Assignment 10: Data Scraping

Yuechen Huang

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(here)

getwd()
```

```
## [1] "/Users/shiqizheng/Desktop/ENV872/EDE_Fall2023"
```

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

```
## {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 <- Durham_LWSP_web %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>% html_text()
PWSID <- Durham_LWSP_web %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>% html_text()
Ownership <- Durham_LWSP_web %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>% html_text()
MGD <- Durham_LWSP_web %>%
  html_nodes("th~ td+ 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.)

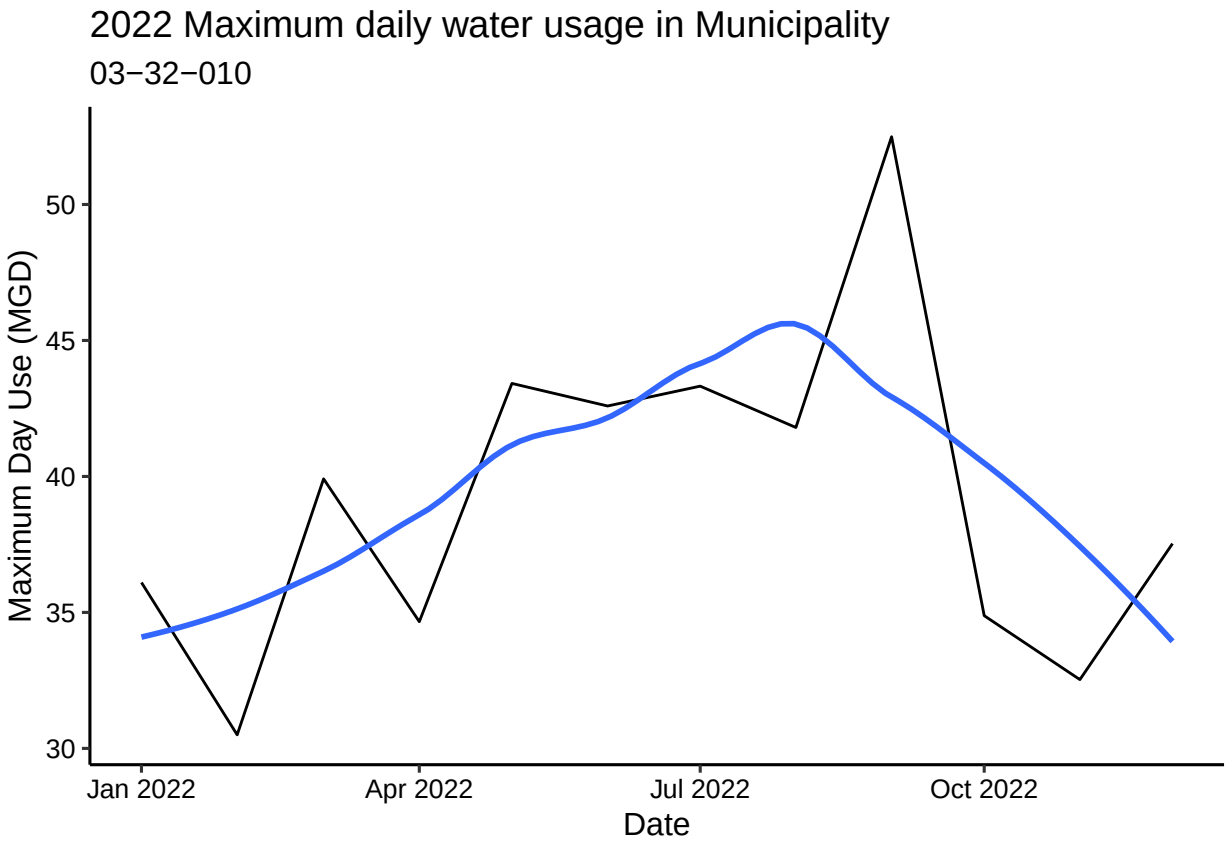
   > TIP: Use `rep()` to repeat a value when creating a dataframe.

   > 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 2022

```r
#4
Month_wrong_order <- Durham_LWSP_web %>%
  html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>% html_text()
df_raw <- data.frame("Month" = Month_wrong_order,
                     "Year" = rep(2022,12),
                     "Maximum_Day_Use" = as.numeric(MGD))

df_processed <- df_raw %>% mutate(
  "Water_System_Name" = !!water_system,
  "PWSID" = !!PWSID,
  "Ownership" = !!Ownership,
  "Date" = my(paste(Month,"-",Year)))

#5
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(mytheme)

ggplot(df_processed,aes(x=Date,y=Maximum_Day_Use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2022 Maximum daily water usage in",Ownership),
       subtitle = PWSID,
       y="Maximum Day Use (MGD)",
       x="Date")
```

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

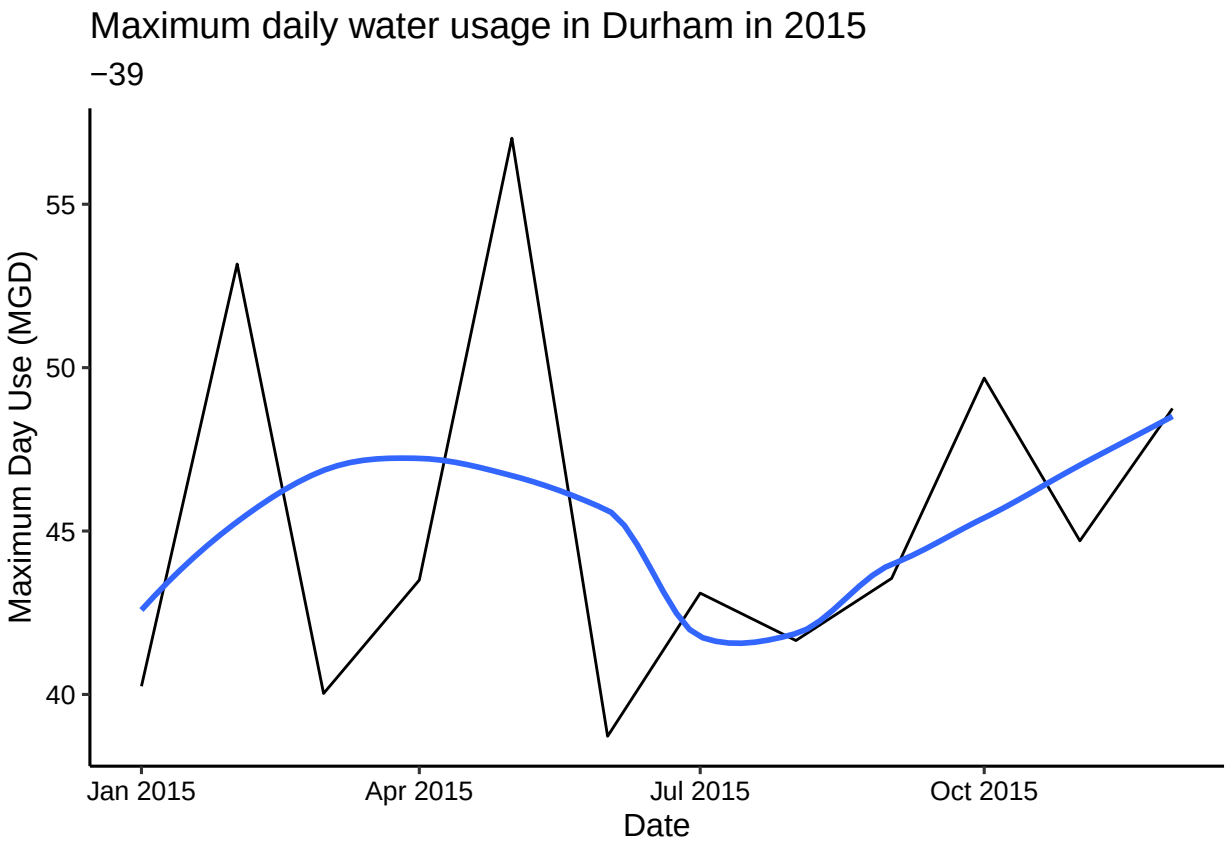
```
#6.
Scrape.fun <- function(the_PWSID,the_year){
  the_website <- read_html(
    paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
           the_PWSID, '&year=', the_year))

  water_system <- the_website %>%
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>% html_text()
  PWSID <- the_website %>%
    html_nodes("td tr:nth-child(1) td:nth-child(5)") %>% html_text()
  Ownership <- the_website %>%
    html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>% html_text()
  MGD <- the_website %>%
    html_nodes("th~ td+ td") %>% html_text()

  df_raw <- data.frame("Month" = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),
                       "Year" = rep(the_year,12),
                       "Maximum_Day_Use" = as.numeric(MGD))

  df_processed <- df_raw %>% mutate(
```

```
    "Water_System_Name" = !!water_system,
    "PWSID" = !!the_PWSID,
    "Ownership" = !!Ownership,
    "Date" = my(paste(Month,"-",Year)))
  return(df_processed)
}
```

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

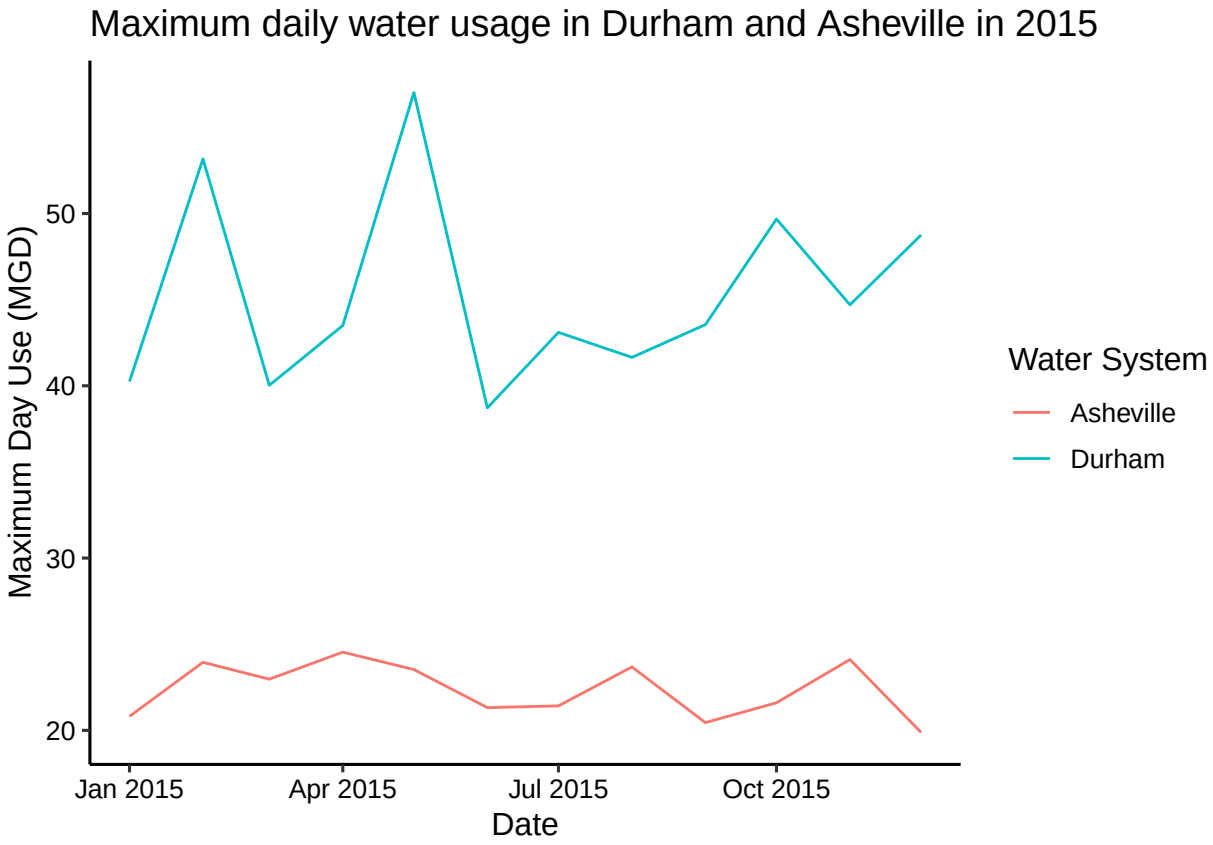
```
#7
Durham_df_2015 <- Scrape.fun("03-32-010",2015)
ggplot(Durham_df_2015,aes(x=Date,y=Maximum_Day_Use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Maximum daily water usage in Durham in 2015"),
       subtitle = 03-32-010,
       y="Maximum Day Use (MGD)",
       x="Date")
```

```
## ‘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_df_2015 <- Scrape.fun("01-11-010",2015) # can't run this code
Combined_df_2015 <- rbind(Durham_df_2015,Asheville_df_2015)
Plot_Ash_Dur_2015 <- ggplot(Combined_df_2015,aes(x=Date,y=Maximum_Day_Use, color = Water_System_Name)) +
  geom_line() +
  labs(title = paste("Maximum daily water usage in Durham and Asheville in 2015"),
       y="Maximum Day Use (MGD)",
       x="Date",
       color = "Water System")
Plot_Ash_Dur_2015
```



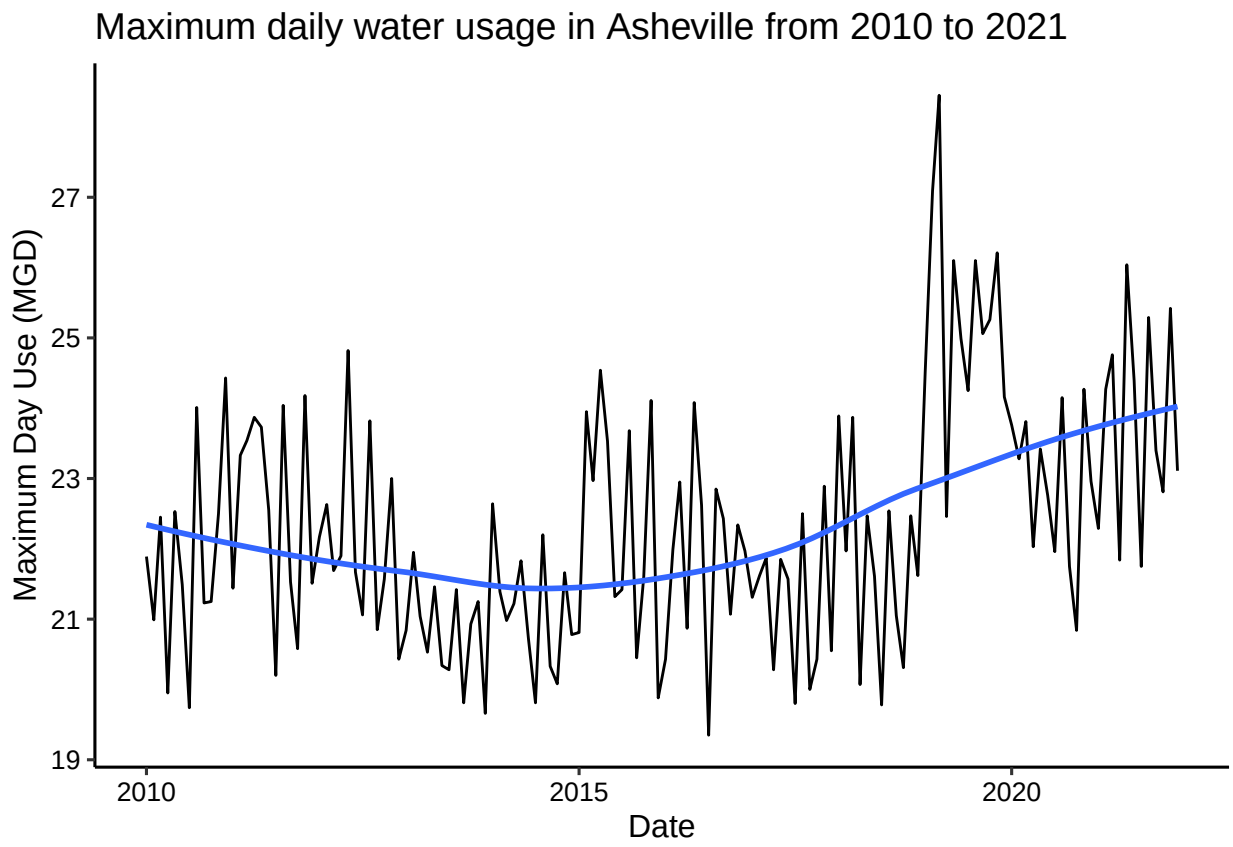
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

   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
the_years <- rep(2010:2021)
the_PWSID_Ash <- "01-11-010"

the_dfs_2010_2021 <- map2(the_PWSID_Ash,the_years,Scrape.fun) %>% bind_rows()
Plot_Ash_2010_2021 <- ggplot(the_dfs_2010_2021,aes(x=Date,y=Maximum_Day_Use)) +
```

```
  geom_line() + geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Maximum daily water usage in Asheville from 2010 to 2021"),
       y="Maximum Day Use (MGD)",
       x="Date")
Plot_Ash_2010_2021
```

```
## `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 2010 to 2015, The water usage decreases over time while from 2015 to 2021, the water usage increases.