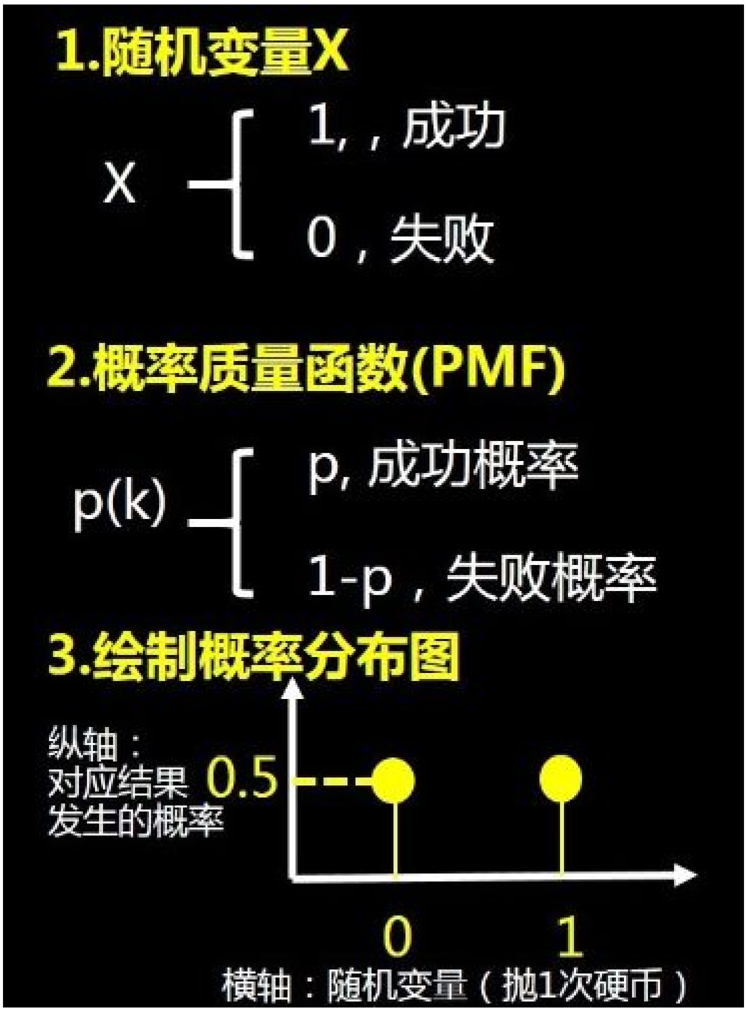
3.9常见分布函数

3.9.1 0-1分布函数（伯努利分布）

***运用场景：伯努利实验就是在相同的条件下来进行多次的随机实验，每两个实验之间是互不影响的，而且每次实验的结果只有两个。比如抛硬币就是伯努利实验。**

***表达式：P(X=1)=pP(X=0)=1-P**

***以 求抛一次硬币，硬币正面朝上的概率 为例：**



***上图就是抛硬币的伯努利分布的实现过程。下面是python实现伯努利实验：首先在进行实现之前我们需要导入scipy来进行伯努利实验，然后按照如下代码进行：**

In [3]:
```python
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt

'''
第一步，定义随机变量：1次抛硬币
成功指正面朝上记录为1，失败指反面朝上记录为0

'''
X=np.arange(0,2,1)
X
```

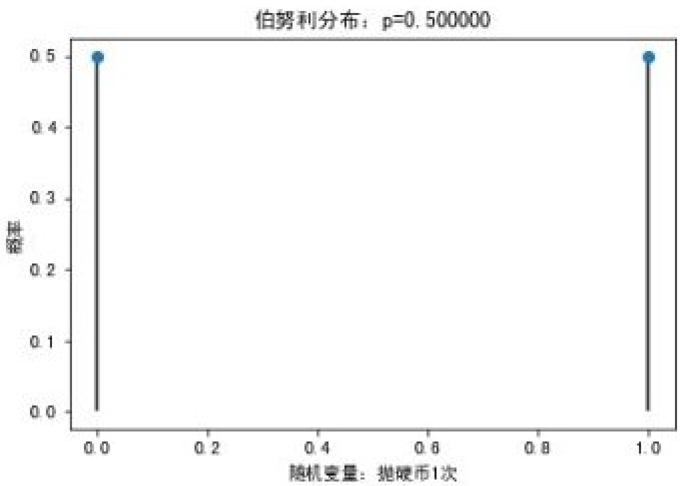
Out[3]: array([0, 1])

In [11]:
```python
#第二步 对应分布的概率：概率质量函数（PMF）
#它返回一个列表，列表中的每个元素表示随机变量中对应的概率。

p=0.5 # 硬币朝上的概率
pList = stats.bernoulli.pmf(X,p)
pList
```

Out[11]: array([0.5, 0.5])

In [11]:
```python
#第二步 对应分布的概率：概率质量函数（PMF）
#它返回一个列表，列表中的每个元素表示随机变量中对应的概率。

p=0.5 # 硬币朝上的概率
pList = stats.bernoulli.pmf(X,p)
pList
```

Out[11]: array([0.5, 0.5])

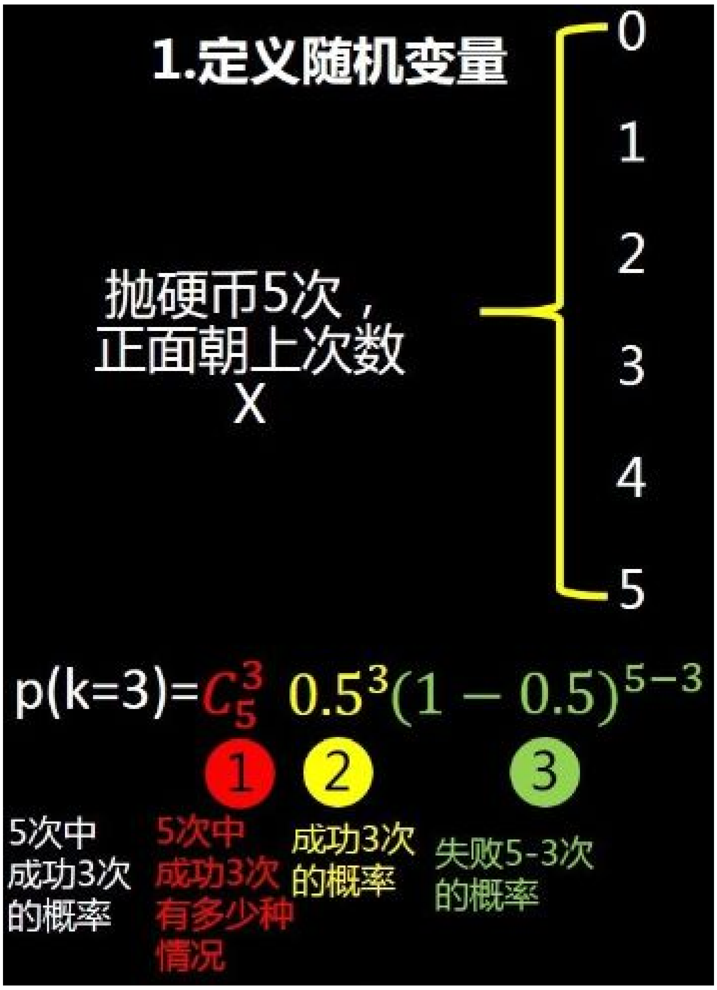


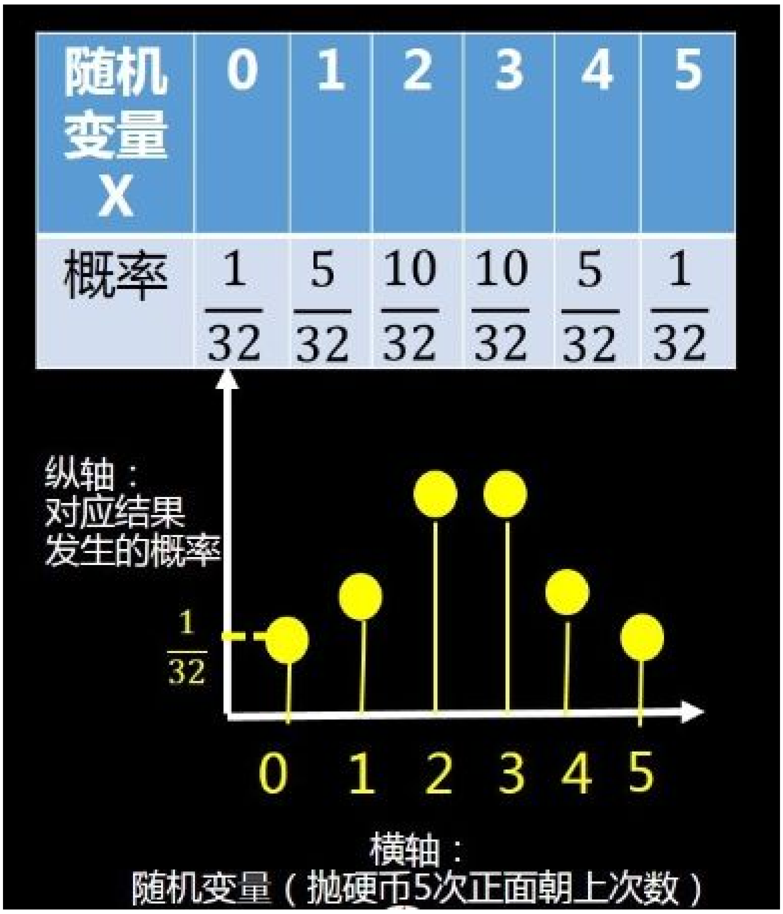
3.9.2 二项分布

***运用场景：二项分布的二项就代表该事件有两种结果，成功或者失败。当我们想了解成功k次的概率是多少的时候就需要用到二项分布的概率计算公式。再简单来理解二项分布就是做了n次的伯努利实验。**

***表达式：** $P(X=k)=C_n^k p^k (1-p)^{n-k}$

***以 求抛五次硬币，至少有三次以上正面朝上的概率 为例：**





这个图就是二项分布的概率分布图。

下面我们用python来实现以下二项分布：套路还是和上面的伯努利分布一样，定义随机变量，生成随机事件，然后求取不同事件对应的概率值，最后画图。

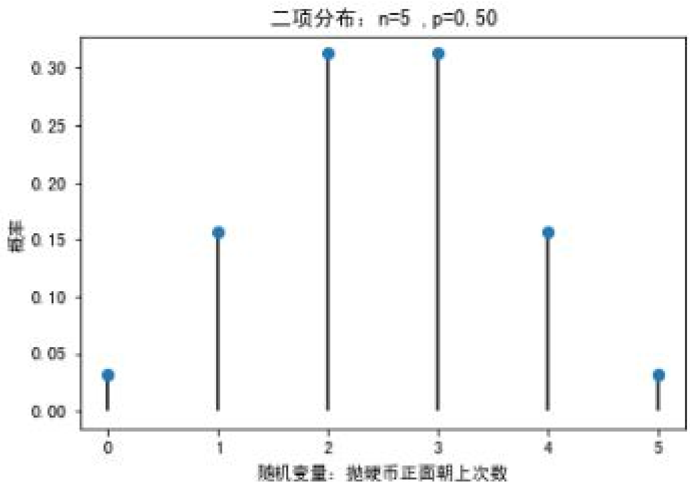
In [6]:
```
#二项分布
#定义随机变量，5次抛硬币， 正面朝上次数
n = 5
p = 0.5
X = np.arange(0,n+1,1)
X
```

Out [6]:
```
array([0, 1, 2, 3, 4, 5])
```

In [7]:
```
pList = stats.binom.pmf(X, n, p)
pList
```

Out [7]:
```
array([0.03125, 0.15625, 0.3125 , 0.3125 , 0.15625, 0.03125])
```

In [8]:
```
plt.plot(X, pList, marker= 'o',linestyle = 'None')
plt.vlines(X, 0, pList)
plt.xlabel('随机变量：抛硬币正面朝上次数')
plt.ylabel('概率')
plt.title('二项分布：n=%i ,p=%.2f' %(n,p))
plt.show()
```



3.9.3 几何分布

运用场景：

几何分布和二项分布二者非常的相近，几何分布的特征：

*做某件事的次数是固定的，次数用n表示，n次某事件是相互独立的。

*每一次事件都有两个可能的结果：成功或者失败。

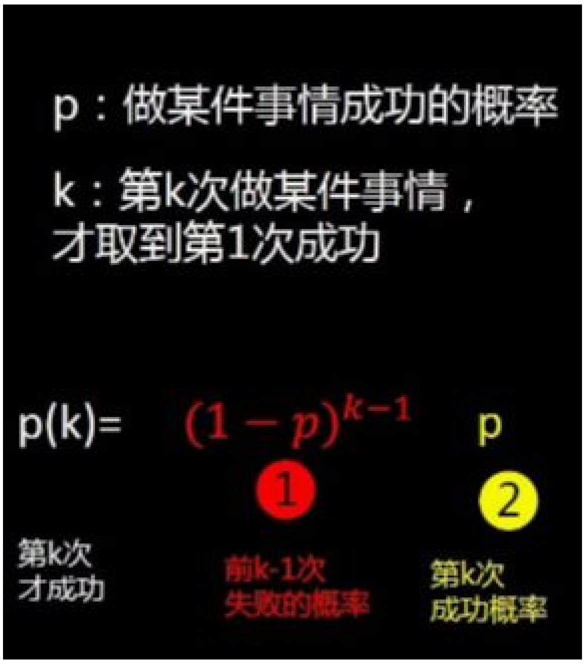
*每一次成功的概率都相等，成功的概率用p来表示。

*想知道第k次做某件事情，才能取得一次成功的概率。

$$P\left(X=k\right)=\left(1-p\right)^{k-1}p$$

性质：

$$E\left(X\right)=\frac{1}{p}Var\left(X\right)=\frac{1-p}{p^2}$$



python：

In [18]:
```
#几何分布，和前面概率分布的套路一样一样的
'''
需要说明一下arange函数，就是生成等差数列的，
arange(a,b,c) a是起始值，b是终止值，但是最后生成的值不包含b，c是步长，
也就是等差数列相邻两个数之间的差值
'''
k = 5
p = 0.6
X = np.arange(1,k+1,1)
X
```

Out[18]:
```
array([1, 2, 3, 4, 5])
```

In [19]:
```
pList = stats.geom.pmf(X,p)
pList
```

Out[19]:
```
array([0.6    , 0.24    , 0.096   , 0.0384 , 0.01536])
```

In [20]:
```
#绘图
plt.plot(X, pList, marker = 'o', linestyle = 'None')
plt.vlines(X, 0, pList)
plt.xlabel('随机变量：表白第k次才首次成功')
plt.ylabel('概率')
plt.title('几何分布:p =%.2f' %p)
plt.show()
```