Assignment 4: Data Wrangling

Merve Onal

## 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).

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

#1a   
library(tidyverse)  
library(lubridate)  
library(here)  
  
#1b   
getwd()

## [1] "/home/guest/EDA\_Spring2024"

here()

## [1] "/home/guest/EDA\_Spring2024"

#1c   
# I read all four raw data files associated with the EPA Air dataset  
EPAair\_O3\_NC2018\_raw <- read.csv(here("./Data/Raw/EPAair\_O3\_NC2018\_raw.csv"),  
stringsAsFactors = TRUE)  
EPAair\_O3\_NC2019\_raw <- read.csv(here("./Data/Raw/EPAair\_O3\_NC2019\_raw.csv"),  
stringsAsFactors = TRUE)  
EPAair\_PM25\_NC2018\_raw <- read.csv(here  
("./Data/Raw/EPAair\_PM25\_NC2018\_raw.csv"), stringsAsFactors = TRUE)  
EPAair\_PM25\_NC2019\_raw <- read.csv(here  
("./Data/Raw/EPAair\_PM25\_NC2019\_raw.csv"), stringsAsFactors = TRUE)  
#2   
# I apply the glimpse() function to reveal dimensions, column names, and structure for the dataset  
glimpse(EPAair\_O3\_NC2018\_raw)

## 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…

glimpse(EPAair\_O3\_NC2019\_raw)

## 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\_raw)

## 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\_raw)

## 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.

1. Change the Date columns to be date objects.
2. Select the following columns: Date, DAILY\_AQI\_VALUE, Site.Name, AQS\_PARAMETER\_DESC, COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE
3. For the PM2.5 datasets, fill all cells in AQS\_PARAMETER\_DESC with “PM2.5” (all cells in this column should be identical).
4. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace “raw” with “processed”.

#3  
# I convert Date columns to date objects  
EPAair\_O3\_NC2018\_raw$Date <- as.Date(EPAair\_O3\_NC2018\_raw$Date, format = "%m/%d/%Y")  
EPAair\_O3\_NC2019\_raw$Date <- as.Date(EPAair\_O3\_NC2019\_raw$Date, format = "%m/%d/%Y")  
EPAair\_PM25\_NC2018\_raw$Date <- as.Date(EPAair\_PM25\_NC2018\_raw$Date, format = "%m/%d/%Y")  
EPAair\_PM25\_NC2019\_raw$Date <- as.Date(EPAair\_PM25\_NC2019\_raw$Date, format = "%m/%d/%Y")  
  
#4  
selected\_data1 <- select(EPAair\_O3\_NC2018\_raw,  
Date, DAILY\_AQI\_VALUE,  
Site.Name, AQS\_PARAMETER\_DESC,  
COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE)  
selected\_data2 <- select(EPAair\_O3\_NC2019\_raw,  
Date, DAILY\_AQI\_VALUE,  
Site.Name, AQS\_PARAMETER\_DESC,  
COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE)  
selected\_data3 <- select(EPAair\_PM25\_NC2018\_raw,  
Date, DAILY\_AQI\_VALUE,  
Site.Name, AQS\_PARAMETER\_DESC,  
COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE)  
selected\_data4 <- select(EPAair\_PM25\_NC2019\_raw,  
Date, DAILY\_AQI\_VALUE,  
Site.Name, AQS\_PARAMETER\_DESC,  
COUNTY, SITE\_LATITUDE, SITE\_LONGITUDE)  
#5  
# For the PM2.5 datasets, I fill all cells in AQS\_PARAMETER\_DESC with "PM2.5"  
selected\_data3 <- selected\_data3 %>%  
mutate(AQS\_PARAMETER\_DESC = "PM2.5")  
selected\_data4 <- selected\_data4 %>%  
mutate(AQS\_PARAMETER\_DESC = "PM2.5")  
#6  
# Now I define the folder path for the Processed folder  
processed\_folder <- "./Data/Processed/"  
# Here I save the processed datasets with the updated file names  
write.csv(selected\_data1, file.path(processed\_folder,  
"EPAair\_O3\_NC2018\_processed.csv"), row.names = FALSE)  
write.csv(selected\_data2, file.path(processed\_folder,  
"EPAair\_O3\_NC2019\_processed.csv"), row.names = FALSE)  
write.csv(selected\_data3, file.path(processed\_folder,  
"EPAair\_PM25\_NC2018\_processed.csv"), row.names = FALSE)  
write.csv(selected\_data4, file.path(processed\_folder,  
"EPAair\_PM25\_NC2019\_processed.csv"), row.names = FALSE)

## Combine datasets

1. Combine the four datasets with rbind. Make sure your column names are identical prior to running this code.
2. 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.

1. 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.
2. Call up the dimensions of your new tidy dataset.
3. Save your processed dataset with the following file name: “EPAair\_O3\_PM25\_NC1819\_Processed.csv”

#7   
# I make sure that the column names are identical across the datasets  
#This works outside of knitting if(all(colnames(selected\_data1) == colnames(selected\_data2) &&  
#colnames(selected\_data2) == colnames(selected\_data3) &&  
#colnames(selected\_data3) == colnames(selected\_data4)))  
#I combine the four datasets using rbind  
combined\_data <- rbind(selected\_data1, selected\_data2, selected\_data3, selected\_data4)  
#8   
#I install packages install.packages("dplyr")  
library(dplyr)  
library(lubridate)  
#I define the list of common sites  
common\_sites <- 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"  
)  
# I filter only the common sites  
filtered\_data <- combined\_data %>%  
filter(Site.Name %in% common\_sites)  
# I group by date, site name, AQS parameter, and county, and calculate daily means  
grouped\_data <- filtered\_data %>%  
group\_by(Date, Site.Name, AQS\_PARAMETER\_DESC, COUNTY) %>%  
summarize(  
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 the `.groups` argument.

#I add columns for Month and Year  
grouped\_data <- grouped\_data %>%  
mutate(  
Month = month(Date),  
Year = year(Date)  
)  
#I check the dimensions of the final dataset, which should be 14,752 x 9  
dim(grouped\_data)

## [1] 14752 9

#9  
library(tidyr)  
#I spread the data to have AQI values for ozone and PM2.5 in separate columns  
spread\_data <- grouped\_data %>%  
pivot\_wider(  
id\_cols = c(Date, Site.Name, COUNTY, Year, Month, meanLAT, meanLONG),  
names\_from = AQS\_PARAMETER\_DESC,  
values\_from = meanAQI,  
names\_prefix = "AQI\_"  
)  
#I check the structure of the spread\_data  
glimpse(spread\_data)

## Rows: 8,976  
## Columns: 9  
## Groups: Date, Site.Name [8,976]  
## $ Date <date> 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01,…  
## $ Site.Name <fct> Bryson City, Castle Hayne, Clemmons Middle, Durham Armory, G…  
## $ COUNTY <fct> Swain, New Hanover, Forsyth, Durham, Mecklenburg, Forsyth, E…  
## $ Year <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, …  
## $ Month <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ meanLAT <dbl> 35.43477, 34.36417, 36.02600, 36.03296, 35.24010, 36.11069, …  
## $ meanLONG <dbl> -83.44213, -77.83861, -80.34200, -78.90404, -80.78568, -80.2…  
## $ AQI\_PM2.5 <dbl> 35.0, 13.0, 24.0, 31.0, 20.0, 22.0, 14.0, 28.0, 15.0, 24.0, …  
## $ AQI\_Ozone <dbl> NA, NA, NA, NA, 32, NA, NA, 34, NA, NA, NA, NA, NA, NA, NA, …

#10  
# I call up the dimensions of the new tidy dataset  
dim(spread\_data)

## [1] 8976 9

#11  
# I save my processed dataset with the specified file name  
write.csv(spread\_data, file.path(processed\_folder,  
"EPAair\_O3\_PM25\_NC1819\_Processed.csv"), row.names = FALSE)

## Generate summary tables

1. 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.
2. Call up the dimensions of the summary dataset.

#12  
library(dplyr)  
#I group the data by Site Name, Month and Year  
summary\_data <- spread\_data %>%  
group\_by(Site.Name, Month, Year)  
#I calculate the mean AQI values for ozone and PM2.5  
summary\_data <- spread\_data %>%  
group\_by(Site.Name, Month, Year) %>%  
summarize(mean\_AQI\_Ozone = mean(AQI\_Ozone), mean\_AQI\_PM25 = mean(AQI\_PM2.5)) %>%  
#I remove instances where mean ozone values are not available  
drop\_na(mean\_AQI\_Ozone)

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

#I view the summary data frame  
head(summary\_data)

## # A tibble: 6 × 5  
## # Groups: Site.Name, Month [4]  
## Site.Name Month Year mean\_AQI\_Ozone mean\_AQI\_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

#13  
# I call up the dimensions of the summary dataset  
dim(summary\_data)

## [1] 182 5

1. 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: We used the ‘drop\_na’ function rather than ‘na.omit’ because drop\_na allows us to selectively remove rows with missing values based on specific columns; thus, it offers more flexibility and customization. Na.omit is used to remove any rows from a data frame that contain missing (NA) values. In our case, we wanted to retain rows with missing PM2.5 values while removing rows with missing ozone values; thus, ‘drop\_na’ function meets our requirements.