Research Methods 3 Multivariate Analysis using R

Thouseef Syed

11/16/2019

Contents

Chapter 1

Research Methods that were implemented to perform Multivariate Analysis

1.1 1. Principal Component Analysis (PCA)

1.2 2. Barycentric Discriminant Analysis (BADA)

1.3 3. Multiple Component Analysis (MCA)

1.4 4. Discriminant Correspondence Analysis (DiCA)

1.5 5. Partial Least Squares Correlation (PLSC)

1.6 6. Multiple Factor Analysis (MFA)

1.7 7. Correspondence Analysis (CA)

1.8 8. DiSTATIS

Chapter 2

Principal Component Analysis

This file contains the cookbook for analysing on how decisions can be influenced by past judgements and has 4 sections:

Method - It highlights how each method works and the type(s) of data is can analyze

Data Set - It introduce the data set to be analyzed

Results - It gives a *concise* walk-through and interpretation of the analysis and results.

Summary - It briefly ties things together.

2.1 Method: PCA

Principal component analysis (PCA) is used to analyze one table of quantitative data. PCA mixes the input variables to give new variables, called principal components. The first principal component is the line of best fit. It is the line that maximizes the inertia (similar to variance) of the cloud of data points. Subsequent components are defined as orthogonal to previous components, and maximize the remaining inertia.

PCA gives one map for the rows (called factor scores), and one map for the columns (called loadings). These 2 maps are related, because they both are described by the same components. However, these 2 maps project different kinds of information onto the components, and so they are *interpreted differently*. Factor scores are the coordinates of the row observations. They are interpreted by the distances between them, and their distance from the origin. Loadings describe the column variables. Loadings are interpreted by the angle between them, and their distance from the origin.

The distance from the origin is important in both maps, because squared distance from the mean is inertia (variance, information; see sum of squares as in ANOVA/regression). Because of the Pythagorean Theorem, the total information contributed by a data point (its squared distance to the origin) is also equal to the sum of its squared factor scores. ## Data set: Drive.RData Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test. It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

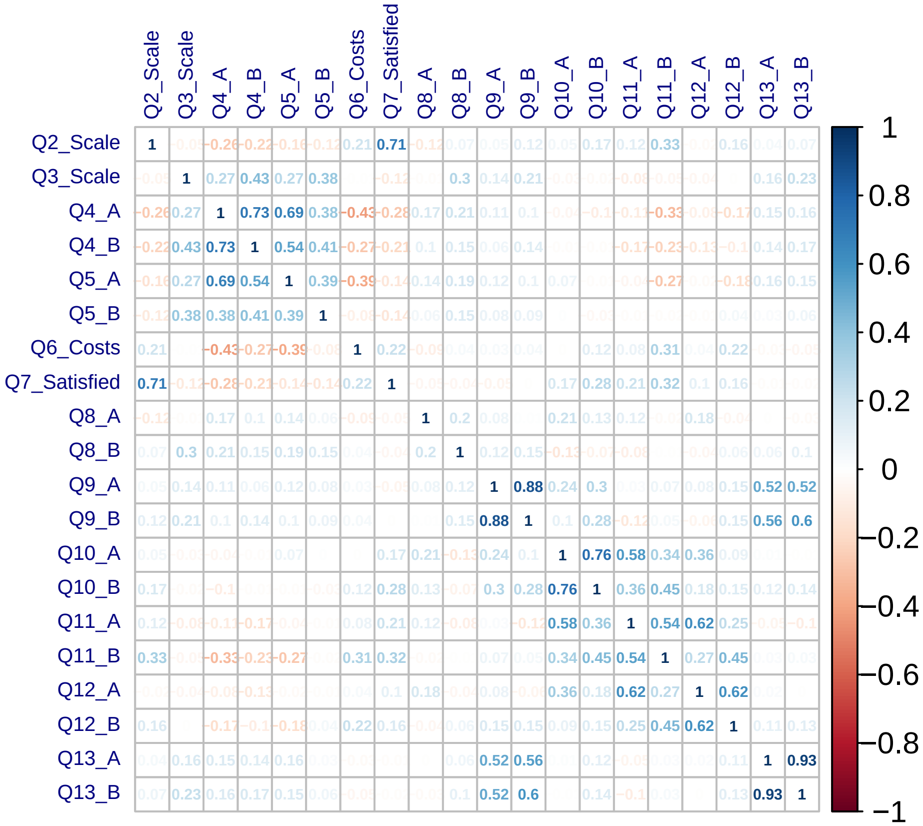
- Q_1: Implementation Decision
- Q2_Scale: Implementation Decision & judgement
- Q3_Scale: Road Distance
- Q4_A: Current Average Speed (A)
- Q4_B: Current Average Speed (B)
- Q5_B: Speed after reconstruction (B)
- Q6_Costs: Cost
- Q7_Satisfied: Satisfied with implementation and judgement
- Q8_A: Travel Time Saved (A)
- Q8_B: Travel Time Saved (B)
- Q9_A: Importance of Road Distance (A)
- Q9_B: Importance of Road Distance (B)
- Q10_A: Importance of speed before increase (A)
- Q10_B: Importance of speed before increase (B)
- Q11_A: Importance of speed after increase (A)
- Q11_B: Importance of speed after increase (B)
- Q12_A: Importance of time saving (A)
- Q12_B: Importance of time saving (B)
- Q13_A: Importance of Costs (A)
- Q13_B: Importance of Costs (B)

```
##    Q2_Scale Q3_Scale  Q4_A  Q4_B  Q5_A  Q5_B Q6_Costs Q7_Satisfied  Q8_A
## 1     88.91    34.44  8.95 21.79 17.32 45.91    68.48        92.80 19.46
## 2     97.28    32.49 15.76 22.57 54.86 59.73   100.00        93.39 13.23
## 3     81.32    43.77  8.56 20.62 19.65 49.81    66.54        89.49 81.91
## 4     89.49    25.10  3.11 13.81  8.37 33.07    65.56        85.80 66.54
## 5     76.26    50.19  4.28 17.70 14.20 41.05    25.68        50.19 22.76
## 6     70.04    33.85  7.39 17.32 16.34 47.86    66.54        67.90 25.10
##    Q8_B  Q9_A  Q9_B  Q10_A  Q10_B  Q11_A  Q11_B  Q12_A  Q12_B Q13_A Q13_B
## 1 25.10 16.15 15.95 100.00  24.51 100.00  25.49 100.00  36.77 48.64 48.83
## 2 42.22 44.75 44.75  19.07  19.65  75.10  86.58  35.60  55.84  3.89  5.45
## 3 62.06 87.55 87.35 100.00 100.00 100.00 100.00 100.00 100.00  0.00  0.00
## 4 22.37  0.78  0.78  92.02 100.00  96.89 100.00   0.58   0.97  1.56  0.78
## 5 50.58 26.07 74.71  34.24  39.69  47.28  51.56  30.35  62.45 35.02 57.98
## 6 42.41 49.42 49.42  72.57  57.39  80.74  56.42  67.70  64.59 50.97 51.36
```

2.2 The Correlation plot

Correlogram is a graph of correlation matrix. It is very useful to highlight the most correlated variables in a data table. In this plot, correlation coefficients is colored according to the value. Correlation matrix can be also reordered according to the degree of association between variables. In this case, it is observed that there is high correlation between Q4_A & Q4_B (Current Average Speed of both roads),Q5_A & Q5_B (Speed after reconstruction). Also, there is strong correlation between Q9_A & Q9_B (Importance of road distance), Q10_A & Q10_B (Importance of Speed before increase of both roads),Q11_A & Q11_B (Importance of Speed before after of both roads) and Q12_A & Q12_B (Importance of total time saved)

```
cor.res <- cor(data.pca2)
corrplot(cor.res,method="number" ,tl.cex = 0.7, tl.col = "navyblue",cl.cex = 1,number.cex=0.5)
```



```
cor.plot <- recordPlot() # you need this line to be able to save the figure to PPT lat
```

2.3 Analysis

Since, each variable is measured on different units, the columns were scaled and centered. The rows are color-coded by the DESIGN variable, data.choice.

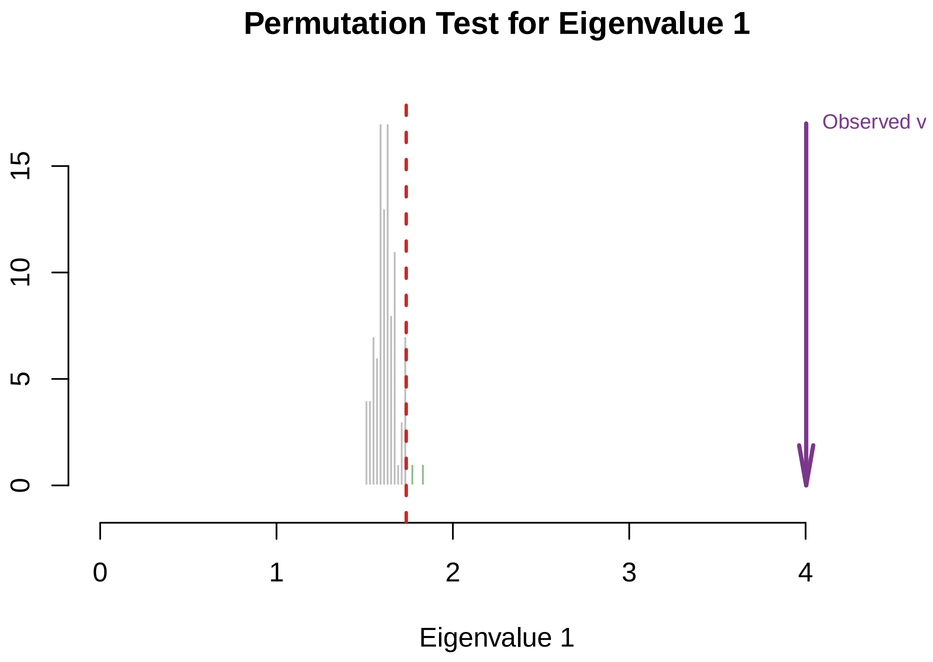
- `center = TRUE`: substracts the mean from each column
- `scale = TRUE`: after centering (or not), scales each column to have a sum of squares of 1 (see the help for different scaling options)
- `DESIGN`: colors the observations (rows)
- `graphs = FALSE`: this gives you plots from `epPCA`, but make sure to flag it `FALSE` for Rmarkdown to run correctly

```
res_pcaInf <- epPCA.inference.battery(data.pca2, center = TRUE, scale = "SS1", DESIGN =
## [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 note
## ============================================================================
```

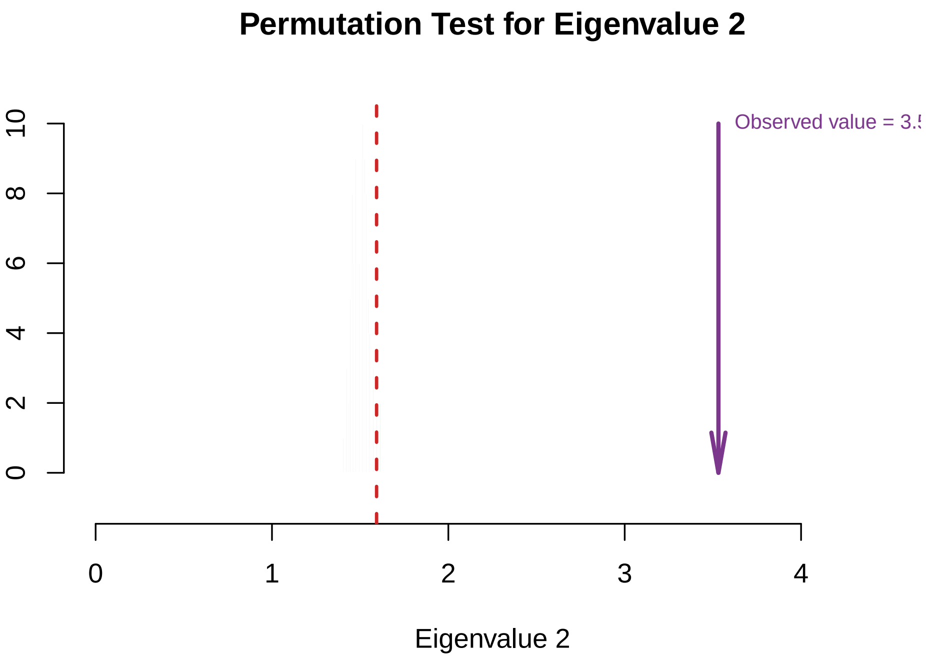
2.4 Inference PCA:

2.5 Testing the eigenvalues

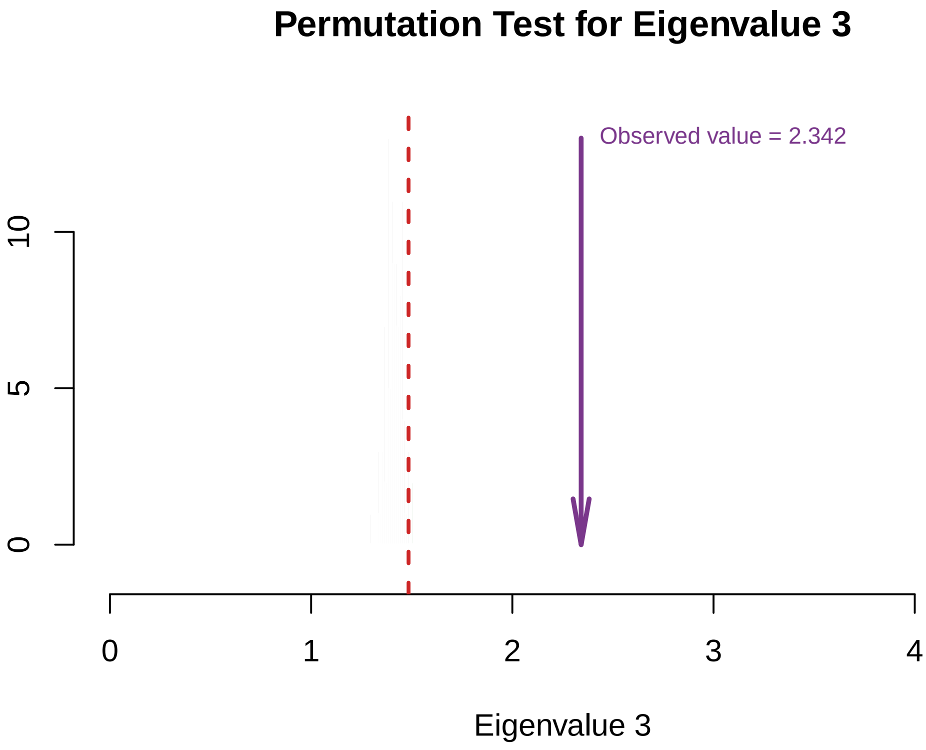
```
zeDim = 1
pH1 <- prettyHist(
  distribution = res_pcaInf$Inference.Data$components$eigs.perm[,zeDim],
           observed = res_pcaInf$Fixed.Data$ExPosition.Data$eigs[zeDim],
           xlim = c(0, 4.5), # needs to be set by hand
           breaks = 20,
           border = "white",
           main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



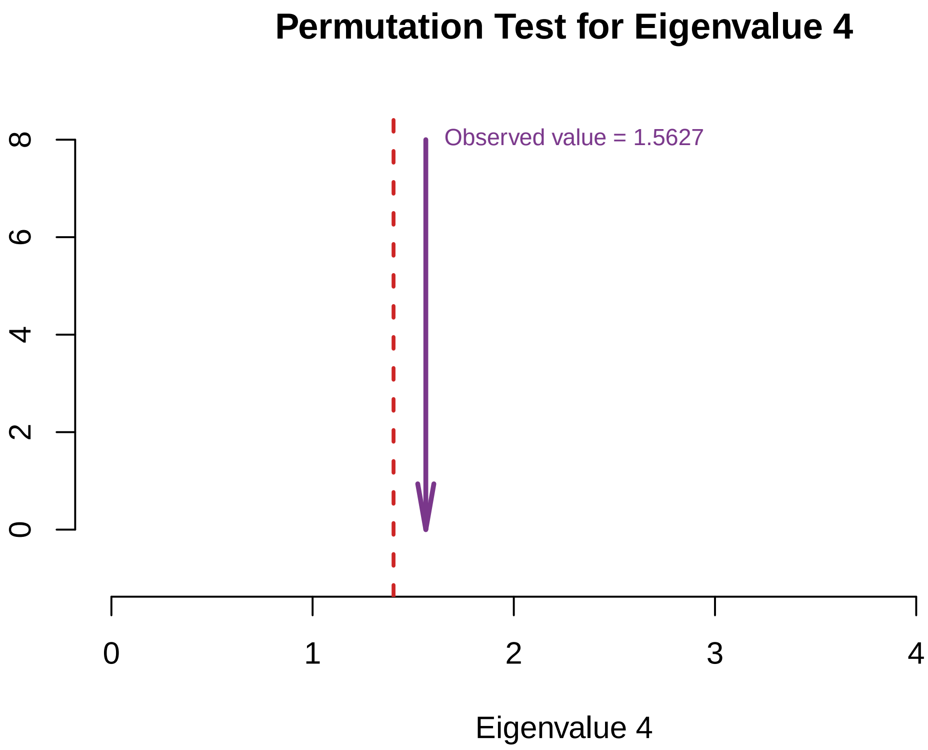
```
eigs1 <- recordPlot()

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))
```



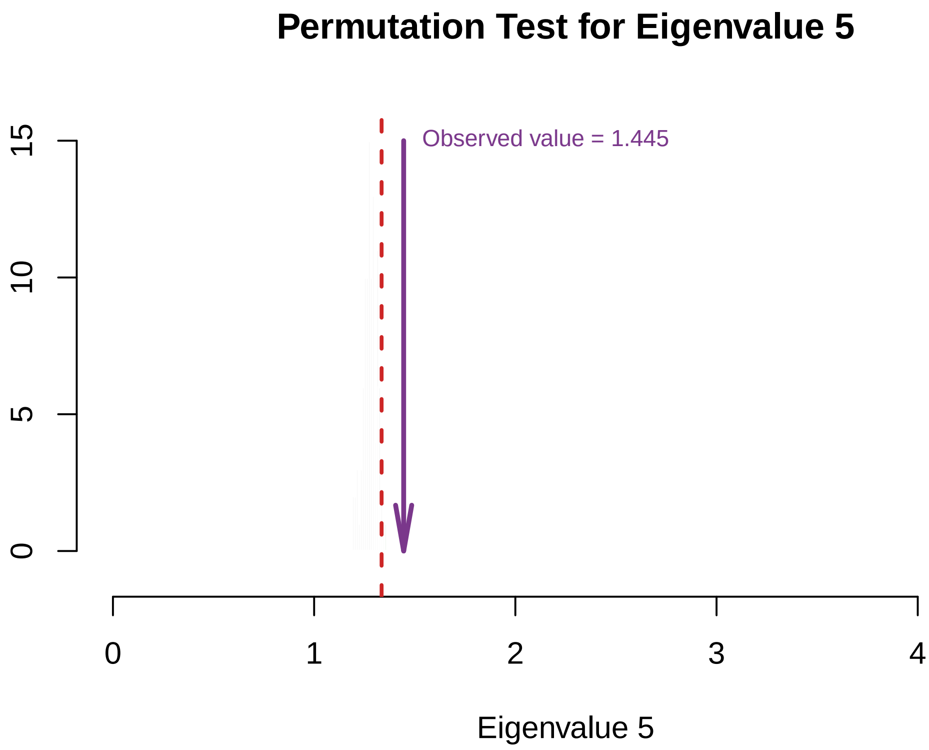
```
eigs2 <- recordPlot()

zeDim = 3
pH3 <- 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))
```



```
eigs3 <- recordPlot()

zeDim = 4
pH4 <- pH3 <- 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))
```



```
eigs4 <- recordPlot()

zeDim = 5
ph5 <- pH4 <- pH3 <- 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))
```

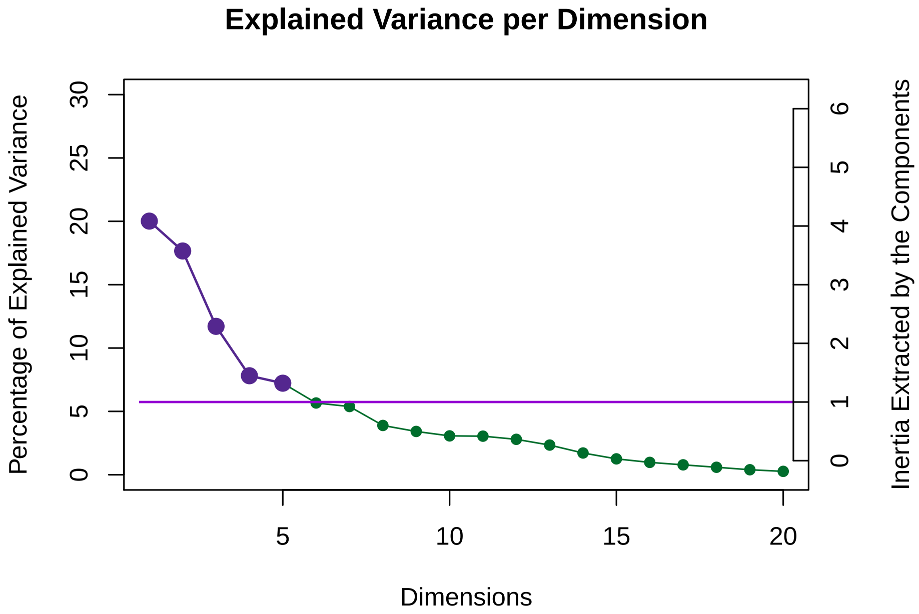


```
eigs5 <- recordPlot()
```

2.6 Scree Plot

The scree plot shows the eigenvalues, the amount of information on each component. The number of components (the dimensionality of the factor space) is min(nrow(DATA), ncol(DATA)) minus 1. According to the scree plot about 5 components/dimensions must be interpreted.

```
my.scree <- PlotScree(ev = res_pcaInf$Fixed.Data$ExPosition.Data$eigs,
                      p.ev = res_pcaInf$Inference.Data$components$p.vals,plotKaiser = TRUE,color4
```

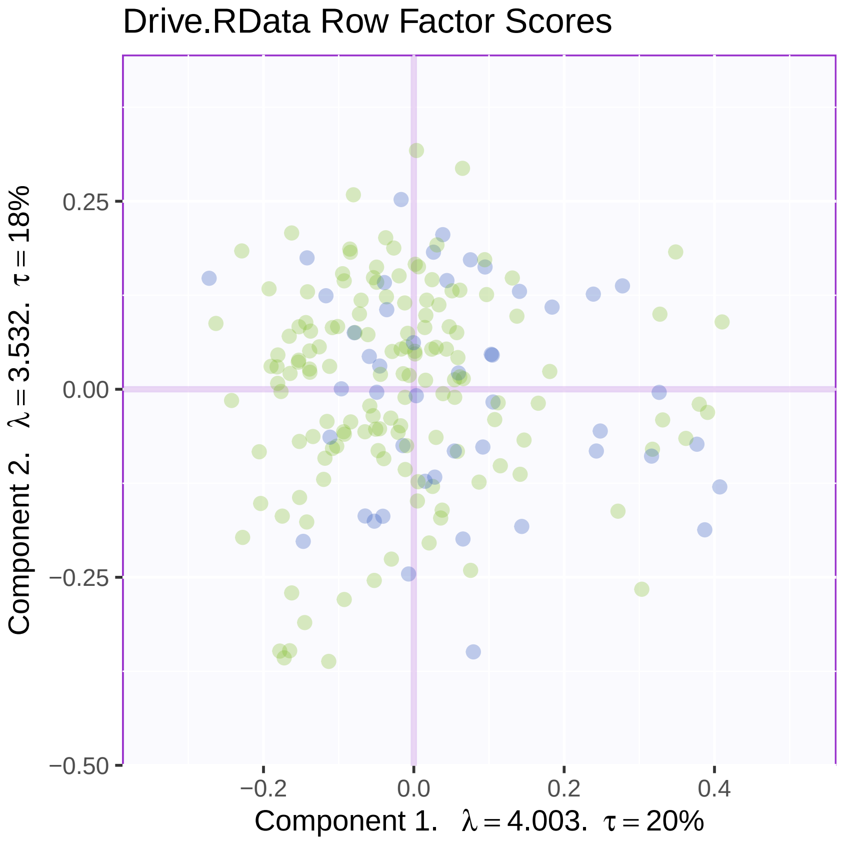


```
my.scree <- recordPlot() # you need this line to be able to save them in the end
```

2.7 Factor scores

Factor scores are the coordinates of the 191 participant's choices on the components. The distances between them show which participants had similar choices(Choice 1/Choice 2) . Factor scores (choices) can be color-coded to help interpret the components. Choice 1 corresponds to Group A and Choice 2 corresponds to Group B.

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

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

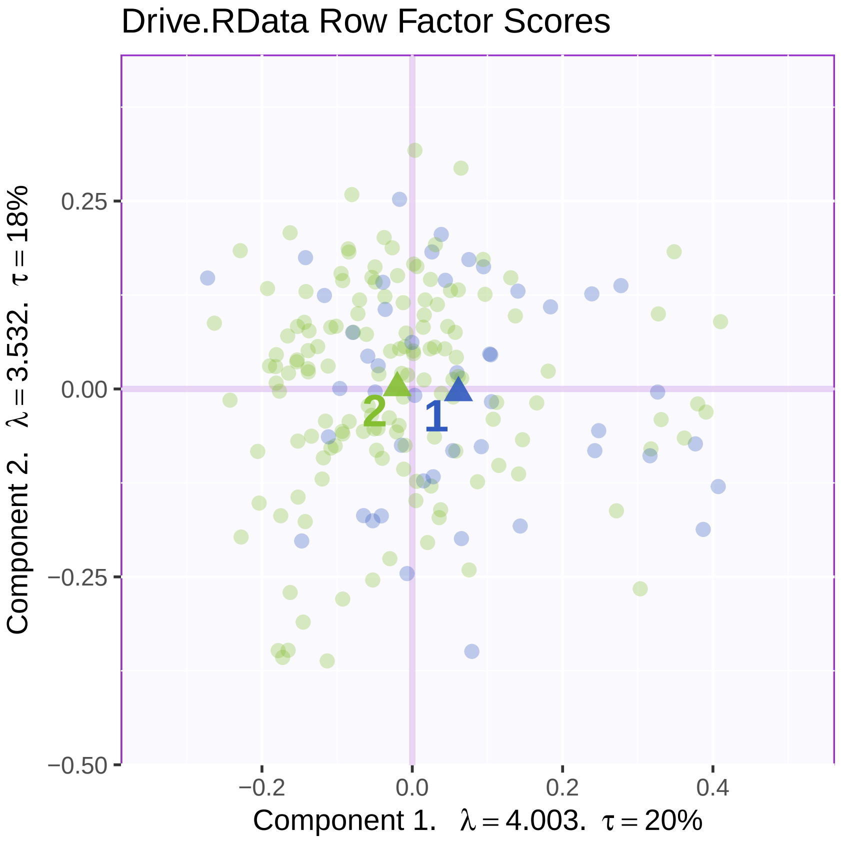


Obtain the color for each group:

2.8 With group means

Plot them!

```
fi.mean.plot <- createFactorMap(fi.mean,
                                alpha.points = 0.9,
                                col.points = grp.col[rownames(fi.mean)],
                                col.labels = grp.col[rownames(fi.mean)],
                                pch = 17,
                                       cex = 3.5,
                                       text.cex = 6)
fi.WithMean <- my.fi.plot$zeMap_background + my.fi.plot$zeMap_dots + fi.mean.plot$zeMap
fi.WithMean
```



*

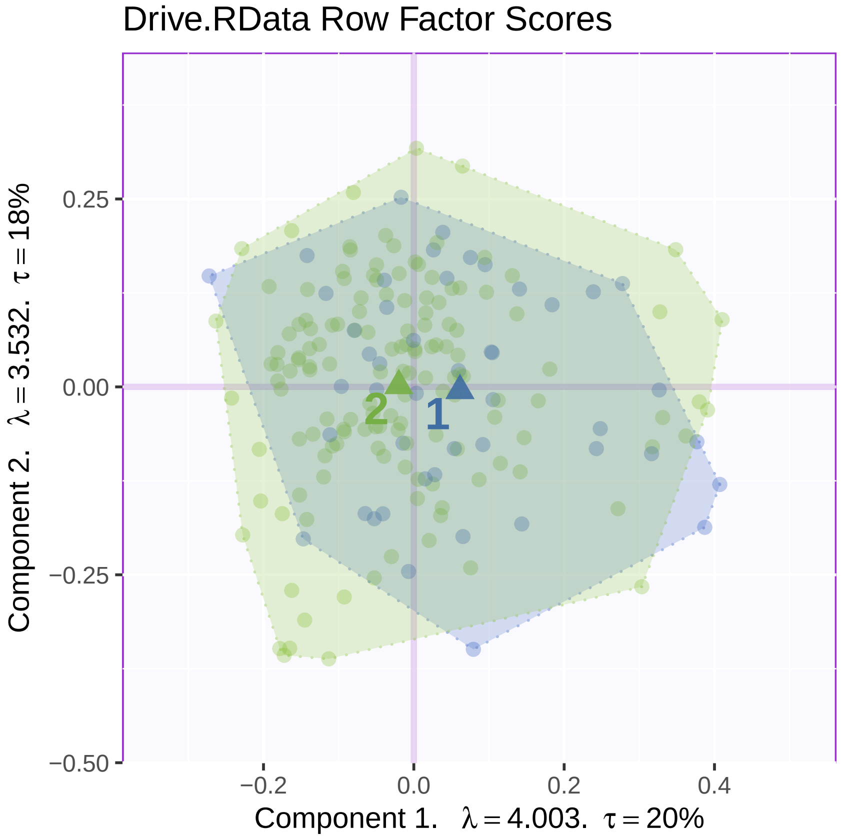
Component 1: 39 & 165 VS 49 & 156 (All Group B) * Component 2: 118(Group A) & 34(Group A) VS 145(Group B) & 77(Group B)

- Choice 1 (Group A) selected by the participants is indicated by the color Blue (Current Avg. Speed=15mph & Speed after reconstruction =29mph)
- Choice 2 (Group B)selected by the participants is indicated by the color Green (Current Avg. Speed=35mph & Speed after reconstruction =71mph)

2.9 Tolerance interval

We can plot the tolerance interval

```
TIplot <- MakeToleranceIntervals(res_pcaInf$Fixed.Data$ExPosition.Data$fi,
                            design = as.factor(data.choice),
                            # line below is needed
                            names.of.factors =  c("Dim1","Dim2"), # needed
                            col = grp.col[rownames(fi.mean)],
                            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_dots +

fi.WithMeanTI
```



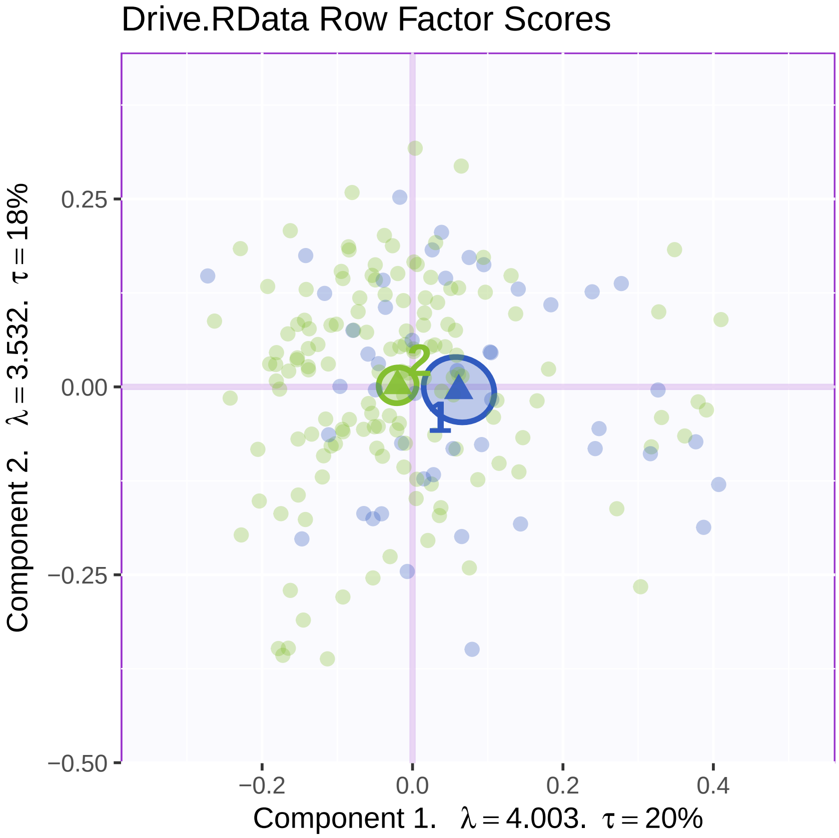
2.10 Bootstrap interval

We can also add the bootstrap interval for the group means to see if these group means are significantly different.

First step: bootstrap the group means

Second step: plot it!

```
# Check other parameters you can change for this function
bootCI4mean <- MakeCIEllipses(fi.boot$BootCube[,c(1:2),], # get the first two componen
                              col = grp.col[rownames(fi.mean)])

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

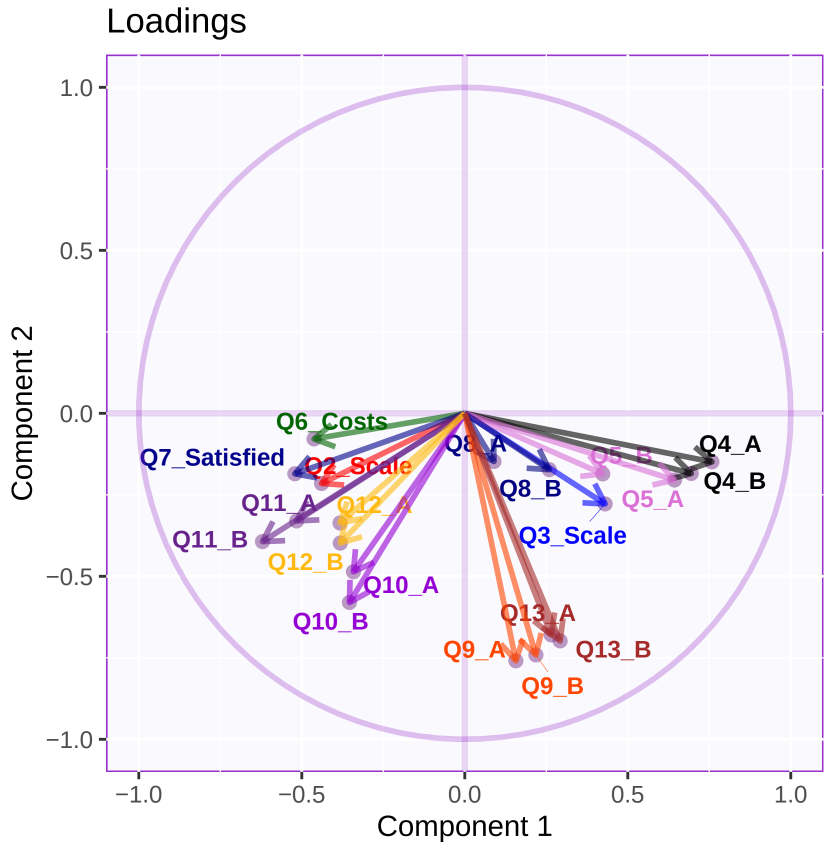


2.11 Loadings

Loadings describe the similarity (angular distance) between the variables. Here the input variables are the inputs received from the participants. Loadings show how the input variables relate to each other. Loadings also show which variables are important for (which components load on) a certain component.

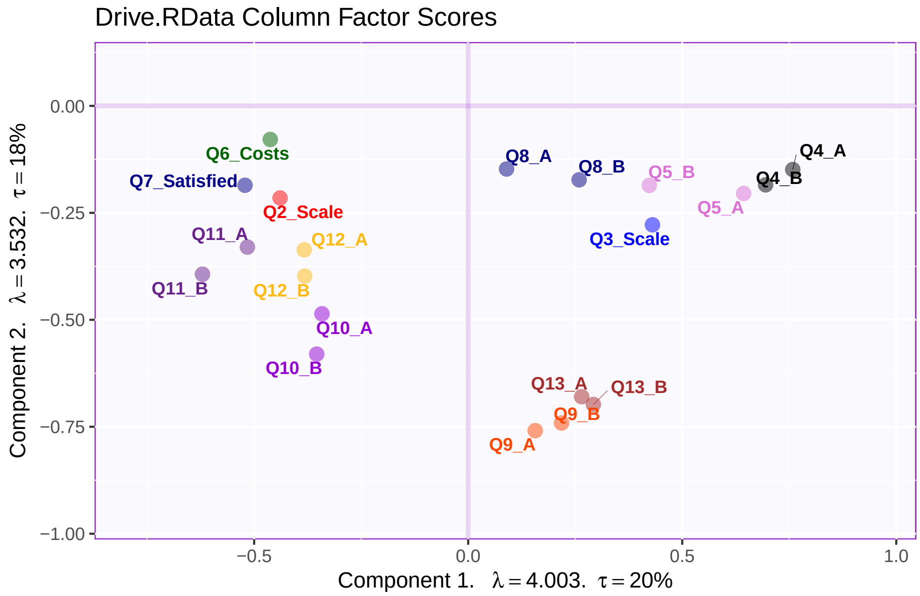
The color codes indicate the individual questions for both Group A and Group B. Both the questions relating to both the roads(A & B) are represented in the same color. Therefore, it is evident that the individual questions for both the roads (A&B) are positively correlated.

```
cor.loading <- t(cor(data.pca2, res_pcaInf$Fixed.Data$ExPosition.Data$fi))
rownames(cor.loading) <- colnames(cor.loading)
label.color2 <- c("red", "blue", "black",  "black","orchid","orchid","darkgreen","blue4","navyblu
loading.plot <- createFactorMap(t(cor.loading),
                                constraints = list(minx = -1, miny = -1,
                                                   maxx = 1, maxy = 1),
                                col.points = res_pcaInf$Fixed.Data$Plotting.Data$fj.col,col.label
LoadingMapWithCircles <- loading.plot$zeMap +
  addArrows(t(cor.loading), color = label.color2) +
  addCircleOfCor() + xlab("Component 1") + ylab("Component 2")

LoadingMapWithCircles
```



You can also include the variance of each component and plot the factor scores for the columns (i.e., the variables):

```
my.fj.plot <- createFactorMap(res_pcaInf$Fixed.Data$ExPosition.Data$fj, # data
                              title = "Drive.RData Column Factor Scores", # title of the
                              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 = label.color2, # color of the dots
                              col.labels = label.color2, # color for labels of dots
                              )

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



- Component 1: Q11_B VS Q4_A (Importance of speed after increase VS Current Average Speed)
- Component 2: Q9_A (Importance of Road Distance)

2.12 Bootstrap Ratio of columns

**Note: This is not the same as the contribution bars

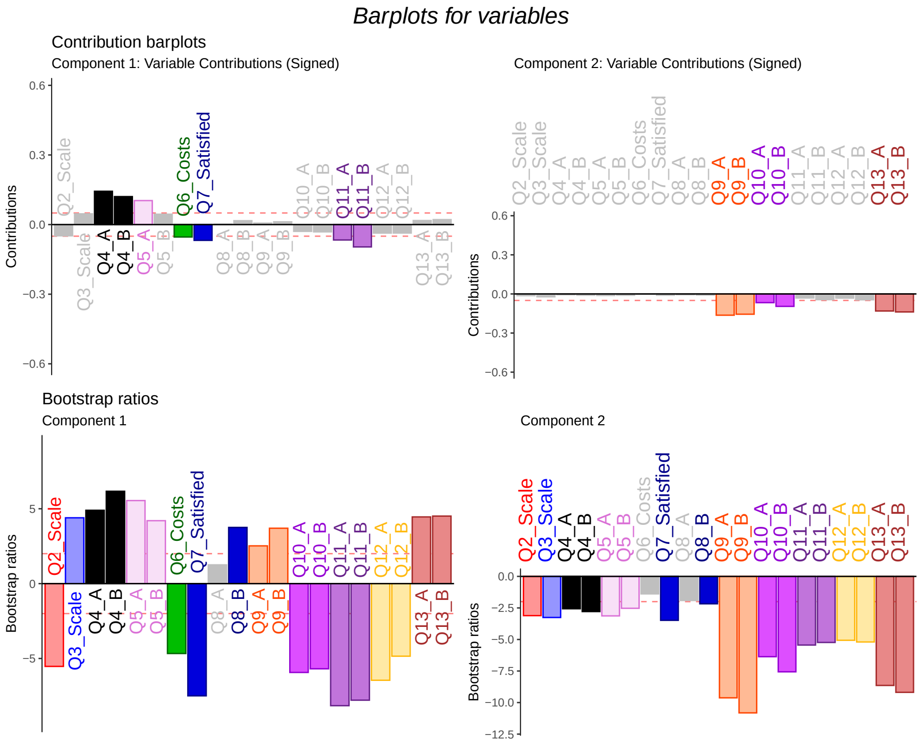
2.13 Component 1

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

#res_pcaInf$Fixed.Data$Plotting.Data$fj.col

# plot contributions for component 1
ctrJ.1 <- PrettyBarPlot2(signed.ctrJ[,1],
                         threshold = 1 / NROW(signed.ctrJ),
                         font.size = 5,
                         color4bar = gplots::col2hex(label.color2), # we need hex code
                         ylab = 'Contributions',
                         ylim = c(1.2*min(signed.ctrJ), 1.2*max(signed.ctrJ))
```

```
) + ggtitle("Contribution barplots", subtitle = 'Component 1: Variable Contributions (S

# plot contributions for component 2
ctrJ.2 <- PrettyBarPlot2(signed.ctrJ[,2],
                         threshold = 1 / NROW(signed.ctrJ),
                         font.size = 5,
                         color4bar = gplots::col2hex(label.color2), # we need hex code
                         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(label.color2), # we need hex code
                  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(label.color2), # we need hex code
                  ylab = 'Bootstrap ratios'
                  #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("",subtitle = paste0('Component ', laDim))
```

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

```
  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))
  )
```



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

2.14 Summary

When the factor scores and loadings were interpretted, the PCA revealed:

- Component 1: Participants of Group A and Group B overestimated the speed after reconstruction
- Component 2: Majority of Group A & Group B underestimated the

```
a) Importance of the road distance
b) Cost for reconstructing the road.
```

Inference: Based on the Bootstrap ratios :-

Component 1: It is evident that the Current Average Speed (Q4_A & Q4_B), Speed after reconstruction (Q5_A & Q5_B), Importance of speed before and after increase (Q10_A,Q10_B,Q11_A & Q11_B) and Importance of Cost(Q13_A & Q13_B) are signficant for the analysis.

Component 2: It is evident that the Importance of speed before and after increase (Q10_A,Q10_B,Q11_A & Q11_B) and Importance of Cost(Q13_A & Q13_B) are signficant for the analysis.

Based on the Inference:

Component 1: It is evident that they also underestimated the Importance of speed before increase and overestimated the importance of cost.

Component 2: It is evident that they also underestimated Importance of speed before and after increase. Also, the Importance of total time saved.

Chapter 3

BADA

3.1 Method BADA on PCA: Note on PCA

Principal component analysis (PCA) is used to analyze one table of quantitative data. PCA mixes the input variables to give new variables, called principal components. The first principal component is the line of best fit. It is the line that maximizes the inertia (similar to variance) of the cloud of data points. Subsequent components are defined as orthogonal to previous components, and maximize the remaining inertia.

PCA gives one map for the rows (called factor scores), and one map for the columns (called loadings). These 2 maps are related, because they both are described by the same components. However, these 2 maps project different kinds of information onto the components, and so they are *interpreted differently*. Factor scores are the coordinates of the row observations. They are interpreted by the distances between them, and their distance from the origin. Loadings describe the column variables. Loadings are interpreted by the angle between them, and their distance from the origin.

The distance from the origin is important in both maps, because squared distance from the mean is inertia (variance, information; see sum of squares as in ANOVA/regression). Because of the Pythagorean Theorem, the total information contributed by a data point (its squared distance to the origin) is also equal to the sum of its squared factor scores.

3.2 BADA (Barycentric Discriminant Analysis)

Barycentric discriminant analysis (BADA) is a robust version of discriminant analysis that is used to assign, to pre-defined groups (also called categories), observations described by multiple variables. By contrast with traditional discriminant analysis, BADA can be used even when the number of observations is smaller than the number of variables—This makes BADA particularly suited for the analysis of Big Data.

3.3 Data set: Drive.RData

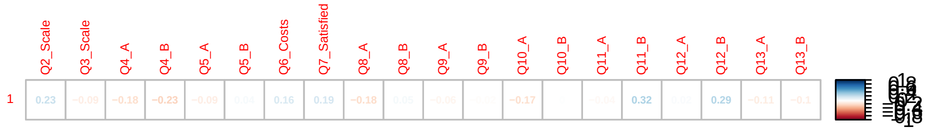
Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test. It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

- Q_1: Implementation Decision
- Q2_Scale: Implementation Decision & judgement
- Q3_Scale: Road Distance
- Q4_A: Current Average Speed (A)
- Q4_B: Current Average Speed (B)
- Q5_A: Speed after reconstruction (A)
- Q5_B: Speed after reconstruction (B)
- Q6_Costs: Cost
- Q7_Satisfied: Satisfied with implementation and judgement
- Q8_A: Travel Time Saved (A)
- Q8_B: Travel Time Saved (B)
- Q9_A: Importance of Road Distance (A)
- Q9_B: Importance of Road Distance (B)
- Q10_A: Importance of speed before increase (A)
- Q10_B: Importance of speed before increase (B)
- Q11_A: Importance of speed after increase (A)
- Q11_B: Importance of speed after increase (B)
- Q12_A: Importance of time saving (A)
- Q12_B: Importance of time saving (B)
- Q13_A: Importance of Costs (A)
- Q13_B: Importance of Costs (B)

3.4 Heatmap

The Heatmap indicates that many participants mostly answered the question on Q7_Satisfied :Satisfied with implementation and judgement,Q2_Scale: Implementation Decision & judgement,Q12_B: Importance of time saving (B),Q11_B: Importance of speed after increase (B).

```
# Heatmap
Design <- as.numeric(data.choice[1:191])
 X <- as.data.frame(data.pca2[1:191,1:20])
# heatmap(t(Design%*%X))

heatmap.cor <- cor(Design,X)
# Plot it with corrplot
corrplot((heatmap.cor), method = "number",tl.cex = 0.5,number.cex = 0.4, cl.cex = 0.8)
```

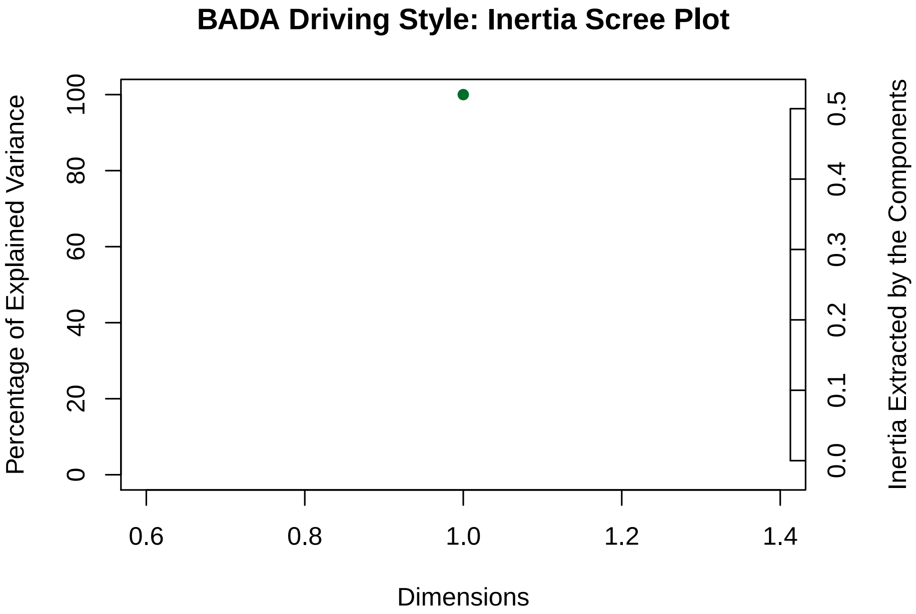


3.5 Run BADA

3.6 Scree Plot

The scree plot shows the eigenvalues, the amount of information on each component. The number of components (the dimensionality of the factor space) is min(nrow(DATA), ncol(DATA)) minus 1. According to the scree plot about 1 components must be interpreted.

```
PlotScree(ev = resBADA$TExPosition.Data$eigs,
          title = 'BADA Driving Style: Inertia Scree Plot',
          plotKaiser = FALSE,
          color4Kaiser = ggplot2::alpha('darkorchid4', .5),
          lwd4Kaiser  = 2)
```



```
# Save the plot
a0002.Scree.sv <- recordPlot()
```

3.7 Observations

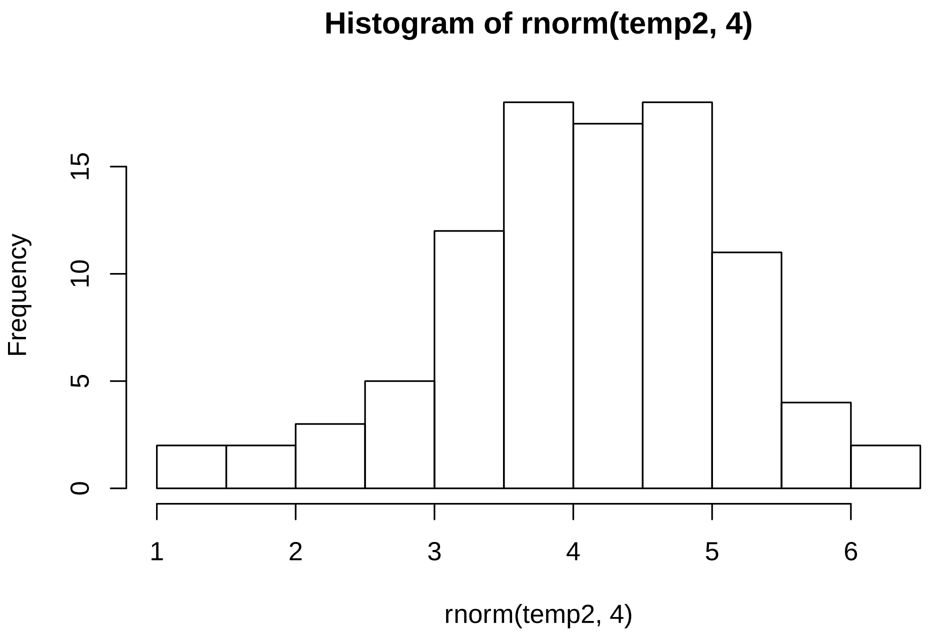
Factor scores are the coordinates of the 191 participant's choices on the components. The distances between them show which participants had similar choices(Choice 1/Choice 2) . Factor scores (choices) can be color-coded to help interpret the components. Choice 1 corresponds to Group A and Choice 2 corresponds to Group B.

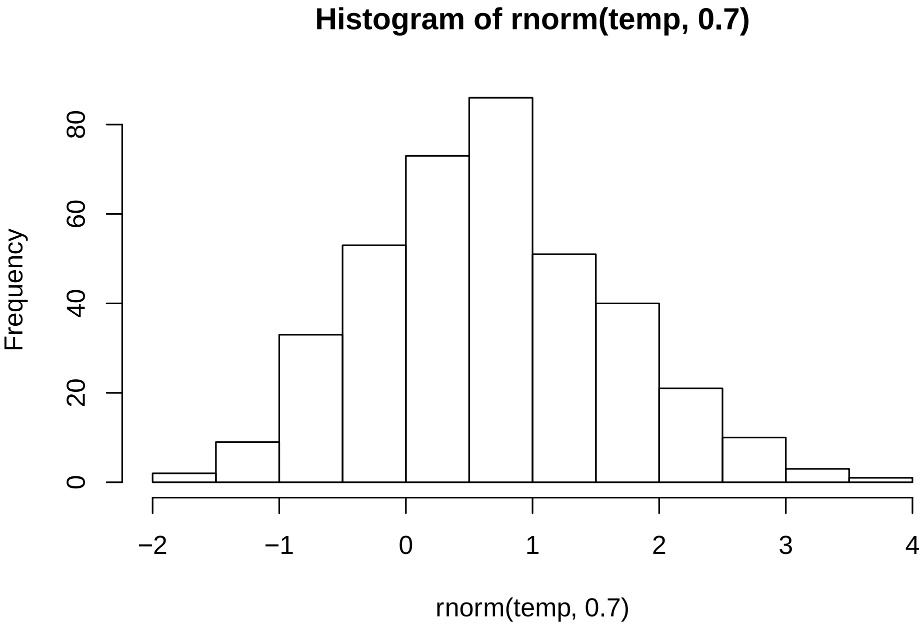
The below histograms indicates the distribution of participants who selected choice 1 (indicated in blue) corresponds to Group A and choice 2 (indicated in green) corresponds to Group B

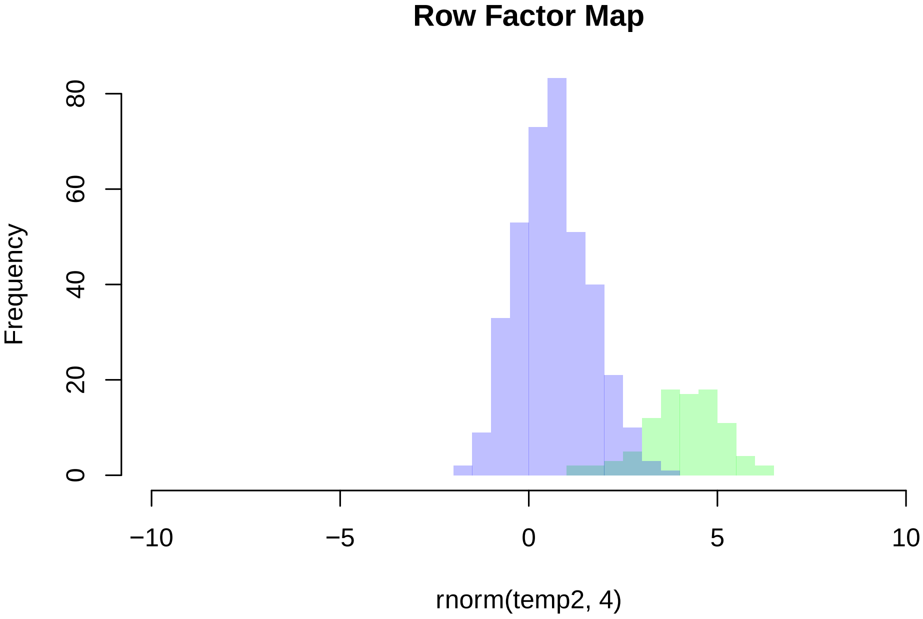
```
#  Observations and means ----
# Observations ----
#_____________________________________________________________________
# I-set map ----
# a graph of the observations
Imap <- PTCA4CATA::createFactorMap(
  resBADA$TExPosition.Data$fii,
  col.points = Group_Colors,
  col.labels = Group_Colors,
  alpha.points = .2
)

Fii <-resBADA$TExPosition.Data$fii
```

```
# make labels ----
label4Map <- createxyLabels.gen(1,2,
                                lambda = resBADA$TExPosition.Data$eigs,
                                tau = resBADA$TExPosition.Data$t)
```

```
##        V1                  V2
##  Min.   :-1.08400   Min.   :1.000
##  1st Qu.:-0.41655   1st Qu.:2.000
##  Median :-0.02619   Median :2.000
##  Mean   :-0.01796   Mean   :1.754
##  3rd Qu.: 0.42596   3rd Qu.:2.000
##  Max.   : 0.98588   Max.   :2.000
```







3.8 Confidence Intervals

Component 1 :As observed only 1 component is significant.Therefore the distributions of both Groups A & B is normally distributed across component 1.

```
# # Confidence intervals
# # Bootstrapped CI ----
# #______________________________________________________________________
# Create Confidence Interval Plots
fi.boot <- resBADA.inf$Inference.Data$boot.data$fi.boot.data$boots
# We want to use the rownames of fi.boot as reference to get the correct
# color. However, the original rownames include "." and don't match with
# the original row names. So, the `sub` function was used to get rid of
# the "." by replacing all "." in the rownames of fi.boot as an empty
# string.
rownames(fi.boot) <- sub("[[:punct:]]","",rownames(fi.boot))
# use function MakeCIEllipses from package PTCA4CATA
GraphElli <- PTCA4CATA::MakeCIEllipses(resBADA.inf$Inference.Data$boot.data$fi.boot.data$boots,
                                       col = col4Means[rownames(fi.boot)], # use rownames as refe
                                       p.level = .95
)
```

```
# create the I-map with Observations, means and confidence intervals
#
a004.bada.withCI <-  Imap$zeMap_background + Imap$zeMap_dots +
                     MapGroup$zeMap_dots + MapGroup$zeMap_text +
                     GraphElli + label4Map +
                     ggtitle('BADA: Group Centers with CI and Observations')
```

3.9 Loadings

Loadings describe the similarity (angular distance) between the variables. Here the input variables are the inputs received from the participants. Loadings show how the input variables relate to each other. Loadings also show which variables are important for (which components load on) a certain component.

The color codes indicate the individual questions for both Group A and Group B.

Component 1: Q10_B (Speed before increase for road B) & Q5_B (Speed after reconstruction) VS Q10_A (Speed before increase for road A)

Along with Q3_Scale (Road Distance) VS Q5_A (Speed after reconstruction for road A)

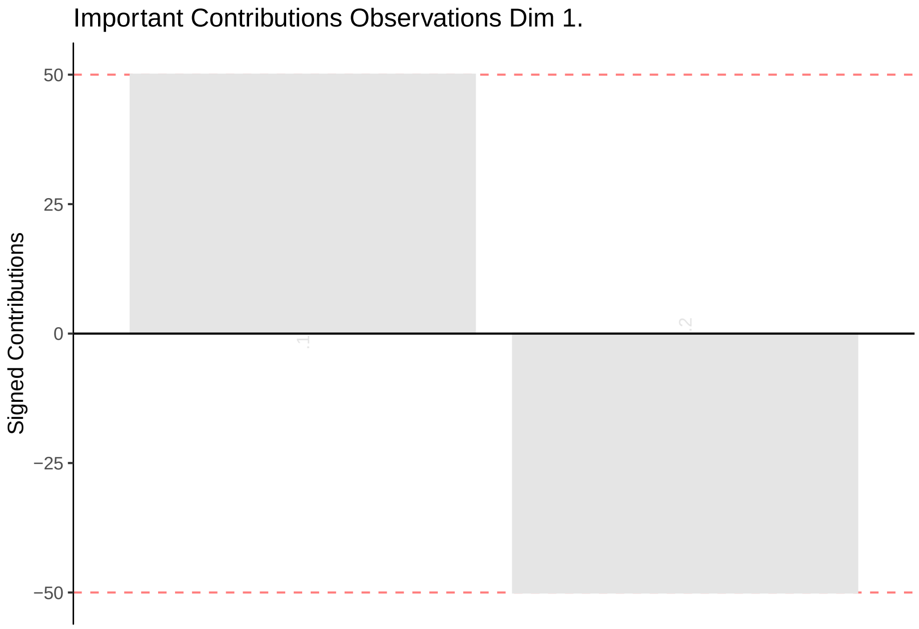
Therefore, the parameters are inversely correlated.

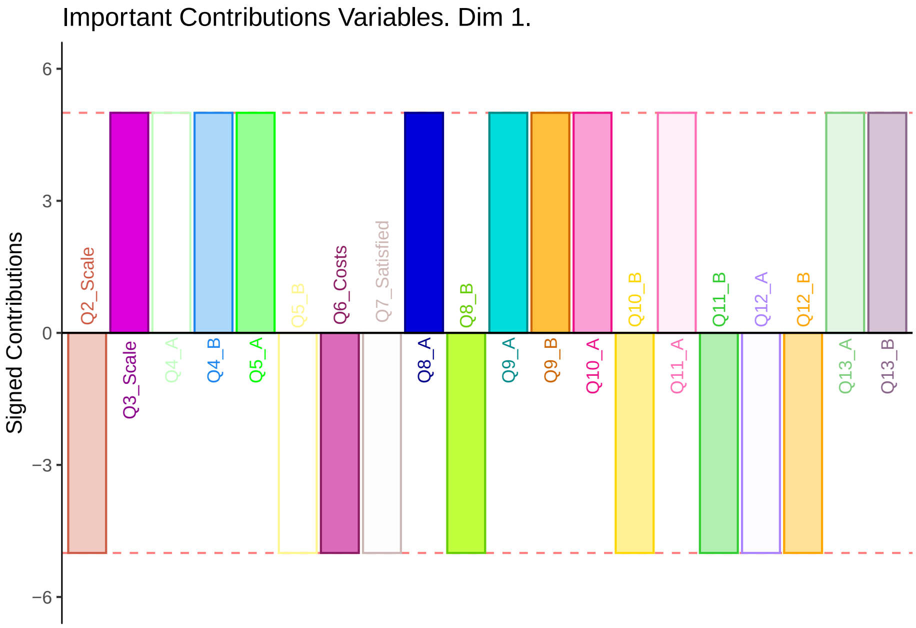
3.10 Contributions barplots

The Contribution barplots indicate the variables that are significant and contribute the most to the respective component.

Component 1: As observed all the variables are important and contribute significantly to the first component.

Component 2: This component does not exist since there is only one component that is significant.



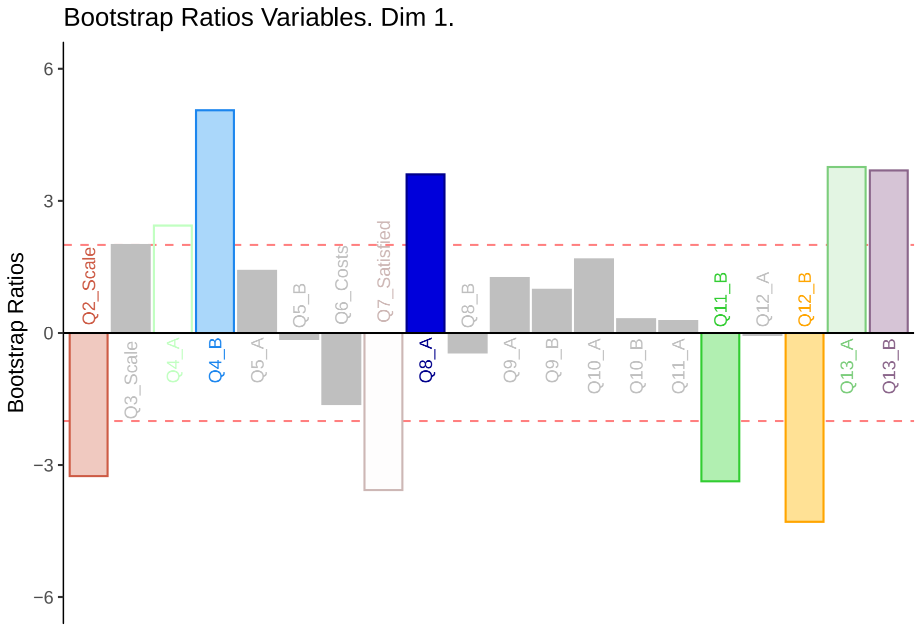


3.11 Bootstrap Ratios

Bootstrap ratios represent the stability and significance of the variables or column factor scores.

Component 1: As observed Q4_B (Current average speed for Road B) & Q12_B (Importance of total time saved for Road B) is significant.

Component 2: This component does not exist since there is only one component that is significant.



3.12 Fixed Model

From the Fixed model we can clearly say that, an accuracy of 55.49% is acheived and it could be used for training of the model.

```
#Fixed Model

row.names(resBADA.inf$Inference.Data$loo.data$fixed.confuse) <- c("1","2")
colnames(resBADA.inf$Inference.Data$loo.data$fixed.confuse) <- c("1","2")
resBADA.inf$Inference.Data$loo.data$fixed.confuse
```

```
##    1  2
## 1 27 65
## 2 20 79
```

```
resBADA.inf$Inference.Data$loo.data$fixed.acc
```

```
## [1] 0.5549738
```

3.13 Random Model

However, the Random model is used for validation and an accuracy of 42.93% is obtained.

```
#Random Model
row.names(resBADA.inf$Inference.Data$loo.data$loo.confuse) <- c("1", "2")
colnames(resBADA.inf$Inference.Data$loo.data$loo.confuse) <- c("1", "2")
resBADA.inf$Inference.Data$loo.data$loo.confuse
```

```
##    1  2
## 1 24 86
## 2 23 58
```

```
resBADA.inf$Inference.Data$loo.data$loo.acc
```

```
## [1] 0.4293194
```

3.14 Summary

Component 1 (Row Factor Scores): As observed only 1 component is significant.Therefore the distributions of both Groups A & B is normally distributed across component 1 and centered around a point.

Component 1 (Column Factor Scores): Q10_B (Speed before increase for road B) & Q5_B (Speed after reconstruction) VS Q10_A (Speed before increase for road A)

Interpretation: Both for Groups A & B the means are centered around a common point. Therefore, the speed before and after increase are significant to the component.

Chapter 4

Multiple Correspondence Analysis

4.1 Method: MCA

Multiple correspondence analysis (MCA) is an extension of corre- spondence analysis (CA) which allows one to analyze the pattern of relationships of several categorical dependent variables. As such, it can also be seen as a generalization of principal component anal- ysis when the variables to be analyzed are categorical instead of quantitative. Because MCA has been (re)discovered many times, equivalent methods are known under several different names such as optimal scaling, optimal or appropriate scoring, dual scaling, homogeneity analysis, scalogram analysis, and quantification method.

Technically MCA is obtained by using a standard correspon- dence analysis on an indicator matrix (i.e., a matrix whose entries are 0 or 1). The percentages of explained variance need to be cor- rected, and the correspondence analysis interpretation of inter- point distances needs to be adapted.

4.2 Data set: Drive.RData

Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test. It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

- Q_1: Implementation Decision
- Q2_Scale: Implementation Decision & judgement

- Q3_Scale: Road Distance
- Q4_A: Current Average Speed (A)
- Q4_B: Current Average Speed (B)
- Q5_A: Speed after reconstruction (A)
- Q5_B: Speed after reconstruction (B)
- Q6_Costs: Cost
- Q7_Satisfied: Satisfied with implementation and judgement
- Q8_A: Travel Time Saved (A)
- Q8_B: Travel Time Saved (B)
- Q9_A: Importance of Road Distance (A)
- Q9_B: Importance of Road Distance (B)
- Q10_A: Importance of speed before increase (A)
- Q10_B: Importance of speed before increase (B)
- Q11_A: Importance of speed after increase (A)
- Q11_B: Importance of speed after increase (B)
- Q12_A: Importance of time saving (A)
- Q12_B: Importance of time saving (B)
- Q13_A: Importance of Costs (A)
- Q13_B: Importance of Costs (B)

```
load("Drive.RData")

data.pca2 <- data.pca[1:191,2:21]

head(data.pca2)
```

```
##   Q2_Scale Q3_Scale  Q4_A  Q4_B  Q5_A  Q5_B Q6_Costs Q7_Satisfied  Q8_A
## 1    88.91    34.44  8.95 21.79 17.32 45.91    68.48        92.80 19.46
## 2    97.28    32.49 15.76 22.57 54.86 59.73   100.00        93.39 13.23
## 3    81.32    43.77  8.56 20.62 19.65 49.81    66.54        89.49 81.91
## 4    89.49    25.10  3.11 13.81  8.37 33.07    65.56        85.80 66.54
## 5    76.26    50.19  4.28 17.70 14.20 41.05    25.68        50.19 22.76
## 6    70.04    33.85  7.39 17.32 16.34 47.86    66.54        67.90 25.10
##    Q8_B  Q9_A  Q9_B  Q10_A  Q10_B  Q11_A  Q11_B  Q12_A  Q12_B Q13_A Q13_B
## 1 25.10 16.15 15.95 100.00  24.51 100.00  25.49 100.00  36.77 48.64 48.83
## 2 42.22 44.75 44.75  19.07  19.65  75.10  86.58  35.60  55.84  3.89  5.45
## 3 62.06 87.55 87.35 100.00 100.00 100.00 100.00 100.00 100.00  0.00  0.00
## 4 22.37  0.78  0.78  92.02 100.00  96.89 100.00   0.58   0.97  1.56  0.78
## 5 50.58 26.07 74.71  34.24  39.69  47.28  51.56  30.35  62.45 35.02 57.98
## 6 42.41 49.42 49.42  72.57  57.39  80.74  56.42  67.70  64.59 50.97 51.36
```

```
data.choice <- data.pca[["Q1_Choice"]] # Observation (row names)
#For more info,
#see: "?Drive.RData"
#also, type: "data.", and Tab to explore additional info
```

4.3 Including Plots

```
summary(data.pca)
```

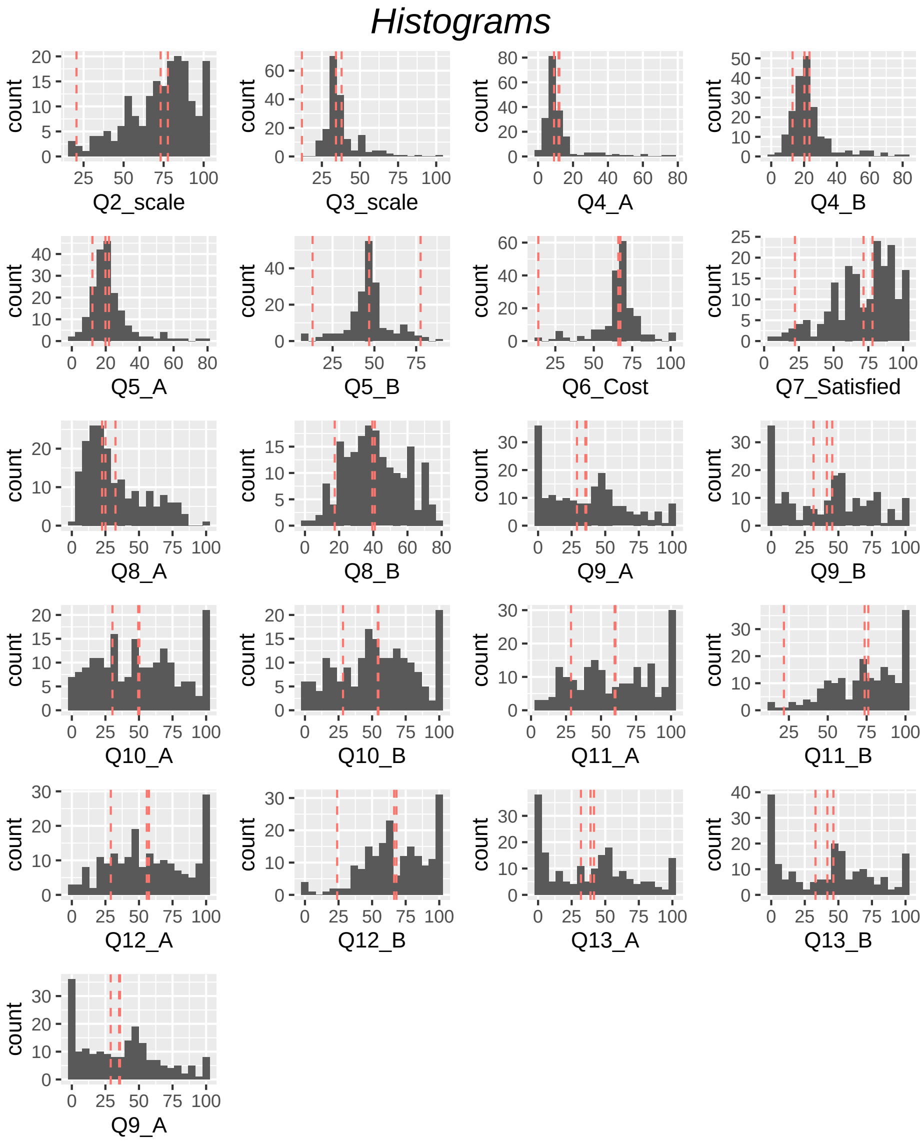
```
##  Q1_Choice    Q2_Scale         Q3_Scale           Q4_A
##  1: 47     Min.   : 16.12   Min.   : 21.21   Min.   : 0.580
##  2:144     1st Qu.: 59.83   1st Qu.: 31.39   1st Qu.: 6.605
##            Median : 77.63   Median : 34.24   Median : 9.140
##            Mean   : 73.12   Mean   : 37.95   Mean   :12.175
##            3rd Qu.: 89.01   3rd Qu.: 39.53   3rd Qu.:12.160
##            Max.   :100.00   Max.   :100.00   Max.   :77.480
##       Q4_B            Q5_A            Q5_B          Q6_Costs
##  Min.   : 1.56   Min.   : 0.78   Min.   : 6.61   Min.   : 12.65
##  1st Qu.:16.44   1st Qu.:15.36   1st Qu.:41.98   1st Qu.: 64.23
##  Median :20.23   Median :19.81   Median :46.89   Median : 67.32
##  Mean   :23.28   Mean   :21.90   Mean   :46.86   Mean   : 65.98
##  3rd Qu.:25.66   3rd Qu.:24.39   3rd Qu.:50.39   3rd Qu.: 71.40
##  Max.   :83.30   Max.   :79.96   Max.   :86.96   Max.   :100.00
##   Q7_Satisfied         Q8_A             Q8_B            Q9_A
##  Min.   :  3.31   Min.   :  0.00   Min.   : 0.78   Min.   :  0.000
##  1st Qu.: 58.93   1st Qu.: 14.87   1st Qu.:28.02   1st Qu.:  9.335
##  Median : 78.02   Median : 25.05   Median :39.42   Median : 35.210
##  Mean   : 71.55   Mean   : 32.50   Mean   :40.88   Mean   : 35.881
##  3rd Qu.: 89.41   3rd Qu.: 47.48   3rd Qu.:53.99   3rd Qu.: 51.990
##  Max.   :100.00   Max.   :100.00   Max.   :79.38   Max.   :100.000
##       Q9_B            Q10_A            Q10_B            Q11_A
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  4.08
##  1st Qu.:  9.33   1st Qu.: 24.57   1st Qu.: 33.23   1st Qu.: 37.94
##  Median : 45.44   Median : 49.42   Median : 54.67   Median : 59.73
##  Mean   : 41.43   Mean   : 50.26   Mean   : 54.21   Mean   : 60.28
##  3rd Qu.: 65.18   3rd Qu.: 72.28   3rd Qu.: 75.36   3rd Qu.: 85.31
##  Max.   :100.00   Max.   :100.00   Max.   :100.00   Max.   :100.00
##      Q11_B            Q12_A            Q12_B            Q13_A
##  Min.   : 13.62   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 58.31   1st Qu.: 34.30   1st Qu.: 50.58   1st Qu.:  5.74
##  Median : 76.07   Median : 55.84   Median : 66.41   Median : 41.63
##  Mean   : 73.74   Mean   : 57.44   Mean   : 68.14   Mean   : 39.03
##  3rd Qu.: 92.61   3rd Qu.: 82.22   3rd Qu.: 87.84   3rd Qu.: 60.99
##  Max.   :100.00   Max.   :100.00   Max.   :100.00   Max.   :100.00
##      Q13_B
##  Min.   :  0.00
##  1st Qu.:  6.42
##  Median : 46.30
##  Mean   : 41.80
##   3rd Qu.: 65.95
##   Max.   :100.00
```

```
resPCA <- epPCA(DATA = meas.questions,scale = 'SS1', DESIGN = group.questions.choice,g
```

4.4 Construct histograms and observe the variables

Now lets us construct the histograms of individual variables and cut them into 4 quartiles for obtaining the foctor levels.

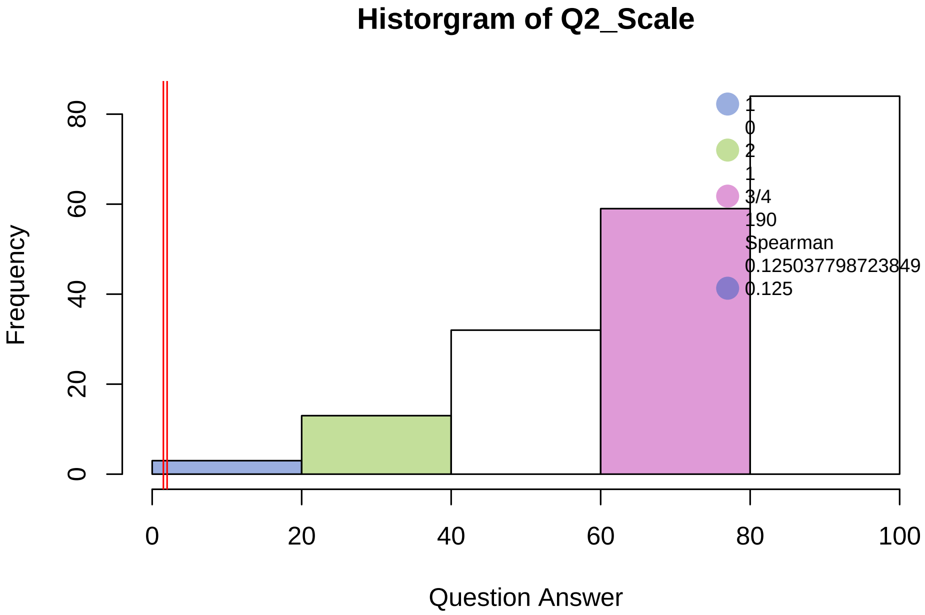
```
gridExtra::grid.arrange(as.grob(hist.Q2_scale), as.grob(hist.Q3_scale), as.grob(hist.Q4
                        as.grob(hist.Q8_A),as.grob(hist.Q8_B),as.grob(hist.Q9_A),as.gro
                        as.grob(hist.Q10_A),as.grob(hist.Q10_B),as.grob(hist.Q11_A),as
                        as.grob(hist.Q12_A),as.grob(hist.Q12_B),as.grob(hist.Q13_A),as
                        as.grob(hist.Q9_A),
                        ncol=4, top = textGrob("Histograms",gp=gpar(fontsize=18,font=3
```

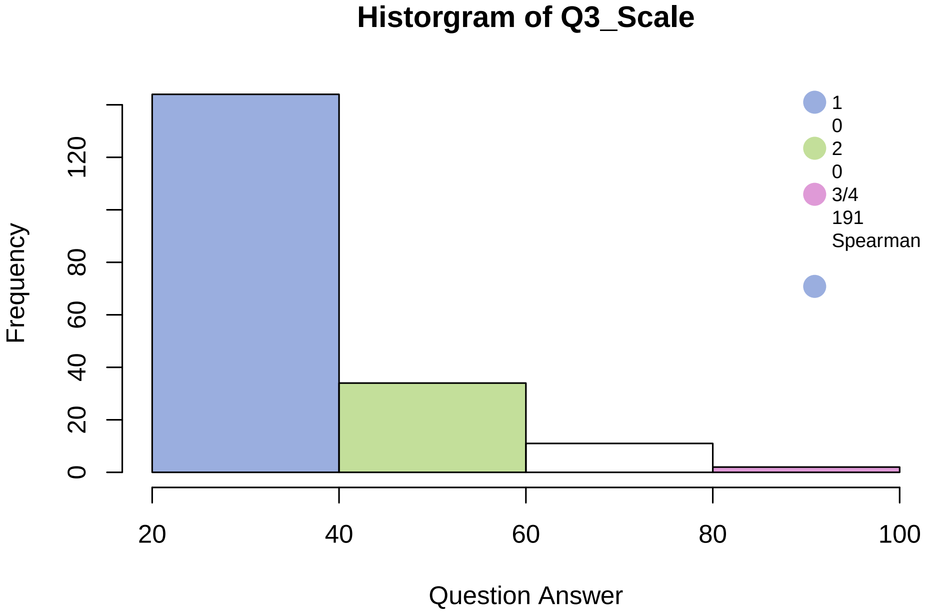


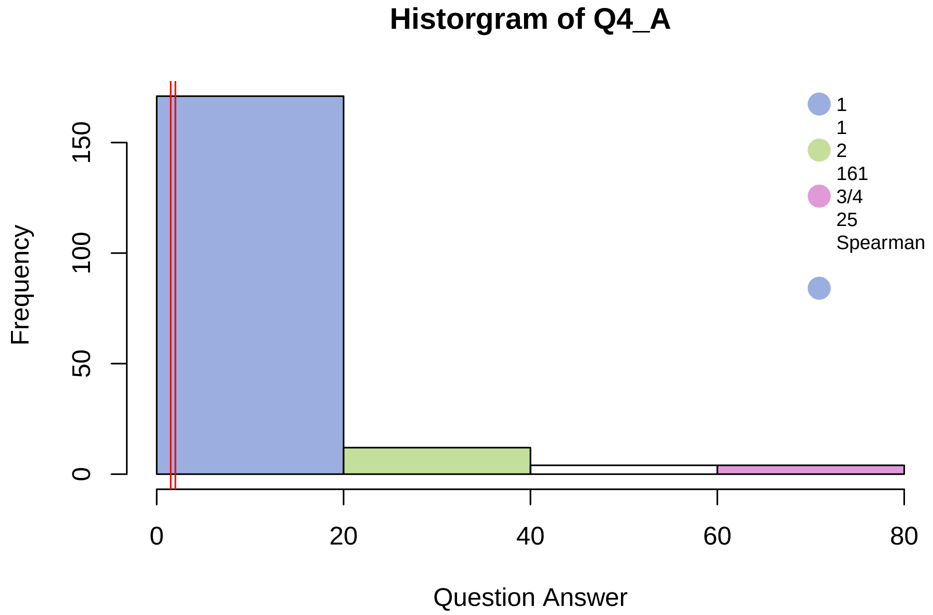
```
all.hist <- recordPlot()
```

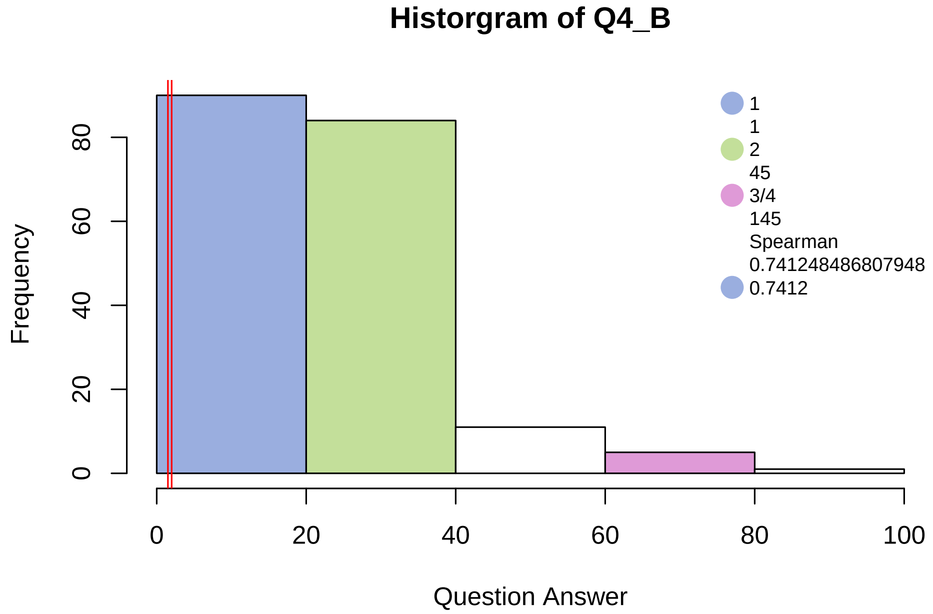
4.5 Binning of Dataset

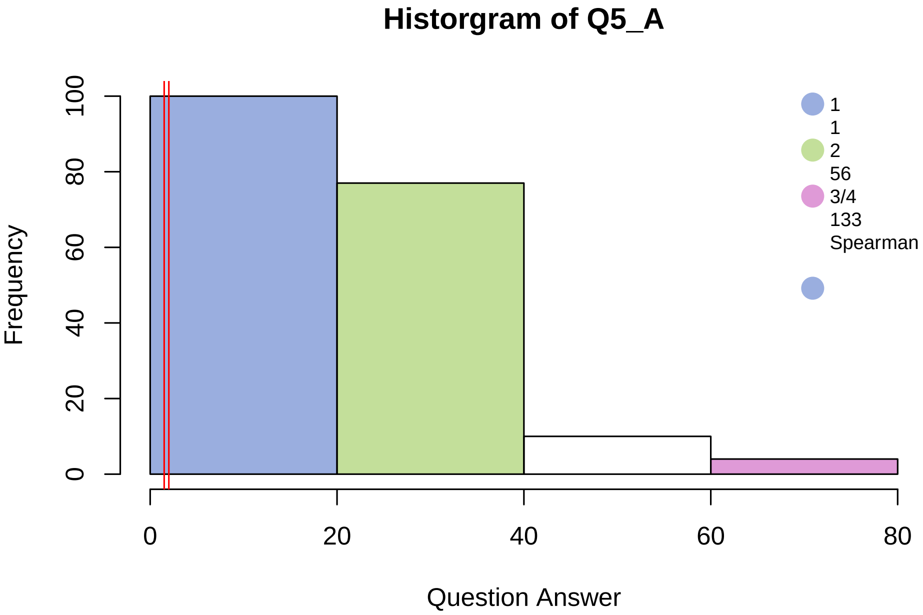
```
## Warning in cor(data.pca2[, i], as.numeric(recode), method = "spearman"):
## the standard deviation is zero
```

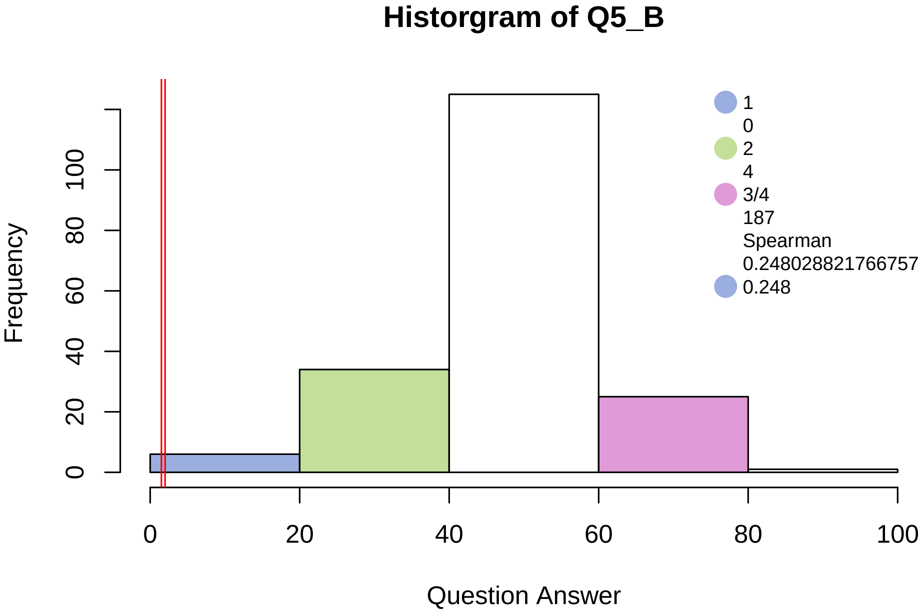

Historgram of Q2_Scale
Frequency
0
20
40
60
80
0
20
40
60
80
100
Question Answer
1
0
2
1
3/4
190
Spearman
0.125037798723849
0.125

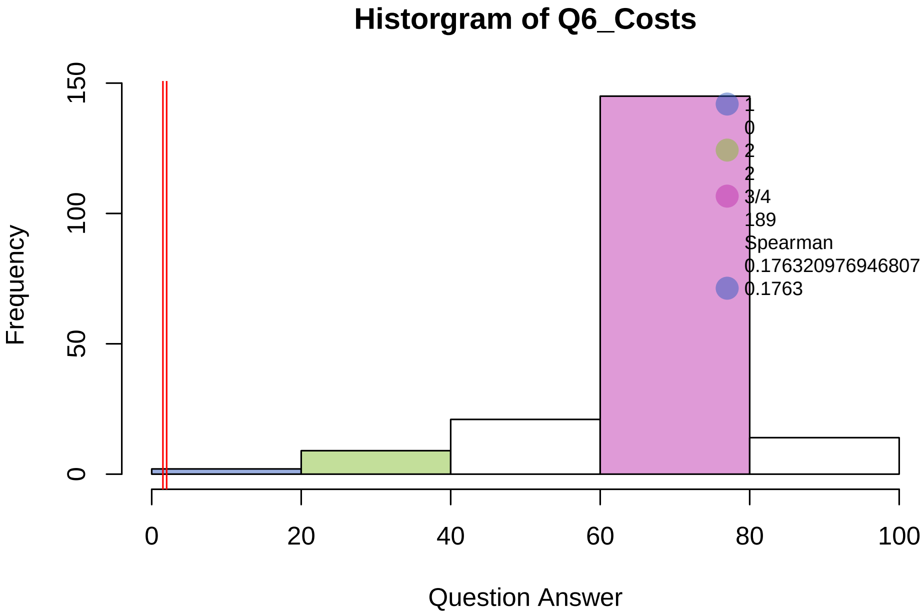

Historgram of Q3_Scale
Frequency
0
20
40
60
80
120
20
40
60
80
100
Question Answer
1
0
2
0
3/4
191
Spearman

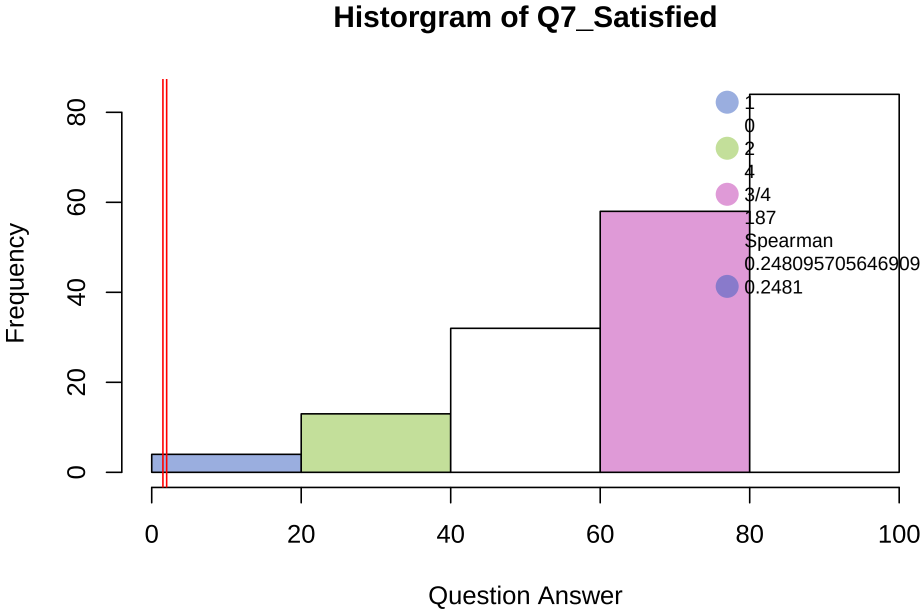

Historgram of Q4_A
Frequency
0
50
100
150
0
20
40
60
80
Question Answer
1
1
2
161
3/4
25
Spearman

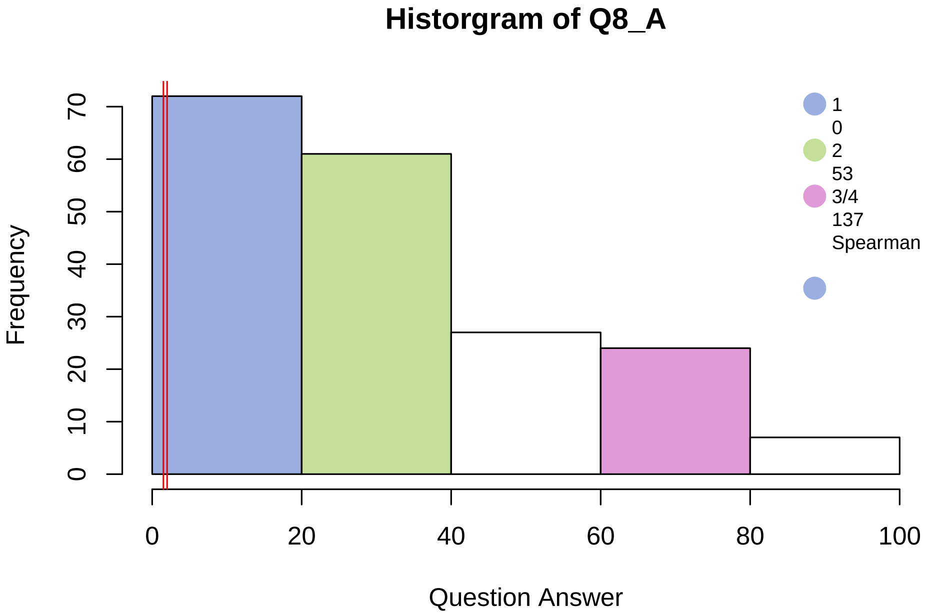

Historgram of Q4_B
Frequency
0
20
40
60
80
0
20
40
60
80
100
Question Answer
1
1
2
45
3/4
145
Spearman
0.741248486807948
0.7412

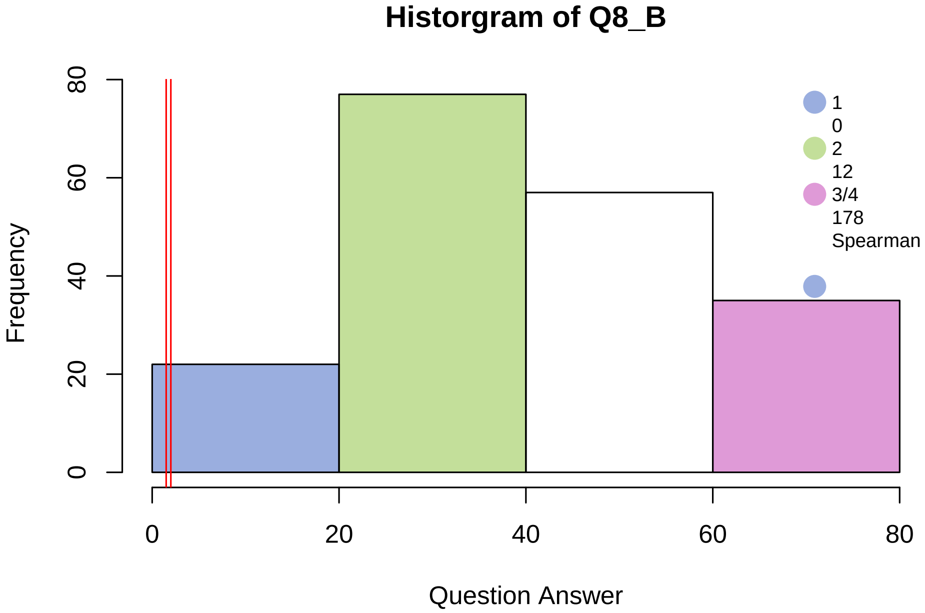

Historgram of Q5_A
Frequency
0
20
40
60
80
100
0
20
40
60
80
Question Answer
1
1
2
56
3/4
133
Spearman

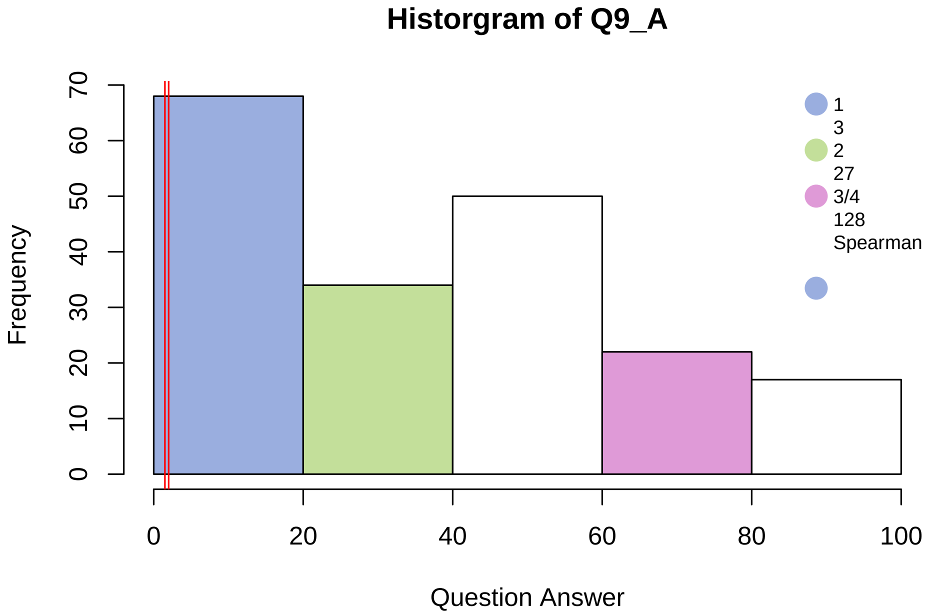

Historgram of Q5_B
Frequency
0
20
40
60
80
100
0
20
40
60
80
100
Question Answer
1
0
2
4
3/4
187
Spearman
0.248028821766757
0.248

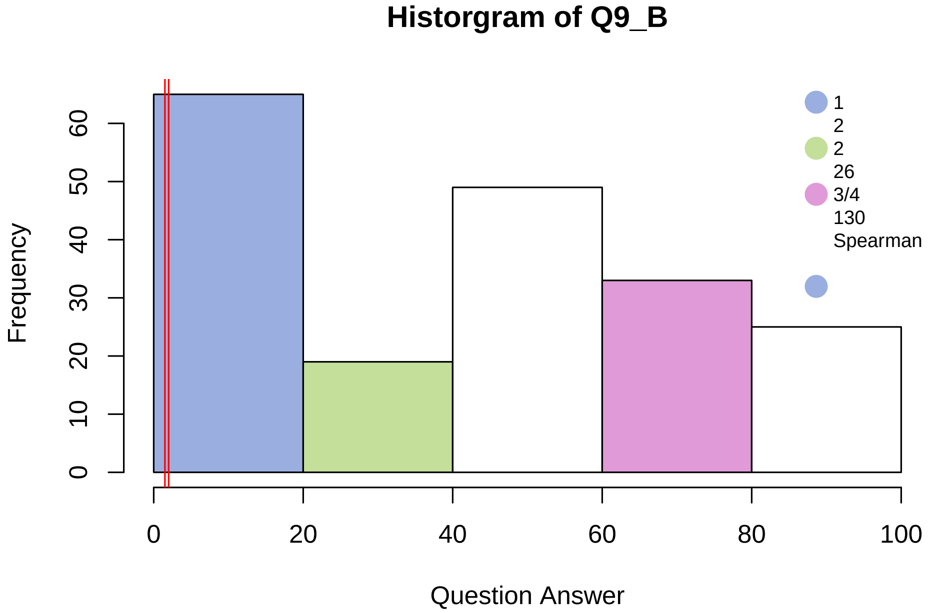

Historgram of Q6_Costs

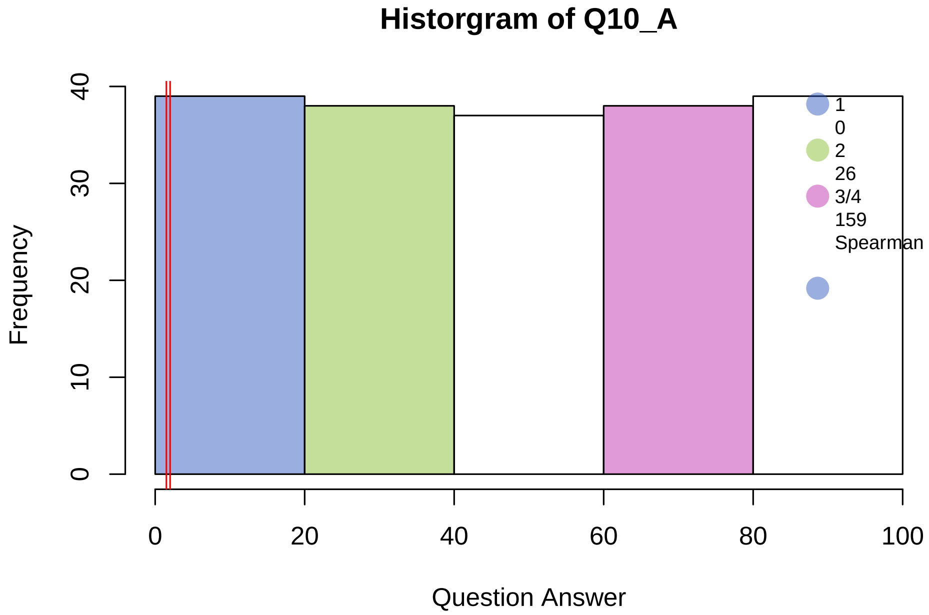

Historgram of Q7_Satisfied

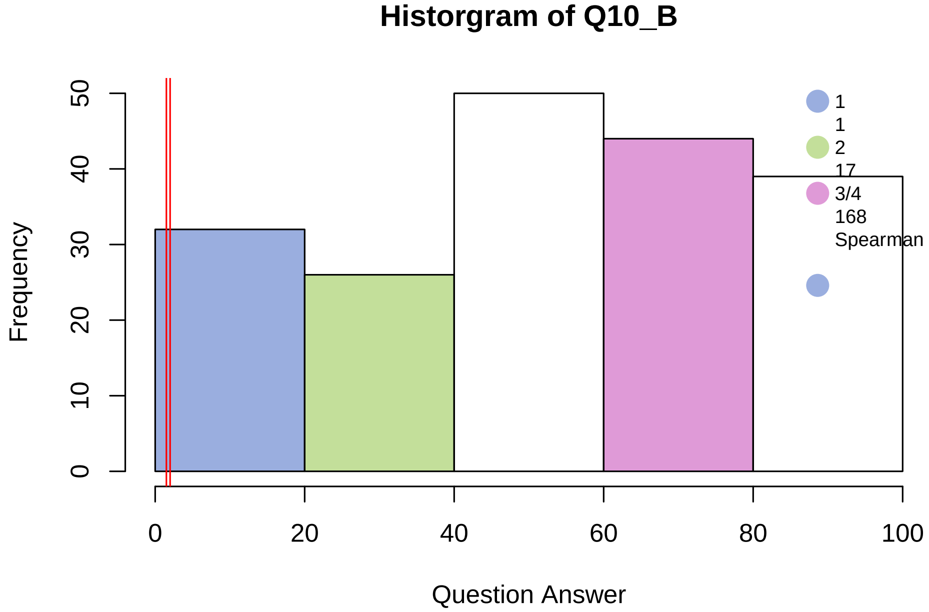

Historgram of Q8_A
Frequency
0
10
20
30
40
50
60
70
0
20
40
60
80
100
Question Answer
1
0
2
53
3/4
137
Spearman

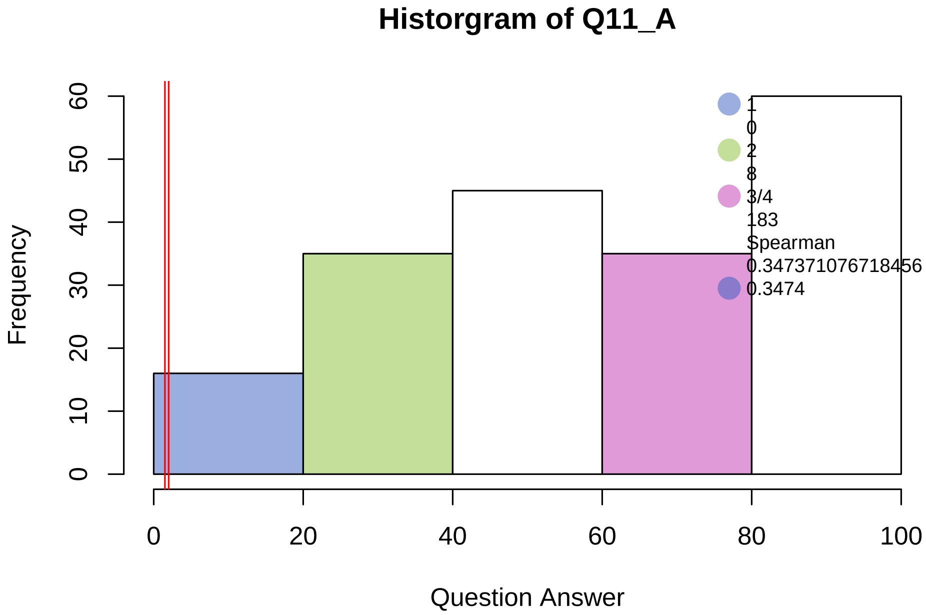

Historgram of Q8_B
Frequency
0
20
40
60
80
0
20
40
60
80
Question Answer
1
0
2
12
3/4
178
Spearman

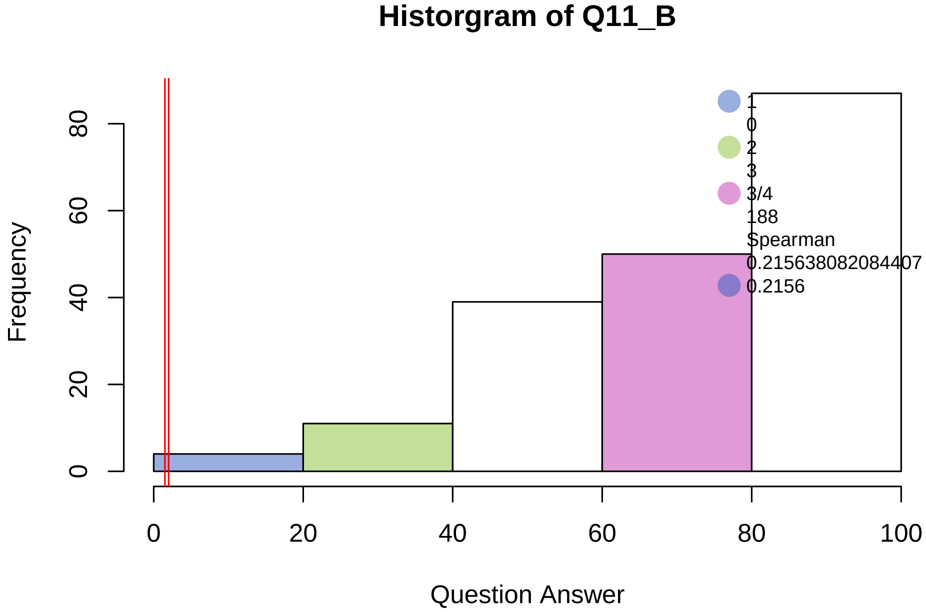

Historgram of Q9_A
Frequency
0
10
20
30
40
50
60
70
0
20
40
60
80
100
Question Answer
● 1
3
● 2
27
● 3/4
128
Spearman
●

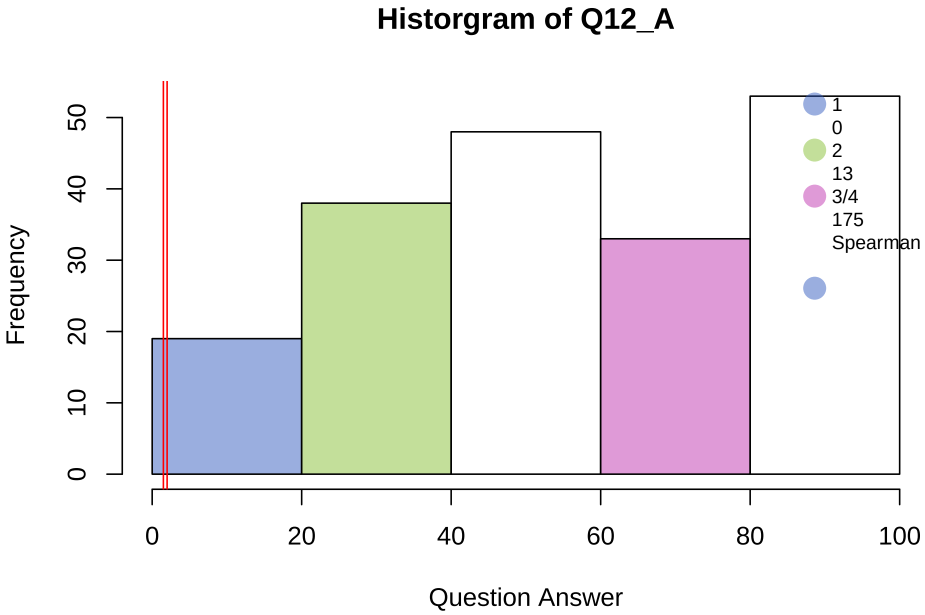

Historgram of Q9_B
Frequency
0
10
20
30
40
50
60
0
20
40
60
80
100
Question Answer
● 1
2
● 2
26
● 3/4
130
Spearman
●

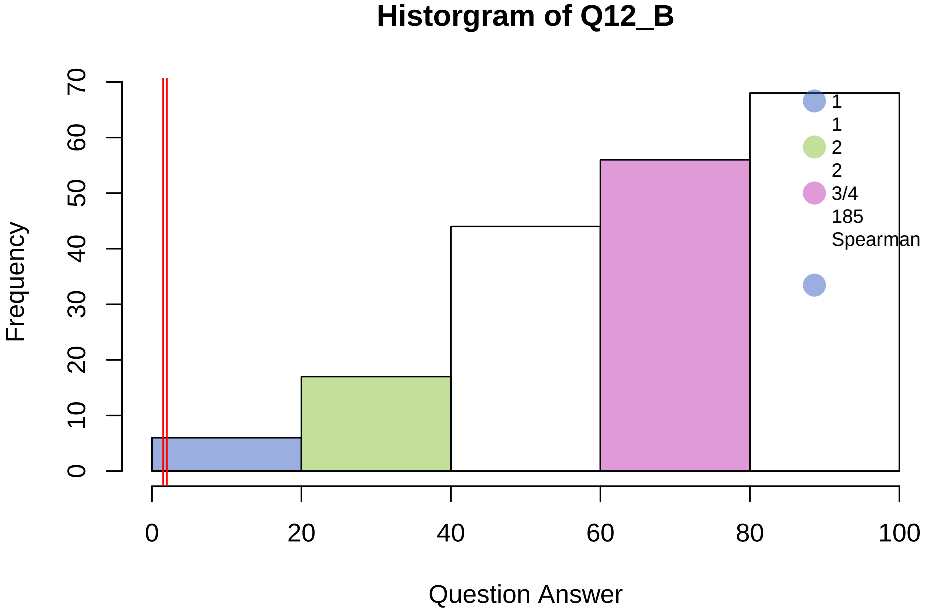

Historgram of Q10_A
Frequency
0
10
20
30
40
0
20
40
60
80
100
Question Answer
● 1
0
● 2
26
● 3/4
159
Spearman
●

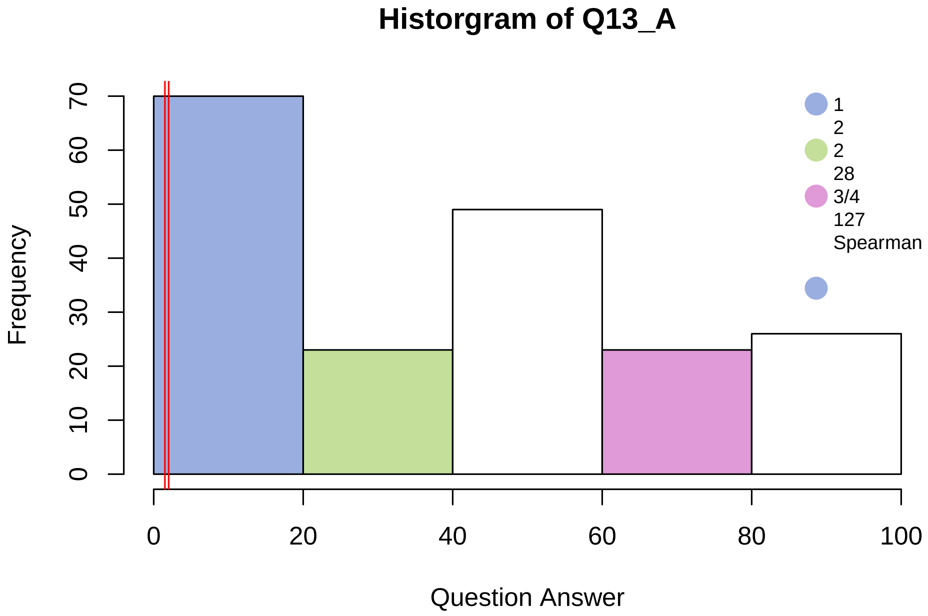

Historgram of Q10_B
Frequency
0
10
20
30
40
50
0
20
40
60
80
100
Question Answer
● 1
1
● 2
17
● 3/4
168
Spearman
●


Historgram of Q11_A
Frequency
0
10
20
30
40
50
60
0
20
40
60
80
100
Question Answer
1
0
2
8
3/4
183
Spearman
0.347371076718456
0.3474


Historgram of Q11_B
Frequency
0
20
40
60
80
0
20
40
60
80
100
Question Answer
1
0
2
3
3/4
188
Spearman
0.215638082084407
0.2156


Historgram of Q12_A
Frequency
0
10
20
30
40
50
0
20
40
60
80
100
Question Answer
1
0
2
13
3/4
175
Spearman


Historgram of Q12_B
Frequency
0
10
20
30
40
50
60
70
0
20
40
60
80
100
Question Answer
1
1
2
2
3/4
185
Spearman



```
## [1] "You have 159 NAs in your data. makeNominalData automatically imputes NA with the mean of
```

```
# recode for  MCA ----
# Initialized recoded df

meas.questions.Recoded <- data.frame(row.names = row.names(meas.questions))
```

```
#  Recode as 4 quartiles
irec = which(colnames(meas.questions)=='Q2_Scale')
meas.questions.Recoded[,colnames(meas.questions)[irec]] <- BinQuant(
                              meas.questions[,irec], nClass = 2, stem = '')
```

4.6 Check the spearman's rank correlation

4.7 Observe the sample size

From the spearman's rank correlation we receive an accuracy of 86.62%.

```
#>-----check the recoded results---------
#> check the spearman's rank correlation

    cor(meas.questions[,irec],as.numeric(meas.questions.Recoded[,colnames(meas.questions)[irec]])
## [1] 0.8662873
```

```
#> show sample size
    table(meas.questions.Recoded[,colnames(meas.questions)[irec]])
```

```
##
##  1  2
## 96 95
```

4.8 Disjunctively coded matrix

A disjunctively coded matrix is obtained to observe the factor levels we liked to achieve in order to interpret which groups (either A or B) answered questions less and more frequently.

```
# Disjunctively coded
meas.questions.Recoded.dis <- makeNominalData(meas.questions.Recoded)
meas.questions.Recoded.dis[1:5,1:6]
```

```
##   Q2_Scale.2 Q2_Scale.1 Q3_Scale.2 Q3_Scale.1 Q4_A.1 Q4_A.2
## 1          1          0          1          0      1      0
## 2          1          0          0          1      0      1
## 3          1          0          1          0      1      0
## 4          1          0          0          1      1      0
## 5          0          1          1          0      1      0
```

4.9 Run the MCA

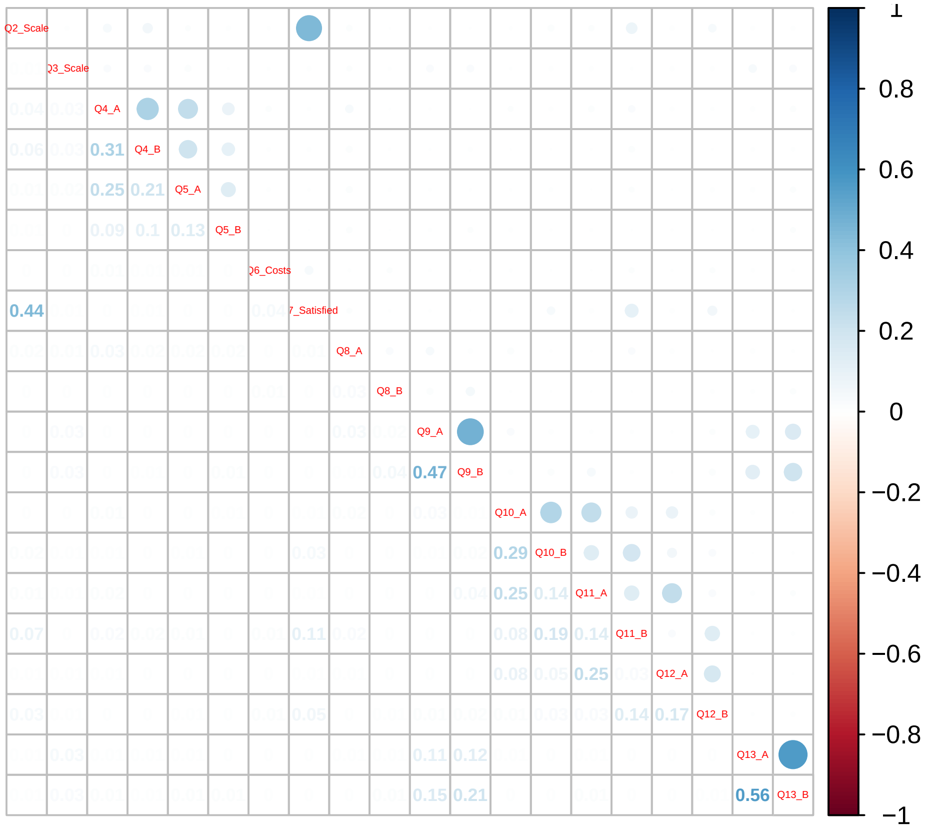
```
resMCA <- epMCA(DATA = meas.questions.Recoded.dis, make_data_nominal = FALSE, DESIGN =
```

4.10 Inference battery

4.11 Pseudo Heat Map Correlation

From the heat map it is observed that there is a strong correlation between the current average speed (Q__4A & Q__4B) and ofcourse the important parameters- importance of road distance (Q__9A & Q__9B), speed before and after increase (Q__10A,Q__10B,Q__11A & Q__11B) and Importance of cost (Q__13A & Q__13B).

```
#  GRAPHS -----
#______________________________________________________________________
# Pseudo Heat Map ----
corrMatBurt.list <- phi2Mat4BurtTable(meas.questions.Recoded)
corr4MCA <- corrplot.mixed(as.matrix(corrMatBurt.list$phi2.mat,
                           title = "Phi2: (squared) Correlation Map for MCA"),tl.cex=0.3,number.c
```



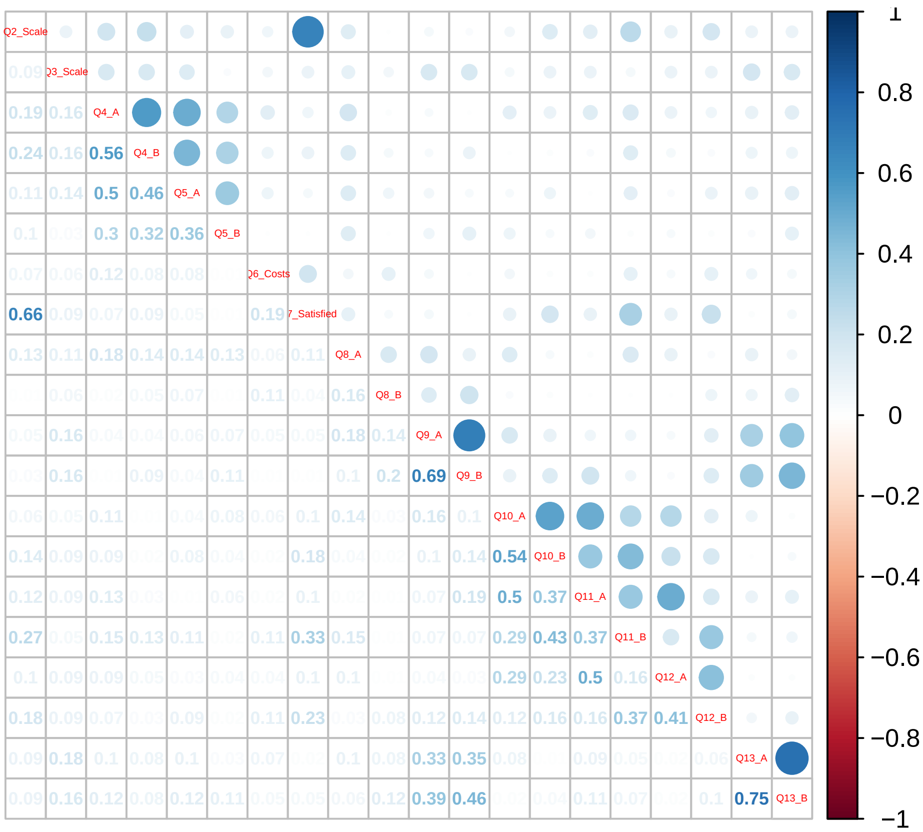
```
a0001a.corMat.phi2 <- recordPlot()
```

4.12 Pseudo Heat Map. Correlation

This heatmap is similar to the previous one.The correlation is compared with PCA.

```
# Pseudo Heat Map. Correlation ----
# We need correlation to compare with PCA
```

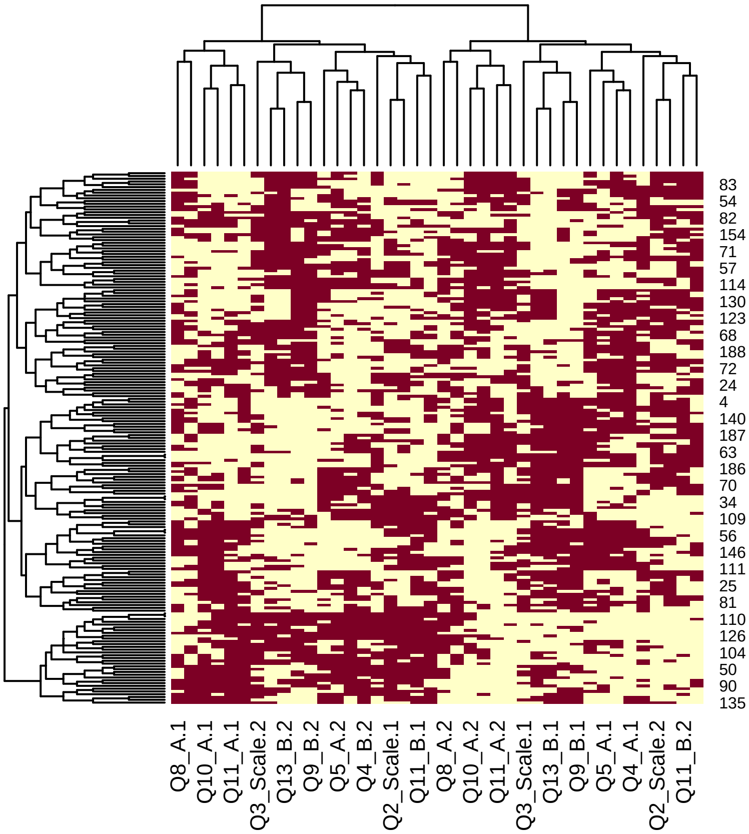
```
corrMatBurt.list <- phi2Mat4BurtTable(meas.questions.Recoded)
corr4MCA.r <- corrplot.mixed(as.matrix(sqrt(corrMatBurt.list$phi2.mat),
                                     title = "Phi: Correlation Map for MCA"),tl.cex=0.3
```



```
a0001b.corMat.phi <- recordPlot()
```

4.13 Heatmap after binning

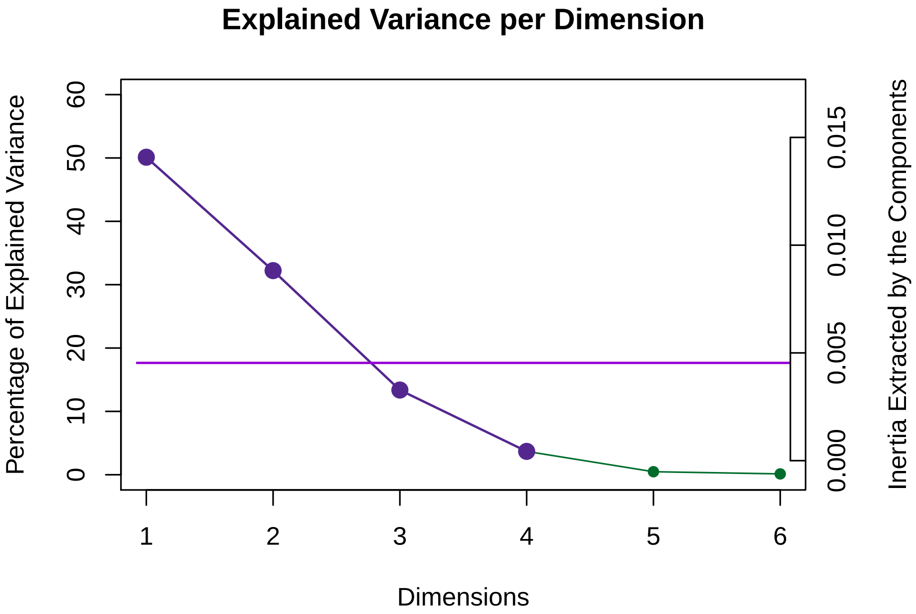
```
heatmap(x=meas.questions.Recoded.dis)
```



4.14 The Scree Plot:

From the Scree Plot we can say that 2 dimension are significant and need to be interpreted.

```
# Scree ----
PlotScree(ev = resMCA$ExPosition.Data$eigs,p.ev = resMCA.inf$Inference.Data$components$p.vals, pl
```



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

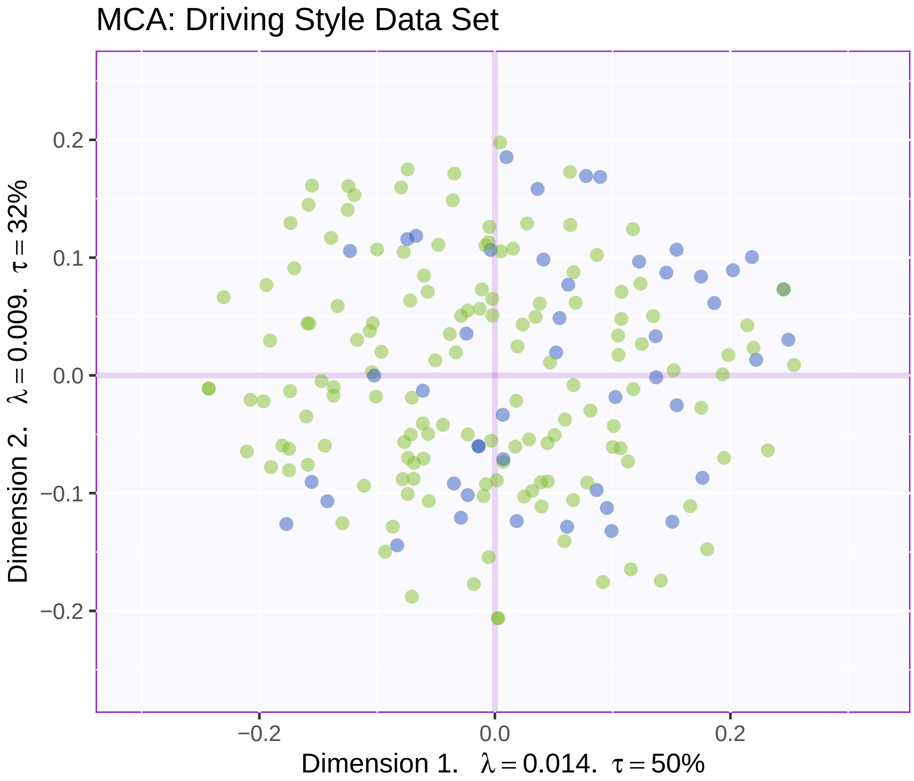
4.15 The Graph of Observations: Factor scores

Factor scores are the coordinates of the 191 participant's choices on the components. The distances between them show which participants had similar choices(Choice 1/Choice 2) . Factor scores (choices) can be color-coded to help interpret the components. Choice 1 corresponds to Group A and Choice 2 corresponds to Group B. As we notice the observations we can clearly say that Group A (indicated in the color blue) and the Group B (indicated in the color green) are at the extremes of the dimension 1.

```
# I-set map ----
# a graph of the observations
data.pca.Imap <- PTCA4CATA::createFactorMap(
  title = 'MCA: Driving Style Data Set',
  resMCA$ExPosition.Data$fi, 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

  col.points = resMCA$Plotting.Data$fi.col,
  col.labels = resMCA$Plotting.Data$fi.col,
  display.labels = FALSE,
  alpha.points = .5
)
```

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

```
a002.Map.I <- data.pca.Imap$zeMap + label4Map
print(a002.Map.I)
```

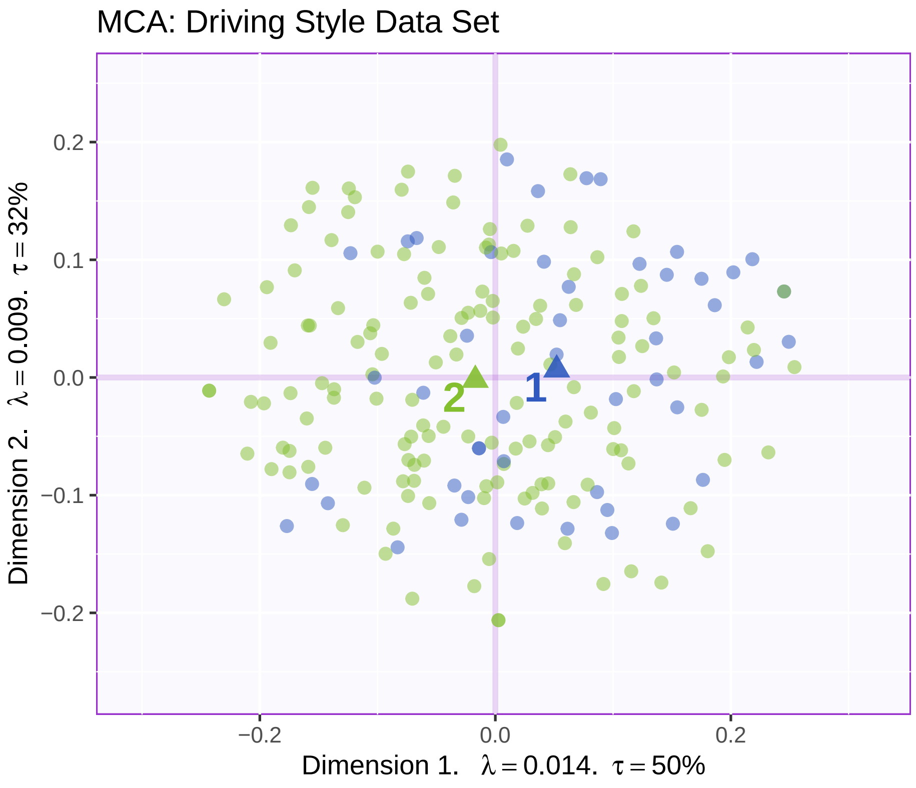


4.16 Obtain the color for each group:

```
# get index for the first row of each group
grp.ind <- order(group.questions.choice)[!duplicated(sort(group.questions.choice))]
grp.col <-resMCA.inf$Fixed.Data$Plotting.Data$fi.col[grp.ind] # get the color
grp.name <- group.questions.choice[grp.ind] # get the corresponding groups
names(grp.col) <- grp.name
```

4.17 With group means

```
fi.mean.plot <- createFactorMap(fi.mean,
                                alpha.points = 0.9,
                                col.points = grp.col[rownames(fi.mean)],
                                col.labels = grp.col[rownames(fi.mean)],
                                pch = 17,
                                cex = 3.5,
                                text.cex = 6)
fi.WithMean <- data.pca.Imap$zeMap_background + data.pca.Imap$zeMap_dots + fi.mean.plot
fi.WithMean
```



- Component 1: Group A Vs Group B

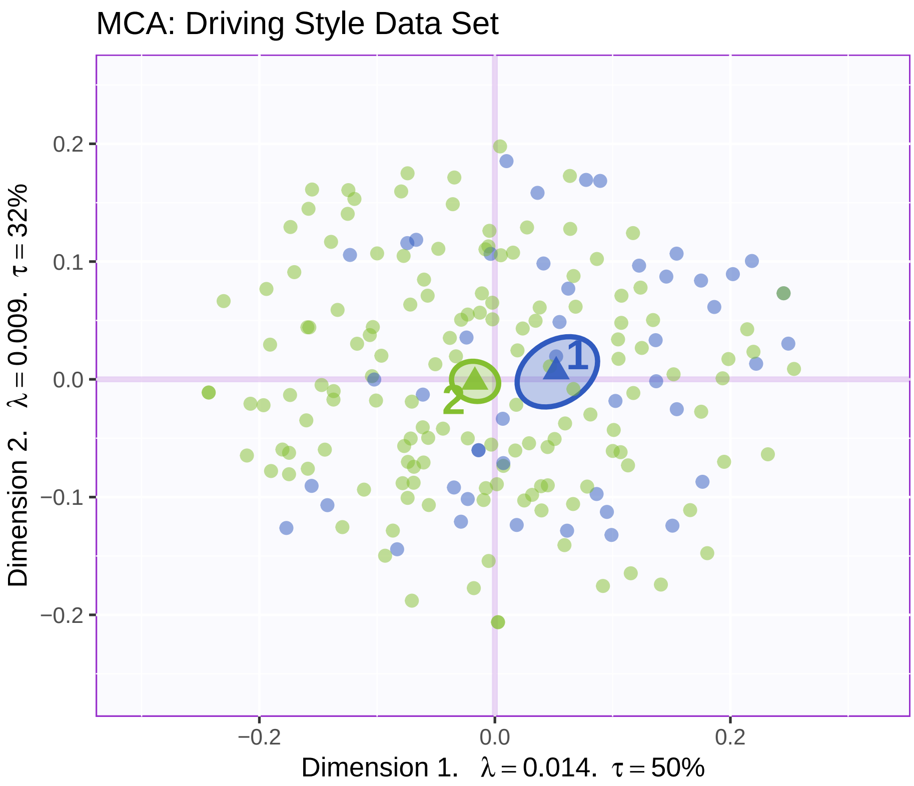
- Component 2: Group A Vs Group B

- Choice 1 (Group A) selected by the participants is indicated by the color Blue (Current Avg. Speed=15mph & Speed after reconstruction =29mph)

- Choice 2 (Group B)selected by the participants is indicated by the color Green (Current Avg. Speed=35mph & Speed after reconstruction =71mph)

4.18 Confidence Intervals

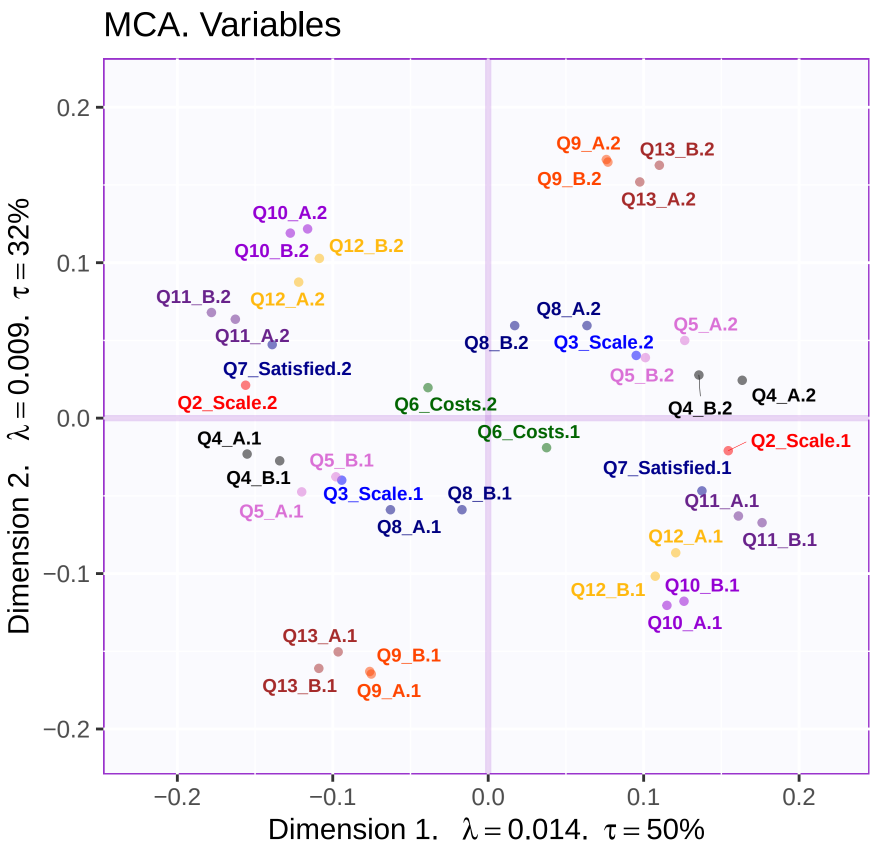
```
bootCI4mean <- MakeCIEllipses(fi.boot$BootCube[,c(1:2),], # get the first two components
                              col = grp.col[rownames(fi.mean)])

fi.WithMeanCI <- data.pca.Imap$zeMap_background + bootCI4mean + data.pca.Imap$zeMap_dots + fi.mea
fi.WithMeanCI
```



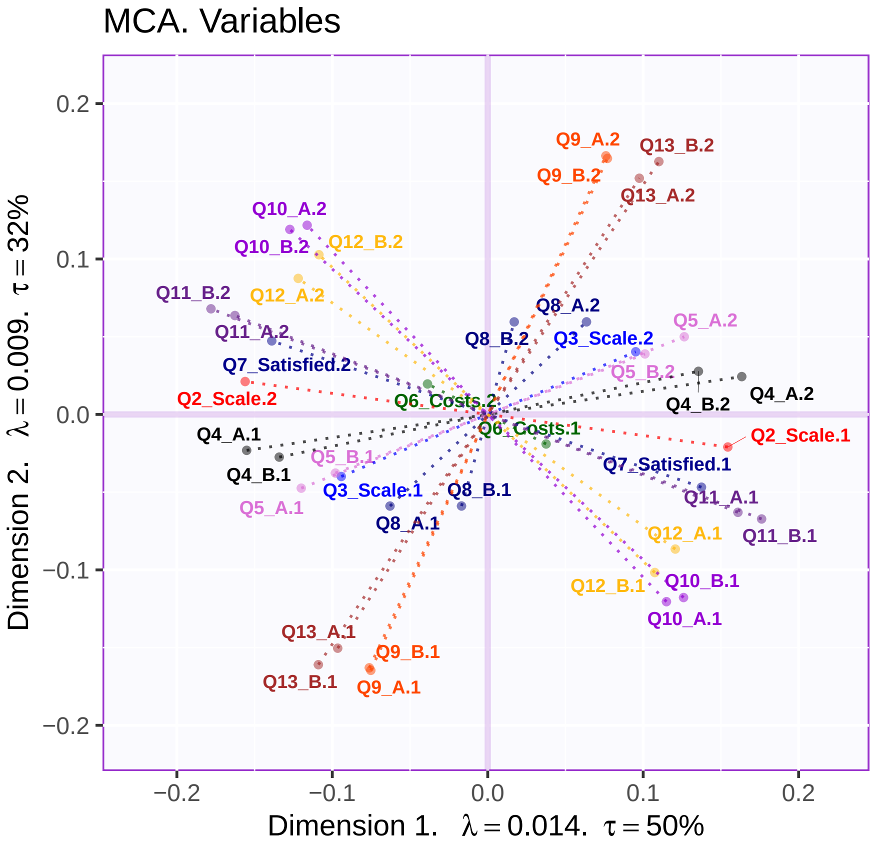
4.19 Column Factor Scores

4.20 Make the J-maps

```
# make the J-maps ----
 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
 print(b001.BaseMap.Fj)
```

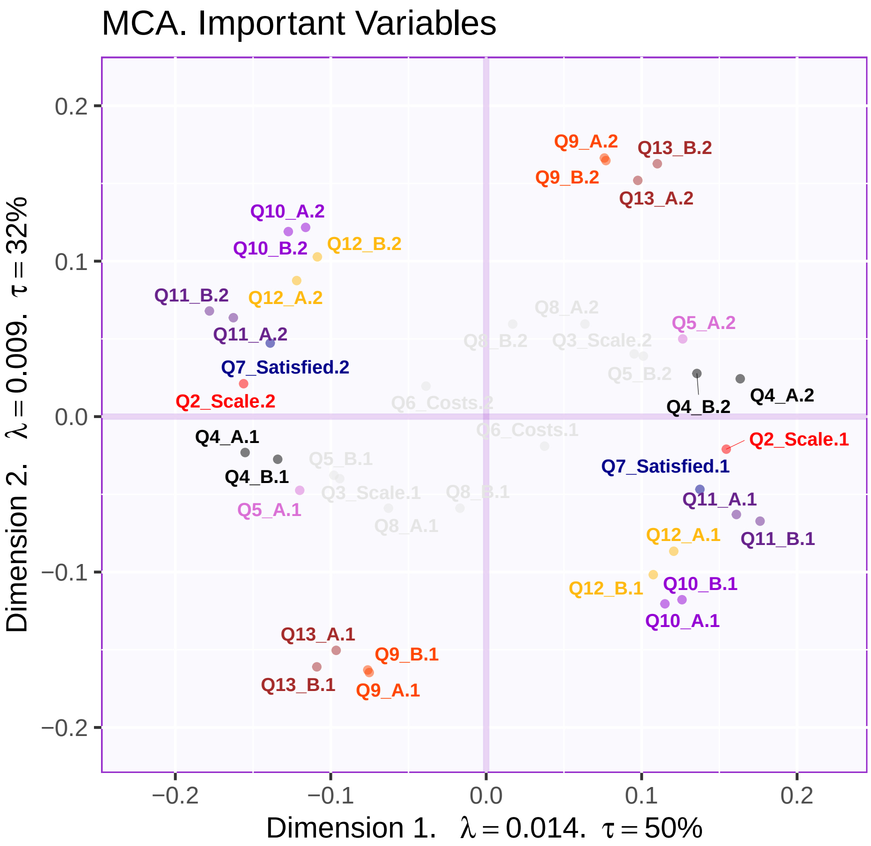


```
print(b003.MapJ)
```



4.21 Levels of variables: map with only important variables

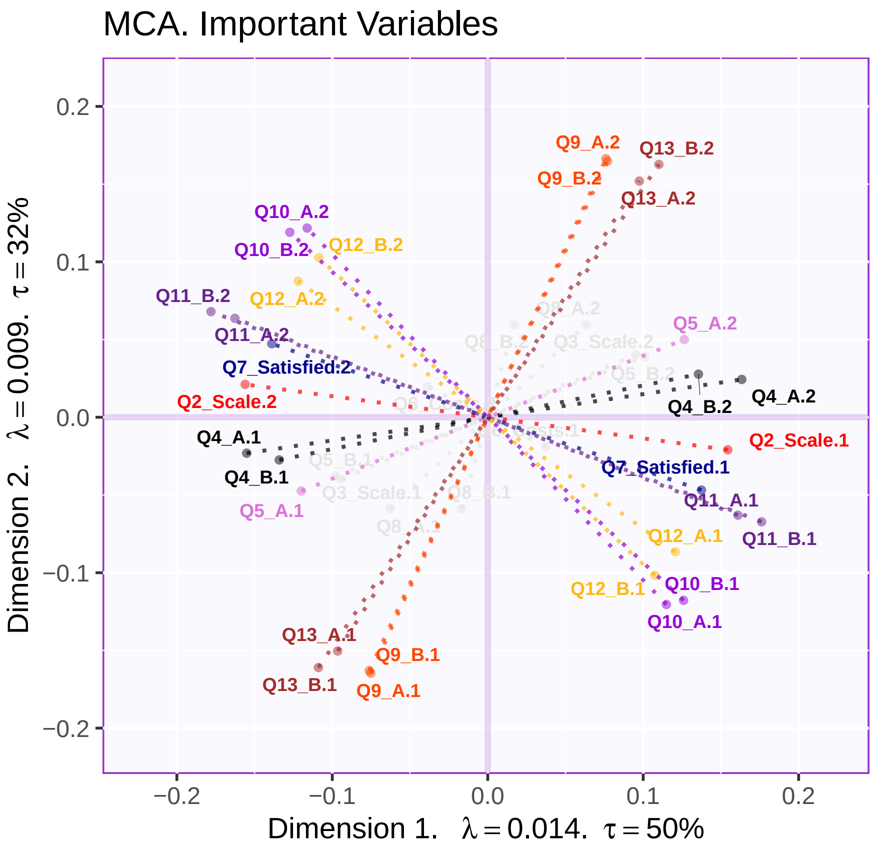
```
ctrK <- ctr4Variables(resMCA$ExPosition.Data$cj)

var12 <- data4PCCAR::getImportantCtr(ctr = ctrK,
eig = resMCA$ExPosition.Data$eigs)
importantVar <- var12$importantCtr.1or2
col4ImportantVar <- col4Var
col4NS <- 'gray90'
col4ImportantVar[!importantVar] <- col4NS



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

col4Levels.imp <- data4PCCAR::coloringLevels(rownames(Fj),
col4ImportantVar)
BaseMap.Fj.imp <- createFactorMap(X = Fj , # resMCA$ExPosition.Data$fj,
axis1 = axis1, axis2 = axis2,
title = 'MCA. Important Variables',
col.points = col4Levels.imp$color4Levels,
cex = 1,
col.labels = col4Levels.imp$color4Levels,
text.cex = 2.5,
force = 2)
b0010.BaseMap.Fj <- BaseMap.Fj.imp$zeMap + ctr.labels
print(b0010.BaseMap.Fj)
```



4.22 Levels of variables: map with important variables and lines

When observed carefully the question on cost which was the most frequently answered completely disappears in the plot.

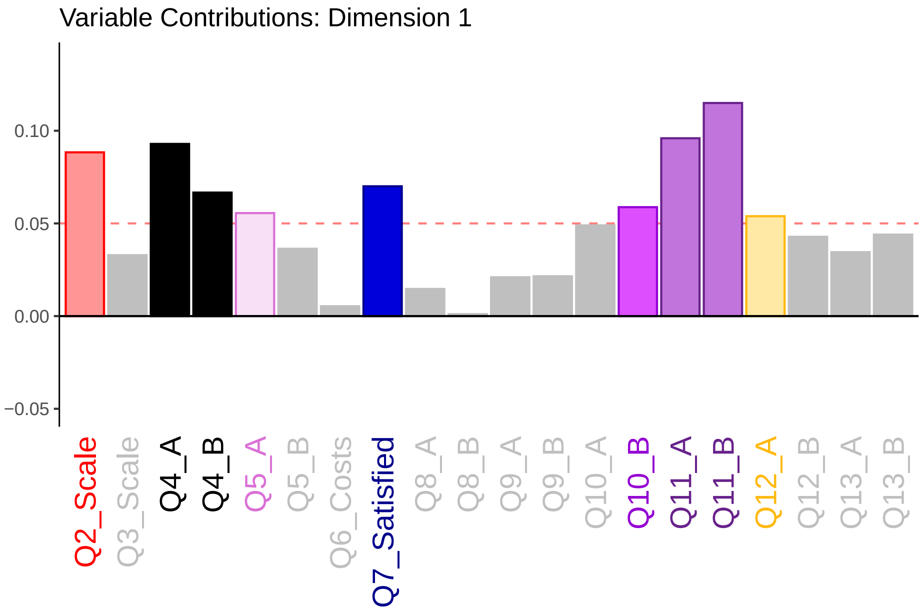
```
lines4J <- addLines4MCA(Fj, col4Var = col4Levels.imp$color4Variables, size = .7)
b0020.BaseMap.Fj <- b0010.BaseMap.Fj + lines4J
print( b0020.BaseMap.Fj)
```



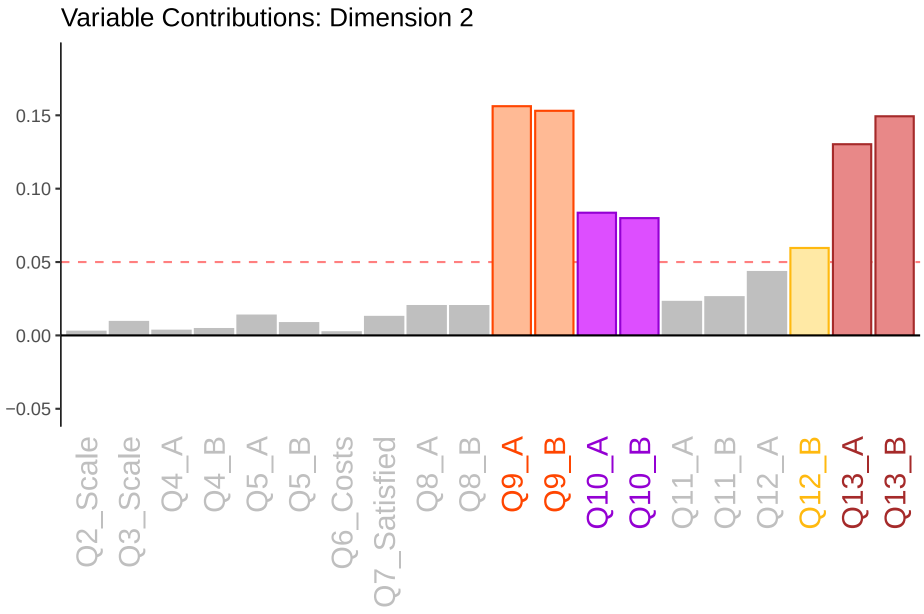
4.23 Contributions of variables

Component 1:For Dimension 1 current average speed (Q4_A & Q4_B) and Importance of speed after increase (Q11_A & Q11_B) contribute the most.Hence, those questions were rarely answered. Compenent 2: For Dimension 2 Importance of road distance(Q9_A & Q9_B) and Importance of cost (Q13_A & Q13_B) contribute the most.Hence, those questions were rarely answered.

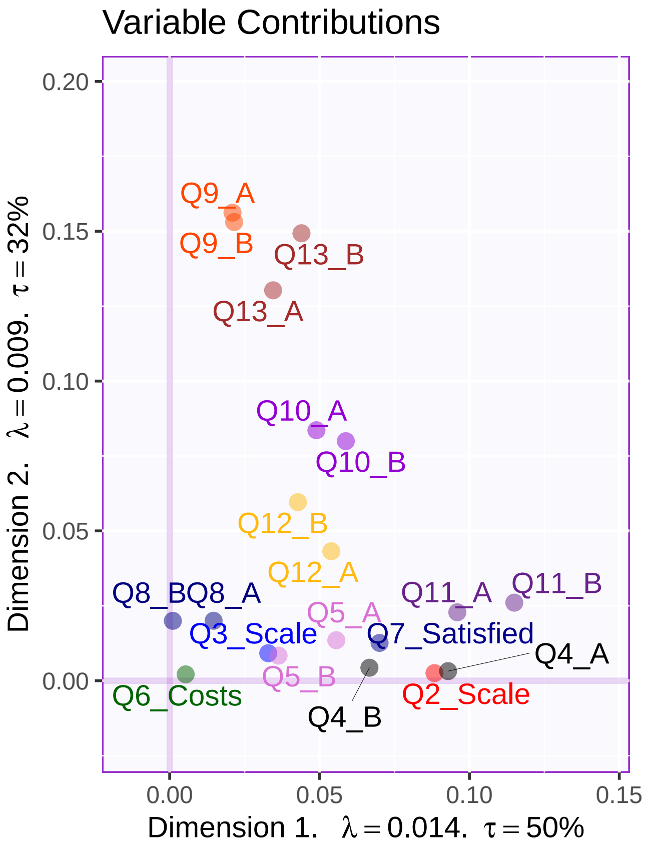
```
# Variables
col4Var <- c('red', 'blue', 'black',  'black','orchid','orchid','darkgreen','blue4','na

col4Levels <- data4PCCAR::coloringLevels(
rownames(resMCA$ExPosition.Data$fj), col4Var)
col4Labels <- col4Levels$color4Levels

ctrK <- ctr4Variables(resMCA$ExPosition.Data$cj)
# Do it ctr graph ----
# Exercise: Make a graph for the variable contributions

# Contribution plot for Component 1
ctrK1 <- ctrK[,1]
names(ctrK1) <- rownames(ctrK)

a0005.ctrK1 <- PrettyBarPlot2(ctrK1 ,
main = 'Variable Contributions: Dimension 1',
ylim = c(-.05, 1.2*max(ctrK1 )),
font.size = 5,
threshold = 1 / nrow(ctrK),
color4bar = gplots::col2hex(col4Var)
)
print(a0005.ctrK1)
```



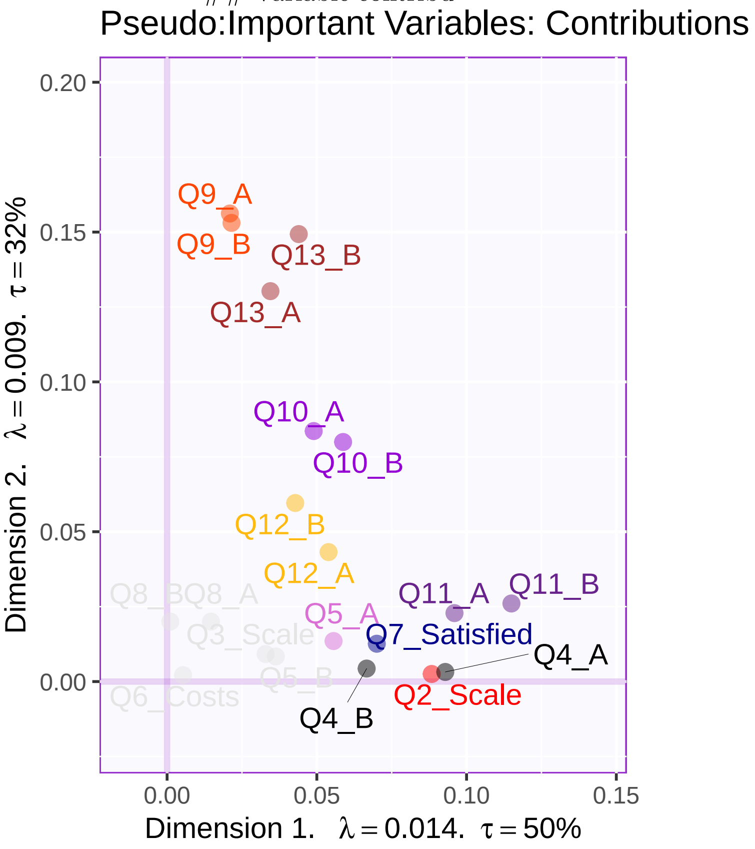
```
# Contribution plot for Component 1
ctrK2 <- ctrK[,2]
names(ctrK2) <- rownames(ctrK)
a0006.ctrK2 <- PrettyBarPlot2(ctrK2,
main = 'Variable Contributions: Dimension 2',
ylim = c(-.05, 1.2*max(ctrK2)),
threshold = 1 / nrow(ctrK),
font.size = 5,color4bar = gplots::col2hex(col4Var)
)

print(a0006.ctrK2)
```



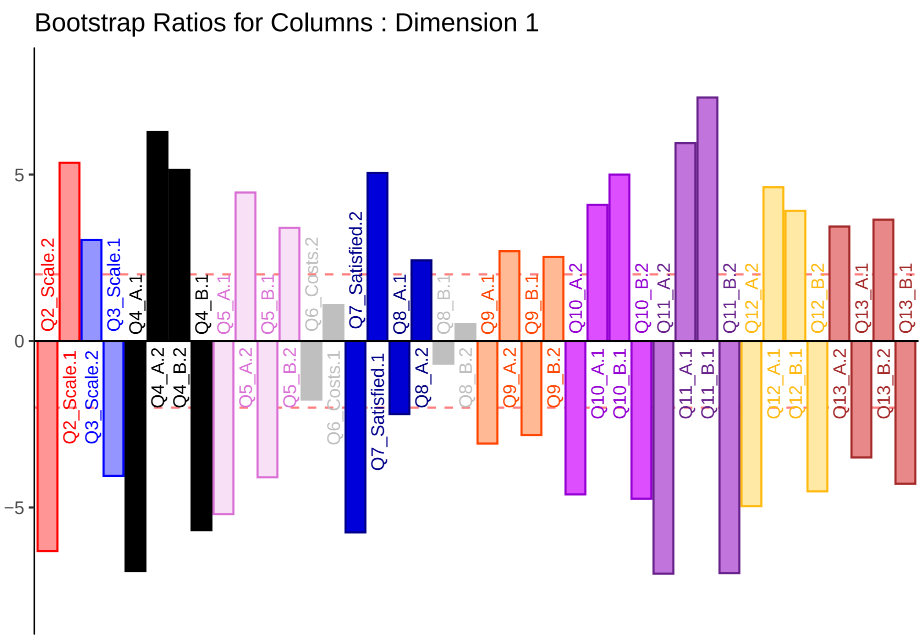
4.24 Pseudo Factor Plots


Variable Contributions

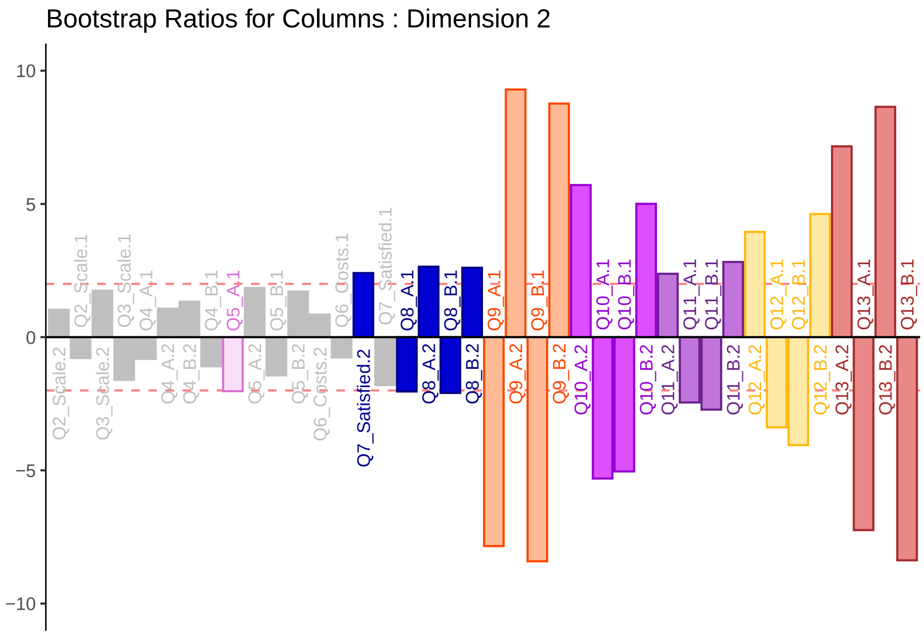
## Variable contribution plot with important variables only


Pseudo:Important Variables: Contributions

4.25 Bootstrap ratios for Levels of Variables



4.26 Bootstrap for dimension 2



It is observed that the Bootstrap ratios for the questions on Importance of road distance(Q9_A & Q9_B) and Importance of cost (Q13_A & Q13_B) are proven to be significant for both the groups A (indicated in blue) & B(indicated in green)

4.27 Summary

Component 1:

Rows: Group A Vs Group B Cols: Q11_B (Group B) VS Q11_B (Group A) (Importance of speed after increase)

Interpret: Participants of Group A & Group B had rarely answered the question on Importance of speed after increase

Component 2:

Rows: Group A Vs Group B Cols: Q9_B (Group B) VS Q9_B (Group A) (Importance of Road Distance)

Interpret: Majority of Group A & Group B rarely answered (less frequent) a) Importance of the road distance b) Cost for reconstructing the road.

Chapter 5

DiCA

5.1 Method DiCA on MCA: Note about MCA

Multiple correspondence analysis (MCA) is an extension of corre- spondence analysis (CA) which allows one to analyze the pattern of relationships of several categorical dependent variables. As such, it can also be seen as a generalization of principal component anal- ysis when the variables to be analyzed are categorical instead of quantitative. Because MCA has been (re)discovered many times, equivalent methods are known under several different names such as optimal scaling, optimal or appropriate scoring, dual scaling, homogeneity analysis, scalogram analysis, and quantification me- thod.

Technically MCA is obtained by using a standard correspon- dence analysis on an indicator matrix (i.e., a matrix whose entries are 0 or 1). The percentages of explained variance need to be cor- rected, and the correspondence analysis interpretation of inter- point distances needs to be adapted.

5.2 DiCA:

As the name indicates, discriminant correspondence analysis (DCA) is an extension of discriminant analysis (DA) and correspondence analysis (CA). Like discriminant analysis, the goal of DCA is to cat- egorize observations in predefined groups, and like correspon- dence analysis, it is used with nominal variables. The main idea behind DCA is to represent each group by the sum of its observations and to perform a simple CA on the groups by variables matrix. The original observations are then projected as supplementary elements and each observation is assigned to the closest group. The comparison between the a priori and the a posteriori classifications can be used to assess the quality of the discrimination. A similar procedure can be used to assign new observations to categories. The stability of the analysis can be evalu- ated using cross-validation techniques such as jackknifing or boot- strapping.

5.3 Data set: Drive.RData

Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test. It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

- Q_1: Implementation Decision
- Q2_Scale: Implementation Decision & judgement
- Q3_Scale: Road Distance
- Q4_A: Current Average Speed (A)
- Q4_B: Current Average Speed (B)
- Q5_A: Speed after reconstruction (A)
- Q5_B: Speed after reconstruction (B)
- Q6_Costs: Cost
- Q7_Satisfied: Satisfied with implementation and judgement
- Q8_A: Travel Time Saved (A)
- Q8_B: Travel Time Saved (B)
- Q9_A: Importance of Road Distance (A)
- Q9_B: Importance of Road Distance (B)
- Q10_A: Importance of speed before increase (A)
- Q10_B: Importance of speed before increase (B)
- Q11_A: Importance of speed after increase (A)
- Q11_B: Importance of speed after increase (B)
- Q12_A: Importance of time saving (A)
- Q12_B: Importance of time saving (B)
- Q13_A: Importance of Costs (A)
- Q13_B: Importance of Costs (B)

```
load("Drive.RData")

data.pca2 <- data.pca[1:191,2:21]

head(data.pca2)
```

```
##   Q2_Scale Q3_Scale  Q4_A  Q4_B  Q5_A  Q5_B Q6_Costs Q7_Satisfied  Q8_A
## 1    88.91    34.44  8.95 21.79 17.32 45.91    68.48        92.80 19.46
## 2    97.28    32.49 15.76 22.57 54.86 59.73   100.00        93.39 13.23
## 3    81.32    43.77  8.56 20.62 19.65 49.81    66.54        89.49 81.91
## 4    89.49    25.10  3.11 13.81  8.37 33.07    65.56        85.80 66.54
## 5    76.26    50.19  4.28 17.70 14.20 41.05    25.68        50.19 22.76
## 6    70.04    33.85  7.39 17.32 16.34 47.86    66.54        67.90 25.10
##    Q8_B  Q9_A  Q9_B  Q10_A  Q10_B  Q11_A  Q11_B  Q12_A  Q12_B Q13_A Q13_B
## 1 25.10 16.15 15.95 100.00  24.51 100.00  25.49 100.00  36.77 48.64 48.83
## 2 42.22 44.75 44.75  19.07  19.65  75.10  86.58  35.60  55.84  3.89  5.45
## 3 62.06 87.55 87.35 100.00 100.00 100.00 100.00 100.00 100.00  0.00  0.00
## 4 22.37  0.78  0.78  92.02 100.00  96.89 100.00   0.58   0.97  1.56  0.78
## 5 50.58 26.07 74.71  34.24  39.69  47.28  51.56  30.35  62.45 35.02 57.98
## 6 42.41 49.42 49.42  72.57  57.39  80.74  56.42  67.70  64.59 50.97 51.36
```

```
data.choice <- data.pca[["Q1_Choice"]] # Observation (row names)

#For more info,
#see: "?Drive.RData"
#also, type: "data.", and Tab to explore additional info
```

5.4 Run PCA

5.5 Recode for MCA

```
# recode for  MCA ----
# Initialized recoded df

meas.questions.Recoded <- data.frame(row.names = row.names(meas.questions))
```

5.6 Check the spearman's rank correlation

From the spearman's rank correlation we receive an accuracy of 86.62%.

```
## [1] 0.8662873
```

```
##
##  1  2
## 96 95
```

5.7 Disjunctively coded

A disjunctively coded matrix is obtained to observe the factor levels we liked to achieve in order to interpret which groups (either A or B) answered questions less and more frequently.

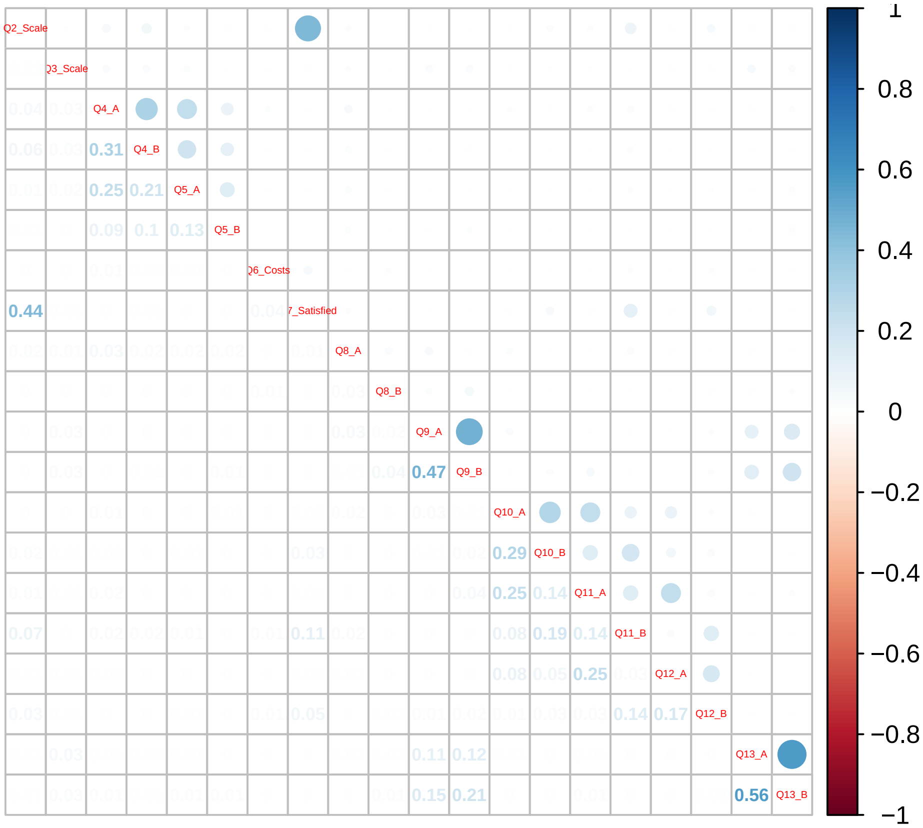
```
# Disjunctively coded
meas.questions.Recoded.dis <- makeNominalData(meas.questions.Recoded)

meas.questions.Recoded.dis[1:5,1:4]
```

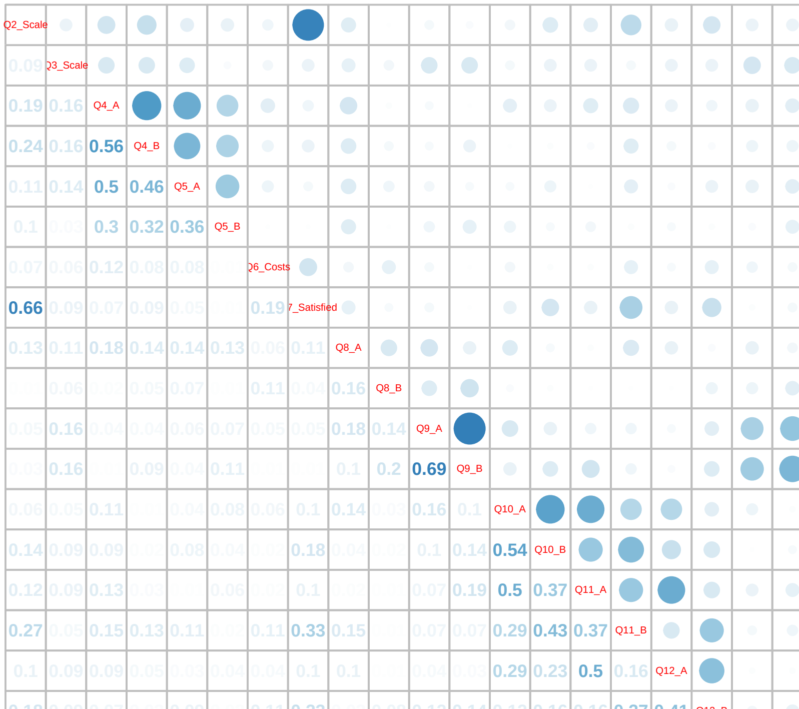
```
##   Q2_Scale.2 Q2_Scale.1 Q3_Scale.2 Q3_Scale.1
## 1          1          0          1          0
## 2          1          0          0          1
## 3          1          0          1          0
## 4          1          0          0          1
## 5          0          1          1          0
```

5.8 Run the MCA

5.9 Inference battery

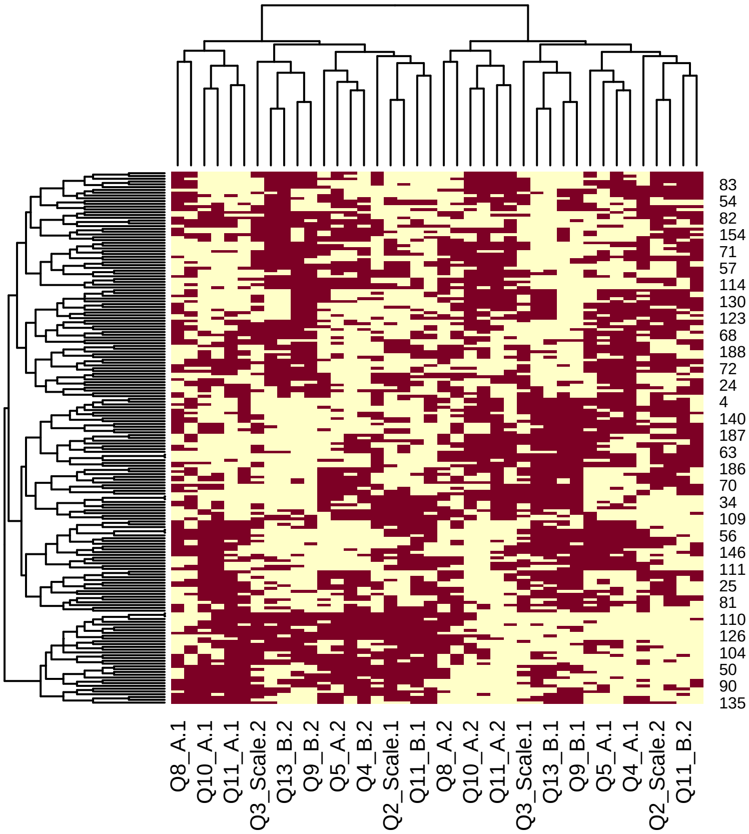
5.10 Pseudo Heat Map Correlation





## Heatmap after binning

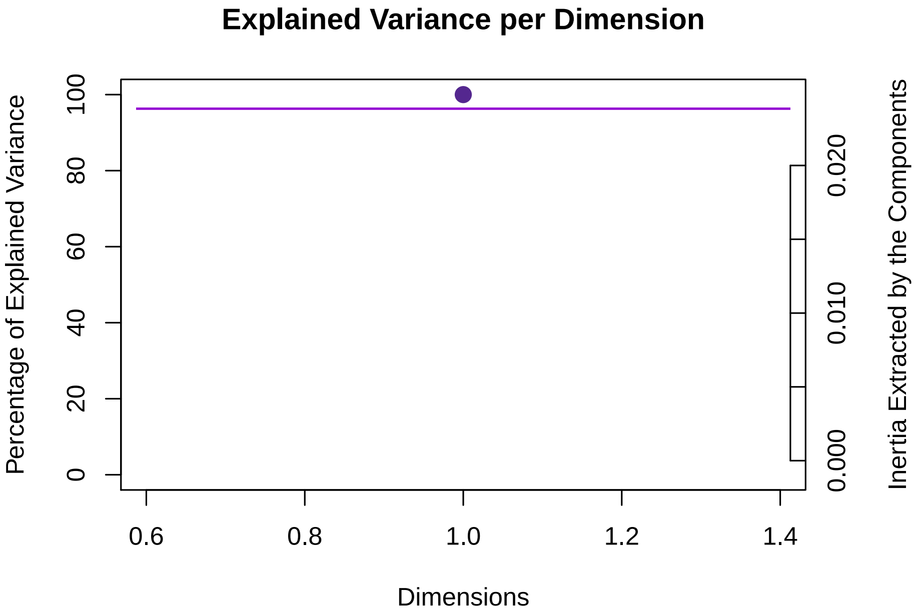
```
heatmap(x=meas.questions.Recoded.dis)
```



5.11 Scree Plot

The scree plot shows the eigenvalues, the amount of information on each component. The number of components (the dimensionality of the factor space) is min(nrow(DATA), ncol(DATA)) minus 1. According to the scree plot about 1 components must be interpreted.

```
# Scree ----
PlotScree(ev = resDiCA$TExPosition.Data$eigs,p.ev = resDiCA.inf$Inference.Data$components$p.vals,
```

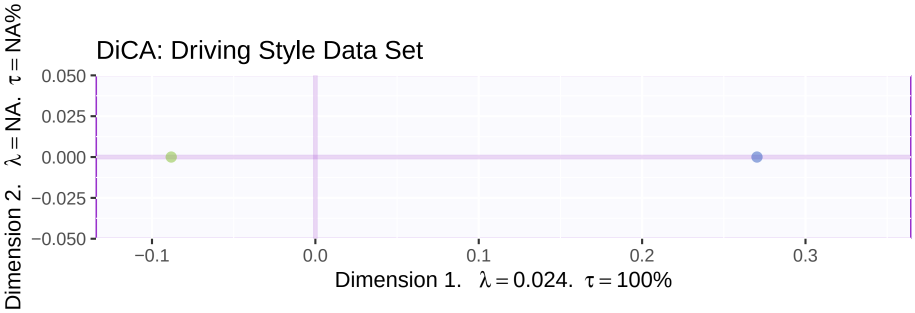


```
a001a.screePlot <- recordPlot()
```

5.12 I-set map

5.13 A graph of the observations

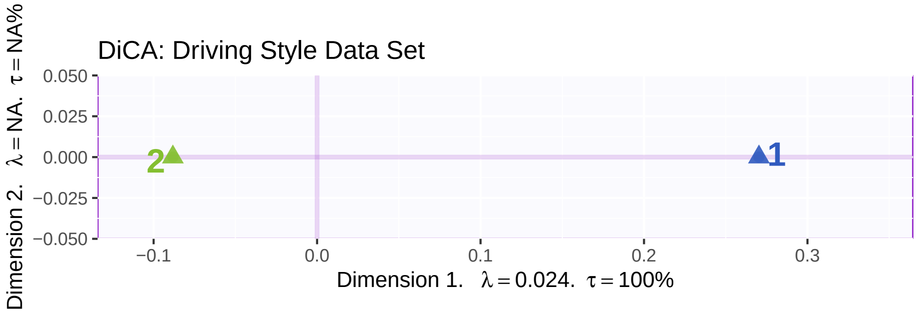
As we notice the observations we can clearly say that Group A (indicated in the color blue) and the Group B (indicated in the color green) are at the extremes of the dimension 1.



5.14 With group means

- Component 1:Group A Vs Group B
- Component 2:Group A Vs Group B
- Choice 1 (Group A) selected by the participants is indicated by the color Blue (Current Avg. Speed=15mph & Speed after reconstruction =29mph)
- Choice 2 (Group B)selected by the participants is indicated by the color Green (Current Avg. Speed=35mph & Speed after reconstruction =71mph)

```
fi.mean.plot <- createFactorMap(fi.mean,
                                alpha.points = 0.9,
                                col.points = grp.col[rownames(fi.mean)],
                                col.labels = grp.col[rownames(fi.mean)],
                                pch = 17,
                                cex = 3.5,
                                text.cex = 6)
fi.WithMean <- data.pca.Imap$zeMap_background + data.pca.Imap$zeMap_dots + fi.mean.plot$zeMap_dot
fi.WithMean
```



5.14.1 Confidence Intervals

Component 1:Group A Vs Group B is distributed along component 1 with Group A on one end and Group B on the other. Choice 1 (Group A) selected by the participants is indicated by the color Blue (Current Avg. Speed=15mph & Speed after reconstruction =29mph) Choice 2 (Group B)selected by the participants is indicated by the color Green (Current Avg. Speed=35mph & Speed after reconstruction =71mph)

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

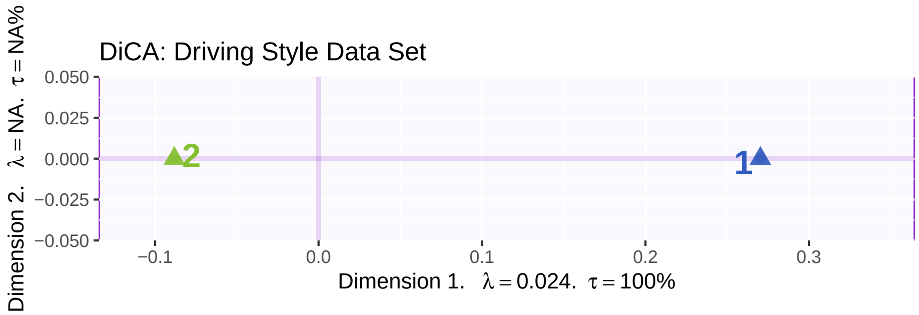
```
fi.WithMeanCI <- data.pca.Imap$zeMap_background + bootCI4mean + data.pca.Imap$zeMap_do
fi.WithMeanCI
```

```
## Warning: Removed 50 rows containing non-finite values (stat_ellipse).
```

```
## Warning: Computation failed in `stat_ellipse()`:
## missing value where TRUE/FALSE needed
```

```
## Warning: Removed 171 rows containing non-finite values (stat_ellipse).
```

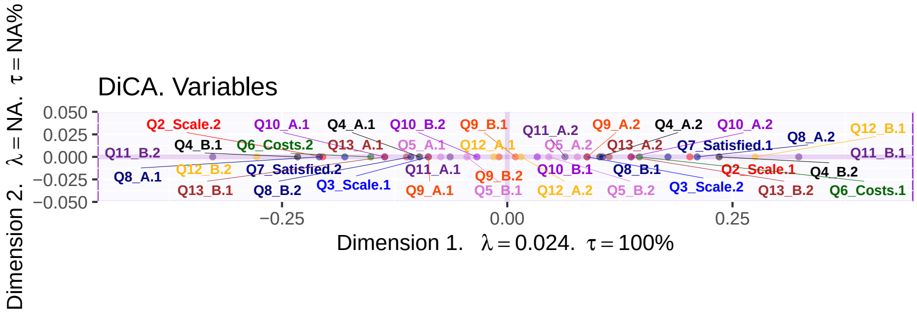
```
## Warning: Computation failed in `stat_ellipse()`:
## missing value where TRUE/FALSE needed
```



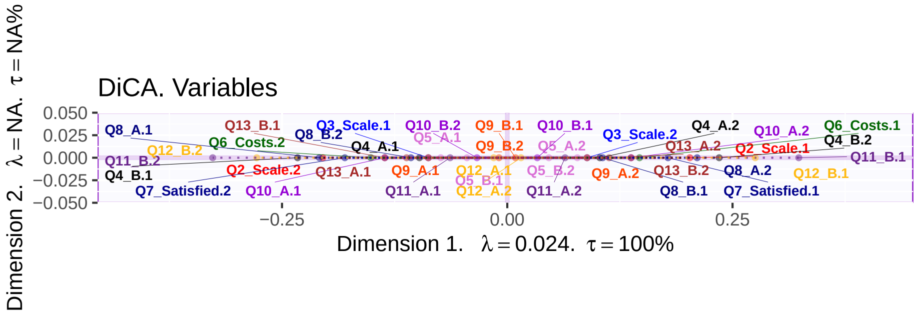
5.14.2 Column Factor Scores

The DiCA Variables are distributed along the component 1. Therefore it is noticed that, Q11_B (Speed after increase) were the questions that were rarely answered for both the groups. Also, Q9 on Importance of road distance & Q12 on Importance on total time saved were the questions that were frequently answered the participants.

```
# make the J-maps ----
 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
 b001.BaseMap.Fj
```



```
b003.MapJ
```

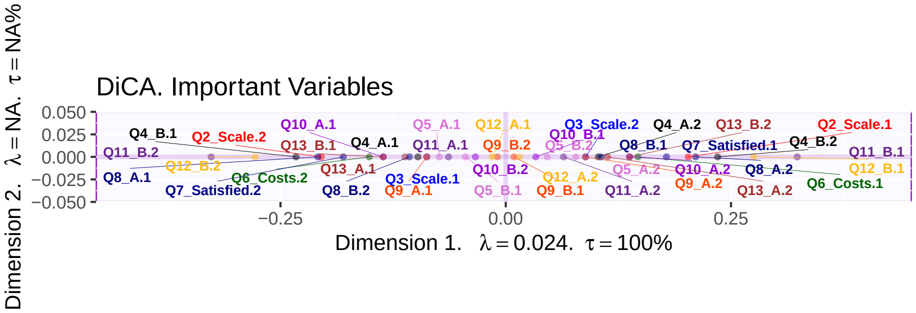


5.15 Levels of variables: map with only important variables

It is observed that all the variables are important for the component 1.

```
ctrK <- ctr4Variables(resDiCA$TExPosition.Data$cj)

var12 <- data4PCCAR::getImportantCtr(ctr = ctrK,
eig = resDiCA$TExPosition.Data$eigs)
importantVar <- var12$importantCtr.1or2
col4ImportantVar <- col4Var
col4NS <- 'gray90'
col4ImportantVar[!importantVar] <- col4NS



ctr.labels <- createxyLabels.gen(
1,2, lambda = resDiCA$TExPosition.Data$eigs,
tau = resDiCA$TExPosition.Data$t
)
```

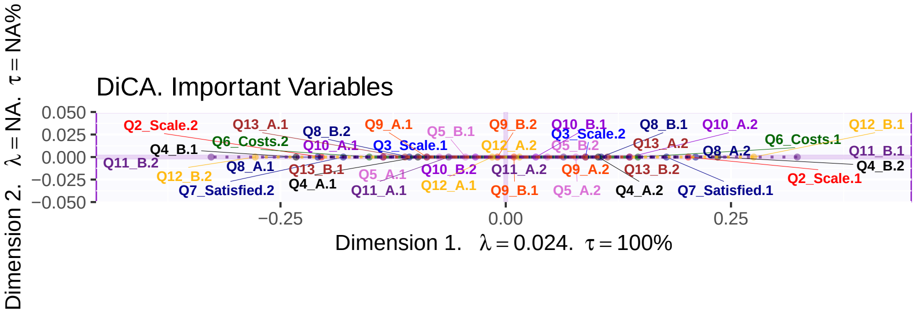
```
col4Levels.imp <- data4PCCAR::coloringLevels(rownames(Fj),
col4ImportantVar)
BaseMap.Fj.imp <- createFactorMap(X = Fj , # resDiCA$TExPosition.Data$fj,
axis1 = axis1, axis2 = axis2,
title = 'DiCA. Important Variables',
col.points = col4Levels.imp$color4Levels,
cex = 1,
col.labels = col4Levels.imp$color4Levels,
text.cex = 2.5,
force = 2)
b0010.BaseMap.Fj <- BaseMap.Fj.imp$zeMap + ctr.labels
b0010.BaseMap.Fj
```



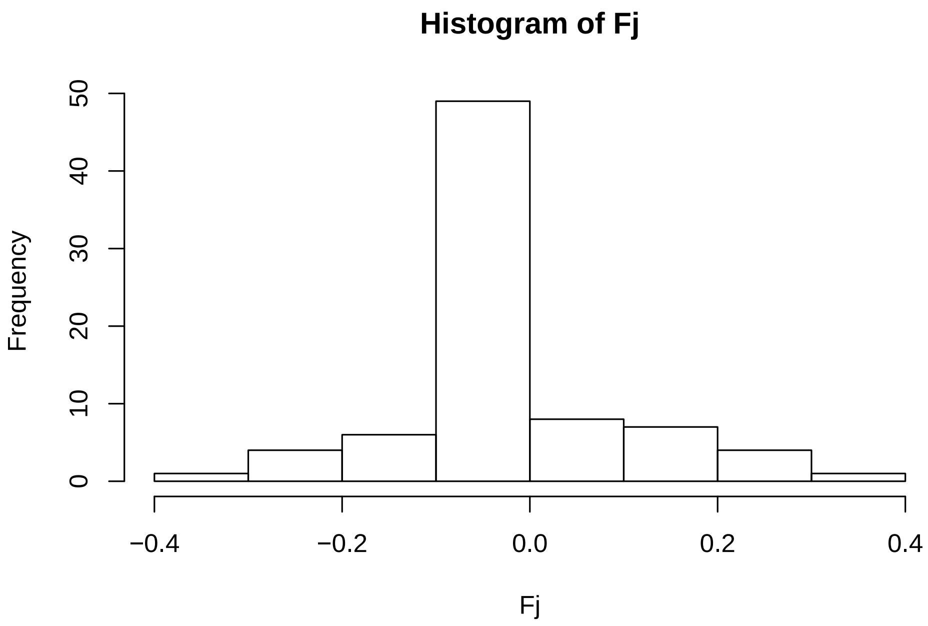
5.16 Levels of variables: map with important variables and lines

When observed carefully the question on cost which was the most frequently answered completely disappears in the plot.

```
lines4J <- addLines4MCA(Fj, col4Var = col4Levels.imp$color4Variables, size = .7)
b0020.BaseMap.Fj <- b0010.BaseMap.Fj + lines4J
b0020.BaseMap.Fj
```



```
hist(Fj)
```

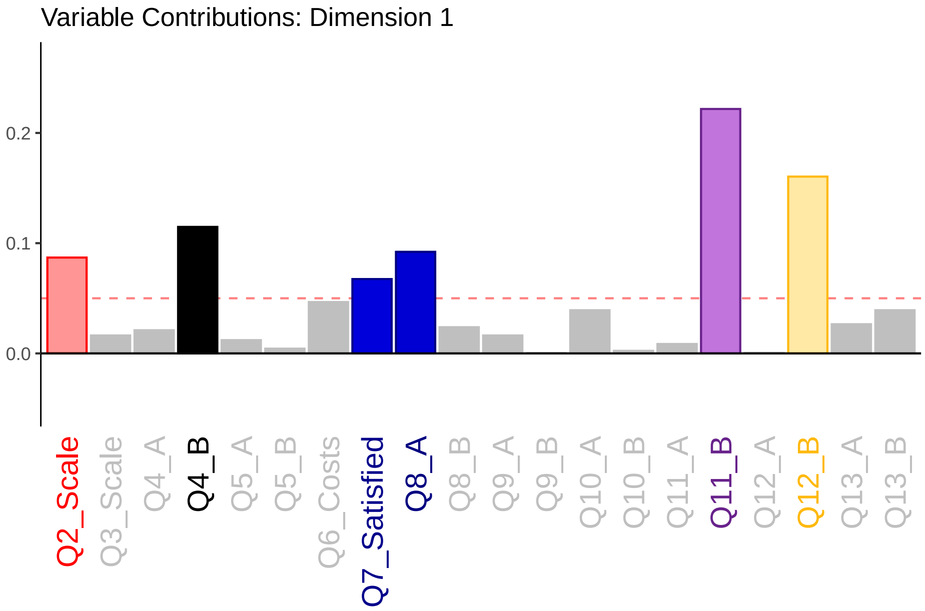


5.17 Contributions of variables

Component 1:For Dimension 1 Q4_B (Current Average speed for road B) ,Q11_B (Importance of speed after increase) and Q12_B (Importance of total time saved) contribute the most.Hence, those questions were rarely answered. Compenent 2: It does not exist

```
# Variables
col4Var <- c('red', 'blue', 'black',  'black','orchid','orchid','darkgreen','blue4','navyblue','r
col4Levels <- data4PCCAR::coloringLevels(
rownames(resDiCA$TExPosition.Data$fj), col4Var)
col4Labels <- col4Levels$color4Levels


ctrK <- ctr4Variables(resDiCA$TExPosition.Data$cj)
# Do it ctr graph ----
# Exercise: Make a graph for the variable contributions

# Contribution plot for Component 1
ctrK1 <- ctrK[,1]
names(ctrK1) <- rownames(ctrK)
```

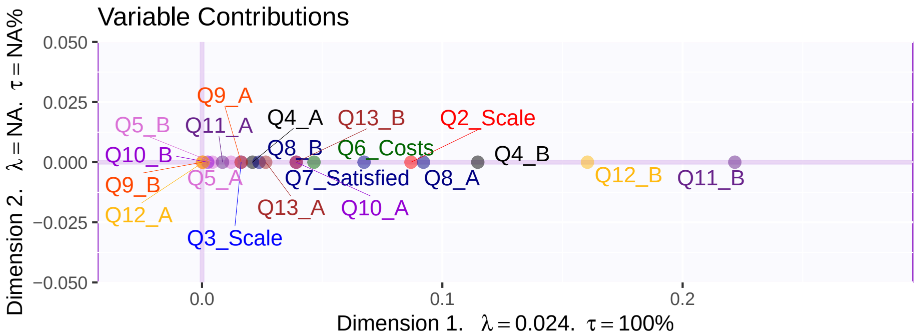
```
a0005.ctrK1 <- PrettyBarPlot2(ctrK1 ,
main = 'Variable Contributions: Dimension 1',
ylim = c(-.05, 1.2*max(ctrK1 )),
font.size = 5,
threshold = 1 / nrow(ctrK),
color4bar = gplots::col2hex(col4Var)
)
a0005.ctrK1
```



```
# Contribution plot for Component 1
ctrK2 <- ctrK[,2]
names(ctrK2) <- rownames(ctrK)
a0006.ctrK2 <- PrettyBarPlot2(ctrK2,
main = 'Variable Contributions: Dimension 2',
ylim = c(-.05, 1.2*max(ctrK2)),
threshold = 1 / nrow(ctrK),
font.size = 5,color4bar = gplots::col2hex(col4Var)
)
```

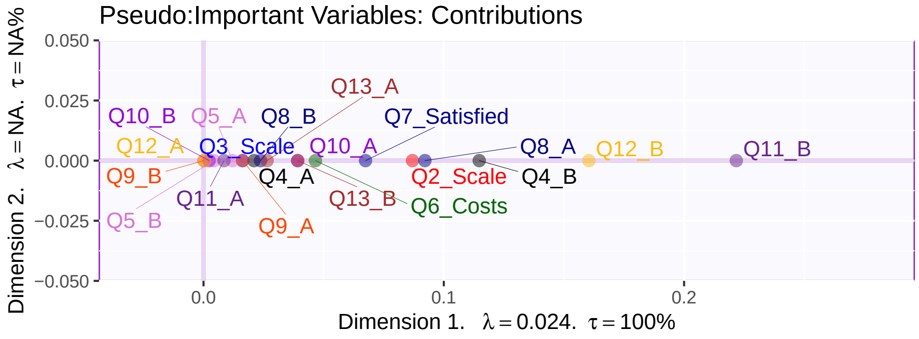
5.18 Pseudo Factor Plots

```
ctrV12 <- PTCA4CATA::createFactorMap(X = ctrK,
title = "Variable Contributions",
col.points = col4Var,
col.labels = col4Var,
alpha.points = 0.5,
cex = 2.5,
alpha.labels = 1,
text.cex = 4,
font.face = "plain",
font.family = "sans")
ctr.labels <- createxyLabels.gen(
1,2, lambda = resDiCA$TExPosition.Data$eigs,
tau = resDiCA$TExPosition.Data$t
)
a0007.Var.ctr12 <- ctrV12$zeMap + ctr.labels
#
a0007.Var.ctr12
```



## Variable contribution plot with important variables only

```
var12 <- data4PCCAR::getImportantCtr(ctr = ctrK,
eig = resDiCA$TExPosition.Data$eigs)
importantVar <- var12$importantCtr.1or2
col4ImportantVar <- col4Var
col4NS <- 'gray90'
col4ImportantVar[!importantVar] <- col4NS

ctrV12.imp <- PTCA4CATA::createFactorMap(X = ctrK,
title = "Pseudo: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 + ctr.labels
#
a0008.Var.ctr12.imp
```



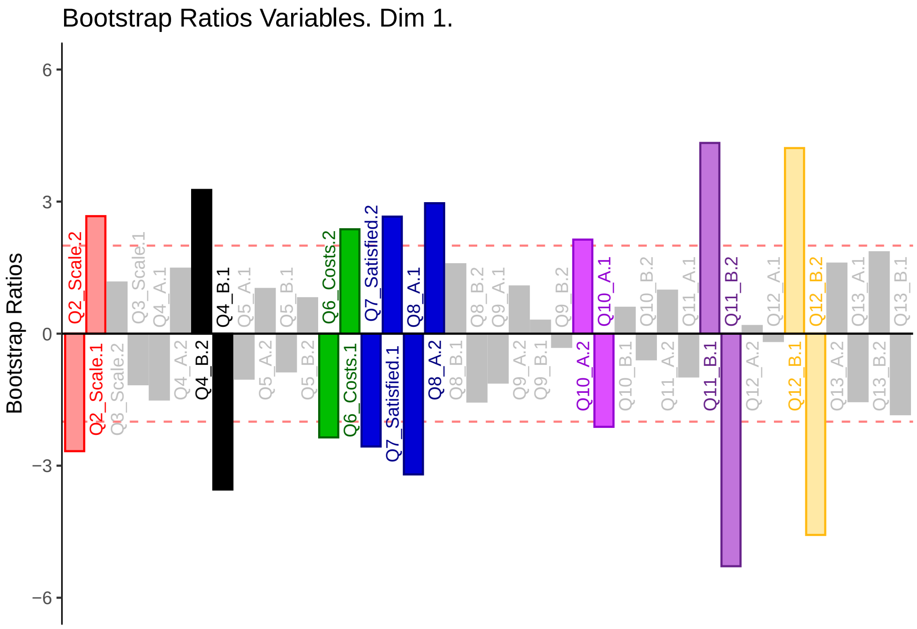
5.19 Bootstrap ratios for Levels of Variables

It is observed that the Bootstrap ratios for the questions on Importance of road distance(Q9_A & Q9_B), Q11_B (Importance of speed after increase) and Q12_B (Importance of total time saved) are proven to be significant for both the groups A (indicated in blue) & B (indicated in green)

Component 2: It does not exist.

It is observed that the Bootstrap ratios for the questions on Importance of road distance(Q9_A & Q9_B) and Importance of cost (Q13_A & Q13_B) are proven to be significant for both the groups A (indicated in blue) & B(indicated in green)

```
col4Var2 <- c('red','red', 'blue', 'blue', 'black', 'black',  'black', 'black','orchid
#BR. 1 ====

BRj <- resDiCA.inf$Inference.Data$boot.data$fj.boot.data$tests$boot.ratios
# BR1
d001.plotBRj.1 <- PrettyBarPlot2(
  bootratio = BRj[,1],
  threshold = 2,
  ylim = NULL,
  color4bar = gplots::col2hex(col4Var2),
  color4ns = "gray75",
  plotnames = TRUE,
  main = 'Bootstrap Ratios Variables. Dim 1.',
  ylab = "Bootstrap Ratios")
d001.plotBRj.1
```



5.20 Fixed Model

From the Fixed model we can clearly say that, an accuracy of 71.72% is acheived and it could be used for training of the model.

```
#Fixed Model

row.names(resDiCA.inf$Inference.Data$loo.data$fixed.confuse) <- c("1","2")
colnames(resDiCA.inf$Inference.Data$loo.data$fixed.confuse) <- c("1","2")
resDiCA.inf$Inference.Data$loo.data$fixed.confuse
```

```
##    1   2
## 1 32  39
## 2 15 105
```

```
resDiCA.inf$Inference.Data$loo.data$fixed.acc
```

```
## [1] 0.7172775
```

5.21 Random Model

However, the Random model is used for validation and an accuracy of 42.93% is obtained.

```
#Random Model
row.names(resDiCA.inf$Inference.Data$loo.data$loo.confuse) <- c("1", "2")
colnames(resDiCA.inf$Inference.Data$loo.data$loo.confuse) <- c("1", "2")
resDiCA.inf$Inference.Data$loo.data$loo.confuse
```

```
##    1  2
## 1 29 45
## 2 18 99
```

```
resDiCA.inf$Inference.Data$loo.data$loo.acc
```

```
## [1] 0.6701571
```

5.22 Summary

Component 1: It is evident that row factor scores are far apart, drawing a line between Group A and Group B.

The DiCA Variables are distributed along the component 1. Therefore it is noticed that, Q11_B (Speed after increase) were the questions that were rarely answered for both the groups. Also, Q9 on Importance of road distance & Q12 on Importance on total time saved were the questions that were frequently answered by the participants.

Therefore, it is observed that the Bootstrap ratios for the questions on Importance of road distance(Q9_A & Q9_B), Q11_B (Importance of speed after increase) and Q12_B (Importance of total time saved) are proven to be significant for both the groups A (indicated in blue) & B (indicated in green) and were rarely answered.

Chapter 6

Partial Least Square Correlation

6.1 Data set 1: Drive.RData

We find the correlation/commonalities between the two tables(Dataset 1 & Dataset 2). Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test.

It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

- Q14_Choice: Implementation Decision
- Q15_Scale: Implementation Decision & judgement
- Q16_Scale: Road Distance
- Q17_A: Current Average Speed (A)
- Q17_B: Current Average Speed (B)
- Q18_A: Speed after reconstruction (A)
- Q18_B: Speed after reconstruction (B)
- Q19_Costs: Cost
- Q20_Satisfied: Satisfied with implementation and judgement
- Q21_A: Travel Time Saved (A)
- Q21_B: Travel Time Saved (B)
- Q22_A: Importance of Road Distance (A)
- Q22_B: Importance of Road Distance (B)
- Q23_A: Importance of speed before increase (A)
- Q23_B: Importance of speed before increase (B)
- Q24_A: Importance of speed after increase (A)
- Q24_B: Importance of speed after increase (B)
- Q25_A: Importance of time saving (A)

- Q25_B: Importance of time saving (B)
- Q26_A: Importance of Costs (A)
- Q26_B: Importance of Costs (B)

6.2 Data set 2: State-Trait Anxiety Inventory (STAI)

The State-Trait Anxiety Inventory (STAI) is a psychological inventory based on a 4-point Likert scale (6-point in this study).It consists of 40 questions on a self-report basis.The STAI measures two types of anxiety – state anxiety, or anxiety about an event, and trait anxiety, or anxiety level as a personal characteristic. Higher scores are positively correlated with higher levels of anxiety.

```r
load("Drive.RData")
data1 <- data.survey2
data2 <- data.survey3

data.design <- data.pca$Q1_Choice

data.design <- as.matrix(as.numeric(data.pca$Q1_Choice))
# make the design into a vector
data.design.vec <- colSums(t(data.design)*c(1:ncol(data.design)))

head(data1)
```

```
##   Q14_Choice Q15_Scale Q16_Scale Q17_A Q17_B Q18_A Q18_B Q19_Scale
## 1          1    100.00     34.63 11.87 26.26 21.60 48.44     33.07
## 2          1     28.40     50.78 11.09 17.70 59.92 48.64     56.81
## 3          1    100.00     64.01  8.56 27.24 51.95 28.02     67.12
## 4          1    100.00     28.60  6.03 21.01 11.48 32.68     71.79
## 5          1     81.13     61.48  8.56 19.07 13.81 28.02     31.32
## 6          1     97.86     30.54 11.67 21.79 19.65 50.19     65.37
##   Q20_Scale Q21_A Q21_B Q22_A Q22_B  Q23_A Q23_B  Q24_A  Q24_B  Q25_A
## 1    100.00 63.04 28.79 65.37 48.25 100.00 36.19 100.00  34.05 100.00
## 2     89.88 66.73 50.19 10.70 12.65   3.89  3.50  31.71  44.75  56.61
## 3    100.00 46.11 24.32 92.80 93.39 100.00 99.42 100.00 100.00 100.00
## 4    100.00 47.67 25.10  0.58  0.58   0.39  0.58   0.19   0.58 100.00
## 5     64.79 69.84 32.88 60.89 43.58  57.59 36.58  63.04  45.53  70.04
## 6     92.80 66.54 30.74 50.39 50.19  62.45 47.86  57.20  70.43  87.94
##    Q25_B Q26_A Q26_B
## 1  51.56 34.24 59.14
## 2  44.94  2.33  5.06
## 3 100.00  0.00  0.19
## 4  87.35  1.36  1.36
## 5  40.47 45.14 44.94
## 6  41.83 48.83 49.42
```

```
head(data2)
```

```
##   STAI1_1 STAI1_2 STAI1_3 STAI1_4 STAI1_5 STAI1_6
## 1       4       1       1       4       4       1
## 2       4       1       1       4       3       1
## 3       4       1       1       3       3       1
## 4       2       2       2       2       3       2
## 5       3       2       1       3       2       2
## 6       2       1       1       3       3       1
```

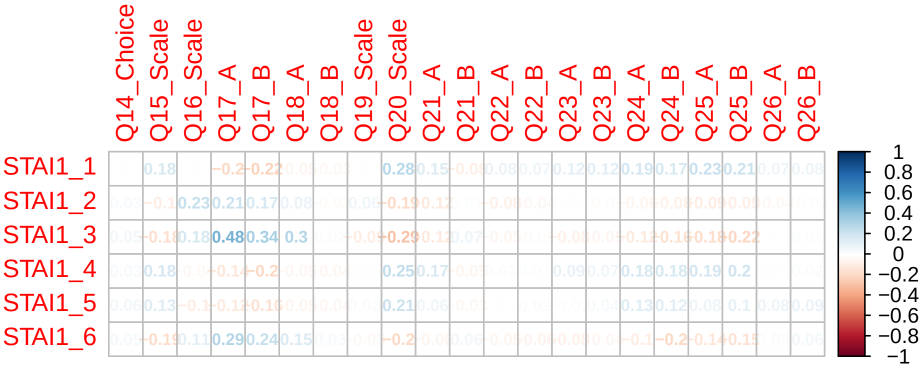
6.3 Method PLSC

Partial Least Square Correlation (PLSC) generalizes the idea of correlation between two variables to two tables. It was originally developed by Tucker (51), and refined by Bookstein . This technique is particularly popular in brain imaging because it can handle the very large data sets gener- ated by these techniques and can easily be adapted to handle sophisticated experimental designs. For PLSC, both tables play a similar role (i.e., both are dependent variables) and the goal is to analyze the information common to these two tables. This is obtained by deriving two new sets of variables (one for each table) called latent variables that are obtained as linear combinations of the original variables. These latent variables, which describe the observations, are required to "explain" the largest portion of the covariance between the two tables. The original variables are described by their saliences. For each latent variable, the X or Y variable saliences have a large magnitude, and have large weights for the computation of the latent variable. Therefore, they have contributed a large amount to creating the latent variable and should be used to interpret that latent variable (i.e., the latent variable is mostly "made" from these high contributing variables). By analogy with principal component analysis, the latent variables are akin to factor scores and the saliences are akin to loadings.

6.4 The data pattern

In PLSC, we analyze the covariance (cross-product) of the two quantitative table. As a result, we try to plot this covariance matrix.From the Correlation plot, it is observed that participants had shown strong levels of anxiety, typically level 3 for the questions on current average speed (Q17_A & Q17_B) for both the roads. Also, they had shown negative correlation for the question on Satisfaction and judgement (Q20_Scale).

```
# Compute the covariance matrix
XY.cor <- cor(data1,data2)
# Plot it with corrplot
corrplot(t(XY.cor), method = "number",tl.cex = 1,number.cex = 0.7)
```



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

6.5 Analysis

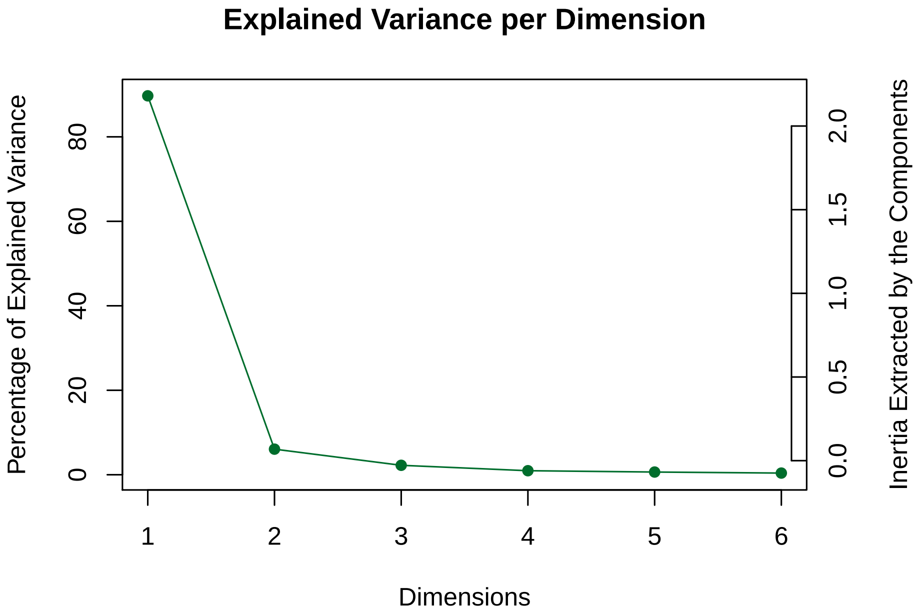
```
# run PLSC
pls.res <- tepPLS(data1,data2, DESIGN = data.design, make_design_nominal = FALSE, grapl
```

```
## [1] "Group Assignment Matrix is incorrect: too many items in the DESIGN matrix! Crea
```

6.6 Scree Plot

From the Scree Plot, it is evident that just one/two dimension needs to interpreted. Therefore, inference is needed to help us understand which dimensions are important to us.

```
my.scree <- PlotScree(ev = pls.res$TExPosition.Data$eigs)
```



6.7 The number of eigen values:

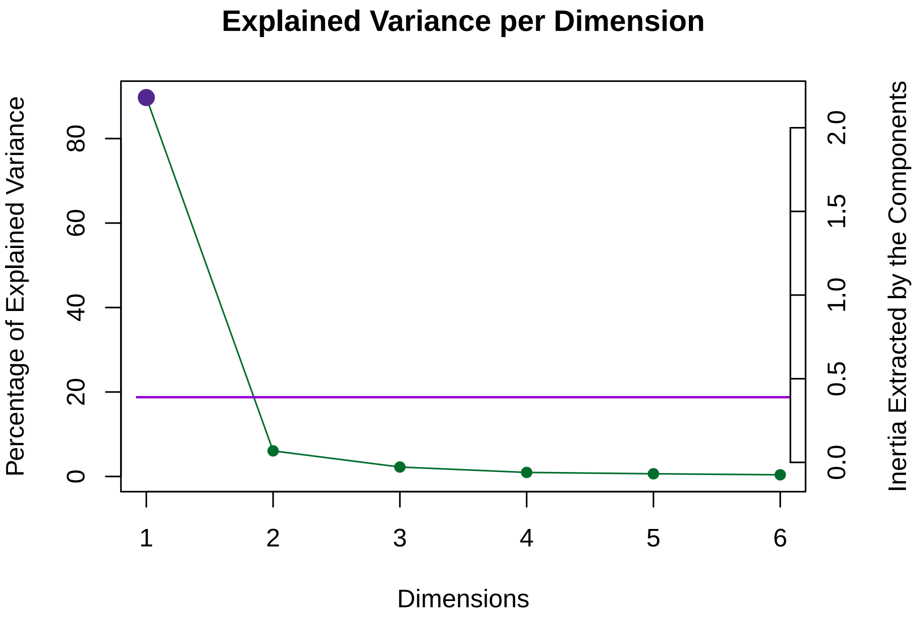
6.8 Scree (Permuted)

From the scree plot, we can say that only 1 component can be interpreted.

```
# First: Go for a permutation test
resPerm4PLSC <- perm4PLSC(data1, # First Data matrix
                          data2, # Second Data matrix
                          nIter = 1000 # How mny iterations
                          )
# to see what results we have
resPerm4PLSC
```

```
## ------------------------------------------------------------------------------
##  Results of Permutation Test for PLSC of X'*Y = R
##  for Omnibus Inertia and Eigenvalues
## ------------------------------------------------------------------------------
## $ fixedInertia      the Inertia of Matrix X
## $ fixedEigenvalues  an L*1 vector of the eigenvalues of X
## $ pOmnibus          the probablity associated to the Inertia
## $ pEigenvalues      an L* 1 matrix of p for the eigenvalues of X
## $ permInertia       vector of the permuted Inertia of X
## $ permEigenvalues   matrix of the permuted eigenvalues of X
## ------------------------------------------------------------------------------
```

```
# Now, can you add this to your "my.scree" scree plot?
#-----------------------------
my.scree <- PlotScree(ev = pls.res$TExPosition.Data$eigs,p.ev = resPerm4PLSC$pEigenvalu
```



```
#-----------------------------

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

6.9 Factor scores maps

Exercise 2: Make the graphs for PLSC

plot1: the observations viewed from the 1st latent variable of both tables.

plot2: the observations viewed from the 2nd latent variable of both tables.

plot3: the column loadings of the 1st component of X and Y.-> colors of the columns are in col4Var and col4Var2.

plot4: the column loadings of the 2nd component of X and Y. –> colors of the columns are in col4Var and col4Var2.

First, we start from computing all the things we need to create the plots.

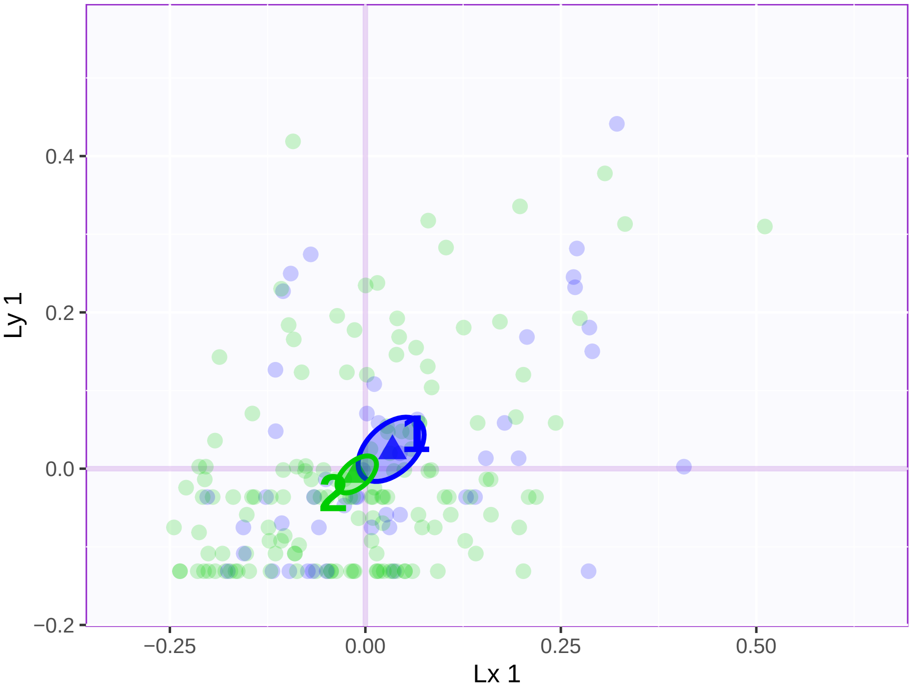
Next, we can start plotting:

6.10 plot1

Component 1: Group A vs Group B : Lx1 vs Ly1

Group A: Participants who selected Road A are indicated in the color blue. Group B: Participants who selected Road B are indicated in the color green.

```
plot.lv1 <- createFactorMap(latvar.1,
                         col.points =group.levels,
                         col.labels =group.levels,
                         alpha.points = 0.2
                         )

plot1.mean <- createFactorMap(lv.1.group,
                              col.points = col4Means,
                              col.labels = col4Means,
                              cex = 4,
                              pch = 17,text.cex = 8,
                              alpha.points = 0.8)

plot1.meanCI <- MakeCIEllipses(lv.1.group.boot$BootCube[,c(1:2),], # get the first two components
                              col = col4Means,
                              names.of.factors = c("Lx 1", "Ly 1")
                              )

plot1 <- plot.lv1$zeMap_background + plot.lv1$zeMap_dots + plot1.mean$zeMap_dots + plot1.mean$zeM
plot1
```

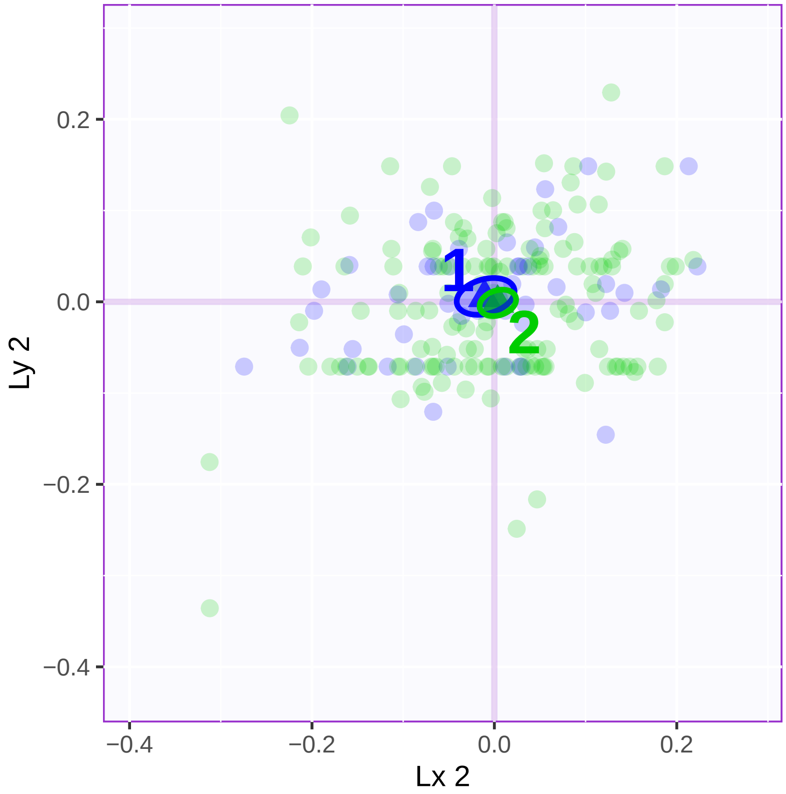


6.11 plot2

Component 2: Group A vs Group B : Lx2 vs Ly2

Group A: Participants who selected Road A are indicated in the color blue. Group B: Participants who selected Road B are indicated in the color green.

```
# For the first plot, the first component of the latent variable of X is the x-axis, a
latvar.2 <- cbind(pls.res$TExPosition.Data$lx[,2],pls.res$TExPosition.Data$ly[,2])
colnames(latvar.2) <- c("Lx 2", "Ly 2")

# compute means
lv.2.group <- getMeans(latvar.2,data.design)
col4Means = recode(rownames(lv.2.group), '1' = 'blue', '2' = 'green3')

# get bootstrap intervals of groups
lv.2.group.boot <- Boot4Mean(latvar.2, data.design)
colnames(lv.2.group.boot$BootCube) <- c("Lx 2", "Ly 2")



plot.lv2 <- createFactorMap(latvar.2,
                         col.points = group.levels,
                         col.labels = group.levels,
                         alpha.points = 0.2
                         )

plot2.mean <- createFactorMap(lv.2.group,
                              col.points = col4Means,
                              col.labels = col4Means,
                              cex = 4,
                              pch = 17,text.cex = 8,
                              alpha.points = 0.8)

plot2.meanCI <- MakeCIEllipses(lv.2.group.boot$BootCube[,c(1:2),], # get the first two components
                              col = col4Means ,
                              names.of.factors = c("Lx 2", "Ly 2")
                              )

plot2 <- plot.lv2$zeMap_background + plot.lv2$zeMap_dots + plot2.mean$zeMap_dots + plot2.mean$zeM
plot2
```



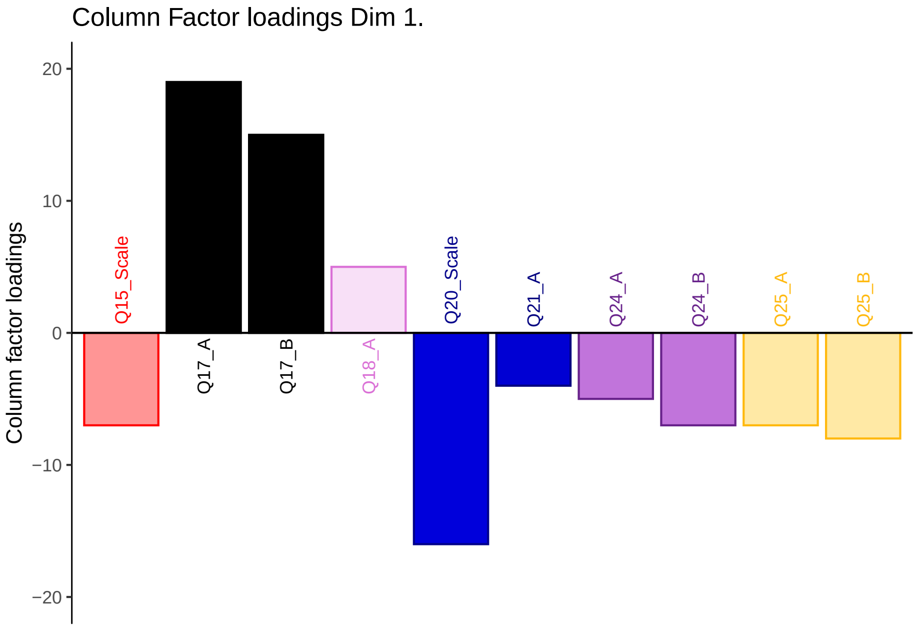
6.12 plot3 : Saliences for Table 1

The column loadings of the 1st component of X and Y. Column Factor Scores for Table 1:

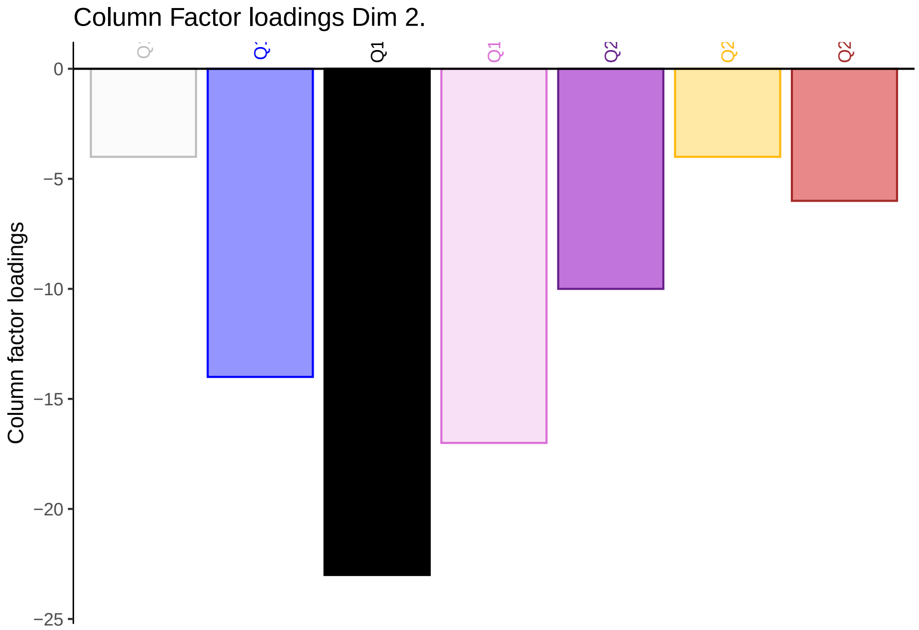
Component 1: Q17_A & B (Current Average Speed) VS Q20_Scale (Satisfied with implementation and judgement)

Component 2: Q16, Q17A & Q18A: Road Distance, Current Avg. Speed and Speed after reconstruction

```
col4Var <- c('grey','red', 'blue', 'black',  'black','orchid','orchid','darkgreen','blu

#_____________________________________________________________________
Fj <- pls.res$TExPosition.Data$fj
Fi <- pls.res$TExPosition.Data$fi



column1 <- pls.res$TExPosition.Data$ci
Fi <- pls.res$TExPosition.Data$fi
signed.column1 <- column1 * sign(Fi)

#round(100*signed.column1[,1])

c001.plotCol.11 <- PrettyBarPlot2(
                       bootratio = round(100*signed.column1[,1]),

                       line.alpha = 0,
                       color4bar = gplots::col2hex(col4Var),
                       color4ns =  gplots::col2hex(col4Var),
                       signifOnly=TRUE,
                       plotnames = TRUE,
                       main = 'Column Factor loadings Dim 1.',
                       ylab = "Column factor loadings")

c001.plotCol.11
```



```
c001.plotCol.22 <- PrettyBarPlot2(
                       bootratio = round(100*signed.column1[,2]),
                        line.alpha = 0,
                       color4bar = gplots::col2hex(col4Var),
                       color4ns =  gplots::col2hex(col4Var),
                       signifOnly=TRUE,
                       plotnames = TRUE,
                       main = 'Column Factor loadings Dim 2.',
                       ylab = "Column factor loadings")

c001.plotCol.22
```



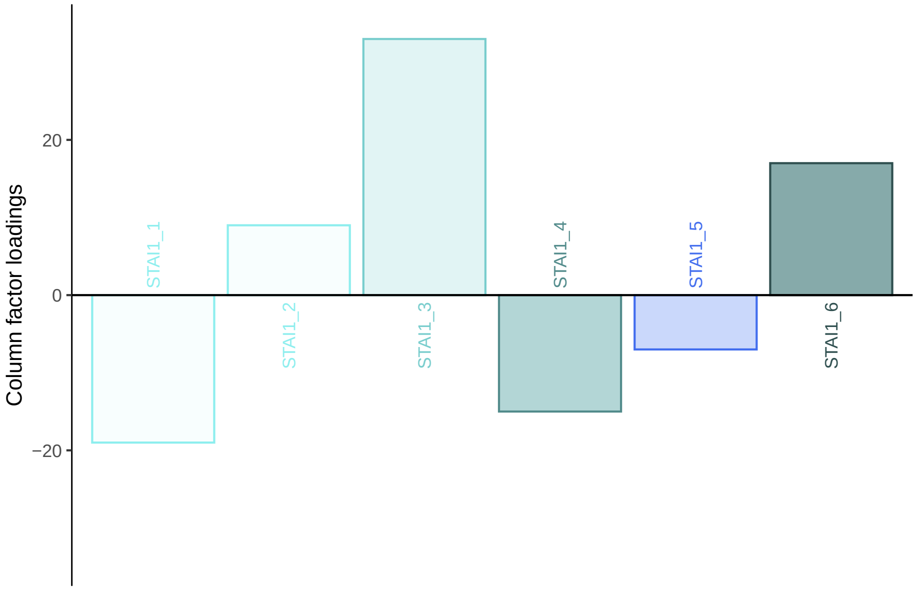
6.13 plot4 : Saliences for Table 2

The column loadings of the 2nd component of X and Y

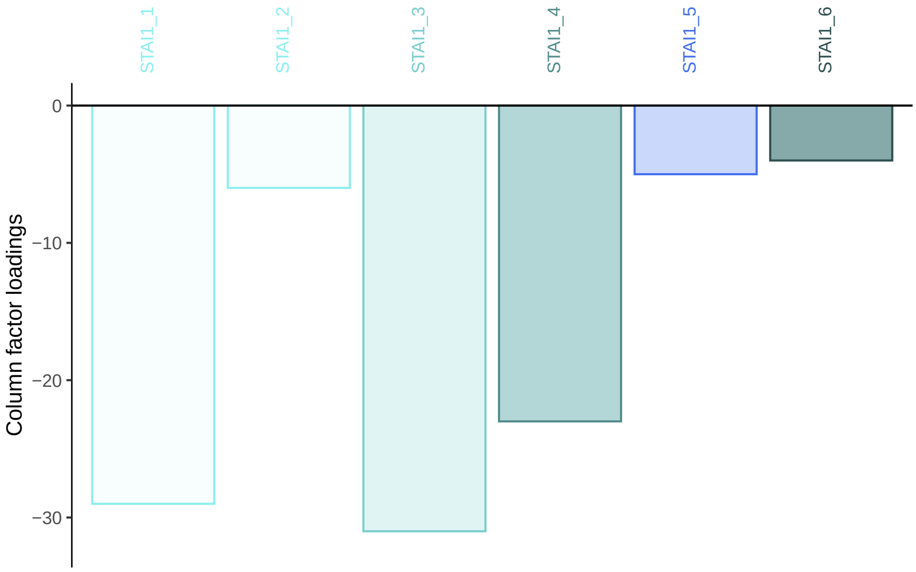
Component 1: State-Trait Anxiety Inventory (STAI) Level 1 & 6 VS Level 3

Component 2: State-Trait Anxiety Inventory (STAI) Level 1,3 & 4

Column Factor loadings Dim 1.



Column Factor loadings Dim 2.



6.14 Contributions and bootstrap ratios barplots

6.15 Contribution barplots

For PLSC, we also plot the contributions for both rows and columns

6.16 Dataset/Table 1:

Component 1: It is quite evident that Q17_A & B (Current Average Speed) VS Q20_Scale (Satisfied with implementation and judgement) contribute the most to the 1st Component.

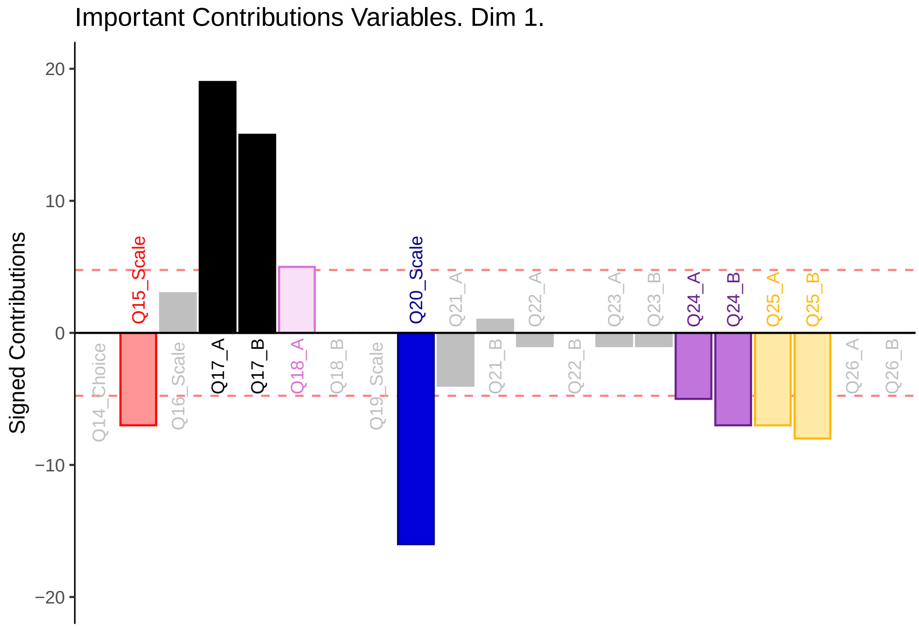
Component 2: It is quite evident that Q16, Q17A & Q18A: Road Distance, Current Avg. Speed and Speed after reconstruction contribute the most to the 2nd Component.

6.17 Dataset/Table 2:

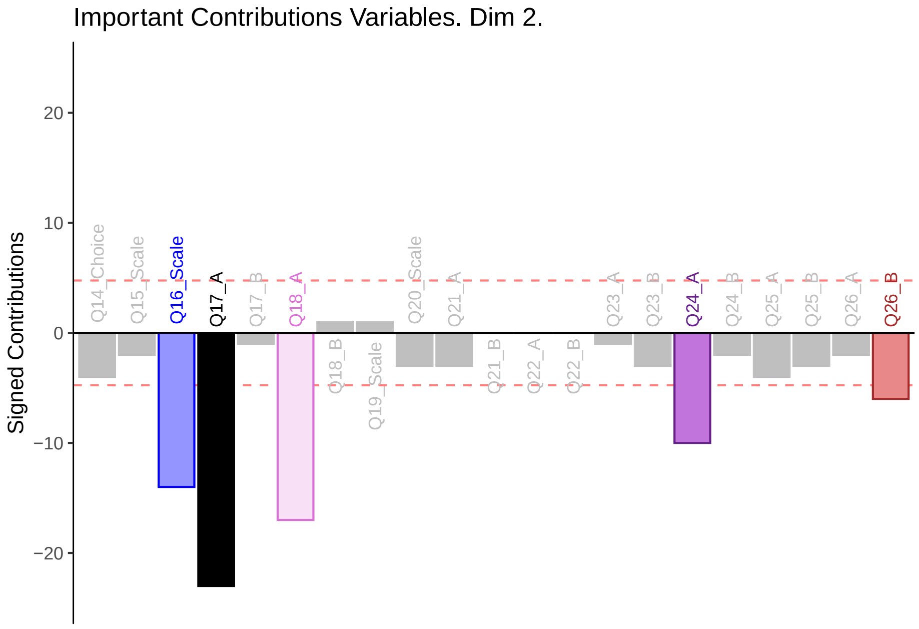
Component 1: It is quite evident that State-Trait Anxiety Inventory (STAI) Level 1,3 and 6 contribute the most to the 1st Component.

Component 2: It is quite evident that State-Trait Anxiety Inventory (STAI) Level 1,3 & 4 contribute the most to the 2nd Component.

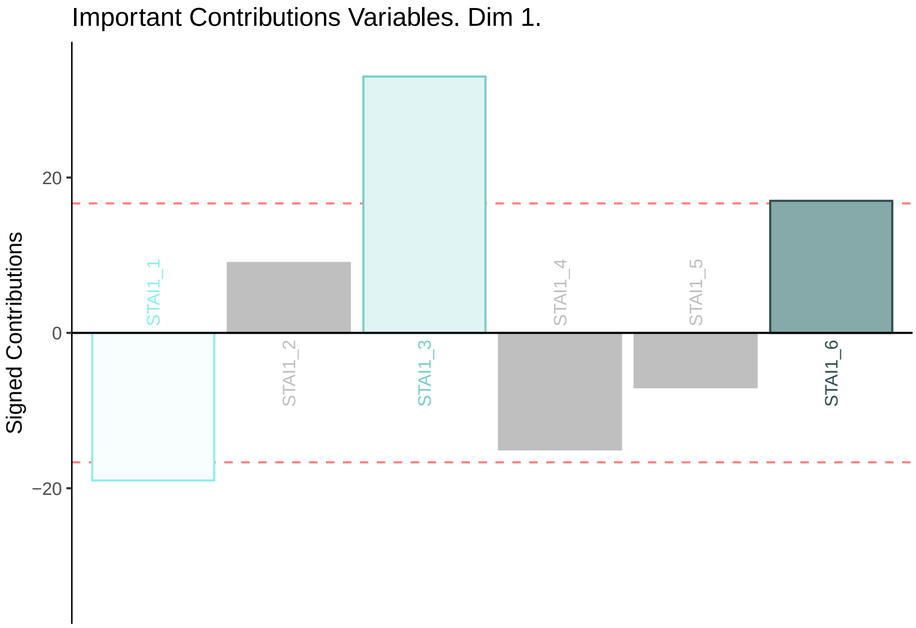
```
# Ctr J-set
###### 1 ====
col4Var <- c('yellow','red', 'blue', 'black',  'black','orchid','orchid','darkgreen','

ctri <- pls.res$TExPosition.Data$ci
signed.ctri <- ctri * sign(Fi)
# BR1
c001.plotCtri.1 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctri[,1]),
                        threshold = 100 / nrow(signed.ctri),
                        ylim = NULL,
                        color4bar = gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Important Contributions Variables. Dim 1.',
                        ylab = "Signed Contributions")

c001.plotCtri.1
```



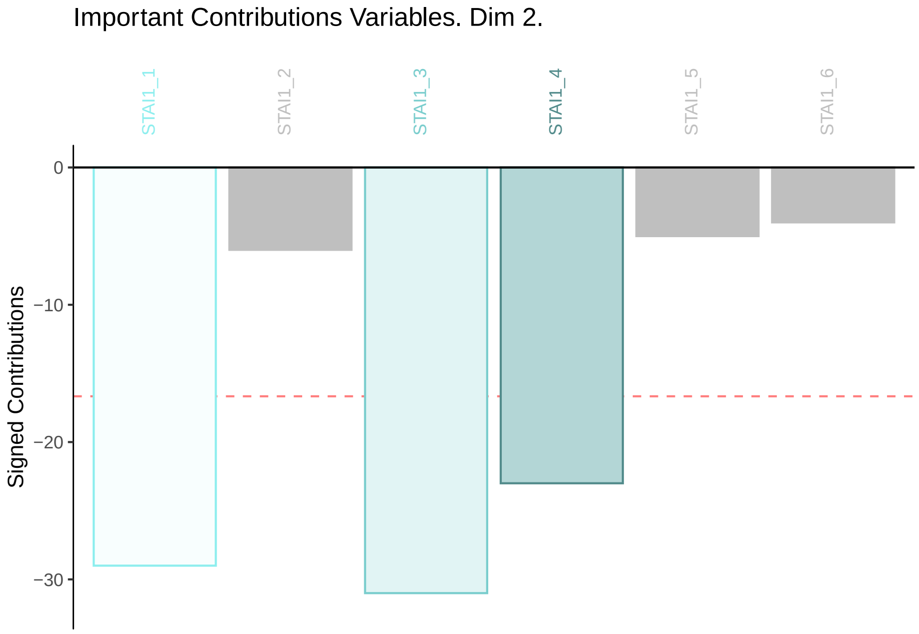
```
c001.plotCtri.2 <- PrettyBarPlot2(
                       bootratio = round(100*signed.ctri[,2]),
                       threshold = 100 / nrow(signed.ctri),
                       ylim = NULL,
                       color4bar = gplots::col2hex(col4Var),
                       color4ns = "gray75",
                       plotnames = TRUE,
                       main = 'Important Contributions Variables. Dim 2.',
                       ylab = "Signed Contributions")

c001.plotCtri.2
```



```
ctrj <- pls.res$TExPosition.Data$cj
signed.ctrj <- ctrj * sign(Fj)
# BR2
col4Var2 <- c('darkslategray2','darkslategray2','darkslategray3','darkslategray4','roya

c001.plotCtrj.1 <- PrettyBarPlot2(
                       bootratio = round(100*signed.ctrj[,1]),
                       threshold = 100 / nrow(signed.ctrj),
                       ylim = NULL,
                       color4bar = gplots::col2hex(col4Var2),
                       color4ns = "gray75",
                       plotnames = TRUE,
                       main = 'Important Contributions Variables. Dim 1.',
                       ylab = "Signed Contributions")

c001.plotCtrj.1
```



```
c001.plotCtrj.2 <- PrettyBarPlot2(
                       bootratio = round(100*signed.ctrj[,2]),
                       threshold = 100 / nrow(signed.ctrj),
                       ylim = NULL,
                       color4bar = gplots::col2hex(col4Var2),
                       color4ns = "gray75",
                       plotnames = TRUE,
                       main = 'Important Contributions Variables. Dim 2.',
                       ylab = "Signed Contributions")

c001.plotCtrj.2
```



6.18 Table 1:

Component 1: From the Bootstraps, we can observe that the Current Average Speed (Q17_A & Q17_B), Importance of Speed after increase (Q24_A & Q24_B) and Importance of total time saved (Q25_A & Q25_B) for both the roads prove to be significant.

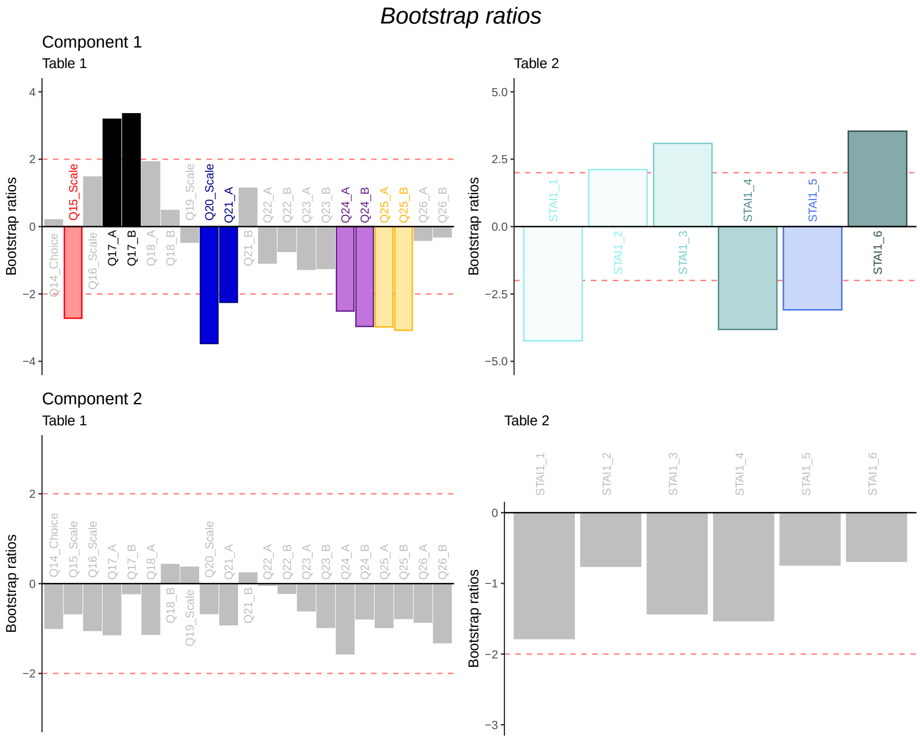
Component 2:

The Bootstraps do not exist, therefore insignificant.

6.19 Table 2:

Component 1: From the Bootstraps, we can observe that State-Trait Anxiety Inventory (STAI) Level 1,4 & 6 prove to be significant.

Component 2: The Bootstraps do not exist, therefore insignificant.



6.20 Summary

From the latent variables:

Component 1: Group A vs Group B : Lx1 vs Ly1

Component 2: Group A vs Group B : Lx2 vs Ly2

6.21 From the scores of Table 1:

Component 1: Q17_A & B (Current Average Speed) VS Q20_Scale (Satisfied with implementation and judgement)

Component 2: Q16, Q17A & Q18A: Road Distance, Current Avg. Speed and Speed after reconstruction

Interpretation: Participants of Group A & Group B had shown Level 3 of Anxiety for the question on current average speed & were not satisfied with their judgement.

6.22 From the scores of Table 2:

Component 1: State-Trait Anxiety Inventory (STAI) Level 1 & 6 VS Level 3

Component 2: State-Trait Anxiety Inventory (STAI) Level 1,3 & 4

Interpretation: Majority of Group A & Group B had shown extreme levels of anxiety (either 1 or 6) for most of the questions on Road Distance, Current Avg. Speed and Speed after reconstruction.

Therefore, as far as the psychometrics are concerned PLSC extracted most of the information about the participants and how these factors influenced the responses for the respective questions.

Chapter 7

Multiple Factor Analysis

7.1 Method: 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 these tables.

7.2 Data set: Drive.RData

Drive.RData is a native data set in R. It consists the driving style of many experienced drivers who's decisions are put to test. It measures the 191 individuals decisions(rows) on 21 quantitative variables (columns) which was measured by the Visual Analog Signal(VAS).

- Q_1: Implementation Decision
- Q2_Scale: Implementation Decision & judgement
- Q3_Scale: Road Distance

- Q4_A: Current Average Speed (A)
- Q4_B: Current Average Speed (B)
- Q5_B: Speed after reconstruction (B)
- Q6_Costs: Cost
- Q7_Satisfied: Satisfied with implementation and judgement
- Q8_A: Travel Time Saved (A)
- Q8_B: Travel Time Saved (B)
- Q9_A: Importance of Road Distance (A)
- Q9_B: Importance of Road Distance (B)
- Q10_A: Importance of speed before increase (A)
- Q10_B: Importance of speed before increase (B)
- Q11_A: Importance of speed after increase (A)
- Q11_B: Importance of speed after increase (B)
- Q12_A: Importance of time saving (A)
- Q12_B: Importance of time saving (B)
- Q13_A: Importance of Costs (A)
- Q13_B: Importance of Costs (B)

7.3 Load data

```r
load("Drive.RData")

data.pca2 <- data.pca[1:191,2:21]
data.design <- as.matrix(as.numeric(data.pca$Q1_Choice))
#head(data.pca2)
library(pander)
pander::pander(data.pca2[1:5,1:20])
```

Table 7.1: Table continues below

Q2_Scale	Q3_Scale	Q4_A	Q4_B	Q5_A	Q5_B	Q6_Costs	Q7_Satisfied
88.91	34.44	8.95	21.79	17.32	45.91	68.48	92.8
97.28	32.49	15.76	22.57	54.86	59.73	100	93.39
81.32	43.77	8.56	20.62	19.65	49.81	66.54	89.49
89.49	25.1	3.11	13.81	8.37	33.07	65.56	85.8
76.26	50.19	4.28	17.7	14.2	41.05	25.68	50.19

Table 7.2: Table continues below

Q8_A	Q8_B	Q9_A	Q9_B	Q10_A	Q10_B	Q11_A	Q11_B	Q12_A	Q12_B
19.46	25.1	16.15	15.95	100	24.51	100	25.49	100	36.77

Q8_A	Q8_B	Q9_A	Q9_B	Q10_A	Q10_B	Q11_A	Q11_B	Q12_A	Q12_B
13.23	42.22	44.75	44.75	19.07	19.65	75.1	86.58	35.6	55.84
81.91	62.06	87.55	87.35	100	100	100	100	100	100
66.54	22.37	0.78	0.78	92.02	100	96.89	100	0.58	0.97
22.76	50.58	26.07	74.71	34.24	39.69	47.28	51.56	30.35	62.45

Q13_A	Q13_B
48.64	48.83
3.89	5.45
0	0
1.56	0.78
35.02	57.98

7.4 Run MFA

```r
# run mfa

data.mfa <- cbind2(data.pca2,data.survey2)
data.mfa <- cbind2(data.mfa,data.survey3)

Grouping <- data.mfa[1,]
Grouping[1:20] <- 'E1'
Grouping[21:41] <- 'E2'
Grouping[42:47] <-  'E3'

run.mfa.data <- mpMFA(data.mfa, Grouping, DESIGN = data.design, graphs = FALSE)
```

```
## [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"
```

7.5 Rv

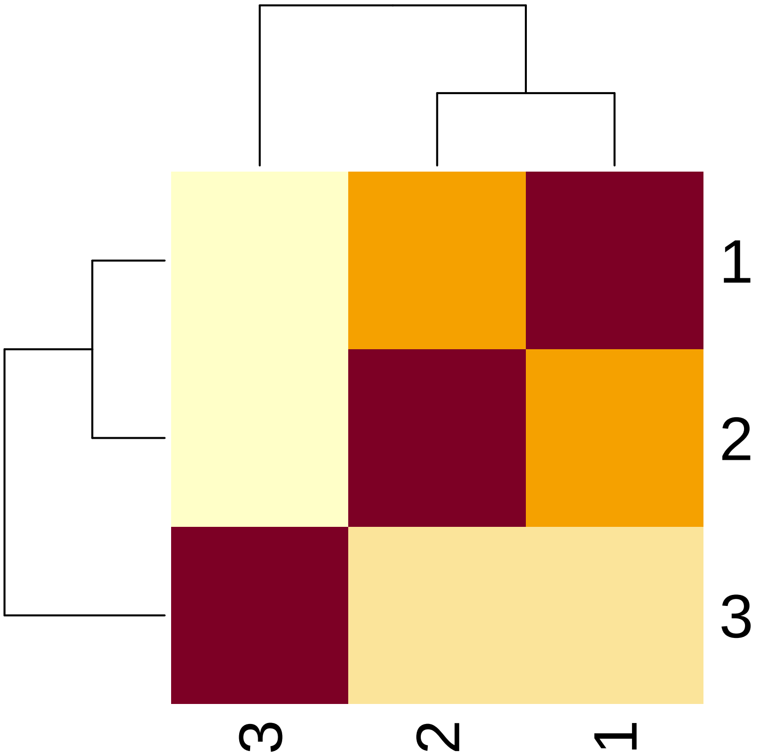
For the Heatmap: + 1 signifies Table 1 or data.pca

- 2 signifies Table 2 or datasurvey2

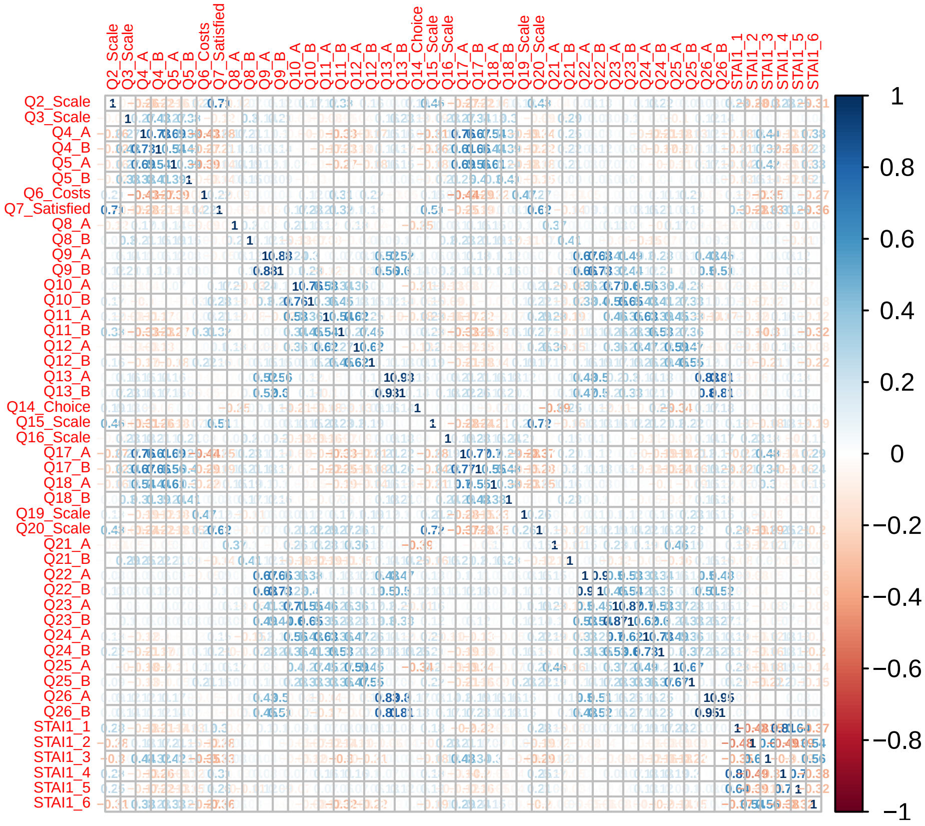
- 3 signifies Table 3 or datasurvey3

We can observe that the current average speed, spped after construction important parameters (Road distance, speed before and after increase and cost) are strongly correlated and consistent in both the tables.

```
heatmap(run.mfa.data$mexPosition.Data$InnerProduct$RVMatrix)
```



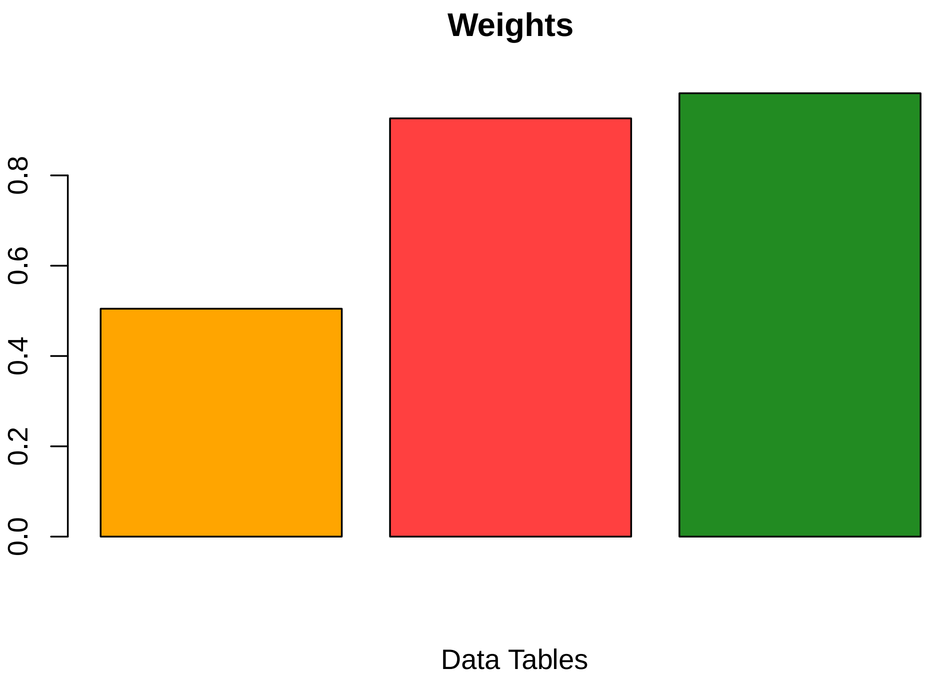
```
# Compute the covariance matrix
heatmap.cor <- cor(data.mfa,data.mfa)
# Plot it with corrplot
corrplot(t(heatmap.cor), method = "number",tl.cex = 0.5,number.cex = 0.4)
```



7.6 Weights (alpha in the paper) applied to each data table

The weights are applied to each data table. Data table 1 is indicated in the color orange, data table 2 in red and data table 3 in green

```
Eig.tab <- run.mfa.data$mexPosition.Data$Compromise$compromise.eigs
Alpha <- (1/sqrt(Eig.tab))

weight <- Alpha

plot.weights <-barplot(weight, main= "Weights",
        col = as.vector(c("orange","brown1","forestgreen") ),
        xlab= " Data Tables")
```

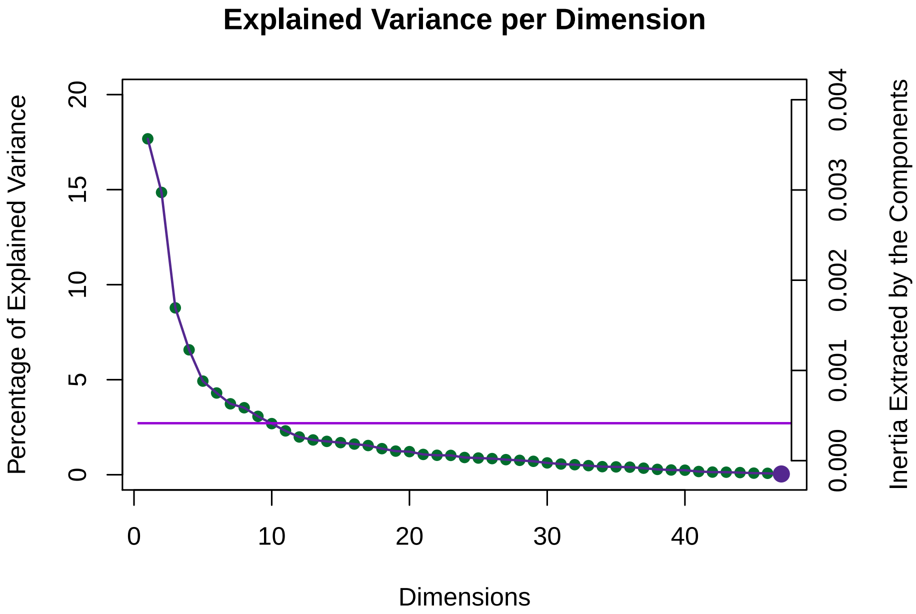


7.7 Eigenvalue and explained variance of MFA

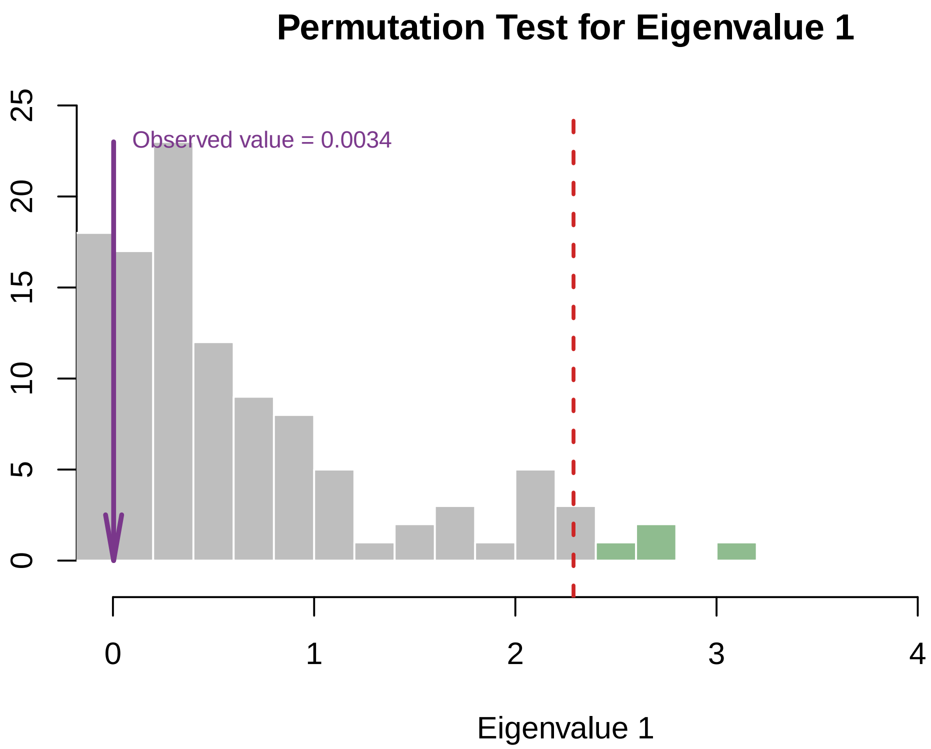
7.8 Scree Plot

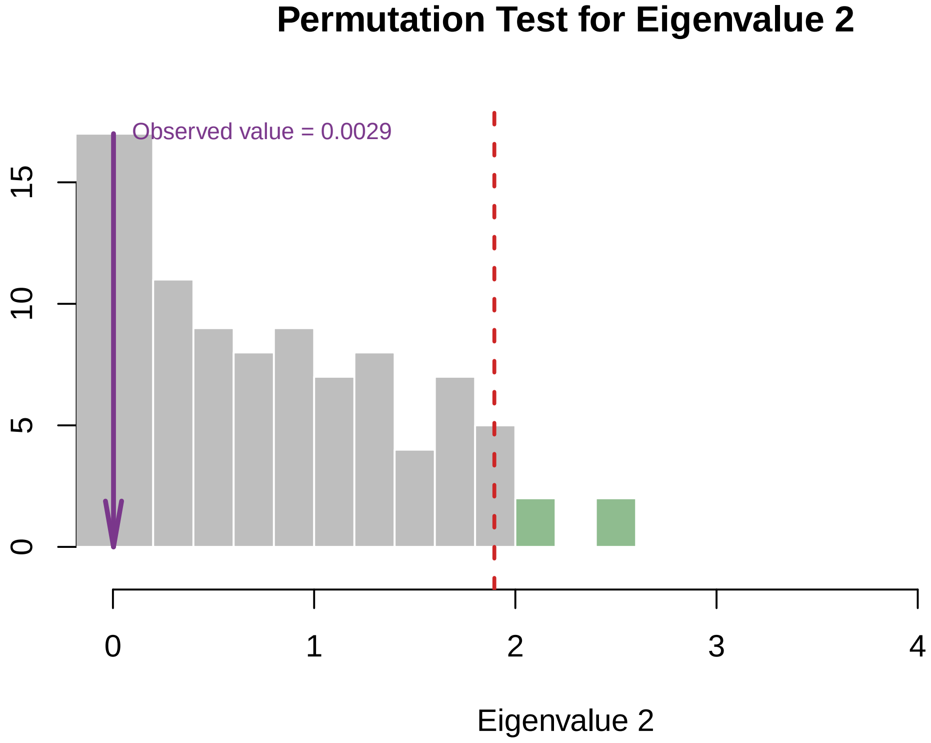
From the Scree Plot it is evident that 4 to 5 dimension need to be interpreted.

```
my.scree <- PlotScree(ev = Eig4scree,
                      run.mfa.data$mexPosition.Data$Compromise$compromise.t,plotKaiser =
```


Explained Variance per Dimension
Percentage of Explained Variance
0
5
10
15
20
Inertia Extracted by the Components
0.000
0.001
0.002
0.003
0.004
0
10
20
30
40
Dimensions

7.9 Permutations





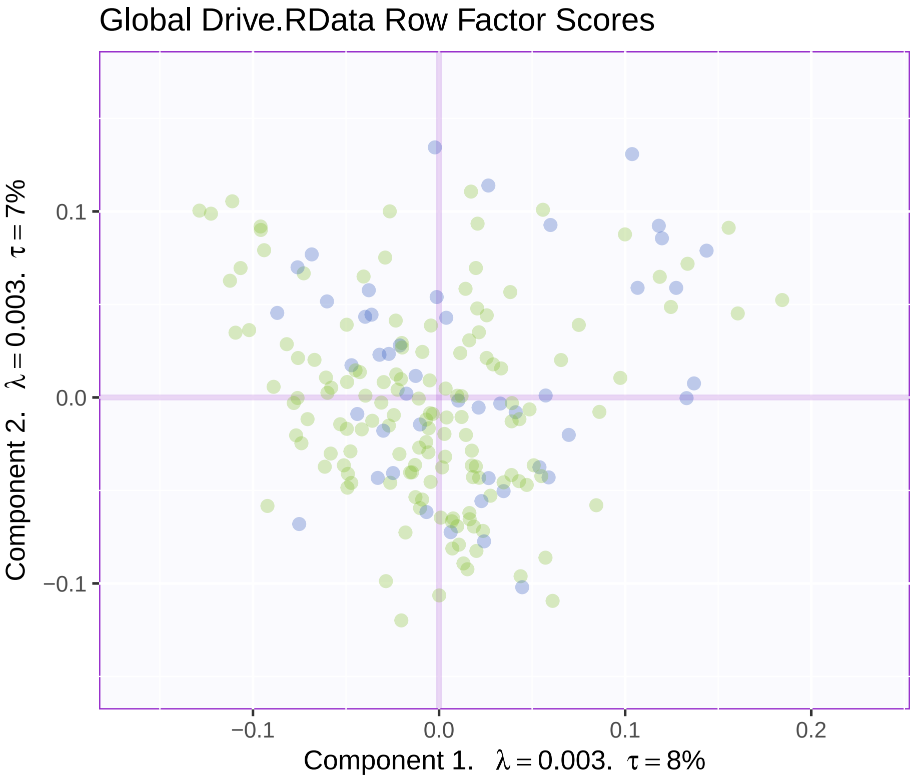
7.10 Global factor scores of the rows: how the rows are projected onto the space from the perspective of all tables

7.11 Plot Row Factor Scores

Just earlier row factor plots in previous methods, we observe a spread of both groups(Groups A & B)

```
my.fi.plot <- createFactorMap(run.mfa.data$mexPosition.Data$Table$fi, # data
                           title = "Global Drive.RData Row Factor Scores", # title of
                           axis1 = 1, axis2 = 2, # which component for x and y axes
                           pch = 19, # the shape of the dots (google `pch`)
                           cex = 2, # the size of the dots
                           text.cex = 2.5, # the size of the text
                           alpha.points = 0.3,
                           col.points = run.mfa.data$Plotting.Data$fi.col, # color of
                           col.labels = run.mfa.data$Plotting.Data$fi.col, # color for
                            display.labels = FALSE)
```

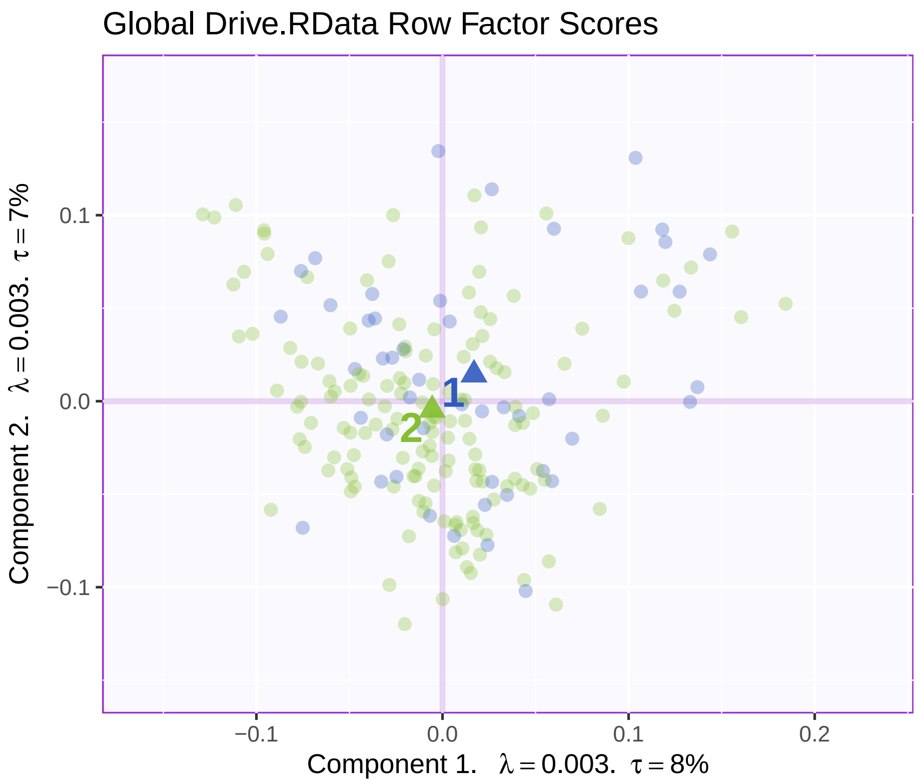
```
label4Map <- createxyLabels.gen(1,2,
                            lambda = run.mfa.data$mexPosition.Data$Table$eigs,
                            tau = round(run.mfa.data$mexPosition.Data$Table$t),
                            axisName = "Component "
                            )
fi.plot <- my.fi.plot$zeMap + label4Map # you need this line to be able to save them i
fi.plot
```



7.12 Means

It is evident that blue signifies Group A and green signifies Group B.

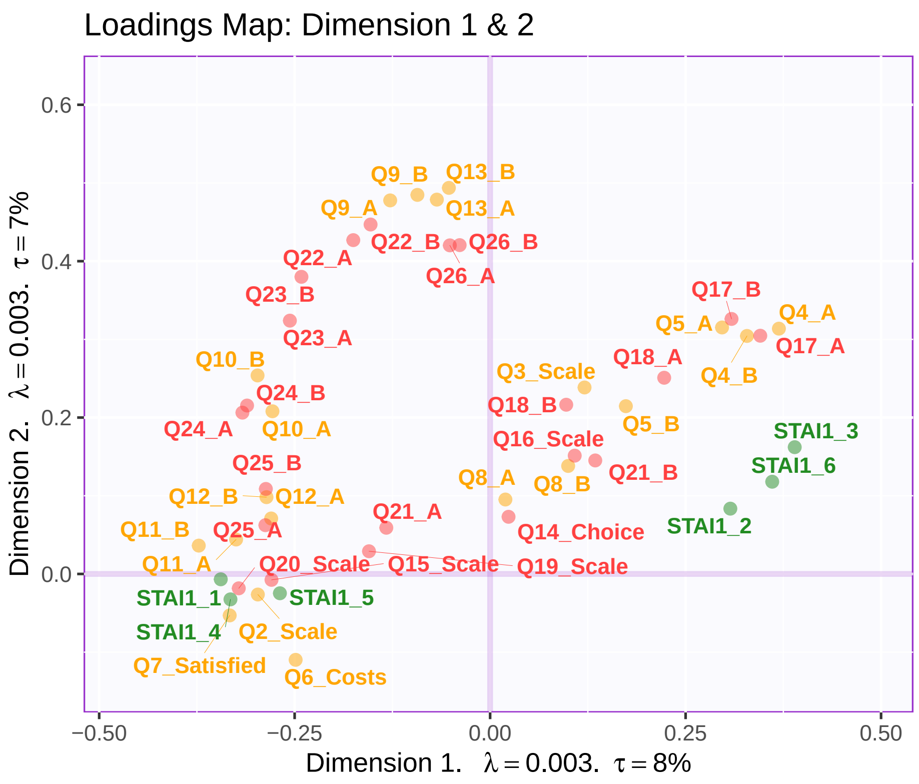
```
# get index for the first row of each group
grp.ind <- order(data.design)[!duplicated(sort(data.design))]
grp.col <- run.mfa.data$Plotting.Data$fi.col[grp.ind] # get the color
grp.name <- data.design[grp.ind] # get the corresponding groups
names(grp.col) <- grp.name
group.mean <- aggregate(run.mfa.data$mexPosition.Data$Table$fi,
                    by = list(data.design), # must be a list
                    mean)

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 = 0.9,
                               col.points = grp.col[rownames(fi.mean)],
                               col.labels = grp.col[rownames(fi.mean)],
                               pch = 17,
                               cex = 3.5,
                               text.cex = 6)
fi.WithMean <- my.fi.plot$zeMap_background + my.fi.plot$zeMap_dots + fi.mean.plot$zeMap
fi.WithMean
```



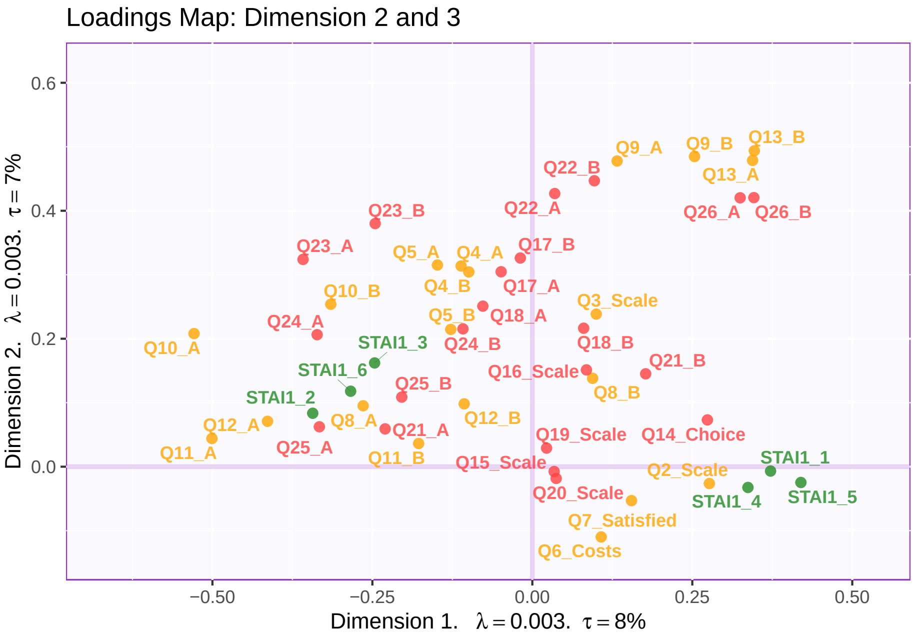
7.13 Column Factor scores

From Colummn factor scores it is evident that the plots are consistent for both Tables 1 & 2. Table 1 : Plots are indicated in the color orange. For Table 2: Plots are indicated in the color red. However, for Table 3: the anxiety levels are indicated in the color green. For Component 1: STAI (1,4,5) VS STAI (2,3,6) For Component 2: Importance of Cost (Q_13A & Q_13B,Q_26A & Q_26B) for both tables 1 & 2 VS Q6cost (estimated cost)

```
Q <- run.mfa.data$mexPosition.Data$Table$Q

n1<-20
c1<-rep("data.pca2", each=n1)
n2<-21
c2<-rep("data.survey2",each=n2)
n3<-6
c3<-rep("data.survey3", each=n3)

col.design <- c(c1,c2,c3)

col4Var <- col.design
col4Var <- recode(col4Var,"data.pca2"='orange',"data.survey2"='brown1',"data.survey3"='forestgree
label4Map.mfa <- createxyLabels.gen(1,2,
                                    lambda = run.mfa.data$mexPosition.Data$Table$eigs,
                                    tau = run.mfa.data$mexPosition.Data$Table$t)

baseMap.j <- createFactorMap(Q, #constraints = constraints.sym,
                             col.points =col4Var,
                             col.labels = col4Var,
                             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(col4Var),
                                        alpha.points =  .8,
                                        alpha.labels = .8,
                                        col.labels   =  gplots::col2hex(col4Var),
                                        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
```



7.14 Partial factor scores of the rows:

The partial factor scores tell us how the rows are viewed from the persepctive of each table. It depicts the similarity of the components for each of the groups

```
# Compute partial map



col4pfi2 <- col.design
col4pfi2 <- recode(col4pfi2,"data.pca2"='orange',"data.survey2"='brown1',"data.survey3"='forestgr


F_j<- run.mfa.data$mexPosition.Data$Table$partial.fi.array

run.mfa.data$mexPosition.Data$InnerProduct$alphaWeights
```

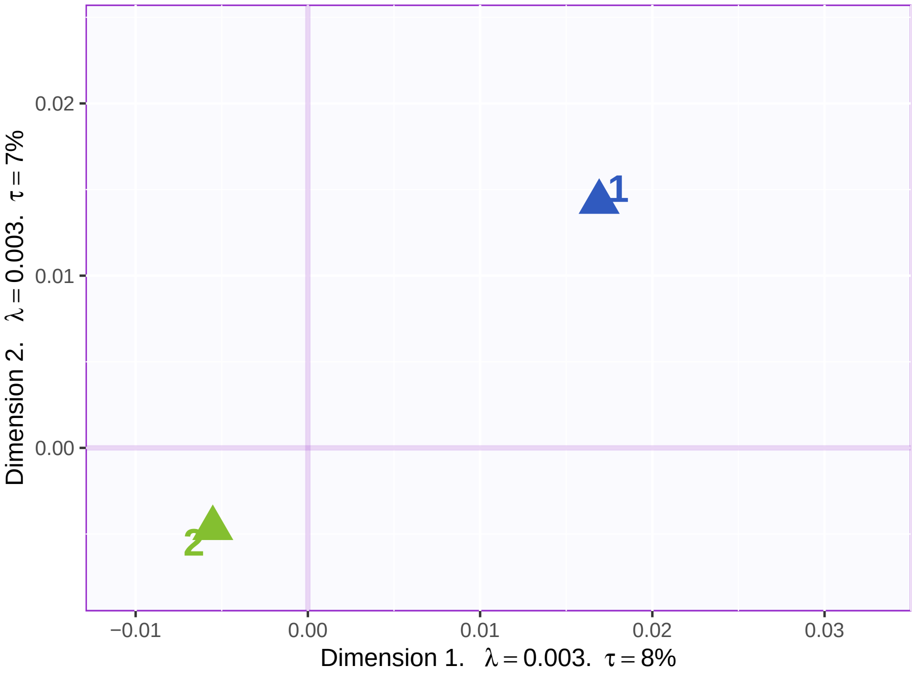
```
##             [,1]      [,2]      [,3]
## [1,] 0.3333333 0.3333333 0.3333333
```

```
Eig.tab <- run.mfa.data$mexPosition.Data$Compromise$compromise.eigs
alpha_j <- 1/sqrt(Eig.tab)
```

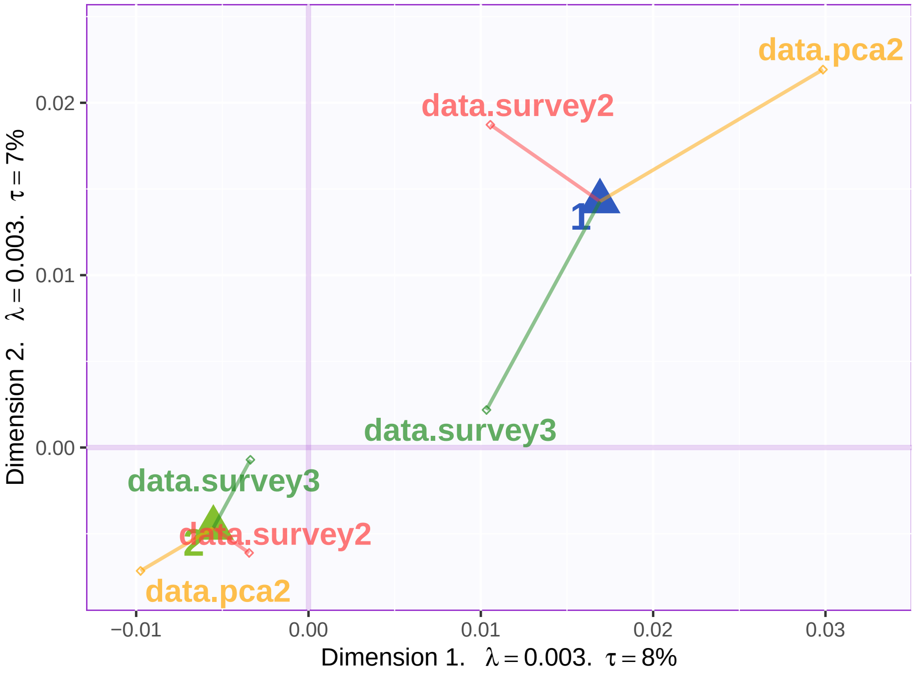
```
data_tables<- col.design
code4Groups<- unique(data_tables)
nK<- length(code4Groups)


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(data.design),
#                     # must be a list
#                      mean
#                      ))

meanfk <-
  apply(F_k, c(2,3), FUN = function(x){
  aggregate(x, by = list(data.design), mean)$x
  })
dim(meanfk)
```

```
## [1]  2 47  3
```

```
 mean.plot <- createFactorMap(fi.mean,
                                constraints = minmaxHelper4Partial(fi.mean, meanfk, ax
                              alpha.points = 1,
                                display.labels = TRUE,
                                col.points = grp.col[rownames(fi.mean)],
                                col.labels = grp.col[rownames(fi.mean)],
                                pch = 17,
                                cex = 6,
                                text.cex = 6
                                )

Fi.meanonly.plot<- mean.plot$zeMap_background+mean.plot$zeMap_dots + mean.plot$zeMap_text+ label4

Fi.meanonly.plot
```



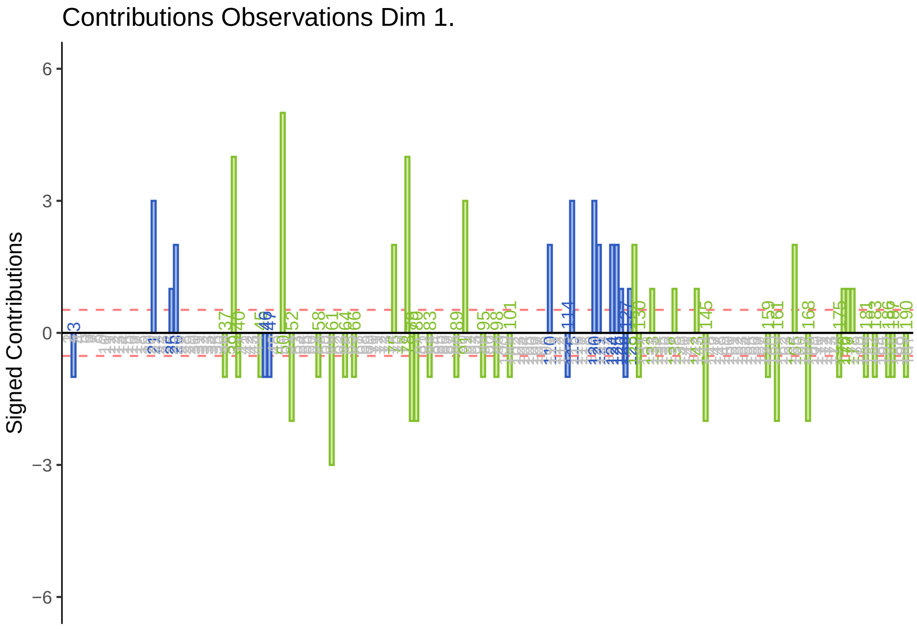
```
pf.means <- createPartialFactorScoresMap(
                                        factorScores = fi.mean,
                                        partialFactorScores = meanfk,
                                        axis1 = 1, axis2 = 2,
                                        colors4Items = as.vector(col4pfi2), #as.vector(grp.col[r
colors4Blocks = c("orange","brown1","forestgreen"),
                                        names4Partial = dimnames(meanfk)[[3]], #
                                        font.labels = 'bold',
                                        size.labels = 4.8,
)

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



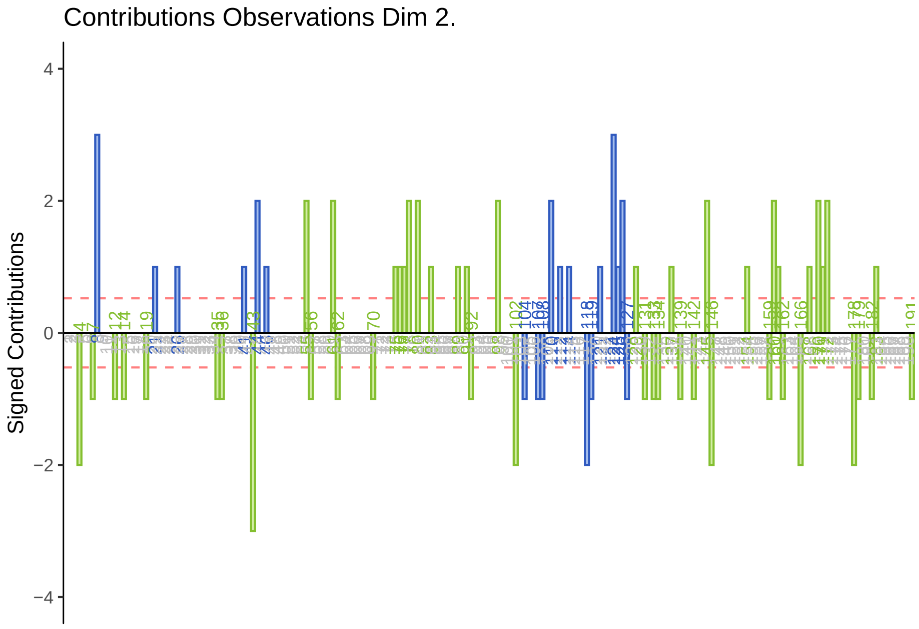
7.15 Contribution Plots

The contribution plots for both the groups A & B are shown in both components.

```
Fi <- run.mfa.data$mexPosition.Data$Table$fi
# col4Var <- 4*c('red', 'blue', 'black',  'black','orchid','orchid','darkgreen','blue4

ctri <- run.mfa.data$mexPosition.Data$Table$ci
signed.ctri <- ctri * sign(Fi)
# BR1
c001.plotCtri.1 <- PrettyBarPlot2(
                          bootratio = round(100*signed.ctri[,1]),
                          threshold = 100 / nrow(signed.ctri),
                          ylim = NULL,
                          color4bar = run.mfa.data$Plotting.Data$fi.col, #gplots::col2hex
                          color4ns = "gray75",
                          plotnames = TRUE,
                          main = 'Contributions Observations Dim 1.',
                          ylab = "Signed Contributions")

c001.plotCtri.1
```

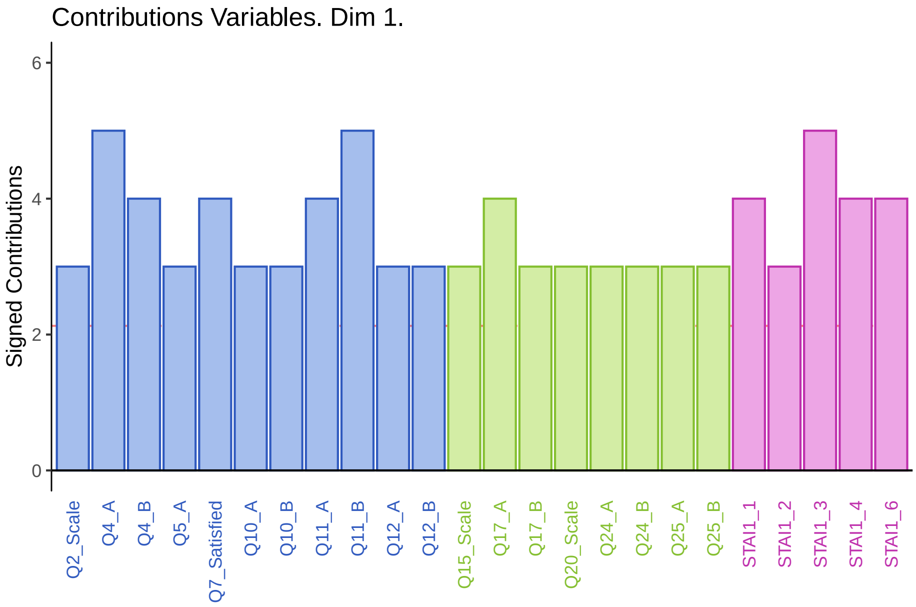


```
c001.plotCtri.2 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctri[,2]),
                        threshold = 100 / nrow(signed.ctri),
                        ylim = NULL,
                        color4bar =run.mfa.data$Plotting.Data$fi.col,#gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Contributions Observations Dim 2.',
                        ylab = "Signed Contributions")

c001.plotCtri.2
```


Contributions Observations Dim 2.

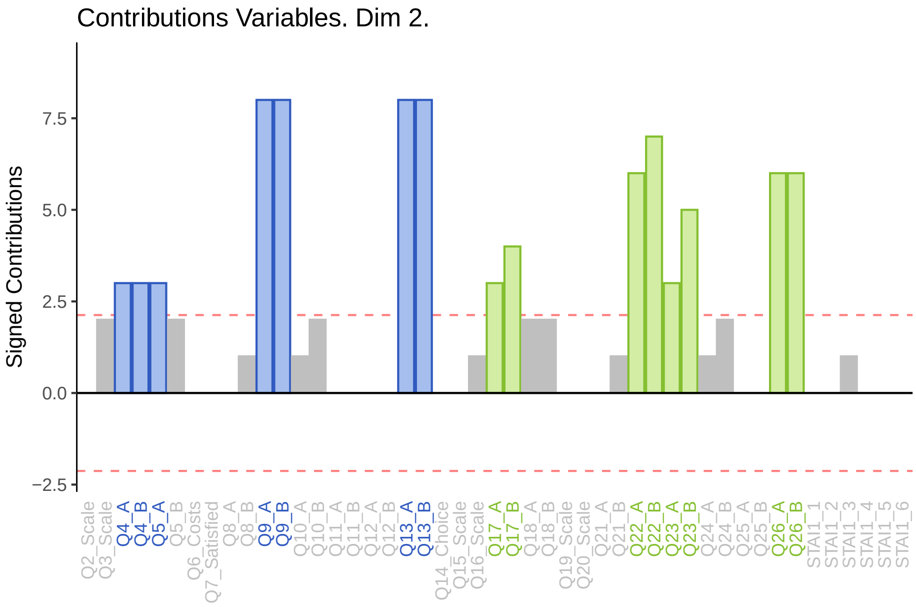
From the contribution plots, we can say that the variables that contributed the most throughout the study have been consistent across both the tables for component 1. Majority of the group members had shown extreme levels of anxiety for questions on current average speed (Q__4A,Q__4B,Q__17A & Q__17B) and speed before and after increase(Q__10A,Q__10B,Q__11A & Q__11B).

```
Fj <- run.mfa.data$mexPosition.Data$Table$cj
# col4Var <- c('red', 'blue', 'black',  'black','orchid','orchid','darkgreen','blue4',
#
ctrj <- run.mfa.data$mexPosition.Data$Table$cj
signed.ctrj <- ctrj * sign(Fj)
# BR1
c001.plotCtrj.1 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctrj[,1]),
                        threshold = 100 / nrow(signed.ctrj),
                        ylim = NULL,
                        color4bar = run.mfa.data$Plotting.Data$fj.col,# gplots::col2hex
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Contributions Variables. Dim 1.',
                        ylab = "Signed Contributions",
                        signifOnly = TRUE)

c001.plotCtrj.1
```



However for Component 2: + The current average speed for the tables (Q__4A,Q__4B,Q__17A & Q__17B), + Importance of road distance (Q__9A,Q__9B,Q__17A & Q__17B), + Importance of cost (Q__13A,Q__13B,Q__26A & Q__26B) are consistent across both the tables.

```
c001.plotCtrj.2
```



7.16 Summary

It is evident that just like PCA, MFA has provided with similar results that Importance of road distance and Cost for reconstructing the road were consistent in both data tables.

For Component 1:

Majority of the group members had shown extreme levels of anxiety for questions oncurrent average speed (Q__4A,Q__4B,Q__17A & Q__17B) and speed before and after increase(Q__10A,Q__10B,Q__11A & Q__11B).

For Component 2:

- The current average speed for the tables (Q__4A,Q__4B,Q__17A & Q__17B),
- Importance of road distance (Q__9A,Q__9B,Q__17A & Q__17B),
- Importance of cost (Q__13A,Q__13B,Q__26A & Q__26B) are consistent across both the tables.

Chapter 8

CA

8.1 Correspondence analysis

Correspondence analysis (CA) is a generalized principal component analysis tailored for the analysis of qualitative data. Originally, ca was created to analyze contingency tables, but, ca is so versatile that it is used with a lot of other data table types. The goal of correspondence analysis is to transform a data table into two sets of factor scores: One for the rows and one for the columns. The factor scores give the best represen- tation of the similarity structure of the rows and the columns of the table. In addition, the factors scores can be plotted as maps, which display the essential information of the original table. In these maps, rows and columns are displayed as points whose coordinates are the factor scores and where the dimensions are called factors. Interestingly, the factor scores of the rows and the columns have the same variance and, therefore, both rows and columns can be conveniently represented in one single map.

8.2 Dataset

The Dataset used for CA is the superhero power dataset. It consists of 41 rows and 4 columns. The 41 rows refer to the different type of superhero powers and the 4 columns refer to the good and bad type of superheros (male and female)

The supplementary data consists information of the neutral female and male superheros.

```r
#Get the data ----
load("superhero and power.rda")
```

```
superhero.power.new <- superhero.power*100
supplementary.hero.new <- supplementary.hero*100

X <- (as.matrix(superhero.power.new))

superhero.sup.obs <- superhero.power.new[1:41,]
superhero.sup.var <- supplementary.hero.new[,1:2]

head(superhero.power.new)
```

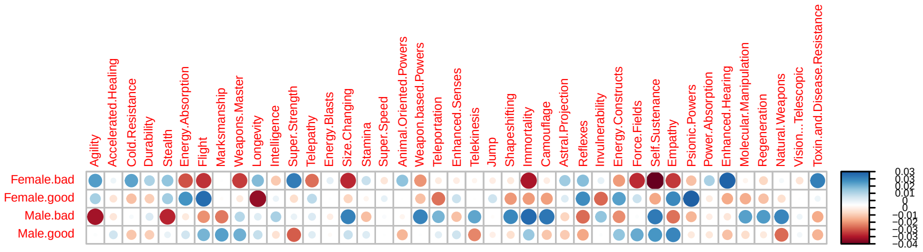
```
##                        Female.bad Female.good Male.bad Male.good
## Agility                 42.857143   35.971223 30.51948 39.510490
## Accelerated.Healing     25.714286   20.863309 27.27273 29.370629
## Cold.Resistance         11.428571    5.755396  9.74026  7.342657
## Durability              42.857143   31.654676 46.75325 39.160839
## Stealth                 22.857143   20.143885 14.93506 22.027972
## Energy.Absorption        5.714286   12.230216 10.38961 11.888112
```

8.3 The data pattern

But remember, chi-square is in counts, but CA analyzed probabilities (i.e., the profiles). So, we need to divide the chi-square statistics by the total sum of the data. Also, the chi-square statistic adds the chi-squares in all cells and give one number. In CA, however, we keep the pattern of chi-squares instead of adding all of them up.

Plot this residual:

```
corrplot(t(Inertia.cells), is.cor = FALSE,win.asp=1.0,tl.cex=0.5,cl.cex = 0.4)
```



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

The Correlation plot clearly tells us that: 1) Male villains had great power of Immortality, Self- Sustenance and Natural weapons and are highly correlated. 2)

Female superheros had the powers flight, psionic powers, empathy and reflexes are highly correlated and female villains had strong negative correlations with force fields, self-sustenance and empathy. 3) For both male super heros and villains show great correlation as far as empathy and force fields are concerned.

8.4 Analysis:

Since, each variable is measured on different units, the columns were scaled and centered. The rows are color-coded by the DESIGN variable, data.choice.

- `center = TRUE`: substracts the mean from each column
- `scale = TRUE`: after centering (or not), scales each column to have a sum of squares of 1 (see the help for different scaling options)
- `DESIGN`: colors the observations (rows)
- `graphs = FALSE`: this gives you plots from `epPCA`, but make sure to flag it `FALSE` for Rmarkdown to run correctly

There are two different ways to present the CA analysis: symmetric and asymmetric. You will show both, but please summarize these analyses and tell me which one do you think is a better way to illustrate the results in the end.

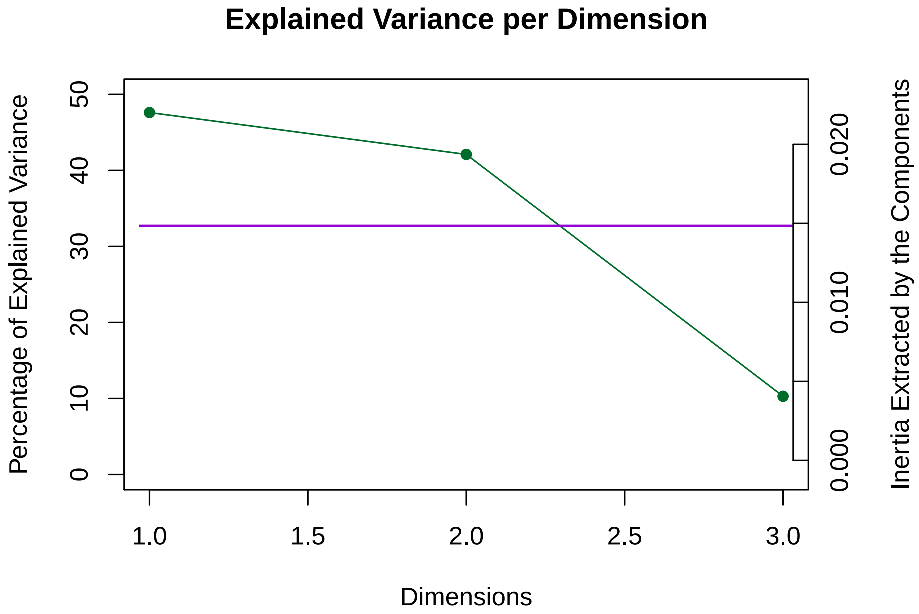
8.5 Asymmetric Plots

CA treats rows and columns symmetrically, and so their roles are equivalent. In some cases, how- ever, rows and columns can play different roles, and this symmetry can be misleading. In this case the roles are asymmetric and the plots can reflect this asymmetry by normalizing one set such that the variance of its factor scores is equal to 1 for each factor.

8.6 Scree Plot

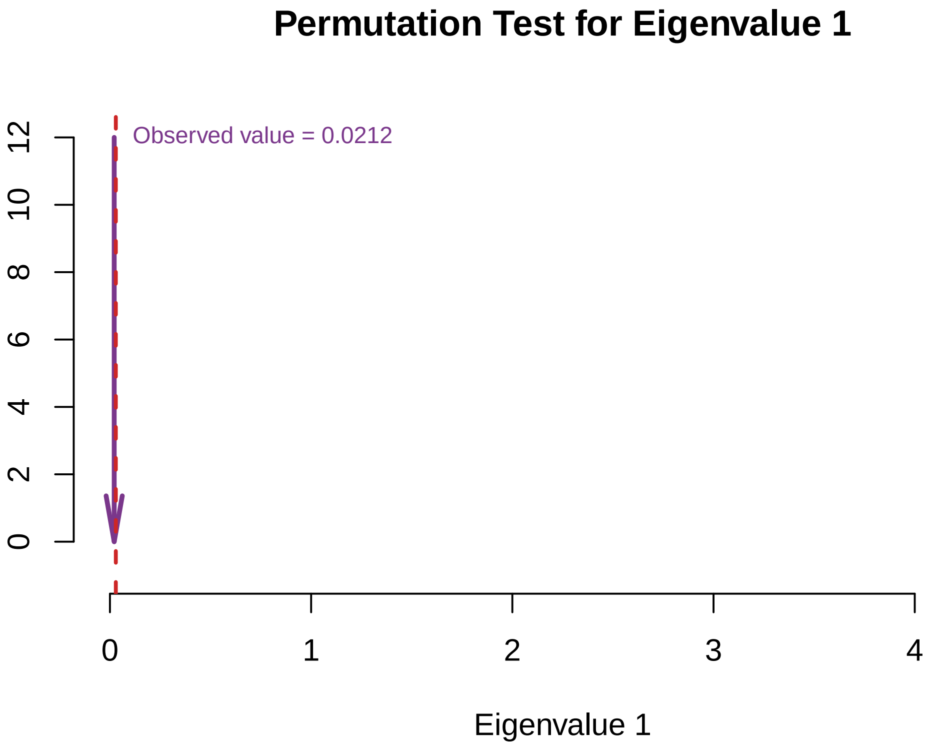
Shows the results from permutation with Scree plot by adding the estimated p-values to the PlotScree function. It is evident from the scree plot that 2 dimensions are significant and need to be interpreted.

```
my.scree <- PlotScree(ev = resCA.sym$ExPosition.Data$eigs,
                   p.ev = resCAinf.sym4bootJ$Inference.Data$components$p.vals,plotKaiser = TRUE,c
```

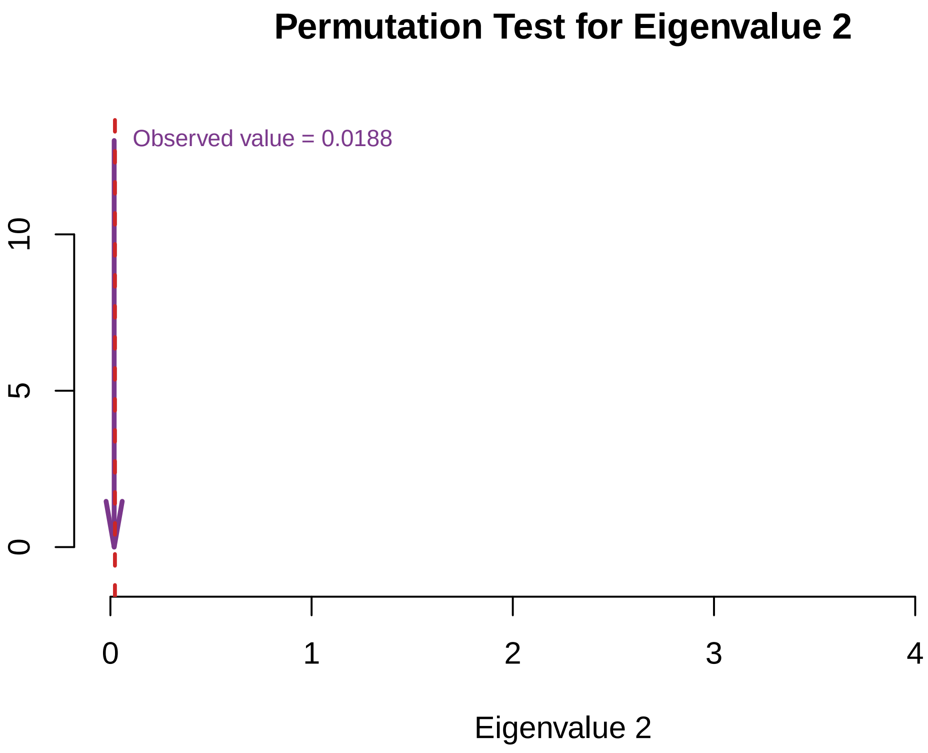


## Permutations

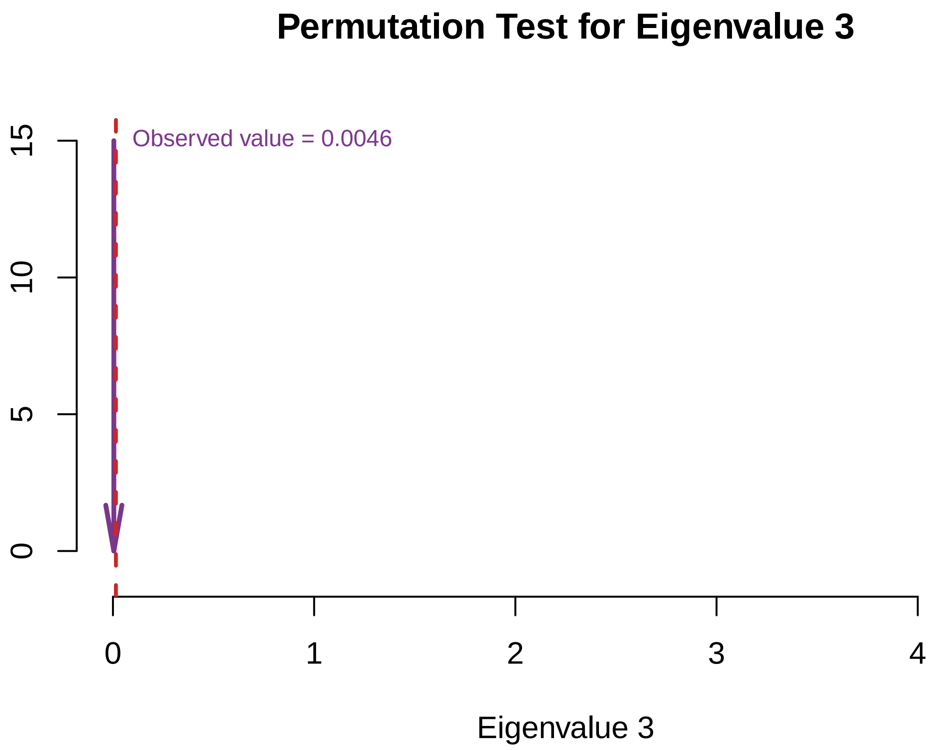
```
zeDim = 1
pH1 <- prettyHist(
  distribution = resCAinf.sym4bootJ$Inference.Data$components$eigs.perm[,zeDim],
           observed = resCAinf.sym4bootJ$Fixed.Data$ExPosition.Data$eigs[zeDim],
           xlim = c(0, 4.5), # needs to be set by hand
           breaks = 20,
           border = "white",
           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 = resCAinf.sym4bootJ$Inference.Data$components$eigs.perm[,zeDim],
           observed = resCAinf.sym4bootJ$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()

zeDim = 3
pH3 <- pH2 <- pH1 <- prettyHist(
  distribution = resCAinf.sym4bootJ$Inference.Data$components$eigs.perm[,zeDim],
           observed = resCAinf.sym4bootJ$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))
```



```r
eigs3 <- recordPlot()
```

8.7 Plot the asymmetric factor scores

First show an asymmetric plot without the labels for the rows/columns you project inside the simplex.

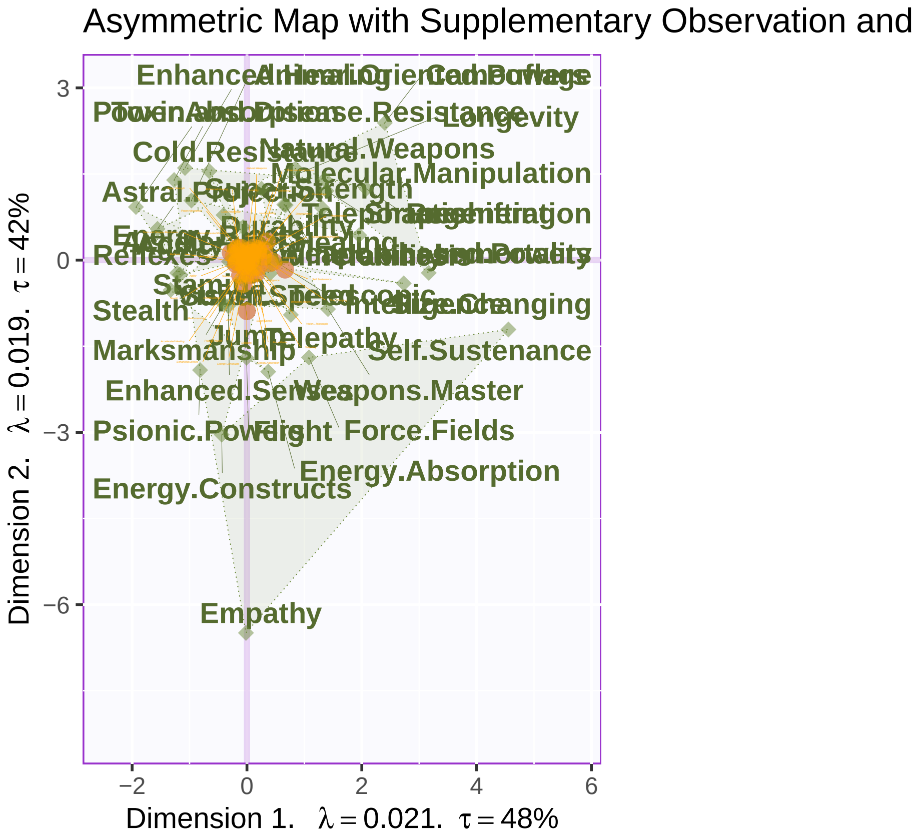
You need to interpret two things here:

1) the despersion of the data points in the simplex
2) the eigenvalue

The yellow points indicate the observations(rows) of the dataset which are nothing but the superhero powers are clustered around the center. The vertices of the simplex Indicate the gender along with type of the superhero/villain.

```r
# Your asymmetric factor scores
asymMap  <- createFactorMapIJ(Fi,Fj.a)
# With supplementary elements
```

```
mapSup <- createFactorMapIJ(as.data.frame(obs.sup$fii),
                            as.data.frame(var.sup$fjj)  ,
                            col.points.i = "Orange",
                            col.labels.i = 'Orange' ,
                            font.face.i = 'italic',
                            alpha.labels.i = 0.8,
                            alpha.points.i = 0.8,
                            col.points.j = 'red',
                            col.labels.j = 'red',
                            alpha.labels.j = 0.8,
                            font.face.j = 'italic',
                            alpha.points.j = 4,
                            constraints = constraints.sup,
                            text.cex.i=0.5, text.cex.j = 0.5
)
# Make the simplex visible
zePoly.J <-  PTCA4CATA::ggdrawPolygon(Fj.a,
                                      color = 'darkolivegreen4',
                                      size = .2,
                                      fill =  'darkolivegreen4',
                                      alpha = .1)
# Labels
labels4CA <- createxyLabels(resCA = resCA.asym)

# Combine all elements you want to include in this plot
map.IJ.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.IJ.sup.asym
```

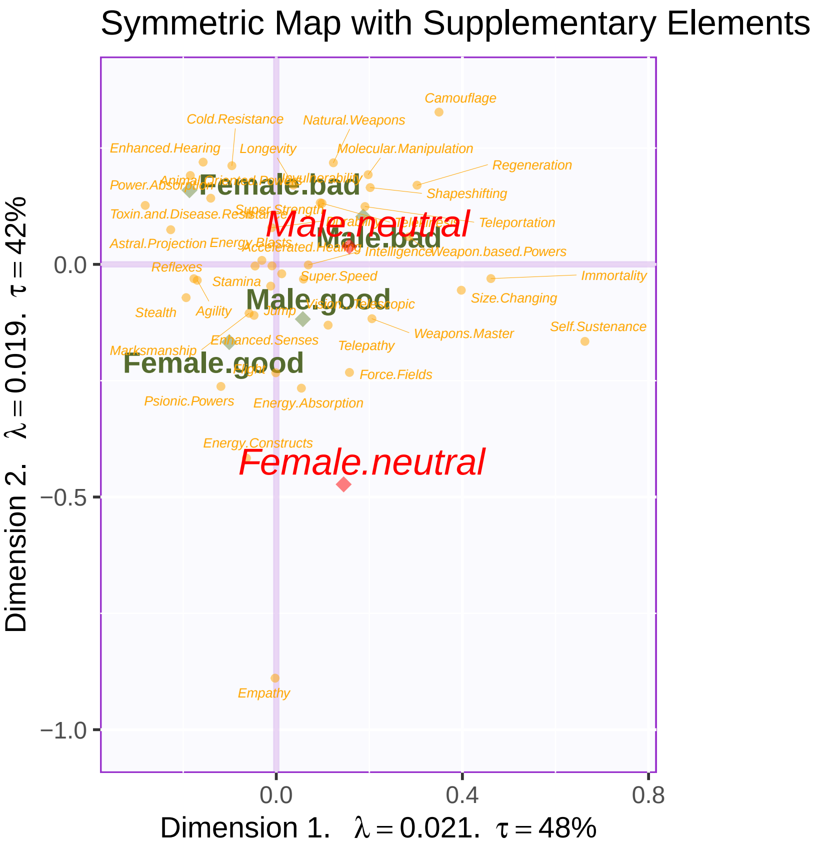


8.8 Plot the symmetric plot

Next, show the symmetric plot with all labels printed. Since there are larger number of rows and columns, we will plot in two different plots.

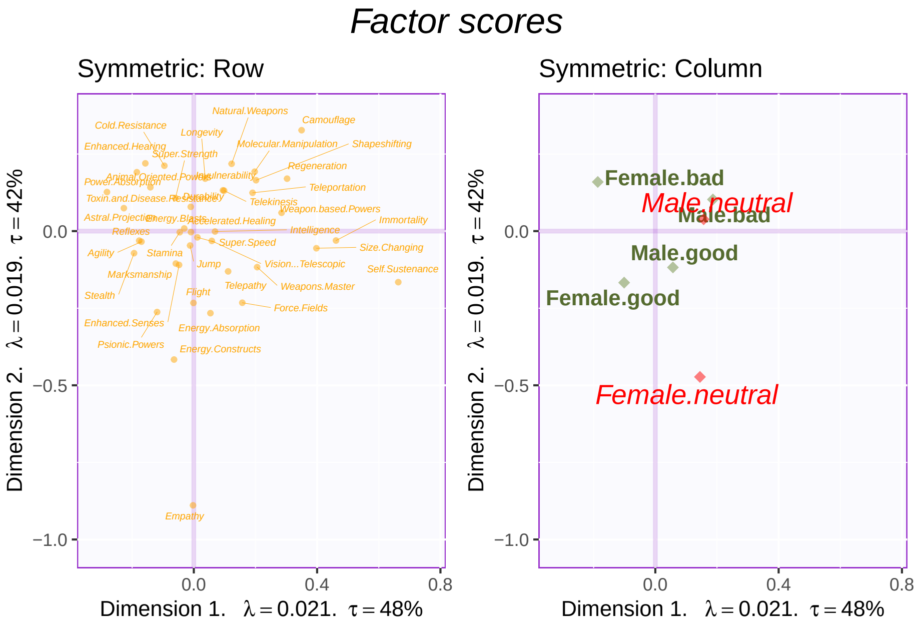
8.9 This is a biplot:

From the symmetric plot we can say that superheroes that are neutral are along the second component. Most importantly, the good superheroes are on one side together and the villains are on the other.



8.10 Row factor and Column Factor Scores with confidence intervals

```
grid.arrange(
    map.sepI.sup.sym, map.sepJ.sup.sym,
    ncol = 2,nrow = 1,
    top = textGrob("Factor scores", gp = gpar(fontsize = 18, font = 3))
  )
```

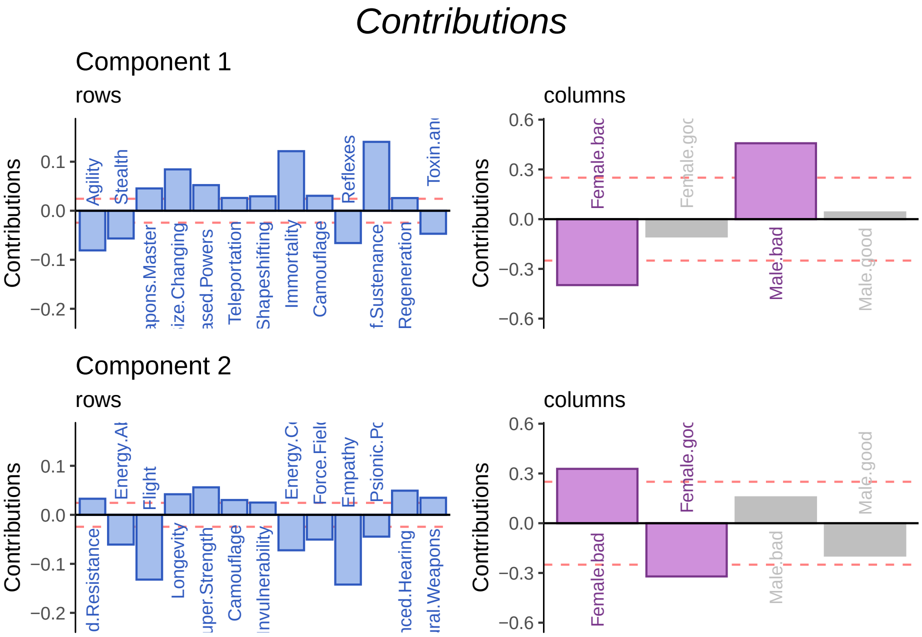


8.11 Contributions and bootstrap ratios barplots

8.12 Contribution barplots

For CA, we plot the contributions for both rows and 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))
  )
```



Component1: We can say that Agility, Stealth, Weapons.master, Size changing, Teleportation, shape.shifting, Immortality, Reflexes, self-sustenance, regeneration, camouflage and toxin.and.disease.resistance were the powers that were widely observed in villains (both male and females).

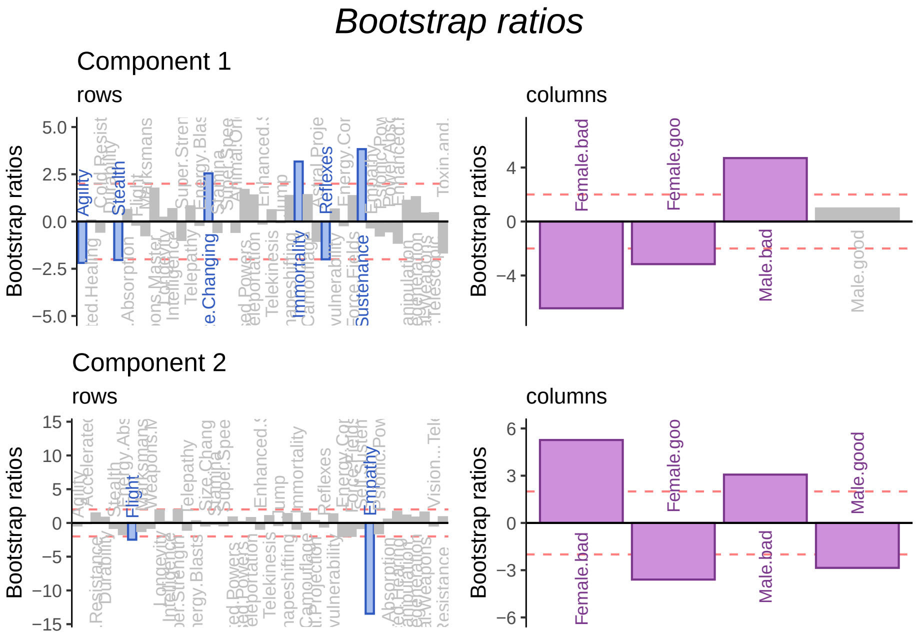
Component 2: We can say that cold.resistance, energy.absorption, flight, super.strength, longevity, camouflage, Invulnerability, force.field, empathy, psioninc.powers, enhanced hearing and natural.weapons were the powers widely observed in female superheros and villains.

8.13 Bootstrap ratios

The Bootstrap ratios signify the stability of the variables or column factor scores. Therefore, it illustrates the significance of the variables to the component or in other words the dimensions.

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))
  )
```

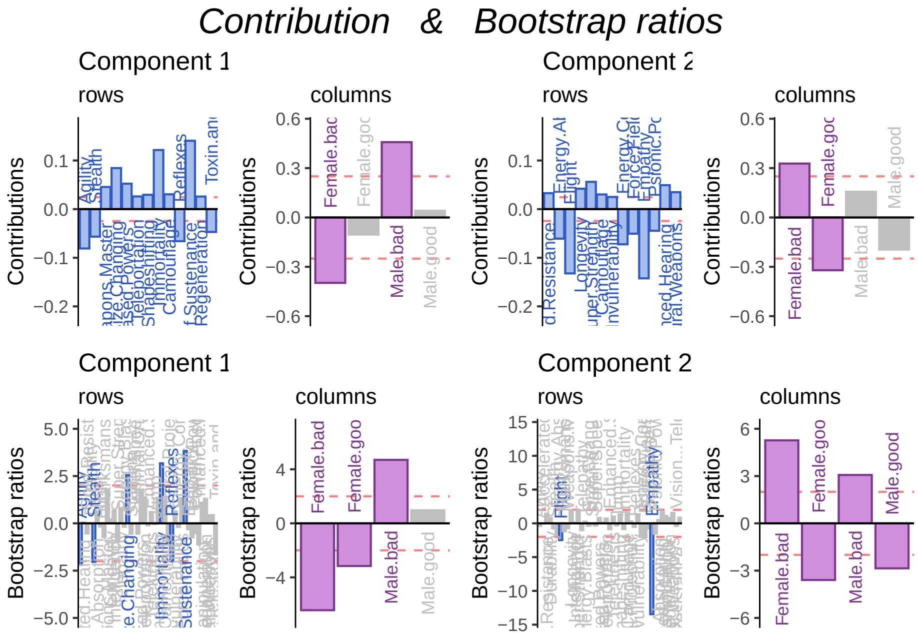


The Bootstrap ratios clearly indicate that Agility, Stealth, Size.changing, Immortality and Self Sustenance are highly significant and widely observed in Bad Female and Male superheros for Component 1.

The Bootstrap ratios clearly indicate that Flight, Energy.Constructs and Empathy are highly significant were widely observed in all the superheros for Component 2.

You can also arrange these plots to put contribution and bootstrap ratio plots side by side.

```
grid.arrange(
     as.grob(ctrI.1),as.grob(ctrJ.1),as.grob(ctrI.2),as.grob(ctrJ.2),as.grob(ba001.BR1.I),as.grob
    ncol = 4,nrow = 2,
    top = textGrob("Contribution   &   Bootstrap ratios", gp = gpar(fontsize = 18, font = 3))
  )
```



8.14 Summary

When we interpret the factor scores and loadings together, the CA revealed:

I prefer symmetric plot for the CA as the supplementary variables are visible with respoect to to the active observations. It is easier to interpret the superheros that were not too extreme in terms of their super powers.

Component 1: We can clearly say bad female and male superheros had Agility, Stealth, Self Sustenence and Immortality

Component 2: Both bad and good female superheros empathized and had great flight and energy absorbing abilities

Chapter 9

DiSTATIS

9.1 Method : DiSTATIS

DISTATIS is a generalization of classical multidimensional scaling (MDS) whose goal is to analyze a single distance matrix. By contrast, the goal of DISTATIS is to analyze a set of distance matrices. In order to compare distance matrices, DISTATIS combines them into a common structure called a compromise and then projects the original distance matrices onto this compromise. Each data set to be analyzed is called a study and corresponds to a distance matrix obtained on a common set of objects. For example, these distance matrices may correspond to measurements taken at different times. In this case, the first data set corresponds to the data collected at time t = 1, the second one to the data collected at time t = 2 and so on. The goal of the analysis, then, is to evaluate if the relative positions of the objects are stable over time. The different measurements, however, do not need to represent time. For example, the data sets can be distance matrices obtained by different methods or algorithms. The goal of the analysis, then, is to evaluate if there is an agreement between the methods.

9.2 The Dataset

The stimuli are white wines made from the grape Chenin Blanc. The wines whose names start with F are French, the wines whose names start with S are from South African.

The four files: Each corresponds to a condition in a 2 (judges from France vs South Africa) x 2 (With vs Without Information; i.e., without or without the labels) design.

For the name of the files: - fr means France and sa means South Africa: The wines that began with the letter F were French wines and the wines that began with the letter S were South African wines.

Where the judges are from (and also where the data were collected). The data from French judges were collected in France and the data from South African judges were collected in South Africa - info means with Information and no info means without information:

In the “no info” condition; the judges were not told anything except that the wines were made with Chenin Blanc.

In the “info” condition the judges were provided with the labels of the wines.

Note: The vocabulary is somewhat tricky because we have French and English vocabulary that will need to projected. The two English tables can be concatenated to make only one and same for the French tables.

	J3	J4	J5	J6	J7	J8	J9	J10	J11	J13
FCAR	4	5	1	1	2	2	2	3	2	1
SRUD	4	1	1	2	1	8	2	1	4	1
FBAU	2	2	3	4	1	7	1	1	2	2
FROC	3	1	1	3	2	4	2	2	4	3
SFED	1	5	5	1	1	7	2	3	3	2

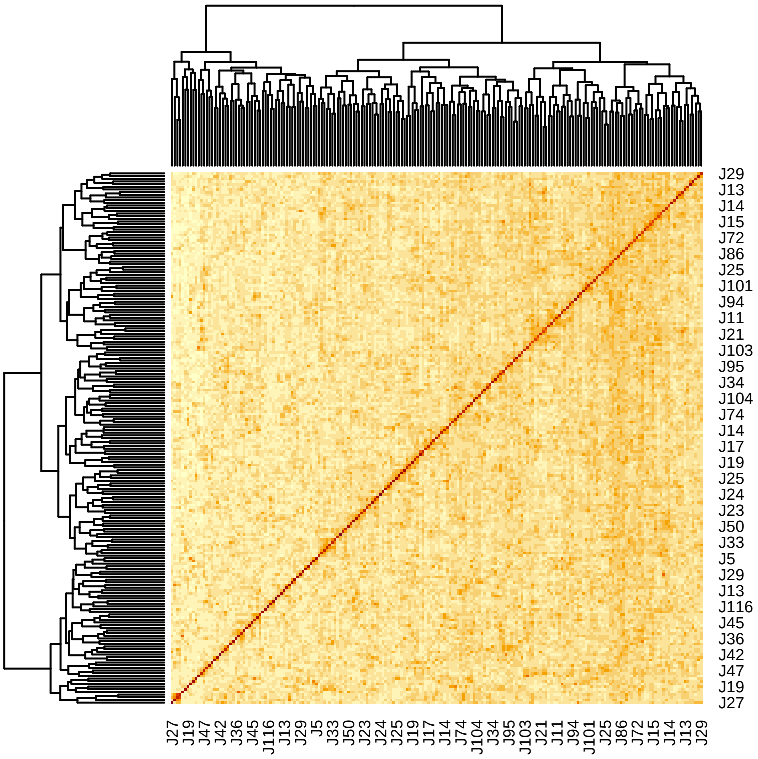
9.3 Get the brick of distance

The individual tables are transformed into distance cubes and DiSTATIS is performed on them.

9.4 Heatmap of Rv

The heatmap indicates the judges are strongly correlated with each other

```
heatmap(resDistatis$res4Cmat$C)
```

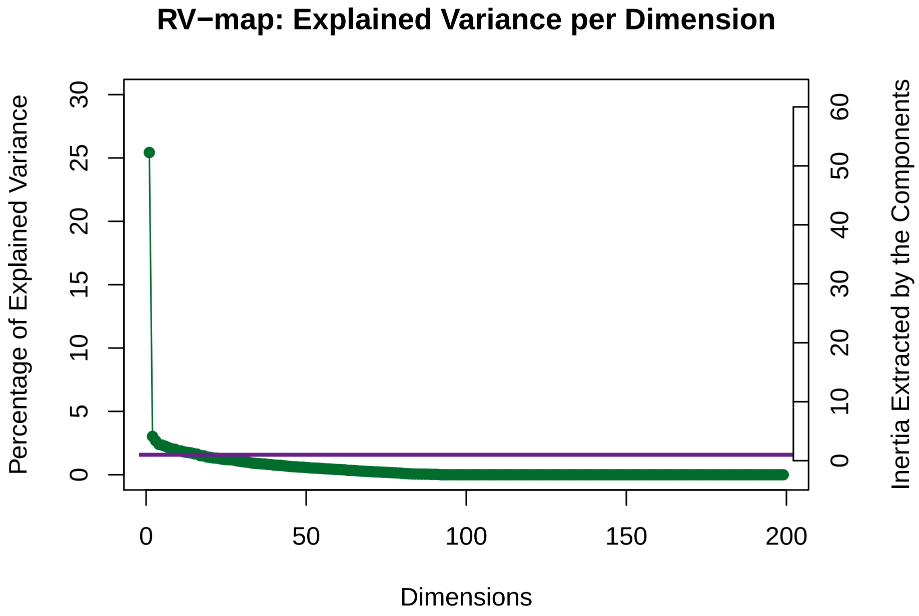


9.5 Projection of the vocabulary as supplementary elements

9.6 Scree Plot

From the Scree plot it is evident that 1 to 2 dimensions need to be interpreted.

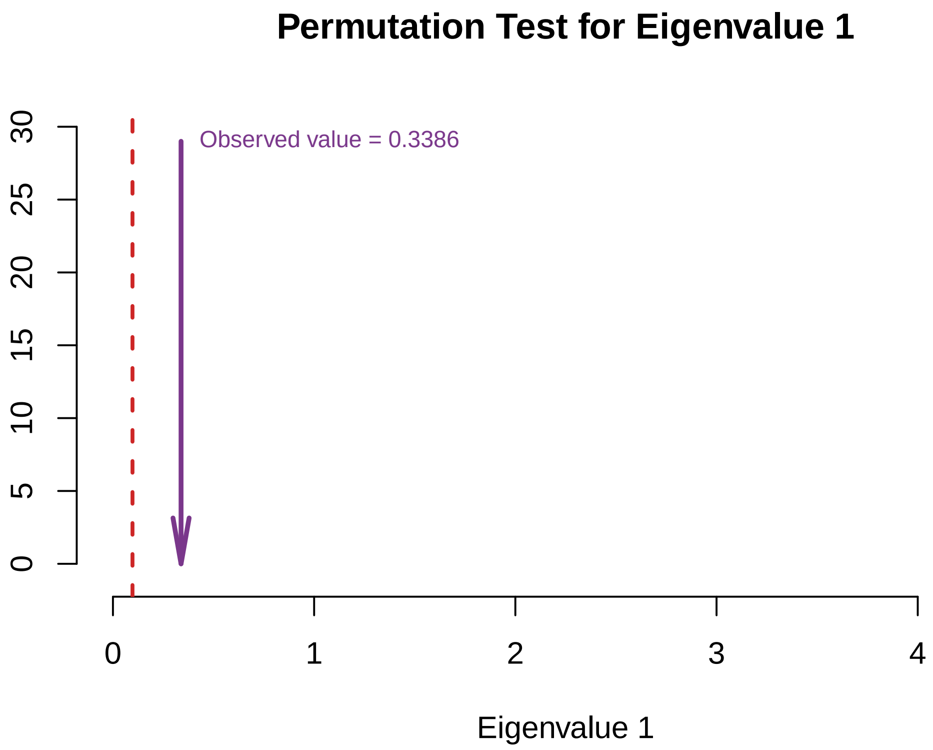
```
 # 5.A. A scree plot for the RV coef. Using standard plot (PTCA4CATA)
scree.rv.out <- PlotScree(ev = resDistatis$res4Cmat$eigValues,
title = "RV-map: Explained Variance per Dimension", plotKaiser = TRUE)
```



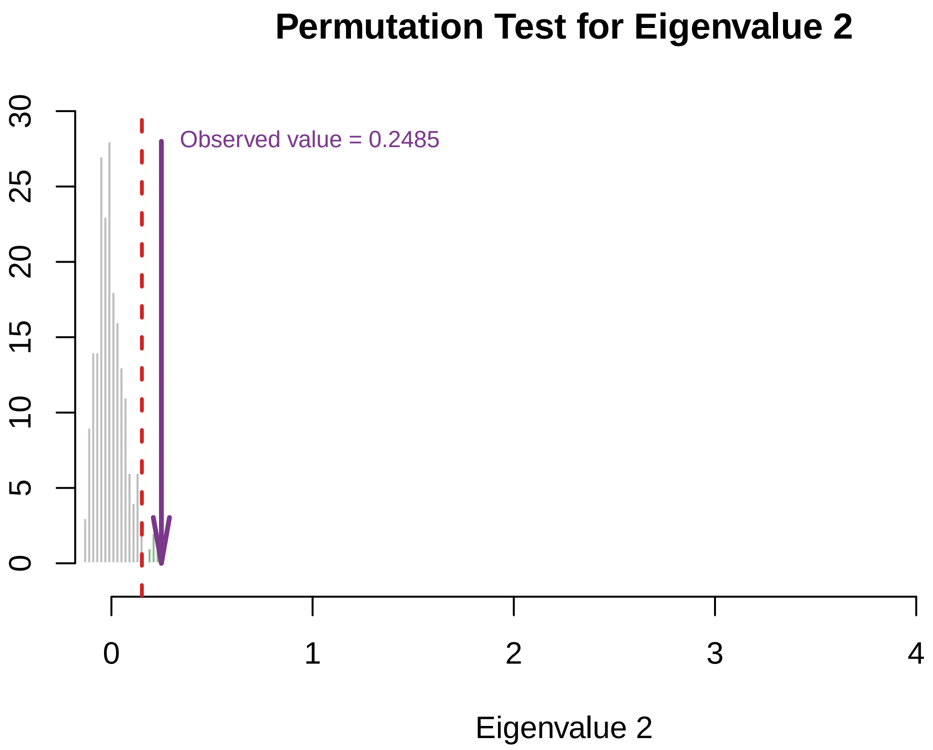
```
a1.Scree.RV <- recordPlot() # Save the plot
```

9.7 Permutations

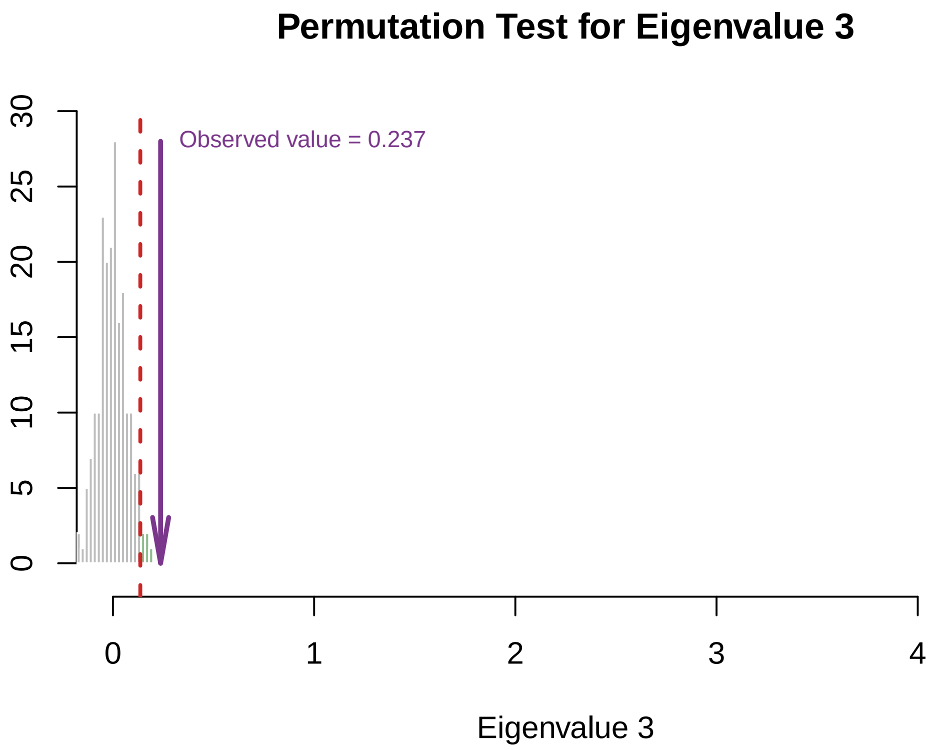
```
#Testing the eigenvalues
zeDim = 1
pH1 <- prettyHist(
 distribution = resDistatis$res4Cmat$eigVector[,zeDim],
          observed = resDistatis$res4Splus$eigValues[zeDim],
          xlim = c(0, 4.5), # needs to be set by hand
          breaks = 20,
          border = "white",
          main = paste0("Permutation Test for Eigenvalue ",zeDim),
          xlab = paste0("Eigenvalue ",zeDim),
          ylab = "",
          counts = FALSE,
          cutoffs = c( 0.975))
```



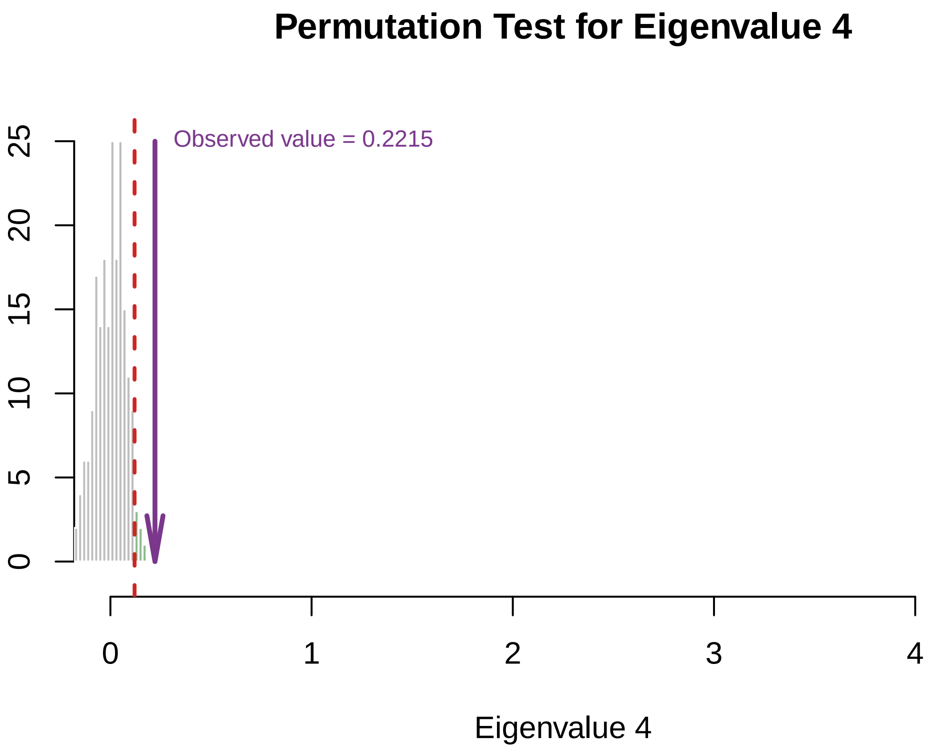
```
eigs1 <- recordPlot()

zeDim = 2
pH2 <- pH1 <- prettyHist(
distribution = resDistatis$res4Cmat$eigVector[,zeDim],
          observed = resDistatis$res4Splus$eigValues[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))
```



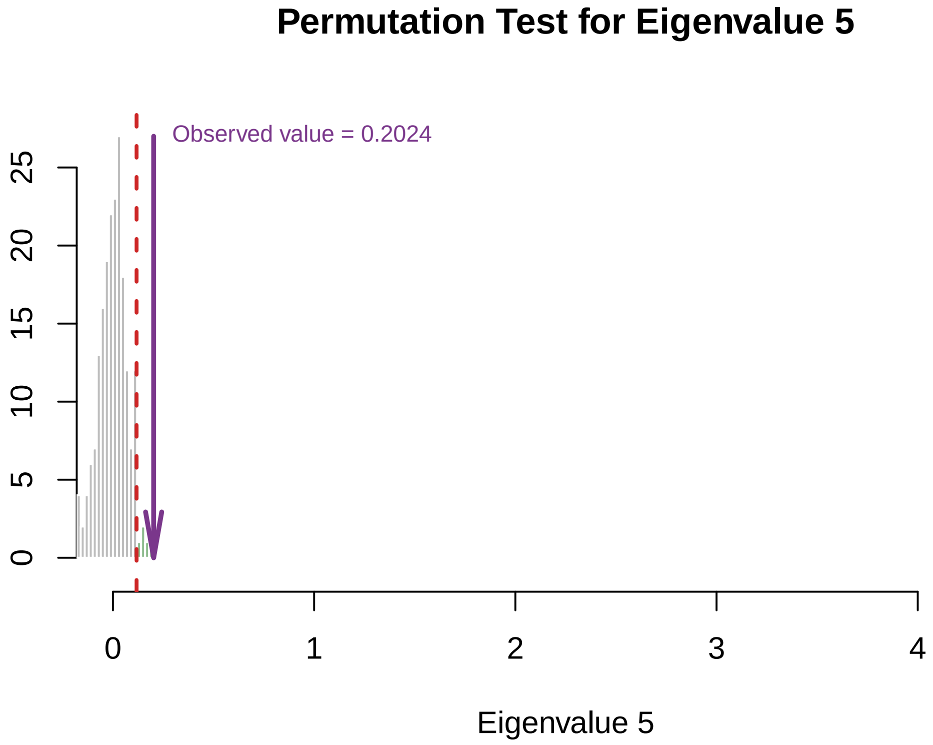
```
eigs2 <- recordPlot()

zeDim = 3
pH3 <- pH2 <- pH1 <- prettyHist(
distribution = resDistatis$res4Cmat$eigVector[,zeDim],
          observed = resDistatis$res4Splus$eigValues[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))
```



```
eigs3 <- recordPlot()

zeDim = 4
pH4 <- pH3 <- pH2 <- pH1 <- prettyHist(
distribution = resDistatis$res4Cmat$eigVector[,zeDim],
          observed = resDistatis$res4Splus$eigValues[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))
```



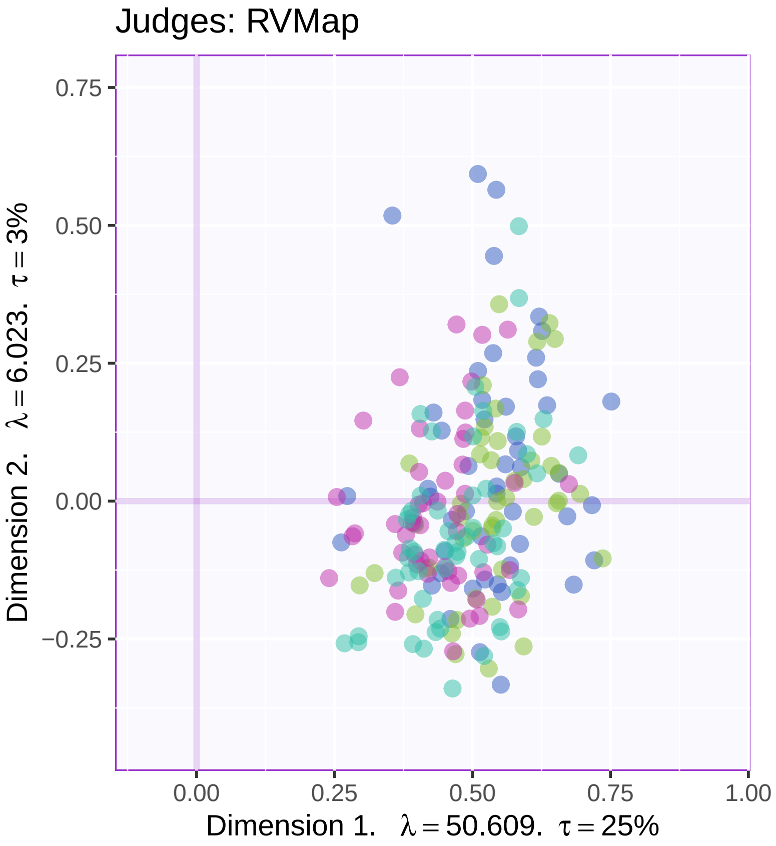
```
eigs4 <- recordPlot()

zeDim = 5
ph5 <- pH4 <- pH3 <- pH2 <- pH1 <- prettyHist(
distribution = resDistatis$res4Cmat$eigVector[,zeDim],
          observed = resDistatis$res4Splus$eigValues[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))
```



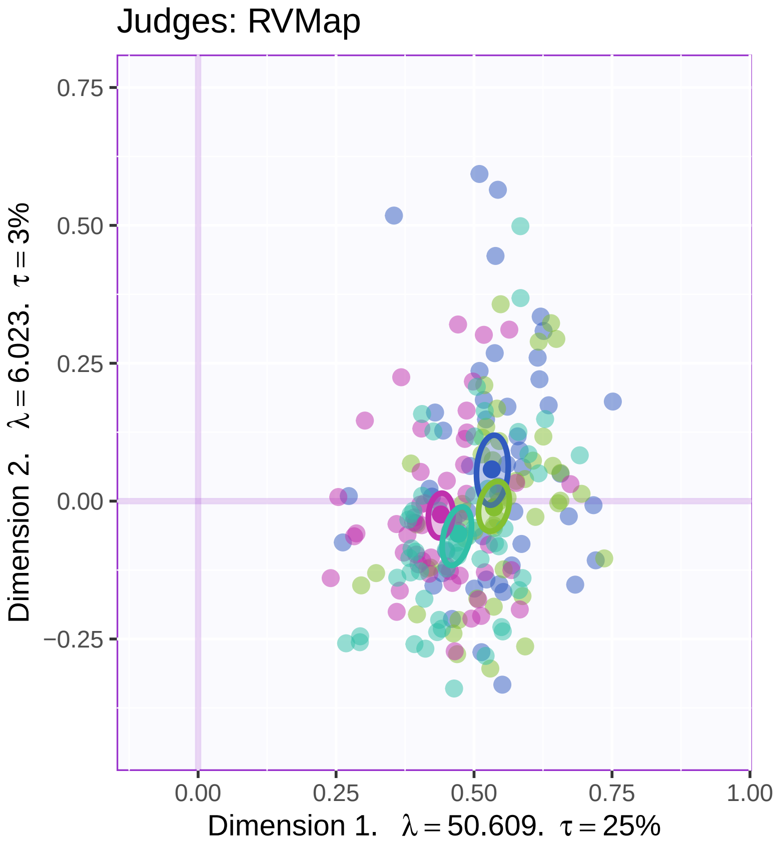
```
eigs5 <- recordPlot()
```

9.8 Factor Map for RV

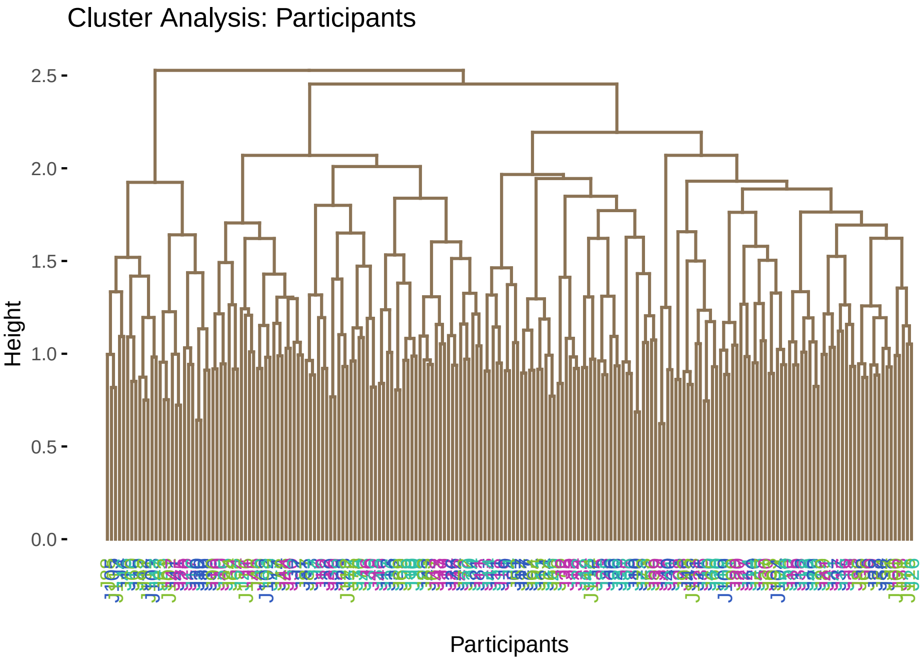
It is clearly evident that the 4 groups are centered and pretty close to each other in the Rv Factor map.



```r
print(a2d.gg.RVMap.CI )
```

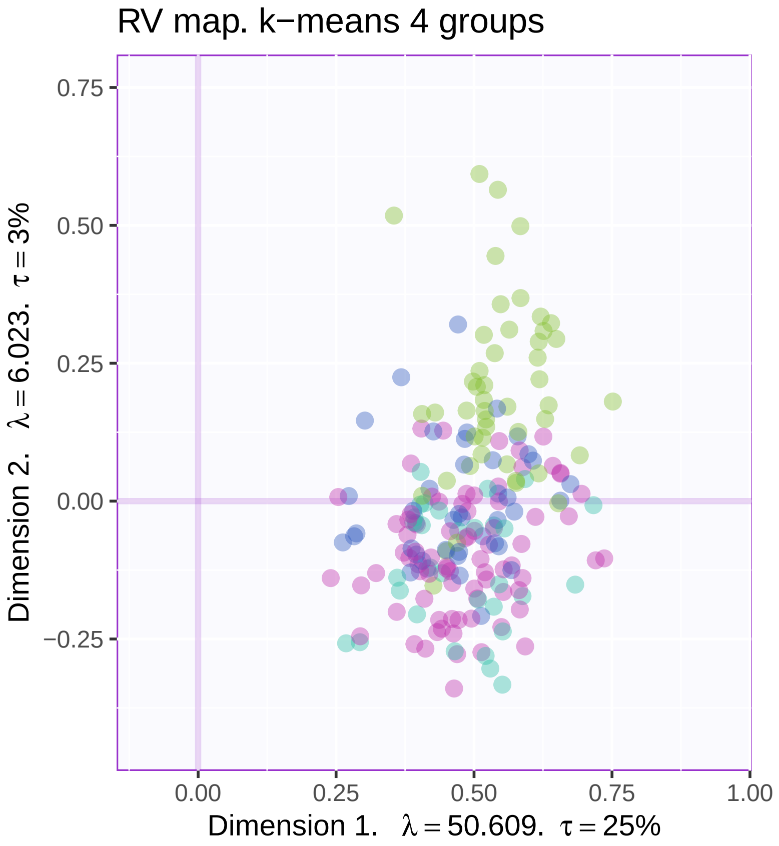


```r
print(a05.tree4participants)
```



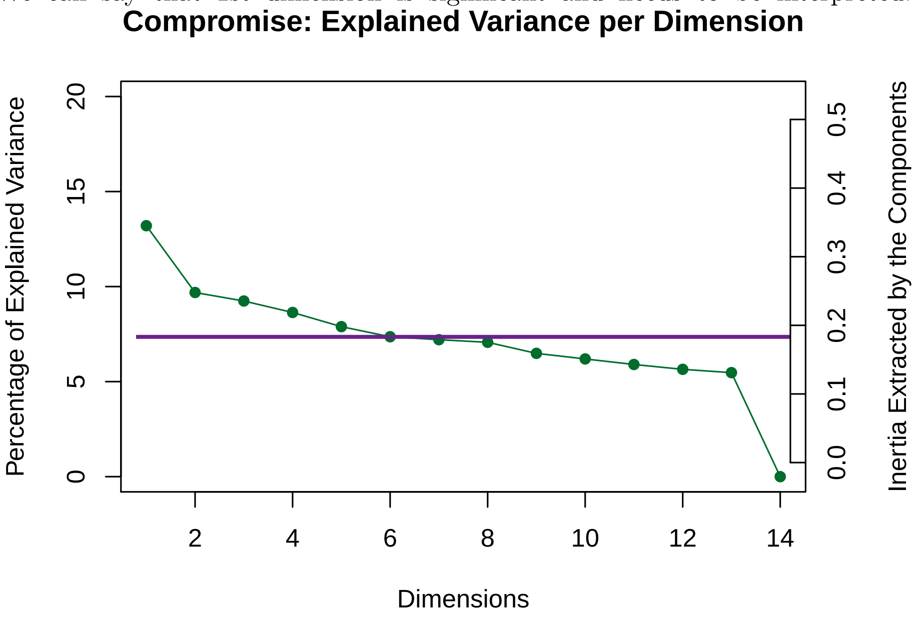
9.9 K-means

K-means tells us there are 4 groups that we can interpret and overlap each other mostly.

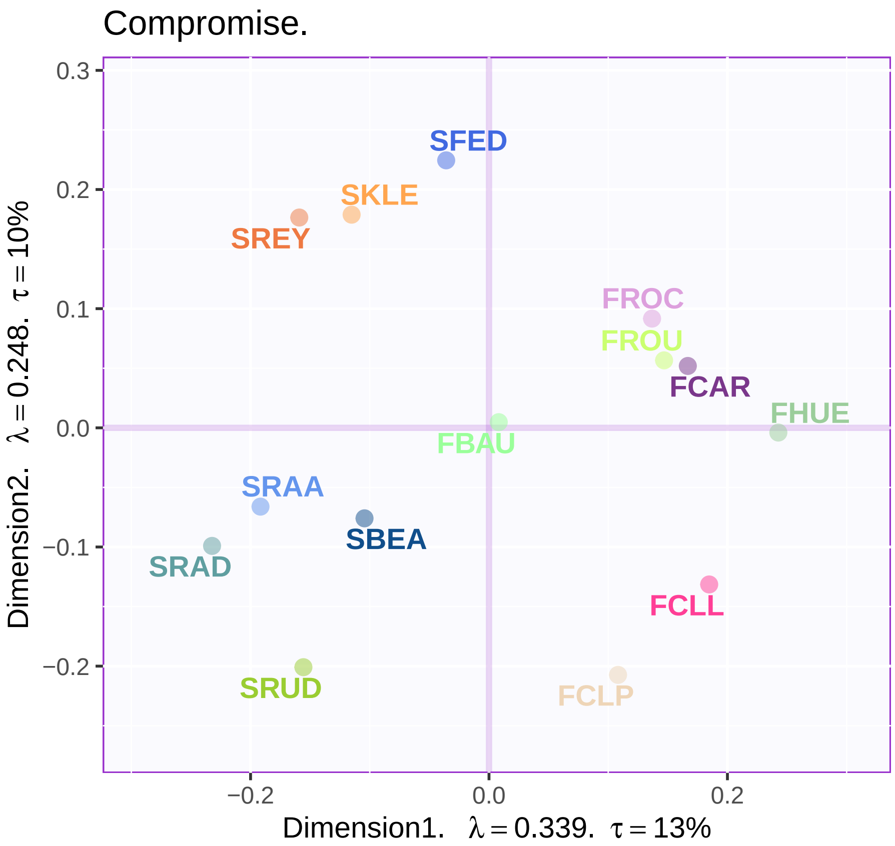

RV map. k−means 4 groups
Dimension 2. λ = 6.023. τ = 3%
Dimension 1. λ = 50.609. τ = 25%
0.75
0.50
0.25
0.00
−0.25
0.00
0.25
0.50
0.75
1.00

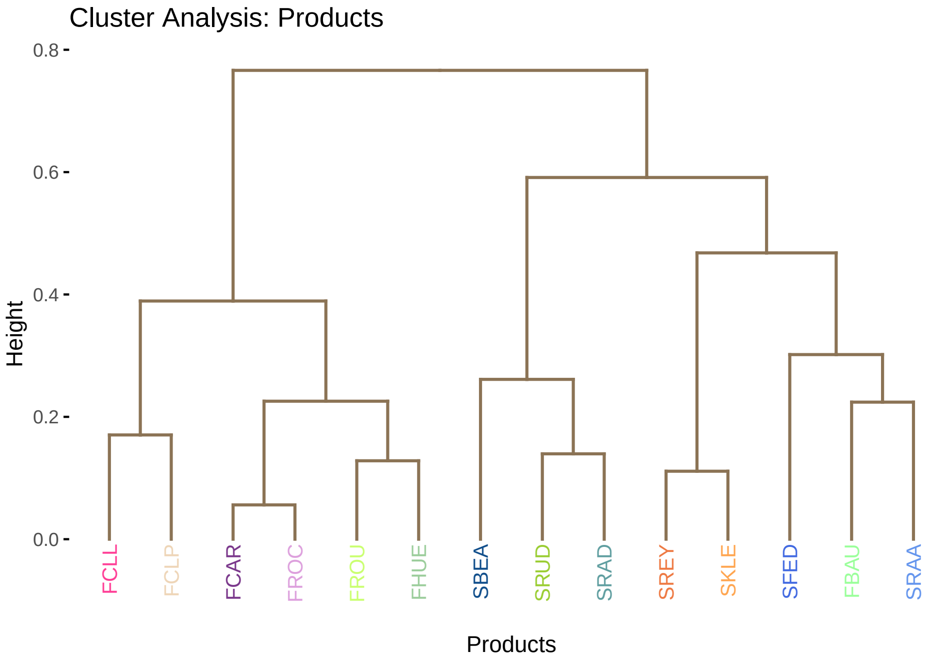
9.10 The Compromis: Scree Plot

We can say that 1st dimension is significant and needs to be interpreted.



9.11 The compromise

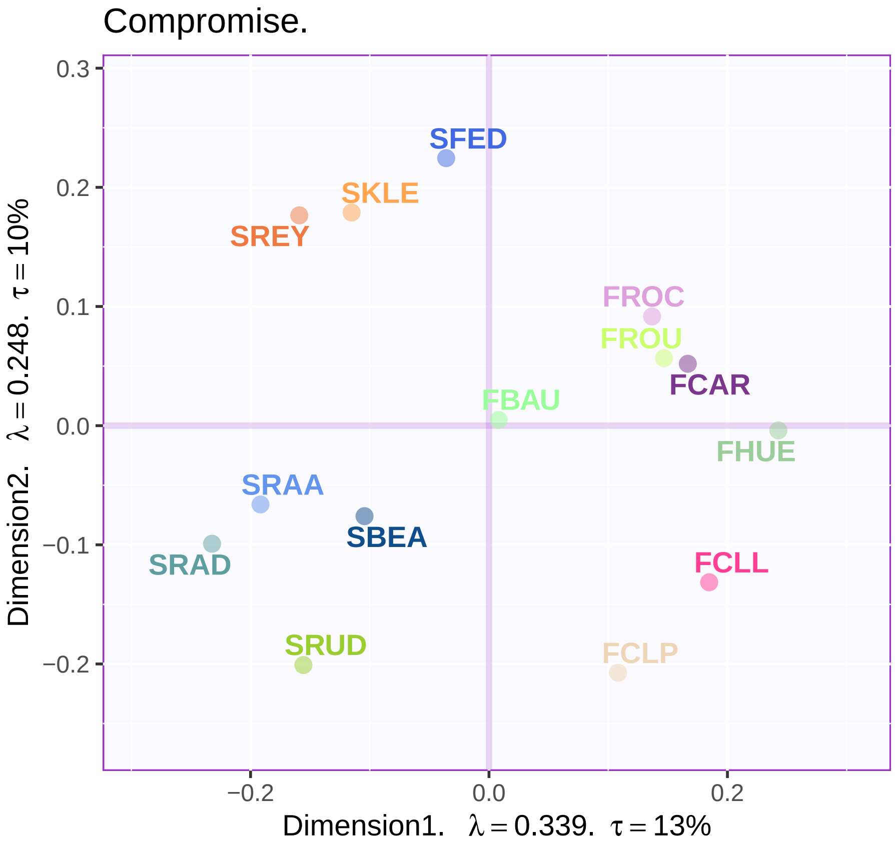




9.12 Map of compromise with partial factor scores

colored by wines

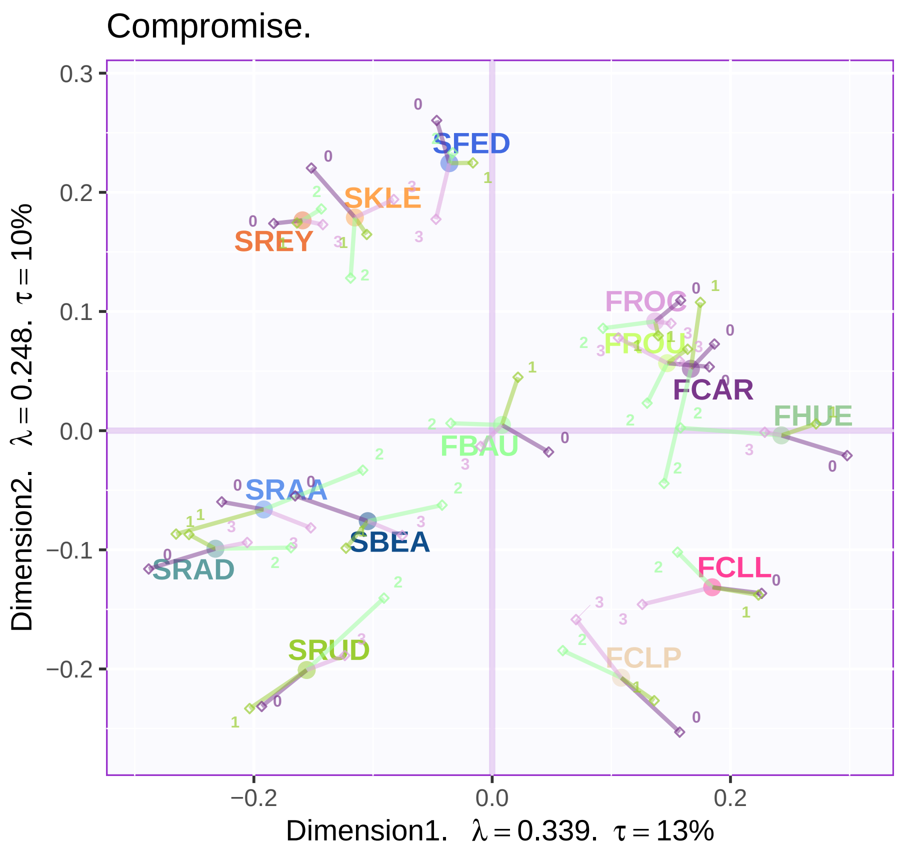
```
print(d1.partialFS.map.byProducts )
```



## Map of Compromise with partial factor scores: colored by Judges' groups It is observed that French and the South African judges are divided by DiSTATIS.

The number 0 signifies that the Judge is from South Africa and has information about the wines. The number 1 signifies that the Judge is from France and has information about the wines. The number 2 signifies that the Judge is from South Africa and has information about the wines. The number 3 signifies that the Judge is from France and has information about the wines.

```
print(d2.partialFS.map.byCategories)
```

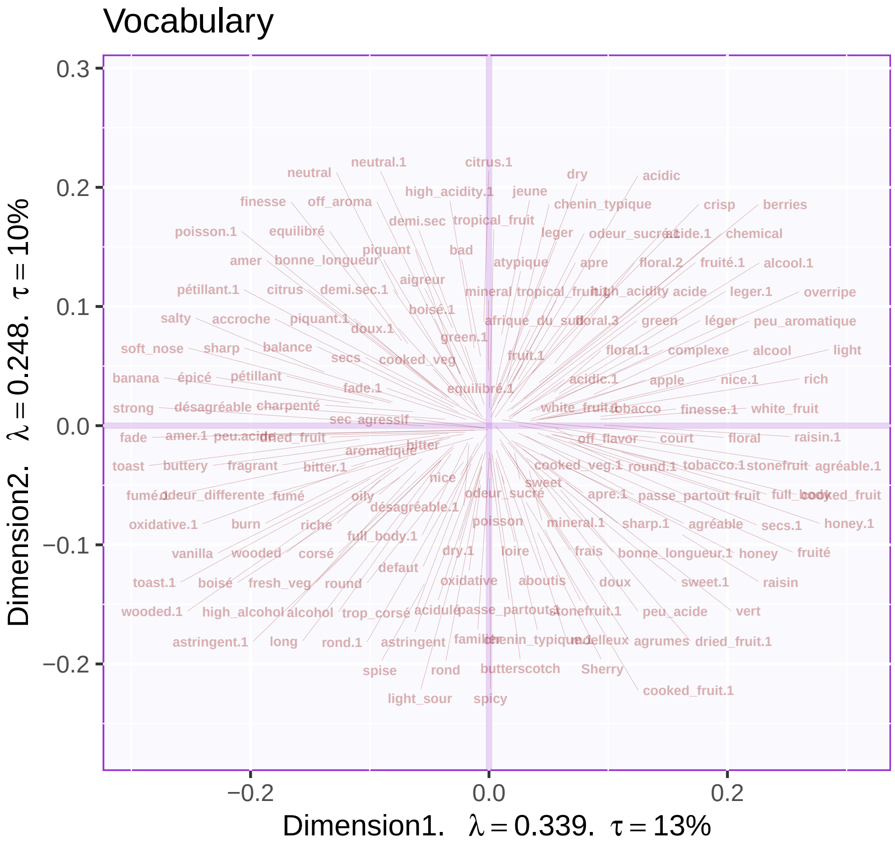


9.13 Create vocabulary graphs

In this plot we can observe that the wine properties are mapped onto the respective judges, either from France or South Africa.

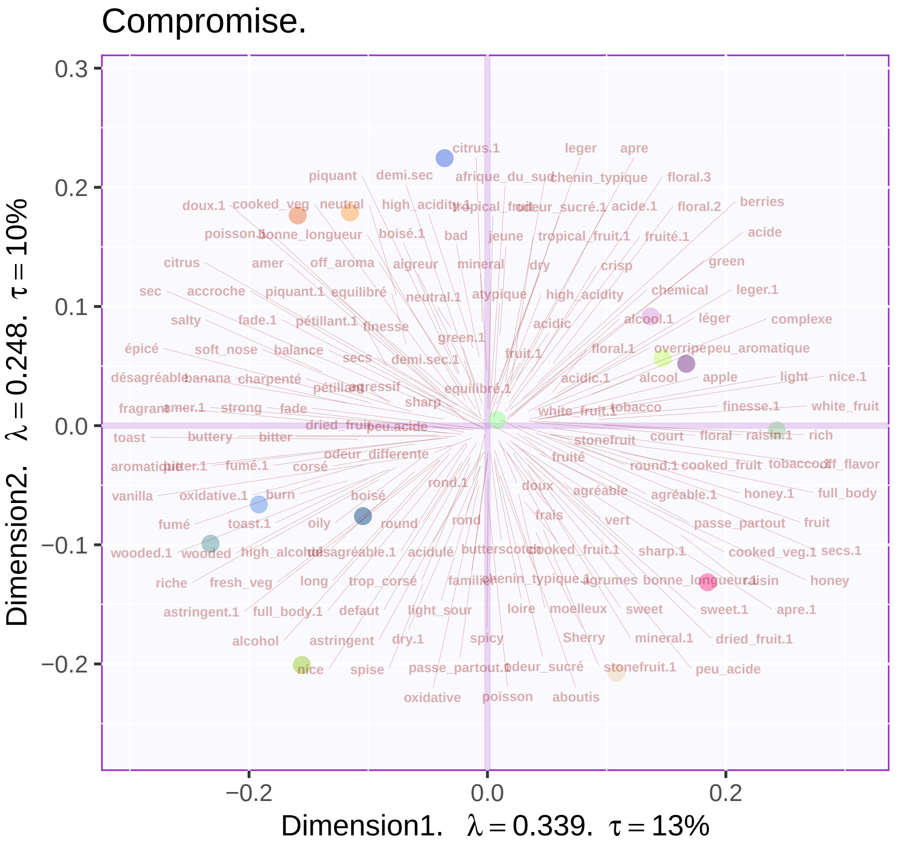
9.14 Print the graph vocabulary (without the wine dots)

```
print(e1.gg.voc.bary.gr)
```

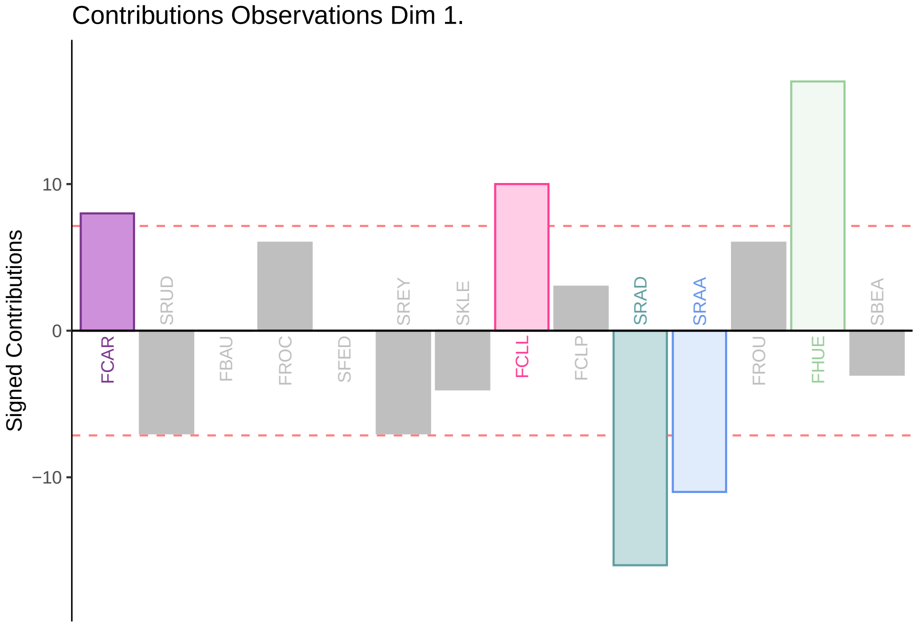


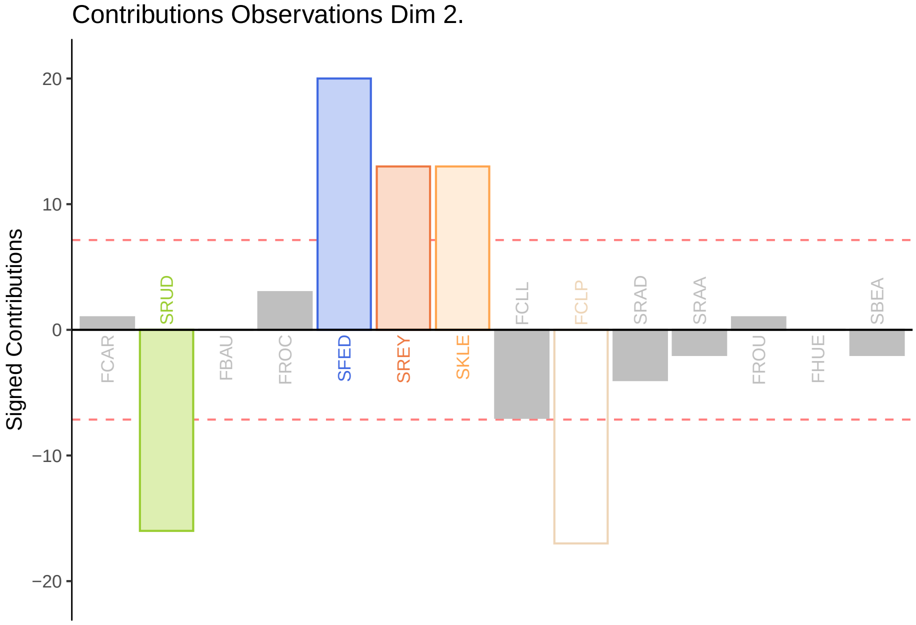
## Print the graph vocabulary (with the wine dots)

```
print(b5.gg.voc.bary.dots.gr)
```


Compromise.
0.3
0.2
0.1
0.0
−0.1
−0.2
Dimension2. λ = 0.248. τ = 10%
citrus.1
leger
apre
piquant
demi.sec
afrique_du_sud
chenin_typique
floral.3
doux.1
cooked_veg
neutral
high_acidity.1
tropical_fruit
odeur_sucré.1
acide.1
floral.2
berries
poisson.1
bonne_longueur
boisé.1
bad
jeune
tropical_fruit.1
fruité.1
acide
citrus
amer
off_aroma
aigreur
mineral
dry
crisp
green
sec
accroche
piquant.1
equilibré
neutral.1
atypique
high_acidity
chemical
leger.1
salty
fade.1
pétillant.1
finesse
acidic
alcool.1
léger
complexe
green.1
épicé
soft_nose
balance
secs
demi.sec.1
fruit.1
floral.1
overripe
peu_aromatique
désagréable
banana
charpenté
pétillant
agressif
equilibré.1
acidic.1
alcool
apple
light
nice.1
sharp
fragrant
amer.1
strong
fade
white_fruit.1
tobacco
finesse.1
white_fruit
dried_fruit
peu.acide
toast
buttery
bitter
stonefruit
court
floral
raisin.1
rich
odeur_differente
fruité
aromatique
bitter.1
fumé.1
corsé
round.1
cooked_fruit
tobacco.1
off_flavor
rond.1
doux
agréable
vanilla
oxidative.1
burn
boisé
agréable.1
honey.1
full_body
fumé
toast.1
oily
round
rond
frais
vert
passe_partout
fruit
wooded.1
wooded
high_alcohol
désagréable.1
acidulé
butterscotch
cooked_fruit.1
sharp.1
cooked_veg.1
secs.1
riche
fresh_veg
long
trop_corsé
familier
chenin_typique.1
agrumes
bonne_longueur.1
raisin
honey
astringent.1
full_body.1
defaut
light_sour
loire
moelleux
sweet
sweet.1
apre.1
alcohol
astringent
dry.1
spicy
Sherry
mineral.1
dried_fruit.1
nice
spise
passe_partout.1
odeur_sucré
stonefruit.1
peu_acide
oxidative
poisson
aboutis
−0.2
0.0
0.2
Dimension1. λ = 0.339. τ = 13%

9.15 Contributions





9.16 Summary

It is observed that French and the South African judges are divided by DiSTATIS.

The number 0 signifies that the Judge is from South Africa and has information about the wines. The number 1 signifies that the Judge is from France and has information about the wines. The number 2 signifies that the Judge is from South Africa and has information about the wines. The number 3 signifies that the Judge is from France and has information about the wines.