Multiple Correspondance Analysis

Richa Singh

10/10/2018

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

## Multiple Correspondance Analysis

Multiple correspondence analysis (MCA) is an extension of correspondence analysis (CA) which allows one to analyze the pattern of relationships of several categorical dependent variables. MCA is used to analyze a set of observations described by a set of nominal variables. Each nominal variable comprises several levels, and each of these levels is coded as a binary variable. MCA can also accommodate quantitative variables by recoding them as bins.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 correspondence analysis on an indicator matrix (a matrix whose entries are 0 or 1). The percentages of explained variance need to be corrected, and the correspondence analysis interpretation of inter-point distances needs to be adapted.

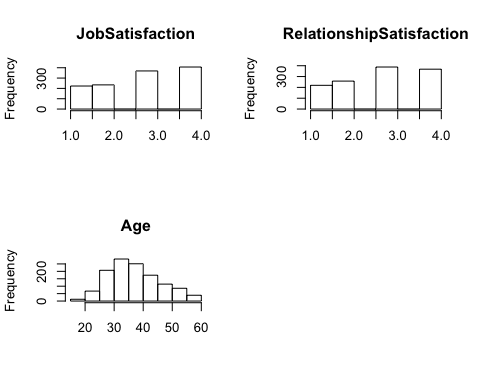
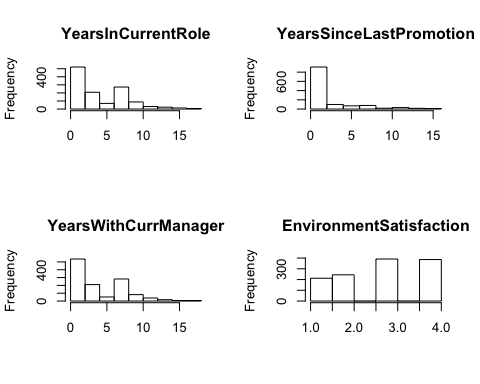
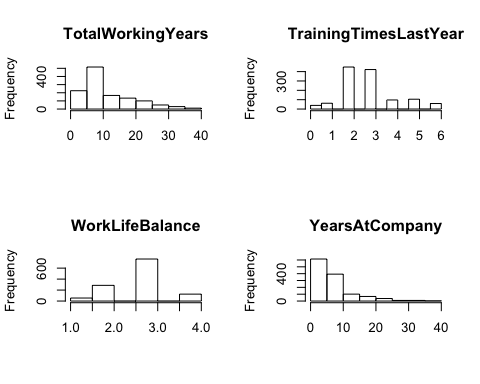
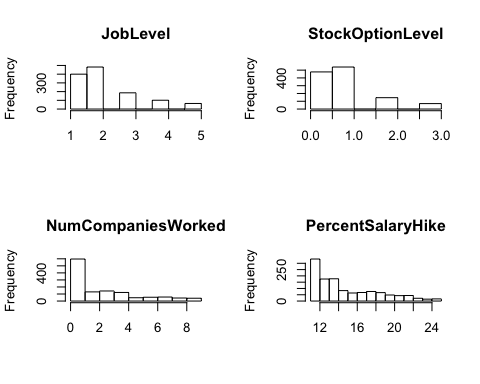
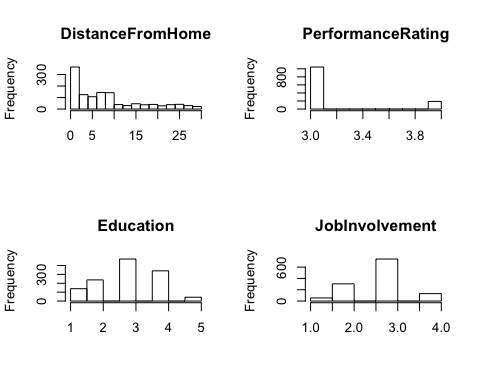
## Dataset

The dataset consists of 1233 observations and 32 variables describing the HR-IBM Employees. The variables Sub,Age,Monthly Income,Daily Rate,Hourly Rate, MonthlyRate,DistanceFromeHome,PerformanceRating, Education, JobInvolvement, Joblevel,StockOptionLevel,PercentSalaryHike,TotalWorkingYears,TrainingTimesLastYear,WorkLifeBalance,WorkLifeBalance,YearsAtCompany,YearsInCurrentRole,YearsSinceLastPromotion,YearsWithCurrManager,EnvironmentSatisfaction,JobSatisfaction,RelationshipSatisfaction,Attrition,BusinessTravel,EducationField,Gender,JobRole,MaritalStatus,OverTime are Qualitative variable.

my\_data <- read.csv("IBM-HR-Emplyee-NoAttrition.csv")  
cols <- colnames(my\_data)  
rownames(my\_data) <- my\_data$Subj  
data1 <- my\_data[,11:32]  
data1$Age <- my\_data$Age

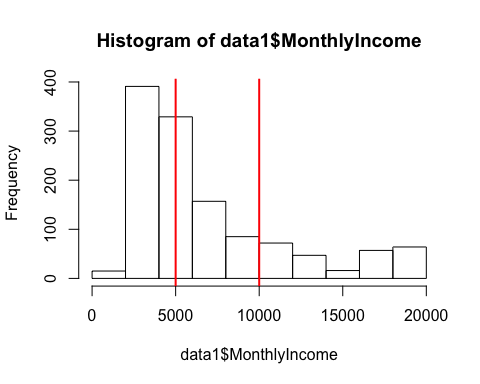
## Distribution of the quantitative data

col\_no = 1  
par(mfrow=c(2,2))   
for (val in colnames(data1)){  
 hist(data1[,col\_no],breaks = 10, main=val, xlab = "")  
 col\_no = col\_no +1  
}



## Binning the quantitative data

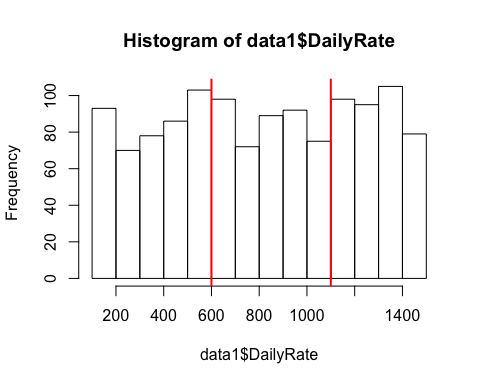
hist(data1$MonthlyIncome)   
abline(v = c(5000,10000), col = "red", lwd =2)



# Binning of Monthly Income   
data1$MonthlyIncome <- cut(data1$MonthlyIncome,breaks = c(min(data1$MonthlyIncome)-1,5000 ,10000, max(data1$MonthlyIncome)+1),labels = c(1,2,3))  
cor(as.numeric(data1$MonthlyIncome),my\_data$MonthlyIncome,method = "spearman")

## [1] 0.9229362

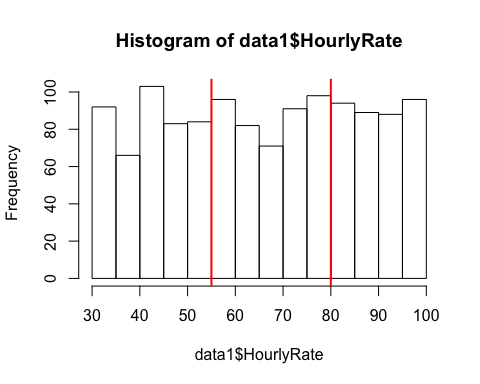
hist(data1$DailyRate)   
abline(v = c(600,1100), col = "red", lwd =2)



data1$DailyRate <- cut(data1$DailyRate,breaks = c(min(data1$DailyRate)-1,600 ,1100, max(data1$DailyRate)+1),labels = c(1,2,3))  
cor(as.numeric(data1$DailyRate),my\_data$DailyRate,method = "spearman")

## [1] 0.9422108

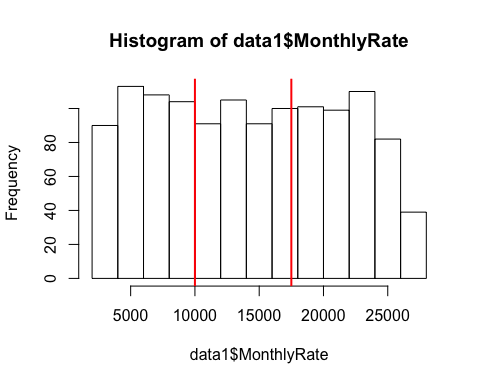
hist(data1$HourlyRate)   
abline(v = c(55,80), col = "red", lwd =2)



#Binning hourly rate  
data1$HourlyRate <- cut(data1$HourlyRate,breaks = c(min(data1$HourlyRate)-1,55 ,80, max(data1$HourlyRate)+1),labels = c(1,2,3))  
cor(as.numeric(data1$HourlyRate),my\_data$HourlyRate,method = "spearman")

## [1] 0.9419043

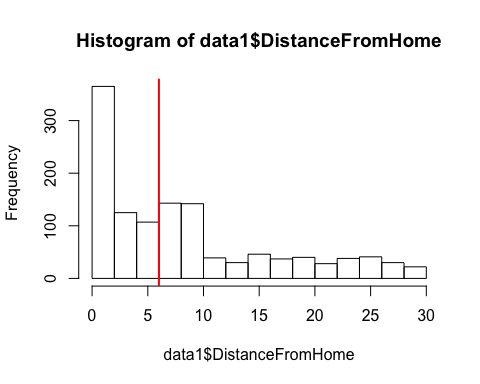
hist(data1$MonthlyRate)   
abline(v = c(10000,17500), col = "red", lwd =2)



#Binning Monthly rate   
data1$MonthlyRate <- cut(data1$MonthlyRate,breaks = c(min(data1$MonthlyRate)-1,10000 ,17500, max(data1$MonthlyRate)+1),labels = c(1,2,3))  
cor(as.numeric(data1$MonthlyRate),my\_data$MonthlyRate,method = "spearman")

## [1] 0.9412661

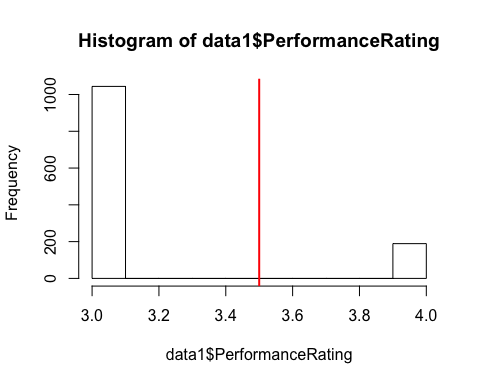
hist(data1$DistanceFromHome)   
abline(v = c(6), col = "red", lwd =2)



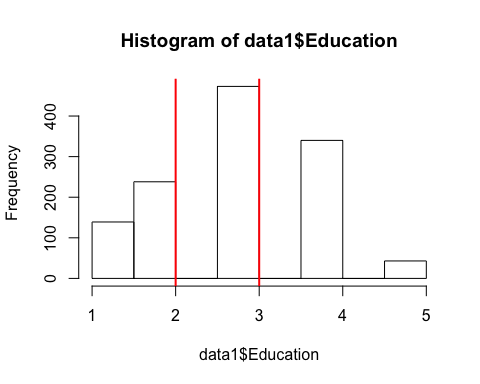
#Binning Distance From Home   
data1$DistanceFromHome <- cut(data1$DistanceFromHome,breaks = c(min(data1$DistanceFromHome)-1,6 , max(data1$DistanceFromHome)+1),labels = c(1,2))  
cor(as.numeric(data1$DistanceFromHome),my\_data$DistanceFromHome,method = "spearman")

## [1] 0.8689751

hist(data1$PerformanceRating)   
abline(v = c(3.5), col = "red", lwd =2)



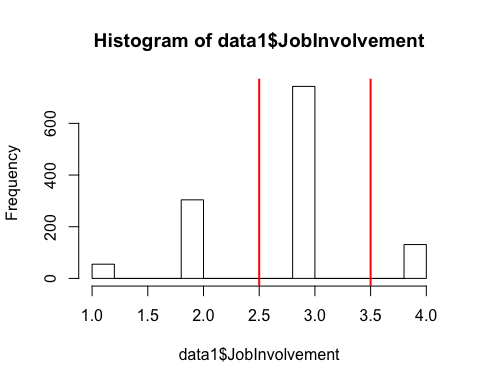
hist(data1$Education)   
abline(v = c(2,3), col = "red", lwd =2)



#Binning for Education  
data1$Education <- cut(data1$Education,breaks = c(min(data1$Education)-1, 2,3 , max(data1$Education)+1),labels = c(1,2,3))  
cor(as.numeric(data1$Education),my\_data$Education,method = "spearman")

## [1] 0.9840498

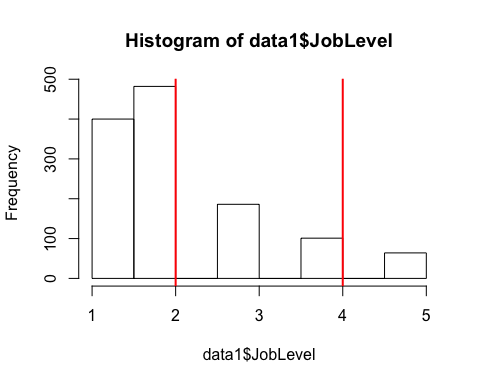
hist(data1$JobInvolvement)   
abline(v = c(2.5,3.5), col = "red", lwd =2)



#Binning for Job Involvement   
data1$JobInvolvement <- cut(data1$JobInvolvement,breaks = c(min(data1$JobInvolvement)-1,2.5,3.5, max(data1$JobInvolvement)+1),labels = c(1,2,3))  
cor(as.numeric(data1$JobInvolvement),my\_data$JobInvolvement,method = "spearman")

## [1] 0.9937007

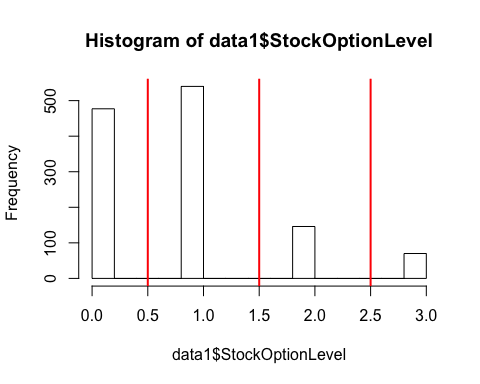
hist(data1$JobLevel)   
abline(v = c(2,4), col = "red", lwd =2)



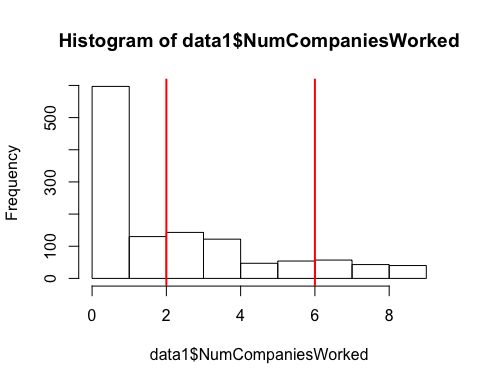
#Binning JobLevel  
data1$JobLevel <- cut(data1$JobLevel,breaks = c(min(data1$JobLevel)-1,2,4, max(data1$JobLevel)+1),labels = c(1,2,3))  
cor(as.numeric(data1$JobLevel),my\_data$JobLevel,method = "spearman")

## [1] 0.8298885

hist(data1$StockOptionLevel)   
abline(v = c(0.5,1.5,2.5), col = "red", lwd =2)

 Leaving the Stock Option Level as it is

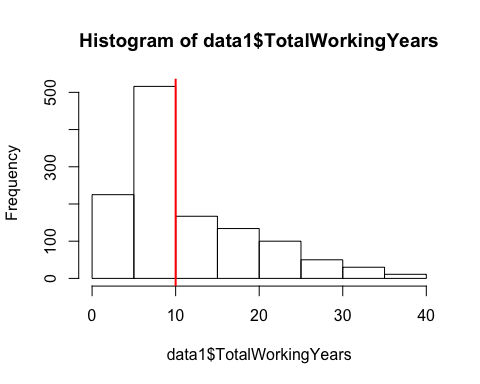
hist(data1$NumCompaniesWorked)   
abline(v = c(2,6), col = "red", lwd =2)



#Binning Number of Companies Worked   
data1$NumCompaniesWorked <- cut(data1$NumCompaniesWorked,breaks = c(min(data1$NumCompaniesWorked)-1, 2,6, max(data1$NumCompaniesWorked)+1),labels = c(1,2,3))  
cor(as.numeric(data1$NumCompaniesWorked),my\_data$NumCompaniesWorked,method = "spearman")

## [1] 0.8974507

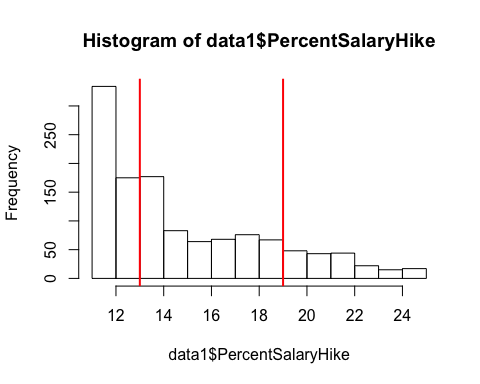
hist(data1$TotalWorkingYears)   
abline(v = c(10), col = "red", lwd =2)



## Binning total working years  
data1$TotalWorkingYears <- cut(data1$TotalWorkingYears,breaks = c(min(data1$TotalWorkingYears)-1, 10, max(data1$TotalWorkingYears)+1),labels = c(1,2))  
cor(as.numeric(data1$TotalWorkingYears),my\_data$TotalWorkingYears,method = "spearman")

## [1] 0.8502664

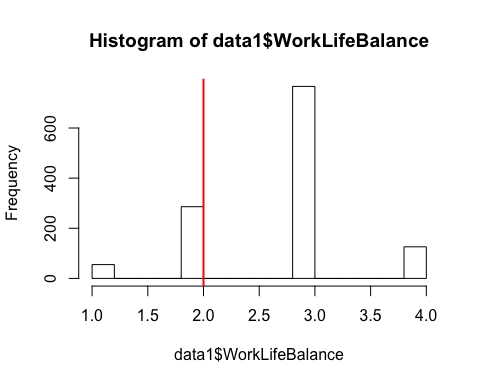
hist(data1$PercentSalaryHike)   
abline(v = c(13,19), col = "red", lwd =2)



## Binning percentage Salary Hike  
data1$PercentSalaryHike <- cut(data1$PercentSalaryHike,breaks = c(min(data1$PercentSalaryHike)-1, 13,19, max(data1$PercentSalaryHike)+1),labels = c(1,2,3))  
cor(as.numeric(data1$PercentSalaryHike),my\_data$PercentSalaryHike,method = "spearman")

## [1] 0.9244308

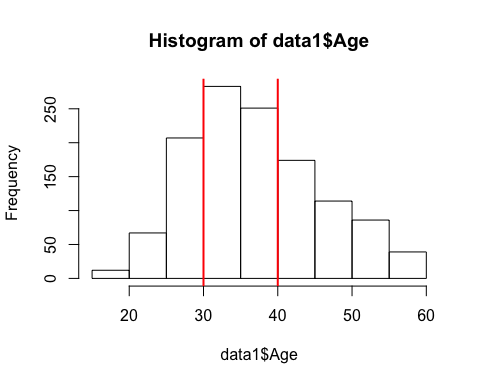
hist(data1$WorkLifeBalance)   
abline(v = c(2), col = "red", lwd =2)



## Binning WorkLifeBalance  
data1$WorkLifeBalance <- cut(data1$WorkLifeBalance,breaks = c(min(data1$WorkLifeBalance)-1, 2, max(data1$WorkLifeBalance)+1),labels = c(1,2))  
cor(as.numeric(data1$PercentSalaryHike),my\_data$PercentSalaryHike,method = "spearman")

## [1] 0.9244308

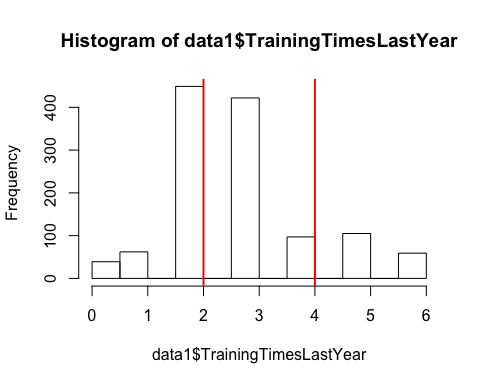
hist(data1$Age)   
abline(v = c(30,40), col = "red", lwd =2)



## Binning Age   
data1$Age <- cut(data1$Age,breaks = c(min(data1$Age)-1, 30,40, max(data1$Age)+1),labels = c(1,2,3))  
cor(as.numeric(data1$Age),my\_data$Age,method = "spearman")

## [1] 0.9326494

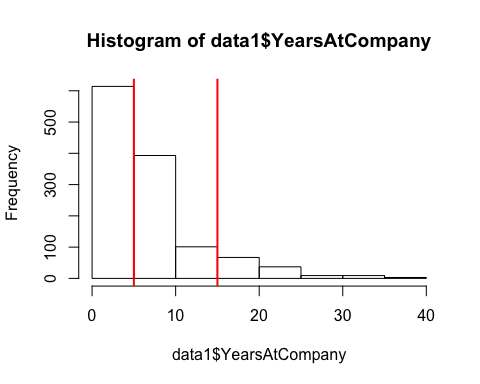
hist(data1$TrainingTimesLastYear)   
abline(v = c(2,4), col = "red", lwd =2)



## Binning TrainingTimesLastYear  
data1$TrainingTimesLastYear <- cut(data1$TrainingTimesLastYear,breaks = c(min(data1$TrainingTimesLastYear)-1, 2,4, max(data1$TrainingTimesLastYear)+1),labels = c(1,2,3))  
cor(as.numeric(data1$TrainingTimesLastYear),my\_data$TrainingTimesLastYear,method = "spearman")

## [1] 0.9573808

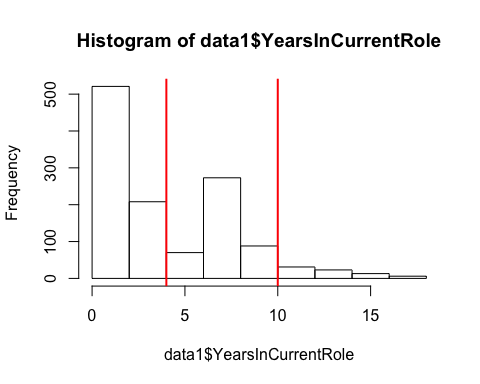
hist(data1$YearsAtCompany)   
abline(v = c(5,15), col = "red", lwd =2)



## Binning Years at Company   
data1$YearsAtCompany <- cut(data1$YearsAtCompany,breaks = c(min(data1$YearsAtCompany)-1, 5,15, max(data1$YearsAtCompany)+1),labels = c(1,2,3))  
cor(as.numeric(data1$YearsAtCompany),my\_data$YearsAtCompany,method = "spearman")

## [1] 0.9036617

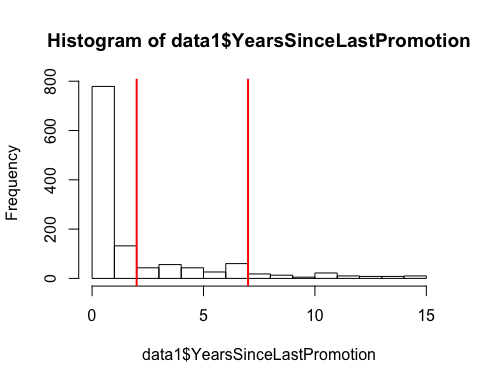
hist(data1$YearsInCurrentRole)   
abline(v = c(4,10), col = "red", lwd =2)



## Binning Years in Current Role   
data1$YearsInCurrentRole <- cut(data1$YearsInCurrentRole,breaks = c(min(data1$YearsInCurrentRole)-1, 4,10, max(data1$YearsInCurrentRole)+1),labels = c(1,2,3))  
cor(as.numeric(data1$YearsInCurrentRole),my\_data$YearsInCurrentRole,method = "spearman")

## [1] 0.8764758

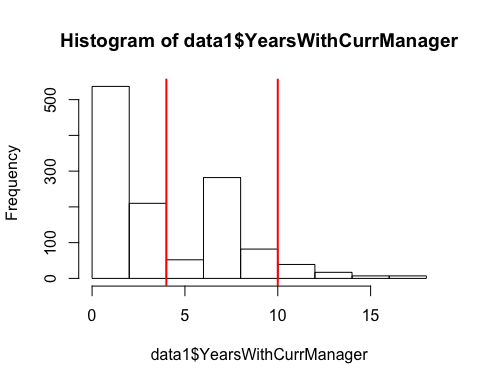
hist(data1$YearsSinceLastPromotion)   
abline(v = c(2,7), col = "red", lwd =2)



## Binning Years since last Promotion  
data1$YearsSinceLastPromotion <- cut(data1$YearsSinceLastPromotion,breaks = c(min(data1$YearsSinceLastPromotion)-1, 2,7, max(data1$YearsSinceLastPromotion)+1),labels = c(1,2,3))  
cor(as.numeric(data1$YearsSinceLastPromotion),my\_data$YearsSinceLastPromotion,method = "spearman")

## [1] 0.7976611

hist(data1$YearsWithCurrManager)   
abline(v = c(4,10), col = "red", lwd =2)



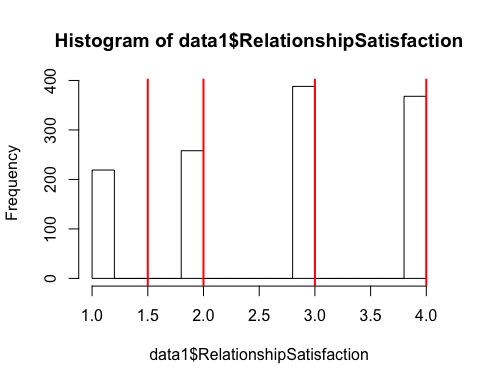
## Binning Years with current Manager   
data1$YearsWithCurrManager <- cut(data1$YearsWithCurrManager,breaks = c(min(data1$YearsWithCurrManager)-1, 4,10, max(data1$YearsWithCurrManager)+1),labels = c(1,2,3))  
cor(as.numeric(data1$YearsWithCurrManager),my\_data$YearsWithCurrManager,method = "spearman")

## [1] 0.8693187

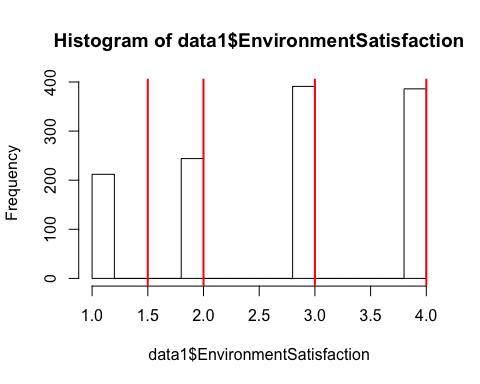
hist(data1$JobSatisfaction)   
abline(v = c(1.5,2,3,4), col = "red", lwd =2)



hist(data1$RelationshipSatisfaction)   
abline(v = c(1.5,2,3,4), col = "red", lwd =2)



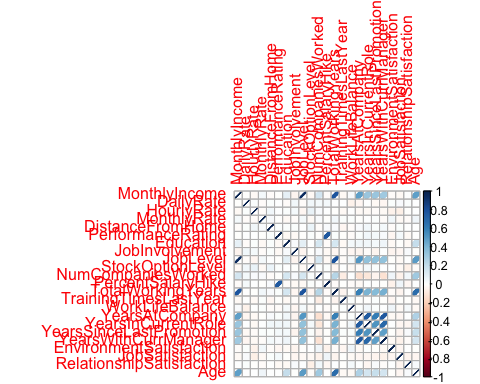
hist(data1$EnvironmentSatisfaction)   
abline(v = c(1.5,2,3,4), col = "red", lwd =2)

 ## Heat Map

library(corrplot)

## corrplot 0.84 loaded

data2 <- my\_data[,11:32]  
data2$Age <- my\_data$Age  
cor.my\_data <- cor(data2)  
corrplot(cor.my\_data, method = "ellipse")



library(data4PCCAR)  
library(PTCA4CATA)  
library(corrplot)  
library(ggplot2)  
library(ExPosition)

## Loading required package: prettyGraphs

library(InPosition)

##   
## Attaching package: 'InPosition'

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

# Heat Map for Burt Table  
#heatMapIJ <- makeggHeatMap4CT(data2,colorAttributes = col4J, fontSize.x = 15)  
#print(heatMapIJ)

str(data1)

## 'data.frame': 1233 obs. of 23 variables:  
## $ MonthlyIncome : Factor w/ 3 levels "1","2","3": 2 1 1 1 1 1 2 2 1 1 ...  
## $ DailyRate : Factor w/ 3 levels "1","2","3": 1 3 1 2 3 3 1 3 2 1 ...  
## $ HourlyRate : Factor w/ 3 levels "1","2","3": 2 2 1 2 3 2 1 3 3 1 ...  
## $ MonthlyRate : Factor w/ 3 levels "1","2","3": 3 3 2 2 1 2 1 2 2 2 ...  
## $ DistanceFromHome : Factor w/ 2 levels "1","2": 2 1 1 1 1 2 2 2 2 2 ...  
## $ PerformanceRating : int 4 3 3 3 4 4 4 3 3 3 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 1 3 1 1 2 1 2 2 2 1 ...  
## $ JobInvolvement : Factor w/ 3 levels "1","2","3": 1 2 2 2 3 2 1 2 3 1 ...  
## $ JobLevel : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 2 1 1 1 ...  
## $ StockOptionLevel : int 1 0 1 0 3 1 0 2 1 0 ...  
## $ NumCompaniesWorked : Factor w/ 3 levels "1","2","3": 1 1 3 1 2 1 1 2 1 1 ...  
## $ PercentSalaryHike : Factor w/ 3 levels "1","2","3": 3 1 1 1 3 3 3 1 1 1 ...  
## $ TotalWorkingYears : Factor w/ 2 levels "1","2": 1 1 1 1 2 1 1 2 1 1 ...  
## $ TrainingTimesLastYear : Factor w/ 3 levels "1","2","3": 2 2 2 1 2 1 1 2 3 2 ...  
## $ WorkLifeBalance : Factor w/ 2 levels "1","2": 2 2 2 1 1 2 2 1 2 2 ...  
## $ YearsAtCompany : Factor w/ 3 levels "1","2","3": 2 2 1 2 1 1 2 2 1 2 ...  
## $ YearsInCurrentRole : Factor w/ 3 levels "1","2","3": 2 2 1 2 1 1 2 2 1 2 ...  
## $ YearsSinceLastPromotion : Factor w/ 3 levels "1","2","3": 1 2 1 2 1 1 1 2 1 1 ...  
## $ YearsWithCurrManager : Factor w/ 3 levels "1","2","3": 2 1 1 2 1 1 2 2 1 2 ...  
## $ EnvironmentSatisfaction : int 3 4 1 4 3 4 4 3 1 4 ...  
## $ JobSatisfaction : int 2 3 2 4 1 3 3 3 2 3 ...  
## $ RelationshipSatisfaction: int 4 3 4 3 1 2 2 2 3 4 ...  
## $ Age : Factor w/ 3 levels "1","2","3": 3 2 1 2 3 1 2 2 2 1 ...

data1$PerformanceRating <- as.factor(data1$PerformanceRating)  
data1$JobInvolvement <- as.factor(data1$JobInvolvement)  
data1$StockOptionLevel <- as.factor(data1$StockOptionLevel)  
data1$EnvironmentSatisfaction <- as.factor(data1$EnvironmentSatisfaction)  
data1$JobSatisfaction <- as.factor(data1$JobSatisfaction)  
data1$RelationshipSatisfaction <- as.factor(data1$RelationshipSatisfaction)

## Scree plot+ Inference Results

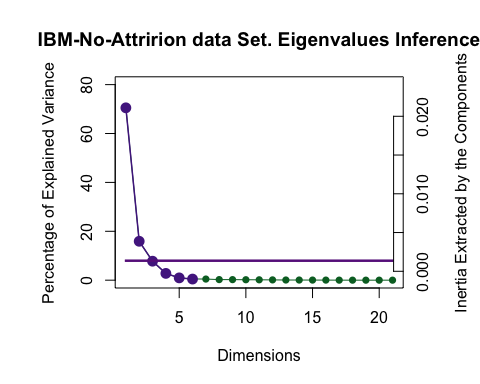
resMCA.sym <- epMCA(data1 ,make\_data\_nominal = TRUE ,DESIGN = my\_data$Department ,make\_design\_nominal = TRUE,graphs = FALSE, symmetric = TRUE)  
  
resMCA.asym <- epMCA(data1 ,make\_data\_nominal = TRUE ,DESIGN = my\_data$Department ,make\_design\_nominal = TRUE,graphs = FALSE, symmetric = FALSE)  
  
resMCA.inf <- epMCA.inference.battery(data1, make\_data\_nominal = TRUE, DESIGN = my\_data$Department ,make\_design\_nominal =TRUE,graphs = FALSE)

## [1] "It is estimated that your iterations will take 0.15 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

resMCA.sym1 <- epMCA(data1 ,make\_data\_nominal = TRUE ,DESIGN = my\_data$Gender ,make\_design\_nominal = TRUE,graphs = FALSE, symmetric = TRUE)  
  
resMCA.asym1 <- epMCA(data1 ,make\_data\_nominal = TRUE ,DESIGN = my\_data$Gender ,make\_design\_nominal = TRUE,graphs = FALSE, symmetric = FALSE)  
  
resMCA.inf1 <- epMCA.inference.battery(data1, make\_data\_nominal = TRUE, DESIGN = my\_data$Gender ,make\_design\_nominal =TRUE,graphs = FALSE)

## [1] "It is estimated that your iterations will take 0.13 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

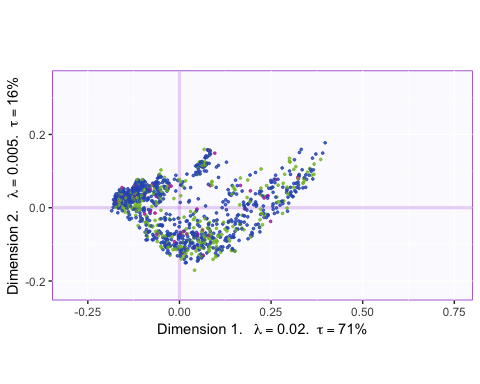
PlotScree(ev = resMCA.sym$ExPosition.Data$eigs,   
 p.ev = resMCA.inf$Inference.Data$components$p.vals,  
 title = 'IBM-No-Attririon data Set. Eigenvalues Inference',  
 plotKaiser = TRUE  
)



## Factor Maps for I for symmetrical plot for component 1 and 2

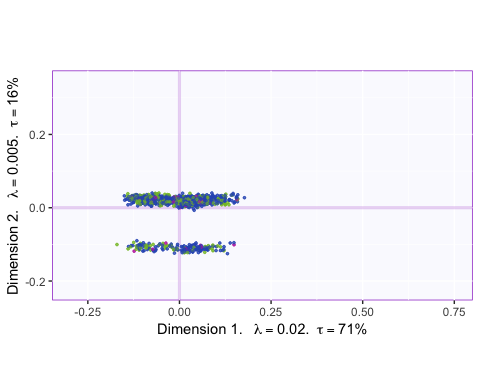
Symmetric plot deals with the average comparison

constraints.mca <- minmaxHelper(mat1 = resMCA.sym$ExPosition.Data$fi, mat2 = resMCA.sym$ExPosition.Data$fj)  
constraints.asym <- minmaxHelper(mat1 = resMCA.asym$ExPosition.Data$fi , mat2 = resMCA.asym$ExPosition.Data$fi)  
  
color <- prettyGraphsColorSelection(NCOL(data1))  
baseMap.i1 <- createFactorMap(resMCA.sym$ExPosition.Data$fi,constraints = constraints.mca,  
 col.points = resMCA.sym$Plotting.Data$fi.col, axis1 = 1, axis2 = 2,  
 cex = 1, pch = 20,  
 display.labels = FALSE  
)  
label4Map1 <- createxyLabels.gen(1,2,  
 lambda =resMCA.sym$ExPosition.Data$eigs,  
 tau = resMCA.sym$ExPosition.Data$t)  
  
a1 <- baseMap.i1$zeMap + baseMap.i1$zeMap\_dots +label4Map1  
print(a1)



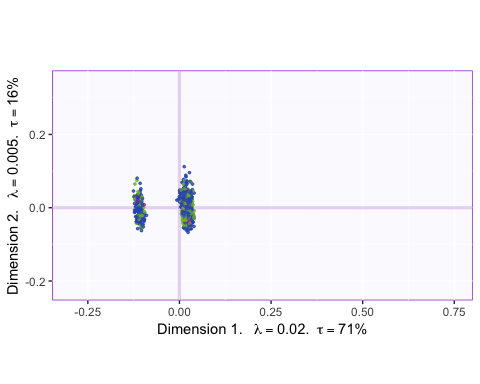
## Factor Maps for I for symmetrical plot for component 2 and 3

color <- prettyGraphsColorSelection(NCOL(data1))  
baseMap.i2 <- createFactorMap(resMCA.sym$ExPosition.Data$fi,constraints = constraints.mca,  
 col.points = resMCA.sym$Plotting.Data$fi.col, axis1 = 2, axis2 = 3,  
 cex = 1, pch = 20,  
 display.labels = FALSE  
)  
label4Map2 <- createxyLabels.gen(1,2,  
 lambda =resMCA.sym$ExPosition.Data$eigs,  
 tau = resMCA.sym$ExPosition.Data$t)  
  
a2 <- baseMap.i2$zeMap + baseMap.i2$zeMap\_dots +label4Map2  
print(a2)



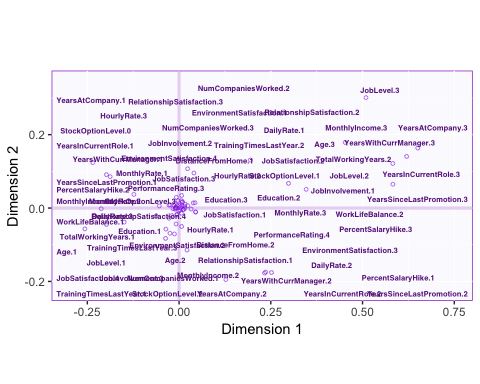
## Factor Maps for I for symmetrical plot for component 3 and 4

color <- prettyGraphsColorSelection(NCOL(data1))  
baseMap.i3 <- createFactorMap(resMCA.sym$ExPosition.Data$fi,constraints = constraints.mca,  
 col.points = resMCA.sym$Plotting.Data$fi.col, axis1 = 3, axis2 = 4,  
 cex = 1, pch = 20,  
 display.labels = FALSE  
)  
label4Map3 <- createxyLabels.gen(1,2,  
 lambda =resMCA.sym$ExPosition.Data$eigs,  
 tau = resMCA.sym$ExPosition.Data$t)  
  
a3 <- baseMap.i3$zeMap + baseMap.i3$zeMap\_dots +label4Map3  
print(a3)



## Factor Maps for J for symmetrical plot for component 1

baseMap.j1 <- createFactorMap(resMCA.sym$ExPosition.Data$fj, constraints = constraints.mca,  
 color.points = "black",text.cex = 2,cex = 1,pch = 21, axis1 = 1, axis2 = 2)  
  
print(baseMap.j1$zeMap)



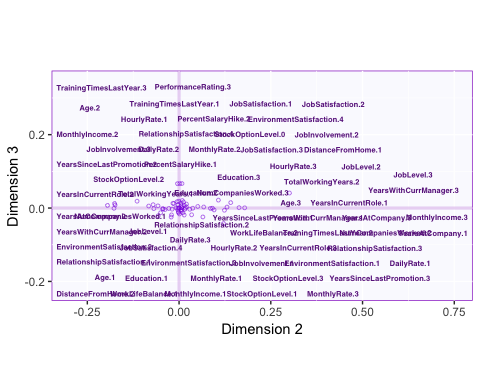
dev.new()  
print(baseMap.i1$zeMap + baseMap.j1$zeMap\_dots + baseMap.j1$zeMap\_text)

## Factor Maps for J for symmetrical plot for component 2

baseMap.j2 <- createFactorMap(resMCA.sym$ExPosition.Data$fj, constraints = constraints.mca,  
 color.points = "black",text.cex = 2,cex = 1,pch = 21, axis1 = 2, axis2 = 3)  
  
print(baseMap.j2$zeMap)

## Warning: Removed 2 rows containing missing values (geom\_point).

## Warning: Removed 2 rows containing missing values (geom\_text\_repel).



dev.new()  
print(baseMap.i2$zeMap + baseMap.j2$zeMap\_dots + baseMap.j2$zeMap\_text)

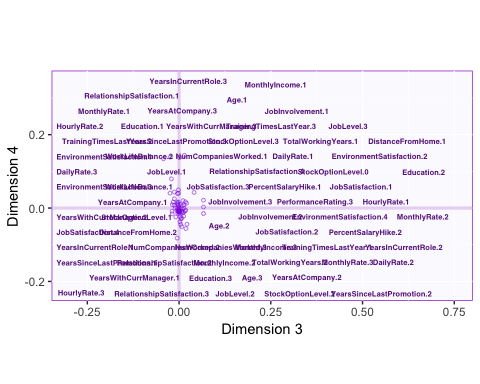
## Warning: Removed 2 rows containing missing values (geom\_point).  
  
## Warning: Removed 2 rows containing missing values (geom\_text\_repel).

## Factor Maps for J for symmetrical plot for component 3

baseMap.j3 <- createFactorMap(resMCA.sym$ExPosition.Data$fj, constraints = constraints.mca,  
 color.points = "black",text.cex = 2,cex = 1,pch = 21, axis1 = 3,axis2 = 4)  
  
print(baseMap.j3$zeMap)

## Warning: Removed 2 rows containing missing values (geom\_point).

## Warning: Removed 2 rows containing missing values (geom\_text\_repel).

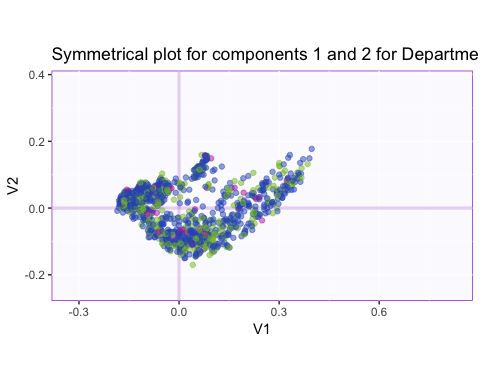


dev.new()  
print(baseMap.i3$zeMap + baseMap.j3$zeMap\_dots + baseMap.j3$zeMap\_text)

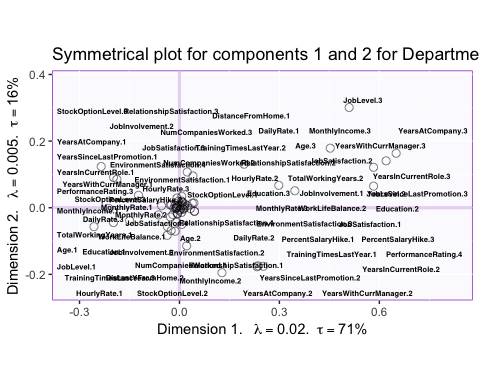
## Warning: Removed 2 rows containing missing values (geom\_point).  
  
## Warning: Removed 2 rows containing missing values (geom\_text\_repel).

## Biplot for symmetrical plot for component 1 and 2 for Department datatype

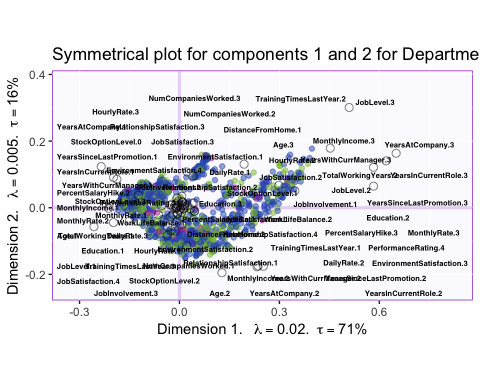
col4J1 <- prettyGraphsColorSelection(NCOL(data1))  
symMap1 <- createFactorMapIJ(resMCA.sym$ExPosition.Data$fi,resMCA.sym$ExPosition.Data$fj,  
 col.points.i = resMCA.sym$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 2.5, pch.i = 20,   
 pch.j = 21, cex.j = 2.5,text.cex.j =2, axis1 = 1,axis2 = 2, title = "Symmetrical plot for components 1 and 2 for Department Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
  
labels4MCA1 <- createxyLabels(resCA = resMCA.sym, x\_axis = 1,y\_axis = 2)  
  
map.IJ.sym1 <- symMap1$baseMap + symMap1$I\_points   
map.IJ.sym2 <- symMap1$baseMap + symMap1$J\_labels + symMap1$J\_points + labels4MCA1  
map.IJ.sym3 <- symMap1$baseMap + symMap1$I\_points +  
 symMap1$J\_labels + symMap1$J\_points + labels4MCA1  
print(map.IJ.sym1)



print(map.IJ.sym2)

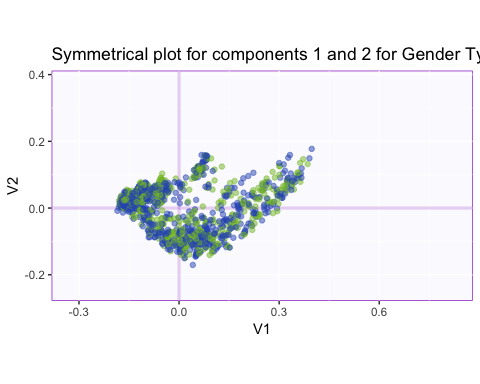


print(map.IJ.sym3)

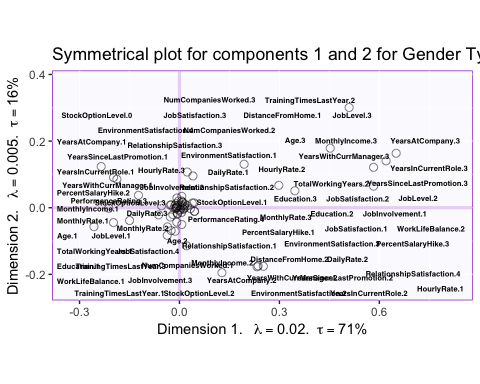


## Biplot for symmetrical plot for component 1 and 2 for Gender datatype

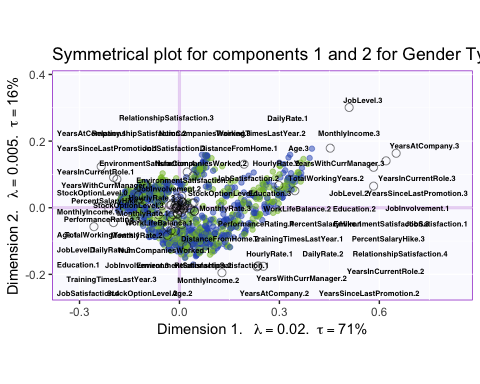
col4J1 <- prettyGraphsColorSelection(NCOL(data1))  
symMap1 <- createFactorMapIJ(resMCA.sym1$ExPosition.Data$fi,resMCA.sym1$ExPosition.Data$fj,  
 col.points.i = resMCA.sym1$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym1$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 2.5, pch.i = 20,   
 pch.j = 21, cex.j = 2.5,text.cex.j =2, axis1 = 1,axis2 = 2, title = "Symmetrical plot for components 1 and 2 for Gender Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
  
labels4MCA1 <- createxyLabels(resCA = resMCA.sym1, x\_axis = 1,y\_axis = 2)  
  
map.IJ.sym1 <- symMap1$baseMap + symMap1$I\_points   
map.IJ.sym2 <- symMap1$baseMap + symMap1$J\_labels + symMap1$J\_points + labels4MCA1  
map.IJ.sym3 <- symMap1$baseMap + symMap1$I\_points +  
 symMap1$J\_labels + symMap1$J\_points + labels4MCA1  
print(map.IJ.sym1)



print(map.IJ.sym2)

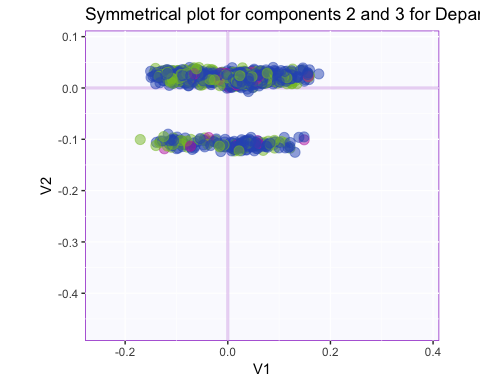


print(map.IJ.sym3)

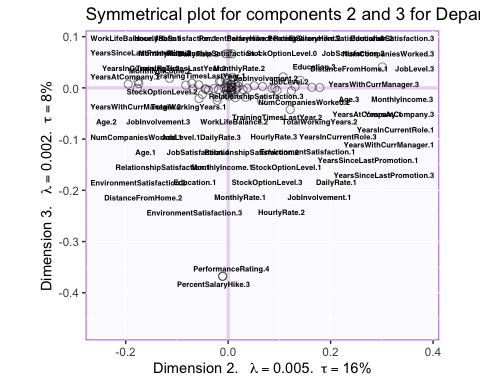


## Biplot for symmetrical plot for component 2 and 3 for Department datatype

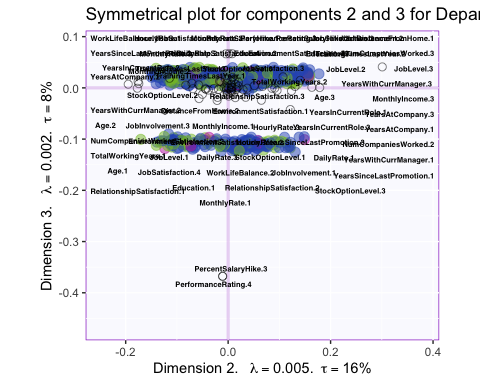
symMap2 <- createFactorMapIJ(resMCA.sym$ExPosition.Data$fi,resMCA.sym$ExPosition.Data$fj,  
 col.points.i = resMCA.sym$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 2,axis2 = 3, title = "Symmetrical plot for components 2 and 3 for Department Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
labels4MCA2 <- createxyLabels(resCA = resMCA.sym, x\_axis = 2, y\_axis = 3)  
  
map.IJ.sym11 <- symMap2$baseMap + symMap2$I\_points   
map.IJ.sym21 <- symMap2$baseMap + symMap2$J\_labels + symMap2$J\_points + labels4MCA2  
map.IJ.sym31 <- symMap2$baseMap + symMap2$I\_points +  
 symMap2$J\_labels + symMap2$J\_points + labels4MCA2  
print(map.IJ.sym11)



print(map.IJ.sym21)

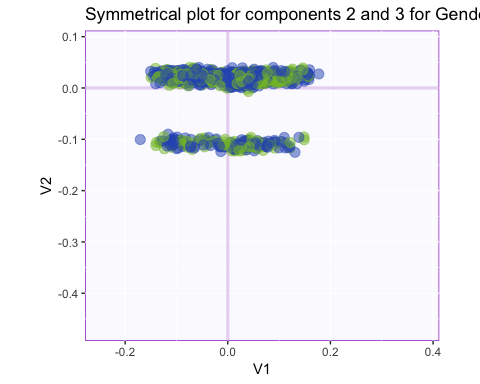


print(map.IJ.sym31)

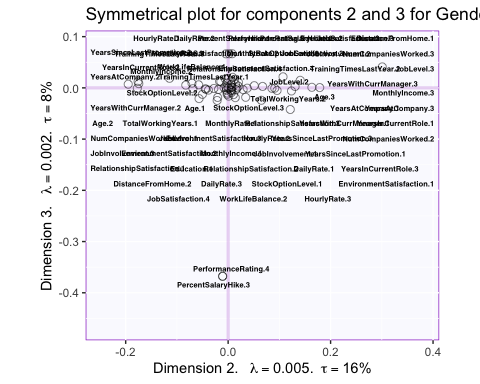


## Biplot for symmetrical plot for component 2 and 3 for Gender datatype

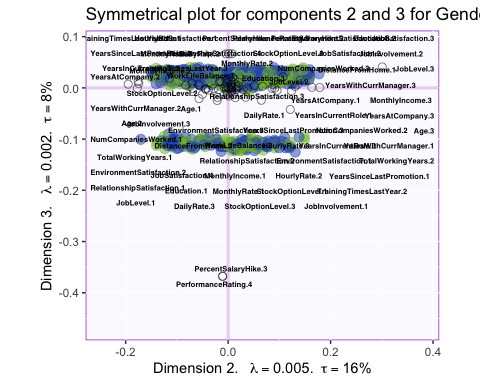
symMap2 <- createFactorMapIJ(resMCA.sym1$ExPosition.Data$fi,resMCA.sym1$ExPosition.Data$fj,  
 col.points.i = resMCA.sym1$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym1$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 2,axis2 = 3, title = "Symmetrical plot for components 2 and 3 for Gender Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
labels4MCA2 <- createxyLabels(resCA = resMCA.sym1, x\_axis = 2, y\_axis = 3)  
  
map.IJ.sym11 <- symMap2$baseMap + symMap2$I\_points   
map.IJ.sym21 <- symMap2$baseMap + symMap2$J\_labels + symMap2$J\_points + labels4MCA2  
map.IJ.sym31 <- symMap2$baseMap + symMap2$I\_points +  
 symMap2$J\_labels + symMap2$J\_points + labels4MCA2  
print(map.IJ.sym11)



print(map.IJ.sym21)

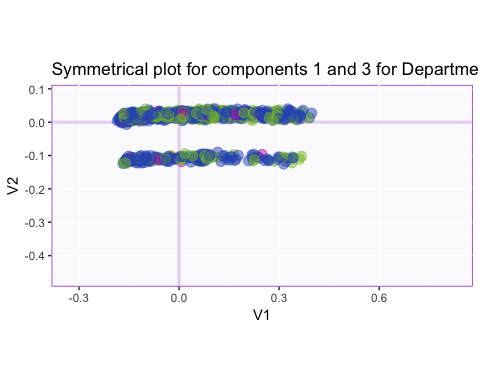


print(map.IJ.sym31)

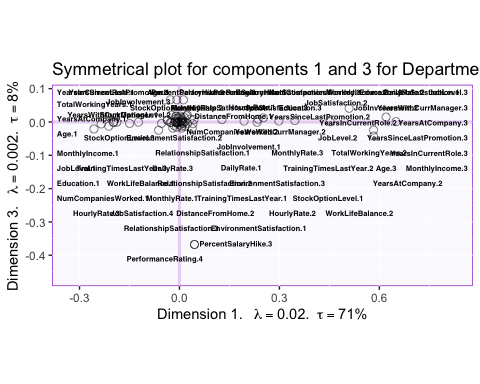


## Biplot for symmetrical plot for component 1 and 3 for Department datatype

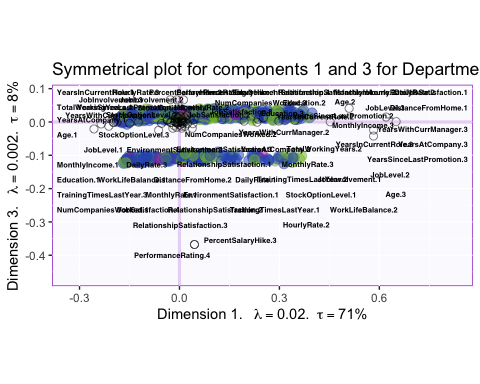
symMap3 <- createFactorMapIJ(resMCA.sym$ExPosition.Data$fi,resMCA.sym$ExPosition.Data$fj,  
 col.points.i = resMCA.sym$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 1,axis2 = 3, title = "Symmetrical plot for components 1 and 3 for Department Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
labels4MCA3 <- createxyLabels(resCA = resMCA.sym, x\_axis = 1,y\_axis = 3)  
  
map.IJ.sym31 <- symMap3$baseMap + symMap3$I\_points   
map.IJ.sym32 <- symMap3$baseMap + symMap3$J\_labels + symMap3$J\_points + labels4MCA3  
map.IJ.sym33 <- symMap3$baseMap + symMap3$I\_points +  
 symMap3$J\_labels + symMap3$J\_points + labels4MCA3  
print(map.IJ.sym31)



print(map.IJ.sym32)

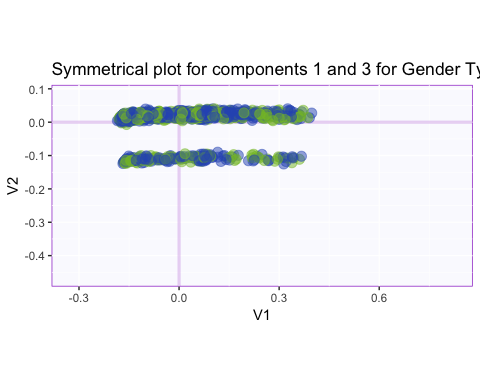


print(map.IJ.sym33)

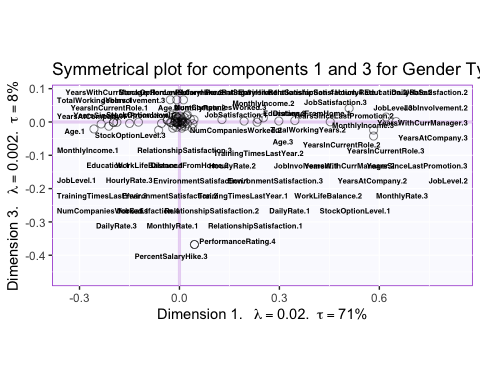


## Biplot for symmetrical plot for component 1 and 3 for Gender datatype

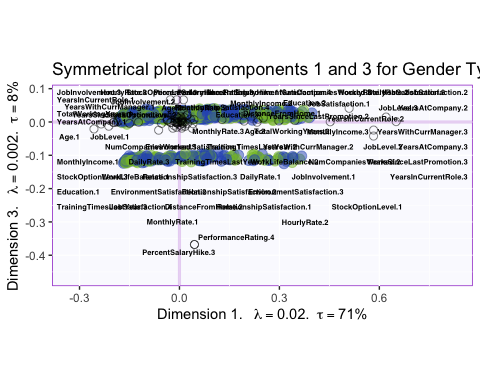
symMap3 <- createFactorMapIJ(resMCA.sym1$ExPosition.Data$fi,resMCA.sym1$ExPosition.Data$fj,  
 col.points.i = resMCA.sym1$Plotting.Data$fi.col,  
 col.points.j = "black",  
 col.labels.i = resMCA.sym1$Plotting.Data$fi.col ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 1,axis2 = 3, title = "Symmetrical plot for components 1 and 3 for Gender Type",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
labels4MCA3 <- createxyLabels(resCA = resMCA.sym1, x\_axis = 1,y\_axis = 3)  
  
map.IJ.sym31 <- symMap3$baseMap + symMap3$I\_points   
map.IJ.sym32 <- symMap3$baseMap + symMap3$J\_labels + symMap3$J\_points + labels4MCA3  
map.IJ.sym33 <- symMap3$baseMap + symMap3$I\_points +  
 symMap3$J\_labels + symMap3$J\_points + labels4MCA3  
print(map.IJ.sym31)



print(map.IJ.sym32)

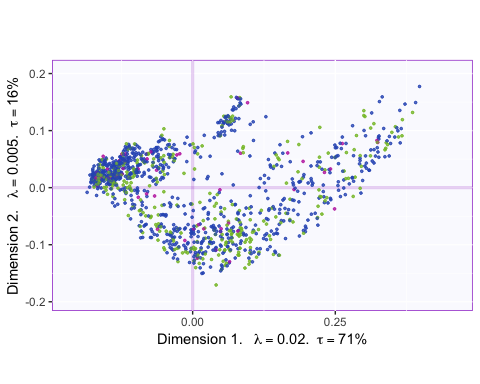


print(map.IJ.sym33)



## Factor Maps for I for asymmetrical plot

color <- prettyGraphsColorSelection(NCOL(data1))  
baseMap.i <- createFactorMap(resMCA.asym$ExPosition.Data$fi,constraints = constraints.asym,  
 col.points = resMCA.asym$Plotting.Data$fi.col,  
 cex = 1, pch = 20,  
 display.labels = FALSE  
)  
label4Map <- createxyLabels.gen(1,2,  
 lambda =resMCA.asym$ExPosition.Data$eigs,  
 tau = resMCA.asym$ExPosition.Data$t)  
  
a <- baseMap.i$zeMap + baseMap.i$zeMap\_dots +label4Map  
print(a)

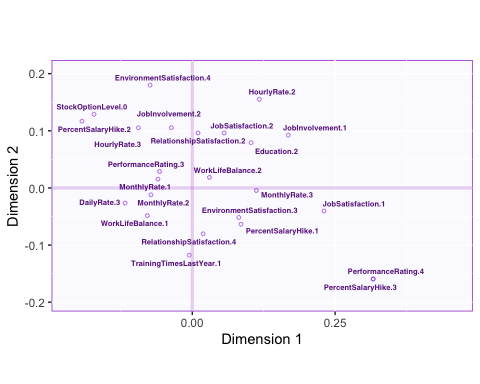


## Factor Maps for J for asymmetrical plot

baseMap.j <- createFactorMap(resMCA.asym$ExPosition.Data$fj, constraints = constraints.asym,  
 color.points = "black",text.cex = 2,cex = 1,pch = 21)  
  
print(baseMap.j$zeMap)

## Warning: Removed 45 rows containing missing values (geom\_point).

## Warning: Removed 45 rows containing missing values (geom\_text\_repel).



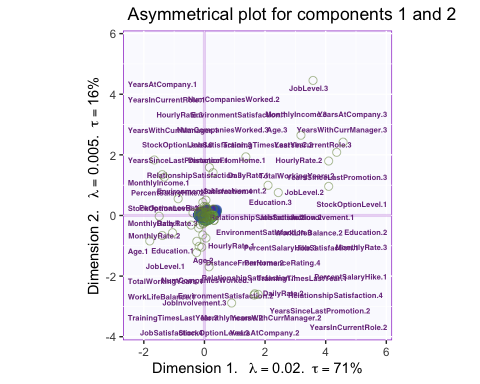
dev.new()  
print(baseMap.i$zeMap + baseMap.j$zeMap\_dots + baseMap.j$zeMap\_text)

## Warning: Removed 45 rows containing missing values (geom\_point).  
  
## Warning: Removed 45 rows containing missing values (geom\_text\_repel).

## Biplot for asymmetrical plot for component 1

Asymmetric plot deals with the attraction by the variables

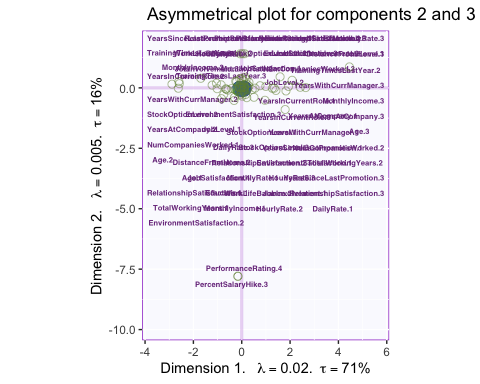
asymMap1 <- createFactorMapIJ(resMCA.asym$ExPosition.Data$fi,resMCA.asym$ExPosition.Data$fj,  
 col.points.i = resMCA.asym$Plotting.Data$fi.col,  
 col.labels.j = resMCA.asym$Plotting.Data$fj.col,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2,   
 alpha.axes = 0.2,alpha.points.i = 1, axis1 = 1, axis2 = 2, title = " Asymmetrical plot for components 1 and 2")  
  
alabels4MCA1 <- createxyLabels(resCA = resMCA.asym)  
  
map.IJ.asym1 <- asymMap1$baseMap + asymMap1$I\_points +  
 asymMap1$J\_labels + asymMap1$J\_points + alabels4MCA1  
print(map.IJ.asym1)



## Biplot for asymmetrical plot for component 2

Asymmetric plot deals with the attraction by the variables

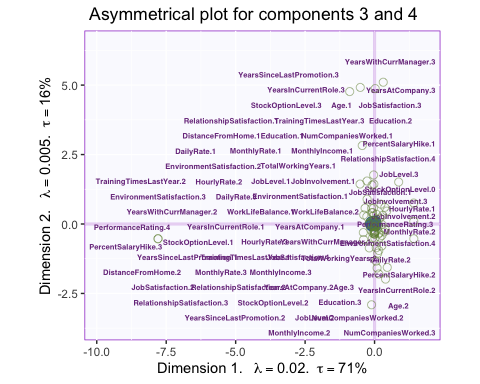
asymMap2 <- createFactorMapIJ(resMCA.asym$ExPosition.Data$fi,resMCA.asym$ExPosition.Data$fj,  
 col.points.i = resMCA.asym$Plotting.Data$fi.col,  
 col.labels.j = resMCA.asym$Plotting.Data$fj.col,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2,   
 alpha.axes = 0.2,alpha.points.i = 1, axis1 = 2, axis2 = 3, title = " Asymmetrical plot for components 2 and 3")  
  
alabels4MCA2 <- createxyLabels(resCA = resMCA.asym)  
  
map.IJ.asym2 <- asymMap2$baseMap + asymMap2$I\_points +  
 asymMap2$J\_labels + asymMap2$J\_points + alabels4MCA2  
print(map.IJ.asym2)



## Biplot for asymmetrical plot for component 3

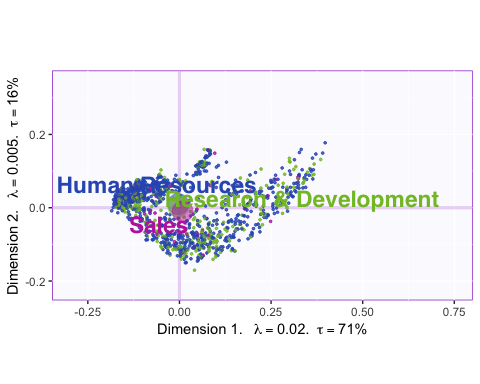
Asymmetric plot deals with the attraction by the variables

asymMap3 <- createFactorMapIJ(resMCA.asym$ExPosition.Data$fi,resMCA.asym$ExPosition.Data$fj,  
 col.points.i = resMCA.asym$Plotting.Data$fi.col,  
 col.labels.j = resMCA.asym$Plotting.Data$fj.col,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2,   
 alpha.axes = 0.2,alpha.points.i = 1, axis1 = 3, axis2 = 4, title = " Asymmetrical plot for components 3 and 4")  
  
alabels4MCA3 <- createxyLabels(resCA = resMCA.asym)  
  
map.IJ.asym3 <- asymMap3$baseMap + asymMap3$I\_points +  
 asymMap3$J\_labels + asymMap3$J\_points + alabels4MCA3  
print(map.IJ.asym3)



## Bootstrap Interval

BootCube.Gr <- Boot4Mean(resMCA.asym$ExPosition.Data$fi,   
 design = my\_data$Department,  
 niter = 100,  
 suppressProgressBar = TRUE)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Bootstrap ratios ----  
bootRatios.Gr <- boot.ratio.test(BootCube.Gr$BootCube)  
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Mean Map  
# create the map for the means  
# get the means by groups  
  
dataMeans <- getMeans(resMCA.asym$ExPosition.Data$fi, my\_data$Department)  
# a vector of color for the means  
col4data <- resMCA.asym$Plotting.Data$fi.col  
col4Means <- unique(col4data)  
# the map  
MapGroup <- createFactorMap(dataMeans,  
 # use the constraint from the main map  
 constraints = constraints.mca,  
 col.points = col4Means,  
 cex = 7, # size of the dot (bigger)  
 col.labels = col4Means,  
 text.cex = 6)  
# The map with observations and group means  
a003.Map.I.withMeans <- a1 +  
 MapGroup$zeMap\_dots + MapGroup$zeMap\_text  
print(a003.Map.I.withMeans)



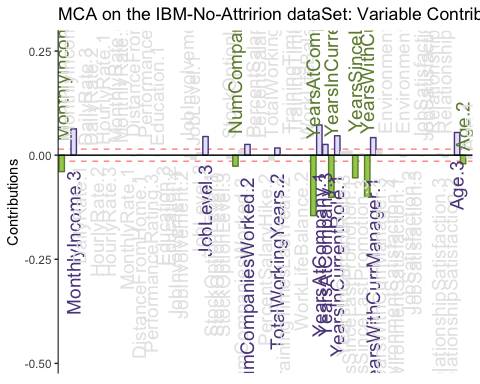
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Create the ellipses  
# Bootstrapped CI ----  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Create Confidence Interval Plots  
# use function MakeCIEllipses from package PTCA4CATA  
GraphElli <- MakeCIEllipses(BootCube.Gr$BootCube[,1:2,],  
 names.of.factors = c("Dimension 1","Dimension 2"),  
 col = col4Means,  
 p.level = .95  
)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# create the I-map with Observations, means and confidence intervals  
#  
a004.Map.I.withCI <- a1 + MapGroup$zeMap\_text + GraphElli  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# plot it!  
dev.new()  
print(a004.Map.I.withCI)

## Contribution for variables

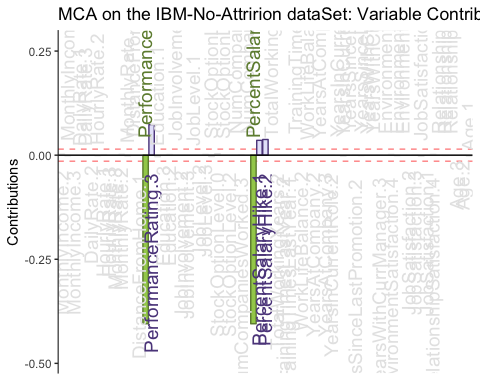
signed.ctrJ <- resMCA.sym$ExPosition.Data$cj \* sign(resMCA.sym$ExPosition.Data$fj)  
b003.ctrJ.s.1 <- PrettyBarPlot2(signed.ctrJ[,1],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon data Set: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ), 1.2\*max(signed.ctrJ))  
)  
print(b003.ctrJ.s.1)



b004.ctrJ.s.2 <- PrettyBarPlot2(signed.ctrJ[,2],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ), 1.2\*max(signed.ctrJ))  
)  
print(b004.ctrJ.s.2)

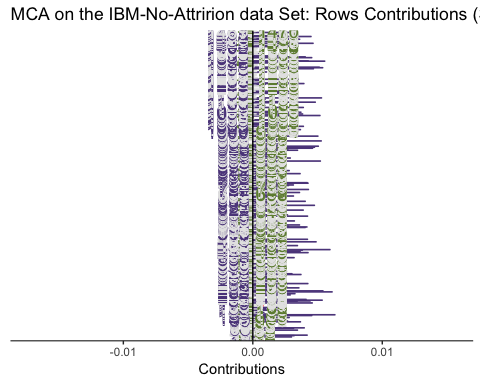


b004.ctrJ.s.3 <- PrettyBarPlot2(signed.ctrJ[,3],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ), 1.2\*max(signed.ctrJ))  
)  
print(b004.ctrJ.s.3)

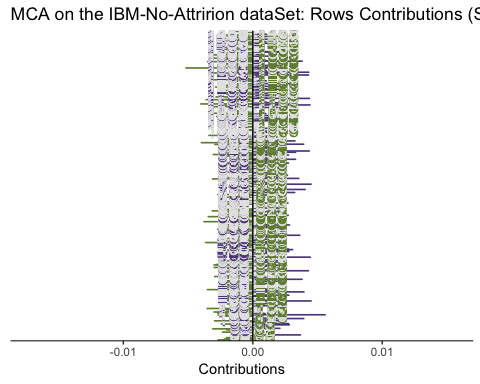


## Contribution for Rows

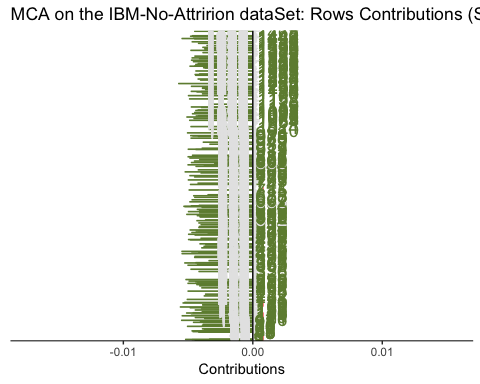
signed.ctri <- resMCA.sym$ExPosition.Data$ci \* sign(resMCA.sym$ExPosition.Data$fi)  
b003.ctri.s.1 <- PrettyBarPlot2(signed.ctri[,1],  
 threshold = 1 / NROW(signed.ctri),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon data Set: Rows Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctri), 1.2\*max(signed.ctri)), horizontal = FALSE  
)  
print(b003.ctri.s.1)



b004.ctri.s.2 <- PrettyBarPlot2(signed.ctri[,2],  
 threshold = 1 / NROW(signed.ctri),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon dataSet: Rows Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctri), 1.2\*max(signed.ctri)) , horizontal = FALSE  
)  
print(b004.ctri.s.2)

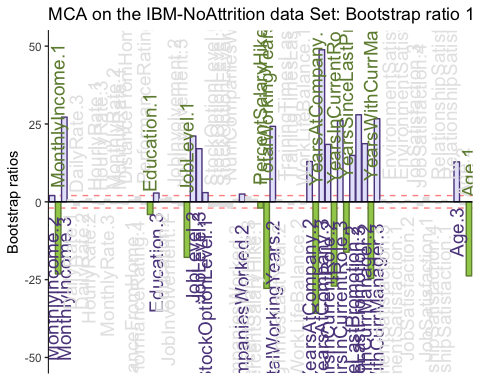


b004.ctri.s.3 <- PrettyBarPlot2(signed.ctri[,3],  
 threshold = 1 / NROW(signed.ctri),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'MCA on the IBM-No-Attririon dataSet: Rows Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctri), 1.2\*max(signed.ctri)),horizontal = FALSE  
)  
print(b004.ctri.s.3)

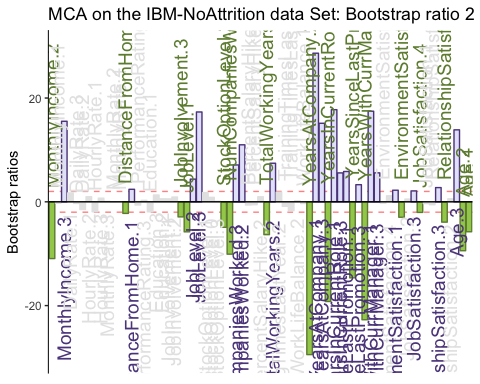


## Bootstrap Ratios for Variables

BR <- resMCA.inf$Inference.Data$fj.boots$tests$boot.ratios  
laDim = 1  
ba001.BR1 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0( 'MCA on the IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
)  
print(ba001.BR1)



#  
laDim = 2  
ba002.BR2 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0(  
 'MCA on the IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR2)



laDim = 3  
ba002.BR3 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 main = paste0(  
 'MCA on the IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR3)



## Summary

Questions to ask : Comparison between PCA nad MCA Asymmetrical plot factor map for i and j and biplot