Matplotlib

Matplotlib is a widely used plotting library for the Python programming language that provides a flexible and comprehensive way to create. # a variety of static, animated, and interactive viusalizations. # To install Matplotlib => pip install matplotlib

Line Plot:-

A line plot (or line chart) is a type of data visualization used to display information as a series of data points connected by straight line segments. It is commonly used to represent data over time, making it useful for showing trends and changes.

In [53]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
print("Libraries")
```

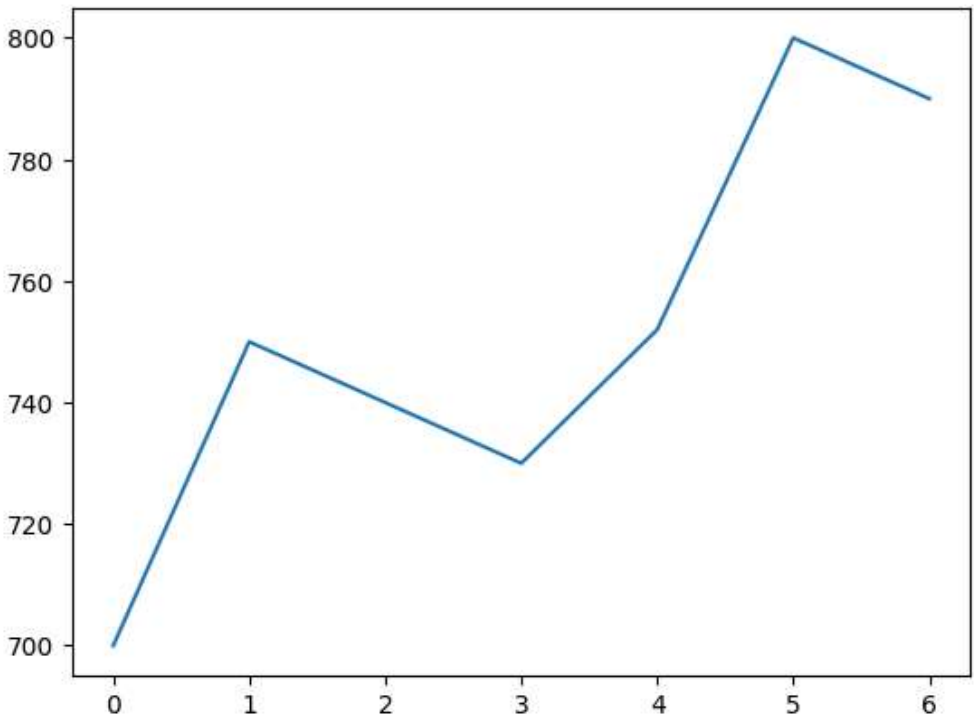
```
Libraries
```

data visualation.....

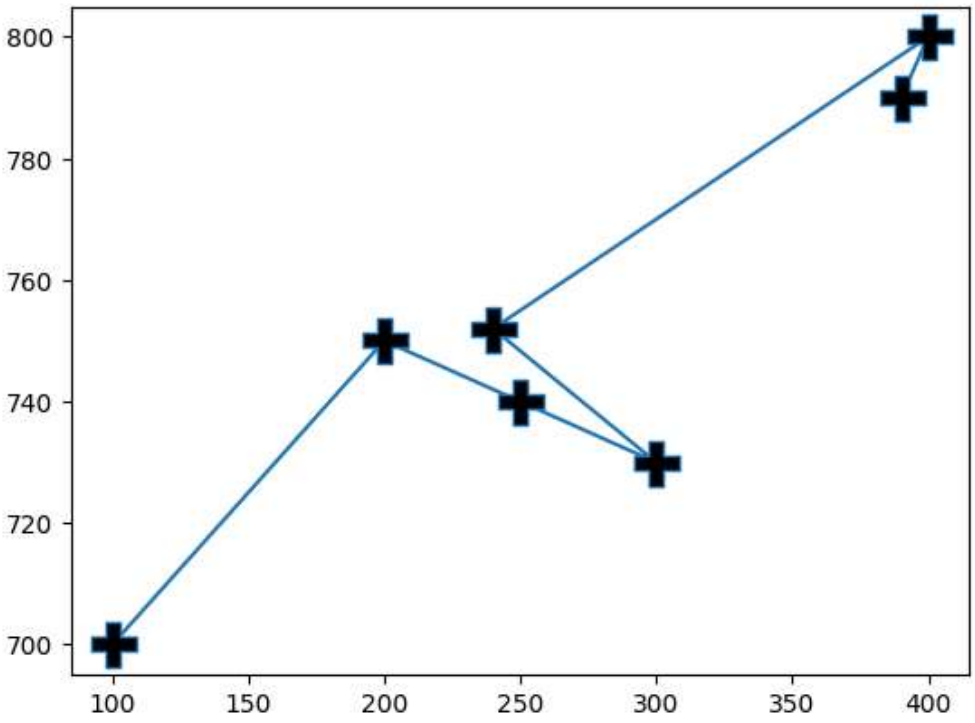
In [54]:
```
country_population = [700,750,740,730,752,800,790]
country_population
```

Out[54]:
```
[700, 750, 740, 730, 752, 800, 790]
```

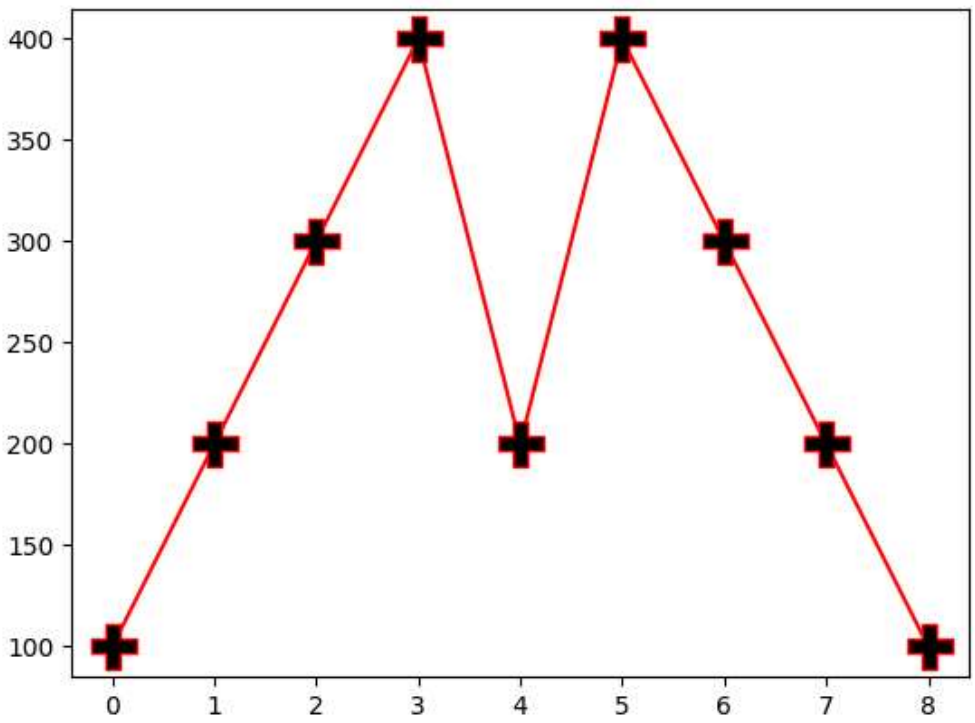
In [55]:
```
plt.plot(country_population)
plt.show()
```



In [56]:
```
country_population = [700,750,740,730,752,800,790]
x_axis = [100,200,250,300,240,400,390]
plt.plot(x_axis,country_population,marker = 'P',ms = 18,mfc = 'black')
plt.show()
```

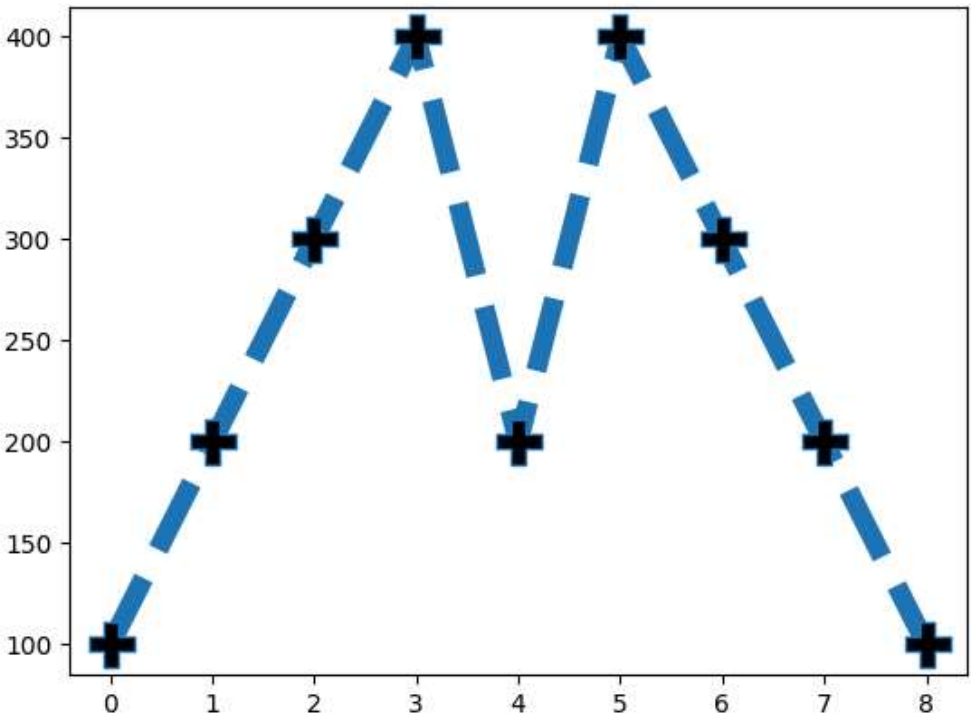


In [57]:
```
country_population = [100,200,300,400,200,400,300,200,100]
#x_axis = [100,200,250,300,240,400,390]
plt.plot(country_population,marker = 'P',ms = 18,mfc = 'black',color='red')
plt.show()
```



In [58]:
```
country_population = [100,200,300,400,200,400,300,200,100]
#x_axis = [100,200,250,300,240,400,390]
```

```
plt.plot(country_population,marker = 'P',ms = 18,mfc = 'black',ls='dashed',lw=8)
plt.show()
```

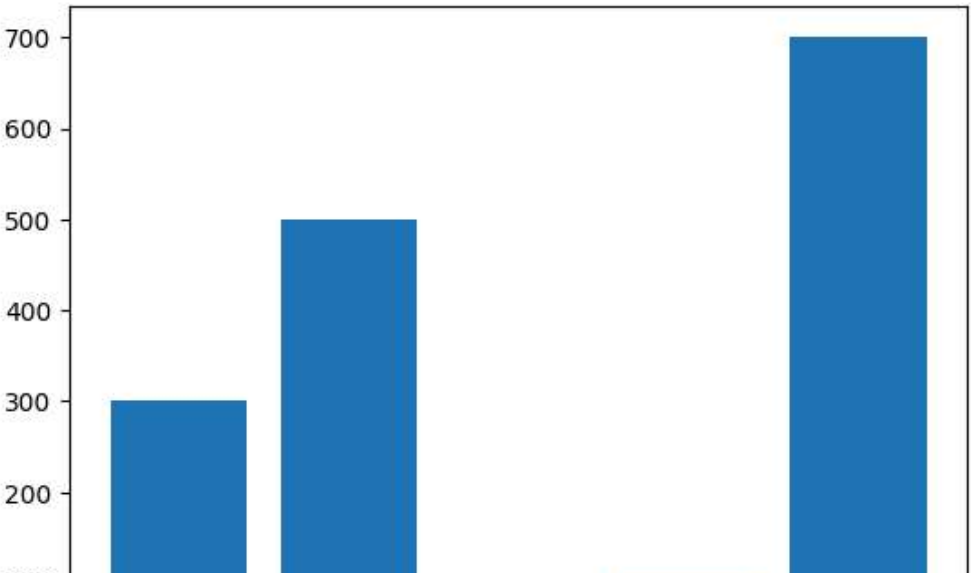


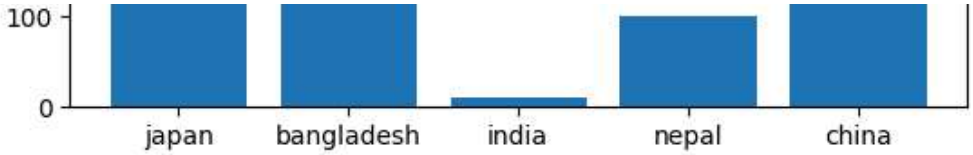
Bar plot:

Bar plots are a versatile and widely used type of plot in data visualization for displayong categorical data with rectangular bars.

Each bar's length or height is proportional to the value it represents. Bar plots can be vertical or horizontal and are useful for comparing different groups or tracking changes over time.

In [59]:
```
country_population = [300,500,10,100,700]
country_name = ['japan','bangladesh','india','nepal','china']
plt.bar(country_name,country_population)
plt.show()
```





In [60]:
```
df = pd.read_csv('Used_Bikes.csv')
df.head()
```

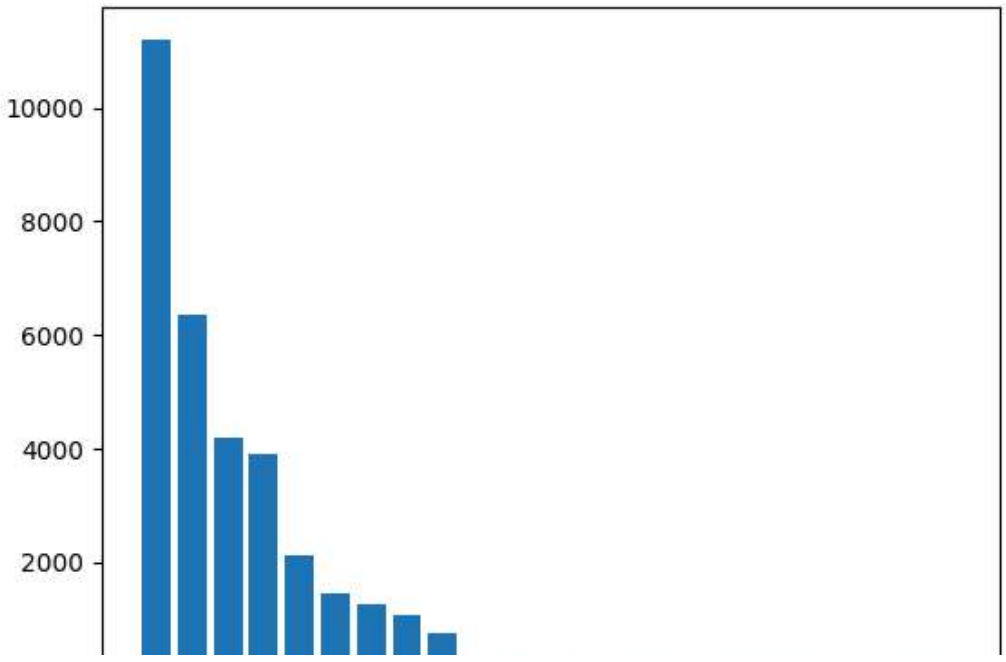
Out[60]:

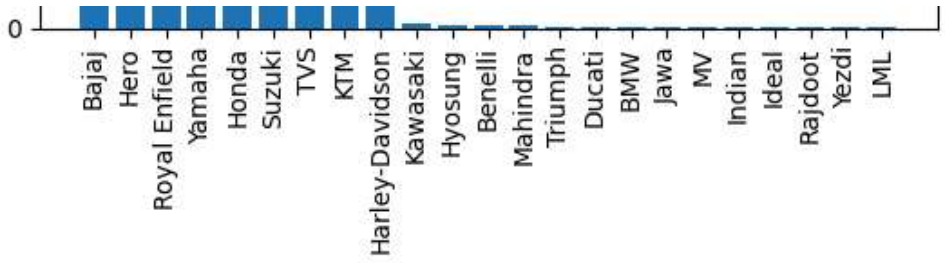
	bike_name	price	city	kms_driven	owner	age	power	brand
0	TVS Star City Plus Dual Tone 110cc	35000.0	Ahmedabad	17654.0	First Owner	3.0	110.0	TVS
1	Royal Enfield Classic 350cc	119900.0	Delhi	11000.0	First Owner	4.0	350.0	Royal Enfield
2	Triumph Daytona 675R	600000.0	Delhi	110.0	First Owner	8.0	675.0	Triumph
3	TVS Apache RTR 180cc	65000.0	Bangalore	16329.0	First Owner	4.0	180.0	TVS
4	Yamaha FZ S V 2.0 150cc-Ltd. Edition	80000.0	Bangalore	10000.0	First Owner	3.0	150.0	Yamaha

In [61]:
```
df['brand'].value_counts().keys()
```

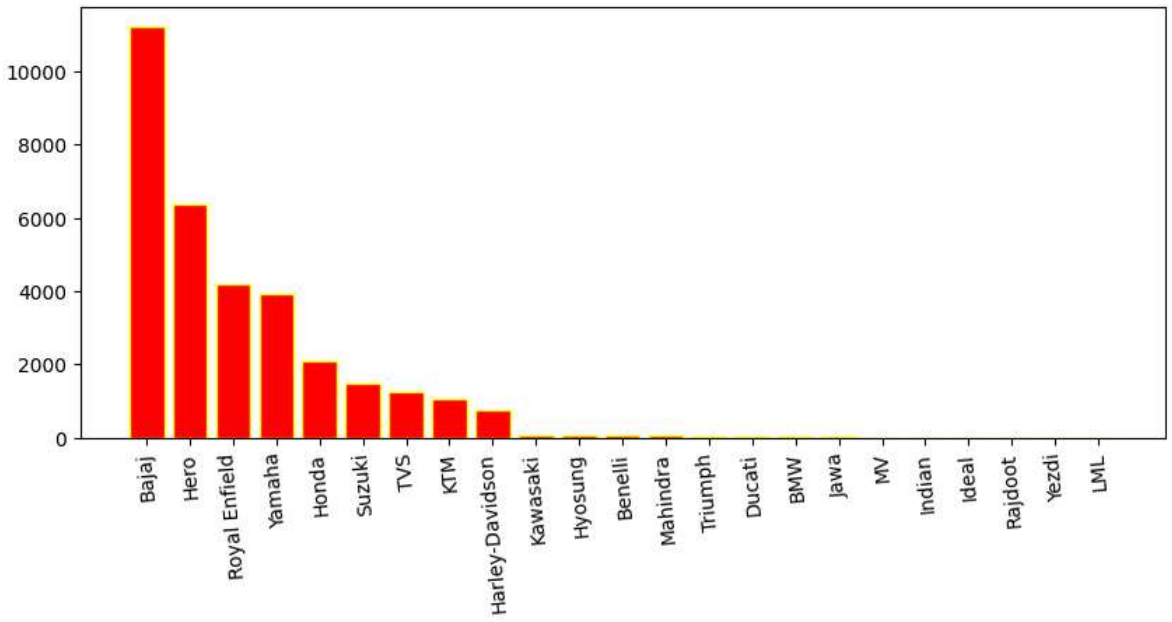
Out[61]:
```
Index(['Bajaj', 'Hero', 'Royal Enfield', 'Yamaha', 'Honda', 'Suzuki', 'TVS',
       'KTM', 'Harley-Davidson', 'Kawasaki', 'Hyosung', 'Benelli', 'Mahindra',
       'Triumph', 'Ducati', 'BMW', 'Jawa', 'MV', 'Indian', 'Ideal', 'Rajdoot',
       'Yezdi', 'LML'],
      dtype='object', name='brand')
```

In [62]:
```
no_of_count = list(df['brand'].value_counts().values)
brand_name = list(df['brand'].value_counts().keys())
plt.bar(brand_name,no_of_count)
plt.xticks(rotation = 90)
plt.show()
```





In [63]:
```
plt.figure(figsize = (10,4))     # (width , height)
brand_name = list(df['brand'].value_counts().keys())
no_of_count = list(df['brand'].value_counts().values)
plt.bar(brand_name,no_of_count,color = 'red',ec = 'yellow')
plt.xticks(rotation = 95)
plt.savefig('brandpopulation')
plt.show()
```



Histogram

A histogram is a type of bar chart that displays the distribution of a dataset by grouping data into bins (or intervals) and showing the frequency (count) of data points within each bin. It is commonly used to visualize the distribution of numerical data and to identify patterns such as skewness, modality, and spread.

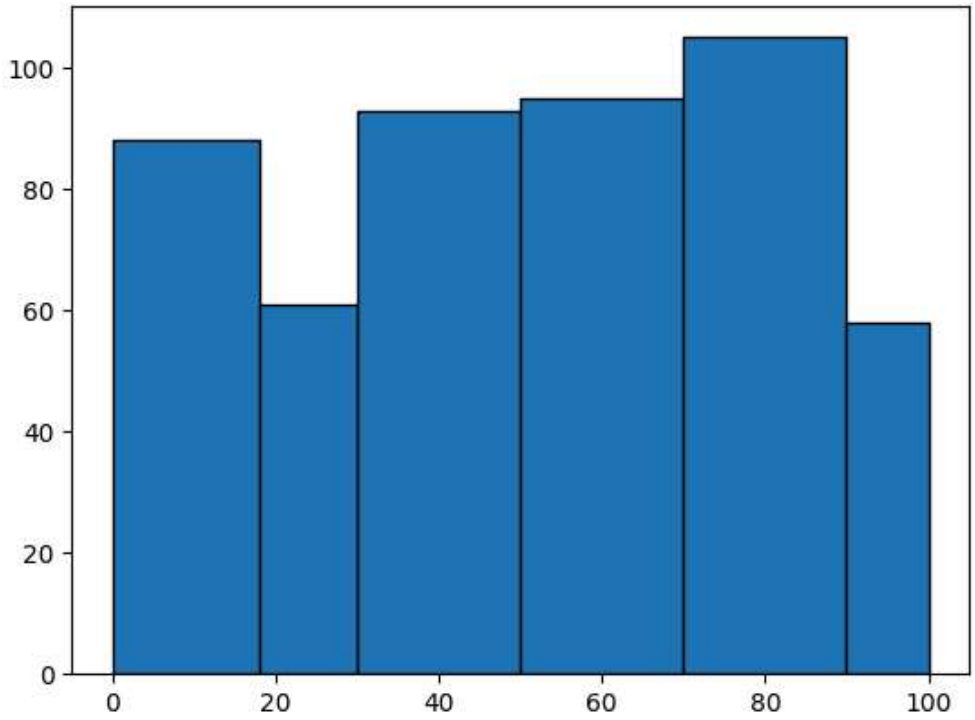
In [64]:
```
age = np.random.randint(1,100,500)
age
```

Out[64]:
```
array([81, 41, 65, 76, 34, 38, 78, 41, 43, 41, 24,  5,  7, 66, 29, 16, 45,
       20, 44, 53, 51, 31, 83, 63, 78, 44, 40, 34, 81, 16,  1, 35, 53, 47,
       27, 85, 78,  5, 80,  9, 42, 50, 18,  3, 84, 74, 61, 31, 78, 27, 73,
       81, 86, 82, 91,  1, 68,  9, 64, 51, 26, 74, 15, 68, 28, 31,  8, 74,
       55, 87, 70, 65, 98, 40, 16, 77, 60, 27, 22, 23, 61,  8, 41, 90,  5,
       27, 64, 53, 28, 90, 84, 28, 63, 66, 49,  9, 76, 54, 61, 13, 81, 89,
       55, 45, 94, 22, 28,  1, 92, 23, 45, 12, 49,  5, 99, 10, 58, 88, 35,
       72, 97, 75, 45, 45, 82, 29,  7, 53, 92, 12, 87,  7, 64, 96, 43, 40,
       71, 91, 82, 87, 13, 57, 55, 52, 96, 38, 31,  1, 78, 49, 74, 16, 88,
       84, 92, 22, 59, 85, 21, 59, 37,  1, 88, 41,  1, 17, 79,  7, 10, 25,
```

```
       40, 99, 31, 23, 47, 71,  8, 85, 54, 94, 23, 25, 13, 40, 69, 84, 12,
       40, 92, 27, 16, 35, 71, 16, 52, 19, 66,  7, 98, 57,  7,  1, 14, 75,
       75, 64,  8, 26, 36, 48, 90, 70, 74, 61, 68, 28, 71,  3, 17, 38,  3,
        9, 92, 77, 78, 50, 44,  7, 12, 25, 40, 72, 54, 80, 69, 24, 43, 36,
       46, 81, 78, 12, 41, 23, 64, 76, 90, 59, 49, 45, 38, 73, 46,  8, 93,
       34, 47, 60, 71, 57, 75, 97, 15, 50, 92, 85, 98, 31,  1, 40,  7, 13,
       57, 98, 58, 21, 81, 66, 75, 35,  9,  7, 38, 66, 40, 95, 71, 23, 44,
       44, 10, 13, 87, 49, 16, 57, 84, 18, 20, 67, 16, 92, 94, 78,  4, 36,
        8,  1, 17, 57, 38, 91,  7, 87, 28, 48, 55, 44, 81, 34, 51, 49, 56,
       94, 37, 49, 95, 96, 29, 19,  4, 85, 18, 99, 40, 38, 42,  2, 87, 15,
       70, 86, 78, 35, 75, 56, 11, 42, 88, 23, 82, 18, 80, 31, 94, 91, 34,
       15, 75, 63, 31, 94, 13, 86, 59, 86, 60, 84, 21,  4, 70, 54, 28, 85,
       27,  5, 24, 73, 58,  6, 33, 32, 59, 83, 84, 82, 57, 96, 90, 24, 64,
       79, 69, 52, 27, 73, 77, 92, 53, 32, 98, 64, 93,  8, 24, 54, 10, 99,
       98, 10, 38,  6, 19, 65, 30, 71, 69, 93, 93, 38, 44, 16, 20, 34, 25,
       16, 56, 27,  2, 81,  1, 93, 24, 63, 74, 58, 33, 95, 94, 19, 54, 69,
       94, 73, 18, 26, 36, 78, 68, 91, 54, 54, 39, 10, 60, 74, 27, 58, 87,
       27, 64, 81, 90, 78, 90, 70, 99,  6, 69, 60, 96, 69, 69, 63, 40, 77,
       99, 15, 74, 85, 82, 63, 20, 69, 82, 55, 40, 94, 16, 12, 35, 63, 68,
       24, 96, 54,  2, 44, 96, 24], dtype=int32)
```

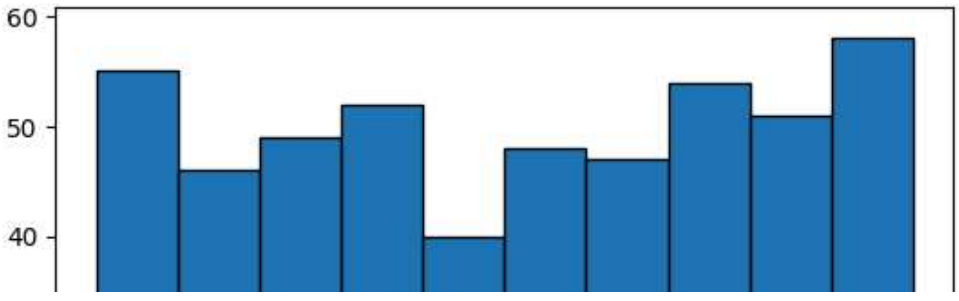
In [65]:

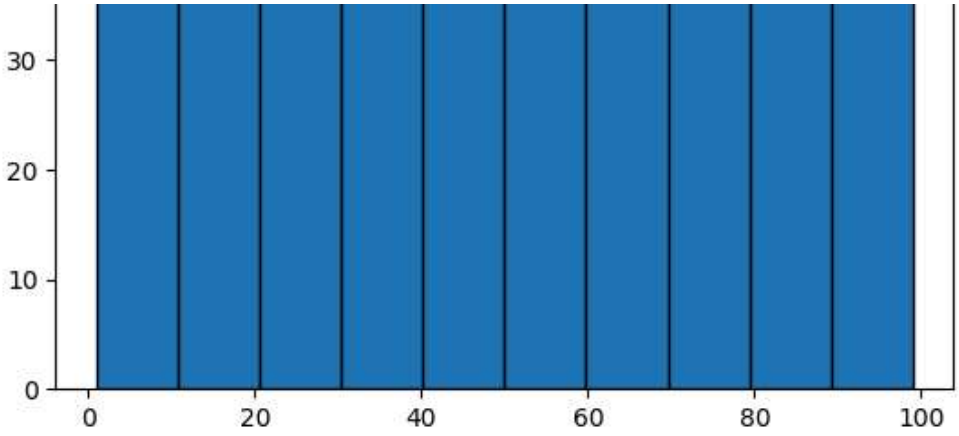
```
bin = [0,18,30,50,70,90,100]
plt.hist(age,bins=bin,ec='k')
plt.show()
```



In [66]:

```
#bin = [0,18,30,50,70,90,100]
plt.hist(age,ec='k')
plt.show()
```



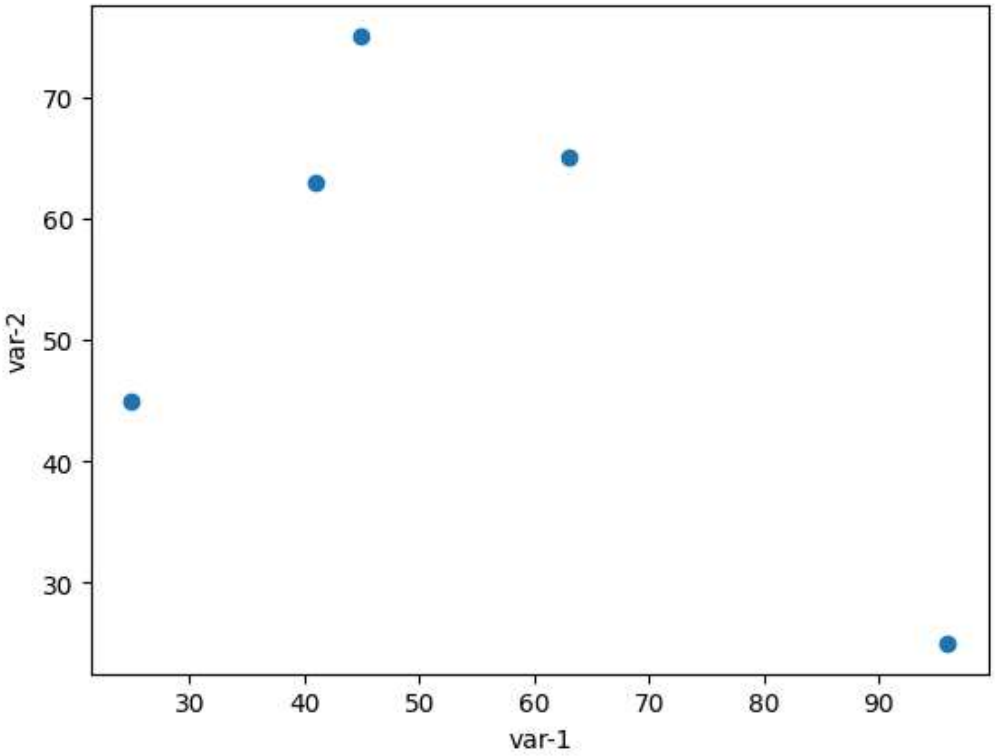


Scatter Plot:-

A scatter plot is a type of data visualization that uses dots to represent the values of two different variables. Each dot on the plot represents an individual data point, with its position determined by the values of the two variables being plotted on the x-axis and y-axis.

In [67]:
```
var1 = [25,41,63,96,45]
var2 = [45,63,65,25,75]

plt.scatter(var1,var2)
plt.xlabel("var-1")
plt.ylabel("var-2")
plt.show()
```



In [68]:
```
df = pd.read_csv('Used_Bikes.csv')
df.head()
```

Out[68]:

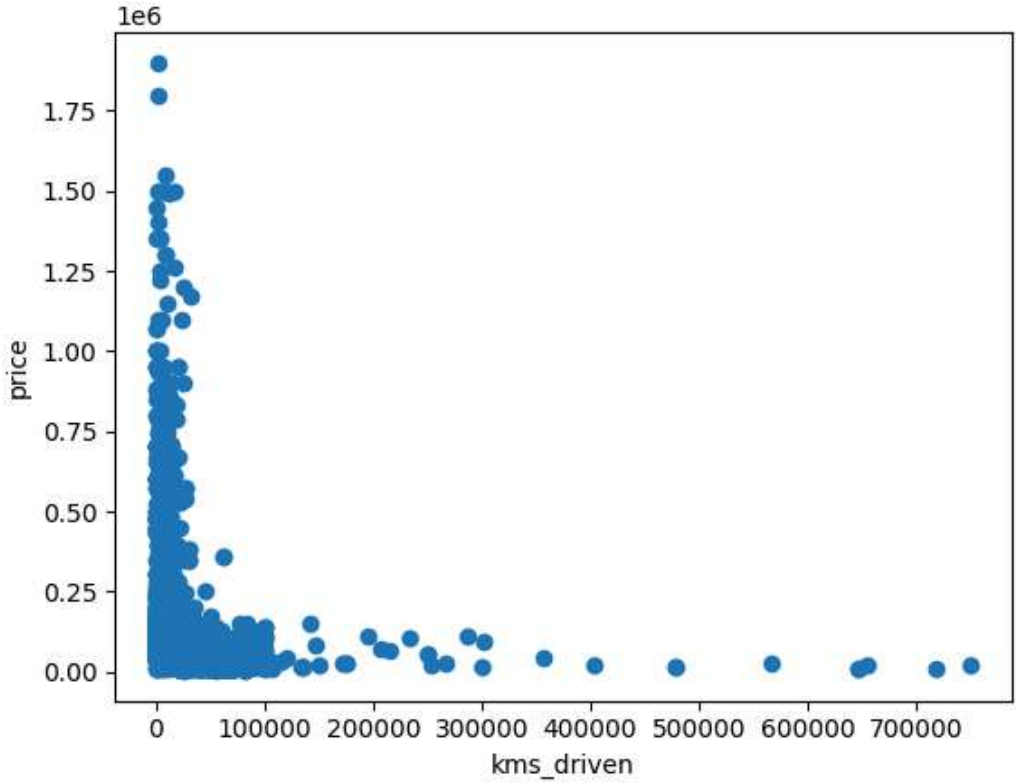
	bike_name	price	city	kms_driven	owner	age	power	brand

Out[68]:

	bike_name	price	city	kms_driven	owner	age	power	brand
0	TVS Star City Plus Dual Tone 110cc	35000.0	Ahmedabad	17654.0	First Owner	3.0	110.0	TVS
1	Royal Enfield Classic 350cc	119900.0	Delhi	11000.0	First Owner	4.0	350.0	Royal Enfield
2	Triumph Daytona 675R	600000.0	Delhi	110.0	First Owner	8.0	675.0	Triumph
3	TVS Apache RTR 180cc	65000.0	Bangalore	16329.0	First Owner	4.0	180.0	TVS
4	Yamaha FZ S V 2.0 150cc-Ltd. Edition	80000.0	Bangalore	10000.0	First Owner	3.0	150.0	Yamaha

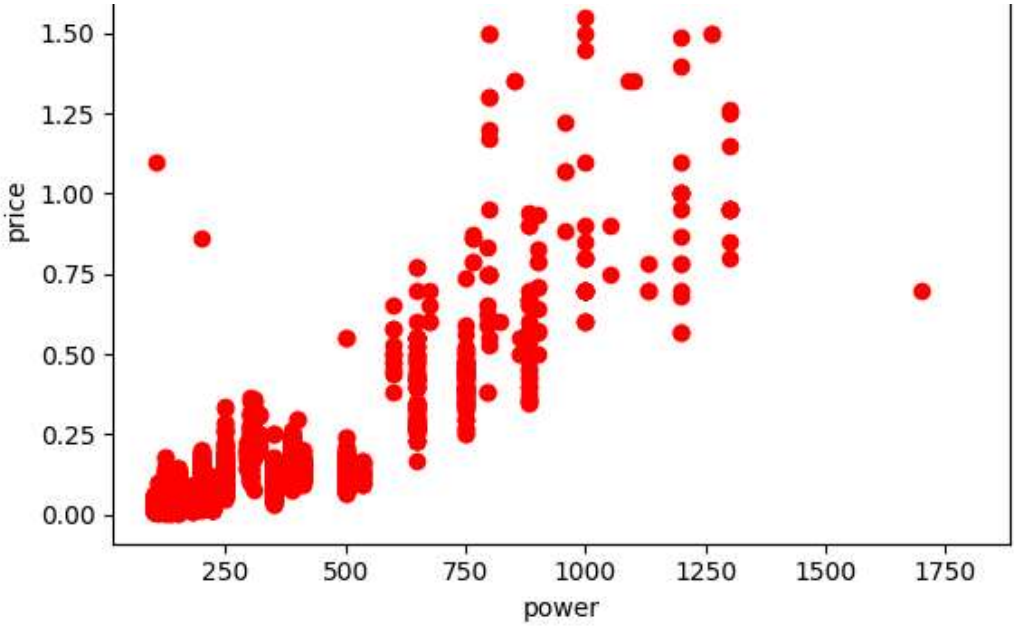
In [69]:
```
plt.scatter(df['kms_driven'],df['price'])
plt.xlabel("kms_driven")
plt.ylabel("price")
plt.show
```

Out[69]: <function matplotlib.pyplot.show(close=None, block=None)>



In [70]:
```
plt.scatter(df['power'],df['price'],color='red')
plt.xlabel("power")
plt.ylabel("price")
plt.show
```

Out[70]: <function matplotlib.pyplot.show(close=None, block=None)>

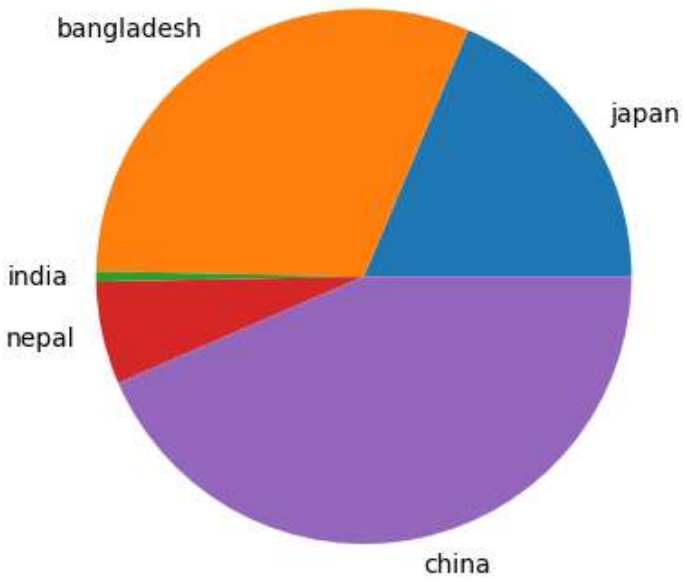


Pie Plot:-

A pie plot, also known as a pie chart, is a circular statistical graphic that is divided into slices to illustrate numerical proportions. Each slice represents a category's contribution to the whole, with the size of each slice corresponding to its proportion of the total.

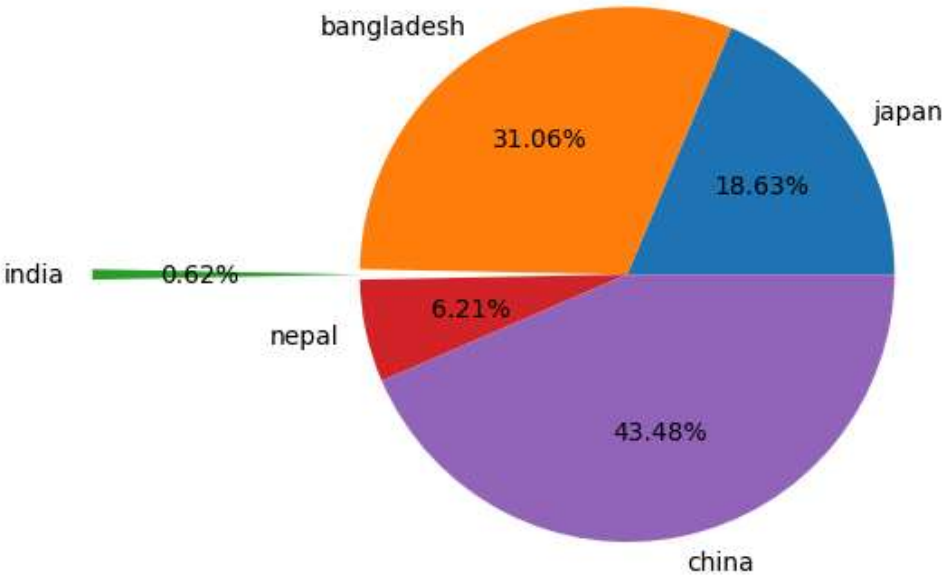
In [71]:
```
country_population = [300,500,10,100,700]
country_name = ['japan','bangladesh','india','nepal','china']

plt.pie(country_population,labels=country_name)
plt.show()
```



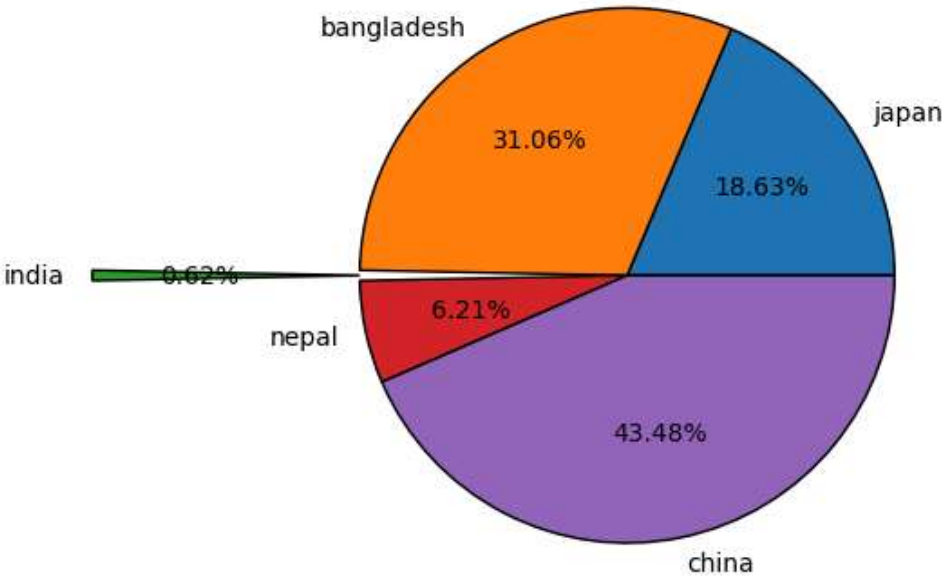
In [72]:
```
country_population = [300,500,10,100,700]
country_name = ['japan','bangladesh','india','nepal','china']
```

```
explosion = [0,0,1,0,0]
plt.pie(country_population,labels=country_name,autopct="%2.2f%%",explode = explosion)
plt.show()
```



In [73]:

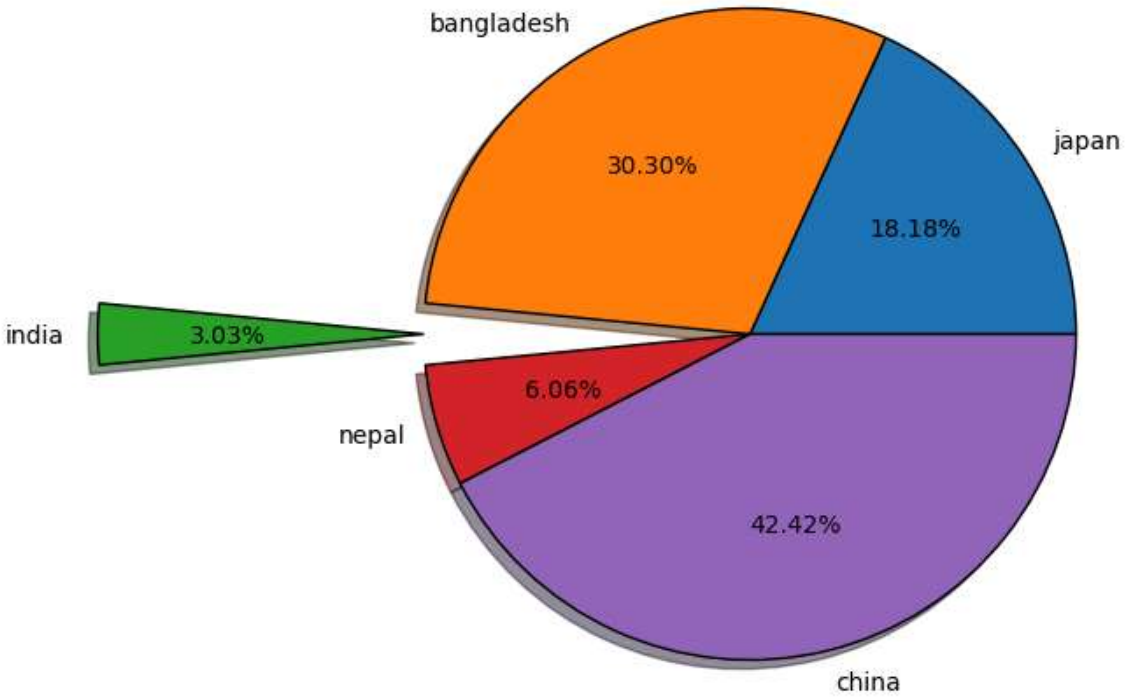
```
country_population = [300,500,10,100,700]
country_name = ['japan','bangladesh','india','nepal','china']
explosion = [0,0,1,0,0]
plt.pie(country_population,labels=country_name,autopct="%2.2f%%",explode = explosion,wedge
plt.show()
```



In [74]:

```
plt.figure(figsize=(6,8))
country_population = [300,500,50,100,700]
country_name = ['japan','bangladesh','india','nepal','china']
explosion = [0,0,1,0,0]
```

```
plt.pie(country_population,labels=country_name,autopct="%2.2f%%",explode = explosion,wedge
plt.show()
```



Box plot:-

A box plot (or box-and-whisker plot) is a statistical graphic used to display the distribution of a dataset through its quartiles. It visually summarizes the central tendency, variability, and potential outliers in the data.

The minimun

First Quartile(Q1)

Median(Q2)

Third Quartile(Q3)

The Maximum

It is used for identifying outliers

In [75]:

```
df = pd.read_csv("Used_Bikes.csv")
df.drop_duplicates(inplace = True)
ktm = df[df['brand'] == "KTM"]

# Creating a Box plot
plt.boxplot(ktm['price'])
plt.show()
```


800000
700000
600000
500000