Assignment 10: Data Scraping

Chrissie Pantoja

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(tidyverse)
library(rvest)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2023 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=2023

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

```
#2

#Set the URL
```

```
theURL <-
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023'

#Read the contents of the webpage
webpage <- read_html(theURL)
```

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)".

```
#3

water_system_name <- webpage %>% html_nodes('table:nth-child(7) tr:nth-child(1) td:nth-child(2)') %>% h

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

ownership <- webpage %>% html_nodes('table:nth-child(7) tr:nth-child(2) td:nth-child(4)') %>% html_text

water_supply_sources <- webpage %>% html_nodes(':nth-child(31) td:nth-child(9) , tr:nth-child(2) :nth-c

class(water_supply_sources)
```

```
## [1] "character"
```

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.

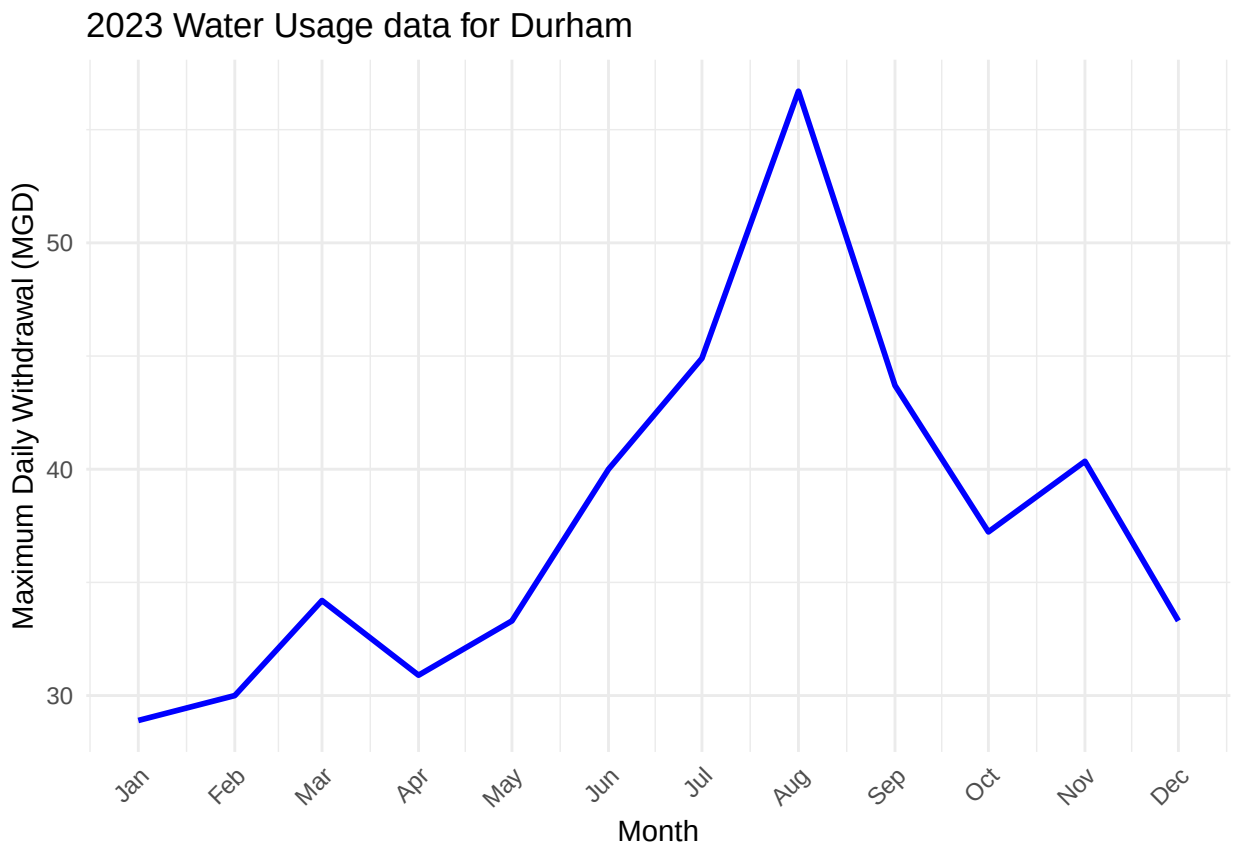
> NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc... Or, you could scrape month values from the web page...

```r
# Manually create the month vector (order the months as needed)
months <- c("Jan", "May", "Sept", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

# Create a dataframe with the scraped data and the month column
df_withdrawals <- data.frame(
  Month = months,
  water_system_name = rep(water_system_name, length(months)),
  PWSID = rep(PWSID, length(months)),
  ownership = rep(ownership, length(months)),
  water_supply_sources = as.numeric(water_supply_sources),
  Year = as.numeric(2023)) %>%
mutate(
  date = my(paste(Month,"-",Year)))
```

5. Create a line plot of the maximum daily withdrawals across the months for 2023, making sure, the months are presented in proper sequence.

```r
ggplot(df_withdrawals, aes(x= date,y = water_supply_sources, group = 1)) +
  geom_line(group = 1, color = "blue", size = 1) +
  labs(title = paste("2023 Water Usage data for Durham"),
       y = "Maximum Daily Withdrawal (MGD)",
       x = "Month") +
       scale_x_date(date_breaks = "1 month", date_labels = "%b") +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readability
```



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, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.

# Define the scraping function
scrape.it <- function(pwsid, year) {

  # Construct the URL using the provided pwsid and year
  the_url <- paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', pwsid, '&year=', year)

  # Fetch the website content
  the_website <- read_html(the_url)

  # Scrape the relevant data
  water_system_name <- the_website %>% html_nodes('table:nth-child(7) tr:nth-child(1) td:nth-child(2)')
  PWSID <- the_website %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
  ownership <- the_website %>% html_nodes('table:nth-child(7) tr:nth-child(2) td:nth-child(4)') %>% html
  water_supply_sources <- the_website %>% html_nodes('th~ td+ td') %>% html_text()


  # Manually create the month vector (order the months as needed)
  months <- c("Jan", "May", "Sept", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

  # Create a dataframe with the scraped data and the month column
  df <- data.frame(
    Month = months,
    water_system_name = rep(water_system_name, length(months)),
    PWSID = rep(PWSID, length(months)),
    ownership = rep(ownership, length(months)),
    water_supply_sources = as.numeric(water_supply_sources),
    Year = as.numeric(year)) %>%
  mutate(
    date = my(paste(Month,"-",Year)))


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

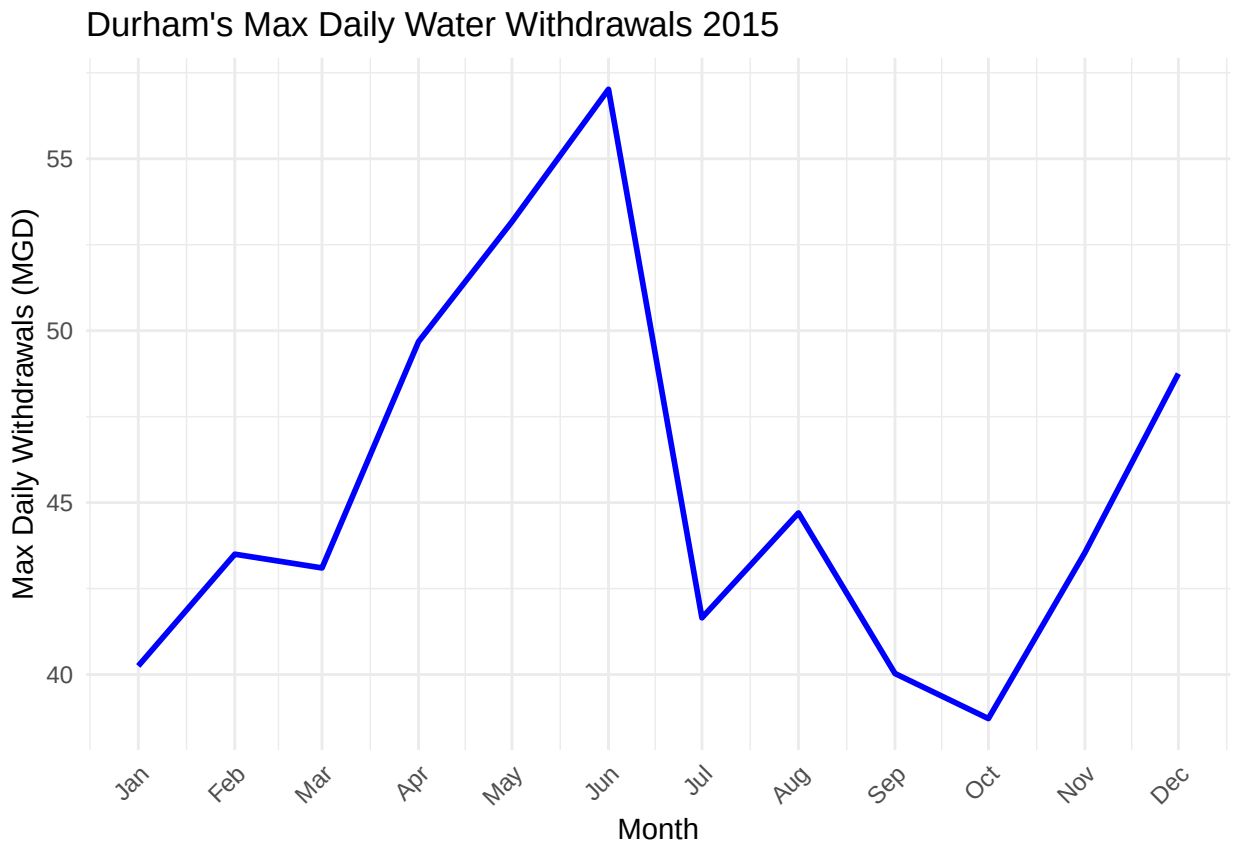
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7

# Extract data for Durham (PWSID = '03-32-010') and year = 2015
df_durham_2015 <- scrape.it("03-32-010",2015)
view(df_durham_2015)

# Plot max daily withdrawals for each month
ggplot(df_durham_2015, aes(x = date, y = water_supply_sources)) +
```

```
  geom_line(group = 1, color = "blue", size = 1) +
  labs(
    title = "Durham's Max Daily Water Withdrawals 2015",
    x = "Month",
    y = "Max Daily Withdrawals (MGD)"
  ) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



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.

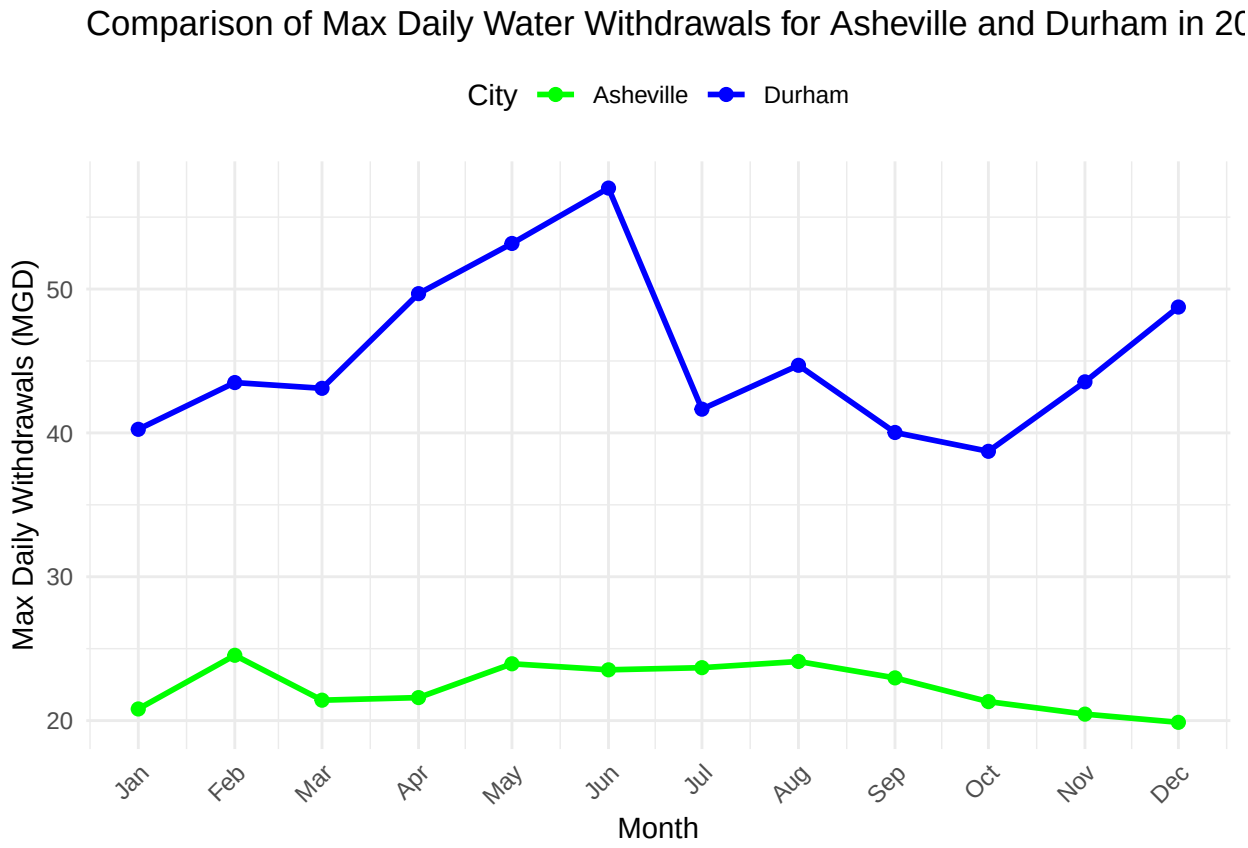
```
#8

df_asheville_2015 <- scrape.it("01-11-010", 2015)

# Combine the data for both cities
df_combined <- bind_rows(df_durham_2015, df_asheville_2015)

# Plot the comparison of water withdrawals for Asheville and Durham
ggplot(df_combined, aes(x = date, y = water_supply_sources, color = water_system_name, group = water_sys
```

```
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Comparison of Max Daily Water Withdrawals for Asheville and Durham in 2015",
    x = "Month",
    y = "Max Daily Withdrawals (MGD)",
    color = "City"
  ) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "top"
  ) +
  scale_color_manual(values = c("Durham" = "blue", "Asheville" = "green"))
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022.Add a smoothed line to the plot (method = 'loess').

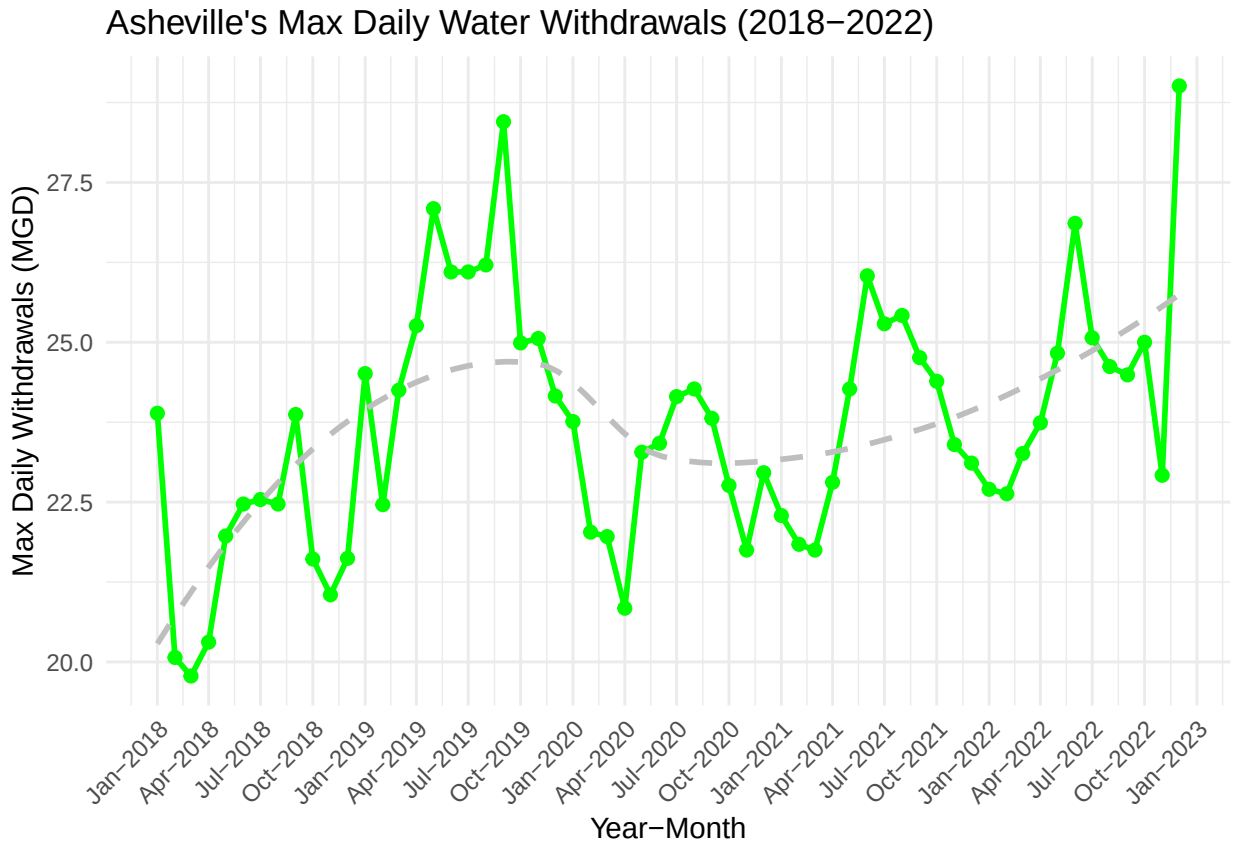
   TIP: See Section 3.2 in the "10_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```r
#9

# Define the years and location
years <- 2018:2022
pwsid <- "01-11-010"

# Fetch Asheville's data for 2018-2022
df_asheville_all <- bind_rows(map2(rep(pwsid, length(years)), years, scrape.it))

# Plot Asheville's max daily withdrawals with a smoothed line
ggplot(df_asheville_all, aes(x = date, y = water_supply_sources)) +
 geom_line(color = "green", size = 1) +
 geom_point(color = "green", size = 2) +
 geom_smooth(method = "loess", color = "grey", linetype = "dashed", se = FALSE) +
 scale_x_date(date_labels = "%b-%Y", date_breaks = "3 months") +
 labs(
 title = "Asheville's Max Daily Water Withdrawals (2018-2022)",
 x = "Year-Month",
 y = "Max Daily Withdrawals (MGD)"
 ) +
 theme_minimal() +
 theme(
 axis.text.x = element_text(angle = 45, hjust = 1),
 legend.position = "none"
 )
```



Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time? > Answer: > Based on the plot, Asheville's water usage shows a complex pattern over time. While there is a general upward trend from 2018 to 2020, the curve experiences a dip around 2020 before resuming its upward trajectory and increasing again from half year of 2020 to 2023. This suggests that water consumption in Asheville has not followed a simple linear increase but has fluctuated with periods of growth and decline.