Assignment 10: Data Scraping

Allison Barbaro

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(rvest)
library(dplyr)
library(ggplot2)
library(lubridate)

getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2022 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=2022

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

```
#2
the_URL <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
```

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), with the first value being "27.6400".

```
#3
water.system.name <- the_URL %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

PWSID <- the_URL %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

ownership <- the_URL %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()

max.withdrawals.mgd <- the_URL %>%
  html_nodes("th~ td+ td") %>%
  html_text()

max.withdrawals.mgd <- as.numeric(max.withdrawals.mgd)
```

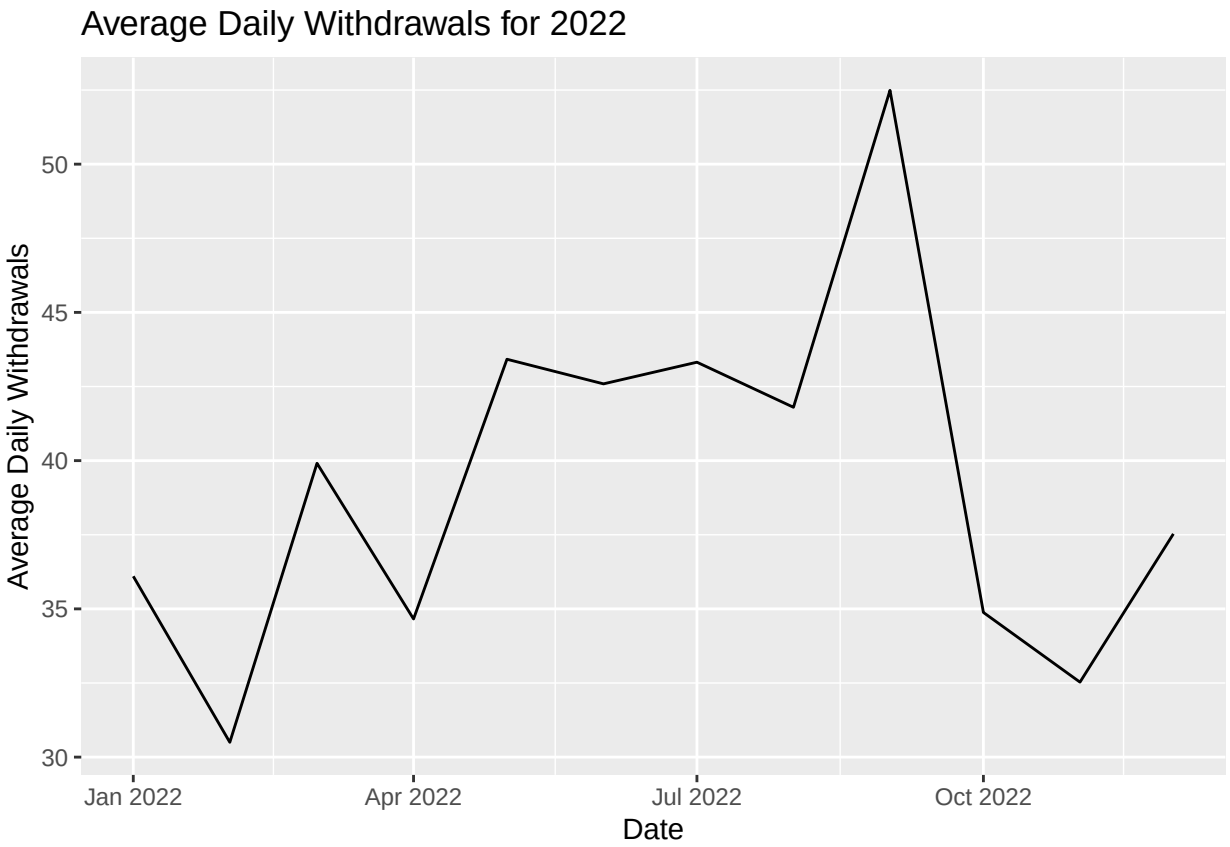
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

> 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 average daily withdrawals across the months for 2022

```
#4

Date <- c("01/2022", "05/2022", "09/2022", "02/2022",
          "06/2022", "10/2022", "03/2022", "07/2022",
          "11/2022", "04/2022", "08/2022", "12/2022")

df_withdrawals <- data.frame("System Name" = rep(water.system.name),
                             "PWSID" = rep(PWSID),
                             "Ownership" = rep(ownership),
                             "Month" = c("Jan", "May","Sept",
                                         "Feb", "June", "Oct", "Mar",
                                         "July", "Nov", "Apr", "Aug",
                                         "Dec"),
                             "Year" = rep(2022),
                             "Date" = Date,
                             "Max Day Use" = max.withdrawals.mgd)

df_withdrawals$Date <- my(df_withdrawals$Date)

#5
ggplot(df_withdrawals) +
  geom_line(aes(x = Date, y = Max.Day.Use)) +
  labs(x = "Date", y = "Average Daily Withdrawals",
       title = "Average Daily Withdrawals for 2022")
```



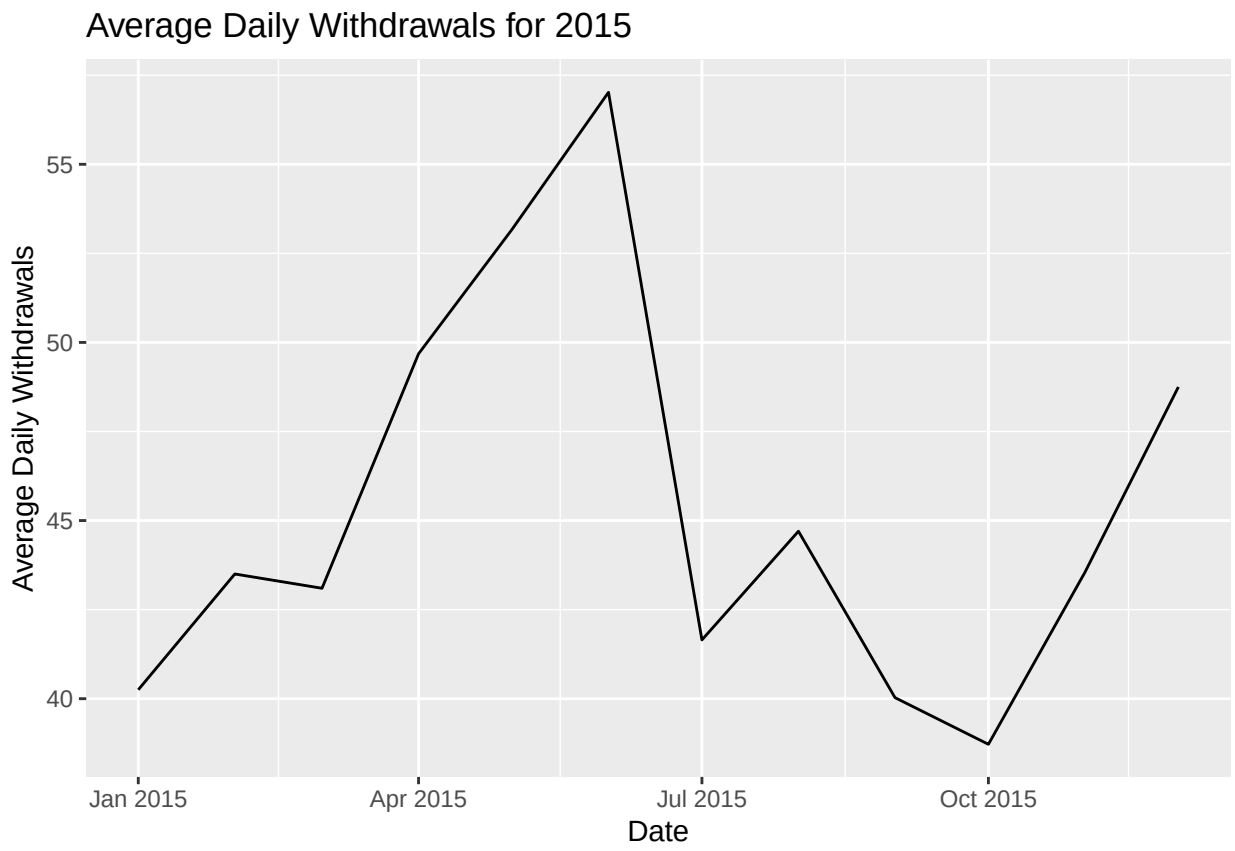
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
scrape.it <- function(the_PWSID, the_year){

the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                                the_PWSID, '&year=', the_year))

#Scrape the data
water.system.name.1 <- the_website %>%
  html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()

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

ownership.1 <- the_website %>%
  html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()

max.withdrawals.mgd.1 <- the_website %>%
  html_nodes('th~ td+ td') %>%
```

```
  html_text()

max.withdrawals.mgd.1 <- as.numeric(max.withdrawals.mgd.1)

#Convert to a dataframe

Month <- c('Jan', 'May', 'Sep', 'Feb', 'Jun',
           'Oct', 'Mar', 'Jul', 'Nov', 'Apr', 'Aug', 'Dec')

df_withdrawals_function <- data.frame("System Name" = rep(water.system.name.1),
                              "PWSID" = rep(PWSID.1),
                              "Ownership" = rep(ownership.1),
                              "Month" = Month,
                              "Year" = rep(the_year),
                              "Date" = my(paste0(Month, "-", the_year)),
                              "Max Day Use" = max.withdrawals.mgd.1)
df_withdrawals_function <- arrange(df_withdrawals_function, Date)
}
```

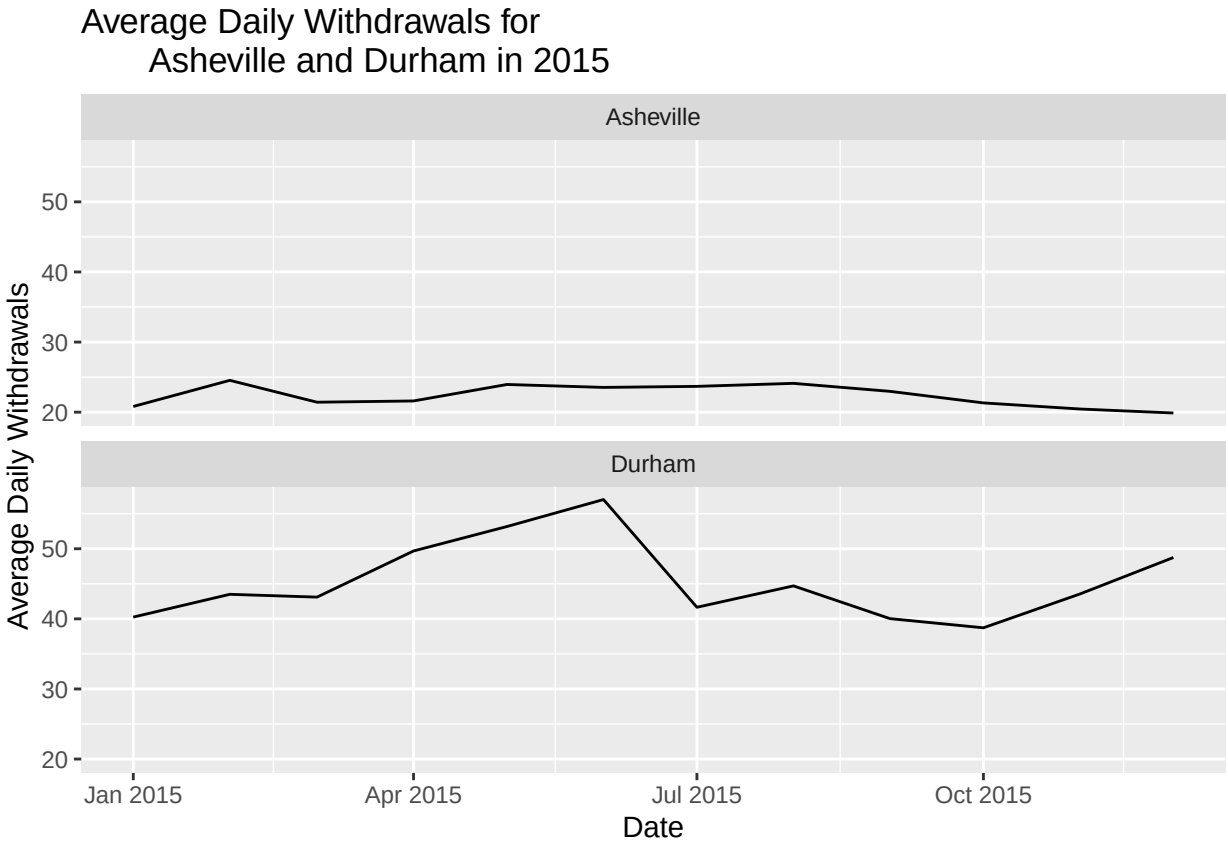
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
dataframe.2015 <- scrape.it("03-32-010", 2015)

ggplot(dataframe.2015) +
  geom_line(aes(x = Date, y = Max.Day.Use)) +
  labs(x = "Date", y = "Average Daily Withdrawals",
       title = "Average Daily Withdrawals for 2015")
```



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
Asheville.2015 <-scrape.it('01-11-010', 2015)
view(Asheville.2015)

Asheville.Durham <- rbind(Asheville.2015, dataframe.2015)

ggplot(Asheville.Durham) +
  geom_line(aes(x = Date, y = Max.Day.Use)) +
  labs(x = "Date", y = "Average Daily Withdrawals",
       title =
         "Average Daily Withdrawals for
       Asheville and Durham in 2015") +
  facet_wrap(vars(System.Name), nrow = 2)
```

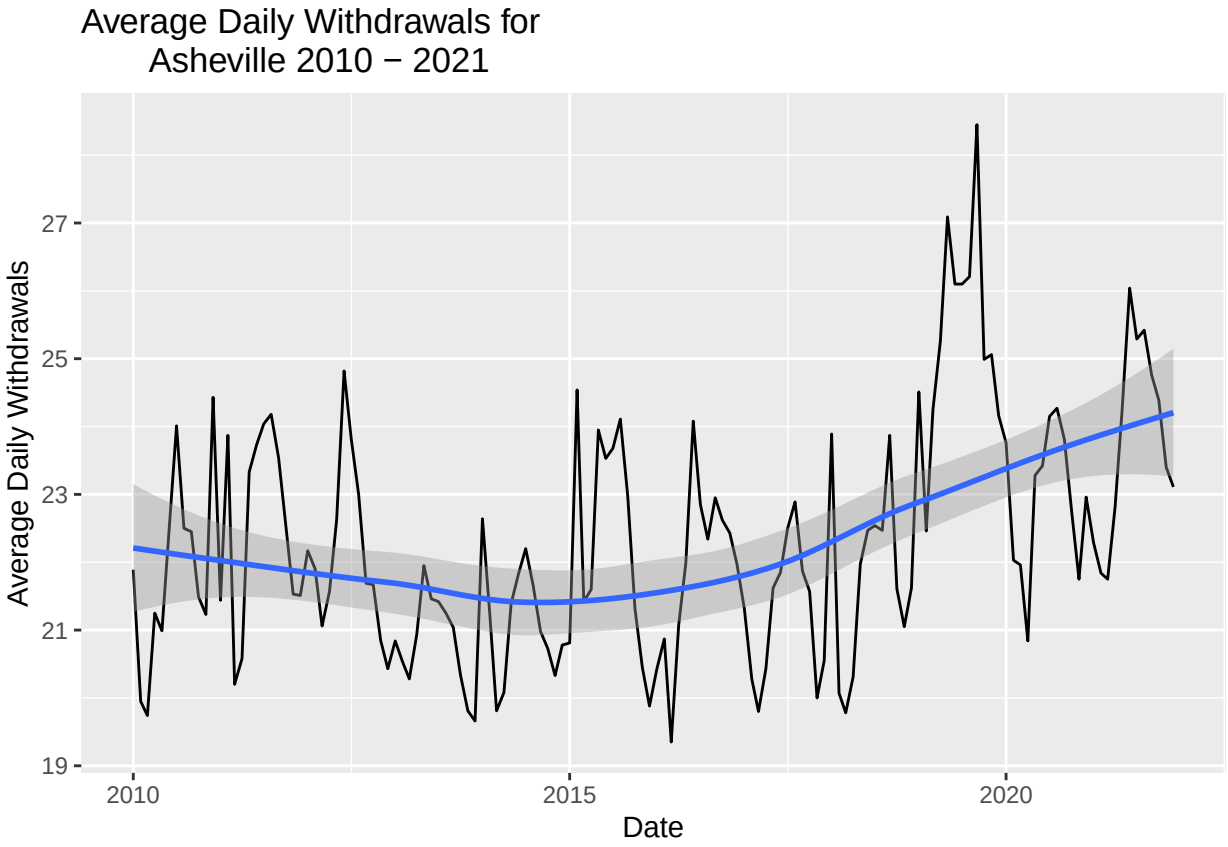


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   TIP: See Section 3.2 in the "09_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
the_years <- c(2010, 2011, 2012, 2013, 2014, 2015, 2016,
               2017, 2018, 2019, 2020, 2021)

the_sites <- rep('01-11-010', length(the_years))

Asheville.dfs <- map2(the_sites, the_years, scrape.it)

Asheville.combined <- bind_rows(Asheville.dfs)

ggplot(Asheville.combined, aes(x = Date, y = Max.Day.Use)) +
  geom_line() +
  geom_smooth(method = 'loess') +
  labs(x = "Date", y = "Average Daily Withdrawals",
       title =
         "Average Daily Withdrawals for
       Asheville 2010 - 2021")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



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

Asheville's water usage is increasing over time.