ASSIGNMENT 1
Output on the Console window

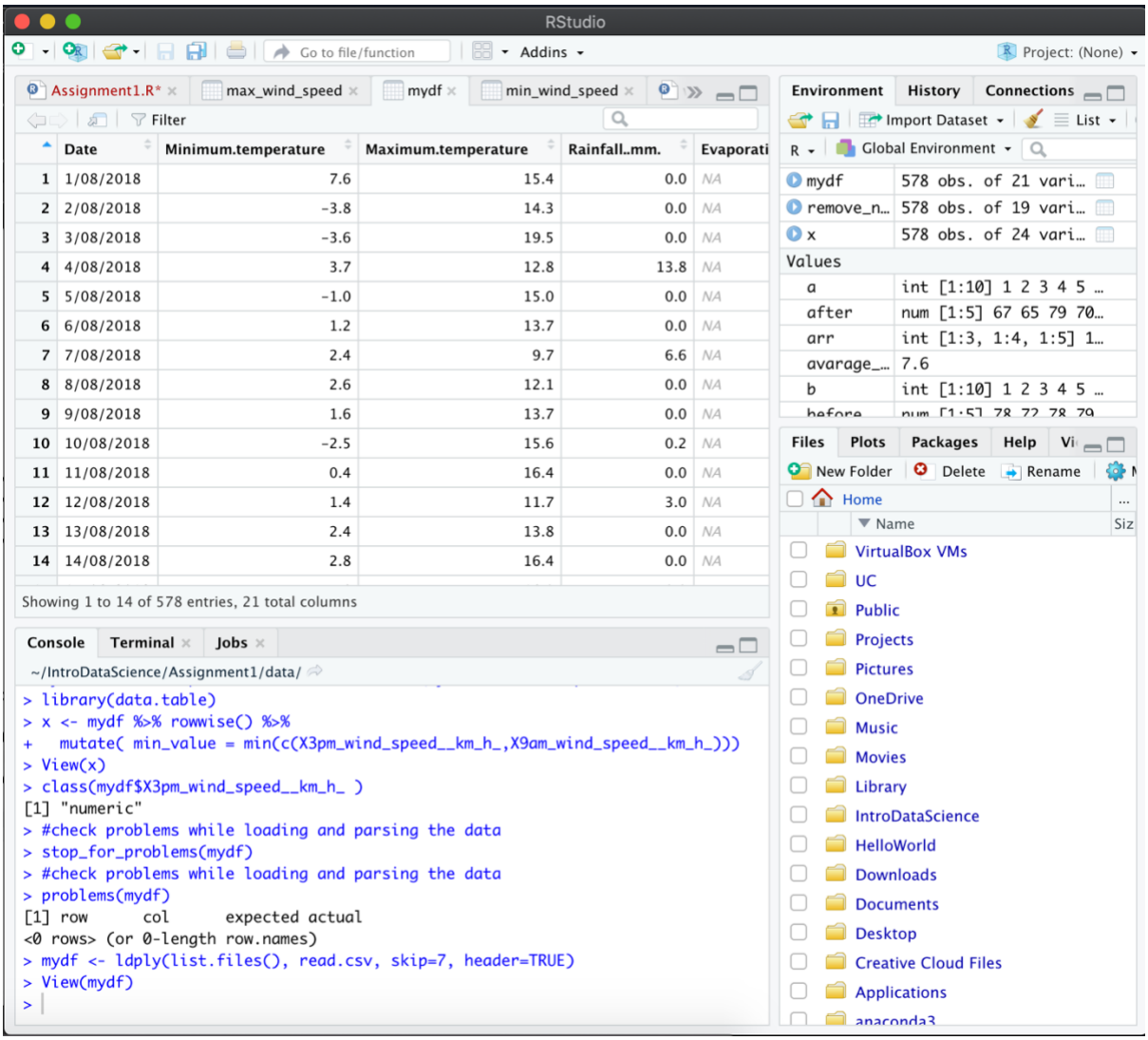
###################### PART A ######################

1. Load these files into working directory, one by one

```
> data1 <- read.csv("data/201808.csv", skip = 7)

> data2 <- read.csv("data/201809.csv", skip = 7)

> data3 <- read.csv("data/201810.csv", skip = 7)

> data4 <- read.csv("data/201811.csv", skip = 7)

> data5 <- read.csv("data/201812.csv", skip = 7)

> data6 <- read.csv("data/201901.csv", skip = 7)

> data7 <- read.csv("data/201902.csv", skip = 7)

> data8 <- read.csv("data/201903.csv", skip = 7)

> data9 <- read.csv("data/201904.csv", skip = 7)

> data10 <- read.csv("data/201905.csv", skip = 7)

> data11 <- read.csv("data/201906.csv", skip = 7)

> data12 <- read.csv("data/201907.csv", skip = 7)

> data13 <- read.csv("data/201908.csv", skip = 7)

> data14 <- read.csv("data/201909.csv", skip = 7)

> data15 <- read.csv("data/201910.csv", skip = 7)

> data16 <- read.csv("data/201911.csv", skip = 7)

> data17 <- read.csv("data/201912.csv", skip = 7)

> data18 <- read.csv("data/202001.csv", skip = 7)

> data19 <- read.csv("data/202002.csv", skip = 7)
```

2. Concatenate all the data of these file into one data frame

```
> list.files()
 [1] "201808.csv" "201809.csv" "201810.csv" "201811.csv" "201812.csv"
 [6] "201901.csv" "201902.csv" "201903.csv" "201904.csv" "201905.csv"
[11] "201906.csv" "201907.csv" "201908.csv" "201909.csv" "201910.csv"
[16] "201911.csv" "201912.csv" "202001.csv" "202002.csv"
```

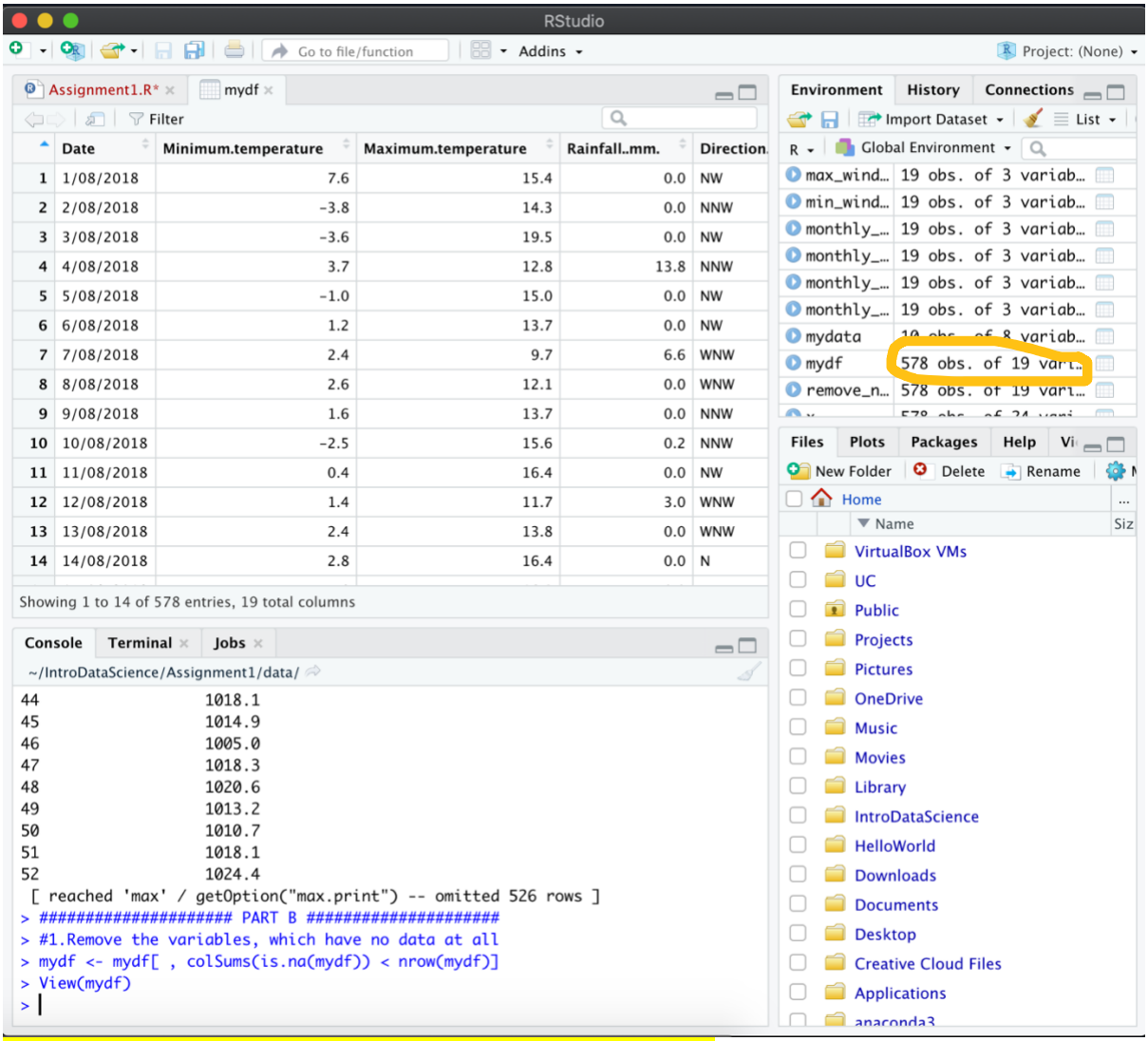



3. Check problems while loading and parsing the data

```
> #check problems while loading and parsing the data
> stop_for_problems(mydf)
> #check problems while loading and parsing the data
> problems(mydf)
[1] row      col      expected actual
<0 rows> (or 0-length row.names)
```

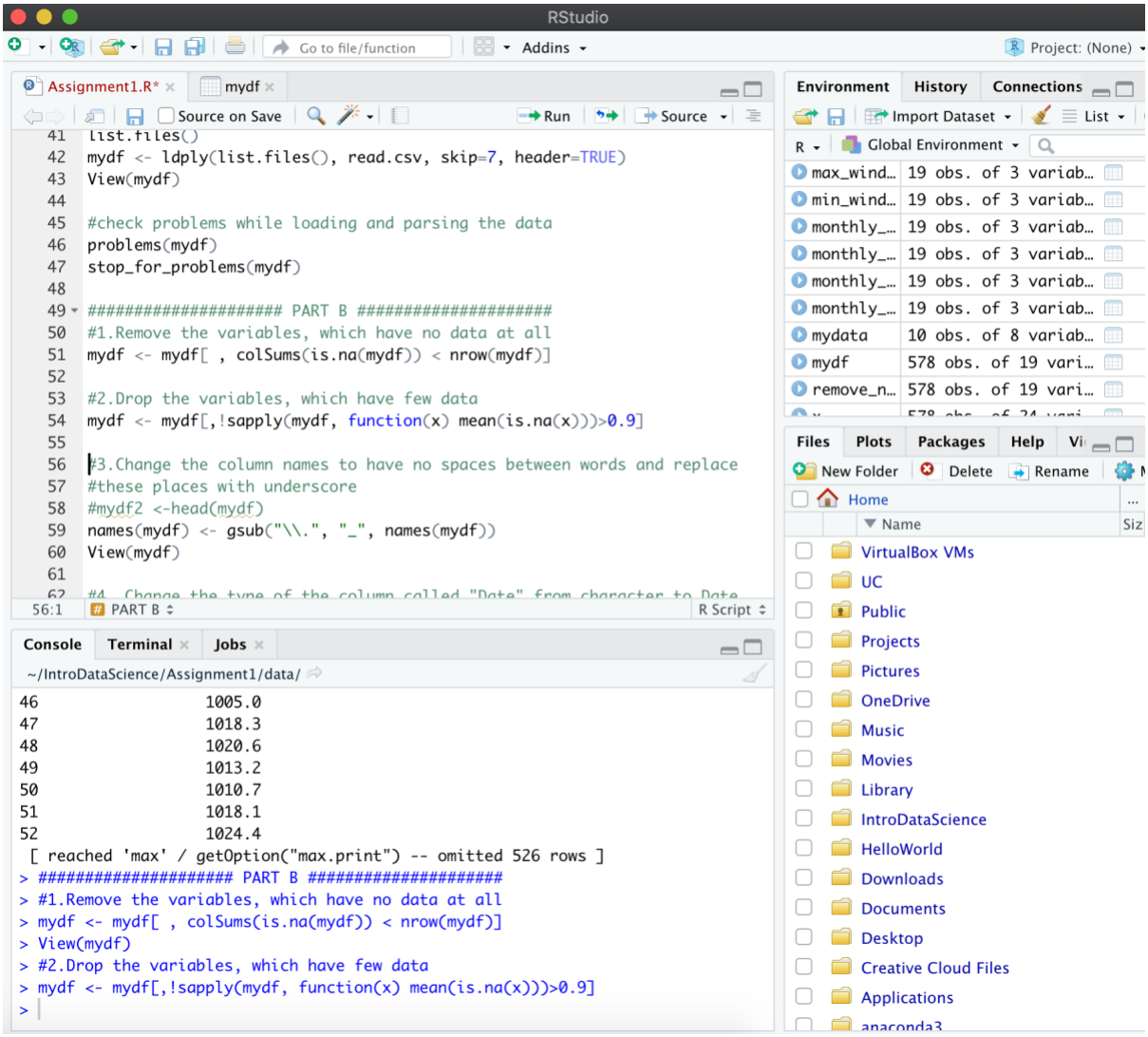
###################### PART B ######################

1. Remove the variables, which have no data at all

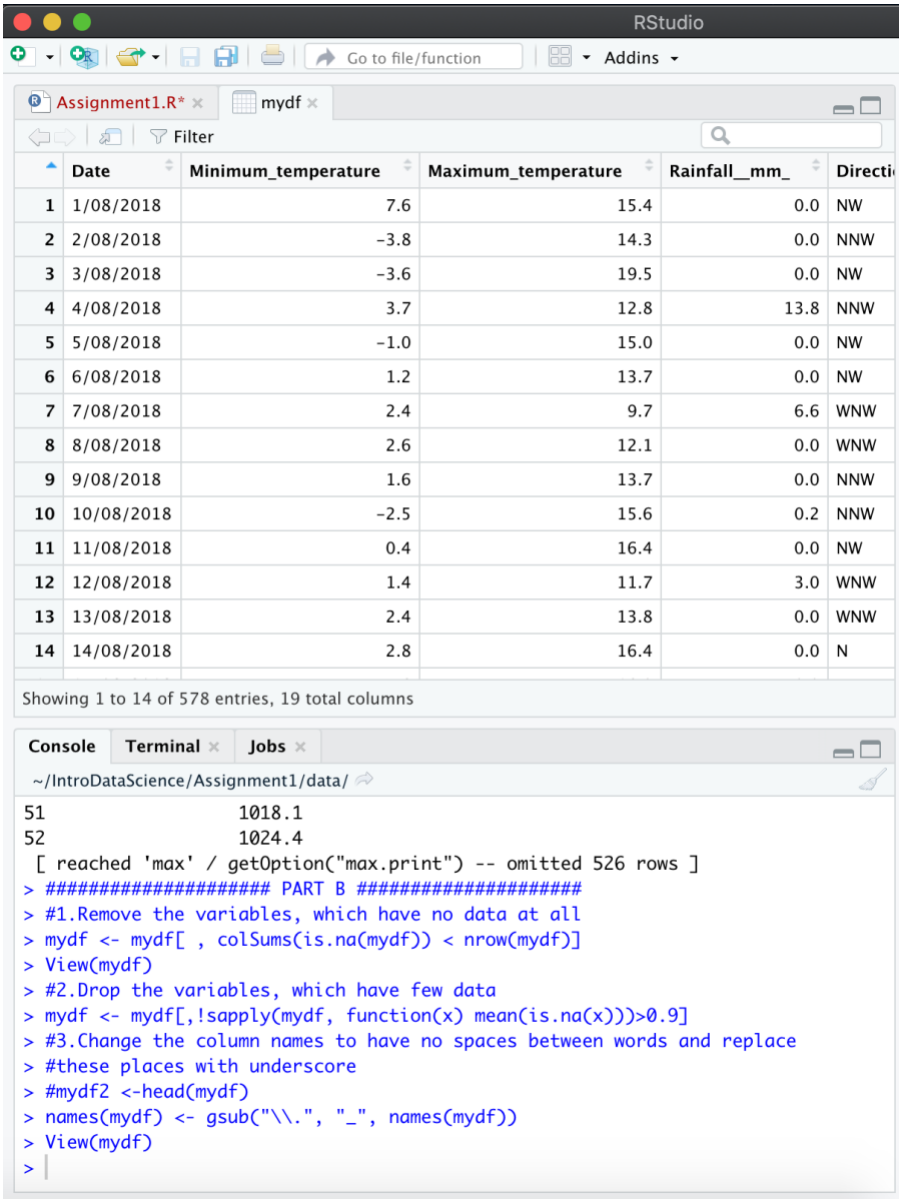



*There used to be 21 variables, down to 19 variables now

2. Drop the variables, which have few data




3. Change the column names to have no spaces between words and replace these places with underscore




| | Date | Minimum_temperature | Maximum_temperature | Rainfall__mm_ | Directi |
|---|---|---|---|---|---|
| 1 | 1/08/2018 | 7.6 | 15.4 | 0.0 | NW |
| 2 | 2/08/2018 | -3.8 | 14.3 | 0.0 | NNW |
| 3 | 3/08/2018 | -3.6 | 19.5 | 0.0 | NW |
| 4 | 4/08/2018 | 3.7 | 12.8 | 13.8 | NNW |
| 5 | 5/08/2018 | -1.0 | 15.0 | 0.0 | NW |
| 6 | 6/08/2018 | 1.2 | 13.7 | 0.0 | NW |
| 7 | 7/08/2018 | 2.4 | 9.7 | 6.6 | WNW |
| 8 | 8/08/2018 | 2.6 | 12.1 | 0.0 | WNW |
| 9 | 9/08/2018 | 1.6 | 13.7 | 0.0 | NNW |
| 10 | 10/08/2018 | -2.5 | 15.6 | 0.2 | NNW |
| 11 | 11/08/2018 | 0.4 | 16.4 | 0.0 | NW |
| 12 | 12/08/2018 | 1.4 | 11.7 | 3.0 | WNW |
| 13 | 13/08/2018 | 2.4 | 13.8 | 0.0 | WNW |
| 14 | 14/08/2018 | 2.8 | 16.4 | 0.0 | N |

4. Change the type of the column called "Date" from character to Date data type

```
> #4. Change the type of the column called "Date" from character to Date
> #data type
> mydf$Date <- as.Date(paste(mydf$Date), format= "%d/%m/%Y")
> class(mydf$Date)
[1] "Date"
```

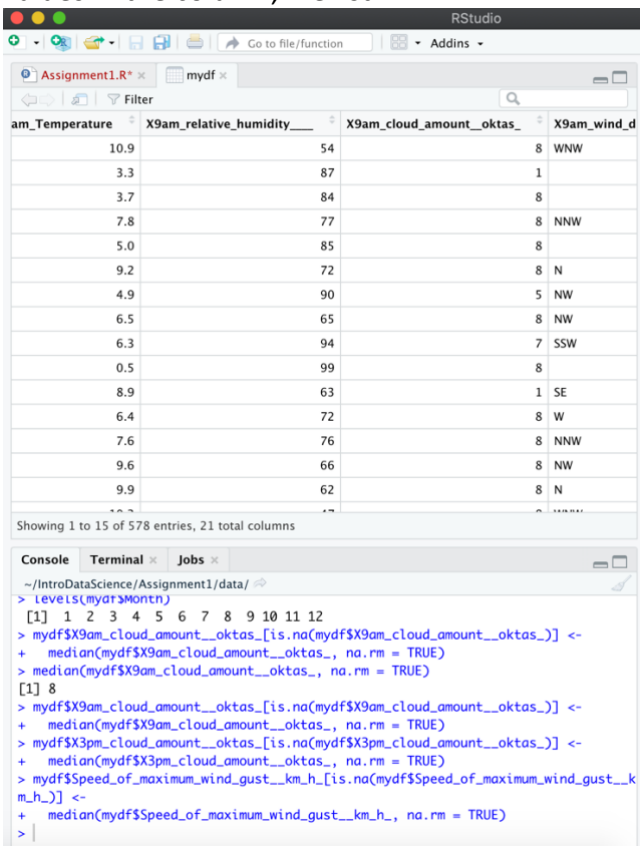
5. Add two new columns for the month and year of he data in each file, you may extract the contents of this column from the Date column. Please note that the data are collected for the 19 months across 3 years

```
Console  Terminal  Jobs
~/IntroDataScience/Assignment1/data/
> mydf %>% separate(Date, c("Year","Month","Date"),"-")
   Year Month Date Minimum_temperature Maximum_temperature Rainfall__mm_
1  2018    08   01                 7.6                15.4           0.0
2  2018    08   02                -3.8                14.3           0.0
3  2018    08   03                -3.6                19.5           0.0
4  2018    08   04                 3.7                12.8          13.8
5  2018    08   05                -1.0                15.0           0.0
6  2018    08   06                 1.2                13.7           0.0
7  2018    08   07                 2.4                 9.7           6.6
8  2018    08   08                 2.6                12.1           0.0
9  2018    08   09                 1.6                13.7           0.0
10 2018    08   10                -2.5                15.6           0.2
11 2018    08   11                 0.4                16.4           0.0
12 2018    08   12                 1.4                11.7           3.0
13 2018    08   13                 2.4                13.8           0.0
```

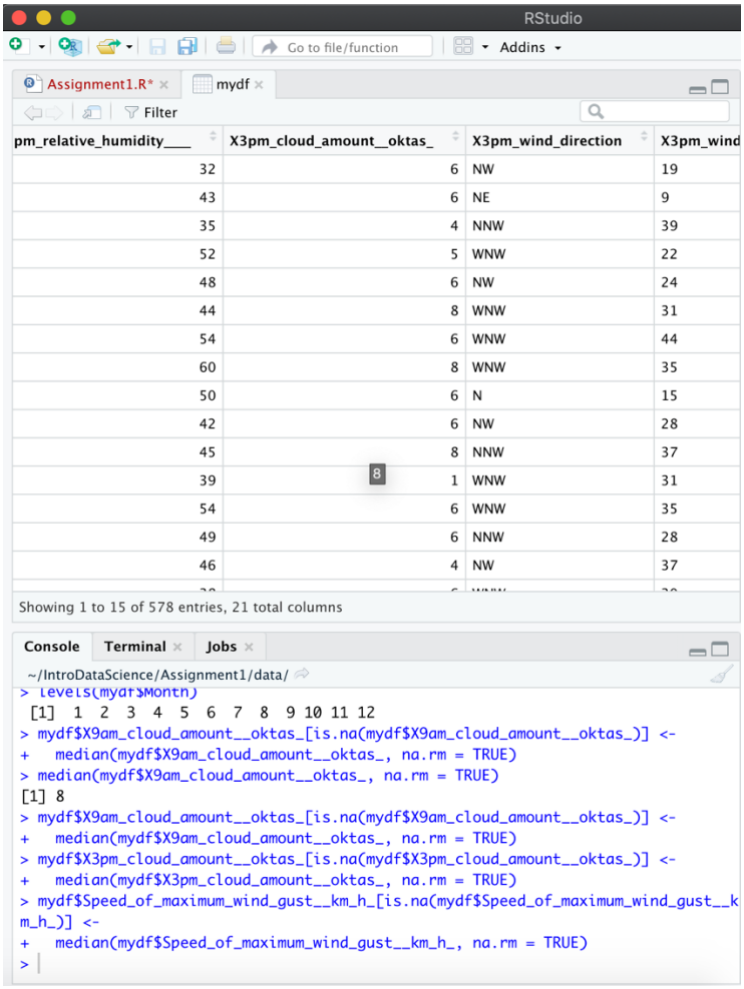
6. Change the type of the "Month" and "Year" columns from character to Ordinal with levels as the number of months in a year (12) and no of year(3)

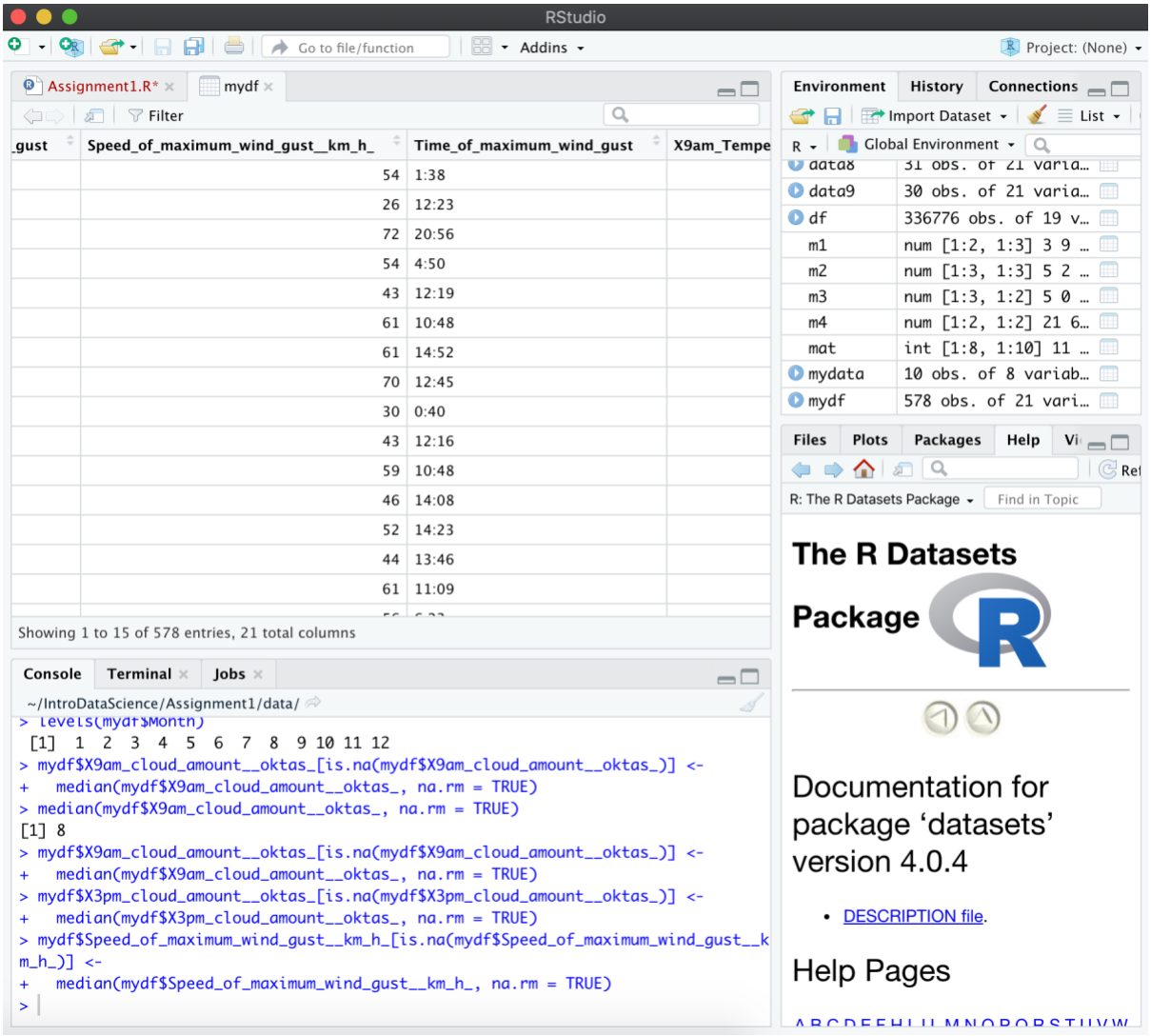
```
> #6. Change the type of the "Month" and "Year" columns from character to
> #Ordinal with levels as the number of months in a year (12) and no of year(3)
> mydf$Year <- as.numeric(mydf$Year)
> levels(mydf$Year)=c(2018, 2019, 2020)
> mydf$Month <- as.numeric(mydf$Month)
> levels(mydf$Month)=c(1,2,3,4,5,6,7,8,9,10,11,12)
> levels(mydf$Year)
[1] 2018 2019 2020
> levels(mydf$Month)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
>
```

7. For all the numeric columns, replace the remaining NAs with the median of the values in the column, if exist



8.


RStudio
Go to file/function
Addins
Assignment1.R*
mydf
Filter
pm_relative_humidity___
X3pm_cloud_amount__oktas_
X3pm_wind_direction
X3pm_wind
32 6 NW 19
43 6 NE 9
35 4 NNW 39
52 5 WNW 22
48 6 NW 24
44 8 WNW 31
54 6 WNW 44
60 8 WNW 35
50 6 N 15
42 6 NW 28
45 8 NNW 37
39 1 WNW 31
54 6 WNW 35
49 6 NNW 28
46 4 NW 37
Showing 1 to 15 of 578 entries, 21 total columns
Console
Terminal
Jobs
~/IntroDataScience/Assignment1/data/
> levels(mydf$Month)
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> mydf$X9am_cloud_amount__oktas_[is.na(mydf$X9am_cloud_amount__oktas_)] <-
+ median(mydf$X9am_cloud_amount__oktas_, na.rm = TRUE)
> median(mydf$X9am_cloud_amount__oktas_, na.rm = TRUE)
[1] 8
> mydf$X9am_cloud_amount__oktas_[is.na(mydf$X9am_cloud_amount__oktas_)] <-
+ median(mydf$X9am_cloud_amount__oktas_, na.rm = TRUE)
> mydf$X3pm_cloud_amount__oktas_[is.na(mydf$X3pm_cloud_amount__oktas_)] <-
+ median(mydf$X3pm_cloud_amount__oktas_, na.rm = TRUE)
> mydf$Speed_of_maximum_wind_gust__km_h_[is.na(mydf$Speed_of_maximum_wind_gust__km_h_)] <-
+ median(mydf$Speed_of_maximum_wind_gust__km_h_, na.rm = TRUE)
>




###################### PART C ######################

1. Print the summary of 'Minimum_temperature', '9am_temperature', 'Speed_of_maximum_wind_gust_(km/h)'

```
> #1a. Print the summary of 'Minimum_temperature'
> summary(mydf$Minimum_temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -6.40    2.00    8.00    7.84   13.60   26.70
> #1b. Print the summary of '9am_temperature'
> summary(mydf$X9am_Temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.50    9.20   14.70   14.09   18.60   34.50
> #1b. Print the summary of '9am_temperature'
> summary(mydf$X9am_Temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.50    9.20   14.70   14.09   18.60   34.50
> #1c. Print the summary of 'Speed_of_maximum_wind_gust_(km/h)'
> summary(mydf$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
>
```

2. Extracting the average of minimum temperature per month and year monthly

```
> #2.Extracting the average of minimum temperature per month and year
> #monthly
> aggregate( Minimum_temperature ~ Month + Year , mydf , mean)
   Month Year Minimum_temperature
1      8 2018          0.76451613
2      9 2018          1.82000000
3     10 2018          7.29677419
4     11 2018         10.47666667
5     12 2018         13.74516129
6      1 2019         17.67741935
7      2 2019         12.90714286
8      3 2019         12.00000000
9      4 2019          7.46666667
10     5 2019          3.33870968
11     6 2019         -0.08333333
12     7 2019          0.66774194
13     8 2019          0.10967742
14     9 2019          2.12333333
15    10 2019          6.21290323
16    11 2019          9.48666667
17    12 2019         13.09677419
18     1 2020         15.24193548
19     2 2020         15.06896552
> #yearly
> aggregate( Minimum_temperature ~ Year , mydf , mean)
  Year Minimum_temperature
1 2018            6.829412
2 2019            7.061370
3 2020           15.158333
>
```

3. Extracting the average of speed of maximum wind gust by direction of maximum wind gust

Assignment1.R | average_windgust | mydf

Filter

| | Direction_of_maximum_wind_gust | Speed_of_maximum_wind_gust__km_h_ |
|---|---|---|
| 1 | | 43.00000 |
| 2 | E | 41.94118 |
| 3 | ENE | 40.06667 |
| 4 | ESE | 42.54545 |
| 5 | N | 40.02381 |
| 6 | NE | 34.00000 |
| 7 | NNE | 32.14286 |
| 8 | NNW | 44.42268 |
| 9 | NW | 49.98305 |
| 10 | S | 44.71429 |
| 11 | SE | 39.27778 |
| 12 | SSE | 38.28571 |
| 13 | SSW | 40.66667 |
| 14 | SW | 49.50000 |
| 15 | W | 46.62069 |
| 16 | WNW | 50.20635 |
| 17 | WSW | 41.25000 |

Showing 1 to 17 of 17 entries, 2 total columns

Console | Terminal | Jobs

~/IntroDataScience/Assignment1/data/

```
+   aggregate(Speed_of_maximum_wind_gust__km_h_ ~ Direction_of_maximum_wind_gust
 , mydf , mean)
> View(average_windgust)
> view(average_windgust)
> #3. Extracting the average of speed of maximum wind gust by direction
> #of maximum wind gust
> average_windgust <-
+   aggregate(Speed_of_maximum_wind_gust__km_h_ ~ Direction_of_maximum_wind_gust
 , mydf , mean)
> view(average_windgust)
>
```

4. Which month was dry, if any? And in which year?

```
> #4. Which month was dry, if any? And in which year?
> monthly_rainfall <- aggregate(Rainfall__mm_  ~ Month + Year , mydf , FUN = sum
 )
> if (0.0 %in% monthly_rainfall$Rainfall__mm_){
+    driest_month <-
+      summarise(monthly_rainfall, year= Year[which.min(monthly_rainfall$Rainfall
__mm_)],
+                driest_month= Month[which.min(Rainfall__mm_)],
+                driest_value= min(Rainfall__mm_))
+ } else {
+   print("There is no dry month")
+ }
[1] "There is no dry month"
>
```

5. What about the humidity, which month in the ACT has the highest humidity level in 2019

```
> mydf_2019 <- filter(mydf, mydf$Year =="2019")
>
> aggregate(X9am_relative_humidity____+X3pm_relative_humidity____
+           ~ Month + Year , mydf_2019 , FUN = sum )
   Month Year X9am_relative_humidity____ + X3pm_relative_humidity____
1      1 2019                                                   2971
2      2 2019                                                   2661
3      3 2019                                                   3503
4      4 2019                                                   3508
5      5 2019                                                   4253
6      6 2019                                                   4348
7      7 2019                                                   4195
8      8 2019                                                   3786
9      9 2019                                                   2894
10    10 2019                                                   2618
11    11 2019                                                   2113
12    12 2019                                                   2063
> summarise(monthly_humidity,
+           max_month= Month[which.max(`X9am_relative_humidity____ + X3pm_relative_humidity____`)],
+           max_value= max(`X9am_relative_humidity____ + X3pm_relative_humidity____`))
  max_month max_value
1         6      4348
>
```

6. For 2019, extract the minimum, maximum and average temperature, wind speed and humidity per month and per quarter in 2019 only

7. Plot the histogram/bar-charts for each variable of the previous question