Table of Contents

INTRODUCTION

In this weather dataset, we are going to investigate and experiment with different data values to answer the following questions:

1. What are the factors that affect implementation and location sites of wind turbines?
2. What are the factors that affect the installation and location sites of solar panels?
3. What are the factors that cause natural disasters?
4. What are the factors that determine agricultural location and choices?
5. What are the factors that affect weather prediction?

DATA IMPORT

```
#LOE HUI LIN
#TP060359
#-----------------------------------------------------------------------------------------------------------------
#READING WEATHER DATASET
weather<-read.csv(file = "C:\\Users\\Loe Hui Lin\\Desktop\\PDFA\\Assignment\\weather.csv", header=TRUE, sep=",")
View(weather)
```

Figure 1.0 Importing data

As shown in *Figure 1.0*, the variable *weather* is assigned to the generated data frame containing data from the weather dataset which is read from the provided specific file path with the help of the built-in *read.csv()* function. The *header* parameter is set to *TRUE* to get the first row of values in the dataset as column names (which will be the variable names for data frame as shown in further examples later); while the *sep* parameter is set to a value of "," as the file to be read is of csv (comma-separated values file) type with the default delimiter of a comma (Swcarpentry.github.io., 2021). The *View()* function is later used as a means to invoke a spreadsheet-style data viewer of the *weather* variable for data frame content checking to ensure the csv file is properly read (McPherson, 2021).

DATA CLEANING

Checking for Missing Data

```
#----------------------------------------
#DATA CLEANING (HANDLING MISSING DATA)
#----------------------------------------
#CHECK FOR MISSING VALUES (NA)
summary(weather)

#Sunshine NA: 3
#WindGustSpeed NA: 2
#WindSpeed9am NA: 7

length(which(is.na(weather$WindGustDir)))
length(which(is.na(weather$WindDir9am)))
length(which(is.na(weather$WindDir3pm)))

#WindGustDir NA: 3
#WindDir9am NA: 31
#WindDir3pm NA: 1
```

Figure 2.0 Checking for missing data

The *summary()* function is used to provide a brief statistical information rundown in each dataset column to facilitate checking process for missing values. The returned result is that *Sunshine*, *WindGustSpeed* and *WindSpeed9am* columns contained missing numerical values. On the other hand, the *length()* function is used to check for missing categorical values in dataset columns such as *WindGustDir*, *WindDir9am* and *WindDir3pm*.

```
> summary(weather)
    MinTemp          MaxTemp         Rainfall       Evaporation        Sunshine      WindGustDir
 Min.   :-5.300   Min.   : 7.60   Min.   : 0.000   Min.   : 0.200   Min.   : 0.000   Length:366
 1st Qu.: 2.300   1st Qu.:15.03   1st Qu.: 0.000   1st Qu.: 2.200   1st Qu.: 5.950   Class :character
 Median : 7.450   Median :19.65   Median : 0.000   Median : 4.200   Median : 8.600   Mode  :character
 Mean   : 7.266   Mean   :20.55   Mean   : 1.428   Mean   : 4.522   Mean   : 7.909
 3rd Qu.:12.500   3rd Qu.:25.50   3rd Qu.: 0.200   3rd Qu.: 6.400   3rd Qu.:10.500
 Max.   :20.900   Max.   :35.80   Max.   :39.800   Max.   :13.800   Max.   :13.600
                                                                    NA's   :3

 WindGustSpeed    WindDir9am         WindDir3pm         WindSpeed9am     WindSpeed3pm     Humidity9am
 Min.   :13.00   Length:366         Length:366         Min.   : 0.000   Min.   : 0.00   Min.   :36.00
 1st Qu.:31.00   Class :character   Class :character   1st Qu.: 6.000   1st Qu.:11.00   1st Qu.:64.00
 Median :39.00   Mode  :character   Mode  :character   Median : 7.000   Median :17.00   Median :72.00
 Mean   :39.84                                         Mean   : 9.652   Mean   :17.99   Mean   :72.04
 3rd Qu.:46.00                                         3rd Qu.:13.000   3rd Qu.:24.00   3rd Qu.:81.00
 Max.   :98.00                                         Max.   :41.000   Max.   :52.00   Max.   :99.00
 NA's   :2                                             NA's   :7
  Humidity3pm     Pressure9am      Pressure3pm       Cloud9am        Cloud3pm        Temp9am
 Min.   :13.00   Min.   : 996.5   Min.   : 996.8   Min.   :0.000   Min.   :0.000   Min.   : 0.100
 1st Qu.:32.25   1st Qu.:1015.4   1st Qu.:1012.8   1st Qu.:1.000   1st Qu.:1.000   1st Qu.: 7.625
 Median :43.00   Median :1020.1   Median :1017.4   Median :3.500   Median :4.000   Median :12.550
 Mean   :44.52   Mean   :1019.7   Mean   :1016.8   Mean   :3.891   Mean   :4.025   Mean   :12.358
 3rd Qu.:55.00   3rd Qu.:1024.5   3rd Qu.:1021.5   3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:17.000
 Max.   :96.00   Max.   :1035.7   Max.   :1033.2   Max.   :8.000   Max.   :8.000   Max.   :24.700

    Temp3pm       RainToday           RISK_MM       RainTomorrow
 Min.   : 5.10   Length:366         Min.   : 0.000   Length:366
 1st Qu.:14.15   Class :character   1st Qu.: 0.000   Class :character
 Median :18.55   Mode  :character   Median : 0.000   Mode  :character
 Mean   :19.23                      Mean   : 1.428
 3rd Qu.:24.00                      3rd Qu.: 0.200
 Max.   :34.50                      Max.   :39.800
```

Missing numerical data

```
> length(which(is.na(weather$WindGustDir)))
[1] 3
> length(which(is.na(weather$WindDir9am)))
[1] 31
> length(which(is.na(weather$WindDir3pm)))
[1] 1
```

Missing categorical data

Figure 3.0 Missing numerical and categorical data

Handling Missing Data

```
#-------------------------------------------------------------------------------------------------------------
#FILLING IN MISSING VALUES WITH MEAN VALUE(NA)
#filling in missing Sunshine values
Sunshine_Fill <- weather$Sunshine
Sunshine_Fill[is.na(Sunshine_Fill)] <- round(mean(Sunshine_Fill,na.rm=TRUE), digits = 1)
#to show NA values are replaced in Sunshine
Sunshine_Fill2 <- table(is.na(Sunshine_Fill))
Sunshine_Fill2

#filling in missing WindGustSpeed values
WindGustSpeed_Fill <- weather$WindGustSpeed
WindGustSpeed_Fill[is.na(WindGustSpeed_Fill)] <- round(mean(WindGustSpeed_Fill,na.rm=TRUE), digits = 1)
#to show NA values are replaced in WindGustSpeed
WindGustSpeed_Fill2 <- table(is.na(WindGustSpeed_Fill))
WindGustSpeed_Fill2

#filling in missing WindSpeed9am values
WindSpeed9am_Fill <- weather$WindSpeed9am
WindSpeed9am_Fill[is.na(WindSpeed9am_Fill)] <- round(mean(WindSpeed9am_Fill,na.rm=TRUE), digits = 1)
#to show NA values are replaced in WindSpeed9am
WindSpeed9am_Fill2 <- table(is.na(WindSpeed9am_Fill))
WindSpeed9am_Fill2
```

Figure 4.0 Handling missing data with mean value

The function of *is.na()* is used to check for missing values in each dataset columns which will be determined if the program returns a value of *TRUE* or *FALSE* (*TRUE* if there is existing missing value, *FALSE* if there are none). All of the previously mentioned dataset columns containing the missing values are replaced with the mean values of their own respective columns by using the *mean()* function. The *round()* function is also used to facilitate better data visibility by rounding off all values to one decimal place as indicated by the value set in the *digits* parameter of the *round()* function. After replacing missing values, the *is.na()* and *table()* function is used to display frequency of any existing NA values in the form of table as shown in *Figure 5.0* (GeeksforGeeks, 2020).

```
> #FILLING IN MISSING VALUES WITH MEAN VALUE(NA)
> #filling in missing Sunshine values
> Sunshine_Fill <- weather$Sunshine
> Sunshine_Fill[is.na(Sunshine_Fill)] <- round(mean(Sunshine_Fill,na.rm=TRUE), digits = 1)
> #to show NA values are replaced in Sunshine
> Sunshine_Fill2 <- table(is.na(Sunshine_Fill))
> Sunshine_Fill2

FALSE
  366

> #filling in missing WindGustSpeed values
> WindGustSpeed_Fill <- weather$WindGustSpeed
> #filling in missing WindGustSpeed values
> WindGustSpeed_Fill <- weather$WindGustSpeed
> WindGustSpeed_Fill[is.na(WindGustSpeed_Fill)] <- round(mean(WindGustSpeed_Fill,na.rm=TRUE), digits = 1)
> #to show NA values are replaced in WindGustSpeed
> WindGustSpeed_Fill2 <- table(is.na(WindGustSpeed_Fill))
> WindGustSpeed_Fill2

FALSE
  366

> #filling in missing WindSpeed9am values
> WindSpeed9am_Fill <- weather$WindSpeed9am
> WindSpeed9am_Fill[is.na(WindSpeed9am_Fill)] <- round(mean(WindSpeed9am_Fill,na.rm=TRUE), digits = 1)
> #to show NA values are replaced in WindSpeed9am
> WindSpeed9am_Fill2 <- table(is.na(WindSpeed9am_Fill))
> WindSpeed9am_Fill2

FALSE
  366
```

Figure 5.0 After handling missing numerical data

```
#-----------------------------------------------------------------------------------------------------------------
#FILLING IN MISSING VALUES WITH MODE VALUE(NA)
#filling in missing  WindGustDir values
WindGustDir_Fill <- weather$WindGustDir
WindGustDir_Fill[is.na(WindGustDir_Fill)] <- names(table(WindGustDir_Fill))[table(WindGustDir_Fill) == max(table(WindGustDir_Fill))]
#to show NA values are replaced in WindGustDir
table(WindGustDir_Fill)

#filling in missing WindDir9am values
WindDir9am_Fill <- weather$WindDir9am
WindDir9am_Fill[is.na(WindDir9am_Fill)] <- names(table(WindDir9am_Fill))[table(WindDir9am_Fill) == max(table(WindDir9am_Fill))]
#to show NA values are replaced in WindDir9am
table(WindDir9am_Fill)
```

Figure 6.0 Handling missing categorical data with mode value

Similar to the previous attempt of handling missing data, missing categorical values in columns such as *WindGustDir* and *WindDir9am* are replaced with their respective mode categorical values by using the *names()* function to set NA values to modal values with the help of *max()* function (Rdocumentation.org., 2021).

However, missing data in *WindDir3pm* dataset column is not handled (by dropping NA value using *na.omit()* function) as this will result in dataset column of having only 365 data values instead of 366, which will cause compatibility issues later on with other dataset columns that have 366 values (in data visualization part). In addition to that, *WindDir3pm* has bimodal data (containing two different mode values) hence it is impossible to replace NA with a mode value. The last justification would be the count of NA value is only **one**, hence the percentage is too trivial to cause significant fluctuations in data variation.

```
> #FILLING IN MISSING VALUES WITH MODE VALUE(NA)
> #filling in missing  WindGustDir values
> WindGustDir_Fill <- weather$WindGustDir
> WindGustDir_Fill[is.na(WindGustDir_Fill)] <- names(table(WindG
e(WindGustDir_Fill))]
> #to show NA values are replaced in WindGustDir
> table(WindGustDir_Fill)
WindGustDir_Fill
  E ENE ESE   N  NE NNE NNW  NW   S  SE SSE SSW  SW   W WNW WSW
 37  30  23  21  16   8  44  76  22  12  12   5   3  20  35   2
> #filling in missing WindDir9am values
> WindDir9am_Fill <- weather$WindDir9am
> WindDir9am_Fill[is.na(WindDir9am_Fill)] <- names(table(WindDir
ndDir9am_Fill))]
> #to show NA values are replaced in WindDir9am
> table(WindDir9am_Fill)
WindDir9am_Fill
  E ENE ESE   N  NE NNE NNW  NW   S  SE SSE SSW  SW   W WNW WSW
 22   8  29  31   4   8  36  30  27  78  40  17   7   8  16   5
```

Figure 7.0 After handling missing categorical data

Renaming Dataset Columns

```
#renaming columns after data cleaning
SS <- Sunshine_Fill
WGS <- WindGustSpeed_Fill
WS9 <- WindSpeed9am_Fill
WGD <- WindGustDir_Fill
WD9 <- WindDir9am_Fill

MinT <- weather$MinTemp
MaxT <- weather$MaxTemp
RF <- weather$Rainfall
E <- weather$Evaporation
WS3 <-weather$WindSpeed3pm
WD3 <- weather$WindDir3pm
H9 <- weather$Humidity9am
H3 <- weather$Humidity3pm
P9 <-weather$Pressure9am
P3 <- weather$Pressure3pm
C9 <- weather$Cloud9am
C3 <- weather$Cloud3pm
T9 <- weather$Temp9am
T3 <- weather$Temp3pm
RToday <- weather$RainToday
RTmr <- weather$RainTomorrow
RiskMM <- weather$RISK_MM
```

```
#creating + renaming new columns with updated values
install.packages("dplyr")
library(dplyr)
weather = mutate(weather, SS=SS)
weather = mutate(weather, WGS=WGS)
weather = mutate(weather, WS9=WS9)
weather = mutate(weather, WGD=WGD)
weather = mutate(weather, WD9=WD9)

#renaming existing columns with new names
#before renaming
colnames(weather)
weather <- weather %>% rename(MinT = MinTemp, MaxT = MaxTemp, RF = Rainfall,
                              E = Evaporation, WD3 = WindDir3pm, WS3 = WindSpeed3pm,
                              H9 = Humidity9am, H3 = Humidity3pm, P9 = Pressure9am,
                              P3 = Pressure3pm, C9 = Cloud9am, C3 = Cloud3pm,
                              T9 = Temp9am, T3 = Temp3pm, RToday = RainToday,
                              RTmr = RainTomorrow, RiskMM = RISK_MM)

#after renaming
colnames(weather)
```

Figure 8.0 Renaming and adding new dataset columns

All existing variable names are reassigned to new and abbreviated version of variable names to facilitate coding process. The "*dplyr*" package is installed and loaded to utilize the *mutate()* function so that the previously mentioned dataset columns which have gone through data cleaning could be added into the existing weather dataset as new columns with new column names. As for the unchanged columns in data cleaning process, they will be renamed (aided by the *rename()* function) with their data values unchanged in the dataset. The difference in column names before and after renaming could be shown using the *colnames()* function.

```
> #before renaming
> colnames(weather)
 [1] "MinTemp"       "MaxTemp"       "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
 [7] "WindGustSpeed" "WindDir9am"    "WindDir3pm"    "WindSpeed9am"  "WindSpeed3pm"  "Humidity9am"
[13] "Humidity3pm"   "Pressure9am"   "Pressure3pm"   "Cloud9am"      "Cloud3pm"      "Temp9am"
[19] "Temp3pm"       "RainToday"     "RISK_MM"       "RainTomorrow"  "SS"            "WGS"
[25] "WS9"           "WGD"           "WD9"

> #after renaming
> colnames(weather)
 [1] "MinT"          "MaxT"          "RF"            "E"             "Sunshine"      "WindGustDir"
 [7] "WindGustSpeed" "WindDir9am"    "WD3"           "WindSpeed9am"  "WS3"           "H9"
[13] "H3"            "P9"            "P3"            "C9"            "C3"            "T9"
[19] "T3"            "RToday"        "RiskMM"        "RTmr"          "SS"            "WGS"
[25] "WS9"           "WGD"           "WD9"
```

Figure 9.0 After renaming and adding new dataset columns

QUESTIONS & ANALYSIS

Before data visualization

Before performing data visualization, variables named as *Q1, Q2, Q3, Q4, and Q5* will be assigned newly created data frames consisting of a few selected columns from the weather dataset (combined as a vector using *c()* function) for ease of use in further analysis of the specific question. This action is achieved with the help of *data.frame()* and *subset()* functions.

Question 1: What are the factors that affect location choice for wind turbines?

Code Snippet 1.1

```
#-------------------------------------------------------------------------------------------------
#QUESTION1: WHAT ARE THE FACTORS THAT AFFECT LOCATION CHOICE FOR WIND TURBINES?
#-------------------------------------------------------------------------------------------------
#ANALYSIS1.1: Distribution of Wind Speed & Wind Direction determines location for wind turbines
#VARIABLES USED: WS9, WS3, WD3
#CONCLUSION: North direction at 9am has the highest + most of the low wind speeds
#CONCLUSION: West direction at 3pm has high consistency in average wind speed

#selecting specific columns for Question1
Q1 <- data.frame(subset (weather, select = c("WGD", "WGS", "WD9", "WD3", "WS9", "WS3",
                                             "P9", "P3", "T3", "T9")))
View(Q1)

#binning WD3 values into 4 categories: South, East, West, North
Directions <- Q1$WD3
Wind_Directions <- ifelse(Directions %in% c("S", "SE", "SSE", "SSW", "SW"), "South",
                   ifelse(Directions %in% c("E", "ENE", "ESE"), "East",
                     ifelse(Directions %in% c("W", "WNW", "WSW"), "West",
                       "North")))
table(Wind_Directions)

#plotting histogram for distribution of wind speed in different wind directions
#wind speed distribution histogram at 9am
library(ggplot2)
Q1_hist <- ggplot(Q1, aes(x = WS9, fill = Wind_Directions, color = Wind_Directions)) +
  geom_histogram(bins = 50, binwidth = 1, position = "identity", alpha = 0.5) +
                 scale_x_continuous(name = "Wind Speed at 9am (mph)",
                                    breaks = seq(0, 41, 5), limits = c(0, 41)) +
                 scale_y_continuous(name = "Count") +
                 ggtitle("Distribution of Wind Speed at 9am Across Wind Directions")
Q1_hist

#distribution of histograms for wind speed at 3pm
Q1_hist2 <- ggplot(Q1, aes(x = WS3, fill = Wind_Directions, color = Wind_Directions)) +
  geom_histogram(bins = 50, binwidth = 1, position = "identity", alpha = 0.5) +
                 scale_x_continuous(name = "Wind Speed at 3pm (mph)",
                                    breaks = seq(0, 52, 5), limits = c(0, 52)) +
                 scale_y_continuous(name = "Count") +
                 ggtitle("Distribution of Wind Speed at 3pm Across Wind Directions")
Q1_hist2
```

Figure 10.0 Code snippet for analysis 1.1

```
> table(Wind_Directions)
Wind_Directions
 East North South  West
   57   168    43    98
```

Figure 11.0 After binning wind directions

In reference to Question 1, the distribution of wind speed across all wind directions are taken into consideration as a potentially significant factor. There are a

total of 16 different wind directions which are binned and categorized into 4 major categories named as "*South*", "*East*", "*West*" and "*North*" (as shown in *Figure10.0* at line 138 – 141). This action increases data visibility (as shown in *Figure 12.0*) and facilitate the process of plotting histograms to show distribution of wind speeds across these wind directions. The histograms could be understood such that x axis is the wind speed while y axis is the count of wind speed. The bars are grouped by the previously mentioned wind directions categories.

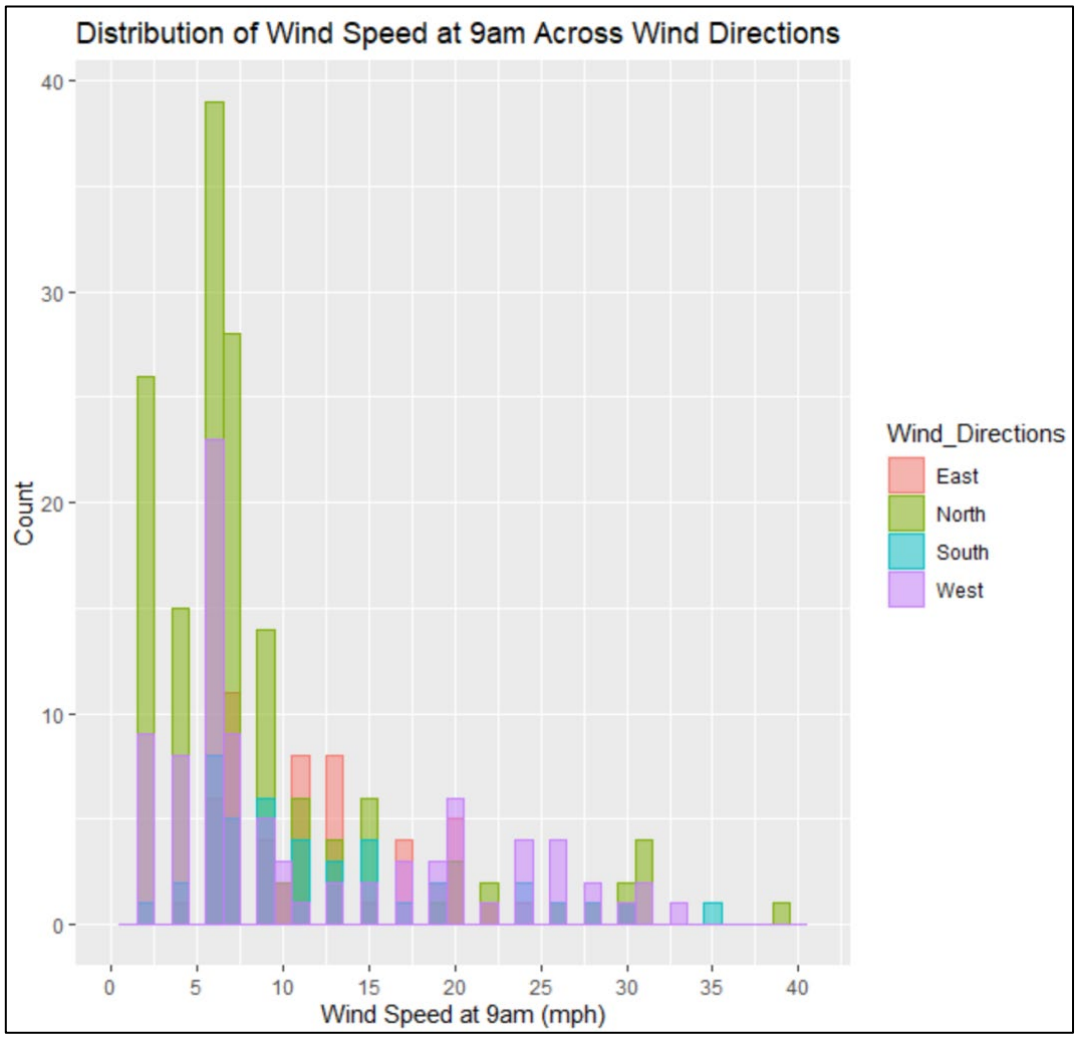

Figure 12.0 Distribution of wind speed at 9 am

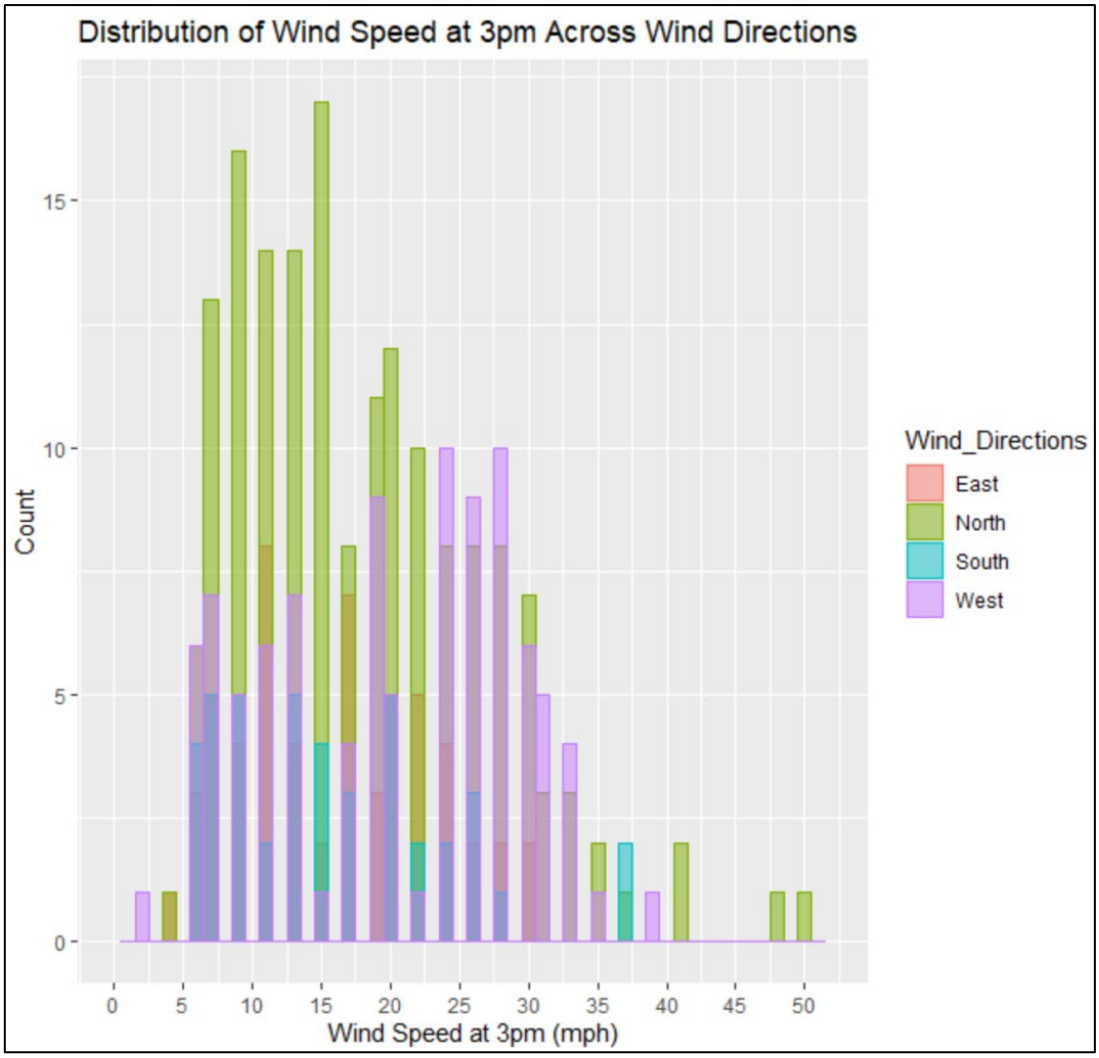

Figure 13.0 Distribution of wind speed at 3 pm

Analysis 1.1

It is observed that North winds have the highest frequency regardless of the time of the day as shown in both Figure and (approximately 39 at 9am and 17 at 3pm). However, *figure 13.0* shows that West winds have the most stable consistency in wind speed as it maintains a frequency between 5 to 10 in its higher ranged wind speeds (approximately between 5 mph to 30 mph).

Findings 1.1

Wind is a critical determinant for wind turbines as amount of harnessed wind energy and efficiency of procuring wind energy is dependent on continuous and stable inclination of wind direction and wind speed. This finding is supported by the fact that among the five major wind farms of the United States (Texas, Iowa, Oklahoma, Kansas and Illinois) which contributes as much as 58% of the total generated electricity in 2020 are mostly situated in the western regions of the country (Eia.gov., 2021). The term "installed wind power capacity" is defined as the amount of electical energy generated from wind turbines with optimal wind conditions in megawatts (Irishenvironment.com., 2021); and *Figure 14.0* has illustrated some of the highest installed wind power capacity states such as Texas and Oklahoma which are located in the Southwestern region whilst Kansas, Iowa and Illinois are located in the Midwestern region. In fact, the Roscoe Wind Farm, Horse Hollow Wind Energy Center and Capricorn Ridge are just a few examples of the world's largest utility-scale wind farms found in Texas (Constructionreviewonline.com., 2021).

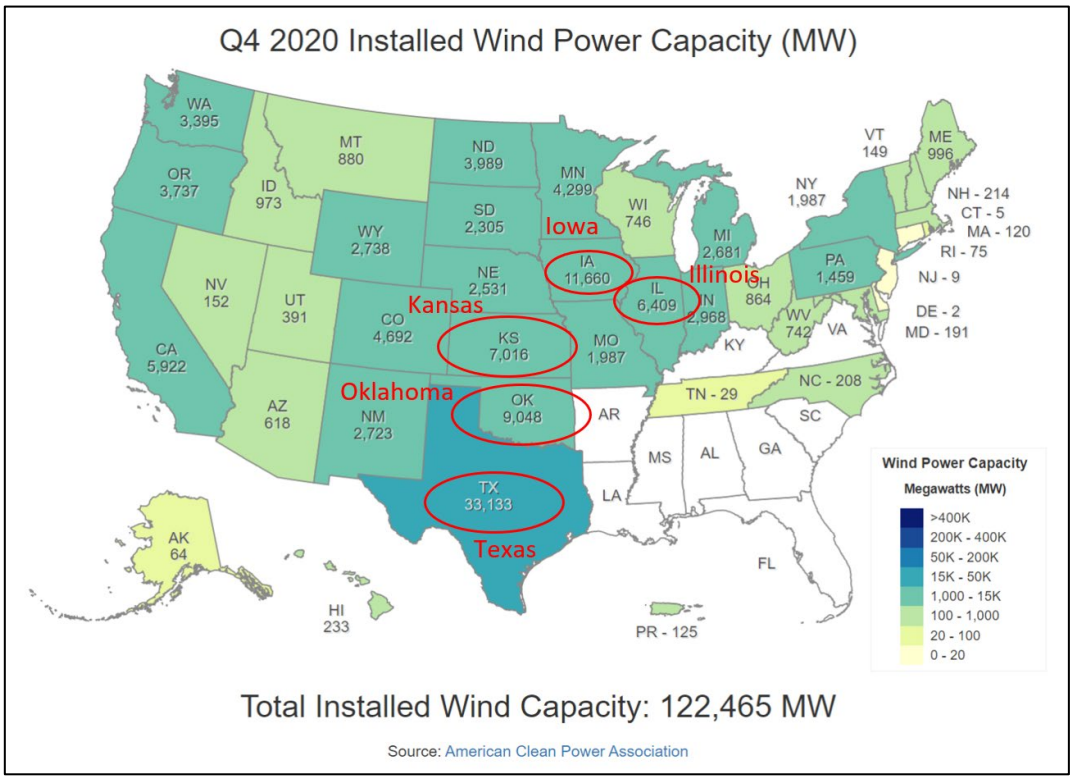

Figure 14.0 Installed Wind Power Capacity in United States (Windexchange.energy.gov., 2021)

Secondary to wind direction, wind speed consistency is another dominant factor as well. Most wind turbines built in United States are of Class 2, which are designed to adapt to medium wind environments (Eia.gov., 2021) where annual average wind speed should be approximately at 18 miles per hour (Hindawi.com., 2021). According to *Figure 13.0*, most wind speeds are distributed over the range of 5 mph to 30 mph; hence proving that the consistency of annual wind speed not only decides if wind should be the primary renewable energy source of a country, but affect the proportion difference in design of wind turbines as well (as indicated in *Figure 15.0*).

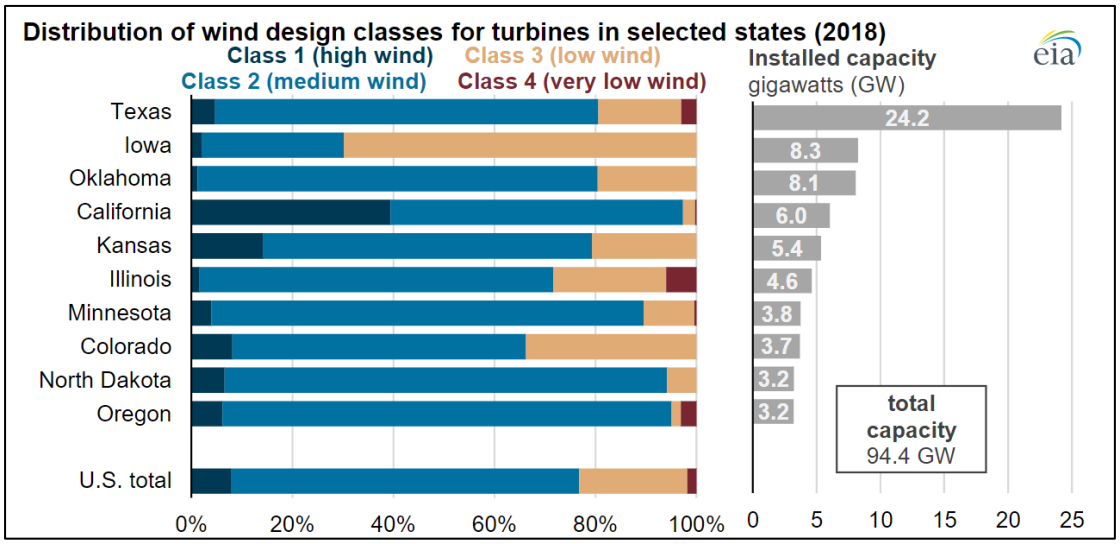

Figure 15.0 Distribution of wind design classes for wind turbines in United States (Eia.gov., 2021)

Code snippet 1.2

```
166 #-------------------------------------------------------------------------------------------------------------
167 #ANALYSIS1.2: Change in Wind Direction throughout a day determines location choice for wind turbines
168 #VARIABLES USED: WD9, WD3
169 #CONCLUSION: South East is most windy at 9am (21%)
170 #CONCLUSION: North West & West North West are most windy at 3pm (17%)
171
172 #Generating WD9am piechart
173 WD9am <- c()
174 WD9am[1] = nrow(Q1[WD9 == "N",])
175 WD9am[2] = nrow(Q1[WD9 == "S",])
176 WD9am[3] = nrow(Q1[WD9 == "E",])
177 WD9am[4] = nrow(Q1[WD9 == "W",])
178 WD9am[5] = nrow(Q1[WD9 == "SE",])
179 WD9am[6] = nrow(Q1[WD9 == "SW",])
180 WD9am[7] = nrow(Q1[WD9 == "NE",])
181 WD9am[8] = nrow(Q1[WD9 == "NW",])
182 WD9am[9] = nrow(Q1[WD9 == "ENE",])
183 WD9am[10] = nrow(Q1[WD9 == "ESE",])
184 WD9am[11] = nrow(Q1[WD9 == "NNE",])
185 WD9am[12] = nrow(Q1[WD9 == "NNW",])
186 WD9am[13] = nrow(Q1[WD9 == "SSE",])
187 WD9am[14] = nrow(Q1[WD9 == "SSW",])
188 WD9am[15] = nrow(Q1[WD9 == "WNW",])
189 WD9am[16] = nrow(Q1[WD9 == "WSW",])
190
191 WindName <- c("North","South","East","West",
192               "South East", "South West","North East", "North West",
193               "East North East", "Ease South East", "North North East",
194               "North North West", "South South East","South South West",
195               "West North West", "West South West")
196
197 Q1pie <- round(WD9am/sum(WD9am)*100)
198
199 WindName <- paste(WindName, Q1pie)
200 WindName <- paste(WindName,"%",sep = "")
201
202 pie(WD9am, labels = WindName, main="Pie chart of Wind Direction at 9am",
203     radius = 5, border = "white", col = rainbow(length(windName)))
```

Figure 16.0 Code snippet for analysis 1.2

The number of rows containing the particular wind direction in column *WD9* of *Q1* data frame (as indicated by *"N"*, *"S"*, *"E"*, *"W"*, *"SE"*, *"SW"*, *"NE"*, *"NW"*, *"ENE"*, *"ESE"*, *"NNE"*, *"NNW"*, *"SSE"*, *"SSW"*, *"WN"*, *"WSW"*) are counted using *nrow()* function. The returned values are then assigned to the vector *WD9am* in their respective index. Another vector named *WindName* is created to store the proper wind direction names and this vector will be used to label the frequency percentage of the generated pie chart later (in *Figure 16.0* at line 199-200). All values stored in the index of *WD9am* vector are divided by the total row count and multiplied with 100 to get their frequency percentage. The *pie()* function is then used to generate a pie chart to display the processed and calculated frequency percentages of all wind directions.

```
205  #check frequency of WD3 values
206  #WD3 <- factor(Question1$WD3)
207  WD3_Table <- table(factor(Q1$WD3))
208  WD3_Table
209
210  #removing NA values from WD3
211  WD3 <- na.omit(Q1$WD3)
212
213  #Generating WD3pm piechart
214  WD3pm <- c()
215  WD3pm[1] = nrow(Q1[WD3 == "N",])
216  WD3pm[2] = nrow(Q1[WD3 == "S",])
217  WD3pm[3] = nrow(Q1[WD3 == "E",])
218  WD3pm[4] = nrow(Q1[WD3 == "W",])
219  WD3pm[5] = nrow(Q1[WD3 == "SE",])
220  WD3pm[6] = nrow(Q1[WD3 == "SW",])
221  WD3pm[7] = nrow(Q1[WD3== "NE",])
222  WD3pm[8] = nrow(Q1[WD3== "NW",])
223  WD3pm[9] = nrow(Q1[WD3 == "ENE",])
224  WD3pm[10] = nrow(Q1[WD3 == "ESE",])
225  WD3pm[11] = nrow(Q1[WD3 == "NNE",])
226  WD3pm[12] = nrow(Q1[WD3 == "NNW",])
227  WD3pm[13] = nrow(Q1[WD3 == "SSE",])
228  WD3pm[14] = nrow(Q1[WD3 == "SSW",])
229  WD3pm[15] = nrow(Q1[WD3 == "WNW",])
230  WD3pm[16] = nrow(Q1[WD3 == "WSW",])
231
232  Q1pie2 <- round(WD3pm/sum(WD3pm)*100)
233
234  WindName <- paste(WindName, Q1pie2)
235  WindName <- paste(WindName,"%",sep = "")
236
237  pie(WD3pm, labels = WindName, main="Pie chart of Wind Direction at 3pm",
238      radius = 6.1, border = "white", col = rainbow(length(WindName)))
```

Figure 17.0 Code snippet for analysis 1.2

```
> #check frequency of WD3 values
> #WD3 <- factor(Question1$WD3)
> WD3_Table <- table(factor(Q1$WD3))
> WD3_Table

  E ENE ESE   N  NE NNE NNW  NW   S  SE SSE SSW  SW   W WNW WSW
 17  13  27  30  15  14  47  61  14  12   7   6   4  26  61  11
```

Figure 18.0 Checking frequency of WD3 values

Similar to the explanation given in *Figure 16.0*, the only two differences would be the usage of column *WD3* instead of *WD9*; and the removal of a single NA value (as shown in *Figure 18.0* when total frequency does not add up to 366) as performed in *Figure 17.0* at line 211 with the help of *na.omit()* function.

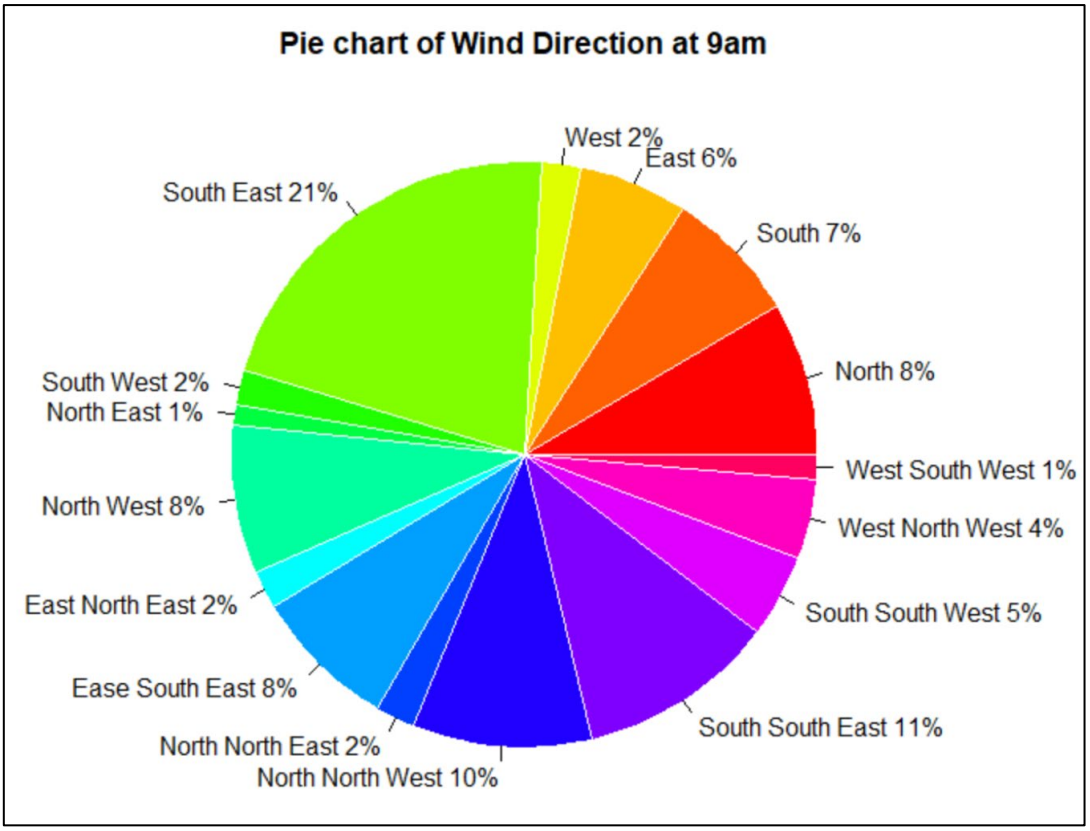

Figure 19.0 Pie chart of wind direction at 9 am

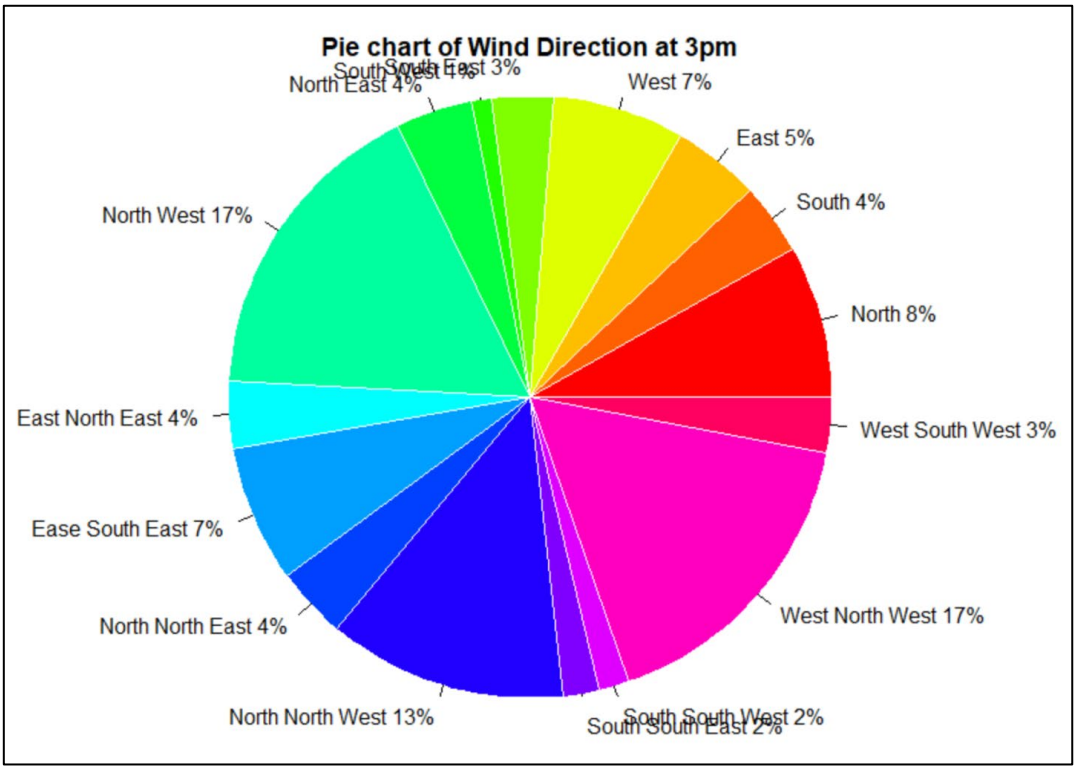

Figure 20.0 Pie chart of wind direction at 3 pm

Analysis 1.2

According to *Figure 19.0* and *Figure 20.0*, wind direction varies throughout the day between 9 am and 3 pm. As shown in the figures above, South East wind direction occupies majority of the pie chart in *Figure 19.0* with a percentage of 21% whilst North West and West North West wind directions took up a percentage of 17% in *Figure 20.0*. However, both percentages do not exceed 50%, hence showing changes in wind direction is not high.

Findings 1.2

It is concluded that the change in wind direction can bring significant impact to the working efficiency of wind turbines. The reason is that changes in directional wind shear, which is defined as the sudden shift of wind direction with height; can change the availability of air power through wind turbines as well as its ability to procure the energy from the wind. Tests on simulating directional wind shear to measure wind turbine efficiency are done and has proven that larger directional wind shear undermines wind turbine performance under different wind speeds (Gomez and Lundquist, 2019). The change in wind direction contributes to the difference in velocity between the air and turbine blades, hence causing the turbine blades to operate at compromised blade pitch angles (the optimum blade angle where generated power by wind turbines is at maximum) (A.R., Pandey, Sunil, N.S., Mugundhan, V., Velamati, 2016). Hence, it is best if changes in directional wind shear is reduced to the minimum if wind turbines are built in spaces where rate of change in wind direction is not high to optimize working efficiency of wind turbines.

Code snippet 1.3

```
#-------------------------------------------------------------------------
#ANALYSIS1.3: Temperature Affects Atmospheric Pressure
#VARIABLES USED: P3, P9, WS3, WS9
#CONCLUSION: Atmospheric pressure is directly proportional to temperature

#generating scatterplot
#AC209B = purple = Pressure at 9am
#866B86 = brown = Pressure at 3pm
cols <-c("9AM" = "#AC209B", "3PM" = "#866B86")

Q1_scatterplot <- ggplot(Q1) +
  geom_point(aes(x=T9,y=P9, colour = "9AM")) +
  geom_smooth(aes(x=T9,y=P9, colour = "9AM"), method = "lm") +
  geom_point(aes(x=T3,y=P3, colour = "3PM")) +
  geom_smooth(aes(x=T3,y=P3, colour = "3PM"), method = "lm") +
  labs(x = "Temperature (degrees)", y = "Atmosheric Pressure (hpa)",
       title = "Relation between Temperature and Atmospheric Pressure") +
  scale_colour_manual(name="Time",values=cols2,
                      guide = guide_legend(override.aes=aes(fill=NA)))
Q1_scatterplot
```

Figure 21.0 Code snippet for analysis 1.3

A scatter plot is generated to investigate and compare the relationship between temperature and atmospheric pressure at different times of the day (9am and 3pm).

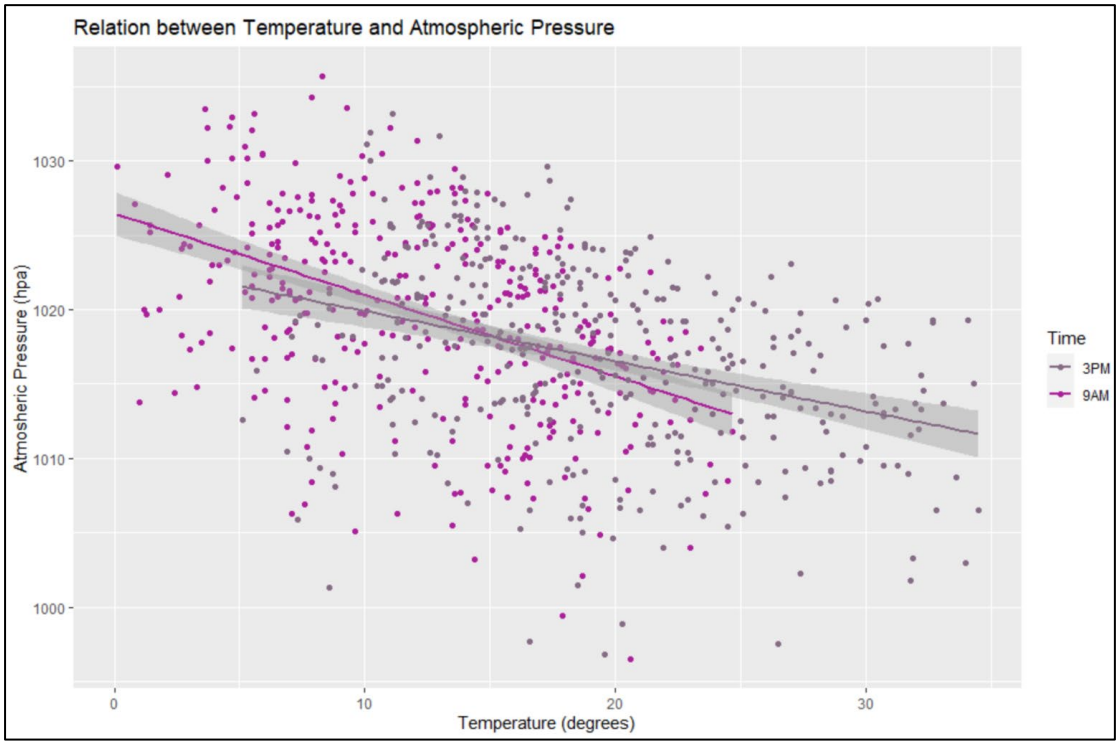

Figure 22.0 Scatter plot of relation between temperature and atmospheric pressure

Analysis 1.3

Temperature is found to be inversely proportional to atmospheric pressure regardless of different times of the day. The steep decline in both colored regression lines suggest that atmospheric pressure decreases as temperature increases. However, there is a comparatively steeper decline in the regression line representative of 9 am (purple line).

Findings 1.3

The reason why increase in temperature will cause decline in atmospheric pressure is because air molecules begin moving faster and further from one another due to increase in kinetic energy and also the fact that there are no constraints of pressure on the air molecules at high altitudes (Writer, 2021).

Code snippet 1.4

```
#-------------------------------------------------------------------------------
#ANALYSIS1.4: Atmospheric Pressure affects Wind Speed
#VARIABLES USED: P3, P9, WS3, WS9
#CONCLUSION: Wind Speed is inversely proportional to Atmospheric Pressure

#generating scatterplot
#4D9A24 = green = Wind Speed at 9am
#287CD1 = blue = Wind Speed at 3pm
cols2 <-c("9AM" = "#4D9A24", "3PM" = "#287CD1")

Q1_scatterplot2 <- ggplot(Q1) +
              geom_point(aes(x=P9,y=WS9, colour = "9AM")) +
              geom_smooth(aes(x=P9,y=WS9, colour = "9AM"), method = "lm") +
              geom_point(aes(x=P3,y=WS3, colour = "3PM")) +
              geom_smooth(aes(x=P3,y=WS3, colour = "3PM"), method = "lm") +
              labs(x = "Atmospheric pressure (hpa)", y = "Wind Speed (mph)",
                 title = "Relation between Atmospheric Pressure and Wind Speed") +
              scale_colour_manual(name="Time",values=cols,
                                  guide = guide_legend(override.aes=aes(fill=NA)))
Q1_scatterplot2
```

Figure 23.0 Code snippet for analysis 1.4

A scatter plot is generated to investigate and compare the relationship between atmospheric pressure and wind speed at different times of the day (9am and 3pm).

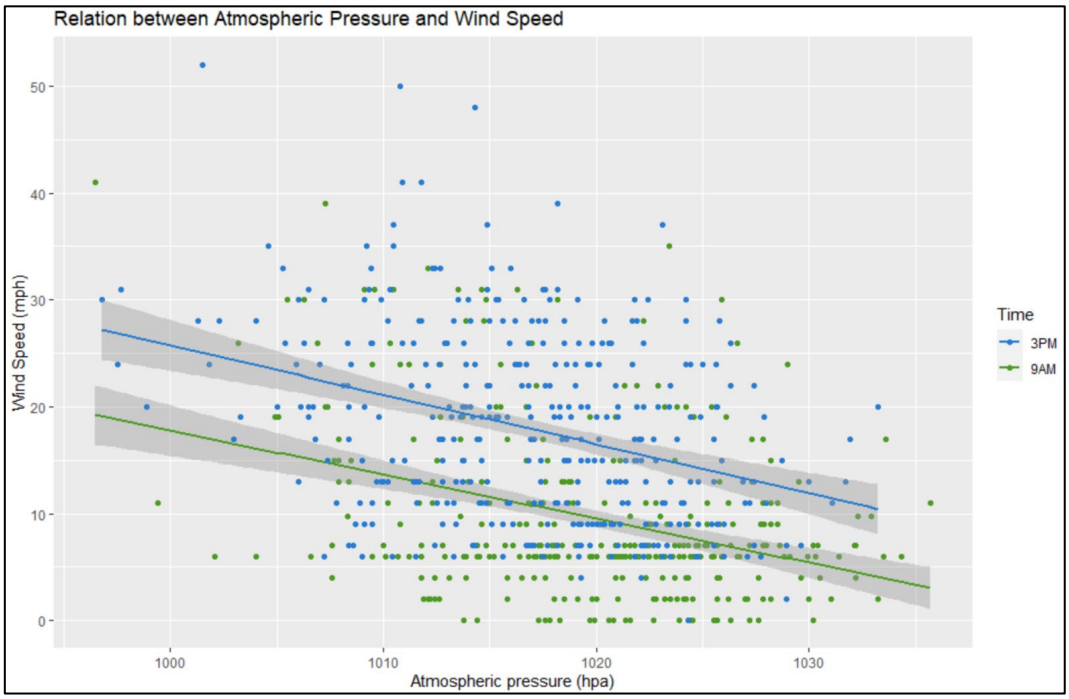

Figure 24.0 Scatter plot of relation between atmospheric pressure and wind speed

Analysis 1.4

It is observed that wind speed is inversely proportional to atmospheric pressure regardless of different times of the day. The steep decline in both colored regression lines indicate that wind speed decreases as atmospheric pressure increases. However, values of wind speed and atmospheric pressure is comparatively lower in the morning (at 9am).

Findings 1.4

The result of this investigation is caused by the difference in atmospheric pressure gradient. The atmospheric pressure gradient exists when hot air rises (which causes high atmospheric pressure) and cold air sinks (which causes low atmospheric pressure). Cool air will move along the atmospheric pressure gradient and rush in to replace the empty space left by the hot air in an attempt to equalize the atmospheric pressure difference. This cycle ultimately leads to the formation of wind (Gellert, 2021). In general, rising air currents in low atmospheric pressure moves faster than sinking air currents in high atmospheric pressure. This is why wind speeds are expected to be higher in lower atmospheric pressure environments. The comparatively lower atmospheric pressure in the morning could be justified that the temperature change in the morning is not as great as in the afternoon. This is because the sun is already out in the sky since 9 am and the sun might start to slowly set from 3 pm, which explains the less significant change in temperature in the morning. As there is smaller change in temperature, there is smaller change in atmospheric pressure which leads to lower wind speeds in general.

Code Snippet 1.5

```
#-------------------------------------------------------------------------------
#ANALYSIS1.3: Air density as determinant of efficiency of wind turbines
#VARIABLES USED: T3, T9, P9, P3
#CONCLUSION: Air density decreases as temperature increases (inversely proportional)
#CONCLUSION: Increase in air density also increases energy received by wind turbines

#calculating average pressure and temperature
Avg_P <- round(((P9 + P3)/2), digits = 1)
AvgP_Pascal <- Avg_P * 100

Avg_T <- round(((T3 + T9)/2), digits = 1)
AvgT_Kelvin <- Avg_T + 273.15

#function to calculate air density
AirDensity <- function (AvgT_Kelvin, AvgP_Pascal)
{
  WP = (28.9647*0.001)*AvgP_Pascal
  RT = 8.314*AvgT_Kelvin
  result = round((WP/RT), digits = 4)
  return(result)
}

AirDen <- AirDensity(AvgT_Kelvin, AvgP_Pascal)

#renaming variables + add new columns to data frame
Air_Density <- AirDen
Average_Temperature <- Avg_T

Q1 = mutate(Q1, Air_Density=Air_Density)
Q1 = mutate(Q1, Average_Temperature=Average_Temperature)

#generate scatterplot of air density against temperature
Q1_scatterplot <- ggplot(Question1, aes(x = Average_Temperature, y = Air_Density)) +
                    geom_point() +
                    geom_smooth(method = "lm") +
                    scale_y_continuous(breaks = seq(1.1, 1.3, by = 0.05)) +
                    scale_x_continuous(breaks = seq(0, 30, by = 5)) +
                    labs(title="Distribution of Average Air Density",
                         x="Average Temperature", y="Average Air Density (in kg/m^3)")
Q1_scatterplot
```

Figure 25.0 Code snippet for analysis 1.5

A new function is created to investigate the relationship between air pressure and temperature. The function named *AirDensity* takes in calculated values of average atmospheric pressure and average temperature (in units of Pascal and Kelvin) as parameters to calculate air density with the following formula:

Since:

$$\mathcal{p} = \frac{mRT}{VM}$$

$$\rho = \frac{m}{V}$$

Hence:

$$\rho = \frac{\mathcal{p}M}{RT}$$

Where:

$\wp$ = Air density

m = Mass

R = Gas constant (8.314 $J\ mol^{-1}K^{-1}$)

T = Temperature

V = Volume

M = Molecular weight of air (28.9647 x $10^{-3}kg/mol$)

ρ = Air density

The deduced values of air density and average temperature are then added into *Q1* data frame in the form of new dataset columns (*Air_Density* and *Average_Temperature*). A scatter plot is plotted with average temperature on the x axis and average air density on the y axis alongside a regression line.

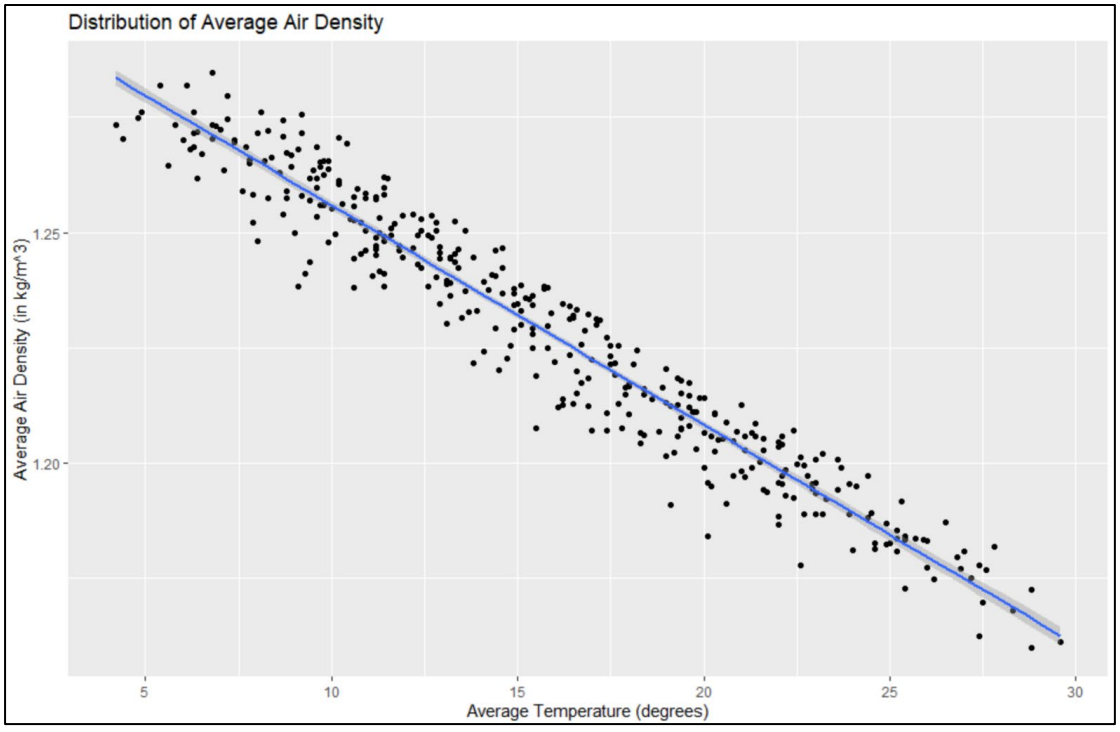

Figure 26.0 Scatter plot of distribution of average air density

Analysis 1.5

As shown in *Figure 26.0*, average air density decreases as average temperature increases. There are no obvious signs of outliers and the plotted points are laying close to the regression line, hence proving there is strong correlation between the two variables; which inherently declares that atmospheric pressure and temperature plays an important role in determining working efficiency of wind turbines.

Findings 1.5

In general, higher altitude results in lower air density when atmospheric pressure decreases. Essentially, kinetic energy is dependent on mass of a moving body; hence power in wind is dependent on mass of air (Danish Wind Industry Association, 2021). Since wind power is directly proportional to air density (wind power increases as air density increases), wind farms should be built at sites with low altitude in order to maximize performance of wind turbines. Aside from atmospheric pressure, temperature affects air density as well. This correlation could be explained by the occurrence of wind. Wind blows when there is uneven heating of air by the Sun, in which hot air rises and cool air rushes in to fill in the empty space left from the rising hot air (BOBBY, 2014). Due to this difference in temperature, air density is increased when air has become “heavier” when it is cooled; which explains the negative regression line (average air density decreases as average temperature increases). This finding supports the reasoning of building wind farms near shores or on shores where altitude is low and winds are frequent.

Question 2: What are the factors that affect location choice for solar panels?

Code snippet 2.1

```
#-----------------------------------------------------------------------------------------------------------------------
#QUESTION2: FACTORS THAT AFFECT LOCATION CHOICE FOR SOLAR PANELS
#-----------------------------------------------------------------------------------------------------------------------
#ANALYSIS2.1: Distribution of Sunshine Intensity
#VARIABLES USED: P9, P3, SS
#CONCLUSION: Lowest pressure + highest sunshine = best location for collecting solar energy
#CONCLUSION: High pressure + highest sunshine = second best location for collecting solar energy
#CONCLUSION: Unable to fulfill first point*
#CONCLUSION: Best: Pressure between 1015 to 1025 under very high sunshine intensity category

#selecting specific columns for Question2
Q2 <- data.frame(subset (weather, select = c("SS", "RiskMM", "P9", "P3", "MinT", "MaxT", "C9", "C3")))
View(Q2)

#calculate average pressure of the day + adding new column into data frame
Avg_P <- round(((P9 + P3)/2), digits = 1)
Q2 = mutate(Q2, Avg_P=Avg_P)

#binning sunlight hours into categories + adding new column into data frame
SS_gap <- c(0, 3, 6, 9, 14)
SS_label <- c("Low", "Medium", "High", "Very High")
SS_group <- cut(Q2$SS, breaks = SS_gap, include.lowest = TRUE, right = TRUE, labels = SS_label)
Q2_SS <- SS_group
Q2 = mutate(Q2, Q2_SS=Q2_SS)

#calculate mean value of each average pressure distribution under each sunshine category
library(plyr)
mu <- ddply(Q2, "Q2_SS", summarise, grp.mean=mean(Avg_P))
mu

#generating histogram for average pressure distribution against sunlight hours
Q2_barchart <- ggplot(Q2, aes(x = Avg_P))+
               geom_histogram(color="#FF5733", fill="#F5B468", binwidth = 1) + facet_wrap("Q2_SS")
legend2 = c("Low Sunshine: 0 - 3", "Medium Sunshine: 3 - 6", "High Sunshine: 6 - 9", "Very High Sunshine: 9 - 14")
Q2_barchart + geom_vline(data = mu, aes(xintercept = grp.mean, color=legend2), linetype="dashed") +
                         labs(title="Distribution of Average Pressure Against Sunshine Intensity",
                         x="Average Atmospheric Pressure", y = "Count of Sunshine Intensity") +
                         guides(col = guide_legend("Mean Atmpsheric Pressure Under\nDifferent Sunshine Intensities"))
```

Figure 27.0 Code snippet

In reference to Question 2, average atmospheric pressure is calculated (from *P9* and *P3* in *Q2* data frame) and the range of sunlight hours is binned into different sunshine intensity categories: "*Low*" (0 – 3 hours), "*Medium*" (3 – 6 hours), "*High*" (6 – 9 hours) and "*Very High*" (9 – 14 hours) with the help of *cut()* function. The package *plyr* is loaded to utilize the *ddply()* function to return *Q2* as a data frame after using *grp.mean()* function in order to calculate mean value of different atmospheric pressures grouped by different categories of sunshine intensities.

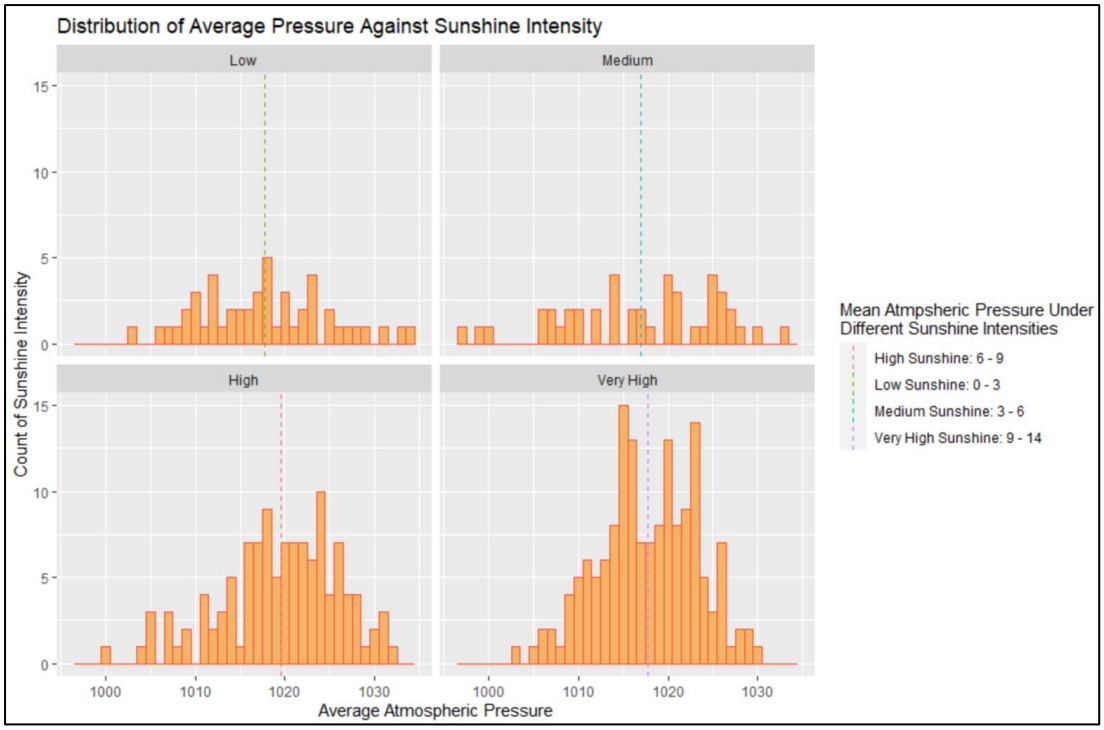

Figure 28.0 Distribution of average pressure against sunshine intensity

Analysis 2.1

Based on the generated four individual histograms above, we can know that the probability of having sunny weather throughout the year is very high. There are higher frequencies of "*High*" and "*Very High*" sunshine intensities comparing to "*Low*" and "*Medium*". Moreover, it is possible to deduce the sunshine intensity group with the highest count of sunlight hours at highest mean average atmospheric pressure given the mean average atmospheric pressures were already indicated by the colorful dotted lines in each respectively grouped histogram (in this case, it would be "*High*" sunshine intensity at mean average atmospheric pressure of approximately 1020 hpa.

Findings 2.1

This finding is supported by the solar information gathered by the National Renewable Energy Laboratory where there is quite a number of states that have an abundance of direct normal solar irradiance on an annual basis. Direct normal solar irradiance (DNI) is defined as the amount of solar radiation a surface is able to receive

per unit area in perpendicular to the direction of solar rays emitted from the sun (Sciencedirect.com., 2021). California, Arizona, New Mexico, Nevada, Utah and Colorado are some of the states that are able to obtain most solar radiance as illustrated in *Figure 29.0* where the mentioned states are colored with maroon or deep shades of red and orange (mentioned states are circled with white circles). Majority of the regions in the United States seem to receive a good proportion of sunlight as well (as indicated by large patches of shades in orange) whilst states around the edges of the country (mostly north) receive lesser or maybe close to none sunlight. As a higher atmospheric pressure is associated with indication of a clear weather due to higher pressure system (Society, N., 2021) , the generated histogram is able to tell us United States can expect steady and reliable sunshine intensity of about 6 to 9 hours on a clear day. On the other hand, the Figure suggest the best locations (with highest direct normal solar irradiance) to build solar panels on.

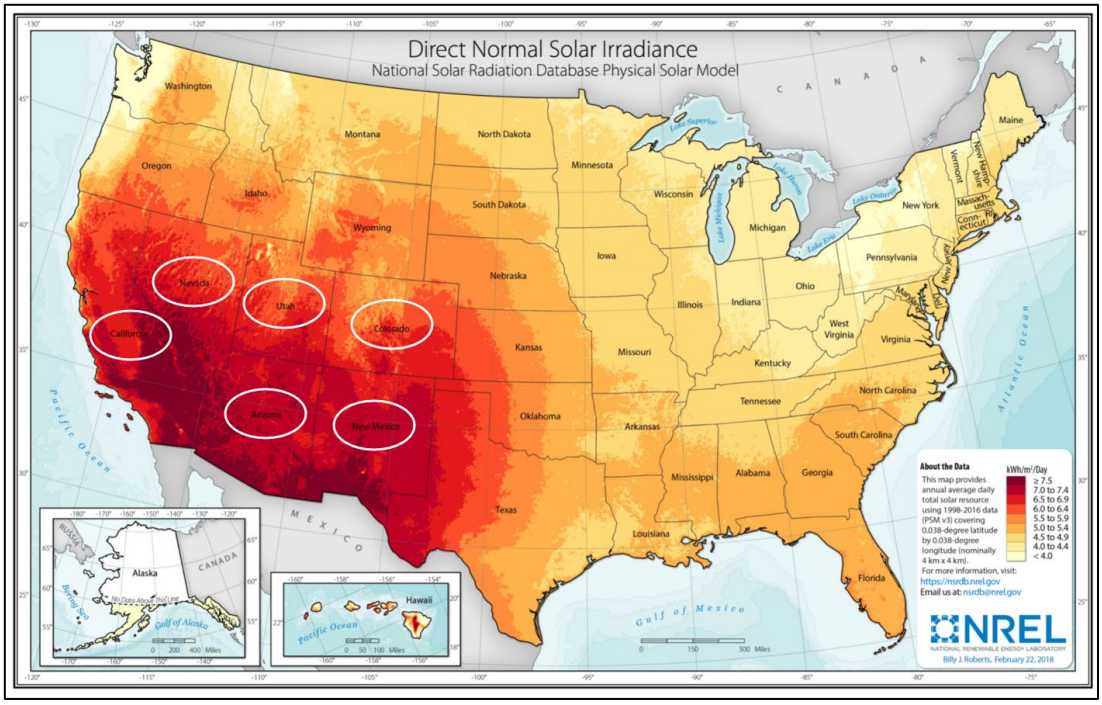

Figure 29.0 Direct Normal Solar Irradiance (Nrel.gov., 2018)

Code snippet 2.2

```
#-------------------------------------------------------------------------------------------
#ANALYSIS2.2: Temperature will affect efficiency of solar panels
#VARIABLES USED: MaxT, MinT, SS
#CONCLUSION: Solar panels produce less power on hot temps. (lower temps. are favorable)

#generating boxplot (MaxT & SS)
Q2_boxplot <- ggplot(Q2, aes(x = Q2_SS, y = MaxT)) +
                    geom_boxplot(aes(color = factor(Q2_SS))) +
                      stat_summary(funn.y = mean, geom="point", shape=23, size=6)
Q2_boxplot + labs( x = "Sunshine Intensity", y = "Maximum Temperature",
                   title = "Relation between Sunshine Intensity and Maximum Temperature") +
                   scale_y_continuous(breaks = seq(0, 36, by = 5)) +
                   guides(col = guide_legend("Sunshine Intensities"))

#generating boxplot (MinT & SS)
Q2_boxplot2 <- ggplot(Q2, aes(x = Q2_SS, y = MinT)) +
                    geom_boxplot(aes(color = factor(Q2_SS))) +
                      stat_summary(fun.y=mean, geom="point", shape=23, size=6)
Q2_boxplot2 + labs(x = "Sunshine Intensity", y = "Minimum Temperature",
                   title = "Relation between Sunshine Intensity and Minimum Temperature") +
                   scale_y_continuous(breaks = seq(-6, 21, by = 5)) +
                   guides(col = guide_legend("Sunshine Intensities"))
```

Figure 30.0 Code snippet

Eight boxplots are generated with maximum temperature and minimum temperature as variables on the y axis (as indicated by *MaxT* and *MinT* from *Q2* data frame) and sunshine intensity (binned and deduced from *Figure 30.0*) as variable on the x axis. The *stat_summary()* function is also used to display the mean values of the variables on the y axis.

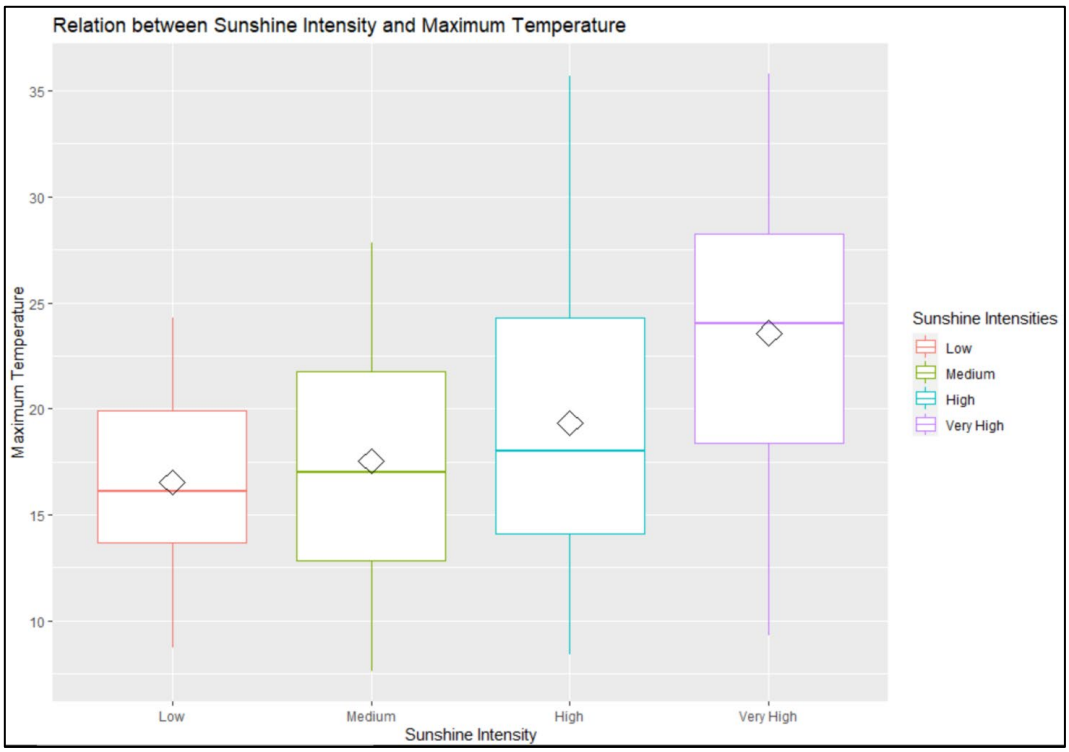

Figure 31.0 Box plots (max temperature)

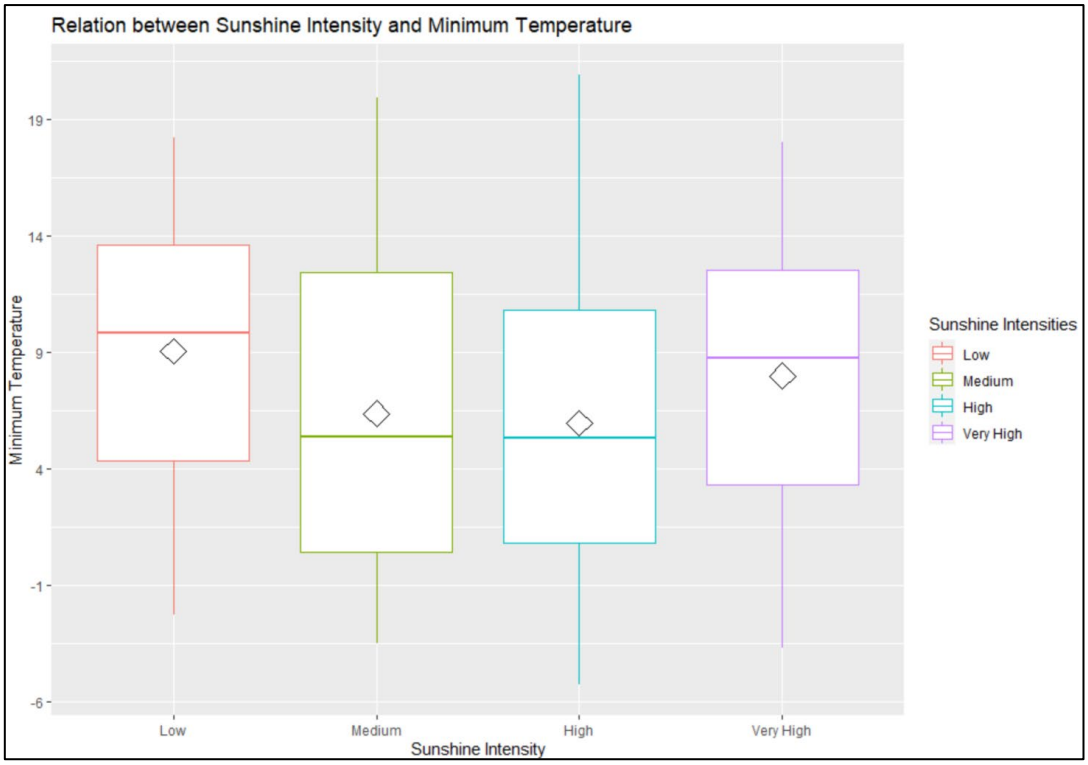

Figure 32.0 Box plots (min temperature)

Analysis 2.2

As shown from both figures above, no means of maximum nor minimum temperature is aligned with their respective median values in each boxplot (diamond-shaped box is not aligned with median line in box plot). However, there seems to be an increase in interquartile range in boxplots across different sunshine intensity categories when temperature is at maximum. In addition to that, difference in range of maximum and minimum temperature of each box plot is bigger when temperature is at maximum (refer to *Figure 31.0*).

Findings 2.2

It might sound counter-intuitive, yet studies have shown that increase in temperature will reduce efficiency of photovoltaic cells (commonly known as solar cells) which solar panels are assembled from. According to research, output efficiency of photovoltaic cells is reduced by 10 to 25% when they are tested at temperature of 25 degrees under factory standard test conditions (CED Greentech, 2021). This is because supplied heat does not affect the amount of solar energy received by a solar panel, yet

it does affect the amount of energy it could produce as output. The efficiency of solar panels could be calculated as:

$$Effiency = \frac{Max\ current\ x\ Max\ voltage}{input\ from\ the\ sun}$$

In the case where temperature is increased, voltage decreases while current increases. However, rate at which voltage decreases is faster than rate at which current increases; which leads to lowered efficiency in solar panels. Needless to say, energy output from solar panels decreases as its efficiency suffers (Renvu.com., 2021). With the plotted boxplots, we are able to make decisions on where to place the solar panels on locations where there is abundance of sunlight and temperature does not exceed 25 degrees.

Code snippet 2.3

```
#-------------------------------------------------------------------------------------------
#ANALYSIS2.3: Cloud Coverage will affect efficiency of solar panels
#VARIABLES USED: C9, C3
#CONCLUSION: higher cloud coverage = higher solar shading = lower solar panel efficiency

#calculating average cloud coverage
Avg_C <- round(((C9 + C3)/2), digits = 1)
Q2 = mutate(Q2, Avg_C=Avg_C)
View(Q2)

Q2_hist2 <- ggplot(data=Q2, aes(Avg_C)) +
  geom_histogram(col="white", fill="#92D8E7", binwidth = 1) +
  scale_x_continuous(breaks = seq(0, 8, by = 1)) +
  scale_y_continuous(breaks = seq(0, 100, by = 10))+
  labs(title="Distribution of Average Cloud Coverage") +
  labs(x="Cloud Coverage (in oktas)", y="Count")
Q2_hist2
```

Figure 33.0 Code snippet

Average cloud coverage is calculated from cloud coverage values at 9am and 3pm in oktas (deduced from *C9* and *C3* in *Q2* data frame) and then added as a new dataset column into *Q2* dataset. A histogram is plotted to display distribution of average cloud coverage throughout the whole year in the United States.

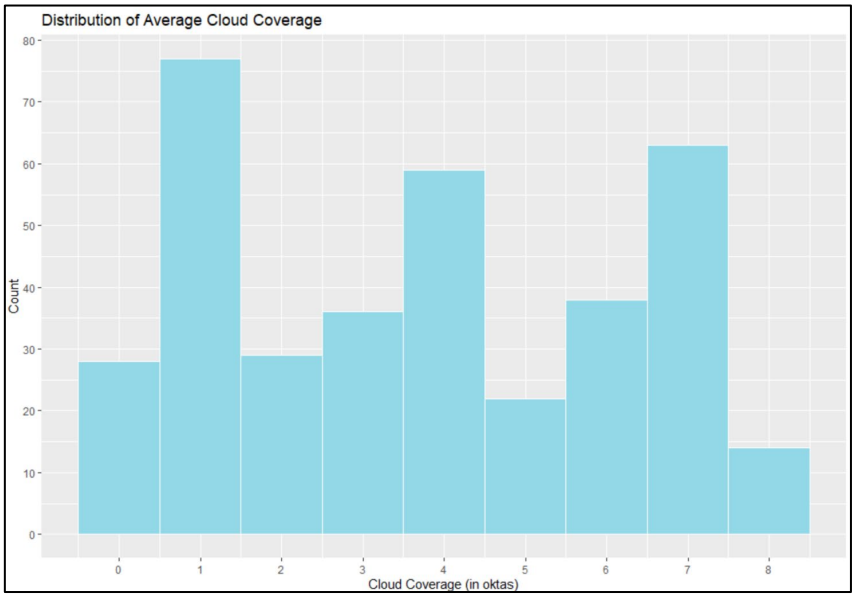

Figure 34.0 Distribution of average cloud coverage

Analysis 2.3

The above histogram shows a visibly uneven distribution of average cloud coverage with obvious peaks when average cloud coverage is at values of 1, 4 and 7 oktas. The highest frequency of average cloud coverage is 1 okta while lowest frequency of average cloud coverage is 8 oktas.

Findings 2.3

Amount of cloud is measured in oktas, a unit representative of how many eights of the sky is covered in clouds ranging from 0 okta (clear sky with complete absence of cloud) to 8 oktas (overcast where the sky is completely covered with clouds) (Met Office, 2021). The accuracy of the resulting histogram from *Figure 34.0* could be supported by the percentage of sky cover in the United States' weather forecast diagram below. The greyer the area it is, the higher the sky cover percentage is forecasted to be. On the contrary, the bluer the area it is, the lower the sky cover percentage is forecasted to be. We could see that there are seemingly lesser grey areas appearing on the map of United States comparing to the blue areas, which reflects the results produced by the histogram in *Figure 34.0*. This finding also informs us that solar shading is a factor we have to take into account of as solar shading occurs when solar panels are unable to receive solar rays when they are blocked off by environmental obstructions such as clouds. It is important to take note that solar shading on one panel will reduce the power output of solar panels to zero; as the panels are wired and assembled in such a way that the output will be reduced to a level of current that runs through the weakest panel (RV Solar Solution, 2021). Hence it is important to position the solar panels in a way that no there is the least environmental obstruction throughout the year that will get in the way of these working solar panels before finalizing the photovoltaic system.

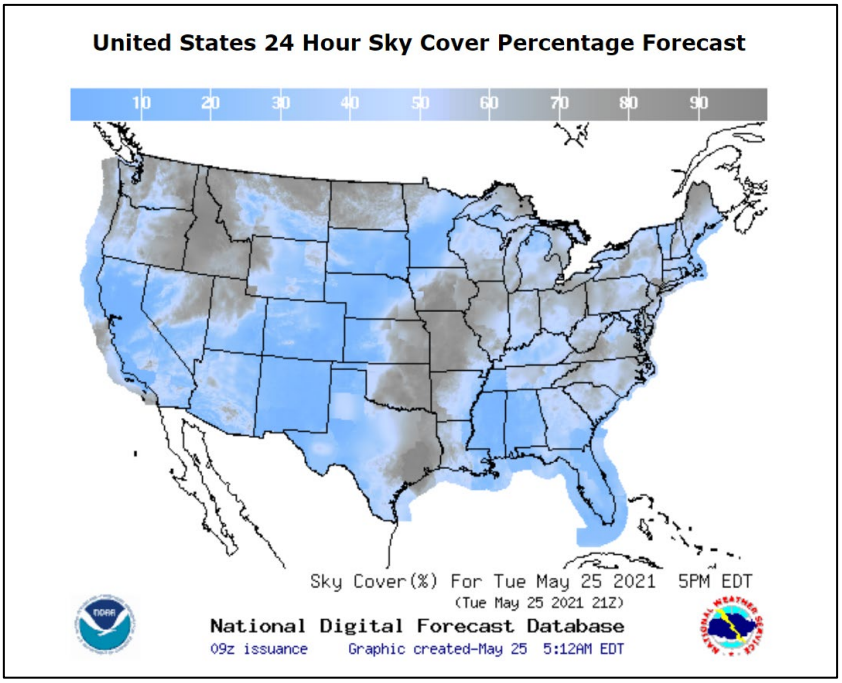

Figure 35.0 Sky cover percentage in U.S. (Eldoradoweather.com., 2021)

Question 3: What are the factors that might cause natural disasters in the United States?

Code snippet 3.1

```r
#-----------------------------------------------------------------------------------------------------------------
#QUESTION3: FACTORS THAT MIGHT CAUSE NATURAL DISASTERS
#-----------------------------------------------------------------------------------------------------------------
#ANALYSIS3.1: Correlation between temp diff and evaporation = (increased wind speed at sea)
#VARIABLES USED: MinT, MaxT, E
#CONCLUSION: High temp diff will cause increase in evaporation (as indicated by linear regression)

#selecting specific columns for Question3
Q3 <- data.frame(subset (weather, select = c("RF", "RiskMM", "WGD", "WGS", "P9", "P3", "MaxT", "MinT", "E")))
View(Q3)

#calculating difference in temp + adding new column
TempDiff <- round((MaxT - (MinT)), digits = 1)
Q3 = mutate(Q3, TempDiff = TempDiff)

#binning temperature difference
TD_gap <- c(0, 7, 14, 23)
TD_label <- c("Low", "Medium", "High")
TD_group <- cut(Q3$TempDiff, breaks = TD_gap, include.lowest = TRUE, right = TRUE, labels = TD_label)
Q3_TD <- TD_group
Q3 = mutate(Q3, Q3_TD=Q3_TD)

#generating scatter plot
Q3_scatterplot <- ggplot(Q3, aes(x = TempDiff, y = E, color = Q3_TD)) +
                    geom_point() +
                      geom_smooth(method = lm, se = FALSE, fullrange = TRUE) +
                        guides(col = guide_legend("Difference in Temperature")) +
                          labs(x = "Difference in Temperature (degrees)", y = "Evaporation",
                               title = "Relation between Evaporation and Temperature Difference") +
                            scale_y_continuous(breaks = seq(0, 14, by = 3)) +
                            scale_x_continuous(breaks = seq(0, 25, by = 5))
Q3_scatterplot
```

Figure 36.0 Code snippet

The variable *TempDiff* is derived from calculating difference in temperature from *MaxT* and *MinT* columns in *Q3* data frame and rounding the derived values up to one decimal place (this variable is later added as a new column into *Q3* data frame). The derived TempDiff values are then binned into different categories of temperature difference in degrees: "*Low*" (0 -7), "*Medium*" (7 – 14) and "*High*" (14 – 23) with the help of *cut()* function. A scatterplot is then plotted with variable *TempDiff* on the y axis and categories of temperature difference as the variable on the x axis.

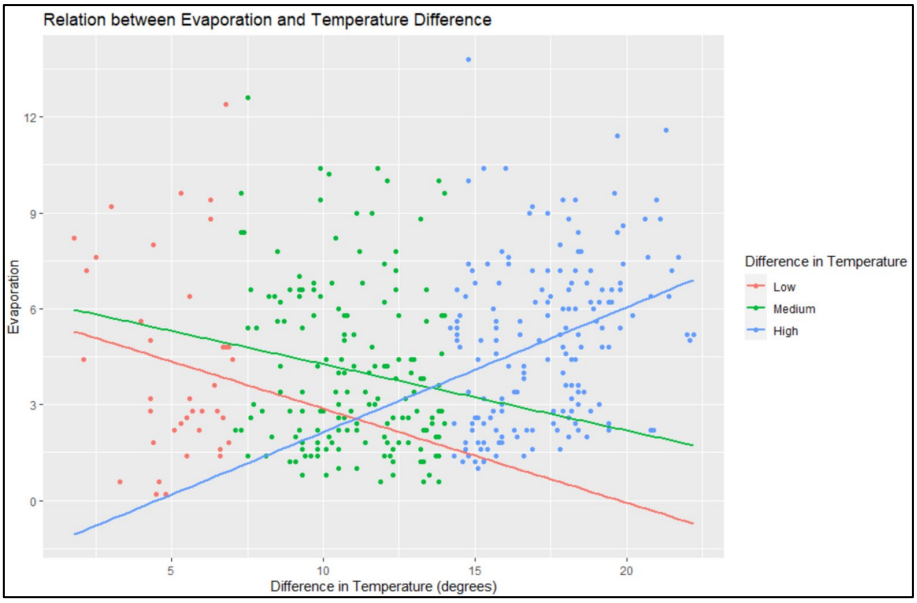

Figure 37.0 Scatter plot of relation between evaporation and temperature difference

Analysis 3.1

As shown in the scatterplot above, there is strong positive linear regression in the case where difference in temperature is high and evaporation rate increases. However, linear regression lines are negative in the case where difference in temperature is classified as low or medium. We would be taking the strong positive linear regression into account for our analysis as the slope of positive linear regression line is steeper (blue line) than the other two lines, hinting there is a stronger relation between high temperature difference and evaporation rate.

Findings 3.1

Research has shown that increase in temperature does increase rate of evaporation. This is due to the increase in temperature supplies heat to water around us. The water will absorb heat and this is where evaporation takes place to transform water to water vapour. After that, water vapour will rise to the air and condense to become precipitation, rain or snow before repeating the nature cycle all over again. Given that the ocean consists of an estimate of 96% of free water in addition to the fact that pace of global warming is increasing at an alarming rate; it is undeniable that the greater the

heat is supplied to this large body of water, the greater the evaporation rate which will inevitably lead to more extreme weathers (Lipsett, 2012).

Code snippet 3.2

```
453 #-------------------------------------------------------------------------
454 #ANALYSIS3.2: Distribution of wind gust direction
455 #VARIABLES USED: WGD
456 #CONCLUSION: Most wind gusts come from North West (21%)
457
458 #generating wind gust direction pie chart
459 library(plotrix)
460 WindNameT <- c("N","S","E","W","SE", "SW","NE", "NW",
461                "ENE", "ESE", "NNW", "SSE",
462                "SSW", "WNW", "WSW")
463
464 WindNameT <- paste(WindNameT, Q3pie)
465 WindNameT <- paste(WindNameT,"%",sep = "")
466
467 pie3D(WG_D, labels = WindNameT, main="Pie chart of Wind Gust Direction",
468       radius = 1, border = "white", col = rainbow(length(WindNameT)))
```

Figure 38.0 Code snippet

Names of all wind directions are abbreviated (as indicated by *"N"*, *"S"*, *"E"*, *"W"*, *"SE"*, *"SW"*, *"NE"*, *"NW"*, *"ENE"*, *"ESE"*, *"NNE"*, *"NNW"*, *"SSE"*, *"SSW"*, *"WN"*, *"WSW"*) and stored as a vector in variable *WindNameT*. The package *plotrix* is loaded in order to use *pie3D()* function to plot a 3D pie chart.

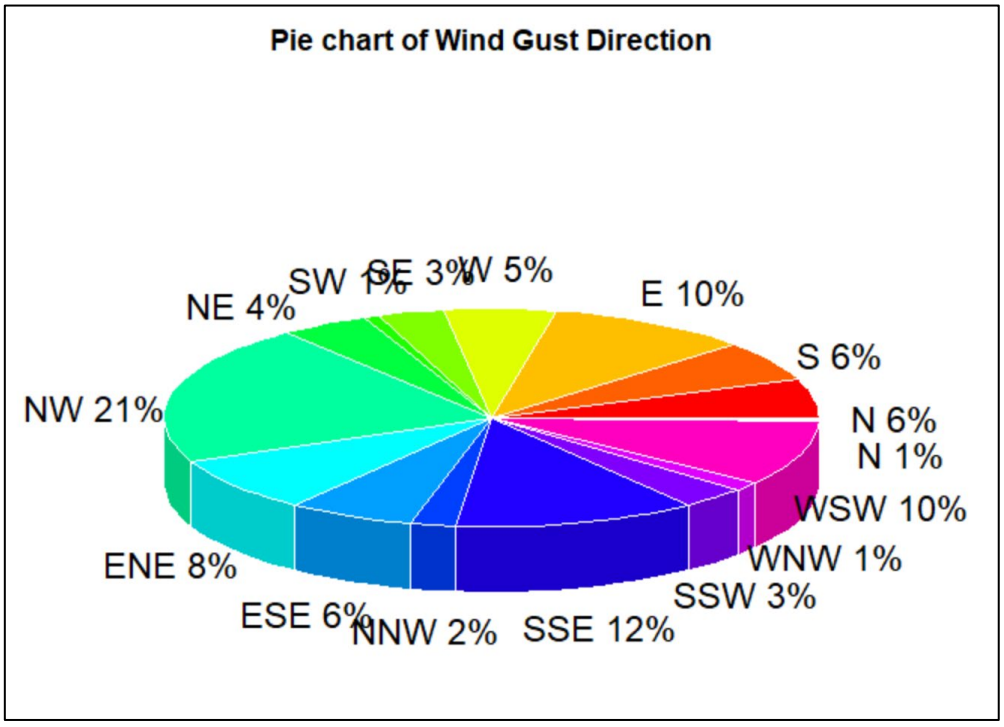

Figure 39.0 Pie chart of Wind Gust Direction

Analysis 3.2

It is observed that wind gusts from North West have the highest frequency among all directions as it occupies a percentage of 21%. The runner-ups would be South South East, West South West and East directions with percentages of 12%, 10% and 10%. The rest of the directions have not much wind gusts as their percentages do not exceed 10%.

Findings 3.2

Prevailing winds are winds that generally blows in a single direction over a specific area of the Earth. More often than not, prevailing winds blows towards east-west direction due to Earth's rotation which causes the Coriolis effect (when wind system twists rightward in the Northern hemisphere and twist leftward in the Southern hemisphere) (Society, 2021). United States being in the Northern and Western hemispheres have prevailing winds that blows from west to east (Trees-energy-conservation.extension.org., 2021), which justified why most wind gusts blows from North West direction. This is helpful in predicting the probability of cyclone formation (commonly known as "hurricane") because cyclone forms when air from surrounding areas with higher air pressure rush into a low-pressure area and creates a rapid inward circulation of air, fueled by heat and evaporation into the air (Gpm.nasa.gov., 2021). Studies have shown that these weather events rotate counter-clockwise in the Northern hemisphere and rotate clockwise in the Southern hemisphere with potential occurrence number of 10 to 15 annually (Marchigiani et al, 2013).

Code snippet 3.3

```
#-------------------------------------------------------------------------------------------------------
#ANALYSIS3.3: Distribution of wind gust speed in North West
#VARIABLES USED: WGS
#CONCLUSION:Mean wind speed is around 45, highest wind speed can reach close to 100 (dangerous)

#selecting wind gust speeds with North West direction
filter_NW <- filter(Q3, WGD %in% c("NW"))
filter_NW <- filter_NW[, 4]
filter_NW

#put selected values into new data frame
SubQ3 <- rbind(data.frame(WindDirection = "NW", WindSpeed = filter_NW))
View(SubQ3)

#generating histogram for wind speed in north west
Q3_hist <- ggplot(data=SubQ3, aes(WindSpeed)) +
            geom_histogram(col="white", fill="steelblue", binwidth = 3) +
            geom_vline(aes(xintercept=mean(WindSpeed)),
            color="black", linetype="dashed", size=.5) +
            scale_x_continuous(breaks = seq(0, 100, by = 5)) +
            scale_y_continuous(breaks = seq(0, 20, by = 2))+
            labs(title="Histogram for Wind Gust Speed in North West Direction") +
            labs(x="Wind Gust Speed (mph)", y="Count")
Q3_hist
```

Figure 40.0 Code snippet

The *filter()* function is used to find rows in column *WGD* (representative of wind gust direction) where wind gust direction is of "*NW*". Once found, the selected values (values of wind gust direction and wind speed) are used to create a new data frame named as *SubQ3*. A histogram is then plotted to find out distribution of wind gust speed in the direction of North West along with its mean of wind gust speed.

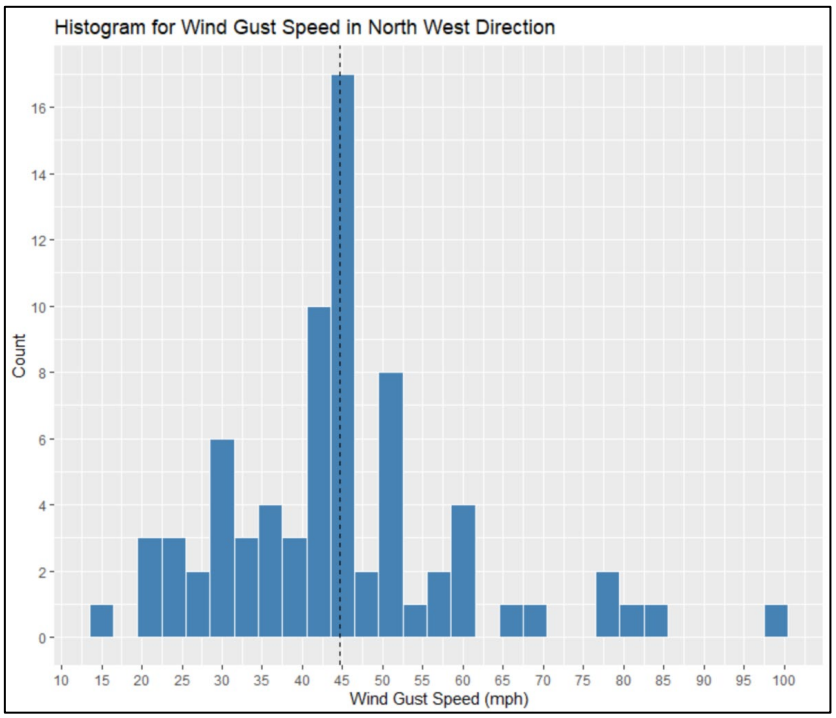

Figure 41.0 Histogram of wind gust speed in North West

Analysis 3.3

Frequency of wind gust speed centering around 45 mph is the highest among all wind gust speeds with a count of 16.5. The histogram is right-skewed, with more frequencies of wind gust speeds at the lower end of the range. There are a few outliers which are wind gust speeds in the range of approximately 75 mph to 100 mph when the mean of wind gust speed is close to 45 mph).

Findings 3.3

The High Wind Hazard Map provides a table of high wind threat levels in accordance to their respective wind gust speeds (Weather.gov., 2021).

High Wind Threat Level	Wind Gust Speed	Descriptions
Extreme	> 58 mph	Damaging high wind - Extreme threat to life and property from high wind
High	40 mph – 57 mph	High wind - High threat to life and property from high wind
Moderate	26 mph – 39 mph (sustained speeds) 35 mph – 57 mph (frequent wind gusts)	Very windy - Moderate threat to life and property from high wind
Low	21 mph – 25 mph (sustained speeds) 30 mph – 35 mph (frequent wind gusts)	Windy - Low threat to life and property from high wind
Very Low	20 mph (sustained speeds) 25 mph – 30 mph (frequent wind gusts)	Breezy to Windy

		- Very low threat to life and property from high wind
Non – Threatening	-	Breezy - No discernable threat to life and property from high wind

According to the table above, it is considered as dangerous when wind gust speed surpassed 40 mph as the associated risks and dangers are considered as high and extreme threat levels. That being said, United States are exposed to more turbulence and wind gust induced natural disasters. Natural disaster mitigation strategies should be devised in advance to prepare for whatever havoc the storms bring in order to not only save monetary property as well as human lives too.

Q4 What are the factors that determine agriculture sites?

Code Snippet 4.1

```
#-------------------------------------------------------------------------------------------------
#QUESTION4: FACTORS THAT DETERMINE AGRICULTURE SITES
#-------------------------------------------------------------------------------------------------
#ANALYSIS4.1: Pattern of humidity throughout the year
#VARIABLES USED: H9, H3
#CONCLUSION: Humidity is lower at 3pm comparing to 9am
#CONCLUSION: Crops should avoid areas where humidity is high

#selecting specific columns for Question4
Q4 <- data.frame(subset (weather, select = c("H3", "H9", "T3", "T9", "RF")))
View(Q4)

#generating smooth graph
day = 1:366
YearH9 = cbind(H9,day,"9am")
YearH3 = cbind(H3,day,"3pm")
YearH = as.data.frame(rbind(YearH9,YearH3))
colnames(YearH) = c("Humidity","Day","Time")
YearH

Q4_smooth = ggplot(data=YearH, aes(x=as.numeric(Day), y=as.numeric(Humidity), group=Time)) +
              geom_smooth(aes(color=Time))+
              scale_x_continuous("Number of Days",breaks = seq(0,366,by = 100)) +
              scale_y_continuous("Humidity",breaks = seq(0,100,by = 10)) +
              labs(title="Humidity Pattern Throughtout One Year")
Q4_smooth
```

Figure 42.0 Code snippet

A smooth graph is plotted to observe the pattern of humidity at different times (9am and 3pm) throughout a year with the number of days as variable on the x axis and humidity on the y axis.

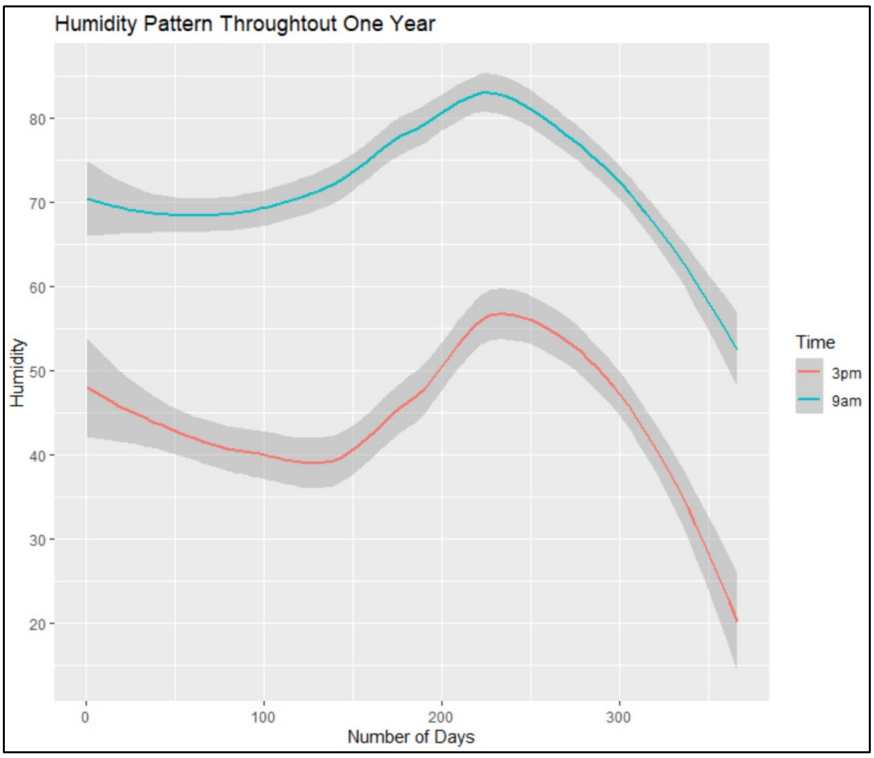

Figure 42.0 Humidity pattern graph

Analysis 4.1

As observed from *Figure 42.0*, humidity at 3pm is lower comparing to humidity at 9am in general. Both lines achieved different values of minimum and maximum humidity: lowest humidity at 9am is between the range of 50 to 55 while highest humidity at 9am is between the range of 80 to 85. Meanwhile, lowest humidity at 3pm is 20 and highest humidity at 3pm is between the range of 55 to 60. The only similarity both lines share is that their humidity decreases slightly at first and increases to their maximum point before decreasing exponentially at a much steeper slope than their initial decrease.

Findings 4.1

It is advised to not plant crops in areas where humidity is high as this environment factor has a close-knit relationship with pest and disease management. This is because humidity is an essential requirement needed for spore germination, multiplication and penetration of bacteria (Encyclopedia Britannica, 2021). Plant crops might fail and die due to promotion of bacteria and fungus growth; or you might find yourself suffering from harvesting crops of good quality when introduction of pests due to high humidity has spoiled your harvest (Polygongroup.com., 2021). However, it is also important to note that even though high humidity brings bad news to crop production, low humidity have its fair share of adverse effects as well. Low humidity leads to decrease in rate of photosynthesis and transpiration in plants, which inherently cause them to compromise growth in order to converse water and not die of wilting. Moreover, the harvested plants might not be of good quality as well even when the duration taken for the crops to grow to its desirable size is longer than usual (Pthorticulture.com., 2021). The graph in *figure 42.0* shows that it is best if farmers avoid planting and crops after August as there will be a sharp decline in humidity which might lead to production of poor-quality crops with the burden of reduced profits and increased production costs.

Code snippet 4.2

```
#-------------------------------------------------------------------------------------------------
#ANALYSIS4.2: Correlation between average temperature and humidity
#VARIABLES USED: H9, H3, T3, T9
#CONCLUSION: There is correlation between temperature and humidity (higher temp = lower humidity)
#CONCLUSION: Crops should avoid areas where humidity is high

#calculating average humidity & temperature + add to Question4 data frame
Avg_H <- ((H9 + H3)/2)
View(Avg_H)

Avg_T <- ((T9 + T3)/2)
View(Avg_T)

Q4 = mutate(Q4, Avg_H=Avg_H)
Q4 = mutate(Q4, Avg_T=Avg_T)

#generating scatterplot for avg humidity and temp.
Q4_scatterplot <- ggplot(Q4, aes(x = Avg_T, y = Avg_H)) +
                  geom_point(color="cornflowerblue",
                             size = 2,
                             alpha=.5) +
                  geom_smooth(method = "lm") +
                  scale_y_continuous(breaks = seq(0, 100, by = 10)) +
                  scale_x_continuous(breaks = seq(0, 30, by = 5)) +
                  labs(x = "Average Temperature", y = "Average Humidity",
                      title = "Relation Between Average Temperature and Average Humidity")
Q4_scatterplot
```

Figure 43.0 Code snippet

Values of average humidity and average temperature are derived and calculated from dataset columns *H9*, *H3*, *T9* and *T3* before they are assigned to variables *Avg_H* and *Avg_T*. These variables are later added as new columns of values into *Q4* dataset in order to plot a scatter plot where correlation between average temperature and average humidity is explored and investigated.

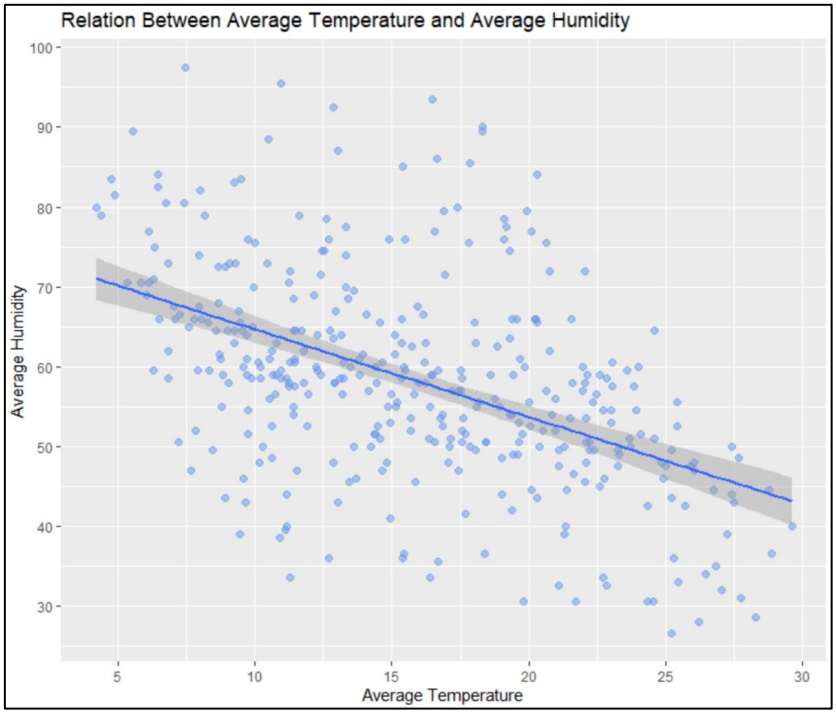

Figure 44.0 Scatter plot of relation between average temperature and average humidity

Analysis 4.2

We could see that there is strong correlation between the two variables: average humidity and average temperature (as indicated by the negative slope of linear regression line). The scatter plot also demonstrates that average humidity is inversely proportional to average temperature, as average humidity decreases as average temperature increases.

Findings 4.2

In reference to *figure 42.0*, we could link both graphs together in the sense that the temperature is suggesting changes in season throughout a year in the United States. As temperature in air increases, warmer air can hold more water molecules which increases relative humidity (humidity is defined as the amount of water vapor in air) (Society, 2021). In other words, since humidity is affected by changes in temperature; changes in temperature act as indicator of changes in seasons as well. Hence, we could say that different seasons have different relative humidity values, which determines timing of certain production of different crops. This finding could facilitate farmers in better decision-making in terms of when should they get prepped to plant crops, when to not start planting crops, what crops should be planted in specific seasons and etc. For instance, on periods when average temperatures are prone to be unexpectedly high; farmers should put a pause on crop production as the change in climate implies the arrival of summer season followed by possible drought. Assuming farmers did not pay attention to changes in temperature or humidity, the planted crops are most likely to fail as moisture level in soil decreases and became drier than usual (United States Environmental Protection Agency, 2021).

Code snippet 4.3

```
#-------------------------------------------------------------------------------
#ANALYSIS4.3: Rainfall in a year (Does rainfall affect crops production?)
#VARIABLES USED: RF
#CONCLUSION: Not much rainfall throughout the year (only 2 months worth of rainfall)
#CONCLUSION: Droughts(?) Possible famine/ poor crop production

#generating histogram for rainfall distribution
Q4_hist <- ggplot(data=Q4, aes(RF)) +
  geom_histogram(col="white", fill="steelblue", binwidth = 2) +
  scale_x_continuous(breaks = seq(0, 40, by = 5)) +
  labs(title="Distribution of Rainfall",
    subtitle="on an annual basis") +
  labs(x="Rainfall (in mm)", y="Count")
Q4_hist
```

Figure 45.0 Code snippet

```
#-------------------------------------------------------------------------------
#ANALYSIS4.3: Distribution of rainfall on rainy days
#VARIABLES USED: RF
#CONCLUSION: Dry weather, not much rain even on rainy days

#selecting rainfall with more than 1 mm
filter_RF <- filter(Q3, RF > 1)
filter_RF <- filter_RF[, 1]
filter_RF

#put selected values into new data frame
SubQ4 <- rbind(data.frame(Rain = "TRUE", RainFall = filter_RF))
View(SubQ4)

#generating histogram for rainfall distribution (that is not 0 and more than 1)
Q4_hist2 <- ggplot(data=SubQ4, aes(filter_RF)) +
  geom_histogram(col="white", fill="steelblue", binwidth = 2) +
  scale_x_continuous(breaks = seq(0, 50, by = 5)) +
  scale_y_continuous(breaks = seq(0, 20, by = 5)) +
  labs(title="Distribution of Rainfall",
       subtitle="On rainy days") +
  labs(x="Rainfall (in mm)", y="Count")
Q4_hist2
```

Figure 46.0 Code snippet

Initially, only one histogram is plotted to observe annual distribution of rainfall in United States (*figure 47.0*). However, upon observing the extremely uneven distribution in *figure 47.0*, another histogram is plotted to delve deeper into the area where distribution of rainfall in United States is heavier and more concentrated by using the *filter()* function to subset *Q3* data frame where values stored in *RF* dataset column is more than 1. The retained rows of data are then placed into a new data frame named as *SubQ4* (used for plotting of the second histogram).

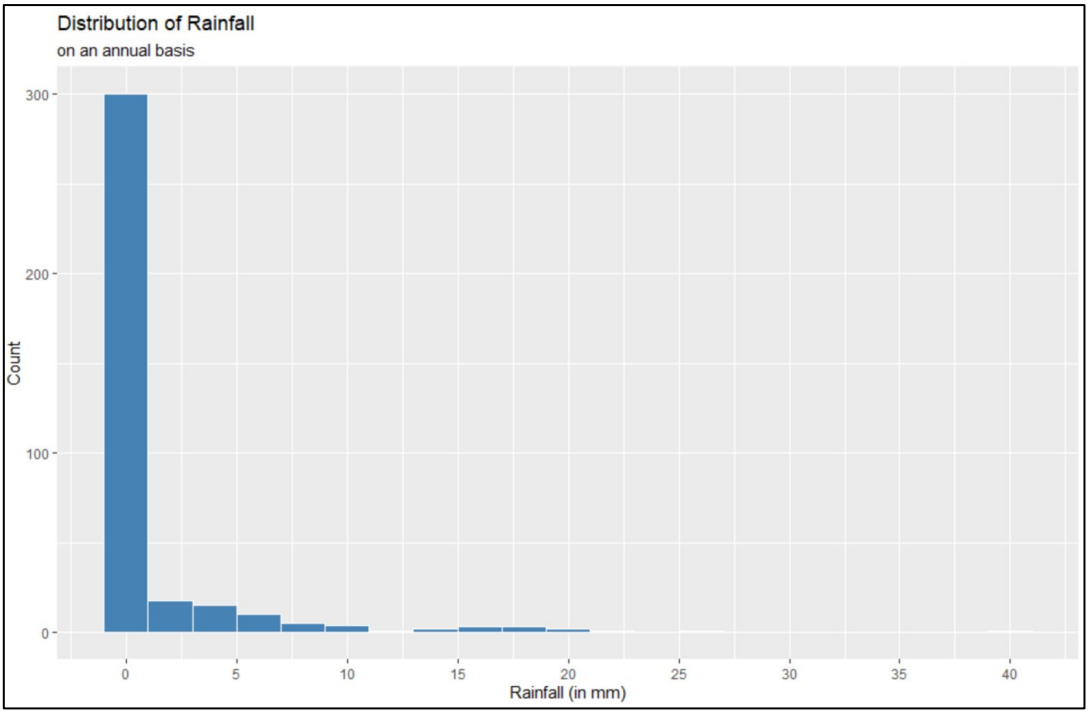

Figure 47.0 Distribution of rainfall

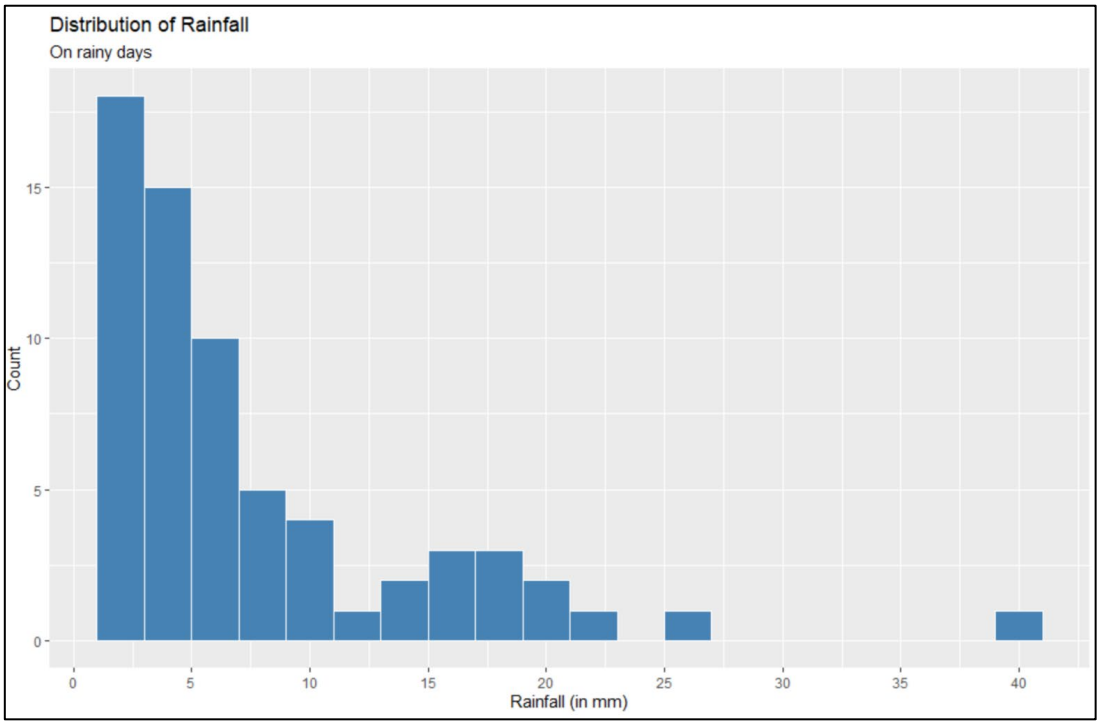

Figure 48.0 Distribution of rainfall on rainy days

Analysis 4.3

The first histogram in figure shows annual distribution of rainfall in United States. There is unequal distribution as highest count of rainfall is 300 (where rainfall is at 0) while lowest count of rainfall is approximately or close to 0. We might not be able to make accurate deductions of the data we received from this histogram given that data is so unnormalized. Hence, we narrow down the area of data analysis where we only focus on investigating data where rainfall is not equal to 0 and more than 1 in the second histogram as shown in *figure 48.0*. From the right skewed histogram in *figure 48.0*, we could see that most of the amount of rainfall in United States are clustered around the lower ends of the rainfall range (between 0 to 20) even though there are presence of outliers as well (between rainfall range of 25 to 40 with an estimate count of 2).

Findings 4.3

From the analysis above, we can conclude that dry weather (or dry seasons) is prevalent in the United States as justified by the drastic comparison of frequency in rainfall amount. Even if there are instances of rainy days, amount of rainfall is still considerably low on an annual basis. This information could be an important cue for farmers to start planning on what crops to be planted or start prepping planting conditions to fulfil requirements of specific crops. When there is insufficient rainfall or precipitation, farmers might have to consider other methods of procuring water source to sustain their crops, such as irrigation. Irrigation is a man-made, artificial process where water is supplied to plants alike in controlled amounts. Research has shown that about 80% to 90% of United States utilize the technology of irrigation to cope with precipitation shortage. The decision to utilize and implement irrigation is crucial not only in terms of food production issues, but on an economic scale as well; for agriculture accounts for 55% of the total value of crop sales in United States back in 2007 (Student Materials, 2021). The pie chart below shows how reliant and dependent United States is on the technology of irrigation:

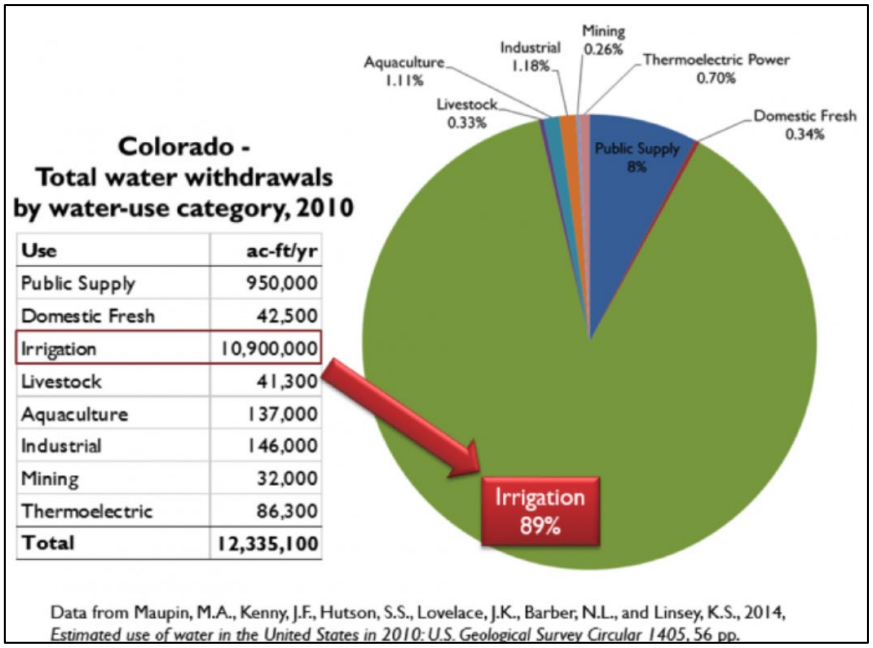

Figure 49.0 Total water withdrawals (Student Materials, 2021)

Code snippet 4.4

```
#-----------------------------------------------------------------------------------------------------------------
#ANALYSIS4.4: Distribution of Evaporation
#VARIABLES USED: E
#CONCLUSION: High evaporation rate since day 1, decreases around day 150 and increase again around day 250
#CONCLUSION: Average daily evaporation rate is 4.521858mm (high)
#CONCLUSION: Lost of soil water and will affect plant grows

#generating line graph
day = 1:366
YearEvap = as.data.frame(cbind(E,day))
YearEvap

Q4_line = ggplot(YearEvap,aes(x=day,y=E))+
            geom_line()+
            geom_hline(aes(yintercept=mean(E)), color="green", linetype="dashed", size=1) +
            labs(title="Distribution of Evaporation") +
            labs(x="Day", y="Evaporation (mm)")
Q4_line
mean(E)
```

Figure 50.0 Code snippet

A line chart is plotted to show distribution of evaporation throughout a year with evaporation mean as indicated in a green dashed line in *figure 51.0*.

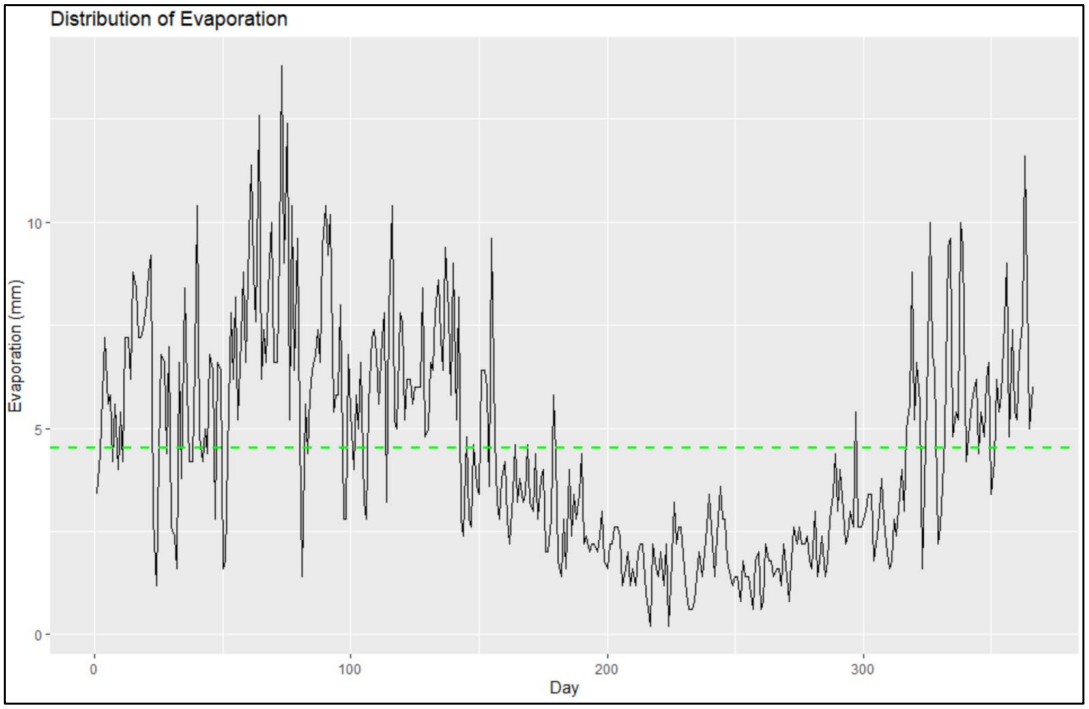

Figure 51.0 Line chart of distribution of evaporation

Analysis 4.4

As observed from *figure 51.0*, trend of evaporation in United States increases in the beginning of the year before it dips to the lowest point of almost 0 mm at about half a year in before it picks up and eventually increases towards the end of the year. The highest rate of evaporation as observed is approximately at 14 mm.

Findings 4.4

Evaporation is closely related to transpiration process in plants. Transpiration describes the water movement in plants in which water is lost in the form of water vapor. The water absorbed by plants usually contributes to the transpiration process in plants, hence the term "evapotranspiration" (ET for short) is defined as the measure of water use by plants in when water evaporation from soil and transpiration from plants' leaves occurs (as illustrated in *figure 52.0*). From the distribution of evaporation graph as shown above, farmers are able to decide what crops to grow by checking the value and trend of evaporation throughout the year. In *figure 53.0*, a list of common crops grown in California are displayed along with the water amount they needed to sustain. Judging from the amount of water use for each plant, farmers are able to decide which period of evaporation is most favorable for a specific plant to thrive. For example, grain needed about 1 to 2 feet of water application depth to thrive (as stated in *figure 53.0*). Hence, farmers should start planting grain in the beginning or near the end of the year as evaporation rate is high during these two periods, and grain does not need that much of water to sustain.

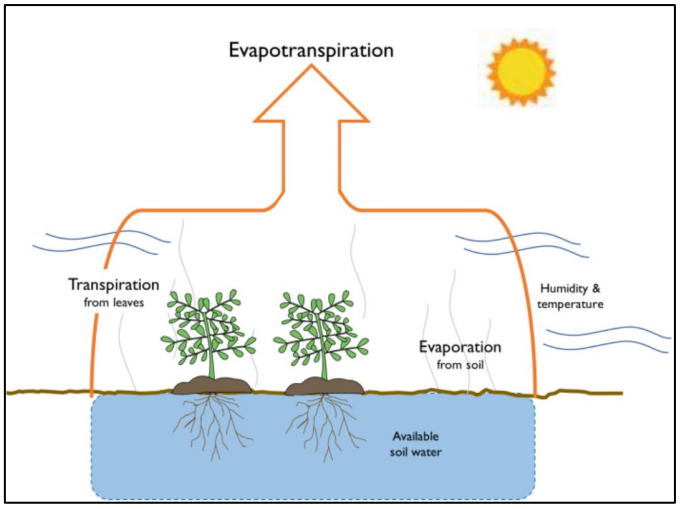

Figure 52.0 Evapotranspiration (Student Materials, 2021)

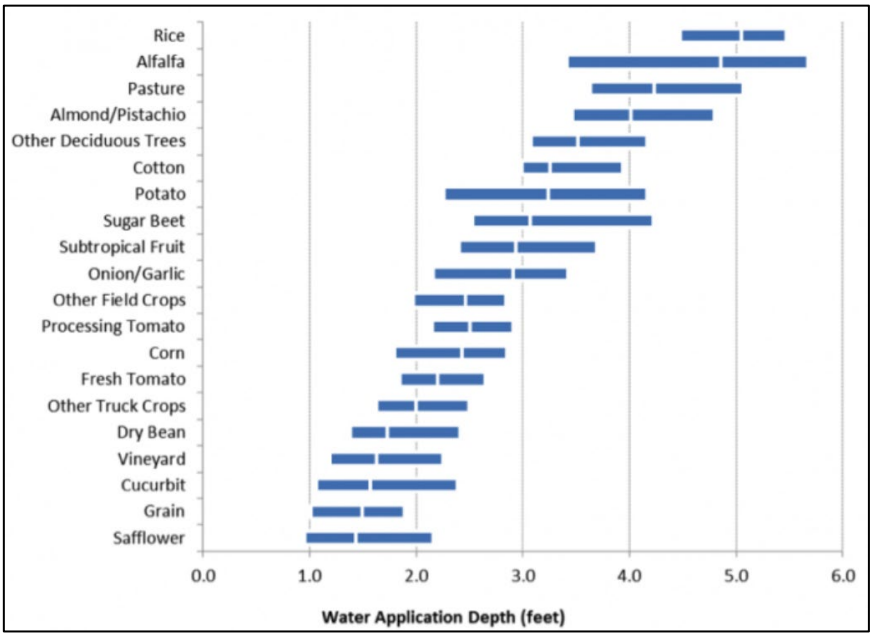

Figure 53.0 Water application depth needed for common crops in US (Student Materials, 2021)

Question 5: What are the factors that affect weather prediction?

Code snippet 5.1

```
#-----------------------------------------------------------------------------------------------------------
#QUESTION5: WEATHER PREDICTION TO DETERMINE ACTIVITIES
#-----------------------------------------------------------------------------------------------------------
#ANALYSIS5.1: Frequency of days with change in weather (Probability of rainy or sunny days?)
#VARIABLES USED: RToday, RTmr
#CONCLUSION: Mostly sunny throughout the year as indicated by No_No relation
#CONCLUSION: Same frequency for having rainy days in 1 out of 2 days as indicated by Yes_No and No_Yes

#selecting specific columns for Question5
Q5 <- data.frame(subset (weather, select = c("RiskMM", "RToday", "RTmr", "C9", "C3")))
View(Q5)

#selecting values to reflect change in weather
Rain_Today = Q5$RToday
Rain_Tmr = Q5$RTmr
RainRelations = paste(Rain_Today,Rain_Tmr,sep="_")
table(RainRelations)

#generating bar chart to frequency of days with change in weather
barplot(table(RainRelations), col = "pink", ylim = c(0, 300),
        main = "Count of Days with Change in Weather",
        xlab = "Categories of Days with Change in Weather",
        ylab = "Count")
```

Figure 54.0 Code snippet

A bar chart is plotted to determine frequency of weather change. Weather change variable is binned into four different categories: "*No_No*", "*No_Yes*", "*Yes_No*" and "*Yes_Yes*". These categories are derived from values of variables *RToday* and *RTmr* (representative of probability of Rain Today and Rain Tomorrow).

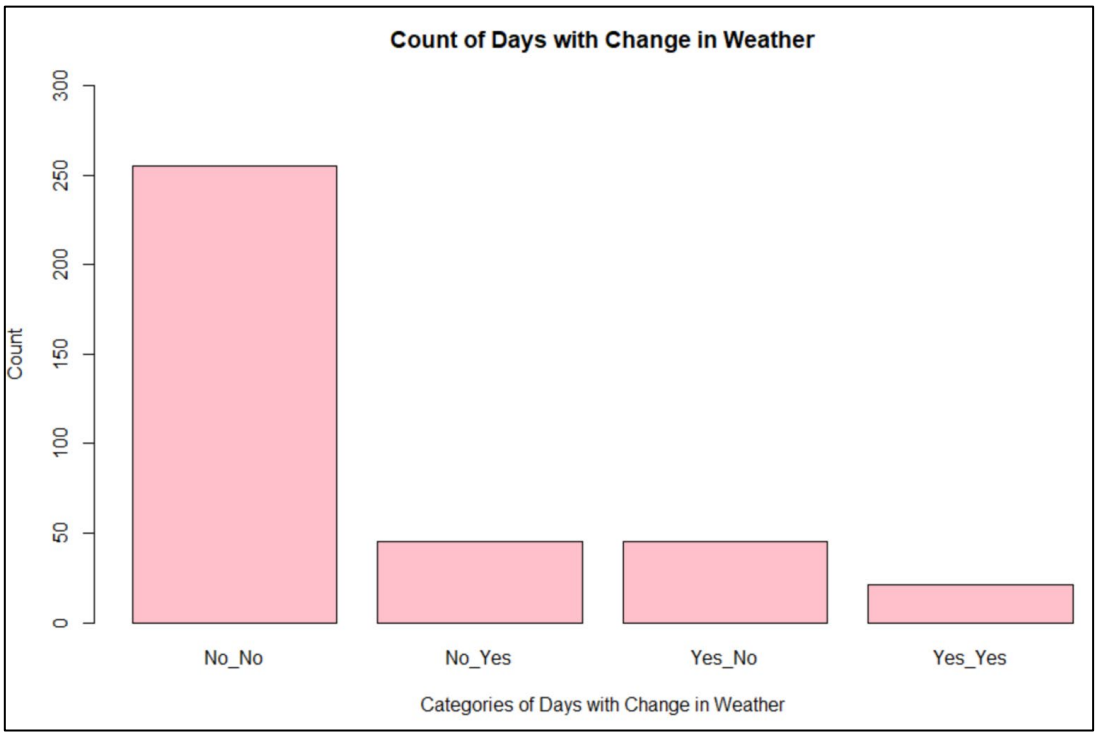

Figure 55.0 Bar chart

Analysis 5.1

From the bar chart above, probability of not raining tomorrow is way higher than probability of raining tomorrow. This is because frequency of category "*No_No*" and "*Yes_No*" combined together reaches a value of 300, while frequency of category "*No_Yes*" and "*Yes_Yes*" combined together reaches a value of 65 only.

Findings 5.1

There is no distinct indication or implication as to value of variable *RTmr* is dependent on which factor as even if value of *RToday* is "Yes", value of *RTmr* could still be either "No" or "Yes"; and the same results are applied given that the value of *RToday* is "No". Hence, we can conclude that probability of raining tomorrow is not related to the current state of today's weather. However, probability of having sunny days is still way higher than rainy days based on the frequency difference as mentioned above.

Code snippet 5.2

```
#-----------------------------------------------------------------------------------------------------
#ANALYSIS5.2: Distribution of cloud cover in oktas at specific time intervals (9am & 3pm)
#VARIABLES USED: C9, C3
#CONCLUSION: Sky is often clear, since cloud okta of 1 has highest frequency in both time intervals
#CONCLUSION: Sky is cloudier at 3pm, seems to have increase in okta values (higher chance of raining?)

#generating histogram for cloud distribution at 9am
Q5_hist <- ggplot(data=Q5, aes(C9)) +
  geom_histogram(col="white", fill="#D16103", binwidth = 1) +
  scale_x_continuous(breaks = seq(0, 8, by = 1)) +
  labs(title="Distribution of Cloud Cover in Oktas at 9am") +
  labs(x="Cloud (in Oktas)", y="Count")
Q5_hist

#generating histogram for cloud distribution at 3pm
Q5_hist2 <- ggplot(data=Q5, aes(C3)) +
  geom_histogram(col="white", fill="#D16103", binwidth = 1) +
  scale_x_continuous(breaks = seq(0, 8, by = 1)) +
  labs(title="Distribution of Cloud Cover in Oktas at 3pm") +
  labs(x="Cloud (in Oktas)", y="Count")
Q5_hist2
```

Figure 56.0 Code snippet

Histograms of cloud cover at different time intervals (9am and 3pm) are plotted.

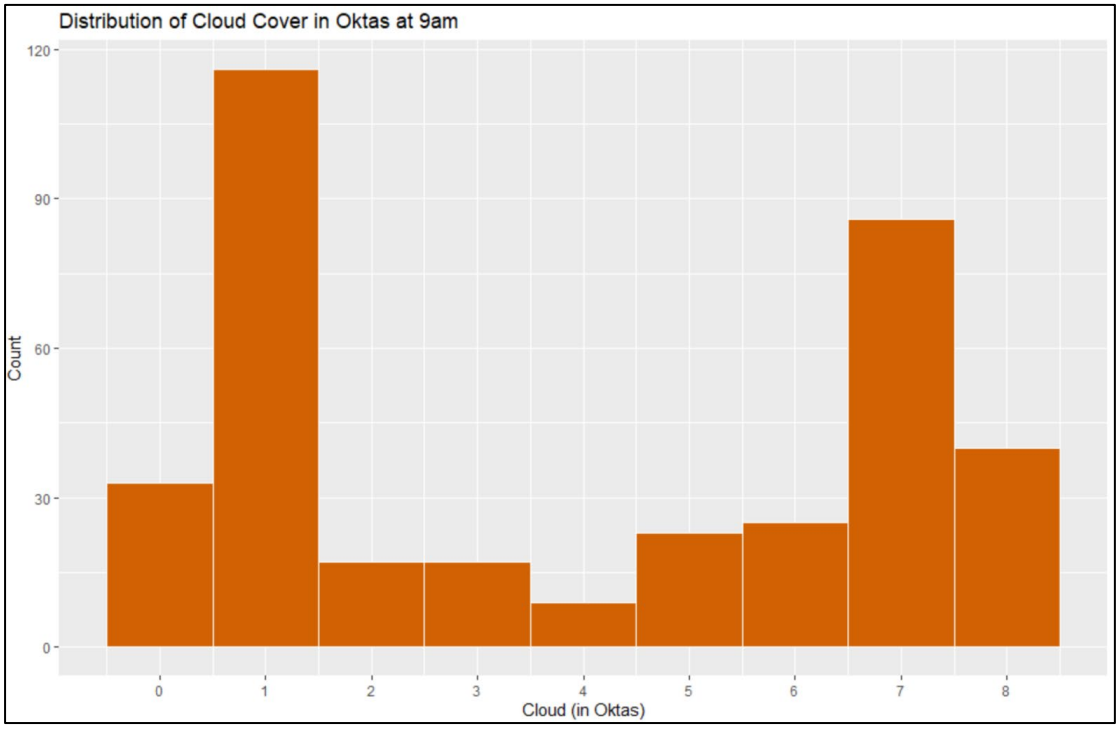

Figure 57.0 Distribution of cloud cover at 9am

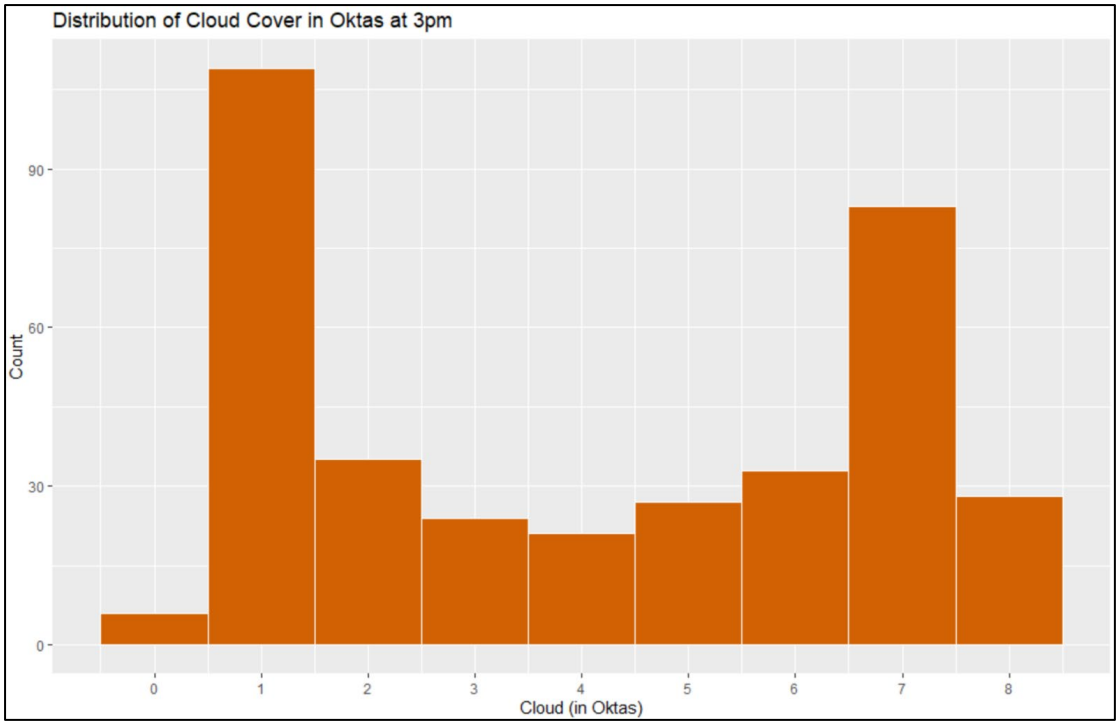

Figure 58.0 Distribution of cloud cover at 3pm

Analysis 5.2

Despite the difference in time intervals, cloud cover with a value of 1 okta has the highest frequency while cloud cover with value of 7 oktas have the second highest frequency. However, different time intervals have varied cloud cover okta values that have the lowest frequency. Cloud cover with okta value of 4 has the lowest frequency at 9am while cloud cover with the okta value of 0 has the lowest frequency at 9pm.

Findings 5.2

Clouds have heavy impact on both weather and climate; and the precipitation conditions over a particular region is heavily affected by the type and amount of clouds (cloud cover). Clouds are able to both reflect and trap heat from solar radiation of the Sun. In general, higher level clouds in the atmosphere are able to trap heat within Earth and lower level clouds are able to reflect heat back to the atmospheric space and maintain a cool temperature over a region. One of the effects that cloud cover bring is that it could reduce the cooling effect in the process of radiative cooling; where solar

heat absorbed by the ground during the day is released at night (Society, 2021). This phenomenon will increase temperature of the ground's surface and upset the balance of air pressure and humidity in air. With the shift of changes in place, intensity of weather will be affected as well. For example, change in temperature and humidity are higher when value of cloud cover is high. The greater the imbalance exists amid different weather controlling factors, the greater intensity of the weather (rainy or stormy weathers) will be.

Code snippet 5.3

```
#-------------------------------------------------------------------------------------------------------------
#ANALYSIS5.3: Proportion of evaporation under different sunshine intensity
#VARIABLES USED: E, SS
#CONCLUSION: Higher sunshine intensity = higher evaporation
#CONCLUSION: Low sunshine intensity but high evaporation = high precipitaion
evap_gap <- c(0,2,4,6,8,10,12,14)
evap_label <- c("0~2","2~4","4~6","6~8","8~10","10~12","12~14")
evap_group <- cut(as.numeric(E), breaks = evap_gap, include.lowest = TRUE, right = TRUE, labels = evap_label)
evap_group
combine1 = cbind(E,as.data.frame(evap_group))

SS_gap <- c(0,2,4,6,8,10,12,14)
SS_label <- c("0~2","2~4","4~6","6~8","8~10","10~12","12~14")
SS_group <- cut(as.numeric(Sunshine_Fill), breaks = SS_gap, include.lowest = TRUE, right = TRUE, labels = SS_label)
SS_group
combine2 = cbind(test,as.data.frame(SS_group))
combine2

Q5_hist3 = ggplot(combine2, aes(x=SS_group, color=evap_group,fill=evap_group)) +
  geom_histogram(, position="identity",stat="count")+
  labs(x = "Sunshine intensity ", y = "Frequency",
       title = "Proportion of Evporation Groups Under Different Sunshine Intensity Category")
Q5_hist3
```

Figure 58.0 Code snippet

Rate of evaporation and sunshine hours are both binned into to promote better data visibility. Both variables are binned into seven different categories with an interval of two into variables *evap_group* and *SS_group*. A stacked bar chart is then generated where proportion of different evaporation groups are shown under each sunshine intensity group (grouped by different evaporation groups).

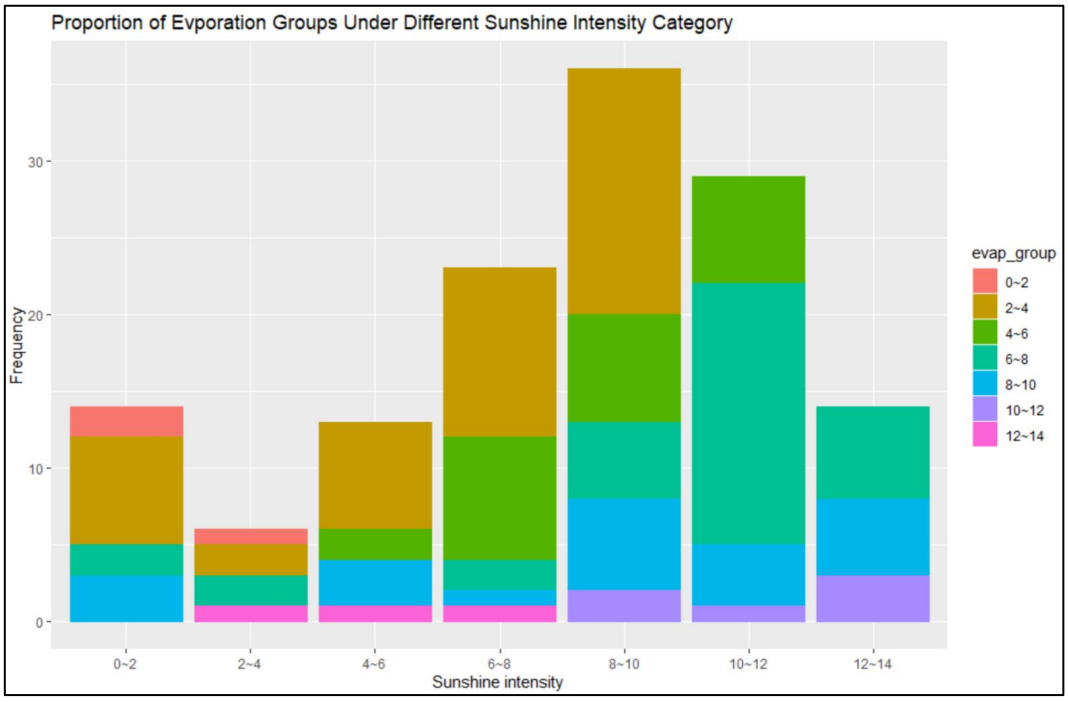

Figure 58.0 Stacked bar chart

Analysis 5.3

We could see that most evaporation takes place when number of sunshine hours is within the range of 8 to 10 while the least evaporation that takes place is when number of sunshine hours is within the range of 2 to 4. The greatest proportion of evaporation occurring is of 2 to 4 (as most bars in *figure 58.0* is occupied by brown colour) while the least evaporation group that occurs is 12 to 14 (indicated by pink layers in the bars).

Findings 5.3

Based on the diagram of stacked bar chart in *figure 58.0*, we can conclude that number of sunshine hours affect the amount of evaporation takes place in a day, as supported by the increasing frequency in evaporation in accordance to sunshine hours. There is a pattern of higher degree of evaporation happening in bars of sunshine hours with higher values. However, we also spotted outliers in bars under sunshine hours of 2 to 4, 4 to 6 and 6 to 8 as there are evaporation groups of the value 12 to 14 occupying the bars. This could be justified with the explanation that as evaporation increases, this process itself also contributes to the operation of water cycle where water is evaporated into water vapour which rise up to the air. After that, these water vapour will be cooled and condensed into clouds before they form precipitation. When formation of clouds increases, it makes sense that sunshine hours will be affected as:

1. More clouds are formed and managed to prevent sunlight from passing through cloud cover.
2. More precipitation is formed and induces rainy or stormy weathers; and the absence of sunny weather explains the reduced number of sunlight hours.

EXTRA FEATURES

1. position = “identity” & alpha = 0.5

```
geom_histogram(bins = 50, binwidth = 1, position = "identity", alpha = 0.5)
```

Figure 59.0

The alpha argument is set to a value of 0.5 to make the histogram appear as semi-transparent and allow more visibility in histogram distribution pattern.

2. scale_x_continuous() & scale_y_continuous()

```
scale_x_continuous(name = "Wind Speed at 9am (mph)",
                   breaks = seq(0, 41, 5), limits = c(0, 41)) +
scale_y_continuous(name = "Count") +
```

Figure 60.0

The scale_x_continuous() and scale_y_continuous() functions are used to control the scales of x and y axis for better data readability.

3. geom_smooth() & method = “lm”

```
Q1_scatterplot <- ggplot(Q1) +
  geom_point(aes(x=T9,y=P9, colour = "9AM")) +
  geom_smooth(aes(x=T9,y=P9, colour = "9AM"), method = "lm") +
  geom_point(aes(x=T3,y=P3, colour = "3PM")) +
  geom_smooth(aes(x=T3,y=P3, colour = "3PM"), method = "lm") +
```

Figure 61.0

The geom_smooth() function is used to adding smoothed regression lines by passing the argument of method = “lm” within the function.

4. scale_colour_manual()

```
scale_colour_manual(name="Time",values=cols,
                    guide = guide_legend(override.aes=aes(fill=NA)))
```

Figure 62.0

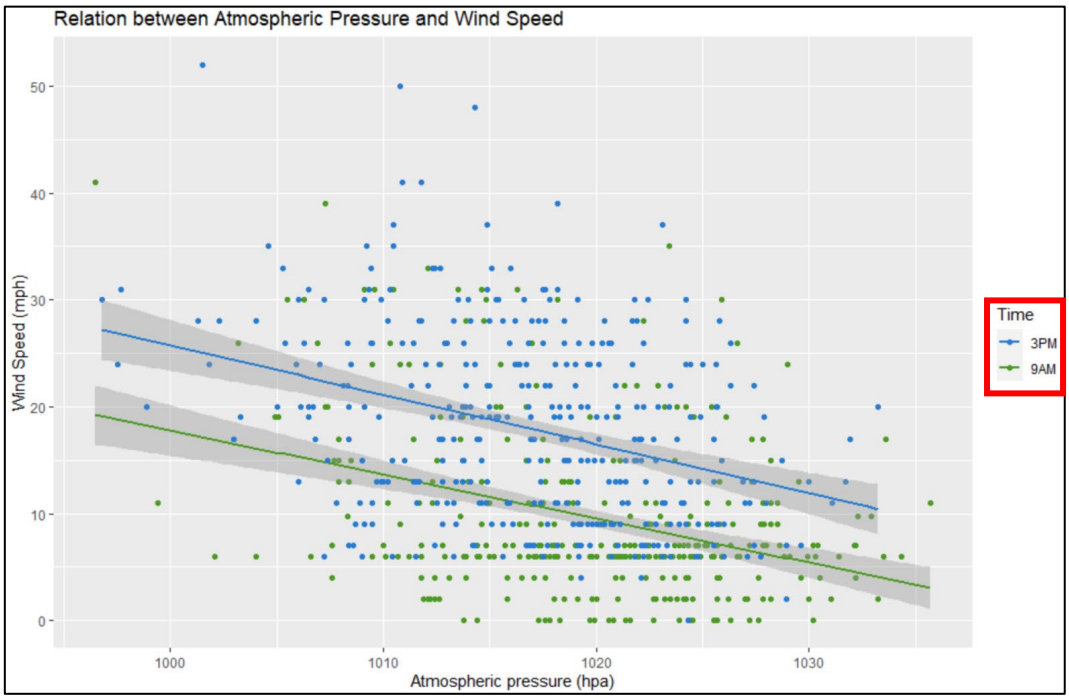

Figure 63.0

The scale_colour_manual() function is used to create a discrete scale because I wanted to customize the default legend settings to my own liking by renaming the legend title as "Time" and legend components as the "9AM" and "3PM" as shown in figure and figure .

5. cut()

```
SS_gap <- c(0, 3, 6, 9, 14)
SS_label <- c("Low", "Medium", "High", "Very High")
SS_group <- cut(Q2$SS, breaks = SS_gap, include.lowest = TRUE, right = TRUE, labels = SS_label)
Q2_SS <- SS_group
```

Figure 64.0

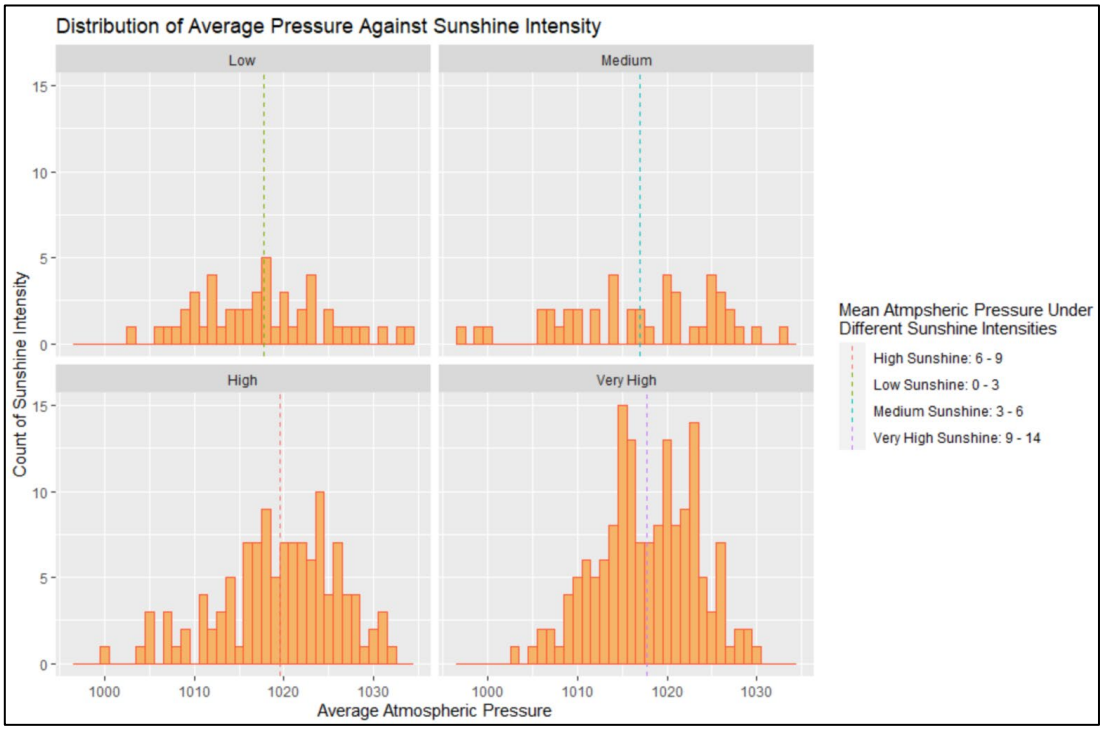

Figure 65.0

The cut() function is used to bin certain dataset column values into different categories for better data visibility as shown in *figure* and *figure* where sunshine hours are binned into four different categories as shown in the legend.

6. geom_vline()

```
Q2_histogram + geom_vline(data = mu, aes(xintercept = grp.mean, color=legend2), linetype="dashed") +
                          labs(title="Distribution of Average Pressure Against Sunshine Intensity",
                          x="Average Atmospheric Pressure", y = "Count of Sunshine Intensity") +
                          guides(col = guide_legend("Mean Atmpsheric Pressure Under\nDifferent Sunshine Intensities"))
```

Figure 66.0

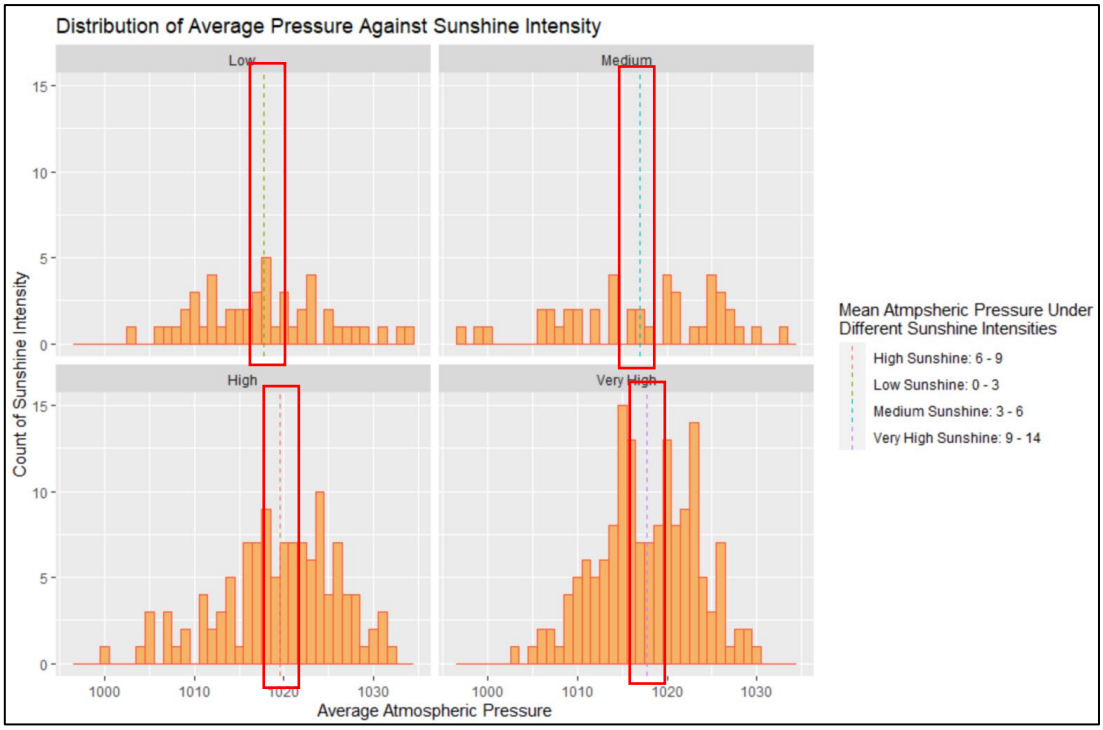

Figure 67.0

The geom_vline() function is used to draw a horizontal line to indicate mean value (note the dashed line in *figure 67.0*).

7. funn.y = mean, geom= “point”, shape = 23

```
Q2_boxplot <- ggplot(Q2, aes(x = Q2_SS, y = MaxT)) +
                    geom_boxplot(aes(color = factor(Q2_SS))) +
                      stat_summary(funn.y = mean, geom="point", shape=23, size=6)
```

Figure 68.0

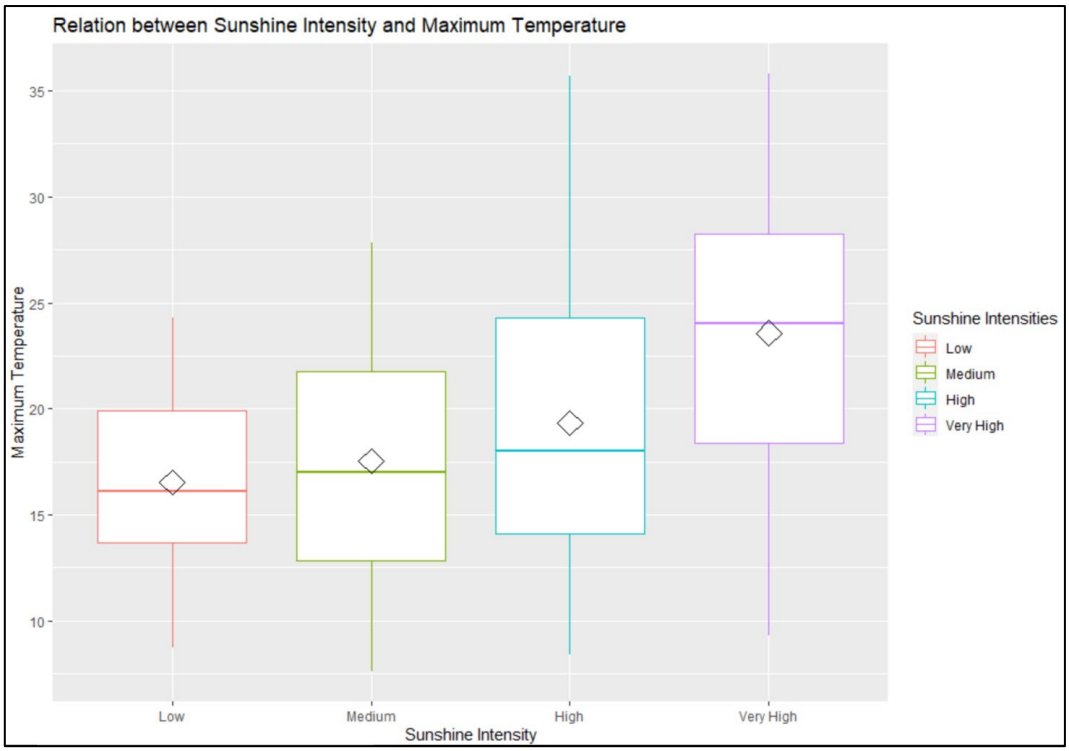

Figure 69.0

The arguments passed into the stat_summary() function is used to draw the diamond boxes (representatives as medians) in each box plot.

8. geom_hline()

```
Q4_line = ggplot(YearEvap,aes(x=day,y=E))+
            geom_line()+
            geom_hline(aes(yintercept=mean(E)), color="green", linetype="dashed", size=1) +
            labs(title="Distribution of Evaporation") +
            labs(x="Day", y="Evaporation (mm)")
```

Figure 70.0

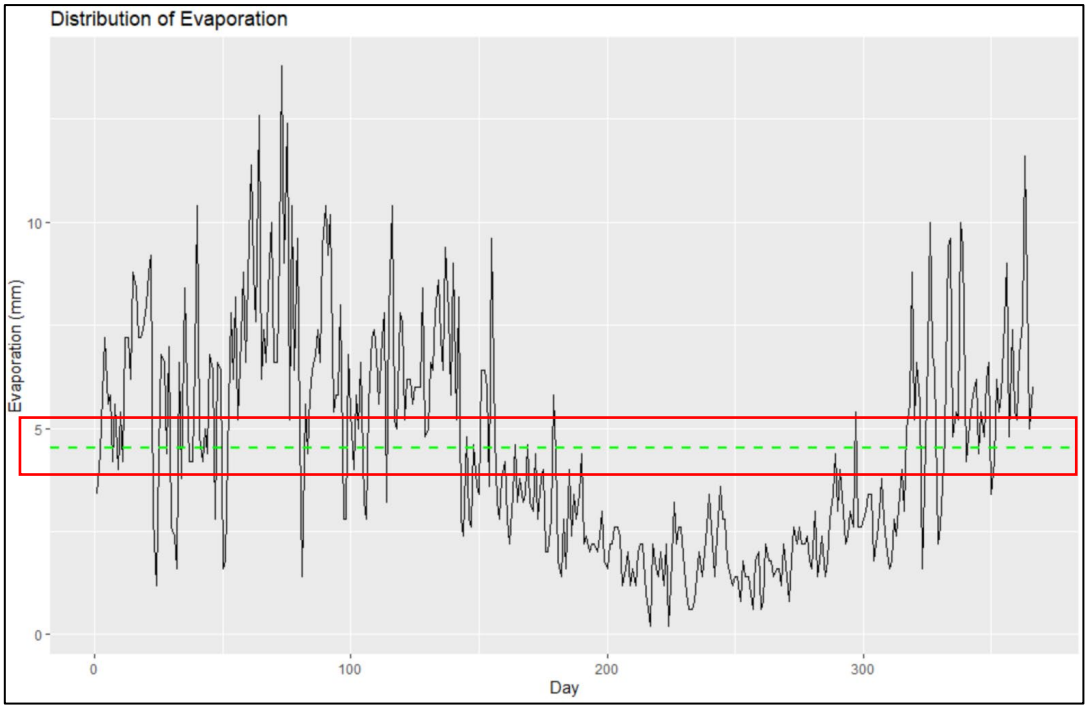

Figure 71.0

The geom_hline() function is used to add horizontal line into graph where in this context, the horizontal line is used to represent the mean line as shown in *figure 71.0.*

CONCLUSION

All of the questions stated in introduction has been answered with the generated graphs supported with in-dept analysis and findings. Assumptions are made based on personal speculations with the addition of external help from available online resources.

REFERENCES

McPherson, J. (2021) *Using the Data Viewer*. [Online] Available at: https://support.rstudio.com/hc/en-us/articles/205175388-Using-the-Data-Viewer

[Accessed: 20 May 2021].

Swcarpentry.github.io. (2021) *Reading and Writing CSV Files – Programming with R*. [Online] Available at: https://swcarpentry.github.io/r-novice-inflammation/11-supp-read-write-csv/ [Accessed: 20 May 2021].

GeeksforGeeks. (2020) *Create a Tabular representation of Data in R Programming - table() Function - GeeksforGeeks*. [Online] Available at: https://www.geeksforgeeks.org/create-a-tabular-representation-of-data-in-r-programming-table-function/#:~:text=table()%20function%20in%20R,the%20form%20of%20a%20table.

[Accessed: 20 May 2021].

Rdocumentation.org. (2021) *names function - RDocumentation*. [Online] Available at: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/names [Accessed: 20 May 2021].

Eia.gov. (2021) *Where wind power is harnessed - U.S. Energy Information Administration (EIA)*. [Online] Available at: https://www.eia.gov/energyexplained/wind/where-wind-power-is-harnessed.php#:~:text=Locations%20of%20U.S.%20wind%20power,338%20billion%20kilowatthours%20(kWh).&text=The%20five%20states%20with%20the,Oklahoma%2C%20Kansas%2C%20and%20Illinois. [Accessed: 22 May 2021].

Irishenvironment.com. 2021. *» Installed Capacity*. [Online] Available at: https://www.irishenvironment.com/iepedia/installed-capacity/#:~:text=For%20wind%20turbines%2C%20it%20describes,megawatts%20(%3D1%20million%20watts). [Accessed: 22 May 2021].

Windexchange.energy.gov. (2021) *WINDExchange: U.S. Installed and Potential Wind Power Capacity and Generation*. [Online] Available at: https://windexchange.energy.gov/maps-data/321[Accessed: 22 May 2021].

Constructionreviewonline.com. (2021) *Top 10 largest wind farms in the world.* [Online] Available at: https://constructionreviewonline.com/biggest-projects/top-10-largest-wind-farms-in-the-world/#:~:text=Jiuquan%20wind%20Power%20Base%20is,Xinjiang%20Provinces%20in%20Gansu%2C%20China. [Accessed: 22 May 2021].

Eia.gov. (2021) *Most wind capacity in the United States is designed for a medium wind speed environment* [Online] Available at: https://www.eia.gov/todayinenergy/detail.php?id=41474 [Accessed: 22 May 2021].

Hindawi.com. 2021. *Table 1 | Effect of Wind Turbine Classes on the Electricity Production of Wind Farms in Cyprus Island*. [Online] Available at: https://www.hindawi.com/journals/cpis/2013/750958/tab1/ [Accessed: 22 May 2021].

Gomez, M. and Lundquist, J. (2020) 'The effect of wind direction shear on turbine performance in a wind farm in central Iowa', Wind Energ.Sc.Discuss., pp. 134

https://www.nrel.gov/docs/fy20osti/76100.pdf

A.R.,S., Pandey, M., Sunil, N., N.S., S., Mugundhan, V. and Velamati, R. (2016) 'Numerical study of effect of pitch angle on performance characteristics of a HAWT' , *Engineering Science and Technology, an International Journal*, Volume 19(1), pp. 637.

https://www.sciencedirect.com/science/article/pii/S221509861500155X#:~:text=2.-,For%20a%20given%20wind%20velocity%2C%20there%20is%20an%20optimum%20pitch,in%20%3D%2025.1%20m%2Fs.

Gellert, A. (2021) *How Does Pressure Affect Wind?*. [Online] Available at: https://sciencing.com/pressure-affect-wind-23262.html [Accessed 24 May 2021].

BOBBY. (2014) *All You Should Know About Wind Farms - News about Energy Storage, Batteries, Climate Change and the Environment*. [Online] Available at: http://www.upsbatterycenter.com/blog/know-wind-farms/ [Accessed: 24 May 2021].

Danish Wind Industry Association. (2021) *The Energy in the Wind: Air Density and Rotor Area*. [Online] Available at: http://xn--drmstrre-64ad.dk/wp-content/wind/miller/windpower%20web/en/tour/wres/enerwind.htm#:~:text=The%20kinetic%20energy%20in%20the,is%20received%20by%20the%20turbine.&text=At%

20high%20altitudes%2C%20(in%20mountains,the%20air%20is%20less%20dense. [Accessed: 24 May 2021].

Writer, S., (2021) *How Does Atmospheric Temperature Affect Air Pressure?*. [Online] Reference.com. Available at: https://www.reference.com/science/temperature-affect-air-pressure-90b37da760fa9d12 [Accessed: 24 May 2021].

Nrel.gov. (2021) *Solar Resource Data, Tools, and Maps*. [Online] Available at: https://www.nrel.gov/gis/solar.html [Accessed: 25 May 2018].

Sciencedirect.com. (2021) *Direct Normal Irradiation - an overview | ScienceDirect Topics*. [Online] Available at: https://www.sciencedirect.com/topics/engineering/direct-normal-irradiation#:~:text=Direct%20Normal%20Irradiation%20(DNI)%20is,current%20position%20in%20the%20sky. [Accessed: 25 May 2021].

Society, N., (2021) *Atmospheric Pressure*. [Online] National Geographic Society. Available at: https://www.nationalgeographic.org/encyclopedia/atmospheric-pressure/#:~:text=Atmospheric%20pressure%20is%20an%20indicator,lead%20to%20fair%2C%20calm%20weather. [Accessed: 25 May 2021].

CED Greentech. (2021) *How Does Heat Affect Solar Panel Efficiencies?*. [Online] Available at: https://www.cedgreentech.com/article/how-does-heat-affect-solar-panel-efficiencies [Accessed: 25 May 2021].

Renvu.com. (2021) *How Temperature Affects Solar Panel Efficiency*. [online] Available at: https://www.renvu.com/Learn/How-Temperature-Affects-Solar-Panel-Efficiency [Accessed: 25 May 2021].

Eldoradoweather.com. (2021) *United States 24 Hour Cloud Cover Percentage Forecast Map*. [Online] Available at: https://www.eldoradoweather.com/forecast/graphical-forecast/sky24hr.html [Accessed: 25 May 2021].

Met Office. (2021) *How we measure cloud*. [Online] Available at: https://www.metoffice.gov.uk/weather/guides/observations/how-we-measure-cloud [Accessed: 25 May 2021].

RV Solar Solution. (2021) *EFFECT OF SHADING ON SOLAR SYSTEM AND BEST WAY TO HANDLE THEM.* [Online] Available at: https://rvsolarsolution.com/effect-of-shading-on-solar-system-and-best-way-to-handle-them/ [Accessed: 25 May 2021].

Lipsett, L. (2012) *Storms, Floods, and Droughts*. [Online] Woods Hole Oceanographic Institution. Available at: https://www.whoi.edu/oceanus/feature/storms-floods-and-droughts/ [Accessed: 25 May 2021].

Trees-energy-conservation.extension.org. (2021) *Understanding and determining prevailing winds – Trees for Energy Conservation*. [Online] Available at: https://trees-energy-conservation.extension.org/understanding-and-determining-prevailing-winds/ [Accessed: 25 May 2021].

Society, N. (2021) *Wind*. [Online] National Geographic Society. Available at: https://www.nationalgeographic.org/encyclopedia/wind/ [Accessed: 25 May 2021].

Gpm.nasa.gov. (2021) *How do Hurricanes Form? | Precipitation Education*. [Online] Available at: https://gpm.nasa.gov/education/articles/how-do-hurricanes-form [Accessed: 25 May 2021].

Marchigiani, R., Gordy, S., Cipolla, J., Adams, R. C., Evans, D. C., Stehly, C., Galwankar, S., Russell, S., Marco, A. P., Kman, N., Bhoi, S., Stawicki, S. P., & Papadimos, T. J. (2013) 'Wind disasters: A comprehensive review of current management strategies', *International journal of critical illness and injury science*, 3(2), pp. 130–142.

Weather.gov. (2021) *Wind Threat Description*. [Online] Available at: https://www.weather.gov/mlb/seasonal_wind_threat#:~:text=%22Damaging%20high%20wind%22%20with%20sustained,with%20a%20high%20wind%20warning.&text=%22A%20High%20Threat%20to%20Life,of%2040%20to%2057%20mph. [Accessed: 25 May 2021].

Polygongroup.com. (2021) *How Humidity Affects the Growth of Plants*. [Online] Available at: https://www.polygongroup.com/en-US/blog/how-humidity-affects-the-growth-of-

plants/#:~:text=When%20conditions%20are%20too%20humid,and%20thrive%20in%20moist%20soil. [Accessed: 25 May 2021].

Encyclopedia Britannica. (2021) *Plant disease - Epiphytotics*. [Online] Available at: https://www.britannica.com/science/plant-disease/Epiphytotics [Accessed: 25 May 2021].

Pthorticulture.com. (2021) *How Does Humidity Influence Crop Quality? | PRO-MIX*. [Online] Available at: https://www.pthorticulture.com/en/training-center/how-does-humidity-influence-crop-quality/ [Accessed: 25 May 2021].

Society, N. (2021) *Humidity*. [Online] National Geographic Society. Available at: https://www.nationalgeographic.org/encyclopedia/humidity/#:~:text=Humidity%20is%20the%20amount%20of,usually%20explained%20as%20relative%20humidity

[Accessed: 25 May 2021].

United States Environmental Protection Agency. (2021) *Climate Impacts on Agriculture and Food Supply | Climate Change Impacts | US EPA*. [Online] Available at: https://19january2017snapshot.epa.gov/climate-impacts/climate-impacts-agriculture-and-food-supply_.html [Accessed: 25 May 2021].

Student Materials. 2021. *Water Sources for Crops*. [Online] Available at: https://serc.carleton.edu/integrate/teaching_materials/food_supply/student_materials/1093 [Accessed: 27 May 2021].

Student Materials. (2021) *Evapotranspiration and Crop Water Use*. [Online] Available at: https://serc.carleton.edu/integrate/teaching_materials/food_supply/student_materials/1091 [Accessed: 27 May 2021].

Society. (2021) *Cloud Cover*. [Online] National Geographic Society. Available at: https://www.nationalgeographic.org/encyclopedia/cloud-cover/ [Accessed: 27 May 2021].