Data Visualization

```
#Importing the required libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching packages ──────────────────────────────────────────────────────
──────────────── tidyverse 1.3.0 ──
```

```
## ✓ tibble  2.1.3     ✓ purrr   0.3.3
## ✓ tidyr   1.0.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
```

```
## ── Conflicts ─────────────────────────────────────────────────────────────
──────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(grid)
library(RColorBrewer)
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(scatterplot3d)
library(plotrix)
library(rlang)
```

```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, list_along, modify, prepend,
##     splice
```

```
library(dataQualityR)
```

Problem 1: (Forest Fires)

```
#loading the csv
df_ForestFires <- read.csv('forestfires.csv')
head(df_ForestFires)
```

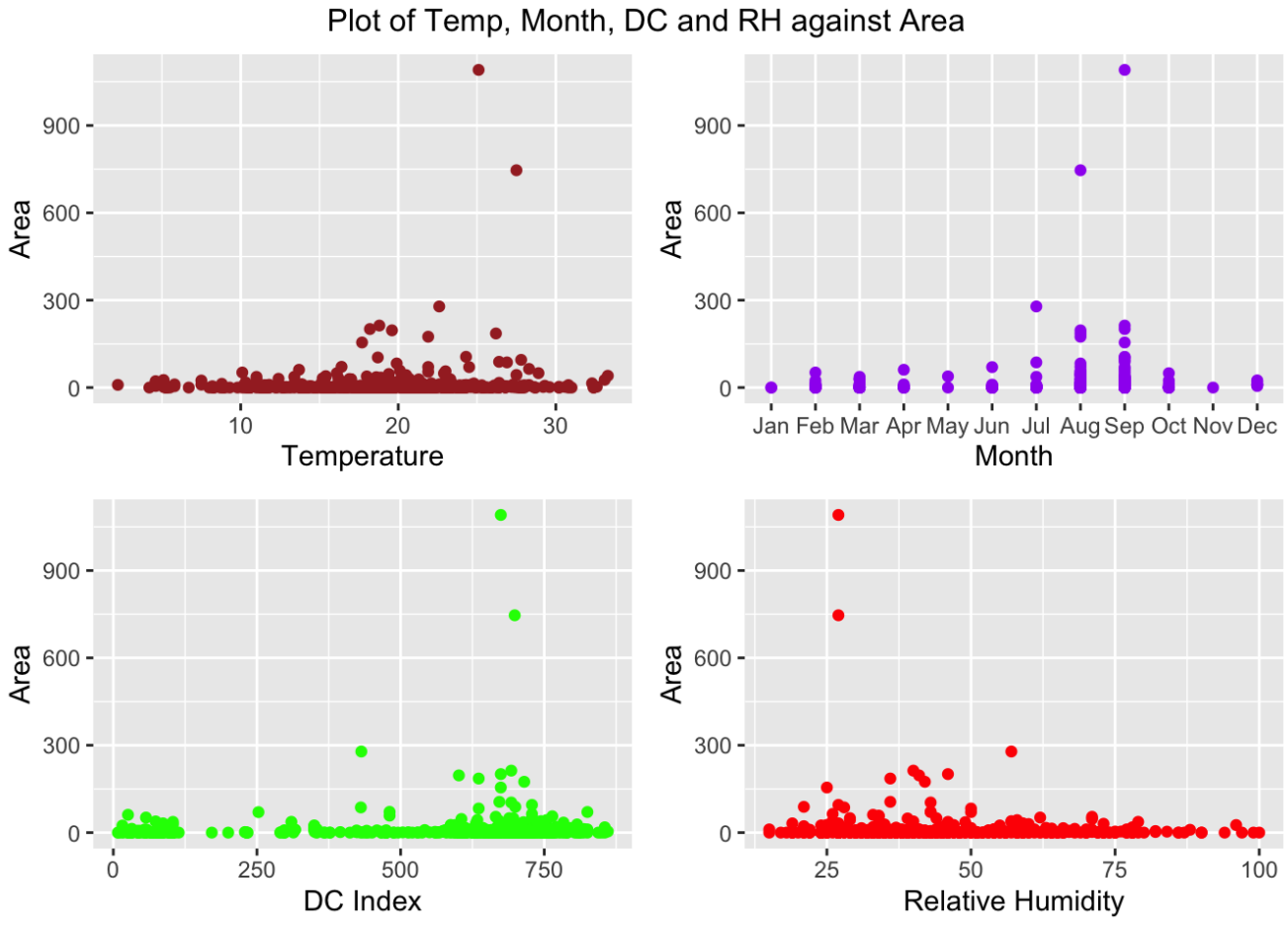
```
##   X Y month day FFMC  DMC    DC  ISI temp RH wind rain area
## 1 7 5   mar fri 86.2 26.2  94.3  5.1  8.2 51  6.7  0.0    0
## 2 7 4   oct tue 90.6 35.4 669.1  6.7 18.0 33  0.9  0.0    0
## 3 7 4   oct sat 90.6 43.7 686.9  6.7 14.6 33  1.3  0.0    0
## 4 8 6   mar fri 91.7 33.3  77.5  9.0  8.3 97  4.0  0.2    0
## 5 8 6   mar sun 89.3 51.3 102.2  9.6 11.4 99  1.8  0.0    0
## 6 8 6   aug sun 92.3 85.3 488.0 14.7 22.2 29  5.4  0.0    0
```

```
summary(df_ForestFires)
```

```
##        X               Y           month      day          FFMC      
##  Min.   :1.000   Min.   :2.0   aug    :184   fri:85   Min.   :18.70  
##  1st Qu.:3.000   1st Qu.:4.0   sep    :172   mon:74   1st Qu.:90.20  
##  Median :4.000   Median :4.0   mar    : 54   sat:84   Median :91.60  
##  Mean   :4.669   Mean   :4.3   jul    : 32   sun:95   Mean   :90.64  
##  3rd Qu.:7.000   3rd Qu.:5.0   feb    : 20   thu:61   3rd Qu.:92.90  
##  Max.   :9.000   Max.   :9.0   jun    : 17   tue:64   Max.   :96.20  
##                                (Other): 38   wed:54                  
##       DMC              DC              ISI              temp      
##  Min.   :  1.1   Min.   :  7.9   Min.   : 0.000   Min.   : 2.20  
##  1st Qu.: 68.6   1st Qu.:437.7   1st Qu.: 6.500   1st Qu.:15.50  
##  Median :108.3   Median :664.2   Median : 8.400   Median :19.30  
##  Mean   :110.9   Mean   :547.9   Mean   : 9.022   Mean   :18.89  
##  3rd Qu.:142.4   3rd Qu.:713.9   3rd Qu.:10.800   3rd Qu.:22.80  
##  Max.   :291.3   Max.   :860.6   Max.   :56.100   Max.   :33.30  
##                                                                  
##        RH              wind            rain              area        
##  Min.   : 15.00   Min.   :0.400   Min.   :0.00000   Min.   :   0.00  
##  1st Qu.: 33.00   1st Qu.:2.700   1st Qu.:0.00000   1st Qu.:   0.00  
##  Median : 42.00   Median :4.000   Median :0.00000   Median :   0.52  
##  Mean   : 44.29   Mean   :4.018   Mean   :0.02166   Mean   :  12.85  
##  3rd Qu.: 53.00   3rd Qu.:4.900   3rd Qu.:0.00000   3rd Qu.:   6.57  
##  Max.   :100.00   Max.   :9.400   Max.   :6.40000   Max.   :1090.84  
##
```

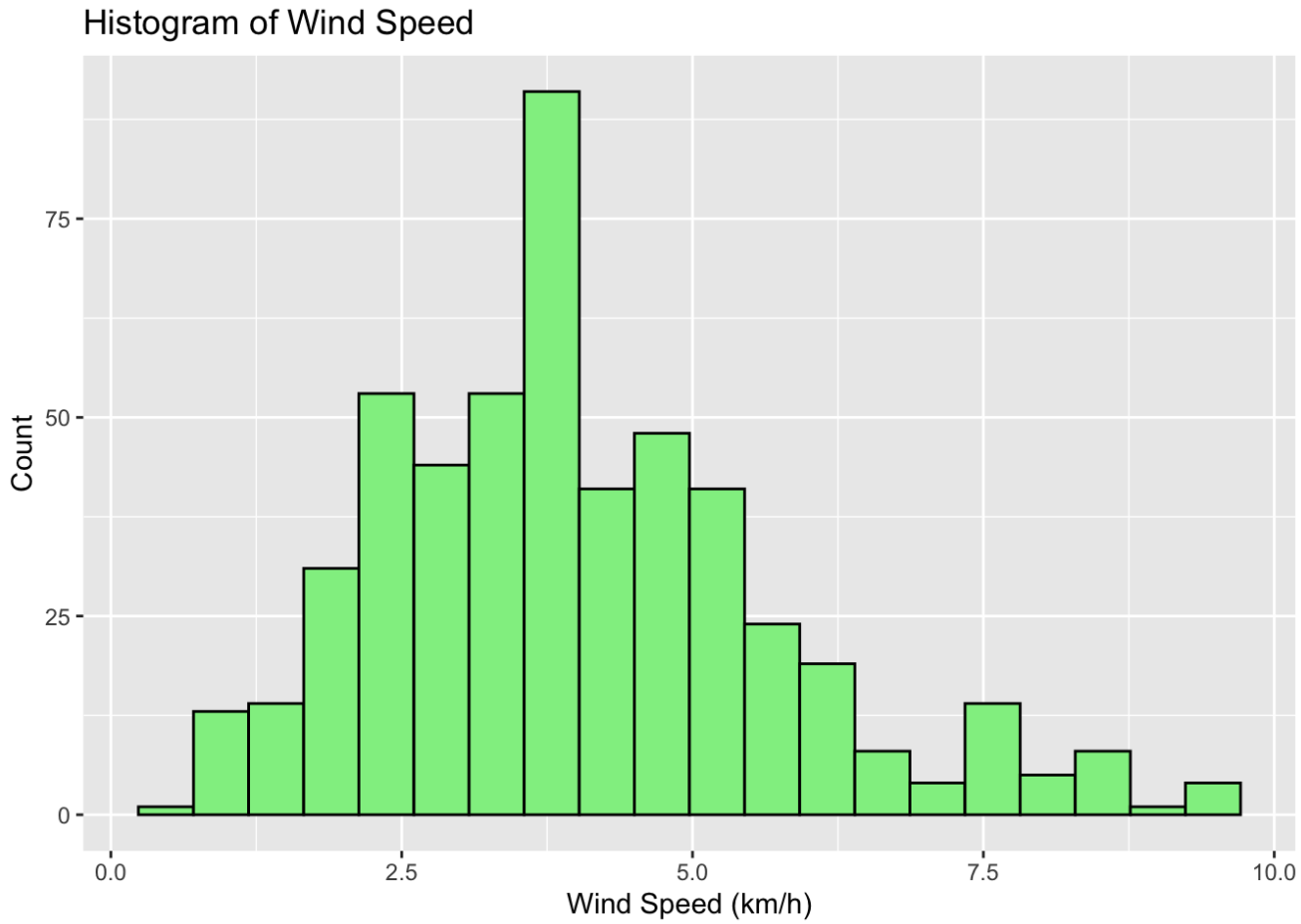
a.

```
#Plot of Temperature v/s Area
plot_temp <- ggplot(df_ForestFires) +
  geom_point(mapping = aes(x = temp, y = area), colour = "brown") +
  labs(x = "Temperature", y = "Area")

#Plot of Month v/s Area
df_ForestFires$month <- factor(df_ForestFires$month, levels = c("jan", "feb", "mar", "apr"
, "may", "jun", "jul", "aug","sep", "oct", "nov", "dec"), ordered = TRUE)
plot_month <- ggplot(df_ForestFires) +
  geom_point(mapping = aes(x = month, y = area), colour = "purple") +
  labs(x = "Month", y = "Area") +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug","Sep"
, "Oct", "Nov", "Dec"))

#Plot of DC v/s Area
plot_DC <- ggplot(df_ForestFires) +
  geom_point(mapping = aes(x = DC, y = area), colour = "green") +
  labs(x = "DC Index", y = "Area")

#Plot of RH v/s Area
plot_RH <- ggplot(df_ForestFires) +
  geom_point(mapping = aes(x = RH, y = area), colour = "red") +
  labs(x = "Relative Humidity", y = "Area")

#Arranging all the plots in a 2 * 2 matrix
grid.arrange(plot_temp, plot_month, plot_DC, plot_RH, ncol = 2, nrow = 2, top = textGrob(
"Plot of Temp, Month, DC and RH against Area"))
```



b.

```
ggplot(df_ForestFires, mapping = aes(x = wind)) +
  geom_histogram(bins = 20, fill = "light green", color = "black") +
  labs(x = "Wind Speed (km/h)", y = "Count", title = "Histogram of Wind Speed")
```



c.

Summary statistics of wind:

```
#summary statistics by using individual functions of r
minimum_wind <- min(df_ForestFires$wind)
cat("Minimum wind speed:", minimum_wind, "km/h \n")
```

```
## Minimum wind speed: 0.4 km/h
```

```
Q1_wind <- quantile(df_ForestFires$wind, 0.25)
cat("First quantile wind speed:", Q1_wind, "km/h \n")
```

```
## First quantile wind speed: 2.7 km/h
```

```
median_wind <- quantile(df_ForestFires$wind, 0.50)
cat("Median wind speed:", median_wind, "km/h \n")
```

```
## Median wind speed: 4 km/h
```

```
mean_wind <- mean(df_ForestFires$wind)
cat("Mean wind speed:", mean_wind, "km/h \n")
```

```
## Mean wind speed: 4.017602 km/h
```

```
Q3_wind <- quantile(df_ForestFires$wind, 0.75)
cat("Third quantile wind speed:", Q3_wind, "km/h \n")
```

```
## Third quantile wind speed: 4.9 km/h
```

```
maximum_wind <- max(df_ForestFires$wind)
cat("Maximum wind speed:", maximum_wind, "km/h \n")
```

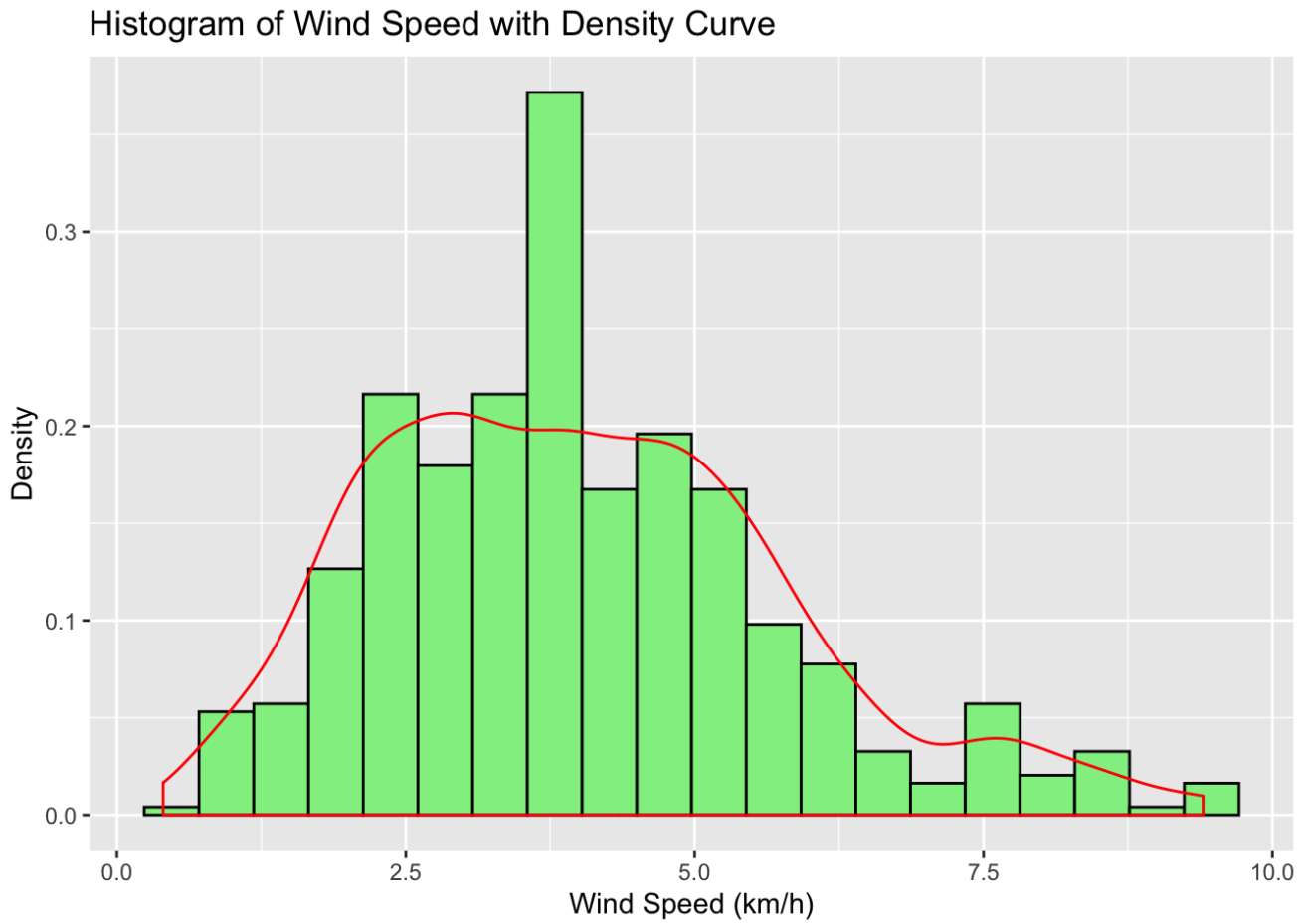
```
## Maximum wind speed: 9.4 km/h
```

```
#summary statistics by using summary function of r
summary(df_ForestFires$wind)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.400   2.700   4.000   4.018   4.900   9.400
```

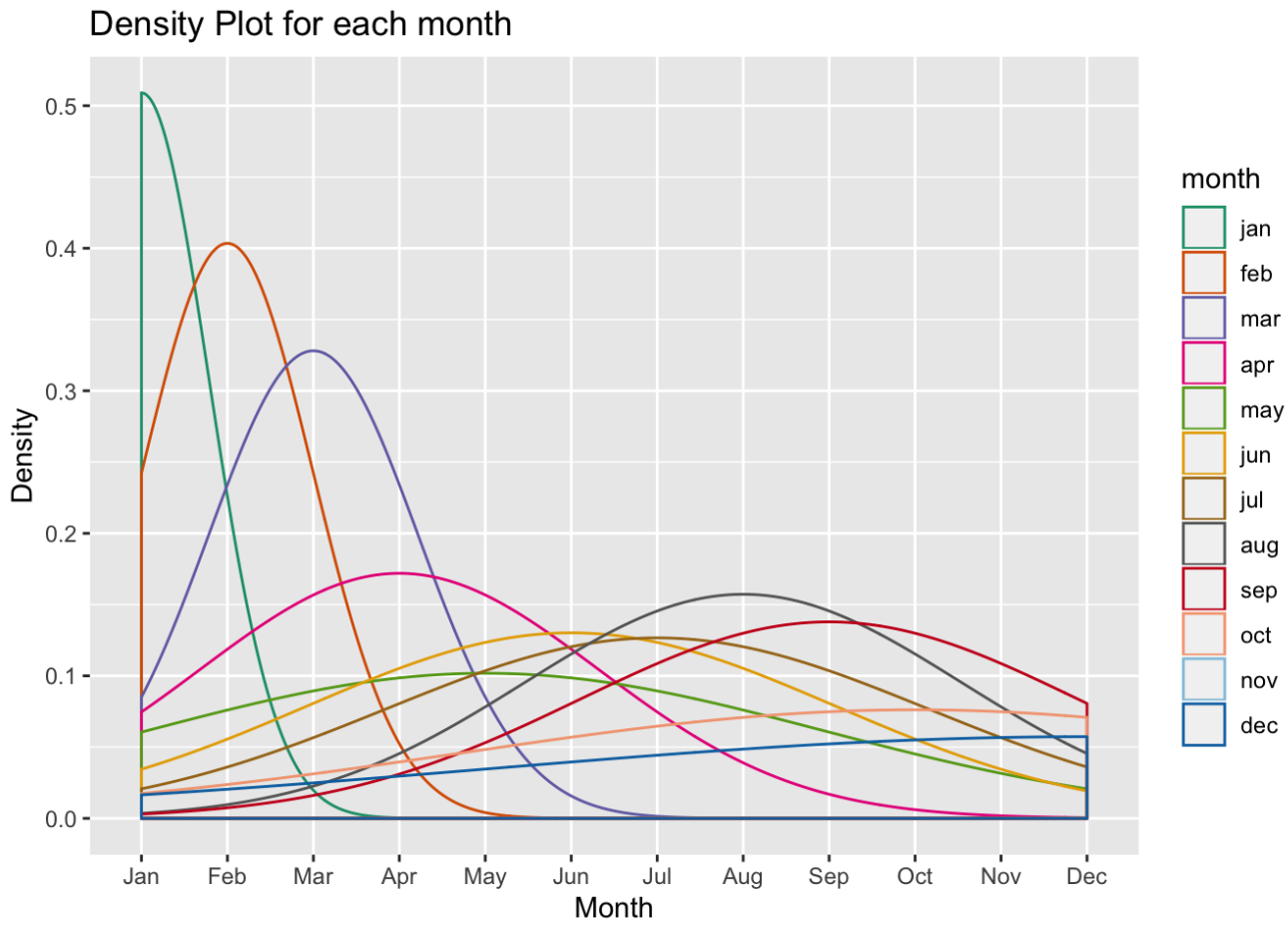
d.

```
#adding the density line to the histogram
ggplot(df_ForestFires, mapping = aes(x = wind)) +
  geom_histogram(mapping = aes(y = ..density..), bins = 20, fill = "light green", color =
"black") +
  geom_density(color = "red") +
  labs(x = "Wind Speed (km/h)", y = "Density", title = "Histogram of Wind Speed with Densi
ty Curve")
```



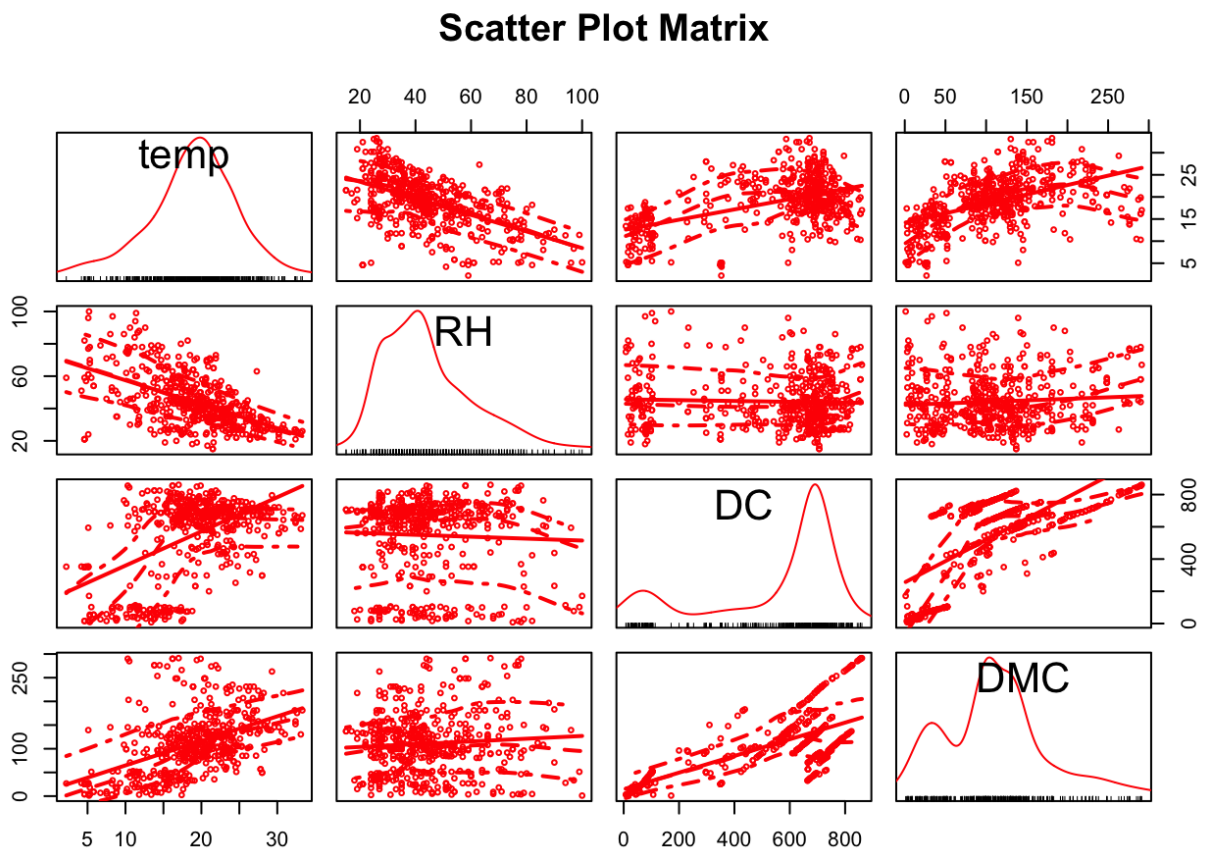
e.

```
#selecting colours for the density plots for each month
my_colors <- c(brewer.pal(name = "Dark2", n = 8), brewer.pal(name = "RdBu", n = 4))

ggplot(df_ForestFires, aes(x = df_ForestFires$month, colour = month)) +
  geom_density() +
  labs(x = "Month", y = "Density", title = "Density Plot for each month") +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug","Sep"
, "Oct", "Nov", "Dec")) +
   scale_color_manual(values = my_colors)
```



f.

```
#Scatter Plot for the given columns
scatterplotMatrix(~temp + RH + DC + DMC, data = df_ForestFires, spread = FALSE, lty.smooth
= 2, cex = 0.5, col = "red", main = "Scatter Plot Matrix")
```



Interpretation:

The following can be interpreted regarding the correlation between variables plotted on the scatter plot matrix:

1. RH has a moderate negative correlation with **temp**.

2. DC has a weak positive correlation with **temp**.

3. DMC has a weak positive correlation with **temp**.

4. DC has a no correlation with **RH**.

5. DMC has a no correlation with **RH**.

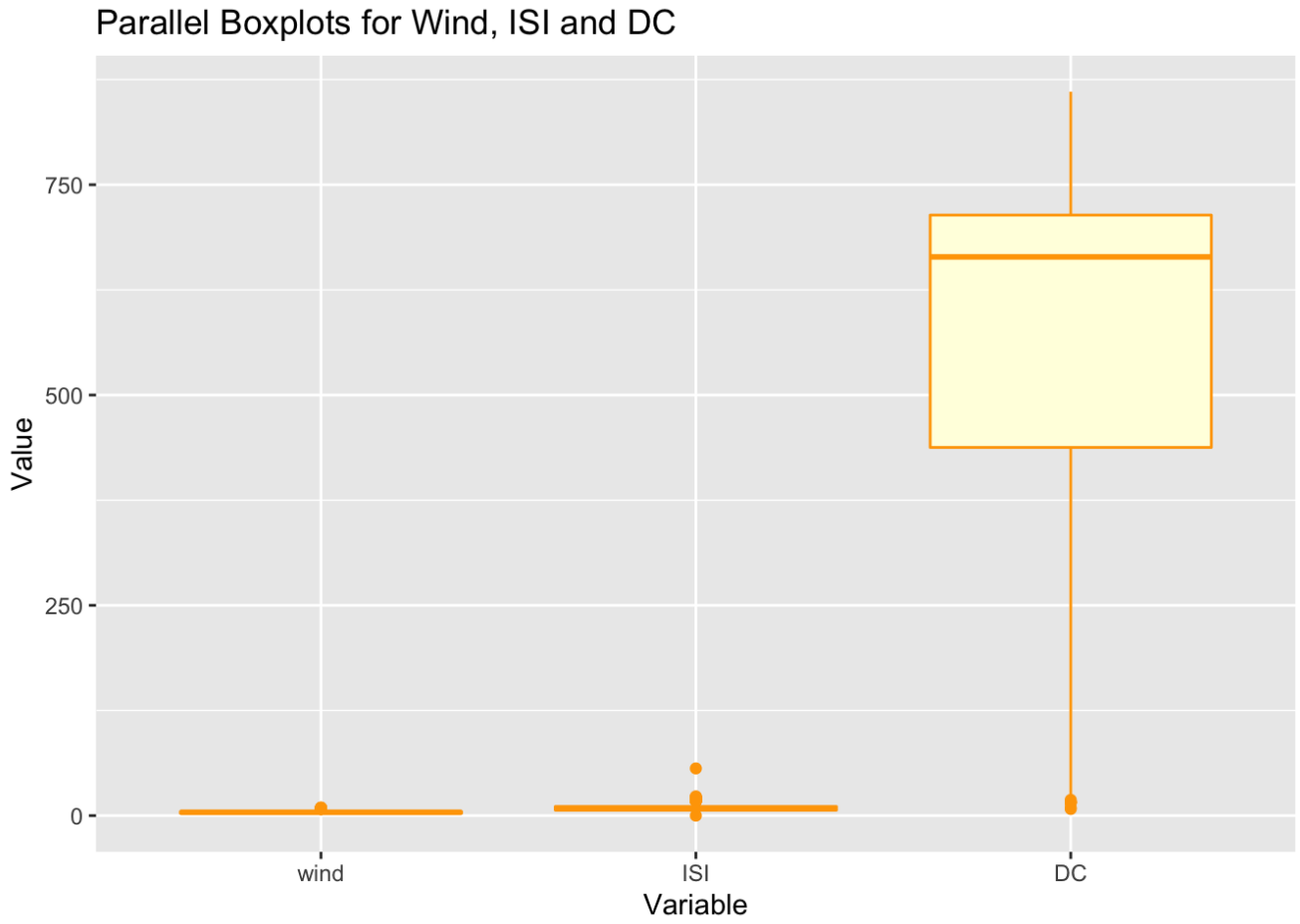
6. DMC has a moderate positive correlation with **DC**.

g.

```
#creating a new dataframe having the columns, wind, ISI and DC
df_boxplot <- df_ForestFires[c("wind","ISI","DC")]
#head(df_boxplot)

#using the melt function to take data in wide format and stack as a set of columns into a
 single column of data
meltData <- melt(df_boxplot)
```

```
## Using  as id variables
```

```
#head(meltData)

ggplot(meltData, aes(variable, value)) +
  geom_boxplot(color = "orange", fill = "light yellow") +
  labs(x = "Variable", y = "Value", title = "Parallel Boxplots for Wind, ISI and DC")
```

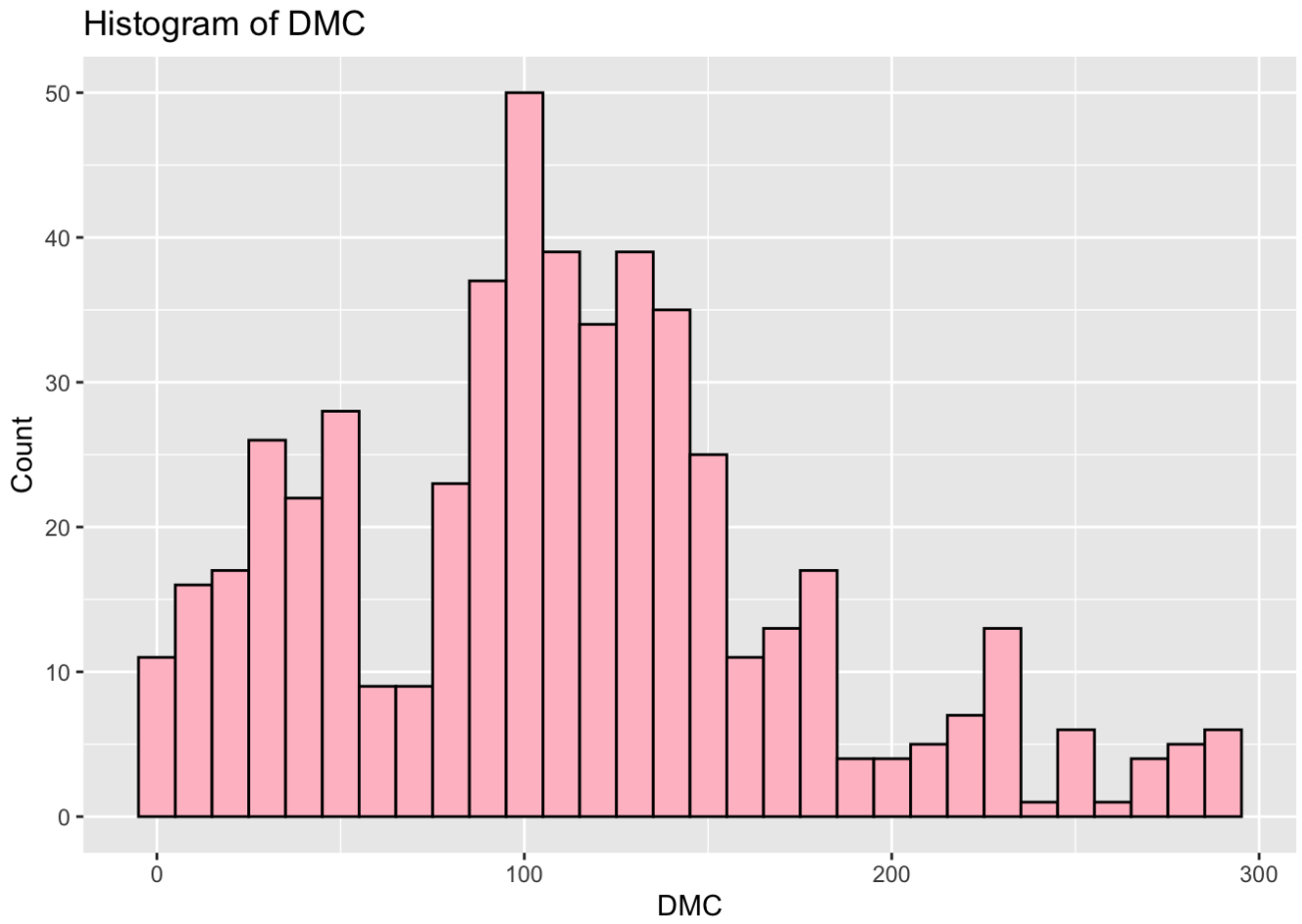


Interpretation:

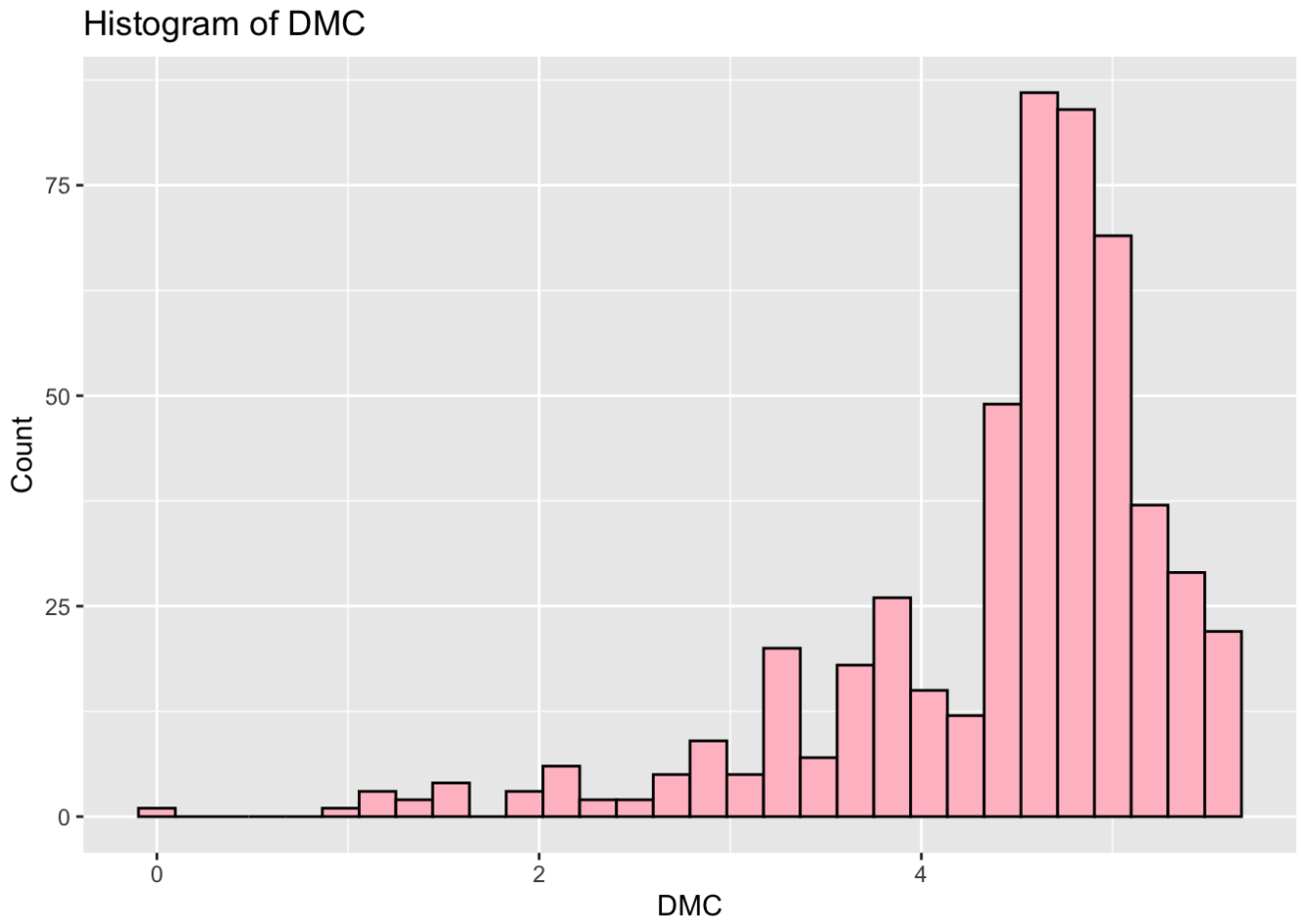
By plotting parallel boxplots, the data from three distributions are displayed in the same chart using the same measurement scale. Yes, there are outliers in all three distributions but not too many. The distribution of wind and ISI is almost similar but very different from DC. Also, the distribution of DC is left-skewed.

h.

```
ggplot(df_ForestFires, mapping = aes(x = DMC)) +
  geom_histogram(bins = 30, fill = "pink", color = "black") +
  labs(x = "DMC", y = "Count", title = "Histogram of DMC")
```




```
#Histogram of log transformation of DMC
ggplot(df_ForestFires, mapping = aes(x = log(DMC))) +
  geom_histogram(bins = 30, fill = "pink", color = "black") +
  labs(x = "DMC", y = "Count", title = "Histogram of DMC")
```



Interpretation:

Log transformation is generally used to transform skewed data to approximately normal. It is assumed that if the original data follows log-normal distribution approximately, then the log-transformed data will approximately follow normal distribution. When we applied log to DMC, the data became more skewed. The skewness can be seen towards the left. Infact, log transformation aggravated the problem of skewness in this scenario. Thus, we can say that orginal DMC data does not follow log-normal distribution.

Problem 2: (Twitter Accounts)

```
#loading the csv
df_Twitter <- read.csv('M01_quasi_twitter.csv')
head(df_Twitter)
```

```
##   screen_name created_at_month created_at_day created_at_year country
## 1         CNN                2              9            2007     USA
## 2      osbrFe               11             21            2009   India
## 3         WSJ                4              1            2007   India
## 4        ninc                3             24            2007     USA
## 5    nssubies                4             23            2009     USA
## 6        BNCC                2              9            2009 England
##         location friends_count followers_count statuses_count favourites_count
## 1  Miami Florida          1087        22187643          60246             1122
## 2         Mumbai          5210         6692814          93910             3825
## 3      Bangalore          1015         6257020         118465             1143
## 4 North Carolina           338         3433218          78082                0
## 5         Nevada           641         2929559          93892              226
## 6       Coventry           917         2540842          59397             2122
##   favourited_count dob_day dob_year dob_month gender mobile_favourites_count
## 1           105005      29     1999         4 female                       0
## 2            40487      24     1991        10 female                       0
## 3            87968       4     1997         3   male                       0
## 4            25943      22     1998         8   male                       0
## 5            32589       9     1963        11 female                       0
## 6            19760       1     1995         1 female                       0
##   mobile_favourited_count education experience age  race     wage
## 1                       0         8          0  29 white 16.31000
## 2                 5032191        15          0   0 white 17.91000
## 3                       0         9          0  32 white 15.71000
## 4                       0         9         44  40 white  7.00000
## 5                       0        13         24  45 white 17.87000
## 6                       0        15         21  14 white 14.10839
##   retweeted_count retweet_count height
## 1               1            30    156
## 2               1             6    162
## 3               2            65    168
## 4               0             8    180
## 5               1             7    162
## 6               2            64    158
```

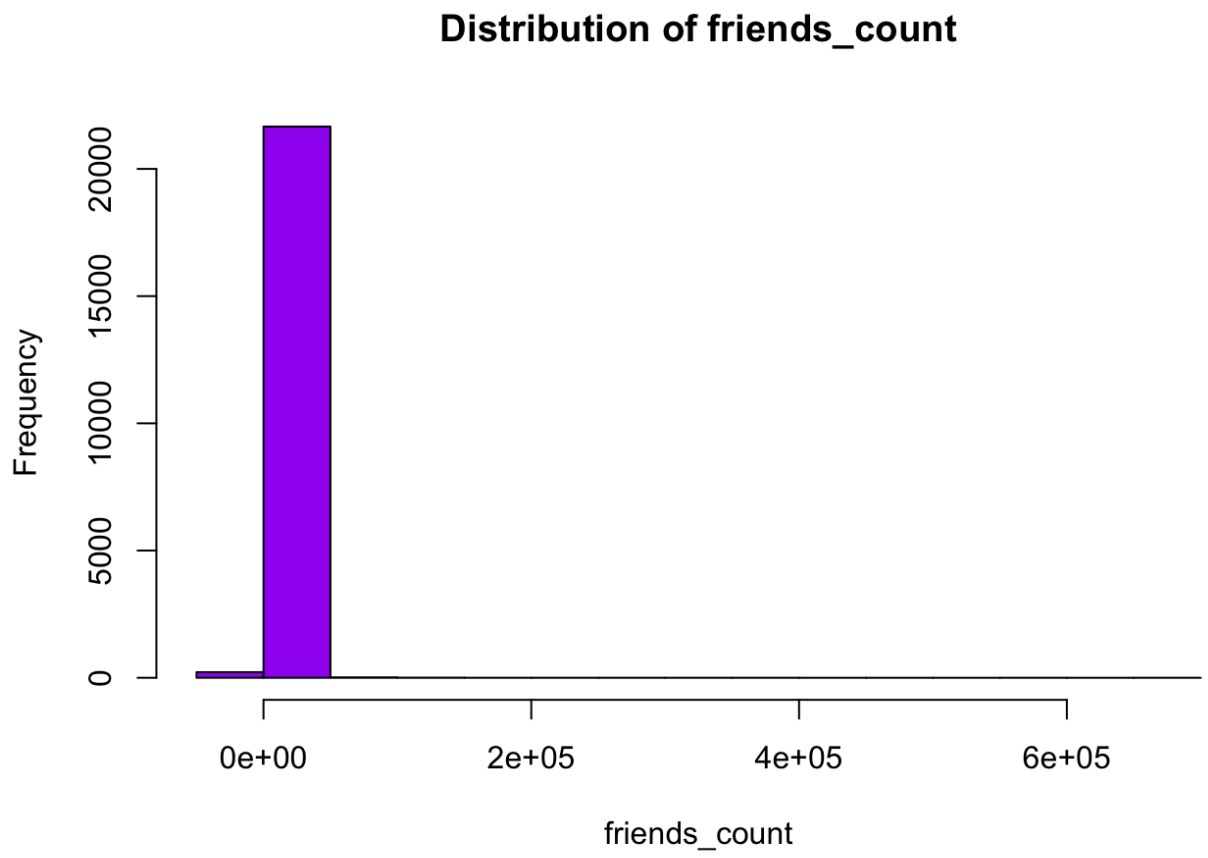
```
summary(df_Twitter)
```

```
##     screen_name    created_at_month created_at_day  created_at_year
##  +5400E1. :    1   Min.   : 1.000   Min.   : 1.00   Min.   :2006
##  000D0se7 :    1   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.:2009
##  001apdov :    1   Median : 6.000   Median :16.00   Median :2011
##  001RBTePh:    1   Mean   : 6.069   Mean   :15.78   Mean   :2011
##  003B0K2  :    1   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:2013
##  007unfasa:    1   Max.   :12.000   Max.   :31.00   Max.   :2015
##  (Other)  :21910
##       country                      location     friends_count
##  USA      :14905   Mexico              :  122   Min.   :   -84
##  Canada   :  943   Boston              :  108   1st Qu.:   123
##  India    :  890   Montreal            :  107   Median :   324
##  Earth    :  516   Nevada              :   80   Mean   :  1058
##  England  :  467   Bangalore           :   79   3rd Qu.:   849
##  Australia:  291   Indianapolis Indiana:   76   Max.   :660549
##  (Other)  : 3904   (Other)             :21344
##  followers_count    statuses_count    favourites_count  favourited_count
##  Min.   :       0   Min.   :      1   Min.   :      0   Min.   :     0.00
##  1st Qu.:     105   1st Qu.:    558   1st Qu.:     16   1st Qu.:     2.00
##  Median :     336   Median :   2341   Median :    164   Median :     9.00
##  Mean   :    5859   Mean   :  12486   Mean   :   2217   Mean   :    92.24
##  3rd Qu.:    1075   3rd Qu.:   9348   3rd Qu.:    950   3rd Qu.:    36.00
##  Max.   :22187643   Max.   :1136198   Max.   :1140139   Max.   :105005.00
##
##     dob_day         dob_year      dob_month           gender
##  Min.   : 1.00   Min.   :1900   Min.   :   1.000   female: 7319
##  1st Qu.: 5.00   1st Qu.:1965   1st Qu.:   3.000   male  :14569
##  Median :13.00   Median :1982   Median :   6.000   NA's  :   28
##  Mean   :13.49   Mean   :1976   Mean   :   6.398
##  3rd Qu.:21.00   3rd Qu.:1990   3rd Qu.:   9.000
##  Max.   :35.00   Max.   :2000   Max.   :1992.000
##
##  mobile_favourites_count mobile_favourited_count   education
##  Min.   :     0.0        Min.   :      0         Min.   : 3.0
##  1st Qu.:     0.0        1st Qu.:      0         1st Qu.:11.0
##  Median :     0.0        Median :      0         Median :13.0
##  Mean   :   152.9        Mean   :    649         Mean   :12.5
##  3rd Qu.:     0.0        3rd Qu.:      0         3rd Qu.:14.0
##  Max.   :377123.0        Max.   :5032191         Max.   :24.0
##
##    experience          age                        race            wage
##  Min.   :-32.00   Min.   :-6.00   white           :18032   Min.   :  5.00
##  1st Qu.:  0.00   1st Qu.:28.00   latino          : 1115   1st Qu.: 13.52
##  Median :  7.00   Median :36.00   asian           :  960   Median : 20.36
##  Mean   : 10.88   Mean   :35.54   persian         :  376   Mean   : 22.97
##  3rd Qu.: 20.00   3rd Qu.:44.00   hispanic        :  353   3rd Qu.: 28.40
##  Max.   : 74.00   Max.   :91.00   pacific islander:  276   Max.   :104.97
##                                   (Other)         :  804
##  retweeted_count    retweet_count         height
##  Min.   :  0.0000   Min.   :   0.00   Min.   :  1.0
##  1st Qu.:  0.0000   1st Qu.:   0.00   1st Qu.:165.0
##  Median :  1.0000   Median :   3.00   Median :172.0
##  Mean   :  0.9715   Mean   :  52.73   Mean   :171.5
##  3rd Qu.:  1.0000   3rd Qu.:  19.00   3rd Qu.:178.0
```

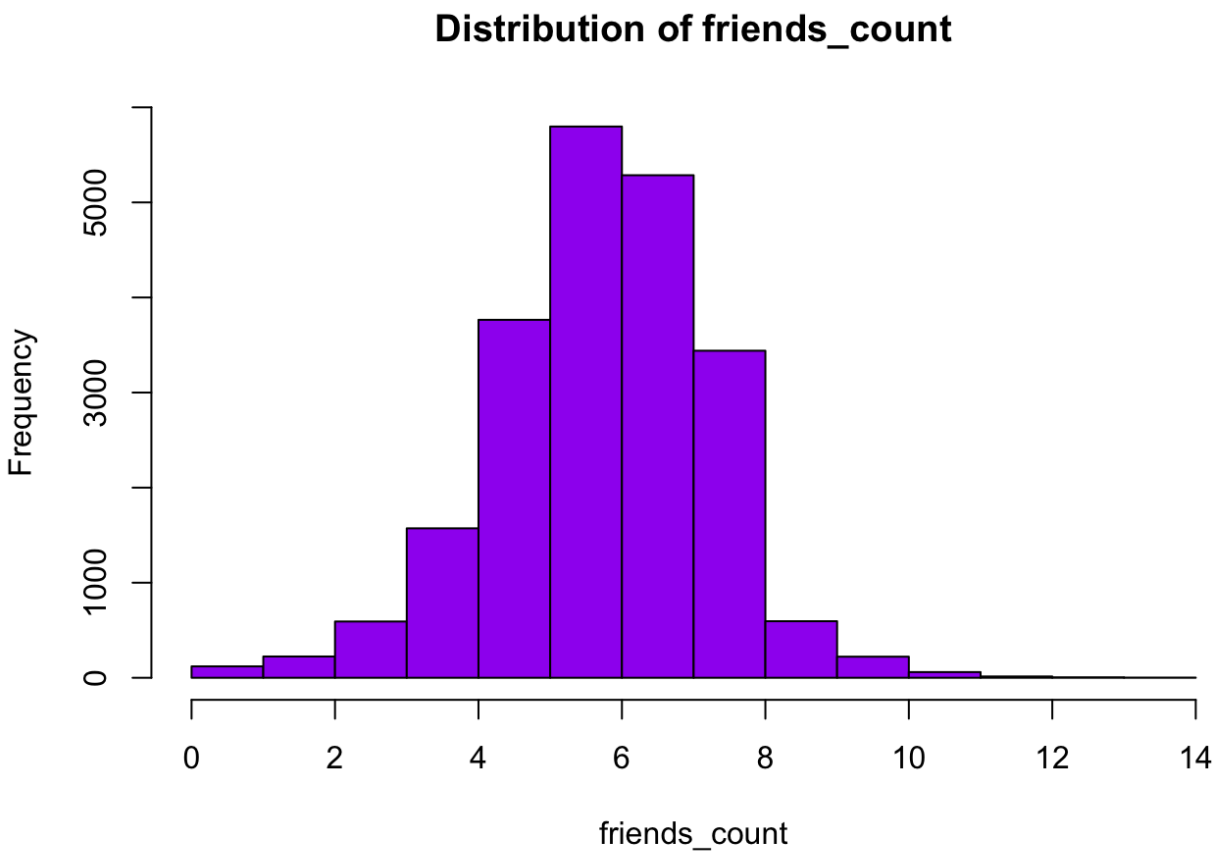
```
##  Max.   :705.0000   Max.   :5506.00   Max.   :203.0
##
```

a.

```
#histogram of friends_count
hist(df_Twitter$friends_count, main = "Distribution of friends_count", xlab = "friends_cou
nt", col = "purple")
```




```
#histogram of log(friends_count)
hist(log(df_Twitter$friends_count), main = "Distribution of friends_count", xlab = "friend
s_count", col = "purple")
```



Interpretation:

We have plotted the histogram to analyse the data for the variable friends_count. The data is right skewed, that is, the data has positive skewed distribution. The right skewed data has all different mean, median and mode. Thus, for this variable, the mode is the highest point in the plot whereas the mean and median falls to the right of this peak (or mode). As in a skewed distribution, mean is always closer to the tail, so in this right skewed distribution mean must be toward the right of the median, or we can say that, mean must be higher than the median. Also, the data seems to have a lot of outliers. We have also plotted the log of friends_count to know more about the data and it seems that the data is approximately normally distributed.

b.

```
#summary statistics by using individual functions of r
minimum_count <- min(df_Twitter$friends_count)
cat("Minimum friends count:", minimum_count, "\n")
```

```
## Minimum friends count: -84
```

```
Q1_count <- quantile(df_Twitter$friends_count, 0.25)
cat("First quantile of friends count:", Q1_count, "\n")
```

```
## First quantile of friends count: 123
```

```
median_count <- quantile(df_Twitter$friends_count, 0.50)
cat("Median friends count:", median_count, "\n")
```

```
## Median friends count: 324
```

```
mean_count <- mean(df_Twitter$friends_count)
cat("Mean friends count:", mean_count, "\n")
```

```
## Mean friends count: 1057.911
```

```
Q3_count <- quantile(df_Twitter$friends_count, 0.75)
cat("Third quantile of friends count:", Q3_count, "\n")
```

```
## Third quantile of friends count: 849
```

```
maximum_count <- max(df_Twitter$friends_count)
cat("Maximum friends count:", maximum_count, "\n")
```

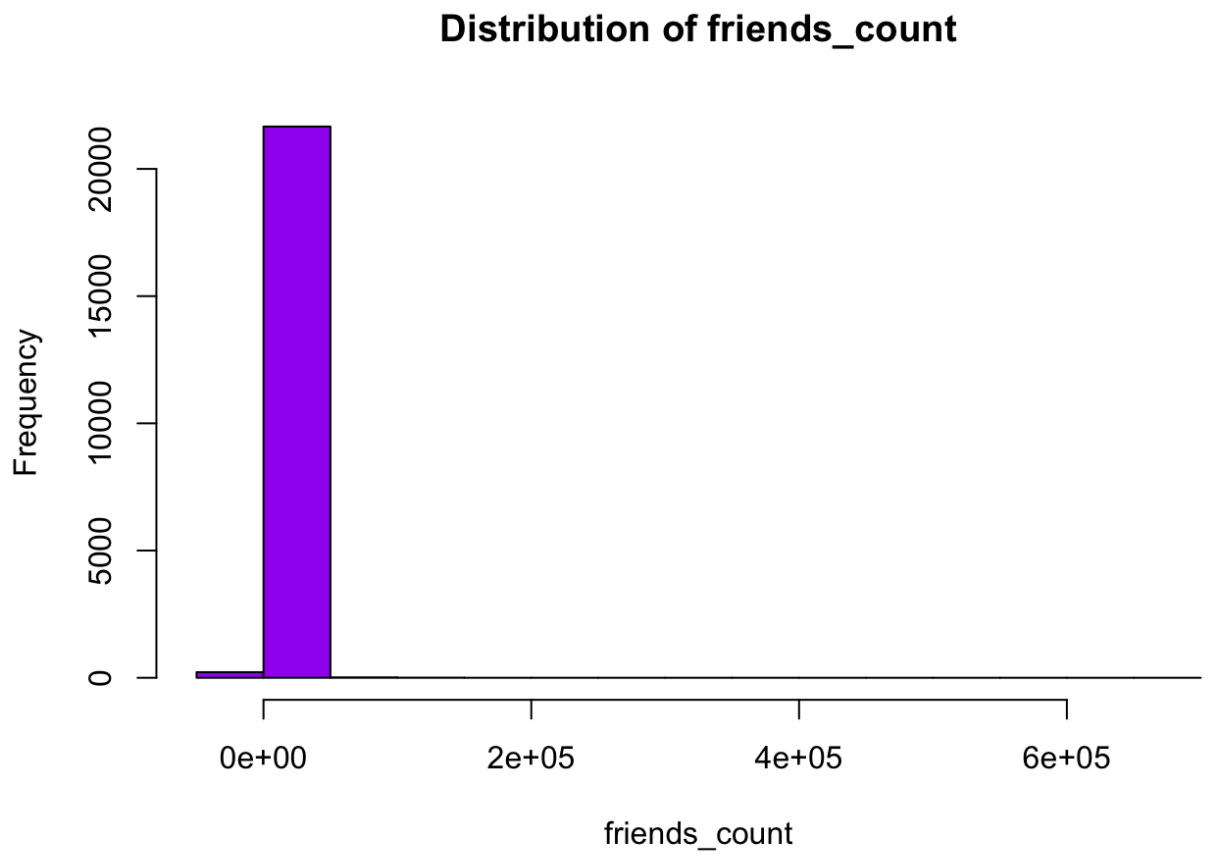
```
## Maximum friends count: 660549
```

```
#summary statistics by using summary function of r
summary(df_Twitter$friends_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     -84     123     324    1058     849  660549
```

C.

```
#checking data quality of friends_count variable

#histogram of friends_count
hist(df_Twitter$friends_count, main = "Distribution of friends_count", xlab = "friends_cou
nt", col = "purple")
```




```
#data quality statistics
sum(is.na(df_Twitter$friends_count))
```

```
## [1] 0
```

```
unique_friends <- unique(df_Twitter$friends_count)
length(unique_friends)
```

```
## [1] 3162
```

```
length(which(df_Twitter$friends_count == 0))
```

```
## [1] 220
```

```
class(df_Twitter$friends_count)
```

```
## [1] "integer"
```

```
length(df_Twitter$friends_count)
```

```
## [1] 21916
```

We have already calculated the summary statistics of friends_count variable in part (b). So, we now have the following information regarding the variable, friends_count:

1. Minimum friends count: -84
2. First quantile of friends count: 123
3. Median friends count: 324
4. Mean friends count: 1057.911
5. Third quantile of friends count: 849
6. Maximum friends count: 660549
7. Number of missing values: 0
8. Number of unique values: 3162
9. Number of zero values: 220
10. Number of values: 21916
11. Class: integer

```
#code to generate two temporary files with data quality of the entire dataset using dataQu
alityR package in r
num.file <- paste(tempdir(), "/dq_num.csv", sep = "") #creating temporary file with numeri
c variables
cat.file <- paste(tempdir(), "/dq_cat.csv", sep = "") #creating temporary file with catego
rical variables
checkDataQuality(df_Twitter, out.file.num = num.file, out.file.cat = cat.file)
```

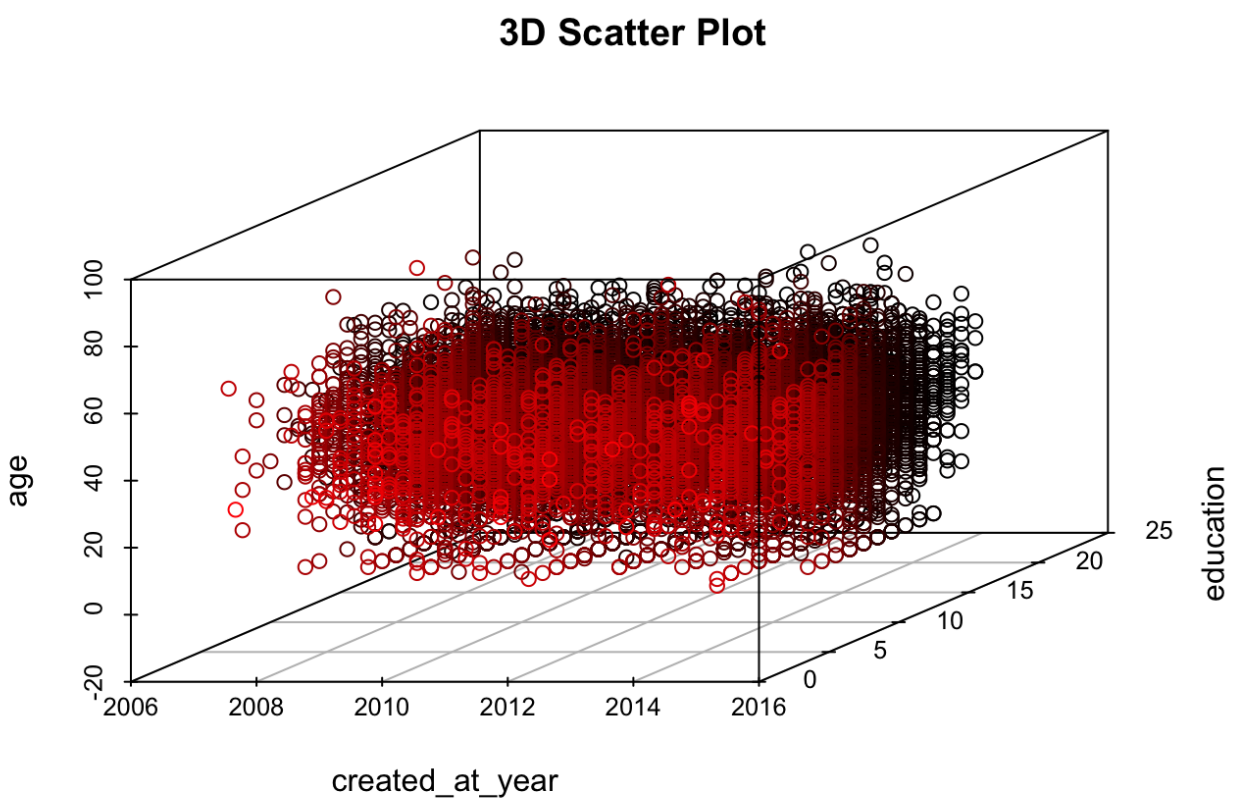
```
## Check for numeric variables completed // Results saved to disk // Time difference of 0.
2098949 secs
## Check for categorical variables completed // Results saved to disk // Time difference o
f 0.1364231 secs
```

Interpretation:

The data for friends_count variable has no missing value with a total of 21916 values and 3162 unique values. Also, there are 220 values with zeroes. It is interesting to note that friends_count has negative value as well. So, the minimum is a negative value, -84 and maximum is 660549. Also, the data is right skewed. It seems that the data skewed to the right due to lower boundary in the variable data due to which the mean is lying to the right of median with values 1057.911 and 324, respectively.

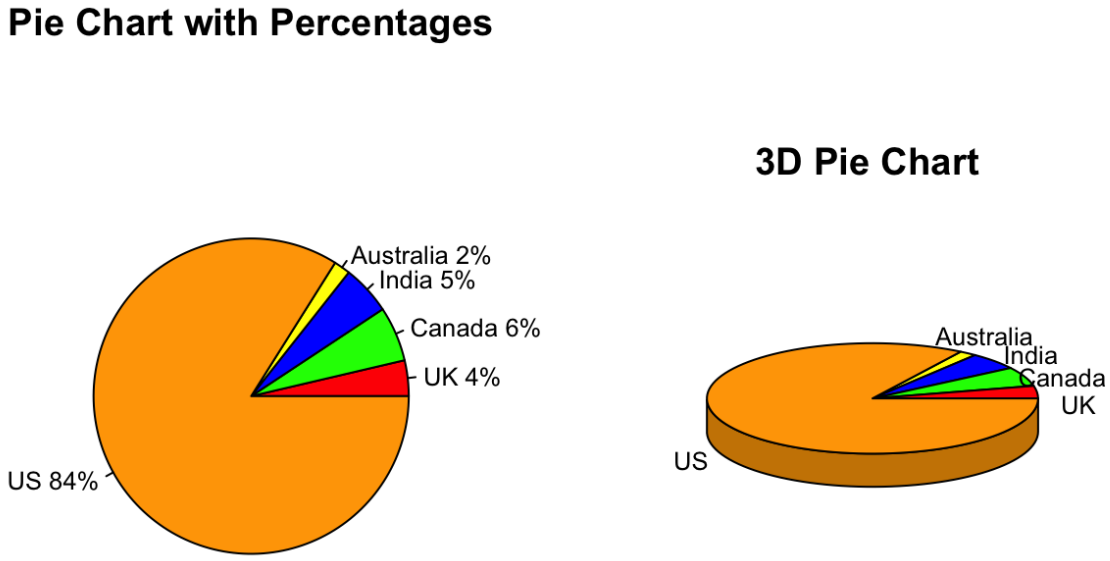
d.

```
#3D scatterplot with highlight
scatterplot3d(df_Twitter$created_at_year, df_Twitter$education, df_Twitter$age, highlight.
3d = TRUE, main = "3D Scatter Plot", xlab = "created_at_year", ylab = "education", zlab =
"age")
```



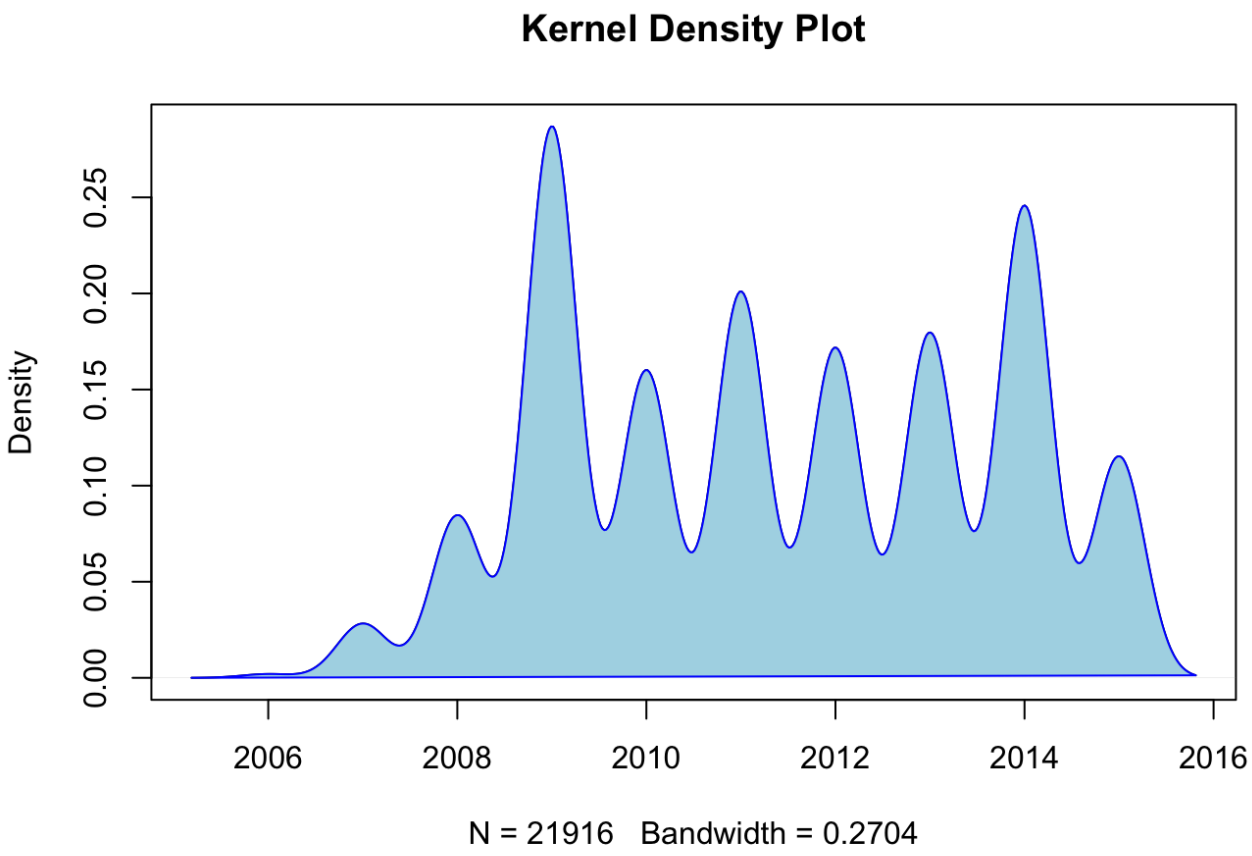
e.

```
#arranging pie charts in 1 row and 2 columns
par(mfrow = c(1, 2))

#percentage pie chart
slices <- c(650, 1000, 900, 300, 14900)
labels_1 <- c("UK", "Canada", "India", "Australia", "US")
pct <- round(slices/sum(slices)*100)
labels_2 <- paste(labels_1, " ", pct, "%", sep = "")
pie(slices, labels = labels_2, col = c("red", "green", "blue", "yellow", "orange"), main =
"Pie Chart with Percentages", cex = 0.8)

#3D pie chart
pie3D(slices, labels = labels_1, col = c("red", "green", "blue", "yellow", "orange"), labe
lcex = 0.8, main = "3D Pie Chart ")
```



f.

```
#finding the density of created_at_year variable
d <- density(df_Twitter$created_at_year)
plot(d, main = "Kernel Density Plot")
polygon(d, col = "light blue", border = "blue")
```



Interpretation:

The kernel density plot produces smooth curve estimating the probability density function of a continuous variable with peaks of the density plot displaying where values are concenterated over the interval. Here, the continuous variable is time period and the continuous smooth curve depicts the probability density function of that variable. We can see that the plot depicts a comb distribution with alternate high and low peaks. This could have occured due to rounding off or some mistake. We can also see that the highest peak is for year 2009, which means that the maximum screen names have been created in the year 2009.

Problem 3: (Insurance Claims)

```
#loading the csv
df_insurance <- read.csv('raw_data.csv')
head(df_insurance)
```

```
##           A          B C  D
## 1  8.257164 -0.6560755 6  8
## 2 10.557378 -0.7158294 7  8
## 3  8.744211  0.7996106 7  5
## 4  6.555028  1.5832173 6 10
## 5  9.362121  1.0272024 7  8
## 6  9.020671  0.7197130 7 12
```

```
summary(df_insurance)
```

```
##        A                B                   C             D
##  Min.   : 3.902   Min.   :-3.17616   Min.   :2.0   Min.   : 2.000
##  1st Qu.: 7.793   1st Qu.:-0.63195   1st Qu.:5.0   1st Qu.: 7.000
##  Median : 9.072   Median : 0.03412   Median :6.0   Median : 9.000
##  Mean   : 9.079   Mean   : 0.03063   Mean   :6.3   Mean   : 8.919
##  3rd Qu.:10.395   3rd Qu.: 0.67029   3rd Qu.:7.0   3rd Qu.:11.000
##  Max.   :14.794   Max.   : 2.96851   Max.   :9.0   Max.   :18.000
```

a.

```
#normalizing the data using the scale function and creating another dataframe
Ndata <- as.data.frame(scale(df_insurance))
head(Ndata)
```

```
##             A          B          C          D
## 1 -0.46047167 -0.6870000 -0.2019694 -0.2931233
## 2  0.82780052 -0.7467798  0.4705888 -0.2931233
## 3 -0.18769316  0.7693173  0.4705888 -1.2500845
## 4 -1.41378095  1.5532638 -0.2019694  0.3448509
## 5  0.15837732  0.9970078  0.4705888 -0.2931233
## 6 -0.03285735  0.6893851  0.4705888  0.9828251
```

```
summary(Ndata)
```

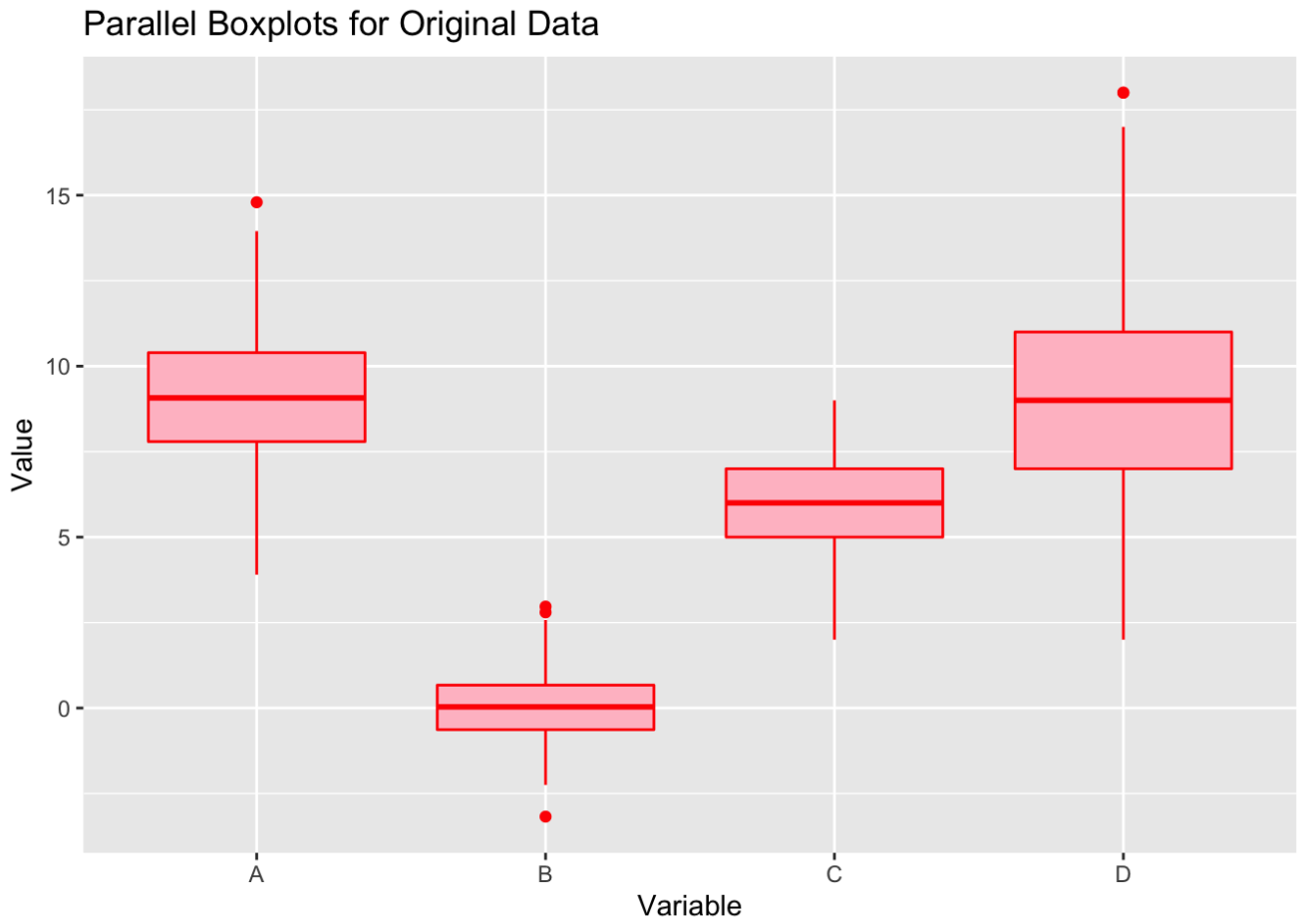
```
##        A                   B                   C                 D
##  Min.   :-2.899878   Min.   :-3.208180   Min.   :-2.8922   Min.   :-2.20705
##  1st Qu.:-0.720321   1st Qu.:-0.662867   1st Qu.:-0.8745   1st Qu.:-0.61211
##  Median :-0.003837   Median : 0.003492   Median :-0.2020   Median : 0.02586
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.736648   3rd Qu.: 0.639936   3rd Qu.: 0.4706   3rd Qu.: 0.66384
##  Max.   : 3.200864   Max.   : 2.939157   Max.   : 1.8157   Max.   : 2.89675
```

b.

```
#using the melt function to take data in wide format and stack as a set of columns into a
 single column of data
meltData_1 <- melt(df_insurance)
```

```
## Using  as id variables
```

```
#head(meltData_1)

ggplot(meltData_1, aes(variable, value)) +
  geom_boxplot(color = "red", fill = "pink") +
  labs(x = "Variable", y = "Value", title = "Parallel Boxplots for Original Data")
```

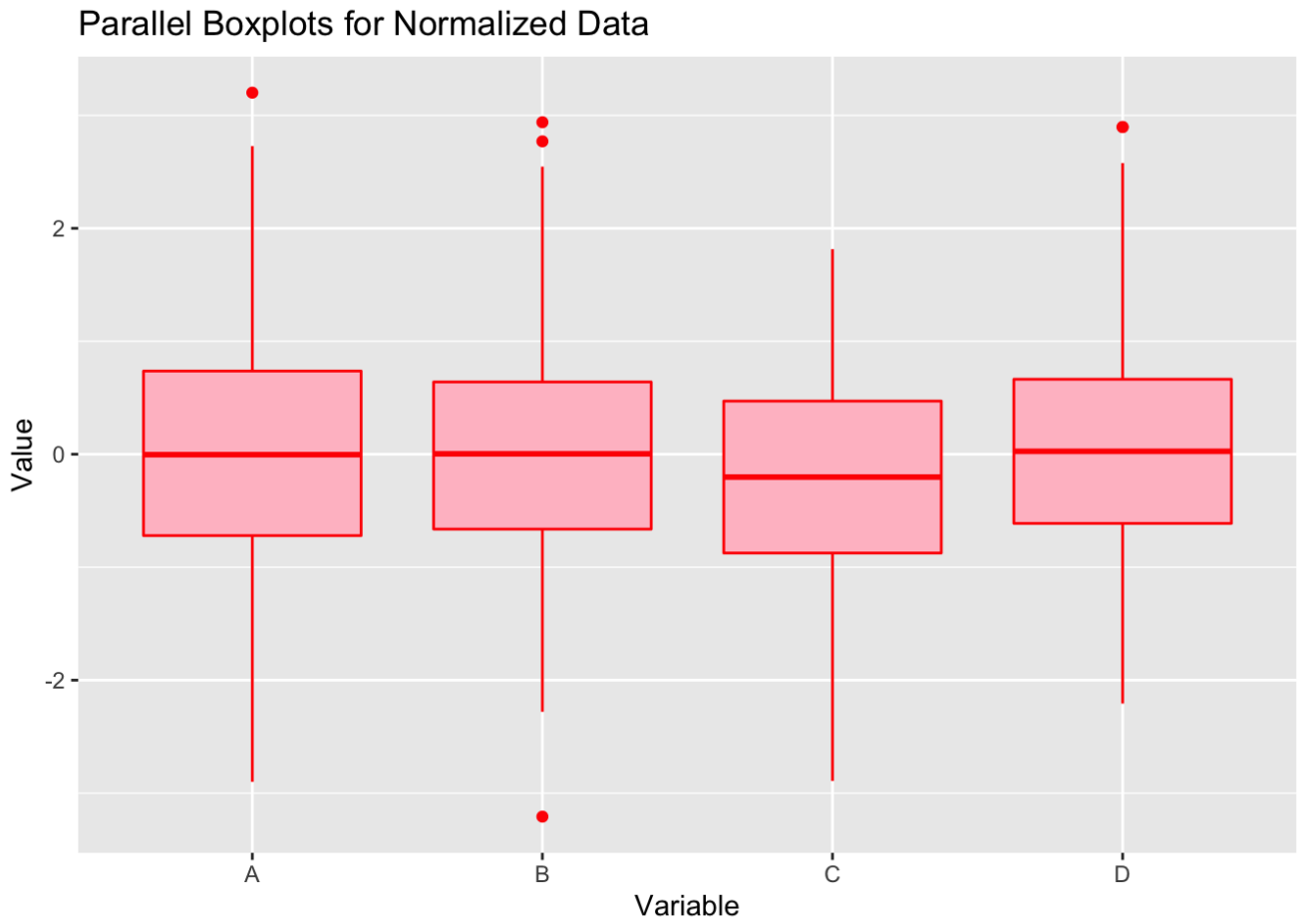


c.

```
#using the melt function to take data in wide format and stack as a set of columns into a
 single column of data
meltData_2 <- melt(Ndata)
```

```
## Using  as id variables
```

```
#head(meltData_2)

ggplot(meltData_2, aes(variable, value)) +
  geom_boxplot(color = "red", fill = "pink") +
  labs(x = "Variable", y = "Value", title = "Parallel Boxplots for Normalized Data")
```



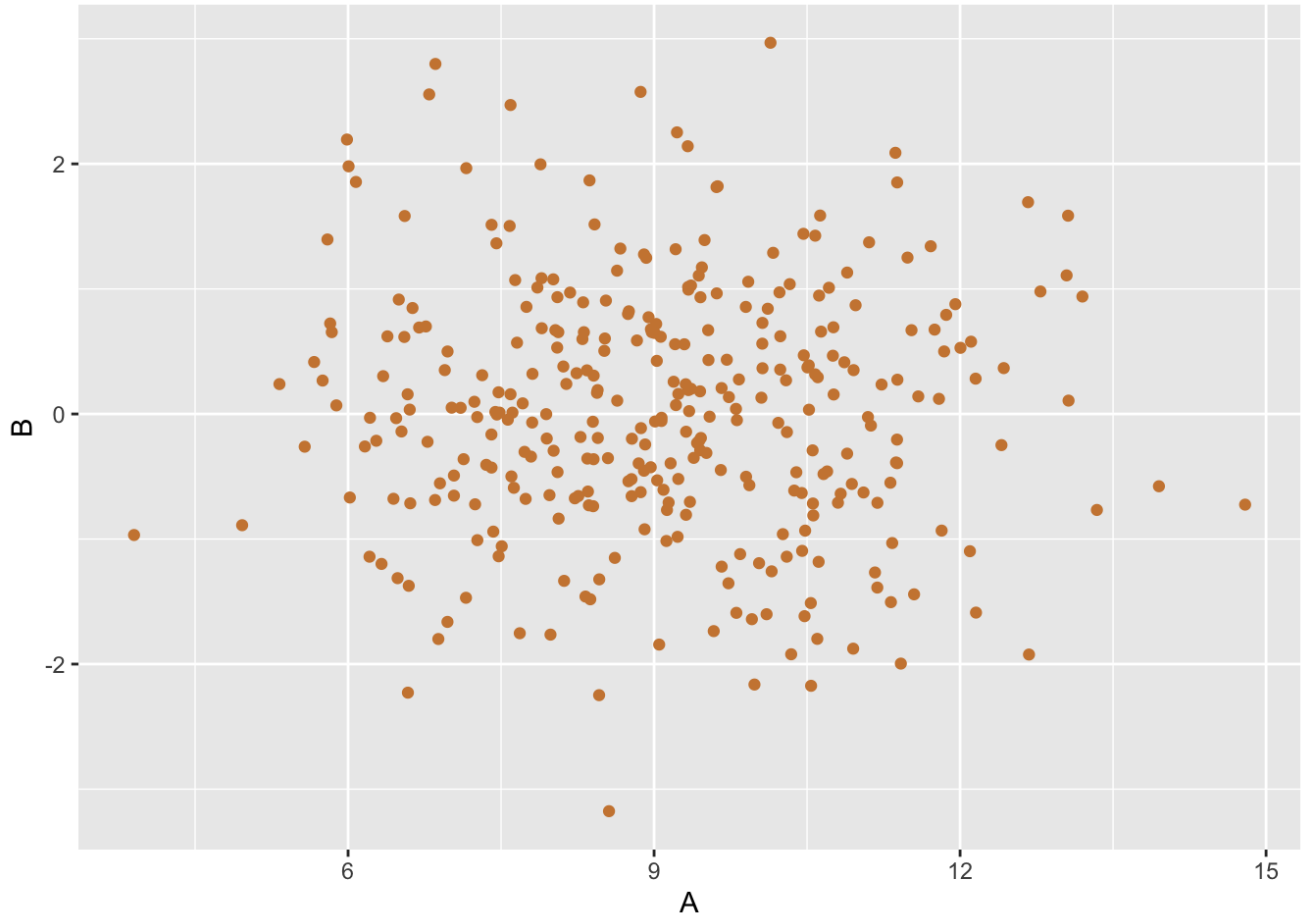
d.

From the boxplot of original data, we can notice that there is a good separation of variables based on values except between variables A and D. The distribution of all the variables look almost symmetrical. Also, variables A, B and D has a few outliers while C does not.

We normalized the data to change the values of columns with numeric data in the dataset to a common scale. This is done without affecting the range of values in the columns.
From the normalized boxplot, we can see that there is not much separation in variables A, B and D based on value. We can also see that variables A and B look more symmetrical than variables C and D. Also, variables A, B and D has some outliers whereas variable C has no outlier.

e.

```
#scatter plot for A and B
ggplot(df_insurance) +
  geom_point(mapping = aes(x = A, y = B), colour = "tan3") +
  labs(x = "A", y = "B")
```



Interpretation:

It can be interpreted from the scatter plot that A and B have no correlation, which means that A and B have no relation or dependence to each other.
Let us check the correlation between the two variables using a function of R.

```
cor(df_insurance$A, df_insurance$B)
```

```
## [1] -0.03059086
```

The correlation coefficient has value -0.03059086, which depicts that A and B are independent of each other. Thus, it would be correct to say that Sustainability and Carbon foorptint are not related to each other.