In [2]:
```
import pandas as pd
array=[1,3,5,7,9,11]
series_obj=pd.Series(array)
arr=series_obj.values
reshapped_arr=arr.reshape((3,2))
reshapped_arr
```

Out[2]:
```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]], dtype=int64)
```

In [7]:
```
import pandas as pd
array=['anjali','nisha','chhoti','gaurav','saurav','sandhya','vishu','gautam','koma
series_obj=pd.Series(array)
arr=series_obj.values
reshapped_arr=arr.reshape((5,2))
reshapped_arr
```

Out[7]:
```
array([['anjali', 'nisha'],
       ['chhoti', 'gaurav'],
       ['saurav', 'sandhya'],
       ['vishu', 'gautam'],
       ['komal', 'shivu']], dtype=object)
```

In [8]:
```
import pandas as pd
values=[['monday',65000,50000],
       ['tuesday',68000,45000],
       ['wednesday',70000,55000],
       ['thursday',60000,47000],
       ['friday',49000,25000],
       ['saturday',54000,35000],
       ['sunday',100000,70000]]
df=pd.DataFrame(values,columns=['days','patiens','recovery'])
df
```

Out[8]:

	days	patiens	recovery
0	monday	65000	50000
1	tuesday	68000	45000
2	wednesday	70000	55000
3	thursday	60000	47000
4	friday	49000	25000
5	saturday	54000	35000
6	sunday	100000	70000

In [13]:
```
reshapped_df=df.melt(id_vars=['days'])
reshapped_df
```

Out[13]:

	days	variable	value
0	monday	patiens	65000
1	tuesday	patiens	68000
2	wednesday	patiens	70000
3	thursday	patiens	60000
4	friday	patiens	49000
5	saturday	patiens	54000
6	sunday	patiens	100000
7	monday	recovery	50000
8	tuesday	recovery	45000
9	wednesday	recovery	55000
10	thursday	recovery	47000
11	friday	recovery	25000
12	saturday	recovery	35000
13	sunday	recovery	70000
14	monday	deaths	1500
15	tuesday	deaths	7250
16	wednesday	deaths	1400
17	thursday	deaths	4200
18	friday	deaths	3000
19	saturday	deaths	2000
20	sunday	deaths	4550

In [16]:

```
import pandas as pd
values=[['monday',65000,50000,1500],
        ['tuesday',68000,45000,7250],
        ['wednesday',70000,55000,1400],
        ['thursday',60000,47000,4200],
        ['friday',49000,25000,3000],
        ['saturday',54000,35000,2000],
        ['sunday',100000,70000,4550]]
df=pd.DataFrame(values,columns=['days','patients','recovery','deaths'])
df
```

Out[16]:

	days	patients	recovery	deaths
0	monday	65000	50000	1500
1	tuesday	68000	45000	7250
2	wednesday	70000	55000	1400
3	thursday	60000	47000	4200
4	friday	49000	25000	3000
5	saturday	54000	35000	2000
6	sunday	100000	70000	4550

In [17]:
```
reshapped_df=df.melt(id_vars=['patients'])
reshapped_df
```

Out[17]:

	patients	variable	value
0	65000	days	monday
1	68000	days	tuesday
2	70000	days	wednesday
3	60000	days	thursday
4	49000	days	friday
5	54000	days	saturday
6	100000	days	sunday
7	65000	recovery	50000
8	68000	recovery	45000
9	70000	recovery	55000
10	60000	recovery	47000
11	49000	recovery	25000
12	54000	recovery	35000
13	100000	recovery	70000
14	65000	deaths	1500
15	68000	deaths	7250
16	70000	deaths	1400
17	60000	deaths	4200
18	49000	deaths	3000
19	54000	deaths	2000
20	100000	deaths	4550

In [19]:
```
import pandas as pd
values=[[101,'anjali',455,'football'],
        [111,'chhoti',250,'chess'],
        [192,'gaurav',495,'cricket'],
        [201,'sai',400,'ludo'],
        [105,'radha',350,'badminton'],
        [118,'vansh','450','badminton']]
df=pd.DataFrame(values,columns=['ID','name','marks','sports'])
df
```

Out[19]:

	ID	name	marks	sports
0	101	anjali	455	football
1	111	chhoti	250	chess
2	192	gaurav	495	cricket
3	201	sai	400	ludo
4	105	radha	350	badminton
5	118	vansh	450	badminton

In [21]:
```
reshapped_df=df.pivot(index='name',columns='sports')
reshapped_df
```

Out[21]:

	ID					ma				
sports	**badminton**	**chess**	**cricket**	**football**	**ludo**	**badminton**	**chess**	**cricket**	**football**	**lu**
name										
anjali	NaN	NaN	NaN	101.0	NaN	NaN	NaN	NaN	455	N
chhoti	NaN	111.0	NaN	NaN	NaN	NaN	250	NaN	NaN	N
gaurav	NaN	NaN	192.0	NaN	NaN	NaN	NaN	495	NaN	N
radha	105.0	NaN	NaN	NaN	NaN	350	NaN	NaN	NaN	N
sai	NaN	NaN	NaN	NaN	201.0	NaN	NaN	NaN	NaN	4
vansh	118.0	NaN	NaN	NaN	NaN	450	NaN	NaN	NaN	N

In [23]:
```
reshapped_df=df.pivot(index='ID',columns='marks',values='sports')
reshapped_df
```

Out[23]:

marks	**250**	**350**	**400**	**455**	**495**	**450**
ID						
101	NaN	NaN	NaN	football	NaN	NaN
105	NaN	badminton	NaN	NaN	NaN	NaN
111	chess	NaN	NaN	NaN	NaN	NaN
118	NaN	NaN	NaN	NaN	NaN	badminton
192	NaN	NaN	NaN	NaN	cricket	NaN
201	NaN	NaN	ludo	NaN	NaN	NaN

In [26]:
```
reshapped_df=df.pivot(index='ID',columns='marks',values='sports')
df_new=reshapped_df.reset_index()
```

```
df_new
```

Out[26]:

marks	ID	250	350	400	455	495	450
0	101	NaN	NaN	NaN	football	NaN	NaN
1	105	NaN	badminton	NaN	NaN	NaN	NaN
2	111	chess	NaN	NaN	NaN	NaN	NaN
3	118	NaN	NaN	NaN	NaN	NaN	badminton
4	192	NaN	NaN	NaN	NaN	cricket	NaN
5	201	NaN	NaN	ludo	NaN	NaN	NaN

In [27]:

```
import pandas as pd
df=pd.DataFrame({'a':['anjali','gaurav','saurav'],
                 'b':['mca','bca','btech'],
                 'c':[21,22,23]})
df
```

Out[27]:

	a	b	c
0	anjali	mca	21
1	gaurav	bca	22
2	saurav	btech	23

In [28]:

```
df.pivot(index='a',columns='b',values='c')
```

Out[28]:

b	bca	btech	mca
a			
anjali	NaN	NaN	21.0
gaurav	22.0	NaN	NaN
saurav	NaN	23.0	NaN

In [29]:

```
df.pivot(index='a',columns='b',values=['c','a'])
```

Out[29]:

	c			a		
b	bca	btech	mca	bca	btech	mca
a						
anjali	NaN	NaN	21	NaN	NaN	anjali
gaurav	22	NaN	NaN	gaurav	NaN	NaN
saurav	NaN	23	NaN	NaN	saurav	NaN

In [33]:
```
import pandas as pd
df=pd.DataFrame({'name':['gaurav','anjali','nisha','monu','saurav','komal'],
                 'gender':['m','f','f','m','m','f'],
                 'age':[45,6,4,36,12,43]})
df
```

Out[33]:

	name	gender	age
0	gaurav	m	45
1	anjali	f	6
2	nisha	f	4
3	monu	m	36
4	saurav	m	12
5	komal	f	43

In [35]:
```
def age_bucket(Age):
    if Age<=18:
        return"<18"
    else:
        return">18"
df['Age Group']=df['age'].apply(age_bucket)
df['Age Group']
```

Out[35]:
```
0    >18
1    <18
2    <18
3    >18
4    <18
5    >18
Name: Age Group, dtype: object
```

In [36]:
```
Gender=pd.DataFrame(df.gender.value_counts(normalize=True)*100).reset_index()
Gender.columns=['gender','%gender']
df=pd.merge(left=df,right=Gender,how='inner',on=['gender'])
df
```

Out[36]:

	name	gender	age	Age Group	%gender
0	gaurav	m	45	>18	50.0
1	monu	m	36	>18	50.0
2	saurav	m	12	<18	50.0
3	anjali	f	6	<18	50.0
4	nisha	f	4	<18	50.0
5	komal	f	43	>18	50.0

In [39]:
```
table=pd.pivot_table(df,index=['gender','%gender','Age Group'],
                     values=['name'],aggfunc={'name':'count',})
table
```

Out[39]:

			name
gender	**%gender**	**Age Group**	
f	**50.0**	**<18**	2
		>18	1
m	**50.0**	**<18**	1
		>18	2

In [ ]: