# Assignment 09: Data Scraping

Israel Golden

## 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] "/Users/israelgolden/Desktop/School/MEM/Semester 4/ENV 872/GitHub Repos/Environmental_Data_Analy
```

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

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=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')
```

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()
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] "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"
```

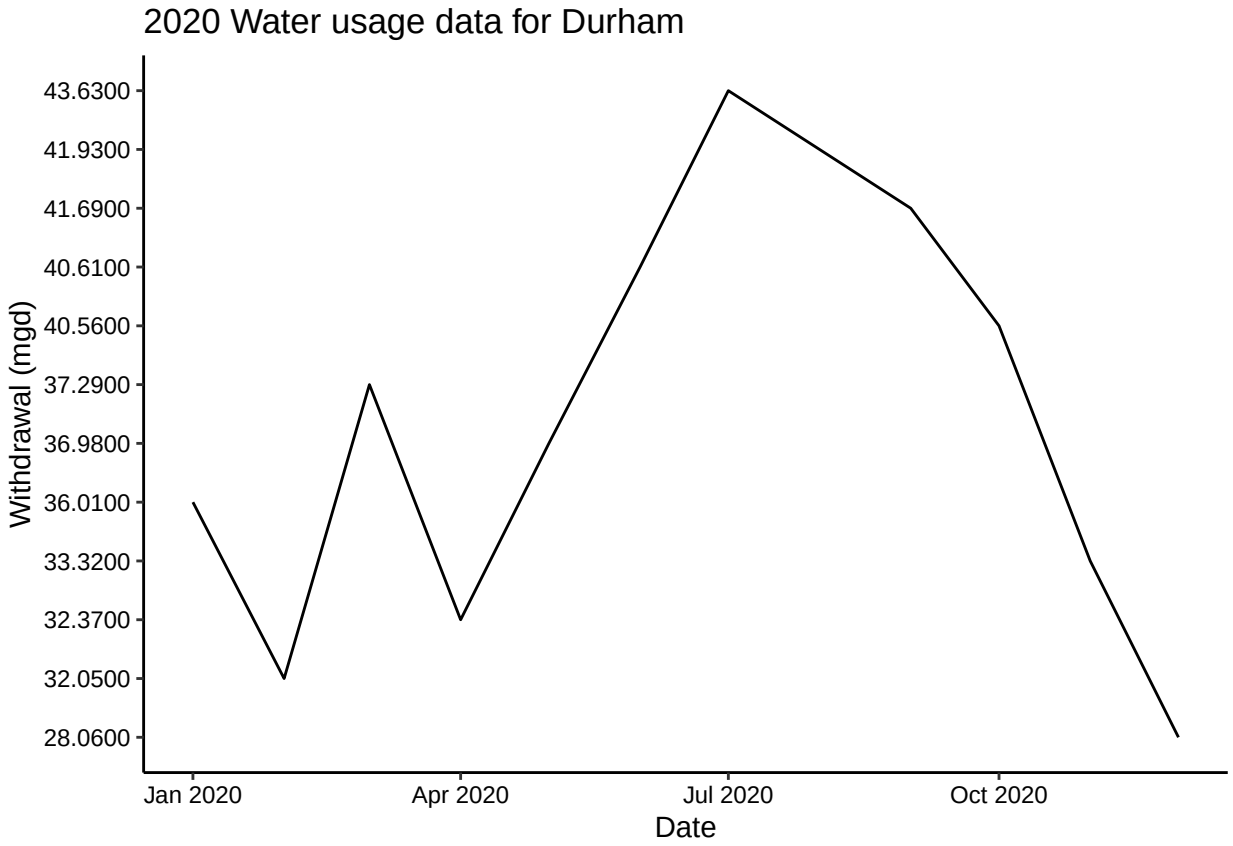
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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...

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

```
#4
month <-(c("01","05","09","02","06","10",
           "03","07","11","04","08","12"))

df_withdrawals <- data.frame("Month" = month,
                             "Year" = rep(2020,12),
                             "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

df_withdrawals <- df_withdrawals %>%
  mutate(Water_System_Name = !!water.system.name,
         Ownership = !!ownership,
         PSWID = !!pswid,
         Date = my(paste(Month,"-",Year)))

#5
ggplot(df_withdrawals,aes(x=Date, y=max.withdrawals.mgd, group = 1)) +
  geom_line() +
  labs(title = paste("2020 Water usage data for",water.system.name),
       y="Withdrawal (mgd)",
       x="Date")
```




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**.

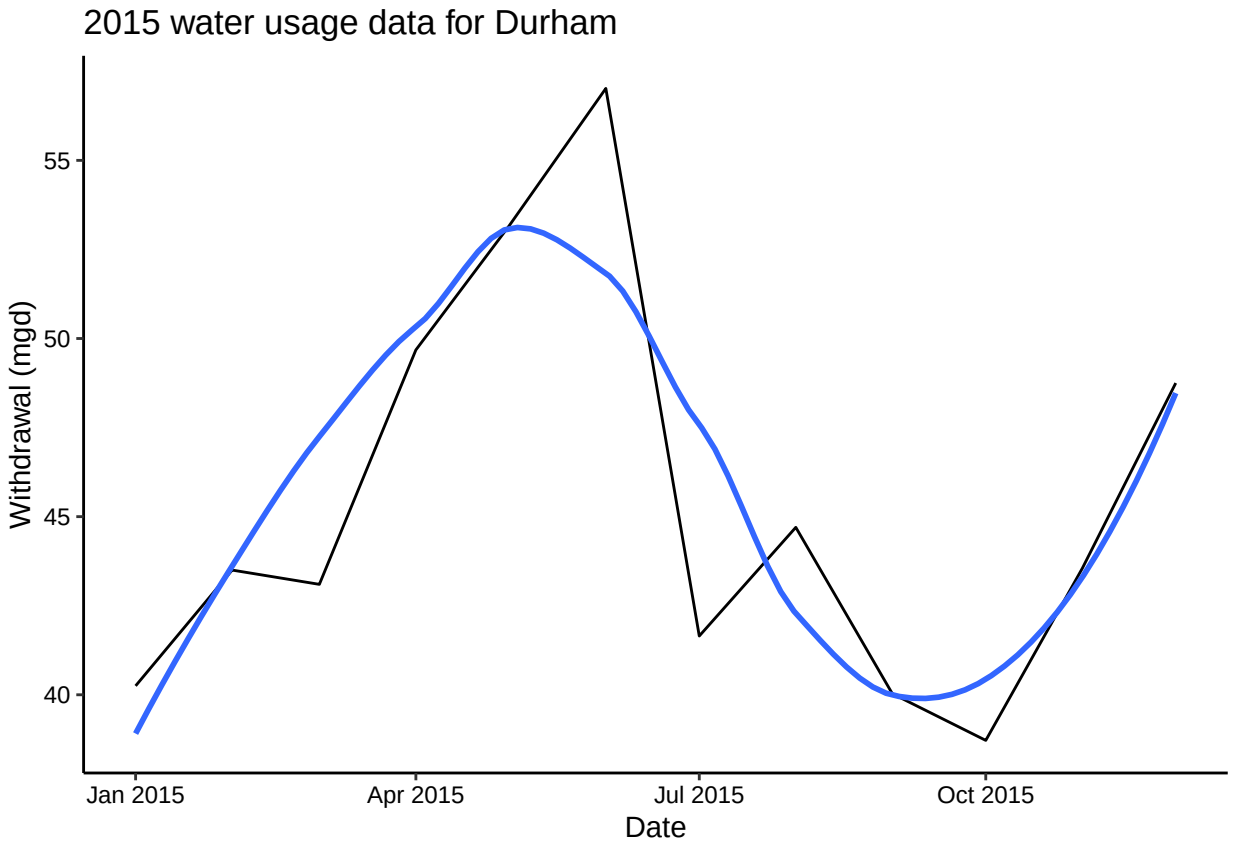
```
#6.

#Construct the scraping web address, i.e. its URL
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
the_pwsid <- '03-32-010'
the_year <- 2015
the_scrape_url <- paste0(the_base_url, the_pwsid, '&year=', the_year)
print(the_scrape_url)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2015"
```

```
#Retrieve the website contents
webpage <- read_html(the_scrape_url)

#Set the element address variables (determined in the previous step)
water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
max_withdrawals_tag <- 'th~ td+ td'
pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'

#Scrape the data items
water.system.name <- webpage %>% html_nodes(water_system_name_tag) %>% html_text()
```

```
ownership <- webpage %>% html_nodes(ownership_tag) %>%  html_text()
max.withdrawals.mgd <- webpage %>% html_nodes(max_withdrawals_tag) %>% html_text()
pwsid <- webpage %>% html_nodes(pwsid_tag) %>% html_text()

#Construct a dataframe from the scraped data
  scraped_df <- data.frame("Month" = month,
                           "Year" = rep(the_year,12),
                           "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd)) %>%
    mutate(WaterSystem = !!water.system.name,
           Ownership = !!ownership,
           PWSID = !!pwsid,
           Date = my(paste(Month,"-",Year)))

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

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



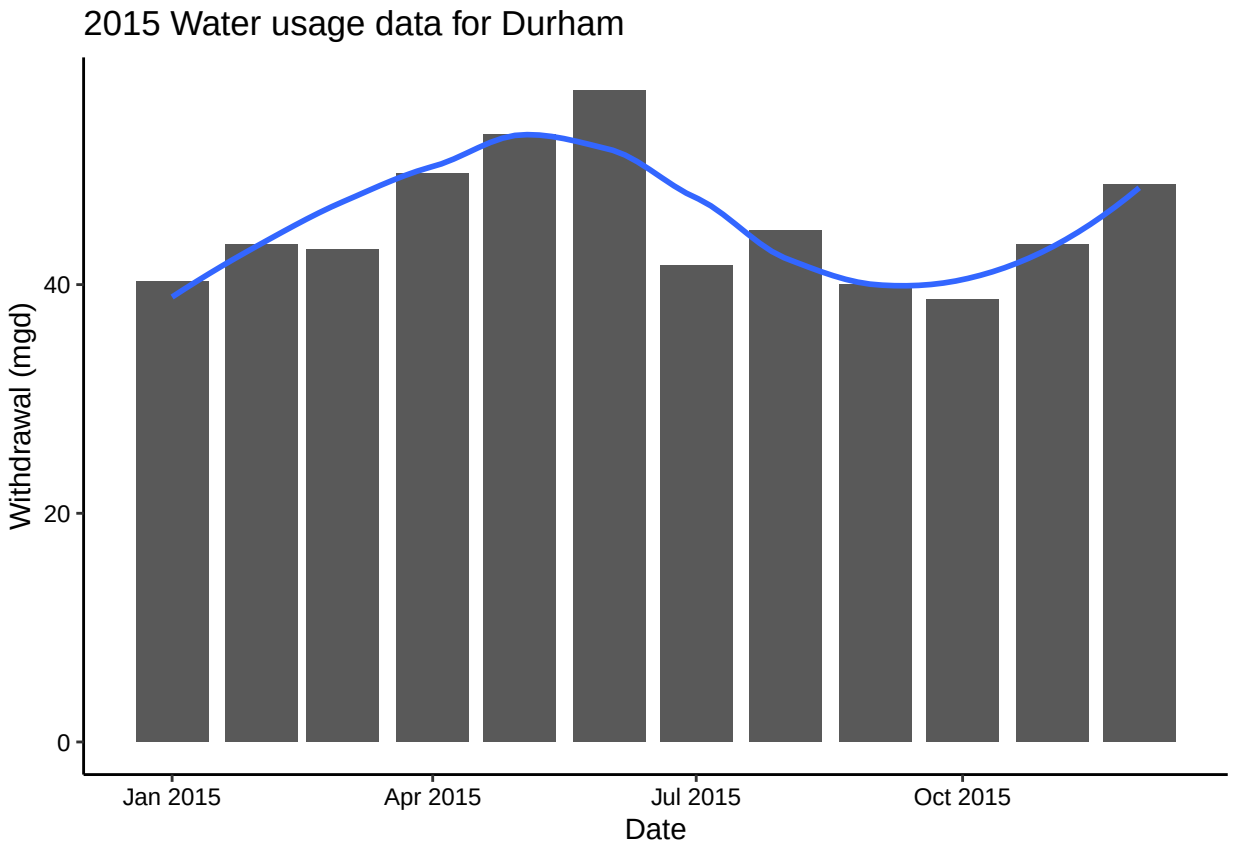

```
####################
```

```
scrape.it <- function(the_pwsid ,the_year){
  #Retrieve the website contents
  base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
  scrape_url <- paste0(base_url,the_pwsid,'&year=',the_year)
  webpage <- read_html(scrape_url)

 #Set the element address variables (determined in the previous step)
  water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  max_withdrawals_tag <- 'th~ td+ td'
  pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'

  #Scrape the data items
  water.system.name <- webpage %>% html_nodes(water_system_name_tag) %>% html_text()
  ownership <- webpage %>% html_nodes(ownership_tag) %>%  html_text()
  max.withdrawals.mgd <- webpage %>% html_nodes(max_withdrawals_tag) %>% html_text()
  pwsid <- webpage %>% html_nodes(pwsid_tag) %>% html_text()

  #Convert to a dataframe
  scraped_df <- data.frame("Month" = month,
                           "Year" = rep(the_year,12),
                           "max.withdrawals.mgd" = as.numeric(max.withdrawals.mgd)) %>%
    mutate(water.system.name = !!water.system.name,
           Ownership = !!ownership,
           PWSID = !!pwsid,
           Date = my(paste(Month,"-",Year)))

  #Pause for a moment - scraping etiquette
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

  #Return the dataframe
  return(scraped_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(the_pwsid = '03-32-010', the_year = "2015")

#the_year <- 2015
ggplot(df2015,aes(x=Date, y=max.withdrawals.mgd, group = 1)) +
  geom_col() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(the_year,"Water usage data for",water.system.name),
       y="Withdrawal (mgd)",
       x="Date")
```

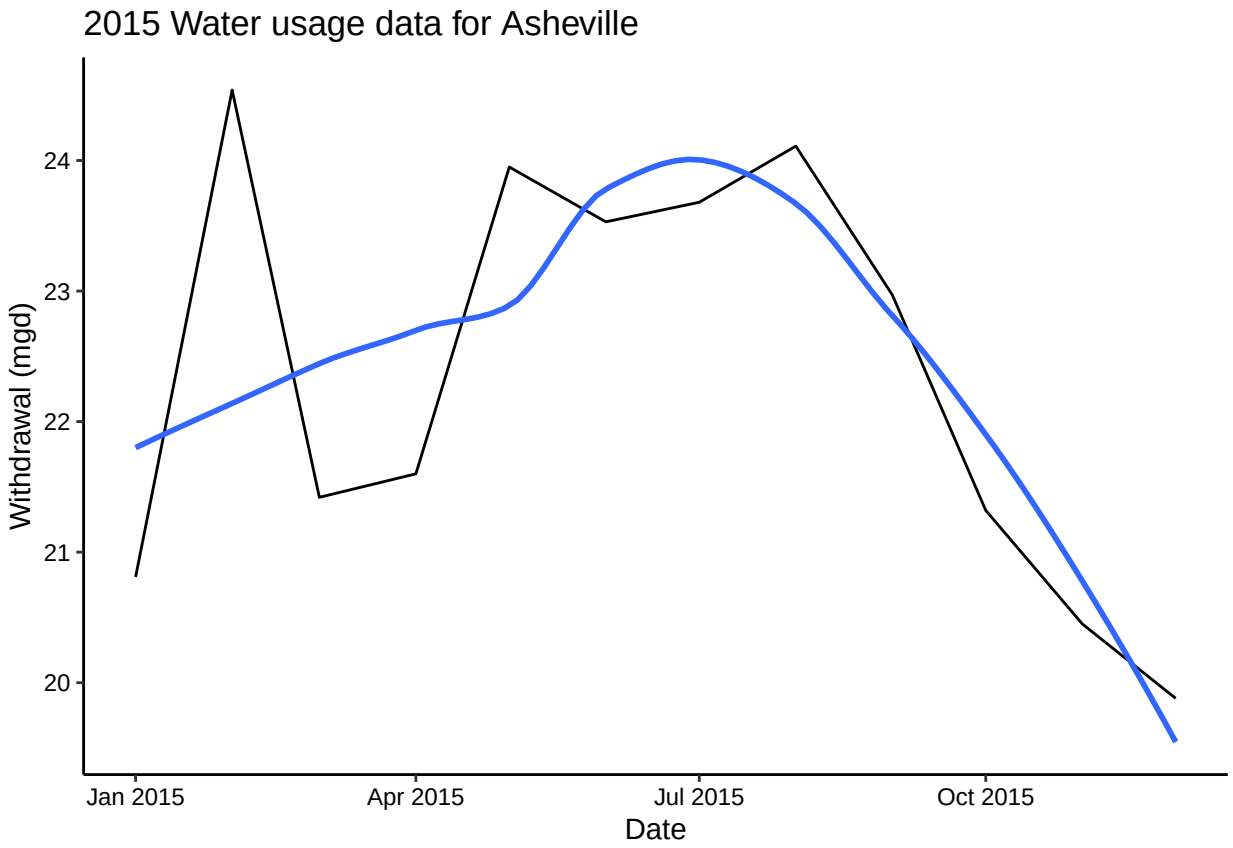
```
## `geom_smooth()` using formula 'y ~ x'
```



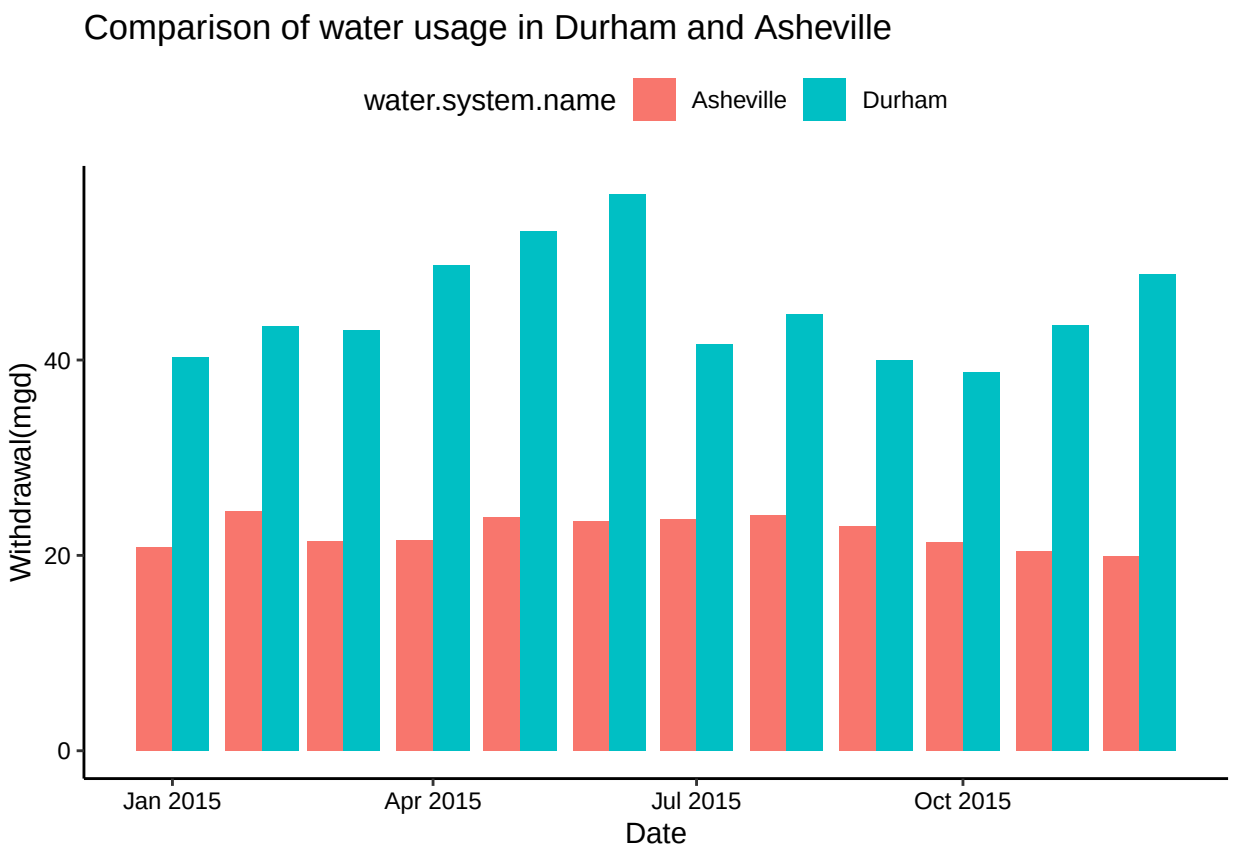

8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares the Asheville to Durham's water withdrawals.

```
#8

avldf2015 <- scrape.it(the_pwsid = '01-11-010', the_year = '2015')

ggplot(avldf2015, aes(x=Date, y=max.withdrawals.mgd, group = 1)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(the_year,"Water usage data for",avldf2015$water.system.name),
       y="Withdrawal (mgd)",
       x="Date")
```

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

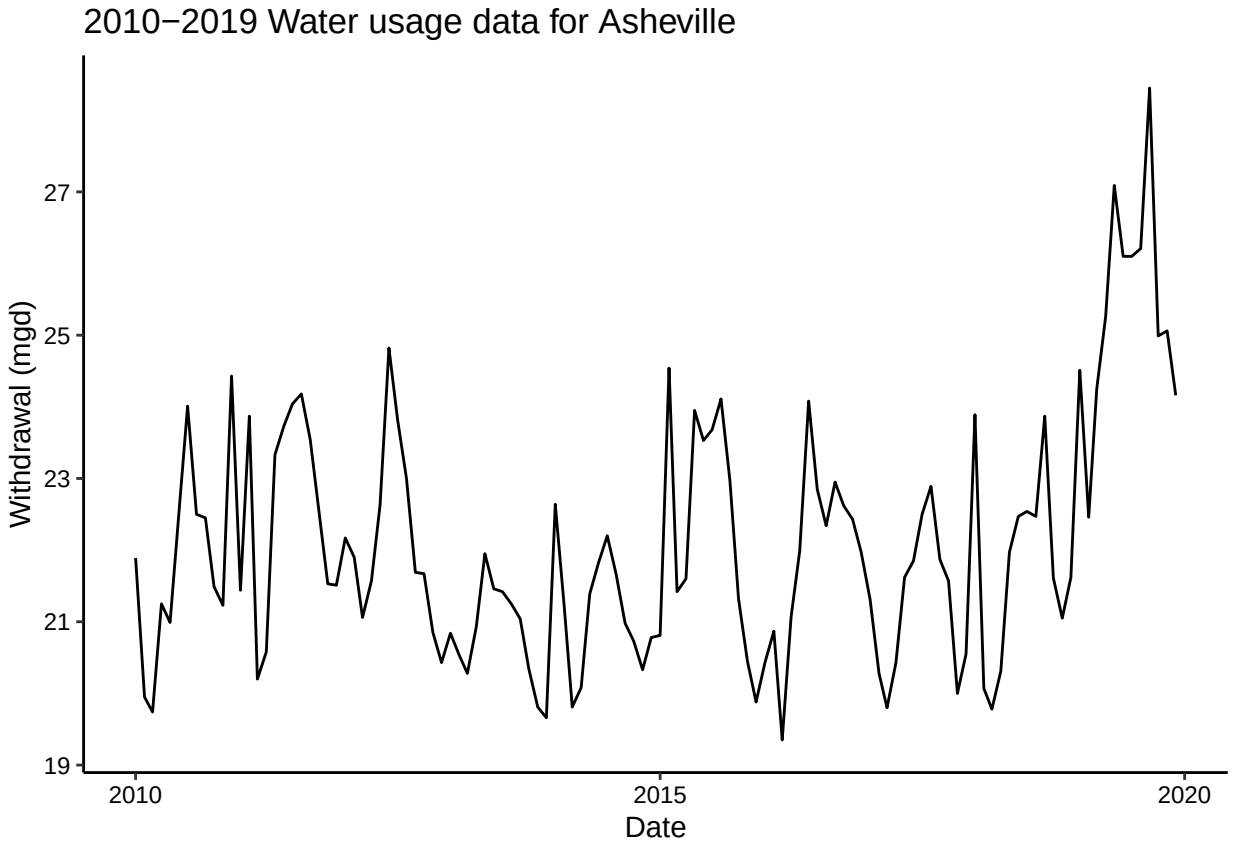



```
combined_df <- bind_rows(avldf2015,df2015)

ggplot(combined_df, aes(x = Date, y=max.withdrawals.mgd, fill = water.system.name)) +
  geom_col(position = "dodge") +
  labs(title = "Comparison of water usage in Durham and Asheville", 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
year_range <- seq(2010,2019)
ashevilledf <- year_range %>% map(scrape.it,the_pwsid = '01-11-010') %>% bind_rows()

ggplot(ashevilledf,aes(x=Date, y=max.withdrawals.mgd, group = 1)) +
  geom_line() +
  labs(title = paste("2010-2019 Water usage data for",ashevilledf$water.system.name),
       y="Withdrawal (mgd)",
       x="Date")
```




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

There does appear to be a trend of increasing water usage between 2015 and 2020 - probably a result of the booming tourism and brewery industry!!!