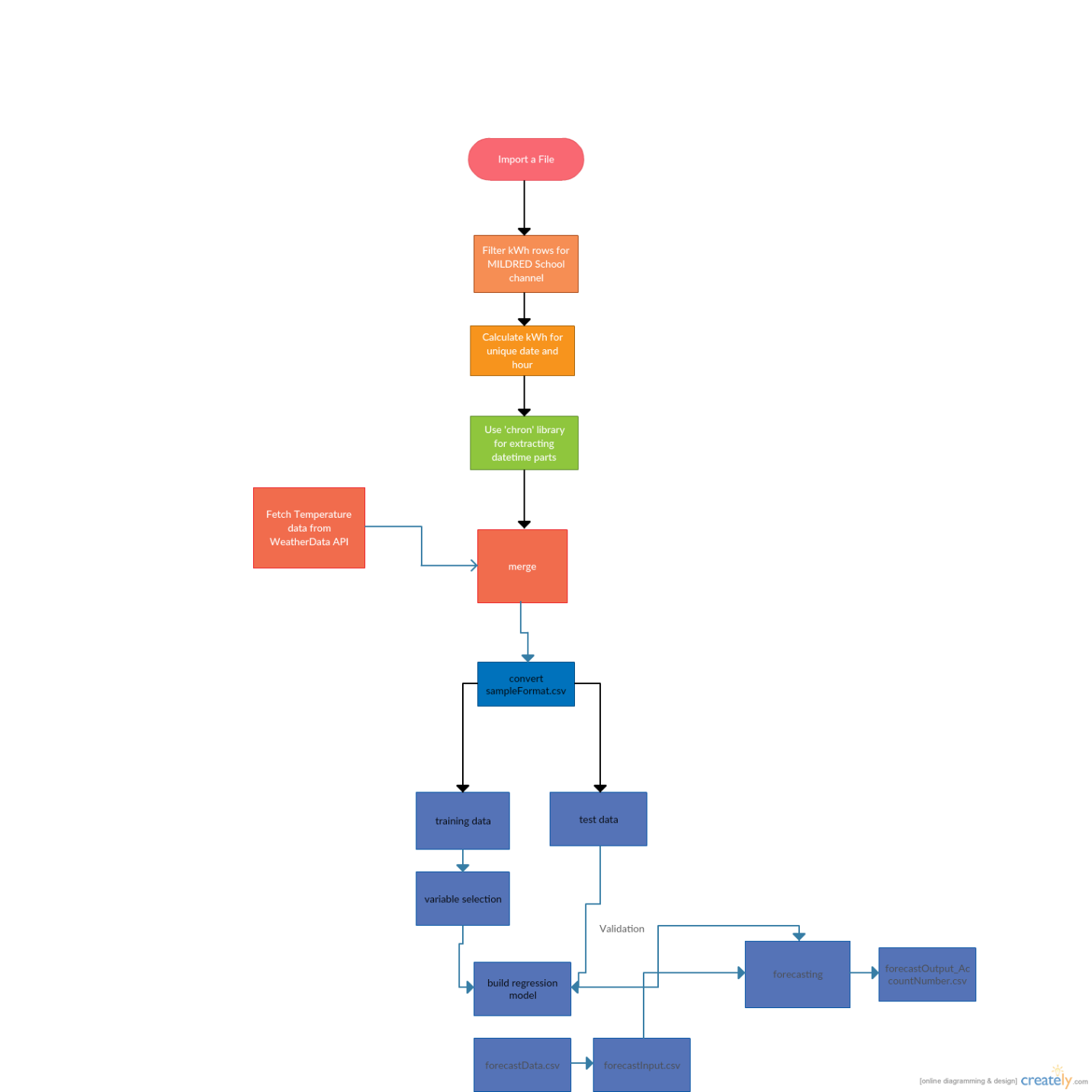
**Team members:** Hina Gandhi, Shivam Goel, Nikita Anand

**Case Study:**

The city of Boston has hired you to build a forecasting model to predict their energy usage. In order to do so, you are planning to use multi-linear regression to model Power usage as a function of multiple variables (temperature, day of week, month, weekday, hour of day etc.).

**Flowchart:**

****

**Part1 – Algorithm Implementation**

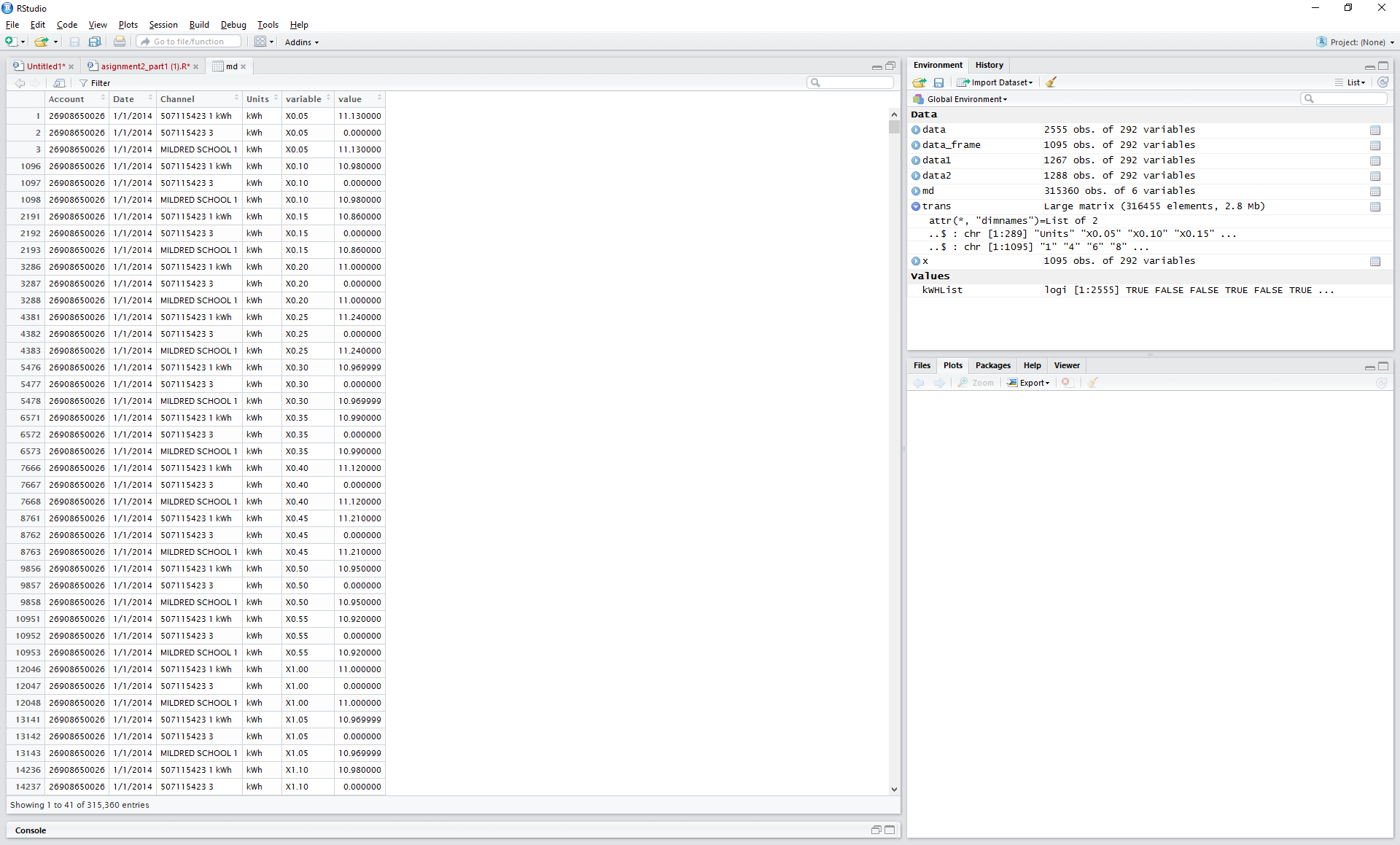
***Script****: asignment2\_part1.R*

***Packages****: chron, dplyr, weatherData, devtools, reshape2*

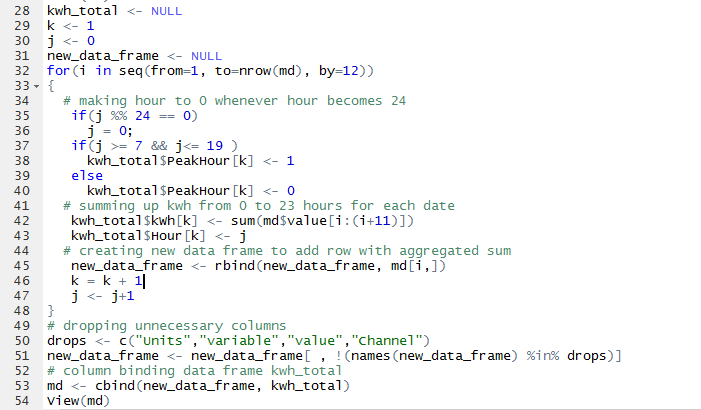
***Steps****:*

**1. Data wrangling and cleansing**

* *Set path using setwd() for the data set(rawData1.csv,rawData2.csv) required to be cleaned*
* *Read and merge the two csv for the complete data from January to December 2014 into one dataframe*
* *Consider the values of power outage with “kWh” as unit of measure for MILDRED School 1 as channel*
* *Stack the data into a proper format using* ***melt()*** *in library* ***reshape2***



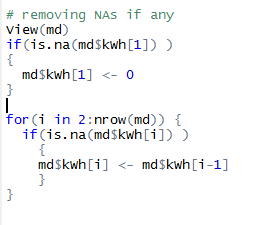
* *In order to collect hourly power data, prepare a new data frame,* ***sum up the observations for every 12 entries of kWh****. Create an* ***Hour*** *column, which states hour of that time in the format 0-23*



* *For the calculation of* ***PeakHour****, prepare an algorithm which formats the value to be assigned ‘1’ if the Hour column is between 0-19(7AM-7PM) else ‘0’ if between (7PM-7AM)*
* *In order to create columns* ***Day****,* ***Year****,* ***Month,******Day of Week, IsWeekday*** *(format used is if weekday ‘1’ will be assigned, if not ‘0’ shall be assigned), we have used* ***‘chron’*** *library*

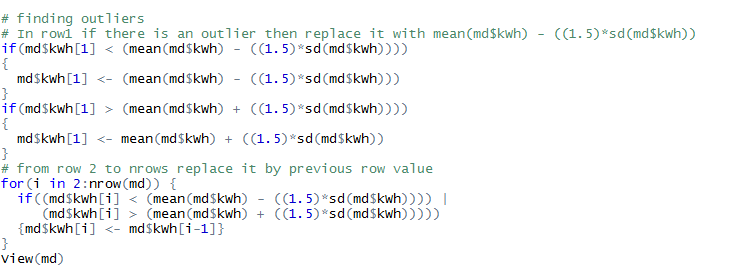
***Cleansing:***

* ***Removing NA:*** *Since we inferred the kWh for each hour, we are assuming that there will not be a significant change in the power outage over an hour’s difference. So, we have* ***replaced the NA values in the kWh column with the previous hour’s kWh observation****. Also, if the 1st observation in the dataset has NA, then we are assigning the kWh value to be ‘0’*



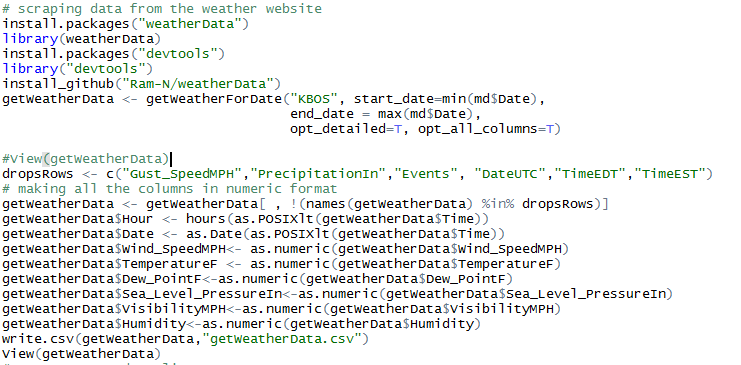
* ***Removing Outliers:*** *We found certain outliers in kWh column, which were replaced by the following formula: If an outlier is in the 1st row of the dataset and is less than* ***(mean(md$kWh) - ((1.5)\*sd(md$kWh))) replace it by the value achieved from this formula*** *(mean(md$kWh) - ((1.5)\*sd(md$kWh))).*

*For rest of the rows starting from row 2 till end, replace it with the previous observation.*



**2. wunderground API call**

* *Call the weatherData API for each date from the rawdata file we extracted before. Keep the start date for weatherData API to be min(data$Date) and end date to be max(data$Date)*
* *Convert each column data to numeric and remove the columns which are not required.*
* *Remove NA and find-remove outliers with the previous observations*



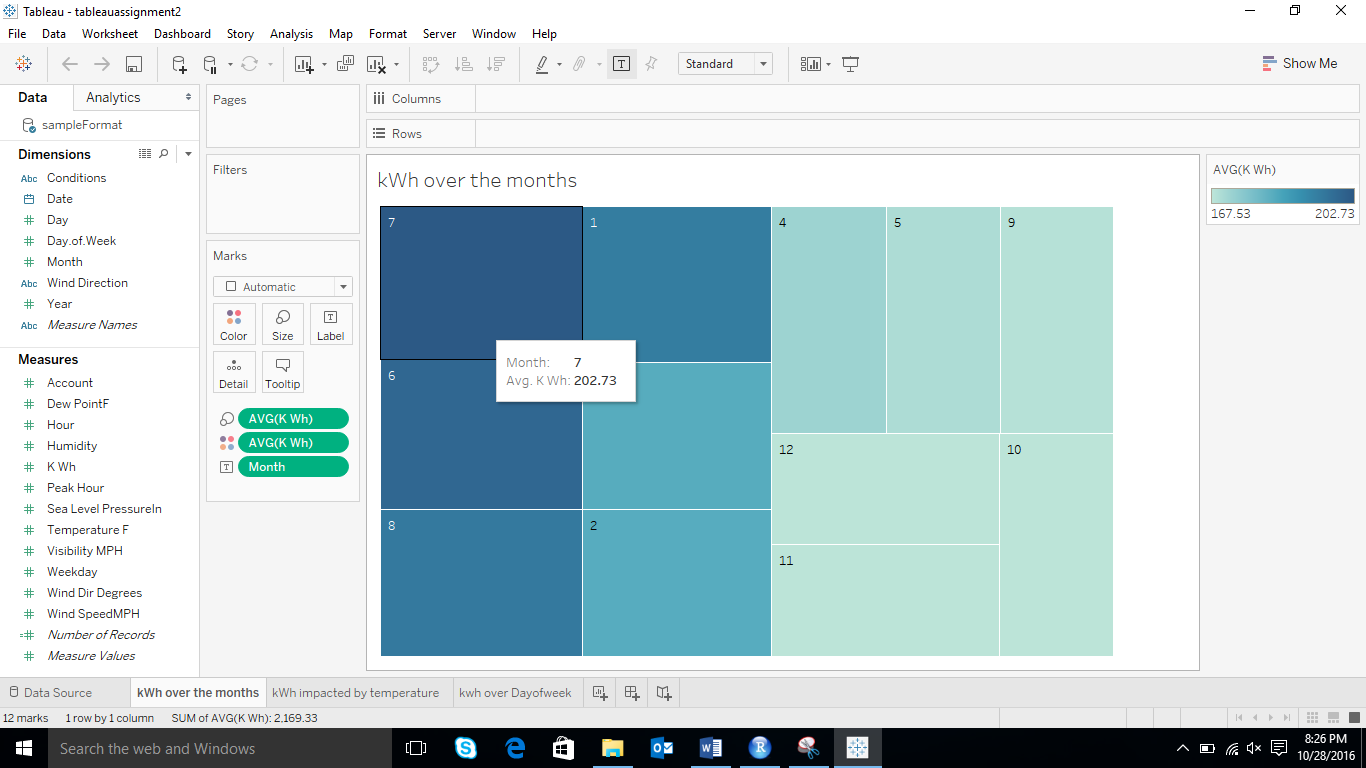
* ***Observation****: There were certain days in the data retrieved from weatherData for which we observed more than one entry of TemperatureF, Dew\_PointF etc and other numeric columns within each hour. Take aggregate for such hours of the day*



* *Combine the rest of the columns like Date, Hour, PeakHour etc using* ***‘left\_join’ in ‘dplyr’.*** *Save the entire data frame to* ***‘SampleFormat.csv’***

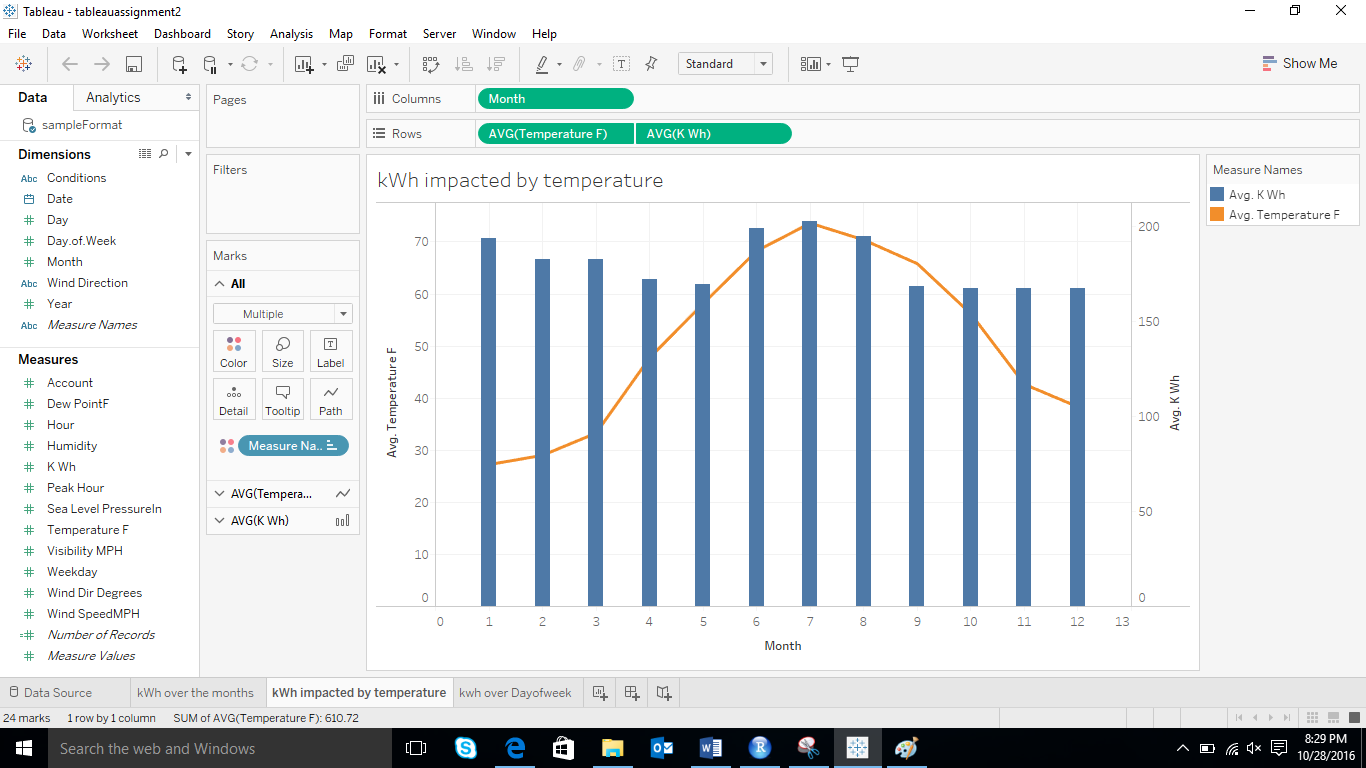
**Data Visualization Using Tableau**

**Power Outage over Months in 2014**



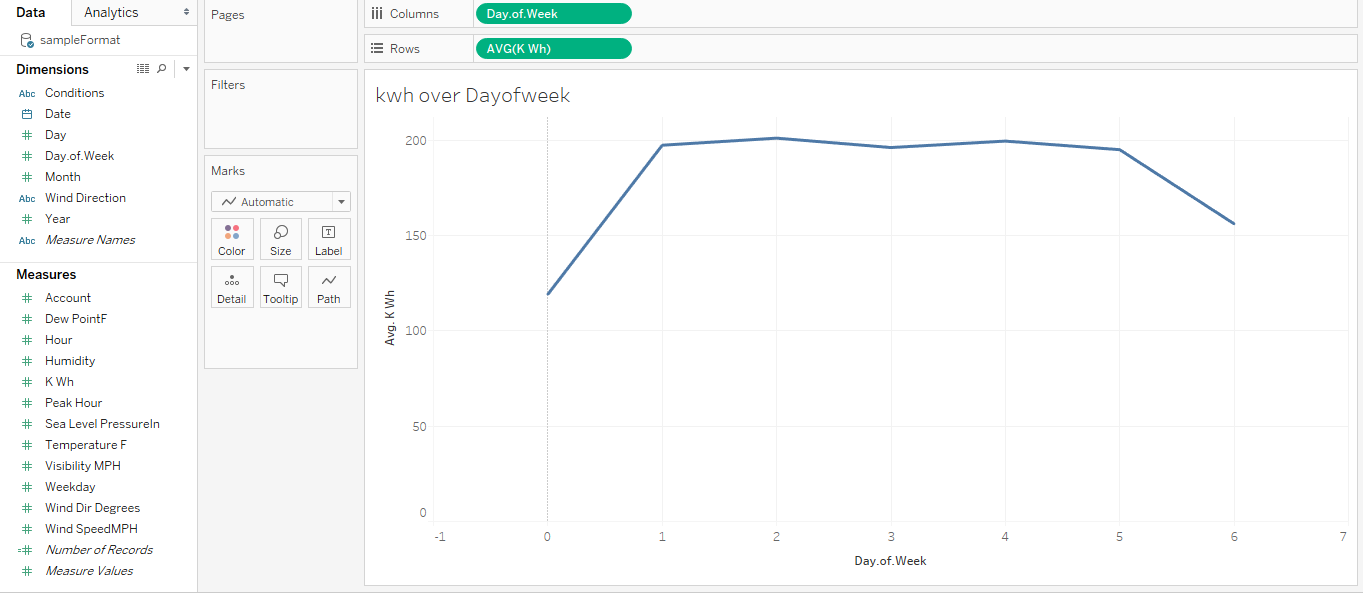
* *Power Outage was observed to be at its max during the months of July, June and August for the year of 2014*

***Power Outage Impacted by Temperature Change***



* *We can observe that as the temperature falls during the months of September-December, there is a stability in power outage, and the usage is increasing during the summer months.*

***Power Outage during Day of Week***



* *Power outage appears to be stable during weekdays.*

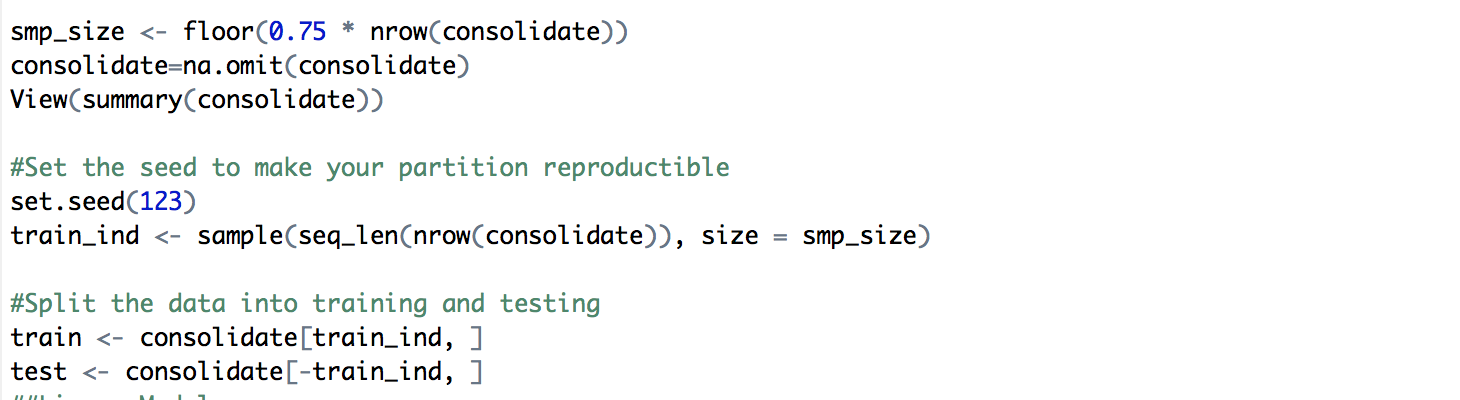
**Part2 – Multiple-Linear Regression**

***Script****: RegressionModel*

***Packages****: ISLR, leaps,* *forecast,glmnet*

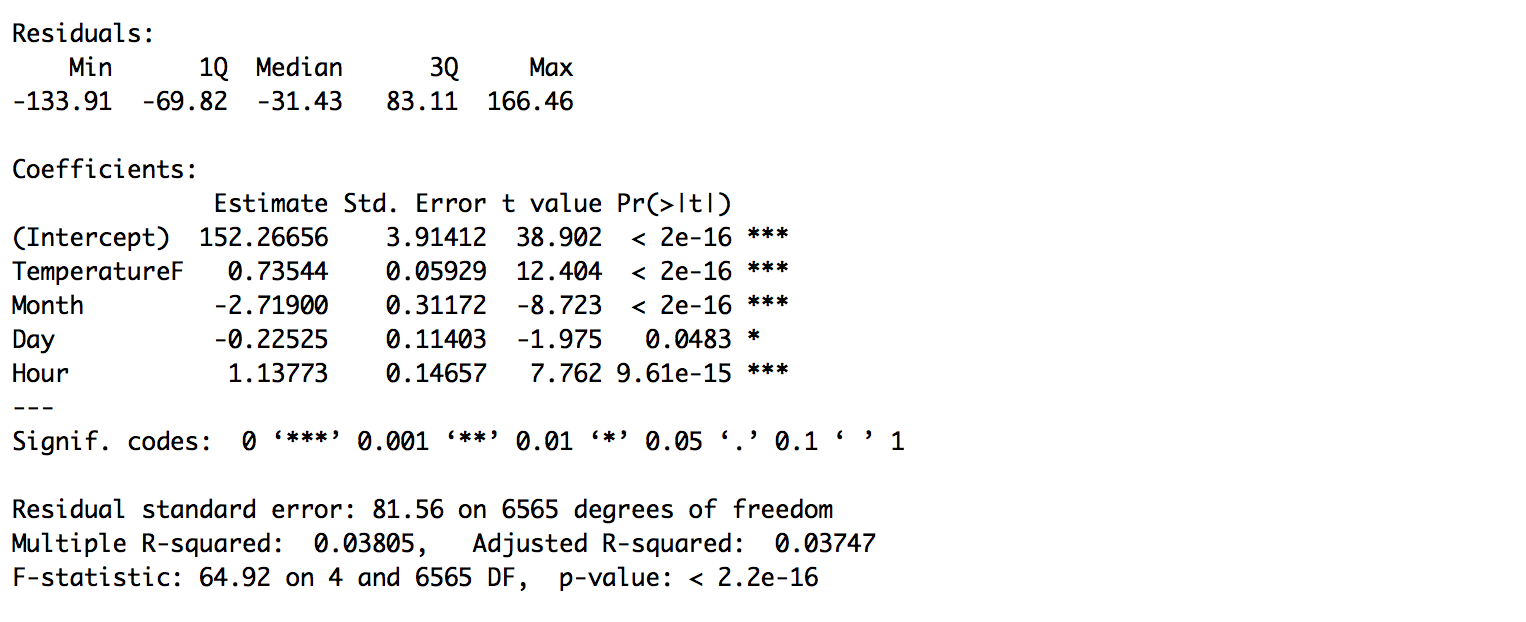
***Steps****: In this stage, we have performed the following steps to build a multiple linear regression model and select the variables with most influential regression coefficient.*

* *Verify that the sampleforecast.csv file in placed in the working directory.*
* *Next, read the csv and set the sample size to 70%*
* *Now set the seed to make the partition reproducible.*
* *Split the data to training and test data.*



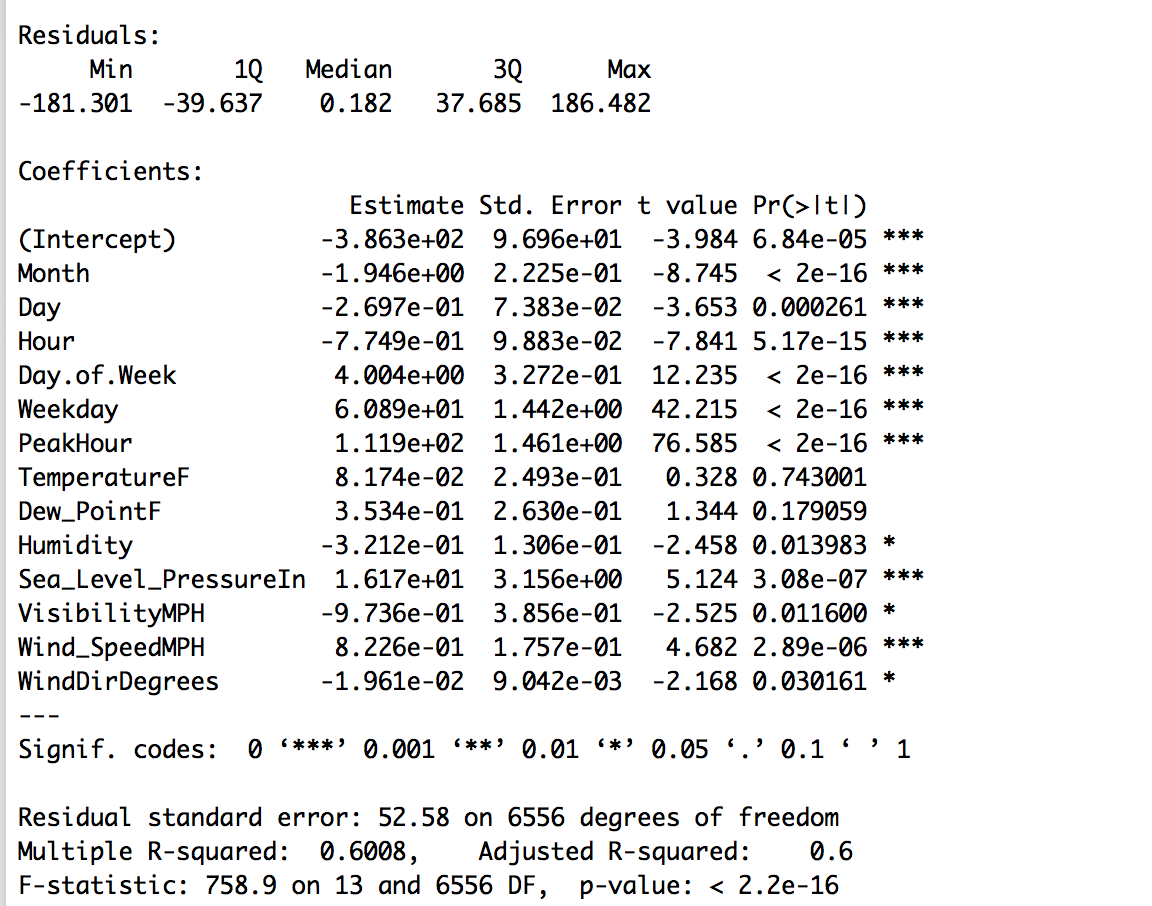
**Variable Selection**

* *Now we applied the linear regression model by selecting the response column as “kWh”, selecting factors as temperature day, month, hour.*

**

**Co-efficients, Std.errror, t-value & p-value**

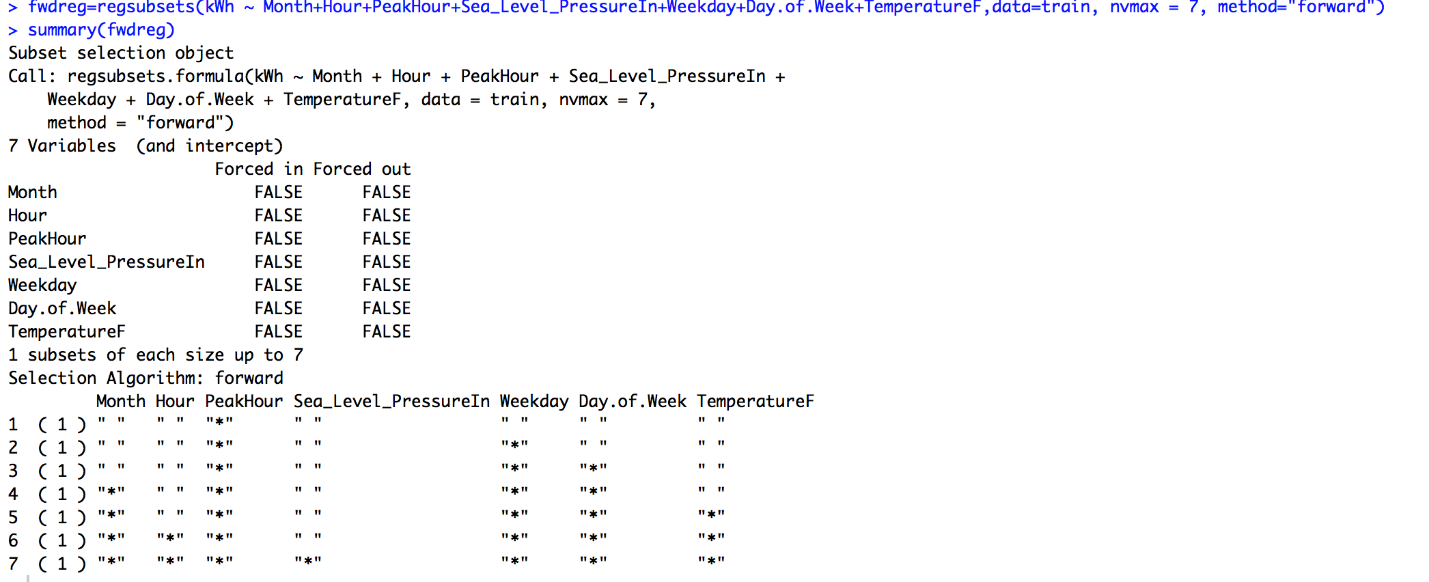
* *Now from below metrics we can see that there are lot of factors that influence power outage/consumption. We have also learned in the lecture that p-value and t-value plays an important role in determining the factors more the t-value more appropriate is the factor. As we can see that Peak Hour, Day of week, Wind Speed, Sea level Pressure plays the important role so they need to be included in model*

****

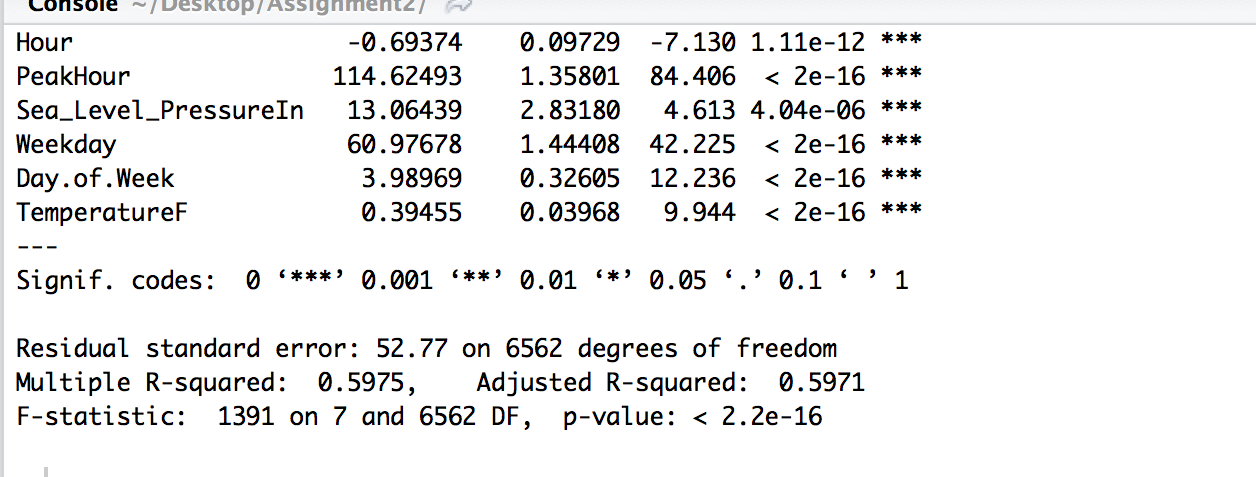
**Forward Search:**

* *Now we run forward search with 7 variables.*

*Result*

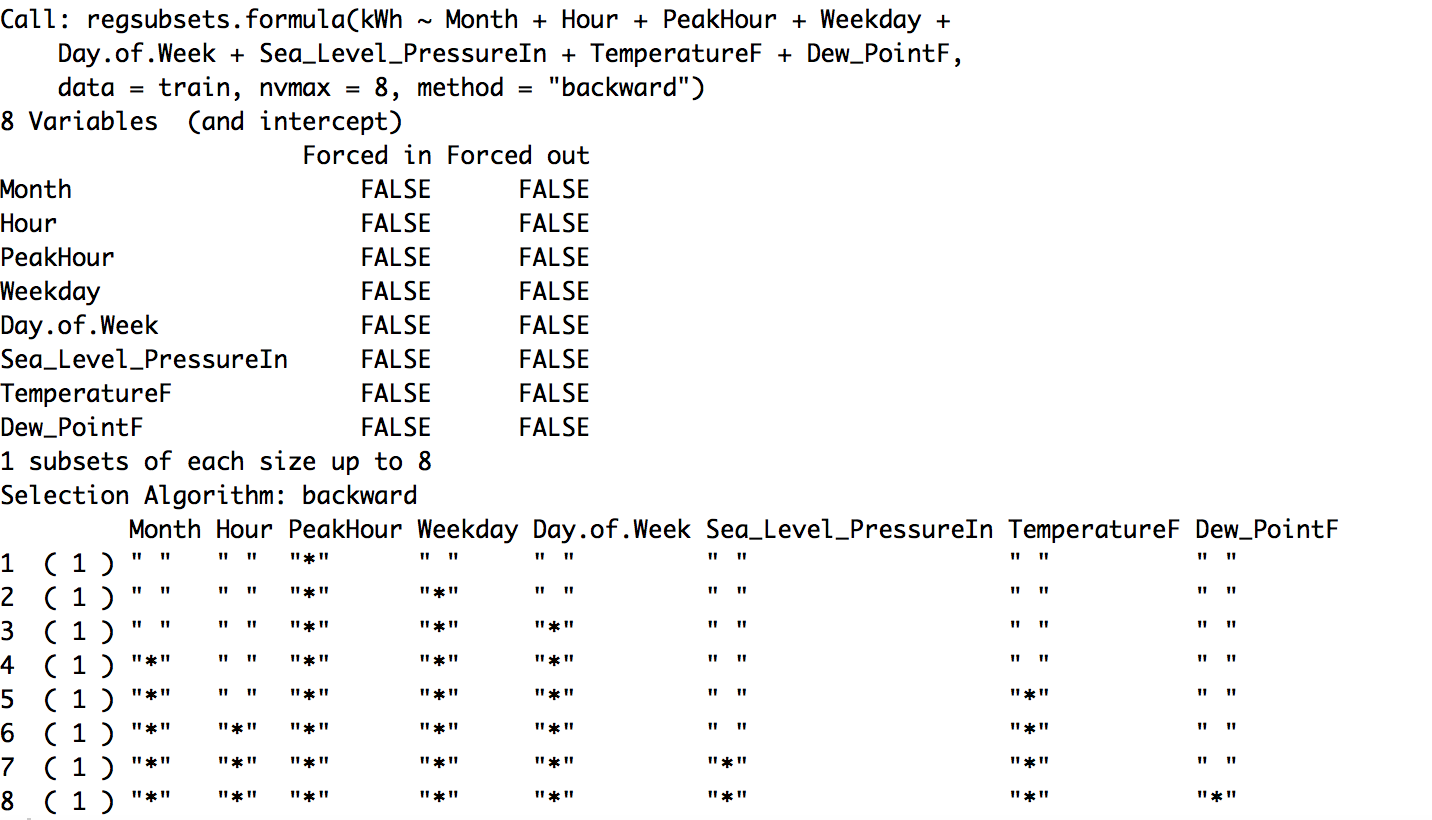
**

*we found that R square value was around .59 if we use forward regression*

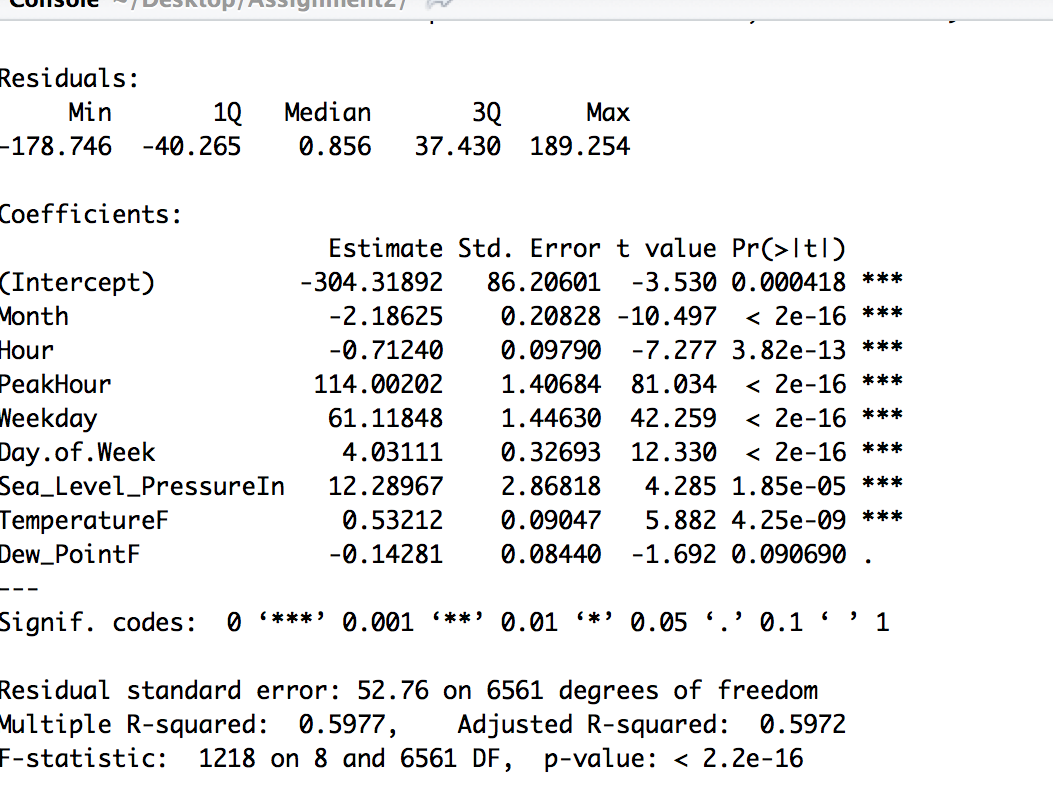
**

**Backward Search:**

*As the R square value was around .59, we thought to see if there is some improvement with backward regression. Now we run backward search with 8 variables*

**

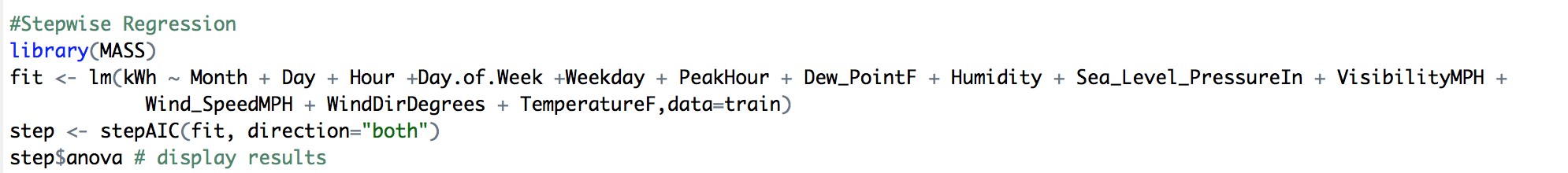
*Checking the R square value if we take parameters from backward regression:*

**

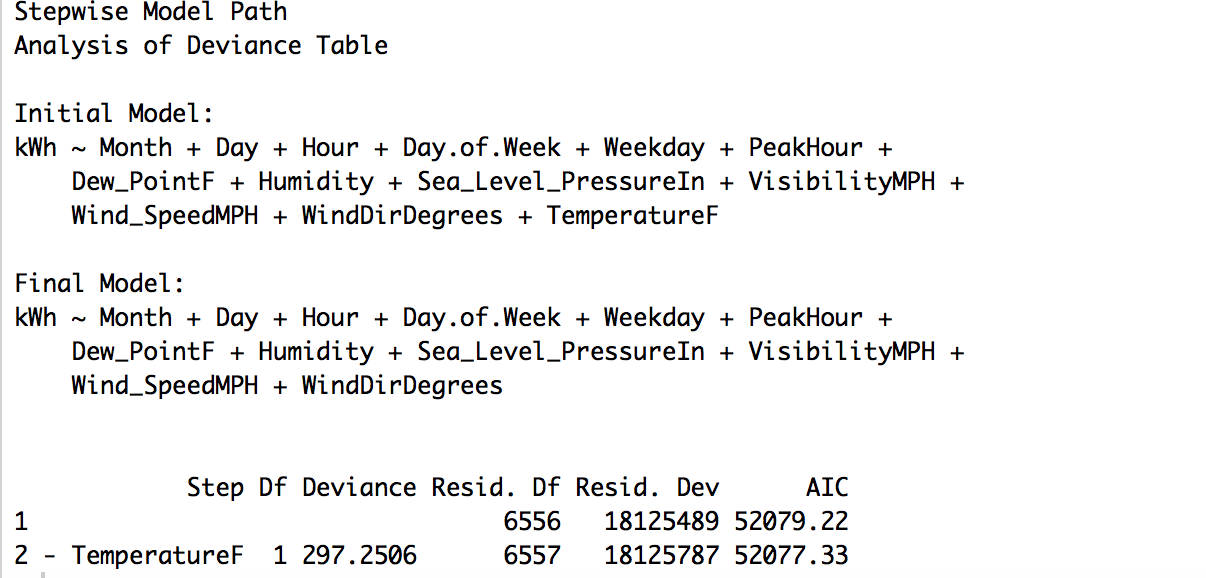
*We can see that forward and backward regression does not make much difference in R square value*

**Stepwise Search:**

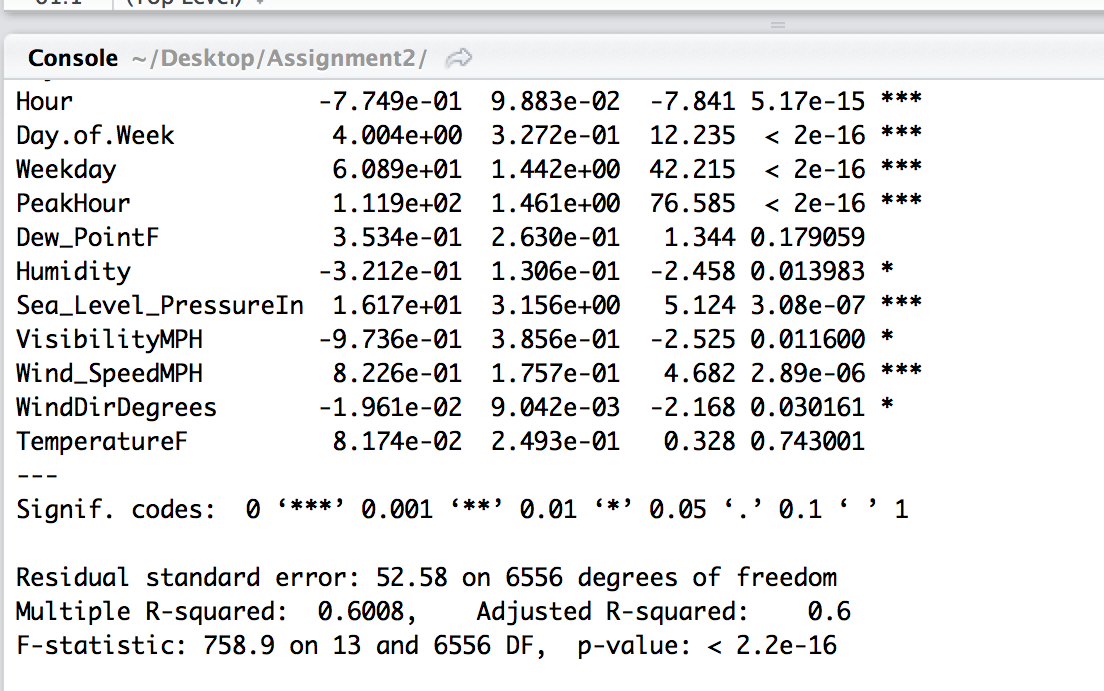
*Now we run Stepwise search to see if we can get better result with stepwise. We have used StepAIC that performs stepwise model selection by AIC. If we give direction = “both”, it will perform both forward and backward search*

**

*Result*

**

*Checking R square value with parameters obtained from stepwise regression.*

******

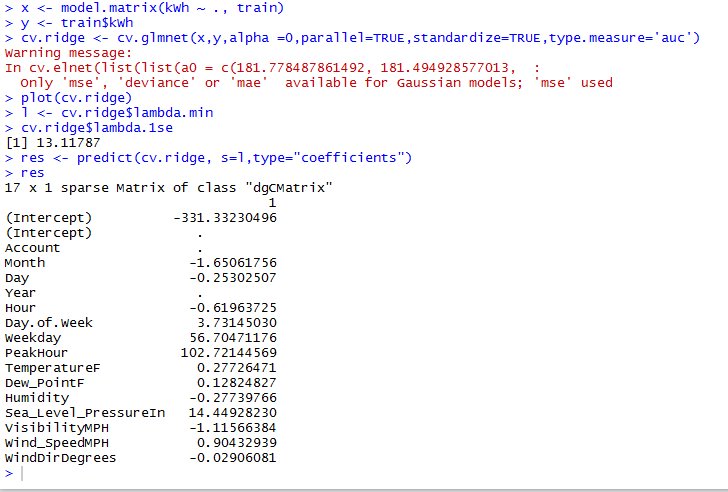
*We could see that stepwise has given better result than forward and backward. So we will move forward with parameters we have obtained from stepwise*

**Ridge:**

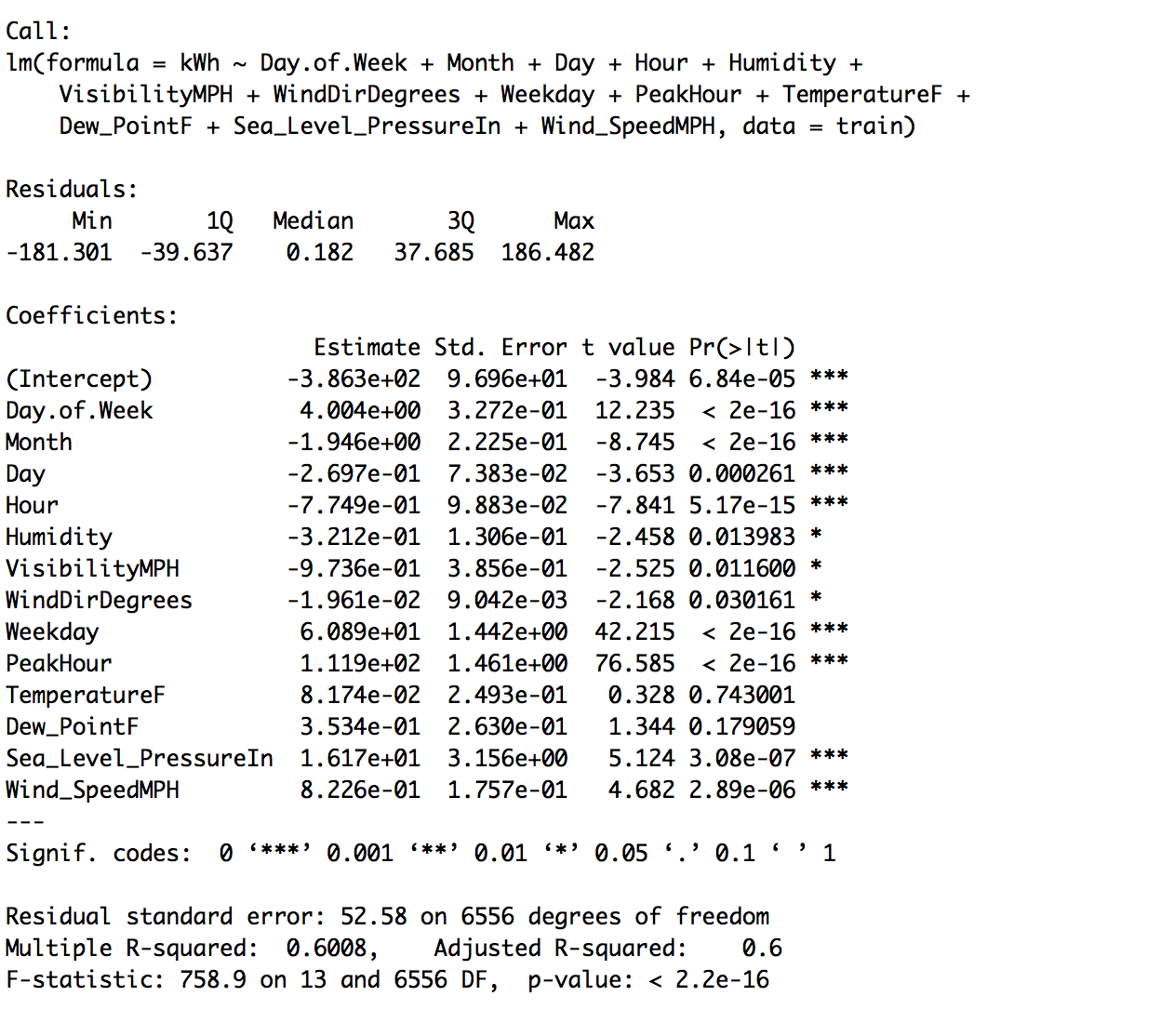
*We have used Ridge and Lasso regression techniques to find if we could get better result in parameter selection.*

*Ridge regression: Ridge regression has cv.glmnet method that input parameters as response variable, predicting parameters, alpha =0 (for ridge) and measure =”auc”(area under ROC curve)*

*We gave parameters to the ridge method and the output has the parameters with coefficient value as shown in below screenshot:*

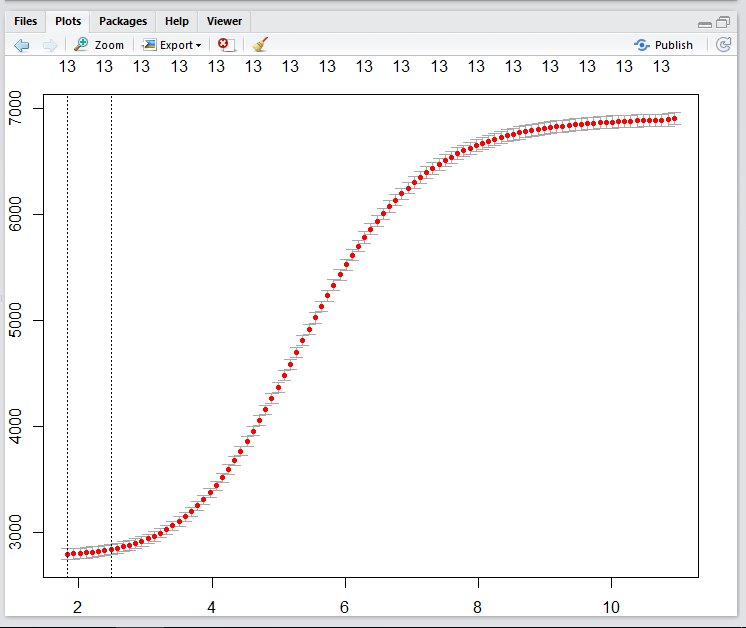
**

*The parameter that has some coefficient value, we chose those parameters to determine the R square value.*

**

*We found that the value of R square was same as stepwise regression.*

**Graph: For the following graphs we have, x-axis: log(lambda), y-axis: Mean Squared Error**

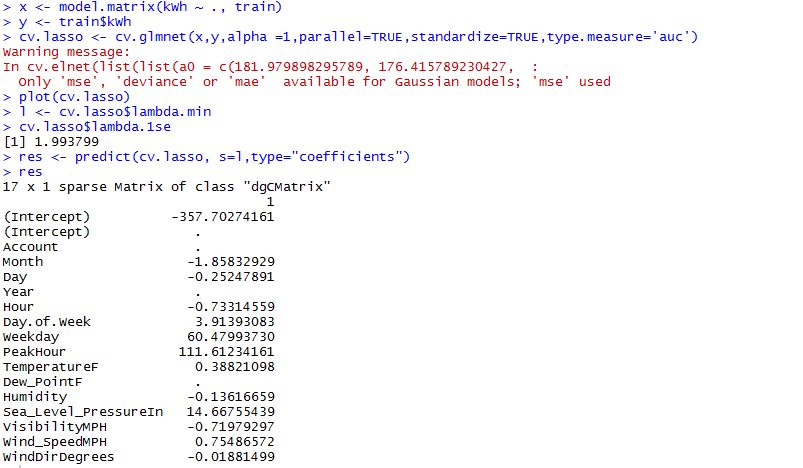
**

**Lasso:**

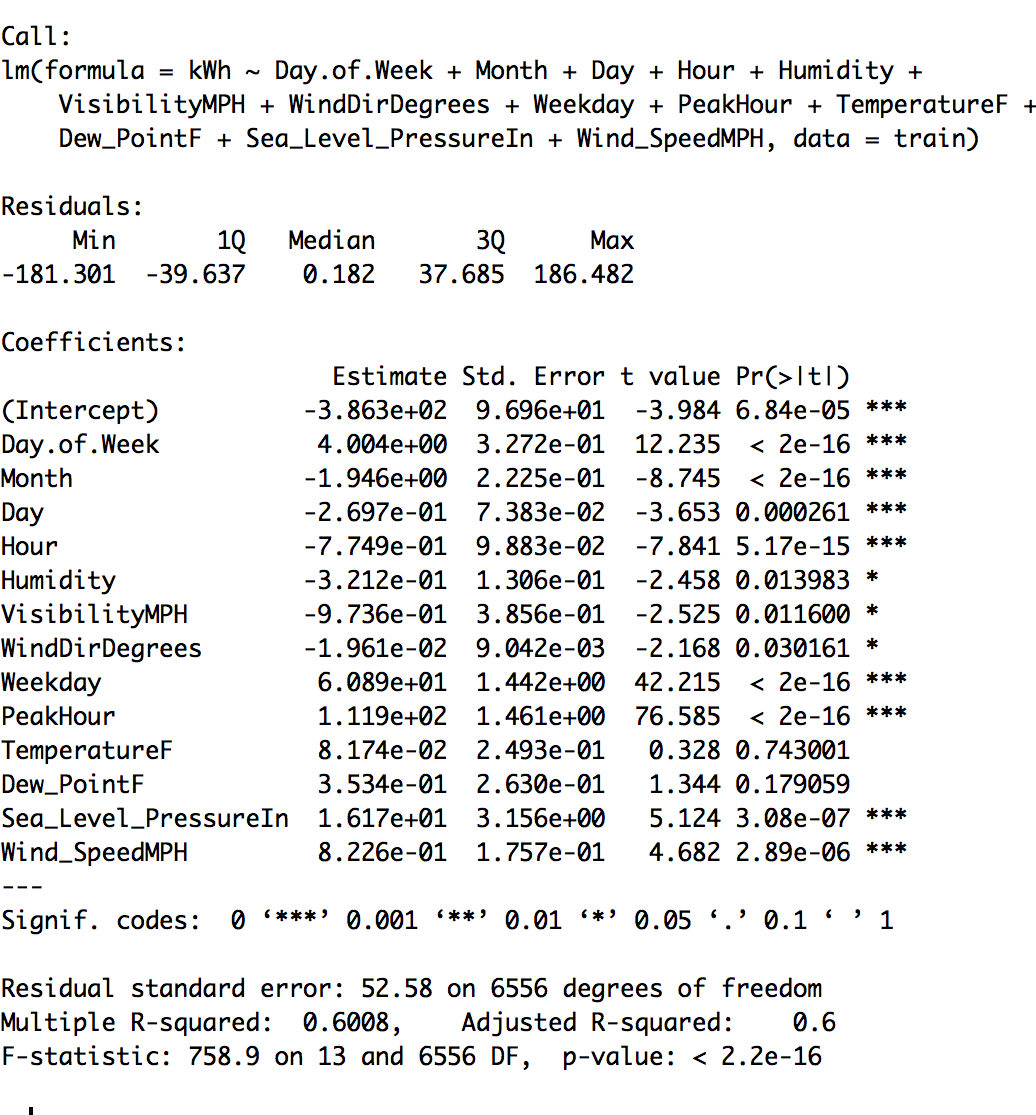
*Lasso is another regression technique.*

*Lasso regression: Lasso regression has cv.glmnet method that input parameters as response variable, predicting parameters, alpha =1 (for lasso) and measure =”auc”(area under ROC curve)*

*Now we run Lasso search*

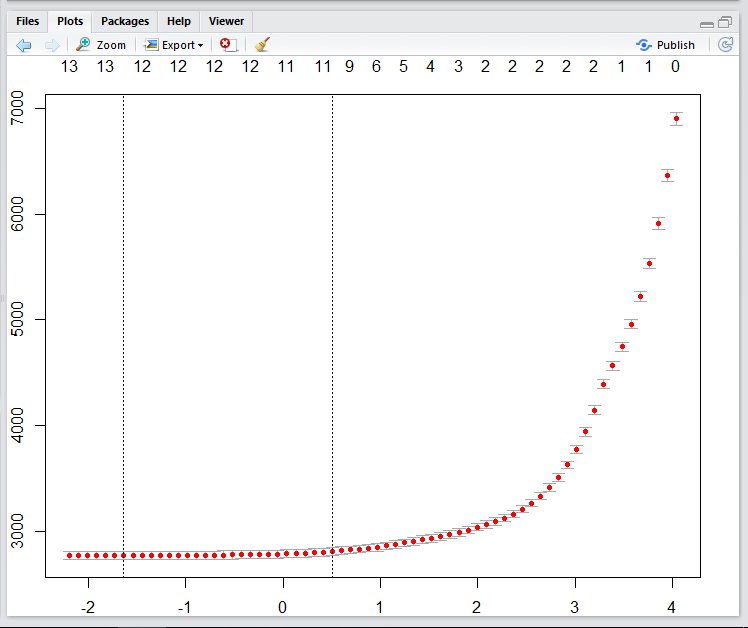
******

* *We took all the parameters that have some coefficient value and tried to find out its R square value*

**

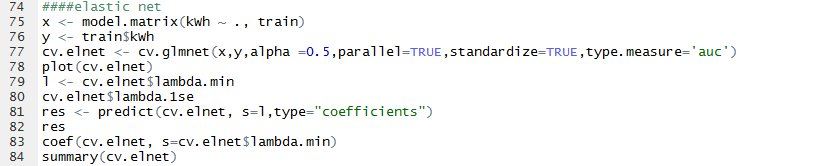
*We conclude that stepwise, ridge and lasso gave us the same R square value with 13 parameters.*

***Graph:* For the following graphs we have, x-axis: log(lambda), y-axis: Mean Squared Error**

******

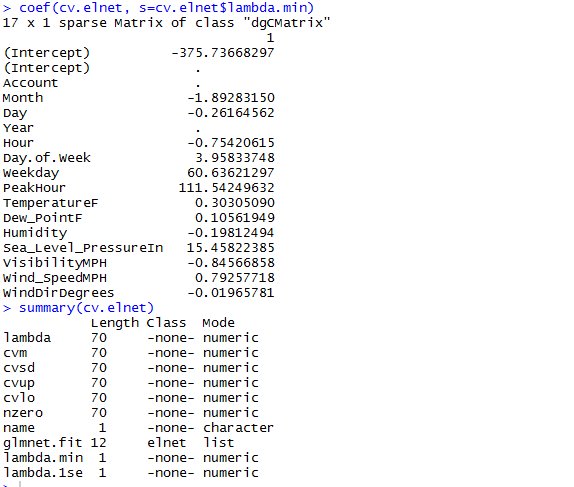
***Elastic Net: Elastic net is another technique of regression. After getting parameters from ridge and lasso, we have tried elastic net.***

*Elastic net regression has cv.glmnet method that input parameters as response variable, predicting parameters, alpha =0.5 (for elastic net) and measure =”auc”(area under ROC curve)*

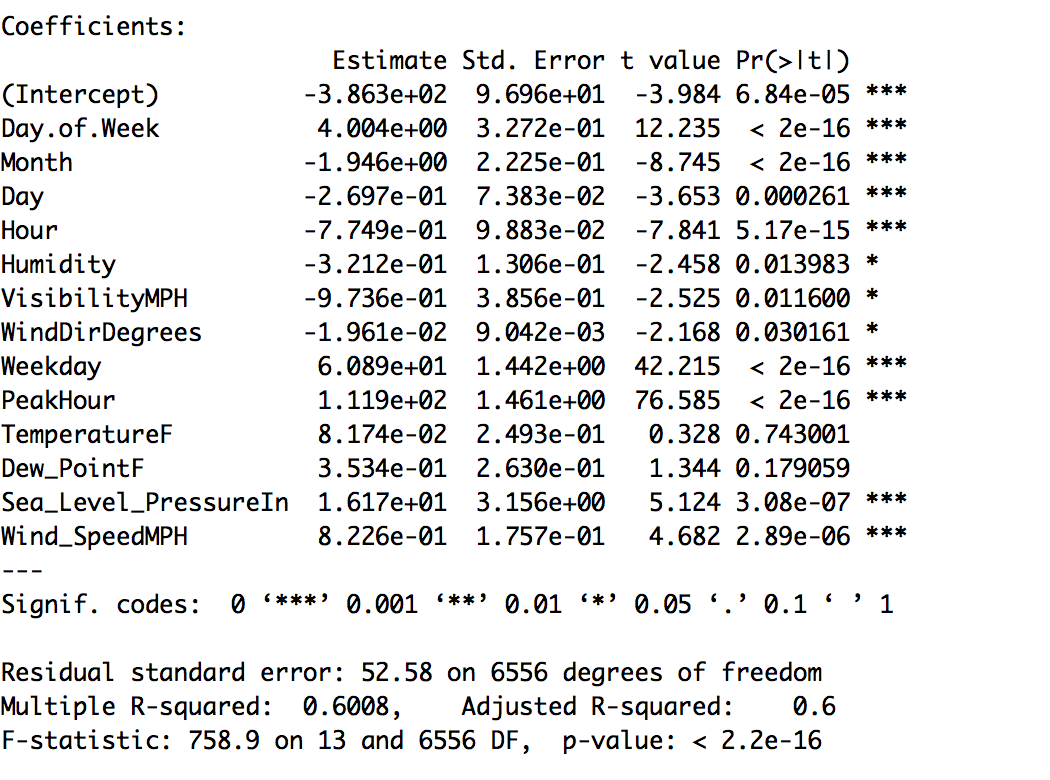
******

***Result:***

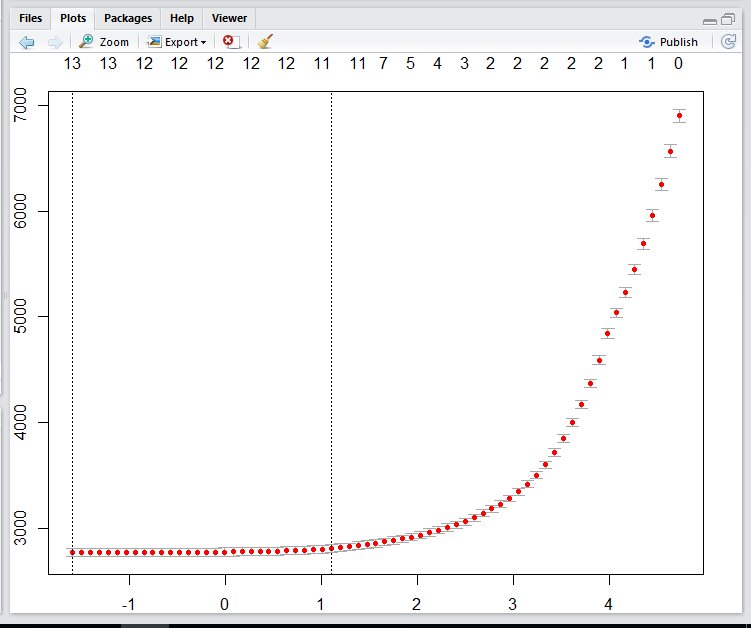
*We took all the parameters that have some coefficient value and tried to find out its R square value*

******

*The R square value from elastic net:*

******

***Graph:* For the following graphs we have, x-axis: log(lambda), y-axis: Mean Squared Error**

******

***Linear Regression:***

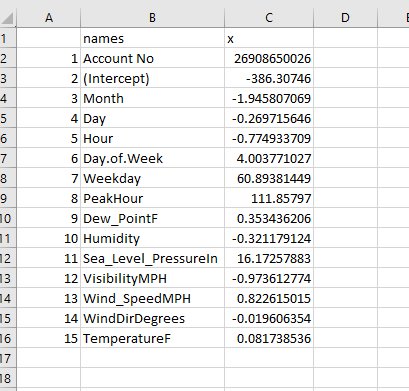
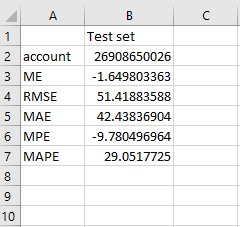
* *Based on the output of Stepwise Regression, we conclude that the following variables have high influence on the power(kwh) – Month, Day, Hour , Day of .Week , Weekday , Peak Hour, Dew\_PointF, TempeartureF, Humidity, Sea\_Level\_PressureIn, VisibilityMPH, Wind\_SpeedMPH ,WindDirDegrees*

***Result****:*

*The following are results we achieved after running the model on rawdata.*

***Adjusted R-squared – 0.6***

*The output of the script will be two files – RegressionOutput.csv and PerformanceMetrics.csv*

*** ***

***RegressionOutput PerformanceMetrics***

*RMS - The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed*

*MAPE - The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of accuracy of a method for constructing fitted time series values in statistics, specifically in trend estimation. It usually expresses accuracy as a percentage*

*MAE - In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes.*

**Part3 – Forecast**

***Script****: assignmen2\_part3.R*

***Packages****: chron, forecast*

***Steps****:*

* *Forecast the power(kWh)using predict function as shown below. The model was built in last part. Using the model and the forecastInput.csv as parameters.*

**

* *The predicted values were written in forecastOutput\_AccountNumber.csv*

