# Assignment 10: Data Scraping

Kendall Barton

## 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(rvest)
library(tidyverse)
library(ggplot2)

getwd()
```

```
## [1] "/Users/kendallbarton/Downloads/EDE_Fall2023"
```

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.)

```r
#2
webpage <- 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)".

```r
#3
system_name <- webpage %>%
        html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
        html_text()
#print(system_name)

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

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

max_day_use <- webpage %>%
        html_nodes("th~ td+ td") %>%
        html_text()
#print(max_day_use)
max_day_use <- as.numeric(max_day_use)
```

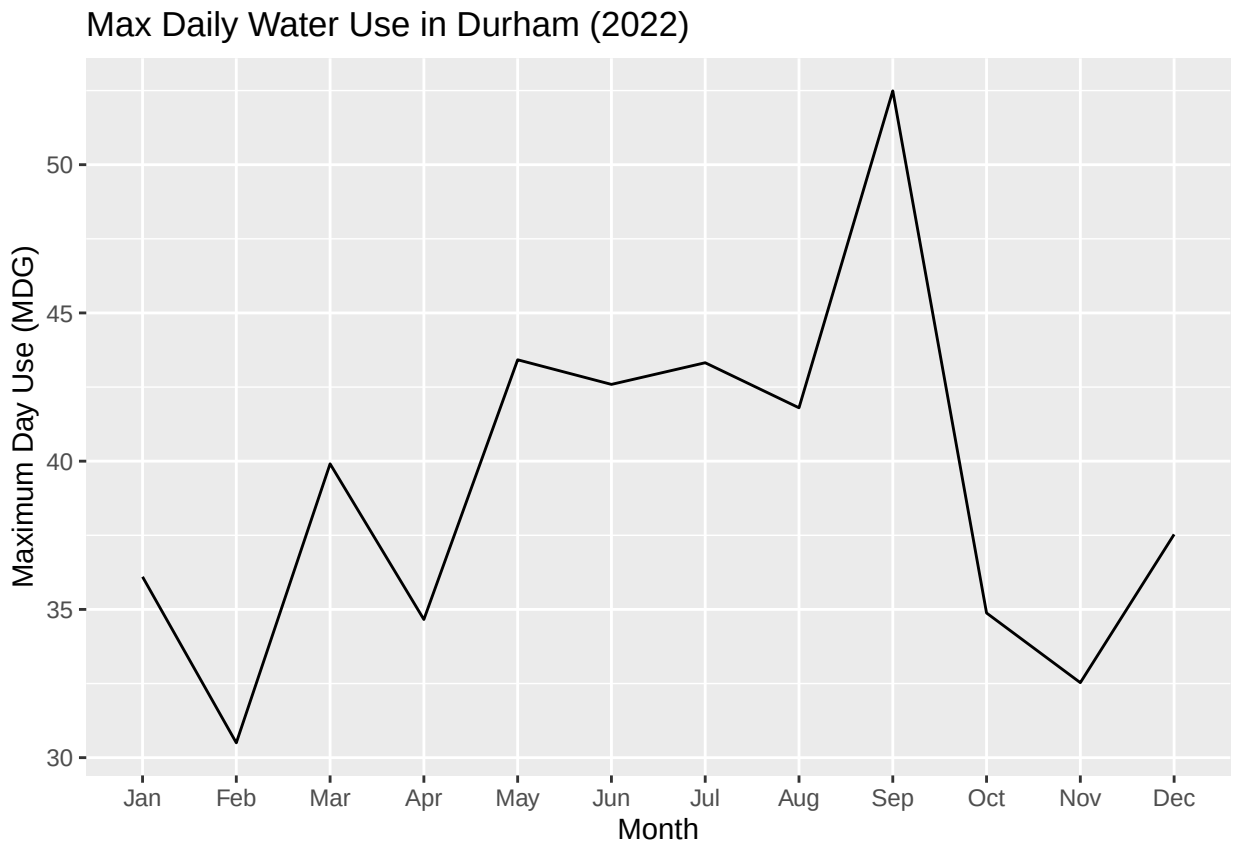
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   > 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 2022

```
#4
month_name <- webpage %>%
        html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
        html_text()
month_name <- factor(month_name, levels = month.abb)

water_df <- data.frame(rep(system_name, length(max_day_use)), #make df
                       rep(PWSID, length(max_day_use)), rep(ownership, length(max_day_use)),
                       max_day_use, month_name)

water_df <- water_df %>% #get date
        mutate("Month_Year" = make_date(month = match(water_df$month_name, month.abb),
                                                year = 2022))

#5
ggplot(water_df, aes(y = max_day_use, x = month_name)) + #graph
        geom_line(aes(group = 1)) +
        labs(y = "Maximum Day Use (MDG)", x = "Month",
             title = "Max Daily Water Use in Durham (2022)")
```

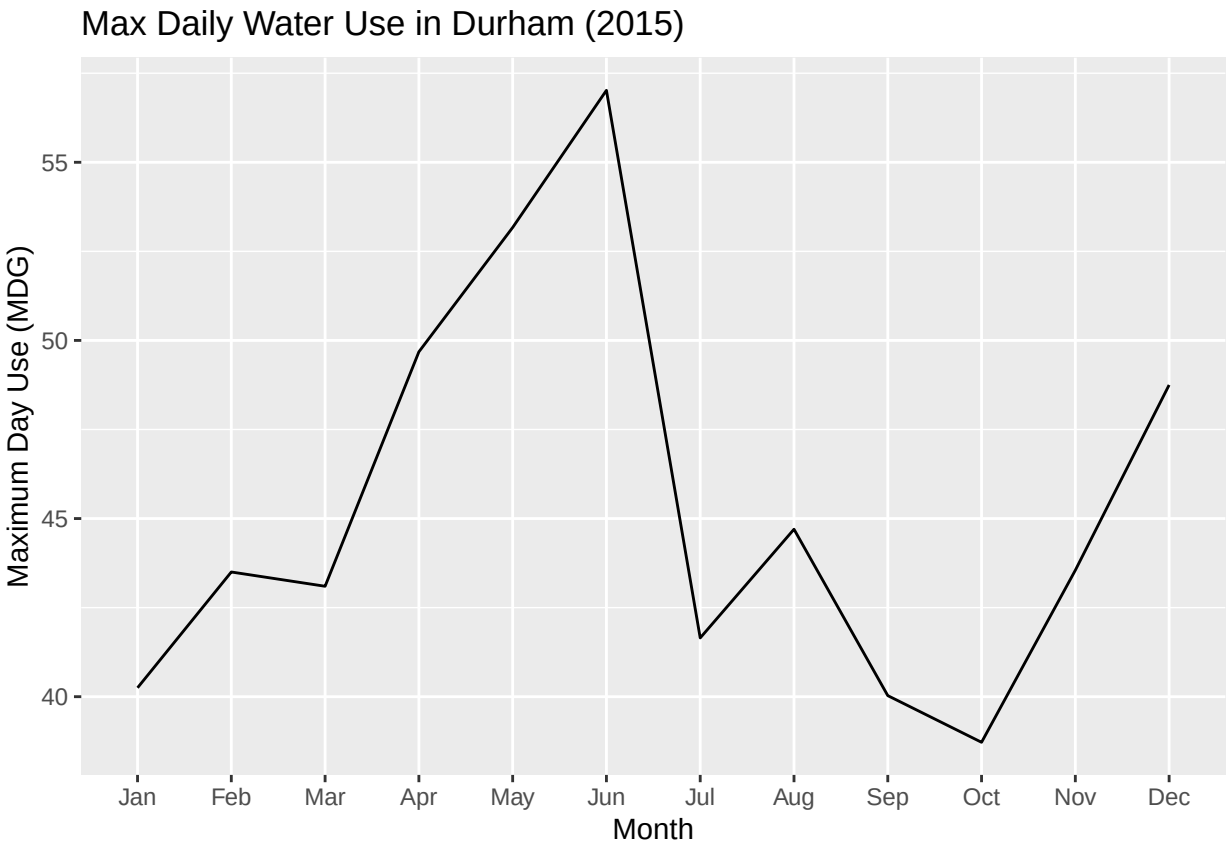



6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```r
#6.
#I'm using a for loop for this function

scrape_data <- function(ID, Year, retrival_df = to_retrieve) {
  #retrival_df makes it easy to get different types of data columns by changing df if you want
  #I set the default retrival_df to to_retrieve to make sure function will work with map2
  webp <- read_html(paste('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                          ID,'&year=',Year, sep = "")) #webpage changes depending on parameters
  comp <- webp %>%
    html_nodes("h6+ div") %>%
    html_text()
    df_result <- data.frame(c(seq(1,12))) #create inital df with correct number of rows
    for (idx in seq(1,nrow(retrival_df))) {
      #iterate through each column we want (from retrvial_df)
      cur_data <- webp %>% #get data
        html_nodes(retrival_df[idx,2]) %>%
        html_text()
      if (length(cur_data) == 1) { #rep if scraped data is just one thing (like owner)
        cur_data <- rep(cur_data, 12)
      }
      if (retrival_df[idx, 1] == "max_day_use") { #must be numeric to plot
        cur_data <- as.numeric(cur_data)
      }
      #print(cur_data)
      df_result <- cbind(df_result, cur_data) #put current data into df as column
    }
    colnames(df_result) <- c("ID_init", retrival_df[,1]) #name df using retrival_df
    #month can't be scraped because its postion changes depending on page
    df_result$month_name <- factor(c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul",
                                     "Nov", "Apr", "Aug", "Dec"), levels = month.abb)
    df_result$Month_Year <- make_date(month = match(df_result$month_name, month.abb),
                                          year = Year) #create date column
    return(df_result)
}

#this is the retrival_df
to_retrieve <- data.frame(c("system_name", "PWSID", "ownership", "max_day_use"),
                          c("div+ table tr:nth-child(1) td:nth-child(2)",
                            "td tr:nth-child(1) td:nth-child(5)",
                            "div+ table tr:nth-child(2) td:nth-child(4)", "th~ td+ td"))
```

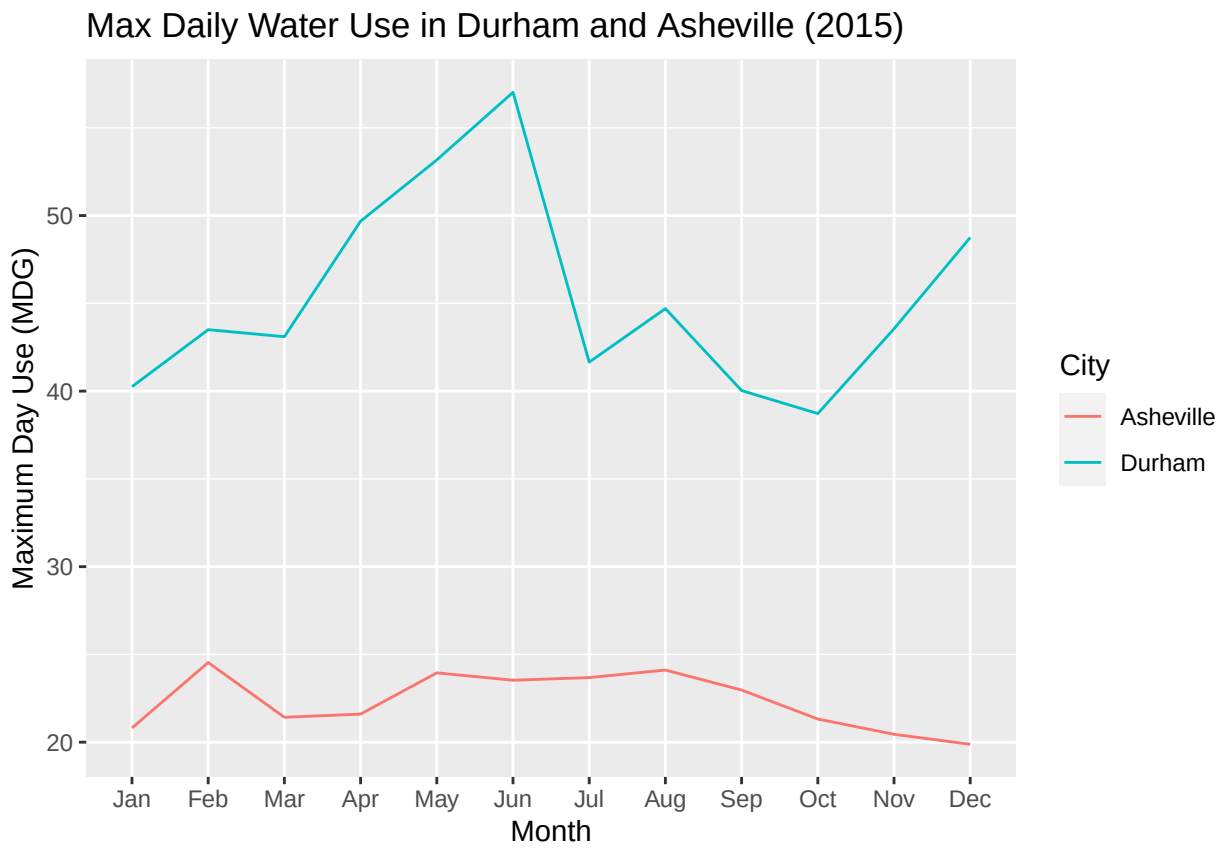
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```r
#7

Durham.2015 <- scrape_data("03-32-010", "2015") #get data

ggplot(Durham.2015, aes(y = max_day_use, x = month_name)) +
        geom_line(aes(group = 1)) +
        labs(y = "Maximum Day Use (MDG)", x = "Month",
             title = "Max Daily Water Use in Durham (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_data("01-11-010", "2015", to_retrieve)

DA.2015 <- rbind(Durham.2015, Asheville.2015) #combine city dfs

ggplot(DA.2015, aes(y = max_day_use, x = month_name)) +
  geom_line(aes(color = system_name, group = system_name)) +
  labs(y = "Maximum Day Use (MDG)", x = "Month",
       title = "Max Daily Water Use in Durham and Asheville (2015)", color = "City")
```



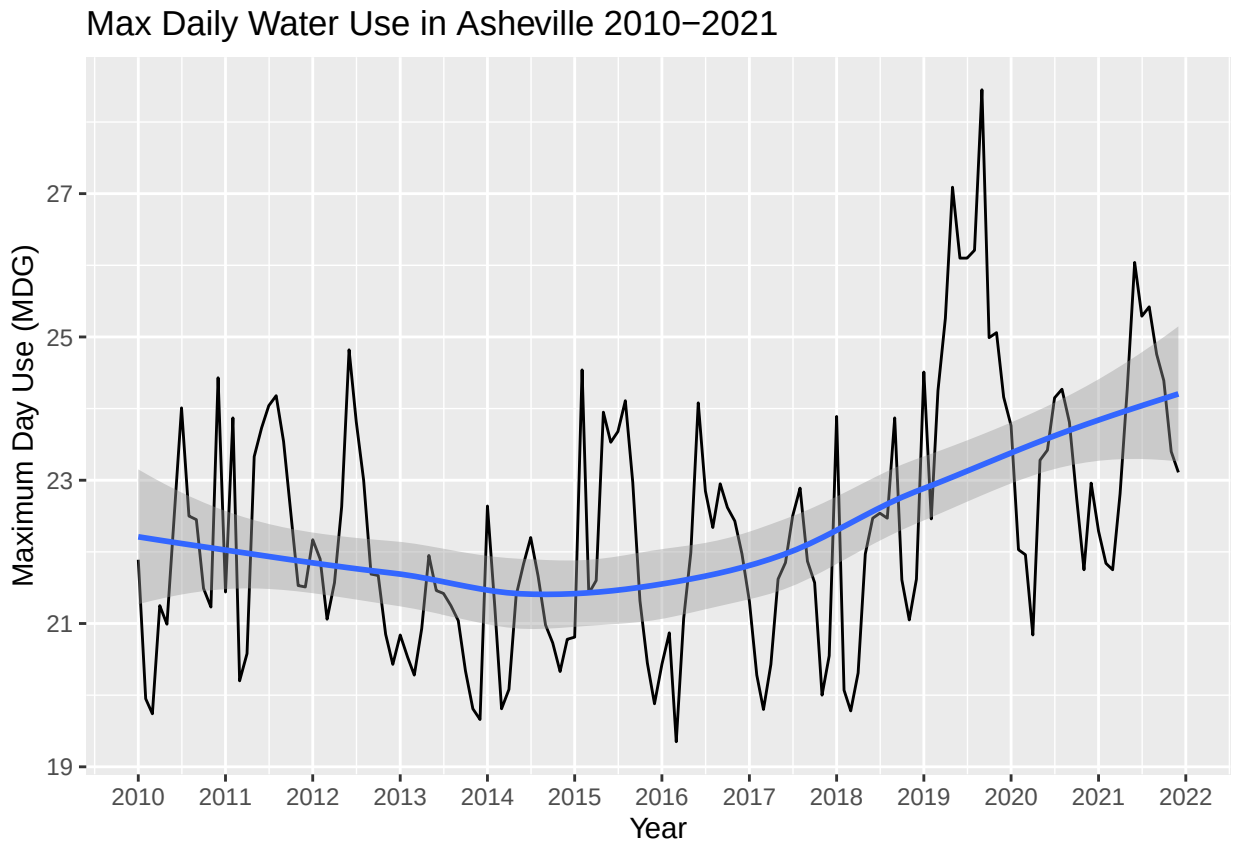

9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   TIP: See Section 3.2 in the "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
#I made two implementations and tried to determine which is more efficient,
#but which is faster seemed to be inconsistent: 3/5 times for loop was faster, 2/5 map2 was

#start.time.1 <- Sys.time()

Asheville.long <- data.frame()
for(year in seq(2010,2021)) {
  year_data <- scrape_data("01-11-010", year, to_retrieve)
  Asheville.long <- rbind(Asheville.long, year_data)
}

#end.time.1 <- Sys.time()
#how.long.1 <- (end.time.1 - start.time.1)

#start.time.2 <- Sys.time()

years <- seq(2010,2021)
```

```
place <- rep("01-11-010", length(years))
Asheville.dfs <- map2(place, years, scrape_data)
Asheville.long2 <- bind_rows(Asheville.dfs)

#end.time.2 <- Sys.time()
#how.long.2 <- (end.time.2 - start.time.2)

#print(how.long.1)
#print(how.long.2)

ggplot(Asheville.long2, aes(y = max_day_use, x = Month_Year)) +
  geom_line(aes(group = 1)) +
  labs(y = "Maximum Day Use (MDG)", x = "Year",
       title = "Max Daily Water Use in Asheville 2010-2021") +
  scale_x_date(date_labels = "%Y", date_breaks = "years") +
  geom_smooth(method = "loess")
```

```
## `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? > Answer: It looks like Asheville was slightly decreasing in water useage from from 2010 to about 2015, and has been increasing since then. >