Assignment 2

2022-09-09

Part 1

Red Wine Quality

a.) Reads winequality-red.csv into 'redwine'.

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

summary(redwine)
```

```
##  fixed_acidity   volatile_acidity  citric_acid    residual_sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free_sulfur_dioxide total_sulfur_dioxide    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

b.) Determines the median of red wine quality and mean of alcohol taste score. The median is 6 and the mean is 10.42298.

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

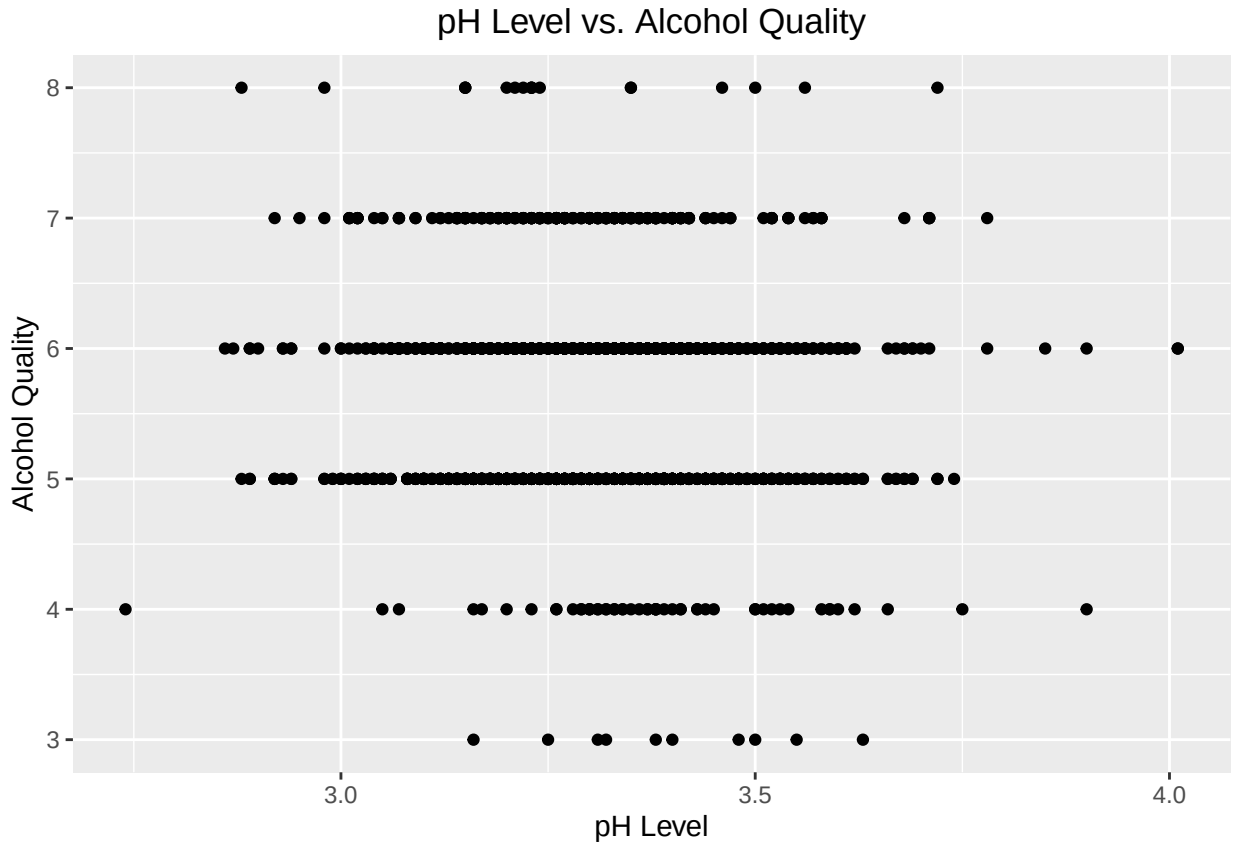
```
## [1] 6
```

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

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

c.) Plots a scatter plot of pH to alcohol quality.

```
library(ggplot2)
ggplot(redwine, aes(x = pH, y = quality)) +
  geom_point() +
  labs(x = "pH Level", y = "Alcohol Quality", title = "pH Level vs. Alcohol Quality") +
  theme(plot.title = element_text(hjust = 0.5))
```



d.) Creates a new column in the data frame 'redwine' called 'ALevel' and populates it with either 'High' or 'Medium', depending on the alcohol level.

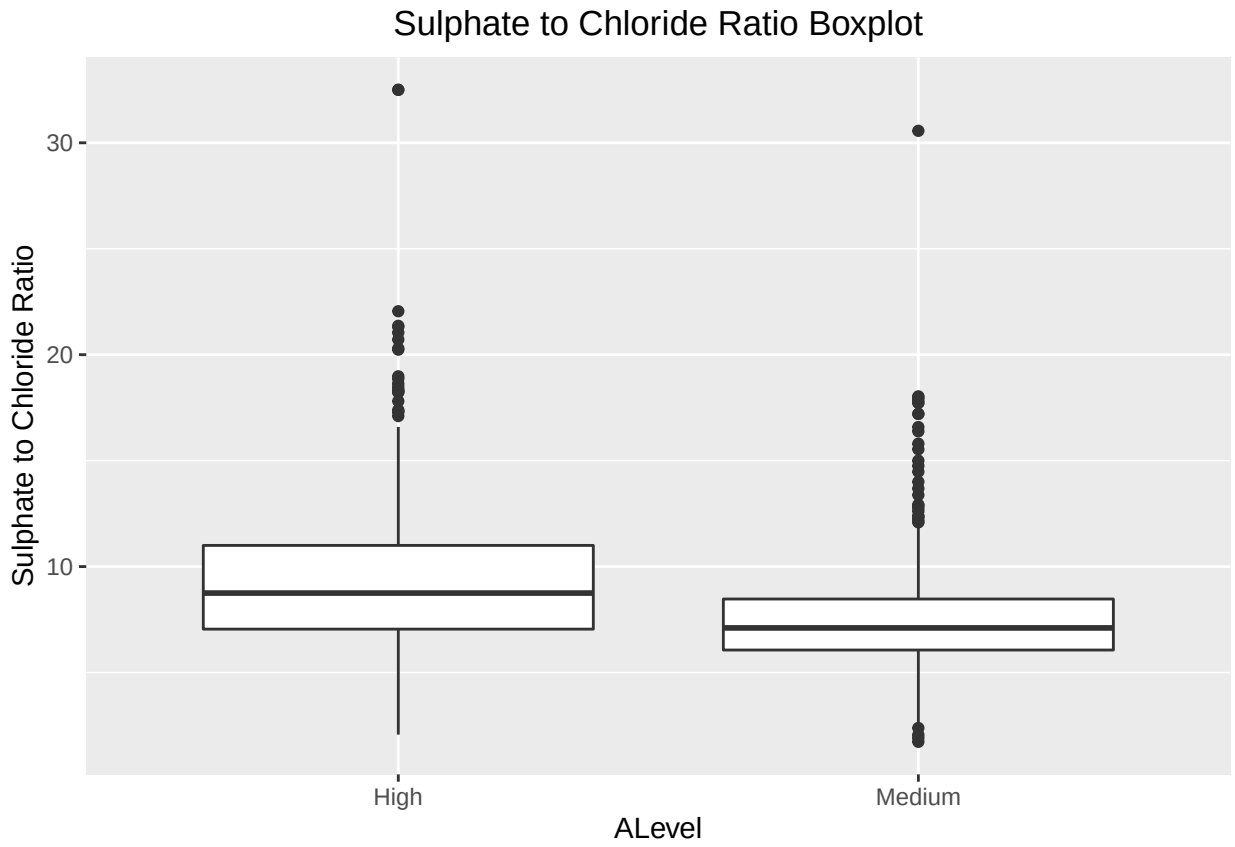
```
redwine$ALevel = " "
for (i in redwine$alcohol)
{
  if (i > 10.2)
    redwine$ALevel[which(redwine$alcohol == i)] = "High"
  else
    redwine$ALevel[which(redwine$alcohol == i)] = "Medium"
}

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              : chr  "Medium" "Medium" "Medium" "Medium" ...
```

Determines the sulphate to chloride ratio, as well as converting 'ALevel' into a factor, before plotting the two variables on a box plot.

```
sulphate_to_chloride = redwine$sulphates / redwine$chlorides
redwine$ALevel = as.factor(redwine$ALevel)
ggplot(redwine, aes(x = ALevel, y = sulphate_to_chloride)) +
  geom_boxplot() +
  labs(y = "Sulphate to Chloride Ratio", title = "Sulphate to Chloride Ratio Boxplot") +
  theme(plot.title = element_text(hjust = 0.5))
```



Determines the amount of red wine who's alcohol level is considered as 'High' or exceeding 10.2. The answer is 757.

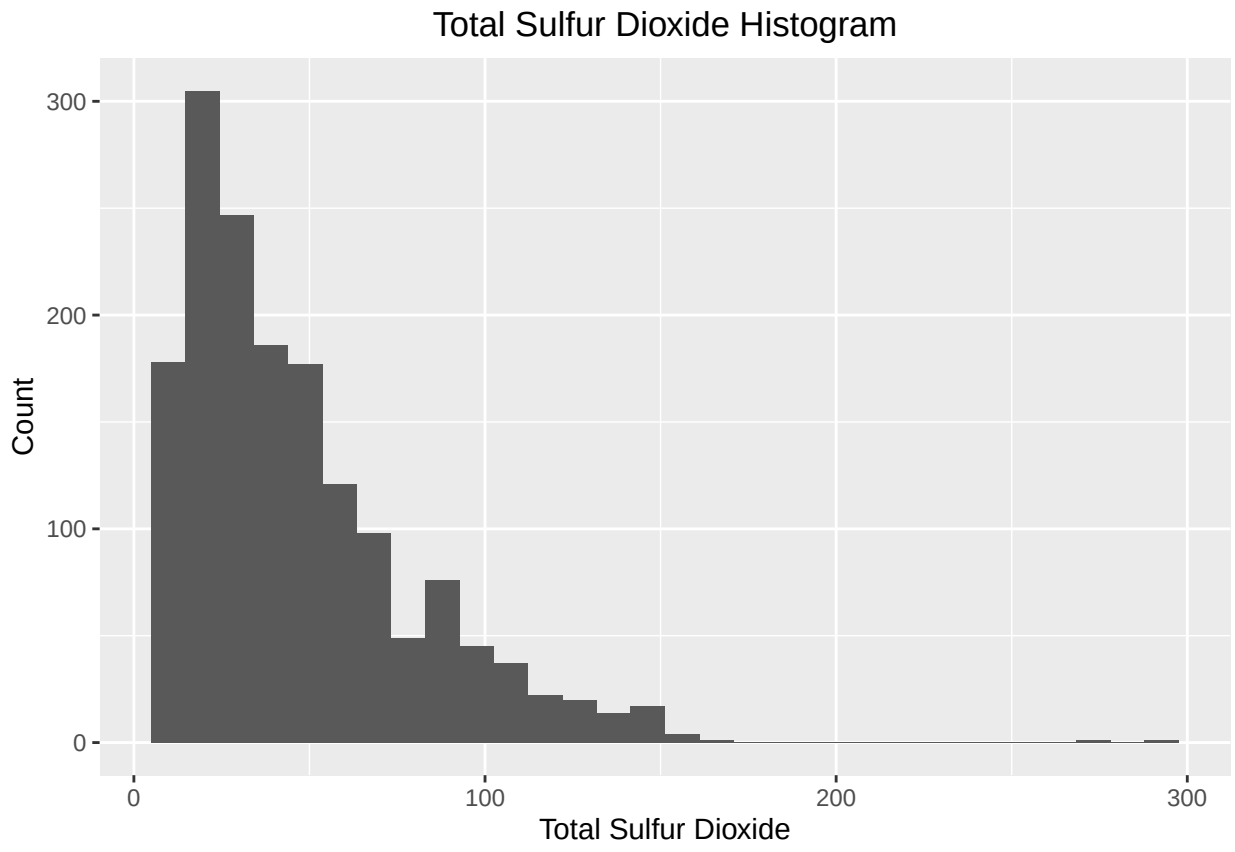
```
sum(redwine$ALevel == "High")
```

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

e.) A histogram of 'total_sulfur_dioxide' for both 'High' and 'Medium' 'ALevel' labels.

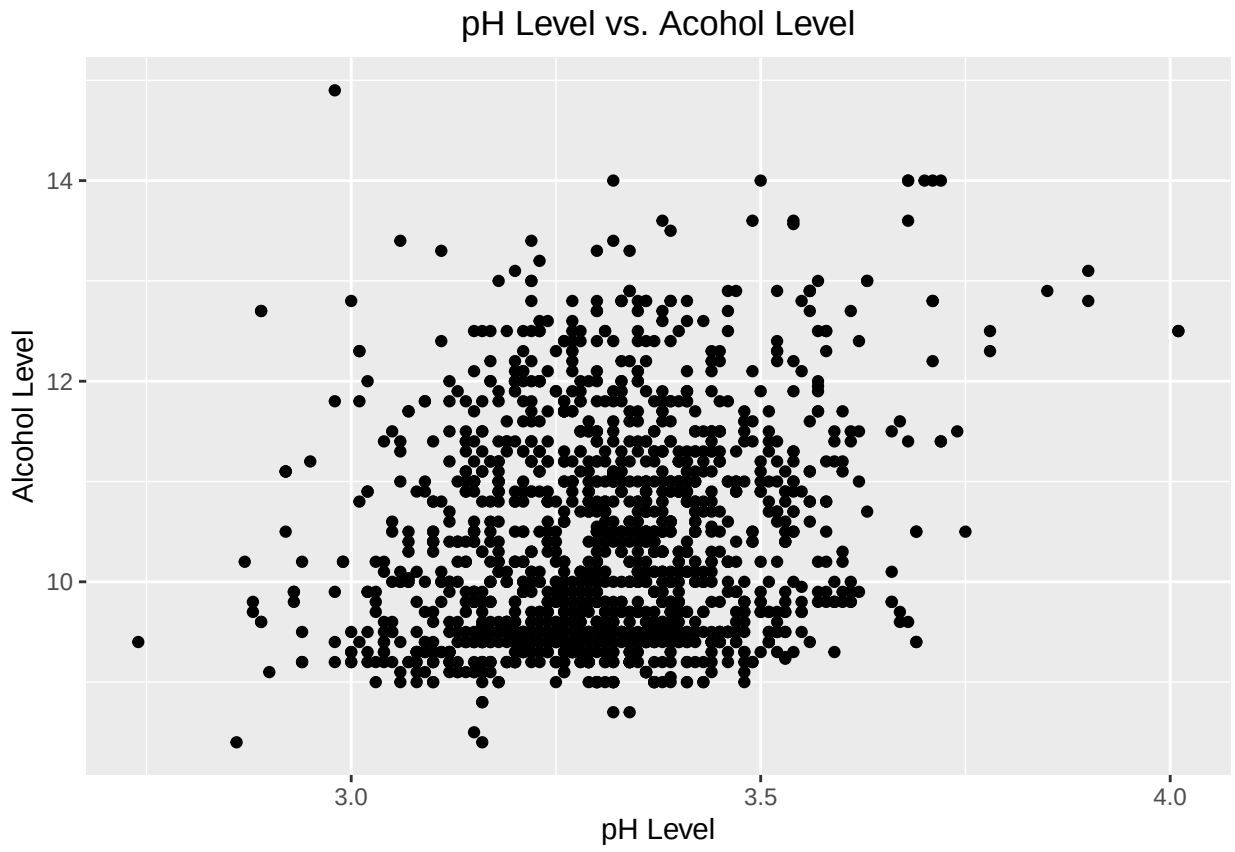
```
ggplot(redwine, aes(x = total_sulfur_dioxide)) +
  geom_histogram() +
  labs(x = "Total Sulfur Dioxide", y = "Count", title = "Total Sulfur Dioxide Histogram") +
  theme(plot.title = element_text(hjust = 0.5))
```

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



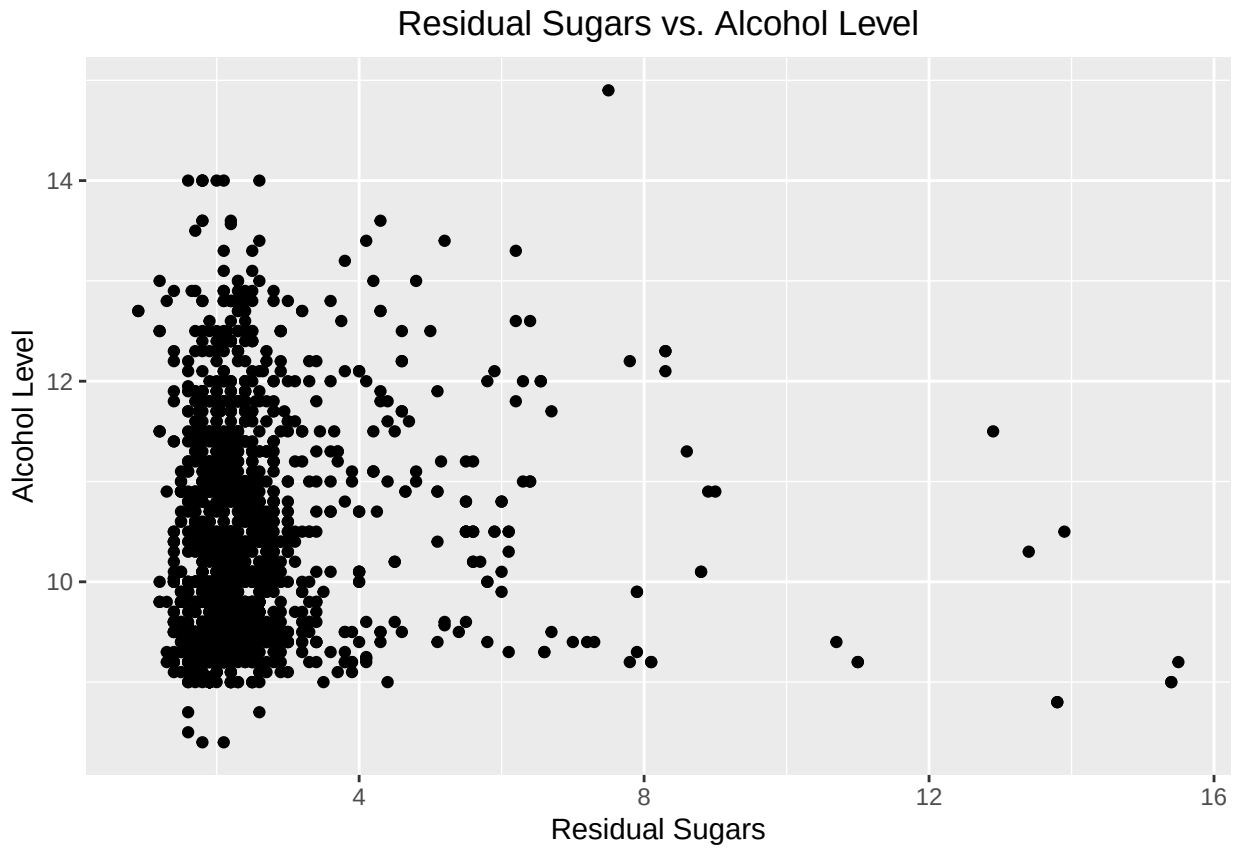
f.) Scatter plot of pH vs. alcohol level.

```
ggplot(redwine, aes(x = pH, y = alcohol)) +
  geom_point() +
  labs(x = "pH Level", y = "Alcohol Level", title = "pH Level vs. Acohol Level") +
  theme(plot.title = element_text(hjust = 0.5))
```



Scatter plot of residual sugar vs. alcohol level.

```
ggplot(redwine, aes(x = residual_sugar, y = alcohol)) +
  geom_point() +
  labs(x = "Residual Sugars", y = "Alcohol Level", title = "Residual Sugars vs. Alcohol Level") +
  theme(plot.title = element_text(hjust = 0.5))
```



From the two plots, it can be hypothesized that there are much more people who prefer acidic, less alcoholic and less sweet wine. According to the first plot, there is an abundant amount of people who prefer an acidic wine which is less alcoholic, and based on the second plot, more people preferred less sweetened wine regardless of whether they preferred strong or weaker alcohol.

Part 2

Forest Fires

The file, 'forestfires.csv' is read into 'forest_fires', and the column 'DC' in the data frame is converted into integer.

```
forest_fires = read.csv("forestfires.csv", sep = ",", header = TRUE)
forest_fires$DC = as.integer(forest_fires$DC)
```

```
## Warning: NAs introduced by coercion
```

```
summary(forest_fires)
```

```
##       X.2             X.1              X               Y          month
##  Min.   :  1   Min.   :  1   Min.   :1.000   Min.   :2.0   Length:517
```

```
##  1st Qu.:130   1st Qu.:130   1st Qu.:3.000   1st Qu.:4.0   Class :character
##  Median :259   Median :259   Median :4.000   Median :4.0   Mode  :character
##  Mean   :259   Mean   :259   Mean   :4.669   Mean   :4.3
##  3rd Qu.:388   3rd Qu.:388   3rd Qu.:7.000   3rd Qu.:5.0
##  Max.   :517   Max.   :517   Max.   :9.000   Max.   :9.0
##
##      day                 FFMC            DMC              DC
##  Length:517         Min.   :18.70   Min.   :  1.1   Min.   :  7.0
##  Class :character   1st Qu.:90.20   1st Qu.: 68.6   1st Qu.:436.0
##  Mode  :character   Median :91.60   Median :108.3   Median :664.0
##                     Mean   :90.64   Mean   :110.9   Mean   :547.4
##                     3rd Qu.:92.90   3rd Qu.:142.4   3rd Qu.:713.0
##                     Max.   :96.20   Max.   :291.3   Max.   :860.0
##                                                     NA's   :1
##       ISI              temp             RH              wind
##  Min.   : 0.000   Min.   : 2.20   Min.   : 15.00   Min.   :0.400
##  1st Qu.: 6.500   1st Qu.:15.50   1st Qu.: 33.00   1st Qu.:2.700
##  Median : 8.400   Median :19.30   Median : 42.00   Median :4.000
##  Mean   : 9.022   Mean   :18.89   Mean   : 44.29   Mean   :4.018
##  3rd Qu.:10.800   3rd Qu.:22.80   3rd Qu.: 53.00   3rd Qu.:4.900
##  Max.   :56.100   Max.   :33.30   Max.   :100.00   Max.   :9.400
##
##       rain              area
##  Min.   :0.00000   Min.   :   0.00
##  1st Qu.:0.00000   1st Qu.:   0.00
##  Median :0.00000   Median :   0.52
##  Mean   :0.02166   Mean   :  12.85
##  3rd Qu.:0.00000   3rd Qu.:   6.57
##  Max.   :6.40000   Max.   :1090.84
##
```

a.) X is Quantitative, Y is Quantitative, month is Qualitative, day is Qualitative, FFMC is Quantitative, DMC is Quantitative, DC is Quantitative, ISI is Quantitative, temp is Quantitative, RH is Quantitative, wind is Quantitative, rain is Quantitative, area is Quantitative.

b.) Determines the ranges, means and standard deviations for each column of 'forest_fires', and a data frame for all three were made using the data calculated.

```
predictor_r = c("X min", "X max", "Y min", "Y max", "FFMC min", "FFMC max", "DMC min", "DMC max", "DC m
                "ISI min", "ISI max", "temp min", "temp max", "RH min", "RH max", "wind min", "wind max
                "rain max", "area min", "area max")

range = c(range(forest_fires$X), range(forest_fires$Y), range(forest_fires$FFMC), range(forest_fires$DM
          range(forest_fires$DC), range(forest_fires$ISI), range(forest_fires$temp), range(forest_fires
          range(forest_fires$wind), range(forest_fires$rain), range(forest_fires$area))
ranges = data.frame(predictor_r, range)


predictor = c("X", "Y", "FFMC", "DMC", "DC", "ISI", "temp", "RH", "wind", "rain", "area")

mean = c(mean(forest_fires$X), mean(forest_fires$Y), mean(forest_fires$FFMC), mean(forest_fires$DMC),
         mean(forest_fires$DC), mean(forest_fires$ISI), mean(forest_fires$temp), mean(forest_fires$RH),
         mean(forest_fires$wind), mean(forest_fires$rain), mean(forest_fires$area))
means = data.frame(predictor, mean)
```

```
standard_deviation = c(sd(forest_fires$X), sd(forest_fires$Y), sd(forest_fires$FFMC), sd(forest_fires$DM
                       sd(forest_fires$DC), sd(forest_fires$ISI), sd(forest_fires$temp), sd(forest_fires
                       sd(forest_fires$wind), sd(forest_fires$rain), sd(forest_fires$area))
stand_devs = data.frame(predictor, standard_deviation)

str(ranges)
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ predictor_r: chr  "X min" "X max" "Y min" "Y max" ...
##  $ range      : num  1 9 2 9 18.7 ...
```

```
str(means)
```

```
## 'data.frame':    11 obs. of  2 variables:
##  $ predictor: chr  "X" "Y" "FFMC" "DMC" ...
##  $ mean     : num  4.67 4.3 90.64 110.87 NA ...
```

```
str(stand_devs)
```

```
## 'data.frame':    11 obs. of  2 variables:
##  $ predictor         : chr  "X" "Y" "FFMC" "DMC" ...
##  $ standard_deviation: num  2.31 1.23 5.52 64.05 NA ...
```

Determines the most frequency of days of the week in which a forest fire occurred and creates a vector 'week'. And as the vector is sorted from Monday to Sunday, Sunday has the highest frequency.

```
week = c(sum(forest_fires$day == "mon"), sum(forest_fires$day == "tue"), sum(forest_fires$day == "wed")
         sum(forest_fires$day == "thu"), sum(forest_fires$day == "fri"), sum(forest_fires$day == "sat")
         sum(forest_fires$day == "sun"))

week
```

```
## [1] 74 64 54 61 85 84 95
```

c.) Determines the same thing as in b.), however not putting into account rows 40-80 of the 'forest_fire' data frame.

```
new_range = c(range(forest_fires$X[c(1:39, 81:517)]), range(forest_fires$Y[c(1:39, 81:517)]),
              range(forest_fires$FFMC[c(1:39, 81:517)]), range(forest_fires$DMC[c(1:39, 81:517)]),
              range(forest_fires$DC[c(1:39, 81:517)]), range(forest_fires$ISI[c(1:39, 81:517)]),
              range(forest_fires$temp[c(1:39, 81:517)]), range(forest_fires$RH[c(1:39, 81:517)]),
              range(forest_fires$wind[c(1:39, 81:517)]), range(forest_fires$rain[c(1:39, 81:517)]),
              range(forest_fires$area[c(1:39, 81:517)]))
new_ranges = data.frame(predictor_r, new_range)

new_mean = c(mean(forest_fires$X[c(1:39, 81:517)]), mean(forest_fires$Y[c(1:39, 81:517)]),
             mean(forest_fires$FFMC[c(1:39, 81:517)]), mean(forest_fires$DMC[c(1:39, 81:517)]),
             mean(forest_fires$DC[c(1:39, 81:517)]), mean(forest_fires$ISI[c(1:39, 81:517)]),
             mean(forest_fires$temp[c(1:39, 81:517)]), mean(forest_fires$RH[c(1:39, 81:517)]),
             mean(forest_fires$wind[c(1:39, 81:517)]), mean(forest_fires$rain[c(1:39, 81:517)]),
```

```
             mean(forest_fires$area[c(1:39, 81:517)]))
new_means = data.frame(predictor, new_mean)

new_standard_deviation = c(sd(forest_fires$X[c(1:39, 81:517)]), sd(forest_fires$Y[c(1:39, 81:517)]),
                           sd(forest_fires$FFMC[c(1:39, 81:517)]), sd(forest_fires$DMC[c(1:39, 81:517)]),
                           sd(forest_fires$DC[c(1:39, 81:517)]), sd(forest_fires$ISI[c(1:39, 81:517)]),
                           sd(forest_fires$temp[c(1:39, 81:517)]), sd(forest_fires$RH[c(1:39, 81:517)]),
                           sd(forest_fires$wind[c(1:39, 81:517)]), sd(forest_fires$rain[c(1:39, 81:517)]),
                           sd(forest_fires$area[c(1:39, 81:517)]))
new_stand_devs = data.frame(predictor, new_standard_deviation)

str(new_ranges)
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ predictor_r: chr  "X min" "X max" "Y min" "Y max" ...
##  $ new_range  : num  1 9 2 9 18.7 ...
```

```
str(new_means)
```

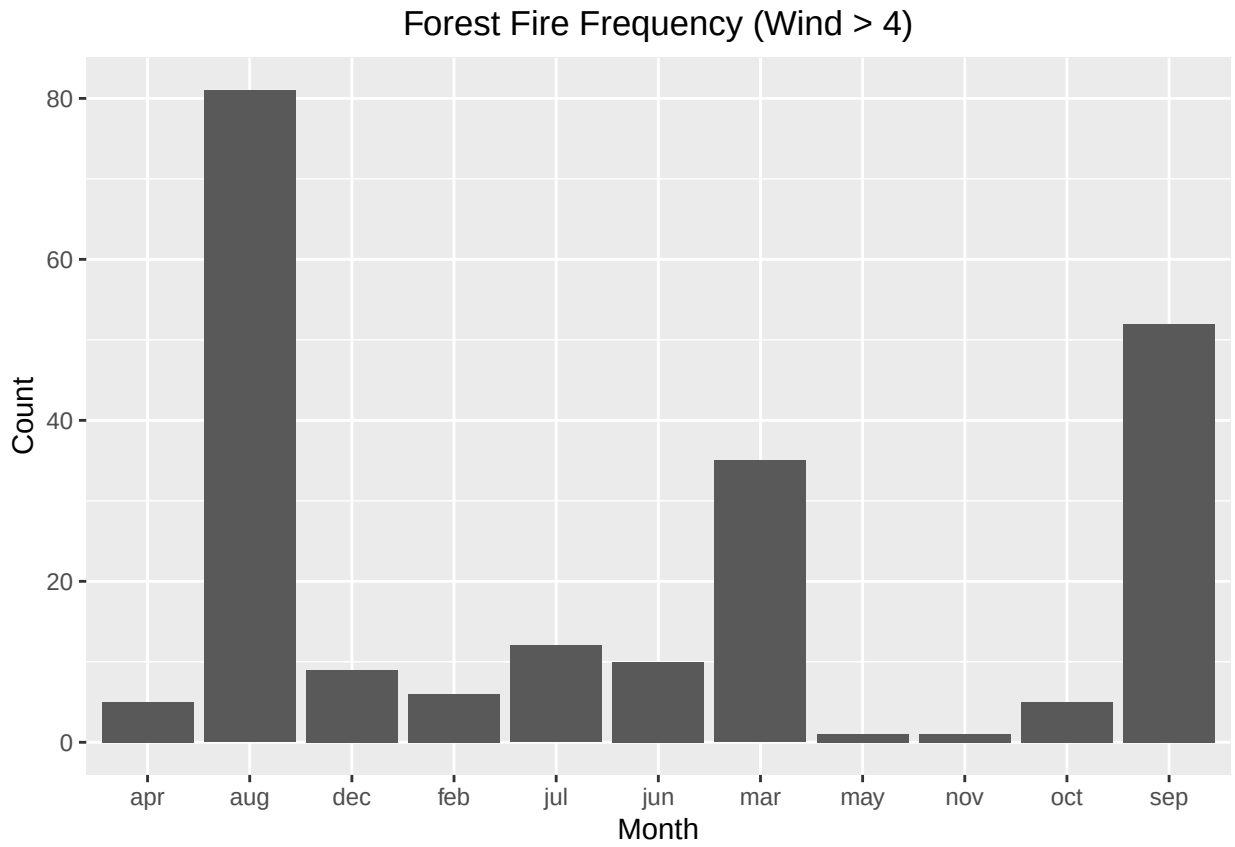
```
## 'data.frame':    11 obs. of  2 variables:
##  $ predictor: chr  "X" "Y" "FFMC" "DMC" ...
##  $ new_mean : num  4.76 4.36 90.66 113.47 NA ...
```

```
str(new_stand_devs)
```

```
## 'data.frame':    11 obs. of  2 variables:
##  $ predictor             : chr  "X" "Y" "FFMC" "DMC" ...
##  $ new_standard_deviation: num  2.34 1.16 5.68 65.05 NA ...
```

d.) Generates a bar plot showing the frequency of forest forest in each month with a wind speed of greater than 4 (wind > 4). As such, the plot shows that the month August has the most forest fires with wind > 4.

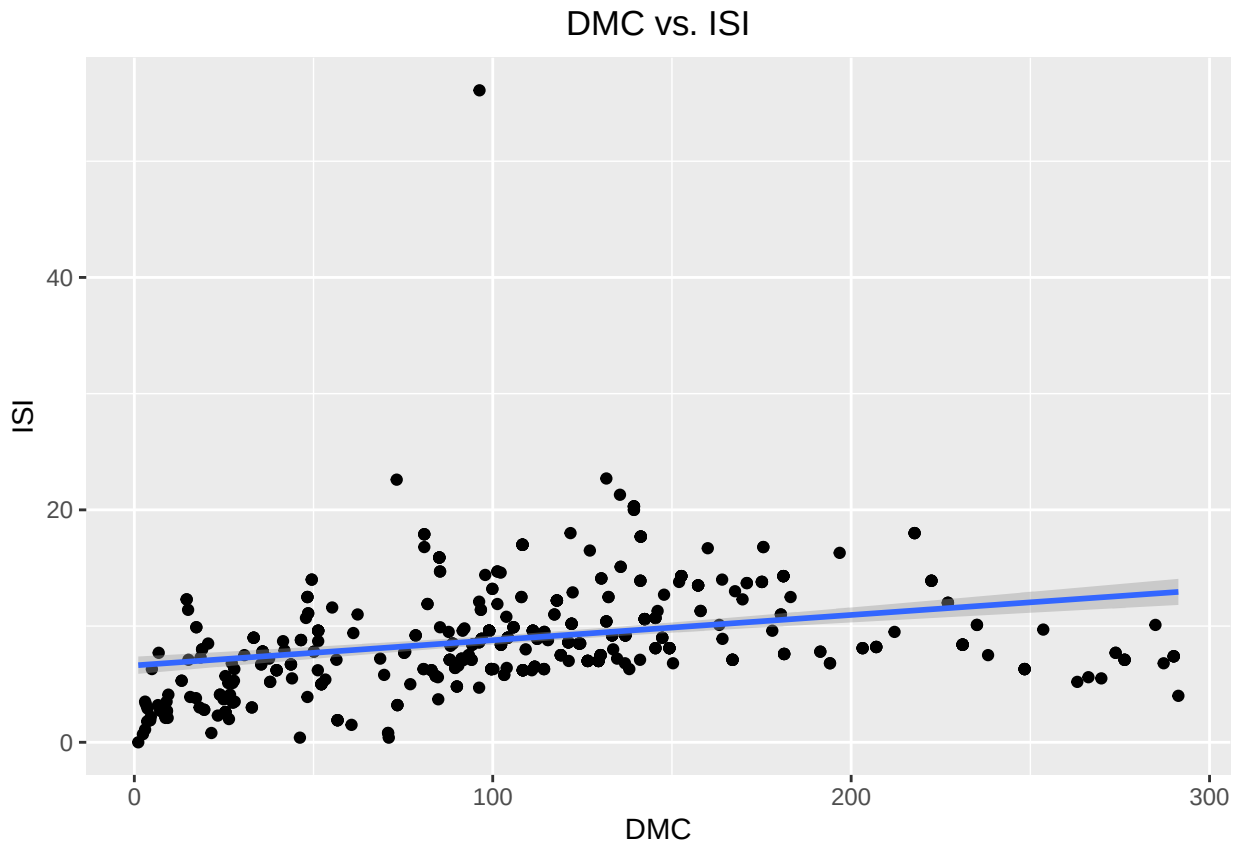
```
filter_data = forest_fires[(forest_fires$wind > 4), ]$month
forest_fires_new = data.frame(filter_data)

ggplot(forest_fires_new, aes(x = filter_data)) +
  geom_bar() +
  labs(x = "Month", y = "Count", title = "Forest Fire Frequency (Wind > 4)") +
  theme(plot.title = element_text(hjust = 0.5))
```



e.) By plotting on a scatter plot, the Duff Moisture Code Index (DMC) with the Initial Spread Index (ISI), a correlation matrix with these two variables was made.

```
ggplot(forest_fires, aes(x = DMC, y = ISI)) +
  geom_point() +
  labs(x = "DMC", y = "ISI", title = "DMC vs. ISI") +
  geom_smooth(method = 'lm') +
  theme(plot.title = element_text(hjust = 0.5))
```

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




```
correlation_matrix = matrix(0, nrow = length(forest_fires$DMC))
correlation_matrix[,1] = correlation_matrix[,1] + 1:517
correlation_matrix = cbind(correlation_matrix, forest_fires$DMC)
correlation_matrix = cbind(correlation_matrix, forest_fires$ISI)
colnames(correlation_matrix) = c("No.", "DMC", "ISI")

str(correlation_matrix)
```

```
##  num [1:517, 1:3] 1 2 3 4 5 6 7 8 9 10 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "No." "DMC" "ISI"
```

f.)

In order to predict wind speed based on the other variables, pairing it with FFMC, DMC, ISI,temp, and/or RH would be useful. This is because creating a plot with both wind speed and one other variables shows a set of points all relating to a wind speed staying constant regardless of the other variable at random. In turn, this allows us to better understand the wind speed as it may stay constant for several points of the other variable, or it might change.