Assignment 4: Data Wrangling

GuruBandaa Khalsa

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A03_DataExploration.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.

The completed exercise is due on Friday, Oct7th @ 5:00pm.

Set up your session

1. Check your working directory, load the `tidyverse` and `lubridate` packages, and upload all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in as 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. Explore the dimensions, column names, and structure of the datasets.

```
# 1. I checked my working directory using the getwd() function, loaded the
# `tidyverse` and `lubridate` packages, and uploaded all four raw data files
# associated with the EPA Air dataset.
getwd()
```

```
## [1] "/Users/survivormangb/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analy
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2
```

```
library(lubridate)
EPA1 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv", stringsAsFactors = TRUE)
EPA2 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv", stringsAsFactors = TRUE)
EPA3 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
EPA4 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)

# 2. I explored the dimensions, column names, and structure of the datasets
# using the dim(), View(), and str() functions.
dim(EPA1)
```

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

```
dim(EPA2)
```

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

```
dim(EPA3)
```

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

```
dim(EPA4)
```

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

```
colnames(EPA1)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```
colnames(EPA2)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```
colnames(EPA3)
```

```
##  [1] "Date"                           "Source"
##  [3] "Site.ID"                        "POC"
##  [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
##  [7] "DAILY_AQI_VALUE"                "Site.Name"
##  [9] "DAILY_OBS_COUNT"                "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"             "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"                      "CBSA_NAME"
## [15] "STATE_CODE"                     "STATE"
## [17] "COUNTY_CODE"                    "COUNTY"
## [19] "SITE_LATITUDE"                  "SITE_LONGITUDE"
```

```
colnames(EPA4)
```

```
##  [1] "Date"                           "Source"
##  [3] "Site.ID"                        "POC"
##  [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
##  [7] "DAILY_AQI_VALUE"                "Site.Name"
##  [9] "DAILY_OBS_COUNT"                "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"             "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"                      "CBSA_NAME"
## [15] "STATE_CODE"                     "STATE"
## [17] "COUNTY_CODE"                    "COUNTY"
## [19] "SITE_LATITUDE"                  "SITE_LONGITUDE"
```

```
str(EPA1)
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62
##  $ 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 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 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 1
##  $ 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 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 ...
```

```
str(EPA2)
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                                : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 1 2 3 4 5
##  $ Source                              : Factor w/ 2 levels "AirNow","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  0.029 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 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 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 1
##  $ 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 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 ...
```

```
str(EPA3)
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2018",..: 2 5 8 11 14 17
##  $ 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 ...
##  $ 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 ...
```

```
str(EPA4)
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 3 6 9 12 15 18
##  $ 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 ...
##  $ 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 ...
```

Wrangle individual datasets to create processed files.

3. Change date to date
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. I changed date to date.
EPA1$Date <- as.Date(EPA1$Date, format = "%m/%d/%Y")
EPA2$Date <- as.Date(EPA2$Date, format = "%m/%d/%Y")
EPA3$Date <- as.Date(EPA3$Date, format = "%m/%d/%Y")
EPA4$Date <- as.Date(EPA4$Date, format = "%m/%d/%Y")

# 4. I used the select() function to select the following columns: Date,
# DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE,
# SITE_LONGITUDE
EPA1.select <- select(EPA1, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA2.select <- select(EPA2, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA3.select <- select(EPA3, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA4.select <- select(EPA4, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

# 5. I filled all cells in AQS_PARAMETER_DESC with 'PM2.5' in the PM2.5
# datasets.
EPA3.select$AQS_PARAMETER_DESC <- "PM2.5"
EPA4.select$AQS_PARAMETER_DESC <- "PM2.5"

# 6. I renamed all four processed datasets and used the write.csv() function to
# save them in the Processed folder.
EPAair_O3_NC2018_processed.csv <- EPA1.select
EPAair_O3_NC2019_processed.csv <- EPA2.select
EPAair_PM25_NC2018_processed.csv <- EPA3.select
EPAair_PM25_NC2019_processed.csv <- EPA4.select

write.csv(EPAair_O3_NC2018_processed.csv, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2018_p
write.csv(EPAair_O3_NC2019_processed.csv, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2019_p
write.csv(EPAair_PM25_NC2018_processed.csv, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2
write.csv(EPAair_PM25_NC2019_processed.csv, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2
```

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 all 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)
- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site, 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_NC1718_Processed.csv"

```
# 7. I used the rbind() function to combine the four EPA datasets and the
# nrow() function to double check that there are 37,893 records.
EPA.bind <- rbind(EPAair_O3_NC2018_processed.csv, EPAair_O3_NC2019_processed.csv,
    EPAair_PM25_NC2018_processed.csv, EPAair_PM25_NC2019_processed.csv)

nrow(EPA.bind)
```

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

```
# 8. I used a pipe function to wrangle this new dataset so that it fills the
# above-listed conditions using the filter(), group_by(), filter(),
# summarise(), and mutate() functions.  I then used the dim() function to
# double check that the dimensions of the dataset are 14,752 x 9.
EPA.bind.processed <- EPA.bind %>%
    group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
    filter(Site.Name == "Linville Falls" | Site.Name == "Durham Armory" | Site.Name ==
        "Leggett" | Site.Name == "Hattie Avenue" | Site.Name == "Clemmons Middle" |
        Site.Name == "Mendenhall School" | Site.Name == "Frying Pan Mountain" | Site.Name ==
        "West Johnston Co." | Site.Name == "Garinger High School" | Site.Name ==
        "Castle Hayne" | Site.Name == "Pitt Agri. Center" | Site.Name == "Bryson City" |
        Site.Name == "Millbrook School") %>%
    summarise(meanAQI = mean(DAILY_AQI_VALUE), meanLAT = mean(SITE_LATITUDE), meanLONG = mean(SITE_LONG
    mutate(Month = month(Date), Year = year(Date))
```

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

```
dim(EPA.bind.processed)
```

```
## [1] 14752     9
```

```
# 9. I used the pivot_wider() function to spread the datasets such that AQI
# values for ozone and PM2.5 are in separate columns.
EPA.bind.spread <- pivot_wider(EPA.bind.processed, names_from = AQS_PARAMETER_DESC,
    values_from = meanAQI)

# 10. I used the dim() function to call up the dimensions of the new tidy
# dataset.  It is 8,976 x 9.
dim(EPA.bind.spread)
```

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

```
# 11. I used the write.csv() function to save the processed dataset with the
# following file name: 'EPAair_O3_PM25_NC1718_Processed.csv'.
EPAair_O3_PM25_NC1718_Processed.csv <- EPA.bind.spread
write.csv(EPAair_O3_PM25_NC1718_Processed.csv, row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM
```

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 a month and year combination (i.e. row) does not have corresponding ozone and PM2.5 data (use the function `drop_na` in your pipe).

13. Call up the dimensions of the summary dataset.

```
# 12a. I used the split-apply-combine strategy to generate a summary data frame
# with the data grouped by site, month, and year.  I also used the summarise()
# function to generate the AQI values for ozone and PM2.5 for each group.
EPAair_O3_PM25_NC1718_Summary <- EPAair_O3_PM25_NC1718_Processed.csv %>%
    group_by(Site.Name, Month, Year) %>%
    summarise(meanOzoneAQI = mean(Ozone), meanPM2.5AQI = mean(PM2.5))
```

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

```
# 12b. I added a pipe with the drop_na() function to remove instances where a
# month and year combination (i.e. row) does not have corresponding ozone and
# PM2.5 data.
EPAair_O3_PM25_NC1718_Summary <- EPAair_O3_PM25_NC1718_Processed.csv %>%
    group_by(Site.Name, Month, Year) %>%
    summarise(meanOzoneAQI = mean(Ozone), meanPM2.5AQI = mean(PM2.5))
```

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

```
drop_na(EPAair_O3_PM25_NC1718_Summary)
```

```
## # A tibble: 101 x 5
## # Groups:   Site.Name, Month [74]
##    Site.Name    Month  Year meanOzoneAQI meanPM2.5AQI
##    <fct>        <dbl> <dbl>        <dbl>        <dbl>
##  1 Bryson City      3  2018         41.6         34.7
##  2 Bryson City      4  2018         44.5         28.2
##  3 Bryson City      4  2019         45.4         26.7
##  4 Bryson City      7  2019         30.4         33.6
##  5 Bryson City      9  2018         25.4         25.1
##  6 Bryson City     10  2018         31           31.3
##  7 Castle Hayne     4  2018         48.7         14.9
##  8 Castle Hayne     4  2019         45.1         14.3
##  9 Castle Hayne     5  2019         42.8         16.5
## 10 Castle Hayne     7  2018         36.5         15.5
## # ... with 91 more rows
```

```
View(EPAair_O3_PM25_NC1718_Summary)

# 13. I used the dim() function to call up the dimensions of the summary
# dataset.  They are 308 x 5.
dim(EPAair_O3_PM25_NC1718_Summary)
```

```
## [1] 308   5
```

14. Why did we use the function `drop_na` rather than `na.omit`?

    Answer: We used the function `drop_na` rather than `na.omit` because `drop_na` removes rows with missing data in any columns present. The 'na.omit' function removes all incomplete cases of a singledata object. In this case, we want to remove instances where a month and year combination does not have corresponding ozone and PM2.5 data.