Assignment 10: Data Scraping

Ruiqing Li

OVERVIEW

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

Directions

1. Rename this file `<RuiqingLi>_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(here); here()
```

```
## [1] "/Users/ruiqingli"
```

```
library(rvest)

getwd()
```

```
## [1] "/Users/ruiqingli"
```

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

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

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

```
#3
water.system.name <- LWSP.Webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water.system.name
```

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

```
PWSID <- LWSP.Webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
PWSID
```

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

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

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

```
max.withdrawals.mgd  <- LWSP.Webpage %>%
  html_nodes('th~ td+ td') %>%
  html_text()
max.withdrawals.mgd
```

```
##  [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…

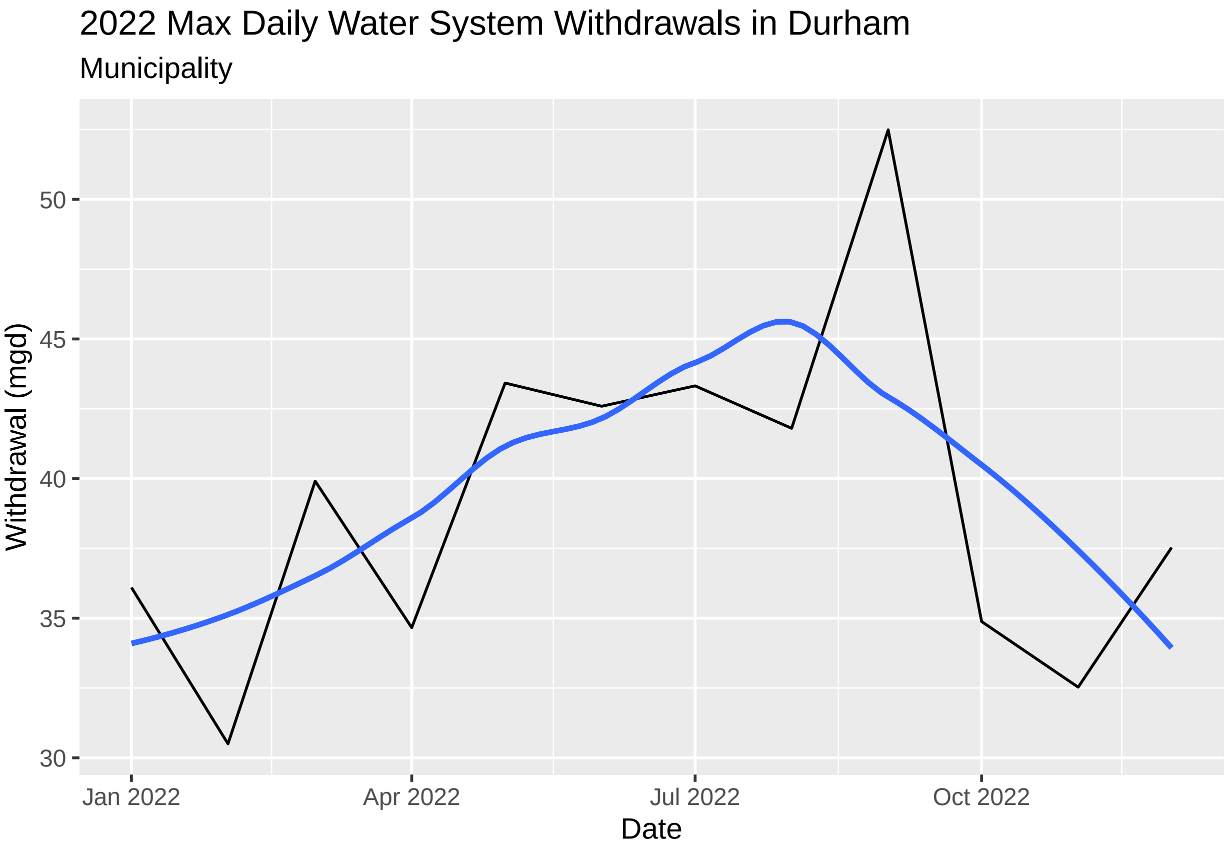
5. Create a line plot of the average daily withdrawals across the months for 2022

```
df_LWSP_withdrawals <- data.frame(
"Month" = month(c(1,5,9,2,6,10,3,7,11,4,8,12)),
"Year" = rep(2022,12),
"max.withdrawals.mgd" = as.numeric(max.withdrawals.mgd),
"water.system.name" = water.system.name,
"PWSID" = PWSID,
"ownership" = ownership)

df_LWSP_withdrawals <- df_LWSP_withdrawals %>%
mutate(Date = my(paste(Month,"-",Year)))

df_LWSP_withdrawals
```

```
##    Month Year max.withdrawals.mgd water.system.name     PWSID    ownership
## 1      1 2022               36.10            Durham 03-32-010 Municipality
## 2      5 2022               43.42            Durham 03-32-010 Municipality
## 3      9 2022               52.49            Durham 03-32-010 Municipality
## 4      2 2022               30.50            Durham 03-32-010 Municipality
## 5      6 2022               42.59            Durham 03-32-010 Municipality
## 6     10 2022               34.88            Durham 03-32-010 Municipality
## 7      3 2022               39.91            Durham 03-32-010 Municipality
## 8      7 2022               43.32            Durham 03-32-010 Municipality
## 9     11 2022               32.53            Durham 03-32-010 Municipality
## 10     4 2022               34.66            Durham 03-32-010 Municipality
## 11     8 2022               41.80            Durham 03-32-010 Municipality
## 12    12 2022               37.53            Durham 03-32-010 Municipality
##           Date
## 1  2022-01-01
## 2  2022-05-01
## 3  2022-09-01
## 4  2022-02-01
## 5  2022-06-01
## 6  2022-10-01
## 7  2022-03-01
## 8  2022-07-01
## 9  2022-11-01
## 10 2022-04-01
## 11 2022-08-01
## 12 2022-12-01
```

```
#5
ggplot(df_LWSP_withdrawals,aes(x=Date,y=max.withdrawals.mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2022 Max Daily Water System Withdrawals in",water.system.name),
       subtitle = ownership,
       y="Withdrawal (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**.

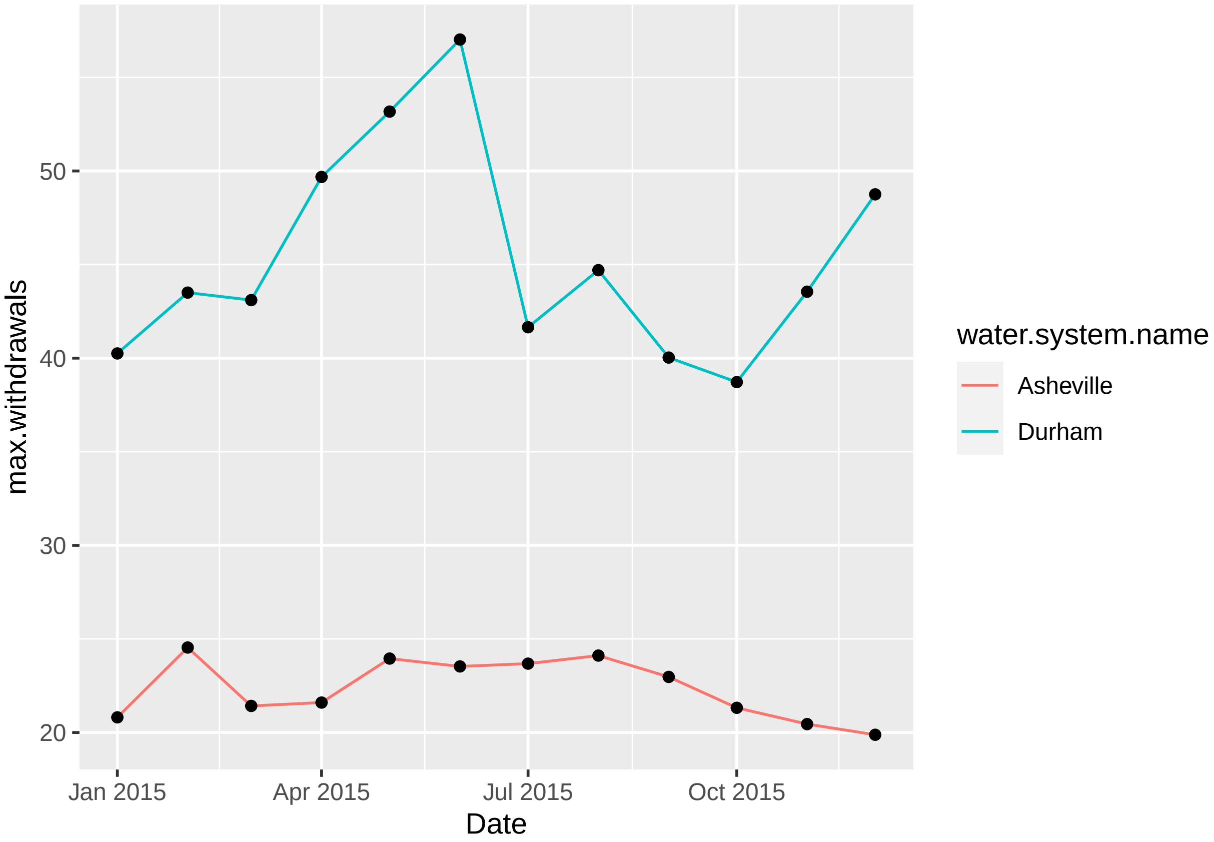
```
#6.
scrape.it <- function(the_year, the_PWSID){
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=', the_PWSID,

  the_water.system.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_ownership_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
  the_data_tag <- 'th~ td+ td'

  the_water.system.name <- the_website %>% html_nodes(the_water.system.name_tag) %>% html_text()
  the_PWSID <- the_website %>%   html_nodes( the_PWSID_tag) %>%  html_text()
  the_ownership <- the_website %>% html_nodes(the_ownership_tag) %>% html_text()
  max_withdrawals <- the_website %>% html_nodes(the_data_tag) %>% html_text()

df_LWSP_withdrawals <- tibble(
"Month" = month(c(1,5,9,2,6,10,3,7,11,4,8,12)),
"max.withdrawals" = as.numeric(max_withdrawals),
"water.system.name" = the_water.system.name,
"PWSID" = the_PWSID,
"ownership" = the_ownership)

df_LWSP_withdrawals <- df_LWSP_withdrawals %>%
  mutate(Year = the_year)%>%
  mutate(Date = my(paste0(Month,"-",Year)))

  return(df_LWSP_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
Durham_df_new <- scrape.it(2015,'03-32-010')
view(Durham_df_new)
```

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_the_df_new <- scrape.it(2015,'01-11-010')
view(Asheville_the_df_new)

combined<-
  rbind(Durham_df_new,Asheville_the_df_new)

ggplot(combined,aes(x=Date,y=max.withdrawals)) +
  geom_line(aes(color = water.system.name )) +
  geom_point()
```



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').

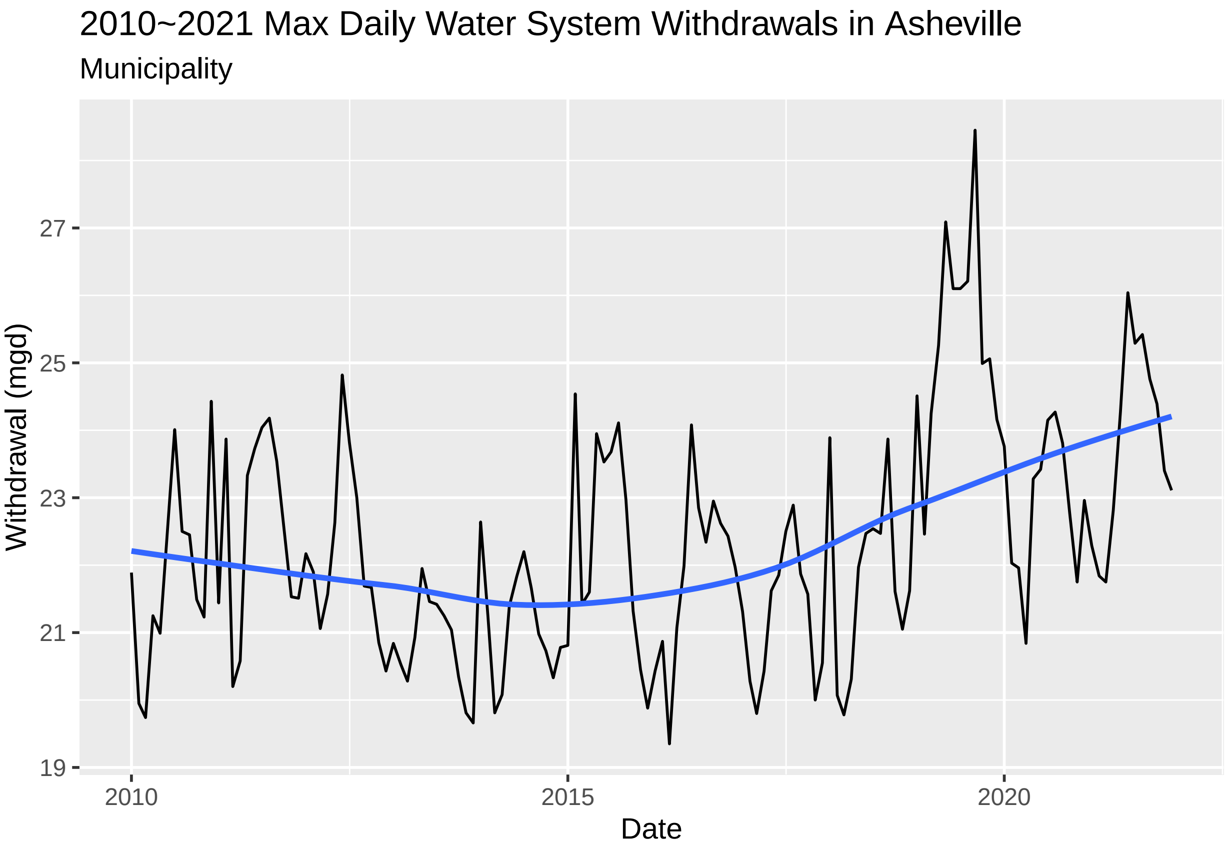
   TIP: See Section 3.2 in the "09_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```
#9
the_year = rep(2010:2021)
the_PWSID = '01-11-010'

Asheville_the_dfs <- lapply(X = the_year,
                  FUN = scrape.it,
                  the_PWSID=the_PWSID)

Asheville_the_dfs <- map(the_year,scrape.it,the_PWSID=the_PWSID)

Asheville_the_df<- bind_rows(Asheville_the_dfs)

Asheville_the_df
```

```
## # A tibble: 144 x 7
##    Month max.withdrawals water.system.name PWSID     ownership   Year Date
##    <dbl>           <dbl> <chr>             <chr>     <chr>      <int> <date>
##  1     1            21.9 Asheville         01-11-010 Municipal~  2010 2010-01-01
##  2     5            21.0 Asheville         01-11-010 Municipal~  2010 2010-05-01
##  3     9            22.4 Asheville         01-11-010 Municipal~  2010 2010-09-01
##  4     2            20.0 Asheville         01-11-010 Municipal~  2010 2010-02-01
##  5     6            22.5 Asheville         01-11-010 Municipal~  2010 2010-06-01
##  6    10            21.5 Asheville         01-11-010 Municipal~  2010 2010-10-01
##  7     3            19.7 Asheville         01-11-010 Municipal~  2010 2010-03-01
##  8     7            24.0 Asheville         01-11-010 Municipal~  2010 2010-07-01
##  9    11            21.2 Asheville         01-11-010 Municipal~  2010 2010-11-01
## 10     4            21.2 Asheville         01-11-010 Municipal~  2010 2010-04-01
## # ... with 134 more rows
```

```
ggplot(Asheville_the_df,aes(x=Date,y=max.withdrawals)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "2010~2021 Max Daily Water System Withdrawals in Asheville",
       subtitle = ownership,
       y="Withdrawal (mgd)",
       x="Date")
```

```
## ‘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?