Final Project

Priyank Thakkar

06/12/2021

1. INTRODUCTION

This data set is all about the students who graduate from the Universities of USA. I am curious to know that does your Major in your study matter or not for your economic success. What is the general trend right now. And, what are the steps we can take before selecting the Major to boost your odds for the same. And mostly what is the share of the women in the different categories.

2. DATA

This data is from the GitHub repository, from Fivethirtyeight.com. All data is from American Community Survey 2010-2012 Public Use Micro data series. All Three files in the repository contains basic earnings and labor force information. “recent-grads.csv” contains a more detailed breakdown, including by sex and by the type of job they got. “grad-students.csv” contains details on graduate school attendees. Here, we have used the data from the file of “recent-grades.csv”. And here is the bifurcation of all the Headers and their Descriptions. Our data contains 174x21 size of entries.

Where, we can see that 174 are rows, they represent distinct 174 majors, from all Major Categories available in the data. Moreover, all 21 columns and their representation are below.

Header	**Description**
• Rank:	Rank by median earnings
• Major_code:	Major code
• Major:	Major description
• Major_category:	Category of major
• Total:	Total number of people with major
• Sample_size:	Sample size (un-weighted) of full-time, year-round ONLY
• Men:	Male graduates
• Women:	Female graduates
• ShareWomen:	Women as share of total
• Employed:	Number employed
• Full_time:	Employed 35 hours or more
• Part_time:	Employed less than 35 hours

- Full_time_year_round: Employed at least 50 weeks and at least 35 hours
- Unemployed: Number unemployed
- Unemployment_rate: Unemployed / (Unemployed + Employed)
- Median: (Normalized) Median earnings of full-time, year-round workers
- P25th: 25th percentile of earnings
- P75th: 75th percentile of earnings
- College_jobs: Number with job requiring a college degree
- Non_college_jobs: Number with job not requiring a college degree
- Low_wage_Jobs: Number in low-wage service jobs

For our variable study, we may need all the variables later on for more explorations. But, here are few of the variables that we can focus on for PCA and EDA.

- Total, Men, Women
- ShareWomen
- Mean
- Employed
- Unemployed
- Unemployment_rate
- College_jobs
- Non_college_jobs
- Low_wage_jobs

2.1 Load the libraries and Data

```
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_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

setwd("D:\\SJSU_HW\\GitHubSJSU\\RStudio_Learning\\Data_set")
data_grade <- read.csv("recent_grads.csv")
recent_grade <- as_tibble(data_grade)

#head of original data
head(recent_grade)
```

```
## # A tibble: 6 x 21
##    Rank Major_code Major Total   Men Women Major_category ShareWomen Sampl
e_size
##   <int>      <int> <chr> <int> <int> <int> <chr>               <dbl>
<int>
## 1   173       3501 LIBR~  1098   134   964 Education           0.878
2
## 2   172       5203 COUN~  4626   931  3695 Psychology & ~      0.799
21
## 3   170       5201 EDUC~  2854   522  2332 Psychology & ~      0.817
7
## 4   171       5202 CLIN~  2838   568  2270 Psychology & ~      0.800
13
## 5   169       3609 ZOOL~  8409  3050  5359 Biology & Lif~      0.637
47
## 6   167       6001 DRAM~ 43249 14440 28809 Arts                0.666
357
## # ... with 12 more variables: Employed <int>, Full_time <int>, Part_time <
int>,
## #   Full_time_year_round <int>, Unemployed <int>, Unemployment_rate <dbl>,
## #   Median <int>, P25th <int>, P75th <int>, College_jobs <int>,
## #   Non_college_jobs <int>, Low_wage_jobs <int>
```

```
#summary of data
summary(recent_grade)
```

```
##       Rank       Major_code      Major               Total       
##  Min.   :  1   Min.   :1100   Length:173         Min.   :   124  
##  1st Qu.: 44   1st Qu.:2403   Class :character   1st Qu.:  4550  
##  Median : 87   Median :3608   Mode  :character   Median : 15104  
##  Mean   : 87   Mean   :3880                      Mean   : 39370  
##  3rd Qu.:130   3rd Qu.:5503                      3rd Qu.: 38910  
##  Max.   :173   Max.   :6403                      Max.   :393735  
##                                                  NA's   :1       
##       Men              Women        Major_category       ShareWomen    
##  Min.   :   119   Min.   :     0   Length:173         Min.   :0.0000  
##  1st Qu.:  2178   1st Qu.:  1778   Class :character   1st Qu.:0.3360  
##  Median :  5434   Median :  8386   Mode  :character   Median :0.5340  
##  Mean   : 16723   Mean   : 22647                      Mean   :0.5222  
##  3rd Qu.: 14631   3rd Qu.: 22554                      3rd Qu.:0.7033  
##  Max.   :173809   Max.   :307087                      Max.   :0.9690  
##  NA's   :1        NA's   :1                           NA's   :1       
##   Sample_size        Employed        Full_time        Part_time     
##  Min.   :   2.0   Min.   :     0   Min.   :   111   Min.   :     0  
##  1st Qu.:  39.0   1st Qu.:  3608   1st Qu.:  3154   1st Qu.:  1030  
##  Median : 130.0   Median : 11797   Median : 10048   Median :  3299  
##  Mean   : 356.1   Mean   : 31193   Mean   : 26029   Mean   :  8832  
##  3rd Qu.: 338.0   3rd Qu.: 31433   3rd Qu.: 25147   3rd Qu.:  9948  
##  Max.   :4212.0   Max.   :307933   Max.   :251540   Max.   :115172  
## 
```

```
##  Full_time_year_round   Unemployed    Unemployment_rate     Median
##  Min.   :   111       Min.   :    0   Min.   :0.00000   Min.   : 22000
##  1st Qu.:  2453       1st Qu.:  304   1st Qu.:0.05031   1st Qu.: 33000
##  Median :  7413       Median :  893   Median :0.06796   Median : 36000
##  Mean   : 19694       Mean   : 2416   Mean   :0.06819   Mean   : 40151
##  3rd Qu.: 16891       3rd Qu.: 2393   3rd Qu.:0.08756   3rd Qu.: 45000
##  Max.   :199897       Max.   :28169   Max.   :0.17723   Max.   :110000
##
##      P25th           P75th          College_jobs    Non_college_jobs
##  Min.   :18500   Min.   : 22000   Min.   :     0   Min.   :     0
##  1st Qu.:24000   1st Qu.: 42000   1st Qu.:  1675   1st Qu.:  1591
##  Median :27000   Median : 47000   Median :  4390   Median :  4595
##  Mean   :29501   Mean   : 51494   Mean   : 12323   Mean   : 13284
##  3rd Qu.:33000   3rd Qu.: 60000   3rd Qu.: 14444   3rd Qu.: 11783
##  Max.   :95000   Max.   :125000   Max.   :151643   Max.   :148395
##
##  Low_wage_jobs
##  Min.   :    0
##  1st Qu.:  340
##  Median : 1231
##  Mean   : 3859
##  3rd Qu.: 3466
##  Max.   :48207
##

# check if data is tibble
is_tibble(recent_grade)

## [1] TRUE
```

2.2 Remove Missing values and Creat Variables.

```
# Check if any NA values
recent_grade %>% summarise(Total_Count = n())

## # A tibble: 1 x 1
##   Total_Count
##         <int>
## 1         173

filter(recent_grade, is.na(Total) | is.na(Men) | is.na(Women) | is.na(ShareWo
men)) %>%
  summarise(Missing_Count = n())

## # A tibble: 1 x 1
##   Missing_Count
##           <int>
## 1             1

# Drop the NA
new_recent_grade <- drop_na(recent_grade)
```

```r
# Generate our new Data only from the variables that are in need.
batch <- select(new_recent_grade,
                Median,
                Total,
                Employed,
                Full_time,
                Unemployed,
                College_jobs,
                Non_college_jobs,
                Low_wage_jobs,
                ShareWomen,
                Major_category,
                Unemployment_rate)
is_tibble(batch)

## [1] TRUE

head(batch)

## # A tibble: 6 x 11
##   Median Total Employed Full_time Unemployed College_jobs Non_college_jobs
##    <int> <int>    <int>     <int>      <int>        <int>            <int>
## 1  22000  1098      742       593         87          288              338
## 2  23400  4626     3777      3154        214         2403             1245
## 3  25000  2854     2125      1848        148         1488              615
## 4  25000  2838     2101      1724        368          986              870
## 5  26000  8409     6259      5043        304         2771             2947
## 6  27000 43249    36165     25147       3040         6994            25313
## # ... with 4 more variables: Low_wage_jobs <int>, ShareWomen <dbl>,
## #   Major_category <chr>, Unemployment_rate <dbl>
```

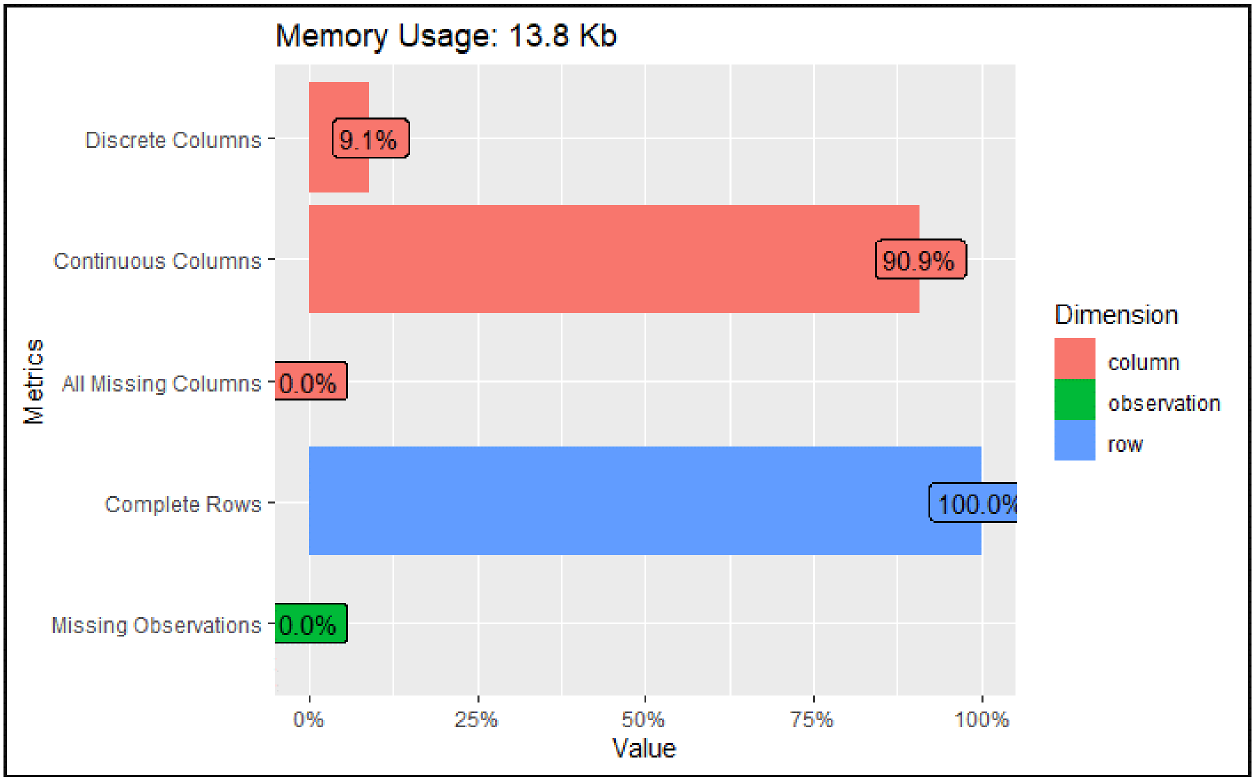
3. EDA

Here we, will do some exploratory data analysis that gives us few good reviews, on how our data is, and what it means. So, let's ask few questions to ourselves for the dataset we have.

3.1 Is our data Continuous?

For that, let's add library called **DataExplorer** that will show us the how tidy our dataset is in matters of Columns, Rows and Missing values.

```r
library(DataExplorer)

## Warning: package 'DataExplorer' was built under R version 4.1.2

batch %>% plot_intro()
```

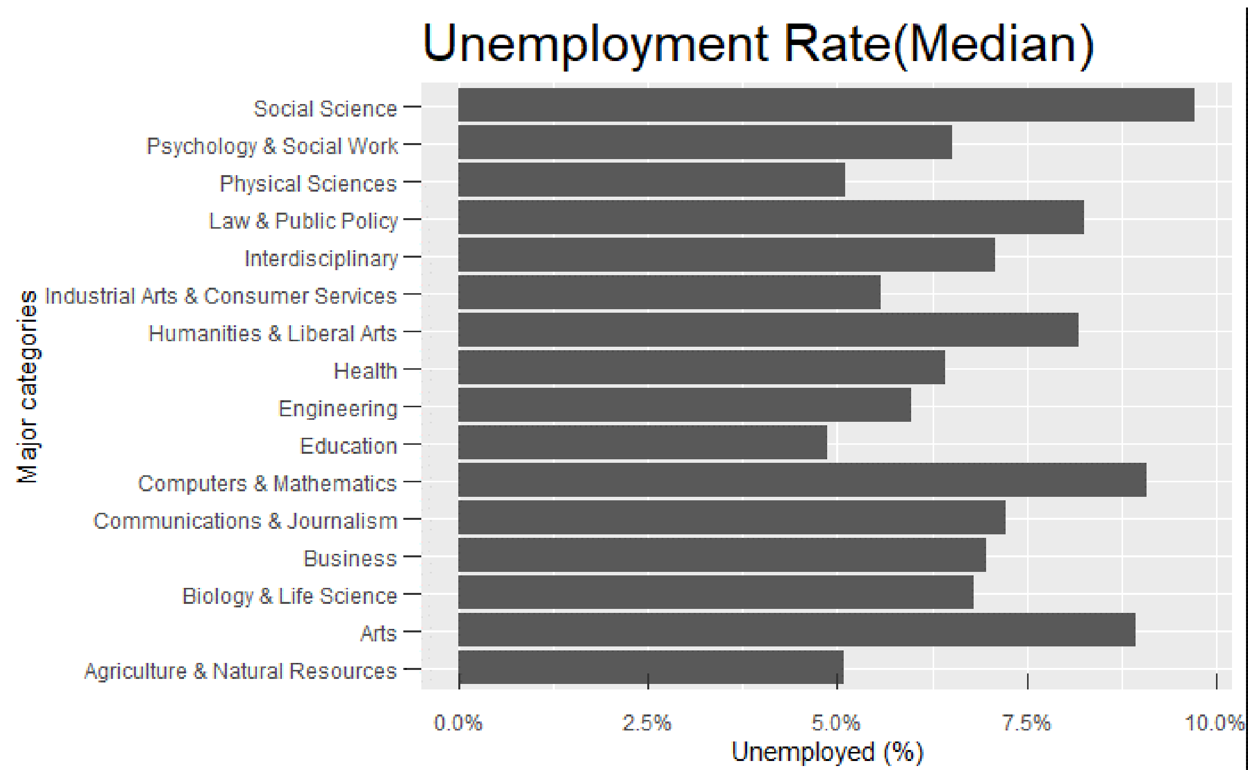


3.2 What is Unemployment Rate of different Majors?

For this we will get the information about the Median Unemployment rate for department major, from all the different department, regardless of their majors. Which will give us some of the insights to predict about the Department Major itself, that which one is better over another in the view for Employment.

```
(Unemp <- group_by(batch, Major_category) %>%
   summarise(Avg_Unemployed_Rate = median(Unemployment_rate)))

## # A tibble: 16 x 2
##    Major_category                          Avg_Unemployed_Rate
##    <chr>                                                 <dbl>
##  1 Agriculture & Natural Resources                      0.0509
##  2 Arts                                                 0.0895
##  3 Biology & Life Science                               0.0680
##  4 Business                                             0.0697
##  5 Communications & Journalism                          0.0722
##  6 Computers & Mathematics                              0.0908
##  7 Education                                            0.0488
##  8 Engineering                                          0.0598
##  9 Health                                               0.0643
## 10 Humanities & Liberal Arts                            0.0817
## 11 Industrial Arts & Consumer Services                  0.0557
## 12 Interdisciplinary                                    0.0709
```

```
## 13 Law & Public Policy                                    0.0825
## 14 Physical Sciences                                      0.0511
## 15 Psychology & Social Work                               0.0651
## 16 Social Science                                         0.0972

bar <- ggplot(data = Unemp)+
    geom_col(mapping = aes(x= Major_category, y = Avg_Unemployed_Rate))

bar + coord_flip() +
  theme(
    legend.box.background = element_rect(),
    legend.box.margin = margin(6, 6, 6, 6)
  ) +
  labs(
    title = "Unemployment Rate(Median)",
    x = "Major categories",
    y = "Unemployed (%)"
  ) +
  theme(plot.title = element_text(size = rel(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"))
  ) + scale_y_continuous(labels = scales::percent)
```

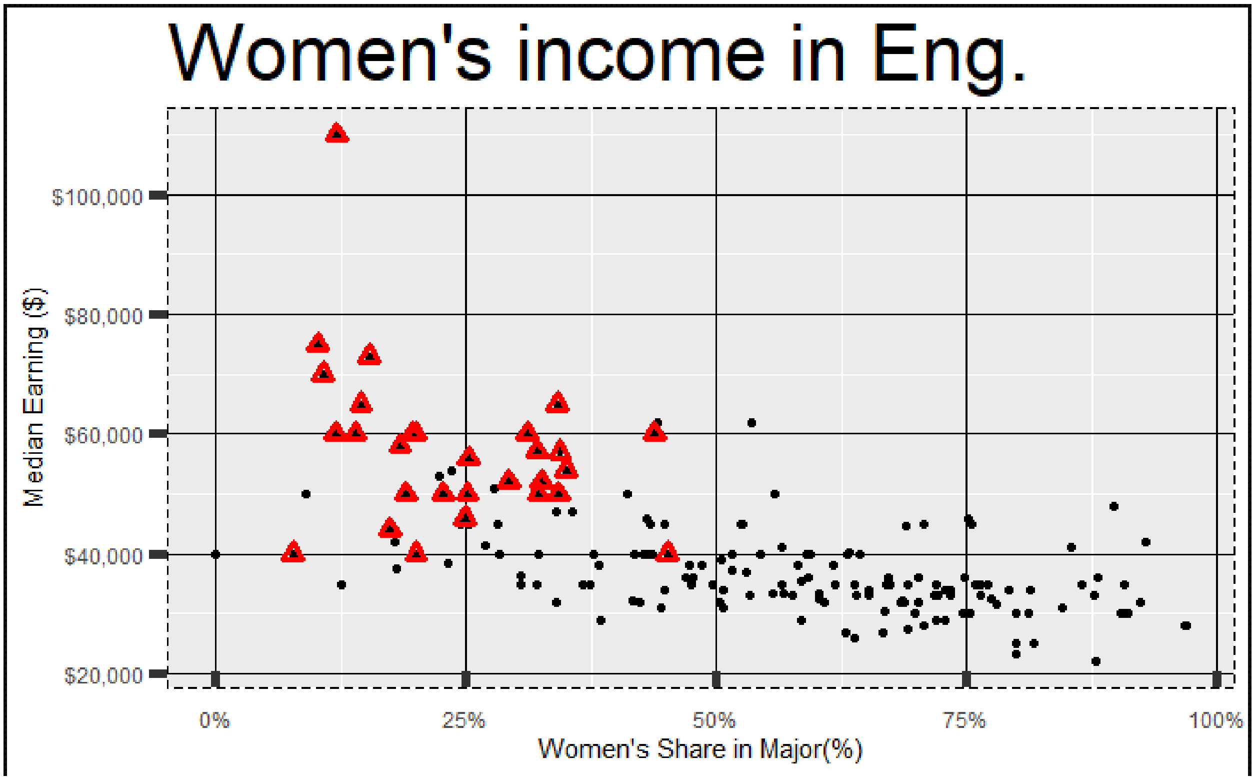


As we can see from the above graph, the median unemployment rate for the “**Social Science”, “Computer & Math”** and **“Arts” are very high.**

Over here we can see that there are only a few of the major where the unemployment rate Is very high above 7.5%. So, as we count its only 5 of the Major categories. But, just from this we can not predict or assume these fields from the major has the lowest income, we have to explore more to make a stand on this. So, now how about we calculate the income for the women in the field of the Engineering and can see, that does less unemployment means the higher Income here?

3.3 What is Women’s share in income from Engineering Department?

```
Eng <- filter(batch, Major_category == "Engineering")

w_income <- ggplot(data = recent_grade, mapping = aes(x = ShareWomen, y = Med
ian)) +
  geom_point(shape = 20, fill = NA , size = 2, stroke = 1 ) +
  geom_point(data = Eng, mapping = aes(x = ShareWomen, y = Median), color = '
red', shape = 24, stroke = 2) +
  labs(title = "Women's income in Eng.",
       x = "Women's Share in Major(%)",
       y = "Median Earning ($)"
      )

w_income +
  theme(plot.title = element_text(size = rel(3))) +
  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"))
  ) +

  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::dollar)

## Warning: Removed 1 rows containing missing values (geom_point).
```



As you can see from the Scatter plot, we can interpret that the major higher income for the women comes from Engineering department. So, we can stand on one of the statements from the EDA2 that Engineering shows the trend of higher salary income and less Unemployment rate, which are very good results. For the all-different Major, it may show the trends as this.

4. PCA (Principal Component Analysis)

First get the data on which we have to perform the PCA, because non numeric data is not allowed in this process, so lets take this variable from below as our new data on which we can perform the PCA.

- Total
- Mean
- Employed
- Unemployed
- College_jobs
- Non_college_jobs
- Low_wage_jobs

Now, let’s perform the PCA on this data to find out more thing. 1. Which are the component important to us? 2. For the Linear regression model how many components we can select to do the regression and all such.

Let’s see how to calculate the PCA from the scratch for this dataset.

- Calculate Correlation matrix.
- Calculate Scaled Covariance for later use of Principal component score.
- Get Eigen Values and Vectors from Correlation matrix.
- For Percent Variance apply below formula.

Now, see the formula for the Percent variance.

- $PercentVariance = EigenValues/sum(EigenValues)$

4.1 Calculate Correlation matrix

```
# Make another batch to work on PCA we only need numeric data.
batch <- select(new_recent_grade,
                Median,
                Total,
                Employed,
                Full_time,
                Unemployed,
                College_jobs,
                Non_college_jobs,
                Low_wage_jobs)

# Calculate the Correlation
cor <- cor(batch)

# Calculate the Scaled Covariance = correlation
scaled <- scale(batch)
ScaleCov <- cov(scaled)
ScaleCov

##                       Median      Total   Employed   Full_time Unemployed
## Median            1.00000000 -0.1067377 -0.1043987 -0.07903094 -0.1236223
## Total            -0.10673767  1.0000000  0.9962140  0.98933921  0.9747684
## Employed         -0.10439869  0.9962140  1.0000000  0.99583083  0.9688554
## Full_time        -0.07903094  0.9893392  0.9958308  1.00000000  0.9600422
## Unemployed       -0.12362234  0.9747684  0.9688554  0.96004220  1.0000000
## College_jobs     -0.04706015  0.8004648  0.7971931  0.77213457  0.7133619
## Non_college_jobs -0.17181510  0.9412471  0.9412360  0.93302113  0.9564672
## Low_wage_jobs    -0.20715284  0.9355096  0.9271223  0.90471364  0.9553251
##                  College_jobs Non_college_jobs Low_wage_jobs
## Median            -0.04706015       -0.1718151    -0.2071528
```

```
## Total              0.80046477        0.9412471     0.9355096
## Employed           0.79719311        0.9412360     0.9271223
## Full_time          0.77213457        0.9330211     0.9047136
## Unemployed         0.71336190        0.9564672     0.9553251
## College_jobs       1.00000000        0.6128772     0.6497199
## Non_college_jobs   0.61287716        1.0000000     0.9756995
## Low_wage_jobs      0.64971995        0.9756995     1.0000000
```

4.2 Get Eigen Vectors and values

```
eigenCor <- eigen(cor)
eigenCor
```

```
## eigen() decomposition
## $values
## [1] 6.3917353264 1.0052630585 0.4678831857 0.0847638817 0.0335580129
## [6] 0.0124231860 0.0035941464 0.0007792023
##
## $vectors
##             [,1]        [,2]         [,3]         [,4]        [,5]
[,6]
## [1,]  0.05936931  0.97867199  0.172292097  0.092944131  0.01665855  0.0071
90521
## [2,] -0.39368557  0.04732512 -0.030671002 -0.180798304 -0.05097168  0.3223
53344
## [3,] -0.39304171  0.05033561 -0.030703512 -0.303770915  0.13453056  0.1882
63467
## [4,] -0.38880325  0.07374196  0.002930248 -0.556934584  0.18175654  0.0987
38022
## [5,] -0.38816978  0.01155923  0.158226965  0.006146671 -0.84021144 -0.3286
84013
## [6,] -0.31286291  0.14070202 -0.859254731  0.315253721  0.06524716 -0.1861
58217
## [7,] -0.37943940 -0.05930844  0.359579723  0.168149513  0.48265673 -0.6602
78195
## [8,] -0.37855510 -0.09242162  0.275266032  0.654617074  0.05310841  0.5217
17291
##            [,7]         [,8]
## [1,]  0.0024738  0.004232850
## [2,] -0.8360659 -0.060226276
## [3,]  0.2620608  0.792179030
## [4,]  0.3770798 -0.590114946
## [5,]  0.1006053  0.002682685
## [6,]  0.0471314 -0.059014384
## [7,] -0.1598090  0.021741251
## [8,]  0.2285627 -0.128876055
```

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.

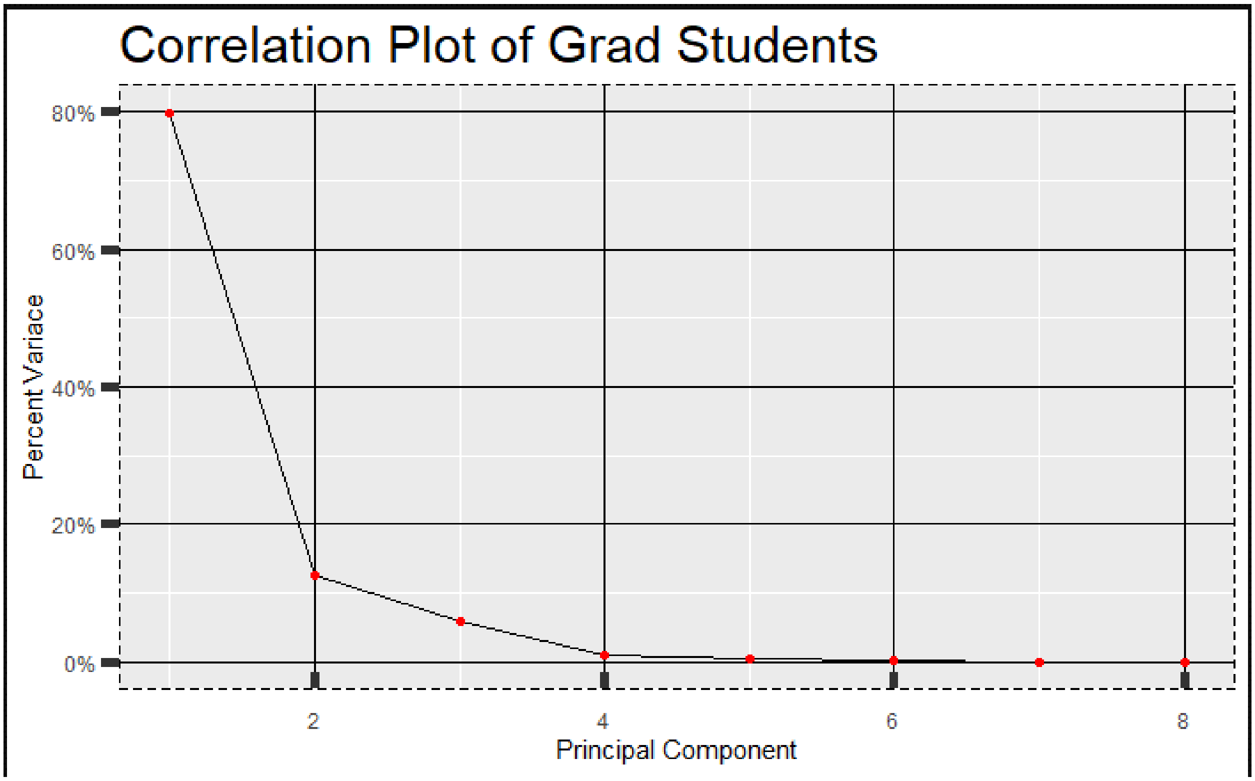
We can calculate Cumulative percent variance just to see that how many columns represents major portion of the data information.

```
# Percent Variance
PV <- eigenCor$values/sum(eigenCor$values)
PV

## [1] 7.989669e-01 1.256579e-01 5.848540e-02 1.059549e-02 4.194752e-03
## [6] 1.552898e-03 4.492683e-04 9.740029e-05

# Cumulative Percent Variance
cumsum(PV)

## [1] 0.7989669 0.9246248 0.9831102 0.9937057 0.9979004 0.9994533 0.9999026
## [8] 1.0000000
```

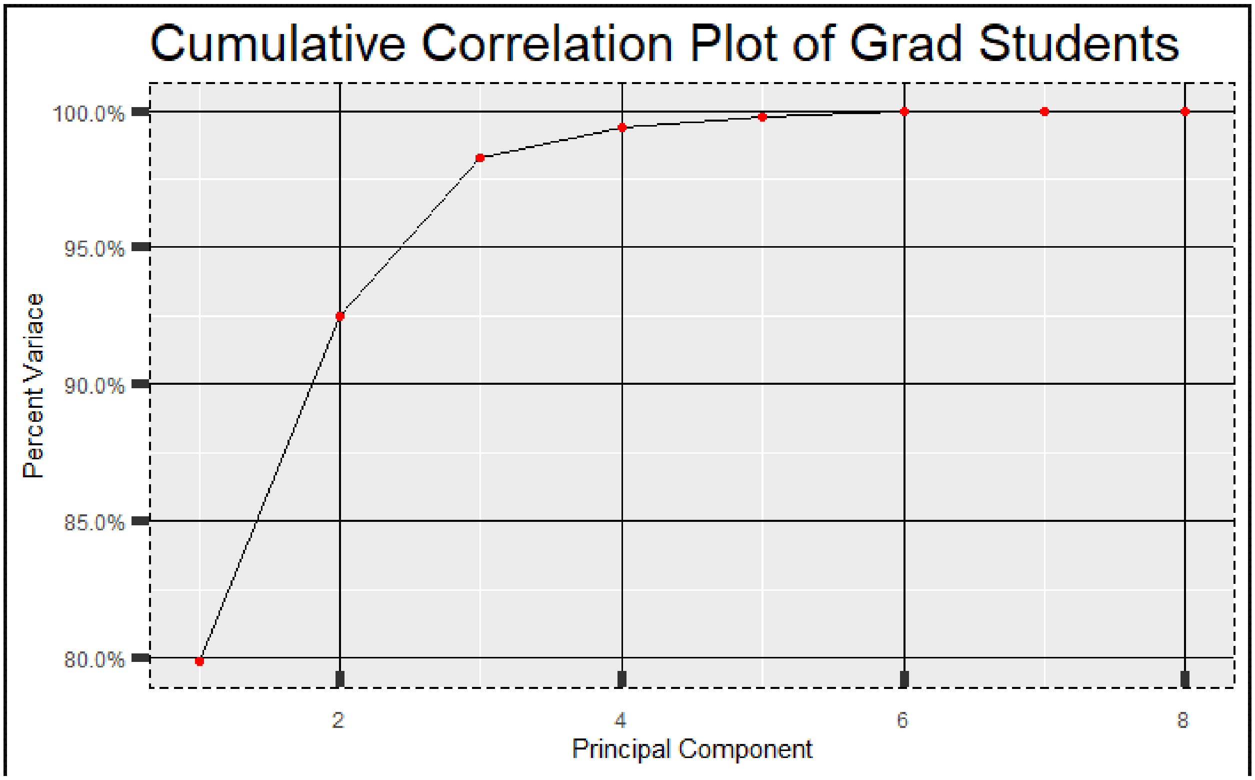
4.3 Plot the PV and CPV

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

```
SwCorPlot <- qplot(c(1:8),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 Grad Students") +
  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(1:8),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 Grad Students") +
  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 172 data points onto the first eigen vectors, the projected values are called the first principal component.

From above graph, we can say that. 1st component contains **79%** data, 2nd component contains **92%** and if we include 3rd component the total data will be **98%**. So no need to add more component, they have very negligent data available which does not bother us.

4.4 Principal Components Score.

```
selectedEigenValues <- eigenCor$vectors[,1:3]
colnames(selectedEigenValues) = c("PC1", "PC2", "PC3")
row.names(selectedEigenValues) = colnames(batch)
selectedEigenValues

##                        PC1         PC2          PC3
## Median          0.05936931  0.97867199  0.172292097
## Total          -0.39368557  0.04732512 -0.030671002
## Employed       -0.39304171  0.05033561 -0.030703512
## Full_time      -0.38880325  0.07374196  0.002930248
## Unemployed     -0.38816978  0.01155923  0.158226965
## College_jobs   -0.31286291  0.14070202 -0.859254731
## Non_college_jobs -0.37943940 -0.05930844  0.359579723
## Low_wage_jobs  -0.37855510 -0.09242162  0.275266032
```

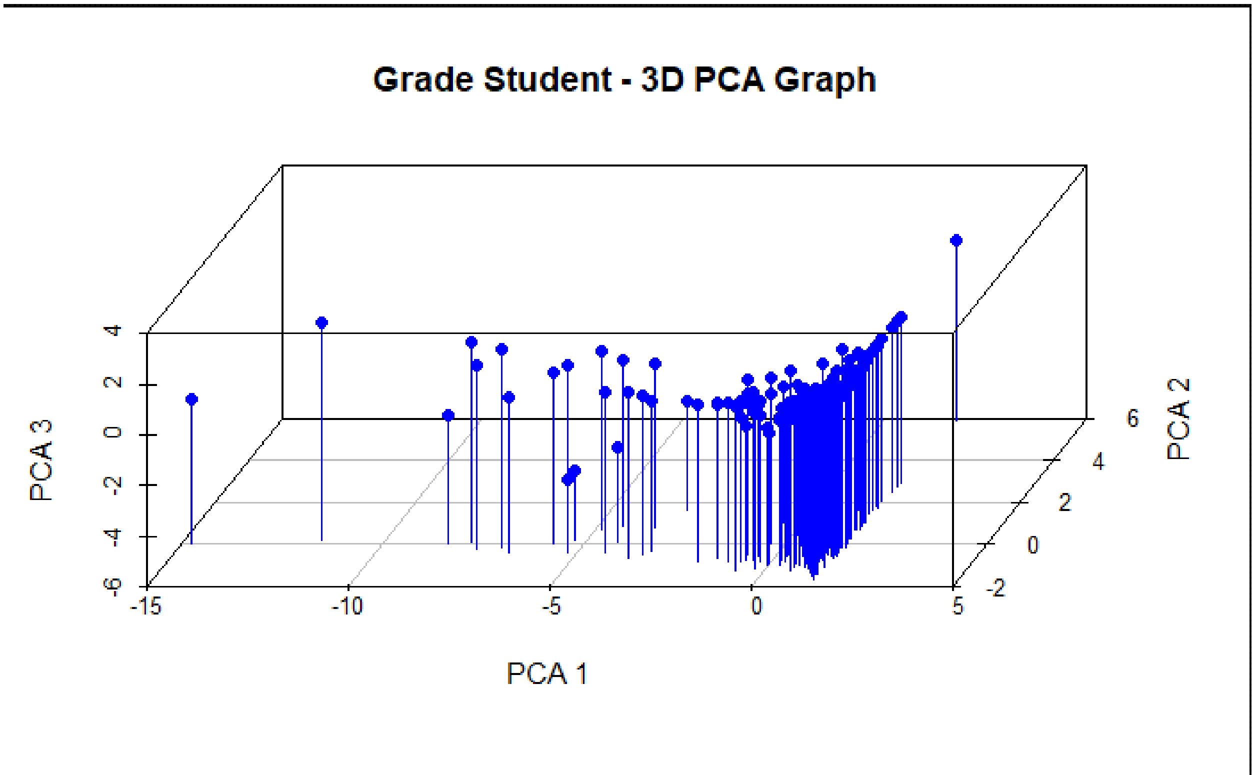
```r
# Principal component scores for batch data
PC1 <- as.matrix(scaled) %*% selectedEigenValues[,1]
PC2 <- as.matrix(scaled) %*% selectedEigenValues[,2]
PC3 <- as.matrix(scaled) %*% selectedEigenValues[,3]

# get it into one data frame and see the head
PC <- data.frame(PC1,PC2,PC3)
head(PC)
```

```
##          PC1       PC2        PC3
## 1  1.4176397 -1.651318 -0.1814043
## 2  1.2926365 -1.511237 -0.2257221
## 3  1.3784662 -1.381465 -0.1840436
## 4  1.3330857 -1.392210 -0.1301746
## 5  1.1817198 -1.288035 -0.1583238
## 6 -0.6796683 -1.269742  0.5038478
```

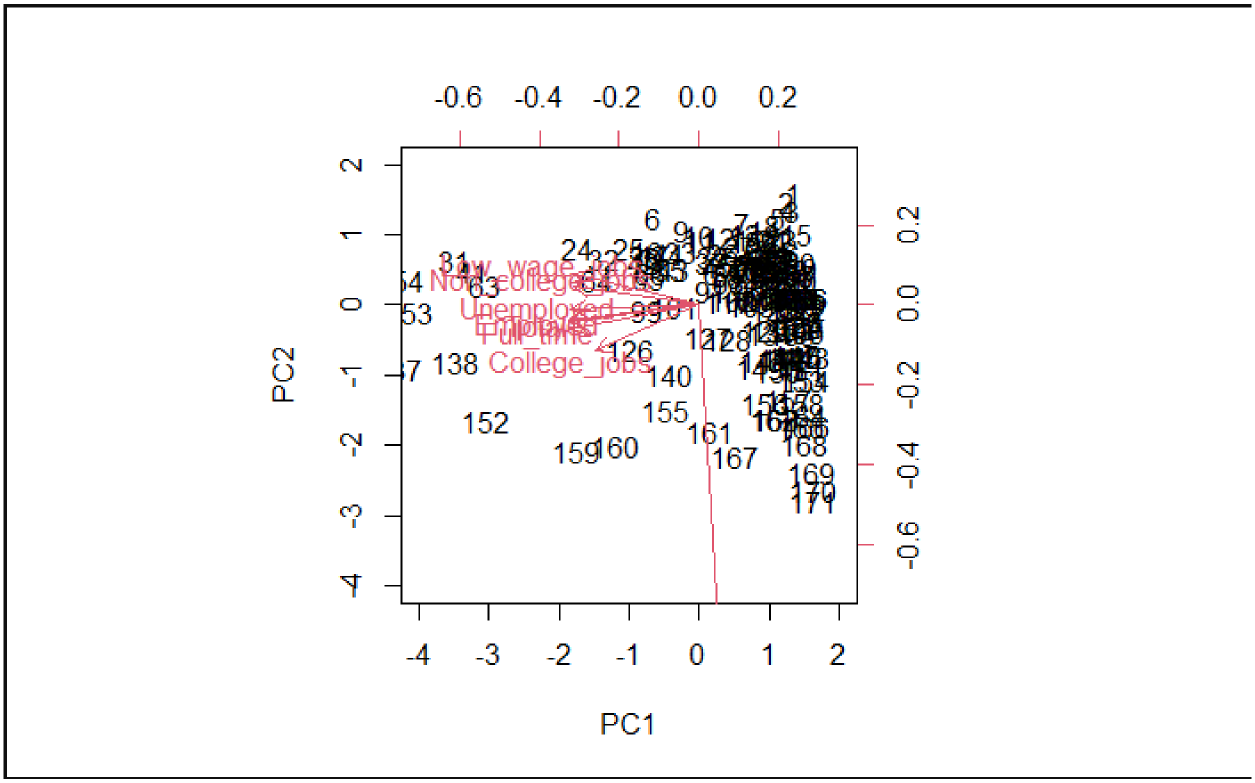
4.5 Plot the PCA

We, have selected three principal components so lets visualize them. * Visualize PC1, PC2 and PC3 together. * Visualize PC1 and PC2 only.

```r
library(scatterplot3d)

scatterplot3d(PC[,1:3], angle = 75, pch = 16,
              main = "Grade Student - 3D PCA Graph",
              xlab = "PCA 1",
              ylab = "PCA 2",
              zlab = "PCA 3",
              color = "Blue",
              type = "h"
              )
```




```r
# Calculate the biplot with the variable vectors
results <- prcomp(batch, scale = TRUE)
biplot(results, scale = 0.01,  expand=1, xlim=c(-3.0, 2.0), ylim=c(-4.0, 2.0)
)
```



By looking at the whole process, and plotting we conclude that we can take two Principal component for our further study of this dataset.

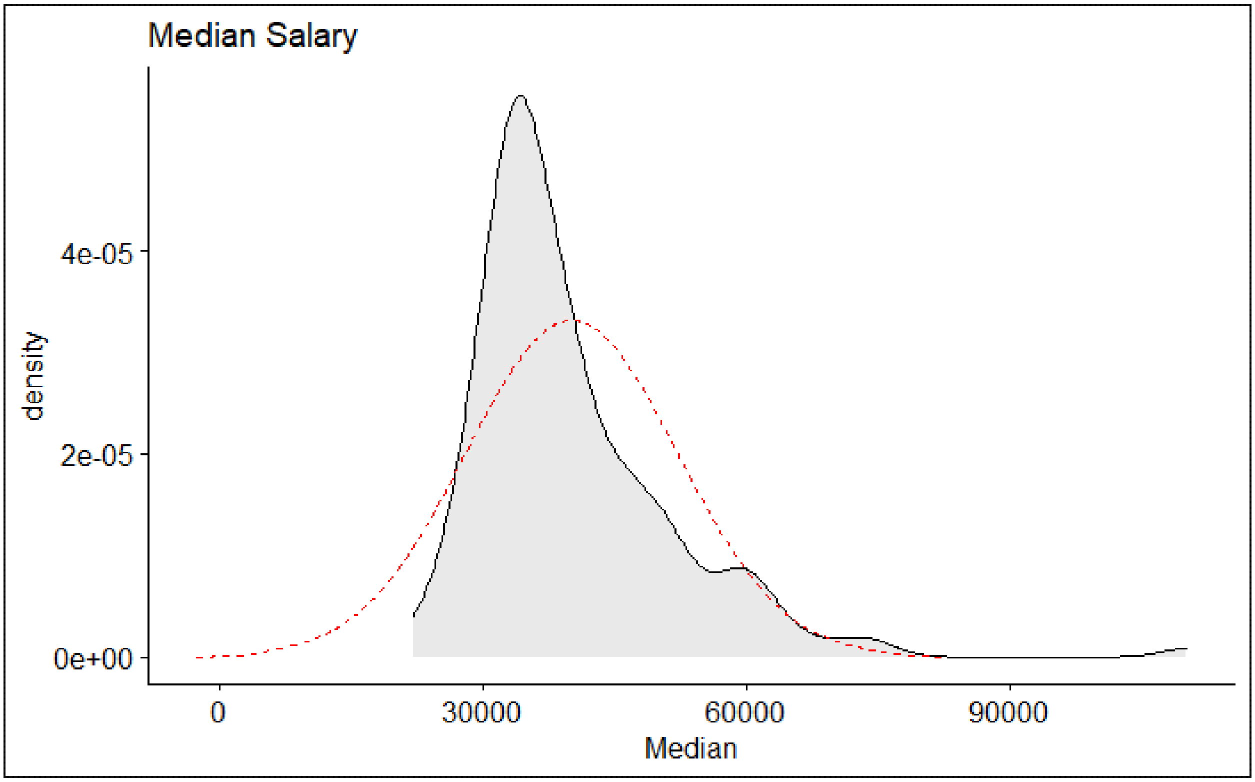
5. Linear Regression

Linear regression is the next step up after correlation matrix calculation, which we did for the calculation of the PCA. It is used when we want to predict the value of a variable based on the value of another variable. The variable we want to predict is called the dependent variable (or sometimes, the outcome variable).

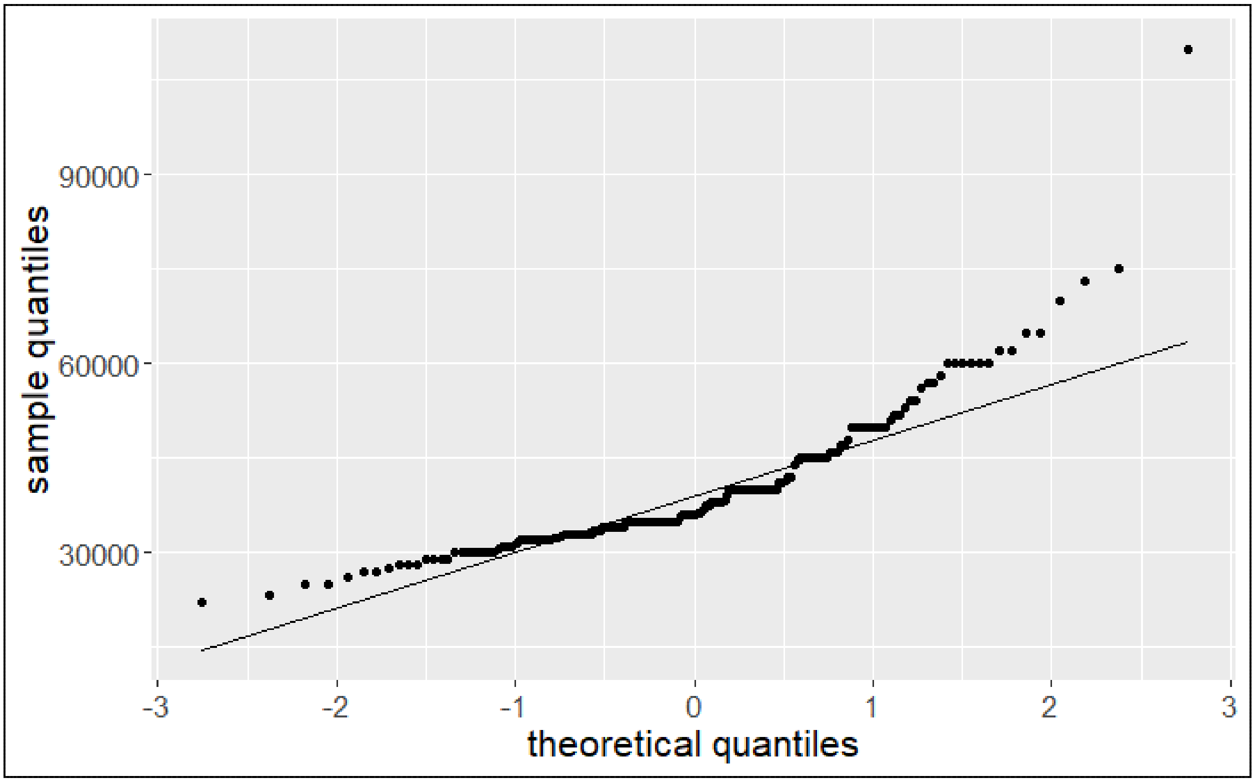
Now, before going to the regression part lets ask few things to ourselves, and check the basic things before going to the regression part. By doing the regression, we want to fit the model for predicting the Income.

5.1 Check the Histogram of response variable

```
library(ggpubr)

## Warning: package 'ggpubr' was built under R version 4.1.2

library(moments)
# Distribution of Median variable
ggdensity(batch, x = "Median", fill = "lightgray", title = "Median Salary") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```




```
Fertility <- ggplot(data = batch, aes(sample = Median))

Fertility +
  stat_qq(distribution = stats::qnorm) + stat_qq_line() +
  labs(y = 'sample quantiles', x = 'theoretical quantiles') +
  theme(text = element_text(size = 16))
```



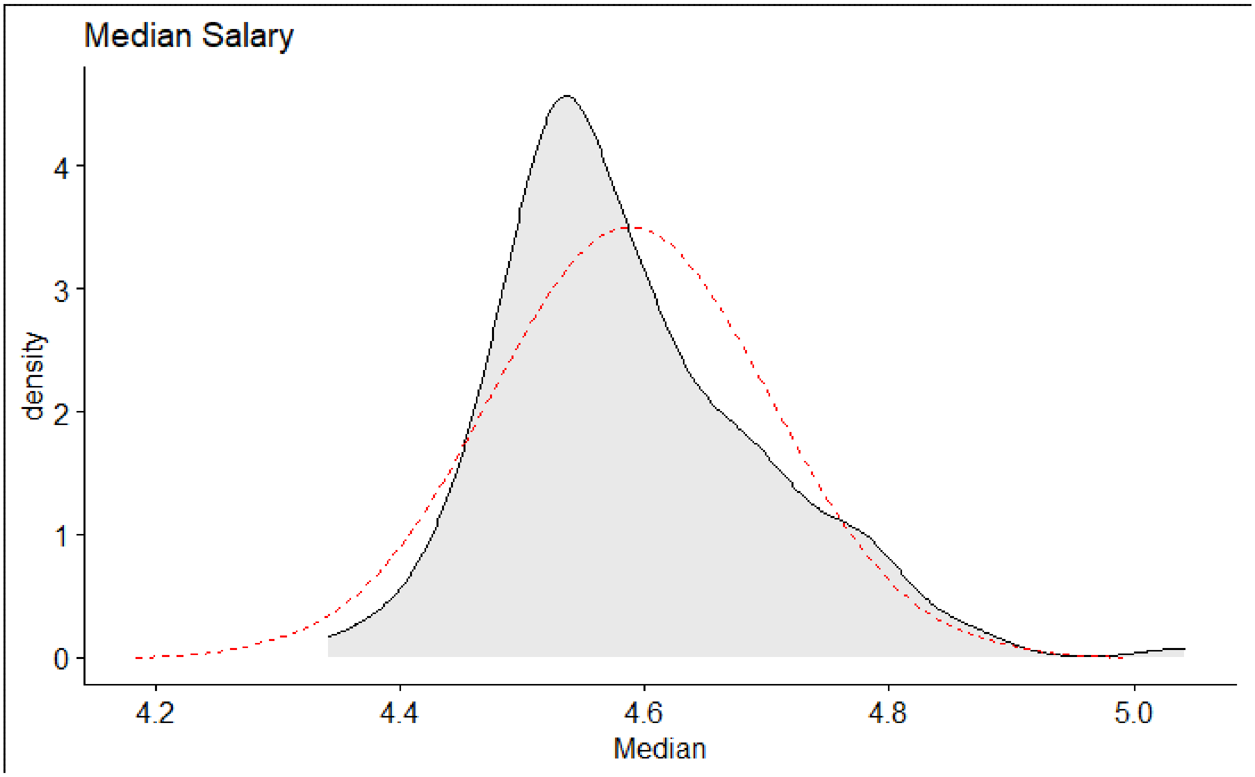

```
# Check the skewness
skewness(batch$Median, na.rm = TRUE)

## [1] 2.047032
```

Skewness is a measure of symmetry for a distribution. The value can be positive, negative or undefined. In a skewed distribution, the central tendency measures (mean, median, mode) will not be equal. Which you can see our here.

Generally, when **Mode < Median < Mean** we can call our graphs as Positively skewed. The most frequent values are low; tail is toward the high values (on the right-hand side).

And as we saw the value of the skewness is 2.047032. So, we can say that this value is high. Now, in a trial to transform them to the normal distribution. we have to apply the log(x).

```
batch$Median = log10(batch$Median)

# Log Distribution of Median variable
ggdensity(batch, x = "Median", fill = "lightgray", title = "Median Salary") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")
```




```
# Check the skewness of the transformed data.
skewness(batch$Median, na.rm = TRUE)

## [1] 0.8487541
```

As we can see over here is our skewness is decreased and that number looks like 0.8487541. Which is very less then 2. So, this transformed data helps us to train the data more efficiently because it transformed under the bell curve of the Normal Distribution.

Note that transformation makes the interpretation of the analysis much more difficult. For example, if you run a t-test for comparing the mean of two groups after transforming the data, you cannot simply say that there is a difference in the two groups' means. Now, you have the added step of interpreting the fact that the difference is based on the log transformation. For this reason, transformations are usually avoided unless necessary for the analysis to be valid.

So, for the Validation values, whenever you get the results and you want to interpret it into the real values from the Original Distribution, then you might need to take the Anti log of the data.

5.2 Fit the model

```
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

temp <- PC
temp = cbind(temp,Salary = batch$Median)
#head(temp)
modelPC <- lm(Salary ~ .,data=temp)
summary(modelPC)

##
## Call:
## lm(formula = Salary ~ ., data = temp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.189029 -0.004738  0.003335  0.014539  0.020629
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 4.5883714  0.0017107 2682.155  < 2e-16 ***
## PC1         0.0058482  0.0006786    8.618 4.84e-15 ***
## PC2         0.1042671  0.0017112   60.932  < 2e-16 ***
## PC3         0.0172294  0.0025083    6.869 1.20e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02244 on 168 degrees of freedom
## Multiple R-squared:  0.958,  Adjusted R-squared:  0.9573
## F-statistic:  1278 on 3 and 168 DF,  p-value: < 2.2e-16
```

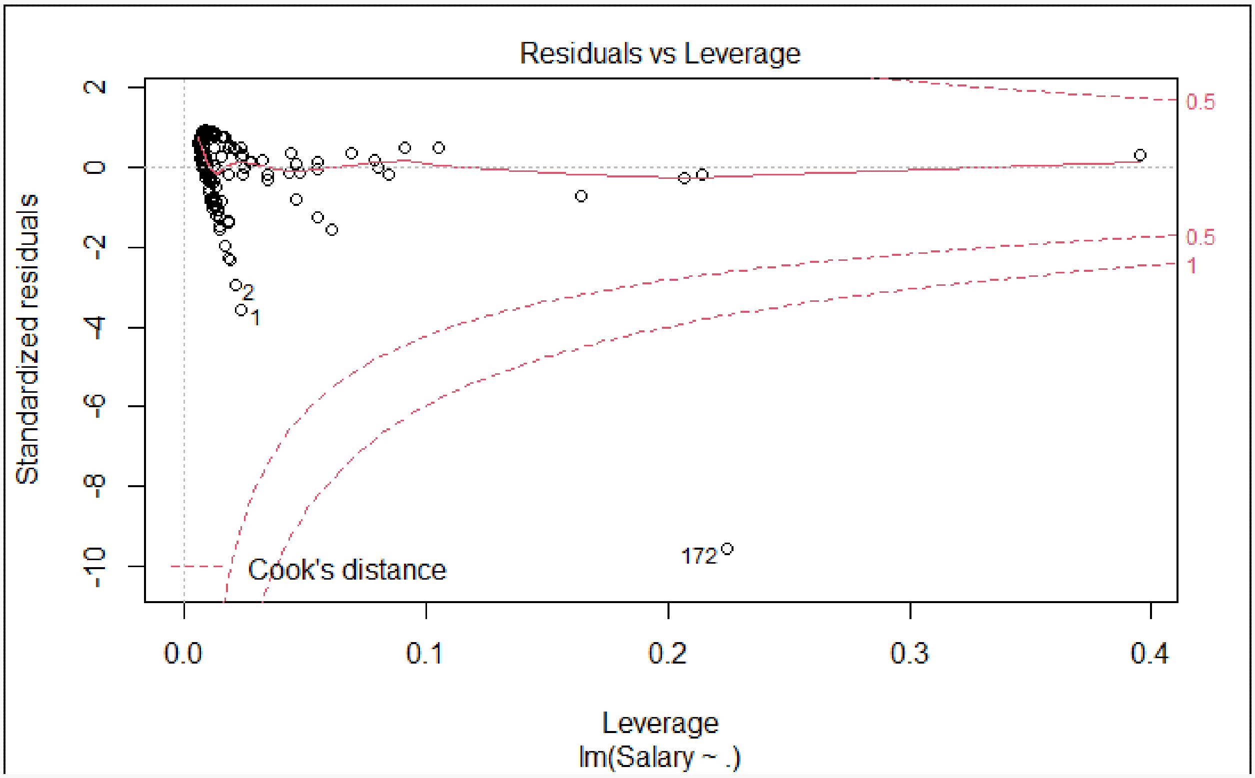
From the above values we can say that our model is statistically different from zero (p-value < 0.05) which was our null hypothesis, because the values for all four columns used for prediction (PC1, PC2, PC3) is less than 0.05. The Intercept also has a p-value less than 0.05 and hence is significant.

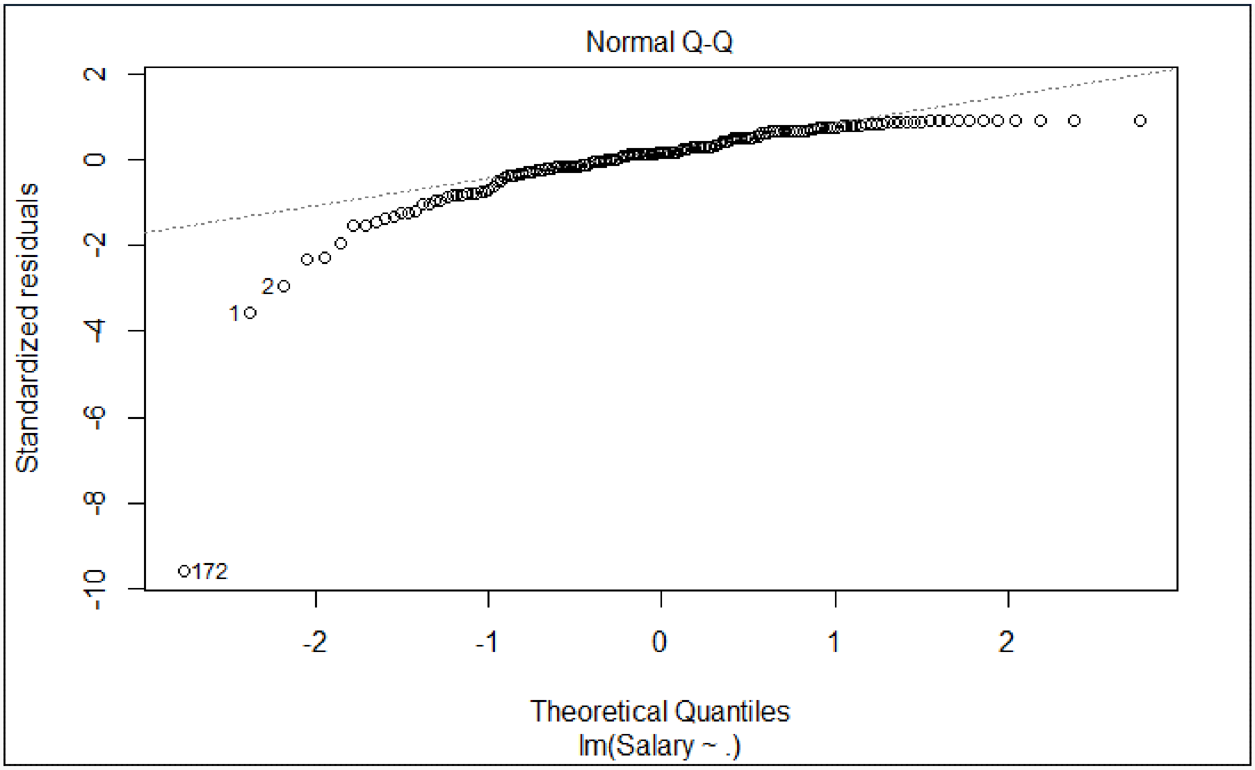
Moreover, our model is predictive since Adjusted R-squared is 0.9573 which is greater than 0.95. Thus our model is able to explain the variance to a very high degree.

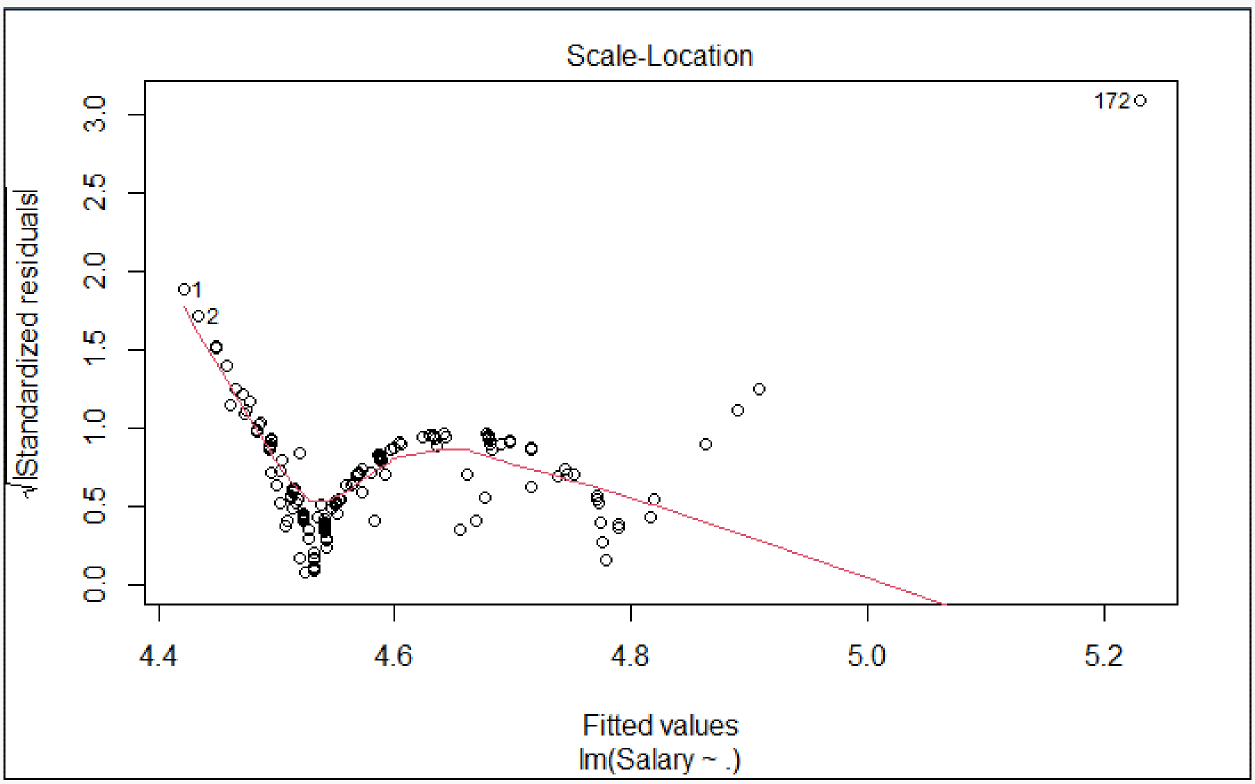
In addition, our Residual Standard error should be near to Zero, and in our case it is 0.02244, which is fantastic. Mean in our residual vs Fitted plot, we might get the best fit. The lesser the error the greater the model.

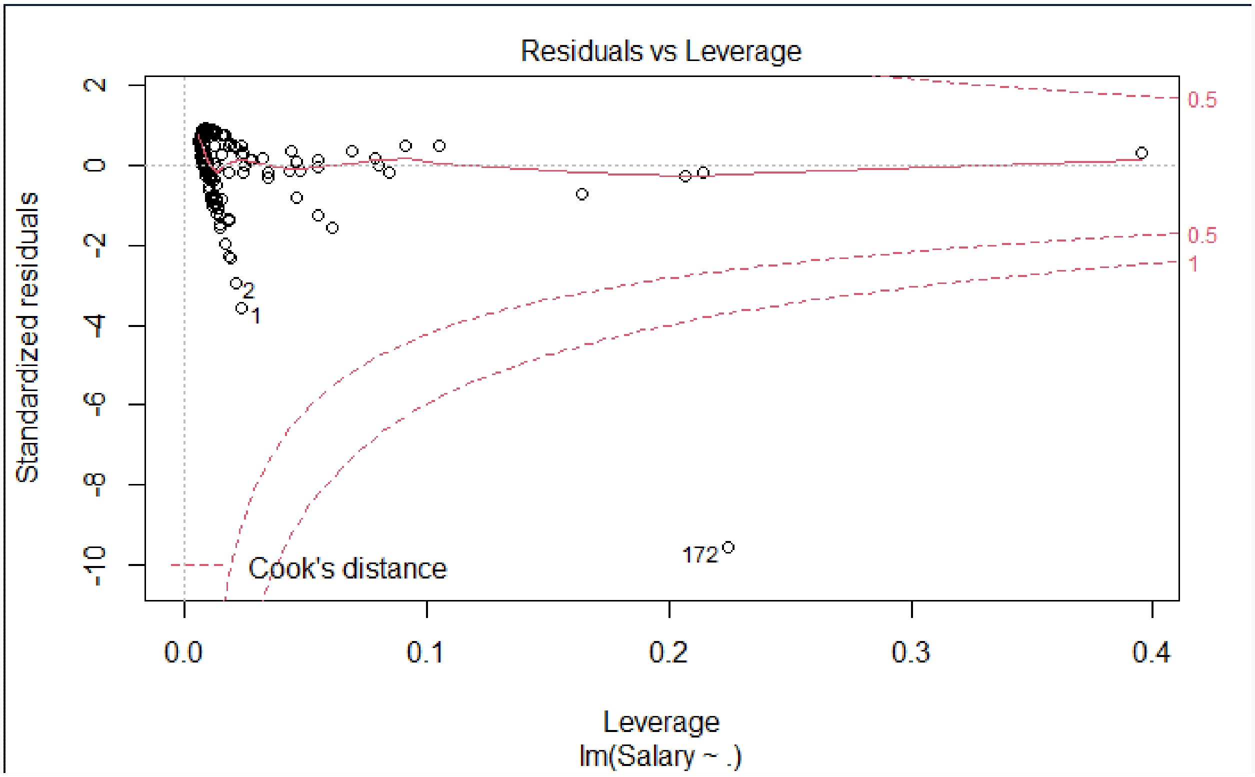
5.3 Residual Analysis

```
plot(modelPC)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Salary ~ .)
172
1
2


Scale-Location
√|Standardized residuals|
Fitted values
lm(Salary ~ .)
172
1
2



Summary

For the over finding of this project is, PCA is really so powerful tool to do the dimensional reduction. So that we can interpret the generated the PCA to predict the linear regression model. In the starting we had 8 different variables, which is really so hard to handle, PCA help us to find the important dimensions, and we were able to use the that components to a liner regression.

Another finding from this project is that if you have really skewed data, you're training and testing effect there for their accuracy, because the sample data taken from the distribution does not perfectly resembles the Normal distribution. So, Transforming the data is one of the best ways to get the good results, after getting the bell curve of that dataset.

For the further in future exploration, we can implement the PCR (Principal Component Regression), this kind of model use the PCA internally to generate the model. This model used when our data has the Multicollinearity, it's very had to relay on the P-values when all the component are very corelated to each other. For us, we faced the same condition in the correlation, that our variables look so correlated, at this kind of time, we can check the VIF (Variance Inflation Factor). If that factor is >5 then the variables are very correlated.

That was the limitation or the challenge for me to implement. But, certainly this is in my bucket list to explore, because we come across the Multicollinearity more often then we think.