Assignment 10: Data Scraping

Monisha Eadala

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

```
# To check your working directory
getwd()
```

```
## [1] "/Users/monishaeadala/Environmental_Data_Analytics_2020/Lessons/sf-lesson-20200303"
```

```
# To load the necessary packages
library(tidyverse)
library(viridis)
#install.packages("rvest")
library(rvest)
#install.packages("ggrepel")
library(ggrepel)
```

```
# To set my ggplot theme
mytheme <- theme_classic(base_size = 14) +
  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.

```
# To specify website to be scraped
url <- "https://www.epa.gov/nutrient-policy-data/waters-assessed-impaired-
due-nutrient-related-causes"

# To read 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.

```
# To scrape the Rivers table, with every column except year
State <- 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()

# To turn it into a data frame
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
# To 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
# To make sure R knows that the numeric columns are numbers
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.

```
# To scrape the Lakes table, with every column except year
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_text()

# To turn it into a data frame
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
# To filter out states with no data
Lakes <- Lakes %>%
  filter(State != "Hawaii" & State != "Pennsylvania")

# 8
# To remove non-numeric characters
Lakes$Lakes.Assessed.mi2 <- str_replace(Lakes$Lakes.Assessed.mi2,
                                                    pattern = "([,])",
replacement = "")
Lakes$Lakes.Assessed.mi2 <- str_replace(Lakes$Lakes.Assessed.mi2,
                                                    pattern = "([,])",
replacement = "")
Lakes$Lakes.Assessed.mi2 <- str_replace(Lakes$Lakes.Assessed.mi2,
                                                    pattern = "([.])",
replacement = "") # There seems to be mistake with the Alabama data; so we
are removing the "." which was supposed to be ","
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$Lakes.Impaired.mi2 <- str_replace(Lakes$Lakes.Impaired.mi2,
                                                    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
# To make sure R knows that the numeric columns are numbers
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         : chr  "430976" "5981" "114976" "64778" ...
##  $ Lakes.Assessed.percent     : chr  "88" "0" "34" "13" ...
##  $ Lakes.Impaired.mi2         : chr  "81740" "1137" "4895" "6513" ...
##  $ Lakes.Impaired.percent     : chr  "19" "19" "4" "10" ...
##  $ Lakes.Impaired.percent.TMDL: chr  "53" "73" "9" "71" ...
```

```
Lakes$Lakes.Assessed.mi2 <- as.numeric(Lakes$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':    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  430976 5981 114976 64778 1051246 ...
##  $ Lakes.Assessed.percent     : num  88 0 34 13 50 95 47 100 54 82 ...
##  $ Lakes.Impaired.mi2         : 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`.

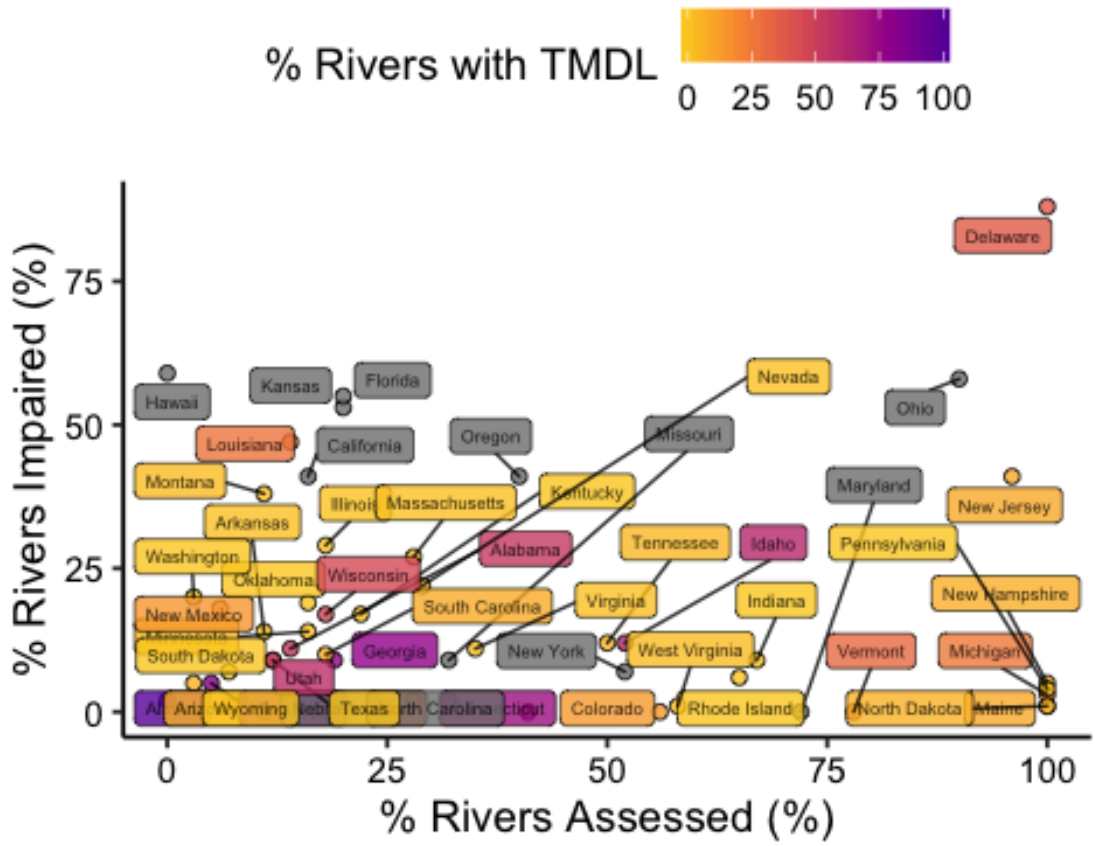
```
# To join the two data frames
RiversnLakes <- full_join(Rivers, Lakes)
```

```
## 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)

```
# To create a graph that compares the relationship between the rivers
imparied and rivers assessed across the states
ggplot(Rivers, aes(x = Rivers.Assessed.percent,
                   y = Rivers.Impaired.percent, fill =
Rivers.Impaired.percent.TMDL)) +
  geom_point(shape = 21, size = 2, alpha = 0.8) +
  scale_fill_viridis_c(option = "plasma", begin = 0.2, end = 0.9, direction =
-1) +
  geom_label_repel(aes(label = State), nudge_x = -5, nudge_y = -5,
                   size = 2, alpha = 0.8) +
   labs(x = "% Rivers Assessed (%)",
        y = "% Rivers Impaired (%)",
        fill = "% Rivers with TMDL")
```




```
cor(Rivers$Rivers.Assessed.percent, Rivers$Rivers.Impaired.percent) # Gives
us a correlation value between -1 and 1

## [1] -0.01607445
```

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

From the graph, we can tell that: 1. There are more number of states between 0-25% rivers assessed, and more numbers of states between 0-25% rivers impaired. 2. There are fewer number of states with over 50% of their rivers assessed, and even fewer states with over 50% of their rivers impaired. 3. Delaware seems to be the only state at more than 75% if its rivers assessed and also impaired. 4. There doesn't seem to be a strong correlation between the % of rivers impaired and the % of rivers assessed. 5. More states with higher % of rivers impaired have lower % of their rivers covered under TMDL or are marked NA; while most states with lower % of rivers impaired have more % of them covered under TMDL. Therefore, there seems to be a negative correlation between % of rivers impaired and % of rivers with TMDL. Similarly, more states with higher % of rivers assessesed have lower % of rivers with TMDL, while more number of states with lower % of rivers assesssed have higher % of rivers with TMDL. Additionally, the correlation value -0.016 tells us that there is an extremely poor/low but possibly negative correlation between % of rivers impaired and % of rivers assessed.