Assignment 10: Data Scraping

Danielle Butler

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
#Install familiar packages
library(tidyverse);library(lubridate);library(viridis);library(here)
here()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

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

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

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

# Set theme
mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

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
#Fetch the web resources from the URL
webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024')
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_name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water_system_name
```

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

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

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

```
ownership <- webpage %>%
  html_nodes("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] "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"
```

```
month <- webpage %>%
  html_nodes(".fancy-table:nth-child(30) tr+ tr th") %>%
  html_text()
month
```

```
##  [1] "Jan" "May" "Sep" "Feb" "Jun" "Oct" "Mar" "Jul" "Nov" "Apr" "Aug" "Dec"
```

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.

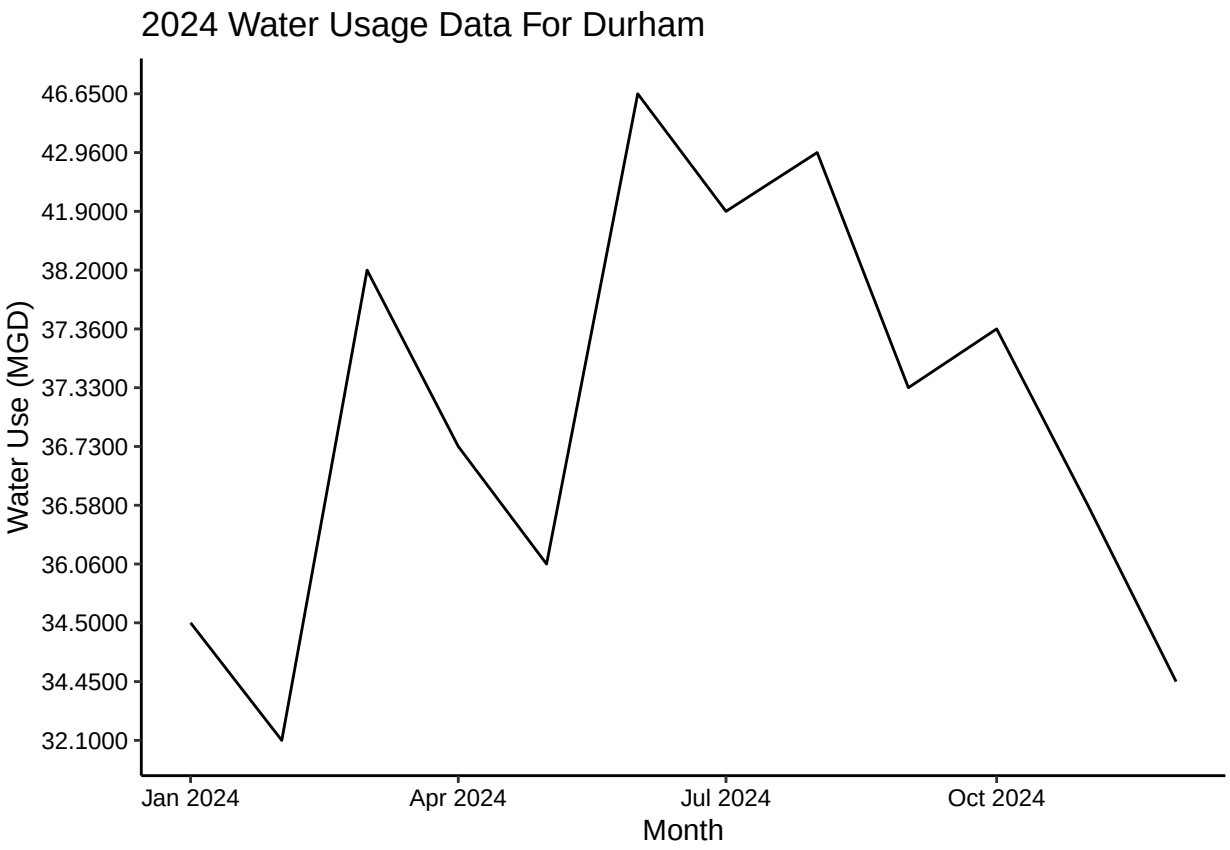
```
#4
df_water_use <- data.frame(Max_Day_Use = max_day_use, Month = month,
                           Ownership = ownership, PWSID = pwsid,
                           Water_System_Name = water_system_name)

df_water_use$Month <- as.Date(paste("2024", df_water_use$Month, "01",
                                    sep = "-"), format = "%Y-%b-%d")
df_water_use$month_label <- format(df_water_use$Month, "%b")

#5
library(ggplot2)

ggplot(df_water_use, aes(x = Month, y = max_day_use,
```

```
                        group = 1)) +
  geom_line() +
  labs(title = paste("2024 Water Usage Data For", water_system_name),
    y = "Water Use (MGD)",
    x = "Month")
```



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.
#Construct the scraping web address, i.e. its URL
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_pwsid <- '03-32-010'
the_year <- 2024
the_scrape_url <- paste0(the_base_url, 'pwsid=', the_pwsid, '&year=', the_year)
print(the_scrape_url)
```

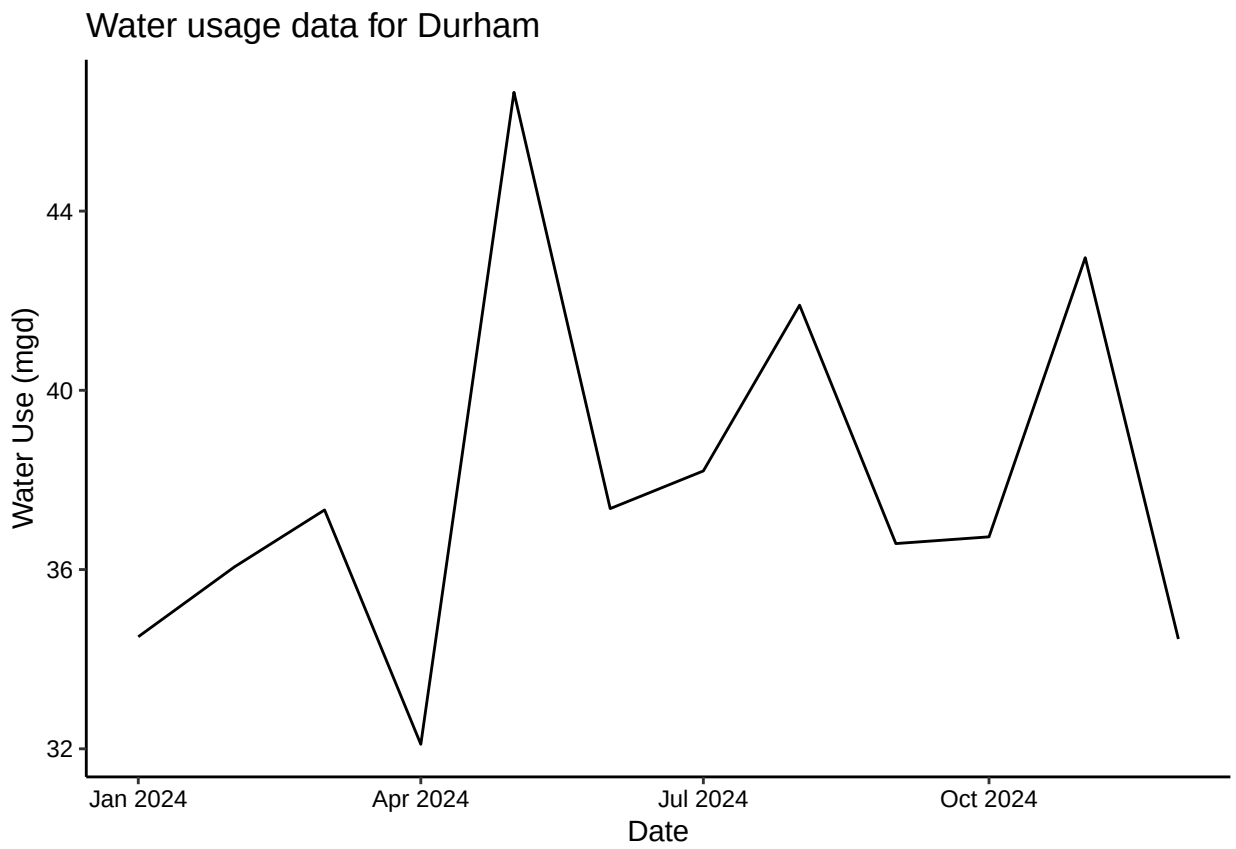
```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2024"
```

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

#Set the element address variables (determined in the previous step)
the_water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
the_pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
the_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
the_data_tag <- 'th~ td+ td'
the_month_tag <- '.fancy-table:nth-child(30) tr+ tr th'

#Scrape the data items
the_water_system_name <- the_website %>% html_nodes(the_water_system_name_tag) %>% html_text()
the_pwsid <- the_website %>%  html_nodes(the_pwsid_tag) %>%  html_text()
the_ownership <- the_website %>% html_nodes(the_ownership_tag) %>% html_text()
the_data <- the_website %>% html_nodes(the_data_tag) %>% html_text()
the_month <- the_website %>% html_nodes(the_month_tag) %>% html_text()

#Construct a dataframe from the scraped data
df_water_use <- data.frame("Month" = rep(1:12),
                           "Year" = rep(the_year,12),
                           "Max_Water_Use" = as.numeric(the_data)) %>%
  mutate(Ownership = !!the_ownership,
         Water_System_Name = !!the_water_system_name,
         PWSID = !!the_pwsid,
         Year = !!the_year,
         Date = my(paste(Month,"-",Year)))
#plot
library(ggplot2)
ggplot(df_water_use,aes(x=Date,y=Max_Water_Use)) +
  geom_line() +
  labs(title = paste("Water usage data for",the_water_system_name),
       y="Water Use (mgd)",
       x="Date")
```




```
#Create our scraping function
scrape.it <- function(the_pwsid, the_year){

#Retrieve the website contents
the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?',
                                'pwsid=', the_pwsid, '&year=', the_year))

#Set the element address variables (determined in the previous step)
the_water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
the_pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
the_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
the_data_tag <- 'th~ td+ td'
the_month_tag <- '.fancy-table:nth-child(30) tr+ tr th'

#Scrape the data items
the_water_system_name <- the_website %>% html_nodes(the_water_system_name_tag) %>% html_text()
the_pwsid <- the_website %>%  html_nodes(the_pwsid_tag) %>%  html_text()
the_ownership <- the_website %>% html_nodes(the_ownership_tag) %>% html_text()
the_data <- the_website %>% html_nodes(the_data_tag) %>% html_text()
the_month <- the_website %>% html_nodes(the_month_tag) %>% html_text()

#Construct a dataframe from the scraped data
df_water_use <- data.frame("Month" = rep(1:12),
                           "Year" = rep(the_year,12),
                           "Max_Water_Use" = as.numeric(the_data)) %>%
  mutate(Ownership = !!the_ownership,
```

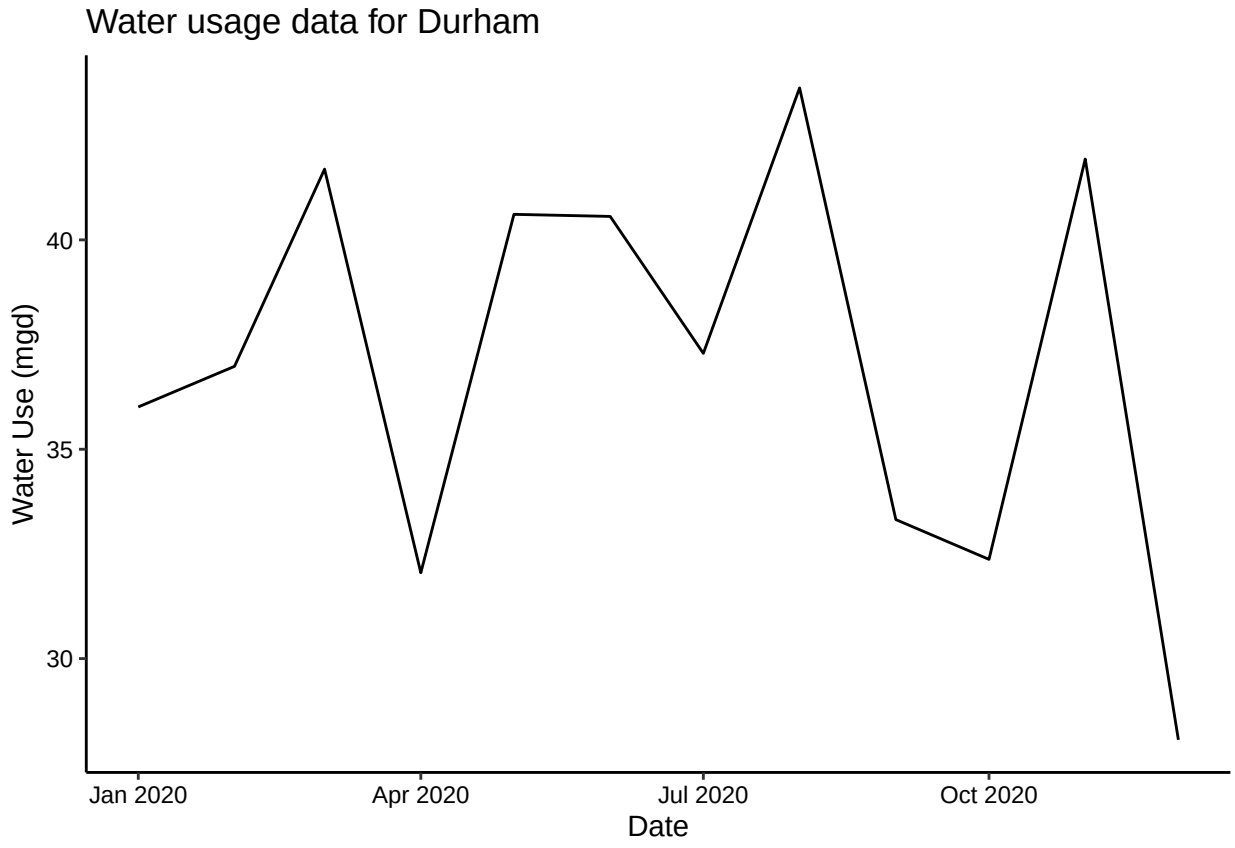
```
          Water_System_Name = !!the_water_system_name,
          PWSID = !!the_pwsid,
          Year = !!the_year,
          Date = my(paste(Month,"-",Year)))

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

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

```
#7 Extract
durham_df <- scrape.it('03-32-010',2020)
view(durham_df)

#Plot
library(ggplot2)
ggplot(durham_df,aes(x=Date,y=Max_Water_Use)) +
  geom_line() +
  labs(title = paste("Water usage data for",the_water_system_name),
       y="Water Use (mgd)",
       x="Date")
```

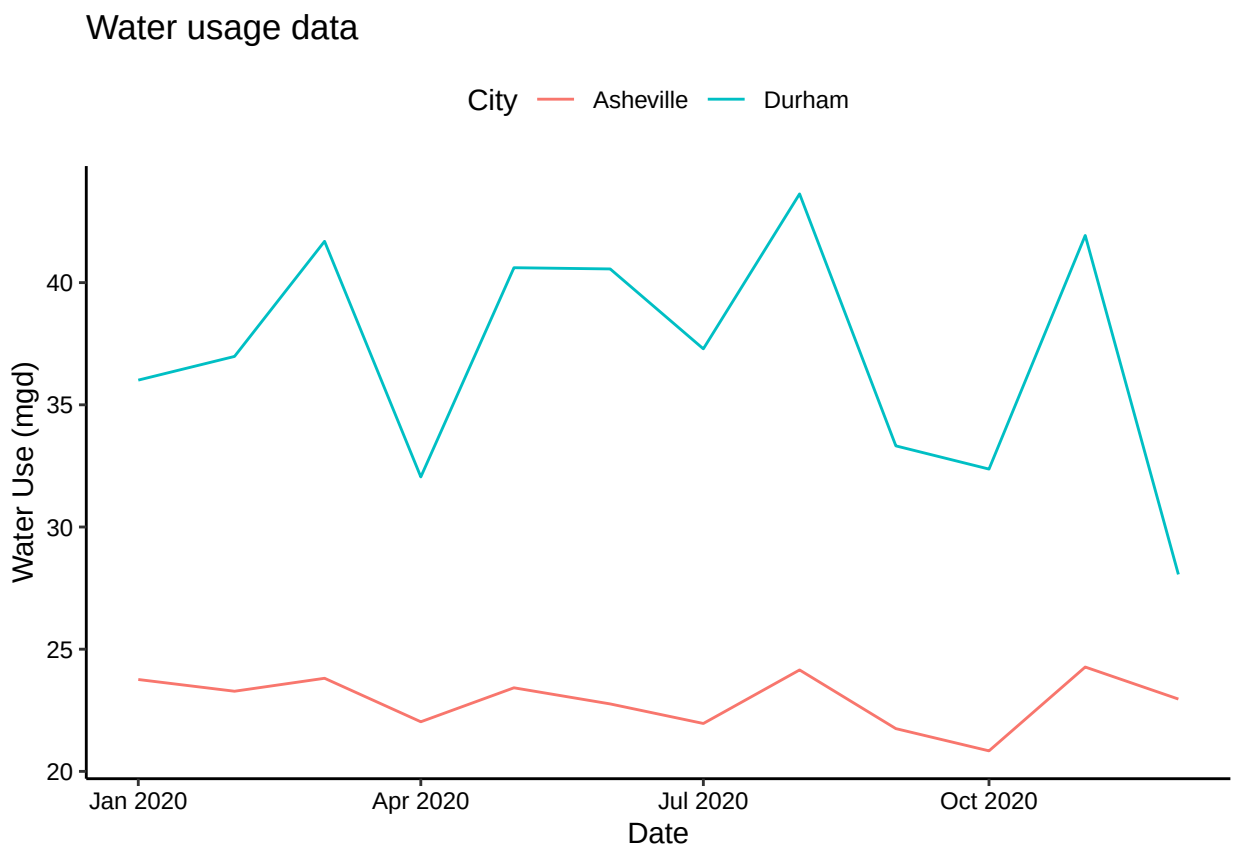


8. Use the function above to extract data for Asheville (PWSID = ‘01-11-010’) in 2020. Combine this data with the Durham data collected above and create a plot that compares Asheville’s to Durham’s water withdrawals.

```r
#8
asheville_df <- scrape.it('01-11-010',2020)
view(asheville_df)

combined_data <- bind_rows(asheville_df,durham_df)

#Plot
ggplot(combined_data,aes(x=Date,y=Max_Water_Use, color=Water_System_Name, group = Water_System_Name)) +
  geom_line() +
  labs(title = paste("Water usage data"),
       y="Water Use (mgd)",
       x="Date",
       color="City")
```



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

the_years = rep(2018:2023)
```

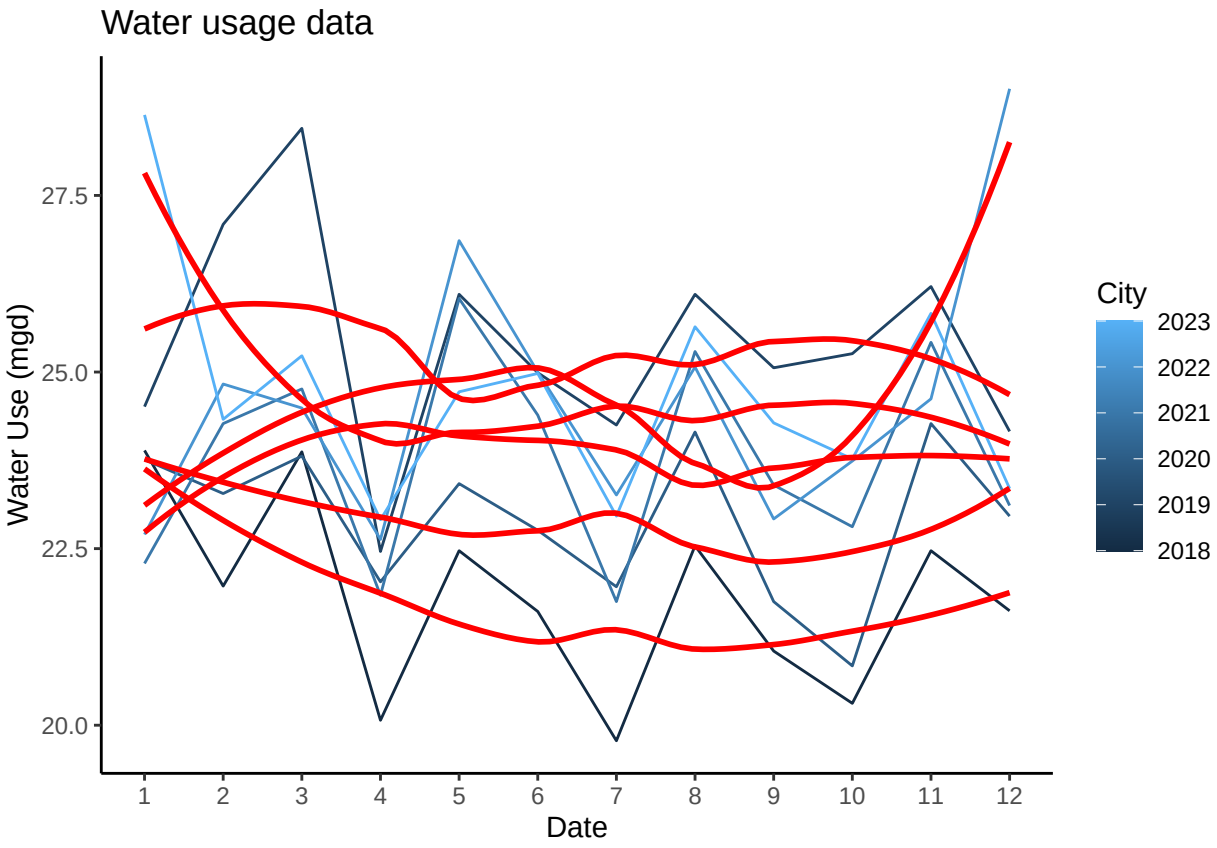
```
my_pwsid = '01-11-010'

the_dfs <- lapply(X = the_years,
                  FUN = scrape.it,
                  the_pwsid=my_pwsid)

the_df_multiyear <- bind_rows(the_dfs)

ggplot(the_df_multiyear,aes(x=Month,y=Max_Water_Use,color=Year,group=Year)) +
  geom_line() +
  scale_x_continuous(breaks = 1:12) +
  geom_smooth(method="loess",se=FALSE,color='red') +
  theme_classic() +
  labs(title = paste("Water usage data"),
       y="Water Use (mgd)",
       x="Date",
       color="City")
```

```
## `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: From what I can tell, no. It varies monthly, but cannot say definitively if it varies over year. >