Clearing up some project 3 work

November 29, 2023

Administrative Notes

The purpose of today’s lecture:

- To clear up a lot of the mess from Monday
- To make it easier to get Project 3 done

First: Conda

- I tried to avoid the use of Conda to install Python packages in this course, because in general pip is easier.
- But there are a few students every semester who can't get jupyter notebooks installed/running using pip
  - Monday I advised students to just use Jupyter running in your browser, without installing it
  - It turns out that won't work because you are not able to open local files on your laptop with that version of Notebook
- So we're going to go ahead and recommend that people who can't get this working with pip try Conda
- There's an announcement on Blackboard from Tuesday explaining how
  - Let's look at that

Moving on to the Monday mess

Data structures: a core concept of Computer Science

My early solution to this problem involved creating a list of states as a constant, like this:

```
STATES= ["Alabama","Alaska","Arizona","Arkansas","California","Colorado",
"Connecticut","Delaware","Florida","Georgia","Hawaii","Idaho","Illinois",
  "Indiana","Iowa","Kansas","Kentucky","Louisiana","Maine","Maryland",
  "Massachusetts","Michigan","Minnesota","Mississippi","Missouri","Montana",
  "Nebraska","Nevada","New Hampshire","New Jersey","New Mexico","New York",
  "North Carolina","North Dakota","Ohio","Oklahoma","Oregon","Pennsylvania",
  "Rhode Island","South Carolina","South Dakota","Tennessee","Texas","Utah",
  "Vermont","Virginia","Washington","West Virginia","Wisconsin","Wyoming"]
```

I then used that constant to merge lists and build data structures I needed for the later parts of the problem

- I hadn't properly prepared to present that in class, violating the "Rule of the 6 P's"
- "Proper Prior Preparation Prevents Poor Performance"

A note:

I cleaned up the code that produced the 2D list I called “core_results”

- I used .strip(‘\”’) to get rid of all the double-quote marks to make it easier to read and match
- I converted year, and both vote totals, to integers to facilitate graphics later on

But there’s a better way

So walking back to the office after class on Monday, I realized that this is Python, and Python uses dictionaries, and this problem is just tailor-made for a great big dictionary

- I posted a video on the YouTube channel Monday night explaining it, but I want to walk through it now.

Look at the rest of the assignment and see what data you will need to produce the requested graphics

That data would be easy to produce if you had a dictionary whose keys were the states

The values would themselves be a dictionary summarizing the results for each state

Let’s build that

Jupyter notebook walk-through
You want a dictionary with states as keys:

{ “ALABAMA”: {},

“ALASKA”: {},

…

}

Details, please?

As you can see, the value associated with each key is itself going to be a dictionary.

What's in this inner dictionary?

It seems like we ought to have each election year as a key.

And as a value for each key - each election year - we should have a list consisting of the two party names, and the number of votes each party got

- See next slide for illustration

{

“ALABAMA”::

{1976: ['DEMOCRAT', 659170, 'REPUBLICAN', 504070],,

1980: ['REPUBLICAN', 654192, 'DEMOCRAT', 636730],

1984: ['REPUBLICAN', 872849, 'DEMOCRAT', 551899]],

…}

“ALASKA”:

{1976: ['REPUBLICAN', 71555, 'DEMOCRAT', 44058],

1980: ['REPUBLICAN', 86112, 'DEMOCRAT', 41842],

1984: ['REPUBLICAN', 138377, 'DEMOCRAT', 62007],

…}

}

How do we build this structure? An example

```
results = {}
for i in range(len(core_results)):
    year = core_results[i][0]
    state = core_results[i][1]
    party = core_results[i][2]
    votes = core_results[i][3]
    if not state in results.keys():
        results[state] = {year:[party,votes]}
    else:
        if not year in results[state].keys():
            results[state][year] = [party, votes]
        else:
            results[state][year].append(party)
            results[state][year].append(votes)
```

Now let’s test that code

```
"""
This does NOT quite solve step 8 in the assignment; it's to illustrate how this works
and what the results are
"""
for state in results.keys():
    d = 0
    r = 0
    for year in results[state].keys():
        if results[state][year][0] == 'DEMOCRAT':
            if results[state][year][1]> results[state][year][3]:
                d += 1
            else:
                r += 1
        else:
            if results[state][year][1]> results[state][year][3]:
                r += 1
            else:
                d += 1
    print (state)
    print ("D wins: ", d)
    print("R Wins ", r)
```

So now we need some graphics

Okay, let’s start using matplotlib to produce graphics providing the answers to the last few parts ot the project

(more Jupyter notebook work)