Assignment 10: Data Scraping

Fiona Price

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. Load in the necessary packages.
library(tidyverse)
library(lubridate)
library(here); here()
```

```
## [1] "/home/guest/ede_fall2024"
```

```
library(rvest)

#Check 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.)

```
#2. Read the contents into a webpage object.
webpage <- read_html(
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
webpage
```

```
## {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)".

```
#3. Using the chrome SelectorGadget extension, collect the water system name.
water_system_name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
#Check to see that it's "Durham."
water_system_name
```

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

```
#Using the chrome SelectorGadget extension, collect the PWSID.
PWSID <- webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
#Check to see that it's "03-32-010."
PWSID
```

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

```
#Using the chrome SelectorGadget extension, collect the Ownership data.
ownership <- webpage %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
#Check to see that it's "Municipality."
ownership
```

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

```
#Using the chrome SelectorGadget extension, collect the MGD data.
MGD <- webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()
#Check to see that it's a vector of 12 numeric values.
MGD
```

```
##  [1] "28.9000" "33.3000" "43.7000" "30.0000" "40.0000" "37.2300" "34.2000"
##  [8] "44.9000" "40.3500" "30.9000" "56.7000" "33.3000"
```

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...

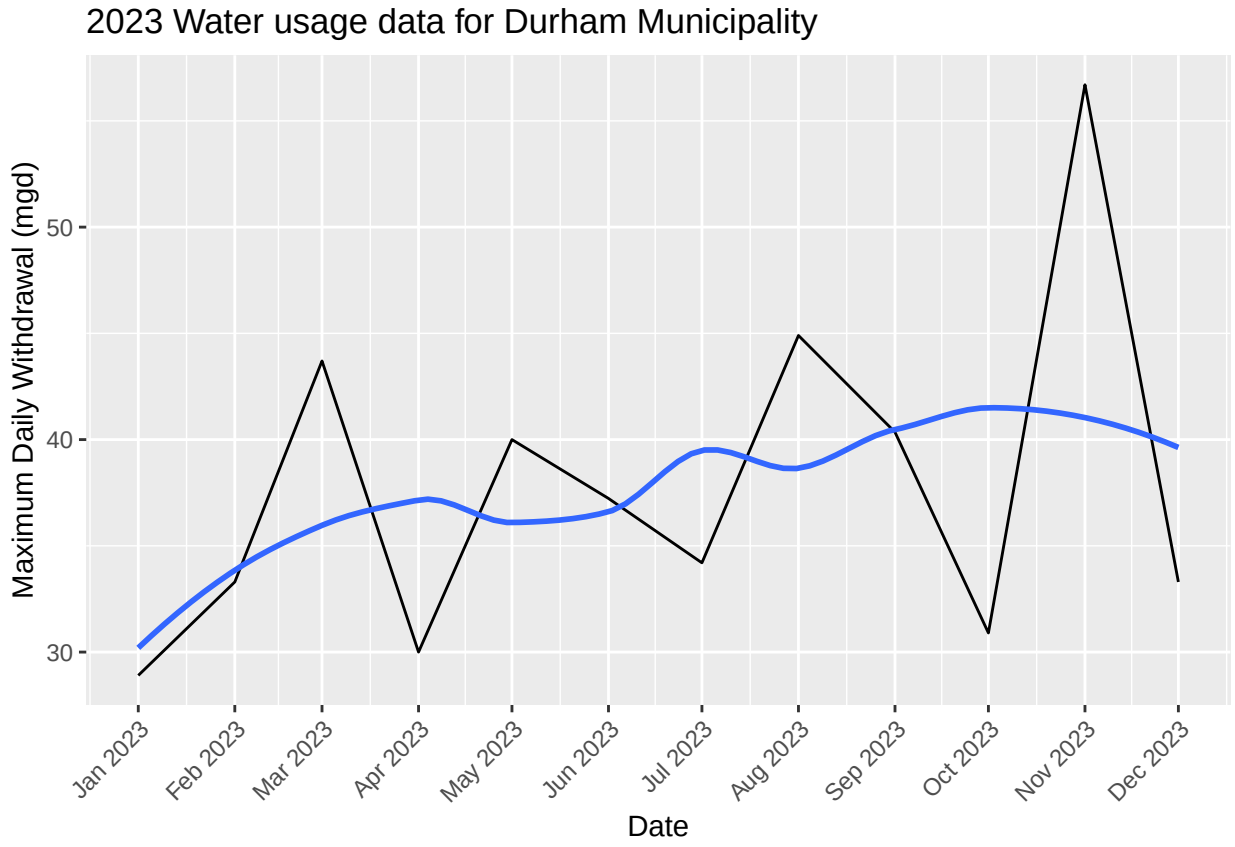
5. Create a line plot of the maximum daily withdrawals across the months for 2023, making sure, the months are presented in proper sequence.

```
#4. Start with a dataframe that has a month, year, and MGD column.
df_withdrawals <- data.frame("Month" = rep(1:12),
                      #Create a column that repeats the month for 1-12
                             "Year" = rep(2023,12),
                      #Create a column that repeats 2023 12 times
                             "Max_Daily_Use" = as.numeric(MGD))
                      #Create a numeric MGD column

#Now mutate the dataframe to include the system name, the PWSID, the ownership
#type, and the date.
df_withdrawals <- df_withdrawals %>%
  mutate(System_Name = !!water_system_name,
         PWSID = !!PWSID,
         Ownership_Type = !!ownership,
         Date = my(paste(Month,"-",Year))) #Paste my new year and month columns
#created above to a date column
```

```r
#5. Create a line plot of the maximum daily withdrawals across the months for 2023.
ggplot(df_withdrawals,aes(x=Date,y=Max_Daily_Use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2023 Water usage data for",water_system_name, ownership),
       y="Maximum Daily Withdrawal (mgd)",
       x="Date") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + #Rotate to include
  #more x-axis labels
  scale_x_date(
    date_breaks = "1 month", #Adjust x-axis labels to include all months
    date_labels = "%b %Y")      #Adjust label to include name of month
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



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.
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_pwsid <- '03-32-010'
```

```r
the_year <- 2015
the_scrape_url <- paste0(the_base_url, 'pwsid=', the_pwsid, '&year=', the_year)
print(the_scrape_url)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2015"
```

```r
#Create our scraping function
scrape.it <- function(the_year, the_pwsid){

  #Retrieve the website contents
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?',
                                  'pwsid=', the_pwsid, '&year=', the_year))

  #Set the element address variables (determined in the previous step)
  system_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
  mgd_tag <- 'th~ td+ td'

  #Scrape the data items
  the_system <- the_website %>% html_nodes(system_tag) %>% html_text()
  the_ownership <- the_website %>% html_nodes(ownership_tag) %>% html_text()
  the_pwsid <- the_website %>% html_nodes(pwsid_tag) %>% html_text()
  the_mgd <- the_website %>% html_nodes(mgd_tag) %>% html_text()

  #Convert to a dataframe
  df_withdrawals <- data.frame("Month" = rep(1:12),
                               "Year" = rep(the_year,12),
                               "Max_Daily_Withdrawals" = as.numeric(the_mgd)) %>%
    mutate(System_Name = !!the_system,
         Ownership_Type = !!the_ownership,
         PWSID = !!the_pwsid,
         Date = my(paste(Month,"-",Year)))
  #Return the dataframe
  return(df_withdrawals)
}
```

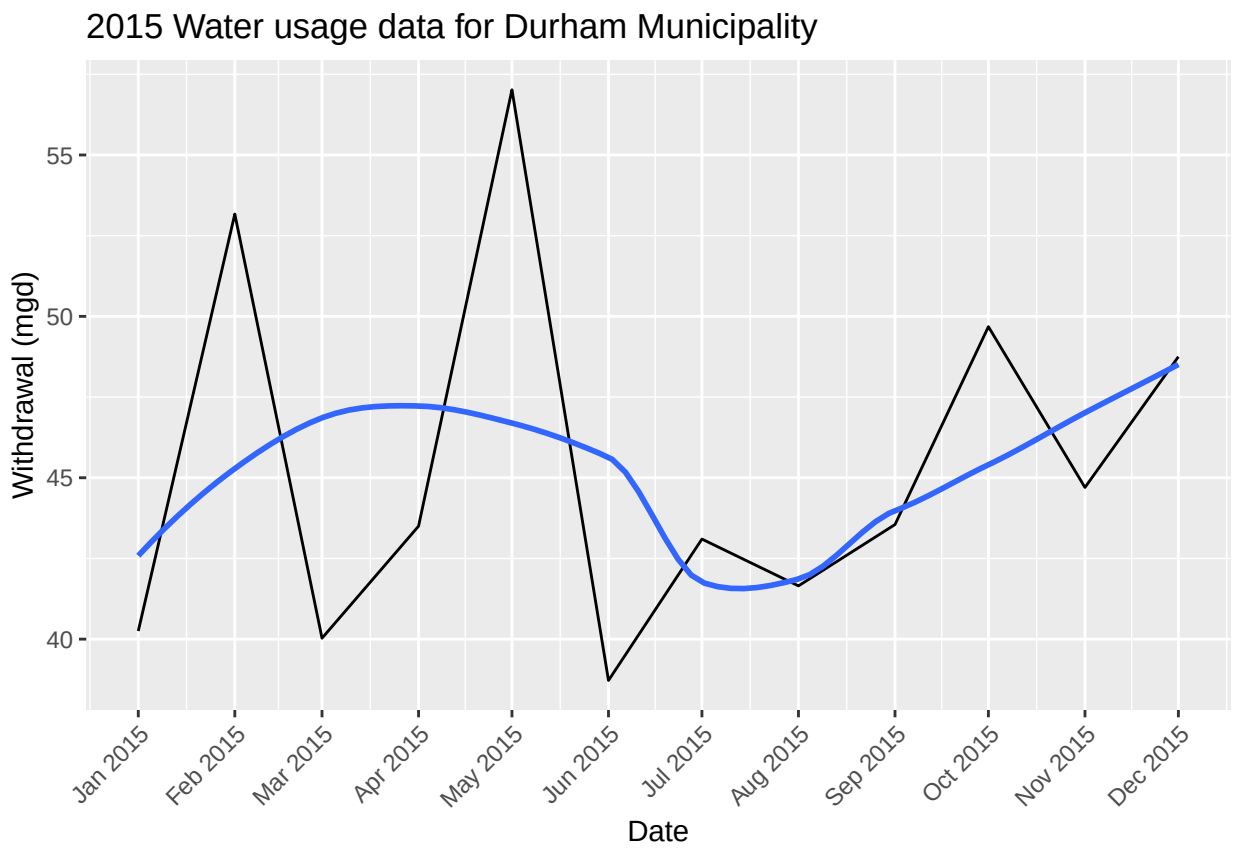
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```r
#7. Extract data for Durham 2015.
df_withdrawals <- scrape.it(2015,'03-32-010')
view(df_withdrawals)

#Plot daily max withdrawals.
ggplot(df_withdrawals,aes(x=Date,y=Max_Daily_Withdrawals)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(the_year,"Water usage data for", df_withdrawals$System_Name,
                     df_withdrawals$Ownership_Type),
       y="Withdrawal (mgd)",
       x="Date") +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + #Rotate to include
  #more x-axis labels
  scale_x_date(
    date_breaks = "1 month", #Adjust x-axis labels to include all months
    date_labels = "%b %Y")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

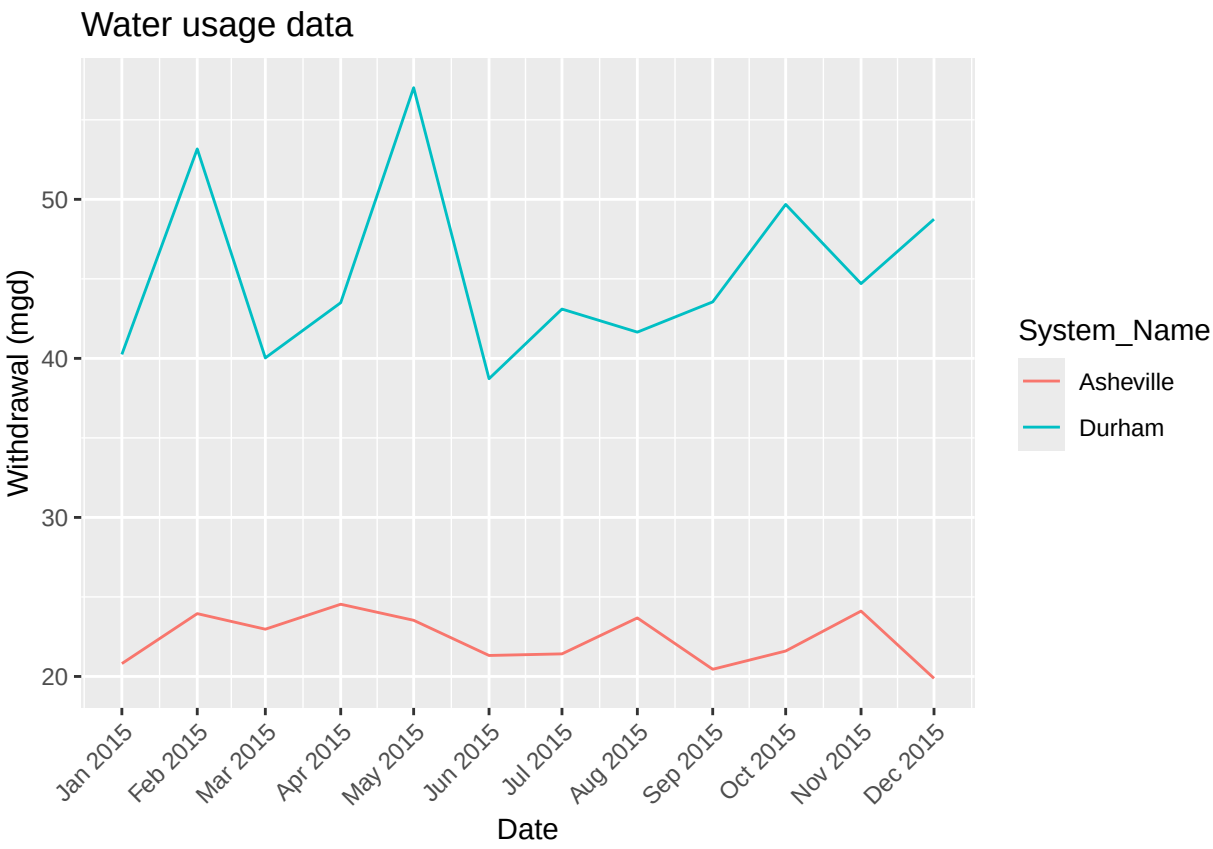


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8. Extract data for Asheville 2015.
asheville_df <- scrape.it(2015,'01-11-010')
#Combine with the Durham data.
combined_mgd <- rbind(df_withdrawals, asheville_df)

#Plot Asheville vs. Durham water withdrawals.
ggplot(combined_mgd,aes(x=Date,y=Max_Daily_Withdrawals,color=System_Name)) +
  geom_line() +
  labs(title = "Water usage data",
       y="Withdrawal (mgd)",
       x="Date") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) + #Rotate to include
```

```
  #more x-axis labels
  scale_x_date(
    date_breaks = "1 month", #Adjust x-axis labels to include all months
    date_labels = "%b %Y")
```



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').

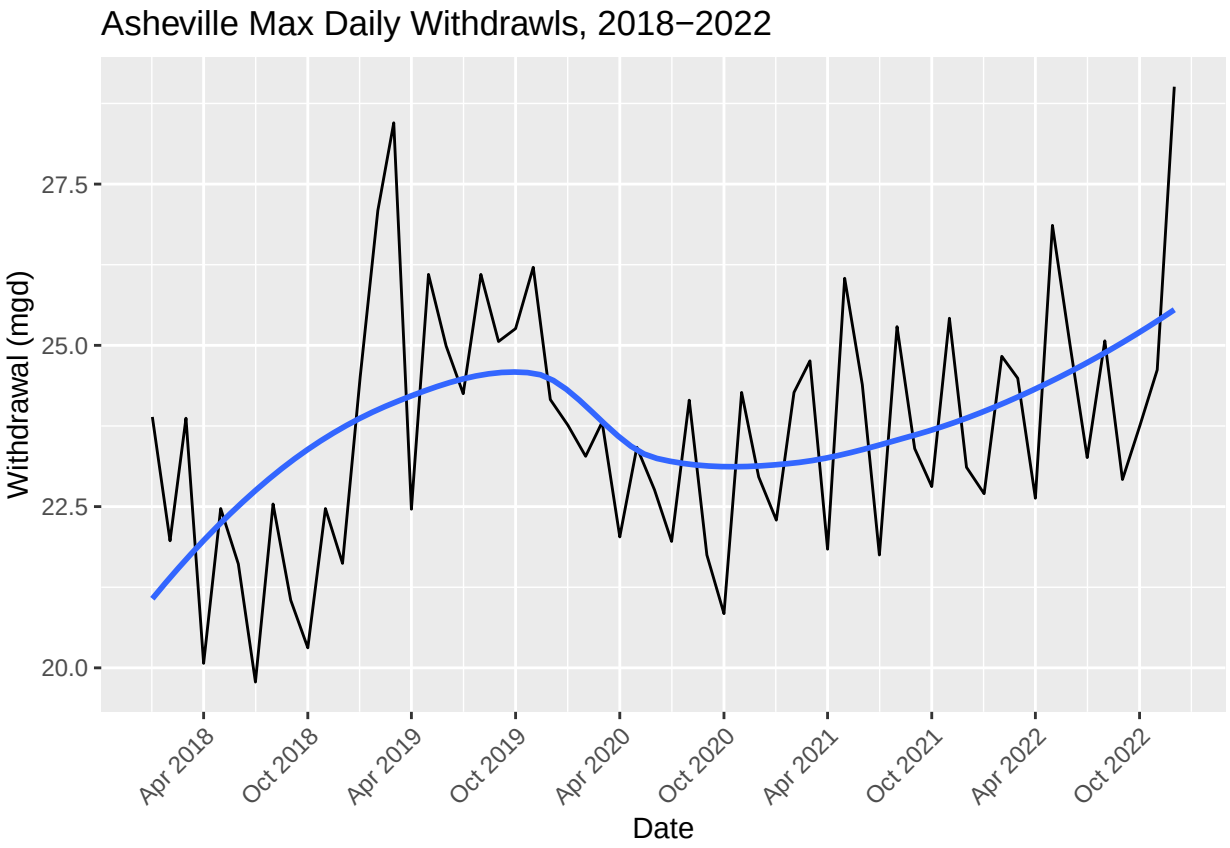
   TIP: See Section 3.2 in the "10_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```
#9. Subset the years we want to look at.
the_years = (2018:2022)
asheville_pwsid <- '01-11-010'
#Bind the dataframes together, based on the years and pwsid.
asheville_dfs <- map2(the_years, asheville_pwsid, scrape.it) %>%
  bind_rows()

#Plot. Add a smoothed line.
ggplot(asheville_dfs,aes(y = Max_Daily_Withdrawals, x=Date)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "Asheville Max Daily Withdrawls, 2018-2022",
```

```
      y="Withdrawal (mgd)",
      x="Date") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + #Rotate to include
#more x-axis labels
scale_x_date(
  date_breaks = "6 months", #Adjust x-axis labels to include every half a year
  date_labels = "%b %Y")
```

```
## `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: Yes, Asheville's max daily water use seems to be increasing over time, though there was a dip in 2020 (possibly related to COVID-19). >