Assignment 4: Data Wrangling

Andrew Brantley

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 Monday, Feb 7 @ 7:00pm.

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. 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/AndrewBrantley/Library/CloudStorage/Box-Box/Environmental Data Analytics/GithubRepos/Envi
```

```
library(tidyverse)
library(lubridate)

EPA.O3.NC2018 <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv")
EPA.O3.NC2019 <- read.csv("../Data/Raw/EPAair_O3_NC2019_raw.csv")
EPA.PM25.NC2018 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv")
EPA.PM25.NC2019 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv")

#2

#getting dimensions of datasets
dim(EPA.O3.NC2018)
```

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

```
dim(EPA.O3.NC2019)
```

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

```
dim(EPA.PM25.NC2018)
```

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

```
dim(EPA.PM25.NC2019)
```

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

```
#getting column names for each dataset
colnames(EPA.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(EPA.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"
```

```
#checking out the structure of these datasets with `head` function
head(EPA.O3.NC2018)
```

```
##         Date Source   Site.ID POC Daily.Max.8.hour.Ozone.Concentration UNITS
## 1 03/01/2018    AQS 370030005   1                                 0.043   ppm
## 2 03/02/2018    AQS 370030005   1                                 0.046   ppm
## 3 03/03/2018    AQS 370030005   1                                 0.047   ppm
## 4 03/04/2018    AQS 370030005   1                                 0.049   ppm
## 5 03/05/2018    AQS 370030005   1                                 0.047   ppm
## 6 03/06/2018    AQS 370030005   1                                 0.030   ppm
##   DAILY_AQI_VALUE            Site.Name DAILY_OBS_COUNT PERCENT_COMPLETE
## 1              40 Taylorsville Liledoun              17              100
```

```
## 2              43 Taylorsville Liledoun              17              100
## 3              44 Taylorsville Liledoun              17              100
## 4              45 Taylorsville Liledoun              17              100
## 5              44 Taylorsville Liledoun              17              100
## 6              28 Taylorsville Liledoun              17              100
##   AQS_PARAMETER_CODE AQS_PARAMETER_DESC CBSA_CODE                  CBSA_NAME
## 1              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
## 2              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
## 3              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
## 4              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
## 5              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
## 6              44201              Ozone     25860 Hickory-Lenoir-Morganton, NC
##   STATE_CODE          STATE COUNTY_CODE    COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1         37 North Carolina           3 Alexander       35.9138        -81.191
## 2         37 North Carolina           3 Alexander       35.9138        -81.191
## 3         37 North Carolina           3 Alexander       35.9138        -81.191
## 4         37 North Carolina           3 Alexander       35.9138        -81.191
## 5         37 North Carolina           3 Alexander       35.9138        -81.191
## 6         37 North Carolina           3 Alexander       35.9138        -81.191
```

```
head(EPA.PM25.NC2018)
```

```
##         Date Source   Site.ID POC Daily.Mean.PM2.5.Concentration    UNITS
## 1 01/02/2018    AQS 370110002   1                            2.9 ug/m3 LC
## 2 01/05/2018    AQS 370110002   1                            3.7 ug/m3 LC
## 3 01/08/2018    AQS 370110002   1                            5.3 ug/m3 LC
## 4 01/11/2018    AQS 370110002   1                            0.8 ug/m3 LC
## 5 01/14/2018    AQS 370110002   1                            2.5 ug/m3 LC
## 6 01/17/2018    AQS 370110002   1                            4.5 ug/m3 LC
##   DAILY_AQI_VALUE      Site.Name DAILY_OBS_COUNT PERCENT_COMPLETE
## 1              12 Linville Falls               1              100
## 2              15 Linville Falls               1              100
## 3              22 Linville Falls               1              100
## 4               3 Linville Falls               1              100
## 5              10 Linville Falls               1              100
## 6              19 Linville Falls               1              100
##   AQS_PARAMETER_CODE                      AQS_PARAMETER_DESC CBSA_CODE CBSA_NAME
## 1              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
## 2              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
## 3              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
## 4              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
## 5              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
## 6              88502 Acceptable PM2.5 AQI & Speciation Mass        NA
##   STATE_CODE          STATE COUNTY_CODE COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1         37 North Carolina          11  Avery      35.97235      -81.93307
## 2         37 North Carolina          11  Avery      35.97235      -81.93307
## 3         37 North Carolina          11  Avery      35.97235      -81.93307
## 4         37 North Carolina          11  Avery      35.97235      -81.93307
## 5         37 North Carolina          11  Avery      35.97235      -81.93307
## 6         37 North Carolina          11  Avery      35.97235      -81.93307
```

Wrangle individual datasets to create processed files.

3. Change date to a date object
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

# processing EPA.O3.NC2018 dataset
EPA.O3.NC2018$Date <- as.Date(EPA.O3.NC2018$Date, format = "%m/%d/%Y")
EPA.O3.NC2018.processed <- select(EPA.O3.NC2018, Date, DAILY_AQI_VALUE,
                                  Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)

#saving to processed data folder
write.csv(EPA.O3.NC2018.processed, row.names = FALSE,
          "../Data/Processed/EPA_O3_NC2018_processed")

#4

# processing EPA.O3.NC2019 dataset
EPA.O3.NC2019$Date <- as.Date(EPA.O3.NC2019$Date, format = "%m/%d/%Y")
EPA.O3.NC2019.processed <- select(EPA.O3.NC2019, Date, DAILY_AQI_VALUE,
                                  Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)

#saving to processed data folder
write.csv(EPA.O3.NC2019.processed, row.names = FALSE,
          "../Data/Processed/EPA_O3_NC2019_processed")

#5

# processing EPA.PM25.NC2018 dataset
EPA.PM25.NC2018$Date <- as.Date(EPA.PM25.NC2018$Date, format = "%m/%d/%Y")
EPA.PM25.NC2018.processed <- select(EPA.PM25.NC2018, Date, DAILY_AQI_VALUE, Site.Name,
                                    AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
EPA.PM25.NC2018.processed$AQS_PARAMETER_DESC <- c("PM 2.5")

#saving to processed data folder
write.csv(EPA.PM25.NC2018.processed, row.names = FALSE,
          "../Data/Processed/EPA_PM25_NC2018_processed")

#6

# processing EPA.PM25.NC2018 dataset
EPA.PM25.NC2019$Date <- as.Date(EPA.PM25.NC2019$Date, format = "%m/%d/%Y")
EPA.PM25.NC2019.processed <- select(EPA.PM25.NC2019, Date, DAILY_AQI_VALUE, Site.Name,
                                    AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
EPA.PM25.NC2019.processed$AQS_PARAMETER_DESC <- c("PM 2.5")

#saving to processed data folder
write.csv(EPA.PM25.NC2019.processed, row.names = FALSE,
          "../Data/Processed/EPA_PM25_NC2019_processed")
```

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:

- Filter records to include just the 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 `intersect` function can figure out common factor levels if we didn't give you this list...)
- 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_NC2122_Processed.csv"

```
#7

EPA.O3.PM25.2018.2019 <- rbind(EPA.O3.NC2018.processed, EPA.O3.NC2019.processed,
                               EPA.PM25.NC2018.processed, EPA.PM25.NC2019.processed)

#8

EPA.O3.PM25.2018.2019.subset <-
  EPA.O3.PM25.2018.2019 %>%
  filter(Site.Name %in% c("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")) %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlatitude = mean(SITE_LATITUDE),
            meanlongitude = mean(SITE_LONGITUDE)) %>%
  separate(Date, c("Year", "Month", "Day"), sep = "-")
```

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

```
#9

# spreading dataset with PM2.5 and Ozone columns
EPA.Air.subset.spread <- pivot_wider(EPA.O3.PM25.2018.2019.subset,
                                     names_from = AQS_PARAMETER_DESC, values_from = meanAQI)

#10

dim(EPA.Air.subset.spread)
```

```
## [1] 8246    9
```

```
#11
```

```
write.csv(EPA.Air.subset.spread, row.names = FALSE,
          "../Data/Processed/EPAair_O3_PM25_NC2122_Processed.csv")
```

Generate summary tables

12a. Use the split-apply-combine strategy to generate a summary data frame from your results from Step 9 above. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group.

12b. BONUS: Add a piped statement to 12a that removes rows where both mean ozone and mean PM2.5 have missing values.

13. Call up the dimensions of the summary dataset.

```
#12(a,b)

EPA.Air.summaries <-
  EPA.Air.subset.spread %>%
  group_by(Site.Name, Year, Month) %>%
  summarise(meanOzone = mean(Ozone),
            meanPM25 = mean(`PM 2.5`)) %>%
  filter(!is.na(meanOzone) | !is.na(meanPM25))
```

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

```
#13

dim(EPA.Air.summaries)
```

```
## [1] 268   5
```

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

    Answer: The `na.omt` function deletes any row that has even just one NA in any column which is not the output we are looking for. On the other hand, the `drop_na` function allows you to delete rows based on them havign NA's in specific columns. In this case, to remove jusr rows with NA's in both mean columns we use `is.na` within the `filter` function in order to use the OR operator to remove only rows with NA's in both columns while keeping rows that have data in at least one of these two columns.