Assignment 10: Data Scraping

Kathleen Mason

Total points:

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

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., "Salk_A06_GLMs_Week1.Rmd") prior to submission.

The completed exercise is due on Tuesday, April 7 at 1:00 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

```
getwd()
```

```
## [1] "/Users/kathleenmason/Documents/DUKE/Data Analytics/Environmental_Data_Analytics_2020/Assignments
```

```
library(ggplot2)
library(tidyverse)
library(rvest)
library(ggrepel)
library(stringr)
library(dplyr)
library(viridis)

mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black", angle=90),
        legend.position = "top")
theme_set(mytheme)
```

2. Indicate the EPA impaired waters website (https://www.epa.gov/nutrient-policy-data/waters-assessed-impaired-due-nutrient-related-causes) as the URL to be scraped.

```
url <- "https://www.epa.gov/nutrient-policy-data/waters-assessed-impaired-due-nutrient-related-causes"
webpage <- read_html(url)
```

3. Scrape the Rivers table, with every column except year. Then, turn it into a data frame.

```
State <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(1)") %>%
  html_text()
Rivers.Assessed.mi2 <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(2)") %>%
  html_text()
Rivers.Assessed.percent <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(3)") %>%
  html_text()
Rivers.Impaired.mi2 <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(4)") %>%
  html_text()
Rivers.Impaired.percent <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(5)") %>%
  html_text()
Rivers.Impaired.percent.TMDL <- webpage %>%
  html_nodes("table:nth-child(8) td:nth-child(6)") %>%
  html_text()

Rivers <- data.frame(State, Rivers.Assessed.mi2,
                     Rivers.Assessed.percent, Rivers.Impaired.mi2,
                     Rivers.Impaired.percent,
                     Rivers.Impaired.percent.TMDL)
```

4. Use `str_replace` to remove non-numeric characters from the numeric columns.

5. Set the numeric columns to a numeric class and verify this using `str`.

```
# 4
Rivers$Rivers.Assessed.mi2 <-
  str_replace(Rivers$Rivers.Assessed.mi2,
              pattern = "([,])", replacement = "")

Rivers$Rivers.Assessed.percent <-
  str_replace(Rivers$Rivers.Assessed.percent,
              pattern = "([%])", replacement = "")

Rivers$Rivers.Impaired.mi2 <-
  str_replace(Rivers$Rivers.Impaired.mi2,
              pattern = "([,])", replacement = "")

Rivers$Rivers.Impaired.percent<-
  str_replace(Rivers$Rivers.Impaired.percent,
              pattern = "([%])", replacement = "")

Rivers$Rivers.Impaired.percent.TMDL <-str_replace(Rivers$Rivers.Impaired.percent.TMDL,
          pattern = "([%])", replacement = "")
Rivers$Rivers.Impaired.percent.TMDL <-str_replace(Rivers$Rivers.Impaired.percent.TMDL,
          pattern = "([±])", replacement = "")

# 5
str(Rivers)
```

```
## 'data.frame':    50 obs. of  6 variables:
```

```
##  $ State                       : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Rivers.Assessed.mi2         : chr  "10538" "602" "2764" "9979" ...
##  $ Rivers.Assessed.percent     : chr  "14" "0" "3" "11" ...
##  $ Rivers.Impaired.mi2         : chr  "1146" "15" "144" "1440" ...
##  $ Rivers.Impaired.percent     : chr  "11" "2" "5" "14" ...
##  $ Rivers.Impaired.percent.TMDL: chr  "53" "100" "6" "2" ...
```

```
Rivers$Rivers.Assessed.mi2 <-
  as.numeric(Rivers$Rivers.Assessed.mi2)
Rivers$Rivers.Assessed.percent <-
  as.numeric(Rivers$Rivers.Assessed.percent)
```

```
## Warning: NAs introduced by coercion
```

```
Rivers$Rivers.Impaired.mi2 <-
  as.numeric(Rivers$Rivers.Impaired.mi2)
Rivers$Rivers.Impaired.percent <-
  as.numeric(Rivers$Rivers.Impaired.percent)
Rivers$Rivers.Impaired.percent.TMDL <- as.numeric(Rivers$Rivers.Impaired.percent.TMDL)
```

6. Scrape the Lakes table, with every column except year. Then, turn it into a data frame.

```
State <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(1)") %>%
  html_text()
Lakes.Assessed.acre <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(2)") %>%
  html_text()
Lakes.Assessed.percent <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(3)") %>%
  html_text()
Lakes.Impaired.acre <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(4)") %>%
  html_text()


Lakes.Impaired.percent <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(5)") %>%
  html_text()
Lakes.Impaired.percent.TMDL <- webpage %>%
  html_nodes("table:nth-child(14) td:nth-child(6)") %>%
  html_text()

Lakes <- data.frame(State, Lakes.Assessed.acre,
                    Lakes.Assessed.percent, Lakes.Impaired.acre,
                    Lakes.Impaired.percent,
                    Lakes.Impaired.percent.TMDL)
```

7. Filter out the states with no data.
8. Use `str_replace` to remove non-numeric characters from the numeric columns.
9. Set the numeric columns to a numeric class and verify this using `str`.

```
# 7
Lakes <- Lakes %>%
  filter(State != "Hawaii" & State != "Pennsylvania")
```

```
# 8
Lakes$Lakes.Assessed.acre <-
  str_replace(Lakes$Lakes.Assessed.acre,
              pattern = "([,])", replacement = "")
Lakes$Lakes.Assessed.percent <-
  str_replace(Lakes$Lakes.Assessed.percent,
              pattern = "([%])", replacement = "")
Lakes$Lakes.Impaired.acre <-
  str_replace(Lakes$Lakes.Impaired.acre,
              pattern = "([,])", replacement = "")
Lakes$Lakes.Impaired.percent <-
  str_replace(Lakes$Lakes.Impaired.percent,
              pattern = "([%])", replacement = "")
Lakes$Lakes.Impaired.percent.TMDL <- str_replace(Lakes$Lakes.Impaired.percent.TMDL,
          pattern = "([%])", replacement = "")
Lakes$Lakes.Impaired.percent.TMDL <- str_replace(Lakes$Lakes.Impaired.percent.TMDL,
      pattern = "([±])", replacement = "")
# 9
str(Lakes)
```

```
## 'data.frame':    48 obs. of  6 variables:
##  $ State                      : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Lakes.Assessed.acre        : chr  "430.976" "5981" "114976" "64778" ...
##  $ Lakes.Assessed.percent     : chr  "88" "0" "34" "13" ...
##  $ Lakes.Impaired.acre        : chr  "81740" "1137" "4895" "6513" ...
##  $ Lakes.Impaired.percent     : chr  "19" "19" "4" "10" ...
##  $ Lakes.Impaired.percent.TMDL: chr  "53" "73" "9" "71" ...
```

```
Lakes$Lakes.Assessed.acre <-
  as.numeric(Lakes$Lakes.Assessed.acre)
```

```
## Warning: NAs introduced by coercion
```

```
Lakes$Lakes.Assessed.percent <-
  as.numeric(Lakes$Lakes.Assessed.percent)
```

```
## Warning: NAs introduced by coercion
```

```
Lakes$Lakes.Impaired.acre<-
  as.numeric(Lakes$Lakes.Impaired.acre)
Lakes$Lakes.Impaired.percent <-
  as.numeric(Lakes$Lakes.Impaired.percent)
Lakes$Lakes.Impaired.percent.TMDL <- as.numeric(Lakes$Lakes.Impaired.percent.TMDL)
str(Lakes)
```

```
## 'data.frame':    48 obs. of  6 variables:
##  $ State                      : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Lakes.Assessed.acre        : num  431 5981 114976 64778 NA ...
##  $ Lakes.Assessed.percent     : num  88 0 34 13 50 95 47 100 54 82 ...
##  $ Lakes.Impaired.acre        : num  81740 1137 4895 6513 473954 ...
##  $ Lakes.Impaired.percent     : num  19 19 4 10 45 7 12 88 82 2 ...
##  $ Lakes.Impaired.percent.TMDL: num  53 73 9 71 NA 0 7 69 NA 20 ...
```

10. Join the two data frames with a `full_join`.

```
LakesandRivers<- full_join(Rivers, Lakes)
```

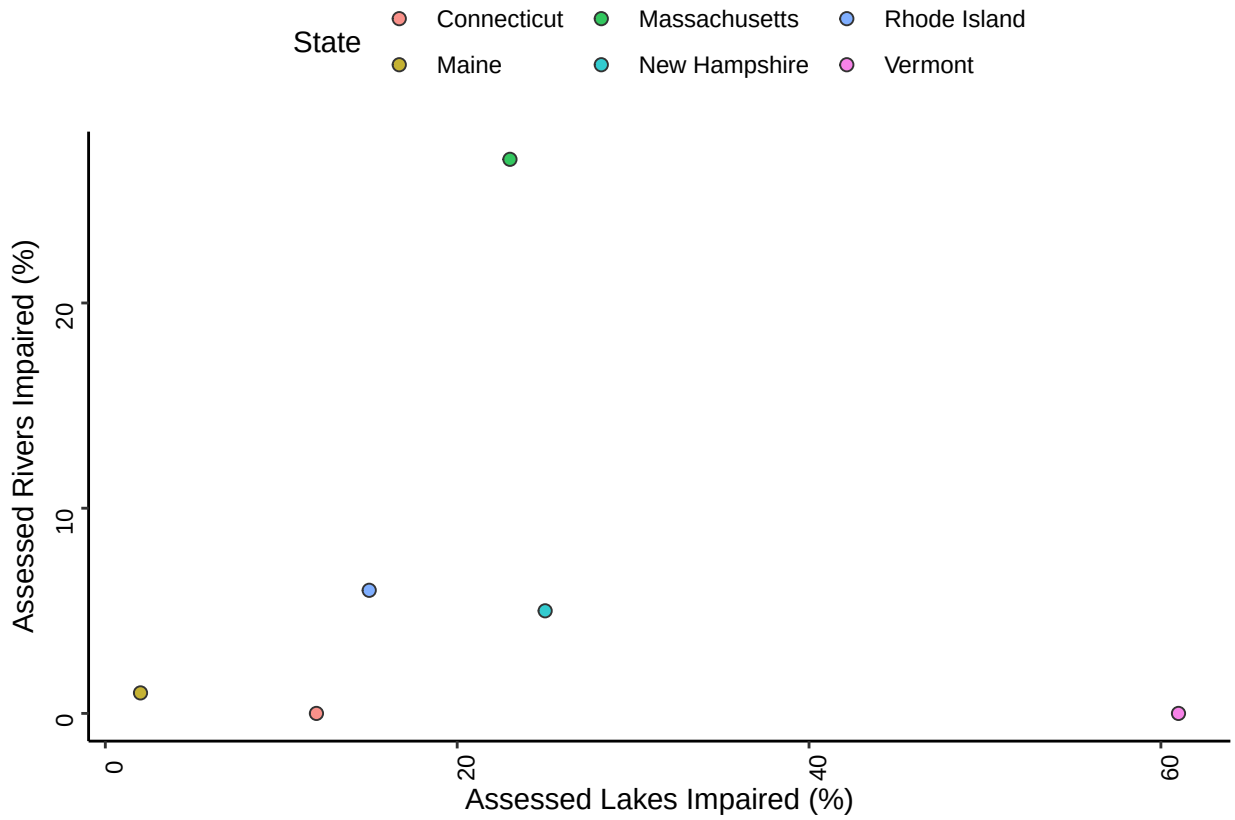
```
## Joining, by = "State"
```

11. Create one graph that compares the data for lakes and/or rivers. This option is flexible; choose a relationship (or relationships) that seem interesting to you, and think about the implications of your findings. This graph should be edited so it follows best data visualization practices.

(You may choose to run a statistical test or add a line of best fit; this is optional but may aid in your interpretations)

```
NewEngland<- LakesandRivers %>%
  filter(State == "Massachusetts" | State == "Maine" | State == "New Hampshire" | State == "Vermont" | S


Impaired_percent_NE<- ggplot(NewEngland,
                             aes(x=Lakes.Impaired.percent, y= Rivers.Impaired.percent , fill=State)) +
  geom_point(shape = 21, size = 2, alpha = 0.8)+
  ylab(expression("Assessed Rivers Impaired (%)"))+
  xlab("Assessed Lakes Impaired (%)")+
  mytheme
print(Impaired_percent_NE)
```



12. Summarize the findings that accompany your graph. You may choose to suggest further research or data collection to help explain the results.

    I was interested to see if New England States specifically had the percent of assessed impaired to be similar for both rivers and lakes. I assumed if a large percent of lakes were impaired, than rivers would also be as they can be connected. However, I found out that this is not the case for every state in New England. For example, Vermont has an extremely high number of lakes impaired, but a percent of zero for rivers impaired. You could say, well maybe Vermont actually assesses less rivers, but Vermont assessed 78% of rivers, and 100% of lakes, which isn't drastically

different. Maybe a statistical test needs to be done to show whether the percent of assessment for lakes and rivers plays a role in these relationships.