Assignment 4: Data Wrangling

Logan Loadholtz

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.

```
#1
getwd()
```

```
## [1] "/Users/loganloadholtz/Documents/DATA/Environmental_Data_Analytics_2021/Assignments"
```

```
library(tidyverse)
library(lubridate)

library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=50),tidy=TRUE)

EPAair_PM25_NC2019 <- read.csv("~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Raw/EPAair_PM25_
                               stringsAsFactors = FALSE)

EPAair_PM25_NC2018 <- read.csv("~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Raw/EPAair_PM25_
                               stringsAsFactors = FALSE)

EPAair_O3_NC2019 <-read.csv("~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Raw/EPAair_O3_NC20
                            stringsAsFactors = FALSE)

EPAair_O3_NC2018 <-read.csv("~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Raw/EPAair_O3_NC20
                            stringsAsFactors = FALSE)
```

```r
#2
dim(EPAair_PM25_NC2018)
```

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

```r
dim(EPAair_PM25_NC2019)
```

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

```r
dim(EPAair_O3_NC2018)
```

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

```r
dim(EPAair_O3_NC2019)
```

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

```r
#all datasets have 20 columns, so that is good for us to see that they are all containing
#the same information

colnames(EPAair_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
#I will just take column names of the first set here

str(EPAair_PM25_NC2018, width=80, strict.width="cut")
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : chr  "01/02/2018" "01/05/2018" "01/08/2018"..
##  $ Source                        : chr  "AQS" "AQS" "AQS" "AQS" ...
##  $ Site.ID                       : int  370110002 370110002 370110002 37011000..
##  $ 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...
##  $ UNITS                         : chr  "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" "ug/"..
##  $ DAILY_AQI_VALUE               : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                     : chr  "Linville Falls" "Linville Falls" "Li"..
##  $ 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 10..
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88..
##  $ AQS_PARAMETER_DESC            : chr  "Acceptable PM2.5 AQI & Speciation Ma"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : chr  "" "" "" "" ...
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : chr  "North Carolina" "North Carolina" "No"..
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : chr  "Avery" "Avery" "Avery" "Avery" ...
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

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
class(EPAair_O3_NC2018$Date)
```

```
## [1] "character"
```

```
#Here, we can see that date is a factor variable. We will need to change that to date

EPAair_O3_NC2018$Date <- as.Date(EPAair_O3_NC2018$Date, format = "%m/%d/%Y")
class(EPAair_O3_NC2018$Date)
```

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

```
EPAair_O3_NC2019$Date <- as.Date(EPAair_O3_NC2019$Date, format = "%m/%d/%Y")
class(EPAair_O3_NC2019$Date)
```

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

```
EPAair_PM25_NC2018$Date <- as.Date(EPAair_PM25_NC2018$Date, format = "%m/%d/%Y")
class(EPAair_PM25_NC2018$Date)
```

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

```
EPAair_PM25_NC2019$Date <- as.Date(EPAair_PM25_NC2019$Date, format = "%m/%d/%Y")
class(EPAair_PM25_NC2019$Date)
```

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

```
#4
EPAair_O3_NC2018_select <- select(EPAair_O3_NC2018, Date, DAILY_AQI_VALUE, Site.Name,
                                  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair_O3_NC2019_select <- select(EPAair_O3_NC2019, Date, DAILY_AQI_VALUE, Site.Name,
                                  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair_PM25_NC2018_select <- select(EPAair_PM25_NC2018, Date, DAILY_AQI_VALUE, Site.Name,
                                    AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair_PM25_NC2019_select <- select(EPAair_PM25_NC2019, Date, DAILY_AQI_VALUE, Site.Name,
                                    AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#5

EPAair_PM25_NC2018_select <- mutate(EPAair_PM25_NC2018_select, AQS_PARAMETER_DESC = "PM2.5")

EPAair_PM25_NC2019_select <- mutate(EPAair_PM25_NC2019_select, AQS_PARAMETER_DESC = "PM2.5")

#6
write.csv(EPAair_PM25_NC2018_select, row.names = FALSE, file=
            "~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Processed/
```

```
          EPAair_PM25_NC2018_select_Processed.csv")

write.csv(EPAair_PM25_NC2019_select, row.names = FALSE, file=
            "~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Processed/
          EPAair_PM25_NC2019_select_Processed.csv")

write.csv(EPAair_O3_NC2018_select, row.names = FALSE, file=
            "~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Processed/
          EPAair_O3_NC2018_select_Processed.csv")

write.csv(EPAair_O3_NC2019_select, row.names = FALSE, file=
            "~/Documents/DATA/Environmental_Data_Analytics_2021/Data/Processed/
          EPAair_O3_NC2019_select_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
AirQuality_O3_PM25 <- rbind(EPAair_O3_NC2018_select, EPAair_O3_NC2019_select,
                            EPAair_PM25_NC2018_select, EPAair_PM25_NC2019_select)

#8
AirQuality_O3_PM25_processed <-
  AirQuality_O3_PM25 %>%
  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) %>%
  filter(!is.na(DAILY_AQI_VALUE) & !is.na(SITE_LATITUDE) & !is.na(SITE_LONGITUDE)) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlat= mean(SITE_LATITUDE),
            meanlong= 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

AirQuality_O3_PM25_wide <- pivot_wider(AirQuality_O3_PM25_processed, names_from =AQS_PARAMETER_DESC,
                                       values_from = meanAQI)
#10
dim(AirQuality_O3_PM25_wide)
```

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

```
#11
write.csv(AirQuality_O3_PM25_wide, row.names = FALSE, file=
            "~/Documents/DATA/Environmental_Data_Analytics_2021/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
Summary_AirQuality_O3_PM25_wide <-
  AirQuality_O3_PM25_wide %>%
  group_by(Site.Name, month, year) %>%
  summarise(meanO3 = mean(Ozone),
            meanPM25 =mean(PM2.5))
```

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

```
#12b
Summary_AirQuality_O3_PM25_wide_removeNA <-
  Summary_AirQuality_O3_PM25_wide %>%
  drop_na(month, year)




#13
dim(Summary_AirQuality_O3_PM25_wide_removeNA)
```

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

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

    Answer: drop__na is used for specific columns. Here, we specified that we want to remove rows with NA values from the month and year column. na.omit drops NA values from the whole dataframe.