Assignment 10: Data Scraping

Annabelle White

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(lubridate)
getwd()
```

```
## [1] "C:/Users/ardwh/OneDrive/Documents/NSOE/env872/EDA-Spring2023/Assignments"
```

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
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), with the first value being "27.6400".

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

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

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

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

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

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

```
max.withdrawals.mgd <- 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"
```

```
# The 27.6400 number cited does not appear on this page
```

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

```
#4
months <- webpage %>%
  html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
  html_text()
months
```

```
##  [1] "Jan" "May" "Sep" "Feb" "Jun" "Oct" "Mar" "Jul" "Nov" "Apr" "Aug" "Dec"
```

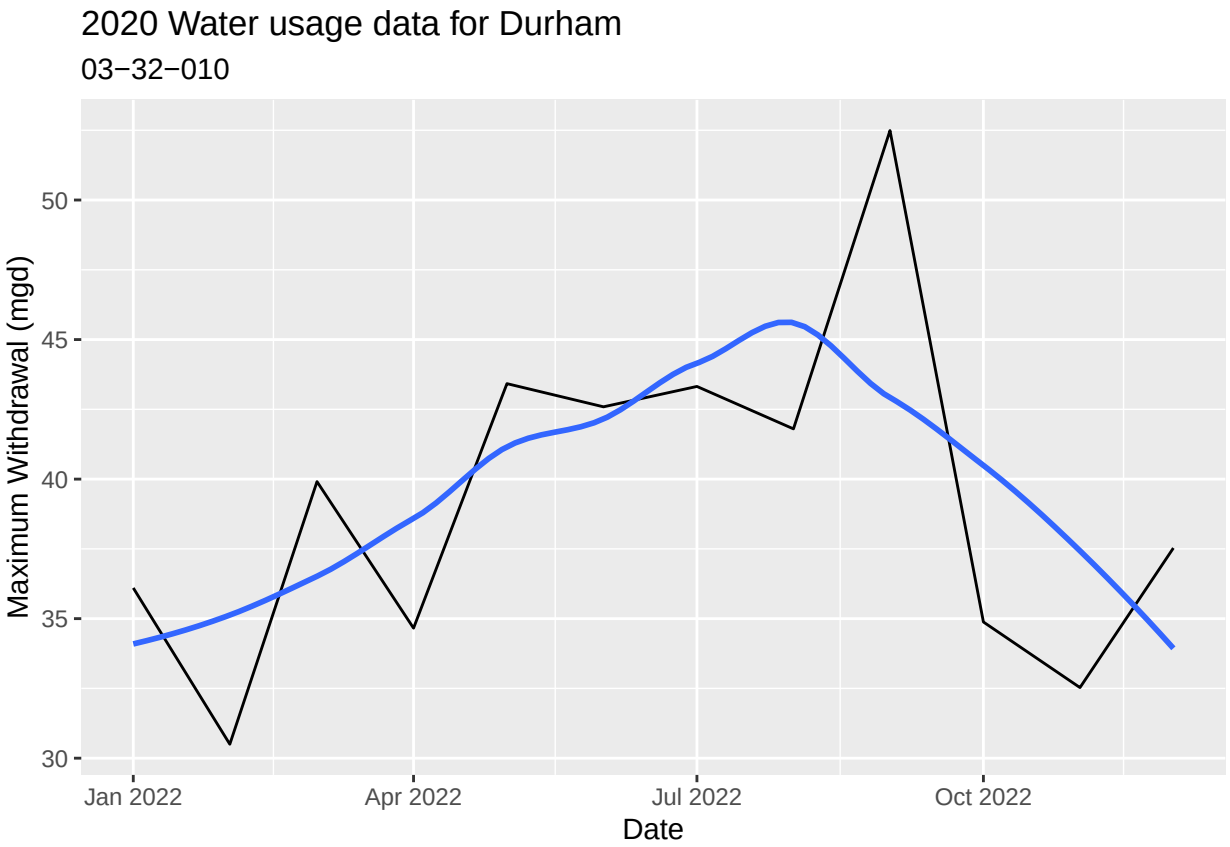
```
#Create a dataframe of withdrawals
df_withdrawals <- data.frame("Month" = months,
                             "Year" = 2022,
                             "WaterSystem" = water.system.name,
                             "PWSID" = PWSID,
                             "Ownership" = ownership,
                             "Max_Withdrawals_mgd" = as.numeric(max.withdrawals.mgd))

#Modify the dataframe to include the date (as date object)
df_withdrawals <- df_withdrawals %>%
  mutate(Date = my(paste(Month,"-",Year)))

#5

#Plot
ggplot(df_withdrawals,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2020 Water usage data for",water.system.name),
       subtitle = PWSID,
       y="Maximum 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) {

  #Retrieve the website contents
  the_website <- read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php',
                                  '?pwsid=', the_pwsid, '&year=', the_year))

  #Set the element address variables (determined in the previous step)
  system_tag <- "div+ table tr:nth-child(1) td:nth-child(2)"
  PWSID_tag <- "td tr:nth-child(1) td:nth-child(5)"
  ownership_tag <- "div+ table tr:nth-child(2) td:nth-child(4)"
  max_withdrawals_tag <- "th~ td+ td"

  #Scrape the data items
  the_system_name <- the_website %>% html_nodes(system_tag) %>% html_text()
  the_PWSID <- the_website %>%   html_nodes(PWSID_tag) %>%  html_text()
  ownership <- the_website %>% html_nodes(ownership_tag) %>% html_text()
  max_withdrawals <- the_website %>% html_nodes(max_withdrawals_tag) %>% html_text()

  #Construct a dataframe from the scraped data
```

```
  df_withdrawals <- data.frame("Month" = c("Jan", "May", "Sep", "Feb", "Jun",
                                           "Oct", "Mar", "Jul", "Nov", "Apr",
                                           "Aug", "Dec"),
                               "Year" = the_year,
                               "Max_Withdrawals_mgd" = as.numeric(max_withdrawals)) %>%
    mutate(Water_System = !!the_system_name,
           PWSID = !!the_PWSID,
           Ownership = !!ownership,
           Date = my(paste(Month,"-",Year)))

  Sys.sleep(1) # Scraping etiquette
  return(df_withdrawals)
}

# The HTML tag for months varies on different pages
# So I instead made it a static list to keep the function consistent
```

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

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

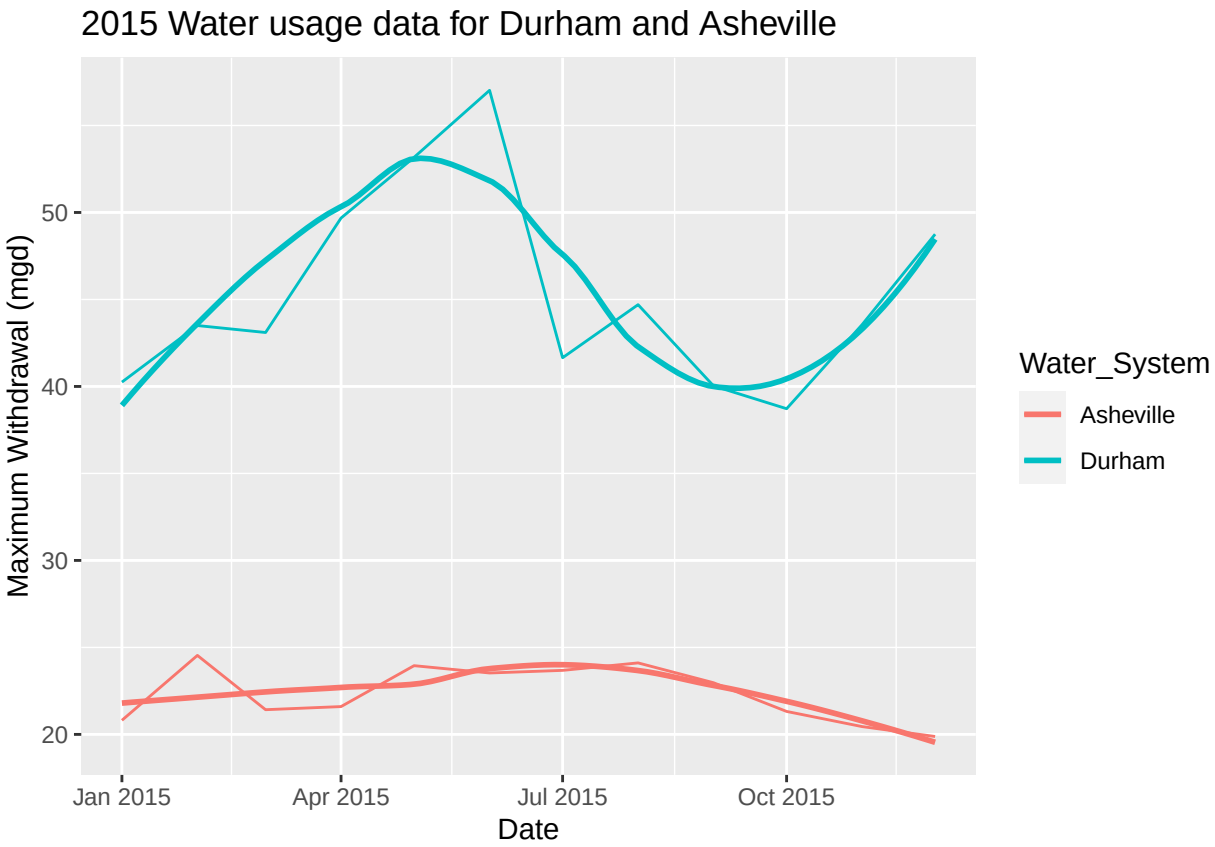
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville’s to Durham’s water withdrawals.

```
#8
df_asheville <- scrape.it(2015, '01-11-010')

df_both <- rbind(df_durham, df_asheville)

#Plot
ggplot(df_both,aes(x=Date,y=Max_Withdrawals_mgd,
                           color = Water_System)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2015 Water usage data for Durham and Asheville"),
       y="Maximum Withdrawal (mgd)",
       x="Date")
```

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



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.

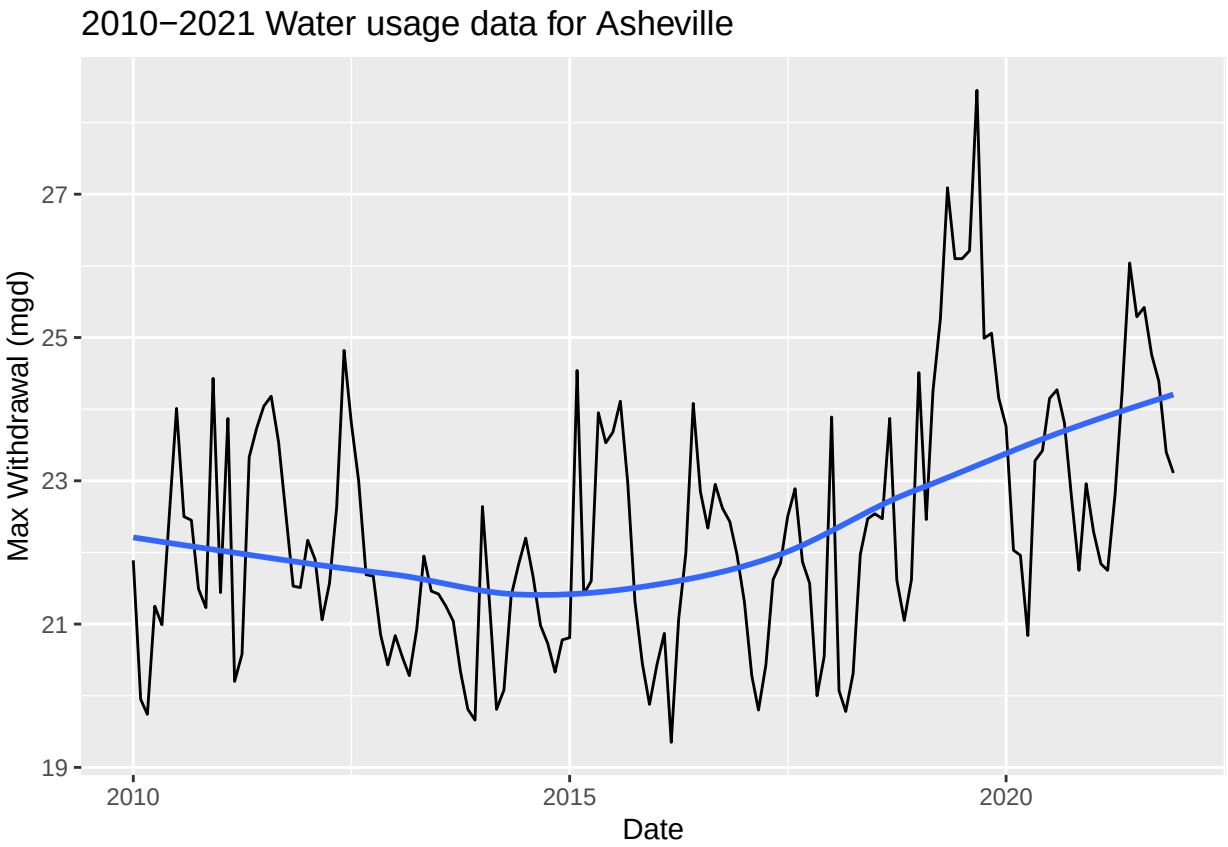
```
#9

# Run scrape function for each year in range
the_dfs <- map(rep(2010:2021),scrape.it,the_pwsid='01-11-010')

#Conflate the returned dataframes into a single dataframe
the_df <- bind_rows(the_dfs)

#Plot, because it's fun and rewarding
ggplot(the_df,aes(x=Date,y=Max_Withdrawals_mgd)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("2010-2021 Water usage data for Asheville"),
       y="Max 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? Maximum daily withdrawals slightly decreased from 2010 to 2015, but increased steadily from 2015 to 2021, surpassing the initial levels in 2010. The trend appears to be that Asheville's water usage has increased over time.