# Assignment 10: Data Scraping

Rebecca Cox

## 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)
here()
```

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

```
library(rvest)
library(dplyr)
library(zoo)
```

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

webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023')
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)".

```
#3

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

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

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

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

months <- c(1,5,9,2,6,10,3,7,11,4,8,12) #in order from the site
```

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.)

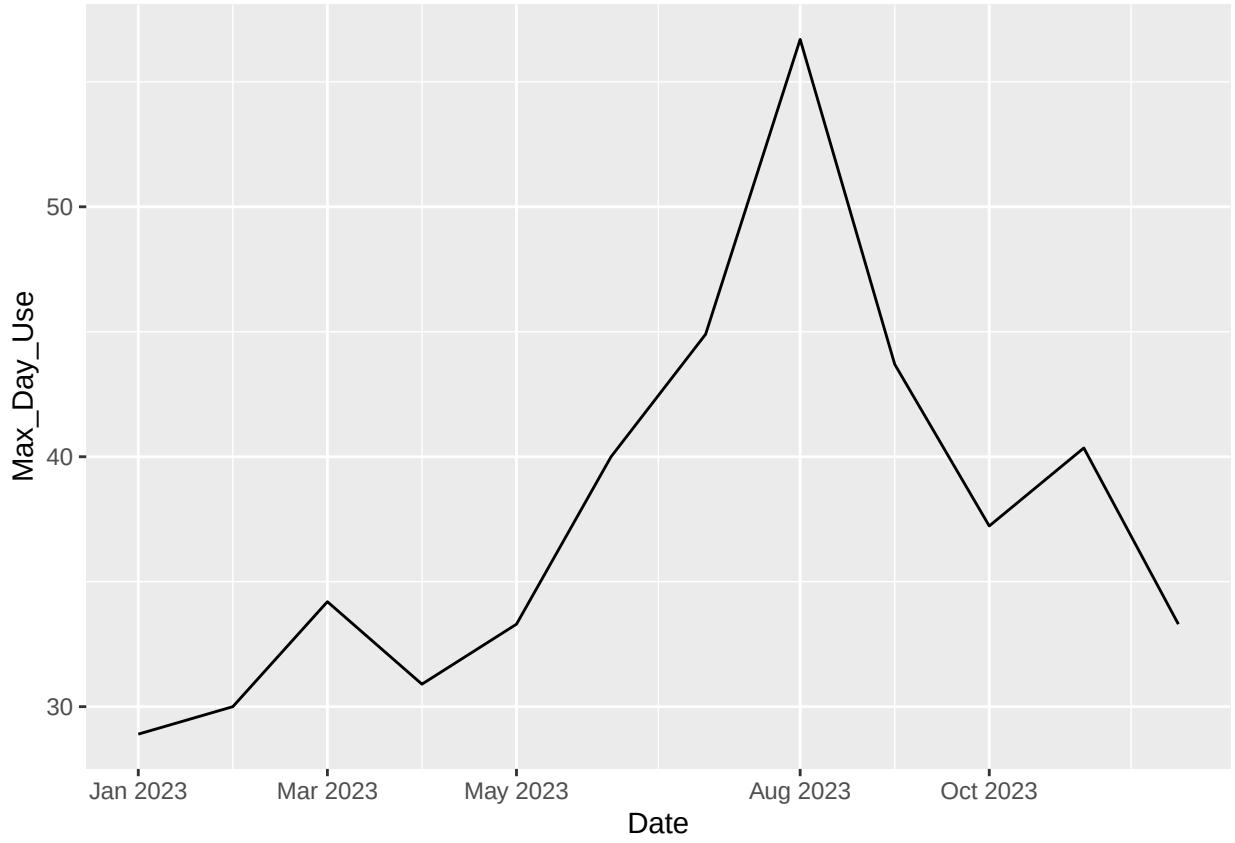
   TIP: Use `rep()` to repeat a value when creating a dataframe.

   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

dataframe <- data.frame(Month = months,
                        Date = as.yearmon(paste(month.abb[months],"2023")),
                        Water_System = water_system,
                        PWSID = pwsid,
                        Ownership = ownership,
                        Max_Day_Use = as.numeric(max_day_use_month))

dataframe <- arrange(dataframe, Month)

#5

ggplot(dataframe, aes(x=Date,y=Max_Day_Use)) +
  geom_line()
```

```
## Warning: The ‘trans‘ argument of ‘continuous_scale()‘ is deprecated as of ggplot2 3.5.0.
## i Please use the ‘transform‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



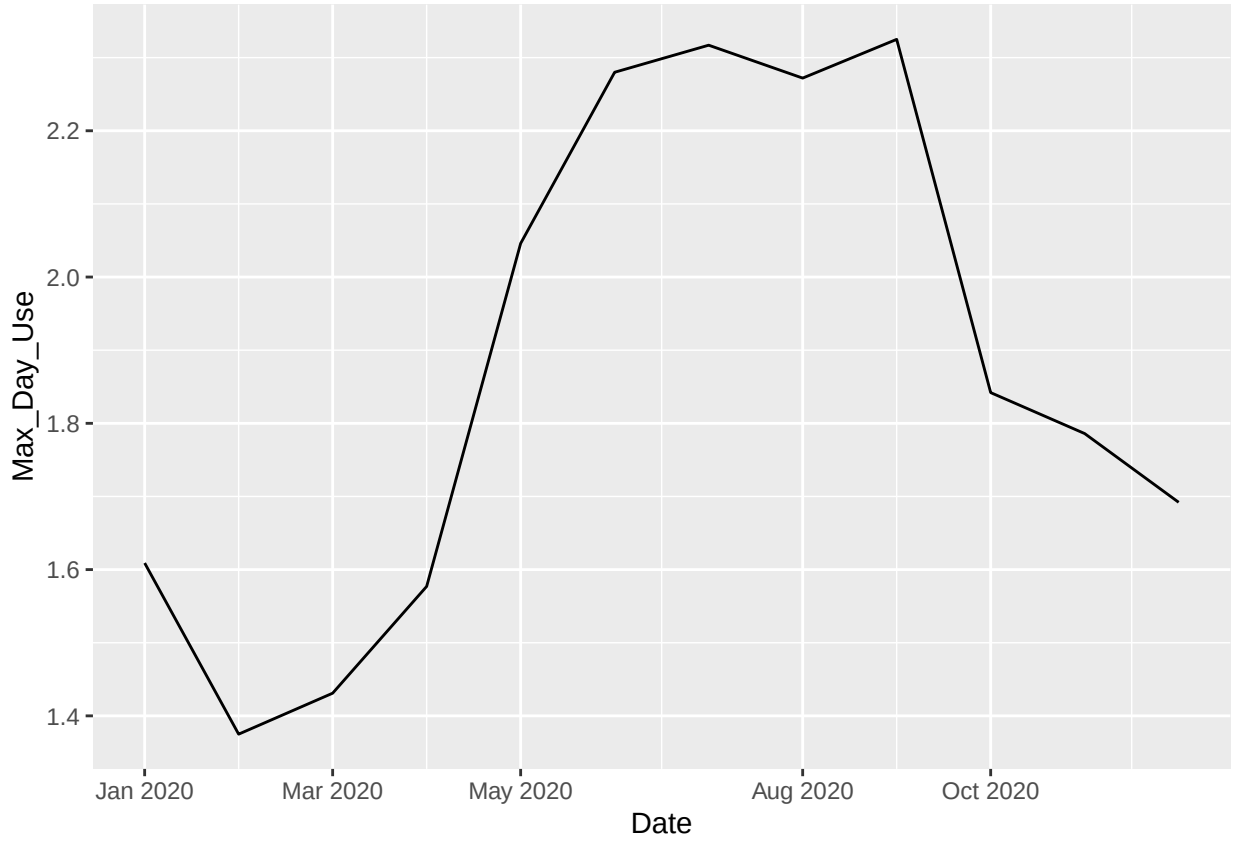

6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
#6.

#Create our scraping function
scrape.it <- function(the_year, the_pwsid){
   #Construct the scraping web address, i.e. its URL
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_scrape_url <- paste0(the_base_url, 'pwsid=', the_pwsid, '&year=', the_year)
print(the_scrape_url)

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

#Set the element address variables (determined in the previous step)

water_system_tag <- "div+ table tr:nth-child(1) td:nth-child(2)"
ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
max_day_use_month_tag <- "th~ td+ td"

#Scrape the data items
the_water_system <- the_website %>%
```

```
  html_nodes(water_system_tag) %>%
  html_text()

the_ownership <- the_website %>%
  html_nodes(ownership_tag) %>%
  html_text()

the_max_day_use_month <- the_website %>%
  html_nodes(max_day_use_month_tag) %>%
  html_text()

#Construct a dataframe from the scraped data
the_df <- data.frame(Water_System = the_water_system,
                     PWSID = the_pwsid,
                     Ownership = the_ownership,
                     Month = months,
                     Date = as.yearmon(paste(month.abb[months],the_year)),
                     Max_Day_Use = as.numeric(the_max_day_use_month))

the_df <- arrange(the_df, Month)

  #Return the dataframe
  return(the_df)
}

example.df <- scrape.it(2020,"04-28-025")
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=04-28-025&year=2020"
```

```
ggplot(example.df, aes(x=Date,y=Max_Day_Use)) +
  geom_line()
```

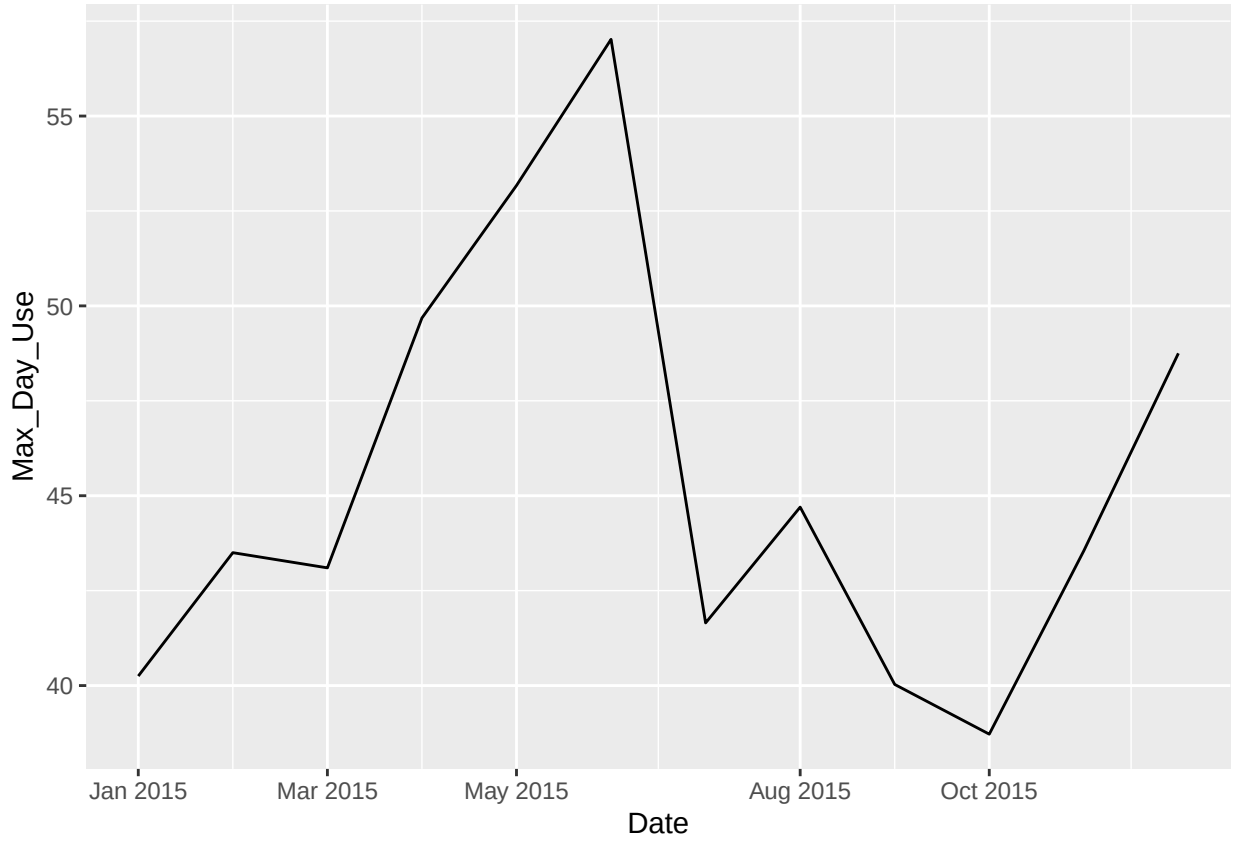



7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7

durham.df <- scrape.it(2015,'03-32-010')
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2015"
```

```
ggplot(durham.df, aes(x=Date,y=Max_Day_Use)) +
  geom_line()
```

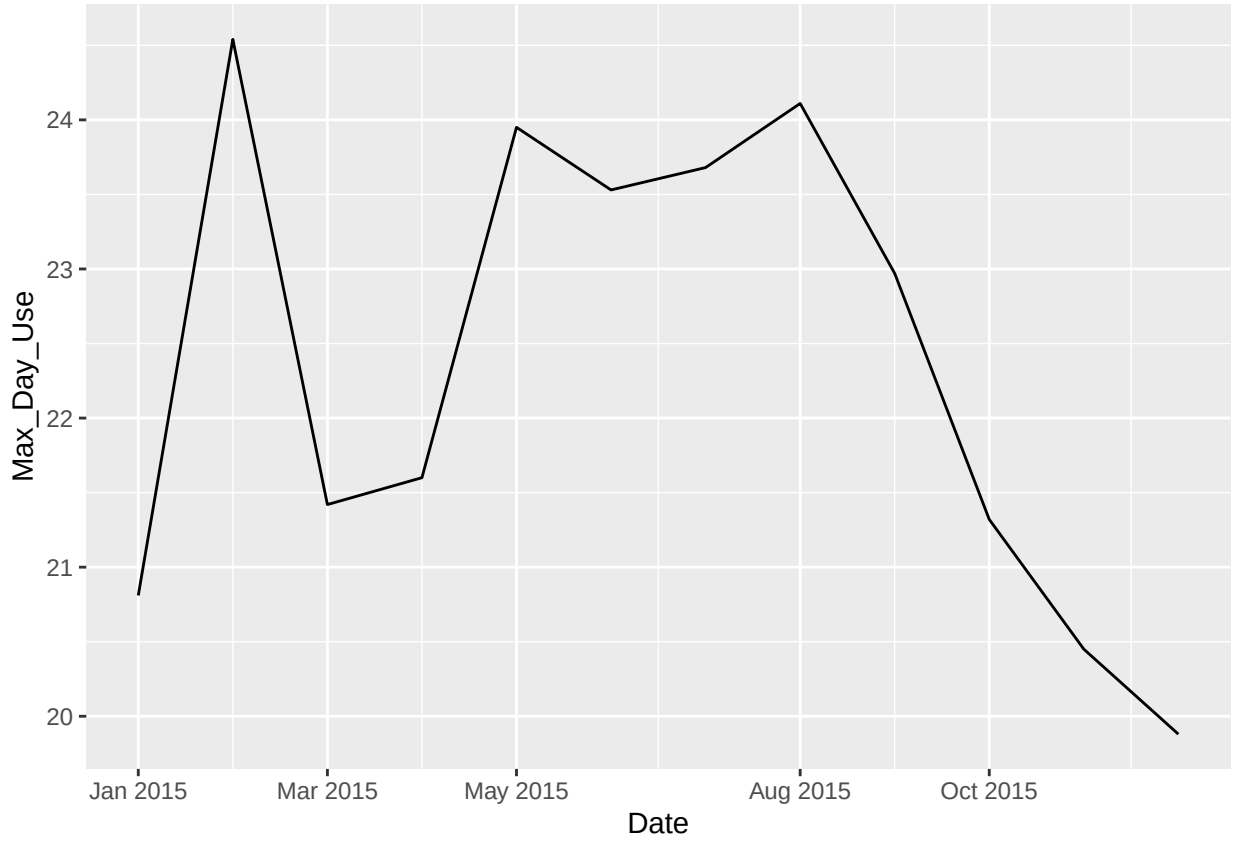



8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

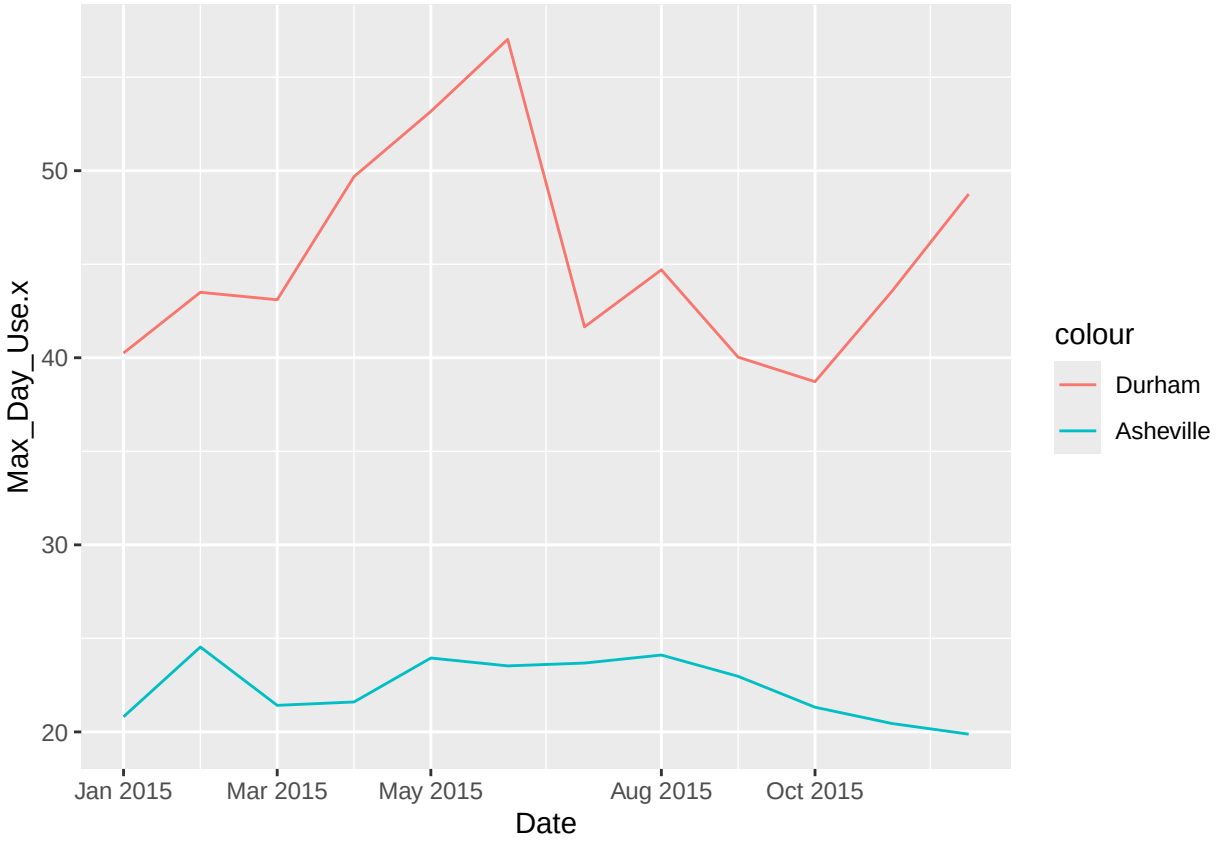
```
#8

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

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2015"
```

```
ggplot(asheville.df, aes(x=Date,y=Max_Day_Use)) +
  geom_line()
```




```
both_df <- left_join(durham.df,asheville.df,by=c("Month","Date"))

ggplot(both_df) +
  geom_line(aes(x=Date,y=Max_Day_Use.x, color = 'blue'))+
  geom_line(aes(x=Date,y=Max_Day_Use.y,color = 'red'))+
  scale_color_discrete(labels = c("Durham","Asheville"))
```




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.

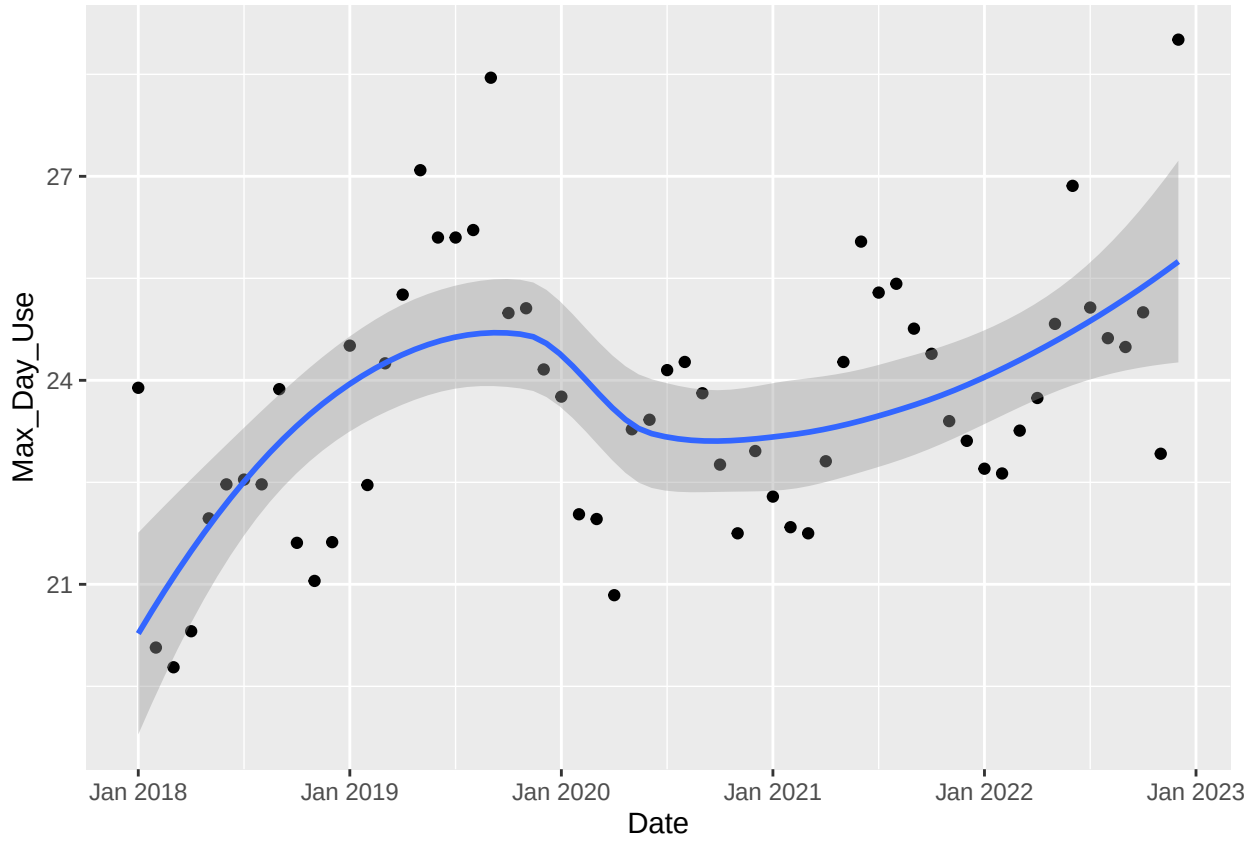
```
#9


the_years = c(2018,2019,2020,2021,2022)

id_for_df <- rep("01-11-010",5)

ash_over_the_years <- map2(the_years, id_for_df, 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"
```

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

```r
#Plot
ggplot(ash_years_combo,aes(x = Date, y=Max_Day_Use)) +
  geom_point()+
  geom_smooth(method = 'loess')
```

```
## `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. Water usage is increasing over time. It was <21 MGD at the beginning of 2018 and increased until about December 2019. It decreased slightly until the summer of 2020, and has been increasing consistently since.