

day38 rolling the dice
Due Monday 11/27/23

Here's the setup: Your sister is taking a probability math class, and one of her problems is to calculate the probability of tossing a pair of six sided dice and coming up with a seven. She calculated 0.167 to the nearest three places, but it's an even-numbered problem, so the answer is not in the back of the math book. She asks you to toss a pair of dice a hundred thousand times and count the number of sevens and divide by a hundred thousand and see if you get something close to 0.167. Instead of doing that, you decide to write a Python program to simulate the tossing of a pair of dice a hundred thousand times, and calculate the probability of tossing dice that add to seven.

Now, to do this we'll have to review a little bit about probability. In math, probability is the number of times something happens divided by the number of possible outcomes. For example, if you flip a penny, the probability that it will land with the head side up is $1/2$, because there is one side of a normal coin that has a head on it, and there are two sides of the coin total. If you flip a coin 50 times, you can count how many times heads turned up and divide it by 50 (the total number of flips) to get what we call the experimental probability of a coin landing heads.

In the end, this problem isn't that complicated: we need to pretend to roll two dice 100,000 times. We'll use a for loop for this (for x in range(100000):) Inside the loop we need to pretend to roll two regular six sided dice. We add the two dice together, and if they add up to 7, we add 1 to a counter. When the 100,000 tries are done, we simply print the count divided by 100,000.

Some tips:

1. Put "import random" up at the top of your program.
2. Create a count variable and set it to 0.
3. Make a for loop (for x in range(100000):)
4. Inside the for loop, store a random number from 1 to 6 in a variable diceOne. You can do this like this:

```
diceOne = random.randint(1,6)
```
5. Store another random number (same way) into a variable diceTwo.
6. Use an if statement to check if diceOne + diceTwo is equal to 7. If it is, add one to the count variable.
7. After the for loop, print count divided by 100000.

This might seem hard, but let's all agree that it's obviously much easier than actually rolling two dice one hundred thousand times, adding the numbers, and counting how many times the numbers add up to 7!

Print your results like this (your exact values will vary, but should be close to mine):

```
Probability of rolling 2 dice that add to 7: 0.16947  
( 16947 rolls out of 100,000. )
```

Continued on next page

That was part 1. Here is part 2:

Now, copy your code and paste it below what you've done so far. Change this second block of code so that you are rolling three dice. Count how many times that rolling three dice gets you three 1s. (You can check if adding all three adds up to 3 to see if this has happened). You can do it like this:

```
if diceOne == 1 and diceTwo == 1 and diceThree == 1:
```

Report your results in the same way as before like this (again, your numbers will be different, but close to mine):

```
Probability of rolling 3 dice and getting 3 ones: 0.00465  
( 465 rolls out of 100,000. )
```

Here is part 3:

Copy your second block of code and paste it below what you've done so far.

Modify this last block of code to roll 5 dice. Count and report how many dice are all the same when you roll them, which in technical terms is called a "Yahtzee". You can do this like this:

```
if diceOne == diceTwo == diceThree == diceFour == diceFive:
```

This line checks and counts all the times that all five dice are the same number. Report it like this, and as usual, your values will be different from mine, but close:

```
Probability of rolling a Yahtzee: 0.00072  
( 72 rolls out of 100,000. )
```

Here is part 4:

Figure out a way to count how many of what value Yahtzees were found and print out the results like this:

```
Probability of rolling a Yahtzee: 0.00064  
( 64 rolls out of 100,000. )  
Yahtzees of 1: 14  
Yahtzees of 2: 10  
Yahtzees of 3: 10  
Yahtzees of 4: 9  
Yahtzees of 5: 9  
Yahtzees of 6: 12
```

One way to do this involves a list. You can create a list that has a bunch of zeroes in it, then every time you get a Yahtzee, you add one to that spot on the list (`count[diceOne] = count[diceOne] + 1`).

An easier way to do this part is to create one variable for each type of Yahtzee (for example, `y1 = 0`, `y2 = 0`, and so on). Then, inside your part 3 code, if you found a

Yahtzee, do a check to see if `diceOne == 1`, and if it is, add one to `y1`. Do the same for if `diceOne == 2`, and so on. Then you will have how many Yahtzees were 1s in `y1`, and so on.

Finally, part 5:

Add a block of code that counts how many times the five dice give you 1, 2, 3, 4 and a 5 (one of each number). Print out the results like this:

```
Probability of rolling 5 dice and getting 1, 2, 3, 4, and 5: 0.01568  
( 1568 rolls out of 100,000. )
```

To figure out if you have one of each of 1, 2, 3, 4, and 5, you can create an empty list, then append all the dice to it, then do something like this:

```
if 1 in dice and 2 in dice and 3 in dice and 4 in dice and 5 in dice:
```

In all of your tests, make sure your results are close to mine (or you did it wrong!)