Assignment 10: Data Scraping

Nikki Shintaku

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/nikkishintaku/Desktop/Environmental872/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
library(viridis)
library(rvest)
library(ggrepel)
library(ggthemes)

mytheme <- theme_stata(base_size = 14, base_family = "sans", scheme = "s2mono") +
  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-nu
as the URL to be scraped.

```r
#url for data scraping
url <- "https://www.epa.gov/nutrient-policy-data/waters-assessed-impaired-due-nutrient-related-causes"

# Reading the HTML code from the website
webpage <- read_html(url)
```

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

```r
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.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`.

```r
# 4 Remove non-numeric characters
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 Set to numeric
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.acres <- 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.acres <- 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.acres, Lakes.Assessed.percent,
                    Lakes.Impaired.acres, 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 Filter out States with no data
Lakes <- Lakes %>%
  filter(State != "Hawaii" & State != "Pennsylvania")

# 8 Remove non-numeric characters
Lakes$Lakes.Assessed.acres <- str_replace(Lakes$Lakes.Assessed.acres,
                                          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$Lakes.Impaired.acres <- str_replace(Lakes$Lakes.Impaired.acres,
                                          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 Set as numeric
Lakes$Lakes.Assessed.acres <- as.numeric(Lakes$Lakes.Assessed.acres)
```

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

```
Lakes$Lakes.Assessed.percent <- as.numeric(Lakes$Lakes.Assessed.percent)
Lakes$Lakes.Impaired.acres <- as.numeric(Lakes$Lakes.Impaired.acres)
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.acres       : num  431 5981 114976 64778 NA ...
##  $ Lakes.Assessed.percent     : num  88 0 34 13 50 95 47 100 54 82 ...
##  $ Lakes.Impaired.acres       : 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`.

```
#Full Join table of river and lakes
Rivers_lakes_combined <- full_join(Rivers, Lakes)
```

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

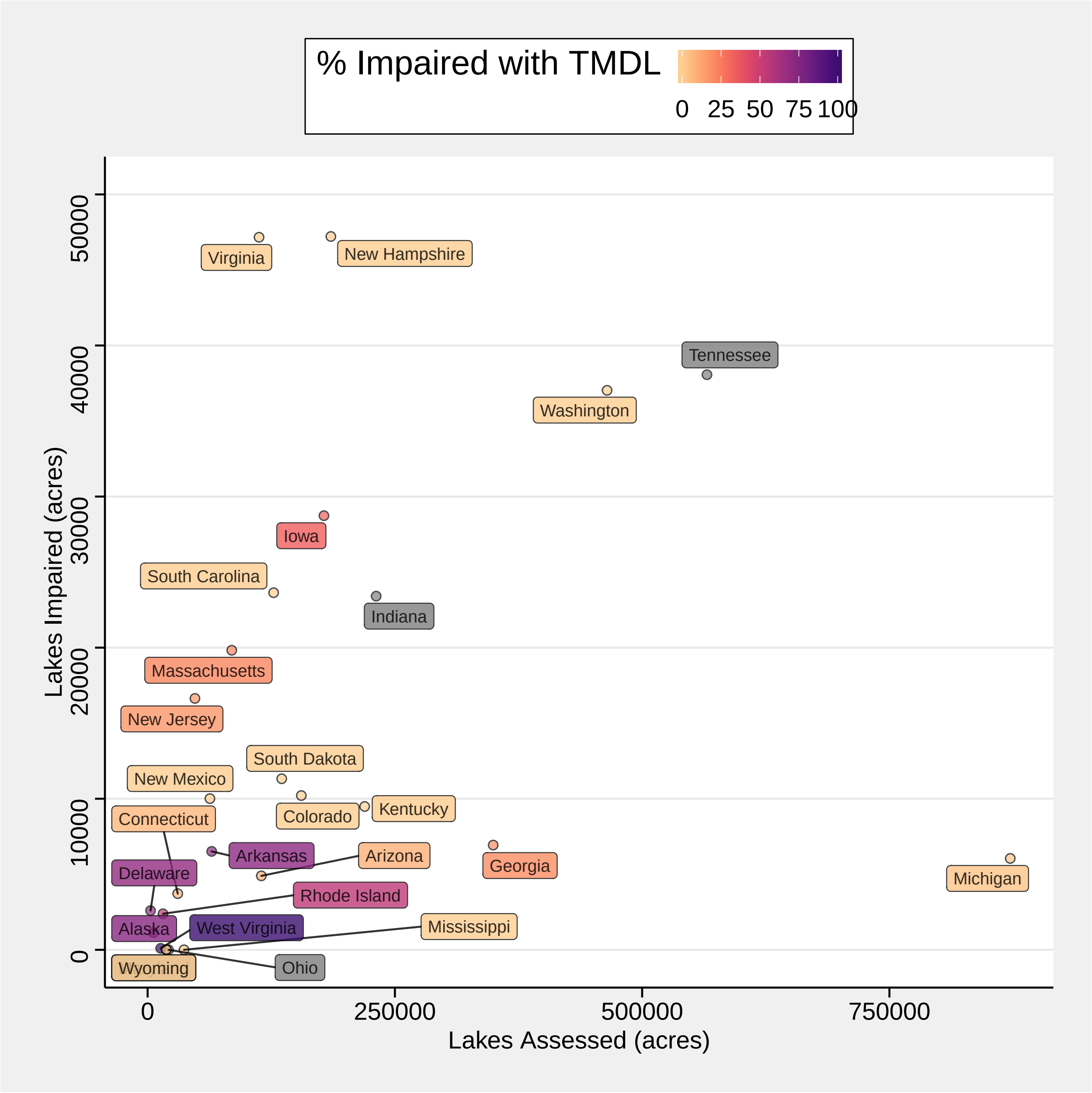
11. Create one graph that compares the data for lakes and/or rivers. This option is flexible; choose a relationship (or relationships) that seem interesting to you, and think about the implications of your findings. This graph should be edited so it follows best data visualization practices.

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

```
ggplot(Rivers_lakes_combined, aes(x = Lakes.Assessed.acres, y = Lakes.Impaired.acres,
                                  fill = Lakes.Impaired.percent.TMDL)) +
  geom_point(shape = 21, size = 2, alpha = 0.7) +
  scale_fill_viridis(option = "magma", begin = 0.2, end = 0.9, direction = -1) +
  ylim(0, 50000) +
  geom_label_repel(aes(label = State), nudge_x = -4, nudge_y = -4,
                   size = 3, alpha = 0.8) +
  labs(x = "Lakes Assessed (acres)",
       y = "Lakes Impaired (acres)",
       fill = "% Impaired with TMDL")
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```

```
## Warning: Removed 24 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.

Lakes in states that have low assessed acres also have low assessed impaired acres. However, these lakes have a higher percent impaired with TMDL. These findings suggest that more lakes need to be assessed in order to determine if the lake is impaired, and those will affect the percent impaired with TMDL too. In addition, the data holds a lot of NAs in percent impaired with TMDL so data collection is needed for the Lakes dataset to be accurate.