Data Extraction Procedures and Methods

Phase 1: Extracting Data from the Bechdel Test

- Access the Bechdel Test API documentation and utilize the available free URL (https://bechdeltest.com/api/v1/getAllMovies) to extract all movies available in their database.
- Create a function using this URL to retrieve the available movies.
    - Each movie returned by this API possesses the following properties: IMDb ID, movie ID, movie title, release year, and rating.
    - The rating is a value between 0 and 3 that reflects whether the movie passed the Bechdel Test. Movies with a rating of 3 satisfy all three test criteria. A rating of 2 indicates that only two criteria are met; a rating of 1 means only one criterion is fulfilled; and a rating of 0 signifies that the movie does not meet any of the test's requirements.
- Year Filtering
    - Within the extraction function, I implemented a filter to exclude movies released before 1930. This decision is based on the fact that starting from that year, there was a reduction in silent films, which, by their nature, could hardly meet the criterion of dialogue between women (one of the Bechdel Test's requirements).
- Organizing by Release Order
    - I applied filtering to ensure that the function returns all retrieved movies in order of their release year for better data organization.

Phase 2: Cross-referencing BechdelTest Movies with OMDb

- Access the OMDb API documentation (chosen for being free and containing relevant information for our research) and generate multiple API keys. Some of the keys used to access the API were:
    - 73ea06cd
    - f5e35efc
    - 50d0eb84
    - 8e51c382
    - etc.
- Create a function that receives the IMDb ID of each movie returned by the BechdelTest API as a parameter.
    - I developed a function that takes the IMDb ID of each movie from the BechdelTest API. Since the OMDb ID format requires the "tt" prefix, I added this prefix to the ID returned by the API. The URL used to fetch each movie's details from OMDb was as follows: `https://www.omdbapi.com/?i=tt\(imdbID)&apikey=\(myapikey)`, where `\(imdbID)`

is the movie's IMDb ID and `\(myapikey)` is the personal key generated from the OMDb documentation.

Phase 3: Combining and Saving Final Movie Data*

- After querying OMDb, the obtained details were added to the `CompleteMovie` structure, which combines the BechdelTest movie information with the detailed data from OMDb. This structure contains the following properties:
  - `year`: The movie's release year (provided by BechdelTest).
  - `rating`: The BechdelTest rating (from 0 to 3, indicating how well the movie passed the test).
  - `title`: The movie title.
  - `imdbid`: The movie's IMDb ID.
  - `details`: The complete details returned by the OMDb API, including information such as cast, directors, genre, among others.

Phase 4: Filtering Extracted Data

- Create a function that classifies movies based on their nationality (USA or Others).
  - By extracting the nationality, the program counted the number of American films and non-American films.
- Create a function that categorizes movies by decade.
  - By extracting the decade, the program tallied the number of movies per decade.
- For each movie processed, the program updated an array tracking the number of extracted movies (indicating how many movies are being analyzed—2,367 non-American films and 5,228 American films).

Phase 5: Creating and Manipulating JSONs

- Create a function that transforms the extracted information into JSONs based on nationality (USA, Brazil, and Others).
- We observed that there were very few Brazilian films and decided to combine them with Others.
- Since the nationality JSONs were created every two decades (because a single API key could not support extracting all movies), after creating all JSONs, I merged them to have only one JSON per nationality (USA and Others, with Brazil merged into Others).

Phase 6: Creating and Manipulating CSVs

- Create functions that generate CSVs from the JSONs.
- CSV Structure:

  a. Movies CSV:
  - Movie title
  - Decade it belongs to
  - Test score
  - Nationality (USA and Others)
  - Genre

  b. Decade CSV:
  - Decade
  - Percentage of rating 0 (fail)
  - Percentage of rating 1
  - Percentage of rating 2
  - Percentage of rating 3 (by nationality - USA and Others)