Week 7 Assignment

Alice Ding

2023-03-07

Overview

I've picked 3 fiction books, each written by people of color and telling stories from their own personal lens. These brings different cultures through the authors' own personal experiences, but each story is inherently American in nature as well.

I've put the following information in `html`, `xml`, and `json` formats:

- Title
- Author(s)
- Publish Year
- Genre
- Synopsis

```
html <- "https://github.com/addsding/data607/raw/main/assignment7/books.html"
xml <- "https://github.com/addsding/data607/raw/main/assignment7/books.xml"
json <- "https://github.com/addsding/data607/raw/main/assignment7/books.json"
```

Let's work on getting these into dataframes!

Extracting the Data

HTML

I'll be using the `rvest` package for this portion of the assignment.

```
html_df <- as.data.frame(read_html(html) |> html_table(fill=TRUE))
head(html_df)
```

```
##               Title                  Author Publish.Year   Genre
## 1          Pachinko             Min Jin Lee         2017 Fiction
## 2 The Joy Luck Club                Amy Tan         2006 Fiction
## 3  Punching the Air Ibi Zoboi, Yusef Salaam         2020 Fiction
##
## 1 In the early 1900s, teenaged Sunja, the adored daughter of a crippled fisherman, falls for a wealth
## 2
## 3
```

Looks pretty good! One of the column names though has a period rather than a space/underscore – let's change that!

```
html_df <- html_df |> rename("Publish_Year" = "Publish.Year")
head(html_df)
```

```
##               Title                  Author Publish_Year   Genre
## 1          Pachinko             Min Jin Lee         2017 Fiction
## 2 The Joy Luck Club                 Amy Tan         2006 Fiction
## 3  Punching the Air Ibi Zoboi, Yusef Salaam         2020 Fiction
##
## 1 In the early 1900s, teenaged Sunja, the adored daughter of a crippled fisherman, falls for a wealth
## 2
## 3
```

Seems relatively clean! For the `Author` column, we can try splitting that into 2 columns just so two authors aren't in one column for the third book.

```
html_df <- html_df |>
  separate_wider_delim(Author, delim=", ", names = c("Author_1", "Author_2"), too_few = "align_start")

head(html_df)
```

```
## # A tibble: 3 x 6
##   Title             Author_1    Author_2     Publish_Year Genre   Synopsis
##   <chr>             <chr>       <chr>               <int> <chr>   <chr>
## 1 Pachinko          Min Jin Lee <NA>                 2017 Fiction "In the early~
## 2 The Joy Luck Club Amy Tan     <NA>                 2006 Fiction "Four mothers~
## 3 Punching the Air  Ibi Zoboi   Yusef Salaam         2020 Fiction "The story th~
```

Looks good! Let's move onto `json`.

JSON

I'll be using the `jsonlite` package for this portion of the assignment.

```
json_df <- jsonlite::fromJSON(json)
json_df <- as.data.frame(json_df)
head(json_df)
```

```
##         books.title            books.author books.publishYear books.genre
## 1          Pachinko             Min Jin Lee              2017     Fiction
## 2 The Joy Luck Club                 Amy Tan              2006     Fiction
## 3  Punching the Air Ibi Zoboi, Yusef Salaam              2020     Fiction
##
## 1 In the early 1900s, teenaged Sunja, the adored daughter of a crippled fisherman, falls for a wealth
## 2
## 3
```

For the `author` column, the third book causes it to be a list since there are two authors. Let's try to split that column into two to match the HTML data frame.

```
json_df <- json_df |> unnest_wider(books.author, names_sep="author")
head(json_df)
```

```
## # A tibble: 3 x 6
##   books.title       books.authorauthor1 books.authorau~1 books~2 books~3 books~4
##   <chr>             <chr>               <chr>              <int> <chr>   <chr>
## 1 Pachinko          Min Jin Lee         <NA>                2017 Fiction "In th~
## 2 The Joy Luck Club Amy Tan             <NA>                2006 Fiction "Four ~
## 3 Punching the Air  Ibi Zoboi           Yusef Salaam        2020 Fiction "The s~
## # ... with abbreviated variable names 1: books.authorauthor2,
## #   2: books.publishYear, 3: books.genre, 4: books.synopsis
```

Looks perfect! Now just clean up the column names and it should be good to go.

```
json_df <- json_df |> rename("title" = "books.title",
                             "author_1" = "books.authorauthor1",
                             "author_2" = "books.authorauthor2",
                             "publish_year" = "books.publishYear",
                             "genre" = "books.genre",
                             "synopsis" = "books.synopsis"
                             )
head(json_df)
```

```
## # A tibble: 3 x 6
##   title             author_1    author_2     publish_year genre   synopsis
##   <chr>             <chr>       <chr>               <int> <chr>   <chr>
## 1 Pachinko          Min Jin Lee <NA>                 2017 Fiction "In the early~
## 2 The Joy Luck Club Amy Tan     <NA>                 2006 Fiction "Four mothers~
## 3 Punching the Air  Ibi Zoboi   Yusef Salaam         2020 Fiction "The story th~
```

Awesome, XML next.

XML

I'll be using the `xml2` and `tibble` packages for this portion of the assignment.

```
# get the XML
xml_books <- read_xml(xml)

# pull each column
title <- xml_text(xml_find_all(xml_books, xpath = "//title"))
author <- xml_text(xml_find_all(xml_books, xpath = "//author"))
year <- xml_text(xml_find_all(xml_books, xpath = "//publishYear"))
genre <- xml_text(xml_find_all(xml_books, xpath = "//genre"))
synopsis <- xml_text(xml_find_all(xml_books, xpath = "//synopsis"))

# turn it into a df; for author, it seems that we have to append the 2nd author for the third book sepa
xml_df <- tibble(title = title
                 , author_1 = author[1:3]
                 , publish_year = year
                 , genre = genre
```

```
                , synopsis = synopsis
                )
author_2 <- c(NA, NA, author[4])
xml_df$author_2 <- author_2

head(xml_df)
```

```
## # A tibble: 3 x 6
##   title             author_1    publish_year genre   synopsis            autho~1
##   <chr>             <chr>       <chr>        <chr>   <chr>               <chr>
## 1 Pachinko          Min Jin Lee 2017         Fiction "In the early 1900~ <NA>
## 2 The Joy Luck Club Amy Tan     2006         Fiction "Four mothers, fou~ <NA>
## 3 Punching the Air  Ibi Zoboi   2020         Fiction "The story that I ~ Yusef ~
## # ... with abbreviated variable name 1: author_2
```

Looks perfect!

Conclusion

With these methods, I would say the data frames are basically the same due to my own manipulation. From the start, the data frames were certainly not exactly the same. For example, I do see now that the XML method does not make `publish_year` an int, however the data could be cleaned up a little more for sure to get things correct. The order of the columns is also not the same for XML due to how I structured it. I think ultimately, most fields would be imported into a similar data frame, it's just lists that make things complicated too as well as object types.

Working with each data structure, I found HTMl to be the easiest to work with due to the simplicity of the package and the fact that it kind of already starts as a table to begin with anyway. JSON and XML are similar in nature with how they're structured and how they're created, but the packages vary in difficulty to use overall.

I think the code I've written can be reworked for multiple books with more fields as well for JSON and HTML, however the way I implemented the XML variation makes it a little more work if there are more columns to work with. It's also very much not repeatable if the books I had were in a different order – I would have to rewrite it to fit my use case depending on the data presented.

Overall, I found this exercise to be interesting and fun in its own way as I got to work with different data structures and found a way to make the same three data frames!