Assignment 10: Data Scraping

Christopher Starr

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
#Install familiar packages
#install.packages("viridis")
#install.packages("dataRetrieval")
#install.packages("tidycensus")
library(tidyverse);library(lubridate);library(viridis);library(here)
here()
```

```
## [1] "/Users/christopherstarr/SpringENV2025"
```

```
#install.packages("tinytex")
#tinytex::install_tinytex()
library("tinytex")

#install.packages("dplyr")
library(dplyr)

#install.packages("rvest")
library(rvest)
```

```r
#install.packages('xml2')
library('xml2')

#install.packages("dataRetrieval")
library(dataRetrieval)

#install.packages("tidycensus")
library(tidycensus)

# Set theme
mytheme <- theme_gray() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2024 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=2024

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

```r
#2

# Specify the URL
url <- "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024"

# Read the HTML content
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)“.

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

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

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

# scrape Maximum Day Use for each month
max_day_use <- webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()
```

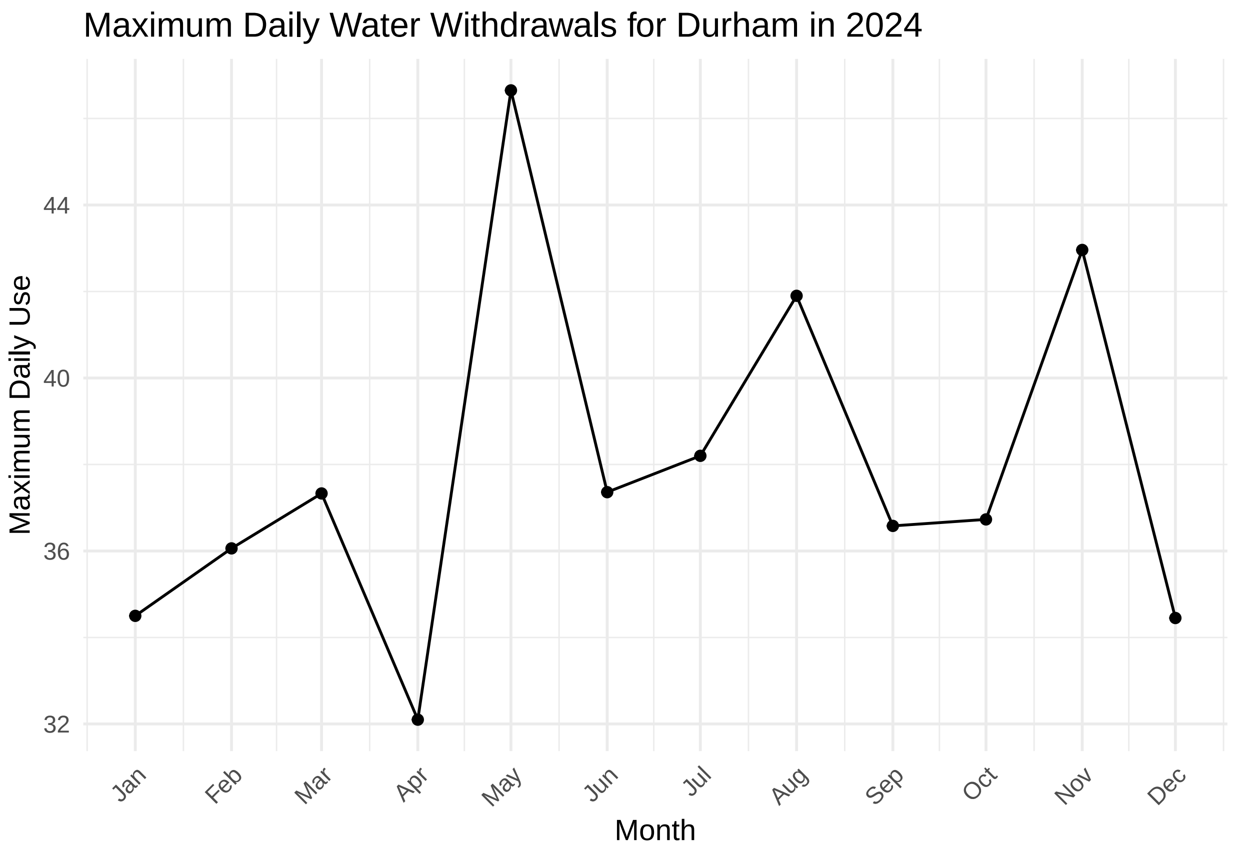
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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… Or, you could scrape month values from the web page…

5. Create a line plot of the maximum daily withdrawals across the months for 2024, making sure, the months are presented in proper sequence.

```r
#4
#Convert to dataframe
water_data <- data.frame(
  "Month" = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")[1:leng
  "Water_System_Name" = rep(water_system_name, length(max_day_use)),
  "PWSID" = rep(pwsid, length(max_day_use)),
  "Ownership" = rep(ownership, length(max_day_use)),
  "Max_Day_Use" = as.numeric(gsub('[^0-9.]', '', max_day_use))
) %>%
  mutate(Date = as.Date(paste("2024", match(Month, month.abb), "01", sep="-")),
         Year = 2024)

#5
ggplot(water_data, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = paste("Maximum Daily Water Withdrawals for", water_system_name, "in 2024"),
       x = "Month",
       y = "Maximum Daily Use") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



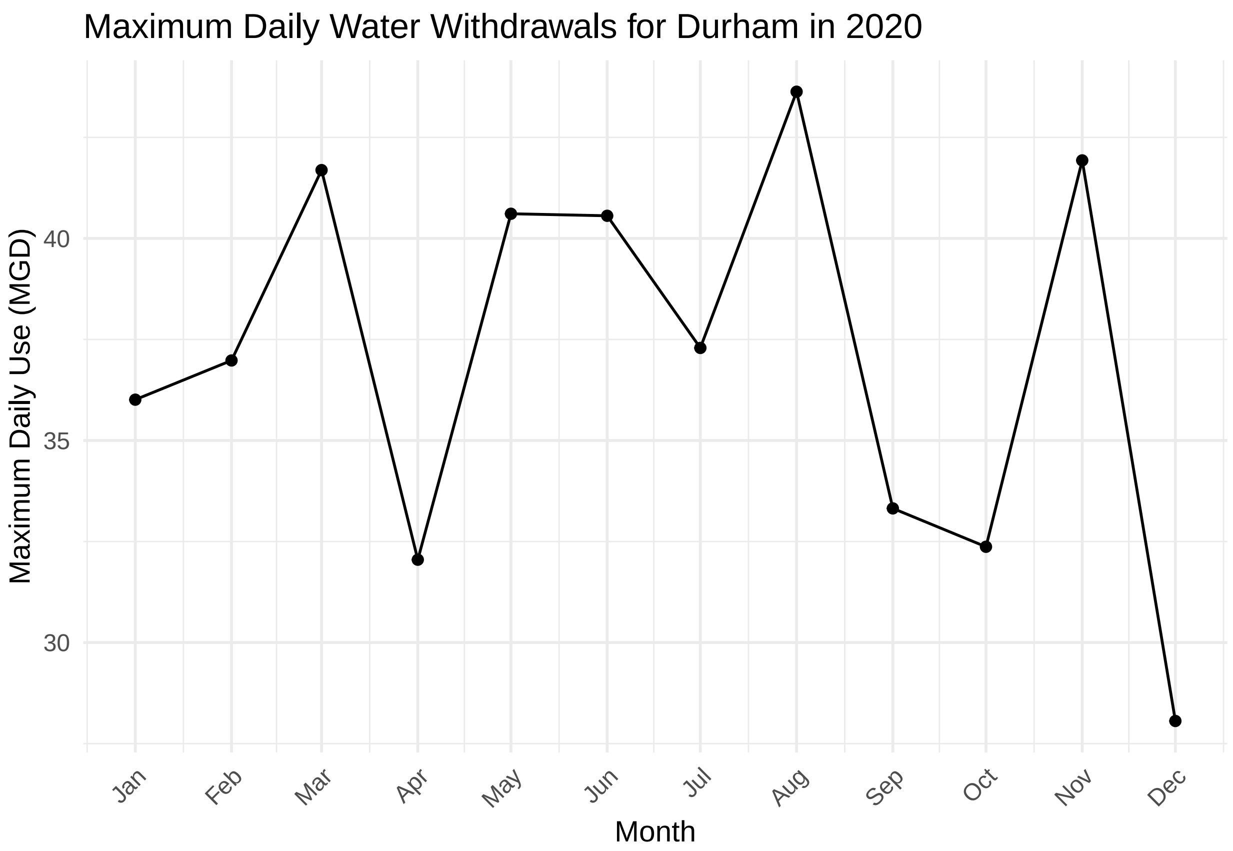
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function with two input - “PWSID” and “year” - that:

- Creates a URL pointing to the LWSP for that PWSID for the given year
- Creates a website object and scrapes the data from that object (just as you did above)
- Constructs a dataframe from the scraped data, mostly as you did above, but includes the PWSID and year provided as function inputs in the dataframe.
- Returns the dataframe as the function’s output

```
#6.
# Create function to scrape water data
scrape_water_data <- function(PWSID, year) {
  # Construct URL based on PWSID and year
  url <- paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=", PWSID, "&year=", year)

  # Read the HTML content
  webpage <- read_html(url)
  # Scrape water system name
  water_system_name <- webpage %>%
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
    html_text()

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

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

  # Scrape Maximum Day Use for each month
  max_day_use <- webpage %>%
    html_nodes("th~ td+ td") %>%
    html_text()

  # Convert to dataframe
  water_data <- data.frame(
    "Month" = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")[1:l
    "Water_System_Name" = rep(water_system_name, length(max_day_use)),
    "PWSID" = rep(PWSID, length(max_day_use)),  # Use the input PWSID
    "Ownership" = rep(ownership, length(max_day_use)),
    "Max_Day_Use" = as.numeric(gsub('[^0-9.]', '', max_day_use)),
    "Year" = rep(as.numeric(year), length(max_day_use))  # Use the input year
  ) %>%
    mutate(Date = as.Date(paste(Year, match(Month, month.abb), "01", sep="-")))

  return(water_data)
}
```

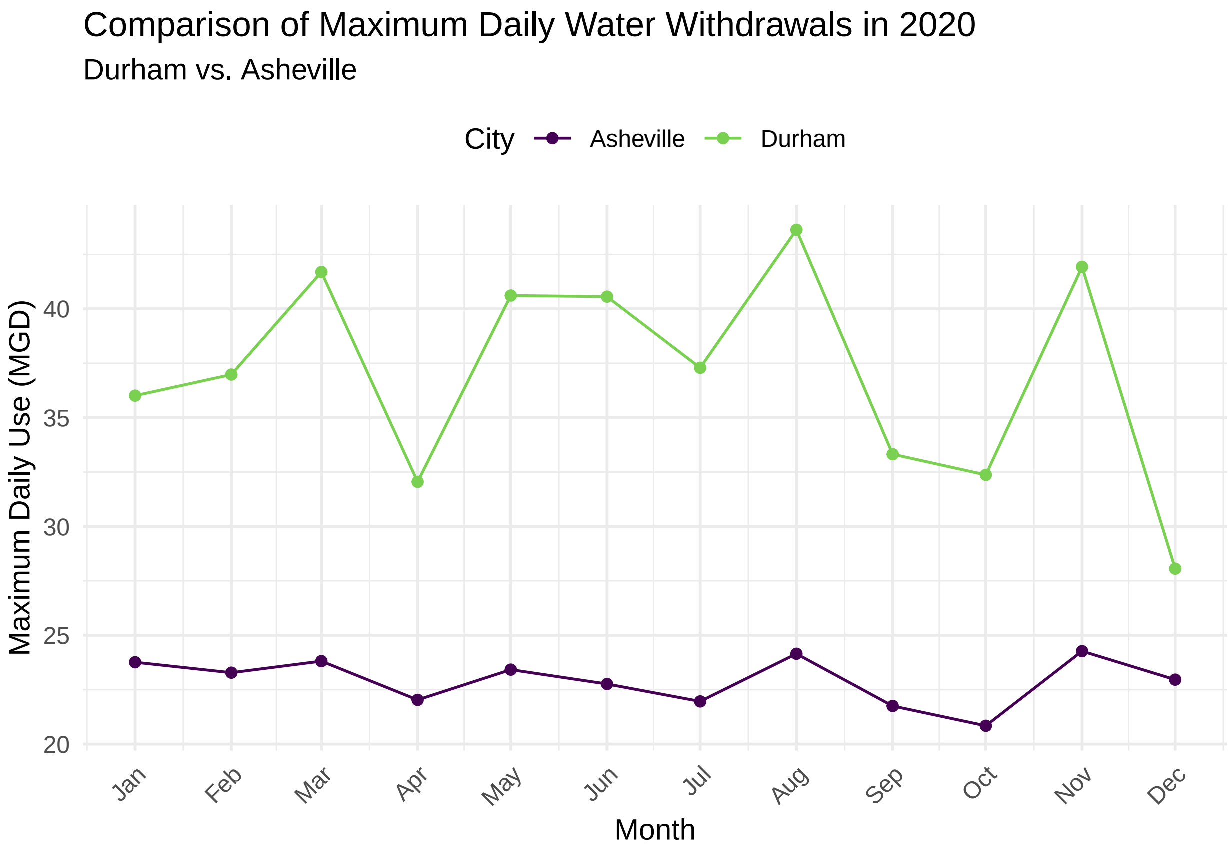
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2020

```
#7
# Get Durham 2020 data
durham_2020 <- scrape_water_data(PWSID = "03-32-010", year = 2020)

# Plot the data
ggplot(durham_2020, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = paste("Maximum Daily Water Withdrawals for", unique(durham_2020$Water_System_Name), "in
       x = "Month",
       y = "Maximum Daily Use (MGD)") +
  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 2020. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```r
#8
# Get Asheville 2020 data
asheville_2020 <- scrape_water_data(PWSID = "01-11-010", year = 2020)

# Combine the datasets
combined_2020 <- bind_rows(
  durham_2020 %>% mutate(City = "Durham"),
  asheville_2020 %>% mutate(City = "Asheville")
)

# Create comparison plot
ggplot(combined_2020, aes(x = Date, y = Max_Day_Use, color = City, group = City)) +
  geom_line() +
  geom_point() +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = "Comparison of Maximum Daily Water Withdrawals in 2020",
       subtitle = "Durham vs. Asheville",
       x = "Month",
       y = "Maximum Daily Use (MGD)",
       color = "City") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "top") +
  scale_color_viridis_d(end = 0.8)
```

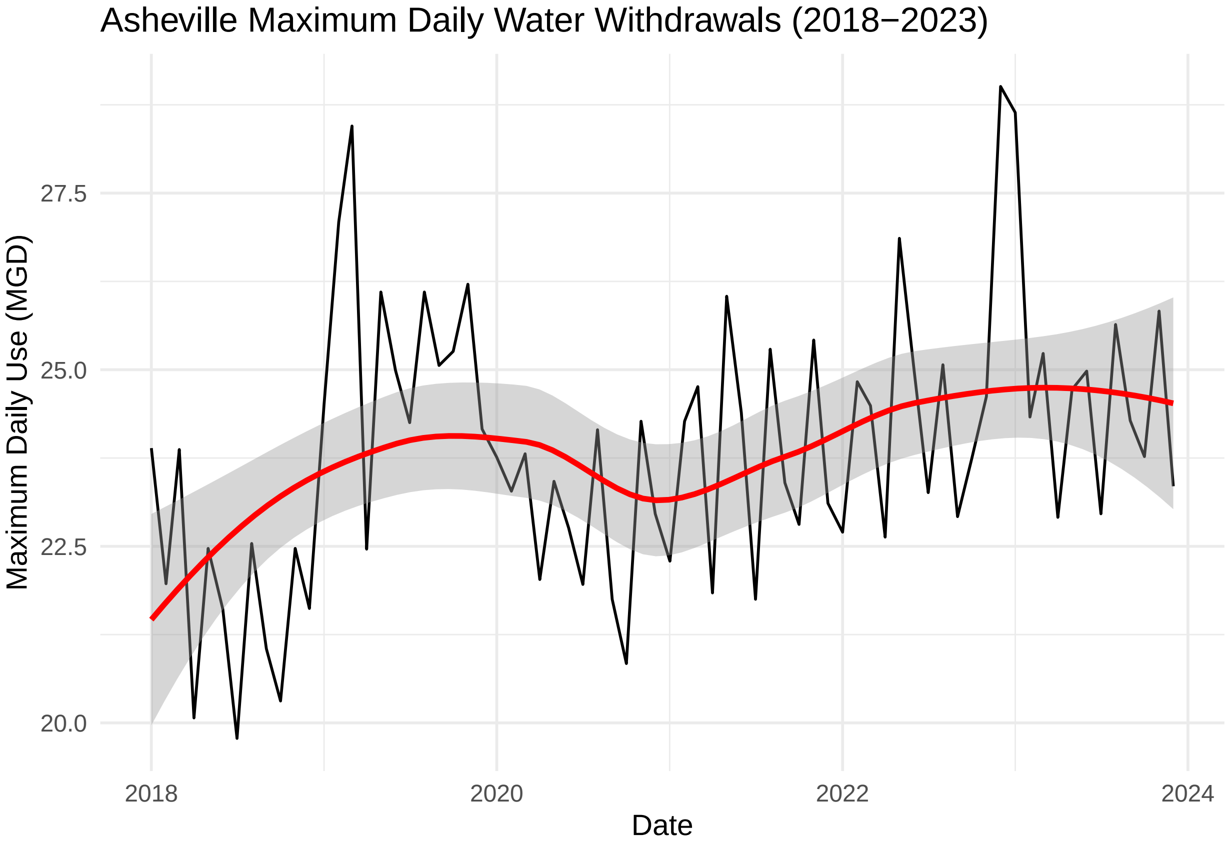


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2023.Add a smoothed line to the plot (method = 'loess').

   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, and use that to construct your plot.

```
#9
# Define years to analyze
years <- 2018:2023

# Create a vector of Asheville's PWSID repeated for each year
pwsids <- rep("01-11-010", length(years))

# Use map2 to apply function to each PWSID-year pair
asheville_multiyear <- map2(
  pwsids,
  years,
  ~scrape_water_data(PWSID = .x, year = .y)
) %>%
  bind_rows()

# Plot the data with a loess smoothing line
ggplot(asheville_multiyear, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_smooth(method = "loess", color = "red") +
  labs(title = "Asheville Maximum Daily Water Withdrawals (2018-2023)",
       x = "Date",
       y = "Maximum Daily Use (MGD)") +
  theme_minimal()
```

```
## ‘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? > Answer: It looks like they are slowly needing to withdraw more water over time. >