Assignment 09: Data Scraping

Megan Lundequam

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
library(tidyverse)
library(lubridate)
library(rvest)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
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
webpageDurham <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020')
webpageDurham
```

```
## {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 <- webpageDurham %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water.system.name
```

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

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

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

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

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

```
max.withdrawals.mgd <- webpageDurham %>%
  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.

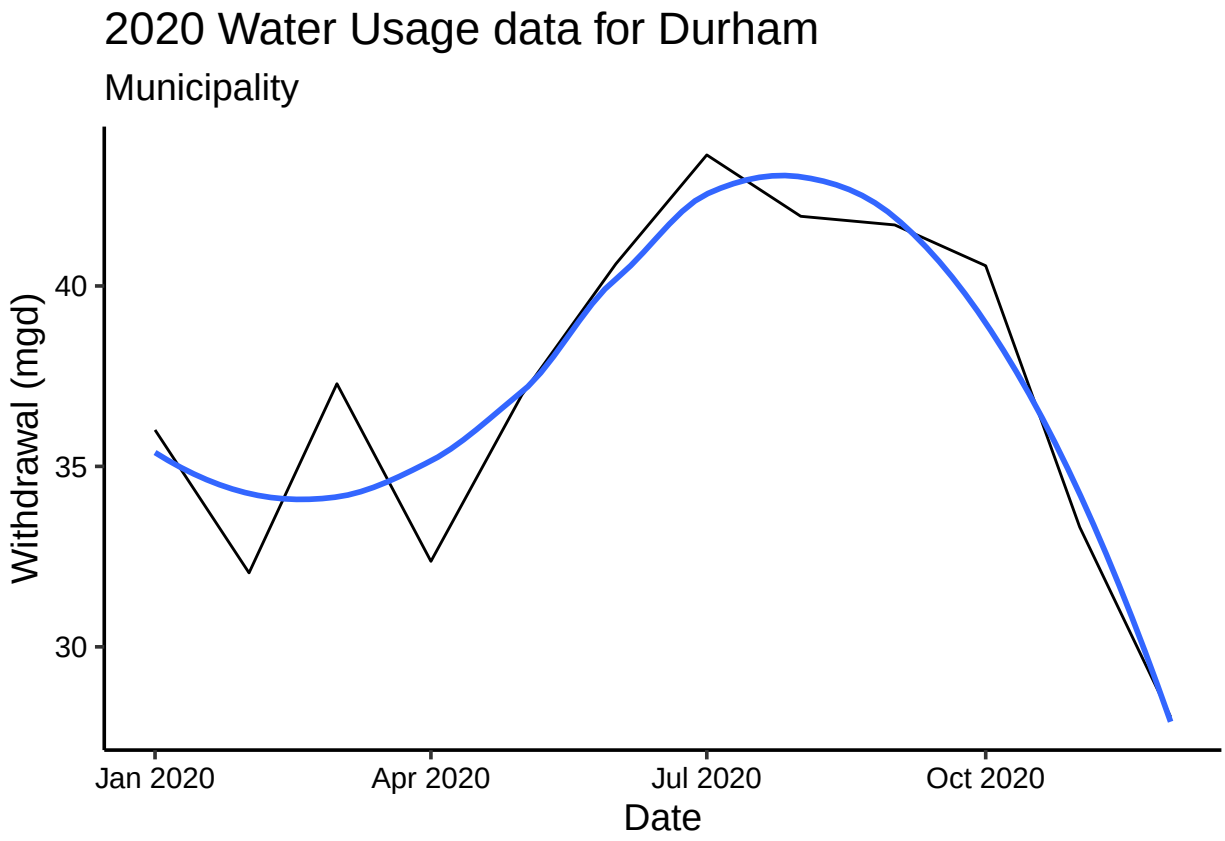
NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc... "Jan", "May", "Sept", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "April", "August", "Dec"

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

```
#4
durham_withdrawals <- 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)) %>%
  mutate(Water_System_Name = !!water.system.name,
         PWSID = !!pwsid,
         Ownership = !!ownership,
         Date = my(paste(Month,"-",Year)))

#5
ggplot(durham_withdrawals,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2020 Water Usage data for",water.system.name),
       subtitle = ownership,
       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 site scraped**.

```
#6.
#Create our scraping function
scrape.it <- function(the_pwsid, the_year){

  #Retrieve the website contents
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?',
                                  'pwsid=', the_pwsid, '&year=', the_year))

  #Set the element address variables (determined in the previous step)
  the_watersystem_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  the_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  the_pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
  the_data_tag <- 'th~ td+ td'

  #Scrape the data items
  the_watersystem_name <- the_website %>% html_nodes(the_watersystem_name_tag) %>% html_text()
  the_ownership <- the_website %>%   html_nodes(the_ownership_tag) %>%  html_text()
  the_pwsid_type <- the_website %>% html_nodes(the_pwsid_tag) %>% html_text()
  max_withdrawals <- the_website %>% html_nodes(the_data_tag) %>% html_text()
  month <- c(1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12)

  #Convert to a dataframe
  durham_withdrawals_df <- data.frame("Month" = month,
                                      "Year" = rep(the_year,12),
                                      "Max_Withdrawals_mgd" = as.numeric(max_withdrawals)) %>%
    mutate(Water_System_Name = !!the_watersystem_name,
           Ownership = !!the_ownership,
           PWSID = !!the_pwsid_type,
           Date = my(paste(Month,"-",Year)))


  #Return the dataframe
  return(durham_withdrawals_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
durham_2015_df <- scrape.it('03-32-010',2015)
view(durham_2015_df)
```

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.

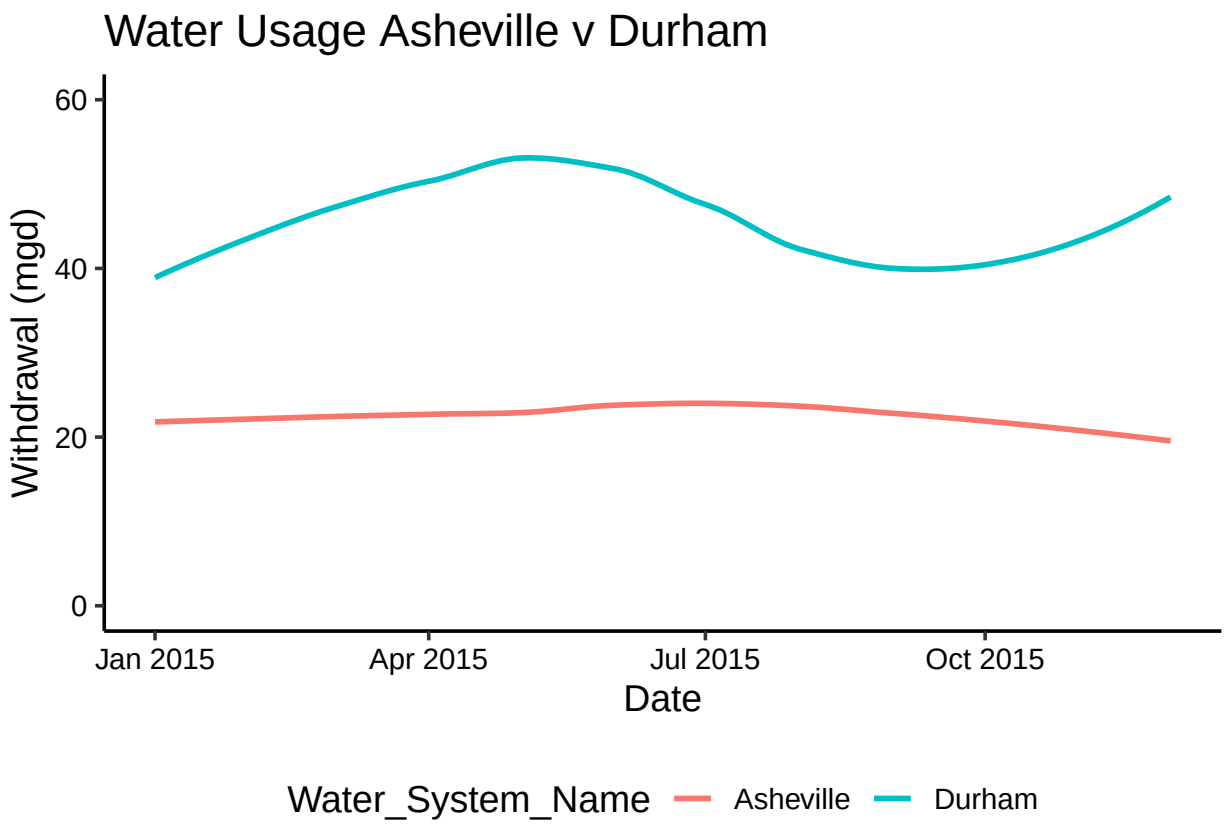
```
#8
asheville_2015_df <- scrape.it('01-11-010',2015)
view(asheville_2015_df)

ashville_durham_2015 <- full_join(asheville_2015_df, durham_2015_df)
```

```
## Joining, by = c("Month", "Year", "Max_Withdrawals_mgd", "Water_System_Name", "Ownership", "PWSID", "
```

```
ggplot(ashville_durham_2015,aes(x=Date,y=Max_Withdrawals_mgd,color=Water_System_Name)) +
  #geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  ylim(0, 60) +
  labs(title = "Water Usage Asheville v Durham",
       y="Withdrawal (mgd)",
       x="Date")
```

```
## `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.

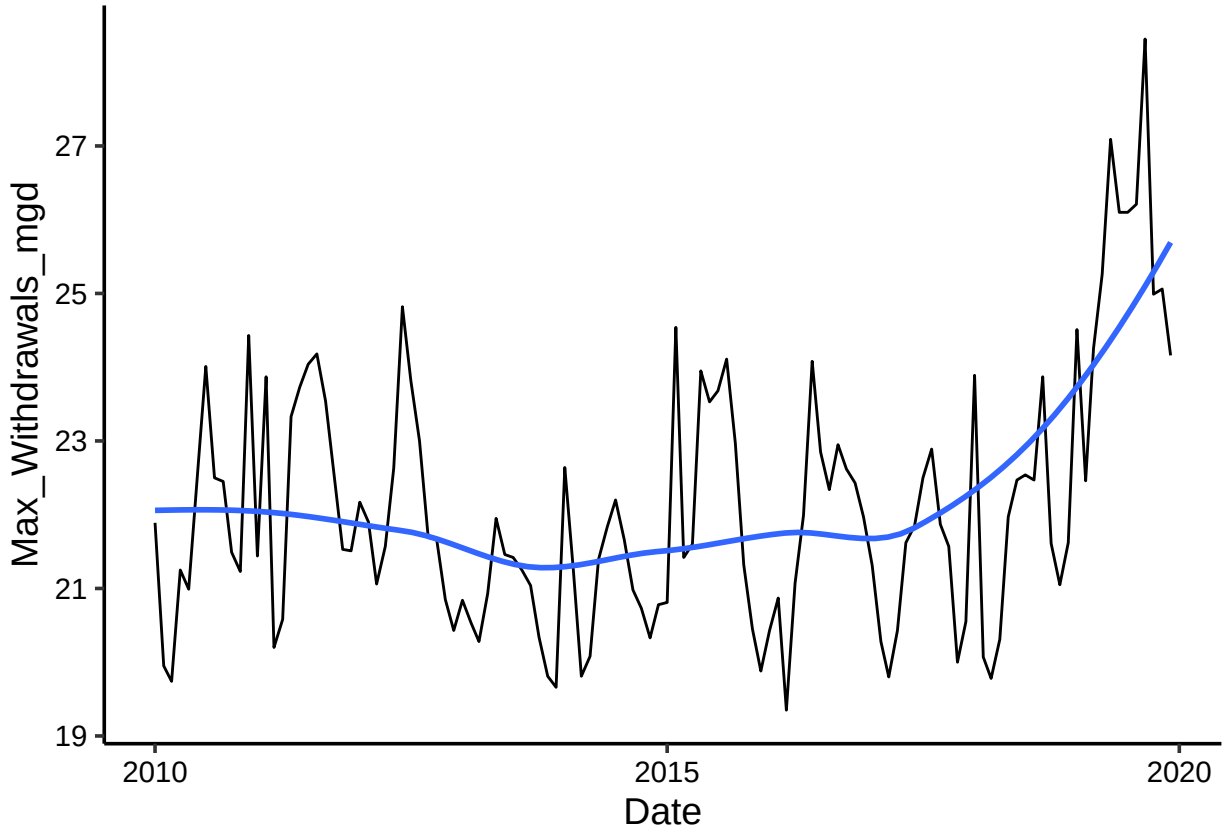
```
#9
#Set the inputs to scrape years 2015 to 2020 for the site "0004-0001"
the_years = rep(2010:2019) #creating a vector of years
ash_pwsid <- '01-11-010' #set pwsid

#Use lapply to apply the scrape function
ash_dfs <- lapply(X = the_years,
                  FUN = scrape.it,
                  the_pwsid=ash_pwsid)

#Conflate the returned dataframes into a single dataframe
ash_df <- bind_rows(ash_dfs)
```

```
#Plot, because it's fun and rewarding
ggplot(ash_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE)
```

```
## `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! Based on this plot, water use in Asheville appears to be increasing!