Assignment 09: Data Scraping

Natasha Jacob

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 Checking working directory and loading required packages
getwd()
```

```
## [1] "/Users/natashajacob/Desktop/EDA872/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
library(rvest)
library(lubridate)

# Setting the ggplot theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
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 Indicating the website as the URL to be scraped
website_URL <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
website_URL
```

```
## {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:
- Average 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 Scraping Water system name
water.system.name <- website_URL %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water.system.name
```

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

```
#Scraping PSWID
pwsid <- website_URL %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
pwsid
```

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

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

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

```
#Scraping Average Daily Use (MGD) for each month
max.withdrawals.mgd <- website_URL %>%
  html_nodes("th~ td+ td , th~ td+ td") %>%
  html_text()
max.withdrawals.mgd
```

```
##  [1] "36.0100" "36.9800" "41.6900" "32.0500" "40.6100" "40.5600" "37.2900"
```

```
##  [8] "43.6300" "33.3200" "32.3700" "41.9300" "28.0600"
```

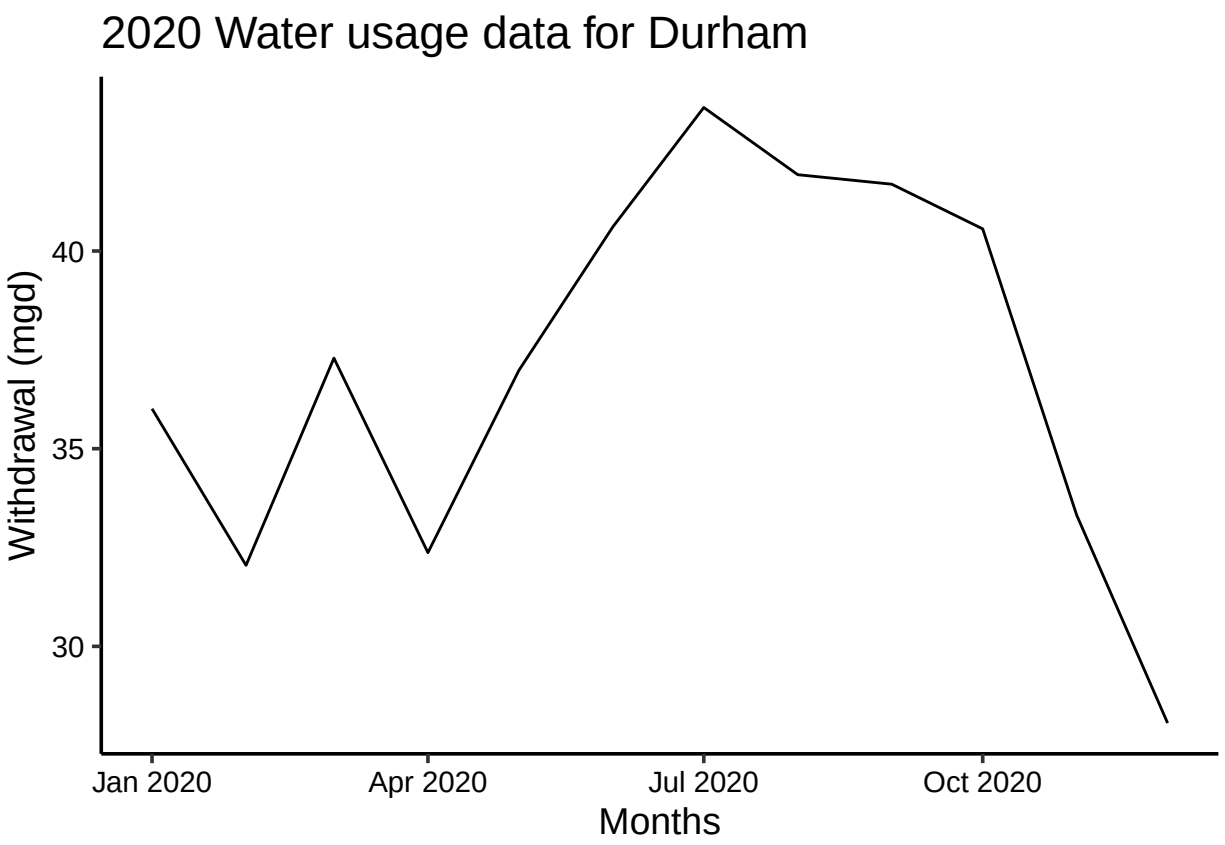
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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

```r
#4 Converting the scraped data into a dataframe
df_nc_water <- data.frame("Year" = rep(2020,12),
                          "Month" = c("Jan", "May", "Sept", "Feb", "June",
                                      "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
                          "Max_withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

# Modifying the data frame

df_nc_water <- df_nc_water %>%
  mutate(Water_system_name = !!water.system.name,
         PWSID = !!pwsid,
         Ownership = !!ownership,
         Date = my(paste(Month,"-",Year)),
         Month = month(Date))

#5 Plotting the max daily withdrawal across the months for 2020

Max_daily_withdrawal_plot <-
  ggplot(df_nc_water, aes(x = Date, y = Max_withdrawals_mgd)) +
  geom_line() +
  labs(title = paste("2020 Water usage data for",water.system.name),
       x = "Months",
       y = "Withdrawal (mgd)") +
  mytheme
print(Max_daily_withdrawal_plot)
```



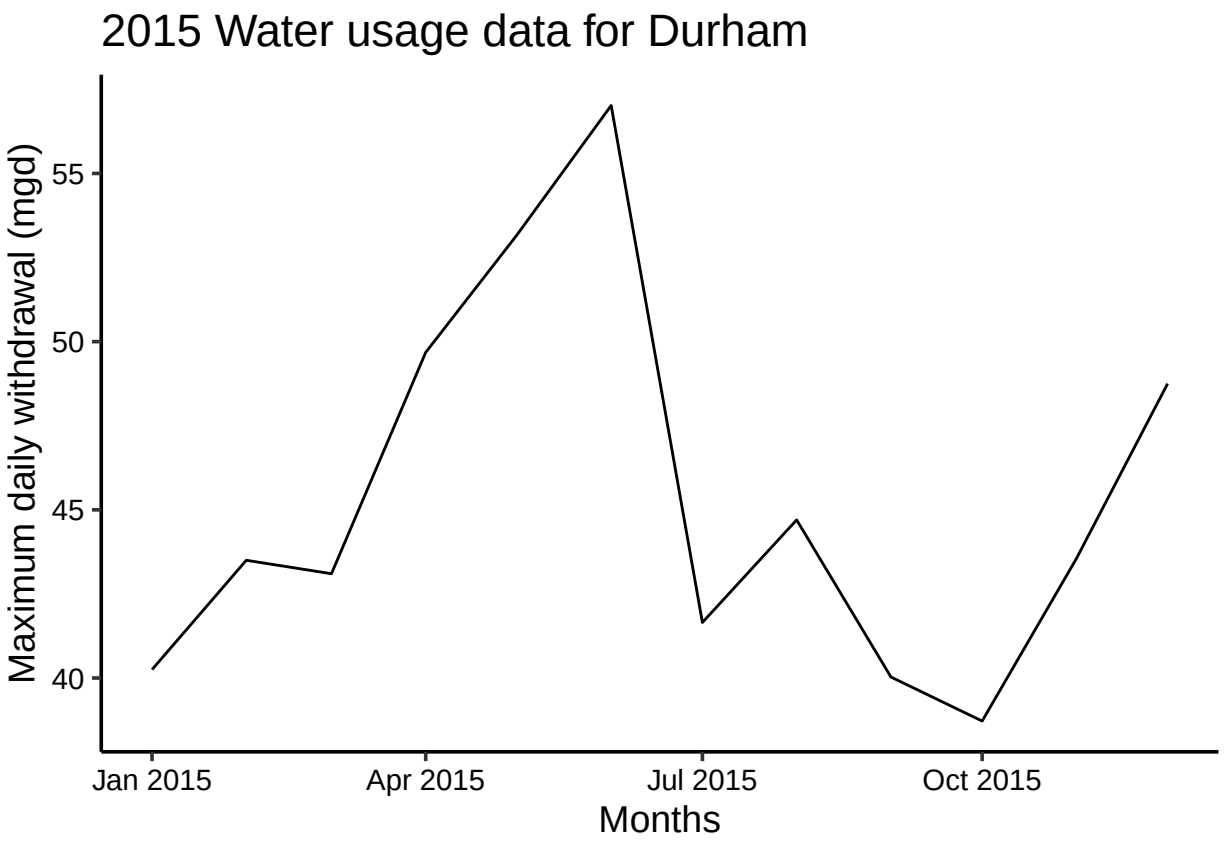
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
#6. Constructing a function for scraping data

the_pwsid <- '03-32-010'
the_year <- 2020


scrape_it <- function(the_year, the_pwsid)
  {
base_url <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                             the_pwsid, "&year=", the_year))


#Set the element address variables (determined in the previous step)
water_system_name_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)'
data_tag <- 'th~ td+ td , th~ td+ td'

#Scrape the data items
the_water_system <- base_url %>%
  html_nodes(water_system_name_tag) %>%
  html_text()

the_pwsid <- base_url %>%
```

```
  html_nodes(pwsid_tag) %>%
  html_text()

the_ownership <- base_url %>%
  html_nodes(ownership_tag) %>%
  html_text()

the_withdrawals <- base_url %>%
  html_nodes(data_tag) %>%
  html_text()

#Construct a dataframe from the scraped data
df_withdrawals <- data.frame("Year" = rep(the_year,12),
                             "Month" = c("Jan", "May", "Sept", "Feb", "June",
                                         "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
                               "Max_withdrawal_mgd" = as.numeric(the_withdrawals)) %>%
  mutate(the_water_system_name = !!the_water_system,
         the_pwsid_name = !!the_pwsid,
         the_ownership_name = !!the_ownership,
         Date = my(paste(Month,"-",Year)),
         Month = month(Date))

return(df_withdrawals)
}
```

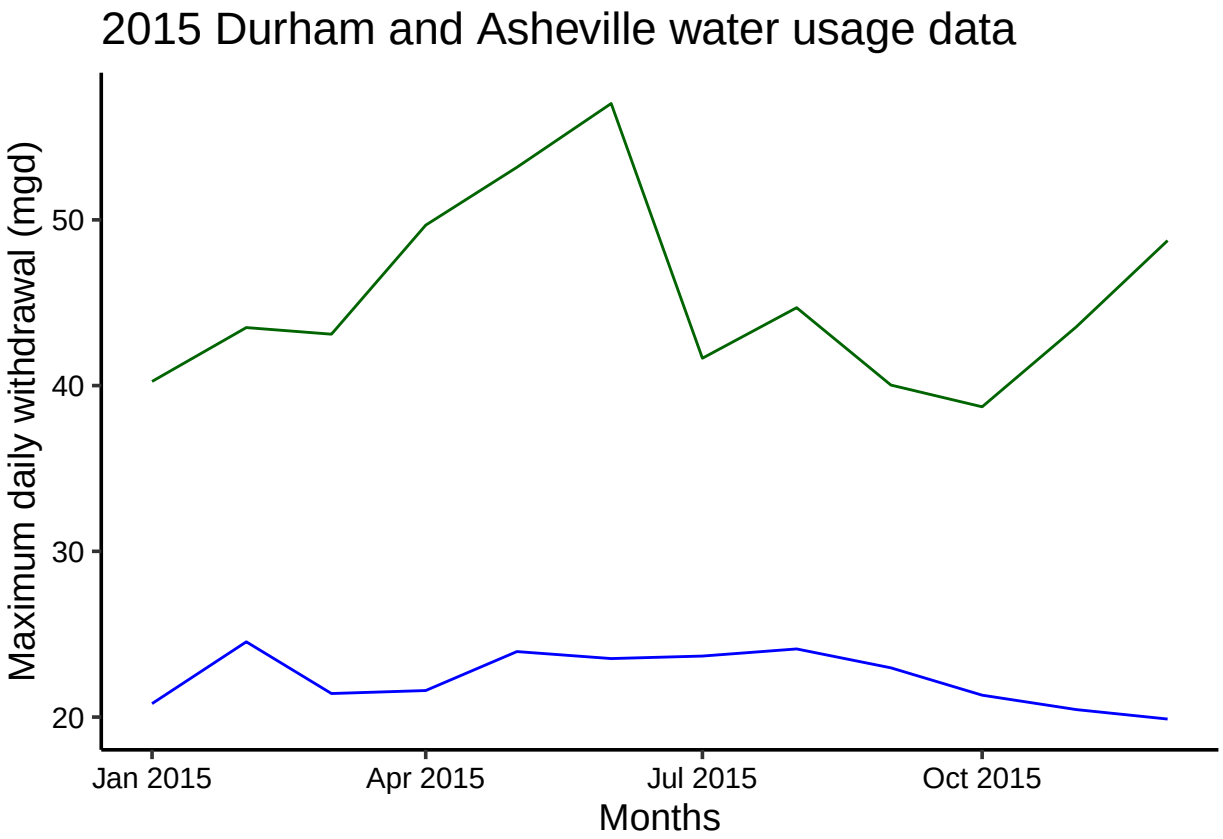
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7 Extracting and plotting max daily withdrawals for Durham (for each month in 2015)

Durham_2015 <- scrape_it(2015,'03-32-010')


Durham_2015_water_usage <-
  ggplot(Durham_2015, aes(x = Date, y = Max_withdrawal_mgd)) +
  geom_line() +
  labs(title = '2015 Water usage data for Durham',
       x = "Months",
       y = "Maximum daily withdrawal (mgd)") +
  mytheme
print(Durham_2015_water_usage)
```



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.

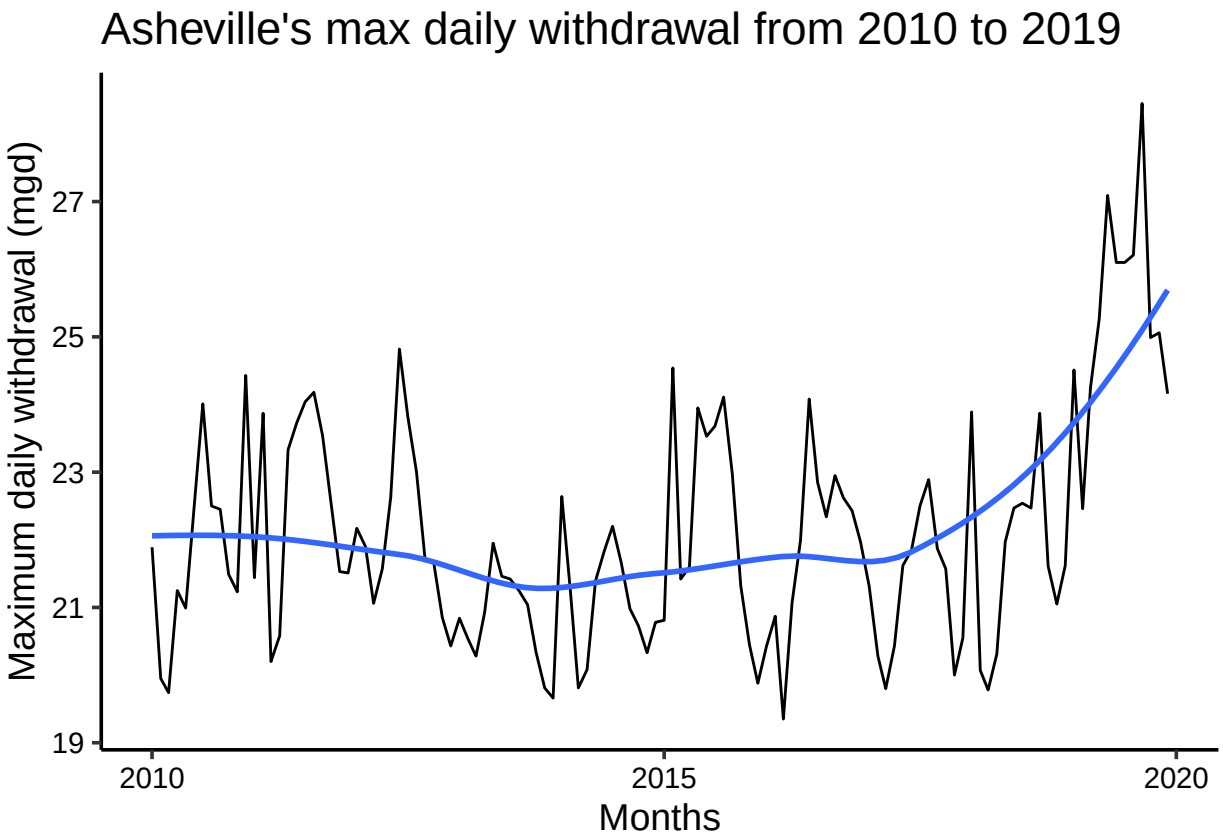
```
#8
Asheville_2015 <- scrape_it(2015,'01-11-010')


Asheville_2015_water_usage <-
  ggplot () +
  geom_line(data = Durham_2015, aes(x = Date, y = Max_withdrawal_mgd), color = "darkgreen") +
  geom_line(data = Asheville_2015, aes(x = Date, y = Max_withdrawal_mgd), color = "blue") +
  scale_color_manual(name = "", breaks = c("darkgreen", "blue"),
                     labels = c("Durham", "Asheville"), guide = "legend") +
  labs(title = '2015 Durham and Asheville water usage data',
       x = "Months",
       y = "Maximum daily withdrawal (mgd)") +
  mytheme
print(Asheville_2015_water_usage)
```



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 <- seq(2010, 2019)
my_pwsid <- '01-11-010'

Asheville2010to2019 <- the_years %>%
  map(scrape_it,my_pwsid <- '01-11-010') %>%
  bind_rows()

Asheville2010to2019_plot <-
ggplot(Asheville2010to2019, aes(x = Date, y = Max_withdrawal_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "Asheville's max daily withdrawal from 2010 to 2019",
       x = "Months",
       y = "Maximum daily withdrawal (mgd)") +
  mytheme
print(Asheville2010to2019_plot)
```

```
## `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?

Yes. An increasing trend in water usage can be observed over a period of time for Asheville.