A · P · U

ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

INDIVIDUAL ASSIGNMENT

CT127-3-2-PFDA

NAME: LOKE WENG KHAY

TP NO:

SUBJECT: PROGRAMMING FOR DATA ANALYSIS

DATE ASSIGNED: 28-05-21

Table of Contents

Introduction

R is a language and environment for statistical computing and graphics purposes. R was created by John Chambers and his colleagues at Bell Laboratories (R-Project, n.d.).

The weather forecast has been vital information to have as it plays a crucial role in determining the weather condition, visibility, and many more factors. Weather information is essential for people like aircraft pilots, farmer, constructor builder and many more. This is why weather information is crucial, as without having good knowledge of the weather, it can cause accidents or deaths.

In this assignment, we are given a dataset regarding the United States weather. This dataset contains 22 columns and 366 rows and data related to Temperature, Wind Speed, Rainfall size, and much more information related to the weather data. This assignment aims to provide valuable analysis to provide justification and good recommendations. The dataset will be used to generate helpful analysis. In this report, we will be creating five questions and answering each question based on the analysis generated.

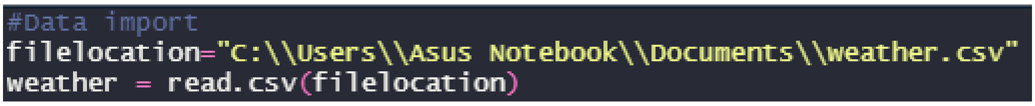
Assumption

The weather has played a crucial role in our life. This is because the climate controls the distribution of rainwater on earth. This is very important as all living organisms on earth use water to survive. This is because severe weather conditions such as hurricanes, tornadoes and floods have a significant impact on human civilisations. A prominent example of this was Hurricane Katrina, which hit a massive area of the United States Gulf Coast, killing nearly 2,000 people and displacing more than a million people from their homes (Anon., 20114). This is why knowing the weather enables people to predict and prepare for the worst weather conditions.

Based on the dataset given, we can assume that all 366 data were collected on the same day and same time but only in different location in the United States of America.

Data Import

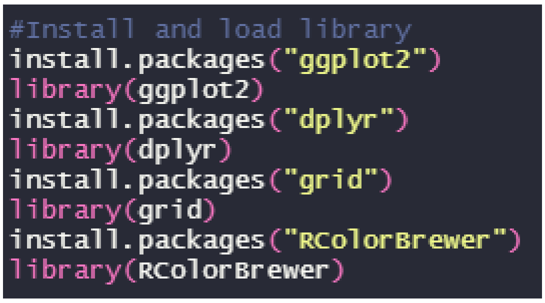
The first step is to import the dataset into RStudio. The read.csv() function will be used to import all the data from the “weather.csv” file.



```
#Data import
filelocation="C:\\Users\\Asus Notebook\\Documents\\weather.csv"
weather = read.csv(filelocation)
```

Figure 1: R code to import the U.S. Weather Dataset

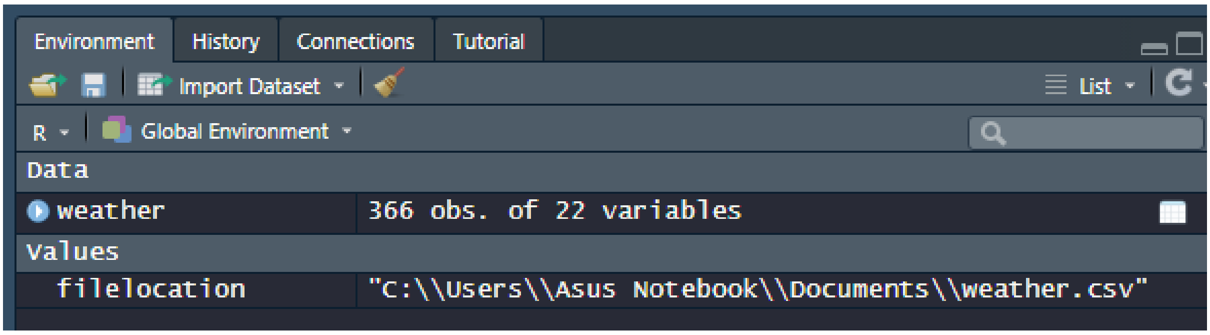
Before we can load the library, we will have to install the packages by using install.packages() function to install it into the RStudio. The library function is used to load the library before it can be used. “ggplot2” is a package used to create elegant data visualisations like a line graph, histogram, and many more graph options. Package “dplyr” is used to allow more accessible examination for the dataset in RStudio. Package “grid” is used to resize and rearrange the graphs using any of the available coordinate systems. Package “RColorBrewer” is used to create nice looking colour palettes, especially for thematic maps.



```
#Install and load library
install.packages("ggplot2")
library(ggplot2)
install.packages("dplyr")
library(dplyr)
install.packages("grid")
library(grid)
install.packages("RColorBrewer")
library(RColorBrewer)
```

Figure 2: R code to install and load library

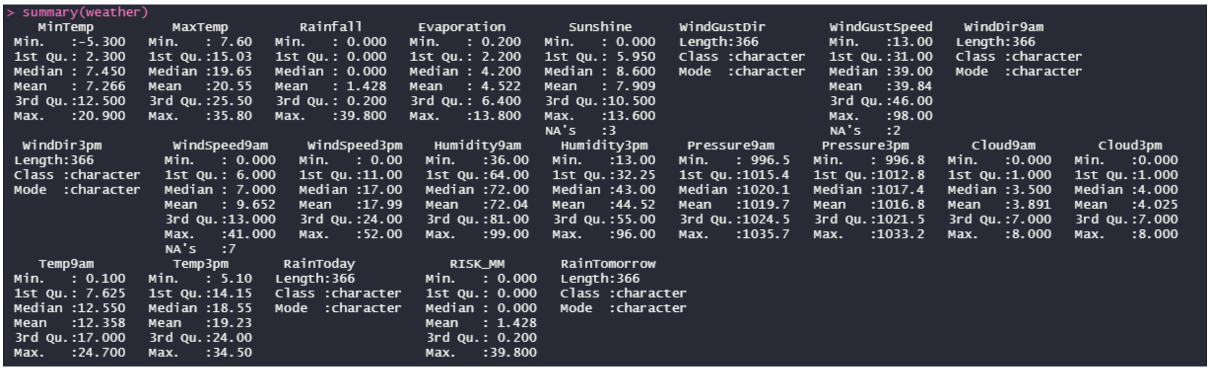
Once the code is executed, you can see in figure 3 that the data from the dataset is saved in RStudio Environment.



```
Data
weather        366 obs. of 22 variables
Values
filelocation   "C:\\Users\\Asus Notebook\\Documents\\weather.csv"
```

Figure 3: After Compilation of R code

To view the data, we can use the R code “summary(weather)” to generate all the necessary information of all variables in the dataset. Below is the code used and the expected output from the R code.

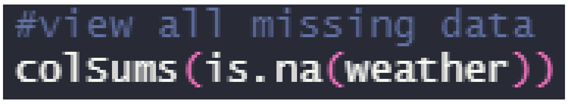


```
> summary(weather)
    MinTemp          MaxTemp         Rainfall        Evaporation        Sunshine       WindGustDir        WindGustSpeed    WindDir9am       
 Min.   :-5.300   Min.   : 7.60   Min.   : 0.000   Min.   : 0.200   Min.   : 0.000   Length:366         Min.   :13.00   Length:366        
 1st Qu.: 2.300   1st Qu.:15.03   1st Qu.: 0.000   1st Qu.: 2.200   1st Qu.: 5.950   Class :character   1st Qu.:31.00   Class :character  
 Median : 7.450   Median :19.65   Median : 0.000   Median : 4.200   Median : 8.600   Mode  :character   Median :39.00   Mode  :character  
 Mean   : 7.266   Mean   :20.55   Mean   : 1.428   Mean   : 4.522   Mean   : 7.909                      Mean   :39.84                     
 3rd Qu.:12.500   3rd Qu.:25.50   3rd Qu.: 0.200   3rd Qu.: 6.400   3rd Qu.:10.500                      3rd Qu.:46.00                     
 Max.   :20.900   Max.   :35.80   Max.   :39.800   Max.   :13.800   Max.   :13.600                      Max.   :98.00                     
                                                                    NA's   :3                           NA's   :2                         
  WindDir3pm         WindSpeed9am     WindSpeed3pm    Humidity9am     Humidity3pm     Pressure9am      Pressure3pm        Cloud9am        Cloud3pm    
 Length:366         Min.   : 0.000   Min.   : 0.00   Min.   :36.00   Min.   :13.00   Min.   : 996.5   Min.   : 996.8   Min.   :0.000   Min.   :0.000  
 Class :character   1st Qu.: 6.000   1st Qu.:11.00   1st Qu.:64.00   1st Qu.:32.25   1st Qu.:1015.4   1st Qu.:1012.8   1st Qu.:1.000   1st Qu.:1.000  
 Mode  :character   Median : 7.000   Median :17.00   Median :72.00   Median :43.00   Median :1020.1   Median :1017.4   Median :3.500   Median :4.000  
                    Mean   : 9.652   Mean   :17.99   Mean   :72.04   Mean   :44.52   Mean   :1019.7   Mean   :1016.8   Mean   :3.891   Mean   :4.025  
                    3rd Qu.:13.000   3rd Qu.:24.00   3rd Qu.:81.00   3rd Qu.:55.00   3rd Qu.:1024.5   3rd Qu.:1021.5   3rd Qu.:7.000   3rd Qu.:7.000  
                    Max.   :41.000   Max.   :52.00   Max.   :99.00   Max.   :96.00   Max.   :1035.7   Max.   :1033.2   Max.   :8.000   Max.   :8.000  
                    NA's   :7                                                                                                                         
    Temp9am          Temp3pm       RainToday            RISK_MM        RainTomorrow      
 Min.   : 0.100   Min.   : 5.10   Length:366         Min.   : 0.000   Length:366        
 1st Qu.: 7.625   1st Qu.:14.15   Class :character   1st Qu.: 0.000   Class :character  
 Median :12.550   Median :18.55   Mode  :character   Median : 0.000   Mode  :character  
 Mean   :12.358   Mean   :19.23                      Mean   : 1.428                     
 3rd Qu.:17.000   3rd Qu.:24.00                      3rd Qu.: 0.200                     
 Max.   :24.700   Max.   :34.50                      Max.   :39.800                     
```

Figure 4: Summary of the Weather Dataset

Data Pre-Processing

View all missing data

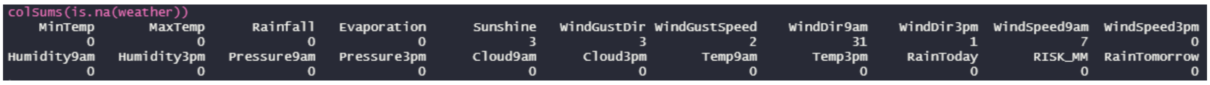


```r
#view all missing data
colSums(is.na(weather))
```

Figure 5: R code to view all missing data in the dataset

Before any data cleaning can be done, we must view all the missing data present in each column in the dataset. The figure below represents the number of missing data for each attribute in the weather dataset.

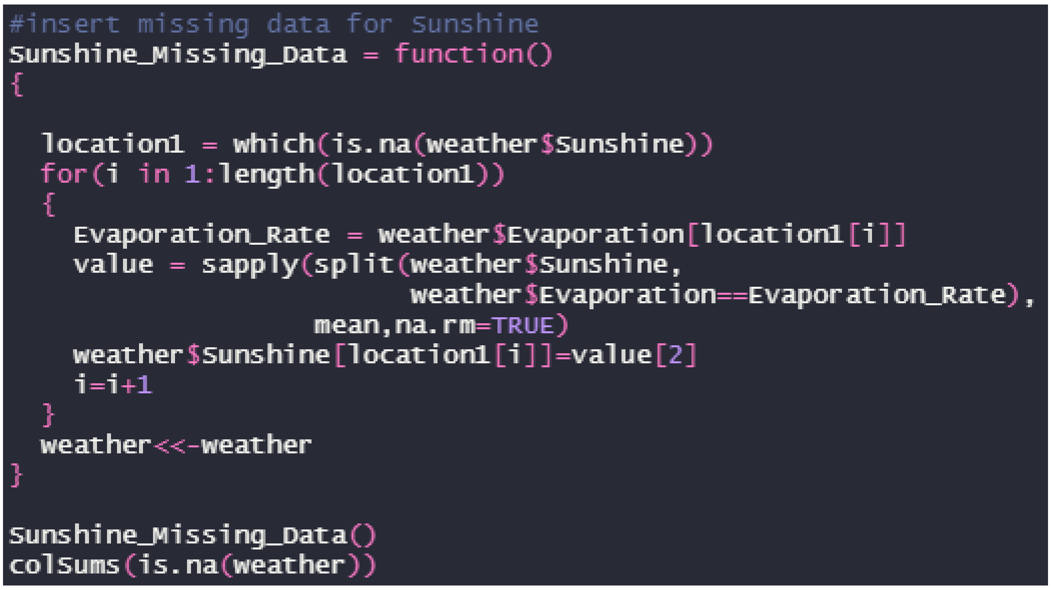
Output



```
colSums(is.na(weather))
      MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm 
            0             0             0             0             3             3             2            31             1             7             0 
  Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow 
            0             0             0             0             0             0             0             0             0             0             0 
```

Figure 6: Output of all missing data

Treat Missing data for variable Sunshine

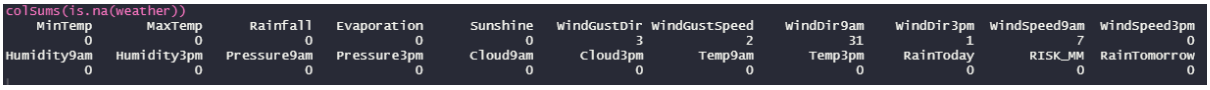


```r
#insert missing data for Sunshine
Sunshine_Missing_Data = function()
{

  location1 = which(is.na(weather$Sunshine))
  for(i in 1:length(location1))
  {
    Evaporation_Rate = weather$Evaporation[location1[i]]
    value = sapply(split(weather$Sunshine,
                         weather$Evaporation==Evaporation_Rate),
                   mean,na.rm=TRUE)
    weather$Sunshine[location1[i]]=value[2]
    i=i+1
  }
  weather<<-weather
}

Sunshine_Missing_Data()
colSums(is.na(weather))
```

Figure 7: R code to insert value to the missing data in variable “Sunshine”

Based on figure 7, we will use multiple imputations to treat the variable “Sunshine” missing data. It will refer to data from variable “Evaporation” then insert the mean value for variable “Sunshine” based on variable “Evaporation”. Once the function code is executed, all missing data for the variable “Sunshine” will be treated.

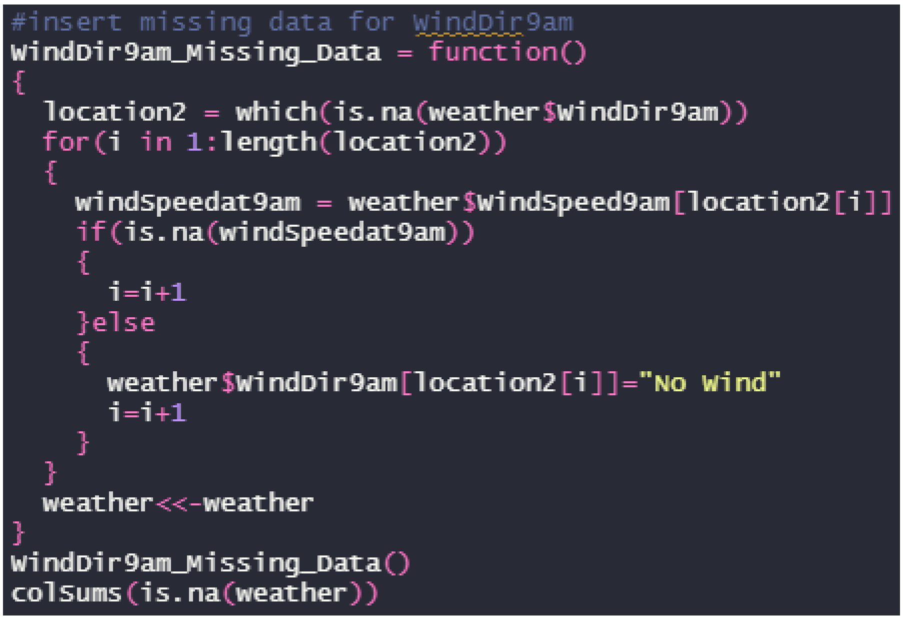
Output



```
colSums(is.na(weather))
      MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm
            0             0             0             0             0             3             2            31             1             7             0
  Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow
            0             0             0             0             0             0             0             0             0             0             0
```

Figure 8: Output of all missing data after treating variable “Sunshine”

Treat Missing data for variable WindDir9am

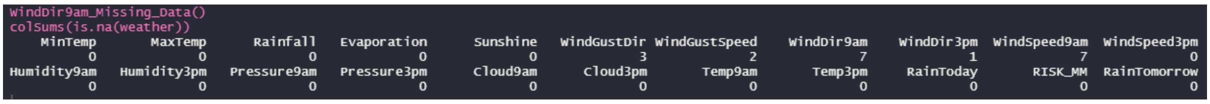


```
#insert missing data for WindDir9am
WindDir9am_Missing_Data = function()
{
  location2 = which(is.na(weather$WindDir9am))
  for(i in 1:length(location2))
  {
    windSpeedat9am = weather$WindSpeed9am[location2[i]]
    if(is.na(windSpeedat9am))
    {
      i=i+1
    }else
    {
      weather$WindDir9am[location2[i]]="No Wind"
      i=i+1
    }
  }
  weather<<-weather
}
WindDir9am_Missing_Data()
colSums(is.na(weather))
```

Figure 9: R code to insert value to missing data in variable "WindDir9am"

Based on figure 9, we will use multiple imputations to treat the variable "WindDir9am" missing data. It will refer to data from the variable "WindSpeed9am" then treat the missing data for the variable "WindDir9am". This is because some of the data in the variable "WindSpeed9am" have a value of "0". This can indicate that there is no wind activity at this hour. This is the reason why some value in the variable "WindDir9am" is "na". Once this function code is executed, some missing data with the value of "0" in variable "WindSpeed9am" will have data inserted into variable "WindDir9am" as "No Wind".

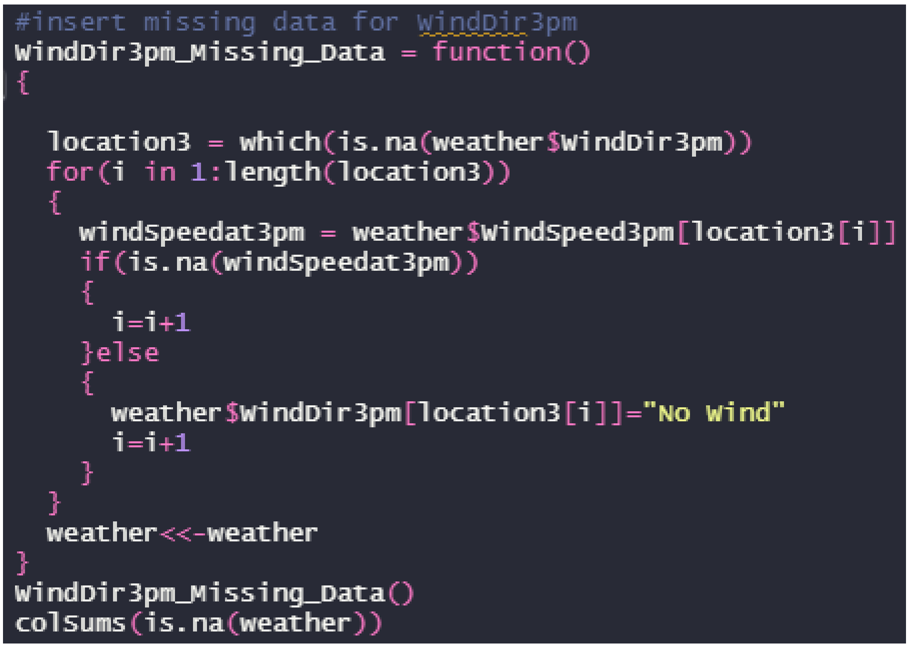
Output



```
WindDir9am_Missing_Data()
colSums(is.na(weather))
      MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm 
            0             0             0             0             0             3             2             7             1             7             0 
  Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow 
            0             0             0             0             0             0             0             0             0             0             0 
```

Figure 10: Output of all missing data after treating variable "WindDir9am"

Treat Missing data for variable WindDir3pm

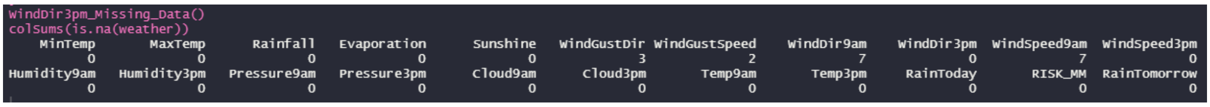


```r
#insert missing data for WindDir3pm
WindDir3pm_Missing_Data = function()
{

  location3 = which(is.na(weather$WindDir3pm))
  for(i in 1:length(location3))
  {
    windSpeedat3pm = weather$WindSpeed3pm[location3[i]]
    if(is.na(windSpeedat3pm))
    {
      i=i+1
    }else
    {
      weather$WindDir3pm[location3[i]]="No Wind"
      i=i+1
    }
  }
  weather<<-weather
}
WindDir3pm_Missing_Data()
colSums(is.na(weather))
```

Figure 11: R code to insert value to missing data for variable “WindDir3pm”

Based on figure 11, we will use multiple imputations to treat the variable “WindDir3pm” missing data. It will refer to data from the variable “WindSpeed3pm” then treat the missing data for the variable “WindDir3pm”. This is because some of the data in the variable “WindSpeed3pm” have a value of “0”; this can indicate that there is no wind activity at this hour. This is the reason why some value in the variable “WindDir3pm” is “na”. Once this function code is executed, some missing data with the value of “0” in variable “WindSpeed3pm” will have data inserted into variable “WindDir3pm” as “No Wind”.

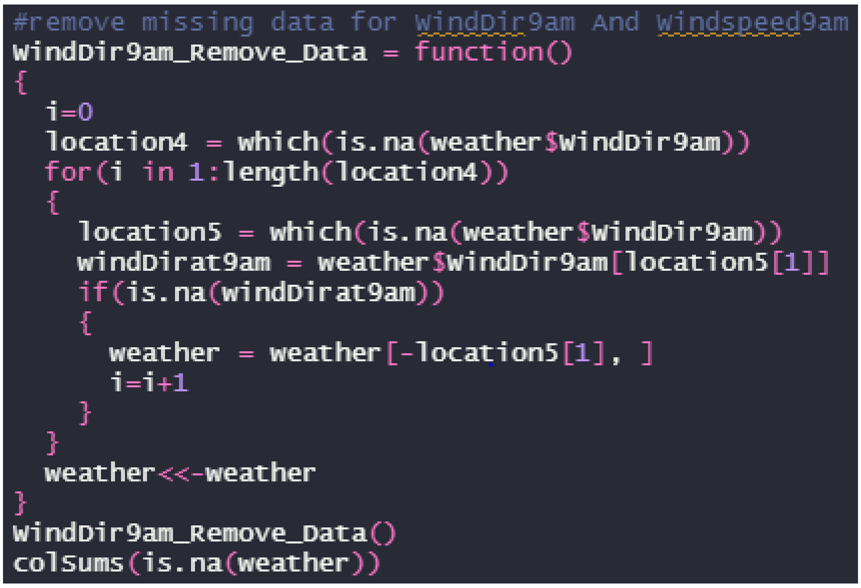
Output



```
WindDir3pm_Missing_Data()
colSums(is.na(weather))
      MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm 
            0             0             0             0             0             3             2             7             0             7             0 
  Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow 
            0             0             0             0             0             0             0             0             0             0             0 
```

Figure 12: Output of all missing data after treating variable “WindDir3pm”

Remove Missing data for variable WindDir9am and WindSpeed9am

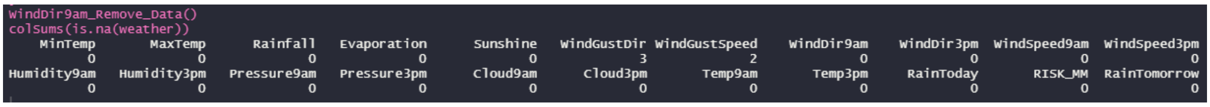


```
#remove missing data for WindDir9am And Windspeed9am
WindDir9am_Remove_Data = function()
{
  i=0
  location4 = which(is.na(weather$WindDir9am))
  for(i in 1:length(location4))
  {
    location5 = which(is.na(weather$WindDir9am))
    windDirat9am = weather$WindDir9am[location5[1]]
    if(is.na(windDirat9am))
    {
      weather = weather[-location5[1], ]
      i=i+1
    }
  }
  weather<<-weather
}
WindDir9am_Remove_Data()
colSums(is.na(weather))
```

Figure 13: Remove missing data that are not useful in variable “WindDir9am” and “WindSpeed9am”

In figure 13, after treating some missing data for the variable “WindDir9am” in figure 9, the rest of the missing data cannot be treated. This is because both variable “WindDir9am” and variable “WindSpeed9am” has the value “na” in them. With this, we will remove all the seven rows of data that are not useful for the analysis.

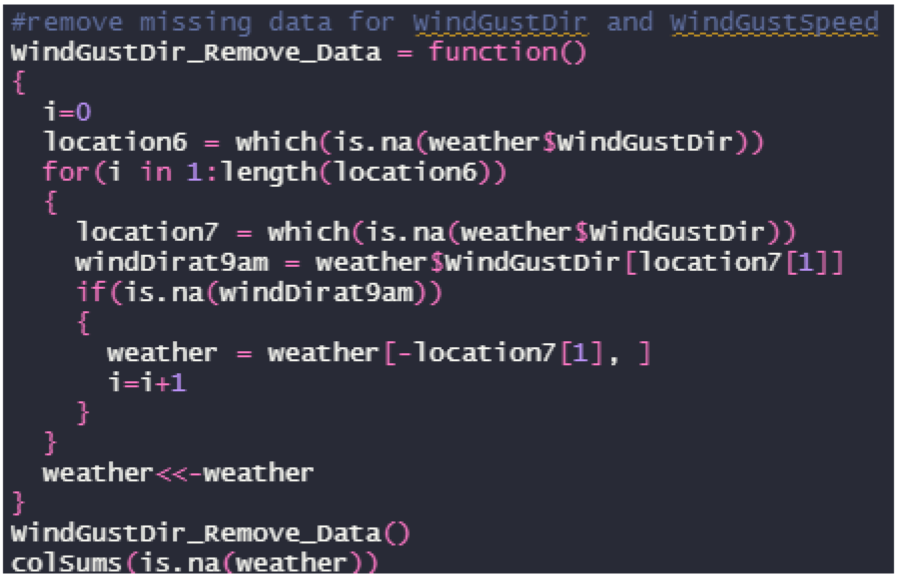
Output



```
WindDir9am_Remove_Data()
colSums(is.na(weather))
      MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm
            0             0             0             0             0             3             2             0             0             0             0
  Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow
            0             0             0             0             0             0             0             0             0             0             0
```

Figure 14: Output of all missing data after removing variable “WindDir9am” and “WindSpeed9am”

Remove Missing data for variable WindGustDir and WindGustSpeed

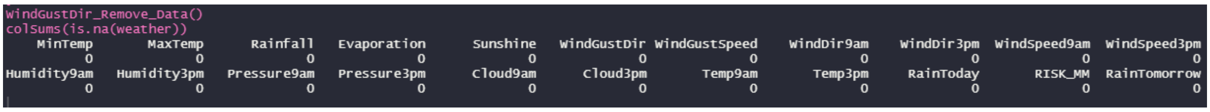


```
#remove missing data for WindGustDir and WindGustSpeed
WindGustDir_Remove_Data = function()
{
  i=0
  location6 = which(is.na(weather$WindGustDir))
  for(i in 1:length(location6))
  {
    location7 = which(is.na(weather$WindGustDir))
    windDirat9am = weather$WindGustDir[location7[1]]
    if(is.na(windDirat9am))
    {
      weather = weather[-location7[1], ]
      i=i+1
    }
  }
  weather<<-weather
}
WindGustDir_Remove_Data()
colSums(is.na(weather))
```

Figure 15: Remove missing data that are not useful in attribute “WindGustDir” and “WindGustSpeed”

In figure 15, for variable “WindGustDir” and variable “WindGustSpeed”, both of them have the value “na” in it. With this, we will remove all the three rows of data that are not useful for the analysis.

Output



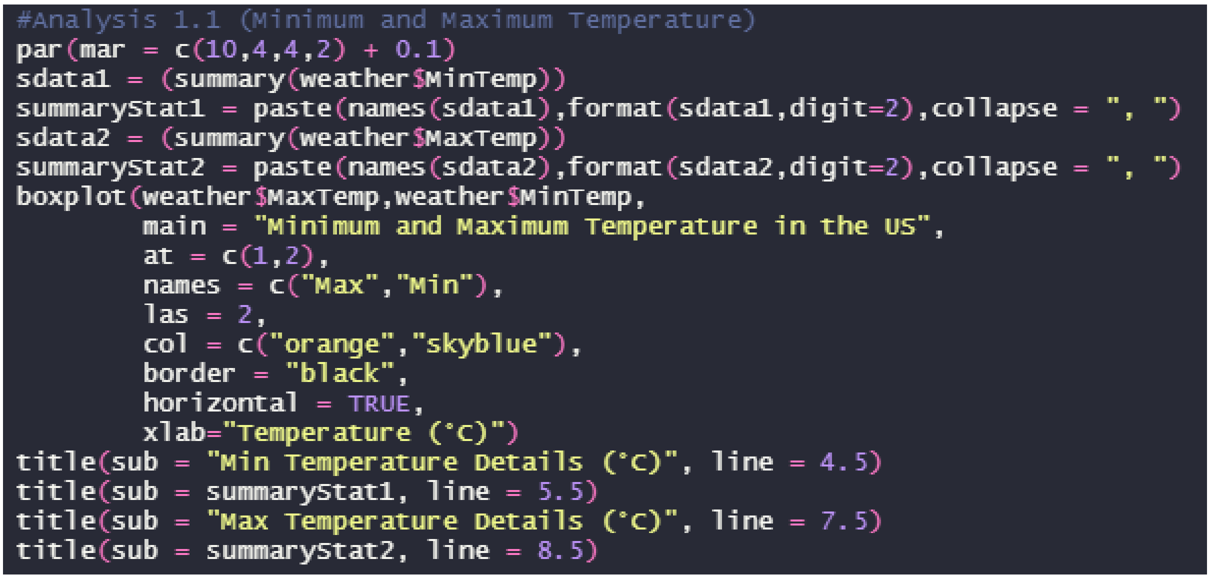
```
WindGustDir_Remove_Data()
colSums(is.na(weather))
    MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm 
          0             0             0             0             0             0             0             0             0             0             0 
Humidity9am   Humidity3pm   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm     RainToday       RISK_MM  RainTomorrow 
          0             0             0             0             0             0             0             0             0             0             0 
```

Figure 16: Output of all missing data after removing variable “WindGustDir” and “WindGustSpeed”

Question 1: Is the United States currently an excellent time to travel to?

This is crucial as it allows the tourist to plan their visit to the United States. This information is important as it enables the tourist to understand the current season, temperature, etc. when planning to visit places in the United States.

Analysis 1.1: Find the Season in the United States of America?



```
#Analysis 1.1 (Minimum and Maximum Temperature)
par(mar = c(10,4,4,2) + 0.1)
sdata1 = (summary(weather$MinTemp))
summaryStat1 = paste(names(sdata1),format(sdata1,digit=2),collapse = ", ")
sdata2 = (summary(weather$MaxTemp))
summaryStat2 = paste(names(sdata2),format(sdata2,digit=2),collapse = ", ")
boxplot(weather$MaxTemp,weather$MinTemp,
        main = "Minimum and Maximum Temperature in the US",
        at = c(1,2),
        names = c("Max","Min"),
        las = 2,
        col = c("orange","skyblue"),
        border = "black",
        horizontal = TRUE,
        xlab="Temperature (°C)")
title(sub = "Min Temperature Details (°C)", line = 4.5)
title(sub = summaryStat1, line = 5.5)
title(sub = "Max Temperature Details (°C)", line = 7.5)
title(sub = summaryStat2, line = 8.5)
```

Figure 17: R code for Minimum and Maximum Temperature of the United States in Box Plot

Based on figure 17, it is the R code to create a box plot displaying the minimum and maximum temperature in the United States. Using a box plot is a suitable way to provide a visual summary of the data—Example, range, mean and median of temperatures.

Based on figure 17, the par() function is used to set or query graphical parameters. It will allow space for the details that are listed below the graph. Variable “sdata1” and variable “sdata2” are used to store a summary of variable “MinTemp” and variable “MaxTemp” using the summary() function. The variable “summaryStat1” and variable “summaryStat2” will contain the reformatted data that will display the data in one line. The boxplot() function is used to create the box plot with the title name, x-axis name, y-axis name, colour and many more. Last but not least, the title() function is used to display the summary data below the graph.

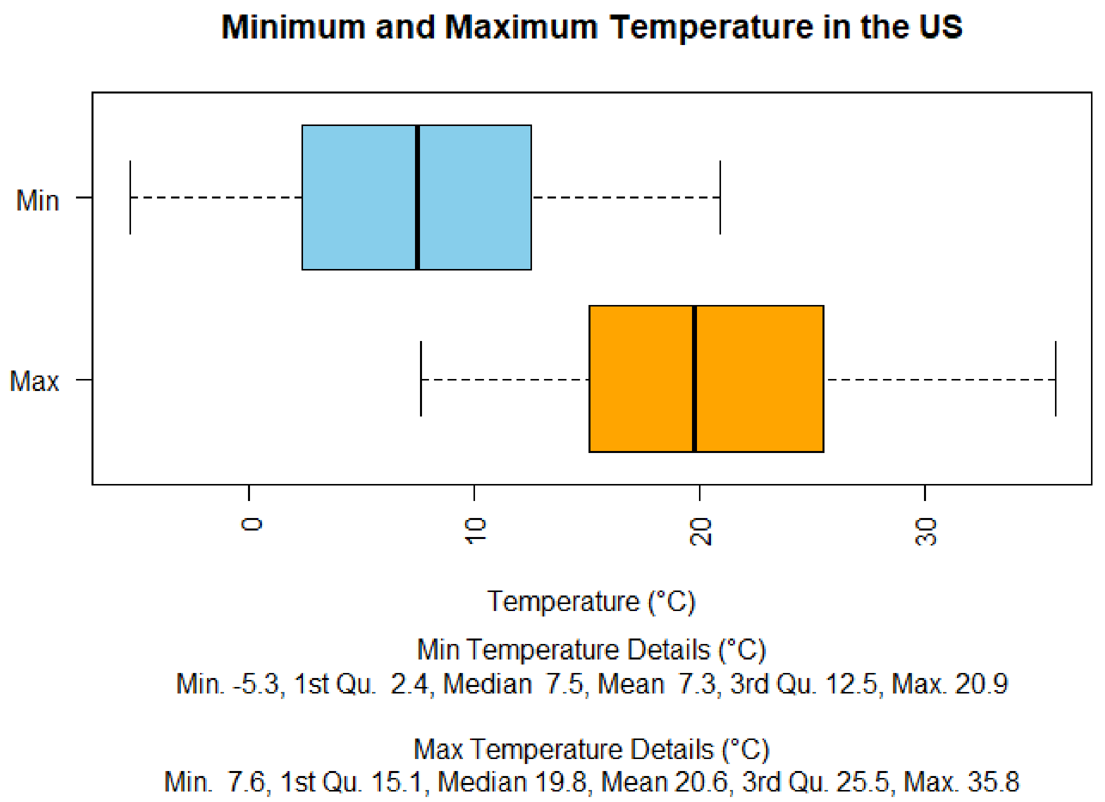
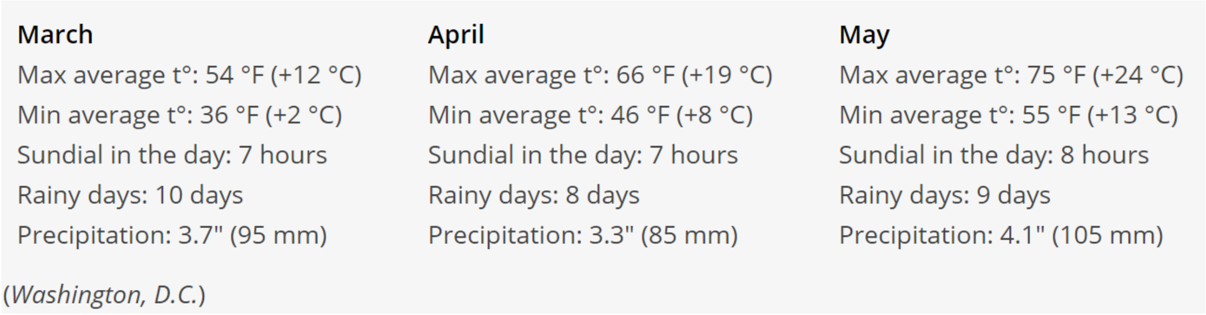


Figure 18: Box Plot – Minimum and Maximum Temperature in the United States

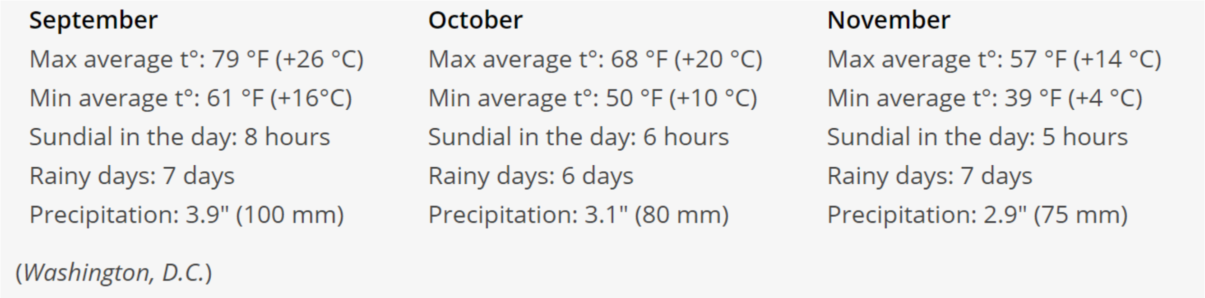
Based on the figure analysis above, it is representing the minimum and maximum temperature in the United States of America. The minimum temperature is currently in the range of 2.4 °C to 12.5 °C. The median minimum temperature is 7.5 °C, and the mean minimum temperature is 7.3 °C. While for the maximum temperature, it is currently in the range of 15.1 °C to 25.5 °C. The median maximum temperature is 19.8 °C, and the mean maximum temperature is 20.6 °C.



March	April	May
Max average t°: 54 °F (+12 °C)	Max average t°: 66 °F (+19 °C)	Max average t°: 75 °F (+24 °C)
Min average t°: 36 °F (+2 °C)	Min average t°: 46 °F (+8 °C)	Min average t°: 55 °F (+13 °C)
Sundial in the day: 7 hours	Sundial in the day: 7 hours	Sundial in the day: 8 hours
Rainy days: 10 days	Rainy days: 8 days	Rainy days: 9 days
Precipitation: 3.7" (95 mm)	Precipitation: 3.3" (85 mm)	Precipitation: 4.1" (105 mm)

(*Washington, D.C.*)

Figure 19: Spring Season Temperatures (Anon., n.d.)



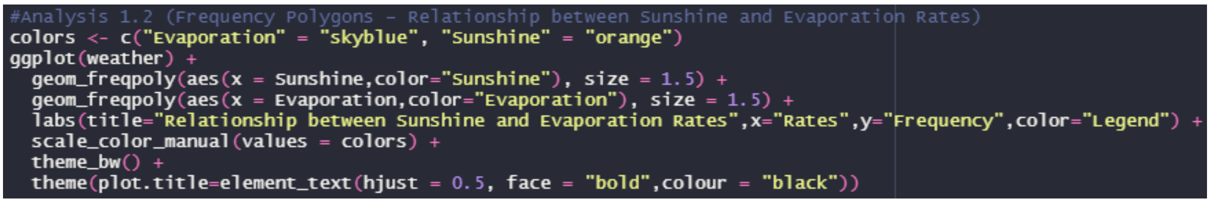
September	October	November
Max average t°: 79 °F (+26 °C)	Max average t°: 68 °F (+20 °C)	Max average t°: 57 °F (+14 °C)
Min average t°: 61 °F (+16°C)	Min average t°: 50 °F (+10 °C)	Min average t°: 39 °F (+4 °C)
Sundial in the day: 8 hours	Sundial in the day: 6 hours	Sundial in the day: 5 hours
Rainy days: 7 days	Rainy days: 6 days	Rainy days: 7 days
Precipitation: 3.9" (100 mm)	Precipitation: 3.1" (80 mm)	Precipitation: 2.9" (75 mm)

(*Washington, D.C.*)

Figure 20: Autumn Season Temperatures (Anon., n.d.)

Figure 19 and figure 20 represent the average minimum and maximum temperature by months for the Spring and Autumn season from an article (Anon., n.d.). With the figure above, we can predict that the current season in the United States is currently either in the Autumn Season or Spring Season. This is because the difference between the mean minimum temperature and the mean maximum temperature in the dataset and the figure above has a difference of 1 °C to 3 °C for the average minimum temperature and a difference of 0 °C - 1 °C for the average maximum temperature. The tourists are expected to bring extra clothes, because the temperature can drop below 10 °C. In conclusion, it is considered the best time to travel to the United States as it is currently not too cold like in the winter and not too hot like in the summer.

Analysis 1.2: Find the amount of sunshine and evaporation rates?



```r
#Analysis 1.2 (Frequency Polygons - Relationship between Sunshine and Evaporation Rates)
colors <- c("Evaporation" = "skyblue", "Sunshine" = "orange")
ggplot(weather) +
  geom_freqpoly(aes(x = Sunshine,color="Sunshine"), size = 1.5) +
  geom_freqpoly(aes(x = Evaporation,color="Evaporation"), size = 1.5) +
  labs(title="Relationship between Sunshine and Evaporation Rates",x="Rates",y="Frequency",color="Legend") +
  scale_color_manual(values = colors) +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 21: R code for Relationship between Sunshine and Evaporation Rates in Frequency Polygons

Based on figure 21, it is the R code to create a frequency polygon displaying the relationship between sunshine and evaporation Rates. The reason for using frequency polygon is because frequency polygons can distribute cumulative frequency in order.

Based on figure 21, the variable “colors” declares the colour type before it is used in the ggplot() function. The ggplot() function describes the dataset used. Furthermore, geom_freqpoly() is used to indicate the type of graph being used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. Moreover, the scale_color_manual() function is used to allow you to specify your own set of mappings from levels in the data to aesthetic values. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

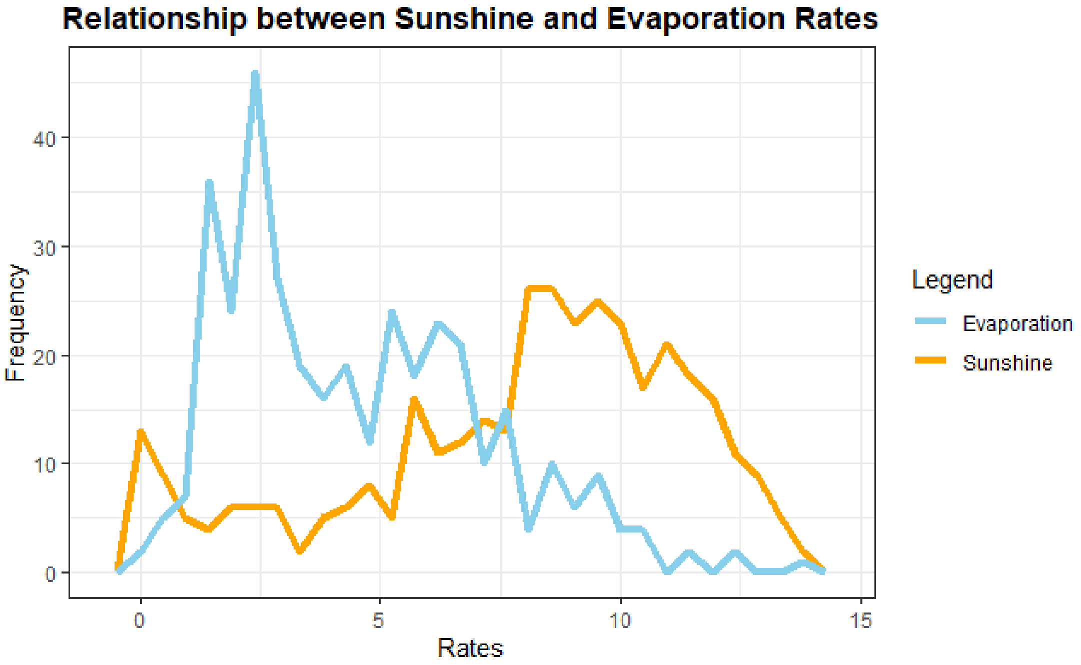
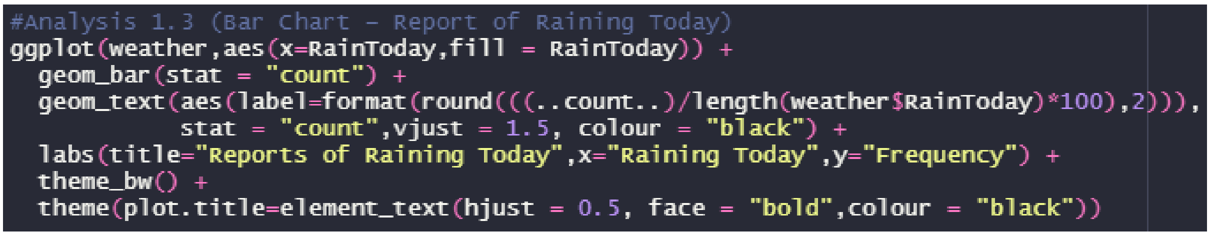


Figure 22: Frequency Polygons – Relationship between Sunshine and Evaporation Rates

Based on figure 22 represents the Relationship between Sunshine and Evaporation Rates in frequency polygons. From the figure, we can indicate that the more sunshine exposure, the less the evaporation rate. The only viable assumption is that the concept of more sunshine exposure increase evaporation rates may not be 100% true. Based on a study by Xiongjiang Yu and Jinliang Xu, they found that due to competing effects, the presence of sunlight hinders evaporation in the initial stage (Bandari, 2020). The tourist are advised to bring lotions and wear a cap when exploring the city. This is to avoid sunburns when exploring the city. In conclusion, it is still an excellent time to travel to the United States as the weather is suitable and there is a proper amount of sunshine exposure for good photographs.

Analysis 1.3: Find the number of reports of raining today?



```
#Analysis 1.3 (Bar Chart - Report of Raining Today)
ggplot(weather,aes(x=RainToday,fill = RainToday)) +
  geom_bar(stat = "count") +
  geom_text(aes(label=format(round(((..count..)/length(weather$RainToday)*100),2))),
            stat = "count",vjust = 1.5, colour = "black") +
  labs(title="Reports of Raining Today",x="Raining Today",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 23: R code for bar chart – Report of Raining Today

Based on figure 23, it is the R code to create a bar chart displaying the number of reports of raining today. The reason for using bar graphs is because they are an effective way to compare items between different groups.

Based on figure 23, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_bar() is used to indicate the type of graph being used. The geom_text() function is used to insert text to the bar chart graph. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

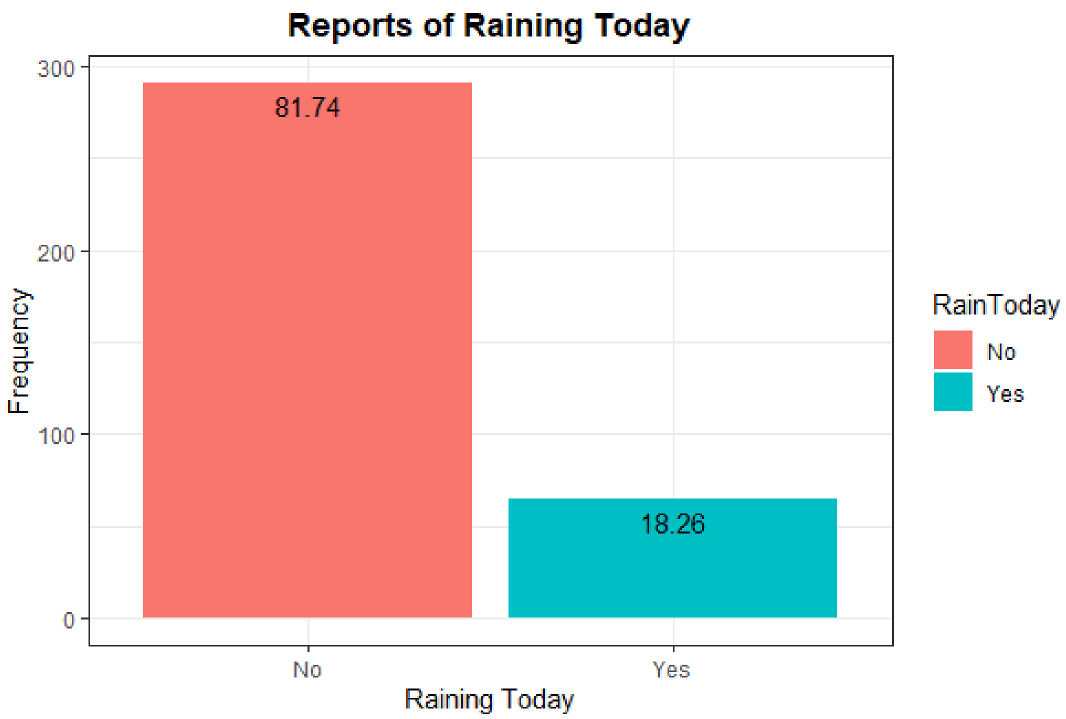


Figure 24: Bar Chart – Report of Raining Today

Based on figure 24 represents the Reports of Raining Today in a bar chart. From the figure, we can indicate that the possibility of raining today is 18% which is relatively low as almost 81% of the report states that it will not rain today. The only viable assumption that there is currently a lack of cloud formation in the area (Doyle, 2021). The other logical assumption is that the weather in the United States is presently windy or warm, with a low probability of rain. In conclusion, although the possibility of rain is low, tourist is advised to bring an umbrella, raincoat and extra clothes just in case it rains.

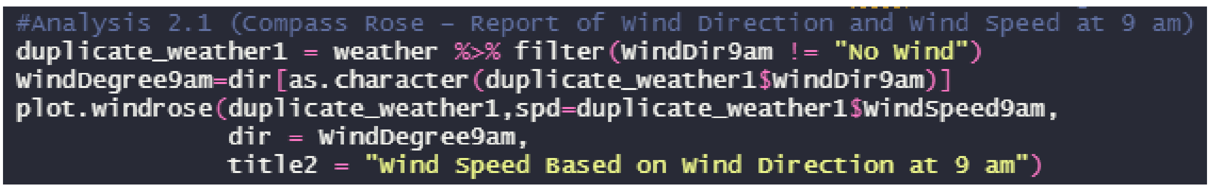
Conclusion

Based on question 1, we can conclude that it is an excellent time to visit the United States. This is because it is currently either in the Autumn or Spring Season in the United States, representing the best month to travel to us because it is not too hot like in the summer or not too cold like in the winter, based on analysis 1.1. Furthermore, tourist can also take beautiful pictures as the weather in the US is not too cloudy and have a high amount of sunshine with low probability or raining, based on analysis 1.2 and analysis 1.3.

Question 2: Is 9 am an excellent time to have a Formula One United States Grand Prix?

The question is crucial as it describes how windy the circuit will be as it affects the downforce of the Formula One car (Noble, 2017). It also helps to understand the surrounding temperature as it allows to predict the use of the correct type of tyre compound in the circuit (Anon., 2018). This information enables the teams to plan their strategy before or during the race, which will affect the win or loss of the Formula One United States Grand Prix.

Analysis 2.1: Find the wind direction and wind speed at 9 am?



```
#Analysis 2.1 (Compass Rose - Report of Wind Direction and Wind Speed at 9 am)
duplicate_weather1 = weather %>% filter(WindDir9am != "No Wind")
WindDegree9am=dir[as.character(duplicate_weather1$WindDir9am)]
plot.windrose(duplicate_weather1,spd=duplicate_weather1$WindSpeed9am,
              dir = WindDegree9am,
              title2 = "Wind Speed Based on Wind Direction at 9 am")
```

Figure 25: R code for compass rose graph – Report on Wind Direction and Wind Speed at 9 am

Based on figure 25, it is the R code to create a compass rose displaying the Wind Direction and Wind Speed at 9 am. The reason for using compass rose is that they can report the wind direction and wind speed simultaneously. The compass rose is like a radar, and it provides helpful information to provide an analysis.

Based on figure 25, the variable “duplicate_weather1” contains a copy of the dataset weather, but it is filtered not to contain any “No Wind” data in variable “WindDir9am” by using the piping method. Variable “WindDegree9am” will contain the converted data from wind direction to wind degree. The “plot.windrose()” function will create the compass rose graph accordingly.

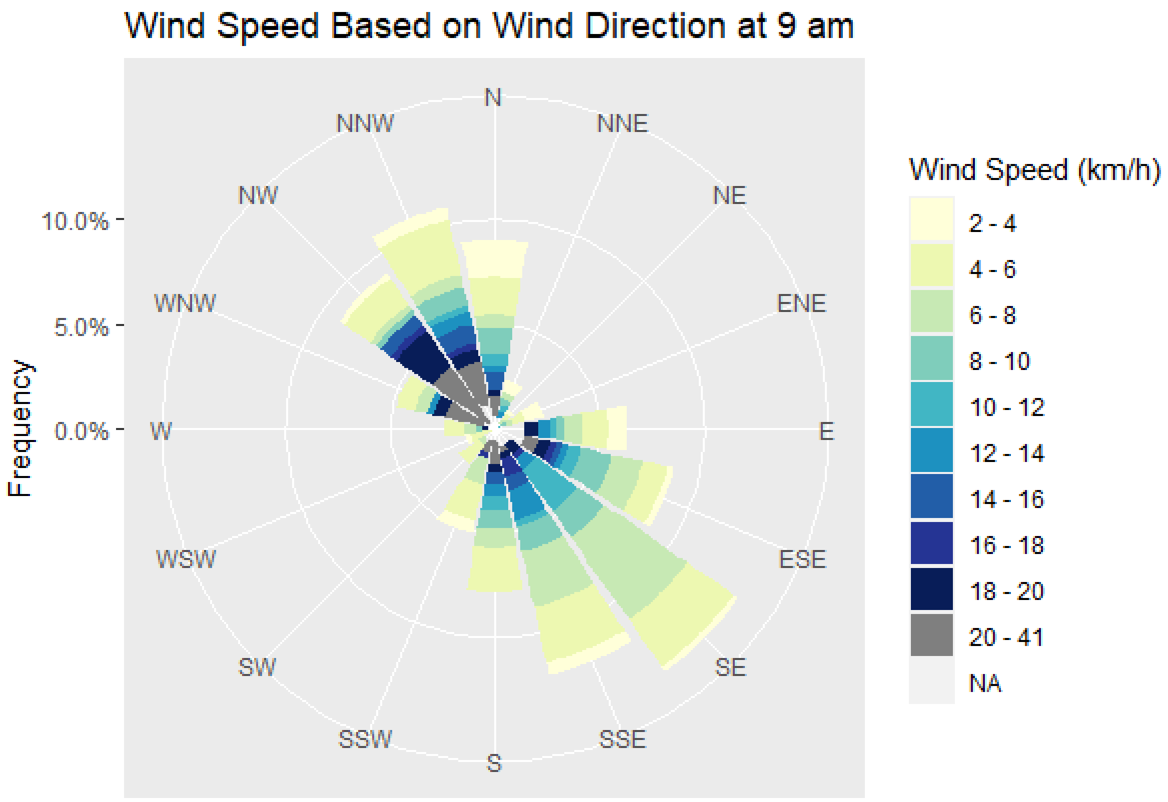
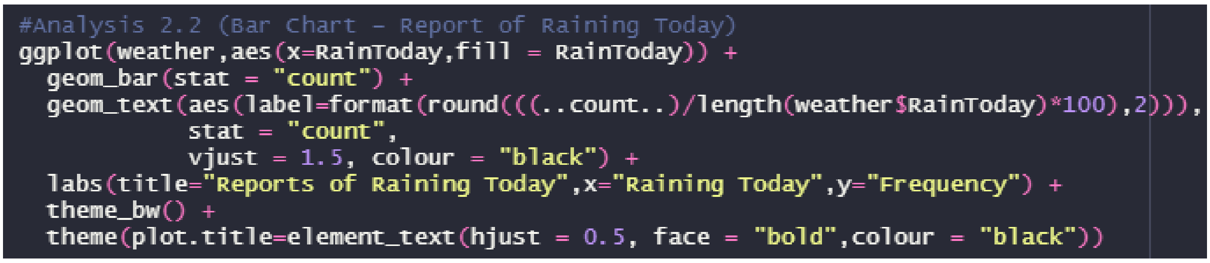


Figure 26: Compass Rose – Report of Wind Direction and Wind Speed at 9 am

Based on figure 26 shows the wind speed at 9 am. From the figure, we can see that there is not much wind activity at 9 am as the highest wind speed report is at 6-8 km/h. The only viable assumption is that there is less wind activity in the morning than in the afternoon and night. This is due to the cycle of warming during the day and cooling at night explains why (Halblaub, 2014). In conclusion, we can indicate that it is an excellent time to have the Formula One United States Grand Prix at 9 am as there is less wind activity, allowing the Formula One car to have more downforce to be applied at this hour (Noble, 2017).

Analysis 2.2: Find the number of reports of raining today?



```
#Analysis 2.2 (Bar Chart - Report of Raining Today)
ggplot(weather,aes(x=RainToday,fill = RainToday)) +
  geom_bar(stat = "count") +
  geom_text(aes(label=format(round(((..count..)/length(weather$RainToday)*100),2))),
            stat = "count",
            vjust = 1.5, colour = "black") +
  labs(title="Reports of Raining Today",x="Raining Today",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 27: R code for bar chart – Report of Raining Today

Based on figure 27, it is the R code to create a bar chart displaying the number of reports of raining today. The reason for using bar graphs is because they are an effective way to compare items between different groups.

Based on figure 27, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_bar() is used to indicate the type of graph being used. The geom_text() function is used to insert text to the bar chart graph. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

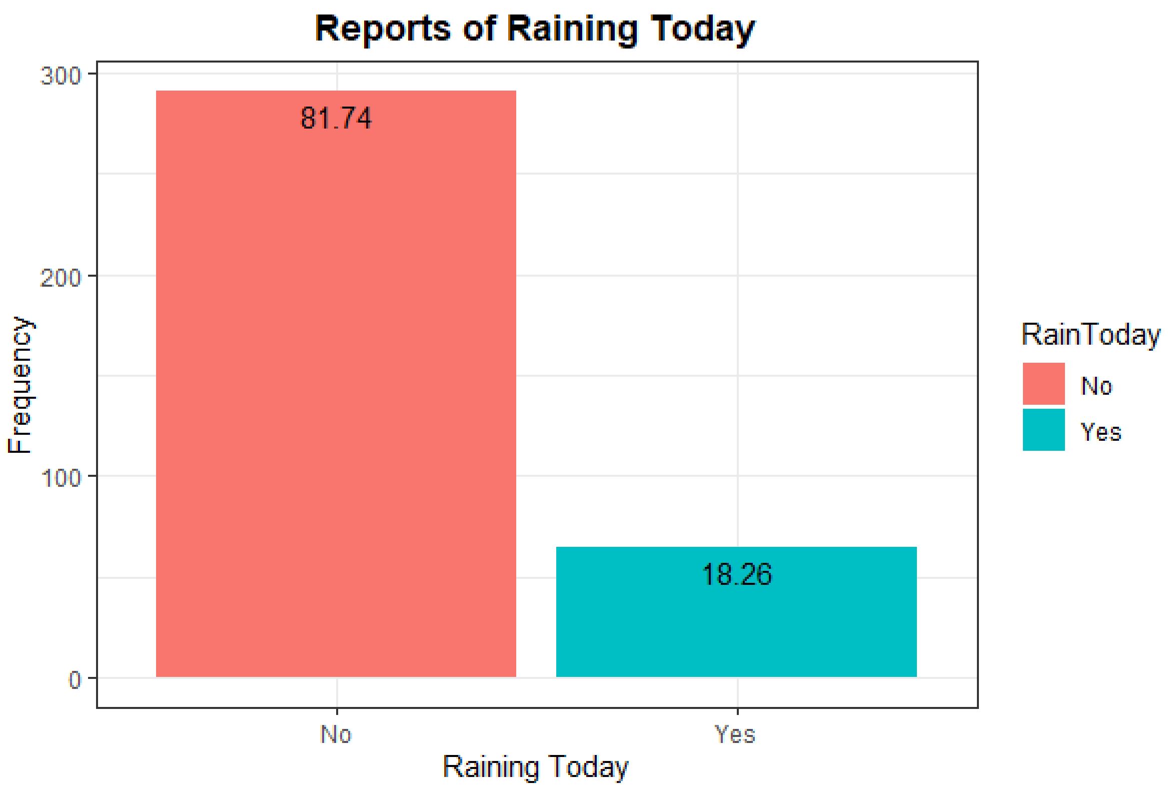
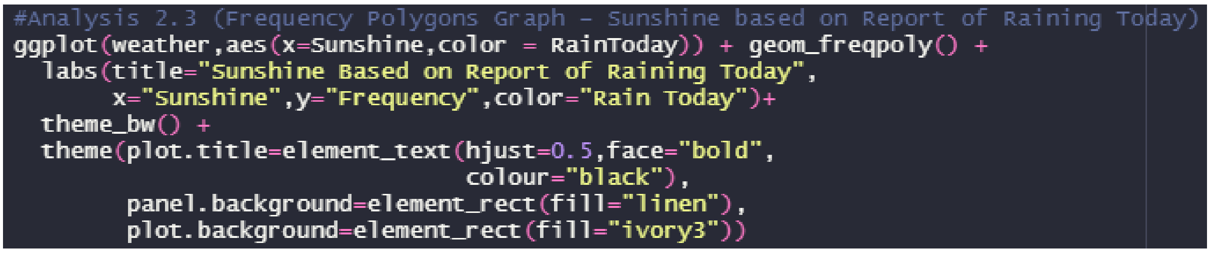


Figure 28: Bar Chart– Report of Raining Today

Based on figure 28 represents the Reports of Raining Today in a bar chart. From the figure, we can indicate that the possibility of raining today is 18% which is relatively low as almost 81% of the report states that it will not rain today. The only viable assumption that there is currently a lack of cloud formation in the area (Doyle, 2021). The other logical assumption is that the weather in the United States is presently windy or warm, with a low probability of rain. In conclusion, although the possibility of rain is low, the Formula One team can still have the Formula One United States Grand Prix, but they are expected to have one set of wet and intermediate compound tyres just in case it rains in the circuit (Anon., n.d.).

Analysis 2.3: Find the sunshine exposure based on reports of raining today?



```
#Analysis 2.3 (Frequency Polygons Graph - Sunshine based on Report of Raining Today)
ggplot(weather,aes(x=Sunshine,color = RainToday)) + geom_freqpoly() +
  labs(title="Sunshine Based on Report of Raining Today",
       x="Sunshine",y="Frequency",color="Rain Today")+
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5,face="bold",
                                colour="black"),
        panel.background=element_rect(fill="linen"),
        plot.background=element_rect(fill="ivory3"))
```

Figure 29: R code for frequency polygons graph – Sunshine based on Reports of Raining Today

Based on figure 29, it is the R code to create a frequency polygon displaying the Sunshine based on Report of Raining Today. The reason for using frequency polygons graphs is because frequency polygons are an excellent choice for displaying cumulative frequency distributions.

Based on figure 29, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_freqpoly() is used to indicate the type of graph being used. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

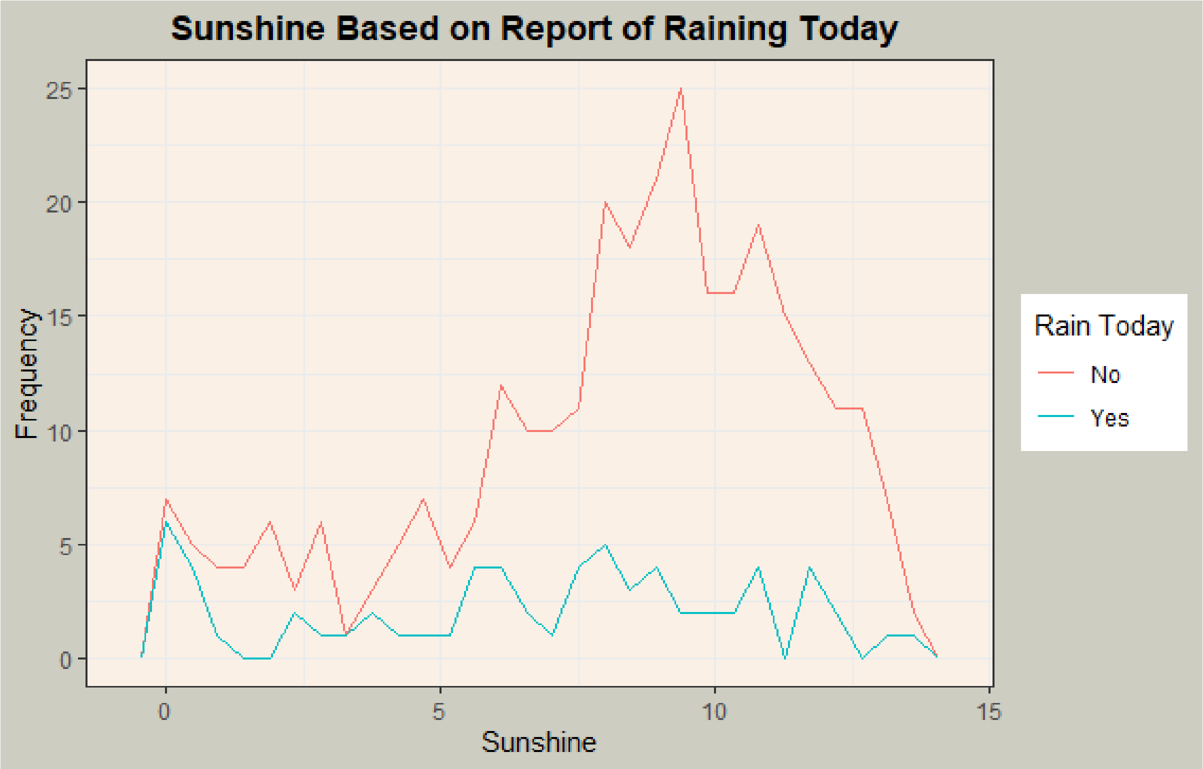
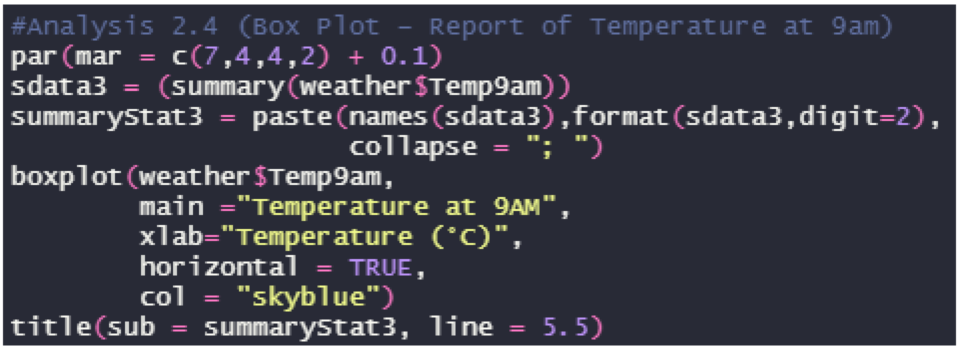


Figure 30: Frequency Polygons – Sunshine based on Report of Raining Today

Based on figure 30 represents the sunshine based on reports of Raining Today in Frequency Polygons. From the figure, the low probability of raining today is due to many reports of the high amount of sunshine exposure in the areas with no rain. This is because of the lack of clouds formed in the area, which leads to high Sunshine exposure (Doyle, 2021). In conclusion, the Formula One United States Grand Prix is best to happen at 9 am as the track heats up. This is because it helps to get better performance in the tyres of the Formula One Cars as different types of Formula One Cars tyres have a different operating temperature which allows better speed and performance of the Formula One Cars (Anon., 2018).

Analysis 2.4: Find the temperature at 9 am?

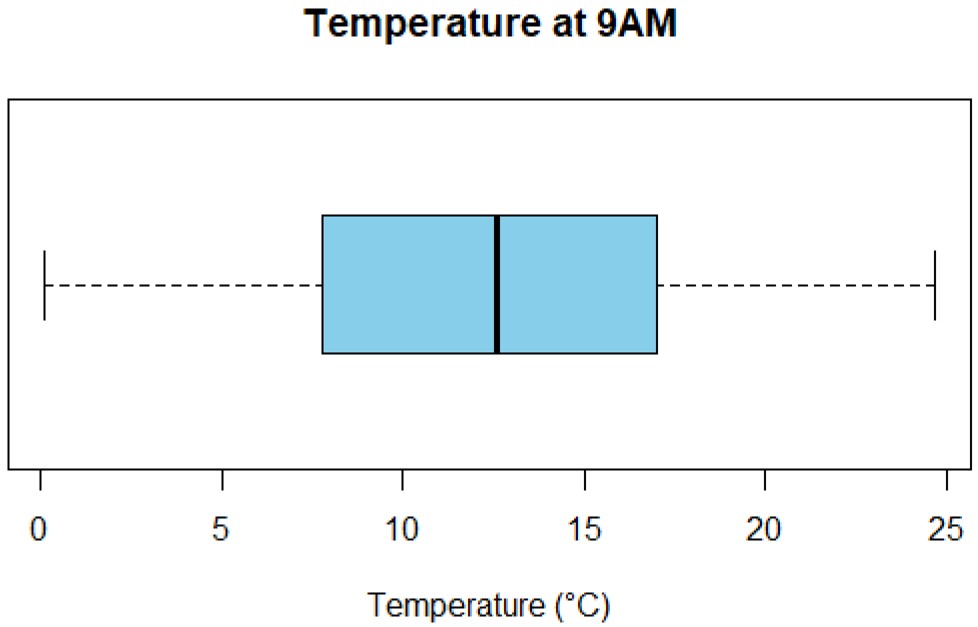


```
#Analysis 2.4 (Box Plot - Report of Temperature at 9am)
par(mar = c(7,4,4,2) + 0.1)
sdata3 = (summary(weather$Temp9am))
summaryStat3 = paste(names(sdata3),format(sdata3,digit=2),
                     collapse = "; ")
boxplot(weather$Temp9am,
        main ="Temperature at 9AM",
        xlab="Temperature (°C)",
        horizontal = TRUE,
        col = "skyblue")
title(sub = summaryStat3, line = 5.5)
```

Figure 31: R code for box plot – Report of Temperature at 9 am

In figure 31, it is the R code to create a box plot displaying the temperature at 9 am. The reason for using a box plot is because a box plot is an excellent choice for providing helpful information such as mean, median and range.

Based on figure 31, the par() function is used to set or query graphical parameters. It will allow space for the details that are listed below the graph. Variable "sdata3" are used to store a summary of variable "Temp9am" using the summary() function. The variable "summaryStat3" will contain the reformatted data that will display the data in one line. The boxplot() function is used to create the box plot with the title name, x-axis names, y-axis names, colour and many more. Last but not least, the title() function is used to display the summary data below the graph.

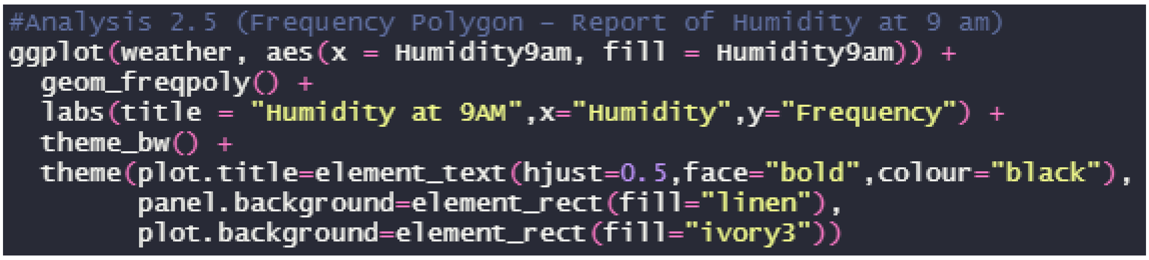


Min. 0.1; 1st Qu. 7.8; Median 12.6; Mean 12.4; 3rd Qu. 17.0; Max. 24.7

Figure 32: Box Plot – Report of Temperature at 9 am

Based on figure 32 represents the temperature at 9 am in a box plot. From the figure, we can indicate that the temperature at 9 am is between the range of 7.6 °C to 17 °C, with the median temperature at 12.6 °C and mean temperature at 12.4 °C. The can indicate that surrounding temperature is not too hot and not too cold (O'MARA, 2019). This is important because if the temperature is to high driver will have high chances of dehydration. In conclusion, the Formula One United States Grand Prix is best to have it at 9 am as it helps the drivers to stay cool in a two and half hour race in the Formula One Car (RACERS, n.d.).

Analysis 2.5: Find the humidity at 9 am?



```
#Analysis 2.5 (Frequency Polygon - Report of Humidity at 9 am)
ggplot(weather, aes(x = Humidity9am, fill = Humidity9am)) +
  geom_freqpoly() +
  labs(title = "Humidity at 9AM",x="Humidity",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5,face="bold",colour="black"),
        panel.background=element_rect(fill="linen"),
        plot.background=element_rect(fill="ivory3"))
```

Figure 33: R code for frequency polygon graph – Report of Humidity at 9 am

Based on figure 33, it is the R code to create a frequency polygon displaying the humidity at 9 am. The reason for using frequency polygons graphs is because frequency polygons are an excellent choice for displaying cumulative frequency distributions.

Based on figure 33, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_freqpoly() is used to indicate the type of graph being used. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

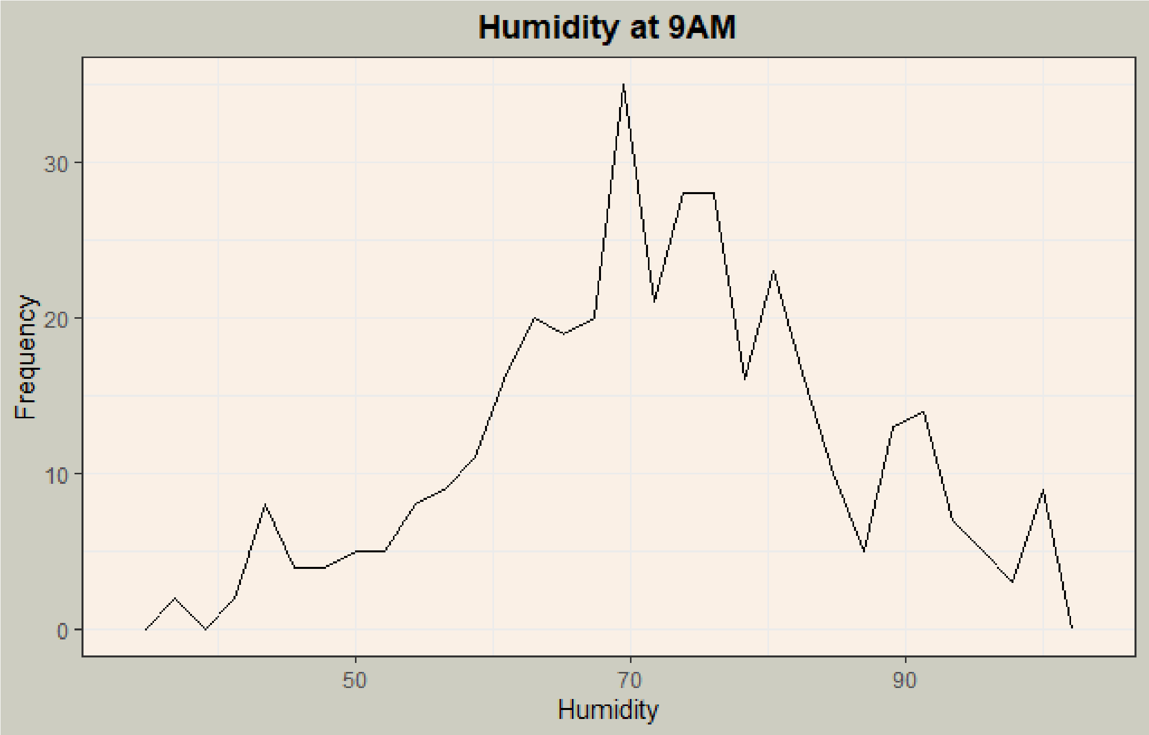


Figure 34: Frequency Polygon – Report of Humidity at 9 am

Based on figure 34 represents the Humidity at 9 am. From the figure, we can see that there are many reports of humidity at 60% onwards. This can indicate that there is a possibility of fog as there are high numbers of humidity reports at 70%. This is because more water vapour in the air forms fogs when the environment is more humid (Dunn, 2011). In conclusion, it is not suitable to have a Formula One United States Grand Prix at this hour as the visibility is not very good and high humidity level affects the engine performance of the Formula One Cars, which cause loss of power and speed (Admin, 2016).

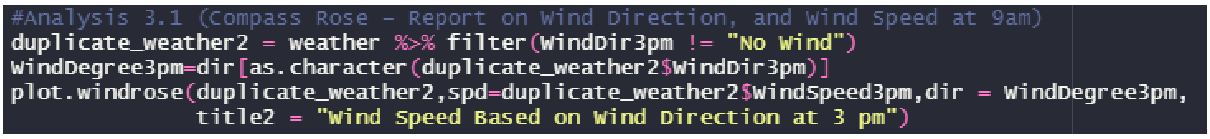
Conclusion

Based on question 2, we can conclude that 9 am is an excellent time to have a Formula One United States Grand Prix. This is because the wind speed and temperature are suitable as they are not much wind activity and the temperature is not too hot. This is crucial as it allows the Formula One car to have the best performance when there is less wind activity, and the surrounding temperature is low. Furthermore, the possibility of raining today is low, and there is a high sunshine exposure that helps the Formula One cars to have the best performance. Although the humidity is not too good, the Formula One United States Grand Prix still can go on as it makes the sport more competitive for the drivers and the Formula One Teams.

Question 3: Is 3 pm a good time to have a golf competition?

The question is crucial as it describes the weather conditions of the golf course. The organiser needs this information to plan whether it is a suitable time to have a golf competition or not. This is crucial to prevent safety hazard to the crowds, golfer and nearby houses.

Analysis 3.1: Find the wind direction and wind speed at 3 pm?



```
#Analysis 3.1 (Compass Rose - Report on Wind Direction, and Wind Speed at 9am)
duplicate_weather2 = weather %>% filter(WindDir3pm != "No Wind")
WindDegree3pm=dir[as.character(duplicate_weather2$WindDir3pm)]
plot.windrose(duplicate_weather2,spd=duplicate_weather2$WindSpeed3pm,dir = WindDegree3pm,
              title2 = "Wind Speed Based on Wind Direction at 3 pm")
```

Figure 35: R code for compass rose – Report on Wind Direction and Wind Speed at 3 pm

Based on figure 35, it is the R code to create a compass rose to display the Wind Direction and Wind Speed at 3 pm. The reason for using compass rose is that they can report the wind direction and wind speed simultaneously. The compass rose is like a radar, and it provides valuable information to give an analysis.

Based on figure 35, the variable “duplicate_weather2” contains a copy of the dataset weather, but it is filtered to remove all “No wind” data from variable “WindDir3pm” using the piping method. Variable “WindDegree3pm” contains the converted data from wind direction to wind degree. The “plot.windrose()” function will create the compass rose graph accordingly.

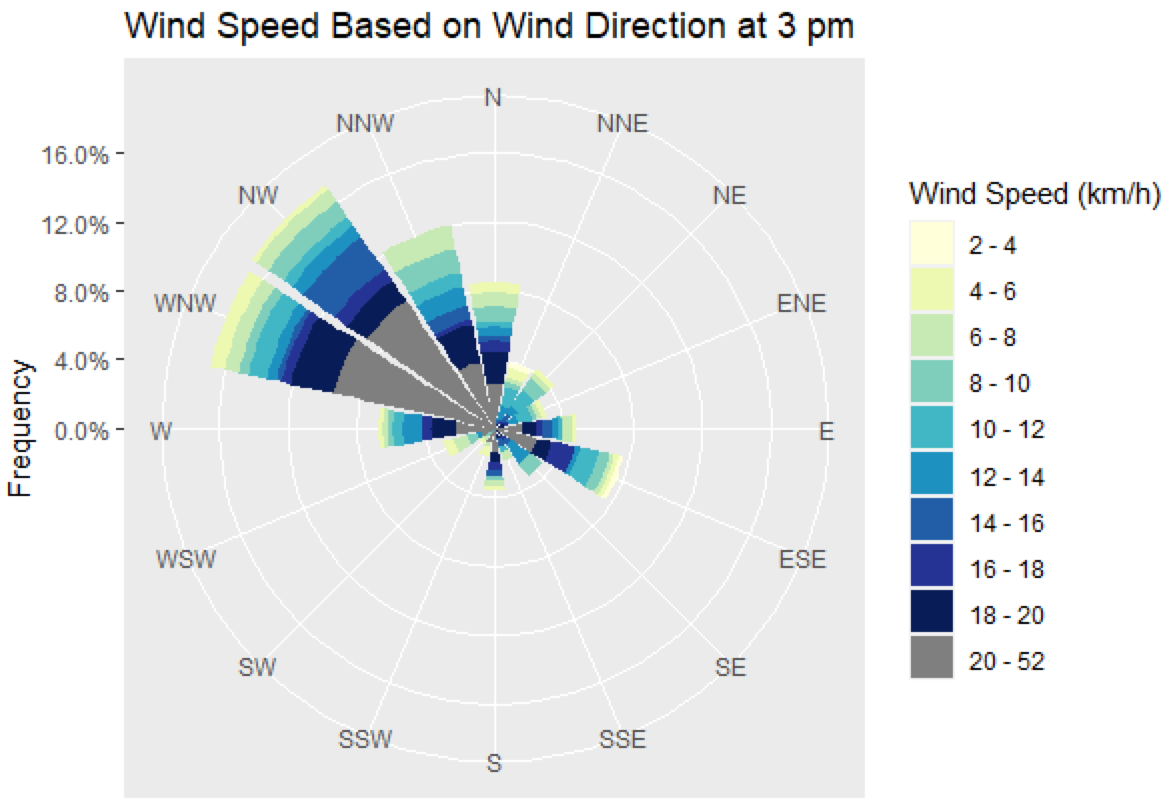
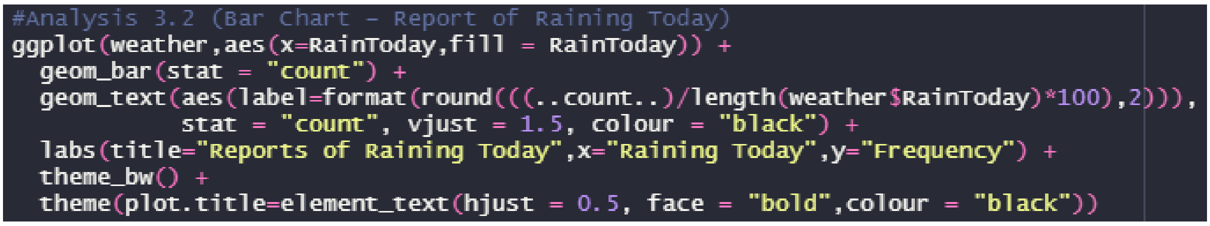


Figure 36: Compass Rose – Report of Wind Speed based on Wind Direction at 3 pm

Based on figure 36 shows the wind speed based on wind direction at 3 pm. From the figure, we can see that there is a lot of wind activity at 3 pm with high wind speed reports above 20 km/h. The only viable assumption is that there is more wind activity in the afternoon than in the morning. This is due to much of the tendency to be windier during daylight hour driven by sunlight and solar heating (Skilling, 2019). In conclusion, we can indicate that it is not a good time to have a golf competition at 3 pm as high wind speed will affect the golf balls distance and directions. It can also cause a safety hazard as the golfer cannot predict the ball's path due to strong winds.

Analysis 3.2: Find the number of reports of raining today?



```r
#Analysis 3.2 (Bar Chart - Report of Raining Today)
ggplot(weather,aes(x=RainToday,fill = RainToday)) +
  geom_bar(stat = "count") +
  geom_text(aes(label=format(round(((..count..)/length(weather$RainToday)*100),2))),
            stat = "count", vjust = 1.5, colour = "black") +
  labs(title="Reports of Raining Today",x="Raining Today",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 37: R code for bar chart – Report of Raining Today

Based on figure 37, it is the R code to create a bar chart displaying the number of reports of raining today. The reason for using bar graphs is because they are an effective way to compare items between different groups.

Based on figure 37, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_bar() is used to indicate the type of graph being used. The geom_text() function is used to insert text to the bar chart graph. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

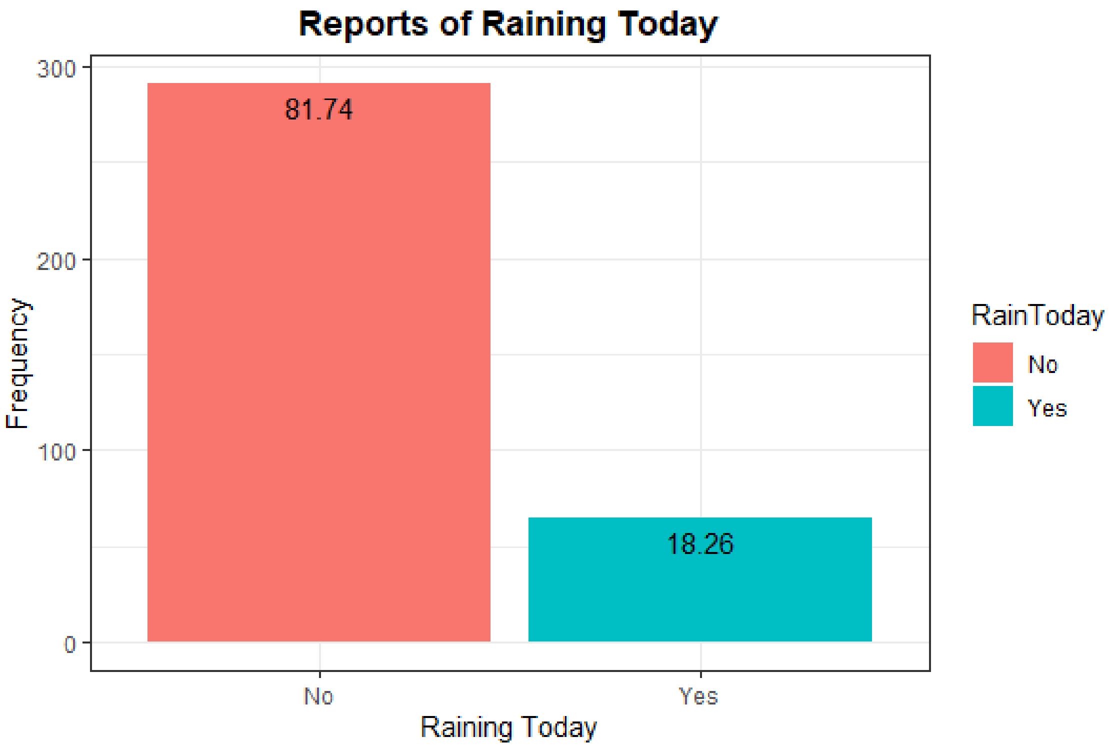
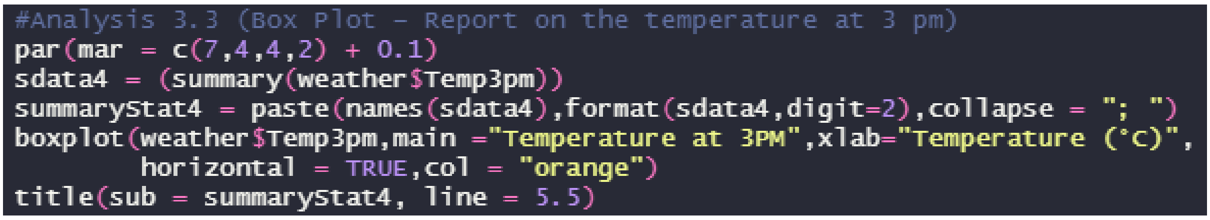


Figure 38: Bar Chart– Report of Raining Today

Based on figure 38 represents the Reports of Raining Today in a bar chart. From the figure, we can indicate that the possibility of raining today is 18% which is relatively low as almost 81% of the report states that it will not rain today. The only viable assumption that there is currently a lack of cloud formation in the area (Doyle, 2021). The other logical assumption is that the weather in the United States is presently windy or warm, with a low probability of rain. In conclusion, although the possibility of rain is low, the golf competition can still go on.

Analysis 3.3: Find the temperature at 3 pm?

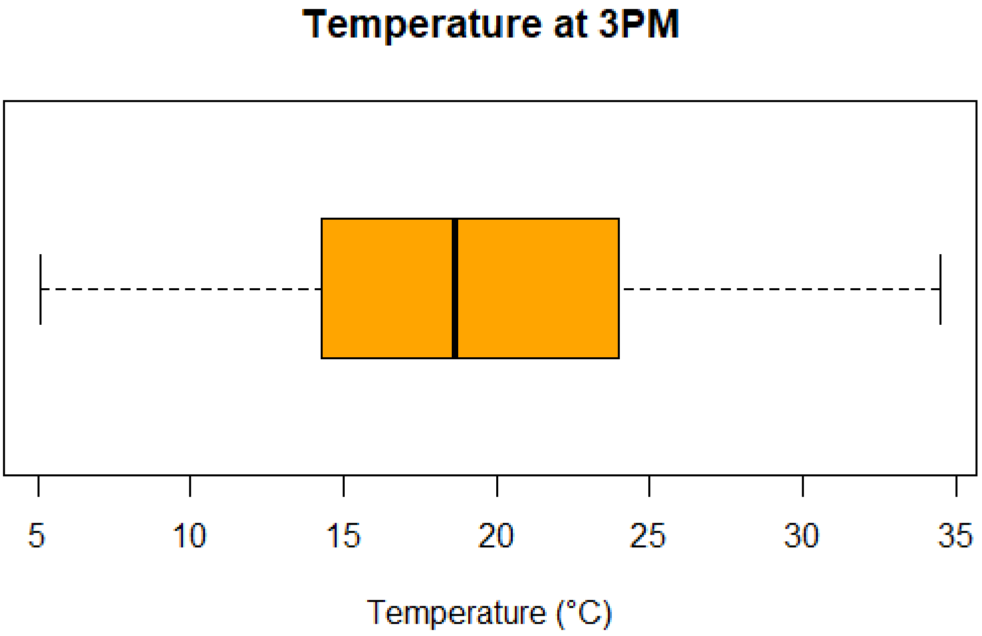


```
#Analysis 3.3 (Box Plot - Report on the temperature at 3 pm)
par(mar = c(7,4,4,2) + 0.1)
sdata4 = (summary(weather$Temp3pm))
summaryStat4 = paste(names(sdata4),format(sdata4,digit=2),collapse = "; ")
boxplot(weather$Temp3pm,main ="Temperature at 3PM",xlab="Temperature (°C)",
        horizontal = TRUE,col = "orange")
title(sub = summaryStat4, line = 5.5)
```

Figure 39: R code for box plot – Report on the temperature at 3 pm

In figure 39, it is the R code to create a box plot displaying the temperature at 3 pm. The reason for using a box plot is because a box plot is an excellent choice for providing helpful information such as mean, median and range.

Based on figure 39, the par() function is used to set or query graphical parameters. It will allow space for the details that are listed below the graph. Variable “sdata4” are used to store a summary of variable “Temp3pm” using the summary() function. The variable “summaryStat4” will contain the reformatted data that will display the data in one line. The boxplot() function is used to create the box plot with the title name, x-axis names, y-axis names, colour and many more. Last but not least, the title() function is used to display the summary data below the graph.

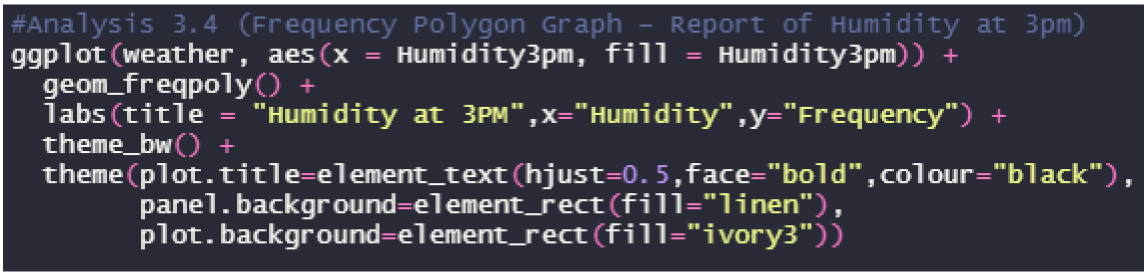


Min. 5.1; 1st Qu. 14.3; Median 18.6; Mean 19.2; 3rd Qu. 24.0; Max. 34.5

Figure 40: Box Plot – Report of Temperature at 3 pm

Based on figure 40 represents the temperature at 3 pm in a box plot. From the figure, we can indicate that the temperature at 3 pm is between the range of 14.3 °C to 24 °C, with the median temperature at 18.6 °C and mean temperature at 19.2 °C. With this, it can indicate that the temperature is also a little warmer but still playable. This is because the perfect temperature to play golf is 10 °C to 16 °C (Anon., 2015). In conclusion, it is still possible to have a golf competition, but the golfer is advised to bring a hat to prevent sunburns.

Analysis 3.4: Find the humidity at 3 pm?



```
#Analysis 3.4 (Frequency Polygon Graph - Report of Humidity at 3pm)
ggplot(weather, aes(x = Humidity3pm, fill = Humidity3pm)) +
  geom_freqpoly() +
  labs(title = "Humidity at 3PM",x="Humidity",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5,face="bold",colour="black"),
        panel.background=element_rect(fill="linen"),
        plot.background=element_rect(fill="ivory3"))
```

Figure 41: R code for frequency polygon graph – Report of Humidity at 3 pm

Based on figure 41, it is the R code to create a frequency polygon displaying the humidity at 3 pm. The reason for using frequency polygons graphs is because frequency polygons are an excellent choice for displaying cumulative frequency distributions.

Based on figure 41, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_freqpoly() is used to indicate the type of graph being used. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

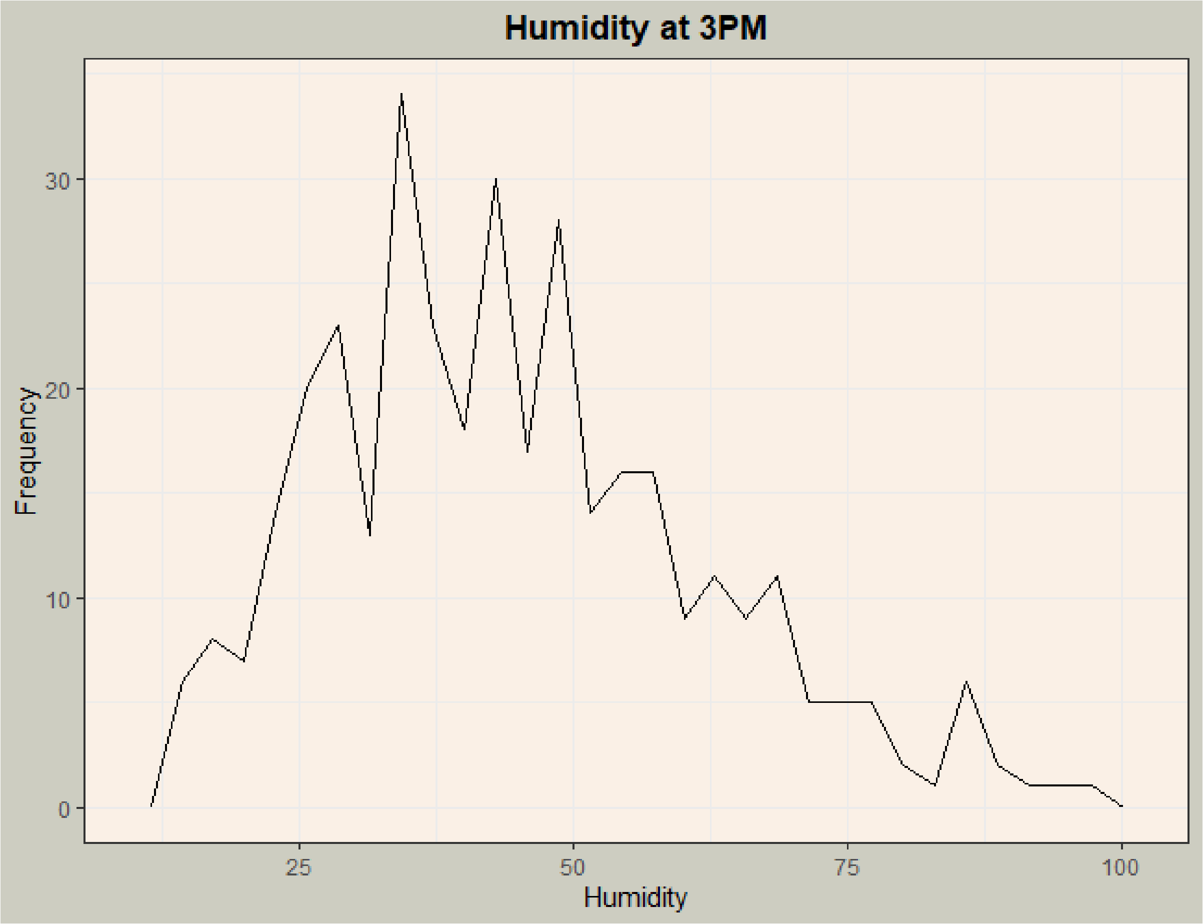


Figure 42: Box Plot – Report of Humidity at 3 pm

Based on figure 42 represents the Humidity at 3 pm. We can see that there are many humidity reports at 25% to 50% from the figure. This can indicate that there is a clear sky view with a low probability of fog. This is because the humidity drops during the day as the temperature rises and usually reaches its lowest value in the middle or late afternoon when the day's maximum temperature is recorded (SKILLING, 2014). In conclusion, it is suitable to have a golf competition at 3 pm as golfers can see where the ball will go and the hole for the golf ball.

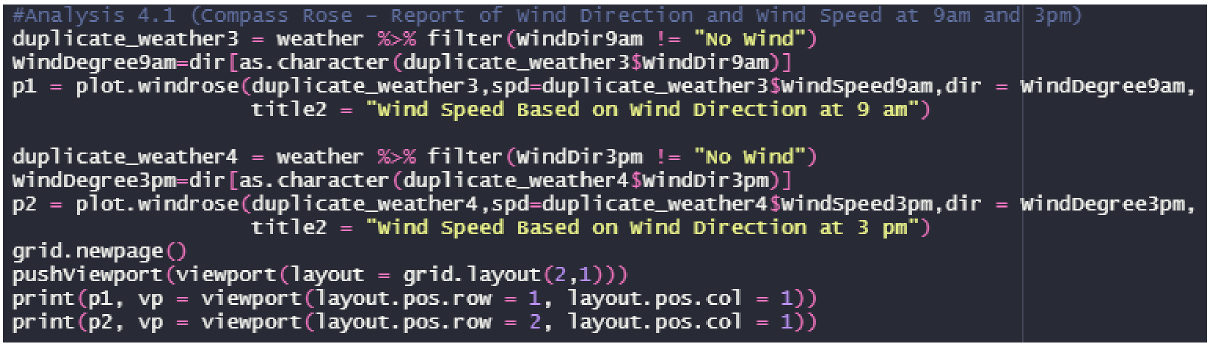
Conclusion

Based on question 3, it is considered not a good time to have a golf competition at 3 pm. This is because, at 3 pm, the wind speed is very high, which is above 20 km/h, based on analysis 3.1. With this, it is considered a safety hazard to the crowds and the golfer. Not only that, but it is also a little warmer to have a golf competition at 3 pm, based on analysis 3.3. Although the humidity is considered good and the raining report is low, it is still not advised to have a gold competition at 3 pm as the wind speed is a crucial factor that cannot be disregarded.

Question 4: When is a good time for planes to depart and land in the United States airport?

The question is crucial to the pilot and the airport as it allows them to plan how they would want to land or depart from the airport. The pilot is expected to calculate the wind speed and wind direction to prevent the pilot from over-speeding when landing or taking off from the airport. If wind speeds are too high, the pilot has to call an abort and delay the flight to reduce the risk of an accident. This information is crucial as it involved human lives.

Analysis 4.1: Find the wind direction and wind speed at 9 am and 3 pm?



```
#Analysis 4.1 (Compass Rose - Report of Wind Direction and Wind Speed at 9am and 3pm)
duplicate_weather3 = weather %>% filter(WindDir9am != "No Wind")
WindDegree9am=dir[as.character(duplicate_weather3$WindDir9am)]
p1 = plot.windrose(duplicate_weather3,spd=duplicate_weather3$WindSpeed9am,dir = WindDegree9am,
                   title2 = "Wind Speed Based on Wind Direction at 9 am")

duplicate_weather4 = weather %>% filter(WindDir3pm != "No Wind")
WindDegree3pm=dir[as.character(duplicate_weather4$WindDir3pm)]
p2 = plot.windrose(duplicate_weather4,spd=duplicate_weather4$WindSpeed3pm,dir = WindDegree3pm,
                   title2 = "Wind Speed Based on Wind Direction at 3 pm")
grid.newpage()
pushViewport(viewport(layout = grid.layout(2,1)))
print(p1, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))
print(p2, vp = viewport(layout.pos.row = 2, layout.pos.col = 1))
```

Figure 43: R code for compass rose graph – Report on Wind Direction and Wind Speed at 9 am and 3 pm

Based on figure 43, it is the R code to create a compass rose to display the Wind Direction and Wind Speed at 9 am and 3 pm. The reason for using compass rose is that they can report the wind direction and wind speed simultaneously. The compass rose is like a radar, and it provides valuable information to give an analysis.

Based on figure 43, the variable “duplicate_weather3” contains a copy of the dataset weather, but it is filtered not to contain any “No Wind” data in variable “WindDir9am” by using the piping method. Variable “WindDegree9am” will contain the converted data from wind direction to wind degree. Variable “p1” will contain the R code to create the compass rose for Wind Direction based on wind speed at 9 am.

The variable “duplicate_weather4” contains a copy of the dataset weather, but it is filtered not to contain any “No Wind” data in variable “WindDir3pm” by using the piping method. Variable “WindDegree3pm” will contain the converted data from wind direction to wind degree. Variable “p2” will contain the R code to create the compass rose for Wind Direction based on wind speed at 3 pm.

The grid.newpage() function creates a new page. The “pushViewport(viewport(layout=grid.layout(2,1)))” functions to adjust the page to fit 2 images in 1 page. The following print() function will display both graph in the same page.

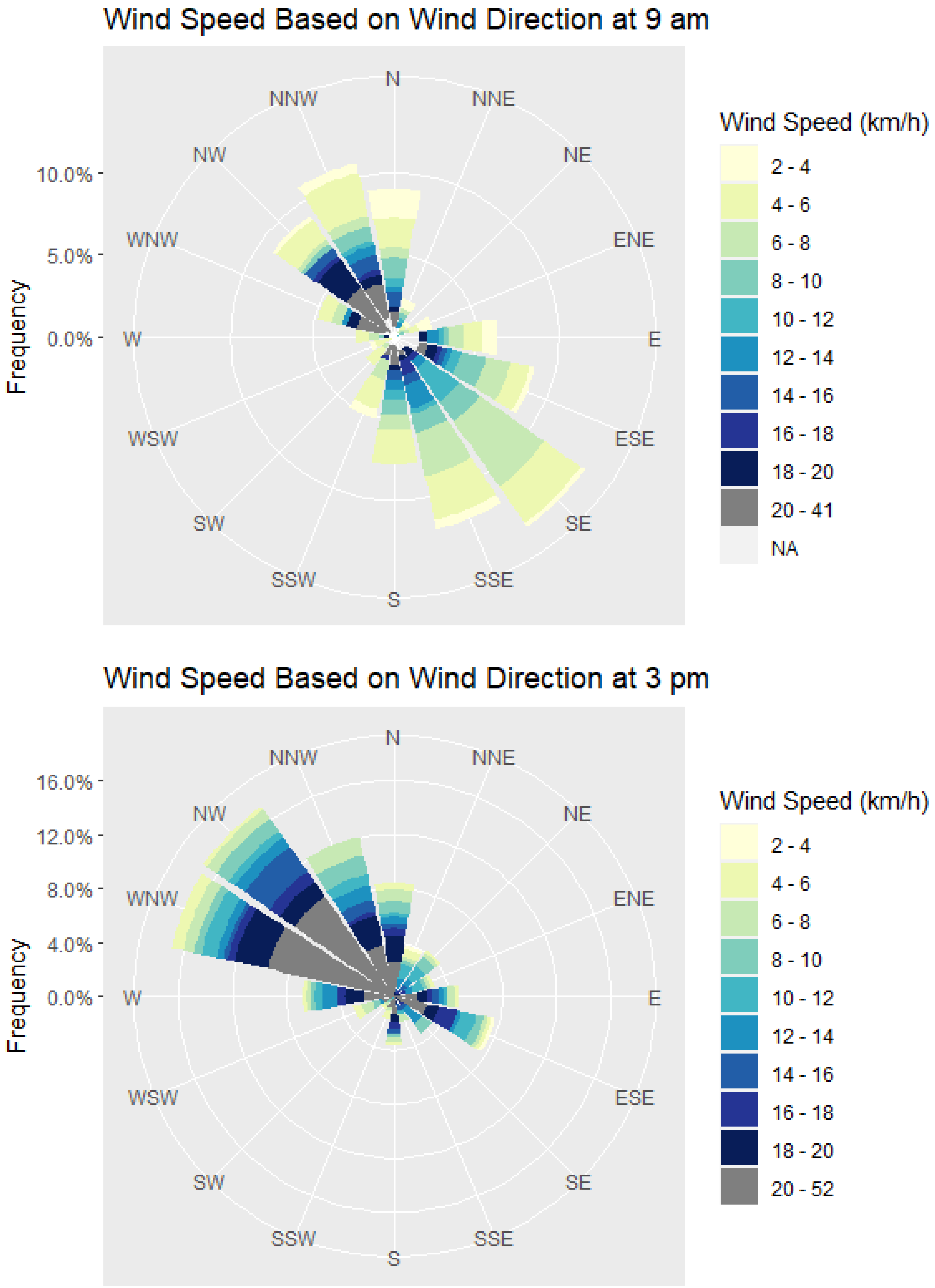
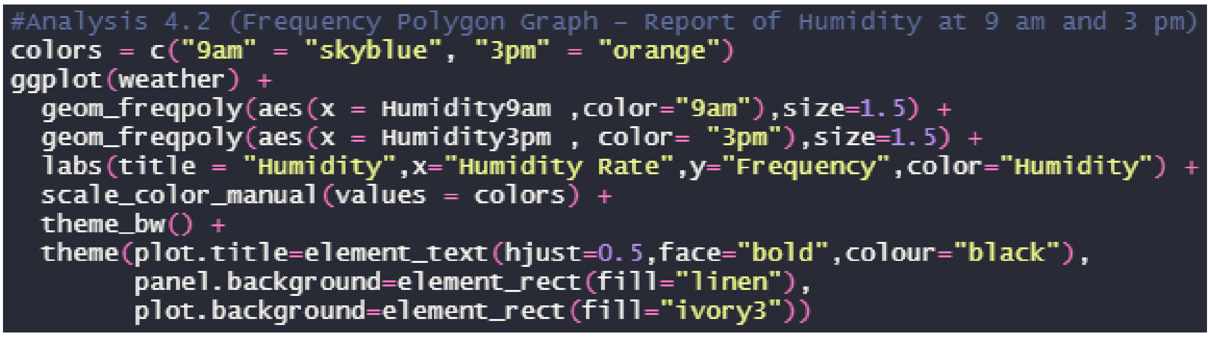


Figure 44: Compass Rose – Report of Wind Direction and Wind Speed at 9 am and 3 pm

Figure 44 shows the wind speed based on wind direction at 9 am and the wind speed based on wind direction at 3 pm. From the figure, we can see that there is less wind activity at 9 am as the most report of wind speeds are at 6-8 km/h, while at 3 pm, we can see that there is a lot of wind activity at 3 pm with a high report of wind speed above 20 km/h. The only viable assumption is that there is more wind activity in the afternoon than in the morning. This is due to much of the tendency to be windier during daylight hour driven by sunlight and solar heating (Skilling, 2019). In conclusion, the best time to land a plane in the United States airport is at 9 am as there is less wind activity, and it also lowers the risks of accidents or errors.

Analysis 4.2: Find the humidity at 9 am and 3 pm?



```
#Analysis 4.2 (Frequency Polygon Graph - Report of Humidity at 9 am and 3 pm)
colors = c("9am" = "skyblue", "3pm" = "orange")
ggplot(weather) +
  geom_freqpoly(aes(x = Humidity9am ,color="9am"),size=1.5) +
  geom_freqpoly(aes(x = Humidity3pm , color= "3pm"),size=1.5) +
  labs(title = "Humidity",x="Humidity Rate",y="Frequency",color="Humidity") +
  scale_color_manual(values = colors) +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5,face="bold",colour="black"),
        panel.background=element_rect(fill="linen"),
        plot.background=element_rect(fill="ivory3"))
```

Figure 45: R code for frequency polygon graph – Report of Humidity at 9 am and 3 pm

Based on figure 45, it is the R code to create a frequency polygon displaying the humidity at 9 am and 3 pm. The reason for using frequency polygons graphs is because frequency polygons are an excellent choice for displaying cumulative frequency distributions.

Based on figure 45, the "colors" variable is used to declare the colour type before used in the ggplot() function. The ggplot() function describes the dataset used. Furthermore, geom_freqpoly() is used to indicate the type of graph being used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. Moreover, the scale_color_manual() function is used to allow you to specify your own set of mappings from levels in the data to aesthetic values. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

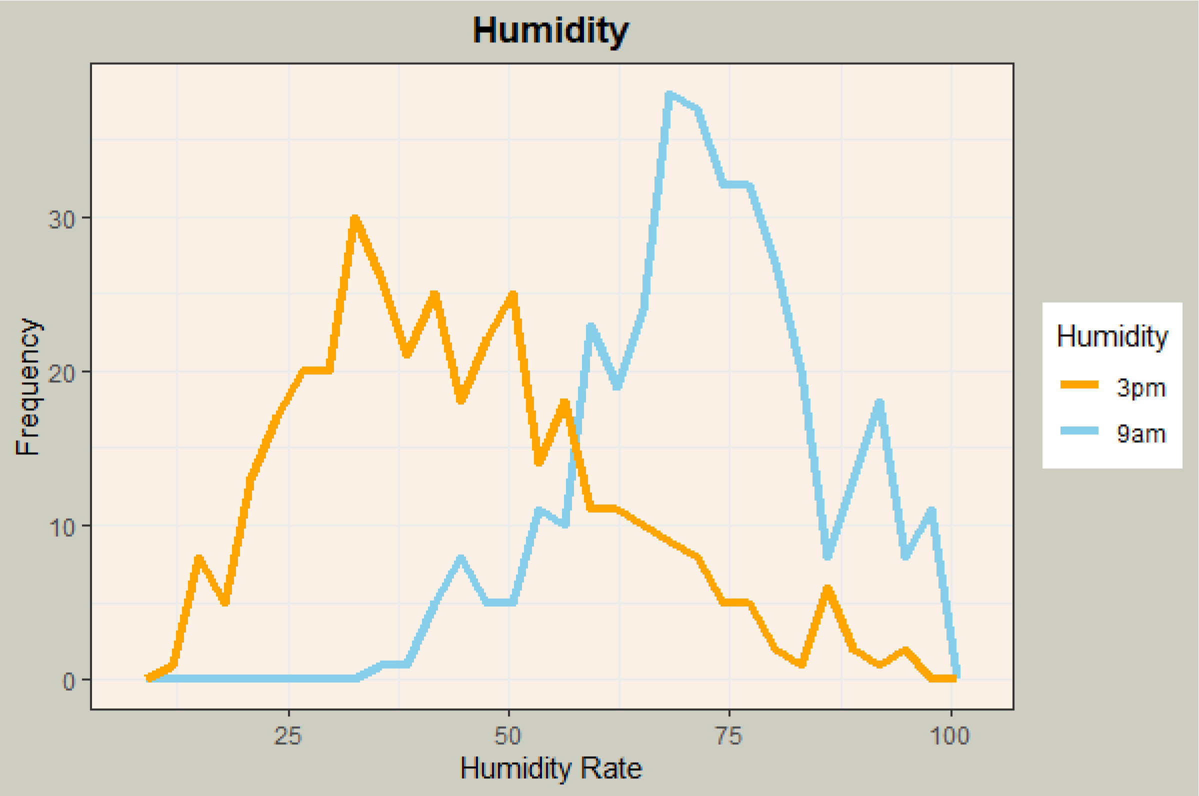
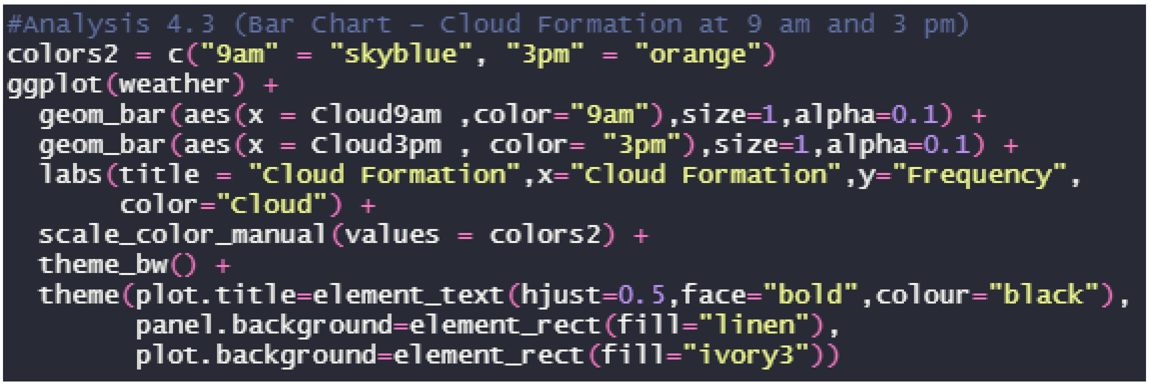


Figure 46: Frequency Polygons – Humidity Rates at 9 am and 3 pm

The figure above represents the humidity between 9 am and 3 pm in the United States. Based on the figure, we can indicate that the humidity rates are higher in the morning at 9 am compared to in the afternoon at 3 pm. This can be due to the effect of the surrounding temperature (Anon., n.d.). This is because the humidity is highest in the morning as the temperature is coolest at that time and otherwise in the afternoon (Anon., n.d.). In conclusion, the best time to land or depart planes from the US airport is at 3 pm as visibility is better than in the morning, where the fog is the heaviest at 9 am.

Analysis 4.3: Find the cloud formation at 9 am and 3 pm?



```r
#Analysis 4.3 (Bar Chart - Cloud Formation at 9 am and 3 pm)
colors2 = c("9am" = "skyblue", "3pm" = "orange")
ggplot(weather) +
  geom_bar(aes(x = Cloud9am ,color="9am"),size=1,alpha=0.1) +
  geom_bar(aes(x = Cloud3pm , color= "3pm"),size=1,alpha=0.1) +
  labs(title = "Cloud Formation",x="Cloud Formation",y="Frequency",
       color="Cloud") +
  scale_color_manual(values = colors2) +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5,face="bold",colour="black"),
        panel.background=element_rect(fill="linen"),
        plot.background=element_rect(fill="ivory3"))
```

Figure 47: R code for Bar Chart – Cloud Formation at 9 am and 3 pm

Based on figure 47, it is the R code to create a bar chart displaying the cloud formation at 9 am and 3 pm. The reason for using bar graphs is because they are an effective way to compare items between different groups.

Based on figure 47, the “colors2” variable is used to declare the colour type before used in the ggplot() function. The ggplot() function describes the dataset used. Furthermore, geom_bar() is used to indicate the type of graph being used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. Moreover, the scale_color_manual() function is used to allow you to specify your own set of mappings from levels in the data to aesthetic values. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

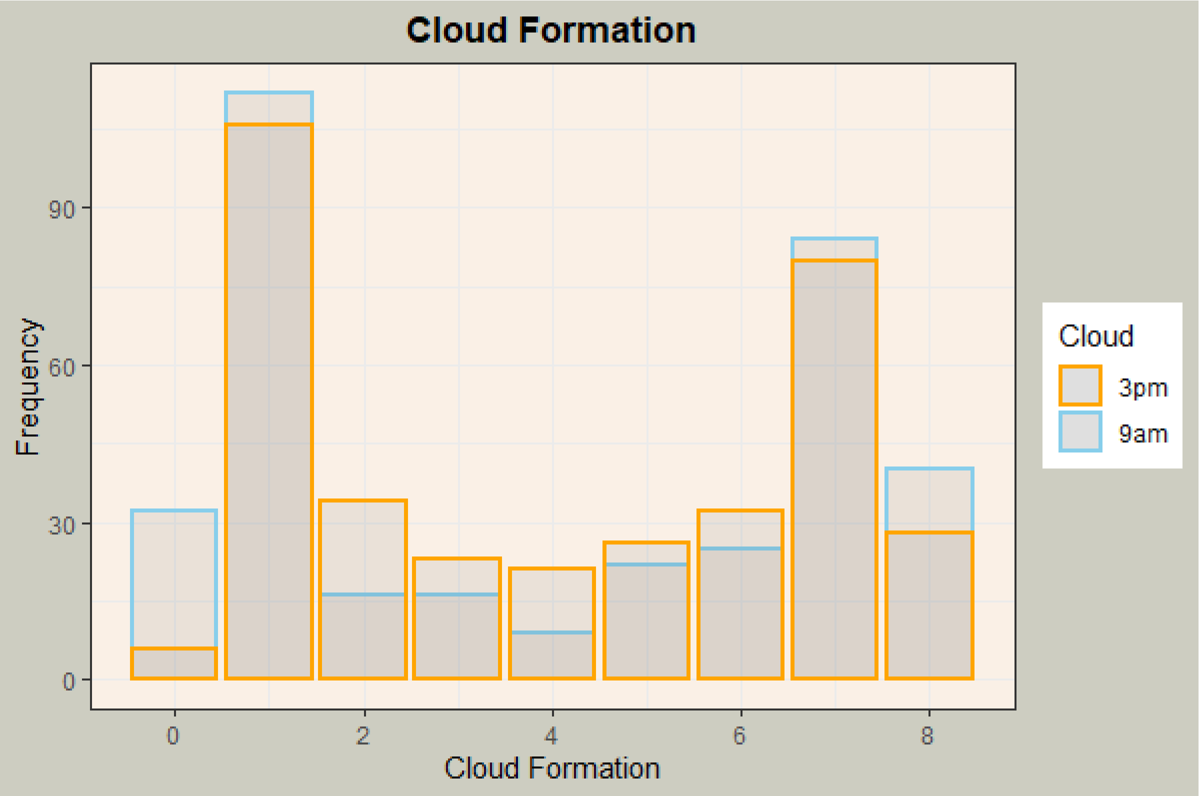
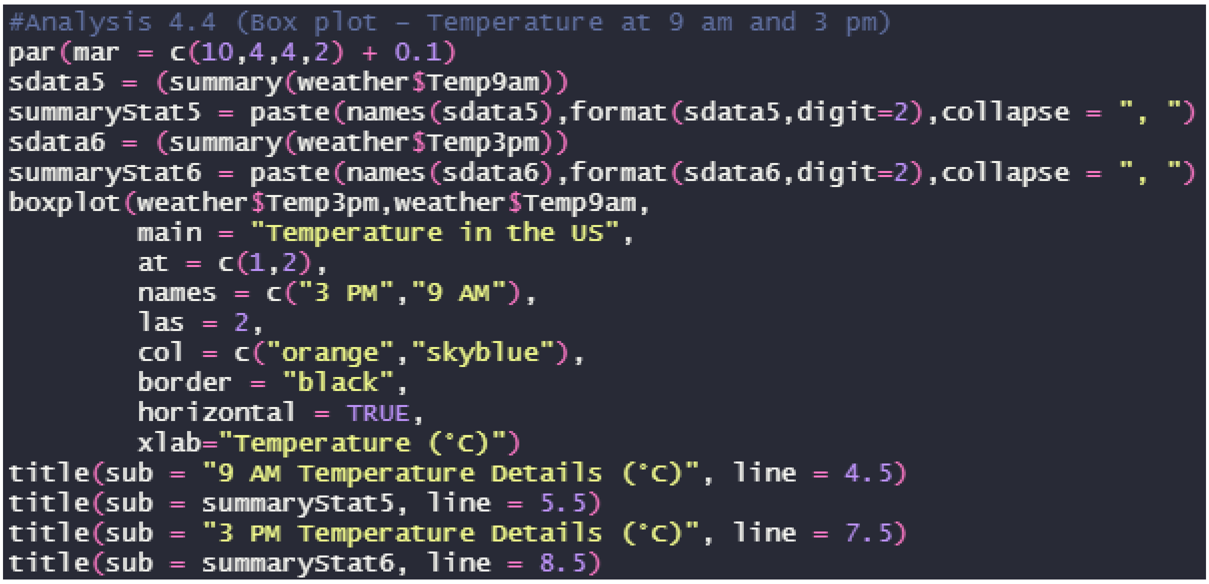


Figure 48: Bar Chart – Cloud Formation at 9 am and 3 pm

Based on figure 48 represents the cloud formation between 9 am and 3 pm in the United States. Based on the figure, we can indicate that the cloud formation for 9 am and 3 pm are the same as high reports are reporting cloud formation at 1. The viable assumption is that there is a lack of evaporation in that area (Anon., n.d.). This is because the more evaporation happens in that area, the more clouds can be formed (Anon., n.d.). In conclusion, both times at 9 am, or 3 pm are considered a good time to land or depart their planes from the United States airport as visibility is deemed to be good. This is because there is less cloud in both time.

Analysis 4.4: Find the temperature at 9 am and 3 pm?



```
#Analysis 4.4 (Box plot - Temperature at 9 am and 3 pm)
par(mar = c(10,4,4,2) + 0.1)
sdata5 = (summary(weather$Temp9am))
summaryStat5 = paste(names(sdata5),format(sdata5,digit=2),collapse = ", ")
sdata6 = (summary(weather$Temp3pm))
summaryStat6 = paste(names(sdata6),format(sdata6,digit=2),collapse = ", ")
boxplot(weather$Temp3pm,weather$Temp9am,
        main = "Temperature in the US",
        at = c(1,2),
        names = c("3 PM","9 AM"),
        las = 2,
        col = c("orange","skyblue"),
        border = "black",
        horizontal = TRUE,
        xlab="Temperature (°C)")
title(sub = "9 AM Temperature Details (°C)", line = 4.5)
title(sub = summaryStat5, line = 5.5)
title(sub = "3 PM Temperature Details (°C)", line = 7.5)
title(sub = summaryStat6, line = 8.5)
```

Figure 49: R code for Box plot – Temperature at 9 am and 3 pm

Figure 49 is the R code to create a box plot displaying the temperature at 9 am and 3 pm. The reason for using a box plot is because a box plot is an excellent choice for providing helpful information such as mean, median and range.

Based on figure 49, the par() function is used to set or query graphical parameters. It will allow space for the details that are listed below the graph. Variable “sdata5” and variable “sdata6” are used to store summary for variable “Temp9am” and variable “Temp3pm” using the summary() function. The variable “summaryStat5” and variable “summaryStat6 will contain the reformatted data that will display the data in one line. The boxplot() function is used to create the box plot with the title name, x-axis names, y-axis names, colour and many more. Last but not least, the title() function is used to display the summary data below the graph.

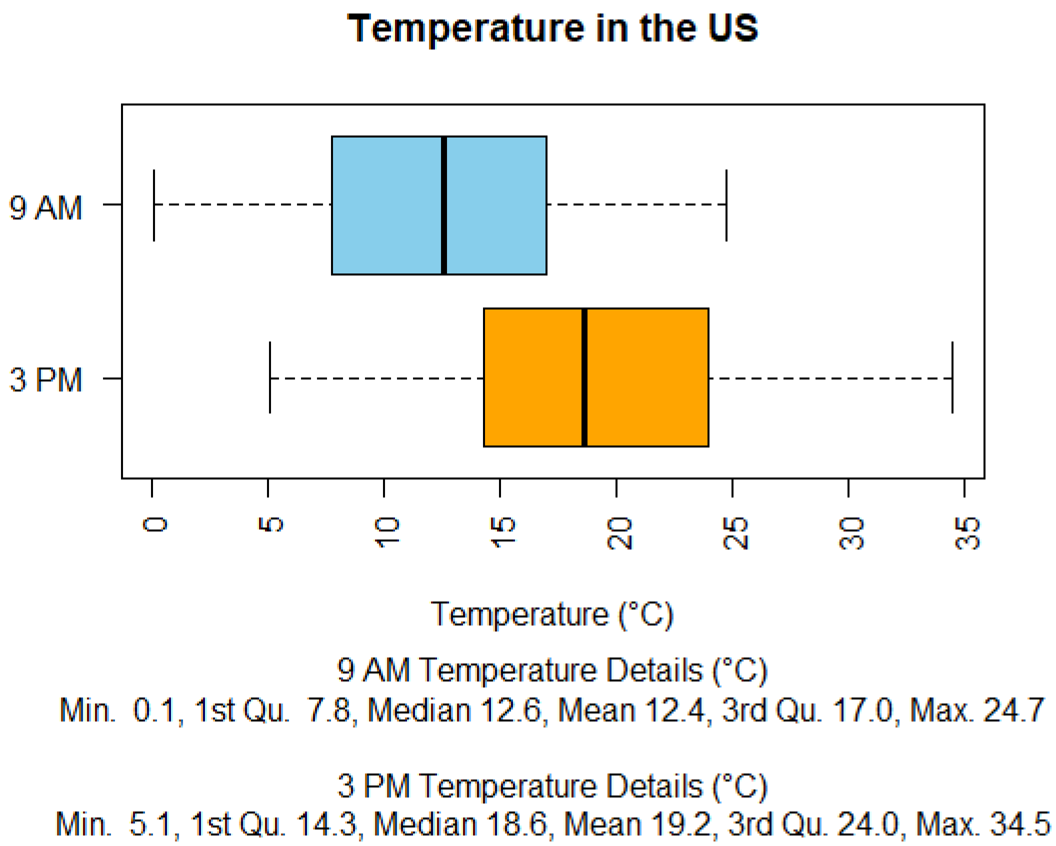


Figure 50: Box plot – Temperature at 9 am and 3 pm

Based on figure 50 represents the temperature between 9 am and 3 pm in the United States. Based on the figure, we can indicate that the surrounding temperature at 9 is cooler than the surrounding temperature at 3 pm. The viable assumption is that the temperature in the morning is cooler than the temperature in the afternoon. This is because the sun is vertically over your head in the afternoon compared to the morning, where the sun is just rising (Anon., 2019). In conclusion, the best time for planes to depart is at 9 am, as the hotter the surrounding temperature, the more power and fuel are required to gain the same thrust and lift as they would in cooler climates (Cappucci, 2018).

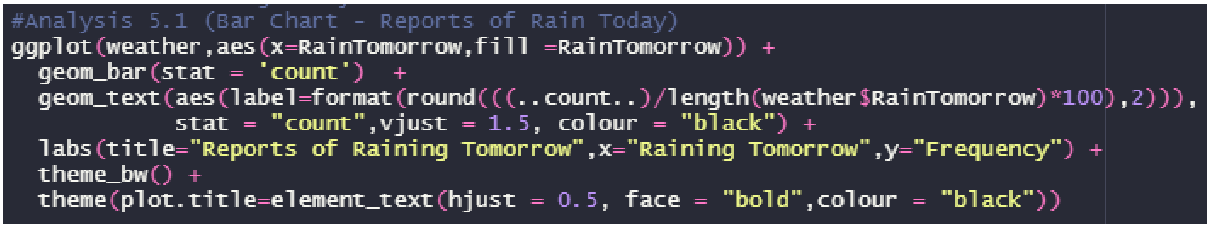
Conclusion

Based on question 4, the best time for planes to depart or land at the United States airport is 3 pm. This is because there is less probability of fog at 3 pm compared to 9 am. Although the temperature and the wind speed says otherwise, the pilot is advised to use more fuel and power to overcome the high temperature at 3 pm and the wind speed at 3 pm. The National Transportation Safety Board study report shows that more than two-thirds of all weather-related aviation crashes have been fatal (Law, 2020). Fog, snow and other natural elements make flying more difficult for pilots (Law, 2020). This is why departing or landing with a high fog concentration is not advised, increasing the risk of aviation accidents.

Question 5: Is it possible to have an outdoor activity tomorrow?

This question is crucial for tourists as it allows tourists to plan activity or sports that require information about the weather like boating sport, hot air balloon, and many more. This information is crucial as allow tourist to prepare and plan activities for the next day, that the tourist will want to visit or have some fun activities in the United States.

Analysis 5.1: Find the number of reports of raining tomorrow?



```r
#Analysis 5.1 (Bar Chart - Reports of Rain Today)
ggplot(weather,aes(x=RainTomorrow,fill =RainTomorrow)) +
  geom_bar(stat = 'count')  +
  geom_text(aes(label=format(round(((..count..)/length(weather$RainTomorrow)*100),2))),
            stat = "count",vjust = 1.5, colour = "black") +
  labs(title="Reports of Raining Tomorrow",x="Raining Tomorrow",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 51: R code for Bar Chart – Reports of Raining Today

Based on figure 51, it is the R code to create a bar chart displaying the number of reports of raining today. The reason for using bar graphs is because they are an effective way to compare items between different groups.

Based on figure 51, the ggplot() function describes the dataset used, while the aes() function is used to know which variable will be used from the dataset declared in the ggplot() function. Furthermore, geom_bar() is used to indicate the type of graph being used. The geom_text() function is used to insert text to the bar chart graph. The function of the labs() defines the title name, x-axis name, y-axis name, and legend table name. The theme_bw() function and theme() function enables the modification of text colour for title, adjustment of the title positions, and many more.

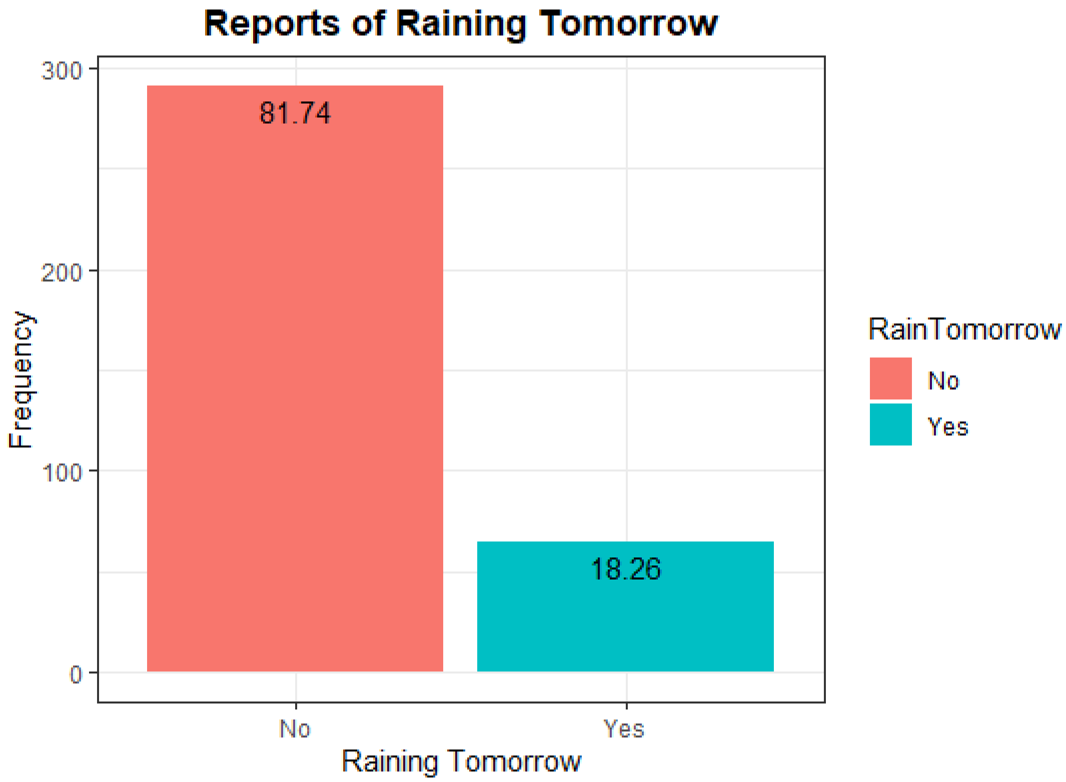
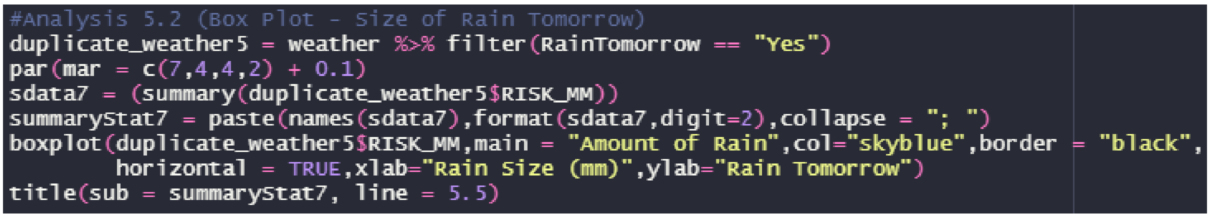


Figure 52: Box Plot – Report of Raining Tomorrow

Based on figure 52 represents the Reports of Raining Tomorrow in a bar chart. From the figure, we can indicate that the possibility of raining tomorrow is also relatively low, with 18% of reports raining tomorrow as 81% of the report states that it will not rain tomorrow. The only logical assumption for this is that there is currently a lack of clouds formation to have a downpour (Anon., n.d.). In conclusion, there is still a possibility it could rain tomorrow, tourists are still able to do some outdoor sport tomorrow, but they are advised to bring an umbrella just if it rains.

Analysis 5.2: Find how heavy is the rain tomorrow, if it rains?



```
#Analysis 5.2 (Box Plot - Size of Rain Tomorrow)
duplicate_weather5 = weather %>% filter(RainTomorrow == "Yes")
par(mar = c(7,4,4,2) + 0.1)
sdata7 = (summary(duplicate_weather5$RISK_MM))
summaryStat7 = paste(names(sdata7),format(sdata7,digit=2),collapse = "; ")
boxplot(duplicate_weather5$RISK_MM,main = "Amount of Rain",col="skyblue",border = "black",
        horizontal = TRUE,xlab="Rain Size (mm)",ylab="Rain Tomorrow")
title(sub = summaryStat7, line = 5.5)
```

Figure 53: R code for Box Plot – Amount of Rain Tomorrow

In figure 53, it is the R code to create a box plot displaying the Amount of Rain Tomorrow. The reason for using a box plot is because a box plot is an excellent choice for providing helpful information such as mean, median and range.

Based on figure 53, the variable “duplicate_weather5” contains a copy of the dataset weather, but it is filtered to remove “No” data in variable “RainTomorrow” using the piping method. The par() function is used to set or query graphical parameters. It will allow space for the details that are listed below the graph. Variable “sdata7” are used to store a summary of variable “RISK_MM” using the summary() function. The variable “summaryStat7” will contain the reformatted data that will display the data in one line. The boxplot() function is used to create the box plot with the title name, x-axis names, y-axis names, colour and many more. Last but not least, the title() function is used to display the summary data below the graph.

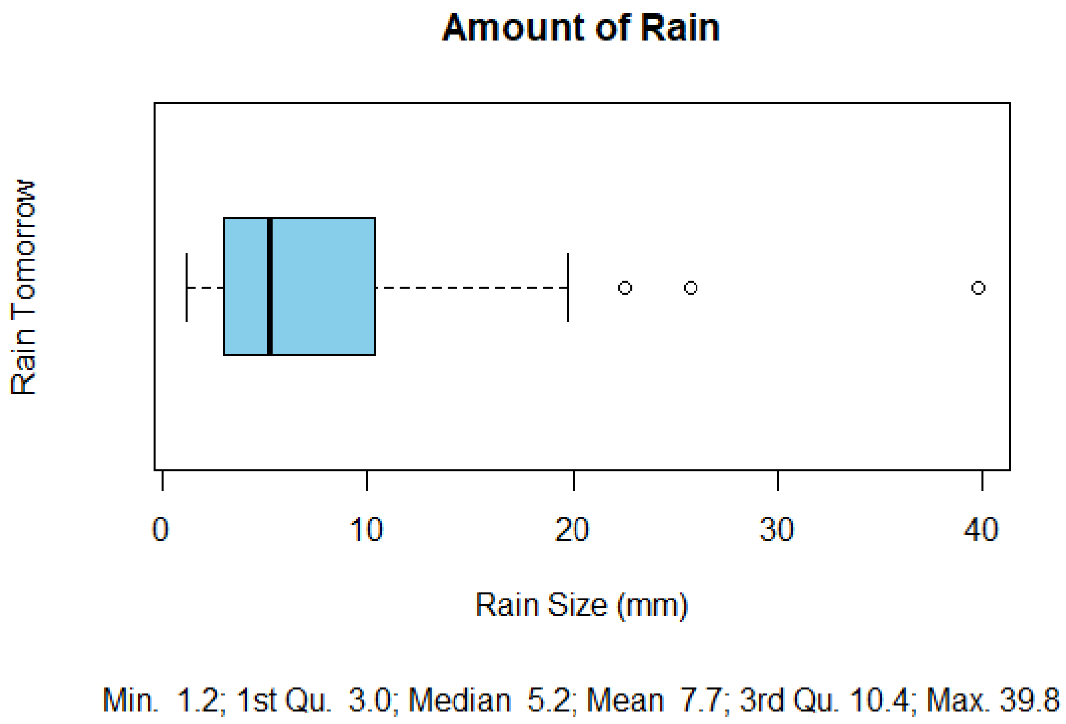


Figure 54: Box plot – Amount of Rain Tomorrow

Based on figure 54, which represent the amount of rain tomorrow in a box plot. From the figure, we can indicate that the mean rain intensity in a 24 hour period is 7.7 mm. This could mean that it will have to be pretty typical for a continuously light-to-moderate rainy day if it rains (Petty, 2020). In conclusion, tourist can still visit places in the United States, but they are not advised to do any outdoor sports. Tourist is also recommended to bring an umbrella just in case it rains.

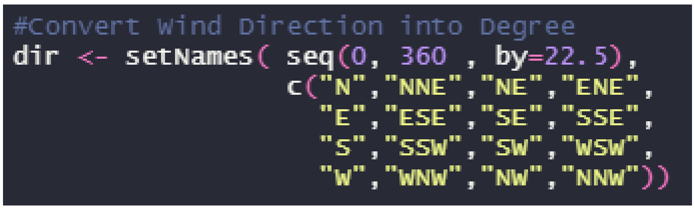
Conclusion

Based on question 5, tourist can plan to do some outdoor activity tomorrow as the probability of raining tomorrow is low based on analysis 5.1. Furthermore, if it rains, tourist is advised to bring an umbrella as it indicates that the rain will be a continuously light-to-moderate rainy day, based on analysis 5.2.

Extra Features

Extra Features 1

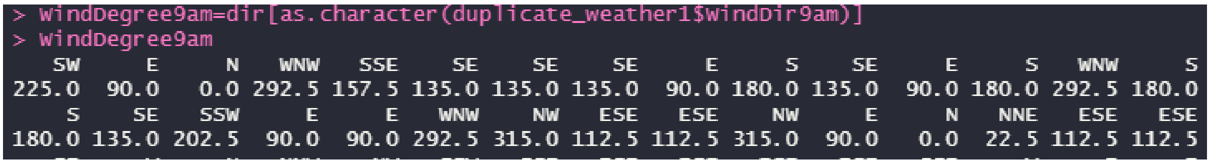
Convert Wind Direction into Wind Degree



```
#Convert Wind Direction into Degree
dir <- setNames( seq(0, 360 , by=22.5),
                 c("N","NNE","NE","ENE",
                   "E","ESE","SE","SSE",
                   "S","SSW","SW","WSW",
                   "W","WNW","NW","NNW"))
```

Figure 55: R Code to convert Wind Direction into Wind Degree

Example use of the function code



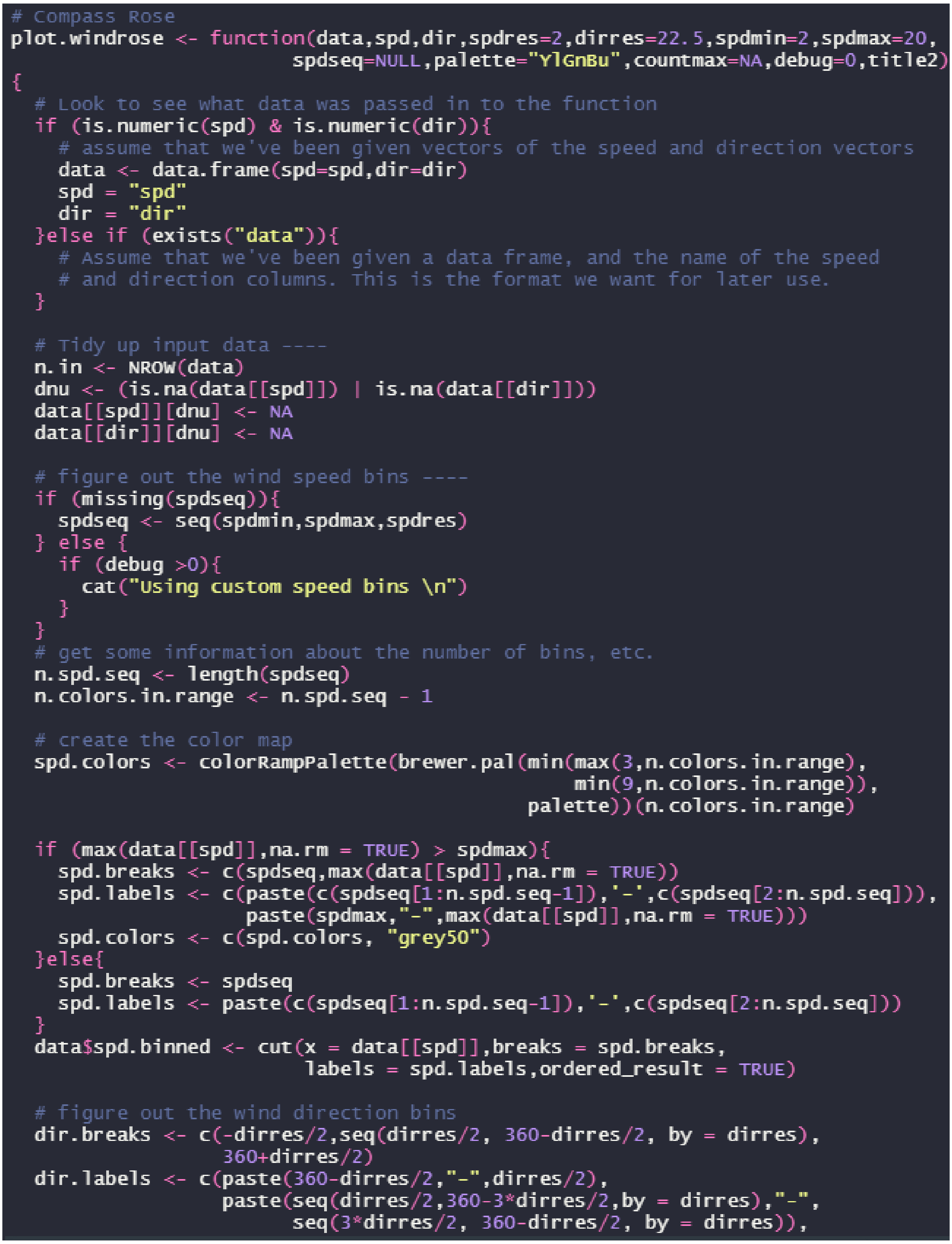
```
> WindDegree9am=dir[as.character(duplicate_weather1$WindDir9am)]
> WindDegree9am
   SW     E     N   WNW   SSE    SE    SE    SE     E     S    SE     E     S   WNW     S 
225.0  90.0   0.0 292.5 157.5 135.0 135.0 135.0  90.0 180.0 135.0  90.0 180.0 292.5 180.0 
    S    SE   SSW     E     E   WNW    NW   ESE   ESE    NW     E     N   NNE   ESE   ESE 
180.0 135.0 202.5  90.0  90.0 292.5 315.0 112.5 112.5 315.0  90.0   0.0  22.5 112.5 112.5 
```

Figure 56: Output after converting Wind Direction into Wind Degree

The function for this is to convert the current data of Wind Direction to Wind Degree. This function is crucial as the compass rose graph only accept the wind direction in degree format instead of cardinal format.

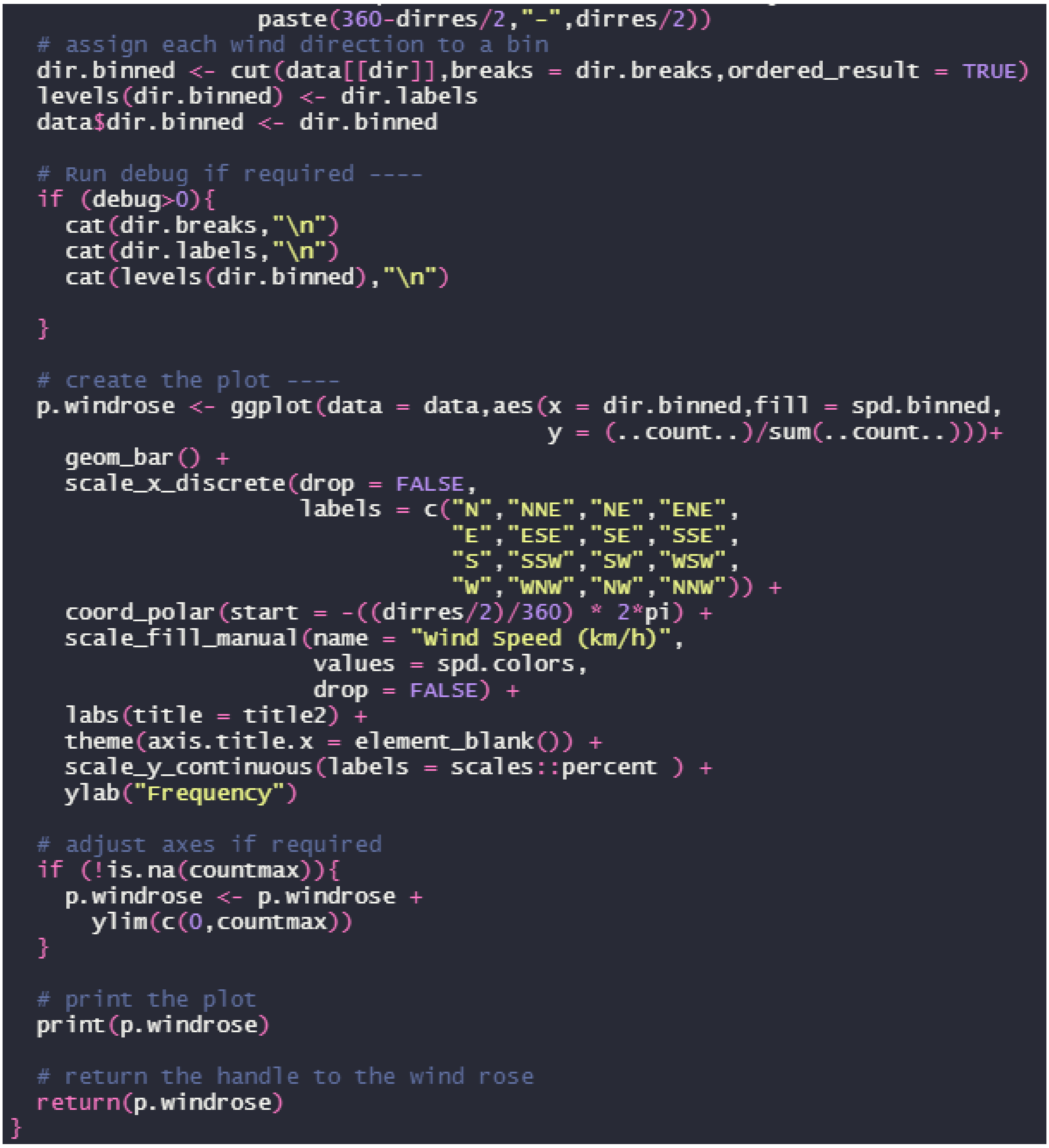
Extra Features 2

Compass Rose Function R Code



```
# Compass Rose
plot.windrose <- function(data,spd,dir,spdres=2,dirres=22.5,spdmin=2,spdmax=20,
                        spdseq=NULL,palette="YlGnBu",countmax=NA,debug=0,title2)
{
  # Look to see what data was passed in to the function
  if (is.numeric(spd) & is.numeric(dir)){
    # assume that we've been given vectors of the speed and direction vectors
    data <- data.frame(spd=spd,dir=dir)
    spd = "spd"
    dir = "dir"
  }else if (exists("data")){
    # Assume that we've been given a data frame, and the name of the speed
    # and direction columns. This is the format we want for later use.
  }

  # Tidy up input data ----
  n.in <- NROW(data)
  dnu <- (is.na(data[[spd]]) | is.na(data[[dir]]))
  data[[spd]][dnu] <- NA
  data[[dir]][dnu] <- NA

  # figure out the wind speed bins ----
  if (missing(spdseq)){
    spdseq <- seq(spdmin,spdmax,spdres)
  } else {
    if (debug >0){
      cat("Using custom speed bins \n")
    }
  }
  # get some information about the number of bins, etc.
  n.spd.seq <- length(spdseq)
  n.colors.in.range <- n.spd.seq - 1

  # create the color map
  spd.colors <- colorRampPalette(brewer.pal(min(max(3,n.colors.in.range),
                                                min(9,n.colors.in.range)),
                                            palette))(n.colors.in.range)

  if (max(data[[spd]],na.rm = TRUE) > spdmax){
    spd.breaks <- c(spdseq,max(data[[spd]],na.rm = TRUE))
    spd.labels <- c(paste(c(spdseq[1:n.spd.seq-1]),'-',c(spdseq[2:n.spd.seq])),
                    paste(spdmax,"-",max(data[[spd]],na.rm = TRUE)))
    spd.colors <- c(spd.colors, "grey50")
  }else{
    spd.breaks <- spdseq
    spd.labels <- paste(c(spdseq[1:n.spd.seq-1]),'-',c(spdseq[2:n.spd.seq]))
  }
  data$spd.binned <- cut(x = data[[spd]],breaks = spd.breaks,
                         labels = spd.labels,ordered_result = TRUE)

  # figure out the wind direction bins
  dir.breaks <- c(-dirres/2,seq(dirres/2, 360-dirres/2, by = dirres),
                  360+dirres/2)
  dir.labels <- c(paste(360-dirres/2,"-",dirres/2),
                  paste(seq(dirres/2,360-3*dirres/2,by = dirres),"-",
                        seq(3*dirres/2, 360-dirres/2, by = dirres)),
```

Figure 57: R Code to create a compass rose graph – part 1 (Clifton, 2017)



```
                  paste(360-dirres/2,"-",dirres/2))
  # assign each wind direction to a bin
  dir.binned <- cut(data[[dir]],breaks = dir.breaks,ordered_result = TRUE)
  levels(dir.binned) <- dir.labels
  data$dir.binned <- dir.binned

  # Run debug if required ----
  if (debug>0){
    cat(dir.breaks,"\n")
    cat(dir.labels,"\n")
    cat(levels(dir.binned),"\n")

  }

  # create the plot ----
  p.windrose <- ggplot(data = data,aes(x = dir.binned,fill = spd.binned,
                                       y = (..count..)/sum(..count..)))+
    geom_bar() +
    scale_x_discrete(drop = FALSE,
                     labels = c("N","NNE","NE","ENE",
                                "E","ESE","SE","SSE",
                                "S","SSW","SW","WSW",
                                "W","WNW","NW","NNW")) +
    coord_polar(start = -((dirres/2)/360) * 2*pi) +
    scale_fill_manual(name = "Wind Speed (km/h)",
                      values = spd.colors,
                      drop = FALSE) +
    labs(title = title2) +
    theme(axis.title.x = element_blank()) +
    scale_y_continuous(labels = scales::percent ) +
    ylab("Frequency")

  # adjust axes if required
  if (!is.na(countmax)){
    p.windrose <- p.windrose +
      ylim(c(0,countmax))
  }

  # print the plot
  print(p.windrose)

  # return the handle to the wind rose
  return(p.windrose)
}
```

Figure 58: R Code to create a compass rose graph – part 2 (Clifton, 2017)

Example of Compass Rose in R

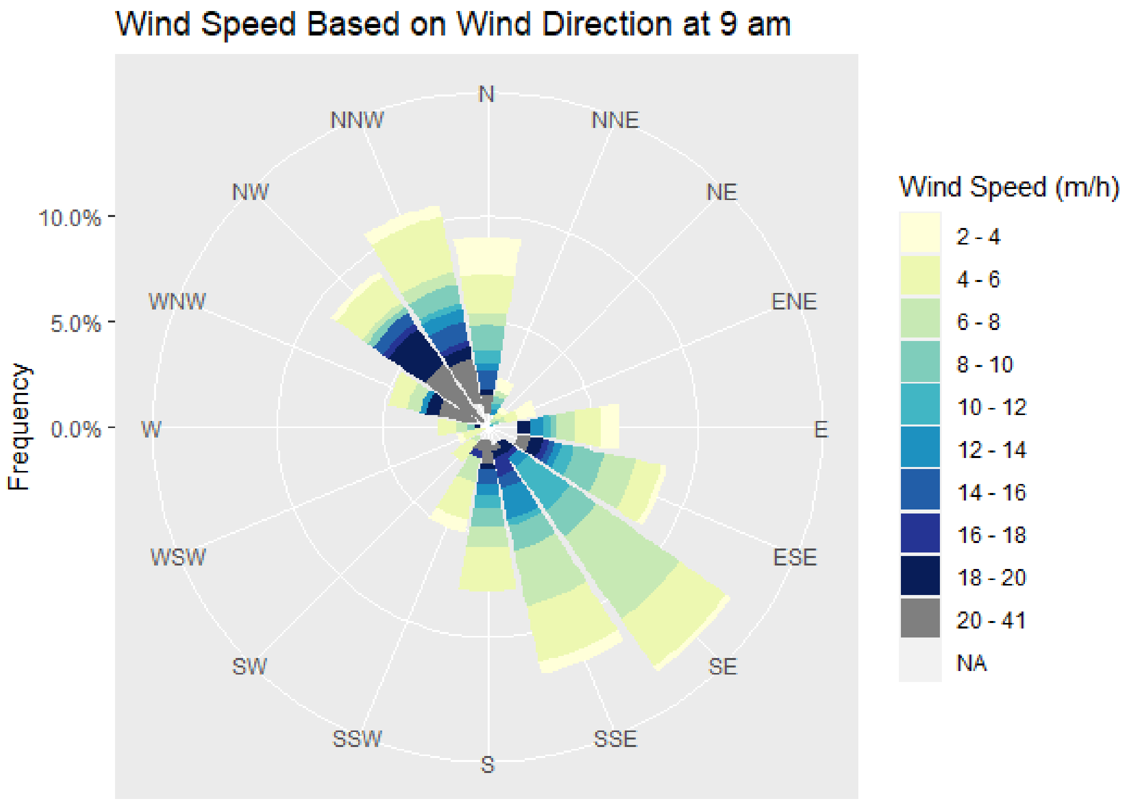
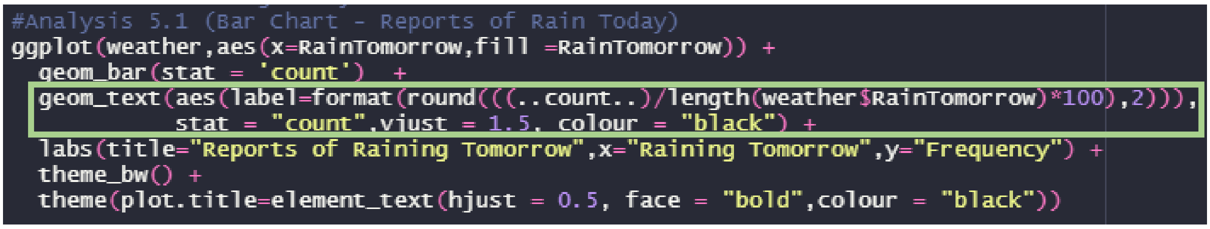


Figure 59: Compass Rose Graph to create a compass rose graph

The addition of this function allows better visualisation of the graph to understand the wind speed and wind direction of a specific area and time.

Extra Features 3

Addition of statistics in bar chart



```r
#Analysis 5.1 (Bar Chart - Reports of Rain Today)
ggplot(weather,aes(x=RainTomorrow,fill =RainTomorrow)) +
  geom_bar(stat = 'count')  +
  geom_text(aes(label=format(round(((..count..)/length(weather$RainTomorrow)*100),2))),
            stat = "count",vjust = 1.5, colour = "black") +
  labs(title="Reports of Raining Tomorrow",x="Raining Tomorrow",y="Frequency") +
  theme_bw() +
  theme(plot.title=element_text(hjust = 0.5, face = "bold",colour = "black"))
```

Figure 60: R code for statistical bar chart for the report of raining tomorrow

Based on figure 60, the geom_text() function is crucial for displaying summary statistics displayed in the bar chart. For this function, it will show the percentage in the bar chart. The figure below is an example of a bar chart with statistics in the bar chart

Example of Bar Chart with statistics in R

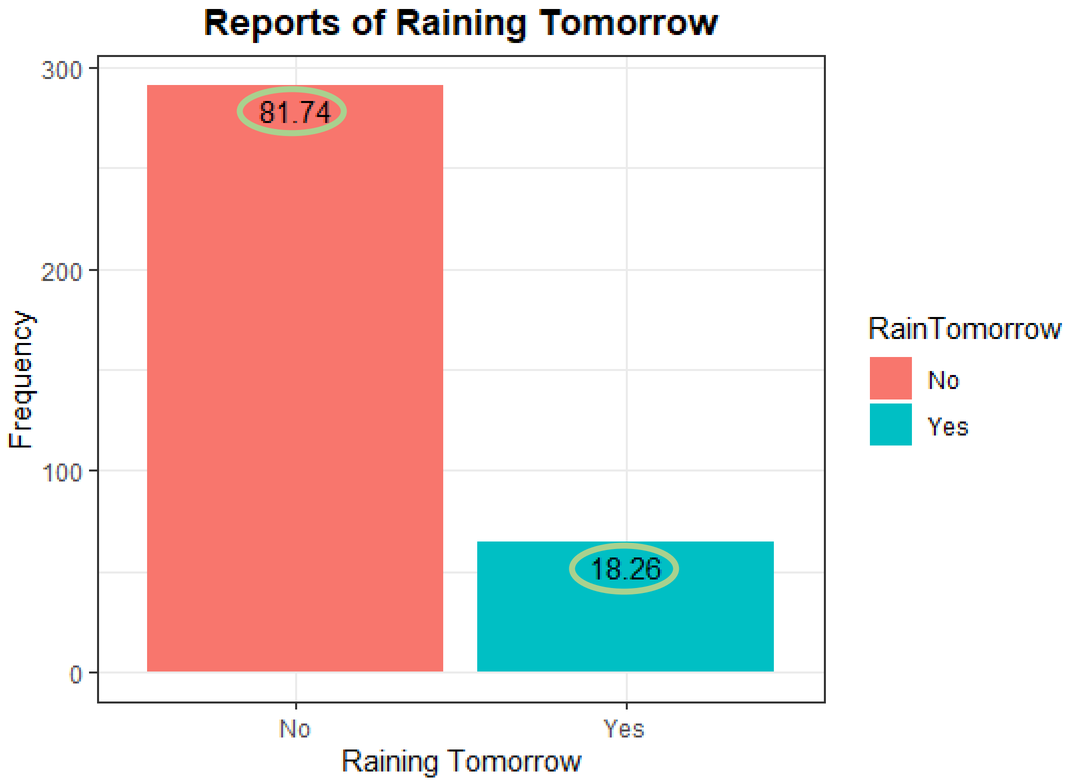
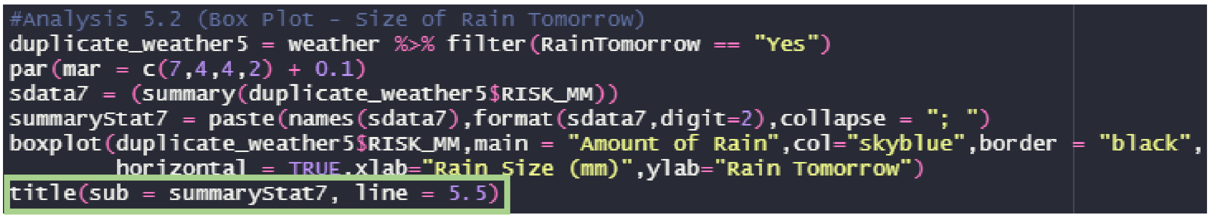


Figure 61: Statistical value in the bar chart

This function is crucial as it allows more precise prediction for the user. It can give an accurate percentage of the report in the bar chart. This is because, without this function, it will be hard to see the actual percentage of the bar chart.

Extra Features 4



```
#Analysis 5.2 (Box Plot - Size of Rain Tomorrow)
duplicate_weather5 = weather %>% filter(RainTomorrow == "Yes")
par(mar = c(7,4,4,2) + 0.1)
sdata7 = (summary(duplicate_weather5$RISK_MM))
summaryStat7 = paste(names(sdata7),format(sdata7,digit=2),collapse = "; ")
boxplot(duplicate_weather5$RISK_MM,main = "Amount of Rain",col="skyblue",border = "black",
        horizontal = TRUE,xlab="Rain Size (mm)",ylab="Rain Tomorrow")
title(sub = summaryStat7, line = 5.5)
```

Figure 62: R code for summary statistical for box plot

The title() function is used to display the summary statistics below the box plot graph. The figure below is an example of the use of this function in a box plot graph.

Example of summary statistics in box plot in R

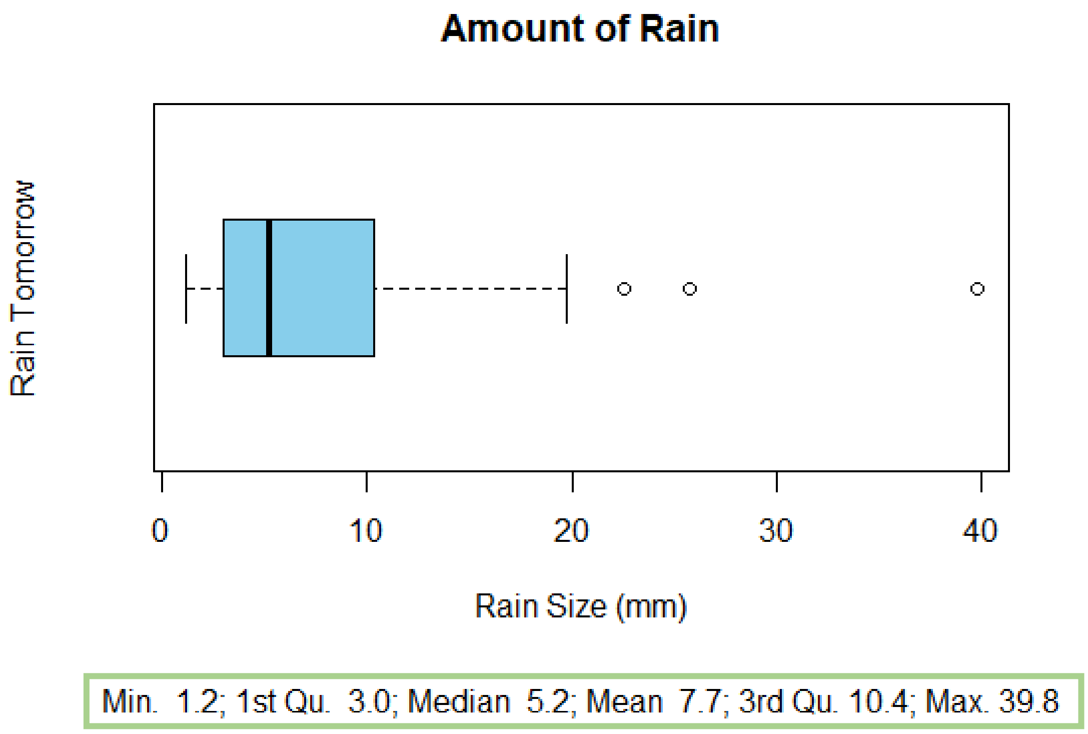


Figure 63: Summary Statistics below box plot graph in R

This function is crucial as it allows more precise prediction for the user. It can give an accurate value of the box plot. This is because, without this function, it will be hard to see the actual value of the box plot.

References

Admin, 2016. *KIA.* [Online]
Available at: https://www.mikemurphykia.com/blog/how-does-humidity-affect-car-performance/#:~:text=Overall%2C%20an%20engine%20can%20feel,and%20out%20through%20your%20exhaust.
[Accessed 12 May 2021].

Anon., 20114. *UCSB ScienceLine.* [Online]
Available at: http://scienceline.ucsb.edu/getkey.php?key=4580
[Accessed 22 May 2021].

Anon., 2015. *Golf News.* [Online]
Available at: https://www.shipsticks.com/blog/is-it-a-good-day-for-golf-5-tips-for-picking-the-best-time-to-play/
[Accessed 13 May 2021].

Anon., 2018. *AMG PETRONAS FORMULA ONE TEAM.* [Online]
Available at: https://www.mercedesamgf1.com/en/news/2018/04/insight-temperature-talk/#:~:text=Two%20temperatures%20play%20an%20important,air%20temperature%20and%20track%20temperature.&text=Variations%20in%20temperature%20impact%20many,level%20and%20the%20degradation%20
[Accessed 12 May 2021].

Anon., 2019. *Topper Learning.* [Online]
Available at: https://www.topperlearning.com/answer/why-is-noon-hotter-than-morning/6dyy4gcc
[Accessed 14 May 2021].

Anon., n.d. *Center For Science Education.* [Online]
Available at: https://scied.ucar.edu/learning-zone/clouds/how-clouds-form#:~:text=Water%20vapor%20gets%20into%20air,and%20is%20under%20less%20pressure.&text=The%20vapor%20becomes%20small%20water,and%20a%20cloud%20is%20formed.
[Accessed 14 May 2021].

Anon., n.d. *Current Results.* [Online]
Available at: https://www.currentresults.com/Weather/US/average-state-temperatures-in-winter.php
[Accessed 4 May 2021].

Anon., n.d. *National Weather Service.* [Online]
Available at:
https://www.weather.gov/lmk/humidity#:~:text=In%20general%2C%20assuming%20the%20dewpoint,the%20air%20temperature%20is%20highest.
[Accessed 14 May 2021].

Anon., n.d. *Pirelli.* [Online]
Available at: https://www.pirelli.com/tires/en-us/motorsport/f1/tires
[Accessed 12 May 2021].

Anon., n.d. *SciJinks.* [Online]
Available at:
https://scijinks.gov/rain/#:~:text=Clouds%20are%20made%20of%20water,fall%20to%20Earth%20as%20rain.
[Accessed 16 May 2021].

Anon., n.d. *Seasons of the Year.* [Online]
Available at: https://seasonsyear.com/USA
[Accessed 7 May 2021].

Bandari, A., 2020. *AIP Scilight.* [Online]
Available at: Extra Features/Functions 1
[Accessed 15 May 2021].

Cappucci, M., 2018. *The Washington Post.* [Online]
Available at: https://www.washingtonpost.com/news/capital-weather-gang/wp/2018/07/27/sometimes-its-too-hot-for-airplanes-to-fly-heres-why/
[Accessed 19 May 2021].

Clifton, A., 2017. *Stack Overflow.* [Online]
Available at: https://stackoverflow.com/questions/17266780/wind-rose-with-ggplot-r
[Accessed 28 May 2021].

Doyle, H., 2021. *ClimateKids.* [Online]
Available at: https://climatekids.nasa.gov/cloud-climate/#:~:text=Clouds%20can%20block%20light%20and,that%20heat%20from%20the%20Sun.
[Accessed 10 May 2021].

Dunn, M. G., 2011. *National Geographic.* [Online]
Available at:
https://www.nationalgeographic.org/encyclopedia/fog/#:~:text=Fog%20happens%20when%20it's%20very,around%20these%20microscopic%20solid%20particles.
[Accessed 10 May 2021].

Halblaub, J., 2014. *NOAA's National Weather Service.* [Online]
Available at: https://www.weather.gov/media/publications/front/14feb-front.pdf
[Accessed 10 May 2021].

Law, C., 2020. *THE NATIONAL LAW REVIEW.* [Online]
Available at: https://www.natlawreview.com/article/most-common-causes-aviation-accidents
[Accessed 17 May 2021].

Noble, J., 2017. *AUTOSPORT.* [Online]
Available at: https://www.autosport.com/f1/news/mercedes-driver-valtteri-bottas-says-wind-affects-2017-f1-cars-more-5020736/5020736/
[Accessed 12 May 2021].

O'MARA, K., 2019. *myfitnesspal.* [Online]
Available at: https://blog.myfitnesspal.com/whats-the-ideal-temperature-for-optimal-training/
[Accessed 10 May 2021].

Petty, G. W., 2020. *Quora.* [Online]
Available at: https://www.quora.com/What-does-8-mm-to-10-mm-rainfall-mean-in-layman-terms-Do-you-consider-it-as-a-light-or-medium-or-heavy-rain-day
[Accessed 16 May 2021].

RACERS, F., n.d. *FLOW RACERS.* [Online]
Available at: https://flowracers.com/blog/how-f1-drivers-stay-cool/
[Accessed 12 May 2021].

R-Project, n.d. *R-Project.* [Online]
Available at: https://www.r-project.org/about.html
[Accessed 22 May 2021].

SKILLING, T., 2014. *Chicago Tribute.* [Online]
Available at: https://www.chicagotribune.com/weather/ct-wea-1005-asktom-20141004-column.html
[Accessed 13 May 2021].

Skilling, T., 2019. *WGN9.* [Online]
Available at: https://wgntv.com/weather/why-does-it-always-seem-to-be-windier-during-the-day-than-at-night/#:~:text=Much%20of%20the%20tendency%20for,the%20air%20immediately%20above%20it.&text=Sun%2Dinduced%20heating%20disappears%20with,of%20darkness%2C%20and%20winds%20
[Accessed 12 May 2021].