Data 607 - Assignment 5 - Working with Document File Formats

William Jasmine

2022-09-21

Introduction

This assignment involves creating three different document files of different type (`.xml`, `.html`, and `.xml`), and loading them into R dataframes. As such, the following three files were created by hand: `physics_books.html`, `physics_books.xml`, and `physics_books.json`. They each contain the same information (details regarding three physics textbooks), the only difference being that each of them uses their own specific file format. The following code goes through the process of loading and cleaning each of these files into tidy R dataframes. Each of the files can be found on Github.

HTML File

Ingesting File

The easiest file to load and clean is likely the `.html` file, since HTML gives a user the ability to store data in tables that when published to websites very much resemble R dataframes. The `XML` package has a function called `readHTMLTable` that makes the transition from HTML table to R dataframe pretty seamless. It is implemented below:

```
html_str <- read_file('https://raw.githubusercontent.com/williamzjasmine/CUNY_SPS_DS/master/DATA_607/Hon
html_df <- readHTMLTable(html_str)
html_df <- html_df[[1]]
glimpse(html_df)
```

```
## Rows: 3
## Columns: 9
## $ Title                     <chr> "Introduction to Quantum Mechanics", "A Firs~
## $ Author                    <chr> "David J. Griffiths", "Barton Zwieback", "Ro~
## $ Publication_Info.company  <chr> "Pearson Prentice Hall", "Cambridge Universi~
## $ Publication_Info.year     <chr> "2005", "2009", "2009"
## $ Publication_Info.location <chr> "Upper Saddle River, New Jersey, 07468", "Ne~
## $ Amazon_Info.price         <chr> "$61.28", "$70.99", "$49.99"
## $ Amazon_Info.rating        <chr> "4.6", "4.6", "N/A"
## $ Amazon_Info.num_reviews   <chr> "1,225", "86", "0"
## $ Amazon_Info.link          <chr> "https://www.amazon.com/Introduction-Quantum~
```

As can be seen above, `html_df` is now and R dataframe that contains all of the HTML table data.

Cleaning the Data

The first step that we can take in cleaning this data is renaming some of the column names, as their current format is slightly "clunky" to work with.

```r
new_col_names <- colnames(html_df) %>%
  tolower() %>%
    str_replace_all("publication_info.", 'publisher_') %>%
      str_replace_all("amazon_info.", 'amazon_')

colnames(html_df) <- new_col_names
glimpse(html_df)
```

```
## Rows: 3
## Columns: 9
## $ title              <chr> "Introduction to Quantum Mechanics", "A First Cours~
## $ author             <chr> "David J. Griffiths", "Barton Zwieback", "Roman V. ~
## $ publisher_company  <chr> "Pearson Prentice Hall", "Cambridge University Pres~
## $ publisher_year     <chr> "2005", "2009", "2009"
## $ publisher_location <chr> "Upper Saddle River, New Jersey, 07468", "New York,~
## $ amazon_price       <chr> "$61.28", "$70.99", "$49.99"
## $ amazon_rating      <chr> "4.6", "4.6", "N/A"
## $ amazon_num_reviews <chr> "1,225", "86", "0"
## $ amazon_link        <chr> "https://www.amazon.com/Introduction-Quantum-Mechan~
```

The next step is to convert all the quantitative columns fields into actual numerical fields, as they are all currently character fields. The methodology is a bit different for each one given the data cleaning they require, but the following cell converts the `publisher_year`, `amazon_price`, `amazon_rating`, and `amazon_num_reviews` field into numerical fields:

```r
html_df$amazon_price <- html_df$amazon_price %>%
  str_replace_all("\\$", '')

html_df$amazon_num_reviews <- html_df$amazon_num_reviews %>%
  str_replace_all(",", '')

html_df$amazon_rating <- html_df$amazon_rating %>%
  na_if("N/A")

html_df <- html_df %>%
  mutate(
    publisher_year = as.integer(publisher_year),
    amazon_price = as.numeric(amazon_price),
    amazon_num_reviews = as.integer(amazon_num_reviews),
    amazon_rating = as.numeric(amazon_rating)
  )

glimpse(html_df)
```

```
## Rows: 3
## Columns: 9
## $ title              <chr> "Introduction to Quantum Mechanics", "A First Cours~
## $ author             <chr> "David J. Griffiths", "Barton Zwieback", "Roman V. ~
## $ publisher_company  <chr> "Pearson Prentice Hall", "Cambridge University Pres~
## $ publisher_year     <int> 2005, 2009, 2009
## $ publisher_location <chr> "Upper Saddle River, New Jersey, 07468", "New York,~
## $ amazon_price       <dbl> 61.28, 70.99, 49.99
## $ amazon_rating      <dbl> 4.6, 4.6, NA
## $ amazon_num_reviews <int> 1225, 86, 0
## $ amazon_link        <chr> "https://www.amazon.com/Introduction-Quantum-Mechan~
```

Now that all the columns are of the correct type, the last data cleaning step is to split the `publisher_location` column into three new columns, `publisher_city`, `publisher_state`, `publisher_zip`. This is done in the cell below:

```
html_df <- html_df %>%
  mutate(
    publisher_city = str_extract(publisher_location, '^(.+?),'),
    publisher_state = str_extract(publisher_location, ',(.+?),'),
    publisher_zip = str_extract(publisher_location, ',( \\d{5})')
  ) %>%
    select(!publisher_location)

html_df$publisher_city <- html_df$publisher_city %>%
  str_replace_all(",", '')

html_df$publisher_state <- html_df$publisher_state %>%
  str_replace_all(",", '')

html_df$publisher_zip <- html_df$publisher_zip %>%
  str_replace_all(', ', '')

glimpse(html_df)
```

```
## Rows: 3
## Columns: 11
## $ title              <chr> "Introduction to Quantum Mechanics", "A First Cours~
## $ author             <chr> "David J. Griffiths", "Barton Zwieback", "Roman V. ~
## $ publisher_company  <chr> "Pearson Prentice Hall", "Cambridge University Pres~
## $ publisher_year     <int> 2005, 2009, 2009
## $ amazon_price       <dbl> 61.28, 70.99, 49.99
## $ amazon_rating      <dbl> 4.6, 4.6, NA
## $ amazon_num_reviews <int> 1225, 86, 0
## $ amazon_link        <chr> "https://www.amazon.com/Introduction-Quantum-Mechan~
## $ publisher_city     <chr> "Upper Saddle River", "New York", "Boca Raton"
## $ publisher_state    <chr> " New Jersey", " New York", " Florida"
## $ publisher_zip      <chr> "07468", "10006", "33487"
```

The data above is in its final clean state, and is ready to be analyzed.

XML File

The next file to read in is the XML file. This can be done using the `XML` package's `xmlToDataFrame` function

```
xml_str <- read_file('https://raw.githubusercontent.com/williamzjasmine/CUNY_SPS_DS/master/DATA_607/Home
xml_df <- xmlToDataFrame(xml_str, collectNames = TRUE)
glimpse(xml_df)
```

```
## Rows: 3
## Columns: 4
## $ Title            <chr> "Introduction to Quantum Mechanics", "A First Course ~
## $ Author           <chr> "David J. Griffiths", "Barton Zwieback", "Roman V. Kr~
## $ Publication_Info <chr> "Pearson Prentice Hall2005Upper Saddle River, New Jer~
## $ Amazon_Info      <chr> "$61.284.61,225https://www.amazon.com/Introduction-Qu~
```

As can be seen in the output above, this file format is slightly harder to ingest, seeing as the nested "Publication_Info" and "Amazon_Info" columns we not parsed and separated into their individual columns.

JSON File

The last file to be ingested is the JSON file, which is done below using the `fromJSON` function.

```
json_str <- read_file("https://raw.githubusercontent.com/williamzjasmine/CUNY_SPS_DS/master/DATA_607/Hom

json_nested <- fromJSON(json_str)
json_unlisted <-
  lapply(json_nested, function(x) {
    x[sapply(x, is.null)] <- NA
    unlist(x)
  })

json_df <- as.data.frame(do.call("cbind", json_unlisted))

json_df
```

```
##
## Book.Title                                                                           Introduction to Qu
## Book.Author                                                                                         Dav
## Book.Publication.Info.company                                                                   Pearso
## Book.Publication.Info.year
## Book.Publication.Info.location                                                   Upper Saddle River, Ne
## Book.Amazon.Info.price
## Book.Amazon.Info.rating
## Book.Amazon.Info.num_reviews
## Book.Amazon.Info.link           https://www.amazon.com/Introduction-Quantum-Mechanics-David-Griffith
## Book.Title.1                                                                           A First Course I
## Book.Author.1
## Book.Publication.Info.company.1                                                             Cambridge U
## Book.Publication.Info.year.1
## Book.Publication.Info.location.1                                                             New York,
## Book.Amazon.Info.price.1
## Book.Amazon.Info.rating.1
## Book.Amazon.Info.num_reviews.1
## Book.Amazon.Info.link.1                     https://www.amazon.com/First-Course-String-Theory-2n
## Book.Title.2                                                       Cold Molecules: Theory, Experimen
## Book.Author.2
## Book.Publication.Info.company.2
## Book.Publication.Info.year.2
## Book.Publication.Info.location.2                                                            Boca Raton
## Book.Amazon.Info.price.2
## Book.Amazon.Info.rating.2
## Book.Amazon.Info.num_reviews.2
## Book.Amazon.Info.link.2           https://www.amazon.com/Cold-Molecules-Theory-Experiment-Applicatio
```

This file is probably the most complex to be parsed, given the structure of the JSON format. However, via use of the `sapply` and `unlist` functions above, the data is transformed so that it can be restructured into a usable R dataframe.