

Assignment 4: Data Wrangling

Eva May

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/evamay/Desktop/ENV872/Environmental.Data.Analytics.2021/Assignments"

library(tidyverse)
library(lubridate)
library(stringi)
Ozone.18 <- read.csv("../Data/Raw/EPAair_03_NC2018_raw.csv", header=TRUE)
Ozone.19 <- read.csv("../Data/Raw/EPAair_03_NC2019_raw.csv", header=TRUE)
PM.18 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv", header=TRUE)
PM.19 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv", header=TRUE)

#2
dim(Ozone.18)

## [1] 9737 20

colnames(Ozone.18)

## [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(Ozone.18)
```

```
## 'data.frame': 9737 obs. of 20 variables:
## $ Date : chr "03/01/2018" "03/02/2018" "03/03/2018" "03/04/2018" ..
## $ Source : chr "AQ5" "AQ5" "AQ5" "AQ5" ...
## $ 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 0.049 ...
## $ UNITS : chr "ppm" "ppm" "ppm" "ppm" ...
## $ DAILY_AQI_VALUE : int 40 43 44 45 44 28 33 41 45 40 ...
## $ Site.Name : chr "Taylorsville Liledoun" "Taylorsville Liledoun" "Taylorsville Liledoun" ...
## $ 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 ...
## $ AQ5_PARAMETER_CODE : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201 ...
## $ AQ5_PARAMETER_DESC : chr "Ozone" "Ozone" "Ozone" "Ozone" ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860 ...
## $ CBSA_NAME : chr "Hickory-Lenoir-Morganton, NC" "Hickory-Lenoir-Morganton, NC" ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : chr "North Carolina" "North Carolina" "North Carolina" "North Carolina" ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
## $ COUNTY : chr "Alexander" "Alexander" "Alexander" "Alexander" ...
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
dim(Ozone.19)
```

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

```
colnames(Ozone.19)
```

```
## [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] "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(Ozone.19)
```

```
## 'data.frame': 10592 obs. of 20 variables:
## $ Date : chr "01/01/2019" "01/02/2019" "01/03/2019" "01/04/2019" ..
## $ Source : chr "AirNow" "AirNow" "AirNow" "AirNow" ...
## $ 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 ...
## $ UNITS : chr "ppm" "ppm" "ppm" "ppm" ...
## $ DAILY_AQI_VALUE : int 27 17 15 20 34 34 27 35 35 28 ...
## $ Site.Name : chr "Taylorsville Liledoun" "Taylorsville Liledoun" "Taylorsville Liledoun" ...
## $ 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 : chr "Ozone" "Ozone" "Ozone" "Ozone" ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860 ...
## $ CBSA_NAME : chr "Hickory-Lenoir-Morganton, NC" "Hickory-Lenoir-Morganton, NC" ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : chr "North Carolina" "North Carolina" "North Carolina" "North Carolina" ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
## $ COUNTY : chr "Alexander" "Alexander" "Alexander" "Alexander" ...
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
dim(PM.18)
```

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

```
colnames(PM.18)
```

```
## [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(PM.18)
```

```
## 'data.frame': 8983 obs. of 20 variables:
## $ Date : chr "01/02/2018" "01/05/2018" "01/08/2018" "01/11/2018" ...
## $ Source : chr "AQS" "AQS" "AQS" "AQS" ...
## $ 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 : chr "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" ...
## $ DAILY_AQI_VALUE : int 12 15 22 3 10 19 8 10 18 7 ...
## $ Site.Name : chr "Linville Falls" "Linville Falls" "Linville Falls" "Linville Falls" ...
## $ 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 : chr "Acceptable PM2.5 AQI & Speciation Mass" "Acceptable PM2.5 AQI & Speciation Mass" ...
## $ 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" "North Carolina" "North Carolina" ...
## $ 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 ...
```

```
dim(PM.19)
```

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

```
colnames(PM.19)
```

```
## [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(PM.19)
```

```
## 'data.frame': 8581 obs. of 20 variables:
## $ Date : chr "01/03/2019" "01/06/2019" "01/09/2019" "01/12/2019" ...
## $ Source : chr "AQS" "AQS" "AQS" "AQS" ...
## $ 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 : chr "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" "ug/m3 LC" ...
## $ DAILY_AQI_VALUE : int 7 4 5 26 11 5 6 6 15 7 ...
## $ Site.Name : chr "Linville Falls" "Linville Falls" "Linville Falls" "Linville Falls" ...
## $ 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 : chr "Acceptable PM2.5 AQI & Speciation Mass" "Acceptable PM2.5 AQI & Speciation Mass" ...
## $ 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" "North Carolina" "North Carolina" ...
## $ 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(Ozone.18$Date)

## [1] "character"
#date format as character is mm/dd/YYYY
Ozone.18$Date <- as.Date(Ozone.18$Date, format = "%m/%d/%Y")
class(Ozone.18$Date)

## [1] "Date"
class(Ozone.19$Date)

## [1] "character"
#date format as character is mm/dd/YYYY
Ozone.19$Date <- as.Date(Ozone.19$Date, format = "%m/%d/%Y")
class(Ozone.19$Date)

## [1] "Date"
class(PM.18$Date)

## [1] "character"
#date format as character is mm/dd/YYYY
PM.18$Date <- as.Date(PM.18$Date, format = "%m/%d/%Y")
class(PM.18$Date)

## [1] "Date"
class(PM.19$Date)

## [1] "character"
#date format as character is mm/dd/YYYY
PM.19$Date <- as.Date(PM.19$Date, format = "%m/%d/%Y")
class(PM.19$Date)

## [1] "Date"

#4
Ozone.18.b <- select(Ozone.18, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                     COUNTY, SITE_LATITUDE:SITE_LONGITUDE)

Ozone.19.b <- select(Ozone.19, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                     COUNTY, SITE_LATITUDE:SITE_LONGITUDE)
```

```

PM.18.b <- select(PM.18, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                  COUNTY, SITE_LATITUDE:SITE_LONGITUDE)

PM.19.b <- select(PM.19, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                  COUNTY, SITE_LATITUDE:SITE_LONGITUDE)

#5
length(PM.18.b$AQS_PARAMETER_DESC)

## [1] 8983

PM.18.b$AQS_PARAMETER_DESC[1:8983] <- "PM2.5"

length(PM.19.b$AQS_PARAMETER_DESC)

## [1] 8581

PM.19.b$AQS_PARAMETER_DESC[1:8581] <- "PM2.5"

#6
write.csv(Ozone.18.b, row.names=FALSE, file = "../Data/Processed/EPAair_O3_NC2018_processed.csv")

write.csv(Ozone.19.b, row.names=FALSE, file = "../Data/Processed/EPAair_O3_NC2019_processed.csv")

write.csv(PM.18.b, row.names=FALSE, file = "../Data/Processed/EPAair_PM2.5_NC2018_processed.csv")

write.csv(PM.19.b, row.names=FALSE, file = "../Data/Processed/EPAair_PM2.5_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
EPA.18.19 <- rbind(Ozone.18.b, Ozone.19.b, PM.18.b, PM.19.b)

#8
class(EPA.18.19$Site.Name)

## [1] "character"

Ozone.18.b$Site.Name <- as.factor(Ozone.18.b$Site.Name)
Ozone.19.b$Site.Name <- as.factor(Ozone.19.b$Site.Name)
PM.18.b$Site.Name <- as.factor(PM.18.b$Site.Name)

```

```
PM.19.b$Site.Name <- as.factor(PM.19.b$Site.Name)
site1 <- Ozone.18.b$Site.Name
site2 <- Ozone.19.b$Site.Name
site3 <- PM.18.b$Site.Name
site4 <- PM.19.b$Site.Name
```

```
intr1 <- intersect(site1, site2)
intr2 <- intersect(site3, site4)
intr <- intersect(intr1, intr2)
intr
```

```
## [1] "Linville Falls"      "Durham Armory"      "Leggett"
## [4] "Hattie Avenue"       "Clemmons Middle"   "Mendenhall School"
## [7] "Frying Pan Mountain" "West Johnston Co."  "Garinger High School"
## [10] "Castle Hayne"        "Pitt Agri. Center"  "Bryson City"
## [13] ""                    "Millbrook School"
```

```
intr <- stri_remove_empty(intr)
EPA.18.19.w <- EPA.18.19 %>%
  filter(Site.Name %in% intr) %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlat = mean(SITE_LATITUDE),
            meanlong = mean(SITE_LONGITUDE))
```

```
## `summarise()` regrouping output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC' (override with `.groups`
#getting an override error here but doesn't seem to negatively impact output so ignoring it
EPA.18.19.w$Month <- month(EPA.18.19.w$date)
EPA.18.19.w$Year <- year(EPA.18.19.w$date)
#9
#should I be using above wrangled dataset or separate ozone and PM datasets here?
EPA.18.19.w2 <- pivot_wider(EPA.18.19.w, names_from = AQS_PARAMETER_DESC, values_from = meanAQI)
#10
dim(EPA.18.19.w2)
```

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

```
#11
```

```
write.csv(EPA.18.19.w2, row.names=FALSE, file = "../Data/Processed/EPAair_03_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
EPA.wrangled.sum <- EPA.18.19.w2 %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(Mean.AQI.O3 = mean(Ozone), Mean.AQI.PM = mean(PM2.5))
```

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

#12b

```
EPA.wrangled.sum1 <- EPA.wrangled.sum %>%  
  na.omit(Month:Year)
```

```
EPA.wrangled.sum2 <- EPA.wrangled.sum %>%  
  drop_na(Month:Year)
```

#13

```
dim(EPA.wrangled.sum2)
```

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

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

Answer: Here, the original summary dataframe without using any NA functions has 308 rows. There are NA values in columns other than Month and Year, but none in those two date columns. When we use `na.omit`, even though we can write code to try to specify that it should be used only on the Month and Year columns, we end up removing all rows with any values of NA, regardless of which column the NAs are in (leaving us with 101 rows). Conversely, when we use `drop_na`, we are able to target the two specific columns, leaving us with the same number of rows as the original dataframe. `Na.omit` removes all rows with NA values from the dataframe, while `drop_na` only removes rows with NA values in the columns specified in the code.