Assignment 10: Data Scraping

Kayla Emerson

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
#load packages
library(tidyverse)
library(rvest)
library(here)
library(dataRetrieval)
library(lubridate)

#check working directory
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2023 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=2023

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

```
#2
#Set the URL
theURL <-
  read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
```

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
#collect data by scraping website and create objects

waterSystemName <- theURL  %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

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

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

MGD <- theURL  %>%
  html_nodes("th~ td+ td") %>%
  html_text()
```

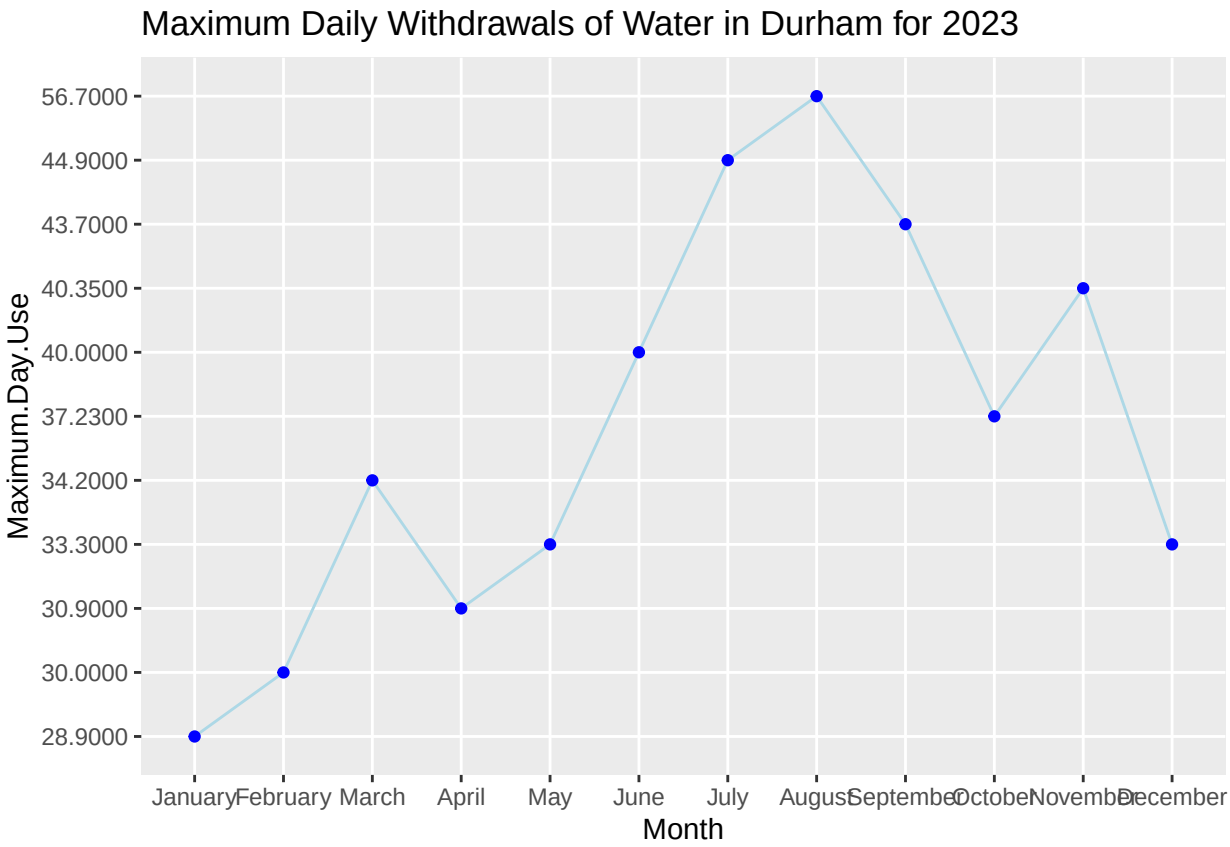
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

> 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 2023, making sure, the months are presented in proper sequence.

```
#4
#create dataframe from scraped data

df_water <- data.frame("Water System Name" = waterSystemName,
                       "PWSID" = PWSID,
                       "Ownership" = Ownership,
                       "Maximum Day Use" = as.numeric(gsub(".", "",
                                                           MGD)),
                       "Month" = c("January", "February", "March", "April", "May",
                                   "June", "July", "August", "September",
                                   "October", "November", "December"))

df_water$Month <- factor(df_water$Month, levels = c("January", "February", "March", "April", "May",
                                   "June", "July", "August", "September",
                                   "October", "November", "December"))


#manually enter MGD values because they are out of order
df_water$Maximum.Day.Use[1] <- MGD[1]
df_water$Maximum.Day.Use[2] <- MGD[4]
df_water$Maximum.Day.Use[3] <- MGD[7]
df_water$Maximum.Day.Use[4] <- MGD[10]
df_water$Maximum.Day.Use[5] <- MGD[2]
df_water$Maximum.Day.Use[6] <- MGD[5]
df_water$Maximum.Day.Use[7] <- MGD[8]
df_water$Maximum.Day.Use[8] <- MGD[11]
df_water$Maximum.Day.Use[9] <- MGD[3]
df_water$Maximum.Day.Use[10] <- MGD[6]
df_water$Maximum.Day.Use[11] <- MGD[9]
df_water$Maximum.Day.Use[12] <- MGD[12]


#5
#Create a line plot
plot_5 <- df_water %>%
  ggplot(aes(x = Month, y = Maximum.Day.Use)) +
  geom_line(group = 1, color = "lightblue") +
  geom_point(color = "blue") +
  labs(title = "Maximum Daily Withdrawals of Water in Durham for 2023")

print(plot_5)
```



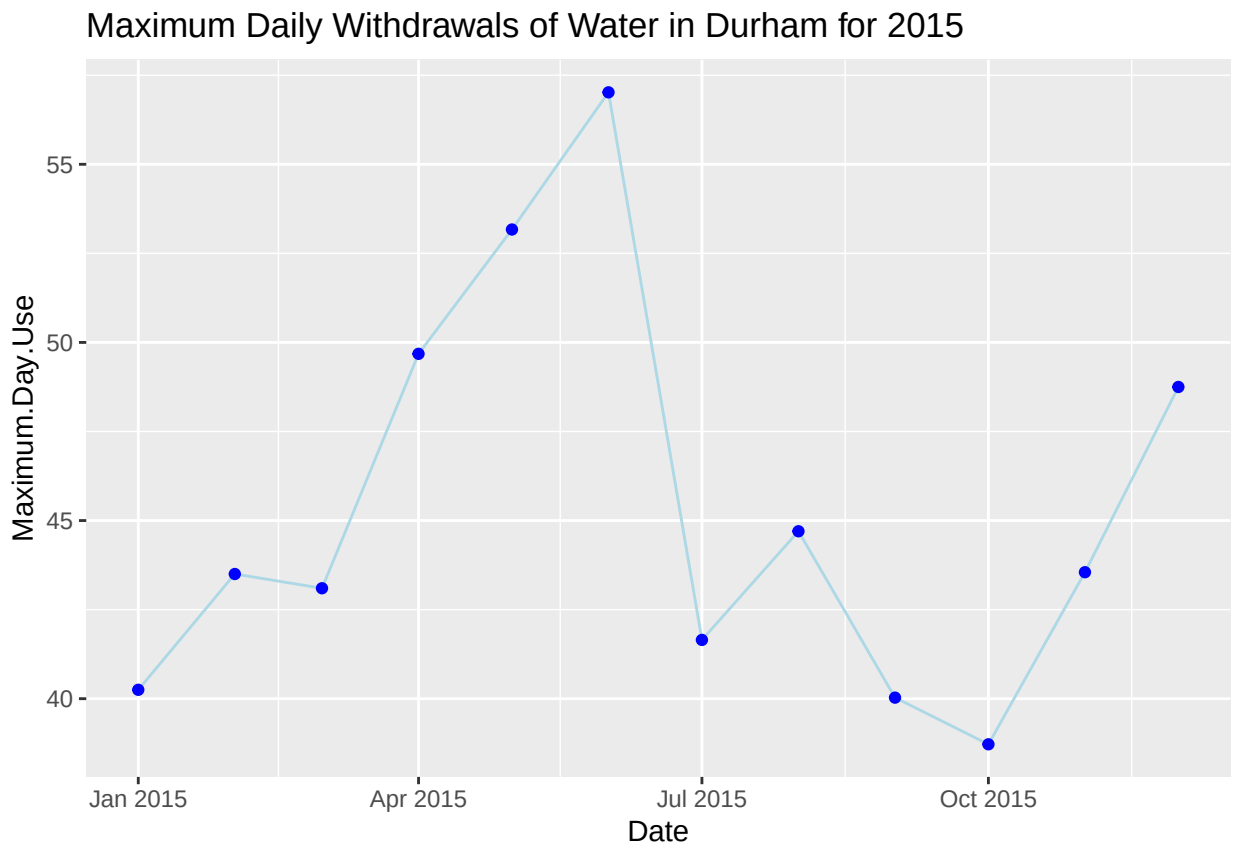
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.
#create function
#define function
scrape.it <- function(any_pwsid, any_year){
  #Get the proper url
  the_url <- paste(
    "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=", any_pwsid,
    "&year=", any_year, sep = "")

  #Fetch the website
  the_website <- read_html(the_url)

  #set knowns
  PWSID <- any_pwsid
  year <- any_year

  #Scrape the data
  waterSystemName <- the_website  %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text(trim = TRUE)
```

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

  MGD <- the_website  %>%
  html_nodes("th~ td+ td") %>%
  html_text(trim = TRUE)

  #convert MGD to numeric
  MGD <- as.numeric(MGD)

  #manual month column
  months <- c("January", "May", "September", "February", "June", "October", "March",
              "July", "November", "April", "August", "December")

  #manual date column to get everything in chronological order
  dates <- as.Date(paste(year, months, "1"), format = "%Y %b %d")

  #Convert to dataframe
  df_target <- data.frame(
    "Date" = dates,
    "Year" = rep(year, length(months)),
    "Month" = months,
    "Water System Name" = waterSystemName,
    "PWSID" = PWSID,
    "Ownership" = Ownership,
    "Maximum Day Use" = MGD)


  #Return the dataframe
  return(df_target)
}
```

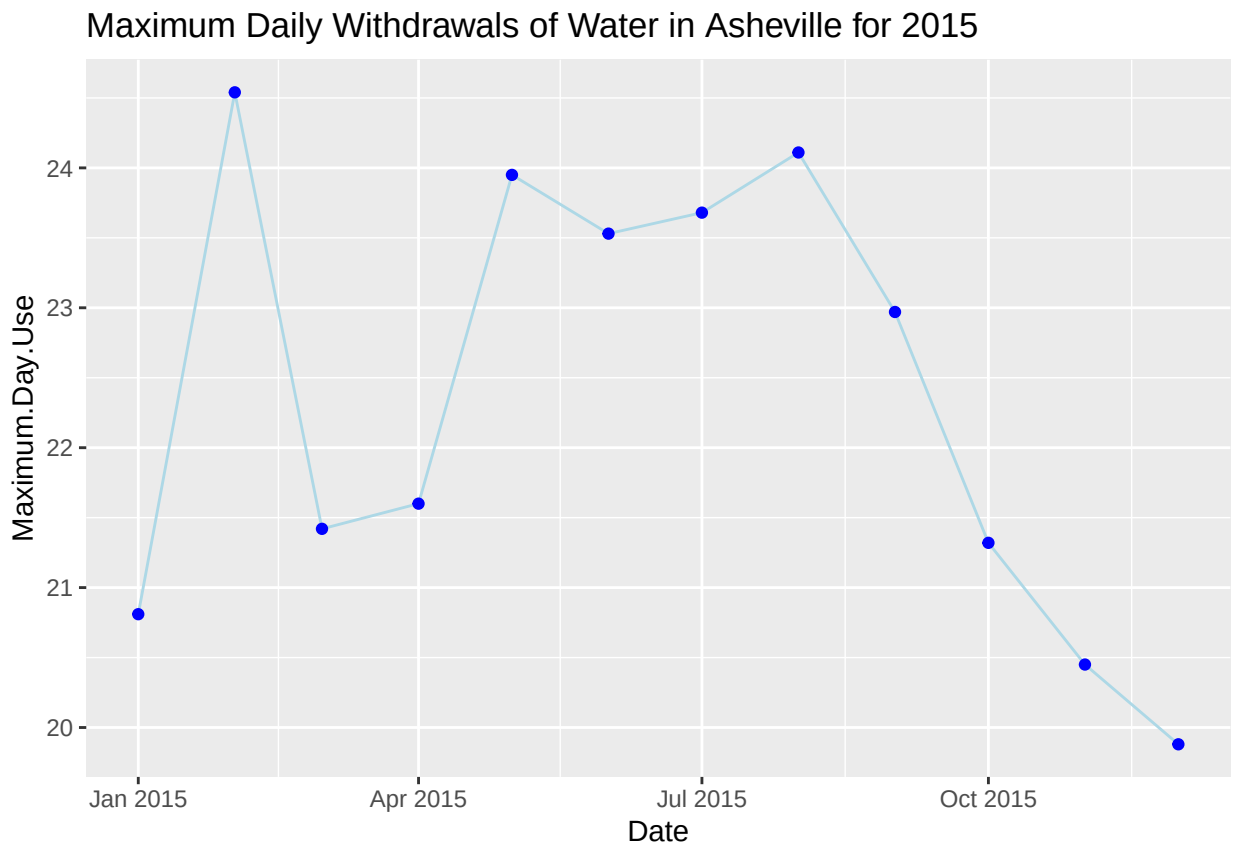
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7
#run the function
test_2015_durm <- scrape.it("03-32-010", 2015)

#plot the results
plot_7 <- test_2015_durm %>%
  ggplot(aes(x = Date, y = Maximum.Day.Use)) +
  geom_line(group = 1, color = "lightblue") +
  geom_point(color = "blue") +
  labs(title = "Maximum Daily Withdrawals of Water in Durham for 2015")

print(plot_7)
```

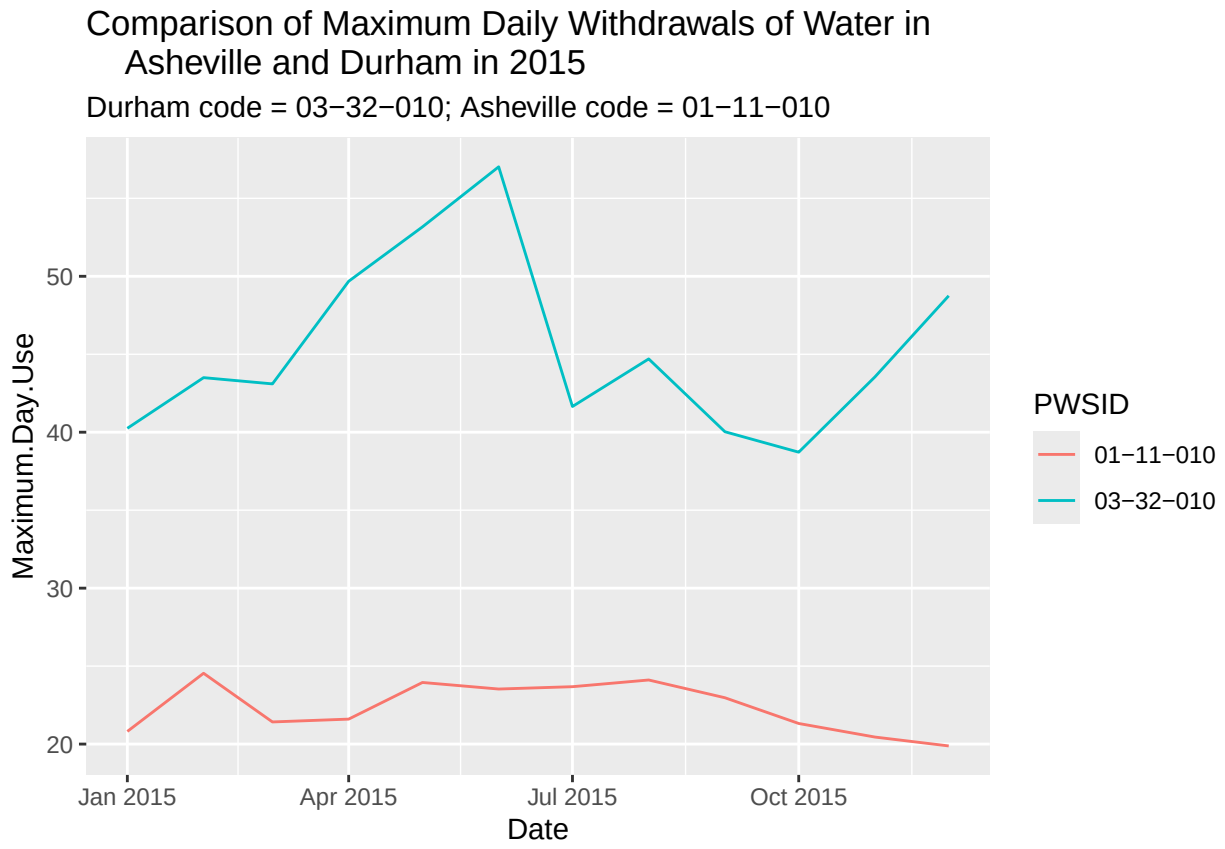


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8
#run function for Asheville in 2015
test_2015_ash <- scrape.it("01-11-010", 2015)

#plot the results
plot_8 <- test_2015_ash %>%
  ggplot(aes(x = Date, y = Maximum.Day.Use)) +
  geom_line(group = 1, color = "lightblue") +
  geom_point(color = "blue") +
  labs(title = "Maximum Daily Withdrawals of Water in Asheville for 2015")

print(plot_8)
```




```
#combined data frames
ash_and_durm_2015 <- rbind(test_2015_durm, test_2015_ash)

#plot both together
plot_8b <- ash_and_durm_2015 %>%
  ggplot(aes(x = Date, y = Maximum.Day.Use, color = PWSID)) +
  geom_line() +
  labs(
    title = "Comparison of Maximum Daily Withdrawals of Water in
    Asheville and Durham in 2015",
    subtitle = "Durham code = 03-32-010; Asheville code = 01-11-010"
  )

print(plot_8b)
```

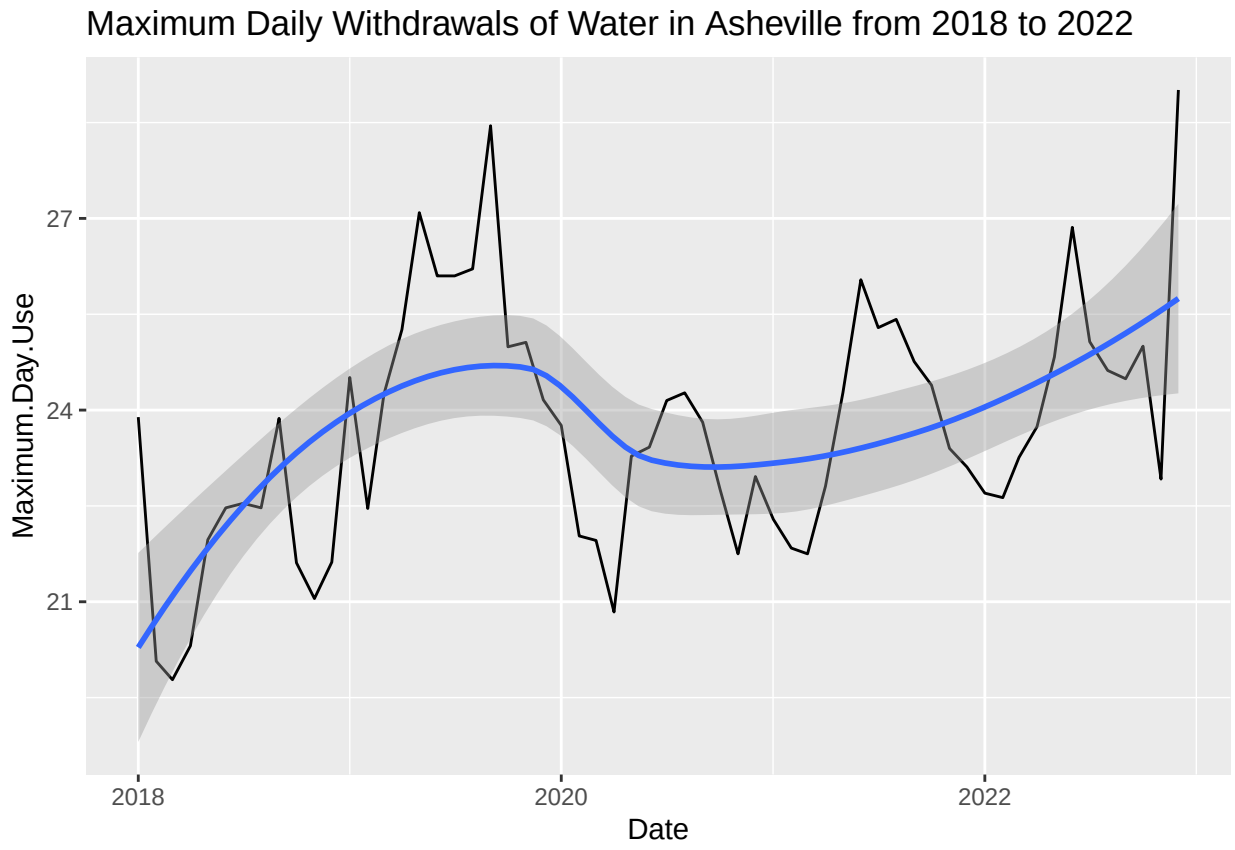


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022.Add a smoothed line to the plot (method = 'loess').

   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
#make years a vector
years_2018to2022 <- 2018:2022

#make Asheville ID vector of equal length
ashe_PWSID_repeated  <- rep("01-11-010", length(years_2018to2022))

#use map2 with 2 vectors of equal length to retrieve all data
ashe_2018to2022 <- map2(ashe_PWSID_repeated, years_2018to2022,
                        scrape.it) %>% bind_rows()

#plot
plot_9 <- ashe_2018to2022 %>%
  ggplot(aes(x = Date, y = Maximum.Day.Use)) +
  geom_line() +
  geom_smooth(method = "loess", se = TRUE) +
  labs(title = "Maximum Daily Withdrawals of Water in Asheville from 2018 to 2022")

print(plot_9)
```

```
## `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: > I think there is a somewhat of an upwards trend in water usuage over time, but it's not a strong trend. Usuage increased from 2018 to 2020, but then fell a little until 2021 where it thne started to increase again.