Assignment 10: Data Scraping

Kim Myers

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/Temp/Documents/Duke/S20/DataAnalytics/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
library(rvest)
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 3.6.3
```

```
mytheme <- theme_bw() +
  theme(axis.text = element_text(color = "dark gray"),
        legend.position = "right")
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$Rivers.Assessed.mi2 <- str_replace(Rivers$Rivers.Assessed.mi2,
                                          pattern = "([,])", replacement = "")
Rivers$Rivers.Assessed.percent <- str_replace(Rivers$Rivers.Assessed.percent,
                                          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
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)
```

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_te

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

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 != "Pennsylvania" & State != "Hawaii")

# 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$Lakes.Assessed.percent <- str_replace(Lakes$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$Lakes.Impaired.percent <- str_replace(Lakes$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$Lakes.Assessed.acre <- as.numeric(Lakes$Lakes.Assessed.acre)
```

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

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

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

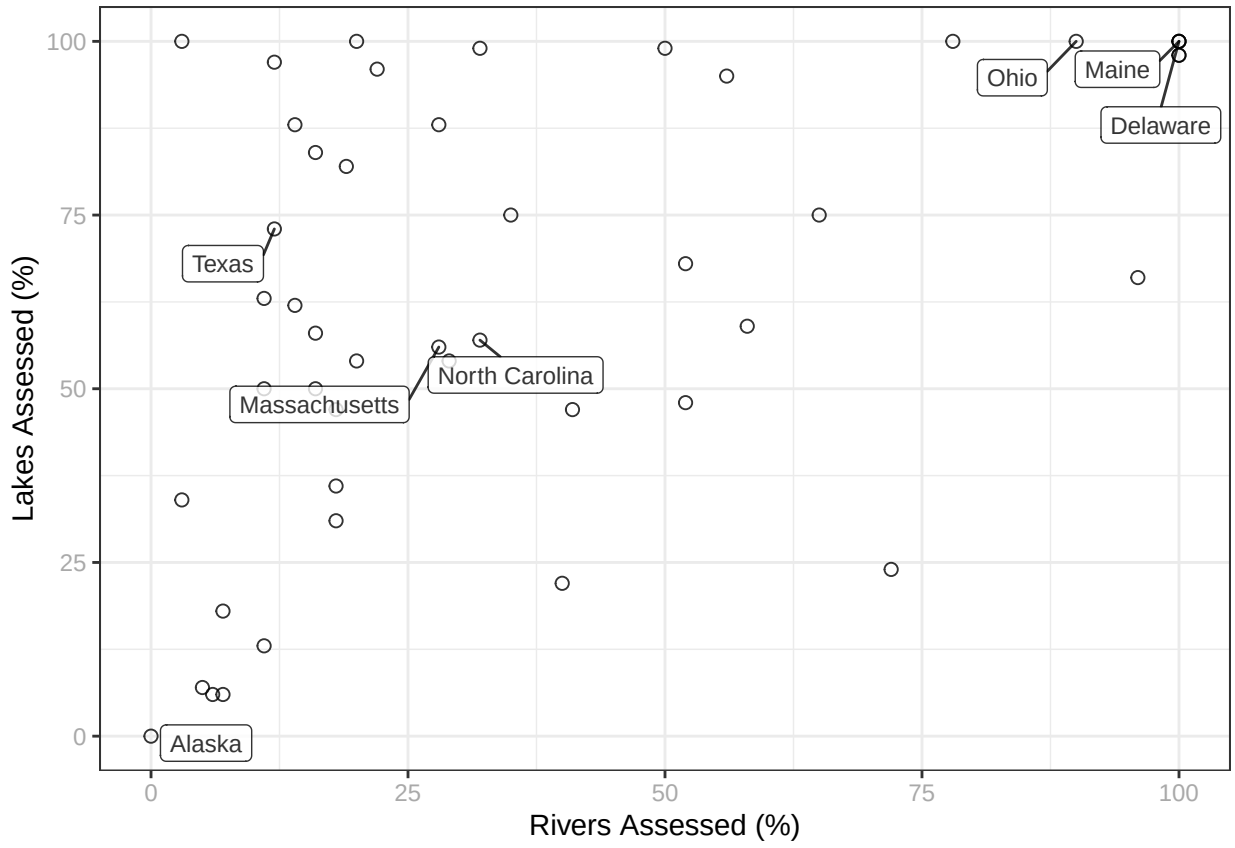
```
riverslakes <- full_join(Rivers,Lakes,"State")
```

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)

```
ggplot(riverslakes,aes(x=Rivers.Assessed.percent,y=Lakes.Assessed.percent)) +
  geom_point(shape = 21, size = 2, alpha = 0.8) +
  scale_fill_viridis_c(option = "inferno", begin = 0.2, end = 0.9, direction = -1) +
  geom_label_repel(data = subset(riverslakes, State %in% c("North Carolina", "Ohio", "Alaska", "Texas",
                   aes(label = State), nudge_x = -5, nudge_y = -5, size = 3, alpha = 0.8) +
  labs(x = "Rivers Assessed (%)", y = "Lakes Assessed (%)") +
  xlim(0,100) +
  ylim(0,100)
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
shapiro.test(riverslakes$Rivers.Assessed.percent)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  riverslakes$Rivers.Assessed.percent
## W = 0.85407, p-value = 1.997e-05
```

```
shapiro.test(riverslakes$Lakes.Assessed.percent)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  riverslakes$Lakes.Assessed.percent
## W = 0.94222, p-value = 0.01978
```

```
Assessed.kw <- kruskal.test(riverslakes$Rivers.Assessed.percent ~ riverslakes$Lakes.Assessed.percent)
Assessed.kw
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  riverslakes$Rivers.Assessed.percent by riverslakes$Lakes.Assessed.percent
## Kruskal-Wallis chi-squared = 37.312, df = 33, p-value = 0.2774
```

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

    Across all U.S. states, there is no significant difference between percent lakes assessed and percent

rivers assessed (Kruskal-Wallis, chi-squared=37.312, p-value=0.2774). In my initial hypothesis, I estimated that lakes would be more frequently sampled than rivers because, in most states, they are fewer in number. However, these findings suggest that states do not exhibit a sampling bias toward lakes or rivers.