Assignment 4: Data Wrangling

Kaichun Yang

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A03_DataExploration.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.

The completed exercise is due on Friday, Oct7th @ 5:00pm.

Set up your session

1. Check your working directory, load the `tidyverse` and `lubridate` packages, and upload 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).

```
# 1
library(tidyverse)
library(lubridate)
EPA_O3_2018 = read.csv(file = "E:/EDA-Fall2022/Data/Raw/EPAair_O3_NC2018_raw.csv")
EPA_O3_2019 = read.csv(file = "E:/EDA-Fall2022/Data/Raw/EPAair_O3_NC2019_raw.csv")
EPA_PM25_2018 = read.csv(file = "E:/EDA-Fall2022/Data/Raw/EPAair_PM25_NC2018_raw.csv")
EPA_PM25_2019 = read.csv(file = "E:/EDA-Fall2022/Data/Raw/EPAair_PM25_NC2019_raw.csv")
```

2. Explore the dimensions, column names, and structure of the datasets.

```
# 2
```

Wrangle individual datasets to create processed files.

3. Change date to date
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
EPA_O3_2018$Date <- as.Date(EPA_O3_2018$Date, "%m/%d/%Y")
EPA_O3_2019$Date <- as.Date(EPA_O3_2019$Date, "%m/%d/%Y")
EPA_PM25_2018$Date <- as.Date(EPA_PM25_2018$Date, "%m/%d/%Y")
EPA_PM25_2019$Date <- as.Date(EPA_PM25_2019$Date, "%m/%d/%Y")

# 4
EPA_O3_2018_S <- select(EPA_O3_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_O3_2019_S <- select(EPA_O3_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_PM25_2018_s <- select(EPA_PM25_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_PM25_2019_s <- select(EPA_PM25_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

# 5
EPA_PM25_2018_s$AQS_PARAMETER_DESC <- "PM2.5"
EPA_PM25_2019_s$AQS_PARAMETER_DESC <- "PM2.5"

# 6
write.csv(EPA_O3_2018_S, row.names = FALSE, file = "E:/EDA-Fall2022/Data/Raw/EPAair_O3_NC2018_processed
write.csv(EPA_O3_2019_S, row.names = FALSE, file = "E:/EDA-Fall2022/Data/Raw/EPAair_O3_NC2019_processed
write.csv(EPA_PM25_2018_s, row.names = FALSE, file = "E:/EDA-Fall2022/Data/Raw/EPAair_PM25_NC2018_proce
write.csv(EPA_PM25_2019_s, row.names = FALSE, file = "E:/EDA-Fall2022/Data/Raw/EPAair_PM25_NC2019_proce
```

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 all 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)
- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site, 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_NC1718_Processed.csv”

```
# intersect figure out common factor level

library(dplyr)
library(lubridate)

# 7
EPA_data <- rbind(EPA_O3_2018_S, EPA_O3_2019_S, EPA_PM25_2018_s, EPA_PM25_2019_s)

# 8
EPA_data_2 <- EPA_data %>%
    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") %>%
    group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
    summarise(meanaqi = mean(DAILY_AQI_VALUE), meanlat = mean(SITE_LATITUDE), meanlog = mean(SITE_LONGI
        .groups = "keep") %>%
    mutate(Year = year(Date), Month = month(Date))
print(EPA_data_2)
```

```
## # A tibble: 14,752 x 9
## # Groups:   Date, Site.Name, AQS_PARAMETER_DESC, COUNTY [14,752]
##    Date       Site.Name       AQS_P~1 COUNTY meanaqi meanlat meanlog  Year Month
##    <date>     <chr>           <chr>   <chr>    <dbl>   <dbl>   <dbl> <dbl> <dbl>
##  1 2018-01-01 Bryson City     PM2.5   Swain       35    35.4   -83.4  2018     1
##  2 2018-01-01 Castle Hayne    PM2.5   New H~      13    34.4   -77.8  2018     1
##  3 2018-01-01 Clemmons Middle PM2.5   Forsy~      24    36.0   -80.3  2018     1
##  4 2018-01-01 Durham Armory   PM2.5   Durham      31    36.0   -78.9  2018     1
##  5 2018-01-01 Garinger High ~ Ozone   Meckl~      32    35.2   -80.8  2018     1
##  6 2018-01-01 Garinger High ~ PM2.5   Meckl~      20    35.2   -80.8  2018     1
##  7 2018-01-01 Hattie Avenue   PM2.5   Forsy~      22    36.1   -80.2  2018     1
##  8 2018-01-01 Leggett         PM2.5   Edgec~      14    36.0   -77.6  2018     1
##  9 2018-01-01 Millbrook Scho~ Ozone   Wake        34    35.9   -78.6  2018     1
## 10 2018-01-01 Millbrook Scho~ PM2.5   Wake        28    35.9   -78.6  2018     1
## # ... with 14,742 more rows, and abbreviated variable name
## #   1: AQS_PARAMETER_DESC
```

```
# 9
EPA_data_3 <- EPA_data_2 %>%
    pivot_wider(names_from = "AQS_PARAMETER_DESC", values_from = "meanaqi")
print(EPA_data_3)
```

```
## # A tibble: 8,976 x 9
## # Groups:   Date, Site.Name, COUNTY [8,976]
##    Date       Site.Name           COUNTY meanlat meanlog  Year Month PM2.5 Ozone
##    <date>     <chr>               <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2018-01-01 Bryson City         Swain     35.4   -83.4  2018     1    35    NA
##  2 2018-01-01 Castle Hayne        New H~    34.4   -77.8  2018     1    13    NA
##  3 2018-01-01 Clemmons Middle     Forsy~    36.0   -80.3  2018     1    24    NA
##  4 2018-01-01 Durham Armory       Durham    36.0   -78.9  2018     1     31     NA
##  5 2018-01-01 Garinger High Scho~ Meckl~    35.2   -80.8  2018     1     20     32
##  6 2018-01-01 Hattie Avenue       Forsy~    36.1   -80.2  2018     1     22     NA
##  7 2018-01-01 Leggett             Edgec~    36.0   -77.6  2018     1     14     NA
##  8 2018-01-01 Millbrook School    Wake      35.9   -78.6  2018     1     28     34
##  9 2018-01-01 Pitt Agri. Center   Pitt      35.6   -77.4  2018     1     15     NA
## 10 2018-01-01 West Johnston Co.   Johns~    35.6   -78.5  2018     1     24     NA
## # ... with 8,966 more rows
```

```
# 10
dim(EPA_data_3)
```

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

```
# 11
write.csv(EPA_data_3, row.names = FALSE, file = "E:/EDA-Fall2022/Data/Raw/EPAair_O3_PM25_NC1718_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 a month and year are not available (use the function `drop_na` in your pipe).

13. Call up the dimensions of the summary dataset.

```
# 12a
EPA_data_summary <- EPA_data_3 %>%
    group_by(Site.Name, Month, Year) %>%
    summarise(meanaqi_pm = mean(PM2.5), meanaqi_o3 = mean(Ozone), .groups = "keep")
print(EPA_data_summary)
```

```
## # A tibble: 308 x 5
## # Groups:   Site.Name, Month, Year [308]
##    Site.Name   Month  Year meanaqi_pm meanaqi_o3
##    <chr>       <dbl> <dbl>      <dbl>      <dbl>
##  1 Bryson City     1  2018       38.9       NA
##  2 Bryson City     1  2019       29.8       NA
##  3 Bryson City     2  2018       27.2       NA
##  4 Bryson City     2  2019       33.0       NA
##  5 Bryson City     3  2018       34.7       41.6
##  6 Bryson City     3  2019       NA         42.5
##  7 Bryson City     4  2018       28.2       44.5
##  8 Bryson City     4  2019       26.7       45.4
##  9 Bryson City     5  2018       NA         NA
## 10 Bryson City     5  2019       NA         39.6
## # ... with 298 more rows
```

```
# 12b
EPA_data_summary_2 <- drop_na(EPA_data_summary)
print(EPA_data_summary_2)
```

```
## # A tibble: 101 x 5
## # Groups:   Site.Name, Month, Year [101]
##    Site.Name    Month  Year meanaqi_pm meanaqi_o3
##    <chr>        <dbl> <dbl>      <dbl>      <dbl>
##  1 Bryson City      3  2018       34.7       41.6
##  2 Bryson City      4  2018       28.2       44.5
##  3 Bryson City      4  2019       26.7       45.4
##  4 Bryson City      7  2019       33.6       30.4
##  5 Bryson City      9  2018       25.1       25.4
##  6 Bryson City     10  2018       31.3       31
##  7 Castle Hayne     4  2018       14.9       48.7
##  8 Castle Hayne     4  2019       14.3       45.1
##  9 Castle Hayne     5  2019       16.5       42.8
## 10 Castle Hayne     7  2018       15.5       36.5
## # ... with 91 more rows
```

```
# 13
dim(EPA_data_summary_2)
```

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

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

    Answer: drop_na() drops rows where any column specified by … contains a missing value. na.omit returns the object with incomplete cases removed.