Assignment 4: Data Wrangling (Fall 2024)

Rachael Stephan

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.

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. Add the appropriate code to reveal the dimensions of the four datasets.

```
#1a Load necessary packages
library(tidyverse); library(lubridate); library(here)

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

```
## [1] "/home/guest/Duke_R/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/Duke_R/EDE_Fall2024"
```

```
#1c read in EPA air data set
epaair_o3_2018 <- read.csv(file = here('./Data/Raw/EPAair_O3_NC2018_raw.csv'),
                           stringsAsFactors = TRUE)

epaair_o3_2019 <- read.csv(file = here('./Data/Raw/EPAair_O3_NC2019_raw.csv'),
                           stringsAsFactors = TRUE)

epaair_pm25_2018 <- read.csv(file = here('./Data/Raw/EPAair_PM25_NC2018_raw.csv'),
                           stringsAsFactors = TRUE)

epaair_pm25_2019 <- read.csv(file = here('./Data/Raw/EPAair_PM25_NC2019_raw.csv'),
                           stringsAsFactors = TRUE)

#2 Check dimensions of data sets
dim(epaair_o3_2018)
```

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

```
dim(epaair_o3_2019)
```

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

```
dim(epaair_pm25_2018)
```

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

```
dim(epaair_pm25_2019)
```

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

All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern?

epaair_o3_2018: 9737 rows, 20 columns

epaair_o3_2019: 10592 rows, 20 columns

epaair_pm25_2018: 8983 rows, 20 columns

epaair_pm25_2019: 8581 rows, 20 columns

Yes. My data sets follow this pattern.

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 Change each dataset's date column to date objects by checking the format and
  #using the corresponding lubridate function. Results are not printed.

head(epaair_o3_2018$Date, 3)
epaair_o3_2018$Date <- mdy(epaair_o3_2018$Date)

head(epaair_o3_2019$Date, 3)
epaair_o3_2019$Date <- mdy(epaair_o3_2019$Date)

head(epaair_pm25_2018$Date, 3)
epaair_pm25_2018$Date <- mdy(epaair_pm25_2018$Date)

head(epaair_pm25_2019$Date, 3)
epaair_pm25_2019$Date <- mdy(epaair_pm25_2019$Date)

#4-5 Select columns of interest and change the values of AQS_PARAMETER_DESC to desired value
epaair_o3_2018_processed <-
  epaair_o3_2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)

epaair_o3_2019_processed <-
  epaair_o3_2019 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE)

epaair_pm25_2018_processed <-
  epaair_pm25_2018 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE) %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

epaair_pm25_2019_processed <-
  epaair_pm25_2019 %>%
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY,
         SITE_LATITUDE, SITE_LONGITUDE) %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

#6 Save the processed files to the processed folder
write.csv(
  epaair_o3_2018_processed,
  file = here("./Data/Processed/EPAair_O3_NC2018_processed.csv"),
  row.names=FALSE)

write.csv(
  epaair_o3_2019_processed,
  file = here("./Data/Processed/EPAair_O3_NC2019_processed.csv"),
  row.names=FALSE)

write.csv(
  epaair_pm25_2018_processed,
```

```
  file = here("./Data/Processed/EPAair_PM25_NC2018_processed.csv"),
  row.names=FALSE)

write.csv(
  epaair_pm25_2019_processed,
  file = here("./Data/Processed/EPAair_PM25_NC2019_processed.csv"),
  row.names=FALSE)
```

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
#check column names of datasets to see if they are the same
colnames(epaair_o3_2018_processed)
```

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

```
colnames(epaair_o3_2019_processed)
```

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

```r
colnames(epaair_pm25_2018_processed)
```

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

```r
colnames(epaair_pm25_2019_processed)
```

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

```r
#combine columns
epaair_all <- rbind(epaair_o3_2018_processed,
                    epaair_o3_2019_processed,
                    epaair_pm25_2018_processed,
                    epaair_pm25_2019_processed)

#8
#make vector of wanted site names,
sites <- as.factor(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"))

#perform pipe
epaair_all_processed <-
  epaair_all %>%
  filter(Site.Name %in% sites) %>% #filter for wanted sites
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY)%>% #group
  summarise(mean_AQI = mean(DAILY_AQI_VALUE), #summarize values of interest
            mean_lat = mean(SITE_LATITUDE),
            mean_long = mean(SITE_LONGITUDE))%>%
  mutate(Month = month(Date)) %>% #add column for month
  mutate(Year = year(Date)) %>% #add column for year
  select(Date, Year, Month, COUNTY, Site.Name, AQS_PARAMETER_DESC, #reorder
         mean_AQI:mean_long)
```

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

```r
#check dims
dim(epaair_all_processed)
```

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

```r
#9
#use pivot wider to separate o3 and pm2.5
```

```
epaair_all_processed_wide <- pivot_wider(
  epaair_all_processed,
  names_from = AQS_PARAMETER_DESC,
  values_from = mean_AQI)

#10
#get new dims and take a brief look at matrix with tibble
dim(epaair_all_processed_wide)
```

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

```
tibble(epaair_all_processed_wide)
```

```
## # A tibble: 8,976 x 9
##    Date        Year Month COUNTY      Site.Name   mean_lat mean_long PM2.5 Ozone
##    <date>     <dbl> <dbl> <fct>       <fct>          <dbl>     <dbl> <dbl> <dbl>
##  1 2018-01-01  2018     1 Swain       Bryson City     35.4     -83.4    35    NA
##  2 2018-01-01  2018     1 New Hanover Castle Hay~     34.4     -77.8    13    NA
##  3 2018-01-01  2018     1 Forsyth     Clemmons M~     36.0     -80.3    24    NA
##  4 2018-01-01  2018     1 Durham      Durham Arm~     36.0     -78.9    31    NA
##  5 2018-01-01  2018     1 Mecklenburg Garinger H~     35.2     -80.8    20    32
##  6 2018-01-01  2018     1 Forsyth     Hattie Ave~     36.1     -80.2    22    NA
##  7 2018-01-01  2018     1 Edgecombe   Leggett         36.0     -77.6    14    NA
##  8 2018-01-01  2018     1 Wake        Millbrook ~     35.9     -78.6    28    34
##  9 2018-01-01  2018     1 Pitt        Pitt Agri.~     35.6     -77.4    15    NA
## 10 2018-01-01  2018     1 Johnston    West Johns~     35.6     -78.5    24    NA
## # i 8,966 more rows
```

```
#11
#save new table to CSV
write.csv(epaair_all_processed_wide, 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 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
#perform pipe to get summary
epaair_all_summary <-
  epaair_all_processed_wide %>%
  group_by(Site.Name, Year, Month)%>%
  summarise(mean_o3_AQI = mean(Ozone),
            mean_pm25_AQI = mean(PM2.5))%>%
  drop_na(mean_o3_AQI)
```

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

```
#13
#get new dims and take a brief look at matrix with tibble
dim(epaair_all_summary)
```

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

```
tibble(epaair_all_summary)
```

```
## # A tibble: 182 x 5
##    Site.Name    Year Month mean_o3_AQI mean_pm25_AQI
##    <fct>       <dbl> <dbl>       <dbl>         <dbl>
##  1 Bryson City  2018     3        41.6          34.7
##  2 Bryson City  2018     4        44.5          28.2
##  3 Bryson City  2018     6        37.8          NA
##  4 Bryson City  2018     7        34.6          NA
##  5 Bryson City  2018     8        30.8          NA
##  6 Bryson City  2018     9        25.4          25.1
##  7 Bryson City  2018    10        31            31.3
##  8 Bryson City  2019     3        42.5          NA
##  9 Bryson City  2019     4        45.4          26.7
## 10 Bryson City  2019     5        39.6          NA
## # i 172 more rows
```

14. Why did we use the function `drop_na` rather than `na.omit`? Hint: replace `drop_na` with `na.omit` in part 12 and observe what happens with the dimensions of the summary date frame.

    Answer: The `na.omit` function does not allow for column control. It will drop any row/observation that contains NA without discrimination. The `drop_na` function will allow specification for which column(s) to examine for NA values. `drop_na` will only drop rows/oberservations in