Assignment 4: Data Wrangling (Fall 2024)

Aye Nyein Thu

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).

2. Add the appropriate code to reveal the dimensions of the four datasets.

```
#1a Loading packages
#install(tidyverse)
#install(lubridate)
#install(here)
library(tidyverse)
library(lubridate)
library(here)

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

```
## [1] "/home/guest/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
#1c Loading/ Reading the EPA Air datasets
EPAirOzone_2018 <- read.csv(
  file = here('./Data/Raw/EPAair_O3_NC2018_raw.csv'),
  stringsAsFactors = TRUE
)

EPAirOzone_2019 <- read.csv(
  file = here('./Data/Raw/EPAair_O3_NC2019_raw.csv'),
  stringsAsFactors = TRUE
)

EPAirPM2.5_2018 <- read.csv(
  file = here('./Data/Raw/EPAair_PM25_NC2018_raw.csv'),
  stringsAsFactors = TRUE
)

EPAirPM2.5_2019 <- read.csv (
  file = here('./Data/Raw/EPAair_PM25_NC2019_raw.csv'),
  stringsAsFactors = TRUE
)

#2 Dimensions of the EPA Air datasets
dim(EPAirOzone_2018)
```

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

```r
dim(EPAirOzone_2019)
```

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

```r
dim(EPAirPM2.5_2018)
```

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

```r
dim(EPAirPM2.5_2019)
```

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

```r
#2 Checking column names of the EPA Air datasets
colnames(EPAirOzone_2018)
```

```
##  [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(EPAirOzone_2019)
```

```
##  [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(EPAirPM2.5_2018)
```

```
##  [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(EPAirPM2.5_2019)
```

```
##  [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"
```

All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern?

> Answer: As checked in the above column names, all four EPA Air datasets have 20 variables or columns in total. The raw datasets titled as "EPAair_O3_NC2018_raw.csv" and "EPAair_O3_NC2019_raw.csv" have the same columns and different values in rows as both of them are measuring the concentration of ozone in 2018 and 2019 respectively. Similarly, the raw datasets of "EPAair_PM25_NC2018_raw.csv" and "EPAair_PM25_NC2019_raw.csv" have the same columns and unique rows as they are measuring the PM2.5 concentration in the air. Apart from this differences, the rest columns are the same in all four data sets.

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 Changing the Date columns to be date objects
class(EPAirOzone_2018$Date)
```

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

```
EPAirOzone_2018$Date <- mdy(EPAirOzone_2018$Date)
class(EPAirOzone_2018$Date)
```

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

```
class(EPAirOzone_2019$Date)
```

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

```
EPAirOzone_2019$Date <- mdy(EPAirOzone_2019$Date)
class(EPAirOzone_2019$Date)
```

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

```r
class(EPAirPM2.5_2018$Date)
```

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

```r
EPAirPM2.5_2018$Date <- mdy(EPAirPM2.5_2018$Date)
class(EPAirPM2.5_2018$Date)
```

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

```r
class(EPAirPM2.5_2019$Date)
```

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

```r
EPAirPM2.5_2019$Date <- mdy(EPAirPM2.5_2019$Date)
class(EPAirPM2.5_2019$Date)
```

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

```r
#4 Selecting columns
EPAirOzone_2018_7Col <- select(
  EPAirOzone_2018,
  Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
  COUNTY, SITE_LATITUDE, SITE_LONGITUDE
)

EPAirOzone_2019_7Col <- select(
  EPAirOzone_2019,
  Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
  COUNTY, SITE_LATITUDE, SITE_LONGITUDE
)

EPAirPM2.5_2018_7Col <- select(
  EPAirPM2.5_2018,
  Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
  COUNTY, SITE_LATITUDE, SITE_LONGITUDE
)

EPAirPM2.5_2019_7Col <- select(
  EPAirPM2.5_2019,
  Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
  COUNTY, SITE_LATITUDE, SITE_LONGITUDE
)

#5 Filling AQS_PARAMETER_DESC with PM2.5 in the PM2.5 datasets
EPAirPM2.5_2018_7Col <- mutate (
  EPAirPM2.5_2018_7Col, AQS_PARAMETER_DESC="PM2.5")

EPAirPM2.5_2019_7Col <- mutate (
  EPAirPM2.5_2019_7Col, AQS_PARAMETER_DESC="PM2.5")
```

```
#6 Saving processed datasets in the Processed folder
write.csv(
  EPAirOzone_2018_7Col,
  file=here('./Data/Processed/EPAair_O3_NC2018_processed.csv'),
  row.names = FALSE
)

write.csv(
  EPAirOzone_2019_7Col,
  file=here('./Data/Processed/EPAair_O3_NC2019_processed.csv'),
  row.names = FALSE
)

write.csv(
  EPAirPM2.5_2018_7Col,
  file=here('./Data/Processed/EPAair_PM25_NC2018_processed.csv'),
  row.names = FALSE
)

write.csv(
  EPAirPM2.5_2019_7Col,
  file=here('./Data/Processed/EPAair_PM25_NC2019_processed.csv'),
  row.names = FALSE
)
```

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 Step1: Checking the column names of the processed datasets
colnames(EPAirOzone_2018_7Col)
```

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

```
colnames(EPAirOzone_2019_7Col)
```

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

```
colnames(EPAirPM2.5_2018_7Col)
```

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

```
colnames(EPAirPM2.5_2019_7Col)
```

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

```
#7 Step2: Combining the four datasets into one
EPAir_Combined <- rbind(
  EPAirOzone_2018_7Col,EPAirOzone_2019_7Col,
  EPAirPM2.5_2018_7Col,EPAirPM2.5_2019_7Col)

#8 Step1: Checking the unique values/ site names of each dataset
unique(EPAirOzone_2018_7Col$Site.Name)
```

```
##  [1] Taylorsville Liledoun
##  [2] Linville Falls
##  [3] Cranberry
##  [4] Bent Creek
##  [5] Lenoir (city)
##  [6] Beaufort
##  [7] Cherry Grove
##  [8] Wade
##  [9] Honeycutt School
## [10] Durham Armory
## [11] Leggett
## [12] Hattie Avenue
## [13] Clemmons Middle
## [14] Union Cross
## [15] Joanna Bald
```

```
## [16] Butner
## [17] Mendenhall School
## [18] Waynesville School
## [19] Frying Pan Mountain
## [20] Purchase Knob
## [21] OZONE MONITOR ON SW SIDE OF TOWER/MET EQUIPMENT 10FT ABOVE TOWER
## [22] West Johnston Co.
## [23] Blackstone
## [24] Lenoir Co. Comm. Coll.
## [25] Crouse
## [26] Coweeta
## [27] Jamesville School
## [28] Garinger High School
## [29] University Meadows
## [30] Candor
## [31] Castle Hayne
## [32] Bushy Fork
## [33] Pitt Agri. Center
## [34] Bethany sch.
## [35] Rockwell
## [36] Bryson City
## [37]
## [38] Monroe School
## [39] Millbrook School
## [40] Mt. Mitchell
## 40 Levels:  Beaufort Bent Creek Bethany sch. Blackstone ... West Johnston Co.
```

```
unique(EPAirOzone_2019_7Col$Site.Name)
```

```
##  [1] Taylorsville Liledoun  Linville Falls         Cranberry
##  [4] Bent Creek             Lenoir (city)         Beaufort
##  [7] Cherry Grove           Wade                  Honeycutt School
## [10] Durham Armory          Leggett               Hattie Avenue
## [13] Clemmons Middle        Union Cross           Joanna Bald
## [16] Butner                 Mendenhall School     Waynesville School
## [19] Frying Pan Mountain    Purchase Knob         West Johnston Co.
## [22] Lenoir Co. Comm. Coll. Crouse                Coweeta
## [25] Jamesville School      Garinger High School  University Meadows
## [28] Candor                 Castle Hayne          Bushy Fork
## [31] Pitt Agri. Center      Bethany sch.          Rockwell
## [34] Bryson City                                  Monroe School
## [37] Millbrook School       Mt. Mitchell
## 38 Levels:  Beaufort Bent Creek Bethany sch. Bryson City Bushy Fork ... West Johnston Co.
```

```
unique(EPAirPM2.5_2018_7Col$Site.Name)
```

```
##  [1] Linville Falls
##  [2] Board Of Ed. Bldg.
##  [3] Hickory Water Tower
##  [4] William Owen School
##  [5] Lexington water tower
##  [6] Durham Armory
##  [7] Leggett
```

```
##  [8] Hattie Avenue
##  [9] Clemmons Middle
## [10] Mendenhall School
## [11] Frying Pan Mountain
## [12]
## [13] PM2.5 COLOCATED MONITORS LOCATED ON TOP OF BUILDING
## [14] West Johnston Co.
## [15] Blackstone
## [16] Garinger High School
## [17] Montclaire Elementary School
## [18] Remount
## [19] Spruce Pine Hospital
## [20] Candor: EPA CASTNet Site
## [21] Castle Hayne
## [22] Pitt Agri. Center
## [23] Bryson City
## [24] Millbrook School
## [25] Triple Oak
## 25 Levels:  Blackstone Board Of Ed. Bldg. ... William Owen School
```

```r
unique(EPAirPM2.5_2019_7Col$Site.Name)
```

```
##  [1] Linville Falls
##  [2] Board Of Ed. Bldg.
##  [3] Hickory Water Tower
##  [4] William Owen School
##  [5] Lexington water tower
##  [6] Durham Armory
##  [7] Leggett
##  [8] Hattie Avenue
##  [9] Clemmons Middle
## [10] Mendenhall School
## [11] Frying Pan Mountain
## [12]
## [13] PM2.5 COLOCATED MONITORS LOCATED ON TOP OF BUILDING
## [14] West Johnston Co.
## [15] Garinger High School
## [16] Montclaire Elementary School
## [17] Remount
## [18] Spruce Pine Hospital
## [19] Candor: EPA CASTNet Site
## [20] Castle Hayne
## [21] Northampton County
## [22] Pitt Agri. Center
## [23] Bryson City
## [24] Millbrook School
## [25] Triple Oak
## 25 Levels:  Board Of Ed. Bldg. Bryson City ... William Owen School
```

```r
#8 Step2: Removing blank and NA rows
EPAirOzone_2018_7Col_Site.Name <- EPAirOzone_2018_7Col %>%
  filter(Site.Name != "" & !is.na(Site.Name))
```

```r
EPAirOzone_2019_7Col_Site.Name <- EPAirOzone_2019_7Col %>%
  filter(Site.Name != "" & !is.na(Site.Name))

EPAirPM2.5_2018_7Col_Site.Name <- EPAirPM2.5_2018_7Col %>%
  filter(Site.Name != "" & !is.na(Site.Name))

EPAirPM2.5_2019_7Col_Site.Name <- EPAirPM2.5_2019_7Col %>%
  filter(Site.Name != "" & !is.na(Site.Name))

#8 Step3: Using intersect() function to find the common site names in all datasets
EPAir_CommonSiteName <- Reduce(
  intersect, list(EPAirOzone_2018_7Col_Site.Name$Site.Name,
                  EPAirOzone_2019_7Col_Site.Name$Site.Name,
                  EPAirPM2.5_2018_7Col_Site.Name$Site.Name,
                  EPAirPM2.5_2019_7Col_Site.Name$Site.Name)
)
print(EPAir_CommonSiteName)
```

```
##  [1] "Linville Falls"       "Durham Armory"        "Leggett"
##  [4] "Hattie Avenue"        "Clemmons Middle"      "Mendenhall School"
##  [7] "Frying Pan Mountain"  "West Johnston Co."    "Garinger High School"
## [10] "Castle Hayne"         "Pitt Agri. Center"    "Bryson City"
## [13] "Millbrook School"
```

```r
#8 Step4: Checking the class of Date column
class(EPAir_Combined$Date)
```

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

```r
#8 Step5: Data Wrangling
EPAir_Combined_Processed <- EPAir_Combined %>%
  filter(Site.Name %in% c("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")) %>%
  group_by(Date,Site.Name,AQS_PARAMETER_DESC,COUNTY) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanLatitude = mean(SITE_LATITUDE),
            meanLongitude = mean(SITE_LONGITUDE)) %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date))

#8 Step6: Dimension of the processed dataset
dim(EPAir_Combined_Processed)
```

```
## [1] 14752     9
```

```r
#9 Spreading the dataset
EPAir_Combined_Processed_Spread <- EPAir_Combined_Processed %>%
  pivot_wider(
```

```
    names_from = AQS_PARAMETER_DESC,
    values_from = meanAQI
  )

#10 Dimension of the dataset
dim(EPAir_Combined_Processed_Spread)
```

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

```
#11 Saving the processed dataset
write.csv(
  EPAir_Combined_Processed_Spread,
  file = here('./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv'),
  row.names = FALSE
)
```

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 The summary dataset
EPAir_Summary <- EPAir_Combined_Processed_Spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanOzone = mean(Ozone),
            meanPM2.5 = mean(PM2.5)) %>%
  drop_na(meanOzone)

#13 Dimensions of the summary dataset
dim(EPAir_Summary)
```

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

```
#14 drop.na Vs. na.omit
EPAir_SummaryOmitNA <- EPAir_Combined_Processed_Spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanOzone = mean(Ozone),
            meanPM2.5 = mean(PM2.5)) %>%
  na.omit(meanOzone)

#14 Dimensions of the summary data set using na.omit
dim(EPAir_SummaryOmitNA)
```

```
## [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: We used the 'drop_na' function as we would like to omit NA in the mean values of ozone only while keeping the mean values of PM2.5. If we use na.omit, the observations of the dataset has gone down from 182 to 101 as the 'na.omit' function removes missing values from both ozone and PM2.5.