

Assignment 4: Data Wrangling

Jacob Freedman

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 Loading packages and loading in 4 EPA raw datasets
```

```
library(tidyverse)
library(lubridate)
```

```
EPA.Ozone.NC.2018 <- read.csv("./Data/Raw/EPAair_03_NC2018_raw.csv", stringsAsFactors = TRUE)
EPA.Ozone.NC.2019 <- read.csv("./Data/Raw/EPAair_03_NC2019_raw.csv", stringsAsFactors = TRUE)
EPA.PM25.NC.2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
EPA.PM25.NC.2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)
```

```
# 2 Looking at dimensions, column names, and structure of 4 datasets.
```

```
dim(EPA.Ozone.NC.2018)
```

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

```
dim(EPA.Ozone.NC.2019)
```

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

```
dim(EPA.PM25.NC.2018)
```

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

```
dim(EPA.PM25.NC.2019)
```

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

```
colnames(EPA.Ozone.NC.2018)
```

```
## [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] "AQ5_PARAMETER_CODE"
## [12] "AQ5_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(EPA.Ozone.NC.2019)
```

```
## [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] "AQ5_PARAMETER_CODE"
## [12] "AQ5_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(EPA.PM25.NC.2018)
```

```
## [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(EPA.PM25.NC.2019)
```

```
## [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"
```

```
str(EPA.Ozone.NC.2018)
```

```
## 'data.frame': 9737 obs. of 20 variables:
## $ Date : Factor w/ 364 levels "01/01/2018","01/02/2018",...: 60 61 62
## $ Source : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Max.8.hour.Ozone.Concentration: num 0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0.049 ...
## $ UNITS : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 40 43 44 45 44 28 33 41 45 40 ...
## $ Site.Name : Factor w/ 40 levels "", "Beaufort",...: 35 35 35 35 35 35 35 35 35 35 ...
## $ DAILY_OBS_COUNT : int 17 17 17 17 17 17 17 17 17 17 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201 ...
## $ AQS_PARAMETER_DESC : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860 ...
## $ CBSA_NAME : Factor w/ 17 levels "", "Asheville, NC",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
## $ COUNTY : Factor w/ 32 levels "Alexander","Avery",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
str(EPA.Ozone.NC.2019)
```

```
## 'data.frame': 10592 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2019","01/02/2019",...: 1 2 3 4 5
## $ Source : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Max.8.hour.Ozone.Concentration: num 0.029 0.018 0.016 0.022 0.037 0.037 0.029 0.038 0.038 0.038 ...
## $ UNITS : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 27 17 15 20 34 34 27 35 35 28 ...
## $ Site.Name : Factor w/ 38 levels "", "Beaufort",...: 33 33 33 33 33 33 33 33 33 33 ...
## $ DAILY_OBS_COUNT : int 24 24 24 24 24 24 24 24 24 24 ...
```

```
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201 ...
## $ AQS_PARAMETER_DESC : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860 ...
## $ CBSA_NAME : Factor w/ 15 levels "", "Asheville, NC", ...: 8 8 8 8 8 8 8 8 8 8 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
## $ COUNTY : Factor w/ 30 levels "Alexander", "Avery", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
str(EPA.PM25.NC.2018)
```

```
## 'data.frame': 8983 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2018", "01/02/2018", ...: 2 5 8 11 14 17 ...
## $ Source : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 ...
## $ 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 1.7 ...
## $ UNITS : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 12 15 22 3 10 19 8 10 18 7 ...
## $ Site.Name : Factor w/ 25 levels "", "Blackstone", ...: 15 15 15 15 15 15 15 15 15 15 ...
## $ DAILY_OBS_COUNT : int 1 1 1 1 1 1 1 1 1 1 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 88502 88502 88502 88502 88502 88502 88502 88502 88502 88502 ...
## $ AQS_PARAMETER_DESC : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int NA NA NA NA NA NA NA NA NA NA ...
## $ CBSA_NAME : Factor w/ 14 levels "", "Asheville, NC", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 11 11 11 11 11 11 11 11 11 11 ...
## $ COUNTY : Factor w/ 21 levels "Avery", "Buncombe", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 36 36 36 36 36 ...
## $ SITE_LONGITUDE : num -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
str(EPA.PM25.NC.2019)
```

```
## 'data.frame': 8581 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2019", "01/02/2019", ...: 3 6 9 12 15 18 ...
## $ Source : Factor w/ 2 levels "AirNow", "AQS": 2 2 2 2 2 2 2 2 2 2 ...
## $ Site.ID : int 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Mean.PM2.5.Concentration: num 1.6 1 1.3 6.3 2.6 1.2 1.5 1.5 3.7 1.6 ...
## $ UNITS : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 7 4 5 26 11 5 6 15 7 ...
## $ Site.Name : Factor w/ 25 levels "", "Board Of Ed. Bldg.", ...: 14 14 14 14 14 14 14 14 14 14 ...
## $ DAILY_OBS_COUNT : int 1 1 1 1 1 1 1 1 1 1 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 88502 88502 88502 88502 88502 88502 88502 88502 88502 88502 ...
## $ AQS_PARAMETER_DESC : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int NA NA NA NA NA NA NA NA NA NA ...
## $ CBSA_NAME : Factor w/ 14 levels "", "Asheville, NC", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 11 11 11 11 11 11 11 11 11 11 ...
```

```
## $ COUNTY : Factor w/ 21 levels "Avery","Buncombe",...: 1 1 1 1 1 1 1 1 1 1 ..
## $ 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 Changing the dates from factors to dates.

```
EPA.Ozone.NC.2018$Date <- as.Date(EPA.Ozone.NC.2018$Date, format = "%m/%d/%Y")
EPA.Ozone.NC.2019$Date <- as.Date(EPA.Ozone.NC.2019$Date, format = "%m/%d/%Y")
EPA.PM25.NC.2018$Date <- as.Date(EPA.PM25.NC.2018$Date, format = "%m/%d/%Y")
EPA.PM25.NC.2019$Date <- as.Date(EPA.PM25.NC.2019$Date, format = "%m/%d/%Y")
```

4 Selecting for only the desired columns and saving in a new dataframe.

```
EPA.Ozone.NC.2018.Refined <- select(EPA.Ozone.NC.2018, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA.Ozone.NC.2019.Refined <- select(EPA.Ozone.NC.2019, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA.PM25.NC.2018.Refined <- select(EPA.PM25.NC.2018, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA.PM25.NC.2019.Refined <- select(EPA.PM25.NC.2019, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
```

5 Replacing all values in AQS_PARAMETER_DESC column with the value 'PM2.5'

```
EPA.PM25.NC.2018.Refined$AQS_PARAMETER_DESC <- c("PM2.5")
EPA.PM25.NC.2019.Refined$AQS_PARAMETER_DESC <- c("PM2.5")
```

Verifying dimensions of the dataframes.

```
dim(EPA.Ozone.NC.2018.Refined)
```

```
## [1] 9737 7
```

```
dim(EPA.Ozone.NC.2019.Refined)
```

```
## [1] 10592 7
```

```
dim(EPA.PM25.NC.2018.Refined)
```

```
## [1] 8983 7
```

```
dim(EPA.PM25.NC.2019.Refined)
```

```
## [1] 8581 7
```

6 Exporting the new dataframes into the Processed Data folder

```
write.csv(EPA.Ozone.NC.2018.Refined, file = "./Data/Processed/EPAair_03_NC2018_Processed.csv",
  row.names = FALSE)
```

```
write.csv(EPA.Ozone.NC.2019.Refined, file = "../Data/Processed/EPAair_O3_NC2019_Processed.csv",
  row.names = FALSE)
write.csv(EPA.PM25.NC.2018.Refined, file = "../Data/Processed/EPAair_PM25_NC2018_Processed.csv",
  row.names = FALSE)
write.csv(EPA.PM25.NC.2019.Refined, file = "../Data/Processed/EPAair_PM25_NC2019_Processed.csv",
  row.names = FALSE)
```

Combine datasets

- Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.
- 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.
- 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.
- Call up the dimensions of your new tidy dataset.
- Save your processed dataset with the following file name: “EPAair_O3_PM25_NC1718_Processed.csv”
#I am assuming this should be 18/19?

7 Combining 4 refined datasets into one data frame.

```
EPA.NC.TOTAL <- rbind(EPA.Ozone.NC.2018.Refined, EPA.Ozone.NC.2019.Refined, EPA.PM25.NC.2018.Refined,
  EPA.PM25.NC.2019.Refined)
```

```
dim(EPA.NC.TOTAL)
```

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

8 Selecting data for certain sites, grouping by multiple parameters. Then

taking mean AQI values to ensure one row represents one day for one site.

Also adding month and year columns.

```
EPA.NC.TOTAL.SelectedSites <- EPA.NC.TOTAL %>%
```

```
  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), 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 the `.groups` argument.
```

```
dim(EPA.NC.TOTAL.SelectedSites) #Verifying dimensions of new data frame.

## [1] 14752      9

# 9 Splitting Ozone and PM2.5 values using the pivot_wider function.
EPA.NC.TOTAL.SelectedSites.Spread <- pivot_wider(EPA.NC.TOTAL.SelectedSites, names_from = AQS_PARAMETER,
  values_from = meanAQI)

# 10 Displaying dataframe dimensions.
dim(EPA.NC.TOTAL.SelectedSites.Spread)

## [1] 8976      9

# 11 Saving the processed file.
write.csv(EPA.NC.TOTAL.SelectedSites.Spread, file = "./Data/Processed/EPAair_03_PM25_NC1819_Processed.csv",
  row.names = FALSE)
```

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 Creating a summary data frame grouped by site, month, and year. Removing
# any rows with NA for Ozone or PM2.5 values.
EPA.NC.TOTAL.Summary <- EPA.NC.TOTAL.SelectedSites.Spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(meanOZONE = mean(Ozone), meanPM2.5 = mean(PM2.5)) %>%
  drop_na(meanOZONE, meanPM2.5)

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

# 13 Displaying dimensions of new dataframe.
dim(EPA.NC.TOTAL.Summary)

## [1] 101      5
```

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

Answer: `drop_na` allows us to remove rows with NAs in specific columns, while `na.omit` is only able to remove rows with NAs in any column. For example, if we wanted to only remove rows with NAs in Ozone, we could not use `na.omit` because it would also remove rows with NAs in PM2.5