Assignment 4: Data Wrangling

Rachel Landman

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., "Salk_A04_DataWrangling.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 4 at 1:00 pm.

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] "/Users/rmlandman/Desktop/Data Analytics/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
library(lubridate)

EPA.O3.NC18.data <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv")
EPA.O3.NC19.data <- read.csv("../Data/Raw/EPAair_O3_NC2019_raw.csv")

EPA.PM25.NC18.data <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv")
EPA.PM25.NC19.data <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv")

#2
dim(EPA.O3.NC18.data)
```

```
## [1] 9737    7
```

```
colnames(EPA.O3.NC18.data)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.O3.NC18.data)
```

```
## 'data.frame':    9737 obs. of  7 variables:
##  $ Date              : Factor w/ 364 levels "2018-01-01","2018-01-02",..: 60 61 62 63 64 65 66 67 68
##  $ 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 35 35 35 ...
```

```
##  $ AQS_PARAMETER_DESC: Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY            : Factor w/ 32 levels "Alexander","Avery",..: 1 1 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 ...
```

```
dim(EPA.O3.NC19.data)
```

```
## [1] 10592     7
```

```
colnames(EPA.O3.NC19.data)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.O3.NC19.data)
```

```
## 'data.frame':    10592 obs. of  7 variables:
##  $ Date              : Factor w/ 365 levels "2019-01-01","2019-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 33 33 33 ...
##  $ AQS_PARAMETER_DESC: Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY            : Factor w/ 30 levels "Alexander","Avery",..: 1 1 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 ...
```

```
dim(EPA.PM25.NC18.data)
```

```
## [1] 8983    7
```

```
colnames(EPA.PM25.NC18.data)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.PM25.NC18.data)
```

```
## 'data.frame':    8983 obs. of  7 variables:
##  $ Date              : Factor w/ 365 levels "2018-01-01","2018-01-02",..: 2 5 8 11 14 17 20 23 26 29
##  $ 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 15 ...
##  $ AQS_PARAMETER_DESC: Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1 1 1 1 1 1 1
##  $ 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 ...
```

```
dim(EPA.PM25.NC19.data)
```

```
## [1] 8581    7
```

```
colnames(EPA.PM25.NC19.data)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.PM25.NC19.data)
```

```
## 'data.frame':    8581 obs. of  7 variables:
##  $ Date              : Factor w/ 365 levels "2019-01-01","2019-01-02",..: 3 6 9 12 15 18 21 24 27 30
##  $ 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 14 14 14 14
##  $ AQS_PARAMETER_DESC: Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1 1 1 1 1 1 1
##  $ 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

EPA.O3.NC18.data$Date <- as.Date(EPA.O3.NC18.data$Date, format = "%Y-%m-%d")
class(EPA.O3.NC18.data$Date) #check class
```

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

```
view(EPA.O3.NC18.data) #check to make sure date looks good

EPA.O3.NC19.data$Date <- as.Date(EPA.O3.NC19.data$Date, format = "%Y-%m-%d")
class(EPA.O3.NC19.data$Date) #check class
```

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

```
view(EPA.O3.NC19.data) #check to make sure date looks good

EPA.PM25.NC18.data$Date <- as.Date(EPA.PM25.NC18.data$Date, format = "%Y-%m-%d")
class(EPA.PM25.NC18.data$Date) #check class
```

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

```
view(EPA.PM25.NC18.data) #check to make sure date looks good

EPA.PM25.NC19.data$Date <- as.Date(EPA.PM25.NC19.data$Date, format = "%Y-%m-%d")
class(EPA.PM25.NC19.data$Date) #check class
```

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

```
view(EPA.PM25.NC19.data) #check to make sure date looks good

#4

EPA.O3.NC18.data.select <- select(EPA.O3.NC18.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES

EPA.O3.NC19.data.select <- select(EPA.O3.NC19.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES

EPA.PM25.NC18.data.select <- select(EPA.PM25.NC18.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER

EPA.PM25.NC19.data.select <- select(EPA.PM25.NC19.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER


#5
```

```
EPA.PM25.NC18.data.AQI <- mutate(EPA.PM25.NC18.data.select, AQS_PARAMETER_DESC = ("PM2.5"))
view(EPA.PM25.NC18.data.AQI)

EPA.PM25.NC19.data.AQI <- mutate(EPA.PM25.NC19.data.select, AQS_PARAMETER_DESC = ("PM2.5"))
view(EPA.PM25.NC19.data.AQI)


#6

write.csv(EPA.O3.NC18.data.select, row.names = FALSE, file = "../Data/Processed/EPAair_O3_NC2018_process

write.csv(EPA.O3.NC19.data.select, row.names = FALSE, file = "../Data/Processed/EPAair_O3_NC2019_process

write.csv(EPA.PM25.NC18.data.select, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2018_pr

write.csv(EPA.PM25.NC19.data.select, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2019_pr
```

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
colnames(EPA.O3.NC18.data.select)
```

```
## [1] "Date"                "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.O3.NC18.data.select)
```

```
## 'data.frame':    9737 obs. of  7 variables:
##  $ Date              : Date, format: "2018-03-01" "2018-03-02" ...
##  $ 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 35 35 35 ...
##  $ AQS_PARAMETER_DESC: Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY            : Factor w/ 32 levels "Alexander","Avery",..: 1 1 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 ...
```

```
EPA.O3.NC18.data.select$Date <- as.Date(EPA.O3.NC18.data.select$Date, format = "%Y-%m-%d")

colnames(EPA.O3.NC19.data.select)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.O3.NC19.data.select)
```

```
## 'data.frame':    10592 obs. of  7 variables:
##  $ Date              : Date, format: "2019-01-01" "2019-01-02" ...
##  $ 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 33 33 33 ...
##  $ AQS_PARAMETER_DESC: Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY            : Factor w/ 30 levels "Alexander","Avery",..: 1 1 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 ...
```

```
EPA.O3.NC19.data.select$Date <- as.Date(EPA.O3.NC19.data.select$Date, format = "%Y-%m-%d")

colnames(EPA.PM25.NC18.data.AQI)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
EPA.PM25.NC18.data.select$Date <- as.Date(EPA.PM25.NC18.data.select$Date, format = "%Y-%m-%d")
str(EPA.PM25.NC18.data.select)
```

```
## 'data.frame':    8983 obs. of  7 variables:
##  $ Date              : Date, format: "2018-01-02" "2018-01-05" ...
##  $ 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 15 ...
##  $ AQS_PARAMETER_DESC: Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1 1 1 1 1 1 1
##  $ 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 ...
```

```
colnames(EPA.PM25.NC19.data.AQI)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
str(EPA.PM25.NC19.data.select)
```

```
## 'data.frame':    8581 obs. of  7 variables:
##  $ Date              : Date, format: "2019-01-03" "2019-01-06" ...
##  $ 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 14 14 14 14
##  $ AQS_PARAMETER_DESC: Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1 1 1 1 1 1 1
##  $ 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 ...
```

```
EPA.PM25.NC19.data.select$Date <- as.Date(EPA.PM25.NC19.data.select$Date, format = "%Y-%m-%d")
```

```
### Had to change the date columns to a date because I got an error message when trying to rbind.
### Once I changed them to a date, the error went away

EPA.AQI.18_19 <- rbind(EPA.O3.NC18.data.select, EPA.O3.NC19.data.select, EPA.PM25.NC18.data.AQI, EPA.PM

class(EPA.AQI.18_19$Date)
```

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

```
EPA.AQI.18_19$Date <- as.Date(EPA.AQI.18_19$Date, format = "%Y-%m-%d")

#8

EPA.AQI.18_19.combined <-
  EPA.AQI.18_19 %>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(meanaqi = mean(DAILY_AQI_VALUE),
            meanlatitude = mean(SITE_LATITUDE),
            meanlongitude = mean(SITE_LONGITUDE)) %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date))

dim(EPA.AQI.18_19.combined)
```

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

```
#9
EPA.AQI.18_19.spread <- spread(EPA.AQI.18_19.combined, AQS_PARAMETER_DESC, meanaqi)
colnames(EPA.AQI.18_19.spread)
```

```
## [1] "Date"          "Site.Name"     "COUNTY"        "meanlatitude"
## [5] "meanlongitude" "month"         "year"          "Ozone"
## [9] "PM2.5"
```

```
#10
dim(EPA.AQI.18_19.spread)
```

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

```
#11
write.csv(EPA.AQI.18_19.spread, row.names = FALSE, file = "../Data/Processed/EPAair_O3_PM25_NC1718_Proc
```

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

EPA.AQI.18_19.summaries <-
  EPA.AQI.18_19.spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanAQIO3 = mean(Ozone),
```

```
            meanAQIPM25 = mean (PM2.5))

#12b

EPA.AQI.18_19.summaries <-
  EPA.AQI.18_19.spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanAQIO3 = mean(Ozone),
            meanAQIPM25 = mean (PM2.5))%>%
  drop_na(month) %>%
  drop_na(year)

#13
dim(EPA.AQI.18_19.summaries)
```

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

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

    Answer: drop_na allowed us to just drop the NAs from the month and year column. na.omit would have also dropped NAs from the PM2.5 and O3 columns.