Homework #1

Group 4

5/10/2020

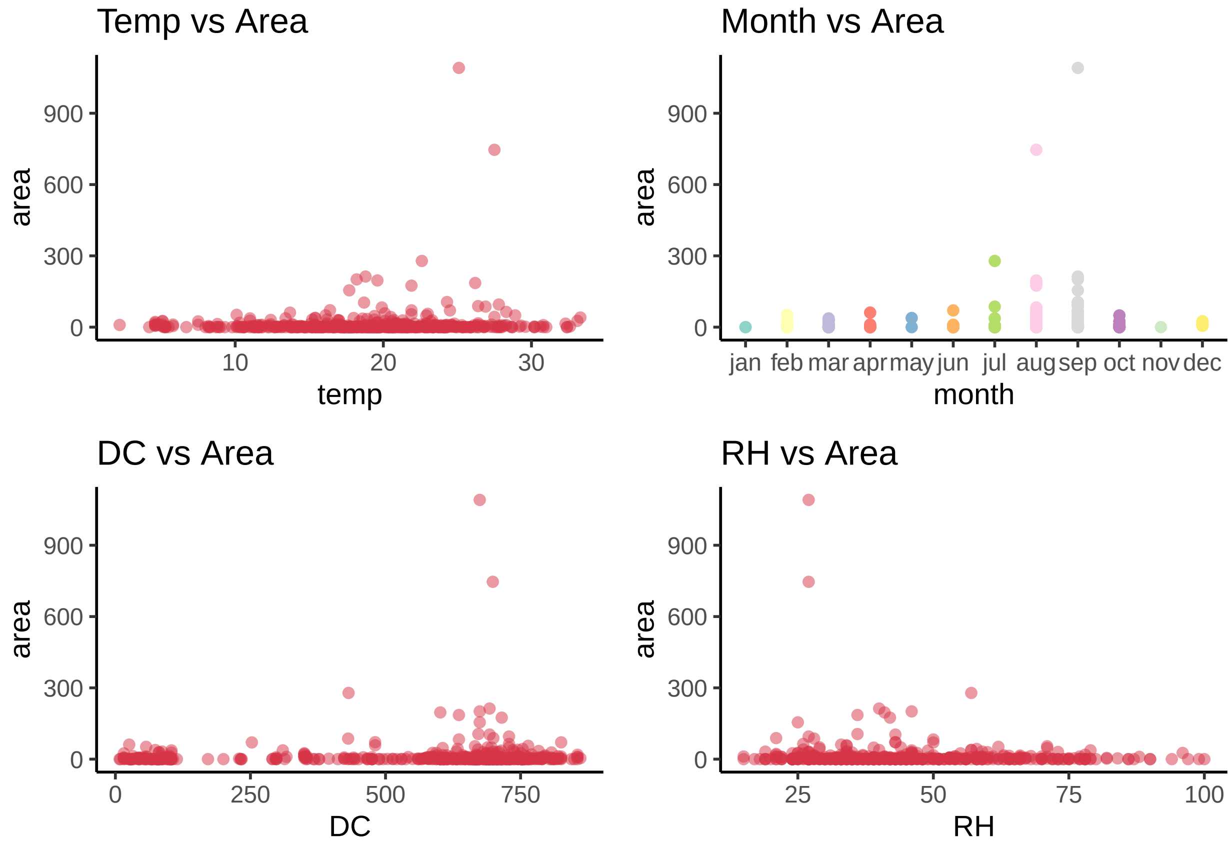
First, let's import the required libraries

```r
library(ggplot2)
library(GGally)
library(cowplot)
library(dplyr)
library(reshape2)
library(fitdistrplus)
library(scatterplot3d)
library(plotrix)
library(RColorBrewer)
```

Problem 1

(a) Plot area vs.temp, area vs. month, area vs. DC, area vs. RH for January through December combined in one graph. Hint: Place area on Y axis and use 2x2 matrix to place the plots adjacent to each other.

```r
# Import the Dataframe
forestfires <- data.frame(read.csv("./data/forestfires.csv"),
  stringsAsFactors = FALSE
)

# Convert the month column into factors and sort from Jan-Dec
forestfires$month <- factor(forestfires$month,
  levels = c(
    "jan", "feb", "mar",
    "apr", "may", "jun",
    "jul", "aug", "sep",
    "oct", "nov", "dec"
  )
)

# Create 4 scatter plots
p1 <- ggplot(forestfires, aes(temp, area)) +
  geom_point(color = "#d63447", alpha = 0.5) +
  ggtitle("Temp vs Area") +
  theme_classic()
p2 <- ggplot(forestfires, aes(month, area, color = month)) +
  geom_point() +
  scale_color_brewer(palette = "Set3") +
  theme_classic() +
  theme(legend.position = "none") +
  ggtitle("Month vs Area")
p3 <- ggplot(forestfires, aes(DC, area)) +
  geom_point(color = "#d63447", alpha = 0.5) +
  ggtitle("DC vs Area") +
  theme_classic()
p4 <- ggplot(forestfires, aes(RH, area)) +
  geom_point(color = "#d63447", alpha = 0.5) +
  ggtitle("RH vs Area") +
  theme_classic()

# Arrange plots P1-P4 into a 2x2 grid
plot_grid(p1, p2, p3, p4)
```



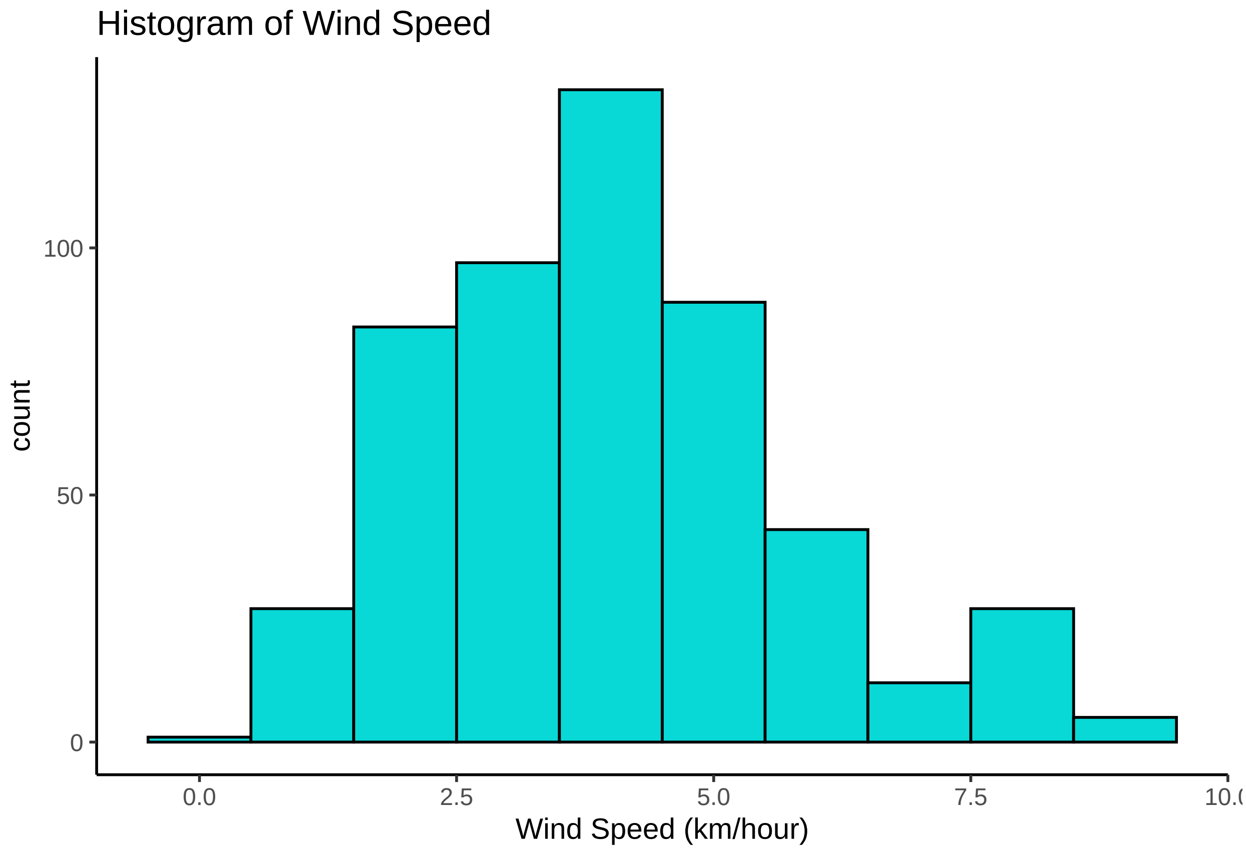
```r
rm(list = c("p1", "p2", "p3", "p4", "fig"))
```

```
## Warning in rm(list = c("p1", "p2", "p3", "p4", "fig")): object 'fig' not found
```

(b) Plot the histogram of wind speed (km/h).

```r
# Create the Wind-Historgram
wind_hist <- ggplot(forestfires, aes(wind)) +
  geom_histogram(bins = 10, fill = "#08d9d6", color = "black") +
  theme_classic() +
  ggtitle("Histogram of Wind Speed") +
  labs(x = "Wind Speed (km/hour)")

# Plot
```

```r
plot(wind_hist)
```



```r
rm(wind_hist)
```

(c) Compute the summery statistics (min, 1Q, mean, median, 3Q, max,) of part b.

```r
# Calculate the Quantiles
quantiles <- quantile(forestfires$wind)

# Print
cat("Minimum Wind Speed is :", quantiles[[1]], "\n")
```

```
## Minimum Wind Speed is : 0.4
```

```r
cat("1st Quantile of Wind Speed is :", quantiles[[2]], "\n")
```

```
## 1st Quantile of Wind Speed is : 2.7
```

```r
cat("Mean Wind Speed is :", mean(forestfires$wind), "\n")
```

```
## Mean Wind Speed is : 4.017602
```

```r
cat("Median Wind Speed is :", quantiles[[3]], "\n")
```

```
## Median Wind Speed is : 4
```

```r
cat("3rd Quartile of Wind Speed is :", quantiles[[4]], "\n")
```

```
## 3rd Quartile of Wind Speed is : 4.9
```

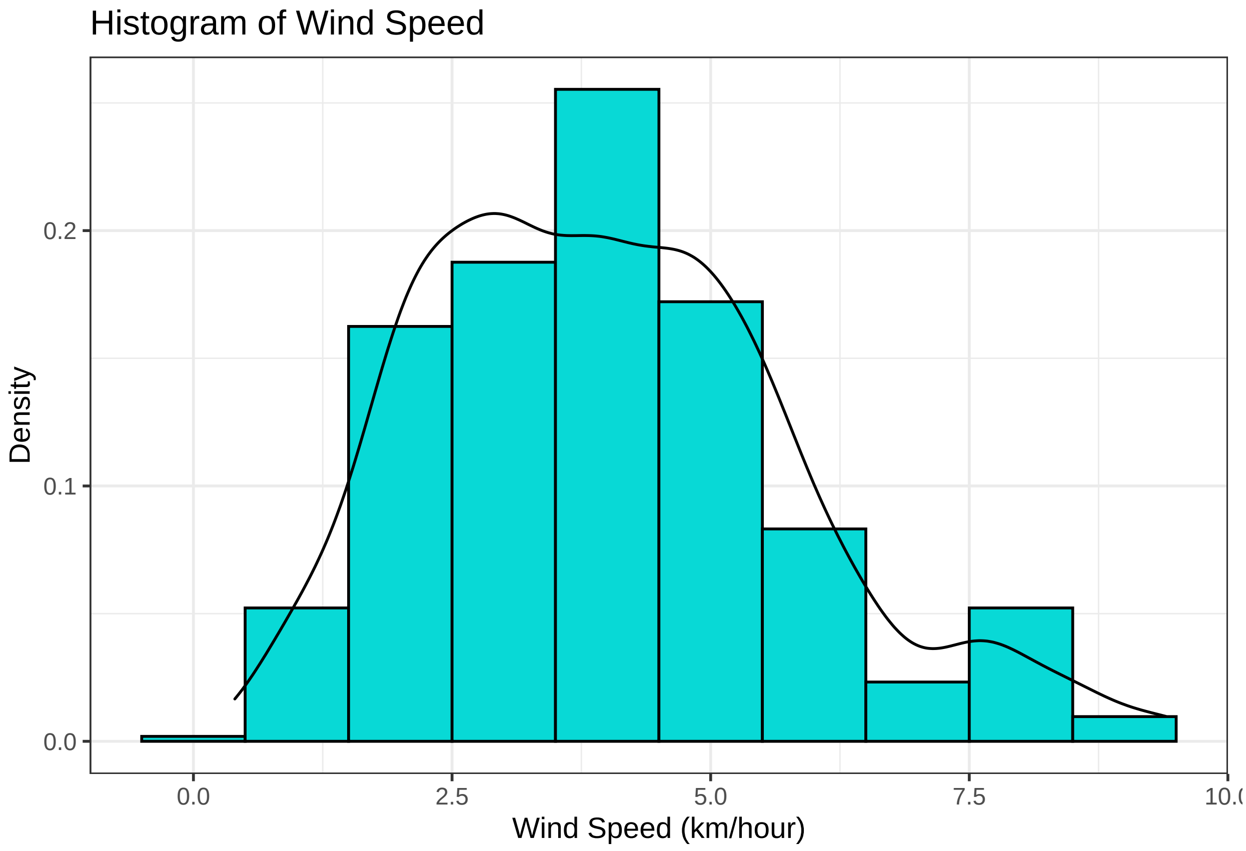
```r
cat("Maximum Wind Speed is :", quantiles[[5]], "\n")
```

```
## Maximum Wind Speed is : 9.4
```

```r
rm(quantiles)
```

(d) Add a density line to the histogram in part b.

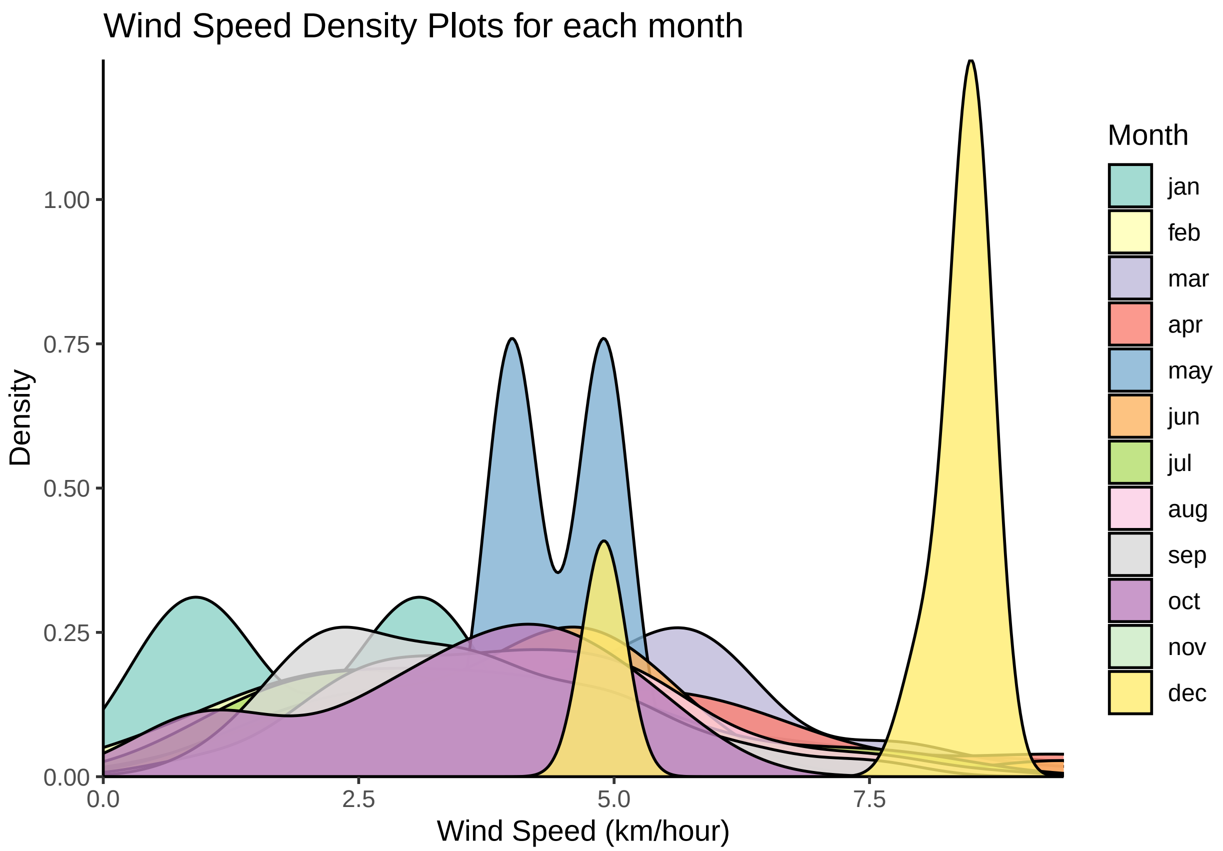
```r
# Create the Histogram and Density Plot
wind_hist_plus_density <- ggplot(forestfires, aes(x = wind, y = ..density..)) +
  geom_histogram(bins = 10, colour = "black", fill = "#08d9d6") +
  geom_density(aes(y = ..density..), color = "black") +
  ylab("Density") +
  xlab("Wind Speed (km/hour)") +
  ggtitle("Histogram of Wind Speed") +
  theme_bw()

plot(wind_hist_plus_density)
```



```r
rm(wind_hist_plus_density)
```

(e) Plot the wind speed density function of all months in one plot. Use different colors for different months in the graph to interpret your result clearly. [Hint: use ggplot + geom_density or qplot(geom=density)]

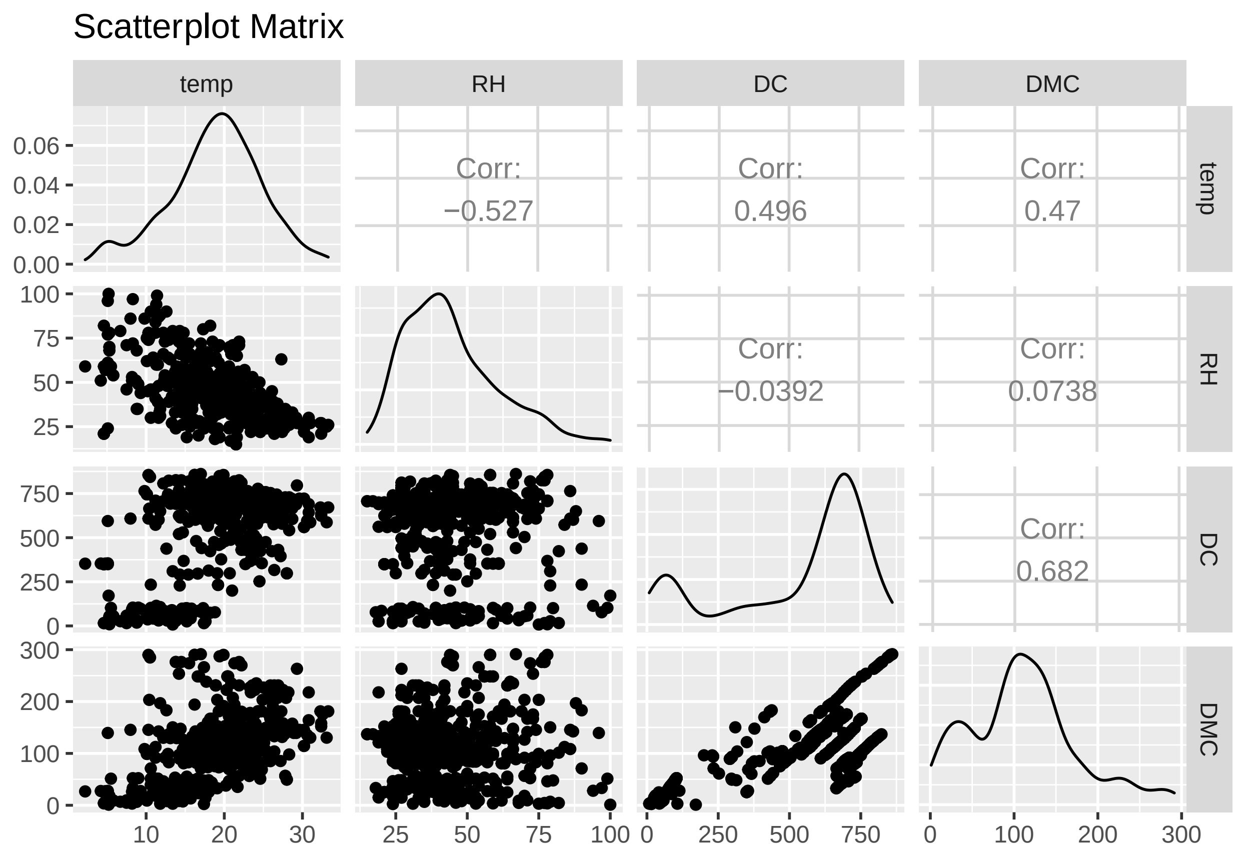
```r
# Create month wise density plot
month_density_plot <- ggplot(forestfires, aes(
  x = wind,
  y = ..density..,
  fill = month
)) +
  geom_density(alpha = 0.8) +
  scale_fill_brewer(palette = "Set3") +
  theme_classic() +
  ggtitle("Wind Speed Density Plots for each month") +
  labs(x = "Wind Speed (km/hour)", y = "Density", fill = "Month") +
  scale_x_continuous(expand = c(0, 0), limits = c(0, NA)) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, NA))

plot(month_density_plot)
```



```
rm(month_density_plot)
```

(f) Plot the scatter matrix for temp, RH, DC and DMC. How would you interpret the result in terms of correlation among these data?

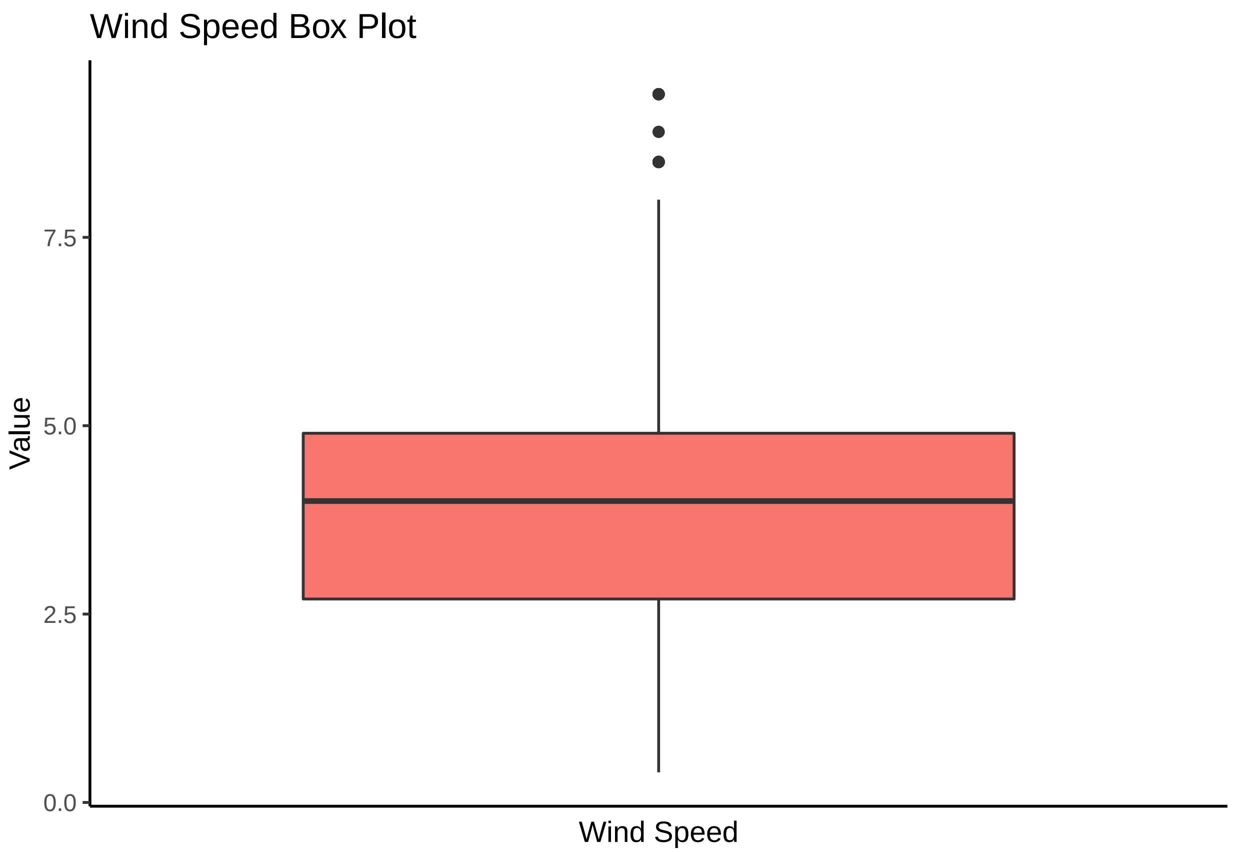
```
# Plot ScatterMatrix
ggpairs(forestfires,
  title = "Scatterplot Matrix",
  columns = c("temp", "RH", "DC", "DMC")
)
```



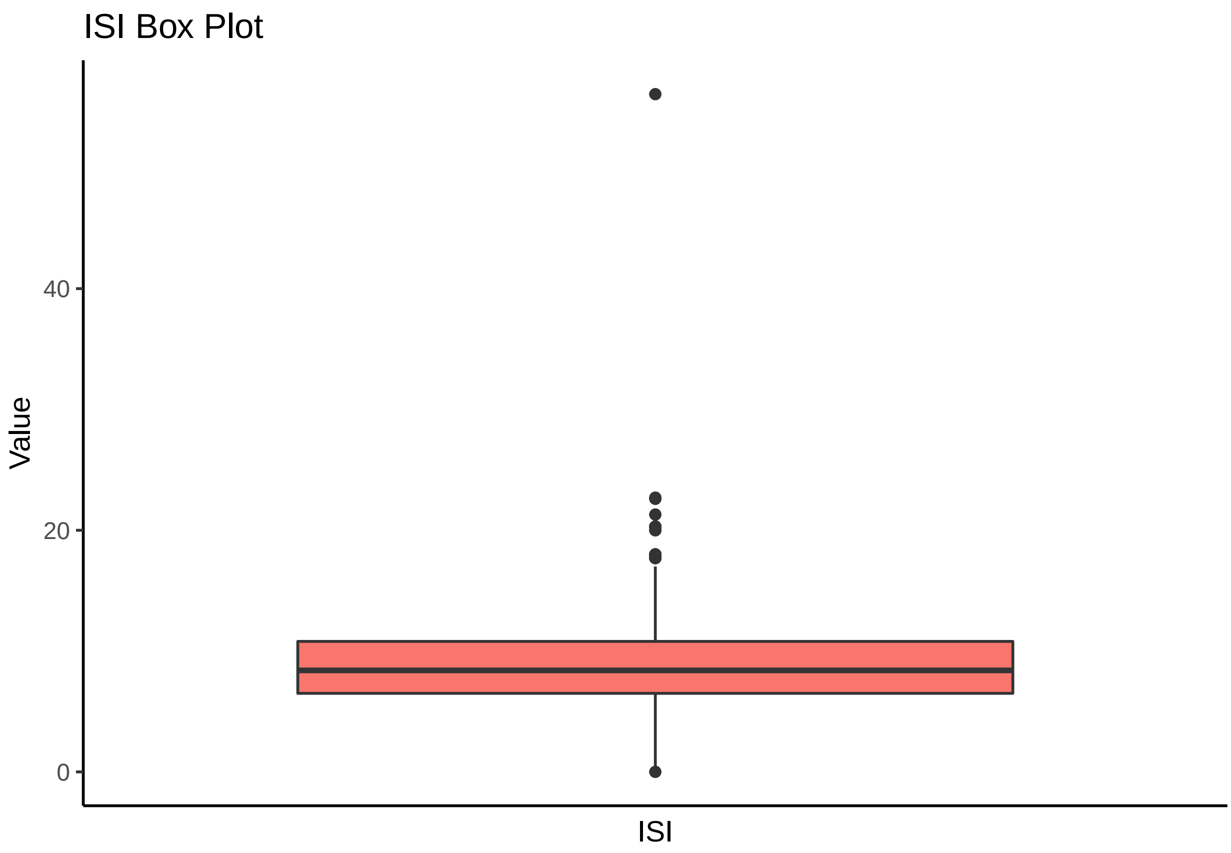
Looking at the Scatter Matrix, we can make the following conclusions - (1) Temp and RH are negatively correlated with corr. coeff of -0.527. (2) Temp and DC are positively correlated with corr. coeff of 0.496. (3) Temp and DC are positively correlated with corr. coeff of 0.47. (4) RH has no correlation with DC and DMS whatsoever. (5) DC and DMC are very strongly correlated with a corr. coeff of 0.682.

(g) Create boxplot for wind, ISI and DC. Are there any anomalies/outliers? Interpret your result.

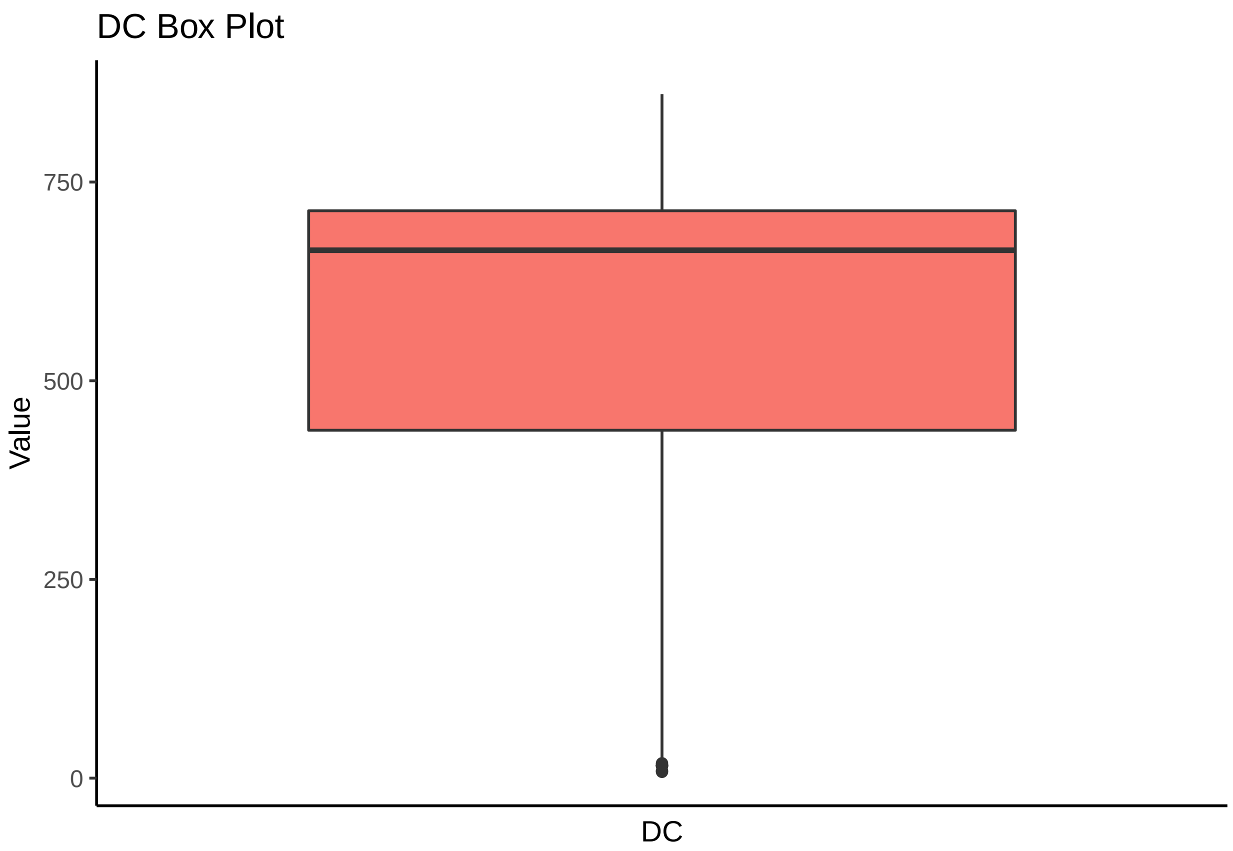
```
# Create a temporary dataframe and reshape it
suppressMessages({
  df <- melt(forestfires)
})

# Plot 3 boxplots for Wind, ISI, and DC
ggplot(df %>% filter(variable == "wind"), aes(x = variable, y = value, fill = variable)) +
  geom_boxplot() +
  theme_classic() +
  theme(
    legend.position = "none",
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank()
  ) +
  ggtitle("Wind Speed Box Plot") +
  labs(x = "Wind Speed", y = "Value")
```



```r
ggplot(df %>% filter(variable == "ISI"), aes(x = variable, y = value, fill = variable)) +
  geom_boxplot() +
  theme_classic() +
  theme(
    legend.position = "none",
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank()
  ) +
  ggtitle("ISI Box Plot") +
  labs(x = "ISI", y = "Value")
```



```
ggplot(df %>% filter(variable == "DC"), aes(x = variable, y = value, fill = variable)) +
  geom_boxplot() +
  theme_classic() +
  theme(
    legend.position = "none",
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank()
  ) +
  ggtitle("DC Box Plot") +
  labs(x = "DC", y = "Value")
```

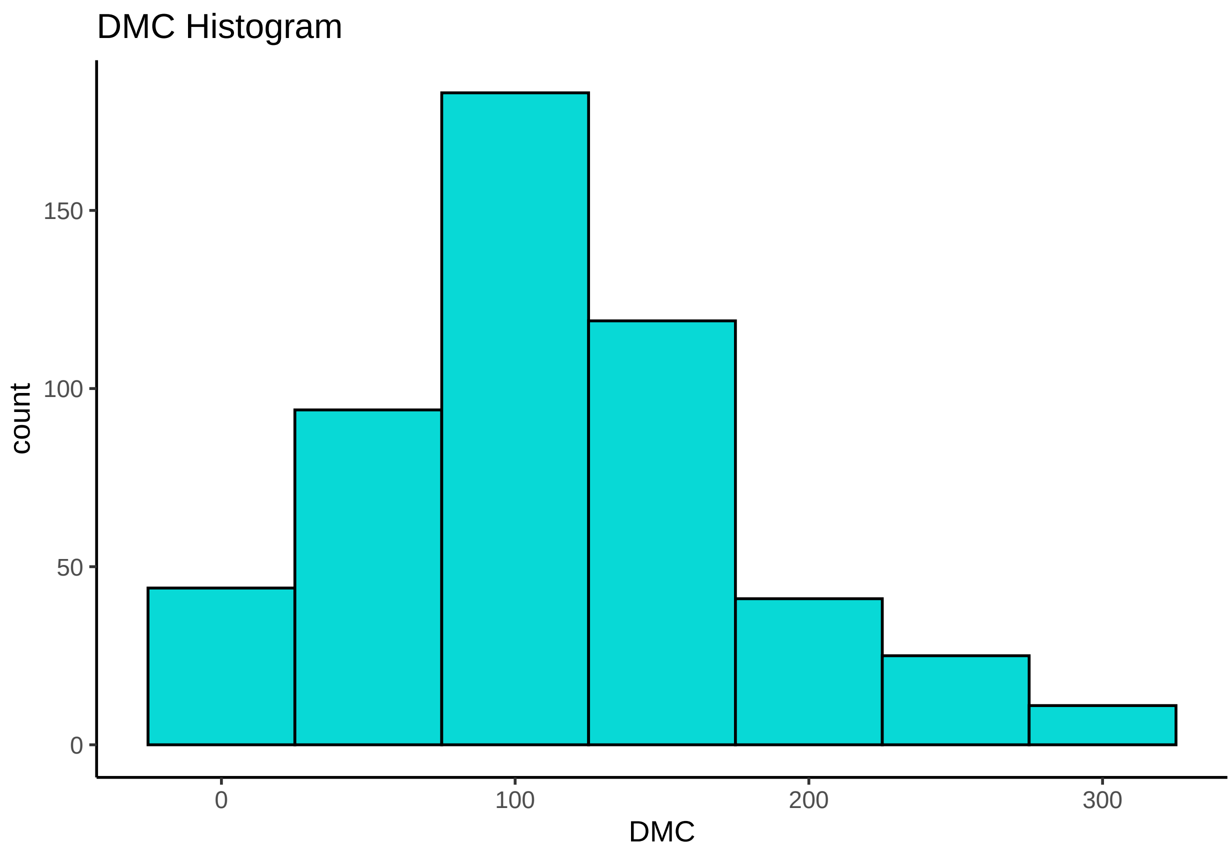


```r
rm(df)
```

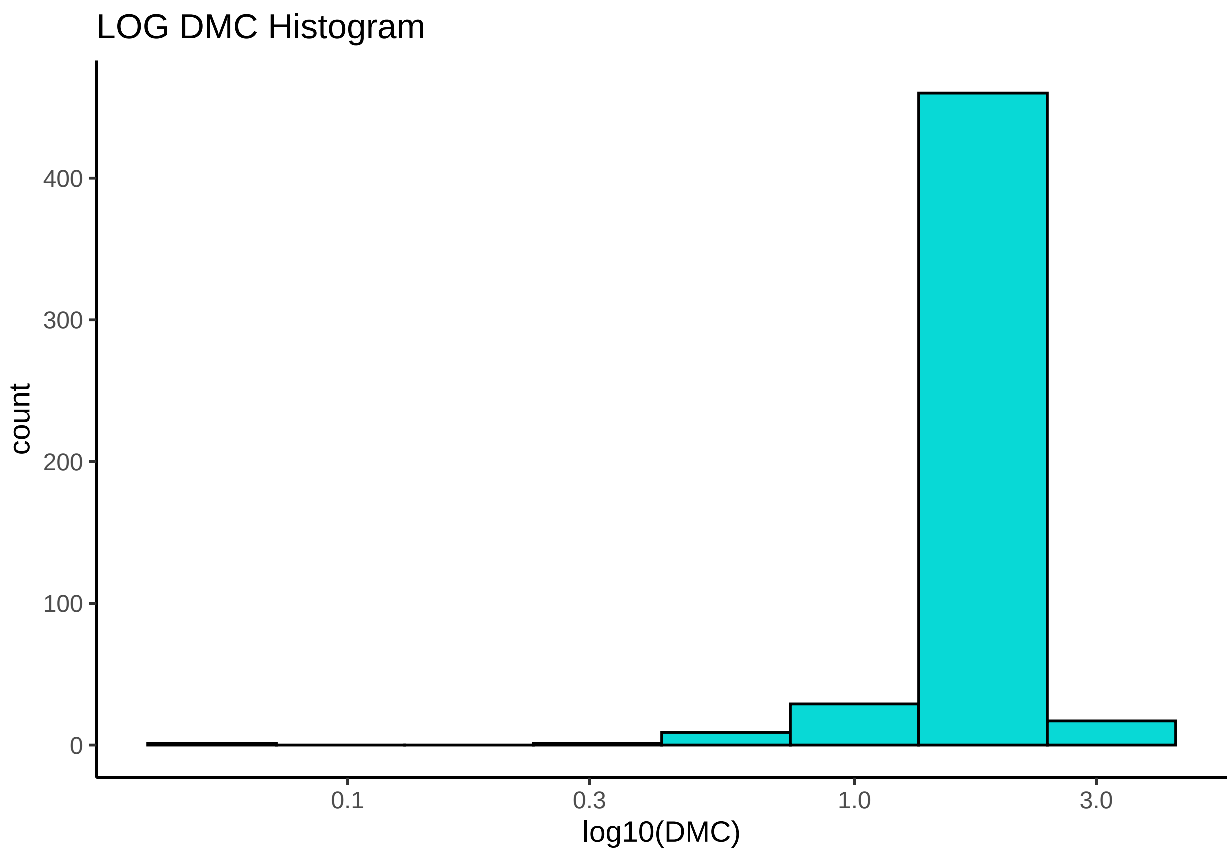
We can make the following observations, (1) Wind has only 3 outliers, the rest of the values are between 0.4 - 7.5 approximately. (2) ISI has quite a few outliers exceeding the value 18. The biggest outlier is of a value 56.1 (3) DC has only 2 outliers which are below the value 15.

(h) Create the histogram of DMC. Create the histogram of log of DMC. Compare the result and explain your answer.

```r
# Plot normal histogram
ggplot(forestfires, aes(DMC)) +
  geom_histogram(binwidth = 50, color = "black", fill = "#08d9d6") +
  theme_classic() +
  ggtitle("DMC Histogram")
```



```
# Plot log scale histogram
ggplot(forestfires, aes(log10(DMC))) +
  geom_histogram(bins = 8, color = "black", fill = "#08d9d6") +
  theme_classic() +
  ggtitle("LOG DMC Histogram") +
  scale_x_log10()
```



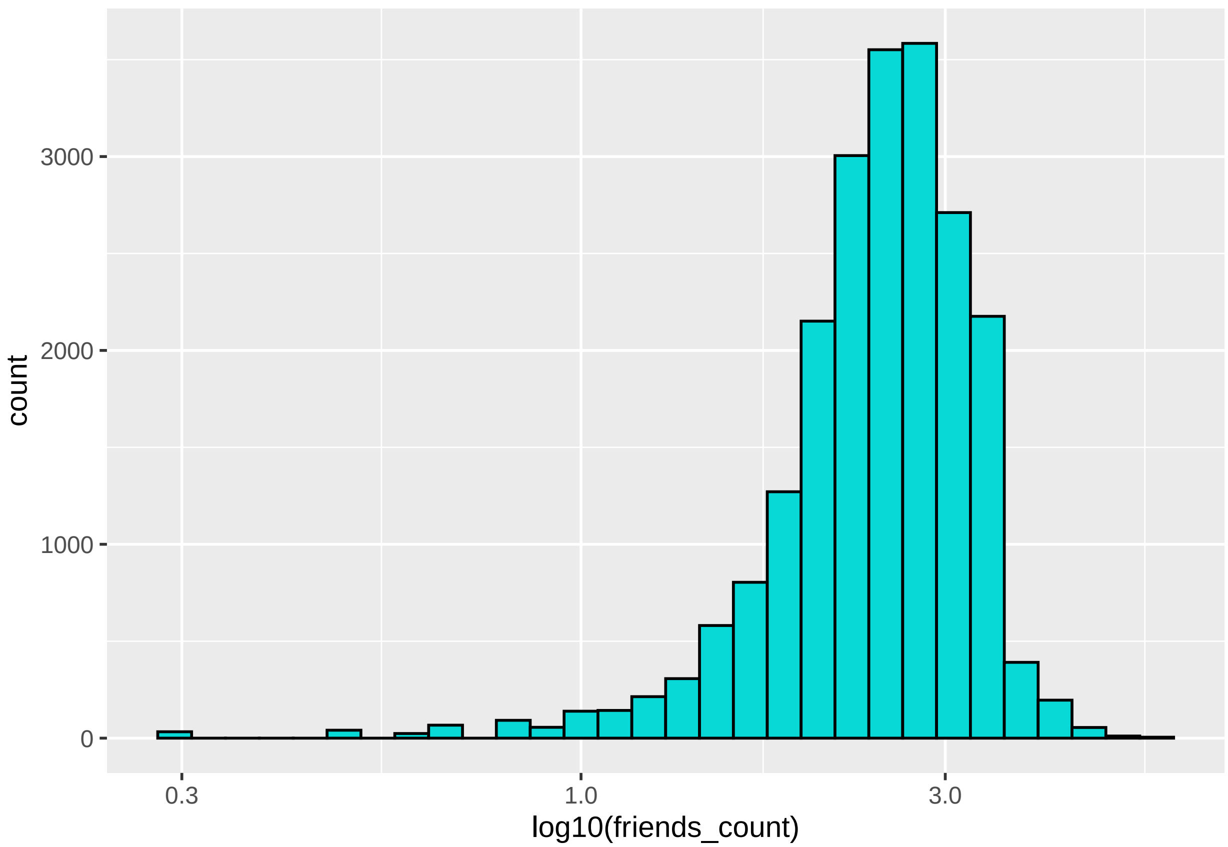
We can make the following observations - (1) The normal histogram of DMC tells us that it follows a normal distribution with most values of the range 100-150. It is slightly left skewed.

(2) The Histogram of log(DMC) supports the observations made in (1). Since most of the DMC values are between 100-300, it makes sense that the log(DMC) histogram shows the majority of values at 2 (since log10(X>100) lies in range 1-2).

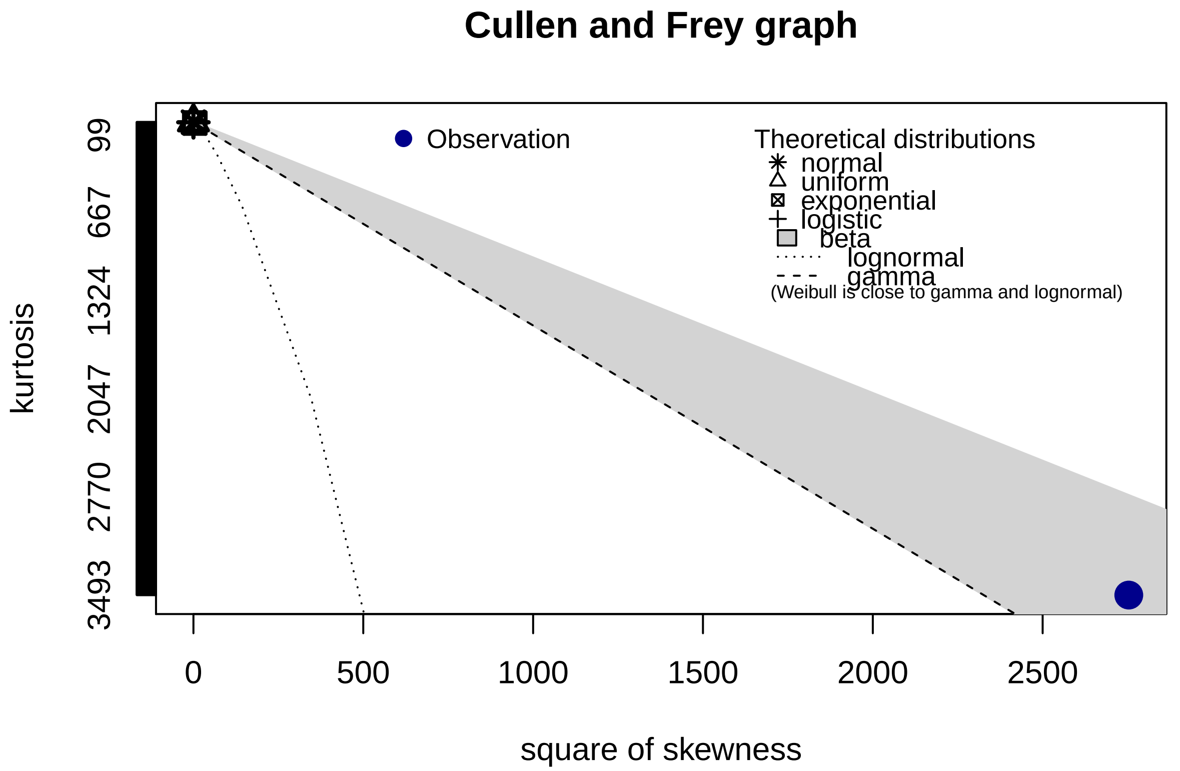
Problem 2

(a) How are the data distributed for friend_count variable?

```r
# Import the Twitter Data csv
M01_quasi_twitter <- data.frame(read.csv("./data/M01_quasi_twitter.csv"))

# Filter out negative values
friends_count <- filter(M01_quasi_twitter, friends_count > 0)

# Since the data is skewed, we can't plot the normal histogram.
# So we plot the log scale histogram
ggplot(friends_count, aes(log10(friends_count))) +
  geom_histogram(color = "black", fill = "#08d9d6") +
  scale_x_log10()
```

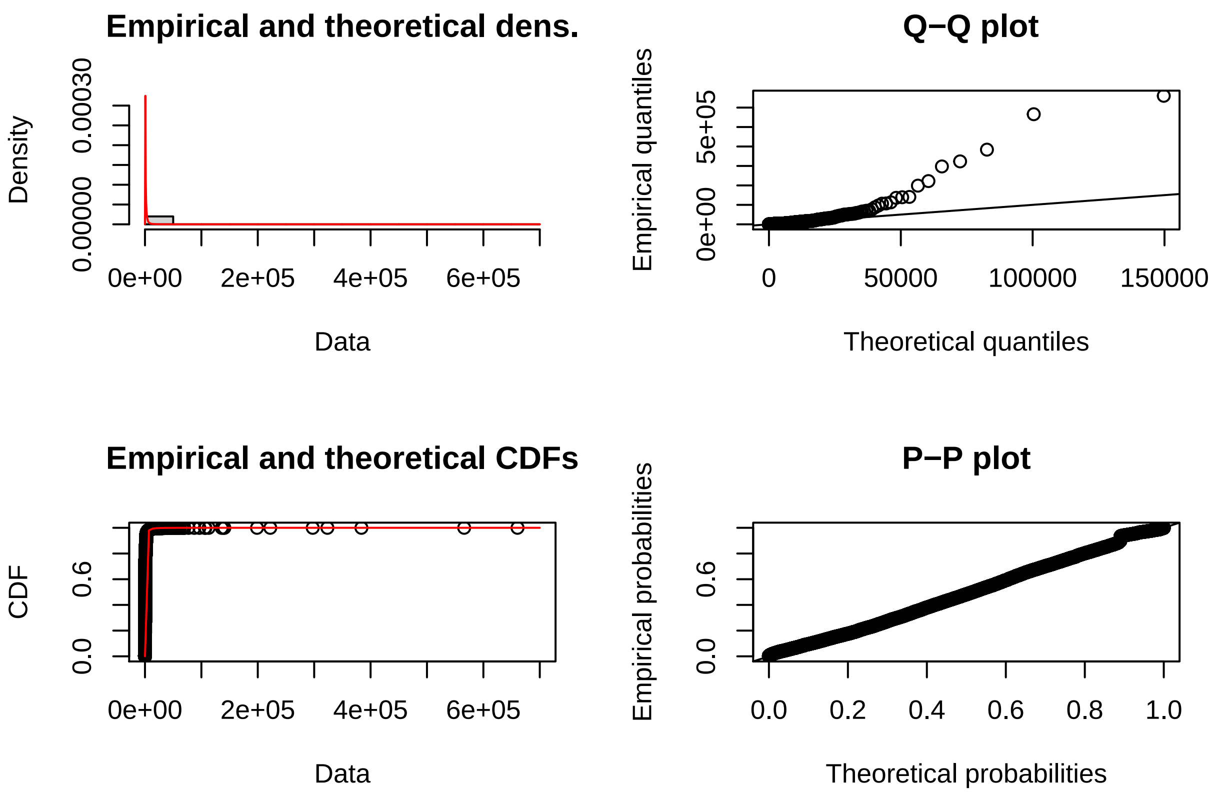


```
# Use the descdist function from the fitdistrplus package to fit a distribution
descdist(friends_count$friends_count)
```



```
## summary statistics
## ------
## min:  1   max:  660549
## median:  330
## mean:  1068.691
## estimated sd:  8165.629
## estimated skewness:  52.47439
## estimated kurtosis:  3492.611
```

```
# Let's try and verify these observations
fit_lnorm <- fitdist(friends_count$friends_count, "lnorm")

# Plot goodness-of-fit plots
plot(fit_lnorm)
```



```
rm(friends_count, fit_lnorm)
```

From the log histogram as well as the goodness-of-fit-plots, we can say that the friend_count variable follows a lognormal distribution.

(b) Compute the summery statistics (min, 1Q, mean, median, 3Q, max) on friend_count.

```
# Calculate the Quantiles
quantiles <- quantile(M01_quasi_twitter$friends_count)

# Print
cat("Minimum friends_count is :", quantiles[[1]], "\n")
```

```
## Minimum friends_count is : -84
```

```
cat("1st Quantile of friends_count is :", quantiles[[2]], "\n")
```

```
## 1st Quantile of friends_count is : 123
```

```
cat("Mean friends_count is :", mean(M01_quasi_twitter$friends_count), "\n")
```

```
## Mean friends_count is : 1057.911
```

```
cat("Median friends_count is :", quantiles[[3]], "\n")
```

```
## Median friends_count is : 324
```

```
cat("3rd Quartile of friends_count is :", quantiles[[4]], "\n")
```

```
## 3rd Quartile of friends_count is : 849
```

```r
cat("Maximum friends_count is :", quantiles[[5]], "\n")
```

```
## Maximum friends_count is : 660549
```

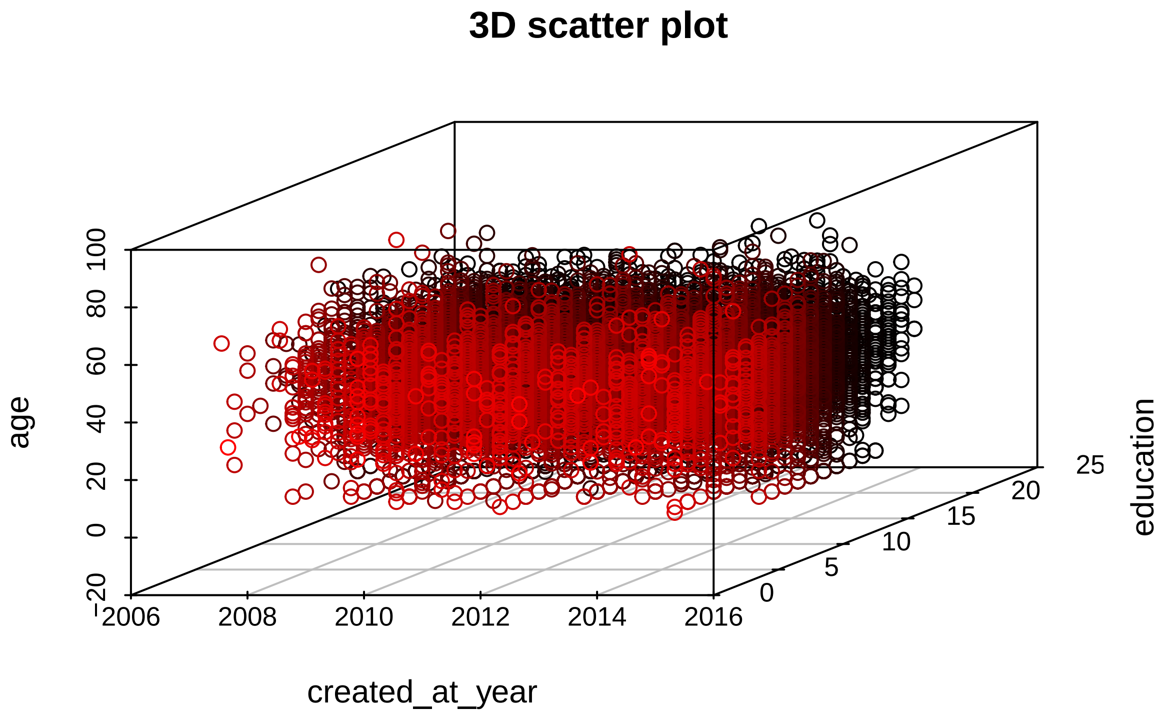
```r
rm(quantiles)
```

(c) How is the data quality in friend_count variable? Interpret your answer.

The friends_count variable is a highly skewed variable. It also has incorrect data as one user has -84 friends which is not possible. Although most of the users have friends_count in the hundreds, a few users with friends_count in hundreds of thousands skew the distribution significantly. Overall, the quality of the friend_count variable is not that good.

(d) Produce a 3D scatter plot with highlighting to impression the depth for variables below on M01_quasi_twitter.csv dataset. created_at_year, education, age. Put the name of the scatter plot "3D scatter plot".

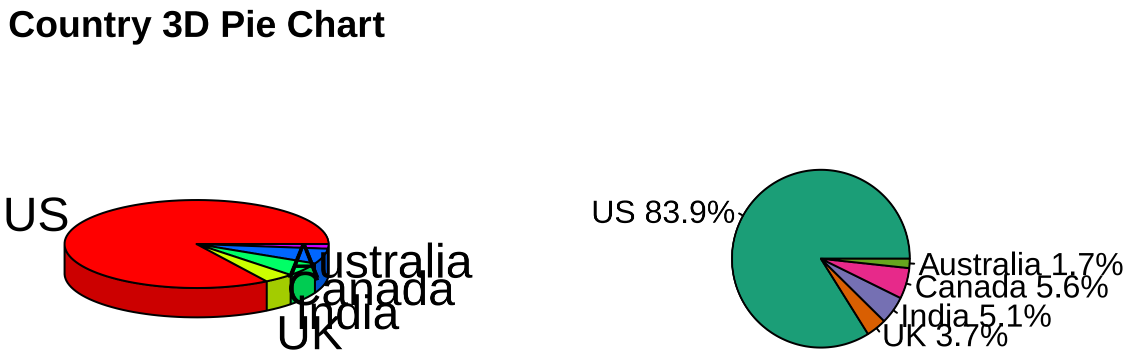
```r
# Create and Plot the 3D Scatter Plot
scatterplot3d(
  x = M01_quasi_twitter$created_at_year,
  y = M01_quasi_twitter$education,
  z = M01_quasi_twitter$age,
  main = "3D scatter plot",
  xlab = "created_at_year",
  ylab = "education",
  zlab = "age",
  highlight.3d = TRUE
)
```



(e) Consider 650, 1000, 900, 300 and 14900 tweeter accounts are in UK, Canada, India, Australia and US, respectively. Plot the percentage Pie chart includes percentage amount and country name adjacent to it, and also plot 3D pie chart for those countries along with the percentage pie chart. Hint: Use C=(1, 2) matrix form to plot the charts together.

```
# Create a temporary dataframe for the data
df <- data.frame(
  "country" = c("UK", "Canada", "India", "Australia", "US"),
  "num_accounts" = c(650, 1000, 900, 300, 14900)
)

# Mutate the dataframe to create a percentages columns
df <- df %>%
  group_by(country) %>%
  arrange(desc(country)) %>%
  mutate(prop = 100 * round(num_accounts / sum(df$num_accounts), 3)) %>%
  mutate(percentage_labels = paste0(prop, "%"))

# Define the grid, 1 row x 2 columns
par(mfcol = c(1, 2), mar = c(5, 5, 5, 5))

# 3D Pie chart
pie3D(df$prop,
  radius = 0.9,
  labels = df$country,
  main = "Country 3D Pie Chart"
)
```

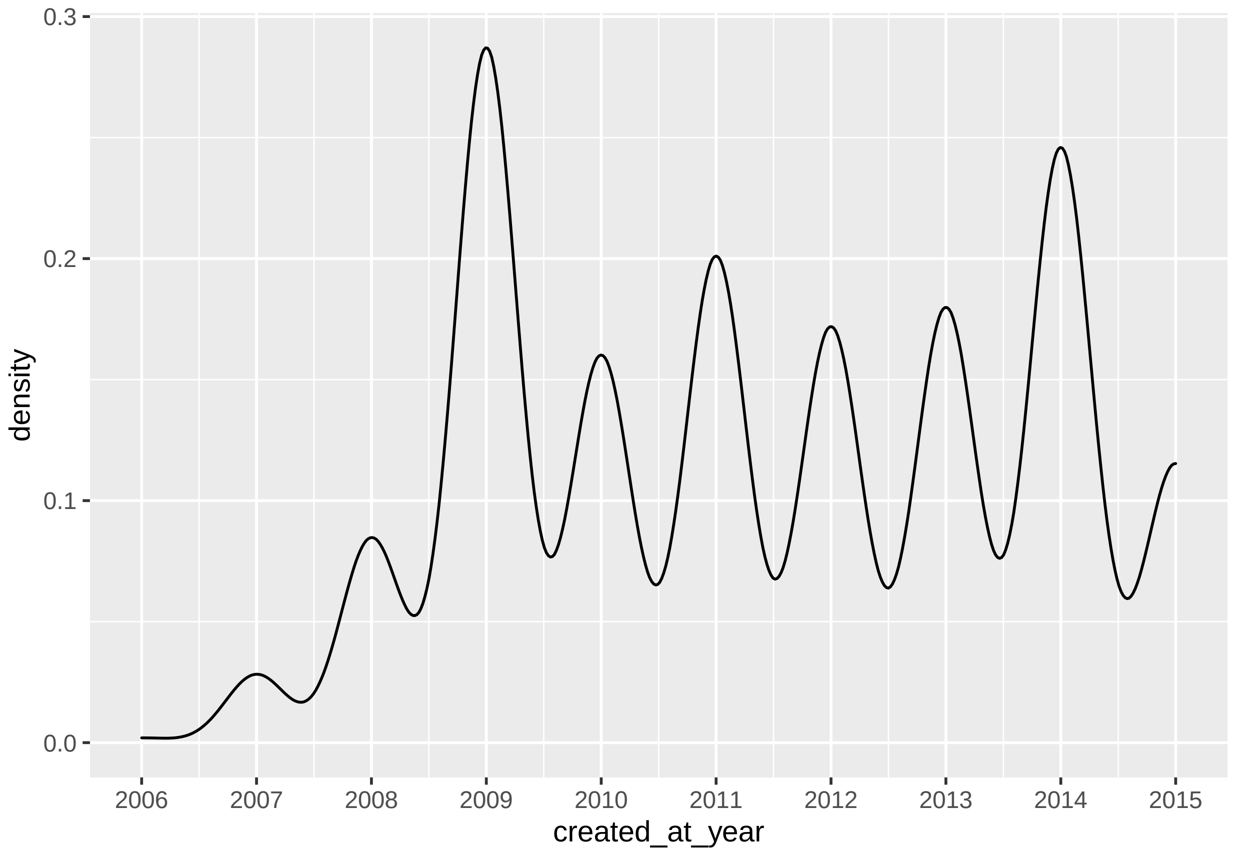
```r
# Regular Pie Chart
pie(df$prop,
  labels = paste0(df$country, " ", df$percentage_labels),
  col = brewer.pal(5, "Dark2"),
  cex = 1
)
```



```r
rm(df)
```

(f) Create kernel density plot of created_at_year variable and interpret the result

```r
ggplot(M01_quasi_twitter, aes(x = created_at_year)) +
  geom_density() +
  scale_x_continuous(
    breaks = c(
      2006, 2007, 2008,
      2009, 2010, 2011,
      2012, 2013, 2014,
      2015, 2016, 2017
    ),
    labels = c(
      "2006", "2007", "2008",
      "2009", "2010", "2011",
      "2012", "2013", "2014",
      "2015", "2016", "2017"
    )
  )
```



We can observe that the highest number of accounts are created in the year 2009. We can also observe that a sudden surge in the number of accounts was observed in 2014.

Problem 3

(a) Standardize the data and create new dataset with standardized data and name it Ndata.

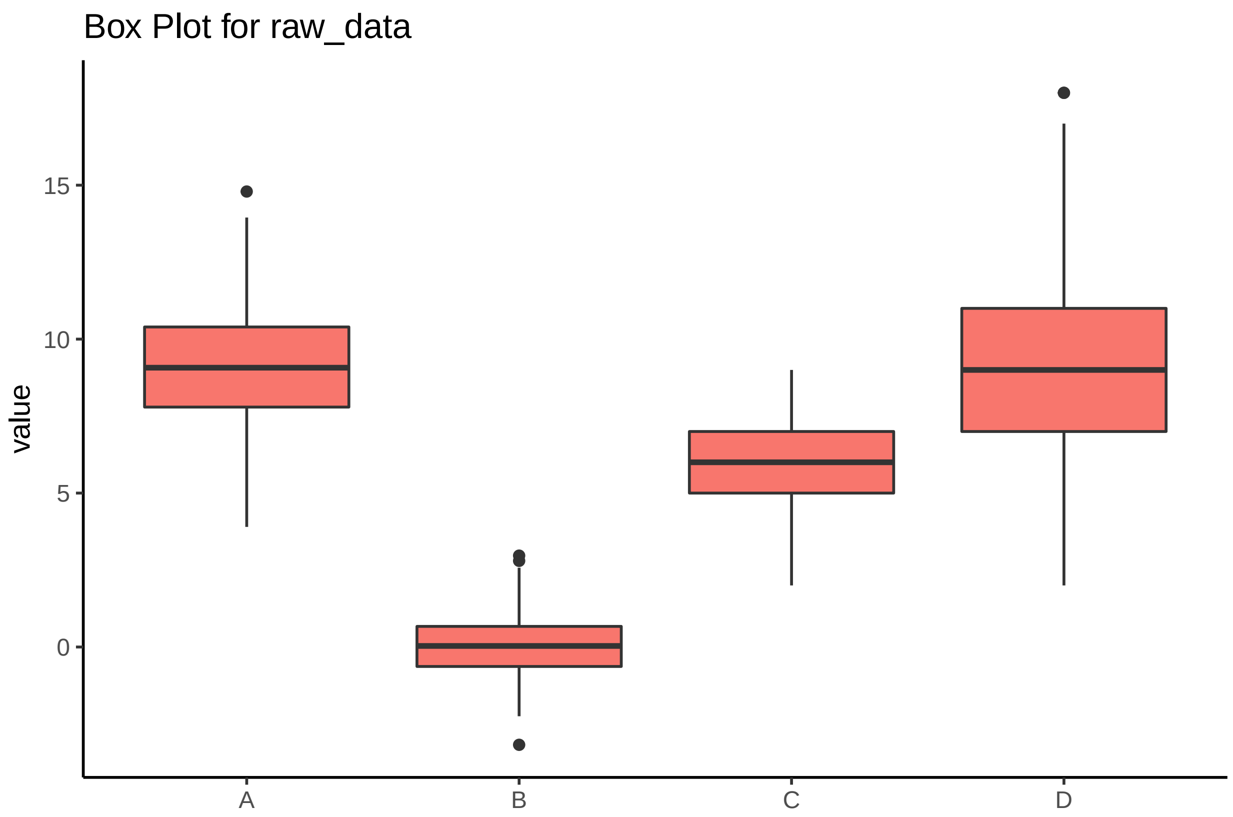
```
# Import the raw_data.csv file
raw_data <- data.frame(read.csv("./data/raw_data.csv"), stringsAsFactors = FALSE)

# Standardize the data frame using the scale function
Ndata <- as.data.frame(scale(raw_data))
```

(b) Create the boxplot of all the variables in their original form.

```
# Reshape the dataframe to a shape suitable for plotting
df <- melt(raw_data)
```

```
## No id variables; using all as measure variables
```

```
# Create box plots
ggplot(df, aes(variable, value, fill = "value")) +
  geom_boxplot() +
  theme_classic() +
  ggtitle("Box Plot for raw_data") +
  labs(x = "") +
  theme(legend.position = "none")
```



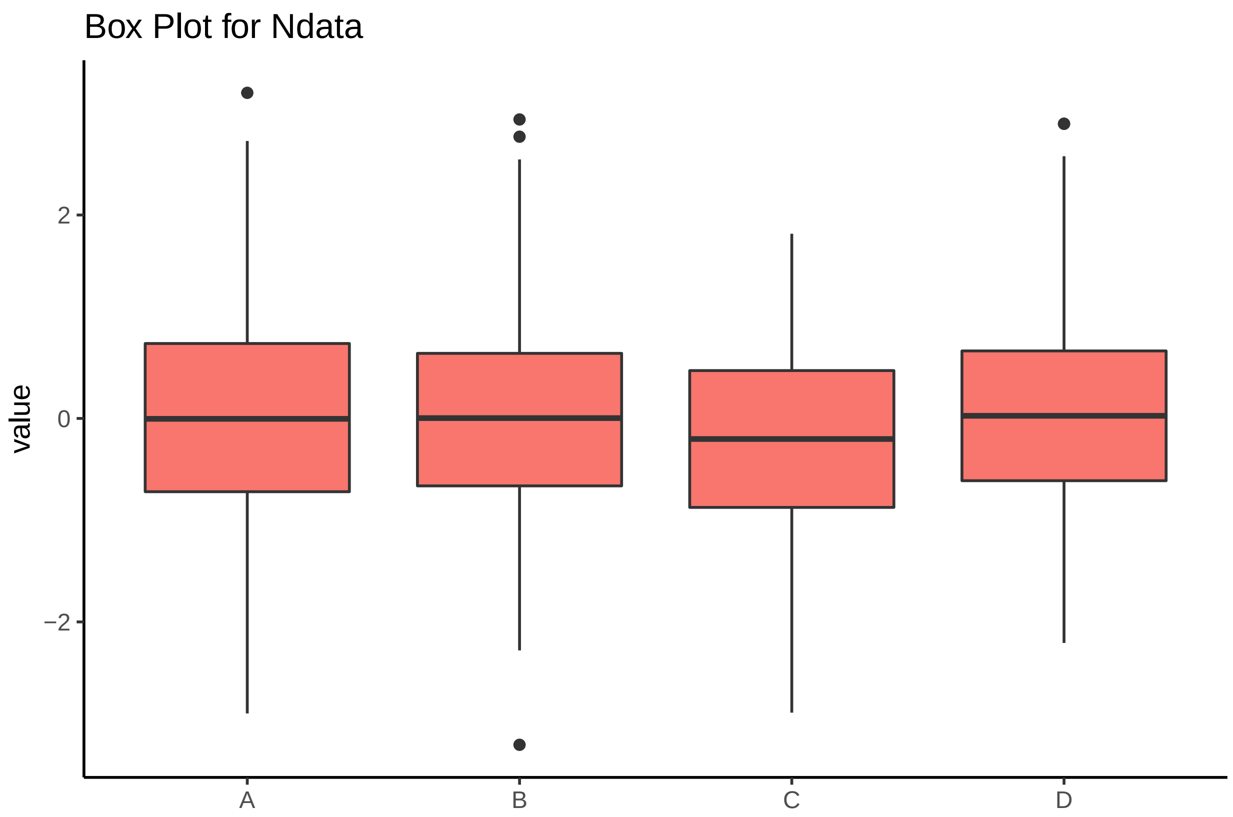
```
rm(df)
```

(c) Create boxplot of all the variables in their standardized form

```
# Reshape the dataframe to a shape suitable for plotting
df <- melt(Ndata)
```

```
## No id variables; using all as measure variables
```

```
# Create box plots
ggplot(df, aes(variable, value, fill = "value")) +
  geom_boxplot() +
  theme_classic() +
  ggtitle("Box Plot for Ndata") +
  labs(x = "") +
  theme(legend.position = "none")
```



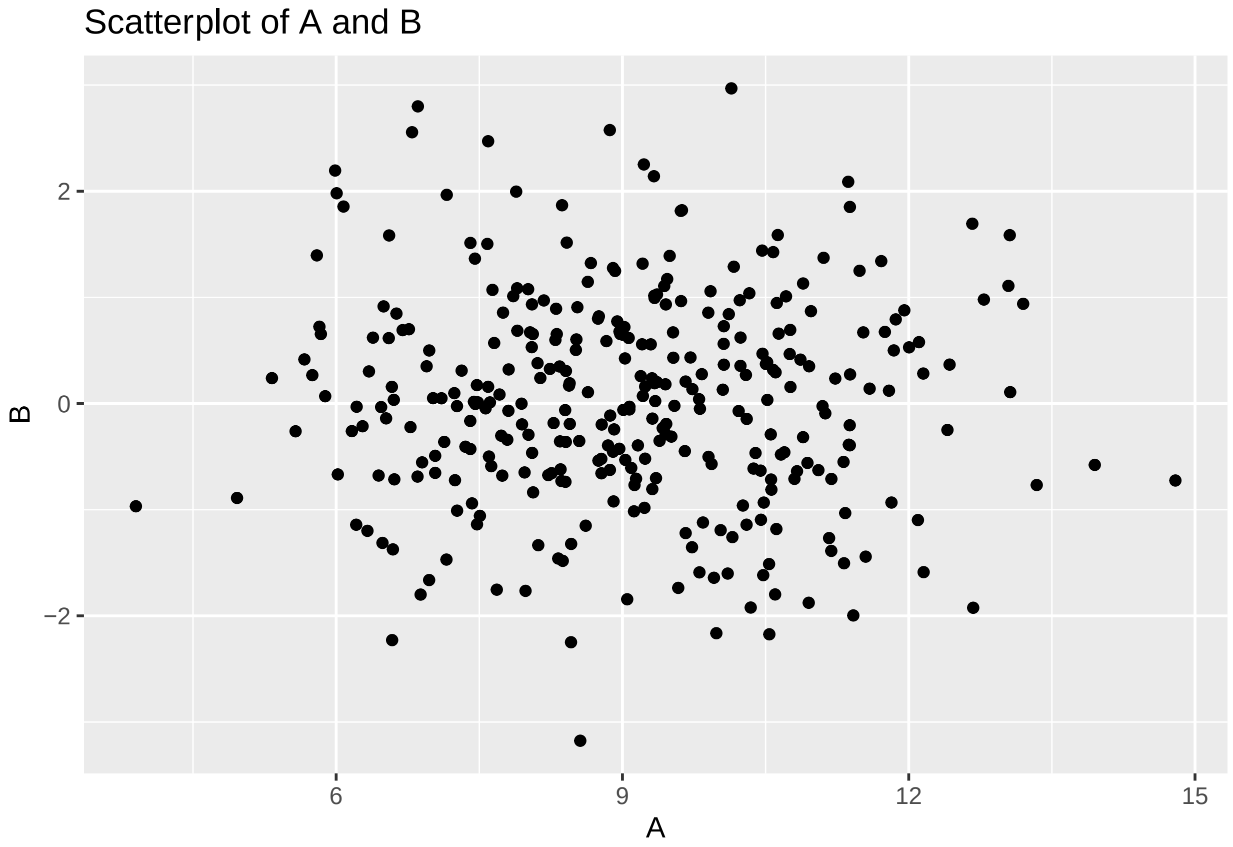
```
rm(df)
```

(d) Compare the result of part b and part c; interpret your answer.

- As we can see from both boxplots, the standardization step transforms all of the columns to a std. normal distribution with a mean of 0 and a standard deviation of 1. This also brings the data to a same scale.

(e) Prepare scatter plot of variables A and B. How are the data correlated in these variables? Interpret your answer.

```
# Creating the scatterplots
ggplot(raw_data, aes(A, B)) +
  geom_point() +
  ggtitle("Scatterplot of A and B")
```



By analyzing the scatterplot of A and B, it is safe to say that they have little to no correlation among themselves, i.e they are independent.