Assignment 10: Data Scraping

Grace Randall

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(rvest)
getwd()
```

```
## [1] "/home/guest/module1/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')
```

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
System_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()

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

Max_day_use <- as.numeric(Max_day_use)
```

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.)

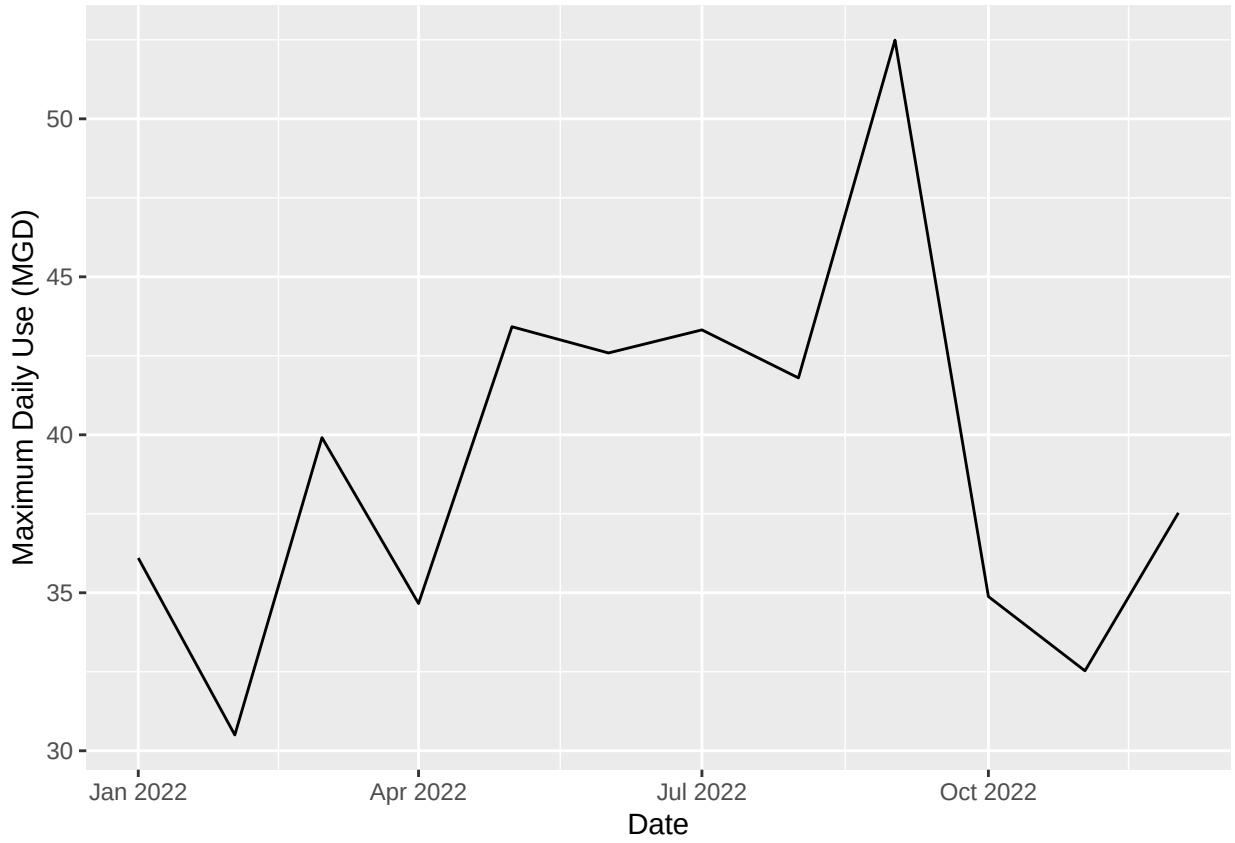
   TIP: Use `rep()` to repeat a value when creating a dataframe.

   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
The_months <-
rbind("Jan", "May", "Sep", "Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug","Dec")

durham_2022 <- data.frame(
  rep(System_name),
  rep(PWSID),
  rep(Ownership),
  Max_day_use,
  The_months)

colnames(durham_2022) <-
  cbind("System_name","PWSID","Ownership", "Max_day_use", "month")
```

```
durham_2022 <- mutate(durham_2022, Date = dmy(paste0("1/",month,"/2022")))
durham_2022 <- durham_2022[order(durham_2022$Date),]

#5
ggplot(durham_2022,aes(Date,Max_day_use))+
  geom_line()+
  labs(x="Date", y="Maximum Daily Use (MGD)")
```



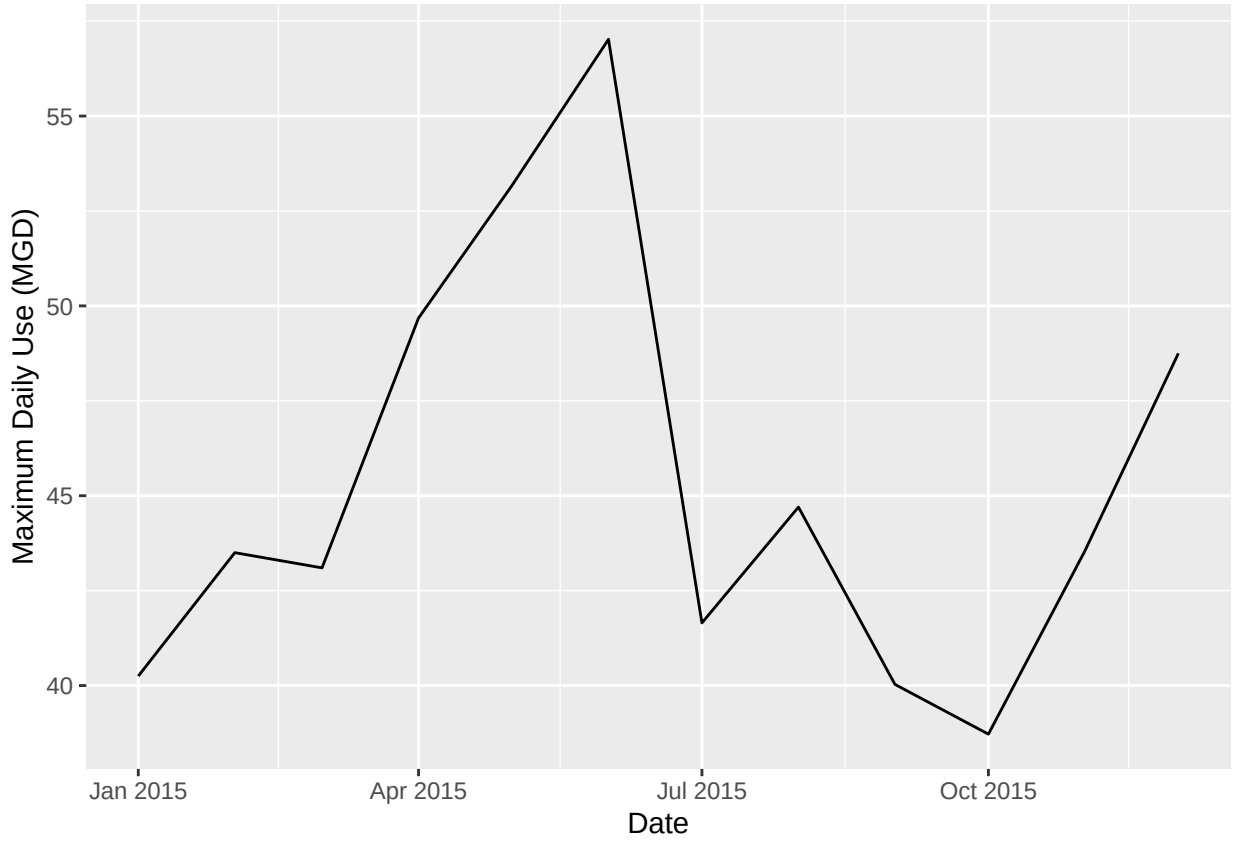
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
scrape.it <- function(the_year, PWSID){

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

  #scrape data
System_name <- the_website  %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

PWSID <- 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_day_use <- the_website  %>%
  html_nodes("th~ td+ td") %>%
  html_text()

Max_day_use <- as.numeric(Max_day_use)

  #Convert to a dataframe
The_months <-
rbind("Jan", "May", "Sep", "Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug","Dec")

df <- data.frame(
  rep(System_name),
  rep(PWSID),
  rep(Ownership),
  Max_day_use,
  The_months)

colnames(df) <- cbind("System_name","PWSID","Ownership", "Max_day_use", "month")
df <- mutate(df, Date = dmy(paste0("1/",month,"/",the_year)))
df <- df[order(df$Date),]

  Sys.sleep(1) #uncomment this if you are doing bulk scraping!

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

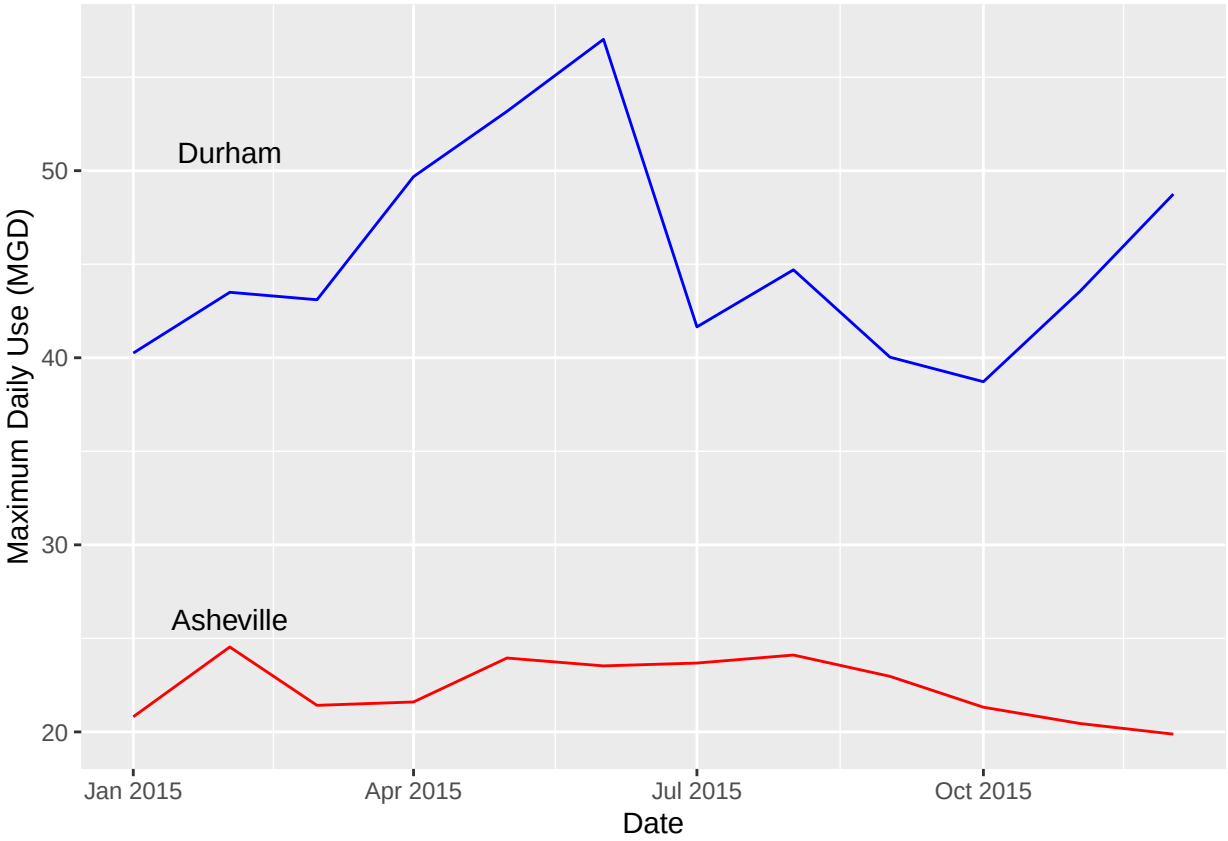
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7
durham_2015 <- scrape.it(2015,'03-32-010')

ggplot(durham_2015,aes(Date,Max_day_use))+
  geom_line()+
  labs(x="Date", y="Maximum Daily Use (MGD)")
```



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.it(2015,'01-11-010')

ggplot()+
  geom_line(data=durham_2015,aes(x=Date,y=Max_day_use),color="blue")+
  geom_line(data=asheville_2015,aes(x=Date,y=Max_day_use),color="red")+
  labs(x="Date", y="Maximum Daily Use (MGD)")+
  annotate(geom = "text",x=ymd("2015-02-01"),y=51,label="Durham")+
  annotate(geom = "text",x=ymd("2015-02-01"),y=26,label="Asheville")
```



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.

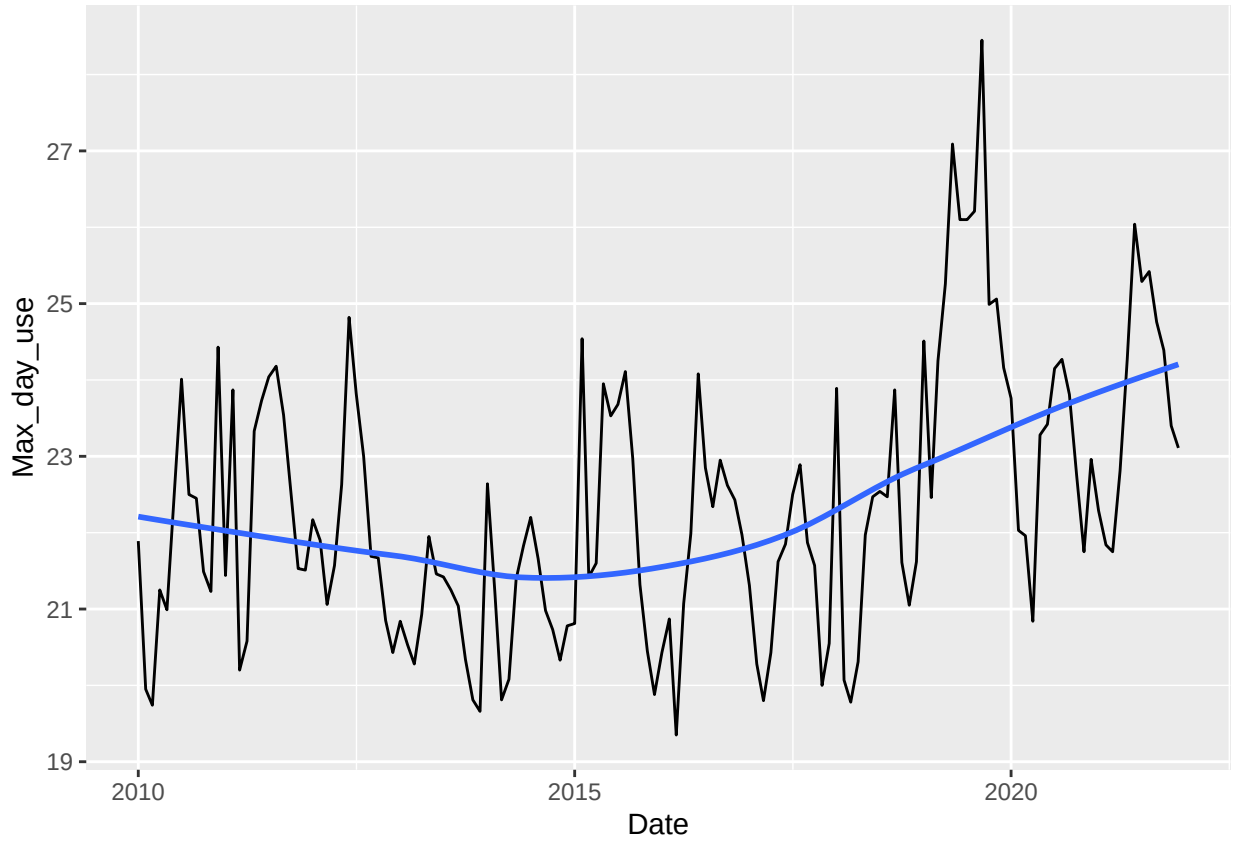
```
#9
the_years = rep(2010:2021)
my_facility = '01-11-010'

multiyear_asheville  <- cross2(the_years,my_facility) %>%
  map(lift(scrape.it)) %>%
  bind_rows()
```

```
## Warning: `cross2()` was deprecated in purrr 1.0.0.
## i Please use `tidyr::expand_grid()` instead.
## i See <https://github.com/tidyverse/purrr/issues/768>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `lift()` was deprecated in purrr 1.0.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
ggplot(multiyear_asheville,aes(Date,Max_day_use))+
  geom_line()+
  geom_smooth(method="loess",se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
  labs(x="Date", y="Maximum Daily Use (MGD)")
```

```
## $x
## [1] "Date"
##
## $y
## [1] "Maximum Daily Use (MGD)"
##
## attr(,"class")
## [1] "labels"
```

Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time? > Answer: >It looks like there was an upward trend in water usage over time afte a dip around 2015.