Assignment 10: Data Scraping

David Liddle

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
#Install familiar packages
library(tidyverse);library(lubridate);library(viridis);library(here)

#install.packages("rvest")
library(rvest)

#install.packages("dataRetrieval")
library(dataRetrieval)

#install.packages("tidycensus")
library(tidycensus)
#2 Build ggplot theme
david_theme <- theme_classic(base_size = 14) +
theme(axis.text = element_text(color = "black"),
legend.position = "right",
panel.grid.minor = element_line(color = "gray", linetype = "solid"),
panel.grid.major = element_line(color = "gray", linetype = "solid"),
legend.background = element_rect(fill = "gray"))
#Set the theme
theme_set(david_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.)

```
#2 Fetch the web resources from the URL
webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
webpage
```

```
## {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)".

```
#3
WaterSystem_name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1)
             td:nth-child(2)") %>%
  html_text()

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

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

```
MGD <- webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()
```

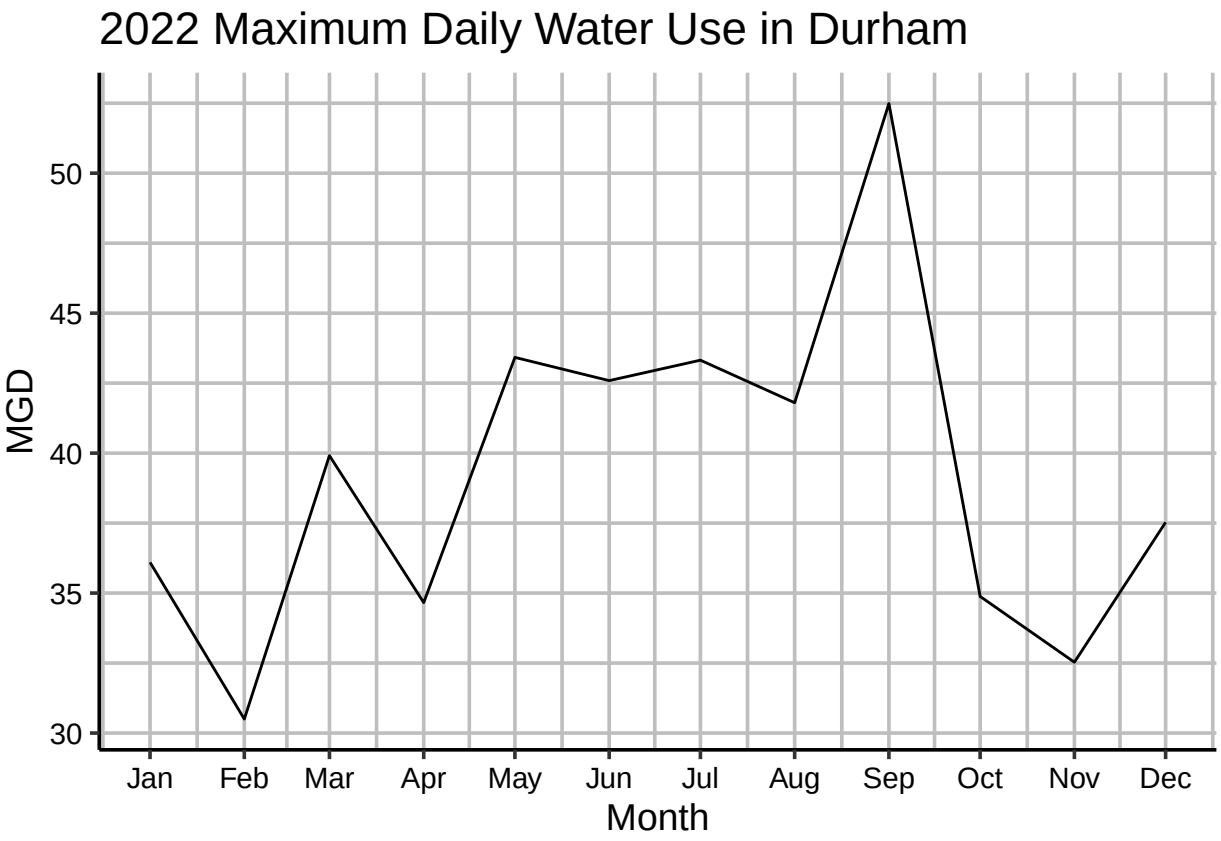
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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 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 Convert scraped data to data frame
#Create dataframe

df_MGD_1 <- data.frame(
  WaterSystem_name = WaterSystem_name,
  PWSID = PWSID,
  Ownership = Ownership,
  MGD = as.numeric(MGD),
  Month = c(1,5,9,2,6,10,3,7,11,4,8,12),
  Year = 2022)

#Making a date column
df_MGD_1$Date <- as.Date(my(paste(df_MGD_1$Month, "-", df_MGD_1$Year)))

#5 Create Line plot
ggplot(df_MGD_1,aes(x=Date,y=MGD)) +
  geom_line() +
  labs(title ="2022 Maximum Daily Water Use in Durham",
       x = "Month",y = "MGD") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```



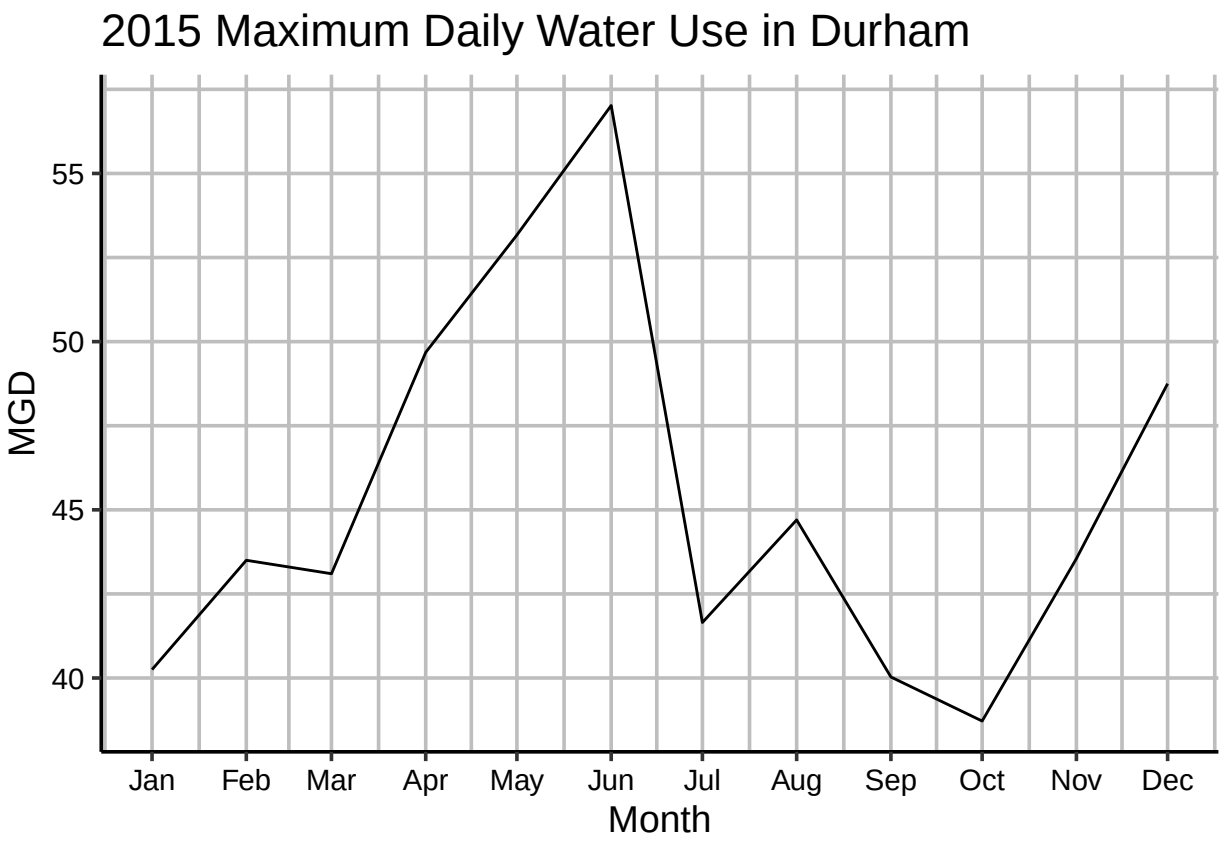
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
#Construct the scraping web address, i.e. its URL
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
the_facility <- '03-32-010'
the_year <- 2015

#Retrieve the website contents
scrape.it <- function(the_facility, the_year){
the_website <- read_html(paste0(the_base_url, the_facility, '&year=', the_year))

#Set the element address variables (determined in the previous step)
the_city <- "div+ table tr:nth-child(1) td:nth-child(2)"
the_PWSID <- "td tr:nth-child(1) td:nth-child(5)"
the_ownership <- "div+ table tr:nth-child(2) td:nth-child(4)"
the_MGD <- "th~ td+ td"

#Scrape the data items
the_city <- the_website %>% html_nodes(the_city) %>% html_text()
the_PWSID <- the_website %>% html_nodes(the_PWSID) %>%  html_text()
the_ownership <- the_website %>% html_nodes(the_ownership) %>% html_text()
the_MGD <- the_website %>% html_nodes(the_MGD) %>% html_text()
the_Month <- c(1,5,9,2,6,10,3,7,11,4,8,12)
```

```
#Construct a dataframe from the scraped data
df_mgd_data <- data.frame(
"Month" = the_Month,
"Year" = rep(the_year),
"Withdrawals" = as.numeric(the_MGD)) %>%
mutate(
WaterSystem = !!the_city,
PWSID = !!the_PWSID,
Ownership = !!the_ownership,
Date = my(paste(Month, "-",Year))
)
return(df_mgd_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_MGD <- scrape.it('03-32-010', 2015)

#Create Line plot
ggplot(Durham_2015_MGD,aes(x=Date,y=Withdrawals)) +
  geom_line() +
  labs(title ="2015 Maximum Daily Water Use in Durham",
       x = "Month",y = "MGD") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```

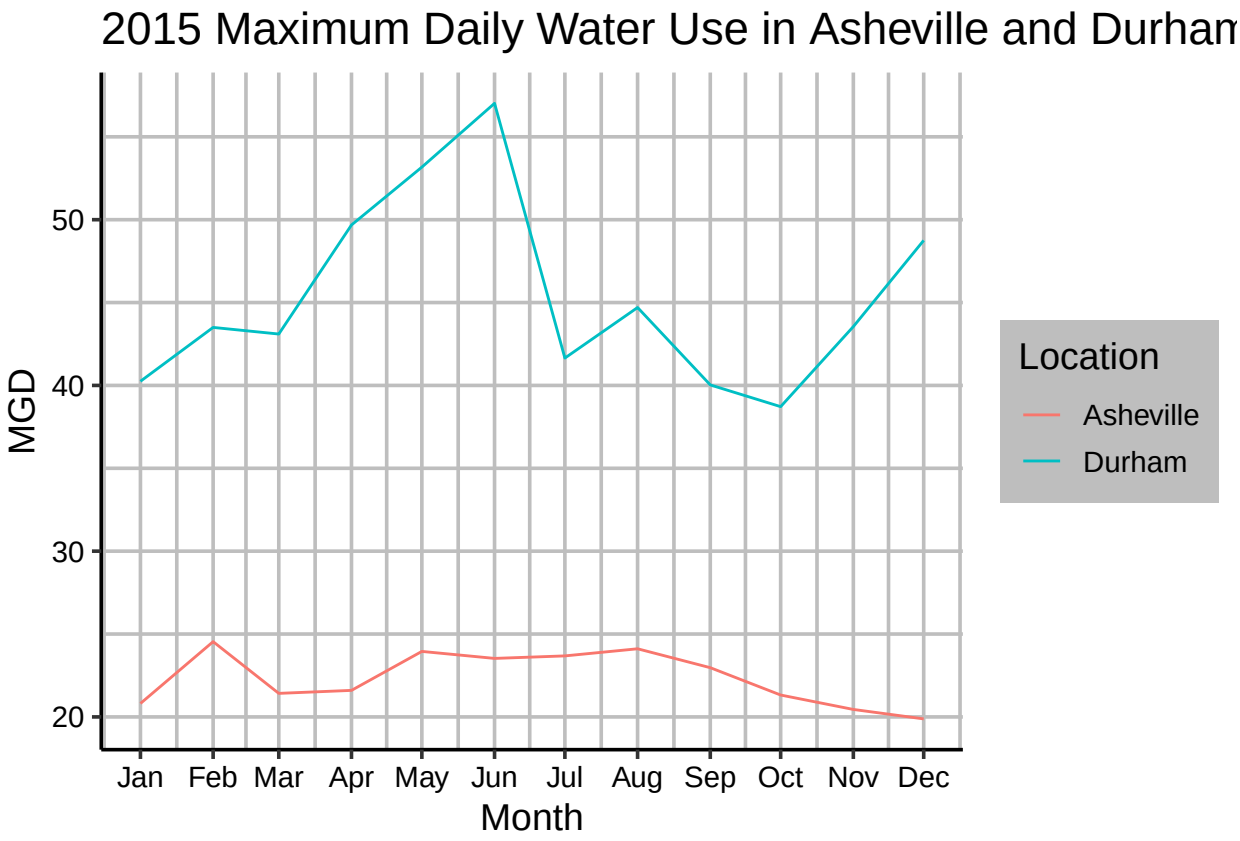


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8

Asheville_2015_MGD <- scrape.it('01-11-010', 2015)

Combined_MGD <- full_join(Asheville_2015_MGD,Durham_2015_MGD)
```

```
## Joining with `by = join_by(Month, Year, Withdrawals, WaterSystem, PWSID,
## Ownership, Date)`
```

```
#Create Line plot comparing Asheville to Durham
ggplot(Combined_MGD,aes(x=Date,y=Withdrawals, color=WaterSystem)) +
  geom_line() +
  labs(title ="2015 Maximum Daily Water Use in Asheville and Durham",
       x = "Month",y = "MGD", color="Location") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b")
```

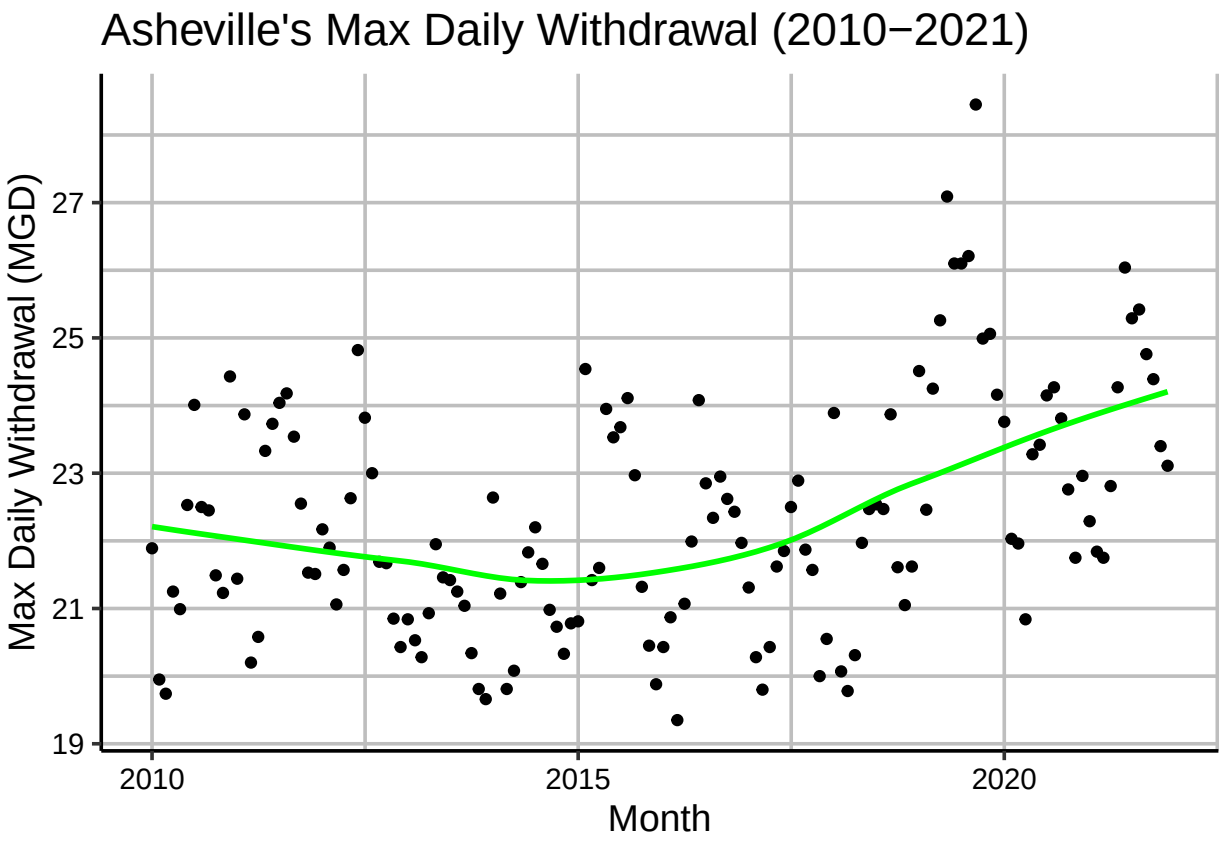


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   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.

```r
#9

# Specify the years and facility ID
years <- 2010:2021
facility_id <- '01-11-010'  # Assuming Asheville's facility ID is constant

# Use map2 to iteratively run the function over two inputs
df_asheville <- map2(facility_id, years, scrape.it) %>%
  bind_rows()

# Plot max daily withdrawal for Asheville over the years with a smoothed line
ggplot(df_asheville, aes(x = Date, y = Withdrawals)) +
  geom_point() +
  geom_smooth(method = 'loess', se = FALSE, color="green") +
  labs(title = "Asheville's Max Daily Withdrawal (2010-2021)",
       x = "Month", y = "Max Daily Withdrawal (MGD)")
```

```
## `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? > Answer: Just by looking at the graph, we can see a downward trend from 2010-2015, where the maximum water usage decreased slightly. Between 2015-2021 the water usage started a sharper upward trend. >