Assignment 10: Data Scraping

Emma Kaufman

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(lubridate)
library(here); here()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
library(rvest)

getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

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
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

Water_System <- 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()
```

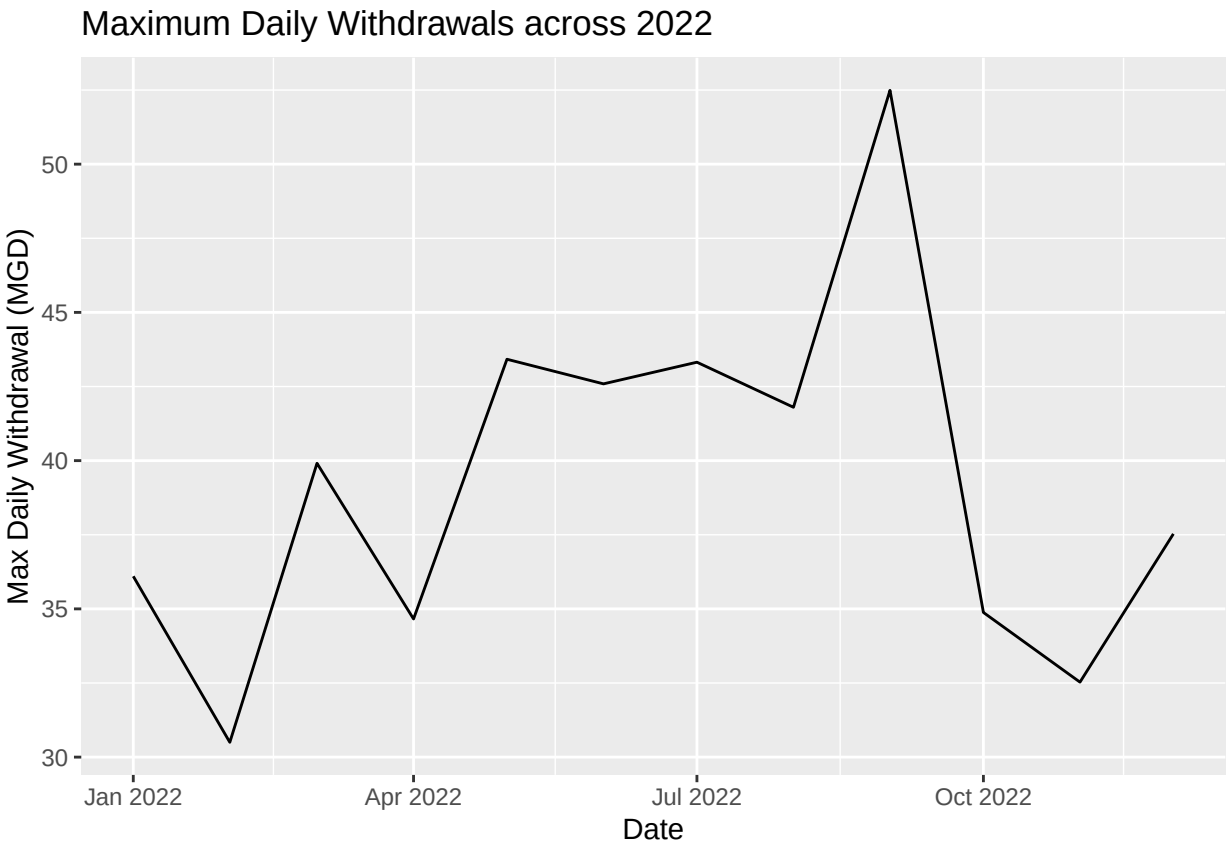
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

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
Month<- webpage %>%
  html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
  html_text()

year= '2022'

df <- data.frame(
  'Water System Name' = Water_System,
  'PWSID' = PWSID,
  'Ownership' = Ownership,
  'MGD' =  as.numeric(gsub(',','', MGD)),
  'Month' = Month,
  'Year' = year
) %>%
  mutate(Date = my(paste(Month, Year)))
#5
ggplot(df,aes(x=Date,y=MGD)) +
  geom_line() +
  labs(x = "Date",
       y = "Max Daily Withdrawal (MGD)",
       title= "Maximum Daily Withdrawals across 2022")
```



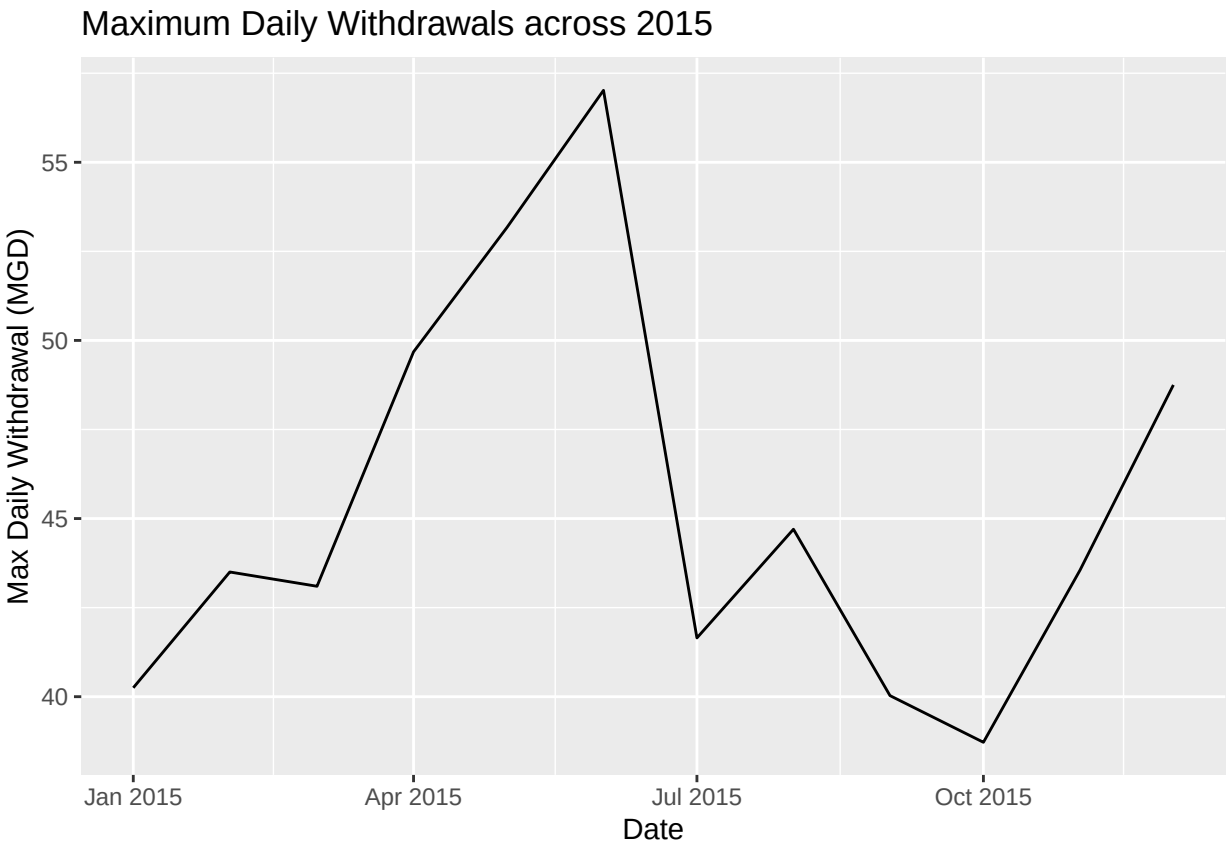
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.

#Create our scraping function
scrape.it <- function(the_year, the_facility){

  #Retrieve the website contents
  the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
  the_scrape_url <- read_html(paste0(
    'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
    the_facility,
    '&year=',
    the_year))

 #Set the element address variables
  Water_System_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)"
  MGD_tag <- "th~ td+ td"
  #Month_tag<- ".fancy-table:nth-child(31) tr+ tr th"

  #Scrape the data items
```

```
  Water_System <-  the_scrape_url %>% html_nodes(Water_System_tag) %>% html_text()
  PWSID <-  the_scrape_url %>% html_nodes(PWSID_tag) %>% html_text()
  Ownership <-  the_scrape_url %>% html_nodes(Ownership_tag) %>% html_text()
  MGD <-  the_scrape_url %>% html_nodes(MGD_tag) %>% html_text()
 # Month<- the_scrape_url %>% html_nodes(Month_tag) %>% html_text()

  #Convert to a dataframe
 df_Water <- data.frame(
  'Water System Name' = rep(Water_System),
  'PWSID' = rep(PWSID),
  'Ownership' = Ownership,
  'MGD' =  as.numeric(gsub(',','', MGD)),
  'Month' = c('Jan', 'May', 'Sep', 'Feb', 'Jun', 'Oct', 'Mar', 'Jul', 'Nov', 'Apr', 'Aug', 'Dec'),
  'Year' = rep(the_year)
) %>%
  mutate(Date = my(paste(Month, Year)))
  #Pause for a moment - scraping etiquette
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

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

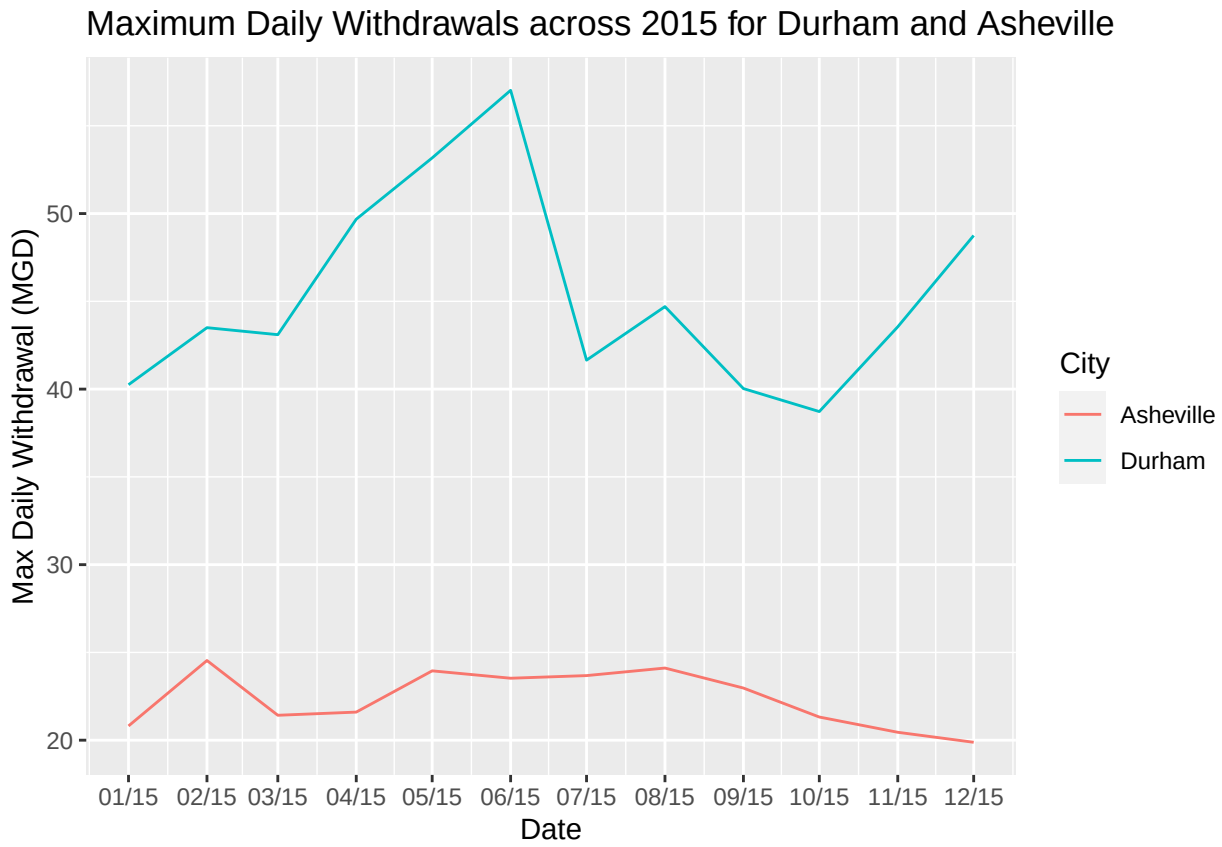
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

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

ggplot(the_df,aes(x=Date,y=MGD)) +
  geom_line() +
  labs(x = "Date",
       y = "Max Daily Withdrawal (MGD)",
       title= paste0("Maximum Daily Withdrawals across 2015"))
```



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

combined_df <- rbind(the_df, Asheville_df)

ggplot(combined_df, aes(x = Date, y = MGD, color = Water.System.Name)) +
  geom_line() +
  scale_x_date(date_breaks = "1 month", date_labels = '%m/%y')+
  labs(
       y = "Max Daily Withdrawal (MGD)",
       title = "Maximum Daily Withdrawals across 2015 for Durham and Asheville",
       color= "City")
```

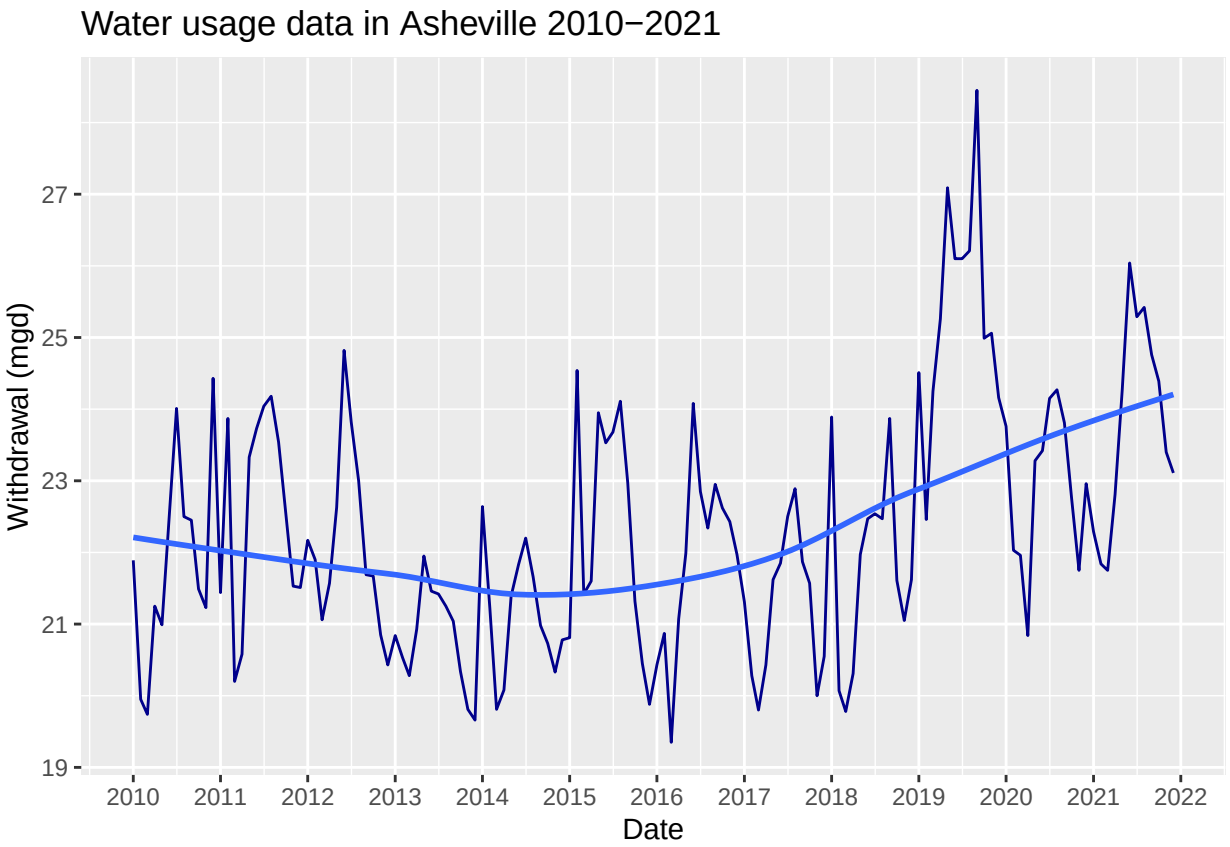


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021. Add a smoothed line to the plot (method = 'loess').

   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.

```
#9
years <- seq(2010,2021)

#
Asheville_many <- map(years, scrape.it, the_facility= '01-11-010') %>%
  bind_rows()

#Plot
ggplot(Asheville_many,aes(x=Date,y=MGD)) +
  geom_line(color= 'darkblue') +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "Water usage data in Asheville 2010-2021",
       y="Withdrawal (mgd)")+
   scale_x_date(date_breaks = "1 year", 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? > Answer: From 2010-2016 it looks like water usage has a slight decrease, but water usage has substantially increased from 2016 until 2021. Overall I would say there is a general increase in water usage over time in Asheville. >