

BIKE RENTAL COUNT

Mainak Sarkar

22th June 2019

Contents

Table of Figures:	3
INTRODUCTION	6
1. Problem Statement	6
2. Data.....	6
Chapter 2	8
METHODOLOGY	8
1. Data Preprocessing.....	8
1.1. Missing Value Analysis:.....	8
1.2. Outlier Analysis:	10
1.3. Data Manipulation:.....	13
1.4. Feature Creation:.....	14
1.5. Removing Unnecessary Variables:	14
1.6. Exploratory Data Analysis:.....	15
1.7. Feature Selection:.....	30
1.8. Feature Scaling:	41
2. Model Building	43
2.1. Multiple Linear Regression:	43
2.2. Decision Tree Regression:	47
2.3. Random Forest Regression:	49
2.4. SVR (Support Vector Regression):.....	49
2.5. KNN (K-Nearest Neighbours):	49
Chapter 3	53
CONCLUSION	53
1. Model Evaluation:	53
a) Mean Absolute Percentage Error (MAPE):.....	53
b) Root Mean Squared Error (RMSE):	53
2. Model Selection:	54
3. Preparation of Test Cases:.....	55
4. Hyper Parameter Tuning and Model Training:	56
5. Validation of test case output:	58
6. Conclusion :.....	63
7. References :.....	63

Table of Figures:

Figure 1:Sample of data (top 6 rows)	7
Figure 2:Sample of data (bottom 6 rows)	7
Figure 3:Missing value count and percentage	9
Figure 4:Histogram of hum	10
Figure 5:Histogram of windspeed	10
Figure 6:Boxplot of hum	11
Figure 7:Boxplot of windspeed	11
Figure 8:Count and percentage of outliers	12
Figure 9:Table for detection of Best Method for Imputation	12
Figure 10:Summary of "hum"	13
Figure 11:Summary of "casual" & "registered"	13
Figure 12:Stacked Bar Plot of season v/s mean of users across different type of users	15
Figure 13:Stacked Bar Plot of year v/s mean of users across different type of users	15
Figure 14:Stacked Bar Plot of month v/s mean of users across different type of users	16
Figure 15:Stacked Bar Plot of weekday v/s mean of users across different type of users	16
Figure 16:Stacked Bar Plot of holiday v/s mean of users across different type of users	17
Figure 17:Stacked Bar Plot of workingday v/s mean of users across different type of users	17
Figure 18:Stacked Bar Plot of weathersit v/s mean of users across different type of users	18
Figure 19:Scatterplot of year v/s casual counts	18
Figure 20:Scatterplot of year v/s registered counts	19
Figure 21:Scatterplot of year v/s total counts	19
Figure 22:Scatterplot of weathersit v/s casual counts	20
Figure 23:Scatterplot of weathersit v/s registered counts	20
Figure 24:Scatterplot of weathersit v/s total counts	21
Figure 25:Scatterplot of temp v/s casual counts	21
Figure 26:Scatterplot of temp v/s registered counts	22
Figure 27:Scatterplot of temp v/s total counts	22
Figure 28:Scatterplot of feeled temp v/s casual counts	23
Figure 29:Scatterplot of feeled temp v/s registered counts	23
Figure 30:Scatterplot of feeled temp v/s total counts	24
Figure 31:Scatterplot of humidity v/s casual counts	24
Figure 32:Scatterplot of humidity v/s registered counts	25
Figure 33:Scatterplot of humidity v/s total counts	25
Figure 34:Scatterplot of windspeed v/s casual counts	26
Figure 35:Scatterplot of windspeed v/s registered counts	26
Figure 36:Scatterplot of windspeed v/s total counts	27
Figure 37:Histogram of Casual users	27
Figure 38:Histogram of Log of Casual users	28
Figure 39:Histogram of Registered users	28
Figure 40:Histogram of Log of Registered users	29
Figure 41:Correlation Plot	30

Figure 42:Correlation value of “temp” and dependent variable	30
Figure 43:Correlation value of “atemp” and dependent variable	31
Figure 44:Correlation value of “weathersit” and dependent variable	31
Figure 45:Correlation value of “hum” and dependent variable	31
Figure 46:Chi-square Test between “season” and “mnth”	32
Figure 47:Chi-square Test between “holiday” and “workingday”	32
Figure 48:Box Plot of Season v/s Cnt	33
Figure 49:ANOVA Test of Season	33
Figure 50:Box Plot of Month v/s Cnt	34
Figure 51:ANOVA Test of Month	34
Figure 52:Box Plot of Weekday v/s Cnt	35
Figure 53:ANOVA Test of Weekday	35
Figure 54:Box Plot of Holiday v/s Cnt	35
Figure 55:ANOVA Test of Holiday	36
Figure 56:Box Plot of Workingday v/s Cnt	36
Figure 57:ANOVA Test of Workingday	36
Figure 58:Box Plot of Weekday v/s Casual	37
Figure 59:ANOVA Test of Weekday	37
Figure 60:Box Plot of Holiday v/s Casual	37
Figure 61:ANOVA Test of Holiday	38
Figure 62:Box Plot of Workingday v/s Casual	38
Figure 63:ANOVA Test of Workingday	38
Figure 64:Box Plot of Weekday v/s Registered	39
Figure 65:ANOVA Test of Weekday	39
Figure 66:Box Plot of Holiday v/s Registered	39
Figure 67:ANOVA Test of Holiday	40
Figure 68:Box Plot of Workingday v/s Registered	40
Figure 69:ANOVA Test of Workingday	40
Figure 70:Q-Q Plot of atemp	41
Figure 71:Histogram Plot of atemp	41
Figure 72:Q-Q Plot of windspeed	42
Figure 73:Histogram Plot of Windspeed	42
Figure 74:Table of weathersit	43
Figure 75:Linear Regression model	44
Figure 76:VIF of model	45
Figure 77:Linear Regression model for casual counts	45
Figure 78:VIF for casual counts	46
Figure 79:Linear Regression model for registered counts	46
Figure 80:VIF for casual counts	47
Figure 81:Decision Tree for Casual Counts	48
Figure 82:Decision Tree for Casual Counts	48
Figure 83:K-values for casual	50
Figure 84:Elbow Curve for casual	51
Figure 85:K-values for registered	51

Figure 86:Elbow Curve for registered.....	52
Figure 87>Error Metrics for Casual Counts in R	54
Figure 88>Error Metrics for Casual Counts in Python	54
Figure 89>Error Metrics for Registered Counts in R	54
Figure 90>Error Metrics for Registered Counts in Python.....	54
Figure 91>Error Metrics of Total Counts in R	55
Figure 92>Error Metrics of Total Counts in Python.....	55
Figure 93:Parameter tuning for casual count for SVR model in R.....	56
Figure 94:Parameter tuning for casual count for KNN model in Python.....	56
Figure 95:Parameter tuning for registered count for SVR model in R	57
Figure 96:Parameter tuning for registered count for RF model in Python.....	57
Figure 97:Stacked Bar Plot of season for output variables.....	58
Figure 98:Stacked Bar Plot of year for output variables.....	58
Figure 99:Scatterplot of year vs total count of users for output variable	59
Figure 100:Stacked Bar Plot of weekday for output variables in R	59
Figure 101:Stacked Bar Plot of weekday for output variables in Python	60
Figure 102:Stacked Bar Plot of holiday for output variables	60
Figure 103:Stacked Bar Plot of weathersit for output variables	61
Figure 104:Scatterplot of weathersit vs total count of users for output variable	61
Figure 105:Scatterplot of atemp vs total count of users for output variable	62
Figure 106:Scatterplot of windspeed vs total count of users for output variable	62

Chapter 1

INTRODUCTION

1. Problem Statement

A bike rental company has collected historical data about the count of bike rentals for past 2 years. The objective of this case is predication of daily bike rental count based on the environmental and seasonal settings. Our task is to apply different regression models on the data and provide them with a system that could predict the number of rentals on a particular day accurately.

2. Data

The historical data that has been collected contains the following variables:

- **instant** – record index
- **dteday** – date on which the data is collected
- **season** – (1: springer, 2: summer, 3: fall, 4: winter)
- **yr** - year (0: 2011, 1:2012)
- **mnth** - month (1 to 12)
- **holiday** - whether it is a holiday or not (extracted from holiday schedule)
- **weekday** - an integer indicating the day of the week (0: Sunday, 6: Saturday)
- **workingday** - If the day is neither weekend nor holiday then value is 1, otherwise 0.
- **weathersit** - extracted from Freemeteeo (**1**: Clear, Few clouds, Partly cloudy, Partly cloudy; **2**: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist; **3**: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds; **4**: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog).
- **temp** - Normalized temperature in Celsius.
The values are derived via $(t - t_{\min}) / (t_{\max} - t_{\min})$; $t_{\min} = -8$, $t_{\max} = +39$.
- **atemp** – Normalized feeled temperature in Celsius.
The values are derived via $(t - t_{\min}) / (t_{\max} - t_{\min})$; $t_{\min} = -16$, $t_{\max} = +50$.
- **hum** – Normalized humidity. The values are divided to 100 (max).
- **windspeed** – Normalized windspeed. The values are divided to 67 (max).
- **casual** - count of casual users.
- **registered** - count of registered users.
- **cnt** - count of total rental bikes including both casual and registered.

Here “casual”, “registered” and “cnt” are the dependent variables that we need to predict.

Given below is a sample of dataset from the top and bottom of the dataset respectively:

```
> head(data)
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.1604460	331	654	985
2	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.2485390	131	670	801
3	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.2483090	120	1229	1349
4	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.1602960	108	1454	1562
5	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.1869000	82	1518	1600
6	6	2011-01-06	1	0	1	0	4	1	1	0.204348	0.233209	0.518261	0.0895652	88	1518	1606

Figure 1:Sample of data (top 6 rows)

```
> tail(data)
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
726	726	2012-12-26	1	1	12	0	3	1	3	0.243333	0.220333	0.823333	0.316546	9	432	441
727	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
728	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
729	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
730	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
731	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

Figure 2:Sample of data (bottom 6 rows)

Chapter 2

METHODOLOGY

1. Data Preprocessing

Data preprocessing or data cleaning is one of the most crucial step of building a machine learning model. Almost 80% of the time is dedicated to the data preprocessing. Because if we feed messy or uncleaned data to the model then it will generate irrelevant and wrong results. In the data mining process the data need to be pre-processed first to make them quality data to acquire the quality analysis and information to make quality decision.

The first step of data preprocessing is to check the class of each variable and then transform them as required.

Real world data are generally incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), Noisy (containing errors or outliers) and Inconsistent (containing discrepancies in codes or names), so to prepare the data for mining by using following processes is known as data preprocessing.

1.1. Missing Value Analysis:

In statistics, missing data or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. Missing data can occur because of nonresponse or no information is provided for one or more items or for a whole unit.

Sometimes the data is found to contain a lot of missing values. Also, missing data may reduce the precision of calculated statistics because there is less information than originally planned. Thus, missing value analysis is a very important part of data cleaning. It can be done in two ways:

- Detecting and deletion of the rows containing missing values
- Imputing the missing values by statistical methods like - mean, median or by KNN imputation or by prediction

The figure below shows the number of missing values in our dataset and their percentages with respect to respective columns.

Bike_Rental_Count_Mainak_R.R		missing_value
Filter		
	apply.data..2..function.x...	percentage
instant	0	0
dteday	0	0
season	0	0
yr	0	0
mnth	0	0
holiday	0	0
weekday	0	0
workingday	0	0
weathersit	0	0
temp	0	0
atemp	0	0
hum	0	0
windspeed	0	0
casual	0	0
registered	0	0
cnt	0	0

Figure 3:Missing value count and percentage

So we can see that there is no missing values in our dataset.

1.2.Outlier Analysis:

In statistics, an outlier is a data point that differs significantly 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.

The outliers of a dataset can be understood by checking the histogram plots of the variables.

Histogram of Humidity

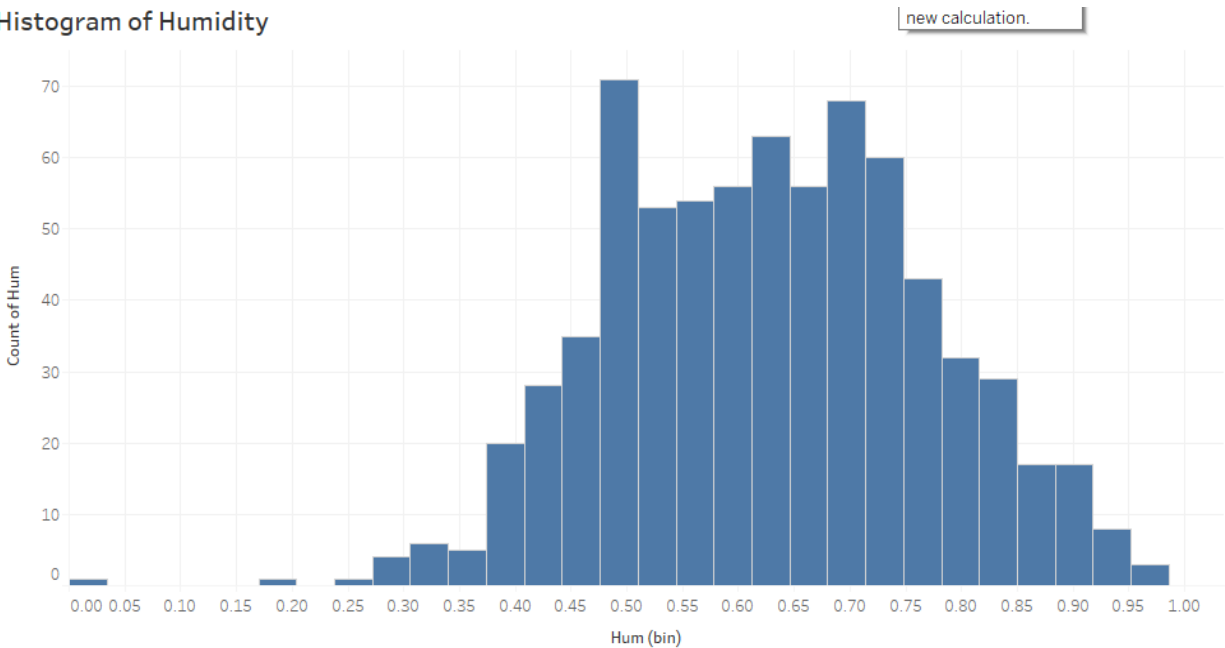


Figure 4:Histogram of hum

Histogram of Windspeed

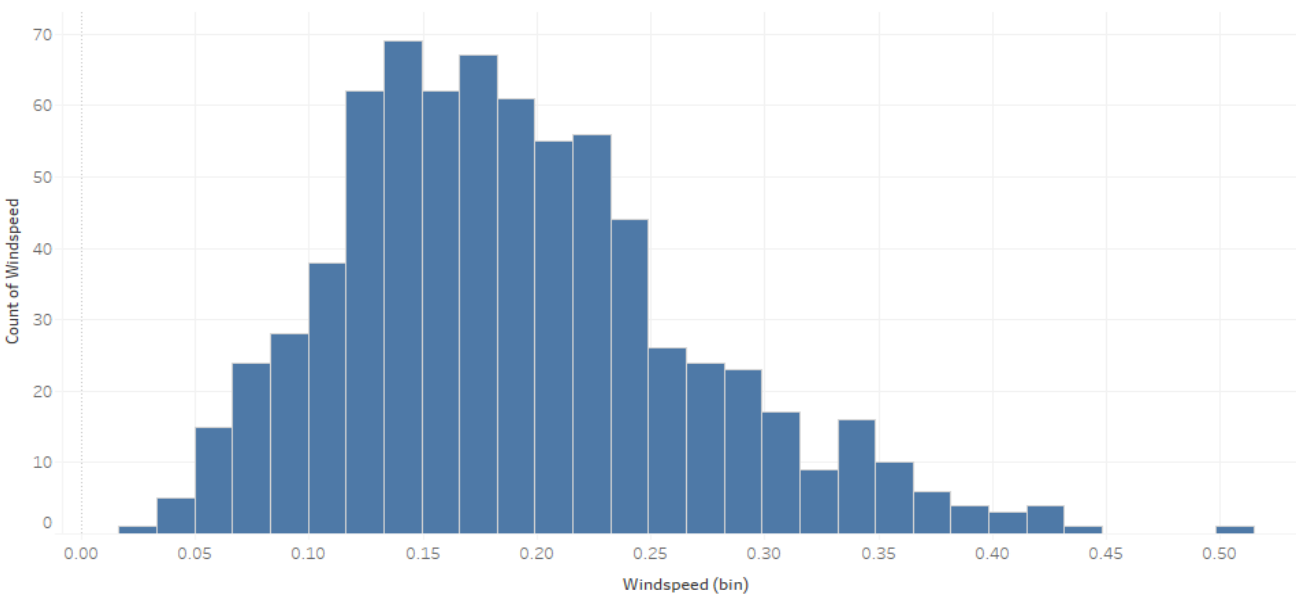


Figure 5:Histogram of windspeed

We can see from the distributions that there are some bins in 'hum' and 'windspeed' variables which are away from the main bin. So, this might be due to the presence of outliers and extreme values.

Now we have carried out our investigation further by checking the box plots of the variables.

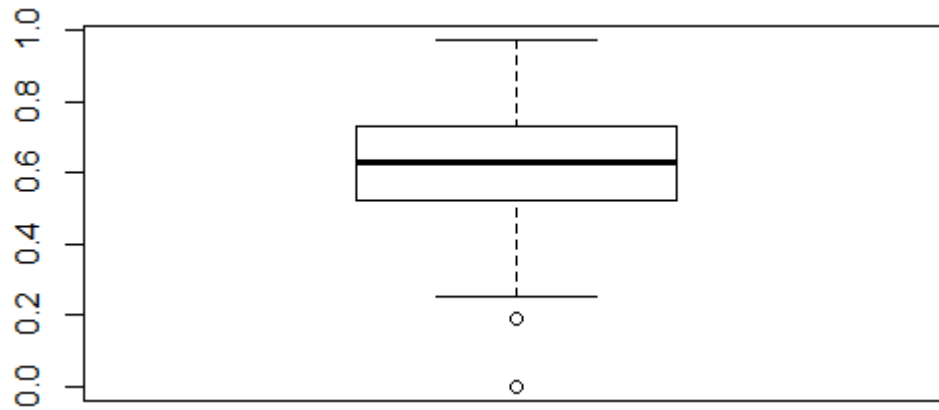


Figure 6:Boxplot of hum

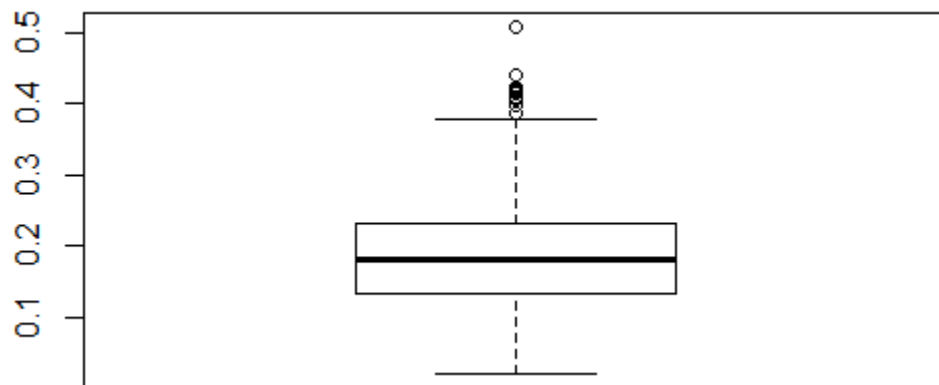


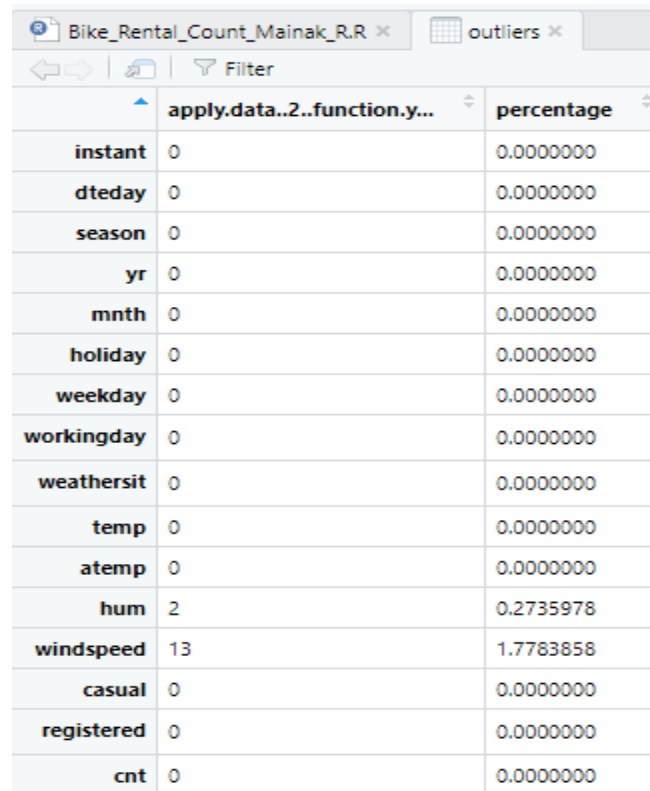
Figure 7:Boxplot of windspeed

So here the data points above the upper fence and below the lower fence (outside the box plot) show the presence of outliers.

Dealing with these outliers is a very essential part of our analysis. It can be done by following two processes:

- Deleting the outliers
- Replacing the outliers with NAs and then imputing them by statistical methods like-mean, median or by KNN imputation or by prediction

The figure below shows the number of NAs after replacing the outliers with NAs and their percentages with respect to respective columns.

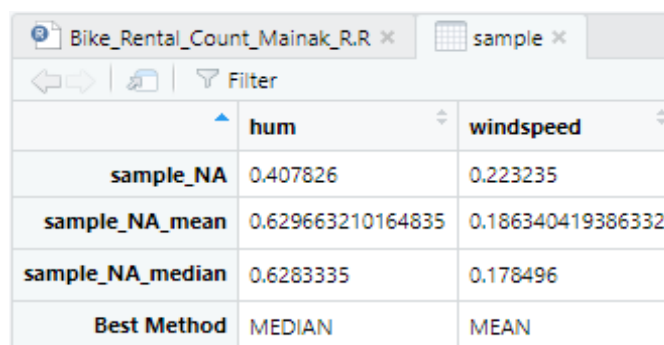


	apply.data..2..function.y...	percentage
instant	0	0.0000000
dteday	0	0.0000000
season	0	0.0000000
yr	0	0.0000000
mnth	0	0.0000000
holiday	0	0.0000000
weekday	0	0.0000000
workingday	0	0.0000000
weathersit	0	0.0000000
temp	0	0.0000000
atemp	0	0.0000000
hum	2	0.2735978
windspeed	13	1.7783858
casual	0	0.0000000
registered	0	0.0000000
cnt	0	0.0000000

Figure 8:Count and percentage of outliers

As we know that presence of outliers can affect our models a lot and as the no. of observations in the dataset is small, so on deleting them will result in loss of data. Thus, we have opted for imputation of the values.

An algorithm is designed to find the best method for imputing outliers with respect to each column. The output table shows the following:



	hum	windspeed
sample_NA	0.407826	0.223235
sample_NA_mean	0.629663210164835	0.186340419386332
sample_NA_median	0.6283335	0.178496
Best Method	MEDIAN	MEAN

Figure 9:Table for detection of Best Method for Imputation

So, we have applied the above mentioned methods to the respective columns to impute in place of NAs and get rid of the outliers.

****NOTE :** We know that the hum and windspeed can take any values in real case scenarios. Like on the day of storm the windspeed would be very high which will result in low count of bike rentals. So, the values detected here as outliers can actually come to the model during test cases. But as we know that most of the statistical models are based on the assumption that the data should not contain any outliers, so we have eliminated the outliers instead of keeping them.

1.3. Data Manipulation:

The “temp”, “atemp”, “hum” and “windspeed” variables are in their normalized form. But to understand the values and data better we have converted them to their actual form. In the problem sheet the method of conversion of these variables to their normalized values are given. So we have just applied the inverse method to convert them to actual values.

Next we have applied different real-life constraint check to our variables:

- i) **hum** : We know that the humidity on a particular day can never be more than or equal to 100 and less than or equal to 0. So, checking the summary of the “hum” variable.

```
summary(data$hum)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 25.42  52.23   62.75   62.94  73.02   97.25
```

Figure 10:Summary of “hum”

- ii) **casual & registered** : We know that the count of bike can never be less than 0. So checking the summary of “casual” and “registered” count.

```
summary(data$casual)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.0   315.5   713.0   848.2  1096.0  3410.0
summary(data$registered)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   20   2497   3662   3656   4776   6946
```

Figure 11:Summary of “casual” & “registered”

- iii) **cnt** : We know that the “cnt” variable represents total count which is the sum of “casual” and “registered” variables. So, we have checked if there is any row in “cnt” variable which voids the above condition. And we got 0 such rows, so there is no irrelevant rows.
- iv) **Yr** : Year is kept as a numerical variable so that our model is able to capture the relationship between the given years and predict for future years.
- v) **Weathersit** : The data set contains only 3 types of weathersit, but according to problem statement there are 4 types. So, we kept weathersit as numerical variable so that it could capture the relationship between them and can predict for instances of number 4.

1.4. Feature Creation:

Here for this problem we have tried creating some new features from our existing variables to see if they could explain the variance of the dependent variable better.

- **week_number** : We have extracted the week number from the “dteday” variable to check if the rental count follow any patterns depending on the week number. But this variable is highly collinear with the month and season variable. So, we have omitted this variable for further consideration.
- **weekend** : we have created the “weekend” variable to see if a date is weekend or not and if the weekend have some impact on the rental counts. The value of “weekend” is 1, when “holiday” and “workingday” both are 0; i.e.- it is a weekend if it is neither a holiday nor a working day. We have noticed that the rental count increases over the weekend, so It is a significant variable. But it shows high collinearity with “holiday”, “weekday” and “workingday”. Thus, we have deleted this variable from our analysis.
- **avg_temp** : This variable contain values which is the average of “temp” and “atemp” variables. But this variable shows high multicollinearity with “temp” and “atemp”, moreover the “atemp” variable shows better correlation with the dependent variable. So, we have not considered this variable as well.

***NOTE : As all the above 3 variables doesn't hold any value for our analysis so we have deleted those variables from our code to make the code compact and more understandable.

1.5. Removing Unnecessary Variables:

We have omitted “instant” and “dteday” variable as they doesn't hold any value to predict the dependent variable.

1.6.Exploratory Data Analysis:

It involves visualizing the data for understanding and analyzing purpose and finding different insights.

- **Stacked Bar Plot of season v/s mean of users across different type of users:**

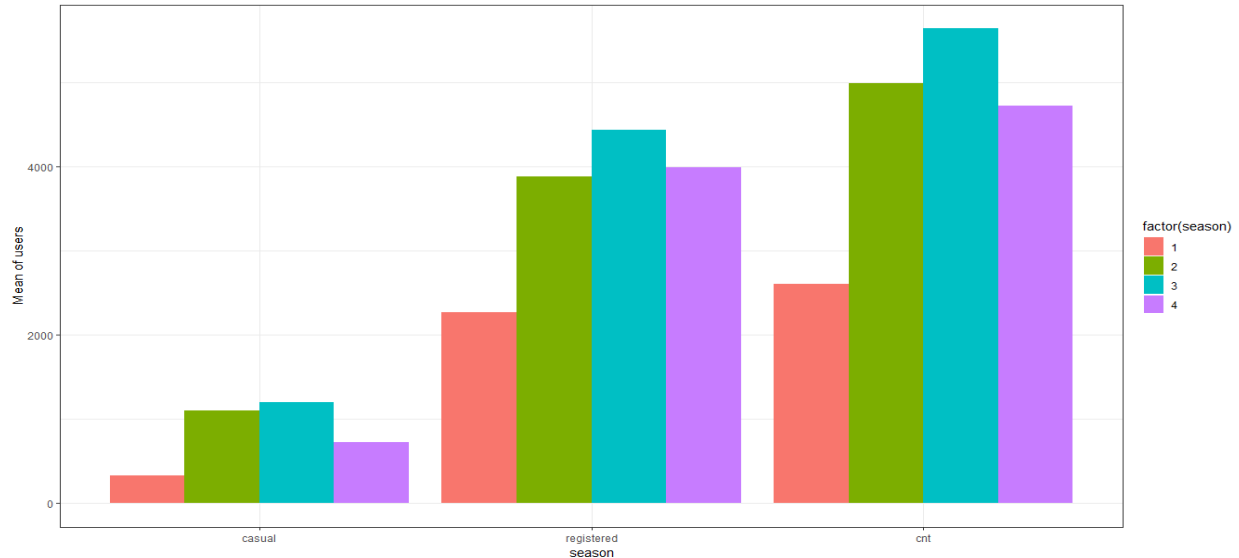


Figure 12:Stacked Bar Plot of season v/s mean of users across different type of users

We can see that casual, registered and total count all are highest for season 3 and lowest for season 1. So, when season is "fall" people are more likely to rent a bike and during "springer" they are least likely.

- **Stacked Bar Plot of year v/s mean of users across different type of users:**

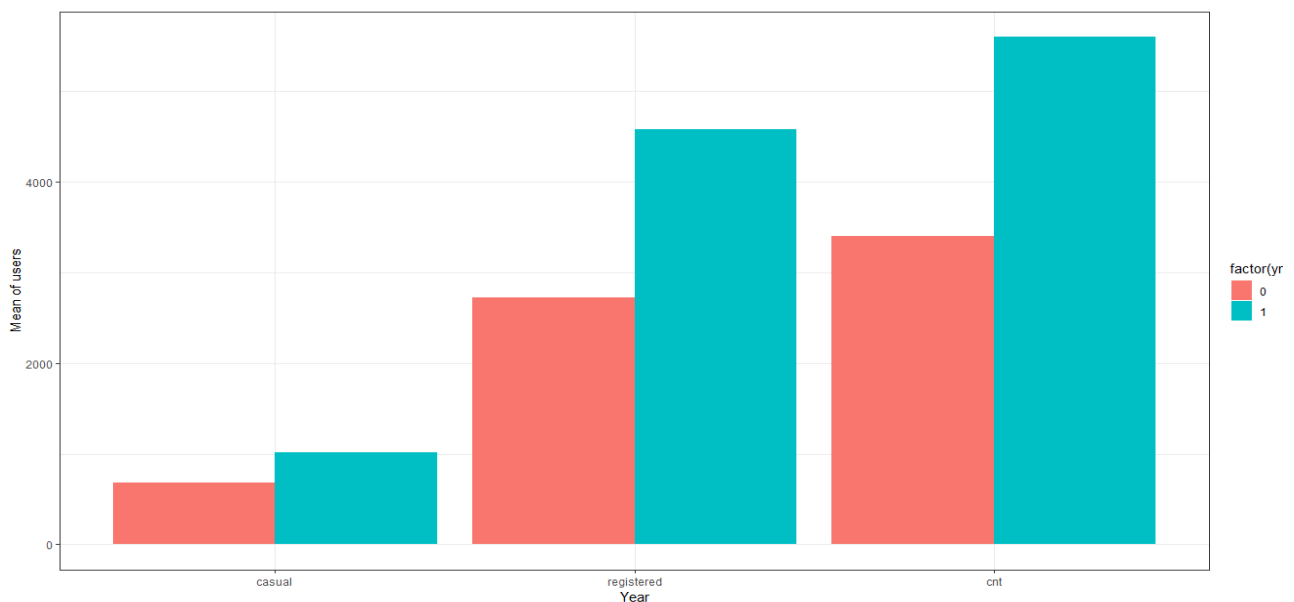


Figure 13:Stacked Bar Plot of year v/s mean of users across different type of users

From the plots we can say that the count of registered users had heavily increased in 2012 compared to 2011 which is almost 1.5 times and casual users also increased a bit, so the rental company is doing well. So, the rental will be higher for future years as well.

- **Stacked Bar Plot of month v/s mean of users across different type of users:**

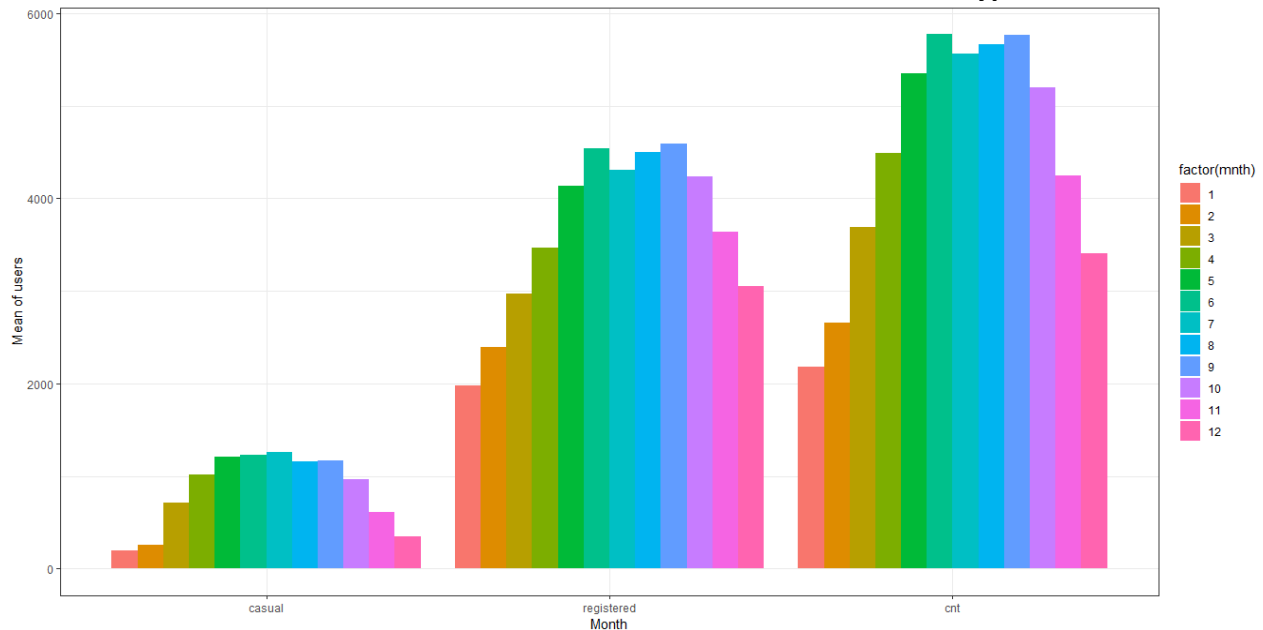


Figure 14:Stacked Bar Plot of month v/s mean of users across different type of users

we can see that the number of users is comparatively more from month 5 to 9 and is lowest in month 1. So, we can see that this variable is describing almost the same information like season.

- **Stacked Bar Plot of weekday v/s mean of users across different type of users:**

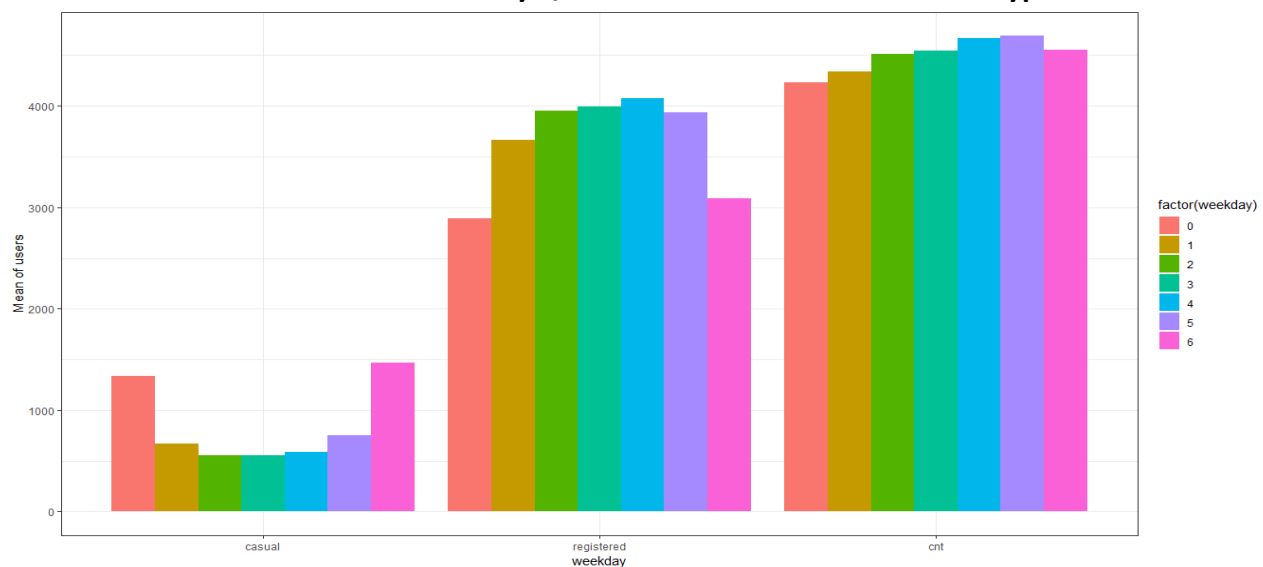


Figure 15:Stacked Bar Plot of weekday v/s mean of users across different type of users

we can see that the number of casual users is more during weekends as casual users take bikes for trips. While number of registered users is more during weekdays as they mainly consist of office workers. And mean of total count is almost same across days.

- **Stacked Bar Plot of holiday v/s mean of users across different type of users:**

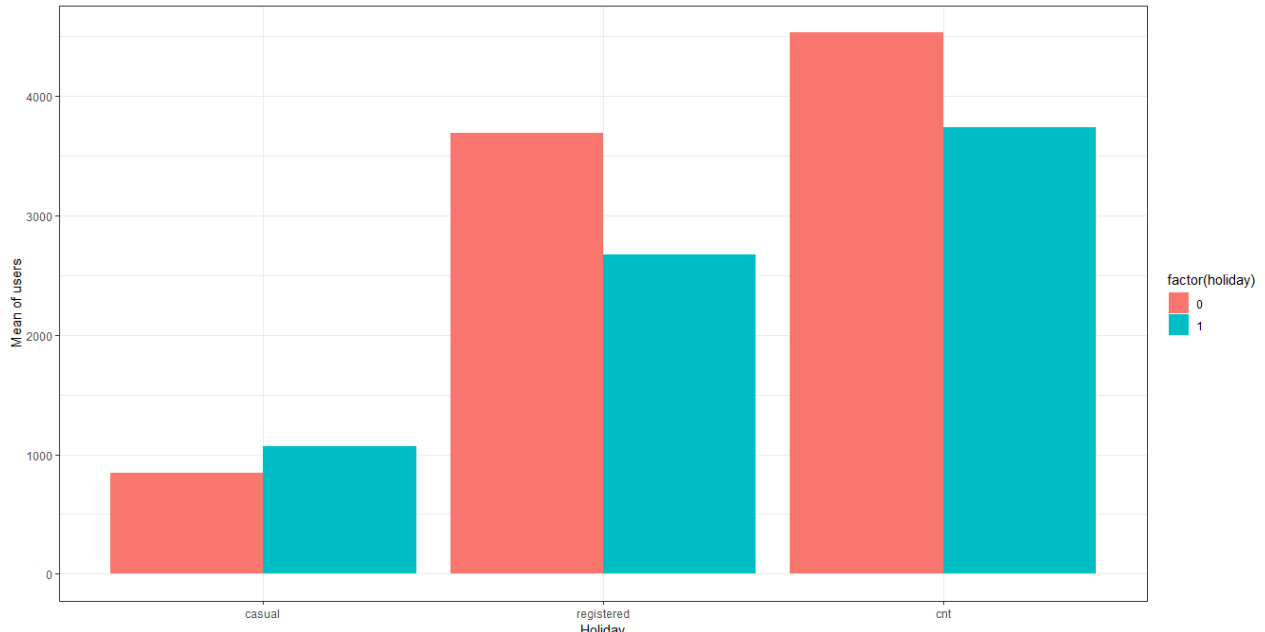


Figure 16: Stacked Bar Plot of holiday v/s mean of users across different type of users

So, if it is a holiday the number of casual users is more and registered users is less as compared to a non-holiday.

- **Stacked Bar Plot of workingday v/s mean of users across different type of users:**

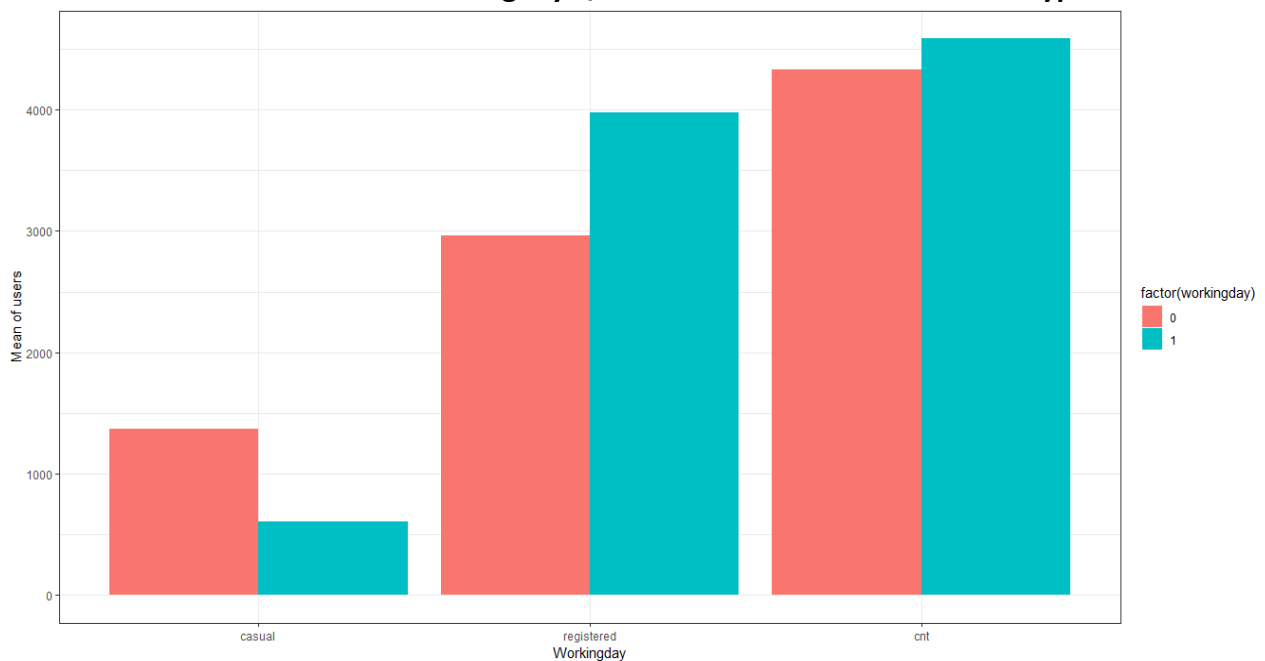


Figure 17: Stacked Bar Plot of workingday v/s mean of users across different type of users

Naturally if it is a working day the number of registered users is more as people goes to work by renting a bike. And on non-working day the count of casual users is high. So, it gives almost same information like the holiday variable.

- **Stacked Bar Plot of weathersit v/s mean of users across different type of users:**

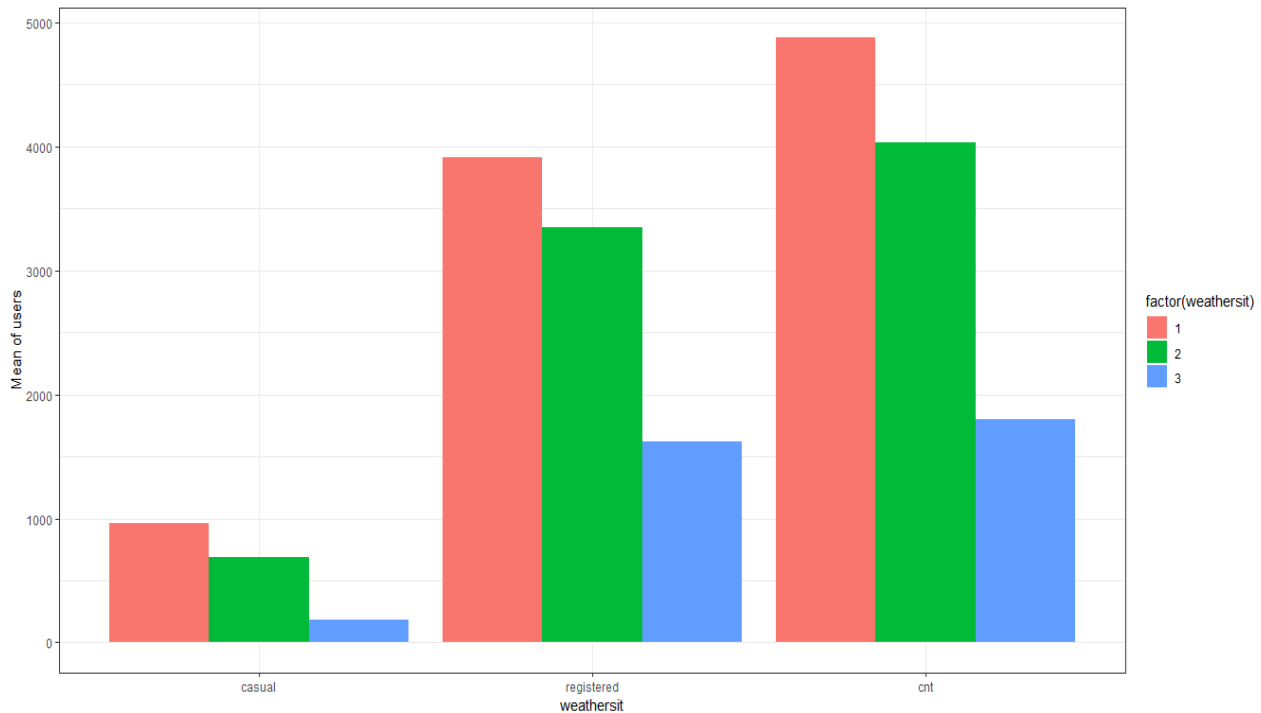


Figure 18:Stacked Bar Plot of weathersit v/s mean of users across different type of users

So, we can say that weather situation no. 1 is most preferable for bike renting. And weather situation number 4 will be least as the weather condition is deteriorating from 1 to 4.

- **Scatter plot of Year and Different type of users:**

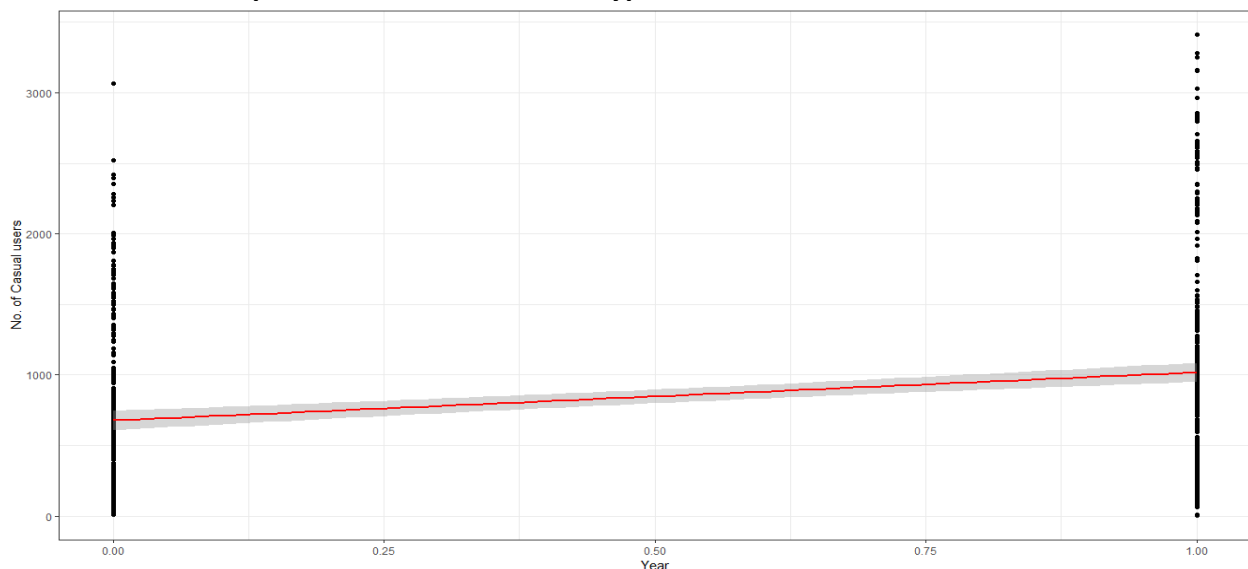


Figure 19:Scatterplot of year v/s casual counts

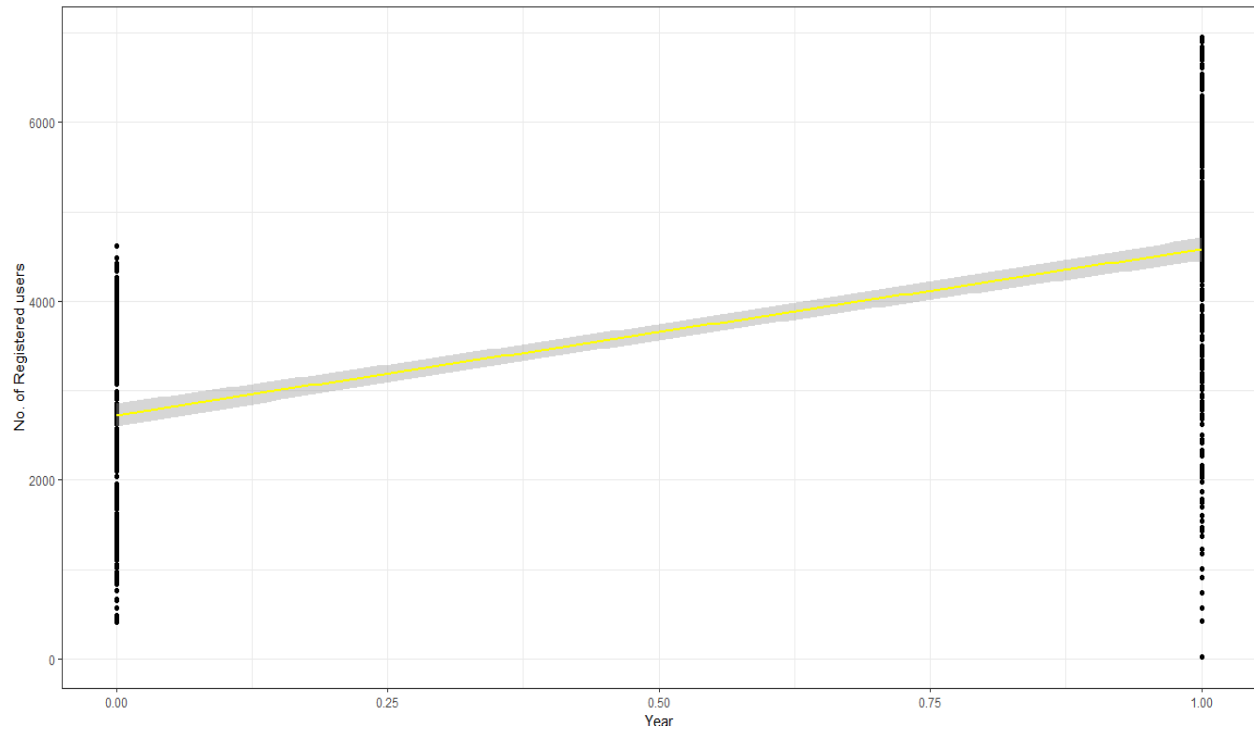


Figure 20:Scatterplot of year v/s registered counts

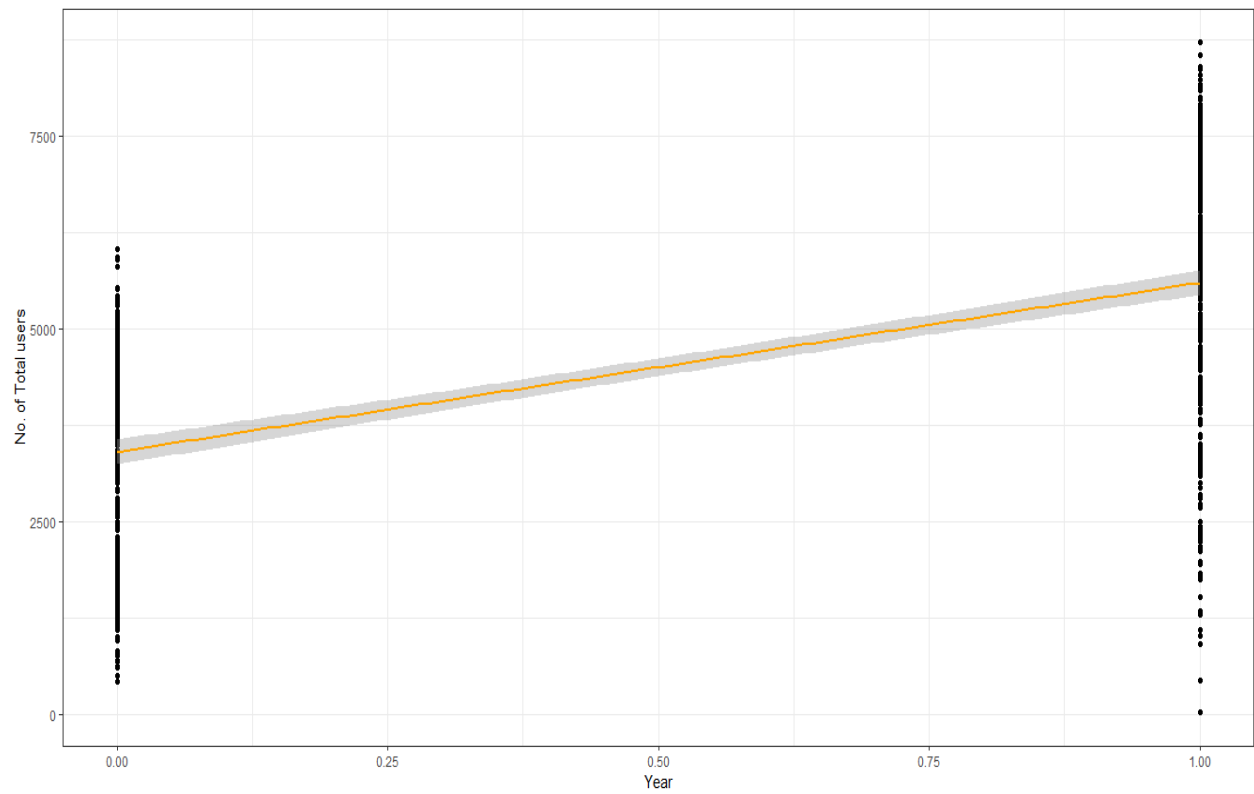


Figure 21:Scatterplot of year v/s total counts

The regression line shows a positive relationship between year and count of users. So, we can say that the no. of rentals is progressively increasing.

- Scatter plot of Weather Situation and Different type of users:

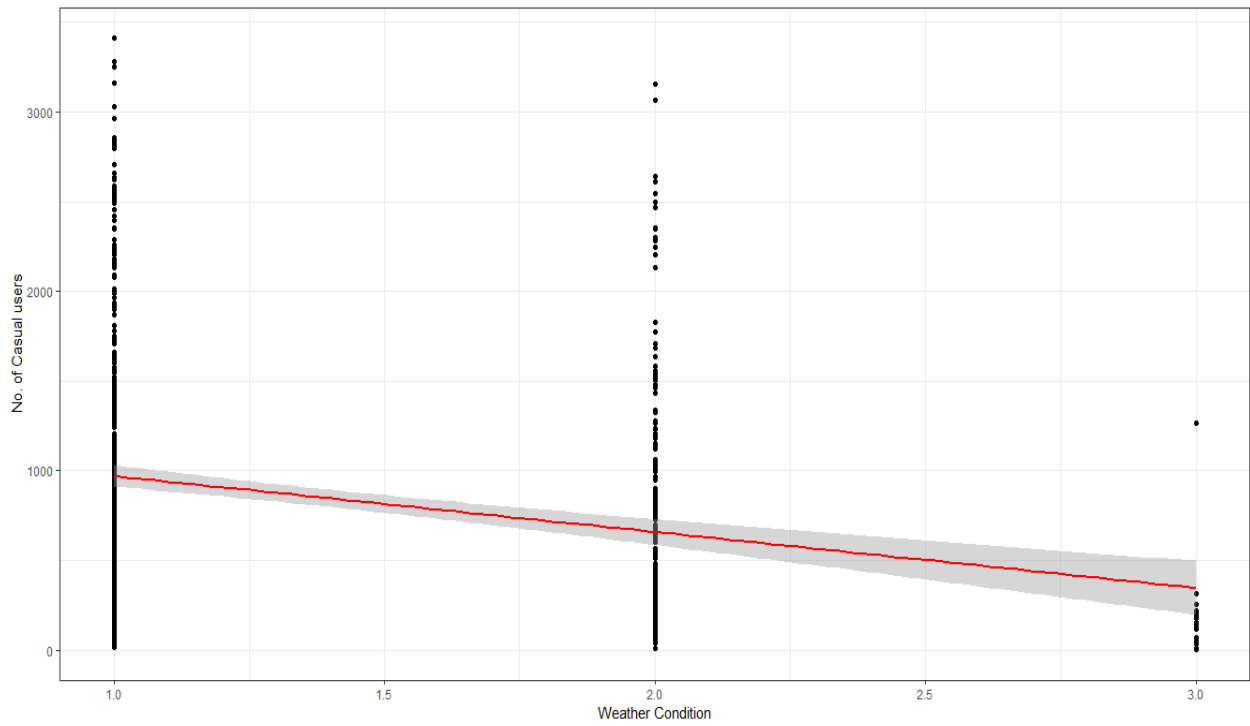


Figure 22:Scatterplot of weathersit v/s casual counts

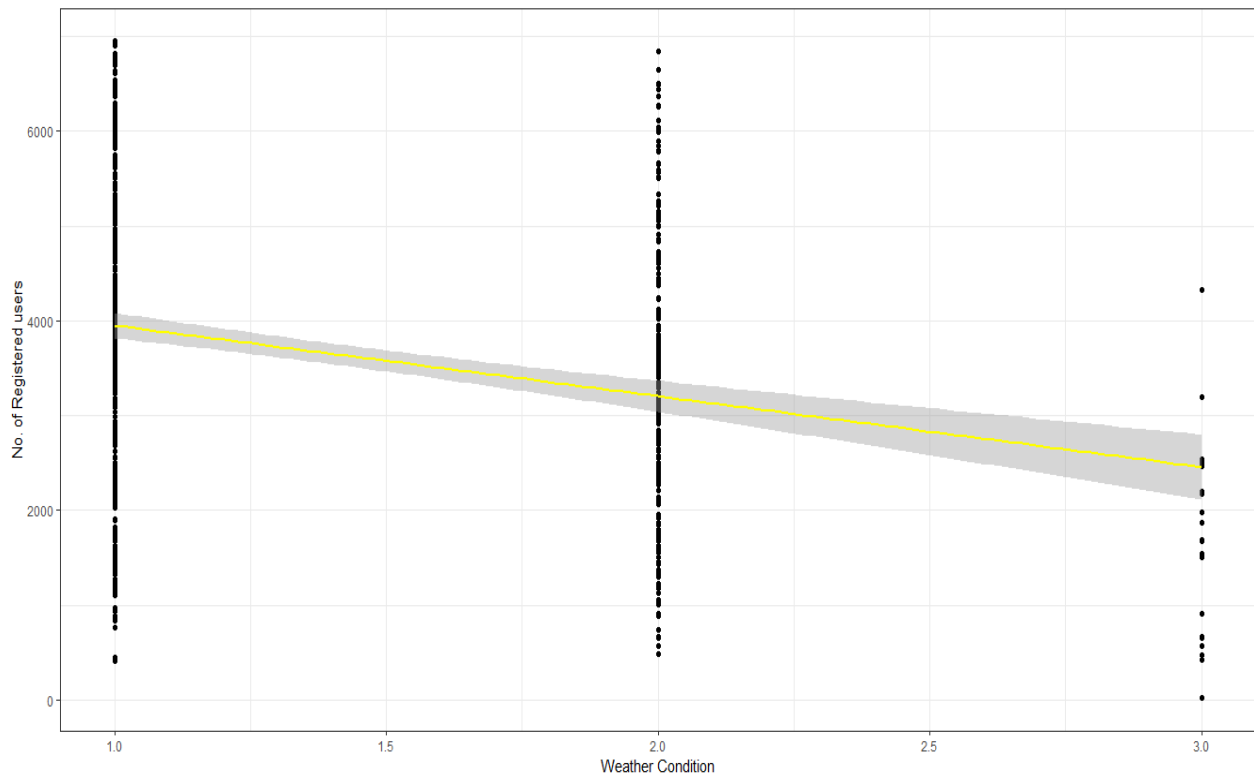


Figure 23:Scatterplot of weathersit v/s registered counts

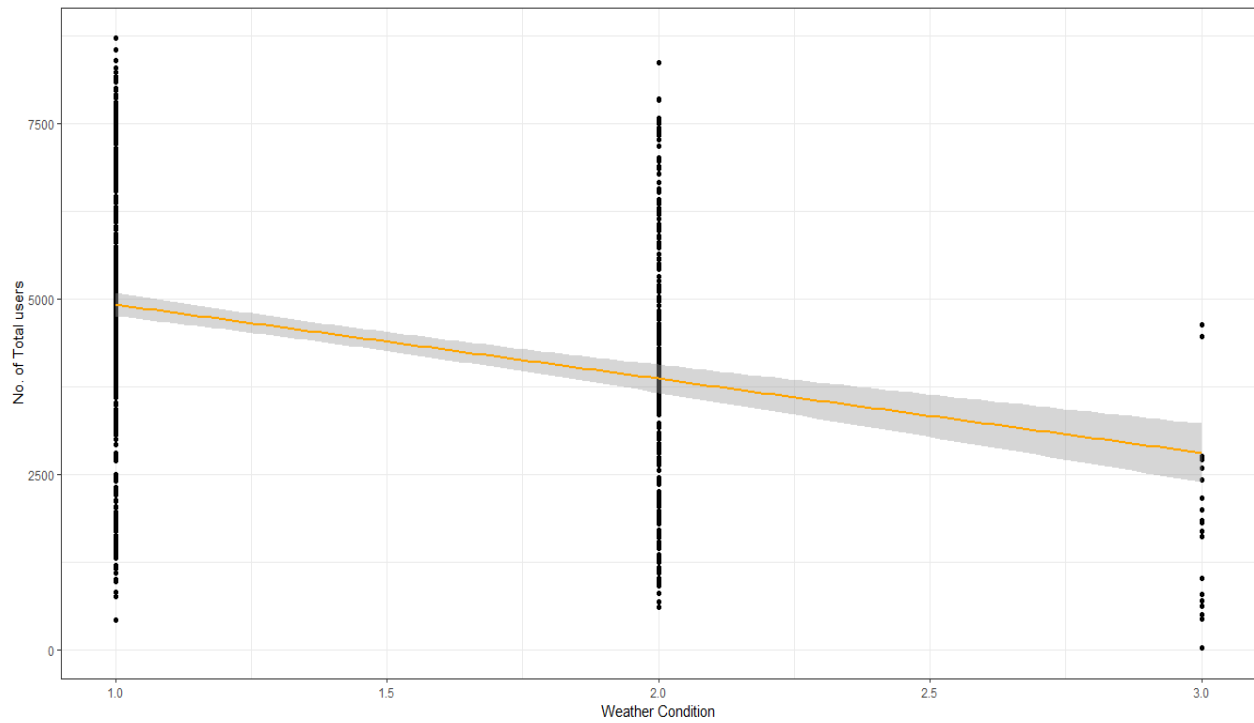


Figure 24:Scatterplot of weathersit v/s total counts

The regression line shows a negative relationship between weather and number of users. So as the weather value is increasing the weather condition is getting worse so number of rentals is decreasing. So, for weather condition 4 the count should be lesser.

- **Scatter plot of Temperature and Different type of users:**

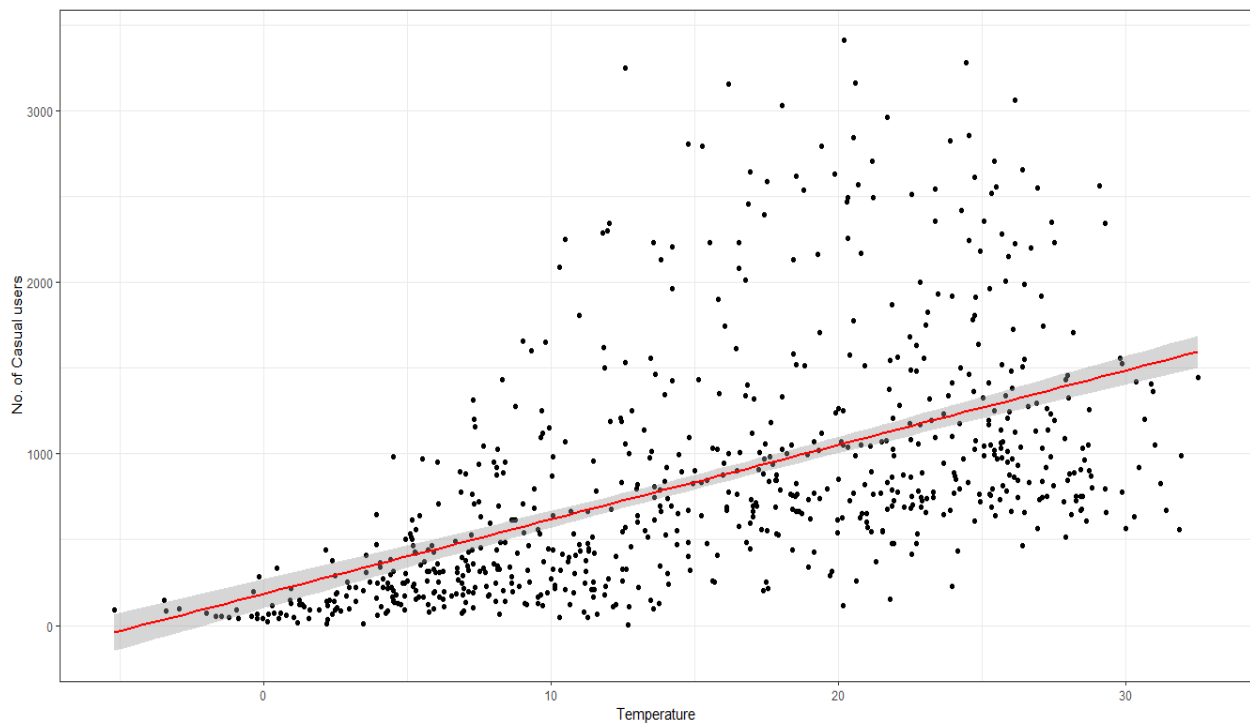


Figure 25:Scatterplot of temp v/s casual counts

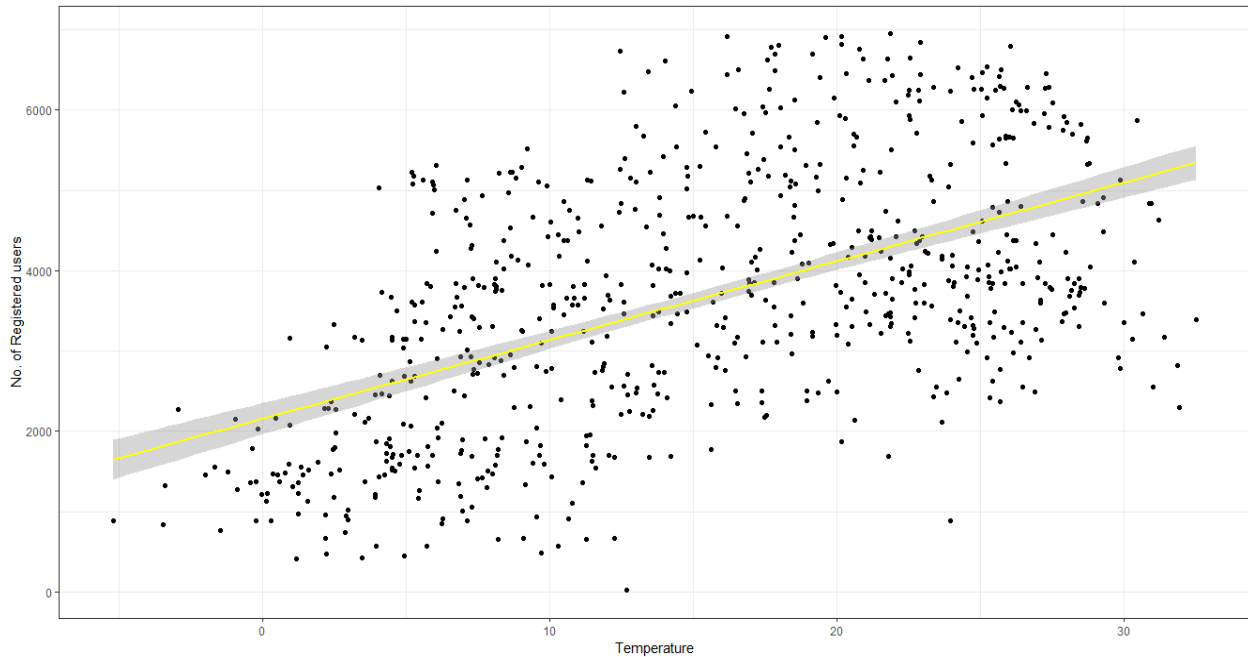


Figure 26: Scatterplot of temp v/s registered counts

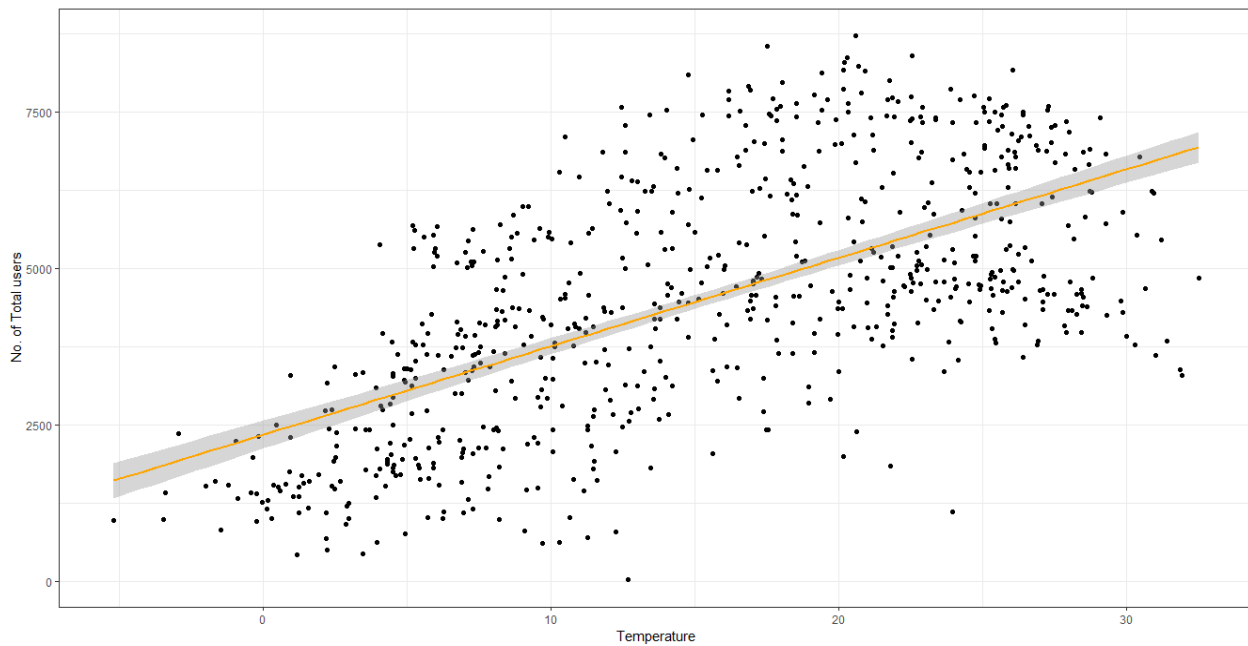


Figure 27: Scatterplot of temp v/s total counts

From the we can see that the curve is steeper in case of registered users as they have to rent bike for offices, thus we can see at least some good number of users even in case of extremes of temperature while in case of casual users, if temperature is too low then number of users is low and if temperature is too high then number of casual users decreases, and mainly we can see most casual users during moderate temperatures.

- Scatter plot of Feeled Temperature and Different type of users:

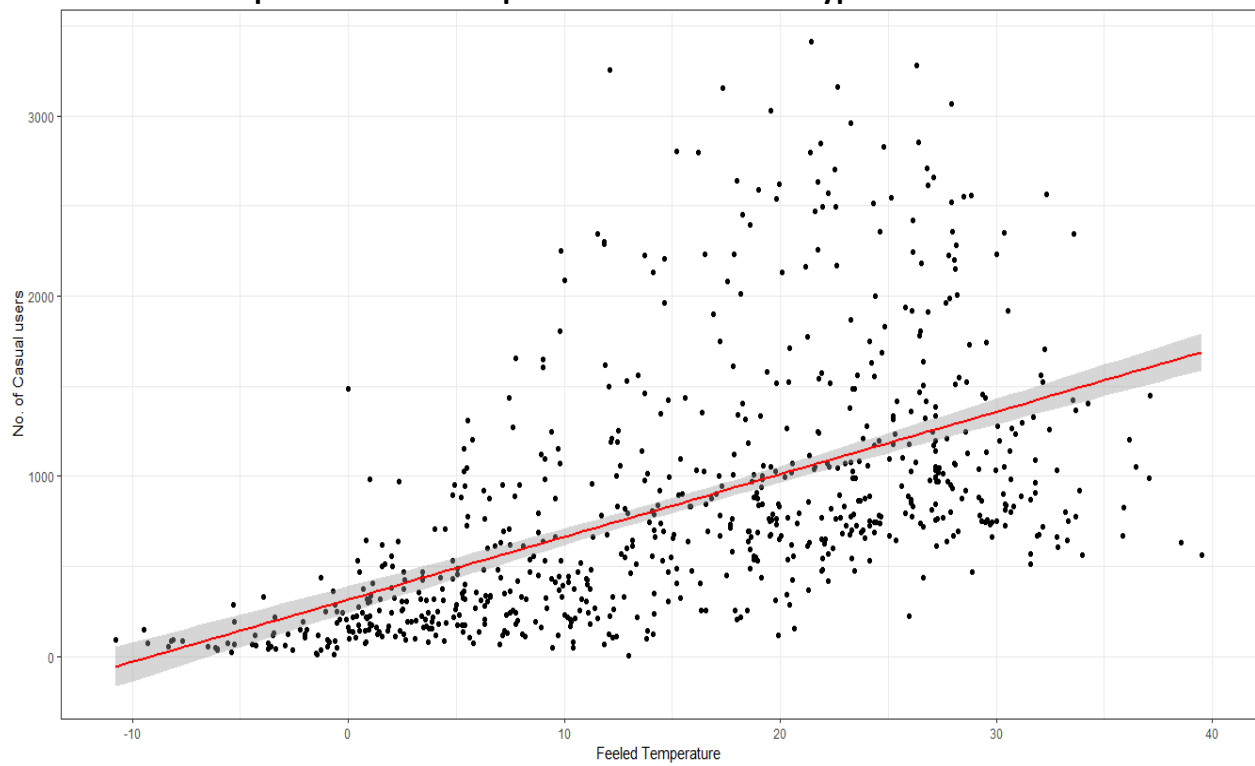


Figure 28: Scatterplot of feeled temp v/s casual counts

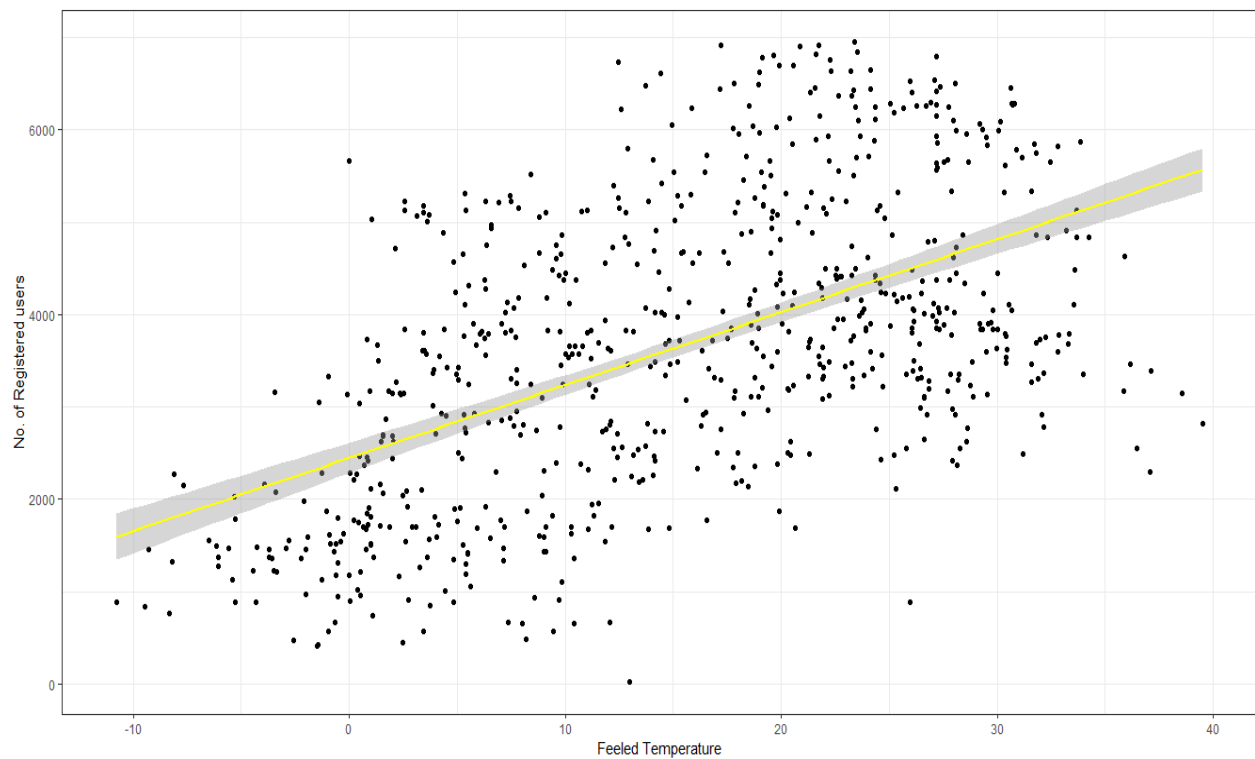


Figure 29: Scatterplot of feeled temp v/s registered counts

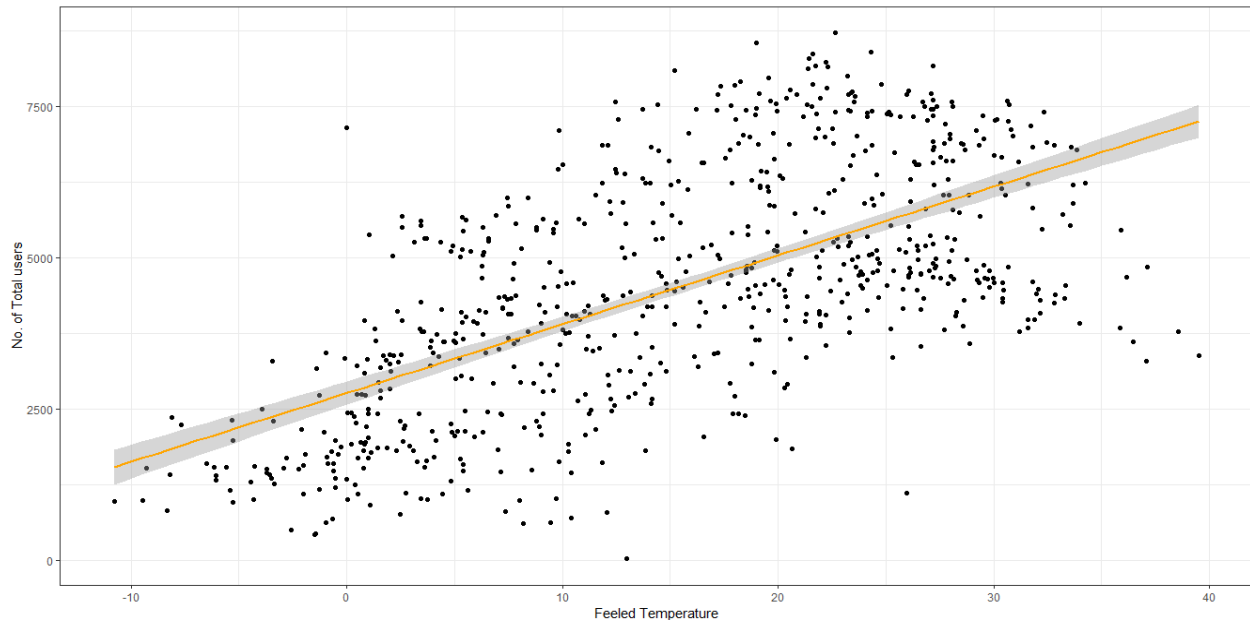


Figure 30:Scatterplot of felted temp v/s total counts

From the we can see that the curve is steeper in case of registered users as they have to rent bike for offices, thus we can see at least some good number of users even in case of extremes of temperature. In case of casual users, if temperature is too low then number of users is low and if temperature is too high then number of casual users decreases, and mainly we can see most casual users during moderate temperatures.

We can say this variable gives us almost the same information as temp.

- **Scatter plot of Humidity and Different type of users:**

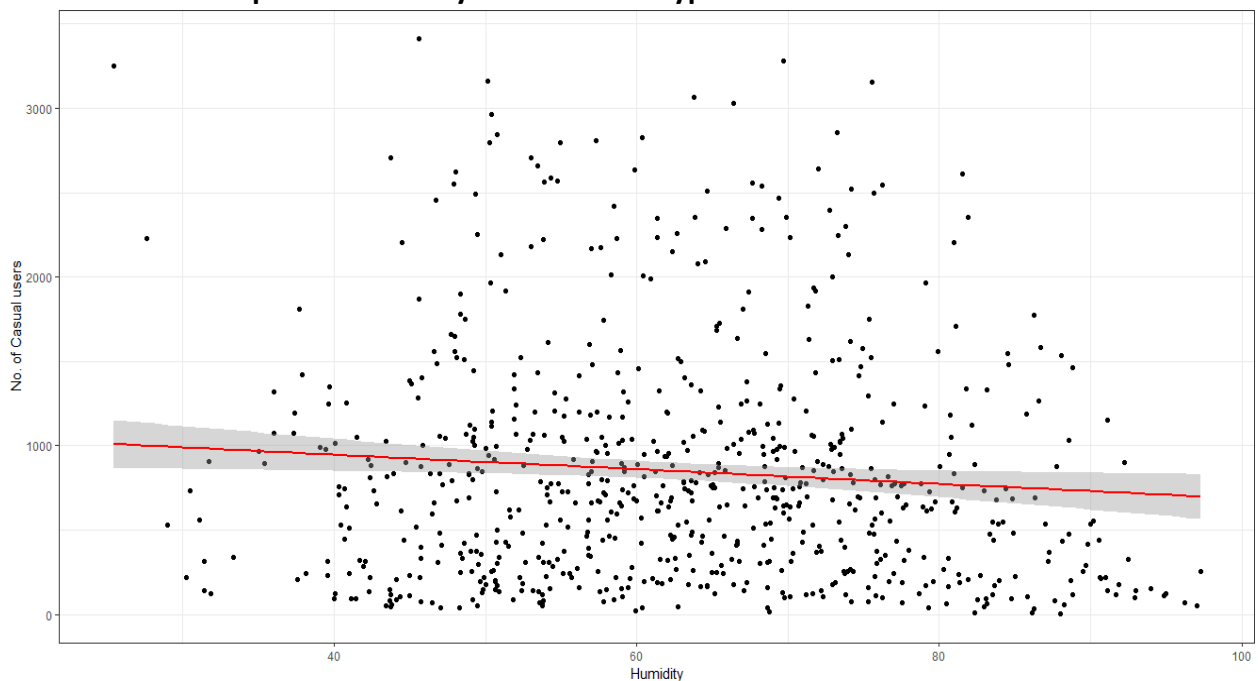


Figure 31:Scatterplot of humidity v/s casual counts

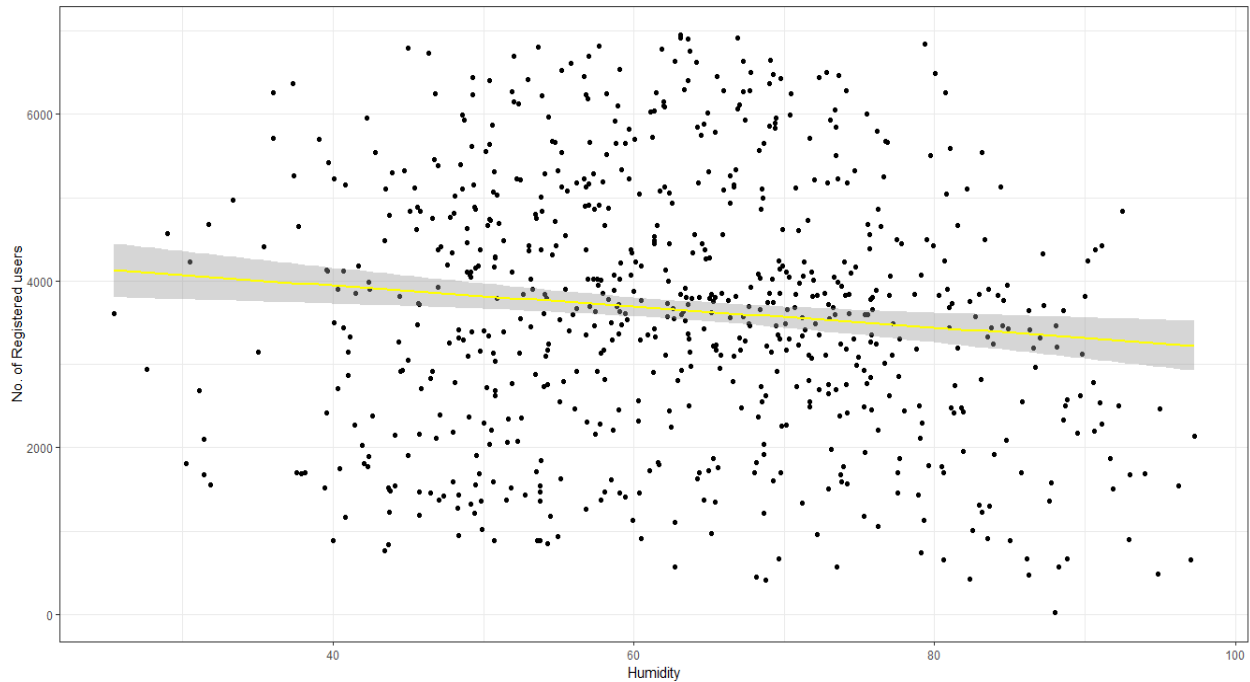


Figure 32:Scatterplot of humidity v/s registered counts

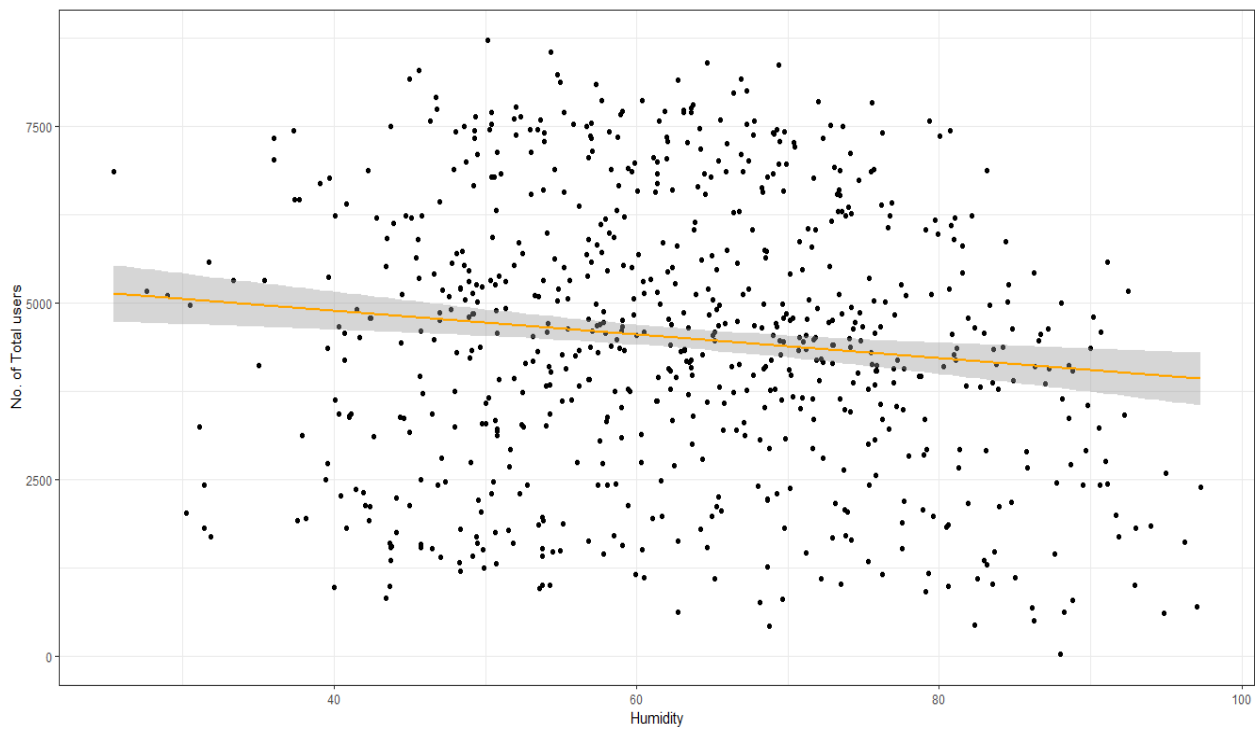


Figure 33:Scatterplot of humidity v/s total counts

From the plot we can see that the regression line is almost straight. So, no. of users is very less dependent on humidity. Thus, change of humidity doesn't affect the rental count very much except in extreme cases. Humidity is somehow negatively related to the number of users. As humidity increases people are a bit less likely to rent a bike.

- Scatter plot of Windspeed and Different type of users:

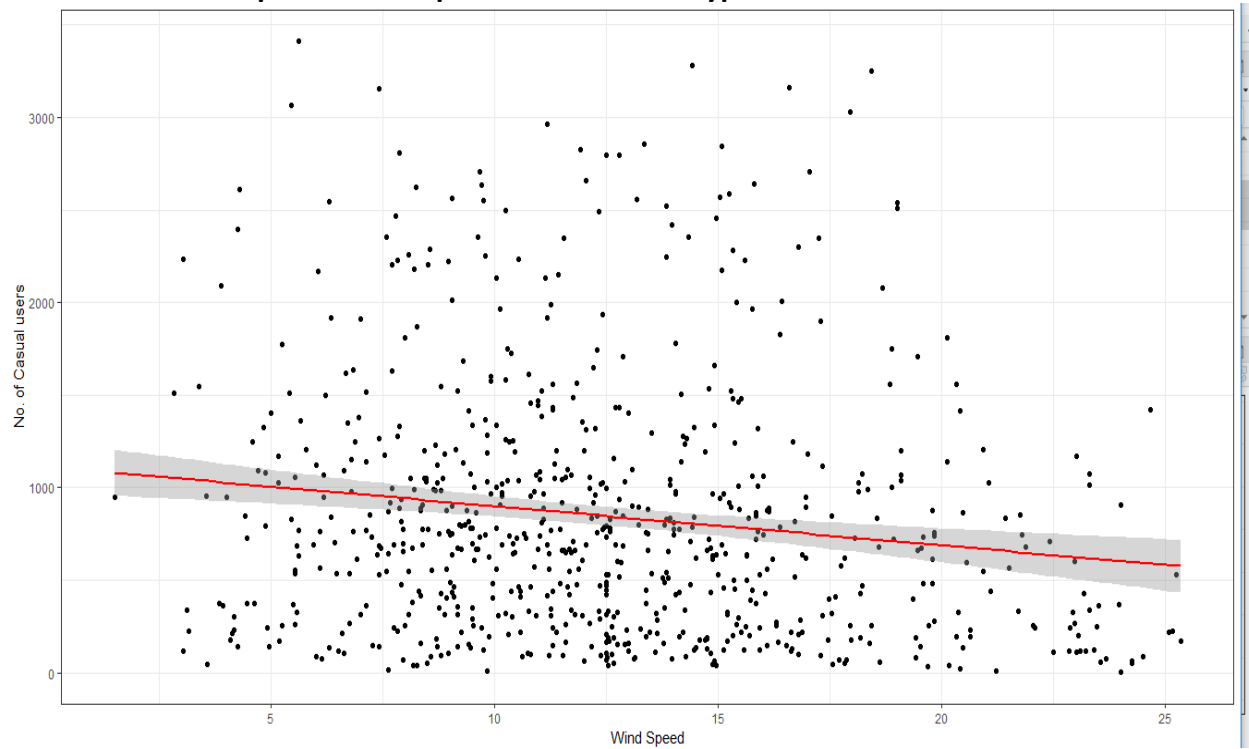


Figure 34:Scatterplot of windspeed v/s casual counts

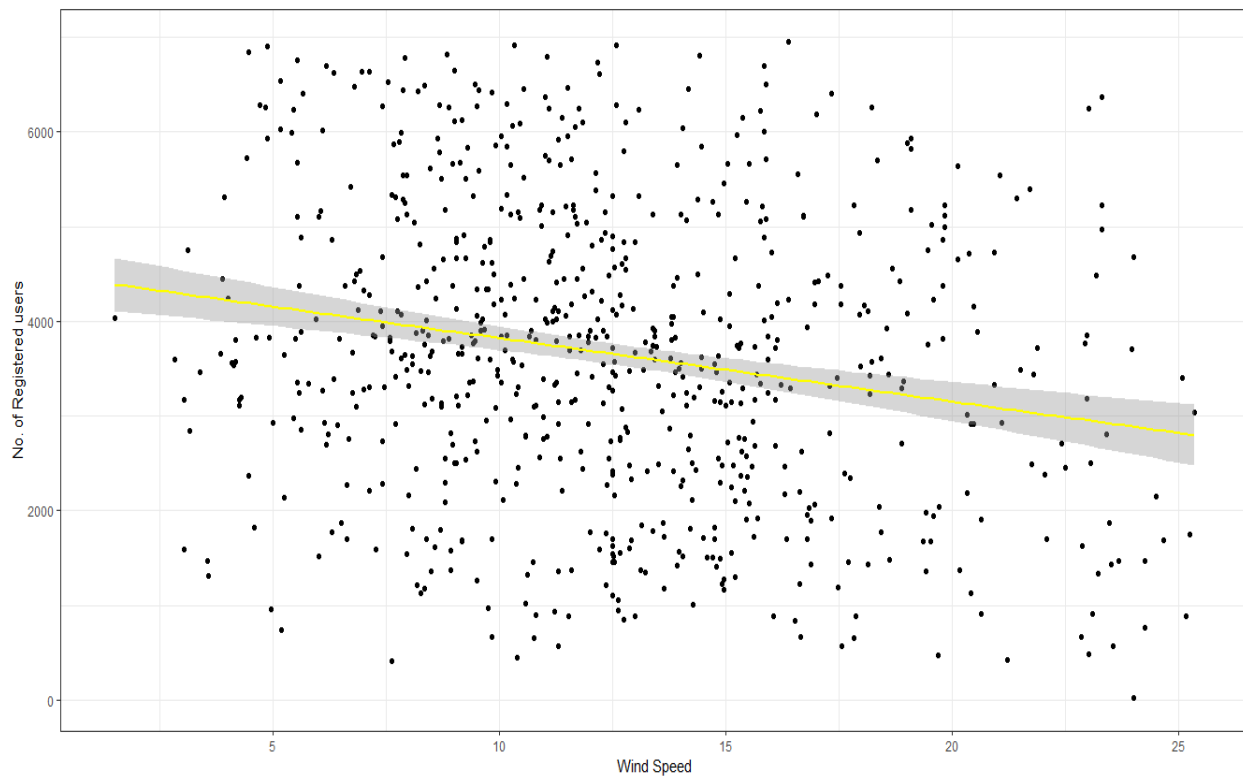


Figure 35:Scatterplot of windspeed v/s registered counts

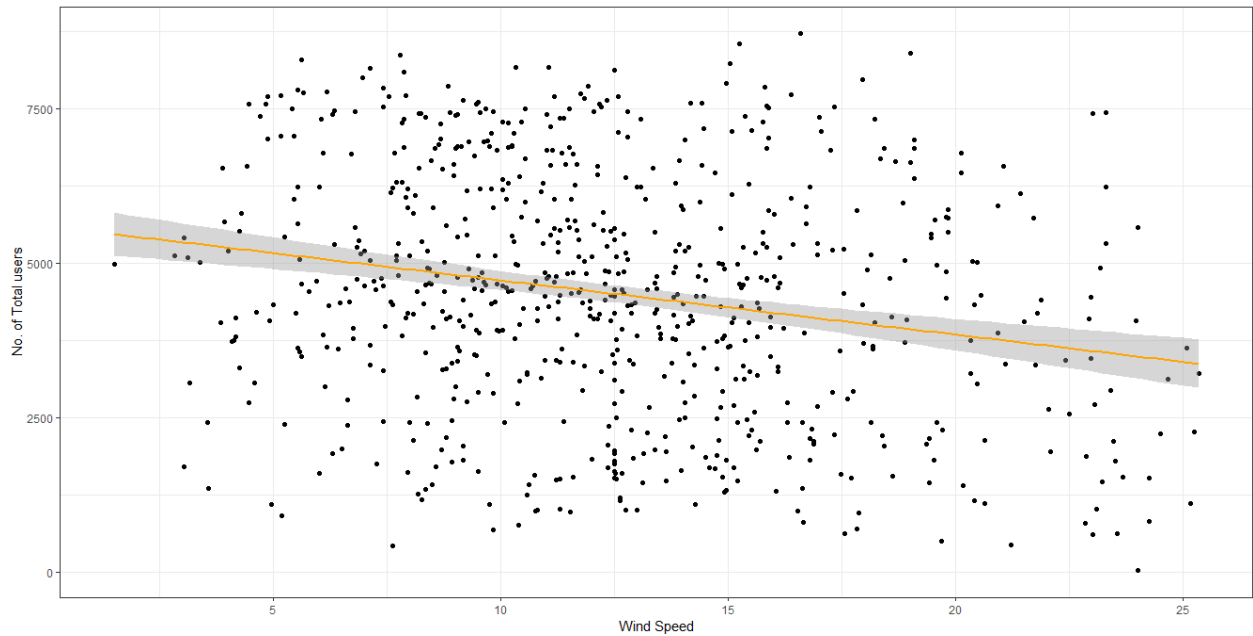


Figure 36:Scatterplot of windspeed v/s total counts

From the plots we can say that wind speed is also negatively related to the no. of users. People are less likely to rent a bike if the wind speed is too high.

- **Histogram plot of Casual users:**

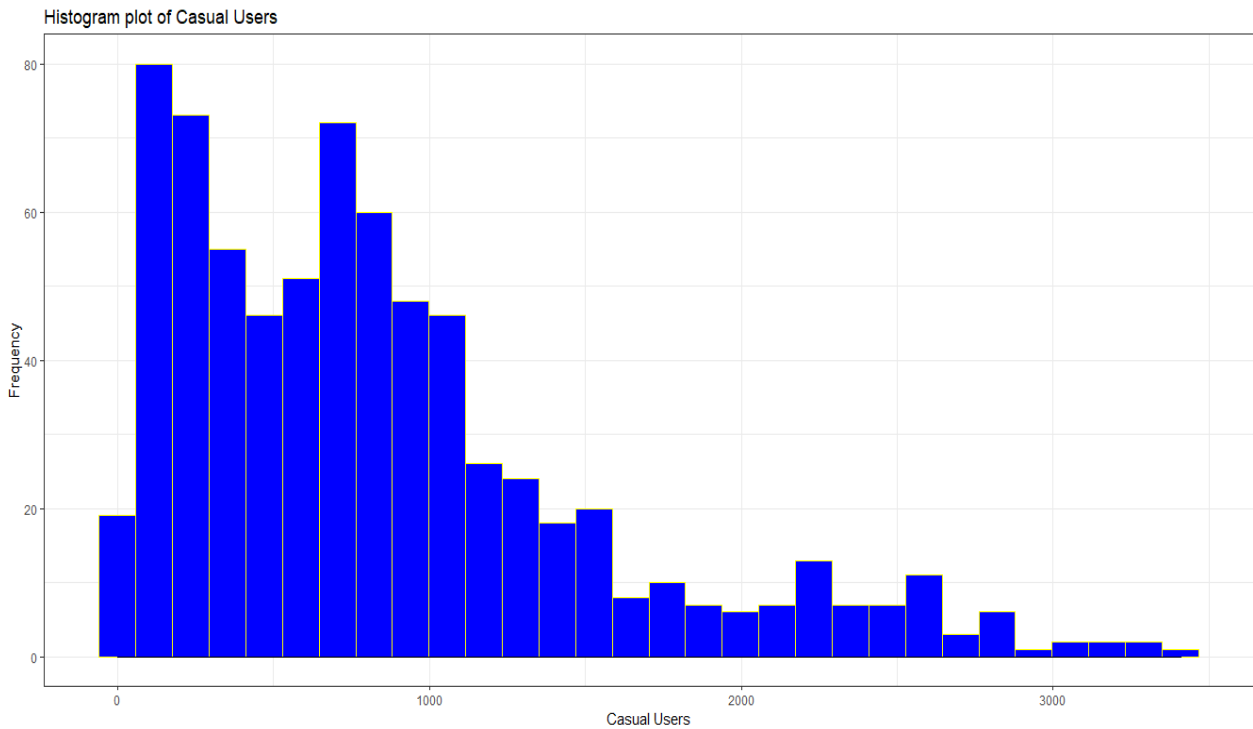


Figure 37:Histogram of Casual users

We can clearly see that the plot is right skewed.

- **Histogram plot of Logarithm of Casual users:**

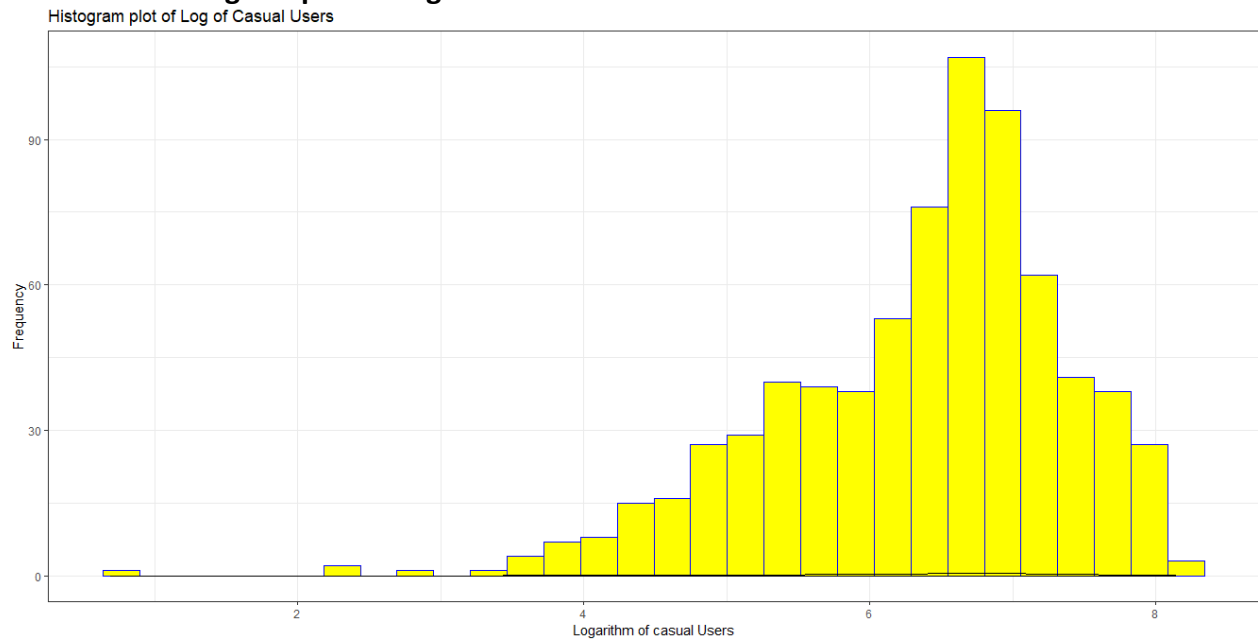


Figure 38:Histogram of Log of Casual users

We can see that log of casual users have a smooth, bell-shaped curve compared to that of casual users. So, we would use log of casual for further analysis.

- **Histogram plot of Registered users:**

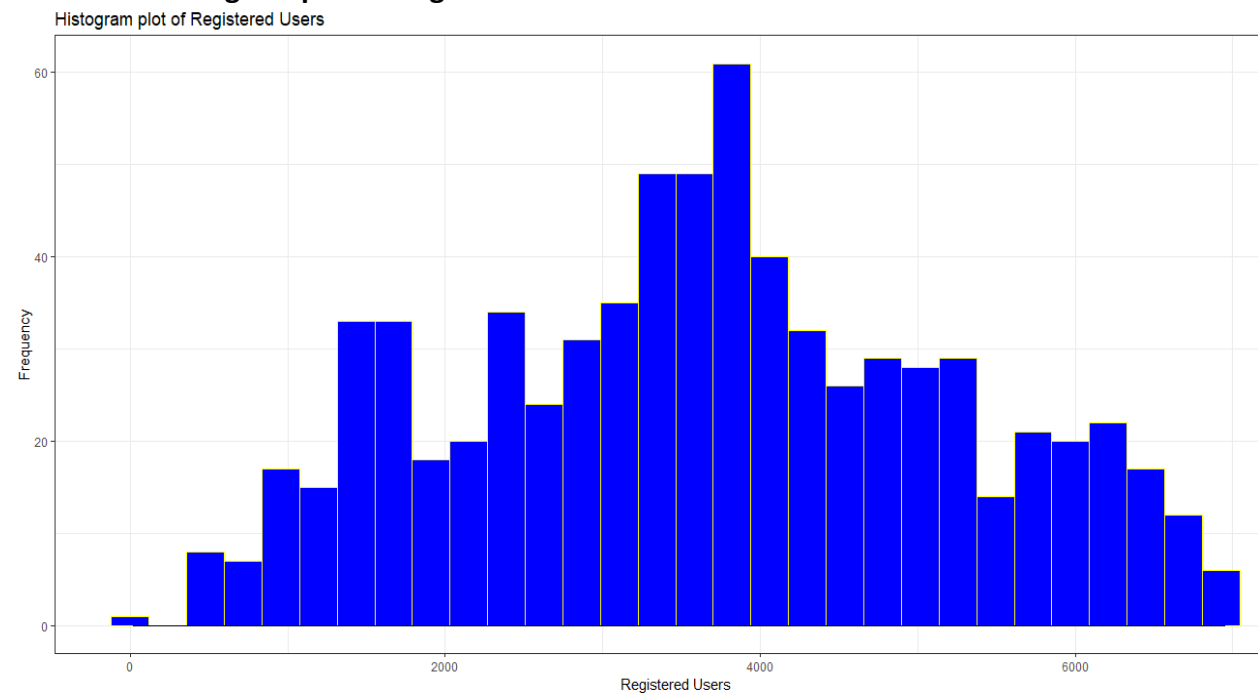


Figure 39:Histogram of Registered users

We can see that the distribution is somehow normally distributed.

- **Histogram plot of Logarithm of Registered users:**

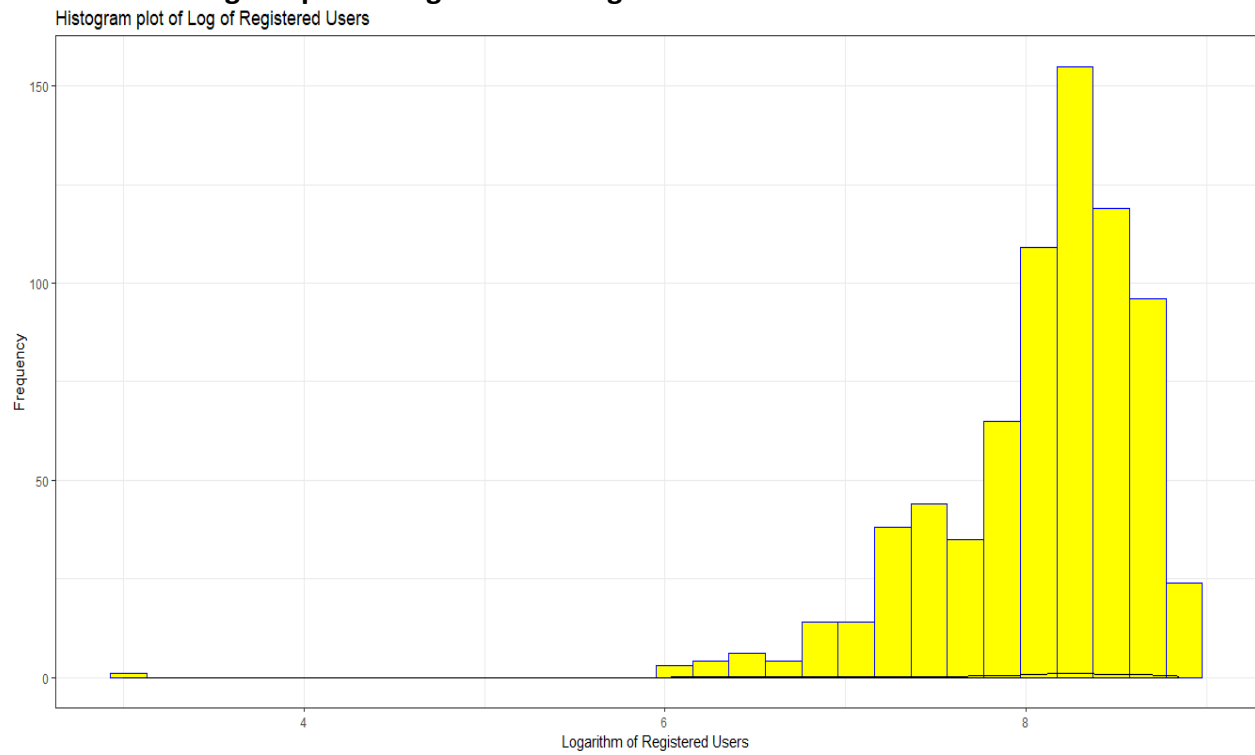


Figure 40:Histogram of Log of Registered users

In this case taking logarithm makes the histogram a bit left skewed. The original curve is much better to fit so we would retain the original values of registered users.

1.7.Feature Selection:

■ Correlation Analysis:

It is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables. This type of analysis is done to check the multi collinearity effect. If two or more independent variables are strongly correlated then only one of them is enough to predict the dependent variable so, others need to be removed. While a strong correlation between a dependent and independent variable is highly appreciable.

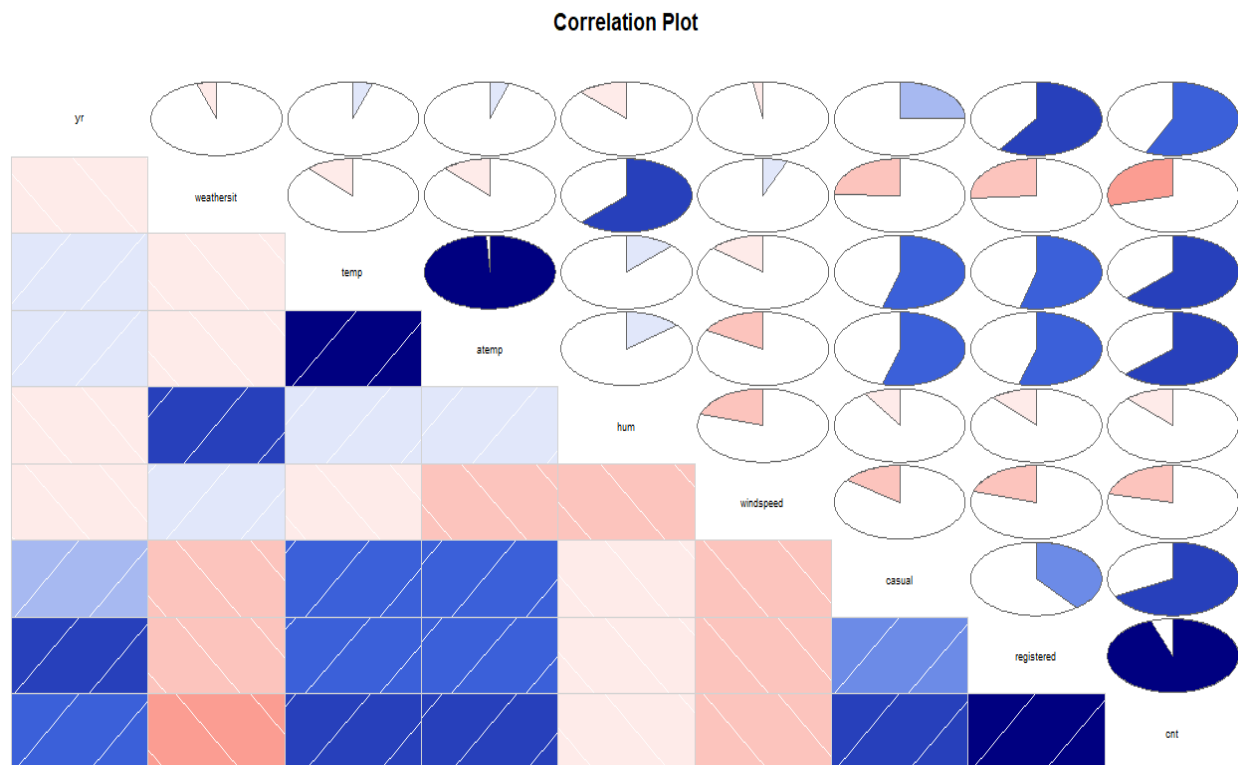


Figure 41:Correlation Plot

- From the correlation plot we can also see that variables "temp" and "atemp" are highly correlated. To decide which one to delete we will check the correlation coefficient of each of them with the dependent variable.

Pearson's product-moment correlation

```
data: data[, 8] and data[, 14]
t = 21.759, df = 729, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5814369 0.6695422
sample estimates:
cor
0.627494
```

Figure 42:Correlation value of "temp" and dependent variable

```

Pearson's product-moment correlation

data: data[, 9] and data[, 14]
t = 21.965, df = 729, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5853376 0.6727918
sample estimates:
      cor
0.6310657

```

Figure 43:Correlation value of “atemp” and dependent variable

Correlation between dependent variable and "atemp" (0.63) is slightly more than that with "temp" (0.62). So, we would delete "temp" from our dataset.

- From the correlation plot we can also see that variables "weathersit" and "hum" are highly correlated. To decide which one to delete we will check the correlation coefficient of each of them with the dependent variable.

```

Pearson's product-moment correlation

data: data[, 7] and data[, 14]
t = -8.4101, df = 729, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3620963 -0.2298340
sample estimates:
      cor
-0.2973912

```

Figure 44:Correlation value of “weathersit” and dependent variable

```

Pearson's product-moment correlation

data: data[, 9] and data[, 14]
t = -2.8657, df = 729, p-value = 0.004281
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.17670597 -0.03328593
sample estimates:
      cor
-0.1055448

```

Figure 45:Correlation value of “hum” and dependent variable

Correlation between dependent variable and "weathersit" (-0.30) is slightly more than that with "hum" (-0.10). So, we would delete "hum" from our dataset.

▪ **Chi-squared Test:**

A chi-squared test, also written as χ^2 test, is any statistical hypothesis test where the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true. The chi-squared test is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories.

- From the EDA we have expected that there can be a correlation between "season" and "mnth" variables as they almost give us the same information. So, checking Chi-square value between "season" and "mnth".

Pearson's Chi-squared test

```
data: table(data$season, data$mnth)
X-squared = 1765.1, df = 33, p-value < 2.2e-16
```

Figure 46: Chi-square Test between "season" and "mnth"

As p-value is less than 0.05, so collinearity exists between the two variables. So, we will go for ANOVA test later to check the better predictor and the other will be omitted.

- From the EDA we have seen that "holiday" and "workingday" variables almost gives us the same information in alternate ways. So, checking Chi-square value between "holiday" and "workingday".

Pearson's Chi-squared test with Yates' continuity correction

```
data: table(data$holiday, data$workingday)
X-squared = 43.598, df = 1, p-value = 4.033e-11
```

Figure 47: Chi-square Test between "holiday" and "workingday"

As p-value is less than 0.05, so collinearity exists between the two variables. So we will check the better predictor and the other will be omitted.

▪ **ANOVA (Analysis of Variance) Test:**

Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples. We can use ANOVA to prove/disprove if all the medication treatments were equally effective or not.

We have applied ANOVA test for our variables and found the following insights:

➤ “cnt” as dependent variable:

- Season :

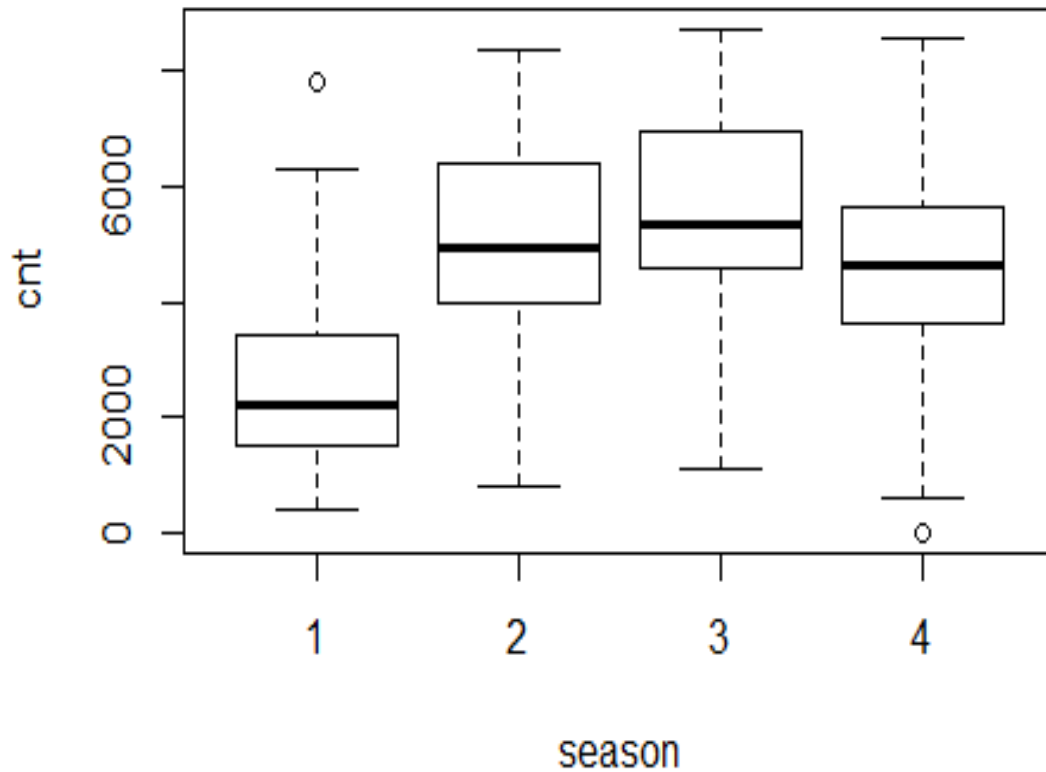


Figure 48:Box Plot of Season v/s Cnt

From the box plot it is evident that not all population means, across the groups, are equal. So, we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(cnt ~ season, data = data))
              Df    Sum Sq  Mean Sq F value Pr(>F)
season          3  9.506e+08 316865289   128.8 <2e-16 ***
Residuals     727  1.789e+09   2460715
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 49:ANOVA Test of Season

p-value less than 0.05 indicates the importance of the variable.

- **Month :**

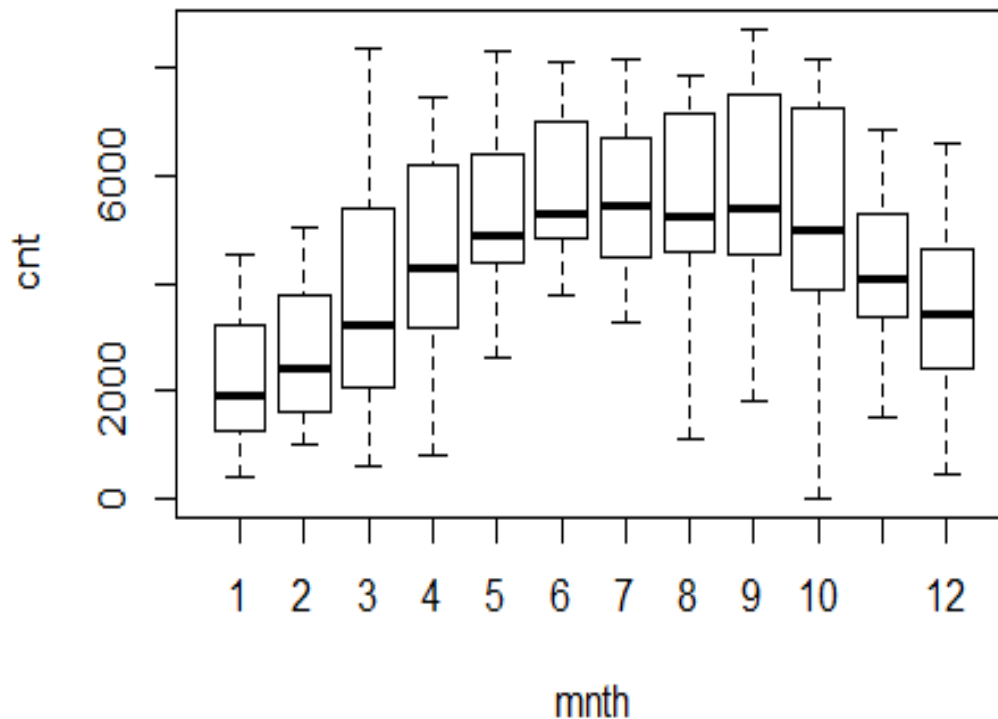


Figure 50:Box Plot of Month v/s Cnt

From the box plot it is evident that not all population means, across the groups, are equal. So we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(cnt ~ mnth, data = data))
              Df    Sum Sq Mean Sq F value Pr(>F)
mnth           11 1.070e+09  97290206   41.9 <2e-16 ***
Residuals     719 1.669e+09   2321757
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 51:ANOVA Test of Month

p-value less than 0.05 indicates the importance of the variable.

***NOTE : So we can see that both "season" and "month" have p-value less than 0.05. So, both of them are significant predictors. But we know because of multicollinearity we need to delete one of them. So, looking at the F-value we can see that F-value is more for "season" (129) than that of "mnth" (42). So "season" explains more variance of the dependent variable. Thus, we will delete "mnth" from our analysis.

- **Weekday :**

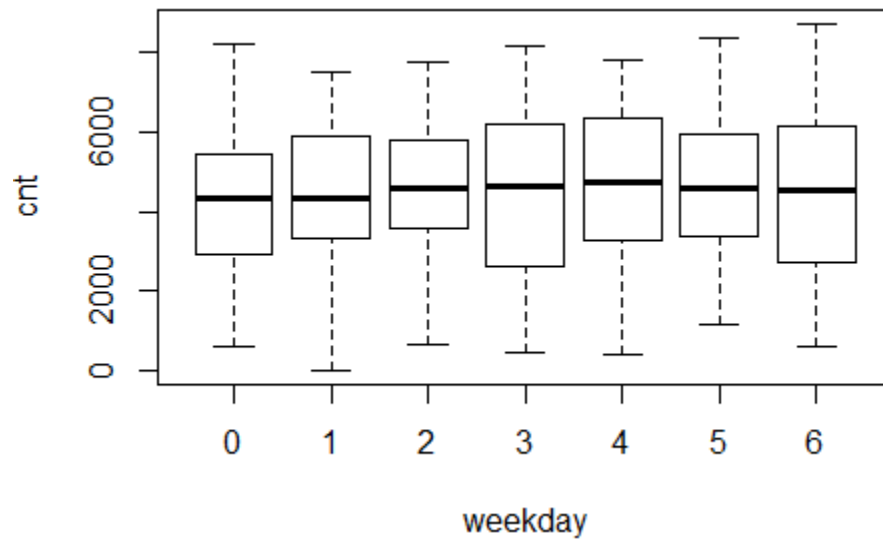


Figure 52:Box Plot of Weekday v/s Cnt

In this case we can see that all the population means across the groups are almost equal. So, we do not have enough evidence to reject the Null Hypothesis (H_0). Thus, this variable can't explain the variance of the dependent variable significantly.

```
> summary(aov(cnt ~ weekday, data = data))
              Df    Sum Sq Mean Sq F value Pr(>F)
weekday       6 1.766e+07 2943170   0.783  0.583
Residuals    724 2.722e+09 3759498
```

Figure 53:ANOVA Test of Weekday

P-value being greater than 0.05 indicates that the variable is not a significant predictor for "cnt".

- **Holiday :**

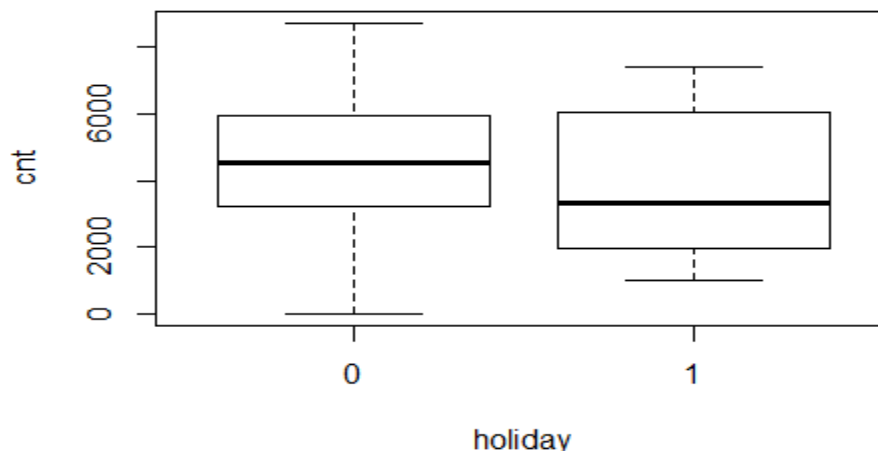


Figure 54:Box Plot of Holiday v/s Cnt

```
> summary(aov(cnt ~ holiday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
holiday  1 1.280e+07 12797494   3.421 0.0648 .
Residuals 729 2.727e+09 3740381
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 55:ANOVA Test of Holiday

In this case though the means across the groups are not equal but we can see that the p_value is slightly greater than 0.05. So, we can say that it is not a significant predictor for "cnt".

- **Workingday :**

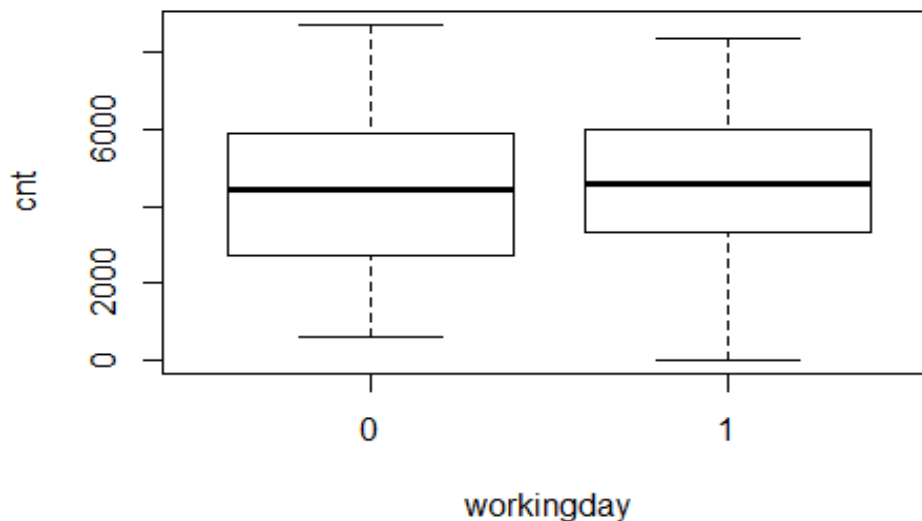


Figure 56:Box Plot of Workingday v/s Cnt

In this case we can see that all the population means across the groups are almost equal. So, we do not have enough evidence to reject the Null Hypothesis (H0). Thus, this variable can't explain the variance of the dependent variable significantly.

```
> summary(aov(cnt ~ workingday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
workingday  1 1.025e+07 10246038   2.737 0.0985 .
Residuals 729 2.729e+09 3743881
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 57:ANOVA Test of Workingday

P-value being greater than 0.05 indicates that the variable is not a significant predictor for "cnt".

***NOTE : So we have seen that that p_values of "weekday", "holiday", "workingday" is greater than 0.05. So, these variables does not influence the total count of users. But there is a chance that they have some effect on casual and registered users. As we have checked from the bar plots

earlier that count of casual and registered users changes on these variables. So, we will do ANOVA tests on these variables using casual and registered user as dependent variable.

➤ **“casual” as dependent variable:**

• **Weekday :**

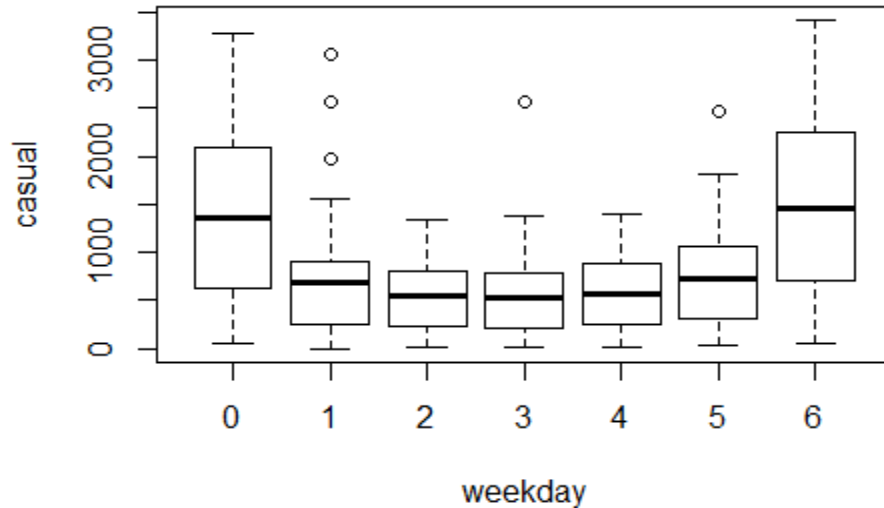


Figure 58:Box Plot of Weekday v/s Casual

From the box plot it is evident that not all population means, across the groups, are equal. So we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(casual ~ weekday, data = data))
              Df    Sum Sq Mean Sq F value Pr(>F)
weekday        6  94265703 15710950   45.52 <2e-16 ***
Residuals     724 249893120   345156
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 59:ANOVA Test of Weekday

p-value less than 0.05 indicates the importance of the variable.

• **Holiday :**

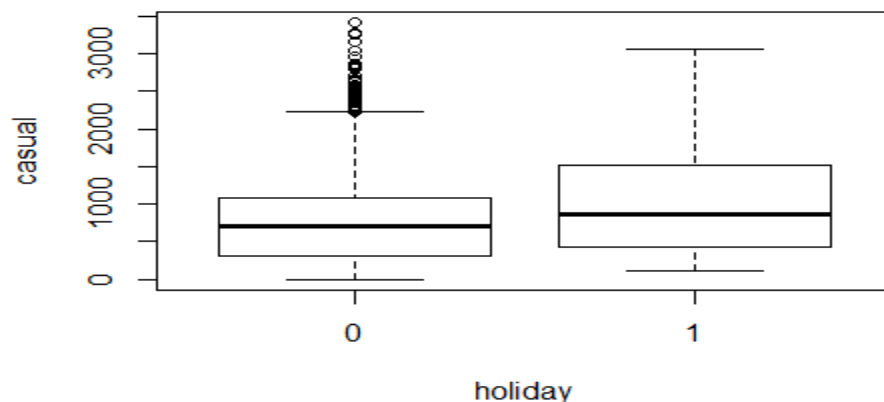


Figure 60:Box Plot of Holiday v/s Casual

From the boxplot we can see that there is not much difference between the means across the groups. So, we will check the p-value.

```
> summary(aov(casual ~ holiday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
holiday  1  1013785 1013785    2.154   0.143
Residuals 729 343145037  470706
```

Figure 61:ANOVA Test of Holiday

P-value being greater than 0.05 indicates that the variable is not a significant predictor for “cnt”.

- **Workingday :**

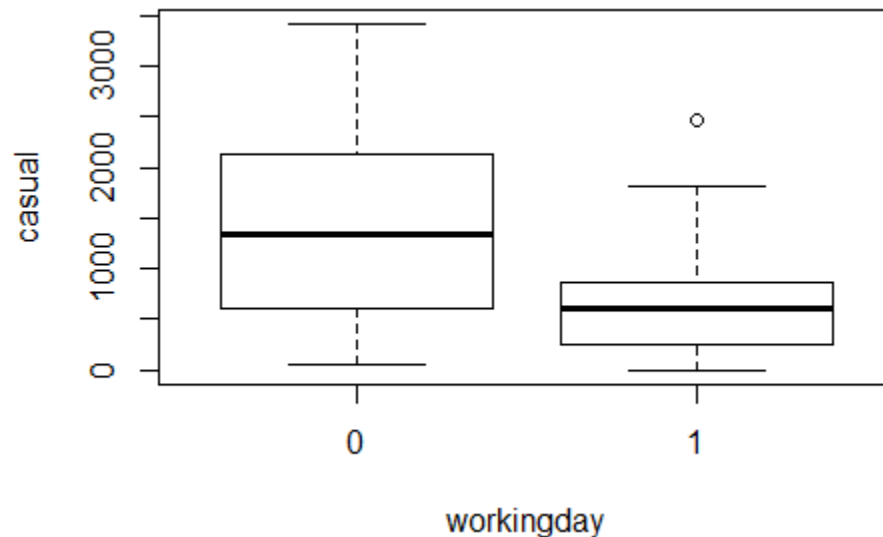


Figure 62:Box Plot of Workingday v/s Casual

From the box plot it is evident that not all population means, across the groups, are equal. So we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(casual ~ workingday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
workingday  1  92361829 92361829   267.4 <2e-16 ***
Residuals 729 251796993   345401
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 63:ANOVA Test of Workingday

p-value less than 0.05 indicates the importance of the variable.

So, only holiday has a p-value less than 0.05.

➤ “registered” as dependent variable:

- Weekday :

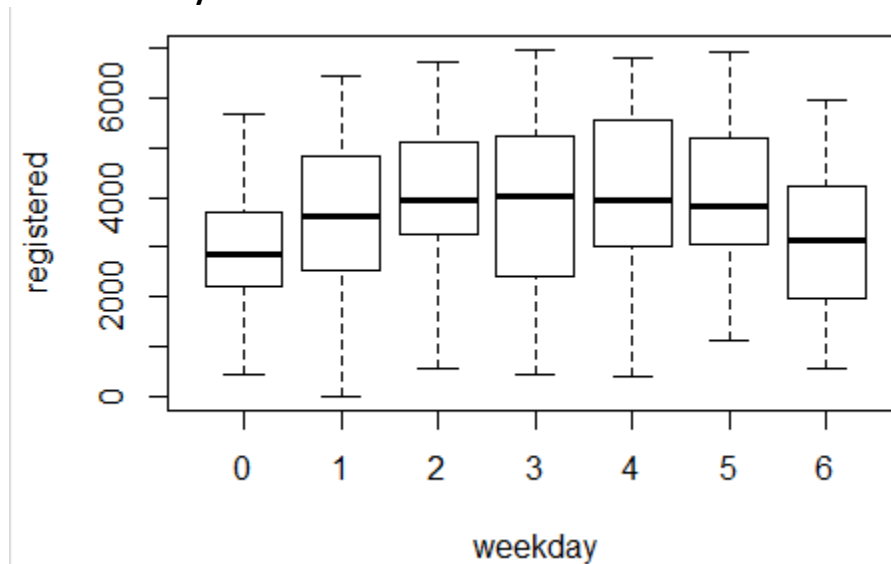


Figure 64:Box Plot of Weekday v/s Registered

From the box plot it is evident that not all population means, across the groups, are equal. So we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(registered ~ weekday, data = data))
      Df    Sum Sq Mean Sq F value    Pr(>F)
weekday    6 1.438e+08  23959857   10.62 2.52e-11 ***
Residuals 724 1.633e+09   2256012
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 65:ANOVA Test of Weekday

p-value less than 0.05 indicates the importance of the variable.

- Holiday :

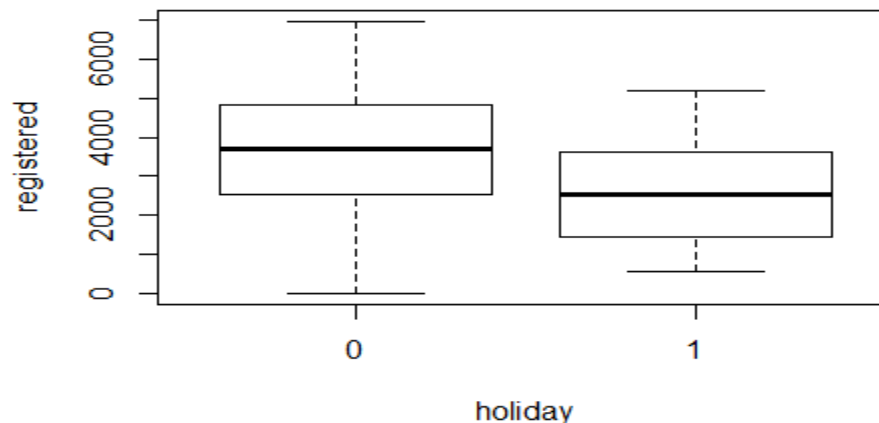


Figure 66:Box Plot of Holiday v/s Registered

From the box plot it is evident that not all population means, across the groups, are equal. So we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(registered ~ holiday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
holiday  1 2.102e+07 21015140   8.724 0.00324 **
Residuals 729 1.756e+09 2408912
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 67:ANOVA Test of Holiday

p-value less than 0.05 indicates the importance of the variable.

- **Workingday :**

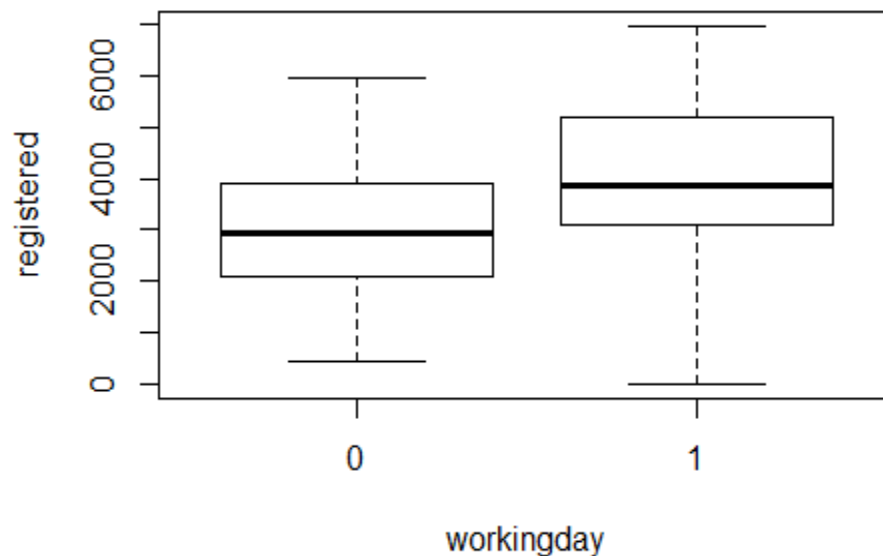


Figure 68:Box Plot of Workingday v/s Registered

From the box plot it is evident that not all population means, across the groups, are equal. So, we can reject the Null Hypothesis (H_0). Thus, the variable is significant in explaining the variance of dependent variable.

```
> summary(aov(registered ~ workingday, data = data))
      Df Sum Sq Mean Sq F value Pr(>F)
workingday  1 1.641e+08 164133237  74.18 <2e-16 ***
Residuals 729 1.613e+09 2212591
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 69:ANOVA Test of Workingday

p-value less than 0.05 indicates the importance of the variable.

***NOTE : All the 3 variables have p-value less than 0.05, so they are significant predictors. Moreover, we can see that the few variables are not correlated to the total count - "cnt" variable. But are correlated to "casual" and "registered" users. So, we would predict casual and registered users individually and then sum them up to find the total count of users.

1.8.Feature Scaling:

Feature scaling is a method used to standardize the range of independent variables or features of data. As most of the algorithm in machine learning calculates the distance between the independent variables so feature scaling is an important part of data preprocessing. If feature scaling is not applied then the model can be biased towards the variable having higher range of values.

There are 2 process to apply feature scaling:

- i) **Normalization** : All the data points are brought between the range 0 to 1
- ii) **Standardization** : Data point are scaled with respect to standard deviation. Where the mean is brought to 0 and s.d. to 1.

But standardization can only be applied on a normally distributed data. So we will check the distribution of our numerical variables to find which method to apply.

- **Atemp :**

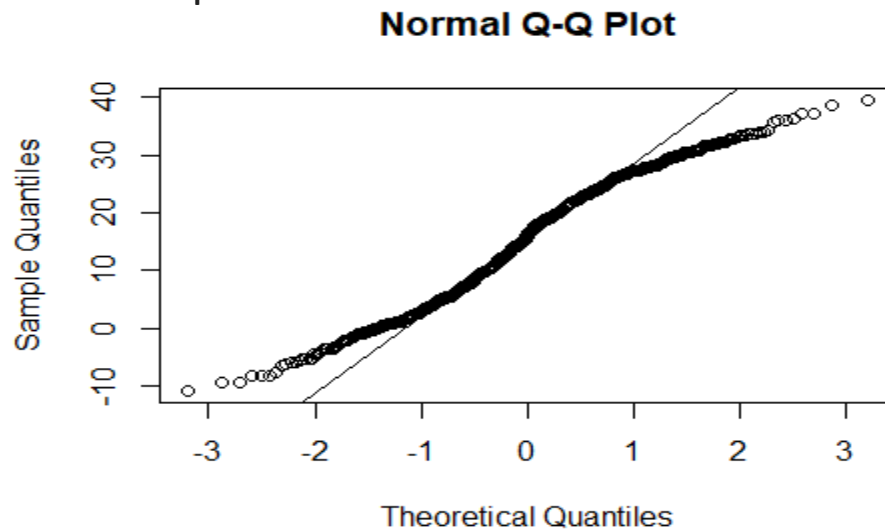


Figure 70:Q-Q Plot of atemp

The q plot deviates from the q line and have a curvy nature.

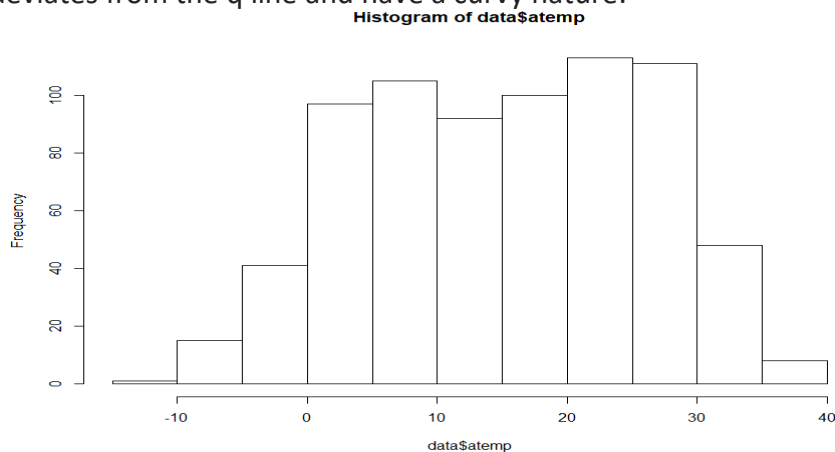


Figure 71:Histogram Plot of atemp

Histogram doesn't show a perfect normal distribution

- Windspeed :

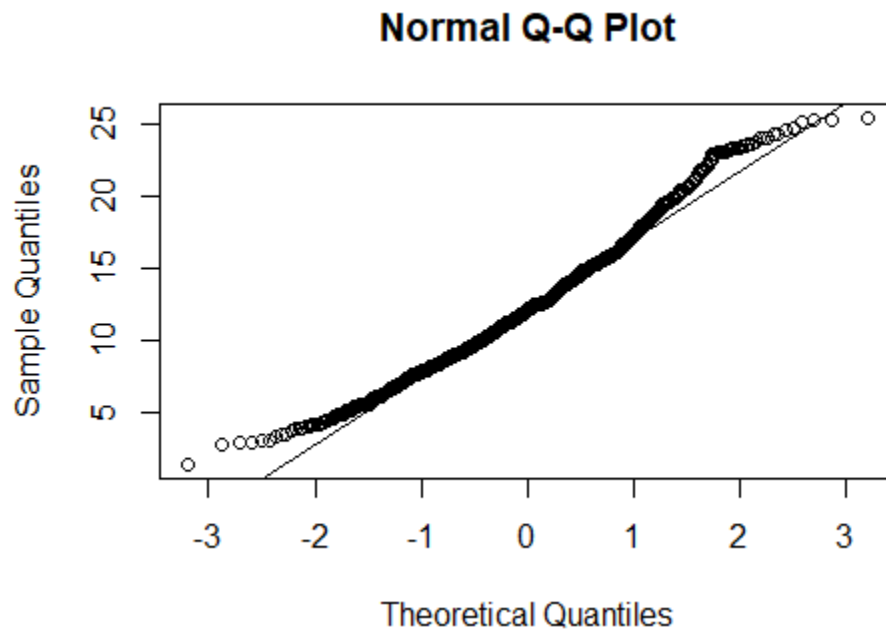


Figure 72:Q-Q Plot of windspeed

The q plot deviates from the q line at the extreme ends.

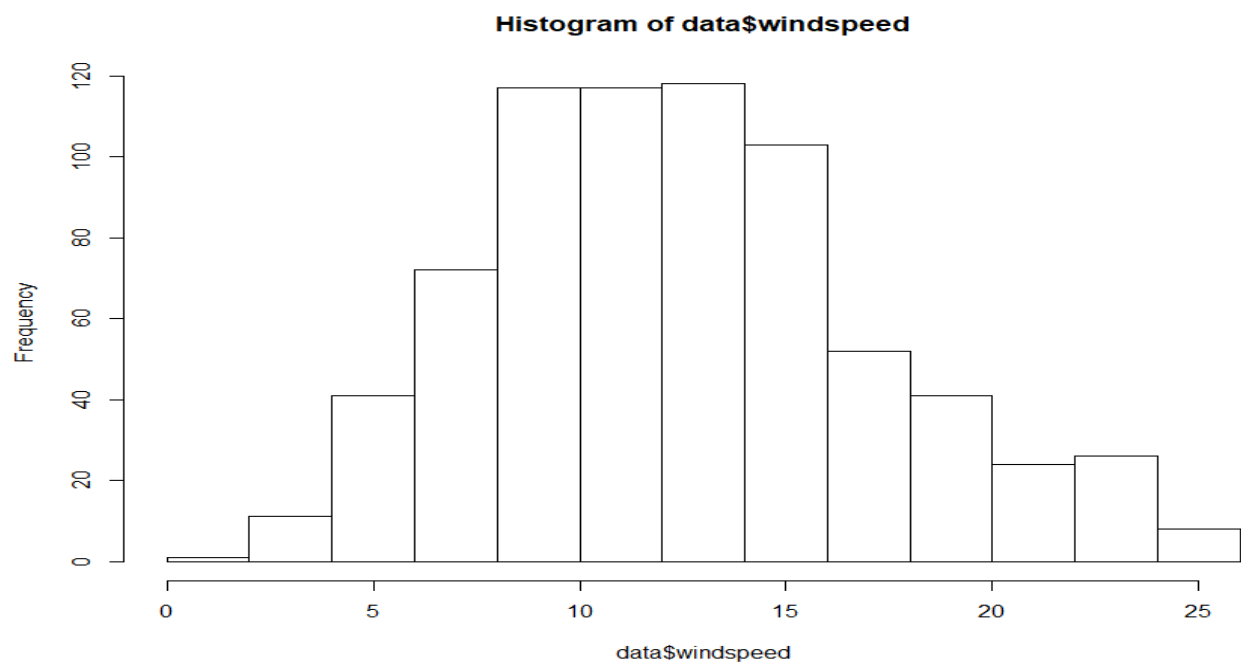


Figure 73:Histogram Plot of Windspeed

The histogram is very close to normal distribution.

So, normalization would be applied to scale as the all the variables are not normally distributed. But we have to apply different techniques to normalize the data based on the distribution of the variable.

- **Year** : Normalization is not required in case of year as there are only two values 0 and 1. If there are more than two values then normalization is required.
- **Weathersit** : It contains 4 categories. So it is normalized taking maximum value of 4 and minimum value of 1.
- **ATemp** : It is normalized taking maximum value of +50 and minimum value of -16 as per the problem statement.
- **Windspeed** : It is normalized by dividing each value by 67 as per the problem statement.

2. Model Building

As we have prepared and cleaned the data, now the next step involves feeding the data to model and train it. But to check the error metrics we need to divide the dataset into training and test sets. But on looking through the data we saw that there are very low instances of records for weathersit = 3 shown in figure below.

```
> table(data$weathersit)

 1    2    3
463 247  21
```

Figure 74: Table of weathersit

So, if we divide the records randomly then there might be a case that no records of such cases come to the test data. Thus, to fix that we have applied Stratified Sampling with “weathersit” as the strata variable. Moreover as the total number of data is small and we need enough data to train the model to increase accuracy so we have divided the data with 85% on the training set which is used to train the model and 15% as the test set which is used to predict on the independent variables of the test set. Then the predicted values are compared to the dependent variable of the test set to calculate the error metrics.

Here we have to predict the bike counts based on the independent variable. Count being a continuous numerical variable we have to apply different regression models on the data. We have applied several regression models from simple to complex separately on the casual and registered counts one by one and then at last we have compared the results.

2.1. Multiple Linear Regression:

Linear Regression is a statistical model. So when the model is trained it determines the best fit line based on lowest error. Then it saves the coefficients or weights related to each variable. This coefficient determines how much the variable is able to explain the variance of the dependent variable. And the variable having higher coefficient is of more importance.

At first multiple linear regression model is applied to the training set and we got the following result.

```
call:
lm(formula = log_casual ~ ., data = train_casual)

Residuals:
    Min       1Q   Median       3Q      Max
-4.7599 -0.2075  0.0264  0.2931  1.3449

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.99568    0.11192   44.638 < 2e-16 ***
season2       0.70251    0.07387    9.510 < 2e-16 ***
season3       0.33138    0.09592    3.455 0.000589 ***
season4       0.53159    0.06308    8.427 2.58e-16 ***
yr            0.39032    0.03974    9.823 < 2e-16 ***
holiday1      0.64483    0.12315    5.236 2.26e-07 ***
weekday1     -0.81509    0.07581   -10.752 < 2e-16 ***
weekday2     -0.89040    0.07361   -12.097 < 2e-16 ***
weekday3     -0.92778    0.07469   -12.422 < 2e-16 ***
weekday4     -0.85927    0.07403   -11.607 < 2e-16 ***
weekday5     -0.56900    0.07484    -7.603 1.10e-13 ***
weekday6      0.09427    0.07349    1.283 0.200098
workingday1    NA         NA         NA      NA
weathersit    -1.29144    0.11056   -11.681 < 2e-16 ***
atemp         3.48703    0.21720   16.055 < 2e-16 ***
windspeed     -0.89517    0.28901    -3.097 0.002043 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4924 on 607 degrees of freedom
Multiple R-squared:  0.7735,    Adjusted R-squared:  0.7682
F-statistic: 148 on 14 and 607 DF,  p-value: < 2.2e-16
```

Figure 75: Linear Regression model

From the coefficient values we can see that one coefficient is not defined because of singularity. So, there might be a presence of multicollinearity. To make our assumption more concrete we have checked the VIF value of the model.

- **VIF (Variance Inflation Factor):**

In statistics, the variance inflation factor (VIF) is the ratio of variance in a model with multiple terms, divided by the variance of a model with one term alone. It is a measure of multi-collinearity in a regression design matrix. The formula for determining VIF is:

$$VIF_i = \frac{1}{1 - R_i^2}$$

Where R^2 is the coefficient of determination.

It provides an index that measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient is increased because of

collinearity. The ideal value of VIF should be 1. If the value is between 5 to 10 then there is presence of multi-collinearity among the variables. But if the value exceeds 10 then there is very high multi-collinearity and it should be taken care of.

The values of VIF for our model gives the following result:

```
> vif(LR_casual1)
Error in vif.default(LR_casual1) :
  there are aliased coefficients in the model
```

Figure 76:VIF of model

As the VIF result shows presence of aliased coefficients and we know that there is multicollinearity between "holiday" and "workingday". So, by trial and error method we saw eliminating "workingday" gives better accuracy. Thus, we eliminated "workingday" from the data.

- **"Casual" as dependent variable :**

So applying linear regression on the casual training set gives us the following result :

```
> summary(LR_casual2)

Call:
lm(formula = log_casual ~ ., data = train_casual)

Residuals:
    Min       1Q   Median       3Q      Max
-4.7599 -0.2075  0.0264  0.2931  1.3449

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.99568    0.11192  44.638 < 2e-16 ***
season2      0.70251    0.07387   9.510 < 2e-16 ***
season3      0.33138    0.09592   3.455 0.000589 ***
season4      0.53159    0.06308   8.427 2.58e-16 ***
yr           0.39032    0.03974   9.823 < 2e-16 ***
holiday1     0.64483    0.12315   5.236 2.26e-07 ***
weekday1    -0.81509    0.07581 -10.752 < 2e-16 ***
weekday2    -0.89040    0.07361 -12.097 < 2e-16 ***
weekday3    -0.92778    0.07469 -12.422 < 2e-16 ***
weekday4    -0.85927    0.07403 -11.607 < 2e-16 ***
weekday5    -0.56900    0.07484  -7.603 1.10e-13 ***
weekday6     0.09427    0.07349   1.283 0.200098
weathersit   -1.29144    0.11056 -11.681 < 2e-16 ***
atemp       3.48703    0.21720  16.055 < 2e-16 ***
windspeed   -0.89517    0.28901  -3.097 0.002043 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4924 on 607 degrees of freedom
Multiple R-squared:  0.7735,    Adjusted R-squared:  0.7682
F-statistic: 148 on 14 and 607 DF,  p-value: < 2.2e-16
```

Figure 77:Linear Regression model for casual counts

As you can see the Adjusted R-squared value, we can almost explain 77% of the data using our multiple linear regression model. Moreover looking at the F-statistic and combined p-value we can reject the Null Hypothesis that target variable does not depend on any of the predictor variables.

Now on checking the VIF value of the model we get the following result:

```
> vif(LR_casual2)
      GVIF Df GVIF^(1/(2*Df))
season  3.307976 3      1.220657
yr       1.012639 1      1.006300
holiday  1.093408 1      1.045662
weekday  1.115194 6      1.009127
weathersit 1.037111 1      1.018387
atemp    3.168974 1      1.780161
windspeed 1.075490 1      1.037058
```

Figure 78:VIF for casual counts

From the above table we can see that the VIF values are close to one which ensures that multi-collinearity doesn't exist among the variables. Thus we have considered all the variables for our further models.

- **“Registered” as dependent variable :**

So applying linear regression on the registered training set gives us the following result :

```
> summary(LR_reg2)

Call:
lm(formula = registered ~ ., data = train_reg)

Residuals:
    Min       1Q   Median       3Q      Max
-3912.8  -321.5    77.2   407.3  1530.1

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   101.04    153.45   0.658  0.510493
season2        845.28    101.29   8.345  4.80e-16 ***
season3        827.90    131.51   6.295  5.89e-10 ***
season4       1315.82     86.50  15.213 < 2e-16 ***
yr           1769.29     54.48  32.474 < 2e-16 ***
holiday1     -1187.79    168.85  -7.035  5.41e-12 ***
weekday1       944.70    103.94   9.089 < 2e-16 ***
weekday2      1121.40    100.92  11.111 < 2e-16 ***
weekday3      1155.05    102.40  11.280 < 2e-16 ***
weekday4      1173.20    101.50  11.558 < 2e-16 ***
weekday5      1110.70    102.61  10.825 < 2e-16 ***
weekday6       290.42    100.77   2.882  0.004090 **
weathersit     -1794.08    151.59 -11.835 < 2e-16 ***
atemp         3465.72    297.80  11.638 < 2e-16 ***
windspeed     -1465.12    396.26  -3.697  0.000238 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 675.1 on 607 degrees of freedom
Multiple R-squared:  0.8169,    Adjusted R-squared:  0.8127
F-statistic: 193.5 on 14 and 607 DF,  p-value: < 2.2e-16
```

Figure 79:Linear Regression model for registered counts

As you can see the Adjusted R-squared value, we can almost explain 81% of the data using our multiple linear regression model. Moreover, looking at the F-statistic and combined p-value we can reject the Null Hypothesis that target variable does not depend on any of the predictor variables.

Now on checking the VIF value of the model we get the following result:

```
> vif(LR_reg2)
      GVIF Df GVIF^(1/(2*Df))
season  3.307976  3    1.220657
yr       1.012639  1    1.006300
holiday  1.093408  1    1.045662
weekday  1.115194  6    1.009127
weathersit 1.037111  1    1.018387
atemp    3.168974  1    1.780161
windspeed 1.075490  1    1.037058
```

Figure 80:VIF for casual counts

From the above table we can see that the VIF values are less than 5 which ensures that multi-collinearity doesn't exist among the variables. Thus, we have considered all the variables for our further models.

2.2. Decision Tree Regression:

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. The decision tree algorithm is mainly based on Information Gain. It will select that parameter as the parent node which will have more information gain value.

Information gain is the difference between information entropy of the system before splitting and information entropy of the system after splitting. Information entropy is the average rate at which information is produced.

$$S = - \sum_i P_i \log P_i$$

Where P_i is the probability of occurrence of dependent variable.

- **“Casual” as dependent variable :**

Now we have applied decision tree regression model to predict our casual count target variable. The visual representation of the decision tree is shown below:

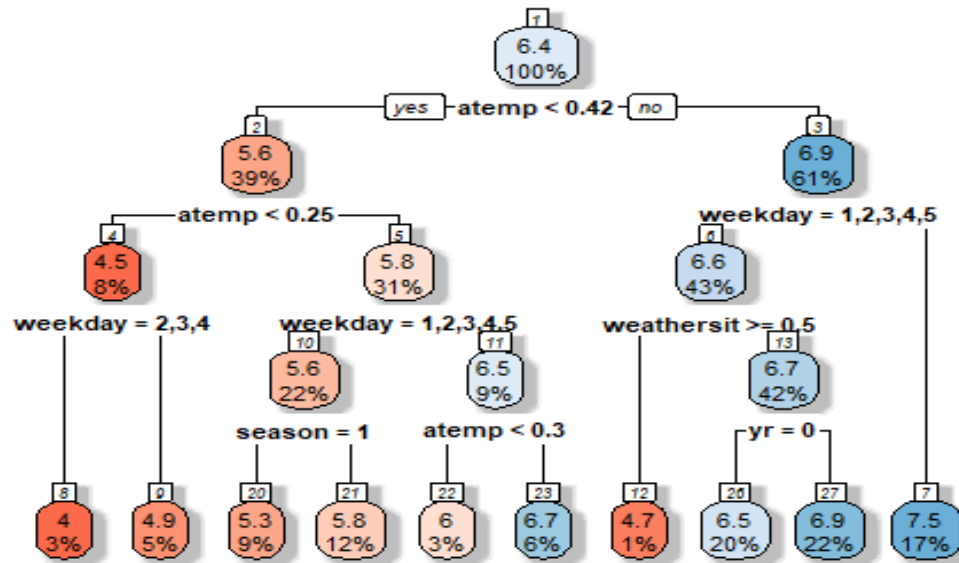


Figure 81:Decision Tree for Casual Counts

From the above image we get the set of rules based on which the decision tree is built.

- **“Registered” as dependent variable :**

Now we have applied decision tree regression model to predict our registered count target variable. The visual representation of the decision tree is shown below:

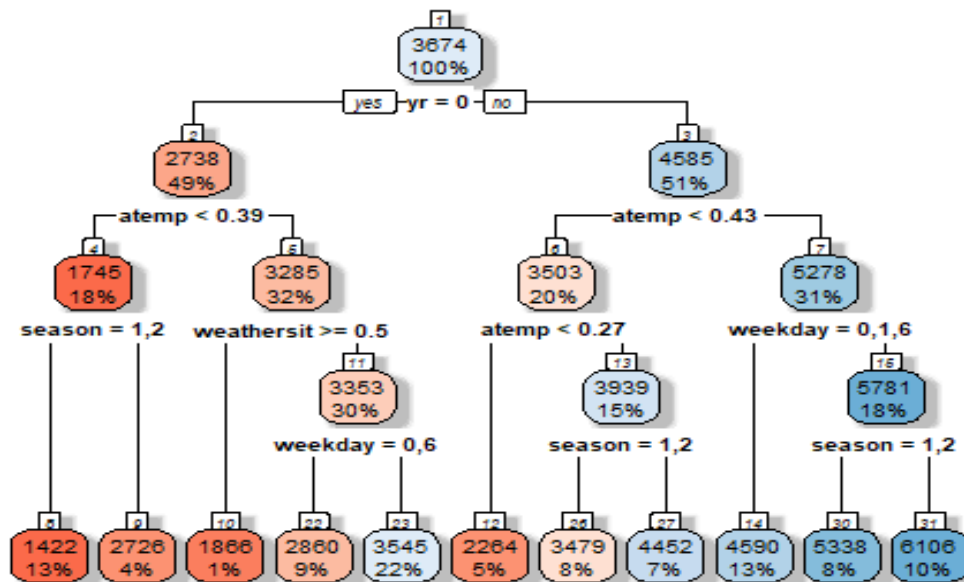


Figure 82:Decision Tree for Casual Counts

From the above image we get the set of rules based on which the decision tree is built.

2.3. Random Forest Regression:

Random Forest is an ensemble that consists of many decision trees. To build each decision tree we use the different portion of the whole data. This reduces error and increases accuracy. The idea behind the Random Forest is that a single decision tree may not be able to explain the variance of the whole data set, so, we use many trees to extract as much variance as possible. The Random Forest algorithm uses the Gini Index to select the parent nodes.

$$\text{Gini} = 1 - \sum (P_i)^2$$

Gini Index measures the amount of impurity of the data. It selects that variable whose Gini Index is lowest. Then for each node it will randomly select few variables(m) to build the first tree and this must be very much less than the number of variables(M) and may be based on the formula ($m = \text{sqrt.}(M)$).

Then the tree takes the bootstrap sample, i.e.- it randomly selects 67% of the observation for training and the remaining 33% for testing. This is called 'Out of Bag' sample method. Then it applies the CART algorithm on the training data, to predict the class of the test data and thus the error of the tree is estimated comparing the actual and the predicted values. Then whatever observation is misclassified is fed to the next decision tree. Then it will keep on splitting until it finds the leaf node based on the error rate. It will build trees until the error no longer decreases. When the same error value will repeat it will stop growing the trees.

We have applied the random forest algorithm to the model without mentioning the number of trees so that it can grow until it finds the lowest error value.

2.4. SVR (Support Vector Regression):

SVR is a type of model in which we try to set the error within a certain threshold while in linear regression we try to minimize the error rate. Support Vector Machine can be applied not only to classification problems but also to the case of regression. Still it contains all the main features that characterize maximum margin algorithm: a non-linear function is learned by linear learning machine mapping into high dimensional kernel induced feature space. The capacity of the system is controlled by parameters that do not depend on the dimensionality of feature space. In the same way as with classification approach there is motivation to seek and optimize the generalization bounds given for regression. They relied on defining the loss function that ignores errors, which are situated within the certain distance of the true value. This type of function is often called – epsilon intensive – loss function.

We have applied SVR on our training set and predicted the values on test set.

2.5. KNN (K-Nearest Neighbours):

The algorithm uses 'feature similarity' to predict values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set. The first step is to calculate the distance between the new point and

each training point. There are various methods for calculating this distance, of which the most commonly known methods are – Euclidian, Manhattan and Hamming distance. Then the value of k is user defined which denotes the number of nearest neighbours that should be considered when a new point comes.

It is a lazy learning method, where the function is only approximated locally; i.e.- the model doesn't save the value of training data like other algorithms instead the value is determined instantly when a new point comes.

One of the challenging task for KNN algorithm is to find the value of K. Here we have imputed different values of 'K' for the algorithm and computed their respective rmse values.

- **“Casual” as dependent variable :**

The table is ordered by descending order of RMSE value.

K	RMSE
15	365.5034
16	365.6341
17	367.5691
14	368.3206
19	368.4748
6	368.6377
18	368.7834
8	370.8990
9	371.7356
20	372.0367
7	372.0957
5	375.6007
10	375.9714
13	377.6907
4	380.0383
2	383.1117
12	383.7428
11	388.8809
3	411.4100
1	444.5874

Figure 83:K-values for casual

From the above table we can see that the value of rmse is lowest for K-value equals to 15. But if we take a very low k value then the model over fits the training data, which results in high error rate on the validation side. On the other hand for a high value of K the model performs poorly on both train and test data sets. One way to find a descent value of K is to plot the rmse values with their respective K values.

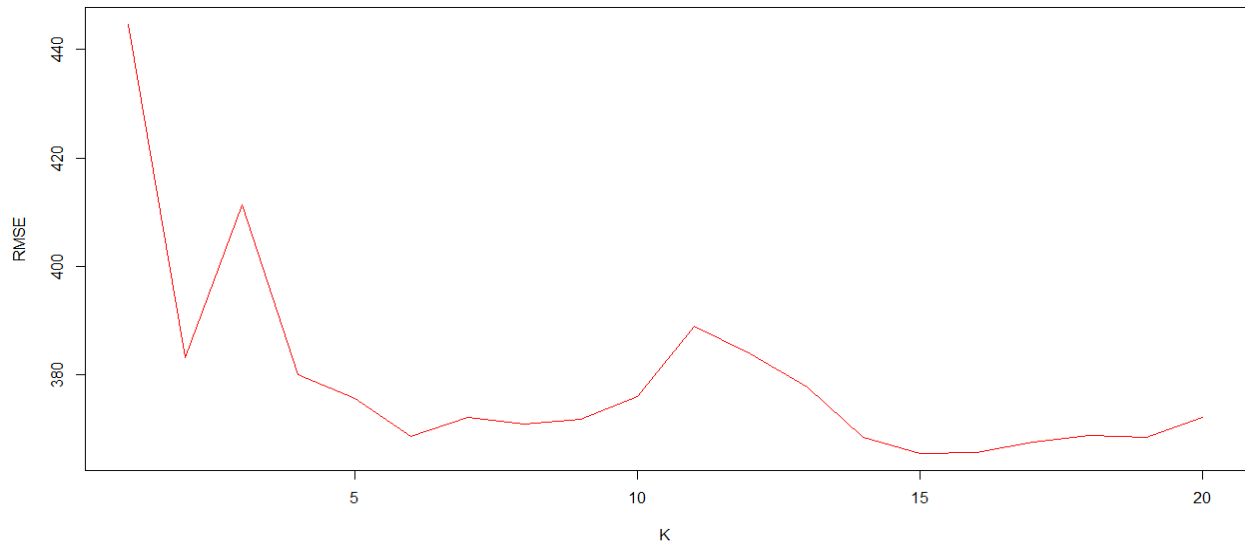


Figure 84:Elbow Curve for casual

The above curve is also known as Elbow Curve. From the plot we can see that the elbow appears at a value of $k = 6$. But here as we want lower error for our model so we have imputed $k = 15$ for our model to predict the test cases.

- **“Registered” as dependent variable :**

The table is ordered by descending order of RMSE value.

K	RMSE
14	680.7707
18	682.3335
17	683.6310
15	684.5108
16	685.6014
13	687.3011
19	690.5072
20	697.9306
12	717.9589
5	735.6496
3	739.0269
4	740.9235
11	747.4335
6	748.2817
8	757.7730
10	761.1702
2	762.6878
9	767.0939
7	770.8213
1	836.4704

Figure 85:K-values for registered

From the above table we can see that the value of rmse is lowest for K-value equals to 15. But if we take a very low k value then the model over fits the training data, which results in high error rate on the validation side. On the other hand for a high value of K the model performs poorly on both train and test data sets. One way to find a descent value of K is to plot the rmse values with their respective K values.

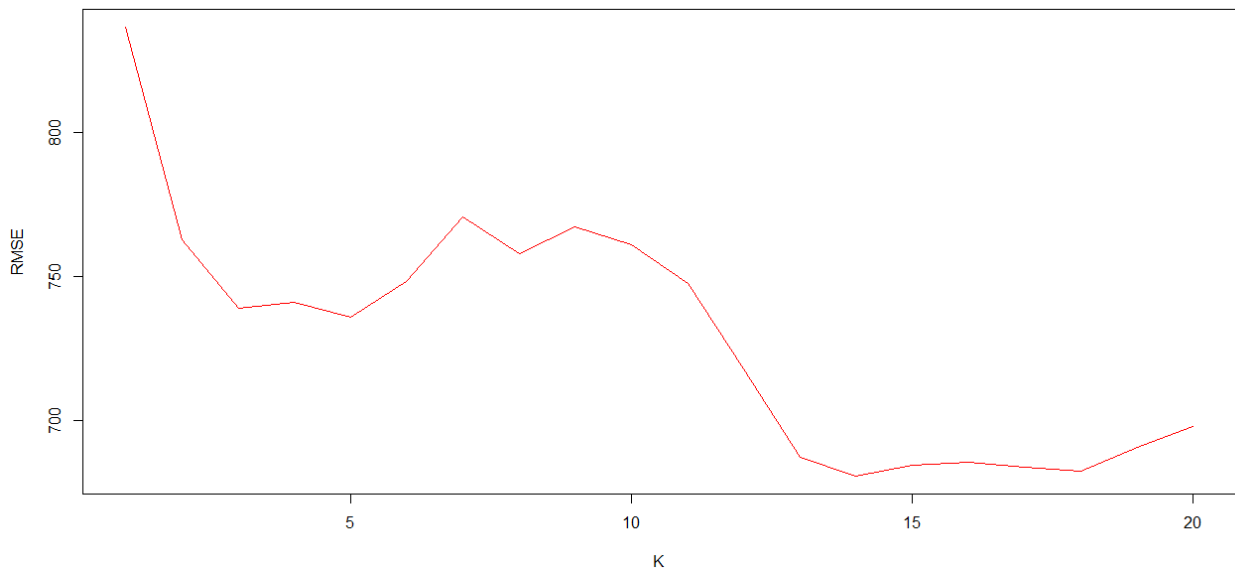


Figure 86:Elbow Curve for registered

The above curve is also known as Elbow Curve. From the plot we can see that the elbow appears at a value of $k = 3$. But here as we want lower error for our model so we have imputed $k = 14$ for our model to predict the test cases.

Now of the different regression models we will choose the best model by comparing error metrics.

Chapter 3

CONCLUSION

1. Model Evaluation:

Now that we have a few models for predicting the target variable and we need to decide which one to choose. There are several methods by which we can compare the models. As the dependent variable is a continuous regression model so we have compared the models based on different error metrics.

a) Mean Absolute Percentage Error (MAPE):

It is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation, also used as a loss function for regression problems in machine learning. It usually expresses accuracy as a percentage, and is defined by the formula:

$$M = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|,$$

Where, A_t is the actual value and F_t is the predicted value. The difference between A_t and F_t is divided by the actual value A_t again. The absolute value in this calculation is summed for every predicted point and is divided by the number of total points n . Multiplying by 100% makes it a percentage error.

Lower MAPE indicates better model.

b) Root Mean Squared Error (RMSE):

The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data—how close the observed data points are to the model's predicted values. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable.

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

Where, X_{obs} is observed values and X_{model} is modelled values at time/place i .

Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

2. Model Selection:

Thus, we have calculated the MAPE and RMSE values for prediction of different models using the predicted values of respective models and the dependent variable of the test set. Then we have saved the data in a table for comparison:

- “Casual” as dependent variable :

	LR_casual_EM	DT_casual_EM	RF_casual_EM	SVR_casual_EM	KNN_casual_EM
MAPE	0.3670973	0.5782545	0.3903357	0.3423904	0.4814765
RMSE	418.4002112	360.7311799	320.0906718	288.3658665	365.5033837

Figure 87:Error Metrics for Casual Counts in R

Index	LR	DT	RF	SVR	KNN
mape	59.5604	68.2909	45.6478	52.2716	55.1105
rmse	647.632	454.982	359.871	393.866	357.787

Figure 88:Error Metrics for Casual Counts in Python

From the above table we can see that the SVR model in R gives the lowest value of both RMSE and MAPE. So, we would choose the SVR model as the best model in R for prediction of casual counts.

While in case of Python we can see that KNN model gives the lowest of both RMSE and MAPE. So, we would choose the KNN model as the best model in Python for prediction of casual counts.

- “Registered” as dependent variable :

	LR_reg_EM	DT_reg_EM	RF_reg_EM	SVR_reg_EM	KNN_reg_EM
MAPE	0.1829264	0.2573963	0.1913548	0.1486539	0.2279697
RMSE	570.4894754	847.7901853	556.1766443	496.8236010	680.7707480

Figure 89:Error Metrics for Registered Counts in R

Index	LR	DT	RF	SVR	KNN
mape	22.4031	19.8152	17.1401	53.7342	19.8148
rmse	779.823	721.763	559.692	1565.04	663.184

Figure 90:Error Metrics for Registered Counts in Python

From the above table we can see that the SVR model in R gives the lowest value of both RMSE and MAPE. So we would choose the SVR model as the best model in R for prediction of registered counts.

While in case of Python we can see that Random Forest model gives the lowest of both RMSE and MAPE. So, we would choose Random Forest model as the best model in Python for prediction of registered counts.

Then we have added the casual and registered count of and rounded it off as the count can not be a fraction. Then we have compared the value with the total counts of the test data to get the following result :

```
cnt_EM
      mape      rmse
0.1546264 661.7866282
```

Figure 91:Error Metrics of Total Counts in R

mape	19.5546
rmse	760.85

Figure 92:Error Metrics of Total Counts in Python

3. Preparation of Test Cases:

As there is no test cases given to us so we made a sample input to check our models output. The process of preparing the test case is discussed here.

- **Categorical and Distinct Numerical Variables:**

We have season (4 groups), holiday (2 groups) and weekday (7 groups) as categorical variables. While yr (2 distinct values) and weathersit (4 distinct values) are the numerical variables having distinct values.

For all these variables we have taken all the possible combinations of values. But as a holiday and weekend is not a possible case so for weekday equal to 0 and 6 we have not considered cases of holiday equals to 1.

- **Continuous Numerical Variables:**

We have 2 continuous variables – atemp and windspeed. As we know in real life cases the temperature and windspeed is very much dependent on the season. So we have calculated the maximum and minimum values of the feeled temperature and windspeed grouping as season. Then we have generated random values in between those numbers to generate the values in our test case.

4. Hyper Parameter Tuning and Model Training:

Now we applied feature scaling on our data to normalize the values between 0 to 1. Then we applied parameter tuning to find the best values for our hyper parameter on SVM model with RMSE as search metric and trained our model with the whole dataset given to us. The result of the grid search is shown below.

- **Casual as dependent variable:**

Support Vector Machines with Radial Basis Function Kernel

731 samples
7 predictor

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 731, 731, 731, 731, 731, 731, ...

Resampling results across tuning parameters:

C	RMSE	Rsquared	MAE
0.25	0.4784683	0.7886044	0.3234849
0.50	0.4576257	0.8002022	0.3079989
1.00	0.4509207	0.8031218	0.3039632

Tuning parameter 'sigma' was held constant at a value of 0.04419207

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were sigma = 0.04419207 and C = 1.

Figure 93:Parameter tuning for casual count for SVR model in R

best_parameter_casual - Dictionary (3 elements)

Key	Type	Size	
algorithm	str	1	auto
n_neighbors	int	1	6
p	int	1	2

Figure 94:Parameter tuning for casual count for KNN model in Python

- Registered as dependent variable:

Support Vector Machines with Radial Basis Function Kernel

731 samples
7 predictor

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 731, 731, 731, 731, 731, 731, ...

Resampling results across tuning parameters:

C	RMSE	Rsquared	MAE
0.25	686.0787	0.8164866	503.0187
0.50	644.5442	0.8334560	466.0478
1.00	619.7648	0.8444759	443.1269

Tuning parameter 'sigma' was held constant at a value of 0.04800068

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were sigma = 0.04800068 and C = 1.

Figure 95:Parameter tuning for registered count for SVR model in R

best_parameter_reg - Dictionary (4 elements)

Key	Type	Size	
bootstrap	str	1	False
n_estimators	int	1	110
oob_score	str	1	False
warm_start	str	1	False

Figure 96:Parameter tuning for registered count for RF model in Python

Then we predicted the output with the best model for casual and registered counts individually. Then these two values are summed up to find the total count variable.

5. Validation of test case output:

In this part we would cross check our output values of test data with the hypothesis built during the EDA phase with the help of several plots and then draw our conclusion out of it.

- **Season :**

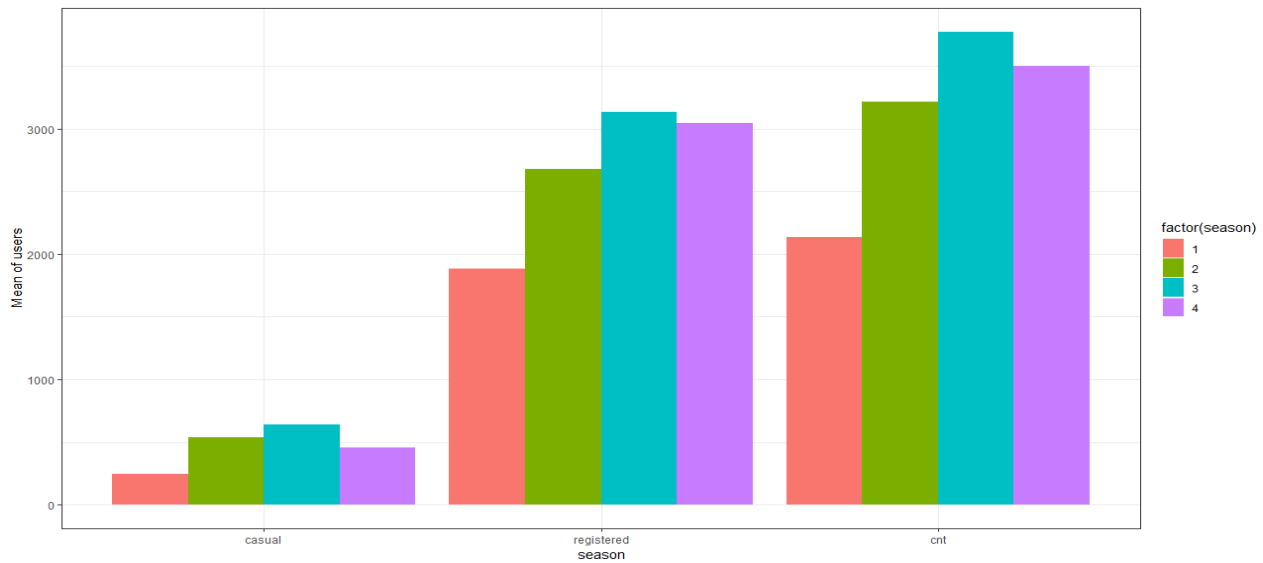


Figure 97: Stacked Bar Plot of season for output variables

We can see that casual, registered and total count all are highest for season 3 and lowest for season 1. So, when season is "fall" people are more likely to rent a bike and during "springer" they are least likely.

This is what we have seen in the EDA phase. So, we can say our hypothesis is true and the model has predicted well for season variable.

- **Year :**

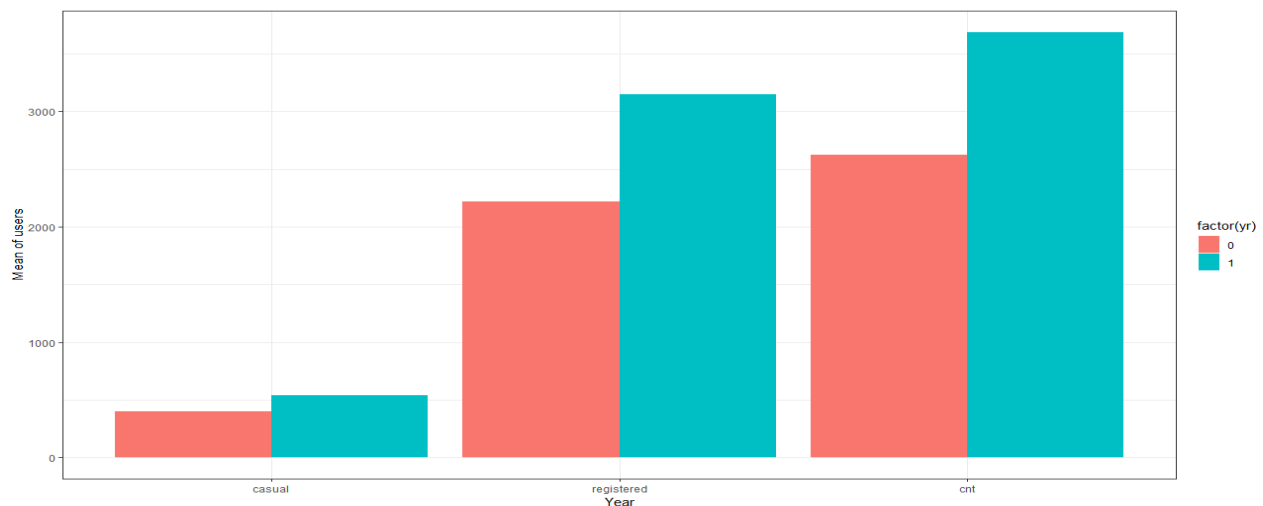


Figure 98: Stacked Bar Plot of year for output variables

From the plots we can say that the count of registered users had heavily increased in 2012 compared to 2011 which is almost 1.5 times and casual users also increased a bit.

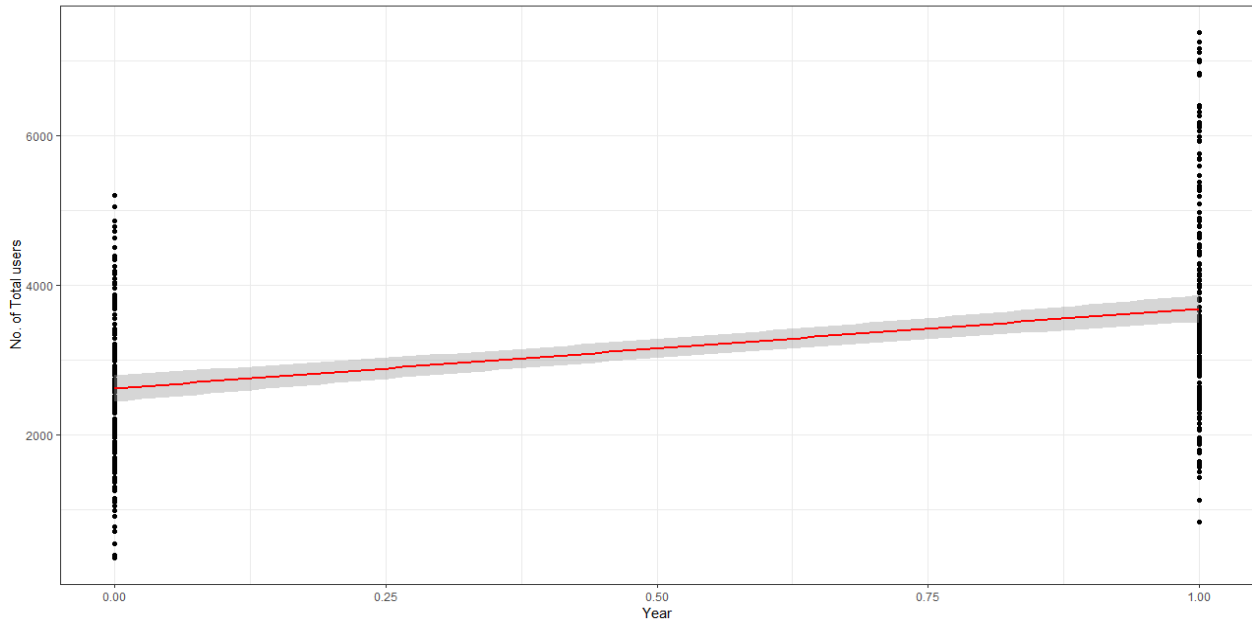


Figure 99:Scatterplot of year vs total count of users for output variable

So, we can see that the regression line shows a positive relationship between year and total users. So as the year has increased the number of rentals has also increased.

This is what we have seen in the EDA phase. So, we can say our hypothesis is true and the model has predicted well for season variable.

- **Weekday :**

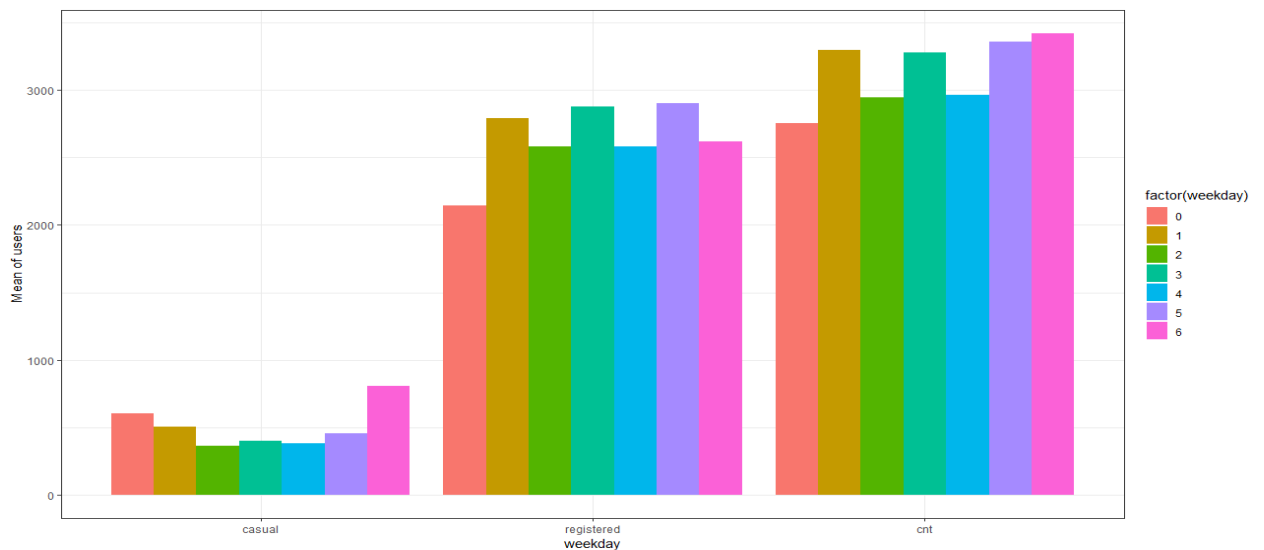


Figure 100:Stacked Bar Plot of weekday for output variables in R

From the plot we can see that the casual count is higher for weekends and lower in weekdays which is absolutely normal and as we have expected. But in case of registered users our model shows some random values. So, we can say that our model has failed to predict the count of registered users based on weekday variable in R.

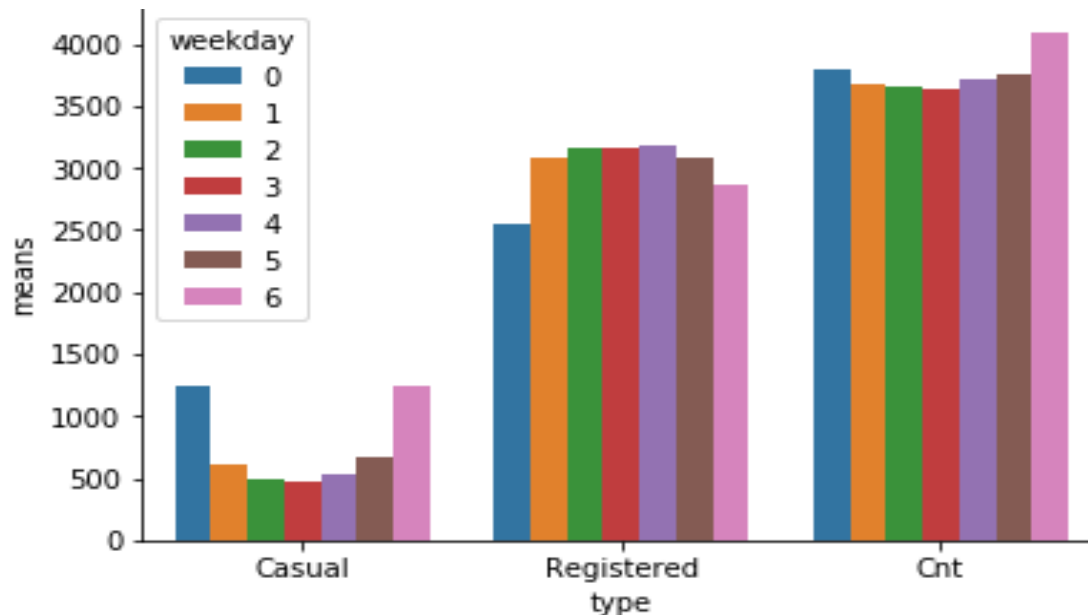


Figure 101:Stacked Bar Plot of weekday for output variables in Python

But in case of Python we can see that that the prediction is correct for both casual and registered counts if we cross verify with the EDA. As casual users use their bike for touring purpose so they are high during weekends. While office goer registered users are higher during weekdays.

- **Holiday :**

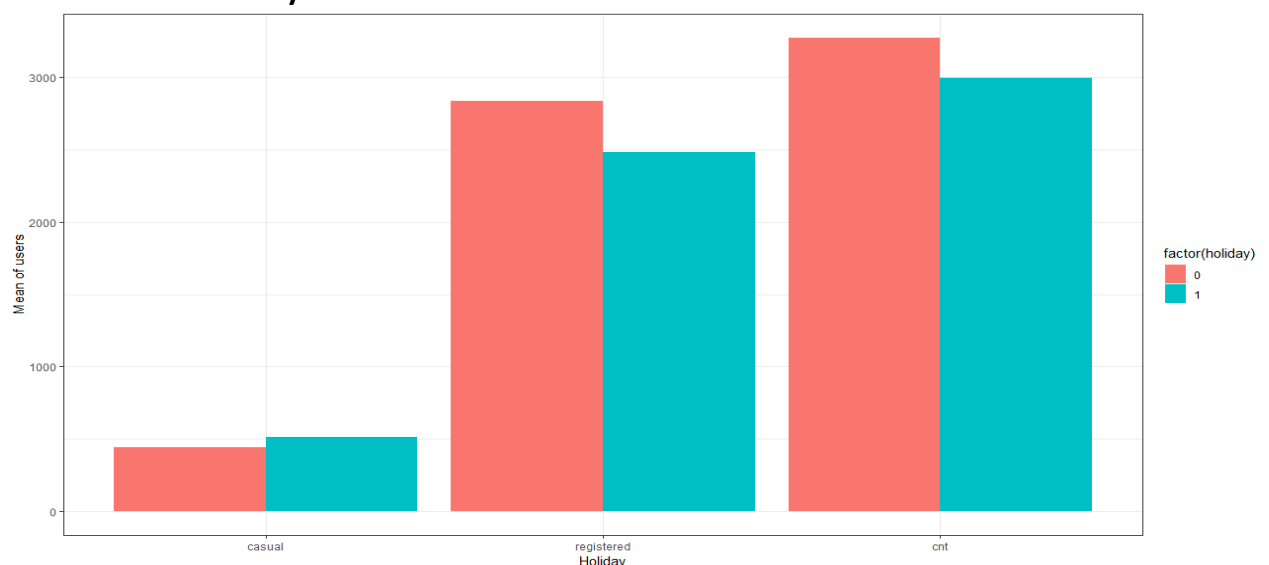


Figure 102:Stacked Bar Plot of holiday for output variables

If it is a holiday then the count of casual users should be more than if the day is a regular day. And the registered users groups contains office goers and professional person so during a holiday they are less likely to rent a bike.

Thus, we can see that our hypothesis is true for both casual and registered users.

- **Weathersit :**

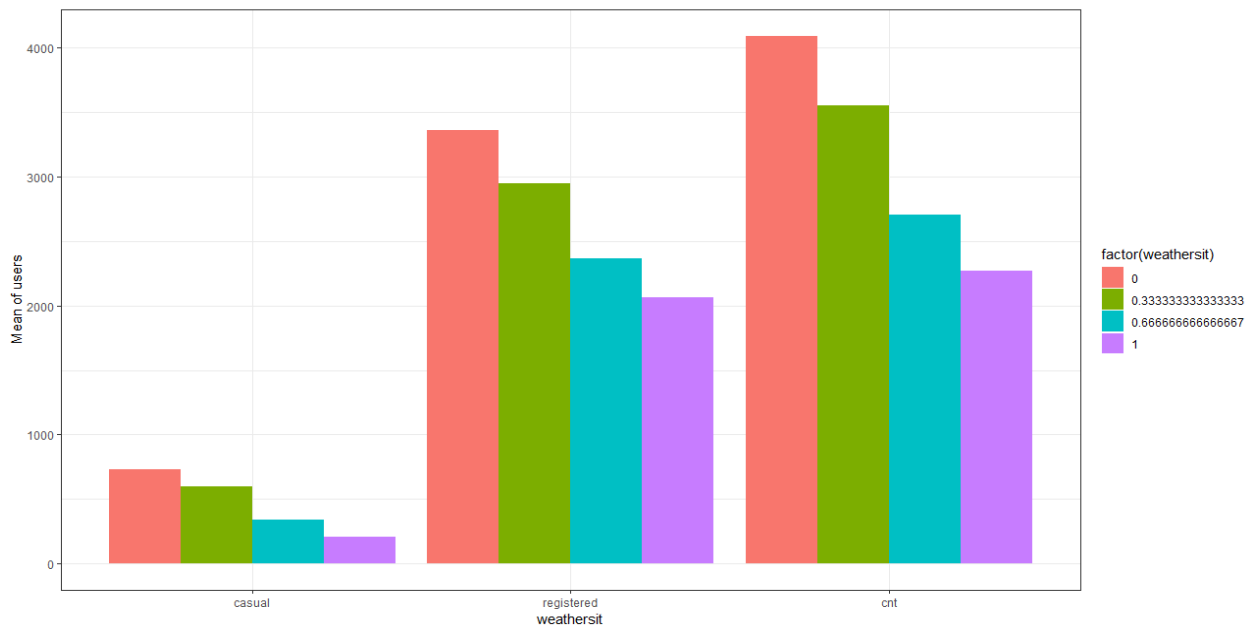


Figure 103:Stacked Bar Plot of weathersit for output variables

From EDA we have seen that weather situation is drastically decreasing from 1 to 4. Thus, we will get very low count of both kind of users for weather condition 4 as the weather is very bad and people are less likely to go out in those condition.

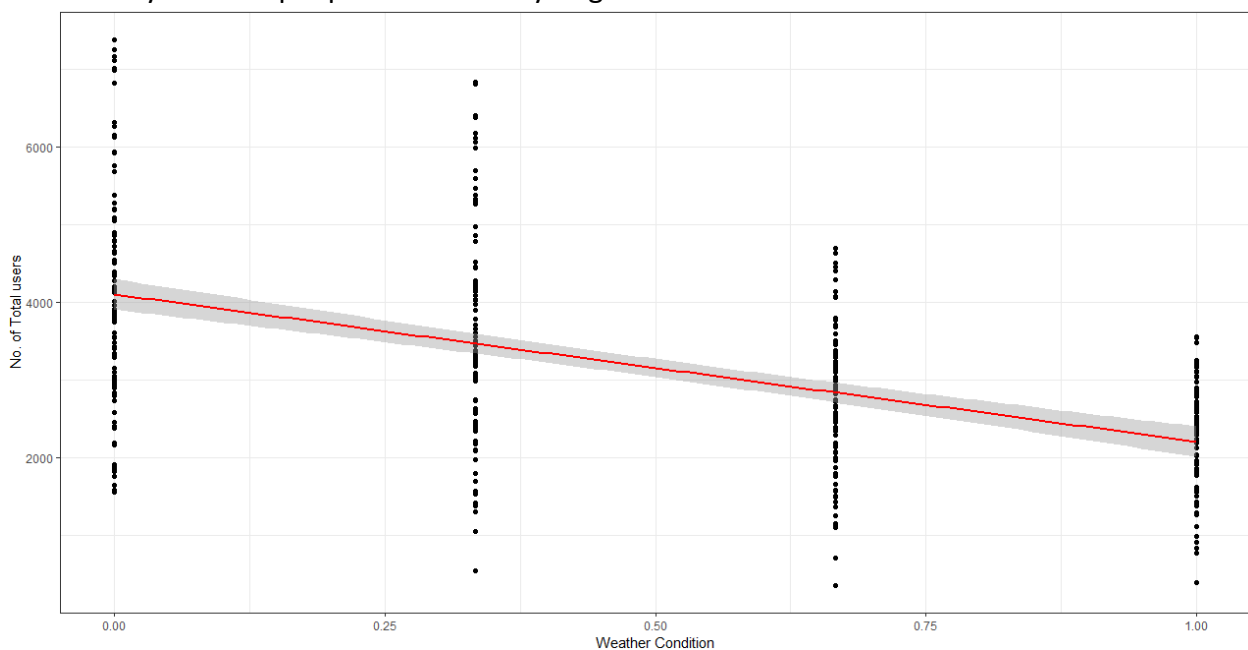


Figure 104:Scatterplot of weathersit vs total count of users for output variable

So, from the above plot we can see that the regression line has shown an inverse relationship between weather condition and count of users. So as the weather condition will increase, i.e.- the weather condition will worsen, the count of rentals will decrease. Thus, from the above plot we can say that our hypothesis is true and our model as predicted very well for weather condition variable.

- **Atemp :**

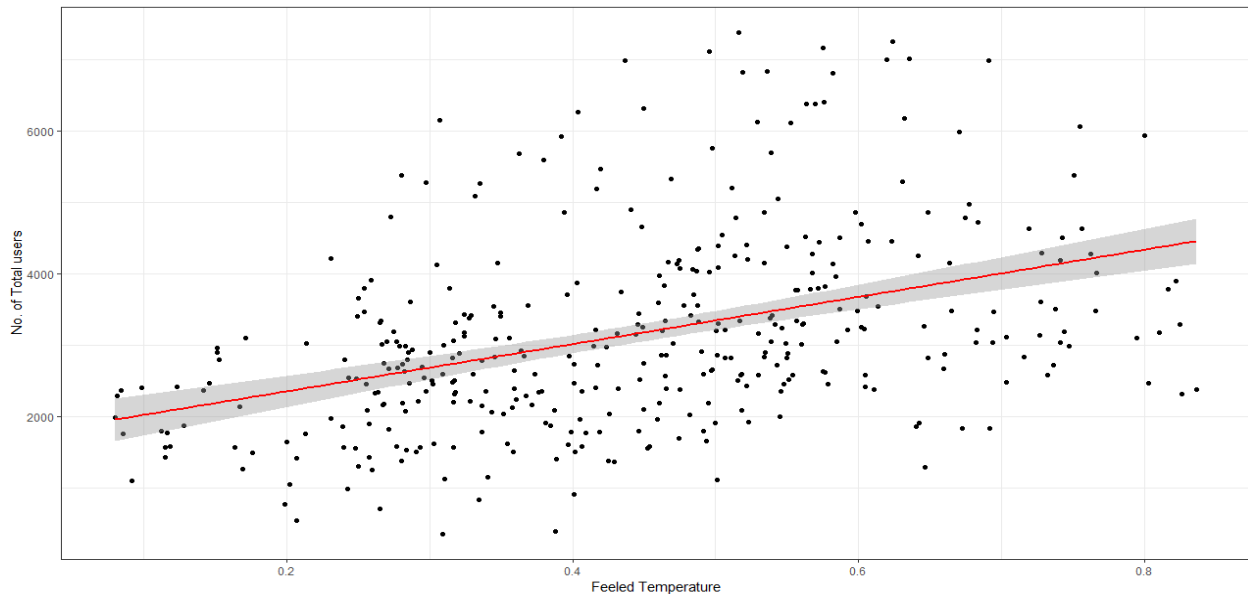


Figure 105: Scatterplot of atemp vs total count of users for output variable

So, from the plot we can see that as the temperature increases people are more likely to rent a bike.

So, we can say that our hypothesis is true in this case as well.

- **Windspeed :**

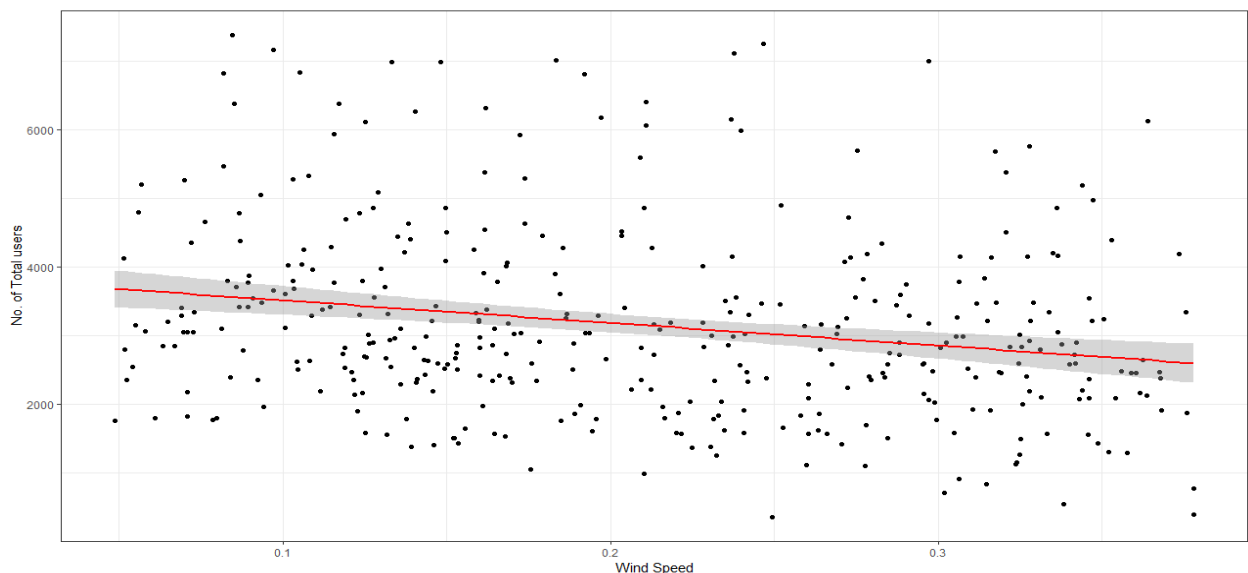


Figure 106: Scatterplot of windspeed vs total count of users for output variable

So, we can see that as the windspeed increases the count of rental decreases. So, people are less likely to rent a bike if the windspeed is high.

Thus, we can see that our hypothesis is true in this case as well.

6. Conclusion :

Thus, from the analysis we have found that there are two types of users for the bike rental company.

- **Casual Users** : This group mainly consists of people who rents the bike for casual use, mainly for tour purpose. So, they rent the bike mainly on holidays and weekends and when the weather condition is favourable for travelling and chilling.
- **Registered Users** : This group consists of office goers and professional people who uses the bike for professional use may be. So, these kind of people mainly rents the bike on weekdays and not on holidays. Moreover, if the weather condition is bad then also we will get at least some registered users.

So, at last we can say that the count of users is more dependent on season, year, holiday, weekday, weather conditions, windspeed and feeld temperature.

Our model predicts very correct results for all the variables and only fails for predicting the registered count of users based on weekdays, which may be due to the presence of other variables.

7. References :

- <https://en.wikipedia.org/>
- <https://learning.edwisor.com/>
- <https://medium.com/>
- <https://www.statisticshowto.datasciencecentral.com/>
- <https://www.theanalysisfactor.com/>