Lab 02 - Probability

Trần Lương Quốc Đại, Nguyễn Quốc Bình
{tlqdai, nqbinh}@it.tdt.edu.vn

25/02/2022

In probability theory, *sample space* of an experiment, or a test, E is a set of all outcomes that may occur. Sample space is usually denoted by S, Ω. For example, in the dice rolling experiment, the sample space of the experiment is a collection of all possible face of the dice after rolling, $\Omega = \{1, 2, 3, 4, 5, 6\}$.

Any subset of the sample space is called *an event*. For example, in the dice experiment, the even face event is 2, 4, and 6. When the subset of the sample space contains only one element, it is called *the basic event*.

1. Define function P

The probability is simply a fraction in which the numerator is the number of events occurring and the denominator is the total number of cases in the sample space.

Calculate the probability of an event even when roll a dice.

```
from fractions import Fraction

def P(event, space):
    '''The probability of an event, given a sample space of equiprobable outcomes.'''
    return Fraction(len(event & space), len(space))

D = {1, 2, 3, 4, 5, 6}
even = {2, 4, 6}

print(P(even, D))
```

The result is *Fraction(1, 2)*.

2. Urn Problems

A closed urn has 23 balls: 8 white, 6 blue and 9 red. Randomly select 6 balls (know that each ball has the same probability of selection). What is the probability of getting:

- Problem 1: all 6 balls are red
- Problem 2: 3 blue balls, 2 white balls and 1 red ball
- Problem 3: exactly 4 white balls

Firstly, we set the symbol for each white ball to be ‘W1’ to ‘W8’ with the first character ‘W’ is the color and the second number character is the order of the ball. Similarly 6 blue balls are ‘B1’ to ‘B6’ and 9 red balls are ‘R1’ to ‘R9’.

```python
urn = cross('W', '12345678') | cross('B', '123456') | cross('R', '123456789')
print(urn)
```

Each event is a set of 6 balls, so the sample space will be the set of all possible cases of 6 balls.

$$U6 = C_{23}^{6} = \frac{23.22.21.20.19.18}{6!} = 100947$$

```python
import itertools

def combos(items, n):
    '''All combinations of n items; each combo as a concatenated str.'''
    return {' '.join(combo) for combo in itertools.combinations(items, n)}

U6 = combos(urn, 6)

print(len(U6))
```

Take 10 random subsets from set U6.

```python
import random

print(random.sample(U6, 10))
```

The data is saved in a list as follows:

[‘R7 W3 B1 R4 B3 W2’,
‘R7 R3 B1 W4 R5 W6’,
‘B5 B4 B6 W1 R3 R2’,
‘W3 B1 R4 W8 R9 W5’,
‘W7 B6 W1 R3 R9 B3’,
‘W3 W1 R8 R2 R1 W5’,
‘B5 R7 R4 W8 R2 W6’,
‘W7 W1 R3 W4 R5 B3’,
‘B4 R7 B2 B1 W6 W2’,
‘W7 B1 W4 W8 W6 W2’]

Problem 1: All 6 balls are red

```
red6 = {s for s in U6 if s.count('R') == 6}

P(red6, U6)
```

Problem 2: 3 blue balls, 2 white balls and 1 red ball

```
b3w2r1 = {s for s in U6 if s.count('B') == 3 and s.count('W') == 2 and s.count('R') == 1}

P(b3w2r1, U6)
```

Problem 3: exactly 4 white balls

```
w4 = {s for s in U6 if s.count('W') == 4}

P(w4, U6)
```

3. Coin experiment

Write a program to calculate the probability of getting the tail when toss a coin in n times. The larger number of tests, the closer to 0.5 we get.

The function $random.randint(0, 1)$ returns a random number of 0 or 1. Conventionally, the tail is 1 and the head is 0.

```
import random
n = 10
count = 0;
for simulations in range(n):
    tosses = random.randint(0, 1)
    if tosses == 1:
        count += 1
print(count/n)
```

Running the above program with n is 10, 100, 1000, etc., and observing the results.

4. Dices experiment

Roll two dices. Write a program to calculate the probability of the even that the first dice gives 1 and the other gives 6.

The function $random.randint(1, 6)$ returns a random integer number from 1 to 6.

```
import random
count = 0;
n = 1000000;
for i in range(n):
    die1 = random.randint(1, 6)
    die2 = random.randint(1, 6)
    if die1==1 and die2==6:
```

```python
        count += 1
print(count/n)
```

Running the above program with n is 10, 100, 1000, etc., and observing the results.

5. Cards experiment

All cards are divided into 4 suits including hearts (♥), diamonds (♦), clubs (♣) and spades (♠). In each suit there are 13 cards including a 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, J (jack), Q (queen), K (king).

Define cards as:

```python
# Import the required function libraries
from itertools import product

# Define ranks, suits and cards
Ranks = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 'J', 'Q', 'K'}
Suits = {'♡', '♢', '♣', '♠'}
Cards = list(product(Ranks, Suits))
print(len(Cards))
print(Cards[:10])
```

Write a function to calculate the experimental probability of drawing the red card with the number of tests n as the parameter:

```python
def simulator_poker(n):
    count = 0;
    for i in range(n):
        index = random.randint(0, 51)
        if Cards[index][1] == '♡' or Cards[index][1] == '♢':
            count += 1
    return count/n
```

Call above function with n = 10, 100, 1000, 10000, ...

```python
print(simulator_poker(10))
print(simulator_poker(100))
print(simulator_poker(1000))
print(simulator_poker(10000))
```

6. Exercises

1. Write a simulator function of rolling two dices n times. Calculate the experimental probability of the event that both dices are even.

   ```python
   def simualtor_dice1(n):
       # your code
   ```

2. Write a simulator function of rolling two dices n times. Calculate the experimental probability of the event that one dice is even and the other is odd.

```
def simualtor_dice2(n):
    # your code
```

3. Write a simulator function of rolling two dices n times. Calculate the experimental probability of the event that two dices are the same.

```
def simualtor_dice3(n):
    # your code
```

4. Write a simulator function of rolling two dices n times. Calculate the experimental probability of the event that one dice gives 1 and the other gives 6.

```
def simualtor_dice4(n):
    # your code
```

5. Write a simulator function of rolling two dices n times. Calculate the experimental probability of the event that sum of two dices larger than 6.

```
def simualtor_dice5(n):
    # your code
```

6. Write a simulator function of taking 5 cards from 52-cards deck. Calculate the experimental probability of the event that 5 cards are all hearts (♥). Parameter n is the number of trials.

```
def simualtor_poker1(n):
    # your code
```

7. Write a simulator function of taking 4 cards from 52-cards deck. Calculate the experimental probability of the event that 4 cards are all in different types (1 heart (♥), 1 diamond (♦), 1 club (♣), and 1 spade (♠)). Parameter n is the number of trials.

```
def simualtor_poker2(n):
    # your code
```

8. A closed urn has 23 balls: 8 white, 6 blue and 9 red. Randomly select 6 balls (know that each ball has the same probability of selection). What is the probability of getting two white, two blue and two red balls.