Assignment 4: Data Wrangling

Meghan Seyler

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 Monday, Feb 7 @ 7:00pm.

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. 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.

```r
#1
getwd()
library(tidyverse)
library(lubridate)

EPAair_03_NC2018<- read.csv("../Data/Raw/EPAair_03_NC2018_raw.csv", stringsAsFactors = TRUE)
EPAair_03_NC2019<- read.csv("../Data/Raw/EPAair_03_NC2019_raw.csv", stringsAsFactors = TRUE)
EPAair_PM25_NC2018<- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
EPAair_PM25_NC2019<- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)

#2
colnames(EPAair_03_NC2018)
head(EPAair_03_NC2018)
summary(EPAair_03_NC2018)
str(EPAair_03_NC2018)
dim(EPAair_03_NC2018)

colnames(EPAair_03_NC2019)
head(EPAair_03_NC2019)
summary(EPAair_03_NC2019)
str(EPAair_03_NC2019)
dim(EPAair_03_NC2019)
```

```
colnames(EPAair_PM25_NC2018)
head(EPAair_PM25_NC2018)
summary(EPAair_PM25_NC2018)
str(EPAair_PM25_NC2018)
dim(EPAair_PM25_NC2018)

colnames(EPAair_PM25_NC2019)
head(EPAair_PM25_NC2019)
summary(EPAair_PM25_NC2019)
str(EPAair_PM25_NC2019)
dim(EPAair_PM25_NC2019)
```

Wrangle individual datasets to create processed files.

3. Change date to a date object
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
EPAair_03_NC2018$Date <- mdy(EPAair_03_NC2018$Date)
EPAair_03_NC2019$Date <- mdy(EPAair_03_NC2019$Date)
EPAair_PM25_NC2018$Date <- mdy(EPAair_PM25_NC2018$Date)
EPAair_PM25_NC2019$Date <- mdy(EPAair_PM25_NC2019$Date)

class(EPAair_03_NC2018$Date)
```

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

```
class(EPAair_03_NC2019$Date)
```

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

```
class(EPAair_PM25_NC2018$Date)
```

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

```
class(EPAair_PM25_NC2019$Date)
```

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

```
#4
EPAair_03_NC2018_7col <- select(EPAair_03_NC2018,Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, C
EPAair_03_NC2019_7col <- select(EPAair_03_NC2019,Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, C
EPAair_PM25_NC2018_7col <- select(EPAair_PM25_NC2018,Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES
EPAair_PM25_NC2019_7col <- select(EPAair_PM25_NC2019,Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES

colnames(EPAair_03_NC2018_7col)
```

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

```
colnames(EPAair_03_NC2019_7col)
```

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

```
colnames(EPAair_PM25_NC2018_7col)
```

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

```
colnames(EPAair_PM25_NC2019_7col)
```

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

```
#5
EPAair_PM25_NC2018_7col$AQS_PARAMETER_DESC<-"PM2.5"
EPAair_PM25_NC2019_7col$AQS_PARAMETER_DESC<-"PM2.5"

head(EPAair_PM25_NC2018_7col$AQS_PARAMETER_DESC)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
```

```
head(EPAair_PM25_NC2019_7col$AQS_PARAMETER_DESC)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
```

```
#6
write.csv(EPAair_O3_NC2018_7col, row.names = FALSE, file = "../Data/Processed/EPAair_O3_NC2018_processe
write.csv(EPAair_O3_NC2019_7col, row.names = FALSE, file = "../Data/Processed/EPAair_O3_NC2019_processe
write.csv(EPAair_PM25_NC2018_7col, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2018_7col
write.csv(EPAair_PM25_NC2019_7col, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2019_7col
```

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:

- Filter records to include just the 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 `intersect` function can figure out common factor levels if we didn't give you this list...)
- 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_NC2122_Processed.csv"

```
#7
EPAair_O3_PM25_2018.2019<-rbind(EPAair_O3_NC2018_7col,EPAair_O3_NC2019_7col,EPAair_PM25_NC2018_7col,EPAa
#8
EPAair_O3_PM25_2018.2019_filter<-EPAair_O3_PM25_2018.2019 %>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanLat = mean(SITE_LATITUDE),
            meanLong = 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
```

```
dim(EPAair_O3_PM25_2018.2019_filter)
```

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

```
EPAair_O3_PM25_2018.2019_filter
```

```
## # A tibble: 14,752 x 9
## # Groups:   Date, Site.Name, AQS_PARAMETER_DESC [14,752]
##    Date       Site.Name   AQS_PARAMETER_D~ COUNTY meanAQI meanLat meanLong Month
##    <date>     <fct>       <fct>            <fct>    <dbl>   <dbl>    <dbl> <dbl>
##  1 2018-01-01 Bryson City PM2.5            Swain       35    35.4    -83.4     1
##  2 2018-01-01 Castle Hay~ PM2.5            New H~      13    34.4    -77.8     1
##  3 2018-01-01 Clemmons M~ PM2.5            Forsy~      24    36.0    -80.3     1
##  4 2018-01-01 Durham Arm~ PM2.5            Durham      31    36.0    -78.9     1
##  5 2018-01-01 Garinger H~ Ozone            Meckl~      32    35.2    -80.8     1
##  6 2018-01-01 Garinger H~ PM2.5            Meckl~      20    35.2    -80.8     1
##  7 2018-01-01 Hattie Ave~ PM2.5            Forsy~      22    36.1    -80.2     1
##  8 2018-01-01 Leggett     PM2.5            Edgec~      14    36.0    -77.6     1
##  9 2018-01-01 Millbrook ~ Ozone            Wake        34    35.9    -78.6     1
## 10 2018-01-01 Millbrook ~ PM2.5            Wake        28    35.9    -78.6     1
## # ... with 14,742 more rows, and 1 more variable: Year <dbl>
```

```
#9
EPAair_O3_PM25_2018.2019_split <- pivot_wider(EPAair_O3_PM25_2018.2019_filter, names_from = AQS_PARAMET

EPAair_O3_PM25_2018.2019_split
```

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

```
#10
dim(EPAair_O3_PM25_2018.2019)
```

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

```
#11
write.csv(EPAair_O3_PM25_2018.2019, row.names = FALSE, file = "../Data/Processed/EPAair_O3_PM25_2018.20
```

Generate summary tables

12a. Use the split-apply-combine strategy to generate a summary data frame from your results from Step 9 above. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group.

12b. BONUS: Add a piped statement to 12a that removes rows where both mean ozone and mean PM2.5 have missing values.

13. Call up the dimensions of the summary dataset.

```
#12(a,b)
EPAair_O3_PM25_2018.2019_split_summaries <-
  EPAair_O3_PM25_2018.2019_split %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(meanAQI_Ozone = mean(Ozone),
            meanAQI_PM2.5 = mean(PM2.5))%>%
   filter(!is.na(meanAQI_Ozone) & !is.na(meanAQI_PM2.5))
```

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

```
EPAair_O3_PM25_2018.2019_split_summaries
```

```
## # A tibble: 101 x 5
## # Groups:   Site.Name, Month [74]
##    Site.Name    Month  Year meanAQI_Ozone meanAQI_PM2.5
##    <fct>        <dbl> <dbl>         <dbl>         <dbl>
##  1 Bryson City      3  2018          41.6          34.7
##  2 Bryson City      4  2018          44.5          28.2
##  3 Bryson City      4  2019          45.4          26.7
##  4 Bryson City      7  2019          30.4          33.6
##  5 Bryson City      9  2018          25.4          25.1
##  6 Bryson City     10  2018          31            31.3
##  7 Castle Hayne     4  2018          48.7          14.9
##  8 Castle Hayne     4  2019          45.1          14.3
##  9 Castle Hayne     5  2019          42.8          16.5
## 10 Castle Hayne     7  2018          36.5          15.5
## # ... with 91 more rows
```

```
#13
dim(EPAair_O3_PM25_2018.2019_split_summaries)
```

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

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

    Answer: Because drop_na is included in the tidyverse and na.omit removes all NAs but drop_na allows us to pick the specific rows with NAs that we would like to remove.