In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st
```

# 第四章 随机变量的数字特征

## 1 数学期望

数学期望简称**期望**，又称为**均值**

随机变量X的期望记为 $E(X)$

### 定义

#### 离散型随机变量的期望

$$E(X)=\sum_{k=1}^{\infty}x_k p_k$$

**例子：**

| X | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *pk* | 0.002 | 0.001 | 0.002 | 0.005 | 0.02 | 0.04 | 0.18 | 0.37 | 0.25 | 0.12 | 0.01 |

**求解：**

In [2]:

```
x = np.arange(11)
p = np.array([0.002, 0.001, 0.002, 0.005, 0.02, 0.04, 0.18, 0.37, 0.25, 0.12, 0.01])
print('E(X) =', np.sum(x*p))
```

```
E(X) = 7.15
```

#### 连续型随机变量的期望

$$E(X)=\int_{-\infty}^{\infty}xf(x)dx$$

**例子：**

两个相互独立的电子装置，它们的寿命 $X_k(k=1,2)$ 服从同一分布率，其概率密度函数为

$$\begin{aligned} f(x) &= \frac{1}{\theta}e^{-x/\theta}, & x>0 \\ f(x) &= 0, & x\le 0 \end{aligned}$$

如果将这两个电子装置串联组成整机，求整机寿命 $N$ 的数学期望。

**求解：**

积分得到分布函数:

$$F(x) = 1 - e^{-x/\theta}, \qquad x > 0$$
$$F(x) = 0, \qquad x \le 0$$

因此$N = min\{X_1, X_2\}$ 的分布函数为

$$F_{min}(x) = 1 - [1 - F(x)]^2 = \begin{cases} 1 - e^{-x/\theta}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

$N$ 的概率密度函数为

$$f_{min}(x) = \begin{cases} \dfrac{2}{\theta} e^{-2x/\theta}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

因此 $N$ 的数学期望为

$$E(N) = \int_{-\infty}^{\infty} x f_{min}(x) dx = \int_0^{\infty} \frac{2x}{\theta} e^{-2x/\theta} dx = \frac{\theta}{2}$$

**用代码验证:**

In [5]:

```
from math import exp
from functools import partial

class DistributionGen(st.rv_continuous):
    """生成概率分布数据"""
    def __init__(self, theta, *args, **kwargs):
        self.theta = theta
        super().__init__(*args, **kwargs)

    def _pdf(self, x):
        if x <= 0:
            return 0
        else:
            return 2*exp(-2*x/self.theta)/self.theta

theta = np.random.uniform(1, 100)
gen = DistributionGen(theta=theta, name='f1')
print('θ\t:', theta)
print('θ/2\t:', theta/2)

# 利用概率分布生成抽样数据，计算抽样结果的平均值
samples = gen.rvs(size=100)
print('样本均值:', samples.mean())
```

```
θ	: 40.42577267086629
θ/2	: 20.212886335433144
样本均值: 18.717479377408466
```

**性质**

$$E(C) = C$$
$$E(CX) = CE(X)$$
$$E(X+Y) = E(X) + E(Y)$$
$$E(XY) = E(X)E(Y) \qquad X, Y \text{相互独立}$$

# 2 方差

## 定义

**方差**

$$D(X) = Var(X) = E\{[X - E(X)]^2\}$$

**标准差**或**均方差** $\sqrt{(D(X))}$，记为 $\sigma(X)$

对于离散型随机变量

$$\begin{aligned} D(X) &= \sum_{k=1}^{\infty} [x_k - E(X)]^2 p_k \\ &= E(X^2) - [E(X)]^2 \end{aligned}$$

对于连续型随机变量

$$D(X) = \int_{-\infty}^{\infty} [x - E()x]^2 f(x) dx$$

**性质**

$$\begin{aligned} D(C) &= 0 \\ D(CX) &= C^2 D(X) \\ D(X+Y) &= D(X) + D(Y) + 2E\{(X - E(X)(Y - E(Y)))\} \end{aligned}$$

**例子:**

求二项分布的 $X \sim B(n, p)$ 的 $D(X)$ 。

随机变量 $X \sim B(n, p)$ 可以分解为n个相互独立的服从以 $p$ 为参数的(0,1)分布的随机变量之和，$E(X_k) = 0 \times (1-p) + 1 \times p = p$, $D(X_k) = [0 - E(X_k)]^2 (1-p) + [1 - E(X_k)]^2 p = p(1-p)$

$$\begin{aligned} D(X) &= D(\sum_{k=1}^{n} X_k) = \sum_{k=1}^{n} D(X_k) \\ &= np(1-p) \end{aligned}$$

**代码验证:**

In [6]:

```python
# 利用概率分布生成抽样数据，计算抽样结果的平均值
n = 100
p = 0.7
samples = st.binom(n=n, p=p).rvs(size=1000)
print('n\t:', n)
print('p\t:', p)
print('np\t:', n*p)
print('np(1-p)\t:', n*p*(1-p))
print('样本均值:', samples.mean())
print('样本方差:', samples.var())
```

```
n       : 100
p       : 0.7
np      : 70.0
np(1-p) : 21.000000000000004
样本均值: 69.922
样本方差: 21.271915999999994
```

# 3 协方差

随机变量 $X$ 与 $Y$ 的**协方差**记为 $Cov(X,Y)$,

$$Cov(X,Y)=E([X-E(X)][Y-E(Y)])$$

而

$$\rho_{XY}=\frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

称为随机变量 $X$ 与 $Y$ 的**相关系数**

将Cov(X, Y)的定义式展开，得

$$Cov(X\ ,\ Y)=E(XY)-E(X)E(Y)$$

**性质**

$Cov(aX,bY)=abCov(X,Y)$

$Cov(X_1+X_2,Y)=Cov(X_1,Y)+Cov(X_2,Y)$

$\rho_{XY}\leq 1$

$\rho_{XY}=1$ 的充要条件是，存在常数a,b使

$$PY=a+bX=1$$

当 $\rho_{XY}=0$ 时，称 $X$ 与 $Y$ **不相关**

**例子:**

| Y X | -2 | -1 | 1 | 2 | P{Y=i} |
|---|---|---|---|---|---|
| 1 | 0 | 1/4 | 1/4 | 0 | 1/2 |
| 4 | 1/4 | 0 | 0 | 1/4 | 1/2 |
| P{X=i} | 1/4 | 1/4 | 1/4 | 1/4 | 1 |

In [7]:

```
x = np.array([-2, -1, 1, 2])
px = np.array([1/4, 1/4, 1/4, 1/4])

y = np.array([1,4])
py = np.array([1/2, 1/2])

e_x = np.sum(x*px)
e_y = np.sum(y*py)
print('E(X)\t=', e_x)
print('E(Y)\t=', e_y)

xy = x*np.array([y]).T
print('XY:')
print(xy)

pxy = px*np.array([py]).T
print('P{XY}:')
print(pxy)

e_xy = (xy*pxy).sum().sum()
print('E(X,Y)\t=', e_xy)

print('Cov(X, Y)\t=', e_xy-e_x*e_y)
```

```
E(X)	= 0.0
E(Y)	= 2.5
XY:
[[-2 -1  1  2]
 [-8 -4  4  8]]
P{XY}:
[[0.125 0.125 0.125 0.125]
 [0.125 0.125 0.125 0.125]]
E(X,Y)	= 0.0
Cov(X, Y)	= 0.0
```

In [ ]:

```

```