Assignment 4: Data Wrangling

Jess Garcia

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 Tuesday, Feb 16 @ 11:59pm.

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
#Check working directory & load packages
getwd()
```

```
## [1] "C:/Users/93jes/Documents/ENV872/Environmental_Data_Analytics_2021"
```

```
library(tidyverse)
library(lubridate)

#Upload & rename four raw data files
EPAair.O3.2018.data <- read.csv("C:/Users/93jes/Documents/ENV872/Environmental_Data_Analytics_2021/Data/

EPAair.O3.2019.data <- read.csv("C:/Users/93jes/Documents/ENV872/Environmental_Data_Analytics_2021/Data/

EPAair.PM25.2018.data <- read.csv("C:/Users/93jes/Documents/ENV872/Environmental_Data_Analytics_2021/Da

EPAair.PM25.2019.data <-read.csv("C:/Users/93jes/Documents/ENV872/Environmental_Data_Analytics_2021/Data


#2. Explore dimensions, column names, & structure of the datasets

dim(EPAair.O3.2018.data)
```

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

```
dim(EPAair.O3.2019.data)
```

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

```
dim(EPAair.PM25.2018.data)
```

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

```
dim(EPAair.PM25.2019.data)
```

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

```
colnames(EPAair.O3.2018.data)
```

```
##  [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(EPAair.O3.2019.data)
```

```
##  [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(EPAair.PM25.2018.data)
```

```
##  [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(EPAair.PM25.2019.data)
```

```
##  [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.O3.2018.data)
```

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

```
str(EPAair.O3.2019.data)
```

```
## '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 3700
```

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

```
str(EPAair.PM25.2018.data)
```

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

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

```
## '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. Change date to date

#EPAair.O3.2018.data
#EPAair.O3.2019.data
#EPAair.PM25.2018.data
#EPAair.PM25.2019.data

#Confirm class of date column is factor and not date variable
class(EPAair.O3.2018.data$Date)
```

```
## [1] "factor"
```

```
class(EPAair.O3.2019.data$Date)
```

```
## [1] "factor"
```

```
class(EPAair.PM25.2018.data$Date)
```

```
## [1] "factor"
```

```
class(EPAair.PM25.2019.data$Date)
```

```
## [1] "factor"
```

```
#Change date to date
EPAair.O3.2018.data$Date <-as.Date(EPAair.O3.2018.data$Date,
                                   format = "%m/%d/%Y")
EPAair.O3.2019.data$Date <-as.Date(EPAair.O3.2019.data$Date,
                                   format = "%m/%d/%Y")
EPAair.PM25.2018.data$Date <-as.Date(EPAair.PM25.2018.data$Date,
                                     format = "%m/%d/%Y")
EPAair.PM25.2019.data$Date <-as.Date(EPAair.PM25.2019.data$Date,
                                     format = "%m/%d/%Y")


#4. Select columns

EPAair.O3.2018.data.selected <- select(EPAair.O3.2018.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAM
```

```
EPAair.O3.2019.data.selected <- select(EPAair.O3.2019.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAM

EPAair.PM25.2018.data.selected <- select(EPAair.PM25.2018.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_P

EPAair.PM25.2019.data.selected <- select(EPAair.PM25.2019.data, Date, DAILY_AQI_VALUE, Site.Name, AQS_P

#5. For PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5"

#view(EPAair.PM25.2018.data)
vignette("dplyr")
```

```
## starting httpd help server ... done
```

```
class(EPAair.PM25.2018.data.selected$AQS_PARAMETER_DESC)
```

```
## [1] "factor"
```

```
EPAair.PM25.2018.data.selected2 <-EPAair.PM25.2018.data.selected%>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

EPAair.PM25.2019.data.selected2 <-EPAair.PM25.2018.data.selected%>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")


#6. Save processed datasets
write.csv(EPAair.O3.2018.data.selected, row.names = FALSE, file = "./Data/Processed/EPAair.O3.2018.data

write.csv(EPAair.O3.2019.data.selected, row.names = FALSE, file = "./Data/Processed/EPAair.O3.2019.data

write.csv(EPAair.PM25.2018.data.selected2, row.names = FALSE, file = "./Data/Processed/EPAair.PM25.2018

write.csv(EPAair.PM25.2019.data.selected2, row.names = FALSE, file = "./Data/Processed/EPAair.PM25.2019
```

Combine datasets

7. Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code. ##use the datasets from #4 before you changed them
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) #Don't have to use intersect for this, just use filter
- 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 (just the one) 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. Combine four datasets with rbind
#Confirm column names are identical
#summary(EPAair.O3.2018.data.selected$Site.Name)
#summary(EPAair.O3.2019.data.selected$Site.Name)
#summary(EPAair.PM25.2018.data.selected$Site.Name)
#summary(EPAair.PM25.2019.data.selected2$Site.Name)


#Combine with rbind
EPAair_O3_PM25_NC1819_tight <- rbind(EPAair.O3.2018.data.selected,
    EPAair.O3.2019.data.selected,
    EPAair.PM25.2018.data.selected2,
    EPAair.PM25.2019.data.selected2)


#8. Wrangle new dataset
#view(EPAair_O3_PM25_NC1819_tight)
class(EPAair_O3_PM25_NC1819_tight$Date)
```

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

```
EPAair_O3_PM25_NC1819_tight.wrangled <- EPAair_O3_PM25_NC1819_tight %>%
  filter(Site.Name == "Linville Falls" | Site.Name == "Durham Armory"| Site.Name == "Leggett" | Site.Nam
  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))%>%
  mutate(Year = year(Date))
```

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

```
#Confirm interim dimensions of dataset are: 14,752 x 9
dim(EPAair_O3_PM25_NC1819_tight.wrangled)
```

```
## [1] 10860     9
```

```
#Dimensions don't match. I have less rows than I'm supposed to, but not sure why. I can't figure it out


#9. Spread aka pivot_wider the dataset
EPAair_O3_PM25_NC1819_wide <- pivot_wider(EPAair_O3_PM25_NC1819_tight.wrangled, names_from = AQS_PARAMET
#Somehow DAILY_AQI_VALUE disappeared as a column when I wrangled the data in question 8, so I guessed I


#10. Dimensions of dataset
dim(EPAair_O3_PM25_NC1819_wide)
```

```
## [1] 8029    9
```

```
#11. Save new combined, tidy, processed dataset
write.csv(EPAair_O3_PM25_NC1819_wide, row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM25_NC1819_
```

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/b. Generate a summary data frame
summary.EPAair_O3_PM25_NC1819_wide <-(EPAair_O3_PM25_NC1819_wide)%>%
  group_by (Site.Name, Month, Year)%>%
  summarise(mean.ozone = mean(Ozone),
            mean.PM2.5 = mean(PM2.5))%>%
  drop_na(Month, Year)
```

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

```
#Try with na.omit instead of drop_na
summary2.EPAair_O3_PM25_NC1819_wide <-(EPAair_O3_PM25_NC1819_wide)%>%
  group_by (Site.Name, Month, Year)%>%
  summarise(mean.ozone = mean(Ozone),
            mean.PM2.5 = mean(PM2.5))%>%
  na.omit(Month, Year)
```

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

```
#13
dim(summary.EPAair_O3_PM25_NC1819_wide)
```

```
## [1] 283   5
```

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

    Answer: We used drop_na rather than na.omit because na.omit will actually remove any rows that contain NA (in month or year) from the dataset completely. Whereas drop_na will just omit or exclude NA from your values but keep the rows/observations that were excluded for having NAs.