Assignment 10: Data Scraping

Xianhang Xie

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.
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse)
library(rvest)
library(lubridate)
library(ggplot2)
#2.
getwd()
```

```
## [1] "C:/Users/11764/Desktop/EDA-Spring2023/Assignments"
```

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
url <- "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022"
webpage <- read_html(url)
```

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 "36.1000.

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

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

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

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

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

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.

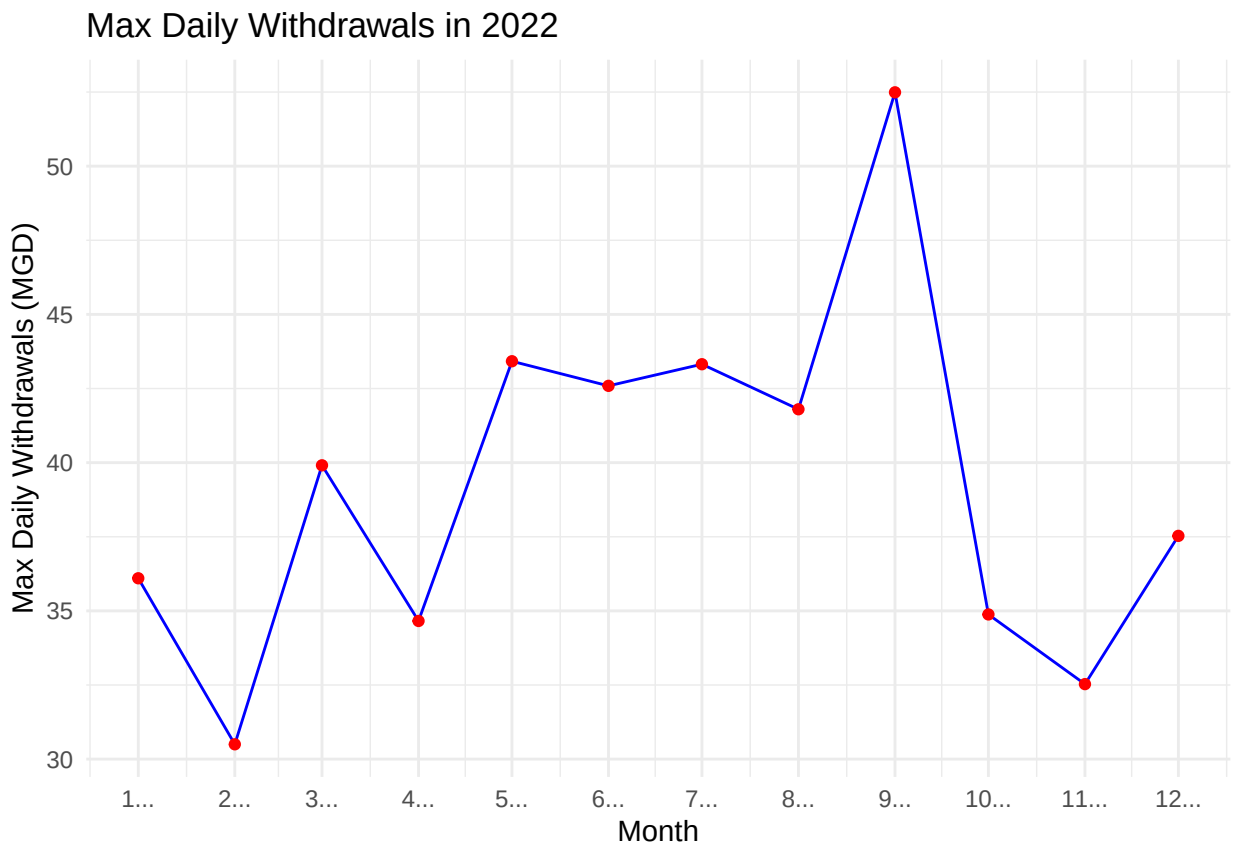
NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc… Or, you could scrape month values from the web page…

5. Create a line plot of the max daily withdrawals across the months for 2022

```
#4
max_withdrawals_mgd <- as.numeric(max.withdrawals.mgd)
year <- rep(2022, length(Month))
date <- my(paste(Month, year, sep = " "))
# Create a dataframe with the scraped data and additional columns
water_data <- data.frame(
  WaterSystemName = rep(water.system.name, length(Month)),
  PWSID = rep(PWSID, length(Month)),
  Ownership = rep(ownership, length(Month)),
  Month = Month,
  Year = year,
  Date = date,
  MaxWithdrawalsMGD = max_withdrawals_mgd
)

head(water_data)
```

```
##   WaterSystemName     PWSID    Ownership Month Year       Date
## 1          Durham 03-32-010 Municipality   Jan 2022 2022-01-01
## 2          Durham 03-32-010 Municipality   May 2022 2022-05-01
## 3          Durham 03-32-010 Municipality   Sep 2022 2022-09-01
## 4          Durham 03-32-010 Municipality   Feb 2022 2022-02-01
## 5          Durham 03-32-010 Municipality   Jun 2022 2022-06-01
## 6          Durham 03-32-010 Municipality   Oct 2022 2022-10-01
##   MaxWithdrawalsMGD
## 1             36.10
## 2             43.42
## 3             52.49
## 4             30.50
## 5             42.59
## 6             34.88
```

```
#5
ggplot(water_data, aes(x = Date, y = MaxWithdrawalsMGD)) +
  geom_line(group = 1, color = "blue") +
  geom_point(color = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  labs(title = "Max Daily Withdrawals in 2022",
       x = "Month",
       y = "Max Daily Withdrawals (MGD)") +
  theme_minimal()
```



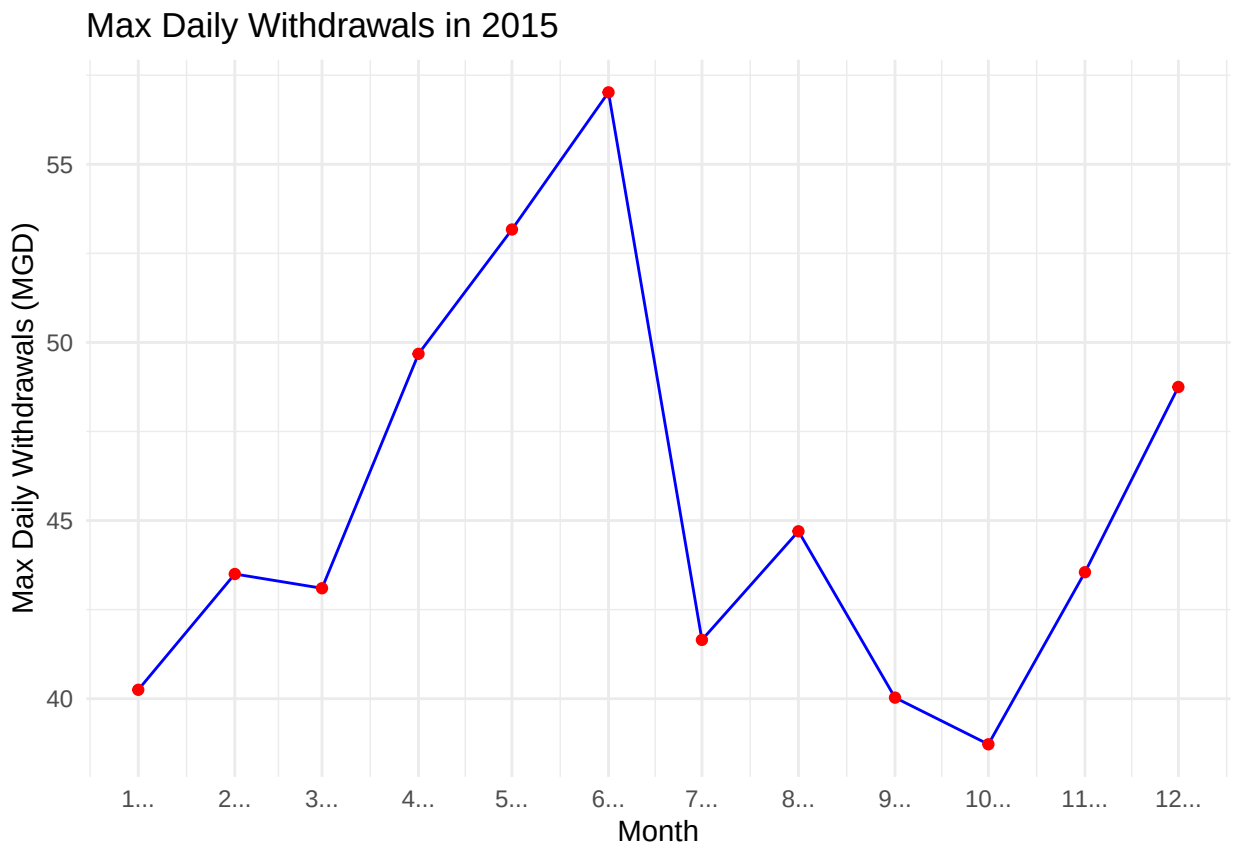
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
#Create our scraping function
scrape.it <- function(pwsid, the_year){

  #Retrieve the website contents
  the_website <- read_html(sprintf("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=%s&year=%s",
)

  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.withdrawals.mgd <- the_website %>%
```

```
  html_nodes("th~ td+ td") %>%
  html_text()

  #Convert to a dataframe
  max_withdrawals_mgd <- as.numeric(max.withdrawals.mgd)
year <- rep(the_year, length(Month))
date <- my(paste(Month, year, sep = " "))
# Create a dataframe with the scraped data and additional columns
water_data <- data.frame(
  WaterSystemName = rep(water.system.name, length(Month)),
  PWSID = rep(PWSID, length(Month)),
  Ownership = rep(ownership, length(Month)),
  Month = Month,
  Year = year,
  Date = date,
  MaxWithdrawalsMGD = max_withdrawals_mgd
)

  #Pause for a moment - scraping etiquette
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

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

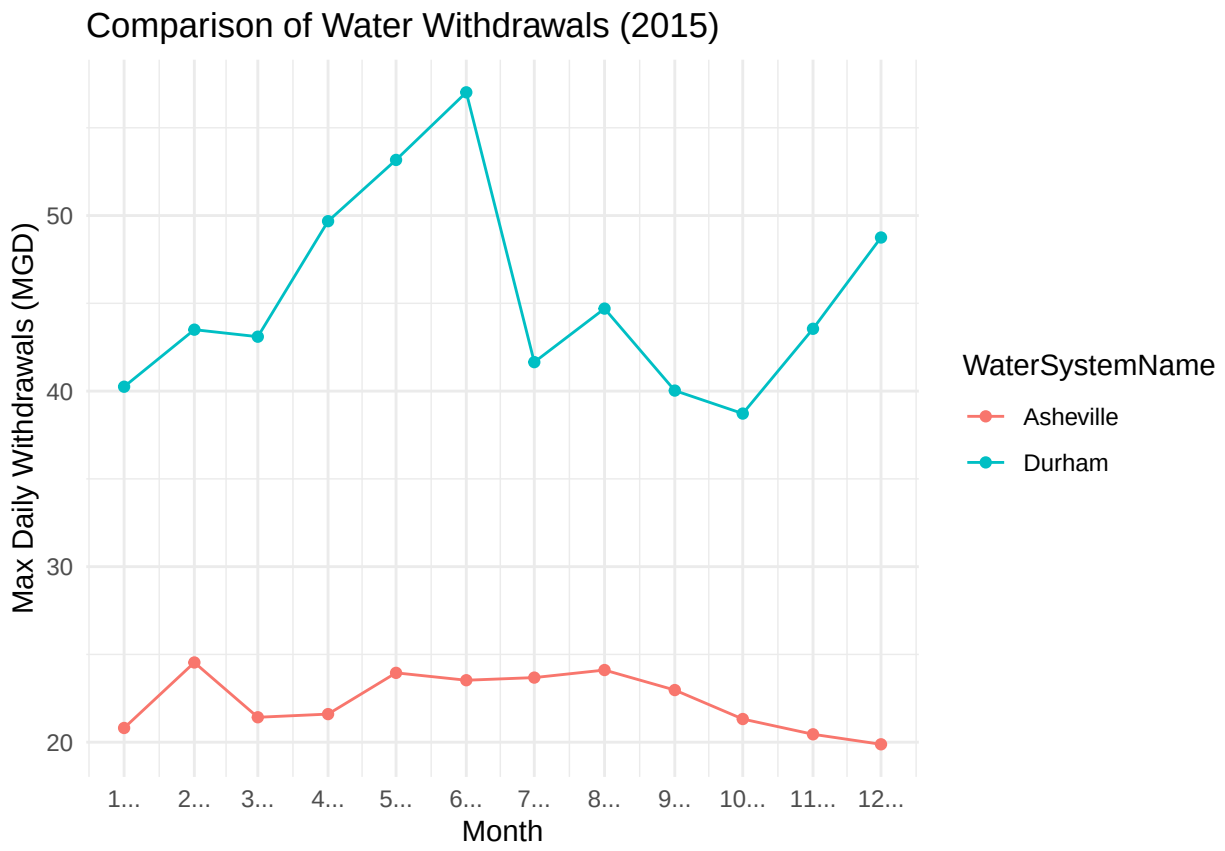
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
the_df <- scrape.it('03-32-010','2015')
ggplot(the_df, aes(x = Date, y = MaxWithdrawalsMGD)) +
  geom_line(group = 1, color = "blue") +
  geom_point(color = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  labs(title = "Max Daily Withdrawals in 2015",
       x = "Month",
       y = "Max Daily Withdrawals (MGD)") +
  theme_minimal()
```



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.

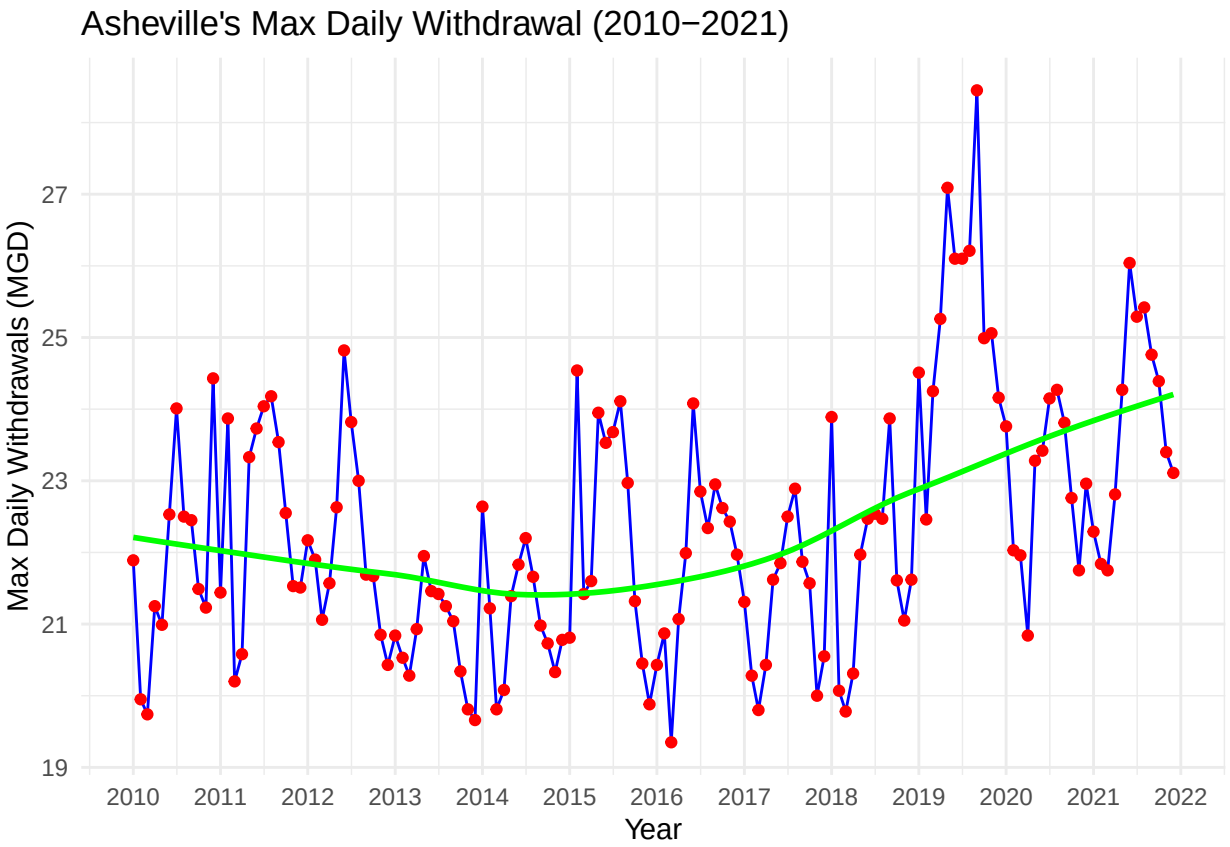
```
#8
the_df <- scrape.it('03-32-010','2015')
the_df2 <- scrape.it('01-11-010','2015')
# Combine the data
combined_data <- rbind(the_df, the_df2)
ggplot(combined_data, aes(x = Date, y = MaxWithdrawalsMGD, color = WaterSystemName)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  labs(title = "Comparison of Water Withdrawals (2015)",
       x = "Month",
       y = "Max Daily Withdrawals (MGD)") +
  theme_minimal()
```



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 <- 2010:2021
theids = rep('01-11-010', length(the_years))
dfs_2020 <- map2(theids,the_years,scrape.it)

#Conflate the returned list of dataframes into a single one
df_2020 <- bind_rows(dfs_2020)
ggplot(df_2020, aes(x = Date, y = MaxWithdrawalsMGD)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  geom_smooth(method = "loess", se = FALSE, color = "green") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(title = "Asheville's Max Daily Withdrawal (2010-2021)",
       x = "Year",
       y = "Max Daily Withdrawals (MGD)") +
  theme_minimal()
```



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

It seems that there is a increasing trend in water usage by the loess smoothing.