Assignment 10: Data Scraping

Sara Sayed

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_10_Data_Scraping.Rmd") prior to submission.

The completed exercise is due on Tuesday, April 6 at 11:59 pm.

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/sarasayed/Documents/Data Analytics/Environmental_Data_Analytics_2021/Assignments"
```

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

mytheme <- theme_classic() +
  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=2019

Indicate this website as the as the URL to be scraped.

```
#2

theURL <- read_html(
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019')
```

3. The data we want to collect are listed below:

- From the "System Information" section:
- Water system name
- PSWID
- Ownership
- From the "Water Supply Sources" section:
- Maximum monthly withdrawals (MGD)

In the code chunk below scrape these values into the supplied variable names.

```
#3

the_facility_name <- theURL %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
the_facility_name
```

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

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

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

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

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

```
MGD <-  theURL %>%
  html_nodes("th~ td+ td") %>%
  html_text()
MGD
```

```
##  [1] "29.6200" "35.7300" "54.0700" "32.3900" "37.8600" "44.3500" "36.4300"
##  [8] "46.0200" "36.0600" "32.6000" "42.0500" "31.2000"
```

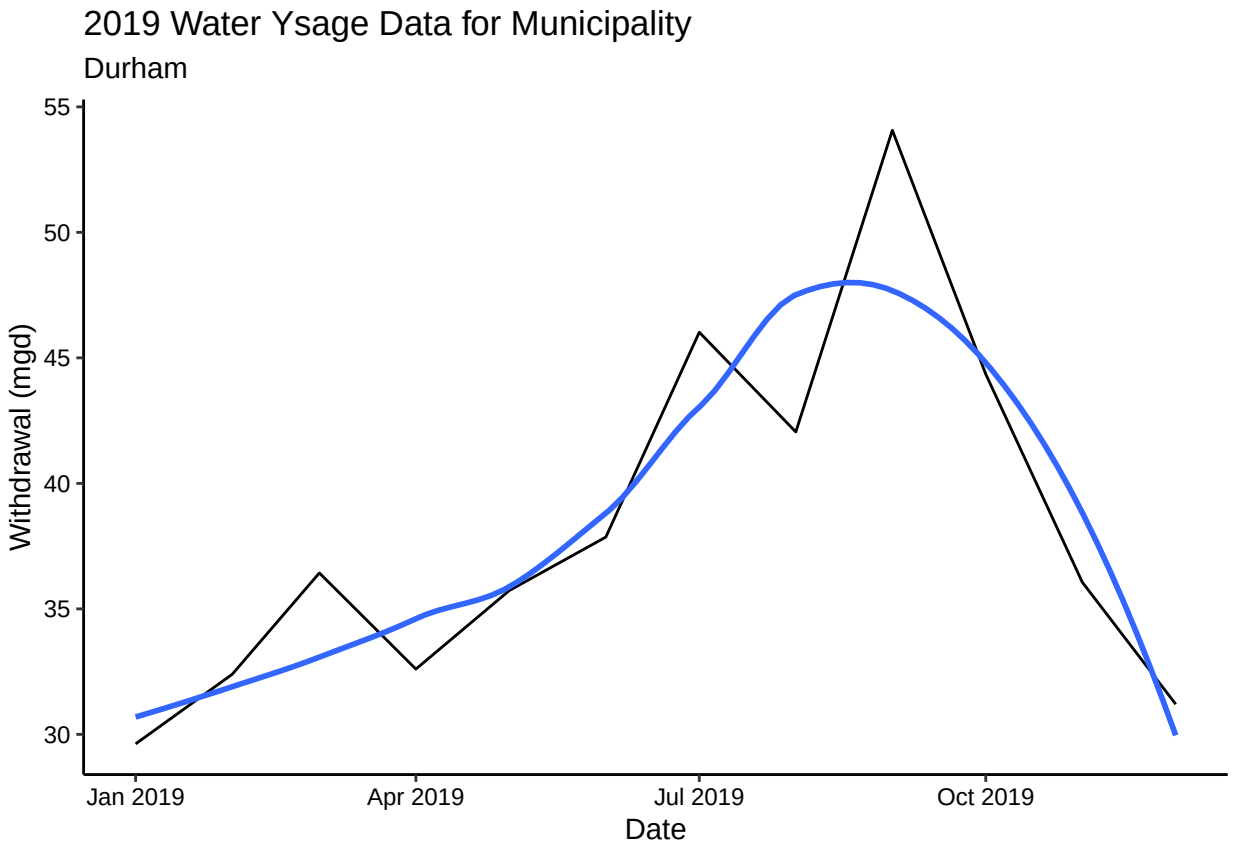
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

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 2019.

```
#4

df_withdrawals <- data.frame("Month" = c("Jan", "May", "Sep", "Feb", "June", "Oct",
                             "Mar", "July", "Nov", "April", "Aug", "Dec"),
                             "Year" = rep(2019,12),
                               Max_Withdrawals_mgd = as.numeric(MGD),
                               Facility_Name = the_facility_name,
                               PWSID = PWSID,
                               Ownership = Ownership)
df_withdrawals$Date <- as.yearmon(paste(df_withdrawals$Year,
                                        df_withdrawals$Month), "%Y %b")
df_withdrawals$Date <- as.Date(df_withdrawals$Date, format = "%b %Y")

#5

ggplot(df_withdrawals,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2019 Water Ysage Data for",Ownership),
       subtitle = the_facility_name,
       y="Withdrawal (mgd)",
       x="Date")
```

```
## `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 data scraped.

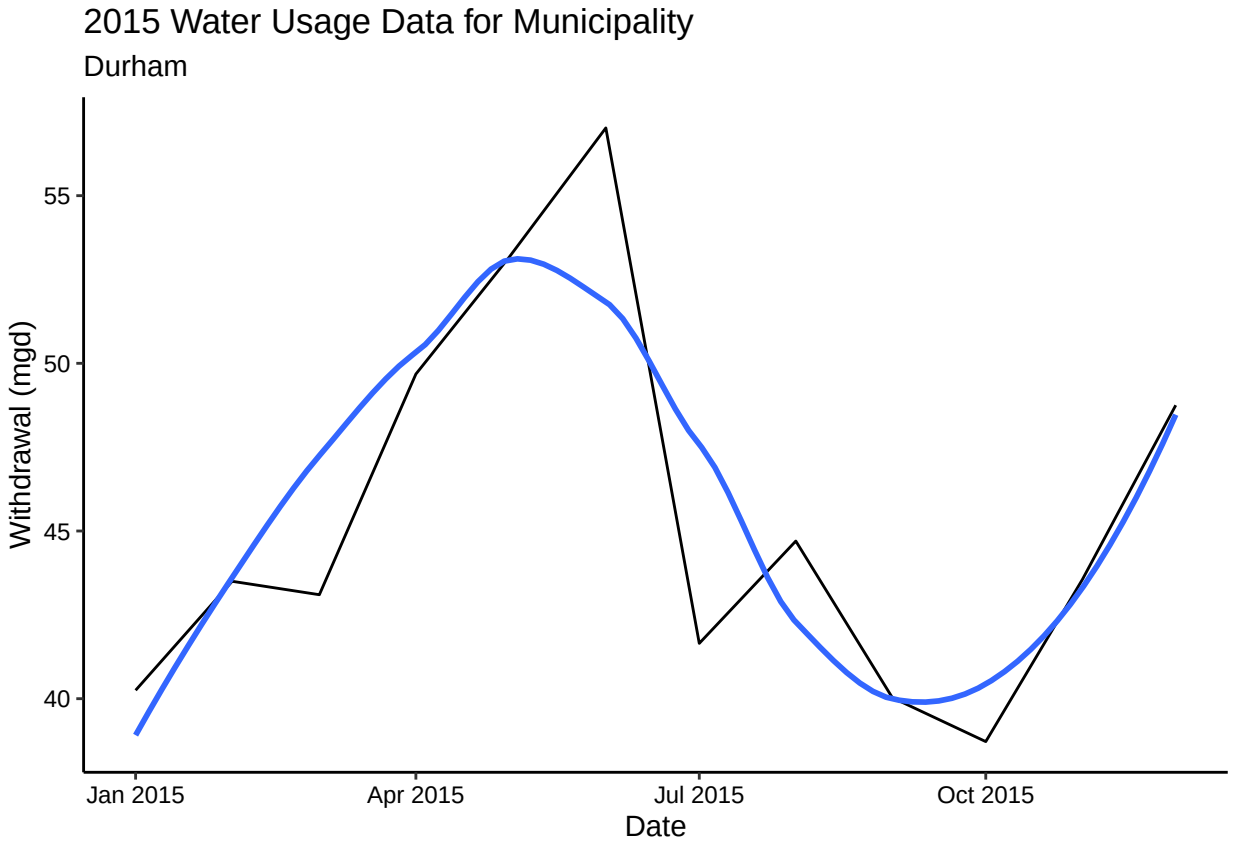
```
#6.

scrape.it <- function(the_year, the_pwsid){
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php',
                                  '?pwsid=',the_pwsid,'&year=', the_year))
  MGD_Tag <- 'th~ td+ td'
  Date_Tag <-'.fancy-table:nth-child(31) tr+ tr th'
  Facility_Tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  Owner_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'

  the_MGD <-the_website %>% html_nodes(MGD_Tag)%>% html_text()
  the_date <- the_website %>% html_nodes(Date_Tag)%>% html_text()
  the_facility <- the_website %>% html_nodes(Facility_Tag)%>% html_text()
  the_owner <-the_website %>% html_nodes(Owner_tag)%>% html_text()
  PWSID <- the_website %>% html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

  df <- data.frame(data.frame("Month" = c("Jan", "May", "Sep", "Feb", "June", "Oct",
                              "Mar", "July", "Nov", "April", "Aug", "Dec"),
                               Year = the_year,
                               Max_Withdrawals_mgd = as.numeric(the_MGD),
                               Facility_Name = the_facility,
                               PWSID = PWSID,
                               Ownership = the_owner))
```

```
  df$Date <- as.yearmon(paste(df$Year, df$Month), "%Y %b")
  df$Date <- as.Date(df$Date, format = "%b %Y")
  return(df)
}
```

7. Use the function above to extract and plot max daily withdrawals for Durham for each month in 2015

```
#7

Durham_2015 <- scrape.it(2015,'03-32-010')
view(Durham_2015)

ggplot(Durham_2015,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2015 Water Usage Data for",Ownership),
       subtitle = the_facility_name,
       y="Withdrawal (mgd)",
       x="Date")
```

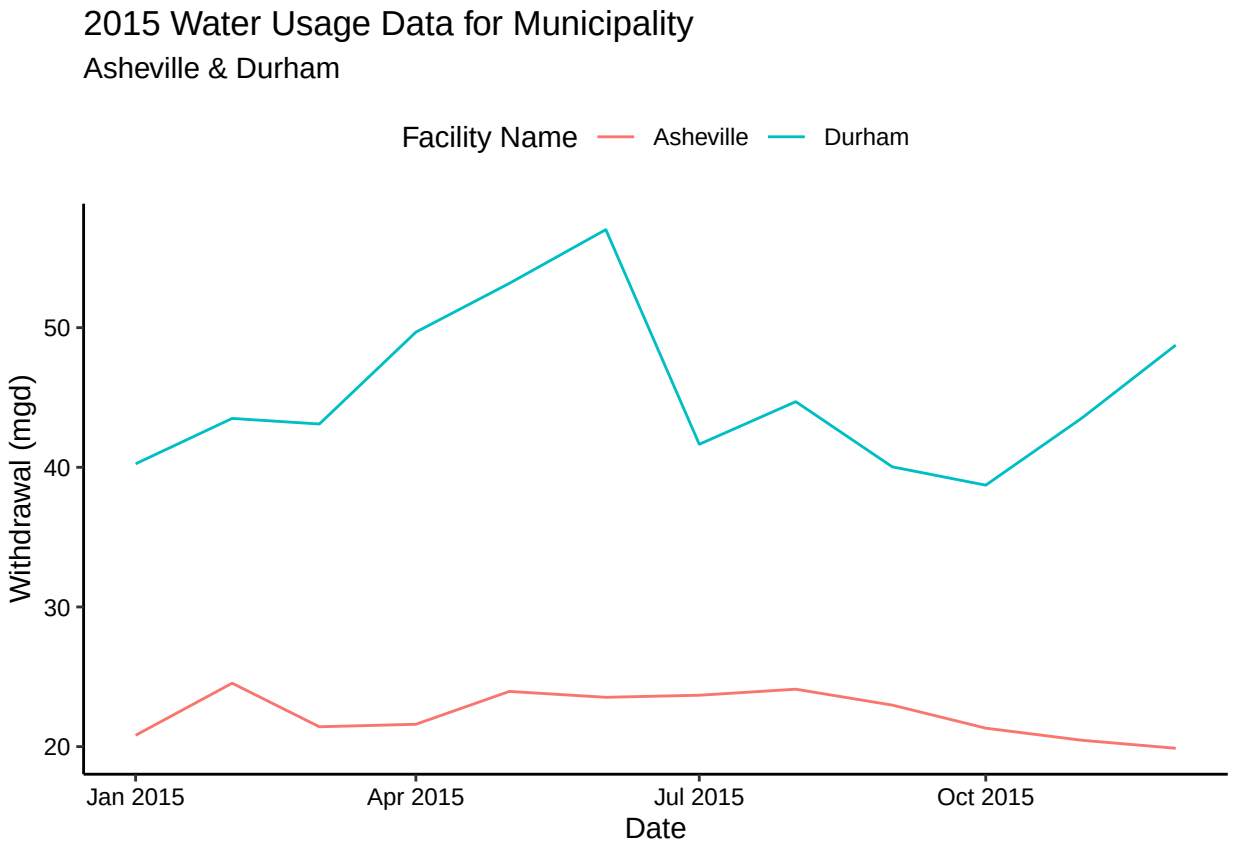
```
## `geom_smooth()` using formula 'y ~ x'
```



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

```
view(Ash_df)

Ash_Dur <- rbind(Durham_2015,Ash_df)

ggplot(Ash_Dur,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line(aes(color = Facility_Name)) +
   scale_color_discrete(name="Facility Name")+
  labs(title = paste("2015 Water Usage Data for",Ownership),
       subtitle = "Asheville & Durham",
       y="Withdrawal (mgd)",
       x="Date")
```

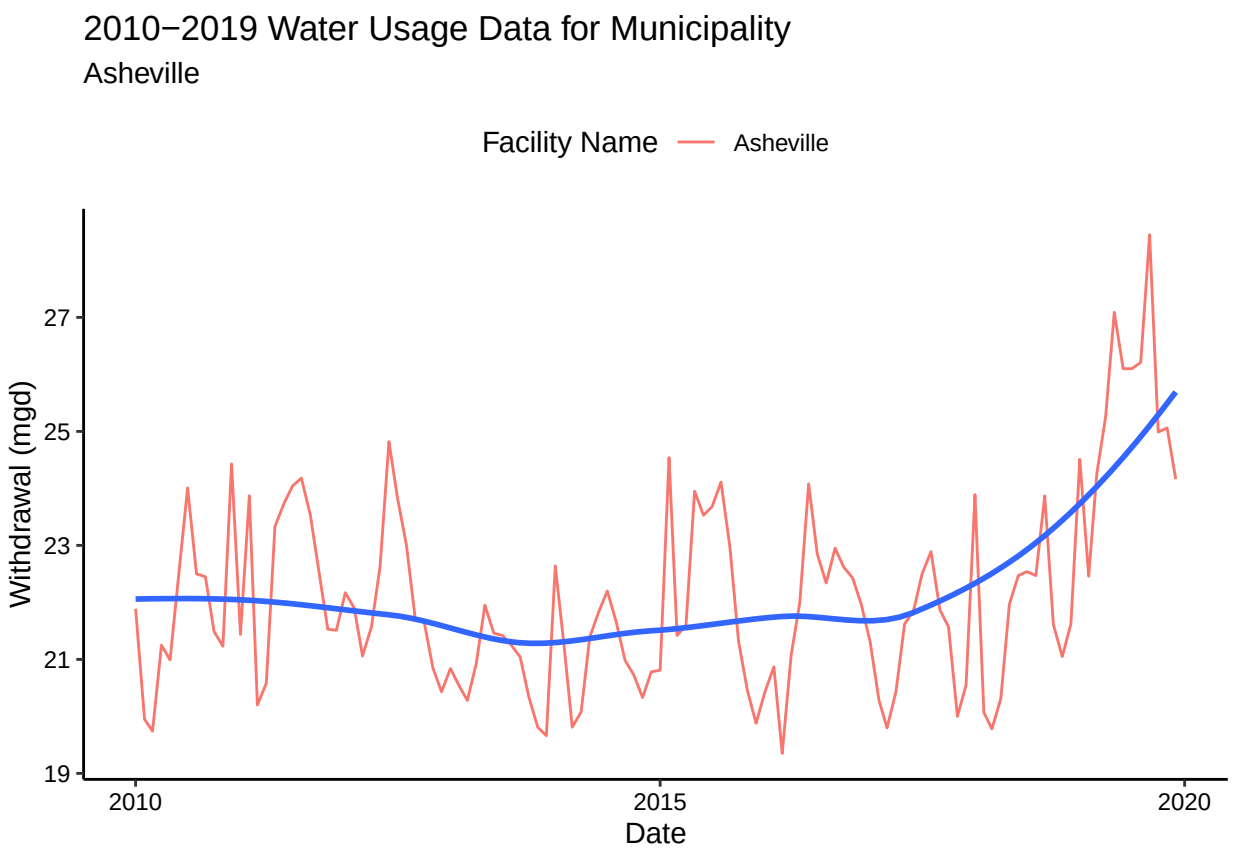


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

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

the_dfs <- map(the_years,scrape.it,the_pwsid=my_facility)

the_df <- bind_rows(the_dfs)

ggplot(the_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line(aes(color = Facility_Name))+
  geom_smooth(method="loess",se=FALSE) +
   scale_color_discrete(name="Facility Name") +
  labs(title = paste("2010-2019 Water Usage Data for",Ownership),
```

```
       subtitle = "Asheville",
       y="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? Asheville's water usage stayed pretty steady in the early 2010's and then began to steadily increase after 2015. This is likely due to the growth in population in the city.