Assignment 4: Data Wrangling

Julia Weinberg

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/juliaweinberg/Desktop/github repos/Environmental_Data_Analytics_2022"
```

```
#install.packages(tidyverse)
library(tidyverse)
#install.packages(lubridate)
library(lubridate)

EPAair2018.O3 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv", stringsAsFactors = TRUE)
#upload EPAair2018

EPAair2019.O3 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv", stringsAsFactors = TRUE)
#upload EPAair2018

EPAair2018.PM25 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
#upload EPAair2018

EPAair2019.PM25 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)
#upload EPAair2018

#2
dim(EPAair2018.O3) #get dimensions of EPAair2018.O3
```

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

```
colnames(EPAair2018.03) #get columns of EPAair2018.03
```

```
##  [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"
```

```
class(EPAair2018.03) #get structure of EPAair2018.03
```

```
## [1] "data.frame"
```

```
dim(EPAair2019.03) #get dimensions of EPAair2019.03
```

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

```
colnames(EPAair2019.03) #get columns of EPAair2019.03
```

```
##  [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"
```

```
class(EPAair2019.O3) #get structure of EPAair2019.O3
```

```
## [1] "data.frame"
```

```
dim(EPAair2018.PM25) #get dimensions of EPAair2018.PM25
```

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

```
colnames(EPAair2018.PM25) #get columns of EPAair2018.PM25
```

```
##  [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"
```

```
class(EPAair2018.PM25) #get structure of EPAair2018.PM25
```

```
## [1] "data.frame"
```

```
dim(EPAair2019.PM25) #get dimensions of EPAair2019.PM25
```

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

```
colnames(EPAair2019.PM25) #get columns of EPAair2019.PM25
```

```
##  [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"
```

```
class(EPAair2019.PM25) #get structure of EPAair2019.PM25
```

```
## [1] "data.frame"
```

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
EPAair2018.O3$Date <- as.Date(EPAair2018.O3$Date, format = "%m/%d/%Y")
#change EPAair2018.O3 to date format
```

```r
EPAair2019.O3$Date <- as.Date(EPAair2019.O3$Date, format = "%m/%d/%Y")
#change EPAair2019.O3 to date format

EPAair2018.PM25$Date <- as.Date(EPAair2018.PM25$Date, format = "%m/%d/%Y")
#change EPAair2018.PM25 to date format

EPAair2019.PM25$Date <- as.Date(EPAair2019.PM25$Date, format = "%m/%d/%Y")
#change EPAair2019.PM25 to date format

#4
EPAair2018.O3.select <- EPAair2018.O3 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)
#select columns for EPAair2018.O3

EPAair2019.O3.select <- EPAair2019.O3 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)
#select columns for EPAair2019.O3

EPAair2018.PM25.select <- EPAair2018.PM25 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)
#select columns for EPAair2018.PM25

EPAair2019.PM25.select <- EPAair2019.PM25 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)
#select columns for EPAair2019.PM25

#5


EPAair2018.PM25.select$AQS_PARAMETER_DESC <- "PM2.5"
#fill AQS_PARAMETER_DESC with PM2.5

EPAair2019.PM25.select$AQS_PARAMETER_DESC <- "PM2.5"
#fill AQS_PARAMETER_DESC with PM2.5


#6

# Save processed  file
write.csv(EPAair2018.O3.select, row.names = FALSE,
          file = "./Data/Processed/EPAair2018.O3.select_Processed.csv")
#save processed file

write.csv(EPAair2019.O3.select, row.names = FALSE,
          file = "./Data/Processed/EPAair2019.O3.select_Processed.csv")
#save processed file

write.csv(EPAair2018.PM25.select, row.names = FALSE,
          file = "./Data/Processed/EPAair2018.PM25.select_Processed.csv")
```

```
#save processed file

write.csv(EPAair2019.PM25.select, row.names = FALSE,
          file = "./Data/Processed/EPAair2019.PM25.select_Processed.csv")
#save processed file
```

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.Data.Total <- rbind(EPAair2018.O3.select, EPAair2018.PM25.select,
                        EPAair2019.O3.select, EPAair2019.PM25.select)
#bind data sets into one

#8

EPA.Data.Wrangle <- EPA.Data.Total %>%
  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")) %>%
  #filter by Site.Name
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>% #group by columns
  summarise(meanAQI = mean(DAILY_AQI_VALUE), #get mean of Daily_Aqi_Value
            meanLatitude = mean(SITE_LATITUDE), #get mean of site_longitude
            meanLongitude = mean(SITE_LONGITUDE))%>% #get mean of site_longitude
  mutate(Month=month(Date), Year=year(Date))
```

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

```
#replace date column with month and year

dim(EPA.Data.Wrangle) #get dimensions of data
```

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

```
#9

EPA.Data.Spread <- pivot_wider(EPA.Data.Wrangle,
names_from = AQS_PARAMETER_DESC, values_from = meanAQI)
#make columns of PM2.5 and Ozone values

#10

dim(EPA.Data.Spread) #get dimensions of data
```

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

```
#11

write.csv(EPA.Data.Spread, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_PM25_NC2122_Processed.csv")
#save processed data
```

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)
#a
EPA.Data.Summary_Final <-
  EPA.Data.Spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(meanOzone = mean(Ozone), meanPM2.5 = mean(PM2.5)) %>%
            filter(!is.na(meanPM2.5) | !is.na(meanOzone)) #remove NA from data
```

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

```
#13
dim(EPA.Data.Summary_Final) #get dimensions of data
```

```
## [1] 292   5
```

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

    Answer: Drop_na removes all the rows with NA in them while na.omit allows you to perform calculations while omiting any NA values.