Assignment 10: Data Scraping

Samantha Pace

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
# Installing packages
library(tidyverse)
library(rvest)
library(lubridate)

getwd() # verifying working directory
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```
setwd("~/R/EDE_Fall2023") #setting wd to 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
# Using read_html command to add website into R
website <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022')
website
```

```
## {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
# Scraping water system name
water_system_name <- website %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water_system_name
```

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

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

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

```
# Scraping Ownership
Ownership <- website %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
Ownership
```

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

```
# Scraping 12 months of Max Daily Use
Max_daily_use <- website %>%
  html_nodes("th~ td+ td") %>%
  html_text()
Max_daily_use
```

```
##  [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...

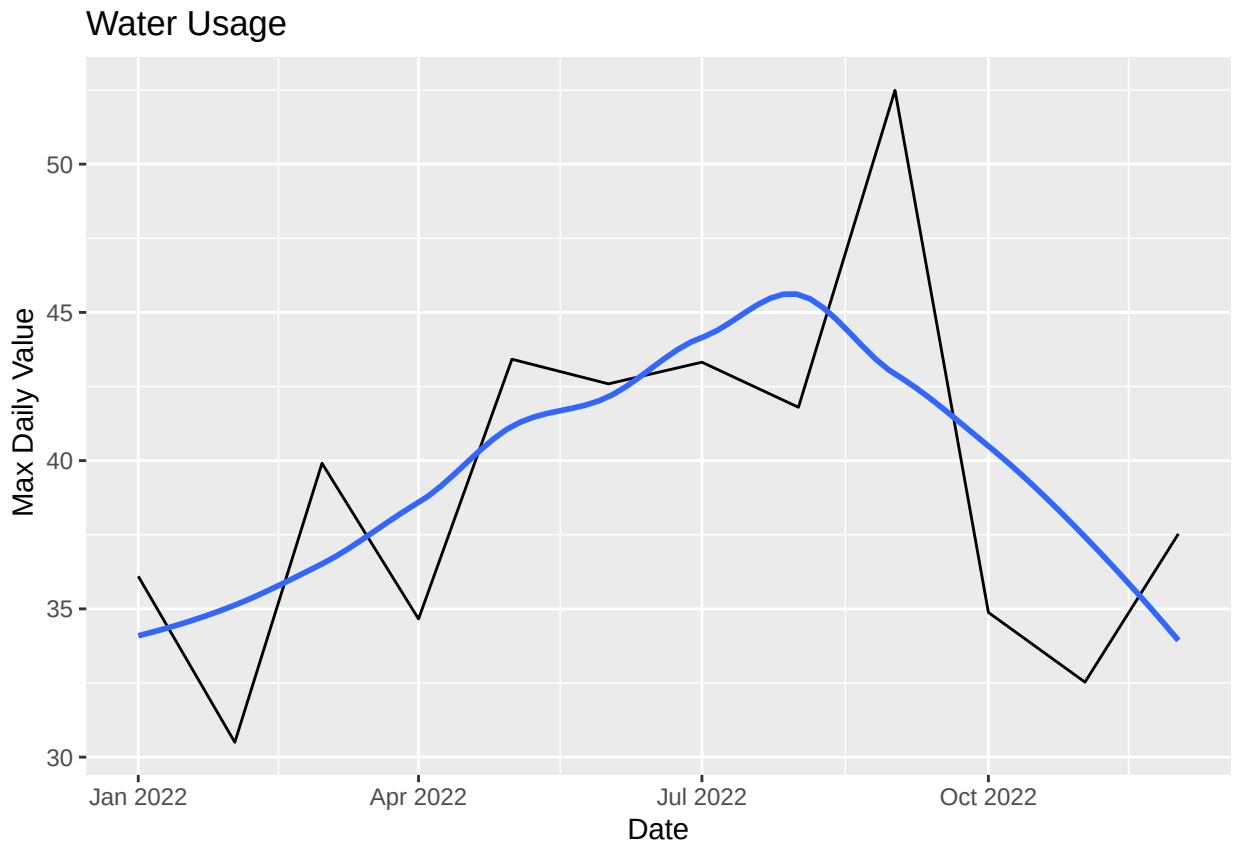
5. Create a line plot of the maximum daily withdrawals across the months for 2022

```
#4
# Creating a data frame
df_water_supply <- data.frame("Month" = c("January", "May", "September", "February", "June", "October",
                              "Year" = rep(2022,12),
                              "Water_System_Name" = rep(water_system_name, 12),
                              "Ownership" = rep(Ownership, 12),
                              "PWSID" = rep(PWSID,12),
                              "Max_daily_use" = as.numeric(Max_daily_use))

#Modifying the dataframe to specify the objects and modify the date
df_water_supply <- df_water_supply %>%
  mutate(Date = my(paste(Month,"-", Year)))

#5
ggplot(df_water_supply, aes(x=Date, y=Max_daily_use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = paste("Water Usage"),
       y=" Max Daily Value",
       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.

# Setting up function
scrape.more <- function(the_year, the_PWSID){

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

   # Organizing code for certain variables
PWSID_tag <-'td tr:nth-child(1) td:nth-child(5)'
Max_daily_use_tag <- 'th~ td+ td'
water_system_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'


  # Scraping code
the_PWSID <- website_function %>% html_nodes(PWSID_tag) %>% html_text()
Max_daily_use <- website_function %>% html_nodes(Max_daily_use_tag) %>% html_text()
water_system_name <- website_function %>% html_nodes(water_system_name_tag) %>%  html_text()

# Convert to a dataframe
df_water_supply_function <- data.frame("Month" = c("January", "May", "September",
                                                   "February", "June", "October",
```

```
                                                    "March", "July", "November",
                                                    "April", "August", "December"),
                                "Year" = rep(the_year,12),
                               "PWSID" = rep(the_PWSID,12),
                                "Water_System_Name" = rep(water_system_name, 12),
                                "Max_daily_use" = as.numeric(Max_daily_use)) %>%
  mutate(Water_System_Name = !!water_system_name,
         PWSID = !!the_PWSID,
         Date = my(paste(Month,"-", Year)))

return(df_water_supply_function)
}
```

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

```
#7
# Using function to scrape the Durham 2015
Durham_2015 <- scrape.more(2015, '03-32-010')
view(Durham_2015)
```

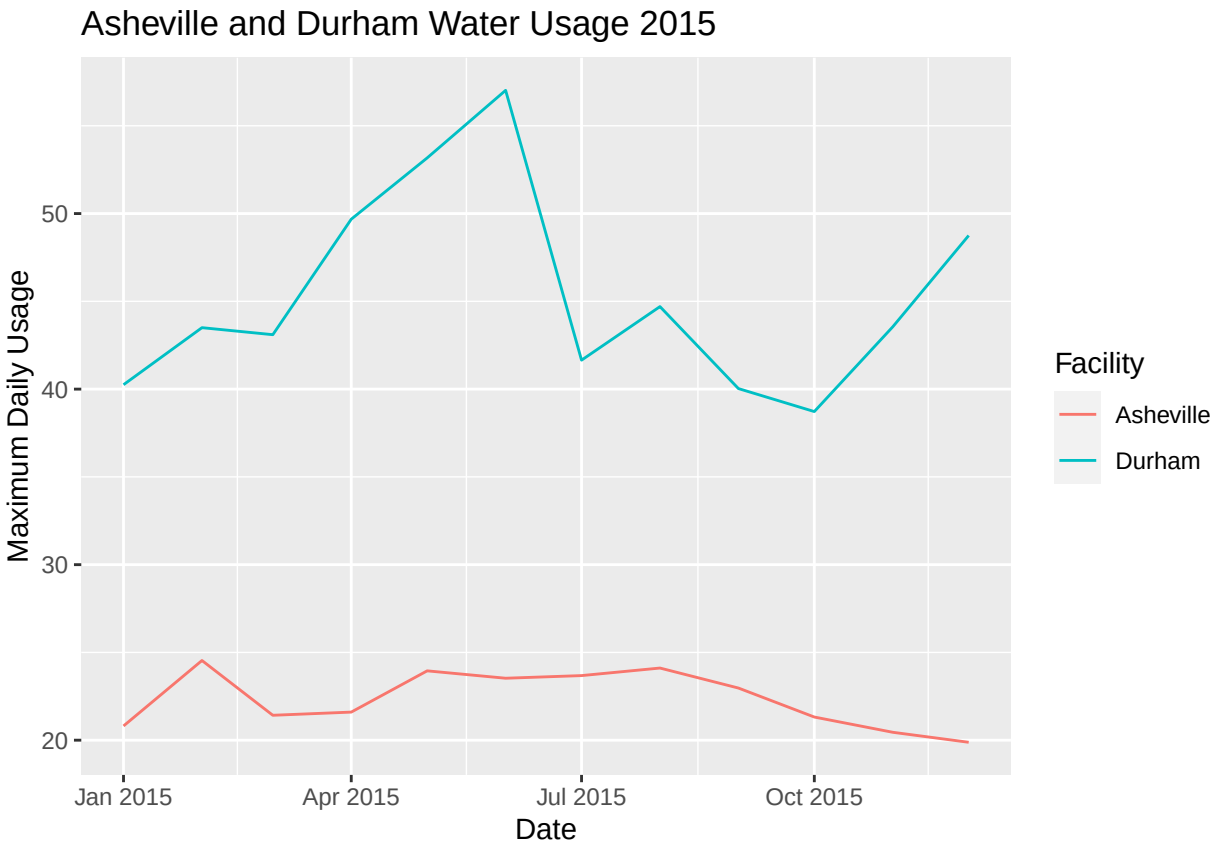
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville’s to Durham’s water withdrawals.

```
#8
# Extracting Asheville data
Asheville_2015 <- scrape.more(2015, '01-11-010')
view(Asheville_2015)

# Combining Asheville and Durham Datasets
Avl_Durham_2015 <- rbind(Durham_2015, Asheville_2015)

view(Avl_Durham_2015)

ggplot(Avl_Durham_2015,
       aes(x = Date, y = Max_daily_use,
           color = Water_System_Name)) +
  geom_line() +
  labs(title = "Asheville and Durham Water Usage 2015",
       y = "Maximum Daily Usage",
       color = "Facility")
```



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.

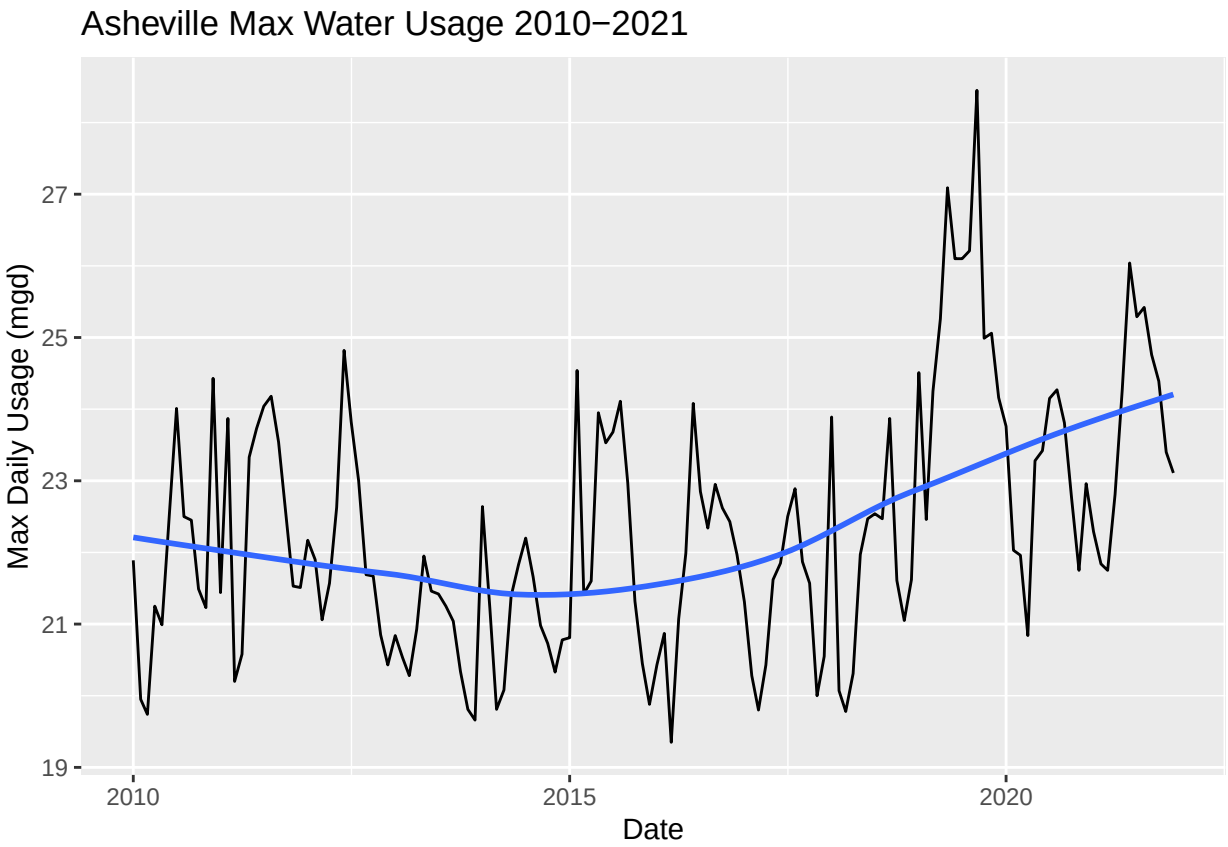
```
#9
# Setting variables
the_years <- c(2010:2021)
PWSID_Avl <- rep("01-11-010",12)

# Mapping the scraping
Avl_2010_2021 <- map2(the_years, PWSID_Avl, scrape.more)

# Binding rows
df_Avl_2010_2021 <- bind_rows(Avl_2010_2021)

# Plotting
ggplot(df_Avl_2010_2021,
       aes(x=Date, y = Max_daily_use)) +
  geom_line() +
  geom_smooth(method="loess",se=FALSE) +
  labs(title = "Asheville Max Water Usage 2010-2021",
       y = "Max Daily Usage (mgd)")
```

```
## `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: By looking at the plot, it does look like Ashevulle has a trend in water usage over time. From 2010 to 2015, there was a slight decrease in max daily usage. From 2015 onward there has been a significant increase in water usage. >