Quorum Legislative Data Tech Test

Table of contents

Problem statement

We’re analyzing publicly available government data. We need to provide our clients with the ability to visualize all of the bills that legislators voted for or against.

Goals

Given the datasets, we must answer the following questions:

1. For every legislator in the dataset, how many bills did the legislator support (voted for the
   bill)? How many bills did the legislator oppose?
2. For every bill in the dataset, how many legislators supported the bill? How many legislators
   opposed the bill? Who was the primary sponsor of the bill?

Non-goals

- Spend more than 2~3 hours doing this project
- Do a complicated solution
- Add complex things like a database

Proposed solution

With Pandas as our data analysis tool, we will make some joins and count and/or distinct votes by bills and legislators.

Our program must output two different `.csv` files, each of them should answer our questions (described in the goals section).

Risks

All the data comes from `.csv` files, if their size increases, we must be aware of the time to load and perform operations over the data.

Each operation (merging tables, pivoting, writing `.csv` files) has different time complexity because of the size and group of data they use to perform.

As of now, the risks are low.

Time complexity of operations

- `pd.read_csv()` and `df.to_csv()` is O(n), where *n* is the number of rows in the file.

- `pd.merge()` depends on the size of the DataFrames being merged and the join type. For example, if we're performing an inner join, the time complexity is O(n * m), where *n* and *m* are the numbers of rows in the two DataFrames being merged. However, if we're performing a left join (which is our case), the time complexity is O(*n* + *m*), because we only need to loop through each row in the left DataFrame once.
- `df.groupby()` is O(*n* log *n*), where *n* is the number of rows in the DataFrame being grouped.
- `df.pivot_table()` is O(n log n), where n is the number of rows in the DataFrame being pivoted.

Does this project have any dependencies?

- Python 3
- Pandas library
- Numpy

Alternative solutions

I considered using Node.js because it is the runtime that I feel comfortable with and to use some panda-like library to achieve our goals. Even tho we have a similar library in the Javascript ecosystem, the size of the community (people using, talking about, and more) is considerably smaller, which makes things hard if we need to google for a problem or examples.

Future improvements

New columns

To account for new columns, such as **Bill Voted On Date** and **Co-Sponsors**, we have to consider where we will gather these columns from.

We can assume that **Bill Voted on Date** will be at the `bills.csv` or at Bill's data source. The same goes for **Co-Sponsors** (an integer that points to the id of the sponsor - Person).

Different types of input

We might need to modify our code to receive a list of legislators or bills and generate a `.csv` report for them.

To account for this new requirement, we need to modify the input of data and how we process them.

With a list of legislators or bills, we need to load more data from a data source (database or a server) and then process like we are doing here.