Assignment 4: Data Wrangling

Keanu Valibia

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

```
# 1a activate required packages
library("tidyverse")
library("lubridate")
library("here")

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

```
## [1] "/home/guest/R/R Projects/EDA_Spring2024"
```

```
# 1c read in csv files into data frames
EPAair.o3.nc2018 <- read.csv("~/R/R Projects/EDA_Spring2024/Data/Raw/EPAair_O3_NC2018_raw.csv",
    stringsAsFactors = TRUE)
EPAair.o3.nc2019 <- read.csv("~/R/R Projects/EDA_Spring2024/Data/Raw/EPAair_O3_NC2019_raw.csv",
    stringsAsFactors = TRUE)
EPAair.pm25.nc2018 <- read.csv("~/R/R Projects/EDA_Spring2024/Data/Raw/EPAair_PM25_NC2018_raw.csv",
```

```r
    stringsAsFactors = TRUE)
EPAair.pm25.nc2019 <- read.csv("~/R/R Projects/EDA_Spring2024/Data/Raw/EPAair_PM25_NC2019_raw.csv",
    stringsAsFactors = TRUE)

# 2 quickly review imports
glimpse(EPAair.o3.nc2018)
```

```
## 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(EPAair.o3.nc2019)
```

```
## 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(EPAair.pm25.nc2018)
```

```
## 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(EPAair.pm25.nc2019)
```

```
## 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 convert Date field from factor to date class,
# taking into account format
EPAair.o3.nc2018$Date <- as.Date(EPAair.o3.nc2018$Date,
    format = "%m/%d/%Y")

EPAair.o3.nc2019$Date <- as.Date(EPAair.o3.nc2019$Date,
    format = "%m/%d/%Y")

EPAair.pm25.nc2018$Date <- as.Date(EPAair.pm25.nc2018$Date,
    format = "%m/%d/%Y")

EPAair.pm25.nc2019$Date <- as.Date(EPAair.pm25.nc2019$Date,
    format = "%m/%d/%Y")

# 4 select fields to keep
EPAair.o3.nc2018.filtered <- select(EPAair.o3.nc2018,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair.o3.nc2019.filtered <- select(EPAair.o3.nc2019,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair.pm25.nc2018.filtered <- select(EPAair.pm25.nc2018,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

EPAair.pm25.nc2019.filtered <- select(EPAair.pm25.nc2019,
    Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
    COUNTY, SITE_LATITUDE, SITE_LONGITUDE)

# 5 update aws parameter values in pm2.5 data
# frames
EPAair.pm25.nc2018.filtered$AQS_PARAMETER_DESC <- "PM2.5"

EPAair.pm25.nc2019.filtered$AQS_PARAMETER_DESC <- "PM2.5"

# 6 write new csv files into processed data
# folder
write.csv(EPAair.o3.nc2018.filtered, row.names = FALSE,
    file = "~/R/R Projects/EDA_Spring2024/Data/Processed/EPAair_O3_NC2018_processed.csv")
```

```r
write.csv(EPAair.o3.nc2019.filtered, row.names = FALSE,
    file = "~/R/R Projects/EDA_Spring2024/Data/Processed/EPAair_O3_NC2019_processed.csv")

write.csv(EPAair.pm25.nc2018.filtered, row.names = FALSE,
    file = "~/R/R Projects/EDA_Spring2024/Data/Processed/EPAair_PM25_NC2018_processed.csv")

write.csv(EPAair.pm25.nc2019.filtered, row.names = FALSE,
    file = "~/R/R Projects/EDA_Spring2024/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"

```r
# 7 combine all files into a single data frame
EPAair.combined <- rbind(EPAair.o3.nc2018.filtered,
    EPAair.o3.nc2019.filtered, EPAair.pm25.nc2018.filtered,
    EPAair.pm25.nc2019.filtered)

glimpse(EPAair.combined)
```

```
## 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

EPAair.combined.filtered <- EPAair.combined %>%
    # filter site name field for specific values.
    # Drop records that do not include the listed
    # values
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 records within selected fields
group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
    # take average of selected fields
summarise(aqiMean = mean(DAILY_AQI_VALUE, na.rm = TRUE),
    latMean = mean(SITE_LATITUDE, na.rm = TRUE), longMean = mean(SITE_LONGITUDE,
        na.rm = TRUE), .groups = "keep") %>%
    # create new month field based on Date field
mutate(Month = month(ymd(Date))) %>%
    # create new year field based on Date field
mutate(Year = year(ymd(Date)))

# 9 turn ozone and pm2.5 values into fields,
# using aqi mean as value
EPAair.combined.filtered.spread <- spread(EPAair.combined.filtered,
    AQS_PARAMETER_DESC, aqiMean)

# 10 call dimensions of new data frame
dim(EPAair.combined.filtered.spread)
```

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

```r
# 11 upload data frame to csv in processed data
# folder
write.csv(EPAair.combined.filtered, row.names = FALSE,
    file = "~/R/R Projects/EDA_Spring2024/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.

```r
# 12
EPAair.summary <- EPAair.combined.filtered.spread %>%
    # group by selected fields
group_by(Site.Name, Month, Year) %>%
    # create new fields based on the mean of
    # ozone and pm2.5 values
```

```
summarise(meanOzone = mean(Ozone), meanPm = mean(PM2.5),
    .groups = "keep") %>%
    # drop any records that have NA value in
    # Ozone specifically
drop_na(meanOzone)

# 13 call dimensions of new data frame
dim(EPAair.summary)
```

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

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: 'drop_na' only removes instances where NA is in a specified field; it ignores other fields in which NA is present. 'na.omit' would remove the entire record, regardless of field specified. Since we want to still keep records even where other fields have an NA value, we would want to use the 'drop_na' function