Data607 - Assign7

Rachel Greenlee

10/5/2020

Introduction

In this assignment I used Notepad to write the data for 3 books in 3 different formats: HTML, XML, and JSON. Next, I load each of these files into R and place in separate data frames.

XML Data

Using the XML and RCurl packages (as xmlParse wouldn't accept my file as XML without RCurl), I access the .xml from my GitHub, parse it, and put it into a data frame.

```
library(XML)
library(RCurl)
library(kableExtra)
```

```
xml_URL <- "https://raw.githubusercontent.com/rachel-greenlee/data607_assign7/master/book_xml.xml"
xml_link <- getURL(xml_URL)
xml_data <- xmlParse(file = xml_link)
xml_df <- xmlToDataFrame(nodes = getNodeSet(xml_data, "//Book"))
kable(xml_df, format = "simple")
```

Title	Author	Genre	Story_Location	Goodreads_Rating	Year_Pu
The Lowland	Jhumpa Lahiri	Fiction	Calcutta, India	3.85	2013
An American Marriage	Tayari Jones	Fiction	Georgia, USA	3.96	2018
Leviathan Wakes	Daniel Abraham, Ty Franck	Science Fiction	The Milky Way	4.25	2011

HTML Data

Conveniently also from the XML package, there is a readHTMLTable function that allows us to read straight into a dataframe.

```
html_URL <- "https://raw.githubusercontent.com/rachel-greenlee/data607_assign7/master/books_html.html"
html_link <- getURL(html_URL)

html_df <- readHTMLTable(html_link)
kable(html_df, format = "simple")
```

Title	Author	Genre	Story_Location	Goodreads_Rating	Year_Pu
The Lowland	Jhumpa Lahiri	Fiction	Calcutta, India	3.85	2013
An American Marriage	Tayari Jones	Fiction	Georgia, USA	3.96	2018
Leviathan Wakes	Daniel Abraham, Ty Franck	Science Fiction	The Milky Way	4.25	2011

JSON Data

The jsonlite package made quick work of reading my JSON file into a dataframe, and I didn't need to use RCurl this time.

```
library("jsonlite")
```

```
json_df <-fromJSON("https://raw.githubusercontent.com/rachel-greenlee/data607_assign7/master/book_json.
kable(json_df, format = "simple")
```

Title	Author	Genre	Story_Location	Goodreads_Rating	Year_Pu
The Lowland	Juhumpa Lahiri	Fiction	Clacutta, India	3.85	2031
An American Marriage	Tayari Jones	Fiction	Georgia, USA	3.96	2018
Leviathan Wakes	Daniel Abraham, Ty Franck	Science Fiction	The Milky Way	4.25	2011

Data Frame Comparisons

By sight these data frames all seem identical, but when we use the all.equal function we see a list of some of the differences between the three data frames.

```
all.equal(html_df, xml_df, json_df)
```

```
##  [1] "Names: 1 string mismatch"
##  [2] "Attributes: < names for current but not for target >"
##  [3] "Attributes: < Length mismatch: comparison on first 0 components >"
##  [4] "Length mismatch: comparison on first 1 components"
##  [5] "Component 1: Modes: list, character"
##  [6] "Component 1: names for target but not for current"
##  [7] "Component 1: Attributes: < Modes: list, NULL >"
##  [8] "Component 1: Attributes: < names for target but not for current >"
##  [9] "Component 1: Attributes: < current is not list-like >"
## [10] "Component 1: Length mismatch: comparison on first 3 components"
## [11] "Component 1: Component 1: Lengths (3, 1) differ (string compare on first 1)"
## [12] "Component 1: Component 2: Lengths (3, 1) differ (string compare on first 1)"
## [13] "Component 1: Component 2: 1 string mismatch"
## [14] "Component 1: Component 3: Lengths (3, 1) differ (string compare on first 1)"
## [15] "Component 1: Component 3: 1 string mismatch"
```

Conclusion

Loading these three formats into data frames was not as difficult as I had expected it would be, though I see from some online resources that as the nodes increase, the code needs to increase as well. In my case, even

before using kable, each data frame looks identical to the next which is pretty amazing! However, when we dig deeper using the all.equal function there are differences between them that could make further analysis easier/harder depending on the task.