# **Part 1: Data Wrangling and cleansing**

Following is the code and the explanation:



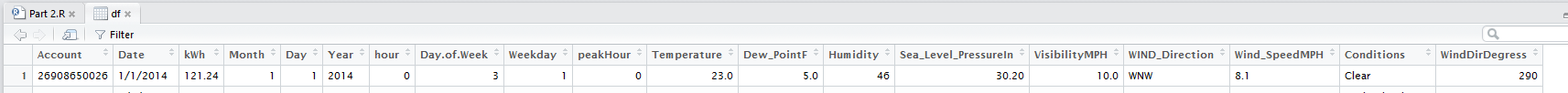
1. Parse through the csv files.
2. Sum up the columns rounding off to the nearest hour.
3. Remove the rows which contains kVARh and the ‘Power Factor’ Units.
4. Remove all data from Channel Field except ‘MILDRED SCHOOL 1’.
5. Add an index to the DataFrame.
6. Create two separate DataFrame
7. One DataFrame consisting of columns Date, Channel and Units.
8. Create another DataFrame consisting of only hourly kWh data
9. Stack the DataFrame which results in the DataFrame consisting of hour and the corresponding kWh values in columnar fashion
10. Now merge the DataFrames by index resulting in the DataFrame which consists of columns as follows:
11. Date, Channel, Units, Hour and kWh values
12. Now Add the other values like month, day, year, hour, Day of Week, Weekday, peakhour these values can be extracted from the Date values.
13. Now by pinging weather underground continuously using the Date as reference we can extract Temperature, Dew\_Point, Humidity, Sea\_Level\_PressureIn, VisibilityMPH, Wind\_Direction, Wind\_SpeedMPH, Conditions, WindDirDegrees
14. These values are then added to the dataframe and the value is then imported into the CSV.

# **Part 2: Multiple Linear Regression**

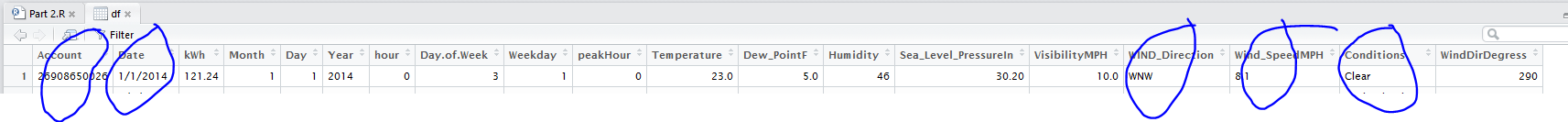
Following is the code and explanations as follows:



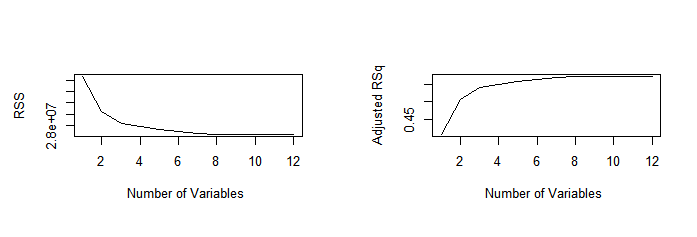
1. Set a working directory.
2. Import the CSV file from the combined data from the rawData1, rawData2 and JSON objects and put it in “sampleformat.csv” file.



1. Remove the columns that do not have numerical values and those that are redundant.

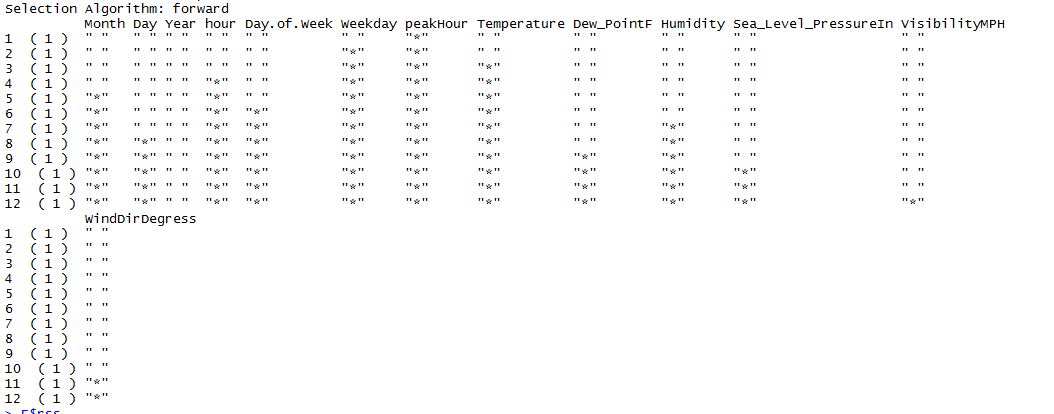


1. Divide the data into train and test as 75%-25% respectively.
2. Searching all subset models up to size number of variables by subsetting it.

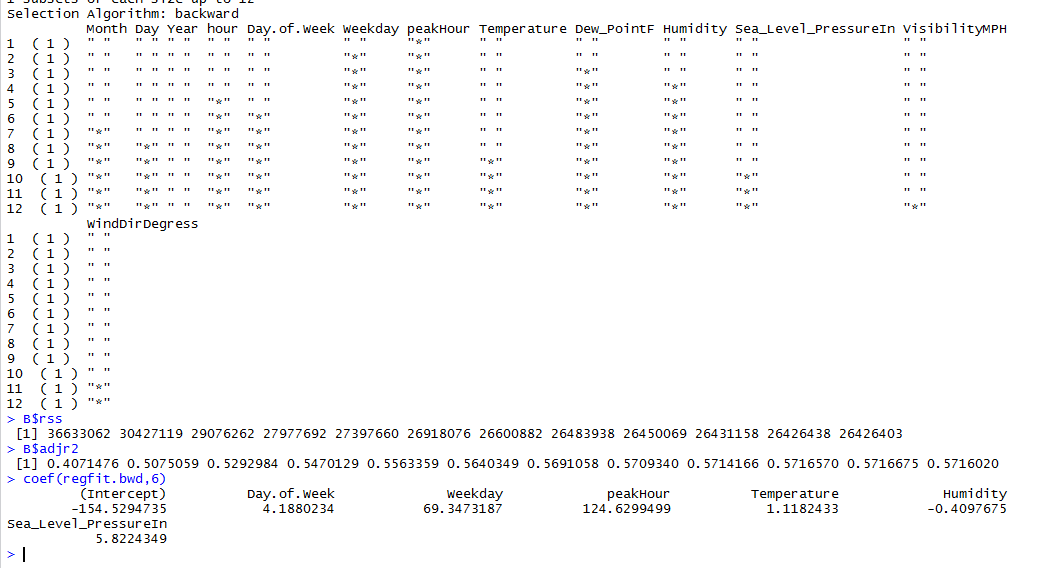


We see the RSS decreases after 7-8 predictors.

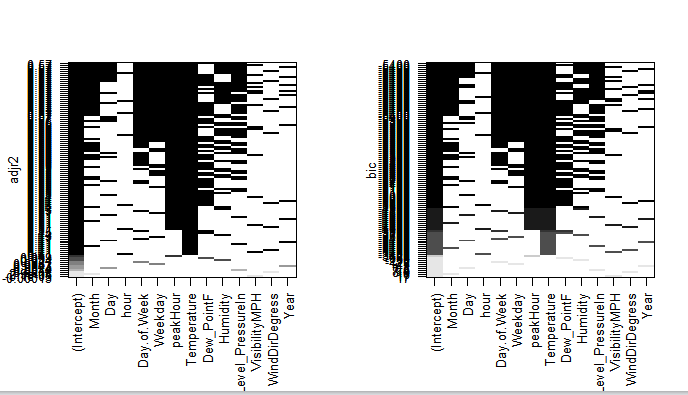
1. On forward and backward selection, we realize the coefficients for the strong predictors.







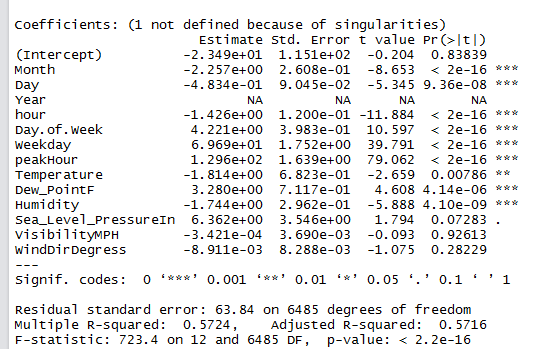
Also by using the exhaustive search we can see the strong predictors:



1. From the above methods we realize that the following predictors are the strongest and help making a good model.

lm.fit=lm(kWh~ ., data=train) #all columns

summary(lm.fit)

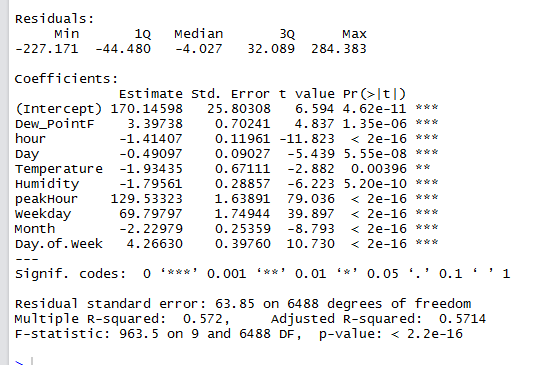


Also when we select all the predictors we see that the R square values are not so great and model doesn’t seem to be the best.

1. We start choosing and removing the different predictors.

lm.fit <- lm(kWh ~ Dew\_PointF + hour + Day + Temperature + Humidity + peakHour + Weekday + Month + Day.of.Week, data = train)

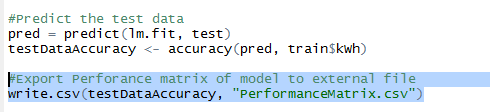
summary(lm.fit)



1. Export the regression outputs to external file “RegressionOutputs.csv”. The contents are as follows.



1. This was using the training data. Now we have created our model and checked its coefficients. We then use our test data to see how accurate our model is. The performance matrix is been exported to a CSV file – “PerformanceMatrix.csv”



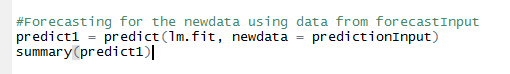


# **Part 3. Forecasting**

Following is the code for scrapping the forecastData.csv file and explanation.



1. We use the model that we created using the train data and testing it against the test data. We read the file the “forecastData”. Cleaning the data and setting it up according to the dataframes we used in the Regression model.
2. We use the library forecast, function **predict()**. To forecast the kWH value using different predictors we get from the above file.



1. Add the column of “kWh” and the “date” in the output dataframe.
2. Extract the required columns as per the sample file given.  
   