Assignment 4: Data Wrangling

Camila Zarate Ospina

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A04_DataWrangling.Rmd") prior to submission.

The completed exercise is due on Tuesday, Feb 16 @ 11:59pm.

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. 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.

```
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff=80),
                      tidy = FALSE)
```

```
#1 Working directory, libraries, reading the datasets.
getwd()
```

```
## [1] "/Users/camilazarate/OneDrive - Duke University/2 Second semester/Data analytics/Environmental_Da
```

```
setwd("/Users/camilazarate/OneDrive - Duke University/2 Second semester/Data analytics/Environmental_Dat
library(tidyverse)
library(lubridate)

PM25_NC2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv",
                        stringsAsFactors = TRUE)
PM25_NC2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv",
                        stringsAsFactors = TRUE)

O3_NC2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv",
                      stringsAsFactors = TRUE)
O3_NC2019 <- read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv",
                      stringsAsFactors = TRUE)

#2 Dimensions of datasets.
colnames(PM25_NC2018)
```

```
##  [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"
```

```r
dim(PM25_NC2018)
```

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

```r
colnames(PM25_NC2019)
```

```
##  [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"
```

```r
dim(PM25_NC2019)
```

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

```r
colnames(O3_NC2018)
```

```
##  [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"
```

```r
dim(O3_NC2018)
```

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

```r
colnames(O3_NC2019)
```

```
##  [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"
```

```r
dim(O3_NC2019)
```

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

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".

```r
#3 Change the dates to Date format.
class(PM25_NC2018$Date)
```

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

```r
PM25_NC2018$Date <- as.Date(PM25_NC2018$Date, format = "%m/%d/%Y")

class(PM25_NC2019$Date)
```

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

```r
PM25_NC2019$Date <- as.Date(PM25_NC2019$Date, format = "%m/%d/%Y")

class(O3_NC2018$Date)
```

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

```r
O3_NC2018$Date <- as.Date(O3_NC2018$Date, format = "%m/%d/%Y")

class(O3_NC2019$Date)
```

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

```r
O3_NC2019$Date <- as.Date(O3_NC2019$Date, format = "%m/%d/%Y")

#4 Select columns
PM25_NC2018_select <- select(PM25_NC2018, Date, DAILY_AQI_VALUE,
                             Site.Name, AQS_PARAMETER_DESC, COUNTY,
                             SITE_LATITUDE, SITE_LONGITUDE)

PM25_NC2019_select <- select(PM25_NC2019, Date, DAILY_AQI_VALUE,
                             Site.Name, AQS_PARAMETER_DESC, COUNTY,
                             SITE_LATITUDE, SITE_LONGITUDE)

O3_NC2018_select <- select(O3_NC2018, Date, DAILY_AQI_VALUE,
                           Site.Name, AQS_PARAMETER_DESC, COUNTY,
                           SITE_LATITUDE, SITE_LONGITUDE)

O3_NC2019_select <- select(O3_NC2019, Date, DAILY_AQI_VALUE,
                           Site.Name, AQS_PARAMETER_DESC, COUNTY,
                           SITE_LATITUDE, SITE_LONGITUDE)

#5 Write PM2.5 in AQS_PARAMETER_DES
PM25_NC2018_select <- mutate(PM25_NC2018_select, AQS_PARAMETER_DESC = "PM2.5")
PM25_NC2019_select <- mutate(PM25_NC2019_select, AQS_PARAMETER_DESC = "PM2.5")

#6 Save processed datasets
write.csv(PM25_NC2018_select, row.names = FALSE, file =
            "./Data/Processed/EPAair_PM25_NC2018_processed.csv")
write.csv(PM25_NC2019_select, row.names = FALSE, file =
            "./Data/Processed/EPAair_PM25_NC2019_processed.csv")
write.csv(O3_NC2018_select, row.names = FALSE, file =
            "./Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(O3_NC2019_select, row.names = FALSE, file =
            "./Data/Processed/EPAair_O3_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 Combine datasets
# Read processed datasets
#PM25_2018 <- read.csv("./Data/Processed/EPAair_PM25_NC2018_processed.csv", stringsAsFactors = TRUE)
#PM25_2019 <- read.csv("./Data/Processed/EPAair_PM25_NC2019_processed.csv", stringsAsFactors = TRUE)
#O3_2018 <- read.csv("./Data/Processed/EPAair_O3_NC2018_processed.csv", stringsAsFactors = TRUE)
#O3_2019 <- read.csv("./Data/Processed/EPAair_O3_NC2019_processed.csv", stringsAsFactors = TRUE)
# Combine datasets
EPA_Air <- rbind(PM25_NC2018_select, PM25_NC2019_select, O3_NC2018_select, O3_NC2019_select)

#8 Wrangle dataset
EPA_Air_Processed <- EPA_Air %>%
  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),
            meanLongitud = mean(SITE_LONGITUDE)) %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date))
```

```
## `summarise()` regrouping output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC' (override with `.groups
```

```
#9 Spread column that contains ozone and PM2.5 values.
# This generates 2 new columns
EPA_Air_spread <- pivot_wider(EPA_Air_Processed, names_from = AQS_PARAMETER_DESC,
                              values_from = meanAQI)

#10
colnames(EPA_Air_spread)
```

```
## [1] "Date"         "Site.Name"    "COUNTY"       "meanLatitude" "meanLongitud"
## [6] "month"        "year"         "PM2.5"        "Ozone"
```

```
dim(EPA_Air_spread)
```

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

```
#11
write.csv(EPA_Air_spread, row.names = FALSE, file =
            "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.

```
# 12 Summaries
# Logic: Take data by site, month and year. Take the mean of all data from 1 site in a specific month i
```

```r
# Summaries using drop_na
EPA_Air_summaries <- EPA_Air_spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanIQ_Ozone = mean(PM2.5),
            meanIQ_pm25 = mean(Ozone)) %>%
  drop_na(month, year)
```

```
## `summarise()` regrouping output by 'Site.Name', 'month' (override with `.groups` argument)
```

```r
# Summaries using na.omit
EPA_Air_summaries2 <- EPA_Air_spread %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanIQ_Ozone = mean(PM2.5),
            meanIQ_pm25 = mean(Ozone)) %>%
  na.omit(month, year)
```

```
## `summarise()` regrouping output by 'Site.Name', 'month' (override with `.groups` argument)
```

```r
#13 Dimensions
dim(EPA_Air_summaries) #drop_na
```

```
## [1] 308   5
```

```r
dim(EPA_Air_summaries2) #na.omit
```

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

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

    Answer: The function "na.omit" omits all the rows that contain NA's in the dataset, instead of only focusing on the columns "month" and "year". The result is a table with no NA's at all and 101 observations. The "drop_na" function only focusses on the columns "month" and "year", and since there are no NA's in those columns, the resulting table keeps 308 observations. Since we only want to remove NA's from the columns "month" and "year", we used "drop_na".