Assignment 2 : R Basics and Exploratory Data Analysis

Krishu Kumar Thapa

2022-09-04

Working on Red Wine Quality data.

Question 1a.

The csv file for red wine quality has been loaded using **read.csv()** as shown in the chunk below:

```
redwine = read.csv('winequality-red.csv', sep=',', header = TRUE)

str(redwine)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed_acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile_acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric_acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual_sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free_sulfur_dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total_sulfur_dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

Also the header has been discarded from being considered as the row from the config *header=TRUE*.

Question 1b.

The median of the *quality* of all wines can be computed with the code below:

```
median(redwine$quality)
```

```
## [1] 6
```

Similarly we can compute the mean *alcohol level* with the help of code below:

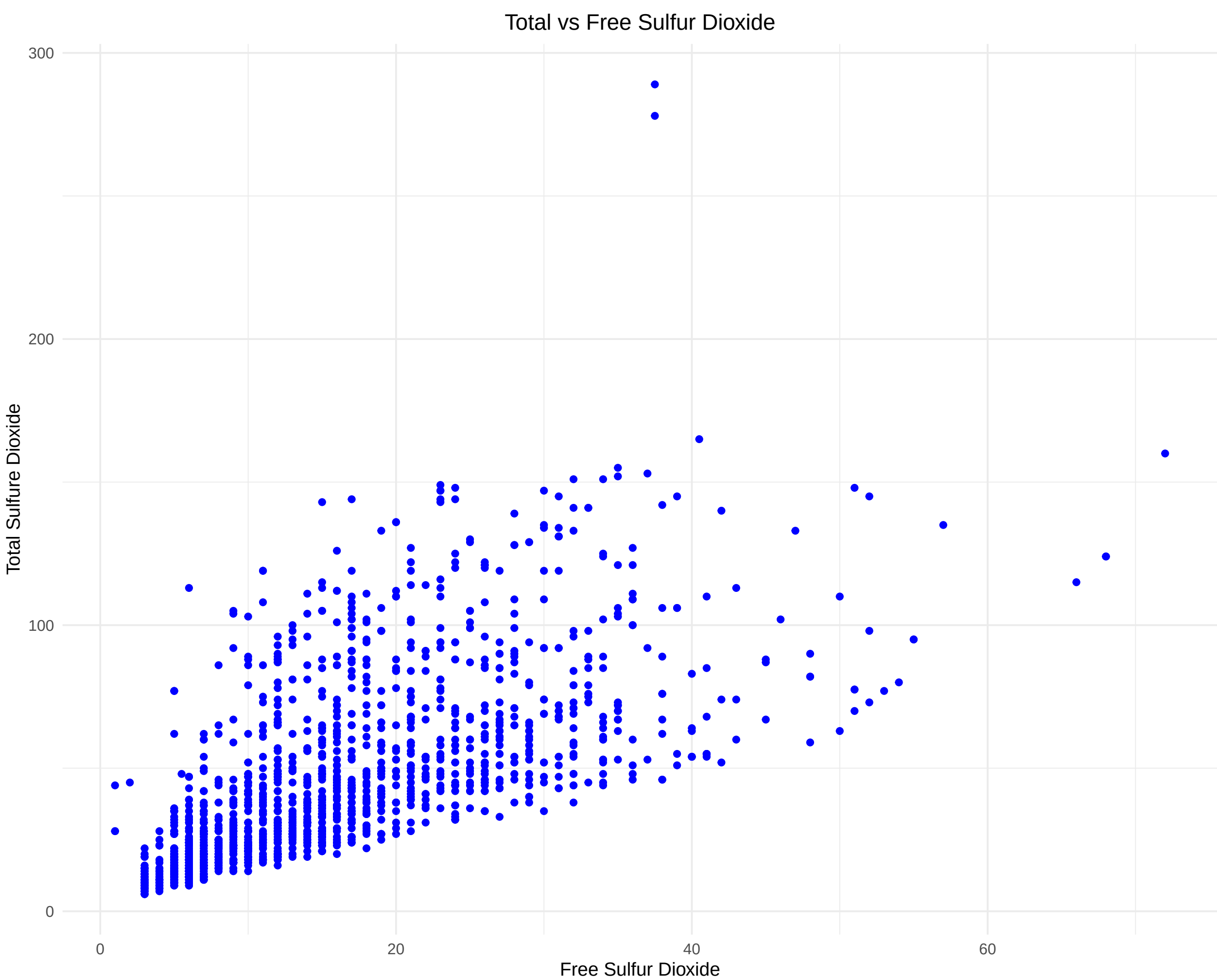
```
mean(redwine$alcohol)
```

```
## [1] 10.42298
```

Question 1c.

Showing the scatter plot between two data level , namely **free_sulfur_dioxide** and **total_sulfur_dioxide**

```
# install.packages(ggplot2)

library(ggplot2)

plotsetup <- ggplot(redwine, aes(x= free_sulfur_dioxide, y= total_sulfur_dioxide))

plotcolor <- plotsetup + geom_point(color= "blue")+theme_minimal()

finalplot <- plotcolor+ labs(x="Free Sulfur Dioxide", y= "Total Sulfure Dioxide",
             title= "Total vs Free Sulfur Dioxide")+
            theme(plot.title = element_text(hjust = 0.5))

finalplot
```



Question 1d.

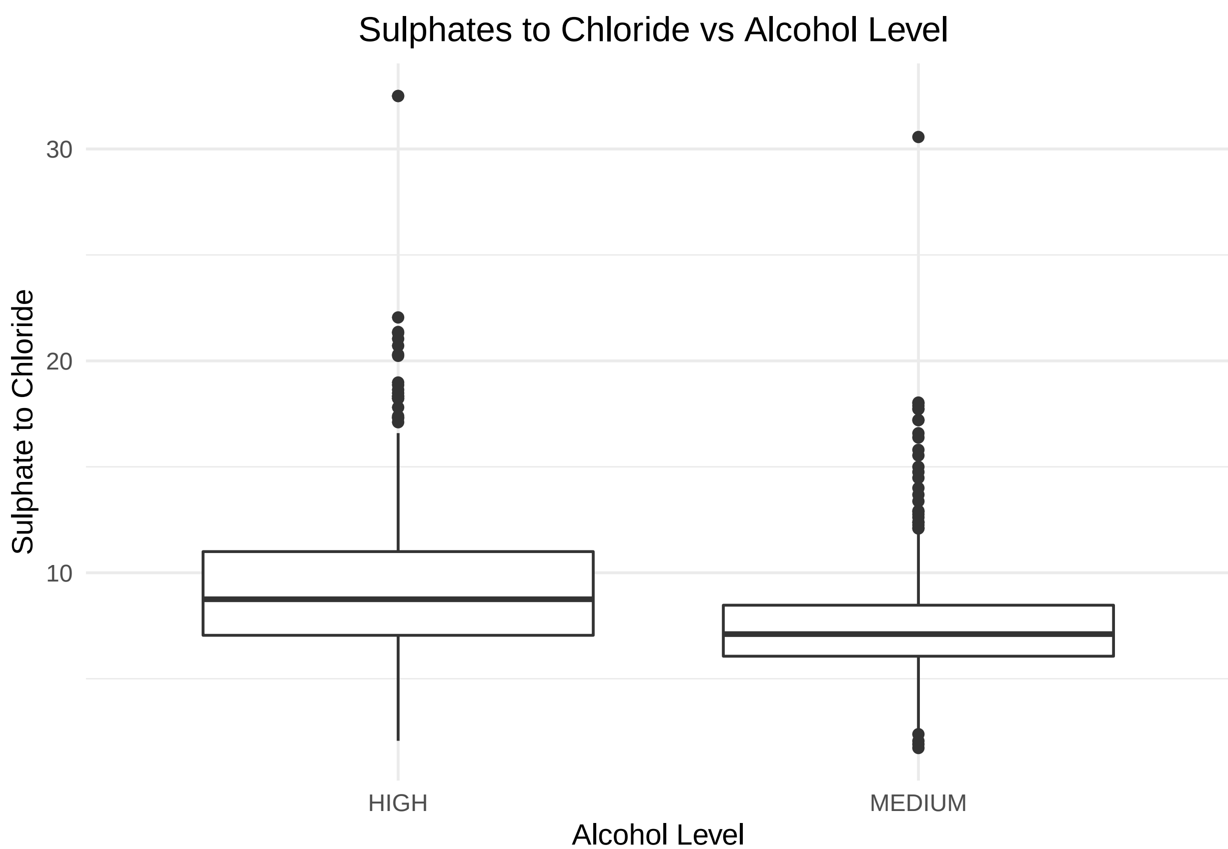
Creating new variable ALevel based on the given condition in the question.

```
alabel <- c('High','Medium')

redwine$ALevel <- as.factor(ifelse(redwine$alcohol > 10.2 , "HIGH", "MEDIUM"))

str(redwine)
```

```
## 'data.frame':    1599 obs. of  13 variables:
##  $ fixed_acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile_acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric_acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual_sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free_sulfur_dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total_sulfur_dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
##  $ ALevel              : Factor w/ 2 levels "HIGH","MEDIUM": 2 2 2 2 2 2 2 2 2 1 ...
```

Creating the plot:

```
redwine$stc <- redwine$sulphates / redwine$chlorides

plotsetup <- ggplot(redwine, aes(x= ALevel, y= stc))

plotcolor <- plotsetup + geom_boxplot()+theme_minimal()

finalplot <- plotcolor+ labs(x="Alcohol Level", y= "Sulphate to Chloride",
             title= "Sulphates to Chloride vs Alcohol Level ")+
            theme(plot.title = element_text(hjust = 0.5))

finalplot
```



Number of samples in **HIGH** is given by:

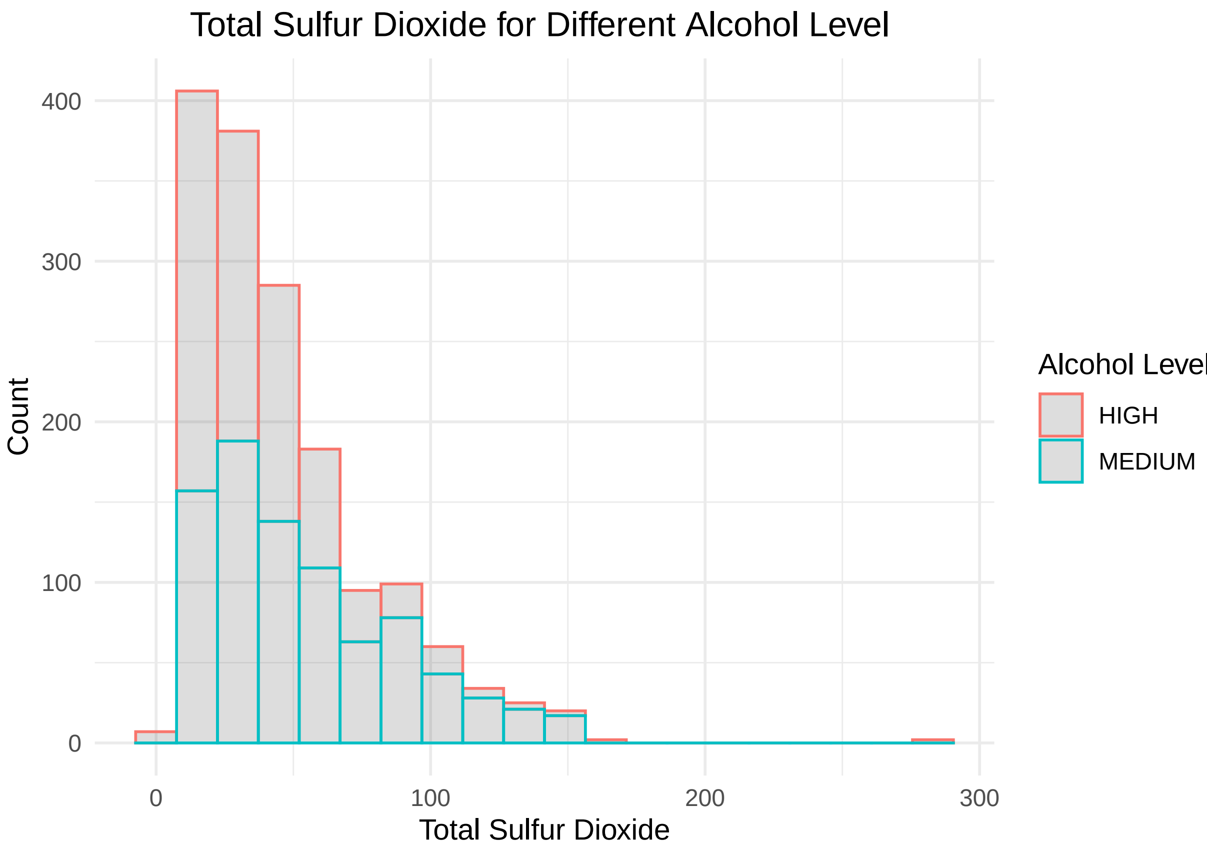
```
length(which(redwine$ALevel== 'HIGH'))
```

```
## [1] 757
```

Question 1e.

Plotting the ALevel against total_sulfur_dioxide:

```
extractedDataFrame <- data.frame(ALevel = redwine$ALevel,
                            total_sulfur_dioxide = redwine$total_sulfur_dioxide)

plotsetup <- ggplot(extractedDataFrame, aes(x= total_sulfur_dioxide,
          color= ALevel))

plotcolor <- plotsetup + geom_histogram(bins='20',alpha= 0.2)+
  scale_color_discrete(name="Alcohol Level")+
  scale_fill_manual(values= c("blue", "red"))+theme_minimal()

finalplot <- plotcolor+ labs(x="Total Sulfur Dioxide", y= "Count",
             title= "Total Sulfur Dioxide for Different Alcohol Level ")+
            theme(plot.title = element_text(hjust = 0.5))

finalplot
```

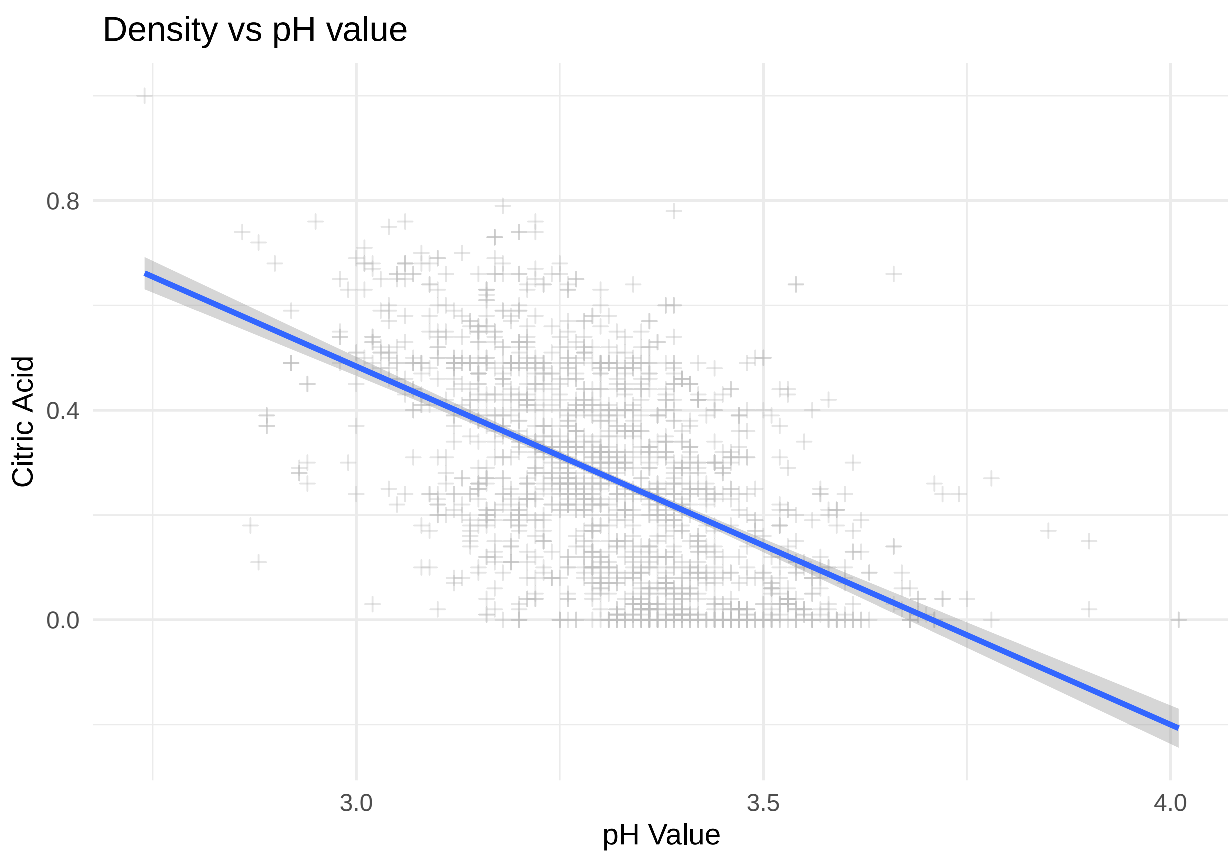


Question 1f.

i. First I want to plot the chart to see the relation of *pH* value with *citric_acid*.

```
ggplot(redwine, aes(x=pH, y=citric_acid))+ geom_point(color='gray', pch=3,
      alpha=0.4)+geom_smooth(method='lm')+theme_minimal()+
  labs(x="pH Value", y="Citric Acid", title=" Density vs pH value")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

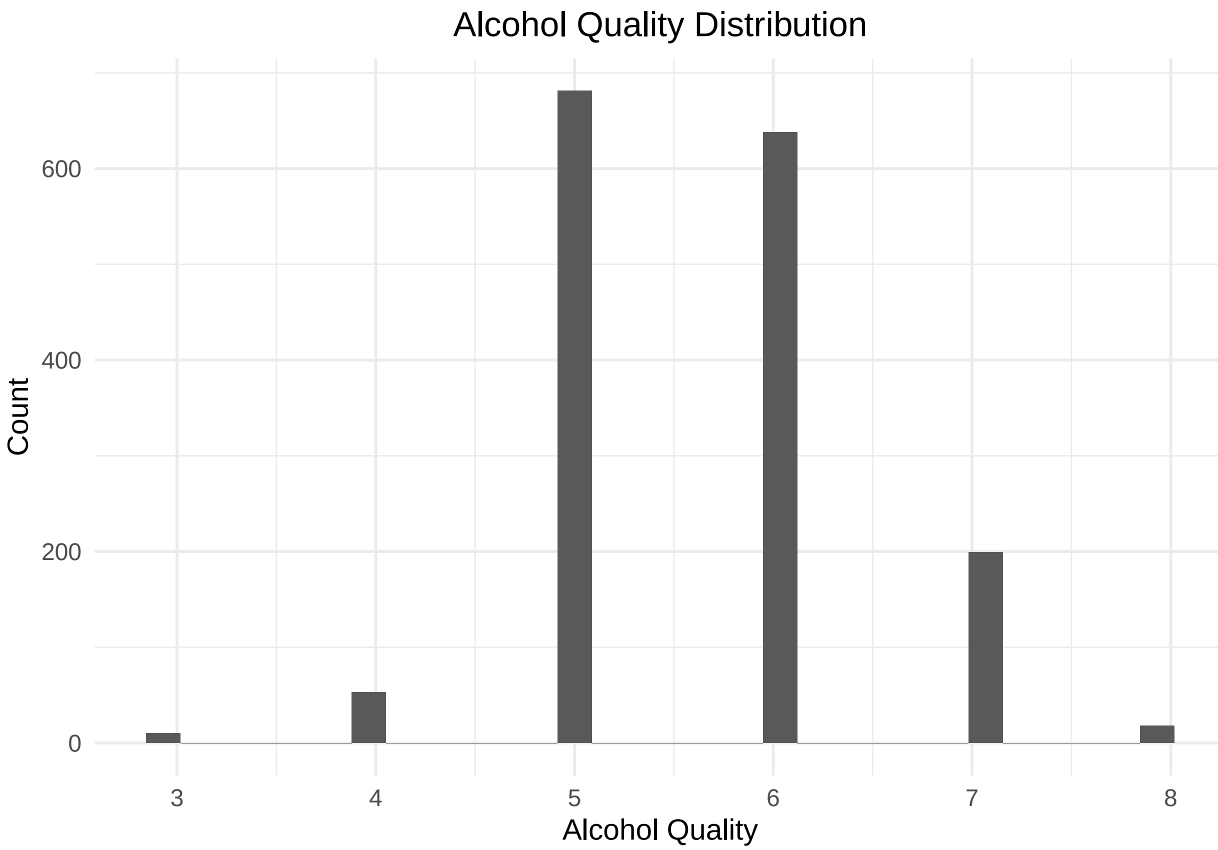


From the plot we can see the pattern that , *citric* and *pH* are **inversely proportional** to each other.

ii. Now the plot is added to analyze the distribution of alcohol of various quantity.

```
ggplot(redwine, aes(x=quality))+ geom_histogram()+theme_minimal()+
  labs(x="Alcohol Quality", y="Count", title="Alcohol Quality Distribution")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It can be seen that the csv file data has more numbers of red wines with the average alcohol quality as compared to that of low and high quality alcohol.

Working on Forest Fires data:

Question 2a.

The csv file for forest fires has been loaded using **read.csv()** as shown in the chunk below:

```
forestfire = read.csv('forestfires.csv', sep=',', header = TRUE)

str(forestfire)
```

```
## 'data.frame':    517 obs. of  15 variables:
##  $ X.2  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X.1  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X    : int  7 7 7 8 8 8 8 8 8 7 ...
##  $ Y    : int  5 4 4 6 6 6 6 6 6 5 ...
##  $ month: chr  "mar" "oct" "oct" "mar" ...
##  $ day  : chr  "fri" "tue" "sat" "fri" ...
##  $ FFMC : num  86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...
##  $ DMC  : num  26.2 35.4 43.7 33.3 51.3 ...
##  $ DC   : chr  "94.3" "669.1" "686.9" "77.5" ...
##  $ ISI  : num  5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...
##  $ temp : num  8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...
##  $ RH   : int  51 33 33 97 99 29 27 86 63 40 ...
##  $ wind : num  6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...
##  $ rain : num  0 0 0 0.2 0 0 0 0 0 0 ...
##  $ area : num  0 0 0 0 0 0 0 0 0 0 ...
```

From the data it can be induced that the *quantitative predictors* are the data fields like **FFMC, DMC, ISI, temp, RH,wind, rain, and area**. Similarly, the qualitative predictors in the data are **month** and **day**.

Changing Qualitative predictors **month** and **day** as factor:

```
  forestfire$month <- as.factor(forestfire$month)
  forestfire$day <- as.factor(forestfire$day)
```

Changing Qualitative predictors **RH** and **DC** as factor:

```
  forestfire$RH <- as.numeric(forestfire$RH)
  suppressWarnings(forestfire$DC <- as.numeric(forestfire$DC))
```

Final strcuture after the adjustment:

```
  str(forestfire)
```

```
## 'data.frame':    517 obs. of  15 variables:
##  $ X.2  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X.1  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X    : int  7 7 7 8 8 8 8 8 8 7 ...
##  $ Y    : int  5 4 4 6 6 6 6 6 6 5 ...
##  $ month: Factor w/ 12 levels "apr","aug","dec",..: 8 11 11 8 8 2 2 2 12 12 ...
##  $ day  : Factor w/ 7 levels "fri","mon","sat",..: 1 6 3 1 4 4 2 2 6 3 ...
```

```
## $ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...
## $ DMC  : num 26.2 35.4 43.7 33.3 51.3 ...
## $ DC   : num 94.3 669.1 686.9 77.5 102.2 ...
## $ ISI  : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...
## $ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...
## $ RH   : num 51 33 33 97 99 29 27 86 63 40 ...
## $ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...
## $ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...
## $ area : num 0 0 0 0 0 0 0 0 0 0 ...
```

Question 2b.

The range,mean and standard deviation of each quantitative predictor is given by:

1. **FFMC:**

```r
mean(forestfire$FFMC, na.rm= TRUE)
range(forestfire$FFMC, na.rm = TRUE)
sd(forestfire$FFMC, na.rm = TRUE)
```

```
## [1] 90.64468
## [1] 18.7 96.2
## [1] 5.520111
```

2. **DMC:**

```r
mean(forestfire$DMC, na.rm= TRUE)
range(forestfire$DMC, na.rm = TRUE)
sd(forestfire$DMC, na.rm = TRUE)
```

```
## [1] 110.8723
## [1]   1.1 291.3
## [1] 64.04648
```

3. **DC:**

```r
mean(forestfire$DC, na.rm= TRUE)
range(forestfire$DC, na.rm = TRUE)
sd(forestfire$DC, na.rm = TRUE)
```

```
## [1] 547.8107
## [1]   7.9 860.6
## [1] 248.2895
```

4. **ISI:**

```r
mean(forestfire$ISI, na.rm= TRUE)
range(forestfire$ISI, na.rm = TRUE)
sd(forestfire$ISI, na.rm = TRUE)
```

```
## [1] 9.021663
```

```
## [1]  0.0 56.1
## [1] 4.559477
```

5. **Temp:**

```
mean(forestfire$temp, na.rm= TRUE)
range(forestfire$temp, na.rm = TRUE)
sd(forestfire$temp, na.rm = TRUE)
```

```
## [1] 18.88917
## [1]  2.2 33.3
## [1] 5.806625
```

6. **RH:**

```
mean(forestfire$RH, na.rm= TRUE)
range(forestfire$RH, na.rm = TRUE)
sd(forestfire$RH, na.rm = TRUE)
```

```
## [1] 44.2882
## [1]  15 100
## [1] 16.31747
```

7. **Wind:**

```
mean(forestfire$wind, na.rm= TRUE)
range(forestfire$wind, na.rm = TRUE)
sd(forestfire$wind, na.rm = TRUE)
```

```
## [1] 4.017602
## [1] 0.4 9.4
## [1] 1.791653
```

8. **Rain:**

```
mean(forestfire$rain, na.rm= TRUE)
range(forestfire$rain, na.rm = TRUE)
sd(forestfire$rain, na.rm = TRUE)
```

```
## [1] 0.02166344
## [1] 0.0 6.4
## [1] 0.2959591
```

9. **Area:**

```
mean(forestfire$area, na.rm= TRUE)
range(forestfire$area, na.rm = TRUE)
sd(forestfire$area, na.rm = TRUE)
```

```
## [1] 12.84729
```

```
## [1]    0.00 1090.84
## [1] 63.65582
```

The day of the week that has highest number of wildfire is given by:

```
table(forestfire$day)
names(which.max(table(forestfire$day)))
```

```
##
## fri mon sat sun thu tue wed
##  85  74  84  95  61  64  54
## [1] "sun"
```

Question 2c:

Removing the data from 40 through 80:

```
modified_forestfire <- forestfire[-c(40:80),]

str(modified_forestfire)
```

```
## 'data.frame':    476 obs. of  15 variables:
##  $ X.2  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X.1  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X    : int  7 7 7 8 8 8 8 8 8 7 ...
##  $ Y    : int  5 4 4 6 6 6 6 6 6 5 ...
##  $ month: Factor w/ 12 levels "apr","aug","dec",..: 8 11 11 8 8 2 2 2 12 12 ...
##  $ day  : Factor w/ 7 levels "fri","mon","sat",..: 1 6 3 1 4 4 2 2 6 3 ...
##  $ FFMC : num  86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...
##  $ DMC  : num  26.2 35.4 43.7 33.3 51.3 ...
##  $ DC   : num  94.3 669.1 686.9 77.5 102.2 ...
##  $ ISI  : num  5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...
##  $ temp : num  8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...
##  $ RH   : num  51 33 33 97 99 29 27 86 63 40 ...
##  $ wind : num  6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...
##  $ rain : num  0 0 0 0.2 0 0 0 0 0 0 ...
##  $ area : num  0 0 0 0 0 0 0 0 0 0 ...
```

Now calculating the mean, range and standard deviation again.

1. **FFMC:**

```
mean(modified_forestfire$FFMC, na.rm= TRUE)
range(modified_forestfire$FFMC, na.rm = TRUE)
sd(modified_forestfire$FFMC, na.rm = TRUE)
```

```
## [1] 90.66429
```

```
## [1] 18.7 96.2
## [1] 5.681003
```

2. **DMC:**

```r
mean(modified_forestfire$DMC, na.rm= TRUE)
range(modified_forestfire$DMC, na.rm = TRUE)
sd(modified_forestfire$DMC, na.rm = TRUE)
```

```
## [1] 113.4664
## [1]   1.1 291.3
## [1] 65.04941
```

3. **DC:**

```r
mean(modified_forestfire$DC, na.rm= TRUE)
range(modified_forestfire$DC, na.rm = TRUE)
sd(modified_forestfire$DC, na.rm = TRUE)
```

```
## [1] 555.5434
## [1]   7.9 860.6
## [1] 244.5121
```

4. **ISI:**

```r
mean(modified_forestfire$ISI, na.rm= TRUE)
range(modified_forestfire$ISI, na.rm = TRUE)
sd(modified_forestfire$ISI, na.rm = TRUE)
```

```
## [1] 9.065756
## [1]  0.0 56.1
## [1] 4.633378
```

5. **Temp:**

```r
mean(modified_forestfire$temp, na.rm= TRUE)
range(modified_forestfire$temp, na.rm = TRUE)
sd(modified_forestfire$temp, na.rm = TRUE)
```

```
## [1] 19.01155
## [1]  2.2 33.3
## [1] 5.848737
```

6. **RH:**

```r
mean(modified_forestfire$RH, na.rm= TRUE)
range(modified_forestfire$RH, na.rm = TRUE)
sd(modified_forestfire$RH, na.rm = TRUE)
```

```
## [1] 44.47269
```

```
## [1]  15 100
## [1] 16.42082
```

7. **Wind:**

```
mean(modified_forestfire$wind, na.rm= TRUE)
range(modified_forestfire$wind, na.rm = TRUE)
sd(modified_forestfire$wind, na.rm = TRUE)
```

```
## [1] 4.013235
## [1] 0.4 9.4
## [1] 1.804279
```

8. **Rain:**

```
mean(modified_forestfire$rain, na.rm= TRUE)
range(modified_forestfire$rain, na.rm = TRUE)
sd(modified_forestfire$rain, na.rm = TRUE)
```

```
## [1] 0.02352941
## [1] 0.0 6.4
## [1] 0.3083964
```

9. **Area:**

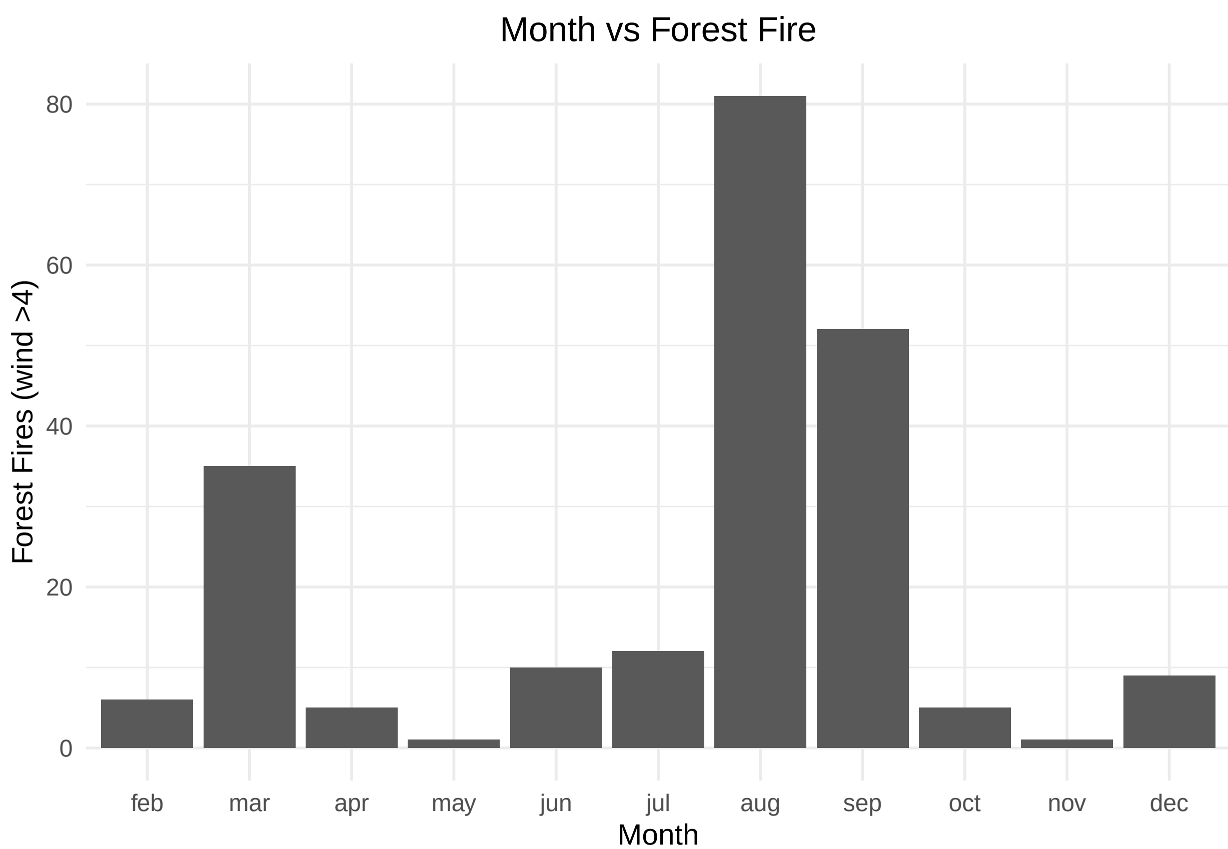
```
mean(modified_forestfire$area, na.rm= TRUE)
range(modified_forestfire$area, na.rm = TRUE)
sd(modified_forestfire$area, na.rm = TRUE)
```

```
## [1] 13.95389
## [1]    0.00 1090.84
## [1] 66.2295
```

Question 2d.

Bar plot showing the count of forest fires in each month for which wind is greater than 4 is shown below:

```
ggplot(forestfire[forestfire$wind >4,], aes(x= factor(month,
  c('jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov','dec')),
  na.rm=TRUE))+geom_bar()+
  labs(x="Month", y="Forest Fires (wind >4)", title="Month vs Forest Fire")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
```



For the month which is more common for high wind forest fires is:

```
extractedInfo <- forestfire[forestfire$wind >4 ,]
names(which.max(table(extractedInfo$month)))
```

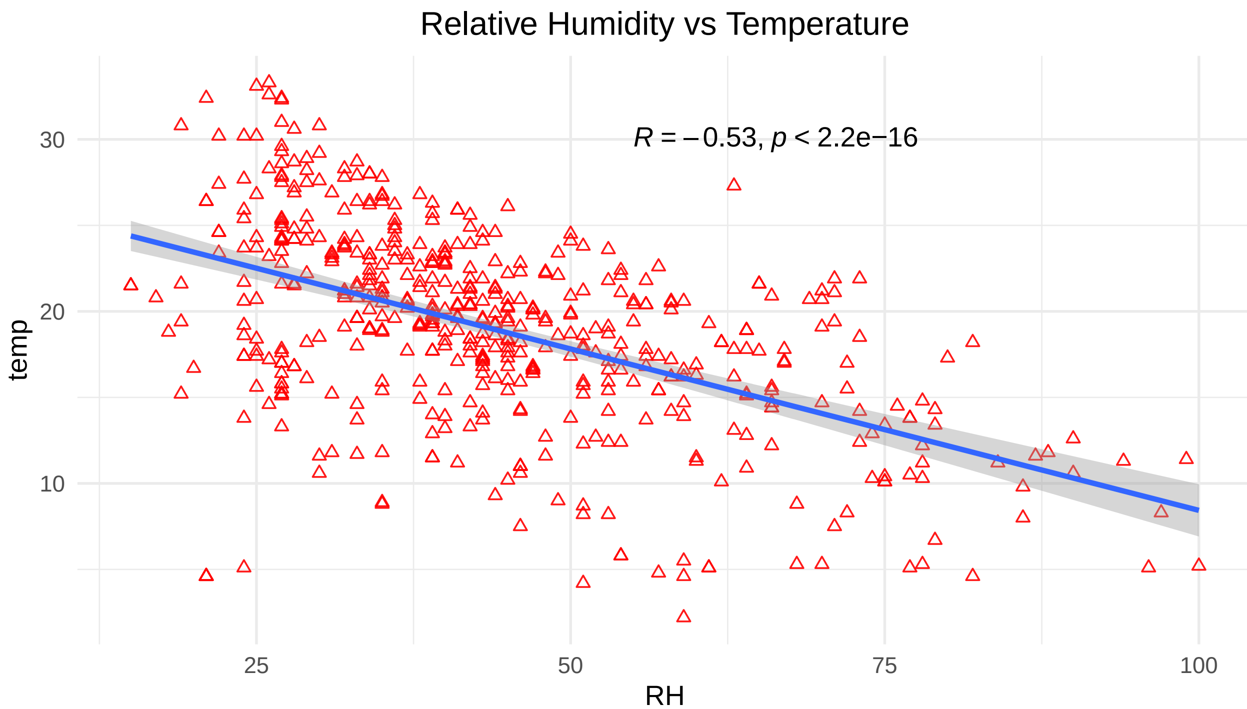
```
## [1] "aug"
```

Question 2e.

Interpreting the predictors graphically:

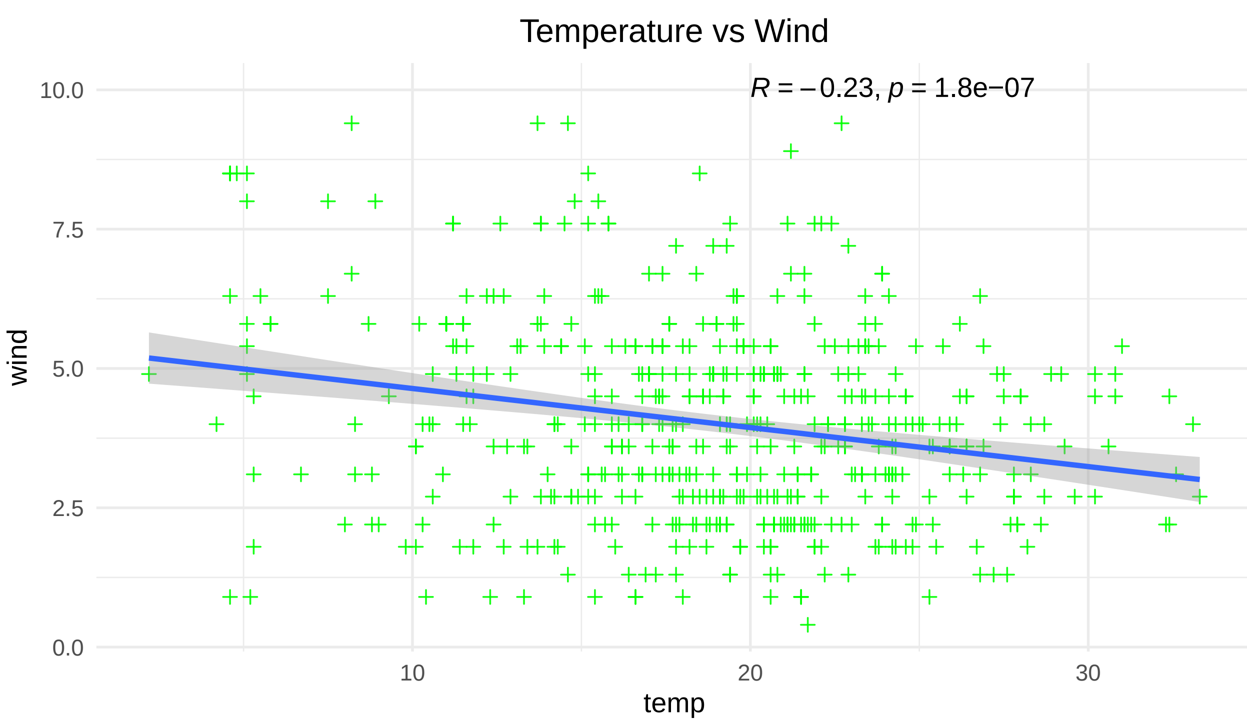
```
library(ggpubr)

ggplot(forestfire, aes(x= RH,y =temp,
  na.rm=TRUE))+geom_point(alpha=0.9,pch=2, color="red")+
    stat_cor(method='pearson',label.x = 55 , label.y = 30)+
  geom_smooth(method='lm')+
  labs(title="Relative Humidity vs Temperature")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



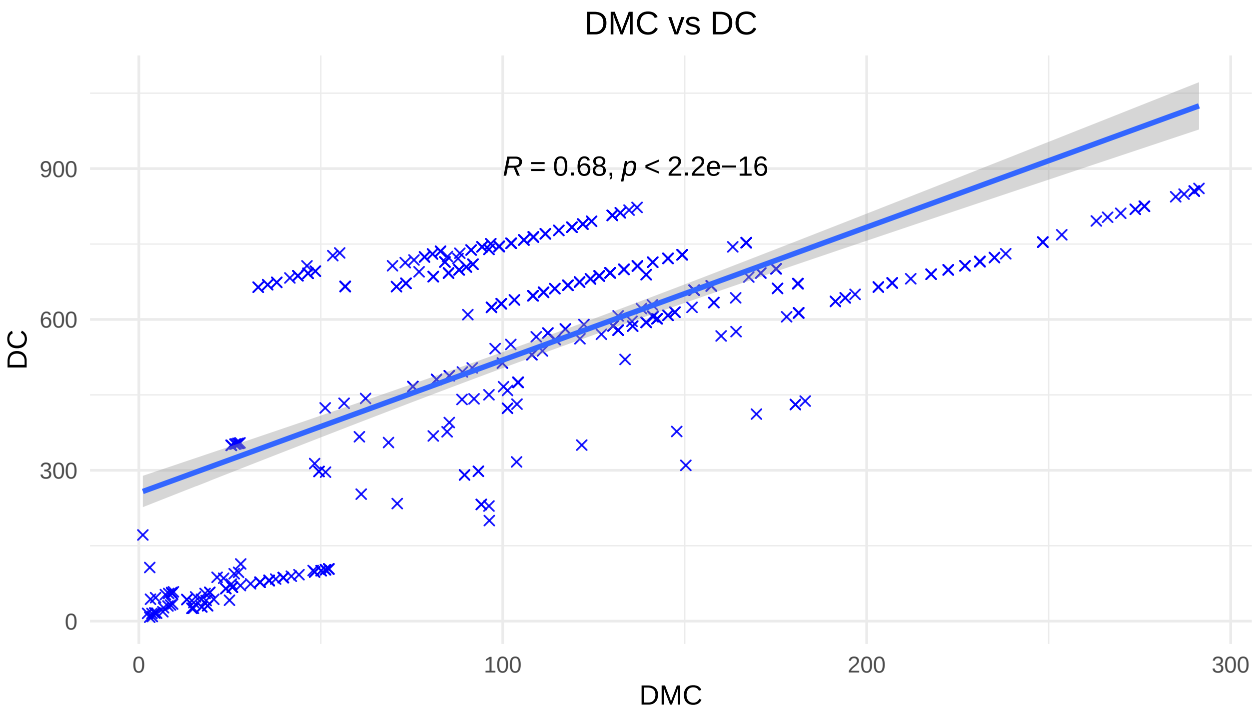

```
ggplot(forestfire, aes(x= temp,y = wind,
  na.rm=TRUE))+geom_point(alpha=0.9,pch=3,color= "green")+
    stat_cor(method='pearson', label.x = 20 , label.y = 10)+
  geom_smooth(method='lm')+
  labs(title="Temperature vs Wind")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



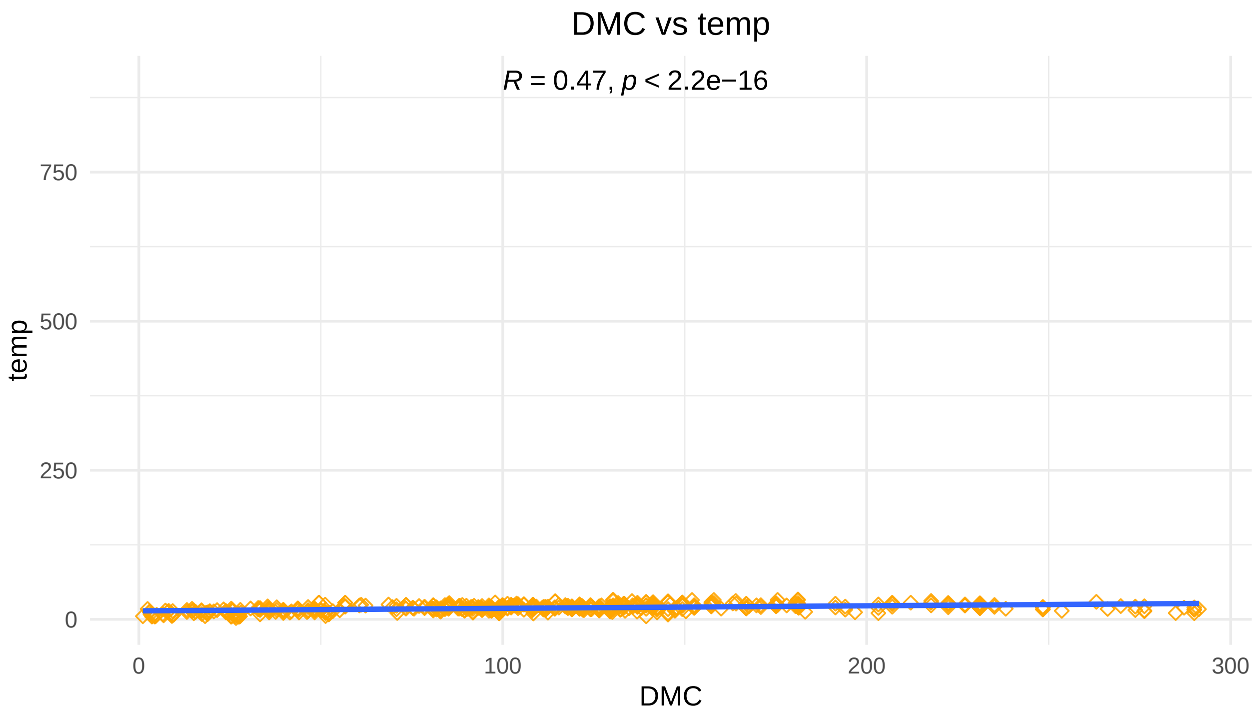
```r
ggplot(forestfire, aes(x= DMC,y = DC,
  na.rm=TRUE))+geom_point(alpha=0.9,pch=4, color="blue")+
    stat_cor(method='pearson', label.x = 100 , label.y = 900)+
  geom_smooth(method='lm')+
  labs(title="DMC vs DC")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_cor).

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing missing values (geom_point).
```




```r
ggplot(forestfire, aes(x= DMC,y = temp,
  na.rm=TRUE))+geom_point(alpha=0.9,pch=5, color="orange")+
    stat_cor(method='pearson', label.x = 100 , label.y = 900)+
  geom_smooth(method='lm')+
  labs(title="DMC vs temp")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The first two charts showing the plots between **temp vs RH** and **temp vs wind** have negative correlation which means they are inversely proportional to eachother. Also this relation is backed by the probability variable for null hypothesis of the correlation.

Similarly in the next two charts which shows the plot between **DMC vs DC** and **DMC vs temp** have the positive correlation which means that they are directly proportional to each other. Also the relation is backed by the probability variable for null hypothesis of the correlation.

One thing that I could notice here is that *temperature (temp)* is a variable which has relation to many other variables in the dataset.

Correlation matrix is given by:

```
releventDataFrame <- data.frame(temp= forestfire$temp , DMC = forestfire$DMC ,
                DC= forestfire$DC , wind = forestfire$wind)

cor(na.omit(releventDataFrame))
```

```
##            temp        DMC         DC       wind
## temp  1.0000000  0.4691086  0.4962830 -0.2273893
## DMC   0.4691086  1.0000000  0.6821522 -0.1053626
## DC    0.4962830  0.6821522  1.0000000 -0.2034749
## wind -0.2273893 -0.1053626 -0.2034749  1.0000000
```

Question 2f.

From the above correlation matrix we can say that wind has a significant correlation to temp and DMC as compared to DC. So, we can use temp and DMC variable to predict the value of wind in the data set. The wind correlation with DC is comparatively weak.