Assignment 10: Data Scraping

Rachel Gonsenhauser

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

```r
#getwd()
library(tidyverse)
library(viridis)
#install.packages("rvest")
library(rvest)
#install.packages("ggrepel")
library(ggrepel)

mytheme <- theme_classic() +
  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.

```r
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.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$Rivers.Assessed.mi2 <- str_replace(Rivers$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
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
Lakes <- Lakes %>%
  filter(State != "Hawaii" & State != "Pennsylvania")

# 8
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
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  "430.976" "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)
```

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

```
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 NA ...
##  $ 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`.

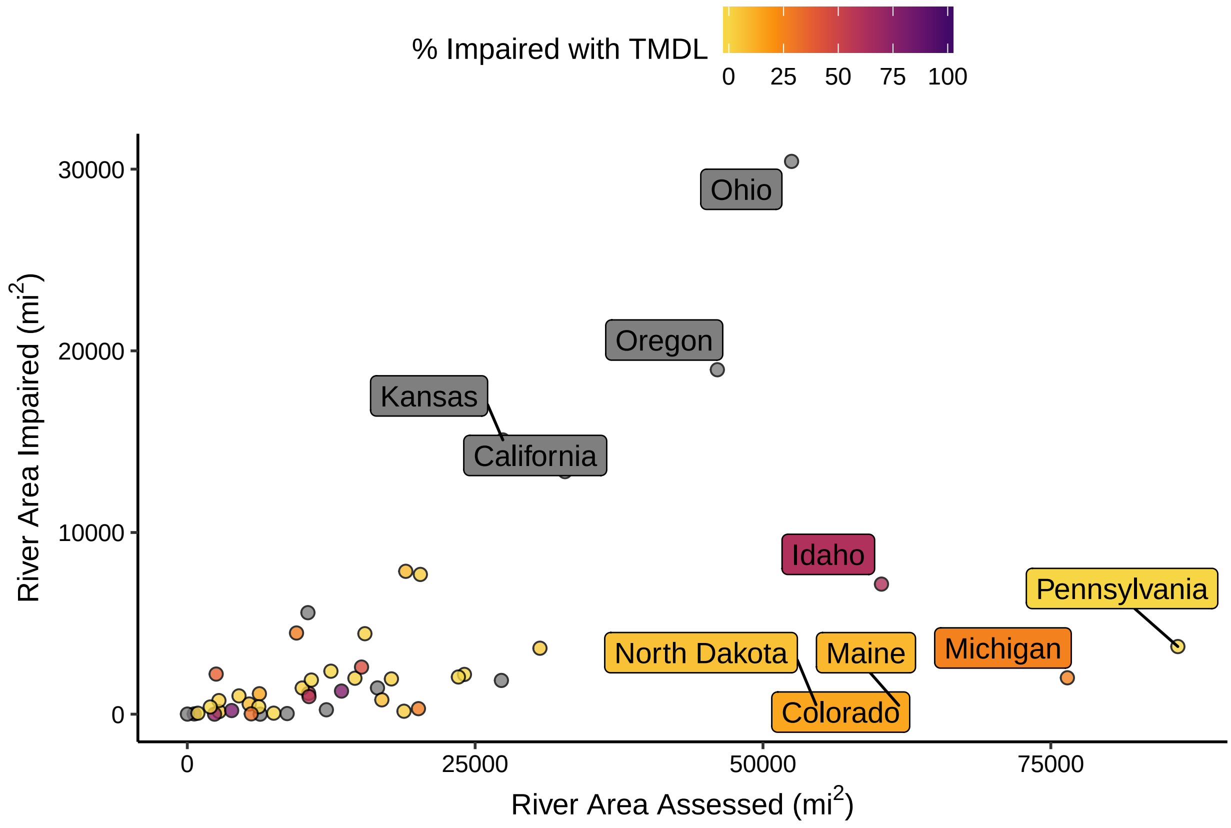
```
Rivers.and.Lakes <- full_join(Rivers, Lakes)
```

```
## 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)

```
# Rivers graph (Figure 1)
ggplot(Rivers.and.Lakes, aes(x = Rivers.Assessed.mi2, y = Rivers.Impaired.mi2,
                             fill = Rivers.Impaired.percent.TMDL)) +
  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(Rivers.and.Lakes , State %in% c("Ohio", "Oregon", "California", "Idaho
  aes(label = State), nudge_x = -500, nudge_y = 200) +
  labs(x = expression("River Area Assessed (mi"^"2"*")"),
   y = expression("River Area Impaired (mi"^"2"*")"),
  fill = "% Impaired with TMDL")
```

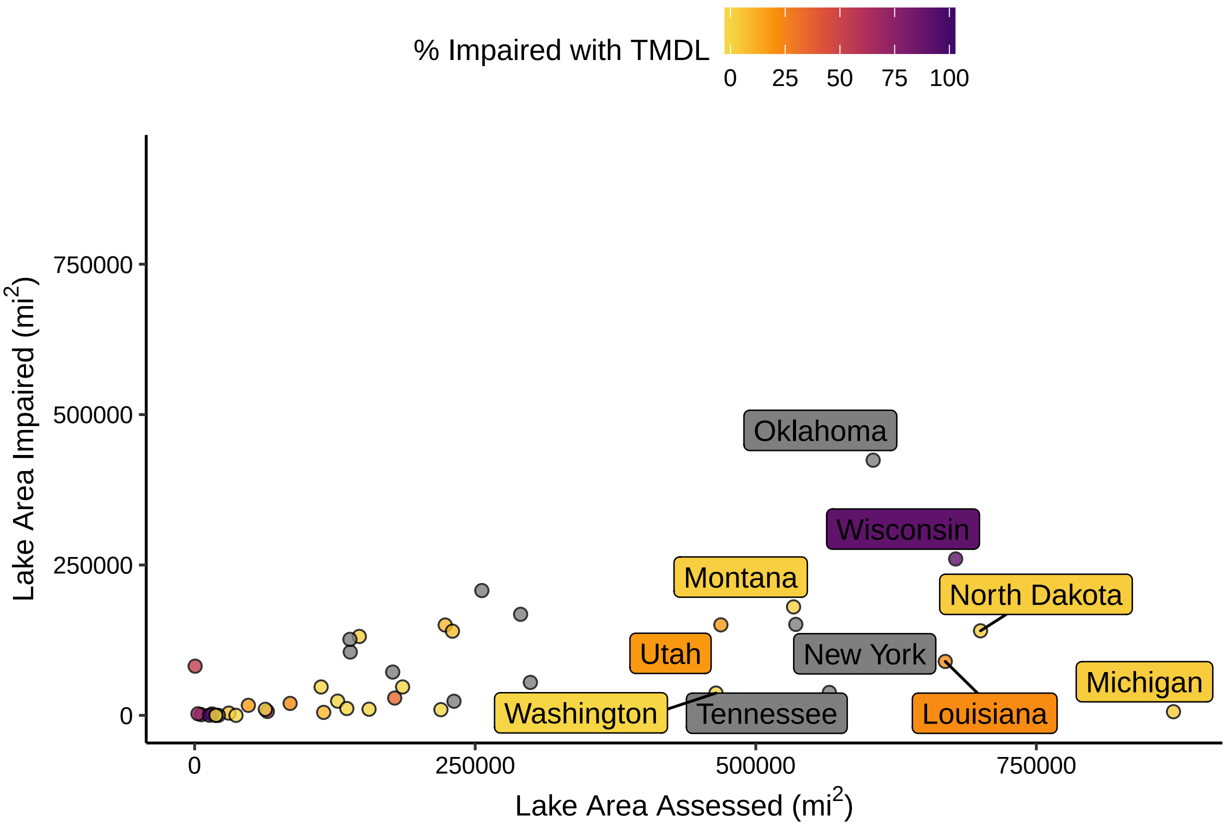


```r
# Rivers simple linear regression
Rivers.regression <- lm(data = Rivers.and.Lakes, Rivers.Impaired.mi2 ~ Rivers.Assessed.mi2 + Rivers.Impa
summary(Rivers.regression)
```

```
##
## Call:
## lm(formula = Rivers.Impaired.mi2 ~ Rivers.Assessed.mi2 + Rivers.Impaired.percent.TMDL,
##     data = Rivers.and.Lakes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2585.2 -1191.8  -422.8   215.4  6025.8
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1389.78579  525.77907   2.643   0.0121 *
## Rivers.Assessed.mi2             0.02586    0.01529   1.691   0.0994 .
## Rivers.Impaired.percent.TMDL   -4.69516   12.11993  -0.387   0.7007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2048 on 36 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.08217,    Adjusted R-squared:  0.03118
## F-statistic: 1.611 on 2 and 36 DF,  p-value: 0.2137
```

```
# Lakes graph (Figure 2)
ggplot(Rivers.and.Lakes, aes(x = Lakes.Assessed.mi2, y = Lakes.Impaired.mi2,
                             fill = Lakes.Impaired.percent.TMDL)) +
  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(Rivers.and.Lakes, State %in% c("Oklahoma", "Wisconsin", "North Dakota"
  aes(label = State), nudge_x = -500, nudge_y = 200) +
  labs(x = expression("Lake Area Assessed (mi"^"2"*")"),
   y = expression("Lake Area Impaired (mi"^"2"*")"),
  fill = "% Impaired with TMDL")
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```



```
# Lakes simple linear regression
Lakes.regression <- lm(data = Rivers.and.Lakes, Lakes.Impaired.mi2 ~ Lakes.Assessed.mi2 + Lakes.Impaire
summary(Lakes.regression)
```

```
##
## Call:
## lm(formula = Lakes.Impaired.mi2 ~ Lakes.Assessed.mi2 + Lakes.Impaired.percent.TMDL,
##     data = Rivers.and.Lakes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -145661  -30230  -12077   18536  112022
##
## Coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 8.589e+03  1.708e+04   0.503 0.619071
## Lakes.Assessed.mi2          1.630e-01  4.388e-02   3.715 0.000898 ***
## Lakes.Impaired.percent.TMDL 3.208e+02  3.558e+02   0.902 0.374866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57890 on 28 degrees of freedom
##   (19 observations deleted due to missingness)
## Multiple R-squared:  0.3317, Adjusted R-squared:  0.2839
## F-statistic: 6.948 on 2 and 28 DF,  p-value: 0.003546
```

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

    Neither the area of rivers assessed nor the percent of impaired rivers that have all impairments addressed by a TMDL/other restoration plan are significant indicators of the area of rivers impaired (Figure 1; multiple linear regression: R2=0.03118, df=2 and 36, p=0.2137). Conversely, while the percent of impaired rivers that have all impairments addressed by a TMDL/other restoration plan is not a significant indicator of the area of rivers impaired, the area of rivers assessed is a significant indicator of the area found to be impaired (Figure 2; multiple linear regression: R2=0.2839, df=2 and 28, p=0.003546). Many states were missing observations for the percent of impaired rivers or lakes whose impairments are addressed by a TMDL/other restoration plan, so further data collection in these states is necessary to establish whether a relationship between this variable and others studied exists.