Assignment 10: Data Scraping

Samantha Jensen

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
# Loading packages
library(tidyverse)
library(rvest)
library(here)
# checking working directory
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

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.)

```r
#2
the_website <-
  read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
```

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 Scraping data and assigning it to variables
system_name <- the_website %>%
  html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
PWSID <- the_website %>%
  html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
Ownership <- the_website %>%
  html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
MDU <- the_website %>% html_nodes('th~ td+ td , th~ td+ td') %>% html_text()
month <- the_website %>% html_nodes('.fancy-table:nth-child(31) tr+ tr th') %>%
  html_text()
```

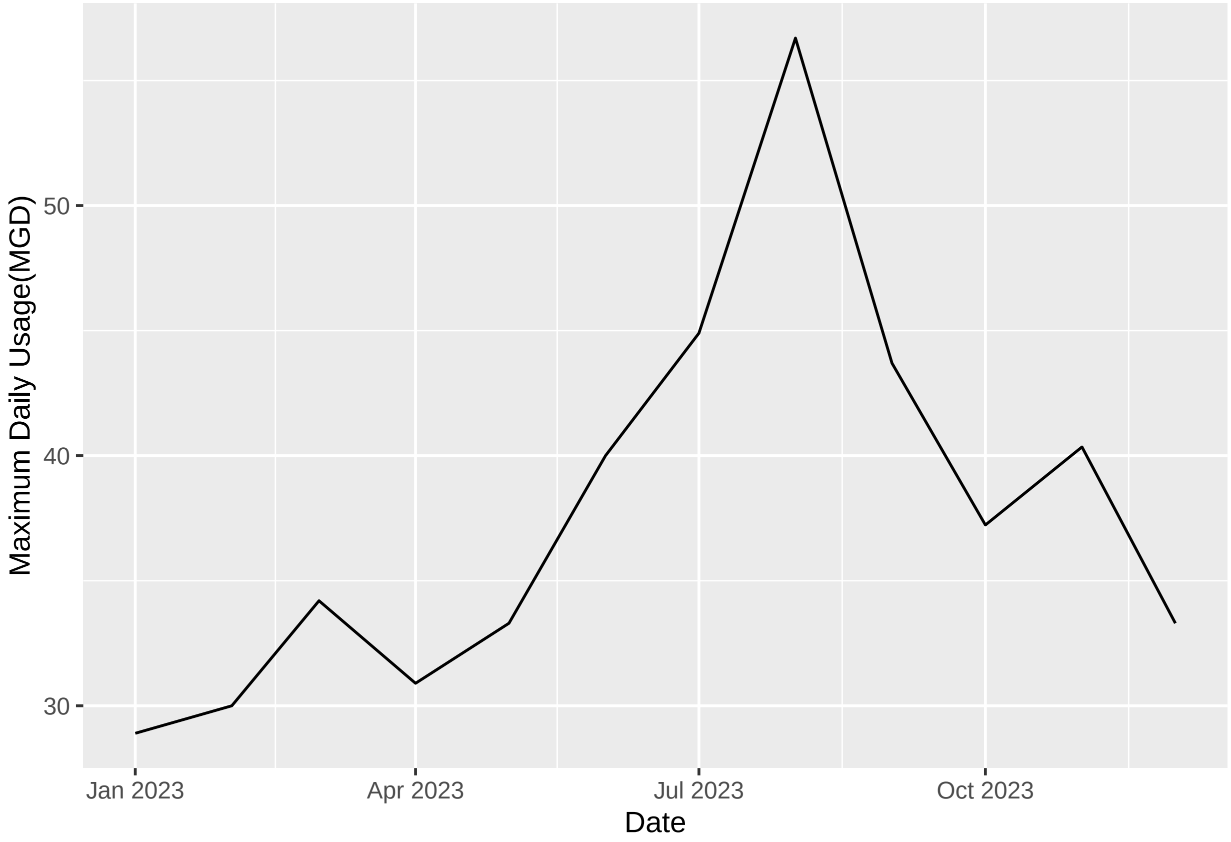
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   > 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 2023, making sure, the months are presented in proper sequence.

```
#4
# Assigning year as 2023
year <- 2023
# creating a df from scraped values
df_mdu <- data.frame("System" = system_name,
                     "PWSID" = PWSID,
                     "Ownership" = Ownership,
                     "Maximum_Daily_Usage" = as.numeric(MDU),
                     "Date" = paste(month, year))
# Changing date column from character format to Date format
intermediate <- paste('01', df_mdu$Date)
df_mdu$Date <- as.Date(intermediate, format = "%d %b %Y")

#5
#plotting max daily usage over time (months)
ggplot(df_mdu, aes(x = Date, y = Maximum_Daily_Usage)) +
  geom_line() +
  labs(y = "Maximum Daily Usage(MGD)")
```

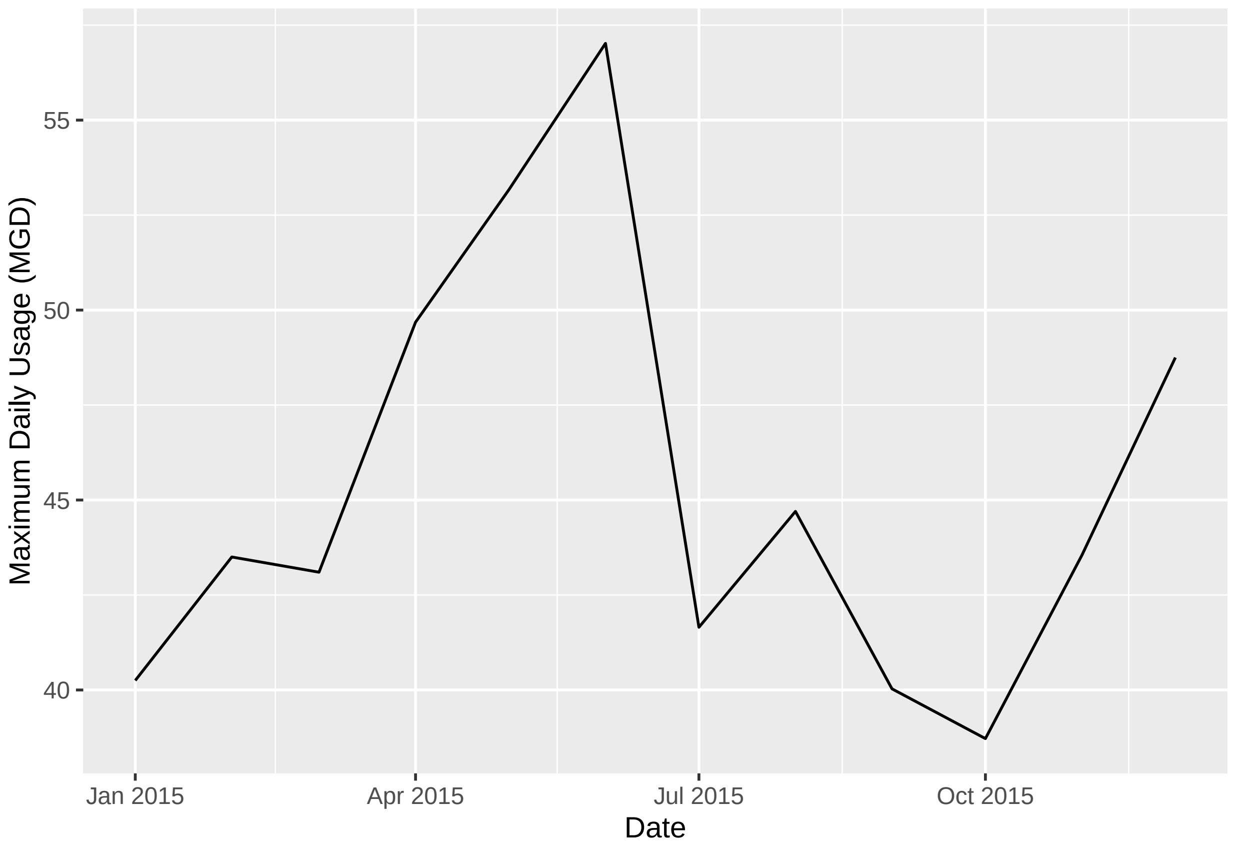


6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```r
#6.
scrape.it <- function(the_year, PWSID_code) {
 #get the URL
   the_url <-
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
           PWSID_code,'&','year=',the_year)

# fetch the website
   the_website <- read_html(the_url)

# Scrape the Data
system_name <- the_website %>%
  html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
PWSID <- the_website %>%
  html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
Ownership <- the_website %>%
  html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
MDU <- the_website %>% html_nodes('th~ td+ td , th~ td+ td') %>% html_text()

# create a data frame
df_mdu <- data.frame("System" = system_name,
                     "PWSID" = PWSID,
                     "Ownership" = Ownership,
                     "Maximum_Daily_Usage" = as.numeric(MDU),
                     "month" = c("Jan", "May", "Sep", "Feb", "Jun", "Oct",
                                 "Mar", 'Jul', "Nov", "Apr", 'Aug', "Dec"),
                     "Date" = paste(month, the_year))
# Changing date column from character format to Date format
intermediate <- paste('01', df_mdu$Date)
df_mdu$Date <- as.Date(intermediate, format = "%d %b %Y")

# Return Data Frame
return(df_mdu)
}
```

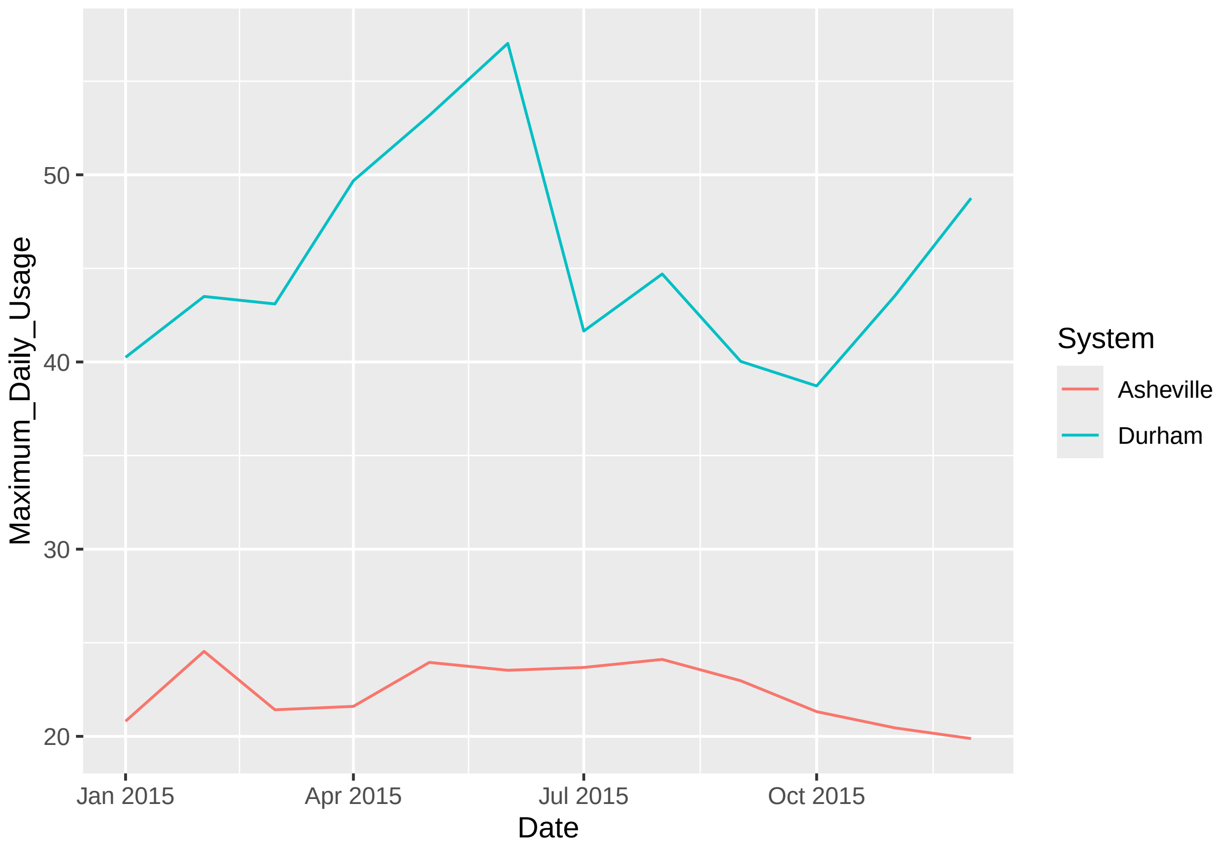
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```r
#7
PWSID_code = '03-32-010'
the_year = '2015'
durham <- scrape.it(the_year, PWSID_code)

ggplot(durham, aes(x = Date, y = Maximum_Daily_Usage)) +
  geom_line() +
  labs(y = "Maximum Daily Usage (MGD)")
```



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
PWSID_code <- '01-11-010'
the_year = '2015'
asheville <- scrape.it(the_year, PWSID_code)

# combining durham and asheville datsets
combined <- bind_rows(durham, asheville)
# plotting both durham and asheville on one line graph
ggplot(data = combined, aes(y = Maximum_Daily_Usage, x = Date, color = System)) +
  geom_line()
```

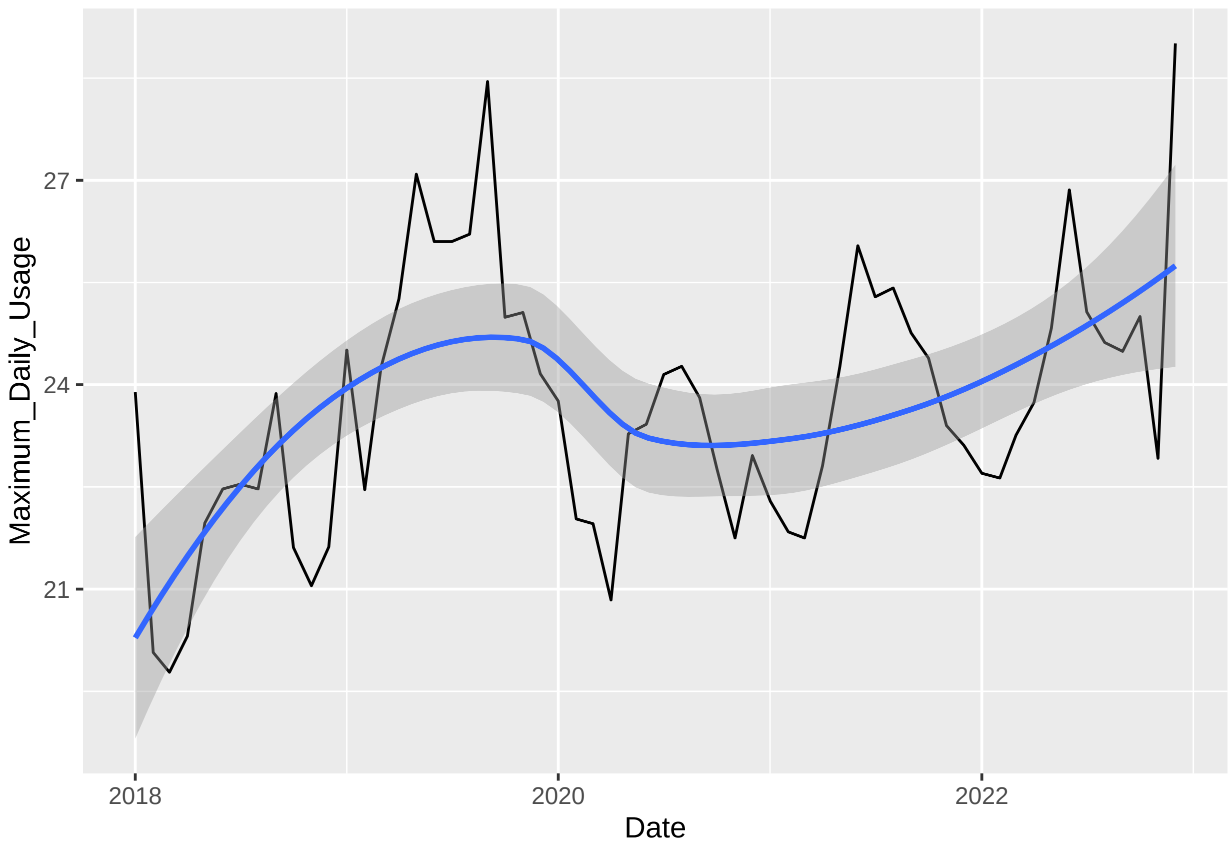


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022.Add a smoothed line to the plot (method = 'loess').

   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
# Set variables
PWSID_code <- '01-11-010'
the_year <- seq(2018,2022)

#"Map" the "scrape.it" function to retrieve data for the sequence of years
dfs_ashville <- map2(the_year, PWSID_code, scrape.it)

# Combining all the dfs into one
dfs_years <- bind_rows(dfs_ashville)

#Plot
ggplot(dfs_years,aes(y = Maximum_Daily_Usage, x=Date)) +
 geom_line() +
  geom_smooth(method = 'loess')
```

```
## '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: By looking at the plot, Asheville's water usage is increasing overtime overall. >