Assignment 10: Data Scraping

Laura Brockington

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(rvest); library(here); library(lubridate) #loading packages

here() #checking working directory
```

```
## [1] "/Users/laura/Desktop/EDA/EDA"
```

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
the_URL <-
  read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022")
the_URL
```

```
## {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
#scraping the data we want into 4 separate variables
water.system.name <- the_URL %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water.system.name
```

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

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

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

```
ownership <- the_URL %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
ownership
```

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

```
max.withdrawals.mgd <- the_URL %>%
  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"
```

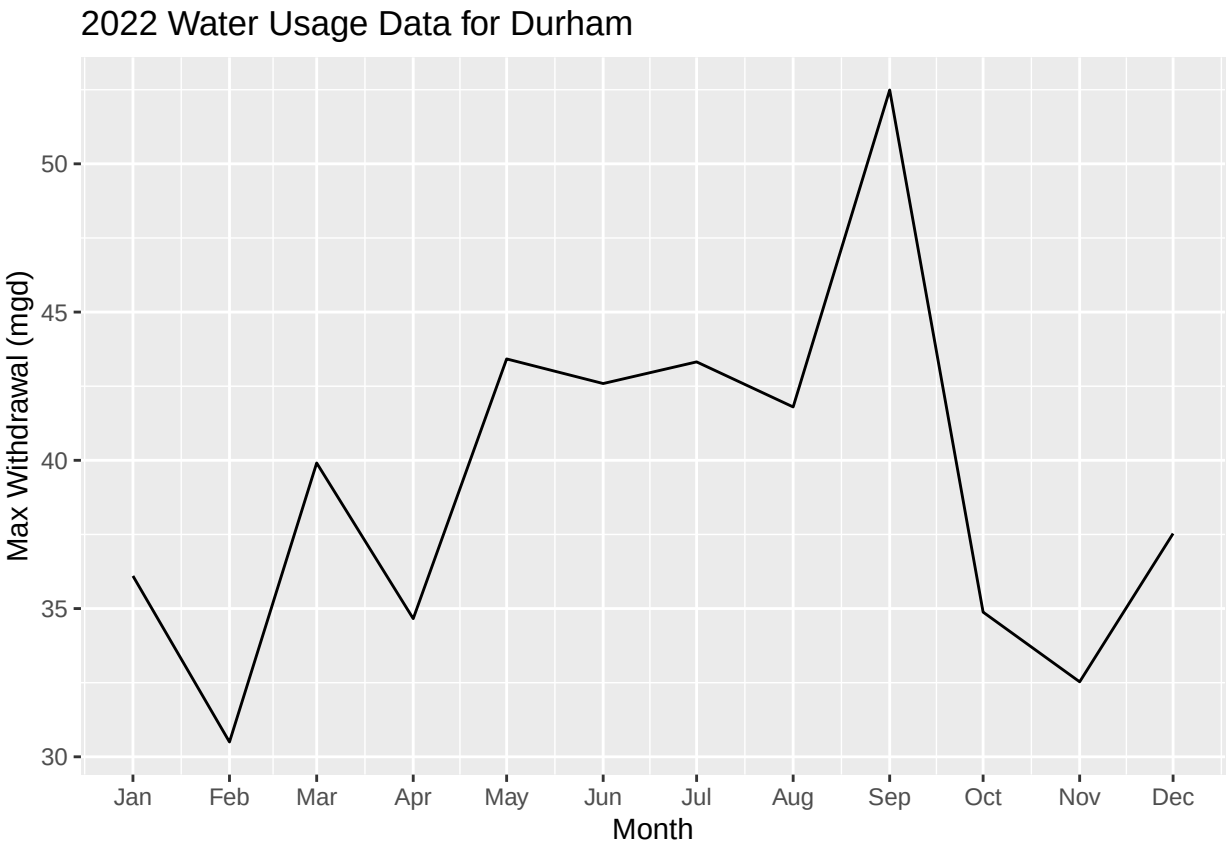
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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
month <- the_URL %>% #scraping month values from website
  html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
  html_text()

df <- data.frame("Month" = match(month, month.abb), #creating dataframe of all variables
                 "Year" = rep(2022),
                 "Water_System" = rep(water.system.name),
                 "PWSID" = rep(PWSID),
                 "Ownership" = rep(ownership),
                 "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

df$Month <- as.numeric(df$Month) #change Month to be numeric
df$Date <- my(paste(df$Month, "-", df$Year)) #create Date column of Month and Year

df <- df[order(df$Date),] #reorder rows by Date
df <- df[, c(7,1,2,3,4,5,6)] #reorder columns to put Date first

view(df)

#5
ggplot(df, #plotting max withdrawal by date for Durham in 2022
       aes(x = Date,
           y = Max_Withdrawals_mgd)) +
  geom_line() +
  labs(title = paste("2022 Water Usage Data for Durham"),
       y="Max Withdrawal (mgd)",
       x="Month") +
  scale_x_date(breaks = df$Date,
               labels = month.abb)
```



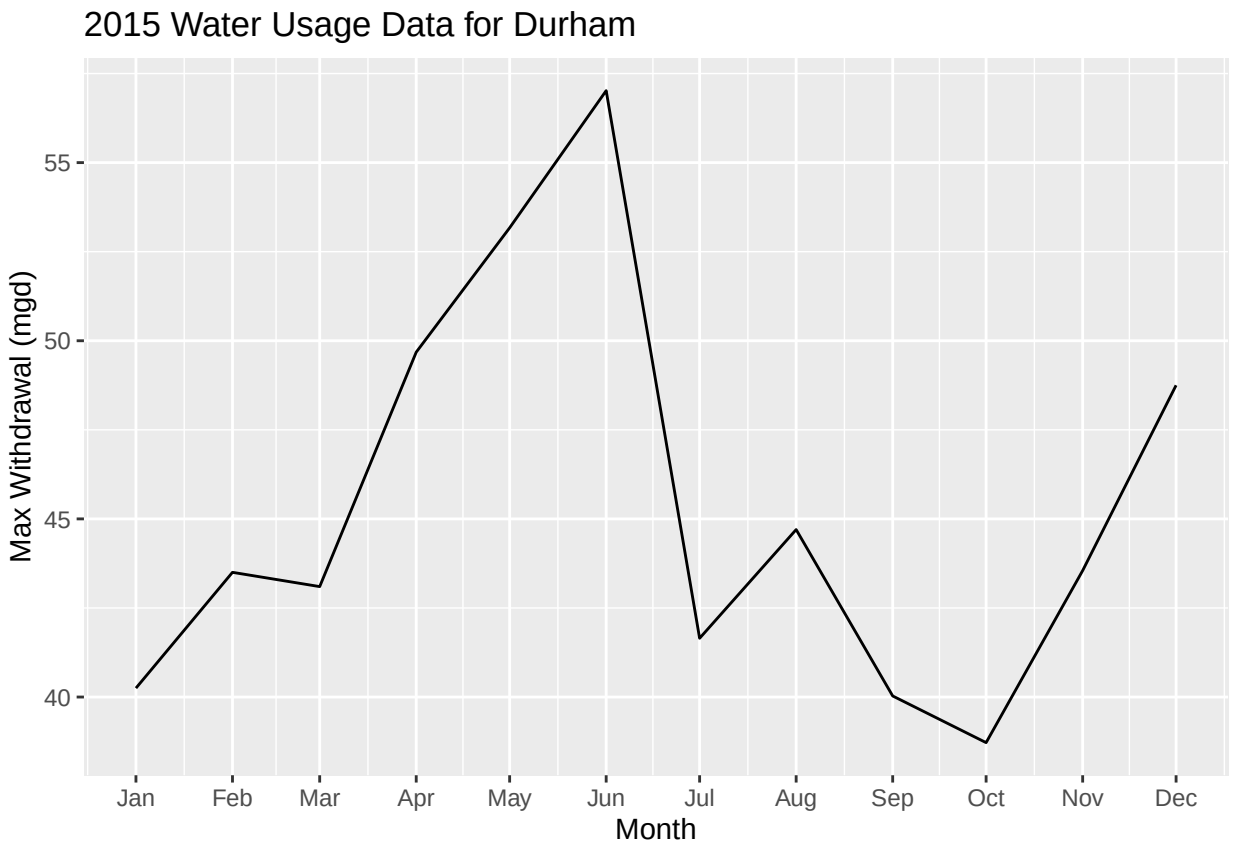
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
scrape.func <- function(Year, PWSID){
  #Retrieving website contents
  website <- read_html(paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=",
                              PWSID,
                              "&year=",
                              Year))
  #Setting element address variables from #4
  water.system.name.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)"
  max.withdrawals.mgd.tag <- "th~ td+ td"
  #Scraping the data items
  water.system.name <- website %>% html_nodes(water.system.name.tag) %>% html_text()
  PWSID <- website %>%   html_nodes(PWSID.tag) %>%  html_text()
  ownership <- website %>% html_nodes(ownership.tag) %>% html_text()
  max.withdrawals.mgd <- website %>% html_nodes(max.withdrawals.mgd.tag) %>% html_text()
  #Converting to a dataframe
  df <- data.frame("Month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                   "Year" = rep(Year, 12),
                   "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd)) %>%
    mutate(Water_System = rep(water.system.name),
```

```
            PWSID = rep(PWSID),
            Ownership = rep(ownership),
            Date = my(paste(Month,"-",Year))) %>%
    arrange(Date)
  #Returning the dataframe
  return(df)
}
```

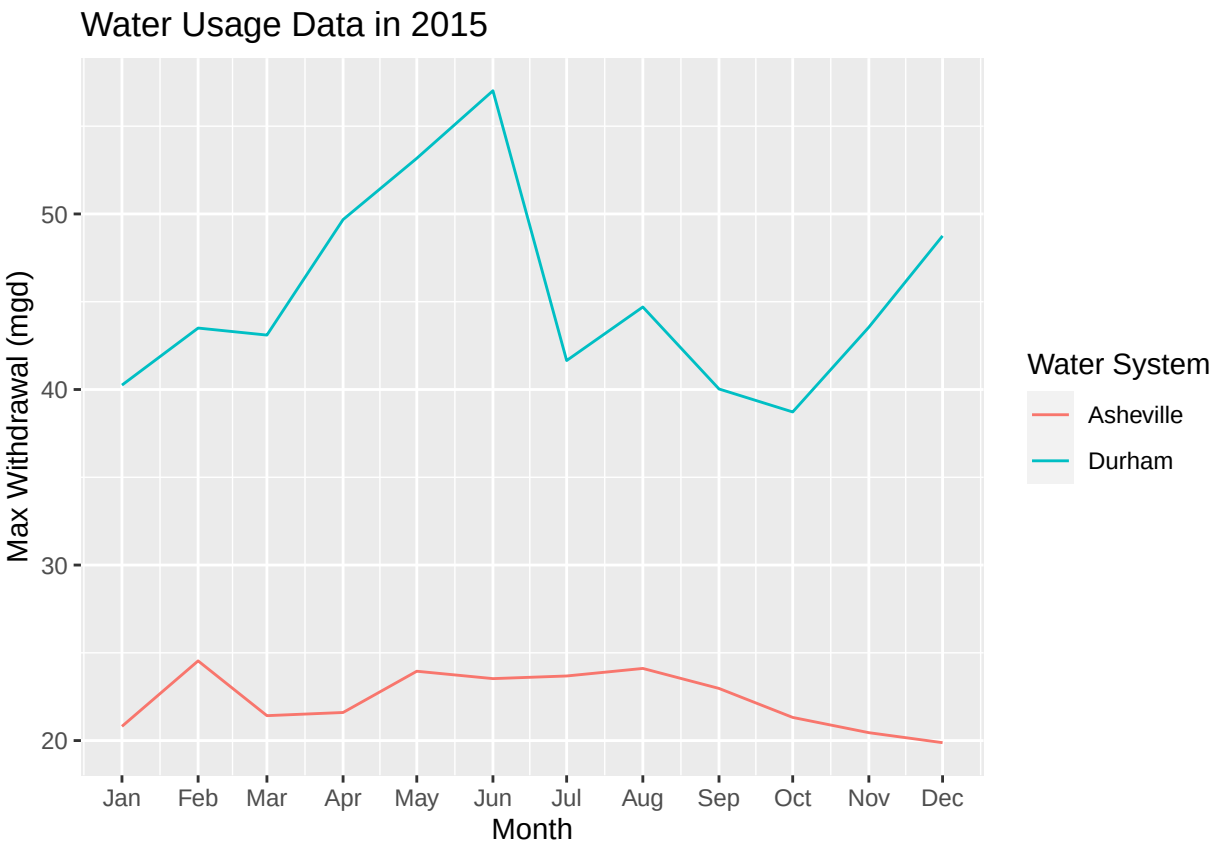
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
#using my function to get max daily withdrawals for Durham in 2015
Durham_2015 <- scrape.func(2015,'03-32-010')
view(Durham_2015)

ggplot(Durham_2015, #plotting max withdrawal by date for Durham in 2015
       aes(x = Date,
           y = Max_Withdrawals_mgd)) +
  geom_line() +
  labs(title = paste("2015 Water Usage Data for Durham"),
       y="Max Withdrawal (mgd)",
       x="Month") +
  scale_x_date(breaks = Durham_2015$Date,
               labels = month.abb)
```



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 my function to get max daily withdrawals for Asheville in 2015
Asheville_2015 <- scrape.func(2015,'01-11-010')
view(Asheville_2015)

Dur_Ash_2015 <- rbind(Durham_2015, Asheville_2015) #combining both dataframes

ggplot(Dur_Ash_2015, #plotting Durham and Asheville water usage in 2015
       aes(x = Date,
           y = Max_Withdrawals_mgd,
           color = Water_System)) +
  geom_line() +
  labs(title = "Water Usage Data in 2015",
       y="Max Withdrawal (mgd)",
       x="Month",
       color = "Water System") +
  scale_x_date(breaks = Durham_2015$Date,
               labels = month.abb)
```

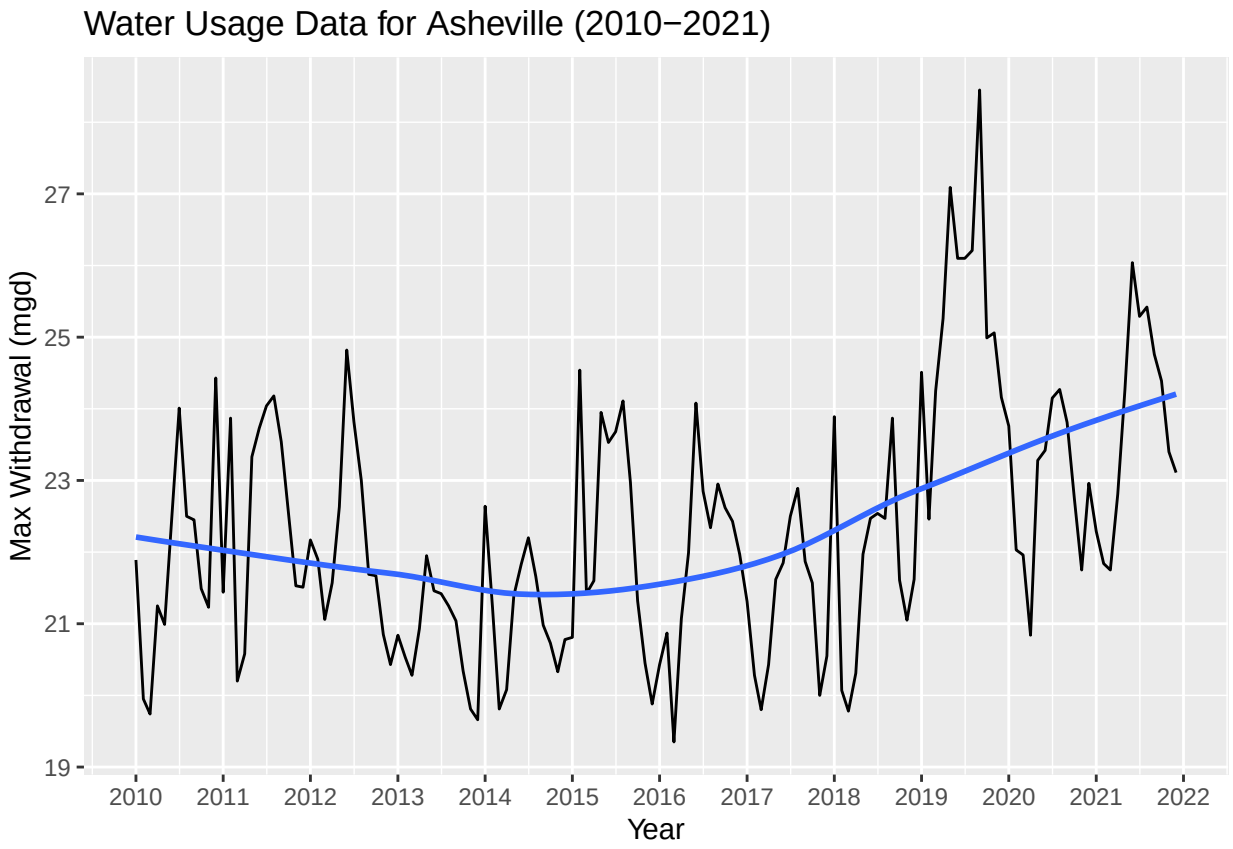


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

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
#creating dataframe of Asheville water usage from 2010 through 2021
Years = c(2010:2021)
Ash_2010_2021 <-  map2(Years,
                       "01-11-010",
                       scrape.func) %>%
  bind_rows()
view(Ash_2010_2021)

ggplot(Ash_2010_2021, #plotting the Asheville data from 2010-2021
       aes(x = Date,
           y = Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method = "loess",
              se = FALSE) +
  labs(title = paste("Water Usage Data for Asheville (2010-2021)"),
       y="Max Withdrawal (mgd)",
       x="Year") +
  scale_x_date(breaks = "years",
               date_labels = "%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? Yes. By visually analyzing the above plot, Asheville has increased it's water usage over time.