Assignment 10: Data Scraping

David Robinson

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

#install.packages("tidyverse")
library(tidyverse)
#install.packages("rvest")
library(rvest)
#install.packages("here")
library(here); here()
```

```
## [1] "C:/Users/dhr20/OneDrive - Duke University/x - Misc/Desktop/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

#Indicate website as URL to be scraped
webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
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)".

```
#3

#Scrape values and assign to four separate variables
the_water_system_name <- webpage %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()

the_PWSID <- webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()

the_ownership <- webpage %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()

the_MGD <- webpage %>%
  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.)

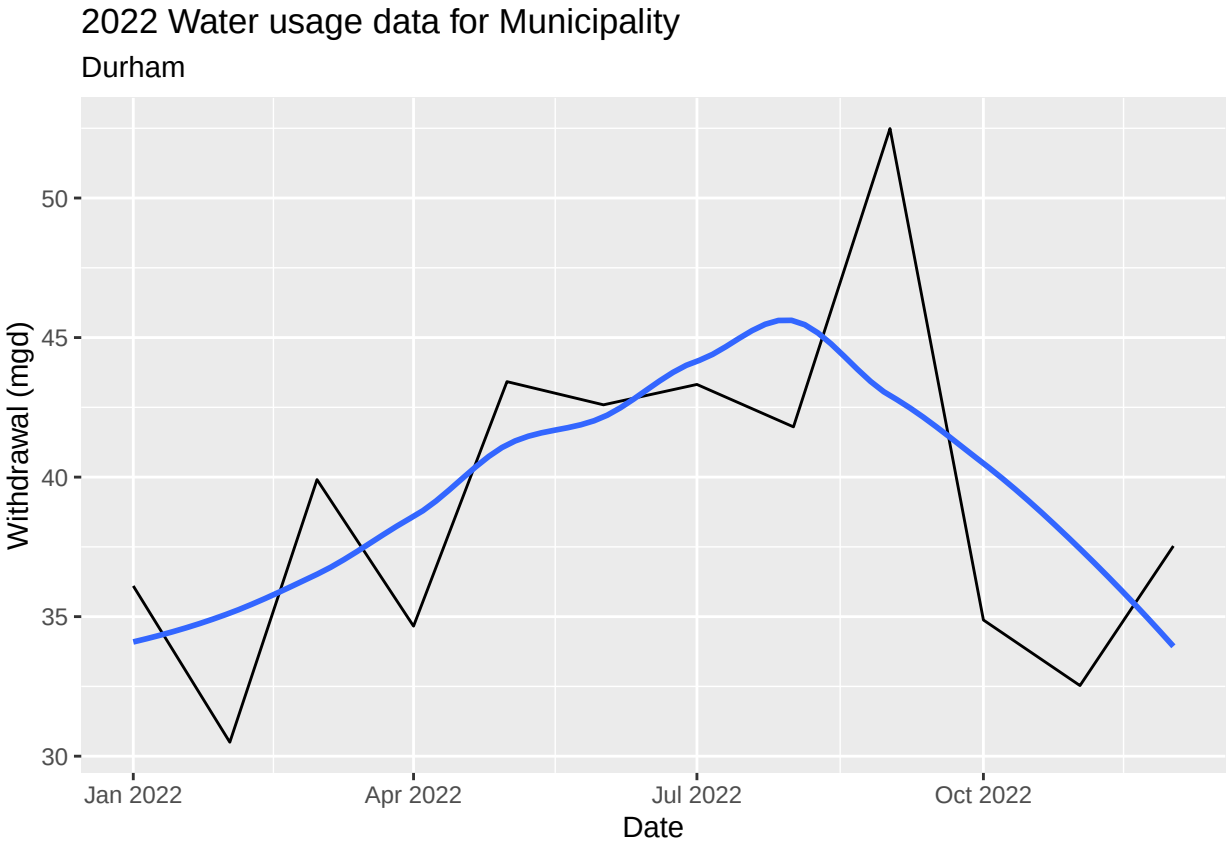
TIP: Use `rep()` to repeat a value when creating a dataframe.

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

#Create list of months in order the data was scraped
the_months = c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar", "Jul", "Nov", "Apr", "Aug", "Dec")

#Convert scraped data into a dataframe
df_withdrawals <- data.frame("Month" = the_months,
                             "Year" = rep(2022,12),
                             "Avg_Withdrawals_mgd" = as.numeric(the_MGD)) %>%
  mutate(Water_System_Name = !!the_water_system_name,
         PWSID = !!the_PWSID,
         Ownership = !!the_ownership,
         Date = my(paste(Month,"-",Year)))

#5

#Create line plot
ggplot(df_withdrawals,aes(x=Date,y=Avg_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2022 Water usage data for",the_ownership),
       subtitle = the_water_system_name,
       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.

#Construct function
scrape.it <- function(the_PWSID_var, the_year_var){
  the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report'
'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022'

the_scrape_url <- paste0(the_base_url, '.php?pwsid=', the_PWSID_var, '&year=', the_year_var)
print(the_scrape_url)

#Retrieve the website contents
webpage_var <- read_html(the_scrape_url)
webpage_var

#Set the element address variables (determined in the previous step)
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_MGD_tag <- 'th~ td+ td'

#Scrape the data items
the_water_system_name_var <- webpage_var %>%
  html_nodes(the_water_system_name_tag) %>%
  html_text()

the_PWSID_var <- webpage_var %>%
  html_nodes(the_PWSID_tag) %>%
  html_text()

the_ownership_var <- webpage_var %>%
  html_nodes(the_ownership_tag) %>%
  html_text()

the_MGD_var <- webpage_var %>%
  html_nodes(the_MGD_tag) %>%
  html_text()

#Construct a dataframe from the scraped data
df_withdrawals_var <- data.frame("Month" = the_months,
                                 "Year" = rep(the_year_var, 12),
                                 "Avg_Withdrawals_mgd" = as.numeric(the_MGD_var)) %>%
  mutate(Water_System_Name = !!the_water_system_name_var,
         PWSID = !!the_PWSID_var,
         Ownership = !!the_ownership_var,
         Date = my(paste(Month,"-",Year)))

return(df_withdrawals_var)
}
```

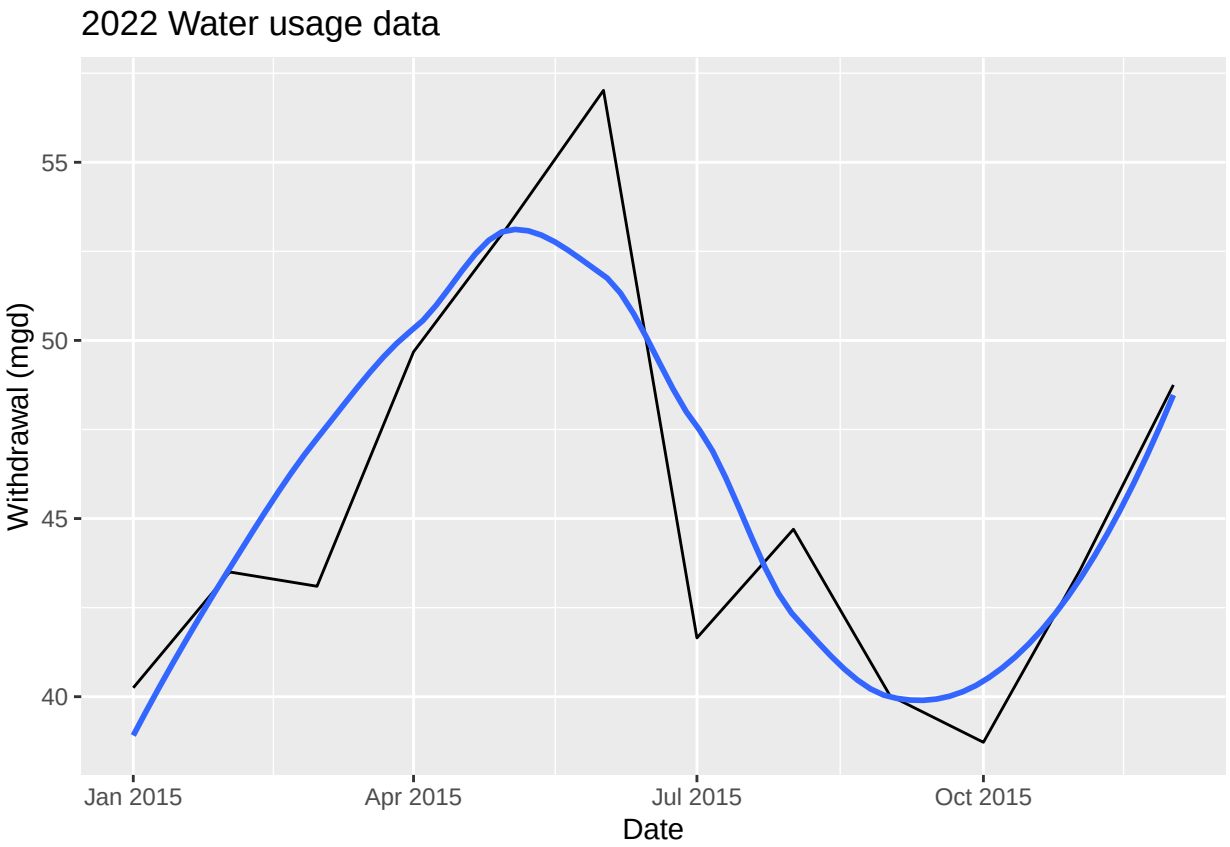
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
#7

#Use function to extract
df_7 <- scrape.it('03-32-010',2015)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2015"
```

```
#Plot
ggplot(df_7,aes(x=Date,y=Avg_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2022 Water usage data"),
       y="Withdrawal (mgd)",
       x="Date")
```

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

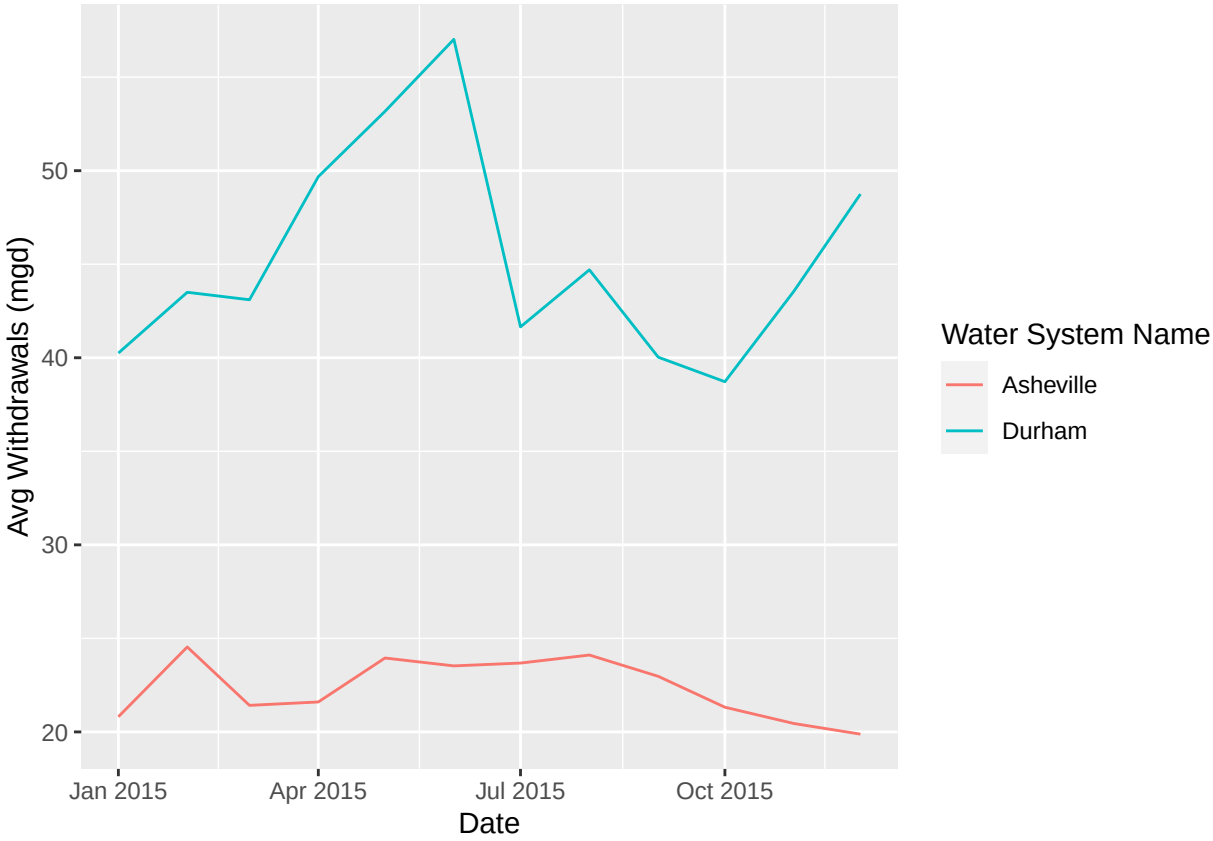


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
#8

#Use function to extract
df_8 <- scrape.it('01-11-010',2015)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2015"
```

```
#Combine data frames
df_combined <- rbind(df_8, df_7)

#Plot
ggplot(data = df_combined,
       aes(x=Date, y=Avg_Withdrawals_mgd, color = Water_System_Name)) +
  geom_line() +
  labs(y = "Avg Withdrawals (mgd)", color = "Water System Name")
```



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.

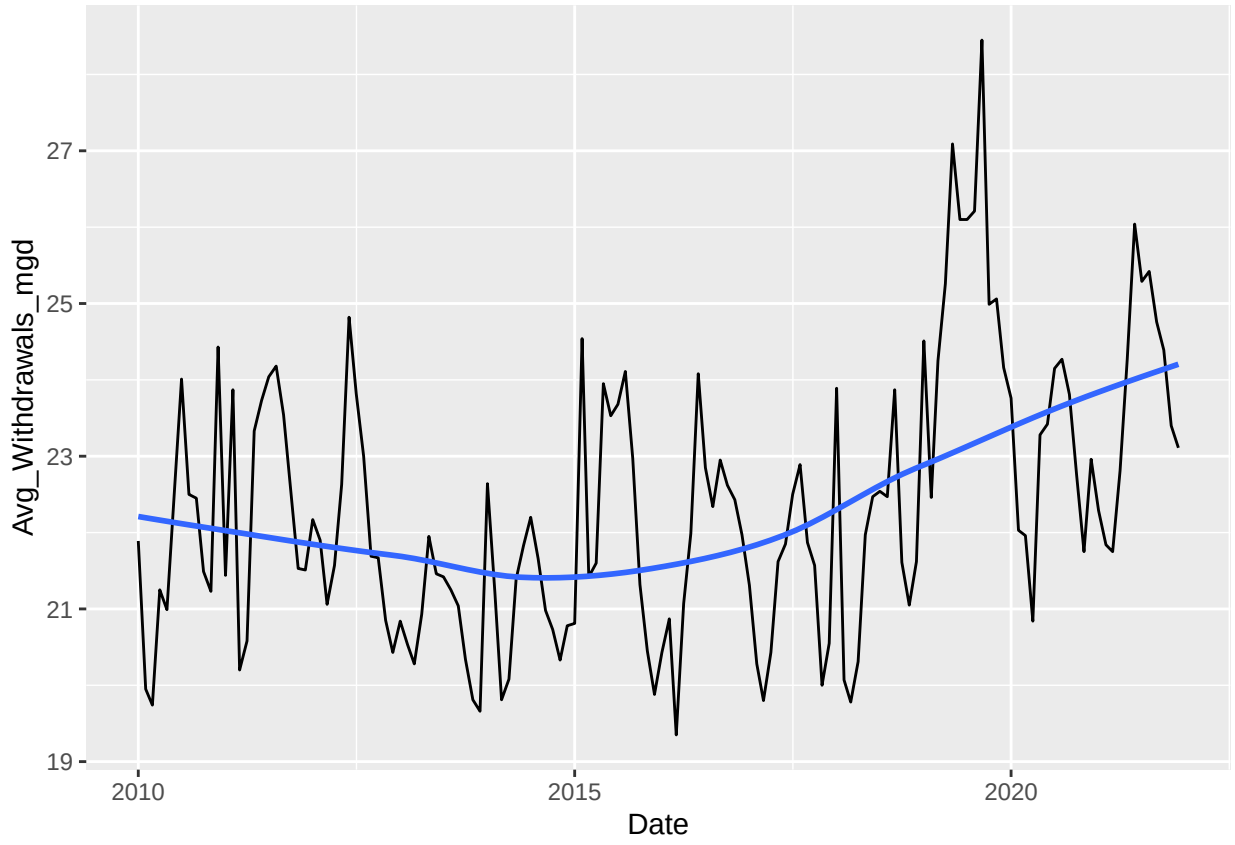
```
#9

#Create list of years
list_of_years <- c(2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021)
list_of_PWSIDs <- rep('01-11-010',12)

#Iteratively run function and combine data frames
dfs_2021 <- map2(list_of_PWSIDs, list_of_years, scrape.it) %>%
  bind_rows()
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2010"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2011"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2012"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2013"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2014"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2015"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2016"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2017"
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2018"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2019"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2020"
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=01-11-010&year=2021"
```

```
#Plot
ggplot(data = dfs_2021,
       aes(x=Date, y=Avg_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE)
```

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




```
  labs(y = "Avg Withdrawals (mgd)")
```

```
## $y
## [1] "Avg Withdrawals (mgd)"
##
## attr(,"class")
## [1] "labels"
```

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 a gradual trend of increased water usage over time (the trendline moves up and to the right).