BANA- 7038: Data Analysis & Methodologies

Bike Sharing System Analysis

Presented By:

Jain Akash (M10612553)

Gupta Nikhil (M10604262)

Anand Mohit (M1064698)

Singh Pawan Deep (M10743860)

Table of Contents

1 Bike Sharing System

1.1 Background

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental, and return has become automatic. Through these systems, users can easily rent a bike from a position and return to another position. Currently, there are about over 500 bike-sharing programs around the world which are composed of over 500 thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Apart from interesting real world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of the important events in the city could be detected via monitoring these data.

1.2 Dataset

Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For instance, weather conditions, precipitation, day of week, season, hour of the day, etc. can affect the rental behaviors. The core data set is related to the two-year historical log corresponding to years 2011 and 2012 from Capital Bikeshare system, Washington D.C., USA which is publicly available in http://capitalbikeshare.com/system-data. We aggregated the data on two hourly and daily basis and then extracted and added the corresponding weather and seasonal information. Weather information are extracted from http://www.freemeteo.com.

1.3 Dataset Characteristics

Both hour.csv and day.csv have the following fields, except hr which is not available in day.csv

```
- instant: record index
- dteday : date
- season : season (1:springer, 2:summer, 3:fall, 4:winter)
- yr : year (0: 2011, 1:2012)
- mnth : month ( 1 to 12)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (extracted from http://dchr.dc.gov/page/holiday-schedule)
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
+ weathersit :
   - 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 divided to 41 (max)
- atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. 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
```

1.4 License(Citation)

[1] Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.

2 Data Exploration

We started off by reading the data set stored in "day.csv" file into a data frame is R using the following command.

```
bikerental<-read.csv("C:/Users/Akash/Desktop/DAM Project/Dataset/day.csv",header = TRUE)
head(bikerental)

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

We can see from the above result the data has been loaded into the dataset bikerental.

2.1 Number of observations in the dataset.

We use the following commands to get the count of the number of observations present in the dataset.

```
nrow(bikerental)

## [1] 731
```

We can see from the result above that there are 731 observations in the dataset. As we know from the data charcteristic that the whole data is divided into the observations collected over 2 years. Hence we will divide the dataset into two subsets, one for model building and one to test the prediction. We use the following command to divide the dataset into two subsets.

```
bikerentalyear1<-subset(bikerental, bikerental$yr == 0)
bikerentalyear2<-subset(bikerental, bikerental$yr == 1)
```

Next, we will count the total observations for each of the year.

```
nrow(bikerentalyear1)

## [1] 365

nrow(bikerentalyear2)

## [1] 366
```

We can see from the output of the above commands that we have 365 observations for 2011 i.e. our training set and we have 366 observation for 2012 which is our prediction set.

2.2 Number of columns in the dataset and their characteristics.

We use the following commands to get the count of the number of columns(variables) present in the dataset.

```
ncol(bikerental)

## [1] 16

summary(bikerental)

##     instant          dteday         season            yr        
##  Min.   :  1.0   1/1/2011 :  1   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:183.5   1/1/2012 :  1   1st Qu.:2.000   1st Qu.:0.0000  
##  Median :366.0   1/10/2011:  1   Median :3.000   Median :1.0000  
##  Mean   :366.0   1/10/2012:  1   Mean   :2.497   Mean   :0.5007  
##  3rd Qu.:548.5   1/11/2011:  1   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :731.0   1/11/2012:  1   Max.   :4.000   Max.   :1.0000  
##                  (Other)  :725                                   
##       mnth          holiday           weekday        workingday   
##  Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000  
##  Median : 7.00   Median :0.00000   Median :3.000   Median :1.000  
##  Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684  
##  3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000  
##  Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000  
##                                                                   
##    weathersit         temp             atemp              hum        
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000  
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200  
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267  
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279  
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302  
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725  
##                                                                      
##    windspeed           casual         registered        cnt      
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22  
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152  
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548  
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504  
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956  
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
```

We can see from the result above that there are 16 columns(variables) in the dataset. Out of these columns, cnt is our response variable and the rest are our covariates.

2.3 Check for null values in the dataset.

We will use the following command to check whether null values are present into dataset or not.

```
nacheck<-is.na(bikerental)
sum(nacheck)

## [1] 0
```

We can see from the above results that there are no missing values in the dataset.

2.4 Converting values from normalized form to actual form.

In our dataset, as explained in the data characteristic, we have certain variables which are normalized. These variables are "temp", "atemp", "hum" and "windspeed". We now convert the variables to their actual values using the following code.

```
#changing normalized values of actual temperature to actual values:
bikerentalyear1$actualtemp <- bikerentalyear1$temp*41
#changing feeled temperature to actual values:
bikerentalyear1$feeltemp <- bikerentalyear1$atemp*50
#changing humidity to actual values:
bikerentalyear1$actualhum <- bikerentalyear1$hum*100
#changing windspeed to actual values:
bikerentalyear1$actualwind <- bikerentalyear1$windspeed*67
```

2.5 Checking correlation between variables in the dataset.

We will use the following function to check the correlation between the variables.

```
cor(bikerentalyear1[3:20])

## Warning in cor(bikerentalyear1[3:20]): the standard deviation is zero

##                   season yr         mnth       holiday      weekday
## season      1.0000000000 NA  0.831032052  0.0002072362 -0.011705146
## yr                    NA  1           NA            NA           NA
## mnth        0.8310320517 NA  1.000000000  0.0328079834  0.012859633
## holiday     0.0002072362 NA  0.032807983  1.0000000000 -0.076086528
## weekday    -0.0117051457 NA  0.012859633 -0.0760865280  1.000000000
## workingday  0.0071365286 NA -0.004288059 -0.2474610821  0.020445487
## weathersit  0.0355084485 NA  0.009729138 -0.0064418544  0.047259261
## temp        0.3733798908 NA  0.288663252 -0.0192724051 -0.039292166
## atemp       0.3829722773 NA  0.301920456 -0.0264481391 -0.042809516
## hum         0.2494507422 NA  0.242532537 -0.0308961010 -0.065931579
## windspeed  -0.2425140393 NA -0.242443274  0.0007344413  0.061525174
## casual      0.2505648515 NA  0.169796954  0.0898532055 -0.019603665
## registered  0.5731658363 NA  0.489148092 -0.1111278777  0.004568869
## cnt         0.5417940707 NA  0.444607187 -0.0491931651 -0.004396295
## actualtemp  0.3733798908 NA  0.288663252 -0.0192724051 -0.039292166
## feeltemp    0.3829722773 NA  0.301920456 -0.0264481391 -0.042809516
## actualhum   0.2494507422 NA  0.242532537 -0.0308961010 -0.065931579
## actualwind -0.2425140393 NA -0.242443274  0.0007344413  0.061525174
##               workingday   weathersit        temp       atemp          hum
## season       0.007136529  0.035508448  0.37337989  0.38297228  0.249450742
## yr                    NA           NA          NA          NA           NA
## mnth        -0.004288059  0.009729138  0.28866325  0.30192046  0.242532537
## holiday     -0.247461082 -0.006441854 -0.01927241 -0.02644814 -0.030896101
## weekday      0.020445487  0.047259261 -0.03929217 -0.04280952 -0.065931579
## workingday   1.000000000  0.108654420  0.04679922  0.04615815  0.034249681
## weathersit   0.108654420  1.000000000 -0.09117466 -0.09689387  0.581475773
## temp         0.046799218 -0.091174656  1.00000000  0.99645761  0.145776184
## atemp        0.046158148 -0.096893869  0.99645761  1.00000000  0.155811515
## hum          0.034249681  0.581475773  0.14577618  0.15581152  1.000000000
## windspeed    0.011954932  0.109309983 -0.11420017 -0.13654376 -0.215718023
## casual      -0.541419395 -0.279370271  0.58103786  0.58115314 -0.032290458
## registered   0.310969417 -0.267346716  0.69813575  0.70338097  0.019412295
## cnt          0.020661433 -0.318274470  0.77121420  0.77529371  0.001898085
## actualtemp   0.046799218 -0.091174656  1.00000000  0.99645761  0.145776184
## feeltemp     0.046158148 -0.096893869  0.99645761  1.00000000  0.155811515
## actualhum    0.034249681  0.581475773  0.14577618  0.15581152  1.000000000
## actualwind   0.011954932  0.109309983 -0.11420017 -0.13654376 -0.215718023
```

```
##              windspeed     casual   registered         cnt  actualtemp
## season     -0.2425140393  0.25056485  0.573165836  0.541794071  0.37337989
## yr                    NA          NA           NA           NA          NA
## mnth       -0.2424432735  0.16979695  0.489148092  0.444607187  0.28866325
## holiday     0.0007344413  0.08985321 -0.111127878 -0.049193165 -0.01927241
## weekday     0.0615251739 -0.01960366  0.004568869 -0.004396295 -0.03929217
## workingday  0.0119549318 -0.54141939  0.310969417  0.020661433  0.04679922
## weathersit  0.1093099833 -0.27937027 -0.267346716 -0.318274470 -0.09117466
## temp       -0.1142001730  0.58103786  0.698135755  0.771214198  1.00000000
## atemp      -0.1365437587  0.58115314  0.703380975  0.775293710  0.99645761
## hum        -0.2157180230 -0.03229046  0.019412295  0.001898085  0.14577618
## windspeed   1.0000000000 -0.19051690 -0.261590496 -0.277999968 -0.11420017
## casual     -0.1905168958  1.00000000  0.396546502  0.708358731  0.58103786
## registered -0.2615904965  0.39654650  1.000000000  0.928880205  0.69813575
## cnt        -0.2779999682  0.70835873  0.928880205  1.000000000  0.77121420
## actualtemp -0.1142001730  0.58103786  0.698135755  0.771214198  1.00000000
## feeltemp   -0.1365437587  0.58115314  0.703380975  0.775293710  0.99645761
## actualhum  -0.2157180230 -0.03229046  0.019412295  0.001898085  0.14577618
## actualwind  1.0000000000 -0.19051690 -0.261590496 -0.277999968 -0.11420017
##               feeltemp    actualhum    actualwind
## season      0.38297228  0.249450742 -0.2425140393
## yr                  NA           NA            NA
## mnth        0.30192046  0.242532537 -0.2424432735
## holiday    -0.02644814 -0.030896101  0.0007344413
## weekday    -0.04280952 -0.065931579  0.0615251739
## workingday  0.04615815  0.034249681  0.0119549318
## weathersit -0.09689387  0.581475773  0.1093099833
## temp        0.99645761  0.145776184 -0.1142001730
## atemp       1.00000000  0.155811515 -0.1365437587
## hum         0.15581152  1.000000000 -0.2157180230
## windspeed  -0.13654376 -0.215718023  1.0000000000
## casual      0.58115314 -0.032290458 -0.1905168958
## registered  0.70338097  0.019412295 -0.2615904965
## cnt         0.77529371  0.001898085 -0.2779999682
## actualtemp  0.99645761  0.145776184 -0.1142001730
## feeltemp    1.00000000  0.155811515 -0.1365437587
## actualhum   0.15581152  1.000000000 -0.2157180230
## actualwind -0.13654376 -0.215718023  1.0000000000
```

From the correlation matrix, we can infer the following:

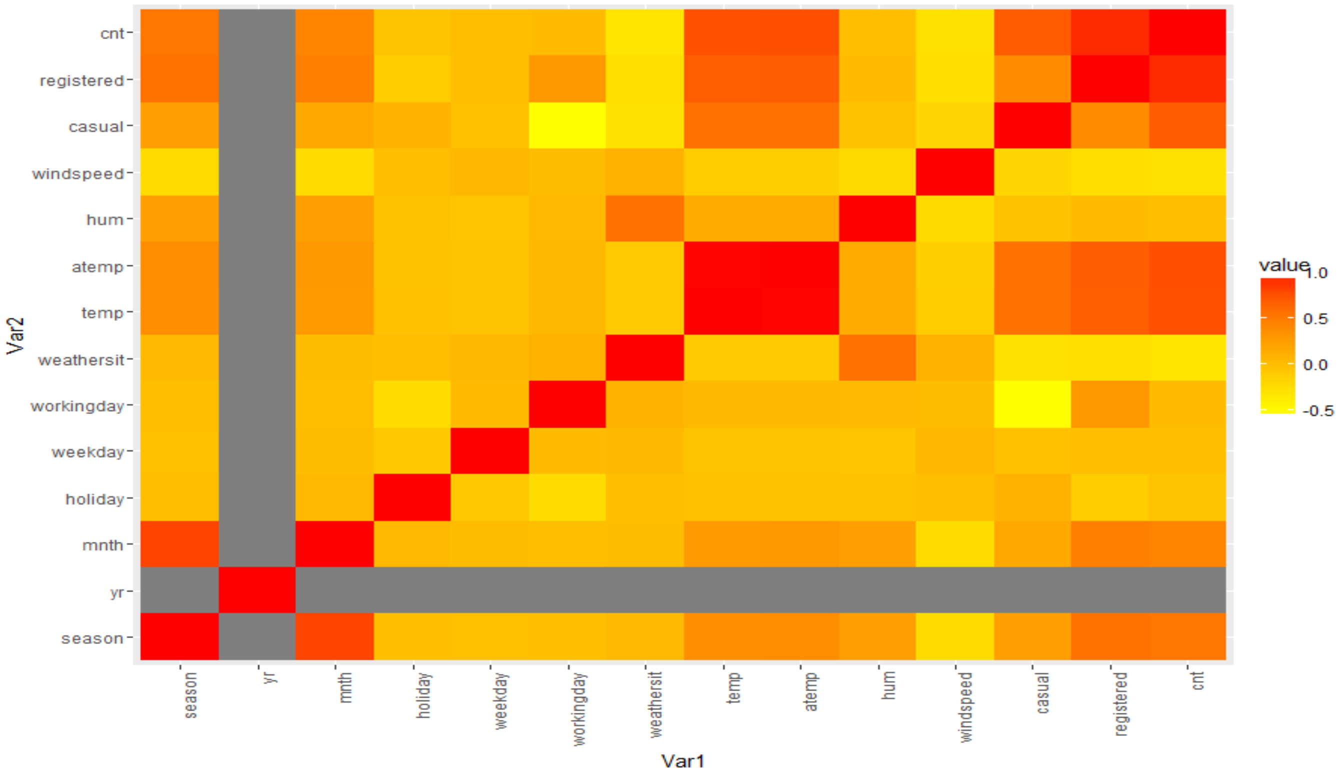
- Count(cnt) is correlated with temp, atemp, season, month, weathersit, casual, registered and windspeed. *Since* temp and atemp are highly correlated with each other and hence we will include only one of them into our model to avoid multicollinearity. We are including actualtemp(actual value of temp)
- Season and month are highly correlated with each other and hence we will include only one of them into our model to avoid multicollinearity. We are including season
- Casual, registered and count is highly correlated. Single count(cnt) is basically the sum of casual and registered, there is no need to include them into the model.

3 Data Visualization

Correlation heatmap for bikerentalyear1:

```
m <- data.frame(bikerentalyear1[,3:16] )
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.2
```

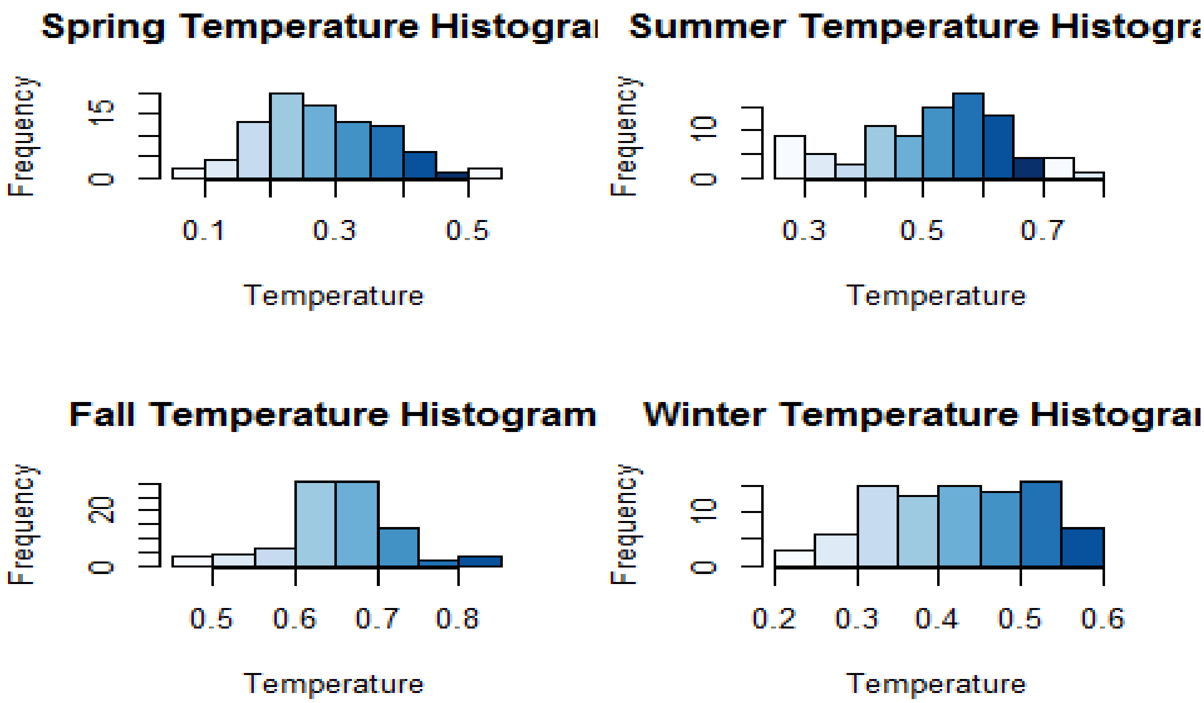
```
library(reshape2)
qplotdate <- qplot(x=Var1, y= Var2, data=melt(cor(m)), fill=value, geom="tile")

## Warning in cor(m): the standard deviation is zero

qplotdate+scale_fill_gradient(low="yellow", high="red"+theme(axis.test.x=element_text (angle=90, hjust=1))
```



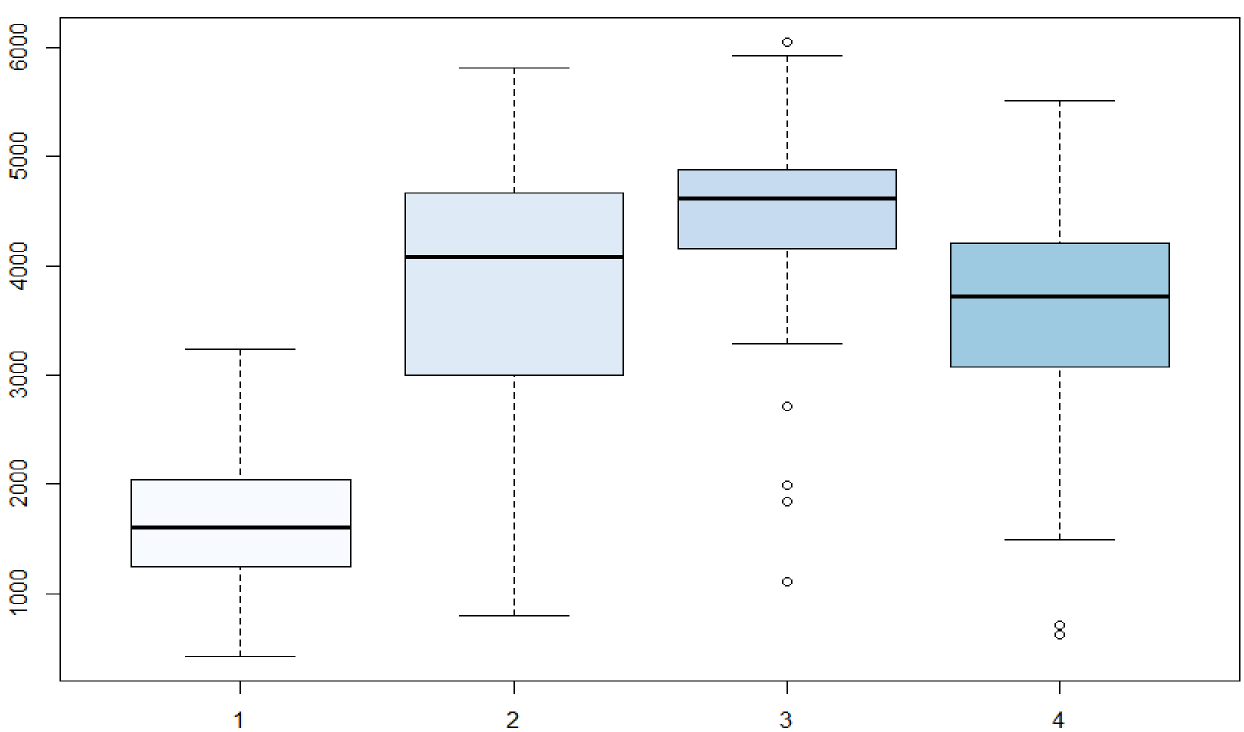
The result of the above heatmap concurs with the results in section 2.5.

Plotting temperatures for different seasons:

```
#temperature ranges for each season:
springer1 <- subset(bikerentalyear1,season == 1)
smean1 <- mean(springer1$atemp)
sstd1 <- sd(springer1$atemp)

summer <- subset(bikerentalyear1,season == 2)
smean2 <- mean(summer$atemp)
sstd2 <- sd(summer$atemp)

fall <- subset(bikerentalyear1,season == 3)
smean3 <- mean(fall$atemp)
sstd3 <- sd(fall$atemp)

winter <- subset(bikerentalyear1,season == 4)
smean4 <- mean(winter$atemp)
sstd4 <- sd(winter$atemp)

par(mfrow = c(2,2))
hist(springer1$atemp, main = "Spring Temperature Histogram", xlab = "Temperature", col = blues9 )
hist(summer$atemp, main = "Summer Temperature Histogram", xlab = "Temperature", col = blues9 )
hist(fall$atemp, main = "Fall Temperature Histogram", xlab = "Temperature", col = blues9 )
hist(winter$atemp, main = "Winter Temperature Histogram", xlab = "Temperature", col = blues9 )
```




```
bikerentalyear1$season <- as.factor(bikerentalyear1$season)
plot(bikerentalyear1$season,bikerentalyear1$cnt, col = blues9)
```



We can see from the histograms and the mean values of the temperature for different seasons that the temperature variations for the different season are in order: Fall, Summer, Winter and then spring.

4 Model Building

4.1 Selection of covariates

From section 2.5, we narrowed down our covariates to actualtemp, season, weathersit, windspeed and hum. Since season and weathersit are categorical variables, we will convert them to factors using the following code:

```
bikerentalyear1$weatherfac <- as.factor(bikerentalyear1$weathersit)
bikerentalyear1$seasonfac <- as.factor(bikerentalyear1$season)
head(bikerentalyear1)

##   instant   dteday season yr mnth holiday weekday workingday weathersit
## 1       1 1/1/2011      1  0    1       0       6          0          2
## 2       2 1/2/2011      1  0    1       0       0          0          2
## 3       3 1/3/2011      1  0    1       0       1          1          1
## 4       4 1/4/2011      1  0    1       0       2          1          1
## 5       5 1/5/2011      1  0    1       0       3          1          1
## 6       6 1/6/2011      1  0    1       0       4          1          1
##       temp    atemp      hum windspeed casual registered  cnt actualtemp
## 1 0.344167 0.363625 0.805833 0.1604460    331        654  985  14.110847
## 2 0.363478 0.353739 0.696087 0.2485390    131        670  801  14.902598
## 3 0.196364 0.189405 0.437273 0.2483090    120       1229 1349   8.050924
## 4 0.200000 0.212122 0.590435 0.1602960    108       1454 1562   8.200000
## 5 0.226957 0.229270 0.436957 0.1869000     82       1518 1600   9.305237
## 6 0.204348 0.233209 0.518261 0.0895652     88       1518 1606   8.378268
##   feeltemp actualhum actualwind weatherfac seasonfac
## 1 18.18125   80.5833  10.749882          2         1
## 2 17.68695   69.6087  16.652113          2         1
## 3  9.47025   43.7273  16.636703          1         1
## 4 10.60610   59.0435  10.739832          1         1
## 5 11.46350   43.6957  12.522300          1         1
## 6 11.66045   51.8261   6.000868          1         1
```

So our new list of covariates are actualtemp, weatherfac, seasonfac, windspeed and hum.

4.2 Model Building

We will start with building a multiple linear regression model by taking the covariates confirmed in section 4.1 and taking the response variable as cnt(count). To build the multiple linear regression model, we will use the following command:

```
attach(bikerentalyear1)
rentalmodel <- lm(cnt~actualtemp+seasonfac+weatherfac+windspeed+hum, data = bikerentalyear1)
summary(rentalmodel)
## Call:
## lm(formula = cnt ~ actualtemp + seasonfac + weatherfac + windspeed +
##     hum, data = bikerentalyear1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2437.19  -351.96    46.09   398.13  1559.13
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1685.680    223.466   7.543 3.86e-13 ***
## actualtemp   103.500      7.547  13.715  < 2e-16 ***
## seasonfac2  1073.883    121.984   8.803  < 2e-16 ***
```

```
## seasonfac3    947.963    160.017   5.924 7.41e-09 ***
## seasonfac4   1453.264    108.052  13.450  < 2e-16 ***
## weatherfac2  -307.801     85.820  -3.587 0.000382 ***
## weatherfac3 -1657.471    190.442  -8.703  < 2e-16 ***
## windspeed   -2309.782    462.086  -4.999 9.08e-07 ***
## hum         -931.339    300.369  -3.101 0.002085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 619.1 on 356 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7984
## F-statistic: 181.2 on 8 and 356 DF,  p-value: < 2.2e-16
```

From the above results, we can see that the 'rentalmodel' has been created and the fit of the model is quite satisfactory.

4.3 Hypothesis testing and Partial Testing

To check if the all the covariates are significant in determining the response variable and to check the overall adequacy of the model, we will perform Hypothesis test and Partial tests.

4.3.1 Checking overall adequacy of the model

To check the adequacy of the model, we will perform the test for significance to test if there is a linear relationship between the response variable and any of the covariates. To perform the test of significance, we use the F-test.

To perform the F-test we first form the null (H_0) and alternate hypothesis (H_1), which are as follows:
H_0**:There is no linear relation between response variable and covarites**
H_1**:There is a linear relation between response variable and covariates**
Explanation: In out Null hypothesis, we assume that there is no collective effect of our covariates on the response variable. Our alternate hypothesis states that our covariates collectively influence the response variable. We execute the following command to get the F-test results, also known as the F-stats for this model:

```
anova(rentalmodel)
```

```
## Analysis of Variance Table
## Response: cnt
##              Df    Sum Sq   Mean Sq  F value    Pr(>F)
## actualtemp    1 411552057 411552057 1073.810 < 2.2e-16 ***
## seasonfac     3  66695717  22231906   58.007 < 2.2e-16 ***
## weatherfac    2  66606458  33303229   86.894 < 2.2e-16 ***
## windspeed     1   6969372   6969372   18.184 2.572e-05 ***
## hum           1   3684698   3684698    9.614  0.002085 **
## Residuals   356 136441748    383263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output of the above query shows that for each of the covariate *p-value < alpha(0.05)*, which clearly indicates that we should reject the null hypothesis at the 5% level of significance, that the covariates collectively have no effect on the response variable.

4.3.2 Hypothesis test for estimate coefficients

To perform t-tests for each of the regression coefficient estimate, we will have to perform the following steps:

- Form the null hypothesis- H_0: β_i = 0, where i = [1,8]
- Form the alternate hypothesis - H_1: β_i != 0, where i = [1,8]
- Obtain t-stat for β_i
- Evaluate the significance of each of the regression coefficient.

Note: H_0: β_i = 0, where i = [1,8], the t-test is basically being done to determine whether the regression coefficient is significant or not and hence in null hypothesis we state that the significance of the regression coefficient is negligible.

To obtain the t-stat for each of the coefficients, we can execute the following commands:

```
summary(rentalmodel)$coef[,3:4]

##              t value     Pr(>|t|)
## (Intercept)  7.543352 3.857367e-13
## actualtemp  13.714939 1.152500e-34
## seasonfac2   8.803438 5.865333e-17
## seasonfac3   5.924143 7.407138e-09
## seasonfac4  13.449704 1.252120e-33
## weatherfac2 -3.586595 3.818410e-04
## weatherfac3 -8.703308 1.217563e-16
## windspeed   -4.998594 9.077875e-07
## hum         -3.100647 2.084954e-03
```

For the results above, below things can be said about the regression coefficients at 5% level of significance:

- β_0(regression coefficient for intercept), is significant because the p-value(3.857e-13) < alpha(0.05) and hence the null hypothesis(β_0 = 0 i.e. intercept being zero) will be rejected
- β_1(regression coefficient for actualtemp), is not significant because the p-value(1.15e-34) < alpha(0.05) and hence the null hypothesis(β_1 = 0 i.e. the effect of actualtemp on cnt is insignificant) will be rejected
- β_2(regression coefficient for seasonfac2), is not significant because the p-value(5.865e-17) < alpha(0.05) and hence the null hypothesis(β_2 = 0 i.e. the effect of seasonfac2 on cnt is insignificant) will not be rejected
- β_3(regression coefficient for seasonfac3), is significant because the p-value(7.40e-08) < alpha(0.05) and hence the null hypothesis(β_3 = 0 i.e. the effect of seasonfac3 on cnt is insignificant) will be rejected
- β_4(regression coefficient for seasonfac4), is significant because the p-value(1.25e-33) < alpha(0.05) and hence the null hypothesis(β_4 = 0 i.e. the effect of seasonfac4 on cnt is significant) will be rejected
- β_5(regression coefficient for weatherfac2), is significant because the p-value(3.818e-04) < alpha(0.05) and hence the null hypothesis(β_5 = 0 i.e. the effect of weatherfac2 on cnt is insignificant) will be rejected
- β_6(regression coefficient for weatherfac3), is significant because the p-value(1.217e-16) < alpha(0.05) and hence the null hypothesis(β_6 = 0 i.e. the effect of weatherfac3 on cnt is insignificant) will be rejected
- β_7(regression coefficient for windspeed), is significant because the p-value(9.077e-07) < alpha(0.05) and hence the null hypothesis(β_7 = 0 i.e. the effect of windspeed on cnt is insignificant) will be rejected
- β_8(regression coefficient for hum), is significant because the p-value(2.084e-03) < alpha(0.05) and hence the null hypothesis(β_8 = 0 i.e. the effect of hum on cnt is insignificant) will be rejected.

5 Taking remaining variables into consideration

Till now we have taken actualtemp, weatherfac, seasonfac, windspeed and hum into consideration. Some variables like mnth, atemp, casual and registered were rejected based on section 2.5. The remaining variables to be taken into consideration are holiday, weekday and workingday.

We will check each of the remaining variables by adding them to the rentalmodel and apply F-test to check their significance.

5.1 Check for holiday

We will make a new model "rentalmodelholi" by adding holiday to the previous model and then apply F-test to it.

```
rentalmodelholi <- lm(cnt~actualtemp+seasonfac+weatherfac+windspeed+hum+holiday, data = bikerentalyear1)
anova(rentalmodel,rentalmodelholi)
```

```
## Analysis of Variance Table
##
## Model 1: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum
## Model 2: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum +
##     holiday
##   Res.Df       RSS Df Sum of Sq      F  Pr(>F)
## 1    356 136441748
## 2    355 135127267  1   1314481 3.4533 0.06395 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the above result, we can see that the p-value(.06) > alpha (.05) and we fail to reject the null hypothesis that holiday has no effect on cnt. Hence, we will not add holiday to the model.

5.2 Check for weekday

We make a new model "rentalmodelweekday" by adding holiday to the previous model and then apply F-test to it.

```
rentalmodelweekday <- lm(cnt~actualtemp+seasonfac+weatherfac+windspeed+hum+weekday, data = bikerentalyear1)
anova(rentalmodel,rentalmodelweekday)
```

```
## Analysis of Variance Table
## Model 1: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum
## Model 2: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum +
##     weekday
##   Res.Df       RSS Df Sum of Sq      F Pr(>F)
## 1    356 136441748
## 2    355 135526779  1    914968 2.3967 0.1225
```

From the above result, we can see that the p-value(.12) > alpha (.05) and we fail to reject the null hypothesis that weekday has no effect on cnt. Hence we will not add weekday to the model.

5.2 Check for workingday

We make a new model "rentalmodelworking" by adding holiday to the previous model and then apply F-test to it.

```
rentalmodelworking <- lm(cnt~actualtemp+seasonfac+weatherfac+windspeed+hum+workingday, data = bikerentalyear1)
anova(rentalmodel,rentalmodelworking)
```

```
## Analysis of Variance Table
## Model 1: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum
## Model 2: cnt ~ actualtemp + seasonfac + weatherfac + windspeed + hum +
##     workingday
##   Res.Df       RSS Df Sum of Sq      F Pr(>F)
## 1    356 136441748
## 2    355 136180737  1    261010 0.6804   0.41
```

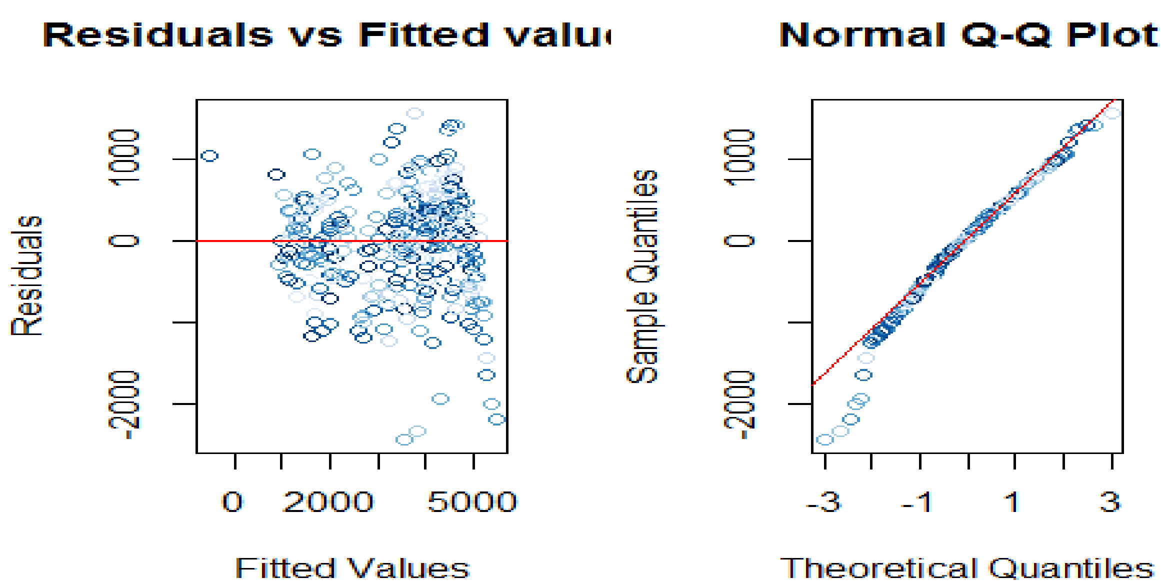
From the above result, we can see that the p-value(.41) > alpha (.05) and we fail to reject the null hypothesis that weekday has no effect on cnt. Hence, we will not add workingday to the model.

Hence our final list of covariates are actualtemp, seasonfac, weatherfac, windspeed, hum.

6 Analysis of Residuals

Since "rentalmodel" is our final model, we will analyze the residuals to check whether they are randomly distributed around zero or not.

```
par(mfrow=c(1,2))
plot(rentalmodel$fitted.values,rentalmodel$residuals, xlab = "Fitted Values", ylab = "Residuals", main = "Residuals vs Fitted values", col = blues9)
abline(h=0, col = "red")
qqnorm(rentalmodel$residuals, col = blues9)
qqline(rentalmodel$residuals, col = "red")
```



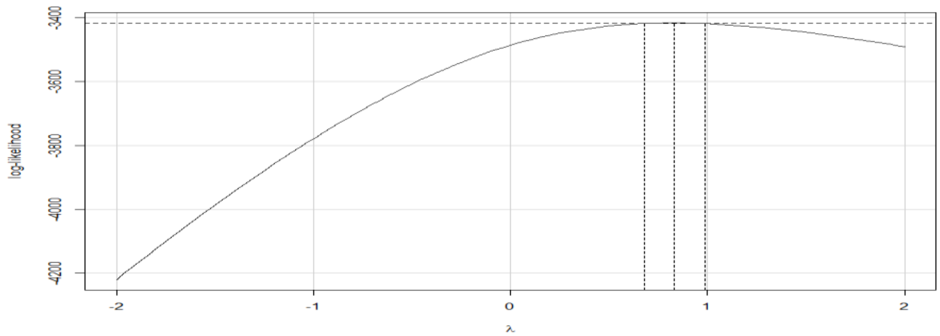
From the above graphs, we can conclude that the residuals are random normally distributed with some outliers.

7 Transforming the model using BoxCox Method

From section 6, evidently, the model is a good fit with random normally distributed residuals. Now, in order to get a good fit with less residual standard error, we will apply the boxcox transformation to out model.

```
par(mfrow=c(1,1))
library(MASS,quietly = TRUE)
```

```
boxCox(rentalmodel)
```



From the above graph, the value of lambda should be between .8 to .99. We tried all the combinations and the best result we got was with lambda = .95. To apply the transformation, we will execute the following code.

```
bikerentalyear1$transcnt <- bikerentalyear1$cnt*.95
rentalmodelFinal<-lm(transcnt~actualtemp+weatherfac+hum+windspeed+seasonfac, data=bikerentalyear1)
summary(rentalmodelFinal)

## Call:
## lm(formula = transcnt ~ actualtemp + weatherfac + hum + windspeed +
##     seasonfac, data = bikerentalyear1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2315.33  -334.37    43.78   378.23  1481.17
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1601.396    212.292   7.543 3.86e-13 ***
## actualtemp     98.325      7.169  13.715  < 2e-16 ***
## weatherfac2  -292.411     81.529  -3.587 0.000382 ***
## weatherfac3 -1574.598    180.919  -8.703  < 2e-16 ***
## hum         -884.772    285.351  -3.101 0.002085 **
## windspeed   -2194.293    438.982  -4.999 9.08e-07 ***
## seasonfac2   1020.189    115.885   8.803  < 2e-16 ***
## seasonfac3    900.565    152.016   5.924 7.41e-09 ***
## seasonfac4   1380.601    102.649  13.450  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 588.1 on 356 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7984
## F-statistic: 181.2 on 8 and 356 DF,  p-value: < 2.2e-16

summary(rentalmodel)
## Call:
## lm(formula = cnt ~ actualtemp + seasonfac + weatherfac + windspeed +
##     hum, data = bikerentalyear1)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2437.19  -351.96    46.09   398.13  1559.13
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1685.680    223.466   7.543 3.86e-13 ***
## actualtemp    103.500      7.547  13.715  < 2e-16 ***
## seasonfac2   1073.883    121.984   8.803  < 2e-16 ***
## seasonfac3    947.963    160.017   5.924 7.41e-09 ***
```

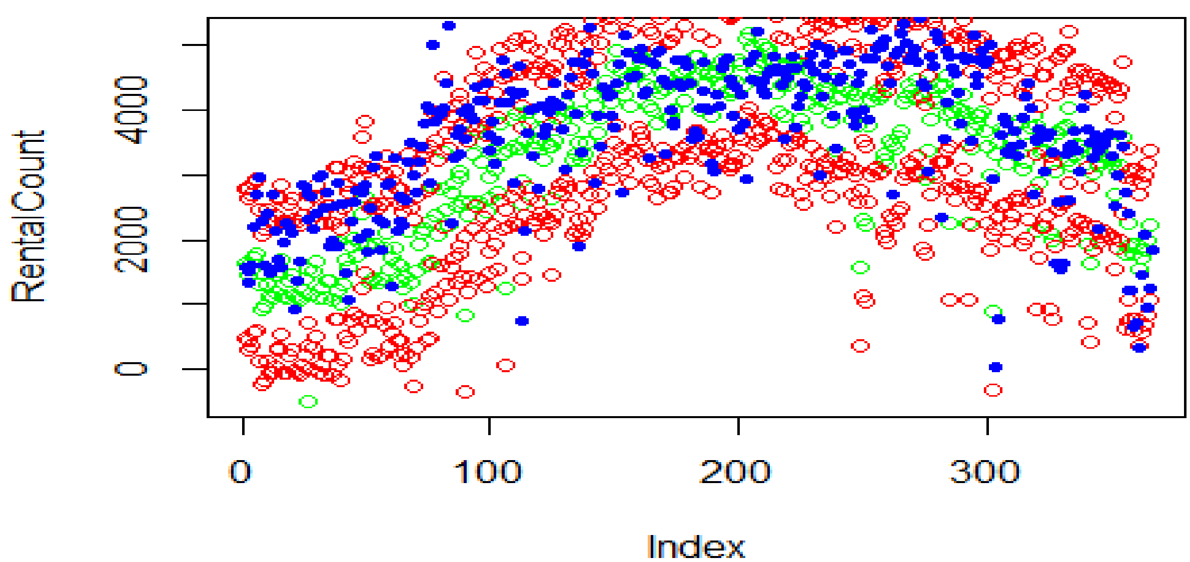
```
## seasonfac4   1453.264    108.052  13.450  < 2e-16 ***
## weatherfac2  -307.801     85.820  -3.587 0.000382 ***
## weatherfac3 -1657.471    190.442  -8.703  < 2e-16 ***
## windspeed   -2309.782    462.086  -4.999 9.08e-07 ***
## hum          -931.339    300.369  -3.101 0.002085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 619.1 on 356 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7984
## F-statistic: 181.2 on 8 and 356 DF,  p-value: < 2.2e-16
```

From the above result, we can see that the transformation did not change the fit of the model but it reduced the Residual standard error and hence we will accept the updated model.

8 Using model to predict the future values

In section 2, we divided the data into two sets, one for model building and one for prediction testing. Now we will use out a final model to get the prediction values for the next year and compare them with the actual observed values. We will use the following commands to perform the prediction:

```
bikerentalyear2$transcnt<-bikerentalyear2$cnt^.95
bikeyear2<-predict(rentalmodelFinal, interval =  "predict")

## Warning in predict.lm(rentalmodelFinal, interval = "predict"): predictions on current data refer to _future_
responses

plot(bikeyear2[,1],type = "p",col="green", ylab = "RentalCount")
points(bikeyear2[,2],type = "p",col="red")
points(bikeyear2[,3],type = "p",col="red")
points(bikerentalyear2$transcnt, type="p", pch = 20 , col = "blue")
```

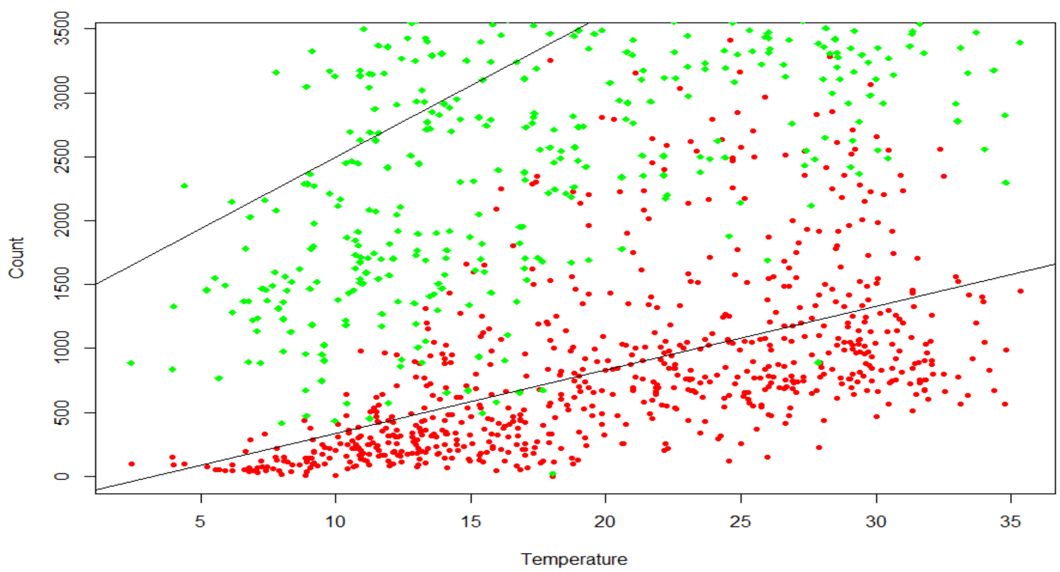


From the above graph, we can see that our final model was able to predict the values for the rental count for the next year quite accurately.

9 Additional Insights

Since our count of rental is divided into casual and registered. We are interested in finding the trend of temperature with bike rentals for registered and casual users. To do that, we execute the following code:

```
plot(bikerental$actualtemp,bikerental$casual,col="red", xlab = "Temperature", ylab = "Count", pch = 20)
points(bikerental$actualtemp,bikerental$registered, col="green", pch = 18)

abline(lm(bikerental$registered~bikerental$actualtemp))

abline(lm(bikerental$casual~bikerental$actualtemp))
```



From the above result, we can conclude that the count of Rental Bikes increase with temperature and this increase is more drastic for registered users as compared to that for casual users.

10 Conclusions

Following conclusions can be made from the above analysis:

- Bike Rental is highly dependent on temperature
- Bike Rental in all depends on the actual temperature, season, Weather, humidity and windspeed
- Bike Rental count is maximum in fall because all the above parameters are optimal in fall
- Bike Rental increases more rapidly with temperature for registered users as compared to that for casual users.