Assignment 4: Data Wrangling

Walker Grimshaw

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on data wrangling.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A04_DataWrangling.pdf") prior to submission.

The completed exercise is due on Thursday, 7 February, 2019 before class begins.

Set up your session

1. Check your working directory, load the `tidyverse` package, 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. Generate a few lines of code to get to know your datasets (basic data summaries, etc.).

```
#1
getwd() ## in project folder, changed to assignment folder for knitting
```

```
## [1] "/Users/walkergrimshaw/Documents/Duke/Courses/Spring_2019/Environmental_Data_Analytics/Assignmen
```

```
suppressMessages(library(tidyverse)) # suppress messages because of knitting LaTex error
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
Ozone.2017.raw <- read.csv("../Data/Raw/EPAair_O3_NC2017_raw.csv", header = T)
Ozone.2018.raw <- read.csv("../Data/Raw/EPAair_O3_NC2018_raw.csv", header = T)
PM25.2017.raw <- read.csv("../Data/Raw/EPAair_PM25_NC2017_raw.csv", header = T)
PM25.2018.raw <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv", header = T)

#2
```

```
## dimensions
# dim(Ozone.2017.raw)
# dim(Ozone.2018.raw)
# dim(PM25.2017.raw)
# dim(PM25.2018.raw)

## column names
# colnames(Ozone.2017.raw)
# colnames(Ozone.2018.raw)
# colnames(PM25.2017.raw)
# colnames(PM25.2018.raw)

## summary statistics
# summary(Ozone.2017.raw)
# summary(Ozone.2017.raw$Site.Name)
# summary(Ozone.2018.raw)
# summary(PM25.2017.raw)
# summary(PM25.2018.raw)
```

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.

```
#3 change date to date format
class(Ozone.2017.raw$Date) # factor
```

```
## [1] "factor"
```

```
Ozone.2017.raw$Date <- as.Date(Ozone.2017.raw$Date, format = "%m/%d/%y")
class(Ozone.2017.raw$Date) # correctly reformatted to date
```

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

```
Ozone.2018.raw$Date <- as.Date(Ozone.2018.raw$Date, format = "%m/%d/%y")
PM25.2017.raw$Date <- as.Date(PM25.2017.raw$Date, format = "%m/%d/%y")
PM25.2018.raw$Date <- as.Date(PM25.2018.raw$Date, format = "%m/%d/%y")

#4
## use select() to select appropriate columns
Ozone.2017.processed <- select(Ozone.2017.raw,
                               Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                               COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
Ozone.2018.processed <- select(Ozone.2018.raw,
                               Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                               COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25.2017.processed <- select(PM25.2017.raw,
                               Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
                               COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
PM25.2018.processed <- select(PM25.2018.raw,
```

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

#5
## Replace data in AQS_parameter_desc column with "PM2.5"
PM25.2017.processed$AQS_PARAMETER_DESC <- "PM2.5"
## could also use PM25.2017.processed <- mutate(PM25.2017.processed, AQS_PARAMETER_DESC = "PM2.5")
PM25.2018.processed$AQS_PARAMETER_DESC <- "PM2.5"

#6
## don't keep row names
write.csv(x = Ozone.2017.processed, row.names = F,
          file = "../Data/Processed/EPAair_O3_NC2017_processed.csv")
write.csv(x = Ozone.2018.processed, row.names = F,
          file = "../Data/Processed/EPAair_O3_NC2018_processed.csv")
write.csv(x = PM25.2017.processed, row.names = F,
          file = "../Data/Processed/EPAair_PM25_NC2017_processed.csv")
write.csv(x = PM25.2018.processed, row.names = F,
          file = "../Data/Processed/EPAair_PM25_NC2018_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:

- Sites: Blackstone, Bryson City, Triple Oak
- Add columns for "Month" and "Year" by parsing your "Date" column (hint: `separate` function or `lubridate` package)

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
## Check column names are all the same
colnames(Ozone.2017.processed) == colnames(Ozone.2018.processed) &&
colnames(Ozone.2017.processed) == colnames(PM25.2017.processed) &&
colnames(Ozone.2017.processed) == colnames(PM25.2018.processed) # TRUE
```

```
## [1] TRUE
```

```
## rbind all 4 datasets
EPA.air <- rbind(Ozone.2017.processed, Ozone.2018.processed, PM25.2017.processed, PM25.2018.processed)

#8

EPA.air.wrangled <- EPA.air %>%
  ## filter by site name to select those rows
  filter(Site.Name == "Blackstone" | Site.Name == "Bryson City" | Site.Name == "Triple Oak") %>%
  ## mutate to separate year and month in new columns
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date))
```

```
#9 Spread dialy_aqi_value by aqs_parameter_description

EPA.air.tidy <- EPA.air.wrangled %>%
  spread(AQS_PARAMETER_DESC, DAILY_AQI_VALUE)

#10

dim(EPA.air.tidy)
```

```
## [1] 1953    9
```

```
#11

write.csv(x = EPA.air.tidy, row.names = F,
          file = "../Data/Processed/EPAair_O3_PM25_NC1718_Processed.csv")
```

Generate summary tables

12. Use the split-apply-combine strategy to generate two new data frames:
    a. A summary table of mean AQI values for O3 and PM2.5 by month
    b. A summary table of the mean, minimum, and maximum AQI values of O3 and PM2.5 for each site
13. Display the data frames.

```
#12a
EPA.air.monthly.summary <- EPA.air.tidy %>%
  group_by(Month) %>%
  summarize(meanO3 = mean(Ozone, na.rm = T),
            meanPM25 = mean(PM2.5, na.rm = T)
            )
#12b
EPA.air.site.summary <- EPA.air.tidy %>%
  group_by(Site.Name) %>%
  summarize(meanO3 = mean(Ozone, na.rm = T),
            minO3 = min(Ozone, na.rm = T),
            maxO3 = max(Ozone, na.rm = T),
            meanPM25 = mean(PM2.5, na.rm = T),
            minPM25 = min(PM2.5, na.rm = T),
            maxPM25 = max(PM2.5, na.rm = T)
            )
```

```
## Warning in min(Ozone, na.rm = T): no non-missing arguments to min;
## returning Inf
```

```
## Warning in max(Ozone, na.rm = T): no non-missing arguments to max;
## returning -Inf
```

```
#13
print(EPA.air.monthly.summary)
```

```
## # A tibble: 12 x 3
##    Month meanO3 meanPM25
##    <dbl>  <dbl>    <dbl>
##  1     1   31.5     34.6
##  2     2   35.5     36.7
```

```
##  3     3   42.4     35.1
##  4     4   44.3     32.5
##  5     5   38.9     31.7
##  6     6   38.7     33.3
##  7     7   38.2     33.1
##  8     8   34.0     33.7
##  9     9   32.6     31.9
## 10    10   32.1     29.3
## 11    11   30.1     36.8
## 12    12   29.8     41.1
```

```
print(EPA.air.site.summary)
```

```
## # A tibble: 3 x 7
##   Site.Name   meanO3 minO3 maxO3 meanPM25 minPM25 maxPM25
##   <fct>        <dbl> <dbl> <dbl>    <dbl>   <int>   <int>
## 1 Blackstone    38.5     8    97     36.7       0      83
## 2 Bryson City   35.2     5    71     32.3       3      78
## 3 Triple Oak   NaN     Inf  -Inf     33.5       0      74
```