lab6Q2.R

mirrien

2022-03-15

```r
library(stringr)

##################################################
################ Helper Functions ###############
##################################################

# Course name (e.g., STAT 240)

# Generalize the operations:
# takes course page as argument and generates corresponding cn

extract_cn <- function(course_page) {
  return(trimws(str_extract(
    trimws(
      gsub("<[^<>]+>"," ",
           gsub("<span>.*</span>","",
                grep('<h1\\sid=\"name\"',course_page, v=T)
           )
      )
    ),
    '\\s[A-Z]+.*')
  )
  )
}

# When a course_page cannot be located, obtain cn from SFU Calendar:

extract_cn2 <- function(course_page) {
  index = grep('<small class=\"course_num',course_page)
  cn = trimws(course_page[(index+1):(index+2)])
  cn = paste(cn[1],cn[2],collapse = " ")
  return(cn)
}


##################################################
# Course title (e.g., Introduction to Data Science)

# Generalize the operations:

extract_tt <- function(course_page) {
  return(trimws(course_page[grep('<h2\\sid=\"title\"',course_page)+1]))
}

# When a course_page cannot be located, obtain tt from SFU Calendar:

extract_tt2 <- function(course_page) {
  tt = trimws(course_page[grep('<small class=\"course_num',course_page)-1])
  return(tt)
}

##################################################
# Instructor

# Generalize the above operations:

extract_ins <- function(course_page) {
  return(gsub("<[^<>]+>","",
              trimws(course_page[grep('<h4>Ins', course_page)+1])
  )
  )
}

##################################################
# Course Times + Location

# Generalize the above operations:

extract_ctl <- function(course_page) {
  # Assume there are two elements: Extract two lines
  ctl = course_page[(grep('<h4>Course', course_page)+1):
                      (grep('<h4>Course', course_page)+2)]

  # if there ARE 2 elements of course times, store as a vector of 2 elements
  if (str_detect(ctl[2],"\\d{1}:\\d{2}")) {
    ctl = trimws(str_replace_all(str_replace_all(ctl,"<br>", " "),
                                 "\\&ndash;",
                                 "-"))
    ctl = gsub("<[^<>]+>","",ctl)
  } else {  # if there is ONLY 1 element of course times, combine substrings
    ctl = str_replace_all(ctl,"\\&ndash;","-")
    ctl = str_replace_all(ctl,"<br>", " ")
    ctl = trimws(paste(gsub("<[^<>]+>","",ctl), collapse = ""))
    ctl = trimws(gsub("\\s{2,}"," ", ctl))
  }
  return(ctl)
}

##################################################
# Test the following courses:
# Spring 2017 EVSC 100 - d100,
# Fall 2018 Stat 452,
# and any of these which are offered this term:
# (STAT100, 201, 203, 270, 330, 350).
# given courses in a list
courses <- list(EVSC100=c(2017,"Spring","EVSC","100","d100"),
                STAT452=c(2018,"Fall","STAT","452","d100"),
                STAT100=c(2022,"spring","stat","100","d100"),
                stat201=c(2022,"sPRING","STAT","201","d100"),
                stat203=c(2022,"sPrInG","Stat","203","d100"),
                stat270=c(2022,"SPRING","sTAT","270","d100"),
                stat330=c(2022,"spring","stat","330","d100"),
                stat350=c(2022,"sprING","StAt","350","d100"))


# produce url corresponding to each courses
course_url <- function(year,term,subject,course_num,section){
  return(sprintf("https://www.sfu.ca/outlines.html?%s/%s/%s/%s/%s",
                 year,
                 tolower(term),
                 tolower(subject),
                 course_num,
                 tolower(section)))
}

# if the above url is N/A, obtain info from SFU Calendar
course_url2 <- function(year,term,subject,course_num) {
  return(sprintf("https://www.sfu.ca/students/calendar/%s/%s/courses/%s/%s.html",
                 year,
                 tolower(term),
                 tolower(subject),
                 course_num))
}

##################################################

# Create a data frame for all test cases:

# Algorithm:

# The function will first examine course info on the "Outline" page.
# If the input section in the given year-term is not available, it will
# redirect to the "SFU Calendar" page and see if other sections are available.
# If there is no other section available,
# it means that the course is not offered in the given year-term
# and the function will fill the data frame with NAs in corresponding entries.

print_df <- function(course_list) {

  # Create an empty data frame to store information.
  # Additionally include Year, Term, and Section to indicate what results
  # were obtained.
  ret = setNames(data.frame(matrix(ncol = 8, nrow = 0)),
                 c("Course_Name",
                   "Year",
                   "Term",
                   "Section",
                   "Course_Title",
                   "Course_Instructor",
                   "Course_Times_and_Location_1",
                   "Course_Times_and_Location_2")
  )

  for (i in seq_along(courses)) {

    # Create an empty vector to store (possibly multiple) course times
    course_time = c()

    url_x = course_url(courses[[i]][1], # year
                       courses[[i]][2], # term
                       courses[[i]][3], # subject
                       courses[[i]][4], # course_num
                       courses[[i]][5]) # section
    course_page = readLines(url_x)

    # if the course page exist, perform all extraction functions
    if (length(extract_cn(course_page)) == 1) {

      names = extract_cn(course_page)
      titles = extract_tt(course_page)
      instructors = extract_ins(course_page)
      course_time = append(course_time,extract_ctl(course_page))

      # Append info to the data frame
      # Missing info (e.g., only one course time available, or no info found
      # for input courses in a given year-term) will be replaced with NA's.
      ret[nrow(ret)+1,] <- c(names,
                             courses[[i]][1],
                             str_to_title(courses[[i]][2]),
                             str_to_title(courses[[i]][5]),
                             titles,
                             instructors,
                             course_time[1],
                             course_time[2])
    }
    else {  # Otherwise, obtain course name and title from SFU calendar,
      # look for other section available.
      # We could have examined if the course exists here, but we assume
      # that input courses should exist
      url_x = course_url2(courses[[i]][1], # year
                          courses[[i]][2], # term
                          courses[[i]][3], # subject
                          courses[[i]][4]) # section
      course_page = readLines(url_x)

      names = extract_cn2(course_page)
      titles = extract_tt2(course_page)

      # if there are other sections, select the first section found
      if (length(grep('\"main-section\"',course_page))!=0) {
        section = course_page[grep('\"main-section\"',course_page)[1]+3]


        # redirect to the outline (i.e., course_page)
        url_x = gsub('.$','',str_extract(section,'http.*\\d\\\"'))
        course_page = readLines(url_x)

        # Store the chosen section number
        section = trimws(gsub("<[^<>]+>","",section))

        instructors = extract_ins(course_page)
        course_time = append(course_time,extract_ctl(course_page))

        ret[nrow(ret)+1,] <- c(names,
                               courses[[i]][1],
                               str_to_title(courses[[i]][2]),
                               str_to_title(section),
                               titles,
                               instructors,
                               course_time[1],
                               course_time[2])

      } else {  # otherwise, fill the df with NA in some columns

        section = NA
        instructors = NA
        course_time = NA

        ret[nrow(ret)+1,] <- c(names,
                               courses[[i]][1],
                               str_to_title(courses[[i]][2]),
                               section,
                               titles,
                               instructors,
                               course_time[1],
                               course_time[2])      }
    }
  }

  return(ret)
}

# Run the function on the test cases
# Turn off garbled warning message
suppressWarnings(print_df(courses))
```

```
##   Course_Name Year   Term Section
## 1    EVSC 100 2017 Spring    D100
## 2    STAT 452 2018   Fall    D100
## 3    STAT 100 2022 Spring    D100
## 4    STAT 201 2022 Spring    D900
## 5    STAT 203 2022 Spring    D100
## 6    STAT 270 2022 Spring    D100
## 7    STAT 330 2022 Spring    <NA>
## 8    STAT 350 2022 Spring    <NA>
##                                            Course_Title Course_Instructor
## 1              Introduction to Environmental Science Marnie Branfireun
## 2                Statistical Learning and Prediction      Brad McNeney
## 3                           Chance and Data Analysis  Richard Lockhart
## 4                  Statistics for the Life Sciences           Wei Lin
## 5 Introduction to Statistics for the Social Sciences     Gamage Perera
## 6        Introduction to Probability and Statistics     Derek Bingham
## 7           Introduction to Mathematical Statistics              <NA>
## 8                Linear Models in Applied Statistics              <NA>
##                   Course_Times_and_Location_1
## 1          Fr 2:30 PM - 4:20 PM SUR 5240, Surrey
## 2        Mo 9:30 AM - 10:20 AM SSCK 9500, Burnaby
## 3         Mo 2:30 PM - 4:20 PM SSCC 9001, Burnaby
## 4         Mo 12:30 PM - 1:20 PM SRYC 2600, Surrey
## 5       Mo 10:30 AM - 12:20 PM SSCC 9002, Burnaby
## 6 Mo, We, Fr 9:30 AM - 10:20 AM WMC 3520, Burnaby
## 7                                            <NA>
## 8                                            <NA>
##                   Course_Times_and_Location_2
## 1                                         <NA>
## 2 We, Fr 9:30 AM - 10:20 AM SSCK 9500, Burnaby
## 3      We 2:30 PM - 3:20 PM SSCC 9001, Burnaby
## 4      Th 12:30 PM - 2:20 PM SRYC 2600, Surrey
## 5     We 10:30 AM - 11:20 AM WMC 3520, Burnaby
## 6                                         <NA>
## 7                                         <NA>
## 8                                         <NA>
```