Assignment 10: Data Scraping

Jonathan Joyner

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)
getwd()
```

```
## [1] "C:/Users/jbjoy/OneDrive/Documents/Grad School/Fall 2023/Environ 872/EDE_Fall2023"
```

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

```
## {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 <- DurhamLWSP %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
Water_System_Name
```

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

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

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

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

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

```
MaxDayUse <- DurhamLWSP %>%
  html_nodes("th~ td+ td") %>%
  html_text()
MaxDayUse
```

```
##  [1] "36.1000" "43.4200" "52.4900" "30.5000" "42.5900" "34.8800" "39.9100"
##  [8] "43.3200" "32.5300" "34.6600" "41.8000" "37.5300"
```

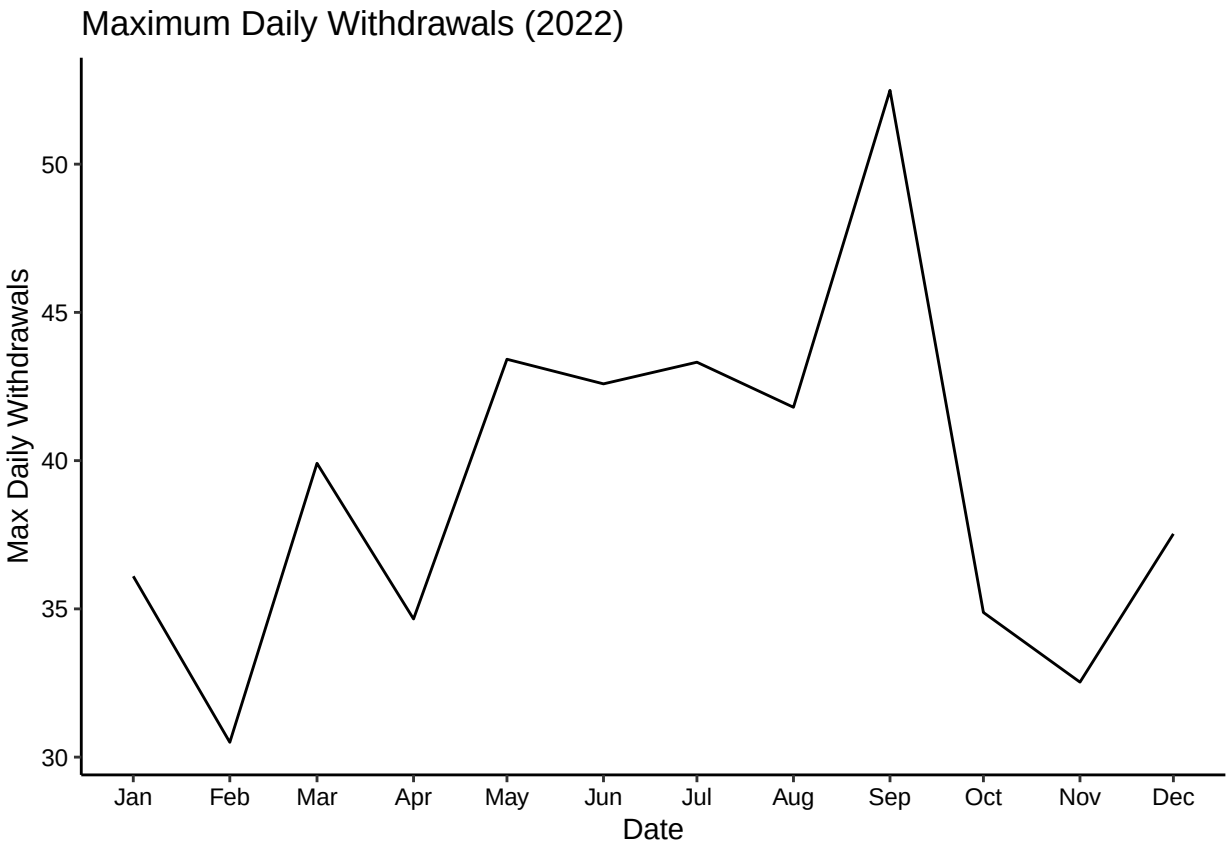
4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   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

```
#4 AI assisted for date object transformation code
custom_month_order <- c("JAN","MAY","SEP",
                        "FEB","JUN","OCT",
                        "MAR","JUL","NOV",
                        "APR","AUG","DEC")

Durham_df <- data.frame("Water_System_Name" = as.character(Water_System_Name),
                        "PWSID" = as.character(DurhamPWSID),
                        "Ownership" = as.character(Ownership),
                        "Max_Day_Use" = as.numeric(MaxDayUse),
                        "Month" = custom_month_order,
                        "Year" = rep(2022))
Durham_df$Date <-
  as.Date(paste(Durham_df$Year, Durham_df$Month, "01", sep = "-"),
                format = "%Y-%b-%d")
#5 AI assisted
ggplot(Durham_df, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  labs(title = "Maximum Daily Withdrawals (2022)",
       x = "Date",
       y = "Max Daily Withdrawals") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month")
```



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.
scrape.it <- function(the_year,the_facility){

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

the_facility_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_Max_Day_Use_tag <- "th~ td+ td"
the_Ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"

the_facility_name <- the_website %>% html_nodes(the_facility_name_tag) %>% html_text()
the_PWSID <- the_website %>% html_nodes(the_PWSID_tag) %>% html_text()
the_Max_Day_Use <- the_website %>% html_nodes(the_Max_Day_Use_tag) %>% html_text()
the_Ownership <- the_website %>% html_nodes(the_Ownership_tag) %>% html_text()

NCDIV_water_scrape <- data.frame(
                        "Water_System_Name" = as.character(the_facility_name),
                        "PWSID" = as.character(the_PWSID),
                        "Ownership" = as.character(the_Ownership),
```

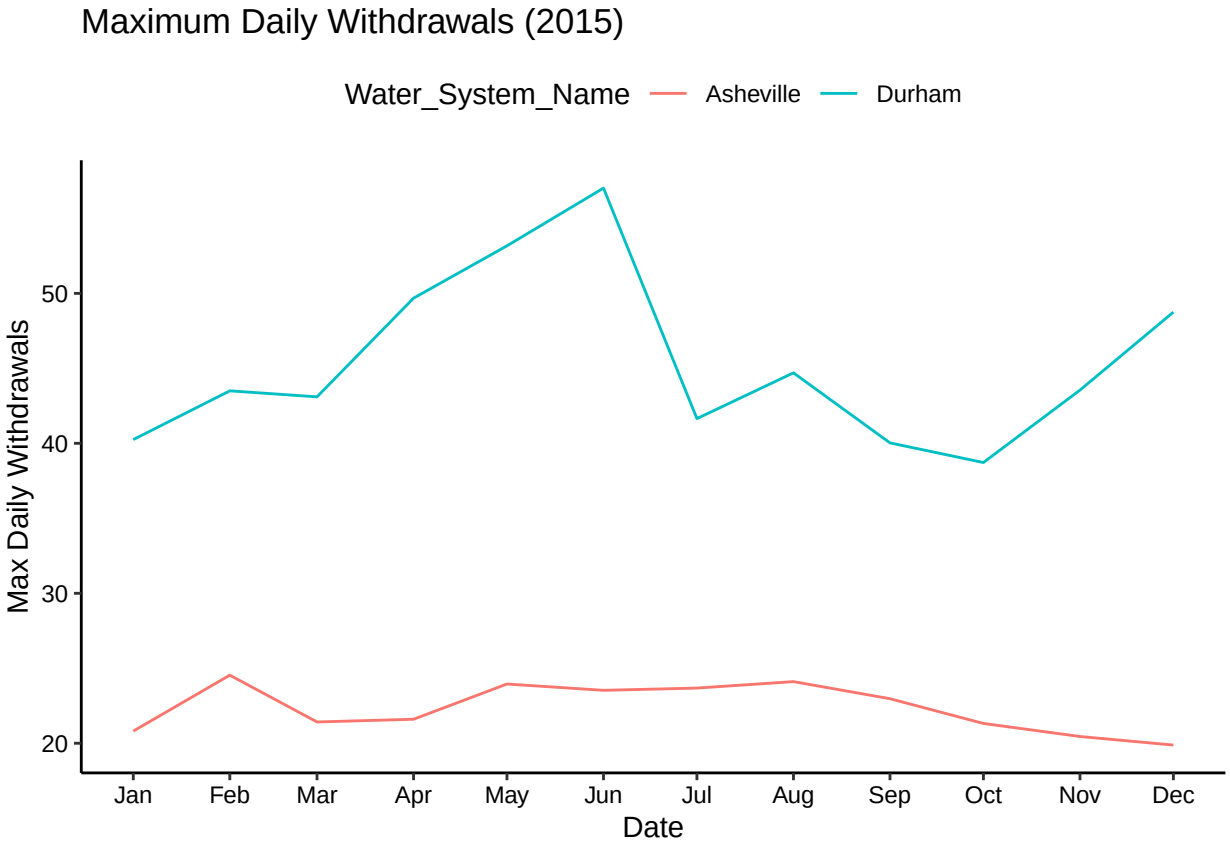
```
                        "Max_Day_Use" = as.numeric(the_Max_Day_Use),
                        "Month" = custom_month_order,
                        "Year" = as.numeric(the_year))
NCDIV_water_scrape$Date <-
  as.Date(paste(NCDIV_water_scrape$Year, NCDIV_water_scrape$Month, "01", sep = "-"),
                format = "%Y-%b-%d")
return(NCDIV_water_scrape)
}
```

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_2015 <- scrape.it(2015,"03-32-010")
view(Durham_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 AI Assisted
Asheville_2015 <- scrape.it(2015,"01-11-010")
view(Asheville_2015)
combined_data <- rbind(Durham_2015, Asheville_2015)
ggplot(combined_data, aes(x = Date, y = Max_Day_Use, color = Water_System_Name)) +
  geom_line() +
  labs(title = "Maximum Daily Withdrawals (2015)",
       x = "Date",
       y = "Max Daily Withdrawals") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month")
```

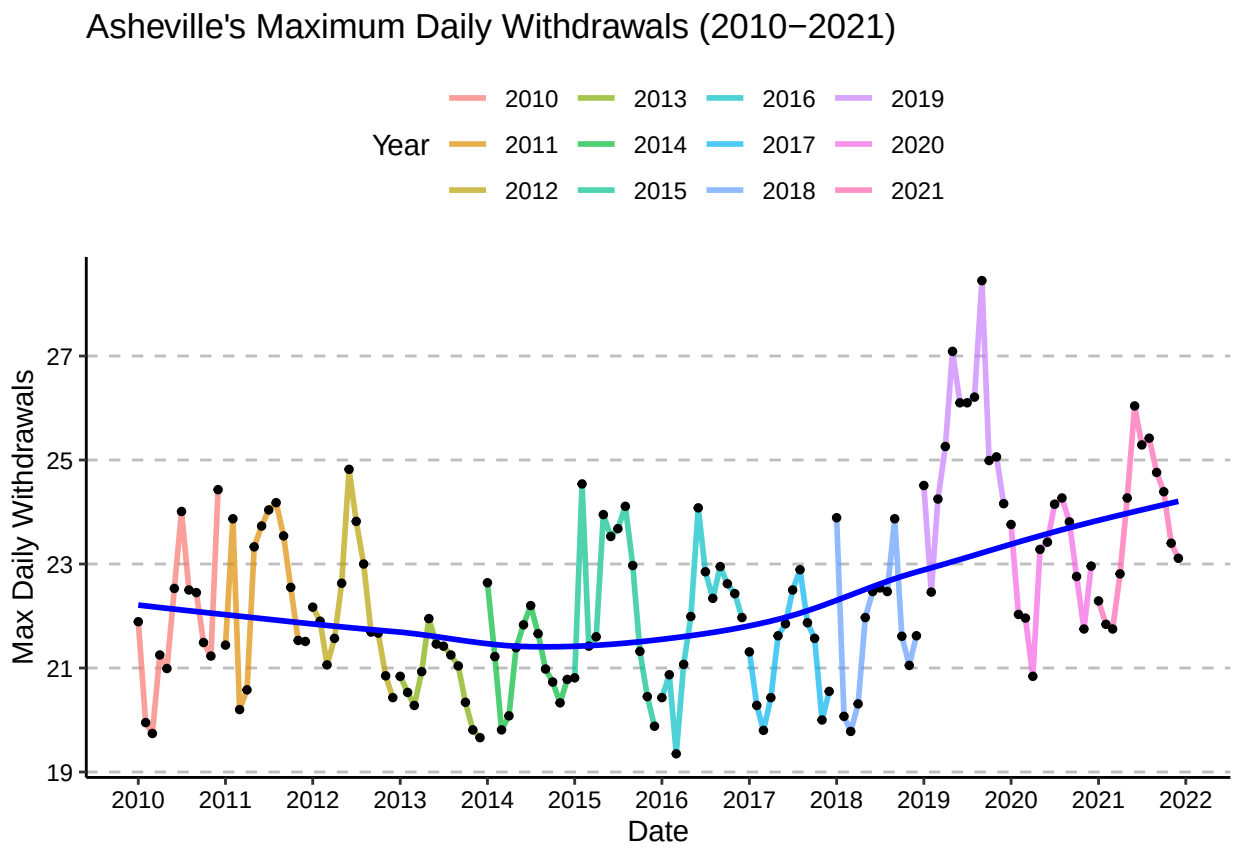


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   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 AI Assisted
Ash_years <- 2010:2021
Ash_facility_id <- "01-11-010"

combined_data_asheville <-
  map2(Ash_years, rep(Ash_facility_id, length(Ash_years)), scrape.it) %>%
  bind_rows()

combined_data_asheville$Year <- as.factor(combined_data_asheville$Year)

ggplot(combined_data_asheville, aes(x = Date, y = Max_Day_Use)) +
  geom_line(aes(color = Year), size = 1, alpha = 0.7) +
  geom_point(aes(color = Year), size = 1, shape = 19,color = 'black') +
  geom_smooth(method = 'loess', se = FALSE, color = 'blue', size = 1) +
  labs(title = "Asheville's Maximum Daily Withdrawals (2010-2021)",
       x = "Date",
       y = "Max Daily Withdrawals") +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  theme(panel.grid.major.y = element_line(color = "gray", linetype = "dashed"))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## ‘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: Ashville was on a downward trend until 2015, but then started trending up. In 2019, water withdrawl spiked noticeably.