

ppml220901038

October 30, 2024

EXP No:1 NUMPY Aim : To install Numpy package and do the basic functions 1.Declare the Numpy array

```
[ ]: import numpy as np
arr=np.array([[1,5,4],[3,5,6]])
print(arr)
```

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

```
[ ]: b=np.zeros((3,6))
print(b)
```

```
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
```

```
[ ]: c=np.full((7,3),4)
print(c)
```

```
[[4 4 4]
 [4 4 4]
 [4 4 4]
 [4 4 4]
 [4 4 4]
 [4 4 4]
 [4 4 4]]
```

```
[ ]: d=np.random.random((5,3))
print(d)
```

```
[[0.38959985 0.90689109 0.11368553]
 [0.25052046 0.04900494 0.38685153]
 [0.37016075 0.97406041 0.87717347]
 [0.69968581 0.55895829 0.96506934]
 [0.54635156 0.93545539 0.64735546]]
```

```
[ ]: e=np.arange(8,40,5)
print(e)
```

```
[ 8 13 18 23 28 33 38]
```

```
[ ]: narr=arr.reshape(3,2)
      print(narr)
      farr=narr.flatten()
      print(farr)
      print(narr.ndim)
      print(narr.shape)
      print(narr.size)
      print(narr.dtype)
```

```
[[1 5]
 [4 3]
 [5 6]]
[1 5 4 3 5 6]
2
(3, 2)
6
int64
```

```
[ ]: print(narr.astype('f'))
```

```
[[1. 5.]
 [4. 3.]
 [5. 6.]]
```

```
[ ]: b=np.array([[1,4,3],[4,5,6],[7,8,9],[11,22,33]])
      print(b)
      print(b[0:3:2])
      print(b[:,1])
      print(b[:, -1])
      print(b[0,1:3])
      print(b[2:,2:])
      print(b[:,2:])
      print(b[2:,2])
```

```
[[ 1  4  3]
 [ 4  5  6]
 [ 7  8  9]
 [11 22 33]]
[[1 4 3]
 [7 8 9]]
[[ 1  4  3]
 [ 4  5  6]
 [ 7  8  9]
 [11 22 33]]
[[11 22 33]
 [ 7  8  9]]
```

```

[ 4  5  6]
[ 1  4  3]]
[4 3]
[[ 9]
 [33]]
[[ 3]
 [ 6]
 [ 9]
 [33]]
[ 9 33]

```

```

[ ]: a=np.array([5,2,3])
      b=np.array([4,5,6])
      print(np.concatenate([a,b]))
      print(np.hstack((a,b)))
      print(np.vstack((a,b)))
      print(np.dstack((a,b)))

```

```

[5 2 3 4 5 6]
[5 2 3 4 5 6]
[[5 2 3]
 [4 5 6]]
[[[5 4]
   [2 5]
   [3 6]]]

```

```

[ ]: a=np.array([1,2,4,6,5,4])
      print(np.where(a==4))
      print(np.where(a%2==0))
      print(np.where(a%2!=0))

```

```

(array([2, 5]),)
(array([1, 2, 3, 5]),)
(array([0, 4]),)

```

```

[ ]: a=np.array([[1,2,3,4],[2,3,6,7],[9,6,7,3],[3,6,0,1]])
      temp=a[[0,1,2,3]]
      print("integer array indexing:")
      print(temp)
      b=np.array([1,2,3,4])
      c=b>2
      temp=b[c]
      print("boolean array indexing:",temp)

```

```

integer array indexing:
[[1 2 3 4]
 [2 3 6 7]
 [9 6 7 3]

```

```
[3 6 0 1]
boolean array indexing: [3 4]
```

```
[ ]: a=np.array([1,9,3,4,5,6])
na=np.array_split(a,3)
print(a)
print(na)
print("Display in another way")
print(na[0])
print(na[1])
print(na[2])
```

```
[1 9 3 4 5 6]
[array([1, 9]), array([3, 4]), array([5, 6])]
Display in another way
[1 9]
[3 4]
[5 6]
```

```
[ ]: import numpy as np
arr = np.array([3, 2, 0, 1])
print(np.sort(arr))
arr = np.array(['mohanram', 'gowtham', 'vijayram'])
print(np.sort(arr))
arr = np.array([True, False, True])
print(np.sort(arr))
arr = np.array([[3, 2, 4], [5, 0, 1]])
print(np.sort(arr))
```

```
[0 1 2 3]
['gowtham' 'mohanram' 'vijayram']
[False True True]
[[2 3 4]
 [0 1 5]]
```

```
[ ]: arr=np.array([40,42,43,44])
x=[True,False,True,False]
narr=arr[x]
print(arr,narr,sep="\n")
na=arr>42
print(na)
print(arr[na])
```

```
[40 42 43 44]
[40 43]
[False False True True]
[43 44]
```

```
[ ]: c=np.array([11,44,33])
      d=np.array([4,5,6])
      print(c+d)
      print(c-d)
      print(c*d)
      print(c/d)
```

```
[15 49 39]
[ 7 39 27]
[ 44 220 198]
[2.75 8.8 5.5 ]
```

```
[ ]: a=np.array([2,4,6,8])
      d=np.array([4,5,6,9])
      print(a+2)
      print(a-2)
      print(a*2)
      print(a/2)
      print(a.dot(b))
```

```
[ 4  6  8 10]
[0 2 4 6]
[ 4  8 12 16]
[1. 2. 3. 4.]
60
```

```
[ ]: import numpy as n
      def a(x,y):
          if(x>y):
              return x-y
          else:
              return x+y
      e=n.vectorize(a)
      c=n.array([1,2,3])
      d=n.array([4,5,6])
      print(e(c,d))
```

```
[5 7 9]
```

EXP No:2 PANDAS Aim:To install pandas and do the DataFrame operations

```
[ ]: import numpy as np
      import pandas as pd
      df = pd.DataFrame()
      print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

```
[ ]: e=pd.Series(['gow','moh','karthi','ram'])
i=pd.Series([102,107,109,114])
c={'Emp':e,'ID':i}
r=pd.DataFrame(c)
print(r)
```

	Emp	ID
0	gow	102
1	moh	107
2	karthi	109
3	ram	114

```
[ ]: r['Age']=pd.Series([20,22,18,24])
print("Add columns")
print(r)
a=pd.DataFrame([['deepak',115,25],['arnav',118,23]],columns=['Emp','ID','Age'])
r=pd.concat([r,a]).reset_index(drop=True) #r=r.append(a)
print("Add rows")
print(r)
```

Add columns

	Emp	ID	Age
0	gow	102	20.0
1	moh	107	22.0
2	karthi	109	18.0
3	ram	114	24.0
4	deepak	115	NaN
5	arnav	118	NaN

Add rows

	Emp	ID	Age
0	gow	102	20.0
1	moh	107	22.0
2	karthi	109	18.0
3	ram	114	24.0
4	deepak	115	NaN
5	arnav	118	NaN
6	deepak	115	25.0
7	arnav	118	23.0

```
[ ]: del r['Age']
print("Delete Column")
print(r)
print("Delete Row")
print(r.drop(5))
```

Delete Column

	Emp	ID
0	gow	102

```

1    moh  107
2  karthi 109
3    ram  114
4  deepak 115
5  arnav  118
6  deepak 115
7  arnav  118
Delete Row
      Emp  ID
0    gow  102
1    moh  107
2  karthi 109
3    ram  114
4  deepak 115
6  deepak 115
7  arnav  118

```

```

[ ]: print("Extract rows")
      print(r.loc[2])
      print("Extract column")
      print(r['Emp'])

```

```

Extract rows
Emp    karthi
ID      109
Name: 2, dtype: object
Extract column
0    gow
1    moh
2  karthi
3    ram
4  deepak
5  arnav
6  deepak
7  arnav
Name: Emp, dtype: object

```

```

[ ]: i=pd.DataFrame([[4,8]]*3,columns=['A','B'])
      print(i)
      print("Sum of values")
      print(i.apply(np.sum,axis=0))
      print(i.apply(np.sum,axis=1))
      print("Square root of values")
      print(i.apply(np.sqrt,axis=0))
      print(i.apply(np.sqrt,axis=1))

```

```

      A  B
0    4  8

```

```

1  4  8
2  4  8
Sum of values
A    12
B    24
dtype: int64
0    12
1    12
2    12
dtype: int64
Square root of values
      A      B
0  2.0  2.828427
1  2.0  2.828427
2  2.0  2.828427
      A      B
0  2.0  2.828427
1  2.0  2.828427
2  2.0  2.828427

```

```

[ ]: i=pd.DataFrame([[1,5,3],[8,5,9],[71,28,11]],columns=['A','B','C'])
print(i)
print("Min of values")
print(i.agg(['min'],axis=0))
print(i.agg(['min'],axis=1))
print("Max of values")
print(i.agg(['max'],axis=0))
print(i.agg(['max'],axis=1))

```

```

      A  B  C
0   1  5  3
1   8  5  9
2  71 28 11
Min of values
      A  B  C
min  1  5  3
      min
0     1
1     5
2    11
Max of values
      A  B  C
max  71 28 11
      max
0     5
1     9
2    71

```



```
[ ]: a=pd.DataFrame(np.random.randn(4,2),index=['0','3','1','2'],columns=['A','B'])
print(a)
a1=a.sort_index()
print(a1)
a2=a.sort_index(ascending=False)
print(a2)
a3=a.sort_values(by='B')
print(a3)
```

```
      A      B
0  0.617879 -0.024229
3  0.307392 -0.774096
1  1.309192  0.680727
2 -0.150329 -0.418555
```

```
      A      B
0  0.617879 -0.024229
1  1.309192  0.680727
2 -0.150329 -0.418555
3  0.307392 -0.774096
```

```
      A      B
3  0.307392 -0.774096
2 -0.150329 -0.418555
1  1.309192  0.680727
0  0.617879 -0.024229
```

```
      A      B
3  0.307392 -0.774096
2 -0.150329 -0.418555
0  0.617879 -0.024229
1  1.309192  0.680727
```

```
[ ]: e=pd.DataFrame({'Id':[1,2,3,4,5], 'Name':['A','B','C','D','E'], 'Age':
    ↳ [20,21,22,20,22]})
f=pd.DataFrame({'Id':[1,2,3,4,5], 'Mark1':[60,87,78,98,90], 'Mark2':
    ↳ [77,88,93,97,88]})
print(pd.merge(e,f,on='Id'))
```

```
   Id  Name  Age  Mark1  Mark2
0    1    A   20     60     77
1    2    B   21     87     88
2    3    C   22     78     93
3    4    D   20     98     97
4    5    E   22     90     88
```

```
[ ]: arr=np.array([1,2,3,4,5])
s=pd.Series(arr)
print(s)
dic={'a':1, 'b':2, 'c':3, 'd':4, 'e':5}
```

```
s=pd.Series(dic)
print(s)
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

```
[ ]: a=pd.Series(5,index=[0,1,2,3,4])
print(a)
b=pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])
print(b)
```

```
0    5
1    5
2    5
3    5
4    5
dtype: int64
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

EXP No:3 LOAD AND STORE Aim: To Create and store Excel / CSV Data Series files and store the Same. Do some basic operations

```
[ ]: import numpy as np
import pandas as pd
e=pd.DataFrame({'Id':[1,2,3,4,5], 'Name':['A','B','C','D','E'], 'Age':
    ↳[20,21,22,20,22]})
f=pd.DataFrame({'Id':[1,2,3,4,5], 'Mark1':[60,87,78,98,90], 'Mark2':
    ↳[77,88,93,97,88]})
g=pd.merge(e,f,on='Id')
print(g)
g.to_excel('Dataframe.xlsx',index=False)
```

```
Id Name Age Mark1 Mark2
```

0	1	A	20	60	77
1	2	B	21	87	88
2	3	C	22	78	93
3	4	D	20	98	97
4	5	E	22	90	88

```
[ ]: a=pd.read_excel('Dataframe.xlsx')
      print(a)
      display(a)
```

	Id	Name	Age	Mark1	Mark2
0	1	A	20	60	77
1	2	B	21	87	88
2	3	C	22	78	93
3	4	D	20	98	97
4	5	E	22	90	88

	Id	Name	Age	Mark1	Mark2
0	1	A	20	60	77
1	2	B	21	87	88
2	3	C	22	78	93
3	4	D	20	98	97
4	5	E	22	90	88

```
[ ]: print(a.columns)
      print(a.shape)
```

```
Index(['Id', 'Name', 'Age', 'Mark1', 'Mark2'], dtype='object')
(5, 5)
```

```
[ ]: df = pd.read_excel('Dataframe.xlsx')
      print(df['Name'])
      print(df.iloc[1])
```

```
0    A
1    B
2    C
3    D
4    E
Name: Name, dtype: object
Id      2
Name    B
Age     21
Mark1   87
Mark2   88
Name: 1, dtype: object
```

```
[ ]: print(df[2:5])
      print(df.iloc[1:4,0:2])
```

	Id	Name	Age	Mark1	Mark2
2	3	C	22	78	93
3	4	D	20	98	97
4	5	E	22	90	88

	Id	Name
1	2	B
2	3	C
3	4	D

```
[ ]: a=pd.DataFrame({'Id':[1,2,3,4,5], 'Name':
      ↳['Arun', 'Ancy', 'Anika', 'Reena', 'Seema'], 'Age':[18,17,18,18,18]})
      a.to_excel('Dataframe1.xlsx', index=False)
```

```
[ ]: b=pd.DataFrame({'Id':[1,2,3,4,5], 'Mark1':[60,87,78,98,90], 'Mark2':
      ↳[77,88,93,97,88]})
      b.to_excel('Dataframe2.xlsx', index=False)
```

```
[ ]: df1=pd.read_excel('Dataframe1.xlsx')
      df2=pd.read_excel('Dataframe2.xlsx')
      c=pd.concat([df1,df2],axis=1) #c=df1.append(df2)
      df3=c.to_excel('Dataframe3.xlsx', index=False)
      print(c)
```

	Id	Name	Age	Id	Mark1	Mark2
0	1	Arun	18	1	60	77
1	2	Ancy	17	2	87	88
2	3	Anika	18	3	78	93
3	4	Reena	18	4	98	97
4	5	Seema	18	5	90	88

```
[ ]: x=pd.DataFrame({'Id':[1,2,3,4,5], 'Mark1':[60,87,78,98,90], 'Mark2':
      ↳[77,88,93,97,88]})
      x.to_excel('Dataframe2.xlsx', index=False)
      y=pd.read_excel('Dataframe2.xlsx')
      s=y.sort_values(by='Mark1') #s=y.sort(by='Mark1')
      s.to_excel('SortDF.xlsx')
      a=pd.read_excel('SortDF.xlsx')
      print(a)
```

	Unnamed: 0	Id	Mark1	Mark2
0	0	1	60	77
1	2	3	78	93
2	1	2	87	88
3	4	5	90	88

EXP No:4 TEXTFILE AND JSON Aim : To open , read and write the text files and basic JSON operation.

```
[ ]: a=open("ppml.txt",'w')
a.write("Python is a programming language")
a.close()
```

```
[ ]: e=open("ppml.txt",'r')
print(e.read())
e.close()
```

Python is a programming language

```
[ ]: e = open("ppml.txt", 'r')
print(e.readline())
e.close()
```

Python is a programming language

```
[ ]: b=open("ppml.txt",'a')
b.write("\nIt is easy to learn")
b.close()
```

```
[ ]: e=open("ppml.txt",'r')
print(e.read())
e.close()
```

Python is a programming language
It is easy to learn

```
[ ]: import json
x='{"Name":"Ancy","Age":20,"City":"Kanyakumari"}'
y=json.loads(x)
z=json.dumps(x)
print(y)
print(z)
```

```
{'Name': 'Ancy', 'Age': 20, 'City': 'Kanyakumari'}
"{\\Name\\":\\\"Ancy\\\",\\\"Age\\\":20,\\\"City\\\":\\\"Kanyakumari\\\"}"
```

EXP No:5 DATA CLEANING AND PREPARATION

```
[ ]: import pandas as pd
import numpy as np
a=pd.DataFrame(np.random.
↳randn(5,3),index=['a','c','e','f','h'],columns=['One','Two','Three'])
print(a)
```

```
a=a.reindex(['a','b','c','d','e','f','g','h'])
print(a)
```

	One	Two	Three
a	0.765749	0.499029	-0.605124
c	2.017507	-0.038392	-0.369111
e	-0.112745	0.396994	0.202176
f	-0.393493	0.657978	1.070524
h	0.143374	-0.053108	0.037991

	One	Two	Three
a	0.765749	0.499029	-0.605124
b	NaN	NaN	NaN
c	2.017507	-0.038392	-0.369111
d	NaN	NaN	NaN
e	-0.112745	0.396994	0.202176
f	-0.393493	0.657978	1.070524
g	NaN	NaN	NaN
h	0.143374	-0.053108	0.037991

```
[ ]: a1=a
print(a.dropna())
```

	One	Two	Three
a	0.765749	0.499029	-0.605124
c	2.017507	-0.038392	-0.369111
e	-0.112745	0.396994	0.202176
f	-0.393493	0.657978	1.070524
h	0.143374	-0.053108	0.037991

```
[ ]: a2=a1
print(a1.fillna(0))
```

	One	Two	Three
a	0.765749	0.499029	-0.605124
b	0.000000	0.000000	0.000000
c	2.017507	-0.038392	-0.369111
d	0.000000	0.000000	0.000000
e	-0.112745	0.396994	0.202176
f	-0.393493	0.657978	1.070524
g	0.000000	0.000000	0.000000
h	0.143374	-0.053108	0.037991

```
[ ]: a3=a2
print(a2.fillna(method='pad'))
```

	One	Two	Three
a	0.765749	0.499029	-0.605124
b	0.765749	0.499029	-0.605124

```
c  2.017507 -0.038392 -0.369111
d  2.017507 -0.038392 -0.369111
e -0.112745  0.396994  0.202176
f -0.393493  0.657978  1.070524
g -0.393493  0.657978  1.070524
h  0.143374 -0.053108  0.037991
```

<ipython-input-4-b927352de9fc>:2: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.

```
print(a2.fillna(method='pad'))
```

```
[ ]: a4=a3
print(a3.fillna(method='bfill'))
```

```
      One      Two      Three
a  0.765749  0.499029 -0.605124
b  2.017507 -0.038392 -0.369111
c  2.017507 -0.038392 -0.369111
d -0.112745  0.396994  0.202176
e -0.112745  0.396994  0.202176
f -0.393493  0.657978  1.070524
g  0.143374 -0.053108  0.037991
h  0.143374 -0.053108  0.037991
```

<ipython-input-5-1f554bed7946>:2: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.

```
print(a3.fillna(method='bfill'))
```

```
[ ]: a5=a4
print(a4.bfill())
```

```
      One      Two      Three
a  0.765749  0.499029 -0.605124
b  2.017507 -0.038392 -0.369111
c  2.017507 -0.038392 -0.369111
d -0.112745  0.396994  0.202176
e -0.112745  0.396994  0.202176
f -0.393493  0.657978  1.070524
g  0.143374 -0.053108  0.037991
h  0.143374 -0.053108  0.037991
```

```
[ ]: print(a['One'].isnull())
print(a['One'].notnull())
```

```
a  False
b   True
c  False
```

```

d      True
e     False
f     False
g      True
h     False
Name: One, dtype: bool
a      True
b     False
c      True
d     False
e      True
f      True
g     False
h      True
Name: One, dtype: bool

```

```

[ ]: b=pd.
      ↪DataFrame([[11, 'a'], [12, 'b'], [13, 'c'], [14, 'd'], [15, 'e'], [103, 'f'], [101, 'g'], [18, 'h']], column
print(b)

```

```

      Age Name
0    11    a
1    12    b
2    13    c
3    14    d
4    15    e
5   103    f
6   101    g
7    18    h

```

```

[ ]: print(b.replace({103:16,101:17}))

```

```

      Age Name
0    11    a
1    12    b
2    13    c
3    14    d
4    15    e
5    16    f
6    17    g
7    18    h

```

EXP NO:6 DATA WRANGLING

```

[ ]: import pandas as pd
      import numpy as np
      d1={"name":["salini", "Mary", "Johncy"], "age": [40,60,38]}
      d2={"Qualified": [True,False,True]}

```



```
df1=pd.DataFrame(d1)
df2=pd.DataFrame(d2)
nd=df1.join(df2)
print(nd)
```

	name	age	Qualified
0	salini	40	True
1	Mary	60	False
2	Johnny	38	True

```
[ ]: df=pd.DataFrame({"team":["A","B","C","D"],"points":[88,89,99,98],"assist":
↳[17,14,16,12],"rebounds":[22,21,25,38]})
print(df)
df1=pd.melt(df,id_vars=['team'],value_vars=['points','assist','rebounds'])
print(df1)
```

	team	points	assist	rebounds
0	A	88	17	22
1	B	89	14	21
2	C	99	16	25
3	D	98	12	38

	team	variable	value
0	A	points	88
1	B	points	89
2	C	points	99
3	D	points	98
4	A	assist	17
5	B	assist	14
6	C	assist	16
7	D	assist	12
8	A	rebounds	22
9	B	rebounds	21
10	C	rebounds	25
11	D	rebounds	38

```
[ ]: b=pd.
↳DataFrame([['Hen',80],['Hen',100],['Parrot',40],['Parrot',30],['Finges',10],['Finges',15]],
print(b)
```

	Name	Speed
0	Hen	80
1	Hen	100
2	Parrot	40
3	Parrot	30
4	Finges	10
5	Finges	15

```
[ ]: b.groupby(['Name']).mean()
```

```
[ ]:      Speed
      Name
Finges   12.5
Hen      90.0
Parrot   35.0
```

```
[ ]: b.groupby(['Name']).sum()
```

```
[ ]:      Speed
      Name
Finges    25
Hen      180
Parrot    70
```

```
[ ]: b.groupby(['Name']).count()
```

```
[ ]:      Speed
      Name
Finges     2
Hen         2
Parrot      2
```

```
[ ]: b.groupby(['Name']).first()
```

```
[ ]:      Speed
      Name
Finges    10
Hen       80
Parrot    40
```

```
[ ]: b.groupby(['Name']).last()
```

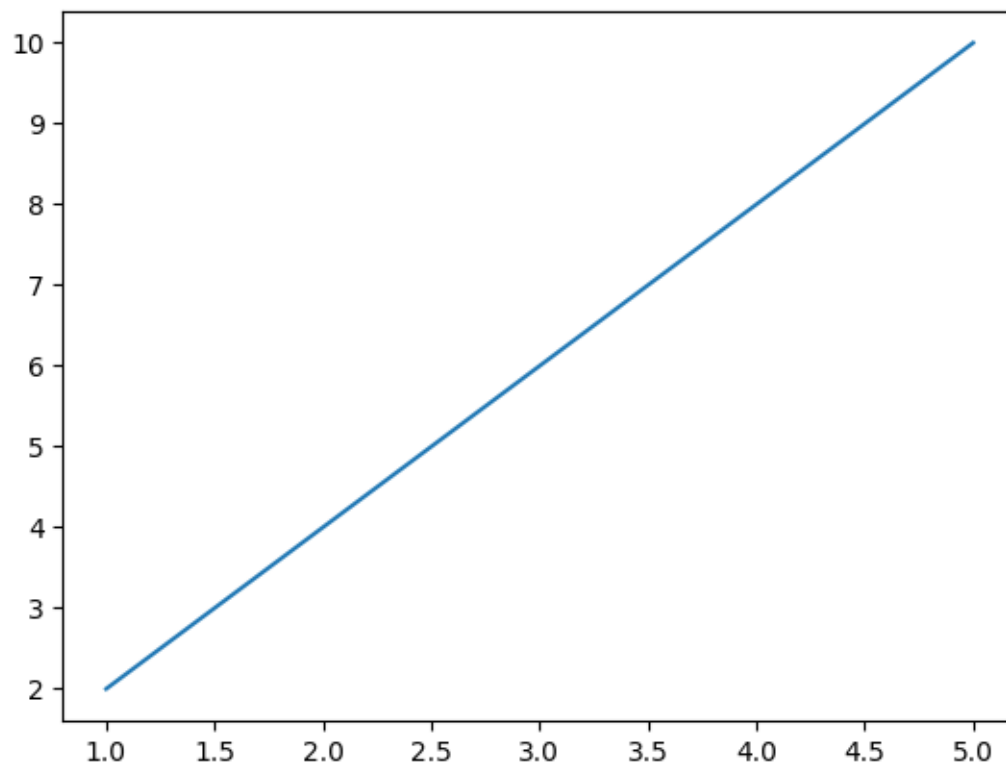
```
[ ]:      Speed
      Name
Finges    15
Hen      100
Parrot    30
```

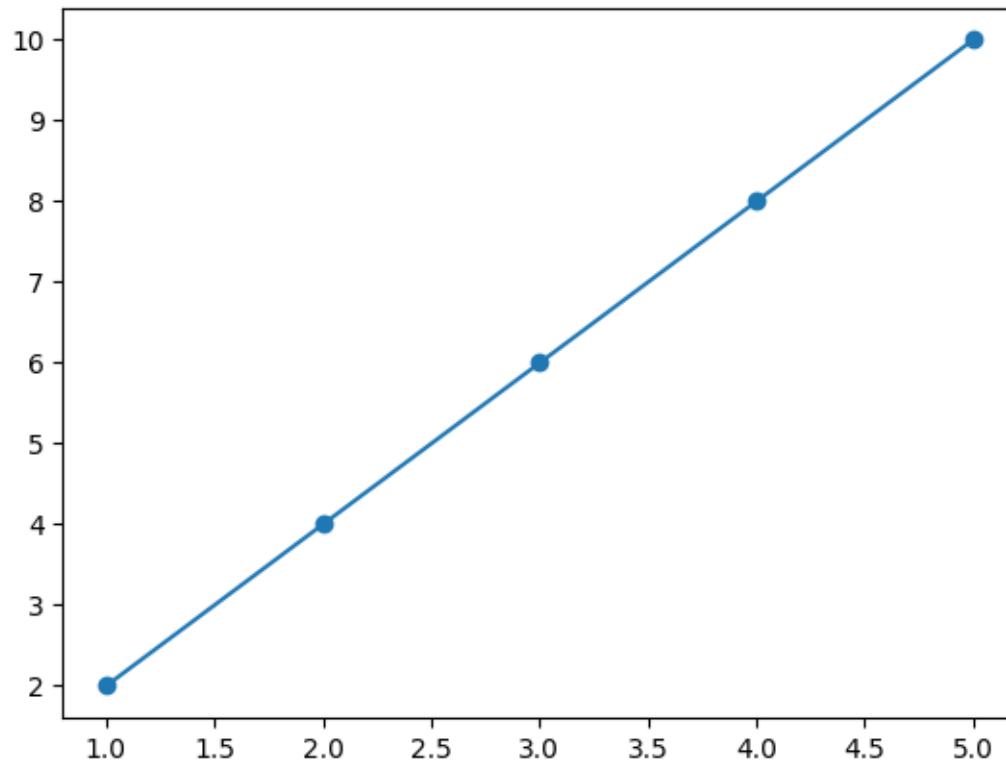
```
[ ]: b.groupby(['Name']).size()
```

```
[ ]: Name
      Finges    2
      Hen       2
      Parrot    2
      dtype: int64
```

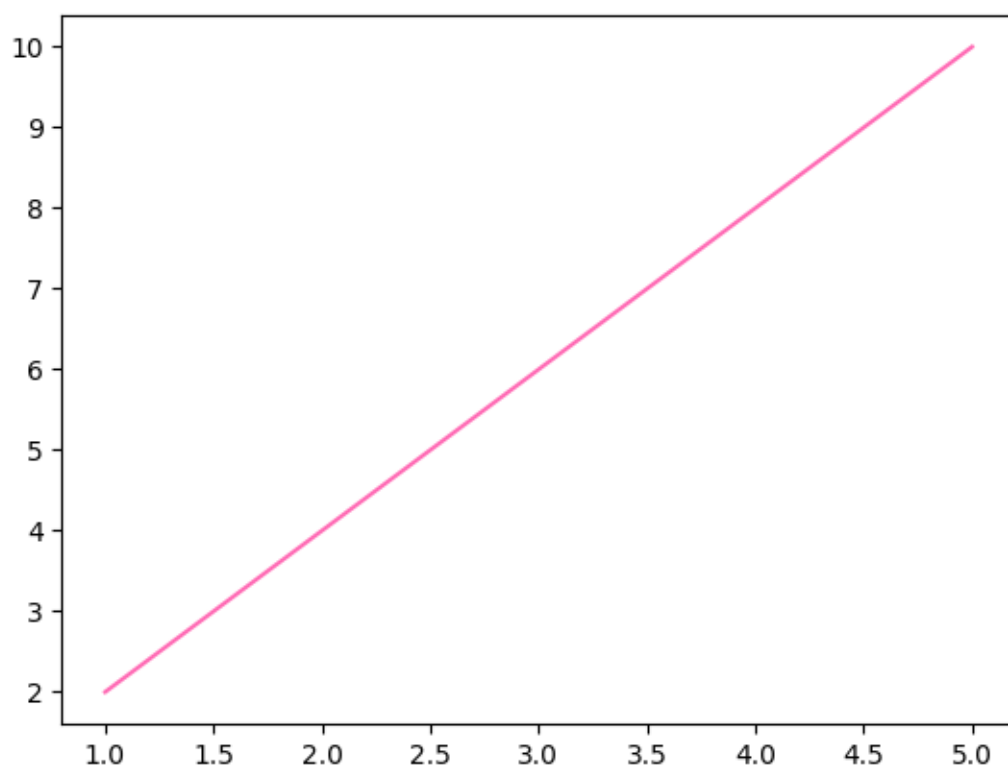
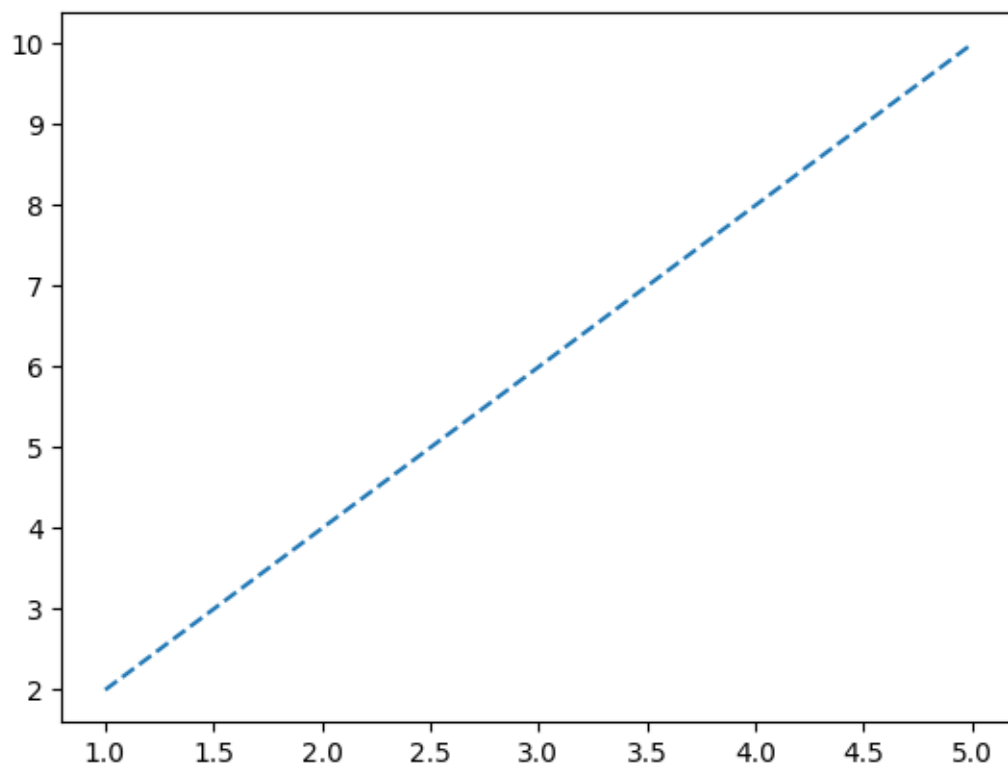
EXP NO: 7 DATA VISUALIZATION

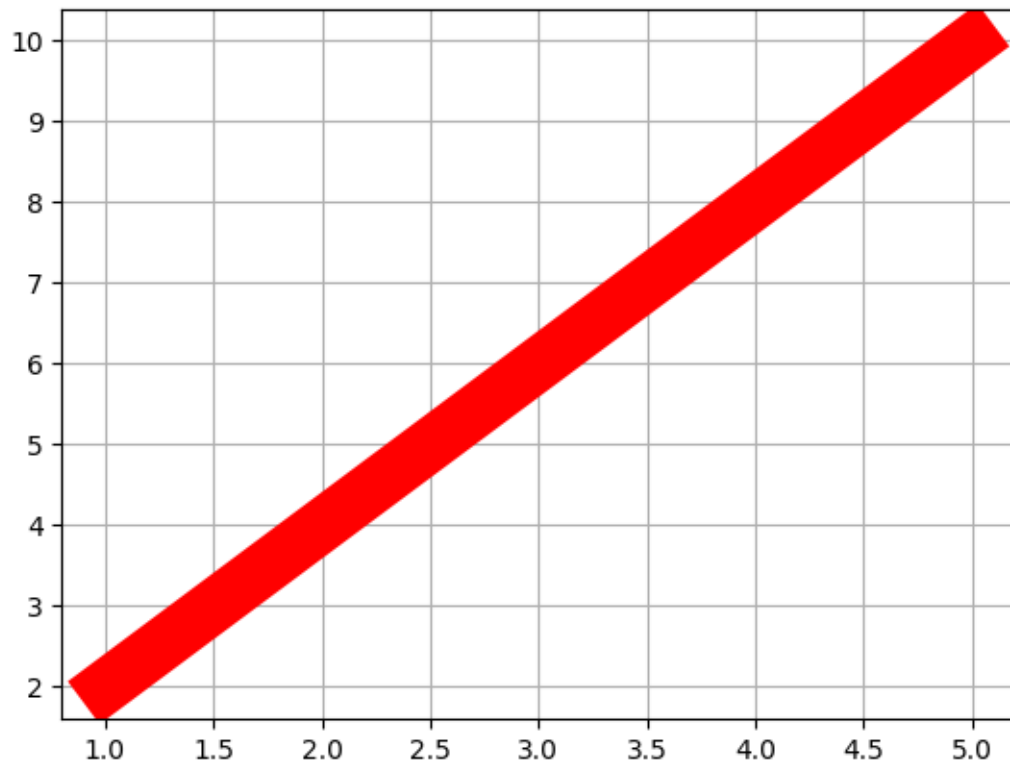
```
[ ]: import matplotlib.pyplot as plt
import numpy as np
x=np.array([1,2,3,4,5])
y=np.array([2,4,6,8,10])
plt.plot(x,y)
plt.show()
plt.plot(x,y,marker='o')
plt.show()
```

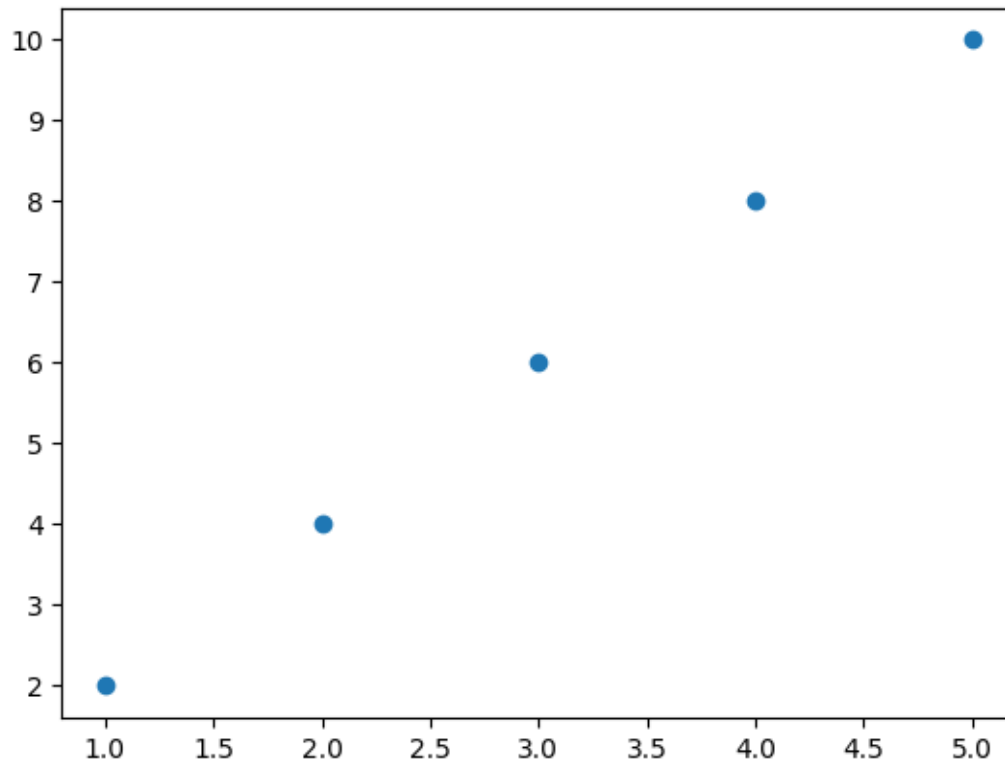




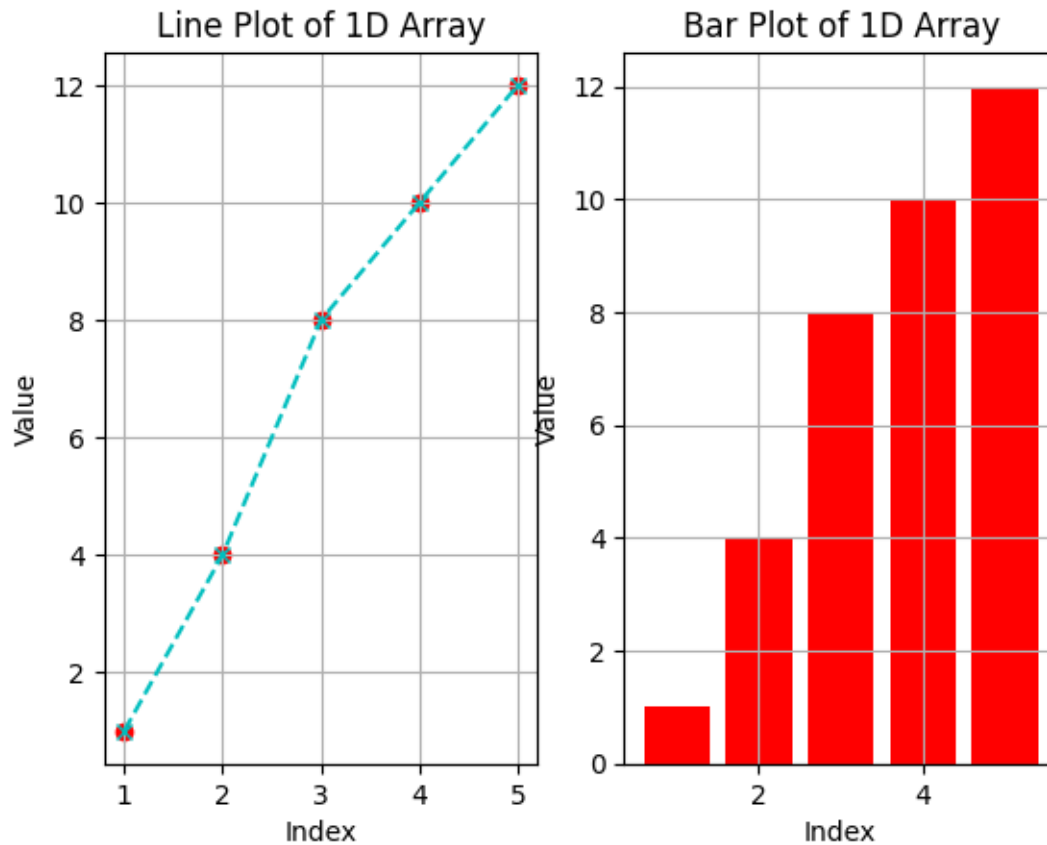
```
[ ]: plt.plot(x,y,linestyle='dashed')
plt.show()
plt.plot(x,y,'hotpink')
plt.show()
plt.plot(x,y,linewidth='20',color='r')
plt.grid()
plt.show()
plt.scatter(x,y)
plt.show()
```





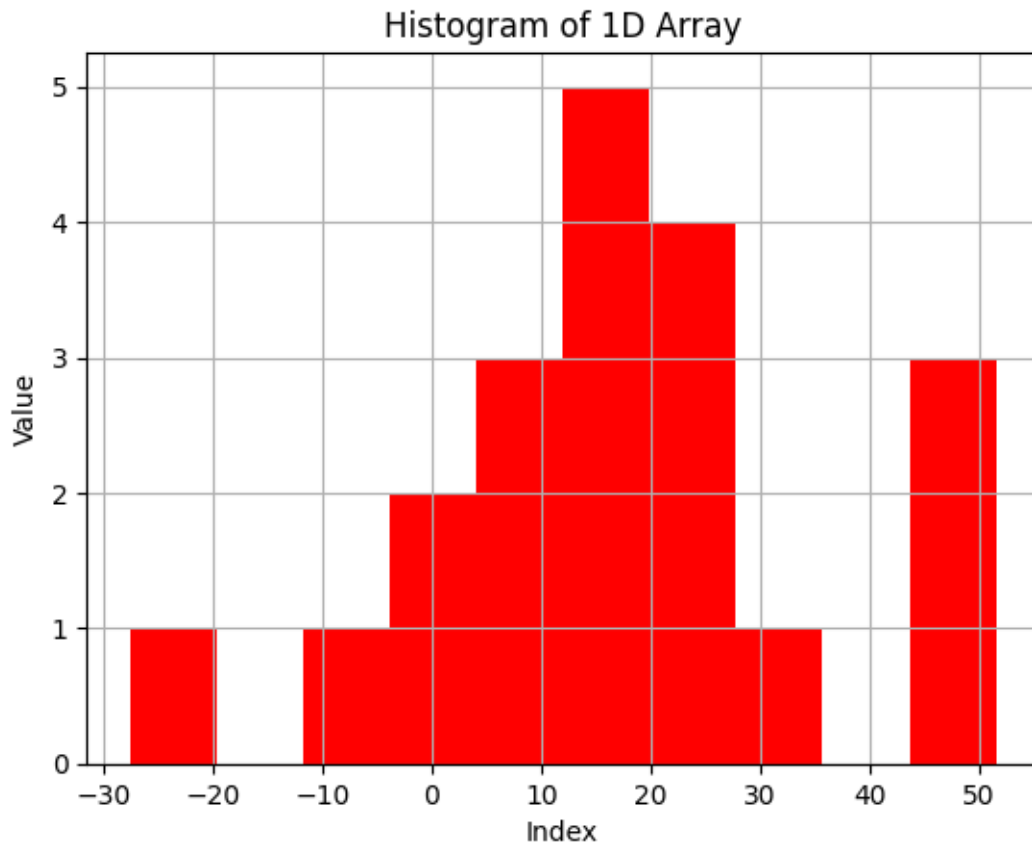


```
[ ]: import matplotlib.pyplot as plt
import numpy as np
a=np.array([1,2,3,4,5])
b=np.array([1,4,8,10,12])
plt.subplot(1,2,1)
plt.plot(a,b,marker='x',linestyle='dashed',color='c')
plt.grid(True)
plt.scatter(a,b,color='r')
plt.xlabel("Index")
plt.ylabel("Value")
plt.title("Line Plot of 1D Array")
plt.subplot(1,2,2)
plt.bar(a,b,color='r')
plt.grid(True)
plt.xlabel("Index")
plt.ylabel("Value")
plt.title("Bar Plot of 1D Array")
plt.show()
```

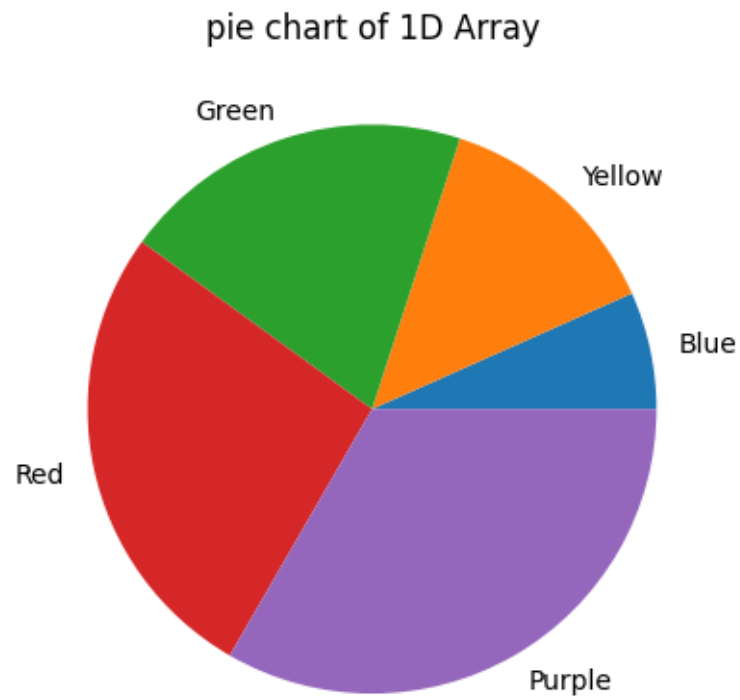


```
[ ]: e=np.random.normal(10,30,20)
print(e)
plt.hist(e,color='r')
plt.grid(True)
plt.xlabel("Index")
plt.ylabel("Value")
plt.title("Histogram of 1D Array")
plt.show()
```

```
[ -7.41795747  15.21535084  28.61545546  24.38023249   1.84268627
 18.32533036  24.12709405 -27.60880011  19.21801693  47.59375346
 21.2178869   11.3099262   16.09357102   9.52715624   4.79865675
 51.57387746  13.71585728  26.63283347  46.04787484   1.89978192]
```

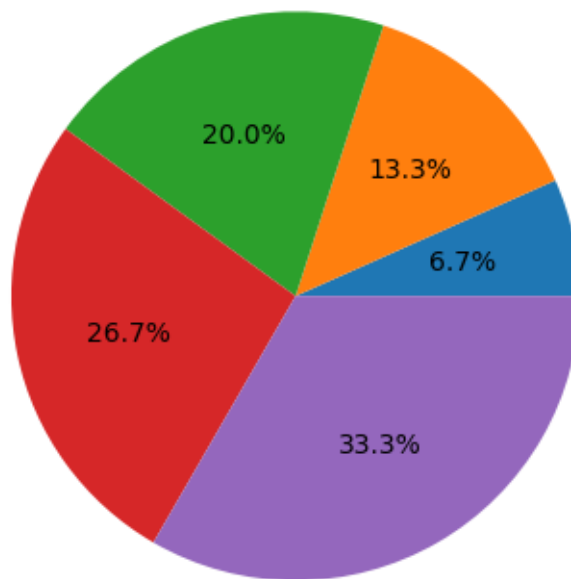



```
[ ]: c=np.array(['Blue','Yellow','Green','Red','Purple'])  
plt.pie(a, labels=c)  
plt.grid(True)  
plt.title("pie chart of 1D Array")  
plt.show()
```



```
[ ]: plt.pie(a, autopct='%1.1f%%')  
plt.title("pie chart of 1D Array")  
plt.show()
```

pie chart of 1D Array



EXP NO:8 TIMESERIES

```
[ ]: import datetime as dt
r=dt.datetime.now()
s=dt.datetime.today()
print(r)
print(s)
```

```
2024-10-30 12:45:13.908469
2024-10-30 12:45:13.908530
```

```
[ ]: t=r+dt.timedelta(days=1)
o=r-dt.timedelta(days=2)
print(t)
print(o)
```

```
2024-10-31 12:45:13.908469
2024-10-28 12:45:13.908469
```

```
[ ]: a=dt.datetime(2020,6,8,23,10,25,7264)
print(a)
```

```
[ ]: print(a.replace(day=26))  
      print(a.replace(month=12))
```

```
[ ]: print(dt.date(2004,10,1).ctime())
```

```
[ ]: print(r.strftime("%Y"))  
      print(r.strftime("%M"))  
      print(r.strftime("%b"))  
      print(r.strftime("%B"))  
      print(r.strftime("%j"))  
      print(r.strftime("%D"))  
      print(r.strftime("%d"))  
      print(r.strftime("%a"))  
      print(r.strftime("%A"))  
      print(r.strftime("%H"))  
      print(r.strftime("%S"))  
      print(r.strftime("%F"))  
      print(r.strftime("%p"))  
      print(r.strftime("%x"))  
      print(r.strftime("%X"))  
      #print(r.strftime("%c"))  
      #print(r.strftime("%I"))  
      #print(r.strftime("%m"))
```