Assignment 10: Data Scraping

Emily Guyu Yang

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Rename this file `<FirstLast>_A10_DataScraping.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 your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Check your working directory

```
#1
library(tidyverse);library(lubridate);library(viridis);library(here)
here()
```

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

```
library(rvest)

library(dataRetrieval)

library(tidycensus)

# Set theme
mytheme <- theme_gray() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)

here()
```

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

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2023 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Scroll down and select the LWSP link next to Durham Municipality.
- Note the web address: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023

Indicate this website as the as the URL to be scraped. (In other words, read the contents into an `rvest` webpage object.)

```
#2
website <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
```

3. The data we want to collect are listed below:

- From the "1. System Information" section:
- Water system name
- PWSID
- Ownership
- From the "3. Water Supply Sources" section:
- Maximum Day Use (MGD) - for each month

In the code chunk below scrape these values, assigning them to four separate variables.

> HINT: The first value should be "Durham", the second "03-32-010", the third "Municipality", and the last should be a vector of 12 numeric values (represented as strings)".

```
#3
water_system_name <- website %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>% html_text
water_system_name
```

```
## [1] "Durham"
```

```
PWSID <- website %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
PWSID
```

```
## [1] "03-32-010"
```

```
ownership <- website %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>% html_text()
ownership
```

```
## [1] "Municipality"
```

```
max_daily_use <- website %>% html_nodes('th~ td+ td') %>% html_text()
max_daily_use
```

```
##  [1] "28.9000" "33.3000" "43.7000" "30.0000" "40.0000" "37.2300" "34.2000"
##  [8] "44.9000" "40.3500" "30.9000" "56.7000" "33.3000"
```

4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   TIP: Use `rep()` to repeat a value when creating a dataframe.

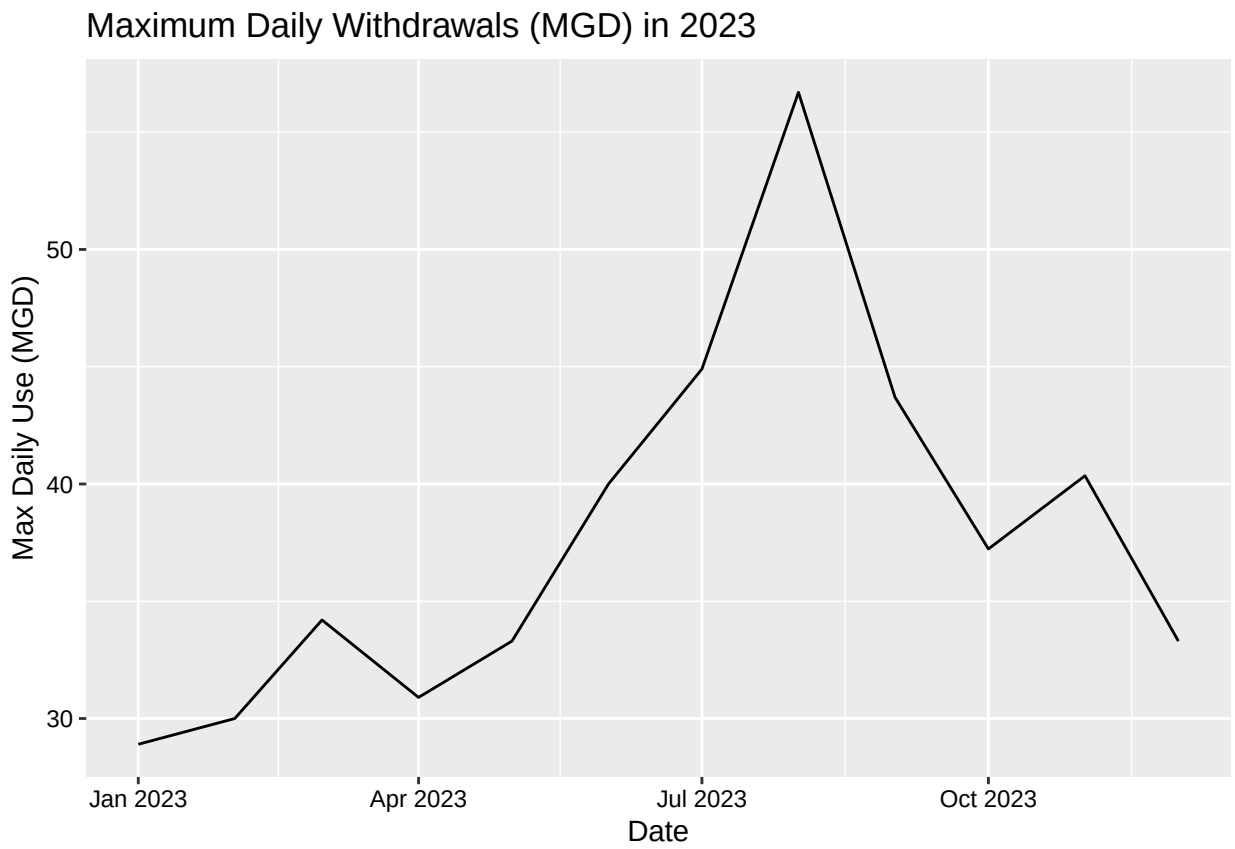
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc... Or, you could scrape month values from the web page...

5. Create a line plot of the maximum daily withdrawals across the months for 2023, making sure, the months are presented in proper sequence.

```
#4
Month <- website %>% html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>% html_text()

water_df <- data.frame(
  Water_system = rep(water_system_name, length(Month)),
  PWSID =  rep(PWSID, length(Month)),
  ownership = rep(ownership, length(Month)),
  max_daily_use = as.numeric(max_daily_use),
  Month = Month
) %>%
  mutate(Year = 2023,
         Month = factor(Month, levels = month.abb),
         Date = as.Date(paste(Year, Month, "1", sep = "-"),
                        format = "%Y-%b-%d")) %>%
  arrange(Date)


#5
ggplot(water_df, aes(x = Date, y = max_daily_use)) +
  geom_line() +
  labs(
    title = "Maximum Daily Withdrawals (MGD) in 2023",
    x = "Date",
    y = "Max Daily Use (MGD)") +
  mytheme
```



6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
scrape.it <- function(Year, PWSID){
  the_url <-
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',PWSID, '&year=', Year)
    print(paste("Scraping URL:", the_url))

    the_website <- read_html(the_url)

    water_system_name <- the_website %>%
      html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
      html_text()
    PWSID <- the_website %>%
      html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
      html_text()
    ownership <- the_website %>%
      html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
      html_text()
    max_daily_use <- the_website %>%
      html_nodes('th~ td+ td') %>%
      html_text()
```

```
    Month <- the_website %>%
      html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
      html_text()

    water_df <- data.frame(
      Water_system = rep(water_system_name, length(Month)),
      PWSID = rep(PWSID, length(Month)),
      ownership = rep(ownership, length(Month)),
      max_daily_use = as.numeric(max_daily_use),
      Month = Month
    ) %>%
      mutate(
        Year = Year,
        Month = factor(Month, levels = month.abb),
        Date = as.Date(paste(Year, Month, "1", sep = "-"), format = "%Y-%b-%d")
        ) %>%
      arrange(Date)

  return(water_df)
}
```

7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

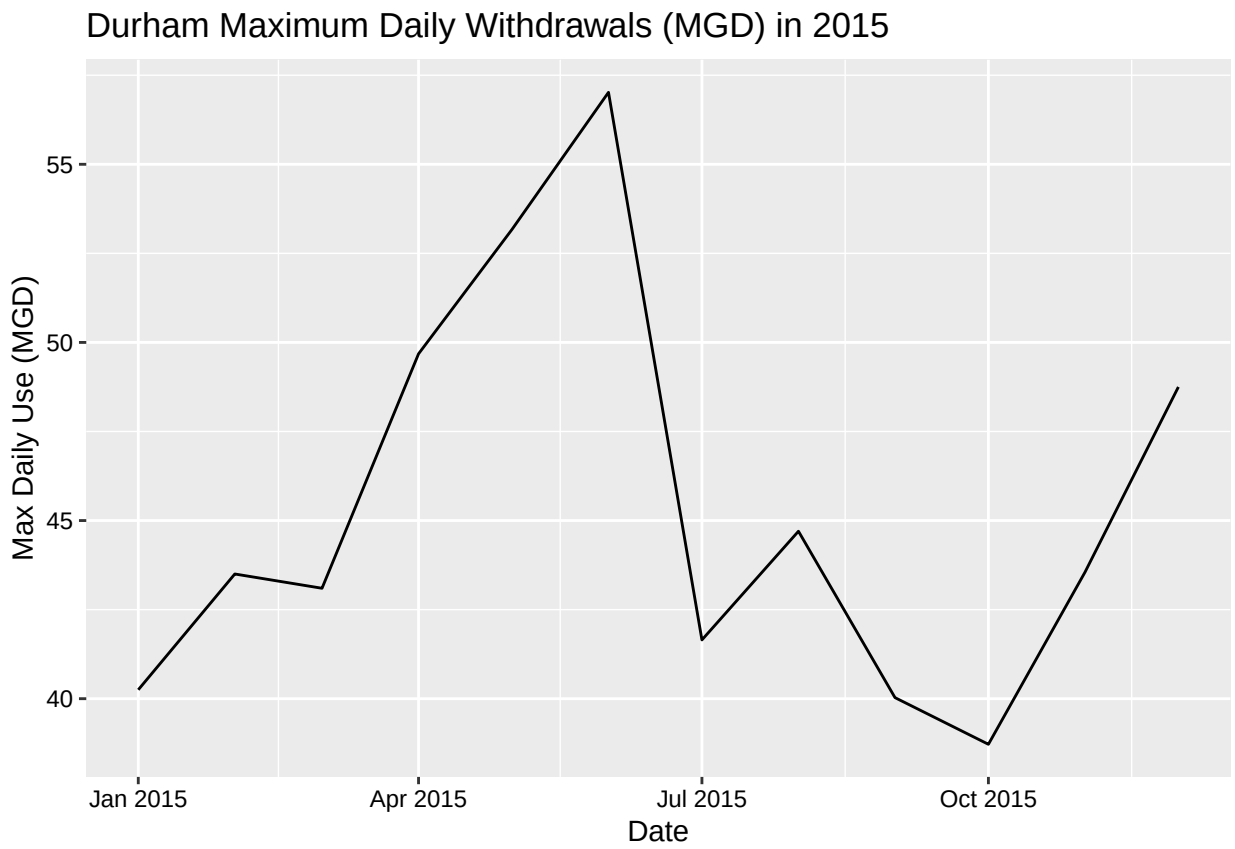
```
#7
water_data_Durham <- scrape.it(Year = 2015, PWSID = "03-32-010")
```

```
## [1] "Scraping URL: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2015"
```

```
head(water_data_Durham)
```

```
##   Water_system     PWSID    ownership max_daily_use Month Year       Date
## 1       Durham 03-32-010 Municipality         40.25   Jan 2015 2015-01-01
## 2       Durham 03-32-010 Municipality         43.50   Feb 2015 2015-02-01
## 3       Durham 03-32-010 Municipality         43.10   Mar 2015 2015-03-01
## 4       Durham 03-32-010 Municipality         49.68   Apr 2015 2015-04-01
## 5       Durham 03-32-010 Municipality         53.17   May 2015 2015-05-01
## 6       Durham 03-32-010 Municipality         57.02   Jun 2015 2015-06-01
```

```
ggplot(water_data_Durham, aes(x = Date, y = max_daily_use)) +
  geom_line() +
  labs(
    title = "Durham Maximum Daily Withdrawals (MGD) in 2015",
    x = "Date",
    y = "Max Daily Use (MGD)") +
  mytheme
```



8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8
#water_data_Asheville <- scrape.it(Year = 2015, PWSID = "01-11-010")
#head(water_data_Asheville)
# I'm not sure why, but this doesn't retrieve values for Asheville.
#I tried multiple times, and the function doesn't work for both Durham and Asheville,
# even with the correct scraping URL.
#I had to write another function specifically for Asheville to retrieve the data:

website.Ash <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2015')

scrape.it.Ash <- function(Year){
  the_url <-
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=', Year)
    print(paste("Scraping URL:", the_url))

    website.Ash <- read_html(the_url)

    water_system_name <- website.Ash %>%
      html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
      html_text()
    PWSID <- website.Ash %>%
```

```
      html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
      html_text()
    ownership <- website.Ash %>%
      html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
      html_text()
    max_daily_use <- website.Ash %>%
      html_nodes('th~ td+ td') %>%
      html_text()
    Month <- website.Ash %>%
      html_nodes(".fancy-table:nth-child(30) tr+ tr th") %>%
      html_text()

    water_df <- data.frame(
      Water_system = rep(water_system_name, length(Month)),
      PWSID = rep(PWSID, length(Month)),
      ownership = rep(ownership, length(Month)),
      max_daily_use = as.numeric(max_daily_use),
      Month = Month
    ) %>%
      mutate(
        Year = Year,
        Month = factor(Month, levels = month.abb),
        Date = as.Date(paste(Year, Month, "1", sep = "-"), format = "%Y-%b-%d")
        ) %>%
      arrange(Date)

  return(water_df)
}

water_data_Asheville <- scrape.it.Ash(Year = 2015)
```

```
## [1] "Scraping URL: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2015"
```

```
head(water_data_Asheville)
```

```
##   Water_system     PWSID    ownership max_daily_use Month Year       Date
## 1    Asheville 01-11-010 Municipality         20.81   Jan 2015 2015-01-01
## 2    Asheville 01-11-010 Municipality         24.54   Feb 2015 2015-02-01
## 3    Asheville 01-11-010 Municipality         21.42   Mar 2015 2015-03-01
## 4    Asheville 01-11-010 Municipality         21.60   Apr 2015 2015-04-01
## 5    Asheville 01-11-010 Municipality         23.95   May 2015 2015-05-01
## 6    Asheville 01-11-010 Municipality         23.53   Jun 2015 2015-06-01
```

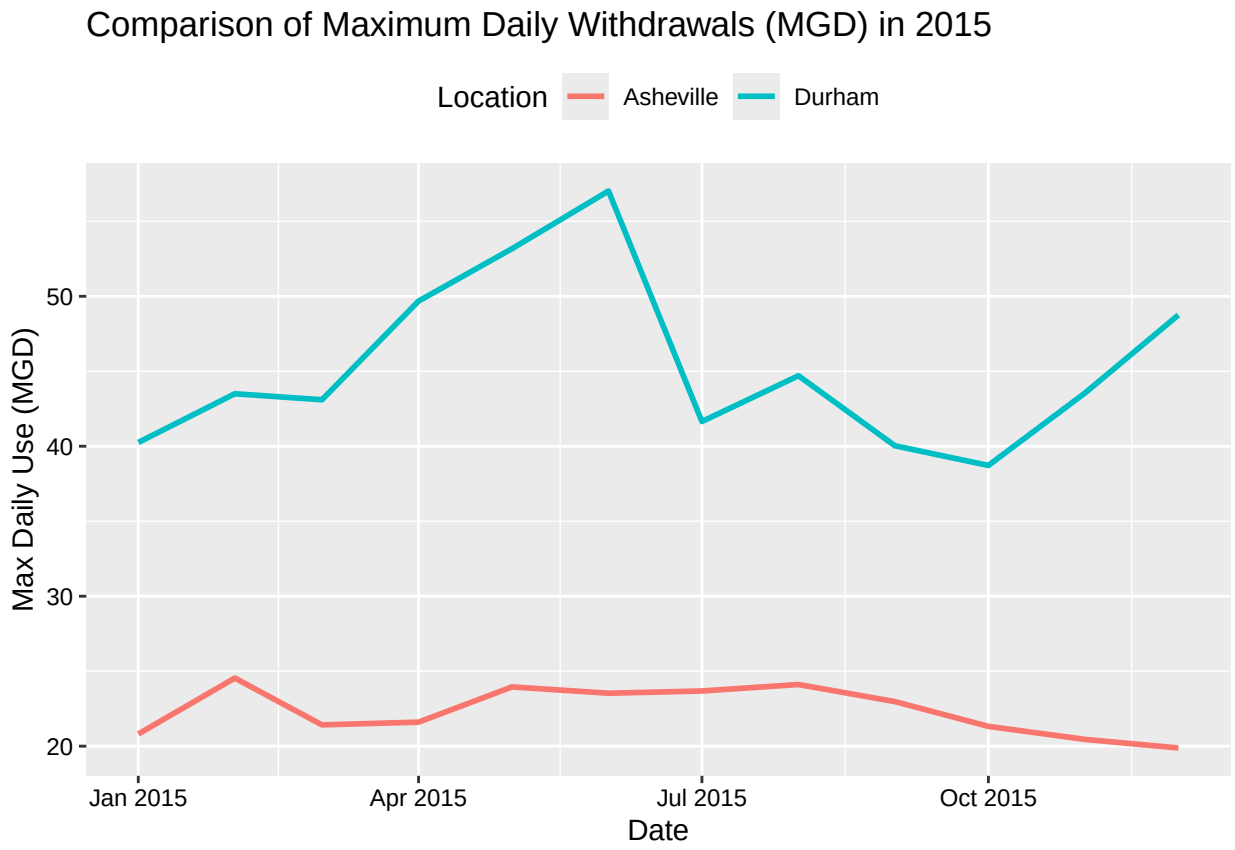
```
#Combing the data
durham_data <- water_data_Durham %>%
  select(Date, max_daily_use) %>%
  rename(Durham_Max_Use = max_daily_use)

asheville_data <- water_data_Asheville %>%
  select(Date, max_daily_use) %>%
  rename(Asheville_Max_Use = max_daily_use)
```

```r
combined_data <- full_join(durham_data, asheville_data, by = "Date")

#plot the data
ggplot(combined_data, aes(x = Date)) +
  geom_line(aes(y = Durham_Max_Use, color = "Durham"), size = 1) +
  geom_line(aes(y = Asheville_Max_Use, color = "Asheville"), size = 1) +
  labs(
    title = "Comparison of Maximum Daily Withdrawals (MGD) in 2015",
    x = "Date",
    y = "Max Daily Use (MGD)",
    color = "Location"
  ) +
  mytheme
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022. Add a smoothed line to the plot (method = 'loess').

   TIP: See Section 3.2 in the "10_Data_Scraping.Rmd" where we apply "map2()" to iteratively

run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```
#9
scrape.it.Ash <- function(Year){
  the_url <-
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=',
           Year)

  print(paste("Scraping URL:", the_url))

    website.Ash <- read_html(the_url)

    water_system_name <- website.Ash %>%
      html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
      html_text()
    PWSID <- website.Ash %>%
      html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
      html_text()
    ownership <- website.Ash %>%
      html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
      html_text()
    max_daily_use <- website.Ash %>%
      html_nodes('th~ td+ td') %>%
      html_text()
    Month <- website.Ash %>%
      html_nodes(".fancy-table:nth-child(30) tr+ tr th") %>%
      html_text()

    water_df <- data.frame(
      Water_system = rep(water_system_name, length(Month)),
      PWSID = rep(PWSID, length(Month)),
      ownership = rep(ownership, length(Month)),
      max_daily_use = as.numeric(max_daily_use),
      Month = Month
    ) %>%
      mutate(
        Year = Year,
        Month = factor(Month, levels = month.abb),
        Date = as.Date(paste(Year, Month, "1", sep = "-"), format = "%Y-%b-%d")
        ) %>%
      arrange(Date)

  return(water_df)
}

#scraping only works for 2019 and 2020.
#asheville_2018 <- scrape.it.Ash(Year = 2018)
asheville_2019 <- scrape.it.Ash(Year = 2019)
```

```
## [1] "Scraping URL: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2019"
```

```
asheville_2020 <- scrape.it.Ash(Year = 2020)
```

```
## [1] "Scraping URL: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2020"
```

```
#asheville_2021 <- scrape.it.Ash(Year = 2021)
#asheville_2022 <- scrape.it.Ash(Year = 2022)
```

Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time? > Answer: >