Assignment 09: Data Scraping

Taro Katayama

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
getwd()
```

```
## [1] "/Users/tarokatayama/Desktop/Duke_Semester_2/Environmental_data_analytics/R_Projects/Environmenta
```

```
library(tidyverse)
library(rvest)
library(lubridate)
Taro_Theme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
theme_set(Taro_Theme)
```

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
the_URL<- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
the_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
- PSWID
- 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
water.system.name <- the_URL%>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)")%>%
  html_text()
water.system.name
```

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

```
pswid <- the_URL%>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)")%>%
  html_text()
pswid
```

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

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

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

```
max.withdrawals.mgd <- the_URL%>%
  html_nodes("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"
```

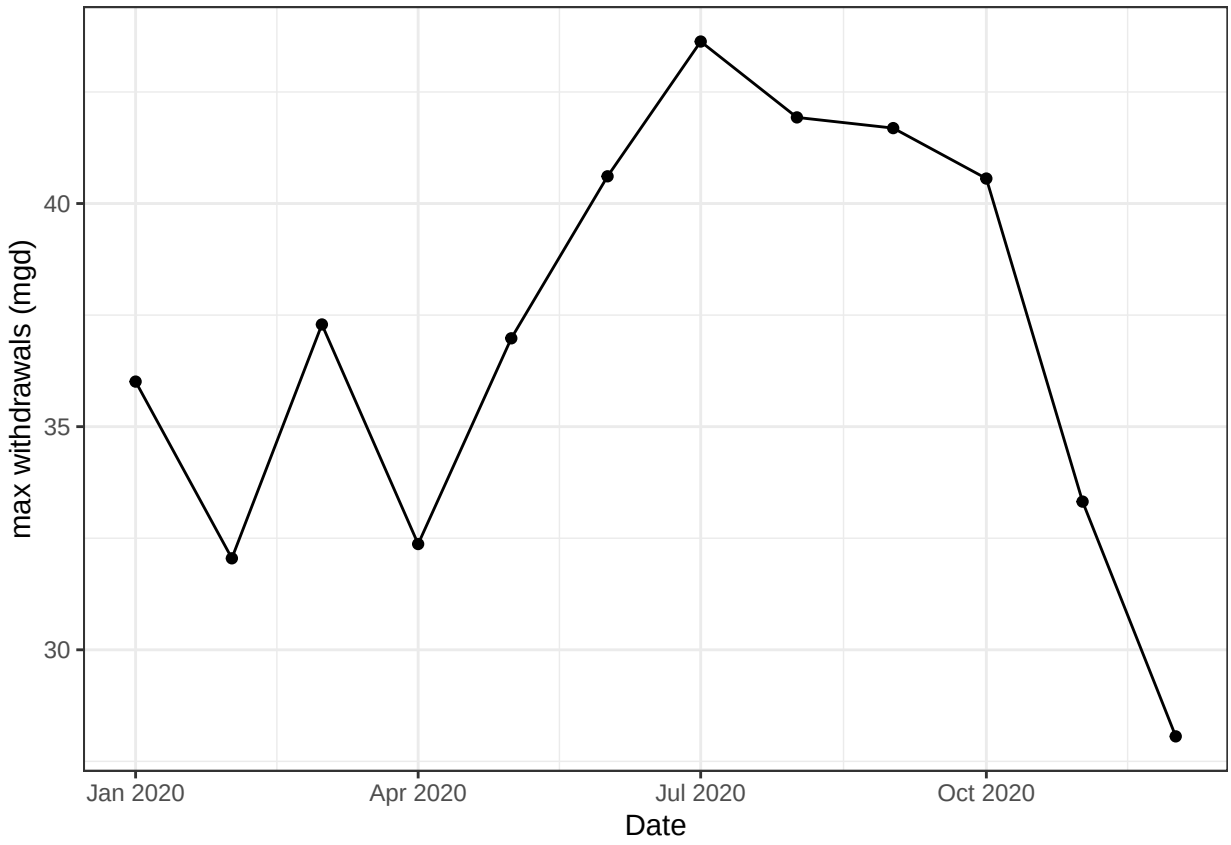
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

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

```
#4
Month= c("01", "05", "09", "02", "06", "10", "03", "07", "11", "04", "08", "12")


df_dirty_durham<- data.frame("Month"=Month,
                             "Year"= rep(2020,12),
                             "max.withdrawals"= as.numeric(max.withdrawals.mgd))

df_dirty_durham<- df_dirty_durham%>%
  mutate(water.system=!!water.system.name,
         PWSID=!!pswid,
         Ownership=!!ownership,
         Date= my(paste0(Month,"-",Year)))

#5
ggplot(df_dirty_durham)+
  geom_line(aes(x=Date, y=max.withdrawals))+
  geom_point(aes(x=Date, y=max.withdrawals))+
  labs(y="max withdrawals (mgd)")+
  theme_bw()
```



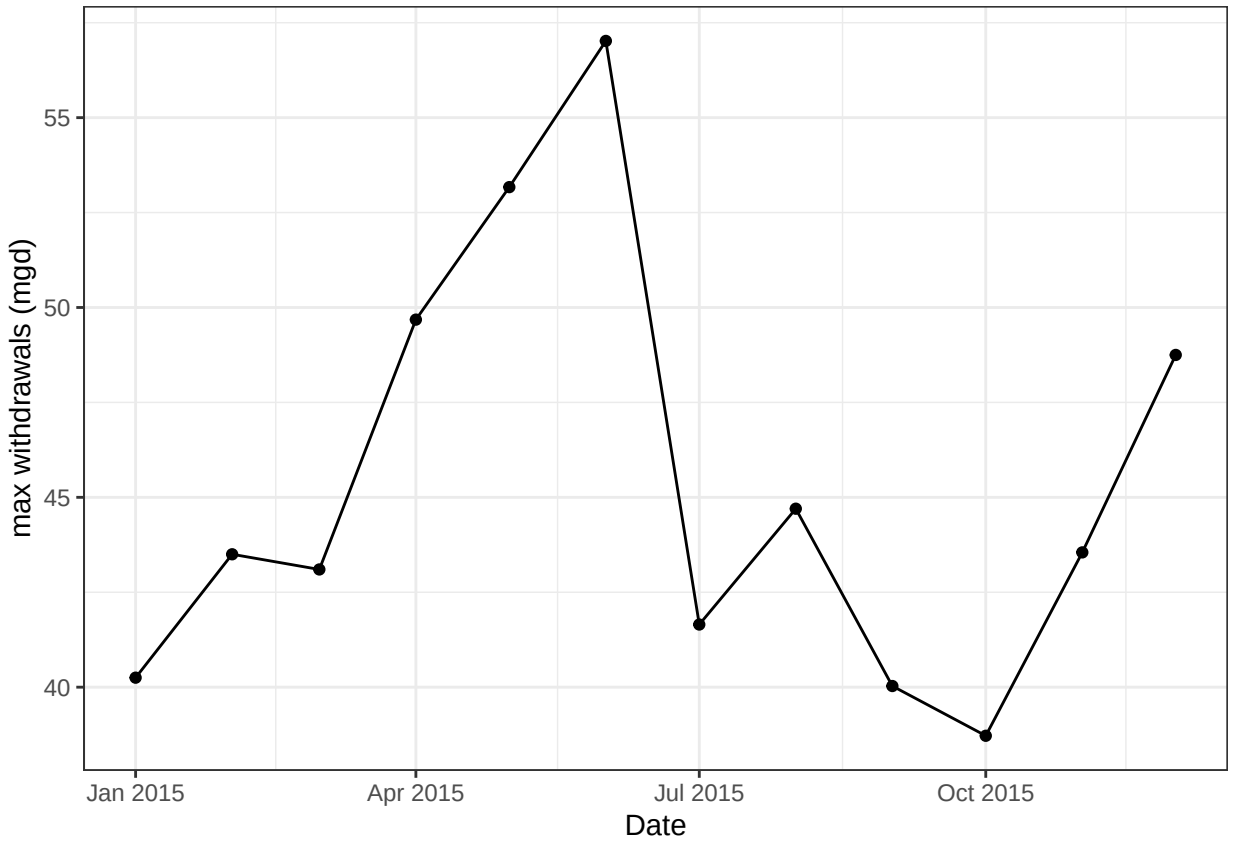
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site scraped**.

```
#6.
scrape_it<- function(the_year, the_pwsid){
  scrapey_scrape<- read_html(
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?',
           'pwsid=', the_pwsid, '&year=',the_year))


  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)'
  max.withdrawal.tag<- 'th~ td+ td'

  Watersystem<- scrapey_scrape%>%
  html_nodes(water.system.tag)%>%
  html_text()

  Pwsid<- scrapey_scrape%>%
  html_nodes(pwsid.tag)%>%
  html_text()

  Ownership<- scrapey_scrape%>%
  html_nodes(ownership.tag)%>%
  html_text()

  MaxWithdrawal<- scrapey_scrape%>%
  html_nodes(max.withdrawal.tag)%>%
  html_text()

  df_<- data.frame("Month"=Month,
                             "Year"= rep(the_year,12),
                             "max.withdrawals"= as.numeric(MaxWithdrawal))

  df_<- df_%>%
  mutate(water.system=!!Watersystem,
         PWSID=!!Pwsid,
         Ownership=!!Ownership,
         Date= my(paste0(Month,"-",Year)))

  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

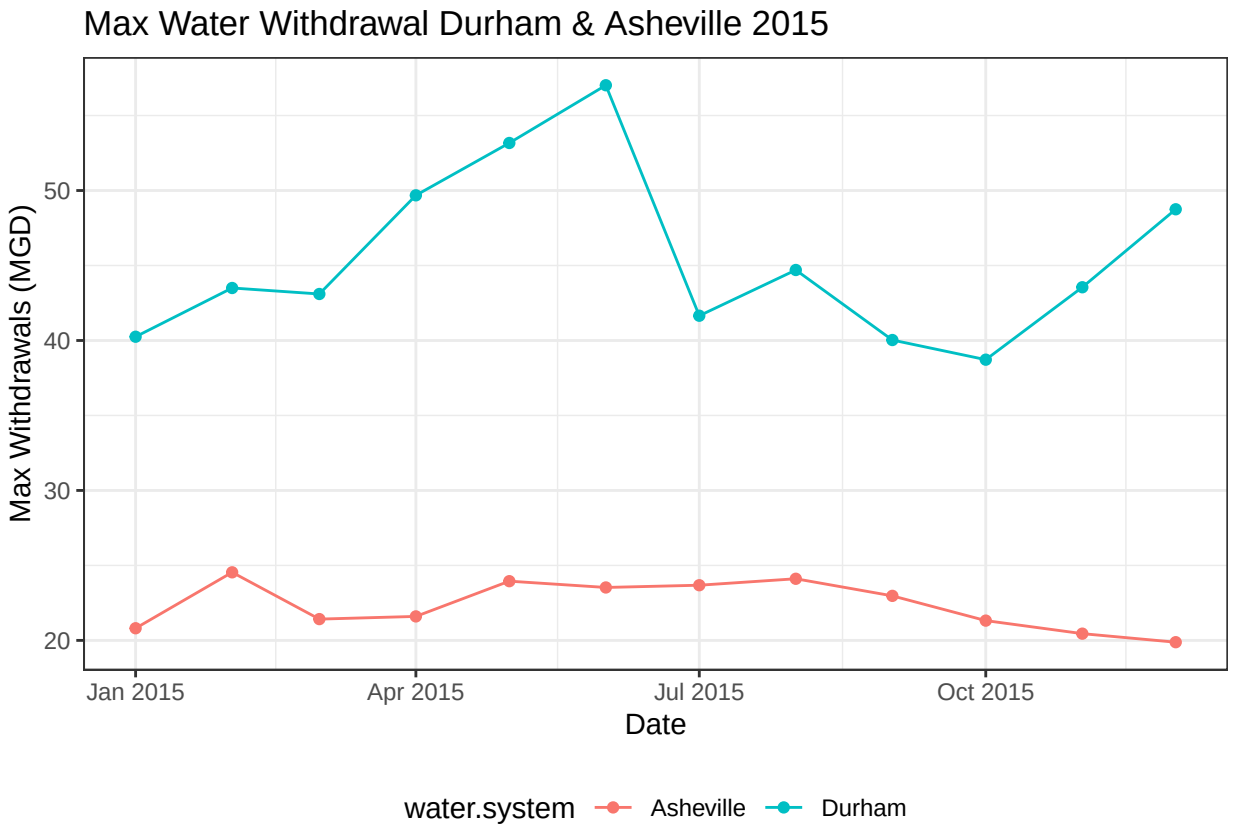
```
#7
durham2015<- scrape_it(2015,'03-32-010')
view(durham2015)

ggplot(durham2015)+
  geom_line(aes(x=Date, y=max.withdrawals))+
  geom_point(aes(x=Date, y=max.withdrawals))+
```

```
  labs(y="max withdrawals (mgd)")+
  theme_bw()
```



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.

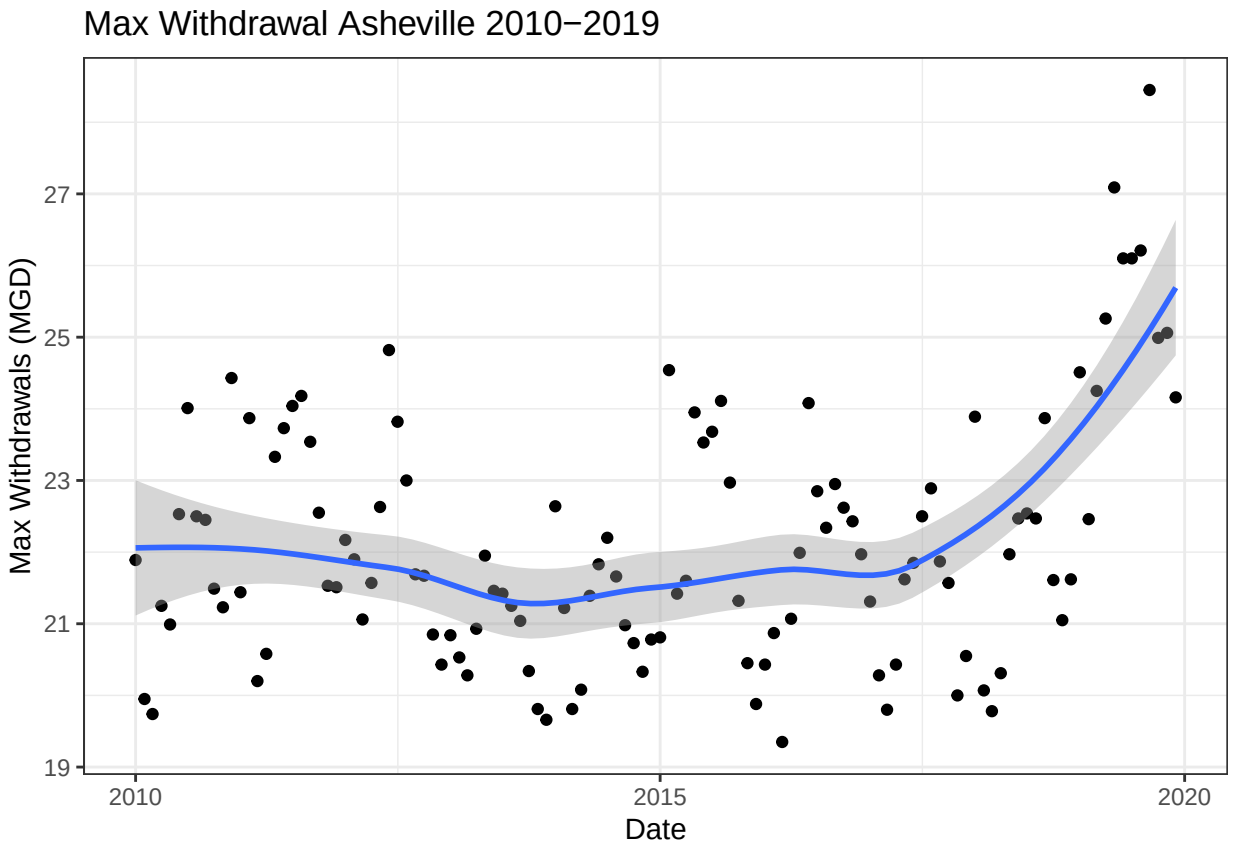
```
#8

Asheville2015<- scrape_it(2015, '01-11-010')

df_combined<-rbind(Asheville2015, durham2015)

ggplot(df_combined)+
  geom_point(aes(x=Date, y=max.withdrawals, color=water.system))+
  geom_line(aes(x=Date, y=max.withdrawals, color=water.system))+
  labs(y="Max Withdrawals (MGD)",
       title = "Max Water Withdrawal Durham & Asheville 2015")+
  theme_bw()+
  theme(legend.position="bottom")
```



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_year = seq(2010,2019)
the_pwsid = '01-11-010'

Asheville2010to2019<- the_year%>%
  map(scrape_it, the_pwsid ='01-11-010')%>%
  bind_rows()

ggplot(Asheville2010to2019)+
  geom_point(aes(x=Date, y=max.withdrawals))+
  geom_smooth(aes(x=Date, y=max.withdrawals))+
  labs(y="Max Withdrawals (MGD)",
       title = "Max Withdrawal Asheville 2010-2019")+
  theme_bw()+
  theme(legend.position="bottom")
```

```
## `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. In the last couple of years, from around 2017 onward, Asheville has an increased max withrdrawal per month. Pre-2017, there was not a noticeable visual trend.