Assignment 10: Data Scraping

Vicky Jia

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/Victoria/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
## Warning: package 'forcats' was built under R version 3.5.2
```

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.5.2
```

```
## Warning: package 'xml2' was built under R version 3.5.2
```

```
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-nutr 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.I
```

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)
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_t

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$Lakes.Assessed.mi2 <- str_replace(Lakes$Lakes.Assessed.mi2,
                                        pattern = "([,])", replacement = "")
Lakes$Lakes.Assessed.mi2 <- str_replace(Lakes$Lakes.Assessed.mi2,
                                        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$Lakes.Impaired.mi2 <- str_replace(Lakes$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
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  431 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`.

```
ImpairedWater <- 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)

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 3.5.2
```

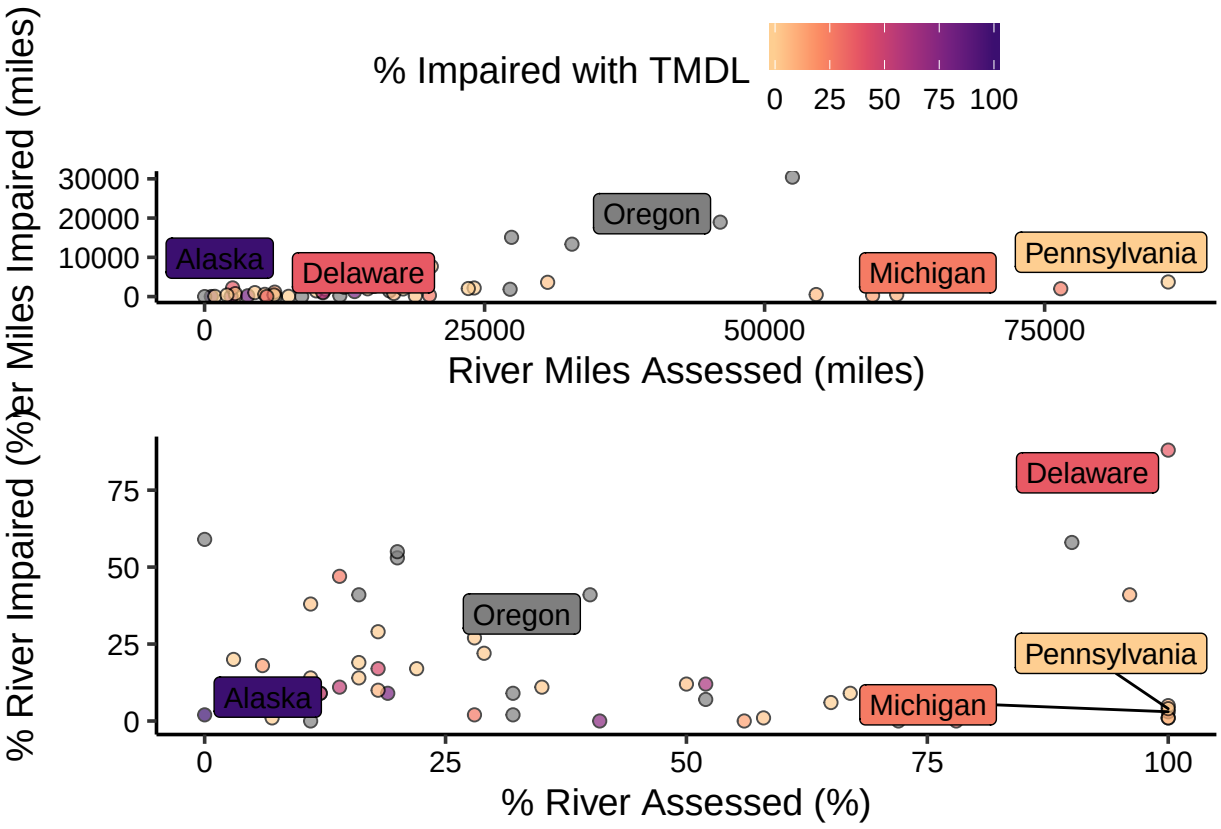
```
library(cowplot)
```

```
##
## ************************************************************
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## ************************************************************
```

```
number <- ggplot(Rivers, aes(x = Rivers.Assessed.mi2, y = Rivers.Impaired.mi2,
                    fill = Rivers.Impaired.percent.TMDL)) +
  geom_point(shape = 21, size = 2, alpha = 0.7) +
  scale_fill_viridis_c(option = "magma", begin = 0.2, end = 0.9, direction = -1) +
  geom_label_repel(data = subset(Rivers, State %in% c("Oregon", "Pennsylvania", "Michigan", "Delaware",
              aes(label = State), nudge_x = -500, nudge_y = 200) +
   labs(x = expression("River Miles Assessed (miles)"),
        y = expression("River Miles Impaired (miles)"),
        fill = "% Impaired with TMDL")

percent <- ggplot(Rivers, aes(x = Rivers.Assessed.percent, y = Rivers.Impaired.percent,
                    fill = Rivers.Impaired.percent.TMDL)) +
  geom_point(shape = 21, size = 2, alpha = 0.7) +
  scale_fill_viridis_c(option = "magma", begin = 0.2, end = 0.9, direction = -1) +
  geom_label_repel(data = subset(Rivers, State %in% c("Oregon", "Pennsylvania", "Michigan", "Delaware",
              aes(label = State), nudge_x = -5, nudge_y = -5) +
   labs(x = expression("% River Assessed (%)"),
        y = expression("% River Impaired (%)"),
        fill = "% Impaired with TMDL")

plot_grid(number + theme(legend.position = "top"),
          percent + theme(legend.position = "none"),
          nrow = 2, align = 'v')
```



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

    This graph shows the relationship between imparied rivers and assessed rivers to indicate the intensity of river impairment among all rivers assessed. Impairment results are based on rivers assessed, so impaired river values only illustrate results for sample rivers, but do not represent the real conditions for all rivers in a certain state. From the top graph, we can only tell the exact distance values of assessed rivers and impaired rivers. For instance, most states only assess less than 25,000 miles of rivers. From this graph, we cannot draw a big picture for assessed rivers; that says, we hardly describe these results under the macroscale for each state. The bottom graph describes these values in a percentage format, which indicates how these results are compared with all rivers in the states. Looking at both graphs together, we can tell the assessment rate for most states are low, and for all assessed river, less than half rivers are impaired. One good example obtained by looking at results in both formats is Delaware. While the actual distance of rivers assess is low in Delaware, the percentage value is high, which indicates there are not too many rivers in Delaware. The impairment results show most rivers, even short, have been impaired. This illustrates the importance of looking at two graphs together. For other results, Alaska is an opposite example from Delaware with all low values, which is hard to describe the significance of assessment results. However, it has a very high percentage of total maximum daily load (TMDL) to reduce pollutant loadings and restore the waterbody, which indicates the extent of water impairment and also states' effective efforts to address impairment criteria for the restoration. Some other examples, Michigan and Pennsylvania, have most rivers assesed and few impaired rivers. However, these two states have low values of TMDL, which might indicate not-so effective states' efforts on addressing numeric criteria of impairment. Oregon indicates a normal relationship between assessment and impairement for most states, which are medium assesment with medium impairment but no data for TMDL.