Being a data scientist means being prepared to work with a wide range of data sources. Thus far, I have relied solely on datasets in Excel and CSV formats. In the previous real-world examples, these datasets were easily obtained from online portals. These formats allow for easy access and mutability with Python and can also be read by programs like Tableau. While these formats are preferred for those reasons, I wanted to provide a real-world tutorial using a different format.

Another format commonly offered by data portals is JavaScript Object Notation or JSON. This format uses JavaScript syntax and is easily parsed by web browsers. While working with Python, JSON offers an effortless approach to transfer data between servers and clients. JSON is especially useful when using real-time or consistently updated data such as currency conversions or weather information. In a real-time scenario, you could access a JSON API from a web server and pull down the latest data as necessary. For the purposes of this post, I will be demonstrating a basic example using JSON to extract data and then create a simple Tableau visualization.

I started by browsing the United States government’s [Project Open Data](https://project-open-data.cio.gov/) catalog. The portal offers a filter to browse by formats such as CVS, XML, or JSON. There, I found a dataset tracking life expectancy in the U.S.A. since 1900. This particular dataset was available in CSV as well as JSON. I want to note that this file includes additional filter criteria such as gender and ethnicity, however, for the simplicity of this post, I decided to only use the “all genders” and “all ethnicities” categories.

We are going to start in Jupyter Notebook with the JSON module.

Code block 1

In the code above, we start by importing the JSON library and then open the link to the dataset in read-only mode.

Code block 2

Next, we convert the content of the file from a string to a JSON object.

Code block 3

By passing in the variable “data” from above, we can look at the JSON content in a legible format. From the code above, we can see details about the dataset. The data itself exists at the bottom of the file in a list format.

Code block 4

I set up the variables “age” and “year” as empty lists because these are the only variables I will be extracting for the visualization. Unfortunately, this dataset does not contain a Python dictionary-like format of keys and values. I, therefore, had to extract the age and year indicators using their index locations from the JSON list. I did this by using a for loop to iterate through all the items and assign the age and year variables to respective lists.

I then saved these two lists to a dictionary in key/value pairings. Finally, I used the JSON dump function to create a new file containing only the age/year dictionary.

Finally, I simply copied the contents of the new JSON file into two columns in a new Excel spreadsheet. This new file was then read by Tableau Desktop to create the visualization below.