Assignment 10: Data Scraping

Camber Vincent

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(rvest)
library(here)
library(purrr)

getwd() #check working directory
```

```
## [1] "/Users/cambervincent/EDA_Fall_2024/Assignments"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2023 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=2023

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

```
#2
#set the scraping website
webpage<-read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
```

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)".

```
#3
water_system<-webpage%>% #scrape water system name
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)")%>%
  html_text()
water_system
```

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

```
pwsid<-webpage%>% #scrape PWSID
  html_nodes("td tr:nth-child(1) td:nth-child(5)")%>%
  html_text()
pwsid
```

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

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

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

```
maximum_use<-webpage%>% #scrape maximum day use
  html_nodes("th~ td+ td")%>%
  html_text()
maximum_use
```

```
##  [1] "28.9000" "33.3000" "43.7000" "30.0000" "40.0000" "37.2300" "34.2000"
##  [8] "44.9000" "40.3500" "30.9000" "56.7000" "33.3000"
```

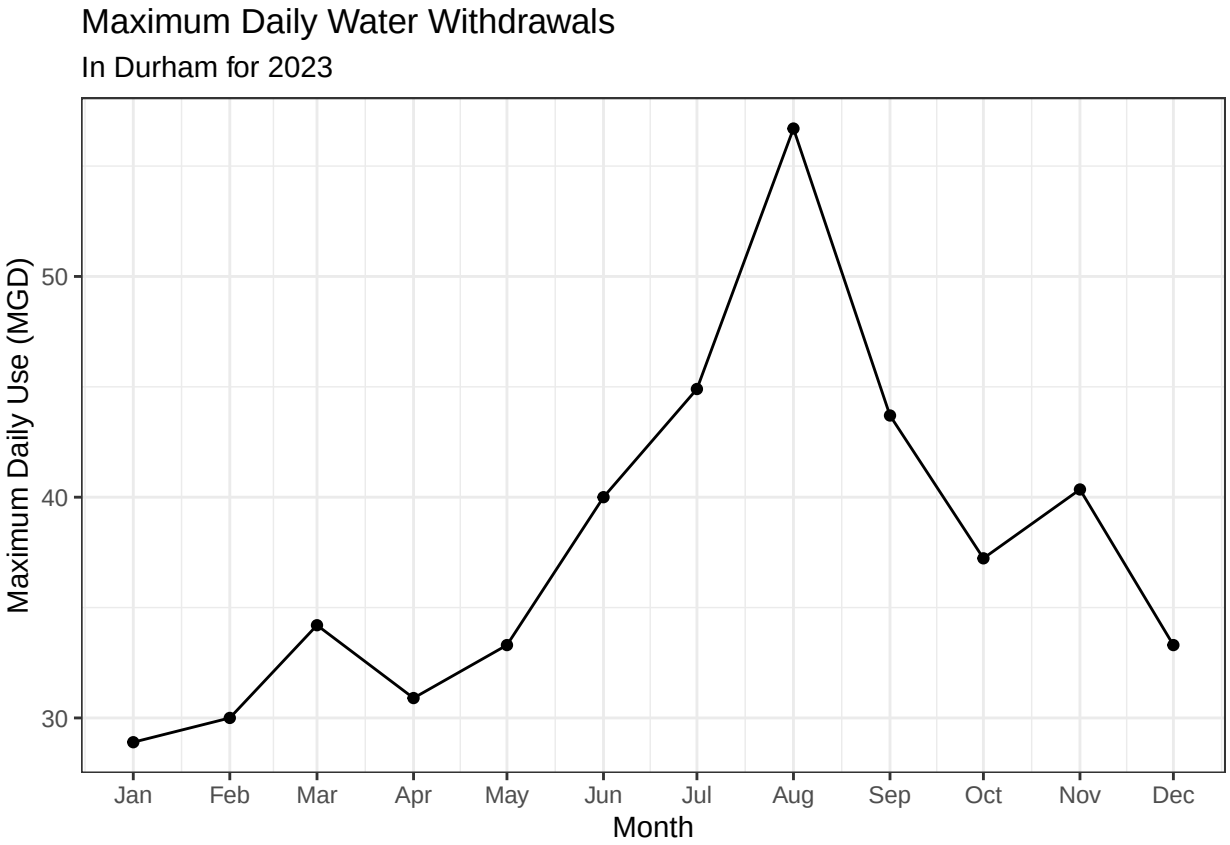
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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 2023, making sure, the months are presented in proper sequence.

```
#4
months<-c("Jan","May","Sep","Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug","Dec") #assign months order
year<-rep(2023,length(months)) #create year column

maximum_use_data<-data.frame( #create dataframe
  Year=year,
  Month=months,
  Water_System_Name=rep(water_system,length(months)), #repeat in all rows
  PWSID=rep(pwsid,length(months)), #repeat in all rows
  Ownership=rep(ownership,length(months)), #repeat in all rows
  Maximum_Day_Use=as.numeric(maximum_use) #covert to numeric format
)

maximum_use_data$Date<-as.Date(
  paste(maximum_use_data$Year,maximum_use_data$Month,"01",sep="-"), #concatenate year and month
  "%Y-%b-%d") #add date column

maximum_use_data<-maximum_use_data[order(maximum_use_data$Date),] #reorder dataframe by month

#5
ggplot(maximum_use_data,aes(x=Date,y=Maximum_Day_Use))+
  geom_line()+
  geom_point()+
  scale_x_date(date_labels="%b",date_breaks="1 month")+ #format month labels
  labs(
    title="Maximum Daily Water Withdrawals",
    subtitle="In Durham for 2023",
    x="Month",
    y="Maximum Daily Use (MGD)")+ #set labels
  theme_bw()
```



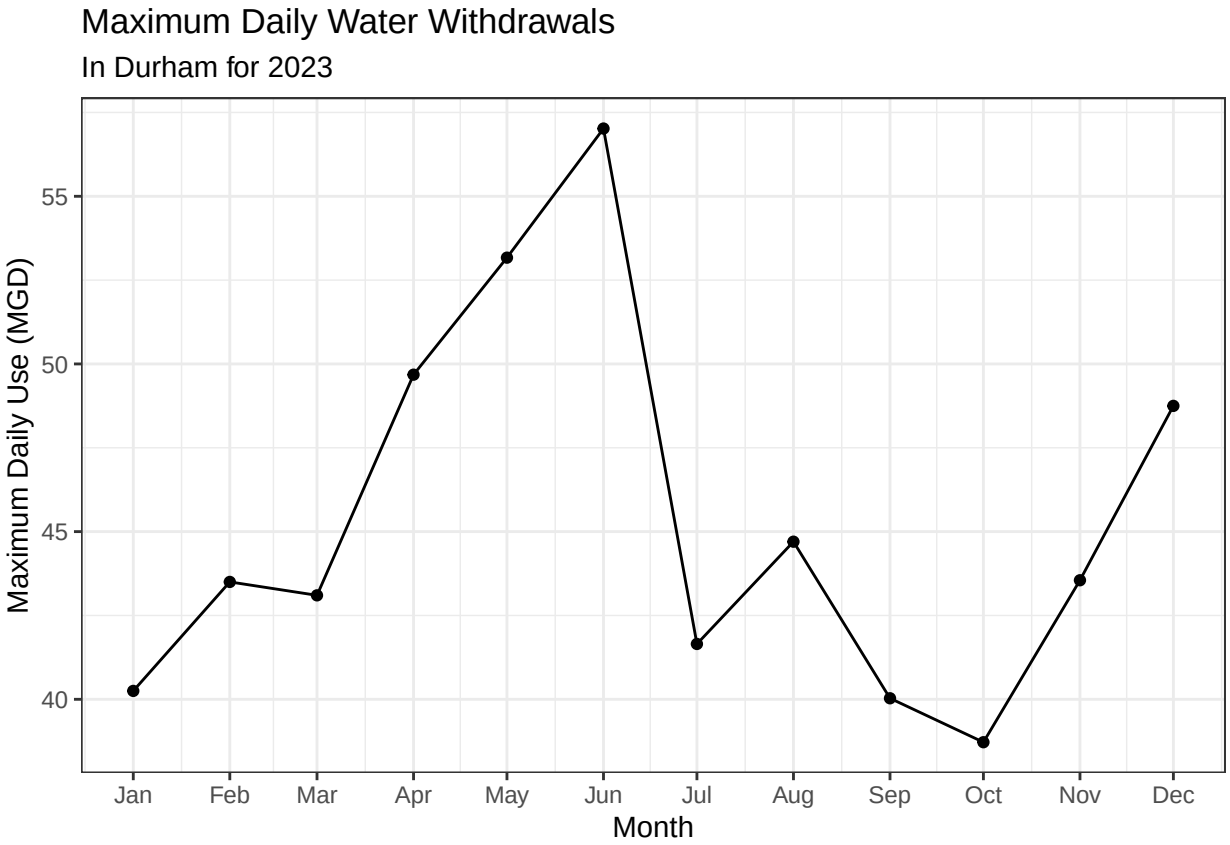
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
scrape_function<-function(pwsid,year){ #two inputs of PWSID and year

  #construct the URL dynamically based on input PWSID and year
  url<-paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=", #paste0 leaves no separators
              pwsid,"&year=",year)

  #read the webpage
  webpage<-read_html(url)

  #scrape water system name, ownership, and maximum daily use
  water_system<-webpage%>%
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)")%>%
    html_text()

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

  maximum_use<-webpage%>%
```

```
    html_nodes("th~ td+ td")%>%
    html_text()

  #create dataframe
  months<-c("Jan","May","Sep","Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug","Dec") #assign months

  maximum_use_data<-data.frame( #create frame
    Year=rep(year,length(months)), #year defined in user inputs
    Month=months,
    Water_System_Name=rep(water_system, length(months)),
    PWSID=rep(pwsid,length(months)), #pwsid defined in user inputs
    Ownership=rep(ownership,length(months)),
    Maximum_Day_Use=as.numeric(maximum_use)
  )

  maximum_use_data$Date<-as.Date( #add date column
    paste(year,maximum_use_data$Month,"01",sep="-"),
    "%Y-%b-%d")

  maximum_use_data<-maximum_use_data[order(maximum_use_data$Date),] #reorder data frame by date

  #return the dataframe
  return(maximum_use_data)
}
```

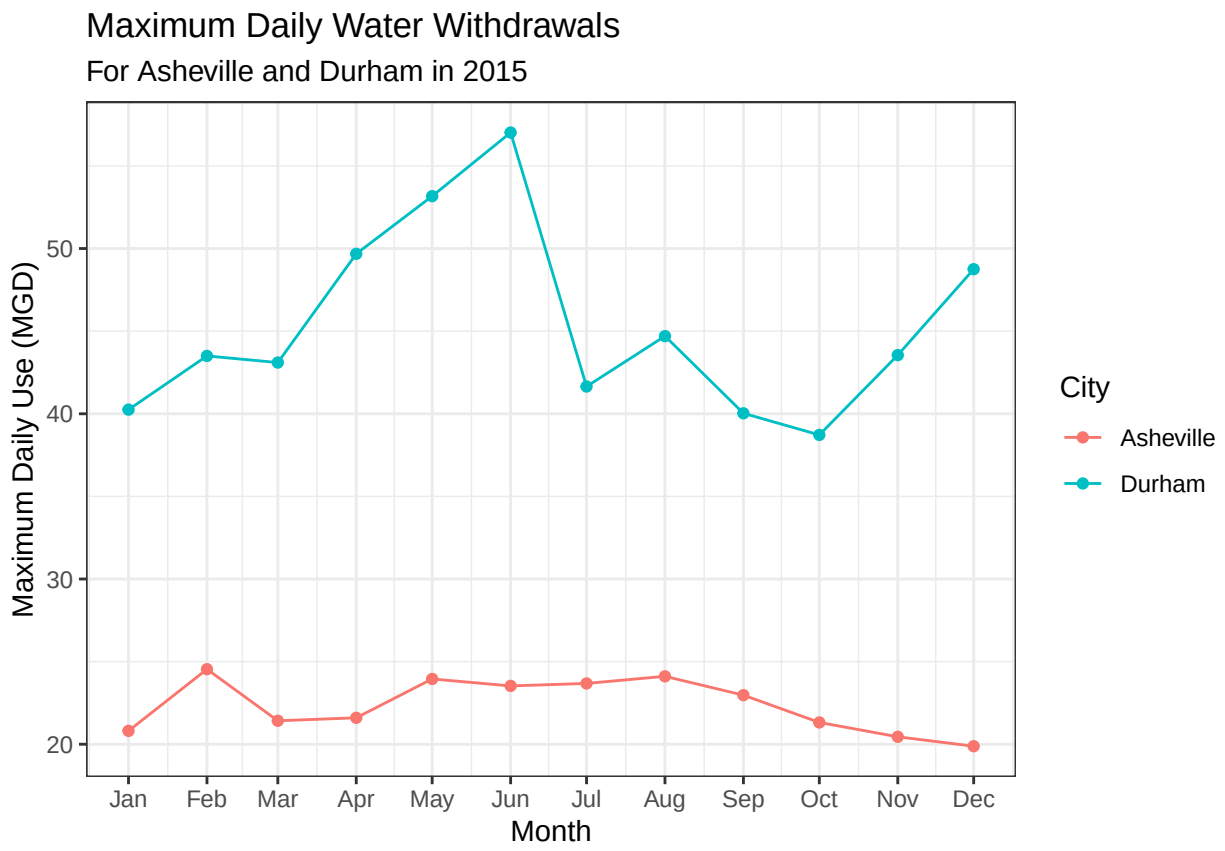
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
durham_2015<-scrape_function('03-32-010',2015) #scrape data

ggplot(durham_2015,aes(x=Date,y=Maximum_Day_Use))+ #plot data
  geom_line()+
  geom_point()+
  scale_x_date(date_labels="%b",date_breaks="1 month")+ #format month labels
  labs(
    title="Maximum Daily Water Withdrawals",
    subtitle="In Durham for 2023",
    x="Month",
    y="Maximum Daily Use (MGD)")+ #set labels
  theme_bw()
```



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
asheville_2015<-scrape_function('01-11-010',2015)

durham_2015$City<-"Durham" #adding city column
asheville_2015$City<-"Asheville" #adding city column

durham_asheville<-rbind(durham_2015,asheville_2015) #combining data frames

ggplot(durham_asheville,aes(x=Date,y=Maximum_Day_Use,color=City))+
  geom_line()+
  geom_point()+
  scale_x_date(date_labels="%b",date_breaks="1 month")+
  labs(title="Maximum Daily Water Withdrawals",
    subtitle="For Asheville and Durham in 2015",
    x="Month",
    y="Maximum Daily Use (MGD)")+
  theme_bw()
```

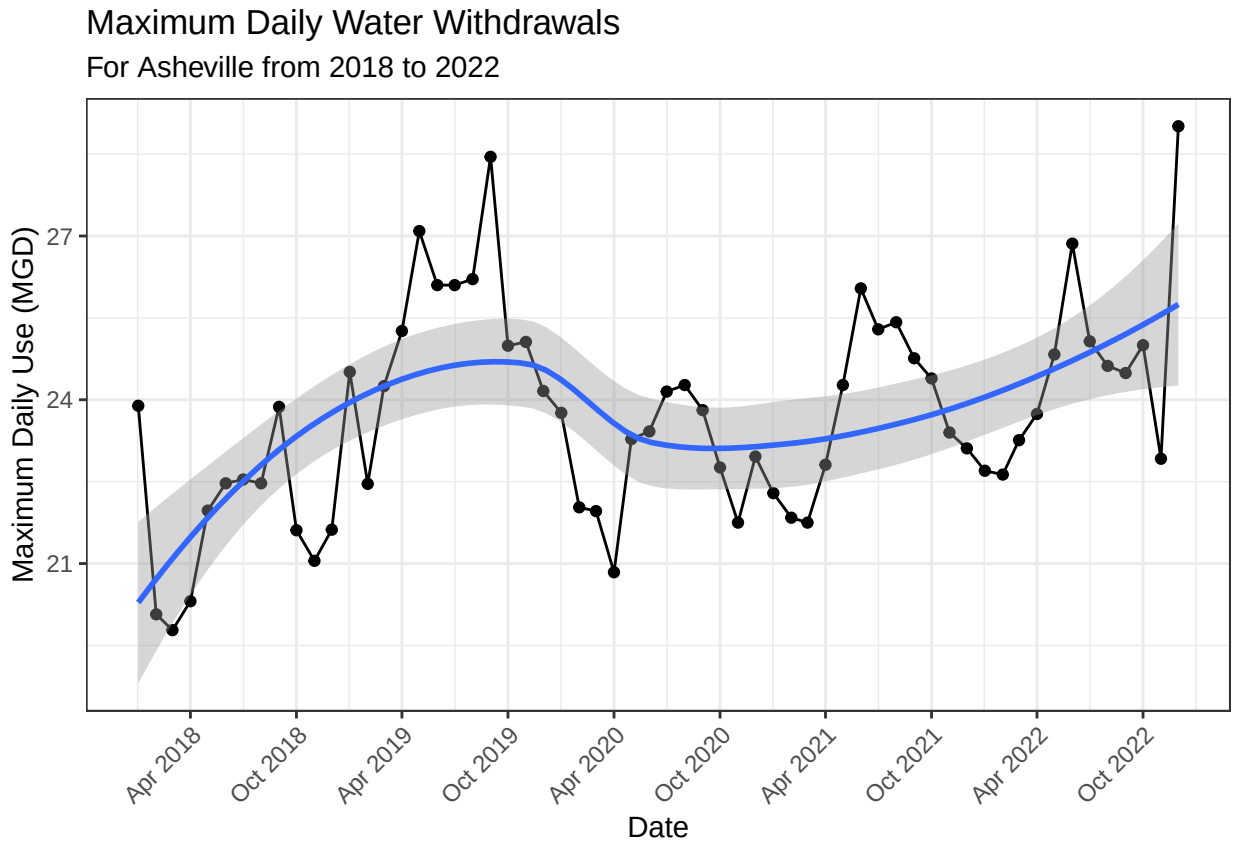


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022.Add a smoothed line to the plot (method = 'loess').

   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
pwsid<-"01-11-010" #define pwsid of Asheville
years<-2018:2022 #define the range of years to scrape

asheville_data_all<-map2( #using map2
  .x=rep(pwsid,length(years)), #pwsid used for each year
  .y=years, #scrape by each year
  .f=scrape_function)%>% #function to apply
  bind_rows() #combine the individual dataframes into one

ggplot(asheville_data_all,aes(x=Date,y=Maximum_Day_Use))+ #create plot
  geom_line()+
  geom_point()+
  geom_smooth(method='loess')+
  scale_x_date(date_labels="%b %Y",date_breaks="6 months")+ #reshaping axis
  labs(title="Maximum Daily Water Withdrawals",
    subtitle="For Asheville from 2018 to 2022",
    x="Date",
```

```
  y="Maximum Daily Use (MGD)")+
theme_bw()+
theme(axis.text.x=element_text(angle=45,hjust=1)) #adjusting x-axis labels
```



Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time?

Answer: Just by looking at the plot, it appears that Asheville has a generally increasing trend in water usage over time. It rose quickly from 2018 through September 2019, fell to April 2020, and then continued to steadily increase through the rest of the mapped data.