**Module 3 – LASSO Regression in R Practice**

**Name: Sunil Raj Thota**

**NUID: 001099670**

**Date: 02/03/2021**

**Title: LASSO Regression in R Practice**

**ALY 6015 – Intermediate Analytics**

**Prof. Roseanna Hopper**

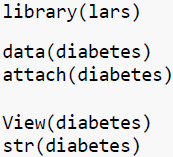
**Introduction**

In this task, I used R and R studio to perform Descriptive Statistics, Linear Regression, and Regularization techniques on various preloaded datasets provided by CRAN. We can notice a list of all data sets in R using the function called data().

The dataset used for the Part A analysis is the “diabetes” to discover the cross validation, regularization, and lm models on thes x, x2, and y variables. Also, used additional packages such as Used lars, glmnet, biglasso, bigmemory, and ncvreg libraries.

**Analysis**

**Exercise 1**

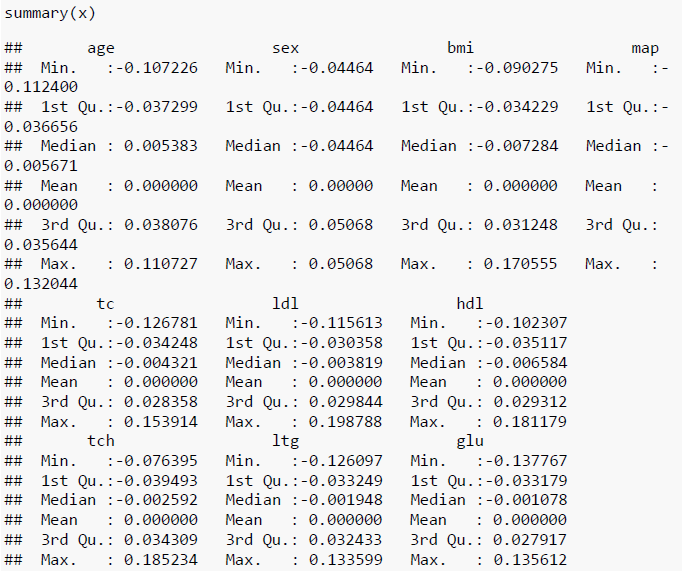
****

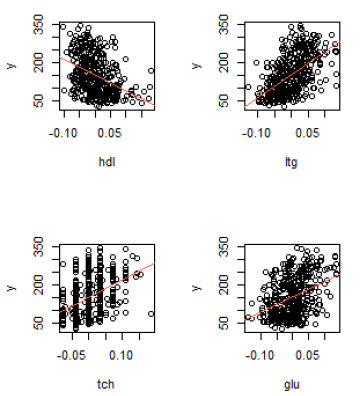
Let's perform some Regularization analysis and techniques using "diabetes" data set. To get this data set we need to install the 'lars' package from the packages tab which is right side to the work space in R Studio. Or we can also install the packages by using install.packages("lars") command. Once it is loaded, we can use it in the code for further analysis and calculations.

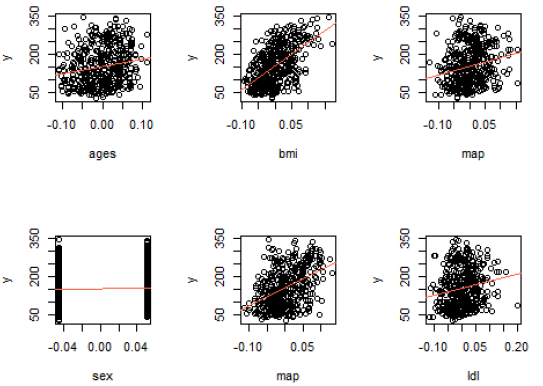
Loaded the "lars" library into the work space. Loaded the diabetes Data set into the Environment. To reduce the repetitive usage of "x" in "diabetes" data set, "attach" is used to set it once throughout the work space. To View the diabetes Data set we use View() command, To observe the structure of the Data set we use str() command, and head () and tail() shows first and last few rows in the Data set

**Exercise 2**

Summary() provides the Descriptive Stats of the x variable in diabetes Data set. Once this is done, let's find out the list of elements which are present in "x" using looping concept (for loop). We noticed 10 variables from the statistics given in the summary. Now, let's plot Scatter plots for these variables and stack them in 2 rows and 5 columns.

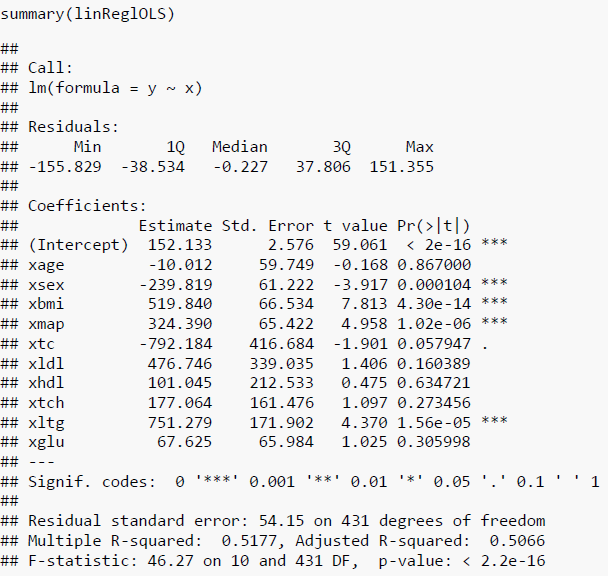


To generate separate Scatter Plots with the line of best fit for all the predictors in "x" as horizontal axis vs "y" as vertical axis. # In this for loop I used 'idx' as index value for "x" variable and increased the value of 'idx' by iterating the loop 10 times. I have included conditional statements to allocate individual x-labels to their respective plots. abline() is used to plot a linear regression line on these plots.



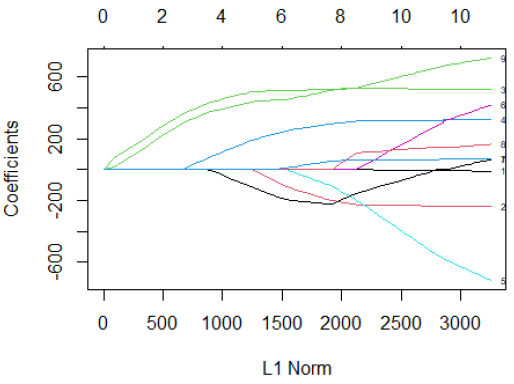
**Exercise 3**

In this, we need to regress "y" on the predictors in "x" using Ordinary Least Squares(OLS). The regression model was taken between "y" and "x". Summary() gives us the descriptive stats and hypothesis testing values like Standard Error, p-Value, t-Value, r-squared value, f-Statistic, Degrees of Freedom, and etc.,

****

This model is used as a baseline model to collate with the next upcoming models.

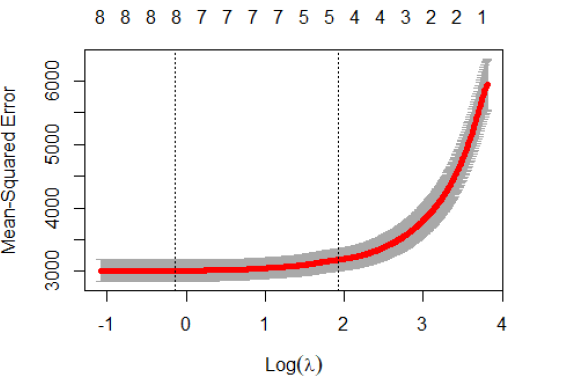
**Exercise 4**

****

LASSO regression is performed and for that to happen we use "glmnet" package from the packages tab to install or simply use install.packages("glmnet") command.

Now, let's load the "glmnet" in our work space to regularize the model using LASSO and plot it using plot(). This plot indicates at which stage each coefficient shrinks to 0. and the lines depicts the values used by various other coefficients.

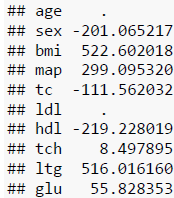
**Exercise 5**

****

Here, Cross Validation is used to get the best value of lambda and plot the curve using plot(). It is possible with cv.glmnet() method. In general, nlambda values > 100. We got min lambda as 0.8730528

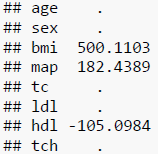
From the plot we can depict that the value of lambda increased when the number of selected variables tapers down.

**Exercise 6**



Here, we use the minimum lambda value again in glmnet() function to get the best fit. There are 3 coefficients namely age, ldl, tch whose values have become 0. It's clear that these variables are not so necessary to determine the value of "y".

**Exercise 7**

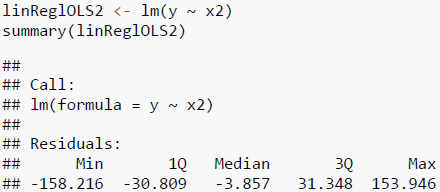
**

****

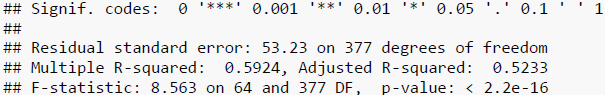
Here, we use the minimum lambda value again in glmnet() function to get the best latest fit # There are 6 coefficients namely age, sx, tc, ldl, tch, and glu whose values have become 0. It's clear that these variables are not so necessary to determine the value of "y". LASSO tells that only 4 variables are necessary on which y depends. Thus, the shrinkage increases.

**Exercise 8**

In this, we need to regress "y" on the predictors in "x2" using Ordinary Least Squares(OLS). The regression model was taken between "y" and "x"2

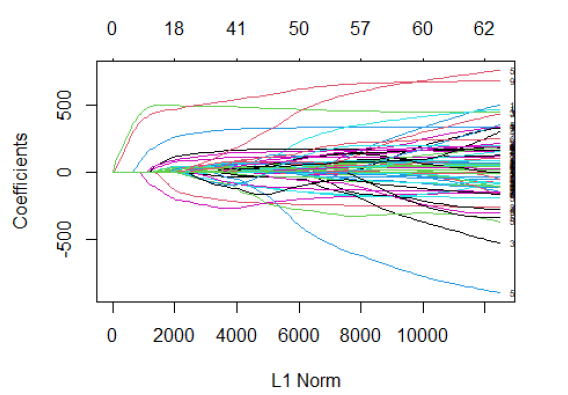


Summary() gives us the descriptive stats and hypothesis testing values like Standard Error, p-Value, t-Value, r-squared value, f-Statistic, Degrees of Freedom, and etc., From this we can see that there are a greater number of predictors in x2 than x.



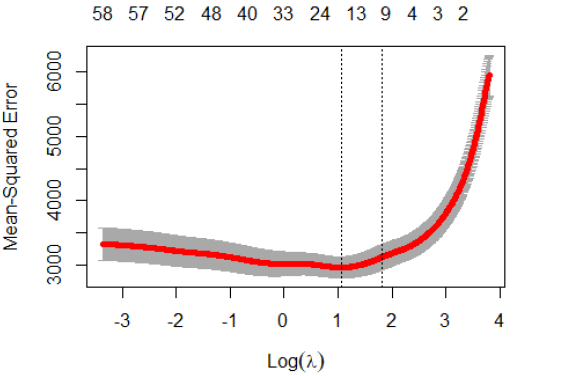
**Exercise 9**

LASSO regression is performed and for that to happen we use "glmnet" package from the packages tab to install or simply use install.packages("glmnet") command.

**

Now, let's load the "glmnet" in our work space to regularize the model using LASSO and plot it using plot(). This plot are complex and the lines depicts the values used by various other coefficients

**Exercise 10**



Here, Cross Validation is used to get the best value of lambda and plot the curve using plot(). It is possible with cv.glmnet() method

From the plot we can depict that the value of lambda increased when the number of selected variables tightens down. Now, we find the min. value of lambda to get the best fit.

Here, we use the minimum lambda value again in glmnet() function to get the best fit. There are 50 coefficients whose values have become 0. It's clear that these variables are not so necessary to determine the value of "y". With this its shrinkage's the variables and it regularizes the model.

**Conclusion**

To perform multiple analytical operations, we use R on a wider range of data like various datasets. R is a very powerful tool to perform analysis which is mainly built by researchers, statisticians, and developers. In the beginning, we did Descriptive Statistics Analysis, Exploratory Data Analysis, and continued with the Linear Regression and Regularization Techniques to get understandings from the provided data. Plotted relevant parameters, to helps us to understand the data in a clear way. Used lars, glmnet, biglasso, bigmemory, and ncvreg libraries.

**References**

[1] https://github.com/YaohuiZeng/biglasso/blob/master/vignettes/biglasso.pdf

[2] Yaohui Zeng, Patrick Breheny (11th March, 2018) The biglasso Package: A Memory- and Computation-Efficient Solver for Lasso Model Fitting with Big Data in R was retrieved from https://arxiv.org/pdf/1701.05936.pdf