Assignment 09: Data Scraping

Yosia Theo Napitupulu

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 setup
getwd()
```

```
## [1] "E:/ENV872/EDA-Fall2022"
```

```
library(tidyverse)
library(lubridate)
library(viridis)
library(ggplot2)
#install.packages("rvest")
library(rvest)

#install.packages("dataRetrieval")
library(dataRetrieval)

#install.packages("tidycensus")
library(tidycensus)

# Set theme
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 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. sraping data from website
the_website <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2021")

the_website
```

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

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

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

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

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

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

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

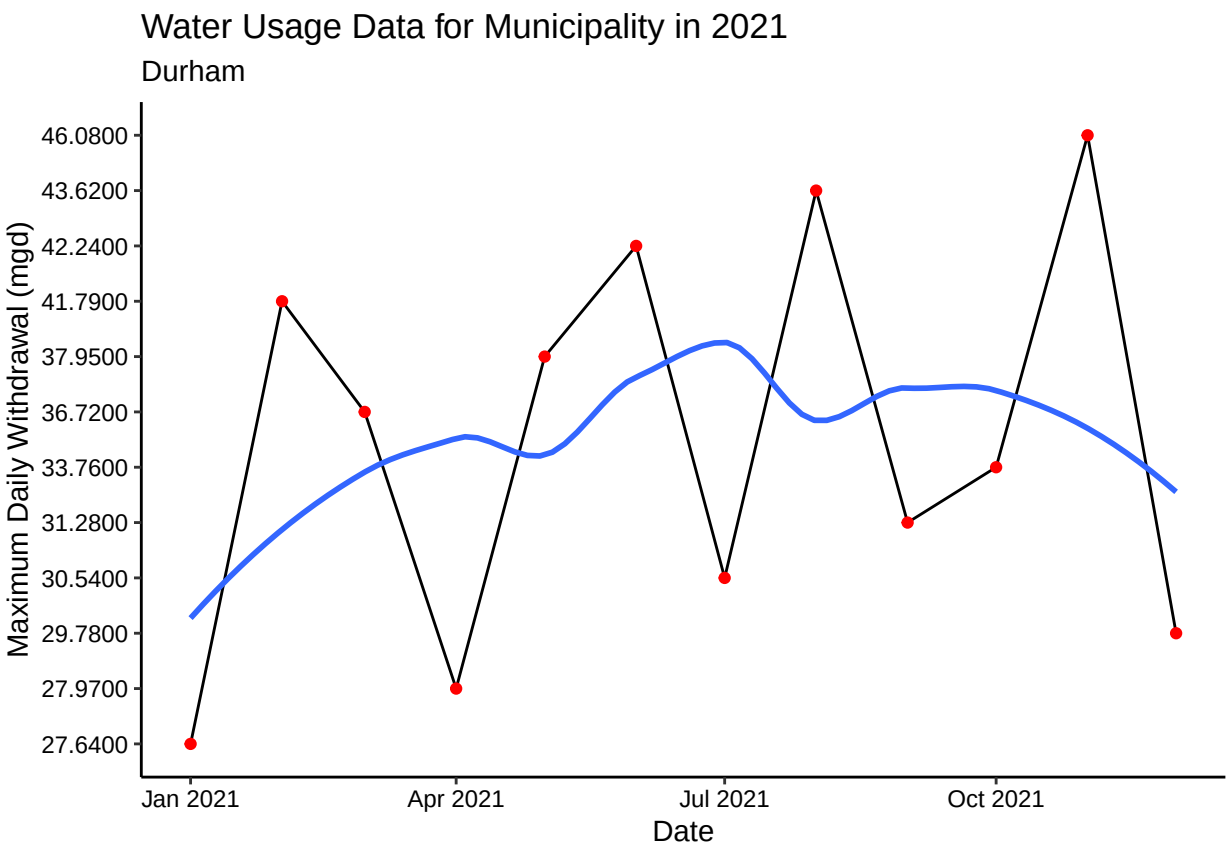
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc...

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

```
#4. creating data frame of withdrawal
months_in_the_year <- c("January", "May", "September", "February", "June",
                        "October", "March", "July", "November", "April",
                        "August", "December")
df_withdrawals <- data.frame("Month" = months_in_the_year,
                    "Year" = rep(2021,12),
                    "Water_System_Name" = water.system.name,
                    "PWSID" = pwsid,
                    "Ownership" = ownership,
                    "Max_Withdrawals_Mgd" = as.numeric(max.withdrawals.mgd))%>%
  mutate(Date = my(paste(Month, "-", Year))) %>%
  arrange(Date)


#5. Plot the max daily withdrawals in 2021
```

```
ggplot(df_withdrawals, aes(x = Date, y = max.withdrawals.mgd, group = 1)) +   geom_line() +
  geom_point(y = max.withdrawals.mgd, color = "red") +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = paste("Water Usage Data for Municipality in 2021"),
       subtitle = water.system.name,
       y = "Maximum Daily 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 (pwsid) scraped**.

```
#6. Construct a scraping function
scrape.it <- function(the_year, pwsid)
{
if (the_year %in% c(1997, 2002, 2006:2021))
{
# Setting inputs for creating the function
# Constructing the scraping web address, i.e. its URL
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
the_scrape_url <- paste0(the_base_url, pwsid, '&', 'year=',the_year)
# Retrieving the website contents
```

```
the_website <- read_html(the_scrape_url)

# Setting the elements address variables (determined under question # 3)
  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"

#Scrapping the data items
  water.system.name <- the_website %>% html_nodes(water.system.name_tag)%>%
    html_text()
  pswid <- 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()

# Constructing a dataframe from the scraped data
# Setting the months in the order they appear in the website
  months_in_the_year <- c("January", "May", "September", "February",
                          "June", "October", "March", "July", "November",
                          "April", "August", "December")

df_withdrawals <- data.frame("Month"= months_in_the_year,
                    "Year" = rep(the_year,12),
                    "Water_system_name" = water.system.name,
                    "PWSID"= pwsid,
                    "Ownership"= ownership,
                    "Max_withdrawals_mgd" =
                      as.numeric(max.withdrawals.mgd)) %>%
  mutate(Date = my(paste(Month,"-",Year))) %>%
  arrange(Date)

return(df_withdrawals)
}

else
return(paste("No data available for the year :", the_year))
}

# Check the data in 2000
scrape.it(2000, "03-32-010")
```

```
## [1] "No data available for the year : 2000"
```

```
# Check the data in 2019
scrape.it(2019, "03-32-010")
```

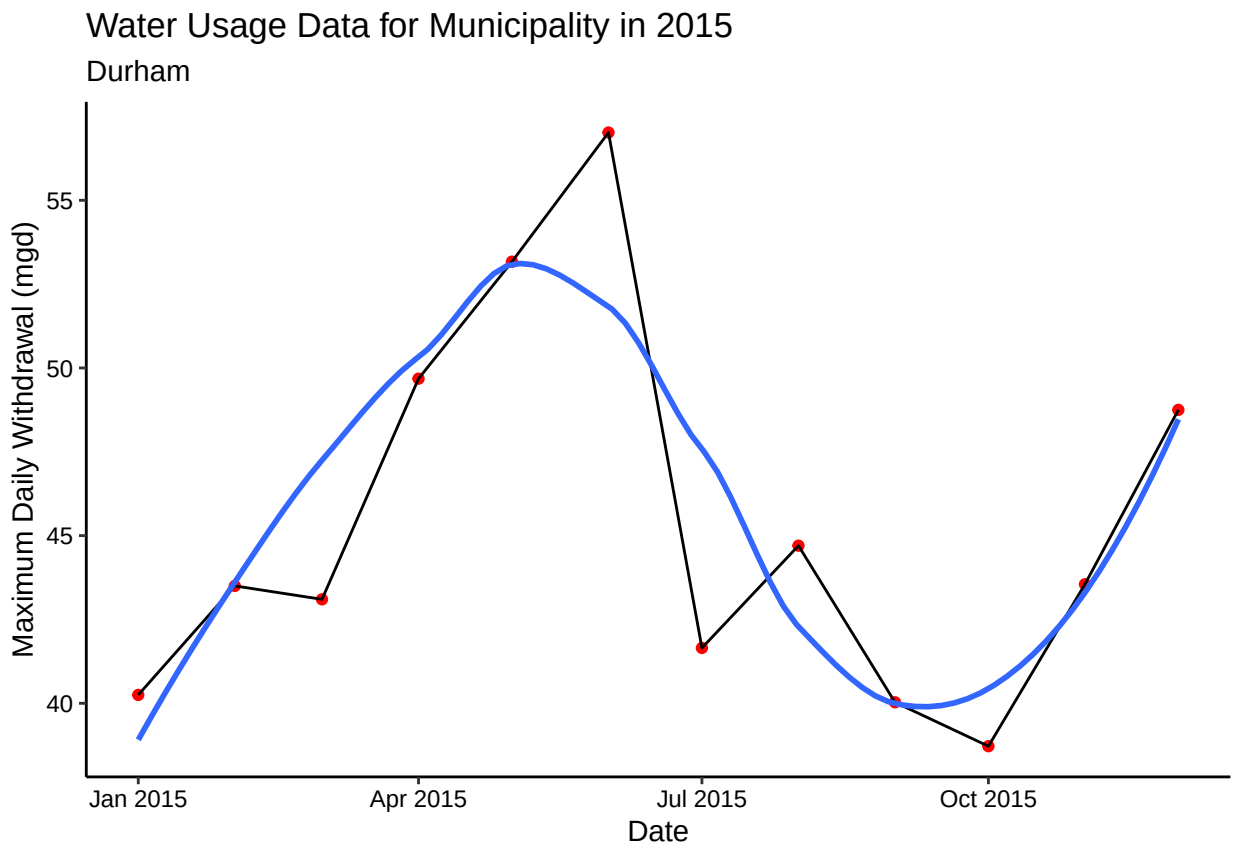
```
##        Month Year Water_system_name     PWSID    Ownership Max_withdrawals_mgd
## 1    January 2019            Durham 03-32-010 Municipality               29.62
## 2   February 2019            Durham 03-32-010 Municipality               32.39
## 3      March 2019            Durham 03-32-010 Municipality               36.43
```

```
## 4      April 2019           Durham 03-32-010 Municipality              32.60
## 5        May 2019           Durham 03-32-010 Municipality              35.73
## 6       June 2019           Durham 03-32-010 Municipality              37.86
## 7       July 2019           Durham 03-32-010 Municipality              46.02
## 8     August 2019           Durham 03-32-010 Municipality              42.05
## 9  September 2019           Durham 03-32-010 Municipality              54.07
## 10   October 2019           Durham 03-32-010 Municipality              44.35
## 11  November 2019           Durham 03-32-010 Municipality              36.06
## 12  December 2019           Durham 03-32-010 Municipality              31.20
##          Date
## 1  2019-01-01
## 2  2019-02-01
## 3  2019-03-01
## 4  2019-04-01
## 5  2019-05-01
## 6  2019-06-01
## 7  2019-07-01
## 8  2019-08-01
## 9  2019-09-01
## 10 2019-10-01
## 11 2019-11-01
## 12 2019-12-01
```

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

```
#7. Extract and plot max daily withdrawals for Durham in 2015
Durham_max_withdrawals_2015 <- scrape.it(2015,'03-32-010')
view(Durham_max_withdrawals_2015)

# Plot the max daily withdrawals Durham in 2015
ggplot(Durham_max_withdrawals_2015,
       aes(x = Date, y = Max_withdrawals_mgd)) +
  geom_point(color = "red") +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = paste("Water Usage Data for Municipality in 2015"),
       subtitle = water.system.name,
       y = "Maximum Daily Withdrawal (mgd)",
       x = "Date")
```

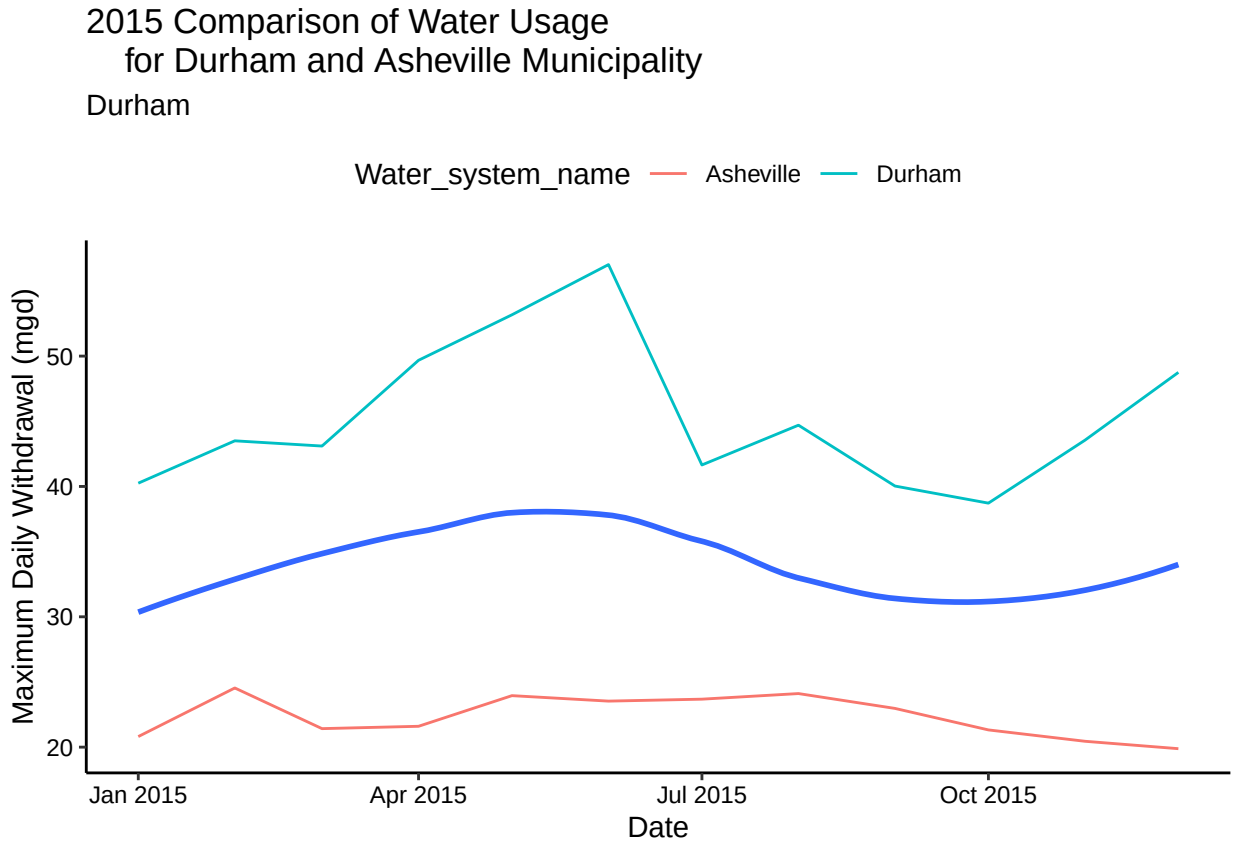
```
## `geom_smooth()` using formula = 'y ~ x'
```



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 Asheville's to Durham's water withdrawals.

```
#8. Fetch and plot Asheville - Durham
# Extract data for Asheville in 2015
Asheville_max_withdrawals_2015 <- scrape.it(2015,'01-11-010')
view(Asheville_max_withdrawals_2015)

# Combining the two dataframes and plotting comparision plots
combined_Ash_Durham <- rbind(Durham_max_withdrawals_2015, Asheville_max_withdrawals_2015)

# Plot
ggplot(combined_Ash_Durham,aes(x = Date, y = Max_withdrawals_mgd)) +
  geom_line(aes(color = Water_system_name)) +
  geom_smooth(method="loess",se = FALSE) +
  labs(title = paste(combined_Ash_Durham$Year, "Comparison of Water Usage
    for Durham and Asheville Municipality"),
    subtitle = water.system.name,
       y = "Maximum Daily Withdrawal (mgd)",
       x = "Date")
```

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



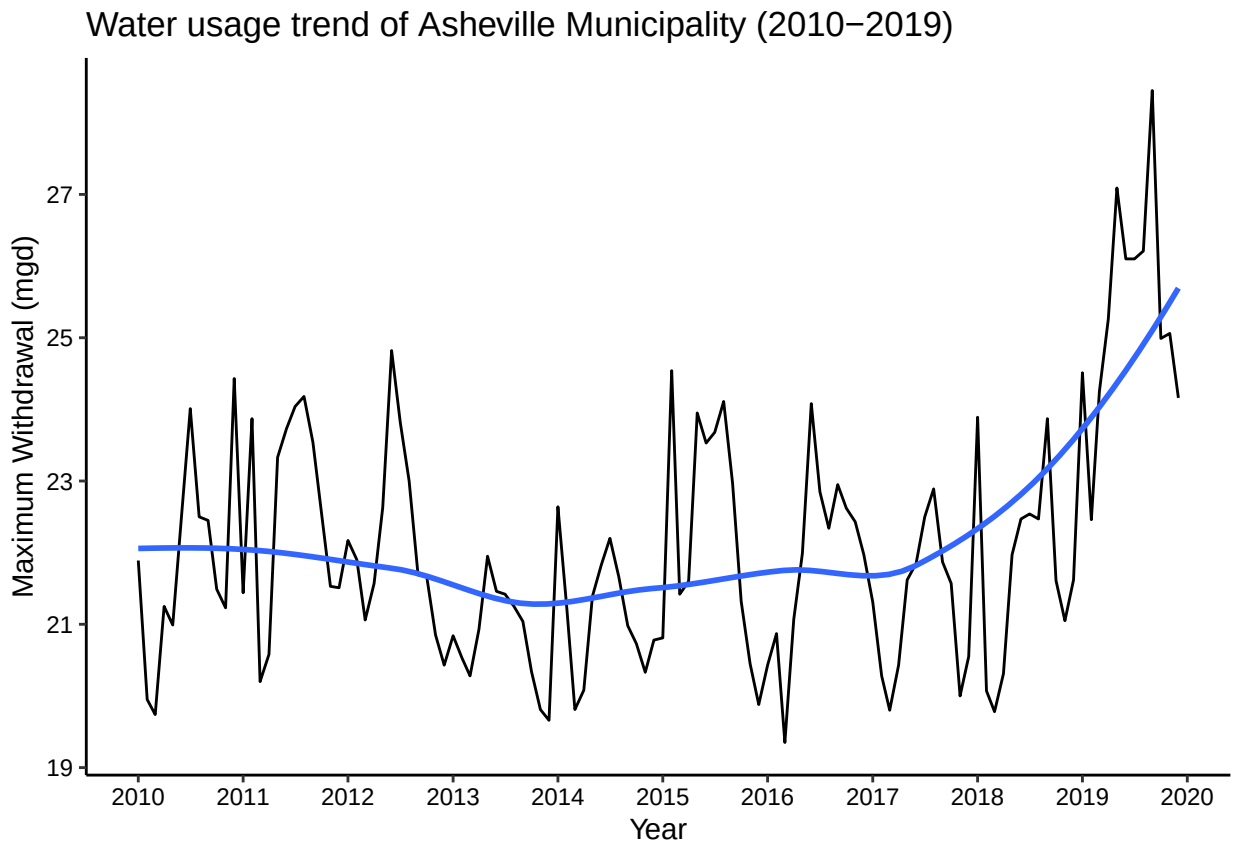
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

   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.

```
#9 Fetch and plot Asheville from 2010 to 2019
the_years = rep(2010:2019)
pwsid = '01-11-010'

# Using lapply in applying the scrape function
#Ash_max_withdrawals_2010_2019 <- lapply(X = the_years,
 #                           FUN = scrape.it, pwsid=pwsid)

Ash_max_withdrawals_2010_2019 <- map2(the_years, pwsid, scrape.it)

# change dataframes into a single dataframe
Ash_max_withdrawals_2010_2019.df <- bind_rows(Ash_max_withdrawals_2010_2019)

# Plot
ggplot(Ash_max_withdrawals_2010_2019.df,
       aes(x=Date, y=Max_withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se = FALSE) +
```

```
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(title = paste("Water usage trend of Asheville Municipality (2010-2019)"),
       subtitle = Ash_max_withdrawals_2010_2019$Water.system.name,
       y="Maximum Withdrawal (mgd)",
       x="Year")
```

```
## `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?

Answer: Yes, it shows a different trend over the periods. The maximum water withdrawal of Asheville from 2010 to 2014 shows a slight decreasing trend. In 2014 to 2017, the trend shows a slight increasing trend, but after 2017 to 2019, the trend show a significant increasing trend. So in general from 2010 to 2019, the water usage has a slight declining trend at the begining, then shifted into an increasing trends from 2017 to 2019.