Assignment 10: Data Scraping

Xuancheng Guo

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(lubridate);library(viridis);library(here)

library(rvest)

library(dataRetrieval)

library(tidycensus)

# 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
a10.url <-
  'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022'

a10.web <- read_html(a10.url)
a10.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.sys.name <- a10.web %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

pwsid <- a10.web %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

ownership <- a10.web %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()

max.day.use <- a10.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.)

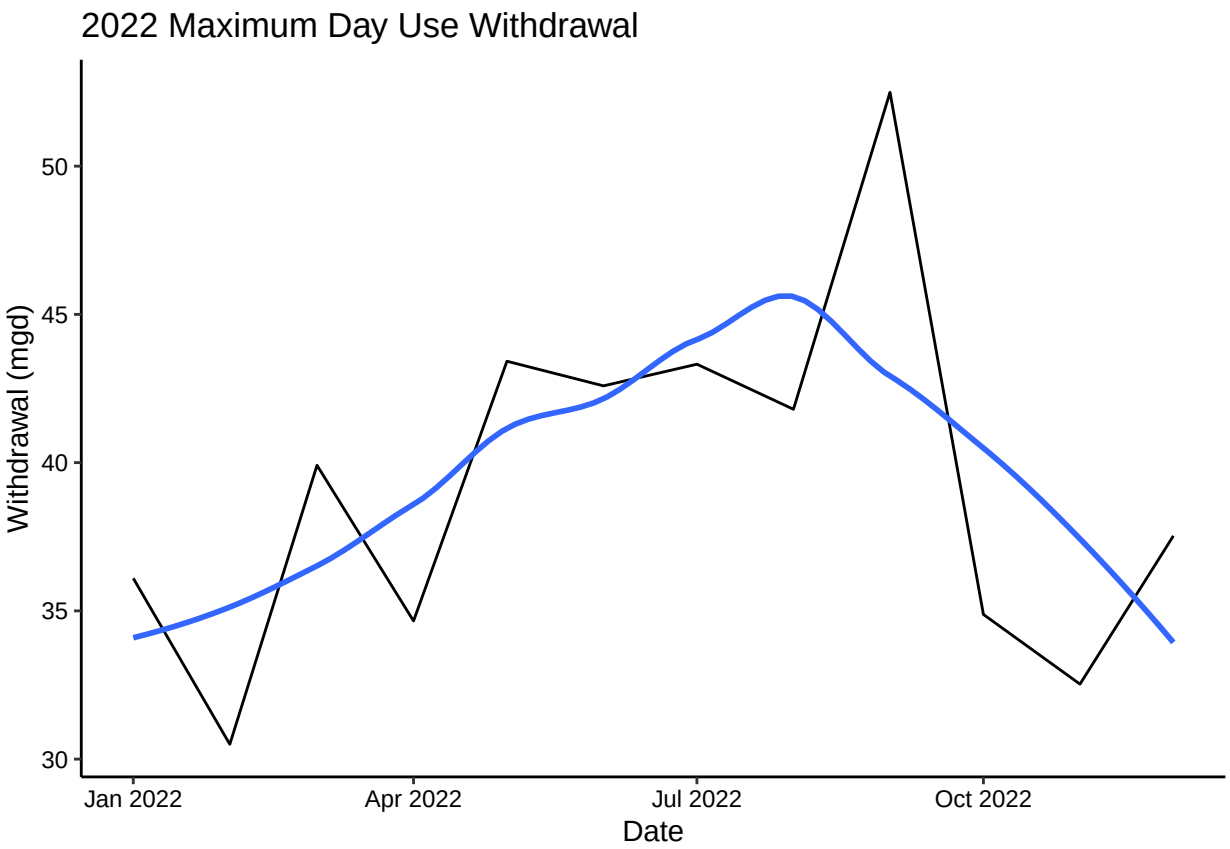
   TIP: Use `rep()` to repeat a value when creating a dataframe.

   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

```
#4
Month <-
  c("Jan", "May", "Sep", "Feb", "Jun", "Oct",
    "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

withdrawal.df <- data.frame(
  "WaterSystemName" = rep(water.sys.name, length(months)),
  "PWSID" = rep(pwsid, length(months)),
  "Ownership" = rep(ownership, length(months)),
  "MaximumDayUse" = as.numeric(max.day.use),
  "Month" = Month
)

withdrawal.df$Date <-
  as.Date(paste("2022", withdrawal.df$Month, "01", sep="-"),
          format="%Y-%b-%d")

withdrawal.df <- withdrawal.df %>%
  arrange(Date) %>%
  select(Date, WaterSystemName, PWSID, Ownership, MaximumDayUse)

#5
withdrawal.plot <-
  ggplot(withdrawal.df, aes(x = Date, y = MaximumDayUse)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "2022 Maximum Day Use Withdrawal",
       y = "Withdrawal (mgd)",
       x = "Date") +
  mytheme

withdrawal.plot
```

```
## `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 <- function(pwsid, year) {
  scrape.website <- read_html(
  paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=',
         pwsid, '&', 'year=', year))

  water.system.name.id <- 'div+ table tr:nth-child(1) td:nth-child(2)'
  pwsid.id <- 'td tr:nth-child(1) td:nth-child(5)'
  ownership.id <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  max.day.use.id <- 'th~ td+ td'

  water.system.name <- scrape.website %>%
    html_nodes(water.system.name.id) %>%
    html_text()

  pwsid <- scrape.website %>%
    html_nodes(pwsid.id) %>%
    html_text()

  ownership <- scrape.website %>%
    html_nodes(ownership.id) %>%
```

```
    html_text()

  max.day.use <- scrape.website %>%
    html_nodes(max.day.use.id) %>%
    html_text()

  Month <-
  c("Jan", "May", "Sep", "Feb", "Jun", "Oct",
    "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

  withdrawal.df <- data.frame(
    "WaterSystemName" = rep(water.system.name, length(months)),
    "PWSID" = rep(pwsid, length(months)),
    "Ownership" = rep(ownership, length(months)),
    "MaximumDayUse" = as.numeric(max.day.use),
    "Month" = Month
  )

  withdrawal.df$Date <-
    as.Date(paste(year, withdrawal.df$Month, "01", sep="-"),
            format="%Y-%b-%d")

  withdrawal.df <- withdrawal.df %>%
    arrange(Date) %>%
    select(Date, WaterSystemName, PWSID, Ownership, MaximumDayUse)


  return(withdrawal.df)
}
```

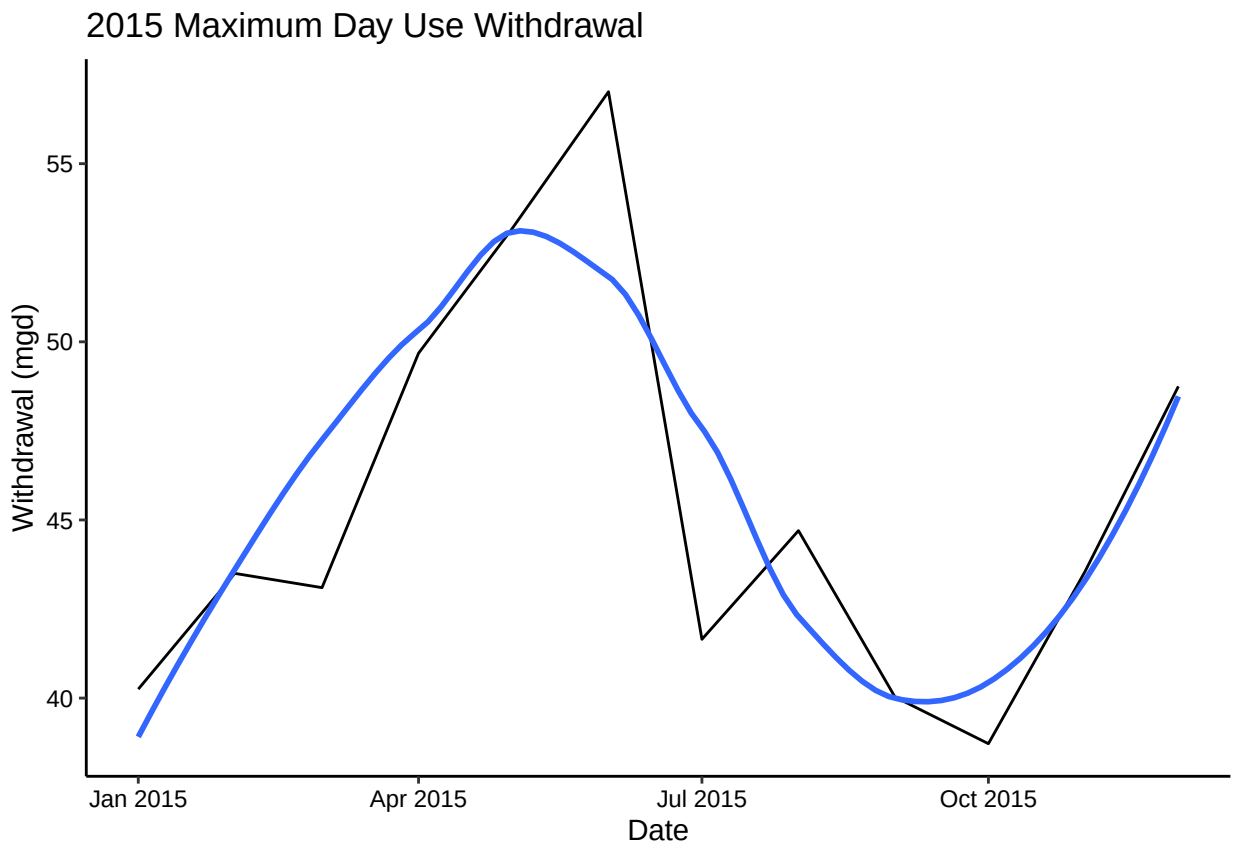
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID='03-32-010') for each month in 2015

```
#7
withdrawal.2015 <-
  scrape('03-32-010', 2015)
withdrawal.2015
```

```
##          Date WaterSystemName     PWSID    Ownership MaximumDayUse
## 1  2015-01-01          Durham 03-32-010 Municipality         40.25
## 2  2015-02-01          Durham 03-32-010 Municipality         43.50
## 3  2015-03-01          Durham 03-32-010 Municipality         43.10
## 4  2015-04-01          Durham 03-32-010 Municipality         49.68
## 5  2015-05-01          Durham 03-32-010 Municipality         53.17
## 6  2015-06-01          Durham 03-32-010 Municipality         57.02
## 7  2015-07-01          Durham 03-32-010 Municipality         41.65
## 8  2015-08-01          Durham 03-32-010 Municipality         44.70
## 9  2015-09-01          Durham 03-32-010 Municipality         40.03
## 10 2015-10-01          Durham 03-32-010 Municipality         38.72
## 11 2015-11-01          Durham 03-32-010 Municipality         43.55
## 12 2015-12-01          Durham 03-32-010 Municipality         48.75
```

```
withdrawal.2015.plot <-
  ggplot(withdrawal.2015, aes(x = Date, y = MaximumDayUse)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "2015 Maximum Day Use Withdrawal",
       y = "Withdrawal (mgd)",
       x = "Date") +
  mytheme

withdrawal.2015.plot
```

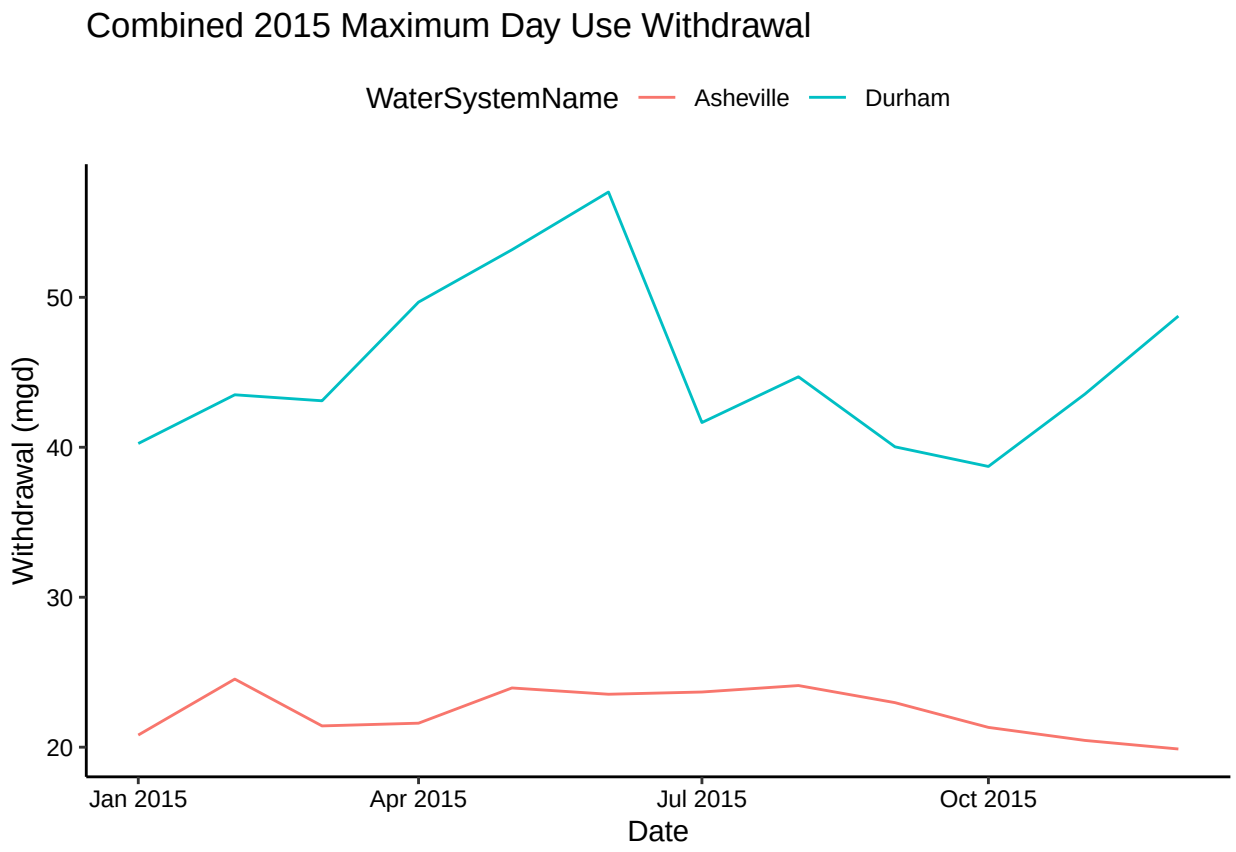
```
## `geom_smooth()` using formula = 'y ~ x'
```



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.2015 <- scrape('01-11-010', 2015)

# combined2015 <- rbind(withdrawal.2015, asheville.2015)
combined2015 <- rbind(asheville.2015, withdrawal.2015)

combined2015.plot <-
```

```
  ggplot(combined2015, aes(x = Date, y = MaximumDayUse)) +
  geom_line(aes(color = WaterSystemName)) +
  labs(title = "Combined 2015 Maximum Day Use Withdrawal",
       y = "Withdrawal (mgd)",
       x = "Date") +
  mytheme

combined2015.plot
```



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.

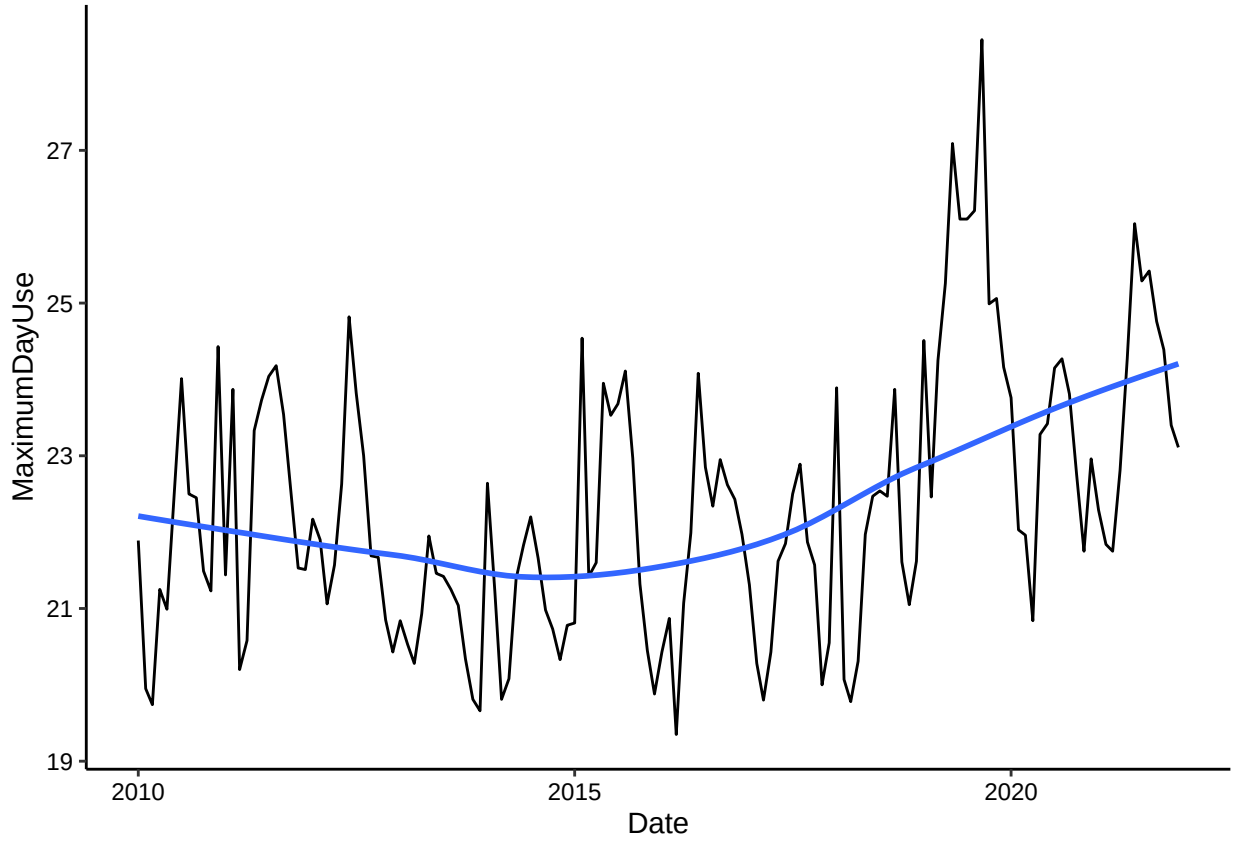
```
#9
years = rep(2010:2021)
q9.pwsid <- "01-11-010"

asheville.11years <- map(years, scrape, pwsid = q9.pwsid)
asheville.years <- bind_rows(asheville.11years)

asheville.years.plot <-
```

```
  ggplot(asheville.years, aes(x = Date, y = MaximumDayUse)) +
  geom_line() +
  geom_smooth(method = 'loess', se = FALSE)
  labs(title = "Asheville Maximum Day Use from 2010 to 2021",
       y = "Withdrawal (mgd)",
       x = "Date") +
  mytheme
```

```
## NULL
```

```
asheville.years.plot
```

```
## `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: There are two trends we can tell from the graph. From 2010 to 2015, the maximum daily usage of water declined slightly. After 2015, there is a larger upward trend for the water usage. >