Assignment 09: Data Scraping

Cassidy White

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] "C:/Users/cassi/OneDrive - Duke University/Documents/School/Grad School/Spring 2022/Environmental
```

```
setwd("C:/Users/cassi/OneDrive - Duke University/Documents/School/Grad School/Spring 2022/Environmental
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.1.3
```

```
library(ggplot2)
library(lubridate)
library(dplyr)

mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
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
the_website<-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:
- 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_website %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>% html_
pswid <- 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.withdrawals.mgd <-  the_website %>% 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. . .

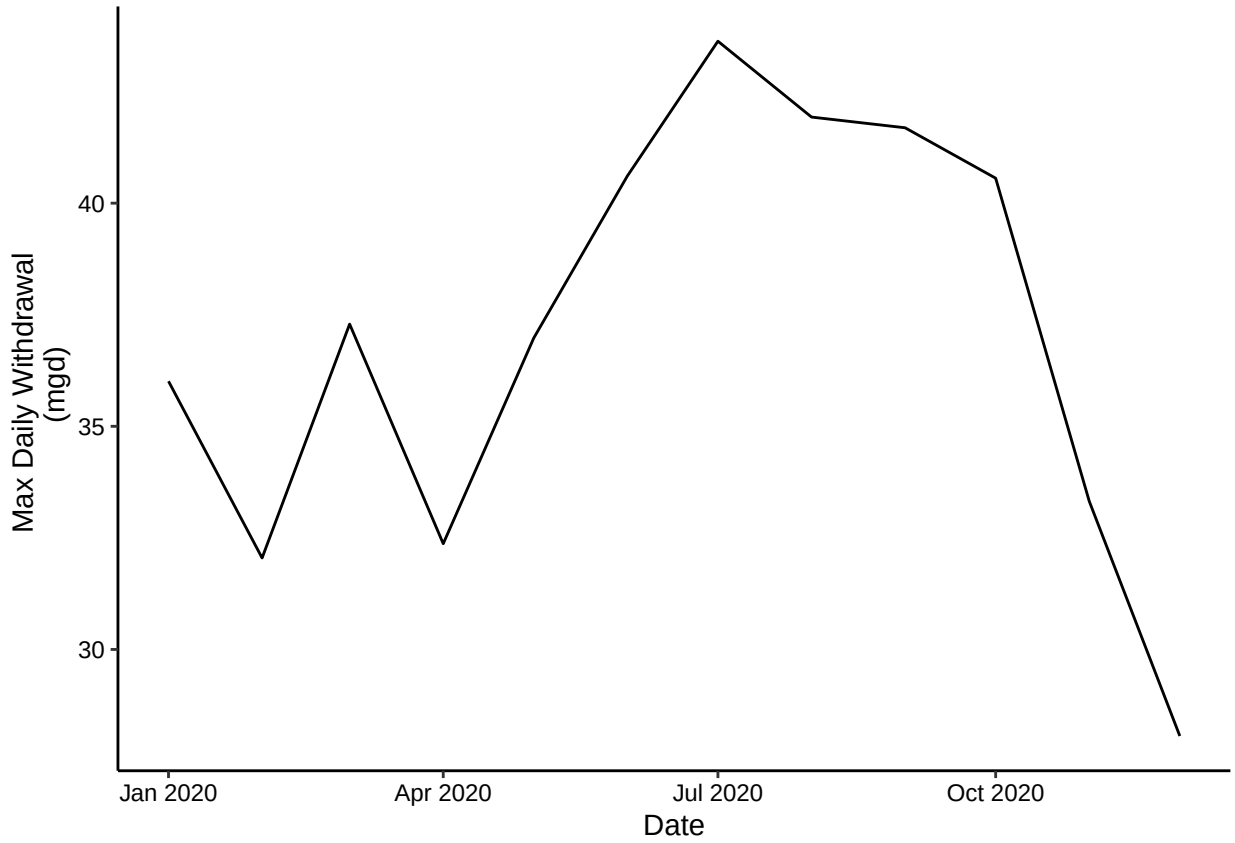
5. Plot the max daily withdrawals across the months for 2020

```
#4
withdrawals_df<-data.frame("Month" = c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "
                           "Year" = rep(2020,12),
                           "Max Daily Withdrawals" = as.numeric(max.withdrawals.mgd)) %>%
                             mutate(System.Name = !! water.system.name,
                           PSWID = !! pswid,
                           Ownership = !! ownership,
                           Date = my(paste(Month,"-",Year)))
```

```
#withdrawals_df<-subset(withdrawals_df, select = -c(Month, Year))
colnames(withdrawals_df)
```

```
## [1] "Month"                 "Year"                  "Max.Daily.Withdrawals"
## [4] "System.Name"           "PSWID"                 "Ownership"
## [7] "Date"
```

```
#5
ggplot(withdrawals_df, aes(x=Date, y= Max.Daily.Withdrawals))+
  geom_line()+
  ylab("Max Daily Withdrawal \n (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 scraped**.

```
#6.

scrape.it <- function(the_year, the_pswid){
   #retrieve website
   the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?','pwsid=',the_psw

   #scrape the data
   water.system.name <- the_website %>% html_nodes('div+ table tr:nth-child(1) td:nth-child(2)') %>% ht
   pswid <- the_website %>% html_nodes('td tr:nth-child(1) td:nth-child(5)') %>% html_text()
   max.withdrawals.mgd <-  the_website %>% html_nodes('th~ td+ td') %>% html_text()
   ownership <- the_website %>% html_nodes('div+ table tr:nth-child(2) td:nth-child(4)') %>% html_text(
```

```
   #convert to data frame
   withdrawals_function_df<-data.frame(
    "Month" = c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
    "Year" = rep(the_year,12),
    "Max Daily Withdrawals" = as.numeric(gsub(",","", max.withdrawals.mgd)),
    "System.Name" = water.system.name,
    "PSWID" = pswid,
    "Ownership" = ownership) %>%

     mutate("Date" = my(paste(Month,"-",Year)))
   #return the data frame
   return(withdrawals_function_df)
}
```

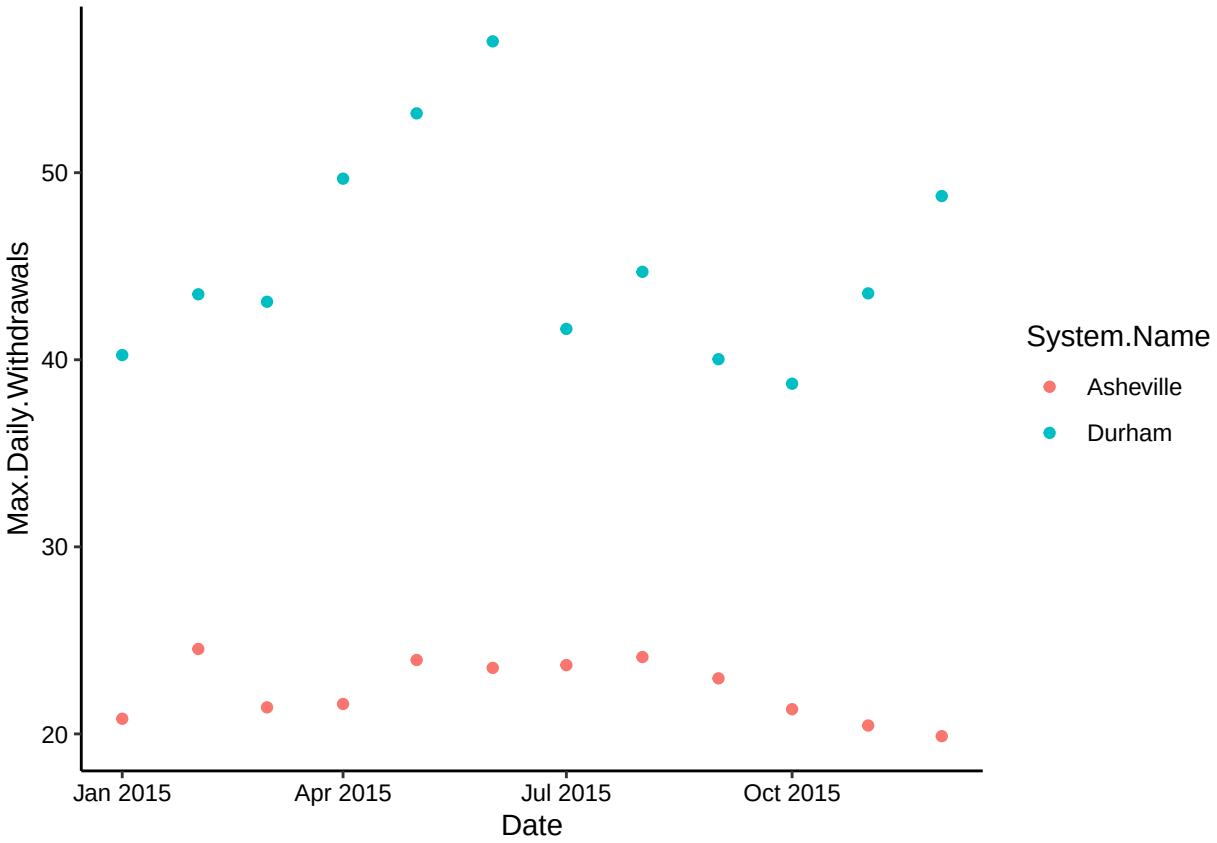
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7
df2015<-scrape.it(2015,"03-32-010")
```

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
dfAsheville<-scrape.it(2015, "01-11-010")
df<-rbind(df2015, dfAsheville)

ggplot(df, aes(x=Date, y=Max.Daily.Withdrawals, color = System.Name))+
  geom_point()
```



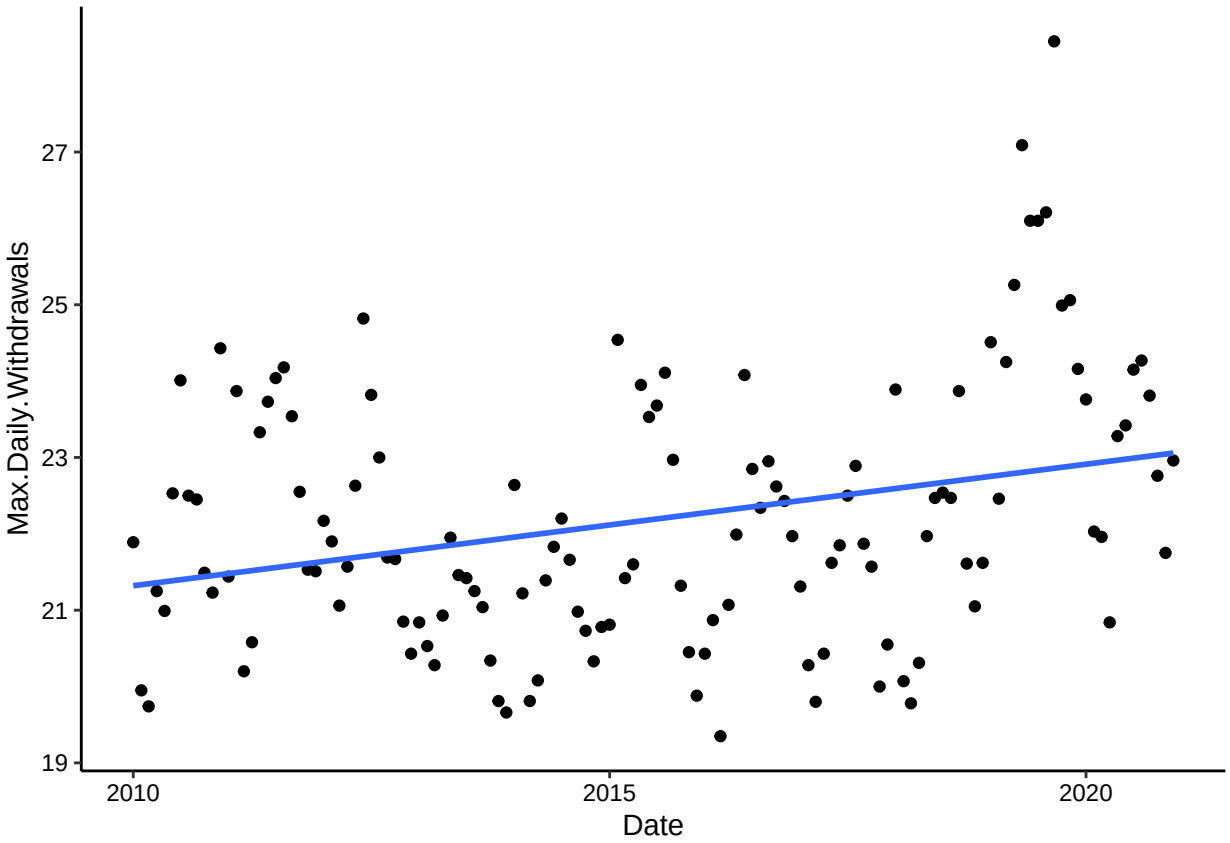
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9
years<-seq(2010, 2020)
Ash2010<-scrape.it(2010, "01-11-010")
Ash2011<-scrape.it(2011, "01-11-010")
Ash2012<-scrape.it(2012, "01-11-010")
Ash2013<-scrape.it(2013, "01-11-010")
Ash2014<-scrape.it(2014, "01-11-010")
Ash2015<-scrape.it(2015, "01-11-010")
Ash2016<-scrape.it(2016, "01-11-010")
Ash2017<-scrape.it(2017, "01-11-010")
Ash2018<-scrape.it(2018, "01-11-010")
Ash2019<-scrape.it(2019, "01-11-010")
Ash2020<-scrape.it(2020, "01-11-010")

dfAsh<-rbind(Ash2010, Ash2011, Ash2012, Ash2013, Ash2014, Ash2015, Ash2016, Ash2017, Ash2018, Ash2019, A
  arrange(Date)

ggplot(dfAsh, aes(x=Date, y=Max.Daily.Withdrawals))+
  geom_point()+
  geom_smooth(method = lm, se = F)
```

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




```
min(dfAsh$Max.Daily.Withdrawals)
```

```
## [1] 19.35
```

```
max(dfAsh$Max.Daily.Withdrawals)
```

```
## [1] 28.45
```

Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time?