In [2]:

```
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

Out[2]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

MODULE-1

DATA CLEANING

In [6]:

```
dataset.isnull().sum().sum()
```

Out[6]:

```
0
```

DATA NORMALIZATION

SIMPLE SCALING NORMALIZATION

In [3]:

```
#simple feature scaling normalization for iMDB ratings column only
dataset['Hits'] = dataset['Hits'] / dataset['Hits'].max()
dataset
```

Out[3]:

	Actor Name	Hits	Flops	Total
0	Vijay	0.50	2	12
1	Ajith	0.50	2	12
2	Suriya	0.60	2	14
3	Karthik	0.25	1	6
4	Rana	0.30	2	8
5	Sivakarthikeyan	0.60	0	12
6	Rajini	0.75	2	17
7	Kamal	0.80	3	19
8	Dulquer	0.60	3	15
9	Ranbir	0.15	2	5
10	Jayam	0.50	2	12
11	Allu	0.60	0	12
12	Fahad	0.50	2	12
13	Nivin	0.50	4	14
14	Nani	1.00	0	20
15	Sharuk	0.50	3	13
16	Aamir	0.60	2	14
17	Tarak	1.00	0	20
18	Akshay	0.50	3	13
19	Dhanush	0.60	3	15
20	Ram	1.00	1	21
21	Samantha	0.50	2	12
22	Nayanthara	0.60	3	15
23	Jyothika	0.75	2	17
24	Anushka	0.80	0	16
25	Shraddha	0.70	2	16
26	Deepika	0.90	3	21
27	Olivia	0.25	0	5
28	Keerthi	0.70	2	16
29	Rashmika	0.60	2	14

MAXIMUM- MINIMUM NORMALZATION

In [4]:

```
#Min-Max Normalization method
dataset_mmn = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset_mmn['Flops']=(dataset_mmn['Flops']-dataset_mmn['Flops'].min()) / (dataset_mmn['Flop
dataset_mmn
```

Out[4]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	0.50	12
1	Ajith	10	0.50	12
2	Suriya	12	0.50	14
3	Karthik	5	0.25	6
4	Rana	6	0.50	8
5	Sivakarthikeyan	12	0.00	12
6	Rajini	15	0.50	17
7	Kamal	16	0.75	19
8	Dulquer	12	0.75	15
9	Ranbir	3	0.50	5
10	Jayam	10	0.50	12
11	Allu	12	0.00	12
12	Fahad	10	0.50	12
13	Nivin	10	1.00	14
14	Nani	20	0.00	20
15	Sharuk	10	0.75	13
16	Aamir	12	0.50	14
17	Tarak	20	0.00	20
18	Akshay	10	0.75	13
19	Dhanush	12	0.75	15
20	Ram	20	0.25	21
21	Samantha	10	0.50	12
22	Nayanthara	12	0.75	15
23	Jyothika	15	0.50	17
24	Anushka	16	0.00	16
25	Shraddha	14	0.50	16
26	Deepika	18	0.75	21
27	Olivia	5	0.00	5
28	Keerthi	14	0.50	16
29	Rashmika	12	0.50	14

Z-SCORE NORMALIZATION

In [5]:

```
#Z-score Normalization: For each value, subtract the mean which is the average of the featu
dataset_zs = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset_zs['Flops'] = (dataset_zs['Flops'] -dataset_zs['Flops'].mean()) / dataset_zs['Flops
dataset_zs
```

Out[5]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	0.149243	12
1	Ajith	10	0.149243	12
2	Suriya	12	0.149243	14
3	Karthik	5	-0.746214	6
4	Rana	6	0.149243	8
5	Sivakarthikeyan	12	-1.641671	12
6	Rajini	15	0.149243	17
7	Kamal	16	1.044700	19
8	Dulquer	12	1.044700	15
9	Ranbir	3	0.149243	5
10	Jayam	10	0.149243	12
11	Allu	12	-1.641671	12
12	Fahad	10	0.149243	12
13	Nivin	10	1.940157	14
14	Nani	20	-1.641671	20
15	Sharuk	10	1.044700	13
16	Aamir	12	0.149243	14
17	Tarak	20	-1.641671	20
18	Akshay	10	1.044700	13
19	Dhanush	12	1.044700	15
20	Ram	20	-0.746214	21
21	Samantha	10	0.149243	12
22	Nayanthara	12	1.044700	15
23	Jyothika	15	0.149243	17
24	Anushka	16	-1.641671	16
25	Shraddha	14	0.149243	16
26	Deepika	18	1.044700	21
27	Olivia	5	-1.641671	5
28	Keerthi	14	0.149243	16
29	Rashmika	12	0.149243	14

MODULE-2

In [13]:

```
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

Out[13]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

In [14]:

```
#Slicing only row
dataset.iloc[0:1]
```

Out[14]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12

In [15]:

```
#Slicing row and coloumn
dataset.iloc[0:4,0:3]
```

Out[15]:

	Actor Name	Hits	Flops
0	Vijay	10	2
1	Ajith	10	2
2	Suriya	12	2
3	Karthik	5	1

In [16]:

```
#Negative Slicing
dataset.iloc[-1:-5:-1,-1:-5:-1]
```

Out[16]:

	Total	Flops	Hits	Actor Name
29	14	2	12	Rashmika
28	16	2	14	Keerthi
27	5	0	5	Olivia
26	21	3	18	Deepika

In [17]:

```
dataset["Hits"]>5
dataset[dataset["Hits"]>5]
```

Out[17]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
28	Keerthi	14	2	16
29	Rashmika	12	2	14

GROUPBY OPERATIONS

In [24]:

```
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

Out[24]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

In [27]:

```
#group by single column
group1=dataset.groupby(['Hits'])
group1
```

Out[27]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000025BBBEBA820>
```

In [28]:

```
#display the groups
print(group1.groups)
print(type(group1))
dataset.loc[7]
```

```
{3: [9], 5: [3, 27], 6: [4], 10: [0, 1, 10, 12, 13, 15, 18, 21], 12: [2, 5,
8, 11, 16, 19, 22, 29], 14: [25, 28], 15: [6, 23], 16: [7, 24], 18: [26], 2
0: [14, 17, 20]}
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

Out[28]:

```
Actor Name    Kamal
Hits             16
Flops             3
Total            19
Name: 7, dtype: object
```

In [29]:

```
# to print max
print("Maximum in each group:\n",group1.max()) # max()
```

```
Maximum in each group:
      Actor Name  Flops  Total
Hits
3        Ranbir      2      5
5        Olivia      1      6
6          Rana      2      8
10        Vijay      4     14
12       Suriya      3     15
14     Shraddha      2     16
15       Rajini      2     17
16        Kamal      3     19
18      Deepika      3     21
20        Tarak      1     21
```

In [30]:

```
# max and min function can be applied over columns with string values
print("Minimum in each group:\n",group1.min())
```

```
Minimum in each group:
      Actor Name  Flops  Total
Hits
3        Ranbir      2      5
5       Karthik      0      5
6          Rana      2      8
10        Ajith      2     12
12        Aamir      0     12
14      Keerthi      2     16
15     Jyothika      2     17
16      Anushka      0     16
18      Deepika      3     21
20         Nani      0     20
```

In [31]:

```
#sum function can be applied over solumns with numerical values
print("Sum across each group:\n",group1.sum())
```

```
Sum across each group:
       Flops  Total
Hits
3          2      5
5          1     11
6          2      8
10        20    100
12        15    111
14         4     32
15         4     34
16         3     35
18         3     21
20         1     61
```

In [32]:

```
#sum function can be applied over solumns with numerical values
print("mean across each group:\n",group1.mean())
```

```
mean across each group:
          Flops      Total
Hits
3     2.000000   5.000000
5     0.500000   5.500000
6     2.000000   8.000000
10    2.500000  12.500000
12    1.875000  13.875000
14    2.000000  16.000000
15    2.000000  17.000000
16    1.500000  17.500000
18    3.000000  21.000000
20    0.333333  20.333333
```

In [33]:

```
#number of instances in dataframe
print("Length of dataframe:",len(dataset))
print("Length of groups:",len(group1))
#to print number of unique groups
print("Number of unique items:\n",group1['Hits'].nunique())
#Finding Most albums based on majority
print(group1.size())
print("Using dataframe object :\n",dataset["Hits"].value_counts())
# to get first row in each group
print(group1.first())
# to get last row in each group
print(group1.last())
```

```
Length of dataframe: 30
Length of groups: 10
Number of unique items:
 Hits
3      1
5      1
6      1
10     1
12     1
14     1
15     1
16     1
18     1
20     1
Name: Hits, dtype: int64
Hits
3      1
5      2
6      1
10     8
12     8
14     2
15     2
16     2
18     1
20     3
dtype: int64
Using dataframe object :
 10    8
12     8
20     3
5      2
15     2
16     2
14     2
6      1
3      1
18     1
Name: Hits, dtype: int64
     Actor Name  Flops  Total
Hits
3         Ranbir      2      5
5        Karthik      1      6
6           Rana      2      8
10         Vijay      2     12
```

```
12        Suriya      2     14
14      Shraddha      2     16
15        Rajini      2     17
16         Kamal      3     19
18       Deepika      3     21
20          Nani      0     20
     Actor Name  Flops  Total
Hits                         
3        Ranbir      2      5
5        Olivia      0      5
6          Rana      2      8
10     Samantha      2     12
12     Rashmika      2     14
14      Keerthi      2     16
15     Jyothika      2     17
16      Anushka      0     16
18      Deepika      3     21
20          Ram      1     21
```

MODULE-3

In [35]:

```
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

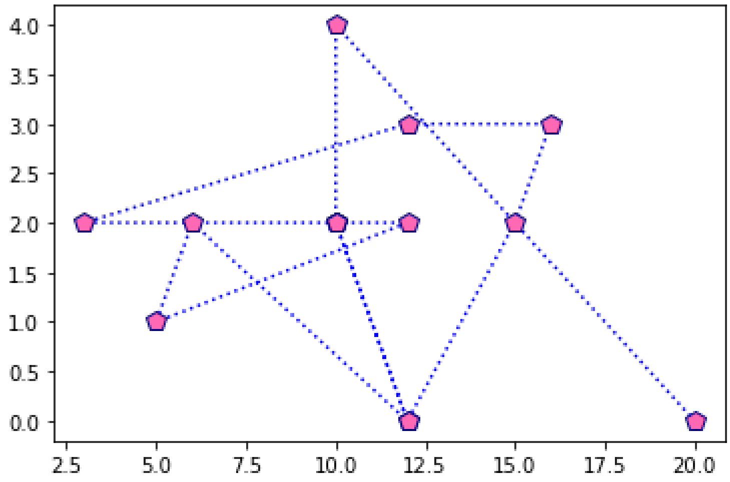
Out[35]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

```
PLOTS
```

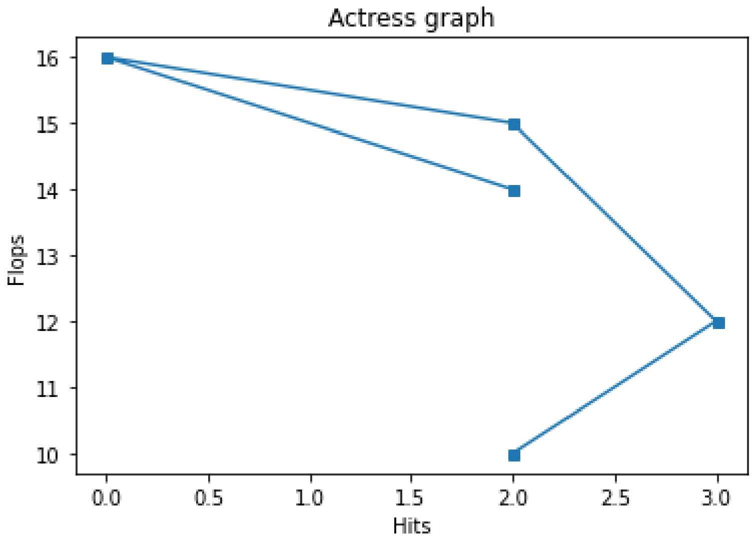
In [36]:

```
# No of Hits
import matplotlib.pyplot as plt
import numpy as np
xaxis = np.array([10,10,12,5,6,12,15,16,12,3,10,12,10,10,20])
yaxis = np.array([2,2,2,1,2,0,2,3,3,2,2,0,2,4,0])
plt.plot(xaxis,yaxis,'p:b',mec='navy',mfc='hotpink',ms=10)
plt.show()
```



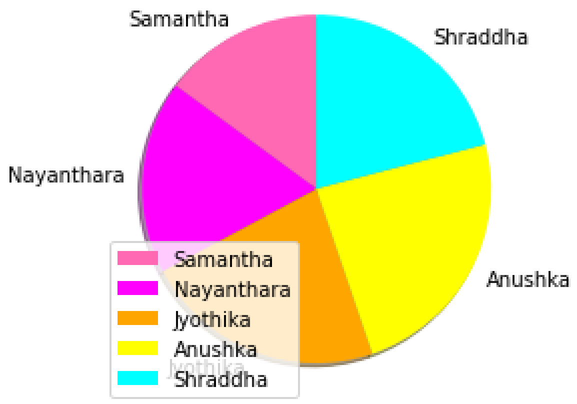
In [37]:

```
#Actress graph
import matplotlib.pyplot as plt
import pandas as pd
sales = pd.DataFrame({'Actress':['Samantha','Nayanthara','Jyothika','Anushka','Shraddha'],
                      'Hits':[10,12,15,16,14],
                      'Flops':[2,3,2,0,2]})
plt.plot(sales["Flops"],sales["Hits"],markersize=5,marker='s')
plt.xlabel('Hits')
plt.ylabel('Flops')
plt.title('Actress graph')
plt.show()
```



In [38]:

```
#piechart
import matplotlib.pyplot as plt
import pandas as pd
data1 = pd.DataFrame({'Actress':['Samantha','Nayanthara','Jyothika','Anushka','Shraddha'],
                      'Hits':[10,12,15,16,14]})
plt.pie(data1['Hits'],labels=data1['Actress'],explode=[0,0,0,0,0], colors=['hotpink','magen
plt.legend(loc='lower left')
plt.show()
```



In [43]:

```
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

Out[43]:

	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

In [45]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv("C:/Users/user/Desktop/adv python proj.csv")
dataset
```

Out[45]:

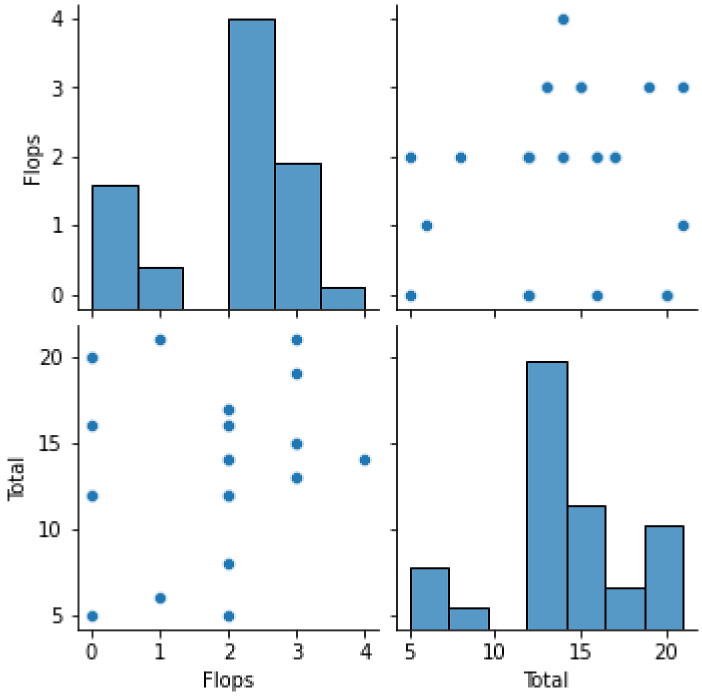
	Actor Name	Hits	Flops	Total
0	Vijay	10	2	12
1	Ajith	10	2	12
2	Suriya	12	2	14
3	Karthik	5	1	6
4	Rana	6	2	8
5	Sivakarthikeyan	12	0	12
6	Rajini	15	2	17
7	Kamal	16	3	19
8	Dulquer	12	3	15
9	Ranbir	3	2	5
10	Jayam	10	2	12
11	Allu	12	0	12
12	Fahad	10	2	12
13	Nivin	10	4	14
14	Nani	20	0	20
15	Sharuk	10	3	13
16	Aamir	12	2	14
17	Tarak	20	0	20
18	Akshay	10	3	13
19	Dhanush	12	3	15
20	Ram	20	1	21
21	Samantha	10	2	12
22	Nayanthara	12	3	15
23	Jyothika	15	2	17
24	Anushka	16	0	16
25	Shraddha	14	2	16
26	Deepika	18	3	21
27	Olivia	5	0	5
28	Keerthi	14	2	16
29	Rashmika	12	2	14

In [46]:

```
#pairplot
sns.pairplot(dataset,vars=['Flops','Total']) #drawing pair plot only for column in the list
```

Out[46]:

```
<seaborn.axisgrid.PairGrid at 0x25bbd6fb9d0>
```

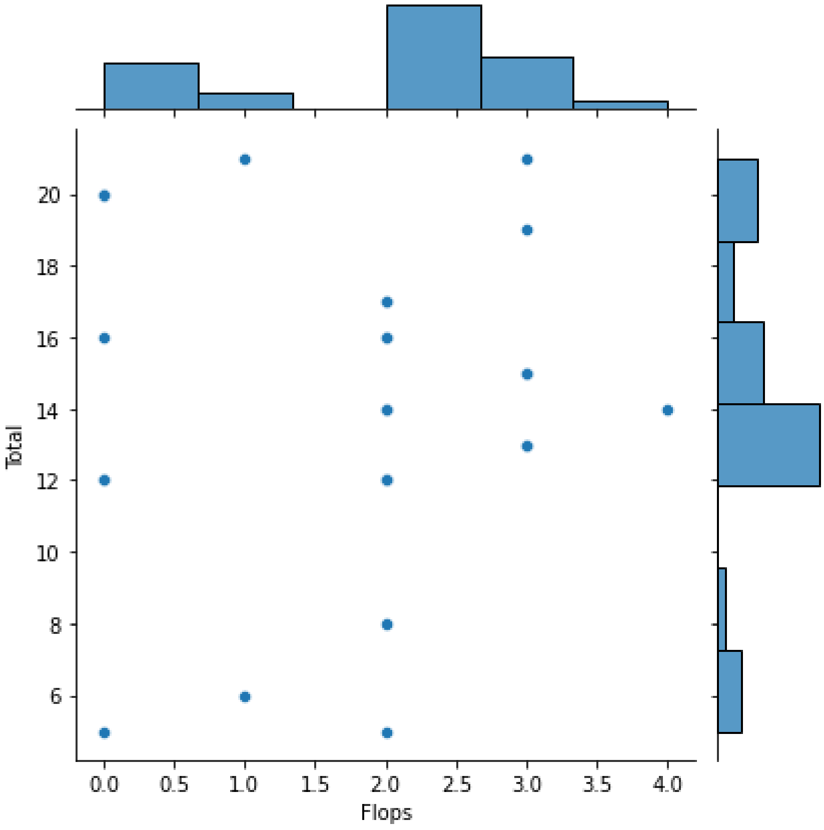


In [48]:

```
#jointplot
sns.jointplot(x=dataset["Flops"],y=dataset["Total"])
```

Out[48]:

```
<seaborn.axisgrid.JointGrid at 0x25bbf920b80>
```

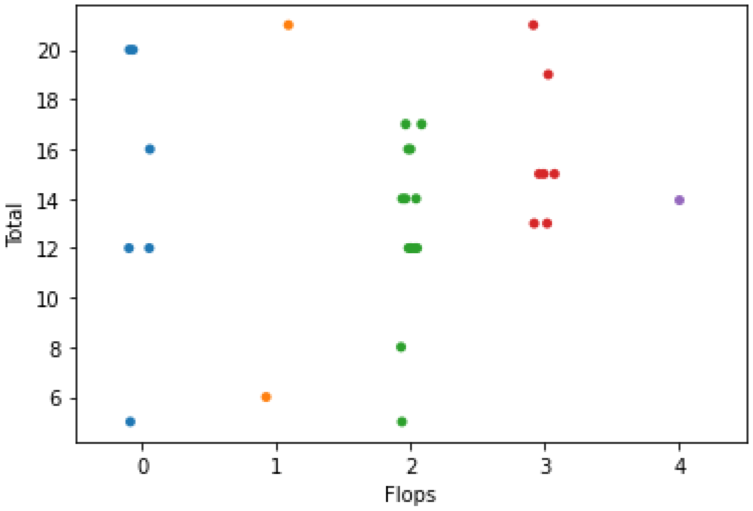


In [49]:

```
#stripplot
sns.stripplot(x=dataset['Flops'],y=dataset['Total'])
```

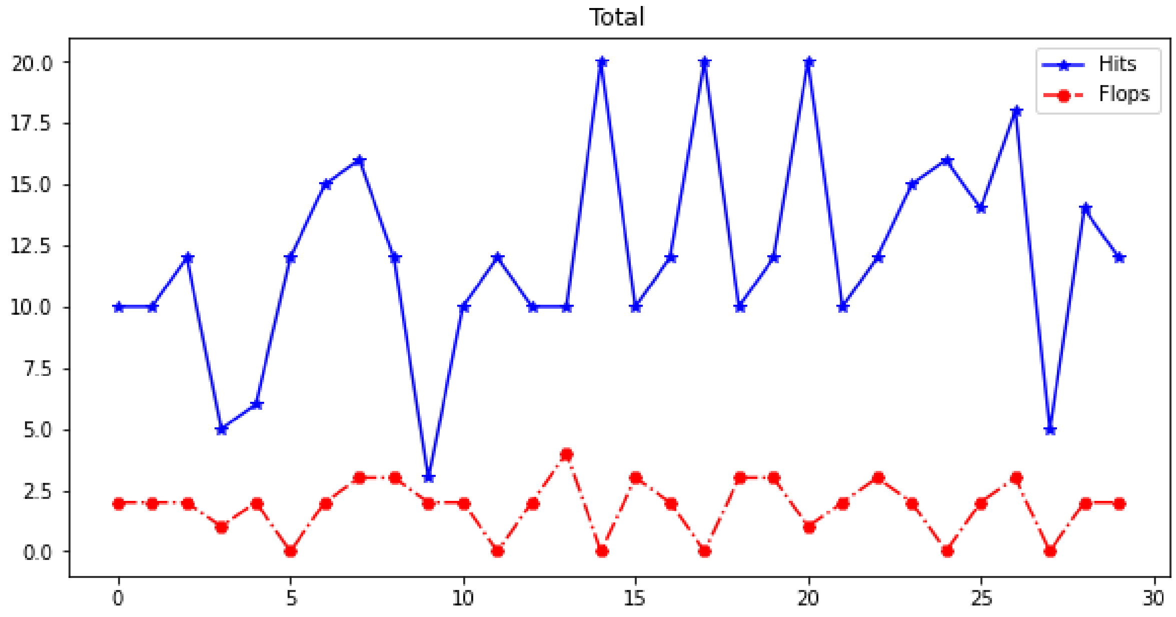
Out[49]:

```
<AxesSubplot:xlabel='Flops', ylabel='Total'>
```



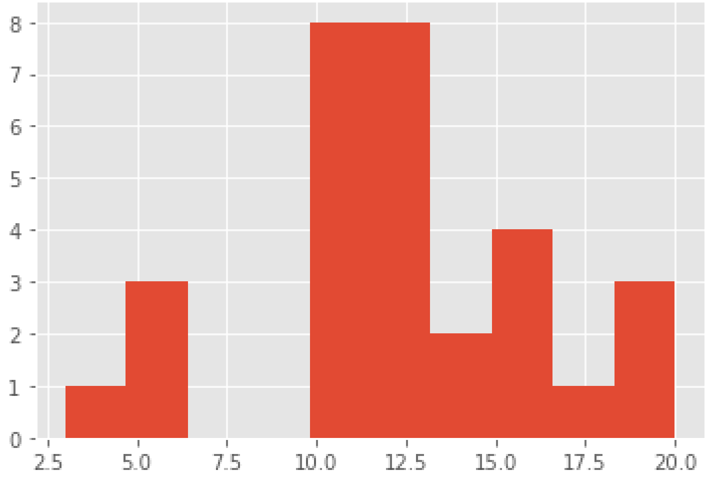
In [51]:

```
import pandas as pd
import matplotlib.pyplot as plt
dataset.plot(kind='line',
y=['Hits','Flops'],
          style=['b-*','r-.H'],
          figsize = (10,5),
          title = "Total",
          label = ['Hits','Flops'] # label should be samelength as of y
         )
plt.show()
```



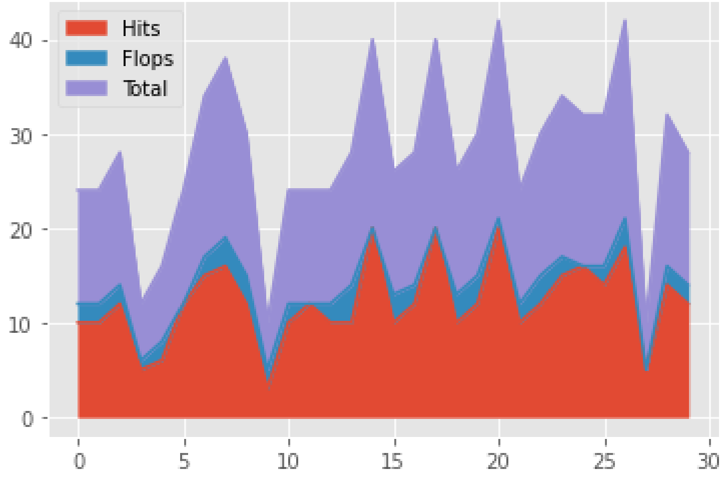
In [52]:

```
# Histogram
plt.style.use('ggplot')
dataset['Hits'].hist()
plt.show()
```



In [53]:

```
#Area plot
dataset.plot.area()
plt.show()
```

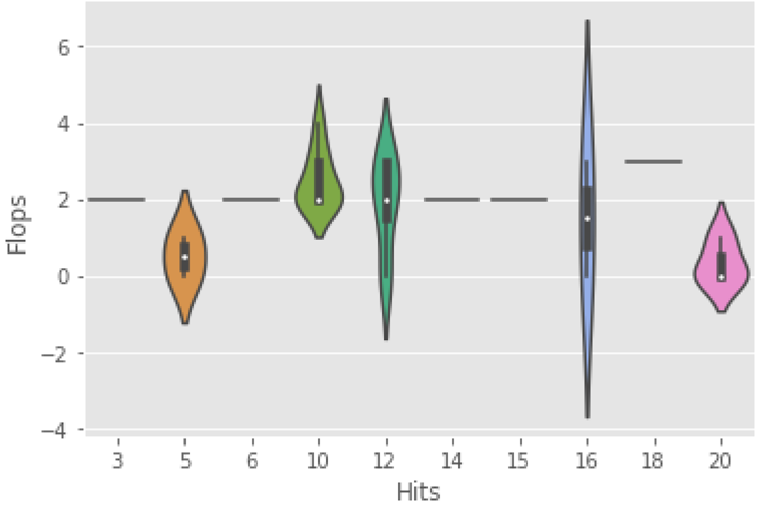


In [54]:

```
# violinplot
sns.violinplot(x=dataset['Hits'],y=dataset['Flops'] )
```

Out[54]:

```
<AxesSubplot:xlabel='Hits', ylabel='Flops'>
```



MODULE-4

```
LINEAR REGRESSION
```

In [2]:

```
#linear regression
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:

```
#import the dataset
dataset = pd.read_csv('C:/Users/user/Desktop/adv python project.csv')
dataset.describe()
```

Out[3]:

	Hits	Flops	Total
count	30.000000	30.000000	30.000000
mean	12.100000	1.833333	13.933333
std	4.285903	1.116748	4.233962
min	3.000000	0.000000	5.000000
25%	10.000000	1.250000	12.000000
50%	12.000000	2.000000	14.000000
75%	14.750000	2.750000	16.000000
max	20.000000	4.000000	21.000000

In [4]:

```
X = dataset.iloc[:, :-1].values # everything except last column
y = dataset.iloc[:, 1].values # only last column
```

In [5]:

```
# Split dataset into Training and Test set
# sklearn.cross_validation - Deprecated
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state= 0)
```

In [7]:

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
y_train = sc_y.fit_transform(np.array(y_train).reshape(-1, 1)) # makes y_train to one colum
X_test = sc_X.fit_transform(X_test)
y_test_org = y_test
y_test = sc_y.fit_transform(np.array(y_test).reshape(-1, 1))
```

In [8]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[8]:

```
LinearRegression()
```

In [9]:

```
# Predict Test set
y_pred = regressor.predict(X_test)
y_pred
```

Out[9]:

```
array([[-0.13736056],
       [-0.13736056],
       [ 1.5109662 ],
       [-0.13736056],
       [ 0.68680282],
       [-1.78568733]])
```

In [10]:

```
# Print unscaled test and predicted values
y_pred_inv = sc_y.inverse_transform(y_pred)
print(pd.DataFrame(np.column_stack((y_test_org, y_pred_inv))))
```

```
     0             1
0  2.0  2.000000e+00
1  2.0  2.000000e+00
2  4.0  4.000000e+00
3  2.0  2.000000e+00
4  3.0  3.000000e+00
5  0.0 -8.881784e-16
```

In [12]:

```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error, explained_va
print("Mean absolute error: %.2f" % mean_absolute_error(y_test, y_pred))
print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))
print("Root Mean squared error: %.2f" % np.sqrt(mean_squared_error(y_test, y_pred)))
print('Variance score: %.2f' % explained_variance_score(y_test, y_pred))
# Coefficient of determination
print('R^2 Square value', r2_score(y_test, y_pred))
```

```
Mean absolute error: 0.00
Mean squared error: 0.00
Root Mean squared error: 0.00
Variance score: 1.00
R^2 Square value 1.0
```

In [ ]:

```

```