Assignment 09: Data Scraping

Kelly Davidson

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A09_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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

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. Setting up my session
getwd() #checking working directory
```

```
## [1] "/home/guest/EDA-Fall2022/EDA-Fall2022"
```

```
library(tidyverse) #loading necessary packages
library(lubridate)
library(rvest)
library(cowplot)

A10_theme <- #defining and setting my ggplot theme
  theme_light(base_size = 10) +
  theme(axis.text = element_text(color = "dark gray"),
  legend.position = "right")
theme_set(A10_theme)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2021 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=2021

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

```
#2. Indicating the NC DEQs Local Water Supply Planning URL to be scraped
Durham_LWSP <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2021")
```

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:
- Maximum Daily 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. Scraping the data (Water System Name, PSWID, Ownership, and Maximum Daily Use) from the URL
water.system.name <- Durham_LWSP %>%  #assigning water system name
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

pswid <- Durham_LWSP %>%   #assigning PWSID
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

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

max.withdrawals.mgd <-  Durham_LWSP %>%  #assigning maximum daily use
  html_nodes("th~ td+ td") %>%
  html_text()
```

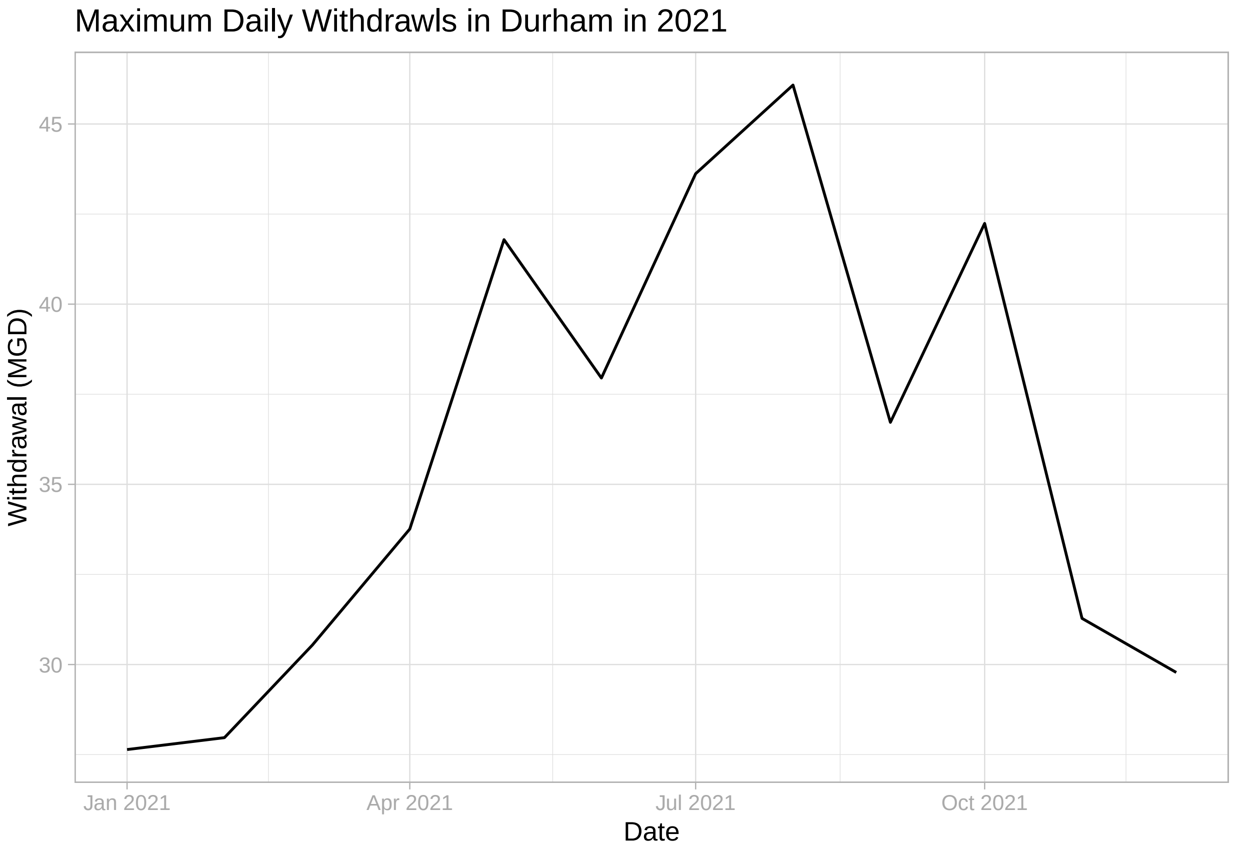
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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…

5. Create a line plot of the maximum daily withdrawals across the months for 2021

```
#4. Converting the scraped data into a dataframe
Durham_LWSP_df <- data.frame("Month" = c("Jan", "May", "Sep",
                                         "Feb", "Jun", "Oct",
                                         "Mar", "Jul", "Nov",
                                         "Apr", "Aug", "Dec"),
                             #creating a month column
                             "Year" = rep(2021), #creaing a year column
                             "Max.withdrawals.mgd" = as.numeric(max.withdrawals.mgd))
#creating the maximum daily withdrawal column

Durham_LWSP_df2 <- Durham_LWSP_df %>%  #wrangling the dataframe
  mutate(Water.system.name = !!water.system.name, #assigning water system name column
         PSWID = !!pswid,  #assigning PSWID column
         Ownership =!!ownership, #assigning ownership column
         Date = my(paste(Month,"-",Year))) %>%  #creating a date column
  arrange(ymd(Date))

#5. Graphing a line plot of the maximum daily withdrawals for 2021
Max_daily_withdrawal_plot <- ggplot(Durham_LWSP_df2, aes(x = Date, y = Max.withdrawals.mgd)) +
  geom_line() + #specifying line plot
  labs(title = "Maximum Daily Withdrawls in Durham in 2021", #assigning a title
  y = "Withdrawal (MGD)") #renaming the y-axis
print(Max_daily_withdrawal_plot)
```

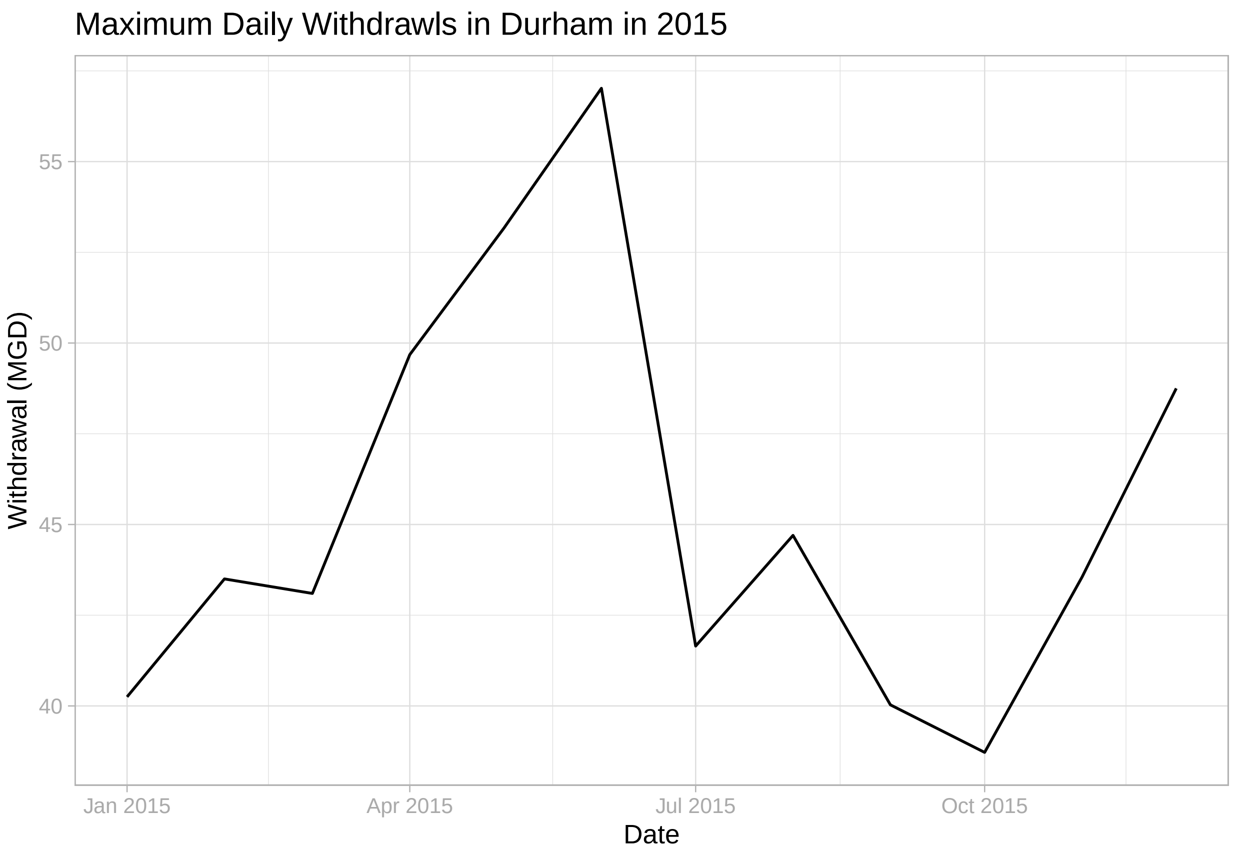


6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6. Constructing a scrape function
scrape.it <- function(Year, PSWID){
  the_website <- read_html(paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=",
                                  PSWID, "&year=", Year)) #retrieving the website contents
  #setting the element address variables
  water_system_tag <- "div+ table tr:nth-child(1) td:nth-child(2)"
  pswid_tag <- "td tr:nth-child(1) td:nth-child(5)"
  ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
  max_withdrawals_tag <- "th~ td+ td"

  #scraping the data items
  water_system <- the_website %>% html_nodes(water_system_tag) %>% html_text()
  pswid <- the_website %>%  html_nodes(pswid_tag) %>%  html_text()
  ownership <- the_website %>%  html_nodes(ownership_tag) %>%  html_text()
  max_withdrawals <- the_website %>%  html_nodes(max_withdrawals_tag) %>%  html_text()

  #converting to a dataframe
  water_df <- data.frame("Month" = c("Jan", "May", "Sep",
                                     "Feb", "Jun", "Oct",
                                     "Mar", "Jul", "Nov",
                                     "Apr", "Aug", "Dec"),
                             "Year" = rep(Year, 12),
                             "Max_withdrawals_mgd" = as.numeric(max_withdrawals)) %>%
    mutate(Water_System = !!water_system,
         PSWID = !!pswid,
         Ownership =!!ownership,
         Date = my(paste(Month,"-",Year))) %>%
    arrange(ymd(Date))

  return(water_df) #return the dataframe
}
```

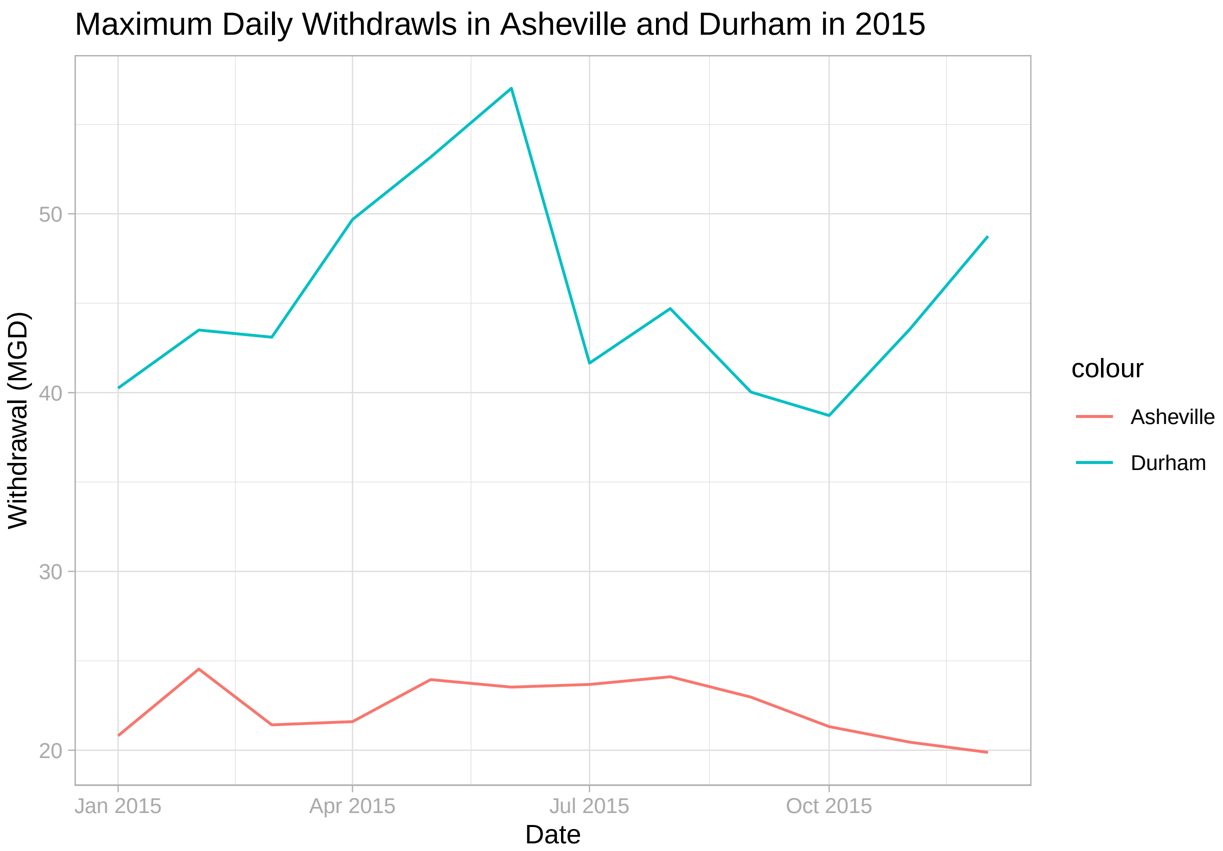
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7. Extracting maximum daily withdrawals for Durham in 2015 using the scrape function
Durham_withdrawals_2015 <- scrape.it(2015, "03-32-010") #utilizing the scrape function
view(Durham_withdrawals_2015)

#plotting maximum daily withdrawals for Durham in 2015
Durham_withdrawals_2015_plot <-
  ggplot(Durham_withdrawals_2015, aes(x = Date, y = Max_withdrawals_mgd)) +
  geom_line() + #specifying line plot
  labs(title = "Maximum Daily Withdrawls in Durham in 2015", #assigning a title
  y = "Withdrawal (MGD)") #renaming the y-axis
print(Durham_withdrawals_2015_plot)
```



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. Extracting maximum daily withdrawals for Asheville in 2015 using the scrape function
Asheville_withdrawals_2015 <- scrape.it(2015, "01-11-010") #utilizing the scrape function
view(Asheville_withdrawals_2015)

#plotting maximum daily withdrawals for Durham and Asheville in 2015
Ash_Durh_2015_plot <- ggplot() +
  geom_line(data = Asheville_withdrawals_2015,
            aes(x = Date, y = Max_withdrawals_mgd, color = "Asheville")) +  #specifying line plot
  geom_line(data = Durham_withdrawals_2015,
            aes(x = Date, y = Max_withdrawals_mgd, color = "Durham")) + #specifying line plot
  labs(title = "Maximum Daily Withdrawls in Asheville and Durham in 2015", #assigning a title
  y = "Withdrawal (MGD)") #renaming the y-axis
print(Ash_Durh_2015_plot)
```

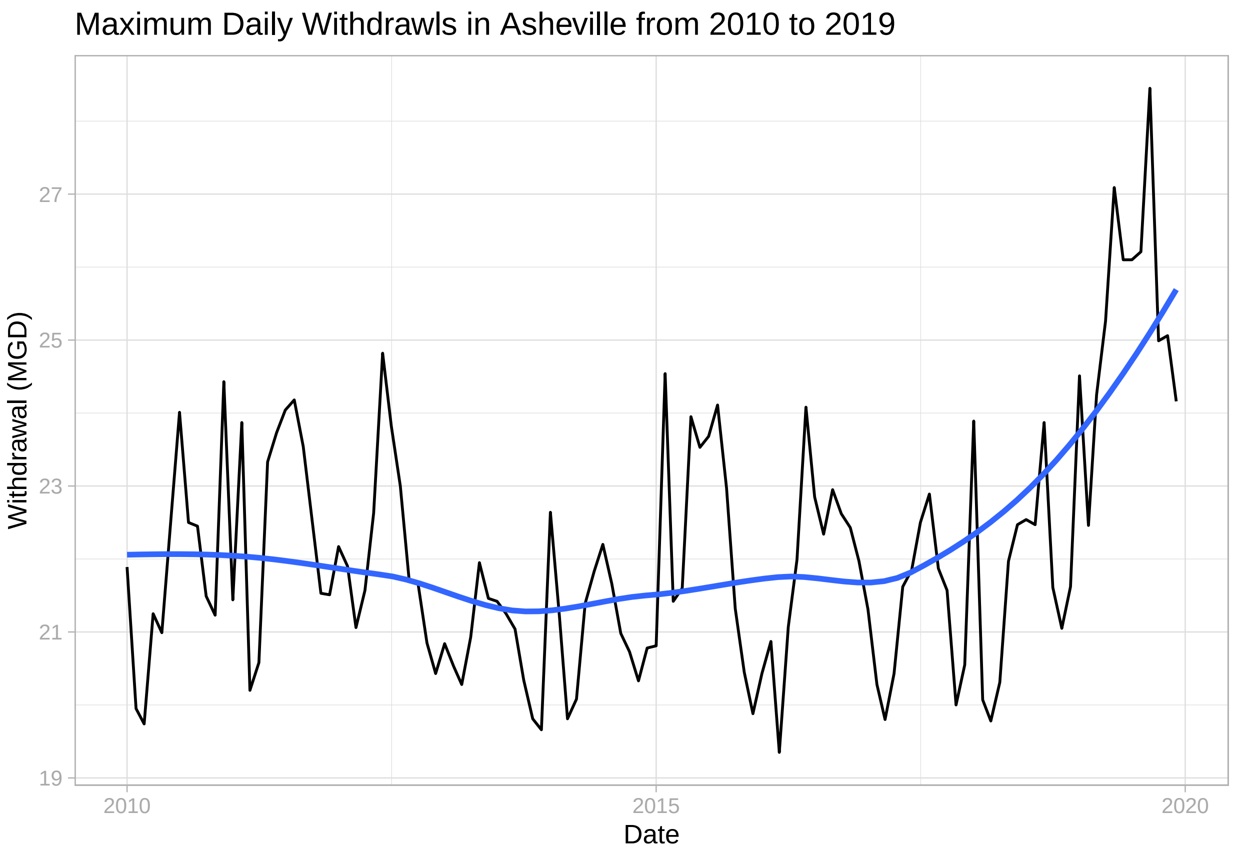


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

   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. Extracting maximum daily withdrawals for Asheville from 2010-2019 using the scrape function
Asheville_years <- rep(2010:2019) #creating a list of the years 2010-2019
Asheville_PSWID <- rep.int("01-11-010", length(Asheville_years)) #creating a list of the PSWID

#using map2 and scrape function to retrieve data from 2010-2019
Asheville_2010_2019 <- map2(Asheville_years, Asheville_PSWID, scrape.it)

#joining the returned list of dataframes into a single one
Asheville_complete <- bind_rows(Asheville_2010_2019)

#plotting maximum daily withdrawals for Asheville from 2010-2019
Asheville_complete_plot <- ggplot(Asheville_complete, aes(x = Date, y = Max_withdrawals_mgd))  +
  geom_line()  + #specifying line plot
  geom_smooth(method="loess",se=FALSE) + #adding a smoothed trendline
  labs(title = "Maximum Daily Withdrawls in Asheville from 2010 to 2019", #assigning a title
  y = "Withdrawal (MGD)") #renaming the y-axis
print(Asheville_complete_plot)
```

```
## `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? Yes, it appears there is an increasing trend of maximum daily withdrawls in Asheville from 2010 to 2019.