Assignment 4: Data Wrangling

Anne Harshbarger

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
getwd() #check working directory
```

```
## [1] "C:/Users/Anne/Documents/ENV872/Environmental_Data_Analytics_2021/Assignments"
```

```
#install.packages("tidyverse", "lubridate") #install packages if needed
library(tidyverse, lubridate) #load packages
#Create 4 dataframes from raw csvs
EPAair_O3_NC2018 <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv",
                             stringsAsFactors = TRUE)
EPAair_O3_NC2019 <- read.csv("../Data/Raw/EPAair_O3_NC2019_raw.csv",
                             stringsAsFactors = TRUE)
EPAair_PM25_NC2018 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv",
                               stringsAsFactors = TRUE)
EPAair_PM25_NC2019 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv",
                               stringsAsFactors = TRUE)
```

```
#2
#Check dimensions, column names, and structure
dim(EPAair_O3_NC2018); colnames(EPAair_O3_NC2018)
```

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

```
##  [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"
```

```
str(EPAair_O3_NC2018, width=80, strict.width="cut")
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","0"..
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 ..
##  $ Site.ID                             : int  370030005 370030005 370030005 37..
##  $ 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...
##  $ UNITS                               : Factor w/ 1 level "ppm": 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",..:..
##  $ 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 ..
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44..
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 ..
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25..
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, N"..
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1..
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Ave"..
##  $ 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(EPAair_O3_NC2019); colnames(EPAair_O3_NC2019)
```

```
## [1] 10592    20

##  [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"
```

```
str(EPAair_O3_NC2019, width=80, strict.width="cut")
```

```
## ’data.frame':    10592 obs. of  20 variables:
##  $ Date                                 : Factor w/ 365 levels "01/01/2019","0"..
##  $ Source                               : Factor w/ 2 levels "AirNow","AQS": 1 ..
##  $ Site.ID                              : int  370030005 370030005 370030005 37..
##  $ 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...
##  $ UNITS                                : Factor w/ 1 level "ppm": 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",..:..
##  $ 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 ..
##  $ AQS_PARAMETER_CODE                   : int  44201 44201 44201 44201 44201 44..
##  $ AQS_PARAMETER_DESC                   : Factor w/ 1 level "Ozone": 1 1 1 1 1 ..
##  $ CBSA_CODE                            : int  25860 25860 25860 25860 25860 25..
##  $ CBSA_NAME                            : Factor w/ 15 levels "","Asheville, N"..
##  $ STATE_CODE                           : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                                : Factor w/ 1 level "North Carolina": 1..
##  $ COUNTY_CODE                          : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                               : Factor w/ 30 levels "Alexander","Ave"..
##  $ 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(EPAair_PM25_NC2018); colnames(EPAair_PM25_NC2018)
```

```
## [1] 8983   20
##  [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_PM25_NC2018, width=80, strict.width="cut")
```

```
## ’data.frame’:    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2"..
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 ..
##  $ Site.ID                       : int  370110002 370110002 370110002 37011000..
##  $ 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...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 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 ..
##  $ 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 10..
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88..
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI &"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1..
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: ..
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
dim(EPAair_PM25_NC2019); colnames(EPAair_PM25_NC2019)
```

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

```
##  [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_PM25_NC2019, width=80, strict.width="cut")
```

```
## ’data.frame’:    8581 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2019","01/02/2"..
##  $ Source                        : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 ..
##  $ Site.ID                       : int  370110002 370110002 370110002 37011000..
##  $ 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..
##  $ DAILY_AQI_VALUE               : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Board Of Ed. Bldg."..
##  $ 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 10..
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88..
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI &"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1..
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: ..
##  $ 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 column to class date for all 4 dataframes
EPAair_O3_NC2018$Date <- as.Date(EPAair_O3_NC2018$Date, format="%m/%d/%Y")
EPAair_O3_NC2019$Date <- as.Date(EPAair_O3_NC2019$Date, format="%m/%d/%Y")
EPAair_PM25_NC2018$Date <- as.Date(EPAair_PM25_NC2018$Date, format="%m/%d/%Y")
EPAair_PM25_NC2019$Date <- as.Date(EPAair_PM25_NC2019$Date, format="%m/%d/%Y")

#4
#Select desired columns for all 4 dataframes
EPAair_O3_NC2018 <- select(EPAair_O3_NC2018, Date, DAILY_AQI_VALUE,
                           Site.Name, AQS_PARAMETER_DESC, COUNTY,
                           SITE_LATITUDE, SITE_LONGITUDE)
EPAair_O3_NC2019 <- select(EPAair_O3_NC2019, Date, DAILY_AQI_VALUE,
                           Site.Name, AQS_PARAMETER_DESC, COUNTY,
                           SITE_LATITUDE, SITE_LONGITUDE)
EPAair_PM25_NC2018 <-select(EPAair_PM25_NC2018, Date, DAILY_AQI_VALUE,
                            Site.Name, AQS_PARAMETER_DESC, COUNTY,
                            SITE_LATITUDE, SITE_LONGITUDE)
```

```r
EPAair_PM25_NC2019 <- select(EPAair_PM25_NC2019, Date, DAILY_AQI_VALUE,
                             Site.Name, AQS_PARAMETER_DESC, COUNTY,
                             SITE_LATITUDE, SITE_LONGITUDE)
#5
#Replace parameter value with "PM2.5" for all cells
EPAair_PM25_NC2018$AQS_PARAMETER_DESC <- "PM2.5"
EPAair_PM25_NC2019$AQS_PARAMETER_DESC <- "PM2.5"

#6
#Create csvs for processed datasets
write.csv(EPAair_O3_NC2018, row.names = FALSE,
          file = "../Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(EPAair_O3_NC2019, row.names = FALSE,
          file = "../Data/Processed/EPAair_O3_NC2019_processed.csv")
write.csv(EPAair_PM25_NC2018, row.names = FALSE,
          file = "../Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(EPAair_PM25_NC2019, row.names = FALSE,
          file = "../Data/Processed/EPAair_PM25_NC2019_processed.csv")
```

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”

```r
#7
#Create combined dataset
EPAair_combined <- rbind(EPAair_O3_NC2018, EPAair_O3_NC2019,
                         EPAair_PM25_NC2018, EPAair_PM25_NC2019)

#8
#test intersect function
# common_sites_O3 <- intersect(EPAair_O3_NC2018$Site.Name, EPAair_O3_NC2019$Site.Name)
# common_sites_PM25 <- intersect(EPAair_PM25_NC2018$Site.Name, EPAair_PM25_NC2019$Site.Name)
# common_sites_both <- intersect(common_sites_O3, common_sites_PM25)

library(lubridate) #Load again to avoid error with knit output file
```

```
##
## Attaching package: ’lubridate’

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
EPAair_tidy <-
  EPAair_combined %>%
  filter(Site.Name == "Linville Falls" | #filter intersecting sites
        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) %>% #Split-apply-combine
  summarise(DAILY_AQI_VALUE = mean(DAILY_AQI_VALUE),
           SITE_LATITUDE = mean(SITE_LATITUDE),
           SITE_LONGITUDE = mean(SITE_LONGITUDE)) %>%
  mutate(Month = month(Date)) %>% #create Month column
  mutate(Year = year(Date)) #create Year column
```

```
## ‘summarise()‘ regrouping output by ’Date’, ’Site.Name’, ’AQS_PARAMETER_DESC’ (override with ‘.groups
```

```
#9
#Separate ozone and PM2.5 columns
EPAair_tidy <- pivot_wider(EPAair_tidy,
                           names_from = AQS_PARAMETER_DESC,
                           values_from = DAILY_AQI_VALUE)

#10
dim(EPAair_tidy) #check dimensions
```

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

```
#11
#save combined processed dataset
write.csv(EPAair_tidy, row.names = FALSE,
          file = "../Data/Processed/EPAair_O3_PM25_NC1718_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_summary <-
  EPAair_tidy %>%
  group_by(Site.Name, Month, Year) %>% #create groups by site, month, year
  summarise(meanAQI_O3 = mean(Ozone),
            meanAQI_PM25 = mean(PM2.5)) #calculate means
```

```
## `summarise()` regrouping output by 'Site.Name', 'Month' (override with `.groups` argument)
```

```
#12b
EPAair_summary2 <-
  EPAair_summary %>%
  drop_na(Month) %>%
  drop_na(Year)

#test effects of na.omit
# EPAair_summary3 <-
#   EPAair_summary %>%
#   na.omit(Month) %>%
#   na.omit(Year)

#compare to drop_na without specifying columns
#EPAair_summary4 <-
#   EPAair_summary %>%
#   drop_na()

#13
  dim(EPAair_summary2) #check dimensions
```

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

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

> Answer: na.omit cannot operate on just one column at a time, but drop_na can. Although I attempted to specify the Month and Year columns with na.omit, na.omit removed rows with an NA in any column; the dimensions of the dataset after running the pipe with na.omit were the same as if I had used drop_na() with no columns specified (101 rows and 5 columns, as opposed to 308 rows and 5 columns with drop_na where columns Month and Year are specified). Meanwhile, drop_na would only have removed rows where the month or year was NA. The month and year were complete for all rows, so no rows were removed from the summary dataset.