GEORGE MASON UNIVERSITY

STAT 515

FINAL DATA MODELLING

&

VISUALIZATION REPORT

GROUP 10

BY

ABHISHEK SHAMBHU (Linear Regression)

AAKANKSHA AUNDHKAR (Random Forest)

JINYI WU (Lasso Regression)

12-10-2018

Multiple Linear Regression Model on Medical Cost Personal Dataset in R Shiny

Introduction:

Insurance companies nowadays are very much curious in predicting the insurance costs considering the various investment factors involved. Due to the increasing demand in predicting the appropriate cost of insurance, building models on this type of dataset will be very much useful and will definitely help insurance companies to take a second opinion and help predict the cost. Moreover, various types of visualizations will also help to better explore the dataset and predict more accurate results. It will also help us to answer the buzz question "Can you accurate predict insurance costs?" and the answer is "Yes". The model prediction in Shiny will give the business user the insurance costs value for the parameter considered and will give the result.

Regarding Shiny:

Shiny is an open - source R package that provides an elegant and powerful UI framework for building web-based applications in R. In my application, I have used Shiny package to make predictions on the new data based on significant parameters and provide predicted insurance cost value as output for the business. I have also showed the summary of the dataset, the dataset table, diagnostic plots, cross-validation plots, and various types of correlation and ggplots such as Scatter plots, Box plots, Frequency plots and Actual/ Predicted plots.

Dataset: (As seen in Shiny)

I had used the Medical Cost Personal Dataset available on Kaggle. My dataset consists of 1338 rows of 7 variables and size of 55 KB. My variable columns include Charges which is the dependent variable or target / response variable and rest of the variables (Age, Sex, BMI, Children, Smoker, Region) as the independent variables or the predictor variables.

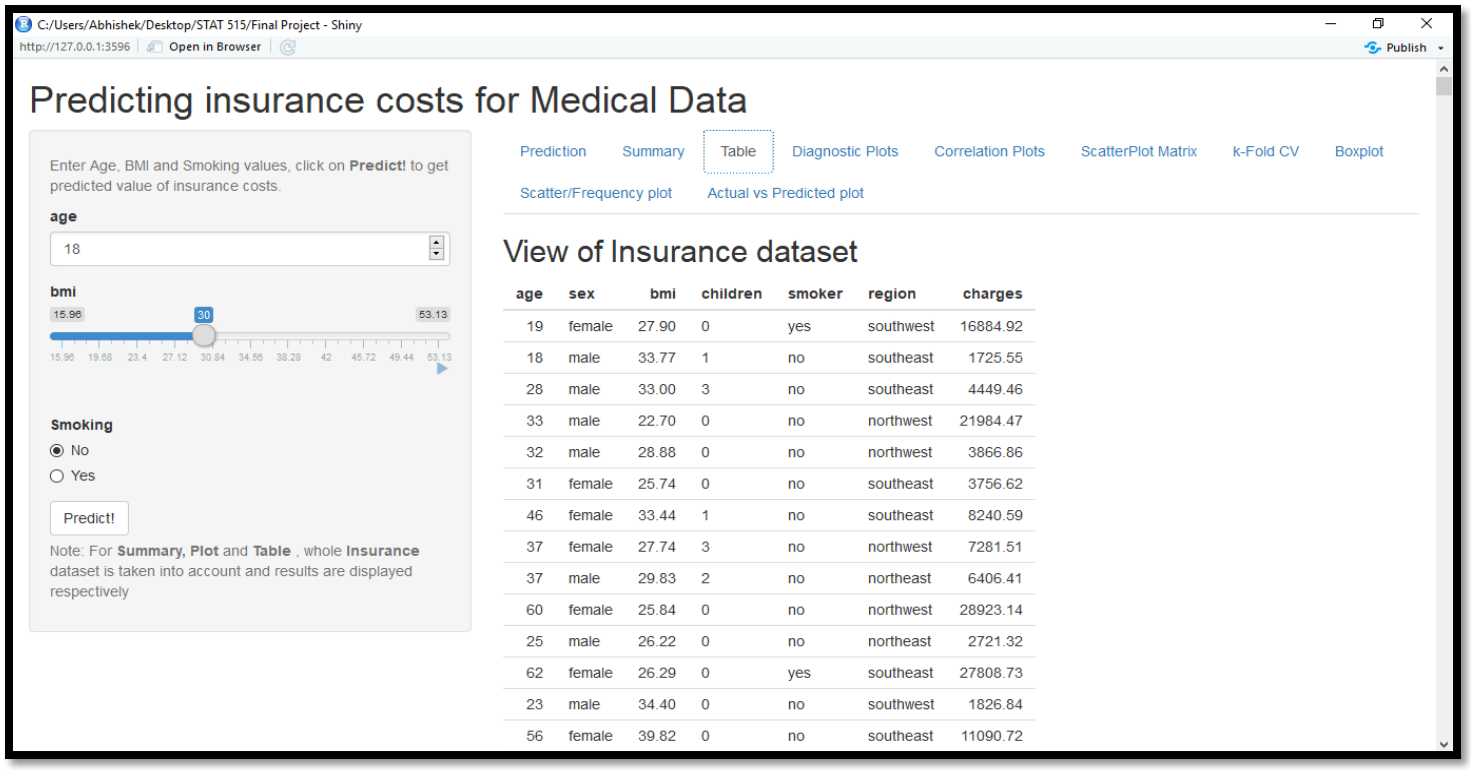


Fig. 1 – Insurance Data set as seen in RShiny Interface

Dataset Analysis:

Dataset analysis is the most important factor while studying the data and needs to be performed in order to understand the nature of the dataset at first glance. Having a look at the dataset using the Summary and the Structure command will give a more detail analysis on the data. The summary of the dataset will let us know the various variables available in the dataset and whether they are categorical or continuous variables. Also, it helps us know the distribution in the data and if there are any NA values present. For our dataset below, we can see that there are no NA values present and for the categorical columns such as Sex, Children, Smoker and Region, it gives us the count.

```
> summary(insurance)
      age            sex            bmi          children smoker           region       charges
 Min.   :18.0   female:662   Min.   :16.0   0:574    no :1064   northeast:324   Min.   : 1122
 1st Qu.:27.0   male  :676   1st Qu.:26.3   1:324    yes: 274   northwest:325   1st Qu.: 4740
 Median :39.0                Median :30.4   2:240               southeast:364   Median : 9382
 Mean   :39.2                Mean   :30.7   3:157               southwest:325   Mean   :13270
 3rd Qu.:51.0                3rd Qu.:34.7   4: 25                               3rd Qu.:16640
 Max.   :64.0                Max.   :53.1   5: 18                               Max.   :63770
>
```

Fig. 2 – Summary of Insurance Dataset

Structure as seen below gives us the class of the dataset. The number of rows and columns in the dataset, datatype and initial values for the columns present in the dataset.

```
> str(insurance)
'data.frame':   1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
 $ children: Factor w/ 6 levels "0","1","2","3",..: 1 2 4 1 1 1 2 4 3 1 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
 $ charges : num  16885 1726 4449 21984 3867 ...
>
```

Fig. 3 – Structure of the Insurance Dataset.

Dataset Exploration:

Firstly, as my dataset had lots of columns whose datatype is factor, I have converted the factor data to numeric for my analysis. Secondly, I have done splitting of the data into 70% train data and 30% test data for predicting on the dataset. I will be performing Multiple Linear Regression Analysis on the dataset to predict the cost. Thirdly, performed building model using train data based on significant variables and predicting the insurance cost on the test data and finally visualizing the data using various plots and graphs.

Visualization Analysis using Shiny:

1. Diagnostic Plots

The diagnostic plots help us to view the residuals in the dataset that will help us know if the model works well in the data. This visualization will not only will help us in building the model but also to improve it in a better way. The diagnostic plots help us to diagnose the data in 4 different ways.

1. Residual vs Fitted
   → This plot helps us to see if there are any non-linear pattern in the data. There can be a non-linear relationship between the outcome variables and the predictors and the pattern can be seen in the plot. If the data had equally spread residuals around a horizontal line without any pattern, then it will indicate us that we don't have non-linear relationships. Our plot below, since it's a red straight line is a good example.
2. Normal Q-Q
   →With this plot we can see if the residuals are normally distributed. In our plot, we can see that residuals follow the dotted plot for a large amount of data, but its tails deviate from the dotted line.
3. Scale-Location
   →This plot tells us about the spread of the data. It can be seen from the below graph, that the data is fairly spread along the range of predictors. However, the red line bends at the left tail which because of data being not equally spread at that point.
4. Residuals vs Leverage
   →This plot helps us to see if there are any outliers or influential cases which might had not been seen during analysis. In our plot, we can hardly see the Cook's distance line since all the cases are pretty well inside and there are no influential cases.

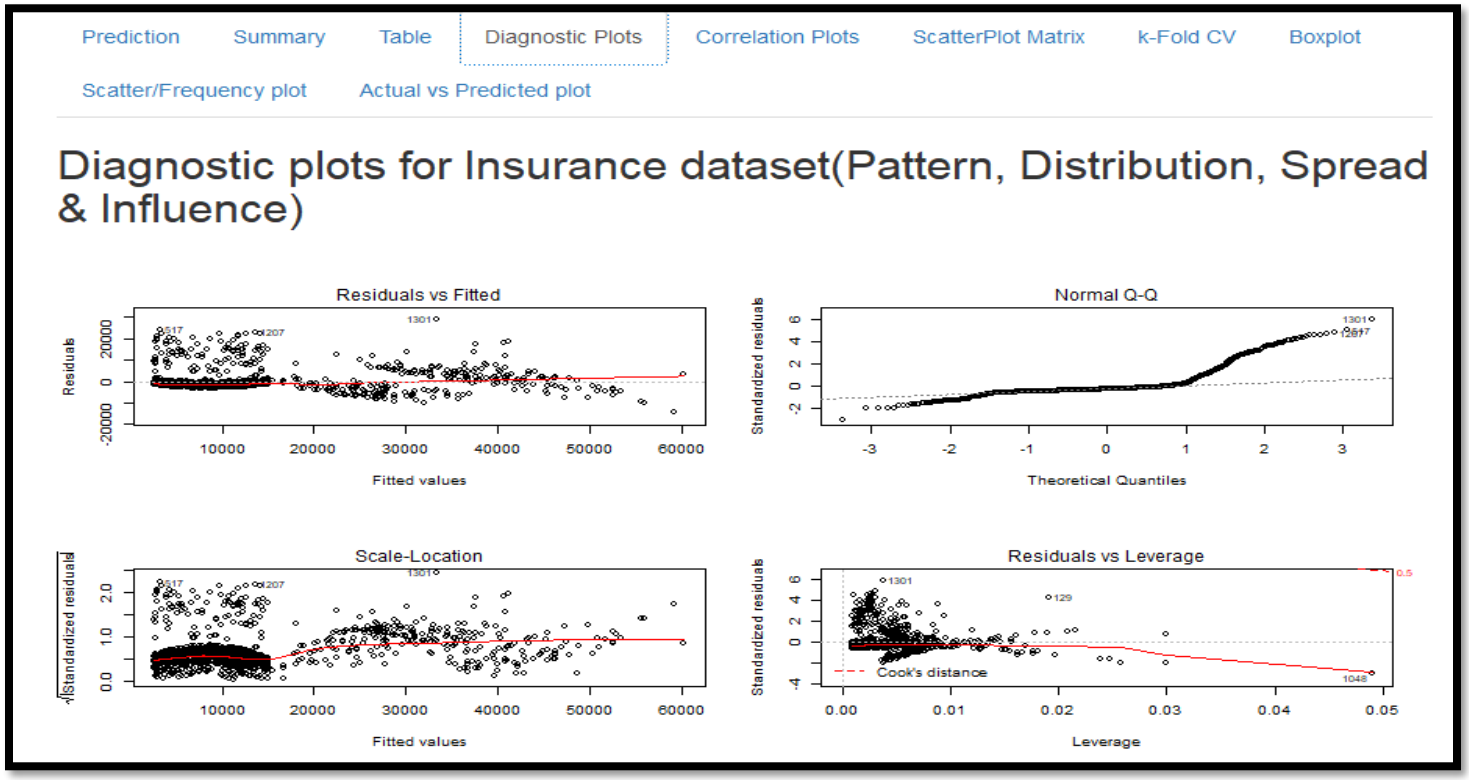


Fig. 4 – Diagnostic Plots

2. **Correlation Plots**

Correlation plots helps us to visualize relationship of one variable with all the other variables in the data. The higher the relationship between the variables, the higher is the correlation. By analysing the plot below, we can see that there are few parameters which are correlated and might show a stronger relationship when building the model.

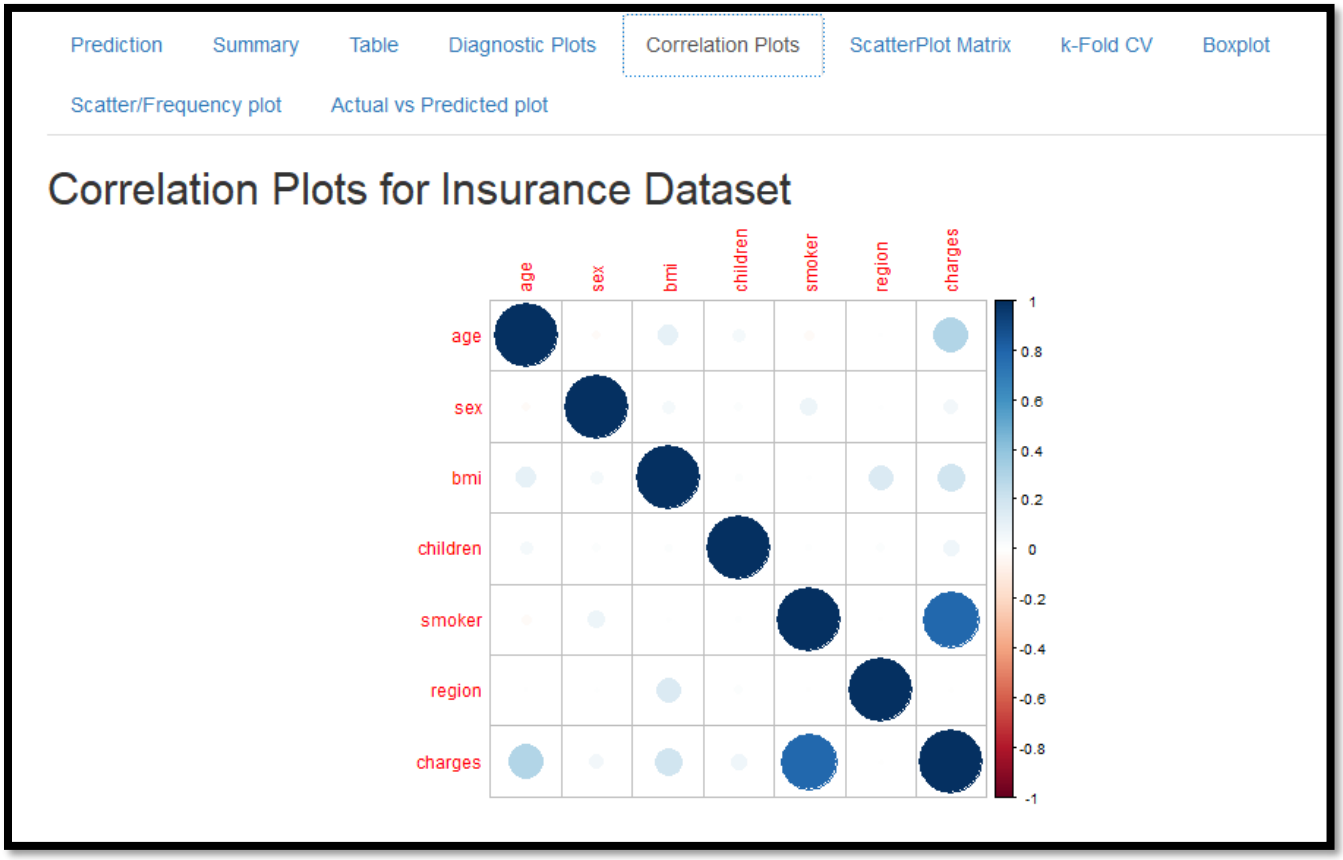


Fig 5- Correlation Plot

3. **Scatterplots and Frequency Plots**

→Scatterplots helps us to look out for relationships and frequency plots help us to have a count of the number of times the value is repeated. In our dataset, for analysis, I tried to visualize few scatterplots and frequency plots. In the first plot, we can see that as the age increases, the insurance charge too shows an increase. So, there is an upward trend in the best fit line. In the second plot, we can see that for BMI<30, the charges are somewhat on the lower side. However, for BMI>30, the charges vary a lot. In the third plot, we can see a scatterplot between number of children and charges. It can be analysed that lower number of children are more likely to incur charges whereas high number of children are less likely to incur. The final plot tells us the distribution of charges and its frequency. It can be seen that most of the insurance cost charges are low and very few were high.

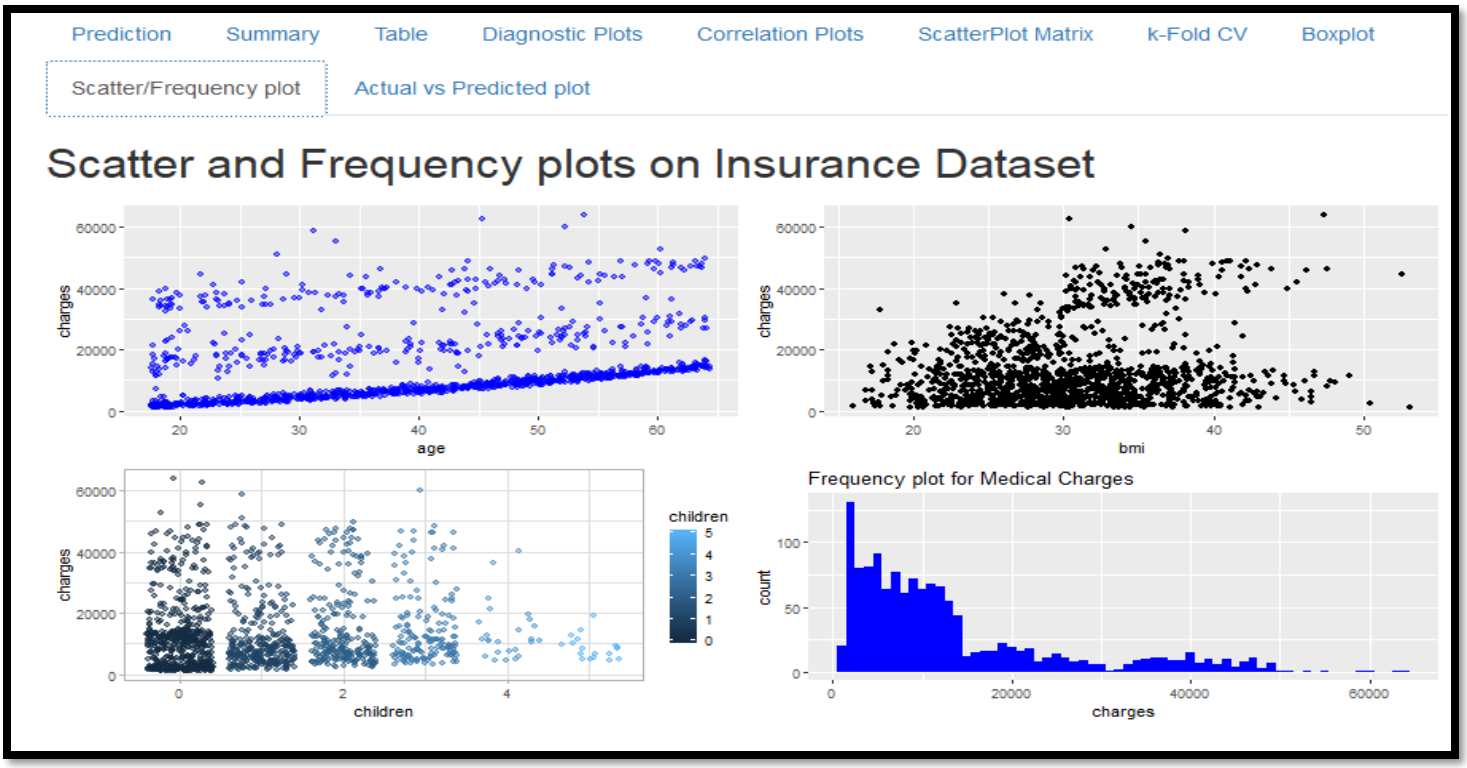


Fig 6 – Scatter and Frequency plot

4. **Boxplots**

→Boxplots helps us to visualize data and see if there are any outliers. Also, it helps us to see the distribution of the data across the various variable value. Even thought the first boxplot didn't showed any variation, we can see in the second boxplot that charges differ a lot for smoking equal to 'no' and smoking equal to 'yes'. In the third boxplot of charges vs children, we can see that number of children equal to five incur less charges. The fourth boxplot shows charges vs the region however there is not much of difference in the variable values.

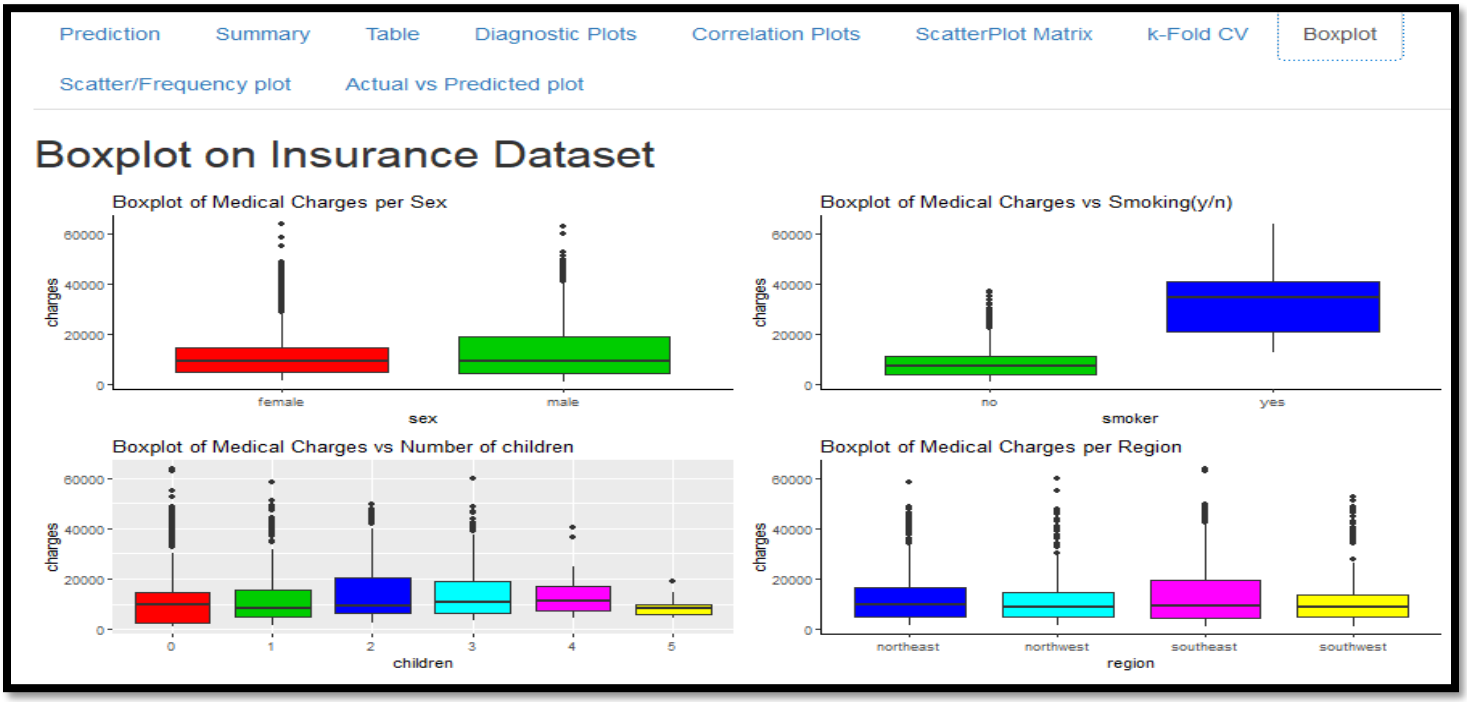


Fig 7 - Boxplots

Building Multiple Linear Regression Model:

The first step which involves in building the multiple linear regression model is to split the data into Train data and Test data. In my dataset, I have performed splitting keeping 70% as Train data and 30% as Test data. Then, I have performed a series of different combinations of Linear Regression Model to see which model gives the best accuracy and with least number of terms. A summary of the model which gives the highest adjusted R^2 value for my dataset can be found below. Here, for this model I have achieved a R square value of 0.836 which means that the model explains 83.6% of the variable predictions correctly. Moreover, since the p-value is very less, the model is highly significant and can be used for predicting values for a newer data.

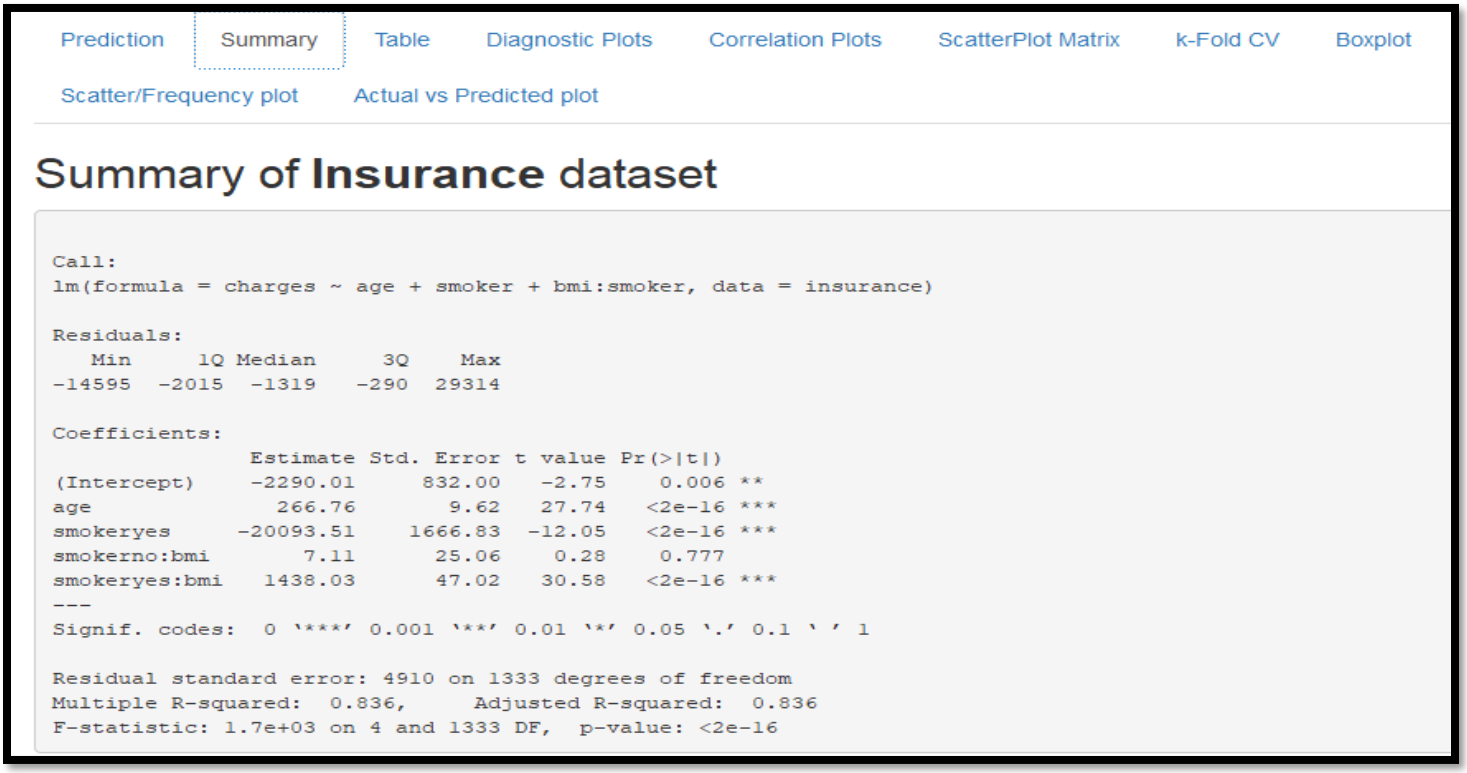


Fig 8- Summary of the Linear Regression Model

Making Predictions:

Here, I am using the model's significant variables as input to predict the insurance charges. This predicted Insurance costs charges will be somewhat similar to the actual charges which will be charged to the customer. Here, my Linear Regression created will accept the variable values as input and predict the cost based on the input values. Below, is a small example of my prediction on the data. Example, if age is 15, BMI as 30 and smoking as no, the insurance cost the customer would be charged would be somewhere near to 2725.

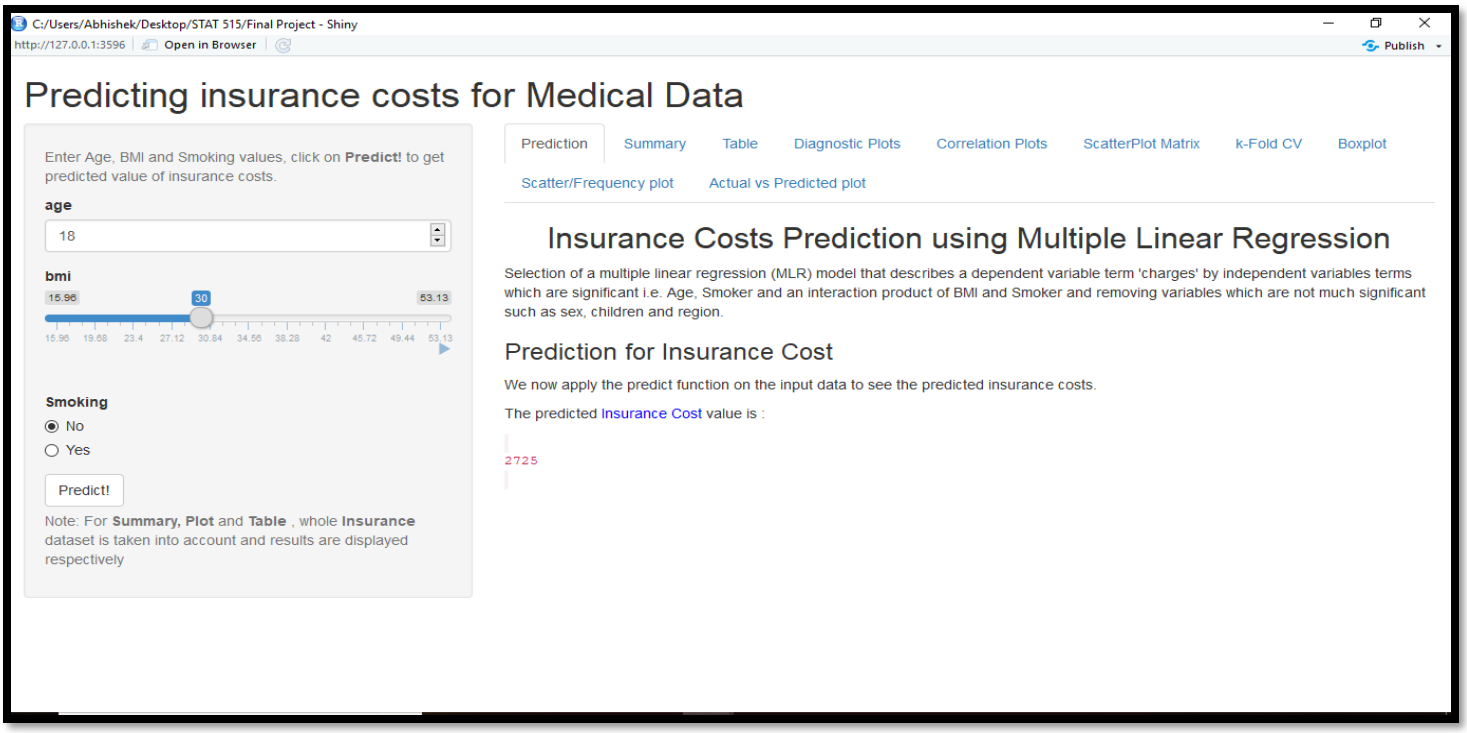


Fig 9 – Prediction of insurance cost in Shiny Interface

Making Cross Validation:

Sometimes it happens that data splitting provides an unbiased estimate of the test error; however, it can be possible that the data is biased and was unnoticed. So, we need to do Cross Validation on the data with multiple folds. In cross validation it happens that the data is equally partitioned into k-folds of equal sizes and the prediction error is calculated for each fold. The model's performance is calculated by taking average of error across the different test sets. In my model, I have performed 2-fold cross validation and visualized it in Shiny, it can be seen that the actual and predicted values are closed to the best fit line for each fold. It means that the model is performing well.

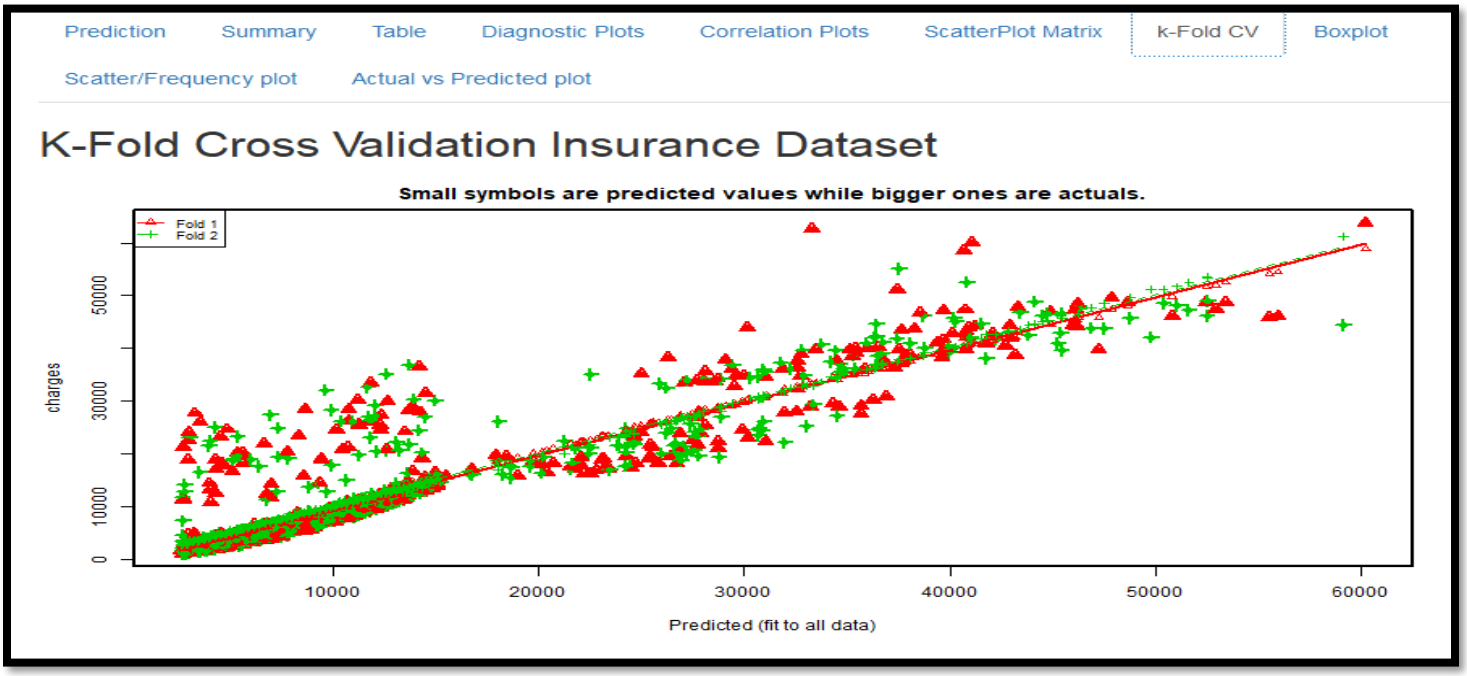


Fig 10 – 2-Fold Cross Validation on the dataset

Visualization for our model with all the significant variables:

Here, since our model used three predictors to predict the value of the response variable, it is always great to visualize the value of the outcome variable 'Charges' based on the three input variables. Here, I have shown a ggplot with Age on the x-axis, the target variable 'Charges' on the y-axis and BMI, Smoke as an aesthetic. For BMI, I have used the 'size' aesthetic and for Smoke, 'color' as an aesthetic.

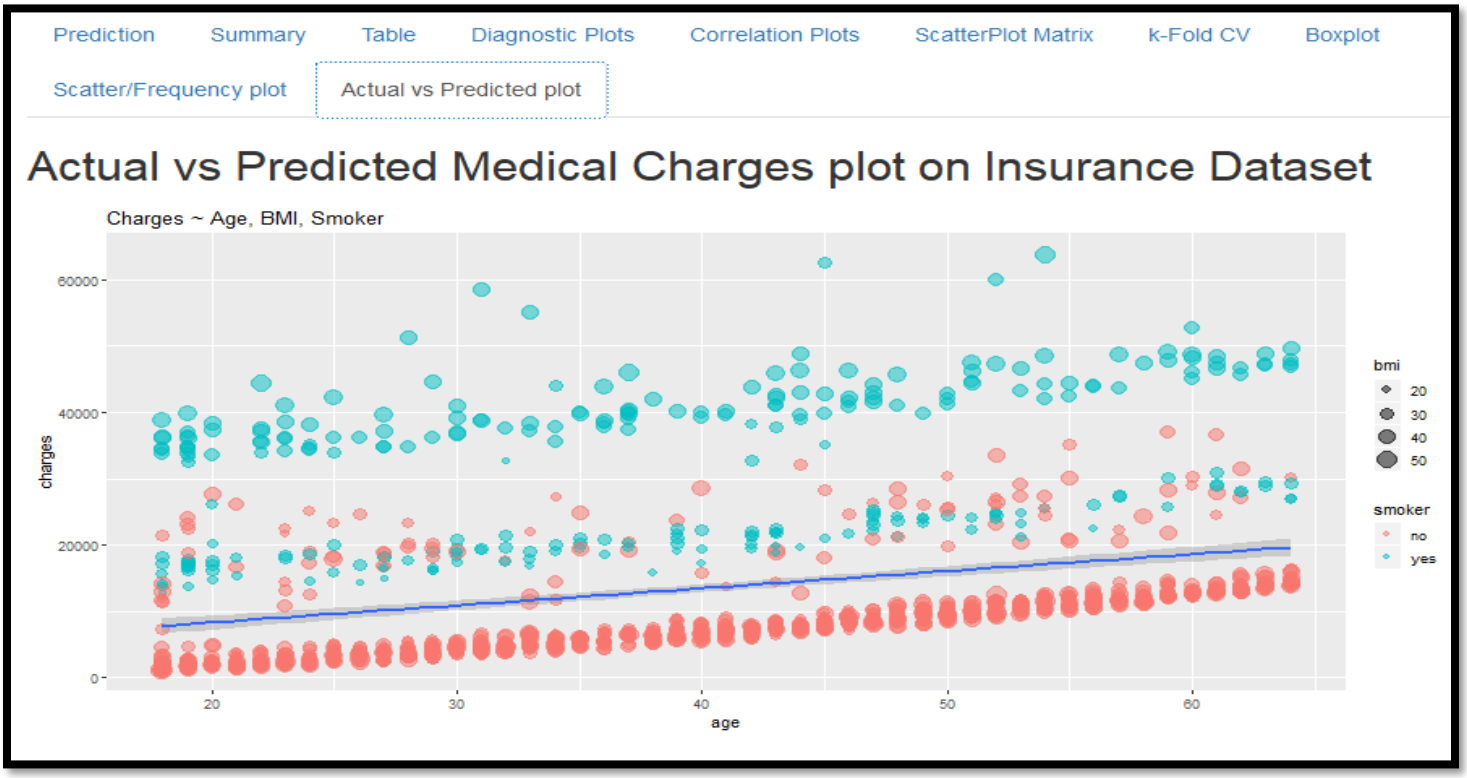


Fig. 11 – Showing Actual vs Predicted Insurance Cost for various dependent variables.

Random Forest Regression

The Random Forest is a standout amongst the best machine learning models for prescient examination, making it a modern workhorse for machine learning. The random forest model is a type of additive model that makes predictions by combining decisions from a sequence of base models.

Various types of models have diverse points of interest. The random forest model is truly adept at taking care of tabular data with numerical features, or categorical features with fewer than hundreds of categories. In contrast to linear models, random forest can catch non-linear communication between the features and the objective.

Description of Dataset:

The refractive index of any glass is its capacity to bend the ray of light. Refractive index of any glass depends upon various constituents in glass such as Iron, Potassium, Calcium, Silica, etc. According to the refractive index, the type of glass is decided and hence it is the one and only categorical variable in the dataset.

The dataset was available on the website 'https://www.kaggle.com/c/glass.csv'
The first few rows of the dataset looks something like this:

	A	B	C	D	E	F	G	H	I	J
1	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
2	1.52101	13.64	4.49	1.1	71.78	0.06	8.75	0	0	1
3	1.51761	13.89	3.6	1.36	72.73	0.48	7.83	0	0	1
4	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0	0	1
5	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0	0	1
6	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0	0	1
7	1.51596	12.79	3.61	1.62	72.97	0.64	8.07	0	0.26	1

(Figure 12: Glass Sample Dataset)

About the Variables :

- RI = It states the Refractive Index of the Glass and its datatype is 'Ratio'
- Na = It states the amount of Sodium content in the Glass and its datatype is 'Ratio'
- Mg = It states the amount of Magnesium content in the Glass and its datatype is 'Ratio'
- Al = It states the amount of Aluminium content in the Glass and its datatype is 'Ratio'
- Si = It states the amount of Silica content in the Glass and its datatype is 'Ratio'
- K = It states the amount of Potassium content in the Glass and its datatype is 'Ratio'
- Ca = It states the amount of Calcium content in the Glass and its datatype is 'Ratio'
- Fe = It states the amount of Iron content in the Glass and its datatype is 'Ratio'
- Type = It states the Type of Glass and is Categorical/Nominal datatype.

The Dataset was already a well-structured Dataset.

Following Random Forest outcomes were obtained on the dataset:

- **Graph 1:- Heat Map for the variable dependency.**

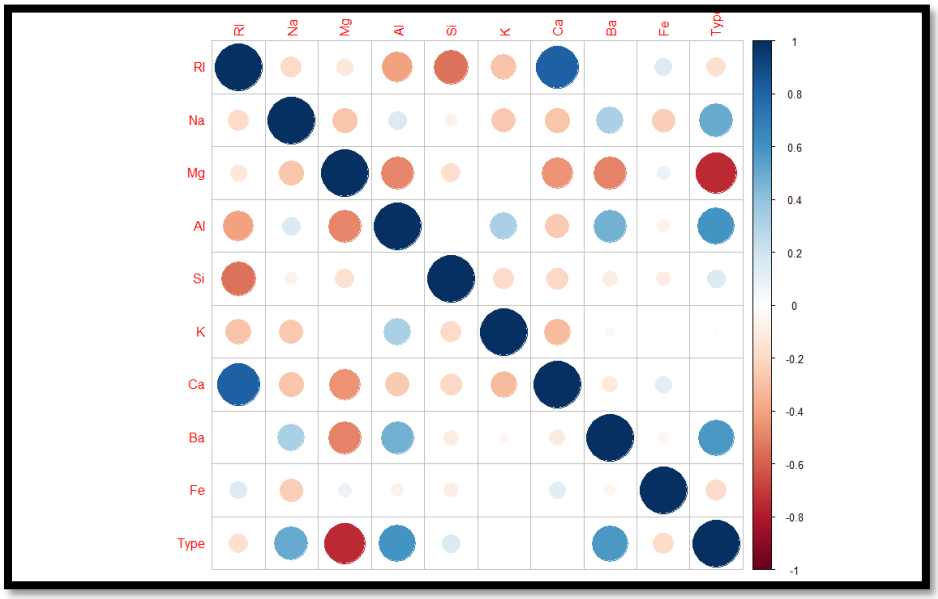
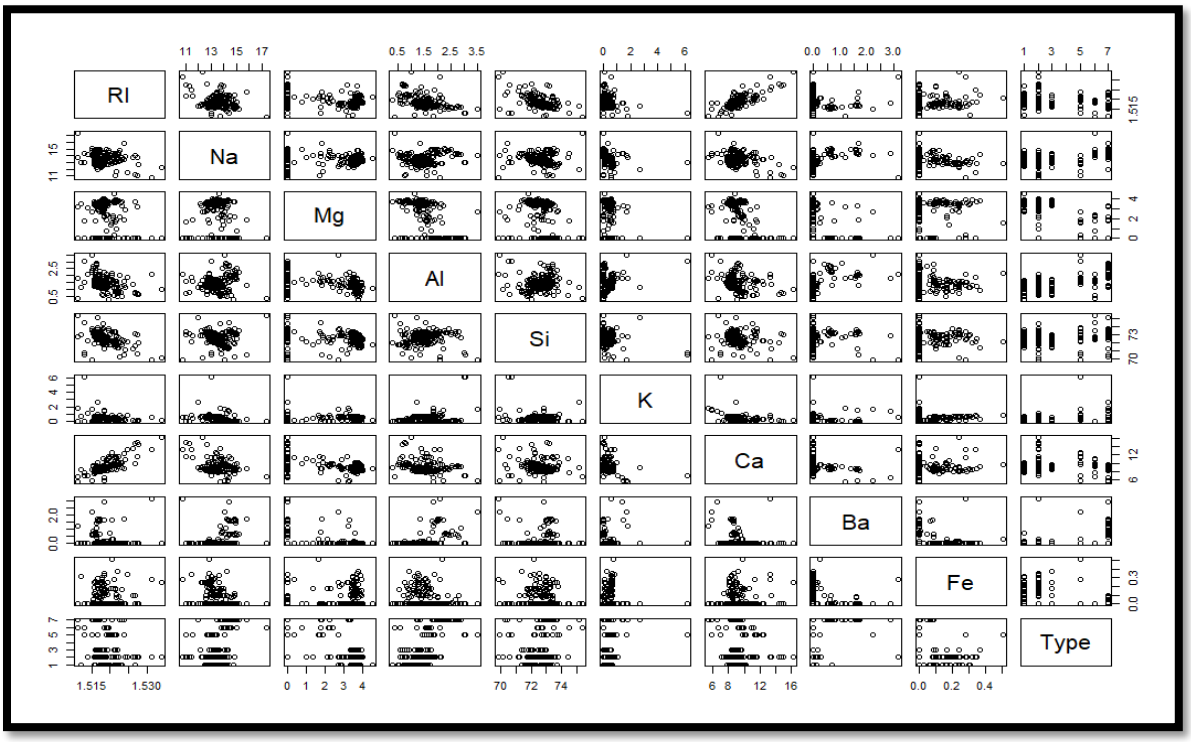


Figure 13 - Correlation Plot on Glass Dataset

The Heat Map is one of the most efficient and understandable way to get to know the relationship between the variables by the color tones which are warmer for intense correlation and cooler for not so dependent variable. We're trying to predict highest Refractive Index, so we care about the first 2 columns/rows in order to know which among the variables has the strongest relationship with wine quality. As the heatmap suggests, Calcium has the strongest correlation with Refractive Index. After Calcium, Silica is the one who has the strongest relationship with the Refractive Index factor.

- **Graph 2 : Correlation of Variables**



(Figure 14: Correlation Matrix)

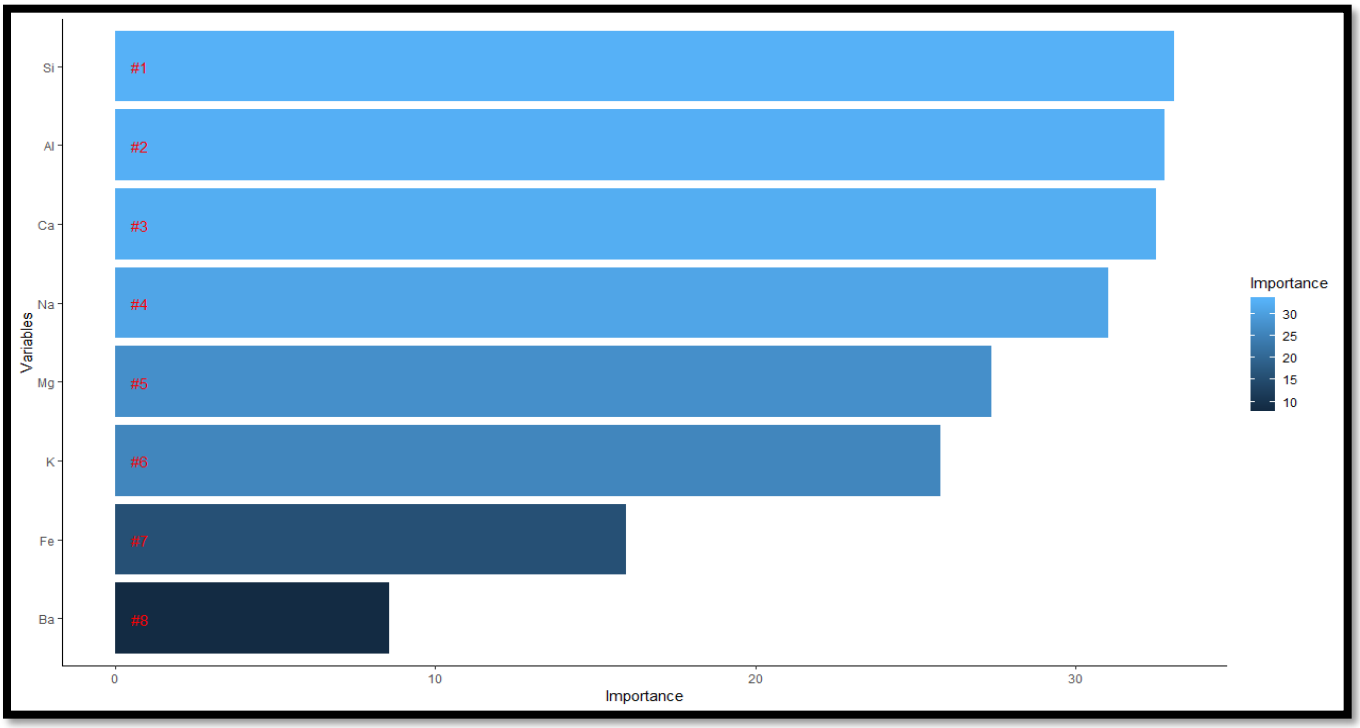
The above graph shows the correlation of the variables in the dataset. How well they are dependent on each other as well how they affect each other are given in the matrix above. For the Refractive Index over here, the most densely corelated variable is the metal Sodium(Na), after that comes the Silica (Si) and most barely scattered or dense is the Barium Column(Ba). Refractive Index is uniform with the type variable over here which is the only categorical Variable over here.

The correlation when checked on the data frame it gave the output which was recorded as :

```
> cor(glass)
               RI          Na          Mg          Al          Si
RI    1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220
Na   -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881
Mg   -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672
Al   -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372
Si   -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000
K    -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085
Ca    0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215
Ba   -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131
Fe    0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073
Type -0.1642372146  0.50289804 -0.744992888  0.59882921  0.15156526
                K           Ca           Ba          Fe          Type
RI   -0.289832711  0.8104026963 -0.0003860189 0.143009609 -0.1642372146
Na   -0.266086504 -0.2754424856  0.3266028795 -0.241346411  0.5028980423
Mg    0.005395667 -0.4437500264 -0.4922621178 0.083059529 -0.7449928875
Al    0.325958446 -0.2595920102  0.4794039017 -0.074402151  0.5988292084
Si   -0.193330854 -0.2087321537 -0.1021513105 -0.094200731  0.1515652579
K     1.000000000 -0.3178361547 -0.0426180594 -0.007719049 -0.0100544638
Ca   -0.317836155  1.0000000000 -0.1128409671  0.124968219  0.0009522246
Ba   -0.042618059 -0.1128409671  1.0000000000 -0.058691755  0.5751614590
Fe   -0.007719049  0.1249682190 -0.0586917554  1.000000000 -0.1882775640
Type -0.010054464  0.0009522246  0.5751614590 -0.188277564  1.0000000000
>
```

(Figure 15: Output of the Correlation)

- **Graph 3: Variable Importance Graph**

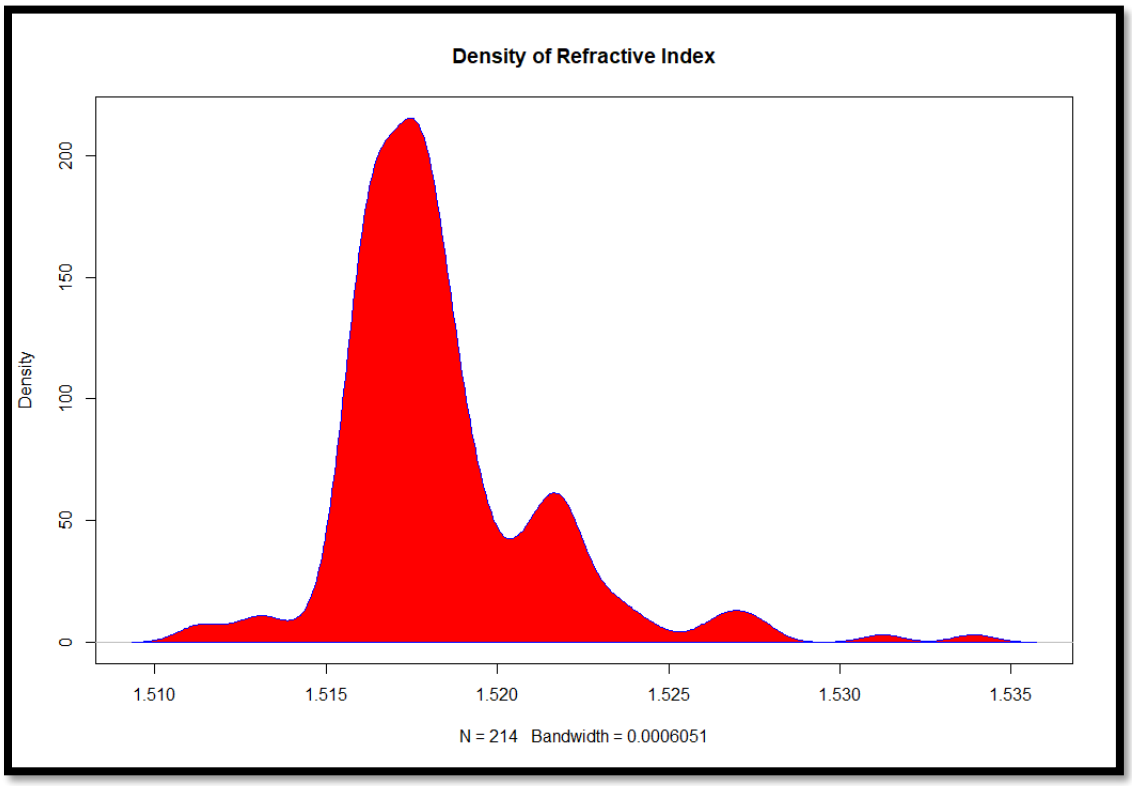


(Figure 16: Variable Importance)

Variable importance plot gives a rundown of the most critical factors in plunging request by a mean decline in Gini. The best factors contribute more to the model than the last ones and furthermore have high predictive power in arranging default and non-default variables.

As according to the graph, it can be seen that the Silica is one of the most effective variables stating that even a small change in amounts of Silica will lead to an effective change in the Refractive index of the glass.

- **Graph 4:- Density Graph**



(Figure 17: Density Graph)

As we can see here the density graph shows that the distribution is Right Skewed distribution stating that most of the glasses have refractive indices in between 1.515 and 1.520. The ones which are out of this bound have either some out of the bound constituents which results into the glass' Refractive index to be out of the normal range.

➢ Why Chose Random Forest?

When Random Forest Model was applied on the glass dataset, the outputs were :

```
> glassRF<-randomForest((RI)~.-Type,glass,ntree=150)
> glassRF

Call:
 randomForest(formula = (RI) ~ . - Type, data = glass, ntree = 150)
               Type of random forest: regression
                     Number of trees: 150
No. of variables tried at each split: 2

          Mean of squared residuals: 2.359373e-06
                    % Var explained: 74.3
>
```

(Figure 18: Model Output)

The mean square error seemed to be perfectly less with a high level of variance percentage of about 74.3%

Lasso Regression

Introduction:

Lasso regression examination is a shrinkage and variable determination technique for direct regression models. The objective of lasso regression is to acquire the subset of indicators that limits expectation blunder for a quantitative reaction variable. The lasso does this by forcing a requirement on the model parameters that causes regression coefficients for a few factors to recoil toward zero. Factors with a regression coefficient equivalent to zero after the shrinkage procedure are rejected from the model. Factors with non-zero regression coefficients factors are most firmly connected with the reaction variable. Logical factors can be either quantitative, straight out or both.

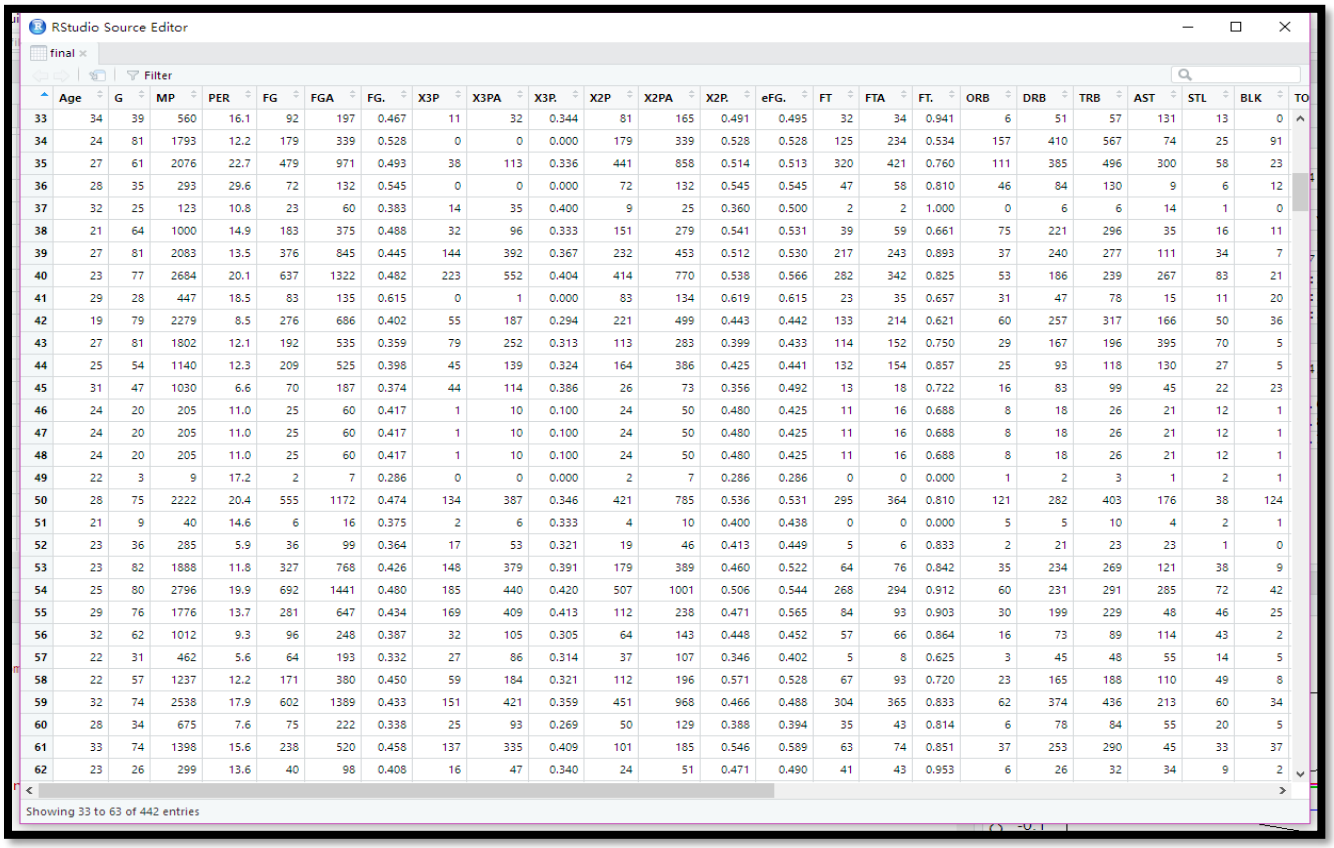
About the Dataset:

The dataset was downloaded from the website : http://www.kaggle.com/seasons_stats.csv

Description:

In this project, I have used a data set about the 2017-2018 NBA player salary. This data set includes data for each player in the NBA. I also wanted to find out if they get paid based on each player's score, rebounding, etc. I did data cleaning and merging data before using this dataset, and I was using two datasets, so I merged the data. For each player, I used up to 35 variables. This will be able to get exactly the results I need. Because the dataset I use is particularly large, my goal is to find out some variables that affect wages through lasso. So I used Lasso regression and Ridge regression as well. Both regressions are carried out with the help of glmnet package in R where glm stands for generalized linear model.

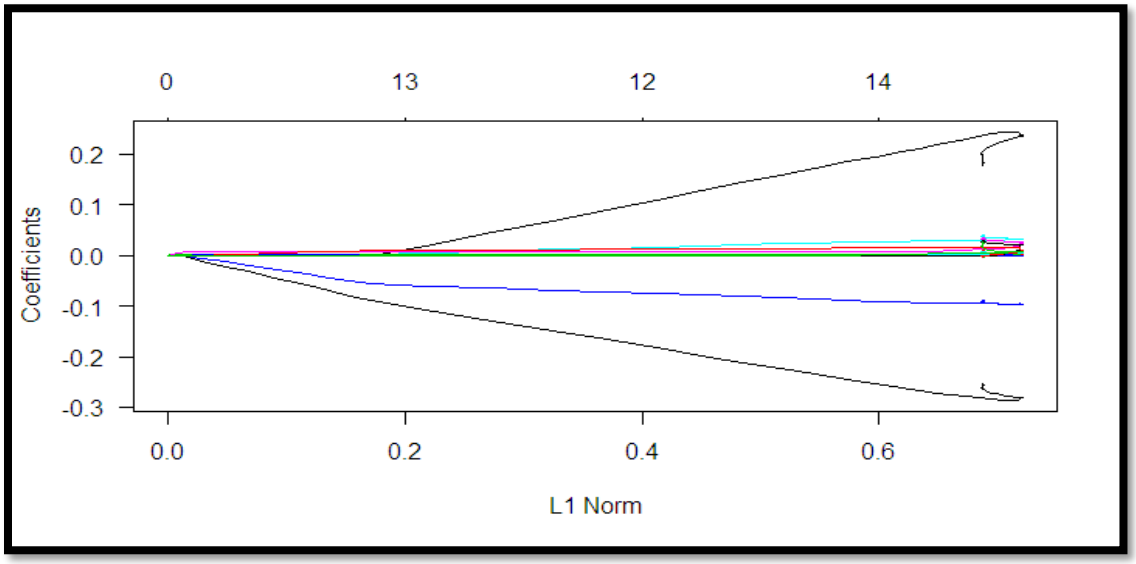
Some rows of the dataset is given below in the following screenshot:



	Age	G	MP	PER	FG	FGA	FG.	X3P	X3PA	X3P.	X2P	X2PA	X2P.	eFG.	FT	FTA	FT.	ORB	DRB	TRB	AST	STL	BLK	TO
33	34	39	560	16.1	92	197	0.467	11	32	0.344	81	165	0.491	0.495	32	34	0.941	6	51	57	131	13	0	
34	24	81	1793	12.2	179	339	0.528	0	0	0.000	179	339	0.528	0.528	125	234	0.534	157	410	567	74	25	91	
35	27	61	2076	22.7	479	971	0.493	38	113	0.336	441	858	0.514	0.513	320	421	0.760	111	385	496	300	58	23	
36	28	35	293	29.6	72	132	0.545	0	0	0.000	72	132	0.545	0.545	47	58	0.810	46	84	130	9	6	12	
37	32	25	123	10.8	23	60	0.383	14	35	0.400	9	25	0.360	0.500	2	2	1.000	0	6	6	14	1	0	
38	21	64	1000	14.9	183	375	0.488	32	96	0.333	151	279	0.541	0.531	39	59	0.661	75	221	296	35	16	11	
39	27	81	2083	13.5	376	845	0.445	144	392	0.367	232	453	0.512	0.530	217	243	0.893	37	240	277	111	34	7	
40	23	77	2684	20.1	637	1322	0.482	223	552	0.404	414	770	0.538	0.566	282	342	0.825	53	186	239	267	83	21	
41	29	28	447	18.5	83	135	0.615	0	1	0.000	83	134	0.619	0.615	23	35	0.657	31	47	78	15	11	20	
42	19	79	2279	8.5	276	686	0.402	55	187	0.294	221	499	0.443	0.442	133	214	0.621	60	257	317	166	50	36	
43	27	81	1802	12.1	192	535	0.359	79	252	0.313	113	283	0.399	0.433	114	152	0.750	29	167	196	395	70	5	
44	25	54	1140	12.3	209	525	0.398	45	139	0.324	164	386	0.425	0.441	132	154	0.857	25	93	118	130	27	5	
45	31	47	1030	6.6	70	187	0.374	44	114	0.386	26	73	0.356	0.492	13	18	0.722	16	83	99	45	22	23	
46	24	20	205	11.0	25	60	0.417	1	10	0.100	24	50	0.480	0.425	11	16	0.688	8	18	26	21	12	1	
47	24	20	205	11.0	25	60	0.417	1	10	0.100	24	50	0.480	0.425	11	16	0.688	8	18	26	21	12	1	
48	24	20	205	11.0	25	60	0.417	1	10	0.100	24	50	0.480	0.425	11	16	0.688	8	18	26	21	12	1	
49	22	3	9	17.2	2	7	0.286	0	0	0.000	2	7	0.286	0.286	0	0	0.000	1	2	3	1	2	1	
50	28	75	2222	20.4	555	1172	0.474	134	387	0.346	421	785	0.536	0.531	295	364	0.810	121	282	403	176	38	124	
51	21	9	40	14.6	6	16	0.375	2	6	0.333	4	10	0.400	0.438	0	0	0.000	5	5	10	4	2	1	
52	23	36	285	5.9	36	99	0.364	17	53	0.321	19	46	0.413	0.449	5	6	0.833	2	21	23	23	1	0	
53	23	82	1888	11.8	327	768	0.426	148	379	0.391	179	389	0.460	0.522	64	76	0.842	35	234	269	121	38	9	
54	25	80	2796	19.9	692	1441	0.480	185	440	0.420	507	1001	0.506	0.544	268	294	0.912	60	231	291	285	72	42	
55	29	76	1776	13.7	281	647	0.434	169	409	0.413	112	238	0.471	0.565	84	93	0.903	30	199	229	48	46	25	
56	32	62	1012	9.3	96	248	0.387	32	105	0.305	64	143	0.448	0.452	57	66	0.864	16	73	89	114	43	2	
57	22	31	462	5.6	64	193	0.332	27	86	0.314	37	107	0.346	0.402	5	8	0.625	3	45	48	55	14	5	
58	22	57	1237	12.2	171	380	0.450	59	184	0.321	112	196	0.571	0.528	67	93	0.720	23	165	188	110	49	8	
59	32	74	2538	17.9	602	1389	0.433	151	421	0.359	451	968	0.466	0.488	304	365	0.833	62	374	436	213	60	34	
60	28	34	675	7.6	75	222	0.338	25	93	0.269	50	129	0.388	0.394	35	43	0.814	6	78	84	55	20	5	
61	33	74	1398	15.6	238	520	0.458	137	335	0.409	101	185	0.546	0.589	63	74	0.851	37	253	290	45	33	37	
62	23	26	299	13.6	40	98	0.408	16	47	0.340	24	51	0.471	0.490	41	43	0.953	6	26	32	34	9	2	

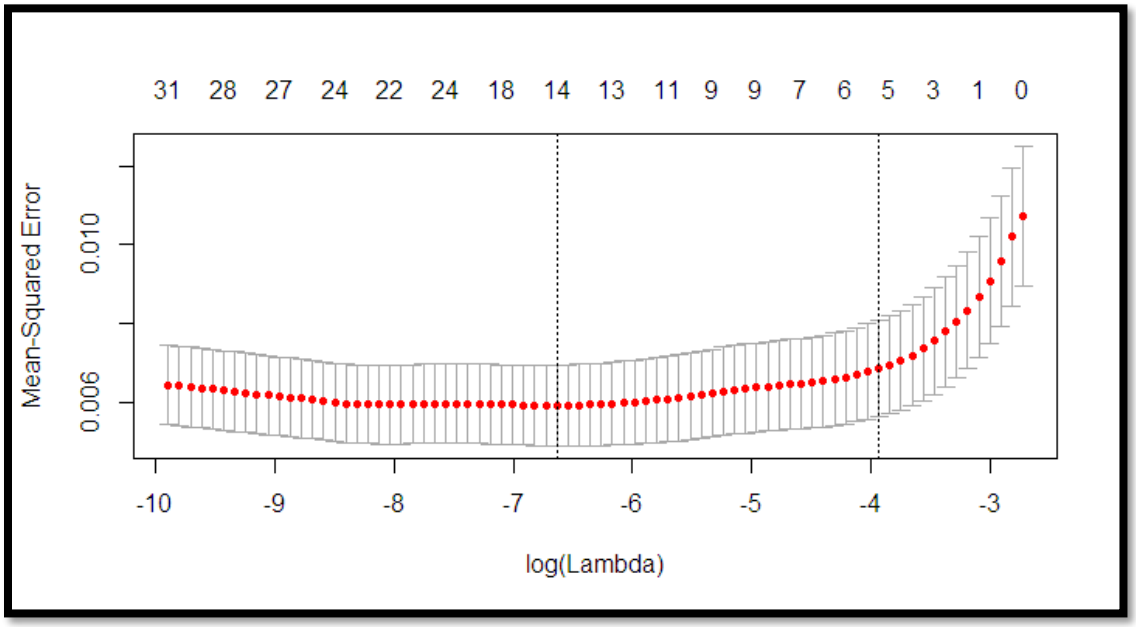
(Figure 19: Sample Dataset of NBA players)

- **Graph 1:- Lasso plot:**



(Figure 20: Lasso Regression Graph)

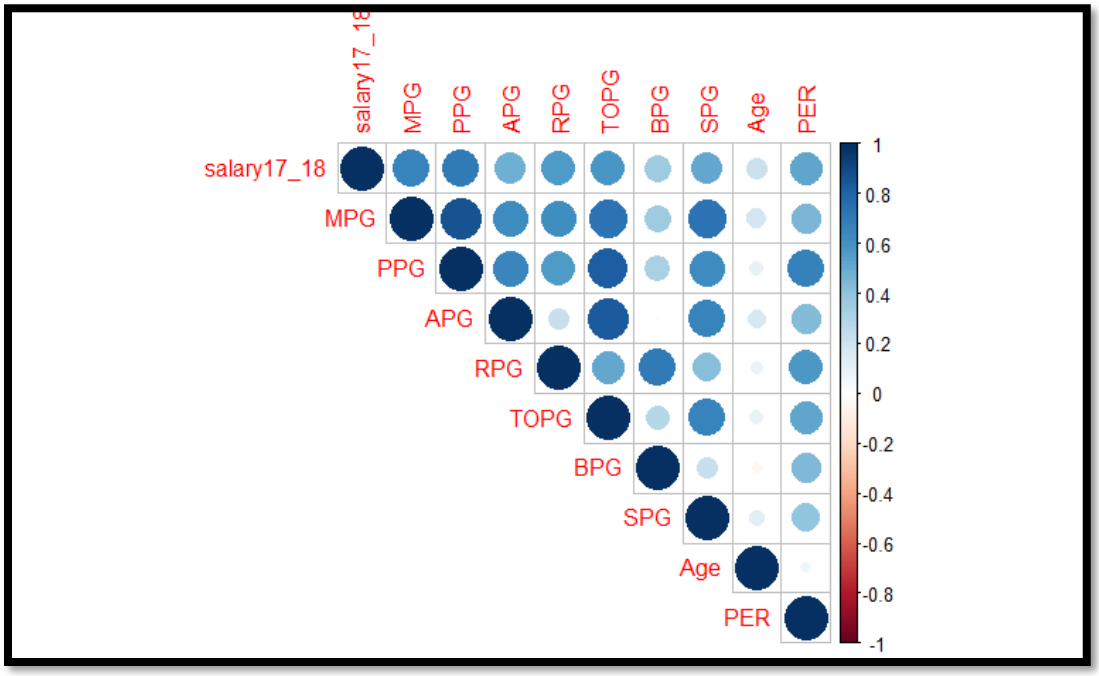
Each curve in the graph represents the trajectory of each independent variable coefficient, the ordinate is the value of the coefficient, the lower abscissa is log (λ), and the upper abscissa is the number of non-zero coefficients in the model at this time. We can see that the independent variables represented by the black and blue lines which do not change when the value of λ is zero, and then change significantly and remarkably as the value of λ becomes larger.



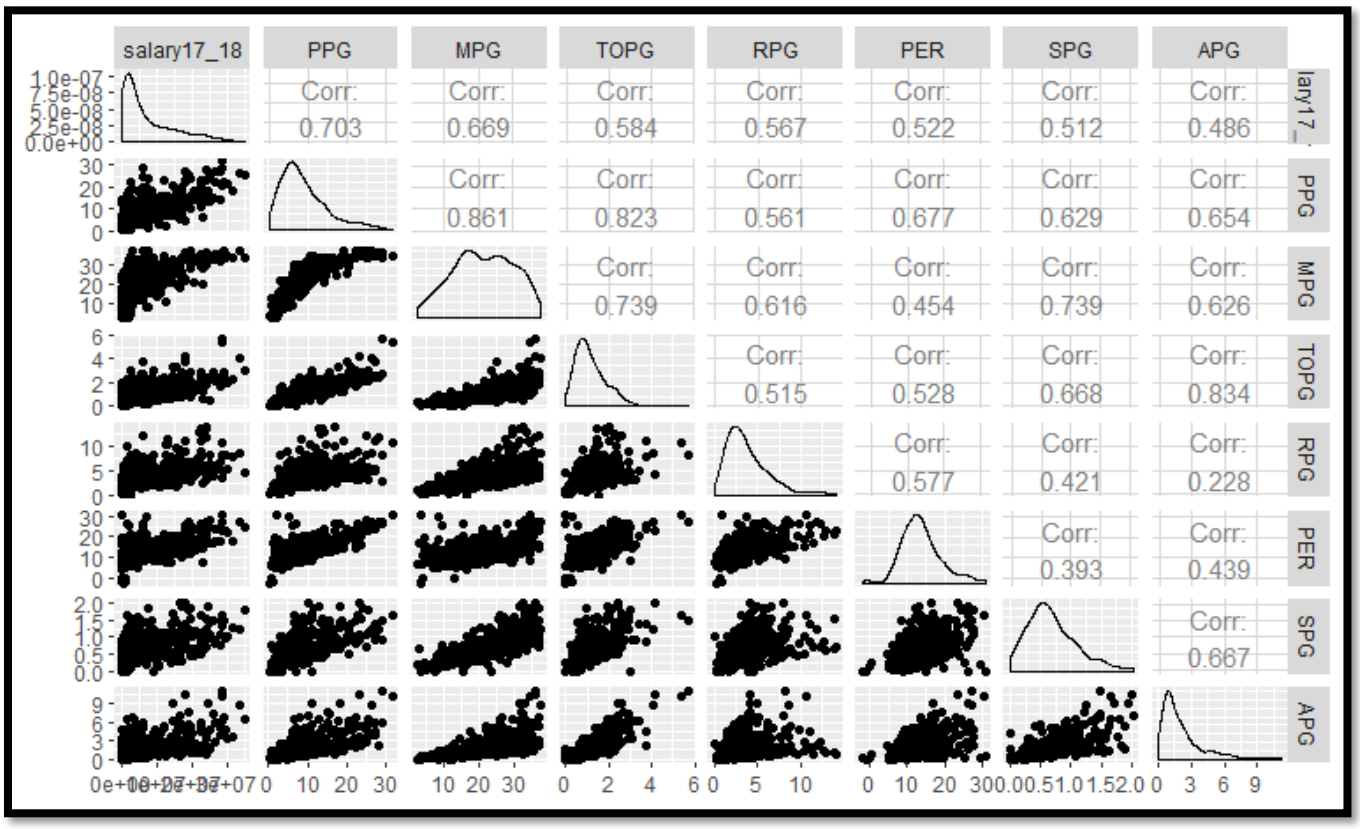
(Figure 21: Error Graph)

In this model, we can get x-axis as λ, and y-axis is mean squared error which is -0.0067. From this plot we can get an interval. Two dotted lines indicate two special λ values. One of them is the one that gets the mean of the smallest target parameter among all the λ values. The other is the λ value that yields the simplest model within a range of variance. Although I used lasso regression, I want to find out more about the relationship between player salary and what it is. So, I am going to use the correlation check.

- **Graph 2:- Correlation check**



(Figure 22: Heat Map)



(Figure 23: Correlation matrix for NBA players)

Through these two correlation checks, we can get a very intuitive comparison. We can get a comparison of Correlation strength.

```
> cor(stats_salary_cor)[,"salary17_18"]
salary17_18         PPG         MPG        TOPG         RPG         PER         SPG         APG
  1.0000000   0.7031051   0.6693910   0.5842982   0.5665350   0.5215509   0.5118549   0.4856552
>
```

(Figure 24: Correlation testing output)

Correlation strength is: PPG > MPG > TOPG > RPG > PER > SPG > APG

Where,

- PPG = Points Per Game
- MPG = Minutes per Game
- TOPG =Turnovers per game
- RPG = Rebounds Per Game
- PER = Player Efficiency Rating
- SPG = Steals per game
- APG = Assists per game

The interesting part of this is that the number of turnover players make is linked to their salary, and the relationship has a positive correlation. So, I interpreted this relationship like this: “the more turnovers they make” means that they are more involved in ball movements in games, which means that players who make turnovers are, at some extend, important to their team. and hence this could be expressed as “Competitiveness”. In fact, there are still many models that can unearth the rules of change in NBA player wages.

RScript

1. Linear Regression

InsuranceLR.r File

```
##Medical Costs Personal Datasets

##Link of the dataset for download
##https://www.kaggle.com/mirichoi0218/insurance

##Installing Packages and loading all the libraries

install.packages("tidyverse")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("randomForest")
install.packages("gridExtra")
install.packages("adegraphics")
install.packages("corrplot")
install.packages("RColorBrewer")
install.packages("psych")

library(shiny)
library(corrplot)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(randomForest)
library(gridExtra)
library(adegraphics)
library(RColorBrewer)
library(psych)

##Read and View the Dataset
insurance <- read.csv("C:/Users/Abhishek/Desktop/STAT 515/Final Project/insurance.csv")
View(insurance)

##Structure and summary of the dataset
head(insurance)
str(insurance)
summary(insurance)
class(insurance)

##Single column summary
summary(insurance$sex)
summary(insurance$smoker)
class(insurance$sex)
class(insurance$smoker)

##Encoding column values in data-- Converting Factor to numeric data
table(as.numeric(insurance$sex))
table(as.numeric(insurance$smoker))
table(as.numeric(insurance$region))
table(as.numeric(insurance$children))
```

```
##Ggplot for Charges vs Age
plot.agescatter <- ggplot(insurance, aes(x = age, y = charges)) +
  geom_jitter(color = "blue", alpha = 0.5) +theme()
plot.agescatter

##Ggplot for Charges vs BMI
plot.bmiscatter <- ggplot(insurance, aes(x = bmi, y = charges)) +geom_point()
plot.bmiscatter

##Ggplot for Charges vs no of children
plot.childrenscatter <- ggplot(insurance, aes(children, charges)) +
  geom_jitter(aes(color = children), alpha = 0.5) +theme_light()
plot.childrenscatter

##Arrangement of Multiple plots-- (nrow=1) means only 1 row
plot.agebmiscatter <- grid.arrange(plot.agescatter, plot.bmiscatter)
q <- grid.arrange(plot.agescatter, plot.bmiscatter,plot.childrenscatter, nrow=2)

##Boxplot for Sex vs Charges
plot.sexbox <- ggplot(insurance, aes(x = sex, y = charges)) +geom_boxplot(fill = c(2:3)) + ggtitle("Boxplot of
Medical Charges per Sex")+theme_classic()
plot.sexbox

##Boxplot for Smoker vs Charges
plot.smokerbox <- ggplot(insurance, aes(x = smoker, y = charges)) +geom_boxplot(fill = c(3:4))+
ggtitle("Boxplot of Medical Charges vs Smoking(y/n)")+theme_classic()
plot.smokerbox

##Boxplot for children vs charges
insurance$children<-as.factor(insurance$children)
plot.childbox <- ggplot(insurance, aes(x = children, y = charges)) +geom_boxplot(fill = c(10:15))+
ggtitle("Boxplot of Medical Charges vs Number of children")
plot.childbox

##Boxplot for region vs charges
plot.regionbox <- ggplot(insurance, aes(x = region, y = charges)) +geom_boxplot(fill = c(4:7))+ ggtitle("Boxplot
of Medical Charges per Region")+theme_classic()
plot.regionbox

r <- grid.arrange(plot.sexbox, plot.smokerbox,plot.childbox,plot.regionbox , nrow=2)

##Ggplot--Scatterplot for Age vs Charges + BMI + Smoker
##Also scatterline showing relationship between Age and charges
ggplot(insurance, aes(x=age, y = charges)) +
  geom_point(aes(color = smoker, size = bmi),alpha=0.5) + #four groups: smoker with high BMI, non-smoker
with high BMI, smoker with low BMI and non-smoker with low BMI --> could indicate that there needs to be an
interaction term between BMI and Smoker.
  ggtitle("Charges ~ Age, BMI, Smoker") +geom_smooth(method='lm')

##Frequency Plot-- Histogram Plot for charges vs count of charges for binwidth=1000
ggplot(data=insurance, aes(x=charges)) +
  geom_histogram(binwidth=1000, fill="blue") + ggtitle("Frequency plot for Medical Charges")

##Correlation Matrix
cor(insurance, use="pairwise.complete.obs")
insurancecorr <- insurance[,1:length(insurance)]
```

```
insurancecorr<-data.matrix(insurancecorr)
round(cor(insurancecorr),2)
corrplot(cor(insurancecorr), method = "circle")
corrplot(cor(insurancecorr), type="upper", order="hclust",
        col=brewer.pal(n=10, name="RdBu"))

#Scatterplot Matrix
pairs(charges~.,data=insurance,
     main="Simple Scatterplot Matrix")
pairs.panels(insurance)

##Splitting data into Train and Test Dataset
data <- round(0.7 * nrow(insurance))
traindata <- sample(1:nrow(insurance), data)
insurance_train <- insurance[traindata, ]
insurance_test <- insurance[-traindata, ]

##Linear Regression Model with all variables
mod1 = lm(charges ~ ., data = insurance_train)
summary(mod1)
##R2--0.7402

mod2 = lm(charges ~ age + bmi +children +smoker, data = insurance_train)
summary(mod2)
##R2--0.7397

mod3 = lm(charges ~ age + bmi +smoker, data = insurance_train)
summary(mod3)
##R2--0.7365

mod4 = lm(charges ~ bmi +smoker, data = insurance_train)
summary(mod4)
##R2--0.6474

mod5 = lm(charges ~ age +smoker, data = insurance_train)
summary(mod5)
##R2--0.7136

mod6 = lm(charges ~ age *bmi *smoker, data = insurance_train)
summary(mod6)
##R2--0.8258

mod7 = lm(charges ~ age +smoker + bmi :smoker, data = insurance_train)
summary(mod7)
plot(mod7)

##R2--0.8262--Final Model with only 3 terms and which is better in terms of R sqr value

mod8 <- lm(charges ~ smoker + age + children + region + smoker * bmi, data = insurance_train)
summary(mod8)
##R2--0.8308

mod9 = lm(charges ~ age +smoker+ bmi :smoker +region:bmi, data = insurance_train)
summary(mod9)
##R2==0.837
```

```
#Saving R-squared
rsq_train <- summary(mod7)$r.squared
rsq_train

#correlation for mod7 is squareroot of R^2
sqrt(0.8262)
##Correlation value of 0.91

#predict data on test set and review summary stats and plots
insurance_test$prediction <- predict(mod7, newdata = insurance_test)
ggplot(insurance_test, aes(x = prediction, y = charges)) +
  geom_point(color = "blue", alpha = 0.7) +
  geom_abline(color = "red") +
  ggtitle("Prediction vs. Real values")
summary(mod7)
AIC(mod7)
##R2--0.8321
##AIC--18596.42

#Prediction Accuracy for the Test Dataset as compared to actual dataset values
cor(x = insurance_test$prediction,y = insurance_test$charges)
# actuals_preds <- data.frame(cbind(actuals=insurance_test$charges, predicteds=predict_ontestdata))
# correlation_accuracy <- cor(actuals_preds)
# correlation_accuracy
##91.88% correctly predicted

##K-Fold Cross Validation
library(DAAG)
cvResults <- suppressWarnings(CVlm(insurance, form.lm=charges ~ age +smoker + bmi :smoker, m=2,
dots=TRUE, seed=10, legend.pos="topleft",  printit=TRUE, main="Small symbols are predicted values while
bigger ones are actuals."));
attr(cvResults, 'ms')
##cv-24367863

##Residual Sum of Squares(Rss)
RSS <- c(crossprod(mod7$residuals))
MSE <- RSS / length(mod7$residuals)
#RMSE <-sqrt(MSE)
RMSE <-sqrt(mean(mod7$residuals^2))
##SSE and SST
SSE<- sum((predict_ontestdata-insurance_test$charges)^2)
SSE
SST<-sum((insurance_test$charges-mean(insurance_test$charges))^2)
SST

#Min Max accuracy and MAPE
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
#77.33%--min_max accuracy
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
#31.23%--mean absolute percentage error

#calculating the residuals for test data
residuals_ontestdata <- insurance_test$charges - insurance_test$prediction
plot(residuals_ontestdata)

Data_test$residuals <- Data_test$charges - Data_test$prediction
```

```
#calculating Root Mean Squared Error
rmse_ontestdata <- sqrt(mean(residuals_0^2))
##rmse-4713

r_sq_1 <- summary(model_1)$r.squared

prediction_1 <- predict(model_1, newdata = Data_test)

residuals_1 <- Data_test$charges - prediction_1
rmse_1 <- sqrt(mean(residuals_1^2))
```

Shiny Code(server.R)

```
library(shiny)

mod7 = lm(charges ~ age +smoker + bmi :smoker, data = insurance)
summary(mod7)
#plot(mod7)

# Define server logic for this application
shinyServer(function(input, output) {

  # Reactive expression to predict the mpg. This is
  # called whenever the inputs change. The renderers defined
  # below then all use the value computed from this expression
  output$myTrans <- renderText({ input$Trans })

  output$charges <- renderText({
    input$actionButton
    isolate({
      # wt ,qsec,am
      newdata = data.frame(age=input$age,bmi=input$bmi, smoker=input$smoker)
      myp  <- predict(mod7,newdata)
      #output$myP <- p[1]
    })
  })

  # Generate diagnostic plot s
  output$myplot <- renderPlot({

    # optional 4 graphs/page
    layout(matrix(c(1,2,3,4),2,2,byrow=T))
    plot(mod7)

  })


  output$myplot1 <- renderPlot({
    library(corrplot)
    # optional 4 graphs/page
    #layout(matrix(c(1,2,3,4),2,2,byrow=T))
    #plot(round(cor(insurancecorr),2))
    corrplot(cor(insurancecorr), method = "circle")
  })
```

```
  # we <- reactive({
  #   w <- as.numeric(input$Weight)
  # })

  output$myplot2 <- renderPlot({

    pairs(charges~.,data=insurance)
  })


  output$myplot3 <- renderPlot({

    cvResults <- suppressWarnings(CVlm(insurance, form.lm=charges ~ age +smoker + bmi :smoker, m=2, dots=TRUE,
seed=10, legend.pos="topleft",  printit=TRUE, main="Small symbols are predicted values while bigger ones are
actuals."));
    attr(cvResults, 'ms')
  })

  output$myplot4 <- renderPlot({

    grid.arrange(plot.sexbox, plot.smokerbox,plot.childbox,plot.regionbox , nrow=2)
  })

  output$myplot5 <- renderPlot({
    plot.freqcnt<-ggplot(data=insurance, aes(x=charges)) +
      geom_histogram(binwidth=1000, fill="blue") + ggtitle("Frequency plot for Medical Charges")

  grid.arrange(plot.agescatter, plot.bmiscatter,plot.childrenscatter,plot.freqcnt, nrow=2)
})
  output$myplot6 <- renderPlot({
    ggplot(insurance, aes(x=age, y = charges)) +
      geom_point(aes(color = smoker, size = bmi),alpha=0.5) + #four groups: smoker with high BMI, non-smoker with high
BMI, smoker with low BMI and non-smoker with low BMI --> could indicate that there needs to be an interaction term
between BMI and Smoker.
      ggtitle("Charges ~ Age, BMI, Smoker") +geom_smooth(method='lm')

      })

  # Generate a summary of the data
  output$summary <- renderPrint({
    summary(mod7)
  })

  # Generate an HTML table view of the data
  output$table <- renderTable({
    data.frame(insurance)
  })
})
```

UI.r(Shiny UI File)

```
library(shiny)
shinyUI(pageWithSidebar(
    # Application title
  headerPanel("Predicting insurance costs for Medical Data"),
```

```
  # Adding widgets
    sidebarPanel(

    helpText("Enter Age, BMI and Smoking values, click on ",strong("Predict!")," to get predicted value of insurance
costs."),
    numericInput("age", "age", value = 18, min = 18, max = 64),
    sliderInput("bmi","bmi",value=30,min=15.96,max=53.13,animate=T, round = FALSE, step = 0.01),
    br(),
    radioButtons("smoker","Smoking",list("No" = "no","Yes" = "yes"),"no"),
    actionButton("actionButton","Predict!",align = "center"),
    helpText("Note: For ",strong("Summary, Plot")," and" ,strong("Table"),", whole",strong("Insurance"),"dataset is taken
into account and results are displayed respectively")

  ),

  # Show a tabset that includes mpg prediction, plot, summary, and table view of mtcars dataset
    mainPanel(
    tabsetPanel(
      tabPanel("Prediction",
               h2("Insurance Costs Prediction using Multiple Linear Regression",align="center") ,
               p("Selection of a multiple linear regression (MLR) model that describes a dependent variable term 'charges' by
independent variables terms which are significant i.e. Age, Smoker and an interaction product of BMI and Smoker and
removing variables which are not much significant such as sex, children and region."),


               h3("Prediction for Insurance Cost"),

               p("We now apply the predict function on the input data to see the predicted insurance costs."),

               p("The predicted",span(" Insurance Cost",style = "color:blue")," value is :"),
               code(textOutput("charges"))
      ),
               # p("The predicted",span(" MPG value",style = "color:blue")," and its",span(" Lower
Bound",style="color:blue"),"and",
               #   span("Upper Bound",style="color:blue")," values are :"),


      tabPanel("Summary",
               h2("Summary of",strong("Insurance"),"dataset"),
               verbatimTextOutput("summary")),
      tabPanel("Table",
               h2("View of Insurance dataset"),
               tableOutput("table")),
      tabPanel("Diagnostic Plots" ,
               h2("Diagnostic plots for Insurance dataset(Pattern, Distribution, Spread & Influence)"),
               plotOutput("myplot")),
      tabPanel("Correlation Plots" ,
               h2("Correlation Plots for Insurance Dataset"),
               plotOutput("myplot1")),
      tabPanel("ScatterPlot Matrix" ,
               h2("Scatterplots for Insurance Dataset"),
               plotOutput("myplot2")),
      tabPanel("k-Fold CV" ,
               h2("K-Fold Cross Validation Insurance Dataset"),
               plotOutput("myplot3")),
      tabPanel("Boxplot" ,
```

```
            h2("Boxplot on Insurance Dataset"),
            plotOutput("myplot4")),
      tabPanel("Scatter/Frequency plot" ,
            h2("Scatter and Frequency plots on Insurance Dataset"),
            plotOutput("myplot5")),
      tabPanel("Actual vs Predicted plot" ,
            h2("Actual vs Predicted Medical Charges plot on Insurance Dataset"),
            plotOutput("myplot6"))
    )
  )
))
```

2. Random Forest Script

```
#Libraries needed

library(ggplot2)
library(ggthemes)
library(corrplot)
library(reshape2)
library(dplyr)
library(randomForest)

#Load in our dataset

glass<-read.csv("glass.csv", header = TRUE, sep = ",")

# summary statistics
str(glass)

summary(glass)

#Scatterplot Matrix of Variables
plot(glass)

#Correlation Heatmap of Variables
corrplot(cor(glass))

#Baseline Random Forest Model
glassRF<-randomForest((RI)~.-Type,glass,ntree=150)
glassRF

# Get importance
importance    <- importance(glassRF)

varImportance <- data.frame(Variables = row.names(importance),
                            Importance = round(importance[ ,'MeanDecreaseGini'],2))

# Create a rank variable based on importance
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Use ggplot2 to visualize the relative importance of variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
```

```
        hjust=0, vjust=0.55, size = 4, colour = 'red') +
  labs(x = 'Variables') +
  coord_flip() +
  theme_classic()

#Density Plot

d= density(glass$RI)
plot(d,main ="Density of Refractive Index")
polygon(d,col="red",border="blue")
```

3. Lasso Regression Script

```
#loading the libraries
library(data.table)
library(corrplot)
library(GGally)
library(PerformanceAnalytics)
library(plotly)
library(tidyverse)
library(caret)

#reading the data into the dataframe
salary.table <-
  read.csv("NBA_season1718_salary.csv")
ss <- read.csv("Seasons_Stats.csv")

#Cleaning the data and explorations
stats17 <-
  ss %>% filter(Year >= 2017) %>%
  select(Year:G, MP, PER, FG:PTS) %>%
  distinct(Player, .keep_all = TRUE) %>%
  mutate(MPG = MP/G, PPG = PTS/G, APG = AST/G,
         RPG = TRB/G, TOPG = TOV/G, BPG = BLK/G,
         SPG = STL/G)

final <- merge(stats17, salary.table, by.x = "Player", by.y = "Player")
names(final)[40] <- "salary17_18"
final <- final[-39]
final<-final[-38]
final <- final[-2]
final <- final[-1]
final <- final[-1]
final <- final[-2]

final[is.na(final)] <- 0

#Lasso Regression
#Creation of Matrix

x <- as.matrix(final[,1:33])
y <- log(final[, 34])

y<-log(y)

#Training the dataframe
nr <- nrow(x)
```

```
trainingSize <- ceiling(nr/2) # half case for training
set.seed(37)
train <- sample(1:nr,trainingSize)

x.train <- x[train,]
y.train <- y[train]

#Testing the dataframe
x.test <- x[-train,]
y.test <- y[-train]
head(x.test)

#Visualizations for Lasso and Ridge Regression
library(glmnet)
lasso.mod <- glmnet(x,y, alpha=1)
plot(lasso.mod,las=1)

set.seed(37)
cv.out = cv.glmnet(x[train,], y[train] ,alpha=1)
plot(cv.out)

#Correlation check
corrplot(cor(final %>%
               select(salary17_18, MPG:SPG,
                      Age, PER, contains("%")),
             use = "complete.obs"),
         method = "circle",type = "upper")


stats_salary_cor <-
  final %>%
  select(salary17_18, PPG, MPG, TOPG, RPG, PER, SPG, APG)
ggpairs(stats_salary_cor)


cor(stats_salary_cor)[,"salary17_18"]
```