Assignment 10: Data Scraping

Desa Bolger

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

```
#1
library(tidyverse)
library(lubridate)
library(viridis)
library(here)
library(ggplot2)

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

here()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
#looks good

#from class notes
mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"))
theme_set(mytheme)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2022 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=2022

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=2022')
```

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
#Water system name
Water_System_Name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

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


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


#VECTOR
```

```
Vector <- webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()


#MONTH
Month <- webpage %>%
  html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
  html_text()
```

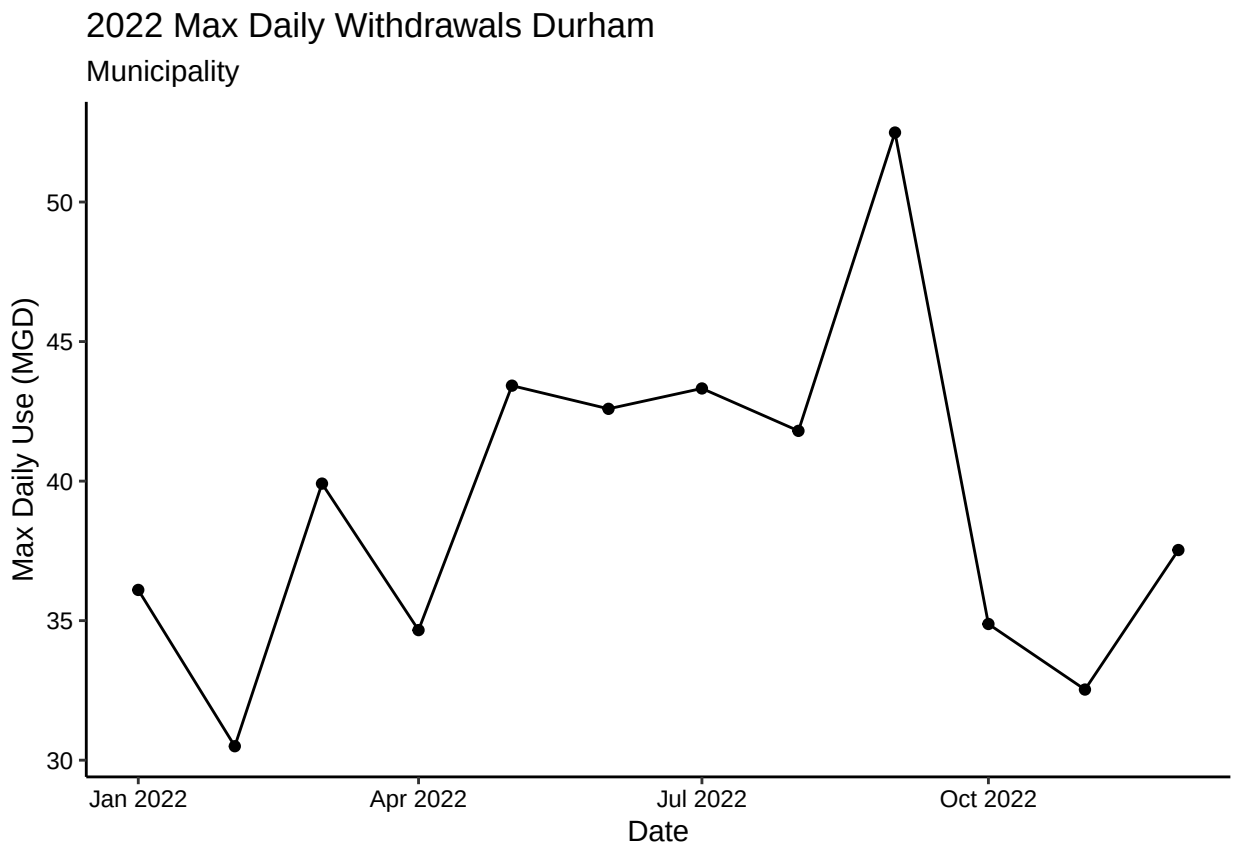
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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 2022

```
#4
df_withdrawals <- data.frame("Month" = Month,
                             "Year" = rep(2022,12),
                             "PWSID" = PWSID,
                             "Water_System_Name" = Water_System_Name,
                             "Ownership" = Ownership,
                             "Vector" = Vector)

df_withdrawals <- df_withdrawals %>%
  mutate(
         Date = my(paste(Month,"-",Year)))


df_withdrawals <- df_withdrawals[order(as.Date(df_withdrawals$Date,
                                               format="%Y/%m/%d")),]
df_withdrawals$Vector <- as.numeric(df_withdrawals$Vector)


#5

ggplot(df_withdrawals, aes(x=Date, y=Vector)) +
  geom_point() +
  geom_path(group = 1) +
  labs(title = paste("2022 Max Daily Withdrawals", Water_System_Name),
       subtitle = Ownership,
       y="Max Daily Use (MGD)",
       x="Date")
```



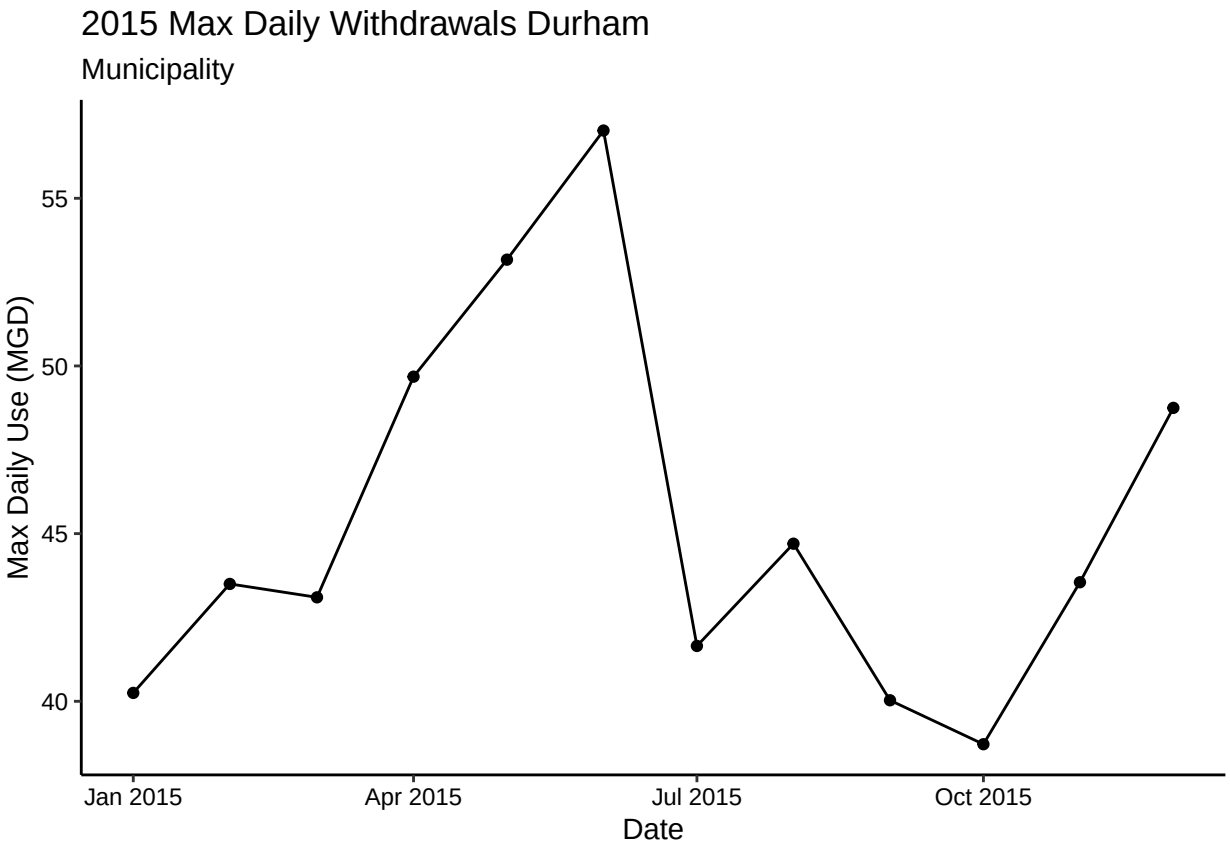
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.

scrape.it <- function(pwsid, the_year){

the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
the_scrape_url <- paste0(the_base_url, pwsid, '&year=', the_year)

#Retrieve the website contents
the_website <- read_html(the_scrape_url)

#Set the element address variables (determined in the previous step)
Water_System_Name <- 'div+ table tr:nth-child(1) td:nth-child(2)'
PWSID <- "td tr:nth-child(1) td:nth-child(5)"
Ownership <- "div+ table tr:nth-child(2) td:nth-child(4)"
Vector <- "th~ td+ td"

#Scrape the data items
Water_System_Name <- the_website %>% html_nodes(Water_System_Name) %>% html_text()
PWSID <- the_website %>% html_nodes(PWSID) %>%  html_text()
Ownership <- the_website %>% html_nodes(Ownership) %>% html_text()
Vector <- the_website %>% html_nodes(Vector) %>% html_text()
```

```
Month <- c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr",
           "Aug", "Dec")

#Construct a dataframe from the scraped data
df_withdrawals <- data.frame("Month" = Month,
                             "Year" = the_year,
                             "PWSID" = PWSID,
                             "Water_System_Name" = Water_System_Name,
                             "Ownership" = Ownership,
                             "Vector" = Vector)

df_withdrawals <- df_withdrawals %>%
  mutate(
         Date = my(paste(Month,"-",Year)))

df_withdrawals <- df_withdrawals[order(as.Date(df_withdrawals$Date,
                                               format="%Y/%m/%d")),]
df_withdrawals$Vector <- as.numeric(df_withdrawals$Vector)

plotting <- ggplot(df_withdrawals, aes(x=Date, y=Vector)) +
  geom_point() +
  geom_path(group =1) +
  labs(title = paste(the_year,"Max Daily Withdrawals", Water_System_Name),
       subtitle = Ownership,
       y="Max Daily Use (MGD)",
       x="Date")
  Sys.sleep(1)

print(plotting)
  return(df_withdrawals)

}
```

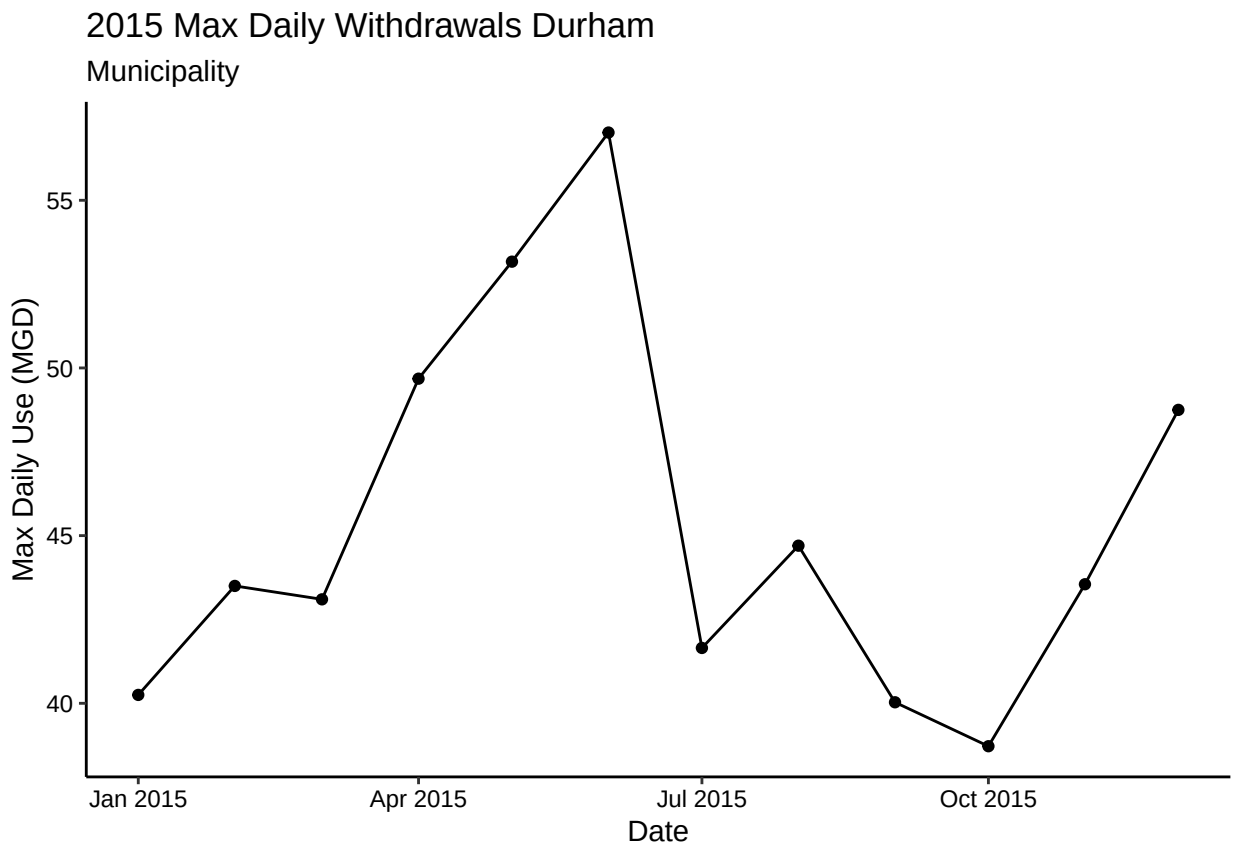
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
NewSet <-scrape.it("03-32-010", 2015)
```

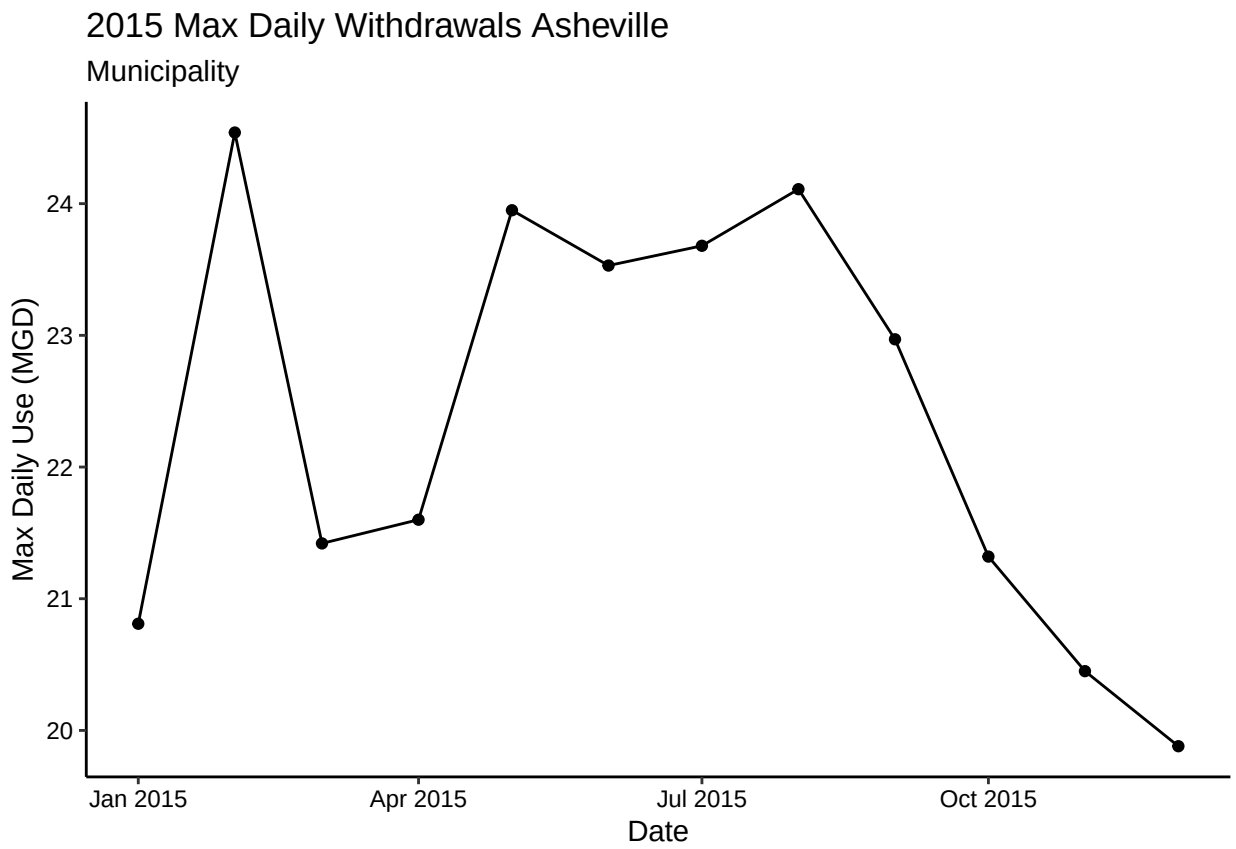


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

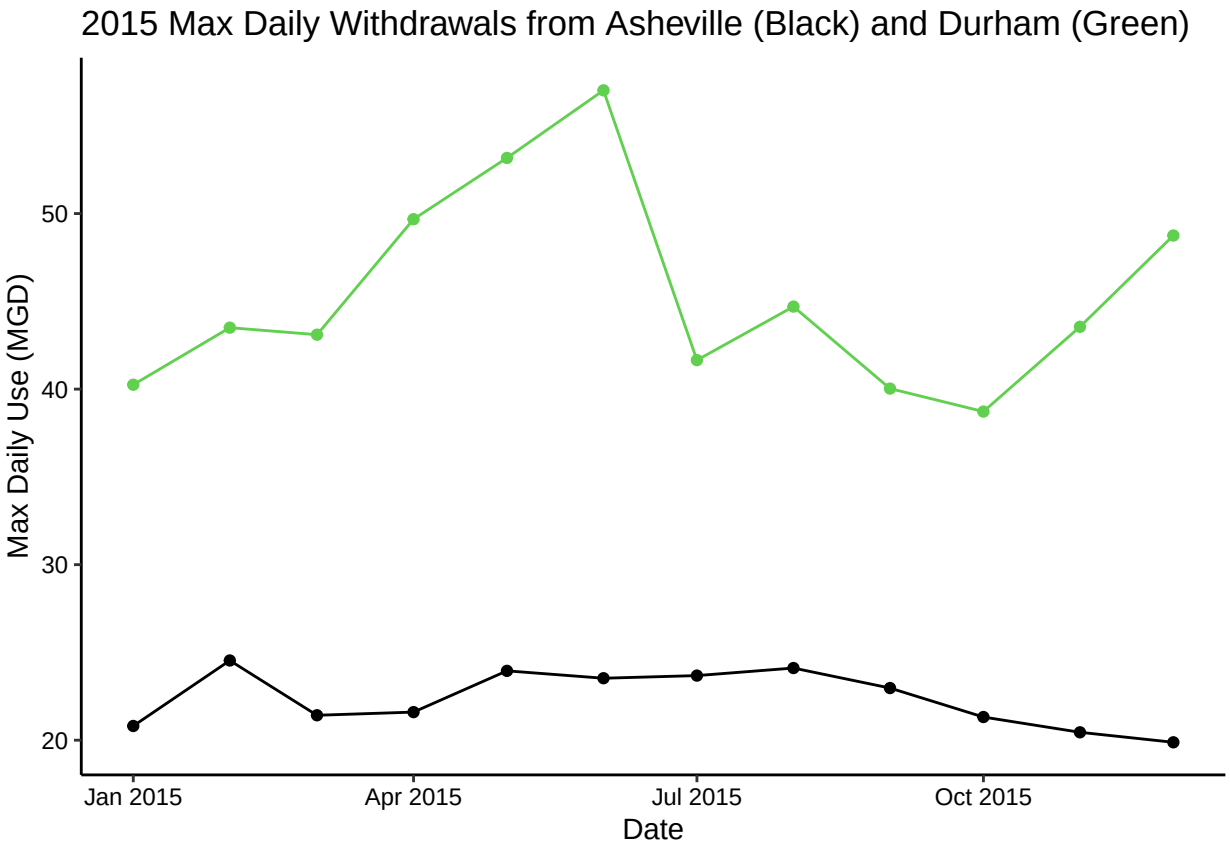
```
#8
NewSet <-scrape.it("03-32-010", 2015)
```




```
NewSet$Vector <- as.numeric(NewSet$Vector)

Asheville <-scrape.it("01-11-010", 2015)
```




```
Asheville$Vector <- as.numeric(Asheville$Vector)


ggplot() +
  geom_point(NewSet, mapping = aes(x=Date, y=Vector),
             color = "03-32-010") +
  geom_path(NewSet, mapping = aes(x=Date, y=Vector, group = 1),
            color = "03-32-010") +
  geom_point(Asheville, mapping = aes(x=Date, y=Vector),
             color = "01-11-010") +
  geom_path(Asheville, mapping = aes(x=Date, y=Vector, group = 1),
            color = "01-11-010") +
  labs(title = paste(NewSet$Year, "Max Daily Withdrawals from",
  Asheville$Water_System_Name, "(Black) and", NewSet$Water_System_Name,
  "(Green)"),
       y="Max Daily Use (MGD)",
       x="Date")
```




```
#add legend
```

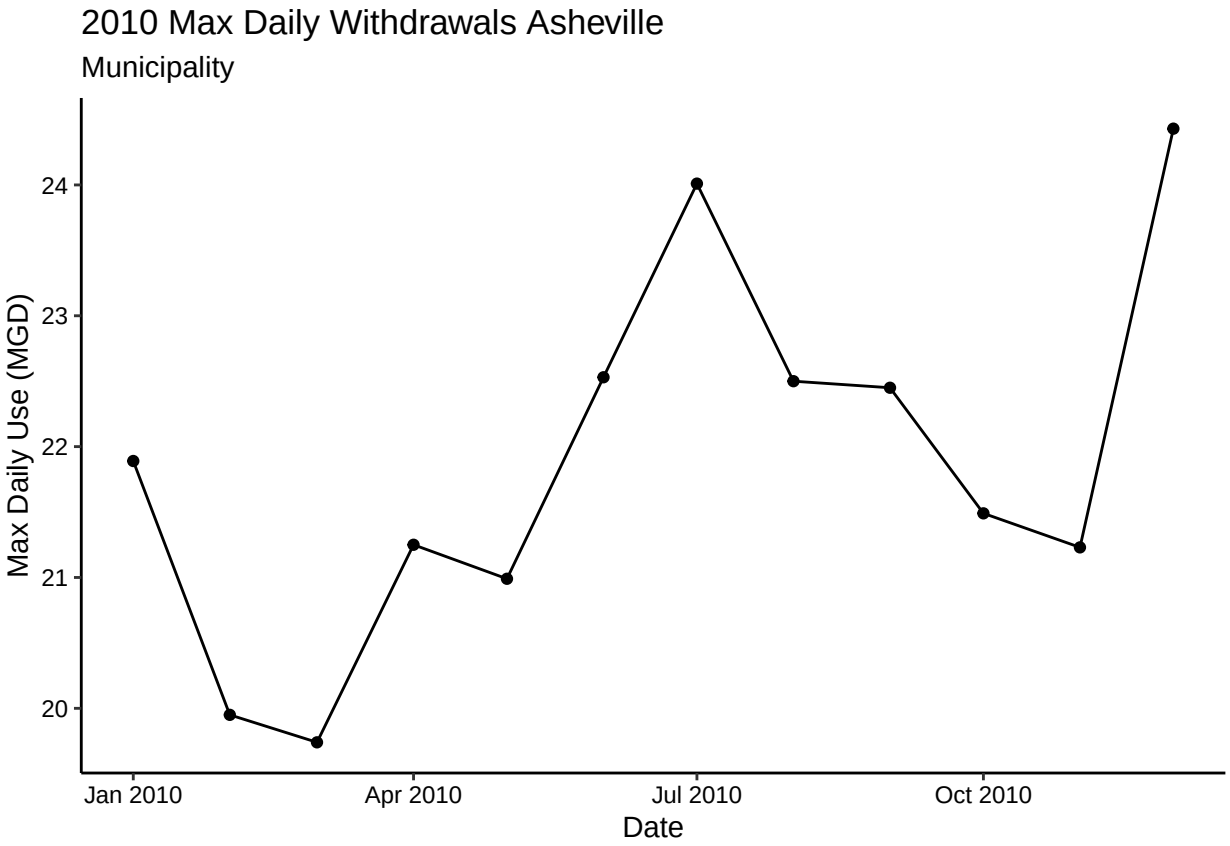
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

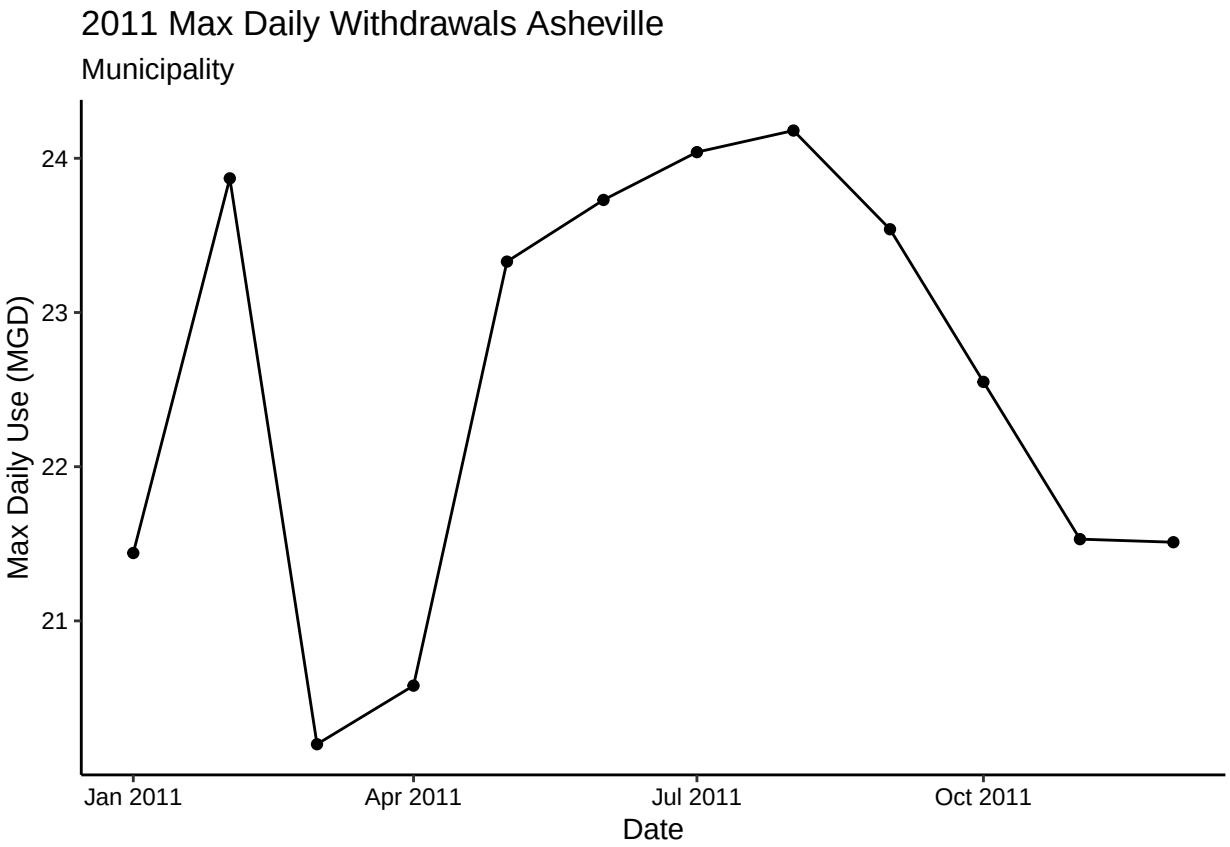
   TIP: See Section 3.2 in the "10_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

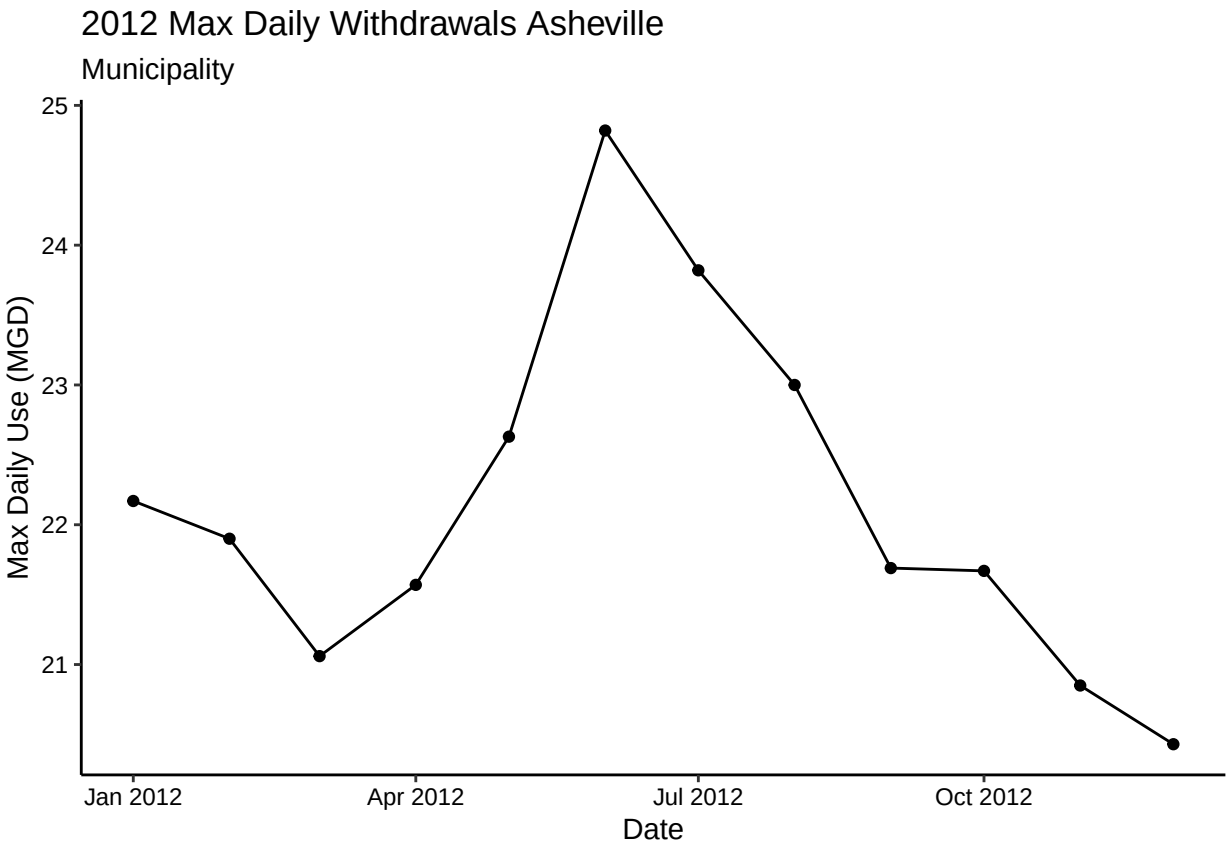
```
#9
#Set the inputs to scrape years
pwsid = "01-11-010"

#Subset the facilities
the_year<- rep(2010:2021)

pwsid <-  rep.int("01-11-010", length(the_year))
print(pwsid)
```

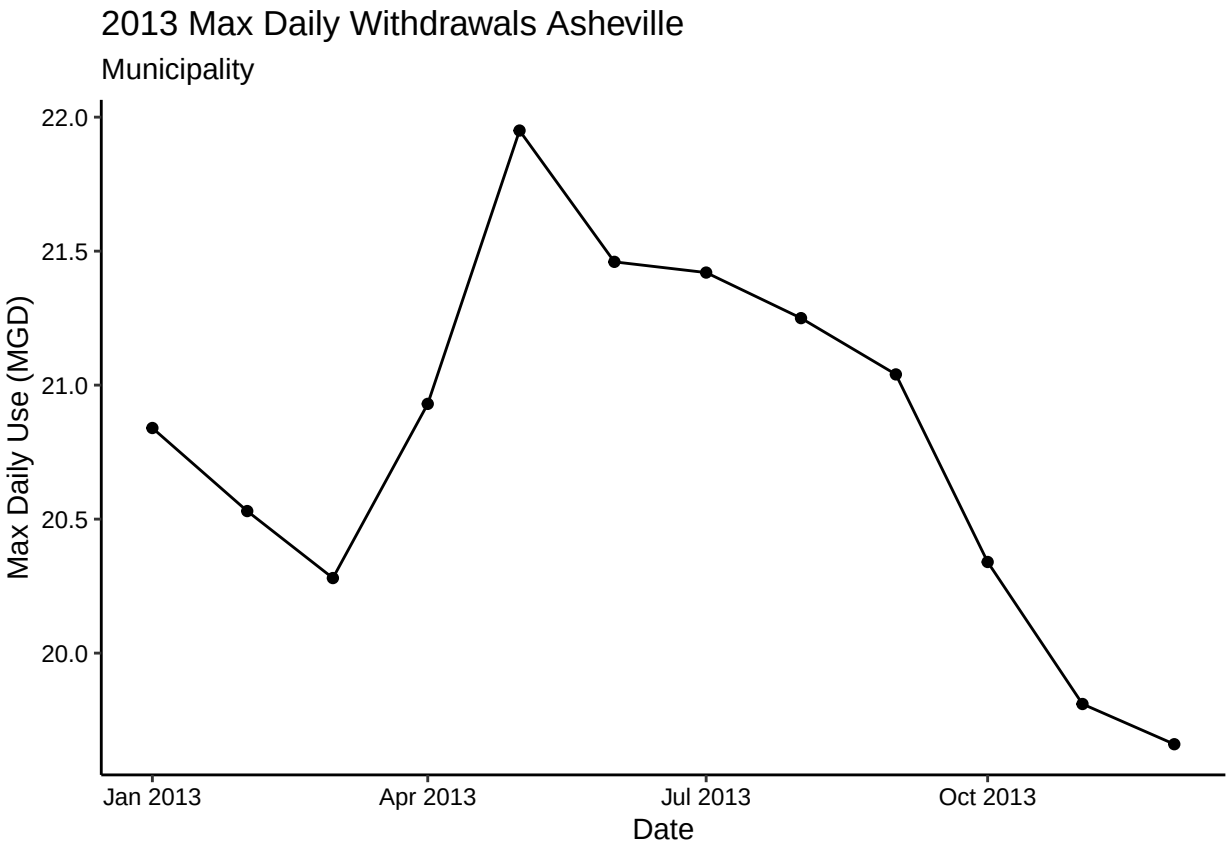
```
##  [1] "01-11-010" "01-11-010" "01-11-010" "01-11-010" "01-11-010" "01-11-010"
##  [7] "01-11-010" "01-11-010" "01-11-010" "01-11-010" "01-11-010" "01-11-010"
```

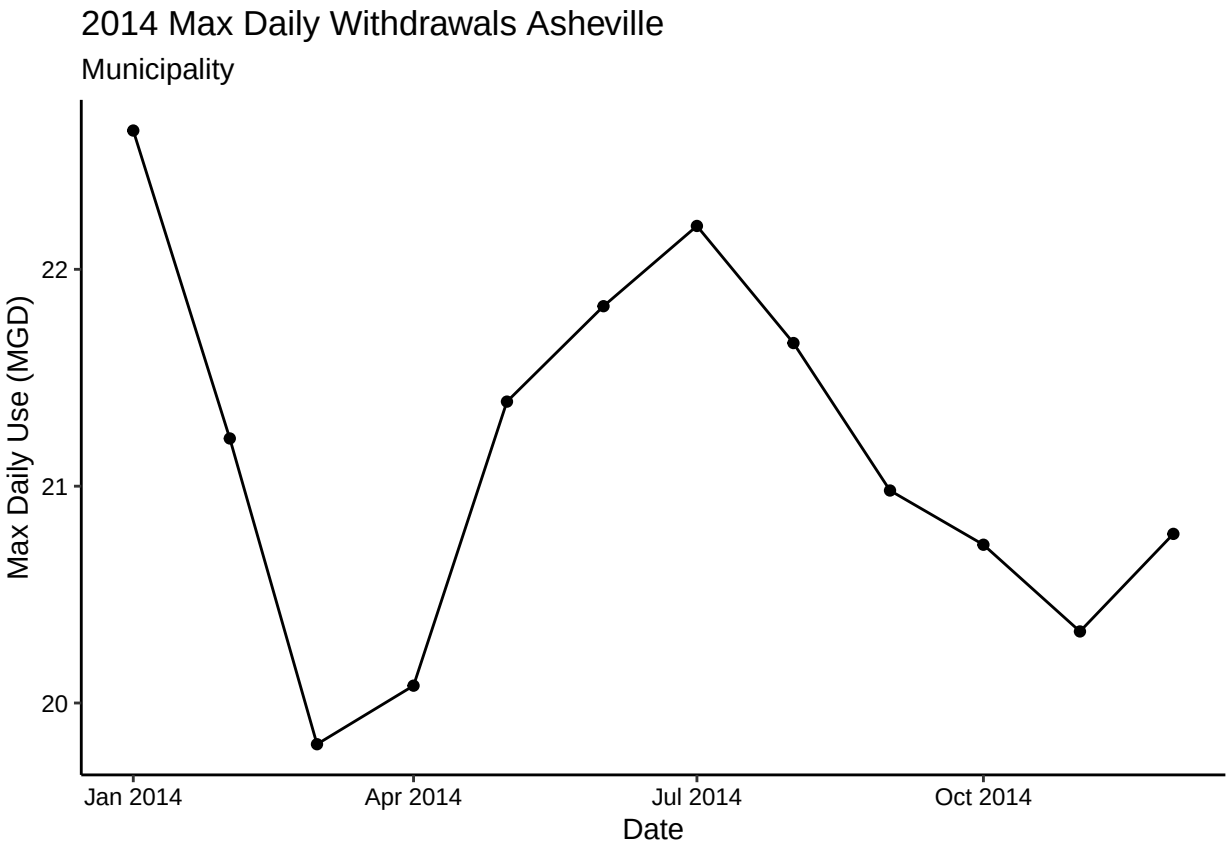
```
#"Map" the "scrape.it" function to retrieve data for all these
dfs_2020 <- map2(pwsid, the_year, scrape.it)
```

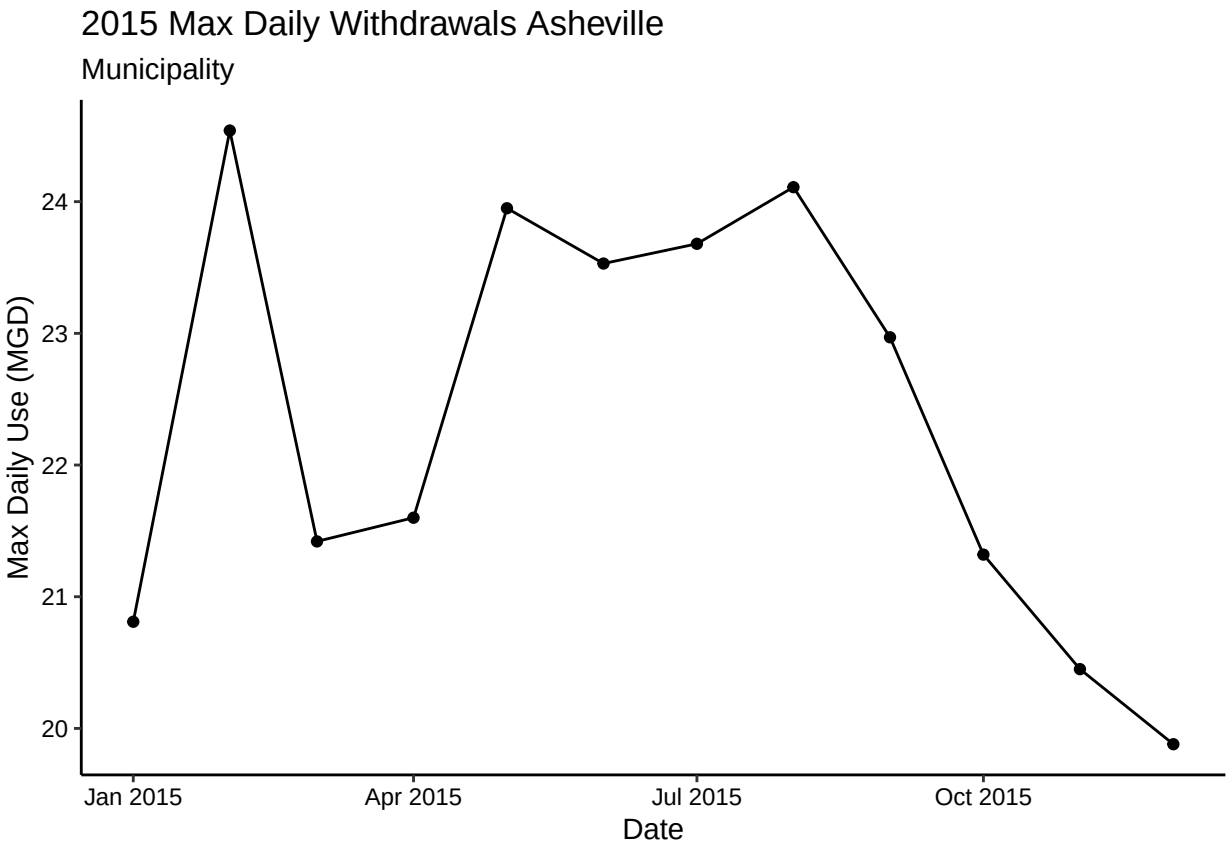


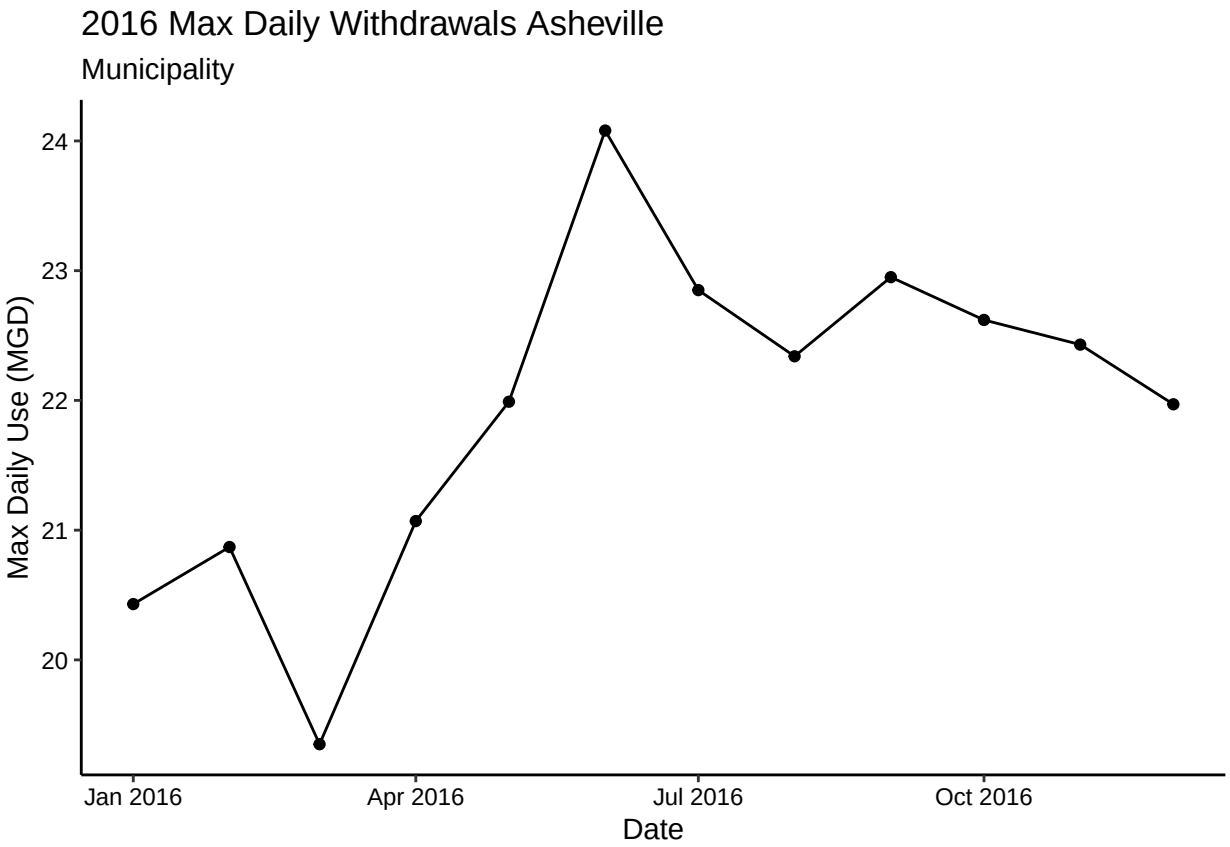

2011 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
24
23
22
21
Jan 2011
Apr 2011
Jul 2011
Oct 2011
Date

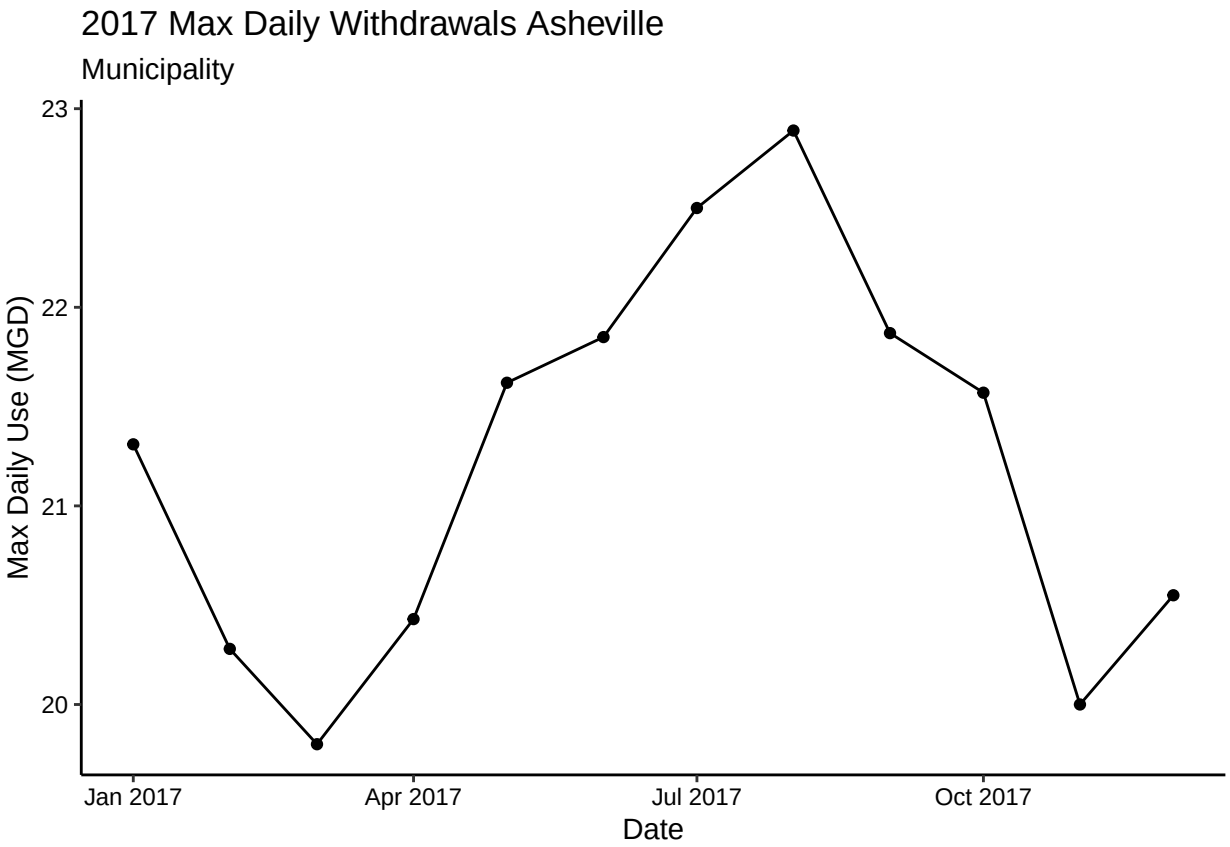

2012 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
25
24
23
22
21
Jan 2012
Apr 2012
Jul 2012
Oct 2012
Date

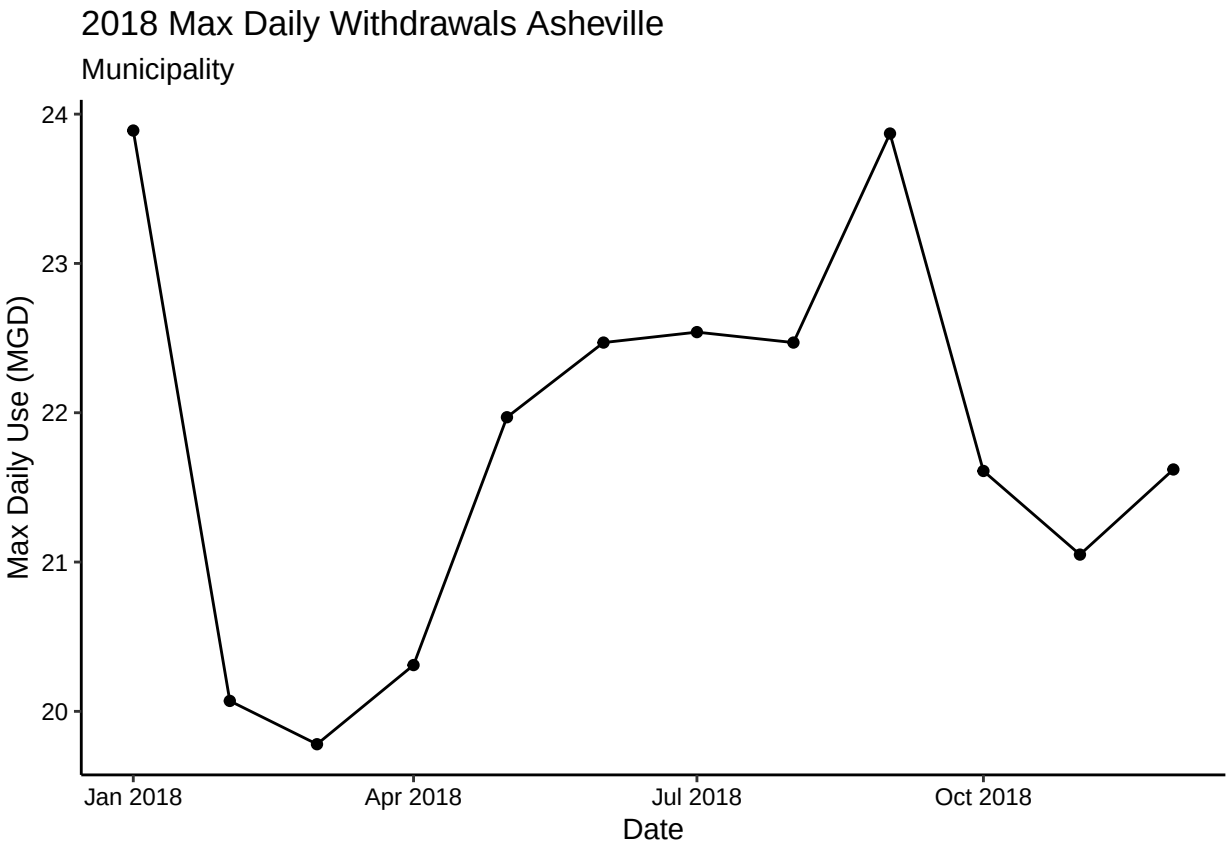

2013 Max Daily Withdrawals Asheville
Municipality
22.0
21.5
21.0
20.5
20.0
Max Daily Use (MGD)
Jan 2013
Apr 2013
Jul 2013
Oct 2013
Date

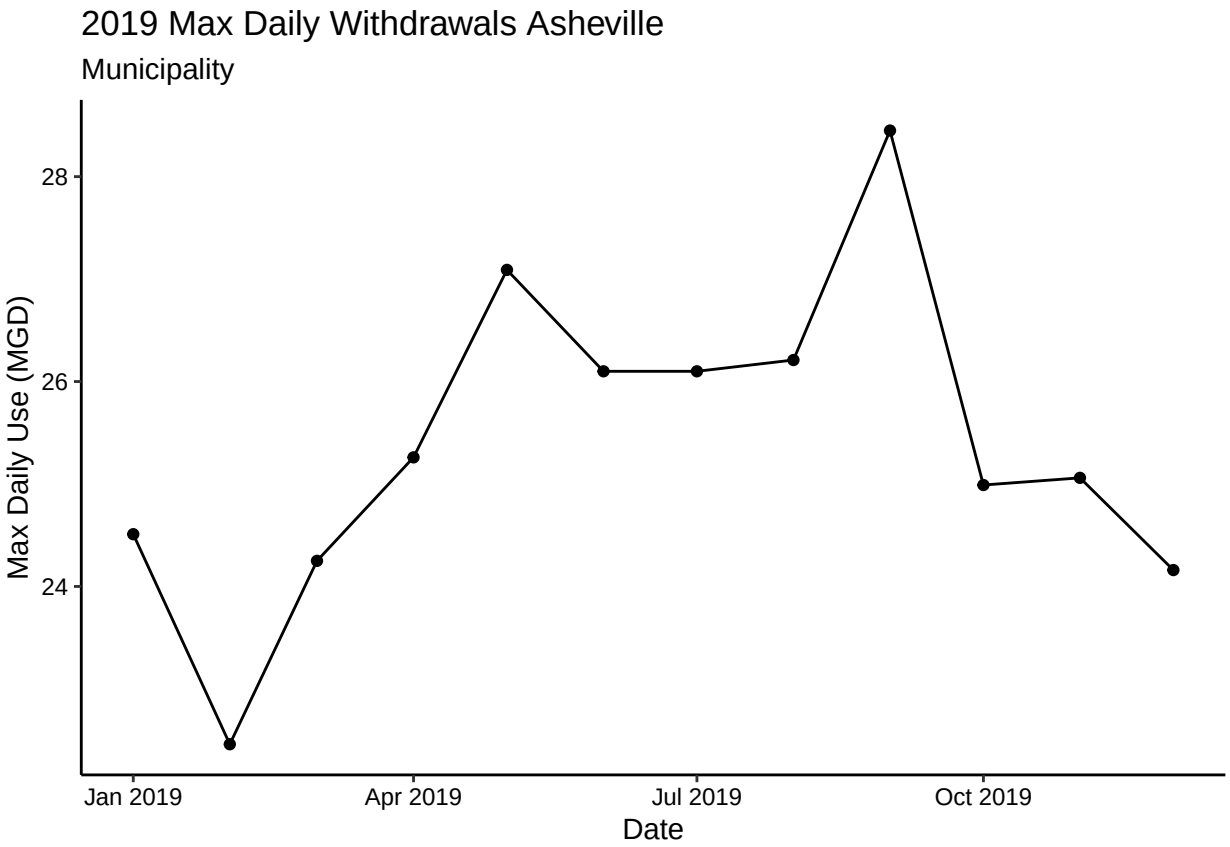

2014 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
22
21
20
Jan 2014
Apr 2014
Jul 2014
Oct 2014
Date

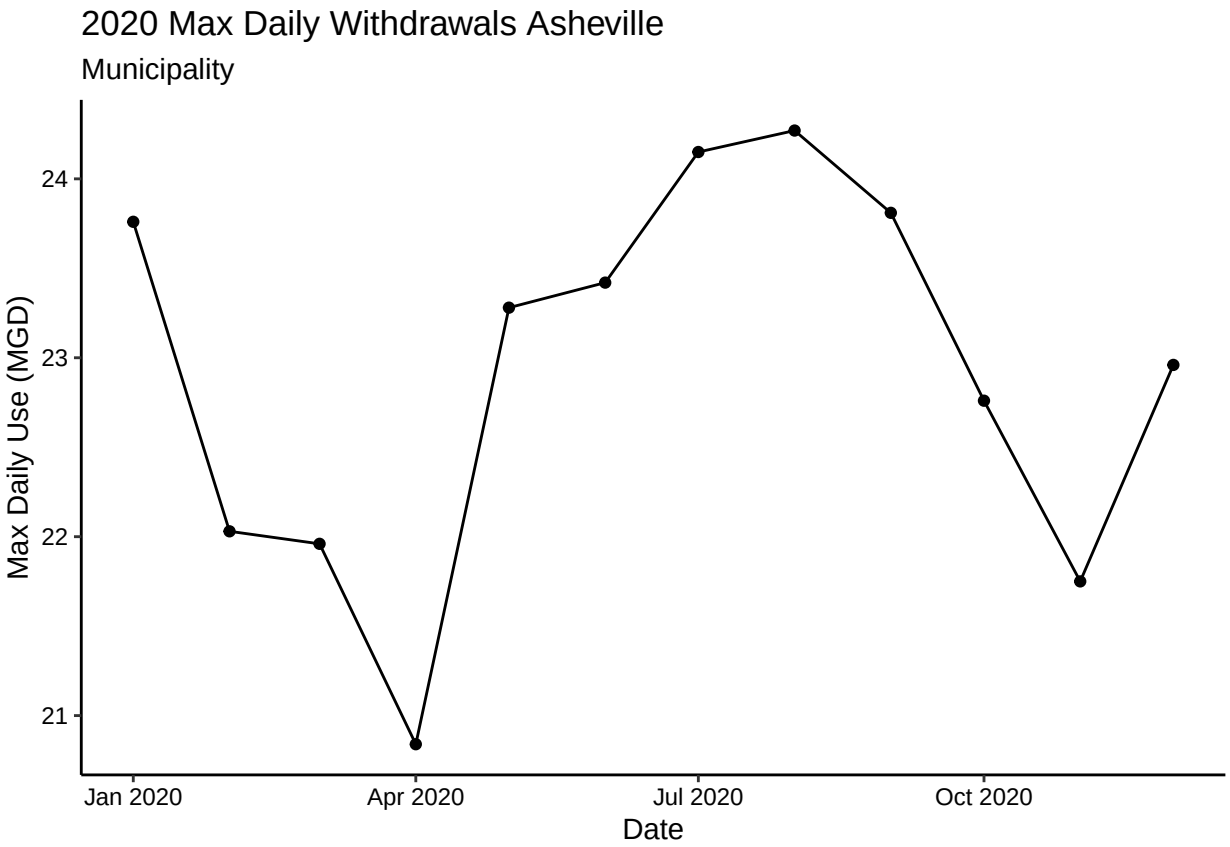

2015 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
24
23
22
21
20
Jan 2015
Apr 2015
Jul 2015
Oct 2015
Date

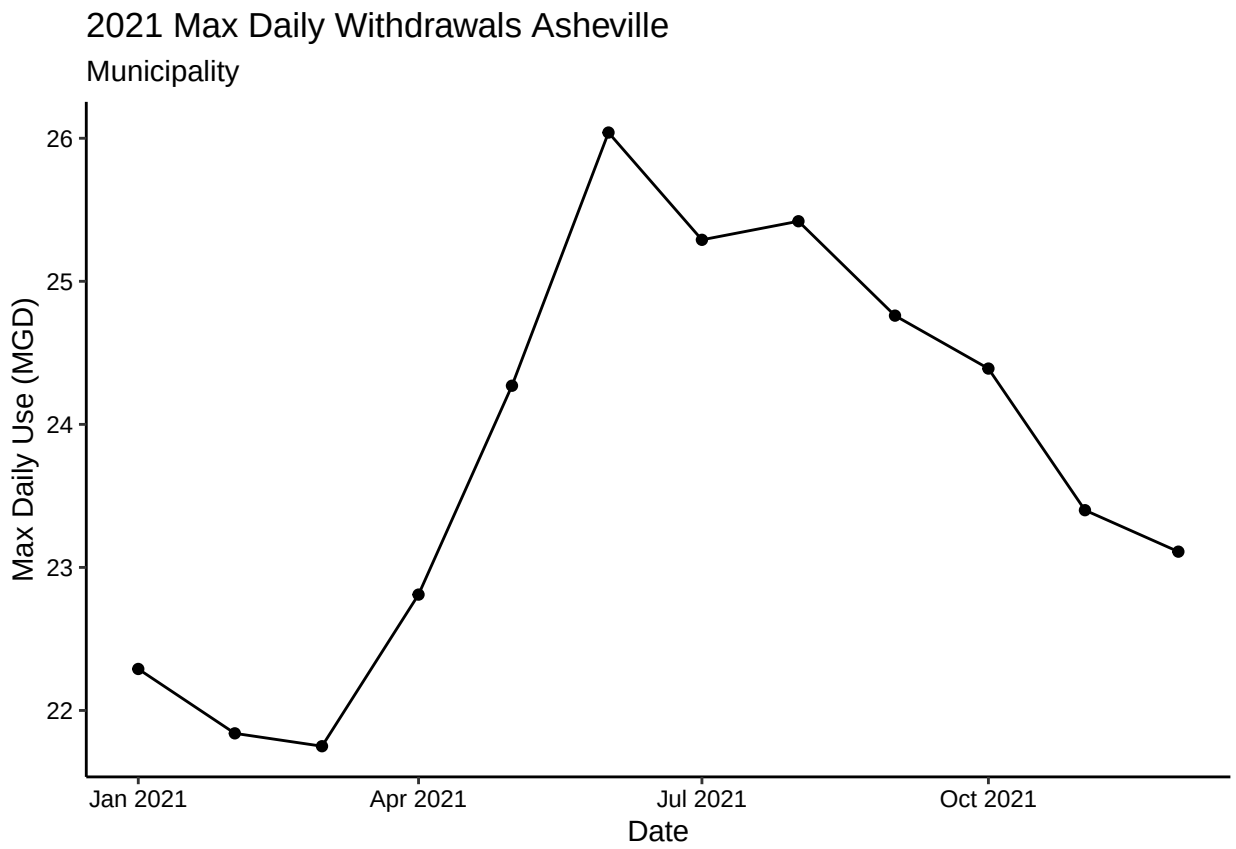

2016 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
24
23
22
21
20
Jan 2016
Apr 2016
Jul 2016
Oct 2016
Date


2017 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
23
22
21
20
Jan 2017
Apr 2017
Jul 2017
Oct 2017
Date


2018 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
24
23
22
21
20
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Date

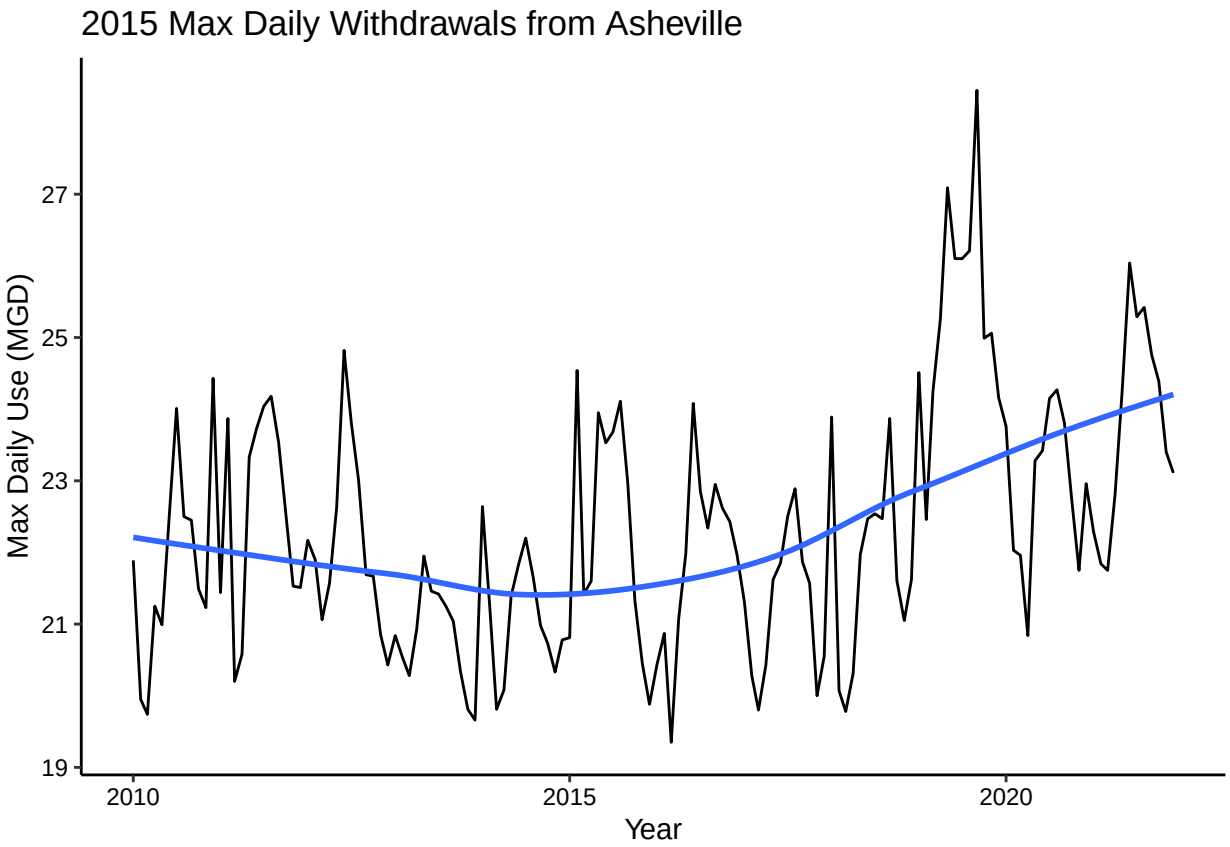

2019 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
28
26
24
Jan 2019
Apr 2019
Jul 2019
Oct 2019
Date


2020 Max Daily Withdrawals Asheville
Municipality
Max Daily Use (MGD)
24
23
22
21
Jan 2020
Apr 2020
Jul 2020
Oct 2020
Date




```
#Conflate the returned list of dataframes into a single one
df_2020 <- bind_rows(dfs_2020)


#Plot
ggplot(df_2020,aes(y = Vector, x=Date)) +
  geom_path(group = 1)+
    geom_smooth(method="loess",se=FALSE) +
  labs(title = paste(Asheville$Year, "Max Daily Withdrawals from",
  Asheville$Water_System_Name),
       y="Max Daily Use (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: It looks like it is increasing over time, though we would want to run statistics to get a better idea.