PCA Inference

Richa Singh

September 22, 2018

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

Principal Component Analysis is the analysis of data to identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information. Here, our desired outcome of the principal component analysis is to project a feature space (our dataset consisting of n d-dimensional samples) onto a smaller subspace that represents our data well. 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. Because of the Pythagorean Theorem, the total information contributed by a data point is also equal to the sum of its squared factor scores.

## 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.

my\_data <- read.csv("IBM-HR-Emplyee-NoAttrition.csv")  
rownames(my\_data) <- my\_data$Subj  
my\_data1 <- my\_data[sapply(my\_data, function(x) !is.factor(x))]  
my\_data1 <- subset(my\_data1, select=-Subj)

## ResearchQuestion

How do all 1233 employees differ on the variables like Gender type vs Department ? Job level vs MonthlyIncome ? etc.

## Results

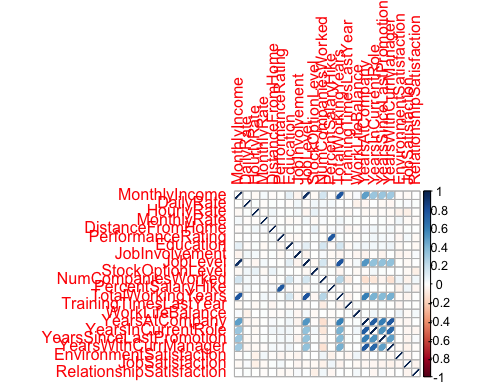
## Correlation Plot

Let’s analyze the correlation plot and find the prinicipal components.

datar <- my\_data1[,2:23]  
datae <- my\_data[,5]  
library(corrplot)

## corrplot 0.84 loaded

cor.my\_data <- cor(datar)  
corrplot(cor.my\_data, method = "ellipse")



DESIGN variables

#Create an empty list, called DESIGN  
library(ExPosition)

## Loading required package: prettyGraphs

DESIGN <- list()  
DESIGN$rows$Region$labels <- unique(my\_data$Department)   
DESIGN$rows$Region$vec <- my\_data$Department  
  
#Convert the vector to a matrix  
DESIGN$rows$Region$mat <- makeNominalData(as.matrix(DESIGN$rows$Region$vec))  
automatic\_colors <- createColorVectorsByDesign(DESIGN$rows$Region$mat)  
DESIGN$rows$Region$color\_groups <- c("pink","blue","green")  
DESIGN$rows$Region$color\_observ <- as.matrix(DESIGN$rows$Region$vec)   
  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="Research & Development")] <- DESIGN$rows$Region$color\_groups[1]  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="Human Resources")] <- DESIGN$rows$Region$color\_groups[2]  
DESIGN$rows$Region$color\_observ[which(DESIGN$rows$Region$vec=="Sales")] <- DESIGN$rows$Region$color\_groups[3]

Because each variable is measured on different units, I choose to center and scale the columns. The rows are color-coded by the DESIGN variable, state.division. \* center = TRUE: substracts the mean from each column \* scale = TRUE: after centering (or not), scales each column (see the help for different scaling options) \* DESIGN: colors the observations (rows)

library(InPosition)  
resPCA <- epPCA(DATA = datar,  
 scale = 'SS1', # Make to use 'SS1' rather than TRUE  
 DESIGN = datae,  
 graphs = FALSE # TRUE first pass only  
)  
# Varimax here ----  
testVari <- data4PCCAR::epVari(resPCA)  
# Inference battery ----  
resPCA.inf <- InPosition::epPCA.inference.battery(DATA = datar,  
 scale = 'SS1', # Make sure to use 'SS1' rather than TRUE  
 DESIGN = datae,  
 graphs = FALSE # TRUE first pass only  
)

## [1] "It is estimated that your iterations will take 0.08 minutes."  
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the progress bar."  
## ===========================================================================

## 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. Here, min(1233,31)-1 give 30 components. The scree plot is used to determine how many of the components should be interpreted.

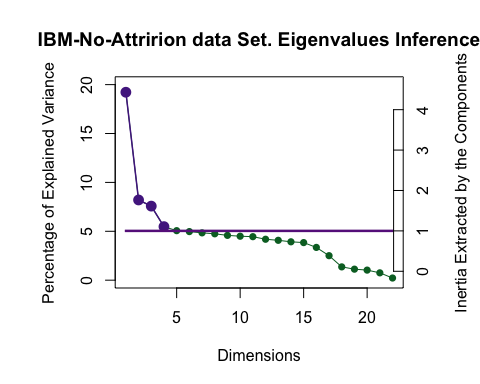
* plot draws the line that connects all data points by type = "l"
* The first points function draws round purple dots.
* The second points function draws black circles around the dots (just to make it prettier).

library(ExPosition)  
library(InPosition)  
library(PTCA4CATA)

##   
## Attaching package: 'PTCA4CATA'

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

library(corrplot)  
library(ggplot2)  
library(data4PCCAR)  
PlotScree(ev = resPCA$ExPosition.Data$eigs,   
 p.ev = resPCA.inf$Inference.Data$components$p.vals,  
 title = 'IBM-No-Attririon data Set. Eigenvalues Inference',  
 plotKaiser = TRUE  
)

 The top 4 compnents can be analyzed.

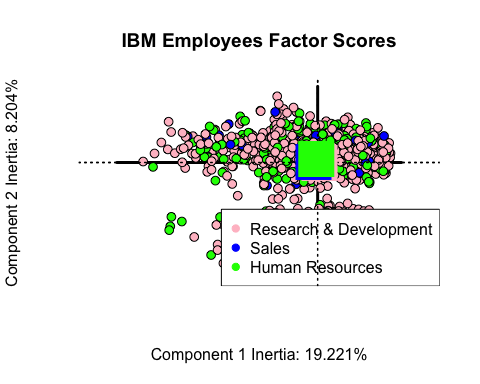
## Factor Scores for Department Variable

## Component 1 Vs Component 2

Factor scores are the coordinates of the 1233 rows of the employees on the components. The distances between them show which all employees are the most similar ones. Factor scores (employees) can be color-coded to help interpret the components.

* prettyPlot helps plot the factor scores. In order to print the result in an Rmd, dev.new needs to be FALSE.

#FACTOR SCORES  
OriginAsDesignMat <- makeNominalData(as.matrix(my\_data$Department))  
Fi2plot <- resPCA$ExPosition.Data$fi  
fi\_all.sum <- t(Fi2plot) %\*% OriginAsDesignMat  
fi\_all.count <- colSums(OriginAsDesignMat)  
fi\_all.mean <- t(fi\_all.sum)/fi\_all.count  
  
#pretty plot between component1 and 2   
name\_the\_plot <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 1 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN$rows$Region$labels, col=DESIGN$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean,   
 col = DESIGN$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)

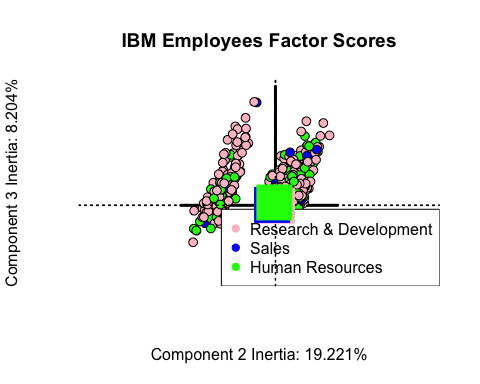


## $col  
## [,1]   
## [1,] "pink"   
## [2,] "blue"   
## [3,] "green"  
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
## [3,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.005971  
##   
## $constraints$miny  
## [1] -0.0006330043  
##   
## $constraints$maxx  
## [1] 0.002674303  
##   
## $constraints$maxy  
## [1] 0.005000752

* Component 1: May be Sales vs Human Resources
* Component 2: No significant inference could be observed

## Component 2 Vs Component 3

name\_the\_plot <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 2, y\_axis = 3,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 3 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN$rows$Region$labels, col=DESIGN$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean,   
 col = DESIGN$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)

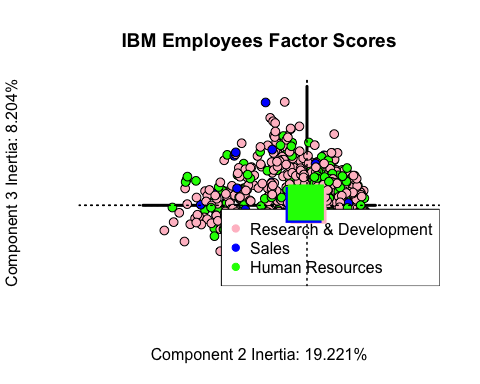


## $col  
## [,1]   
## [1,] "pink"   
## [2,] "blue"   
## [3,] "green"  
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
## [3,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.005971  
##   
## $constraints$miny  
## [1] -0.0006330043  
##   
## $constraints$maxx  
## [1] 0.002674303  
##   
## $constraints$maxy  
## [1] 0.005000752

* Component 2: Definitely Sales vs Human Resources
* Component 3: Maybe divides Sales & Human Rescources VS Research and Development

## Component 1 Vs Component 3

name\_the\_plot <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 1, y\_axis = 3,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 3 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN$rows$Region$labels, col=DESIGN$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean,   
 col = DESIGN$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)



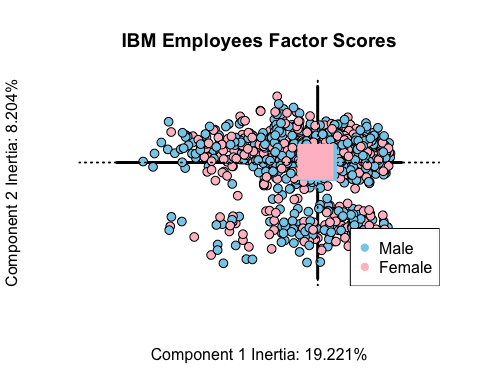
## $col  
## [,1]   
## [1,] "pink"   
## [2,] "blue"   
## [3,] "green"  
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
## [3,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.005971  
##   
## $constraints$miny  
## [1] -0.0006330043  
##   
## $constraints$maxx  
## [1] 0.002674303  
##   
## $constraints$maxy  
## [1] 0.005000752

No significant observation can be concluded both from Component 1 and Component 3

## Factor Scores for Gender Variable

## Component 1 Vs Component 2

#Create an empty list, called DESIGN  
library(ExPosition)  
DESIGN1 <- list()  
DESIGN1$rows$Region$labels <- unique(my\_data$Gender)   
DESIGN1$rows$Region$vec <- my\_data$Gender  
  
#Convert the vector to a matrix  
DESIGN1$rows$Region$mat <- makeNominalData(as.matrix(DESIGN1$rows$Region$vec))  
automatic\_colors <- createColorVectorsByDesign(DESIGN1$rows$Region$mat)  
DESIGN1$rows$Region$color\_groups <- c("skyblue","pink")  
DESIGN1$rows$Region$color\_observ <- as.matrix(DESIGN1$rows$Region$vec)   
  
DESIGN1$rows$Region$color\_observ[which(DESIGN1$rows$Region$vec=="Male")] <- DESIGN1$rows$Region$color\_groups[1]  
DESIGN1$rows$Region$color\_observ[which(DESIGN1$rows$Region$vec=="Female")] <- DESIGN1$rows$Region$color\_groups[2]  
OriginAsDesignMat1 <- makeNominalData(as.matrix(my\_data$Gender))  
Fi2plot1 <- resPCA$ExPosition.Data$fi  
fi\_all.sum1 <- t(Fi2plot1) %\*% OriginAsDesignMat1  
fi\_all.count1 <- colSums(OriginAsDesignMat1)  
fi\_all.mean1 <- t(fi\_all.sum1)/fi\_all.count1  
  
#pretty plot between component1 and 2   
name\_the\_plot1 <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN1$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 1 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN1$rows$Region$labels, col=DESIGN1$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean1,   
 col = DESIGN1$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)

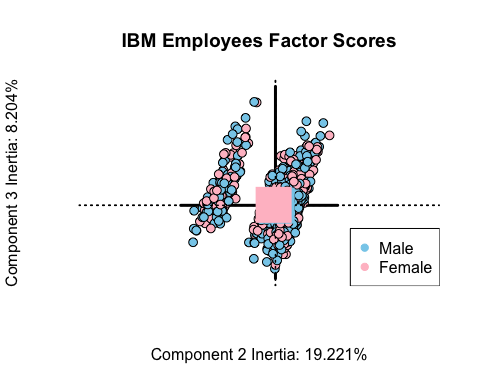


## $col  
## [,1]   
## [1,] "skyblue"  
## [2,] "pink"   
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.003188647  
##   
## $constraints$miny  
## [1] -0.0005701096  
##   
## $constraints$maxx  
## [1] 0.002182394  
##   
## $constraints$maxy  
## [1] 0.0008329746

* Both the means lie almost on the origin so can’t make any significant inference for both the Components.

## Component 2 Vs Component 3

name\_the\_plot1 <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 2, y\_axis = 3,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN1$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 3 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN1$rows$Region$labels, col=DESIGN1$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean1,   
 col = DESIGN1$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)

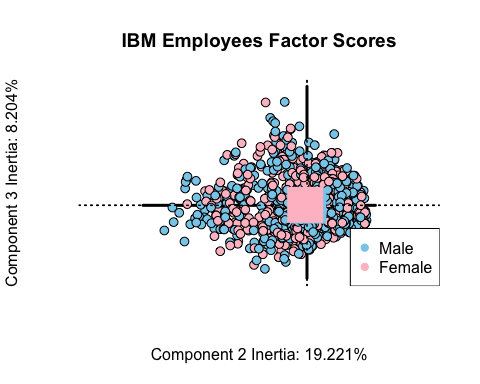


## $col  
## [,1]   
## [1,] "skyblue"  
## [2,] "pink"   
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.003188647  
##   
## $constraints$miny  
## [1] -0.0005701096  
##   
## $constraints$maxx  
## [1] 0.002182394  
##   
## $constraints$maxy  
## [1] 0.0008329746

* Component 2: Maybe we can say that Component 2 divides Female vs Male
* Component 3: No significant inference

## Component 1 Vs Component 3

name\_the\_plot1 <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Factor Scores",  
 x\_axis = 1, y\_axis = 3,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = DESIGN1$rows$Region$color\_observ,   
 display\_names = FALSE,   
 xlab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 3 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)  
legend(x="bottomright", pch = 19, legend = DESIGN1$rows$Region$labels, col=DESIGN1$rows$Region$color\_groups)  
prettyPlot(fi\_all.mean1,   
 col = DESIGN1$rows$Region$color\_groups,   
 display\_names = FALSE,  
 pch = 15,   
 cex = 5.0,  
 dev.new = FALSE,  
 new.plot = FALSE)



## $col  
## [,1]   
## [1,] "skyblue"  
## [2,] "pink"   
##   
## $pch  
## [,1]  
## [1,] 15  
## [2,] 15  
##   
## $constraints  
## $constraints$minx  
## [1] -0.003188647  
##   
## $constraints$miny  
## [1] -0.0005701096  
##   
## $constraints$maxx  
## [1] 0.002182394  
##   
## $constraints$maxy  
## [1] 0.0008329746

## The I-set graphs

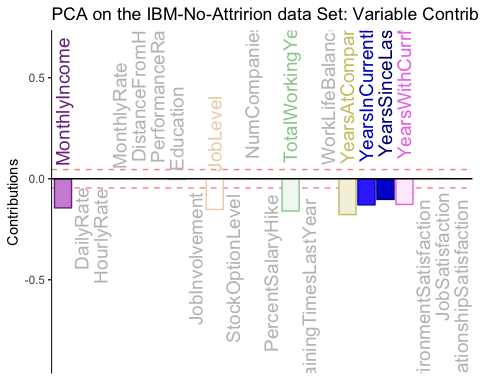
# a graph of the observations  
# a graph of the observations  
  
my\_data1.Imap <- PTCA4CATA::createFactorMap(  
 resPCA$ExPosition.Data$fi,  
 col.points = resPCA$Plotting.Data$fi.col,  
 display.labels = FALSE,  
 alpha.points = .5  
)  
label4Map <- createxyLabels.gen(1,2,  
 lambda =resPCA$ExPosition.Data$eigs,  
 tau = resPCA$ExPosition.Data$t)  
a002.Map.I <- my\_data1.Imap$zeMap + label4Map  
dev.new()  
print(a002.Map.I)

## The J-set graphs

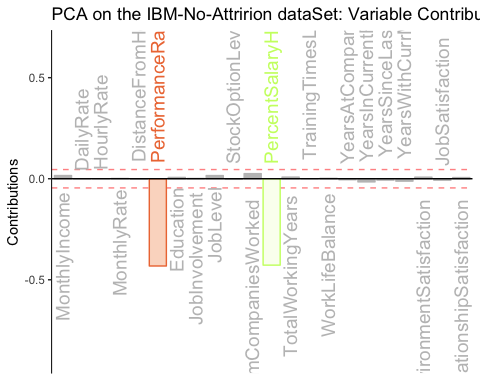
col4J.ibm <- prettyGraphsColorSelection(NCOL(datar))  
baseMap.j <- PTCA4CATA::createFactorMap(resPCA$ExPosition.Data$fj,col.points = col4J.ibm,  
 col.labels = col4J.ibm ,  
 alpha.points = .3)  
# arrows  
zeArrows <- addArrows(resPCA$ExPosition.Data$fj , col = col4J.ibm)  
# A graph for the J-set  
b000.aggMap.j <- baseMap.j$zeMap\_background + # background layer  
 baseMap.j$zeMap\_dots + baseMap.j$zeMap\_text + # dots & labels  
 label4Map + zeArrows  
# We print this Map with the following code  
dev.new()  
print(b000.aggMap.j)

## Contribution

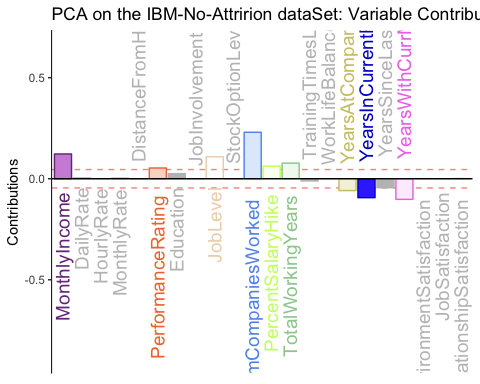
signed.ctrJ <- resPCA$ExPosition.Data$cj \* sign(resPCA$ExPosition.Data$fj)  
b003.ctrJ.s.1 <- PrettyBarPlot2(signed.ctrJ[,1],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'PCA on the IBM-No-Attririon data Set: Variable 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 = 'PCA 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)

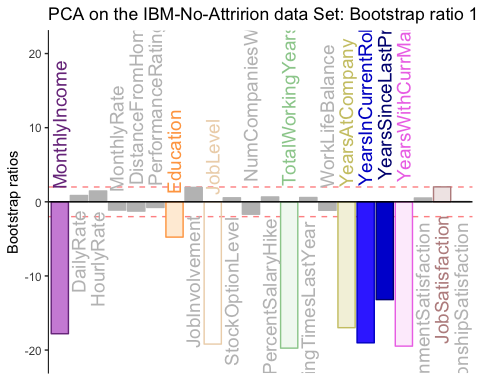


b004.ctrJ.s.3 <- PrettyBarPlot2(signed.ctrJ[,3],  
 threshold = 1 / NROW(signed.ctrJ),  
 font.size = 5,  
 color4bar = gplots::col2hex(col4J.ibm), # we need hex code  
 main = 'PCA on the IBM-No-Attririon dataSet: Variable Contributions (Signed)',  
 ylab = 'Contributions',  
 ylim = c(1.2\*min(signed.ctrJ),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

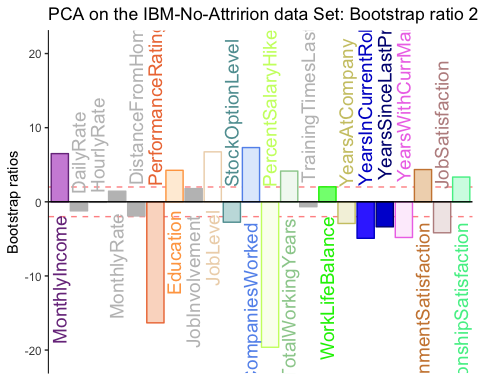
 From the contribution graph it is clear that on the first component Monthly Income, JobLevel , TotalWorkingYears , YearsAtCompany , YearsInCurrentRole , YearsSinceLastPromotion is evident . For the second component Performance Rating and PerformanceSalaryHike are major contributors. For the third component MonthlyIncome, PerformanceRating, NumCompaniesWorked , YearsINCurrentRole are major contributors.

## Bootstrap Ratios

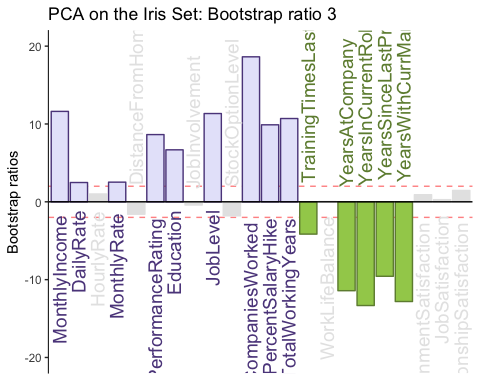
BR <- resPCA.inf$Inference.Data$fj.boots$tests$boot.ratios  
laDim = 1  
ba001.BR1 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0( 'PCA on the IBM-No-Attririon data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
)  
print(ba001.BR1)



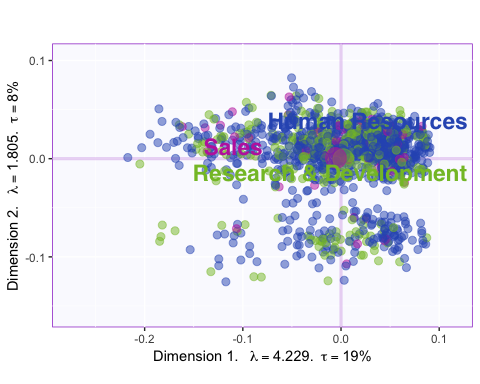
#  
laDim = 2  
ba002.BR2 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 color4bar = gplots::col2hex(col4J.ibm),  
 main = paste0(  
 'PCA on the IBM-No-Attririon data Set: Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios'  
)  
print(ba002.BR2)



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



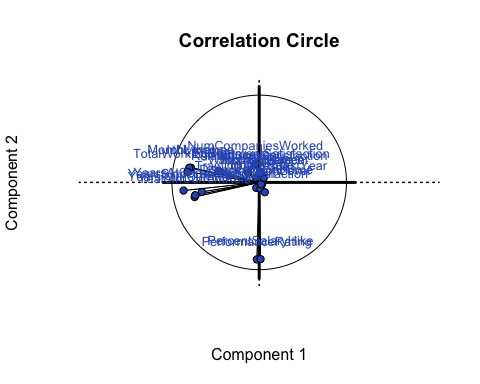
# Bootstrap for CI:  
BootCube.Gr <- PTCA4CATA::Boot4Mean(resPCA$ExPosition.Data$fi,   
 design = datae,  
 niter = 100,  
 suppressProgressBar = TRUE)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Bootstrap ratios ----  
bootRatios.Gr <- boot.ratio.test(BootCube.Gr$BootCube)  
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# eigenvalues: MonteCarlo Approach ----  
#   
random.eigen <- data4PCCAR::monteCarlo.eigen(X = datar, nIter = 100)  
#  
# eigenvalues: Bootstrap approach  
#  
bootstrap.eigen <- data4PCCAR::boot.eigen(datar, nIter = 100)  
# Mean Map  
# create the map for the means  
# get the means by groups  
  
dataMeans <- PTCA4CATA::getMeans(resPCA$ExPosition.Data$fi, datae)  
# a vector of color for the means  
col4data <- resPCA$Plotting.Data$fi.col  
col4Means <- unique(col4data)  
# the map  
MapGroup <- PTCA4CATA::createFactorMap(dataMeans,  
 # use the constraint from the main map  
 constraints = my\_data1.Imap$constraints,  
 col.points = col4Means,  
 cex = 7, # size of the dot (bigger)  
 col.labels = col4Means,  
 text.cex = 6)  
# The map with observations and group means  
a003.Map.I.withMeans <- a002.Map.I +  
 MapGroup$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 <- PTCA4CATA::MakeCIEllipses(BootCube.Gr$BootCube[,1:2,],  
 names.of.factors = c("Dimension 1","Dimension 2"),  
 col = col4Means,  
 p.level = .95  
)  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# create the I-map with Observations, means and confidence intervals  
#  
a004.Map.I.withCI <- a002.Map.I + MapGroup$zeMap\_text + GraphElli  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# plot it!  
dev.new()  
print(a004.Map.I.withCI)

## Correlation Circle

correlationCircle <- correlationPlotter(data\_matrix = my\_data1 , factor\_scores =resPCA$ExPosition.Data$fi , x\_axis = 1, y\_axis = 2, col = NULL ,pch = NULL, xlab = NULL, ylab = NULL, main = "Correlation Circle", asp = 1, dev.new = FALSE)

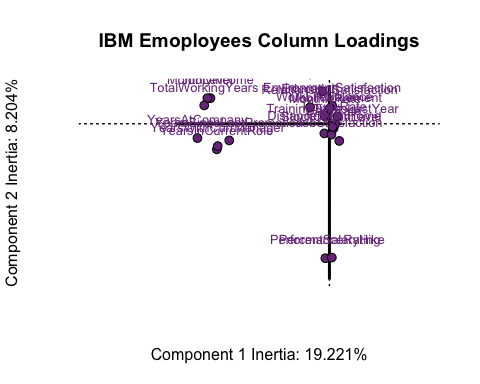


## Loadings

## Component 1 Vs Component 2

Loadings describe the similarity (angular distance) between the variables. 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.

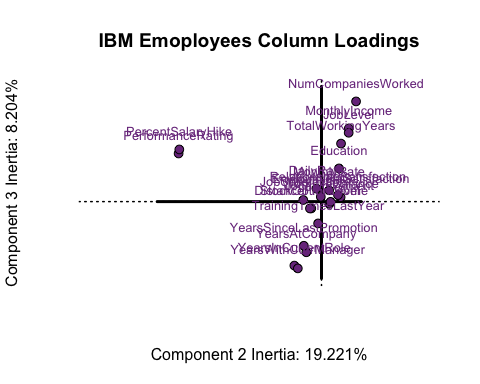
name\_another\_plot <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fj,   
 dev.new=FALSE,  
 main = "IBM Emoployees Column Loadings",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$cj,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = resPCA$Plotting.Data$fj.col ,   
 display\_names = TRUE,   
 xlab = paste0("Component 1 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)



* Component 1: YearsatCompany & JobLevel & MonthlyIncome
* Component 2: Performance Rating VS NumCompaniesWorked

## Component 2 Vs Component 3

name\_another\_plot <- prettyPlot(data\_matrix = resPCA$ExPosition.Data$fj,   
 dev.new=FALSE,  
 main = "IBM Emoployees Column Loadings",  
 x\_axis = 2, y\_axis = 3,   
 contributionCircles = FALSE, contributions = resPCA$ExPosition.Data$cj,   
 display\_points = TRUE, pch = 21, cex = 1.2, col = resPCA$Plotting.Data$fj.col,   
 display\_names = TRUE,   
 xlab = paste0("Component 2 Inertia: ", round(resPCA$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 3 Inertia: ", round(resPCA$ExPosition.Data$t[2],3), "%")  
)



* Component 2: YearsatCompany VS NumCompaniesWorked
* Component 3: Performance Rating & PercentSalaryHike

## Summary

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

* Component 1: Usually the Research and Development people have more work-experience & Monthly Income in comparison to other department type
* Component 2: Human Resources people have high Performance Rating
* Component 3: Males have worked in more number of Companies