DATA 607, Week 7: Working with HTML, XML, and JSON Files

Kevin Kirby

2024-10-09

Overview

This is the week seven for the Fall 2024 edition of DATA 607. This assignments covers the creation, importing, and manipulation of HTML, XML, and JSON files. The assignment stated:

"Pick three of your favorite books on one of your favorite subjects. At least one of the books should have more than one author. For each book, include the title, authors, and two or three other attributes that you find interesting. . . separately create three files which store the book's information in HTML (using an html table), XML, and JSON formats. Create each of these files "by hand."

In the Cursor code editor I use for my day-to-day coding, I created the three files with the following information:

- Title
- Authors
- Release Date
- Total Pages
- Objective History
  - My subjective binary on whether I believe the book is objective history

The files can be found at these links. Please note: by clicking on this like, the file will automatically download to your computer.

- tour_books.html
- tour_books.json
- tour_books.xml

These are these libraries required for this code to work:

```
library(rvest)
library(xml2)
library(jsonlite)
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:jsonlite':
##
##     flatten
```

HTML file work

```
gcp_html <- "https://storage.googleapis.com/data_science_masters_files/2024_fall/data_607_data_manageme

tour_books_h <- read_html(gcp_html)
```

```r
tour_books_html <- tour_books_h %>%
  html_nodes("tr") %>%
  html_nodes("td") %>%
  html_text() %>%
  matrix(ncol = 5, byrow = TRUE)
tour_books_html_df <- as.data.frame(tour_books_html, stringsAsFactors = FALSE)
colnames(tour_books_html_df) <- c("title", "authors", "release_date", "total_pages", "objective_history

cat("HTML df:\n")
```

```
## HTML df:
```

```r
print(head(tour_books_html_df))
```

```
##                                                                          title
## 1                                                The Tour de France: A History
## 2           Yellow Jersey: The Inside Story of the Greatest Cycling Race on Earth
## 3 The Lance Armstrong Years: A Cycling Legend's Rise to Power and Fall from Grace
##                         authors release_date total_pages objective_history
## 1                 Richard Moore   2023-04-11         400              TRUE
## 2 Daniel Friebe, Jeremy Whittle   2019-06-03         352              TRUE
## 3                  Daniel Coyle   2012-02-21         288              TRUE
```

XML file work

```r
gcp_xml <- "https://storage.googleapis.com/data_science_masters_files/2024_fall/data_607_data_managemen

tour_books_x <- read_xml(gcp_xml)
tour_books_xml_df <- tour_books_x %>%
  xml_find_all("//book") %>%
  map_df(function(book) {
    list(
      title = xml_text(xml_find_first(book, "title")),
      authors = xml_text(xml_find_first(book, "authors")),
      release_date = xml_text(xml_find_first(book, "release_date")),
      total_pages = xml_text(xml_find_first(book, "total_pages")),
      objective_history = xml_text(xml_find_first(book, "objective_history"))
    )
  })

cat("\nXML df:\n")
```

```
##
## XML df:
```

```r
print(head(tour_books_xml_df))
```

```
## # A tibble: 3 x 5
##   title                       authors release_date total_pages objective_history
##   <chr>                       <chr>   <chr>        <chr>       <chr>
## 1 The Tour de France: A Hist~ Richar~ 2023-04-11   400         TRUE
## 2 Yellow Jersey: The Inside ~ Daniel~ 2019-06-03   352         TRUE
## 3 The Lance Armstrong Years:~ Daniel~ 2012-02-21   288         TRUE
```

JSON file work

Technically, I didn't need to do the as.data.frame step because json automatically comes in as a dataframe. However, my OCD kicked in and I didn't like that the page numbers in the imported dataframe structure were to the left in the cell while the othersd were to the right. Telling the dataframe to behave as a dataframe made that go away.

```
gcp_json <- "https://storage.googleapis.com/data_science_masters_files/2024_fall/data_607_data_manageme

tour_books_j <- fromJSON(gcp_json, flatten = TRUE)
tour_books_json_df <- as.data.frame(tour_books_j)

cat("\nJSON df:\n")
```

```
##
## JSON df:
```

```
print(head(tour_books_json_df))
```

```
##                                                                      title
## 1                                                The Tour de France: A History
## 2           Yellow Jersey: The Inside Story of the Greatest Cycling Race on Earth
## 3 The Lance Armstrong Years: A Cycling Legend's Rise to Power and Fall from Grace
##                         authors release_date total_pages objective_history
## 1                 Richard Moore   2023-04-11         400              TRUE
## 2 Daniel Friebe, Jeremy Whittle   2019-06-03         352              TRUE
## 3                  Daniel Coyle   2012-02-21         288              TRUE
```

Review

All three files dataframes are identical. When HTML, XML, or JSON files are imported into RStudio, R standardizes how it imports and presents structured data. Since, regardless of sematic formatting differences that support different code base usec cases, they each have key:value pairs that can be cleanly parsed into a standard table. The goal is that, no matter the data source, the data presents in a similar looking dataframe.

Where you and I may disagree is on whether JSON is structured data. I've previously heard you either say or write that you believe JSON is unustructured data (or maybe I'm completely making this up?). I view JSON as structured data that can be easily used, with minimal processing, for analytics.