

CS636 – PROJECT 1

WEB SCRAPING IN R

1. Given an input year, our objective is to extract all articles published in/after that year from each selected journal.
2. The program is expected to store the extracted information into a plain file elegantly.
3. Our program has to be encapsulated into a function which will take the year as a parameter.
4. Stored data could be easily read into R again.
5. We are required to extract the following 9 fields for each article;
 - a. Title
 - b. Authors
 - c. Author Affiliations
 - d. Corresponding Author
 - e. Corresponding Author's Email
 - f. Publish Date
 - g. Abstract
 - h. Key Words
 - i. Full Paper (Text Format)

CHOSEN JOURNAL: HEREDITAS
<https://hereditasjournal.biomedcentral.com/>

I have decided to do this project by myself and chose the Hereditas Journal to work on.

I used rvest and xml2 packages to harvest the data; and css selectors for web scraping.

ENCOUNTERED CHALLENGES

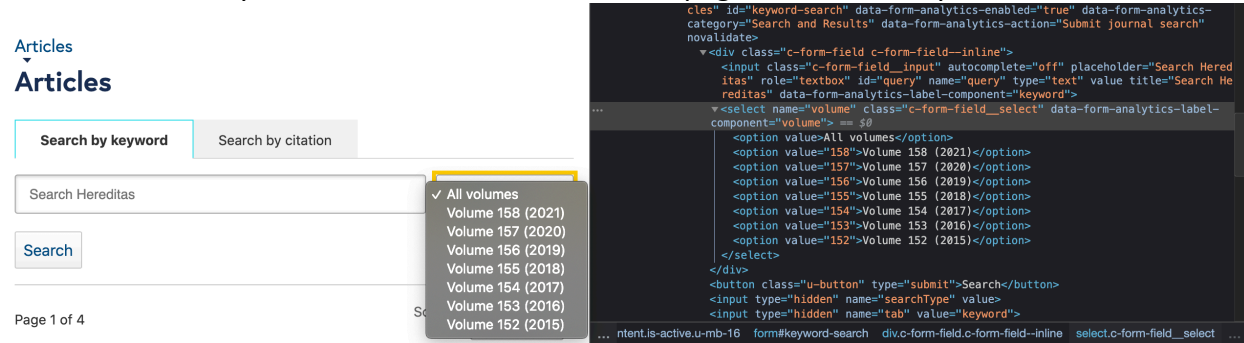
Using CSS Selectors

I had to learn how to use CSS Selectors for Web Scraping. It was a little bit challenging for me. It took some time getting used to it and selecting the correct html parts required trial and error.

Efficiency Problem & Alternative Way

My first strategy was extracting each field separately for all the articles. So, I had a vector of Titles, Authors, etc. But I felt that was less reliable. So, I decided to loop through all the articles one by one. It made more sense but doing so made my program a little slow. Because I basically go through each page and download all the articles. Then, filter it by the given year.

When I finished my code; I realized there is a search page for articles as you can see below.



Instead of going through each page; I could get these urls by volumes (and years). And for a given year, I would only choose the related urls by volumes and only download and extract the related articles published in or after that year. And this would be the most efficient but I didn't want to change my design.

Corresponding Author's Email: Not Available

I searched the page very carefully but I couldn't find corresponding author's email for any of the articles. There is a message box if you want to reach the corresponding author. That is why I assigned "NA" for this field.

Extracting the Full Article

I can say that extracting the field for Full Paper was the biggest challenge I encountered.

I wasn't sure if I should extract the whole text of the "article" element from html. Because the whole article (Full Text) also contains the fields such as Abstract, Corresponding Author information, Affiliations and Keywords. So, to me, It doesn't make sense extracting the whole article element since we have those values already in separate fields. I decided to exclude these parts by using the `:not()` css selector like this;

```
'article section:not([data-title=Abstract]):not([data-title="Author information"]):not([data-title="About this article"])'
```

I liked the logic I mentioned above but I decided not to use this and extract the whole article element instead. I felt like It wasn't up to me to choose what should be included and what shouldn't. Since this Project's requirements didn't give any detailed information about this, I thought it's best to keep the whole article information in the Full Text field.

The other reason why I went with selecting all the text for the article was that the html `<section>` tags weren't the same for all the articles and I wasn't sure how to exclude unwanted sections reliably (as can be seen below).

```
... ><div class="c-pdf-download u-clear-both">...</div>
... ><article itemscope itemtype="http://schema.org">...</article>
  ><div class="c-article-header">...</div>
  ><section aria-labelledby="Abs1" data-title="Introduction">...</section>
  ><section data-title="Case presentation">...</section>
  ><section data-title="Discussion">...</section>
  ><section data-title="Availability of data and materials">...</section>
  ><section data-title="Abbreviations">...</section>
  ><section aria-labelledby="Bib1" data-title="References">...</section>
  ><section data-title="Acknowledgments">...</section>
  ><section data-title="Funding">...</section>
  ><section aria-labelledby="author-information">...</section>
  ><section data-title="Ethics declarations">...</section>
  ><section data-title="Additional information">...</section>
  ><section data-title="Supplementary Information">...</section>
  ><section data-title="Rights and permissions">...</section>
  ><section aria-labelledby="article-info" data-title="Article information">...</section>
</main>
  ><div class="c-article-extras u-text-sm u-hide-phone">...</div>
  ::after
  ><div class="c-journal-footer">...</div>

... ><div class="c-pdf-download u-clear-both">...</div>
... ><article itemscope itemtype="http://schema.org">...</article>
  ><div class="c-article-header">...</div>
  ><section aria-labelledby="Abs1" data-title="Introduction">...</section>
  ><section data-title="Background">...</section>
  ><section data-title="Results">...</section>
  ><section data-title="Discussion">...</section>
  ><section data-title="Conclusions">...</section>
  ><section data-title="Methods">...</section>
  ><section data-title="Availability of data and materials">...</section>
  ><section data-title="Abbreviations">...</section>
  ><section aria-labelledby="Bib1" data-title="References">...</section>
  ><section data-title="Acknowledgments">...</section>
  ><section data-title="Funding">...</section>
  ><section aria-labelledby="author-information">...</section>
  ><section data-title="Ethics declarations">...</section>
  ><section data-title="Additional information">...</section>
  ><section data-title="Supplementary Information">...</section>
  ><section data-title="Rights and permissions">...</section>
  ><section aria-labelledby="article-info" data-title="Article information">...</section>
</main>
  ><div class="c-article-extras u-text-sm u-hide-phone">...</div>
  ::after
  ><div class="c-journal-footer">...</div>
```

The other problem I encountered here was when writing to a csv file. While the file itself was valid, it turns out Excel has a cell maximum length limit. The file wasn't readable directly in Excel so I decided to truncate the string so the data fits in an excel cell.

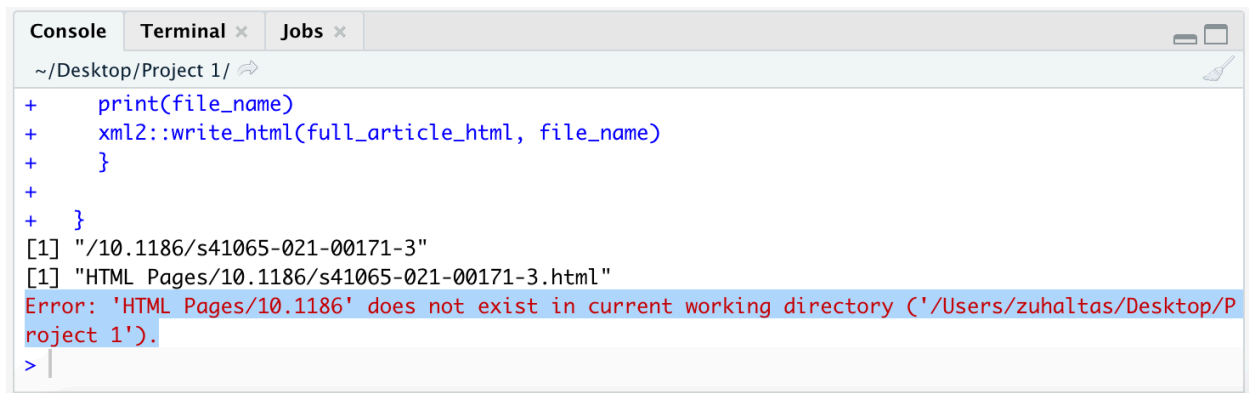
```
substr(full_paper,1,32767)
```

It was still spilling over the cells some times because of some special characters. I used this regular expression I found on stackoverflow to remove junk characters and it solved the problem.

```
Full_Paper = gsub('[^\x20-\x7E]', '', substr(full_paper,1,32767))
```

Downloading HTML files & DOI

The problem here was not having a folder named '10.1186'. I didn't understand why I needed a folder named '10.1186'. It took some time for me to figure out. Turns out it's because we can't use '/' in any kind of file name and there was a '/' in the DOI format; I was trying to assign DOI directly to the file name and it was looking for the '10.1186' folder. I split the DOI by '/' and created a folder for '10.1186' and named the file with the other side of the '/'.

A screenshot of an R console window. The window has tabs for 'Console', 'Terminal', and 'Jobs'. The current directory is '~ / Desktop / Project 1 /'. The console shows several lines of R code being executed, followed by two print statements. The first print statement outputs the path '/10.1186/s41065-021-00171-3'. The second print statement outputs the full file path 'HTML Pages/10.1186/s41065-021-00171-3.html'. Below these, a red error message is displayed: 'Error: 'HTML Pages/10.1186' does not exist in current working directory ('/Users/zuhaltas/Desktop/Project 1').'. The prompt '>' is visible at the bottom.

```
~/Desktop/Project 1/
+   print(file_name)
+   xml2::write_html(full_article_html, file_name)
+   }
+
+ }
[1] "/10.1186/s41065-021-00171-3"
[1] "HTML Pages/10.1186/s41065-021-00171-3.html"
Error: 'HTML Pages/10.1186' does not exist in current working directory ('/Users/zuhaltas/Desktop/Project 1').
> |
```

REFERENCES

<https://stackoverflow.com/questions/38828620/how-to-remove-strange-characters-using-gsub-in-r>

https://www.w3schools.com/css/css_attribute_selectors.asp

<https://www.datacamp.com/community/tutorials/r-web-scraping-rvest>

<https://towardsdatascience.com/tidy-web-scraping-in-r-tutorial-and-resources-ac9f72b4fe47>