Assignment 4: Data Wrangling

Monisha Eadala

## 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., “Salk\_A04\_DataWrangling.Rmd”) prior to submission.

The completed exercise is due on Tuesday, February 4 at 1:00 pm.

## 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() # Checks the working directory

## [1] "/Users/monishaeadala/Environmental\_Data\_Analytics\_2020"

library(tidyverse) # Loads 'tidyverse'  
library(lubridate) # Loads `lubridate`  
  
EPA.Air.O3.2018 <- read.csv("./Data/Raw/EPAair\_O3\_NC2018\_raw.csv") # Imports the file  
EPA.Air.O3.2019 <- read.csv("./Data/Raw/EPAair\_O3\_NC2019\_raw.csv") # Imports the file  
EPA.Air.PM25.2018 <- read.csv("./Data/Raw/EPAair\_PM25\_NC2018\_raw.csv") # Imports the file  
EPA.Air.PM25.2019 <- read.csv("./Data/Raw/EPAair\_PM25\_NC2019\_raw.csv") # Imports the file  
  
#2  
dim(EPA.Air.O3.2018) # Checks the dimensions

## [1] 9737 20

colnames(EPA.Air.O3.2018) # Checks the column names

## [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.Air.O3.2018) # Checks the structure of the dataset

## 'data.frame': 9737 obs. of 20 variables:  
## $ Date : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62 63 64 65 66 67 68 69 ...  
## $ 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 0.043 ...  
## $ 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 ...

dim(EPA.Air.O3.2019) # Checks the dimensions

## [1] 10592 20

colnames(EPA.Air.O3.2019) # Checks the column names

## [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.Air.O3.2019) # Checks the structure of the dataset

## 'data.frame': 10592 obs. of 20 variables:  
## $ Date : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ 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 0.03 ...  
## $ 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 ...

dim(EPA.Air.PM25.2018) # Checks the dimensions

## [1] 8983 20

colnames(EPA.Air.PM25.2018) # Checks the column names

## [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.Air.PM25.2018) # Checks the structure of the dataset

## 'data.frame': 8983 obs. of 20 variables:  
## $ Date : Factor w/ 365 levels "01/01/2018","01/02/2018",..: 2 5 8 11 14 17 20 23 26 29 ...  
## $ 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 ...

dim(EPA.Air.PM25.2019) # Checks the dimensions

## [1] 8581 20

colnames(EPA.Air.PM25.2019) # Checks the column names

## [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.Air.PM25.2019) # Checks the structure of the dataset

## 'data.frame': 8581 obs. of 20 variables:  
## $ Date : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 3 6 9 12 15 18 21 24 27 30 ...  
## $ 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 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.

1. Change date to date
2. Select the following columns: Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE
3. For the PM2.5 datasets, fill all cells in AQS\_PARAMETER\_DESC with “PM2.5” (all cells in this column should be identical).
4. 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(EPA.Air.O3.2018$Date) # Checks the class prior to the changing

## [1] "factor"

EPA.Air.O3.2018$Date <- as.Date(EPA.Air.O3.2018$Date, format = "%m/%d/%Y") # Changes date to date in EPA.Air.O3.2018  
class(EPA.Air.O3.2018$Date) # Checks the class after changing the Date to date

## [1] "Date"

class(EPA.Air.O3.2019$Date) # Checks the class prior to the changing

## [1] "factor"

EPA.Air.O3.2019$Date <- as.Date(EPA.Air.O3.2019$Date, format = "%m/%d/%Y") # Changes date to date in EPA.Air.O3.2019  
class(EPA.Air.O3.2019$Date) # Checks the class after changing the Date to date

## [1] "Date"

class(EPA.Air.PM25.2018$Date) # Checks the class prior to the changing

## [1] "factor"

EPA.Air.PM25.2018$Date <- as.Date(EPA.Air.PM25.2018$Date, format = "%m/%d/%Y") # Changes date to date in EPA.Air.PM25.2018  
class(EPA.Air.PM25.2018$Date) # Checks the class after changing the Date to date

## [1] "Date"

class(EPA.Air.PM25.2019$Date) # Checks the class prior to the changing

## [1] "factor"

EPA.Air.PM25.2019$Date <- as.Date(EPA.Air.PM25.2019$Date, format = "%m/%d/%Y") # Changes date to date in EPA.Air.PM25.2019  
class(EPA.Air.PM25.2019$Date) # Checks the class after changing the Date to date

## [1] "Date"

#4  
EPA.Air.O3.2018.select <- select(EPA.Air.O3.2018, Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE) # Selects Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE from EPA.Air.O3.2018  
  
EPA.Air.O3.2019.select <- select(EPA.Air.O3.2019, Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE) # Selects Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE from EPA.Air.O3.2019  
  
EPA.Air.PM25.2018.select <- select(EPA.Air.PM25.2018, Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE) # Selects Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE from EPA.Air.PM25.2018  
  
EPA.Air.PM25.2019.select <- select(EPA.Air.PM25.2019, Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE) # Selects Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE from EPA.Air.PM25.2018  
  
#5  
EPA.Air.PM25.2018.select <- mutate(EPA.Air.PM25.2018.select, AQS\_PARAMETER\_DESC = "PM2.5") # Fills all cells in AQS\_PARAMETER\_DESC with "PM2.5" in EPA.Air.PM25.2018.select  
  
EPA.Air.PM25.2019.select <- mutate(EPA.Air.PM25.2019.select, AQS\_PARAMETER\_DESC = "PM2.5") # Fills all cells in AQS\_PARAMETER\_DESC with "PM2.5" in EPA.Air.PM25.2019.select  
  
#6  
write.csv(EPA.Air.O3.2018.select, row.names = FALSE, file = "./Data/Processed/EPAair\_O3\_NC2018\_processed.csv") # Saved the EPA.Air.O3.2018.select dataset in the Processed folder by the name "EPAair\_O3\_NC2018\_processed"  
  
write.csv(EPA.Air.O3.2019.select, row.names = FALSE, file = "./Data/Processed/EPAair\_O3\_NC2019\_processed.csv") # Saved the EPA.Air.O3.2019.select dataset in the Processed folder by the name "EPAair\_O3\_NC2019\_processed"  
  
write.csv(EPA.Air.PM25.2018.select, row.names = FALSE, file = "./Data/Processed/EPAair\_PM25\_NC2018\_processed.csv") # Saved the EPA.Air.PM25.2018.select dataset in the Processed folder by the name "EPAair\_PM25\_NC2018\_processed"  
  
write.csv(EPA.Air.PM25.2019.select, row.names = FALSE, file = "./Data/Processed/EPAair\_PM25\_NC2019\_processed.csv") # Saved the EPA.Air.PM25.2019.select dataset in the Processed folder by the name "EPAair\_PM25\_NC2019\_processed"

## Combine datasets

1. Combine the four datasets with rbind. Make sure your column names are identical prior to running this code.
2. 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.

1. 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.
2. Call up the dimensions of your new tidy dataset.
3. Save your processed dataset with the following file name: “EPAair\_O3\_PM25\_NC1718\_Processed.csv”

#7  
EPA.Air.combined <- rbind(EPA.Air.O3.2018.select, EPA.Air.O3.2019.select, EPA.Air.PM25.2018.select, EPA.Air.PM25.2019.select) # Combines the four datasets  
  
#8  
 EPA.Air.combined.piped <-  
 EPA.Air.combined %>%  
 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") %>%  
 dplyr::group\_by(Date, Site.Name, AQS\_PARAMETER\_DESC, COUNTY) %>% # Groups by date, site, aqs parameter, and county  
 dplyr::summarise(meanAQI = mean(DAILY\_AQI\_VALUE),   
 meanLat = mean(SITE\_LATITUDE),  
 meanLong = mean(SITE\_LONGITUDE)) %>% # Takes the mean of the AQI value, latitude, and longitude.  
 mutate(Month = month(Date), Year = year(Date)) # makes new columns "Month" and "Year"  
dim(EPA.Air.combined.piped) # Checks the dimensions of the dataset; In this case they are 14752 by 9

## [1] 14752 9

#9  
EPA.Air.combined.piped.spread <- spread(EPA.Air.combined.piped, AQS\_PARAMETER\_DESC, meanAQI) # Spreads dataset into separate columns 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  
dim(EPA.Air.combined.piped.spread) # Checks the dimensions of the dataset; In this case it is 8976 by 9

## [1] 8976 9

#11  
write.csv(EPA.Air.combined.piped.spread, row.names = FALSE, file = "./Data/Processed/EPAair\_O3\_PM25\_NC1718\_Processed.csv") # Saves the EPA.Air.combined.piped dataset in the Processed folder by the name "EPAair\_O3\_PM25\_NC1718\_Processed.csv"

## Generate summary tables

1. 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).
2. Call up the dimensions of the summary dataset.

#12a  
EPA.Air.combined.piped.spread.summaries <-   
 EPA.Air.combined.piped.spread %>%   
 dplyr::group\_by(Site.Name, Month, Year) %>% # Groups the data by site, month, and year  
 dplyr::summarise(meanO3 = mean(Ozone),  
 meanPM25 = mean(PM2.5)) %>% # Generates the mean AQI values for ozone and PM2.5 for each group  
#12b  
 drop\_na(Month, Year) # Drops the rows where a month and year are not available   
  
#13  
dim(EPA.Air.combined.piped.spread.summaries) # Checks the dimensions of the dataset; In this case it is 308 by 5

## [1] 308 5

1. Why did we use the function drop\_na rather than na.omit?

Answer: ‘na.omit’ would omit all the NAs in the dataset, while in ‘drop\_na’ we can select the specific columns for the rows that contains NAs we want to drop. Thefore, we use the ‘drop\_na’ function since we want to remove instances where only the month and year are not available.