Analytical Methods-I

Group - 1

3/24/2020

Part 1: Here we read the data file and setting names that convey apt description.

```
download.file("https://archive.ics.uci.edu/ml/machine-learning-databases/0027
5/Bike-Sharing-Dataset.zip","Bike.zip")

day <- read.table(unz("Bike.zip", "day.csv"), header=T, quote="\"", sep=",")
```

Libraries we will need in this project

```
library(DataExplorer)
library(corrplot)

library(ggplot2)
library(forcats)
library(ade4)
library(caret)

library(tinytex)
```

Understanding the data

```
#Dimention of the data
dim(day)

## [1] 731  16

#Variable type Identification
str(day)

## 'data.frame':    731 obs. of  16 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4
5 6 7 8 9 10 ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
```

```
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

```r
#Names of data variables
names(day)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
##  [6] "holiday"    "weekday"    "workingday" "weathersit" "temp"
## [11] "atemp"      "hum"        "windspeed"  "casual"     "registered"
## [16] "cnt"
```

```r
#Summary statistics
summary(day)
```

```
##     instant             dteday        season            yr
##  Min.   :  1.0   2011-01-01:  1   Min.   :1.000   Min.   :0.0000
##  1st Qu.:183.5   2011-01-02:  1   1st Qu.:2.000   1st Qu.:0.0000
##  Median :366.0   2011-01-03:  1   Median :3.000   Median :1.0000
##  Mean   :366.0   2011-01-04:  1   Mean   :2.497   Mean   :0.5007
##  3rd Qu.:548.5   2011-01-05:  1   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :731.0   2011-01-06:  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
##
```

#standardise the temp and atemp values which were normalized in the dataset.

```r
day$temp<- day$temp*41
day$atemp<- day$atemp*50
day$hum<- day$hum*100
day$windspeed<-day$windspeed*67
```

Find missing values in data if any.

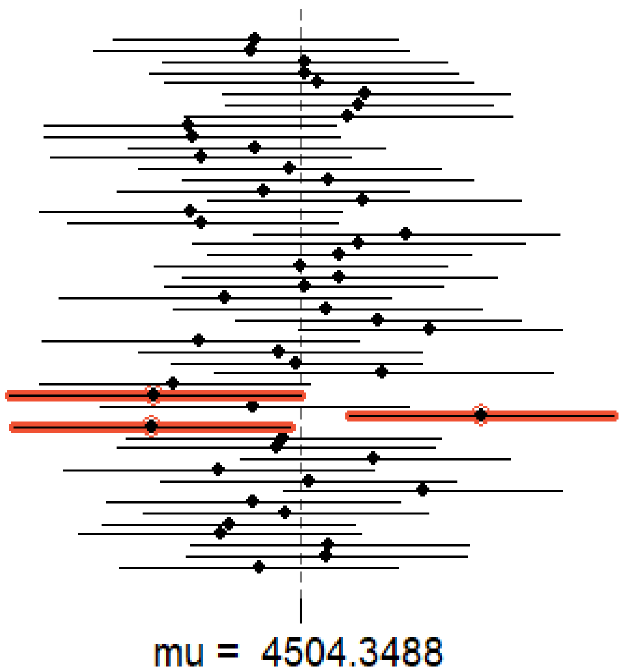
```r
table(is.na(day))

##
## FALSE
## 11696
```

Above we can see that it returned no missing (TRUE) value in the data.

Inference

```r
#creating contains function
contains <- function(lo,hi,m){
        if(m>= lo & m <= hi) return(TRUE)
        else return(FALSE)
}

#creating plot_ci function
plot_ci <- function(lo, hi, m){
        par(mar=c(2, 1, 1, 1), mgp=c(2.7, 0.7, 0))
        k <- 50
        ci.max <- max(rowSums(matrix(c(-1*lo,hi),ncol=2)))

        xR <- m + ci.max*c(-1, 1)
        yR <- c(0, 41*k/40)

        plot(xR, yR, type='n', xlab='', ylab='', axes=FALSE)
        abline(v=m, lty=2, col='#00000088')
        axis(1, at=m, paste("mu = ",round(m,4)), cex.axis=1.15)
        #axis(2)
        for(i in 1:k){
                x <- mean(c(hi[i],lo[i]))
                ci <- c(lo[i],hi[i])
                if(contains(lo[i],hi[i],m)==FALSE){
                        col <- "#F05133"
                        points(x, i, cex=1.4, col=col)
                        #         points(x, i, pch=20, cex=1.2, col=col)
                        lines(ci, rep(i, 2), col=col, lwd=5)
                }
                col <- 1
                points(x, i, pch=20, cex=1.2, col=col)
                lines(ci, rep(i, 2), col=col)
        }
}
```

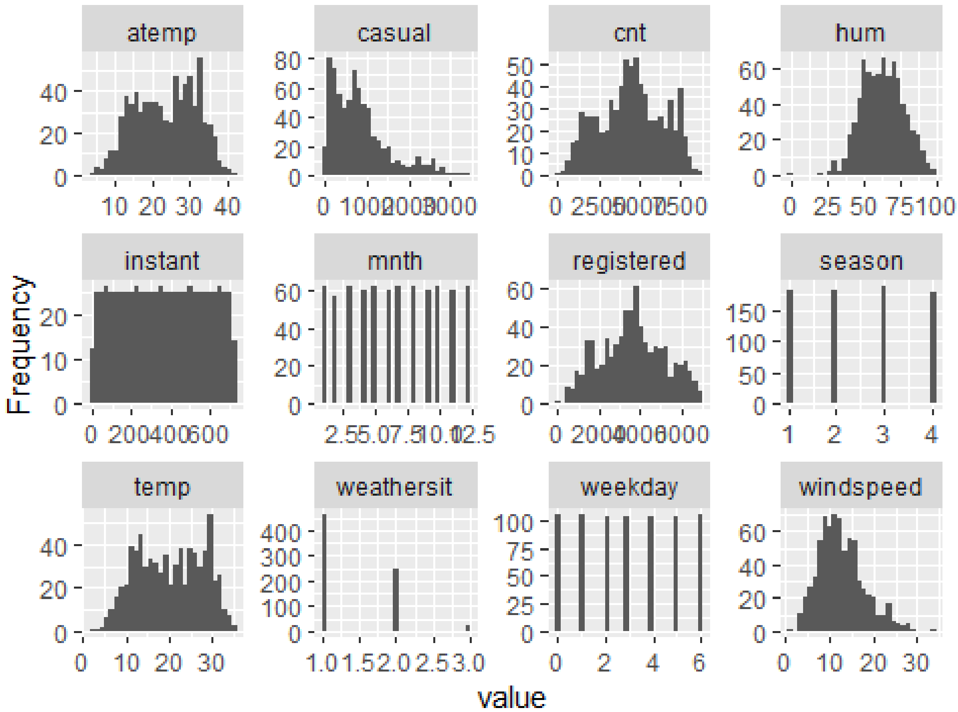
```r
#we will use CI to find mean daily bike rentals for these 2 years using sampl
e mean

set.seed(123)
population <- day$cnt
samp <-sample(day$cnt,60)
mean(samp)

## [1] 4533.633

sample_mean <- mean(samp)
se <- sd(samp) / sqrt(60)
lower <- sample_mean - 1.96 * se
upper <- sample_mean + 1.96 * se
c(lower, upper)

## [1] 4068.163 4999.103

samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60

for(i in 1:50){
        samp <- sample(population, n) # obtain a sample of size n = 60 from t
he population
        samp_mean[i] <- mean(samp)    # save sample mean in ith element of sa
mp_mean
        samp_sd[i] <- sd(samp)        # save sample sd in ith element of samp
_sd
}

lower_vector <- samp_mean - 1.96 * samp_sd / sqrt(n)
upper_vector <- samp_mean + 1.96 * samp_sd / sqrt(n)

c(lower_vector[20], upper_vector[20])

## [1] 3790.178 4708.622

par(mfrow = c(1, 1))
plot_ci(lower_vector, upper_vector, mean(population))

mean(day$cnt)

## [1] 4504.349
```



We used loop function to take 50 random samples of size 60 from the available daily data of 731 days. 47 of the resulting confidence intervals contain the true average number of exclusive relationships that mean 95% proportion of confidence intervals includes the true mean.

Understand the distribution of numerical variables and generate a frequency table for numeric variables.

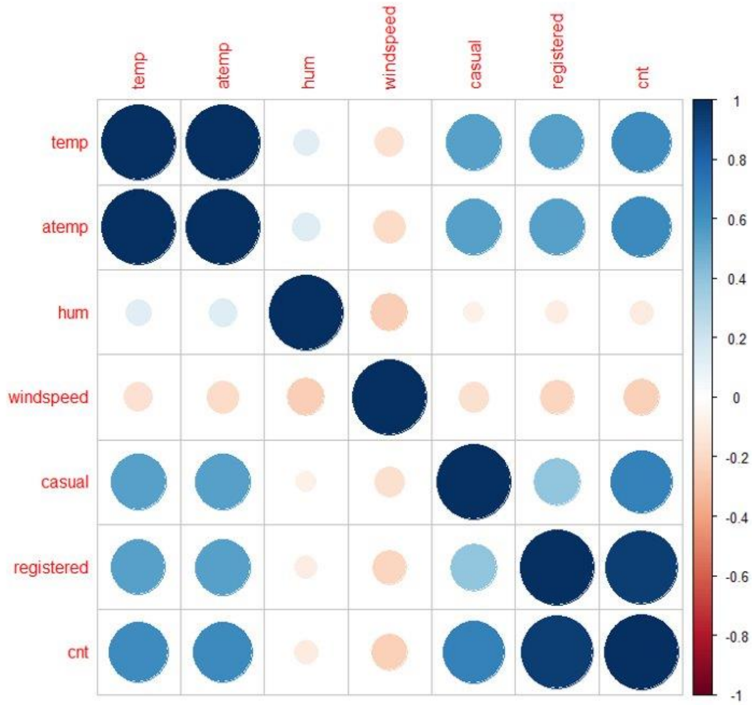
```
plot_histogram(day)
```



Few inferences can be drawn by looking at these histograms: Season has four categories of almost equal distribution. Weather 1 has higher contribution than 2 and 3. Most of the numeric variables are normally distributed.

We will see the Correlation of response variable with explanatory varibales

```
nonums <- unlist(lapply(day, is.numeric))
nums<-day[,nonums]

par(mfrow=c(1,1))
corrplot(cor(nums))
```



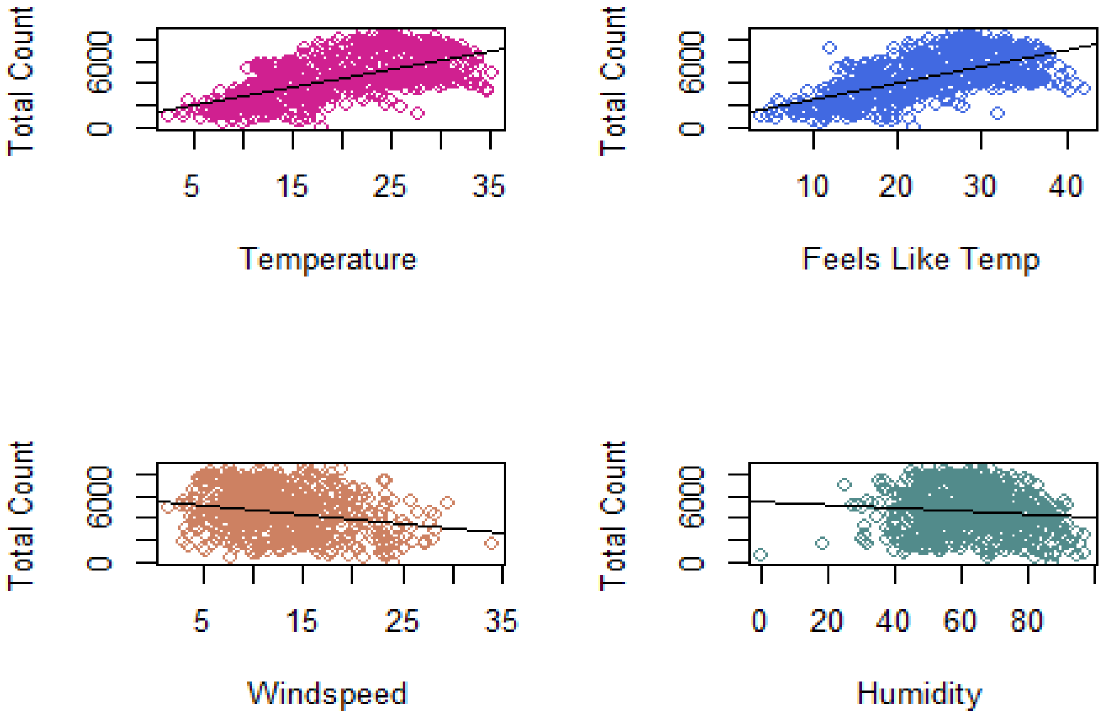

```r
symnum(cor(nums))

##            i s y m hl wk wr wt t a hm wn cs r cn
## instant    1
## season     . 1
## yr         +   1
## mnth       . +   1
## holiday            1
## weekday               1
## workingday               1
## weathersit                  1
## temp         .                 1
## atemp        .                 B 1
## hum                           .      1
## windspeed                               1
## casual                      .     . .       1
## registered , . .         .     . .       .  1
## cnt        , . .               , ,       ,  * 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

We can say registered and casual variables are highly correlated with target variable because target variables is sum of these two variable. Temp, atemp, humidity, windspeed can be good predictors. Humidity and windspeed have negative correlation with counts. Temp and atemp both are highly correlated with

each other that cause multicollinearity so we will try results in our model and remove one of these two variables.

We try and find the relationship between the count (dependent variable) with numeric independent variables (temperature, feels like temperature, humidity and windspeed)

```
par(mfrow=c(2,2))

plot(day$cnt~day$temp ,type = 'p', col= 'violetred', xlab = 'Temperature', yl
ab = 'Total Count')
abline(lm(day$cnt~day$temp))

plot(day$cnt~day$atemp ,type = 'p', col= 'royalblue', xlab = 'Feels Like Temp
', ylab = 'Total Count')
abline(lm(day$cnt~day$atemp))

plot(day$cnt~day$windspeed ,type = 'p', col= 'lightsalmon3', xlab = 'Windspee
d', ylab = 'Total Count')
abline(lm(day$cnt~day$windspeed))

plot(day$cnt~day$hum ,type = 'p', col= 'darkslategray4', xlab = 'Humidity', y
lab = 'Total Count')
abline(lm(day$cnt~day$hum))
```



The graph shows that mostly people prefer to rent bike on good weather which includes high temperature with less humidity and windspeed.

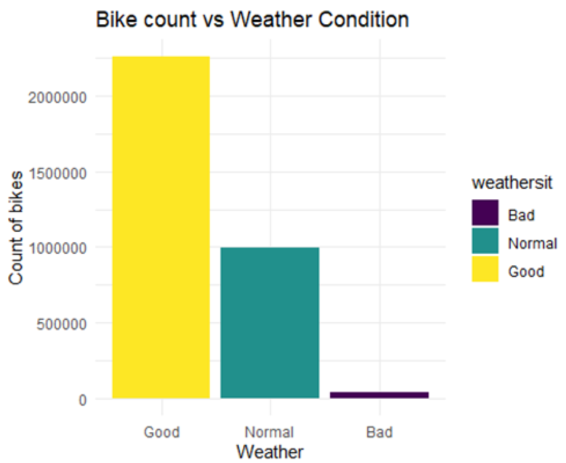
Transform discrete variables into factor variables(season, weather, holiday, workingday, weekday)

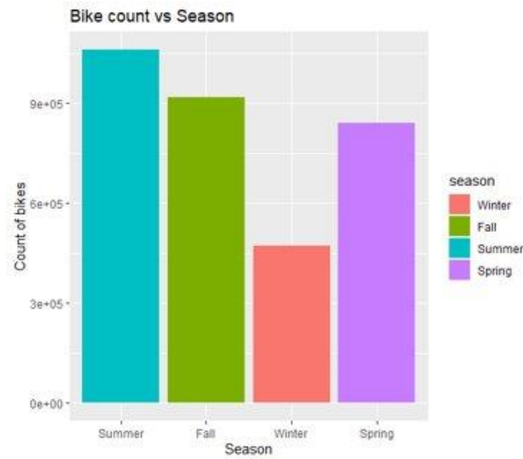
```
day$season<- factor(day$season
                    ,levels = c(1,2,3,4)
                    ,labels = c ("Spring", "Summer", "Fall", "Winter")
)


day$weathersit<-factor(day$weather
                       ,levels = c(3,2,1)
                       ,labels = c("Bad", "Normal", "Good")
                       ,ordered = TRUE)

day$holiday<- factor(day$holiday
                     ,levels = c(0,1)
                     ,labels = c("noholiday", "holiday")
)

day$workingday<-factor(day$workingday
```

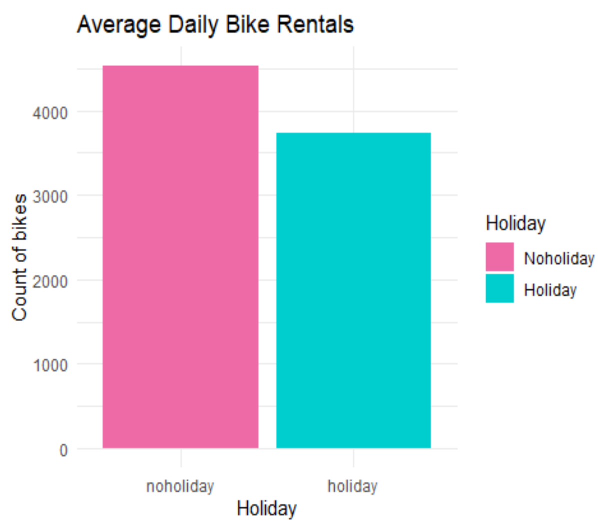
```
                       ,levels = c(0,1)
                       ,labels = c("nonworking", "working")
)

day$weekday<-factor(day$weekday
                    ,levels = c(0,1,2,3,4,5,6)
                    ,labels = c("sun", "mon","tue","wed","thur","fri","sat")
)
```

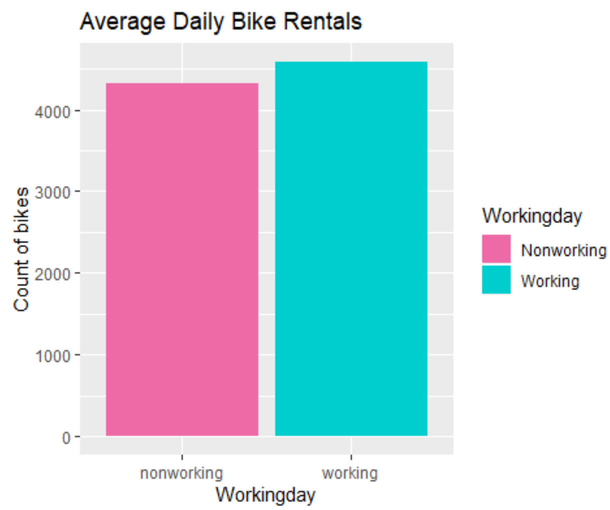
Now we will try and find the relationship between count(independent variable) and categorical dependent variables.

```
ggplot(day, aes(x = fct_infreq(season), y = cnt, fill = season))+
  geom_bar(stat = "identity")+
  labs(title = "Bike count vs Season",
       x = "Season", y = "Count of bikes") +
  theme(legend.position = "right")



ggplot(day, aes(x=fct_infreq(weathersit), y=cnt, fill=weathersit)) +
  geom_bar(stat="identity")+theme_minimal()+
  labs(title = "Bike count vs Weather Condition",
       x = "Weather", y = "Count of bikes") +
  theme(legend.position = "right")



Avg_Bike_Rental1 <- day %>% group_by(holiday) %>%
  summarise(mean = mean(cnt))

ggplot(Avg_Bike_Rental1, aes(x = holiday, y = mean, fill = holiday))+
  geom_bar(stat = "identity")+theme_minimal()+
labs(title = "Average Daily Bike Rentals",
      x = "Holiday", y = "Count of bikes") +
  scale_fill_manual(name = "Holiday",
                    labels = c("Noholiday", "Holiday"),   values = c("hotpink
2", "cyan3"))



Avg_Bike_Rental <- day %>% group_by(workingday) %>%
  summarise(mean = mean(cnt))

ggplot(Avg_Bike_Rental, aes(x = workingday, y = mean, fill = workingday)) +
  geom_bar(stat = "identity", position = "dodge")+
  labs(title = "Average Daily Bike Rentals",
       x = "Workingday", y = "Count of bikes") +
  scale_fill_manual(name = "Workingday",
                    labels = c("Nonworking", "Working"),   values = c("hotpin
k2", "cyan3"))
```





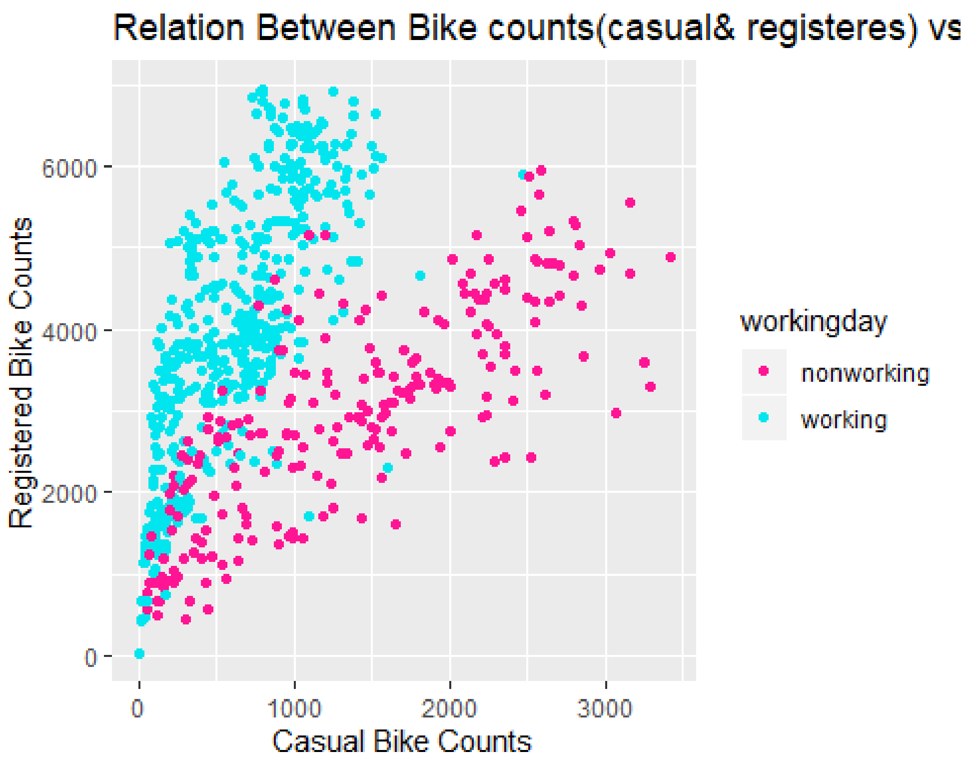




The above graphs show people like to ride in good weather, least bike users in winter season and highest bike users in summer. The average numbers of bike show that there is very less impact of working day and holiday on counts of bikes.

Here we try to find the relationship between working/nonworking and casual/registered

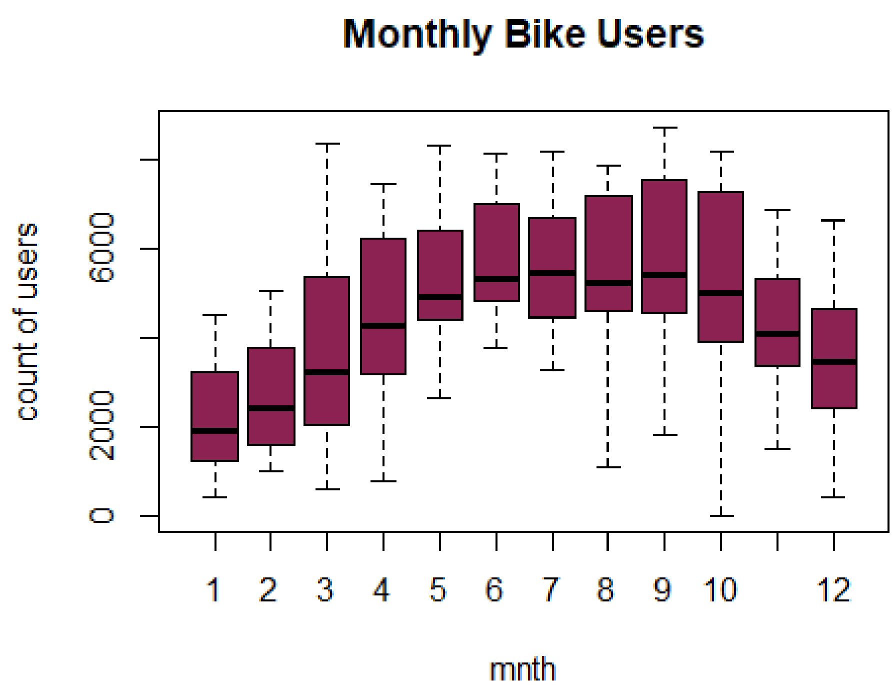
```
ggplot(day, aes(x = casual, y = registered, color = workingday))+
  geom_point()+
  labs(title = "Relation Between Bike counts(casual& registeres) vs Working, Non working")+
   scale_color_manual(values=c("deeppink", "turquoise2")) +
  xlab("Casual Bike Counts") +
  ylab("Registered Bike Counts")
```



The graph shows that mostly working people are registered and use bikes mainly on weekdays. On the other hand, mostly non-working people are casual bikers and prefer to ride on weekends and holidays.

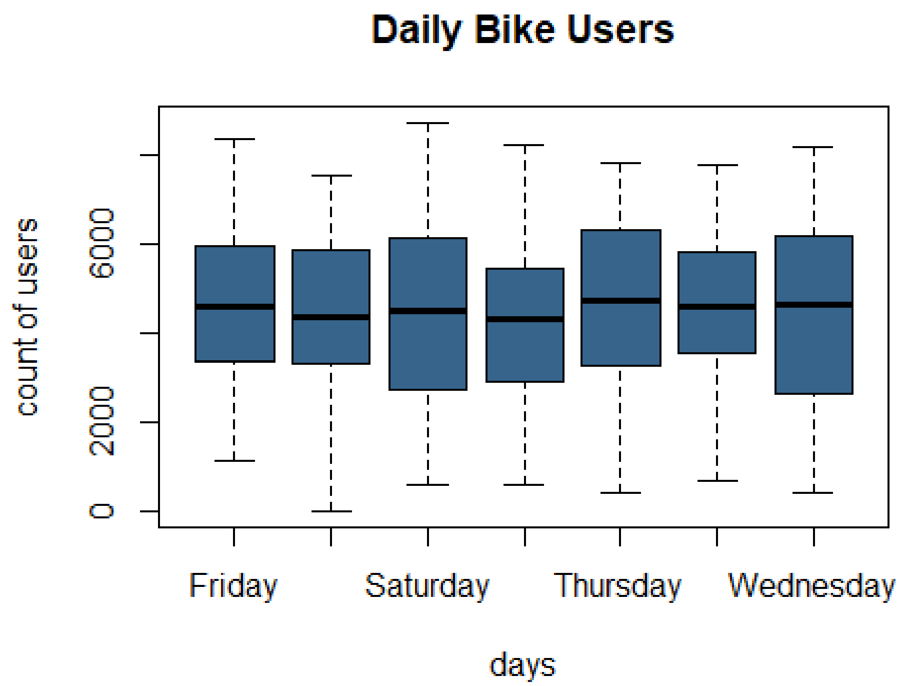
Here we will see the monthly trends of total bike counts

```
boxplot(day$cnt~day$mnth,xlab="mnth", ylab="count of users", col= "violetred4
",
        main = "Monthly Bike Users")
```



Here we will see the daily trends of total bike counts

```
date=substr(day$dteday,1,10)
days<-weekdays(as.Date(date))
day$days=days

par(mfrow=c(1,1))
boxplot(day$cnt~day$days,xlab="days", ylab="count of users", col = "steelblue
4",
        main = "Daily Bike Users")
```



Registered bikers are more as compare to casual bikers

```
#Casual VS Registered bikers
cr <- aggregate(. ~ mnth
                ,data = day[c("casual"
                                ,"registered"
                                ,"mnth")]
                ,sum)
rownames(cr) <- cr$mnth
cr <- cr[c("casual", "registered")]
print(cr)
```

```
##    casual registered
## 1   12042     122891
## 2   14963     136389
## 3   44444     184476
## 4   60802     208292
## 5   75285     256401
## 6   73906     272436
## 7   78157     266791
## 8   72039     279155
## 9   70323     275668
## 10  59760     262592
## 11  36603     218228
## 12  21693     189343
```

Registered users are more than casual users

Time to move to the next step, Data Manipulation . #In this process we wi ll try to change and adjust few data values to make data more sense abse d on our EDA and prior knowledge of the subject.

Let's remove variables which are not important.

```
day$instant<-NULL
day$dteday<- NULL
day$casual<-NULL
day$registered<- NULL
```

We removed casual, registered, dteday, and instrant from data to do linear regression. Casual and registered included in cnt and dteday is not a single independent variable.

Transform Month into quarters for dummy variables

```
day$Quarter <- ceiling(as.numeric(day$mnth) / 3)
day$Quarter<- factor(day$Quarter)
day$mnth = NULL
```

Transform working day and holiday as numeric variable because they have 0,1 value and we don't need dummy variable for these two variables.

```
day$holiday<- as.numeric(day$holiday)
day$workingday<- as.numeric(day$workingday)
```

Here we will create dummy variables for factor variables

```
factor_variables <- sapply(day,is.factor)
day_factor <- day[,factor_variables]

factor.names <- names(day_factor)
day_factor <- as.data.frame(day_factor)
day_factor <- acm.disjonctif(day_factor)
```

Now we will merge this data with our original data

```
day <- day[,-which(names(day) %in% factor.names)]

day <- cbind(day,day_factor)

rm(day_factor,factor_variables,factor.names)

nums <- unlist(lapply(day, is.numeric))
day<-day[,nums]

day$cnt<- as.numeric(day$cnt)
```

```r
day$yr<- as.factor(day$yr)
```

Again, we will transform holiday and workingday as factor for modeling

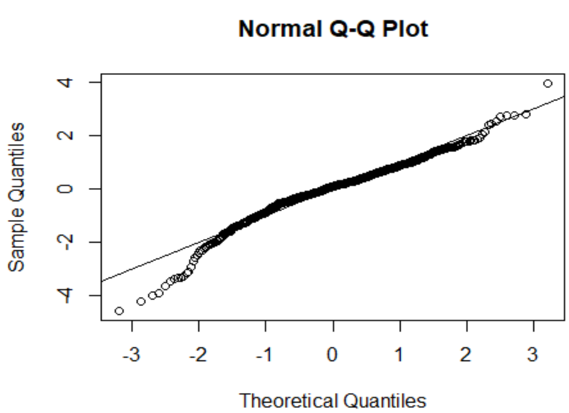
```r
day$holiday<- as.factor(day$holiday)
day$workingday<- as.factor(day$workingday)
```

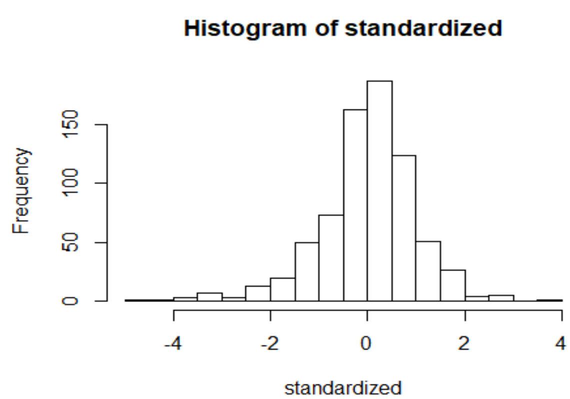
Final Data for Modeling is ready. Before modeling we will check the assumptions

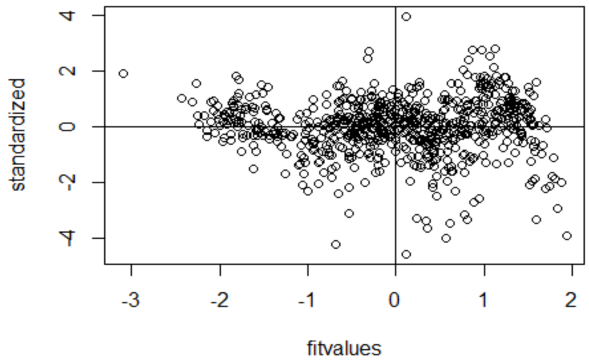
```r
#1. Linearity
linear<- lm(cnt~ ., data = day)
summary(linear)

##
## Call:
## lm(formula = cnt ~ ., data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3602.3  -370.9    70.4   486.0  3118.3
##
## Coefficients: (5 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3256.594    256.230  12.710  < 2e-16 ***
## yr1                 2024.596     60.791  33.304  < 2e-16 ***
## holiday2            -621.598    215.983  -2.878  0.00412 **
## workingday2           -8.478    112.427  -0.075  0.93991
## temp                  82.327     34.003   2.421  0.01572 *
## atemp                 32.252     30.166   1.069  0.28537
## hum                  -11.888      2.944  -4.038 5.98e-05 ***
## windspeed            -39.696      6.397  -6.205 9.27e-10 ***
## season.spring      -1898.018    166.722 -11.384  < 2e-16 ***
## season.summer       -846.833    193.271  -4.382 1.36e-05 ***
## season.fall        -1143.082    182.589  -6.260 6.64e-10 ***
## season.winter             NA         NA      NA       NA
## weekday.sun         -453.128    111.939  -4.048 5.73e-05 ***
## weekday.mon         -233.476    114.920  -2.032  0.04256 *
## weekday.tue         -141.674    112.803  -1.256  0.20955
## weekday.wed          -66.537    113.160  -0.588  0.55673
## weekday.thur         -42.771    112.531  -0.380  0.70400
## weekday.fri               NA         NA      NA       NA
## weekday.sat               NA         NA      NA       NA
## weathersit.Bad     -1948.840    205.379  -9.489  < 2e-16 ***
## weathersit.Normal   -458.701     80.269  -5.715 1.62e-08 ***
## weathersit.Good           NA         NA      NA       NA
## Quarter.1            436.168    163.015   2.676  0.00763 **
## Quarter.2            548.426    203.325   2.697  0.00716 **
## Quarter.3            602.136    187.064   3.219  0.00135 **
```

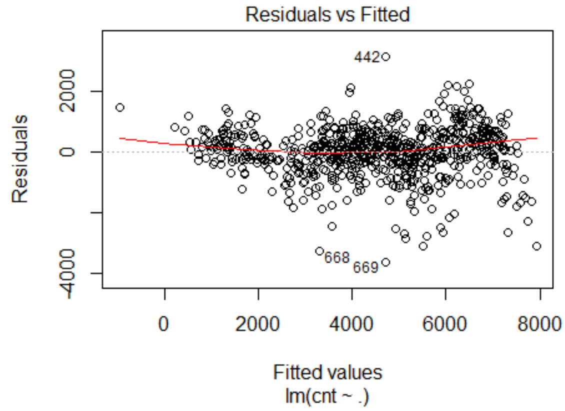
```
## Quarter.4                  NA          NA       NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 808.3 on 710 degrees of freedom
## Multiple R-squared:  0.8307, Adjusted R-squared:  0.8259
## F-statistic: 174.2 on 20 and 710 DF,  p-value: < 2.2e-16

#Create standardized residuals and plot linearity
standardized = rstudent(linear)
qqnorm(standardized)
abline(0,1)

#2 Normality

hist(standardized, breaks = 15)
```

```
mean(linear$residuals)

## [1] 1.668786e-14

#3Homogeneity/Homoscedasticity
fitvalues = scale(linear$fitted.values)
plot(fitvalues, standardized)
abline(0,0)
abline(v = 0)

plot(linear, 1)
```









From the above graph, we can say we met the assumptions for linearity, normality, homogeneity, and homoscedasticity.

Model Building process:-

Split final dataset into train and test dataset

```
set.seed(123)
smp_size <- floor(0.75 * nrow(day))
train_ind <- sample(seq_len(nrow(day)), size = smp_size)

train <- day[train_ind, ]
test <- day[-train_ind, ]
```

We select the multiple linear regression model because our response variable is numeric, and we will use more than 2 explanatory variables in our model.

building a model without the date, casual, registered and instant as the cnt variable includes both casual and registered and the dteday variable is not an independent variable, but consist variable that overlap with variables such as month, working day, holiday.

Model 1

```
model1<- lm(cnt ~ temp +atemp+ hum +windspeed, data = train)
summary(model1)

##
## Call:
## lm(formula = cnt ~ temp + atemp + hum + windspeed, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4816  -1054    -86   1028   3570
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3659.49     420.72   8.698  < 2e-16 ***
## temp           86.01      59.16   1.454    0.147
## atemp          72.27      54.77   1.319    0.188
## hum           -30.34       4.48  -6.772 3.32e-11 ***
## windspeed     -56.18      13.14  -4.275 2.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1447 on 543 degrees of freedom
## Multiple R-squared:  0.4493, Adjusted R-squared:  0.4452
## F-statistic: 110.8 on 4 and 543 DF,  p-value: < 2.2e-16
```

```
prediction<- predict(model1, newdata = train)
prediction1<- predict(model1, newdata = test)
mean((test$cnt- prediction1 )^2)
```

```
## [1] 1837637
```

```
RMSE(prediction1, test$cnt)
```

```
## [1] 1355.595
```

```
AIC(model1)
```

```
## [1] 9537.649
```

```
BIC(model1)
```

```
## [1] 9563.487
```

Model 2

```
model2<- lm(cnt~ temp+ hum+ windspeed, data= day)
summary(model2)
```

```
##
## Call:
## lm(formula = cnt ~ temp + hum + windspeed, data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4780.5 -1082.6   -62.2  1056.5  3653.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4084.363    337.862  12.089  < 2e-16 ***
## temp         161.598      7.148  22.606  < 2e-16 ***
## hum          -31.001      3.840  -8.073 2.83e-15 ***
## windspeed    -71.745     10.581  -6.781 2.48e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1425 on 727 degrees of freedom
## Multiple R-squared:  0.4609, Adjusted R-squared:  0.4587
## F-statistic: 207.2 on 3 and 727 DF,  p-value: < 2.2e-16
```

```
prediction02<- predict(model2, newdata = train)
prediction2<- predict(model2, newdata = test)
mean((test$cnt - prediction2)^2)
```

```
## [1] 1831080
```

```
RMSE(prediction2, test$cnt)
```

```
## [1] 1353.174
```

```
AIC(model2)
```

```
## [1] 12697.73
```

```
BIC(model2)
```

```
## [1] 12720.7
```

Model 3

```
model3<- lm(cnt~ .-atemp, data = day)
summary(model3)
```

```
##
## Call:
## lm(formula = cnt ~ . - atemp, data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3621.8  -361.0    66.7   484.8  3130.6
##
## Coefficients: (5 not defined because of singularities)
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3320.007    249.295  13.318  < 2e-16 ***
## yr1                2023.652     60.791  33.289  < 2e-16 ***
## holiday2           -637.249    215.508  -2.957  0.00321 **
## workingday2         -14.233    112.309  -0.127  0.89919
## temp                117.670      7.963  14.776  < 2e-16 ***
## hum                 -11.702      2.939  -3.981 7.55e-05 ***
## windspeed           -40.954      6.289  -6.512 1.40e-10 ***
## season.spring     -1901.722    166.703 -11.408  < 2e-16 ***
## season.summer      -843.252    193.261  -4.363 1.47e-05 ***
## season.fall       -1153.137    182.365  -6.323 4.52e-10 ***
## season.winter            NA         NA      NA       NA
## weekday.sun        -451.536    111.940  -4.034 6.08e-05 ***
## weekday.mon        -223.863    114.579  -1.954  0.05112 .
## weekday.tue        -134.416    112.610  -1.194  0.23302
```

```
## weekday.wed         -62.752    113.116  -0.555  0.57923    
## weekday.thur        -36.108    112.370  -0.321  0.74805    
## weekday.fri              NA         NA      NA       NA    
## weekday.sat              NA         NA      NA       NA    
## weathersit.Bad    -1962.833    204.982  -9.576  < 2e-16 ***
## weathersit.Normal  -461.834     80.223  -5.757 1.27e-08 ***
## weathersit.Good          NA         NA      NA       NA    
## Quarter.1           432.972    163.004   2.656  0.00808 ** 
## Quarter.2           538.345    203.126   2.650  0.00822 ** 
## Quarter.3           586.993    186.546   3.147  0.00172 ** 
## Quarter.4                NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 808.4 on 711 degrees of freedom
## Multiple R-squared:  0.8304, Adjusted R-squared:  0.8259 
## F-statistic: 183.2 on 19 and 711 DF,  p-value: < 2.2e-16
```

```
prediction03<- predict(model3, newdata = train)
```

```
## Warning in predict.lm(model3, newdata = train): prediction from a rank-def
icient
## fit may be misleading
```

```
prediction3<- predict(model3, newdata = test)
```

```
## Warning in predict.lm(model3, newdata = test): prediction from a rank-defi
cient
## fit may be misleading
```

```
mean(( test$cnt - prediction3)^2)
```

```
## [1] 755504.3
```

```
RMSE(prediction3, test$cnt)
```

```
## [1] 869.1975
```

```
AIC(model3)
```

```
## [1] 11884.31
```

```
BIC(model3)
```

```
## [1] 11980.79
```

Model 4

```
model4<- lm(cnt~ .- atemp-workingday-weekday.tue-weekday.mon-weekday.fri-
              weekday.tue-weekday.wed-weekday.thur-weekday.sat, data = train)
summary(model4)
```

```
##
## Call:
## lm(formula = cnt ~ . - atemp - workingday - weekday.tue - weekday.mon -
##     weekday.fri - weekday.tue - weekday.wed - weekday.thur -
##     weekday.sat, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3804.0  -405.3    46.6   455.8  3228.1
##
## Coefficients: (3 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3286.155    263.583  12.467  < 2e-16 ***
## yr1                 2203.498     67.804  32.498  < 2e-16 ***
## holiday2            -774.770    198.453  -3.904 0.000107 ***
## temp                 114.207      8.899  12.833  < 2e-16 ***
## hum                  -12.866      3.128  -4.114 4.51e-05 ***
## windspeed            -36.967      7.154  -5.168 3.36e-07 ***
## season.spring      -2183.682    182.572 -11.961  < 2e-16 ***
## season.summer      -1093.409    208.113  -5.254 2.16e-07 ***
## season.fall        -1154.809    187.929  -6.145 1.57e-09 ***
## season.winter             NA         NA      NA       NA
## weekday.sun         -287.811     96.233  -2.991 0.002911 **
## weathersit.Bad     -1588.700    231.747  -6.855 1.97e-11 ***
## weathersit.Normal   -489.934     86.873  -5.640 2.77e-08 ***
## weathersit.Good           NA         NA      NA       NA
## Quarter.1            630.977    177.329   3.558 0.000407 ***
## Quarter.2            739.861    221.123   3.346 0.000878 ***
## Quarter.3            564.471    195.479   2.888 0.004039 **
## Quarter.4                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 775.8 on 533 degrees of freedom
## Multiple R-squared:  0.8445, Adjusted R-squared:  0.8404
## F-statistic: 206.8 on 14 and 533 DF,  p-value: < 2.2e-16
```

```
prediction04<- predict(model4, newdata = train)
```

```
## Warning in predict.lm(model4, newdata = train): prediction from a rank-def
icient
## fit may be misleading
```

```
prediction4<- predict(model4, newdata = test)
```

```
## Warning in predict.lm(model4, newdata = test): prediction from a rank-defi
cient
## fit may be misleading
```

```
mean(( test$cnt - prediction4 )^2)
```

```
## [1] 877212.9
RMSE(prediction4, test$cnt)
## [1] 936.5965
AIC(model4)
## [1] 8864.653
BIC(model4)
## [1] 8933.554
```

Interpretation of Models

#First model shows that humidity and windspeed are good predictor with p value < 0.05 and adjusted R2 0.44.Temperature and feels like temperature show multicollinearity because they are highly correlated.

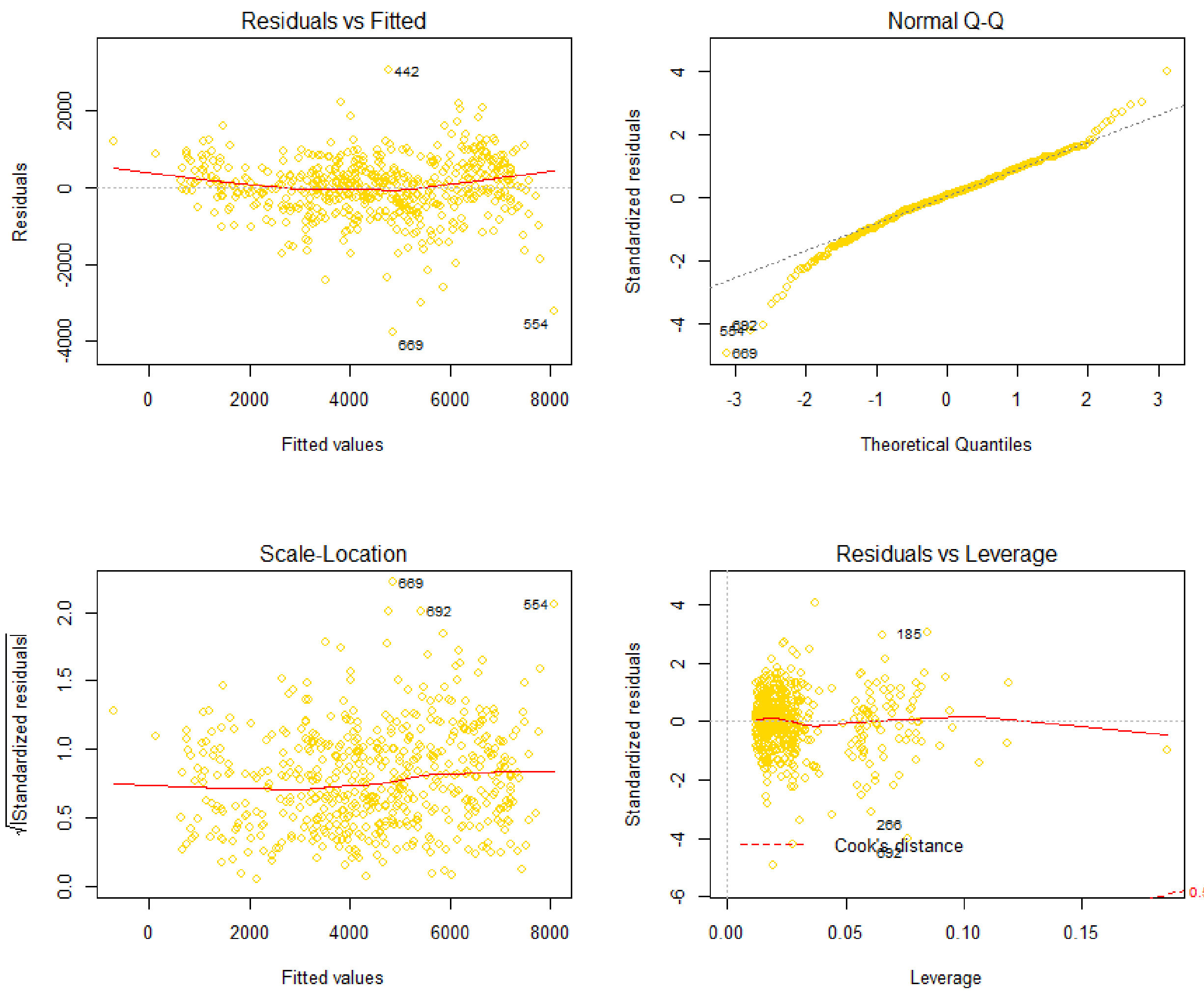
#To avoid multicollinearity, we removed feels like temperature in our model 2 and got results better from 1st model with adjusted R2 0.45 . To get better prediction, we try other variables in our next model.

#Third model includes all variables except feels like temperature and this model fit well with accounts for 82% of the variance. But this has RSE 808 with 19 variables and few variables are not significant. So, we will run model 4 to remove variables which are not significant.

#We run one more model to reduce variable number and this model also fit very well with adjusted R2 of 0.84 which means 84% of the variance can be explained by this model4. All variables are significant with < 0.05 p value. Model has lower AIC and BIC than other models and lower indicates a more parsimonious model, relative to a model fit with a higher AIC. So, based on these results we will select Model 4.

Let's plot our best fit model

```
plot(model4, col = "gold")
```

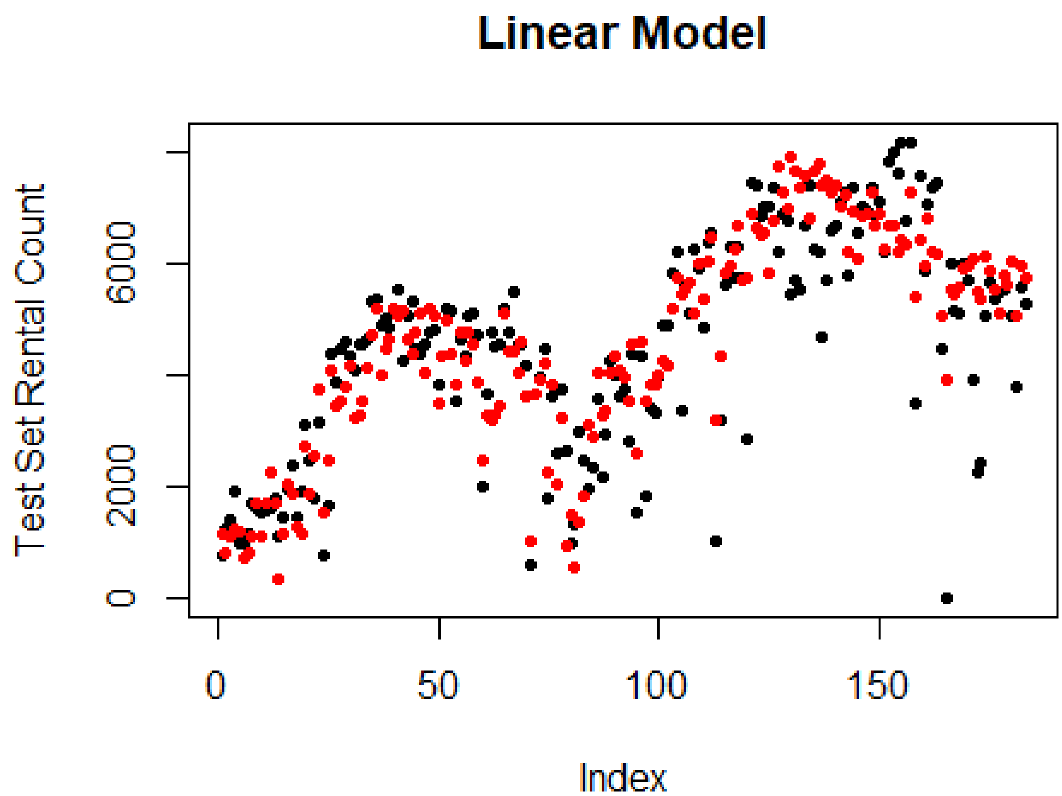


#Interpretation of plot

We built multiple linear regressions with putting all variables against response variable and removed insignificant predictor variables from earlier models. The best fit model achieved 0.84 adjusted R-squared, which indicates a good fit. Also, p value for almost all predictor variables are significant. Though, checking the residual plot and QQ plot, we can see that residuals have n0 pattern and are normally distributed, and residual plot shows slightly curve but close to straight line, which means the model fit the data well.

Here we will plot the prediction of best fit model and see the results

```
par(mfrow= c(1,1))
model4_step<- step(model4)

## Start:  AIC=7305.02
## cnt ~ (yr + holiday + workingday + temp + atemp + hum + windspeed +
##     season.spring + season.summer + season.fall + season.winter +
##     weekday.sun + weekday.mon + weekday.tue + weekday.wed + weekday.thur +
##     weekday.fri + weekday.sat + weathersit.Bad + weathersit.Normal +
##     weathersit.Good + Quarter.1 + Quarter.2 + Quarter.3 + Quarter.4) -
##     atemp - weekday.wed - weekday.thur - weekday.fri - weekday.mon -
##     weekday.tue - weekday.wed - workingday - weathersit.Good -
##     Quarter.4
##
##
## Step:  AIC=7305.02
## cnt ~ yr + holiday + temp + hum + windspeed + season.spring +
##     season.summer + season.fall + weekday.sun + weekday.sat +
##     weathersit.Bad + weathersit.Normal + Quarter.1 + Quarter.2 +
##     Quarter.3
##
##                     Df Sum of Sq       RSS    AIC
## <none>                           318207150 7305.0
## - weekday.sat        1   2609064 320816214 7307.5
## - weekday.sun        1   3922147 322129297 7309.7
## - Quarter.3          1   4710693 322917843 7311.1
## - Quarter.2          1   6444228 324651378 7314.0
## - Quarter.1          1   7766672 325973821 7316.2
## - holiday            1   8258538 326465687 7317.1
## - hum                1  10263132 328470282 7320.4
## - windspeed          1  16274080 334481230 7330.4
## - season.summer      1  16416747 334623897 7330.6
## - weathersit.Normal  1  18929391 337136540 7334.7
## - season.fall        1  22822318 341029468 7341.0
## - weathersit.Bad     1  27946011 346153160 7349.2
## - season.spring      1  85871259 404078409 7433.9
## - temp               1 100865761 419072911 7453.9
## - yr                 1 631750393 949957543 7902.4

plot(test$cnt, main = "Linear Model", ylab = "Test Set Rental Count", pch = 2
0)
points(predict(model5_step, newdata = test), col = "red", pch = 20)
```



Predicting using the attributes from testing dataset and plot them against the true values the graph shows that the spread of the response variable is similar to multilinear model. Still, we cannot depend on this because we worked on a small data and this dataset does not contain more information like daily hours and bike stations, which can help more in accuracy.