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)
```

s...	cat_n...	title	s...	instructor	Tot_enrl
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>

s... <chr>	cat_n... <chr>	title <chr>	s... <chr>	instructor <chr>	Tot_enrl <chr>
CSCI	101	Intro to Computer Science	01	Herring,Brian D	1
CSCI	102	Living With Technology	01	Harris,Keith S	0
CSCI	111	Programming and Algorithms I	02	Gibson,Todd A	4
CSCI	111	Programming and Algorithms I	04	Renner,Renee S	3
CSCI	111	Programming and Algorithms I	06	Renner,Renee S	2
CSCI	211	Programming and Algorithms II	02	Herring,Brian D	7
CSCI	211	Programming and Algorithms II	04	Juliano,Bienvenido A	3
CSCI	211	Programming and Algorithms II	06	Juliano,Bienvenido A	1
CSCI	301W	Comp's Impact on Society (W)	01	Hubbard,Susan K	3
CSCI	311	Algorithms and Data Structures	01	Challinger,Judith A	1

1-10 of 10 rows

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.shtm
l")
Spring2020MATHSched <- make_class_schedule("http://ems.csuchico.edu/APSS/schedule/spr2020/MATH.shtm
l")

# Use RBind to join all previously named tables of choice
singulartable <- rbind(Spring2019CSCISched, Spring2019MATHSched, Spring2020CSCISched, Spring2020MATHS
ched)
head (singulartable, n=10)
```

s... <chr>	cat_n... <chr>	title <chr>	s... <chr>	instructor <chr>	Tot_enrl <chr>
CSCI	101	Intro to Computer Science	01		0
CSCI	102	Living With Technology	01	Juliano,Bienvenido A	26
CSCI	111	Programming and Algorithms I	02	Gibson,Todd A	29
CSCI	111	Programming and Algorithms I	04	Raigoza,Jaime A	49
CSCI	111	Programming and Algorithms I	06	Raigoza,Jaime A	19
CSCI	211	Programming and Algorithms II	02	Donnelly,Patrick J	27
CSCI	211	Programming and Algorithms II	04	Juliano,Bienvenido A	34
CSCI	211	Programming and Algorithms II	06	Juliano,Bienvenido A	14

s…	cat_n…	title	s…	instructor	Tot_enrl
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
CSCI	301W	Comp's Impact on Society (W)	01	Hubbard,Susan K	29
CSCI	311	Algorithms and Data Structures	01	Challinger,Judith A	53

1-10 of 10 rows