Assignment 10: Data Scraping

Ye Khaung Oo

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("ggthemes")
library("ggplot2")
library("lubridate")
library("dplyr")
getwd()
```

```
## [1] "/Users/yekhaungoo/Library/CloudStorage/OneDrive-Personal/EDA Class"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2022 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=2022

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

```
# 2
the_website <- read_html("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022")
```

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 <- html_nodes(the_website, "div+ table tr:nth-child(1) td:nth-child(2)") %>%
    html_text(trim = TRUE)
pwsid <- html_nodes(the_website, "td tr:nth-child(1) td:nth-child(5)") %>%
    html_text(trim = TRUE)
ownership <- html_nodes(the_website, "div+ table tr:nth-child(2) td:nth-child(4)") %>%
    html_text(trim = TRUE)
maximum_day_use <- html_nodes(the_website, "th~ td+ td") %>%
    html_text(trim = TRUE) %>%
    as.numeric()
```

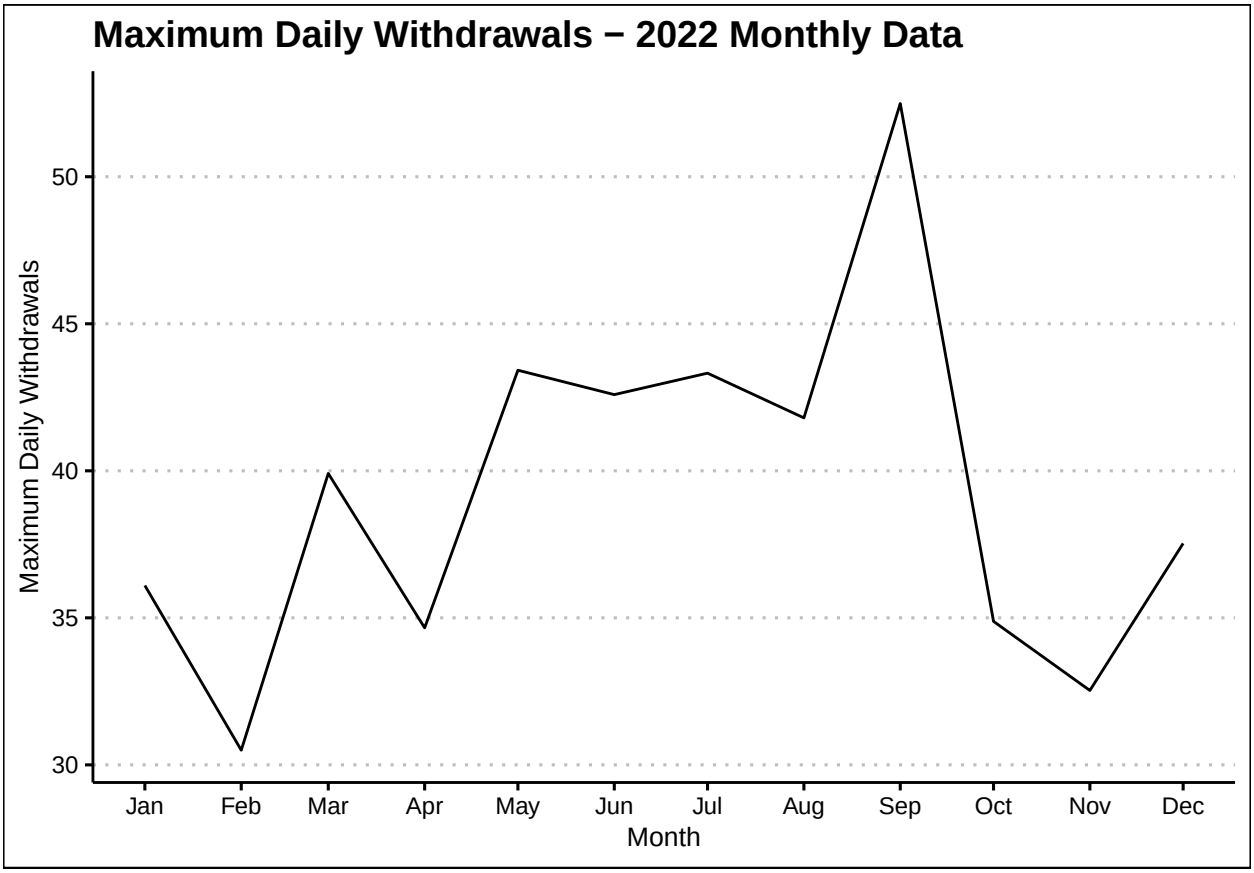
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   > 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 2022

```r
# 4
water_system_name <- rep("Durham", 12)
pwsid <- rep("03-32-010", 12)
ownership <- rep("Municipality", 12)
months <- c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr",
    "Aug", "Dec")
year <- rep("2022", 12)
dates <- as.Date(paste("2022", months, "01", sep = "-"), format = "%Y-%b-%d")

df <- data.frame(df_water_system_name = water_system_name, df_pwsid = pwsid, df_ownership = ownership,
    df_maximum_day_use = maximum_day_use, df_date = dates)

# 5
ggplot(df, aes(x = df_date, y = df_maximum_day_use)) + geom_line(group = 1) + labs(x = "Month",
    y = "Maximum Daily Withdrawals") + scale_x_date(date_labels = "%b", date_breaks = "1 month") +
    theme_clean() + ggtitle("Maximum Daily Withdrawals - 2022 Monthly Data")
```



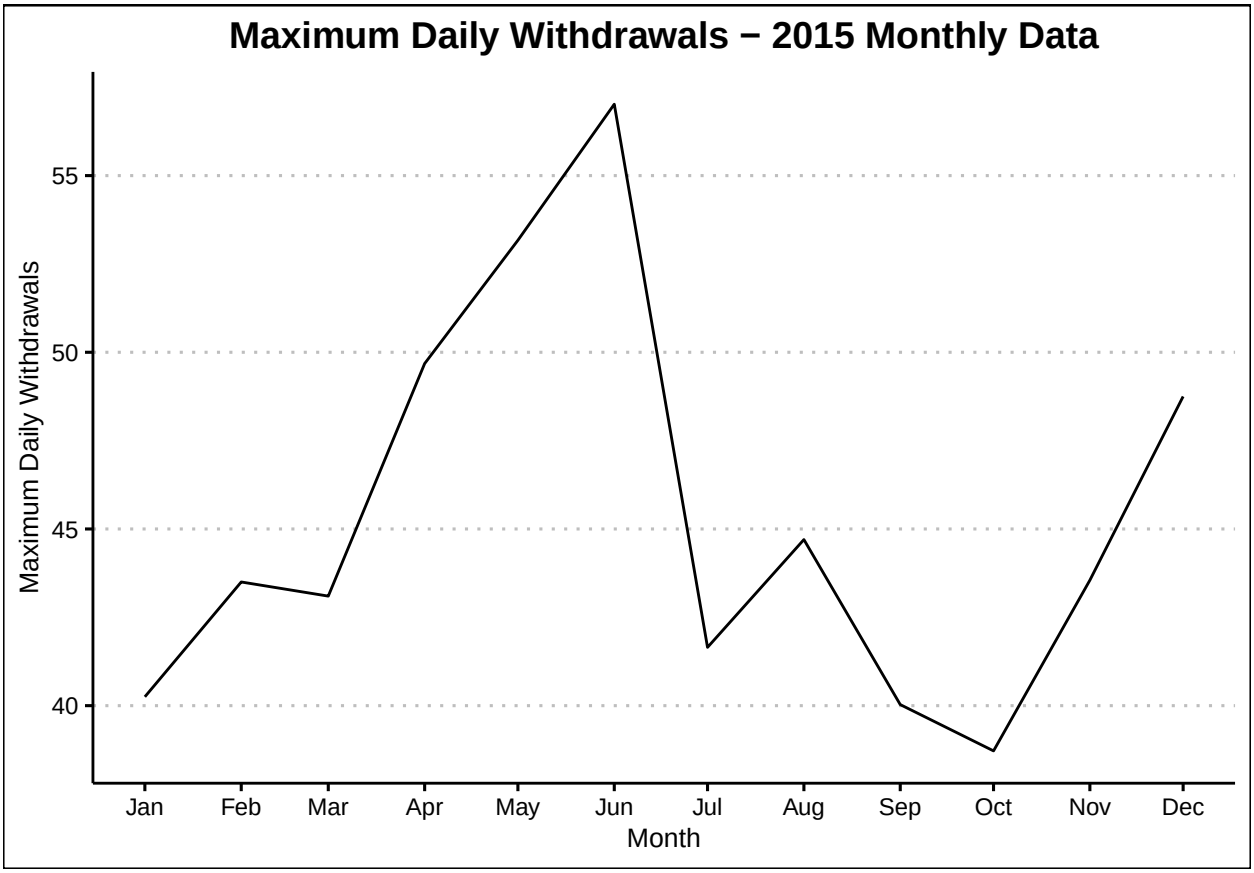
6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```r
# 6.
scraping_data <- function(pwsid, year) {
    the_url <- paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=",
        pwsid, "&year=", year)
```

```
    the_website <- read_html(the_url)

    # scraping data from website
    water_system_name <- html_nodes(the_website, "div+ table tr:nth-child(1) td:nth-child(2)") %>%
        html_text(trim = TRUE)
    pwsid <- html_nodes(the_website, "td tr:nth-child(1) td:nth-child(5)") %>%
        html_text(trim = TRUE)
    ownership <- html_nodes(the_website, "div+ table tr:nth-child(2) td:nth-child(4)") %>%
        html_text(trim = TRUE)
    maximum_day_use <- html_nodes(the_website, "th~ td+ td") %>%
        html_text(trim = TRUE) %>%
        as.numeric()

    # assigning months and mutating into dates
    months <- c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr",
        "Aug", "Dec")
    dates <- as.Date(paste(year, months, "01", sep = "-"), format = "%Y-%b-%d")

    # creating dataframe
    df <- data.frame(df_water_system_name = rep(water_system_name, 12), df_pwsid = rep(pwsid,
        12), df_ownership = rep(ownership, 12), df_maximum_day_use = maximum_day_use,
        df_months = months, df_year = rep(year, 12), df_date = dates)
}
```

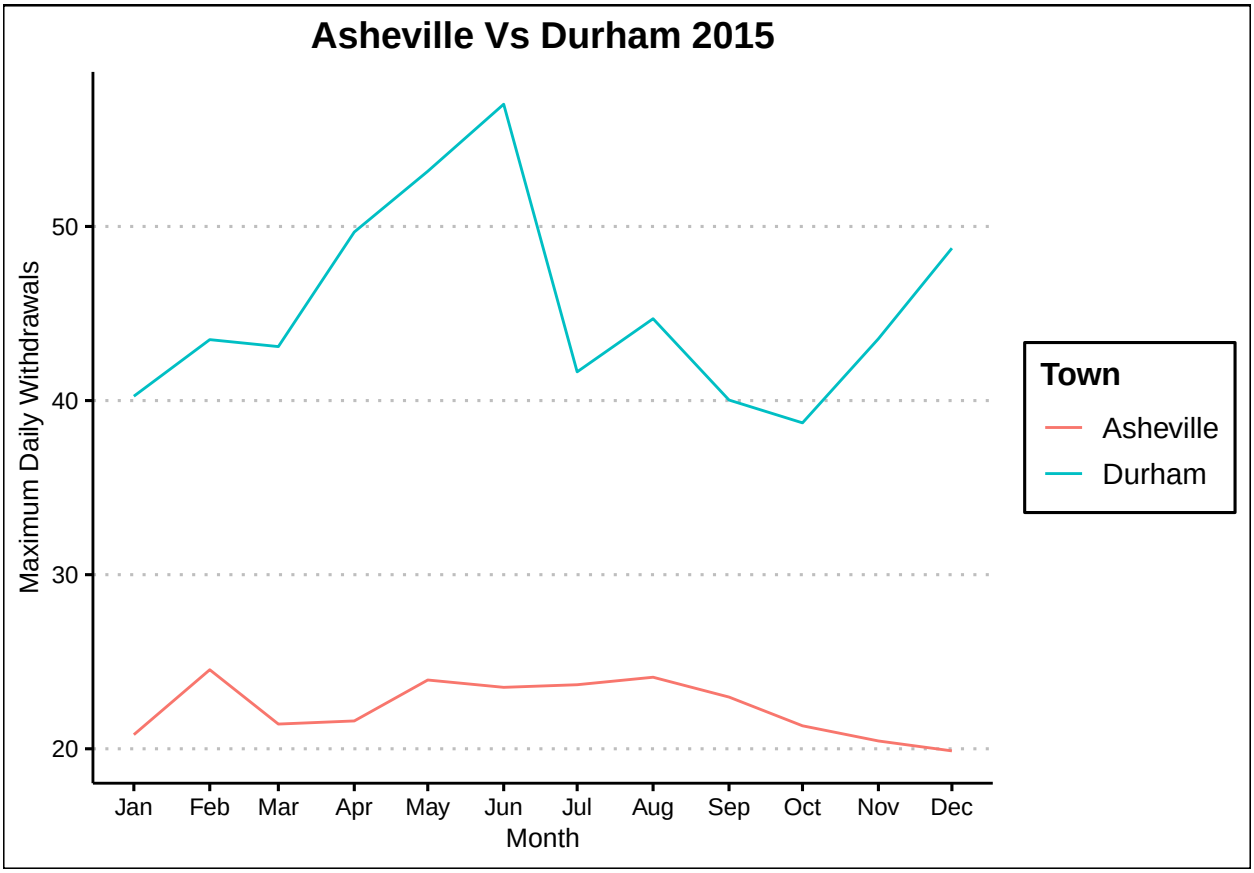
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
# 7
durham <- scraping_data("03-32-010", "2015")

ggplot(durham, aes(x = df_date, y = df_maximum_day_use)) + geom_line(group = 1) +
    labs(x = "Month", y = "Maximum Daily Withdrawals") + scale_x_date(date_labels = "%b",
    date_breaks = "1 month") + theme_clean() + ggtitle("Maximum Daily Withdrawals - 2015 Monthly Data")
    theme(plot.title = element_text(hjust = 0.5))
```



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 <- scraping_data("01-11-010", "2015")

two_towns <- bind_rows(durham, asheville)

ggplot(two_towns, aes(x = df_date, y = df_maximum_day_use, color = df_water_system_name)) +
    geom_line() + labs(x = "Month", y = "Maximum Daily Withdrawals", color = "Town") +
    theme_clean() + scale_x_date(date_labels = "%b", date_breaks = "1 month") + ggtitle("Asheville Vs Du
    theme(plot.title = element_text(hjust = 0.5))
```

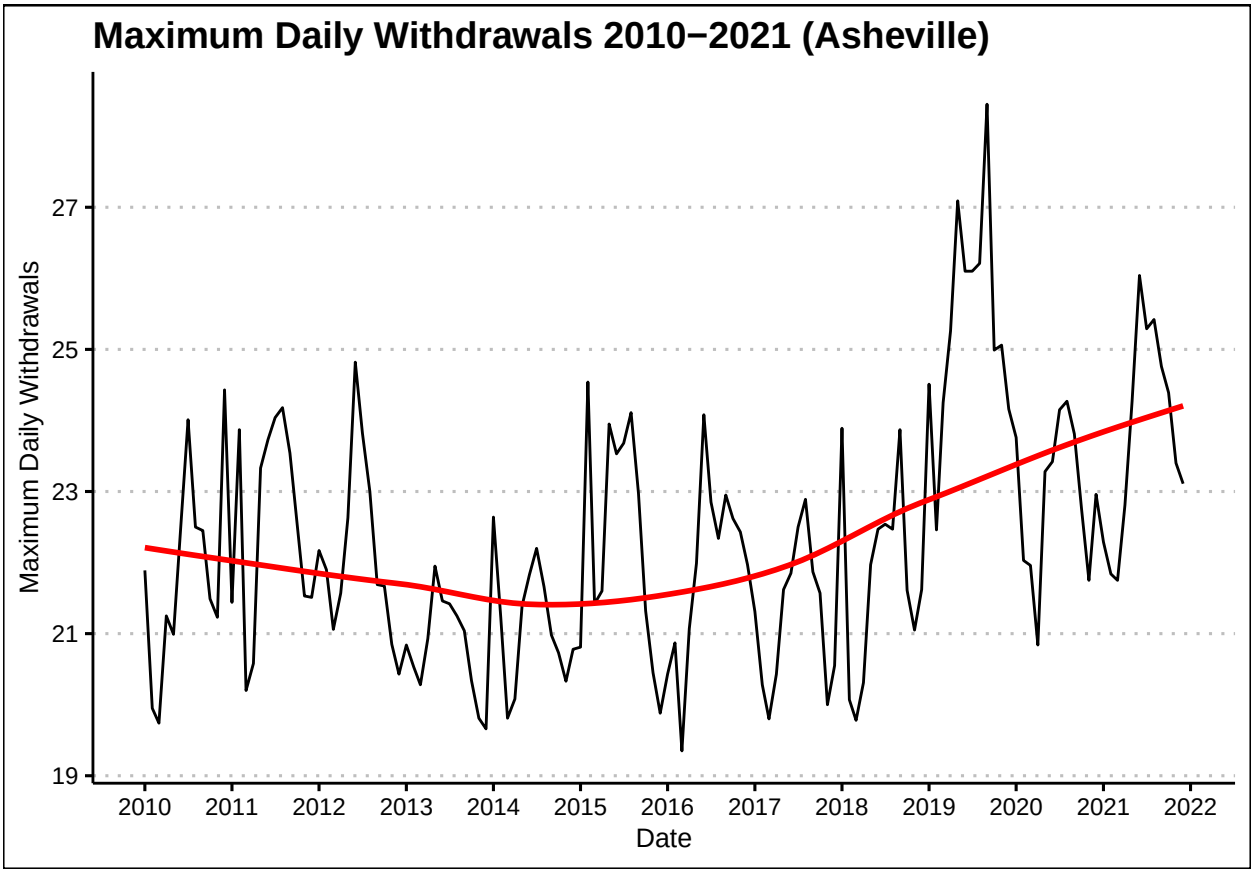


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   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
ash_years <- 2010:2021
ash_pwsid <- rep("01-11-010", length(ash_years))

ash_df <- map2(ash_pwsid, ash_years, scraping_data)
ash_df <- bind_rows(ash_df)

ggplot(ash_df, aes(x = df_date, y = df_maximum_day_use)) + geom_line() + geom_smooth(method = "loess",
    se = FALSE, color = "red") + labs(x = "Date", y = "Maximum Daily Withdrawals",
    color = "Town") + theme_clean() + scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
    ggtitle("Maximum Daily Withdrawals 2010-2021 (Asheville)")
```

```
## `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, there is an upward trend in Asheville's water usage.