Assignment 09: Data Scraping

Jacob Freedman

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Rename this file `<FirstLast>_A09_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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Set your ggplot theme

```
# 1
setwd("~/R/EDA-Fall2022")
getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```
library(tidyverse)
library(rvest)
library(lubridate)

theme1 <- theme_classic(base_size = 12) + theme(axis.text = element_text(color = "black"),
    legend.position = "left", axis.line = element_line(arrow = arrow()))
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2021 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=2021

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

```
# 2
DurhamWaterWebpage <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=
DurhamWaterWebpage
```

```
## {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 Daily 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), with the first value being "27.6400".

```
# 3
water_system_name <- DurhamWaterWebpage %>%
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
    html_text()
water_system_name
```

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

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

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

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

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

```
max_withdrawals_mgd <- DurhamWaterWebpage %>%
    html_nodes("th~ td+ td") %>%
    html_text()
max_withdrawals_mgd
```

```
##  [1] "27.6400" "41.7900" "36.7200" "27.9700" "37.9500" "42.2400" "30.5400"
##  [8] "43.6200" "31.2800" "33.7600" "46.0800" "29.7800"
```

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.

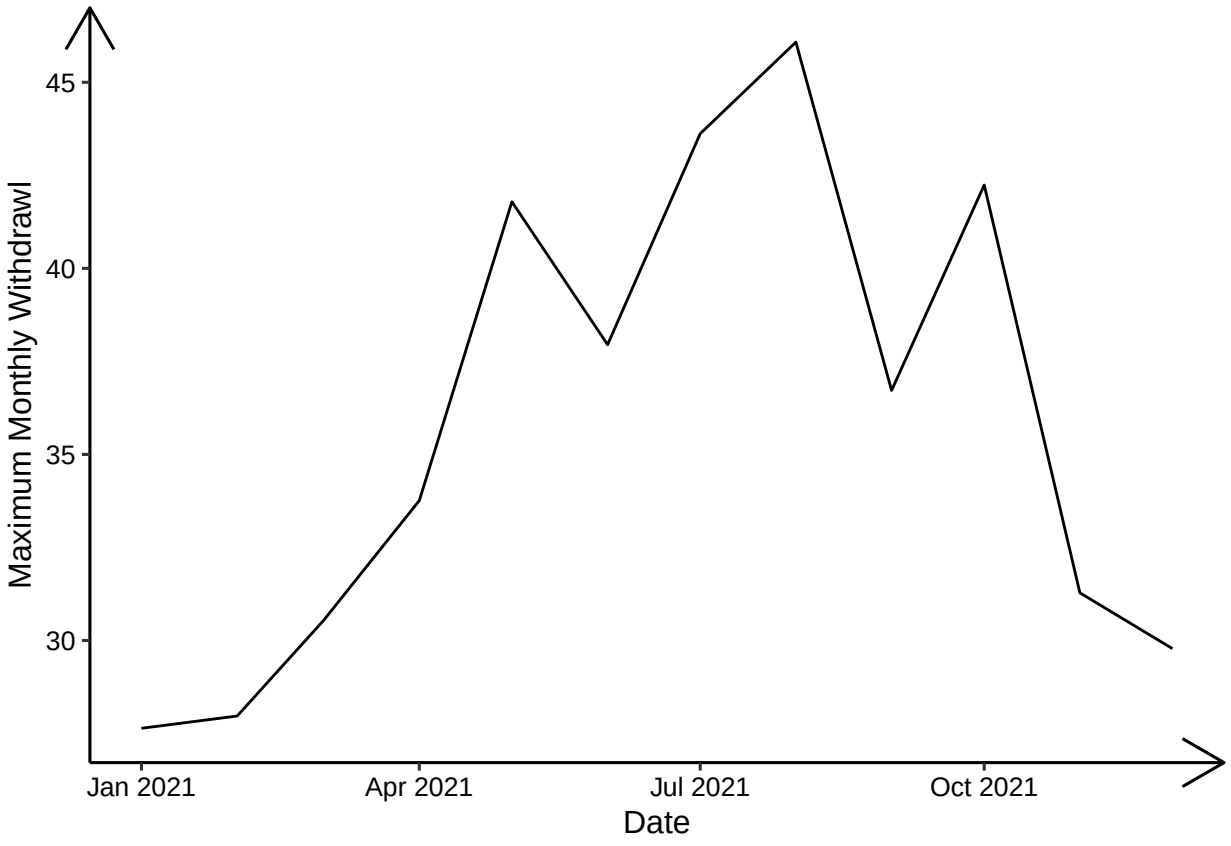
NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc...

5. Create a line plot of the maximum daily withdrawals across the months for 2021

```
# 4
df_max_withdrawals <- data.frame(Month_abbr = c("Jan", "May", "Sep", "Feb", "Jun",
    "Oct", "Mar", "July", "Nov", "Apr", "Aug", "Dec"), Month = c(1, 5, 9, 2, 6, 10,
    3, 7, 11, 4, 8, 12), Year = rep(2021, 12), Max_withdrawals_mgd = as.numeric(max_withdrawals_mgd))

df_max_withdrawals <- df_max_withdrawals %>%
    mutate(Ownership = !!ownership, PWSID = !!pwsid, Water_System = !!water_system_name,
        Date = my(paste(Month, "-", Year))) %>%
    arrange(Month)

# 5
Max_withdrawal_plot <- ggplot(df_max_withdrawals, aes(x = Date, y = Max_withdrawals_mgd)) +
    geom_line() + theme1 + ylab("Maximum Monthly Withdrawl")
print(Max_withdrawal_plot)
```



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. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```
# 6.
the_base_url <- "https://www.ncwater.org/WUDC/app/LWSP/report.php"
pwsid <- "03-32-010"
```

```
the_year <- 2019
Scrape_url <- paste0(the_base_url, "?pwsid=", pwsid, "&year=", the_year)
print(Scrape_url)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019"
```

```
the_website <- read_html(Scrape_url)

Scrape_function <- function(the_year, pwsid) {

    The_website <- read_html(paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php",
        "?pwsid=", pwsid, "&year=", the_year))

    water_system_tag <- "div+ table tr:nth-child(1) td:nth-child(2)"
    PWSID_tag <- "td tr:nth-child(1) td:nth-child(5)"
    Ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
    Max_withdrawal_tag <- "th~ td+ td"

    water_system_name <- The_website %>%
        html_nodes(water_system_tag) %>%
        html_text()
    pwsid <- The_website %>%
        html_nodes(PWSID_tag) %>%
        html_text()
    ownership <- The_website %>%
        html_nodes(Ownership_tag) %>%
        html_text()
    max_withdrawals_mgd <- The_website %>%
        html_nodes(Max_withdrawal_tag) %>%
        html_text()

    df_max_withdrawals <- data.frame(Month_abbr = c("Jan", "May", "Sep", "Feb", "Jun",
        "Oct", "Mar", "July", "Nov", "Apr", "Aug", "Dec"), Month = c(1, 5, 9, 2,
        6, 10, 3, 7, 11, 4, 8, 12), Year = rep(the_year, 12), Max_withdrawals_mgd = as.numeric(max_withd

    df_max_withdrawals <- df_max_withdrawals %>%
        mutate(Ownership = !!ownership, PWSID = !!pwsid, Water_System = !!water_system_name,
            Date = my(paste(Month, "-", Year))) %>%
        arrange(Month)

    Sys.sleep(1)

    return(df_max_withdrawals)
}
```

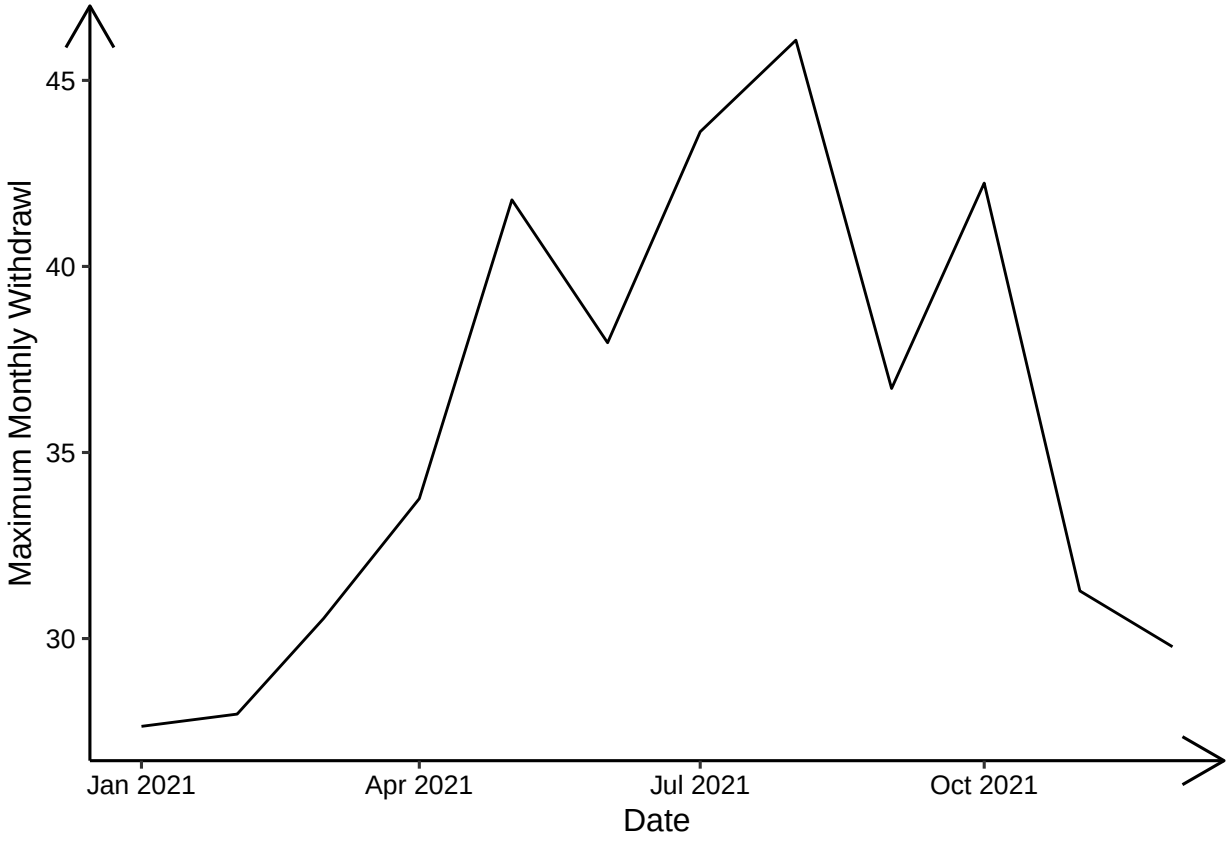
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
# 7
Scraped_df <- Scrape_function(2015, "03-32-010")
head(Scraped_df)
```

```
##   Month_abbr Month Year Max_withdrawals_mgd    Ownership     PWSID Water_System
## 1        Jan     1 2015               40.25 Municipality 03-32-010       Durham
## 2        Feb     2 2015               43.50 Municipality 03-32-010       Durham
```

```
## 3        Mar     3 2015              43.10 Municipality 03-32-010       Durham
## 4        Apr     4 2015              49.68 Municipality 03-32-010       Durham
## 5        May     5 2015              53.17 Municipality 03-32-010       Durham
## 6        Jun     6 2015              57.02 Municipality 03-32-010       Durham
##         Date
## 1 2015-01-01
## 2 2015-02-01
## 3 2015-03-01
## 4 2015-04-01
## 5 2015-05-01
## 6 2015-06-01
```

```
Max_withdrawal_2015 <- ggplot(df_max_withdrawals, aes(x = Date, y = Max_withdrawals_mgd)) +
    geom_line() + theme1 + ylab("Maximum Monthly Withdrawl")
print(Max_withdrawal_2015)
```



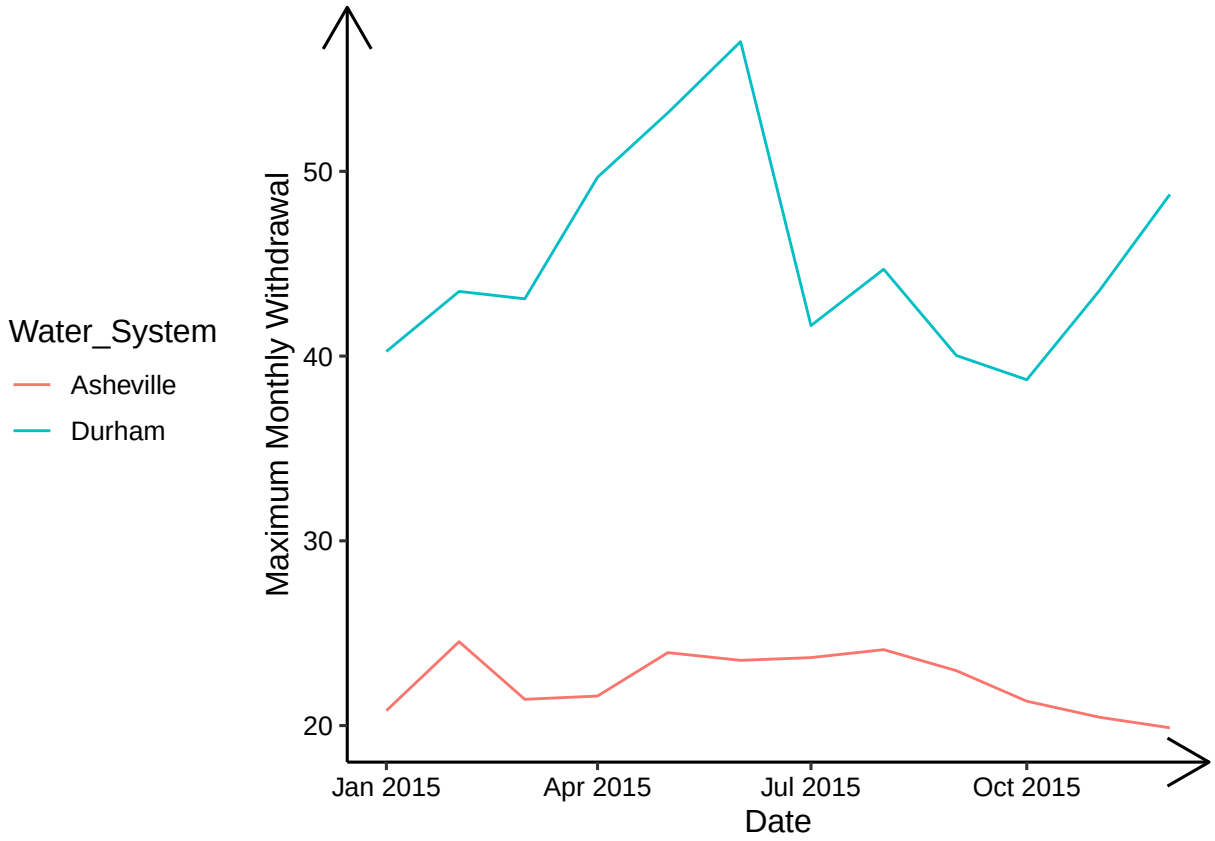
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
# 8
Scraped_df_Asheville <- Scrape_function(2015, "01-11-010")
Scraped_df_Durham <- Scrape_function(2015, "03-32-010")

Asheville_Durham_combined <- rbind(Scraped_df_Asheville, Scraped_df_Durham)

Withdrawal_plot_comparison <- ggplot(Asheville_Durham_combined, aes(x = Date, y = Max_withdrawals_mgd,
```

```
    color = Water_System)) + geom_line() + theme1 + ylab("Maximum Monthly Withdrawal")
print(Withdrawal_plot_comparison)
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

   TIP: See Section 3.2 in the "09_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.

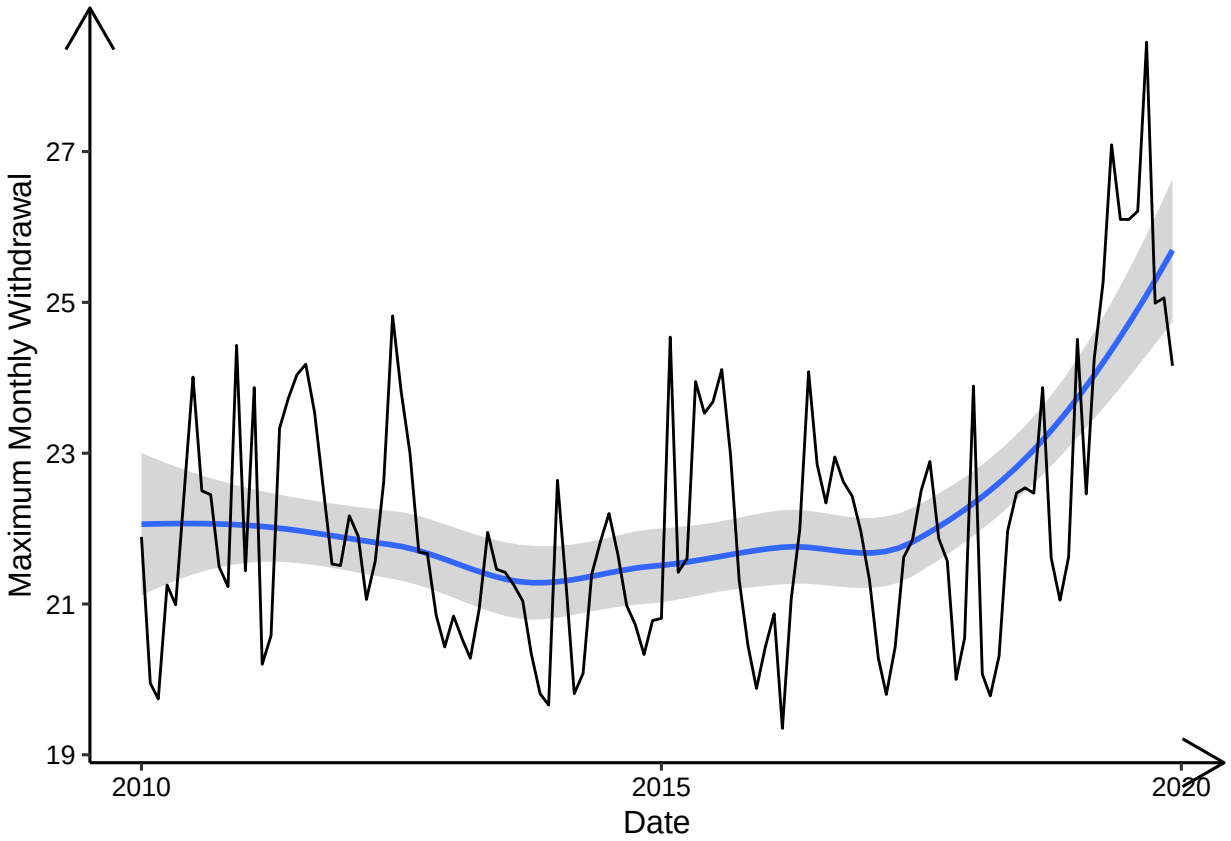
```
# 9

dfs_Asheville_2010to2019 <- map2(seq(2010, 2019), rep("01-11-010", 10), Scrape_function)

Combined_Asheville_2010to2019 <- bind_rows(dfs_Asheville_2010to2019)


Asheville_2010to2019_plot <- ggplot(Combined_Asheville_2010to2019, aes(x = Date,
    y = Max_withdrawals_mgd)) + geom_smooth() + geom_line() + theme1 + ylab("Maximum Monthly Withdrawal
print(Asheville_2010to2019_plot)
```

```
## `geom_smooth()` using method = 'loess' and 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?

The maximum monthly water usage in Asheville appears stable from 2010 - 2017, but jumps dramatically in 2018 and 2019. Perhaps 2018 and 2019 were particularly dry years for the region and the city required more water use to meet constituents' needs.