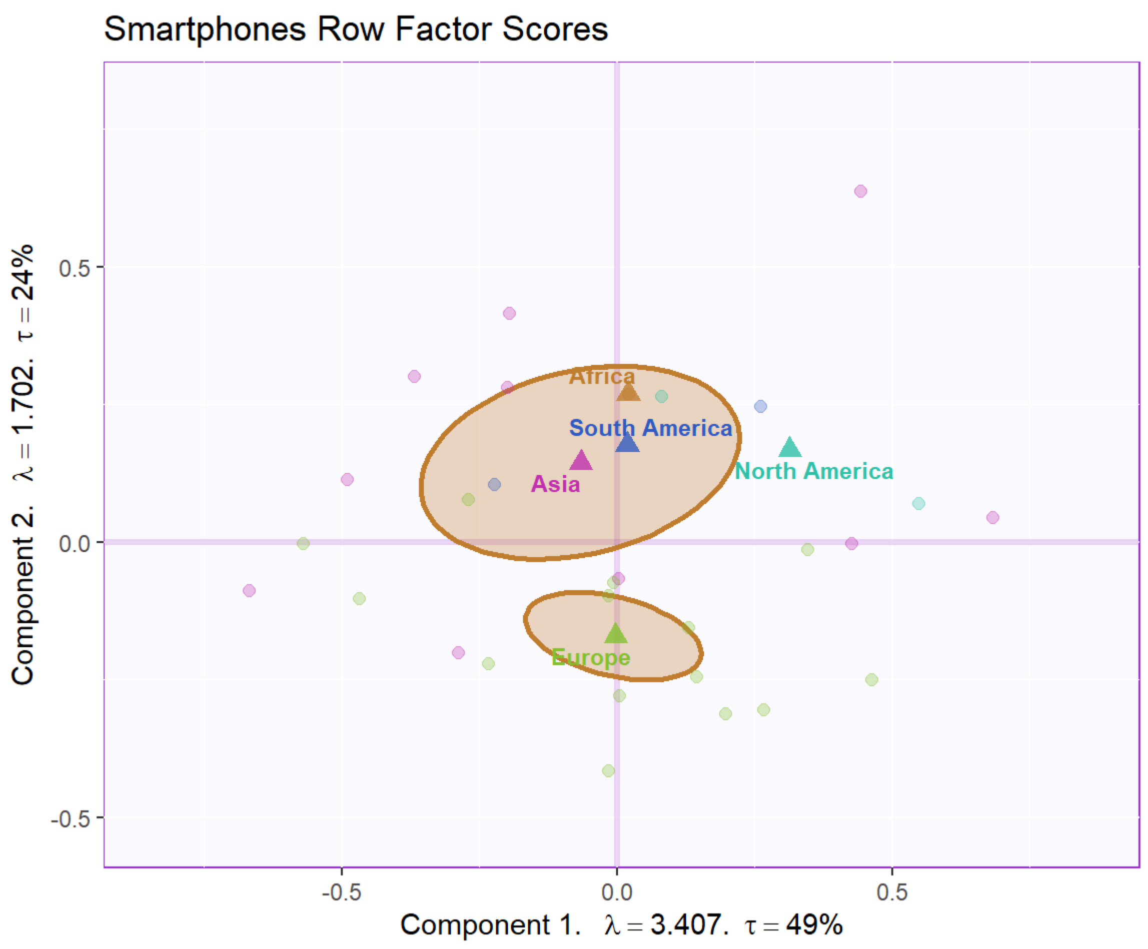
```
## , , 1
##
##                         V-1        V-2         V-3          V-4         V-5
## Asia          -0.01852490  0.1848747  0.03204814 -0.032768061 -0.05452249
## Europe         0.05236686 -0.2338193 -0.02850470 -0.009406332  0.02968075
## North America  0.07983957  0.2648039  0.05711067  0.146567905  0.07387766
## South America -0.22304790  0.1049689 -0.25196212  0.016287208 -0.03993022
##                           V-6          V-7
## Asia          -0.0003369321 -0.001636611
## Europe         0.0062632734 -0.001538622
## North America  0.0123230138 -0.010284617
## South America  0.0038226906  0.009579663
##
## , , 2
##
##                         V-1        V-2         V-3         V-4           V-5
## Asia          -0.05234835  0.1831150  0.04995422 -0.08429833 -0.0472155634
## Europe         0.10415454 -0.2407964 -0.03118989  0.01894700  0.0456419400
## North America  0.31432601  0.1678262 -0.02380424  0.05562691  0.0705263703
## South America  0.26141954  0.2469128 -0.32438286  0.12286121  0.0005551992
##                          V-6          V-7
## Asia          -0.013910415 -0.006023370
## Europe         0.010875497 -0.003416561
## North America  0.009204862  0.003538517
## South America -0.056046763 -0.018656075
##
## , , 3
##
##                         V-1        V-2          V-3         V-4
## Asia          -0.12625279  0.1037437  0.132911395 -0.04414489
## Europe         0.05169058 -0.1935741  0.005643654  0.02946683
## North America  0.07983957  0.2648039  0.057110670  0.14656791
## South America  0.26141954  0.2469128 -0.324382857  0.12286121
##                           V-5         V-6          V-7
## Asia          -0.0513647999  0.01210579 -0.002237133
## Europe         0.0424645205  0.01469234 -0.004331914
## North America  0.0738776573  0.01232301 -0.010284617
## South America  0.0005551992 -0.05604676 -0.018656075
##
## , , 4
##
##                       V-1         V-2         V-3         V-4         V-5
## Asia          -0.27319032  0.03889063  0.09459653 -0.03339403 -0.06347919
## Europe         0.10464416 -0.15324863  0.02443076 -0.04554067  0.02832824
## North America  0.31432601  0.16782615 -0.02380424  0.05562691  0.07052637
## South America  0.01918582  0.17594089 -0.28817249  0.06957421 -0.01968751
##                         V-6          V-7
## Asia          -0.0127076737  0.006366172
## Europe         0.0003158173 -0.003507011
## North America  0.0092048624  0.003538517
## South America -0.0261120364 -0.004538206
```

```
bootCI4mean <- MakeCIEllipses(fi.boot$BootCube[,c(1:2),], # get the first two components
                              col = grp.col[rownames(fi.mean)])

fi.WithMeanCI <- my.fi.plot$zeMap_background + bootCI4mean + my.fi.plot$zeMap_dots + fi.mea
fi.WithMeanCI
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```

```
## Warning: Computation failed in `stat_ellipse()`:
## the leading minor of order 2 is not positive definite
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
## Warning: Computation failed in `stat_ellipse()`:
## the leading minor of order 2 is not positive definite
```



## 2.8 ANALYSIS/HYPOTHESIS

Europe is different from rest of the continents in terms of purchase decision and usage of smartphones.

LOADINGS:

1 IS NOT CORRELATED TO 5 AND 6 IT IS INVERSELY CORRELATED TO 2 AND 3

4 AND 5 CORRELATED

2 AND 3 CORRELATED

6 AND 7 CORRELATED

4 AND 5 ARE NOT CORRELATED TO 2 AND 3

CORPLOT:

ANALYSIS

1 is not significantly correlated to any other variables.

2 is correlated to 3

4 is correlated to 5

6 is correlated to 7

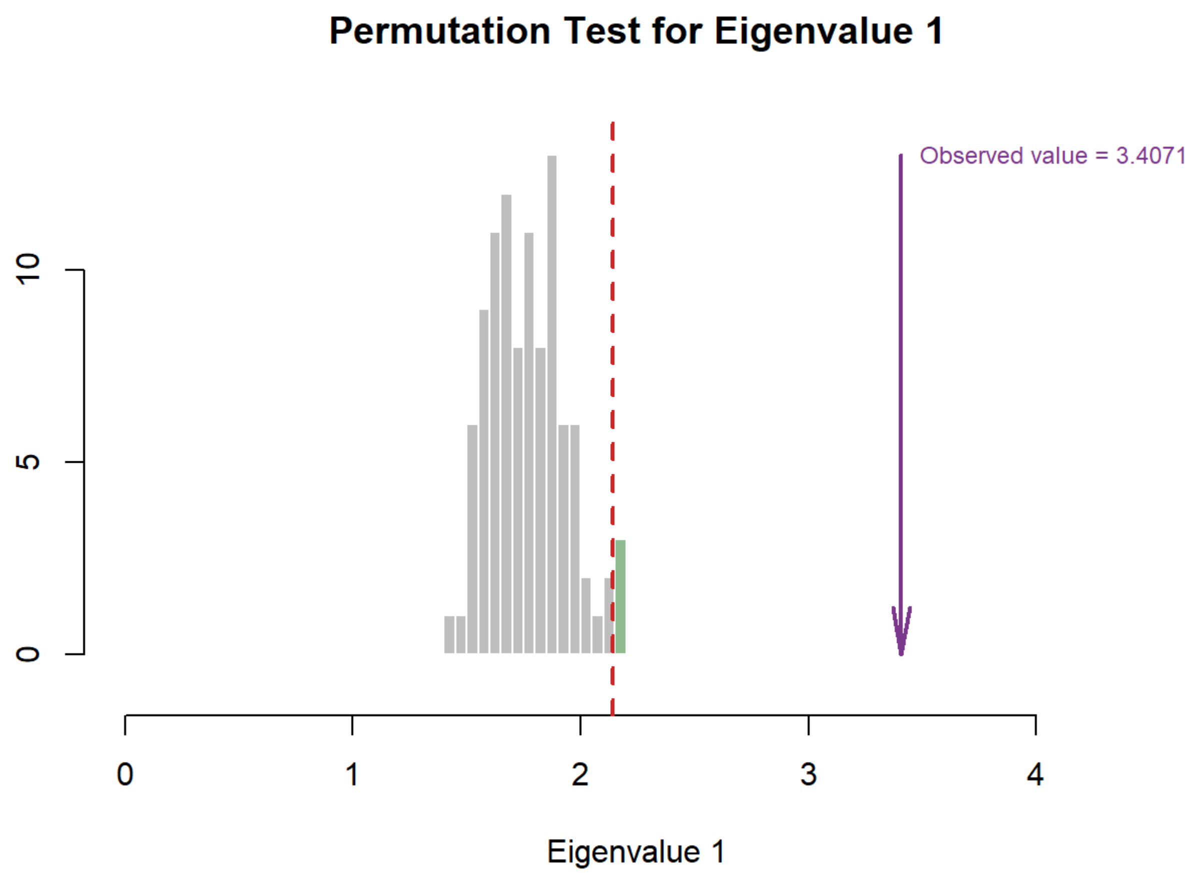
NOTE (Please check the dataset description for understanding which variable is assigned to which number.)

## 2.9 SCREE PERMUTATION TEST

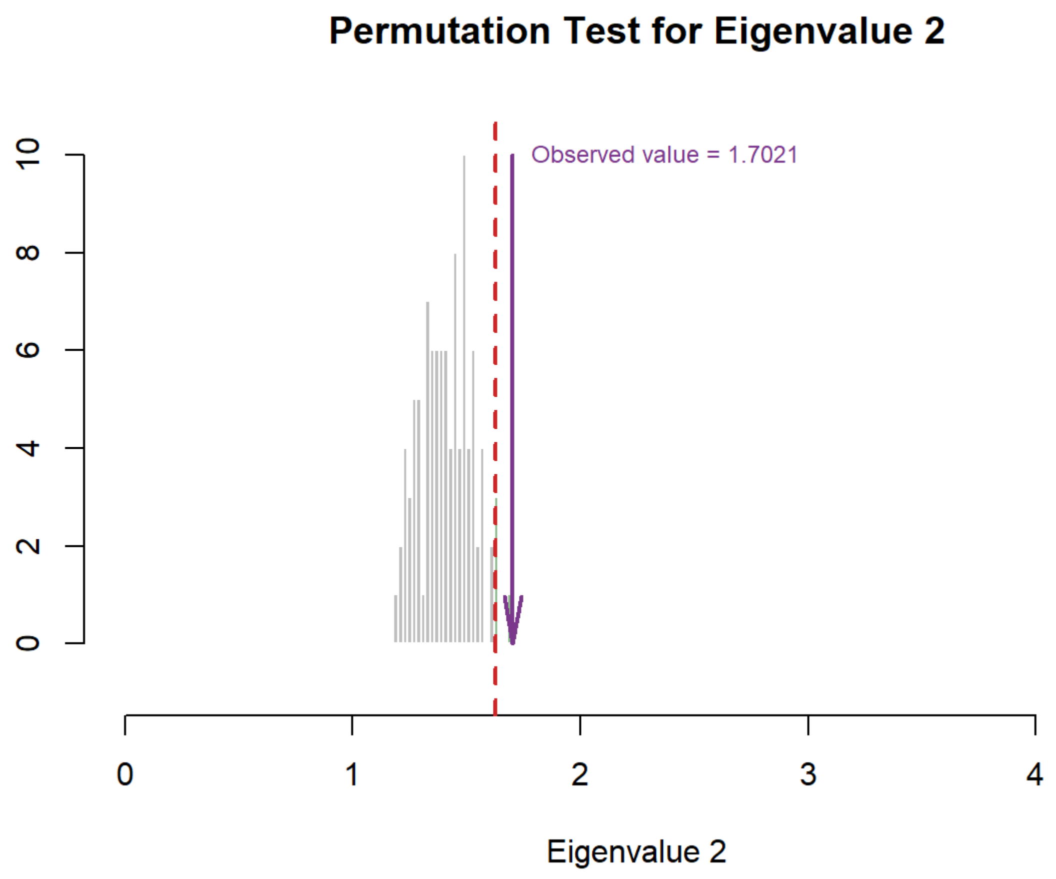
The components 1 and 2 are reliable according to the permutation test.

We can say this by rejecting the null hypothesis.

```
zeDim = 1
pH1 <- prettyHist(
  distribution = res_pcaInf$Inference.Data$components$eigs.perm[,zeDim],
           observed = res_pcaInf$Fixed.Data$ExPosition.Data$eigs[zeDim],
           xlim = c(0, 4.5), # needs to be set by hand
           breaks = 20,
           border = "white",
           main = paste0("Permutation Test for Eigenvalue ",zeDim),
           xlab = paste0("Eigenvalue ",zeDim),
           ylab = "",
           counts = FALSE,
           cutoffs = c( 0.975))
```



```
eigs1 <- recordPlot()

zeDim = 2

pH2 <- pH1 <- prettyHist(

  distribution = res_pcaInf$Inference.Data$components$eigs.perm[,zeDim],

           observed = res_pcaInf$Fixed.Data$ExPosition.Data$eigs[zeDim],

           xlim = c(0, 4.5), # needs to be set by hand

           breaks = 20,

           border = "white",

           main = paste0("Permutation Test for Eigenvalue ",zeDim),

           xlab = paste0("Eigenvalue ",zeDim),

           ylab = "",

           counts = FALSE,

           cutoffs = c(0.975))
```



```
eigs2 <- recordPlot()
```

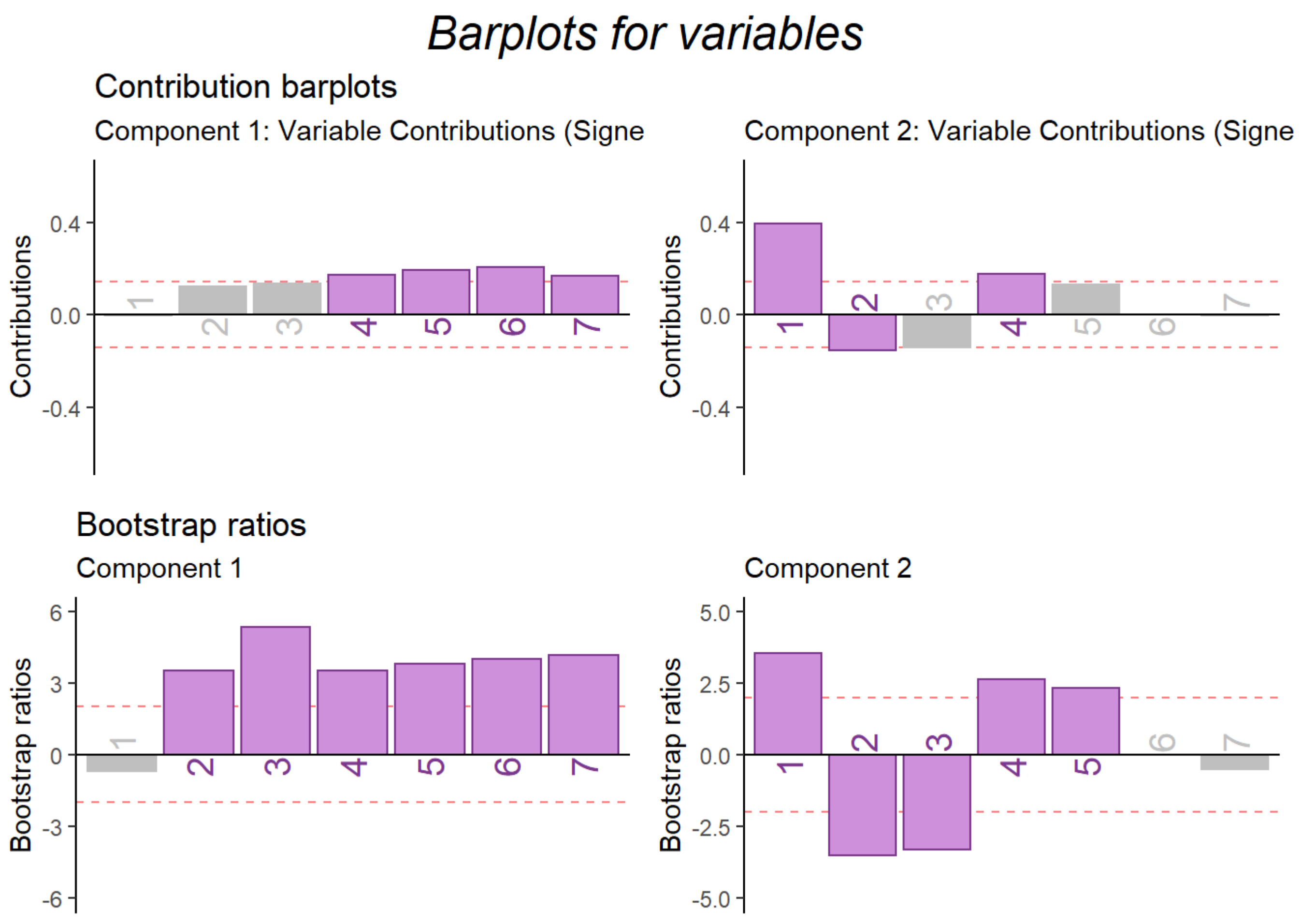
## 2.10 BOOTSTRAP TEST FOR CONTRIBUTIONS

The contributions of variables towards the two components are significant according to the bootstrap test.

Thereby we can consider our interpretations reliable.

```r
signed.ctrJ <- res_pcaInf$Fixed.Data$ExPosition.Data$cj * sign(res_pcaInf$Fixed.Data$ExPosi

# plot contributions for component 1
ctrJ.1 <- PrettyBarPlot2(signed.ctrJ[,1],
                         threshold = 1 / NROW(signed.ctrJ),
                         font.size = 5,
                         color4bar = gplots::col2hex(res_pcaInf$Fixed.Data$Plotting.Data$fj
                         ylab = 'Contributions',
                         ylim = c(1.2*min(signed.ctrJ), 1.2*max(signed.ctrJ))
) + ggtitle("Contribution barplots", subtitle = 'Component 1: Variable Contributions (Signe

# plot contributions for component 2
ctrJ.2 <- PrettyBarPlot2(signed.ctrJ[,2],
                         threshold = 1 / NROW(signed.ctrJ),
                         font.size = 5,
                         color4bar = gplots::col2hex(res_pcaInf$Fixed.Data$Plotting.Data$fj
                         ylab = 'Contributions',
                         ylim = c(1.2*min(signed.ctrJ), 1.2*max(signed.ctrJ))
) + ggtitle("",subtitle = 'Component 2: Variable Contributions (Signed)')



BR <- res_pcaInf$Inference.Data$fj.boots$tests$boot.ratios
laDim = 1

# Plot the bootstrap ratios for Dimension 1
ba001.BR1 <- PrettyBarPlot2(BR[,laDim],
                        threshold = 2,
                        font.size = 5,
                   color4bar = gplots::col2hex(res_pcaInf$Fixed.Data$Plotting.Data$fj.col),
                  ylab = 'Bootstrap ratios'
                  #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("Bootstrap ratios", subtitle = paste0('Component ', laDim))

# Plot the bootstrap ratios for Dimension 2
laDim = 2
```

```
ba002.BR2 <- PrettyBarPlot2(BR[,laDim],
                      threshold = 2,
                      font.size = 5,
                 color4bar = gplots::col2hex(res_pcaInf$Fixed.Data$Plotting.Data$fj.col),
                ylab = 'Bootstrap ratios'
                #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("",subtitle = paste0('Component ', laDim))
grid.arrange(
    as.grob(ctrJ.1),
    as.grob(ctrJ.2),
    as.grob(ba001.BR1),
    as.grob(ba002.BR2),
    ncol = 2,nrow = 2,
    top = textGrob("Barplots for variables", gp = gpar(fontsize = 18, font = 3))
  )
```



```r
BothCtrJ <- recordPlot() # you need this line to be able to save them in the end
```