Assignment 10: Data Scraping

Nadia Barbo

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 packages
library(tidyverse)
library(lubridate)
library(rvest)

#checking working directory
getwd()
```

```
## [1] "C:/Users/nadia/Documents/Duke_/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 URL to be scraped. (In other words, read the contents into an `rvest` webpage object.)

```
#2
webpage <-
↪  read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
↪  #reading webpage into variable
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), with the first value being "27.6400".

```
#3
#getting data for water system name using element tag
water.system.name <- webpage %>%
  html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
water.system.name
```

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

```
#getting data for PWSID using element tag
PWSID <- webpage %>%
  html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
  html_text()
PWSID
```

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

```
#getting data for ownership using element tag
ownership <- webpage %>%
  html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
ownership
```

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

```
#getting data for max daily use using element tag
max.withdrawals.mgd <- webpage %>%
  html_nodes('th~ td+ td') %>%
  html_text()
max.withdrawals.mgd
```

```
##  [1] "36.1000" "43.4200" "52.4900" "30.5000" "42.5900" "34.8800" "39.9100"
##  [8] "43.3200" "32.5300" "34.6600" "41.8000" "37.5300"
```

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.)

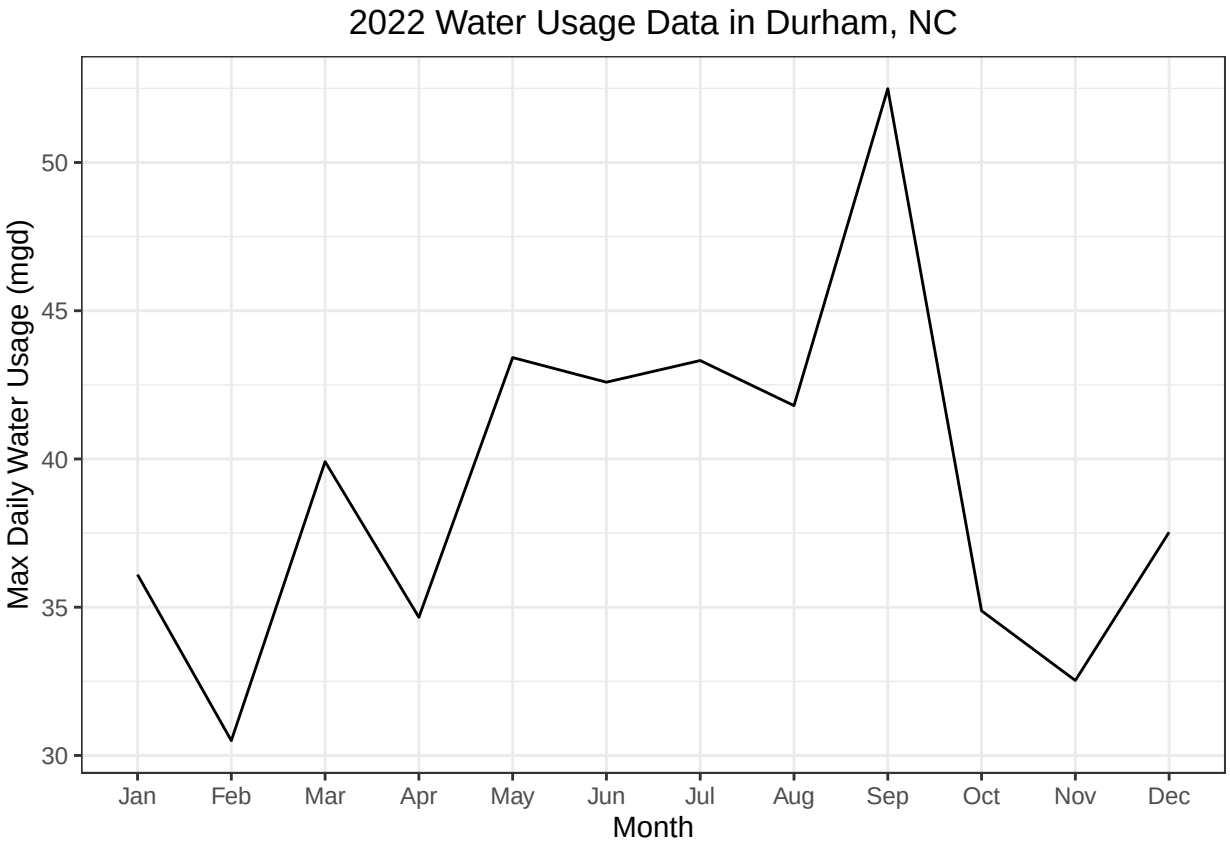
   TIP: Use `rep()` to repeat a value when creating a dataframe.

   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

```
#4
#making dataframe of month, year, and max withdrawal
df_max_withdrawals <- data.frame("Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12),
                                 "Year" = rep('2022'),
                                 "Max.Withdrawals.mgd" = as.numeric(max.withdrawals.mgd))

#adding other variables we extracted to dataframe and adding Date column
df_max_withdrawals <- df_max_withdrawals %>%
  mutate(Water.System.Name = !!water.system.name,
         PWSID = !!PWSID,
         ownership = !!ownership,
         Date = my(paste(Month,"-",Year)))

#5
#plotting max withdrawals by date
ggplot(df_max_withdrawals, aes(x=factor(Month, levels = 1:12, labels = month.abb),
↪  y=Max.Withdrawals.mgd, group=1)) +
  geom_line() +
  #geom_smooth(method="loess",se=FALSE) +
```

```
  labs(title = paste("2022 Water Usage Data in Durham, NC"),
       y="Max Daily Water Usage (mgd)",
       x="Month") +
  theme_bw()+
  theme(plot.title = element_text(hjust=0.5))
```



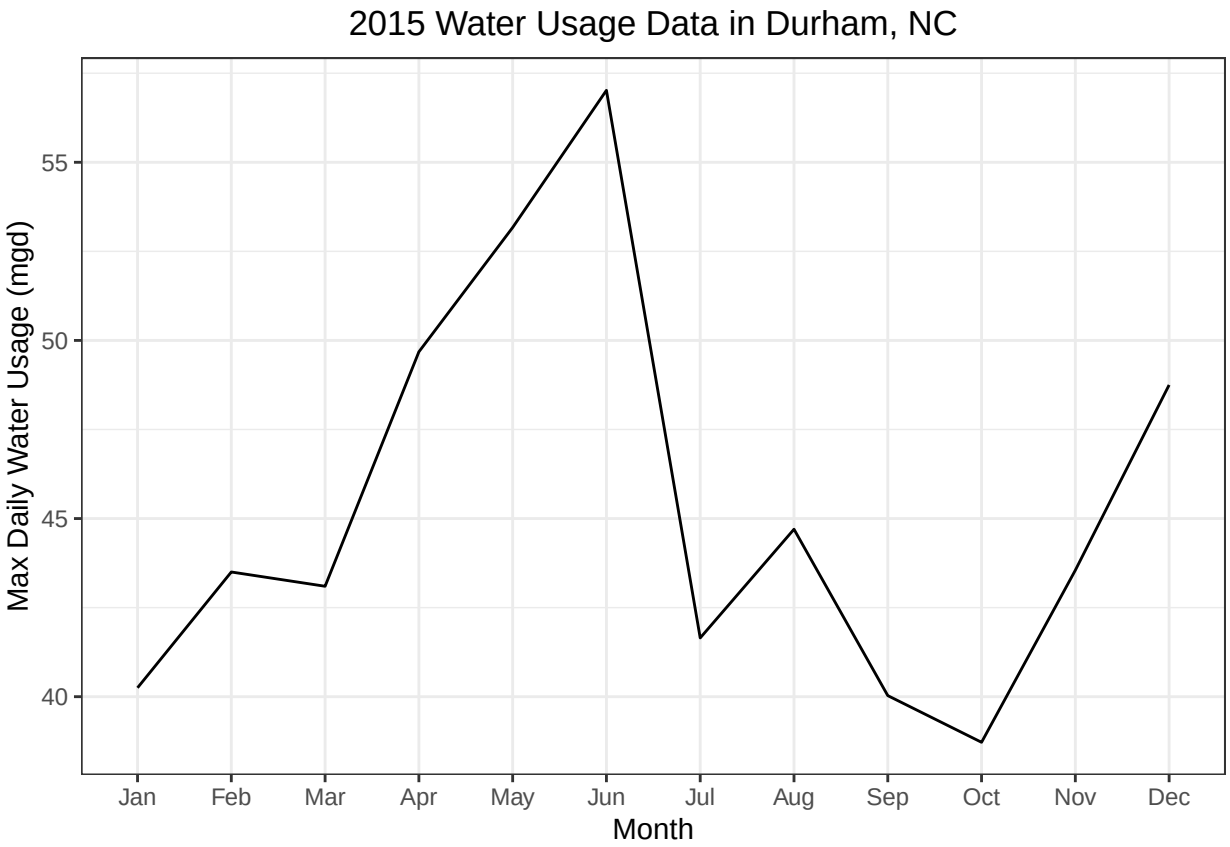
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
#Create our scraping function
scrape.it <- function(the_year, the_pwsid){

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

  #Scrape the data items
  water.system.name <- function_website %>%
    html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
    html_text()
```

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

  ownership <- function_website %>%
    html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
    html_text()

  max.withdrawals.mgd <- function_website %>%
    html_nodes('th~ td+ td') %>%
    html_text()

  #Convert to a dataframe
  df_withdrawals_fromfunction <- data.frame("Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4,
↪  8, 12),
                                "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)))

  #Return the dataframe
  return(df_withdrawals_fromfunction)
}
```

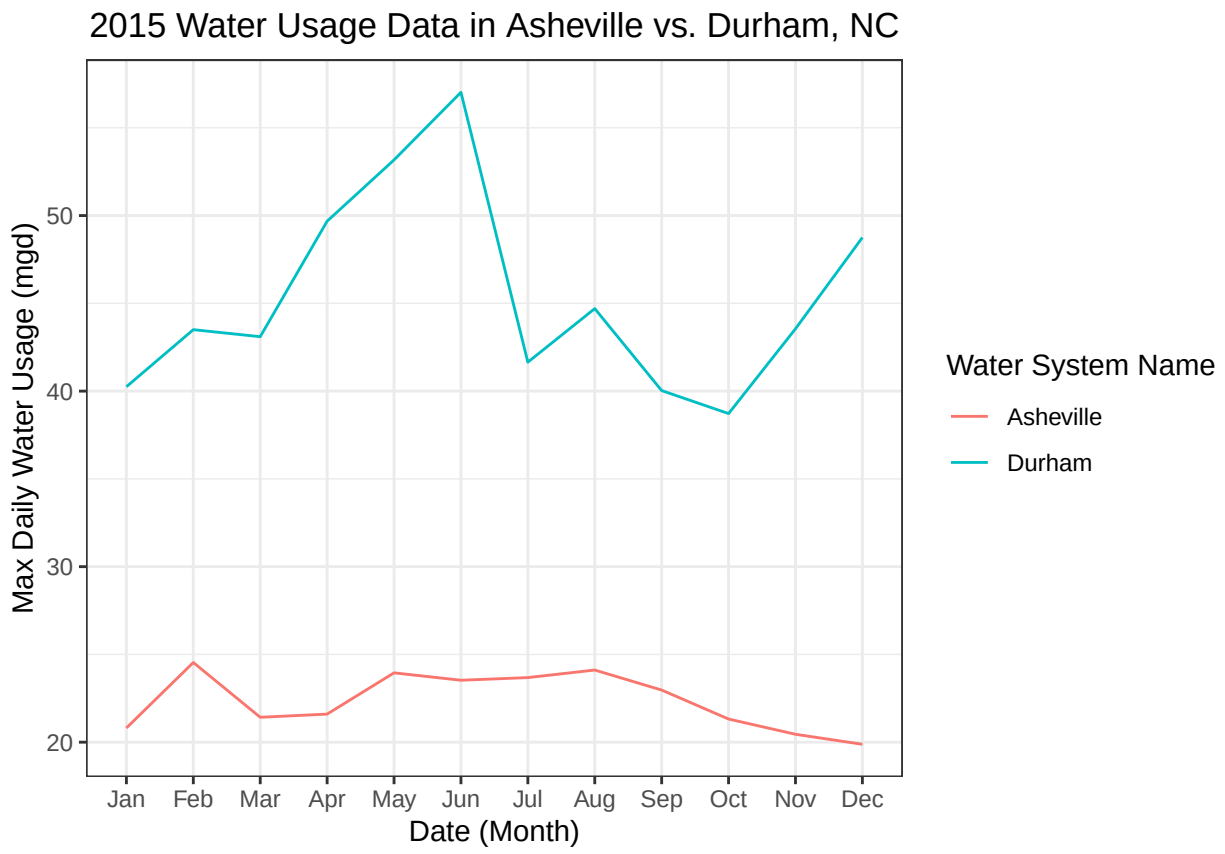
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
#using function above to get Durham data from 2015
durham_2015_withdrawals_df <- scrape.it(2015,'03-32-010')

#plotting max withdrawals by month
ggplot(durham_2015_withdrawals_df, aes(x=factor(Month, levels = 1:12, labels =
↪  month.abb), y=Max.Withdrawals.mgd, group=1)) +
  geom_line() +
  labs(title = paste("2015 Water Usage Data in Durham, NC"),
       y="Max Daily Water Usage (mgd)",
       x="Month") +
  theme_bw()+
  theme(plot.title = element_text(hjust=0.5))
```



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
#using function created to get Asheville data from 2015
asheville_2015_withdrawals_df <- scrape.it(2015,'01-11-010')

#binding asheville and durham dataframes
ashevilledurham_2015_withdrawals_df<- rbind(durham_2015_withdrawals_df,
↪  asheville_2015_withdrawals_df)

#plot Asheville and Durham max withdrawals by date
ggplot(ashevilledurham_2015_withdrawals_df, aes(x=factor(Month, levels = 1:12, labels =
↪  month.abb), y=Max.Withdrawals.mgd, colour=Water.System.Name,
↪  group=Water.System.Name)) +
  geom_line() +
  labs(title = "2015 Water Usage Data in Asheville vs. Durham, NC",
       y="Max Daily Water Usage (mgd)",
       x="Date (Month)",
       colour="Water System Name") +
  theme_bw()+
  theme(plot.title = element_text(hjust=0.5))
```



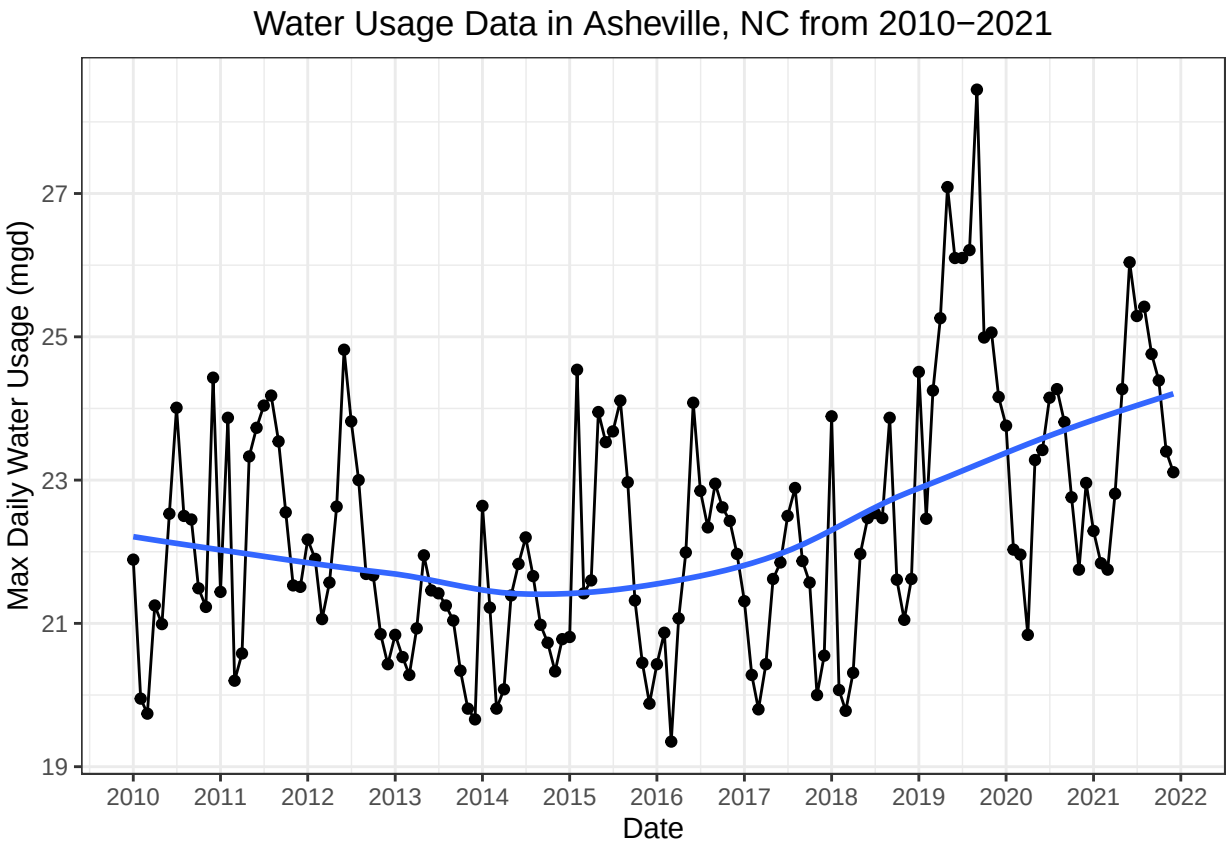
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021. Add a smoothed line to the plot (method = 'loess').

   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
#variable with years we want
asheville_years <- c(2010:2021)

#variable of Asheville PWSID as long as years variable
asheville_pwsid <- rep.int('01-11-010',length(asheville_years))

#map2 used to create dataframes for all years
asheville_range_withdrawals_dfs <- map2(asheville_years, asheville_pwsid, scrape.it)

#combining all the dataframes
asheville_range_withdrawals_df <- bind_rows(asheville_range_withdrawals_dfs)

#plotting Asheville water use data from 2010-2021 and adding smoothed line
ggplot(asheville_range_withdrawals_df,aes(y = Max.Withdrawals.mgd, x=Date)) +
  geom_line() +
  geom_point() +
  geom_smooth(method="loess",se=FALSE) +
```

```
  labs(title = "Water Usage Data in Asheville, NC from 2010-2021",
       y="Max Daily Water Usage (mgd)",
       x="Date") +
  theme_bw()+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(plot.title = element_text(hjust=0.5))
```

```
## `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 appears that water usage in Asheville dipped slightly between 2010-2015 and then began to increase in 2015. This increase has continued since through 2021, with the peak water usage being in 2021.