Problem Set 4

Brent Myers
02/24/26

1 Question 5: JSON Exercise

In this exercise, I downloaded a JSON file containing historical events from a web API using `wget` within an R script. I then converted the JSON data into an R data frame using the `jsonlite` and `tidyverse` packages.

1.1 Part (d): Object Types

- `class(mydf)` returned: `"tbl_df" "tbl" "data.frame"`. This means `mydf` is a tibble, which is the tidyverse version of a data frame.
- `class(mydf$date)` returned: `"character"`. This means the date column is stored as a character string, not as a date object.

1.2 Part (e): First 6 Rows

The first 6 rows of the data frame show historical events from the year 1 AD, including events from the Roman Empire and Asia. The data frame contains 6 columns: `date`, `description`, `lang`, `category1`, `category2`, and `granularity`.

```
# A tibble: 6 x 6
  date  description                          lang  category1 category2 granularity
  <chr> <chr>                                <chr> <chr>     <chr>     <chr>
1 1     Tiberius, under order of Augustus... en    By place  Roman Em... year
2 1     Gaius Caesar and Lucius Aemilius ... en    By place  Roman Em... year
3 1     Gaius Caesar marries Livilla, dau... en    By place  Roman Em... year
4 1     Quirinius becomes a chief advisor... en    By place  Roman Em... year
5 1     Areius Paianeius becomes Archon o... en    By place  Roman Em... year
6 1     The ''Yuanshi'' era of the Chines... en    By place  Asia      year
```

2 Question 6: sparklyr Exercise - (Did Not Complete)

I was **unable** to run the sparklyr exercise because the `sparklyr` package could not be installed on OSCER. The installation failed due to dependency version conflicts—specifically, the version of `dplyr` available on OSCER (0.8.5) was too old for the required `dbplyr` package (which requires dplyr $\geq$ 1.1.2). The R script `PS4b_Myers.R` contains all of the commands that would have been executed had the package been available.

Based on the exercise instructions, the expected answers would be:

- **Step 7:** `df1` would be of class `tbl_df` (a local tibble), while `df` would be of class `tbl_spark` (a Spark DataFrame).

- **Step 8:** The column names in the Spark DataFrame use underscores instead of periods (e.g., `Sepal_Length` instead of `Sepal.Length`), because Spark does not allow periods in column names.

3 Data Sources of Interest

There are several data sources I would be interested in scraping for research purposes:

1. **SEC EDGAR:** The SEC's EDGAR database provides free access to corporate filings, including 10-K and 10-Q reports. These filings contain financial data, footnotes, and disclosures that are useful for empirical accounting research. The SEC provides an API for programmatic access.

2. **FASB Accounting Standards:** The Financial Accounting Standards Board publishes accounting standards (ASCs and ASUs) that govern financial reporting. Scraping and analyzing the text of these standards could support research on the rules-based versus principles-based nature of accounting regulation.

3. **Federal Reserve Economic Data (FRED):** The St. Louis Fed's FRED database offers over 800,000 economic time series, accessible through a free API for R and Python. This data is useful for macroeconomic and financial research.