实验一　矩阵代数

姓名：林鹏　　班级：数科2194　　学号：202141084097

一、基于1.2.1的R中的一些基本函数及运算

In [1]:

```
import numpy as np
```

In [2]:

```
x = np.array([1, 3, 4, 1, 2, 5])    # 创建一个向量
x
```

Out[2]:

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

In [3]:

```
len(x)    #  向量的长度
```

Out[3]:

```
6
```

In [4]:

```
A = x.reshape((2,3), order='F')    # 利用 x 数值按列填充创建一个 2 X 3 矩阵
A
```

Out[4]:

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

In [5]:

```
A.dtype    # 数据的模式
```

Out[5]:

```
dtype('int32')
```

In [6]:

```
type(A)    # 数据的类或类型
```

Out[6]:

```
numpy.ndarray
```

In [7]:

```
A.shape    # 矩阵的维度
```

Out[7]:

```
(2, 3)
```

In [8]:

```
A.T    # 矩阵转置
```

Out[8]:

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

In [9]:

```
A.sum()    # 矩阵求和
```

Out[9]:

```
16
```

In [10]:

```
A.sum(axis=1)    # 矩阵按行求和
```

Out[10]:

```
array([7, 9])
```

In [11]:

```
A.sum(axis=0)    # 矩阵按列求和
```

Out[11]:

```
array([4, 5, 7])
```

In [12]:

```
A.mean()    # 矩阵求平均值
```

Out[12]:

```
2.6666666666666665
```

In [13]:

```
A.mean(axis=1)    # 矩阵按行求平均值
```

Out[13]:

```
array([2.33333333, 3.        ])
```

In [14]:

```
A.mean(axis=0)    # 矩阵按列求平均值
```

Out[14]:

```
array([2. , 2.5, 3.5])
```

In [15]:

```
B = np.array([6, 0, 2, 3, 1, 4]).reshape((2,3))
B
```

Out[15]:

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

In [16]:

```
A + B    # 矩阵相加
```

Out[16]:

```
array([[7, 4, 4],
       [6, 2, 9]])
```

In [17]:

```
A - B    # 矩阵相减
```

Out[17]:

```
array([[-5,  4,  0],
       [ 0,  0,  1]])
```

In [18]:

```
C = np.array([1, 2, 1, 3]).reshape((2,2), order='F')
C
```

Out[18]:

```
array([[1, 1],
       [2, 3]])
```

In [19]:

```
C @ A     # 矩阵相乘
```

Out[19]:

```
array([[ 4,  5,  7],
       [11, 11, 19]])
```

In [20]:

```
np.multiply(A, B)
```

Out[20]:

```
array([[ 6,  0,  4],
       [ 9,  1, 20]])
```

二、计算方阵的一些函数值

In [21]:

```
from numpy.linalg import inv, det, eig, svd
```

In [22]:

```
A = np.array([np.arange(1,6),
              np.array([2, 4, 7, 8, 9]),
              np.array([3, 7, 10, 15, 20]),
              np.array([4, 8, 15, 30, 20]),
              np.array([5, 9, 20, 20, 40])])
# 将5个5为向量按行合并创建一个5阶（对称）方阵
A
```

Out[22]:

```
array([[ 1,  2,  3,  4,  5],
       [ 2,  4,  7,  8,  9],
       [ 3,  7, 10, 15, 20],
       [ 4,  8, 15, 30, 20],
       [ 5,  9, 20, 20, 40]])
```

In [23]:

```
np.diag(A)     # 由矩阵的对角线元素构成的向量
```

Out[23]:

```
array([ 1,  4, 10, 30, 40])
```

In [24]:

```
np.diag(np.diag(A))     # 由向量diag(A)的元素创建对角矩阵
```

Out[24]:

```
array([[ 1,  0,  0,  0,  0],
       [ 0,  4,  0,  0,  0],
       [ 0,  0, 10,  0,  0],
       [ 0,  0,  0, 30,  0],
       [ 0,  0,  0,  0, 40]])
```

In [25]:

```
np.eye(5)     # 创建一个5阶单位矩阵
```

Out[25]:

```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [26]:

```
inv(A)     # 矩阵的逆
```

Out[26]:

```
array([[ 9.78873239e+00, -2.18309859e+00, -1.85915493e+00,
         1.12676056e-01,  1.40845070e-01],
       [-2.18309859e+00,  7.74647887e-01,  7.88732394e-01,
        -1.69014085e-01, -2.11267606e-01],
       [-1.85915493e+00,  7.88732394e-01,  3.94366197e-02,
        -8.45070423e-03,  3.94366197e-02],
       [ 1.12676056e-01, -1.69014085e-01, -8.45070423e-03,
         7.32394366e-02, -8.45070423e-03],
       [ 1.40845070e-01, -2.11267606e-01,  3.94366197e-02,
        -8.45070423e-03,  3.94366197e-02]])
```

In [27]:

```
det(A)     # 矩阵的行列式
```

Out[27]:

```
-355.00000000000006
```

In [28]:

```
eig(A)     #  矩阵的特征值与特征向量
```

Out[28]:

```
(array([70.33488803, 14.44024095,  1.997606  ,  0.09374538, -1.86648037]),
 array([[ 0.10513926,  0.00733125,  0.26673691,  0.95627367, -0.05730686],
        [ 0.20596656,  0.05549834,  0.82858975, -0.22629386,  0.46554035],
        [ 0.39707684, -0.02585507,  0.32382661, -0.18402887, -0.83840992],
        [ 0.5462168 ,  0.78569385, -0.25851683,  0.01391472,  0.13155919],
        [ 0.70035756, -0.61553463, -0.26569873,  0.0164776 ,  0.24443626]]))
```

In [29]:

```
values, vectors = eig(A)[0], eig(A)[1]   # 分别获取特征值与特征向量
```

In [30]:

```
value = values[0]   # 取第一个特征值
```

In [31]:

```
vector = vectors[:,0] # 取第一个特征向量
```

In [32]:

```
left= A.dot(vector)
```

In [33]:

```
right = value * vector
```

In [34]:

```
np.allclose(left, right)    # 比较是否相等（可忽略小数点后五位）
```

Out[34]:

```
True
```

In [35]:

```
np.sum(np.diag(A))    # 矩阵的迹，即对向量diag(A)中的元素求和
```

Out[35]:

```
85
```

三、例1.6.6中的奇异值分解

In [36]:

```
A = np.array([1, 1, 2, -2, 2, 2]).reshape((2,3), order="F")
A
```

Out[36]:

```
array([[ 1,  2,  2],
       [ 1, -2,  2]])
```

In [37]:

```
svd(A)    #  奇异值分解
```

Out[37]:

```
(array([[-0.70710678, -0.70710678],
        [-0.70710678,  0.70710678]]),
 array([3.16227766, 2.82842712]),
 array([[-4.47213595e-01, -4.15130595e-16, -8.94427191e-01],
        [ 7.32343785e-17, -1.00000000e+00,  3.39211021e-16],
        [-8.94427191e-01,  6.18327757e-17,  4.47213595e-01]]))
```