**Question 11.1**

**Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model using: 1. Stepwise regression 2. Lasso 3. Elastic net For Parts 2 and 3, remember to scale the data first – otherwise, the regression coefficients will be on different scales and the constraint won’t have the desired effect. For Parts 2 and 3, use the glmnet function in R.**

**Answer:** Givenbelow are the steps to perform regression model using the 3 variable selection method.

**Step 1**: Load the data in the data frame and run the required necessary packages.

Code:

*set.seed(1)*

*uscrime <- read.table("C://Users/D100793/Desktop/Junk/Georgia Tech/uscrime.txt", header = TRUE,sep = '\t')*

*head(uscrime)*

*install.packages("MASS")*

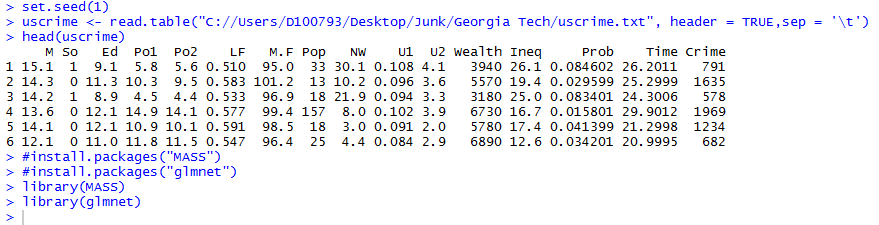
*install.packages("glmnet")*

*library(MASS)*

*library(glmnet)*

*library(caret)*

Output:



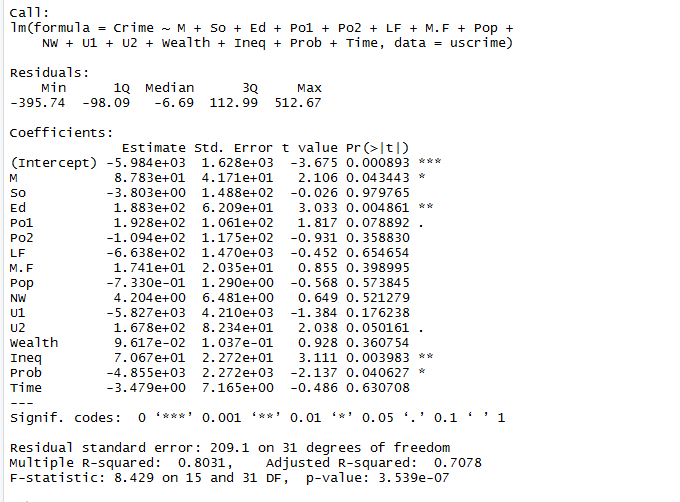
**Step 2**: Applying Linear Regression to the entire model

Code:

*LinearModel <- lm(Crime~M+So+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time , data = uscrime)*

*summary(LinearModel)*

Output:



Analysis: The R-Squared value shows that the model is 80% accurate and M, Ed, Ineq and Prob are significant variables. But however there is no assurance that the model we chose is the best model and above that our model is not even simple. So lets try choosing relevant variables using different variable selection methods in the further steps below.

**Step 3**: Performing stepwise variable selection on the linear regression model.

**Backward Elimination Method:**

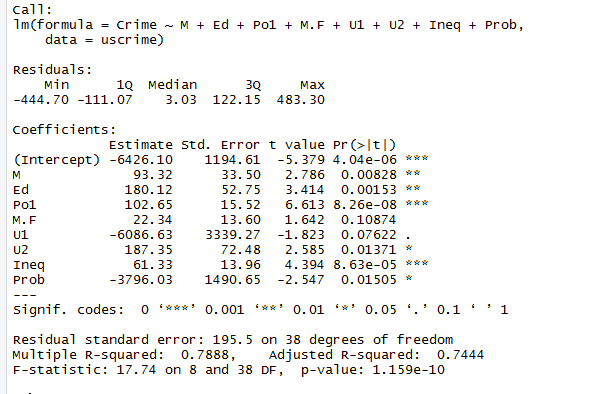
Code:

*#Applying backward elimination to the Linear Model*

*stepRegBackward <- stepAIC(LinearFit, direction = 'backward', trace = FALSE)*

*summary(stepRegBackward)*

Output:



Analysis: By using the backward elimination method the Linear Model is now restricted to 8 variables and all of them except M.F appears to be insignificant. U1 also has a low p value.

Cross Validating the Backward Elimination Model

Code:

*stepRegBackward\_CV = cv.lm(uscrime,stepRegBackward,m = 4)*

*n = length(uscrime$Crime)*

*avg = mean(uscrime$Crime)*

*SSE<-0*

*SSR<-0*

*SST<-0*

*for(i in 1:n){*

*SST = SST + (uscrime$Crime[i] - avg)^2*

*SSE = SSE + (uscrime$Crime[i] - stepRegBackward\_CV$cvpred[i])^2*

*SSR = SSR + (stepRegBackward\_CV$cvpred[i] - avg)^2*

*}*

*SSE*

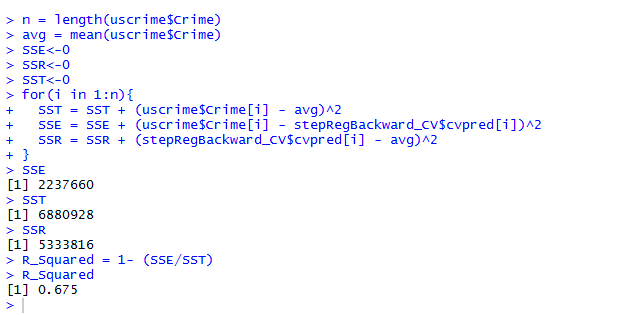
*SST*

*SSR*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

Output:



Analysis: After applying 4 fold cross-validation to the data We see the R-Squared changing from 78% to 67.5 percent which is still a good model and a simpler model as compared to the original linear fit.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Forward Selection Method:**

Code:

#Applying Linear regression with minimul variables

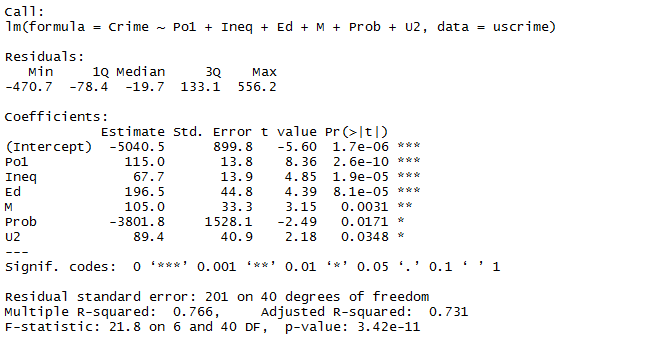
*min\_lm <- lm(Crime ~ 1, data = uscrime )*

#Applying Forward Selection to the Linear Model

*stepRegForward <- step(min\_lm , direction = "forward", scope = (~ M+So+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time) , steps = 1000 ,trace = FALSE )*

*summary(stepRegForward)*

Output:



**Analysis:** The Looking at the forward selection model, the selection model picked up 6 variables and all variables are significant with p-values less than 0.05

Cross Validating the Forward Selection Model

Code:

*stepRegForward <- stepAIC(LinearFit, direction = 'forward', trace = FALSE)*

*summary(stepRegForward)*

*stepRegForward\_CV = cv.lm(uscrime,stepRegForward,m = 4)*

*n = length(uscrime$Crime)*

*avg = mean(uscrime$Crime)*

*SSE<-0*

*SSR<-0*

*SST<-0*

*for(i in 1:n){*

*SST = SST + (uscrime$Crime[i] - avg)^2*

*SSE = SSE + (uscrime$Crime[i] - stepRegForward\_CV$cvpred[i])^2*

*SSR = SSR + (stepRegForward\_CV$cvpred[i] - avg)^2*

*}*

*SSE*

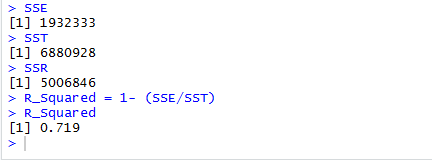
*SST*

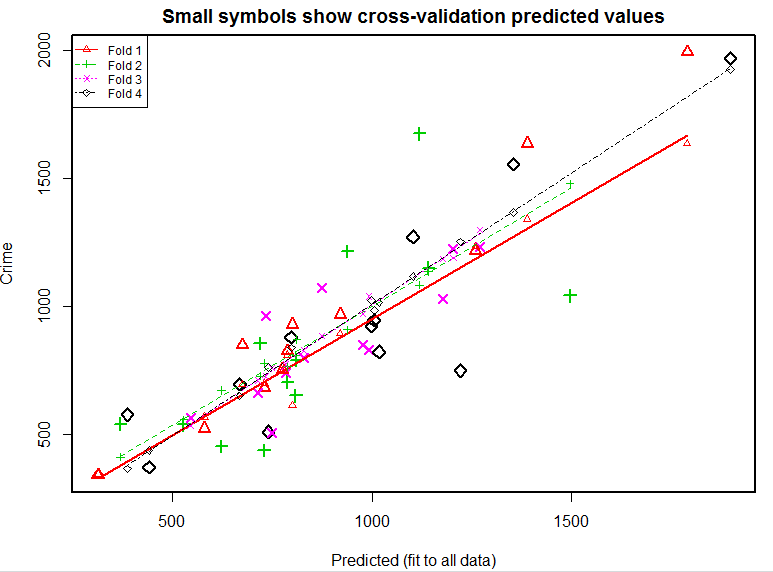
*SSR*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

Output:





**Analysis**: The accuracy of the model has now improved with an R-Squared of 71.9 %. This model only picks up variables that are significant and give us a high accuracy. The model is also relatively simpler.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Both Forward Selection and Backward Elimination Method**

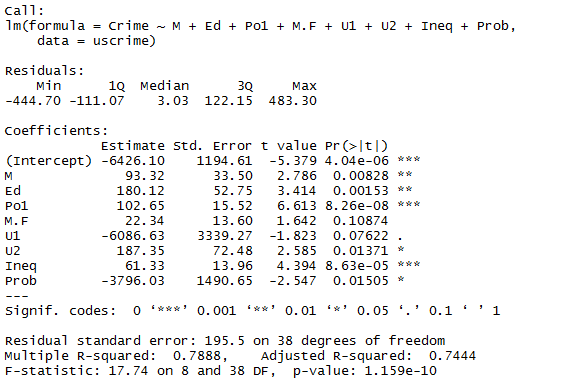
Code:

*#Applying Both Selection to the Linear Model*

*stepRegBoth <- stepAIC(LinearFit, direction = 'both', trace = FALSE)*

*summary(stepRegBoth)*

Output:



Analysis: Combination of both forward selection and backward elimination ended up with 8 variables and 2 variables appear to be not significant. Like backward elimination model this model has same R-Squared

Cross Validating the combination of Backward Elimination and Forward Selection Method:

Code:

*stepRegBoth\_CV = cv.lm(uscrime,stepRegBoth,m = 4)*

*n = length(uscrime$Crime)*

*avg = mean(uscrime$Crime)*

*SSE<-0*

*SSR<-0*

*SST<-0*

*for(i in 1:n){*

*SST = SST + (uscrime$Crime[i] - avg)^2*

*SSE = SSE + (uscrime$Crime[i] - stepRegBoth\_CV$cvpred[i])^2*

*SSR = SSR + (stepRegBoth\_CV$cvpred[i] - avg)^2*

*}*

*SSE*

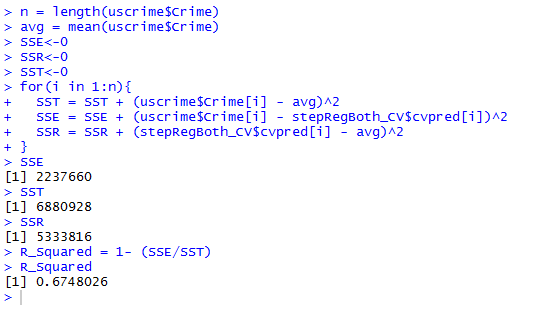
*SST*

*SSR*

*R\_Squared = 1- (SSE/SST)*

*R\_Squared*

OUTPUT:



Analysis: After applying 4 fold cross-validation to the data We see the R-Squared changing from 78% to 67.5 percent which is still a good model and a simpler model as compared to the original linear fit.

Though we have cross validated the data, lets calculate the AIC and BIC of all the stepwise models just as n additional step to see if it is a good criterion to measure the accuracy of the model.

Code:

*AIC(stepRegBackward)*

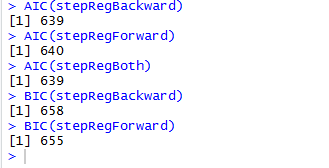
*AIC(stepRegForward)*

*AIC(stepRegBoth)*

*BIC(stepRegBackward)*

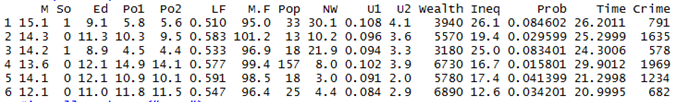
*BIC(stepRegForward)*

Output:



All AIC & BIC are fairly close. So it’s not a good criterion.

**Step 4**: Performing LASSO Variable selection on the dataset.



In order to perform LASSO regression, we first need to scale data. When we look at the nature of the data the predictor “SO” looks like a binary categorical variable which need not be scaled. We also don’t need to scale the response as well. So given below is the code for scaling rest of the data.

Code:

*#Seprating the values to be unscalled*

*UnscaledValues <- c("So","Crime")*

*#scaling the data*

*df <- uscrime[ , !(names(uscrime) %in% UnscaledValues)]*

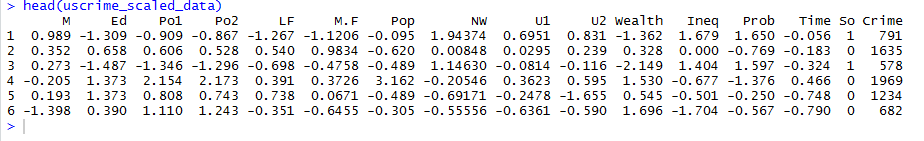
*Scaled\_Values<- scale(df)*

*#binding the data altogether.*

*uscrime\_scaled\_data <- cbind(Scaled\_Values, uscrime[,UnscaledValues])*

*head(uscrime\_scaled\_data)*

Output



Now applying the Lasso Model to the scaled Data

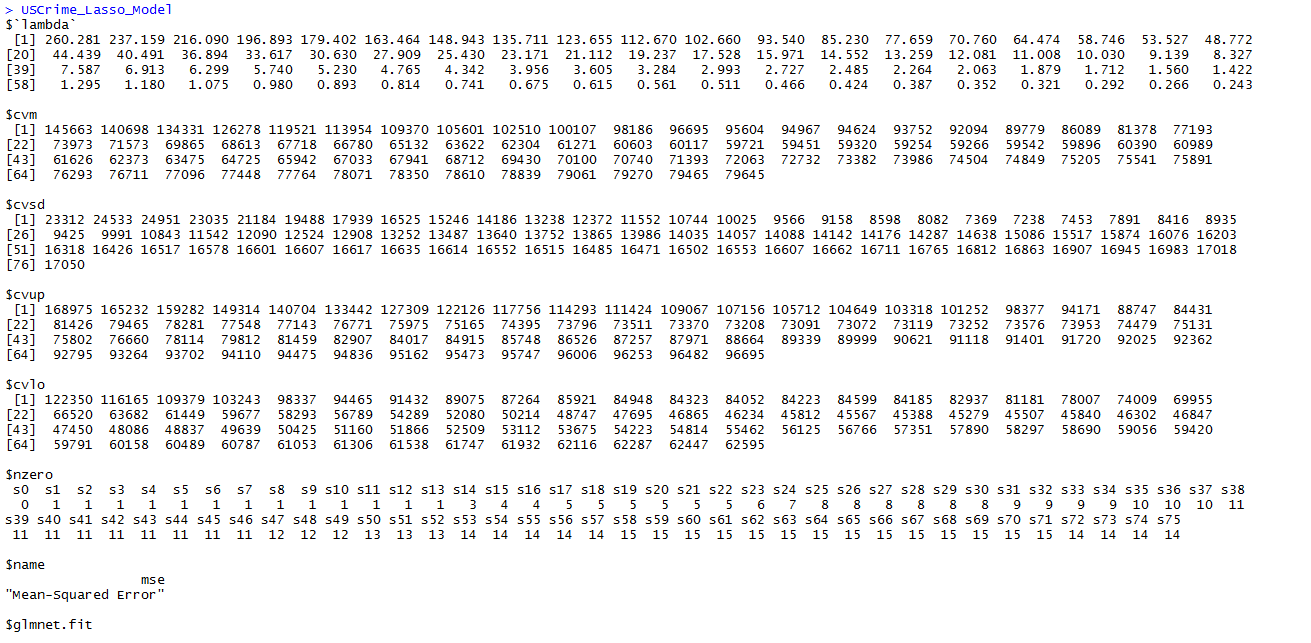
Code:

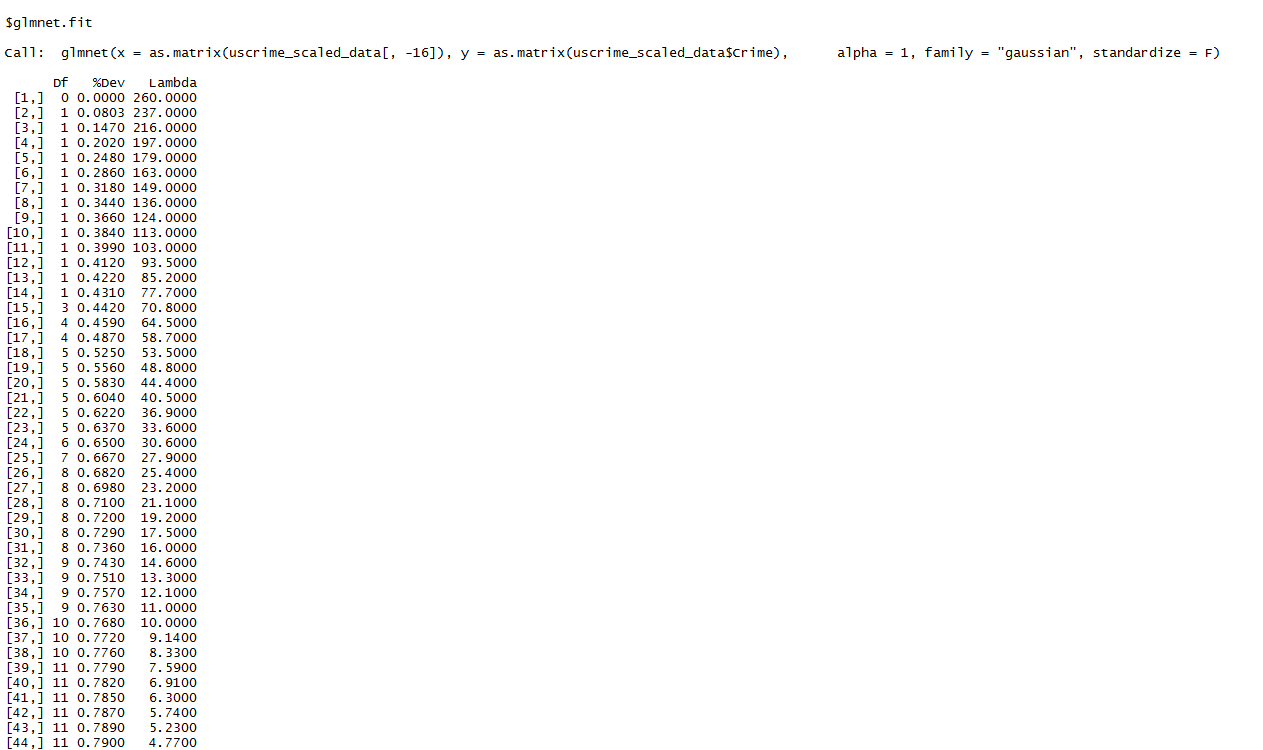
*USCrime\_Lasso\_Model = cv.glmnet(x = as.matrix(uscrime\_scaled\_data[,-16]),y = as.matrix(uscrime\_scaled\_data$Crime),*

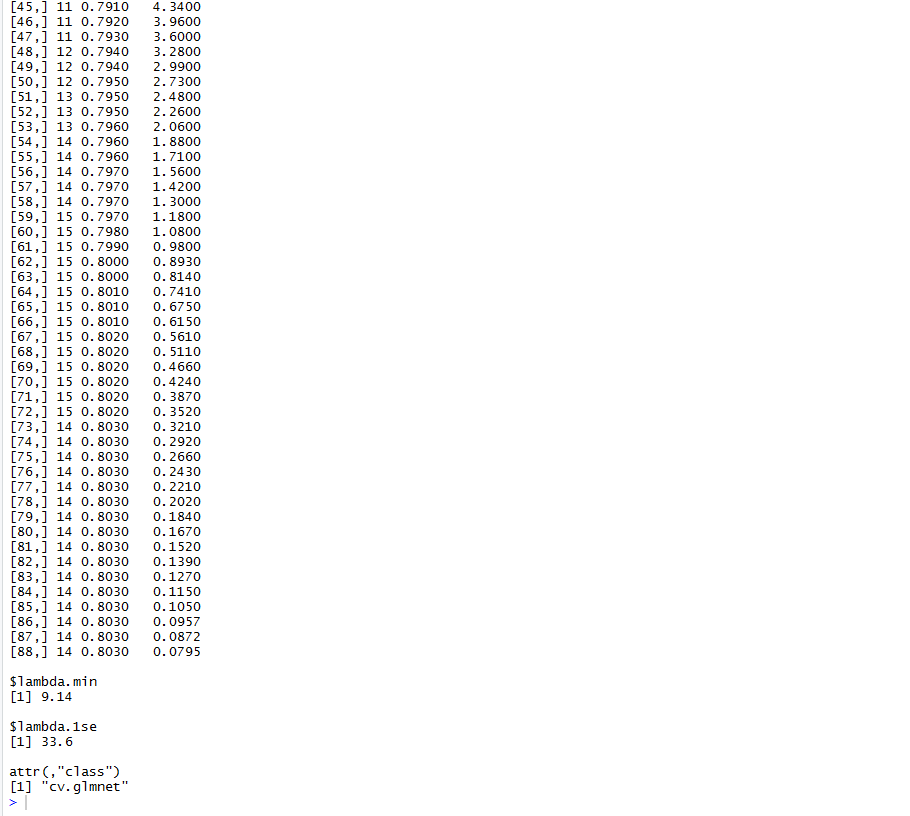
*alpha = 1, nfolds = 5,.measure = "mse", family = "gaussian", standardize = F)*

*USCrime\_Lasso\_Model*

Output:





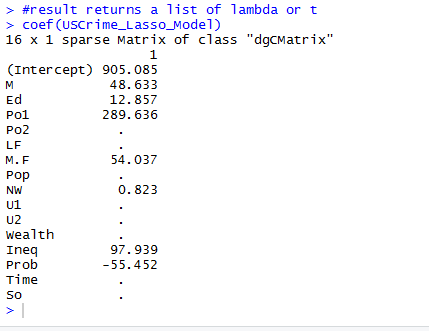


Lets look at the coefficients of the Lasso Model and check which variables are selected by the model

Code:

*coef(USCrime\_Lasso\_Model)*

Output:



The lasso model says only 5 variables are significant. They are M, PO1, M.F, Ineq and Prob

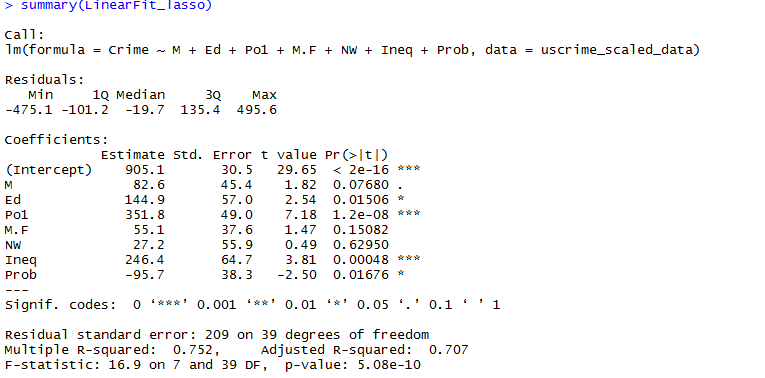
Running Linear regression on the above coefficients

Code:

*LinearFit\_lasso <- lm(Crime ~ M +Ed+ Po1 + M.F +NW+ Ineq + Prob, data = uscrime\_scaled\_data)*

*summary(LinearFit\_lasso)*

Output:



Looking at the above output, we see that by choosing that Lasso method has chosen only 5 variables and the results yield a R-Squared of 75.2% against the 78% as compared to the original linear fit in the first step. This model is much simpler with lesser variable. All the variable have a good p-value which makes them look significant except NW and M.F.

However we would like to refine this model by choosing a value of lambda from which error is minimum.

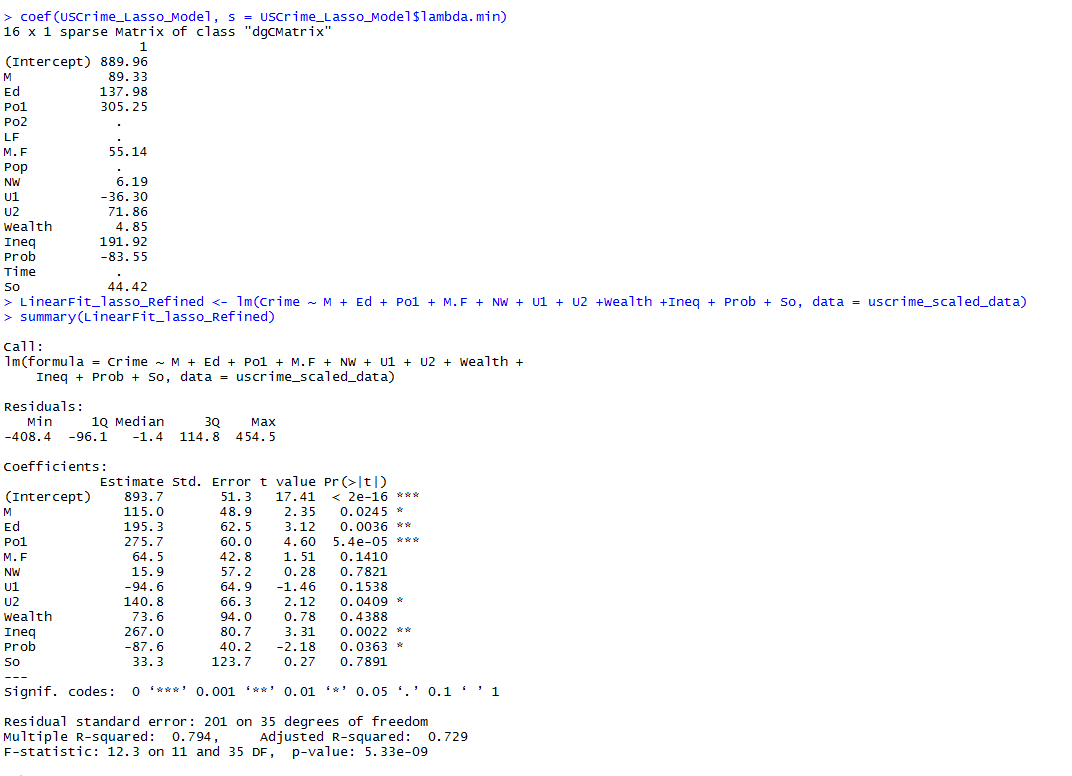
Code:

*coef(USCrime\_Lasso\_Model, s = USCrime\_Lasso\_Model$lambda.min)*

*LinearFit\_lasso\_Refined <- lm(Crime ~ M + Ed + Po1 + M.F + NW + U1 + U2 + Ineq + Prob + So, data = uscrime\_scaled\_data)*

*summary(LinearFit\_lasso\_Refined)*

Output:



Looking at the above output , it appears that the number of variable selection increased in our refined model but the good accuracy increased to 79.4% (R-Squared) which shows that the model is highly accurate. 4 variables are maked as insignificant but they they do contribute to the accuracy of the model.

**Step 5**: Performing Elastic Net Variable selection on the linear regression model.

We can use the existing scaled data. Lets loop over different values of alpha. In each loop we select, minimum cross validation error and lambda.min

Code:

*list\_values <- numeric()*

*find\_alpha <- function(num, uscrime\_scaled\_data){*

*alpha <- num*

*elastic\_net <- cv.glmnet(x=as.matrix(uscrime\_scaled\_data[,-16]),*

*y=as.matrix(uscrime\_scaled\_data[,16]),*

*alpha = alpha,*

*nfolds=5,*

*type.measure="mse",*

*family="gaussian",*

*standardize=FALSE)*

*list\_values <<- cbind(list\_values, c(alpha, min(elastic\_net$cvm),elastic\_net$lambda.min))*

*}*

*#looping over different values of alpha#looping*

*for (i in seq(.01,1,by = .01)){find\_alpha(i,uscrime\_scaled\_data)}*

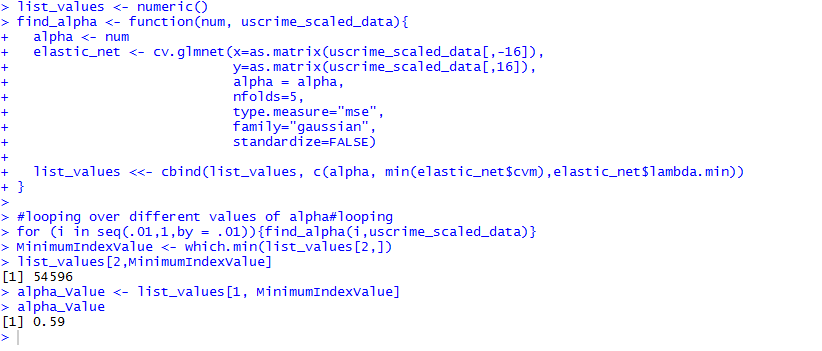
*MinimumIndexValue <- which.min(list\_values[2,])*

*list\_values[2,MinimumIndexValue]*

*alpha\_Value <- list\_values[1, MinimumIndexValue]*

*alpha\_Value*

OUTPUT:



We get an alpha value as 0.59. Lets use alpha = 0.89 and running the elastic net

Code:

*USCrime\_ElasticNet\_Model <- cv.glmnet(x=as.matrix(uscrime\_scaled\_data[,-16]),*

*y=as.matrix(uscrime\_scaled\_data[,16]),*

*alpha = alpha\_Value,*

*nfolds=5,*

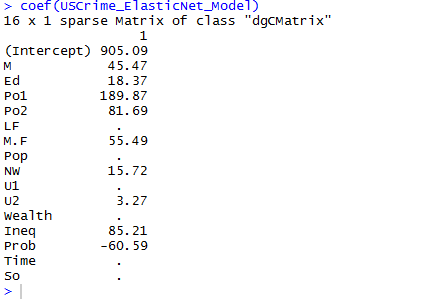
*type.measure="mse",*

*family="gaussian",*

*standardize=FALSE)*

*coef(USCrime\_ElasticNet\_Model)*

OUTPUT:



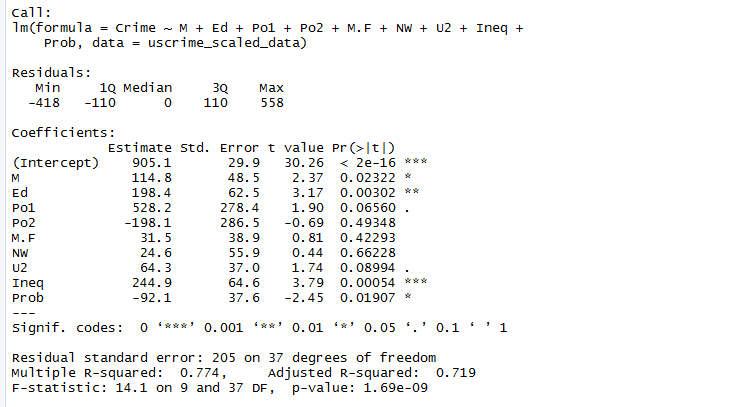
As we look at the above output, it appears that Elastic Net model has chosen 9 variables for Linear Regression. Lets run the Linear regression using the variables chosen by ElasticNet Model.

Code:

*LinearFit\_ElasticNet <- lm(Crime ~ M +Ed+ Po1 + Po2 + M.F +NW+U2 +Ineq + Prob, data = uscrime\_scaled\_data)*

*summary(LinearFit\_ElasticNet)*

OUTPUT:



Analysis: When you look at the output above we see the model is simplified to 8 variables and out of the 8, 5 variables are significant. The R squared value is 78 % which means the model is 78% accurate which is good value and the model is fairly simple.

However we would like to refine this model by choosing a value of lambda from which error is minimum.

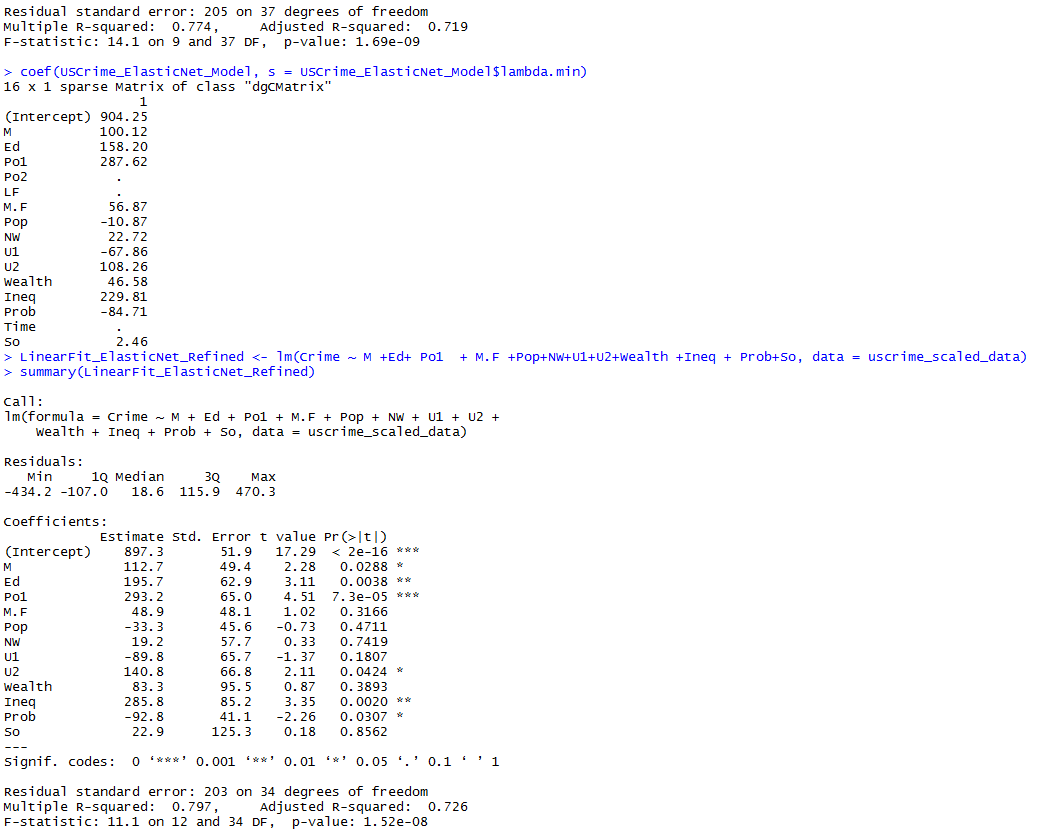
Code

*coef(USCrime\_ElasticNet\_Model, s = USCrime\_ElasticNet\_Model$lambda.min)*

*LinearFit\_ElasticNet\_Refined <- lm(Crime ~ M +Ed+ Po1 + M.F +Pop+NW+U1+U2+Wealth +Ineq + Prob+So, data = uscrime\_scaled\_data)*

*summary(LinearFit\_ElasticNet\_Refined)*

Output



The Accuracy has again improved as compared to the non-refined elastic model. The number of variables chosen has increased with more than 5 non-significant variables. But higher R-Squared value makes this model better.

Given below is the summary of all the analysis done.

