# Assignment 10: Data Scraping

Hannah Smith

## 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/hannahsmith/Documents/ENV872/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
library(viridis)
library(rvest)
#install.packages("ggrepel")
library(ggrepel)

mytheme <- theme_classic() +
  theme(axis.text = element_text(color = "black"),
        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_te

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

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$Rivers.Assessed.percent <- str_replace(Rivers$Rivers.Assessed.percent,
                                               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
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)
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         : num  3 41 20 49 29 36 17 18 2 8 ...
##  $ Rivers.Assessed.percent     : num  14 0 3 11 16 56 41 100 20 19 ...
##  $ Rivers.Impaired.mi2         : num  4 14 13 5 12 28 19 26 40 11 ...
##  $ Rivers.Impaired.percent     : num  11 2 5 14 41 0 0 88 53 9 ...
##  $ Rivers.Impaired.percent.TMDL: num  53 100 6 2 NA 14 73 37 NA 78 ...
```

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.mi2 <- 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.mi2 <- 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_te

Lakes <- data.frame(State, Lakes.Assessed.mi2, Lakes.Assessed.percent, Lakes.Impaired.mi2, Lakes.Impair
```

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$Lakes.Assessed.percent <- str_replace(Lakes$Lakes.Assessed.percent,
                                            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
Lakes$Lakes.Assessed.mi2 <- as.numeric(Lakes$Lakes.Assessed.mi2)
Lakes$Lakes.Assessed.percent <- as.numeric(Lakes$Lakes.Assessed.percent)
```

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

```
Lakes$Lakes.Impaired.mi2 <- as.numeric(Lakes$Lakes.Impaired.mi2)
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.mi2        : num  33 37 6 43 1 14 30 20 2 31 ...
##  $ Lakes.Assessed.percent    : num  88 0 34 13 50 95 47 100 54 82 ...
##  $ Lakes.Impaired.mi2        : num  42 2 31 39 35 4 27 20 45 40 ...
##  $ 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`.

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

```
## 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)

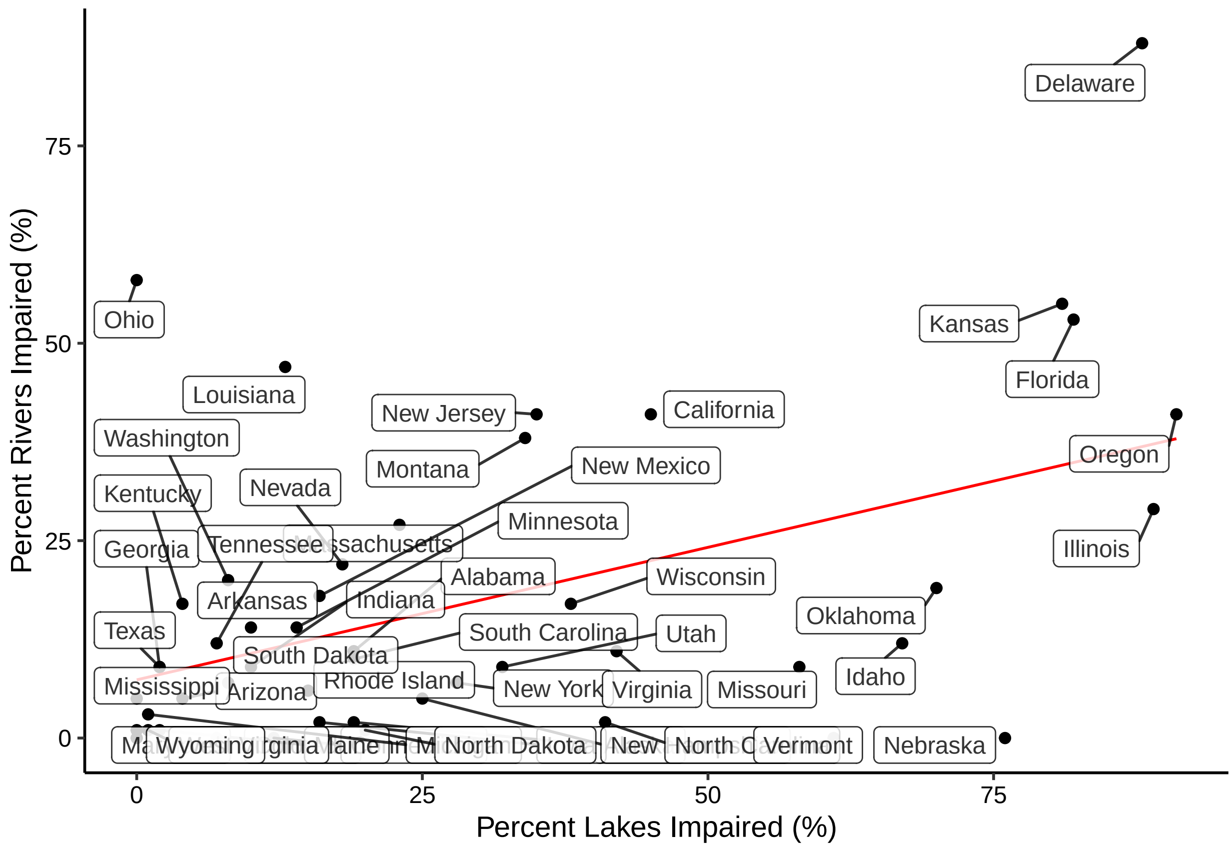
```
LakesRiversGraph <- ggplot(LakesRivers) +
  geom_point(aes(x = Lakes.Impaired.percent, y = Rivers.Impaired.percent)) +
  geom_smooth(method = "lm", se = FALSE, aes(x = Lakes.Impaired.percent, y = Rivers.Impaired.percent),
              color = "red", alpha = 50/100, size = 0.5) +
  geom_label_repel(aes(label = State, x = Lakes.Impaired.percent, y = Rivers.Impaired.percent),
                   nudge_x = -5, nudge_y = -5,
                   size = 3, alpha = 0.8) +
  labs(x = "Percent Lakes Impaired (%)",
       y = "Percent Rivers Impaired (%)")
```

```r
print(LakesRiversGraph)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_label_repel).
```




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

    In general, the percentage of lakes impaired is directly proportional to the percentage of rivers impaired in each state. However, states like Ohio, Vermont, and Nebraska are outliers to this statement. Ohio has approximately 60% impairement in the rivers assessed, while 0% of the assessed lakes are impaired. Vermont, on the other hand, has approximately 60% of the assessed lakes impaired and 0% of the assessed are impaired. A potential problem with this analysis is the states do not have the same percentage of assessed rivers and lakes, so the percent impaired can be misleading across the states as a whole.