Assignment 4: Data Wrangling

Jonathan Joyner

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A04_DataWrangling.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. Ensure that code in code chunks does not extend off the page in the PDF.

The completed exercise is due on Thursday, Sept 28th @ 5:00pm.

Set up your session

1a. Load the `tidyverse`, `lubridate`, and `here` packages into your session.

1b. Check your working directory.

1c. Read in all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Apply the `glimpse()` function to reveal the dimensions, column names, and structure of each dataset.

```r
#1a load lubridate and tidyverse
library(tidyverse)
library(lubridate)
#1b check working directory
getwd()
```

```
## [1] "C:/Users/jbjoy/OneDrive/Documents/Grad School/Fall 2023/Environ 872/EDE_Fall2023"
```

```r
setwd("C:/Users/jbjoy/OneDrive/Documents/Grad School/Fall 2023/Environ 872/EDE_Fall2023")
#1c read in raw data files for EPA air
EPA1<-read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv",stringsAsFactors = TRUE)
EPA2<-read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv",stringsAsFactors = TRUE)
EPA3<-read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv",stringsAsFactors = TRUE)
```

```
EPA4<-read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv",stringsAsFactors = TRUE)
#2 Glimpse each of the 4 named datasets (EPA1,2,3,and 4)
glimpse(EPA1)
```

```
## Rows: 9,737
## Columns: 20
## $ Date                             <fct> 03/01/2018, 03/02/2018, 03/03/201~
## $ Source                           <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS~
## $ Site.ID                          <int> 370030005, 370030005, 370030005, ~
## $ POC                              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.043, 0.046, 0.047, 0.049, 0.047~
## $ UNITS                            <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                  <int> 40, 43, 44, 45, 44, 28, 33, 41, 4~
## $ Site.Name                        <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                  <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                 <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE               <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC               <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                        <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                        <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                       <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                            <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                      <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                           <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                    <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                   <dbl> -81.191, -81.191, -81.191, -81.19~
```

```
glimpse(EPA2)
```

```
## Rows: 10,592
## Columns: 20
## $ Date                             <fct> 01/01/2019, 01/02/2019, 01/03/201~
## $ Source                           <fct> AirNow, AirNow, AirNow, AirNow, A~
## $ Site.ID                          <int> 370030005, 370030005, 370030005, ~
## $ POC                              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.029, 0.018, 0.016, 0.022, 0.037~
## $ UNITS                            <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                  <int> 27, 17, 15, 20, 34, 34, 27, 35, 3~
## $ Site.Name                        <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                  <int> 24, 24, 24, 24, 24, 24, 24, 24, 2~
## $ PERCENT_COMPLETE                 <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE               <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC               <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                        <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                        <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                       <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                            <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                      <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                           <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                    <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                   <dbl> -81.191, -81.191, -81.191, -81.19~
```

```
glimpse(EPA3)
```

```
## Rows: 8,983
## Columns: 20
## $ Date                         <fct> 01/02/2018, 01/05/2018, 01/08/2018, 01/~
## $ Source                       <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                      <int> 370110002, 370110002, 370110002, 370110~
## $ POC                          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 2.9, 3.7, 5.3, 0.8, 2.5, 4.5, 1.8, 2.5,~
## $ UNITS                        <fct> ug/m3 LC, ug/m3 LC, ug/m3 LC, ug/m3 LC,~
## $ DAILY_AQI_VALUE              <int> 12, 15, 22, 3, 10, 19, 8, 10, 18, 7, 24~
## $ Site.Name                    <fct> Linville Falls, Linville Falls, Linvill~
## $ DAILY_OBS_COUNT              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE             <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE           <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC           <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                    <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                   <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                        <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                  <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                       <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE               <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

```
glimpse(EPA4)
```

```
## Rows: 8,581
## Columns: 20
## $ Date                         <fct> 01/03/2019, 01/06/2019, 01/09/2019, 01/~
## $ Source                       <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                      <int> 370110002, 370110002, 370110002, 370110~
## $ POC                          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 1.6, 1.0, 1.3, 6.3, 2.6, 1.2, 1.5, 1.5,~
## $ UNITS                        <fct> ug/m3 LC, ug/m3 LC, ug/m3 LC, ug/m3 LC,~
## $ DAILY_AQI_VALUE              <int> 7, 4, 5, 26, 11, 5, 6, 6, 15, 7, 14, 20~
## $ Site.Name                    <fct> Linville Falls, Linville Falls, Linvill~
## $ DAILY_OBS_COUNT              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE             <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE           <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC           <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                    <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                   <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                        <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                  <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                       <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE               <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

Wrangle individual datasets to create processed files.

3. Change the Date columns to be date objects.

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".

```r
#3 check class of each dataset
class(EPA1$Date)
```

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

```r
class(EPA2$Date)
```

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

```r
class(EPA3$Date)
```

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

```r
class(EPA4$Date)
```

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

```r
#change class of each to date objects
EPA1$Date <-as.Date(EPA1$Date, format = "%m/%d/%Y")
EPA2$Date <-as.Date(EPA2$Date, format = "%m/%d/%Y")
EPA3$Date <-as.Date(EPA3$Date, format = "%m/%d/%Y")
EPA4$Date <-as.Date(EPA4$Date, format = "%m/%d/%Y")
#check class and head of each dataset to confirm changes and ensure dates are displayed correctly
class(EPA1$Date)
```

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

```r
head(EPA1$Date)
```

```
## [1] "2018-03-01" "2018-03-02" "2018-03-03" "2018-03-04" "2018-03-05"
## [6] "2018-03-06"
```

```r
class(EPA2$Date)
```

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

```r
head(EPA2$Date)
```

```
## [1] "2019-01-01" "2019-01-02" "2019-01-03" "2019-01-04" "2019-01-05"
## [6] "2019-01-06"
```

```r
class(EPA3$Date)
```

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

```r
head(EPA3$Date)
```

```
## [1] "2018-01-02" "2018-01-05" "2018-01-08" "2018-01-11" "2018-01-14"
## [6] "2018-01-17"
```

```r
class(EPA4$Date)
```

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

```r
head(EPA4$Date)
```

```
## [1] "2019-01-03" "2019-01-06" "2019-01-09" "2019-01-12" "2019-01-15"
## [6] "2019-01-18"
```

```r
#4 select specific columns for each dataset
EPA1R<-select(EPA1, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_L
EPA2R<-select(EPA2, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_L
EPA3R<-select(EPA3, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_L
EPA4R<-select(EPA4, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_L
#5 use mutate function and pipes to change data in AQS_PARAMETER_DESC columns to PM2.5
EPA3R <-
  EPA3R %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
EPA4R <-
  EPA4R %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
#6 use write function to save processed datasets to Data/Processed file
write.csv(EPA1R, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(EPA2R, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")
write.csv(EPA3R, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(EPA4R, 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 only 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 - but it will include sites with missing site information, which you don't want. . .)

- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site name, 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_NC1819_Processed.csv"

```r
#7 check columns to ensure they match before binding
colnames(EPA1R)
```

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

```r
colnames(EPA2R)
```

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

```r
colnames(EPA3R)
```

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

```r
colnames(EPA4R)
```

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

```r
# NOT APPLICABLE: use rename function and pipes to change differing column title to a matching title
#EPA1R <- EPA1R %>%
            #rename(
            #Concentration = Daily.Max.8.hour.Ozone.Concentration
              #)
#EPA2R <- EPA2R %>%
            #rename(
            #Concentration = Daily.Max.8.hour.Ozone.Concentration
              #)
#EPA3R <- EPA3R %>%
            #rename(
            #Concentration = Daily.Mean.PM2.5.Concentration
```

```
                #)
#EPA4R <- EPA4R %>%
                #rename(
                #Concentration = Daily.Mean.PM2.5.Concentration
                  #)
# use rbind to combine datasets
EPA_Combined <- rbind(EPA1R,EPA2R,EPA3R,EPA4R)
#8 Filter common sites and group by date, site, AQS parameter, and county.
EPA_Combined_F1 <-
  EPA_Combined %>%
    filter(Site.Name == "Linville Falls" | Site.Name == "Durham Armory" | Site.Name == "Leggett"  | Site
    group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
    summarize(meanAQI=mean(DAILY_AQI_VALUE), #Create columns for mean of AQI, latitude and longitude
            meanLAT=mean(SITE_LATITUDE),
            meanLON=mean(SITE_LONGITUDE)) %>%
     mutate(Month=month(Date), #Create columns for month and year using lubridate
            Year=year(Date)
              )
```

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

```
#9 spread the mean AQI values into two columns which are PM2.5 and Ozone
EPA_Combined_Spread <- pivot_wider(EPA_Combined_F1,names_from="AQS_PARAMETER_DESC",values_from="meanAQI
#10 get dimensions of new dataset
dim(EPA_Combined_Spread)
```

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

```
#11 save as processed file
write.csv(EPA_Combined_Spread, row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM25_NC1819_Process
```

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 mean **ozone** values are not available (use the function `drop_na` in your pipe). It's ok to have missing mean PM2.5 values in this result.

13. Call up the dimensions of the summary dataset.

```
#12 Join the combined dataset and the spread dataset so the correct columns are available
EPA_Joined<-full_join(EPA_Combined_F1,EPA_Combined_Spread)
```

```
## Joining with 'by = join_by(Date, Site.Name, COUNTY, meanLAT, meanLON, Month,
## Year)'
```

```
# group the dataset by site, month, and year, get averages for the spread column values, and drop any N.
EPA_Final<-
  EPA_Joined %>%
```

```
  group_by(Site.Name,Month,Year) %>%
  summarize(MeanOZ=mean(Ozone),
            MeanPM=mean(PM2.5)) %>%
  drop_na(MeanOZ)
```

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

```
#13 check the dataset dimensions
dim(EPA_Final)
```

```
## [1] 182    5
```

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

Answer: drop_na works better than na.omit because it isolates the effect in one column. na.omit would drop NA from the entire dataset