Assignment 09: Data Scraping

Sam Saltman

Total points:

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_09_Data_Scraping.Rmd") prior to submission.

Set up

1. Set up your session:

- Check your working directory
- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Set your ggplot theme

```
#1
getwd()
```

```
## [1] "/Users/samsaltman/Documents/R/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
library(lubridate)
library(viridis)
library(rvest)
library(dataRetrieval)
library(tidycensus)

# Set theme
mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2019 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Change the date from 2020 to 2019 in the upper right corner.
- 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=2020

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

```
#2
the_website_2020 <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=20
the_website_2020
```

```
## {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
- PSWID
- Ownership
- From the "3. Water Supply Sources" section:
- Average Daily Use (MGD) - for each month

In the code chunk below scrape these values, assigning them to three 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, with the first value being 36.0100.

```
#3
water.system.name <- the_website_2020 %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>% h
pswid <- the_website_2020 %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
ownership <- the_website_2020 %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>% html_text
average.daily.mgd <- the_website_2020 %>% html_nodes('.fancy-table:nth-child(31) th+ td') %>% html_text
max.withdrawals.mgd <- the_website_2020 %>% html_nodes('th~ td+ td') %>% html_text()
```

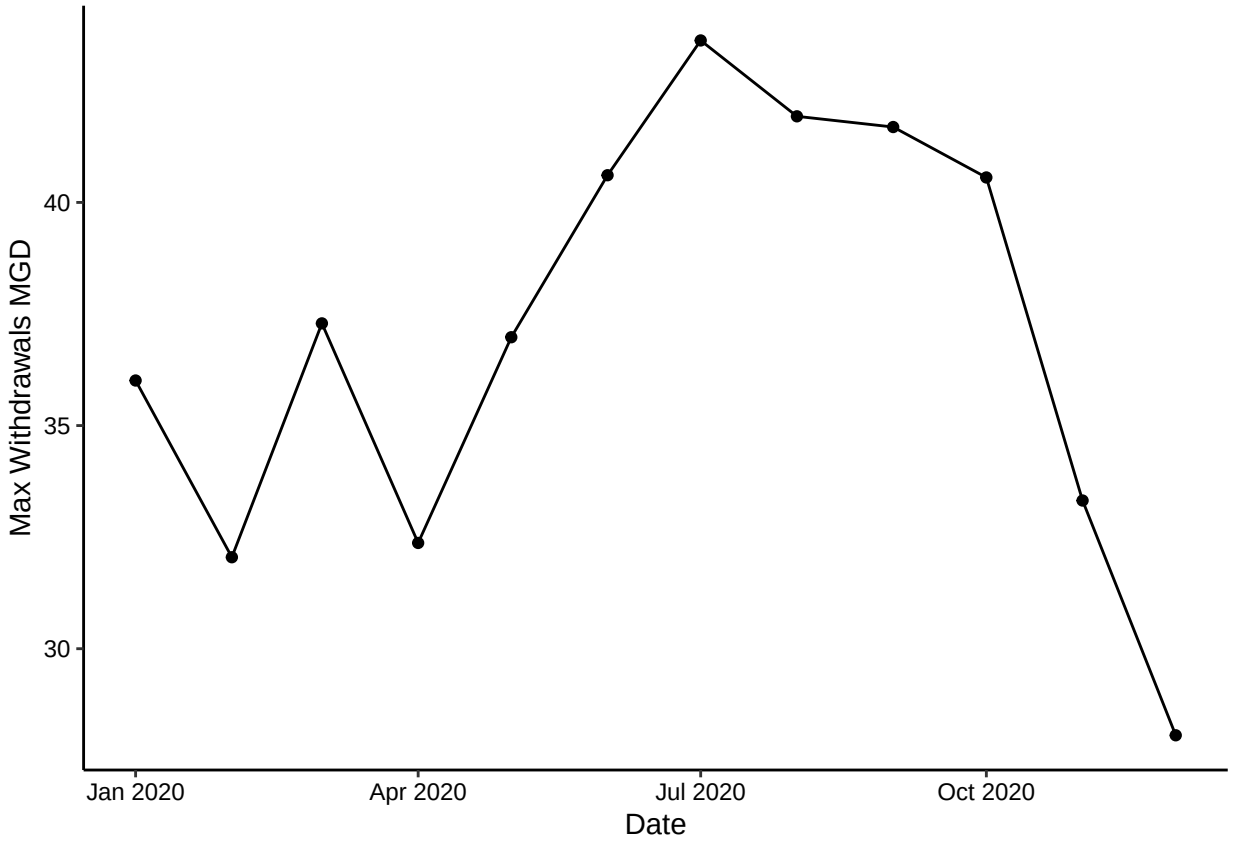
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   > 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 order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc. . .

5. Plot the max daily withdrawals across the months for 2020

```
#4

the_df_2020 <- data.frame("Water System Name" = water.system.name,
                          "PWSID" = pswid,
                          "Ownership" = ownership,
                          "Max Withdrawals MGD" = as.numeric(max.withdrawals.mgd),
                          "Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12),
                          "Year" = rep(2020, 12))
the_df_2020 <- the_df_2020 %>%
```

```
  mutate(Date = my(paste(Month,"-",Year))) %>%
  select("Water.System.Name", "PWSID","Ownership", "Max.Withdrawals.MGD", "Date")


#5
plot_2020 <- ggplot(the_df_2020, aes(x = Date, y = Max.Withdrawals.MGD)) +
  geom_point() +
  geom_line() +
  ylab("Max Withdrawals MGD")
print(plot_2020)
```



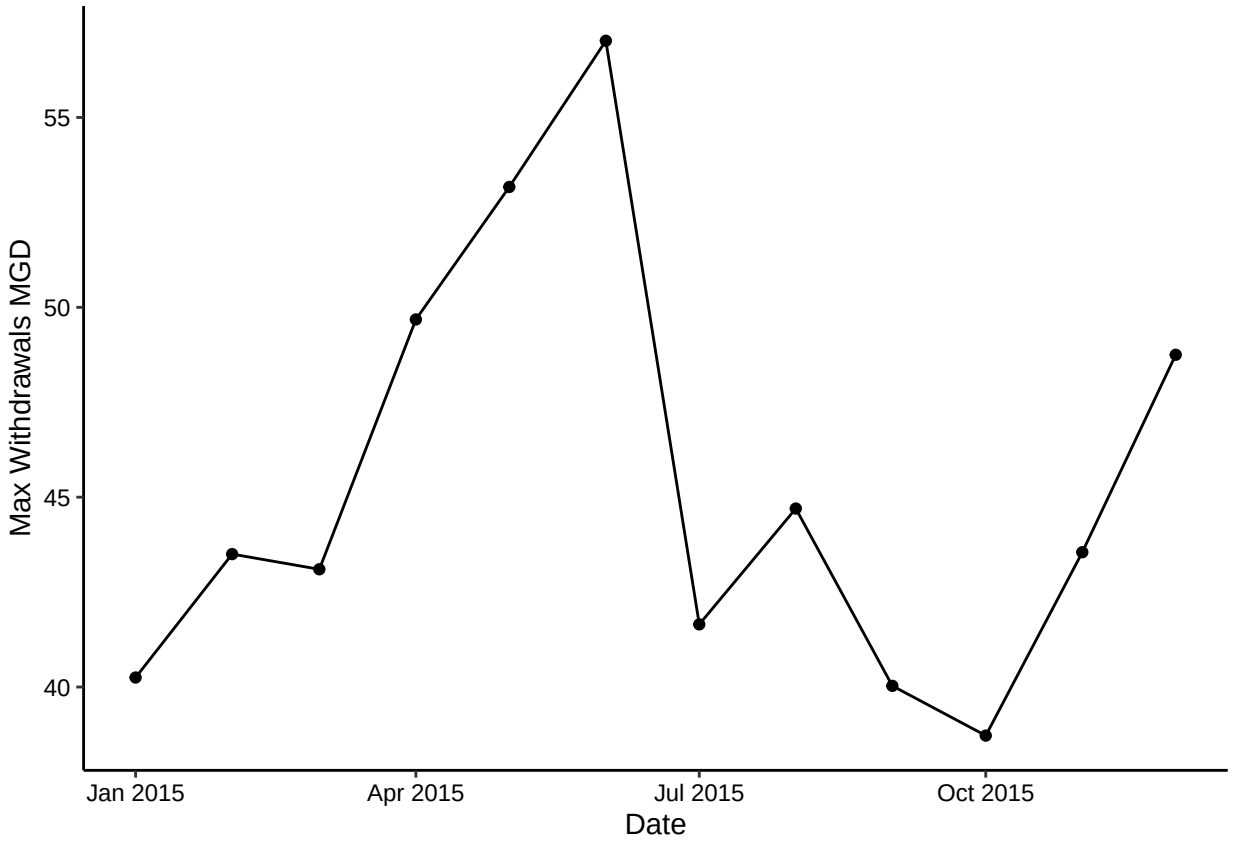
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
#6.
scrape.it <- function(pwsid, year_1){

URL <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',pwsid,'&year=',year_1)

water.system.name <- URL %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>% html_text()
pswid <- URL %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
ownership <- URL %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>% html_text()
average.daily.mgd <- URL %>% html_nodes('.fancy-table:nth-child(31) th+ td') %>% html_text()
max.withdrawals.mgd <- URL %>% html_nodes('th~ td+ td') %>% html_text()

the_df_all <- data.frame("Water System Name" = water.system.name,
```

```
                      "PWSID" = pswid,
                      "Ownership" = ownership,
                      "Max Withdrawals MGD" = as.numeric(max.withdrawals.mgd),
                      "Month" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12),
                      "Year" = rep(year_1, 12))
the_df_all <- the_df_all %>%
  mutate(Date = my(paste(Month,"-",Year))) %>%
  select("Water.System.Name", "PWSID","Ownership", "Max.Withdrawals.MGD", "Date")

return(the_df_all)}
```

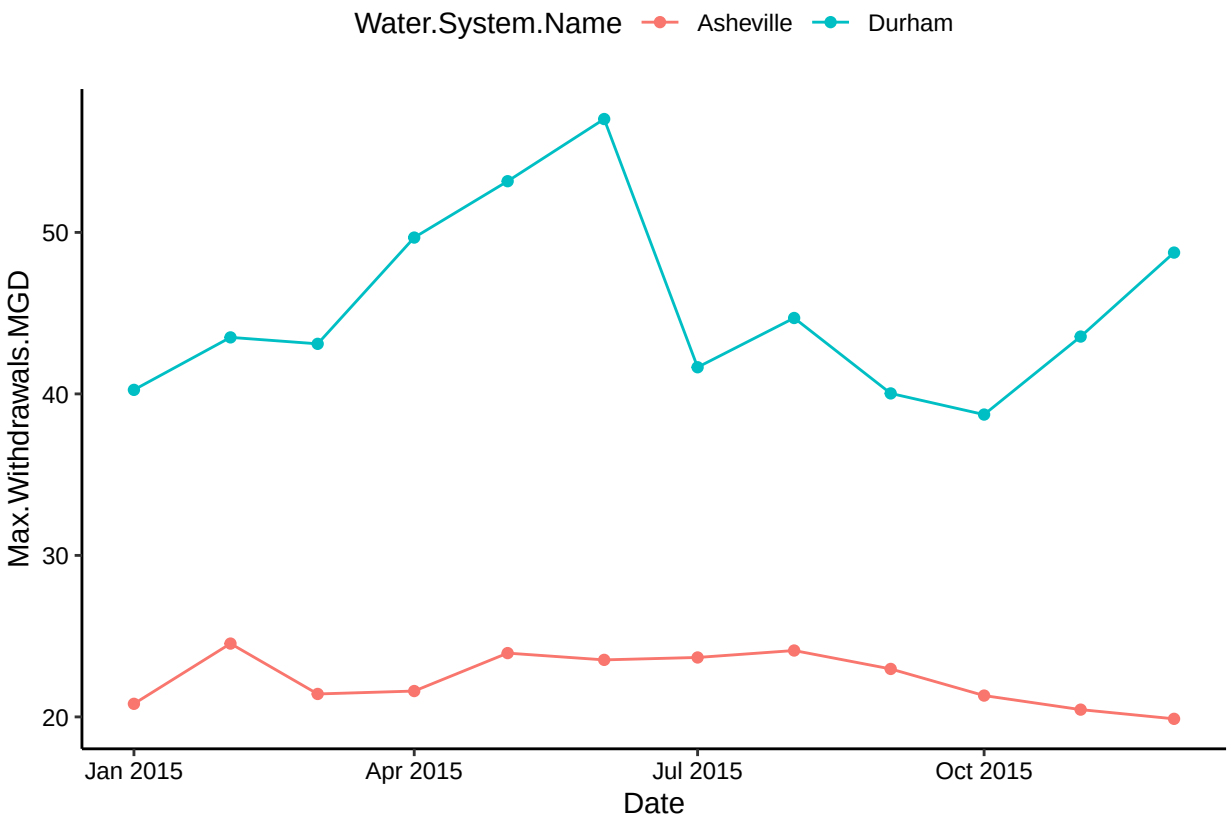
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7
Durham_PWSID0332010_Year2015 <- scrape.it('03-32-010',2015)

view(Durham_PWSID0332010_Year2015)

Plot_Durham_PWSID0332010_Year2015 <- ggplot(Durham_PWSID0332010_Year2015, aes(x = Date, y = Max.Withdrau
  geom_line() +
  geom_point() +
  ylab("Max Withdrawals MGD")
print(Plot_Durham_PWSID0332010_Year2015)
```



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 the Asheville to Durham's water withdrawals.

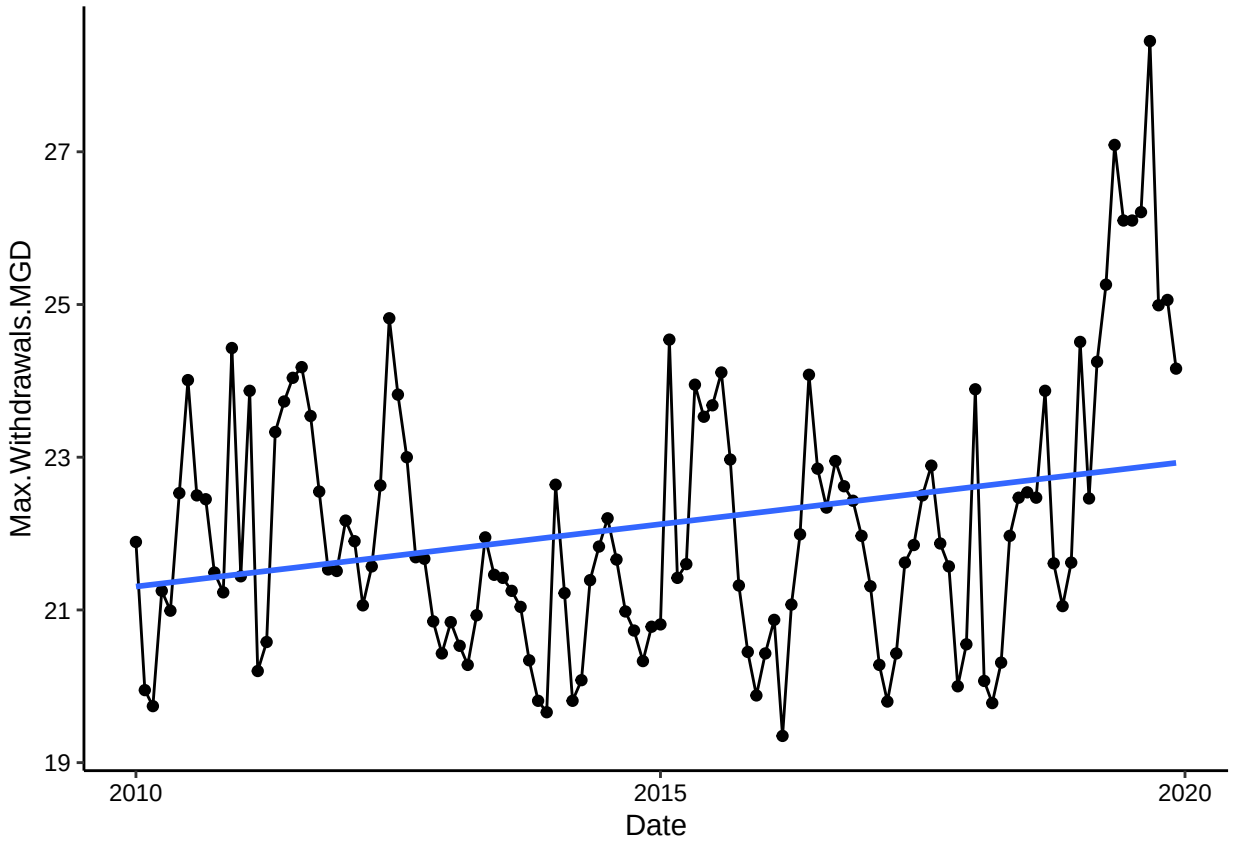
```
#8
Asheville_PWSID0111010_Year2015 <- scrape.it('01-11-010',2015)
view(Asheville_PWSID0111010_Year2015)

Asheville_Durham_2015 <- rbind(Asheville_PWSID0111010_Year2015,Durham_PWSID0332010_Year2015)

Plot_Asheville_Durham_2015 <- ggplot(Asheville_Durham_2015,aes(x = Date, y = Max.Withdrawals.MGD, color
  geom_point()+
  geom_line()
print(Plot_Asheville_Durham_2015)
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9
PWSID_iterate <- '01-11-010'
Years_iterate <- rep(2010:2019)

Asheville_2010_2019 <- map2(PWSID_iterate,Years_iterate, scrape.it)
view(Asheville_2010_2019)

Asheville_Bind_Years <- bind_rows(Asheville_2010_2019)

Plot_Asheville_2010_2019 <- ggplot(Asheville_Bind_Years,aes(x = Date, y = Max.Withdrawals.MGD)) +
  geom_point()+
  geom_line() +
  geom_smooth(se = FALSE, method = "lm")
print(Plot_Asheville_2010_2019)
```

```
## `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, the plot suggests that water usage has been increasing since 2010