Assignment 4: Data Wrangling (Fall 2024)

Gretchen Barbera

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A04_DataWrangling.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. Ensure that code in code chunks does not extend off the page in the PDF.

Set up your session

1a. Load the `tidyverse`, `lubridate`, and `here` packages into your session.

1b. Check your working directory.

1c. Read in all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Add the appropriate code to reveal the dimensions of the four datasets.

```
#1a
install.packages(c("tidyverse", "lubridate","here", "readr"))
```

```
## Installing packages into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- 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
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(readr)
library(dplyr)

#1b
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
#1c
EPAair_PM25_NC2019_raw <- read.csv(
file=here("./Data/Raw/EPAair_PM25_NC2019_raw.csv") ,
stringsAsFactors= TRUE)


#View(EPAair_PM25_NC2019_raw)


EPAair_PM25_NC2018_raw <-  read.csv(
file=here("./Data/Raw/EPAair_PM25_NC2018_raw.csv") ,
stringsAsFactors= TRUE)

#View(EPAair_PM25_NC2018_raw)


EPAair_O3_NC2018_raw <-
   read.csv(
file=here("./Data/Raw/EPAair_O3_NC2018_raw.csv") ,
stringsAsFactors= TRUE)

#View(EPAair_O3_NC2018_raw)

EPAair_O3_NC2019_raw <- read.csv(
  file=here("./Data/Raw/EPAair_O3_NC2019_raw.csv") ,
  stringsAsFactors = TRUE)
```

```
#View(EPAair_O3_NC2019_raw)

#2
dim(EPAair_PM25_NC2018_raw)
```

```
## [1] 8983   20
```

```
dim(EPAair_PM25_NC2019_raw)
```

```
## [1] 8581   20
```

```
dim(EPAair_O3_NC2018_raw)
```

```
## [1] 9737   20
```

```
dim(EPAair_O3_NC2019_raw)
```

```
## [1] 10592    20
```

> All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern?

Yes!

Wrangle individual datasets to create processed files.

3. Change the Date columns to be date objects.
4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE
5. For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5" (all cells in this column should be identical).
6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace "raw" with "processed".

```
#3

EPAair_PM25_NC2018_raw$Date <- mdy(EPAair_PM25_NC2018_raw$Date)
#I am doing the mdy because that is what is shown when
#I open the dataset
 str(EPAair_PM25_NC2018_raw)
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Date, format: "2018-01-02" "2018-01-05" ...
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  2.9 3.7 5.3 0.8 2.5 4.5 1.8 2.5 4.2 1.7 ...
```

```
##  $ UNITS                          : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                      : Factor w/ 25 levels "","Blackstone",..: 15 15 15 15 15 15 15 15 15
##  $ DAILY_OBS_COUNT                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE               : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE             : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC             : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                      : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                     : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                          : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                    : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                         : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                  : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                 : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
#using the lubridate to make sure that I did it right

EPAair_PM25_NC2019_raw$Date <- mdy(EPAair_PM25_NC2019_raw$Date)

str(EPAair_PM25_NC2019_raw)
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                           : Date, format: "2019-01-03" "2019-01-06" ...
##  $ Source                         : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Site.ID                        : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  1.6 1 1.3 6.3 2.6 1.2 1.5 1.5 3.7 1.6 ...
##  $ UNITS                          : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                      : Factor w/ 25 levels "","Board Of Ed. Bldg.",..: 14 14 14 14 14 14
##  $ DAILY_OBS_COUNT                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE               : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE             : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC             : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                      : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                     : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                          : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                    : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                         : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                  : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                 : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
EPAair_O3_NC2018_raw$Date <- mdy(EPAair_O3_NC2018_raw$Date)
str(EPAair_O3_NC2018_raw)
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Date, format: "2018-03-01" "2018-03-02" ...
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 3700
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0
```

```
##  $ UNITS                              : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                    : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name                          : Factor w/ 40 levels "","Beaufort",..: 35 35 35 35 35 35 35 3
##  $ DAILY_OBS_COUNT                    : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                   : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                 : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                 : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                          : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                          : Factor w/ 17 levels "","Asheville, NC",..: 9 9 9 9 9 9 9 9 9
##  $ STATE_CODE                         : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                              : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                             : Factor w/ 32 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                      : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                     : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
EPAair_O3_NC2019_raw$Date <- mdy(EPAair_O3_NC2019_raw$Date)
str(EPAair_O3_NC2019_raw)
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                               : Date, format: "2019-01-01" "2019-01-02" ...
##  $ Source                             : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                            : int  370030005 370030005 370030005 370030005 370030005 37003
##  $ POC                                : int  1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.029 0.018 0.016 0.022 0.037 0.037 0.029 0.038 0.038 0
##  $ UNITS                              : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                    : int  27 17 15 20 34 34 27 35 35 28 ...
##  $ Site.Name                          : Factor w/ 38 levels "","Beaufort",..: 33 33 33 33 33 33 33 3
##  $ DAILY_OBS_COUNT                    : int  24 24 24 24 24 24 24 24 24 24 ...
##  $ PERCENT_COMPLETE                   : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                 : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                 : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                          : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                          : Factor w/ 15 levels "","Asheville, NC",..: 8 8 8 8 8 8 8 8 8
##  $ STATE_CODE                         : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                              : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                             : Factor w/ 30 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                      : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                     : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
#4

EPAair_PM25_NC2018_raw_selected_data <- EPAair_PM25_NC2018_raw %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)

#view(EPAair_2018_raw_selected_data)
#I named my new data selected_data and chose the columns that I am looking for
#using the select function in dplyr

EPAair_PM25_NC2019_raw_selected_data <- EPAair_PM25_NC2019_raw %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
```

```
        SITE_LATITUDE, SITE_LONGITUDE)
#View(EPAair_PM25_NC2019_raw_selected_data)



EPAair_O3_NC2018_raw_Selected_Data <- EPAair_O3_NC2018_raw %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)

#View(EPAair_O3_NC2018_raw_Selected_Data)


EPAair_O3_NC2019_raw_Selected_Data <- EPAair_O3_NC2019_raw %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)

#View(EPAair_O3_NC2019_raw_Selected_Data)

#5

#EPAair_PM25_NC2018_raw_selected_data$AQS_PARAMETER_DESC <- "PM2.5"

#I made it so all the values in this column now read PM2.5
#str(EPAair_O3_NC2018_raw_Selected_Data)

#I had a problem with this code so I had to go back into the data and see that
#this column was a factor- Have to change it to a character.
#it if a factor because I made the StringsAsFactors True earlier

EPAair_PM25_NC2018_raw_selected_data$AQS_PARAMETER_DESC <-
  as.character(EPAair_PM25_NC2018_raw_selected_data$AQS_PARAMETER_DESC)

EPAair_PM25_NC2018_raw_selected_data$AQS_PARAMETER_DESC <- "PM2.5"

#View(EPAair_PM25_NC2018_raw_selected_data)
#once I change from factor to character then it changed


EPAair_PM25_NC2019_raw_selected_data$AQS_PARAMETER_DESC <-
  as.character(EPAair_PM25_NC2019_raw_selected_data$AQS_PARAMETER_DESC)

EPAair_PM25_NC2019_raw_selected_data$AQS_PARAMETER_DESC <- "PM2.5"

#view(EPAair_PM25_NC2019_raw_selected_data)


#6

write.csv(EPAair_PM25_NC2019_raw_selected_data,
          file = "./Data/Processed/EPAair_PM25_NC2019_processed.csv",
          row.names = FALSE)

write.csv(EPAair_PM25_NC2018_raw_selected_data,
```

```
          file=
            "./Data/Processed/EPAair_PM25_NC2018_processed.csv",
          row.names = FALSE)

write.csv(EPAair_O3_NC2018_raw_Selected_Data,
          file = "./Data/Processed/EPAair_O3_NC2018_processed.csv",
          row.names = FALSE)

write.csv(EPAair_O3_NC2019_raw_Selected_Data,
          file = "./Data/Processed/EPAair_O3NC2019_processed.csv",
          row.names = FALSE)
```

Combine datasets

7. Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.
8. Wrangle your new dataset with a pipe function (%>%) so that it fills the following conditions:

- Include only sites that the four data frames have in common:

"Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue",
"Clemmons Middle", "Mendenhall School", "Frying Pan Mountain", "West Johnston Co.", "Garinger High School", "Castle Hayne", "Pitt Agri. Center", "Bryson City", "Millbrook School"

(the function `intersect` can figure out common factor levels - but it will include sites with missing site information, which you don't want...)

- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site name, AQS parameter, and county. Take the mean of the AQI value, latitude, and longitude.
- Add columns for "Month" and "Year" by parsing your "Date" column (hint: `lubridate` package)
- Hint: the dimensions of this dataset should be 14,752 x 9.

9. Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location on a specific date should now occupy only one row.
10. Call up the dimensions of your new tidy dataset.
11. Save your processed dataset with the following file name: "EPAair_O3_PM25_NC1819_Processed.csv"

```
#7

EPAair_O3_NC2018_Processed <-
  read.csv("./Data/Processed/EPAair_O3_NC2018_processed.csv")

#re uploading my new processed data

EPAair_O3NC2019_Processed <- read.csv("./Data/Processed/EPAair_O3NC2019_processed.csv")

EPAair_PM25_NC2018_Processed <- read.csv("./Data/Processed/EPAair_PM25_NC2018_processed.csv")
```

```r
EPAair_PM25_NC2019_Processed <- read.csv("./Data/Processed/EPAair_PM25_NC2019_processed.csv")

#colnames(EPAair_O3_NC2018_Processed)
#colnames(EPAair_O3NC2019_Processed)
#colnames(EPAair_PM25_NC2018_Processed)
#colnames(EPAair_PM25_NC2019_Processed)
#making sure they have identical names
#they do! score

combined_data <- rbind(EPAair_O3_NC2018_Processed,
                       EPAair_O3NC2019_Processed, EPAair_PM25_NC2018_Processed,
                       EPAair_PM25_NC2019_Processed)
#combining all the data


str(combined_data)
```

```
## 'data.frame':    37893 obs. of  7 variables:
##  $ Date              : chr  "2018-03-01" "2018-03-02" "2018-03-03" "2018-03-04" ...
##  $ DAILY_AQI_VALUE   : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name         : chr  "Taylorsville Liledoun" "Taylorsville Liledoun" "Taylorsville Liledoun" "
##  $ AQS_PARAMETER_DESC: chr  "Ozone" "Ozone" "Ozone" "Ozone" ...
##  $ COUNTY            : chr  "Alexander" "Alexander" "Alexander" "Alexander" ...
##  $ SITE_LATITUDE     : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE    : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```r
dim(combined_data)
```

```
## [1] 37893     7
```

```r
#looking at the structure and dimensions to make sure that they are good

#8

common_sites_combined_data <- c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue",
  "Clemmons Middle", "Mendenhall School", "Frying Pan Mountain",
  "West Johnston Co.", "Garinger High School", "Castle Hayne",
  "Pitt Agri. Center", "Bryson City", "Millbrook School")

#I created a vector with the common sites so it will be easier

filtered_data <- combined_data %>%
  filter(Site.Name %in% common_sites_combined_data)


combined_fitlered_data <- filtered_data %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(
    DAILY_AQI_VALUE = mean(DAILY_AQI_VALUE, na.rm = TRUE),
    SITE_LATITUDE = mean(SITE_LATITUDE, na.rm = TRUE),
    SITE_LONGITUDE = mean(SITE_LONGITUDE, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC'.
## You can override using the `.groups` argument.
```

```
#getting the mean values of the different variables


combined_fitlered_data <- combined_fitlered_data %>%
mutate(
  Month = month(Date),
               Year = year(Date))
#I added the month and year columns
#now I have the dimensions 14751 X 19



#9

combined_fitlered_data.spread <-
  pivot_wider(combined_fitlered_data,
names_from = AQS_PARAMETER_DESC,
values_from = DAILY_AQI_VALUE,
values_fill = list(DAILY_AQI_VALUE = NA))

colnames(combined_fitlered_data.spread)
```

```
## [1] "Date"           "Site.Name"      "COUNTY"         "SITE_LATITUDE"
## [5] "SITE_LONGITUDE" "Month"          "Year"           "PM2.5"
## [9] "Ozone"
```

```
#10

dim(combined_fitlered_data.spread)
```

```
## [1] 8976    9
```

```
#11

write.csv (combined_fitlered_data.spread,
          file = "./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv",
                  row.names = FALSE)
```

Generate summary tables

12. Use the split-apply-combine strategy to generate a summary data frame. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group. Then, add a pipe to remove instances where mean **ozone** values are not available (use the function `drop_na` in your pipe). It's ok to have missing mean PM2.5 values in this result.

13. Call up the dimensions of the summary dataset.

```
#12

EPAair_O3_PM25_NC1819_Processed <- read.csv("./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv")

#re up loading the dataset I just made and naming it based on how it was saved

EPAair_O3_PM25_NC1819_Processed_Summary_Data <-
  #my new data is called summay
  EPAair_O3_PM25_NC1819_Processed %>%
  #old data
  group_by(Site.Name, Month, Year) %>%
  #grouping sites
  summarise(
    mean_Ozone_AQI = mean(Ozone, na.rm = TRUE),
    #calculating for mean ozone
    mean_PM25_AQI = mean(PM2.5, na.rm = TRUE)
  ) %>%
  #calculating for mean PM2.5
      na.omit(mean_Ozone_AQI)
```

```
## `summarise()` has grouped output by 'Site.Name', 'Month'. You can override
## using the `.groups` argument.
```

```
#using the drop_na function I drop the rows where the mean value of ozone is NA

#View(EPAair_O3_PM25_NC1819_Processed_Summary_Data)

#13

dim(EPAair_O3_PM25_NC1819_Processed_Summary_Data)
```

```
## [1] 223   5
```

```
#new dimensions are 239 X 5
```

14. Why did we use the function `drop_na` rather than `na.omit`? Hint: replace `drop_na` with `na.omit` in part 12 and observe what happens with the dimensions of the summary date frame.

    Answer: When I ran the na.omit, the dimensions changed from 239 X 5 to 223 X 5. The na.omit removed all the rows that contained any NA in any dataframe and not just the the rows within the Mean_Ozone_AQI. The drop_na function let me just pick the specific column I wanted to omit the NAs from while the na.omit took out NA values from other columns.