

EXPERIMENT 1

NAME: K SRI KALYAN

ROLLNO: R177219180

sapid:500075205

Introduction to numpy

numpy: numpy is mainly used to create multi-dimensional arrays i.e one dimensional a two dimensional and three dimensional and soon we use numpy for calculating matrix and numpy is mainly used for mathematical functions because numpy already contains many pre defined mathematical functions.

Double-click (or enter) to edit

```
import numpy as np
```

```
a=np.arange(15).reshape(3,5)
```

Double-click (or enter) to edit

```
a.shape
```

```
(3, 5)
```

```
a.ndim
```

```
2
```

```
a.dtype.name
```

```
'int64'
```

```
a.itemsize
```

```
8
```

```
a.size
```

```
15
```

```
type(a)
```

```
numpy.ndarray
```

```
b=np.array([6,7,8])
```

```
b
```

```
array([6, 7, 8])
```

```
type(b)
```

```
numpy.ndarray
```

```
b.dtype
```

```
dtype('int64')
```

```
c=np.array([[1,2],[3,4]],dtype=complex)
```

```
c
```

```
array([[1.+0.j, 2.+0.j],  
       [3.+0.j, 4.+0.j]])
```

```
c.dtype.name
```

```
'complex128'
```

```
np.zeros((4,5))
```

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

```
np.ones((3,5))
```

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

```
np.ones((2,3,4), dtype=np.int16)
```

```
array([[[1, 1, 1, 1],  
        [1, 1, 1, 1],  
        [1, 1, 1, 1]],  
       [[1, 1, 1, 1],  
        [1, 1, 1, 1],  
        [1, 1, 1, 1]]], dtype=int16)
```

```
np.empty((2,3))
```

```
array([[4.63832849e-310, 0.00000000e+000, 0.00000000e+000],  
       [0.00000000e+000, 0.00000000e+000, 0.00000000e+000]])
```

```
np.eye(6)
```

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

```
np.arange(10,30,5)
```

```
array([10, 15, 20, 25])
```

```
np.arange(0,2,0.3)
```

```
array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8])
```

```
np.linspace(0,2,9)
```

```
array([0. , 0.25, 0.5 , 0.75, 1. , 1.25, 1.5 , 1.75, 2.  ])
```

```
np.arange(0,11,1)**2
```

```
array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81, 100])
```

```
print(a)
```

```
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]]
```

```
print(a.reshape(5,3))
```

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]
 [12 13 14]]
```

```
print(np.arange(10000))
```

```
[  0    1    2 ... 9997 9998 9999]
```

```
A=np.array([[1,1],[0,1]])
```

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

```
A+B
```

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

A-B

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

A*B


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

A@B

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

A/B

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: RuntimeWarning: divide by zero encountered in divide
"""Entry point for launching an IPython kernel.
array([[0.5 ,  inf],
       [0.  ,  0.25]])
```



```
np.sin(np.arange(0,2 * np.pi,np.pi/6))
```

```
array([ 0.00000000e+00,  5.00000000e-01,  8.66025404e-01,  1.00000000e+00,
        8.66025404e-01,  5.00000000e-01,  1.22464680e-16, -5.00000000e-01,
       -8.66025404e-01, -1.00000000e+00, -8.66025404e-01, -5.00000000e-01])
```

A.dot(B)

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

a[:6:2]

```
array([[ 0,  1,  2,  3,  4],
       [10, 11, 12, 13, 14]])
```

a[::-1]

```
array([[10, 11, 12, 13, 14],
       [ 5,  6,  7,  8,  9],
       [ 0,  1,  2,  3,  4]])
```

a.T

```
array([[ 0,  5, 10],
       [ 1,  6, 11],
       [ 2,  7, 12],
```

```
[ 3,  8, 13],
 [ 4,  9, 14]])
```

```
A.trace()
```

```
2
```

```
A[0] [[False,True]]
```

```
array([1])
```

1.2 Introduction to Pandas

pandas: pandas stands for "python data analysis library" it is mainly used for data analysis and with the help of pandas we can create multi dimensional structured dataset ie to represent the data in 2d ,and we can read data from text file or even a database ,csv,pst files we can read and then convert to form of dataframe so it will be in format of 2d in pandas we have data frame functions for processing the data.

```
import pandas as pd
```

```
s=pd.Series([1,3,5,np.nan,6,8])
```

```
s
```

```
0    1.0
1    3.0
2    5.0
3     NaN
4    6.0
5    8.0
dtype: float64
```

```
dates=pd.date_range('20130101',periods=6)
```

```
dates
```

```
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
```

```
df=pd.DataFrame(np.random.randn(6,4),index=dates,columns=list('ABCD'))
df
```

	A	B	C	D
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846
2013-01-02	0.761210	-0.746418	0.650311	-0.960459
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646
2013-01-04	0.871407	-1.966686	0.553038	-0.423192

```
df2=pd.DataFrame({
    'A':1.0,
    'B' :pd.Timestamp('20130102'),
    'C' :pd.Series(1,index=list(range(4)),dtype='float32'),
    'D' : np.array([3]*4,dtype='int32'),
    'E' : pd.Categorical(['test', 'train','test','train']),
    'F':'foo'
})
df2
```

	A	B	C	D	E	F
0	1.0	2013-01-02	1.0	3	test	foo
1	1.0	2013-01-02	1.0	3	train	foo
2	1.0	2013-01-02	1.0	3	test	foo
3	1.0	2013-01-02	1.0	3	train	foo

```
df2.dtypes

A          float64
B    datetime64[ns]
C          float32
D           int32
E          category
F           object
dtype: object
```

```
df.head()
```

	A	B	C	D
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846
2013-01-02	0.761210	-0.746418	0.650311	-0.960459
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646
2013-01-04	0.871407	-1.966686	0.553038	-0.423192
2013-01-05	1.305279	0.979598	0.105930	-0.309670

```
df.tail(3)
```

	A	B	C	D
2013-01-04	0.871407	-1.966686	0.553038	-0.423192
2013-01-05	1.305279	0.979598	0.105930	-0.309670
2013-01-06	0.564857	-0.238659	-0.680846	-1.977930

```
df.index
```

```
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
```

```
df.columns
```

```
Index(['A', 'B', 'C', 'D'], dtype='object')
```

```
df.to_numpy()
```

```
array([[ -1.1477344 , -0.63731901,  1.19156088, -1.08484556],
       [  0.76120954, -0.74641805,  0.65031098, -0.96045866],
       [ -1.02142493, -0.78015332, -1.34172847,  1.31364564],
       [  0.871407   , -1.96668607,  0.55303753, -0.42319206],
       [  1.30527911,  0.97959808,  0.10593032, -0.30966993],
       [  0.56485675, -0.23865863, -0.68084589, -1.97792971]])
```

```
df2.to_numpy()
```

```
array([[1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'test', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'train', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'test', 'foo'],
       [1.0, Timestamp('2013-01-02 00:00:00'), 1.0, 3, 'train', 'foo']],
      dtype=object)
```

```
df.describe()
```

	A	B	C	D
count	6.000000	6.000000	6.000000	6.000000
mean	0.222266	-0.564939	0.079711	-0.573742
std	1.041710	0.953041	0.936532	1.098831
min	-1.147734	-1.966686	-1.341728	-1.977930
25%	-0.624855	-0.771720	-0.484152	-1.053749
50%	0.663033	-0.691869	0.329484	-0.691825
75%	0.843858	-0.338324	0.625993	-0.338050
max	1.305279	0.979598	1.191561	1.313646

```
df.T
```

	2013-01-01	2013-01-02	2013-01-03	2013-01-04	2013-01-05	2013-01-06
A	-1.147734	0.761210	-1.021425	0.871407	1.305279	0.564857
B	-0.637319	-0.746418	-0.780153	-1.966686	0.979598	-0.238659
C	1.191561	0.650311	-1.341728	0.553038	0.105930	-0.680846
D	-1.084846	-0.960459	1.313646	-0.423192	-0.309670	-1.977930

```
df.sort_index(axis=1,ascending=False)
```

	D	C	B	A
2013-01-01	-1.084846	1.191561	-0.637319	-1.147734
2013-01-02	-0.960459	0.650311	-0.746418	0.761210
2013-01-03	1.313646	-1.341728	-0.780153	-1.021425
2013-01-04	-0.423192	0.553038	-1.966686	0.871407
2013-01-05	-0.309670	0.105930	0.979598	1.305279
2013-01-06	-1.977930	-0.680846	-0.238659	0.564857

```
df.sort_values(by='B')
```

	A	B	C	D
2013-01-04	0.871407	-1.966686	0.553038	-0.423192
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646
2013-01-02	0.761210	-0.746418	0.650311	-0.960459
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846
2013-01-06	0.564857	-0.238659	-0.680846	-1.977930
2013-01-05	1.305279	0.979598	0.105930	-0.309670

```
df['A']
```

2013-01-01	-1.147734
2013-01-02	0.761210
2013-01-03	-1.021425
2013-01-04	0.871407
2013-01-05	1.305279
2013-01-06	0.564857
Freq: D, Name: A, dtype: float64	

```
df[0:3]
```



```
df.loc[dates[0]]
```

	A	B	C	D
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846

```
df.loc[dates[1]]
```

	A	B	C	D
2013-01-02	0.761210	-0.746418	0.650311	-0.960459
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646
2013-01-04	0.871407	-1.966686	0.553038	-0.423192

```
df.loc[dates[0]]
```

A -1.147734
B -0.637319
C 1.191561
D -1.084846
Name: 2013-01-01 00:00:00, dtype: float64

```
df.loc[:, ['A', 'B']]
```

	A	B
2013-01-01	-1.147734	-0.637319
2013-01-02	0.761210	-0.746418
2013-01-03	-1.021425	-0.780153
2013-01-04	0.871407	-1.966686
2013-01-05	1.305279	0.979598
2013-01-06	0.564857	-0.238659

```
df.at[dates[0], 'A']
```

-1.1477344036622141

```
df.iloc[3]
```

A 0.871407
B -1.966686
C 0.553038
D -0.423192
Name: 2013-01-04 00:00:00, dtype: float64

```
df[df['A']>0]
```

	A	B	C	D
2013-01-02	0.761210	-0.746418	0.650311	-0.960459
2013-01-04	0.871407	-1.966686	0.553038	-0.423192
2013-01-05	1.305279	0.979598	0.105930	-0.309670

```
s1=pd.Series([1,2,3,4,5,6], index=pd.date_range('20130102',periods=6))
s1
```

2013-01-02	1
2013-01-03	2
2013-01-04	3
2013-01-05	4
2013-01-06	5
2013-01-07	6
Freq: D, dtype: int64	

```
df['F']=s1
df
```

	A	B	C	D	F
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846	NaN
2013-01-02	0.761210	-0.746418	0.650311	-0.960459	1.0
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646	2.0
2013-01-04	0.871407	-1.966686	0.553038	-0.423192	3.0
2013-01-05	1.305279	0.979598	0.105930	-0.309670	4.0
2013-01-06	0.564857	-0.238659	-0.680846	-1.977930	5.0

```
df.fillna(value=5)
```

	A	B	C	D	F
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846	5.0
2013-01-02	0.761210	-0.746418	0.650311	-0.960459	1.0
2013-01-03	-1.021425	-0.780153	-1.341728	1.313646	2.0
2013-01-04	0.871407	-1.966686	0.553038	-0.423192	3.0
2013-01-05	1.305279	0.979598	0.105930	-0.309670	4.0
2013-01-06	0.564857	-0.238659	-0.680846	-1.977930	5.0

```
pd.isna(df)
```

	A	B	C	D	F
2013-01-01	False	False	False	False	True
2013-01-02	False	False	False	False	False
2013-01-03	False	False	False	False	False
2013-01-04	False	False	False	False	False
2013-01-05	False	False	False	False	False

```
df.mean()
```

```
A    0.222266
B   -0.564939
C    0.079711
D   -0.573742
F    3.000000
dtype: float64
```

Double-click (or enter) to edit

```
df.mean(axis=1)
```

```
2013-01-01    -0.419585
2013-01-02     0.140929
2013-01-03     0.034068
2013-01-04     0.406913
2013-01-05     1.216228
2013-01-06     0.533485
Freq: D, dtype: float64
```

```
df.apply(np.cumsum)
```

	A	B	C	D	F
2013-01-01	-1.147734	-0.637319	1.191561	-1.084846	NaN
2013-01-02	-0.386525	-1.383737	1.841872	-2.045304	1.0
2013-01-03	-1.407950	-2.163890	0.500143	-0.731659	3.0
2013-01-04	-0.536543	-4.130576	1.053181	-1.154851	6.0
2013-01-05	0.768736	-3.150978	1.159111	-1.464521	10.0
2013-01-06	1.333593	-3.389637	0.478265	-3.442450	15.0

```
df.apply(lambda x:x.max()-x.min())
```

```
A    2.453014
B    2.946284
C    2.533289
D    3.291575
F    4.000000
dtype: float64
```

```
s=pd.Series(['A','B','C','Aaba','Baca',np.nan,'CABA','dog','cat'])
s.str.lower()
```

```
0      a
1      b
2      c
3    aaba
4    baca
5     NaN
6    caba
7     dog
8     cat
dtype: object
```

```
df=pd.DataFrame(np.random.randn(10,4))
df
```

	0	1	2	3
0	-1.382005	0.101814	0.484584	1.473365
1	-0.162133	-0.737314	0.050972	2.410700
2	-1.189844	-1.444478	-0.369956	-0.459719
3	1.626640	0.018198	0.804166	-0.684620
4	-2.072829	0.542255	-0.571787	0.159659
5	-0.874753	0.660118	0.775963	-0.064916
6	0.699538	-0.759778	-0.587998	-0.578574
7	1.232714	-1.127591	0.241859	2.019024
8	0.514235	1.214297	-0.551785	0.042895
9	1.400519	1.140763	-0.752840	-1.168360

```
pieces=[df[:3],df[3:7],df[7:]]
pieces
```

	0	1	2	3
0	-1.382005	0.101814	0.484584	1.473365
1	-0.162133	-0.737314	0.050972	2.410700
2	-1.189844	-1.444478	-0.369956	-0.459719
	0	1	2	3
3	1.626640	0.018198	0.804166	-0.684620
4	-2.072829	0.542255	-0.571787	0.159659
5	-0.874753	0.660118	0.775963	-0.064916
6	0.699538	-0.759778	-0.587998	-0.578574
	0	1	2	3
7	1.232714	-1.127591	0.241859	2.019024
8	0.514235	1.214297	-0.551785	0.042895
9	1.400519	1.140763	-0.752840	-1.168360

```
pd.concat(pieces)
```

	0	1	2	3
0	-1.382005	0.101814	0.484584	1.473365
1	-0.162133	-0.737314	0.050972	2.410700
2	-1.189844	-1.444478	-0.369956	-0.459719
3	1.626640	0.018198	0.804166	-0.684620
4	-2.072829	0.542255	-0.571787	0.159659
5	-0.874753	0.660118	0.775963	-0.064916
6	0.699538	-0.759778	-0.587998	-0.578574
7	1.232714	-1.127591	0.241859	2.019024
8	0.514235	1.214297	-0.551785	0.042895
9	1.400519	1.140763	-0.752840	-1.168360

```
left=pd.DataFrame({'key':['foo','foo'],'ival':[1,2]})
```

```
right=pd.DataFrame({'key':['foo','foo'],'ival':[4,5]})
```

```
pd.merge(left,right,on='key')
```

	key	ival_x	ival_y
0	foo	1	4
1	foo	1	5
2	foo	2	4
3	foo	2	5

```
df.groupby(1).sum()
```

023

1

```
df.sort_values(by=1)
```

	0	1	2	3
2	-1.189844	-1.444478	-0.369956	-0.459719
7	1.232714	-1.127591	0.241859	2.019024
6	0.699538	-0.759778	-0.587998	-0.578574
1	-0.162133	-0.737314	0.050972	2.410700
3	1.626640	0.018198	0.804166	-0.684620
0	-1.382005	0.101814	0.484584	1.473365
4	-2.072829	0.542255	-0.571787	0.159659
5	-0.874753	0.660118	0.775963	-0.064916
9	1.400519	1.140763	-0.752840	-1.168360
8	0.514235	1.214297	-0.551785	0.042895

