Web Scraping

Andrew Nguyen

First, we'll load the libraries necessary to read the HTML file and convert the data into tibbles.

```
library(rvest)
```

```
## Loading required package: xml2
```

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

We can now use the read_html function to extract the HTML from a webpage. For this assignment, we will take a CSU Chico class schedule.

```
CSCISPR2019 <- read_html("http://ems.csuchico.edu/APSS/schedule/spr2019/CSCI.shtml")
```

We will now take the HTML and divide it into an entire set of tibble-able data nodes that we will continually process:

```
entiredata <- CSCISPR2019 %>%
              html_nodes(".classrow")
```

Using the entiredata nodes, process further into smaller nodes and converge all into a tibble. According to the assignment document, we must: identify the nodes that contain [at least] the class number [subj and cat num columns], section number [sect], course title [Title], instructor [Instructor], and enrollment [Tot enrl]

```
subj <- entiredata %>%
        html_nodes("td.subj") %>%
        html_text()

cat_num <- entiredata %>%
        html_nodes("td.cat_num") %>%
        html_text()

sect <- entiredata %>%
        html_nodes("td.sect") %>%
        html_text()

Tot_enrl <- entiredata %>%
        html_nodes("td.enrtot") %>%
        html_text()

title <- entiredata %>%
        html_nodes("td.title") %>%
        html_text()

instructor <- entiredata %>%
        html_nodes("td.Instructor") %>%
        html_text()

extable <- tibble(subj= subj,
        cat_num = cat_num,
        title = title,
        sect = sect,
        instructor = instructor,
        Tot_enrl= Tot_enrl)
```

We now have a successful, relatively clean table version of the webpage. We can take what we just did and make a universal function out of it. Thus, a function with similar but generic coding as above should do:

```
make_class_schedule <- function (url) {
  html <- read_html(url)

  entiredata <- html %>%
              html_nodes(".classrow")

  subj <- entiredata %>%
        html_nodes("td.subj") %>%
        html_text()

  cat_num <- entiredata %>%
          html_nodes("td.cat_num") %>%
          html_text()

  sect <- entiredata %>%
          html_nodes("td.sect") %>%
          html_text()

  Tot_enrl <- entiredata %>%
          html_nodes("td.enrtot") %>%
          html_text()

  title <- entiredata %>%
          html_nodes("td.title") %>%
          html_text()

  sect <- entiredata %>%
          html_nodes("td.sect") %>%
          html_text()
  instructor <- entiredata %>%
          html_nodes("td.Instructor") %>%
          html_text()

  table <- tibble(subj= subj,
          cat_num = cat_num,
          title = title,
          sect = sect,
          instructor = instructor,
          Tot_enrl= Tot_enrl)

  return (table)
}
```

Let's test it out for good measure, by taking the Spring 2020 schedule for computer science classes and making a table called "Spring2020CSCISched".

```
Spring2020CSCISched <- make_class_schedule("http://ems.csuchico.edu/APSS/schedule/spr2020/CSCI.shtml")
head(Spring2020CSCISched, n=10)
```

```
## # A tibble: 10 x 6
##    subj  cat_num title                      sect  instructor        Tot_enrl
##    <chr> <chr>   <chr>                      <chr> <chr>             <chr>
##  1 CSCI  101     Intro to Computer Science  01    Herring,Brian D   1
##  2 CSCI  102     Living With Technology     01    Harris,Keith S    0
##  3 CSCI  111     Programming and Algorith~  02    Gibson,Todd A     4
##  4 CSCI  111     Programming and Algorith~  04    Renner,Renee S    3
##  5 CSCI  111     Programming and Algorith~  06    Renner,Renee S    2
##  6 CSCI  211     Programming and Algorith~  02    Herring,Brian D   6
##  7 CSCI  211     Programming and Algorith~  04    Juliano,Bienveni~ 3
##  8 CSCI  211     Programming and Algorith~  06    Juliano,Bienveni~ 1
##  9 CSCI  301W    Comp's Impact on Society~  01    Hubbard,Susan K   3
## 10 CSCI  311     Algorithms and Data Stru~  01    Challinger,Judit~ 1
```

It works! We are assigned now to take the rest of the assigned websites into tibbles and make all of our tibbles into a single one. The tables having similar column names means daisy chaining is a mere wormy task.

```
Spring2019CSCISched <- make_class_schedule("http://ems.csuchico.edu/APSS/schedule/spr2019/CSCI.shtml")
Spring2019MATHSched <- make_class_schedule("http://ems.csuchico.edu/APSS/schedule/spr2019/MATH.shtml")
Spring2020MATHSched <- make_class_schedule("http://ems.csuchico.edu/APSS/schedule/spr2020/MATH.shtml")

# Use RBind to join all previously named tables of choice
singulartable <- rbind(Spring2019CSCISched, Spring2019MATHSched, Spring2020CSCISched, Spring2020MATHSche
head (singulartable, n=10)
```

```
## # A tibble: 10 x 6
##    subj  cat_num title                      sect  instructor        Tot_enrl
##    <chr> <chr>   <chr>                      <chr> <chr>             <chr>
##  1 CSCI  101     Intro to Computer Science  01    " "               0
##  2 CSCI  102     Living With Technology     01    Juliano,Bienveni~ 26
##  3 CSCI  111     Programming and Algorith~ 02    Gibson,Todd A     29
##  4 CSCI  111     Programming and Algorith~ 04    Raigoza,Jaime A   49
##  5 CSCI  111     Programming and Algorith~ 06    Raigoza,Jaime A   19
##  6 CSCI  211     Programming and Algorith~ 02    Donnelly,Patrick~ 27
##  7 CSCI  211     Programming and Algorith~ 04    Juliano,Bienveni~ 34
##  8 CSCI  211     Programming and Algorith~ 06    Juliano,Bienveni~ 14
##  9 CSCI  301W    Comp's Impact on Society~ 01    Hubbard,Susan K   29
## 10 CSCI  311     Algorithms and Data Stru~ 01    Challinger,Judit~ 53
```