Assignment 1 - Solution

Data Description

The following code generates the required dataset for this assignment. It will create a data frame with 92 rows and 13 columns. The variables of this data frame are as following:

- `Date`: Date of the reading (in YYYY-MM-DD format)
- `Temperature`: Temperature reading in Celsius
- `Humidity`: Humidity reading as a percentage
- `Pressure`: Atmospheric pressure in millibars
- `WindSpeed`: Wind speed in kilometers per hour
- `WindDirection`: Wind direction
- `DewPoint`: Dew point temperature in Celsius
- `CloudCover`: Cloud cover as a percentage
- `Precipitation`: Precipitation amount in millimeters
- `Visibility`: Visibility distance in kilometers
- `UVIndex`: UV index reading
- `condition`: The global weather condition over the day
- `Location`: The city of the recorded data

```r
# Set the seed for reproducibility
set.seed(123)

# Create a sequence of dates for the month of March 2023
dates <- seq(from = as.Date("2023-03-01"), to = as.Date("2023-05-31"), by = 1)

# Create a data frame to store the weather data
weather <- data.frame(
  Date = dates,
  Temperature = round(runif(length(dates), 15, 25), 1),
  Humidity = sample(50:100, length(dates), replace = TRUE),
  Pressure = sample(995:1015, length(dates), replace = TRUE),
  WindSpeed = sample(5:20, length(dates), replace = TRUE),
  WindDirection = sample(c("N", "NE", "E", "SE", "S", "SW", "W", "NW"), length(dates), replace = TRUE),
  DewPoint = round(runif(length(dates), 10, 15), 1),
  CloudCover = sample(0:100, length(dates), replace = TRUE),
  Precipitation = round(runif(length(dates), 0, 15), 1),
  Visibility = sample(5:20, length(dates), replace = TRUE),
  UVIndex = sample(1:5, length(dates), replace = TRUE),
  Condition = sample(c("Sunny", "Partly Cloudy", "Rainy", "Snowy"), length(dates), replace = TRUE),
  Location = rep(c("Sydney", "Canberra"), each = length(dates)/2)
)
# do the required here to check the generated data

# check contenets of weather dataframe
print(weather)
```

```
##          Date Temperature Humidity Pressure WindSpeed WindDirection DewPoint
## 1  2023-03-01        17.9       98     1014        17            NW     13.7
## 2  2023-03-02        22.9       83     1010        16             N     11.7
## 3  2023-03-03        19.1       53     1005        19            SW     14.1
## 4  2023-03-04        23.8       62     1010         9            NE     12.7
## 5  2023-03-05        24.4       54     1014        12            NW     11.4
## 6  2023-03-06        15.5      100     1002         9            NE     14.0
## 7  2023-03-07        20.3       74      997        19            NW     10.5
## 8  2023-03-08        23.9       71      998        11            NE     14.2
## 9  2023-03-09        20.5       74     1014         8             S     11.4
## 10 2023-03-10        19.6       81     1006         6            SE     13.8
## 11 2023-03-11        24.6       95     1011        17            SW     14.8
## 12 2023-03-12        19.5       74     1004         5            SE     10.4
## 13 2023-03-13        21.8       72     1014         5             S     14.3
## 14 2023-03-14        20.7       84     1005         6            SE     14.0
## 15 2023-03-15        16.0       89     1002         5            SW     11.9
## 16 2023-03-16        24.0       97     1008         6             N     11.6
## 17 2023-03-17        17.5       79     1015        17            NW     11.0
## 18 2023-03-18        15.4       61     1007        12            NW     12.8
## 19 2023-03-19        18.3       80      996         5             W     14.4
## 20 2023-03-20        24.5       95     1005        19            NE     12.6
## 21 2023-03-21        23.9       79     1007         7             N     12.9
## 22 2023-03-22        21.9       84     1008         6             W     13.3
## 23 2023-03-23        21.4       63     1000         6            NE     12.6
## 24 2023-03-24        24.9       78     1002         9            NW     12.5
## 25 2023-03-25        21.6       81     1006        19             E     10.1
## 26 2023-03-26        22.1       56      998        16            NE     10.2
## 27 2023-03-27        20.4       52     1007        13             W     14.6
## 28 2023-03-28        20.9       72     1008        17             E     13.8
## 29 2023-03-29        17.9       64     1015        14             N     11.0
## 30 2023-03-30        16.5       70     1010        17            SE     13.3
## 31 2023-03-31        24.6       86      995        10            NE     13.3
## 32 2023-04-01        24.0       57     1002        14             E     12.0
## 33 2023-04-02        21.9      100     1002        14             S     14.1
## 34 2023-04-03        23.0       59     1004        10             S     12.7
## 35 2023-04-04        15.2       99     1002        16             W     14.4
## 36 2023-04-05        19.8       91     1012        20            SW     12.8
## 37 2023-04-06        22.6       93     1015        20             E     14.5
## 38 2023-04-07        17.2       83     1003        20            SW     12.9
## 39 2023-04-08        18.2       59     1001         7            SW     12.1
## 40 2023-04-09        17.3       71     1001         8            NE     14.7
## 41 2023-04-10        16.4       61     1004        15            NW     13.5
## 42 2023-04-11        19.1       69     1005        15             S     12.1
## 43 2023-04-12        19.1       95      995         7            SE     10.1
## 44 2023-04-13        18.7       66     1013        17            NE     12.8
## 45 2023-04-14        16.5       95     1004        11             N     12.5
## 46 2023-04-15        16.4       84     1015         7            NW     14.4
## 47 2023-04-16        17.3       89     1007        19            SE     14.1
## 48 2023-04-17        19.7       95     1005        20             N     14.3
## 49 2023-04-18        17.7      100     1005         6            SE     11.8
## 50 2023-04-19        23.6       79     1014        19             S     14.4
## 51 2023-04-20        15.5       64     1001        11            SE     10.8
## 52 2023-04-21        19.4       73     1014         7            NE     11.4
## 53 2023-04-22        23.0       98     1003        13            NE     13.3
```

```
## 54 2023-04-23        16.2       72     1003        17            SE     14.9
## 55 2023-04-24        20.6       92      999        11             W     12.9
## 56 2023-04-25        17.1       56     1008        18             E     12.6
## 57 2023-04-26        16.3       78     1008        19             S     10.3
## 58 2023-04-27        22.5       64     1000        18            SW     14.8
## 59 2023-04-28        24.0       72      995        13            NW     10.6
## 60 2023-04-29        18.7       75     1004         8            NW     10.4
## 61 2023-04-30        21.7       87     1011         6            NE     14.4
## 62 2023-05-01        15.9       95     1011        10             W     12.5
## 63 2023-05-02        18.8       81     1015        14             E     11.7
## 64 2023-05-03        17.7       56     1001        16             E     14.5
## 65 2023-05-04        23.1       76     1015        13             E     10.2
## 66 2023-05-05        19.5       91     1003        16            SW     11.2
## 67 2023-05-06        23.1       54     1014        17             N     13.4
## 68 2023-05-07        23.1       55     1000        17            NW     11.1
## 69 2023-05-08        22.9       65     1012        15             N     11.6
## 70 2023-05-09        19.4       73     1011        11            NW     10.9
## 71 2023-05-10        22.5       81      999        17             E     14.0
## 72 2023-05-11        21.3       70     1014         9            SE     10.7
## 73 2023-05-12        22.1       60      997        15            NW     14.1
## 74 2023-05-13        15.0       85     1008         5            SE     11.7
## 75 2023-05-14        19.8       93      995         7             S     11.9
## 76 2023-05-15        17.2       95      996        16            NE     13.1
## 77 2023-05-16        18.8       68      998         6            NE     10.5
## 78 2023-05-17        21.1       74     1004         9            NW     10.1
## 79 2023-05-18        18.5       88      995         5            SE     15.0
## 80 2023-05-19        16.1       75      999        19             W     12.9
## 81 2023-05-20        17.4       58     1015        16            NE     13.9
## 82 2023-05-21        21.7       56     1002        18             N     14.5
## 83 2023-05-22        19.2       83     1015         7            SE     13.8
## 84 2023-05-23        22.9       97     1007        18             E     14.9
## 85 2023-05-24        16.0       62     1012        13             N     10.2
## 86 2023-05-25        19.3       68     1004         6             S     14.5
## 87 2023-05-26        24.8       96     1000        19             E     14.3
## 88 2023-05-27        23.9       88     1001        10            SE     13.9
## 89 2023-05-28        23.9       53     1003         7            NW     11.9
## 90 2023-05-29        16.8       50     1010         7            NE     10.2
## 91 2023-05-30        16.3       89     1011        13             S     11.8
## 92 2023-05-31        21.5       79     1015         6             S     11.4
##    CloudCover Precipitation Visibility UVIndex     Condition Location
## 1          56          14.9         10       5         Sunny   Sydney
## 2          70           2.2         13       3 Partly Cloudy   Sydney
## 3          86           0.7          5       3         Rainy   Sydney
## 4          71           9.0          8       2         Snowy   Sydney
## 5          58           4.8         18       2         Sunny   Sydney
## 6          73           7.9         20       4         Sunny   Sydney
## 7          36          11.9          7       2 Partly Cloudy   Sydney
## 8          80           1.0          9       1         Snowy   Sydney
## 9          54          10.4          6       4         Rainy   Sydney
## 10          5          14.2          5       1         Sunny   Sydney
## 11         39           4.2         16       5 Partly Cloudy   Sydney
## 12          5           3.7          8       2         Snowy   Sydney
## 13         27           2.4         18       5         Sunny   Sydney
## 14         31           1.1          8       2         Sunny   Sydney
```

```
## 15         48          8.4         15       2         Rainy   Sydney
## 16         98          7.3         14       3         Snowy   Sydney
## 17         91          7.4         14       2         Rainy   Sydney
## 18         83         13.3         15       3         Rainy   Sydney
## 19         64          0.6          9       3         Rainy   Sydney
## 20         68          3.1         10       1         Rainy   Sydney
## 21         74          7.7          8       1 Partly Cloudy   Sydney
## 22         46          3.5          7       5         Rainy   Sydney
## 23         19          8.6          7       1 Partly Cloudy   Sydney
## 24          1          7.2          5       5         Rainy   Sydney
## 25         61          8.5         11       1 Partly Cloudy   Sydney
## 26         62          2.2          9       2         Rainy   Sydney
## 27         61          2.2         17       1 Partly Cloudy   Sydney
## 28          9          6.2         11       3 Partly Cloudy   Sydney
## 29         58         10.2         13       3         Rainy   Sydney
## 30         48          9.8         13       4         Snowy   Sydney
## 31          6         13.7         19       2 Partly Cloudy   Sydney
## 32         60         12.2         12       4         Sunny   Sydney
## 33         37         10.1          5       3         Rainy   Sydney
## 34         35         12.1          9       1         Sunny   Sydney
## 35         22          1.9          9       3         Rainy   Sydney
## 36         33          3.8         20       4 Partly Cloudy   Sydney
## 37         87          5.0         12       2         Snowy   Sydney
## 38          3          6.1         16       5         Sunny   Sydney
## 39         53          9.5          6       2         Sunny   Sydney
## 40         28         12.1         10       1         Rainy   Sydney
## 41         57          3.9          6       2         Sunny   Sydney
## 42         86         12.3          8       3 Partly Cloudy   Sydney
## 43         74          0.2         14       3         Rainy   Sydney
## 44         96          9.8         19       2         Snowy   Sydney
## 45         55         12.2         11       2         Snowy   Sydney
## 46         50          6.9         10       4         Snowy   Sydney
## 47         49          3.1         12       5         Rainy Canberra
## 48         47          0.3          5       2         Snowy Canberra
## 49          2          4.2         12       3 Partly Cloudy Canberra
## 50         32         13.7         15       1 Partly Cloudy Canberra
## 51         93         14.0          9       5 Partly Cloudy Canberra
## 52         33          8.1         15       3         Rainy Canberra
## 53         65          3.3         19       2         Snowy Canberra
## 54         63          7.3         19       1         Rainy Canberra
## 55         39         12.0         13       4         Rainy Canberra
## 56         57          6.1         12       4 Partly Cloudy Canberra
## 57         19          1.6         14       3         Snowy Canberra
## 58         77          4.2         19       1         Sunny Canberra
## 59         24          5.4         19       2         Rainy Canberra
## 60         70          3.9          8       3         Rainy Canberra
## 61         39          7.1         16       1         Rainy Canberra
## 62         60          5.5         13       1         Rainy Canberra
## 63         98          1.8         17       2         Rainy Canberra
## 64         94          0.7         13       2         Rainy Canberra
## 65         21          3.9          8       4         Snowy Canberra
## 66         79         14.5          7       3         Snowy Canberra
## 67         85          7.3         11       2         Sunny Canberra
## 68         92          7.2          8       2         Sunny Canberra
```

```
## 69         55          11.2         12       5         Snowy Canberra
## 70         32          10.0         12       1         Sunny Canberra
## 71         63           0.7          9       3         Sunny Canberra
## 72         52          10.4          9       2         Sunny Canberra
## 73         30           5.4         12       2 Partly Cloudy Canberra
## 74         45          13.3         10       4         Sunny Canberra
## 75         71          11.6          8       4 Partly Cloudy Canberra
## 76         23           2.1          7       4 Partly Cloudy Canberra
## 77         59           4.4         15       5         Snowy Canberra
## 78         29           1.9         14       5         Snowy Canberra
## 79         22           8.8         20       4         Snowy Canberra
## 80        100           8.4         17       4         Snowy Canberra
## 81         95          10.3         11       2         Snowy Canberra
## 82         16           4.7          9       5         Rainy Canberra
## 83         89           9.1         12       2         Snowy Canberra
## 84         58          14.9         20       2         Rainy Canberra
## 85         56          11.1         13       4         Sunny Canberra
## 86         42           1.1         15       5 Partly Cloudy Canberra
## 87         22           6.8         14       3 Partly Cloudy Canberra
## 88         89           0.8         19       3         Sunny Canberra
## 89         99           5.1          7       2         Snowy Canberra
## 90         44          11.0         16       2         Rainy Canberra
## 91          7           0.1         19       1         Rainy Canberra
## 92         78          11.6         14       3         Snowy Canberra
```

```
# to check dimensions of dataframe
dim(weather)
```

```
## [1] 92 13
```

```
# check structure of data frame
str(weather)
```

```
## 'data.frame':    92 obs. of  13 variables:
##  $ Date         : Date, format: "2023-03-01" "2023-03-02" ...
##  $ Temperature  : num  17.9 22.9 19.1 23.8 24.4 15.5 20.3 23.9 20.5 19.6 ...
##  $ Humidity     : int  98 83 53 62 54 100 74 71 74 81 ...
##  $ Pressure     : int  1014 1010 1005 1010 1014 1002 997 998 1014 1006 ...
##  $ WindSpeed    : int  17 16 19 9 12 9 19 11 8 6 ...
##  $ WindDirection: chr  "NW" "N" "SW" "NE" ...
##  $ DewPoint     : num  13.7 11.7 14.1 12.7 11.4 14 10.5 14.2 11.4 13.8 ...
##  $ CloudCover   : int  56 70 86 71 58 73 36 80 54 5 ...
##  $ Precipitation: num  14.9 2.2 0.7 9 4.8 7.9 11.9 1 10.4 14.2 ...
##  $ Visibility   : int  10 13 5 8 18 20 7 9 6 5 ...
##  $ UVIndex      : int  5 3 3 2 2 4 2 1 4 1 ...
##  $ Condition    : chr  "Sunny" "Partly Cloudy" "Rainy" "Snowy" ...
##  $ Location     : chr  "Sydney" "Sydney" "Sydney" "Sydney" ...
```

Submission for Part A: Data Understanding

Please follow this structure:

1- The data set description goes here

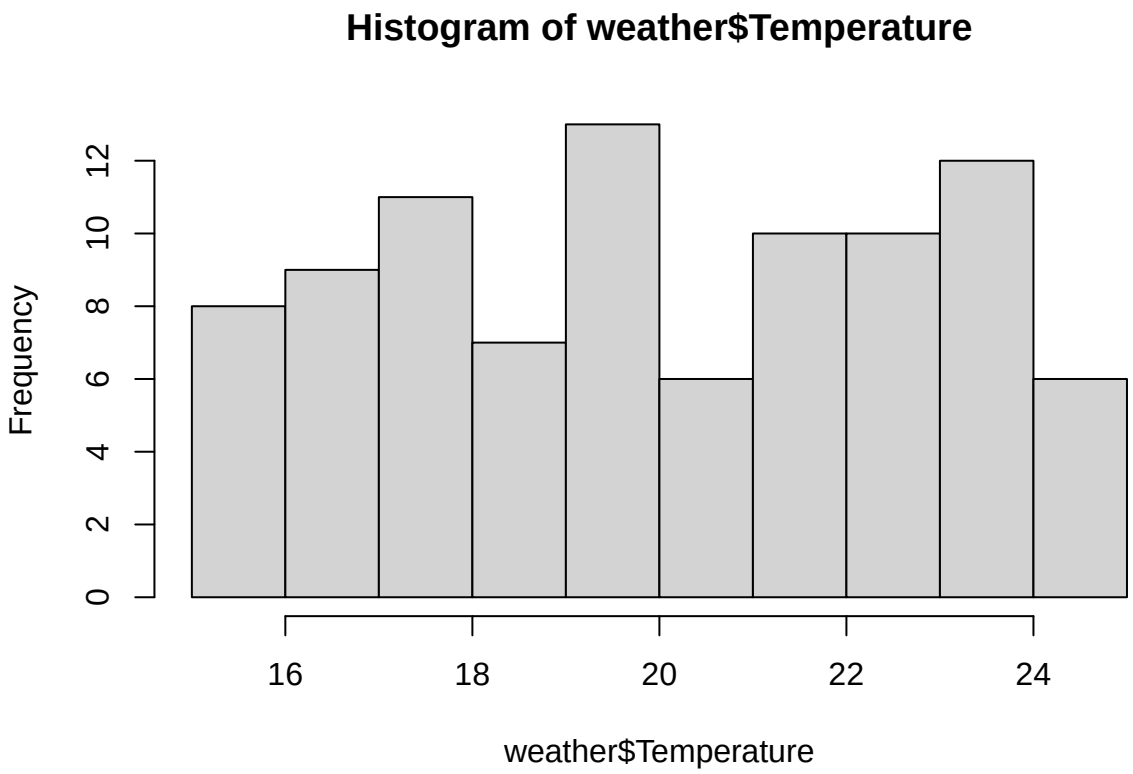
This dataset contains weather-related information for the cities of Sydney and Canberra for the months of March, April and May in 2023. It comprises 92 observations and includes 13 variables. The variables 'WindDirection', 'Condition' and 'Location' are categorical and the rest are quantitative.

This dataset can be used for various types of weather analysis specific to the cities of Sydney and Canberra during the month of March. Researchers and analysts can examine temperature variations, humidity levels, precipitation patterns, and wind characteristics to understand the unique weather conditions in these two cities during that time frame. It can also be valuable for assessing how different weather conditions may have impacted daily life, outdoor activities, or local businesses in Sydney and Canberra during the time period. Furthermore, the data can be used to build predictive models for short-term weather forecasting in these specific locations.

```r
#---------------------------------#

# 2-  The code for task 2 goes here

summary(weather)
```

```
##       Date             Temperature       Humidity         Pressure
##  Min.   :2023-03-01   Min.   :15.00   Min.   : 50.00   Min.   : 995
##  1st Qu.:2023-03-23   1st Qu.:17.48   1st Qu.: 64.00   1st Qu.:1001
##  Median :2023-04-15   Median :19.75   Median : 77.00   Median :1005
##  Mean   :2023-04-15   Mean   :20.05   Mean   : 76.64   Mean   :1006
##  3rd Qu.:2023-05-08   3rd Qu.:22.68   3rd Qu.: 89.00   3rd Qu.:1011
##  Max.   :2023-05-31   Max.   :24.90   Max.   :100.00   Max.   :1015
##    WindSpeed     WindDirection         DewPoint       CloudCover
##  Min.   : 5.00   Length:92          Min.   :10.10   Min.   :  1.00
##  1st Qu.: 7.00   Class :character   1st Qu.:11.40   1st Qu.: 32.00
##  Median :13.00   Mode  :character   Median :12.80   Median : 55.50
##  Mean   :12.39                      Mean   :12.67   Mean   : 53.02
##  3rd Qu.:17.00                      3rd Qu.:14.10   3rd Qu.: 73.25
##  Max.   :20.00                      Max.   :15.00   Max.   :100.00
##  Precipitation      Visibility       UVIndex       Condition
##  Min.   : 0.100   Min.   : 5.00   Min.   :1.000   Length:92
##  1st Qu.: 3.450   1st Qu.: 8.75   1st Qu.:2.000   Class :character
##  Median : 7.150   Median :12.00   Median :3.000   Mode  :character
##  Mean   : 6.917   Mean   :12.10   Mean   :2.793
##  3rd Qu.:10.325   3rd Qu.:15.00   3rd Qu.:4.000
##  Max.   :14.900   Max.   :20.00   Max.   :5.000
##    Location
##  Length:92
##  Class :character
##  Mode  :character
##
##
##
```

```r
#---------------------------------#

# 3-  The code for task 3 goes here

hist(weather$Temperature)
```




```
#---------------------------------#
```

4- The reflection and notes for task 2 and 3 goes here

For task 2, the 'summary()' function is really useful in getting an overall overview of the nature of various variables within the dataset we are dealing with. For instance, if we are looking at the Temperature variable, we can observe that the maximum temperature observed is 24.90 degree celcius and the min tempurature recorded is 15 degree celcius, with the mean at 20.05 degree celcius. Also, it helps in understanding the categorical variables in the dataset. we have three categorical variables in weather dataset, 'WindDirection', 'Condition' and 'Location'.

For task 3, the histogram of the temperature reading suggests that the distribution of the data is bimodal with one mode between 19 degree celcius and 20 degree celcius and another mode between 23 degree celcius and 24 degree celcius.

Submission for part B: Vector and Matrix Manipulation

Please follow this structure:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# to check the data type of the data column
class(weather$Date)
```

```
## [1] "Date"
```

```
# Extract the month and year from the "Date" column
weather$YearMonth <- format(weather$Date, "%Y-%m")

# extract year, month and date
weather <- weather %>%
  separate(col=Date, into=c("Year","Month","Day"), sep="\\-", remove=FALSE)

#---------------------------------#
# 1-  The code for task 1 goes here
# Create a vector containing the average temperature readings for each month.

# using group_by and summarize to find the monthly average temperature
monthly_average_temperature <-
  (
  weather %>%
    group_by(Month) %>%
    summarize(average_temp = mean(Temperature))
  )$average_temp

cat("Q1: Monthly average temperatures: ")
```

```
## Q1: Monthly average temperatures:
```

```
print(monthly_average_temperature)
```

```
## [1] 20.84839 19.29000 19.98710
```

```
#---------------------------------#

# 2-  The code for task 2 goes here
# Create two vector containing the average humidity readings for each month; one for Sydney and another

canberra_monthly_avg_humidity <-
  (
    weather %>%
      filter(Location=='Canberra') %>%
      group_by(Month) %>%
      summarise(average_humidity=mean(Humidity))
  )$average_humidity

sydney_monthly_average_humidity <-
  (
    weather %>%
      filter(Location=='Sydney') %>%
      group_by(Month) %>%
      summarise(average_humidity=mean(Humidity))
  )$average_humidity
```

```r
cat("Q2: Monthly average humidity for Canberra: \n")
```

```
## Q2: Monthly average humidity for Canberra:
```

```r
print(canberra_monthly_avg_humidity)
```

```
## [1] 79.60000 74.64516
```

```r
cat("Q2: Monthly average humidity for Sydney: \n")
```

```
## Q2: Monthly average humidity for Sydney:
```

```r
print(sydney_monthly_average_humidity)
```

```
## [1] 76.16129 78.80000
```

```r
#----------------------------------#

# 3-  The code for task 3 goes here

# Create a matrix containing the average temperature, humidity, pressure, and wind speed readings for e


# first let's calculate the average values of these parameters using tidyverse functions group_by and s
monthly_averages_df <- weather %>%
  group_by(Month) %>%
  summarize(
    average_temp = mean(Temperature),
    average_humidity = mean(Humidity),
    average_pressure = mean(Pressure),
    average_wind_speed = mean(WindSpeed)
    )

monthly_averages_df <- monthly_averages_df %>% select(-Month)

# as the next step, we will convert the df into a matrix
# reference: https://stat.ethz.ch/R-manual/R-devel/library/base/html/data.matrix.html
monthly_averages <- data.matrix(monthly_averages_df)

# is.matrix(monthly_averages)
print("Matrix containing the average temperature, humidity, pressure, and wind speed readings for each m
```

```
## [1] "Matrix containing the average temperature, humidity, pressure, and wind speed readings for each
```

```r
monthly_averages
```

```
##      average_temp average_humidity average_pressure average_wind_speed
## [1,]     20.84839         76.16129         1006.226           11.51613
## [2,]     19.29000         79.20000         1005.167           13.53333
## [3,]     19.98710         74.64516         1005.871           12.16129
```

```r
#----------------------------------#

# 4-  The code for task 4 goes here
# Create another matrix containing the average temperature, humidity, pressure, and wind speed readings


city_averages_df <- weather %>%
```

```
  group_by(Location) %>%
  summarize(
    average_temp = mean(Temperature),
    average_humidity = mean(Humidity),
    average_pressure = mean(Pressure),
    average_wind_speed = mean(WindSpeed)
    )
city_averages_df <- city_averages_df %>% select(-Location)

# as the next step, we will convert the df into a matrix
# reference: https://stat.ethz.ch/R-manual/R-devel/library/base/html/data.matrix.html
city_averages <- data.matrix(city_averages_df)

# is.matrix(city_averages)
print("Matrix containing the average temperature, humidity, pressure, and wind speed readings for each
```

```
## [1] "Matrix containing the average temperature, humidity, pressure, and wind speed readings for each
```

```
city_averages
```

```
##      average_temp average_humidity average_pressure average_wind_speed
## [1,]     19.84565         76.26087         1005.630           12.65217
## [2,]     20.25435         77.02174         1005.891           12.13043
```

```
#----------------------------------#

# 5-  The code for task 5 goes here
# Create an array containing the average temperature, humidity, pressure, wind speed, and UV index read

months_vec <- unique(weather$Month)

# Determine the dimensions of your 3D array
num_rows <- 31
num_cols <- 5
num_months <- length(months_vec)

# Initialize the 3D array with NA values
avg_readings_array <- array(NA, c(num_rows, num_cols, num_months))

# runs a loop to store the data for each month
for (i in 1:length(months_vec)) {

  # filters and selects the required data and stores in a temporary dataframe
  temp_df <- weather %>%
    filter( Month == months_vec[i]) %>%
    select(Temperature,Humidity,Pressure, WindSpeed, UVIndex)

  # converts the temp data frame to a matrix
  avg_read_mat <- as.matrix(temp_df)

  # checks if the rows of the matrix is as per the expected dimensions
  if(nrow(avg_read_mat) < 31) {

    # if the dimensions donot match add more rows with zero values to adjust the dimensions
    avg_read_mat <- rbind(avg_read_mat, numeric(5))
```

```
  }

  # append the matrix to the array
  avg_readings_array[ , ,i] <- avg_read_mat
}


print("PART B: Q5: Array containing the average temperature, humidity, pressure, wind speed, and UV inde
```

```
## [1] "PART B: Q5: Array containing the average temperature, humidity, pressure, wind speed, and UV ind
```

```
avg_readings_array
```

```
## , , 1
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 17.9   98 1014   17    5
##  [2,] 22.9   83 1010   16    3
##  [3,] 19.1   53 1005   19    3
##  [4,] 23.8   62 1010    9    2
##  [5,] 24.4   54 1014   12    2
##  [6,] 15.5  100 1002    9    4
##  [7,] 20.3   74  997   19    2
##  [8,] 23.9   71  998   11    1
##  [9,] 20.5   74 1014    8    4
## [10,] 19.6   81 1006    6    1
## [11,] 24.6   95 1011   17    5
## [12,] 19.5   74 1004    5    2
## [13,] 21.8   72 1014    5    5
## [14,] 20.7   84 1005    6    2
## [15,] 16.0   89 1002    5    2
## [16,] 24.0   97 1008    6    3
## [17,] 17.5   79 1015   17    2
## [18,] 15.4   61 1007   12    3
## [19,] 18.3   80  996    5    3
## [20,] 24.5   95 1005   19    1
## [21,] 23.9   79 1007    7    1
## [22,] 21.9   84 1008    6    5
## [23,] 21.4   63 1000    6    1
## [24,] 24.9   78 1002    9    5
## [25,] 21.6   81 1006   19    1
## [26,] 22.1   56  998   16    2
## [27,] 20.4   52 1007   13    1
## [28,] 20.9   72 1008   17    3
## [29,] 17.9   64 1015   14    3
## [30,] 16.5   70 1010   17    4
## [31,] 24.6   86  995   10    2
##
## , , 2
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 24.0   57 1002   14    4
##  [2,] 21.9  100 1002   14    3
##  [3,] 23.0   59 1004   10    1
```

```
##  [4,] 15.2   99 1002   16    3
##  [5,] 19.8   91 1012   20    4
##  [6,] 22.6   93 1015   20    2
##  [7,] 17.2   83 1003   20    5
##  [8,] 18.2   59 1001    7    2
##  [9,] 17.3   71 1001    8    1
## [10,] 16.4   61 1004   15    2
## [11,] 19.1   69 1005   15    3
## [12,] 19.1   95  995    7    3
## [13,] 18.7   66 1013   17    2
## [14,] 16.5   95 1004   11    2
## [15,] 16.4   84 1015    7    4
## [16,] 17.3   89 1007   19    5
## [17,] 19.7   95 1005   20    2
## [18,] 17.7  100 1005    6    3
## [19,] 23.6   79 1014   19    1
## [20,] 15.5   64 1001   11    5
## [21,] 19.4   73 1014    7    3
## [22,] 23.0   98 1003   13    2
## [23,] 16.2   72 1003   17    1
## [24,] 20.6   92  999   11    4
## [25,] 17.1   56 1008   18    4
## [26,] 16.3   78 1008   19    3
## [27,] 22.5   64 1000   18    1
## [28,] 24.0   72  995   13    2
## [29,] 18.7   75 1004    8    3
## [30,] 21.7   87 1011    6    1
## [31,]  0.0    0    0    0    0
##
## , , 3
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 15.9   95 1011   10    1
##  [2,] 18.8   81 1015   14    2
##  [3,] 17.7   56 1001   16    2
##  [4,] 23.1   76 1015   13    4
##  [5,] 19.5   91 1003   16    3
##  [6,] 23.1   54 1014   17    2
##  [7,] 23.1   55 1000   17    2
##  [8,] 22.9   65 1012   15    5
##  [9,] 19.4   73 1011   11    1
## [10,] 22.5   81  999   17    3
## [11,] 21.3   70 1014    9    2
## [12,] 22.1   60  997   15    2
## [13,] 15.0   85 1008    5    4
## [14,] 19.8   93  995    7    4
## [15,] 17.2   95  996   16    4
## [16,] 18.8   68  998    6    5
## [17,] 21.1   74 1004    9    5
## [18,] 18.5   88  995    5    4
## [19,] 16.1   75  999   19    4
## [20,] 17.4   58 1015   16    2
## [21,] 21.7   56 1002   18    5
## [22,] 19.2   83 1015    7    2
```

```
## [23,] 22.9   97 1007   18    2
## [24,] 16.0   62 1012   13    4
## [25,] 19.3   68 1004    6    5
## [26,] 24.8   96 1000   19    3
## [27,] 23.9   88 1001   10    3
## [28,] 23.9   53 1003    7    2
## [29,] 16.8   50 1010    7    2
## [30,] 16.3   89 1011   13    1
## [31,] 21.5   79 1015    6    3
```

```r
#---------------------------------#

# 6-  The code for task 6 goes here
# Create an array containing the average temperature, humidity, pressure, wind speed, and UV index
# readings for each day of each month, for each city.

# creates a vec containing the distinct cities in dataframe
cities_vec <- unique(weather$Location)

# calculate the dimensions of the 4D array
fourth_dimension <- length(cities_vec)
depth <- length(months_vec)
rows <- 31
cols <- 5

# initialize the 4D array
cities_monthly_avg_arr <- array(NA, c(rows, cols, depth, fourth_dimension))

# creates a loop for each city (iterates n times , if there are n no. of cities)
for(k in 1:fourth_dimension) {

  # extract data from weather dataframe for the city at index k in cities_vec
  city_df <- weather %>%
    filter(Location == cities_vec[k])

  # creates a temporary 3D array to hold monthly data for the city at index k
  # this 3D array will be later appended to the 4D array
  temp_arr <- array(NA, c(rows, cols, depth))

  # loop to extract data for each month
  for (i in 1:length(months_vec)) {

    # extracts data for each month from city data frame (which was already filtered to contain data of
    temp_df <- city_df %>%
      filter( Month == months_vec[i]) %>%
      select(Temperature,Humidity,Pressure, WindSpeed, UVIndex)

    # store the temp_df as a matrix
    avg_read_mat <- as.matrix(temp_df)

    # if the no. of rows doesn't conforms to the expected dimensions add more rows with zero values
    while(nrow(avg_read_mat) < 31) {
      avg_read_mat <- rbind(avg_read_mat, numeric(5))
    }
```

```
    # add the matrix to the i-th slice of the 3D array, each slice represents average readings per day
    temp_arr[ , ,i] <- avg_read_mat
  }

  # add each 3D array to the 4D array, where each depth represents average readings for each day for ea
  # as we only have two cities, the 4D array has a depth of 2
  cities_monthly_avg_arr[ , , , k] <- temp_arr
  # reference from chatGPT on how to assign values to 4D and 3D arrays
}

print("PART B: Q6: Array containing the average temperature, humidity, pressure, wind speed, and UV ind
```

```
## [1] "PART B: Q6: Array containing the average temperature, humidity, pressure, wind speed, and UV in
```

```
cities_monthly_avg_arr
```

```
## , , 1, 1
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 17.9   98 1014   17    5
##  [2,] 22.9   83 1010   16    3
##  [3,] 19.1   53 1005   19    3
##  [4,] 23.8   62 1010    9    2
##  [5,] 24.4   54 1014   12    2
##  [6,] 15.5  100 1002    9    4
##  [7,] 20.3   74  997   19    2
##  [8,] 23.9   71  998   11    1
##  [9,] 20.5   74 1014    8    4
## [10,] 19.6   81 1006    6    1
## [11,] 24.6   95 1011   17    5
## [12,] 19.5   74 1004    5    2
## [13,] 21.8   72 1014    5    5
## [14,] 20.7   84 1005    6    2
## [15,] 16.0   89 1002    5    2
## [16,] 24.0   97 1008    6    3
## [17,] 17.5   79 1015   17    2
## [18,] 15.4   61 1007   12    3
## [19,] 18.3   80  996    5    3
## [20,] 24.5   95 1005   19    1
## [21,] 23.9   79 1007    7    1
## [22,] 21.9   84 1008    6    5
## [23,] 21.4   63 1000    6    1
## [24,] 24.9   78 1002    9    5
## [25,] 21.6   81 1006   19    1
## [26,] 22.1   56  998   16    2
## [27,] 20.4   52 1007   13    1
## [28,] 20.9   72 1008   17    3
## [29,] 17.9   64 1015   14    3
## [30,] 16.5   70 1010   17    4
## [31,] 24.6   86  995   10    2
##
## , , 2, 1
##
##       [,1] [,2] [,3] [,4] [,5]
```

```
##  [1,] 24.0   57 1002   14    4
##  [2,] 21.9  100 1002   14    3
##  [3,] 23.0   59 1004   10    1
##  [4,] 15.2   99 1002   16    3
##  [5,] 19.8   91 1012   20    4
##  [6,] 22.6   93 1015   20    2
##  [7,] 17.2   83 1003   20    5
##  [8,] 18.2   59 1001    7    2
##  [9,] 17.3   71 1001    8    1
## [10,] 16.4   61 1004   15    2
## [11,] 19.1   69 1005   15    3
## [12,] 19.1   95  995    7    3
## [13,] 18.7   66 1013   17    2
## [14,] 16.5   95 1004   11    2
## [15,] 16.4   84 1015    7    4
## [16,]  0.0    0    0    0    0
## [17,]  0.0    0    0    0    0
## [18,]  0.0    0    0    0    0
## [19,]  0.0    0    0    0    0
## [20,]  0.0    0    0    0    0
## [21,]  0.0    0    0    0    0
## [22,]  0.0    0    0    0    0
## [23,]  0.0    0    0    0    0
## [24,]  0.0    0    0    0    0
## [25,]  0.0    0    0    0    0
## [26,]  0.0    0    0    0    0
## [27,]  0.0    0    0    0    0
## [28,]  0.0    0    0    0    0
## [29,]  0.0    0    0    0    0
## [30,]  0.0    0    0    0    0
## [31,]  0.0    0    0    0    0
##
## , , 3, 1
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    0    0    0    0    0
##  [2,]    0    0    0    0    0
##  [3,]    0    0    0    0    0
##  [4,]    0    0    0    0    0
##  [5,]    0    0    0    0    0
##  [6,]    0    0    0    0    0
##  [7,]    0    0    0    0    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0    0    0
## [10,]    0    0    0    0    0
## [11,]    0    0    0    0    0
## [12,]    0    0    0    0    0
## [13,]    0    0    0    0    0
## [14,]    0    0    0    0    0
## [15,]    0    0    0    0    0
## [16,]    0    0    0    0    0
## [17,]    0    0    0    0    0
## [18,]    0    0    0    0    0
## [19,]    0    0    0    0    0
```

```
## [20,]    0    0    0    0    0
## [21,]    0    0    0    0    0
## [22,]    0    0    0    0    0
## [23,]    0    0    0    0    0
## [24,]    0    0    0    0    0
## [25,]    0    0    0    0    0
## [26,]    0    0    0    0    0
## [27,]    0    0    0    0    0
## [28,]    0    0    0    0    0
## [29,]    0    0    0    0    0
## [30,]    0    0    0    0    0
## [31,]    0    0    0    0    0
##
## , , 1, 2
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    0    0    0    0    0
##  [2,]    0    0    0    0    0
##  [3,]    0    0    0    0    0
##  [4,]    0    0    0    0    0
##  [5,]    0    0    0    0    0
##  [6,]    0    0    0    0    0
##  [7,]    0    0    0    0    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0    0    0
## [10,]    0    0    0    0    0
## [11,]    0    0    0    0    0
## [12,]    0    0    0    0    0
## [13,]    0    0    0    0    0
## [14,]    0    0    0    0    0
## [15,]    0    0    0    0    0
## [16,]    0    0    0    0    0
## [17,]    0    0    0    0    0
## [18,]    0    0    0    0    0
## [19,]    0    0    0    0    0
## [20,]    0    0    0    0    0
## [21,]    0    0    0    0    0
## [22,]    0    0    0    0    0
## [23,]    0    0    0    0    0
## [24,]    0    0    0    0    0
## [25,]    0    0    0    0    0
## [26,]    0    0    0    0    0
## [27,]    0    0    0    0    0
## [28,]    0    0    0    0    0
## [29,]    0    0    0    0    0
## [30,]    0    0    0    0    0
## [31,]    0    0    0    0    0
##
## , , 2, 2
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 17.3   89 1007   19    5
##  [2,] 19.7   95 1005   20    2
##  [3,] 17.7  100 1005    6    3
```

```
##  [4,] 23.6   79 1014   19    1
##  [5,] 15.5   64 1001   11    5
##  [6,] 19.4   73 1014    7    3
##  [7,] 23.0   98 1003   13    2
##  [8,] 16.2   72 1003   17    1
##  [9,] 20.6   92  999   11    4
## [10,] 17.1   56 1008   18    4
## [11,] 16.3   78 1008   19    3
## [12,] 22.5   64 1000   18    1
## [13,] 24.0   72  995   13    2
## [14,] 18.7   75 1004    8    3
## [15,] 21.7   87 1011    6    1
## [16,]  0.0    0    0    0    0
## [17,]  0.0    0    0    0    0
## [18,]  0.0    0    0    0    0
## [19,]  0.0    0    0    0    0
## [20,]  0.0    0    0    0    0
## [21,]  0.0    0    0    0    0
## [22,]  0.0    0    0    0    0
## [23,]  0.0    0    0    0    0
## [24,]  0.0    0    0    0    0
## [25,]  0.0    0    0    0    0
## [26,]  0.0    0    0    0    0
## [27,]  0.0    0    0    0    0
## [28,]  0.0    0    0    0    0
## [29,]  0.0    0    0    0    0
## [30,]  0.0    0    0    0    0
## [31,]  0.0    0    0    0    0
##
## , , 3, 2
##
##       [,1] [,2] [,3] [,4] [,5]
##  [1,] 15.9   95 1011   10    1
##  [2,] 18.8   81 1015   14    2
##  [3,] 17.7   56 1001   16    2
##  [4,] 23.1   76 1015   13    4
##  [5,] 19.5   91 1003   16    3
##  [6,] 23.1   54 1014   17    2
##  [7,] 23.1   55 1000   17    2
##  [8,] 22.9   65 1012   15    5
##  [9,] 19.4   73 1011   11    1
## [10,] 22.5   81  999   17    3
## [11,] 21.3   70 1014    9    2
## [12,] 22.1   60  997   15    2
## [13,] 15.0   85 1008    5    4
## [14,] 19.8   93  995    7    4
## [15,] 17.2   95  996   16    4
## [16,] 18.8   68  998    6    5
## [17,] 21.1   74 1004    9    5
## [18,] 18.5   88  995    5    4
## [19,] 16.1   75  999   19    4
## [20,] 17.4   58 1015   16    2
## [21,] 21.7   56 1002   18    5
## [22,] 19.2   83 1015    7    2
```

```
## [23,] 22.9   97 1007   18    2
## [24,] 16.0   62 1012   13    4
## [25,] 19.3   68 1004    6    5
## [26,] 24.8   96 1000   19    3
## [27,] 23.9   88 1001   10    3
## [28,] 23.9   53 1003    7    2
## [29,] 16.8   50 1010    7    2
## [30,] 16.3   89 1011   13    1
## [31,] 21.5   79 1015    6    3
```

```
#---------------------------------#

# 7-  The code for task 7 goes here

# Use matrix multiplication to calculate the product of the transpose of the matrix in task 3 with the

# vector from task 1
cat("Task 1: vector containing monthly average temperatures: \n")
```

```
## Task 1: vector containing monthly average temperatures:
```

```
cat("-----------------------------------------------------\n")
```

```
## -----------------------------------------------------
```

```
print(monthly_average_temperature)
```

```
## [1] 20.84839 19.29000 19.98710
```

```
cat("\n")
```

```
# matrix from task 3
cat("Task 3: matrix containing the average temperature, humidity, pressure, and wind speed readings for
```

```
## Task 3: matrix containing the average temperature, humidity, pressure, and wind speed readings for ea
```

```
cat("---------------------------------------------------------------------\n")
```

```
## ---------------------------------------------------------------------
```

```
print(monthly_averages)
```

```
##      average_temp average_humidity average_pressure average_wind_speed
## [1,]     20.84839         76.16129         1006.226           11.51613
## [2,]     19.29000         79.20000         1005.167           13.53333
## [3,]     19.98710         74.64516         1005.871           12.16129
```

```
cat("\n")
```

```
# matrix transpose
cat("Transpose of the matrix : \n")
```

```
## Transpose of the matrix :
```

```
cat("-----------------------------------------------------\n")
```

```
## -----------------------------------------------------
```

```
mat_transpose <- t(monthly_averages)
print(mat_transpose)
```

```
##                          [,1]       [,2]       [,3]
```

```
## average_temp         20.84839   19.29000   19.98710
## average_humidity     76.16129   79.20000   74.64516
## average_pressure   1006.22581 1005.16667 1005.87097
## average_wind_speed   11.51613   13.53333   12.16129
```

```
cat("\n")
```

```
# matrix multiplication
product <- mat_transpose %*% monthly_average_temperature
rownames(product) <- NULL
cat("\n Product of matrix multiplication:\n")
```

```
##
##  Product of matrix multiplication:
```

```
cat("---------------------------------\n")
```

```
## ---------------------------------
```

```
print(product)
```

```
##            [,1]
## [1,]  1206.2434
## [2,]  4607.5481
## [3,] 60472.2905
## [4,]   744.2196
```

```
#---------------------------------#
```

Submission for part C: Looping and Conditional Statements

Please follow this structure:

```
#---------------------------------#

# 1-  The code for task 1 goes here
# Write a loop that calculates the average pressure reading for each month and stores the results in a

# Initialize lists with month names as names
monthly_pressure_readings_sum <- list()
monthly_pressure_readings_count <- list()

for (i in 1:nrow(weather)) {
  key <- weather$Month[i]
  pressure <- as.numeric(weather$Pressure[i])

  # Check if the key (month) is already in the lists, and if not, initialize it
  if (!(key %in% names(monthly_pressure_readings_sum))) {
    monthly_pressure_readings_sum[[key]] <- 0
    monthly_pressure_readings_count[[key]] <- 0
  }

  # Update the sum and count for the corresponding month
  monthly_pressure_readings_sum[[key]] <- monthly_pressure_readings_sum[[key]] + pressure
  monthly_pressure_readings_count[[key]] <- monthly_pressure_readings_count[[key]] + 1
}
```

```r
average_monthly_pressure <- c()

for (month in names(monthly_pressure_readings_sum)) {
  average_value <- monthly_pressure_readings_sum[[month]]/monthly_pressure_readings_count[[month]]
  average_monthly_pressure <- c(average_monthly_pressure,average_value)
}

print("PART C: Q1: Average pressure reading for each month (using loop) and stores the results in a vec
```

```
## [1] "PART C: Q1: Average pressure reading for each month (using loop) and stores the results in a ve
```

```r
print(average_monthly_pressure)
```

```
## [1] 1006.226 1005.167 1005.871
```

```r
#---------------------------------#

# 2-  The code for task 2 goes here

# Use a conditional statement to determine how many days had a temperature above 25 degrees Celsius in

# variable to store the no. of instances where the Sydney temp is greater than 25 degree celcius
high_temp_count <- 0

# looping through df
for (i in 1:nrow(weather)) {
  # stores location
  location <- weather$Location[i]

  # stores the temperature
  temp <- weather$Temperature[i]

  # checks if location is Sydney and temp is greater than 25
  if(location == 'Sydney' & temp > 25) {
    # updates the temp count
    high_temp_count = high_temp_count + 1
  }
}

# print the output
print("PART: C: Q2: No. of days with temperature above 25 degree celcius in Sydney: ")
```

```
## [1] "PART: C: Q2: No. of days with temperature above 25 degree celcius in Sydney: "
```

```r
print(high_temp_count)
```

```
## [1] 0
```

```r
# to validate
# weather %>%
#   filter(Location == 'Sydney' & Temperature > 25)



#---------------------------------#
```

```r
# 3-  The code for task 3 goes here
# Use a for loop to calculate the average humidity for the days with a temperature below 21 degrees Cel

humidity_sum <- 0
days_count <- 0

for (i in 1:nrow(weather)) {
  # stores location
  location <- weather$Location[i]

  # stores the temperature
  temp <- weather$Temperature[i]

  # checks if location is Canberra and temp is less than 21
  if(location == 'Canberra' & temp < 21) {

    # update days count
    days_count <- days_count + 1

    # sum humidity to calculate average
    humidity_sum <- humidity_sum + weather$Humidity[i]
  }
}

average_humidity <- humidity_sum/days_count

print("PART C: Q3: Average humidity for the days with a temperature below 21 degrees Celsius in Canberra
```

```
## [1] "PART C: Q3: Average humidity for the days with a temperature below 21 degrees Celsius in Canberr
```

```r
print(average_humidity)
```

```
## [1] 77.92593
```

```r
# to validate
# weather %>%
#   filter(Location == 'Canberra' & Temperature < 21) %>%
#   summarise(mean_humidity=mean(Humidity))



#---------------------------------#

# 4-  The code for task 4 goes here
# Use a conditional statement to determine how many days had a UV index reading above 7 in both Canberr

# variable to store the high uv days
high_uv_days <- 0

for (i in 1:nrow(weather)) {
  # stores location
  location <- weather$Location[i]

  # stores the temperature
  uv_index <- weather$UVIndex[i]
```

```
  # checks if location is Sydney or Canberra and UV Index is greater than 7
  if(location %in% c('Canberra', 'Sydney') & uv_index > 7) {

    # update days count
    high_uv_days <- high_uv_days + 1
  }
}

print("PART C: Q4: No. of days that had a UV index reading above 7 in both Canberra and Sydney: ")
```

```
## [1] "PART C: Q4: No. of days that had a UV index reading above 7 in both Canberra and Sydney: "
```

```
print(high_uv_days)
```

```
## [1] 0
```

```
# to validate
# weather %>%
#   filter(Location %in% c('Canberra', 'Sydney') & UVIndex > 7) %>%
#   summarise(count=n())

#---------------------------------#
```

Submission for part D: Data Frame Manipulation

Please follow this structure:

```
#---------------------------------#

# 1-  The code for task 1 goes here
# Load the 5 files using read.csv or read_csv functions and combine them into one data frame.
# Please note that the first 7 rows of each files need to be ignored while loading the data.

# store filenames
file_names <- c('201812.csv', '201811.csv', '201810.csv', '201809.csv', '201808.csv')

# function that appends "data/" to the file names to make it a complete file path
create_file_paths <- function(file_name) {
  file_path <- paste("data",file_name, sep = "/")
  return(file_path)
}

# uses sapply() function to iterate through each file name and create a char vector of filepaths and st
file_paths <- sapply(file_names, create_file_paths)

# function to read a csv file skipping first 7 lines. takes the filepath as input
custom_read_csv <- function(file_path) {
  return(read.csv(file_path, skip=7))
}

# uses lapply() function to iterate through each filepath
# read each csv file as a dataframe and stores it in a list
data_list <- lapply(file_paths, custom_read_csv)
```

```r
# initializes a data frame to store the data from all the csv
combined_df <- data.frame()

# iterates through each df in the list
for(df in data_list) {

  # uses rbind to append each dataframe row-wise to the 'combined_df'
  combined_df <- rbind(combined_df, df)
}


#---------------------------------#

# 2-  The code for task 2 goes here
# Check the dimensions of the combined data frame.

# dimesnion of the combined dataframe
dim(combined_df)
```

```
## [1] 153  21
```

```r
#---------------------------------#

# 3-  The code for task 3 goes here
# Write a for loop to check the structure 'str()' and summary of each column.

# extracts the current column names of the dataframes
col_names <- colnames(combined_df)

col_names
```

```
##  [1] "Date"                              "Minimum.temperature"
##  [3] "Maximum.temperature"               "Rainfall..mm."
##  [5] "Evaporation..mm."                  "Sunshine..hours."
##  [7] "Direction.of.maximum.wind.gust"    "Speed.of.maximum.wind.gust..km.h."
##  [9] "Time.of.maximum.wind.gust"         "X9am.Temperature"
## [11] "X9am.relative.humidity...."        "X9am.cloud.amount..oktas."
## [13] "X9am.wind.direction"               "X9am.wind.speed..km.h."
## [15] "X9am.MSL.pressure..hPa."           "X3pm.Temperature"
## [17] "X3pm.relative.humidity...."        "X3pm.cloud.amount..oktas."
## [19] "X3pm.wind.direction"               "X3pm.wind.speed..km.h."
## [21] "X3pm.MSL.pressure..hPa."
```

```r
# iterates through each column
for(col in col_names) {
  cat("Column Name: ", col, "\n")
  cat('-------------------------\n')

  cat("Structure: \n")
  cat(str(combined_df[[col]]))
  cat("\n")

  cat("Summary: \n")
  print(summary(combined_df[[col]]))
```

```
  cat("\n")
}
```

```
## Column Name:  Date
## -------------------------
## Structure:
##  chr [1:153] "1/12/2018" "2/12/2018" "3/12/2018" "4/12/2018" "5/12/2018" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  Minimum.temperature
## -------------------------
## Structure:
##  num [1:153] 9 9.7 9.1 13 14.2 12.1 11.5 12.6 14.5 16 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.400   2.200   6.500   6.829  11.400  17.800
##
## Column Name:  Maximum.temperature
## -------------------------
## Structure:
##  num [1:153] 29.3 26.1 23.9 28.6 24.7 29.3 32.5 33.5 32.9 29 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.70   15.60   21.90   21.69   26.40   36.80
##
## Column Name:  Rainfall..mm.
## -------------------------
## Structure:
##  num [1:153] 0 0 0 0 0 0 0 0 0 17 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.766   0.200  33.200
##
## Column Name:  Evaporation..mm.
## -------------------------
## Structure:
##  logi [1:153] NA NA NA NA NA NA ...
##
## Summary:
##    Mode    NA's
## logical     153
##
## Column Name:  Sunshine..hours.
## -------------------------
## Structure:
##  logi [1:153] NA NA NA NA NA NA ...
##
## Summary:
```

```
##    Mode    NA's
## logical     153
##
## Column Name:  Direction.of.maximum.wind.gust
## -------------------------
## Structure:
##  chr [1:153] "NW" "WNW" "W" "E" "E" "ENE" "NNE" "NNW" "SSW" "N" "E" "ENE" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  Speed.of.maximum.wind.gust..km.h.
## -------------------------
## Structure:
##  int [1:153] 50 72 59 50 43 31 33 37 41 52 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.00   37.00   44.00   45.23   52.00   81.00
##
## Column Name:  Time.of.maximum.wind.gust
## -------------------------
## Structure:
##  chr [1:153] "12:52" "9:04" "12:49" "16:30" "18:32" "15:33" "20:09" "10:19" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  X9am.Temperature
## -------------------------
## Structure:
##  num [1:153] 17 25 15.9 18.9 16.1 15.9 19.1 22.8 25.2 17.5 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.50    9.60   13.60   13.43   17.00   29.50
##
## Column Name:  X9am.relative.humidity....
## -------------------------
## Structure:
##  int [1:153] 68 28 43 45 69 72 58 48 39 91 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.00   54.00   65.00   64.05   72.00   99.00
##
## Column Name:  X9am.cloud.amount..oktas.
## -------------------------
## Structure:
##  int [1:153] NA NA NA NA 8 7 NA 1 NA 8 ...
##
## Summary:
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   3.250   8.000   5.977   8.000   8.000      67
##
## Column Name:  X9am.wind.direction
## -------------------------
## Structure:
##  chr [1:153] "NW" "NW" "W" "W" "NE" "SE" "SSE" "WNW" "S" "NNE" "NE" "SE" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  X9am.wind.speed..km.h.
## -------------------------
## Structure:
##  chr [1:153] "9" "35" "28" "7" "13" "2" "6" "6" "4" "9" "11" "7" "9" "19" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  X9am.MSL.pressure..hPa.
## -------------------------
## Structure:
##  num [1:153] 1012 998 1009 1013 1021 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   997.7  1013.9  1018.2  1017.4  1021.8  1031.2
##
## Column Name:  X3pm.Temperature
## -------------------------
## Structure:
##  num [1:153] 28 21.4 23.4 27.3 22.4 27.5 31.7 30.9 28.1 26.5 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.40   14.10   19.80   19.97   24.80   36.20
##
## Column Name:  X3pm.relative.humidity....
## -------------------------
## Structure:
##  int [1:153] 17 34 29 25 46 36 21 21 32 47 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   27.00   37.00   39.49   48.00   99.00
##
## Column Name:  X3pm.cloud.amount..oktas.
## -------------------------
## Structure:
##  int [1:153] NA 6 NA NA 5 NA NA 1 7 NA ...
##
## Summary:
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   2.000   6.000   5.239   8.000   8.000      61
##
## Column Name:  X3pm.wind.direction
## -------------------------
## Structure:
##  chr [1:153] "NNW" "WNW" "WNW" "NW" "NE" "NNE" "N" "WNW" "SSE" "N" "ENE" ...
##
## Summary:
##    Length     Class      Mode
##       153 character character
##
## Column Name:  X3pm.wind.speed..km.h.
## -------------------------
## Structure:
##  int [1:153] 30 31 37 13 19 11 11 19 19 28 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   15.00   22.00   21.84   28.00   44.00
##
## Column Name:  X3pm.MSL.pressure..hPa.
## -------------------------
## Structure:
##  num [1:153] 1007 1000 1007 1012 1019 ...
##
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   996.7  1010.7  1015.1  1014.4  1018.6  1027.5
```

```r
#---------------------------------#

# 4-  The code for task 4 goes here
# Write a code to remove the variables, which have no data at all
# (i.e. all the records in these variables are NAs).

# get all the coloumn names of combined_df
combined_df_colnames <- colnames(combined_df)

# initialises a vector toi store column names which has all records as NA
empty_cols <- c()

# loop through each column name
for(col in combined_df_colnames) {
  # no. of records in a array with NA values
  na_index_length <- length(which(is.na(combined_df[[col]])))

  # check if the no. of records in a array with NA values is same a the total no. of rows in the datafr
  if(na_index_length == nrow(combined_df)) {
    empty_cols <- c(empty_cols, col)
  }
}

print(empty_cols)
```

```
## [1] "Evaporation..mm." "Sunshine..hours."
```

```
# removes the empty columns from the dataframe
combined_df <- combined_df %>%
  select(-empty_cols)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(empty_cols)
##
##   # Now:
##   data %>% select(all_of(empty_cols))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#----------------------------------#
# 5-  The code for task 5 goes here
# Write code to change the column names to have no spaces between the words
# and replace these spaces with underscore the _ character.

colnames(combined_df) <- gsub(" ", "_", colnames(combined_df))
# reference: https://www.geeksforgeeks.org/replace-spaces-in-column-names-in-r-dataframe/
colnames(combined_df)
```

```
##  [1] "Date"                              "Minimum.temperature"
##  [3] "Maximum.temperature"               "Rainfall..mm."
##  [5] "Direction.of.maximum.wind.gust"    "Speed.of.maximum.wind.gust..km.h."
##  [7] "Time.of.maximum.wind.gust"         "X9am.Temperature"
##  [9] "X9am.relative.humidity...."        "X9am.cloud.amount..oktas."
## [11] "X9am.wind.direction"               "X9am.wind.speed..km.h."
## [13] "X9am.MSL.pressure..hPa."           "X3pm.Temperature"
## [15] "X3pm.relative.humidity...."        "X3pm.cloud.amount..oktas."
## [17] "X3pm.wind.direction"               "X3pm.wind.speed..km.h."
## [19] "X3pm.MSL.pressure..hPa."
```

```
#----------------------------------#

# 6-  The code for task 6 goes here
# Write code to create two new columns for the month and year from the Date column.

class(combined_df$Date)
```

```
## [1] "character"
```

```
combined_df$Date <- as.Date(combined_df$Date, format = "%d/%m/%Y")

# uses seperate function to
combined_df <- combined_df %>%
  separate(col = Date, into = c("Month", "Year"), remove = FALSE)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 153 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```r
#----------------------------------#

# 7-  The code for task 7 goes here
# Write code to replace the remaining NAs with the median of the values of each column.


combined_df <- combined_df %>%
  mutate(across(where(is.numeric), ~replace_na(.,median(.,na.rm=TRUE))))

# reference: https://www.statology.org/r-replace-na-with-median/

#----------------------------------#

# 8-  The code for task 8 goes here
# Write code to save the combined weather data file with all of these changes to a single file.

write.csv(combined_df, file = "combined_weather.csv")

#----------------------------------#
```

Overall Reflection

In completing this assignment, I gained valuable insights into the fundamental principles of data wrangling and manipulation in the context of data science. The tasks presented in this assignment provided a hands-on experience that allowed me to apply theoretical concepts to real-world data. Here are some key lessons I learned:

Data Understanding is Crucial: The first part of the assignment emphasized the importance of thoroughly understanding the dataset before diving into analysis. I realized that by using summary statistics and visualizations like histograms, I could gain a quick overview of the data's distribution and characteristics. This step is essential for making informed decisions about data manipulation and analysis strategies.

Vector and Matrix Manipulation Skills: Part B challenged me to work with vectors and matrices to calculate averages and perform various calculations. I learned that these techniques are fundamental in data science for aggregating and summarizing data efficiently. Creating arrays also allowed me to work with multi-dimensional data structures, which are common in real-world datasets.

Looping and Conditional Statements: Part C reinforced my understanding of loops and conditional statements. I discovered that loops are powerful tools for automating repetitive tasks, such as calculating monthly averages in this assignment. Conditional statements helped me filter and analyze data based on specific criteria, enhancing my ability to extract meaningful insights.

Data Frame Manipulation: Part D provided a practical experience of working with messy, real-world data. I had to import, clean, and transform data from multiple CSV files, which is a common scenario in data science projects. Renaming columns, handling missing values, and extracting date components were crucial skills I honed in this part.

Data Wrangling is the Foundation: Throughout the assignment, I realized that data wrangling is the foundation of any data science project. Clean, well-structured data is essential for accurate analysis and modeling. The ability to preprocess and manipulate data efficiently is key to deriving meaningful insights and making informed decisions.

The Challenge of Real Data: Working with real-world data in Part D highlighted the challenges that often arise in data science projects. Dealing with missing values and ensuring data consistency required careful

attention and problem-solving skills. I also saw the significance of data cleaning and preprocessing in preparing data for further analysis.

In conclusion, this assignment has deepened my appreciation for the critical role of data wrangling in the data science workflow. It has equipped me with essential skills that I will carry forward into future assignments and, ultimately, into my career as a data scientist. I now understand that data preparation is not only a crucial step but also an art that demands attention to detail and the ability to adapt to the complexities of real-world data. I look forward to applying these lessons in future projects and further honing my data wrangling skills.