Assignment 4: Data Wrangling

Hanna Bliska

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 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. Explore the dimensions, column names, and structure of the datasets.

```
# 1
getwd()
```

```
## [1] "/Users/hbliska/Desktop/EDA-Fall2022"
```

```
# install.packages('tidyverse')
library(tidyverse)
# install.packages('lubridate')
library(lubridate)
EPAair.O3.2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv",
    stringsAsFactors = TRUE)
EPAair.O3.2019 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv",
    stringsAsFactors = TRUE)
EPAair.PM25.2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv",
    stringsAsFactors = TRUE)
EPAair.PM25.2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv",
    stringsAsFactors = TRUE)
```

```
# 2 Exploring EPAair.O3.2018
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)
```

```
## '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 ...
```

```
# Exploring EPAair.O3.2019
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)
```

```
## '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 37003
##  $ 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 ...
```

```
# Exploring EPAair.PM25.2018
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)
```

```
## '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 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 ...
```

```
# Exploring EPAair.PM25.2019
dim(EPAair.PM25.2019)
```

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

```
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"
```

```
str(EPAair.PM25.2019)
```

```
## '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 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
EPAair.O3.2018$Date <- as.Date(EPAair.O3.2018$Date,
    format = "%m/%d/%Y")
class(EPAair.O3.2018$Date)
```

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

```
EPAair.O3.2019$Date <- as.Date(EPAair.O3.2019$Date,
    format = "%m/%d/%Y")
class(EPAair.O3.2019$Date)
```

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

```
EPAair.PM25.2018$Date <- as.Date(EPAair.PM25.2018$Date,
    format = "%m/%d/%Y")
class(EPAair.PM25.2018$Date)
```

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

```
EPAair.PM25.2019$Date <- as.Date(EPAair.PM25.2019$Date,
    format = "%m/%d/%Y")
class(EPAair.PM25.2019$Date)
```

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

```
# used as.Date function to reformat the date
# column from a factor to a date.

# 4
AQI.EPAair.O3.2018 <- select(EPAair.O3.2018, Date,
    DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)

AQI.EPAair.O3.2019 <- select(EPAair.O3.2019, Date,
    DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)

AQI.EPAair.PM25.2018 <- select(EPAair.PM25.2018,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)

AQI.EPAair.PM25.2019 <- select(EPAair.PM25.2019,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)

# used select to create four data frames
# with columns of interest.

# 5
AQI.EPAair.PM25.2018$AQS_PARAMETER_DESC <- "PM2.5"
AQI.EPAair.PM25.2019$AQS_PARAMETER_DESC <- "PM2.5"

# modified column AQS_PARAMETER_DESC to have
# PM2.5 in each row in the two PM2.5 data
# frames.

# 6
write.csv(AQI.EPAair.O3.2018, row.names = FALSE,
```

```
    file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(AQI.EPAair.O3.2019, row.names = FALSE,
    file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")
write.csv(AQI.EPAair.PM25.2018, row.names = FALSE,
    file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(AQI.EPAair.PM25.2019, row.names = FALSE,
    file = "./Data/Processed/EPAair_PM25_NC2019_processed.csv")

# write.csv allowed me to save processed
# data sets.
```

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_NC1819_Processed.csv"

```
# 7
EPAair_O3_PM25_NC1819_Join <- rbind(AQI.EPAair.O3.2018,
    AQI.EPAair.O3.2019, AQI.EPAair.PM25.2018,
    AQI.EPAair.PM25.2019)

# used rbind function to join the O3 and
# PM2.5 data frames, which share identical
# column names.

# 8
EPAair_O3_PM25_NC1819_Processed <- EPAair_O3_PM25_NC1819_Join %>%
    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") %>%
```

```
    group_by(Date, Site.Name, AQS_PARAMETER_DESC,
        COUNTY) %>%
    summarise(meanAQI = mean(DAILY_AQI_VALUE),
        meanlat = mean(SITE_LATITUDE), meanlong = mean(SITE_LONGITUDE)) %>%
    mutate(month = month(Date)) %>%
    mutate(year = year(Date))
```

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

```
dim(EPAair_03_PM25_NC1819_Processed)
```

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

```
# for this pipe, used filter to include the
# sites that all four data frames have in
# common. Then, used group_by to combine
# rows that had the same Date, Site.Name,
# AQS_PARAMETER_DESC, and COUNTY into one
# row and used summarise to produce the mean
# of the DAILY_AQI_VALUE, SITE_LATITUDE, and
# SITE_LONGITUDE in each unique row. checked
# dimensions

# 9
EPAair_03_PM25_NC1819_Processed_Spread <- pivot_wider(EPAair_03_PM25_NC1819_Processed,
    names_from = AQS_PARAMETER_DESC, values_from = meanAQI)

# used pivot_wider to spread AQI values into
# two columns, one for ozone and one for
# PM2.5. Took names from AQS_PARAMETER_DESC
# and values from meanAQI.

# 10
dim(EPAair_03_PM25_NC1819_Processed_Spread)  #checked dimensions
```

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

```
# 11
write.csv(EPAair_03_PM25_NC1819_Processed_Spread,
    row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv")
# saved processed file
```

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 ozone and PM2.5 are not available (use the function `drop_na` in your pipe).

13. Call up the dimensions of the summary dataset.

```
# 12a
Summary.EPAair_03_PM25_NC1819 <- EPAair_03_PM25_NC1819_Processed_Spread %>%
    group_by(Site.Name, month, year) %>%
    summarise(mean.AQI.ozone = mean(Ozone), mean.AQI.PM2.5 = mean(PM2.5))
```

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

```
# generated summary table using pipe. Used
# group_by to combine rows with the same
# Site.Name, month, and year and summarise
# to produce the mean ozone and PM2.5 values
# for each unique row.

# 12b
Up.Summary.EPAair_03_PM25_NC1819 <- Summary.EPAair_03_PM25_NC1819 %>%
    drop_na(mean.AQI.ozone, mean.AQI.PM2.5)

# used drop_na to remove NAs in
# mean.AQI.ozone and mean.AQI.PM2.5 columns.

# 13
dim(Up.Summary.EPAair_03_PM25_NC1819)  #checked the dimensions
```

```
## [1] 101   5
```

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

    Answer: We used the function `drop_na` because we wanted to drop rows containing missing values in specific columns that we provided (in this case, O3 and PM2.5). If we were to use `na.omit` on our data frame, we would remove all NAs. In this case, using `na.omit` would yield the same data frame as `drop_na` because the only columns with NAs in our data frame were O3 and PM2.5, but if that were not the case and NAs were present in other columns, using `drop_na` would be better practice because it would allow us to only remove those NAs in specified columns. I also looked online and some articles recommended using `drop.na` if working with tidyverse, which we are doing in class.