

# Assignment 2: Data!

## Overview

In this assignment, we continue to construct a natural language query system. In the previous assignment, we wrote a procedure to *match* questions typed by a user, called **source**, with **patterns** that we define. This *match* procedure will allow our system to identify what the user is asking. The next step is to enable the system with some data to find the **answers** to the users' questions.

## More Details

Rather than answering questions about movies (as we saw in the examples in assignment 1), our chatbot will be prepared to answer questions about countries. We will be leveraging data from the CIA World Factbook<sup>1</sup>. Specifically, we'll be using the csv files provided in the "country comparison" pages. In the provided compressed file *world\_factbook.zip*, I have provided many of the csv files in a convenient structure. Please feel free to download and add more files locally for use in your chatbot.

<https://www.cia.gov/the-world-factbook/references/guide-to-country-comparisons/>

## Your job

The goal of this assignment is to write a function called *load\_csv*, that takes as input a file name - the path of a csv file that contains some world factbook country comparison data. The provided file must be a csv that includes the following fields: name, ranking, and value.

The function should open and read the provided file. It should then create a dictionary that maps the 'name' field (that is, the country name) to a list of 'ranking' and 'value' for that country.

For example, the "population" dictionary would look like...

```
{'China': ['1', '1,397,897,720'],  
 'India': ['2', '1,339,330,514'],  
 'United States': ['3', '334,998,398'],  
 'Indonesia': ['4', '275,122,131'],  
 ...}
```

Notice that all items in values/lists (e.g. ['1', '1,397,897,720']) are strings.

---

<sup>1</sup> *The World Factbook 2021*. Washington, DC: Central Intelligence Agency, 2021.  
<https://www.cia.gov/the-world-factbook/>

Your job is to come up with your own design for the *load\_csv* function and implement it.

The assert statements provided show examples of correct results for the *l* function. If implemented correctly, your code should pass ALL of these asserts. Additionally, you should create dictionaries for 2 files included but not currently tested and add 2 asserts in the same style to verify that they are functioning properly.

If stuck, review csv files (reading them) from the onramp (or simply google it). Some of you will ask if using Pandas is allowed. I think Pandas is overkill for this assignment and would prefer that you write it without it.