

```
import pandas as pd
import numpy as np
```

```
x=[27,23,34,45,49,69,62,55,21,18]
print(x)
```

```
[27, 23, 34, 45, 49, 69, 62, 55, 21, 18]
```

```
df=pd.DataFrame({"age":x})
df.head()
```

	age
0	27
1	23
2	34
3	45
4	49

```
np.digitize(x, bins=[20,40,60,80])
```

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

```
df['binned']=pd.cut(x=df['age'], bins=[0,20,40,60,80], labels=['a', 'b', 'cc', 'd'])
df.head()
```

	age	binned
0	27	b
1	23	b
2	34	b
3	45	cc
4	49	cc

```
df['binned']=pd.cut(x=df['age'], bins=[0,20,40,60,80])
df.head()
```

	age	binned
0	27	(20, 40]
1	23	(20, 40]
2	34	(20, 40]
3	45	(40, 60]
4	49	(40, 60]

```
import pandas as pd
import numpy as np
import math
```

```
x=[12,4,5,21,7,14,15,18,20]
print(x)
data=np.sort(x)
print(data)
```

```
[12, 4, 5, 21, 7, 14, 15, 18, 20]
[ 4  5  7 12 14 15 18 20 21]
```

```
bin1=np.zeros((3,3))
bin2=np.zeros((3,3))
```

```
for i in range(0,9,3):
    k=int(i/3)
    mean=(x[i]+x[i+1]+x[i+2])/3
    for j in range(3):
        bin1[k,j]=mean
print(bin1)
```

```
[[ 7.         7.         7.         ]
 [14.         14.         14.         ]
 [17.66666667 17.66666667 17.66666667]]
```

```
for i in range(0,9,3):
    k=int(i/3)
    for j in range(3):
        bin2[k,j]=x[i+1]
print(bin2)
```

```
[[ 4.  4.  4.]
 [ 7.  7.  7.]
 [18. 18. 18.]]
```

```
points=[[2,3],[2,4],[4,1],[3,2]]
print(points)
```

```
dismatrix=[]
for p in points:
    list=[]
```

```

for j in points:
    list.append((abs(p[0]-j[0])+abs(p[1]-j[1])))
dismatrix.append(list)

```

```
(dismatrix)
```

```

for i in range(len(points)):
    for j in range(len(points)):
        if i<j:
            dismatrix[i][j]=0

```

```
(dismatrix)
```

```

[[2, 3], [2, 4], [4, 1], [3, 2]]
[[0, 0, 0, 0], [1, 0, 0, 0], [4, 5, 0, 0], [2, 3, 2, 0]]

```

```

import pandas as pd
import numpy as np

dict={'Stud Id':[1,2,3,4,5,6,7],
      'Stud Branch':['AI','Comp','Comp','Comp','AI','AI','AI'],
      'Marks':[17,np.nan,15,16,18,15,np.nan]}
df=pd.DataFrame(dict)
print(df)

print(df.Marks.isnull())

print(df.fillna(0))

```

	Stud Id	Stud Branch	Marks
0	1	AI	17.0
1	2	Comp	NaN
2	3	Comp	15.0
3	4	Comp	16.0
4	5	AI	18.0
5	6	AI	15.0
6	7	AI	NaN

```

0    False
1     True
2    False
3    False
4    False
5    False
6     True

```

```
Name: Marks, dtype: bool
```

	Stud Id	Stud Branch	Marks
0	1	AI	17.0
1	2	Comp	0.0
2	3	Comp	15.0
3	4	Comp	16.0
4	5	AI	18.0
5	6	AI	15.0
6	7	AI	0.0

```
import pandas as pd
```

B

```

x=[12000,145000,320000,45000,90000,134000,200000]

df=pd.DataFrame({'salary':x})
df.head()

min_max=df.copy()
for i in min_max:
    min_max[i]=(min_max[i]-min_max[i].min()/(min_max[i].max()-min_max[i].min()))

print(df,min_max)

```

```

      salary
0    12000
1   145000
2   320000
3    45000
4    90000
5   134000
6  200000
0    0.000000
1    0.431818
2    1.000000
3    0.107143
4    0.253247
5    0.396104
6    0.610390

```

```

import pandas as pd
import numpy as np
x=[12000,145000,320000,45000,90000,134000,200000]

df=pd.DataFrame({'salary':x})
df.head()

z_score=df.copy()
for i in z_score:
    z_score[i]=(z_score[i]-z_score[i].mean())/z_score[i].std()

print(z_score)

```

```

      salary
0   -1.194148
1    0.095587
2    1.792607
3   -0.874138
4   -0.437762
5   -0.011083
6    0.628936

```

```

def mean(arr):
    m=0
    for i in arr:
        m+=i
    res=m//len(arr)
    return res

```

```

def kmean(arr,m1,m2):
    cluster1=[]
    cluster2=[]
    for i in arr:
        if(abs(m1-i)<abs(m2-i)):
            cluster1.append(i)
        elif(abs(m1-i)==abs(m2-i)):
            if(len(cluster1)>len(cluster2)):
                cluster2.append(i)
            else:
                cluster1.append(i)
        else:
            cluster2.append(i)
    c1_mean=mean(cluster1)
    c2_mean=mean(cluster2)

    print('C1',c1_mean)
    print('C2',c2_mean)

    print('cluster1',cluster1)
    print('cluster2',cluster2)

    if(c1_mean==m1 and c2_mean==m2):
        print('-----Final clusters')
        print('cluster1',cluster1)
        print('cluster1',cluster1)
    else:
        kmean(arr,c1_mean,c2_mean)

arr=[2,5,8,12,14,16,22,24,30,32,35]
k=2
kmean(arr,5,14)

```

```

C1 5
C2 23
cluster1 [2, 5, 8]
cluster2 [12, 14, 16, 22, 24, 30, 32, 35]
C1 6
C2 24
cluster1 [2, 5, 8, 12]
cluster2 [14, 16, 22, 24, 30, 32, 35]
C1 8
C2 26
cluster1 [2, 5, 8, 12, 14]
cluster2 [16, 22, 24, 30, 32, 35]
C1 9
C2 28
cluster1 [2, 5, 8, 12, 14, 16]

```

```
cluster2 [22, 24, 30, 32, 35]
C1 9
C2 28
cluster1 [2, 5, 8, 12, 14, 16]
cluster2 [22, 24, 30, 32, 35]
-----Final clusters
cluster1 [2, 5, 8, 12, 14, 16]
cluster1 [2, 5, 8, 12, 14, 16]
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 22:30

