Assignment 10: Data Scraping

Jared Wang

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] "C:/Users/wangc/Box/Home Folder cw369/Private/Duke/Course/Spring 2020/Environmental_Data_Analyti
```

```
library(tidyverse)
library(rvest)
library(ggthemes)

theme.1.0 <- theme_classic()
theme_set(theme.1.0)

theme.hc01 <- theme_hc() +
  theme(axis.title = element_text(family = "serif", size = (10)),
        axis.text = element_text(family = "serif", size = (8), color = "black"),
        legend.title = element_text(size = 10, family = "serif"),
        legend.text = element_text(size = 10, family = "serif"),
        legend.key = element_rect(color = NA, fill = NA),
        legend.background = element_rect(color = NA, fill = NA),
        legend.position = "right")
```

2. Indicate the EPA impaired waters website (https://www.epa.gov/nutrient-policy-data/waters-assessed-impaired-due-nutri 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_t

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 %>%
  mutate(Rivers.Assessed.mi2 = str_replace(Rivers.Assessed.mi2,
                                           pattern = "([,])", replacement = ""),
         Rivers.Assessed.percent = str_replace(Rivers.Assessed.percent,
                                           pattern = "([%])", replacement = ""),
         Rivers.Assessed.percent = str_replace(Rivers.Assessed.percent,
                                           pattern = "([*])", replacement = ""),
         Rivers.Impaired.mi2 = str_replace(Rivers.Impaired.mi2,
                                           pattern = "([,])", replacement = ""),
         Rivers.Impaired.percent = str_replace(Rivers.Impaired.percent,
                                           pattern = "([%])", replacement = ""),
         Rivers.Impaired.percent.TMDL = str_replace(Rivers.Impaired.percent.TMDL,
                                           pattern = "([%])", replacement = ""),
         Rivers.Impaired.percent.TMDL = str_replace(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)
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  10538 602 2764 9979 32803 ...
##  $ Rivers.Assessed.percent    : num  14 0 3 11 16 56 41 100 20 19 ...
##  $ Rivers.Impaired.mi2        : num  1146 15 144 1440 13350 ...
##  $ 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.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 & 8
Lakes.update <- Lakes %>%
  filter(Lakes.Assessed.mi2 != "No data") %>%
  mutate(Lakes.Assessed.mi2 = str_replace(Lakes.Assessed.mi2,
                                          pattern = "([,])", replacement = ""),
         Lakes.Assessed.mi2 = str_replace(Lakes.Assessed.mi2,
                                          pattern = "([,])", replacement = ""),
         Lakes.Assessed.percent = str_replace(Lakes.Assessed.percent,
                                          pattern = "([%])", replacement = ""),
         Lakes.Assessed.percent = str_replace(Lakes.Assessed.percent,
                                          pattern = "([*])", replacement = ""),
         Lakes.Impaired.mi2 = str_replace(Lakes.Impaired.mi2,
                                          pattern = "([,])", replacement = ""),
         Lakes.Impaired.percent = str_replace(Lakes.Impaired.percent,
                                          pattern = "([%])", replacement = ""),
         Lakes.Impaired.percent.TMDL = str_replace(Lakes.Impaired.percent.TMDL,
                                          pattern = "([%])", replacement = ""),
         Lakes.Impaired.percent.TMDL = str_replace(Lakes.Impaired.percent.TMDL,
                                          pattern = "([±])", replacement = ""))
# 9
str(Lakes)
```

```
## 'data.frame':    50 obs. of  6 variables:
##  $ State                     : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Lakes.Assessed.mi2        : Factor w/ 49 levels "1,051,246","1,124,399",..: 33 37 6 43 1 14 30 2
##  $ Lakes.Assessed.percent    : Factor w/ 36 levels "0%","100%","100%*",..: 30 1 10 4 14 31 12 2 15 1
##  $ Lakes.Impaired.mi2        : Factor w/ 47 levels "0","1,137","10,007",..: 42 2 31 39 35 4 27 20 45
##  $ Lakes.Impaired.percent    : Factor w/ 36 levels "0%","1%","10%",..: 10 10 20 3 23 27 4 33 32 11
##  $ Lakes.Impaired.percent.TMDL: Factor w/ 22 levels "±","0%","1%",..: 12 17 20 16 1 2 15 14 1 8 ...
```

```
Lakes$Lakes.Assessed.mi2 <- as.numeric(Lakes$Lakes.Assessed.mi2)
Lakes$Lakes.Assessed.percent <- as.numeric(Lakes$Lakes.Assessed.percent)
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':    50 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  30 1 10 4 14 31 12 2 15 28 ...
##  $ Lakes.Impaired.mi2         : num  42 2 31 39 35 4 27 20 45 40 ...
##  $ Lakes.Impaired.percent     : num  10 10 20 3 23 27 4 33 32 11 ...
##  $ Lakes.Impaired.percent.TMDL: num  12 17 20 16 1 2 15 14 1 8 ...
```

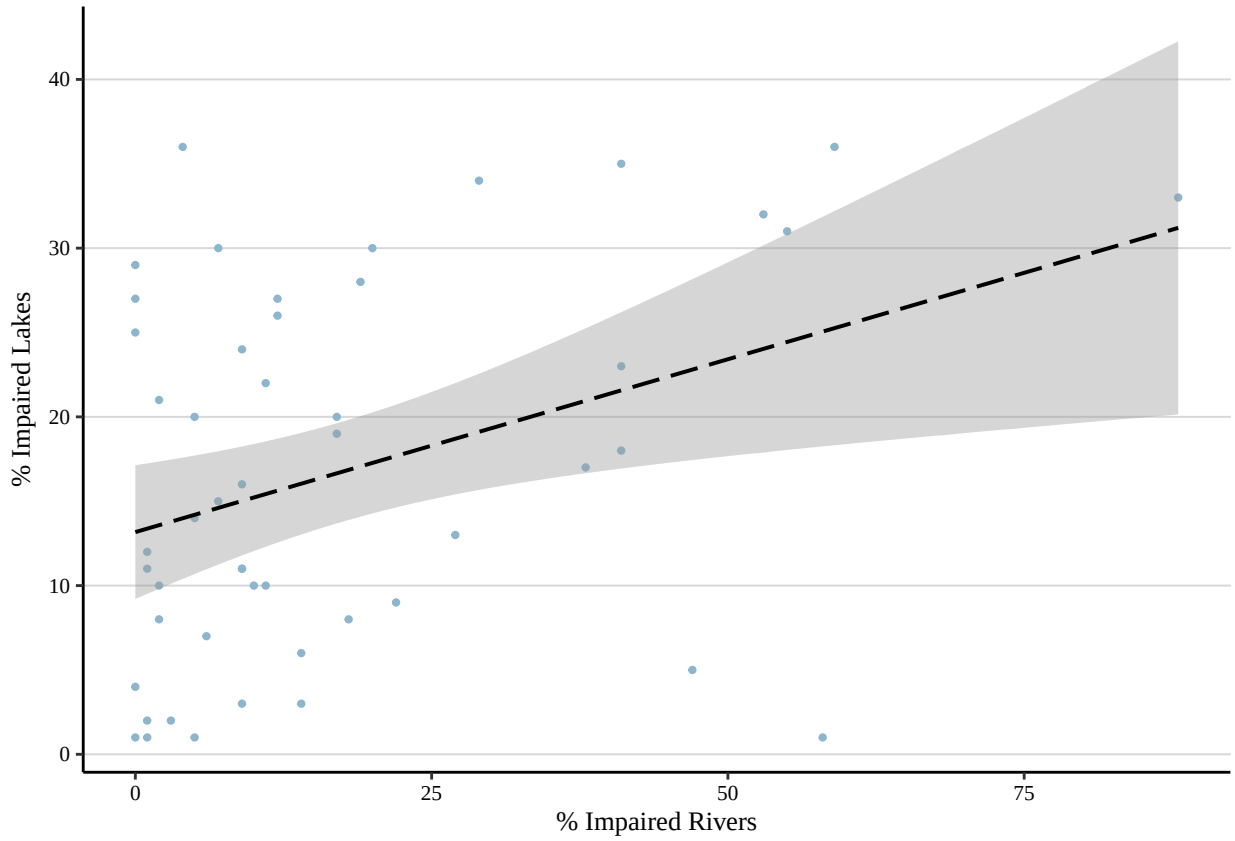
10. Join the two data frames with a `full_join`.

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

```
scat.waters <- ggplot(Waters) +
  geom_point(aes(x = Rivers.Impaired.percent, y = Lakes.Impaired.percent),
             color = "lightskyblue3", size = 0.8, alpha = 1) +
  geom_smooth(aes(x = Rivers.Impaired.percent, y = Lakes.Impaired.percent),
              method = lm,
              lty = 5, lwd = 0.7, color = "black") +
  #scale_x_continuous(limits = c(0, 50)) +
  labs(x = expression(paste("% Impaired Rivers")),
       y = expression(paste("% Impaired Lakes"))) +
  theme.hc01
plot(scat.waters)
```



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

    There is a general trend that states with higher percentage of impaired rivers also tend to have higher percentage of impaired lakes.