# Assignment 09: Data Scraping

Lydie Costes

# Total points:

## OVERVIEW

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

## Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_09_Data_Scraping.Rmd") prior to submission.

## Set up

1. Set up your session:

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

```
#1. Check working directory and load packages
getwd()
```

```
## [1] "/Users/lydiecostes/Documents/Duke/DataAnalytics/GithubRepos/Environmental_Data_Analytics_2022/As
```

```
library(tidyverse)
library(lubridate)
library(rvest)

theme_set(theme_bw())
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2019 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Change the date from 2021 to 2020 in the upper right corner.

- 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=2020

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

```
#2. Read in the website
the_website <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020")
```

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

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

In the code chunk below scrape these values, assigning them to three 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, with the first value being 36.0100.

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

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

```
pswid <- the_website %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
pswid
```

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

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

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

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

```
##  [1] "36.0100" "36.9800" "41.6900" "32.0500" "40.6100" "40.5600" "37.2900"
##  [8] "43.6300" "33.3200" "32.3700" "41.9300" "28.0600"
```

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.

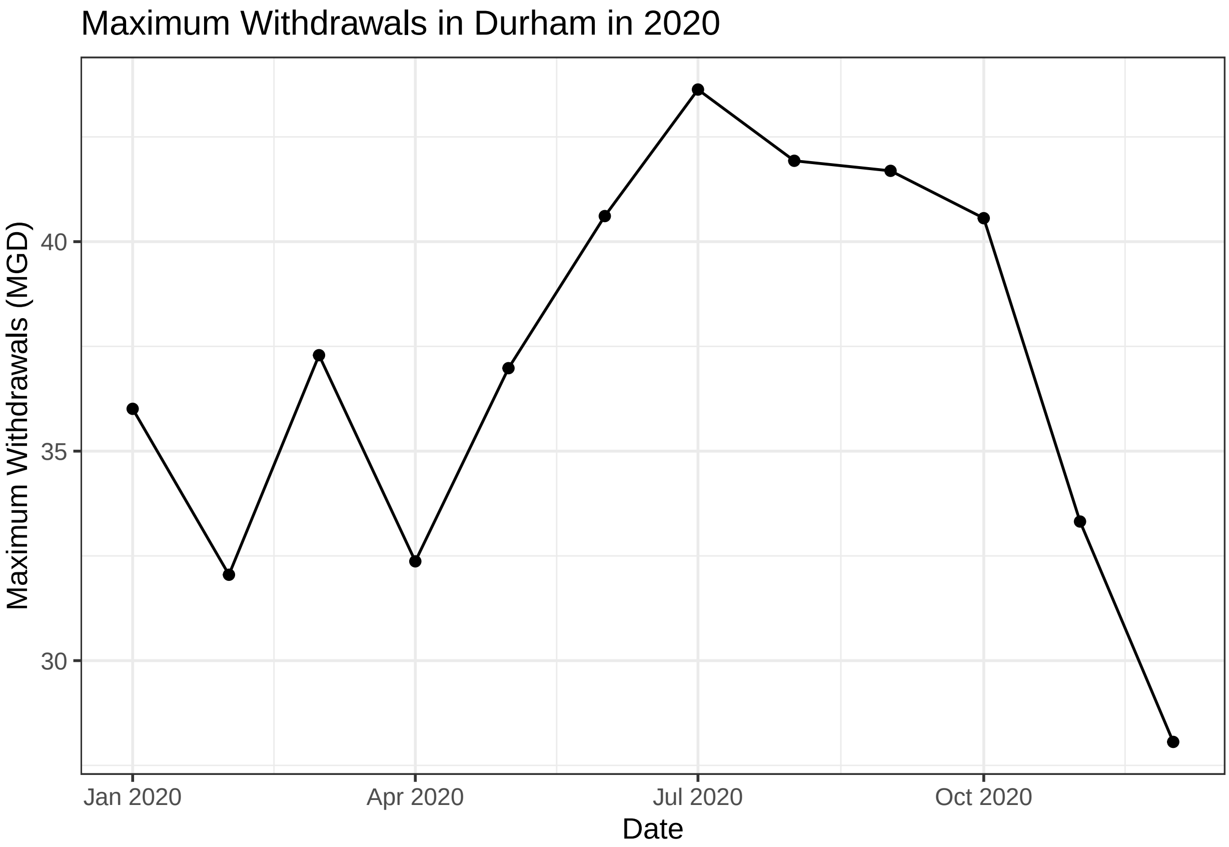
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc…

5. Plot the max daily withdrawals across the months for 2020

```r
#4. Convert data into a dataframe
# Create month and year lists
month <- (c("01", "05", "09", "02", "06", "10",
            "03", "07", "11", "04", "08", "12"))
year <- as.character("2020")

# merge lists into a dataframe
Durham_h2o <- data_frame(month, rep(year), rep(water.system.name),
                         rep(pswid), rep(ownership), max.withdrawals.mgd)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```r
# rename the columns
names(Durham_h2o) <- c("Month", "Year", "WaterSystem", "PSWID",
                       "Ownership", "MaxWithdrawals")

# convert flow data to numeric
Durham_h2o$MaxWithdrawals <- as.numeric(Durham_h2o$MaxWithdrawals)

# add a date column
Durham_h2o <- Durham_h2o %>%
  mutate(Date = my(paste(Month, "-", Year)))

#5. Graph withdrawals by date
ggplot(Durham_h2o, aes(x = Date, y = MaxWithdrawals, group = 1)) +
  geom_line() +
  geom_point() +
  labs(y  = "Maximum Withdrawals (MGD)", title = "Maximum Withdrawals in Durham in 2020")
```

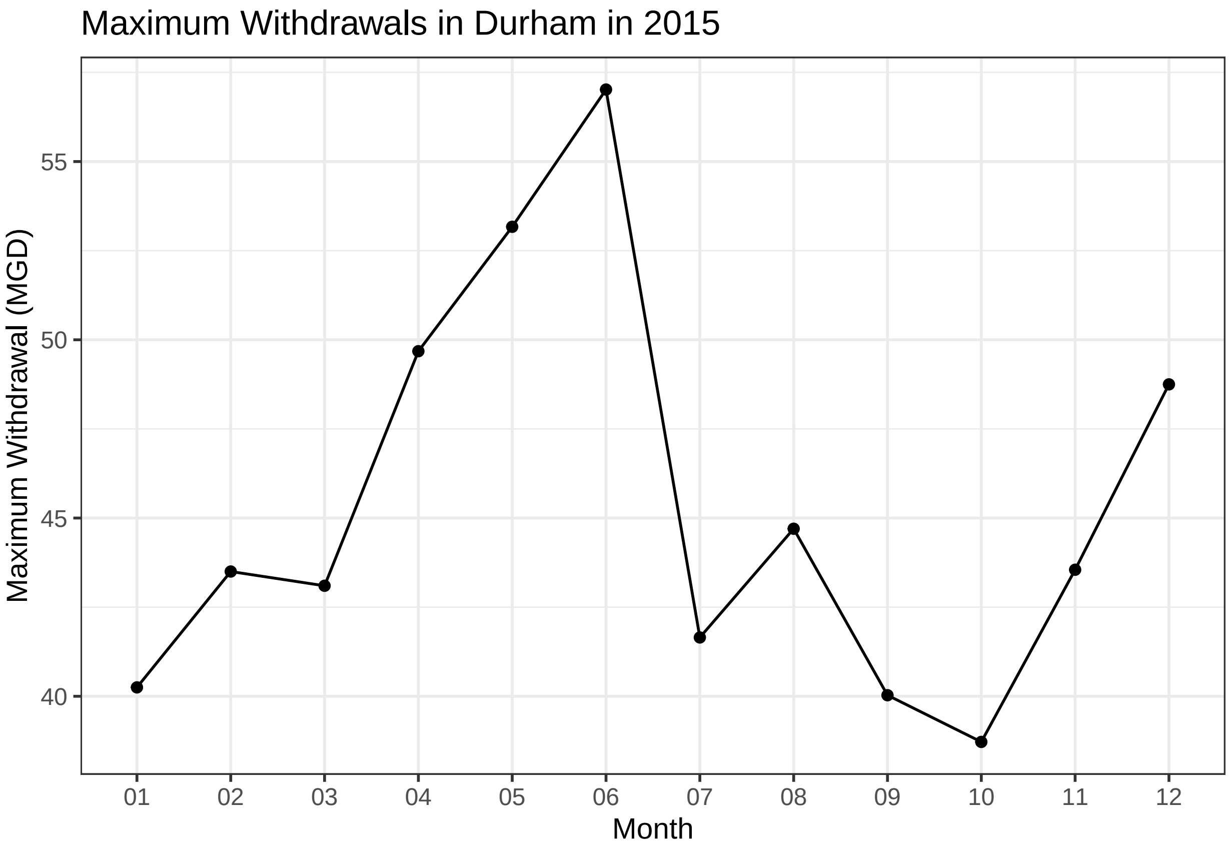


6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
#6. Construct a function to scrape DEQ data using PWSID and year
scrape.it <- function(the_pwsid, the_year){
  #Get the proper url
  the_url <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                              the_pwsid, '&year=', the_year))

  #Fetch the website
the_website  <- the_url

  #Scrape the data
water.system.name <- the_website %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
pswid <- 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.withdrawals.mgd <- the_website %>%
  html_nodes("th~ td+ td") %>%
  html_text()
  #Create month and year lists
month <- (c("01", "05", "09", "02", "06", "10",
            "03", "07", "11", "04", "08", "12"))
year <- as.character(the_year)

  #Convert to dataframe
the_df <- data_frame(month, rep(year), rep(water.system.name),
                     rep(pswid), rep(ownership), max.withdrawals.mgd)

  #Rename columns
names(the_df) <- c("Month", "Year", "WaterSystem", "PSWID",
                   "Ownership", "MaxWithdrawals")

  #Make flow data numeric
the_df$MaxWithdrawals <- as.numeric(the_df$MaxWithdrawals)

  #Add date column
the_df <- the_df %>%
  mutate(Date = my(paste(Month, "-", Year)))

  #Return the dataframe
  return(the_df)
}
```

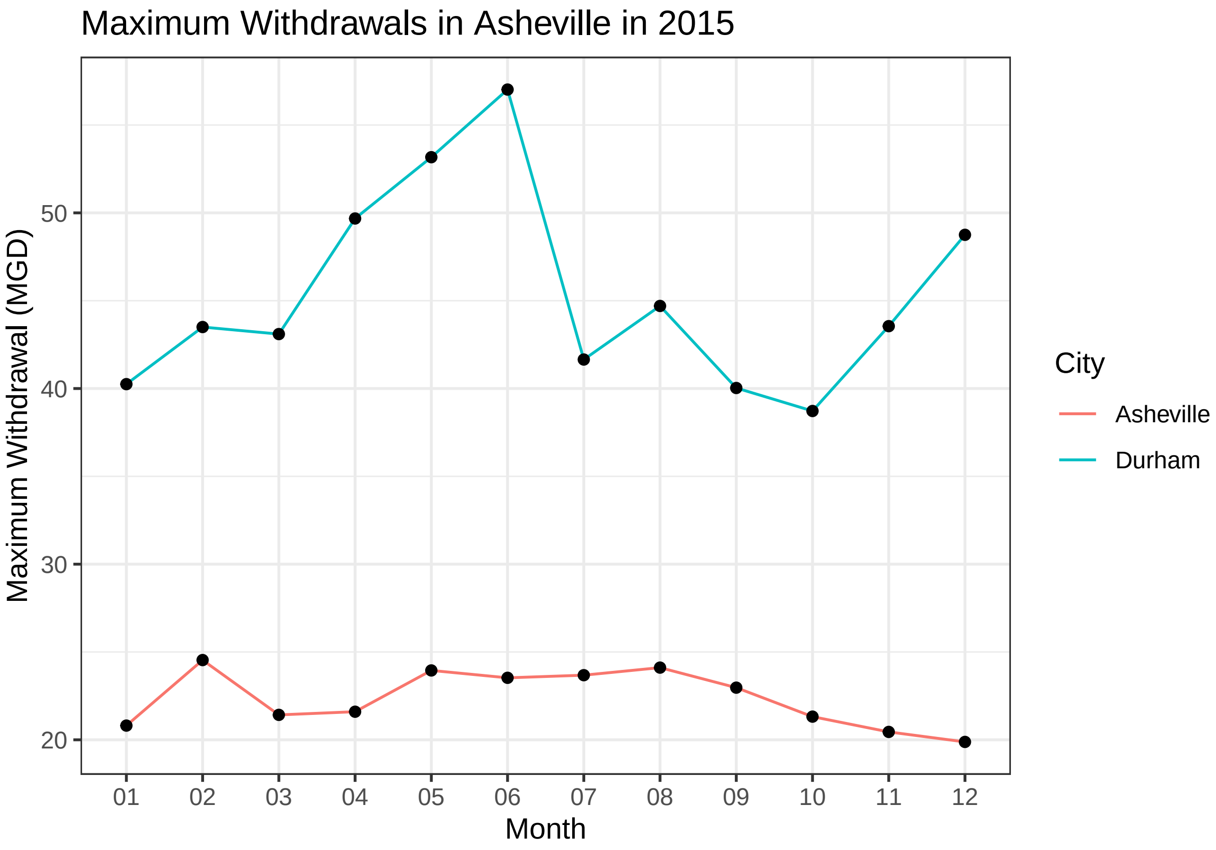
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7. Pull Durham data for 2015 and plot
durham_2015 <- scrape.it(the_pwsid = "03-32-010", the_year = 2015)

ggplot(durham_2015, aes(x = Month, y = MaxWithdrawals, group = 1)) +
  geom_line() +
  geom_point() +
  labs(y = "Maximum Withdrawal (MGD)", title = "Maximum Withdrawals in Durham in 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 the Asheville to Durham's water withdrawals.

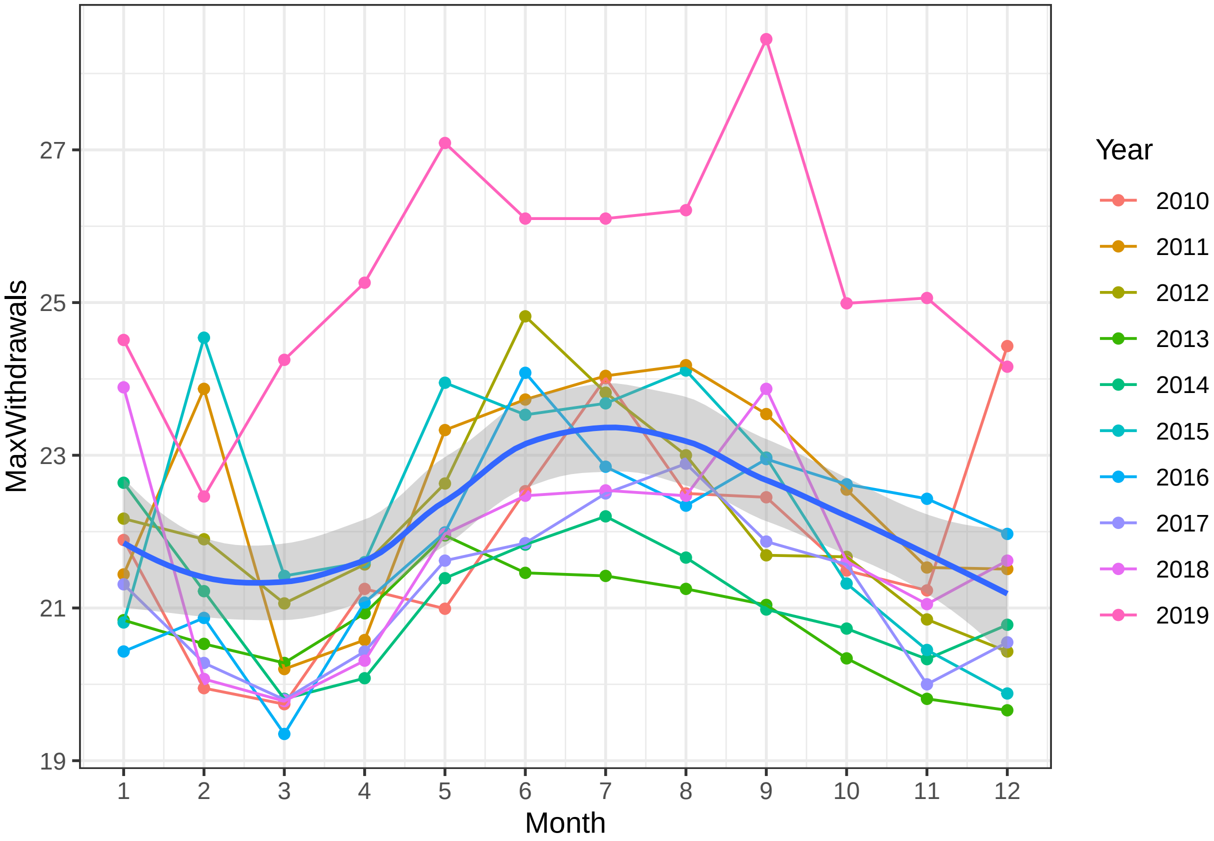
```
#8. Pull Asheville data for 2015 and plot
asheville_2015 <- scrape.it(the_pwsid = "01-11-010", the_year = 2015)

comparison <- rbind(durham_2015, asheville_2015)

ggplot(comparison, aes(x = Month, y = MaxWithdrawals, group = WaterSystem)) +
  geom_line(aes(color = WaterSystem)) +
  geom_point() +
  labs(y = "Maximum Withdrawal (MGD)",
       title = "Maximum Withdrawals in Asheville in 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 2019.Add a smoothed line to the plot.

```
#9. Pull a decade of Asheville data and graph

# Pull 2010
asheville_2010s <- scrape.it(the_pwsid = "01-11-010", the_year=2010)

# Create a vector of the remaining years
the_years <- c(2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)

# Scrape and add each year's data to the df
for (i in 1:length(the_years)){
  df <- scrape.it(the_pwsid = "01-11-010", the_year=the_years[i])
  asheville_2010s <- rbind(asheville_2010s, df)
}

# Re-format data to allow plot to work
asheville_2010s$Year <- as.factor(asheville_2010s$Year)
asheville_2010s$Month <- month(asheville_2010s$Date)

# Plot results
ggplot(asheville_2010s, aes(x = Month, y = MaxWithdrawals)) +
  geom_point(aes(color=Year, group=Year)) +
  geom_line(aes(color=Year, group=Year)) +
  geom_smooth() +
  scale_x_continuous(breaks = c(1:12))
```

```
## `geom_smooth()` using method = 'loess' and 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?

It looks like 2019 was an unusually high water usage year, so that might pull the data in the direction of showing a trend, but otherwise no.