PCA

Richa Singh

07/09/2018

## 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. A possible application would be a pattern classification task, where we want to reduce the computational costs and the error of parameter estimation by reducing the number of dimensions of our feature space by extracting a subspace that describes our data best.

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. 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 are all Quantiitavtive Variables. (Out of which PerformanceRating,JobInvolvement,Joblevel,StockOptionLevel, EnvironmentSatisfaction,JobSatisfaction and RelationshipSatisfaction are Ordinal Variables. Age,DistanceFromHome are ratio Variables)

Attrition,BusinessTravel,EducationField,Gender,JobRole,MaritalStatus,OverTime are Qualitative variable (Nominal Variables).

my\_data <- read.csv("IBM-HR-Emplyee-NoAttrition.csv",header = TRUE)  
rownames(my\_data) <- my\_data$Subj

Analyzing the dataset we see that for calculating the correlation,resPCA and all other inferences we need to convert the Qualitative variable to Quantiitavtive Variables.(i.e Cateogorical variables to numerical quantity) as PCA works for only Quantiitavtive Variables. Also, when we see the Attrition column we observe has no correlation with other variables as the dataset has NoAttrition for any employees and the correlation plot will show NA’s for all the rows concerning Attrition with other variables. To avoid this problem we drop the Attrition column.

my\_data$Attrition <- as.numeric(as.factor(my\_data$Attrition))  
my\_data$BusinessTravel <- as.numeric(as.factor(my\_data$BusinessTravel))  
my\_data$Department <- as.numeric(as.factor(my\_data$Department))  
my\_data$EducationField <- as.numeric(as.factor(my\_data$EducationField))  
my\_data$Gender <- as.numeric(as.factor(my\_data$Gender))  
my\_data$JobRole <- as.numeric(as.factor(my\_data$JobRole))  
my\_data$MaritalStatus <- as.numeric(as.factor(my\_data$MaritalStatus))  
my\_data$OverTime <- as.numeric(as.factor(my\_data$OverTime))  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

my\_data = select(my\_data, -Attrition)

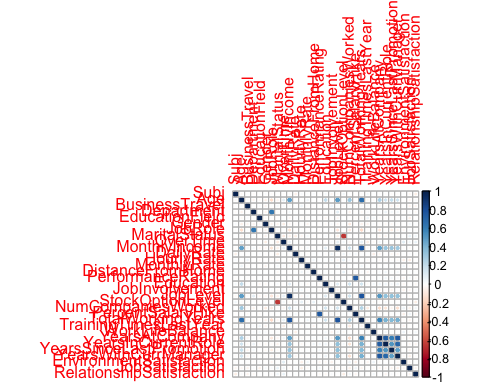
## Results

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

library(corrplot)

## corrplot 0.84 loaded

cor.res <- cor(my\_data)  
corrplot(cor.res,method = "circle")

 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(ExPosition)

## Loading required package: prettyGraphs

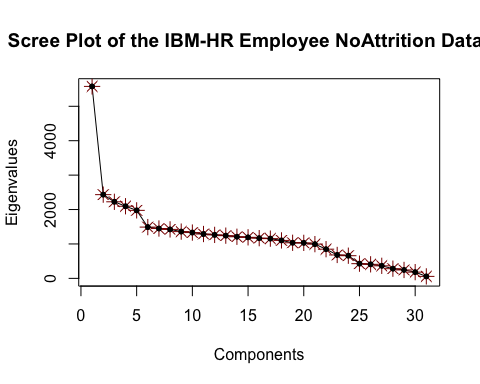
res\_pca <- epPCA(my\_data, center = TRUE, scale = TRUE, DESIGN = my\_data$Subj , graphs = FALSE)

## 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).

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

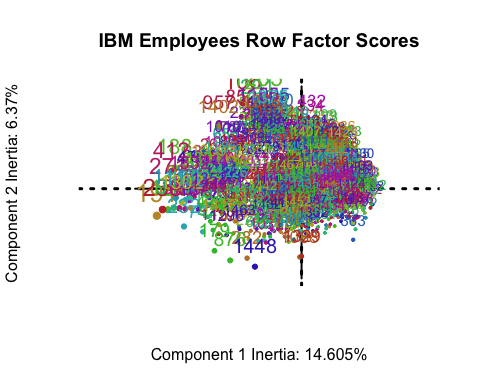


## Factor Scores

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.

name\_the\_plot <- prettyPlot(data\_matrix = res\_pca$ExPosition.Data$fi,   
 dev.new=FALSE,  
 main = "IBM Employees Row Factor Scores",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = TRUE,   
 contributions = res\_pca$ExPosition.Data$ci,   
 display\_points = TRUE, pch = 20, cex = 0.5,   
 col = res\_pca$Plotting.Data$fi.col,  
 fg.line.width = 2,  
 bg.line.width= 3,  
 asp = 1,  
 display\_names = TRUE,  
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$ExPosition.Data$t[2],3), "%")  
)

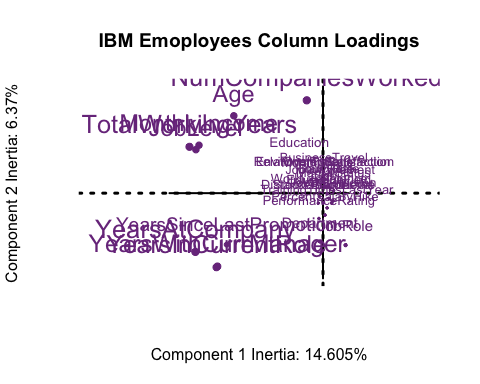
 \* Component 1: 191 & 238 & 124 & 271 & 412 & 919 & 188 & manymore VS 302 & manymore

* Component 2: 1265 & 106 & 291 & 1448 & 771 & manymore VS manymore

## Loadings

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 = res\_pca$ExPosition.Data$fj,   
 dev.new=FALSE,  
 main = "IBM Emoployees Column Loadings",  
 x\_axis = 1, y\_axis = 2,   
 contributionCircles = TRUE,   
 contributions = res\_pca$ExPosition.Data$cj,   
 display\_points = TRUE, pch = 20, cex = 0.5,   
 col = res\_pca$Plotting.Data$fj.col,   
 display\_names = TRUE,   
 fg.line.width = 2,  
 bg.line.width= 3,  
 asp = 1,  
 xlab = paste0("Component 1 Inertia: ", round(res\_pca$ExPosition.Data$t[1],3), "%"),  
 ylab = paste0("Component 2 Inertia: ", round(res\_pca$ExPosition.Data$t[2],3), "%")  
)

 \* Component 1: (TotalWorkingYears & YearsatCompany & JobLevel & MonthlyIncome & YearsInCurrentRole & YearsCurrManager & YearsSinceLastPromotion) VS (No variables on the negative axis) \* Component 2: ( NumCompaniesWorked & Age & MonthlyIncome & TotalWorkingYears & JobLevel ) VS (YearsInCurrentRole & YearsCurrManager & JobRole & YearsSinceLastPromotion & Department)

##Visualization

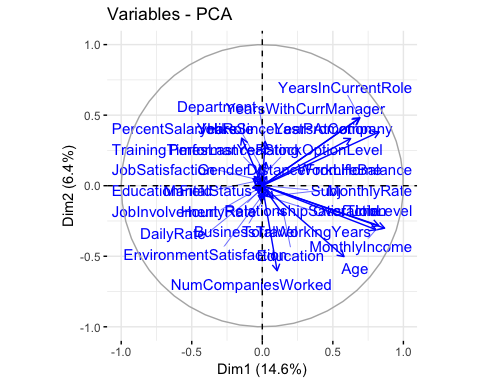
To make the above results more visible I decided to go for a more clear visulization of the contribution of variables to the components.

library(factoextra)

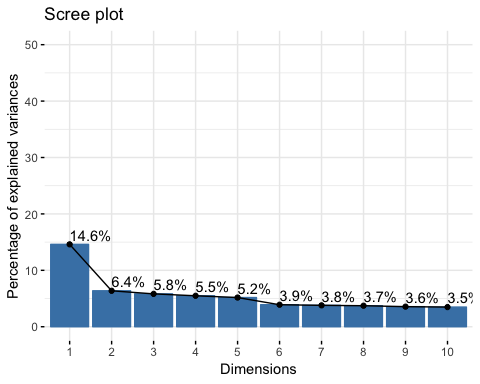
## Loading required package: ggplot2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

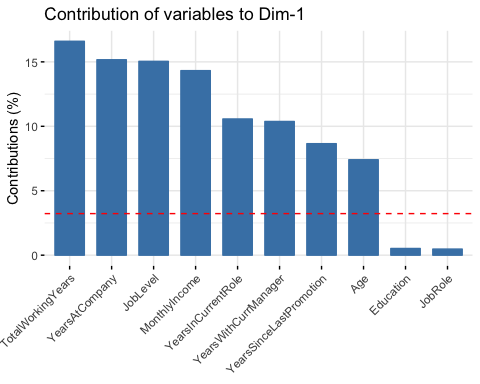
library(FactoMineR)  
res.pca1 <- PCA(my\_data, graph = FALSE)  
fviz\_pca\_var(res.pca1, col.var = "blue",repel = TRUE)



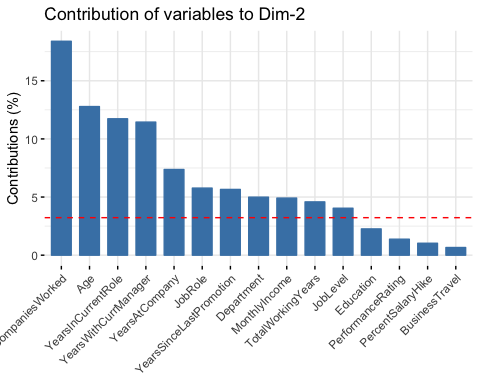
fviz\_screeplot(res.pca1, addlabels = TRUE, ylim = c(0, 50))



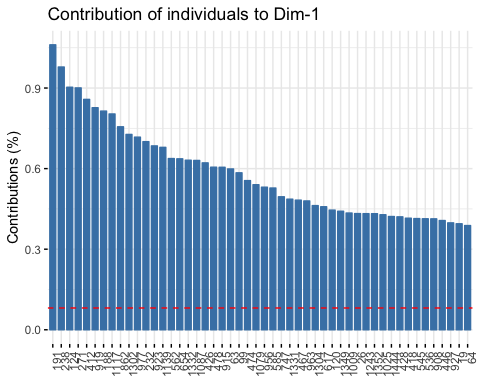
#Contributions of variables to PC1  
fviz\_contrib(res.pca1, choice = "var", axes = 1, top = 10)



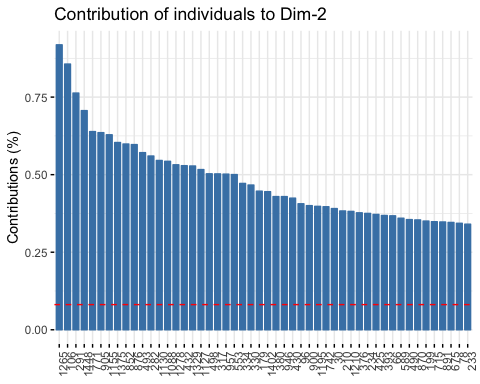
#Contributions of variables to PC2  
fviz\_contrib(res.pca1, choice = "var", axes = 2, top = 15)



fviz\_contrib(res.pca1, choice = "ind", axes = 1,top = 50) + theme(axis.text.x = element\_text(angle = 90))



fviz\_contrib(res.pca1, choice = "ind", axes = 2, top=50) + theme(axis.text.x = element\_text(angle = 90))



##Summary

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

* Component 1: 191 & 238 & 124 & 271 & 412 & 919 & 188 & manymore employees had more working expeirence, monthly income and more senior job level.
* Component 2: 1265 & 106 & 291 & 1448 & 771 & manymore employees had worked in more number of companies and were more aged.