Assignment 4: Data Wrangling

Siyu Dong

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.

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).

```
#Load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2024
```

```
#Check WD
getwd()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

```
#Read all the EPA Air Datasets
NC_O3_2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv", stringsAsFactors = TRUE)
NC_O3_2019 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv", stringsAsFactors = TRUE)
NC_PM25_2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
NC_PM25_2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)
```

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

```
glimpse(NC_O3_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(NC_O3_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(NC_PM25_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(NC_PM25_2019)
```

```
## 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.

```
NC_O3_2018$Date <- as.Date(NC_O3_2018$Date, format = "%m/%d/%Y")
NC_O3_2019$Date <- as.Date(NC_O3_2019$Date, format = "%m/%d/%Y")
NC_PM25_2018$Date <- as.Date(NC_PM25_2018$Date, format = "%m/%d/%Y")
NC_PM25_2019$Date <- as.Date(NC_PM25_2019$Date, format = "%m/%d/%Y")

class(NC_O3_2018$Date)
```

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

```
class(NC_O3_2019$Date)
```

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

```
class(NC_PM25_2018$Date)
```

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

```
class(NC_PM25_2019$Date)
```

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

4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE

```
#For NC_O3_2018
NC_O3_2018_Selected <- NC_O3_2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#For NC_O3_2018
NC_O3_2019_Selected <- NC_O3_2019 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#For NC_PM25_2018
NC_PM25_2018_Selected <- NC_PM25_2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#For NC_PM25_2019
NC_PM25_2019_Selected <- NC_PM25_2019 %>%
  select(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).

```
NC_PM25_2018_Selected <- NC_PM25_2018_Selected %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
head(NC_PM25_2018_Selected$AQS_PARAMETER_DESC, 10) #To display the first ten values
```

```
##  [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

```
NC_PM25_2019_Selected <- NC_PM25_2019_Selected %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
head(NC_PM25_2019_Selected$AQS_PARAMETER_DESC, 10) #To display the first ten values
```

```
##  [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace "raw" with "processed".

```
write.csv(NC_O3_2018_Selected, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_NC2018_processed.csv")

write.csv(NC_O3_2019_Selected, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_NC2019_processed.csv")

write.csv(NC_PM25_2018_Selected, row.names = FALSE,
          file = "./Data/Processed/EPAair_PM25_NC2018_processed.csv")

write.csv(NC_PM25_2019_Selected, 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.

```
#Check the identity of column names
#Combine datasets
NC_Air <- rbind(NC_O3_2018_Selected,
                NC_O3_2019_Selected,
                NC_PM25_2018_Selected,
                NC_PM25_2019_Selected)
dim(NC_Air) #To show the total variable amount equals to the sum of the four selected datasets'
```

```
## [1] 37893     7
```

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"

```
#8
NC_Air_SiteSelected <-
  NC_Air %>%
  drop_na(Site.Name) %>%
  filter(Site.Name == "Linville Falls" | 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")

NC_Air_SiteMeans <-
  NC_Air_SiteSelected %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(mean_AQI = mean(DAILY_AQI_VALUE),
            mean_Latitude = mean(SITE_LATITUDE),
            mean_Longtitude = mean(SITE_LONGITUDE))
```

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

```
head(NC_Air_SiteMeans, 5) #To display the first 5 obs of the df
```

```
## # A tibble: 5 x 7
## # Groups:   Date, Site.Name, AQS_PARAMETER_DESC [5]
##   Date       Site.Name          AQS_PARAMETER_DESC COUNTY mean_AQI mean_Latitude
##   <date>     <fct>              <fct>              <fct>     <dbl>         <dbl>
## 1 2018-01-01 Bryson City        PM2.5              Swain        35          35.4
## 2 2018-01-01 Castle Hayne       PM2.5              New H~       13          34.4
## 3 2018-01-01 Clemmons Middle    PM2.5              Forsy~       24          36.0
## 4 2018-01-01 Durham Armory      PM2.5              Durham       31          36.0
## 5 2018-01-01 Garinger High Sch~ Ozone              Meckl~       32          35.2
## # i 1 more variable: mean_Longtitude <dbl>
```

```
class(NC_Air_SiteMeans$Date) #Check the format of variable Date first
```

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

```
NC_Air_DateModified <-
  NC_Air_SiteMeans %>%
  mutate(Month = month(Date),
         Year = year(Date))
head(NC_Air_DateModified, 5) #To display the first 5 obs of the df
```

```
## # A tibble: 5 x 9
## # Groups:   Date, Site.Name, AQS_PARAMETER_DESC [5]
##   Date       Site.Name          AQS_PARAMETER_DESC COUNTY mean_AQI mean_Latitude
##   <date>     <fct>              <fct>              <fct>     <dbl>         <dbl>
## 1 2018-01-01 Bryson City        PM2.5              Swain        35          35.4
## 2 2018-01-01 Castle Hayne       PM2.5              New H~       13          34.4
## 3 2018-01-01 Clemmons Middle    PM2.5              Forsy~       24          36.0
## 4 2018-01-01 Durham Armory      PM2.5              Durham       31          36.0
## 5 2018-01-01 Garinger High Sch~ Ozone              Meckl~       32          35.2
## # i 3 more variables: mean_Longtitude <dbl>, Month <dbl>, Year <dbl>
```

```
#9
NC_Air_Spread <-
  NC_Air_DateModified %>%
  spread(key = AQS_PARAMETER_DESC, value = mean_AQI)
head(NC_Air_Spread, 5) #To display the first 5 obs of the df
```

```
## # A tibble: 5 x 9
## # Groups:   Date, Site.Name [5]
##   Date       Site.Name    COUNTY mean_Latitude mean_Longtitude Month  Year Ozone
##   <date>     <fct>        <fct>          <dbl>           <dbl> <dbl> <dbl> <dbl>
## 1 2018-01-01 Bryson City  Swain           35.4           -83.4     1  2018    NA
## 2 2018-01-01 Castle Hayne New H~          34.4           -77.8     1  2018    NA
## 3 2018-01-01 Clemmons Mi~ Forsy~          36.0           -80.3     1  2018    NA
## 4 2018-01-01 Durham Armo~ Durham          36.0           -78.9     1  2018    NA
## 5 2018-01-01 Garinger Hi~ Meckl~          35.2           -80.8     1  2018    32
## # i 1 more variable: PM2.5 <dbl>
```

```
#10
dim(NC_Air_Spread)
```

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

```
#11
write.csv(NC_Air_Spread, row.names = FALSE,
          file = "./Data/Processed/EPAair_O3_PM25_NC1819_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 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
NC_Air_Summary <-
  NC_Air_Spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(mean_OZONE = mean(Ozone),
            mean_PM25 = mean(PM2.5)) %>%
  drop_na(mean_OZONE)
```

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

```
head(NC_Air_Summary, 5) #To display the first 5 obs of the df
```

```
## # A tibble: 5 x 5
## # Groups:   Site.Name, Month [3]
##   Site.Name   Month  Year mean_OZONE mean_PM25
##   <fct>       <dbl> <dbl>      <dbl>     <dbl>
## 1 Bryson City     3  2018       41.6      34.7
## 2 Bryson City     3  2019       42.5      NA
## 3 Bryson City     4  2018       44.5      28.2
## 4 Bryson City     4  2019       45.4      26.7
## 5 Bryson City     5  2019       39.6      NA
```

```
NC_Air_Summary2 <-
  NC_Air_Spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(mean_OZONE = mean(Ozone),
            mean_PM25 = mean(PM2.5)) %>%
  na.omit(mean_OZONE)
```

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

```
head(NC_Air_Summary2, 5) #To display the first 5 obs of the df
```

```
## # A tibble: 5 x 5
## # Groups:   Site.Name, Month [4]
##   Site.Name   Month  Year mean_OZONE mean_PM25
##   <fct>       <dbl> <dbl>      <dbl>     <dbl>
## 1 Bryson City     3  2018       41.6      34.7
## 2 Bryson City     4  2018       44.5      28.2
## 3 Bryson City     4  2019       45.4      26.7
## 4 Bryson City     7  2019       30.4      33.6
## 5 Bryson City     9  2018       25.4      25.1
```

```r
#13
dim(NC_Air_Summary)
```

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

```r
dim(NC_Air_Summary2)
```

```
## [1] 101   5
```

14. Why did we use the function `drop_na` rather than `na.omit`? Hint: replace `drop_na` with `na.omit` in part 12 and observe what happens with the dimensions of the summary date frame.

    Answer: When I use na.omit, NA values in mean_PM25 are deleted as well. So `drop_na` only focuses on the target row and removing its missing values, while `na.omit` influences other rows as well.