# Assignment 10: Data Scraping

Nusrat Noor

## 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
library(tidyverse)
library(lubridate)
library(here)
library(rvest)
#loaded in the packages

here() #checked the directory
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```r
mytheme <- theme_gray() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(mytheme)
#created and set theme
```

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.)

```r
#2
website <- read_html(
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
#set scraping website
website
```

```
## {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= ...
```

```r
#checked website
```

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
water_sys <- website %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
#scraped the water system name data
water_sys #checked the variable
```

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

```r
PWSID <- website %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
#scraped the pwsid data
PWSID #checked the variable
```

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

```r
Ownership <- website %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
#scraped the ownership data
Ownership #checked the variable
```

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

```r
MGD_monthly <- website %>%
  html_nodes("th~ td+ td") %>%
  html_text()
#scraped the mgd data
MGD_monthly #checked the variable
```

```
##  [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…

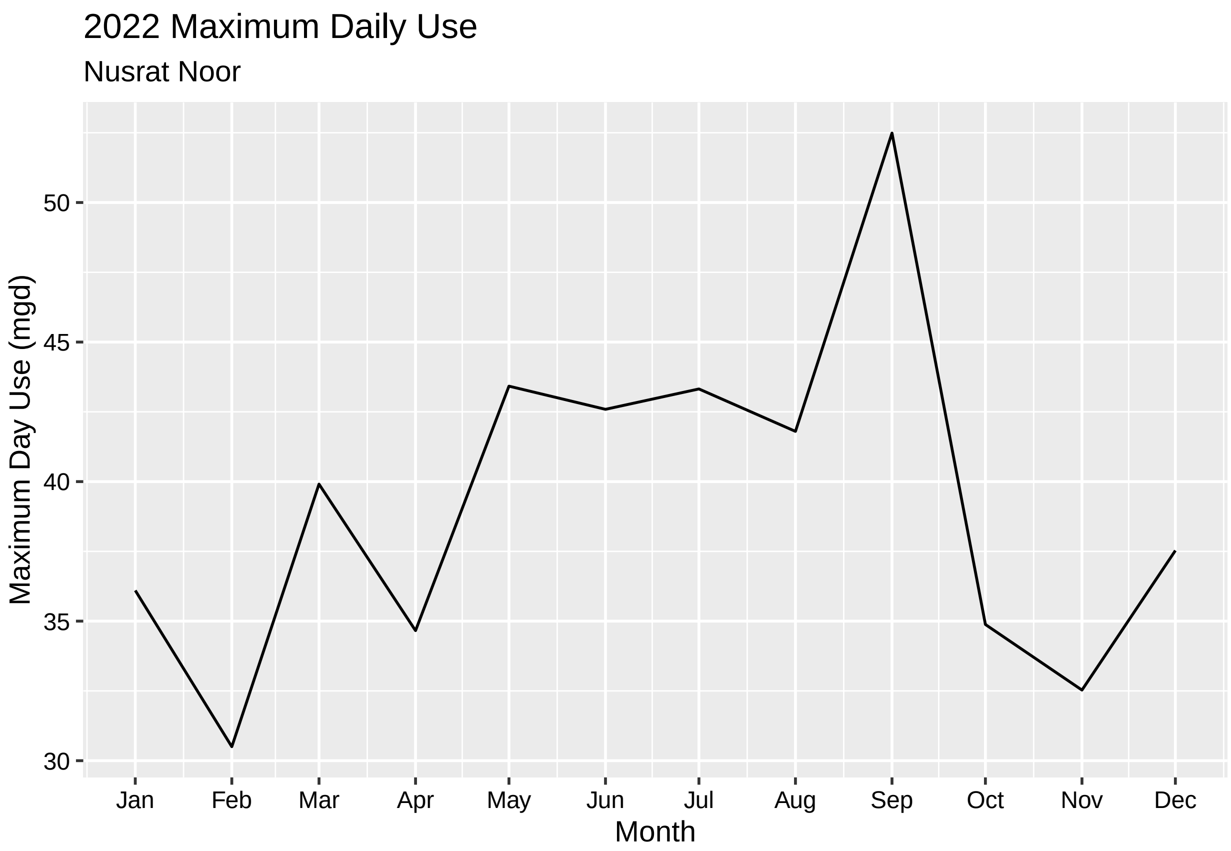
5. Create a line plot of the maximum daily withdrawals across the months for 2022

```
#4
MGD_df <- data.frame(
  #created dataframe
  "Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12),
  #created month column and column bound the months to match the way the data was scraped
                     "Year" = rep(2022,12),
  #created the Year column and set it to repeat 12 times to match the number of rows
                     "Max_withdrawls" = as.numeric(MGD_monthly))
#created the max_withdrawls column and set as numeric

MGD_df <- MGD_df %>%
  mutate(Water_system = !!water_sys,
         PWSID = !!PWSID,
         Ownership = !!Ownership,
         Date = my(paste0(Month,"-",Year)))
#added the other variables as columns to dataframe and created a Date column

#5

ggplot(MGD_df,aes(x=Date, y = Max_withdrawls)) +
  geom_line() +
  labs(y="Maximum Day Use (mgd)", x="Month", title = "2022 Maximum Daily Use",
       subtitle = "Nusrat Noor") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
#created line plot, cleaned labels, and scaled the x-axis to show the months in order
```

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**.

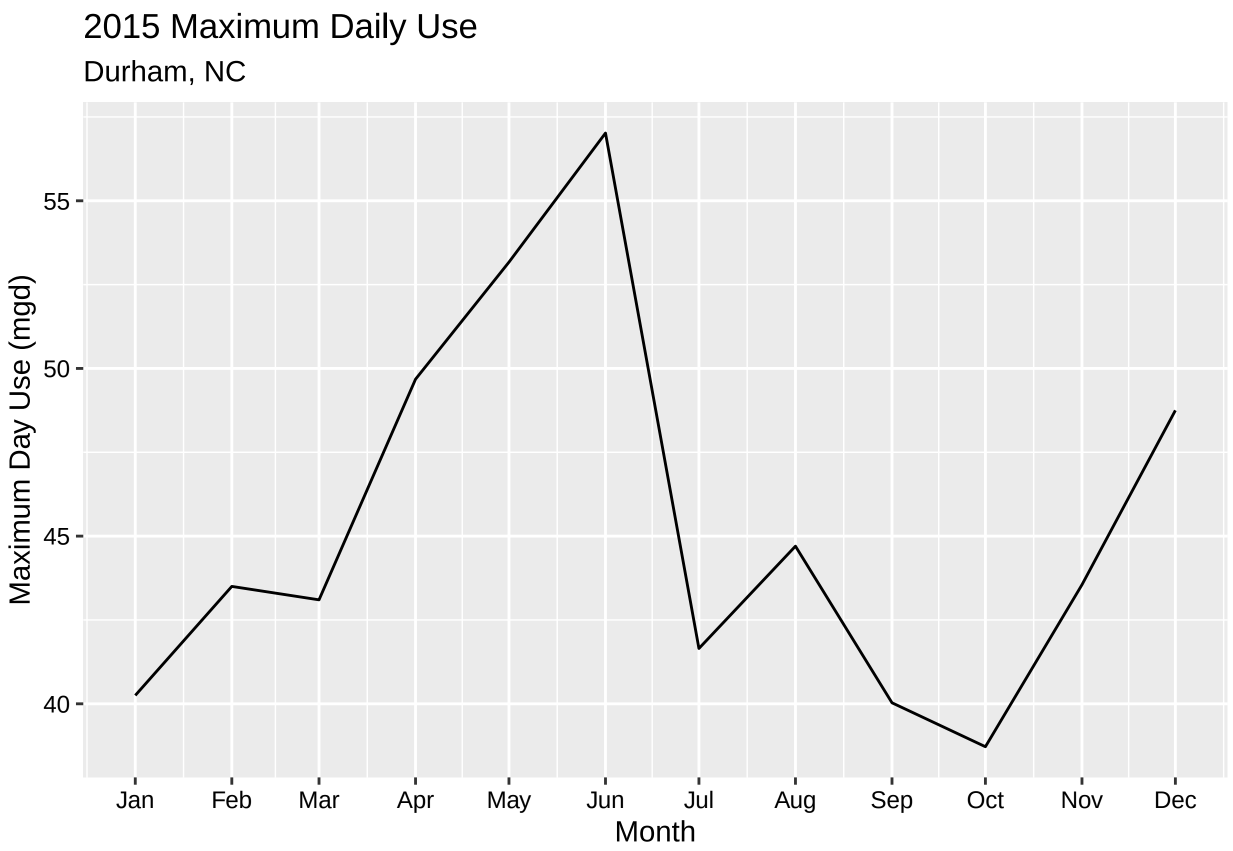
```
#6.
base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_pwsid <- '03-32-010'
the_year <- 2015
#created the website url to scrape from

scrape.it <-function(the_pwsid, the_year){
  #created scrape.it function
  the_website <- read_html(paste0(base_url, 'pwsid=', the_pwsid, '&year=', the_year))
  #retrieved the website contentts
  water_sys_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  PWSID_tag <- 'td tr:nth-child(1) td:nth-child(5)'
  ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  mgd_monthly_tag <- 'th~ td+ td'
  #set element address variables

  water_sys <- the_website %>%
    html_nodes(water_sys_tag) %>%
    html_text()
  PWSID <- the_website %>%
    html_nodes(PWSID_tag) %>%
    html_text()
  Ownership <- the_website %>%
    html_nodes(ownership_tag) %>%
    html_text()
  MGD_monthly <- the_website %>%
    html_nodes(mgd_monthly_tag) %>%
    html_text()
  #scraped the data
  Durham_df <- data.frame("Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12),
                          "Year" = rep(the_year,length(months)),
                          "Max_withdrawls" = as.numeric(MGD_monthly)) %>%
    mutate(Water_system = !!water_sys,
         PWSID = !!PWSID,
         Ownership = !!Ownership,
         Date = my(paste(Month,"-",Year)))
  #converted to dataframe
  return(Durham_df)
  #returned the function
}
```

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

```
#7
the_df <- scrape.it('03-32-010', 2015)
#ran the function for Durham 2015
ggplot(the_df, aes(x=Date, y=Max_withdrawls)) +
  geom_line() +
  labs(y="Maximum Day Use (mgd)", x="Month", title = "2015 Maximum Daily Use",
       subtitle = "Durham, NC") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
#plotted the 2015 max daily usage for durham
```

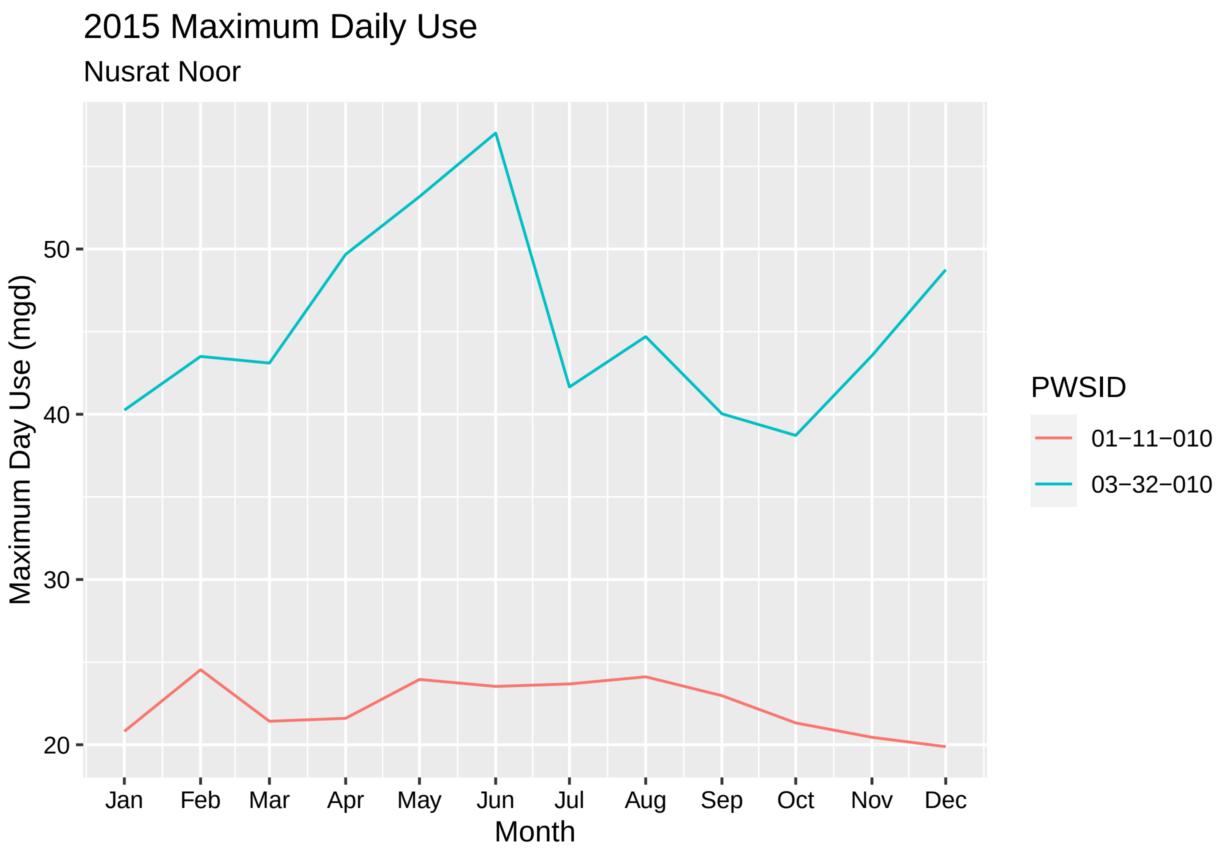
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8
Ash_df <- scrape.it('01-11-010', 2015)
#ran the function for Asheville 2015
combined_df <- full_join(x=the_df, y =Ash_df)
```

```
## Joining with `by = join_by(Month, Year, Max_withdrawls, Water_system, PWSID,
## Ownership, Date)`
```

```
#combined the asheville and durham data frames

ggplot(combined_df, aes(x=Date, y=Max_withdrawls, color = PWSID)) +
  geom_line() +
  labs(y="Maximum Day Use (mgd)", x="Month", title = "2015 Maximum Daily Use",
       subtitle = "Nusrat Noor") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



```
#plotted the combined dataframe
```

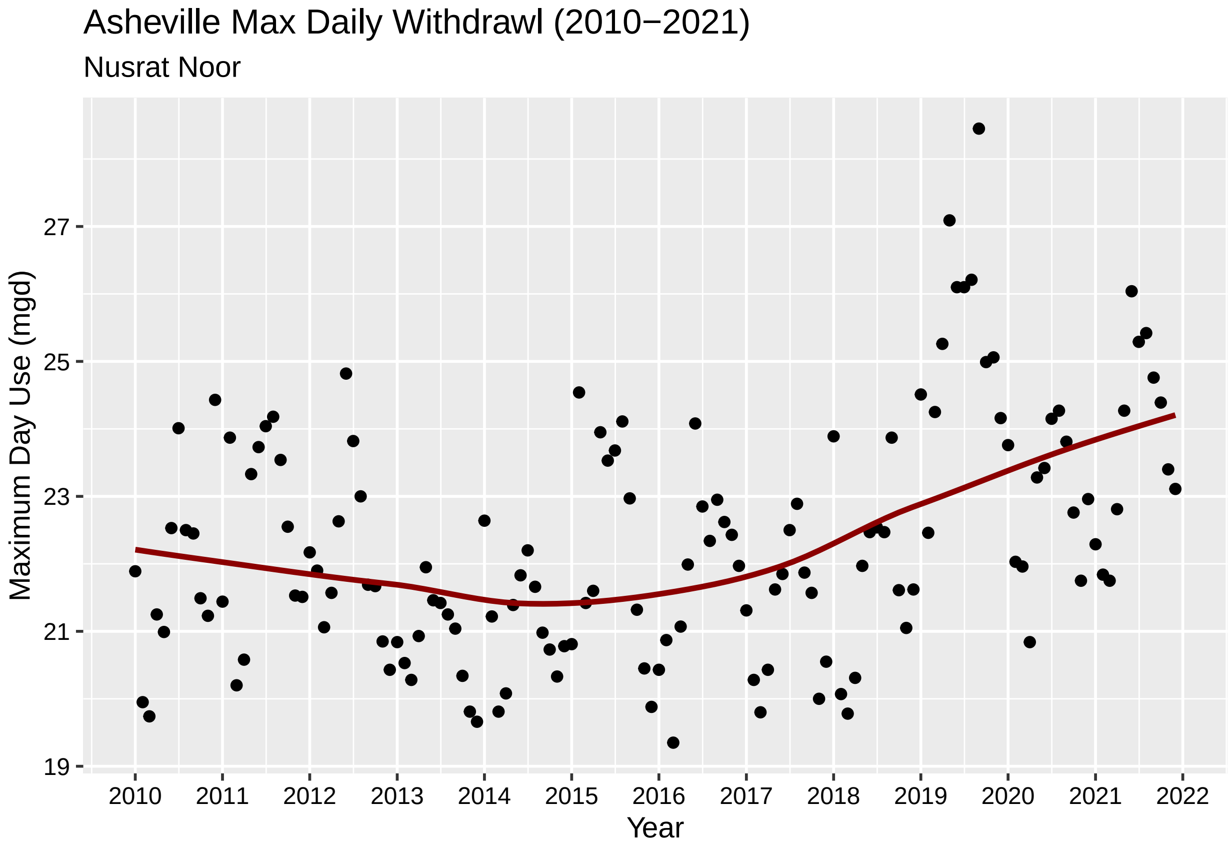
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   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
the_years <- 2010:2021 #set the years wanted
the_pwsid_ash <- '01-11-010' #set the pwsid for asheville

dfs_Ashville <- map2(the_pwsid_ash, the_years, scrape.it) %>%
  bind_rows()
#mapped the scrape.it function to give data for all years and combined to one dataframe
ggplot(dfs_Ashville, aes(x=Date, y =Max_withdrawls)) +
  geom_point() +
  geom_smooth(method = "loess", se=FALSE, color = "darkred") +
  labs(x="Year", y= "Maximum Day Use (mgd)",
       title = "Asheville Max Daily Withdrawl (2010-2021)",
       subtitle = "Nusrat Noor") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")
```

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



```
#plotted the multiyear data
```

Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time? > Answer: The plot shows that the max daily use has increased over time, especially since 2015. >