Assignment 4: Data Wrangling (Fall 2024)

Rosalind Hu

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 Load the `tidyverse`,  `lubridate`, and `here` packages
library(tidyverse)
library(lubridate)
library(here)

#1b Check your working directory
getwd()
```

```
## [1] "/home/guest/ENV872 RosyHu"
```

```
here() #Show where the Project file is
```

```
## [1] "/home/guest/ENV872 RosyHu"
```

```r
#1c Read in all four raw data files associated with the EPA Air dataset, being sure to set string colum

EPAair_data1  <- read.csv(file = here("./Data/Raw/EPAair_O3_NC2018_raw.csv"), stringsAsFactors = TRUE)
EPAair_data2  <- read.csv(file = here("./Data/Raw/EPAair_O3_NC2019_raw.csv"), stringsAsFactors = TRUE)
EPAair_data3  <- read.csv(file = here("./Data/Raw/EPAair_PM25_NC2018_raw.csv"), stringsAsFactors = TRUE
EPAair_data4  <- read.csv(file = here("./Data/Raw/EPAair_PM25_NC2019_raw.csv"), stringsAsFactors = TRUE

#2 Reveal the dimensions of four datasets

str(EPAair_data1)
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 37003
##  $ 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 0.03 0.036 0.044 0.049 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 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",..: 35 35 35 35 35 35 35 3
##  $ DAILY_OBS_COUNT                     : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, NC",..: 9 9 9 9 9 9 9 9 9
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ 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 ...
```

```r
dim(EPAair_data1)
```

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

```r
str(EPAair_data2)
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                                : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 1 2 3 4 5
##  $ Source                              : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 37003
##  $ 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 0.037 0.029 0.038 0.038 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 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",..: 33 33 33 33 33 33 33 3
##  $ DAILY_OBS_COUNT                     : int  24 24 24 24 24 24 24 24 24 24 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ CBSA_CODE                     : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                     : Factor w/ 15 levels "","Asheville, NC",..: 8 8 8 8 8 8 8 8 8
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                   : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                        : Factor w/ 30 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ 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 ...
```

```r
dim(EPAair_data2)
```

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

```r
str(EPAair_data3)
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2018",..: 2 5 8 11 14 17
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ 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 1.7 ...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 1 1 1 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 15 15 15 15 15 15 15 15
##  $ DAILY_OBS_COUNT               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE              : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```r
dim(EPAair_data3)
```

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

```r
str(EPAair_data4)
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 3 6 9 12 15 18
##  $ Source                        : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ 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 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.",..: 14 14 14 14 14 14
##  $ DAILY_OBS_COUNT                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE               : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE             : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC             : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                      : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                     : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                          : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                    : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                         : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                  : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                 : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
dim(EPAair_data4)
```

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

All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern? - Yes, they do have 20 columns as they have 20 variables, and different numbers of rows

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 be date objects
EPAair_data1$Date <- mdy(EPAair_data1$Date)
EPAair_data2$Date <- mdy(EPAair_data2$Date)
EPAair_data3$Date <- mdy(EPAair_data3$Date)
EPAair_data4$Date <- mdy(EPAair_data4$Date)

#4 Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LAT

EPAair_data1.processed <-
  EPAair_data1 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
# View(EPAair_data1.processed)

EPAair_data2.processed <-
  EPAair_data2 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
# View(EPAair_data1.processed)
```

```r
EPAair_data3.processed <-
  EPAair_data3 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
# View(EPAair_data1.processed)

EPAair_data4.processed <-
  EPAair_data4 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
# View(EPAair_data1.processed)


#5 For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5" (all cells in this column

EPAair_data3.processed <-
  EPAair_data3.processed %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5" )



EPAair_data4.processed <-
  EPAair_data4.processed %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5" )


#6 Save all four processed datasets in the Processed folder. Use the same file names as the raw files b
# How can I Use the same file names as the raw files but replace "raw" with "processed".
write.csv(EPAair_data1.processed, row.names = FALSE, file = "./Data/Processed/EPAair_data1.processed.cs
write.csv(EPAair_data2.processed, row.names = FALSE, file = "./Data/Processed/EPAair_data2.processed.cs
write.csv(EPAair_data3.processed, row.names = FALSE, file = "./Data/Processed/EPAair_data3.processed.cs
write.csv(EPAair_data4.processed, row.names = FALSE, file = "./Data/Processed/EPAair_data4.processed.cs
```

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 the four datasets with `rbind`. Make sure your column names are identical prior to running t
# Standardize column names across all datasets (if they differ)
colnames(EPAair_data1.processed)
```

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

```
colnames(EPAair_data2.processed)
```

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

```
colnames(EPAair_data3.processed)
```

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

```
colnames(EPAair_data4.processed)
```

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

```
# Combine the four datasets with `rbind()`
combined_datasets <- rbind(
  EPAair_data1.processed,
  EPAair_data2.processed,
  EPAair_data3.processed,
  EPAair_data4.processed
)

#View(combined_datasets)

#8 Include only sites that the four data frames have in common
#Filter for the specific site names

# Filter for specific site names, remove rows with missing site info, group, and summarize
combined_datasets_processed <- combined_datasets %>%
  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, and County
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%

  # Take the mean of AQI, latitude, and longitude for each group
  summarize(mean_aqui = mean(DAILY_AQI_VALUE, na.rm = TRUE),
            mean_lat = mean(SITE_LATITUDE, na.rm = TRUE),
            mean_lon = mean(SITE_LONGITUDE, na.rm = TRUE)) %>%
  mutate(year = year(Date) , month=month(Date))
```

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

```
dim(combined_datasets_processed)
```

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

```
#9 Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location
combined_datasets_processed_inseperatecolumns <-
  combined_datasets_processed %>%
  pivot_wider(names_from = AQS_PARAMETER_DESC,
              values_from = mean_aqui)

#10 Call up the dimensions of your new tidy dataset.
dim(combined_datasets_processed_inseperatecolumns)
```

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

```
#11 Save your processed dataset with the following file name: "EPAair_O3_PM25_NC1819_Processed.csv"
write.csv(combined_datasets_processed_inseperatecolumns, row.names = FALSE, file = "./Data/Processed/EP
```

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 Generate a summary data frame, and by site, month, and year

# Combine the processed datasets
EPA_combined <- rbind(EPAair_data1.processed, EPAair_data2.processed, EPAair_data3.processed, EPAair_da

# 2. Add `year` and `month` columns
EPA_combined <- EPA_combined %>%
  mutate(year = year(Date),
```

```r
         month = month(Date))

# 3. Group by `Site.Name`, `month`, `year` and calculate the mean AQI for ozone and PM2.5
EPA_summary <- EPA_combined %>%
  group_by(Site.Name, month, year, AQS_PARAMETER_DESC) %>%
  summarise(mean_AQI = mean(DAILY_AQI_VALUE, na.rm = TRUE)) %>%
  pivot_wider(names_from = AQS_PARAMETER_DESC, values_from = mean_AQI) #using pivot_wider to generate a
```

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

```r
# 4. Use `drop_na` to remove rows where values are missing in Ozone
EPA_summary_filtered <- EPA_summary %>%
  drop_na(Ozone)
EPA_summary_filtered
```

```
## # A tibble: 716 x 5
## # Groups:   Site.Name, month, year [716]
##    Site.Name month  year PM2.5 Ozone
##    <fct>     <dbl> <dbl> <dbl> <dbl>
##  1 ""            3  2018  16.7  40.7
##  2 ""            3  2019  NA    45.4
##  3 ""            4  2018  17.4  47.2
##  4 ""            4  2019  NA    47.4
##  5 ""            5  2018  21.2  40
##  6 ""            5  2019  NA    40.3
##  7 ""            6  2018  33.2  37.5
##  8 ""            6  2019  NA    36.1
##  9 ""            7  2018  24.7  35.5
## 10 ""            7  2019  NA    28.7
## # i 706 more rows
```

```r
#13 Dimensions of the summary dataset.
dim(EPA_summary_filtered)
```

```
## [1] 716   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.

```r
EPA_summary2 <- EPA_combined %>%
  group_by(Site.Name, month, year, AQS_PARAMETER_DESC) %>%
  summarise(mean_AQI = mean(DAILY_AQI_VALUE, na.rm = TRUE)) %>%
  pivot_wider(names_from = AQS_PARAMETER_DESC, values_from = mean_AQI)
```

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

```
# 4. Use 'na.omit' instead of 'drop_na'
EPA_summary_filtered2 <- EPA_summary %>%
  na.omit(Ozone)
EPA_summary_filtered2
```

```
## # A tibble: 238 x 5
## # Groups:   Site.Name, month, year [238]
##    Site.Name    month  year PM2.5 Ozone
##    <fct>        <dbl> <dbl> <dbl> <dbl>
##  1 ""               3  2018  16.7  40.7
##  2 ""               4  2018  17.4  47.2
##  3 ""               5  2018  21.2  40
##  4 ""               6  2018  33.2  37.5
##  5 ""               7  2018  24.7  35.5
##  6 ""               8  2018  25.4  29.2
##  7 ""               9  2018  18    25.1
##  8 ""              10  2018  20.7  29.5
##  9 "Blackstone"     1  2018  44.6  34.1
## 10 "Blackstone"     2  2018  38.6  30.6
## # i 228 more rows
```

```
#13 Dimensions of the summary dataset.
dim(EPA_summary_filtered2)
```

```
## [1] 238   5
```

> Answer: The difference between 'drop_na`and`na.omit'is that drop_na only drops the missing value from that particular column while 'na.omit' which will exclude all the missing value from PM2.5 and Ozone, which we can see the dimension only 238 rows with 5 variables