Assignment 09: Data Scraping

Jackie Fahrenholz

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
#get and check wd
setwd("C:/Users/Jackie/Desktop/ENV872/Environmental_Data_Analytics_2022/Assignments")
getwd()
```

```
## [1] "C:/Users/Jackie/Desktop/ENV872/Environmental_Data_Analytics_2022/Assignments"
```

```
#load packages
library(tidyverse)
library(rvest)
```

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

```
library(lubridate)
#make the theme
mytheme <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right") +
  theme(plot.title = element_text(hjust = 0.5))

#set  theme
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
webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
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:
- 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 <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

pwsid <- webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

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

max.withdrawals.mgd <- webpage %>%
  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.)

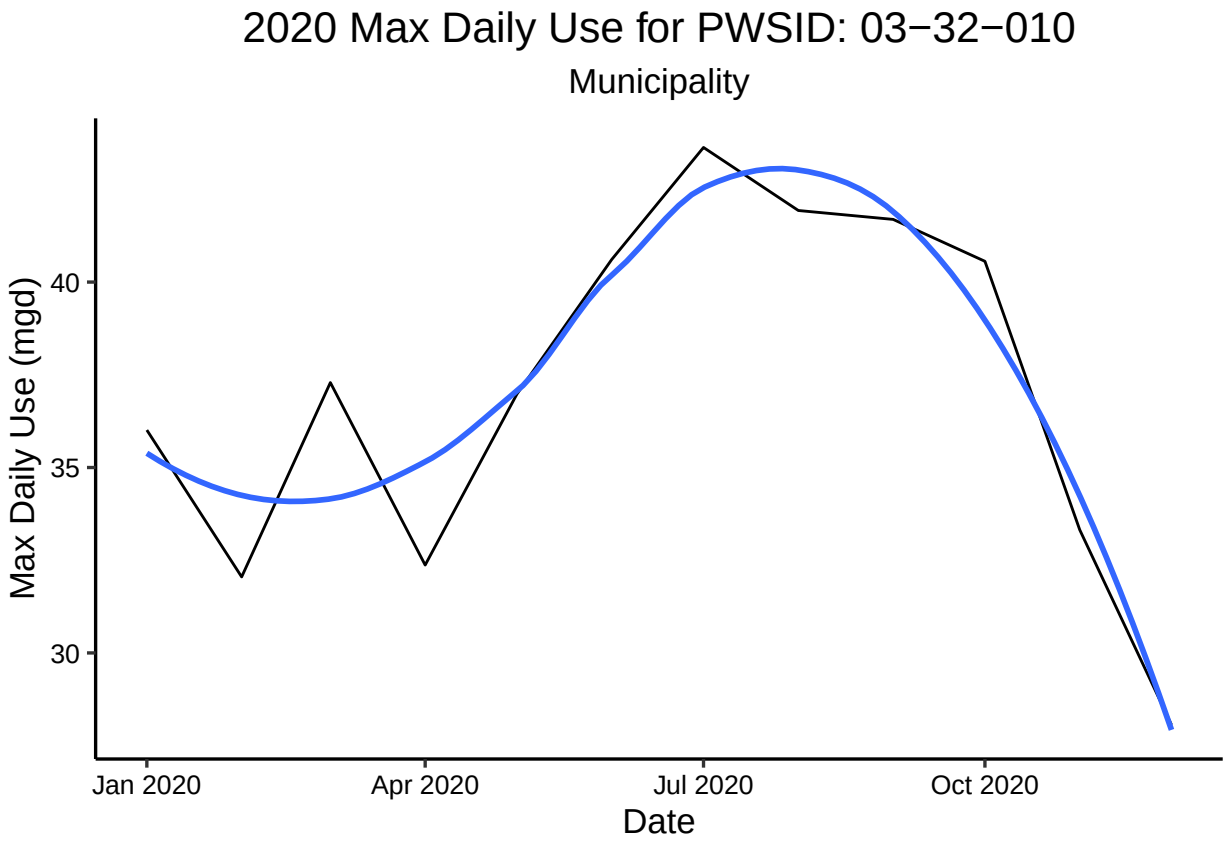
TIP: Use `rep()` to repeat a value when creating a dataframe.

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
#Create a dataframe of withdrawals; remember months aren't in order
df_ncwater <- data.frame("Month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                         "Year" = rep(2020,12),
                             "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

#modify to add the rest of the columns, and include date column
df_ncwater <- df_ncwater %>%
  mutate(Water_system = !!water.system.name,
         PWSID = !!pwsid,
         Ownership = !!ownership,
         Date = my(paste(Month,"-",Year)))

#5
ggplot(df_ncwater,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2020 Max Daily Use for PWSID:",pwsid),
       subtitle = ownership,
       y="Max Daily Use (mgd)",
       x="Date") +
  theme(plot.subtitle = element_text(hjust = 0.5))
```

```
## `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 scraped.**

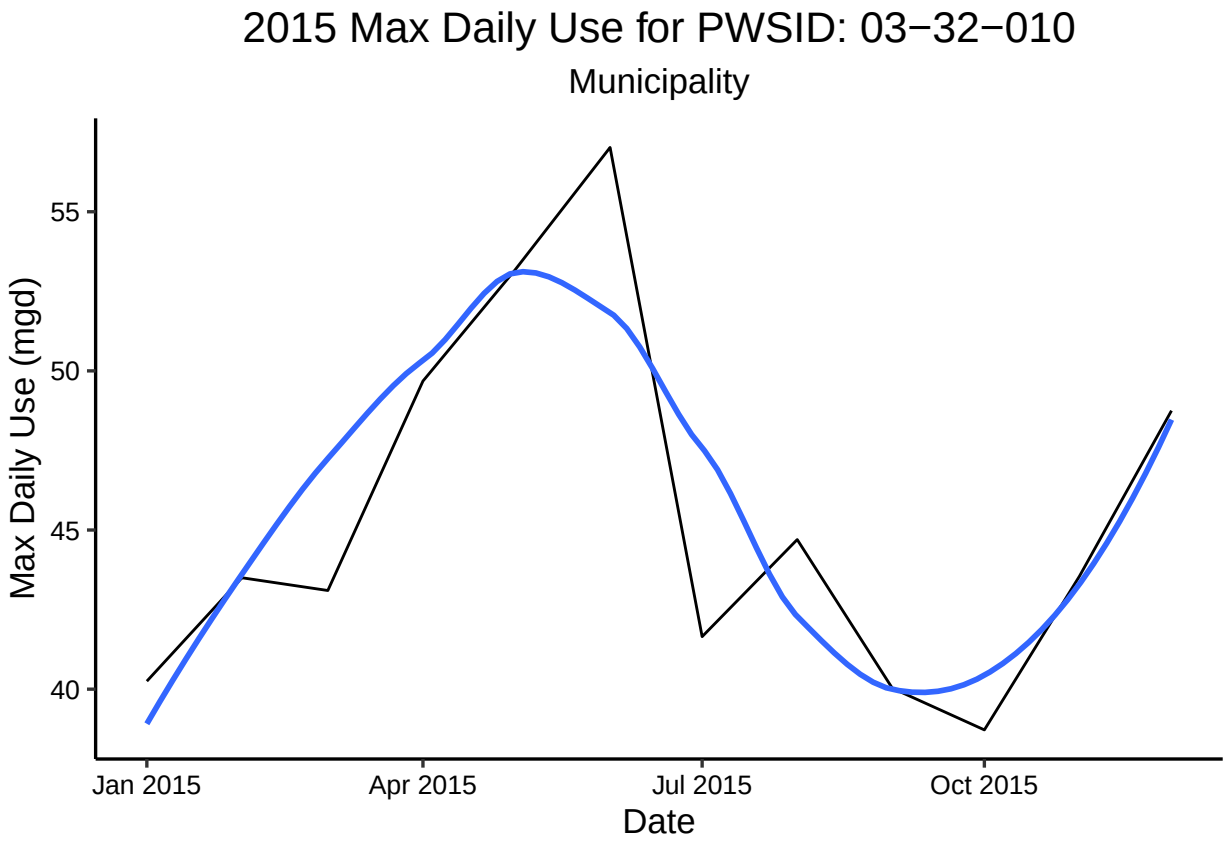
```
#6.

the_year <- 2020

#Create our scraping function
scrape.it <- function(the_year, pwsid){
  #Format based on webpage recall method
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
                                  pwsid,'&year=',the_year))

  #Set the 'tag' for each variable from above
  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)"
  max.withdrawals.mgd.tag <- "th~ td+ td"

  #Scrape the data items
  water.system.name <- the_website %>% html_nodes(water.system.name.tag) %>% html_text()
  pwsid <- the_website %>% html_nodes(pwsid.tag) %>%  html_text()
  ownership <- the_website %>% html_nodes(ownership.tag) %>% html_text()
  max.withdrawals.mgd <- the_website %>% html_nodes(max.withdrawals.mgd.tag) %>% html_text()

  #Convert to a dataframe
```

```
  df_ncwater <- data.frame("Month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                           "Year" = rep(the_year,12),
                               "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd)) %>%
    mutate(Water_system = !!water.system.name,
           PWSID = !!pwsid,
           Ownership = !!ownership,
           Date = my(paste(Month,"-",Year)))

  #Pause for a moment - scraping etiquette
  Sys.sleep(2)

  #Return the dataframe
  return(df_ncwater)
}
```

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

```
#7
test_df <- scrape.it(2015,'03-32-010')
view(test_df)
#plot this
ggplot(test_df, aes(x = Date, y = Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2015 Max Daily Use for PWSID:",pwsid),
       subtitle = ownership,
       y="Max Daily Use (mgd)",
       x="Date") +
  theme(plot.subtitle = element_text(hjust = 0.5))
```

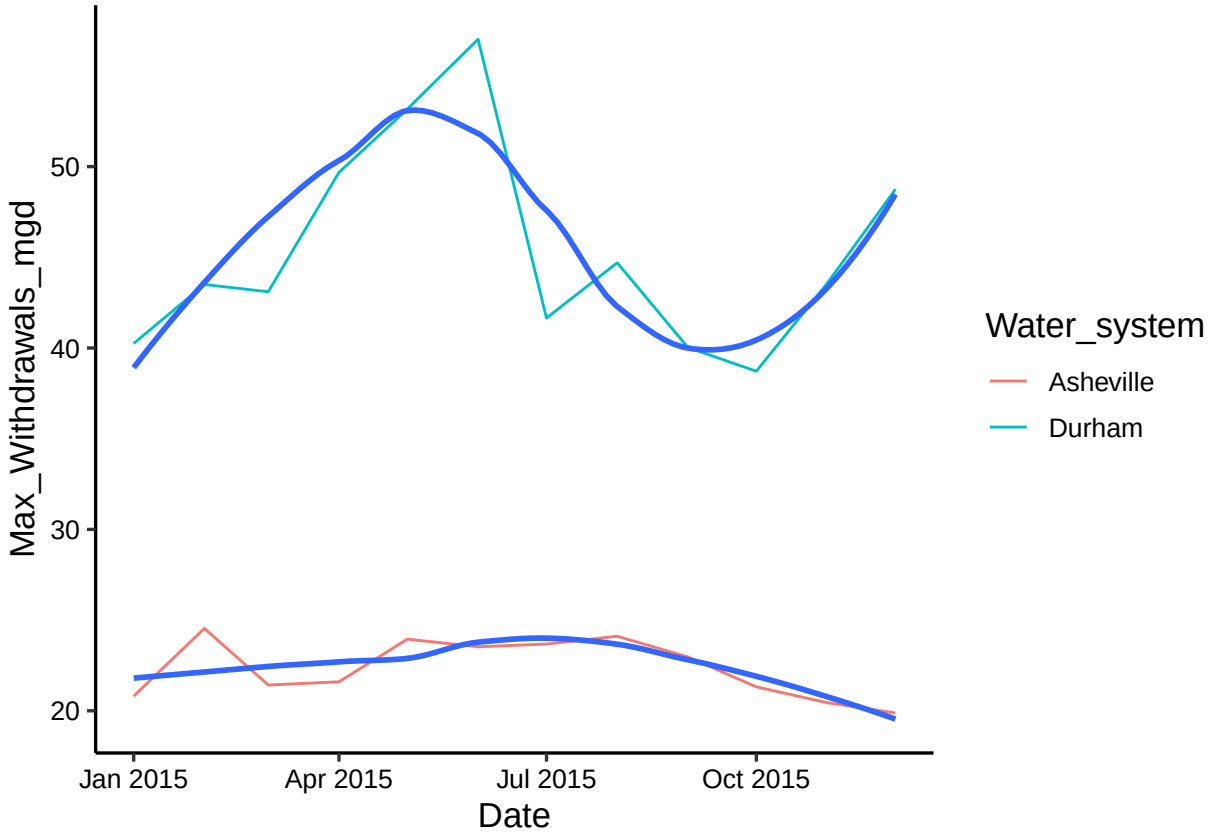
```
## `geom_smooth()` using formula 'y ~ x'
```



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
#create the Asheville dataframe
ash_df <- scrape.it(2015,'01-11-010')
view(ash_df)
#combine the dataframes
merge_df <- rbind(test_df, ash_df)
view(merge_df)
#create a plot that compares them
ggplot(merge_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line(aes(color = Water_system)) +
  geom_smooth(data = test_df, method = "loess", se = FALSE) +
  geom_smooth(data = ash_df, method = "loess", se = FALSE)
```

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




```
  labs(title = paste("2015 Max Daily Use for PWSID:",pwsid),
       subtitle = ownership,
       y="Max Daily Use (mgd)",
       x="Date") +
  theme(plot.subtitle = element_text(hjust = 0.5))
```

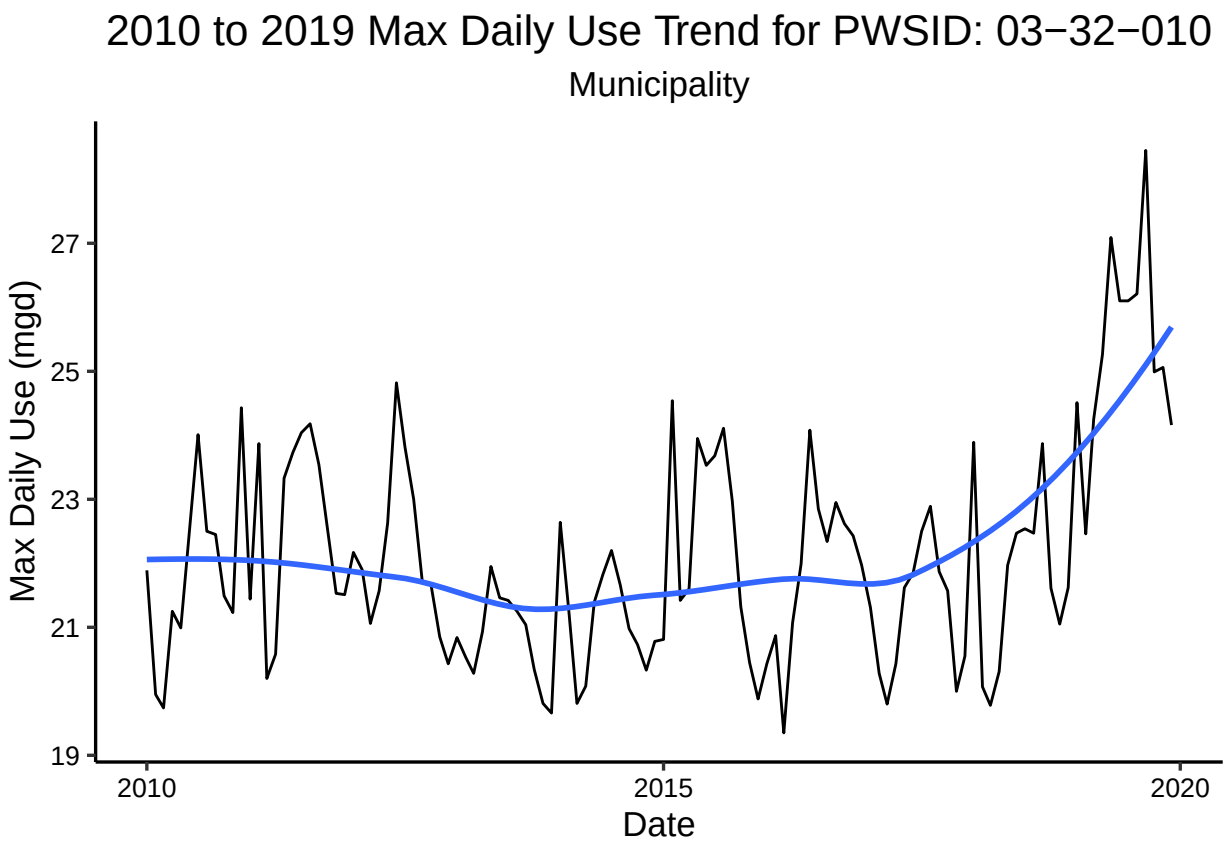
```
## NULL
```

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
#set the variables
the_years = rep(2010:2019)
my_facility = '01-11-010'
#use the lapply method
ash_many_df <- lapply(X = the_years,
                      FUN = scrape.it,
                      pwsid = my_facility)
#conflate
many_df <- bind_rows(ash_many_df)
#then plot it
ggplot(many_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2010 to 2019 Max Daily Use Trend for PWSID:",pwsid),
       subtitle = ownership,
```

```
       y="Max Daily Use (mgd)",
       x="Date") +
  theme(plot.subtitle = element_text(hjust = 0.5))
```

```
## `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 by looking at the plot created above, the trend in water usage over time has increased based on the trendline that was created for the plot.