

Assignment 4: Data Wrangling

Natasha Jacob

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.

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE, echo = T)
```

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 Checking working directory
getwd()
```

```
## [1] "/Users/admin/Desktop/ENV872_EDA/Environmental_Data_Analytics_2022/Assignments"
```

```
# Loading the required packages
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5    v purrr   0.3.4
## v tibble  3.1.6    v dplyr   1.0.7
## v tidyr   1.1.4    v stringr 1.4.0
## v readr   2.1.1    v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
## date, intersect, setdiff, union
```

```
library(dplyr)
```

```
# Uploading the four raw data files associated with the EPA
# air dataset
EPA_03_2018 <- read.csv("../Data/Raw/EPAair_03_NC2018_raw.csv",
  stringsAsFactors = TRUE)
EPA_03_2019 <- read.csv("../Data/Raw/EPAair_03_NC2019_raw.csv",
  stringsAsFactors = TRUE)
EPA_PM25_2018 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv",
  stringsAsFactors = TRUE)
EPA_PM25_2019 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv",
  stringsAsFactors = TRUE)

# 2 Exploring the dimensions, column names, and structure
# of the datasets. EPA 2018 ozone dataset
dim(EPA_03_2018)
```

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

```
colnames(EPA_03_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"
```

```
str(EPA_03_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 ...
## $ 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 ...
## $ 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 ...
```

```
# EPA 2019 ozone dataset
dim(EPA_03_2019)
```

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

```
colnames(EPA_03_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] "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"
```

```
str(EPA_03_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
## $ 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 ...
```

```
# EPA PM2.5 2018 dataset
dim(EPA_PM25_2018)
```

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

```
colnames(EPA_PM25_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"
```

```
str(EPA_PM25_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
## $ 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
## $ 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 ...
```

```
# EPA PM2.5 2019 dataset
dim(EPA_PM25_2019)
```

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

```
colnames(EPA_PM25_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_PM25_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
## $ 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
## $ 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
```

```
## $ 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 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 Changing the date to a date object
EPA_03_2018$Date <- as.Date(EPA_03_2018$Date, format = "%m/%d/%Y")
EPA_03_2019$Date <- as.Date(EPA_03_2019$Date, format = "%m/%d/%Y")
EPA_PM25_2018$Date <- as.Date(EPA_PM25_2018$Date, format = "%m/%d/%Y")
EPA_PM25_2019$Date <- as.Date(EPA_PM25_2019$Date, format = "%m/%d/%Y")

# 4 Selecting the required columns
EPA_03_2018 <- select(EPA_03_2018, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_03_2019 <- select(EPA_03_2019, Date, DAILY_AQI_VALUE, Site.Name,
  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_PM25_2018 <- select(EPA_PM25_2018, Date, DAILY_AQI_VALUE,
  Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPA_PM25_2019 <- select(EPA_PM25_2019, Date, DAILY_AQI_VALUE,
  Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

# 5 Using the mutate function to fill all the cells in
# AQS_PARAMETER_DESC with 'PM2.5'
EPA_PM25_2018$AQS_PARAMETER_DESC <- "PM2.5"
EPA_PM25_2019$AQS_PARAMETER_DESC <- "PM2.5"

# 6 Saving all processed datasets in the processed folder
write.csv(EPA_03_2018, row.names = FALSE, file = "../Data/Processed/EPAair_03_NC2018_Processed.csv")
write.csv(EPA_03_2019, row.names = FALSE, file = "../Data/Processed/EPAair_03_NC2019_Processed.csv")
write.csv(EPA_PM25_2018, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2018_Processed.csv")
write.csv(EPA_PM25_2019, row.names = FALSE, file = "../Data/Processed/EPAair_PM25_NC2019_Processed.csv")

# Checking to see if the date still remains as a date
# object
class(EPA_03_2018$Date)

## [1] "Date"
```

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 Using rbind to combine the four processed datasets
```

```
EPA_air_dataset <- rbind(EPA_O3_2018, EPA_O3_2019, EPA_PM25_2018,  
  EPA_PM25_2019)  
class(EPA_air_dataset$Date)
```

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

```
# 8 Wrangling the new dataset using a pipe function
```

```
EPAair_wrangled_data <- EPA_air_dataset %>%  
  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(AQIMean = mean(EPA_air_dataset$DAILY_AQI_VALUE),  
    LatitudeMean = mean(EPA_air_dataset$SITE_LATITUDE),  
    LongitudeMean = mean(EPA_air_dataset$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.
```

```
# Checking dimensions of EPAair_wrangled_data
```

```
dim(EPAair_wrangled_data)
```

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

```
# 9 Spreading the dataset such that AQI values for ozone
# and PM2.5 are in separate columns
EPAair_wrangled_data_spread <- spread(EPAair_wrangled_data, key = AQS_PARAMETER_DESC,
  value = AQIMean)
```

```
# 10 Checking dimensions of EPAair_wrangled_data_spread
dim(EPAair_wrangled_data_spread)
```

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

```
# 11 Saving EPAair_wrangled_data_spread as a processed file
write.csv(EPAair_wrangled_data_spread, row.names = FALSE, file = "../Data/Processed/EPAair_03_PM25_NC211")
```

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) Using the split-apply-combine strategy to
# generate a summary data frame grouped by site, month, and
# year and generating the mean AQI values for ozone and
# PM2.5 for each group. BONUS: A piped statement that
# removes rows where both mean ozone and mean PM2.5 have
# missing values was added
```

```
Final_summary <- EPAair_wrangled_data_spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(mean_ozone = mean(Ozone), mean_PM25 = mean(PM2.5)) %>%
  filter(!is.na(mean_ozone), !is.na(mean_PM25))
```

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

```
# 13 Checking dimensions of the new dataset
dim(Final_summary)
```

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

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

Answer: `drop_na` cannot be applied to numeric objects (in this case, mean). Hence, the `!is.na` function was used to remove NA's.