CS/INFO 3300 / INFO 5100
Homework 1
Due 11:59pm Tuesday 2/9

Assume you are a pollster trying to guess who will win an election. In fact, 42% of voters support Trump, 38% of voters support Cruz, and 20% support Rubio.

But you don't know that. You can only ask a small number of voters for their opinion, and guess from those responses. How likely is it that you will correctly predict that Trump will win? How much does it depend on the number of voters you ask?

Your work should be in the form of an HTML file called `index.html` with one <p> element per problem. Wrap any Javascript code for each problem in a <script> element. Each script should display sample For example:

```
<p>Problem 0: We use the var statement to declare a variable and
set it to a value.
<div id="p0output"></div>
<script>
var x = 100;
document.getElementById("p0output").innerText = x;
</script>
</p>
```

Create a zip archive containing this file and upload it to CMS.

You may log messages to the console for debugging purposes, but all function output should happen through `return` values. For each problem, you may use your work from the previous problem (though you don't have to). Do not use any other external libraries or resources (Math.random() is ok, jQuery is not).

1. [Voter simulator] Write a function called `uninformedVoter` that returns a random value from `["trump", "cruz", "rubio"]`, with equal (*uniform*) probability. Display a sampled vote (that is, set the contents of the output div to the output of the function). (15 pts)

2. [More realistic voter simulator] Write a function called `voterGenerator` that takes an array of probabilities (three positive numbers that sum to 1.0), and returns

a function. This returned function should take no arguments and return a random value from `["trump", "cruz", "rubio"]`, with the specified probabilities. Use `voterGenerator` to create a function `likelyVoter` with the "correct" probabilities `[0.42, 0.38, 0.2]`. Display a sampled vote from this "likely voter" function. (Note that despite the name this function doesn't represent a specific person, but rather the distribution of what response you would get if you truly sampled randomly from likely voters. Calling it multiple times may produce different values.) (15 pts)

3. [Poll simulator] Write a function called `pollN` that takes two arguments, a number `n` and a function `voter`, and returns an array containing `n` votes from the `voter()` function. For example, calling `pollN(3, likelyVoter)` might return `["trump", "trump", "rubio"]`. Display the results of a poll of 20 likely voters. (20 pts)

4. [Declaring victors] Write a function called `winner` that takes two arguments, a number `n` and a function `voter`, polls `n` voters, and returns the name of the candidate with the most support (break ties randomly). Display the winner of a poll of 20 likely voters. (10 pts)

5. [Repeated polling] Write a function called `multiplePolls` that takes three arguments, a number of voters `n`, a number of polls `k`, and a function `voter`. It should run `k` polls, each with `n` voters, and return the number of correct (Trump) victories. Display the result of this function with 5 polls, each with 100 likely voters. (20 pts)

6. [Sample size, margin of error] Try different numbers of polls and different numbers of voters. Write answers to the following in the output tag for this problem. Explain your answer and show examples (which can be dynamically generated or copy/pasted in). Approximately how many voters do you need to ask in order to get the correct answer most of the time? If you hold the number of voters constant, is it sufficient to average over more polls? [10 pts]

7. Generate a "likely voter" function from a different state, with different "correct" preferences. Display the result of the `multiplePolls` function using your new voter model. Does your new distribution affect the number of voters needed to get an accurate poll? Why or why not? [10 pts]