Assignment 10: Data Scraping

Chunyi Xu

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Rename this file `<FirstLast>_A10_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 your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Check your working directory

```r
#1
#Loading necessary packages
library (tidyverse)
library (rvest)
library (ggplot2)
library (here)
```

```
## Warning: package 'here' was built under R version 4.3.3
```

```r
#Checking working directory is the project folder
here()
```

```
## [1] "/Users/xuchunyi"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2024 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=2024

Indicate this website as the as the URL to be scraped. (In other words, read the contents into an `rvest` webpage object.)

```
#2
#Indicating the website as the as the URL to be scraped
Durham_LWSP_Web <- read_html(
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024')
Durham_LWSP_Web
```

```
## {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
- PWSID
- Ownership
- From the "3. Water Supply Sources" section:
- Maximum Day 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)".

```
#3
#Scraping the four values and assigning them to four separate variables
#Water system name
Water_System_Name <- Durham_LWSP_Web %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
Water_System_Name
```

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

```
#PWSID
PWSID <- Durham_LWSP_Web %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
PWSID
```

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

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

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

```
#Maximum Day Use (MGD) - for each month
MGD <- Durham_LWSP_Web %>%
  html_nodes("th~ td+ td") %>%
  html_text()
MGD
```

```
##  [1] "34.5000" "36.0600" "37.3300" "32.1000" "46.6500" "37.3600" "38.2000"
##  [8] "41.9000" "36.5800" "36.7300" "42.9600" "34.4500"
```

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.

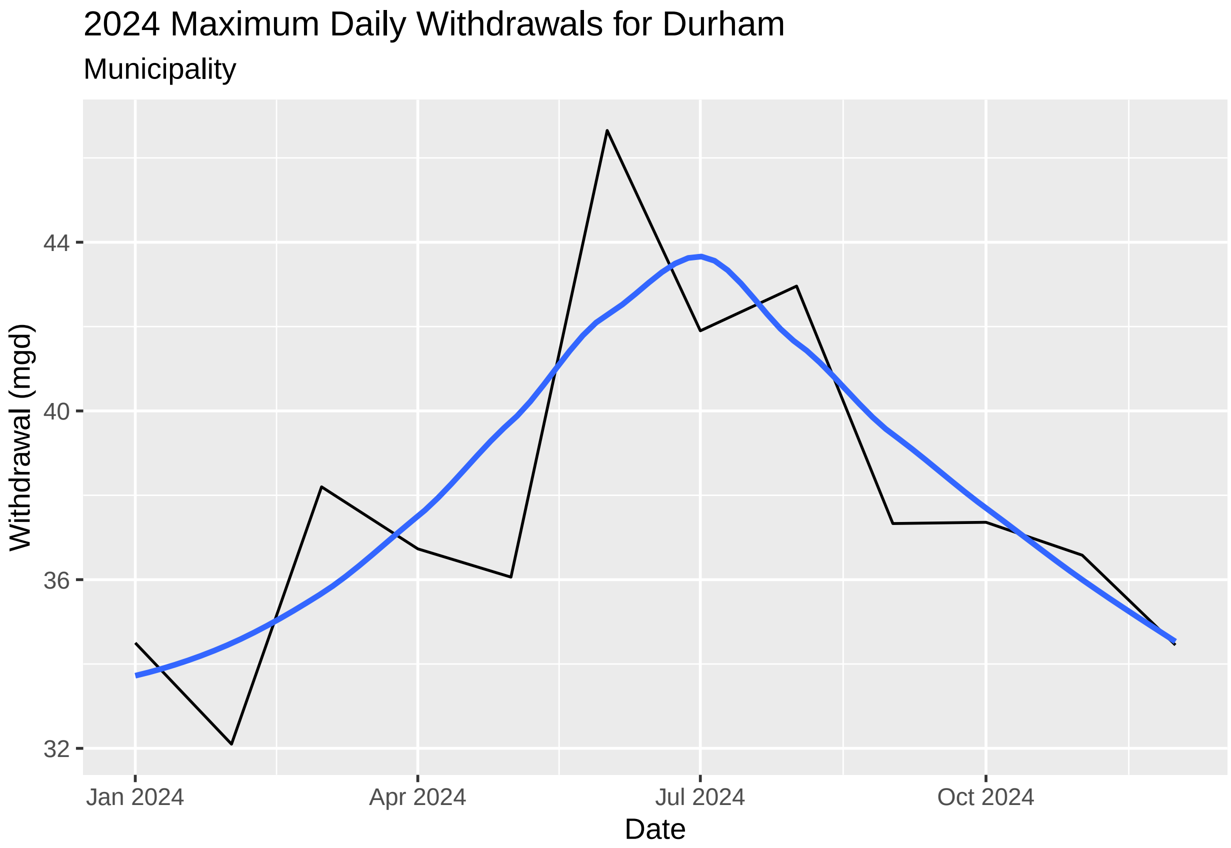
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc… Or, you could scrape month values from the web page…

5. Create a line plot of the maximum daily withdrawals across the months for 2024, making sure, the months are presented in proper sequence.

```
#4
#Create a dataframe of Durham LWSP
DF_LWSP <- data.frame("Month" = c("Jan", "May", "Sept", "Feb","Jun","Oct",
                                  "Mar", "Jul","Nov", "Apr", "Aug", "Dec"),
                             "Year" = rep(2024,12),
                             "Max_Day_Use_mgd" = as.numeric(MGD))

#Modify the dataframe to include other variables as well as the date (as date object)
DF_LWSP <- DF_LWSP %>%
  mutate(Water_System_Name = !!Water_System_Name,
         PWSID = !!PWSID,
         Ownership = !!Ownership,
         Date = my(paste(Month,"-",Year)))

#Reordering the months
DF_LWSP$Month <- factor(DF_LWSP$Month,
                    levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                               "Jul", "Aug", "Sept", "Oct", "Nov", "Dec"),
                    ordered = TRUE)
```

```
DF_LWSP_Ordered <- DF_LWSP[order(DF_LWSP$Month), ]

#5
#Plotting the maximum daily withdrawals across the months for 2024
ggplot(DF_LWSP_Ordered,aes(x=Date,y=Max_Day_Use_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2024 Maximum Daily Withdrawals 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 with two input - "PWSID" and "year" - that:

- Creates a URL pointing to the LWSP for that PWSID for the given year
- Creates a website object and scrapes the data from that object (just as you did above)
- Constructs a dataframe from the scraped data, mostly as you did above, but includes the PWSID and year provided as function inputs in the dataframe.
- Returns the dataframe as the function's output

```
#6.
#Establishing a function
scrape.it <- function(PWSID, year){
  #Constructing the scraping web address
  the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
  the_scrape_url <- paste0(the_base_url, PWSID, '&year=', year)
  print(the_scrape_url)

  #Retrieving the website contents
  LWSP_website <- read_html(the_scrape_url)

  #Setting the element address variables
  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)'
  MDG_tag <- 'th~ td+ td'

  #Scraping the data items
  Water_System_Name_Function <- LWSP_website %>% html_nodes(Water_System_Name_tag) %>% html_text()
  PWSID_Function <- LWSP_website %>%   html_nodes(PWSID_tag) %>%  html_text()
  Ownership_Function <- LWSP_website %>% html_nodes(Ownership_tag) %>% html_text()
  MDG_Function <- LWSP_website %>% html_nodes(MDG_tag) %>% html_text()

  #Constructing a dataframe from the scraped data
  DF_withdrawals_Function <- data.frame("Month" = c("Jan", "May", "Sept", "Feb",
                                                    "Jun","Oct", "Mar", "Jul",
                                                    "Nov", "Apr", "Aug", "Dec"),
                             "Year" = rep(year,12),
                             "Max_Day_Use_mgd" = as.numeric(MDG_Function)) %>%
    mutate(Water_System_Name = !!Water_System_Name_Function,
         PWSID = !!PWSID_Function,
         Ownership = !!Ownership_Function,
         Date = my(paste(Month,"-",Year)))

    #Reordering the months
  DF_withdrawals_Function$Month <- factor(DF_withdrawals_Function$Month,
                   levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                              "Jul", "Aug", "Sept", "Oct", "Nov", "Dec"),
                   ordered = TRUE)

  DF_withdrawals_Function_Ordered <- DF_withdrawals_Function[order(DF_withdrawals_Function$Month), ]

   return(DF_withdrawals_Function_Ordered)
}
```

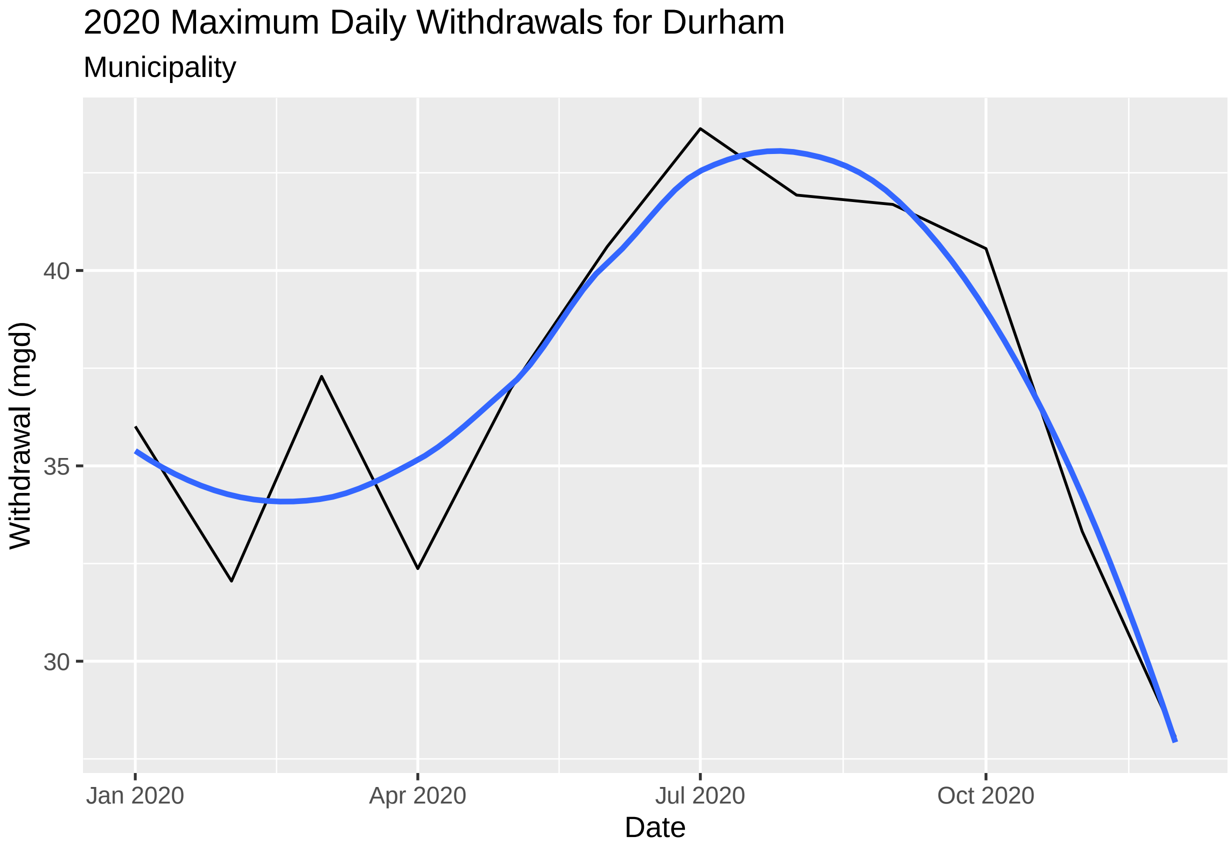
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2020

```
#7
#Running the function
Durham_2020 <- scrape.it('03-32-010',2020)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020"
```

```
view(Durham_2020)

#Plotting the maximum daily withdrawals for Durham across the months for 2020
ggplot(Durham_2020,aes(x=Date,y=Max_Day_Use_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2020 Maximum Daily Withdrawals for Durham"),
       subtitle = Ownership,
       y="Withdrawal (mgd)",
       x="Date")
```

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

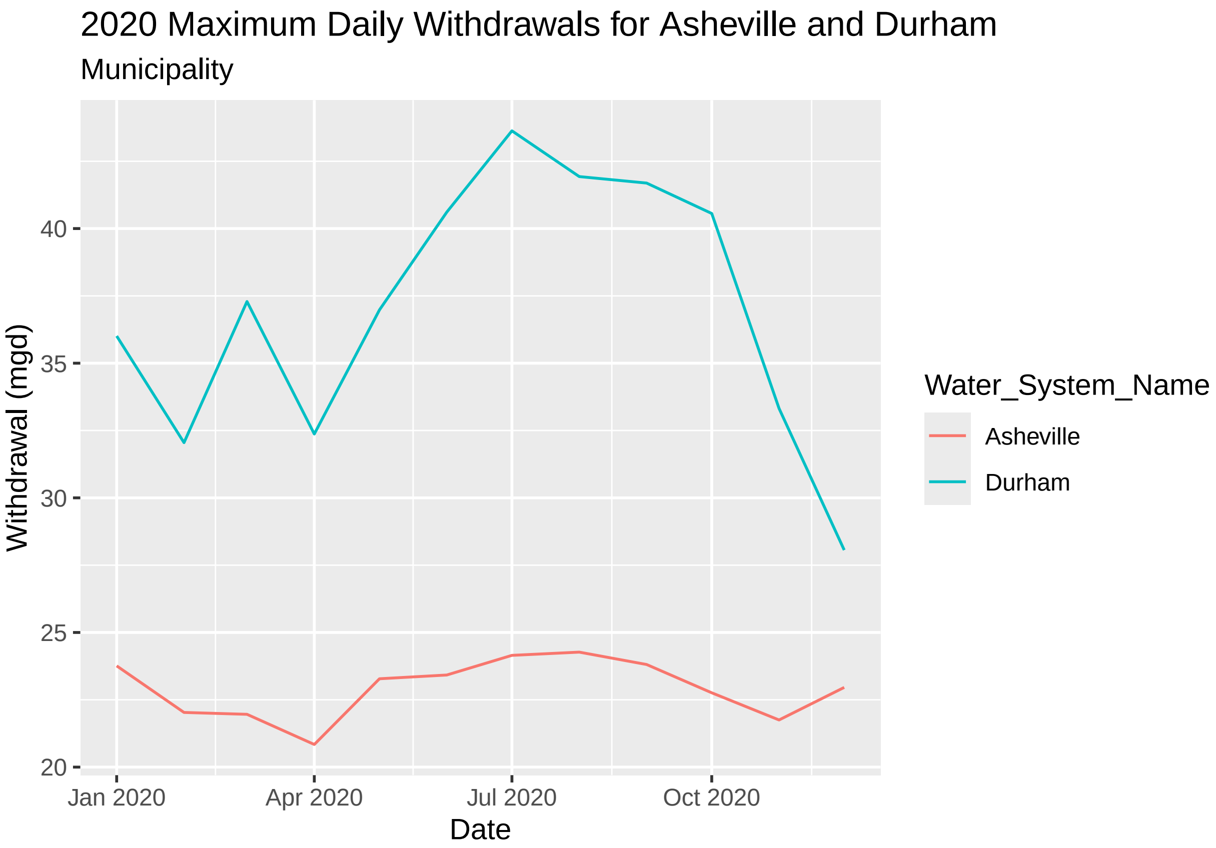


8. Use the function above to extract data for Asheville (PWSID = ‘01-11-010’) in 2020. Combine this data with the Durham data collected above and create a plot that compares Asheville’s to Durham’s water withdrawals.

```
#8
#Running the function
Asheville_2020 <- scrape.it('01-11-010',2020)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2020"
```

```r
view(Asheville_2020)

#Combining the two datasets
combined_Durham_Ash <- rbind(Durham_2020, Asheville_2020)

#Plotting Asheville's and Durham's water withdrawals
ggplot(combined_Durham_Ash,aes(x=Date, y=Max_Day_Use_mgd, color = Water_System_Name)) +
  geom_line() +
  labs(title = paste("2020 Maximum Daily Withdrawals for Asheville and Durham"),
       subtitle = Ownership,
       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 2018 thru 2023.Add a smoothed line to the plot (method = 'loess').

   TIP: See Section 3.2 in the "10_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, and use that to construct your plot.

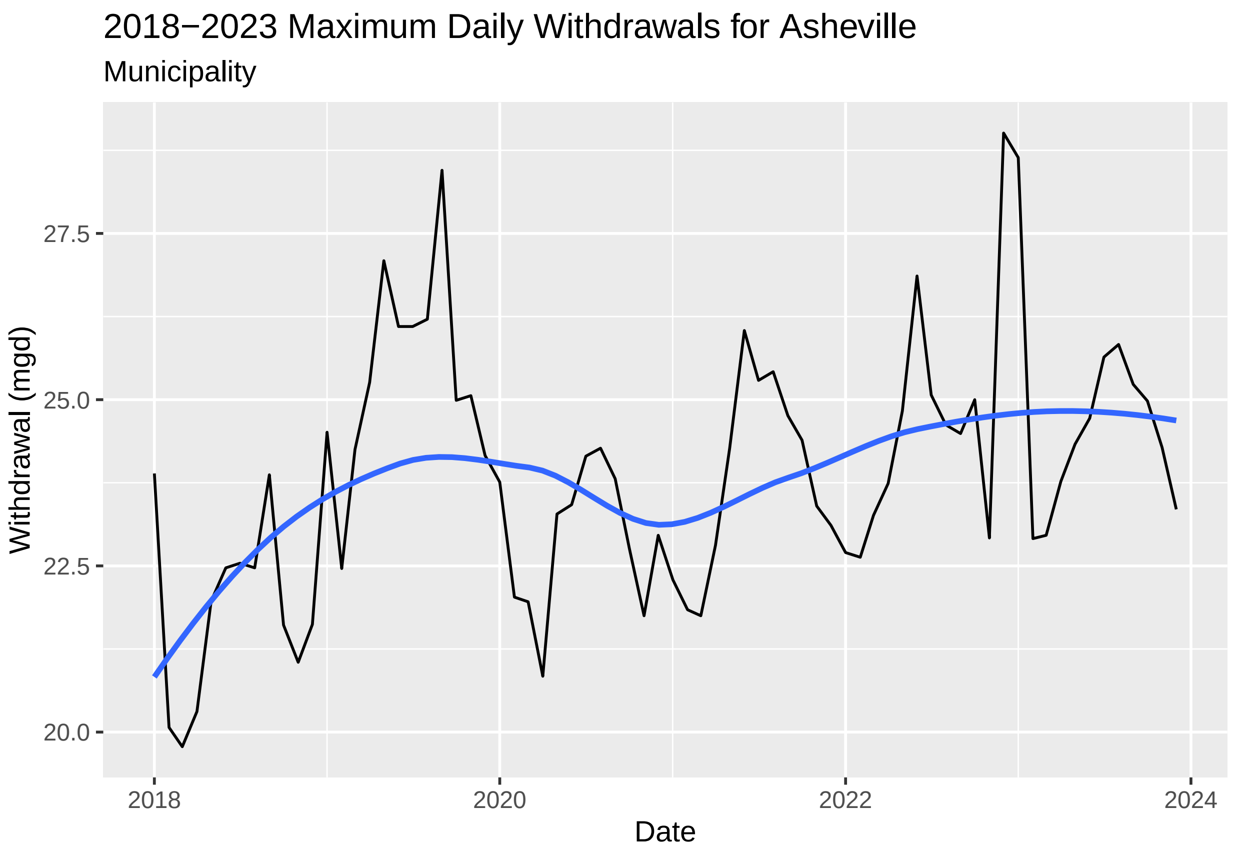
```r
#9
#Create a list of the year we want, the same length as the vector above
year <- rep(2018:2023)
```

```
#Define the PWSID, , the same length as the year above
PWSID <- rep.int('01-11-010',length(year))

#"Map" the "scrape.it" function to retrieve data for all these
Ash_2018_2023 <- map2(PWSID, year, scrape.it)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2018"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2019"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2020"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2021"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2022"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2023"
```

```
#Conflate the returned list of dataframes into a single one
Combined_Ash_2018_2023 <- bind_rows(Ash_2018_2023)

#Plotting Asheville's max daily withdrawal by months for the years 2018 thru 2023
ggplot(Combined_Ash_2018_2023,aes(x=Date,y=Max_Day_Use_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2018-2023 Maximum Daily Withdrawals for Asheville"),
       subtitle = Ownership,
       y="Withdrawal (mgd)",
       x="Date")
```

```
## `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, the graph shows that there is an increasing trend in water usage from 2018 to 2023 in Asheville, although there is a little fluctuation during 2020 to 2022. >