Assignment 10: Data Scraping

Kamil Burak Karayel

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)
#install.packages("rvest")
library(rvest)
library(here)
library(dplyr)
library(lubridate)
library(ggplot2)

getwd()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2025_kbk"
```

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.)

```
#2
#indicate the webpage to be scraped
Durham_LSWP_Page <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=202
```

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
#get water system name from the page
Water_System <- Durham_LSWP_Page %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
Water_System
```

```
## [1] "Durham"
```

```
#get ID number from the page
PWSID <- Durham_LSWP_Page %>%
 html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
 html_text()
PWSID
```

```
## [1] "03-32-010"
```

```
#get owner data from the page
Ownership <- Durham_LSWP_Page %>%
 html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
 html_text()
Ownership
```

```
## [1] "Municipality"
```

```
#get monthly usage data from the page
MaximumDayUse_MGD <- Durham_LSWP_Page %>%
 html_nodes("th~ td+ td") %>%
 html_text()
MaximumDayUse_MGD
```

```
##  [1] "34.5000" "36.0600" "37.3300" "32.1000" "46.6500" "37.3600" "38.2000"
##  [8] "41.9000" "36.5800" "36.7300" "42.9600" "34.4500"
```

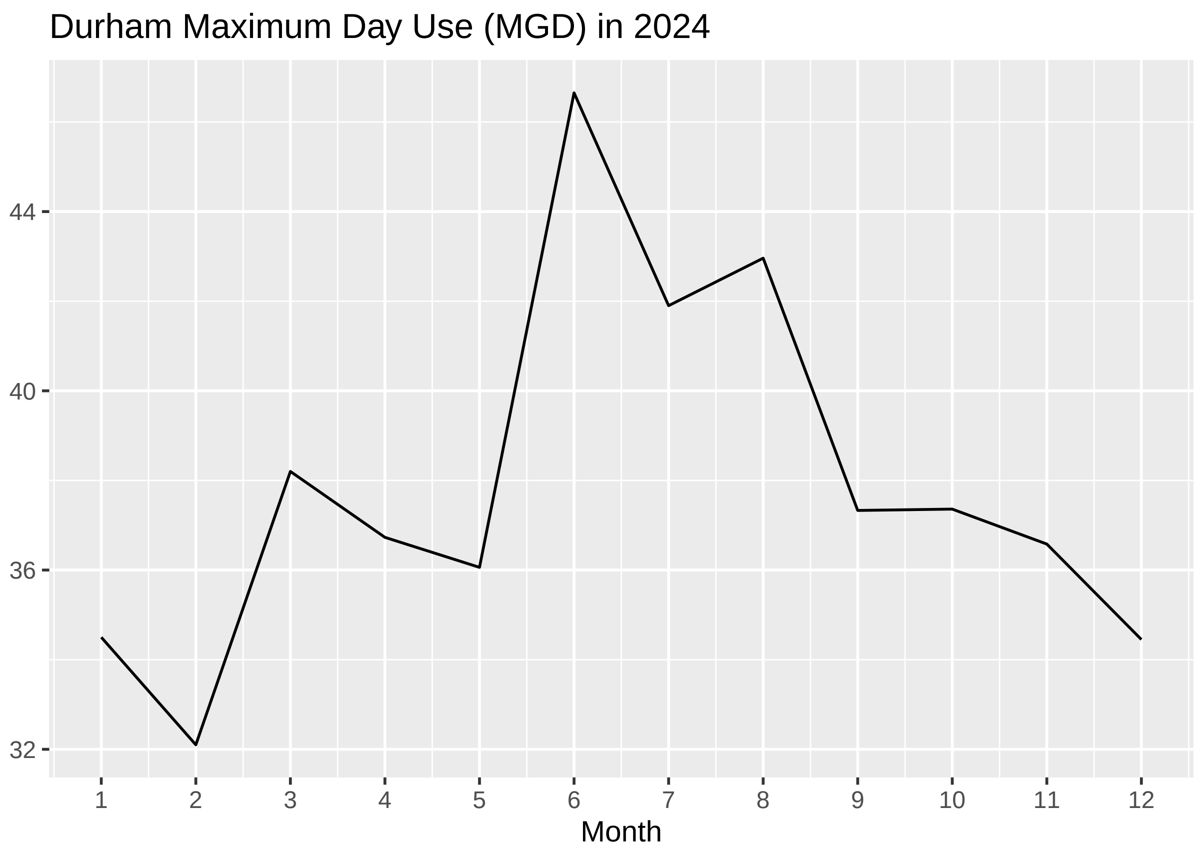
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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
#create a dataframe consists of 7 columns
Q4_df <- data.frame("Water System" = rep(Water_System,12), #repeat water system name 12 times
                    "Owner" = rep(Ownership,12),
                    "PWSID" = rep(PWSID,12),
                    "Month" = c(1,5,9,2,6,10,3,7,11,4,8,12), #create a new column by the order in websi
                    "Year" = rep(2024,12),
                    "MaxDayUse_MGD" = as.numeric(MaximumDayUse_MGD)) %>%
  mutate(Date = my(paste(Month,"-",Year))) %>%  #create a new column consists of Month and Year data
  dplyr::arrange(Month)  #arrange the month column 1 to 12


#5
Q5_plot <- ggplot(Q4_df, aes(x=Month, y=MaxDayUse_MGD, group=1)) +
  geom_line() +
   scale_x_continuous(breaks = 1:12) +
  labs(title="Durham Maximum Day Use (MGD) in 2024",
       y="")

Q5_plot
```



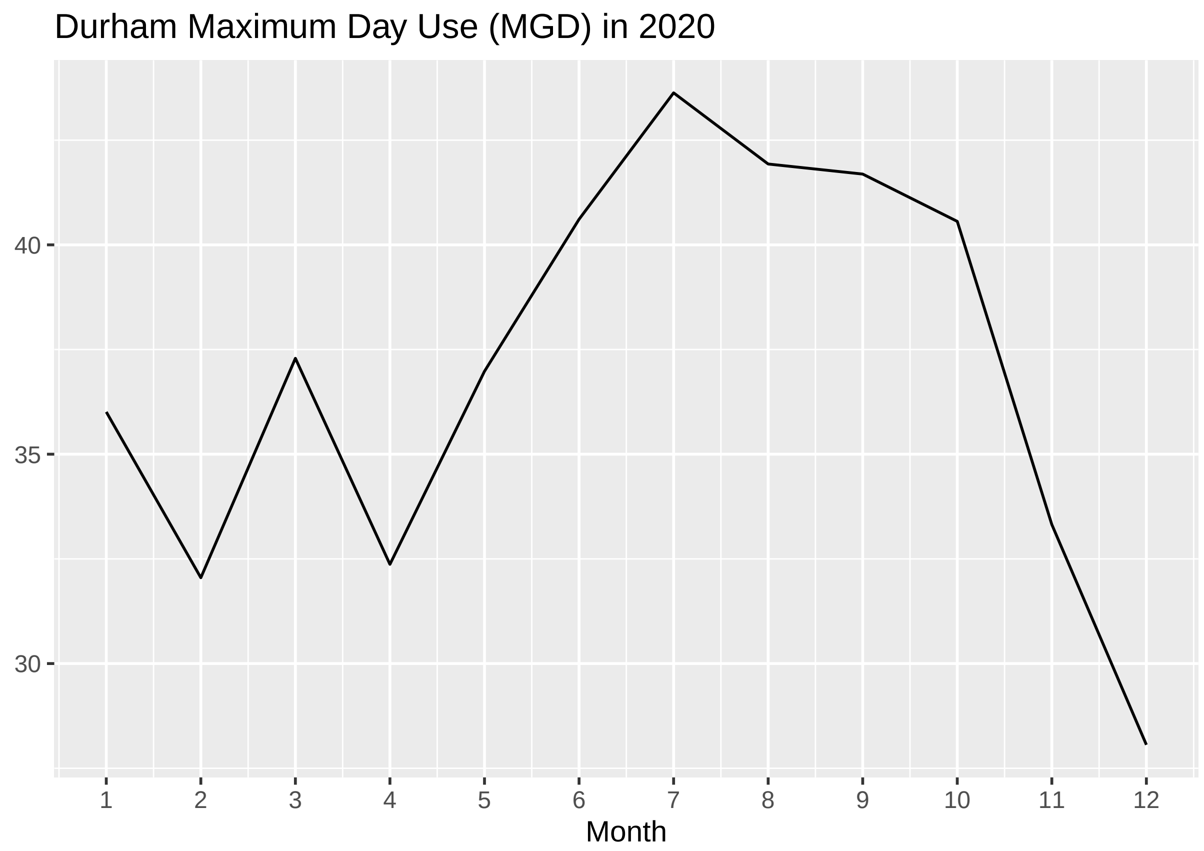
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function with two input - "PWSID" and "year" - that:

- 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 a function for two variables
Q6_function <- function(the_pwsid, the_year) {

  #create website by using variables given by function
  Q6_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                                 the_pwsid, '&year=', the_year))

  Water_System <- Q6_website %>%  #get water system name from the created website
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
    html_text()

  PWSID <- Q6_website %>%
    html_nodes("td tr:nth-child(1) td:nth-child(5)")%>%
    html_text()
  Ownership <- Q6_website %>%
    html_nodes("div+ table tr:nth-child(2) td:nth-child(4)")%>%
    html_text()

  MaximumDayUse_MGD <- Q6_website %>%
    html_nodes("th~ td+ td")%>%
    html_text

  Q6_df <- data.frame("Water_System" = rep(Water_System,12),
                      "Owner" = rep(Ownership,12),
                      "PWSID" = rep(the_pwsid,12),  #use the given variable
                      "Month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                      "Year" = rep(the_year,12),    #use the given variable
                      "MaxDayUse_MGD" = as.numeric(MaximumDayUse_MGD)) %>%
  mutate(Date = my(paste(Month,"-",Year))) %>%
  dplyr::arrange(Month)

  return(Q6_df)
}
```

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

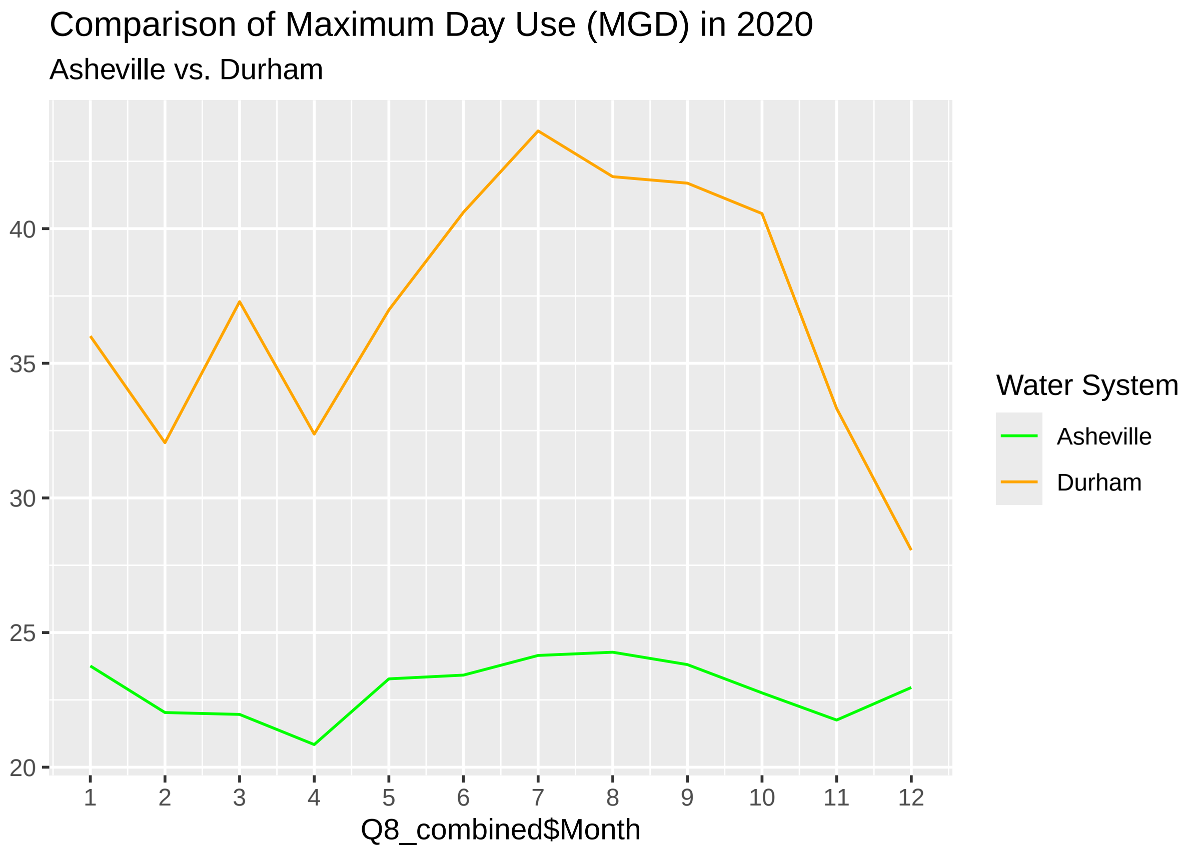
```
#7
#create a new dataframe for Durham in 2020
Q7_df <- Q6_function('03-32-010', 2020)

Q7_plot <- ggplot(Q7_df, aes(x=Month, y=MaxDayUse_MGD, group=1)) +
  geom_line() +
   scale_x_continuous(breaks = 1:12) +
  labs(title="Durham Maximum Day Use (MGD) in 2020",
       y="")

Q7_plot
```



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
#create a new dataframe for Asheville in 2020
Q8_Asheville <- Q6_function('01-11-010', 2020)

#merge two dataframes of Durham and Asheville
Q8_combined <- bind_rows(Q8_Asheville, Q7_df)

Q8_plot <- ggplot(Q8_combined, aes(x=Q8_combined$Month, y=Q8_combined$MaxDayUse_MGD, color=Q8_combined$
  geom_line() +
  scale_x_continuous(breaks = 1:12) +
  labs(title="Comparison of Maximum Day Use (MGD) in 2020",
       subtitle = "Asheville vs. Durham",
       y="",
       color="Water System") +
   scale_color_manual(values = c("Asheville" = "green", "Durham" = "orange"))

Q8_plot
```

```
## Warning: Use of ‘Q8_combined$Month‘ is discouraged.
## i Use ‘Month‘ instead.
## Warning: Use of `Q8_combined$MaxDayUse_MGD` is discouraged.
## i Use `MaxDayUse_MGD` instead.

## Warning: Use of `Q8_combined$Water_System` is discouraged.
## i Use `Water_System` instead.
```

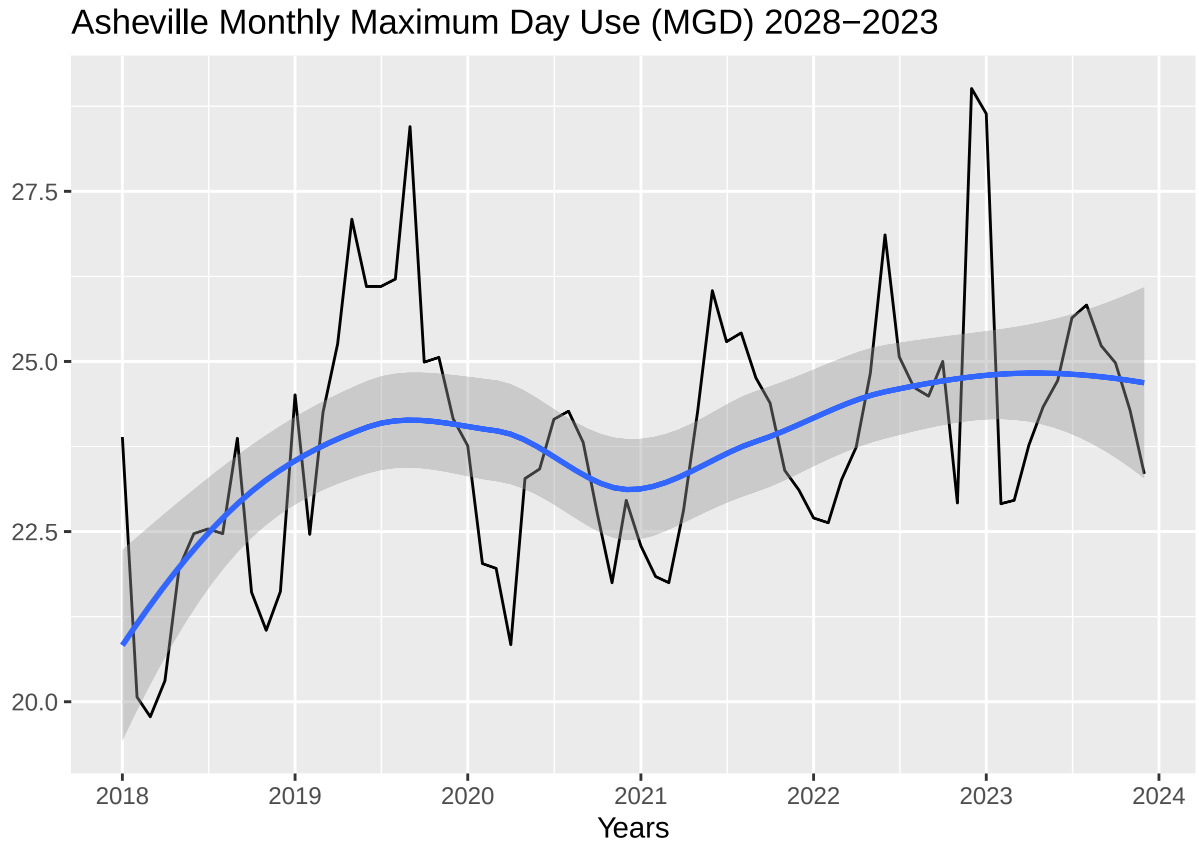


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2023.Add a smoothed line to the plot (method = 'loess').

   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
#use map2 for the function created in Q6 with two variables; ID of Asheville, years from 2018 to 2023
Q9_df <- map2('01-11-010', 2018:2023, Q6_function) %>% bind_rows()

Q9_plot <- ggplot(Q9_df, aes(x=Date, y=MaxDayUse_MGD, group=1)) +
  geom_line() +
  geom_smooth(method = 'loess') +
  labs(title="Asheville Monthly Maximum Day Use (MGD) 2028-2023",
       x="Years",
       y="") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")

Q9_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? > Answer: Just by looking at the plot we can say that Asheville’s water usage has a slight upward trend over the period from 2018 to 2023 >