ABSENTEESIM

**CONTENT:**

1. **Summary and Problem Statement**
2. **Data Overview**
3. **Feature Engineering** 
   1. **Data Conversion**
   2. **Outlier Analysis**
   3. **Data Imputation**
   4. **Correlation**
   5. **Sampling**
4. **Modelling** 
   1. **Linear model with R**
   2. **Linear model with python**
   3. **Cross Validation(Overfit or Underfit)**
   4. **Random Forest with R**
   5. **Random Forest with Python**
5. **Conclusion**
   1. **Model Evaluation**
   2. **Results Graphs**
   3. **Model Selection**

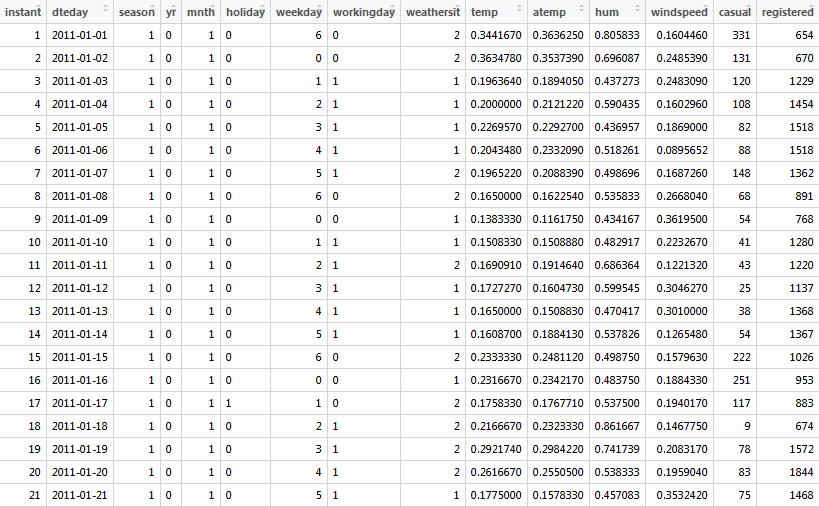
**1. Summary and Problem Statement:**

XYZ is a bike Rental Company. The Company lend the bikes on rent to the customers on daily basis. Now the Company wants to know that how many number of bikes going to be on rent in forthcoming season. The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

**2. Data Overview:**

Our objective is to build the regression model which will be depending upon the various properties of the employ. Given below is sample of the data set that we are using to predict the hours of absenteeism in the company.

Our objective is to build the regression model which will be depending upon the various property of environmental and seasonal conditions. Given below is sample of the data set that we are using to predict the count of bike rent.



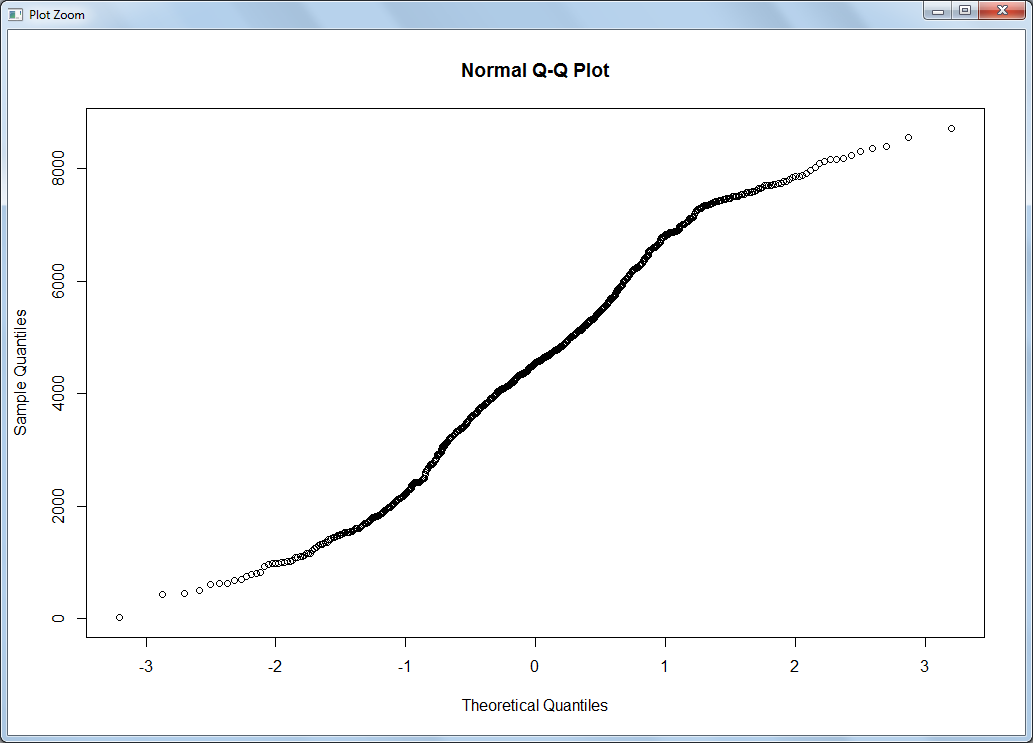
The above tables are the predictor variables which are used to predict the count of bike rent. Below is the sample of dependent variable.

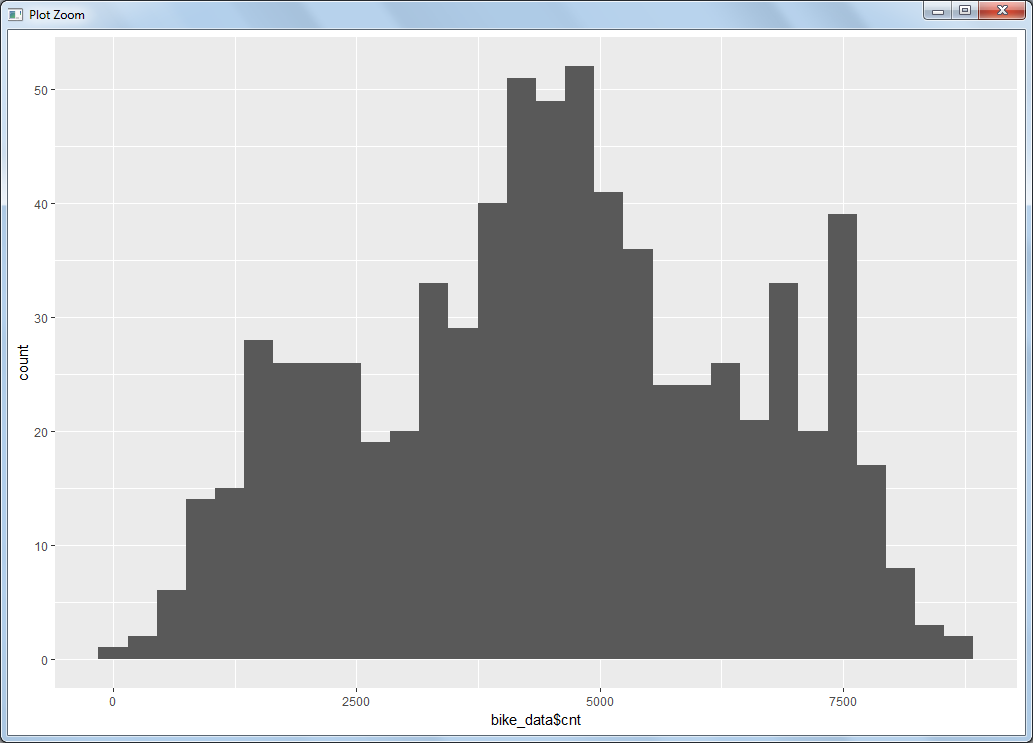
**DEPENDENT VARIABLE**

****

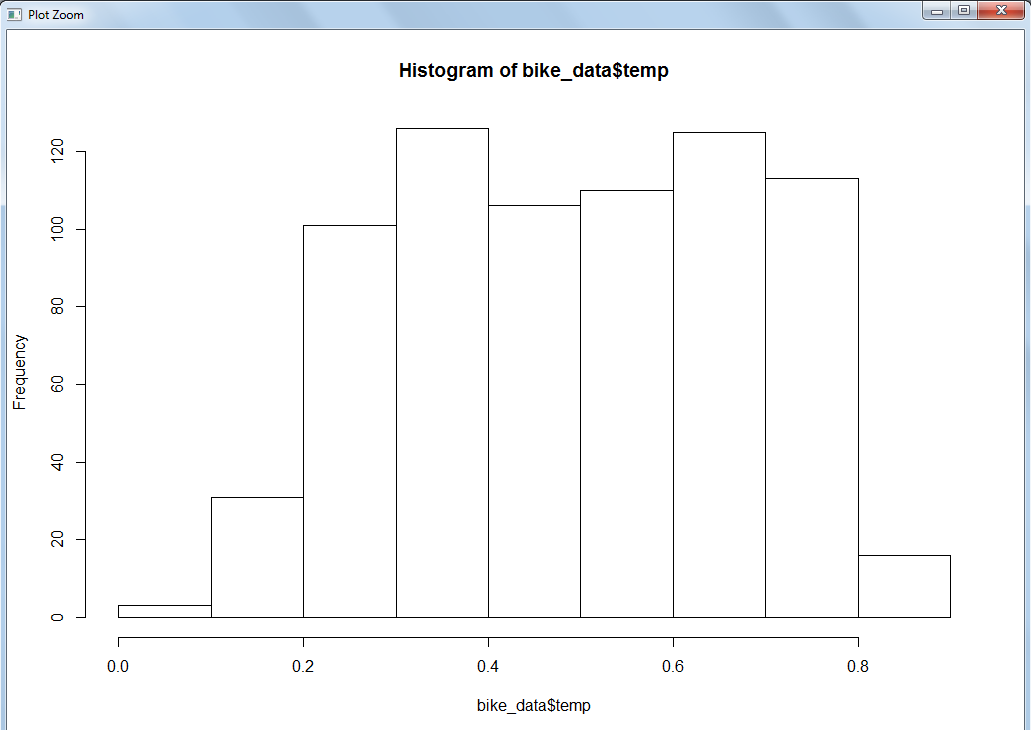
**DISTRIBUTION OF DEPENDENT VARIABLE**

Quantile wise distribution of dependent variable

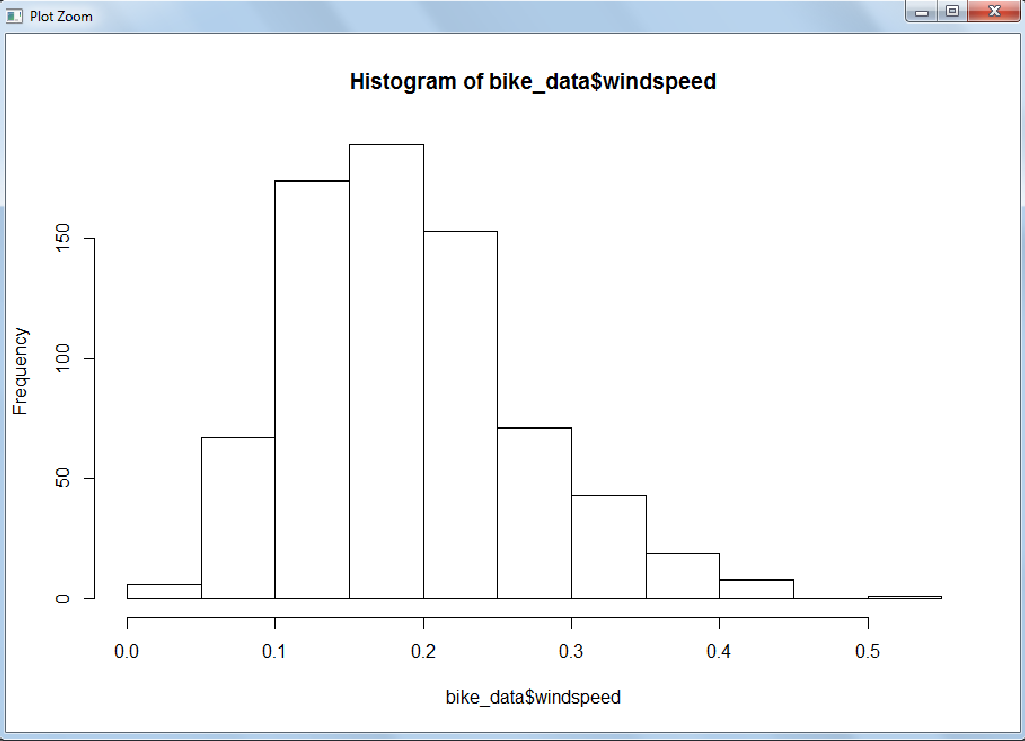




As we have seen the data is not skewed and almost normally distributed.

**Temperature** 

**Windspeed**



**3. Feature Engineering**

**3.1 Data Conversion**

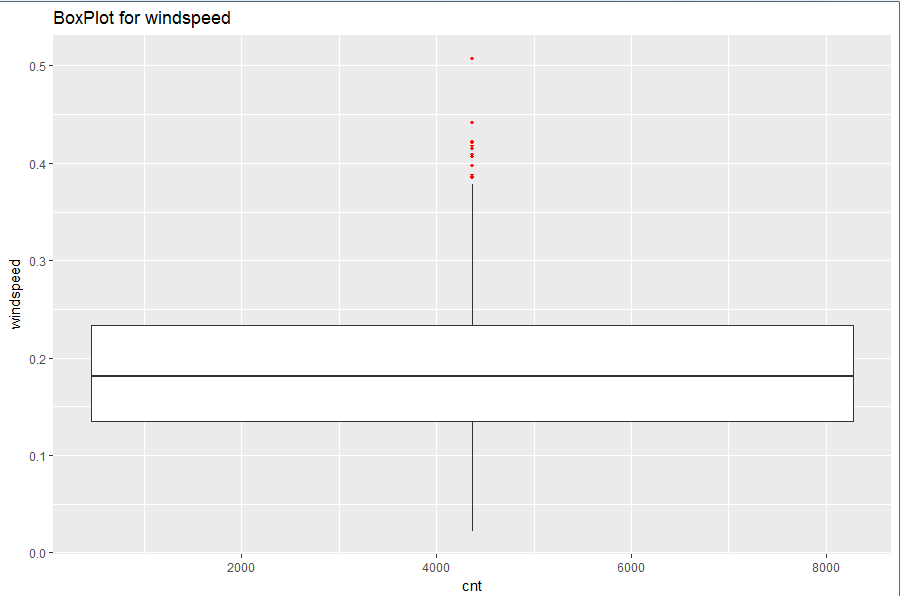
a) Convert the variable into suitable variable category for example year is given as numerical variable but it has to be categorical.

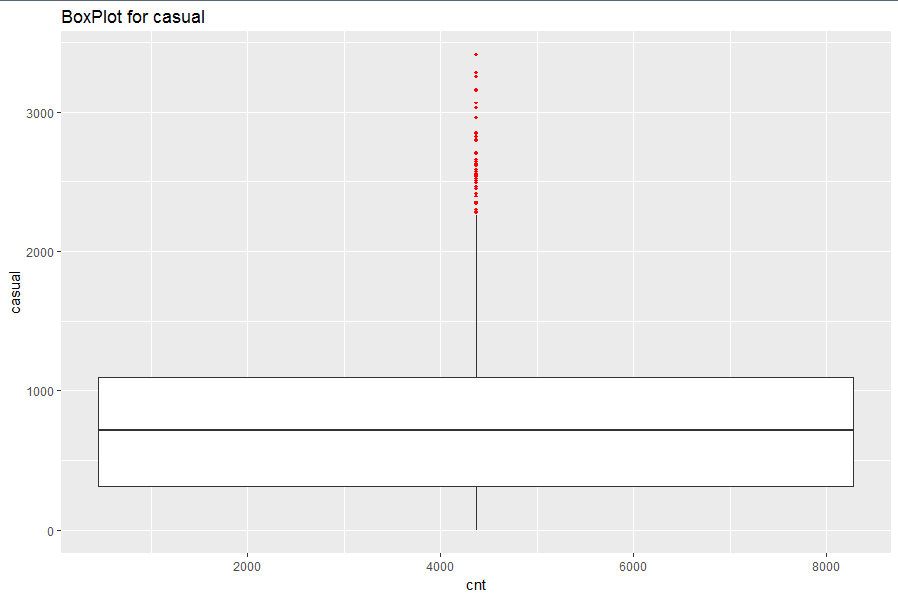
b) Create the dummies for all the categorical values

**3.2 Outlier Analysis**

Outlier is one of the steps of the data pre processing where we have identify the outlier and remove it. Box plot is one of the best ways to find out the outliers in the variable. Below are the outliers for the some of the continuous variable.

**Windspeed**



**Casual**

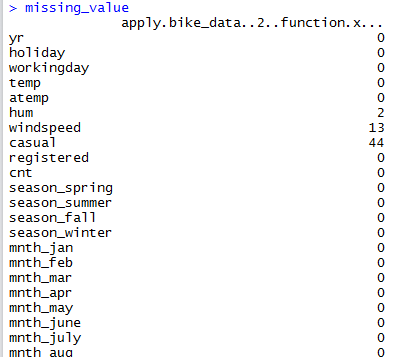
We have seen from the probability distribution of the variable that the data is skewed and now with the help of box plot we confirmed that there is outlier present in the some of the continuous variables. The data which is fall below or above the lower and upper fence is considers as outlier and should be removed as part of data pre processing.

As we don’t have lot of data and the percentage of the outlier data is significantly high we cannot simply delete the data.

We can replace the outlier which NA and later impute as missing value.

**3.3 Data Imputation**

Now we can check the total number of missing value for each of the variables.



From above table we can see that the there are missing value in many of the variables.

Impute the missing value using KNN imputation method.

**3.4 Correlation**

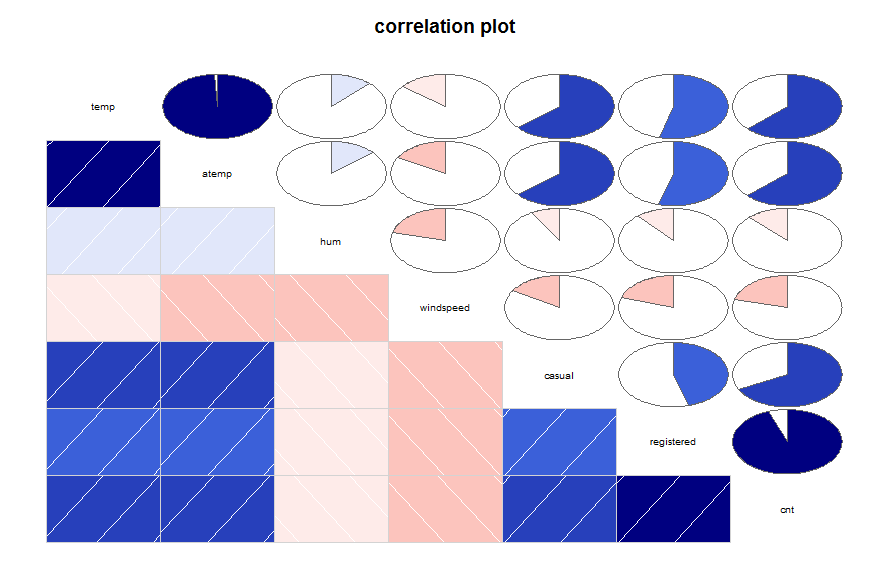
Now we have check for the correlations.

We have to make sure that independent variables should not be correlated, which means two different independent variable should not carry the same information, otherwise it cannot the explain the variance in the dependent variable.

We can drop any one of the such variable from the data set.

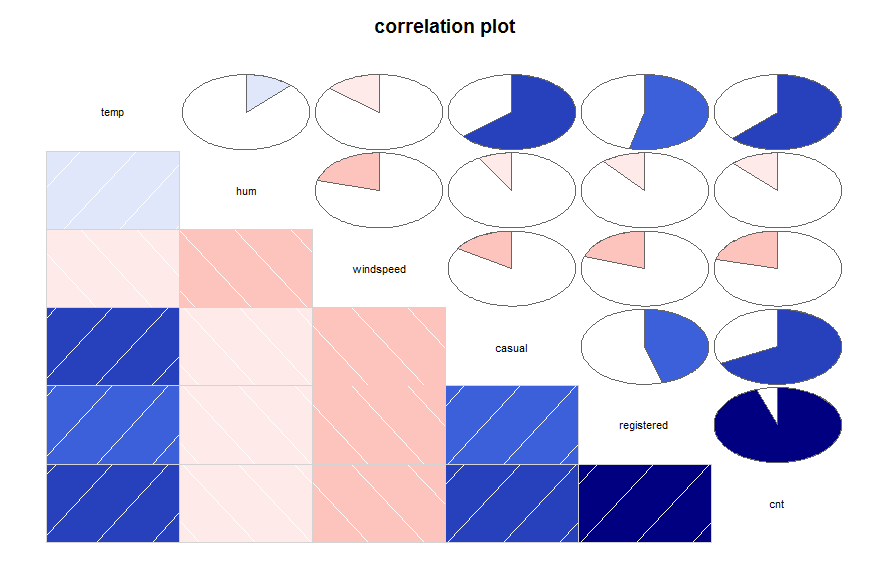
We will use the Scatter plot to visualize the correlation between the independent variables.

**CORRELATION PLOT**



We can see from above plot that temp and atemp is highly positively correlated.

We can drop any one of the variable and registered and cnt are also correlated but it is independent and dependent variable so we can remove this.



Now after dropping the body mass index we can see that there are no other correlated

Variables

**3.5 Sampling**

Sampling is the technique of dividing the data in training and testing data set. Let say we divide out data in ratio of 70:30 for training and testing data set

Training Data Set: Training data set is that data upon which we create out model.

Testing Data Set: Testing data set is that data set upon which we validate our model by means of accuracy matrix.

Here, we have used simple random sampling technique.

**4 Modelling:**

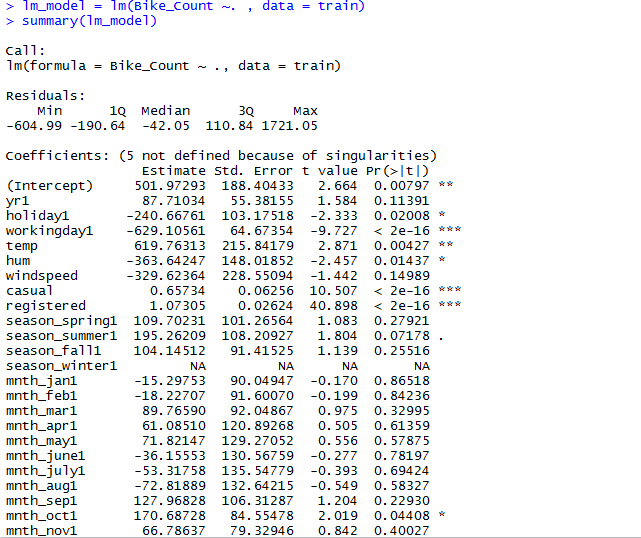
Now we are done with data pre processing. The dependent variable in our data set is a continuous variable therefore we have to build the regression model for that.

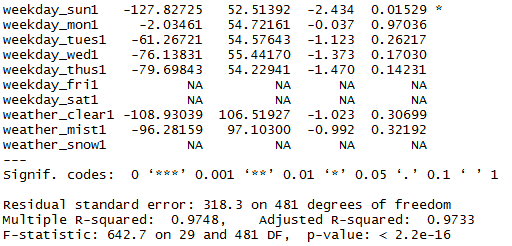
**1. Linear Model with R**

We will create the model by keeping all the independent variables into the model.

lm\_model = lm(Bike\_Count ~. , data = train)

summary(lm\_model)





We can see that there are lots of variables which are not significant and will lead to the errors in the predictions. Star marked variables are most significant variables.

We will create another model with all the significant variables.

lm\_model\_improve = lm(Bike\_Count ~ workingday+temp+casual+registered

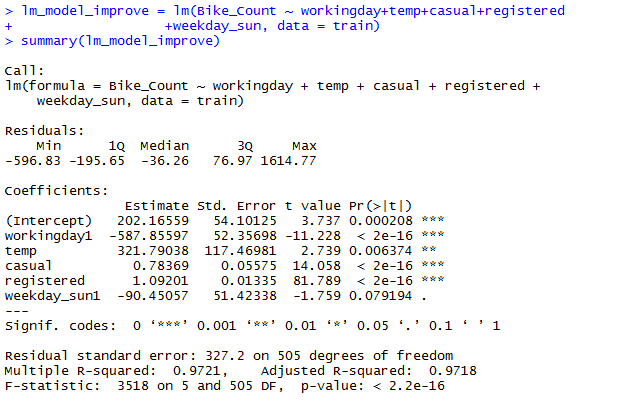
+weekday\_sun, data = train)

summary(lm\_model\_improve)

#Make predictions

prediction\_lm\_improve = predict(lm\_model\_improve,test[,-1])

MAPE(test[,1],prediction\_lm\_improve)



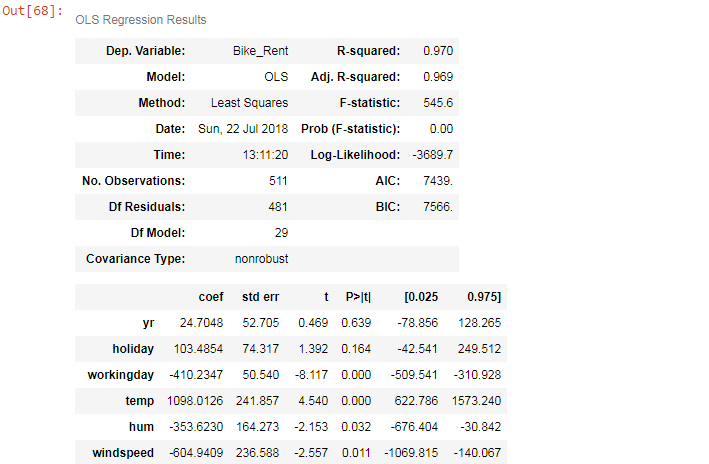
Looking at the significance values of some of the predictor we can see that there is some scope of improvement in this model.

**2. Linear Model with Python**

We will create the model by keeping all the independent variables into the model.

#Build the linear regression model

lm\_model = sm.OLS(train.iloc[:,0],train.iloc[:,1:35]).fit()



Now we will select the significant variable and build the model.

**4.3 Cross Validation(Overfit or Underfit)**

NOTE: I have followed the same procedure for R and Python thats why not writing cross validation for separately but performed seprately

After building the model we have to check for the overfiiting and underfitting of the model

predict result with training dataset independent variable and then calculate error between training dataset actual value and predicted value.

do same thing with test dataset.

now if training dataset has very less error and test dataset has very high error , it means your model has **overfit** the data

if training dataset and test dataset error both are high than it is case of **bias/underfit**.

both error should be less and approx same(training error would be less than test error)

In our case we have found that both the training is less and testing data has high error rate. Check the below code:

#Make predictions with train data

lm\_model\_cv = lm(Bike\_Count ~ workingday+temp+casual+registered

+weekday\_sun , data = train)

#Make predictions with train data

prediction\_lm\_cv = predict(lm\_model\_cv,train[,-1])

#Calculate MAPE for train data set

MAPE(train[,1],prediction\_lm\_cv)

#ERROR on Training data : 7%

#Make predictions with test data

prediction\_lm\_cv\_test = predict(lm\_model\_cv,test[,-1])

#Calculate MAPE for test data set

MAPE(test[,1],prediction\_lm\_cv\_test)

#ERROR on Testing data : 8%

So, we decided to split the data less into ratio(60:40)

**4.4 Random Forest with R**

Random forest is tree based algorithm. For building the model with random forest we have taken the split of the data after cross validation.

First we have tune the mtry and found out the that mtry 22 is best fitted for the model and then create the model using RandomForest function

rf\_improve = randomForest(Bike\_Count ~., data = train,

ntree = 250,

mtry = 22,

importance = TRUE,

proximity = TRUE)

**4.5 Random Forest with Python**

For building the model with random forest in python also we have taken the split of the data after cross validation and build the model on top of that only.

rf\_model = RandomForestRegressor(n\_estimators = 100,oob\_score = True ,random\_state = 0,max\_features = 'auto')

rf\_model.fit(train\_rf.iloc[:,1:35],train\_rf.iloc[:,0])

**5. Conclusion**

**5.1 Model Evaluation**

MAPE and MSE are one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models and generate the error and accuracy.

**Error metric before cross validation**

**Linear Regression R:**

**MAPE :**

prediction\_lm\_improve = predict(lm\_model\_improve,test[,-1])

MAPE(test[,1],prediction\_lm\_improve)

0.077062

Error :8 %

Accuracy : 92%

**Random Forest R:**

(Before and after will be same because already taken the train and test data after cross validation)

rf\_improve\_predict = predict(rf\_improve,test[,-1])

MAPE(test[,1],rf\_improve\_predict)

0.031951

Error : 3%

Accuracy : 97%

**Linear Regression Python:**

lm\_prediction\_improve = lm\_model\_improve.predict(test\_improve)

**MAPE**

MAPE(test.iloc[:,0],lm\_prediction\_improve)

7.461627001561393

Error : 8%

Accuracy : 92%

**Random Forest Python:**

(Before and after will be same because already taken the train and test data after cross validation)

Prediction = rf\_model.predict(test\_rf.iloc[:,1:35])

**MAPE:**

MAPE(test\_rf.iloc[:,0],Prediction)

13.180241064646983

ERROR : 13%

Accuracy: 87%

**Error metric after cross validation**

**Linear Regression In R**

prediction\_lm\_new = predict(lm\_model\_cv\_train,test\_cv)

MAPE

MAPE(test\_cv[,1],prediction\_lm\_new)

0.0342943

MAPE ERROR : 3.4%

ACCURACY : 96.6%

**Linear Regression In Python**

**MAPE:**

lm\_prediction\_train\_cv = lm\_model\_train\_cv.predict(test\_improve\_cv)

MAPE(test\_improve\_cv.iloc[:,0],lm\_prediction\_train\_cv)

1.6826672723477214e-25

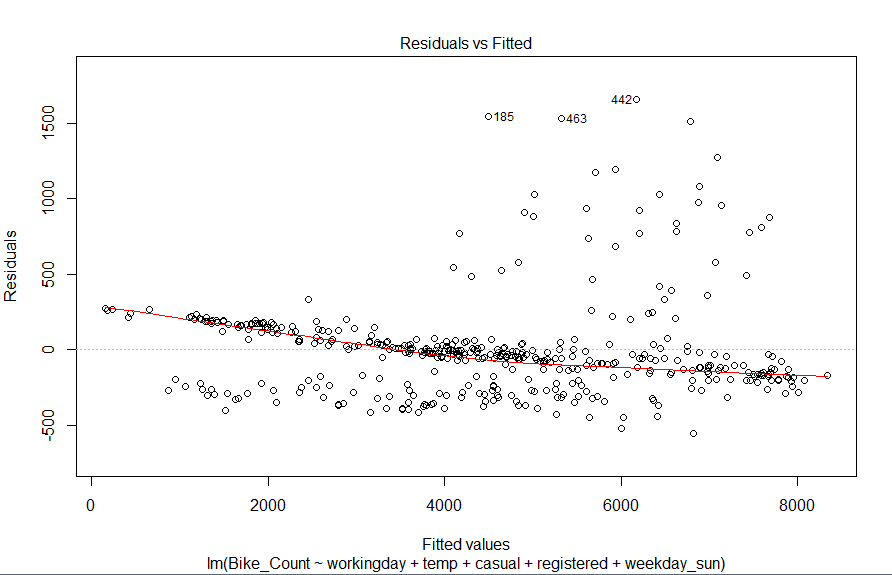
ERROR : 1.6826672723477214e-25

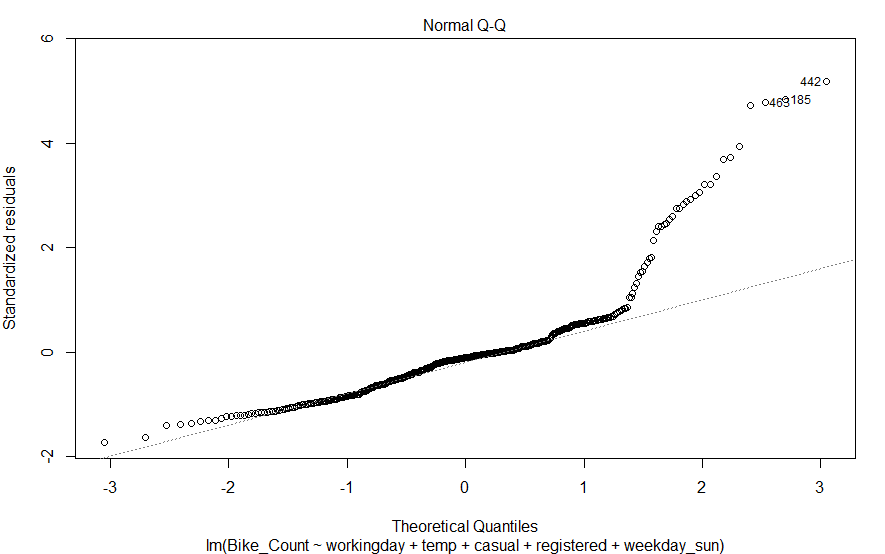
ACCURACY : 99.999999%

**5.2 Results Graphs**

GRAPH FOR LINEAR REGRESSION

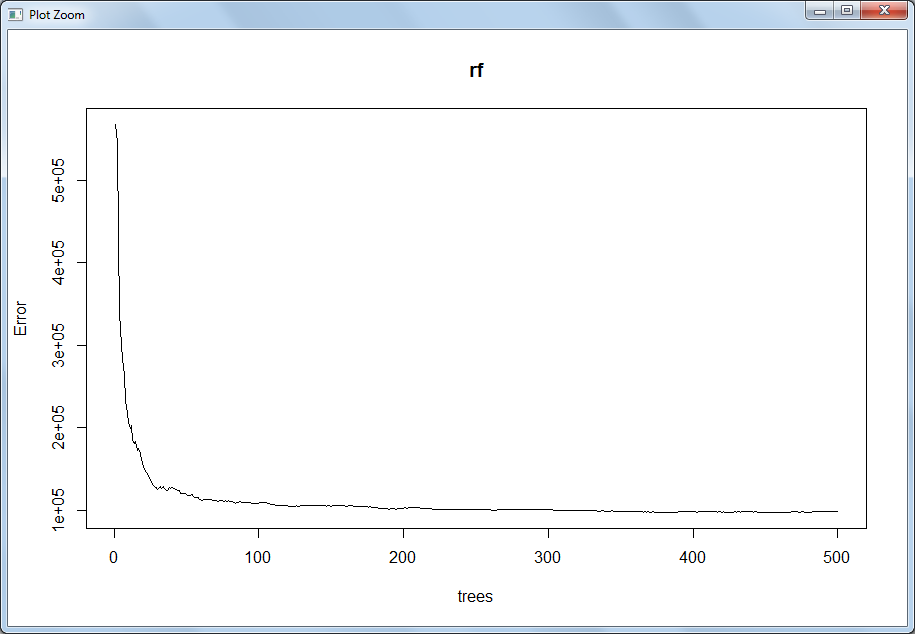
From below two graphs you can see that there is very less hetroskedasticity and error are normally distributed.





GRAPH FOR RANDOM FOREST

From the below graph we can see that as the number of trees increase the error became less.



**5.3 Model Selection**

We can see that the In R Random forest is working fine comparatively linear regression and in python linear regression works fine comparatively random forest.

1) Company should focus on what to increase the business?

As we have seen from the linear regression and random forest there are certain factor which company has keep in mind before renting the bike from going onwards.

a) People usually rent bike on weekends or holidays or when season is good. Company should focus on the weather forecast and give attractive price on above mention calendar time to get the maximum profit.

I have create a new data with Predicted\_dataset that we can get to know how much bikes are going to be on rent in coming year.