Assignment 10: Data Scraping

Rachel Williams

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(rvest)
library(lubridate)
library(here)
library(dataRetrieval)
library(dplyr)

here("/home/guest/EDE_Fall2024")
```

```
## [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

webpage <- 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

System_name <-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()
MGD <- webpage %>%
  html_nodes(".fancy-table:nth-child(48) 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

data <- data.frame (System_name, PWSID, Ownership, MGD)

Month <- webpage %>%
  html_nodes(".fancy-table:nth-child(48) tr+ tr th") %>%
  html_text()

clean_data <- data %>%
  mutate(Month=Month) %>%
  mutate(Year=rep(2023)) %>%
  mutate(Date= as.Date(my(paste(Month,"-",Year))))

clean_data$MGD <- as.numeric(clean_data$MGD)


#5

class(clean_data$MGD)
```

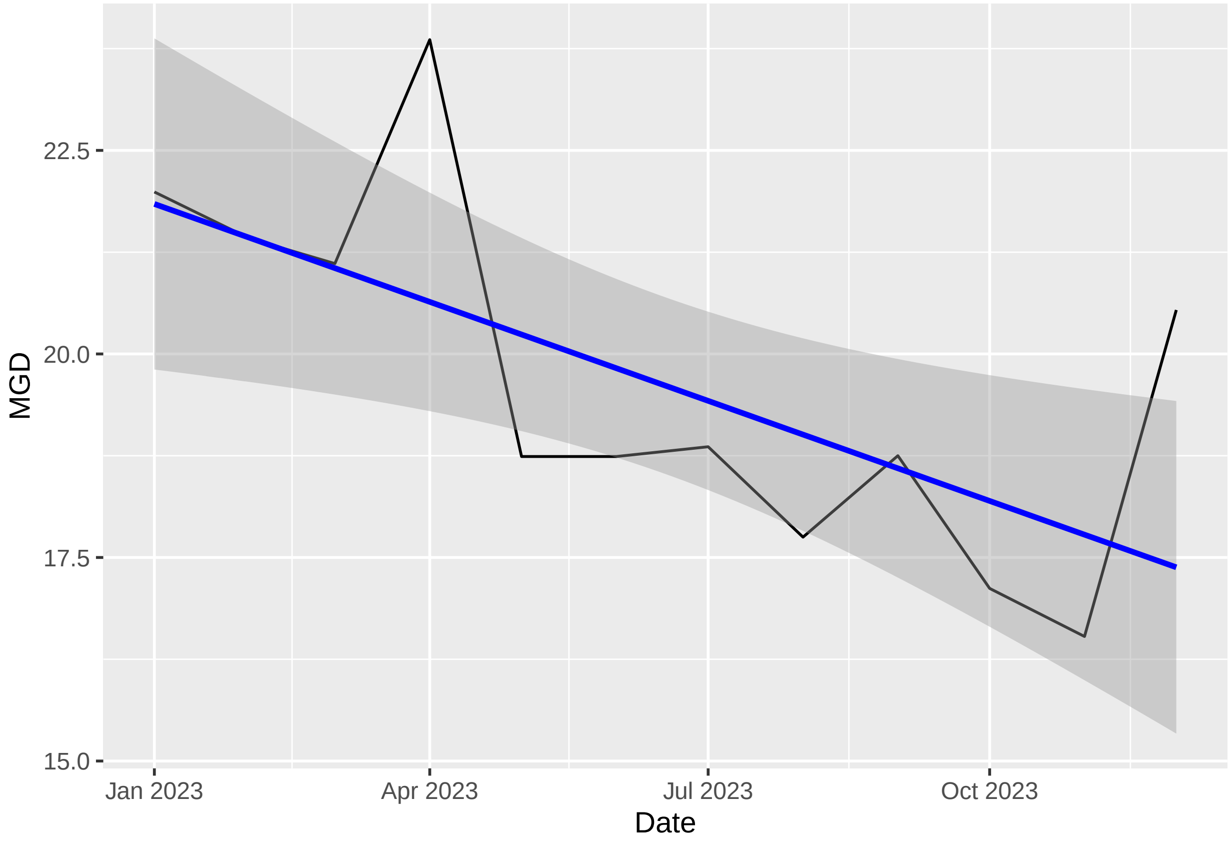
```
## [1] "numeric"
```

```
class(Month)
```

```
## [1] "character"
```

```
ggplot(clean_data, aes(x=Date, y=MGD)) +
  geom_line() + geom_smooth(method = "lm", col = "blue")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



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.

scrape_it <- function (PWSID, the_year){
  web.address <-paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',PWSID,'&year=',the_yea
  print(web.address)

the_url <- read_html(web.address)

System_name <-the_url %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
PWSID <-the_url %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
Ownership <-the_url %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
MGD <-the_url %>%
  html_nodes("th~ td+ td") %>%
  html_text()
clean_data2 <- data.frame (
  "System Name" = rep(System_name, 12),
  "PWSID" = rep(PWSID,12),
  "Ownership" = rep(Ownership, 12),
  "MGD" = as.numeric(MGD),
"Month"= c("Jan", "May", "Sep", "Feb", "Jun","Oct","Mar","Jul","Nov","Apr","Aug","Dec"),
"Year"= rep(the_year, 12) )

clean_data2$Date <- as.Date(my(paste(clean_data2$Month,"-",clean_data2$Year)))

return(clean_data2)
}
```

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

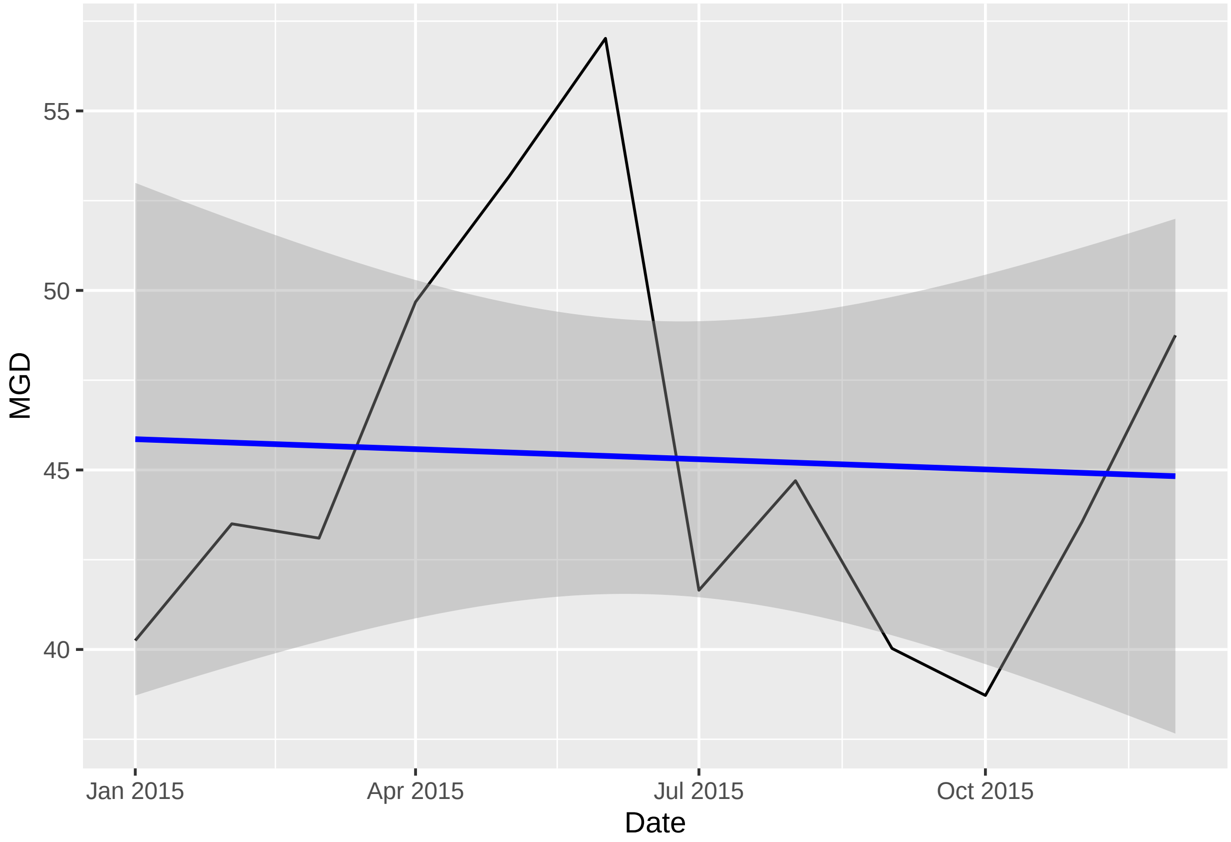
```
#7

Durham <- scrape_it('03-32-010',2015)
```

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

```
ggplot(Durham, aes(x=Date, y=MGD)) +
  geom_line() + geom_smooth(method = "lm", col = "blue")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



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.

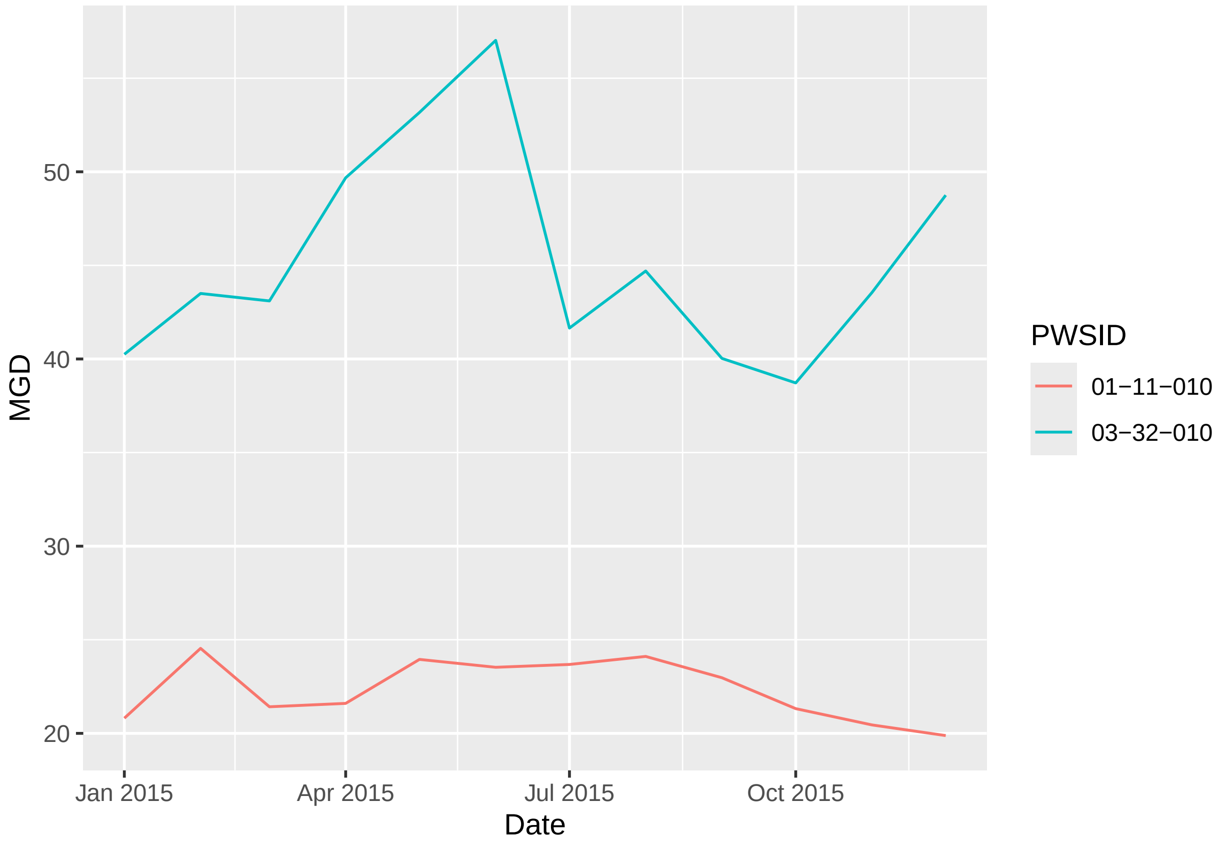
```
#8

Asheville <- scrape_it('01-11-010',2015)
```

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

```
combined_data <- rbind(Durham, Asheville)

ggplot(combined_data, aes(x=Date, y=MGD, color=PWSID)) +
  geom_line()
```



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.

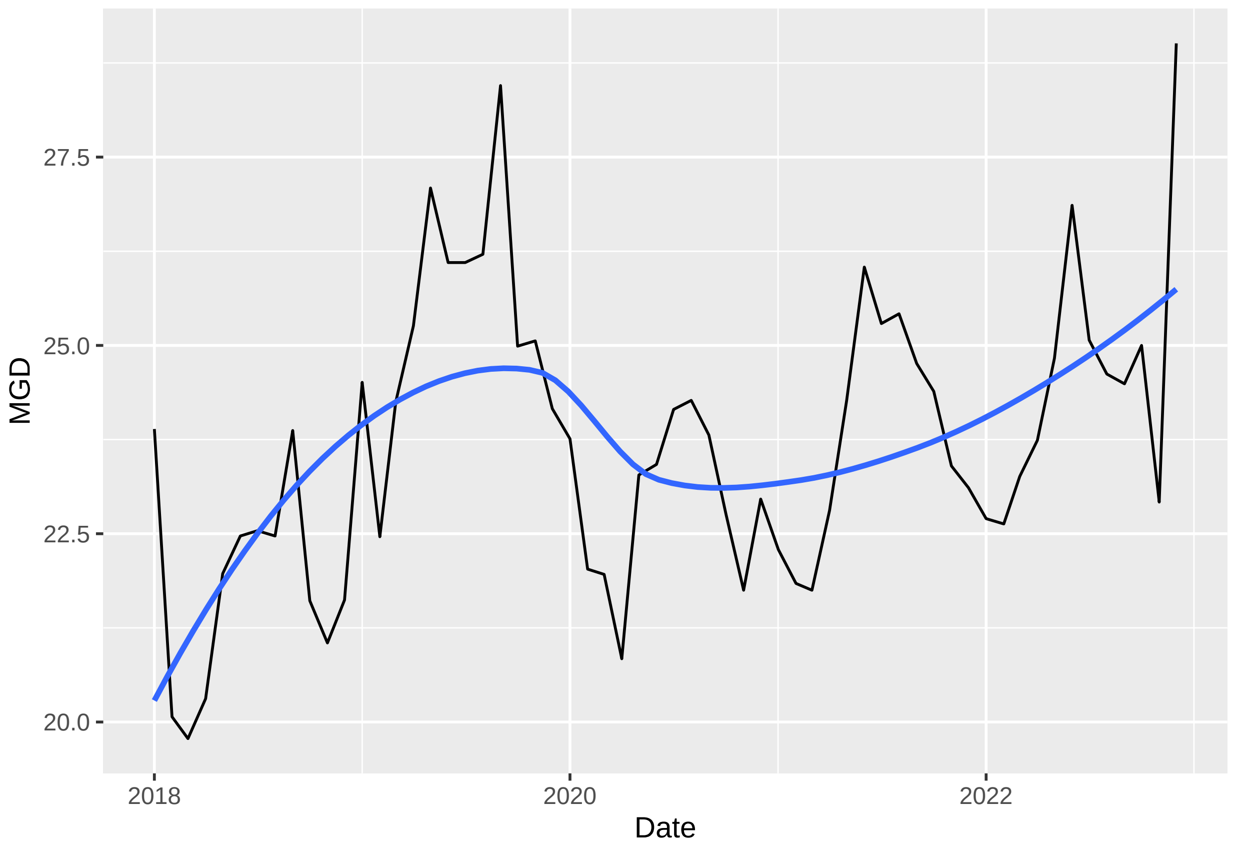
```
#9

Asheville2 <- seq(2018,2022) %>% map(~scrape_it("01-11-010",.)) %>% bind_rows()
```

```
## [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"
```

```
ggplot(Asheville2, aes(x=Date, y=MGD)) +
  geom_line() + geom_smooth(method="loess",se=FALSE)
```

```
## '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: By looking at the plot, Asheville appears to be increasing in water usage over time. >