Assignment 09: Data Scraping

Ying Liu

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] "C:/Users/Alina/Desktop/DUKE_22FALL/872/EDA-Fall2022/Assignments"
```

```
library(tidyverse)
library(lubridate)
library(rvest)
library(cowplot)
mytheme <- theme_half_open(font_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
```

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
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2021'
webpage <- read_html(the_base_url)
```

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()
water.system.name
```

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

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

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

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

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

```
max.withdrawals.mgd <-webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()
max.withdrawals.mgd
```

```
##  [1] "27.6400" "41.7900" "36.7200" "27.9700" "37.9500" "42.2400" "30.5400"
##  [8] "43.6200" "31.2800" "33.7600" "46.0800" "29.7800"
```

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...

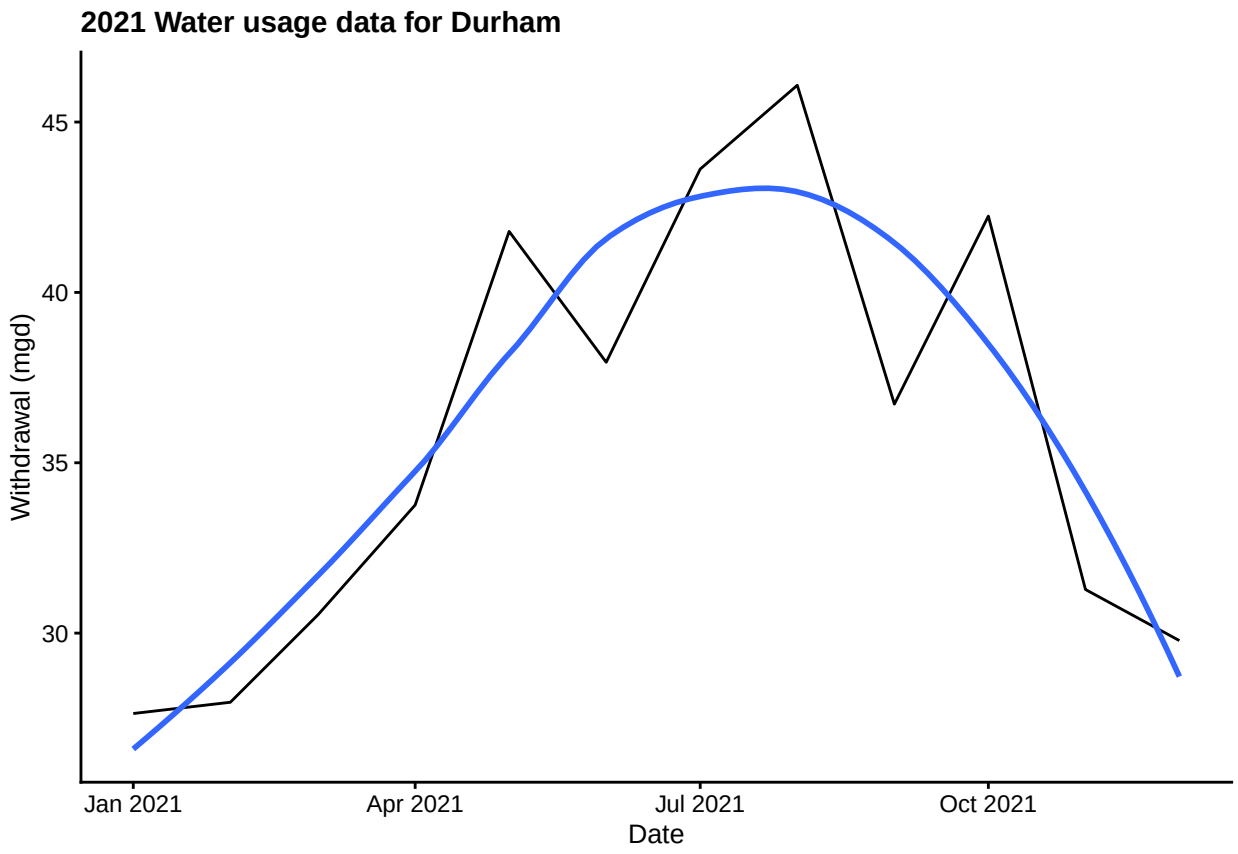
5. Create a line plot of the maximum daily withdrawals across the months for 2021

```
#4
#Create a dataframe of withdrawals
withdrawals <- data.frame("Month" = as.numeric(c("1","5","9","2","6","10","3","7","11","4","8","12")),
                          "Year" = rep(2021,12),
                  "Max_Withdrawals_mgd"=as.numeric(max.withdrawals.mgd))

withdrawals <- withdrawals %>%
  mutate(system_name = rep(water.system.name, length.out = NA, each = 1),
         pswid = rep(pswid, length.out = NA, each = 1),
         Ownership=rep(ownership, length.out = NA, each = 1),
         Date = my(paste(Month,"-",Year)))

#5
ggplot(withdrawals,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2021 Water usage data for",water.system.name),
       y="Withdrawal (mgd)",
       x="Date")+
  mytheme
```

```
## `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.
#Create our scraping function
scrape.it <- function(the_pwsid, the_year){

  #Retrieve the website contents
  the_webpage <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', the_pwsid,
  water.system.name_tag<-'div+ table tr:nth-child(1) td:nth-child(2)'
  pswid_tag<-'td tr:nth-child(1) td:nth-child(5)'
  ownership_tag<-'div+ table tr:nth-child(2) td:nth-child(4)'
  max_withdrawals_tag<-'th~ td+ td'

  water.system.name <- the_webpage %>%html_nodes(water.system.name_tag) %>% html_text()

  ownership <- the_webpage %>% html_nodes(ownership_tag) %>% html_text()
  max.withdrawals.mgd <-the_webpage %>% html_nodes(max_withdrawals_tag) %>% html_text()

 withdrawals <- data.frame("Month" = as.numeric(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))

withdrawals <- withdrawals %>%
```

```
  mutate(system_name = rep(water.system.name, length.out = NA, each = 1),
         pswid = rep(pswid, length.out = NA, each = 1),
         Ownership=rep(ownership, length.out = NA, each = 1),
         Date = my(paste(Month,"-",Year)))

 return(withdrawals)

}
```

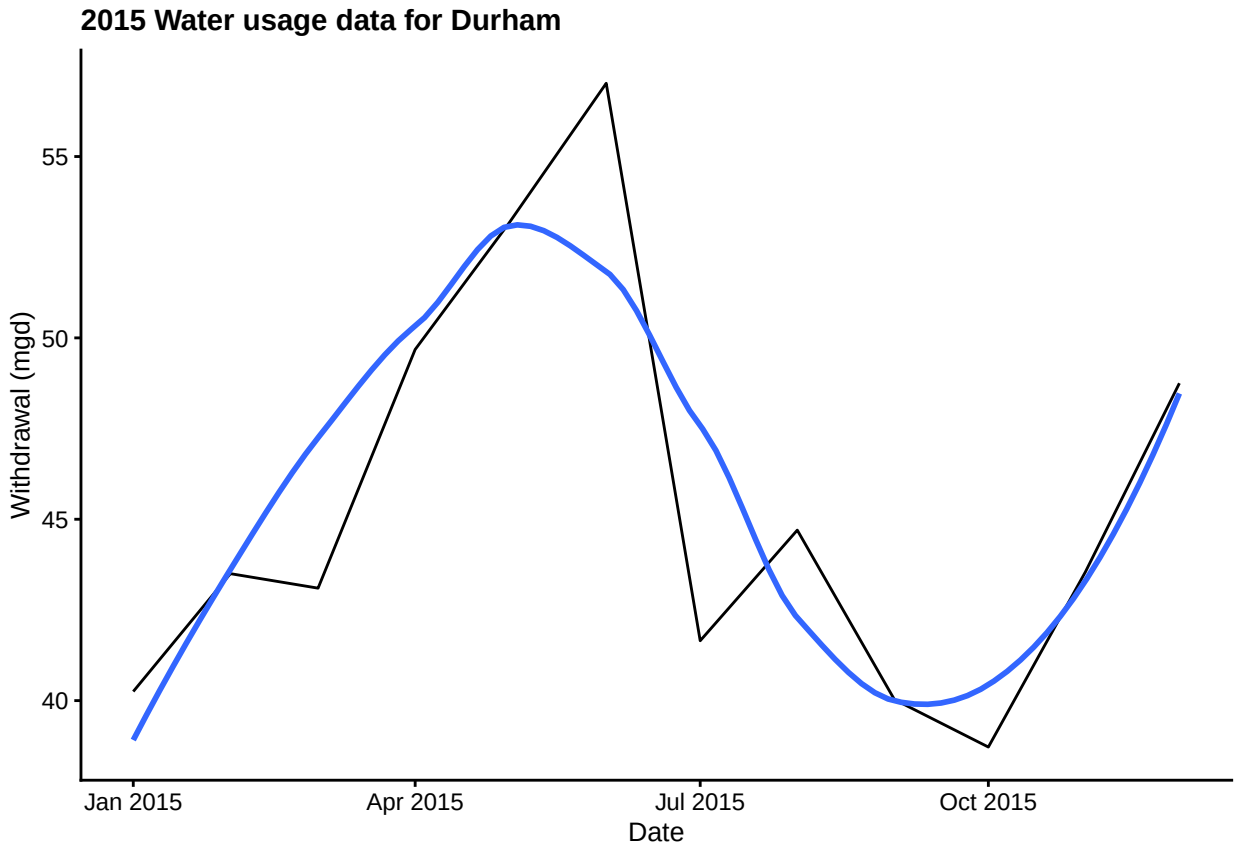
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7

Durham2015<-scrape.it('03-32-010','2015')

ggplot(Durham2015,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(2015,"Water usage data for",water.system.name),
       y="Withdrawal (mgd)",
       x="Date")+
  mytheme
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

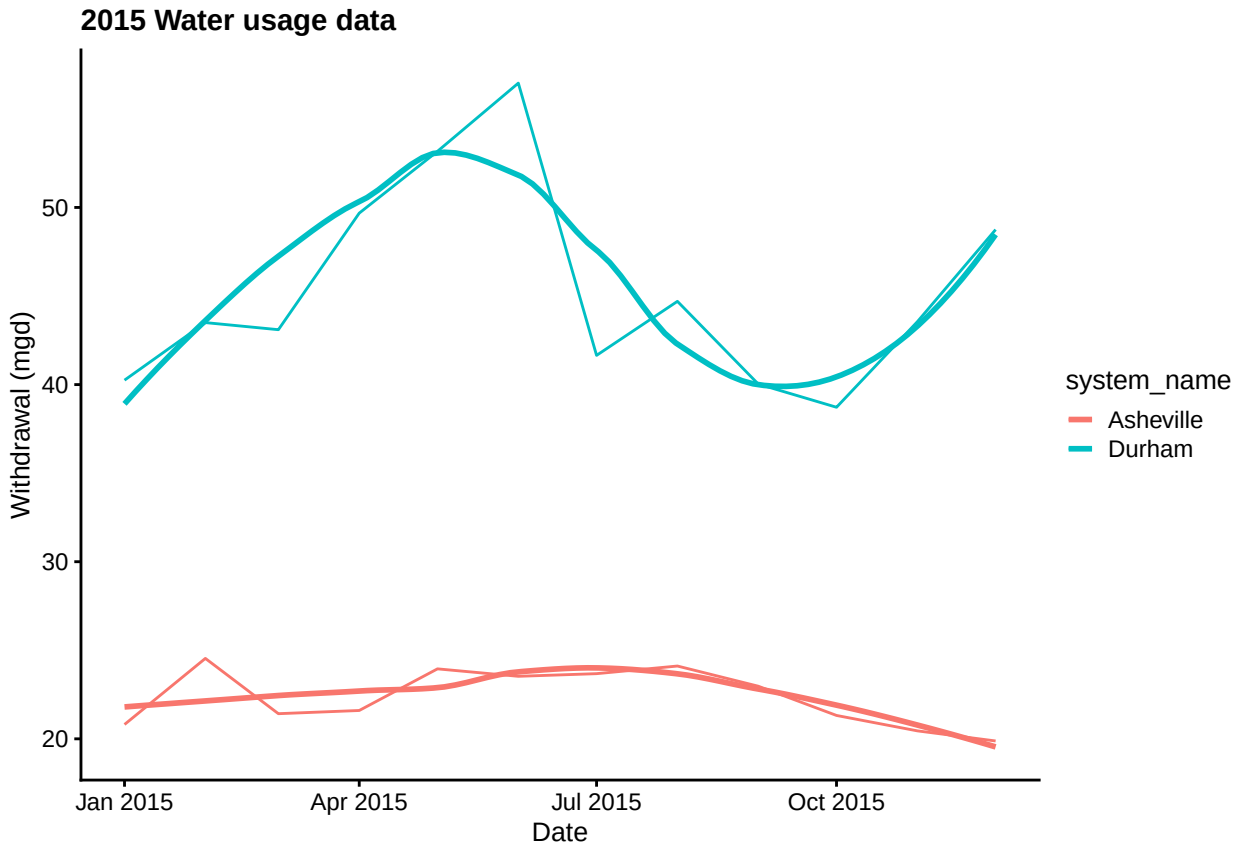


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8
Asheville2015<-scrape.it('01-11-010','2015')
AD2015<-rbind(Durham2015,Asheville2015)

ggplot(AD2015,aes(x=Date,y=Max_Withdrawals_mgd,color=system_name)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(2015,"Water usage data"),
       y="Withdrawal (mgd)",
       x="Date")+
  mytheme
```

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



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.

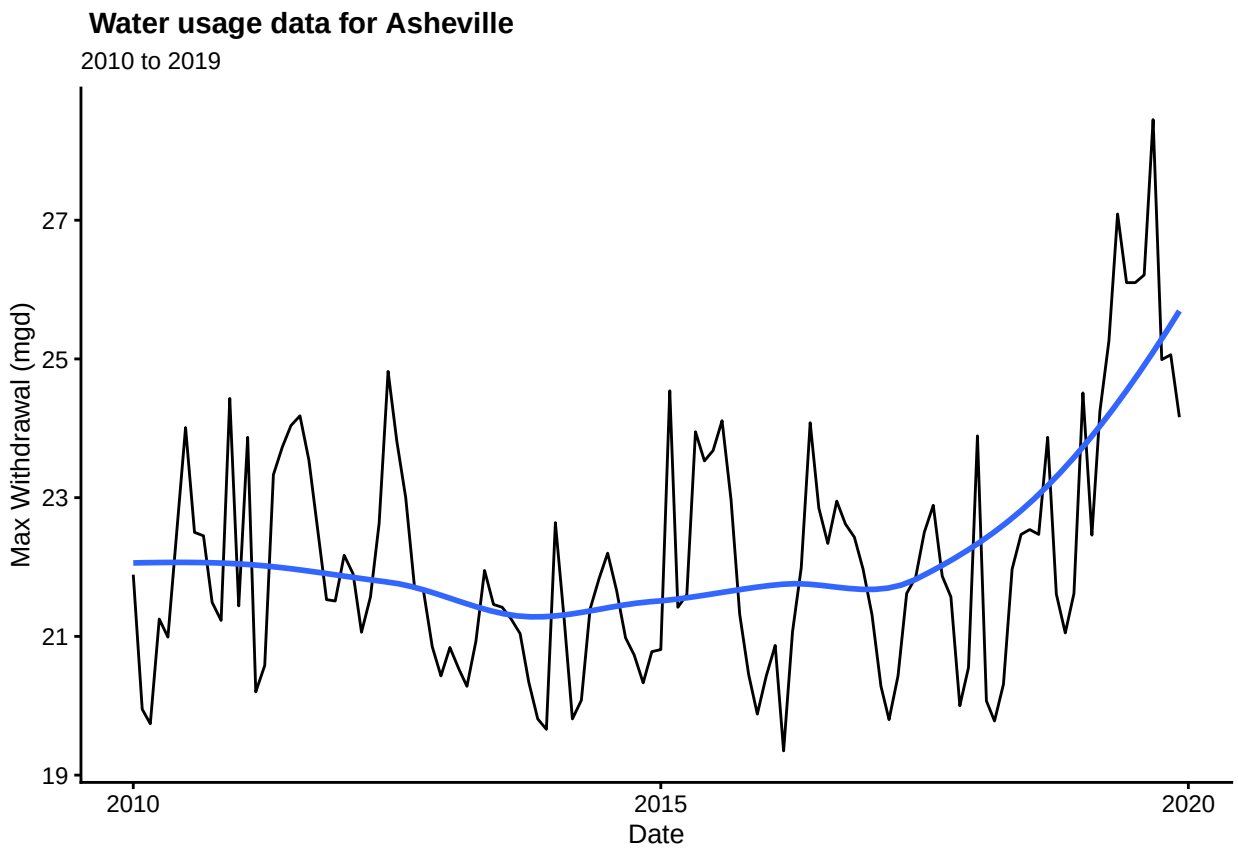
```r
#9
the_years = rep(2010:2019)
my_pwsid = '01-11-010'

the_dfs <- lapply(X = the_years,
                  FUN = scrape.it,
                  the_pwsid=my_pwsid)

the_df <- bind_rows(the_dfs)

ggplot(the_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(" Water usage data for Asheville"),
       subtitle = "2010 to 2019",
       y="Max Withdrawal (mgd)",
       x="Date")+
  mytheme
```

```
## `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, its remains stable form 2010 to 2017 and followed by a significant increase.