Assignment 4: Data Wrangling

Nancy Bao

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
#Check the working directory
getwd()
```

```
## [1] "/Users/Nancy/Desktop/Semester 4/ENV 872L/Environmental_Data_Analytics_2021"
```

```
#working directory is already set to the main folder:
#"/Users/Nancy/Desktop/Semester 4/ENV 872L/Environmental_Data_Analytics_2021"

#Load the packages
library(tidyverse)
library(lubridate)

#Import the four EPA air datasets; set relative file path
epa_air_2018<-read.csv("Data/Raw/EPAair_O3_NC2018_raw.csv",
                        stringsAsFactors = TRUE)
epa_air_2019<-read.csv("Data/Raw/EPAair_O3_NC2019_raw.csv",
                        stringsAsFactors = TRUE)
epa_air_PM2.5_2018<-read.csv("Data/Raw/EPAair_PM25_NC2018_raw.csv",
                              stringsAsFactors = TRUE)
epa_air_PM2.5_2019<-read.csv("Data/Raw/EPAair_PM25_NC2019_raw.csv",
                              stringsAsFactors = TRUE)
#2
```

```
## Explore dimensions
dim(epa_air_2018) # 9737 rows and 20 columns
```

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

```
dim(epa_air_2019) # 10592 rows and 20 columns
```

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

```
dim(epa_air_PM2.5_2018) # 8983 rows and 20 columns
```

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

```
dim(epa_air_PM2.5_2019) #8581 rows and 20 columns
```

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

```
## Explore column names
colnames(epa_air_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] "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"
```

```
colnames(epa_air_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"
```

```
colnames(epa_air_PM2.5_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_air_PM2.5_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"
```

```
#The PM2.5 datasets for 2018 and 2019 has  a column for Daily.Mean.PM2.5.Concentration
#where the remaining datasets instead, have a "Daily.Max.8.hour.Ozone.Concentration"column.


## Explore structures

#https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/str
# I used the URL above to read more on the strict.width and width option
# to figure out how to make the width of str() output fit page
# I wanted the output to fit 8.5x11 page, so I chose width 85

#
str(epa_air_2018,strict.width="cut",width=85)
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/"..
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1..
##  $ Site.ID                             : int  370030005 370030005 370030005 3700300..
##  $ 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...
##  $ UNITS                               : Factor w/ 1 level "ppm": 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 3..
##  $ 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 1..
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 4..
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1..
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 2..
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, NC",..:..
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 ..
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Avery",....
##  $ 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_air_2019,  strict.width="cut",width=85)
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                                : Factor w/ 365 levels "01/01/2019","01/02/"..
##  $ Source                              : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1..
##  $ Site.ID                             : int  370030005 370030005 370030005 3700300..
##  $ 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..
##  $ UNITS                               : Factor w/ 1 level "ppm": 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 3..
##  $ 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 1..
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 4..
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1..
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 2..
##  $ CBSA_NAME                           : Factor w/ 15 levels "","Asheville, NC",..:..
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 ..
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 30 levels "Alexander","Avery",....
##  $ 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_air_PM2.5_2018, strict.width="cut",width=85)
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2018",...
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370..
##  $ 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 ..
##  $ DAILY_AQI_VALUE               : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Blackstone",..: 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 8..
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI & Spec"..
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: 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 ..
##  $ COUNTY_CODE                 : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                      : Factor w/ 21 levels "Avery","Buncombe",..: 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_air_PM2.5_2019, strict.width="cut",width=85)
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                        : Factor w/ 365 levels "01/01/2019","01/02/2019",...
##  $ Source                      : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2..
##  $ Site.ID                     : int  370110002 370110002 370110002 370110002 370..
##  $ 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 ..
##  $ DAILY_AQI_VALUE             : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                   : Factor w/ 25 levels "","Board Of Ed. Bldg.",..: ..
##  $ 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 8..
##  $ AQS_PARAMETER_DESC          : Factor w/ 2 levels "Acceptable PM2.5 AQI & Spec"..
##  $ CBSA_CODE                   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                   : Factor w/ 14 levels "","Asheville, NC",..: 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 ..
##  $ COUNTY_CODE                 : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                      : Factor w/ 21 levels "Avery","Buncombe",..: 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 Change Date from factor variable to date variable with as.Date
class(epa_air_2018$Date)
```

```
## [1] "factor"
```

```
epa_air_2018$Date<-as.Date(epa_air_2018$Date, format="%m/%d/%Y")
class(epa_air_2018$Date) #now a date
```

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

```
#
class(epa_air_2019$Date)
```

```
## [1] "factor"
```

```
epa_air_2019$Date<-as.Date(epa_air_2019$Date, format="%m/%d/%Y")
class(epa_air_2019$Date)#now a date
```

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

```
#
class(epa_air_PM2.5_2018$Date)
```

```
## [1] "factor"
```

```
epa_air_PM2.5_2018$Date<-as.Date(epa_air_PM2.5_2018$Date, format="%m/%d/%Y")
class(epa_air_PM2.5_2018$Date)#now a date
```

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

```
#
class(epa_air_PM2.5_2019$Date)
```

```
## [1] "factor"
```

```
epa_air_PM2.5_2019$Date<-as.Date(epa_air_PM2.5_2019$Date, format="%m/%d/%Y")
class(epa_air_PM2.5_2019$Date)#now a date
```

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

```
#################
#################
#4 Select the following columns: Date, DAILY_AQI_VALUE,
#  Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE
epa_air_2018subset<-select(epa_air_2018,Date, DAILY_AQI_VALUE,
                           Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
epa_air_2019subset<-select(epa_air_2019,Date, DAILY_AQI_VALUE, Site.Name,
                           AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
epa_air_PM2.5_2018subset<-select(epa_air_PM2.5_2018,Date, DAILY_AQI_VALUE,
                                 Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)
epa_air_PM2.5_2019subset<-select(epa_air_PM2.5_2019,Date, DAILY_AQI_VALUE,
                                 Site.Name, AQS_PARAMETER_DESC, COUNTY:SITE_LONGITUDE)

#################
#################
#5 For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5"
epa_air_PM2.5_2018subset$AQS_PARAMETER_DESC<-"PM2.5"
epa_air_PM2.5_2019subset$AQS_PARAMETER_DESC<-"PM2.5"

#changed the subsets to factors
epa_air_PM2.5_2018subset$AQS_PARAMETER_DESC<-
  as.factor(epa_air_PM2.5_2018subset$AQS_PARAMETER_DESC)
epa_air_PM2.5_2019subset$AQS_PARAMETER_DESC<-
  as.factor(epa_air_PM2.5_2019subset$AQS_PARAMETER_DESC)
class(epa_air_PM2.5_2018subset$AQS_PARAMETER_DESC)
```

```
## [1] "factor"
```

```
class(epa_air_PM2.5_2019subset$AQS_PARAMETER_DESC)
```

```
## [1] "factor"
```

```
#################
#################
#6 Save all four processed datasets in the Processed folder
#2018 EPA air data
write.csv(epa_air_2018subset, row.names = FALSE,
         file= "Data/Processed/EPAair_O3_NC2018_Processed.csv")

#2019 EPA air data
write.csv(epa_air_2019subset, row.names= FALSE,
         file = "Data/Processed/EPAair_O3_NC2019_Processed.csv")

#2018 EPA PM2.5 data
write.csv(epa_air_PM2.5_2018subset, row.names=FALSE,
         file="Data/Processed/EPAair_PM25_NC2018_Processed.csv")

#2019 EPA PM2.5 data
write.csv(epa_air_PM2.5_2019subset, row.names=FALSE,
         file="Data/Processed/EPAair_PM25_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 Combine the four datasets with `rbind`
#Import processed EPA air data
epa2018<-read.csv("Data/Processed/EPAair_O3_NC2018_Processed.csv",
                  stringsAsFactors = TRUE)
epa2019<-read.csv("Data/Processed/EPAair_O3_NC2019_Processed.csv",
                  stringsAsFactors = TRUE)
epa2018_pm25<-read.csv("Data/Processed/EPAair_PM25_NC2018_Processed.csv",
                       stringsAsFactors = TRUE)
epa2019_pm25<-read.csv("Data/Processed/EPAair_PM25_NC2019_Processed.csv",
                       stringsAsFactors = TRUE)

#rbind()
EPA_air_2018thru2019<-rbind(epa2018, epa2018_pm25,
                            epa2019, epa2019_pm25)
class(EPA_air_2018thru2019$Date) #class is originally factor
```

```
## [1] "factor"
```

```
#change class to date with as.Date()
EPA_air_2018thru2019$Date<-
```

```
  as.Date(EPA_air_2018thru2019$Date,format="%Y-%m-%d")
class(EPA_air_2018thru2019$Date) #Now class is date
```

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

```
#
class(EPA_air_2018thru2019$DAILY_AQI_VALUE) #class:integer
```

```
## [1] "integer"
```

```
class(EPA_air_2018thru2019$SITE_LATITUDE) #class:numeric
```

```
## [1] "numeric"
```

```
class(EPA_air_2018thru2019$SITE_LONGITUDE) #class: numeric
```

```
## [1] "numeric"
```

```
#######################################################
#8 Wrangle new dataset with pipeline
EPA_air_total_processed <- EPA_air_2018thru2019 %>%
  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(mean_dailyAQI=mean(DAILY_AQI_VALUE),
            mean_Latitude=mean(SITE_LATITUDE),
            mean_Longitude=mean(SITE_LONGITUDE)) %>%
  mutate(Month=month(Date), Year=year(Date))
```

```
## `summarise()` has grouped output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC'. You can override usin
```

```
dim(EPA_air_total_processed) #now the dimensions are 14752 x 9
```

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

```
#9
EPA_air_processed_spread<- pivot_wider(EPA_air_total_processed,
                  names_from = AQS_PARAMETER_DESC, values_from = mean_dailyAQI)

#10 Call up the dimensions of your new tidy dataset.
dim(EPA_air_processed_spread) #8976 x 9
```

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

```
#11 Save your processed dataset.
#Instead of NC1718, I named it NC1819 to fit the years of the data
write.csv(EPA_air_processed_spread,
          row.names=FALSE,file="Data/Processed/EPAair_O3_PM25_NC1819_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
# I just used the dataframe where I spread the data: EPA_air_processed_spread
EPA_air_meanAQI_summary <- EPA_air_processed_spread %>%
  group_by( Site.Name, Month,Year) %>%
  summarise(mean_AQI_ozone =mean(Ozone),
            mean_AQI_PM2.5 =mean(PM2.5))
```

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

```
#12b #I used drop_na for any mean AQI values that did not exist for the specified months and years
EPA_air_meanAQI_summary_cleaned <- EPA_air_meanAQI_summary %>%
  drop_na(mean_AQI_ozone) %>%
  drop_na(mean_AQI_PM2.5)

#13
dim(EPA_air_meanAQI_summary_cleaned)
```

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

```
#14
#?drop_na - I read the R description
#?na.omit - Iread the R description
#I commented the pipe out to test out the na.omit function
#EPA_air_meanAQI_summary_clean <- EPA_air_meanAQI_summary %>%
 # na.omit(mean_AQI_ozone) #>%>
 #na.omit(mean_AQI_PM2.5)
#when I exclude the pipe, the missing values in PM2.5 also get deleted
#so na.omit doesn't specify removing values by column
```

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

    Answer: When I used both functions on the summary data, they provided the same dimensions. I think we used drop_na instead of na.omit because you can specify which rows to remove based on the specified column with drop_na, whereas other columns that may have missing values will get deleted when using na.omit. Since in this dataframe, all the missing values were in the mean AQI data, na.omit and drop_na produced the same dimensions. However, if there was a missing value in Site.Name, it may recognize that missing value and delete that entire row. So drop_na just removes NAs based on the column you want, not other columns as well. I tried this and deleted the na.omit(mean_AQI_PM2.5) pipe from the pipe I tried for #14 and found that even just with the na.omit(mean_AQI_ozone), it deleted missing values for both columns. So na.omit, just deletes all the missing values and cannot specify the missing values by column like with drop_na.