Assignment 4: Data Wrangling

Courtney Horn

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling #Data wrangling is the process of gathering and transforming data

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A04_DataWrangling.Rmd") prior to submission.

The completed exercise is due on Tuesday, Feb 16 @ 11:59pm.

Set up your session

1. Check your working directory, load the `tidyverse` and `lubridate` packages, and upload all four raw data files associated with the EPA Air dataset. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Explore the dimensions, column names, and structure of the datasets.

```
#1
library(tidyverse)
library(lubridate)
getwd()
```

```
## [1] "/Users/courtneyannehorn/Desktop/EDA/EDAfin/Assignments"
```

```
EPA_air_03_2018 <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv",stringsAsFactors = TRUE)
EPA_air_03_2019 <- read.csv("../Data/Raw/EPAair_O3_NC2019_raw.csv",stringsAsFactors = TRUE)
EPA_air_PM25_2018 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv",stringsAsFactors = TRUE)
EPA_air_PM25_2019 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv",stringsAsFactors = TRUE)
#2
```

Wrangle individual datasets to create processed files.

3. Change date to date
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".

```
#olnames(EPA_air_03_2018)
  #Date column
#3
  #EPA_air_03_2018
#class(EPA_air_03_2018$Date)
#head(EPA_air_03_2018$Date)
EPA_air_03_2018$Date <- as.Date(EPA_air_03_2018$Date, format = "%m/%d/%Y")
#class(EPA_air_03_2018$Date)
#head(EPA_air_03_2018$Date)

  #EPA_air_03_2019
#class(EPA_air_03_2019$Date)
#head(EPA_air_03_2019$Date)
EPA_air_03_2019$Date <- as.Date(EPA_air_03_2019$Date, format = "%m/%d/%Y")
#class(EPA_air_03_2019$Date)
#head(EPA_air_03_2019$Date)

  #EPA_air_PM25_2018
#class(EPA_air_PM25_2018$Date)
#head(EPA_air_PM25_2018$Date)
EPA_air_PM25_2018$Date <- as.Date(EPA_air_PM25_2018$Date, format = "%m/%d/%Y")
#class(EPA_air_PM25_2018$Date)
#head(EPA_air_PM25_2018$Date)

   #EPA_air_PM25_2019
#class(EPA_air_PM25_2019$Date)
#head(EPA_air_PM25_2019$Date)
EPA_air_PM25_2019$Date <- as.Date(EPA_air_PM25_2019$Date, format = "%m/%d/%Y")
#class(EPA_air_PM25_2019$Date)
#head(EPA_air_PM25_2019$Date)



#4
  #EPA_air_03_2018
EPA_air_03_2018_subset <- select(EPA_air_03_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
colnames(EPA_air_03_2018_subset)
```

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

```
  #EPA_air_03_2019
EPA_air_03_2019_subset <- select(EPA_air_03_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
colnames(EPA_air_03_2019_subset)
```

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

```
 #EPA_air_PM25_2018
EPA_air_PM25_2018_subset <- select(EPA_air_PM25_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_D
colnames(EPA_air_PM25_2018_subset)
```

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

```
 #EPA_air_PM25_2019
EPA_air_PM25_2019_subset <- select(EPA_air_PM25_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DE
colnames(EPA_air_PM25_2019_subset)
```

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

```
#Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE



#5
  #EPA_air_PM25_2018
#colnames(EPA_air_PM25_2018)
#head(EPA_air_PM25_2018$AQS_PARAMETER_DESC)
EPA_air_PM25_2018$AQS_PARAMETER_DESC <- c("PM2.5")
#colnames(EPA_air_PM25_2018)
#head(EPA_air_PM25_2018_subset$AQS_PARAMETER_DESC)
EPA_air_PM25_2018_subset$AQS_PARAMETER_DESC <- c("PM2.5")

  #EPA_air_PM25_2019
#head(EPA_air_PM25_2019$AQS_PARAMETER_DESC)
EPA_air_PM25_2019$AQS_PARAMETER_DESC <- c("PM2.5")
#head(EPA_air_PM25_2019$AQS_PARAMETER_DESC)
EPA_air_PM25_2019_subset$AQS_PARAMETER_DESC <- c("PM2.5")
#head(EPA_air_PM25_2019_subset$AQS_PARAMETER_DESC)



#6
  #6 6. Save all four processed datasets in the Processed folder. Use the same file names as the raw fi
write.csv(EPA_air_03_2018_subset, row.names = FALSE,
          file ="../Data/Processed/EPAair_O3_NC2018_processed.csv")

write.csv(EPA_air_03_2019_subset, row.names = FALSE,
          file ="../Data/Processed/EPAair_O3_NC2019_processed.csv")

write.csv(EPA_air_PM25_2018_subset, row.names = FALSE,
          file ="../Data/Processed/EPAair_PM25_NC2018_processed.csv")

write.csv(EPA_air_PM25_2019_subset, row.names = FALSE,
          file ="../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 all 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)
- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site, 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_NC1718_Processed.csv”

```
#7
#colnames(EPA_air_03_2018_subset) <- colnames(EPA_air_PM25_2018_subset)
colnames(EPA_air_03_2019_subset) <- colnames(EPA_air_PM25_2018_subset)
colnames(EPA_air_PM25_2019_subset) <- colnames(EPA_air_PM25_2018_subset)
air_dat_comb <-rbind(EPA_air_03_2018_subset, EPA_air_03_2019_subset, EPA_air_PM25_2018_subset, EPA_air_P

#air_dat1 <- rbind(EPA_air_03_2018, EPA_air_03_2019)
#air_dat2 <-rbind(EPA_air_PM25_2018, EPA_air_PM25_2019)
  #just gotta figure out how to change that one col name here
#colnames(air_dat1) <- c("Date","Source","Site.ID","POC",         "Daily.Max.Conce","UNITS"
#colnames(air_dat2) <- c("Date","Source","Site.ID","POC",         "Daily.Max.Conce","UNITS"
#air_dat_comb <- rbind(air_dat1, air_dat2)
#dim(air_dat_comb)
#colnames(air_dat_comb)




#8
air_dat_comb_wrang3 <-
  air_dat_comb %>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
  mutate(month = month(Date)) %>%
  mutate(year = year(Date)) %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMET
   dplyr::summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlat = mean(SITE_LATITUDE),
            meanlong = mean(SITE_LONGITUDE))
```

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

```
air_dat_comb_wrang4 <-
  air_dat_comb %>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMET
   dplyr::summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlat = mean(SITE_LATITUDE),
            meanlong = mean(SITE_LONGITUDE)) %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date))
```

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

```
dim(air_dat_comb_wrang4)
```

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

```
#9.
air_dat_comb_wrang_spread <- pivot_wider(air_dat_comb_wrang4, names_from = AQS_PARAMETER_DESC, values_f
colnames(air_dat_comb_wrang_spread)
```

```
## [1] "Date"      "Site.Name" "COUNTY"    "meanlat"   "meanlong"  "month"
## [7] "year"      "PM2.5"     "Ozone"
```

```
#10
dim(air_dat_comb_wrang_spread)
```

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

```
#the dimesions are 8976, 7



#11
write.csv(EPA_air_PM25_2019_subset, row.names = FALSE,
          file ="./EPAair_O3_PM25_NC1718_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 a month and year are not available (use the function `drop_na` in your pipe).

13. Call up the dimensions of the summary dataset.

```
#12a and b
#specify its dplyr summarise
sumdf <-
  air_dat_comb_wrang_spread %>%
  group_by(Site.Name, month, year) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMETER_DESC
   dplyr::summarise(meanOzone = mean(Ozone),
                    meanPM2.5 = mean(PM2.5)) %>%
  drop_na(month) %>%
  drop_na(year)
```

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

```
  natest <-
  air_dat_comb_wrang_spread %>%
  group_by(Site.Name, month, year) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMETER_DESC
   dplyr::summarise(meanOzone = mean(Ozone),
                    meanPM2.5 = mean(PM2.5)) %>%
  na.omit(month) %>%
  na.omit(year)
```

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

```
#13
dim(sumdf)
```

```
## [1] 308   5
```

```
dim(natest)
```

```
## [1] 101   5
```

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

    Answer: We want to remove rows containing NAs. drop na allows us to only remove rows with nas in month or year. na.omit will remove rows with nas in any column.

Appendix

```
     ##3
 #EPA_air_03_2018
class(EPA_air_03_2018$Date)
head(EPA_air_03_2018$Date)
EPA_air_03_2018$Date <- as.Date(EPA_air_03_2018$Date, format = "%m/%d/%Y")
class(EPA_air_03_2018$Date)
head(EPA_air_03_2018$Date)

 #EPA_air_03_2019
class(EPA_air_03_2019$Date)
head(EPA_air_03_2019$Date)
EPA_air_03_2019$Date <- as.Date(EPA_air_03_2019$Date, format = "%m/%d/%Y")
class(EPA_air_03_2019$Date)
head(EPA_air_03_2019$Date)

 #EPA_air_PM25_2018
class(EPA_air_PM25_2018$Date)
head(EPA_air_PM25_2018$Date)
EPA_air_PM25_2018$Date <- as.Date(EPA_air_PM25_2018$Date, format = "%m/%d/%Y")
class(EPA_air_PM25_2018$Date)
head(EPA_air_PM25_2018$Date)

  #EPA_air_PM25_2019
class(EPA_air_PM25_2019$Date)
head(EPA_air_PM25_2019$Date)
EPA_air_PM25_2019$Date <- as.Date(EPA_air_PM25_2019$Date, format = "%m/%d/%Y")
class(EPA_air_PM25_2019$Date)
head(EPA_air_PM25_2019$Date)

     ##4
 #EPA_air_03_2018
EPA_air_03_2018_subset <- select(EPA_air_03_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
colnames(EPA_air_03_2018_subset)

 #EPA_air_03_2019
EPA_air_03_2019_subset <- select(EPA_air_03_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
colnames(EPA_air_03_2019_subset)

 #EPA_air_PM25_2018
EPA_air_PM25_2018_subset <- select(EPA_air_PM25_2018, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_D
colnames(EPA_air_PM25_2018_subset)

 #EPA_air_PM25_2019
EPA_air_PM25_2019_subset <- select(EPA_air_PM25_2019, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_D
colnames(EPA_air_PM25_2019_subset)


#5
```

```
  #EPA_air_PM25_2018
colnames(EPA_air_PM25_2018)
head(EPA_air_PM25_2018$AQS_PARAMETER_DESC)
EPA_air_PM25_2018$AQS_PARAMETER_DESC <- c("PM2.5")


  #EPA_air_PM25_2019
head(EPA_air_PM25_2019$AQS_PARAMETER_DESC)
EPA_air_PM25_2019$AQS_PARAMETER_DESC <- c("PM2.5")


  ##6
write.csv(EPA_air_03_2018, row.names = FALSE,
          file ="../Data/Processed/EPAair_O3_NC2018_processed.csv")

write.csv(EPA_air_03_2019, row.names = FALSE,
          file ="../Data/Processed/EPAair_O3_NC2019_processed.csv")

write.csv(EPA_air_PM25_2018, row.names = FALSE,
          file ="../Data/Processed/EPAair_PM25_NC2018_processed.csv")

write.csv(EPA_air_PM25_2019, row.names = FALSE,
          file ="../Data/Processed/EPAair_PM25_NC2019_processed.csv")



  ##7
air_dat1 <- rbind(EPA_air_03_2018, EPA_air_03_2019)
air_dat2 <-rbind(EPA_air_PM25_2018, EPA_air_PM25_2019)
  #just gotta figure out how to change that one col name here
colnames(EPA_air_03_2018) <- colnames(EPA_air_PM25_2018)
colnames(EPA_air_03_2019) <- colnames(EPA_air_PM25_2018)
colnames(EPA_air_PM25_2019) <- colnames(EPA_air_PM25_2018)
air_dat_comb <-rbind(EPA_air_03_2018, EPA_air_03_2019, EPA_air_PM25_2018, EPA_air_PM25_2019)

#colnames(air_dat1) <- c("Date","Source","Site.ID","POC",         "Daily.Max.Conce","UNITS"
#colnames(air_dat2) <- c("Date","Source","Site.ID","POC",         "Daily.Max.Conce","UNITS"
#air_dat_comb <- rbind(air_dat1, air_dat2)

  ##8

colnames(air_dat_comb)
class(air_dat_comb$Date)


air_dat_comb_wrang1 <-
  air_dat_comb %>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue", "Clemmons Middl
   mutate(month = month(Date)) %>%
  mutate(year = year(Date)) %>%
  mutate(day = day(Date)) %>%
  group_by(day, Site.Name, AQS_PARAMETER_CODE, COUNTY_CODE) %>%
   summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanlat = mean(SITE_LATITUDE),
            meanlong = mean(SITE_LONGITUDE))
dim(air_dat_comb_wrang1)


#9.
air_dat_comb_wrang_spread <- pivot_wider(air_dat_comb_wrang4, names_from = AQS_PARAMETER_DESC, values_f
colnames(air_dat_comb_wrang_spread)




#10
dim(air_dat_comb_wrang_spread)
#the dimesions are 8976, 7


#11
write.csv(EPA_air_PM25_2019_subset, row.names = FALSE,
          file ="./EPAair_O3_PM25_NC1718_Processed.csv")


#12a and b
#specify its dplyr summarise
sumdf <-
  air_dat_comb_wrang_spread %>%
  group_by(Site.Name, month, year) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMETER_DESC
   dplyr::summarise(meanOzone = mean(Ozone),
                    meanPM2.5 = mean(PM2.5)) %>%
  drop_na(month) %>%
  drop_na(year)

  natest <-
  air_dat_comb_wrang_spread %>%
  group_by(Site.Name, month, year) %>%  #changed AQS_PARAMETER_CODE to AQS_PARAMETER_DESC
   dplyr::summarise(meanOzone = mean(Ozone),
                    meanPM2.5 = mean(PM2.5)) %>%
  na.omit(month) %>%
  na.omit(year)
#13
dim(sumdf)
dim(natest)
```