实验二 随机变量

姓名：林鹏　　班级：数科2194　　学号：202141084097

In [1]:

```
import numpy as np
from numpy.linalg import inv
```

In [2]:

```
A = np.array([2, -1, 4, 0, 1, -1, 1, 3, -2]).reshape((3, 3))    # 创建一个3阶方阵
A
```

Out[2]:

```
array([[ 2, -1,  4],
       [ 0,  1, -1],
       [ 1,  3, -2]])
```

In [3]:

```
Ex = np.array([5, -2, 7])
Ex
```

Out[3]:

```
array([ 5, -2,  7])
```

In [4]:

```
Vx = np.array([4, 1, 2, 1, 9, -3, 2, -3, 25]).reshape(3, 3, order="A")
Vx
```

Out[4]:

```
array([[ 4,  1,  2],
       [ 1,  9, -3],
       [ 2, -3, 25]])
```

In [5]:

```
D = np.sqrt(np.diag(np.diag(Vx)))    # np.sqrt()是平方根函数
D
```

Out[5]:

```
array([[2., 0., 0.],
       [0., 3., 0.],
       [0., 0., 5.]])
```

In [6]:

```
Ey = A.dot(Ex)
Ey = Ey.reshape(3,-1)
Ey
```

Out[6]:

```
array([[ 40],
       [ -9],
       [-15]])
```

In [7]:

```
Vy = A.dot(Vx).dot(A.T)
Vy
```

Out[7]:

```
array([[ 477, -126, -256],
       [-126,   40,   91],
       [-256,   91,  219]])
```

In [8]:

```
Rx = inv(D).dot(Vx).dot(inv(D))
Rx
```

Out[8]:

```
array([[ 1.        ,  0.16666667,  0.2       ],
       [ 0.16666667,  1.        , -0.2       ],
       [ 0.2       , -0.2       ,  1.        ]])
```

验证 2.9

In [9]:

```
import numpy as np
import numpy.linalg as LA
```

In [10]:

```
Vx = np.array([[9, 1, -2], [1, 20, 3], [-2, 3, 12]]).reshape((3,3), order='A')
Vx
```

Out[10]:

```
array([[ 9,  1, -2],
       [ 1, 20,  3],
       [-2,  3, 12]])
```

In [11]:

```
A = np.array([[2,3, 1], [1,-2, 5], [0, 1, -1]]).reshape((3,3), order='A')
A
```

Out[11]:

```
array([[ 2,  3,  1],
       [ 1, -2,  5],
       [ 0,  1, -1]])
```

$$V(y) = \boldsymbol{A}V(x)\boldsymbol{A}'$$

In [12]:

```
Vy = A.dot(Vx).dot(A.T)
Vy
```

Out[12]:

```
array([[250, -26,  48],
       [-26, 305, -76],
       [ 48, -76,  26]])
```

验证2.12

相关矩阵$R = (\rho)_{ij}$和协方差矩阵$\sum = \sigma_{ij}$之间有关系式
$R = D^{-1}\sum D^{-1}$
其中$D = diag(\sqrt{\sigma_{11}}, \sqrt{\sigma_{22}}, \ldots, \sqrt{\sigma_{pp}})$

In [13]:

```
sigma = np.array([[16, -4, 3], [-4, 4, -2], [3, -2, 9]]).reshape((3, 3), order='A')
sigma
```

Out[13]:

```
array([[16, -4,  3],
       [-4,  4, -2],
       [ 3, -2,  9]])
```

In [14]:

```
D = np.diag(np.sqrt(np.diag(sigma)))
D
```

Out[14]:

```
array([[4., 0., 0.],
       [0., 2., 0.],
       [0., 0., 3.]])
```

In [15]:

```
R = LA.inv(D).dot(sigma).dot(LA.inv(D))
R
```

Out[15]:

```
array([[ 1.        , -0.5       ,  0.25      ],
       [-0.5       ,  1.        , -0.33333333],
       [ 0.25      , -0.33333333,  1.        ]])
```