Assignmet 6 [R]\_PCA

## LIBRARIES

Here for this Project, I think we need to add another library for plotting the 3D graph, **Scatterplot3d** is the library that provides information and implementation for the same.

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.4 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(matlib)  
library(scatterplot3d)

## Data set 1 : SWISS

Here, from previous project, we have selected Swiss data set as our 1st data set to perform PCA. This data set is about Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

[,1] Fertility Ig, ‘common standardized fertility measure’

[,2] Agriculture % of males involved in agriculture as occupation

[,3] Examination % draftees receiving highest mark on army examination

[,4] Education % education beyond primary school for draftees.

[,5] Catholic % ‘catholic’ (as opposed to ‘protestant’).

[,6] Infant.Mortality live births who live less than 1 year.

Swiss <- select(swiss,Fertility:Infant.Mortality)  
head(Swiss)

## Fertility Agriculture Examination Education Catholic  
## Courtelary 80.2 17.0 15 12 9.96  
## Delemont 83.1 45.1 6 9 84.84  
## Franches-Mnt 92.5 39.7 5 5 93.40  
## Moutier 85.8 36.5 12 7 33.77  
## Neuveville 76.9 43.5 17 15 5.16  
## Porrentruy 76.1 35.3 9 7 90.57  
## Infant.Mortality  
## Courtelary 22.2  
## Delemont 22.2  
## Franches-Mnt 20.2  
## Moutier 20.3  
## Neuveville 20.6  
## Porrentruy 26.6

Now, we will calculate the Covariance and Correlations our basic as we wants to calculate the PCA. So, lets try which is preferable for us to calculate. Even after that we will need to have the Eigen Values and EigenVectors for the same. As we did that in previous Assignmet we are doing it again just for the references.

# Calculate the Covariance  
cov\_sw <- cov(Swiss)  
cov\_sw

## Fertility Agriculture Examination Education Catholic  
## Fertility 156.04250 100.169149 -64.366929 -79.729510 241.56320  
## Agriculture 100.16915 515.799417 -124.392831 -139.657401 379.90438  
## Examination -64.36693 -124.392831 63.646623 53.575856 -190.56061  
## Education -79.72951 -139.657401 53.575856 92.456059 -61.69883  
## Catholic 241.56320 379.904376 -190.560611 -61.698830 1739.29454  
## Infant.Mortality 15.15619 -4.025851 -2.649537 -2.781684 21.31812  
## Infant.Mortality  
## Fertility 15.156193  
## Agriculture -4.025851  
## Examination -2.649537  
## Education -2.781684  
## Catholic 21.318116  
## Infant.Mortality 8.483802

# Calculate the Correlation   
cor\_sw <- cor(Swiss)  
cor\_sw

## Fertility Agriculture Examination Education Catholic  
## Fertility 1.0000000 0.35307918 -0.6458827 -0.66378886 0.4636847  
## Agriculture 0.3530792 1.00000000 -0.6865422 -0.63952252 0.4010951  
## Examination -0.6458827 -0.68654221 1.0000000 0.69841530 -0.5727418  
## Education -0.6637889 -0.63952252 0.6984153 1.00000000 -0.1538589  
## Catholic 0.4636847 0.40109505 -0.5727418 -0.15385892 1.0000000  
## Infant.Mortality 0.4165560 -0.06085861 -0.1140216 -0.09932185 0.1754959  
## Infant.Mortality  
## Fertility 0.41655603  
## Agriculture -0.06085861  
## Examination -0.11402160  
## Education -0.09932185  
## Catholic 0.17549591  
## Infant.Mortality 1.00000000

# Calculate the Scaled Covariance  
scaled\_sw <- scale(Swiss)  
ScaleCov <- cov(scaled\_sw)  
ScaleCov

## Fertility Agriculture Examination Education Catholic  
## Fertility 1.0000000 0.35307918 -0.6458827 -0.66378886 0.4636847  
## Agriculture 0.3530792 1.00000000 -0.6865422 -0.63952252 0.4010951  
## Examination -0.6458827 -0.68654221 1.0000000 0.69841530 -0.5727418  
## Education -0.6637889 -0.63952252 0.6984153 1.00000000 -0.1538589  
## Catholic 0.4636847 0.40109505 -0.5727418 -0.15385892 1.0000000  
## Infant.Mortality 0.4165560 -0.06085861 -0.1140216 -0.09932185 0.1754959  
## Infant.Mortality  
## Fertility 0.41655603  
## Agriculture -0.06085861  
## Examination -0.11402160  
## Education -0.09932185  
## Catholic 0.17549591  
## Infant.Mortality 1.00000000

eigenCovsw <- eigen(cov\_sw)  
eigenCovsw

## eigen() decomposition  
## $values  
## [1] 1921.562488 466.657132 145.284829 22.649607 13.377631 6.191249  
##   
## $vectors  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] -0.15163143 -0.14270789 0.81454413 -0.49552828 0.12247267 -0.1805890073  
## [2,] -0.28121756 -0.85914886 -0.35256541 -0.24078519 0.02235164 0.0006755062  
## [3,] 0.12207834 0.17688621 -0.18767793 -0.57042350 -0.76887882 -0.0450281625  
## [4,] 0.06329733 0.32260928 -0.40096045 -0.58075837 0.61904343 -0.1032102386  
## [5,] -0.93748965 0.32543441 -0.07870742 0.04104832 -0.08547871 0.0043788222  
## [6,] -0.01131739 0.01498883 0.10014161 -0.17923725 0.05296062 0.9770814149

eigenCorsw <- eigen(cor\_sw)  
eigenCorsw

## eigen() decomposition  
## $values  
## [1] 3.1997570 1.1883082 0.8476098 0.4389287 0.2045337 0.1208626  
##   
## $vectors  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] -0.4569876 0.3220284 0.17376638 0.53555794 0.38308893 0.47295441  
## [2,] -0.4242141 -0.4115132 -0.03834472 -0.64291822 0.37495215 0.30870058  
## [3,] 0.5097327 0.1250167 0.09123696 -0.05446158 0.81429082 -0.22401686  
## [4,] 0.4543119 0.1790495 -0.53239316 -0.09738818 -0.07144564 0.68081610  
## [5,] -0.3501111 0.1458730 -0.80680494 0.09947244 0.18317236 -0.40219666  
## [6,] -0.1496668 0.8111645 0.16010636 -0.52677184 -0.10453530 -0.07457754

Now, we will calculate the Percent Variance, and that will give us the information about, what proportion of total variance is explained by the First, second and till the end of principal component.

We can calculate Cumulative percent variance just to see that how many columns represents major portion of the data information.

PV <- eigenCorsw$values/sum(eigenCorsw$values)  
PV

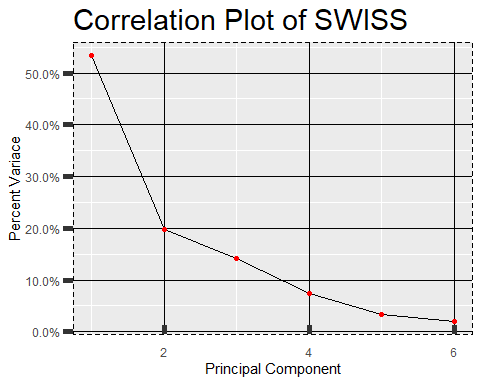
## [1] 0.53329283 0.19805137 0.14126830 0.07315478 0.03408895 0.02014376

#Cumulative percent variance  
cumsum(PV)

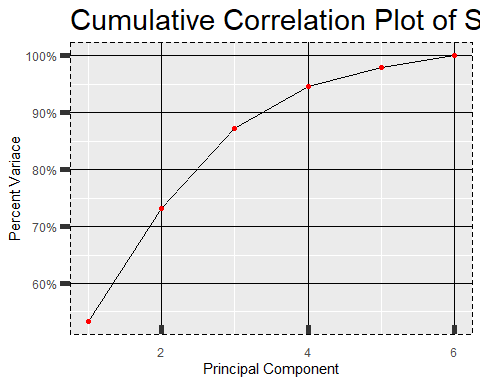
## [1] 0.5332928 0.7313442 0.8726125 0.9457673 0.9798562 1.0000000

Now, using the Graphs we will can see that how many variables we need to reperesent the data and what are the variables we can reduce.

SwCorPlot <- qplot(c(1:6),PV) +  
 geom\_line()+  
 geom\_point(shape = 20,colour = "red", fill = NA , size = 2, stroke = 1 ) +  
 xlab("Principal Component") +  
 ylab("Percent Variace") +  
 ggtitle("Correlation Plot of SWISS") +  
 scale\_y\_continuous(labels = scales::percent)+  
 theme(plot.title = element\_text(size = rel(2))) +  
 theme(panel.grid.major = element\_line(colour = "black")) +  
 theme(panel.border = element\_rect(linetype = "dashed", fill = NA)) +  
 theme(axis.ticks = element\_line(size = 2)) +  
   
 theme(  
 axis.ticks.length.y = unit(.25, "cm"),  
 axis.ticks.length.x = unit(-.25, "cm"),  
 axis.text.x = element\_text(margin = margin(t = .3, unit = "cm"))  
 )   
SwCorPlot



SwCumCorPlot <- qplot(c(1:6),cumsum(PV)) +  
 geom\_line()+  
 geom\_point(shape = 20,colour = "red", fill = NA , size = 2, stroke = 1 ) +  
 xlab("Principal Component") +  
 ylab("Percent Variace") +  
 ggtitle("Cumulative Correlation Plot of SWISS") +  
 scale\_y\_continuous(labels = scales::percent) +  
 theme(plot.title = element\_text(size = rel(2))) +  
 theme(panel.grid.major = element\_line(colour = "black")) +  
 theme(panel.border = element\_rect(linetype = "dashed", fill = NA)) +  
 theme(axis.ticks = element\_line(size = 2)) +  
   
 theme(  
 axis.ticks.length.y = unit(.25, "cm"),  
 axis.ticks.length.x = unit(-.25, "cm"),  
 axis.text.x = element\_text(margin = margin(t = .3, unit = "cm"))  
 )   
SwCumCorPlot

 Now, We will count the Principal Components Score.

The sample principal components are defined as those linear combinations which have maximum sample variance. If we project the 47 data points onto the first eigen vectors, the projected values are called the first principal component.

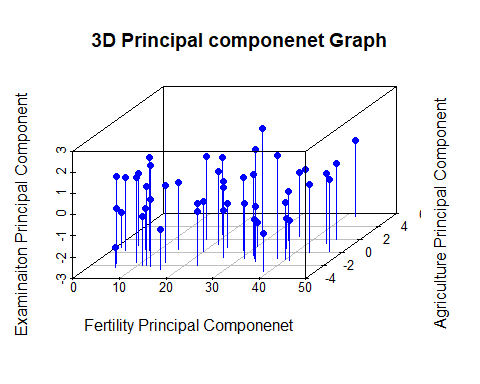
sw\_selectedEigenValues <- eigenCorsw$vectors[,1:4]  
colnames(sw\_selectedEigenValues) = c("Axis 1", "Axis 2", "Axis 3", "Axis 4")  
row.names(sw\_selectedEigenValues) = colnames(Swiss)  
sw\_selectedEigenValues

## Axis 1 Axis 2 Axis 3 Axis 4  
## Fertility -0.4569876 0.3220284 0.17376638 0.53555794  
## Agriculture -0.4242141 -0.4115132 -0.03834472 -0.64291822  
## Examination 0.5097327 0.1250167 0.09123696 -0.05446158  
## Education 0.4543119 0.1790495 -0.53239316 -0.09738818  
## Catholic -0.3501111 0.1458730 -0.80680494 0.09947244  
## Infant.Mortality -0.1496668 0.8111645 0.16010636 -0.52677184

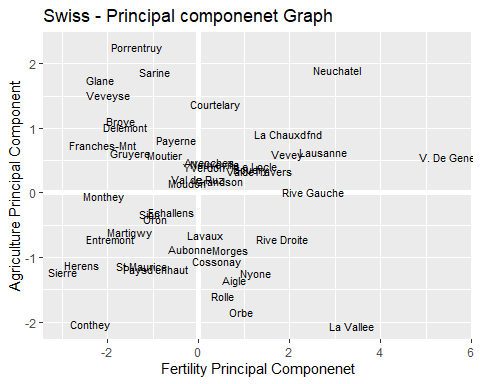
PC1 <- as.matrix(scaled\_sw) %\*% sw\_selectedEigenValues[,1]  
PC2 <- as.matrix(scaled\_sw) %\*% sw\_selectedEigenValues[,2]  
PC3 <- as.matrix(scaled\_sw) %\*% sw\_selectedEigenValues[,3]  
PC4 <- as.matrix(scaled\_sw) %\*% sw\_selectedEigenValues[,4]  
  
PC <- data.frame(French\_speaking\_provinces = row.names(Swiss), PC1,PC2,PC3,PC4)  
head(PC)

## French\_speaking\_provinces PC1 PC2 PC3  
## Courtelary Courtelary 0.3596632 1.3844529 0.8505125  
## Delemont Delemont -1.6166972 1.0150729 -0.5420075  
## Franches-Mnt Franches-Mnt -2.0816947 0.7380445 -0.4676275  
## Moutier Moutier -0.7396119 0.5895183 0.5729651  
## Neuveville Neuveville 0.3774470 0.4440442 0.6215417  
## Porrentruy Porrentruy -1.3545746 2.2673656 -0.3467783  
## PC4  
## Courtelary 0.9012204  
## Delemont 0.5005108  
## Franches-Mnt 1.4858476  
## Moutier 1.0608299  
## Neuveville 0.2434455  
## Porrentruy -0.3044908

scatterplot3d(PC[,1:3], angle = 55, pch = 16,  
 main = "3D Principal componenet Graph",  
 xlab = "Fertility Principal Componenet",  
 ylab = "Agriculture Principal Component",  
 zlab = "Examinaiton Principal Component",  
 color = "Blue",  
 type = "h"  
 )



ggplot(PC, aes(PC1,PC2))+  
 modelr::geom\_ref\_line(h=0) +  
 modelr::geom\_ref\_line(v=0) +  
 geom\_text(aes(label = French\_speaking\_provinces),size =3) +  
 xlab("Fertility Principal Componenet") +  
 ylab("Agriculture Principal Component") +  
 ggtitle("Swiss - Principal componenet Graph")

 **CONCLUSION**: In this data set called “Swiss”, we found that 94% of the data is been represented by the 1st 4 eigen vectors, and other two is not making major impact on the data set. Moreover Axis 1 and Axis 2 along have the 75% of the data. Axis 1 has a strong negative loading for Fertility, Agriculture, Catholic, Infant Mortality and strong negative loadings for Examination and Education. Same as that, Axis 2 has Mostly positive relationship with every section except Agriculture. So, As per me they shows highest contrast and likelyness to the diven data.

Just to make sure and see the behavior of the Examination which is on Axis 3, we have implemented the 3D model to see the dept of the influence on the data.

## DATASET - 2 Attitude for the PCA.

For this data set we did the same process and same Graphs just to interpretation and practice. And the end I have provided my own Comparison and Observation of the exercise. And here are the information of the dataset.

From a survey of the clerical employees of a large financial organization, the data are aggregated from the questionnaires of the approximately 35 employees for each of 30 (randomly selected) departments. The numbers give the percent proportion of favourable responses to seven questions in each department.

**Y rating numeric Overall rating**

X[1] complaints numeric Handling of employee complaints

X[2] privileges numeric Does not allow special privileges

X[3] learning numeric Opportunity to learn

X[4] raises numeric Raises based on performance

X[5] critical numeric Too critical

X[6] advance numeric Advancement

# Calculate the Covariance  
Attitude <- select(attitude, rating:advance)  
at <- drop\_na(Attitude)  
head(at)

## rating complaints privileges learning raises critical advance  
## 1 43 51 30 39 61 92 45  
## 2 63 64 51 54 63 73 47  
## 3 71 70 68 69 76 86 48  
## 4 61 63 45 47 54 84 35  
## 5 81 78 56 66 71 83 47  
## 6 43 55 49 44 54 49 34

cov\_at <- cov(at)  
cov\_at

## rating complaints privileges learning raises critical  
## rating 148.17126 133.77931 63.46437 89.10460 74.68851 18.84253  
## complaints 133.77931 177.28276 90.95172 93.25517 92.64138 24.73103  
## privileges 63.46437 90.95172 149.70575 70.84598 56.67126 17.82529  
## learning 89.10460 93.25517 70.84598 137.75747 78.13908 13.46782  
## raises 74.68851 92.64138 56.67126 78.13908 108.10230 38.77356  
## critical 18.84253 24.73103 17.82529 13.46782 38.77356 97.90920  
## advance 19.42299 30.76552 43.21609 64.19770 61.42299 28.84598  
## advance  
## rating 19.42299  
## complaints 30.76552  
## privileges 43.21609  
## learning 64.19770  
## raises 61.42299  
## critical 28.84598  
## advance 105.85747

# Calculate the Correlation   
cor\_at <- cor(at)  
cor\_at

## rating complaints privileges learning raises critical  
## rating 1.0000000 0.8254176 0.4261169 0.6236782 0.5901390 0.1564392  
## complaints 0.8254176 1.0000000 0.5582882 0.5967358 0.6691975 0.1877143  
## privileges 0.4261169 0.5582882 1.0000000 0.4933310 0.4454779 0.1472331  
## learning 0.6236782 0.5967358 0.4933310 1.0000000 0.6403144 0.1159652  
## raises 0.5901390 0.6691975 0.4454779 0.6403144 1.0000000 0.3768830  
## critical 0.1564392 0.1877143 0.1472331 0.1159652 0.3768830 1.0000000  
## advance 0.1550863 0.2245796 0.3432934 0.5316198 0.5741862 0.2833432  
## advance  
## rating 0.1550863  
## complaints 0.2245796  
## privileges 0.3432934  
## learning 0.5316198  
## raises 0.5741862  
## critical 0.2833432  
## advance 1.0000000

# Calculate the Scaled Covariance  
scaled\_at <- scale(at)  
ScaleCovat <- cov(scaled\_at)  
ScaleCovat

## rating complaints privileges learning raises critical  
## rating 1.0000000 0.8254176 0.4261169 0.6236782 0.5901390 0.1564392  
## complaints 0.8254176 1.0000000 0.5582882 0.5967358 0.6691975 0.1877143  
## privileges 0.4261169 0.5582882 1.0000000 0.4933310 0.4454779 0.1472331  
## learning 0.6236782 0.5967358 0.4933310 1.0000000 0.6403144 0.1159652  
## raises 0.5901390 0.6691975 0.4454779 0.6403144 1.0000000 0.3768830  
## critical 0.1564392 0.1877143 0.1472331 0.1159652 0.3768830 1.0000000  
## advance 0.1550863 0.2245796 0.3432934 0.5316198 0.5741862 0.2833432  
## advance  
## rating 0.1550863  
## complaints 0.2245796  
## privileges 0.3432934  
## learning 0.5316198  
## raises 0.5741862  
## critical 0.2833432  
## advance 1.0000000

eigenCovat <- eigen(cov\_at)  
eigenCovat

## eigen() decomposition  
## $values  
## [1] 519.79278 134.14087 97.06354 85.23927 41.01460 25.74079 21.79436  
##   
## $vectors  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] -0.4467200 0.42184464 -0.2400295 0.1261907 0.20102006 0.47248738  
## [2,] -0.5206244 0.37207702 -0.1432265 -0.1081087 -0.37237853 0.02200456  
## [3,] -0.3757726 -0.07632652 0.6513217 -0.6263275 0.07783612 -0.01814123  
## [4,] -0.4209952 -0.14566925 0.1864780 0.4851365 0.62078169 -0.30156007  
## [5,] -0.3762536 -0.23339685 -0.2239163 0.1041004 -0.44682483 -0.59329347  
## [6,] -0.1300302 -0.39828981 -0.6330345 -0.5170667 0.37798712 0.01806699  
## [7,] -0.2290738 -0.66592166 0.1095758 0.2579729 -0.29490695 0.57678447  
## [,7]  
## [1,] 0.5341317  
## [2,] -0.6474239  
## [3,] 0.1734231  
## [4,] -0.2347416  
## [5,] 0.4374176  
## [6,] -0.1147434  
## [7,] -0.0765914

eigenCorat <- eigen(cor\_at)  
eigenCorat

## eigen() decomposition  
## $values  
## [1] 3.7163758 1.1409219 0.8471915 0.6128697 0.3236728 0.2185306 0.1404378  
##   
## $vectors  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 0.4130048 0.39692583 -0.2634492 0.2341088 -0.143063817 0.41241010  
## [2,] 0.4405379 0.33362706 -0.2256118 0.0022033 0.278064283 0.22805897  
## [3,] 0.3547748 0.09575954 0.1882432 -0.8906996 -0.005272287 -0.07598537  
## [4,] 0.4285613 0.04510225 0.3252857 0.2393794 -0.697628303 -0.35282836  
## [5,] 0.4471312 -0.17917304 -0.0404021 0.2428067 0.556488662 -0.58548329  
## [6,] 0.1853508 -0.60263473 -0.7008081 -0.1493497 -0.292800611 0.01154899  
## [7,] 0.3025594 -0.56979573 0.4956644 0.1152367 0.141732580 0.55201569  
## [,7]  
## [1,] 0.597591161  
## [2,] -0.717205075  
## [3,] 0.174304627  
## [4,] -0.200036529  
## [5,] 0.234335158  
## [6,] -0.056334214  
## [7,] -0.004296286

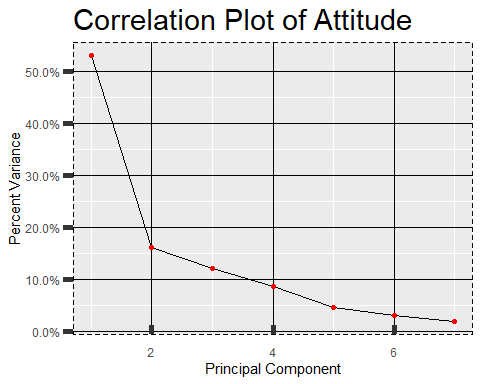
PVat <- eigenCorat$values/sum(eigenCorat$values)  
PVat

## [1] 0.53091082 0.16298884 0.12102736 0.08755281 0.04623897 0.03121866 0.02006254

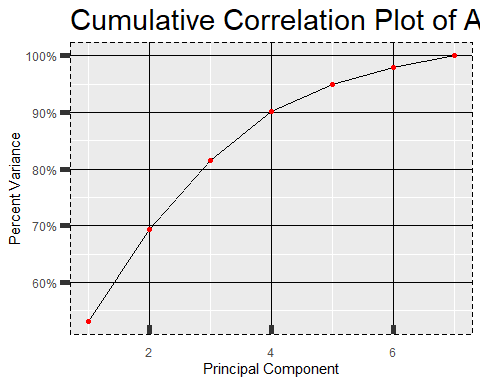
#Cumulative percent variance  
cumsum(PVat)

## [1] 0.5309108 0.6938997 0.8149270 0.9024798 0.9487188 0.9799375 1.0000000

atCorPlot <- qplot(c(1:7),PVat) +  
 geom\_line()+  
 geom\_point(shape = 20,colour = "red", fill = NA , size = 2, stroke = 1 ) +  
 xlab("Principal Component") +  
 ylab("Percent Variance") +  
 ggtitle("Correlation Plot of Attitude") +  
 scale\_y\_continuous(labels = scales::percent)+  
 theme(plot.title = element\_text(size = rel(2))) +  
 theme(panel.grid.major = element\_line(colour = "black")) +  
 theme(panel.border = element\_rect(linetype = "dashed", fill = NA)) +  
 theme(axis.ticks = element\_line(size = 2)) +  
   
 theme(  
 axis.ticks.length.y = unit(.25, "cm"),  
 axis.ticks.length.x = unit(-.25, "cm"),  
 axis.text.x = element\_text(margin = margin(t = .3, unit = "cm"))  
 )   
atCorPlot



atCumCorPlot <- qplot(c(1:7),cumsum(PVat)) +  
 geom\_line()+  
 geom\_point(shape = 20,colour = "red", fill = NA , size = 2, stroke = 1 ) +  
 xlab("Principal Component") +  
 ylab("Percent Variance") +  
 ggtitle("Cumulative Correlation Plot of Attitude") +  
 scale\_y\_continuous(labels = scales::percent) +  
 theme(plot.title = element\_text(size = rel(2))) +  
 theme(panel.grid.major = element\_line(colour = "black")) +  
 theme(panel.border = element\_rect(linetype = "dashed", fill = NA)) +  
 theme(axis.ticks = element\_line(size = 2)) +  
   
 theme(  
 axis.ticks.length.y = unit(.25, "cm"),  
 axis.ticks.length.x = unit(-.25, "cm"),  
 axis.text.x = element\_text(margin = margin(t = .3, unit = "cm"))  
 )   
atCumCorPlot



at\_selectedEigenValues <- eigenCorat$vectors[,1:4]  
colnames(at\_selectedEigenValues) = c("Axis 1", "Axis 2", "Axis 3", "Axis 4")  
row.names(at\_selectedEigenValues) = colnames(at)  
at\_selectedEigenValues

## Axis 1 Axis 2 Axis 3 Axis 4  
## rating 0.4130048 0.39692583 -0.2634492 0.2341088  
## complaints 0.4405379 0.33362706 -0.2256118 0.0022033  
## privileges 0.3547748 0.09575954 0.1882432 -0.8906996  
## learning 0.4285613 0.04510225 0.3252857 0.2393794  
## raises 0.4471312 -0.17917304 -0.0404021 0.2428067  
## critical 0.1853508 -0.60263473 -0.7008081 -0.1493497  
## advance 0.3025594 -0.56979573 0.4956644 0.1152367

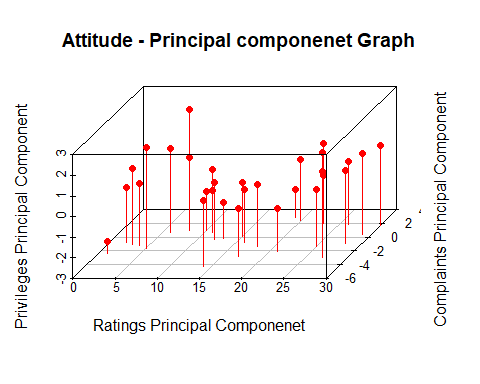
As we can see that 1st colume is showing +ve values and can be the major factor for the attitude, and Complainsts are having -ve relation with the raises, critical and advance. while Privileges shows the positive relations with learning and advance only.

So, we going to use all 3 columns to so represent the major values and their behavior.

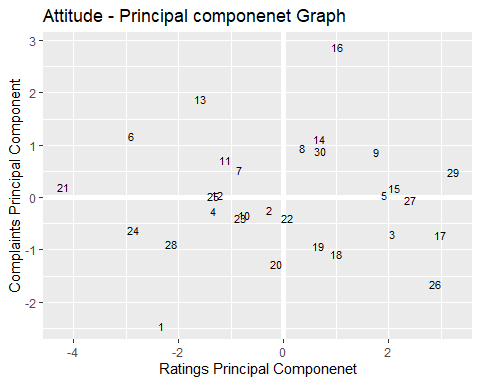
PC1at <- as.matrix(scaled\_at) %\*% at\_selectedEigenValues[,1]  
PC2at <- as.matrix(scaled\_at) %\*% at\_selectedEigenValues[,2]  
PC3at <- as.matrix(scaled\_at) %\*% at\_selectedEigenValues[,3]  
  
PCat <- data.frame(Employee = row.names(at), PC1at,PC2at,PC3at)  
head(PCat)

## Employee PC1at PC2at PC3at  
## 1 1 -2.3276997 -2.44551152 -1.2115473  
## 2 2 -0.2734616 -0.23367075 0.3083785  
## 3 3 2.0691073 -0.70292565 -0.2122343  
## 4 4 -1.3378519 -0.24807722 -1.2399142  
## 5 5 1.9149741 0.04242223 -0.6482546  
## 6 6 -2.8918514 1.17130039 1.6943142

scatterplot3d(PCat[,1:3], angle = 60, pch = 16,  
 main = "Attitude - Principal componenet Graph",  
 xlab = "Ratings Principal Componenet",  
 ylab = "Complaints Principal Component",  
 zlab = "Privileges Principal Component",  
 color = "red",  
 type = "h"  
 )



ggplot(PCat, aes(PC1at,PC2at))+  
 modelr::geom\_ref\_line(h=0) +  
 modelr::geom\_ref\_line(v=0) +  
 geom\_text(aes(label = Employee),size =3) +  
 xlab("Ratings Principal Componenet") +  
 ylab("Complaints Principal Component") +  
 ggtitle("Attitude - Principal componenet Graph")



**CONCLUSION**: In this data set called “Attitude”, we found that 90% of the data is been represented by the 1st 4 eigen vectors, and other two is not making major impact on the data set. Moreover Axis 1 and Axis 2 along have the 69% of the data. Axis 1 has a strong Positive loading for all the parameters. But, Axis 2 shows Negative relations with raises, critical and advance categories. So, As per me first axis shows major data holding for the further interpretations.

Just to make sure and see the behavior of the Examination which is on Axis 3, we have implemented the 3D model to see the dept of the influence on the data.