Assignment 09: Data Scraping

Natalie von Turkovich

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/natalievonturkovich/Documents/DUKE/Courses/Spring 22/ENV_872_EDA/Environmental_Data_Analy
```

```
library(tidyverse)
library(lubridate)
library(viridis)
#install.packages("rvest")
library(rvest)
#install.packages("dataRetrieval")
library(dataRetrieval)
#install.packages("tidycensus")
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.) The RVest scraping workflow is as follows: 1. Connect to the website using the `read_html` function. 2. Locate specific elements in the web site via the node IDs found using Selector Gadget, reading them in using `html_nodes` 3. Read the text value(s) associated with those nodes into the coding environment via `html_text` 4. Wrangle values into a dataframe...

```
#2 Link to the web site using read_html
the_website<- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019')
```

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 %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
pswid <- the_website %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
  html_text()
ownership <- the_website %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
max.withdrawals.mgd <- the_website %>% html_nodes('th~ td+ td') %>%
  html_text()
```

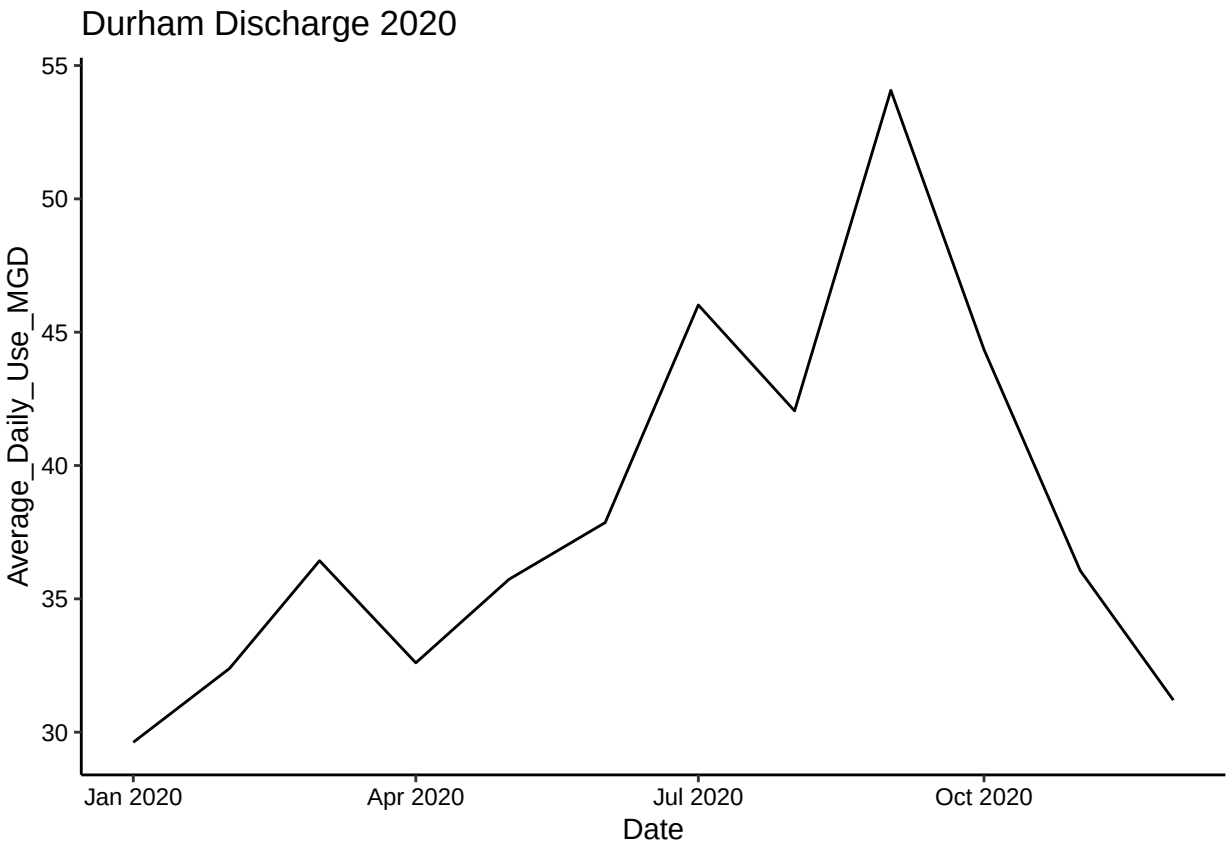
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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 Construct a dataframe from the values

Withdrawals_2020.df<-data.frame(
  "Year" =rep(2020, times = 12),
  "Water_System_Name" =rep(water.system.name, times = 12),
  "PSWID" = rep(pswid, times = 12),
  "Ownership"=rep(ownership, times = 12),
  "Average_Daily_Use_MGD"=as.numeric(max.withdrawals.mgd),
  "Month_Number" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12)
)

Withdrawals_2020.df<-Withdrawals_2020.df %>%
  mutate(Date= my(paste0(Month_Number,"-",Year)))

#5
ggplot(Withdrawals_2020.df, aes(x=Date, y=Average_Daily_Use_MGD))+
    geom_line()+
    labs(title = 'Durham Discharge 2020')
```



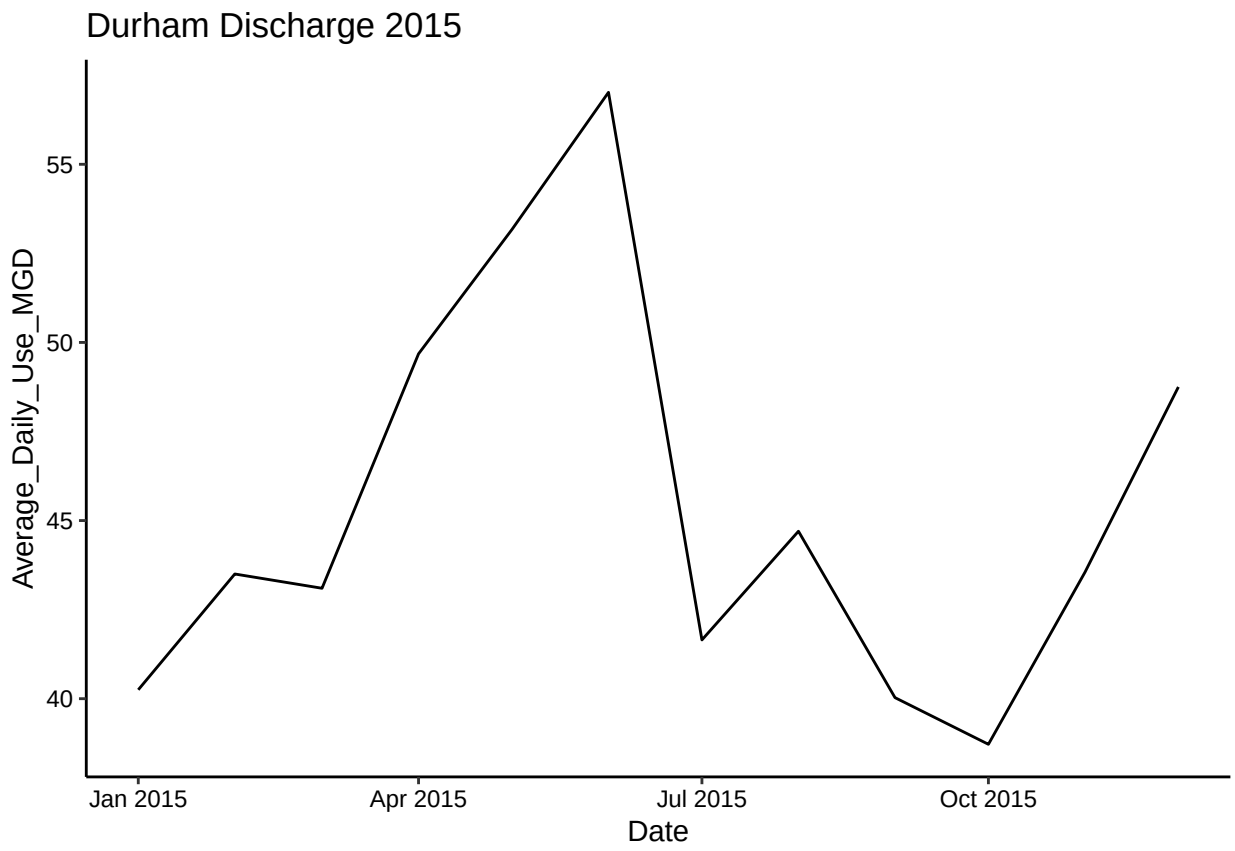
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
#6
scrape.it.2 <- function(the_year, PSWID){
  #Get the proper url
  the_url <- paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', PSWID,
                    '&year=', the_year)

  the_website<- read_html(the_url)

#2&3 Locate elements and read their text attributes into variables
water.system.name <- the_website %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>%
  html_text()
pswid <- the_website %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>%
  html_text()
ownership <- the_website %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>%
  html_text()
max.withdrawals.mgd <- the_website %>% html_nodes('th~ td+ td') %>%
  html_text()

#3 Construct a dataframe from the values
Withdrawals_fn<-data.frame(
  "Year" =rep(the_year, times = 12),
```

```
  "Water_System_Name" =rep(water.system.name, times = 12),
  "PSWID" = rep(pswid, times = 12),
  "Ownership"=rep(ownership, times = 12),
  "Average_Daily_Use_MGD"=as.numeric(max.withdrawals.mgd),
  "Month_Number" = c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12)
)

Withdrawals_fn<-Withdrawals_fn %>%
  mutate(Date= my(paste0(Month_Number,"-",Year)))

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

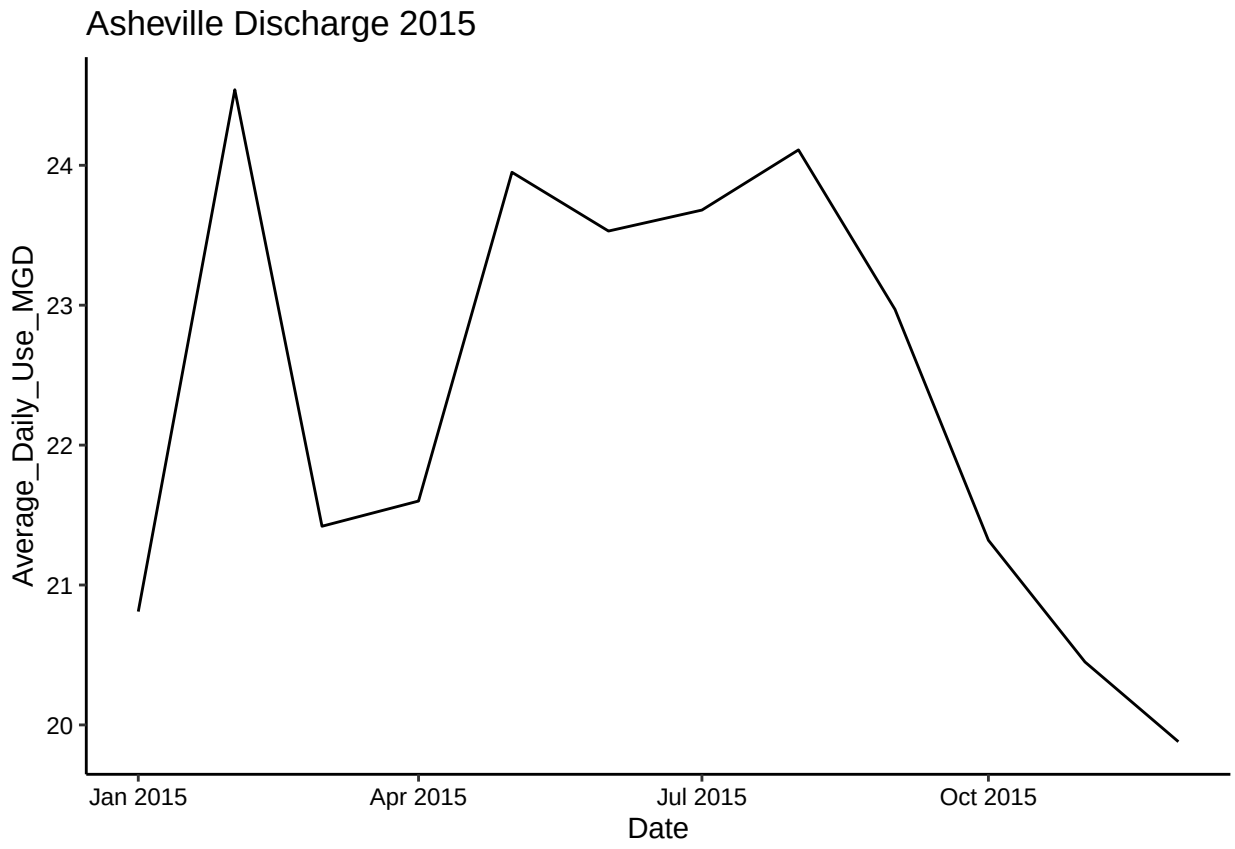
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
Durham_2015<-scrape.it.2(2015, '03-32-010')


Durham_2015.plot<-ggplot(Durham_2015, aes(x=Date, y=Average_Daily_Use_MGD))+
  geom_line()+
  labs(title = 'Durham Discharge 2015')
Durham_2015.plot
```



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_2015<-scrape.it.2(2015, '01-11-010')

Asheville_2015.plot<-ggplot(Asheville_2015, aes(x=Date, y=Average_Daily_Use_MGD))+
  geom_line()+
  labs(title = 'Asheville Discharge 2015')
Asheville_2015.plot
```

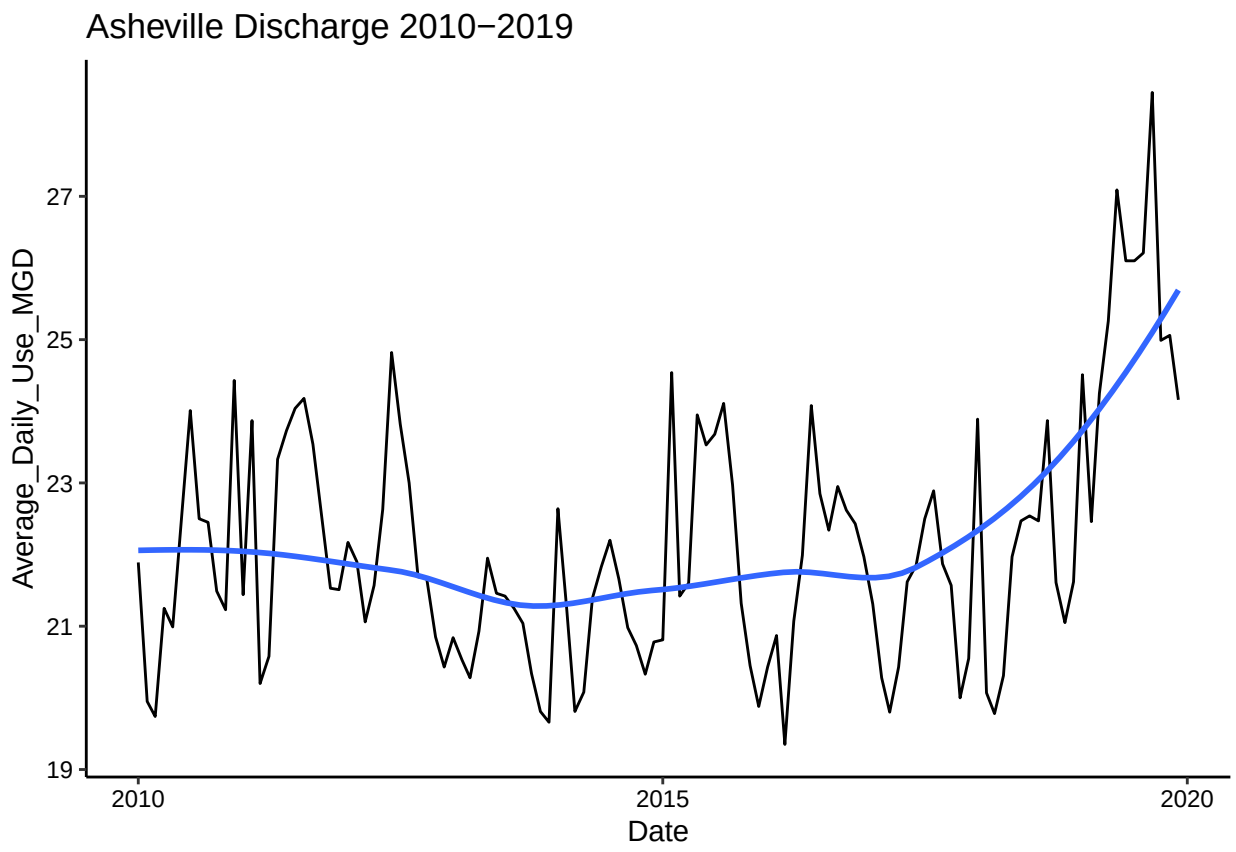


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9
#Set the inputs to scrape years 2010 to 2019
the_years = rep(2010:2019)
PSWID = '01-11-010'

#purrr's map function
Asheville_discharge_dfs <- map(the_years,scrape.it.2,PSWID=PSWID) #tidy map function

#Conflate the returned dataframes into a single dataframe
Asheville_discharge_10_19 <- bind_rows(Asheville_discharge_dfs)
```

```r
#bind row is tidy version of r bind

#Plot, because it's fun and rewarding
ggplot(Asheville_discharge_10_19,aes(x=Date,y=Average_Daily_Use_MGD)) +
  geom_line() +
  labs(title = 'Asheville Discharge 2010-2019')+
  geom_smooth(method="loess",se=FALSE)
```

```
## `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 Asheville does seem to have a trend of increasing average daily water usage over time. Since ~2017 we can see a steady increase in daily use.