

Assignment 4: Data Wrangling

Mandy Hooks

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
library(tidyverse)
library(lubridate)
ozone.2018<-read.csv("/Users/mandyhooks/Environmental_Data_Analytics_2021/Data/Raw/EPAair_03_NC2018_raw
ozone.2019<-read.csv("/Users/mandyhooks/Environmental_Data_Analytics_2021/Data/Raw/EPAair_03_NC2019_raw
PM25.2018<-read.csv("/Users/mandyhooks/Environmental_Data_Analytics_2021/Data/Raw/EPAair_PM25_NC2018_ra
PM25.2019<-read.csv("/Users/mandyhooks/Environmental_Data_Analytics_2021/Data/Raw/EPAair_PM25_NC2019_ra
#2
```

```
#Ozone 2018
colnames(ozone.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] "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.2018)
```

```
## 'data.frame': 9737 obs. of 20 variables:
## $ Date : chr "03/01/2018" "03/02/2018" "03/03/2018" "03/04/2018" ..
## $ Source : chr "AQS" "AQS" "AQS" "AQS" ...
## $ 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 ...
## $ 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(ozone.2018)
```

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

```
#Ozone 2019
```

```
colnames(ozone.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(ozone.2019)
```

```
## '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 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
dim(ozone.2019)
```

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

```
#PM25 2018
```

```
colnames(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(PM25.2018)
```

```
## '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
## $ 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 A
## $ 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 Ca
## $ 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(PM25.2018)
```

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

```
#PM 2019
```

```
colnames(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(PM25.2019)
```

```
## '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
## $ 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 A
## $ 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 Ca
## $ 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(PM25.2019)
```

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

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
```

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

```
#4
```

```
ozone.2018.proc<- ozone.2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
ozone.2019.proc<- ozone.2019 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25.2018.proc<- PM25.2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25.2019.proc<- PM25.2019 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
```

```
#5
```

```
PM25.2018.proc<- mutate(PM25.2018.proc, AQS_PARAMETER_DESC = "PM2.5")
PM25.2019.proc<- mutate(PM25.2019.proc, AQS_PARAMETER_DESC = "PM2.5")
```

```
#6
```

```
write.csv(ozone.2018.proc, row.names = FALSE, file="/Users/mandyhooks/Environmental_Data_Analytics_2021/
write.csv(ozone.2019.proc, row.names = FALSE, file="/Users/mandyhooks/Environmental_Data_Analytics_2021/
write.csv(PM25.2018.proc, row.names = FALSE, file="/Users/mandyhooks/Environmental_Data_Analytics_2021/
write.csv(PM25.2019.proc, row.names = FALSE, file="/Users/mandyhooks/Environmental_Data_Analytics_2021/
```

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
EPAcombined <- rbind(ozone.2018.proc, ozone.2019.proc, PM25.2018.proc, PM25.2019.proc)

#8
EPAcombined.proc <- EPAcombined %>%
  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(meanAQI = mean(DAILY_AQI_VALUE),
            meanLatitude = mean(SITE_LATITUDE),
            meanLongitude = 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 `ungroup()`.
EPAcombined.proc$Date<-as.Date(EPAcombined.proc$Date, format = "%m/%d/%Y")

class(Date)

## [1] "function"

#9
EPAcombined.proc.pivot <- spread(EPAcombined.proc, AQS_PARAMETER_DESC, meanAQI)

#10
dim(EPAcombined.proc)

## [1] 14752      9

#11
write.csv(EPAcombined.proc.pivot, row.names = FALSE, file="/Users/mandyhooks/Environmental_Data_Analyti
```

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
EPAcombined.proc.pivot.summaries <-
  EPAcombined.proc.pivot %>%
  group_by(Site.Name, month, year) %>%
```

```

summarise(meanOzone = mean(Ozone),
           meanPM25 = mean(PM2.5))

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

#12b
EPAcombined.proc.pivot.summaries.drop<-
  EPAcombined.proc.pivot.summaries%>%
  drop_na(month, year)

#view(EPAcombined.proc.pivot.summaries.drop)

#13
dim(EPAcombined.proc.pivot.summaries.drop)

## [1] 308    5

```

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

Answer: `na.omit` drops the rows but not the row numbers, which could create problems with handling the data. `Drop_na` drops the row if there is missing data, so in the end all columns will have the same number of rows