**GRoup - 10**

**Laxmikanta Gauda**

**Ambili Jayachandran**

**Maya Vasudevan K**

**Tarun Kamal Khiani**

**M Sarath**

**M Sarath tarun kamal khiani**

Data Mining

Assignment – HR Employee Attrition Prediction

Contents

[Goal 3](#_Toc483000561)

[Solution 3](#_Toc483000562)

[Exploratory Data Analysis 3](#_Toc483000563)

[Principal Component analysis 6](#_Toc483000564)

[Factor Analysis 8](#_Toc483000565)

[Predict using Multiple regression with all variables 9](#_Toc483000566)

[Predict using Multiple regression with Factors & RAD 10](#_Toc483000567)

[Conclusion 11](#_Toc483000568)

[R Script 13](#_Toc483000569)

# Goal

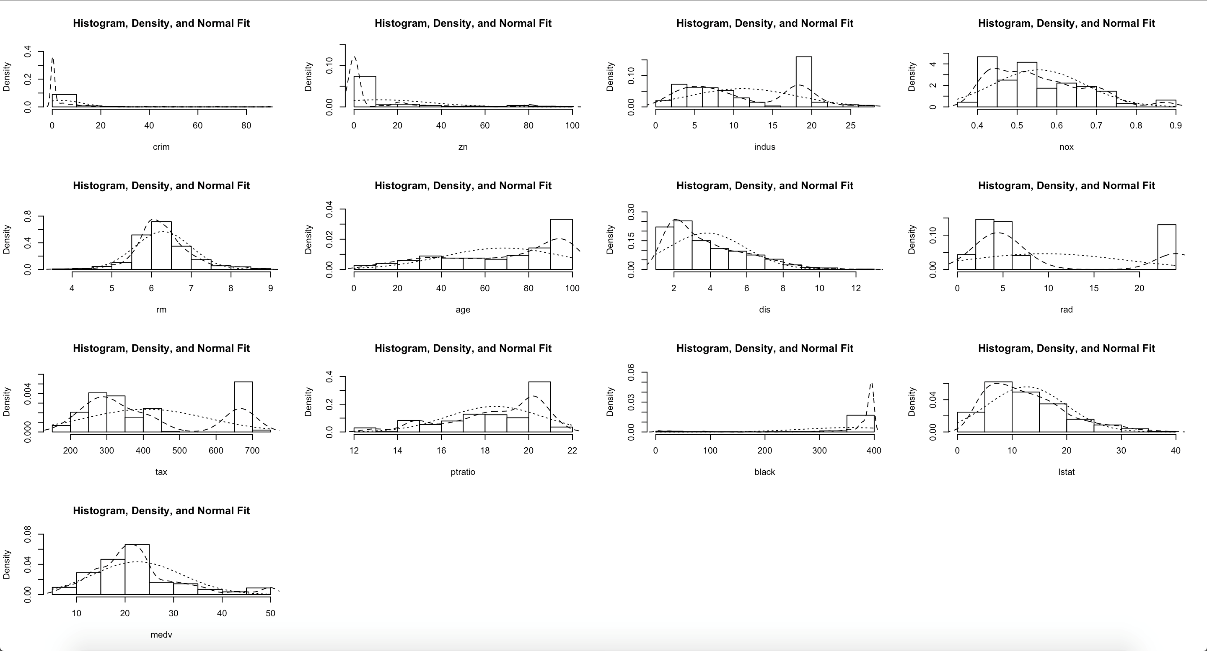
The goal of this assignment is to predict median housing value (**medv**) using the Boston data set available with the library **MASS**

# Solution

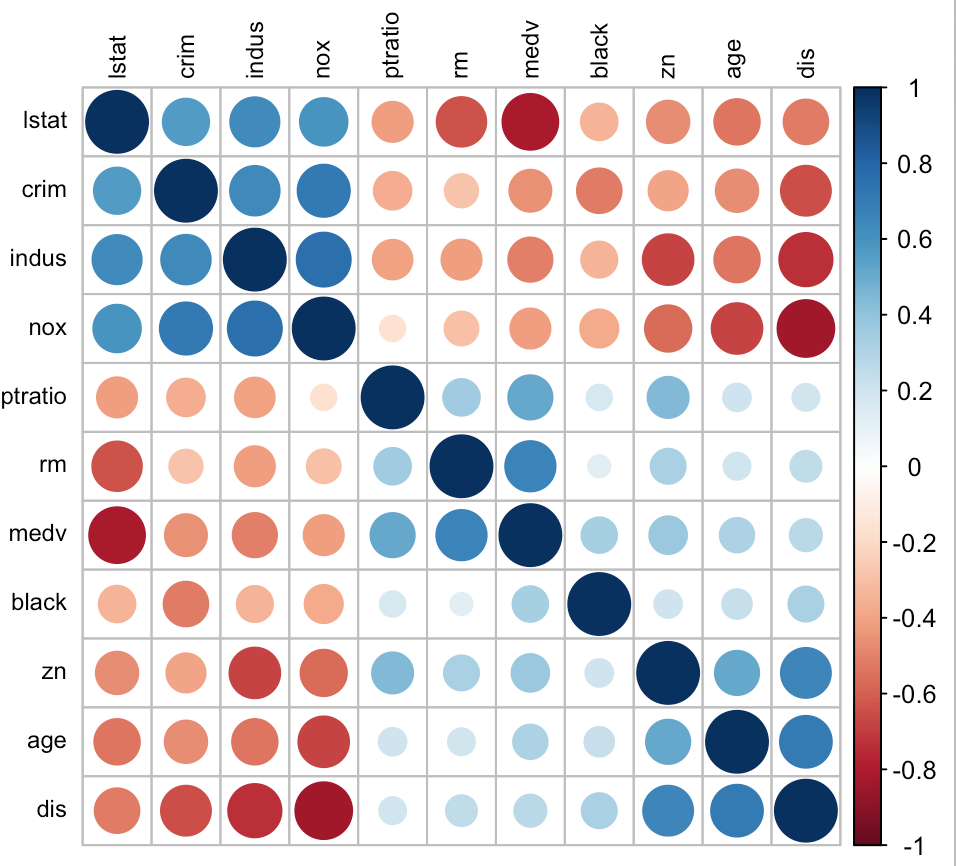
## **Exploratory Data Analysis**

Important Observations:

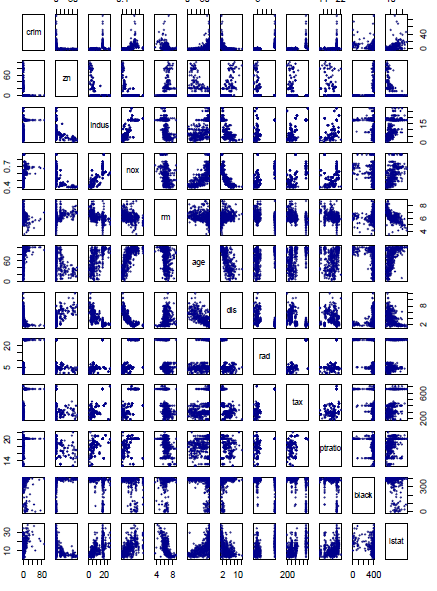
* Variables are not normally distributed. Hence a log transformation would help reduce the skew
  + crim, zn, indus, nox, rm, dis, tax, lstat are positively skewed
  + age, ptratio, black are negatively skewed
  + outliers exist in crim, zn, chas, black



* Variances of the indices are widely different. Hence scaling of variables is necessary for further analysis.
* From the correlation plot, it is evident that several variables are strongly correlated. Several pairwise correlations are over 70%. Hence this data set is a prime candidate for PCA or factor analysis.



* Scatterplot of the variables also reveals the same story as depicted below:



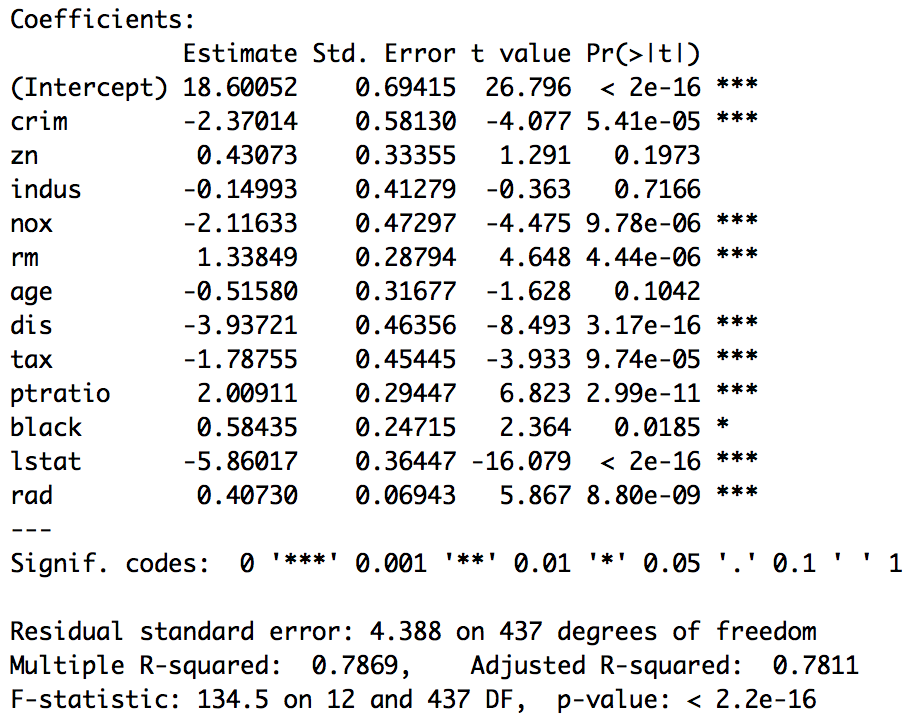
* Now the Data is split into train and test for us to analyse our model.

## **Principal Component analysis**

PCA (principal component analysis) is done on the training data to determine the number of factors.

Before doing PCA on Boston training data, first we need to check the multi-collinearity effect on our model if we do a multiple regression with the 12 explanatory variables against the **medv** response variable.

The multiple regression model using all the variables expresses **78% of the variation** but some of the variables like **zn**, **indus** and **age** is highly significant.



As the variables are highly correlated we shall either use **VIF to eliminate variables** one after another or use a dimension reduction technique like **PCA or Factor Analysis**.

First we check for VIF - because if there is strong multi-collinearity effect then this model won't be a good one to predict and form a good equation.

**crim zn indus nox rm age dis rad tax ptratio black**

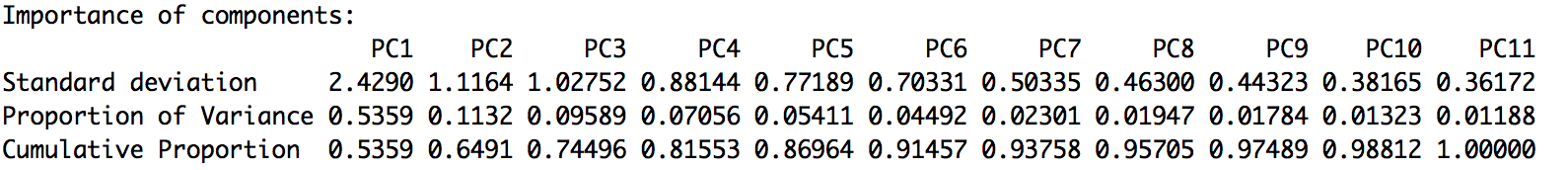
1.79 2.17 3.94 4.33 1.89 2.97 3.74 7.31 8.75 1.84 1.35

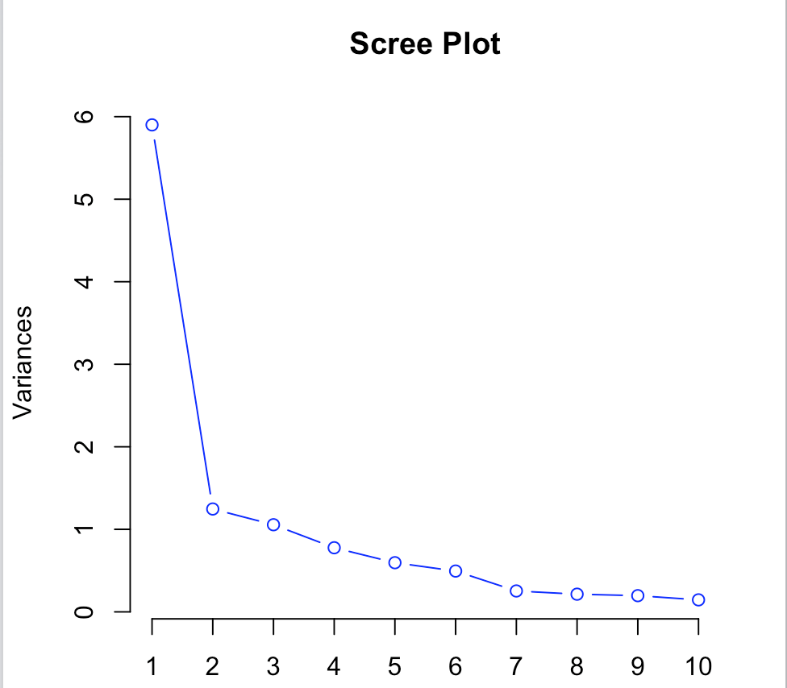
**lstat**

2.95

Clearly we can see several variables are present with a moderate to high collinearity - like **tax, rad, nox** etc. This means in order to **not eliminate variables** for our prediction we should go for PCA to remove the multi-collinearity effect on the model.

**Principal Component Analysis is done on the scaled data.** Summary of the results are given below:





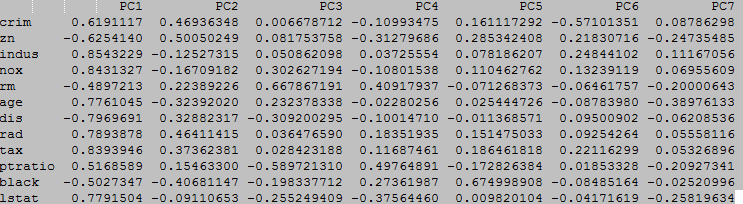
Ideally, we can choose 3 factors (**variance > 1**) but as it **only explains 74.44%** hence decided to **extract 5 factors** that **explains 87% of the variation**.

**Important Observations:**

* After **PC3** from **PC4** onwards the eigenvalue **falls below 1** but it only explains **74.44%** Of the total variances.
* The **First 4 principal components explain almost 82%** of the total variances. However, the first 5 and 6 components explain 87% and 91% respectively, even though the eigenvalue is less than 1.
* Based on this info, for **factor extraction either 4 or max 5 factors** need to be **considered**.

Investigating the correlation (shown below) between the variables and the first 4 or 5 components it is observed that **PC1 has high correlation** (≥ 0.7) with 7 of the 12 variables, while the **other PCs do not have high correlation** with any indices at all. This does not render well to proper interpretation.

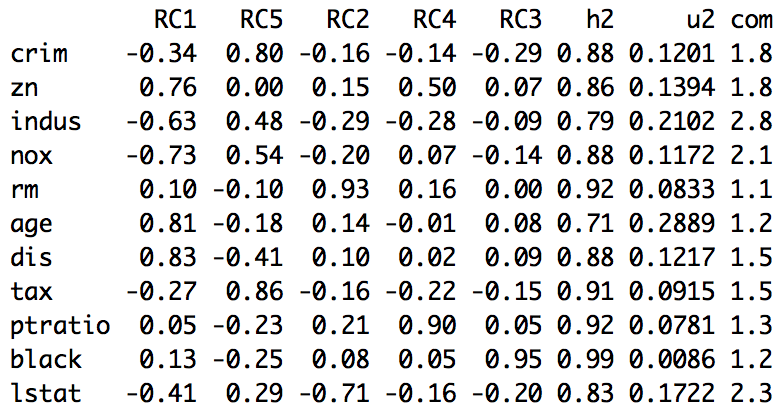
Hence extraction of factors will be the next stage of our analysis.



## **Factor Analysis**

As mentioned before either a 4-factor model or a 5-factor model is to be considered and here we have **extracted 5 factors for our further analysis** as *4 factor model seems not enough to properly segregate each factor with a common name*.

Following is the factor extraction using “**varimax**” rotation technique.



**Factor 1: City Outskirts (RC1)**

**-** **indus**: proportion of non-retail business acres per town

**-** **nox**: nitrogen oxides concentration (parts per 10 million).

**+** **zn**: proportion of residential land zoned for lots over 25,000 sq.ft.

**+** **age**: proportion of owner-occupied units built prior to 1940

**+** **dis**: weighted mean of distances to five Boston employment centres

**Factor 2: High Alert Zone (RC5)**

**+ crim**: per capita crime rate by town

**+ tax**: full-value property-tax rate per \$10,000

**Factor 3: Small Dwellings (RC2)**

**+ rm**: average number of rooms per dwelling

**- lstat**: lower status of the population (percent)

**Factor 4: High Pupil-Teacher Ratio (RC4)**

**+ ptratio**: Pupil-teacher ratio by town

**Factor 5: High Black Population (RC3)**

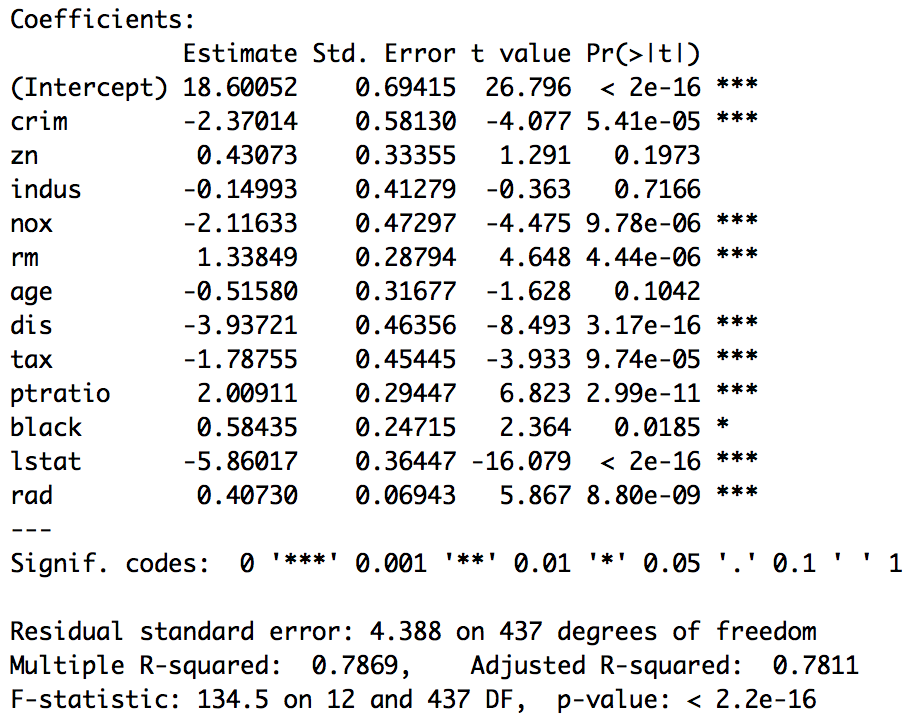
**+ Black**: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

\*[‘+’ indicates Relatively Strong Positive correlation and ‘-’ Relatively Strong Negative Correlation]

## **Predict using Multiple regression with all variables**

Now that we have performed PCA and post PCA have extracted factors using **varimax** **rotation**, we will see the effect it will have on our model as compared to the model that we could generate without factor analysis.

The multiple regression model using all the variables was created earlier and it expressed **78% of the variation** but some of the variables were highly significant and there was a **high multi-collinearity** between some of the variables which was impacting the model.



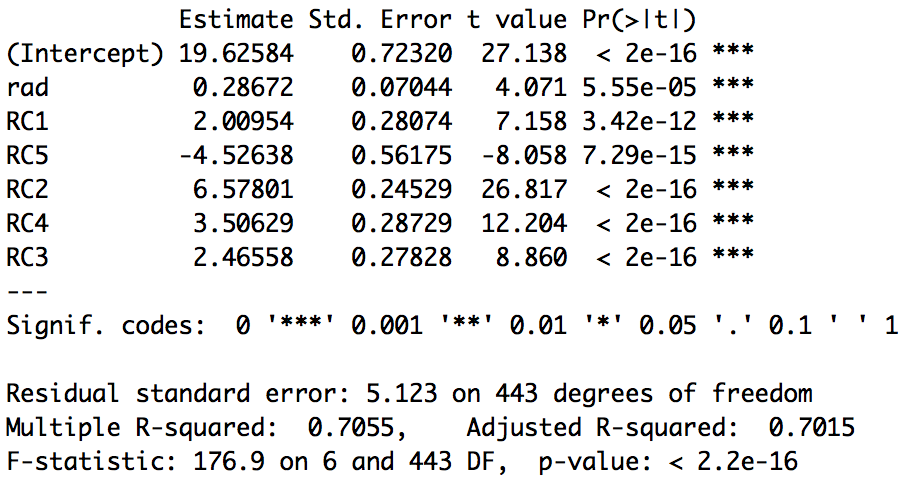
When we validate this model with our test data it gives good prediction for a few data points but for many of them the deviation is **Large** in ratio terms.

This model is good with the training data and looks better on the outset, but this is not a true depiction of the **medv** prediction model as the test data suggests. This is happening because of high multi-collinearity between some of the variables which is impacting the equation.

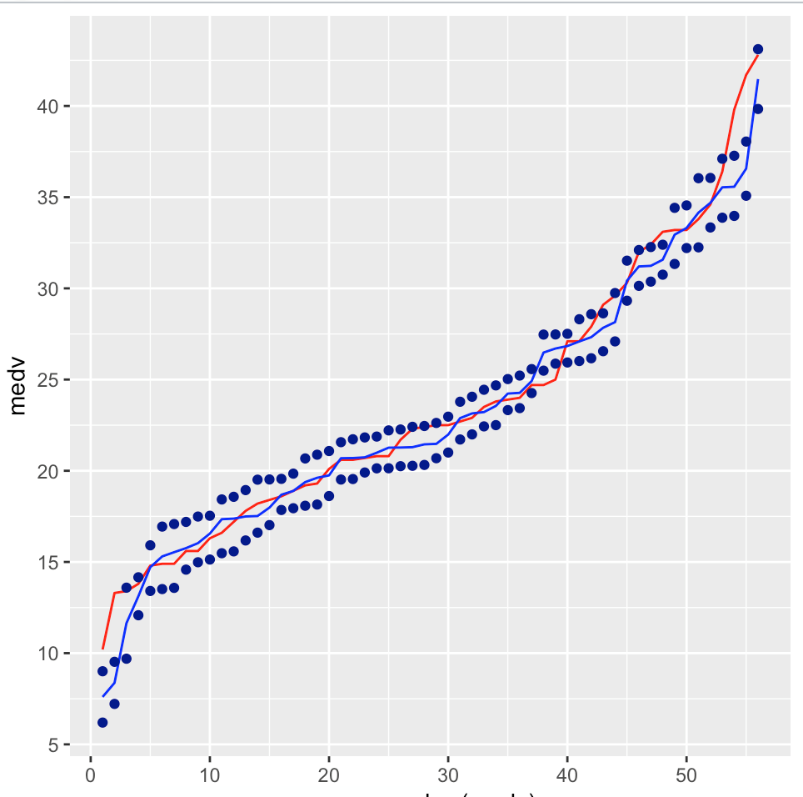
The model is used to predict the mean housing prices for the test data. **The root mean square error** for the test data is: **3.743696**

## **Predict using Multiple regression with Factors & RAD**

The multiple regression model using the factors and rad expresses 70% of the variation. The coefficient of each variable is < 0.05 and can be included in the regression model.



The model is used to predict the mean housing prices for the test data. The root mean square error for the test data is: **3.512422**. The actual values are between the confidence interval of the predicted value. Hence the model is robust.

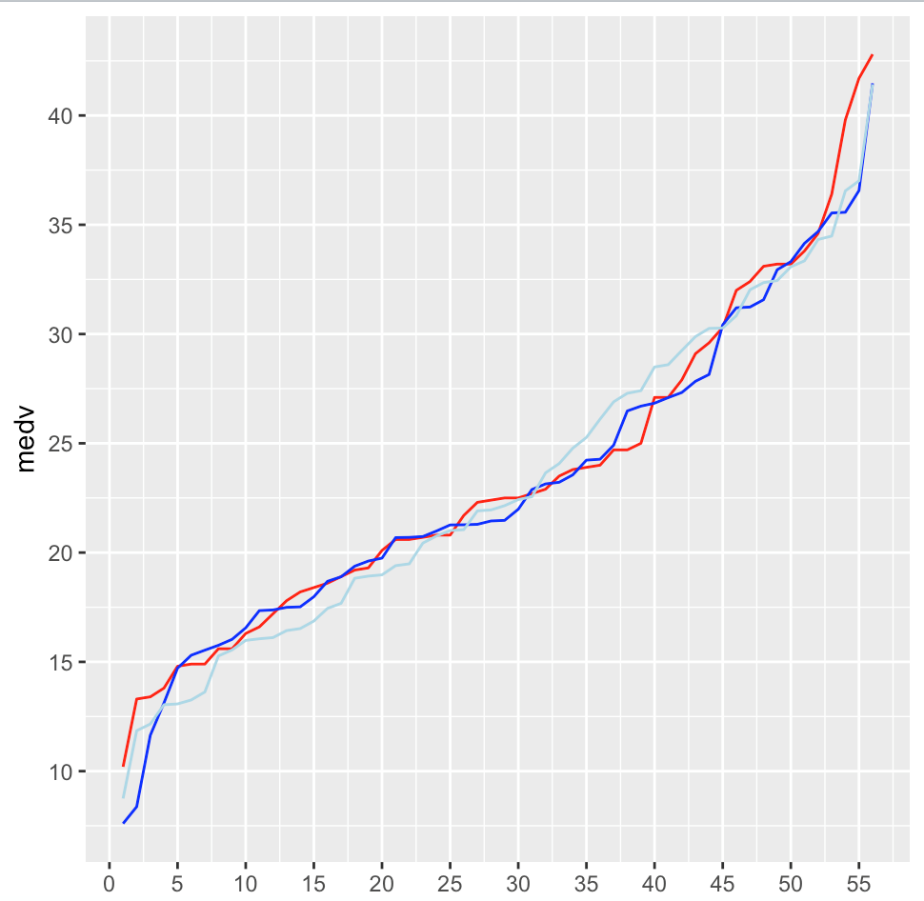


Red – **Actual**, Blue – **Predicted**, Dotted Blue – **Upper & Lower limit**

# Conclusion

* The root mean square error for the test data using the multiple regression model with no factors is: **3.743696**
* The root mean square error for the test data using the multiple regression model with factors is: **3.512422**

Following is the depiction of the confidence Interval of actual vs predicted (fit, upper and lower limit):



**The regression model with the below factors best explains the relationship with the mean house prices in Boston.**

* Factor 1: **City Outskirts**
* Factor 2: **High Alert Zone**
* Factor 3: **Small Dwellings**
* Factor 4: **High Pupil-Teacher Ratio**
* Factor 5: **High Black Population**
* Proximity to the highway: **RAD**

# R Script

The following is R-Script that we used to perform our analysis [Step by Step]:

library(MASS)

library(psych)

library(dplyr)

library(car)

library(corrplot)

library(ggplot2)

library(reshape2)

**# Exploratory Data Analysis**

head(Boston)

dim(Boston)

str(Boston)

**# Take a copy of the Boston data set and start to manipulate it**

cBoston <- Boston

size <- nrow(cBoston)

**# ignore chas as stated by the**

cBoston <- cBoston[,-4]

**# Crim, Zn, Indus, nox, rm, dis, tax, lstat are positively skewed**

**# age, ptratio, black are negatively skewed**

**# Outliers exists in crim, zn, chas, black**

summary(cBoston)

describe(cBoston)

multi.hist(cBoston)

boxplot(cBoston)

**# Can treat skew with the following transformations:**

**# Positive Skew: log10(1+x), Negative Skew: -log10(1+abs(x))**

for(i in c(1,2,3,4,5,7,9,12)) {

cBoston[,i] <- log10(1+cBoston[,i])

}

for(i in c(6,10)) {

cBoston[,i] <- -log10(1+abs(cBoston[,i]))

}

describe(cBoston)

multi.hist(cBoston)

**# We can scale the data as the variances are widely different**

names(cBoston)

var(cBoston[,-c(8, 13)])

medv <- cBoston$medv

rad <- cBoston$rad

cBoston <- data.frame(scale(cBoston[,-c(8, 13)]))

cBoston$medv <- medv

cBoston$rad <- rad

**# lets split the data into train & test**

set.seed(1000)

indexes = sample(size,450)

train = cBoston[indexes,]

test = cBoston[-indexes,]

**# Removing dataset from memory**

rm(cBoston)

**# Variables are correlated**

**# Justify for a dimension reduction technique**

names(train)

r <- round(cor(train[,-c(8,13)]),2)

corrplot(r, order = "hclust", tl.col='black', tl.cex=.75)

**# Some of the variables are highly significant**

**# age & indus can be omitted from linear regression**

lm <- lm(train$medv~., train)

summary(lm)

**# Ideally we can remove variables one by one by using variable inflation factor but we shall use PCA & Factor Analysis for the same**

vif(lm)

**# Predict using the linear regression equation**

pMedv\_lm <- predict(lm, test, interval = "confidence")

**# RMSE - Root mean square error to determine the accuracy of the model**

rmseLMwithTrain <- sqrt( sum( ((lm$fitted.values - train$medv)^2) ) / nrow(train) )

rmseLMwithTest <- sqrt( sum( ((pMedv\_lm[,1] - test$medv)^2) ) / nrow(test) )

rmseLMwithTest

**# 3 PCs explain 75% of the variation and including those which have variances > 1**

names(train)

pcs <- prcomp(train[,-c(12,13)], scale = FALSE)

summary(pcs)

pcs$rotation

plot(pcs, type="l", main="Scree Plot", col="blue")

**# Factor Analysis**

noRotation <- principal(train[,-c(12,13)], nfactors=4, rotate="none")

noRotation

**# Increasing the factors to five as the grouping does not seem right and we would like to cover at least 87% of the variation**

**# Factor 1: City Outskirts (RC1)**

**# - indus: proportion of non-retail business acres per town**

**# - nox: nitrogen oxides concentration (parts per 10 million).**

**# + zn: proportion of residential land zoned for lots over 25,000 sq.ft.**

**# + age: proportion of owner-occupied units built prior to 1940**

**# + dis: weighted mean of distances to five Boston employment centres**

**# Factor 2: High Alert Zone (RC5)**

**# crim: per capita crime rate by town**

**# tax: full-value property-tax rate per \$10,000**

**# Factor 3: Small Dwellings (RC2)**

**# rm: average number of rooms per dwelling**

**# lstat: lower status of the population (percent)**

**# Factor 4: High Pupil-Teacher Ratio (RC4)**

**# ptratio: pupil-teacher ratio by town**

**# Factor 5: High Black Population (RC3)**

**# Black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town**

**# Let's use factory analysis with varimax rotation to understand which dimension can be grouped together**

variMaxRotation <- principal(train[,-c(12,13)], nfactors=5, rotate="varimax")

variMaxRotation

round(cov(variMaxRotation$scores),2)

names(train)

train <- data.frame(train, variMaxRotation$scores)

**# create model based on factors**

colnames(train)

fLM <- lm(train$medv~., train[,c(12:18)])

summary(fLM)

**# Prepare the test data with Factor Weights**

colNames <- colnames(variMaxRotation$weights)

colNames

for (i in colNames) {

test[,i] <- variMaxRotation$weights[1,i] \* test$crim +

variMaxRotation$weights[2,i] \* test$zn +

variMaxRotation$weights[3,i] \* test$indus +

variMaxRotation$weights[4,i] \* test$nox +

variMaxRotation$weights[5,i] \* test$rm +

variMaxRotation$weights[6,i] \* test$age +

variMaxRotation$weights[7,i] \* test$dis +

variMaxRotation$weights[8,i] \* test$tax +

variMaxRotation$weights[9,i] \* test$ptratio +

variMaxRotation$weights[10,i] \* test$black +

variMaxRotation$weights[11,i] \* test$lstat

}

names(test)

pMedv\_flm <- predict(fLM, test[,c(13:18)],interval = "confidence")

rmseFactorswithTrain <- sqrt( sum( ((fLM$fitted.values - train$medv)^2) ) / nrow(train) )

rmseFactorswithTest <- sqrt( sum( ((pMedv\_flm[,1] - test$medv)^2) ) / nrow(test) )

**"R^2 for linear regression and linear regression by using factor"**

summary(lm)

summary(fLM)

**"RMSE without PC:"**

rmseLMwithTrain

rmseLMwithTest

**"RMSE with Facors:"**

rmseFactorswithTrain

rmseFactorswithTest

actual.predicted.test <- data.frame(medv = test$medv,

pMedv.f = pMedv\_flm[,1],

pMedv.wf = pMedv\_lm[,1],

pMedv\_lw = pMedv\_flm[,2],

pMedv\_up = pMedv\_flm[,3])

**# Compare the actual vs predicted with factors and without factors**

ggplot(data = actual.predicted.test) +

geom\_line(aes(x = row\_number(medv), y = medv), colour = "Red") +

geom\_line(aes(x = row\_number(pMedv.f), y = pMedv.f), colour = "Blue") +

geom\_line(aes(x = row\_number(pMedv.wf), y = pMedv.wf), colour = "Light Blue") +

scale\_x\_continuous(breaks = seq(0,200, 5)) +

scale\_y\_continuous(breaks = seq(0,100, 5))

**# Confidence Interval of actual vs predicted (fit, upper and lower limit)**

ggplot(data = actual.predicted.test) +

geom\_line(aes(x = row\_number(medv), y = medv), colour = "Red") +

geom\_line(aes(x = row\_number(pMedv.f), y = pMedv.f), colour = "Blue") +

geom\_point(aes(x = row\_number(pMedv\_lw), y = pMedv\_lw), colour = "Dark Blue") +

geom\_point(aes(x = row\_number(pMedv\_up), y = pMedv\_up), colour = "Dark Blue") +

scale\_x\_continuous(breaks = seq(0,100, 10)) +

scale\_y\_continuous(breaks = seq(0,100, 5))