Assignment 4: Data Wrangling

David Robinson

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.

```
#1a

#Load necessary packages

#install.packages("tidyverse")
library(tidyverse)
#install.packages("lubridate")
library(lubridate)
#install.packages("here")
library(here)

#1b

#Check working directory #Please note decision to use 'here()' over 'getwd()' and 'setwd()'

here()
```

```
## [1] "C:/Users/dhr20/OneDrive - Duke University/1 - Academics/1 - First Year/1 - Fall 2023/2 - Enviro
```

```r
#1c

#Read in the data sets in question

Air_O3_2018 <- read.csv(file = here('Data','Raw','EPAair_O3_NC2018_raw.csv'),
                        stringsAsFactors = T)
Air_O3_2019 <- read.csv(file = here('Data','Raw','EPAair_O3_NC2019_raw.csv'),
                        stringsAsFactors = T)
Air_PM25_2018 <- read.csv(file = here('Data','Raw','EPAair_PM25_NC2018_raw.csv'),
                        stringsAsFactors = T)
Air_PM25_2019 <- read.csv(file = here('Data','Raw','EPAair_PM25_NC2019_raw.csv'),
                        stringsAsFactors = T)

#2

#Apply 'glimpse()'

glimpse(Air_O3_2018)
```

```
## 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(Air_O3_2019)
```

```
## 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~
```

```
glimpse(Air_PM25_2018)
```

```
## 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~
```

```
glimpse(Air_PM25_2019)
```

```
## 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".

```r
#3

#Change the Date columns to be date objects. Note decision to use 'as.Date()' but I
#understand that 'lubridate()' would also have worked here and may be easier

Air_O3_2018$Date <- as.Date(Air_O3_2018$Date, format = "%m/%d/%Y")

Air_O3_2019$Date <- as.Date(Air_O3_2019$Date, format = "%m/%d/%Y")

Air_PM25_2018$Date <- as.Date(Air_PM25_2018$Date, format = "%m/%d/%Y")

Air_PM25_2019$Date <- as.Date(Air_PM25_2019$Date, format = "%m/%d/%Y")

#4

#Select the columns in question

Air_O3_2018_Selection1 <- select(Air_O3_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COU

Air_O3_2019_Selection1 <- select(Air_O3_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COU

Air_PM25_2018_Selection1 <- select(Air_PM25_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,

Air_PM25_2019_Selection1 <- select(Air_PM25_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,

#5

#Fill all cells in AQS_PARAMETER_DESC with "PM2.5"
```

```
Air_PM25_2018_Selection1_replaced <-
  Air_PM25_2018_Selection1 %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

Air_PM25_2019_Selection1_replaced <-
  Air_PM25_2019_Selection1 %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

#6

#Save all four processed datasets in the Processed folder

write.csv(Air_O3_2018_Selection1, row.names = FALSE, file = here('Data','Processed','Air_O3_2018_proces

write.csv(Air_O3_2019_Selection1, row.names = FALSE, file = here('Data','Processed','Air_O3_2019_proces

write.csv(Air_PM25_2018_Selection1_replaced, row.names = FALSE, file = here('Data','Processed','Air_PM2

write.csv(Air_PM25_2019_Selection1_replaced, row.names = FALSE, file = here('Data','Processed','Air_PM2
```

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

#Make sure column names are identical

colnames(Air_O3_2018_Selection1)
```

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

```
colnames(Air_O3_2019_Selection1)
```

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

```
colnames(Air_PM25_2018_Selection1_replaced)
```

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

```
colnames(Air_PM25_2019_Selection1_replaced)
```

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

```
#Combine the four datasets

Air_Bind <- rbind(Air_O3_2018_Selection1, Air_O3_2019_Selection1, Air_PM25_2018_Selection1_replaced, Ai

#8

#Wrangle new dataset to fulfill the conditions in question

Air_Bind_8 <- Air_Bind %>%
  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") %>%
  group_by(Date,Site.Name,AQS_PARAMETER_DESC,COUNTY) %>%
  summarize(
    mean_DAILY_AQI_VALUE = mean(DAILY_AQI_VALUE),
    mean_SITE_LATITUDE = mean(SITE_LATITUDE),
    mean_SITE_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 using the `.groups` argument.
```

```
#9

#Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns

Air_Bind_8_spread9 <- pivot_wider(Air_Bind_8, names_from = AQS_PARAMETER_DESC, values_from = mean_DAILY_
Air_Bind_8_spread9
```

```
## # A tibble: 8,976 x 9
## # Groups:   Date, Site.Name [8,976]
##    Date       Site.Name      COUNTY mean_SITE_LATITUDE mean_SITE_LONGITUDE Month
##    <date>     <fct>          <fct>               <dbl>               <dbl> <dbl>
##  1 2018-01-01 Bryson City    Swain                35.4               -83.4     1
##  2 2018-01-01 Castle Hayne   New H~               34.4               -77.8     1
##  3 2018-01-01 Clemmons Midd~ Forsy~               36.0               -80.3     1
##  4 2018-01-01 Durham Armory  Durham               36.0               -78.9     1
##  5 2018-01-01 Garinger High~ Meckl~               35.2               -80.8     1
##  6 2018-01-01 Hattie Avenue  Forsy~               36.1               -80.2     1
##  7 2018-01-01 Leggett        Edgec~               36.0               -77.6     1
##  8 2018-01-01 Millbrook Sch~ Wake                 35.9               -78.6     1
##  9 2018-01-01 Pitt Agri. Ce~ Pitt                 35.6               -77.4     1
## 10 2018-01-01 West Johnston~ Johns~               35.6               -78.5     1
## # i 8,966 more rows
## # i 3 more variables: Year <dbl>, PM2.5 <dbl>, Ozone <dbl>
```

```
#10

#Call up the dimensions of your new tidy dataset

dim(Air_Bind_8_spread9)
```

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

```
#11

#Save processed dataset

write.csv(Air_Bind_8_spread9, row.names = FALSE, file = here('Data','Processed','EPAair_O3_PM25_NC1819_P
```

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.

```r
#12

#Use the split-apply-combine strategy to generate a summary data frame for the specified conditions

Air_Bind_12_summary <- Air_Bind_8_spread9 %>%
  group_by(Site.Name,Month,Year) %>%
  summarize(
    mean_DAILY_AQI_VALUE_Ozone = mean(Ozone),
    mean_DAILY_AQI_VALUE_PM25 = mean(PM2.5)) %>%
      drop_na(mean_DAILY_AQI_VALUE_Ozone)
```

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

```r
Air_Bind_12_summary
```

```
## # A tibble: 182 x 5
## # Groups:   Site.Name, Month [109]
##    Site.Name   Month  Year mean_DAILY_AQI_VALUE_Ozone mean_DAILY_AQI_VALUE_PM25
##    <fct>       <dbl> <dbl>                      <dbl>                     <dbl>
##  1 Bryson City     3  2018                       41.6                      34.7
##  2 Bryson City     3  2019                       42.5                      NA
##  3 Bryson City     4  2018                       44.5                      28.2
##  4 Bryson City     4  2019                       45.4                      26.7
##  5 Bryson City     5  2019                       39.6                      NA
##  6 Bryson City     6  2018                       37.8                      NA
##  7 Bryson City     6  2019                       34.0                      NA
##  8 Bryson City     7  2018                       34.6                      NA
##  9 Bryson City     7  2019                       30.4                      33.6
## 10 Bryson City     8  2018                       30.8                      NA
## # i 172 more rows
```

```r
#Note for reviewer / TA -- the below code is written here for convenience as part of the answer to #14

#Air_Bind_12_summary_TEST <- Air_Bind_8_spread9 %>%
  #group_by(Site.Name,Month,Year) %>%
  #summarize(
    #mean_DAILY_AQI_VALUE_Ozone = mean(Ozone),
    #mean_DAILY_AQI_VALUE_PM25 = mean(PM2.5)) %>%
      #na.omit(mean_DAILY_AQI_VALUE_Ozone)

#13

#Call up the dimensions of the summary dataset

dim(Air_Bind_12_summary)
```

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

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

Answer: 'na.omit' removes the entire object associated with NAs anywhere, so it also applied to the column / field for PM2.5 (and presumably other fields within the data frame) when we only wanted it to apply to the column / field for Ozone. Testing na.omit resulted in 101 records instead of 182 for 'drop_na()'. We used 'drop_na()' because we wanted to focus on a single column versus the entire data frame.