Bernoulli Variable

In this lesson, we will learn about finding the probability of a Bernoulli variable.

WE'LL COVER THE FOLLOWING

- Probability of a Bernoulli variable

In the example below, we will flip a coin five times in a row and record how many times we obtain tails (varying from 0-5). We will be performing the experiment 1000 times.

We will then compute the cumulative probability, print the values to the screen and make a plot of the cumulative probability function using a bar graph.

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rnd

N = 1000
tails = np.sum(rnd.randint(0, 1+1, (5, 1000)), axis=0)
counttails = np.zeros(6, dtype='int')

for i in range(6):
    counttails[i] = np.count_nonzero(tails == i)

prob = counttails / N
cum_prob = np.cumsum(prob)
print('probababilties:', prob)
print('cumulative probabilities:', cum_prob)
plt.bar(range(0, 6), cum_prob)
plt.xticks(range(0, 6))
plt.xlabel('number of tails in two flips')
plt.ylabel('cumulative probability');
```

Execute the code several times and see that the graph changes a bit every time.

Probability of a Bernoulli variable

In the example above, we computed the probability of a certain number of heads in five flips experimentally. But we can, of course, compute the value exactly by using a few simple formulas.

Consider the random variable Y, which is the outcome of an experiment with two possible values 0 and 1. Let p be the probability of success,

$$p = P(Y = 1)$$

Then Y is said to be a Bernoulli variable. The experiment is repeated n times and we define X as the number of successes in the experiment. The variable X has a Binomial Distribution with parameters n and p. The probability that X takes value k can be computed as

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

The term $\binom{n}{k}$ may be computed with the `comb()` function, which needs to be imported from the `scipy.special` package.

Let's go back to the experiment where we flip a coin five times in a row and record how many times we obtain tails. We will compute the theoretical probability for 0, 1, 2, 3, 4, and 5 tails and compare the answer to the probability computed from 1000 trials, 10000 trials, and 100000 trials. Let' see if we can approach the theoretical value with more trials?

```
import numpy as np
import numpy.random as rnd
from scipy.special import comb

print('Theoretical probabilities:')
for k in range(6):
    y = comb(5, k) * 0.5**k * 0.5**(5 - k)
    print(k, ' tails ', y)

for N in (1000, 10000, 100000):
    tails = np.sum(rnd.randint(0, 1+1, (5, N)), axis=0)
    counttails = np.zeros(6)
    for i in range(6):
        counttails[i] = np.count_nonzero(tails==i)
    print('Probability with', N, 'trials: ', counttails / float(N))
```

In the next lesson, we will learn about the normal continuous random variables.