Scraping_Schedules

Aaron Nicanor

##Libraries

First, I'll begin loading in all the libraries I'll be using in this assignment.

```
suppressMessages(library("rvest"))
suppressMessages(library("dplyr"))
suppressMessages(library("tidyr"))
suppressMessages(library("stringr"))
```

##URLs

I'll now load in all the URLs that I'll be working with.

```
CSCI_S_19 <- "http://ems.csuchico.edu/APSS/schedule/spr2019/CSCI.shtml"
CSCI_S_20 <- "http://ems.csuchico.edu/APSS/schedule/spr2020/CSCI.shtml"
MATH_S_19 <- "http://ems.csuchico.edu/APSS/schedule/spr2019/MATH.shtml"
MATH_S_20 <- "http://ems.csuchico.edu/APSS/schedule/spr2020/MATH.shtml"
```

##Universal Scraping Function

Here I'll begin to code a function that will grab information from a given URL. Note that every other row is labeled as an alt row, so I'll be grabbing them seperately and merging it with the regular rows later on.

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

  #Grabbing all odd rows in the schedule. Will merge later
  OddRow <- html %>% html_nodes(".classrow")

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

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

  section_num_odd <- OddRow %>%
    html_nodes("td.sect") %>%
    html_text() %>%
    as.integer()

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

  enrollment_odd <- OddRow %>%
    html_nodes("td.enrtot") %>%
    html_text() %>%
    as.integer()

  instructor_odd <- OddRow %>%
  html_nodes("td.Instructor") %>%
  html_text()

#Grabbing all even rows in the schedule. Will merge later
EvenRow <- html %>% html_nodes(".classrowalt")

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

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

section_num_even <- EvenRow %>%
  html_nodes("td.sect") %>%
  html_text() %>%
  as.integer()

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

enrollment_even <- EvenRow %>%
  html_nodes("td.enrtot") %>%
  html_text() %>%
  as.integer()

instructor_even <- EvenRow %>%
  html_nodes("td.Instructor") %>%
  html_text()

#EXTRA: Grabbing year and semester
Semester_Year <- html %>%
  html_nodes(".subjpagessubjheader") %>%
  html_text()

#Isolate a phrase within this string that has the pattern of
#a word (with both capital and lowercase letters) followed by a number
#In our case, it'll grab our semester and year
My_Semester_Year <- str_extract(Semester_Year,"[A-z,a-z]+ [0-9]+")

#Creating a tibble of all odd tables
odd_table <- tibble(semester_year=My_Semester_Year,
                    subject=subject_odd,
                    course_num=course_num_odd,
                    section_num=section_num_odd,
                    course_title=course_title_odd,
                    instructor=instructor_odd,
                    enrollment=enrollment_odd)

#Creating a tibble of all even tables
even_tables <- tibble(semester_year=My_Semester_Year,
                        subject=subject_even,
                        course_num=course_num_even,
                        section_num=section_num_even,
                        course_title=course_title_even,
                        instructor=instructor_even,
                        enrollment=enrollment_even)

  #Combining the two tables together
  full_table <- bind_rows(odd_table,even_tables)

  #Combine subject with course number into class number
  full_table$class_num <- paste(full_table$subject, full_table$course_num)

  #Split semester and year into their own columns
  full_table <- separate(full_table, semester_year, into= c("semester","year"), sep= " ")

  #Remove extra columns subject and year (merged into their seperate column by this point)
  full_table <- select(full_table, -subject, -course_num)

  #Move class number to where it should be on the table
  full_table <- full_table[,c(1,2,7,3,4,5,6)]

  return(full_table)
}
```

##Table Creation

Using the function I had just created, I'll made tables with all the URLs I had saved earlier. In addition, I'll compile all the tables I have created into a single table.

```
CSCI_19_Table <- Scrape_Class_Schedule(url = CSCI_S_19)

CSCI_20_Table <-  Scrape_Class_Schedule(url = CSCI_S_20)

MATH_19_Table <-  Scrape_Class_Schedule(url = MATH_S_19)

MATH_20_Table <-  Scrape_Class_Schedule(url = MATH_S_20)

Overall_Table <- rbind(CSCI_19_Table, CSCI_20_Table, MATH_19_Table, MATH_20_Table)
```

Now I can check my work by seeing all the tables we've created.

```
View(CSCI_19_Table)
View(CSCI_20_Table)
View(MATH_19_Table)
View(MATH_20_Table)
View(Overall_Table)
```

Everything seems to check out!