INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT127-3-2-PFDA

Programming for Data Analysis

UC2F2006CS(DA)

HAND OUT DATE: 9th JULY 2020

HAND IN DATE: 1TH SEPTEMBER 2020

PERCENTAGE: 50%

NAME	:	**TEA BOON SERN TP051641**
INTAKE CODE	:	**UC2F2006CS(DA)**
MODULE CODE	:	**CT127-3-2-PFDA**
ASSIGNMENT	:	**HOURLY WEATHER DATA ANALYSIS**
LECTURER	:	**DR. WADDAH WAHEEB HASSAN SAEED**

Table of Contents

Table of Figure

Introduction

This study is going to analysis hourly weather data set by using various of techniques to retrieve necessary information which can be used for decision making in future prediction. The dataset which is being used is related to the hourly meteorological data for LaGuardia Airport (LGA) and John F. Kennedy International Airport (JFK) in United States (USA). It contains a total of 15 columns and 17412 rows of data. The analysis is being carried out via R Studio. A lot of R programming concepts are being applied for doing this analysis such as, data visualisation, data exploration and data manipulation as well. For instance, plot a scatter plot to study the relationship between each of the variables so that can be using for future weather prediction. Also, distribution of the variables to see the pattern of change.

Assumption

There are few columns in the dataset contain missing values such as wind direction, wind speed, pressure and wind gust speed. I assume that replacing the missing values by mean of each month is best fit for the columns of wind speed, pressure and wind gust speed. It is because mean imputation is much easier to be applied and understood by others who just have the basic knowledge in statistical compare to other imputation methods. Meanwhile, I assume that mode imputation is most suitable to replace the missing value of wind direction. As wind direction in degree is considered as categorical data. The most frequency of wind direction of each month will be used to fill the NA of the particular month. Not only that, I also predict that there is a strong relationship between the variable temperature and dew point. Lastly, the result of the analysis can be used to predict future weather data.

Data Pre-Processing

```
#import csv file
data = read.csv(file = "C:/Users/HP/Documents/assign.csv", header=TRUE, sep=",")
head(data)
```

```
  origin year month day hour  temp  dewp humid wind_dir wind_speed wind_gust precip pressure visib        time_hour
1    JFK 2013     1   1    1 39.02 26.06 59.37      260   12.65858        NA      0   1012.6    10 01/01/2013 01:00
2    JFK 2013     1   1    2 39.02 26.06 59.37      270   11.50780        NA      0   1012.4    10 01/01/2013 02:00
3    JFK 2013     1   1    3 39.92 26.96 59.50      260   14.96014        NA      0   1012.7    10 01/01/2013 03:00
4    JFK 2013     1   1    4 39.92 28.04 62.21      250   17.26170        NA      0   1012.6    10 01/01/2013 04:00
5    JFK 2013     1   1    5 39.02 26.96 61.63      260   14.96014        NA      0   1012.1    10 01/01/2013 05:00
6    JFK 2013     1   1    6 37.94 26.96 64.29      260   13.80936        NA      0   1012.6    10 01/01/2013 06:00
```

The code above is applied to import csv file to RStudio and some of the data in the file is shown as well.

```
#Install & load ggplot2 package
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

Code for install and load packages of “ggplot2” and “dplyr”.

```
summary(data)
```

```
> summary(data)
    origin               year          month             day             hour            temp            dewp            humid       
 Length:17412       Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   : 0.00   Min.   :12.02   Min.   :-9.94   Min.   : 12.74  
 Class :character   1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 6.00   1st Qu.:39.92   1st Qu.:26.06   1st Qu.: 46.85  
 Mode  :character   Median :2013   Median : 7.000   Median :16.00   Median :11.00   Median :55.04   Median :42.08   Median : 61.15  
                    Mean   :2013   Mean   : 6.504   Mean   :15.68   Mean   :11.49   Mean   :55.12   Mean   :41.23   Mean   : 62.26  
                    3rd Qu.:2013   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:17.00   3rd Qu.:69.98   3rd Qu.:57.02   3rd Qu.: 78.66  
                    Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :23.00   Max.   :98.96   Max.   :78.08   Max.   :100.00  

    wind_dir       wind_speed        wind_gust         precip             pressure          visib         time_hour        
 Min.   :  0.0   Min.   : 0.000   Min.   :16.11   Min.   :0.000000   Min.   : 983.8   Min.   : 0.000   Length:17412      
 1st Qu.:120.0   1st Qu.: 6.905   1st Qu.:21.86   1st Qu.:0.000000   1st Qu.:1012.9   1st Qu.:10.000   Class :character  
 Median :220.0   Median :10.357   Median :25.32   Median :0.000000   Median :1017.7   Median :10.000   Mode  :character  
 Mean   :201.9   Mean   :11.046   Mean   :26.18   Mean   :0.004183   Mean   :1017.9   Mean   : 9.245                     
 3rd Qu.:300.0   3rd Qu.:14.960   3rd Qu.:29.92   3rd Qu.:0.000000   3rd Qu.:1023.1   3rd Qu.:10.000                     
 Max.   :360.0   Max.   :42.579   Max.   :66.75   Max.   :0.820000   Max.   :1042.1   Max.   :10.000                     
 NA's   :204     NA's   :3        NA's   :13877                      NA's   :1794                                        
```

The summary function is being used to produce result summaries if the results of a variety of model fitting functions and also determine the missing value of each attributes by labelling with NA’s. Based on the result above, there are four variables have missing values which are “wind_dir”, “wind_speed”, “wind_gust” and “pressure”.

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

A function getmode is being generated to find the mode of certain data for this project.

Filling missing values

As the hourly weather data is containing a various of missing values, so some techniques are applied to fill the missing values. Figure below is the code that I applied to fill the missing values.

```
#Data Pre-Processing
#Filling missing data in wind direction by using mode
dirContainNA=filter(data, is.na(wind_dir)) %>%
  select(month) %>%
  unique()

data = mutate(data, wind_dir_n=wind_dir)

for (i in dirContainNA$month){
  fddir= filter(data, month==i, !is.na(wind_dir)) %>%
    select(wind_dir)
  mode = getmode(fddir$wind_dir)
  data = mutate(data, wind_dir_n=ifelse(is.na(wind_dir) & month==i, mode, wind_dir_n))
}
```

```
#Filling missing value for wind speed by using mean
data = mutate(data, wind_speed_n=wind_speed)
wspeed = data$wind_speed
meanspeed = mean(wspeed, na.rm = TRUE)
data = mutate(data, wind_speed_n=ifelse(is.na(wind_speed), meanspeed, wind_speed_n))
```

```
#Filling missing value for pressure by using mean
pressureContainNA=filter(data, is.na(pressure)) %>%
  select(month) %>%
  unique()

data = mutate(data, pressure_n=pressure)

for (i in pressureContainNA$month){
  fd= filter(data, month==i, !is.na(pressure)) %>%
    select(pressure)
  avg = mean(fd$pressure, na.rm = TRUE)
  data = mutate(data, pressure_n=ifelse(is.na(pressure) & month==i, avg, pressure_n))
}
```

```
#Filling missing value for wind_gust
gustContainNA=filter(data, is.na(wind_gust)) %>%
  select(month) %>%
  unique()

data = mutate(data, wind_gust_n=wind_gust)

for (i in gustContainNA$month){
  fd2= filter(data, month==i, !is.na(wind_gust)) %>%
    select(wind_gust)
  avg2 = mean(fd2$wind_gust, na.rm = TRUE)
  data = mutate(data, wind_gust_n=ifelse(is.na(wind_gust) & month==i, avg2, wind_gust_n))
}
```

Analysis Example

Analysis 1: Co-variation between temperature and pressure

```
#Exploratory Data Analysis
#1. Scatter Plot of temperature and pressure.
#In this analysis, the relationship between temperature and pressure of two origin is being analyzed.
ggplot(data, aes(x=temp, y=pressure_n, color=origin, shape = origin))+
  geom_point() + theme_light()+
  labs(title = "Scatter plot of temperature and pressure", x="Temperature (°F)", y= "Pressure (millibars)")+
  geom_smooth(method = "lm", se = FALSE, color = "black")
cor(x=data$temp, y = data$pressure_n, use = "complete.obs")
```

The query above is used to plot a scatter plot between temperature and pressure variables based on each origin to study the relationship between them. The title "Scatter plot of temperature and pressure" is added to the graph, x-axis is labelled as "Temperature (°F)", y-axis is labelled as "Pressure (millibars)" and a black regression equation is added as well by function geom_smooth() to identity the relationship between two of the variables. Method = "lm" is used to plot the line in a linear model and se = FALSE is to remove the confidence intervals around the smooth. Color and shape function is added to identify the data is belonging to which origin as showed in the graph below. Cor() fucntion is used to determine the correlation coefficient between the two variables and use="complete.obs" is to handle missing value by casewise deletion or return error if there are no complete cases.

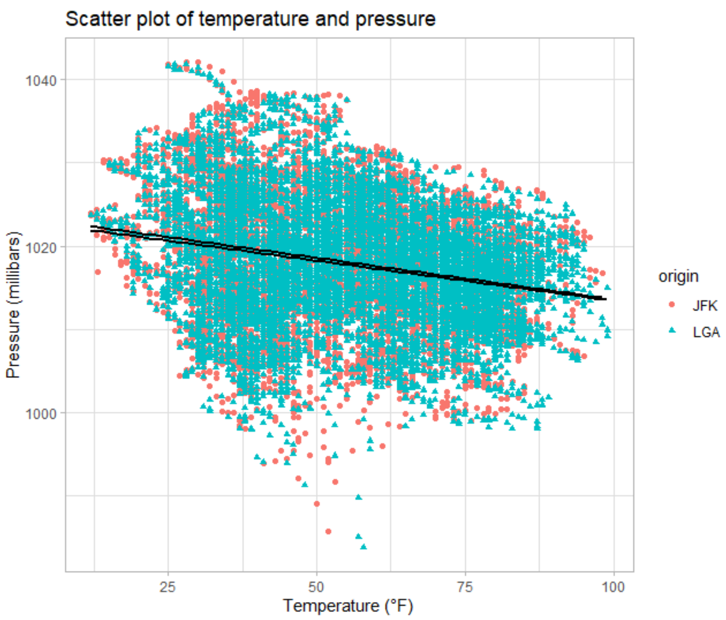

Figure 1: Scatter plot between temperature and pressure

According to the output above, there is a negative linear relationship between the two variables as the regression line is moving down. When temperature increase, pressure will decrease. The correlation between the two variables is -0.2421, indicating the relationship between them is weak.

Analysis 2: Co-variation between temperature and dew point

```
#2. Scatter plot between temperature and dew point
# In this analysis, the relationship between temperature variable and dew point variable is being identify.
ggplot(data, aes(x = temp, y = dewp)) + geom_point()+
  theme_light()+
labs(title="Scatter plot between temperate and dew point", x="Temperature (°F)", y="Dew point (°F)")+
geom_smooth(method="lm")
cor(x=data$temp, y = data$dewp, use = "complete.obs")
```

The code above is applied to plot a scatter plot between temperature and dew point. The title of the plot is labelled as “Scatter plot between temperature and dew point”, while x-axis is relabelled as “Temperature (°F)” and y-axis is “Dew point (°F)”. The background of the graph is being changed by the code of theme_light(). A regression line is drawn with the function geom_smooth(method = “lm”). Method = “lm” is used to plot the line in a linear model. Lastly, cor() fucntion is used to determine the correlation coefficient between the two variables and use=“complete.obs” is to handle missing value by casewise deletion or return error if there are no complete cases.

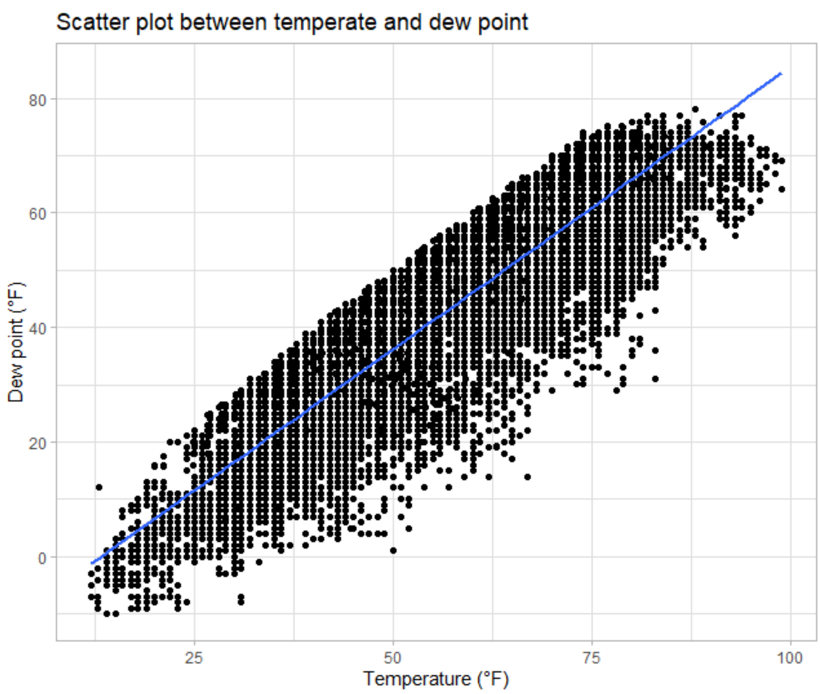

Figure 2: Scatter plot between temperature and dew point

The scatter plot in figure 2 show the co-variation between temperature and dew point. Based on observation, there is a strong positive linear relation between the two variables, meanings that dew point is highly affected by temperature. When temperature increase, dew point will increase. The correlation between the two variables is 0.896.

Analysis 3: Co-variation between temperature and humidity in January

```
#3 Scatter plot between temperature and humidity in January
# For this example, the co-variation between temperature and humidity of January is being analyzed.
data %>%
  filter(month == 1) %>%
  ggplot(aes(x=temp, y = humid)) + geom_point() +
  labs(title = "Scatter plot between temperature and humidity of January", x="Temperature (°F)", y="Humidity")+
  geom_smooth(method="lm")
```

```
jan = data %>%
  filter(month ==1) %>%
  select(temp, humid)
cor(x=jan$temp, y = jan$humid, use = "complete.obs")
```

The code above is applied to plot a scatter plot between temperature and humidity in January. The title of the plot is labelled as “Scatter Plot between temperature and humidity of January”, while x-axis is relabelled as “Temperature (°F)” and y-axis is “Humidity”. A regression line is drawn with the function geom_smooth(method = “lm”). Method = “lm” is used to plot the line in a linear model. Using filter and select function to figure out the data of temperature and humidity in January and stored it in a variable called “jan” and cor() function is being used find the covariance between the two variables in January. Use=“complete.obs” is to handle missing value by casewise deletion or return error if there are no complete cases.

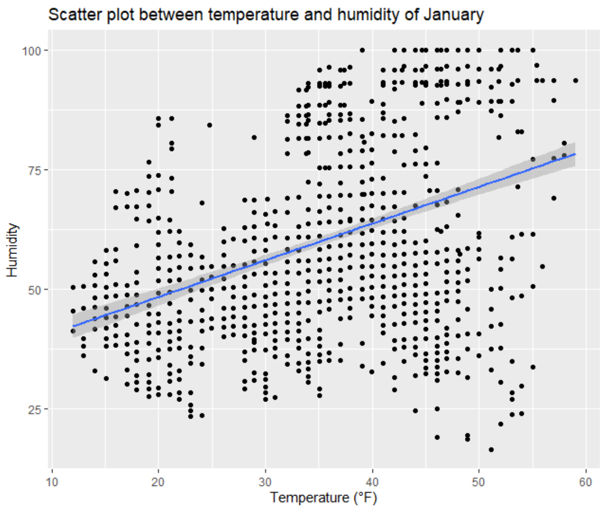

Figure 3: Scatter plot between temperature and humidity

According to the scatter plot above, there is a weak linear relationship between the two variables in January. However, a positive correlation can be observed between the variables, indicates once temperature increase, humidity will slightly increase. The correlation coefficient of the two variables is 0.3758. It means that their relationship is weak.

Analysis 4: Distribution of wind direction of March

```
#4 Histogram of wind direction
# For this example, distribution of wind direction of March is being showed.
march = data %>%
  filter(month == 3)%>%
  select(wind_dir_n)
h1 = ggplot(march, aes(x=wind_dir_n)) + geom_histogram(binwidth = 10) +
       labs(title = "Histogram of Wind Direction of March", x = "Wind Direction")
e1 = ggplot_build(h1)
wind3data = data.frame(xmin = e1$data[[1]]$xmin, xmax = e1$data[[1]]$xmax, y = e1$data[[1]]$y)
```

Using filter and select function to figure out the data of wind direction in March and stored it in a variable called “march”. The code above is applied to plot a histogram of wind direction of March. The title of the plot is labelled as “Histogram of Wind Direction of March”, while x-axis is relabelled as “Wind Direction (degree°)”. The details of the histogram is created by ggplot_bulid() function and stored in variable “e1”. However, for easier identification, the values of xmin, xmax and y is being stored in a data frame.

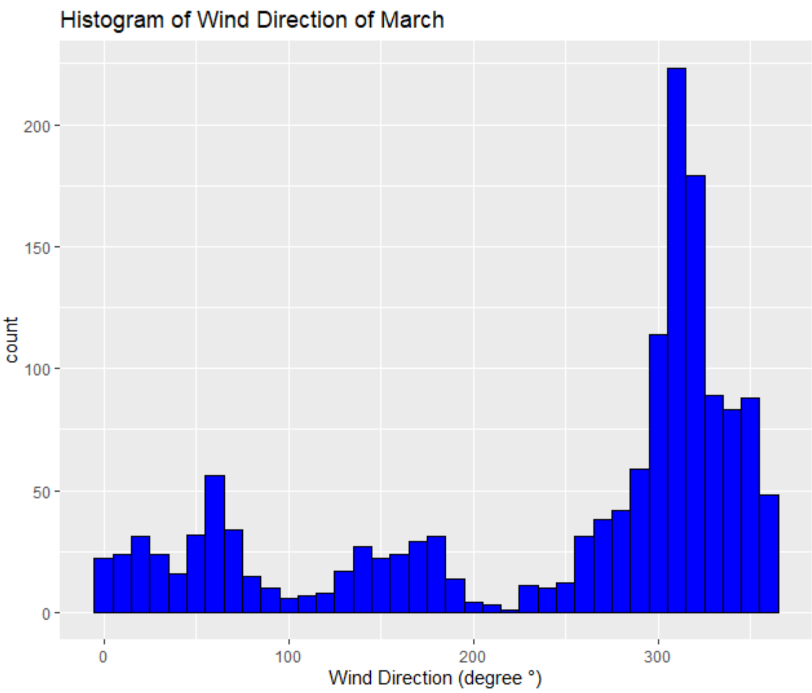

Figure 4: Histogram of Wind Direction of each month

From the histogram above, it shows that the distribution of wind direction of each month. The spread of the histogram is from 0° until 360°. As clearly shown in the figure, the distribution is left-skewed and there is one peak when the wind direction is between 305° and 315°. The frequency of the peak is 223. There are few directions is between 195° and 225°, which are 8 cases only.

Analysis 5: Relationship between humidity and visibility of each month

```
#5 Scatter plot of humidity and visibility
# For this example, the relationship between relative humidity and visibility of each month is analyzed.
ggplot(data, aes(x=humid, y=visib)) + geom_point() + facet_wrap(~month)+
  labs(title = "Scatter plot of humidity and visibility of each month", x="Humidity", y="Visibility (miles)")+
  geom_smooth(method ="lm")
cor(x=data$humid, y = data$visib, use = "complete.obs")
```

The code above is run to plot scatter plot between humidity and visibility group by each month. The title of the plot is labelled as "Scatter plot of humidity and visibility of each month", while x-axis is relabelled as "Humidity" and y-axis is relabelled as "Visibility (miles)". Facet_wrap(~month) function is to group the data by month. A regression line is plotted as well by applying geom_smooth(method = "lm") function. Method = "lm" is used to plot the line in a linear model. Lastly, cor() function is being used find the covariance between the two variables and use="complete.obs" is to handle missing value by casewise deletion or return error if there are no complete cases.

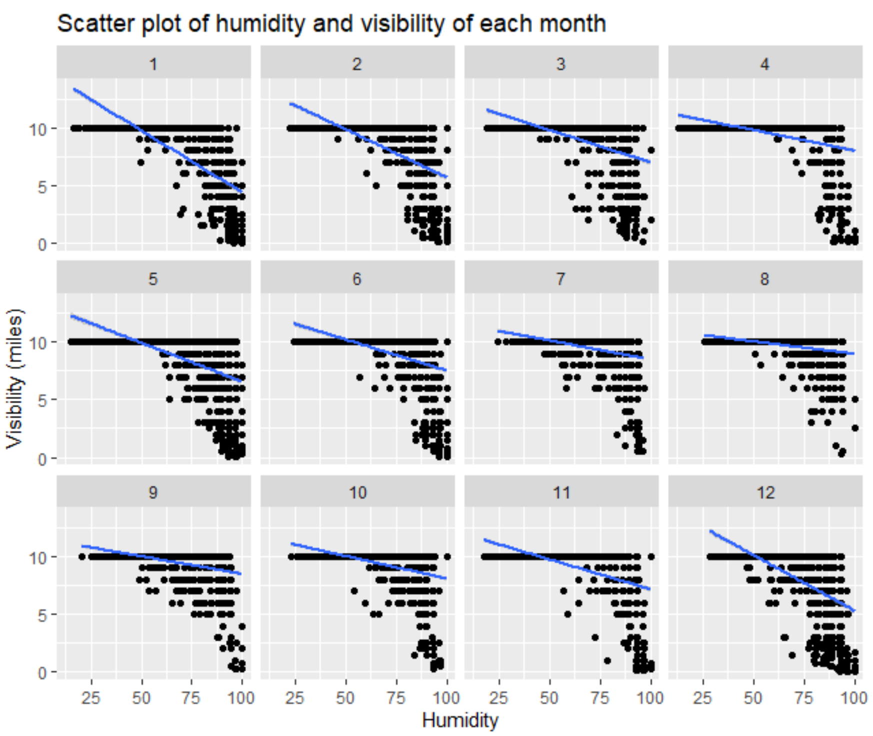

Figure 5: Scatter plot of humidity and visibility of each month

Based on the scatter plot above, there is a negative linear relation between humidity and visibility of each month. The correlation coefficient of the two variables is -0.5186. Therefore, visibility is affected by humidity. When humidity increase, then visibility will decrease.

Analysis 6: Summary statistical of humidity

```
#6 Boxplot of humidity
# For this example, summary statistical along with individual outliers is being showed.
b1 = ggplot(data, aes(y=humid, x=1)) + geom_boxplot() + labs(title = "Boxplot of Humandity", y="Humidity")
e2 = ggplot_build(b1)
```

The code above is run to plot box plot of humidity to show its summary statistical. The title of the plot is labelled as “Boxplot of Humidity”, while y-axis is relabelled as “Humidity”. The details of the boxplot is created by ggplot_bulid() function and stored in variable “e2”.

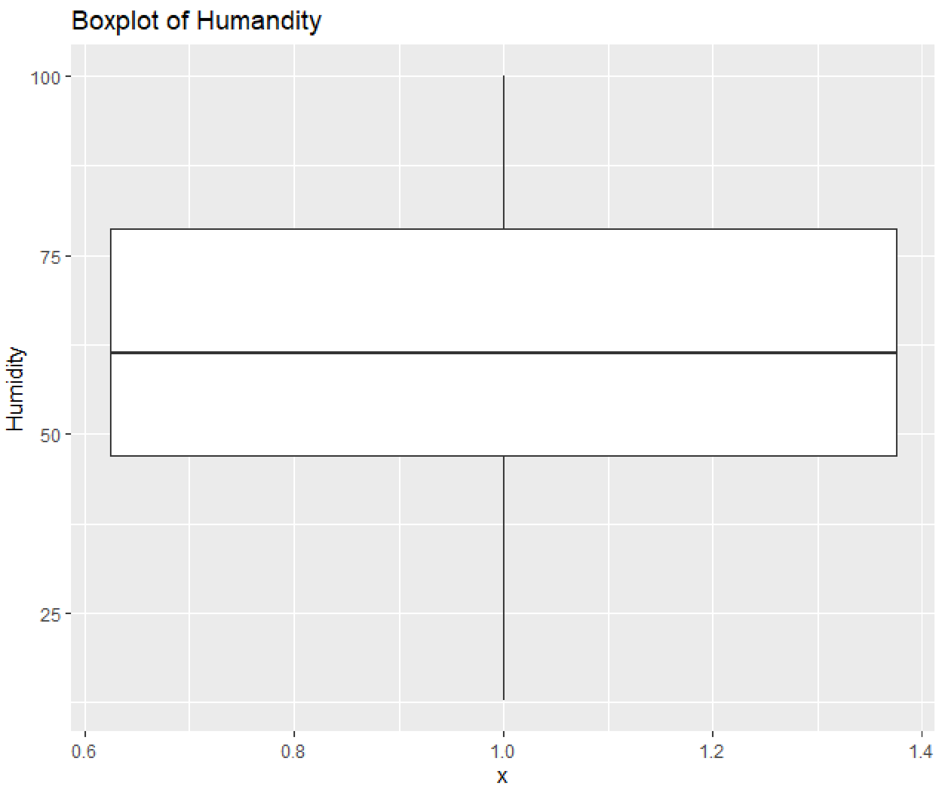

Figure 6: Boxplot of humidity

Based on the boxplot in figure 6, it shows the summary statistical of humidity. Due to the observation, there is no any outlier for the variable. The minimum value for humidity is 12.74, maximum value is 100, median is 61.15, upper quartile is 78.66 and lower quartile is 46.85. Besides that, humidity has a normal distribution and do not has any outlier.

Analysis 7: Distribution of visibility of June

```
#7 Polygon of visibility in June
# For this analysis, distribution of visibility of June of two origin is shown.
pol1 = data %>%
  filter(month == 6) %>%
  ggplot(aes(x=visib, color=origin)) + geom_freqpoly(binwidth = 1) +
  labs(title = "Polygon of visibility of June", x="Visibility (miles)")
pol2 = ggplot_build(pol1)
poldata = data.frame(xmin = pol2$data[[1]]$xmin, xmax = pol2$data[[1]]$xmax, y = pol2$data[[1]]$y)
```

According to the code, it is applied to plot a frequency polygon of visibility of June of two origin to show its distribution. Filter function is being used to filter the data in June. The title of the graph is labelled as “Polygon of Visibility of March” and x-axis is named as “Visibility (miles)”. The information of the graph is shown by using ggplot_build() function and being stored in variable “pol2”. However, for easier identification, the values of xmin, xmax and y is being stored in a data frame, poldata.

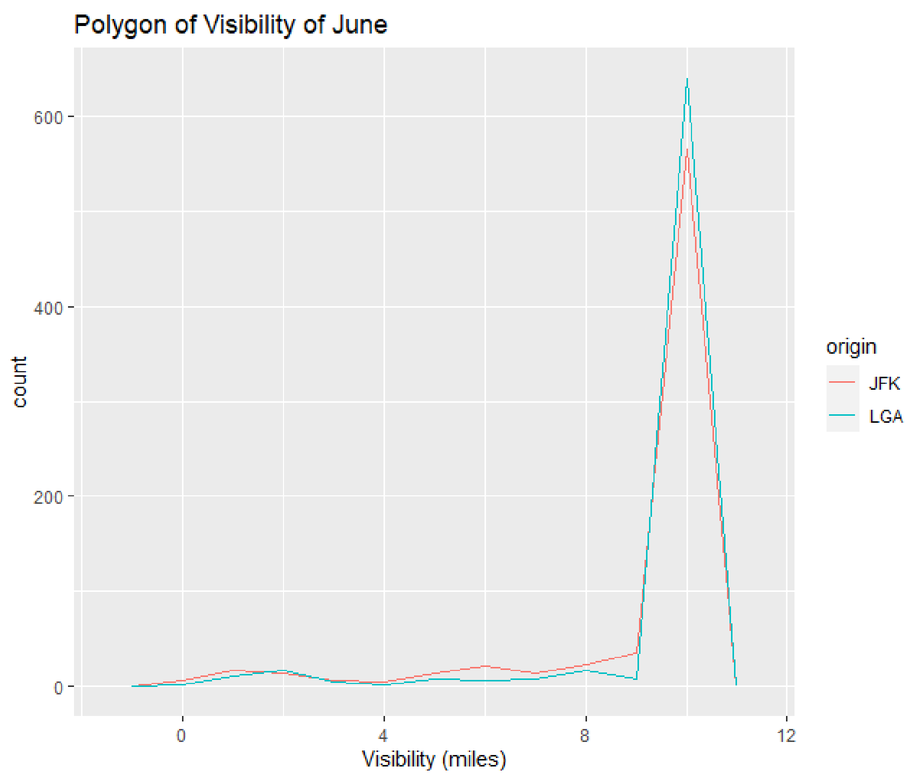

Figure 7: Frequency Polygon - Distribution of Visibility of June

Based on the frequency polygon, the spread of the graph is in range of -0.5 until 11.5 miles. In addition, it clearly shows that the most common value of visibility in June of each origin is between range of 9.5 and 10.5 miles. There are separately 566 and 640 cases in JFK and LGA. However, there are few cases of visibility in range between 0 until 9 and there are no cases with visibility which is more than 10.5 miles.

Analysis 8: Summary statistical of wind speed based on origin

```
#8 Boxplot of wind speed based on origin
# In this example, a boxplot is ploted to identify first quartile, median, third quartile and
# outliers of wind speed of different origins.
g1 = ggplot(data, aes(y = wind_speed_n, x= origin))+geom_boxplot() +
  labs(title = "Boxplot of Wind Speed", x="Origin", y="Wind Speed (mph)")
s1 = ggplot_build(g1)
speedoutlierJFK = data.frame(outlier_JFK = s1$data[[1]]$outliers[[1]])
speedoutlierLGA = data.frame(outlier_LGA = s1$data[[1]]$outliers[[2]])
min(speedoutlierJFK$outlier_JFK)
min(speedoutlierLGA$outlier_LGA)
```

The code above is applied to plot box plot of wind speed based on two different origin to show its summary statistical. The title of the plot is labelled as "Boxplot of Wind Speed", while x-axis is relabelled as "Origin" and y-axis is relabelled as "Wind Speed (mph). The details of the boxplot is created by ggplot_bulid() function and stored in variable "s1". The value of outlier for each origin is separately stored in a data frame. The minimum function is used to identify the starting value of the outliers.

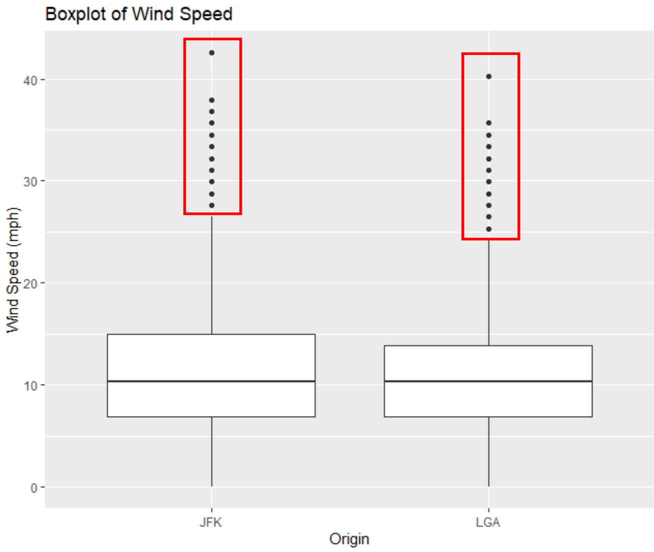

Figure 8: Boxplot of Wind Speed of two Origin

According to the boxplot above, it shows the summary statistical of wind speed based on two origins. From the observation, clearly show that wind speed is having some outliers in both origins and both of them are normal distribution as well. The minimum value, lower quartile and median of wind speed of each origin is same, which is 0mph, 6.9 mph and 10.4mph. It is different for the upper quartile and maximum value in each origin. JFk has an upper quartile of 15mph and maximum value of 26.5mph. However, LGA has an upper quartile of 13.87mph and maximum value of 24.2mph. As conclusion, there is only small difference for the wind speed between the two origin. The value of outliers for JFK is started from 27.6mph while outlier for LGA is started from 25.3mph.

Analysis 9: Relationship of wind speed and pressure

```
#9 Scatter plot of wind speed and pressure
# In this example, the relation between wind speed and pressure is being analyzed.
ggplot(data, aes(x=wind_speed_n, y = pressure_n)) + geom_point() +
  labs(title = "Scatter plot of Wind Speed and Pressure", x="Wind Speed (mph)", y="Pressure (millibars)") +
  geom_smooth(method ="lm")
cor(x=data$wind_gust_n, y = data$dewp, use = "complete.obs")
```

The code above is used to plot a scatter plot of wind speed and pressure to study their relation. The title of the plot is labelled as “Scatter plot of Wind Speed and Pressure”, while x-axis is relabelled as “Wind Speed (mph)” and y-axis is relabelled as “Pressure (millibars)”. geom_smooth(method = “lm”) function is applied to plot a regression line of the two variables. Lastly, correlation coefficient between the two variables is being found by cor function and use=“complete.obs” is to handle missing value by casewise deletion or return error if there are no complete cases.

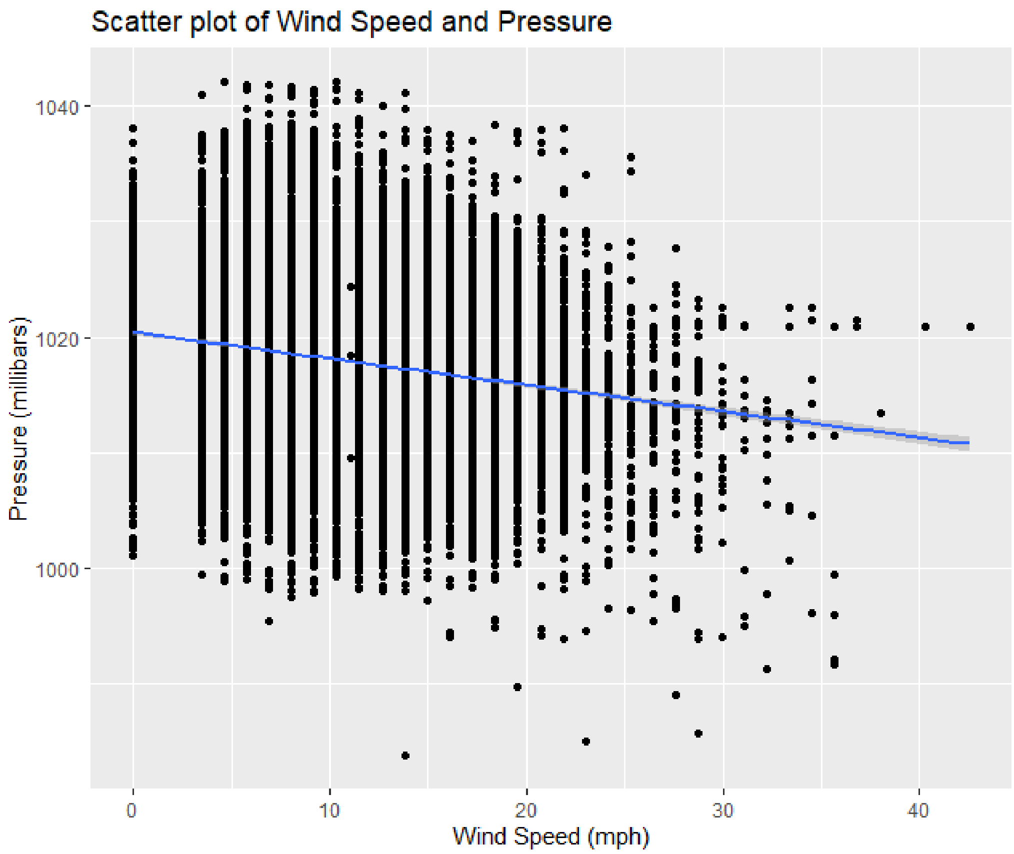

Figure 9: Scatter plot of wind speed and pressure

Figure 9 show the scatter plot of wind speed and pressure. There is a negative linear relation between wind speed and pressure. The correlation coefficient of the two variables is -0.4726 so that the relationship between them is weak. Once wind speed increase, pressure will slightly decrease.

Analysis 10: Summary statistical of Wind Gust Speed

```
#10 Boxplot of wind gust speed
# In this example, a boxplot is plotted to identify lower quartile, median, upper quartile and outliers of wind gust speed.
b2 = ggplot(data,aes(x=1, y=wind_gust_n)) + geom_boxplot() +
  labs(title = "Boxplot of Wind Gust Speed", y="Wind Gust Speed (mph)")
e3 = ggplot_build(b2)
gustoutlier = data.frame(outlier = e3$data[[1]]$outliers[[1]])
min(gustoutlier$outlier)
```

Code above is applied to plot a box plot of wind gust speed to study the summary statistical of the variable. The title of the plot is labelled as “Boxplot of Wind Gust Speed”, while y-axis is relabelled as “Wind Gust Speed (mph). The details of the boxplot is created by ggplot_bulid() function and stored in variable “e3”. The value of outlier is stored in a data frame, gustoutlier. The minimum function is used to identify the starting value of the outliers.

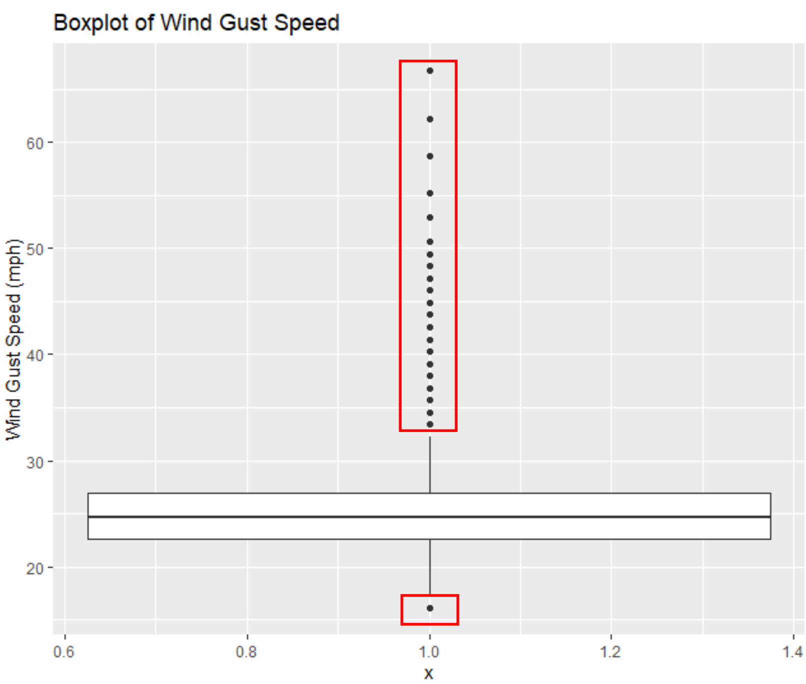

Figure 10: Boxplot of wind gust speed

Regarding to the boxplot of figure 10, it shows the summary statistical of wind gust speed. Due to the observation, there are outliers found in the data from the variable which are shown in a red box and there are two range of outliers. The first range is below 16.1 mph and the second range is more than 34mph. The minimum value for wind gust speed is 17.2617mph, maximum value is 32.22184mph, median is 24.70152mph, upper quartile is 26.92868mph and lower quartile is 22.63911mph. Besides that, wind gust speed has a normal distribution.

Analysis 11: Variation of precipitation

```
#11 Histogram of Percipitation
# In this example, distribution of percipitation is shwon in histogram.
h2 = ggplot(data, aes(x=precip)) + geom_histogram(binwidth = 0.1) +
  labs(title = "Histogram of Percipitation", x = "Percipitation")
e4 = ggplot_build(h2)
precipdata = data.frame(xmin = e4$data[[1]]$xmin, xmax = e4$data[[1]]$xmax, y = e4$data[[1]]$y)
```

The code above is applied to plot a histogram of precipitation. The title of the plot is labelled as "Histogram of Precipitation" and x-axis is relabelled as "Precipitation". The details of the histogram is created by ggplot_bulid() function and stored in variable "e4". However, for easier observation, the values of xmin, xmax and y is being stored in a data frame.

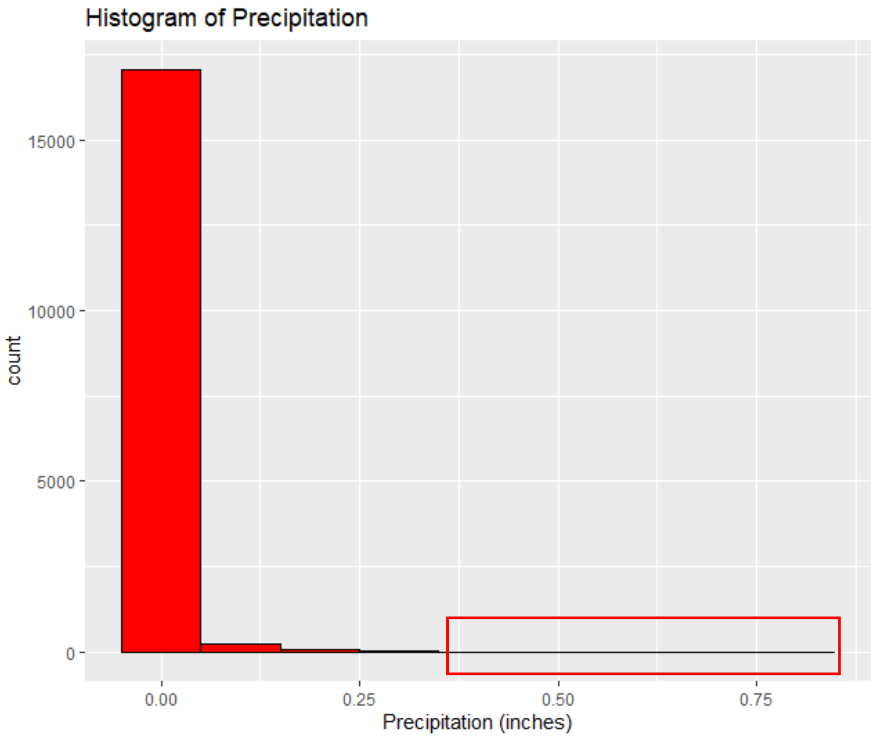

Figure 11: Histogram of Precipitation

Based on the histogram shown in figure 11, it shows that the distribution of precipitation. The spread of the histogram is beginning from -0.05 until 0.85 inches. As clearly shown in the figure, the distribution of precipitation is right-skewed and there is one peak which have the range between -0.05 and 0.05 inches. The frequency of the peak is 17039. There are few precipitations is more than 0.45 inches, which are far away from other data values. Therefore, those values will be considered as outliers which show in red box above.

Analysis 12: Correlation of wind speed and visibility

```
#12 Scatter plot of wind speed and visibility
# In this example, the relationship between wind speed and visibility is being studied.
ggplot(data, aes(x=wind_speed_n, y = visib)) + geom_point() +
  labs(title = "Scatter plot of Wind Speed and Visibility", x="Wind Speed (mph)", y="Visibility (miles)") +
  geom_smooth(method ="lm")
cor(x=data$wind_speed_n, y = data$visib, use = "complete.obs")
```

A scatter plot of wind speed and visibility is plotted by the code above to study their relationship. The title of the plot is labelled as “Scatter plot of Wind Speed and Visibility”, while x-axis is relabelled as “Wind Speed” and y-axis is relabelled as “Visibility”. A regression line between the two variables is being plotted by geom_smooth(method = “lm”) function. Lastly, correlation coefficient between the two variables is being found by cor function and use=“complete.obs” is to handle missing value by casewise deletion or return error if there are no complete cases

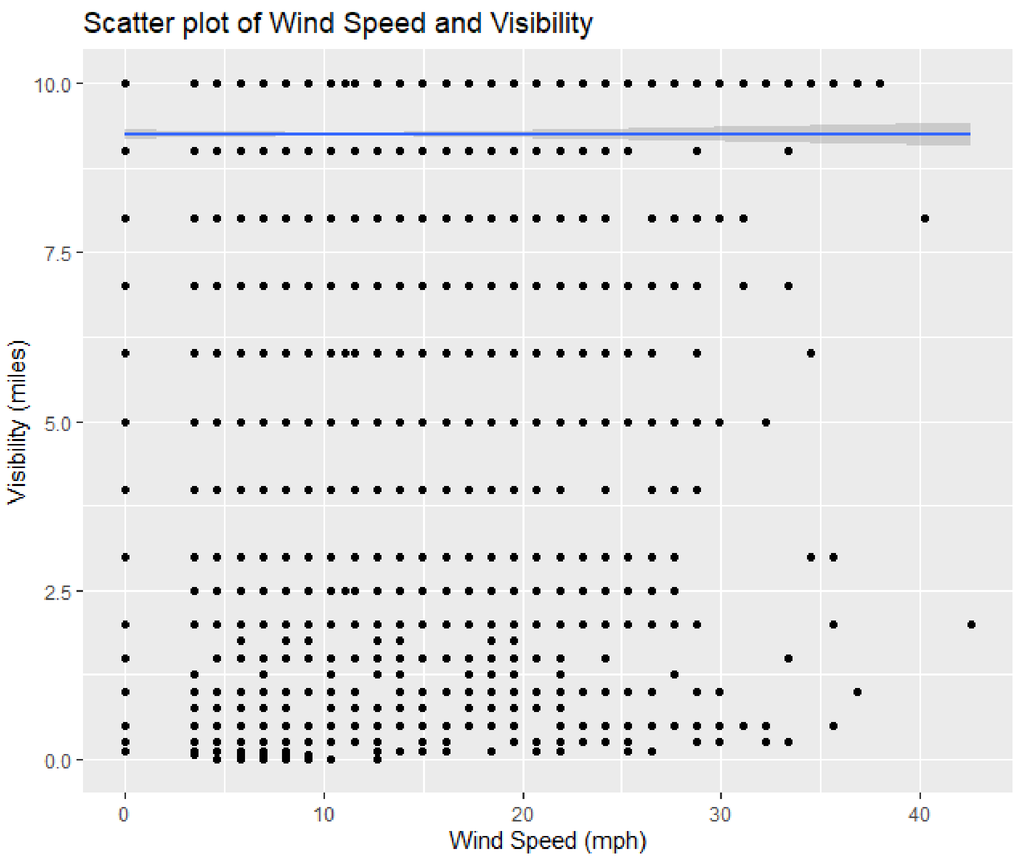

Figure 12: Scatter plot of wind speed and visibility

Figure 12 is showing the scatter plot of wind speed and visibility. The two variables are not corelated with each other as the correlation coefficient of the two variables is 0.0001. Therefore, there is no relationship between the variables.

Analysis 13: Distribution of Pressure

```
#13 Frequency Polygon of pressure
# In this example, distribution of pressure is shown in histogram.
h3 = ggplot(data, aes(x=pressure_n)) + geom_freqpoly(color = "blue") +
   theme_bw() +
  labs(title = "Frequency Polygon of Pressure", x = "Pressure (millibars)")
e5 = ggplot_build(h3)
pressuredata = data.frame(xmin = e5$data[[1]]$xmin, xmax = e5$data[[1]]$xmax, y = e5$data[[1]]$y)
```

Regarding to the code above, it is applied to plot a frequency polygon of pressure to show its distribution. Theme_bw() is the function used to change the background of the graph. The title of the graph is labelled as "Frequency Polygon of Pressure" and x-axis is named as "Pressure (millibars)". The details of the polygon is built by ggplot_build() function and the maximum, minimum value of x and value of y are being stored in a data frame to easier reading.

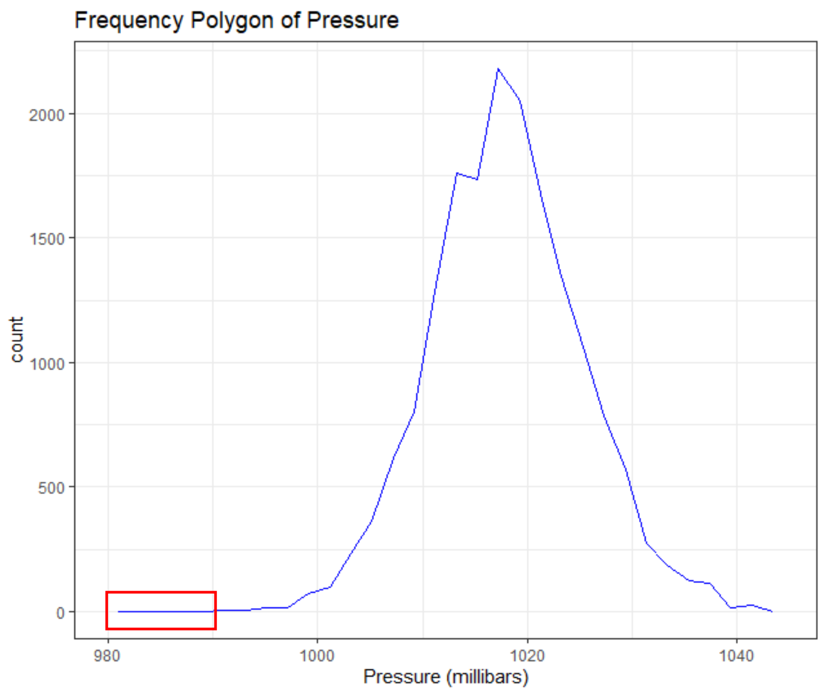

Figure 13: Frequency Polygon of Pressure

According to the frequency polygon shown in figure 13, it shows that the distribution of pressure. The spread of the frequency polygon is in between 982 millibars and 1042 millibars. As clearly shown in the figure, the distribution of pressure is normal distribution and there is one peak at 1018.2397millibars. The frequency of the peak is 2179. There are few cases of pressure which is less than 994.1155millibars, which are far away from other data values. Therefore, those values will be considered as outliers which show in red box above.

Analysis 14: Correlation between temperature and wind gust speed

```
#14 Scatter plot of temperature and wind gust speed
# In this example, relationship between temperature and wind gust speed is being analyzed based on origin.
ggplot(data, aes(x=temp, y = wind_gust_n, color = origin, shape = origin)) + geom_point() +
  labs(title = "Scatter plot of temperature and wind gust speed", x="Temperature (°F)", y="Wind Gust Speed (mph)") +
  geom_smooth(method ="lm", color="black")
cor(x=data$temp, y = data$wind_gust_n, use = "complete.obs")
```

A scatter plot of temperature and wind gust speed is plotted by the code above to study their relationship. The title of the plot is labelled as “Scatter plot of temperature and wind gust speed”, while x-axis is relabelled as “Temperature (°F)” and y-axis is relabelled as “Wind Gust Speed (mph)”. Two black regression line between the two variables based on origin are being plotted by geom_smooth(method = “lm”, color = “black”) function. Lastly, correlation coefficient between the two variables is being found by cor function.

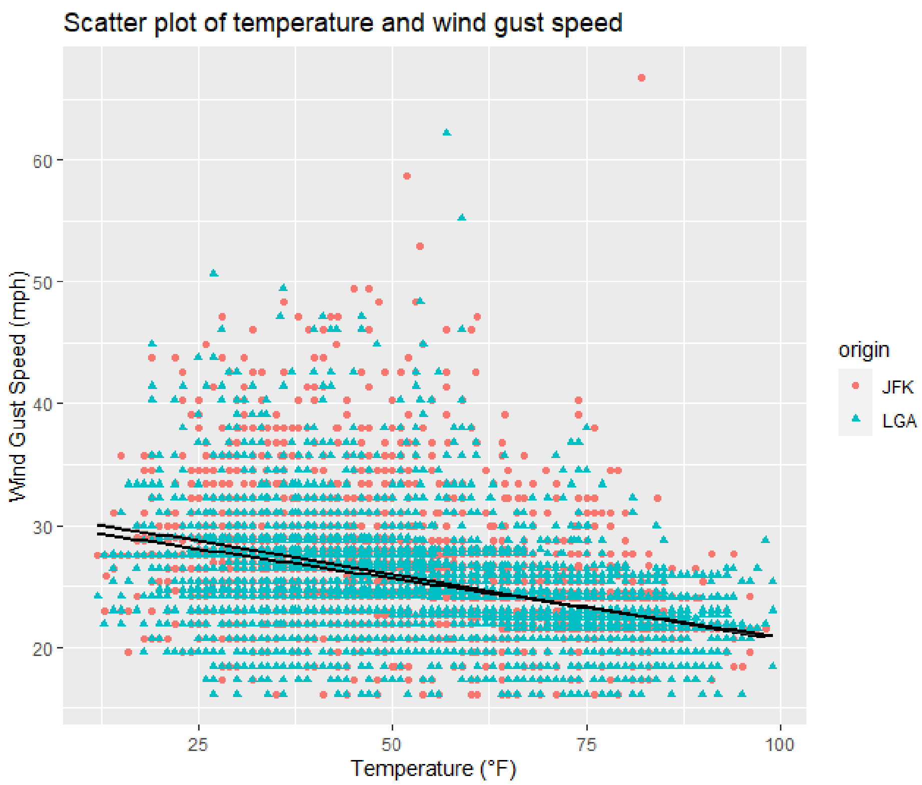

Figure 14: Scatter plot of Temperature and Wind Gust Speed

Based on the scatter plot of figure 14, it clearly shows that there is a negative linear relation between temperature and wind gust speed. The correlation coefficient of the two variables is -0.5241 so that the relationship between them is weak. When temperature increase, wind gust speed will slightly decrease.

Analysis 15: Variance of dew point of July

```
#15 Histogram of Dew Point(July)
# In this analysis, distribution of dew point in July is being showed.
jul = data %>%
  filter(month == 7)%>%
  select(dewp)
h4 = ggplot(jul, aes(x=dewp)) + geom_histogram(color = "white", fill = "black") +
  labs(title = "Histogram of Dew Point in July", x="Dew Point (°F)")
e10 = ggplot_build(h4)
dewdata = data.frame(xmin = e10$data[[1]]$xmin, xmax = e10$data[[1]]$xmax, y = e10$data[[1]]$y)
```

The code above is applied to plot a histogram of dew point of July. Using filter and select function to figure out the data of dew point in July and stored it in a variable called “jul”. The title of the plot is labelled as “Histogram of Dew Point in July” and x-axis is relabelled as “Dew Point (°F)”. The details of the histogram is created by ggplot_bulid() function and stored in variable “e10”. However, for easier observation, the values of xmin, xmax and y is being stored in a data frame.

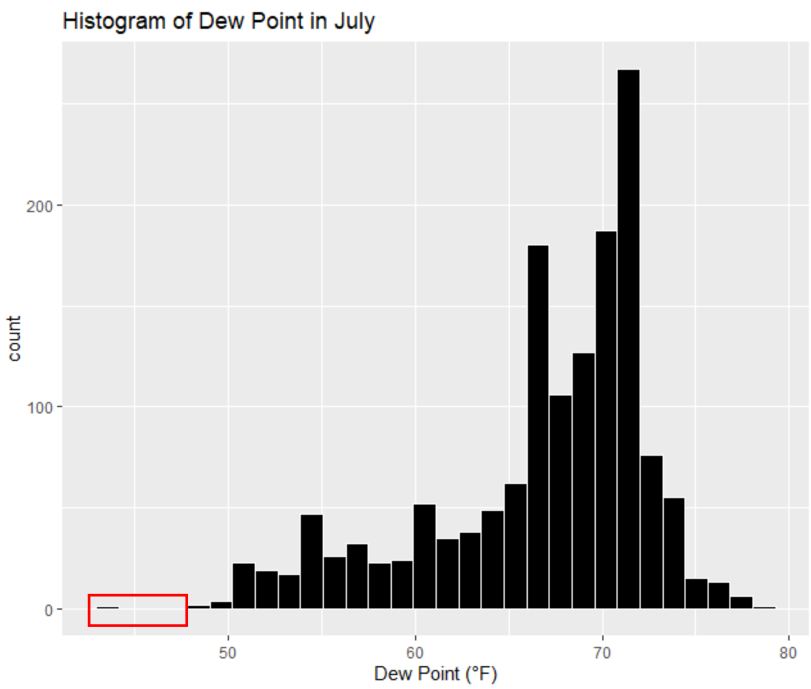

Figure 15: Histogram of Dew Point (July)

Regarding to the histogram shown above, it shows that the distribution of dew point in the month of July. As clearly shown in the figure, the spread of the histogram is between the range of 42°F until 80°F. In addition, the distribution of precipitation is left-skewed and there is one peak which have the range between 70.81°F and 72.02°F. The frequency of the peak is 267. There are few precipitations is less than 47°F, which are far away from other data values. Therefore, those values will be considered as outliers which show in red box above.

Additional feature

Remove outliers for wind speed variable by binning method

```
#Additional feature 1
# In this analysis, binning method is applied to remove outliers of wind speed variable.
#Boxplot of wind speed before applying binning method.
ggplot(data, aes(x=1, y=wind_speed_n)) + geom_boxplot()+
  theme_light()+
  labs(title = "Boxplot of Wind Speed", y="Wind Speed (mph)")

#Boxplot of wind speed after applying binning method.
range = c(-Inf,1,3,7,12,18,24,31,Inf)
newlable = as.integer(c(0,1,2,3,4,5,6,7))
Wind_Speed_Level <- cut(data$wind_speed_n, breaks = range, labels = newlable)
data = mutate(data, Wind_Speed_Level)
box1 = ggplot(data, aes(x=1, y=as.integer(Wind_Speed_Level))) + geom_boxplot()+
  theme_light()+
  labs(title = "Boxplot of Wind Speed Level", y="Wind Speed (mph)")
box2=ggplot_build(box1)
```

The code above is plotting boxplot of wind speed with original data and applying binning method to the data to remove the outliers. The wind speed is being divided into eight level by using break function according to Beaufort Wind Force which has been shown below. Wind speed of level 0 is < 1 mph; level 2 is 1mph <= wind speed <= 3mph, level 3 is 3mph < wind speed <= 7mph, level 4 is 7mph < wind speed <= 12mph, level 5 is 12mph < wind speed <= 18mph, level 6 is 18mph < wind speed <= 24mph, 24mph < wind speed <= 31mph and greater than 31mph is categorized as level 7. A new column “Wind Speed Level” is created to store the values. The graph of boxplot is labelled as “Boxplot of Wind Speed Level” and y-axis is labelled as “Wind Speed (mph)”. The summary of measurement of the boxplot is being built by the ggplot_build function.

Beaufort Wind Force	Wind Average	Speed Range
0	0	<1 kt <1 mph <1 km/h
1	2 kt 2 mph 3 km/h	1-3 kt 1-3 mph 1-5 km/h
2	5 kt 6 mph 9 km/h	4-6 kt 4-7 mph 6-11 km/h
3	9 kt 10 mph 16 km/h	7-10 kt 8-12 mph 12-19 km/h
4	13 kt 16 mph 24 km/h	11-16 kt 13-18 mph 20-28 km/h
5	19 kt 22 mph 34 km/h	17-21 kt 19-24 mph 29-38 km/h
6	24 kt 28 mph 44 km/h	22-27 kt 25-31 mph 39-49 km/h
7	30 kt 35 mph 56 km/h	28-33 kt 32-38 mph 50-61 km/h

(NWS JetStream MAX - Beaufort Wind Force Scale, n.d.)

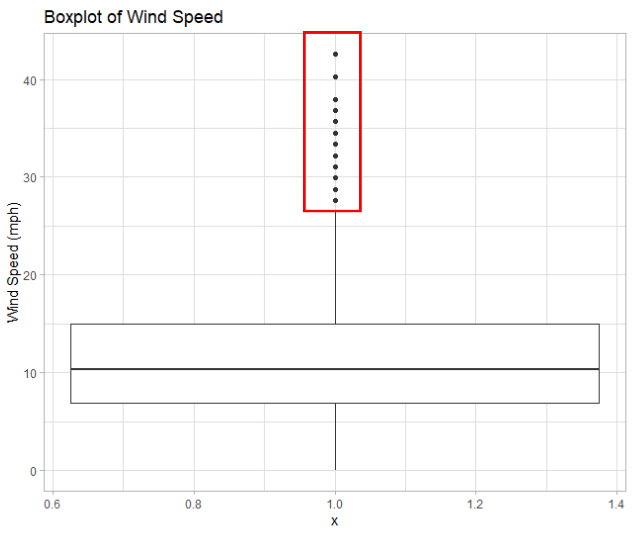

Figure 16: Boxplot of Wind Speed with outliers

According to figure 16, the outliers of wind speed is being shown in the red box.

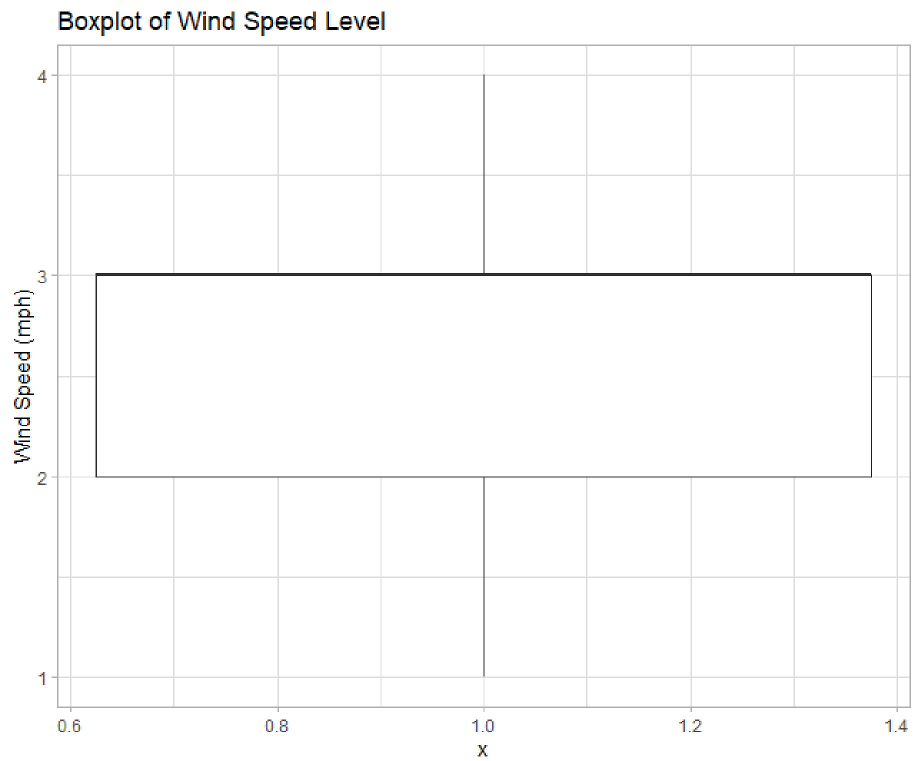

Figure 17: Boxplot of Wind Speed Level

Regarding to the figure 17, there is no any outlier can be detected after applying the binning method to bin the original data. The purpose of remove outliers is to increase the statistically significant of the analysis results (Frost, 2020). It is because outlier will increase the variability in data which leading to decrease statistical power (Frost, 2020). Without outlier, the graph will become tidier.

Hexagonal bin plot of humidity and dew point

```
#Additional feature 2
#In this analysis, hexagonal bins is used to determine the relationship between humidity and dew point.
install.packages("hexbin")
library(hexbin)
hex = ggplot(data, aes(x = humid, y = dewp)) + geom_hex() + theme_bw()+
  labs(title = "Hexagonal bin plot of Humidity and Dew Point", x="Humidity", y="Dew Point (°F)")+
  geom_smooth(method = "lm", color ="red")
cor(x=data$humid, y = data$dewp, use = "complete.obs")
ex = ggplot_build(hex)
hexdata = data.frame(x = ex$data[[1]]$x, y = ex$data[[1]]$y, count = ex$data[[1]]$count)
```

The code above is to install and load package of hexbin. After that, geom_hex() function is used to plot the hexagonal bin plot between humidity and dew point. The title of the hexagonal plot is labelled as “Hexagonal bin plot of Humidity and Dew Point”, x-axis is relabelled as “Humidity” and y-axis is relabelled as “Dew Point (°F)”. A regression line is plotted by the function geom_smooth(method = “lm”) as well. Lastly, the correlation coefficient is shown by cor function and use=“complete.obs” is to handle missing value by casewise deletion or return error if there are no complete cases. The details of the plot are stored in a data frame. The theme_bw() is applied to change the theme of the graph in order to have a better visibility of the graph when projected by a projector during presentation (Hadley et al., n.d.).

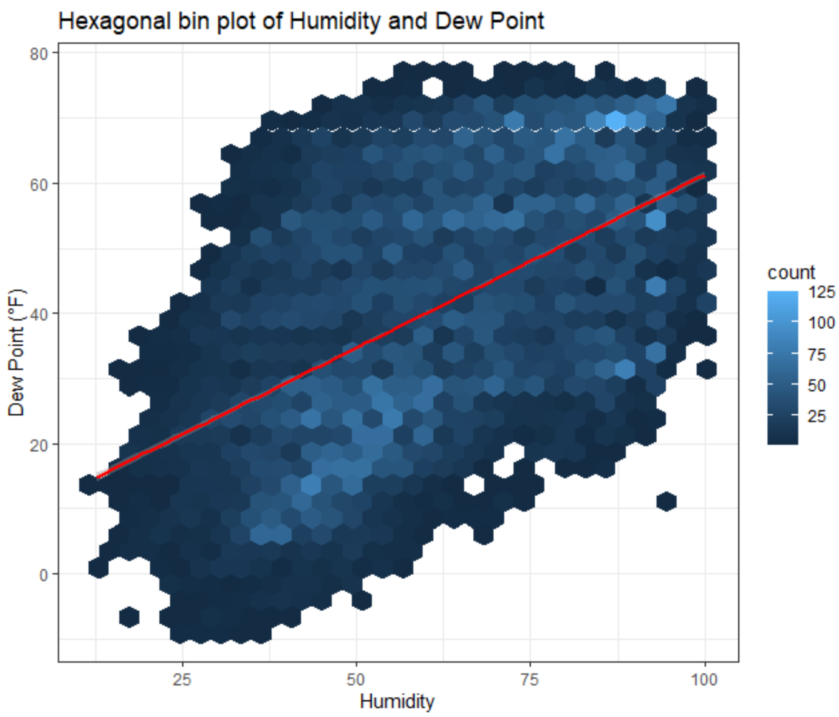

Figure 18: Hexagonal bin plot of humidity and dew point

The main purpose of plotting a hexagonal bin plot to divide the coordinate plane of the variables into 2d bins and display the frequency of each bin by filling colour so the problem of overplot can be solved due to the large size of dataset (Grolemund and Wickham, 2017). By plotting hexagonal plot, it can used to identify the relationship between the two variables. Based on the plot above, it shows that there is a linear positive correlation between the two variables. However, the strength of the relationship is moderate as the correlation coefficient is just 0.53. When humidity increase, dew point will increase as well. Lastly, the bin which has the highest frequency is (x = 87.26, y = 69.57).

Conclusion

For analysing the hourly weather dataset, my knowledge on R programming had been heightened especially on data visualization, data exploration and data manipulation.

Although the analysis has been done, but there is limitation for the analysis as well. The limitation is the method used for replacing the missing value, mean imputation. Mean imputation will lead to bias in multivariate estimation like regression coefficients.

For improving the quality of the analysis, I will recommend to use other imputation method rather than using mean imputation. For instance, using MICE (Multivariate Imputation via Chained Equations), Hmisc (Vidhya, 2016) and so on.

Reference

Frost, J., 2020. *Guidelines For Removing And Handling Outliers In Data - Statistics By Jim*. [online] Statistics By Jim. Available at: https://statisticsbyjim.com/basics/remove-outliers/ [Accessed 17 August 2020].

Grolemund, G. and Wickham, H., 2017. *R For Data Science*. 1st ed. [ebook] O'Reilly. Available at: https://r4ds.had.co.nz/index.html [Accessed 17 August 2020].

Hadley, W., Winston, C., Lionel, H., Thomas, L. and Claus, W., n.d. *Complete Themes — Ggtheme*. [online] Ggplot2.tidyverse.org. Available at: https://ggplot2.tidyverse.org/reference/ggtheme.html [Accessed 22 August 2020].

Support.minitab.com. 2019. *Interpret The Key Results For Bar Chart - Minitab Express*. [online] Available at: https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/graphs/bar-chart/interpret-the-results/interpret-the-results/ [Accessed 17 August 2020].

Vidhya, A., 2016. *R Packages | Impute Missing Values In R*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/ [Accessed 19 August 2020].

Viswa, V. and Shanthi, V., 2015. *R Data Analysis Cookbook*. 1st ed. [ebook] Available at: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783989065/1/ch01lvl1sec20/binning-numerical-data [Accessed 17 August 2020].

Weather.gov. n.d. *NWS Jetstream MAX - Beaufort Wind Force Scale*. [online] Available at: https://www.weather.gov/jetstream/beaufort_max [Accessed 18 August 2020].