Assignment 10: Data Scraping

Courtney Horn

Total points:

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_10_Data_Scraping.Rmd") prior to submission.

The completed exercise is due on Tuesday, April 6 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Set your ggplot theme

```
#1
getwd()
```

```
## [1] "/Users/courtneyannehorn/Desktop/EDA/EDAfin/Assignments"
```

```
library(tidyverse)
library(rvest)
library(ggplot2)
library(lubridate)


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 2019 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Change the date from 2020 to 2019 in the upper right corner.
- 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=2019

https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019 Indicate this website as the as the URL to be scraped.

```
#2
webpage <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019')
```

3. The data we want to collect are listed below:

- From the "System Information" section:
- Water system name
- PSWID
- Ownership
- From the "Water Supply Sources" section:
- Maximum monthly withdrawals (MGD)

In the code chunk below scrape these values into the supplied variable names.

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

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

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

#maximum monthly withdrawals (MGD)
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.)

   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc...

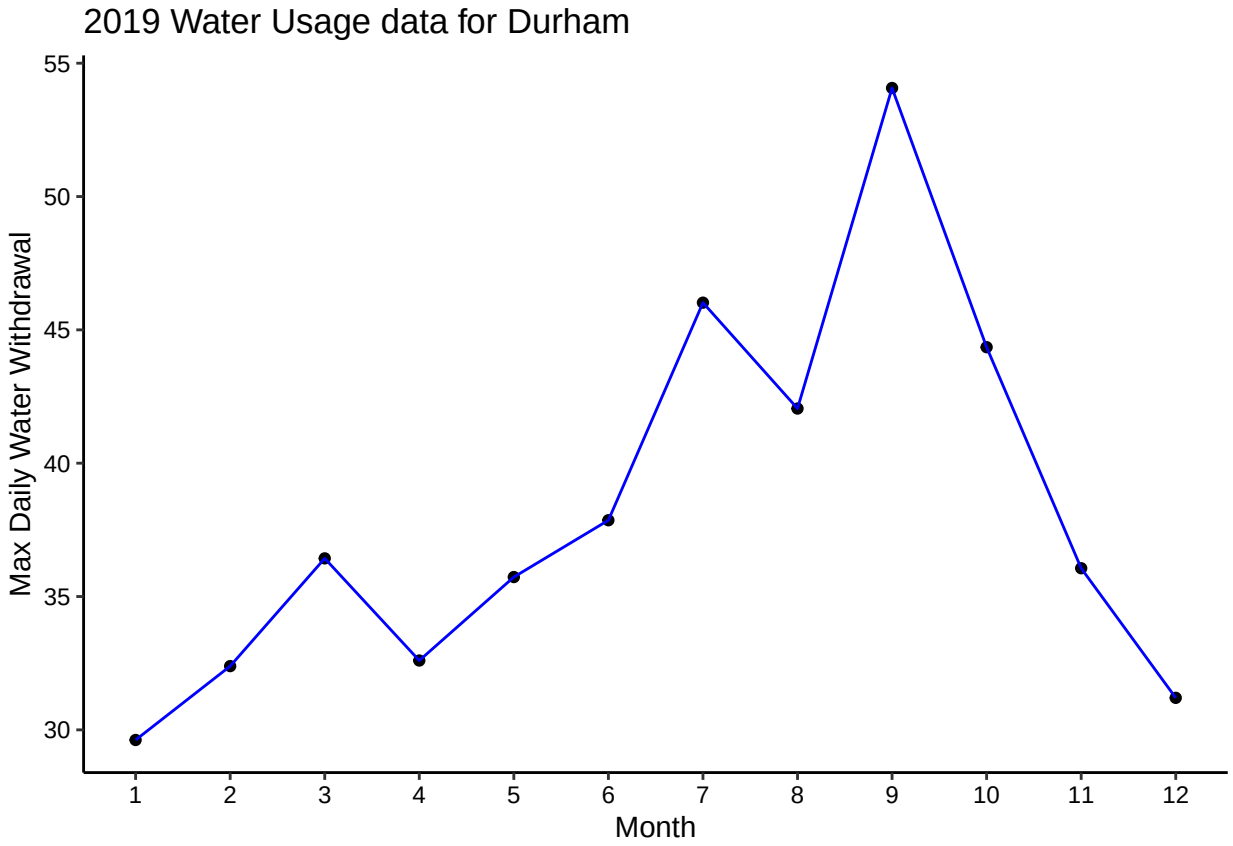
5. Plot the max daily withdrawals across the months for 2019.

```
#4
daily_withdrawals_Df <- data.frame("Month" = rep(1:12),
                                   "Year" = rep(2019,12),
                                   "Water_system_name" = as.character(watsys_name),
```

```
                                    "PSWID" = as.factor(PSWID),
                                    "ownership" =
                                      as.character(Own),
                                    "Max_Daily_Withdrawals" = as.numeric(MGD)
                                  )

#View(daily_withdrawals_Df)

daily_withdrawals_Df2 <- daily_withdrawals_Df %>%
  mutate(Date = my(paste(Month, "-", Year)))

#View(daily_withdrawals_Df2)

#class(daily_withdrawals_Df2$Date)

daily_withdrawals_Df3 <- data.frame("Month" = c("January", "February", "March", "April", "May", "June",
                                    "Year" = rep(2019,12),
                                    "Water_system_name" = as.character(watsys_name),
                                    "PSWID" = as.factor(PSWID),
                                    "ownership" =
                                      as.character(Own)
                                  )
#View(daily_withdrawals_Df3)

daily_withdrawals_Df4 <- daily_withdrawals_Df3 %>%
  mutate(Date = my(paste(Month, "-", Year)))

#View(daily_withdrawals_Df4)

daily_withdrawals_Df5 <- data.frame("Year" = rep(2019,12),
                                    "Water_system_name" = as.character(watsys_name),
                                    "PSWID" = as.factor(PSWID),
                                    "ownership" =
                                      as.character(Own),
                                    "Max_Daily_Withdrawals" = as.numeric(MGD))

daily_withdrawals_Df6 <- daily_withdrawals_Df5 %>%
  mutate(Month = c("January", "May", "September", "February", "June", "October", "March", "July", "Nover
  mutate(Date = my(paste(Month, "-", Year)))

#View(daily_withdrawals_Df6)


daily_withdrawals_Df7 <- daily_withdrawals_Df5 %>%
  mutate(Month = c(1,5,9,2,6,10,3,7,11,4,8,12)) %>%
  mutate(Date = my(paste(Month, "-", Year)))

#View(daily_withdrawals_Df7)

class(daily_withdrawals_Df7$Date)
```

```
## [1] "Date"
```

```r
#the order of the monthly data is January, May, September, Feb, Jun, Oct, Marc, Jul, Nov, Apr, Aug, Dec


#5
max_daily_wd_plot1 <-
  ggplot(daily_withdrawals_Df7, aes(x=Month, y=Max_Daily_Withdrawals)) +
  geom_point() +
  geom_line(color = "blue") +
  scale_x_continuous(breaks = c(1:12))


 #labs(title = paste("2019 Water usage data for Durham"),
      # y="Withdrawal (mgd)",
      #x="Date"))


#print(max_daily_wd_plot1)

max_daily_wd_plot2 <-
  ggplot(daily_withdrawals_Df7, aes(x=Month, y=Max_Daily_Withdrawals)) +
  geom_point() +
  geom_line(color = "blue") +
  scale_x_continuous(breaks = c(1:12),
  labs(title = paste("2019 Water usage data for Durham"),
      y="Withdrawal (mgd)",
      x="Date"))

#max_daily_wd_plot2

max_daily_wd_plot3 <-
  ggplot(daily_withdrawals_Df7, aes(x=Month, y=Max_Daily_Withdrawals)) +
  geom_point() +
  geom_line(color = "blue") +
  labs(title = paste("2019 Water Usage data for Durham"),
      y="Max Daily Water Withdrawal",
      x="Month") +
 scale_x_continuous(breaks = c(1:12))

print(max_daily_wd_plot3)
```



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 data scraped. #https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019

```
#6.
#https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019
#https://www.ncwater.org/WUDC/app/LWSP/report.php?pswid=03-32-010&year=2019

#trying to construct a scraping address
base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_pwsid <- '03-32-010'
the_year <- '2019'
scrape_url <- paste0(base_url, 'pwsid=', the_pwsid, '&', 'year=', the_year)
print(scrape_url)
```

```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2019"
```

```
#trying again to construct a scraping address
base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_pswid <- '03-32-010'
the_year <- '2019'
scrape_url1 <- paste0(base_url, 'pswid=', the_pswid)
print(scrape_url1)
```

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

```
#Trying to retrieve the website contents
the_website <- read_html(scrape_url)
```

```
the_website1 <- read_html(scrape_url1)


#Set the element address variables (determined in the previous step)
pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
watsys_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
Own_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
MGD_tag <- 'th~ td+ td'
#th~ td+ td



#Scrape the data items
pwsid <- the_website %>% html_nodes(pwsid_tag) %>% html_text()
wat_sys_name <- the_website %>%   html_nodes(watsys_name_tag) %>%  html_text()
ownership <- the_website %>% html_nodes(Own_tag) %>% html_text()
MGD_withdrawals <- the_website %>% html_nodes(MGD_tag) %>% html_text()

#Construct a dataframe from the scraped data
df_withdrawals <- data.frame("Month" = rep(1:12),
                             "Year" = rep(the_year,12),
                             "Avg_Withdrawals_mgd" = as.numeric(MGD_withdrawals)) %>%
  mutate(system_name = !!wat_sys_name,
         Ownership = !!ownership,
         Date = my(paste(Month,"-",Year)))

#View(df_withdrawals)

df_withdrawals1 <- df_withdrawals
df_withdrawals1$Date <- as.Date(df_withdrawals1$Date)
class(df_withdrawals1$Date)
```

```
## [1] "Date"
```

```
class(df_withdrawals$Date)
```

```
## [1] "Date"
```

```
#now creating the scraping function
#Create our scraping function
scrape.help <- function(the_year, the_pwsid){

  the_website <- read_html(paste0(base_url, 'pwsid=', the_pwsid, '&', 'year=', the_year))

   #Set the element address variables (determined in the previous step)
pwsid_tag <- 'td tr:nth-child(1) td:nth-child(5)'
watsys_name_tag <- 'div+ table tr:nth-child(1) td:nth-child(2)'
Own_tag <- 'div+ table tr:nth-child(2) td:nth-child(4)'
MGD_tag <- 'th~ td+ td'

  #Scrape the data items
 pwsid <- the_website %>% html_nodes(pwsid_tag) %>% html_text()
wat_sys_name <- the_website %>%   html_nodes(watsys_name_tag) %>%  html_text()
ownership <- the_website %>% html_nodes(Own_tag) %>% html_text()
MGD_withdrawals <- the_website %>% html_nodes(MGD_tag) %>% html_text()
```

```
  #Convert to a dataframe
dfhelp <- data.frame("Month" = rep(1:12),
                             "Year" = rep(the_year,12),
                             "Avg_Withdrawals_mgd" = as.numeric(MGD_withdrawals)) %>%
  mutate(system_name = !!wat_sys_name,
         Ownership = !!ownership,
         pwsid = !!the_pwsid,
         Date = my(paste(Month,"-",Year)))

  #Pause for a moment - scraping etiquette
  Sys.sleep(1) #uncomment this if you are doing bulk scraping!
  #Return the dataframe
  return(dfhelp)
}
```

7. Use the function above to extract and plot max daily withdrawals for Durham for each month in 2015

```
#7

durham2015 <- scrape.help(2015, '03-32-010')
#View(durham2015)
```

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 the Asheville to Durham's water withdrawals.

```
#8

ash2015 <- scrape.help(2015, '01-11-010')
#View(ash2015)

#combine asheville and durham data
dim(durham2015)
```

```
## [1] 12  7
```

```
dim(ash2015)
```

```
## [1] 12  7
```

```
df2015 <- full_join(ash2015, durham2015)
```

```
## Joining, by = c("Month", "Year", "Avg_Withdrawals_mgd", "system_name", "Ownership", "pwsid", "Date")
```

```
#View(df2015)
colnames(df2015)
```

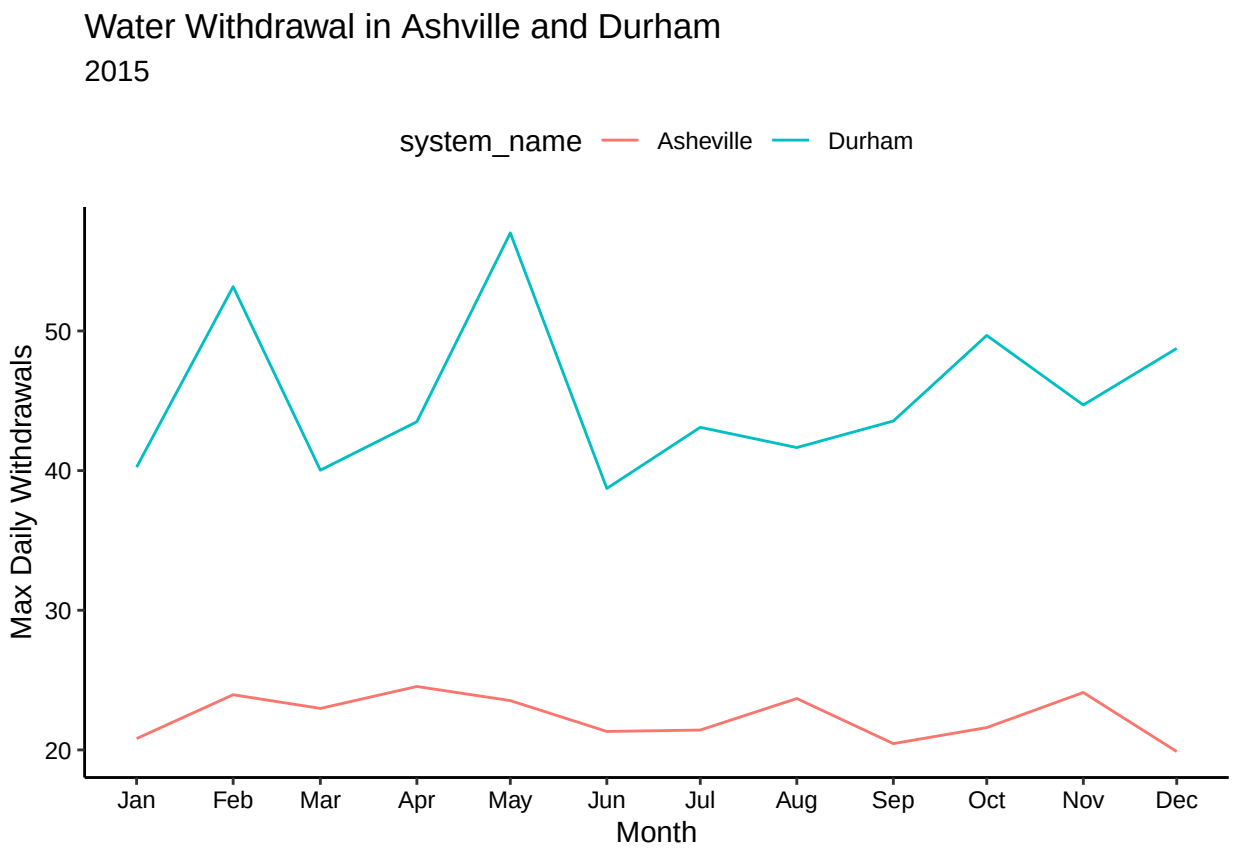
```
## [1] "Month"               "Year"                "Avg_Withdrawals_mgd"
## [4] "system_name"         "Ownership"           "pwsid"
## [7] "Date"
```

```
min(df2015$Date)
```

```
## [1] "2015-01-01"
```

```
ggplot(data=df2015, aes(x = Date, y = Avg_Withdrawals_mgd, color = system_name))+
  geom_line() +
  labs(title = paste("Water Withdrawal in Ashville and Durham"), subtitle = "2015", y = "Max Daily Withd
  scale_fill_discrete(name = "Location") +
```

```
  scale_x_date(limits = as.Date(c("2015-01-01", "2015-12-01")),
    date_breaks = "1 months", date_labels = "%b")
```




```
#Set legend title and labels with a scale function.
```

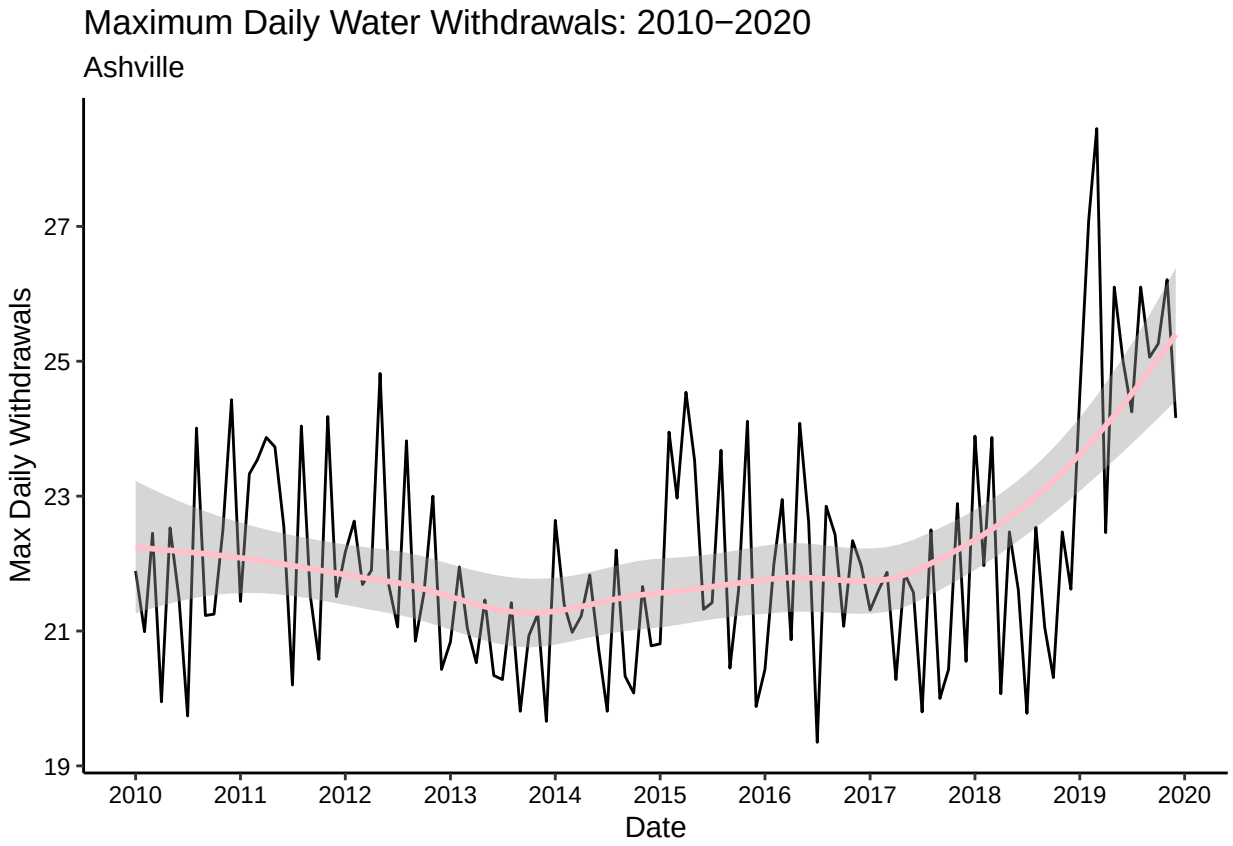
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9

the_years = rep(2010:2019)
my_pwsid = '01-11-010'

#Use lapply to apply the scrape function
the_dfs <- lapply(X = the_years,
                  FUN = scrape.help,
                  the_pwsid=my_pwsid)
#lapply allows you to repeat the scrape.it function for each value in the array of years.
View(the_dfs)

#combining the data frames
the_df <- bind_rows(the_dfs)
#View(the_df)
colnames(the_df)
```

```
## [1] "Month"              "Year"               "Avg_Withdrawals_mgd"
## [4] "system_name"        "Ownership"          "pwsid"
## [7] "Date"
```

```r
#plotting the data
ashplot2 <-
  ggplot(the_df, aes(x=Date, y = Avg_Withdrawals_mgd)) +
  geom_line() +
  scale_x_date(date_labels = "%Y", breaks = "year") +
  labs(title = paste("Maximum Daily Water Withdrawals: 2010-2020"),
       subtitle = "Ashville",
       y="Max Daily Withdrawals",
       x="Date") +
  geom_smooth(color = "pink")

ashplot2
```

```
## `geom_smooth()` using method = 'loess' and 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? Ashville appears to have an upward trend in water usage.