Bike_Sharing_Project

@Akhil

27 February 2019

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

== Background ==

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousands 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 important events in the city could be detected via monitoring these data.

== Data Set == 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 (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 (http://www.freemeteo.com).

== 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
```

Step1: ->Importing Dataset and splitting into test set and training set ->We remove attribute instant as it does not make any contribution to the predictions ->odata is original dataset while data is modified dataset

```
odata=read.csv("D:/Study/PESU IO Data_Analytics_with_R/Final_Project/day.csv")
data=odata[,-1]
data=data[,-1]
set.seed(123)
split = sample.split(data$cnt, SplitRatio = 0.8)
training_set=subset(data,split==TRUE)
test_set=subset(data,split==FALSE)
```

```
print(min(data$cnt))
```

```
## [1] 22
```

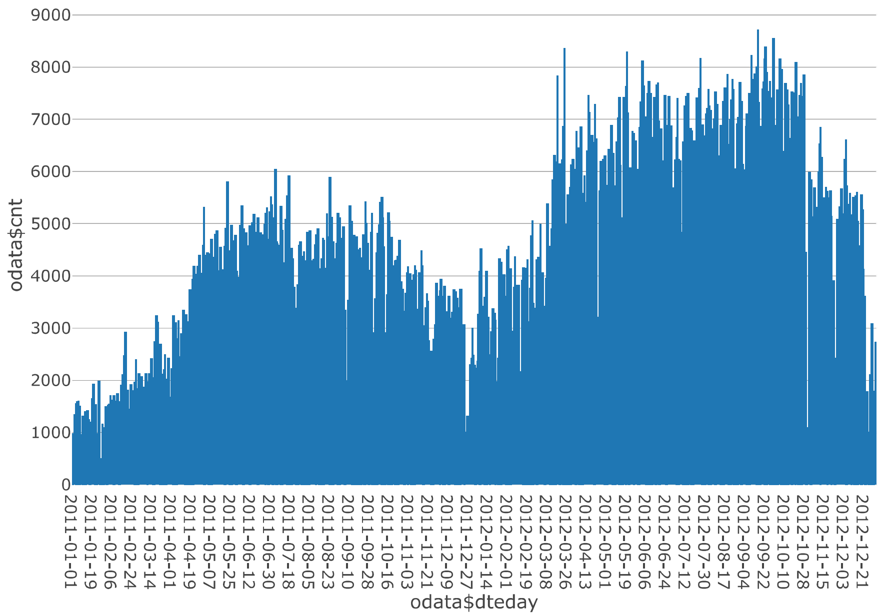
```
trial=odata[order(odata$cnt,decreasing = FALSE),]
print(trial[1,])
```

```
##     instant     dteday season yr mnth holiday weekday workingday
## 668     668 2012-10-29      4  1   10       0       1          1
##     weathersit temp  atemp  hum windspeed casual registered cnt
## 668          3 0.44 0.4394 0.88    0.3582      2         20  22
```

29/10/2012- Hurricane Sandy

->Step2:Visualising data before we proceed to start working with it Plot1: Total Count of rental bikes vs Date

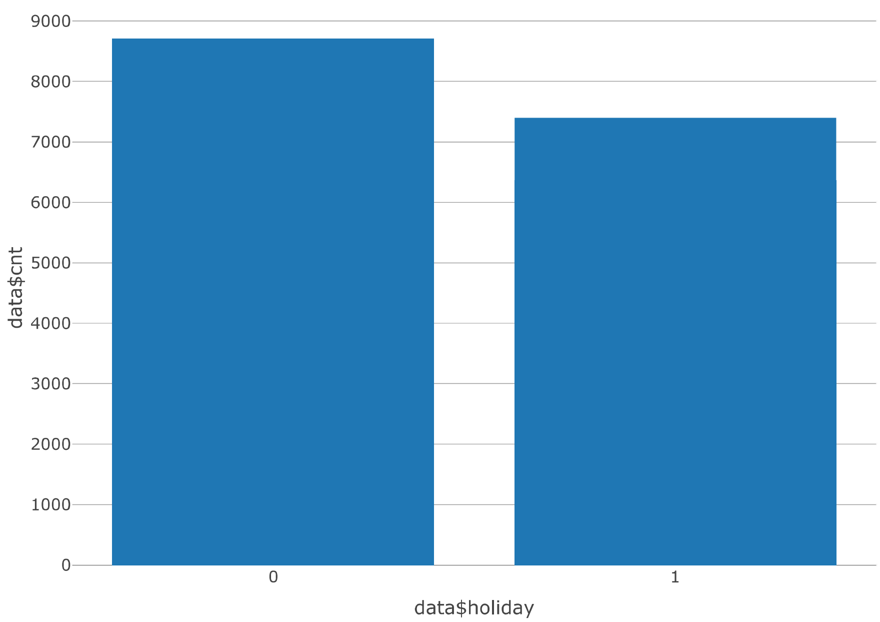
```
plot_ly(odata,x= ~odata$dteday,y= ~odata$cnt,type="bar")
```



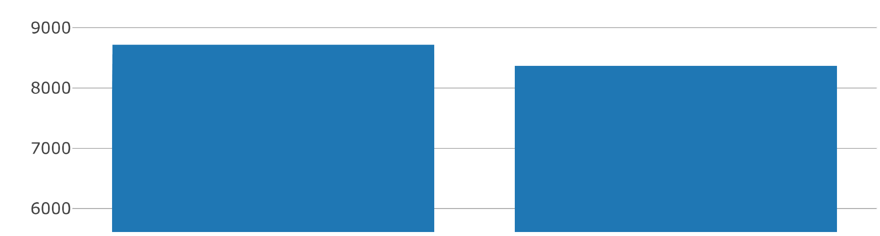
On some days bike count falls to a very low value,we need to analyse why? The count of bikes depends on what factors exactly?

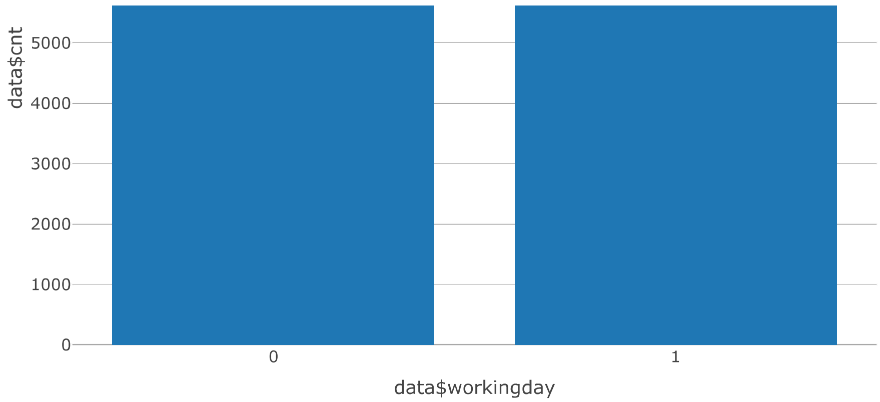
Plot 2: Total count of bikes vs whether day is a holiday or not

```
plot_ly(data,x= ~data$holiday,y= ~data$cnt,type="bar")#Holiday vs count
```




```
plot_ly(data,x= ~data$workingday,y= ~data$cnt,type="bar")#Whether working day vs count
```

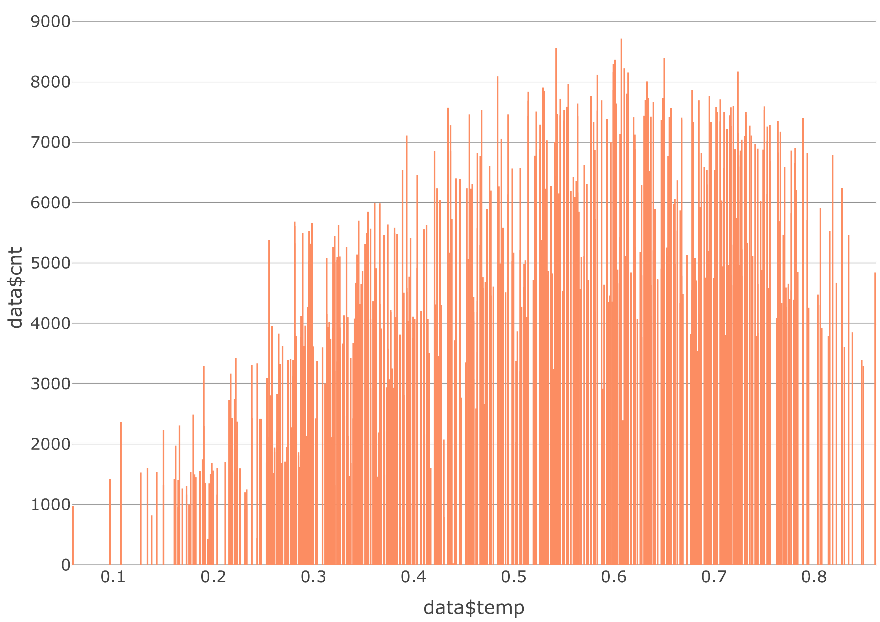




Plot 3:Total count of bikes vs Temperature on that day

```
plot_ly(data,x= ~data$temp,y= ~data$cnt,type="bar",color="red")
```

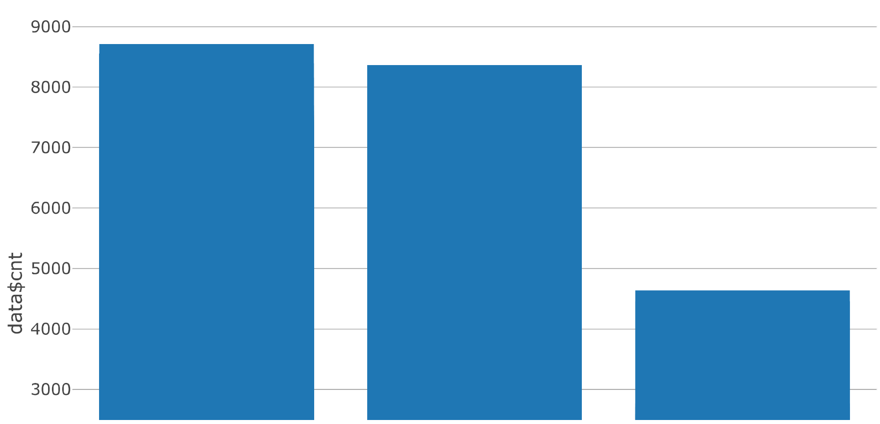
```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning requested palette with 3 different le
vels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning requested palette with 3 different le
vels
```

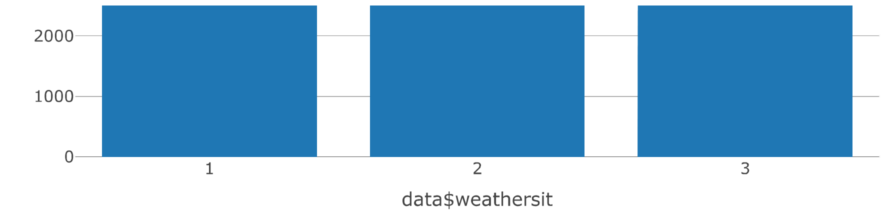


From the above plot one can conclude that when temperatures are low, lesser bikes are rented and while temperatures are moderate the bike count is at its peak

Plot 4:Total count of bikes vs Weather conditions

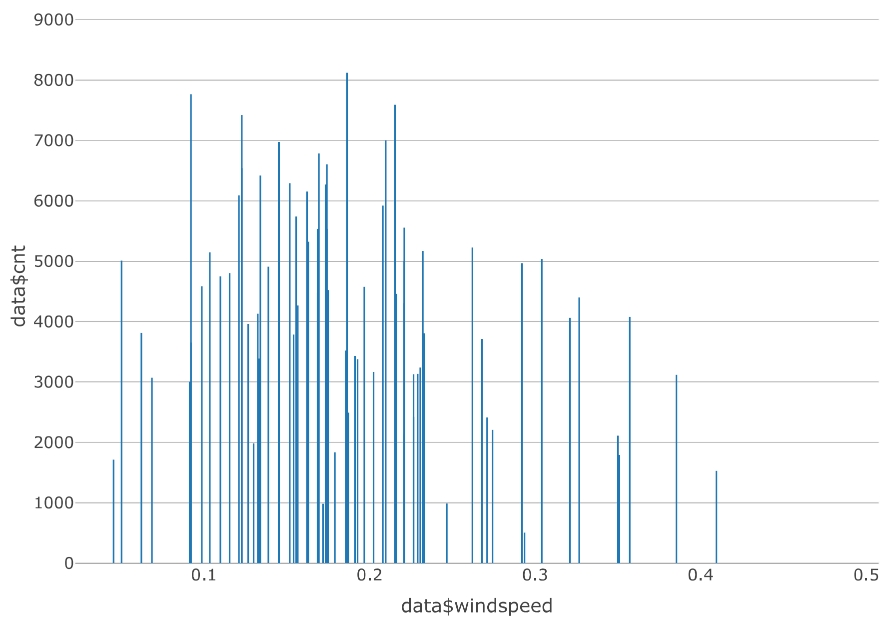
```
plot_ly(data,x= ~data$weathersit,y= ~data$cnt,type="bar")
```



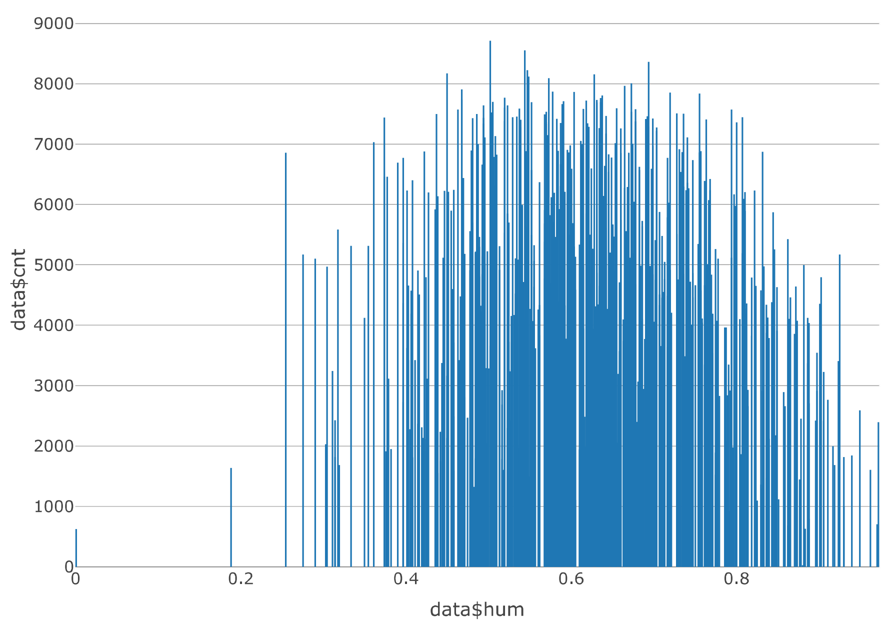


- 1: Clear, Few clouds, Partly cloudy, Partly cloudy - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds Some more additional plots:

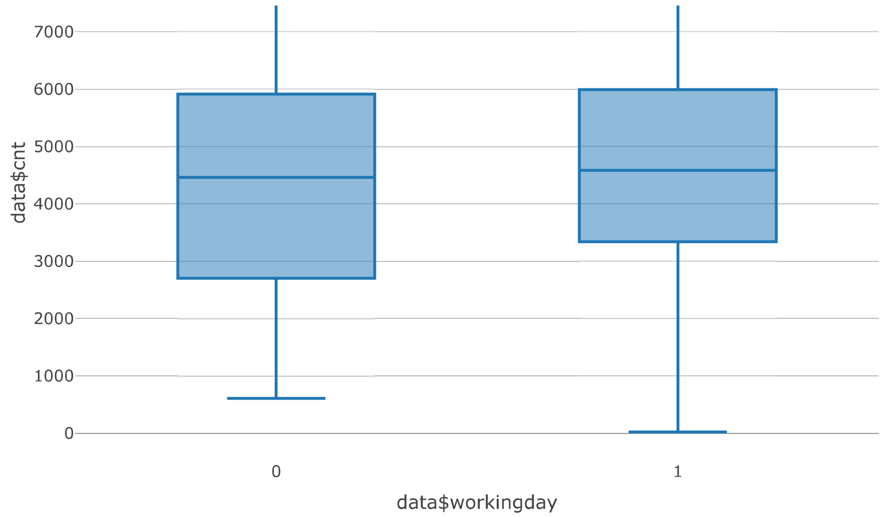
```
plot_ly(data,x=~data$windspeed,y=~data$cnt,type="bar")
```



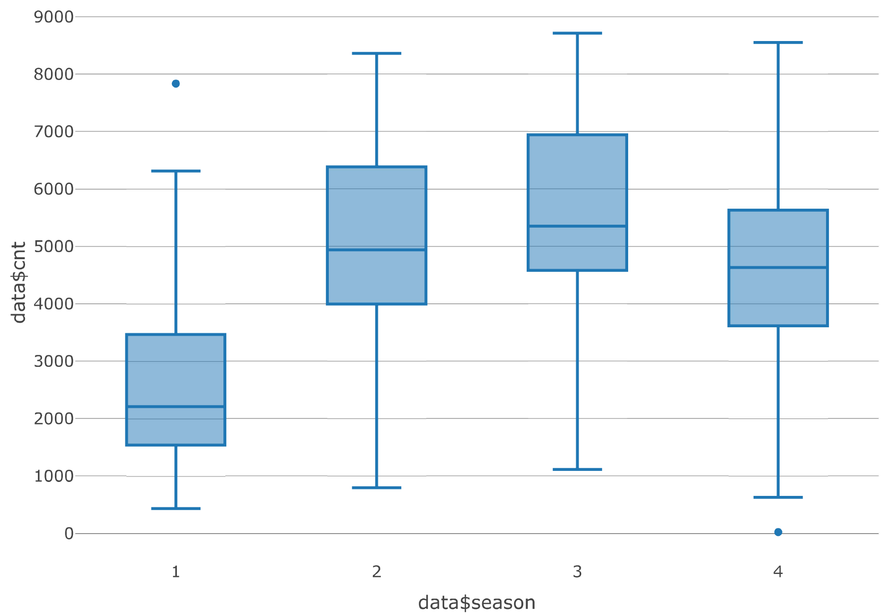

```
plot_ly(data,x=~data$hum,y=~data$cnt,type="bar")
```




```
plot_ly(data,x=~data$workingday,y=~data$cnt,type="box")
```




```
plot_ly(data,x=~data$season,y=~data$cnt,type="box")
```



season (1:springer, 2:summer, 3:fall, 4:winter)

==

Conclusion:From all the above plots we can come to the conclusion that the count of bikes is not dependent on just a single factor but is dependent on several factors. Hence we need to build a suitable model that takes all these factors into account and gives us the best predictions.

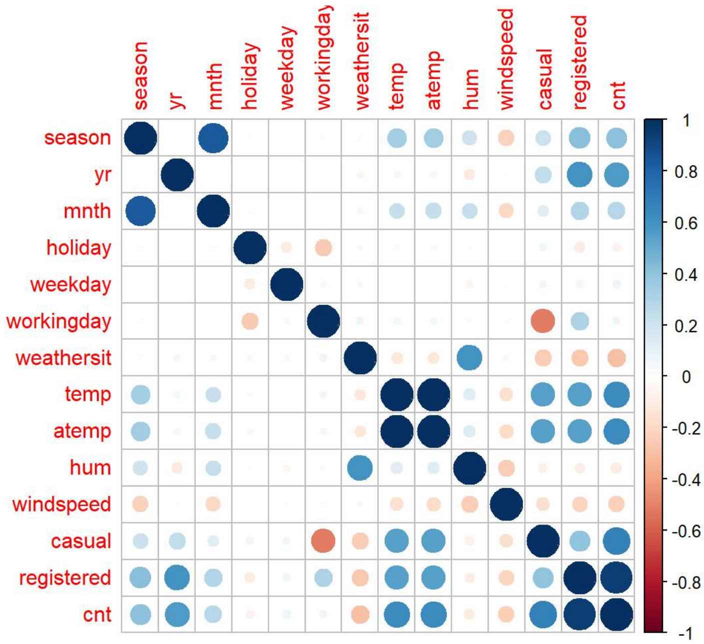
==

Step 3: Plotting the correlation matrix to remove redundant elements

```
print(data.frame(cor(data)))#Converting correlation matrix to a dataframe
```

```
##                  season           yr         mnth      holiday
## season      1.000000000 -0.001844343  0.831440114 -0.010536659
## yr         -0.001844343  1.000000000 -0.001792434  0.007954311
## mnth        0.831440114 -0.001792434  1.000000000  0.019190895
## holiday    -0.010536659  0.007954311  0.019190895  1.000000000
## weekday    -0.003079881 -0.005460765  0.009509313 -0.101960269
## workingday  0.012484963 -0.002012621 -0.005900951 -0.253022700
## weathersit  0.019211028 -0.048726541  0.043528098 -0.034626841
## temp        0.334314856  0.047603572  0.220205335 -0.028555535
## atemp       0.342875613  0.046106149  0.227458630 -0.032506692
## hum         0.205444765 -0.110651045  0.222203691 -0.015937479
## windspeed  -0.229046337 -0.011817060 -0.207501752  0.006291507
## casual      0.210399165  0.248545664  0.123005889  0.054274203
## registered  0.411623051  0.594248168  0.293487830 -0.108744863
## cnt         0.406100371  0.566709708  0.279977112 -0.068347716
##                  weekday   workingday  weathersit          temp
## season     -0.0030798813  0.012484963  0.01921103  0.3343148564
## yr         -0.0054607652 -0.002012621 -0.04872654  0.0476035719
## mnth        0.0095093129 -0.005900951  0.04352810  0.2202053352
## holiday    -0.1019602689 -0.253022700 -0.03462684 -0.0285555350
## weekday     1.0000000000  0.035789674  0.03108747 -0.0001699624
## workingday  0.0357896736  1.000000000  0.06120043  0.0526598102
## weathersit  0.0310874694  0.061200430  1.00000000 -0.1206022365
## temp       -0.0001699624  0.052659810 -0.12060224  1.0000000000
## atemp      -0.0075371318  0.052182275 -0.12158335  0.9917015532
## hum        -0.0522321004  0.024327046  0.59104460  0.1269629390
## windspeed   0.0142821241 -0.018796487  0.03951106 -0.1579441204
## casual      0.0599226375 -0.518044191 -0.24735300  0.5432846617
## registered  0.0573674440  0.303907117 -0.26038771  0.5400119662
## cnt         0.0674434124  0.061156063 -0.29739124  0.6274940090
##                   atemp         hum    windspeed      casual  registered
## season      0.342875613  0.20544476 -0.229046337  0.21039916  0.41162305
## yr          0.046106149 -0.11065104 -0.011817060  0.24854566  0.59424817
## mnth        0.227458630  0.22220369 -0.207501752  0.12300589  0.29348783
## holiday    -0.032506692 -0.01593748  0.006291507  0.05427420 -0.10874486
## weekday    -0.007537132 -0.05223210  0.014282124  0.05992264  0.05736744
## workingday  0.052182275  0.02432705 -0.018796487 -0.51804419  0.30390712
## weathersit -0.121583354  0.59104460  0.039511059 -0.24735300 -0.26038771
## temp        0.991701553  0.12696294 -0.157944120  0.54328466  0.54001197
## atemp       1.000000000  0.13998806 -0.183642967  0.54386369  0.54419176
## hum         0.139988060  1.00000000 -0.248489099 -0.07700788 -0.09108860
## windspeed  -0.183642967 -0.24848910  1.000000000 -0.16761335 -0.21744898
## casual      0.543863690 -0.07700788 -0.167613349  1.00000000  0.39528245
## registered  0.544191758 -0.09108860 -0.217448981  0.39528245  1.00000000
## cnt         0.631065700 -0.10065856 -0.234544997  0.67280443  0.94551692
##                    cnt
## season      0.40610037
## yr          0.56670971
## mnth        0.27997711
## holiday    -0.06834772
## weekday     0.06744341
## workingday  0.06115606
## weathersit -0.29739124
## temp        0.62749401
## atemp       0.63106570
## hum        -0.10065856
## windspeed  -0.23454500
## casual      0.67280443
## registered  0.94551692
## cnt         1.00000000
```

```
corrplot(cor(data))
```



Lets set our threshold correlation value as 0.8 or 80% and hence decide what all parameters we may drop:

mnth — season —->0.831440114 temp — atemp —->0.9917015532

We cannot consider correlation of registered and cnt as cnt is the dependent variable that we need to predict.

Conclusion: We shall drop attributes mnth and atemp and not consider them in building our regressor

```
data=data[,-3]
```

```
data=data[,-8]
```

==

Step 4: Building the regressor model We shall remove columns of casual and registered users on a particular day as these are factors that can be found only after the day has passed only.

Removing Casual User Column:

```
data=data[,-10]
```

Removing Registered User Column:

```
data=data[,-10]
```

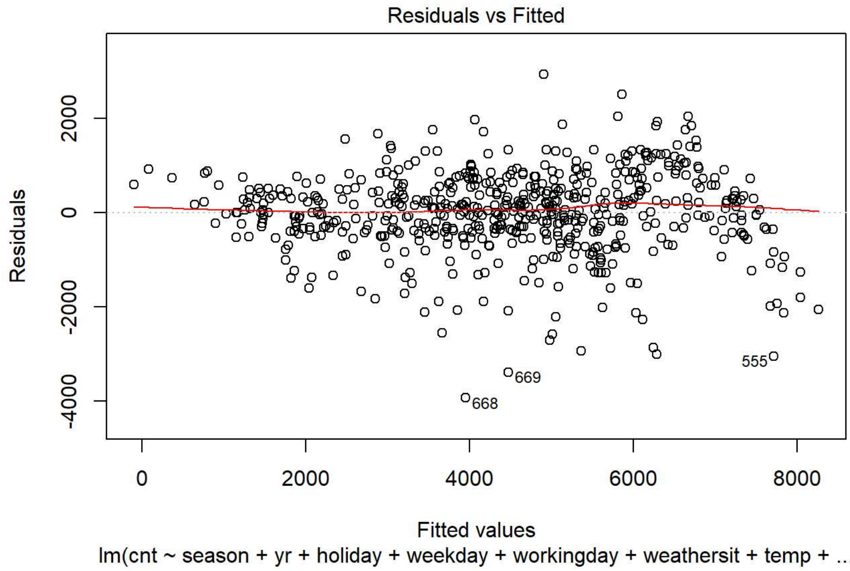
```
regressor=lm(formula = cnt~season+yr+holiday+weekday+workingday+weathersit+temp+hum+windspeed,data=training_set)
#regressor=lm(formula = cnt~registered,data=training_set)
summary(regressor)
```

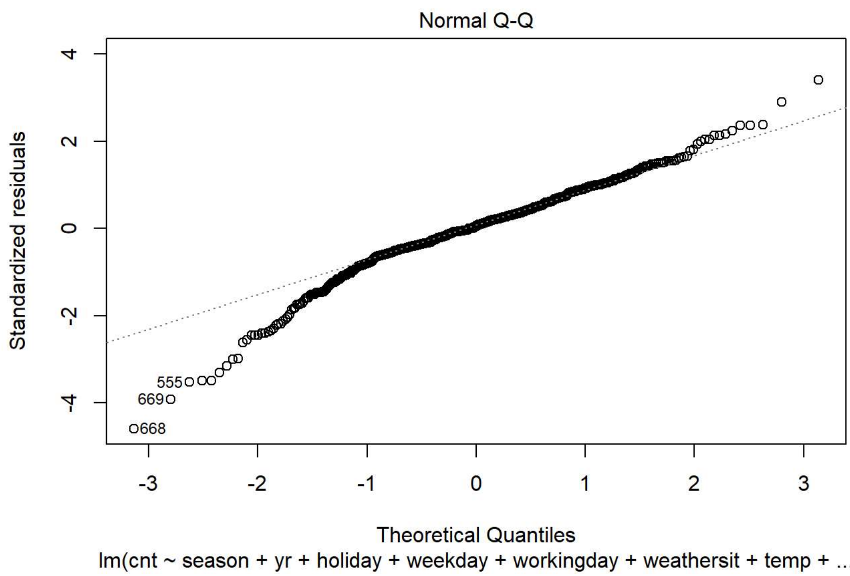
```
##
## Call:
## lm(formula = cnt ~ season + yr + holiday + weekday + workingday +
##     weathersit + temp + hum + windspeed, data = training_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3930.4  -400.4    52.1   524.8  2933.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1654.15     258.60   6.397 3.30e-10 ***
## season        385.58      35.33  10.913  < 2e-16 ***
## yr           2027.55      72.86  27.827  < 2e-16 ***
## holiday      -598.91     217.40  -2.755 0.006058 **
## weekday        66.69      18.09   3.687 0.000249 ***
## workingday    117.23      81.16   1.444 0.149167
## weathersit   -631.66      85.40  -7.396 4.99e-13 ***
## temp         5413.28     216.24  25.033  < 2e-16 ***
## hum         -1001.89     341.19  -2.936 0.003453 **
## windspeed   -2961.27     506.93  -5.842 8.66e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 870.7 on 574 degrees of freedom
## Multiple R-squared:  0.7966, Adjusted R-squared:  0.7934
## F-statistic: 249.8 on 9 and 574 DF,  p-value: < 2.2e-16
```

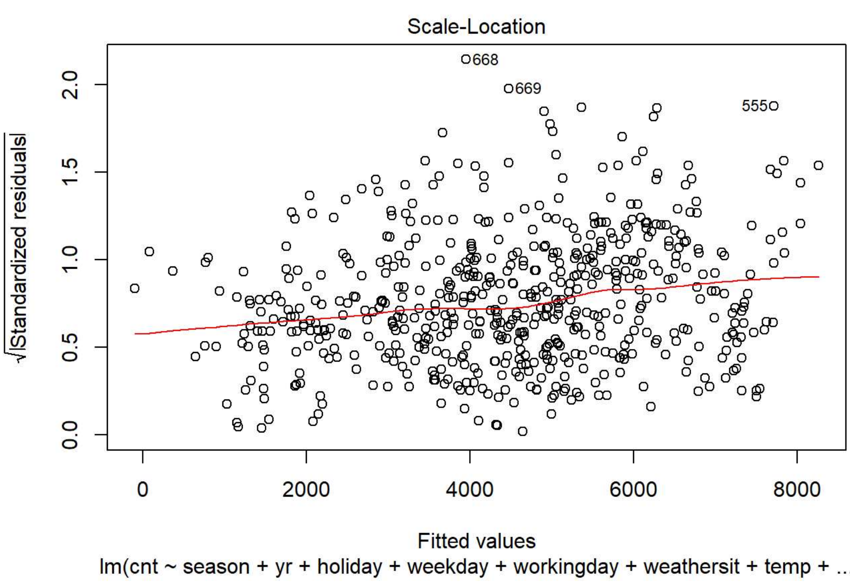
```
y_pred=predict(regressor,newdata=test_set)
rmse(test_set$cnt,y_pred)
```

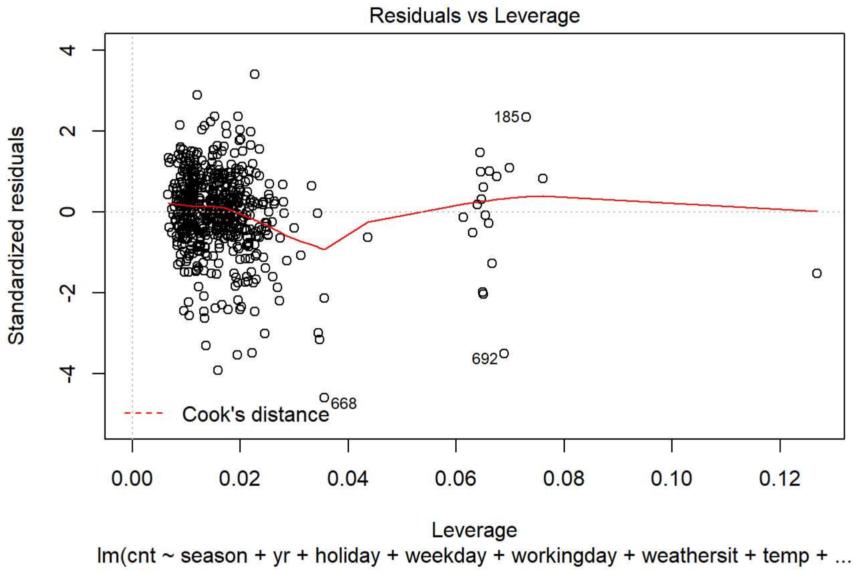
```
## [1] 909.3113
```

```
plot(regressor)
```

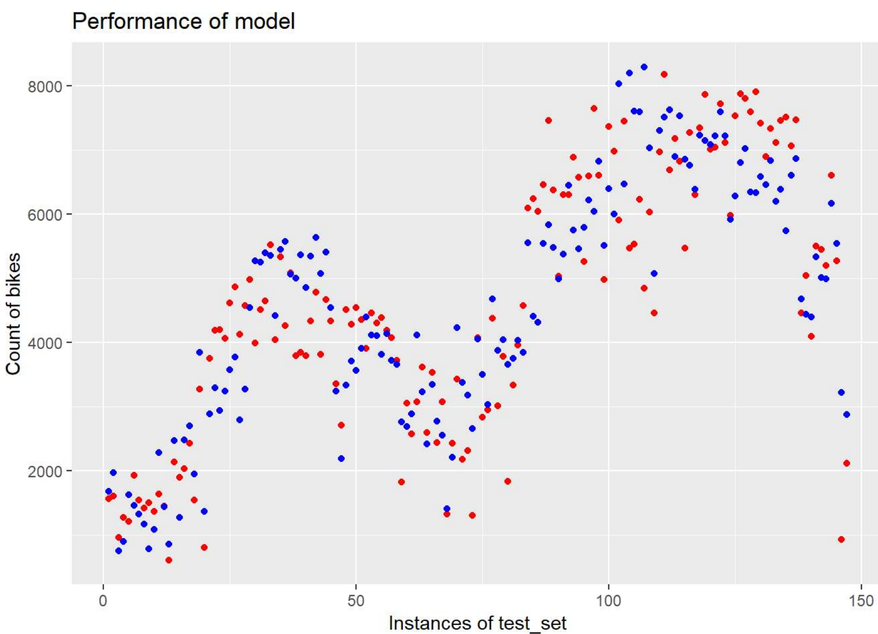

Residuals vs Fitted
Residuals
2000
0
-2000
-4000
555
669
668
0
2000
4000
6000
8000
Fitted values
lm(cnt ~ season + yr + holiday + weekday + workingday + weathersit + temp + ...


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
555
669
668
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(cnt ~ season + yr + holiday + weekday + workingday + weathersit + temp + ...


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
668
669
555
0
2000
4000
6000
8000
Fitted values
lm(cnt ~ season + yr + holiday + weekday + workingday + weathersit + temp + ...




```
library(ggplot2)
ggplot() +
  geom_point(aes(x = c(1:147), y = test_set$cnt),
             colour = 'red') +
  geom_point(aes(x = c(1:147), y = predict(regressor, newdata = test_set)),
            colour = 'blue') +
  ggtitle('Performance of model') +
  xlab('Instances of test_set') +
  ylab('Count of bikes')
```



The blue dots represnt what our model predicted for each instance of the test_set and the red dots represent what the actual values were for the test_set instances

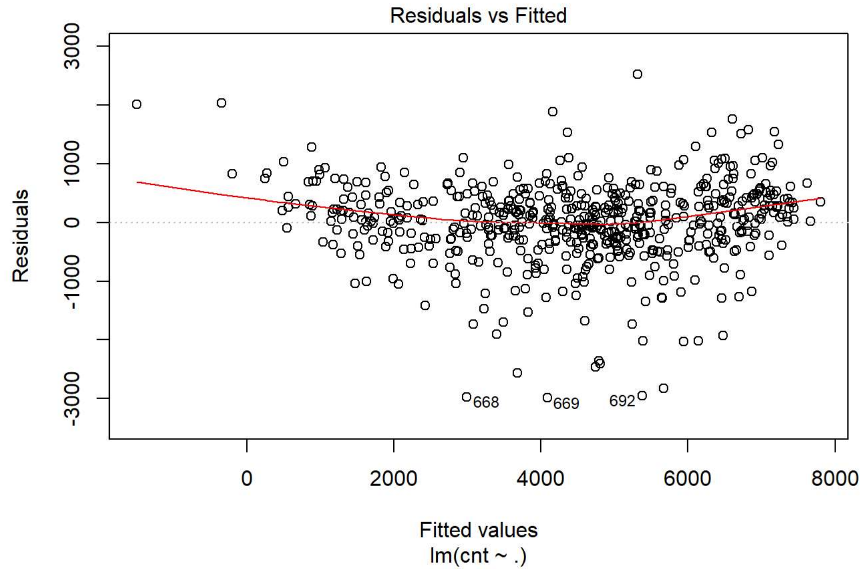
```
data$temp2=data$temp^2
data$temp3=data$temp^3
data$temp4=data$temp^4
data$temp5=data$temp^5
data$temp6=data$temp^6
data$temp6=data$temp^7
data$weathersit2=data$weathersit^2
data$season2=data$season^2
data$season3=data$season^3
training_set2=subset(data,split==TRUE)
test_set2=subset(data,split==FALSE)
poly_reg=lm(formula=cnt ~.,data=training_set2)
summary(poly_reg)
```

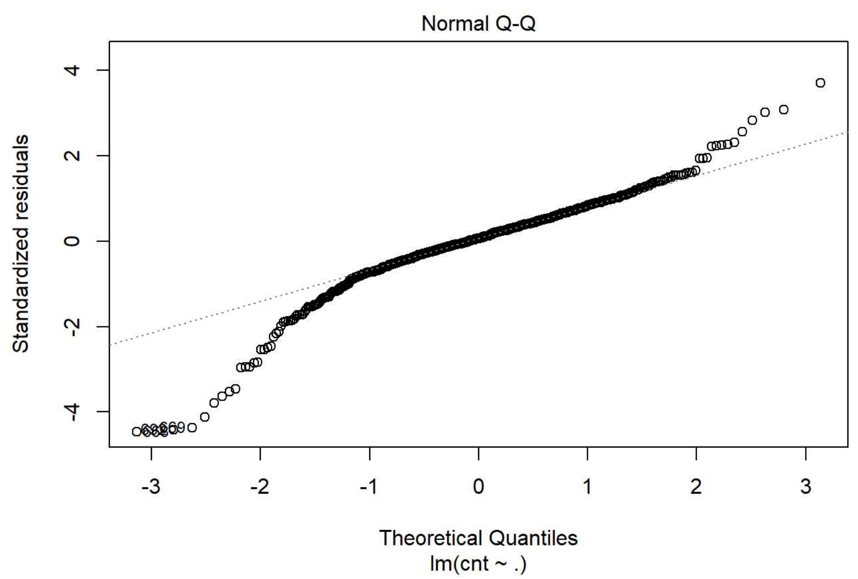
```
##
## Call:
## lm(formula = cnt ~ ., data = training_set2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2990.75  -296.11    45.68   383.35  2521.91
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.885e+03  1.929e+03  -1.496 0.135184
## season       2.771e+03  8.695e+02   3.187 0.001518 **
## yr           1.947e+03  5.869e+01  33.166  < 2e-16 ***
## holiday     -4.516e+02  1.740e+02  -2.595 0.009700 **
## weekday      7.445e+01  1.445e+01   5.153 3.55e-07 ***
## workingday   8.669e+01  6.508e+01   1.332 0.183404
## weathersit   1.217e+03  3.485e+02   3.493 0.000515 ***
## temp         5.406e+04  3.282e+04   1.647 0.100079
## hum         -2.025e+03  2.808e+02  -7.214 1.76e-12 ***
## windspeed   -3.493e+03  4.120e+02  -8.478  < 2e-16 ***
## temp2       -3.755e+05  2.215e+05  -1.695 0.090540 .
## temp3        1.276e+06  7.153e+05   1.783 0.075076 .
## temp4       -2.016e+06  1.145e+06  -1.761 0.078731 .
## temp5        1.308e+06  7.743e+05   1.690 0.091666 .
## temp6       -2.558e+05  1.609e+05  -1.590 0.112327
## weathersit2 -5.522e+02  1.031e+02  -5.356 1.24e-07 ***
## season2     -9.571e+02  4.028e+02  -2.376 0.017834 *
## season3      1.160e+02  5.585e+01   2.077 0.038268 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 692.6 on 566 degrees of freedom
## Multiple R-squared:  0.8731, Adjusted R-squared:  0.8693
## F-statistic:   229 on 17 and 566 DF,  p-value: < 2.2e-16
```

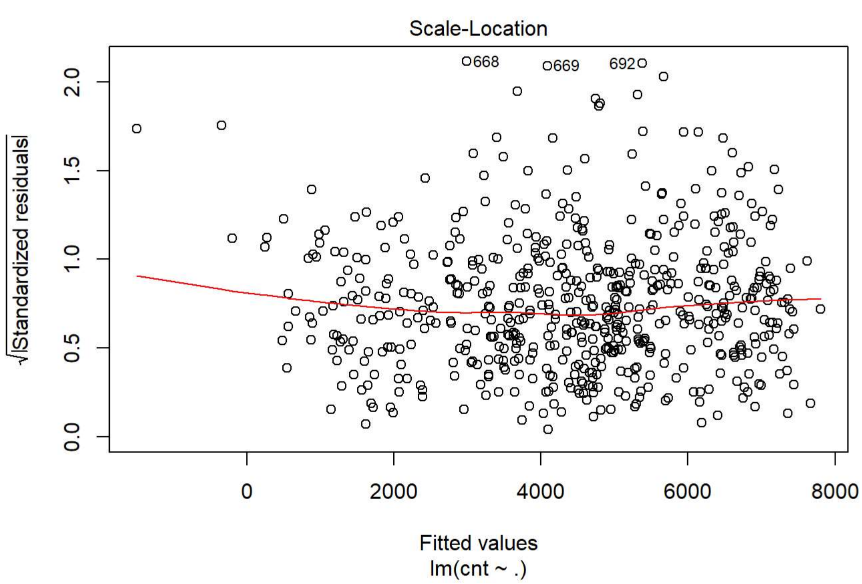
```
y_pred2=predict(poly_reg,newdata=test_set2)
rmse(test_set2$cnt,y_pred2)
```

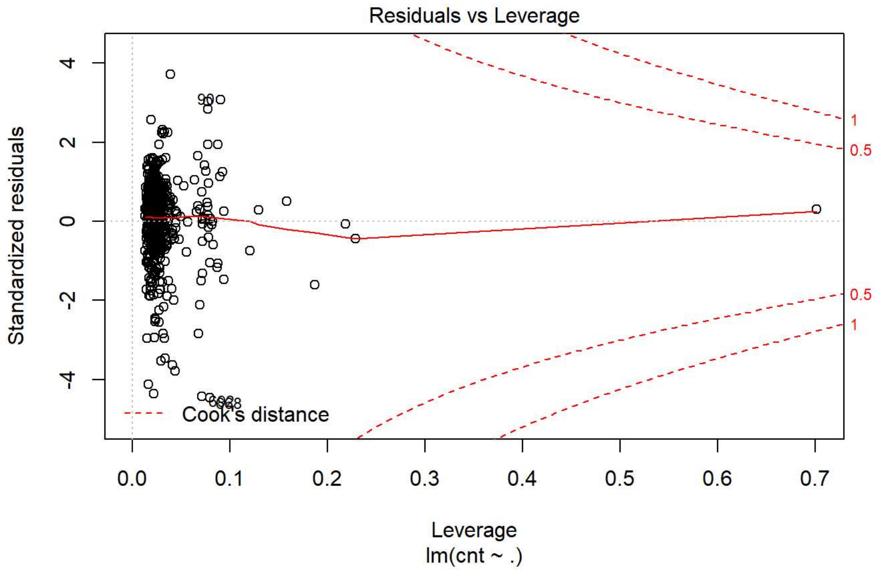
```
## [1] 582.8019
```

```
plot(poly_reg)
```


Residuals vs Fitted
Residuals
3000
1000
0
-1000
-3000
668
669
692
0
2000
4000
6000
8000
Fitted values
lm(cnt ~ .)


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(cnt ~ .)


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
668
669
692
0
2000
4000
6000
8000
Fitted values
lm(cnt ~ .)




```
ggplot() +
  geom_point(aes(x = c(1:147), y = test_set2$cnt),
             colour = 'red') +
  geom_point(aes(x = c(1:147), y = predict(poly_reg, newdata = test_set2)),
            colour = 'blue') +
  ggtitle('Performance of model') +
  xlab('Instances of test_set') +
  ylab('Count of bikes')
```