Introduction to Data Scientist 11372 (UG)

Assignment 1 – Data Wrangling and Exploration

Tuan Anh (Vincent) Nguyen – u3196825

Part A:

Q1 + Q2:

```
> # 1. Load these file into your working dir, one by one
> data_dir <- "data"
> files <- list.files(path=data_dir, pattern = "*.csv", full.names = TRUE)
> files
 [1] "data/201808.csv" "data/201809.csv" "data/201810.csv" "data/201811.csv" "data/201812.csv" "data/201901.csv" "data/201902.csv"
 [8] "data/201903.csv" "data/201904.csv" "data/201905.csv" "data/201906.csv" "data/201907.csv" "data/201908.csv" "data/201909.csv"
[15] "data/201910.csv" "data/201911.csv" "data/201912.csv" "data/202001.csv" "data/202002.csv"
> # 2. Concatenate all the data of these files into 1 data frame. May use loop statement to archieve that
> # read the files one by one and append the new rows (skip the first 7 rows),
> # change the date format of Column "Date"
> df <- data.frame()
> for (i in 1:length(files)){
+   temp <- read_csv(files[i], na='-',skip = 7,col_types = cols(Date = col_date(format = "%d/%m/%Y")))
+   df <- rbind(df, temp)
+ }
> df
# A tibble: 578 x 21
   Date       `Minimum temper~ `Maximum temper~ `Rainfall (mm)` `Evaporation (m~ `Sunshine (hour~ `Direction of m~ `Speed of maxim~
   <date>                <dbl>            <dbl>           <dbl> <lgl>           <lgl>           <chr>                       <dbl>
 1 2018-08-01              7.6             15.4               0 NA              NA              NW                             54
 2 2018-08-02             -3.8             14.3               0 NA              NA              NNW                            26
 3 2018-08-03             -3.6             19.5               0 NA              NA              NW                             72
 4 2018-08-04              3.7             12.8              13.8 NA              NA              NNW                            54
 5 2018-08-05             -1               15                 0 NA              NA              NW                             43
 6 2018-08-06              1.2             13.7               0 NA              NA              NW                             61
 7 2018-08-07              2.4              9.7               6.6 NA              NA              WNW                            61
 8 2018-08-08              2.6             12.1               0 NA              NA              WNW                            70
 9 2018-08-09              1.6             13.7               0 NA              NA              NNW                            30
10 2018-08-10             -2.5             15.6               0.2 NA              NA              NNW                            43
# ... with 568 more rows, and 13 more variables: `Time of maximum wind gust` <time>, `9am Temperature` <dbl>, `9am relative humidity
#   (%)` <dbl>, `9am cloud amount (oktas)` <dbl>, `9am wind direction` <chr>, `9am wind speed (km/h)` <chr>, `9am MSL pressure
#   (hPa)` <dbl>, `3pm Temperature` <dbl>, `3pm relative humidity (%)` <dbl>, `3pm cloud amount (oktas)` <dbl>, `3pm wind
#   direction` <chr>, `3pm wind speed (km/h)` <chr>, `3pm MSL pressure (hPa)` <dbl>
> |
```

Q3:

```
> # 3. Check for problem while loading and parsing the data
> # check for problems
> # Inspect the structure of data project
> spec(df)
cols(
  Date = col_date(format = "%d/%m/%Y"),
  `Minimum temperature` = col_double(),
  `Maximum temperature` = col_double(),
  `Rainfall (mm)` = col_double(),
  `Evaporation (mm)` = col_logical(),
  `Sunshine (hours)` = col_logical(),
  `Direction of maximum wind gust` = col_character(),
  `Speed of maximum wind gust (km/h)` = col_double(),
  `Time of maximum wind gust` = col_time(format = ""),
  `9am Temperature` = col_double(),
  `9am relative humidity (%)` = col_double(),
  `9am cloud amount (oktas)` = col_double(),
  `9am wind direction` = col_character(),
  `9am wind speed (km/h)` = col_character(),
  `9am MSL pressure (hPa)` = col_double(),
  `3pm Temperature` = col_double(),
  `3pm relative humidity (%)` = col_double(),
  `3pm cloud amount (oktas)` = col_double(),
  `3pm wind direction` = col_character(),
  `3pm wind speed (km/h)` = col_double(),
  `3pm MSL pressure (hPa)` = col_double()
)
> problems(df)
[1] row      col      expected actual
<0 rows> (or 0-length row.names)
> # assert that there is no problem
> assertthat::assert_that(nrow(problems(df)) == 0,
+                         msg="There is still problem/s, which you need to fix first")
[1] TRUE
>
```

Note: It's been quite weird after I just my code a little bit it did not shown any problems with the input data, but I know it is still there. As `9 am wind speed` supposed to be numeric not char.

```
> problems(df)
[1] row      col      expected actual
<0 rows> (or 0-length row.names)
> # assert that there is no problem
> assertthat::assert_that(nrow(problems(df)) == 0,
+                         msg="There is still problem/s, which you need to fix first")
[1] TRUE
> df$`3pm wind speed (km/h)` <- gsub("Calm", "0", df$`3pm wind speed (km/h)`) # Change Calm -> 0
> df$`9am wind speed (km/h)` <- gsub("Calm", "0", df$`9am wind speed (km/h)`) # Change Calm -> 0
> df$`3pm wind speed (km/h)` <- sapply(df$`3pm wind speed (km/h)`, as.numeric) # Convert the columns from chr to numeric
> df$`9am wind speed (km/h)` <- sapply(df$`9am wind speed (km/h)`, as.numeric) # Convert the columns from chr to numeric
> # assert that there is no problem
> assertthat::assert_that(nrow(problems(df)) == 0,
+                         msg="There is still problem/s, which you need to fix first")
[1] TRUE
>
```

This code used to change all "Calm in these 2 columns as 0, according to the reference document from BOM, ref: http://www.bom.gov.au/marine/knowledge-centre/reference/wind.shtml

Part B:

Q1: Remove from 21 columns down to 19 columns.

```
> # 1. Remove the variables, which have no data at all (all the records in these variables are NAs)
> df_1 <- df[ , colSums(is.na(df)) < nrow(df) ]
> df_1
# A tibble: 578 x 19
   Date       `Minimum temper~ `Maximum temper~ `Rainfall (mm)` `Direction of m~ `Speed of maxim~ `Time of maximu~ `9am Temperatur~
   <date>              <dbl>           <dbl>           <dbl> <chr>                      <dbl> <time>                       <dbl>
 1 2018-08-01            7.6            15.4               0   NW                            54 01:38                        10.9
 2 2018-08-02           -3.8            14.3               0   NNW                           26 12:23                         3.3
 3 2018-08-03           -3.6            19.5               0   NW                            72 20:56                         3.7
 4 2018-08-04            3.7            12.8              13.8 NNW                           54 04:50                         7.8
 5 2018-08-05           -1              15                 0   NW                            43 12:19                         5
 6 2018-08-06            1.2            13.7               0   NW                            61 10:48                         9.2
 7 2018-08-07            2.4             9.7               6.6 WNW                           61 14:52                         4.9
 8 2018-08-08            2.6            12.1               0   WNW                           70 12:45                         6.5
 9 2018-08-09            1.6            13.7               0   NNW                           30 00:40                         6.3
10 2018-08-10           -2.5            15.6               0.2 NNW                           43 12:16                         0.5
# ... with 568 more rows, and 11 more variables: `9am relative humidity (%)` <dbl>, `9am cloud amount (oktas)` <dbl>, `9am wind
#   direction` <chr>, `9am wind speed (km/h)` <dbl>, `9am MSL pressure (hPa)` <dbl>, `3pm Temperature` <dbl>, `3pm relative humidity
#   (%)` <dbl>, `3pm cloud amount (oktas)` <dbl>, `3pm wind direction` <chr>, `3pm wind speed (km/h)` <dbl>, `3pm MSL pressure
#   (hPa)` <dbl>
>
```

Q2:

```
> # 2. Drop the variables, which have few data  (NAs values are more than 90% of number of records in these variables)
> colSums(is.na(df_1)) # Get the sum of NA values in each table
                          Date              Minimum temperature              Maximum temperature                     Rainfall (mm)
                             0                                0                                0                                 0
   Direction of maximum wind gust Speed of maximum wind gust (km/h)        Time of maximum wind gust                  9am Temperature
                             0                                1                                1                                 0
       9am relative humidity (%)          9am cloud amount (oktas)               9am wind direction             9am wind speed (km/h)
                             0                              290                                0                                 0
          9am MSL pressure (hPa)                  3pm Temperature        3pm relative humidity (%)          3pm cloud amount (oktas)
                             0                                0                                0                               269
              3pm wind direction             3pm wind speed (km/h)           3pm MSL pressure (hPa)
                             0                                0                                0
> df_1 <- df_1[, which(colMeans(!is.na(df_1)) > 0.9)] # remove variables with 90% NAs
> colSums(is.na(df_1))
                          Date              Minimum temperature              Maximum temperature                     Rainfall (mm)
                             0                                0                                0                                 0
   Direction of maximum wind gust Speed of maximum wind gust (km/h)        Time of maximum wind gust                  9am Temperature
                             0                                1                                1                                 0
       9am relative humidity (%)               9am wind direction            9am wind speed (km/h)           9am MSL pressure (hPa)
                             0                                0                                0                                 0
                 3pm Temperature        3pm relative humidity (%)              3pm wind direction             3pm wind speed (km/h)
                             0                                0                                0                                 0
          3pm MSL pressure (hPa)
                             0
>
```

Q3:

```
> # 3. Change the column names to replace space with underscore the "_" character
> colnames(df_1) <- gsub(" ", "_", colnames(df_1))
> colnames(df_1)
 [1] "Date"                          "Minimum_temperature"           "Maximum_temperature"
 [4] "Rainfall_(mm)"                 "Direction_of_maximum_wind_gust" "Speed_of_maximum_wind_gust_(km/h)"
 [7] "Time_of_maximum_wind_gust"     "9am_Temperature"               "9am_relative_humidity_(%)"
[10] "9am_wind_direction"            "9am_wind_speed_(km/h)"         "9am_MSL_pressure_(hPa)"
[13] "3pm_Temperature"               "3pm_relative_humidity_(%)"     "3pm_wind_direction"
[16] "3pm_wind_speed_(km/h)"         "3pm_MSL_pressure_(hPa)"
>
```

Q4 + 5: Note (The Date Columns already been converted to Date while read_csv step already)

```
> # 4. Add 2 news columns (Month, Year) of the data in each file from the column "DATE"
> data_tidier <- df_1 %>%
+   separate(Date, into = c("Year", "Month"), sep = "\\-", convert = TRUE)
Warning message:
Expected 2 pieces. Additional pieces discarded in 578 rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
> data_tidier
# A tibble: 578 x 18
    Year Month Minimum_tempera~ Maximum_tempera~ `Rainfall_(mm)` Direction_of_ma~ `Speed_of_maxim~ Time_of_maximum~ `9am_Temperatur~
   <int> <int>           <dbl>           <dbl>           <dbl> <chr>                      <dbl> <time>                       <dbl>
 1  2018     8             7.6            15.4               0   NW                            54 01:38                        10.9
 2  2018     8            -3.8            14.3               0   NNW                           26 12:23                         3.3
 3  2018     8            -3.6            19.5               0   NW                            72 20:56                         3.7
 4  2018     8             3.7            12.8              13.8 NNW                           54 04:50                         7.8
 5  2018     8            -1              15                 0   NW                            43 12:19                         5
 6  2018     8             1.2            13.7               0   NW                            61 10:48                         9.2
 7  2018     8             2.4             9.7               6.6 WNW                           61 14:52                         4.9
 8  2018     8             2.6            12.1               0   WNW                           70 12:45                         6.5
 9  2018     8             1.6            13.7               0   NNW                           30 00:40                         6.3
10  2018     8            -2.5            15.6               0.2 NNW                           43 12:16                         0.5
# ... with 568 more rows, and 9 more variables: `9am_relative_humidity_(%)` <dbl>, `9am_wind_direction` <chr>,
#   `9am_wind_speed_(km/h)` <dbl>, `9am_MSL_pressure_(hPa)` <dbl>, `3pm_Temperature` <dbl>, `3pm_relative_humidity_(%)` <dbl>,
#   `3pm_wind_direction` <chr>, `3pm_wind_speed_(km/h)` <dbl>, `3pm_MSL_pressure_(hPa)` <dbl>
>
```

Q6:

```
> # 5. Change the type of the "Month" and "Year" columns from Character to Ordinal with levels as the number of months in a year
> data_tidier$Month <- parse_integer(data_tidier$Month)
Error in parse_vector(x, col_integer(), na = na, locale = locale, trim_ws = trim_ws) :
  is.character(x) is not TRUE
> data_tidier$Month <- factor(data_tidier$Month, order = TRUE, levels =c(1:12))
>
> data_tidier$Year <-  parse_integer(data_tidier$Year)
Error in parse_vector(x, col_integer(), na = na, locale = locale, trim_ws = trim_ws) :
  is.character(x) is not TRUE
> data_tidier$Year <- factor(data_tidier$Year, order = TRUE, levels =c(2018, 2019, 2020))
>
> # Check to see if the Ordinal worked! Summary the number of Month and Year.
> summary(data_tidier$Month)
 1  2  3  4  5  6  7  8  9 10 11 12
62 57 31 30 31 30 31 62 60 62 60 62
> summary(data_tidier$Year)
2018 2019 2020
 153  365   60
>
```

Q7:

```
> # 6. For all numeric columns, replace the remaining NAs with the median of values in the columns, if exist
>
> dt <- as_tibble(data_tidier) # Clone from previous step
>
> # I did test out several ways but stuck at keep original df
> # while modify the numeric col only, and it include mean calculating as well
> # Dirty Code for unsolved problem with loop, Sorry if it is hurt your eyes.
> # The code go through all numerical cols and replace NAs value with median of the columns.
> dt$Minimum_temperature[is.na(dt$Minimum_temperature)] <- median(dt$Minimum_temperature, na.rm = TRUE)
> dt$Maximum_temperature[is.na(dt$Maximum_temperature)] <- median(dt$Maximum_temperature, na.rm = TRUE)
> dt$`Rainfall_(mm)`[is.na(dt$`Rainfall_(mm)`)] <- median(dt$`Rainfall_(mm)`, na.rm = TRUE)
> dt$`Speed_of_maximum_wind_gust_(km/h)`[is.na(dt$`Speed_of_maximum_wind_gust_(km/h)`)] <- median(dt$`Speed_of_maximum_wind_gust_(km/h)`, n
a.rm = TRUE)
> dt$`9am_Temperature`[is.na(dt$`9am_Temperature`)] <- median(dt$`9am_Temperature`, na.rm = TRUE)
> dt$`9am_relative_humidity_(%)`[is.na(dt$`9am_relative_humidity_(%)`)] <- median(dt$`9am_relative_humidity_(%)`, na.rm = TRUE)
> #dt$`9am_cloud_amount_(oktas)`[is.na(dt$`9am_cloud_amount_(oktas)`)] <-median(dt$`9am_cloud_amount_(oktas)`, na.rm = TRUE)
> dt$`9am_wind_speed_(km/h)`[is.na(dt$`9am_wind_speed_(km/h)`)] <- median(dt$`9am_wind_speed_(km/h)`, na.rm = TRUE)
> dt$`9am_MSL_pressure_(hPa)`[is.na(dt$`9am_MSL_pressure_(hPa)`)] <- median(dt$`9am_MSL_pressure_(hPa)`, na.rm = TRUE)
> dt$`3pm_Temperature`[is.na(dt$`3pm_Temperature`)] <-  median(dt$`3pm_Temperature`, na.rm = TRUE)
> dt$`3pm_relative_humidity_(%)`[is.na(dt$`3pm_relative_humidity_(%)`)] <- median(dt$`3pm_relative_humidity_(%)`, na.rm = TRUE)
> #dt$`3pm_cloud_amount_(oktas)`[is.na(dt$`3pm_cloud_amount_(oktas)`)] <- median(dt$`3pm_cloud_amount_(oktas)`, na.rm = TRUE)
> dt$`3pm_wind_speed_(km/h)`[is.na(dt$`3pm_wind_speed_(km/h)`)] <- median(dt$`3pm_wind_speed_(km/h)`, na.rm = TRUE)
> dt$`3pm_MSL_pressure_(hPa)`[is.na(dt$`3pm_MSL_pressure_(hPa)`)] <- median(dt$`3pm_MSL_pressure_(hPa)`, na.rm = TRUE)
>
> # check any NA values left in the cols
> dt <- na.omit(dt) # by now, only one row with NA value left, which is fine to remove it
> sum(is.na(dt)) # sum of NAs in data, if = 0 <--> FINISHED WRANGLING
[1] 0
> dt_finished <- as_tibble(dt)
>
```

Quick note for 6: As for Q7 I supposed to have a loop, but for some weird reasons I could not make it work after several hours attempt, so I decided to do it manually to move forward as it took me almost a day to find a way looping through all numeric columns and replace NAs value with median of the columns.

Part C:

Q1:

```
> # 1. Printing the summary (minimum, median, mean, maxiumum) of each of the following variables:
> # Minimum_temperature
> summary(dt_finished$Minimum_temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -6.400   2.000   8.000   7.823  13.600  26.700
> # Maximum_temperature
> summary(dt_finished$Maximum_temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    8.6    16.4    22.9    23.1    28.8    44.0
> # 9am_Temperature
> summary(dt_finished$`9am_Temperature`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.50    9.20   14.70   14.08   18.60   34.50
> # 3pm_Temperature
> summary(dt_finished$`3pm_Temperature`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6.10   14.90   21.30   21.44   27.20   42.30
> # Speed_of_maximum_wind_gust_(km/h)
> summary(dt_finished$`Speed_of_maximum_wind_gust_(km/h)`)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.00   35.00   43.00   44.43   52.00  117.00
>
```

Note: I saw some example have to write the code separately but I found this way more efficient, not sure it is alright to use this interface?

Q2:

```
> # 2. Extracting the average of minimum temperature per month and year.
> dt_finished %>%
+   group_by(Year, Month, Minimum_temperature) %>%
+   summarise(avg_min_temp = mean(Minimum_temperature, na.rm = TRUE))
  avg_min_temp
1     7.822877
> |
```

Q3:

```
> # 3. Extracting the average of maximum temperature per month and year.
> dt_finished %>%
+   group_by(Year, Month, Maximum_temperature) %>%
+   summarise(avg_max_temp = mean(Maximum_temperature, na.rm = TRUE))
  avg_max_temp
1     23.09601
> |
```

Q4:

```
> # 4. Extracting the average of speed of maximum wind gust per direction of maximum wind gust.
> dt_finished %>%
+   group_by(., Direction_of_maximum_wind_gust) %>%
+   summarise(avg_speed_maximum_wind_gust = mean(`Speed_of_maximum_wind_gust_(km/h)`))
  avg_speed_maximum_wind_gust
1                    44.43328
> |
```

Q5:

```
> # 5. Which month has the highest rain fall quality? And in which year?
> highest_rain_fall_by_month <- dt_finished %>%
+   group_by(Month) %>%
+   slice(which.max(`Rainfall_(mm)`)) %>%
+   select(Year, Month, `Rainfall_(mm)`)
>
> # Get the highest rain_fall month from the list
> highest_rain_fall_by_month[which.max(highest_rain_fall_by_month$`Rainfall_(mm)`), ]
# A tibble: 1 x 3
# Groups:   Month [1]
  Year  Month `Rainfall_(mm)`
  <ord> <ord>           <dbl>
1 2020  2                60.4
> |
```

Q6:

```
> # 6. Which months were dry, if any, (i.e. no rainfall at all), And in which year?
> by_month_year_rainfall <- dt_finished %>%
+   group_by(Year, Month) %>%
+   filter(`Rainfall_(mm)`== 0) %>%
+   select(Year, Month) %>%
+   count(.) %>%
+   filter(freq > 24) # Filter months in each year which have more than 24 days without rain
>
> by_month_year_rainfall
  Year Month freq
1 2018     9   25
2 2018    10   25
3 2019     4   29
4 2019     9   25
5 2019    10   27
6 2019    11   28
7 2019    12   29
> |
```

Q7:

```
> # 7. What about the humidity, which month in the ACT
> # has the highest humidity level in the 2019
> humidity_by_month_2019 <- dt_finished %>%
+   group_by(Year, Month) %>%
+   filter(Year == 2019) %>%
+   select(Year ,Month, `9am_relative_humidity_(%)`, `3pm_relative_humidity_(%)`) %>%
+   mutate(avg_humidity = (`9am_relative_humidity_(%)` + `3pm_relative_humidity_(%)`) / 2) %>%
+   slice(which.max(avg_humidity)) %>%
+   summarise(Month = Month[which.max(avg_humidity)],
+             Value = avg_humidity[which.max(avg_humidity)])
> humidity_by_month_2019
  Month Value
1     6    92
> |
```

For this part, I group the year and month then filter to only 2019. Then get the average humidity between 9am and 3pm to then get the maximum of each month then summarise and get the month with most humidity averaging.

Q8:

Note: The code here quite straight forward to get the Min, Max and Average of Temperature, Wind Speed and Humidity per month and quarter so I first have to create a new Quarter table with Ordinal levels of 4 for easier filter and summarise. However, there was 2 small problems with Wind speed and Humidity as it has 2 columns of data for 9am and 3pm so I have to calculate the mean to get the min and max per month and quarter. Which can be quite messy if try to find the Min and Max from each different row within a Month or Year.

The code was really redundant, I have to admit as by the time I going to submit this, I don't have enough time to write function and loop through this to make it less eye-killer.

```
>
> # 8. For 2019, extract the min, max, avg, temperature, wind speed and humidity per month and per quarter in 2019 only.
>
> # Create new quarter column
> month_quarter_2019 <- dt_finished %>%
+   group_by(Year, Month) %>%
+   filter(Year == 2019) %>%
+   mutate(Quarter = ceiling(as.numeric(Month) / 3))
>
> # Ordinal Converter for Quarter
> month_quarter_2019$Quarter <- parse_integer(month_quarter_2019$Quarter)
Error in parse_vector(x, col_integer(), na = na, locale = locale, trim_ws = trim_ws) :
  is.character(x) is not TRUE
> month_quarter_2019$Quarter <- factor(month_quarter_2019$Quarter, order = TRUE, levels =c(1:4))
>
> # Temperature (Min > Max > AVG)
> ## Temperature By Month
> min_temperature_by_month <- month_quarter_2019 %>%
+   group_by(Month) %>%
+   slice(which.min(Minimum_temperature)) %>%
+   select(Month, min_temp_month = Minimum_temperature)
>
> max_temperature_by_month <- month_quarter_2019 %>%
+   group_by(Month) %>%
+   slice(which.max(Maximum_temperature)) %>%
+   select( Month, max_temp_month = Maximum_temperature)
>
> # I calculate the average daily temperature by (min_temp + max_temp) / 2 and get mean by month.
> avg_temperature_by_month <- month_quarter_2019 %>%
+   group_by(Month) %>%
+   mutate(avg_daily_temp = (Maximum_temperature + Minimum_temperature) / 2) %>%
+   aggregate( avg_daily_temp ~ Month, . , mean ) %>%
+   select( Month, avg_temp_month = avg_daily_temp )
>
> # Merge into 1 table
> temp_month <- do.call("cbind", list(min_temperature_by_month, max_temperature_by_month, avg_temperature_by_month))
New names:
* Month -> Month...1
* Month -> Month...3
* Month -> Month...5
> colnames(temp_month)[1] <- "Month"
> temp_month <-  temp_month[c(1, 2, 4, 6)]
>
> ## Temperature By Quarter
> min_temperature_by_quarter <- month_quarter_2019 %>%
+   group_by(Quarter) %>%
+   slice(which.min(Minimum_temperature)) %>%
+   select(Quarter, min_temp_quarter = Minimum_temperature)
>
> max_temperature_by_quarter <- month_quarter_2019 %>%
+   group_by(Quarter) %>%
+   slice(which.max(Maximum_temperature)) %>%
+   select( Quarter, max_temp_quarter = Maximum_temperature)
>
> avg_temperature_by_quarter <- month_quarter_2019 %>%
+   group_by(Quarter) %>%
+   mutate(avg_daily_temp = (Maximum_temperature + Minimum_temperature) / 2) %>%
+   aggregate( avg_daily_temp ~ Quarter, . , mean ) %>%
+   select( Quarter, avg_temp_quarter = avg_daily_temp )
>
> temp_quarter <- do.call("cbind", list(min_temperature_by_quarter, max_temperature_by_quarter, avg_temperature_by_quarter))
New names:
* Quarter -> Quarter...1
* Quarter -> Quarter...3
* Quarter -> Quarter...5
> colnames(temp_quarter)[1] <- "Quarter"
```

```
> colnames(temp_quarter)[1] <- "Quarter"
> temp_quarter <-  temp_quarter[c(1, 2, 4, 6)]
>
> # Wind_Speed (Min > Max > AVG)
> # Due to Wind Speed has 2 variable, one in 9am and the other 3pm so I calculate the average speed daily to find min max.
> avg_wind_speed_daily <- month_quarter_2019 %>%
+   group_by(Month) %>%
+   mutate(avg_daily_windspeed = (`9am_wind_speed_(km/h)` + `3pm_wind_speed_(km/h)`) / 2) %>%
+   select(. , Quarter, Month, `9am_wind_speed_(km/h)`,`3pm_wind_speed_(km/h)`, avg_daily_windspeed)
>
> ## Wind Speed By Month
> min_wind_speed_by_month <- avg_wind_speed_daily %>% # Get Min wind speed from daily average per month
+   group_by(Month) %>%
+   slice(which.min(avg_daily_windspeed)) %>%
+   select( Month, min_ws_month = avg_daily_windspeed)
>
> max_wind_speed_by_month <- avg_wind_speed_daily %>% # Get Max wind speed from daily average per month
+   group_by(Month) %>%
+   slice(which.max(avg_daily_windspeed)) %>%
+   select( Month, max_ws_month = avg_daily_windspeed)
>
> avg_wind_speed_by_month <- avg_wind_speed_daily %>% # Get average wind speed per month from daily average wind_speed
+   group_by(Month) %>%
+   aggregate( avg_daily_windspeed ~ Month, . , mean ) %>%
+   select( Month, avg_ws_month = avg_daily_windspeed )
>
> # Merge into 1 table
> ws_month <- do.call("cbind", list(min_wind_speed_by_month, max_wind_speed_by_month, avg_wind_speed_by_month))
New names:
* Month -> Month...1
* Month -> Month...3
* Month -> Month...5
> colnames(ws_month)[1] <- "Month"
> ws_month <-  ws_month[c(1, 2, 4, 6)]
>
> ## Wind Speed By Quarter
> min_wind_speed_by_quarter <- avg_wind_speed_daily %>%
+   group_by(Quarter) %>%
+   slice(which.min(avg_daily_windspeed)) %>%
+   select(Quarter, min_ws_quarter = avg_daily_windspeed)
>
> max_wind_speed_by_quarter <- avg_wind_speed_daily %>%
+   group_by(Quarter) %>%
+   slice(which.max(avg_daily_windspeed)) %>%
+   select( Quarter, max_ws_quarter = avg_daily_windspeed)
>
> avg_ws_by_quarter <- avg_wind_speed_daily %>%
+   group_by(Quarter) %>%
+   aggregate( avg_daily_windspeed ~ Quarter, . , mean ) %>%
+   select( Quarter, average_ws_quarter = avg_daily_windspeed )
>
> ws_quarter <- do.call("cbind", list(min_wind_speed_by_quarter, max_wind_speed_by_quarter, avg_ws_by_quarter))
New names:
* Quarter -> Quarter...1
* Quarter -> Quarter...3
* Quarter -> Quarter...5
> colnames(ws_quarter)[1] <- "Quarter"
> ws_quarter <-  ws_quarter[c(1, 2, 4, 6)]
>
> # Humidity (Min > Max > AVG)
> avg_humidity_daily <- month_quarter_2019 %>%
+   group_by(Month) %>%
+   mutate(avg_daily_humidity = (`9am_relative_humidity_(%)` + `3pm_relative_humidity_(%)`) / 2) %>%
+   select(. , Quarter, Month, `9am_relative_humidity_(%)`,`3pm_relative_humidity_(%)`, avg_daily_humidity)
>
> ## Humidity By Month
```

```
> min_humidity_by_month <- avg_humidity_daily %>% # Get Min Humidity from daily average per month
+   group_by(Month) %>%
+   slice(which.min(avg_daily_humidity)) %>%
+   select( Month, min_humid_month = avg_daily_humidity)
>
> max_humidity_by_month <- avg_humidity_daily %>% # Get Max Humidity from daily average per month
+   group_by(Month) %>%
+   slice(which.max(avg_daily_humidity)) %>%
+   select( Month, max_humid_month = avg_daily_humidity)
>
> avg_humidity_by_month <- avg_humidity_daily %>% # Get average Humidity per month from daily average avg_humidity_daily
+   group_by(Month) %>%
+   aggregate( avg_daily_humidity ~ Month, . , mean ) %>%
+   select( Month, average_humid_month = avg_daily_humidity )
>
> # Merge into 1 table
> humid_month <- do.call("cbind", list(min_humidity_by_month, max_humidity_by_month, avg_humidity_by_month))
New names:
* Month -> Month...1
* Month -> Month...3
* Month -> Month...5
> colnames(humid_month)[1] <- "Month"
> humid_month <-  humid_month[c(1, 2, 4, 6)]
>
>
>
> ## Humidity By Quarter
> min_humidity_by_quarter <- avg_humidity_daily %>% # Get Min Humidity from daily average per Quarter
+   group_by(Quarter) %>%
+   slice(which.min(avg_daily_humidity)) %>%
+   select( Quarter, min_humid_quarter = avg_daily_humidity)
>
> max_humidity_by_quarter <- avg_humidity_daily %>% # Get Max Humidity from daily average per Quarter
+   group_by(Quarter) %>%
+   slice(which.max(avg_daily_humidity)) %>%
+   select( Quarter, max_humid_quarter = avg_daily_humidity)
>
> avg_humidity_by_quarter <- avg_humidity_daily %>% # Get average Humidity per Quarter from daily avg_humidity_daily
+   group_by(Quarter) %>%
+   aggregate( avg_daily_humidity ~ Quarter, . , mean ) %>%
+   select( Quarter, average_humid_quarter = avg_daily_humidity )
>
> # Merge into 1 table
> humid_quarter <- do.call("cbind", list(min_humidity_by_quarter, max_humidity_by_quarter, avg_humidity_by_quarter))
New names:
* Quarter -> Quarter...1
* Quarter -> Quarter...3
* Quarter -> Quarter...5
> colnames(humid_quarter)[1] <- "Quarter"
> humid_quarter <-  humid_quarter[c(1, 2, 4, 6)]
>
```

```
> # Print the min, max, avg of temp, wind speed, humid per month, quarter
> temp_month
# A tibble: 12 x 4
   Month min_temp_month max_temp_month avg_temp_month
   <ord>          <dbl>          <dbl>          <dbl>
 1 1               13.4           41.6           26.1
 2 2                5.5           36             21.0
 3 3                3.1           34.1           19.0
 4 4                0.2           26.1           15.0
 5 5               -2.7           23.8            9.96
 6 6               -4.9           17.4            6.99
 7 7               -5.1           17.3            7.17
 8 8               -5.2           18.9            7.15
 9 9               -3.1           24.7           10.4
10 10              -0.2           31.7           14.9
11 11               1.6           39             18.2
12 12               4.6           41.1           22.4
> temp_quarter
# A tibble: 4 x 4
  Quarter min_temp_quarter max_temp_quarter avg_temp_quarter
  <ord>              <dbl>            <dbl>            <dbl>
1 1                    3.1             41.6             22.1
2 2                   -4.9             26.1             10.7
3 3                   -5.2             24.7              8.23
4 4                   -0.2             41.1             18.5
> ws_month
# A tibble: 12 x 4
   Month min_ws_month max_ws_month avg_ws_month
   <ord>        <dbl>        <dbl>        <dbl>
 1 1              8.5         32.5         14.8
 2 2              5.5         33           15.6
 3 3              3           37           15.4
 4 4              3           28           11.5
 5 5              3.5         38.5         15.1
 6 6              1           38           11.9
 7 7              3.5         39.5         17.5
 8 8              4.5         35           17.0
 9 9              6.5         35           18.0
10 10             7.5         41.5         18.4
11 11             4.5         38           21.6
12 12             8           36           18.0
> ws_quarter
# A tibble: 4 x 4
  Quarter min_ws_quarter max_ws_quarter average_ws_quarter
  <ord>            <dbl>          <dbl>               <dbl>
1 1                    3             37                15.3
2 2                    1             38.5              12.9
3 3                    3.5           39.5              17.5
4 4                    4.5           41.5              19.3
> humid_month
# A tibble: 12 x 4
   Month min_humid_month max_humid_month average_humid_month
   <ord>           <dbl>          <dbl>                <dbl>
 1 1                23.5           69                   47.9
 2 2                26             65                   47.5
 3 3                31.5           78.5                 56.5
 4 4                40.5           76                   58.5
 5 5                53             89                   68.6
 6 6                56.5           92                   72.5
 7 7                49.5           88.5                 67.7
 8 8                34.5           79.5                 61.1
 9 9                27             76                   48.2
10 10               22             68.5                 42.2
11 11               16             75                   35.2
12 12               11.5           67                   33.3
> humid_quarter
```

```
> humid_quarter
# A tibble: 4 x 4
  Quarter min_humid_quarter max_humid_quarter average_humid_quarter
  <ord>               <dbl>             <dbl>                 <dbl>
1 1                    23.5              78.5                  50.8
2 2                    40.5              92                    66.5
3 3                    27                88.5                  59.1
4 4                    11.5              75                    36.9
> 
```