Assignment 10: Data Scraping

Ayden Schirmacher

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

```r
#1
remove(list=ls())
library(tidyverse)
library(rvest)
library(lubridate)
library(ggplot2)
```

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
website<-read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024')
website
```

```
## {html_document}
## <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
## [1] <head>\n<title>DWR :: Local Water Supply Planning</title>\n<meta http-equ ...
## [2] <body id="plan">\r\n<!--<div id="division-header">\r\n<a name="top" href= ...
```

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
water_system <- website %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

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

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

maximum_day_use<-website%>%
  html_nodes("th~ td+ td")%>%
  html_text()
```

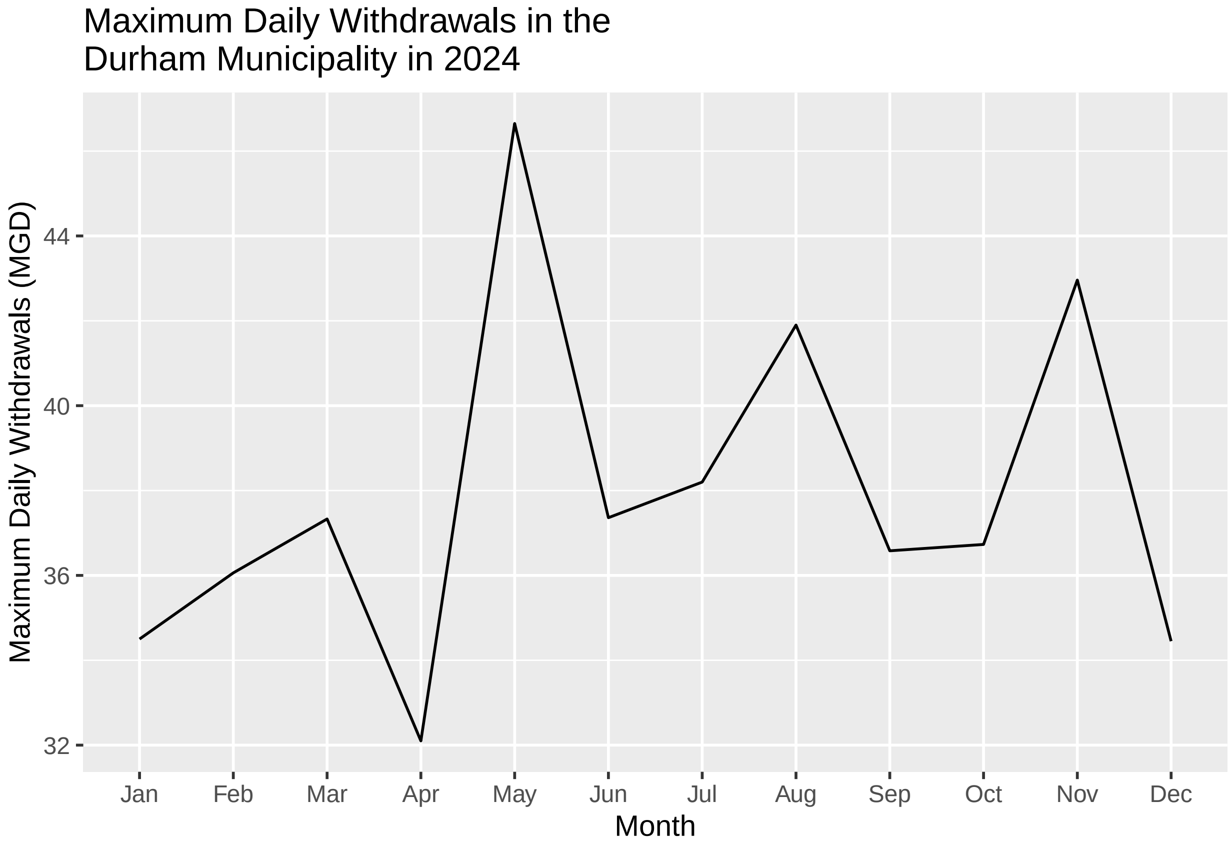
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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.

```
#4

dataframe<-data.frame("Month" = rep(1:12),
                      "Water System" = rep("Durham"),
                      "Ownership" = rep("Municipality"),
                      "PWSID" = rep("03-32-010"),
                      "Maximum Daily Withdrawals" = as.numeric(maximum_day_use)
  )

#5
ggplot(dataframe, aes(x=factor(Month, levels=1:12, labels=month.abb),
                      y=Maximum.Daily.Withdrawals, group=1))+
  geom_line()+
  labs(x="Month", y="Maximum Daily Withdrawals (MGD)",
       title="Maximum Daily Withdrawals in the\nDurham Municipality in 2024")
```



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

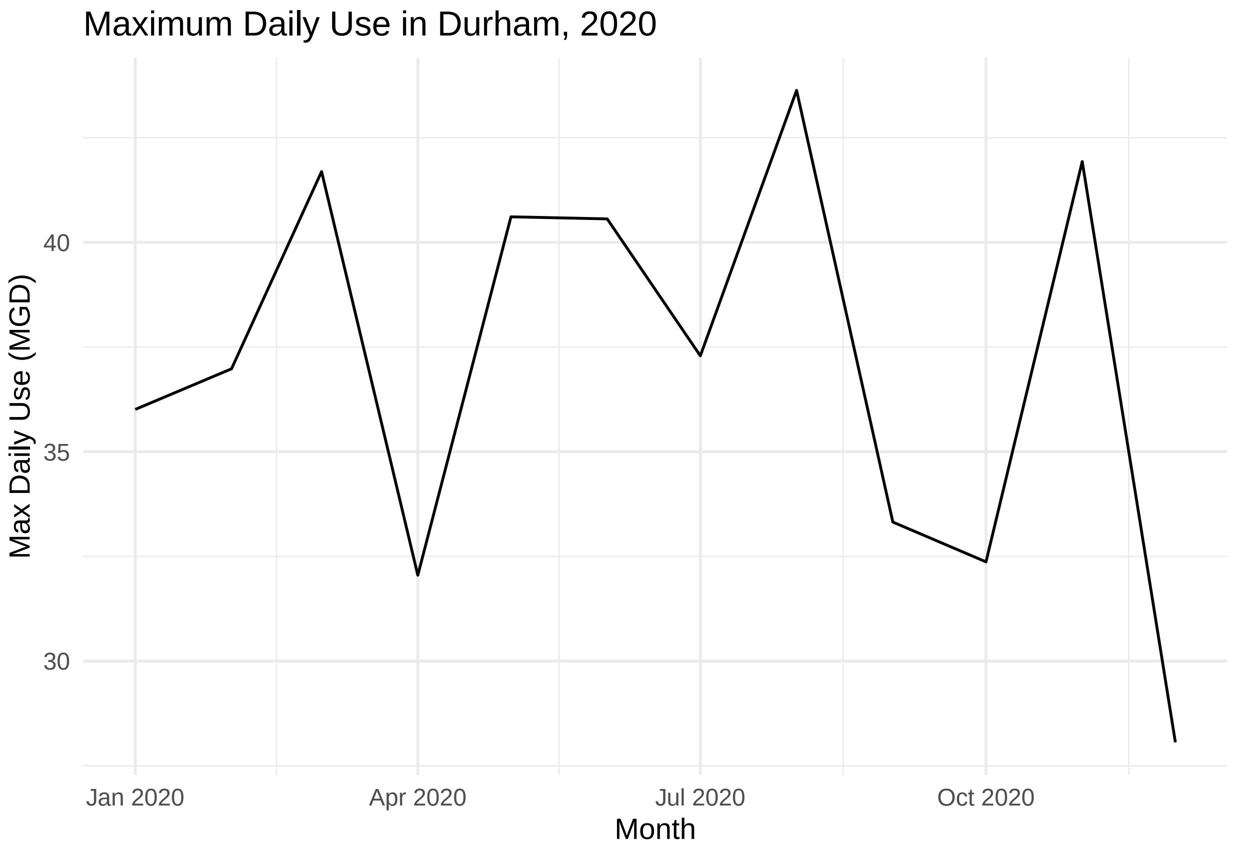
```
#6.

base_url<- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
PWSID<-'03-32-010'
the_year<-2024
scrape_url<- paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', PWSID, '&year=', the_yea
print(scrape_url)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024"
```

```
scrape_website<-read_html(scrape_url)

PWSID_tag<- 'td tr:nth-child(1) td:nth-child(5)'
water_system_tag<- 'div+ table tr:nth-child(1) td:nth-child(2)'
maximum_daily_use_tag<- 'th~ td+ td'
ownership_tag<- 'div+ table tr:nth-child(2) td:nth-child(4)'

PWSID<-website%>%
    html_nodes(PWSID_tag)%>%
    html_text()

WaterSystem<-website%>%
    html_nodes(water_system_tag)%>%
    html_text()

MaxDayUse<-website%>%
    html_nodes(maximum_daily_use_tag)%>%
    html_text()

Ownership<-website%>%
    html_nodes(ownership_tag)%>%
    html_text()


scrape_it <- function(the_year, PWSID) {

  website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', PWSID, '&year=

  PWSID_tag <- 'td tr:nth-child(1) td:nth-child(5)'
  water_system_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  maximum_daily_use_tag <- 'th~ td+ td'
  ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'

  PWSID_value <- website %>% html_nodes(PWSID_tag) %>% html_text()
  WaterSystem_value <- website %>% html_nodes(water_system_tag) %>% html_text()
  MaxDayUse_value <- website %>% html_nodes(maximum_daily_use_tag) %>% html_text()
  Ownership_value <- website %>% html_nodes(ownership_tag) %>% html_text()
  dataframe_LWSP <- data.frame(
  "Month" = rep(1:12),
  "Year" = rep(the_year, 12),
  "Max Daily Use" = as.numeric(MaxDayUse_value)) %>%
  mutate(
    Ownership = Ownership_value,
    PWSID = PWSID_value,
    Water_System = WaterSystem_value,
    Date = my(paste(Month, "-", Year)))

  return(dataframe_LWSP)
}
```

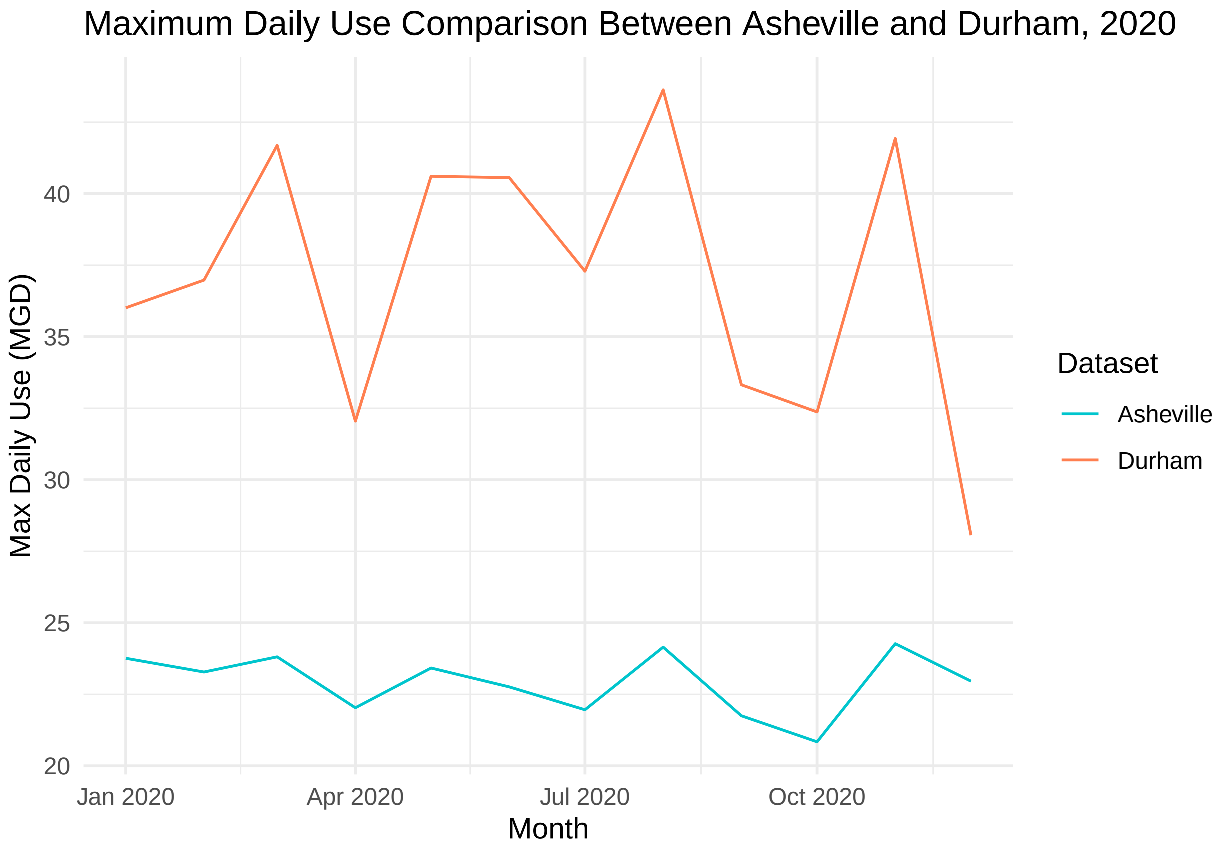
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2020

```
#7
fetch_2020<-scrape_it(2020, '03-32-010')

ggplot(data=fetch_2020) +
  geom_line(aes(x=Date, y=Max.Daily.Use)) +
  theme_minimal() +
  labs(title="Maximum Daily Use in Durham, 2020", x="Month", y="Max Daily Use (MGD)")
```



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.

```
#8

fetch_asheville<-scrape_it(2020, '01-11-010')
fetch_2020<-left_join(fetch_2020, fetch_asheville, by="Date")

ggplot(data=fetch_2020) +
  geom_line(aes(x=Date, y=Max.Daily.Use.x, color="Durham")) +
  geom_line(aes(x=Date, y=Max.Daily.Use.y, color="Asheville")) +
  theme_minimal() +
  labs(title="Maximum Daily Use Comparison Between Asheville and Durham, 2020", x="Month", y="Max Daily
  scale_color_manual(name="Dataset", values=c("Durham"="coral", "Asheville"="turquoise3"))
```

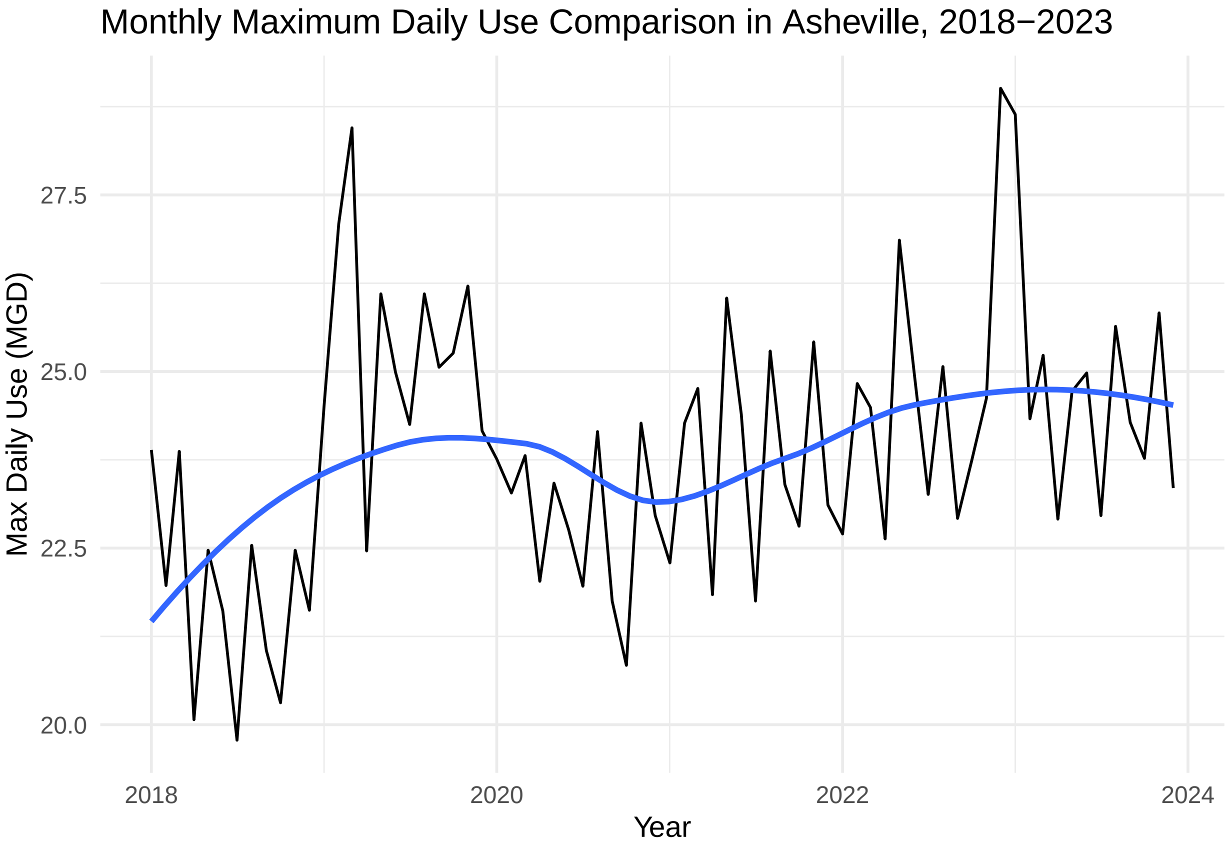


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2023. Add a smoothed line to the plot (method = 'loess').

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
the_years<-rep(2018:2023)
PWSID<-'01-11-010'

asheville_long<-lapply(X = the_years,
                       FUN = scrape_it,
                       PWSID = PWSID)

asheville_combined<-bind_rows(asheville_long)

ggplot(data=asheville_combined, aes(x=Date, y=Max.Daily.Use))+
  geom_line()+
  theme_minimal()+
  geom_smooth(method="loess", se=FALSE)+
  labs(title="Monthly Maximum Daily Use Comparison in Asheville, 2018-2023",
       x="Year",
       y="Max Daily Use (MGD)")
```

```
## `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: > A slight upward trend indicates a slow increase in water usage over time. But the constant oscillations do not seem to necessarily indicate seasonality, so other than the slight upward trend, there does not appear to be a significant trend in water usage over time.