

L4作业

作业

一个人连续掷硬币，直到出现正面朝上的所用的次数记为 n ，他将赢得得 2^n 元。问他赢得的奖金期望值是？

请用Jupyter的markdown先写出解析答案。若最多可以投掷 k 次，用code实现随着 k 的变化，奖金期望值的变化，用图形表示（x轴 k ，y轴对应的期望值）。

1. 解析计算

- 没有过程
- 期望值的函数不是函数的期望值 若 $g(X)$ 不是 X 的线性函数， $E(g(X)) \neq g(E(X))$
- 计算的时候看清楚括号和求和的位置 别约错项了？

设：直到出现正面朝上所用的次数为 n ，正面出现的概率 $p = \frac{1}{2}$

n 是一个几何随机变量：

$$P(n) = (1 - p)^{n-1}p = \left(\frac{1}{2}\right)^n$$

$$E = \sum_{n=1}^{\infty} 2^n \times P(n) = \sum_{n=1}^{\infty} 1 = \infty$$

若最多可以掷 k 次：

$$E = \sum_{n=1}^k 2^n \times P(n) = \sum_{n=1}^k 1 = k$$

作业

一个人连续掷硬币，直到出现正面朝上的所用的次数记为 n ，他将赢得 2^n 元。问他赢得的奖金期望值是？

请用Jupyter的markdown先写出解析答案。若最多可以投掷 k 次，用code实现随着 k 的变化，奖金期望值的变化，用图形表示（x轴 k ，y轴对应的期望值）。

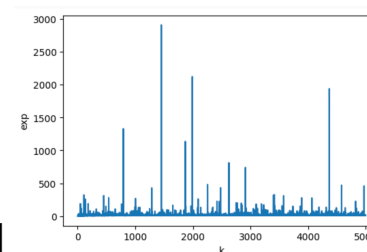
- 直接画了一条 $x=y$ 的直线，没有用随机数
- 几何分布随机数产生：第二个参数是长度，长度不是 k

```
x=np.random.geometric(0.5, events) #[1, 3, 1, 2,.....,1]
```

第 i 个人在第 $x[i]$ 次终于第一次投中了正面

- 概率除以什么
 - 1. 上文生成的样本数1000和下文计算的分子10000不一致 -> 在开头定义一个event变量
 - 2. 要考虑 k 次中都没投到的情况： $k=3$ 时：
$$E(k=3) = \sum_{i=1}^{i \leq 3} p(i) \times 2^i = \sum_{i=1}^{i \leq 3} \frac{N_i}{N_{total}} \times 2^i;$$
 - N_{total} 包括在第1, 2, 3次投出正面的+前3次始终没有投出正面的 ($i > k$); $N_{total} = events$
 - k 次没有投中，奖励是0，不是 $2^0=1$

2. 随机数模拟



作业

一个人连续掷硬币，直到出现正面朝上的所用的次数记为 n ，他将赢得 2^n 元。问他赢得的奖金期望值是？

请用Jupyter的markdown先写出解析答案。若最多可以投掷 k 次，用code实现随着 k 的变化，奖金期望值的变化，用图形表示（x轴 k ，y轴对应的期望值）。

2. 随机数模拟

```
1 import random
2 def simulate(k):
3     n = 0
4     for i in range(k):
5         re = random.choice([0,1])
6         if re ==1:
7             n = i+1
8             return 2**n
9     return 0
10 #输入要求k 输出一次随机试验后拿到的钱
11
12 result = []
13 for i in range(15):
14     sum = 0
15     for k in range(100000):
16         temp = simulate(i)
17         sum = sum + temp
18     result.append(sum/100000)
```

✓ 1.4s

```
1 events = 50000
2 x=np.random.geometric(0.5,events)
3 #第i次试验中，第x[i]次是第一次投中正面
4 def expected_value(k):
5     rewards = 0
6     count = 0
7     for n in range(1, k+1):
8         count = (x==n).sum()
9         rewards += 2**n * count
10    exp = rewards/events
11    return exp
12
13 k_values = list(range(1, 30))
14 expected_values = [expected_value(k) for k in k_values]
```

另外

- 在Jupyter cell里写markdown；markdown中输入数学公式可以用latex语法

- 写必要的注释

```

$$\sum_{i=1}^n 2^i \times \frac{1}{2^i}$$

```

$$\sum_{i=1}^n 2^i \times \frac{1}{2^i}$$

- 变量命名与含义有关

```
p=np.random.geometric(0.2, 1000)
expect = collections.Counter(p<= k)[True]/sample
```

- 在别的地方借鉴代码的时候，弄明白每行代码的意义再抄上去🙏

- 不确定：可以print()出来检查；上网查 xxx怎么用；问GPT
- 交作业的时候，请算出最后答案，如果画了很多张图、尝试了很多方法，请标明哪个是习题要求的结果
- 请交.ipynb作业，不要交html
- jupyter notebook的kernel可以帮你保留变量的数值、导入的包，即使你删掉了那一行。但是把作业交上来的时候，只有代码文本的。请在交作业之前重启kernel，运行一遍，确保你的代码能正常工作