Assignment 4: Data Wrangling

Xuening Tang

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).

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

```
#1
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
library(lubridate)

EPAPM2.5_18 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv",stringsAsFactors = TRUE)
EPAPM2.5_19 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv",stringsAsFactors = TRUE)
EPAO3_18 <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv",stringsAsFactors = TRUE)
EPAO3_19 <- read.csv("../Data/Raw/EPAair_O3_NC2019_raw.csv",stringsAsFactors = TRUE)

#2
dim(EPAO3_18)
```

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

```
dim(EPAO3_19)
```

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

```
dim(EPAPM2.5_18)
```

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

```
dim(EPAPM2.5_19)
```

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

```
colnames(EPAO3_18)
```

```
##  [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(EPAO3_19)
```

```
##  [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(EPAPM2.5_18)
```

```
##  [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(EPAPM2.5_19)
```

```
##  [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"
```

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
EPAO3_18$Date <- as.Date(EPAO3_18$Date, format = "%m/%d/%Y")
EPAO3_19$Date <- as.Date(EPAO3_19$Date, format = "%m/%d/%Y")
EPAPM2.5_18$Date <- as.Date(EPAPM2.5_18$Date, format = "%m/%d/%Y")
EPAPM2.5_19$Date <- as.Date(EPAPM2.5_19$Date, format = "%m/%d/%Y")

#4
EPAO3_18_aq <- select(EPAO3_18, Date, DAILY_AQI_VALUE, Site.Name,
                      AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPAO3_19_aq <- select(EPAO3_19, Date, DAILY_AQI_VALUE, Site.Name,
                      AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPAPM2.5_18_aq <- select(EPAPM2.5_18, Date, DAILY_AQI_VALUE, Site.Name,
                         AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPAPM2.5_19_aq <- select(EPAPM2.5_19, Date, DAILY_AQI_VALUE, Site.Name,
                         AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#5
EPAPM2.5_18_aq$AQS_PARAMETER_DESC <- "PM2.5"
EPAPM2.5_19_aq$AQS_PARAMETER_DESC <- "PM2.5"


#6
write.csv(EPAO3_18_aq, row.names = FALSE,file =
            "../Data/Processed/EPAair_O3_NC2018_Processed.csv")

write.csv(EPAO3_19_aq,row.names = FALSE, file =
            "../Data/Processed/EPAair_O3_NC2019_Processed.csv")

write.csv(EPAPM2.5_18_aq, row.names = FALSE,file =
            "../Data/Processed/EPAair_PM25_NC2018_Processed.csv")

write.csv(EPAPM2.5_19_aq, 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 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"

```
#7
EPA_data <- rbind(EPAO3_18_aq, EPAO3_19_aq,EPAPM2.5_18_aq, EPAPM2.5_19_aq )
dim(EPA_data)
```

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

```
#8
EPA.data.summaries <-
  EPA_data %>%
  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),
            meanLA = mean(SITE_LATITUDE),
            meanLG = mean(SITE_LONGITUDE)) %>%
  mutate(Month=month(Date))%>%
  mutate(Year=year(Date))
```

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

```
#9
EPA.data.summaries.spread <-
  pivot_wider(
    EPA.data.summaries, names_from = AQS_PARAMETER_DESC, values_from = meanAQI)

#10
dim(EPA.data.summaries.spread)
```

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

```
#11
write.csv(EPA.data.summaries.spread, row.names = FALSE,
          file ="../Data/Processed/EPAair_O3_PM25_NC1718_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 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 #12b
EPA.AQ.summaries <-
  EPA.data.summaries.spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(meanPM2.5 = mean(PM2.5),
            meanOzone = mean(Ozone))%>%
  filter(!if_all(c(meanPM2.5, meanOzone), is.na))
```

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

```
##To grader, I tried using 'drop_na' function but always failed with
##just eliminate NA in both columns, it always drop all NA in 'meanPM2.5'
##or 'meanOzone' so i used filter function.
##Here I put the 'drop_na' function code I tried:
##"drop_na(meanPM2.5, meanOzone)"


#13
dim(EPA.AQ.summaries)
```

```
## [1] 292   5
```

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

    Answer: Because 'na.omit' will remove all rows contain NA based on all columns of a data objectin. So in this dataset, we just want to remove NA appeared in all certain columns rather than affecting values in other columns, that's why we use 'drop_na'.