Assignment 09: Data Scraping

Dori Rathmell

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Rename this file `<FirstLast>_A09_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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

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
getwd()
```

```
## [1] "/home/guest/R/EDA Fall/Assignments"
```

```
library(tidyverse)
library(rvest)
library(lubridate)
#loading necessary packages
gg_theme <- theme_gray(base_size = 12) +
  theme(axis.text = element_text(color = "Gray"),
        legend.position = "right")
theme_set(gg_theme)
#setting my theme
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2021 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=2021

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=2021')
#loading the website
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
- PSWID
- Ownership
- From the "3. Water Supply Sources" section:
- Maximum Daily 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), with the first value being "27.6400".

```
#3
water.system.name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
  #scraping the water system name
pswid <- webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
  #scraping the pswid
ownership <- webpage %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
  #scraping the ownership
max.withdrawals.mgd <- webpage %>%
  html_nodes('th~ td+ td , th~ td+ td') %>%
  html_text()
#scraping the actual withdrawal values
```

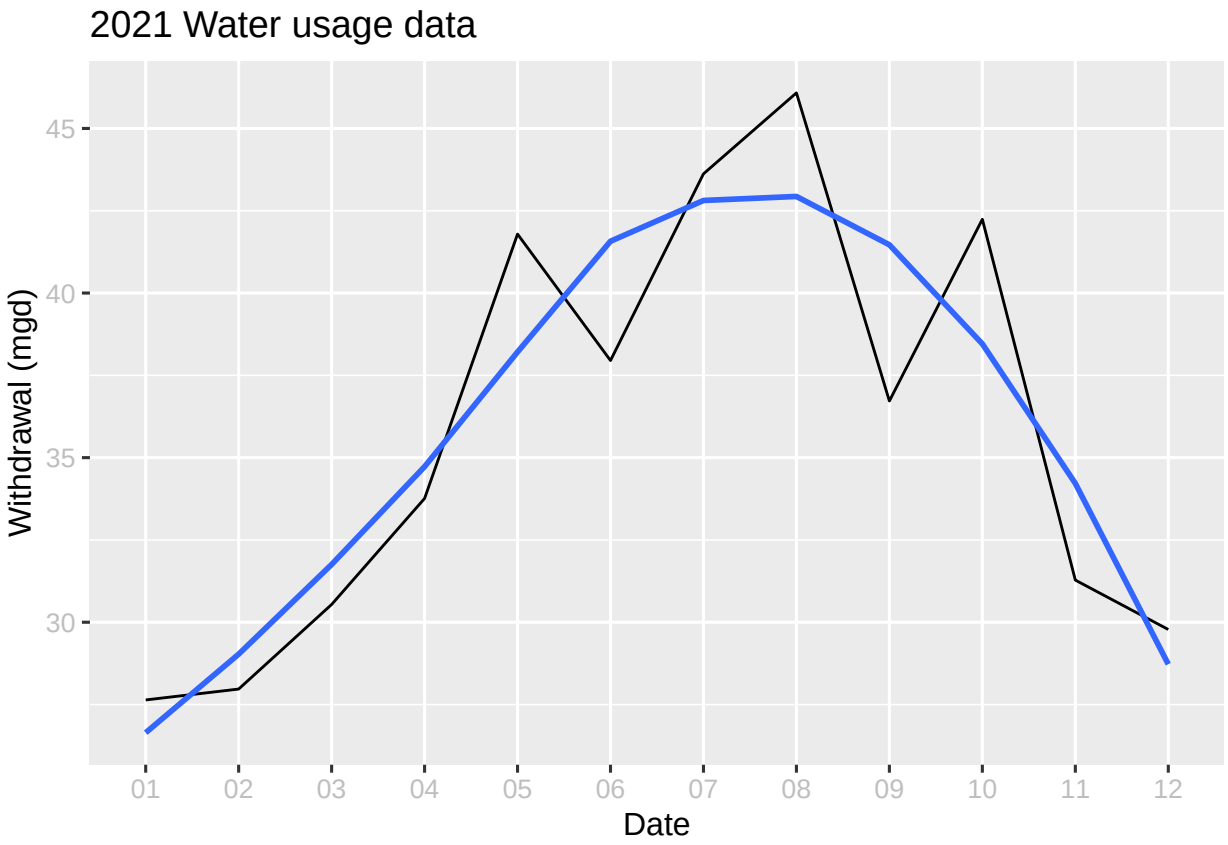
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   > 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...

5. Create a line plot of the maximum daily withdrawals across the months for 2021

```
#4
Month <- c('01','05','09','02','06','10','03','07','11','04','08','12')
#because the max withdrawals loaded out of order, I manually set the months to the order in which the d
scraping.df <- data.frame(Month = (Month),
                          Year = rep(2021,12),
                              Max.withdrawals = as.numeric(max.withdrawals.mgd)
                          )
#scraping the initial data into a dataframe

scraping.df <-scraping.df %>%
  mutate(PWSID = !!pswid,
         Ownership = !!ownership,
         System.Name = !!water.system.name,
         Date = (paste0(Month,"-",Year)))
#mutating the dataframe to add the additional information



#5


  Scrapedplot1<-ggplot(scraping.df,aes(x=Month,y=Max.withdrawals, group=1)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
    theme_set(gg_theme)+
  labs(title = paste("2021 Water usage data"),
       y="Withdrawal (mgd)",
       x="Date")
#creating a plot of the max withdrawals by month in Durham
print(Scrapedplot1)
```

```
## `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 (pwsid) scraped**.

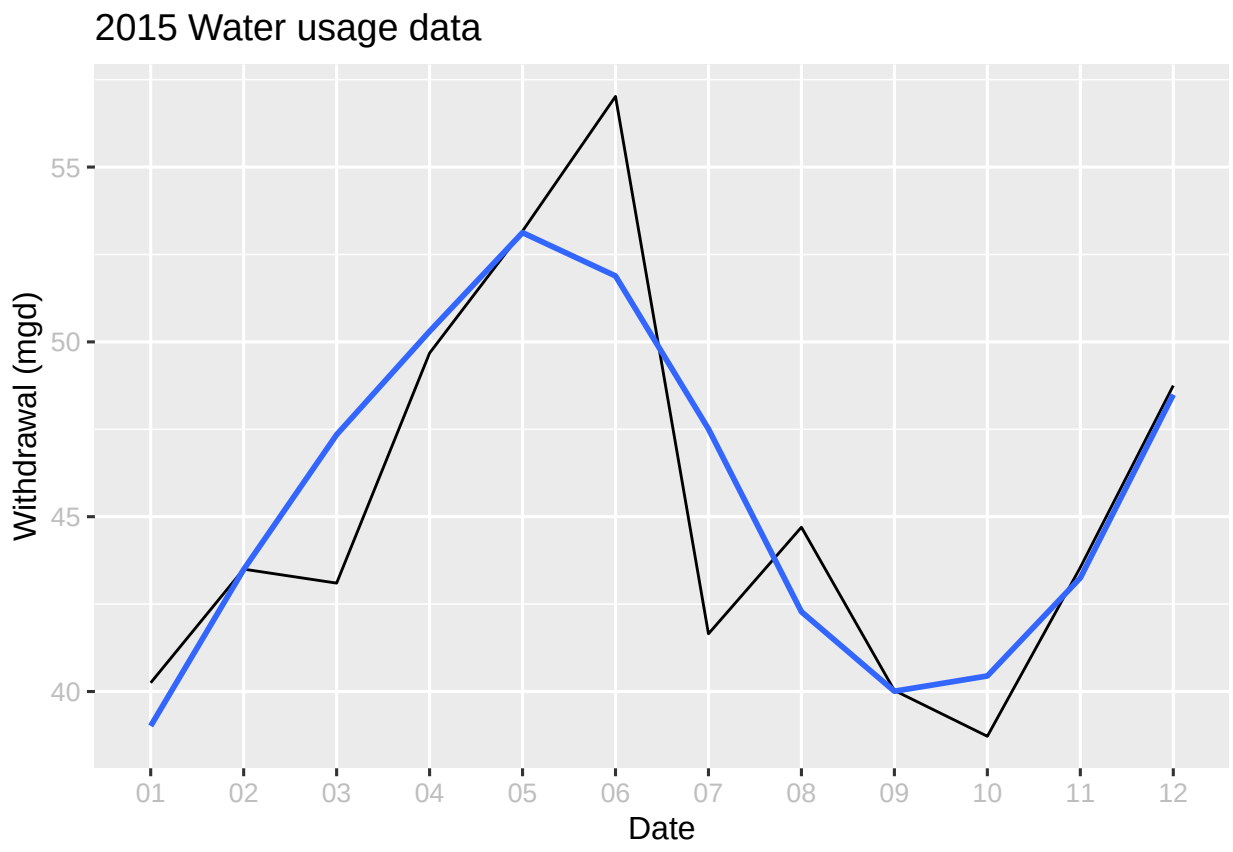
```
#6.

scraped_function <- function(the_pwsid, Year){
  #setting the parameters of the function
  scraped_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',the_pws
  #setting the url so the pwsid and year change as the inputs
  water.system.tag<- "div+ table tr:nth-child(1) td:nth-child(2)"
  ownership.tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
  max.withdrawal.tag <- "th~ td+ td , th~ td+ td"
  pwsid.tag <- "th~ td+ td"
  #assigning the tags to a variable
  the_water_system <- scraped_website %>% html_nodes(water.system.tag) %>% html_text()
  the_ownership <- scraped_website %>%  html_nodes(ownership.tag) %>%  html_text()
  the_max.withdrawal <- scraped_website %>% html_nodes(max.withdrawal.tag) %>% html_text()
  pwsid<- scraped_website %>% html_nodes(pwsid.tag) %>% html_text()
  #scraping the data from the websites within the function
  scraping.df <- data.frame(Month = (Month),
                            Year = Year,
                            Max.withdrawals = as.numeric(the_max.withdrawal),
                            Date = paste0(Month,"-",Year)) %>%
    mutate(PWSID = !!pwsid,
         Ownership = !!the_ownership,
         System.Name = !!the_water_system)
```

```
  #making the dataframe within the function as the end result of the function

return(scraping.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.scrape <- scraped_function(the_pwsid = '03-32-010', Year=2015)
#adjusting the inputs to the function so the function draws data from durham in 2015
  durhamScrapePlot<-ggplot(Durham.scrape,aes(x=Month,y=Max.withdrawals, group=1)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
    theme_set(gg_theme)+
  labs(title = paste("2015 Water usage data"),
       y="Withdrawal (mgd)",
       x="Date")
#creating a plot of the 2015 data
print(durhamScrapePlot)
```

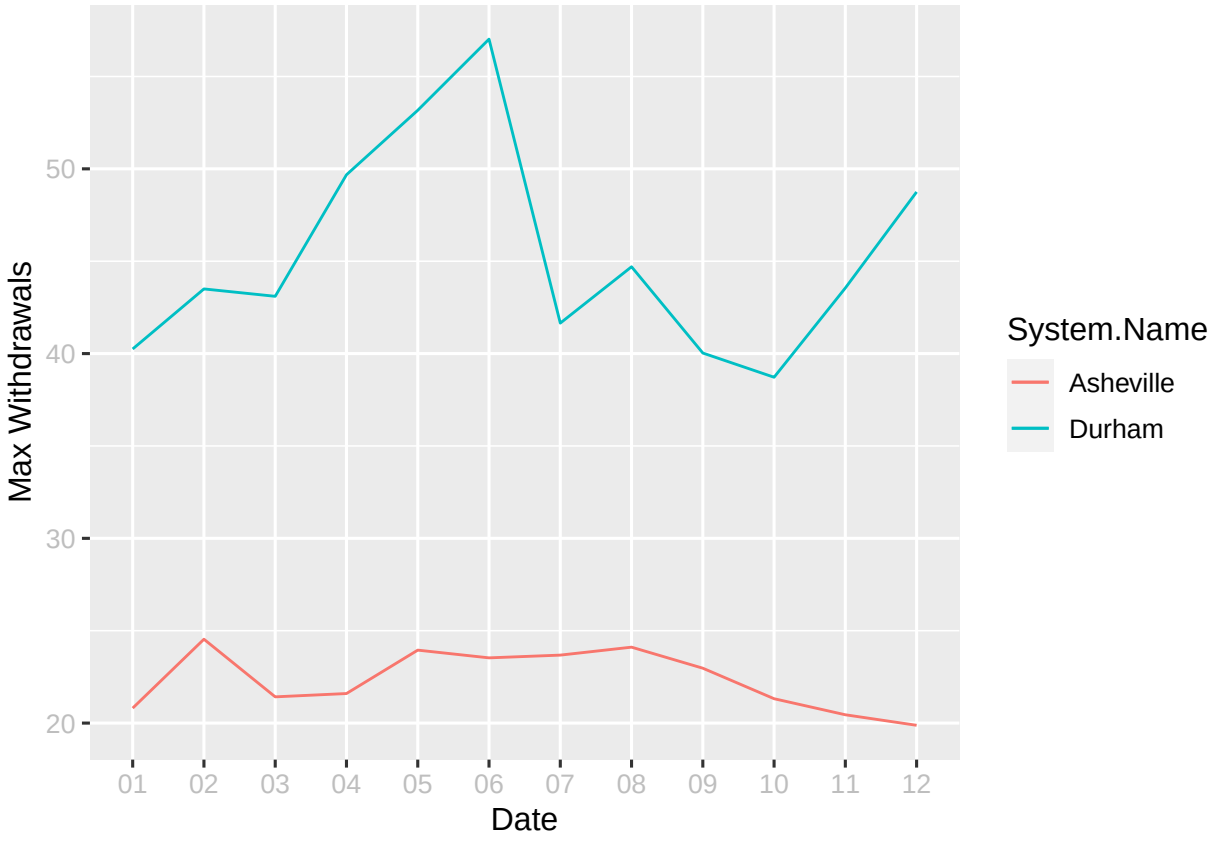
```
## `geom_smooth()` using formula 'y ~ x'
```



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 Asheville's to Durham's water withdrawals.

```
#8
Asheville.scrape <- scraped_function(the_pwsid = '01-11-010', Year=2015)
```

```r
#applying the asheville parameters to create a new dataframe using the earlier function

Asheville.durham.plot <- ggplot()+
  geom_line(data=Asheville.scrape, aes(x=Month, y=Max.withdrawals, color = System.Name, group=1))+
  geom_line(data=Durham.scrape, aes(x=Month, y=Max.withdrawals, color = System.Name, group=1))+
  theme_set(gg_theme)+
  xlab('Date')+
  ylab('Max Withdrawals')
#creating the plot comparing asheville and durham
print(Asheville.durham.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.

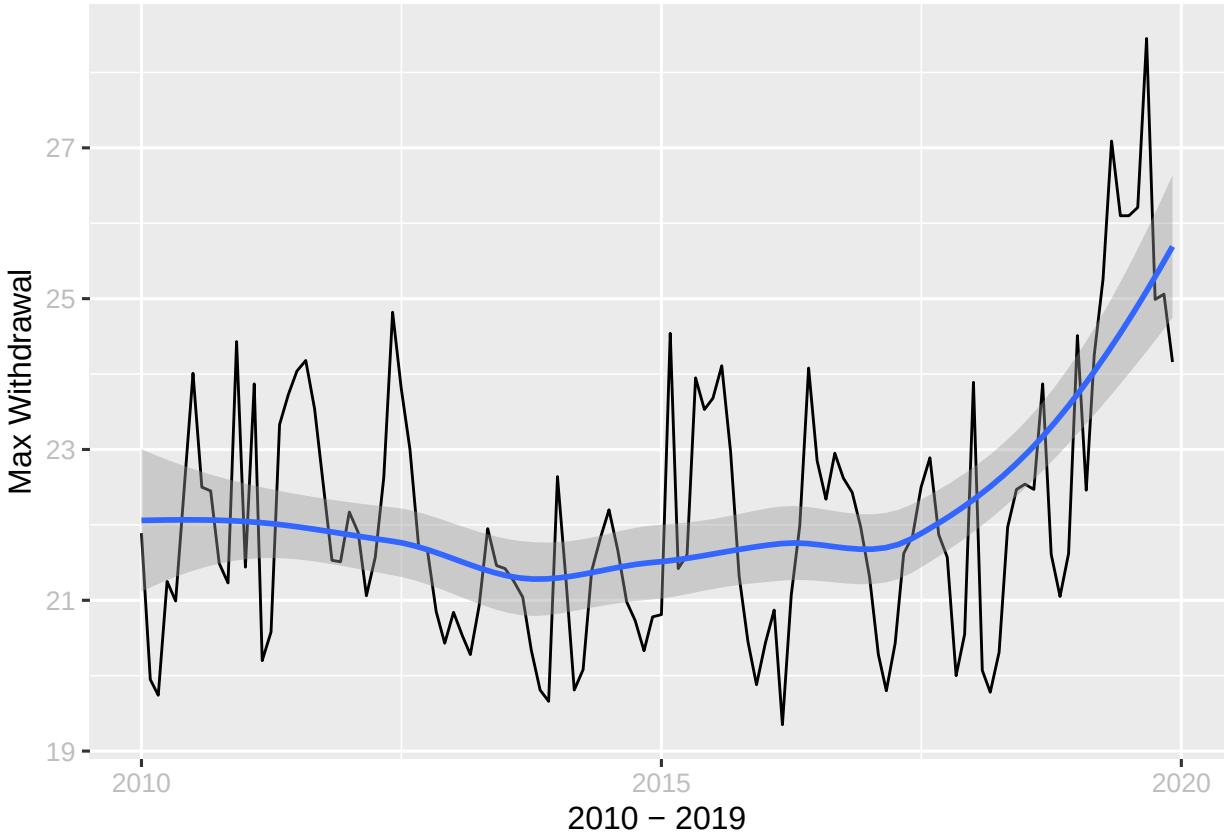
   TIP: See Section 3.2 in the "09_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```r
#9
Last.scrape <- map2("01-11-010", rep(2010:2019), scraped_function)
Asheville.last.scrape <- bind_rows(Last.scrape)
Asheville.last.scrape$new.date <- my(Asheville.last.scrape$Date)
#using the map2 function and bind_rows function to create a dataframe that collects data from 2010-2019
view(Asheville.last.scrape)

class(Asheville.last.scrape$Date)
```

```
## [1] "character"
```

```
final.plot <- ggplot(data=Asheville.last.scrape, aes(x=new.date, y=Max.withdrawals, group=1))+
  geom_line()+
  geom_smooth()+
  theme_set(gg_theme)+
  ylab('Max Withdrawal')+
  xlab('2010 - 2019')
#creating the last plot.
print(final.plot)
```

```
## `geom_smooth()` using method = 'loess' and 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, up until about 2017 the maximum withdrawals were relatively stable but after 2017 the maximum withdrawals experienced an observable increase.