## Additional Thoughts so far for revision

### \*\*7.4 Advanced JSON Handling with `httr2`\*\*

While touched on converting API responses to data frames using \*\*`jsonlite`\*\*, we could delve deeper into JSON handling:

- \*\*Parsing Nested JSON\*\*: Often, APIs return nested JSON structures. Teach how to handle and flatten these structures to convert them into usable data frames.

- \*\*JSON to List\*\*: Before converting to a data frame, JSON can be converted to a list in R. This can be useful for certain types of analyses or data manipulations.

### \*\*7.5 Caching Data with `httr2`\*\*

Given that \*\*`httr2`\*\* has caching capabilities, this can be a valuable addition:

- \*\*Why Cache?\*\*: Explain the benefits of caching, especially when dealing with rate-limited APIs or to speed up development and analysis by not fetching data repeatedly.

- \*\*Implementing Caching\*\*: Show how to use \*\*`req\_cache()`\*\* in \*\*`httr2`\*\* to cache API responses.

### \*\*7.6 Error Handling and Retries with `httr2`\*\*

Building on the capabilities of \*\*`httr2`\*\*:

- \*\*Understanding HTTP Errors\*\*: Briefly explain common HTTP errors like 404 (Not Found) or 429 (Too Many Requests) and why they might occur when working with APIs.

- \*\*Implementing Retries\*\*: Show how to use \*\*`req\_retry()`\*\* in \*\*`httr2`\*\* to automatically retry if a request fails.

### \*\*7.7 Advanced API Techniques\*\*

- \*\*Pagination\*\*: Many APIs return data in pages, especially if there's a lot of data. Teach how to handle paginated responses and fetch all data.

- \*\*Rate Limiting\*\*: Explaining what rate limiting is and how to handle it, possibly by introducing pauses in requests or by using caching.

### \*\*7.8 Introduction to Web Scraping\*\* (Altough, this might be too much.. and even if it's important, for data sourcing, it's kind of a grey area, limited by rate limits or considerations with ToS.)

While APIs are a clean way to get data, not all websites offer them. We could introduce the concept of web scraping:

- \*\*What is Web Scraping?\*\*: Briefly explain the concept and why it might be necessary.

- \*\*Using `rvest` for Web Scraping\*\*: Introduce the \*\*`rvest`\*\* package, which is part of the tidyverse and is used for web scraping. We could provide a simple example of scraping data from a website.

- \*\*Morality of Web Scraping\*\*: via. Ethical, Legal, Technical Considerations