Linear algebra

In [4]:

```
import numpy as np
import pandas as pd
from numpy import linalg as la
```

1. Create 5 matrices with five different dimensions (1-D,2-D,...5-D)

1-D

In [33]:

```
a=np.array([1,5,6])
print(a)
```

```
[1 5 6]
```

2-D

In [26]:

```
b=np.array([[1,3],[7,6]])
print(b)
```

```
[[1 3]
 [7 6]]
```

3-D

In [11]:

```
c=np.array([[1,2,3],[1,3,6],[5,6,2]])
print(c)
```

```
[[1 2 3]
 [1 3 6]
 [5 6 2]]
```

4-D

In [13]:

```
d=np.array([[1,2,3,5],[5,6,1,5],[5,2,6,8],[8,5,6,5]])
print(d)
```

```
[[1 2 3 5]
 [5 6 1 5]
 [5 2 6 8]
 [8 5 6 5]]
```

5-D

In [14]:

```
e=np.array([[1,2,3,5,6],[5,6,1,5,8],[5,2,6,8,5],[8,5,6,5,1],[1,2,3,4,5]])
print(e)
```

```
[[1 2 3 5 6]
 [5 6 1 5 8]
 [5 2 6 8 5]
 [8 5 6 5 1]
 [1 2 3 4 5]]
```

2. Find determinants of 5 matrices and display your output

In [20]:

```
print(la.det(c))
```

```
-1.0000000000000004
```

In [21]:

```
print(la.det(d))
```

```
366.00000000000017
```

In [22]:

```
print(la.det(e))
```

```
166.00000000000023
```

3. Find inverse of the above 5 matrices and display your output

In [28]:

```
print(la.inv(b))
```

```
[[-0.4         0.2       ]
 [ 0.46666667 -0.06666667]]
```

In [29]:

```
print(la.inv(c))
```

```
[[ 30. -14.  -3.]
 [-28.  13.   3.]
 [  9.  -4.  -1.]]
```

```
In [30]:
```

```
print(la.inv(d))
```

```
[[-0.45901639  0.09836066  0.24590164 -0.03278689]
 [ 0.3852459   0.02459016 -0.35519126  0.15846995]
 [ 0.35245902 -0.25409836 -0.21857923  0.25136612]
 [-0.07377049  0.12295082  0.22404372 -0.20765027]]
```

```
In [31]:
```

```
print(la.inv(e))
```

```
[[-2.28313253  0.30722892  0.62650602 -0.38554217  1.69879518]
 [ 2.4939759  -0.21686747 -0.79518072  0.56626506 -1.96385542]
 [-1.44578313 -0.04819277  0.15662651 -0.09638554  1.6746988 ]
 [ 3.36746988 -0.27108434 -0.4939759   0.45783133 -3.20481928]
 [-2.36746988  0.27108434  0.4939759  -0.45783133  2.20481928]]
```

4. Find the rank, diagonal and trace of the 5 matrices

```
In [41]:
```

```
print(la.matrix_rank(a))
print(np.diag(a))
np.trace(a)
```

```
1
[[1 0 0]
 [0 5 0]
 [0 0 6]]

---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-41-081f86137889> in <module>
      1 print(la.matrix_rank(a))
      2 print(np.diag(a))
----> 3 np.trace(a)

<__array_function__ internals> in trace(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py in tr
ace(a, offset, axis1, axis2, dtype, out)
   1707         return asarray(a).trace(offset=offset, axis1=axis1, axis2=
axis2, dtype=dtype, out=out)
   1708     else:
-> 1709         return asanyarray(a).trace(offset=offset, axis1=axis1, axi
s2=axis2, dtype=dtype, out=out)
   1710 
   1711 

ValueError: diag requires an array of at least two dimensions
```

In [42]:

```
print(la.matrix_rank(b))
print(np.diag(b))
np.trace(b)
```

```
2
[1 6]
```

Out[42]:

```
7
```

In [43]:

```
print(la.matrix_rank(c))
print(np.diag(c))
np.trace(c)
```

```
3
[1 3 2]
```

Out[43]:

```
6
```

In [44]:

```
print(la.matrix_rank(d))
print(np.diag(d))
np.trace(d)
```

```
4
[1 6 6 5]
```

Out[44]:

```
18
```

In [45]:

```
print(la.matrix_rank(e))
print(np.diag(e))
np.trace(e)
```

```
5
[1 6 6 5 5]
```

Out[45]:

```
23
```

5. Find Eigen value and eigen vector for 5 matrices

In [50]:

```
print(la.eigvals(b))
x,y=la.eig(b)
print("root(values):",x)
print("matrix(vector):",y)
```

```
[-1.72015325  8.72015325]
root(values): [-1.72015325  8.72015325]
matrix(vector): [[-0.7408145  -0.36220694]
 [ 0.67170966 -0.9320977 ]]
```

In [51]:

```
print(la.eig(b))
```

```
(array([-1.72015325,  8.72015325]), array([[-0.7408145 , -0.36220694],
       [ 0.67170966, -0.9320977 ]]))
```

In [52]:

```
print(la.eigvals(c))
x,y=la.eig(c)
print("root(values):",x)
print("matrix(vector):",y)
```

```
[10.13451661  0.0237294  -4.15824602]
root(values): [10.13451661  0.0237294  -4.15824602]
matrix(vector): [[-0.36342675 -0.71561666 -0.20721858]
 [-0.62893455  0.66585403 -0.61127293]
 [-0.68728621 -0.21102419  0.7638101 ]]
```

In [53]:

```
print(la.eig(c))
```

```
(array([10.13451661,  0.0237294 , -4.15824602]), array([[-0.36342675, -0.7
1561666, -0.20721858],
       [-0.62893455,  0.66585403, -0.61127293],
       [-0.68728621, -0.21102419,  0.7638101 ]]))
```

In [54]:

```
print(la.eigvals(d))
x,y=la.eig(d)
print("root(values):",x)
print("matrix(vector):",y)
```

```
[18.48069387 -3.8501023  -1.1405577   4.50996613]
root(values): [18.48069387 -3.8501023  -1.1405577   4.50996613]
matrix(vector): [[-0.3235841  -0.55737676 -0.50218891  0.08403538]
 [-0.42112466 -0.07128615  0.51319713 -0.81858042]
 [-0.58810893 -0.32157655  0.60106963  0.56629328]
 [-0.60997971  0.76212725 -0.35092777  0.04664847]]
```

In [55]:

```
print(la.eig(d))
```

```
(array([18.48069387, -3.8501023 , -1.1405577 ,  4.50996613]), array([[-0.3
235841 , -0.55737676, -0.50218891,  0.08403538],
       [-0.42112466, -0.07128615,  0.51319713, -0.81858042],
       [-0.58810893, -0.32157655,  0.60106963,  0.56629328],
       [-0.60997971,  0.76212725, -0.35092777,  0.04664847]]))
```

In [56]:

```
print(la.eigvals(e))
x,y=la.eig(e)
print("root(values):",x)
print("matrix(vector):",y)
```

```
[21.48154465  4.11345471  0.74201541 -1.16635746 -2.17065731]
root(values): [21.48154465  4.11345471  0.74201541 -1.16635746 -2.1706573
1]
matrix(vector): [[ 0.34238196 -0.10088896  0.29483976  0.56945723 -0.48914
754]
 [ 0.4719533   0.82636488 -0.49367222 -0.40102237  0.12559151]
 [ 0.54078902 -0.54528626  0.35281054  0.13468898 -0.31773097]
 [ 0.52466994  0.04075889 -0.57894799 -0.60367279  0.75996121]
 [ 0.30381469 -0.08912977  0.45793144  0.3637823  -0.25781282]]
```

In [57]:

```
print(la.eig(e))
```

```
(array([21.48154465,  4.11345471,  0.74201541, -1.16635746, -2.17065731]),
array([[ 0.34238196, -0.10088896,  0.29483976,  0.56945723, -0.48914754],
       [ 0.4719533 ,  0.82636488, -0.49367222, -0.40102237,  0.12559151],
       [ 0.54078902, -0.54528626,  0.35281054,  0.13468898, -0.31773097],
       [ 0.52466994,  0.04075889, -0.57894799, -0.60367279,  0.75996121],
       [ 0.30381469, -0.08912977,  0.45793144,  0.3637823 , -0.2578128
2]]))
```

In [ ]: