Assignment 4: Data Wrangling

Gaby Czarniak

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 tidyverse
library(tidyverse)
# loading lubridate
library(lubridate)
# loading here
library(here)
# loading dplyr
library(dplyr)

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

```
## [1] "/home/guest/gaby-cz_EDE_Fall2023"
```

```
setwd(here())

#1c
# reading in four EPA Air data sets
# setting string columns to be read in as factors
EPAair_O3_2018 <- read.csv("./Data/Raw/EPAair_O3_NC2018_raw.csv", stringsAsFactors = TRUE)
EPAair_O3_2019 <-read.csv("./Data/Raw/EPAair_O3_NC2019_raw.csv", stringsAsFactors = TRUE)
EPAair_PM25_2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)
EPAair_PM25_2019 <- read.csv("./Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)

#2
# applying glimpse function to reveal
# dimensions, column names, and structure of each data set

#colnames(EPAair_O3_2018)
#colnames(EPAair_O3_2019)
#colnames(EPAair_PM25_2018)
#colnames(EPAair_PM25_2019)
#head(EPAair_O3_2018) # seeing the first few observations of the data set
#summary(EPAair_O3_2018)
#str(EPAair_O3_2018) # shows the data type of each column
#dim(EPAair_O3_2018)

# getting glimpse of Ozone data from 2018
# 20 columns, 9,737 rows
glimpse(EPAair_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
# getting glimpse of Ozone data from 2019
# 20 columns, 10,592 rows
glimpse(EPAair_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~
```

```r
# getting glimpse of PM2.5 data from 2018
# 20 columns, 8,983 rows
glimpse(EPAair_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~
```

```
# getting glimpse of PM2.5 data from 2019
# 20 columns, 8,581 rows
glimpse(EPAair_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".

```
#3
# changing Date columns to be date objects
EPAair_O3_2018$Date <- as.Date(EPAair_O3_2018$Date, format = "%m/%d/%Y")
# EPAair_O3_2018$Date (commenting rather than printing because too long)
EPAair_O3_2019$Date <- as.Date(EPAair_O3_2019$Date, format = "%m/%d/%Y")
# EPAair_O3_2019$Date
EPAair_PM25_2018$Date <- as.Date(EPAair_PM25_2018$Date, format = "%m/%d/%Y")
# EPAair_PM25_2018$Date
EPAair_PM25_2019$Date <- as.Date(EPAair_PM25_2019$Date, format = "%m/%d/%Y")
# EPAair_PM25_2019$Date

# checking class to make sure objects were converted to dates
class(EPAair_O3_2018$Date)
```

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

```
class(EPAair_O3_2019$Date)
```

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

```
class(EPAair_PM25_2018$Date)
```

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

```
class(EPAair_PM25_2019$Date)
```

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

```
# returns "Date" for all

#4
# selecting specific columns that are of interest for analysis.
# for each of the four data sets
# putting in the original data as first argument
# then selecting requested columns
# then pulling out a newly created data subset
EPAair_O3_2018_subset <- select(EPAair_O3_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, C
# EPAair_O3_2018_subset (commenting not printing because too long)
EPAair_O3_2019_subset <- select(EPAair_O3_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, C
# EPAair_O3_2019_subset
EPAair_PM25_2018_subset <- select(EPAair_PM25_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES
# EPAair_PM25_2018_subset
EPAair_PM25_2019_subset <- select(EPAair_PM25_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DES
# EPAair_PM25_2019_subset

#5
# filling cells for PM2.5 data sets
# mutating 2018 subset to replace values with "PM2.5"
EPAair_PM25_2018_subset <-
  EPAair_PM25_2018_subset %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
# EPAair_PM25_2018_subset

# mutating 2019 subset to replace values with "PM2.5"
EPAair_PM25_2019_subset <-
  EPAair_PM25_2019_subset %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")
# EPAair_PM25_2019_subset

#6
# saving data sets in Processed folder
# using same names as Raw .csv files, just substituting "raw" with "processed"
# saving O3 2018 subset
write.csv(EPAair_O3_2018_subset, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2018_processed
# saving O3 2019 subset
```

```
write.csv(EPAair_O3_2019_subset, row.names = FALSE, file = "./Data/Processed/EPAair_O3_NC2019_processed
# saving PM2.5 2018 subset
write.csv(EPAair_PM25_2018_subset, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2018_proce
# saving PM2.5 2019 subset
write.csv(EPAair_PM25_2019_subset, row.names = FALSE, file = "./Data/Processed/EPAair_PM25_NC2019_proce
```

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
# using rbind to combine data sets
EPAair_O3PM25 <- rbind(
  EPAair_O3_2018_subset,
  EPAair_O3_2019_subset,
  EPAair_PM25_2018_subset,
  EPAair_PM25_2019_subset
)
# EPAair_O3PM25

# checking dimensions of new data set
dim(EPAair_O3PM25)
```

```
## [1] 37893     7
```

```
# results are 37893 rows and 7 columns

#8
# wrangling data set to fill conditions Wrangle your new dataset with a pipe function (%>%) so that it
```

```r
EPAair_O3PM25_processed <-
  EPAair_O3PM25 %>%
  #including only sites the four data sets have in common
  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")) %>%
  # grouping by date, site name, AQS parameter, and county
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  # taking mean of AQI value, latitude, and longitude
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlatitude = mean(SITE_LATITUDE),
            meanlongitude = mean(SITE_LONGITUDE)
  ) %>%
  # adding month column
  mutate(Month = month(Date)) %>%
  # Month
  # adding day column
  mutate(Day = day(Date))
```

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

```r
  # Day
# EPAair_O3PM25_processed

# checking that new data set dimensions are 14,752 x 9
dim(EPAair_O3PM25_processed)
```

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

```r
# confirmed, dimensions are 14,752 x 9

#9
# spreading data sets
# (from Luana) you should spread the wrangled data frame from part/question 8
# and not the combined data frame from part/question 7.

# spreading O3 and PM2.5 values in separate columns
# using pivot_wider
EPAair_O3PM25_processed_spread <-
  pivot_wider(EPAair_O3PM25_processed,
              names_from = AQS_PARAMETER_DESC, values_from = meanAQI)
# EPAair_O3PM25_processed_spread

#10
# calling dimensions of new data set
# each location on a specific date should now occupy only one row
dim(EPAair_O3PM25_processed_spread)
```

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

```r
# new dimensions are 8976 rows and 9 columns

#11
# saving processed data
write.csv(EPAair_O3PM25_processed_spread, row.names = FALSE, file = "./Data/Processed/EPAair_O3_PM25_NC
```

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
# generating summary data frame
EPAair_O3PM25_NC1819_summary <-
  EPAair_O3PM25_processed_spread %>%
  # adding year column to group by year cleanly
  mutate(Year = year(Date)) %>%
  # rearranging columns
  select(
    Site.Name, Year, Month, Day, COUNTY,
    meanlatitude, meanlongitude, PM2.5, Ozone) %>%
  # group by site, month, and year
  group_by(Site.Name, Month, Year) %>%
  # taking mean of AQI value, latitude, and longitude
  summarise(meanPM25 = mean(PM2.5),
            meanOzone = mean(Ozone)) %>%
            # meanPM25
            # meanOzone
  # removing instances where mean ozone values are not available
  drop_na(meanOzone)
```

```
## Adding missing grouping variables: `Date`
## `summarise()` has grouped output by 'Site.Name', 'Month'. You can override
## using the `.groups` argument.
```

```r
# we can check that NAs have been removed by doing a summary of
# the data set $ column, where we'll see min and max
summary(EPAair_O3PM25_NC1819_summary$meanOzone)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   36.23   42.32   41.34   45.76   59.23
```

```r
# meanOzone no longer contains NAs
summary(EPAair_O3PM25_NC1819_summary$meanPM25)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   11.84   29.30   33.19   32.76   37.74   44.60      81
```

```
# PM25 contains 81 NAs

# EPAair_O3PM25_NC1819_summary
print(head(EPAair_O3PM25_NC1819_summary))
```

```
## # A tibble: 6 x 5
## # Groups:   Site.Name, Month [4]
##   Site.Name   Month  Year meanPM25 meanOzone
##   <fct>       <dbl> <dbl>    <dbl>     <dbl>
## 1 Bryson City     3  2018     34.7      41.6
## 2 Bryson City     3  2019     NA        42.5
## 3 Bryson City     4  2018     28.2      44.5
## 4 Bryson City     4  2019     26.7      45.4
## 5 Bryson City     5  2019     NA        39.6
## 6 Bryson City     6  2018     NA        37.8
```

```
#13
# calling dimensions of summary data set
dim(EPAair_O3PM25_NC1819_summary)
```

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

```
# dimensions are 182 rows and 5 columns
```

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

    Answer: `drop` functions drop all of the "NA" values so that all NAs found do not show up in the new data set. According to R Documentation help, under "drop_na {tidyr}", drop_na "only keeps the 'complete' rows (where no rows contain missing values)" whereas, according to R Documentation help page "na.fail {stats}", when we use na.omit to remove cases, "the row numbers of the cases form the 'na.action' attribute of the result, of class 'omit'". Our use of drop_na is removing rows containing NAs from meanOzone and assigning the resulting rows to the newly created data set (mine being called EPAair_O3PM25_NC1819_summary) via piping, rather than omitting the NAs from the data set that was used as a reference.