(/learn)

Probability Calculator

You will be working on this project with our Replit starter code (https://replit.com/github/freeCodeCamp/boilerplate-probability-calculator).

Suppose there is a hat containing 5 blue balls, 4 red balls, and 2 green balls. What is the probability that a random draw of 4 balls will cont at least 1 red ball and 2 green balls? While it would be possible to calculate the probability using advanced mathematics, an easier way is t write a program to perform a large number of experiments to estimate an approximate probability.

For this project, you will write a program to determine the approximate probability of drawing certain balls randomly from a hat.

First, create a `Hat` class in `prob_calculator.py`. The class should take a variable number of arguments that specify the number of balls each color that are in the hat. For example, a class object could be created in any of these ways:

```
hat1 = Hat(yellow=3, blue=2, green=6)
hat2 = Hat(red=5, orange=4)
hat3 = Hat(red=5, orange=4, black=1, blue=0, pink=2, striped=9)
```

A hat will always be created with at least one ball. The arguments passed into the hat object upon creation should be converted to a `contents` instance variable. `contents` should be a list of strings containing one item for each ball in the hat. Each item in the list should a color name representing a single ball of that color. For example, if your hat is `{"red": 2, "blue": 1}`, `contents` should be `["red", "red", "blue"]`.

The `Hat` class should have a `draw` method that accepts an argument indicating the number of balls to draw from the hat. This method should remove balls at random from `contents` and return those balls as a list of strings. The balls should not go back into the hat during t draw, similar to an urn experiment without replacement. If the number of balls to draw exceeds the available quantity, return all the balls.

Next, create an `experiment` function in `prob_calculator.py` (not inside the `Hat` class). This function should accept the following arguments:

- `hat`: A hat object containing balls that should be copied inside the function.
- `expected_balls`: An object indicating the exact group of balls to attempt to draw from the hat for the experiment. For example, to determine the probability of drawing 2 blue balls and 1 red ball from the hat, set `expected_balls` to `{"blue":2, "red":1}`.
- `num_balls_drawn`: The number of balls to draw out of the hat in each experiment.
- `num_experiments`: The number of experiments to perform. (The more experiments performed, the more accurate the approximate probability will be.)

The `experiment` function should return a probability.

For example, if you want to determine the probability of getting at least two red balls and one green ball when you draw five balls from a h containing six black, four red, and three green. To do this, you will perform `N` experiments, count how many times `M` you get at least two balls and one green ball, and estimate the probability as `M/N`. Each experiment consists of starting with a hat containing the specified ball drawing several balls, and checking if you got the balls you were attempting to draw.

Here is how you would call the `experiment` function based on the example above with 2000 experiments:

```
hat = Hat(black=6, red=4, green=3)
probability = experiment(hat=hat,
                  expected_balls={"red":2,"green":1},
                  num_balls_drawn=5,
                  num_experiments=2000)
```

Since this is based on random draws, the probability will be slightly different each time the code is run.

Hint: Consider using the modules that are already imported at the top of `prob_calculator.py`. Do not initialize random seed within `prob_calculator.py`.

Development

Write your code in `prob_calculator.py`. For development, you can use `main.py` to test your code. Click the "run" button and `main.py` will run.

The boilerplate includes `import` statements for the `copy` and `random` modules. Consider using those in your project.

Testing

The unit tests for this project are in `test_module.py`. We imported the tests from `test_module.py` to `main.py` for your convenience. The tests will run automatically whenever you hit the "run" button.

Submitting

Copy your project's URL and submit it to freeCodeCamp.

Solution Link

ex: https://replit.com/@camperbot/hello

I've completed this challenge

Get a Hint (https://forum.freecodecamp.org/t/462364)

Ask for Help