Assignment 4: Data Wrangling

Brian Mulu Mutua

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A04_DataWrangling.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. Ensure that code in code chunks does not extend off the page in the PDF.

The completed exercise is due on Thursday, Sept 28th @ 5:00pm.

Set up your session

1a. Load the `tidyverse`, `lubridate`, and `here` packages into your session.

1b. Check your working directory.

1c. Read in all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Apply the `glimpse()` function to reveal the dimensions, column names, and structure of each dataset.

```r
#1a Loading necessary packages
library(tidyverse)
library(lubridate)
library(here)

#1b Checking working directory
getwd()
```

```
## [1] "C:/Users/bmm100/Documents/EDE_Fall2023"
```

```
#1c Reading EPA data files
#Importing Ozone dataset for North Carolina for 2018
OzoneNC2018 <- read.csv(file=here("Data/Raw/EPAair_O3_NC2018_raw.csv"), stringsAsFactors
↪  = TRUE)
#Displaying first 3 rows and first 2 columns of the 2018 Ozone dataframe to show
↪  successful import
head(OzoneNC2018,c(3,2))
```

```
##          Date Source
## 1 03/01/2018    AQS
## 2 03/02/2018    AQS
## 3 03/03/2018    AQS
```

```
#Importing Ozone dataset for North Carolina for 2019
OzoneNC2019 <- read.csv(file=here("Data/Raw/EPAair_O3_NC2019_raw.csv"), stringsAsFactors
↪  = TRUE)
#Displaying first 3 rows and first 2 columns of the 2019 Ozone dataframe to show
↪  successful import
head(OzoneNC2019,c(3,2))
```

```
##          Date Source
## 1 01/01/2019 AirNow
## 2 01/02/2019 AirNow
## 3 01/03/2019 AirNow
```

```
#Importing PM2.5 dataset for North Carolina for 2018
PM25NC2018 <- read.csv(file=here("Data/Raw/EPAair_PM25_NC2018_raw.csv"), stringsAsFactors
↪  = TRUE)
#Displaying first 3 rows and first 2 columns of the 2018 Ozone dataframe to show
↪  successful import
head(PM25NC2018,c(3,2))
```

```
##          Date Source
## 1 01/02/2018    AQS
## 2 01/05/2018    AQS
## 3 01/08/2018    AQS
```

```
#Importing PM2.5 dataset for North Carolina for 2019
PM25NC2019 <- read.csv(file=here("Data/Raw/EPAair_PM25_NC2019_raw.csv"), stringsAsFactors
↪  = TRUE)
#Displaying first 3 rows and first 2 columns of the 2019 Ozone dataframe to show
↪  successful import
head(PM25NC2019,c(3,2))
```

```
##          Date Source
## 1 01/03/2019    AQS
## 2 01/06/2019    AQS
## 3 01/09/2019    AQS
```

```r
#2 Using glimpse function to reveal dimensions, column names and dataset structure
glimpse(OzoneNC2018)
```

```
## Rows: 9,737
## Columns: 20
## $ Date                                 <fct> 03/01/2018, 03/02/2018, 03/03/201~
## $ Source                               <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS~
## $ Site.ID                              <int> 370030005, 370030005, 370030005, ~
## $ POC                                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.043, 0.046, 0.047, 0.049, 0.047~
## $ UNITS                                <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                      <int> 40, 43, 44, 45, 44, 28, 33, 41, 4~
## $ Site.Name                            <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                      <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                     <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                   <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                   <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                            <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                            <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                           <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                                <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                          <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                               <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                        <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                       <dbl> -81.191, -81.191, -81.191, -81.19~
```

```r
glimpse(OzoneNC2019)
```

```
## Rows: 10,592
## Columns: 20
## $ Date                                 <fct> 01/01/2019, 01/02/2019, 01/03/201~
## $ Source                               <fct> AirNow, AirNow, AirNow, AirNow, A~
## $ Site.ID                              <int> 370030005, 370030005, 370030005, ~
## $ POC                                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.029, 0.018, 0.016, 0.022, 0.037~
## $ UNITS                                <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                      <int> 27, 17, 15, 20, 34, 34, 27, 35, 3~
## $ Site.Name                            <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                      <int> 24, 24, 24, 24, 24, 24, 24, 24, 2~
## $ PERCENT_COMPLETE                     <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                   <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                   <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                            <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                            <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                           <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                                <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                          <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                               <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                        <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                       <dbl> -81.191, -81.191, -81.191, -81.19~
```

```r
glimpse(PM25NC2018)
```

```
## Rows: 8,983
## Columns: 20
## $ Date                           <fct> 01/02/2018, 01/05/2018, 01/08/2018, 01/~
## $ Source                         <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                        <int> 370110002, 370110002, 370110002, 370110~
## $ POC                            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 2.9, 3.7, 5.3, 0.8, 2.5, 4.5, 1.8, 2.5,~
## $ UNITS                          <fct> 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, 24~
## $ Site.Name                      <fct> Linville Falls, Linville Falls, Linvill~
## $ DAILY_OBS_COUNT                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE               <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE             <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC             <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                      <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                     <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                          <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                    <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                         <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                  <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE                 <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

```r
glimpse(PM25NC2019)
```

```
## Rows: 8,581
## Columns: 20
## $ Date                           <fct> 01/03/2019, 01/06/2019, 01/09/2019, 01/~
## $ Source                         <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                        <int> 370110002, 370110002, 370110002, 370110~
## $ POC                            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 1.6, 1.0, 1.3, 6.3, 2.6, 1.2, 1.5, 1.5,~
## $ UNITS                          <fct> 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, 14, 20~
## $ Site.Name                      <fct> Linville Falls, Linville Falls, Linvill~
## $ DAILY_OBS_COUNT                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE               <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE             <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC             <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                      <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                     <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                          <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                    <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                         <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                  <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE                 <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

Wrangle individual datasets to create processed files.

3. Change the Date columns to be date objects.
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 date columns into date objects using lubridate
OzoneNC2018$Date <- mdy(OzoneNC2018$Date)
#Displaying class to demonstrate successful conversion
class(OzoneNC2018$Date)
```

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

```
OzoneNC2019$Date <- mdy(OzoneNC2019$Date)
#Displaying class to demonstrate successful conversion
class(OzoneNC2019$Date)
```

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

```
PM25NC2018$Date <- mdy(PM25NC2018$Date)
#Displaying class to demonstrate successful conversion
class(PM25NC2018$Date)
```

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

```
PM25NC2019$Date <- mdy(PM25NC2019$Date)
#Displaying class to demonstrate successful conversion
class(PM25NC2019$Date)
```

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

```
#4 Selecting specified columns for each dataset
OzoneNC2018.data <- select(OzoneNC2018, Date, DAILY_AQI_VALUE, Site.Name,
↪  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
OzoneNC2019.data <- select(OzoneNC2019, Date, DAILY_AQI_VALUE, Site.Name,
↪  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25NC2018.data <- select(PM25NC2018, Date, DAILY_AQI_VALUE, Site.Name,
↪  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25NC2019.data <- select(PM25NC2019, Date, DAILY_AQI_VALUE, Site.Name,
↪  AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

#5 Filling all cells in AQS_PARAMETER_DESC with PM2.5
PM25NC2018.data <- mutate(PM25NC2018.data, AQS_PARAMETER_DESC = "PM2.5")
glimpse(PM25NC2018.data$AQS_PARAMETER_DESC)
```

```
##  chr [1:8983] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" ...
```

```
PM25NC2019.data <- mutate(PM25NC2019.data, AQS_PARAMETER_DESC = "PM2.5")
glimpse(PM25NC2019.data$AQS_PARAMETER_DESC)
```

```
##  chr [1:8581] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" ...
```

```
#6 Saving all 4 processed datasets in the processed folder
write.csv(OzoneNC2018.data, row.names = FALSE, file =
↪  here("Data/Processed/EPAair_O3_NC2018_processed.csv"))
write.csv(OzoneNC2019.data, row.names = FALSE, file =
↪  here("Data/Processed/EPAair_O3_NC2019_processed.csv"))
write.csv(PM25NC2018.data, row.names = FALSE, file =
↪  here("Data/Processed/EPAair_PM25_NC2018_processed.csv"))
write.csv(PM25NC2019.data, row.names = FALSE, file =
↪  here("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 only 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 - but it will include sites with missing site information, which you don't want…)

- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site name, 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_NC1819_Processed.csv"

```
#7 Combining the columns using rbind
#Checking column names to confirm they are identical
colnames(OzoneNC2018.data)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(OzoneNC2019.data)
```

```
## [1] "Date"                "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(PM25NC2018.data)
```

```
## [1] "Date"                "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(PM25NC2019.data)
```

```
## [1] "Date"                "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
#Combining dataframes using rbind
NCAirQuality20182019.data <-
↪  rbind(OzoneNC2018.data,OzoneNC2019.data,PM25NC2018.data,PM25NC2019.data)
glimpse(NCAirQuality20182019.data)
```

```
## Rows: 37,893
## Columns: 7
## $ Date               <date> 2018-03-01, 2018-03-02, 2018-03-03, 2018-03-04, 20~
## $ DAILY_AQI_VALUE    <int> 40, 43, 44, 45, 44, 28, 33, 41, 45, 40, 31, 43, 42,~
## $ Site.Name          <fct> Taylorsville Liledoun, Taylorsville Liledoun, Taylo~
## $ AQS_PARAMETER_DESC <fct> Ozone, Ozone, Ozone, Ozone, Ozone, Ozone, Ozone, Oz~
## $ COUNTY             <fct> Alexander, Alexander, Alexander, Alexander, Alexand~
## $ SITE_LATITUDE      <dbl> 35.9138, 35.9138, 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE     <dbl> -81.191, -81.191, -81.191, -81.191, -81.191, -81.19~
```

```r
#8 Wrangling the dataset using the pipe function
NCAirQuality20182019.data <- NCAirQuality20182019.data %>%
  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),
            meanlat = mean(SITE_LATITUDE),
            meanlong = mean(SITE_LONGITUDE)) %>%
  mutate(Month = month(Date), Year = year(Date))
```

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

```
glimpse(NCAirQuality20182019.data)
```

```
## Rows: 14,752
## Columns: 9
## Groups: Date, Site.Name, AQS_PARAMETER_DESC [14,752]
## $ Date               <date> 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01, 20~
## $ Site.Name          <fct> Bryson City, Castle Hayne, Clemmons Middle, Durham ~
## $ AQS_PARAMETER_DESC <fct> PM2.5, PM2.5, PM2.5, PM2.5, Ozone, PM2.5, PM2.5, PM~
## $ COUNTY             <fct> Swain, New Hanover, Forsyth, Durham, Mecklenburg, M~
## $ meanAQI            <dbl> 35.0, 13.0, 24.0, 31.0, 32.0, 20.0, 22.0, 14.0, 34.~
## $ meanlat            <dbl> 35.43477, 34.36417, 36.02600, 36.03296, 35.24010, 3~
## $ meanlong           <dbl> -83.44213, -77.83861, -80.34200, -78.90404, -80.785~
## $ Month              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Year               <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 201~
```

```
#9 Spreading the dataset such that AQI values for ozone and PM2.5 are in separate columns
NCAirQuality20182019.data <- pivot_wider(NCAirQuality20182019.data, names_from =
↪  AQS_PARAMETER_DESC, values_from = meanAQI)
glimpse(NCAirQuality20182019.data)
```

```
## Rows: 8,976
## Columns: 9
## Groups: Date, Site.Name [8,976]
## $ Date      <date> 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01,~
## $ Site.Name <fct> Bryson City, Castle Hayne, Clemmons Middle, Durham Armory, G~
## $ COUNTY    <fct> Swain, New Hanover, Forsyth, Durham, Mecklenburg, Forsyth, E~
## $ meanlat   <dbl> 35.43477, 34.36417, 36.02600, 36.03296, 35.24010, 36.11069, ~
## $ meanlong  <dbl> -83.44213, -77.83861, -80.34200, -78.90404, -80.78568, -80.2~
## $ Month     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Year      <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, ~
## $ PM2.5     <dbl> 35.0, 13.0, 24.0, 31.0, 20.0, 22.0, 14.0, 28.0, 15.0, 24.0, ~
## $ Ozone     <dbl> NA, NA, NA, NA, 32, NA, NA, 34, NA, NA, NA, NA, NA, NA, NA, ~
```

```
#10 Calling up the dimensions of the newly processed dataset
dim(NCAirQuality20182019.data)
```

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

```
#11 Saving the processed dataset
write.csv(NCAirQuality20182019.data, row.names = FALSE, file =
↪  here("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 mean **ozone** values are not available (use the function `drop_na` in your pipe). It's ok to have missing mean PM2.5 values in this result.

13. Call up the dimensions of the summary dataset.

```
#12 Generating summary dataset
NCAirQuality20182019.datasummary <- NCAirQuality20182019.data %>%
  group_by(Site.Name,Month,Year) %>%
  summarise(meanOzone = mean(Ozone), meanPM25 = mean(PM2.5)) %>%
  drop_na(meanOzone)
```

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

```
glimpse(NCAirQuality20182019.datasummary)
```

```
## Rows: 182
## Columns: 5
## Groups: Site.Name, Month [109]
## $ Site.Name <fct> Bryson City, Bryson City, Bryson City, Bryson City, Bryson C~
## $ Month     <dbl> 3, 3, 4, 4, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 3, 4, 4, 5, 5, 6,~
## $ Year      <dbl> 2018, 2019, 2018, 2019, 2019, 2018, 2019, 2018, 2019, 2018, ~
## $ meanOzone <dbl> 41.58065, 42.51613, 44.53333, 45.40000, 39.60000, 37.80000, ~
## $ meanPM25  <dbl> 34.74194, NA, 28.16667, 26.73333, NA, NA, NA, NA, 33.64516, ~
```

```
#13 Calling up the dimensions of the summary dataset
dim(NCAirQuality20182019.datasummary)
```

```
## [1] 182   5
```

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

    Answer: ‘drop_na’ allows us to remove only the rows where the values in the mean ozone column are missing whereas ‘na.omit’ would omit all rows where a value is missing in any of the columns.