Assignment 10: Data Scraping

Gaby Czarniak

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
library(tidyverse)
library(lubridate)
library(viridis)
library(here)
library(dplyr)
here()
```

```
## [1] "/home/guest/gaby-cz_EDE_Fall2023"
```

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

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

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

```
# check working directory
getwd()
```

```
## [1] "/home/guest/gaby-cz_EDE_Fall2023"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2022 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=2022

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

```
#2
# link to provided web site
# use rvest's read_html to read web page into parseable object
the_website <- read_html(
  "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022")
```

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
# Extract elements with provided tags
# Get the text associated with an element

# Water system name
water_system_name <-
  the_website %>%
  html_nodes('table:nth-child(7) tr:nth-child(1) td:nth-child(2)') %>%
  html_text()

# PWSID
```

```
pwsid <-
  the_website %>%
  html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
  html_text()

# Ownership
ownership <-
  the_website %>%
  html_nodes('table:nth-child(7) tr:nth-child(2) td:nth-child(4)') %>%
  html_text()

# Maximum Day Use (MGD)
max_day_use_MGD <-
  the_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 2022

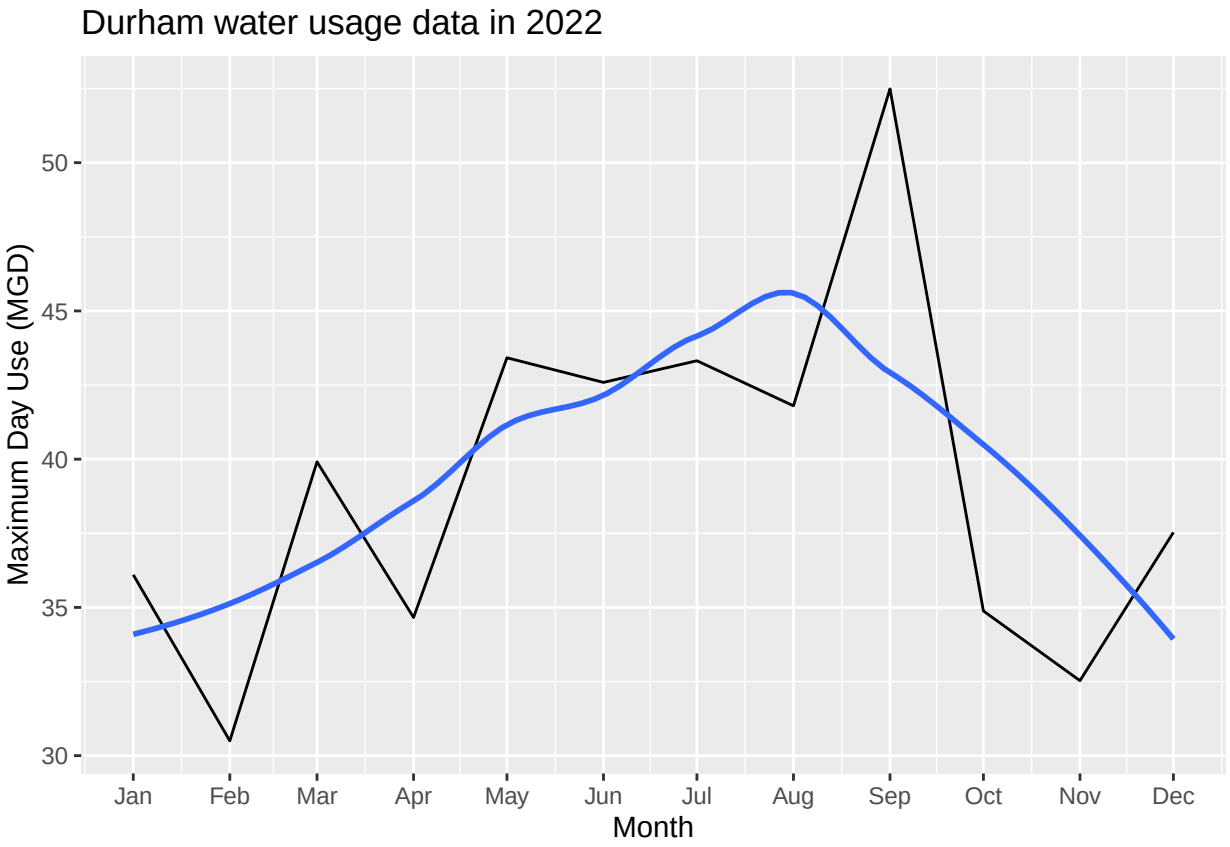
```
#4
# month object
month <- c("Jan", "May", "Sep",
           "Feb", "Jun", "Oct",
           "Mar", "Jul", "Nov",
           "Apr", "Aug", "Dec")
# year object
year <- rep.int(2022,length(max_day_use_MGD))
# date object from the above two objects
date <- paste0(month,' ',year)

# create data frame for the scraped data
durham_2022_LWSP_df <- data.frame(
  "Water System Name" = water_system_name,
  "PWSID" = pwsid,
  "Ownership" = ownership,
  "Max_Day_Use_mgd" = as.numeric(max_day_use_MGD),
  "Date" = my(date),
  "Year" = year)

#5
```

```r
# plot the max daily withdrawals across the months for Durham 2022
ggplot(durham_2022_LWSP_df, aes(x = Date, y = Max_Day_Use_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "Durham water usage data in 2022",
       y="Maximum Day Use (MGD)",
       x="Month") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```

```
## `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. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```r
#6.
# create function that can scrape data for any PWSID and year
scrape.it <- function(the_year, the_PWSID_2){

  #Retrieve the website contents, create a hook into the website
  the_website <- read_html(paste0(
                           'https://www.ncwater.org/WUDC/app/LWSP/report.php?',
                           'pwsid=', the_PWSID_2, '&year=', the_year))
```

```
  #Set the element address variables / tags
  the_water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  the_pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
  the_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  the_max_day_use_tag <- 'th~ td+ td'

  new_month <- c("Jan", "May", "Sep",
                 "Feb", "Jun", "Oct",
                 "Mar", "Jul", "Nov",
                 "Apr", "Aug", "Dec")
  new_year <- rep.int(the_year,length(max_day_use_MGD))
  new_date <- paste0(new_month,' ',new_year)

  #Scrape the data items
  the_water_system_name <- the_website %>%
    html_nodes(the_water_system_name_tag) %>%
    html_text()
  the_pwsid <- the_website %>%
    html_nodes(the_pwsid_tag) %>%
    html_text()
  the_ownership <- the_website %>%
    html_nodes(the_ownership_tag) %>%
    html_text()
  the_max_day_use <- the_website %>%
    html_nodes(the_max_day_use_tag) %>%
    html_text()

#Construct a data frame from the scraped data
scraped_df <- data.frame("Maximum Day Use" = as.numeric(the_max_day_use),
                         "Date" = my(new_date),
                         "Year" = rep(the_year,12)) %>%
                mutate("Water System Name" = !!the_water_system_name,
                         "PWSID" = !!the_pwsid,
                         "Ownership" = !!the_ownership)

# Return the data frame
return(scraped_df)
}
```

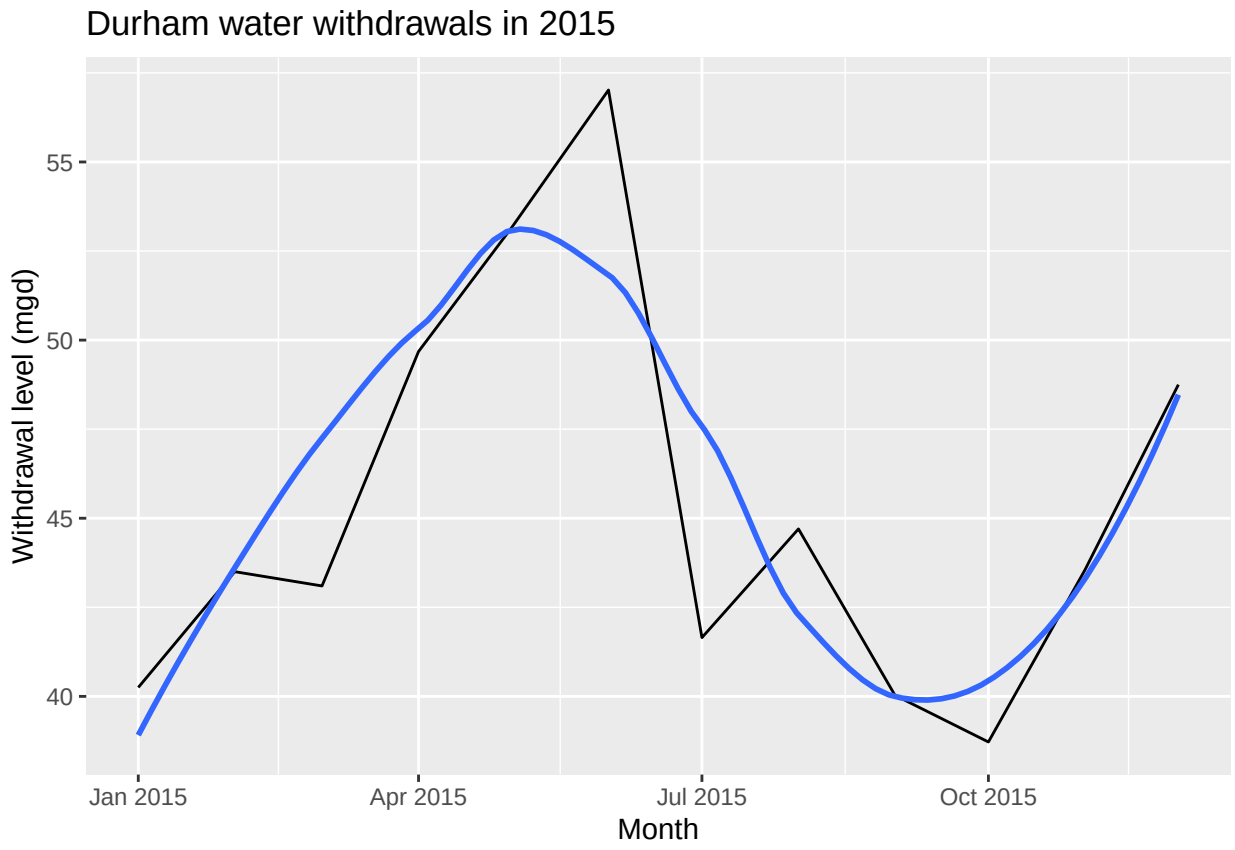
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
# extract 2015 max daily withdrawals data for Durham
Durham_2015_df <- scrape.it(2015,'03-32-010')
view(Durham_2015_df)

# plot the Durham 2015 data
ggplot(Durham_2015_df, aes(x = Date, y = Maximum.Day.Use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Durham water withdrawals in 2015"),
       y="Withdrawal level (mgd)",
```

```
       x="Month")
```

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

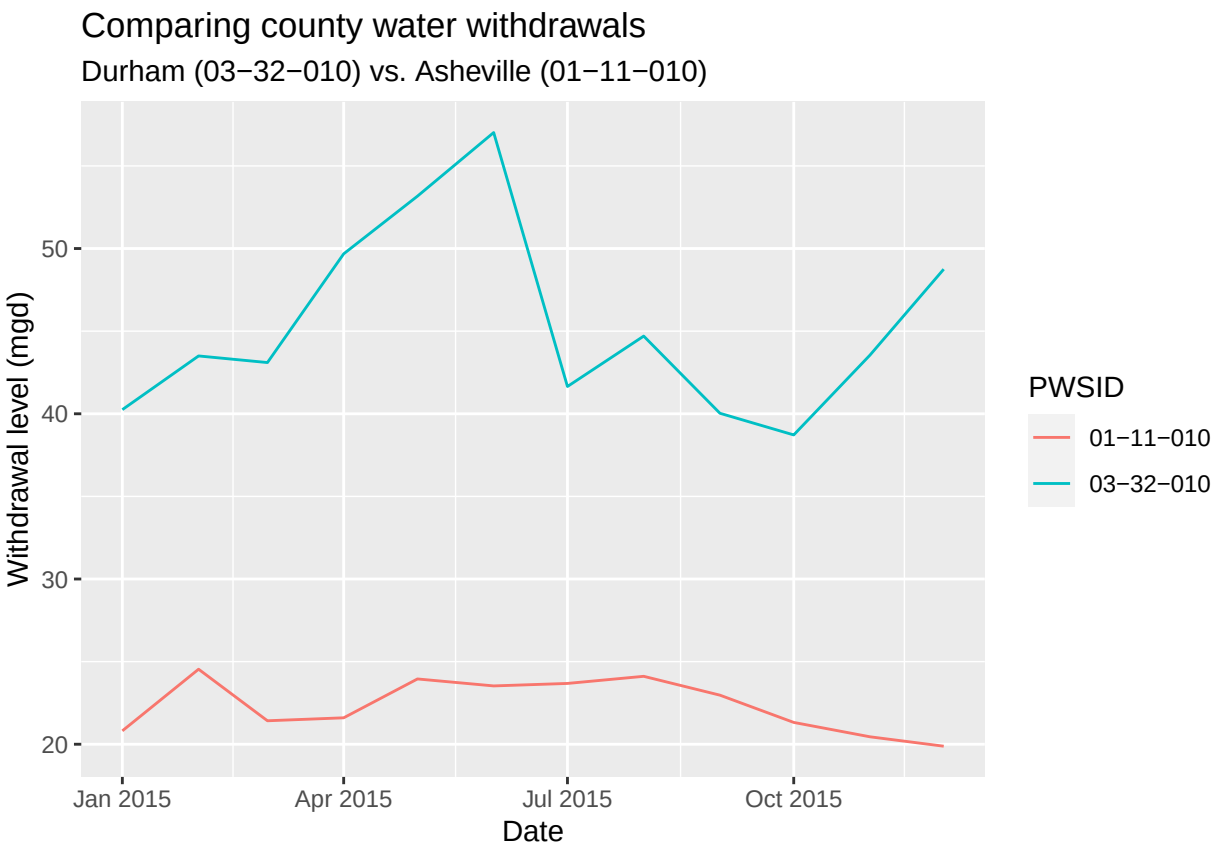


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8
# extract data for Asheville 2015
Asheville_2015_df <- scrape.it(2015,'01-11-010')
view(Asheville_2015_df)

# combine the Durham data with the Asheville data for 2015
combined_max_day_use <- rbind(Durham_2015_df, Asheville_2015_df)

# plot the newly combined df all on one graph
ggplot(combined_max_day_use, aes(x = Date, y = Maximum.Day.Use, color= PWSID)) +
  geom_line() +
  labs(title = paste("Comparing county water withdrawals"),
       subtitle = "Durham (03-32-010) vs. Asheville (01-11-010)",
       y="Withdrawal level (mgd)",
       x="Date")
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.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.

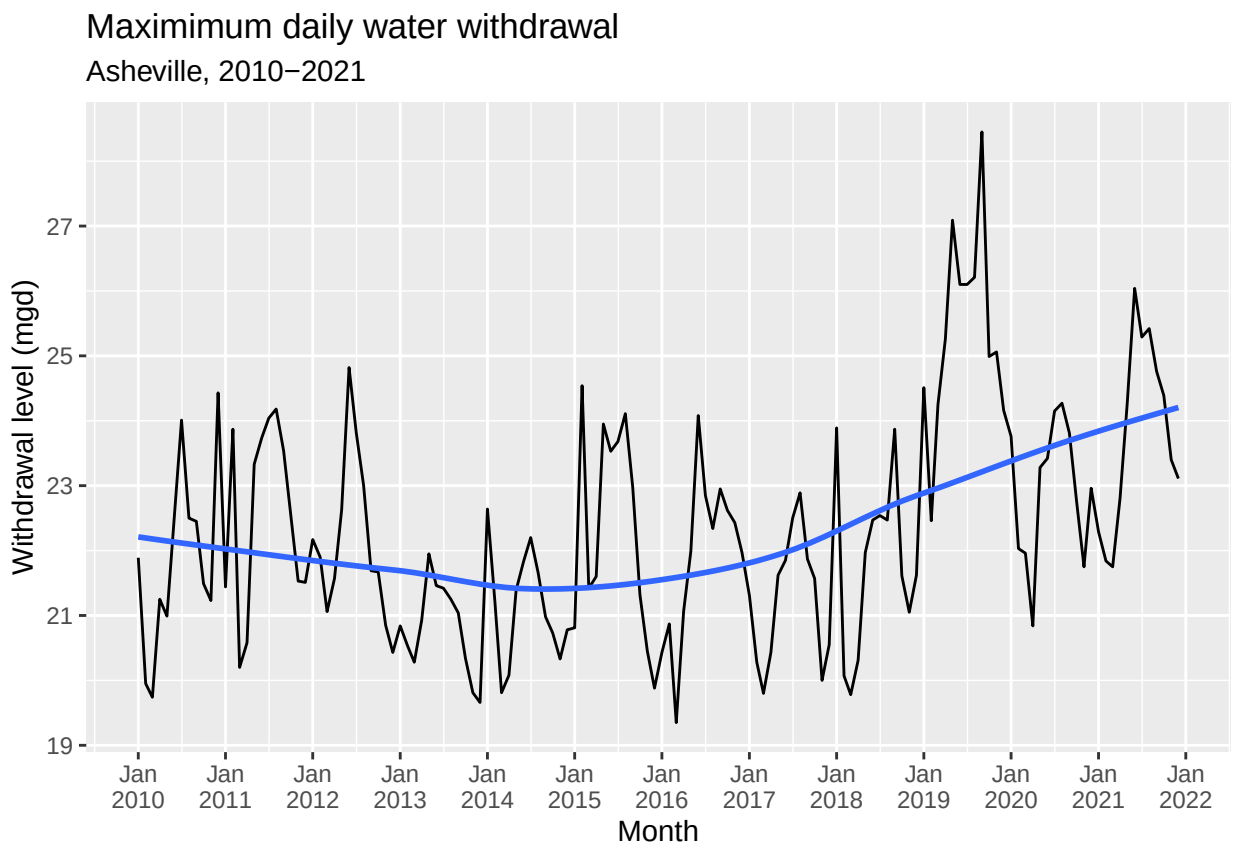
```
#9
# Create sequence of desired years for easy substitution
yearly_seq <- seq(2010,2021)
pwsid_for_seq <- rep('01-11-010', length(yearly_seq))

# "Map" the "scrape.it" function to retrieve data for all these
Seq_of_max_day_use <- map2(yearly_seq, pwsid_for_seq, scrape.it) %>%
  bind_rows() %>%
  arrange(Date)

# Plot the max day use across the sequence of years for Asheville
ggplot(Seq_of_max_day_use, aes(x = Date, y = Maximum.Day.Use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Maximimum daily water withdrawal"),
       subtitle = "Asheville, 2010-2021",
       y="Withdrawal level (mgd)",
       x="Month") +
```

```
  scale_x_date(limits = c(Seq_of_max_day_use$Date[1],
                          Seq_of_max_day_use$Date[144]),
               date_breaks = "1 year",
               date_labels = "%b\n%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: Based on the smoothed line, it seems Asheville does a have a trend in water usage over time–a slight decrease from 2010 to 2015, and a more visible increase from 2015 to 2022. More analysis would be needed to understand the role of the random component and seasonality.