EDWISOR

Project: Prediction of bike rental count on daily based on the environmental and seasonal settings.

Submitted by: Anurag Pratap Singh Chauhan
Date: 18/04/2019

Contents

1 Introduction 3
1.1 Problem Statement 3
1.2 Data 4
2 Methodology 5
2.1 Prepocessing 5
2.1.1 Univariate Analysis 5
2.1.2 Bivariate Selection 7

2.2 Preprocessing Techniques 8
2.2.1 Missing value analysis 8
2.2.2 Outlier analysis 9
2.2.3 Feature Selection 11
2.2.4 Feature Scaling 12
3 Modeling 13
3.1 Model Selection 13
3.1.1 Evaluating Regression Model 13

3.2 Decision Tree 14
3.3 Random Forest 16
3.2 Linear Regression 17

Appendix A - Extra Figures 20
Appendix B - R Code 24
Univariate Analysis 25
Bivariate Analysis 26
Outlier Analysis 27
Feature Selection 27

References 28

1. INTRODUCTION

1.1Problem Statement

Our file “day.csv” contains the daily count of the bike rentals along with the seasonal and weather information between the year 2011 and 2012. Our aim is to predict the count of the bike rentals to automate the system so that we can create a suitable model for future predictions which can be used for various business and research projects.

1.2 Data

Our task is to build Regression model which will give the daily count of rental bikes based on weather and season Given below is a sample of the data set that we are using to predict the count. Before doing that we will change the variable names so that we can avoid the confusion and the data looks more presentable.

Record Index	Date	Season	Year	Month	Holiday	Weekday	Working Day	WeatherSituation
1	1/1/2011	1	0	1	0	6	0	2
2	1/2/2011	1	0	1	0	0	0	2
3	1/3/2011	1	0	1	0	1	1	1
4	1/4/2011	1	0	1	0	2	1	1
5	1/5/2011	1	0	1	0	3	1	1

Table 1.1 : Bike Rental Sample Data (Columns: 1-9)

Temperature	Atemperature	Humidity	Windspeed	Casual Users	Registered Users	Count
0.344167	0.363625	0.805833	0.160446	331	654	985
0.363478	0.353739	0.696087	0.248539	131	670	801
0.196364	0.189405	0.437273	0.248309	120	1229	1349
0.2	0.212122	0.590435	0.160296	108	1454	1562
0.226957	0.22927	0.436957	0.1869	82	1518	1600

Table 1.2 : Bike Rental Sample Data (Columns: 10-16)

Below are the variables we used to predict the count of bike rentals as per our modified data:

S.No	**Variables**
1	Date
2	Season
3	Year
4	Month
5	Holiday
6	Weekday
7	Working Day
8	WeatherSituation
9	Temperature
10	Atemperature
11	Humidity
12	Windspeed
13	Casual Users
14	Registered Users

Keeping these variables in mind we are going to develop our model using various pre-processing techniques, predictive analysis and model evaluation to find the relationship between these variables and the target variable which is monthly rental bike counts.

2.METHODOLOGY

2.1 Pre-processing

Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as a preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user.
It refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. To start this process we will first try and look at all the distributions of the Numeric variables. Most analysis like regression, require the data to be normally distributed.

2.1.1 Univariate Analysis

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and it’s major purpose is to find out whether the data is normally distributed or not because most analysis like regression, require the data to be normally distributed.

In the following figures we have plotted the probability density functions numeric variables present in the data including target variable Count.

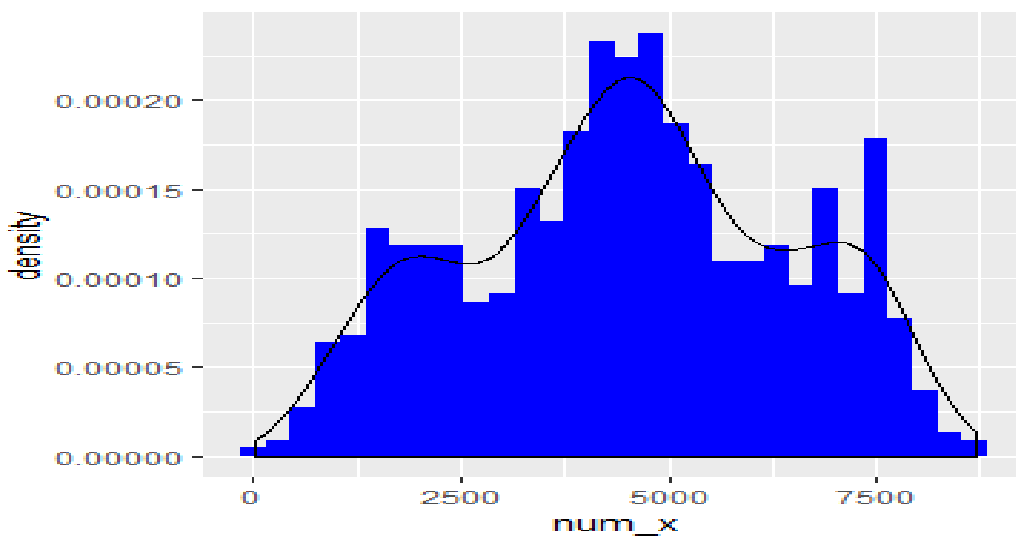

FIG 2.1: Distribution of target variable (COUNT) (R code in Appendix B)

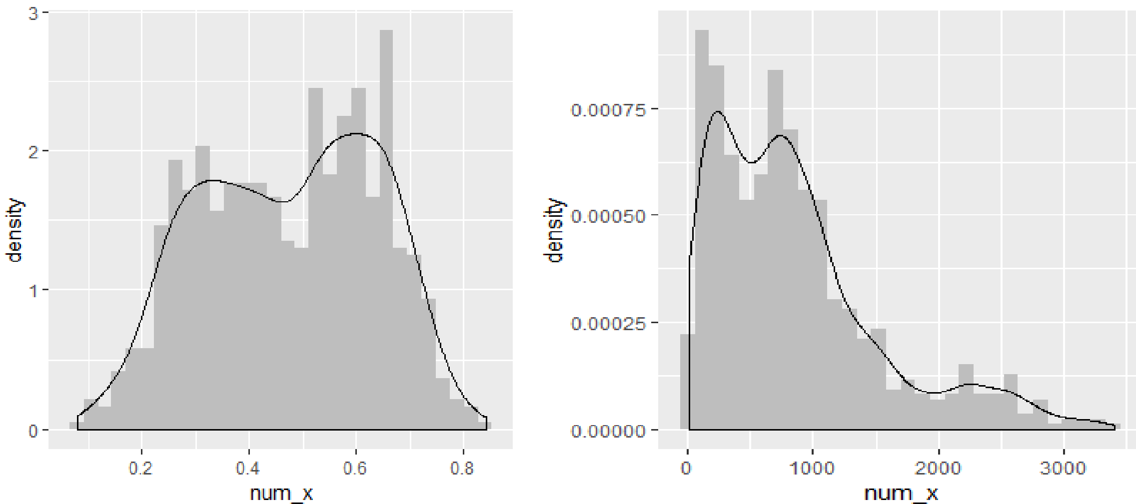

FIG 2.3: Distribution of variable Atemperature and Casual Users

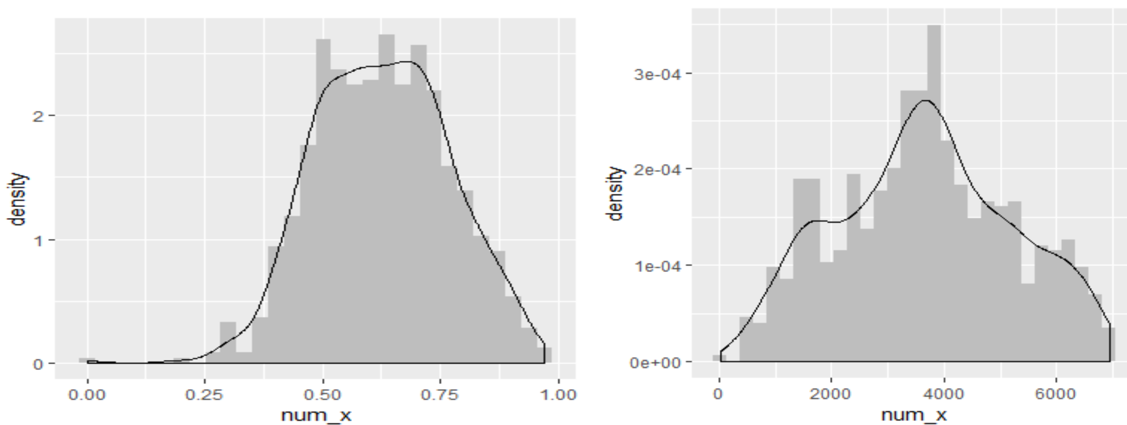

FIG 2.4: Distribution of variable Humidity and Registered Users

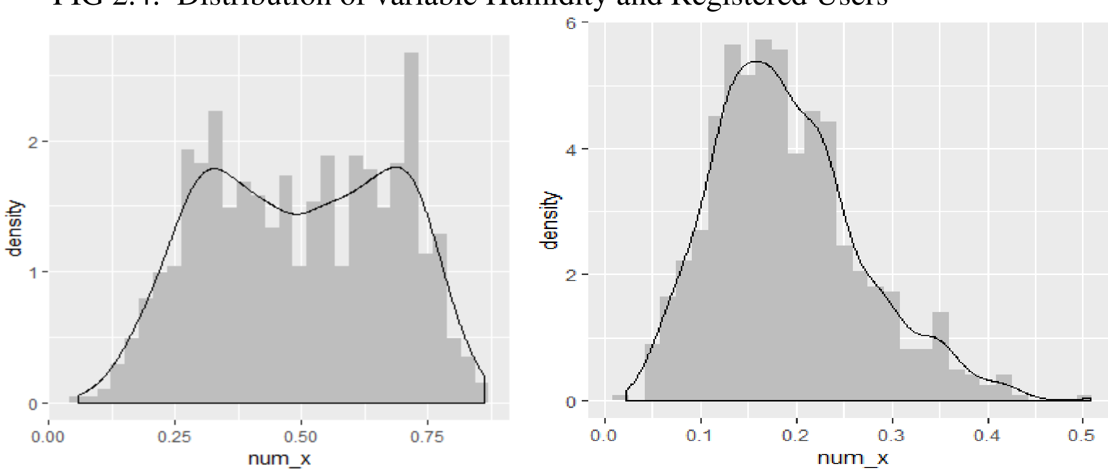

FIG 2.5: Distribution of variable Temperature and Windspeed

From these plots following inferences can be made.

i. Target variable “Count” is normally distributed
ii. Independent variables like ‘Temperature”, “Atemperature”, and ‘Registered Users’ data is distributed normally.
iii. Independent variable ‘Casual Users’ data is slightly skewed to the right so, there are chances of getting outliers.
iv. Other independent variable ‘Humidity’ data is slightly skewed to left; here data is already in normalized form so outliers are discarded. (The values are divided to 100 (max))
v. Windspeed is also skewed a bit but it is also normalized.

2.1.2 Bivariate Analysis

Bivariate analysis is the simultaneous analysis of two variables (attributes). It explores the concept of relationship between two variables, whether there exists an association and the strength of this association, or whether there are differences between two variables and the significance of these difference.

Following figures shows correlation between some of the variables also we have plotted a comprehensive plot of the variables with the target variable.

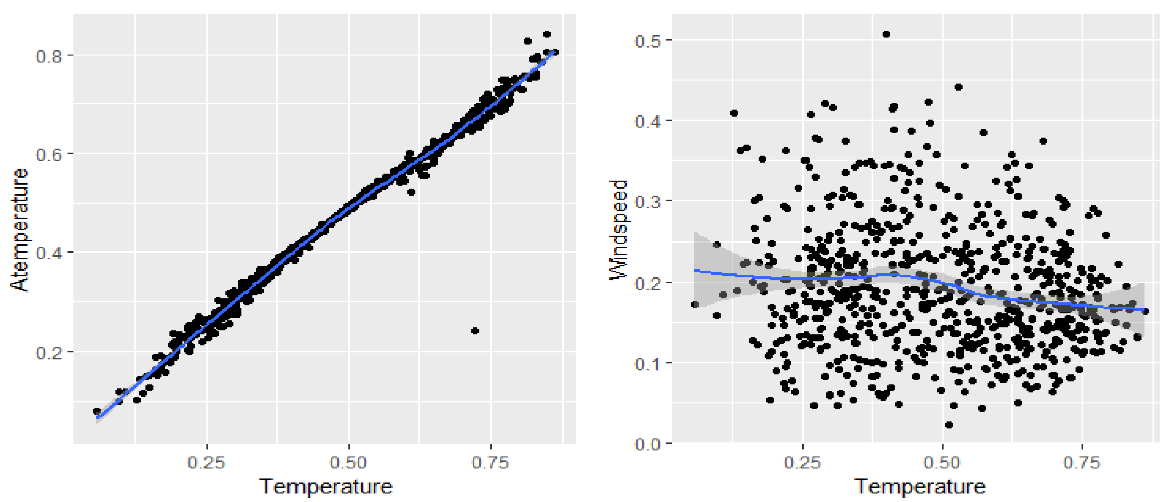

FIG 2.6 Relationship between different variables Temperature and Atemperature (left) and Windspeed and temperature (Right).

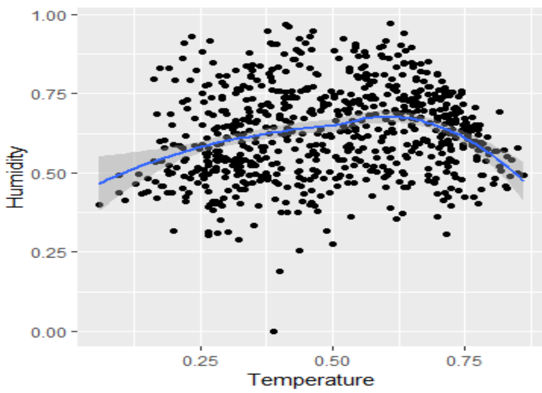

FIG 2.7 : Relation between Humidity and Temperature (left)

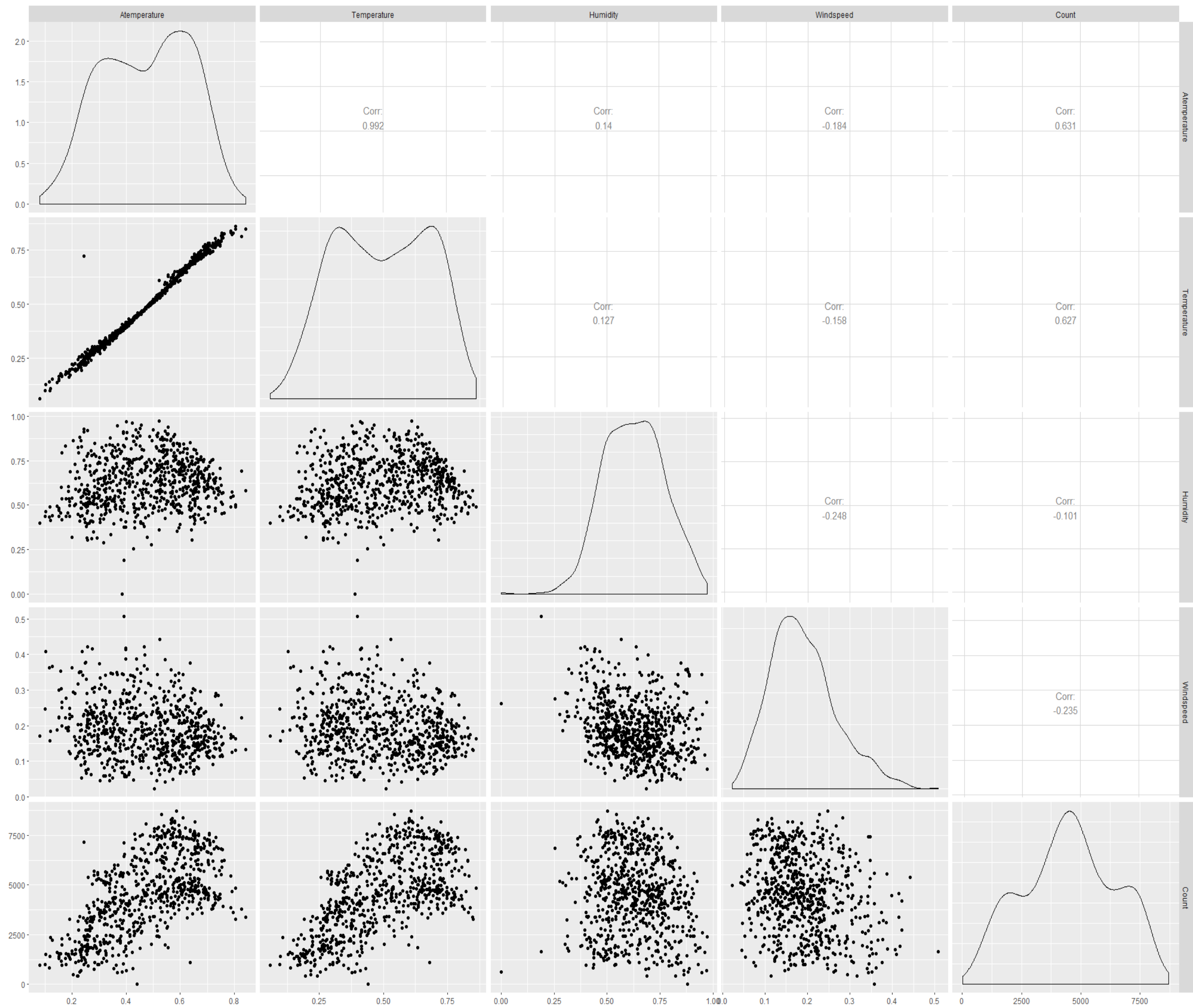

FIG 2.8 Relationship between various Numeric Variables

Various key points to note from here are as follows:

i. This graph shows a very strong correlation between "Temperature and Atemperature".

ii. It shows that very less negative correlation between Temperature and Windspeed.

iii. This plot shows that there is less positive correlation between Count-Humidity .

iv. Also it shows very less negative correlation between Windspeed and Count.

2.2.1 Missing Value Analysis

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models.

Below table illustrates that no missing value is present in the data.

	Columns	Missing_percentage
1	Record index	0
2	Date	0
3	Season	0
4	Year	0
5	Month	0
6	Holiday	0
7	Weekday	0
8	Working Day	0
9	WeatherSituation	0
10	Temperature	0
11	Atemperature	0
12	Humidity	0
13	Windspeed	0
14	CasualUsers	0
15	RegisteredUsers	0
16	Count	0

FIG 2.9: Missing Value analysis.

2.2.2 Outlier Analysis

In statistics, an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses. In following figures we have plotted the boxplots of the various independent variables with respect to dependent variable which is Count. A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set.

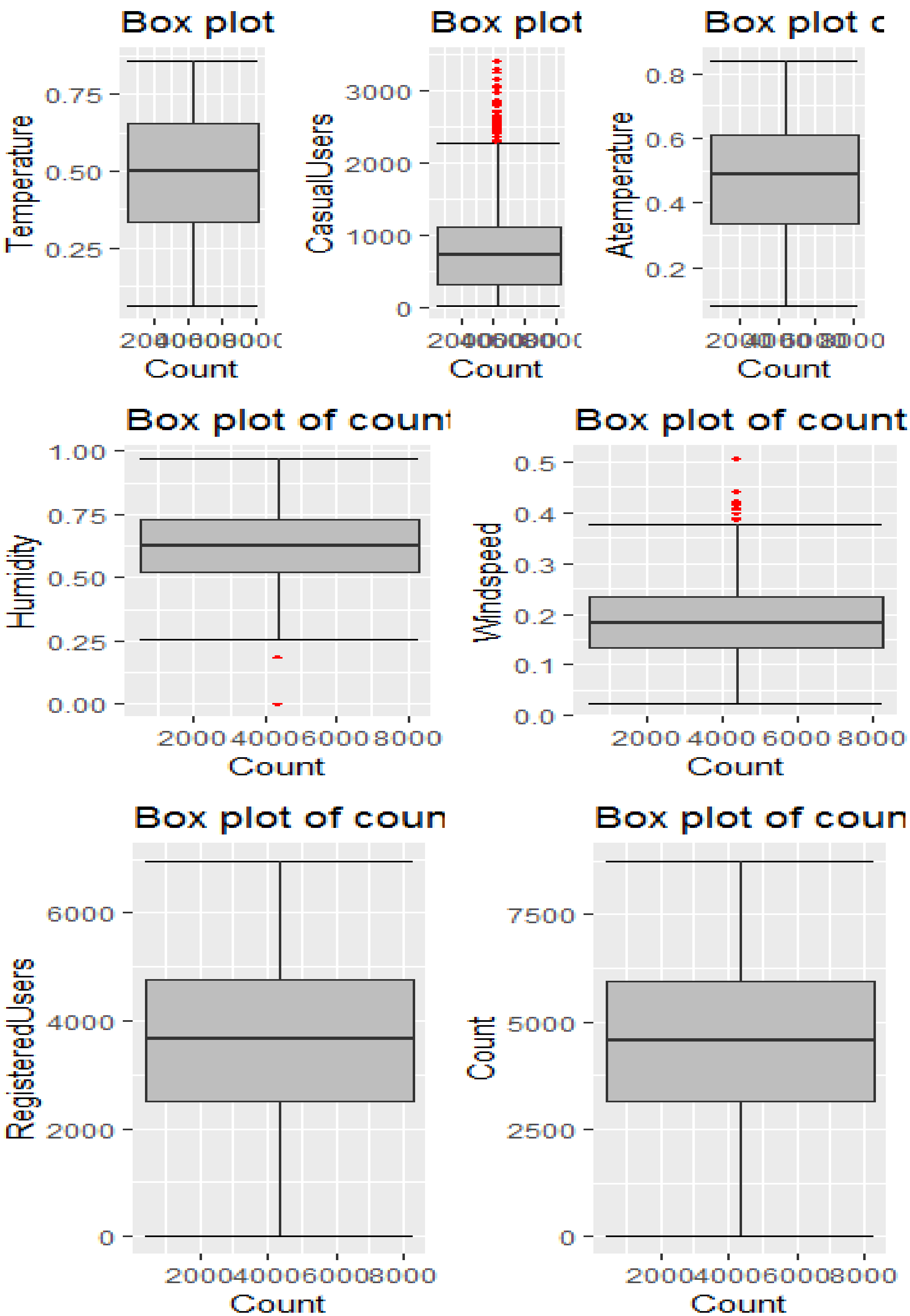

FIG 2.10 Box plot of Various independent variables with variable dependent that is Count after removing the outliers.

2.2.3 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing it.
Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.
This becomes even more important when the number of features are very large. You need not use every feature at your disposal for creating an algorithm. You can assist your algorithm by feeding in only those features that are really important. I have myself witnessed feature subsets giving better results than complete set of feature for the same algorithm or – "Sometimes, less is better!".

We should consider the selection of feature for model based on below criteria

i. The relationship between two independent variable should be less and
ii. The relationship between Independent and Target variables should be high.

Below figure illustrates that relationship between all numeric variables using Corrgram plot.

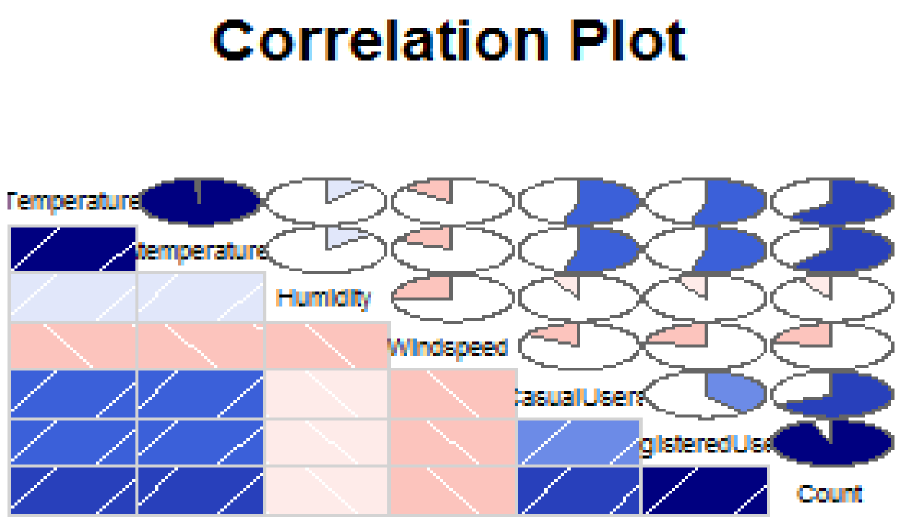

FIG 2.11 Correlation plot of numeric variables

Color dark blue indicates there is strong positive relationship and if darkness is decreasing indicates relation between variables are decreasing.

Color dark Red indicates there is strong negative relationship and if darkness is decreasing indicates relationship between variables are decreasing.

Dimensional Reduction for numeric variables

Above Fig 2.11 shows there is strong relationship between independent variables 'Temperature" and "Atemperature" so considering any one feature enough to predict the better. And it is also showing there is almost no relationship between independent variable "Humidity" and dependent variable "Count" ,so, 'Humidity' is not so important to predict.

Subsetting two independent features 'Atemperature and 'Humidity' from actual dataset.

2.2.4 Feature Scaling

The word "normalization" is used informally in statistics, and so the term normalized data can have multiple meanings. In most cases, when you normalize data you eliminate the units of measurement for data, enabling you to more easily compare data from different places. Some of the more common ways to normalize data include:

Rescaling data to have values between 0 and 1. This is usually called feature scaling. One possible formula to achieve this is.

$$\boxed{x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}}$$

In rental dataset numeric variables like 'Temperature , 'Atemperature,'Humidity' and ' Windspeed' are in normalization form so , we have to Normalize two variables 'CasualUsers' and 'RegisteredUsers'

After normalization 'CasualUsers' and 'RegisteredUsers' variables look like in table below where all values between 0 and 1

```
> df_data_deleted$CasualUsers
  [1] 0.143105698 0.056111353 0.051326664 0.046107003 0.034797738 0.037407569 0.063505872 0.028708134
  [9] 0.022618530 0.016963897 0.017833841 0.010004350 0.015658982 0.022618530 0.095693780 0.108307960
 [17] 0.050021749 0.003044802 0.033057851 0.035232710 0.031752936 0.039582427 0.064375816 0.036537625
 [25] 0.080034798 0.013919095 0.005654632 0.015658982 0.052631579 0.060026098 0.017398869 0.019573728
 [33] 0.030448021 0.025663332 0.037407569 0.042627229 0.153110048 0.051326664 0.026968247 0.022183558
> df_data_deleted$RegisteredUsers
  [1] 0.09153913 0.09384926 0.17455963 0.20704591 0.21628646 0.21628646 0.19376263 0.12575801 0.10799884
 [10] 0.18192319 0.17326018 0.16127635 0.19462893 0.19448455 0.14524978 0.13470979 0.12460295 0.09442680
 [19] 0.22408316 0.26335547 0.20906728 0.12532486 0.11781692 0.18914236 0.25685822 0.06526133 0.05717586
 [28] 0.16012128 0.13788623 0.13514294 0.20776783 0.18668784 0.20704591 0.21209934 0.23101357 0.12777938
 [37] 0.18033497 0.22697083 0.20877852 0.22119550 0.21238810 0.22769275 0.16806237 0.16921744 0.23895466
 [46] 0.27100780 0.31706613 0.33612475 0.16647416 0.12879007 0.19578400 0.25382616 0.24357494 0.19073058
 [55] 0.22018481 0.24371932 0.19419578 0.24458562 0.27187410 0.22263933 0.24689576 0.20459139 0.06800462
```

FIG 2.12 Normalized values of Registered Users and Casual Users

3.MODELLING

3.1 Model Selection

In out earlier stage of analysis we have come to understand that few variables like ‘Temperature’ ,’CasualUsers’, ’RegisteredUsers ‘ are going to play key role in model development , for model development dependent variable may fall under below categories

i. Nominal
ii. Ordinal
iii. Interval
iv. Ratio

In our case dependent variable is interval so, the predictive analysis that we can perform is Regression Analysis

We will start our model building from Decision Tree .

3.1.1 Evaluating Regression Model

The main concept of looking at what is called **residuals** or difference between our predictions f(x[I,]) and actual outcomes y[i].

We are using two methods to evaluating performance of our model.

i. **MAPE** : (Mean Absolute Percent Error) measures the size of the error in percentage terms. It is calculated as the average of the unsigned percentage error.

$$\left(\frac{1}{n}\sum\frac{|Actual - Forecast|}{|Actual|}\right)*100$$

ii. **RMSE :**(Root Mean Square Error) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$

3.2 Decision Tree

A tree has many analogies in real life, and turns out that it has influenced a wide area of **machine learning**, covering both **classification and regression**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

```
##  develop Decision tree model

# ##rpart for regression
fit = rpart(Count ~ ., data = train_feature, method = "anova")

#Predict for new test cases
predictions_DT = predict(fit, test_features[,-12])

print(fit)
#  plotting decision tree


plot(fit)
text(fit)
```

FIG 3.1 Decision Tree Model

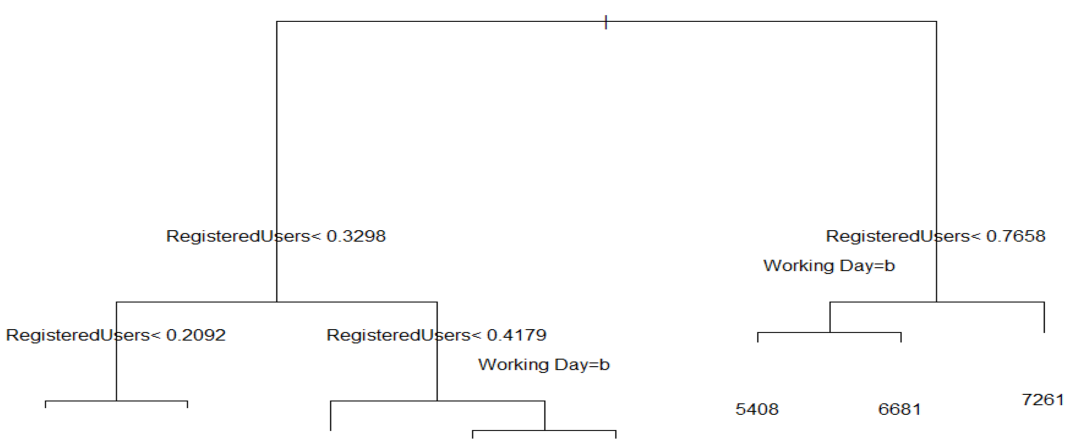

FIG 3.2 Graphical Representation of Decision tree

Look at the above figure 3.2 here decision tree is using only two predictors variables to predict the model , which is not very impressive here the model is overfitted and biased towards only two predictors i.e 'CasualUsers and RegisteredUsers .

3.2.1 Evaluation of Decision Tree Model

```
# Evaluation of Decision tree algoithm.
#MAPE
#calculate MAPE
MAPE = function(y, yhat){
  mean(abs((y - yhat)/y))
}

MAPE(test_features[,12], predictions_DT)

#Error Rate: 0.1474599
#Accuracy: 85.25%
```

```
###Evaluate  Model using RMSE

RMSE <- function(y_test,y_predict) {

  difference = y_test - y_predict
  root_mean_square = sqrt(mean(difference^2))
  return(root_mean_square)

}


RMSE(test_features[,12], predictions_DT)

#RMSE = 637.1391
#####################################################################
```

FIG 3.4 : Evaluation of Decision Tree using MAPE and RMSE

In Figure 3.2.3 Model Accuracy is 1- 0.1475 = 0.8525 which is nearly 85.25% it is not so good and RMSE is 237 which is very high so it's clearly stating that our Decision Tree Model is overfitted and it working well for training data but won't predict good for new set of data.

3.3 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

```
###develop Random Forest model

Rental_rf=randomForest(Count ~ . , data = train_feature)

RF_cnt

plot(RF_cnt)

#Predict for new test cases
predictions_rf = predict( RF_cnt , test_features[,-12])
```

FIG 3.5 Development of Random Forest

```
##MAPE
#calculate MAPE
MAPE(test_features[,12], predictions_DT_two)

#Error Rate: 0.078
#Accuracy: 92.2

###Evaluate  Model using RMSE

RMSE(test_features[,12], predictions_DT_two)

#RMSE = 270
```

FIG 3.6 Evaluation of Random Forest

Fig 3.6 shows Random Forest model performs dramatically better than Decision tree on both training and test data and well also improve the Accuracy (MAPE = 0.078) and decrease the RMSE (270) of the model which is quite impressive.

3.4 Linear Regression

Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

As Linear regression will work well if multicollinearity between the Independent variables are less.

```
#### Develop  Linear Regression Model

#check multicollearity
install.packages('usdm')
library(usdm)
vif(train_feature[,-12])

vifcor(train_feature[,-12], th = 0.9)
# Correleation between two variables is 'season' and 'Month' is 0.82 so, removing one variable from



train_feature_1 = train[,c("Year" ,"Month" ,"Holiday","Weekday","Working Day","WeatherSituation",
                           "Temperature","Windspeed","CasualUsers","RegisteredUsers","Count")]




test_features_1 = test[,c("Year" ,"Month" ,"Holiday","Weekday","Working Day","WeatherSituation",
                          "Temperature","Windspeed","CasualUsers","RegisteredUsers","Count")]

# Linear Regression  model
#run regression model
lm_model = lm(Count ~., data = train_feature_1)

#Summary of the model
summary(lm_model)

# Predict  the Test data
#Predict
predictions_LR = predict(lm_model, test_features_1[,-11])
```

FIG 3.7 : Development of Linear Regression Model.

```
Call:
lm(formula = Count ~ ., data = train_feature_1)

Residuals:
       Min         1Q     Median         3Q        Max
-2.442e-12 -2.725e-13 -3.610e-14  2.752e-13  9.696e-12

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error    t value Pr(>|t|)
(Intercept)         2.200e+01  1.710e-13  1.287e+14  < 2e-16 ***
Year1              -1.821e-13  9.081e-14 -2.006e+00 0.045407 *
Month10            -5.742e-13  1.774e-13 -3.237e+00 0.001286 **
Month11            -3.144e-13  1.496e-13 -2.101e+00 0.036147 *
Month12            -3.884e-13  1.344e-13 -2.890e+00 0.004016 **
Month2             -1.509e-13  1.312e-13 -1.149e+00 0.250883
Month3             -3.936e-13  1.458e-13 -2.700e+00 0.007159 **
Month4             -3.898e-13  1.708e-13 -2.282e+00 0.022892 *
Month5             -4.632e-13  1.952e-13 -2.373e+00 0.018020 *
Month6             -5.496e-13  2.197e-13 -2.501e+00 0.012693 *
Month7             -4.825e-13  2.385e-13 -2.024e+00 0.043528 *
Month8             -4.778e-13  2.240e-13 -2.133e+00 0.033371 *
Month9             -6.737e-13  2.073e-13 -3.250e+00 0.001230 **
Holiday1            8.438e-14  1.970e-13  4.280e-01 0.668536
Weekday1           -3.798e-13  1.399e-13 -2.715e+00 0.006845 **
Weekday2           -3.579e-13  1.431e-13 -2.501e+00 0.012689 *
Weekday3           -4.036e-13  1.461e-13 -2.763e+00 0.005935 **
Weekday4           -2.954e-13  1.459e-13 -2.024e+00 0.043452 *
Weekday5           -3.499e-13  1.331e-13 -2.629e+00 0.008811 **
Weekday6           -1.939e-13  1.140e-13 -1.701e+00 0.089525 .
`Working Day`1             NA         NA         NA       NA
WeatherSituation2   1.837e-13  6.383e-14  2.878e+00 0.004163 **
WeatherSituation3   7.431e-13  1.910e-13  3.890e+00 0.000113 ***
Temperature         4.573e-13  4.002e-13  1.143e+00 0.253705
```


```
Month5             -4.632e-13  1.952e-13 -2.373e+00 0.018020 *
Month6             -5.496e-13  2.197e-13 -2.501e+00 0.012693 *
Month7             -4.825e-13  2.385e-13 -2.024e+00 0.043528 *
Month8             -4.778e-13  2.240e-13 -2.133e+00 0.033371 *
Month9             -6.737e-13  2.073e-13 -3.250e+00 0.001230 **
Holiday1            8.438e-14  1.970e-13  4.280e-01 0.668536
Weekday1           -3.798e-13  1.399e-13 -2.715e+00 0.006845 **
Weekday2           -3.579e-13  1.431e-13 -2.501e+00 0.012689 *
Weekday3           -4.036e-13  1.461e-13 -2.763e+00 0.005935 **
Weekday4           -2.954e-13  1.459e-13 -2.024e+00 0.043452 *
Weekday5           -3.499e-13  1.331e-13 -2.629e+00 0.008811 **
Weekday6           -1.939e-13  1.140e-13 -1.701e+00 0.089525 .
`Working Day`1             NA         NA         NA       NA
WeatherSituation2   1.837e-13  6.383e-14  2.878e+00 0.004163 **
WeatherSituation3   7.431e-13  1.910e-13  3.890e+00 0.000113 ***
Temperature         4.573e-13  4.002e-13  1.143e+00 0.253705
```


```
Windspeed           1.066e-12  3.996e-13  2.668e+00 0.007880 **
CasualUsers         2.299e+03  2.492e-13  9.224e+15  < 2e-16 ***
RegisteredUsers     6.926e+03  3.083e-13  2.247e+16  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.301e-13 on 518 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.92e+32 on 25 and 518 DF,  p-value: < 2.2e-16
```

FIG 3.8 : Various Definition and Values of Linear Regression Model

Here :

Residual standard error: 6.301e-13 on 518 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

Here residual Standard error is quite less so the distance between predicted values f(x[I,]) and actaul values f(x) are very less so this model is predicted almost accurate values.

And Multiple R-Square value is 1 so, we can explain about 100 % of the data using our multiple linear regression model. This is very impressive.

```
# Evaluate Linear Regression Model

MAPE(test_features_1[,11], predictions_LR)

#Error Rate: 1.054427e-16
#Accuracy: 99.9 + accuracy

RMSE(test_features_1[,11], predictions_LR)

#RMSE = 5.104411e-13


# Conclusion  For this Dataset  Linear Regression is  Accuracy  is '99.9'
# and RMSE = 5.104411e-13
```

FIG 3.9 : Evaluation of Linear Regression Model

From above figure it is clearly showing that Model Accuracy is 99.9 % and RMSE is nearly equal to 3.9.

Model Selection:

As we predicted counts for Bike Rental using three Models Decision Tree, Random Forest and Linear Regression as MAPE is high and RMSE is less for the Linear regression Model so conclusion is

Conclusion: - For the Bike Rental Data **Linear Regression** Model is best model to predict the count.

Appendix A - Extra Figures

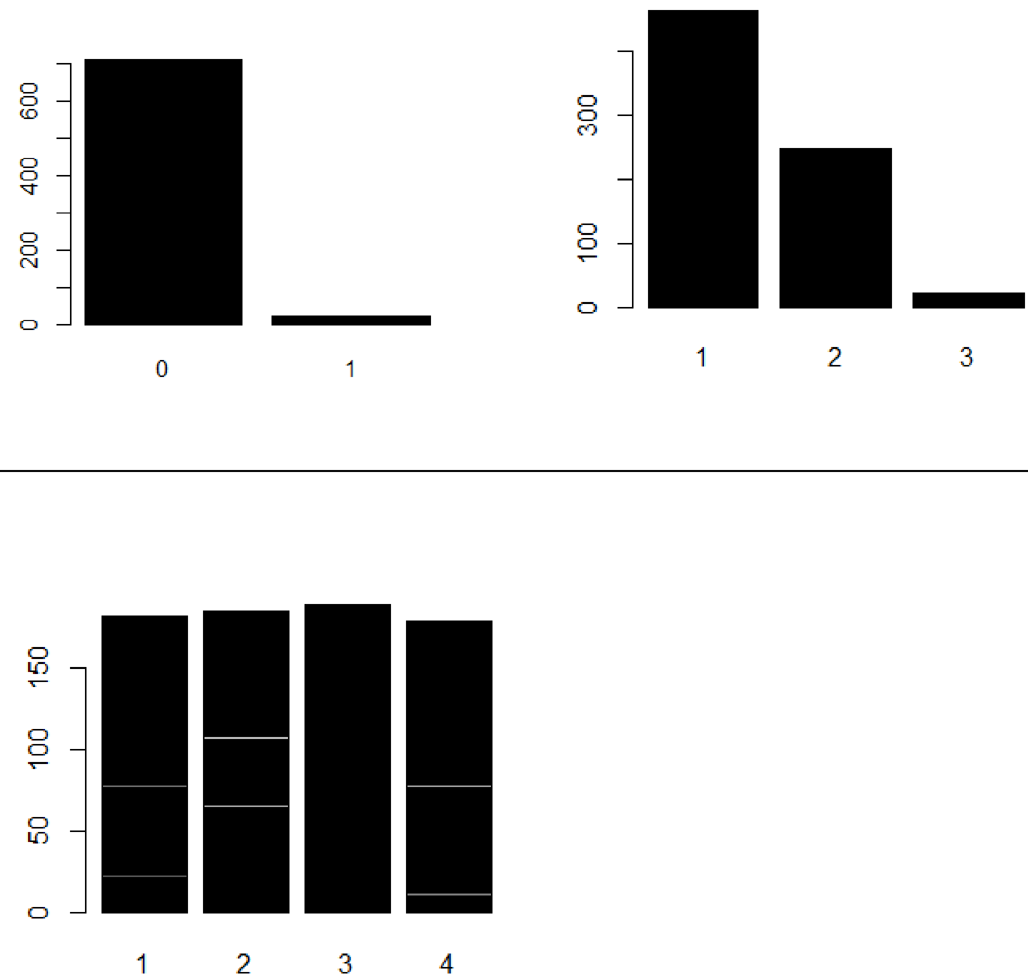

FIG: Relationship of dependent Variable Count with other Independent variables : a) Holiday b) Weather Situation c) Season (respectively)

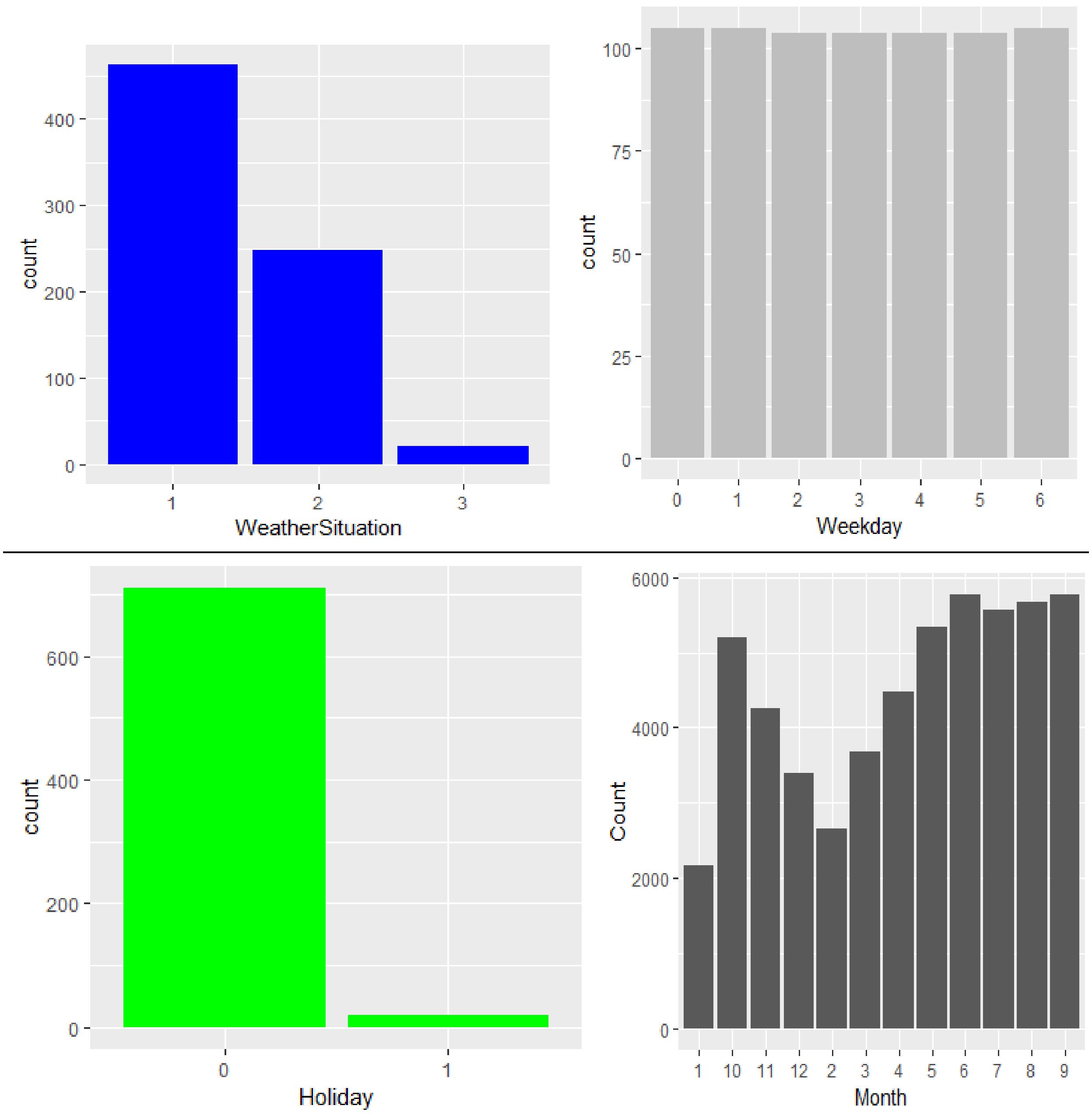

FIG: Relationship of dependent Variable Count with other Independent variables :

a) Weather Situation b) Weekday c) Holiday c) Month (respectively)

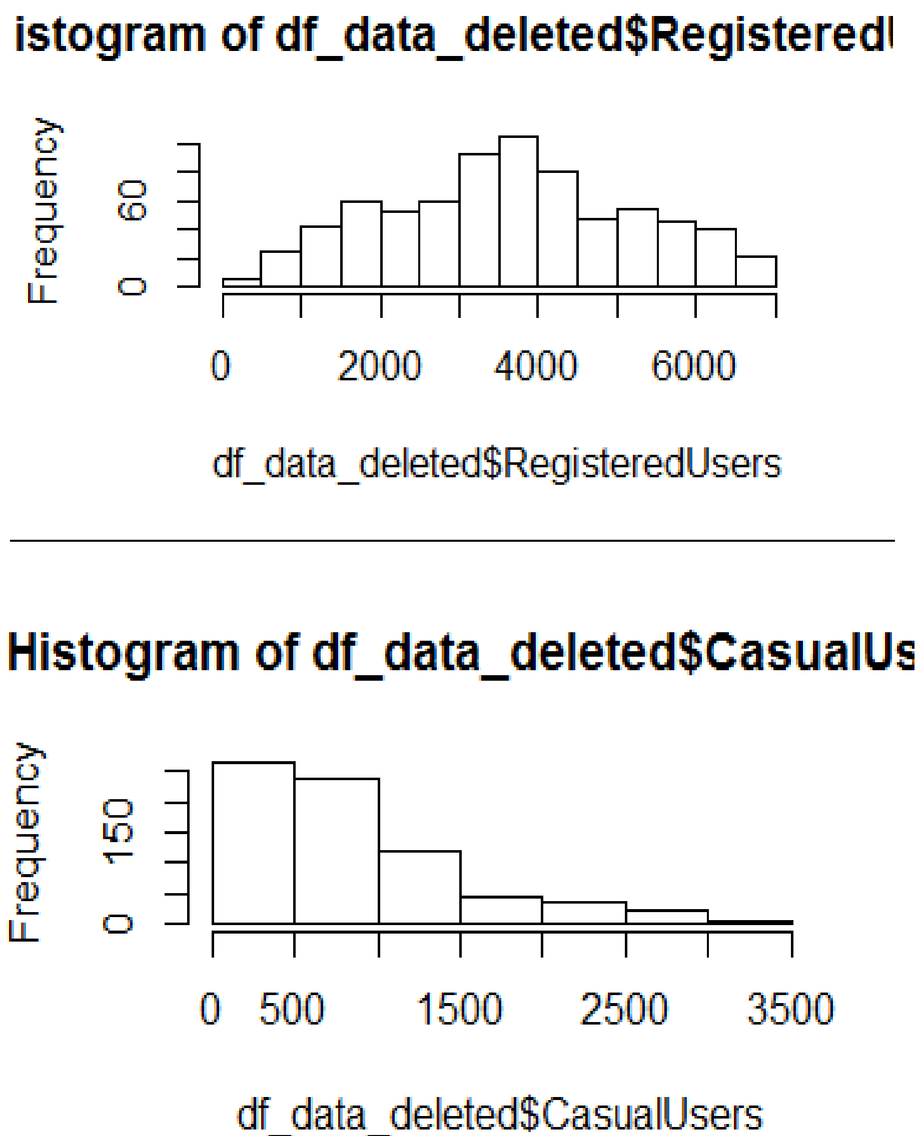

FIG : Distribution of Count in a) RegisteredUsers and b) CasualUsers before normalization.

Appendix B - R Code

```r
######################carrying out univaiate analysis####################################

# function to create univariate distribution of numeric  variables
univariate_numeric <- function(num_x) {


  ggplot(df_data)+
    geom_histogram(aes(x=num_x,y=..density..),
                   fill= "grey")+
    geom_density(aes(x=num_x,y=..density..))
}

# analyze the distribution of  target variable 'Count'
univariate_numeric(df_data$Count)

# analyse the distrubution of  independence variable 'Temperature'
univariate_numeric(df_data$Temperature)

# analyse the distrubution of  independence variable 'Atemperature'
univariate_numeric(df_data$Atemperature)

# analyse the distrubution of  independence variable 'Humidity'
univariate_numeric(df_data$Humidity)

# analyse the distrubution of  independence variable 'Windspeed'
univariate_numeric(df_data$Windspeed)

# analyse the distrubution of  independence variable 'Casual Users'
univariate_numeric(df_data$CasualUsers)

# analyse the distrubution of  independence variable 'Registered Users'
univariate_numeric(df_data$RegisteredUsers)

# Visualize categorical Variable 'Month' with target variable 'Count'
```

R code for FIG 2.1 , 2.2 , 2.3, 2.4, 2.5

```
#######################carring out Bivariate Analysis##################################
#check the relationship between 'Temperatue' and 'Atemperature' variable

ggplot(df_data, aes(x=Temperature,y=Atemperature)) +
  geom_point()+
  geom_smooth()

#This  graph shows a very strong correlation between "Temperature and Atemperature".

#check the relationship between 'Temperature' and 'Humidity' variable

ggplot(df_data, aes(x= Temperature,y=Humidity)) +
  geom_point()+
  geom_smooth()

#it shows  Humidity increases  till temparature is 0.7 and it is decreasing  gradually

#check the relationship between 'Temperature' and 'Windspeed' variable

ggplot(df_data, aes(x= Temperature,y=Windspeed)) +
  geom_point()+
  geom_smooth()

#it shows that very less negative   correlation between  Temperature and Windspeed

#check the relationship between all numeric variable using pair plot

ggpairs(df_data[,c('Atemperature','Temperature','Humidity','Windspeed','Count')])

# this plot shows that there is less +ve correlation between Count-Humidity and less -ve corelation
#and there is strong positive relationship between Temperature- Count and  Atemperature-Count
```

R code for FIG 2.6, 2.7 & 2.8

```
##########################Missing Values Analysis############################################

missing_val = data.frame(apply(df_data,2,function(x){sum(is.na(x))}))
missing_val$Columns = row.names(missing_val)
names(missing_val)[1] =  "Missing_percentage"
missing_val$Missing_percentage = (missing_val$Missing_percentage/nrow(df_data)) * 100
missing_val = missing_val[order(-missing_val$Missing_percentage),]
row.names(missing_val) = NULL
missing_val = missing_val[,c(2,1)]

# there are no missing values in the data.
#Hence no need of any analysis we will proceed with outlier process.
```

R code for FIG 2.9

```r
############################ Outlier Analysis ####################################
#we will start with the outlier analysis only on numerical variables.
numeric_index = sapply(df_data,is.numeric) #selecting only numeric
numeric_data = df_data[,numeric_index]
cnames = colnames(numeric_data)

#Creating loop for boxplot of the numercial variables.

for (i in 1:length(cnames))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (cnames[i]), x = "Count"), data = subset(df_data))+
           stat_boxplot(geom = "errorbar", width = 0.5) +
           geom_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,
                        outlier.size=1, notch=FALSE) +
           theme(legend.position="bottom")+
           labs(y=cnames[i],x="Count")+
           ggtitle(paste("Box plot of count for",cnames[i])))
  }


### Plotting plots together
 gridExtra::grid.arrange(gn1,gn5,gn2,ncol=3)
 gridExtra::grid.arrange(gn6,gn7,ncol=2)
 gridExtra::grid.arrange(gn3,gn4,ncol=2)



 # # #loop to remove from all variables
 for(i in cnames){
 print(i)
  val = df_data[,i][df_data[,i] %in% boxplot.stats(df_data[,i])$out]
  print(length(val))
  df_data = df_data[which(!df_data[,i] %in% val),]
 }
```

R code for Outlier Analysis and FIG 2.10

```r
##################################Feature Selection##############################################

## Correlation Plot

corrgram(df_data[,numeric_index], order = F,
  upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

# shows high dependence of registered users with count and temperature with  atemperature.
# since count is the target variable we will accept both registered user and count but out of temper
# and there is no  relationship between 'humidity' and 'count' therefore we will drop Humidity as we
```

R code for Feature Selection and FIG .2.11

References

- WWW.EDWISOR.COM
- WWW.ANALTICSVIDHYA.COM
- WWW.GOOGLE.COM
- WWW.YOUTUBE.COM