### Solution Review: Predicting Election Results

This lesson discusses the solution review to predicting election results.

# WE'LL COVER THE FOLLOWING ^ Solution 1 Explanation Solution 2 Explanation

# Solution 1#

```
import numpy as np
import numpy.random as rnd

rnd.seed(2)
people = np.zeros(1000000, dtype='int') # arthur is 0
people[490001:] = 1 # ben is 1
poll = rnd.choice(people, 1000)

polled_for_arthur = len(poll[poll == 0])

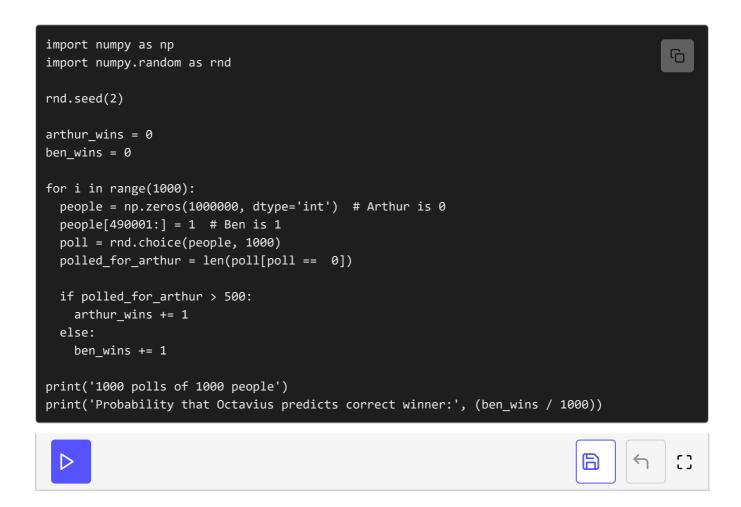
print('polled for A:', polled_for_arthur)
if polled_for_arthur > 500:
    print('Octavius will predict the wrong winner')
else:
    print('Octavius will predict the correct winner')
```

## **Explanation** #

- In line 5, we have created an array of zeros of length 1000000. We then replace the last 510000 zeros with 1.
- In line 7, we randomly choose 1000 elements from people using the method choice and store it in poll.

- In line 9, we compute the total number of people who voted for Arthur.
- Since the number of 0's in poll turns out to be greater than 500, they misrepresent the actual outcome.

# Solution 2 #



# Explanation #

- The code above is similar to the code in solution 1, except that we run the code for 1000 times, using a for loop, in lines 9 18.
- For every loop, we use the if condition in line 15 to check whether Octavius predicted the right result or not.
- Depending on the value of polled\_for\_arthur, we store the result in
  either arthur\_wins or ben\_wins
- In line 21, we compute the probability that Octavius predicted the correct

withter in 1000 poils of 1000 people.

With this solution review, we are done with the random variables chapter. Let's move on to the applications part of the course.