DA5020 - Week 8 Assignment Web Scraping Programaically

2018-03-19

In this week’s assignment, we continue our pursuit for good burgers in specific neighborhoods in the Boston area using data from the yelp website. We will programatically extract specific fields in the data using the rvest package. This assignment will also provide practice in writing functions and loops. Some questions require you to complete code within a partially written function given a specification. Other questions will not provide starter code.

Questions

1. (20 points) Retrieve the contents of the first webpage for the yelp search as specified in Assignment 7 and write R statements to answer the following questions on the retrieved contents:

- How many nodes are direct descendents of the HTML <body> element (the actual visible content of a web page)?

A) 29 nodes are direct descendents of the HTML <body> element.

```
library(rvest)
library(dplyr)
library(stringr)

page <- read_html("https://www.yelp.com/search?
find_desc=Burgers&start=0&l=p:MA:Boston::
%5BAllston/Brighton,Back_Bay,Beacon_Hill,Downtown,Fenway,South_End,We
st_End%5D")

# list the children of the <html> element (the whole page)
html_children(page)

## {xml_nodeset (2)}
## [1] <head>\n<script>           window.yPageStart = new
Date().getTime() ...
## [2] <body id="yelp_main_body" class="jquery country-us logged-
out">\n\n  ...

# get the root of the actual html body
root <- html_node(page, 'body')

#extract html children from root
html_children(root)

## {xml_nodeset (32)}
##  [1] <script>(function (d, w) {\n    var supportsSVG = (\n        !!d.cr ...
##  [2] <noscript>\n    <link rel="stylesheet" href="https://s3-media2.fl.y ...
##  [3] <script id="yelp-js-error-reporting-init-error-reporting" type="app ...
##  [4] <script>                window.yPerfTimings.push(["body:start", (ne ...
##  [5] <div id="fb-root"></div>
##  [6] <div id="wrap" class="lang-en">\n                <div class="page-h ...
##  [7] <script>            yConfig = {"cookies":
{"ADMIN_SEARCH_USERDATA_R ...
##  [8] <noscript><img src="https://sb.scorecardresearch.com/p?
cj=1&amp;c15 ...
##  [9] <script>\n            (function() {\n                var main = nul ...
## [10] <noscript>\n        <img style="display: none;" src="https://pixel. ...
## [11] <script src="https://s3-
media2.fl.yelpcdn.com/assets/2/www/js/4159b ...
## [12] <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-polyfill/ ...
## [13] <script>\n
window.yPerfTimings.push(["ASYNC_JS:load_jque ...
## [14] <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.mi ...
## [15] <script>if(document.readyState === 'interactive')
jQuery.ready();\n ...
## [16] <script src="https://s3-media3.fl.yelpcdn.com/assets/2/www/js/0f4d1
...
## [17] <script src="https://s3-
media4.fl.yelpcdn.com/assets/2/www/js/cc7d3 ...
## [18] <script>\n                yConfig.vendorExternalURLs["plugin-detect ...
## [19] <script src="https://s3-
media1.fl.yelpcdn.com/assets/2/www/js/e90d3 ...
## [20] <script>yelp.www.init.search.Controller({"adVisibilityURI": "/ad_vi ...
## ...
```

- What are the nodes names of the direct descendents of the <body>?

A) “script”, “noscript” & “div” are the nodes names of the direct descendents of the <body>

```
names <- html_children(root) %>%
  html_name()
unique(names)

## [1] "script"   "noscript" "div"
```

- How many of these direct descendents have an id attribute?

A) Four of these direct descendents have an id attribute

```
id <- html_children(root) %>%
  html_attr("id")
id

##  [1] NA
##  [2] NA
##  [3] "yelp-js-error-reporting-init-error-reporting"
##  [4] NA
##  [5] "fb-root"
##  [6] "wrap"
##  [7] NA
##  [8] NA
##  [9] NA
## [10] NA
## [11] NA
## [12] NA
## [13] NA
## [14] NA
## [15] NA
## [16] NA
## [17] NA
## [18] NA
## [19] NA
## [20] NA
## [21] NA
## [22] NA
## [23] NA
## [24] NA
## [25] NA
## [26] NA
## [27] NA
## [28] NA
## [29] "ttdUniversalPixelTag290e816a69e9439f960a9588bc2ffb54"
## [30] NA
## [31] NA
## [32] NA
```

- What is the css selector to select restaurants that are advertisements? (You may not see the ads if you are logged in on Yelp or have an ad blocker running.)

A) “.yloca-tip” is the css selector to select restaurants that are advertisements.

```
ads <- page %>%
  html_nodes(css = ".yloca-tip")
ads

## {xml_nodeset (1)}
## [1] <span class="yloca-tip" data-hovercard-id="1">\n        Ad\n
</span>
```

2. (50 points) Modify following parameterized function get_yelp_sr_one_page to extract a list of businesses on Yelp, for a specific search keyword, a specific location and a specific page of results.

```
# I did not use the example code provided as it did not work properly for me
# I could not extract the addresses using the example no matter which
method I used
# After trying 5 different ways, I finally gave up and did it like below
get_yelp_sr_one_page <- function(key,loc=NA,page=1){
```

```r
  #function for creating URLs
  makeURL <- function(key,loc=NA,page=1){
    pg <- paste("https://www.yelp.com/search?find_desc=",key,sep="")

        ST <- str_extract(loc,",?([A-Z]{2})") #Extract State abbrev if included
        loc <- gsub("(,?\\s?[A-Z]{2})","",loc) #Remove State Abbrev
        loc <- gsub("\\s","+",loc) #format spaces appropriately
        if(is.na(ST)==F & is.character(ST)==T){ST <- paste(ST,":",sep="")
        loc <- paste("&find_loc=",ST,loc,sep="")
        } #Add : to ST abbrev
        pg <- paste(pg,loc,sep="")

    if(page>1){page <- (page-1)*10
    page <- paste("&start=",page,sep="")
    pg <- paste(pg,page,sep="")
    }
    return(pg)
  }
  URL <- makeURL(key,loc,page=1)#Make the URL

  #Get Results
  h <- read_html(URL)
  li <- html_nodes(h,css=".regular-search-result")
  #Extract parameters
  Name <- html_text(html_nodes(li,css=".biz-name"))
  URL <- html_attr(html_nodes(li,css=".biz-name"),"href")
  Price <- nchar(html_text(html_nodes(li,css=".price-range")),type="chars")
  Ser_Cat <- gsub("\\s{2,}","",html_text(html_nodes(li,css=".category-str-list")))
  Telephone <- gsub("\\s{2,}","",html_text(html_nodes(li,css=".biz-phone")))
  NH <- gsub("\\s{2,}","",html_text(html_nodes(li,css=".neighborhood-str-list")))
  add <- html_nodes(li,css="address")
  Street <- gsub("\\s{2,}","",str_extract(add,"(?<=\\n)[A-Za-z0-9\\s]+(?=<br>)"))
  City <- gsub("\\s{2,}","",str_extract(add,"(?<=<br>)[A-Za-z0-9\\s]+(?=,)"))
  State <- str_extract(add,"[A-Z]{2}")
  Zip <- str_extract(add,"[0-9]{5}")
  Avg_rat <- str_extract(html_attr(html_nodes(li,css=".i-stars"),"title"),"\\d.\\d")
  Num_rew <- str_extract(html_text(html_nodes(li,css=".review-count")),"\\d+")
  Rev_URL <- html_attr(html_nodes(li,css="p.snippet a.nowrap"),"href")
  #Create a list of values for error checking
  cols <- list(Name=Name,URL=URL,Price=Price,Ser_Cat=Ser_Cat,Telephone=Telephone,NH=NH,Street=Street,City=City,State=State,Zip=Zip,Avg_rat=Avg_rat,Num_rew=Num_rew,Rev_URL=Rev_URL)
  #Test for missing results, if missing use the for loop provided to extract each value individually and add NA for missing values.
```

```
do.Index <- vector("character")
for(i in seq_along(cols)){
  if (length(cols[[i]]) < length(cols[[1]])) {
    do.Index <- (names(cols)[[i]])
  }
}
if("NH" %in% do.Index==T){
  NH <- vector("character")
  for(i in seq_along(li)){
    node <- ifelse(
      is.null(html_node(li[[i]], css=".neighborhood-str-list")),
      NA,
      html_node(li[[i]], css=".neighborhood-str-list")  %>%
        html_text()
    )
    NH <- append(NH,node,after=length(NH))
    NH <- gsub("\\s{2,}","",NH)
  }
}
if("Telephone" %in% do.Index==T){
  Telephone <- vector("character")
  for(i in seq_along(li)){
    node <- ifelse(
      is.null(html_nodes(li,css=".biz-phone")),
      NA,
      html_nodes(li,css=".biz-phone")  %>%
        html_text()
    )
    Telephone <- append(Telephone,node,after=length(NH))
    Telephone <- gsub("\\s{2,}","",Telephone)
  }
}
if("Price" %in% do.Index==T){
  Price <- vector("character")
  for(i in seq_along(li)){
    Price[i] <- ifelse(
      is.null(html_nodes(li[i],css=".price-range")),
      NA,
      html_nodes(li[i],css=".price-range")  %>% html_text() %>%
        nchar("chars")
    )
    Price <- gsub("\\s{2,}","",Price)
  }
}
if("Street" %in% do.Index==T){
  Street <- vector("character")
  for(i in seq_along(li)){
    node <- ifelse(
      is.null(html_nodes(li,css="address")),
      NA,
        html_nodes(li,css="address")  %>%
          str_extract("(?<=\\n)[A-Za-z0-9\\s]+(?=<br>)")
      )
      Street <- append(Street,node,after=length(Street))
      Street <- gsub("\\s{2,}","",Street)
    }
  }
  if("City" %in% do.Index==T){
    City <- vector("character")
    for(i in seq_along(li)){
      node <- ifelse(
        is.null(html_nodes(li,css="address")),
        NA,
        html_nodes(li,css="address")  %>%
          str_extract("(?<=<br>)[A-Za-z0-9\\s]+(?=,)")
      )
      City <- append(City,node,after=length(City))
      City <- gsub("\\s{2,}","",City)
    }
  }
  if("State" %in% do.Index==T){
    State <- vector("character")
    for(i in seq_along(li)){
      node <- ifelse(
        is.null(html_nodes(li,css="address")),
        NA,
        html_nodes(li,css="address")  %>%
          str_extract("[A-Z]{2}")
      )
      State <- append(State,node,after=length(State))
    }
  }
  if("Zip" %in% do.Index==T){
    Zip <- vector("character")
    for(i in seq_along(li)){
      node <- ifelse(
        is.null(html_nodes(li,css="address")),
        NA,
        html_nodes(li,css="address")  %>%
          str_extract("[0-9]{5}")
      )
      Zip <- append(Zip,node,after=length(Zip))
    }
  }

  pg <-
cbind(Name,URL,Price,Ser_Cat,Telephone,NH,Street,City,State,Zip,Avg_rat,Nu
m_rew,Rev_URL)
  #Create the output matrix (must be a matrix for the 2nd fn to work)
  return(pg)
```

```
}
result <- get_yelp_sr_one_page("burger",loc="Boston,MA",page=1)

## Warning in cbind(Name, URL, Price, Ser_Cat, Telephone, NH, Street, City, :
## number of rows of result is not a multiple of vector length (arg 3)

head(result)

##      Name
## [1,] "Boston Burger Company"
## [2,] "Tasty Burger"
## [3,] "Beta Burger"
## [4,] "Jm Curley"
## [5,] "Wheelhouse"
## [6,] "The Gallows"
##      URL                                             Price
## [1,] "/biz/boston-burger-company-boston-4?osq=burger" "2"
## [2,] "/biz/tasty-burger-boston?osq=burger"            "1"
## [3,] "/biz/beta-burger-boston?osq=burger"             "1"
## [4,] "/biz/jm-curley-boston?osq=burger"               "2"
## [5,] "/biz/wheelhouse-boston-3?osq=burger"            "1"
## [6,] "/biz/the-gallows-boston?osq=burger"             "2"
##      Ser_Cat                                Telephone
## [1,] "Burgers,American (Traditional),Bars" "(857) 233-4560"
## [2,] "Burgers,Hot Dogs,Fast Food"          "(617) 425-4444"
## [3,] "Burgers,Fast Food"                   "(617) 318-6300"
## [4,] "American (New),Lounges"              "(617) 338-5333"
## [5,] "Breakfast & Brunch,Fast Food"        "(617) 422-0082"
## [6,] "Burgers,Bars,American (Traditional)" "(617) 425-0200"
##      NH                   Street                City     State Zip
## [1,] NA                   "1100 Boylston St"    "Boston" "MA"  "02215"
## [2,] "Fenway"             "1301 Boylston St"    "Boston" "MA"  "02215"
## [3,] "Mission Hill"       "1437 Tremont St"     "Boston" "MA"  "02120"
## [4,] "Downtown"           "21 Temple Pl"        "Boston" "MA"  "02111"
## [5,] "Financial District" "63 Broad St"         "Boston" "MA"  "02109"
## [6,] "South End"          "1395 Washington St" "Boston" "MA"  "02118"
##      Avg_rat Num_rew
## [1,] "4.0"   "647"
## [2,] "4.0"   "951"
## [3,] "4.0"   "80"
## [4,] "4.0"   "685"
## [5,] "4.5"   "270"
## [6,] "4.0"   "762"
##      Rev_URL
## [1,] "/biz/boston-burger-company-boston-4?hrid=ZWOps4iCQJv-
_mrLlcOJSw&osq=burger"
## [2,] "/biz/tasty-burger-boston?
hrid=Y3GFyihUns58NRcUa8obvw&osq=burger"
## [3,] "/biz/beta-burger-boston?hrid=286i9CQxazoKsEcW-
3IUgQ&osq=burger"
## [4,] "/biz/jm-curley-boston?hrid=xspWrxeF8I2xQIZQl8pG7Q&osq=burger"
## [5,] "/biz/wheelhouse-boston-3?
hrid=eq6s5jjZshaTy2jOZaIaQQ&osq=burger"
## [6,] "/biz/the-gallows-boston?
hrid=crb9QGgtWWLGkwvVVLx7SQ&osq=burger"
```

3. (20 points) Write a function that reads multiple pages of the search results of any search keyword and location from Yelp.

Note that for some queries, Yelp may get a different number of results per page. You would need to either change the way you calculate the URL parameter, or use the `distinct(df)` function to remove duplicate rows.

```
mult_pages <- function(key,loc,pages){
  mat <- matrix(ncol=13,nrow=0)
  for(i in seq_along(pages)){
  pg <- get_yelp_sr_one_page(key,loc,page=i)
  mat <- rbind(mat,pg)

  }
  df <- as.data.frame(mat,stringsAsFactors=F)
  return(df)
}
result1 <- mult_pages("Vegetarian","Boston,MA",1:5)
head(result1)
```

```
##                     Name
## 1     My Thai Vegan Cafe
## 2             Clover DTX
## 3    Terramia Ristorante
## 4 Whole Heart Provisions
## 5               By Chloe
## 6             Life Alive
##                                                  URL Price
## 1    /biz/my-thai-vegan-cafe-boston-3?osq=Vegetarian     2
## 2           /biz/clover-dtx-boston-2?osq=Vegetarian     1
## 3     /biz/terramia-ristorante-boston?osq=Vegetarian     3
## 4 /biz/whole-heart-provisions-allston?osq=Vegetarian     2
## 5             /biz/by-chloe-boston-5?osq=Vegetarian     2
## 6          /biz/life-alive-cambridge?osq=Vegetarian     2
##                                   Ser_Cat      Telephone
## 1                  Thai,Vegan,Bubble Tea (617) 451-2395
## 2             Sandwiches,Vegetarian,Cafes
## 3               Italian,Gluten-Free,Vegan (617) 523-3112
## 4                  Vegetarian,Vegan,Cafes (617) 202-5041
## 5                             Vegan,Salad (617) 845-1055
## 6 Vegetarian,Vegan,Juice Bars & Smoothies (617) 354-5433
##                      NH          Street      City State   Zip Avg_rat
## 1             Chinatown     3 Beach St    Boston    MA 02111     4.0
## 2                Downtown     27 School St    Boston    MA 02108     4.0
## 3               North End      98 Salem St    Boston    MA 02113     4.0
## 4        Allston/Brighton 487 Cambridge St   Allston    MA 02134     4.5
## 5 Waterfront, South Boston 107 Seaport Blvd    Boston    MA 02210     3.5
## 6          Central Square     765 Mass Ave Cambridge    MA 02139     4.5
##   Num_rew
## 1     719
## 2     148
## 3     253
## 4     238
## 5     193
## 6    1278
##                                                                  Rev_URL
## 1    /biz/my-thai-vegan-cafe-boston-3?
hrid=mvi_RSmLSBpnCOt9eyufQA&osq=Vegetarian
## 2           /biz/clover-dtx-boston-2?
hrid=H7vhT4_LHtcIwqTzUq5AMg&osq=Vegetarian
## 3     /biz/terramia-ristorante-boston?
hrid=5XQAvRBLw0HQ7ptrgVIY_g&osq=Vegetarian
## 4 /biz/whole-heart-provisions-allston?
hrid=aBG0Yw7u_jDvL2jXLwHrXA&osq=Vegetarian
## 5             /biz/by-chloe-boston-5?hrid=dfBxnl3-
3A6Z2ZbrLhwjTw&osq=Vegetarian
## 6          /biz/life-alive-cambridge?
hrid=UPT0XnzxzTeCCMCIofT0iw&osq=Vegetarian
```

4. (10 points) Optimize your function in question 3, add a small wait time (0.5s for example) between each request, so that you don't get banned by Yelp for abusing their website (hint: use Sys.sleep()).

```r
mult_pages <- function(key,loc,pages){
  mat <- matrix(ncol=13,nrow=0)
  for(i in seq_along(pages)){
  pg <- get_yelp_sr_one_page(key,loc,page=i)
  mat <- rbind(mat,pg)
  Sys.sleep(0.5)
  }
  df <- as.data.frame(mat,stringsAsFactors=F)
  return(df)
}
result1 <- mult_pages("Vegetarian","Boston,MA",1:5)
head(result1)
```

```
##                     Name
## 1     My Thai Vegan Cafe
## 2            Clover DTX
## 3    Terramia Ristorante
## 4 Whole Heart Provisions
## 5              By Chloe
## 6            Life Alive
##                                                       URL Price
## 1    /biz/my-thai-vegan-cafe-boston-3?frvs=True&osq=Vegetarian     2
## 2          /biz/clover-dtx-boston-2?frvs=True&osq=Vegetarian     1
## 3     /biz/terramia-ristorante-boston?frvs=True&osq=Vegetarian     3
## 4 /biz/whole-heart-provisions-allston?frvs=True&osq=Vegetarian     2
## 5            /biz/by-chloe-boston-5?frvs=True&osq=Vegetarian     2
## 6         /biz/life-alive-cambridge?frvs=True&osq=Vegetarian     2
##                                  Ser_Cat      Telephone
## 1                 Thai,Vegan,Bubble Tea (617) 451-2395
## 2            Sandwiches,Vegetarian,Cafes
## 3              Italian,Gluten-Free,Vegan (617) 523-3112
## 4                 Vegetarian,Vegan,Cafes (617) 202-5041
## 5                            Vegan,Salad (617) 845-1055
## 6 Vegetarian,Vegan,Juice Bars & Smoothies (617) 354-5433
##                         NH           Street      City State   Zip Avg_rat
## 1                Chinatown       3 Beach St    Boston    MA 02111     4.0
## 2                 Downtown     27 School St    Boston    MA 02108     4.0
## 3                North End      98 Salem St    Boston    MA 02113     4.0
## 4         Allston/Brighton 487 Cambridge St   Allston    MA 02134     4.5
## 5 Waterfront, South Boston 107 Seaport Blvd    Boston    MA 02210     3.5
## 6           Central Square     765 Mass Ave Cambridge    MA 02139     4.5
##   Num_rew
## 1     719
## 2     148
## 3     253
## 4     238
## 5     193
## 6    1278
##                                                                       Rev_URL
## 1    /biz/my-thai-vegan-cafe-boston-3?
frvs=True&hrid=mvi_RSmLSBpnCOt9eyufQA&osq=Vegetarian
## 2          /biz/clover-dtx-boston-2?
frvs=True&hrid=H7vhT4_LHtcIwqTzUq5AMg&osq=Vegetarian
## 3     /biz/terramia-ristorante-boston?
frvs=True&hrid=5XQAvRBLw0HQ7ptrgVIY_g&osq=Vegetarian
## 4 /biz/whole-heart-provisions-allston?
frvs=True&hrid=aBG0Yw7u_jDvL2jXLwHrXA&osq=Vegetarian
## 5            /biz/by-chloe-boston-5?frvs=True&hrid=dfBxnI3-
3A6Z2ZbrLhwjTw&osq=Vegetarian
## 6         /biz/life-alive-cambridge?
frvs=True&hrid=UPT0XnzxzTeCCMCIofT0iw&osq=Vegetarian
```

Submission

You need to submit an .Rmd extension file as well as the generated pdf file. Be sure to state all the assumptions and give explanations as comments in the .Rmd file wherever needed to help us assess your submission. Remember to use the naming convention you have been following in this course so far.