Assignment 09: Data Scraping

Emma Brentjens

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
##checking working directory
getwd()
```

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

```
##loading packages
library(tidyverse)
library(rvest)
library(ggplot2)
library(dplyr)
library(lubridate)

##setting ggplot theme
Emma_theme <- theme_linedraw() +
  theme(axis.text = element_text(color = "black", size = 10),
        legend.position = "right")

theme_set(Emma_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
##reading URL
local_water <- 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 data
water.system.name <- local_water %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

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

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

max.withdrawals.mgd <- local_water %>%
  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.

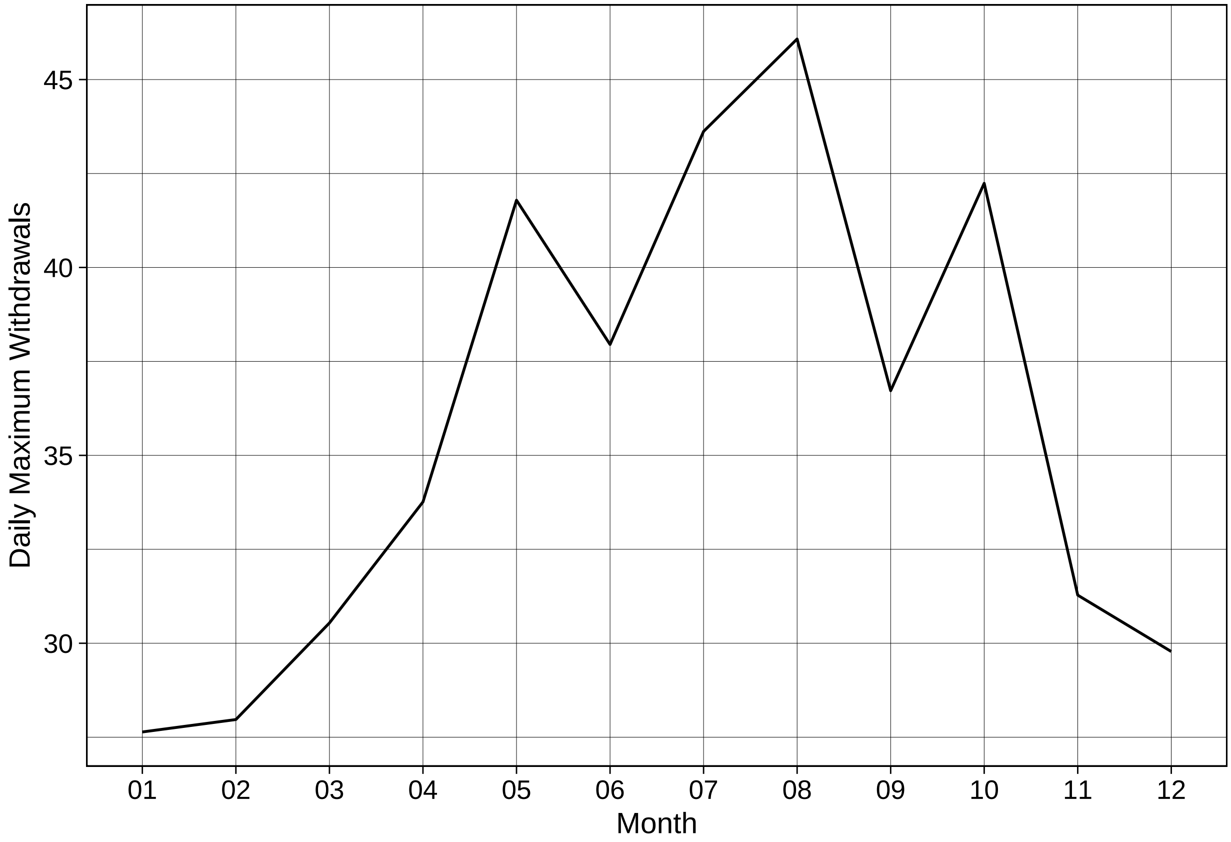
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc…

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

```
#4
##creating dataframe
local_water_df <- data_frame(Water_System_Name=water.system.name,
                             PWSID=pwsid,
                             Ownership=ownership,
                             Daily_Max_Withdrawals=
                               as.numeric(max.withdrawals.mgd),
                             Month=c("01", "05", "09", "02", "06", "10", "03",
                                     "07", "11", "04", "08", "12"),
                             Year="2021",
                             Date=my(paste(Month,"/", Year)))
```

```
## 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.
```

```
#5
##max daily withdrawals plot
withdrawal_plot <- ggplot(data=local_water_df, aes(x=Month, y=Daily_Max_Withdrawals, group=1)) +
  geom_line() +
  ylab("Daily Maximum Withdrawals")

withdrawal_plot
```



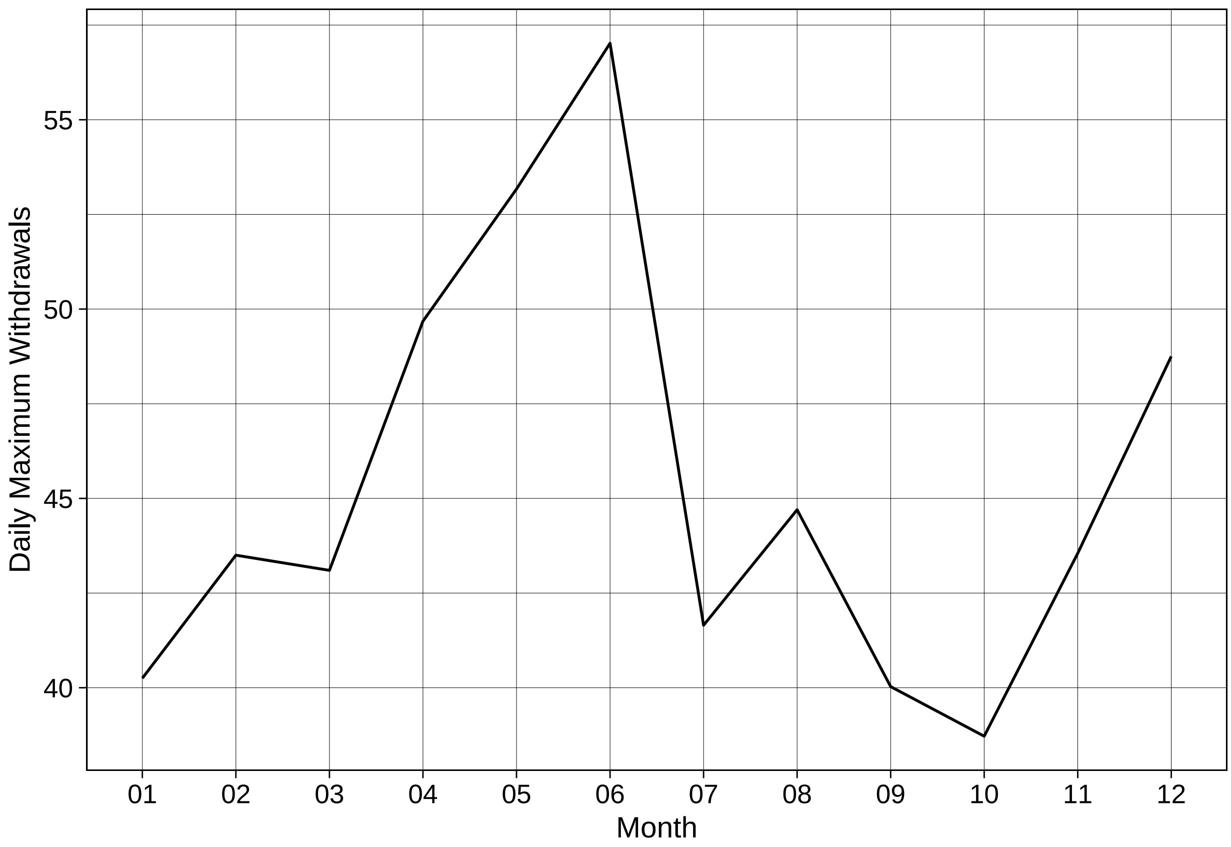
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ

has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped.**

```
#6
##creating function
data_scrape <- function(PWSID, Year){
  url <- read_html(paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=",
                          PWSID, "&year=", Year))

water.system.name_tag <- "div+ table tr:nth-child(1) td:nth-child(2)"
pwsid_tag <- "td tr:nth-child(1) td:nth-child(5)"
ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
max.withdrawals.mgd_tag <- "th~ td+ td"

water.system.name <- url %>%
  html_nodes(water.system.name_tag) %>%
  html_text()
pwsid <- url %>%
  html_nodes(pwsid_tag) %>%
  html_text()
ownership <- url %>%
  html_nodes(ownership_tag) %>%
  html_text()
max.withdrawals.mgd <- url %>%
  html_nodes(max.withdrawals.mgd_tag) %>%
  html_text()

  local_water_df2 <- data_frame(Month=c("01", "05", "09", "02", "06", "10", "03",
                                        "07", "11", "04", "08", "12"),
                                Year=Year,
                                Date=my(paste(Month, "-", Year)),
                                Daily_Max_Withdrawals=
                                     as.numeric(max.withdrawals.mgd)) %>%
    mutate(PWSID= !!pwsid,
           Ownership= !!ownership,
           Water_System_Name= !!water.system.name)

  return(local_water_df2)
}
```

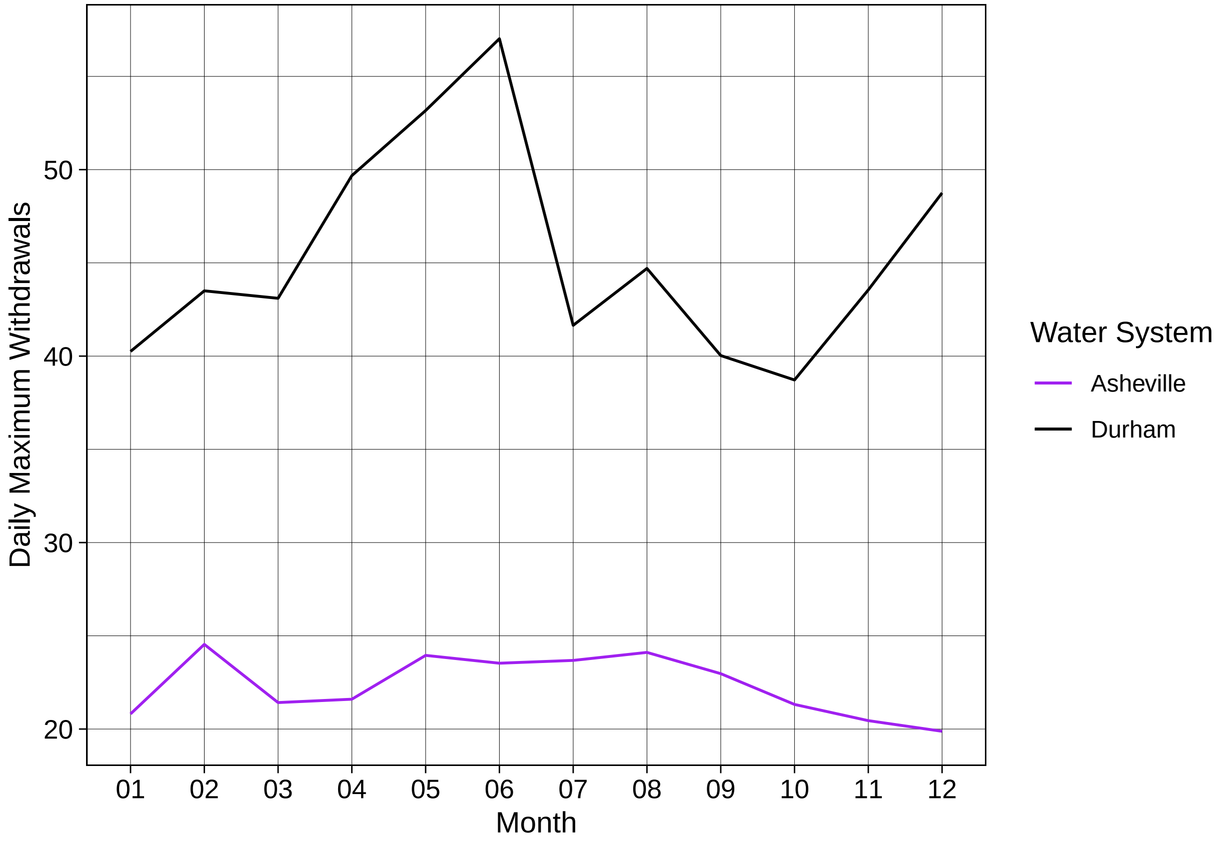
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7
##scraping data from Durham in 2015
Durham_2015 <- data_scrape(PWSID="03-32-010", Year=2015)

##Durham 2015 plot
withdrawal_plot_2015 <- ggplot(data=Durham_2015, aes(x=Month, y=Daily_Max_Withdrawals, group=1)) +
  geom_line() +
  ylab("Daily Maximum Withdrawals")

withdrawal_plot_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
##scraping data from Asheville in 2015
Asheville_2015 <- data_scrape(PWSID="01-11-010", Year=2015)

##Durham and Asheville plot
Durham_Asheville_plot <- ggplot() +
  geom_line(data=Durham_2015, aes(x=Month, y=Daily_Max_Withdrawals,
                                  group=1, color=Water_System_Name)) +
  geom_line(data=Asheville_2015, aes(x=Month, y=Daily_Max_Withdrawals,
                                     color=Water_System_Name, group=1)) +
  scale_color_manual(values=c("purple", "black"), name = "Water System") +
  ylab("Daily Maximum Withdrawals")

Durham_Asheville_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.

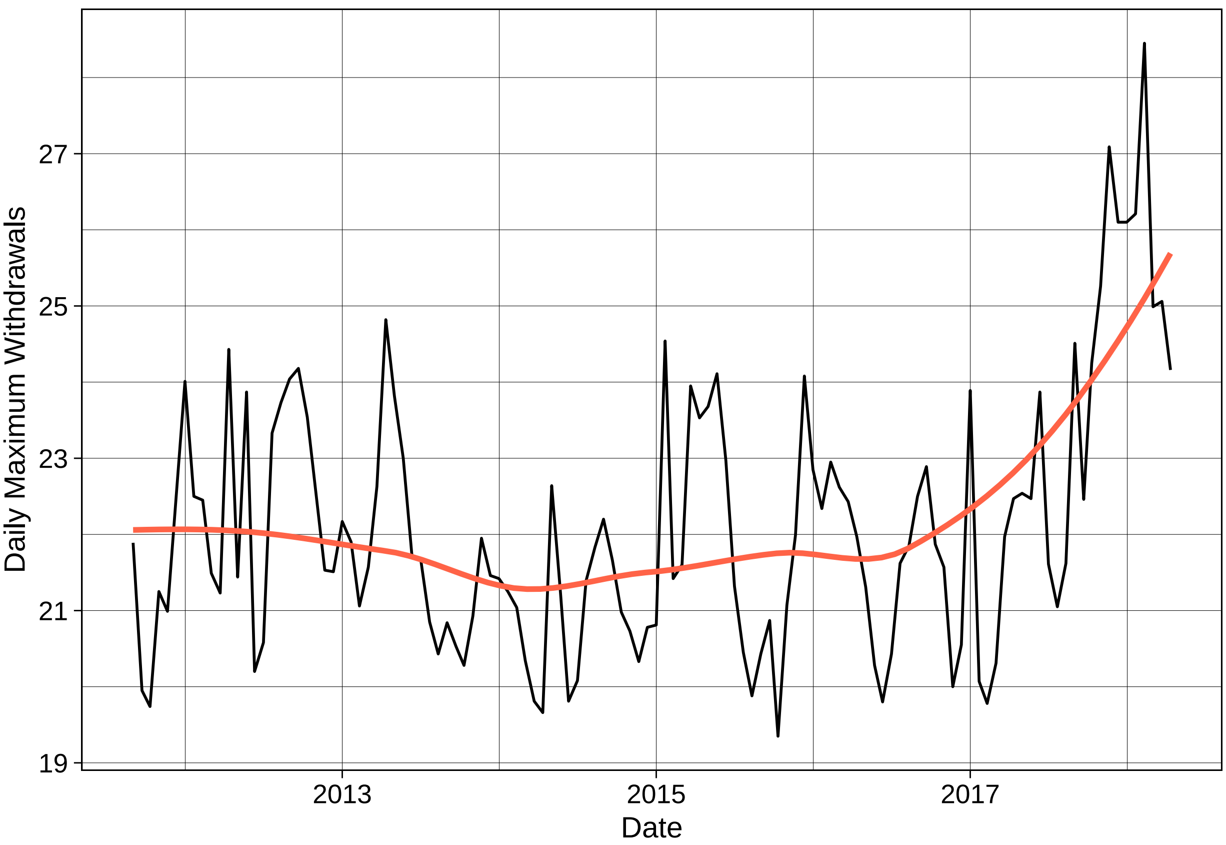
   TIP: See Section 3.2 in the "09_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```
#9
##scraping Asheville data for years 2010-2019
Asheville_data <- map2("01-11-010", rep(2010:2019), data_scrape)
Asheville_data_df <- bind_rows(Asheville_data)
class(Asheville_data_df$Date)
```

```
## [1] "Date"
```

```
##creating Ashevile water withdrawal plot
Asheville_plot <- ggplot(data=Asheville_data_df, aes(x=Date, y=Daily_Max_Withdrawals, group=1)) +
  geom_line() +
  geom_smooth(color="tomato", se=F) +
  ylab("Daily Maximum Withdrawals") +
  scale_x_date(breaks="3 years", labels = c("2011", "2013", "2015", "2017", "2019"))

Asheville_plot
```

```
## `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?
There seems to be a trend of increasing water usage over time.