Assignment 10: Data Scraping

Leah Li

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 required packages
library(tidyverse)
library(rvest)

# Check working directory
getwd()
```

```
## [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.)

```r
#2

# Define the URL to be scraped
url <- "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023"

# Read the webpage
webpage <- read_html(url)

webpage
```

```
## {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)“.

```r
#3

water_system_name <- webpage %>%
  html_node("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

water_system_name
```

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

```r
pwsid <- webpage %>%
  html_node("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

pwsid
```

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

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

ownership
```

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

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

max_day_use
```

```
##  [1] "28.9000" "33.3000" "43.7000" "30.0000" "40.0000" "37.2300" "34.2000"
##  [8] "44.9000" "40.3500" "30.9000" "56.7000" "33.3000"
```

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…

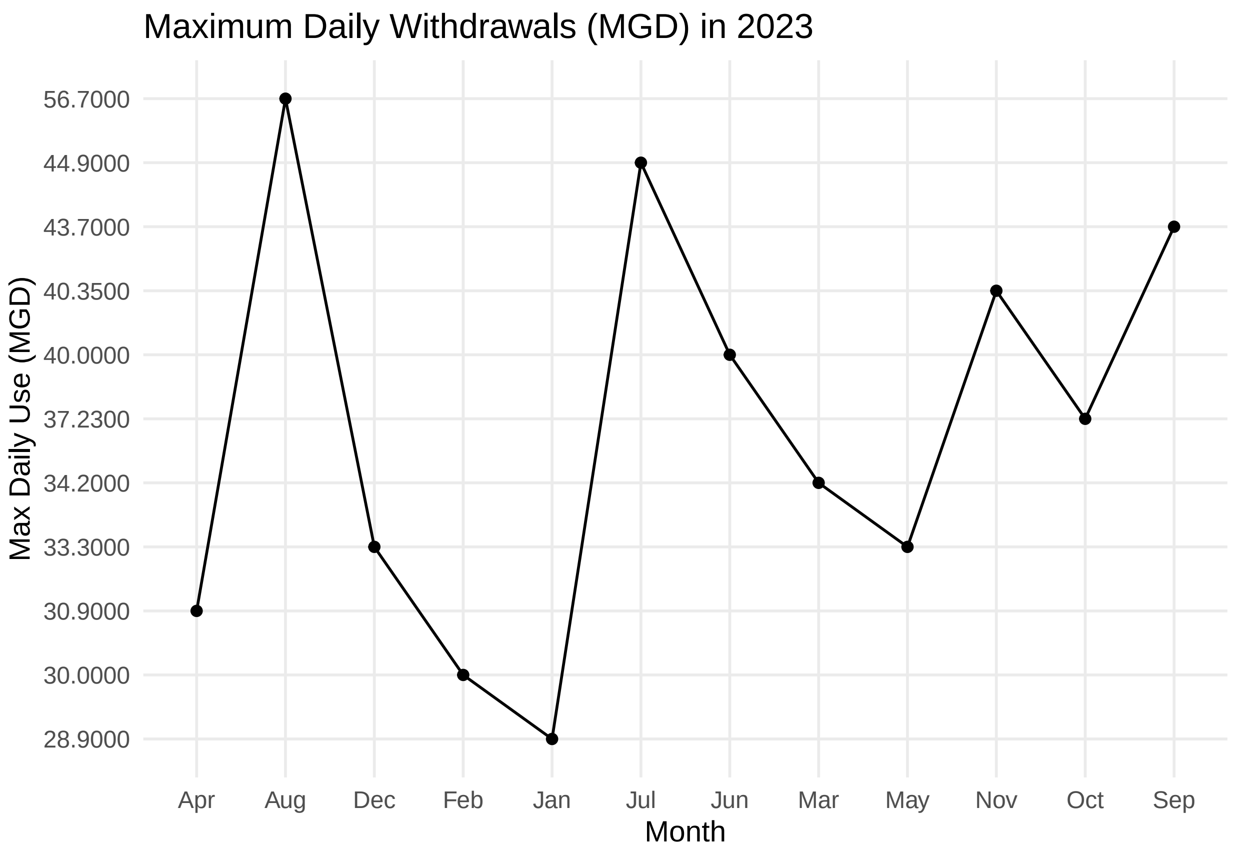
5. Create a line plot of the maximum daily withdrawals across the months for 2023, making sure, the months are presented in proper sequence.

```
#4

# Define the months manually in the order the data appears
months <- c("Jan", "May", "Sep", "Feb", "Jun", "Oct",
            "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

# Define the year manually (2023)
year <- 2023

# Create a dataframe
data <- tibble(
  Month = months,
  Year = rep(year, length(months)),
  Date = as.Date(paste(year, months, "01", sep = "-"), format = "%Y-%b-%d"),
  Water_System_Name = rep(water_system_name, length(months)),
  PWSID = rep(pwsid, length(months)),
  Ownership = rep(ownership, length(months)),
  Max_Day_Use_MGD = max_day_use
)
```

```r
# Sort the dataframe by the Date column
data <- data %>%
  arrange(Date)

# Display the dataframe
data
```

```
## # A tibble: 12 x 7
##    Month  Year Date       Water_System_Name PWSID     Ownership  Max_Day_Use_MGD
##    <chr> <dbl> <date>     <chr>             <chr>     <chr>      <chr>
##  1 Jan    2023 2023-01-01 Durham            03-32-010 Municipal~ 28.9000
##  2 Feb    2023 2023-02-01 Durham            03-32-010 Municipal~ 30.0000
##  3 Mar    2023 2023-03-01 Durham            03-32-010 Municipal~ 34.2000
##  4 Apr    2023 2023-04-01 Durham            03-32-010 Municipal~ 30.9000
##  5 May    2023 2023-05-01 Durham            03-32-010 Municipal~ 33.3000
##  6 Jun    2023 2023-06-01 Durham            03-32-010 Municipal~ 40.0000
##  7 Jul    2023 2023-07-01 Durham            03-32-010 Municipal~ 44.9000
##  8 Aug    2023 2023-08-01 Durham            03-32-010 Municipal~ 56.7000
##  9 Sep    2023 2023-09-01 Durham            03-32-010 Municipal~ 43.7000
## 10 Oct    2023 2023-10-01 Durham            03-32-010 Municipal~ 37.2300
## 11 Nov    2023 2023-11-01 Durham            03-32-010 Municipal~ 40.3500
## 12 Dec    2023 2023-12-01 Durham            03-32-010 Municipal~ 33.3000
```

```r
#5

# Load ggplot2 for plotting
library(ggplot2)

# Create the line plot
ggplot(data, aes(x = Month, y = Max_Day_Use_MGD, group = 1)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Maximum Daily Withdrawals (MGD) in 2023",
    x = "Month",
    y = "Max Daily Use (MGD)"
  ) +
  theme_minimal()
```

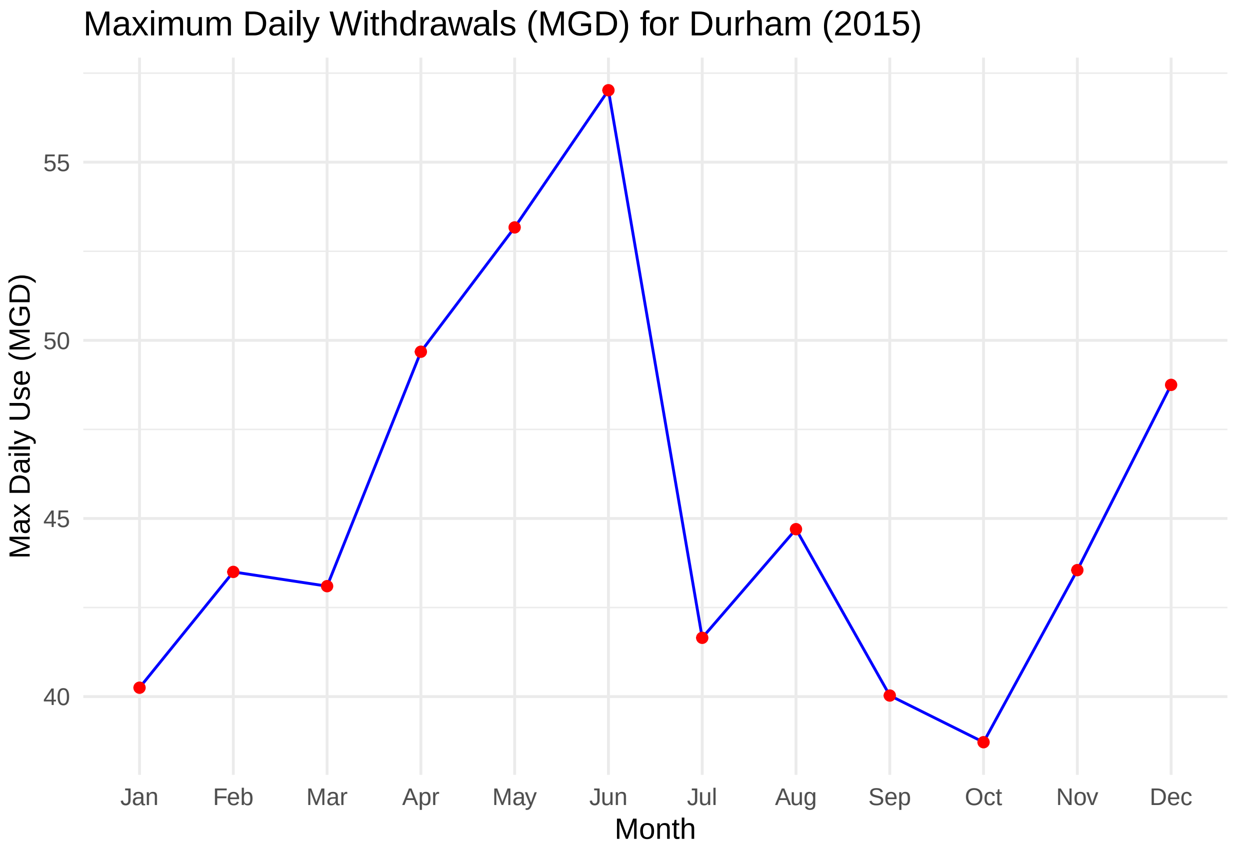


6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.

library(rvest)
library(dplyr)

# Define the scraping function
scrape.it <- function(pwsid, year) {

  # Read the webpage
  the_website <- read_html(paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=", pwsid, "&ye

  # Set the element address variables
  water_system_name2 <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  pwsid2 <- 'div+ table tr:nth-child(1) td:nth-child(5)'
  ownership2 <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  max_day_use2 <- 'th~ td+ td'

  # Scrape the data items
  Water_System_Name <- the_website %>% html_node(water_system_name2) %>% html_text(trim = TRUE)
  PWSID <- the_website %>% html_node(pwsid2) %>% html_text(trim = TRUE)
  Ownership <- the_website %>% html_node(ownership2) %>% html_text(trim = TRUE)
  Max_Day_Use_MGD <- the_website %>% html_nodes(max_day_use2) %>% html_text(trim = TRUE)

  # Ensure Max_Day_Use_MGD is numeric
  Max_Day_Use_MGD <- as.numeric(Max_Day_Use_MGD)

  # Convert to a dataframe
  df_MGD <- data.frame(
    "Month" = c("Jan", "May", "Sep", "Feb", "Jun", "Oct",
                "Mar", "Jul", "Nov", "Apr", "Aug", "Dec"),
    "Year" = rep(year, 12),
    "Max_Day_Use" = Max_Day_Use_MGD
  )


  df_MGD <- df_MGD %>%
    mutate(
      Water_System_Name = Water_System_Name,
      PWSID = PWSID,
      Ownership = Ownership,
      Date = as.Date(paste(Year, Month, "01", sep = "-"), format = "%Y-%b-%d")
    )

  # Ensure the months are in proper order
  df_MGD <- df_MGD %>%
    mutate(Month = factor(Month, levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))) %>%
    arrange(Date)

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

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

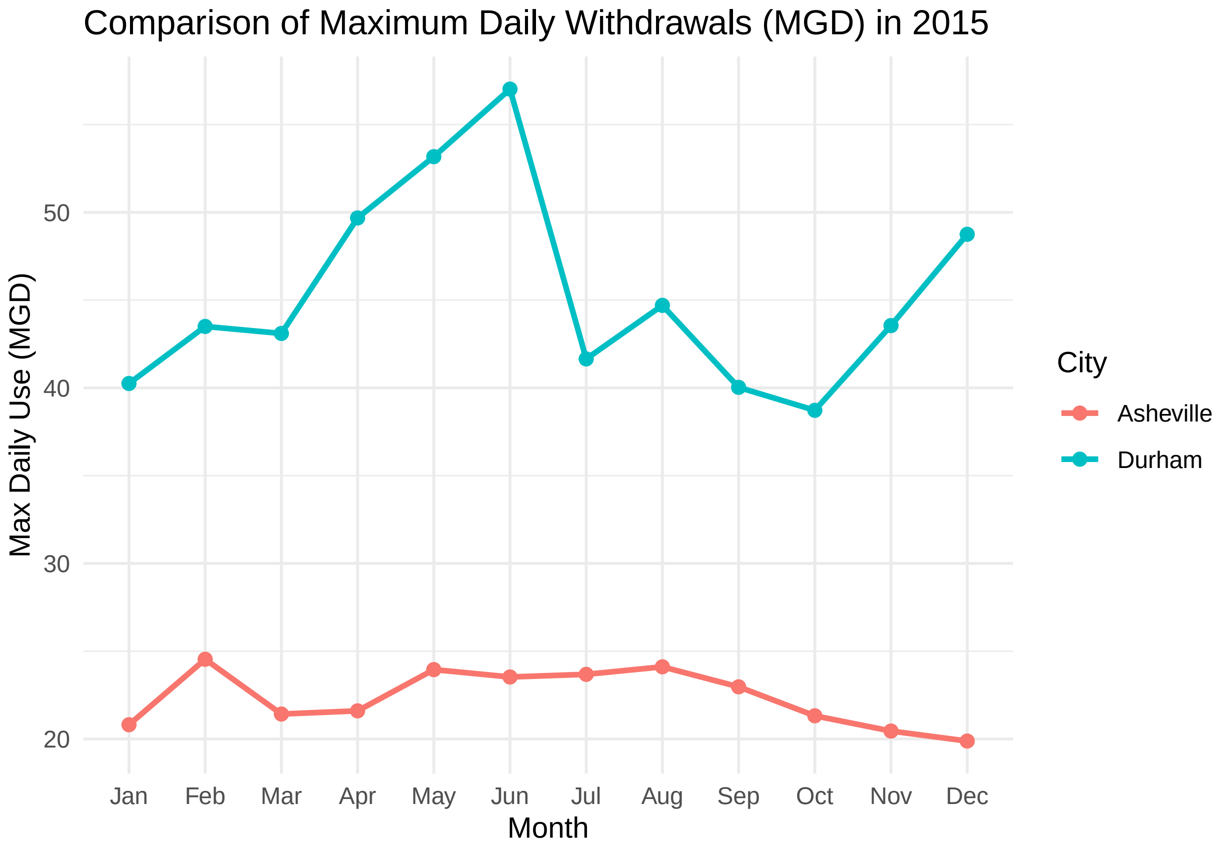
```
#7

# Define the PWSID and year for Durham
pwsid_durham <- "03-32-010"
year_2015 <- 2015

# Call the scrape.it function
df_Durham_2015 <- scrape.it(pwsid_durham, year_2015)

# Create the line plot
Plot2 <- ggplot(df_Durham_2015, aes(x = Month, y = Max_Day_Use, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(
    title = "Maximum Daily Withdrawals (MGD) for Durham (2015)",
    x = "Month",
    y = "Max Daily Use (MGD)"
  ) +
  theme_minimal()

Plot2
```



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
# Extract data for Asheville in 2015
df_Asheville_2015 <- scrape.it("01-11-010", 2015)

# Add a city column to both dataframes
df_Asheville_2015 <- df_Asheville_2015 %>%
  mutate(City = "Asheville")

df_Durham_2015 <- df_Durham_2015 %>%
  mutate(City = "Durham")

# Combine the two dataframes
df_combined <- bind_rows(df_Asheville_2015, df_Durham_2015)

# Create the plot
ggplot(df_combined, aes(x = Month, y = Max_Day_Use, color = City, group = City)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Comparison of Maximum Daily Withdrawals (MGD) in 2015",
    x = "Month",
    y = "Max Daily Use (MGD)",
    color = "City"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

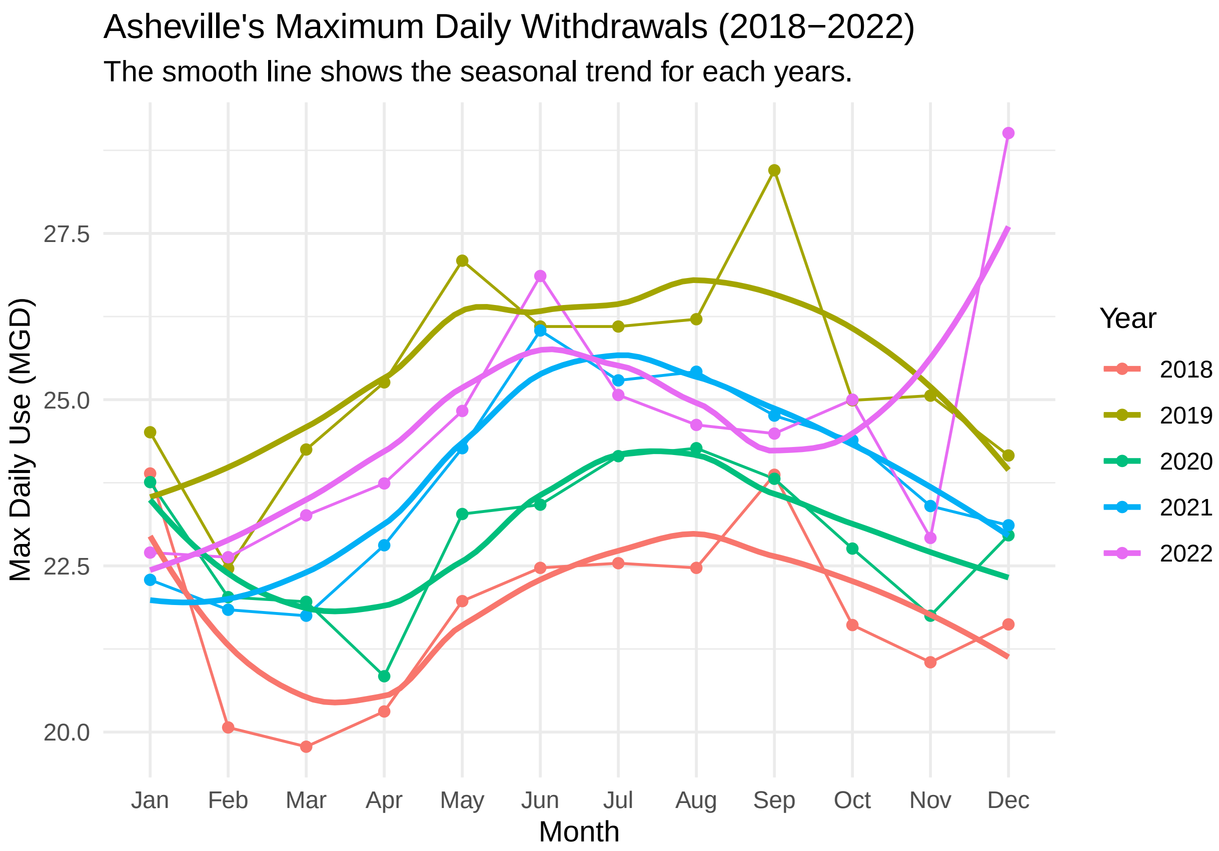


9. Use the code & function you created above to plot Asheville’s max daily withdrawal by months for the years 2018 thru 2022.Add a smoothed line to the plot (method = ‘loess’).

   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.

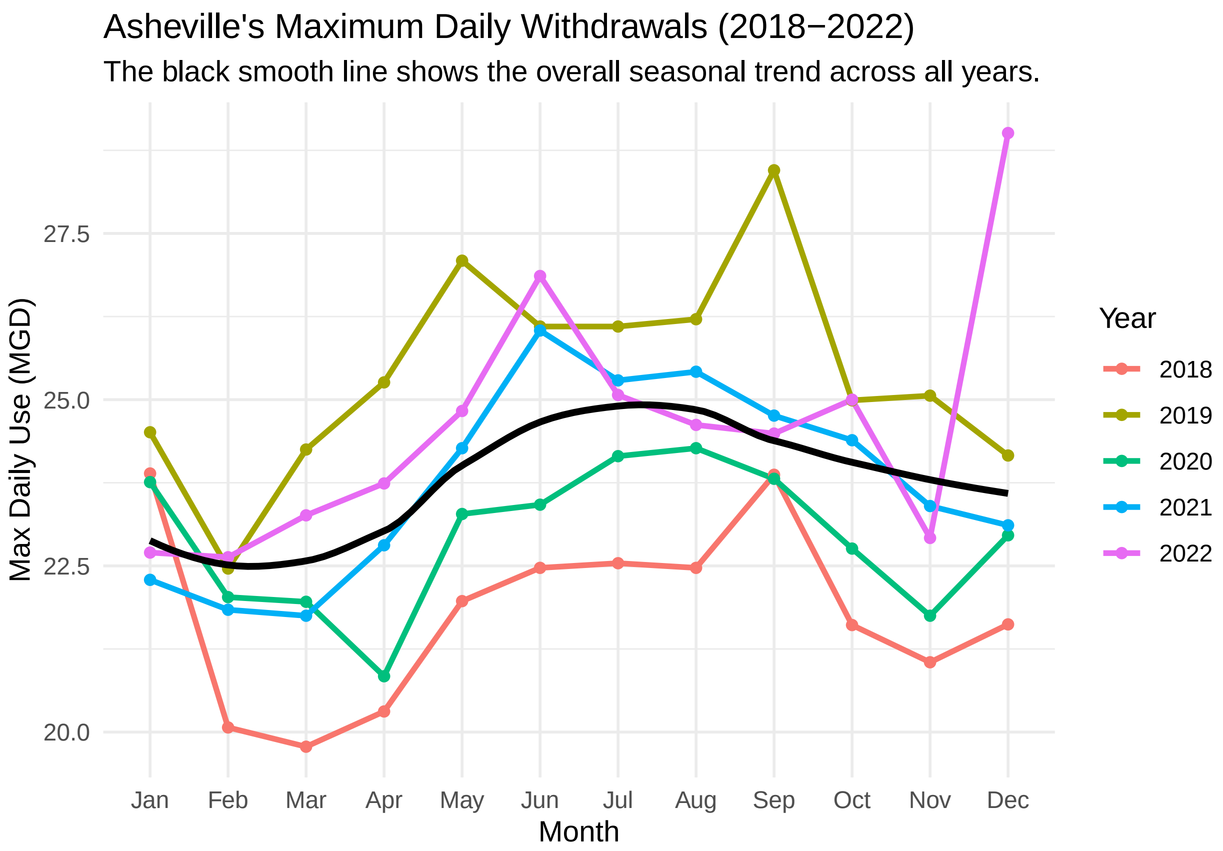
```r
#9
library(purrr)

# Define the PWSID for Asheville
asheville_pwsid <- "01-11-010"

# Define the years for iteration
the_years <- 2018:2022

# Create a vector of PWSIDs
the_pwsids <- rep(asheville_pwsid, length(the_years))

# "Map" the "scrape.it" function to retrieve data for all years
dfs_asheville <- map2(the_pwsids, the_years, scrape.it)

df_asheville <- bind_rows(dfs_asheville)

# Add a column for Year as a factor
df_asheville <- df_asheville %>%
  mutate(Year = factor(Year))

# Plot Asheville's data by year, with a smoothed line for each year
ggplot(df_asheville, aes(x = Month, y = Max_Day_Use, color = Year, group = Year)) +
  geom_line() +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Asheville's Maximum Daily Withdrawals (2018-2022)",
    subtitle = "The smooth line shows the seasonal trend for each years.",
    x = "Month",
    y = "Max Daily Use (MGD)",
    color = "Year"
  ) +
  theme_minimal()
```

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



```
# Plot Asheville's data by year, with one smoothed line for all year
ggplot(df_asheville, aes(x = Month, y = Max_Day_Use)) +
  geom_line(aes(color = Year, group = Year), size = 1) +
  geom_point(aes(color = Year)) +
  geom_smooth(aes(group = 1), method = "loess",
              se = FALSE, color = "black", size = 1.2) +
  labs(
    title = "Asheville's Maximum Daily Withdrawals (2018-2022)",
    subtitle = "The black smooth line shows the overall seasonal trend across all years.",
    x = "Month",
    y = "Max Daily Use (MGD)",
    color = "Year"
  ) +
  theme_minimal()
```

```
## `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:

> Based on the plots, Asheville's maximum daily water withdrawals show a strong seasonal pattern, with higher usage during the summer months (June-August) and lower usage during the winter months (January-February). The year-specific smooth lines in the first plot reveal some variability in water usage across years. For instance, 2019 and 2022 show distinct spikes in specific months like July and December, while 2018 exhibits a more gradual seasonal trend. These differences likely reflect year-specific factors such as weather variations or changes in demand rather than a consistent long-term trend. Overall, Asheville's water withdrawals appear to be relatively stable over the five-year period.
>
> The overall smooth line in the second plot highlights this consistent seasonal fluctuation across all years. However, there does not appear to be a clear upward or downward trend in water usage over time when looking at the combined data from 2018 to 2022.