Assignment 4: Data Wrangling

Emma Wellbaum

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A04_DataWrangling.Rmd") prior to submission.

The completed exercise is due on Tuesday, Feb 16 @ 11:59pm.

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. 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
getwd()
```

```
## [1] "C:/Users/emmaw/Documents/ENV872/Environmental_Data_Analytics_2021"
```

```
library(tidyverse)
library(lubridate)
# upload all four EPA air datasets
EPAair.O3.2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv")
EPAair.O3.2019 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv")
EPAair.PM25.2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv")
EPAair.PM25.2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv")

#2
# The dimensions, column names, and structure of the EPAair.O3.2018 dataset
dim(EPAair.O3.2018)
```

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

```
colnames(EPAair.O3.2018)
```

```
##  [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"
```

```
str(EPAair.O3.2018, width=80, strict.width= "cut")
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : chr  "03/01/2018" "03/02/2018" "03/0"..
##  $ Source                              : chr  "AQS" "AQS" "AQS" "AQS" ...
##  $ Site.ID                             : int  370030005 370030005 370030005 37..
##  $ 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...
##  $ UNITS                               : chr  "ppm" "ppm" "ppm" "ppm" ...
##  $ DAILY_AQI_VALUE                     : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name                           : chr  "Taylorsville Liledoun" "Taylor"..
##  $ 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 ..
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44..
##  $ AQS_PARAMETER_DESC                  : chr  "Ozone" "Ozone" "Ozone" "Ozone" ..
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25..
##  $ CBSA_NAME                           : chr  "Hickory-Lenoir-Morganton, NC" "..
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : chr  "North Carolina" "North Carolin"..
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : chr  "Alexander" "Alexander" "Alexan"..
##  $ 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 ...
```

```
# The dimensions, column names, and structure of the EPAair.O3.2019 dataset
dim(EPAair.O3.2019)
```

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

```
colnames(EPAair.O3.2019)
```

```
##  [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"
```

```
str(EPAair.O3.2019, width=80, strict.width= "cut")
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                                : chr  "01/01/2019" "01/02/2019" "01/0"..
##  $ Source                              : chr  "AirNow" "AirNow" "AirNow" "Air"..
##  $ Site.ID                             : int  370030005 370030005 370030005 37..
##  $ 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...
##  $ UNITS                               : chr  "ppm" "ppm" "ppm" "ppm" ...
##  $ DAILY_AQI_VALUE                     : int  27 17 15 20 34 34 27 35 35 28 ...
##  $ Site.Name                           : chr  "Taylorsville Liledoun" "Taylor"..
##  $ 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 ..
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44..
##  $ AQS_PARAMETER_DESC                  : chr  "Ozone" "Ozone" "Ozone" "Ozone" ..
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25..
##  $ CBSA_NAME                           : chr  "Hickory-Lenoir-Morganton, NC" "..
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : chr  "North Carolina" "North Carolin"..
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : chr  "Alexander" "Alexander" "Alexan"..
##  $ 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 ...
```

```
# The dimensions, column names, and structure of the EPAair.PM25.2018 dataset
dim(EPAair.PM25.2018)
```

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

```
colnames(EPAair.PM25.2018)
```

```
##  [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(EPAair.PM25.2018, width=80, strict.width= "cut")
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : chr  "01/02/2018" "01/05/2018" "01/08/2018"..
##  $ Source                        : chr  "AQS" "AQS" "AQS" "AQS" ...
##  $ Site.ID                       : int  370110002 370110002 370110002 37011000..
##  $ 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...
##  $ UNITS                         : chr  "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" "ug/"..
##  $ DAILY_AQI_VALUE               : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                     : chr  "Linville Falls" "Linville Falls" "Li"..
##  $ 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 10..
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88..
##  $ AQS_PARAMETER_DESC            : chr  "Acceptable PM2.5 AQI & Speciation Ma"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : chr  "" "" "" "" ...
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : chr  "North Carolina" "North Carolina" "No"..
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : chr  "Avery" "Avery" "Avery" "Avery" ...
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```r
# The dimensions, column names, and structure of the EPAair.PM25.2019 dataset
dim(EPAair.PM25.2019)
```

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

```r
colnames(EPAair.PM25.2019)
```

```
##  [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"
```

```r
str(EPAair.PM25.2019, width=80, strict.width= "cut")
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                          : chr  "01/03/2019" "01/06/2019" "01/09/2019"..
##  $ Source                        : chr  "AQS" "AQS" "AQS" "AQS" ...
##  $ Site.ID                       : int  370110002 370110002 370110002 37011000..
##  $ 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                         : chr  "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" "ug/"..
##  $ DAILY_AQI_VALUE               : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                     : chr  "Linville Falls" "Linville Falls" "Li"..
##  $ 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 10..
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88..
##  $ AQS_PARAMETER_DESC            : chr  "Acceptable PM2.5 AQI & Speciation Ma"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
## $ CBSA_NAME                    : chr  "" "" "" "" ...
## $ STATE_CODE                   : int  37 37 37 37 37 37 37 37 37 37 ...
## $ STATE                        : chr  "North Carolina" "North Carolina" "No"..
## $ COUNTY_CODE                  : int  11 11 11 11 11 11 11 11 11 11 ...
## $ COUNTY                       : chr  "Avery" "Avery" "Avery" "Avery" ...
## $ 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. Reformat the Date column in each dataset as a date object
EPAair.O3.2018$Date <- as.Date(EPAair.O3.2018$Date, format = "%m/%d/%Y")
EPAair.O3.2019$Date <- as.Date(EPAair.O3.2019$Date, format = "%m/%d/%Y")
EPAair.PM25.2018$Date <- as.Date(EPAair.PM25.2018$Date, format = "%m/%d/%Y")
EPAair.PM25.2019$Date <- as.Date(EPAair.PM25.2019$Date, format = "%m/%d/%Y")

#4. Select the Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
# SITE_LATITUDE, SITE_LONGITUDE columns from each dataset
EPAair.O3.2018_processed <-
  select(EPAair.O3.2018,
         Date, DAILY_AQI_VALUE,
         Site.Name, AQS_PARAMETER_DESC,
         COUNTY:SITE_LONGITUDE)
EPAair.O3.2019_processed <-
  select(EPAair.O3.2019,
         Date, DAILY_AQI_VALUE,
         Site.Name,
         AQS_PARAMETER_DESC,
         COUNTY:SITE_LONGITUDE)
EPAair.PM25.2018_processed <-
  select(EPAair.PM25.2018,
         Date,
         DAILY_AQI_VALUE,
         Site.Name,
         AQS_PARAMETER_DESC,
         COUNTY:SITE_LONGITUDE)
EPAair.PM25.2019_processed <-
  select(EPAair.PM25.2019,
         Date,
         DAILY_AQI_VALUE,
         Site.Name,
         AQS_PARAMETER_DESC,
         COUNTY:SITE_LONGITUDE)

#5. For the PM2.5 datasets, fill the AQS_PARAMETER_DESC column with "PM2.5"
EPAair.PM25.2018_processed$AQS_PARAMETER_DESC <- "PM2.5"
EPAair.PM25.2019_processed$AQS_PARAMETER_DESC <- "PM2.5"
```

```
#6. Save processed datasets to the processed data folder
write.csv(EPAair.O3.2018_processed, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(EPAair.O3.2019_processed, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")
write.csv(EPAair.PM25.2018_processed, row.names = FALSE,
          file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(EPAair.PM25.2019_processed, row.names = FALSE,
          file = "./Data/Processed/EPAair_PM25_NC2019_processed.csv")
```

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. Combine the four datasets using rbind
EPAair_O3_PM25_NC1819 <- rbind(EPAair.O3.2018_processed,
                               EPAair.O3.2019_processed,
                               EPAair.PM25.2018_processed,
                               EPAair.PM25.2019_processed)

#8. Wrangle the datasets in a pipe
EPAair_O3_PM25_NC1819_processed <-
  EPAair_O3_PM25_NC1819 %>%
  # filter Site.Name to the common sites between the four datasets
  filter(Site.Name %in% c(intersect(
    intersect(EPAair.O3.2018_processed$Site.Name,
              EPAair.O3.2019_processed$Site.Name),
    intersect(EPAair.PM25.2018_processed$Site.Name,
              EPAair.PM25.2019_processed$Site.Name))) & Site.Name != "") %>%
  # group by date, site, AQS parameter, and county
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  # take the mean of the daily AQI value, latitude, and longitude
  summarize(Mean.AQI = mean(DAILY_AQI_VALUE),
            Mean.Latitude = mean(SITE_LATITUDE),
            Mean.Longitude = mean(SITE_LONGITUDE), .groups = "keep") %>%
  # add new columns for month and year based on the date column
  mutate(Month = month(Date),
```

```
         Year = year(Date))

#9. Spread the dataset so the PM2.5 and Ozone AQI values have their own columns
EPAair_O3_PM25_NC1819_processed_wider <-
  EPAair_O3_PM25_NC1819_processed %>%
  pivot_wider(
    id_cols = c(Date, Month, Year, Site.Name:Mean.Longitude),
    names_from = AQS_PARAMETER_DESC,
    values_from = Mean.AQI
    )

#10. Call up the dimensions of the spread dataset
dim(EPAair_O3_PM25_NC1819_processed_wider)
```

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

```
#11. Save the spread and processed dataset to the processed data folder
write.csv(EPAair_O3_PM25_NC1819_processed_wider, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv")
```

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 are not available (use the function `drop_na` in your pipe).
13. Call up the dimensions of the summary dataset.

```
#12a
EPAair_O3_PM25_NC1819_summary <-
  EPAair_O3_PM25_NC1819_processed_wider %>%
  group_by(Site.Name, Month, Year) %>% # group by site, month, and year
  summarize(Mean.PM2.5.AQI = mean(PM2.5), # generate mean AQI values for PM2.5
            Mean.Ozone.AQI = mean(Ozone), .groups = "keep") %>% # and Ozone
#12b
  drop_na(c(Month, Year)) # remove instances where Month and Year are NA

#13
dim(EPAair_O3_PM25_NC1819_summary) # Call up dimensions of the summary dataset
```

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

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

    Answer: We used drop_na rather than na.omit because we were not intereted in removing rows that with an NA value in ANY column, which is what na.omit does. We wanted to drop rows only if there were NA values in two specific columns – Month and Year – simultaneously. Using drop_na is more appropriate for this because we can target the NA values in specific columns. Here, we would not want to drop rows of data based NA values generally, as many of rows have NA values for either PM2.5 or ozone mean AQI values.