Assignment 10: Data Scraping

Xiaoge Zhang

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)
library(lubridate)
library(rvest)

getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

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 <- 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), with the first value being "27.6400".

```
#3
water.system.name <- link %>% html_nodes('table:nth-child(7) tr:nth-child(1) td:nth-child(2)') %>%
  html_text()

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

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

max.withdrawals.mgd <- link %>% html_nodes(':nth-child(31) td:nth-child(9) , tr:nth-child(2) :nth-child
  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...

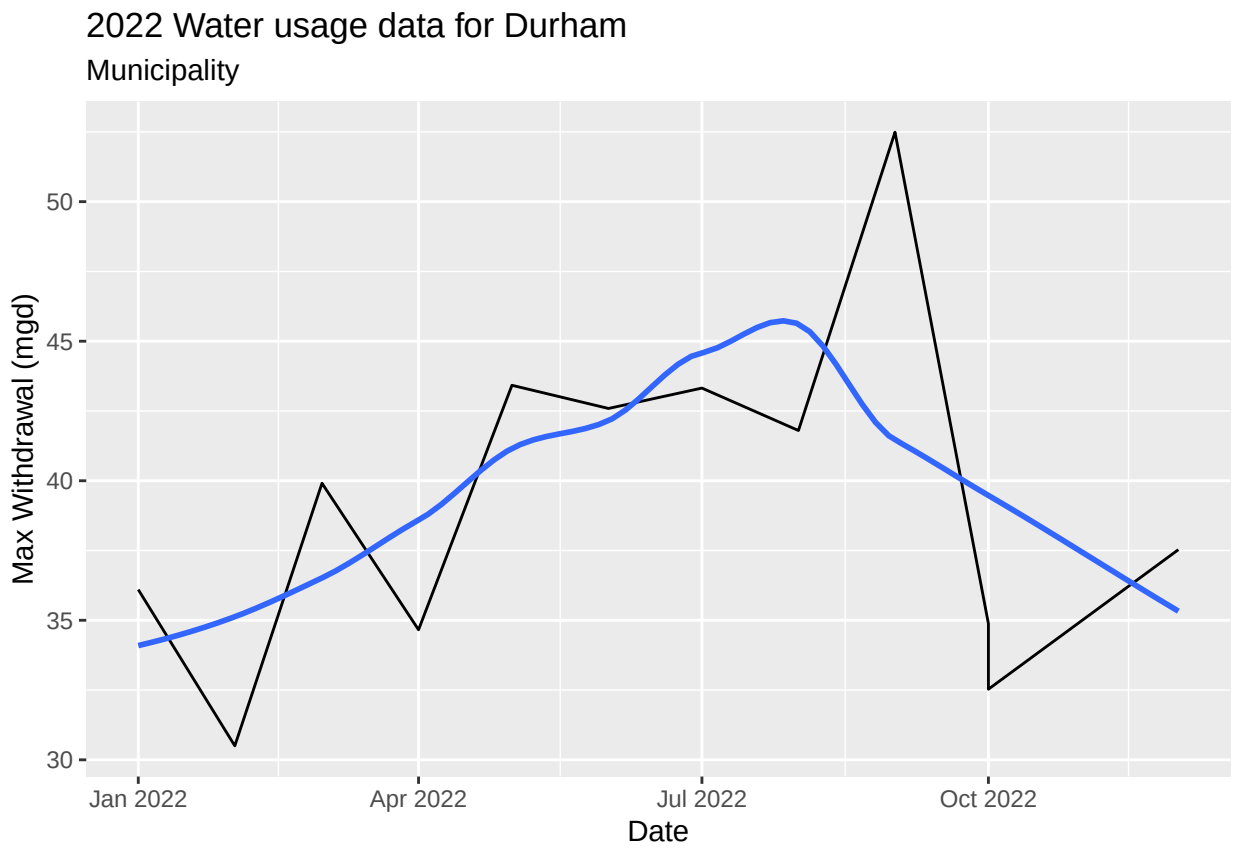
5. Create a line plot of the average daily withdrawals across the months for 2022

```r
#4
months <- c(1, 5, 9, 2, 6, 10, 3, 7, 10, 4, 8, 12)
df_withdrawals <- data.frame("Month" = months,
                             "Year" = rep(2022,12),
                             "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

#Modify the dataframe to include the facility name and type as well as the date (as date object)
df_withdrawals <- df_withdrawals %>%
  mutate(water_system_name = !!water.system.name,
         pwsid = !!PWSID,
         ownership = !!ownership,
         Date = my(paste(Month,"-",Year)))


#5
ggplot(df_withdrawals,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2022 Water usage data for", water.system.name),
       subtitle = ownership,
       y="Max Withdrawal (mgd)",
       x="Date")
```

```
## `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**.

```
#6.
scrape_a10 <- 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,

  water.system.name <- the_website %>%
    html_nodes('table:nth-child(7) 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('table:nth-child(7) tr:nth-child(2) td:nth-child(4)') %>%
    html_text()

  max_withdrawals_mgd <- the_website %>%
    html_nodes(':nth-child(31) td:nth-child(9) , tr:nth-child(2) :nth-child(9), :nth-child(31) td:nth-c
    html_text()

  #Convert to a dataframe
  df_withdrawals <- data.frame("Month" = months,
                               "Year" = rep(the_year,12),
                               "max_withdrawals_mgd" = as.numeric(max_withdrawals_mgd)) %>%
    mutate(water_system_name = !!water.system.name,
        pwsid = !!PWSID,
        ownership = !!ownership,
        Date = my(paste(Month,"-",Year)))

  #Pause for a moment - scraping etiquette
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

  #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

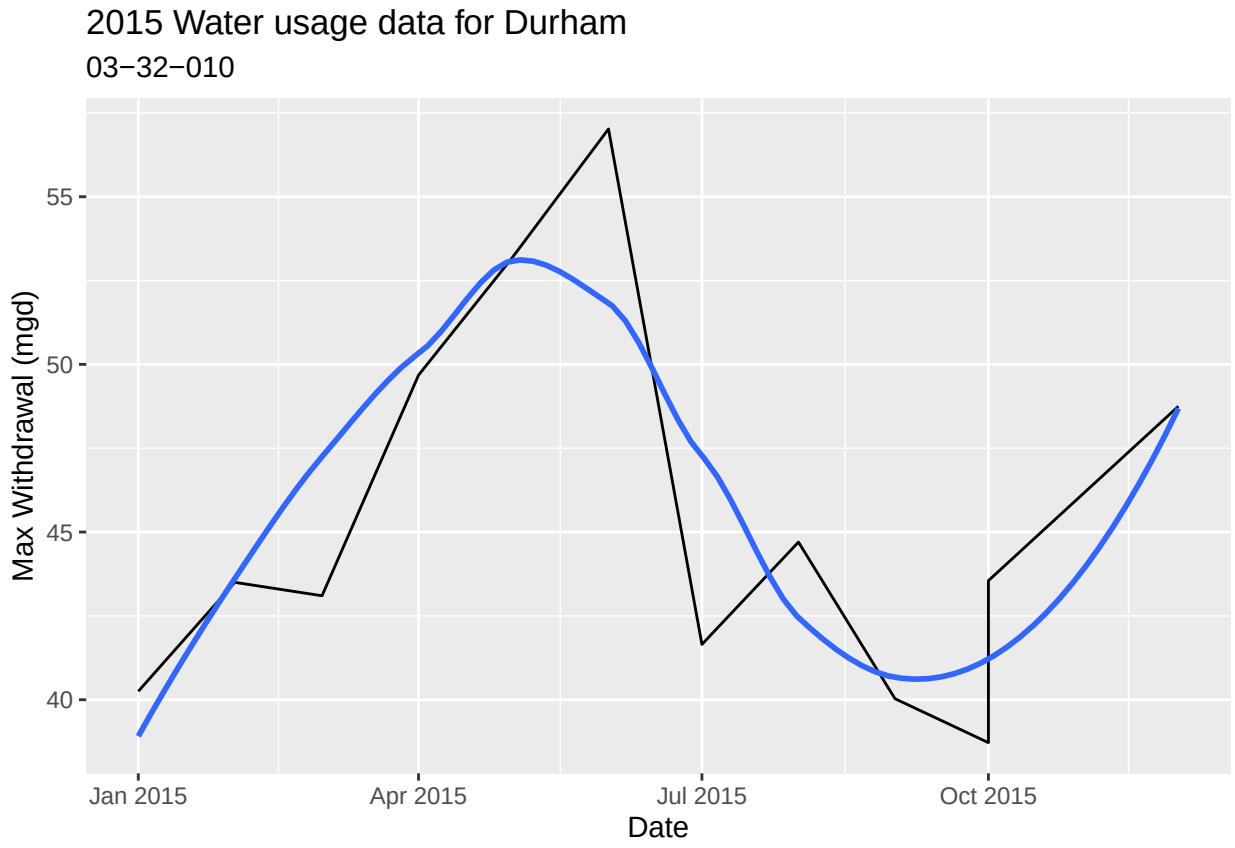
```
#7

df_2015 <- scrape_a10('2015', '03-32-010')

ggplot(df_2015,aes(x=Date,y=max_withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2015 Water usage data for Durham"),
```

```
       subtitle = '03-32-010',
       y="Max Withdrawal (mgd)",
       x="Date")
```

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



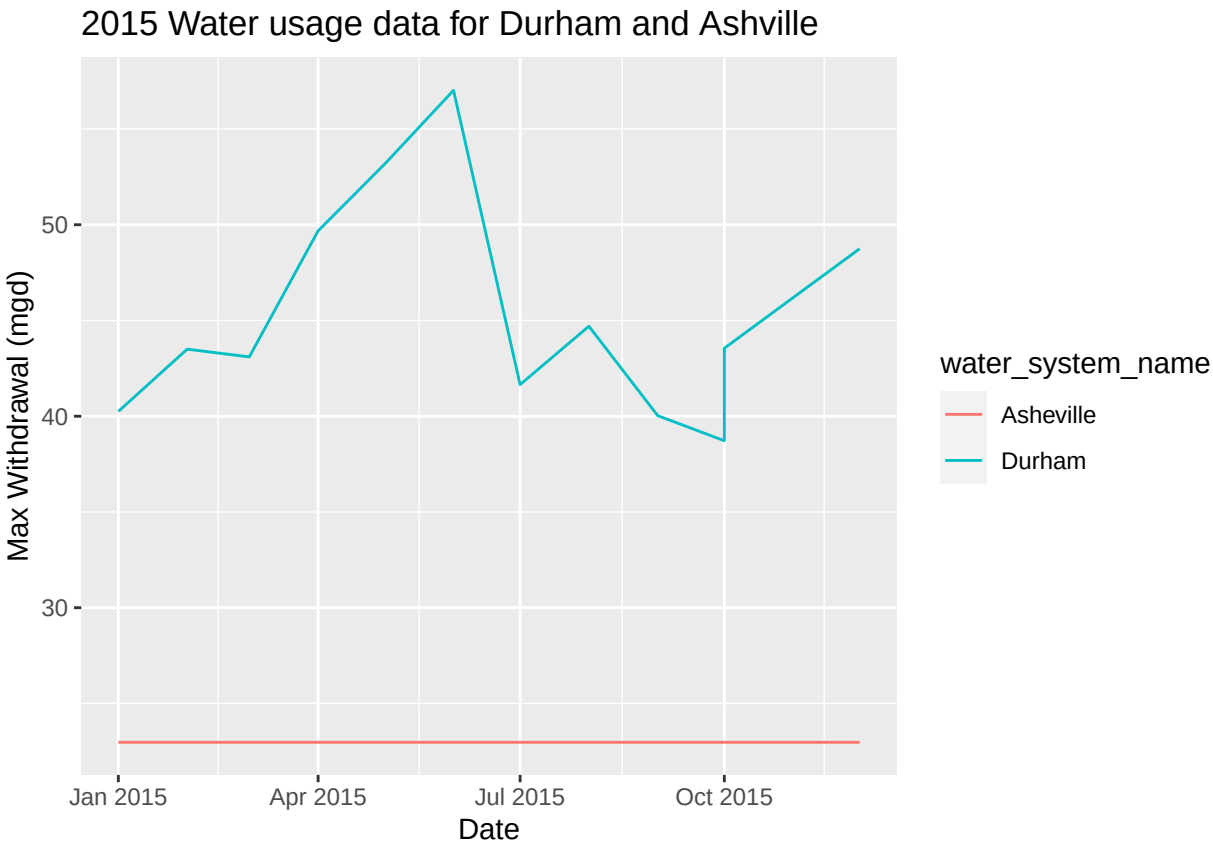
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8

df_2015_ash <- scrape_a10('2015', '01-11-010')

df_durham_ash <- rbind(df_2015, df_2015_ash)

ggplot(df_durham_ash, aes(x=Date,y=max_withdrawals_mgd, color=water_system_name)) +
  geom_line() +
  labs(title = paste("2015 Water usage data for Durham and Ashville"),
       y="Max Withdrawal (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 "09_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.

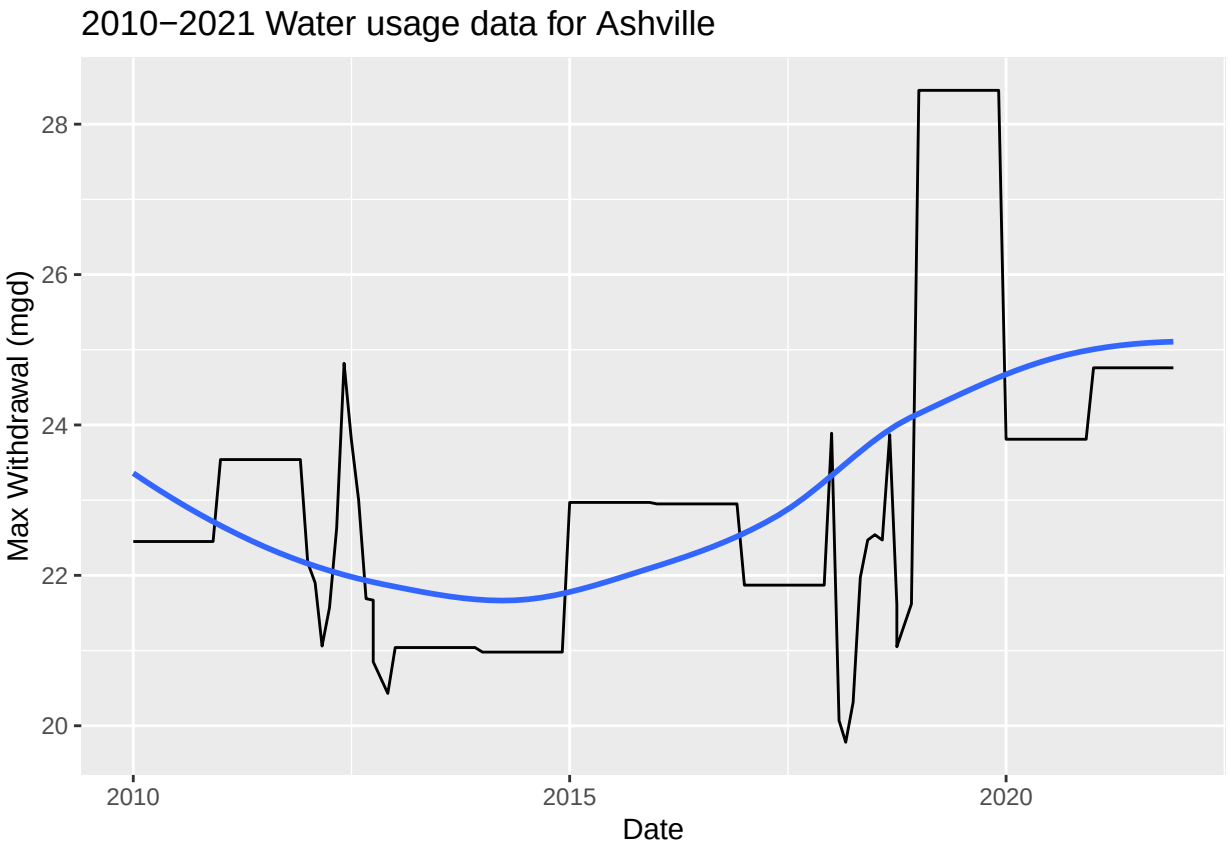
```
#9
years_ash <- rep(2010:2021)

ashville <- map(years_ash, scrape_a10, '01-11-010')

ashville <- bind_rows(ashville)

ggplot(ashville,aes(x=Date,y=max_withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2010-2021 Water usage data for Ashville"),
       y="Max Withdrawal (mgd)",
       x="Date")
```

```
## `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? — It looks like water usage decreased from 2010 to 2015 and then increased from 2015 to 2021.