PLS

Richa Singh

10/25/2018

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

## PLS

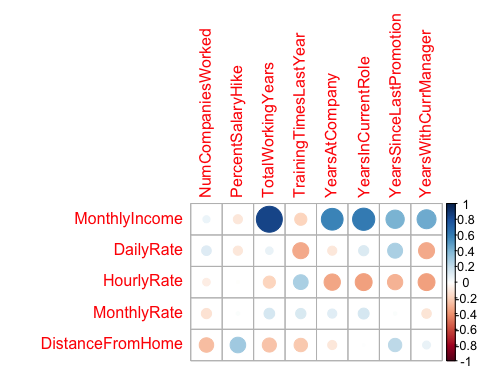
## Dataset : IBM-HR-Employee-NoAttrition

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.

data <- read.csv("IBM-HR-Emplyee-NoAttrition.csv")  
data1 <- data[,11:15]   
data2 <- data[,21:29]   
data2 <- data2[ -c(5) ]  
data$Gender <- as.numeric(as.factor(data$Gender))  
data$Department <- as.numeric(as.factor(data$Department))  
design1 <- data$Gender  
design2 <- data$Department

nN <- 30  
data\_1 <- data1[1:nN,]  
data\_2 <- data2[1:nN,]  
Design1 <- design1[1:nN]  
Design2 <- design2[1:nN]

corrplot::corrplot(cor(data\_1,data\_2))



## Scree Plot

A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each PC.(In the PCA literature, the plot is called a 'Scree' Plot because it often looks like a 'scree' slope, where rocks have fallen down and accumulated on the side of a mountain.)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.

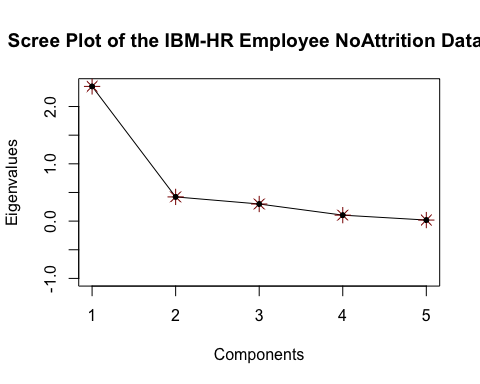
library(TExPosition)

## Loading required package: prettyGraphs

## Loading required package: ExPosition

resPLSC <- tepPLS(data\_1,data\_2,DESIGN = Design1,graphs = FALSE)

name\_the\_scree <- plot(resPLSC$TExPosition.Data$eigs,  
 ylab = "Eigenvalues",  
 xlab = "Components",  
 type = "l",  
 main = "Scree Plot of the IBM-HR Employee NoAttrition Dataset",  
 ylim = c(-1, max(resPLSC$TExPosition.Data$eigs)))  
points(resPLSC$TExPosition.Data$eigs, cex = 1.5, pch = 8, col = "dark red")  
points(resPLSC$TExPosition.Data$eigs, cex = 1, pch = 20, col = "black")



## Factor Maps for I

library(data4PCCAR)  
library(PTCA4CATA)  
library(corrplot)

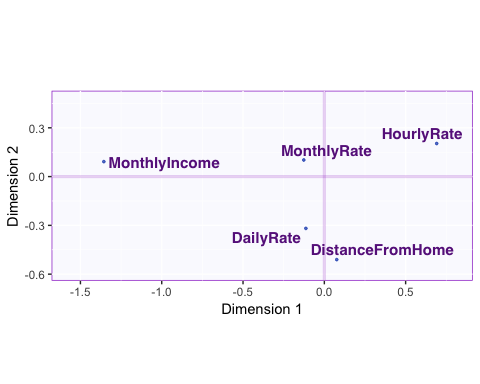
## corrplot 0.84 loaded

library(ggplot2)  
library(ExPosition)  
library(InPosition)

##   
## Attaching package: 'InPosition'

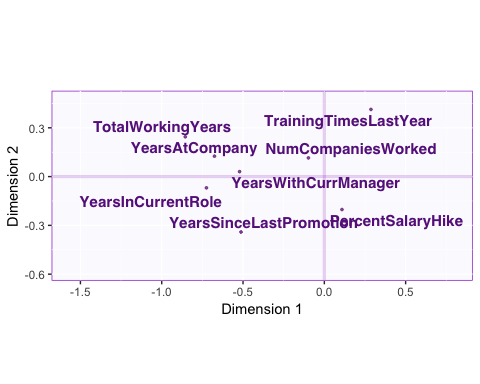
## The following object is masked from 'package:PTCA4CATA':  
##   
## boot.ratio.test

constraints <- minmaxHelper(mat1 = resPLSC$TExPosition.Data$fi, mat2 = resPLSC$TExPosition.Data$fj)  
baseMap.i3 <- createFactorMap(resPLSC$TExPosition.Data$fi,constraints = constraints,  
 col.points = resPLSC$Plotting.Data$fi.col, axis1 = 1, axis2 = 2,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a3 <- baseMap.i3$zeMap + baseMap.i3$zeMap\_dots   
print(a3)



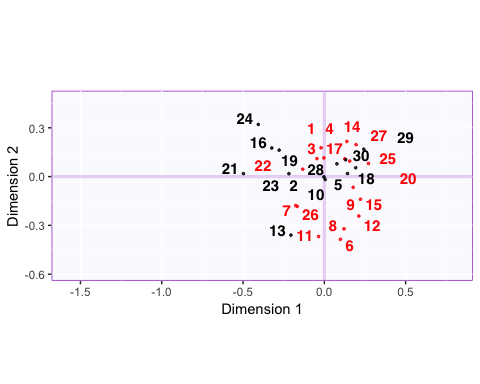
## Factor Maps for J

baseMap.j3 <- createFactorMap(resPLSC$TExPosition.Data$fj,constraints = constraints,  
 col.points = resPLSC$Plotting.Data$fj.col, axis1 = 1, axis2 = 2,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a1 <- baseMap.j3$zeMap + baseMap.j3$zeMap\_dots   
print(a1)



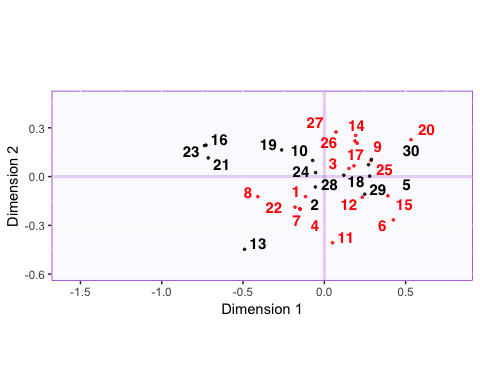
## Factor Map for latent variables lx for Gendertype

baseMap.i <- createFactorMap(resPLSC$TExPosition.Data$lx,constraints = resPLSC$Plotting.Data$constraints,  
 col.points = Design1,  
 col.labels = Design1,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a <- baseMap.i$zeMap + baseMap.i$zeMap\_dots   
print(a)



## Factor Map for latent variables ly for Gendertype

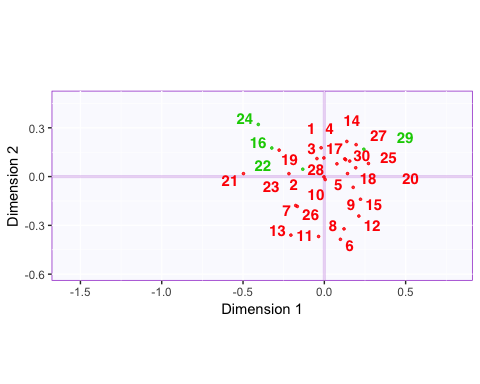
baseMap.i <- createFactorMap(resPLSC$TExPosition.Data$ly,constraints = resPLSC$Plotting.Data$constraints,  
 col.points = Design1,  
 col.labels = Design1,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a <- baseMap.i$zeMap + baseMap.i$zeMap\_dots   
print(a)



## Factor Map for latent variables lx for Departmenttype

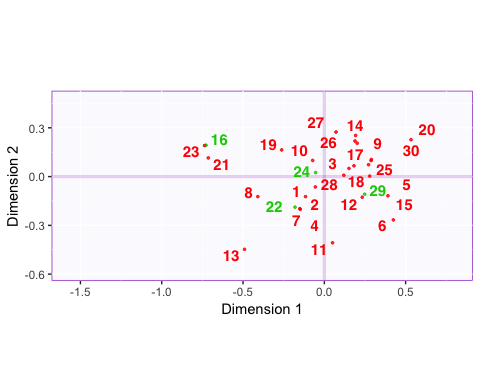
resPLSC1 <- tepPLS(data\_1,data\_2,DESIGN = Design2,graphs = FALSE)

library(PTCA4CATA)  
baseMap.i <- createFactorMap(resPLSC1$TExPosition.Data$lx,constraints = resPLSC1$Plotting.Data$constraints,  
 col.points = Design2,  
 col.labels = Design2,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a <- baseMap.i$zeMap + baseMap.i$zeMap\_dots   
print(a)



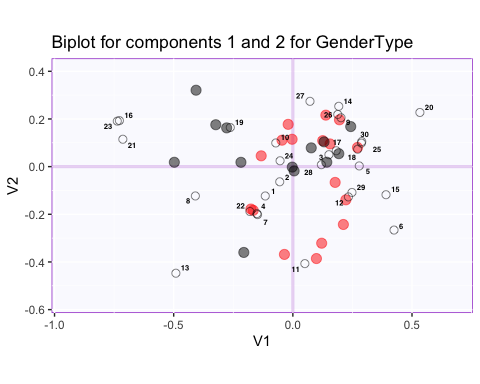
## Factor Map for latent variables ly for Departmenttype

baseMap.i <- createFactorMap(resPLSC1$TExPosition.Data$ly,constraints = resPLSC1$Plotting.Data$constraints,  
 col.points = Design2,  
 col.labels = Design2,  
 cex = 1, pch = 20,  
 display.labels = TRUE  
)  
a <- baseMap.i$zeMap + baseMap.i$zeMap\_dots   
print(a)



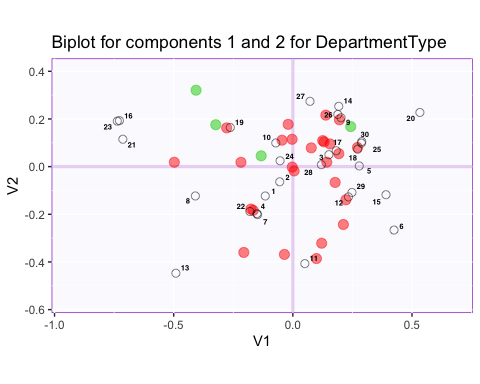
## Biplot for latent Variables for Gender type

symMap <- createFactorMapIJ(resPLSC$TExPosition.Data$lx[,c(1,2)],resPLSC$TExPosition.Data$ly[,c(1,2)],  
 col.points.i = Design1,  
 col.points.j = "black",  
 col.labels.i = Design1 ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 1,axis2 = 2, title = "Biplot for components 1 and 2 for GenderType",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
map.IJ.sym <- symMap$baseMap + symMap$I\_points +  
 symMap$J\_labels + symMap$J\_points   
print(map.IJ.sym)



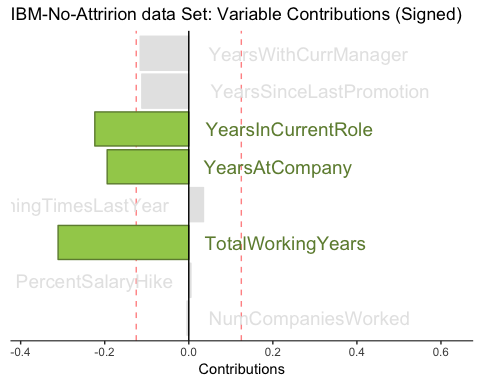
## Biplot for latent Variables for Department type

symMap1 <- createFactorMapIJ(resPLSC1$TExPosition.Data$lx[,c(1,2)],resPLSC1$TExPosition.Data$ly[,c(1,2)],  
 col.points.i = Design2,  
 col.points.j = "black",  
 col.labels.i = Design2 ,  
 col.labels.j = "black" ,  
 cex.i = 5, pch.i = 20,   
 pch.j = 21, text.cex.j =2, axis1 = 1,axis2 = 2, title = "Biplot for components 1 and 2 for DepartmentType",  
 alpha.axes = 0.2,alpha.points.i = 1)  
  
map.IJ1 <- symMap1$baseMap + symMap1$I\_points +  
 symMap1$J\_labels + symMap1$J\_points   
print(map.IJ1)

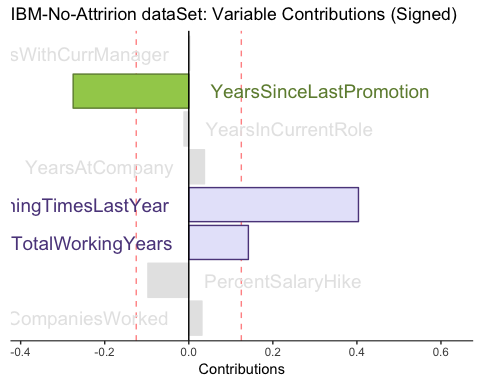


## Contributions for Variables

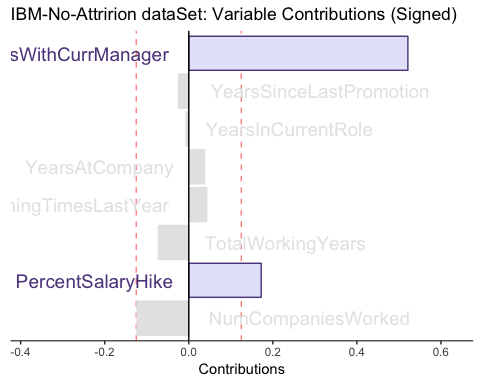
signed.ctrJ <- resPLSC$TExPosition.Data$cj \* sign(resPLSC$TExPosition.Data$fj)  
b003.ctrJ.s.1 <- PrettyBarPlot2(signed.ctrJ[,1],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'IBM-No-Attririon data Set: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ), 1.2\*max(signed.ctrJ)), horizontal = FALSE  
)  
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 = 'IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ), 1.2\*max(signed.ctrJ)), horizontal = FALSE  
)  
print(b004.ctrJ.s.2)

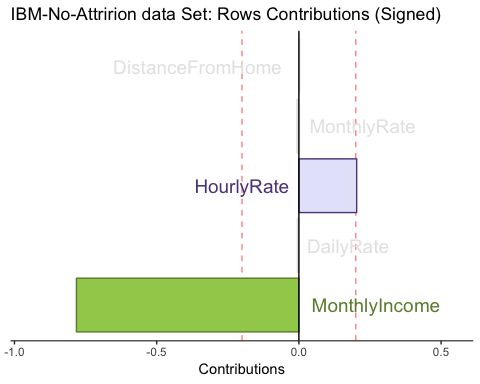


b004.ctrJ.s.3 <- PrettyBarPlot2(signed.ctrJ[,3],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ),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

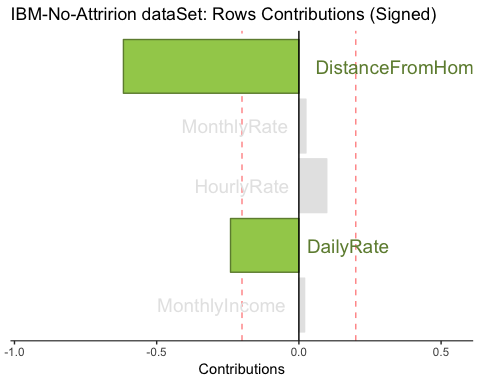


## Contribution for Rows

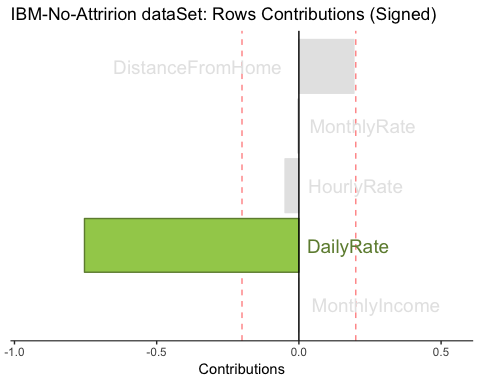
signed.ctri <- resPLSC$TExPosition.Data$ci \* sign(resPLSC$TExPosition.Data$fi)  
b003.ctri.s.1 <- PrettyBarPlot2(signed.ctri[,1],  
 threshold = 1 / NROW(signed.ctri),  
 font.size = 5,  
 # color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'IBM-No-Attririon data Set: Rows Contributions (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 = '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 = '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)



## Load the file

file4PLSCfunctions <- 'inferences4PLSC(4).R'  
source(file4PLSCfunctions)

## Permutation test

resPerm4PLSC <- perm4PLSC(data\_1, # First Data matrix   
 data\_2, # Second Data matrix  
 nIter = 1000 # How mny iterations  
 )  
# to see what results we have  
print(resPerm4PLSC)

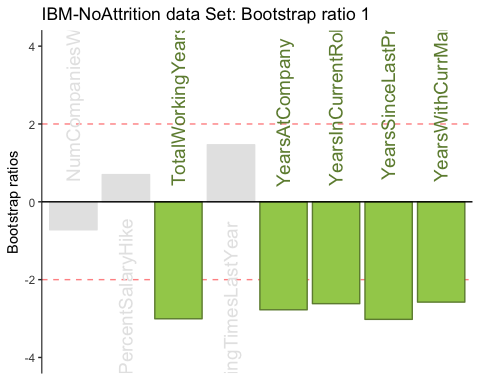
## ------------------------------------------------------------------------------  
## Results of Permutation Tests for CA of Matrix X   
## for Omnibus Inertia and Eigenvalues   
## ------------------------------------------------------------------------------  
## $ fixedInertia the Inertia of Matrix X  
## $ fixedEigenvalues a 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  
## ------------------------------------------------------------------------------

## Bootstrap Ratios for Variables

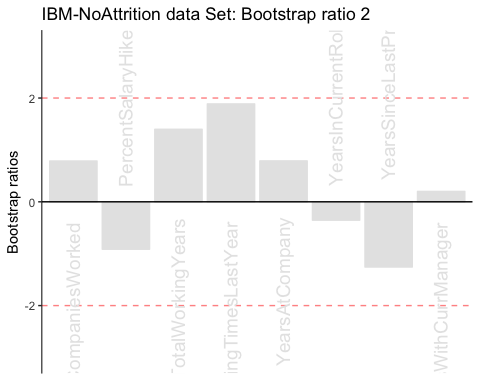
resBoot4PLSC <- Boot4PLSC(data\_1, # First Data matrix   
 data\_2, # Second Data matrix  
 nIter = 1000, # How many iterations  
 Fi = resPLSC$TExPosition.Data$fi,  
 Fj = resPLSC$TExPosition.Data$fj,  
 nf2keep = 3,  
 critical.value = 2,  
 # To be implemented later  
 # has no effect currently  
 alphaLevel = .05)  
print(resBoot4PLSC)

## ------------------------------------------------------------------------------  
## Bootstraped Factor Scores (BFS) and Bootstrap Ratios (BR)   
## for the I and J-sets of a CA (obtained from multinomial resampling of X)   
## ------------------------------------------------------------------------------  
## $ bootstrapBrick.i an I\*L\*nIter Brick of BFSs for the I-Set  
## $ bootRatios.i an I\*L matrix of BRs for the I-Set  
## $ bootRatiosSignificant.i an I\*L logical matrix for significance of the I-Set  
## $ bootstrapBrick.j a J\*L\*nIter Brick of BFSs for the J-Set  
## $ bootRatios.j a J\*L matrix of BRs for the J-Set  
## $ bootRatiosSignificant.j a J\*L logical matrix for significance of the J-Set  
## ------------------------------------------------------------------------------

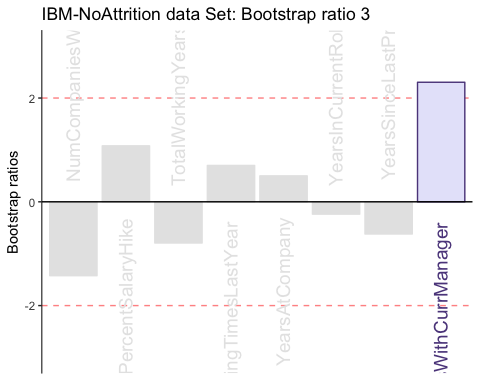
laDim = 1  
ba001.BR1 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.j[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0( 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
)  
print(ba001.BR1)



#  
laDim = 2  
ba002.BR2 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.j[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0(  
 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR2)

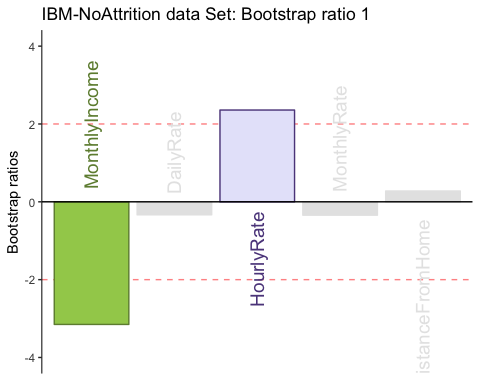


laDim = 3  
ba002.BR3 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.j[,laDim],  
 threshold = 2,  
 font.size = 5,  
 main = paste0(  
 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR3)

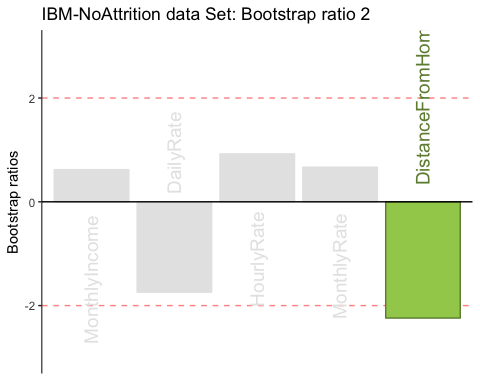


## Bootstrap Ratios for Rows

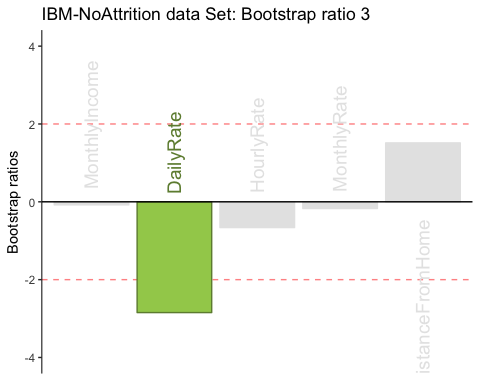
laDim = 1  
ba001.BR11 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.i[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0( 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
)  
print(ba001.BR11)



#  
laDim = 2  
ba002.BR21 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.i[,laDim],  
 threshold = 2,  
 font.size = 5,  
 #color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0(  
 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR21)



laDim = 3  
ba002.BR31 <- PrettyBarPlot2(resBoot4PLSC$bootRatios.i[,laDim],  
 threshold = 2,  
 font.size = 5,  
 main = paste0(  
 'IBM-NoAttrition data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR31)



## Summary