Assignment 4: Data Wrangling

Ayoung Kim

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.

```
#1a #Load 'tidyverse', 'lubridate', and 'here' packages
install.packages("tidyverse")
install.packages("lubridate")
install.packages("here")

library(tidyverse)
library(lubridate)
library(here)

here()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
#1b Checking working directory
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
#1c Read 4 datasets
#Ozone, NC, 2018 data
Ozone.NC.2018 <- read.csv(
```

```
  file=here("./Data/Raw/EPAair_O3_NC2018_raw.csv"),
  stringsAsFactors = TRUE
)

# Ozone, NC, 2019 data
Ozone.NC.2019 <- read.csv(
  file=here("./Data/Raw/EPAair_O3_NC2019_raw.csv"),
  stringsAsFactors = TRUE
)

#PM2.5, NC, 2018 data
PM25.NC.2018 <- read.csv(
  file=here("./Data/Raw/EPAair_PM25_NC2018_raw.csv"),
  stringsAsFactors = TRUE
)

#PM2.5, NC, 2019 data
PM25.NC.2019 <- read.csv(
  file=here("./Data/Raw/EPAair_O3_NC2019_raw.csv"),
  stringsAsFactors = TRUE
)


#2 Using glimpse function to figure out the information for each of the dataset

glimpse(Ozone.NC.2018)
```

```
## 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(Ozone.NC.2019)
```

```
## 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(PM25.NC.2018)
```

```
## 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(PM25.NC.2019)
```

```
## 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~
```

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".

```
#3 Change the Date columns to date object (I used lubridate to change the date columns to date object)

# Ozone,NC, 2018
Ozone.NC.2018$Date <-mdy(Ozone.NC.2018$Date)
class(Ozone.NC.2018$Date)
```

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

```
# Ozone, NC, 2019
Ozone.NC.2019$Date<-mdy(Ozone.NC.2019$Date)
class(Ozone.NC.2019$Date)
```

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

```
#PM2.5, NC, 2018
PM25.NC.2018$Date<-mdy(PM25.NC.2018$Date)
class(PM25.NC.2018$Date)
```

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

```
#PM2.5, NC, 2019
PM25.NC.2019$Date<-mdy(PM25.NC.2019$Date)
class(PM25.NC.2019$Date)
```

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

```
#4 Select specific columns (Used "select" function to select specific columns)

#Ozone, NC, 2018
Ozone.NC.2018<-select(Ozone.NC.2018,Date,DAILY_AQI_VALUE,Site.Name,AQS_PARAMETER_DESC,COUNTY,SITE_LATITU
```

```
#Ozone, NC, 2019
Ozone.NC.2019<-select(Ozone.NC.2019,Date,DAILY_AQI_VALUE,Site.Name,AQS_PARAMETER_DESC,COUNTY,SITE_LATITU

#PM2.5, NC, 2018
PM25.NC.2018<-select(PM25.NC.2018,Date,DAILY_AQI_VALUE,Site.Name,AQS_PARAMETER_DESC,COUNTY,SITE_LATITUDE

#PM2.5, NC, 2019
PM25.NC.2019<-select(PM25.NC.2019,Date,DAILY_AQI_VALUE,Site.Name,AQS_PARAMETER_DESC,COUNTY,SITE_LATITUDE

#5 Mutate
#(Used 'mutate' function to fill all cells in AQS_PARAMETER_DESC column to "PM 2.5")
#PM2.5,NC,2018
PM25.NC.2018<-PM25.NC.2018 %>%
  mutate(AQS_PARAMETER_DESC="PM 2.5")

#PM2.5,NC,2019
PM25.NC.2019<-PM25.NC.2019 %>%
  mutate(AQS_PARAMETER_DESC="PM 2.5")

#6 Saving processed datasets to 'Processed' file
#Ozone,NC,18
write.csv(Ozone.NC.2018, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")

#Ozone,NC,19
write.csv(Ozone.NC.2019, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")

#PM 2.5,NC,2018
write.csv(PM25.NC.2018, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")

#PM 2.5,NC,2019
write.csv(PM25.NC.2019, 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"

```
#7 Combine four datasets with 'rbind' function
rbind.Ozone.NC.PM25 <- rbind(Ozone.NC.2018,Ozone.NC.2019,PM25.NC.2018,PM25.NC.2019)


#8 Wrangle my new dataset
#I included only sites that the four data frames have in common.
#I found the mean of the AQI value, latitude, and longitude.
#I grouped by date, site name, AQS parameter, and county, and add columns for "Month" and Year" using
#Date column.

Ozone.NC.PM25.all<-
  rbind.Ozone.NC.PM25 %>%
  filter(Site.Name %in% c("Linville Falls","Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
  group_by(Date,Site.Name,AQS_PARAMETER_DESC,COUNTY) %>%
  summarise(AQImean = mean(DAILY_AQI_VALUE),
            Lat.mean = mean(SITE_LATITUDE),
            Long.mean = mean(SITE_LONGITUDE))%>%
  mutate(Month = month(Date), Year=year(Date))
```

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

```
#9 Spreading my datasets
#I used pivot_wider to spread my datasets to make AQI values for ozone and PM2.5 are
#in separate columns.
Ozone.NC.PM25.AQI<-pivot_wider(Ozone.NC.PM25.all,names_from = AQS_PARAMETER_DESC, values_from = AQImean

#10 Dimensions of my new dataset (I used dim function)
dim(Ozone.NC.PM25.AQI)
```

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

```
#11 Saving the processed dataset
write.csv(Ozone.NC.PM25.AQI, row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM25_NC1819_Processe
```

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 Summary Data Frame
# I grouped by site, month, and year.
#Also, I generated the mean AQI values for ozone and PM.2.5 then
#removed the ozone values with NA.

Ozone.NC.PM25.AQI.summary<-
  Ozone.NC.PM25.AQI %>%
  group_by(Site.Name,Month,Year) %>%
  summarise(meanozone=mean(Ozone),
            meanPM25=mean('PM2.5'))%>%
  drop_na(meanozone)
```

```
## Warning: There were 283 warnings in `summarise()`.
## The first warning was:
## i In argument: `meanPM25 = mean("PM2.5")`.
## i In group 1: `Site.Name = Bryson City`, `Month = 1`, `Year = 2018`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 282 remaining warnings.
```

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

```
  Ozone.NC.PM25.AQI.summaryomit<-
  Ozone.NC.PM25.AQI %>%
  group_by(Site.Name,Month,Year) %>%
  summarise(meanozone=mean(Ozone),
            meanPM25=mean('PM2.5'))%>%
  na.omit(meanozone)
```

```
## Warning: There were 283 warnings in `summarise()`.
## The first warning was:
## i In argument: `meanPM25 = mean("PM2.5")`.
## i In group 1: `Site.Name = Bryson City`, `Month = 1`, `Year = 2018`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 282 remaining warnings.
```

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

```
#13 Dimension of Ozone.NC.PM25.AQI.Summary
# I used dim to get dimension of the new dataset
dim(Ozone.NC.PM25.AQI.summary)
```

```
## [1] 205   5
```

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

    Answer: We used drop_na because it is useful to remove only rows where meanozone is NA. When I tried with na.omit, I found that all data has been missing.