Assignment 4: Data Wrangling

Melissa Merritt

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.

```
install.packages('formatR')
library(formatR)

knitr::opts_chunk$set(tidy.opts=list(width.cutoff=45), tidy = TRUE)
```

```
# 1
getwd()
```

```
## [1] "/home/guest/Documents/EDA-Fall2022"
```

```
# getwd() allows me to check my working
# directory. I needed to change it in the
# knit options to project directory, and now
# it is correct.

library(tidyverse)
library(lubridate)

O3_NC2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv",
    stringsAsFactors = TRUE)
O3_NC2019 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv",
    stringsAsFactors = TRUE)
PM25_NC2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv",
```

```
    stringsAsFactors = TRUE)
PM25_NC2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv",
    stringsAsFactors = TRUE)

# While importing the data files, I am able
# to name them in a way that makes them easy
# to use and recognize.


# 2
dim(O3_NC2018)
```

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

```
dim(O3_NC2019)
```

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

```
dim(PM25_NC2018)
```

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

```
dim(PM25_NC2019)
```

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

```
# dim() allows me to check the columns and
# rows for each data set.

colnames(O3_NC2018)
```

```
##  [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(O3_NC2019)
```

```
##  [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(PM25_NC2018)
```

```
##  [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(PM25_NC2019)
```

```
##  [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"
```

```
# the colnames() function allows me to see
# the names of each column in the data set.

str(O3_NC2018)
```

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

```
str(O3_NC2019)
```

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

```
str(PM25_NC2018)
```

```
## '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 ...
##  $ Site.ID                          : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                              : int  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 ...
##  $ 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 ...
```

```
str(PM25_NC2019)
```

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

```
# The str() function allows me to check the
# structure of a data set.
```

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
O3_NC2018$Date <- as.Date(O3_NC2018$Date, format = "%m/%d/%Y")
O3_NC2019$Date <- as.Date(O3_NC2019$Date, format = "%m/%d/%Y")
PM25_NC2018$Date <- as.Date(PM25_NC2018$Date,
    format = "%m/%d/%Y")
PM25_NC2019$Date <- as.Date(PM25_NC2019$Date,
    format = "%m/%d/%Y")

class(O3_NC2018$Date)
```

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

```
class(O3_NC2019$Date)
```

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

```
class(PM25_NC2018$Date)
```

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

```
class(PM25_NC2019$Date)
```

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

```
# I was able to use the as.Date() function
# to change the date column to 'date'. I was
# able to check each of the classifications
# with the class() function.

# 4
O3_NC2018_select <- select(O3_NC2018, Date, DAILY_AQI_VALUE,
    Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
O3_NC2019_select <- select(O3_NC2019, Date, DAILY_AQI_VALUE,
    Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
PM25_NC2018_select <- select(PM25_NC2018, Date,
    DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)
PM25_NC2019_select <- select(PM25_NC2019, Date,
    DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY:SITE_LONGITUDE)

# Using the select() function I am able to
# create a new data file with just the
# pertinent information.

# 5

PM25_NC2018_select$AQS_PARAMETER_DESC <- "PM2.5"
PM25_NC2019_select$AQS_PARAMETER_DESC <- "PM2.5"

# Using the column name I was able to rename
# the information in the columns to 'PM2.5'

# 6
write.csv(O3_NC2018_select, row.names = FALSE,
    file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(O3_NC2019_select, row.names = FALSE,
    file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")
write.csv(PM25_NC2018_select, row.names = FALSE,
    file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(PM25_NC2019_select, row.names = FALSE,
    file = "./Data/Processed/EPAair_PM25_NC2019_processed.csv")

# I am able to use the write.csv function to
# save the processed files in the processed
# data folder.
```

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 <- rbind(O3_NC2018_select,
    O3_NC2019_select, PM25_NC2018_select, PM25_NC2019_select)

# Using rbind() function allows me to
# combine the data files because all the row
# names are the same.

# 8
EPAair_O3_PM25_NC1819 <- EPAair_O3_PM25_NC1819 %>%
    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(Mean_AQI = mean(DAILY_AQI_VALUE),
        Mean_latitude = mean(SITE_LATITUDE), Mean_longitude = mean(SITE_LONGITUDE)) %>%
    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.
```

```
# I used the filter() function to gather
# just the sites that existed in all four of
# the data frames. Then I was able to use
# the group_by() function to clarify that
# for each row that had the same values in
# those four columns, we would take the
# means of the Daily AQI Value, the site
# latitude, and the site longitude. Lastly,
```

```
# I used the lubridate package, function
# mutate(), to create 2 new columns, month
# and year.

# 9
EPAair_O3_PM25_NC1819_spread <- pivot_wider(EPAair_O3_PM25_NC1819,
    names_from = AQS_PARAMETER_DESC, values_from = Mean_AQI)

# I used the pivot_wider() function to
# spread the data out. I got the names from
# the original AQS_PARAMETER_DESC column,
# and was able to create two new columns
# (ozone and PM2.5) with the values from the
# Mean_AQI.

# 10
dim(EPAair_O3_PM25_NC1819_spread)
```

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

```
# Once again the dim() function gives me the
# columns and the rows in the data frame.

# 11

write.csv(EPAair_O3_PM25_NC1819_spread, 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_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.
```

```
# 12b
EPAair_O3_PM25_NC1819_summary_NA <- EPAair_O3_PM25_NC1819_summary %>%
    drop_na(mean_AQI_ozone, mean_AQI_PM2.5)

# To summarize the data into our final data
# frame I grouped the data with the
# group_by() function, and then combined the
# ozone and PM2.5 into mean values for the
# rows that shared the same Site.Name,
# Month, and Year. The drop_NA function
# allows me to get rid of the NA values in
```

```
# both the mean_AQI_ozone column and the
# mean_AQI_PM2.5 column.

# 13

dim(EPAair_O3_PM25_NC1819_summary_NA)
```

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

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

    Answer: We used the function 'drop_na' over 'na.omit' because the na.omit function cannot observe a subset of columns. It will always omit all of the NAs. With the drop_na function, we are able to specify that we only want to omit rows that have NAs in the mean ozone column and the mean PM2.5 column. In this case, we get the same results for both, but in other cases it could be useful.