Presidential Election Simulator

A Exploration of Alternate Election Scenarios - Gary Gregg

U.S. Presidential Elections Are Indirect

- The U.S. Constitutional Convention agreed in 1787 on an Electoral College for the purpose of electing the President and the Vice President
- Each state is allocated a number of electors in proportion to its population (same as its Congressional representation), plus two (same as its Senate representation)
- Electors are reapportioned every ten years based on the results of the most recent census (last, 2010)
- 48 states use a winner-take-all approach to determine their electors; a candidate who wins a plurality of the popular vote in these states selects a slate of electors pledged to him (or her)
- 2 states (Maine and Nebraska) assign one elector for a candidate who wins the popular vote in each Congressional districts; two electoral votes then go to a candidate who wins the state at large
- To win the Presidency in the Electoral College, a candidate must win a majority of the electoral votes

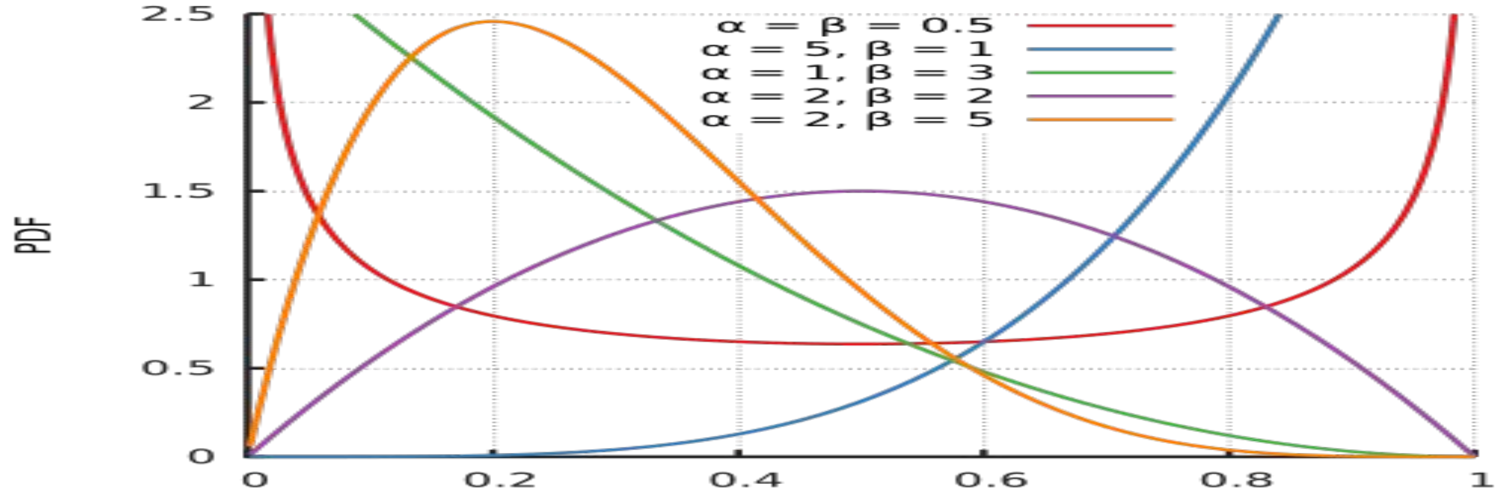
Why Simulate Elections?

- In 1876, 1880, 2000 and 2016 - due primarily to the winner-take-all approach to elector allocation - candidates not receiving at least a plurality of the popular vote became President
- In 1876, above, the candidate who received a **majority** of the popular vote did **not** become President
- In 1824, the candidate who won a plurality of both popular and electoral votes did **not** become President
- More often, the Electoral College magnifies the results of a popular vote win, and does not invert it
- This seemingly unfair system is difficult to explain to new students of politics, and to foreign observers
- Simulations can show how small and random changes in voting can affect the results of close elections
- Slightly changing the voting sentiment in even one key state can affect the results of close elections

Related Work and References

- 270-to-Win has a 2016 Presidential election simulator at (https://www.270towin.com/2016-simulation/)
- 270soft.com has a Presidential election game at (https://270soft.com/us-election-games/president-election-game-2016/)
- Other election simulators do not seem to offer precise mathematical control over the random selection of state-specific election results
- Referencing *Introduction to Probability*, by Blitzstein and Hwang, I determined that the Beta probability distribution is appropriate for binary election results
- A beta distribution can be determined by two parameters, α and β
- If α and β are both greater than one, and equal, the shape of the PDF is a bell curve between 0 and 1

The Beta Distribution



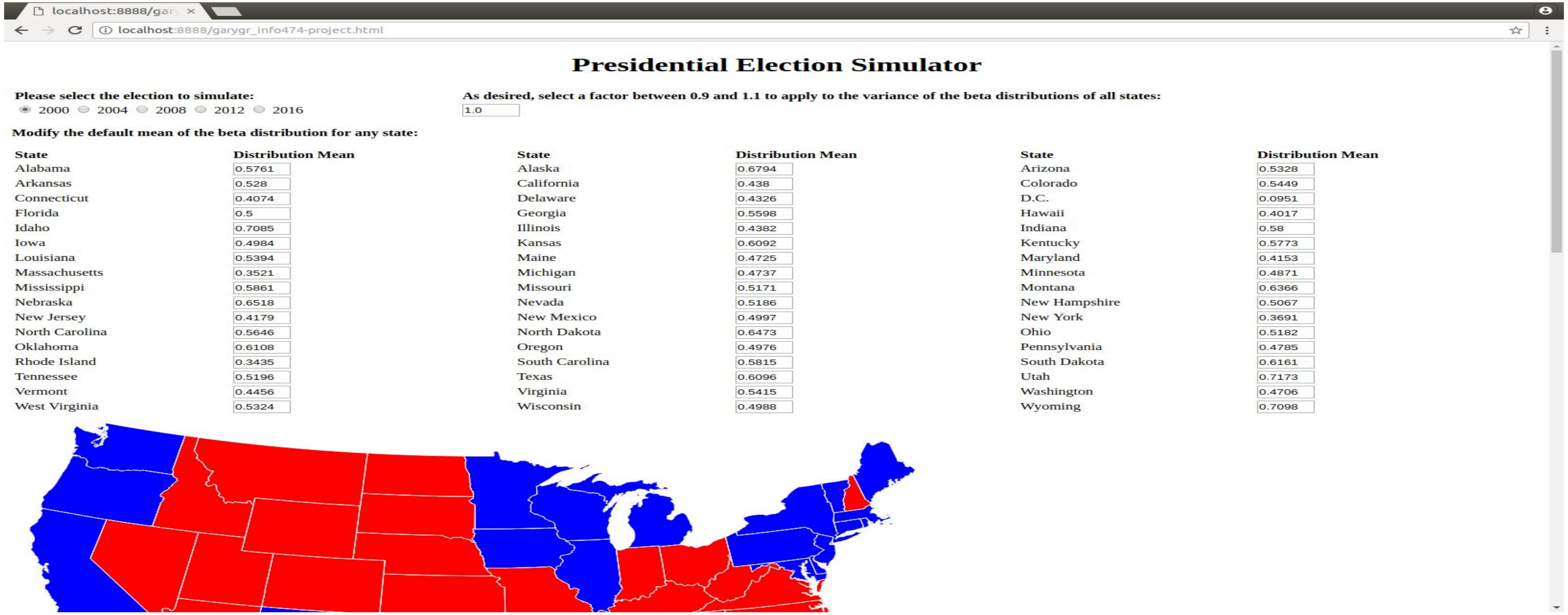
Selecting the Parameters

- Beta distributions can also be uniquely determined by a mean, and a variance
- The current political alignment of the states has been in place since about 2000
- The last twenty years encompass two censuses, and therefore two state elector reapportionments
- I chose to simulate any of the last five Presidential elections
- For the mean of the beta distribution for each state in each election year, I use the **actual election result for that year**
- For the variance of all beta distributions for each state in each election year, I use the variance of the **actual election results in all five elections**
- Less populated states tend to have greater variance; more populated states have less
- Question: Is this a statistically defensible technique? I could not think of anything better

Demo

- I had an idea that I would use a tabular format for user modification of distribution means and variances, plus a tabular format for presentation of simulation results showing popular votes and EVs
- I also wanted to use a variation of a D3 U.S. map created by Mike Bostick (see https://bl.ocks.org/mbostock/4060606)
- Because I had a good conception of what graphical devices I wanted to employ, I bypassed the hand-drawn storyboard phase
- I am a far more experience Java programmer than JavaScript, and I conceived of the idea of coding the simulation engine in Java; the HTML and JavaScript of the visualization must be linked to it
- The simulation engine is not done; the current demo allows only exploration of actual election results
- Possible visualization enhancement: mouseover display of state-specific election results

Demo Screenshot Upper



Presidential Election Simulator

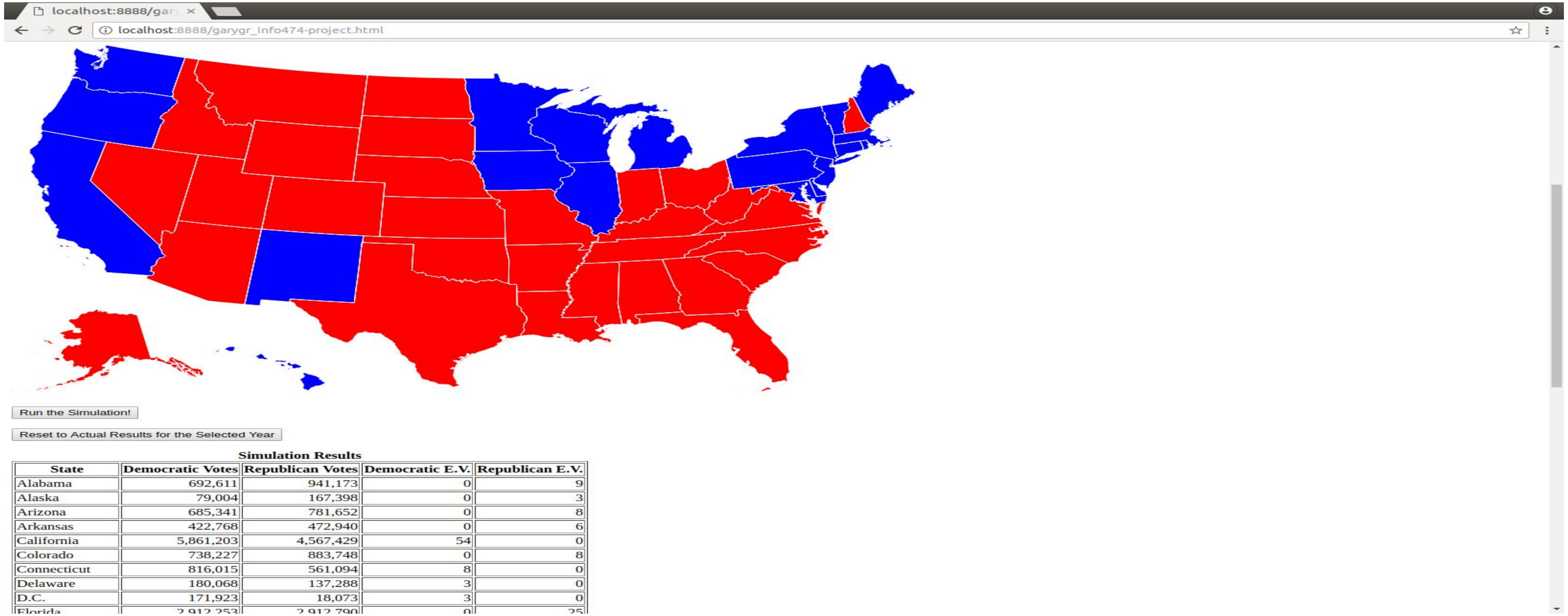
Please select the election to simulate:
[✓ 2000] 2004 2008 2012 2016

As desired, select a factor between 0.9 and 1.1 to apply to the variance of the beta distributions of all states:
1.0

Modify the default mean of the beta distribution for any state:

State	Distribution Mean	State	Distribution Mean	State	Distribution Mean
Alabama	0.5761	Alaska	0.6794	Arizona	0.5328
Arkansas	0.528	California	0.438	Colorado	0.5449
Connecticut	0.4074	Delaware	0.4326	D.C.	0.0951
Florida	0.5	Georgia	0.5598	Hawaii	0.4017
Idaho	0.7085	Illinois	0.4382	Indiana	0.58
Iowa	0.4984	Kansas	0.6092	Kentucky	0.5773
Louisiana	0.5394	Maine	0.4725	Maryland	0.4153
Massachusetts	0.3521	Michigan	0.4737	Minnesota	0.4871
Mississippi	0.5861	Missouri	0.5171	Montana	0.6366
Nebraska	0.6518	Nevada	0.5186	New Hampshire	0.5067
New Jersey	0.4179	New Mexico	0.4997	New York	0.3691
North Carolina	0.5646	North Dakota	0.6473	Ohio	0.5182
Oklahoma	0.6108	Oregon	0.4976	Pennsylvania	0.4785
Rhode Island	0.3435	South Carolina	0.5815	South Dakota	0.6161
Tennessee	0.5196	Texas	0.6096	Utah	0.7173
Vermont	0.4456	Virginia	0.5415	Washington	0.4706
West Virginia	0.5324	Wisconsin	0.4988	Wyoming	0.7098

Demo Screenshot Lower


localhost:8888/gar
localhost:8888/garygr_info474-project.html
Run the Simulation!
Reset to Actual Results for the Selected Year

Simulation Results

State	Democratic Votes	Republican Votes	Democratic E.V.	Republican E.V.
Alabama	692,611	941,173	0	9
Alaska	79,004	167,398	0	3
Arizona	685,341	781,652	0	8
Arkansas	422,768	472,940	0	6
California	5,861,203	4,567,429	54	0
Colorado	738,227	883,748	0	8
Connecticut	816,015	561,094	8	0
Delaware	180,068	137,288	3	0
D.C.	171,923	18,073	3	0
Florida	2,912,253	2,912,790	0	25

Timeline of Tasks

- I worked alone on this project due to a large, concurrent collaborative project for another course
- I used R and RStudio to help manage and auto generate CSV files, and painstakingly entered election results by hand from Wikipedia using their articles for the 2000 through 2016 elections
- Limiting to major party candidates, I calculated results percentages for each state in each year
- I calculated overall election variance for each state considering all five elections
- I added each state’s electoral vote assignments for each election
- I determined that Apache supplies a Java software package for the Beta distribution, integrated this, and wrote a sample Java interface for the simulation engine
- I wrote the HTML and JavaScript code to present the interactive visualization you saw
- Next: Write the Java simulation engine, and integrate it with the visualization

Limitations and Conclusion

- The simulator does not consider third-party candidates
- The simulator assumes winner-take-all electoral votes in Maine and Nebraska; the actual results presented for the 2008 and 2016 elections are therefore off by one electoral vote
- Due to the above, simulation results will also be winner-take-all for Maine and Nebraska
- Thank you! Are there any questions?