Assignment 09: Data Scraping

Student Name

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
setwd("~/Desktop/Duke/Data Analytics/Environmental_Data_Analytics_2022/Data")
library(tidyverse)
library(rvest)
library(lubridate)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        legend.title = element_text(size=10),
        legend.text = element_text(size=8),
        plot.title = element_text(hjust = 0.5))

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.)

```
#2
url <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
```

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:
- Max 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 <- url %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

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

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

max.withdrawals.mgd <- url %>%
  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.)

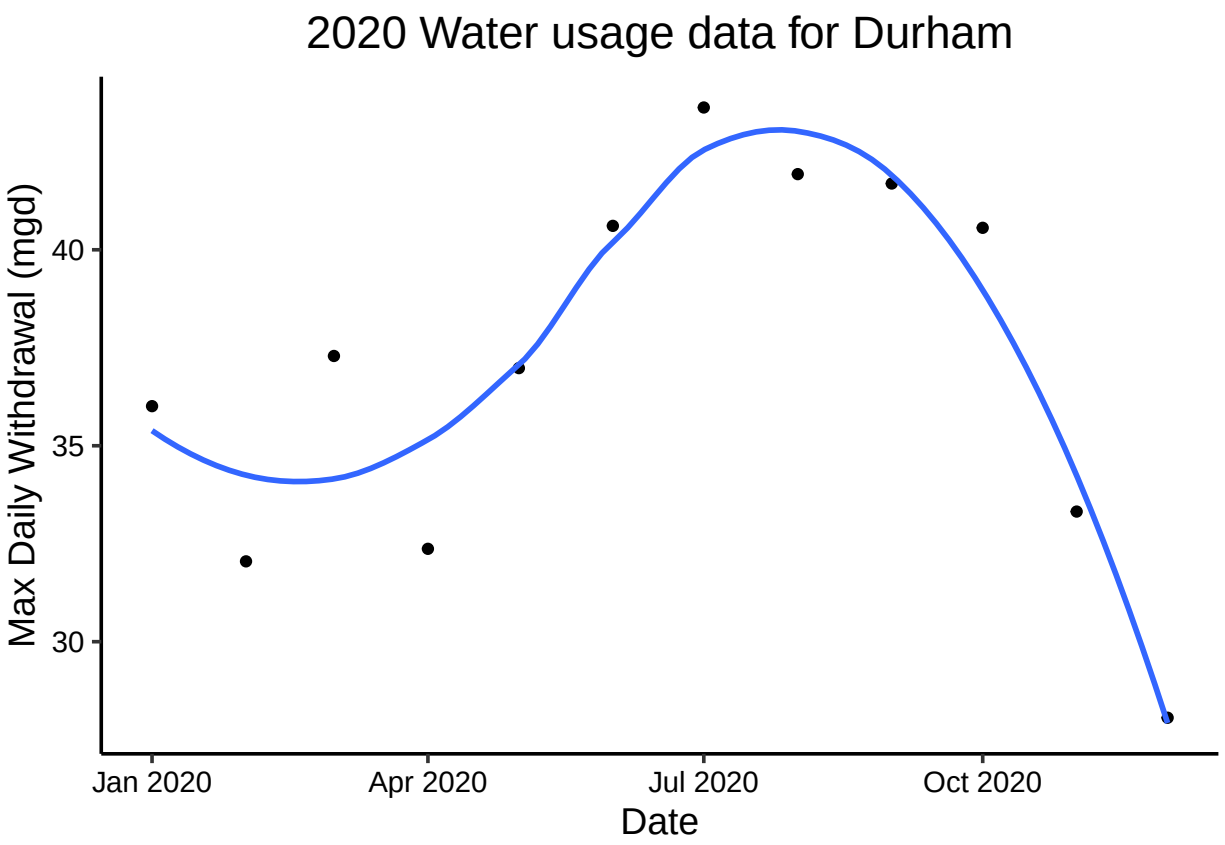
> TIP: Use `rep()` to repeat a value when creating a dataframe.

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
df_withdrawals <- data.frame("Month" = c("Jan", "May", "Sept", "Feb", "Jun",
                                         "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
                             "Year" = 2020,
                             "Name" = water.system.name,
                             "PWSID" = pwsid,
                             "Ownership" = ownership,
                             "Max.daily.withdrawal" = as.numeric(max.withdrawals.mgd))

df_withdrawals <- df_withdrawals %>%
  mutate(Date = my(paste(Month, "-", Year)))

#5
plot_withdrawals <- ggplot(df_withdrawals, aes(x=Date, y=Max.daily.withdrawal)) +
  geom_point() +
  geom_smooth(method="loess", se=FALSE) +
  labs(title = paste("2020 Water usage data for", water.system.name),
       y="Max Daily Withdrawal (mgd)",
       x="Date")
print(plot_withdrawals)
```

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



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**.

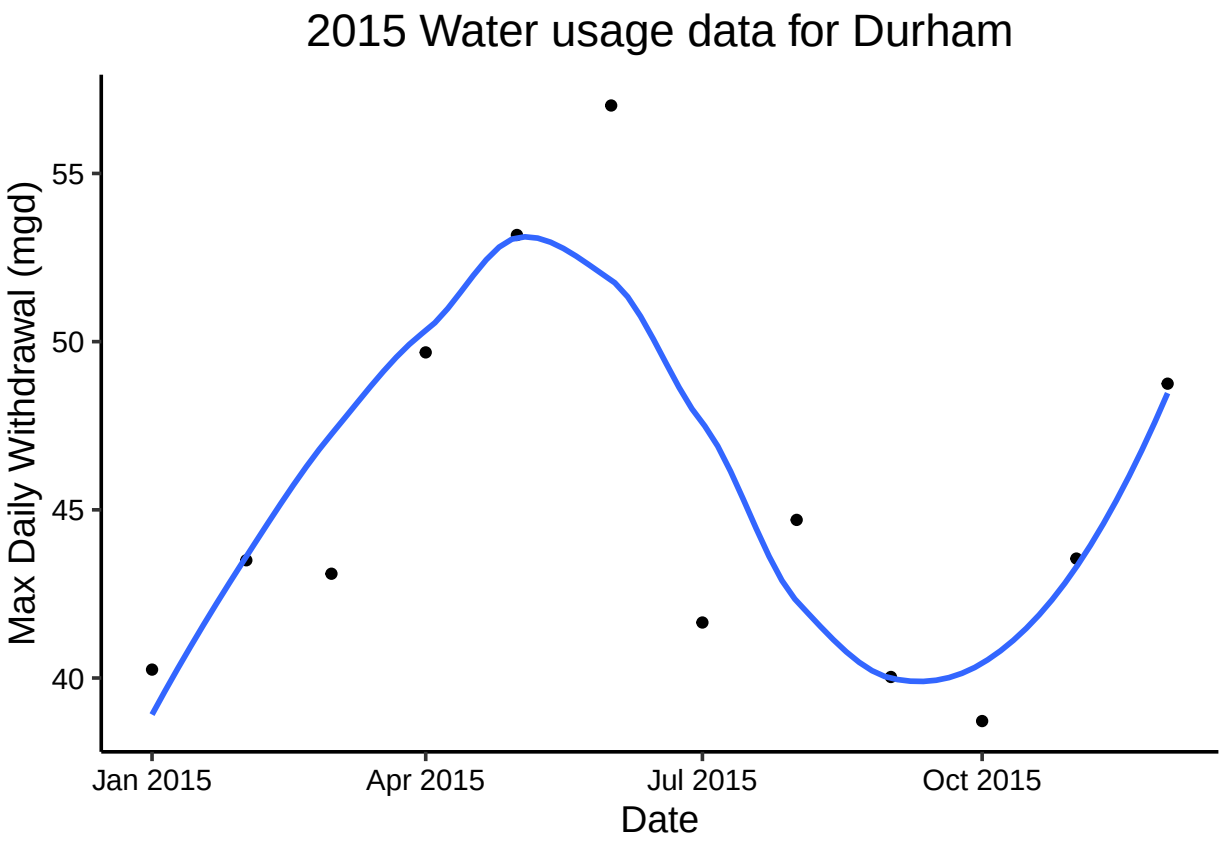
```
#6.
year <- 2015
scrape.it <- function(year, pwsid){
base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
the_scrape_url <- paste0(base_url, pwsid, '&year=', year)
the_website <- read_html(the_scrape_url)

#set element address variables
the_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
the_pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
the_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
the_max_withdrawal_tag <- 'th~ td+ td'

#scrape the data
the_name <- the_website %>% html_nodes(the_name_tag) %>% html_text()
the_pwsid <- the_website %>% html_nodes(the_pwsid_tag) %>% html_text()
the_ownership <- the_website %>% html_nodes(the_ownership_tag) %>% html_text()
the_max_withdrawal <- the_website %>% html_nodes(the_max_withdrawal_tag) %>% html_text()

#Convert to dataframe
df2_withdrawals <- data.frame("Month" = c("Jan", "May", "Sept", "Feb", "Jun",
                                          "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
                              "Year" = 2015,
```

```
                                "Name" = the_name,
                                "PWSID" = the_pwsid,
                                "Ownership" = the_ownership,
                                "Max.daily.withdrawal" = as.numeric(the_max_withdrawal))

df2_withdrawals <- df2_withdrawals %>%
  mutate(Date = my(paste(Month, "-", year)))

#return dataframe
return(df2_withdrawals)
}
```

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

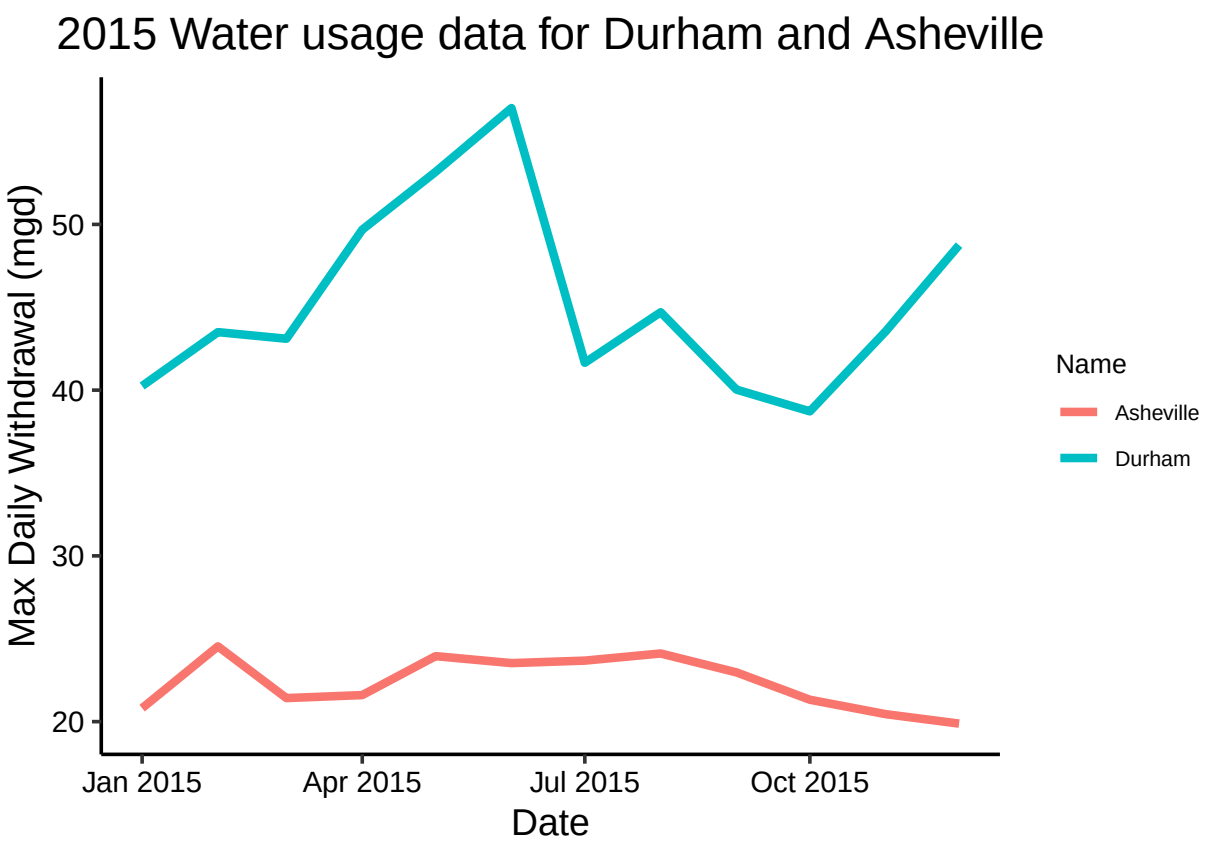
```
#7
Dur2015 <- scrape.it(2015,'03-32-010')

plot2_withdrawals <- ggplot(Dur2015, aes(x=Date, y=Max.daily.withdrawal)) +
  geom_point() +
  geom_smooth(method="loess", se=FALSE) +
  labs(title = paste(year,"Water usage data for", water.system.name),
       y="Max Daily Withdrawal (mgd)",
       x="Date")
print(plot2_withdrawals)
```

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



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
Ash_2015 <- scrape.it(2015, '01-11-010')

#combine dataframes
ash_dur_withdrawal <- rbind(Dur2015, Ash_2015)

#plot comparison
combo_plot <- ggplot(ash_dur_withdrawal, aes(x=Date, y=Max.daily.withdrawal)) +
  geom_line(aes(color=Name), size = 1.5) +
  labs(title = paste(year,"Water usage data for Durham and Asheville"),
       y="Max Daily Withdrawal (mgd)",
       x="Date")
print(combo_plot)
```

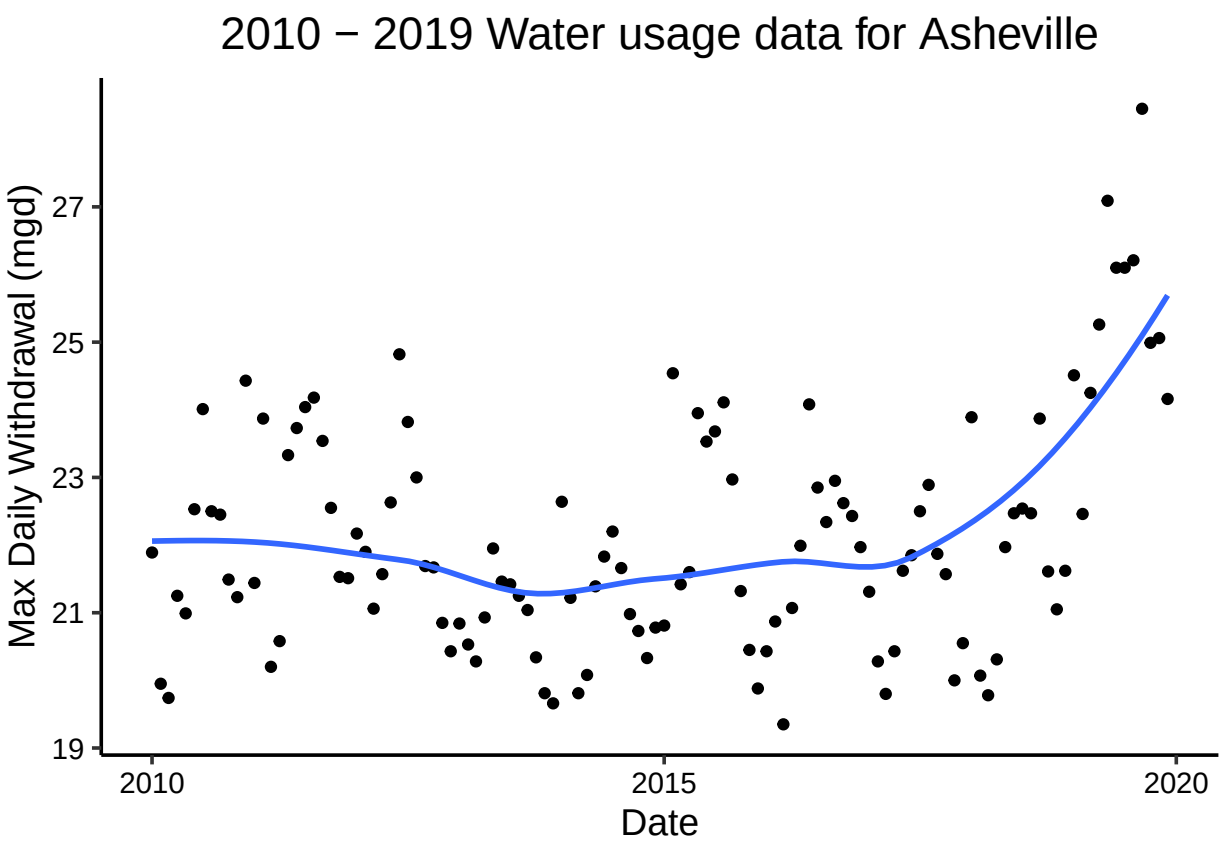


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9
the_years = rep(2010:2019)
pwsid = '01-11-010'

Ash2010to2019 <- map(the_years, scrape.it, pwsid=pwsid)
```

```r
Ash2010to2019 <- bind_rows(Ash2010to2019)

ggplot(Ash2010to2019, aes(x=Date, y=Max.daily.withdrawal)) +
  geom_point() +
  geom_smooth(method="loess", se=FALSE) +
  labs(title = paste("2010 - 2019 Water usage data for Asheville"),
       y="Max Daily Withdrawal (mgd)",
       x="Date")
```

```
## `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: Yes, Asheville has been increasing its water usage overtime from 2010 to 2019.