USC Viterbi
School of Engineering

DSCI 510: Principles of Programming for Data Science

Name: Lab Assignment 10
Due Date: November 12th, 2024 at 4pm PT

Deliverable: A python file named **run.py**

1 Introduction

In this lab, you will continue to build on last week's work with U.S. Presidents, focusing on extracting and analyzing data using web scraping and JSON manipulation. This week, we'll also introduce data consolidation using pandas.

2 Presidents' Party and Terms in JSON Format

Building on last week's assignment, you will create a dictionary where each president's name is the key, and the value is a list containing their political party and the number of terms served. After creating the dictionary, convert it to JSON format.

`https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States`

Instructions:

- Scrape the Wikipedia page for U.S. Presidents.
- For each president, retrieve their political party and the number of terms served.
- Construct a dictionary with the following structure:

```
{
    "George Washington": ["None", 2],
    "John Adams": ["Federalist", 1],
    ...
}
```

Function Specification:

- **Function Name:** `get_president_terms`
- **Arguments:** None
- **Returns:** None (output saved in `presidents_terms.json`)

Example

```
get_president_terms()
# Output saved in presidents_terms.json
```

3 Calculating Approval Rating Changes in JSON Format

In this problem, you will retrieve data from a JSON endpoint hosted at `https://dsci.isi.edu/slides/data/presidents`, which contains each president's approval ratings. You will calculate the difference between each president's approval rating at the start and end of their term, store the result in a dictionary, and convert it to JSON format.

Instructions:

- Use `requests` to fetch data from the endpoint `https://dsci.isi.edu/slides/data/presidents`.
- For each president, calculate the difference in approval rating as `end - start`. If a president doesn't have an `end` rating, skip that entry.
- Construct a dictionary where each president's name is the key, and the value is the calculated approval rating change:

```
{
    "John F. Kennedy": -8,
    "George H. W. Bush": 5,
    ...
}
```

Function Specification:

- **Function Name:** `calculate_approval_changes`
- **Arguments:** None
- **Returns:** None (output saved in `approval_changes.json`)

Example

```
calculate_approval_changes()
# Output saved in approval_changes.json
```

4 Consolidating Data into a DataFrame

For this final problem, you will read both JSON files created in Problems 1 and 2 and consolidate them into a pandas DataFrame.

Instructions:

- Load data from `presidents_terms.json` and `approval_changes.json`.
- Create a DataFrame with columns `President`, `Party`, `Terms`, and `Approval Change`.
- If a president is missing data from the approval changes, mark `NaN` in the `Approval Change` column.
- Display the resulting DataFrame.

Function Specification:

- **Function Name:** `generate_president_dataframe`
- **Arguments:** None
- **Returns:** A pandas DataFrame with columns as specified.

Example

```
df = generate_president_dataframe()
print(df)
```