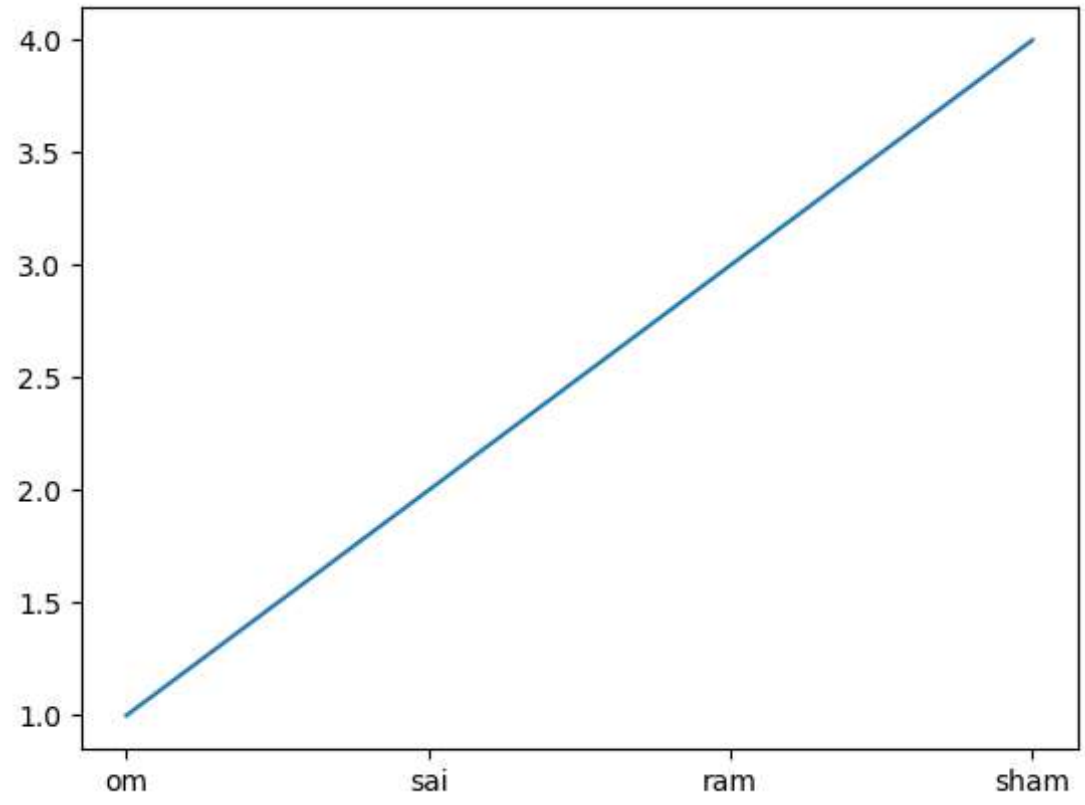
In [13]:

```
#slip2-q2-A
#Write a Python program for Handling Missing Value.
#Replace missing value of salary, age column with mean of that column.(Use Data.csv file).
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\data.csv");
data['age']=data['age'].fillna(data['age'].mean())
data['sal']=data['sal'].fillna(data['sal'].mean())
print(data)
```

```
   country        age           sal purchased
0   fraNCE  44.000000  72000.000000         n
1    spain  27.000000  48000.000000         y
2  germany  30.000000  54000.000000         n
3    spain  38.000000  61000.000000         n
4  germany  40.000000  57077.777778         y
5   fraNCE  35.000000  58000.000000         y
6    spain  38.777778  52000.000000         n
7   fraNCE  48.000000  79000.000000         y
8  germany  50.000000  83000.000000         n
9   fraNCE  37.000000   6700.000000         y
```

In [15]:

```
#slip2-q2-B
#Write a Python program to generate a line plot of name Vs salary
import numpy as np
name=np.array(['om','sai','ram','sham'])
sal=np.array([1,2,3,4])
plt.plot(name,sal)
plt.show()
```



In [24]:

```
#slip2-q2-C
#Download the heights and weights dataset and load the dataset froma given csv file into a
#Print the first, last 10 rows and random 20 rows also display shape of the dataset.
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\SOCR-HeightWeight.csv");
print("fiste 10 rows\n",data.head(10))
print("last 10 rows\n",data.tail(10))
print("random 20 rows\n",data.sample(20))
print("shape",data.shape)
```

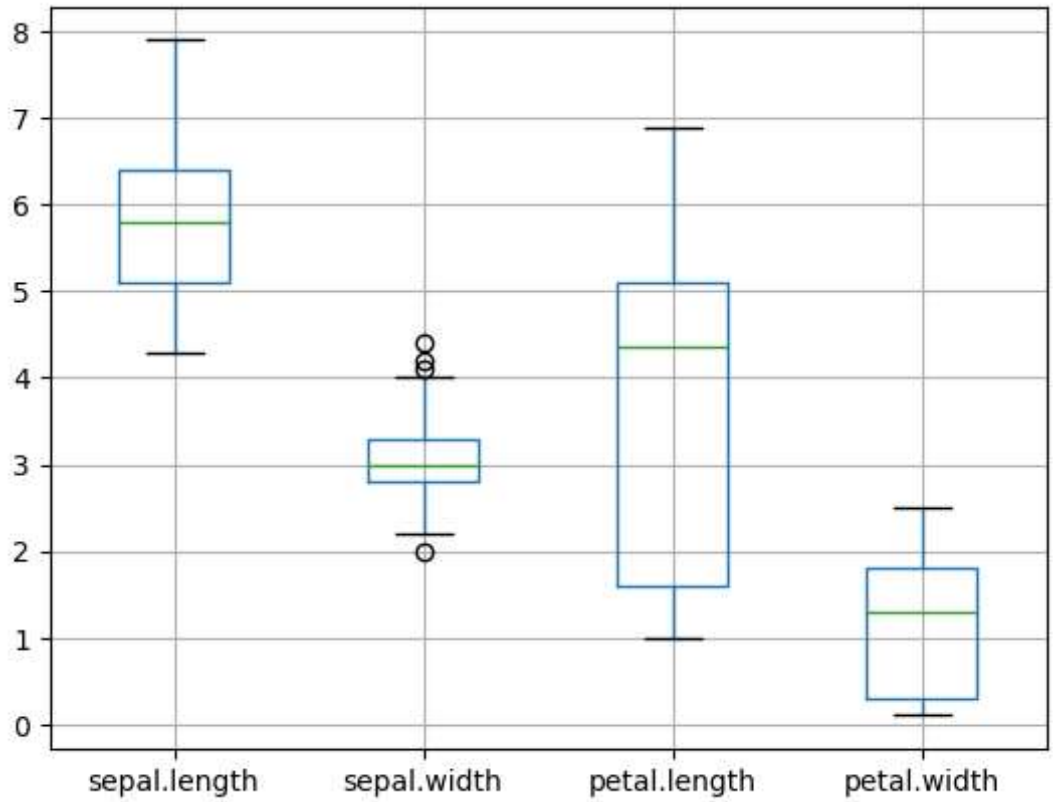
```
fiste 10 rows
    Index  Height(Inches)  Weight(Pounds)
0      1        65.78331        112.9925
1      2        71.51521        136.4873
2      3        69.39874        153.0269
3      4        68.21660        142.3354
4      5        67.78781        144.2971
5      6        68.69784        123.3024
6      7        69.80204        141.4947
7      8        70.01472        136.4623
8      9        67.90265        112.3723
9     10        66.78236        120.6672
last 10 rows
        Index  Height(Inches)  Weight(Pounds)
24990  24991        69.97767        125.3672
24991  24992        71.91656        128.2840
24992  24993        70.96218        146.1936
24993  24994        66.19462        118.7974
24994  24995        67.21126        127.6603
24995  24996        69.50215        118.0312
24996  24997        64.54826        120.1932
24997  24998        64.69855        118.2655
24998  24999        67.52918        132.2682
24999  25000        68.87761        124.8742
random 20 rows
        Index  Height(Inches)  Weight(Pounds)
23832  23833        68.51045       113.11810
9396    9397        68.59889       127.60630
1089    1090        65.34461       122.97310
18264  18265        69.11084       125.01660
15838  15839        66.67182       107.10520
14141  14142        65.33009       112.19830
10352  10353        65.11686       117.04720
9181    9182        69.17193       140.65910
10480  10481        68.80266       138.00140
10025  10026        67.97403       115.52810
24748  24749        63.33113        95.95417
7740    7741        65.18871       115.90000
4997    4998        65.88747       112.31780
2334    2335        70.86149       151.27880
1499    1500        69.57300       138.64610
16839  16840        67.31179       137.49550
7473    7474        70.58047       122.89970
16083  16084        69.65587       127.98940
22059  22060        67.07765       126.04820
22418  22419        64.19111       131.50280
shape (25000, 3)
```

In [53]:

```
#slip3-q2-A
#Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepa
#Petal Width are distributed across the three species. (Use iris.csv dataset)
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\iris.csv");
newdata=data[["sepal.length","sepal.width","petal.length","petal.width"]]
newdata.boxplot()
```

Out[53]:

<AxesSubplot:>



In [26]:

```
#slip3-q2-B
#Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepa
#Petal Width are distributed across the three species. (Use iris.csv dataset)
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\SOCR-HeightWeight.csv");
data.describe()
```

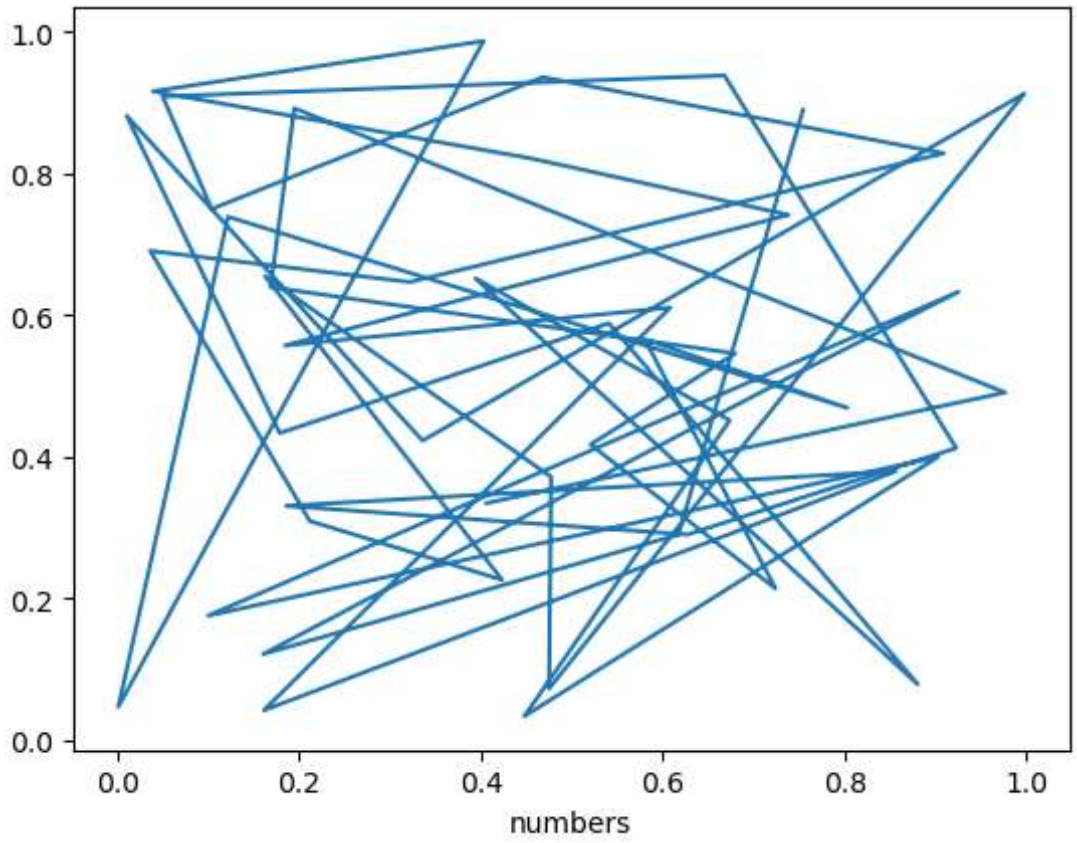
Out[26]:

	Index	Height(Inches)	Weight(Pounds)
count	25000.000000	25000.000000	25000.000000
mean	12500.500000	67.993114	127.079421
std	7217.022701	1.901679	11.660898
min	1.000000	60.278360	78.014760
25%	6250.750000	66.704397	119.308675
50%	12500.500000	67.995700	127.157750
75%	18750.250000	69.272958	134.892850
max	25000.000000	75.152800	170.924000

In [52]:

```
#slip4-q2-A
#Generate a random array of 50 integers and display them using a line chart, scatter plot,
#histogram and box plot. Apply appropriate color, labels and styling options.
import numpy as np
import matplotlib.pyplot as plt
from numpy import random
x=np.random.rand(50)
y=np.random.rand(50)
plt.xlabel("numbers")
print(plt.plot(x,y))
```

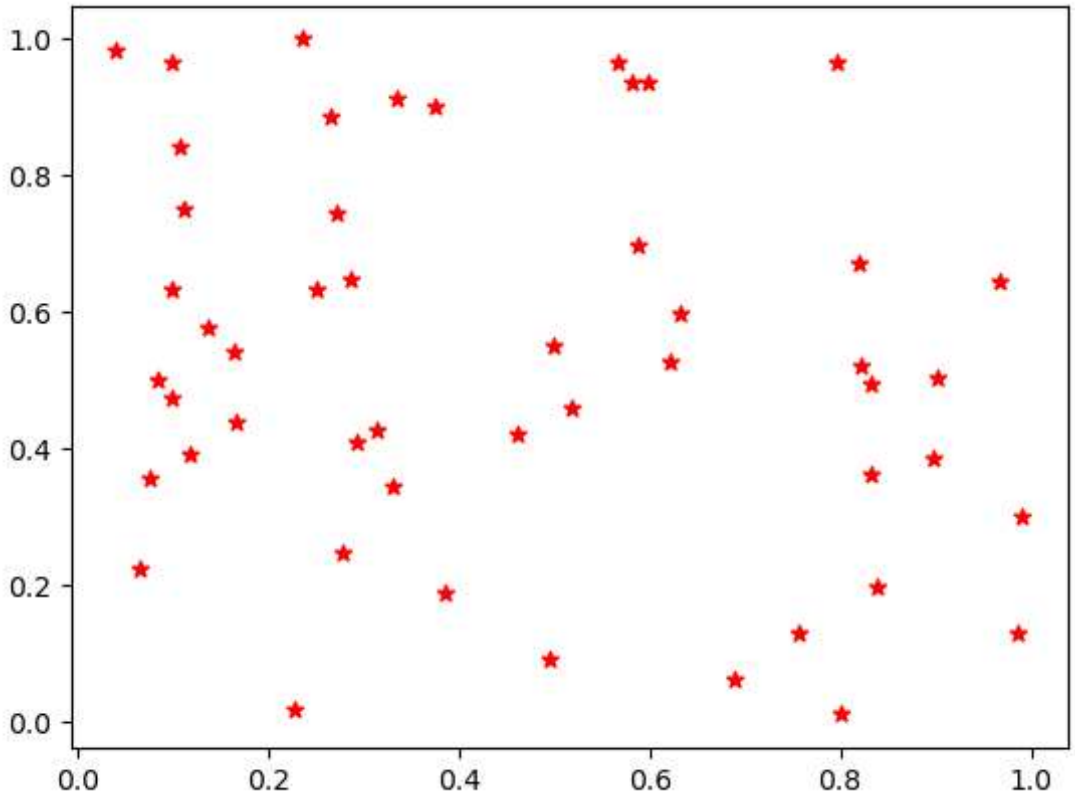
```
[<matplotlib.lines.Line2D object at 0x000001D1ED0E4C70>]
```



In [50]:

```
print(plt.scatter(x,y,marker="*",label='normal',color="r"))
```

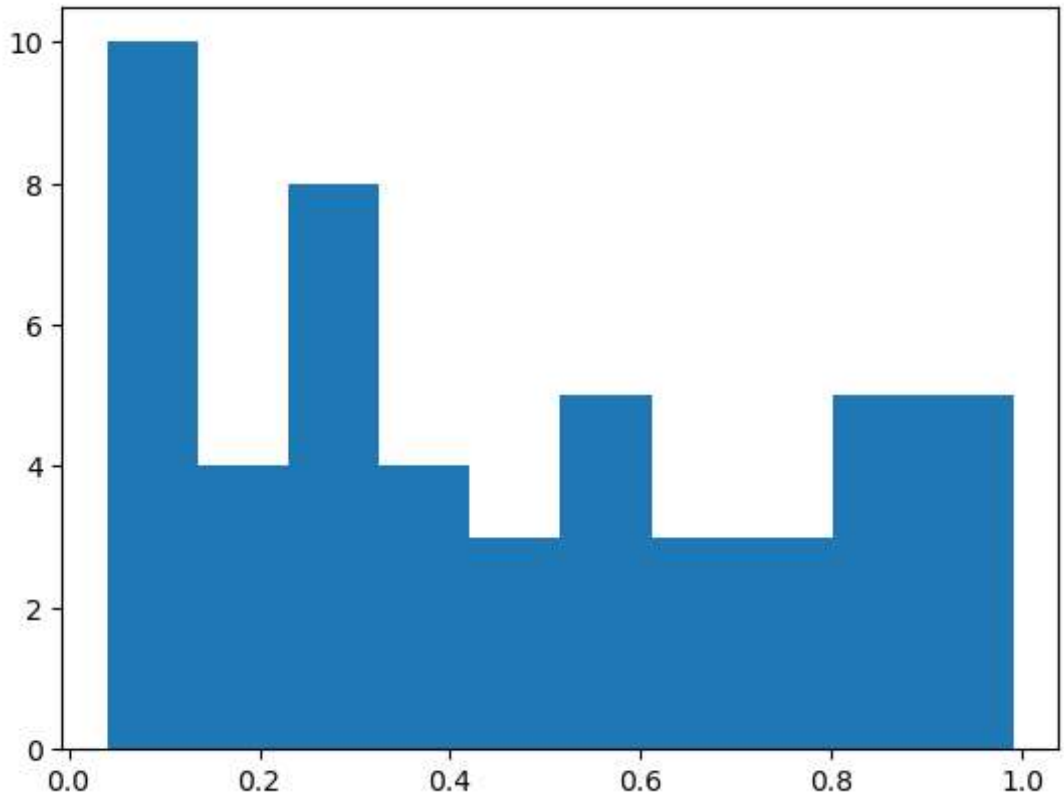
```
<matplotlib.collections.PathCollection object at 0x000001D1ECFB6400>
```



In [44]:

```
print(plt.hist(x))
```

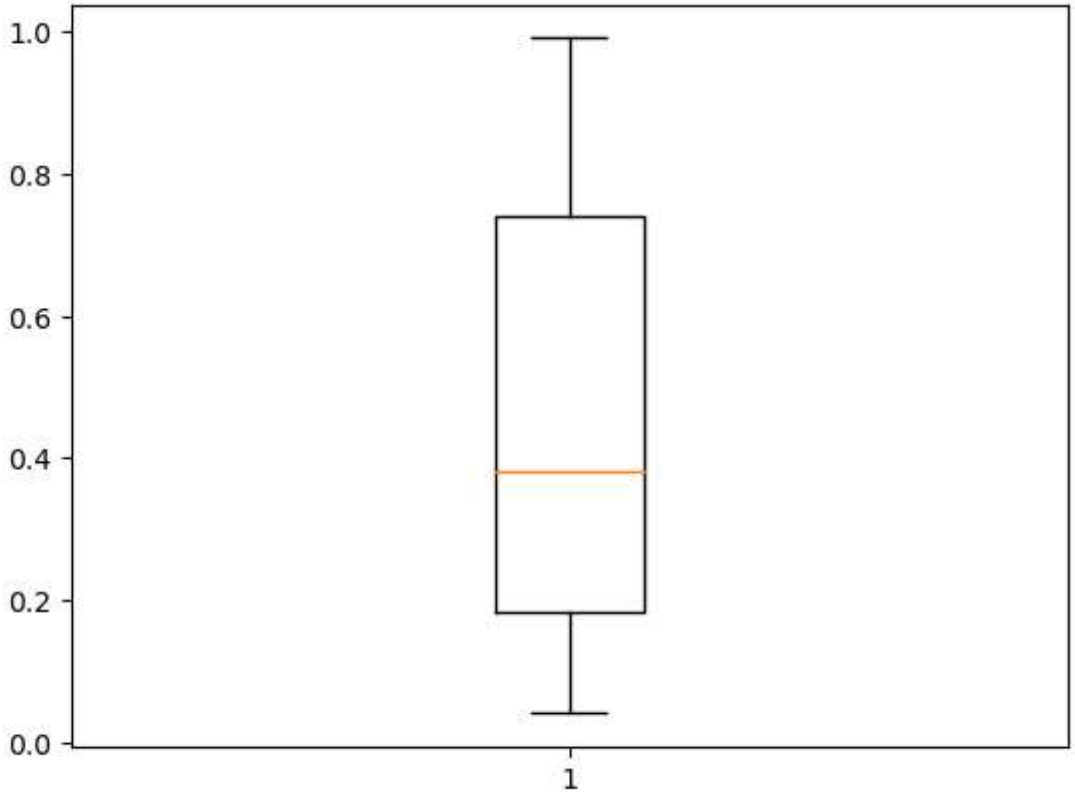
```
(array([10.,  4.,  8.,  4.,  3.,  5.,  3.,  3.,  5.,  5.]), array([0.0407915
2, 0.13582347, 0.23085542, 0.32588737, 0.42091931,
       0.51595126, 0.61098321, 0.70601516, 0.8010471 , 0.89607905,
       0.991111  ]), <BarContainer object of 10 artists>)
```



In [47]:

```
print(plt.boxplot(x))
```

```
{'whiskers': [<matplotlib.lines.Line2D object at 0x000001D1EE996850>, <matpl
otlib.lines.Line2D object at 0x000001D1EE996B20>], 'caps': [<matplotlib.line
s.Line2D object at 0x000001D1EE996DF0>, <matplotlib.lines.Line2D object at 0
x000001D1EE9A3100>], 'boxes': [<matplotlib.lines.Line2D object at 0x000001D1
EE996580>], 'medians': [<matplotlib.lines.Line2D object at 0x000001D1EE9A33D
0>], 'fliers': [<matplotlib.lines.Line2D object at 0x000001D1EE9A36A0>], 'me
ans': []}
```



In [4]:

```
#Write a Python program to print the shape, number of rows-columns, data types,
#feature names and the description of the data(Use User_Data.csv)
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\data.csv");
print(data.shape)
print(data.columns)
print(data.dtypes)
print(data.describe)
data.info()
```

```
(10, 4)
Index(['country', 'age', 'sal', 'purchased'], dtype='object')
country       object
age          float64
sal          float64
purchased     object
dtype: object
<bound method NDFrame.describe of    country   age      sal purchased
0   fraNCE  44.0  72000.0         n
1    spain  27.0  48000.0         y
2  germany  30.0  54000.0         n
3    spain  38.0  61000.0         n
4  germany  40.0      NaN         y
5   fraNCE  35.0  58000.0         y
6    spain   NaN  52000.0         n
7   fraNCE  48.0  79000.0         y
8  germany  50.0  83000.0         n
9   fraNCE  37.0   6700.0         y>
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   country    10 non-null     object
 1   age        9 non-null      float64
 2   sal        9 non-null      float64
 3   purchased  10 non-null     object
dtypes: float64(2), object(2)
memory usage: 448.0+ bytes
```

In [13]:

```
#Write a Python program to perform the following tasks :
#a. Apply OneHot coding on Country column.
#b. Apply Label encoding on purchased column
#(Data.csv have two categorical column the country column, and the purchased column).
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\data.csv");
enc=OneHotEncoder(handle_unknown='ignore')
enc_data=pd.DataFrame(enc.fit_transform(data[['country']]).toarray())
print("OneHot coding on Country column")
print(enc_data)
labelEncoder=LabelEncoder()
data['purchased']=labelEncoder.fit_transform(data['purchased'])
print("Label encoding on purchased column")
print(data)
```

```
OneHot coding on Country column
     0    1    2
0  1.0  0.0  0.0
1  0.0  0.0  1.0
2  0.0  1.0  0.0
3  0.0  0.0  1.0
4  0.0  1.0  0.0
5  1.0  0.0  0.0
6  0.0  0.0  1.0
7  1.0  0.0  0.0
8  0.0  1.0  0.0
9  1.0  0.0  0.0
Label encoding on purchased column
   country   age      sal  purchased
0   fraNCE  44.0  72000.0          0
1    spain  27.0  48000.0          1
2  germany  30.0  54000.0          0
3    spain  38.0  61000.0          0
4  germany  40.0      NaN          1
5   fraNCE  35.0  58000.0          1
6    spain   NaN  52000.0          0
7   fraNCE  48.0  79000.0          1
8  germany  50.0  83000.0          0
9   fraNCE  37.0   6700.0          1
```

In [ ]:

```
#Write a program in python to perform following task :
#Standardizing Data (transform them into a standard Gaussian distribution with a mean of
#0 and a standard deviation of 1) (Use winequality-red.csv)
```

In [21]:

```
#Write a python program to Display column-wise mean, and median for SOCR-HeightWeight datas
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\SOCR-HeightWeight.csv");
print("height mean",np.mean(data['Height(Inches)']))
print("weight mean",np.mean(data['Weight(Pounds)']))
print("height median",np.median(data['Height(Inches)']))
print("weight median",np.median(data['Weight(Pounds)']))
```

```
height mean 67.99311359679979
weight mean 127.07942116079916
height median 67.9957
weight median 127.15775
```

In [22]:

```
#Write a python program to compute sum of Manhattan distance between all pairs of points.
import pandas as pd
from scipy.spatial.distance import cityblock
d1=pd.DataFrame({'a':[1,0,1],
                 'b':[1,8,1],
                 'c':[2,0,1],})
print("mahattam dist bet a& b is",cityblock(d1.a ,d1.b))
print("mahattam dist bet a& c is",cityblock(d1.a ,d1.c))
```

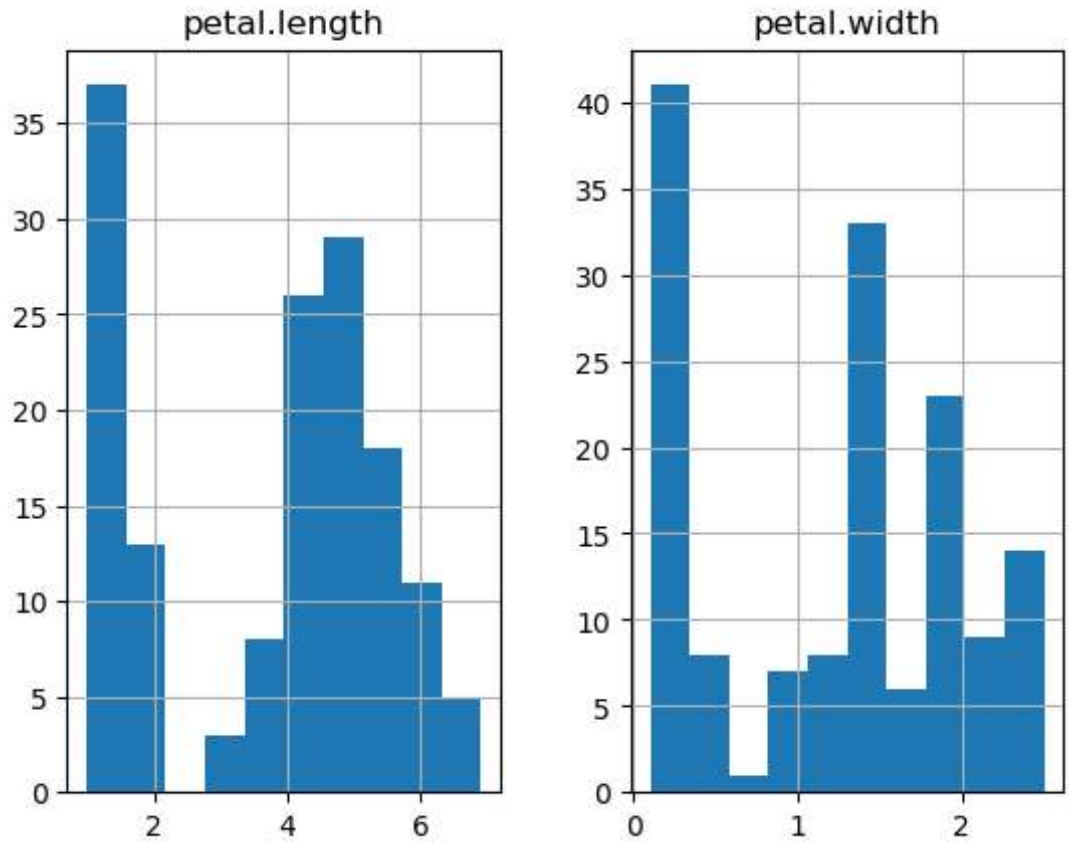
```
mahattam dist bet a& b is 8
mahattam dist bet a& c is 1
```

In [26]:

```
#Write a Python program to create a graph to find relationship between the petal length and
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\iris.csv");
newdata=data[["petal.length","petal.width"]]
newdata.hist()
```

Out[26]:

```
array([[<AxesSubplot:title={'center':'petal.length'}>,
        <AxesSubplot:title={'center':'petal.width'}>]], dtype=object)
```



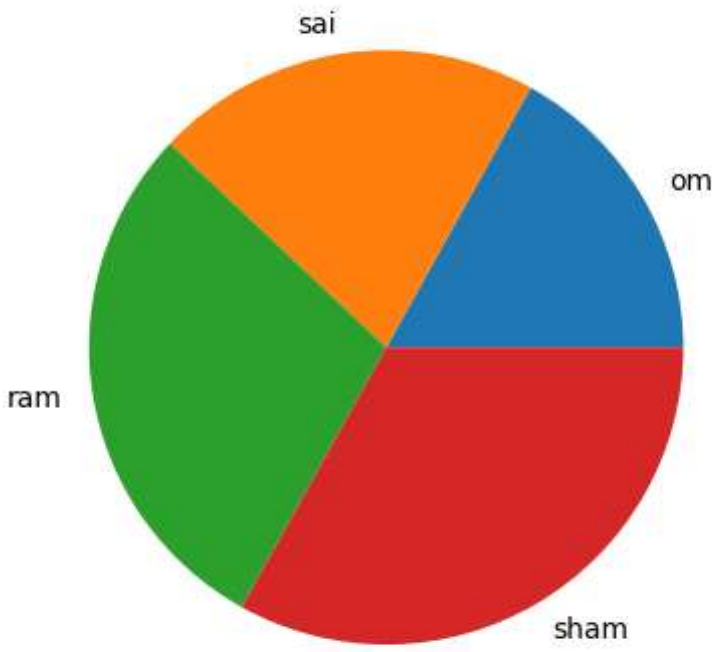
In [28]:

```
#Write a Python program to find the maximum and minimum value of a given flattened array.
import numpy as np
array=np.array([[12,34],[87,89]])
print("max=",np.max(array))
print("min=",np.min(array))
```

```
max= 89
min= 12
```

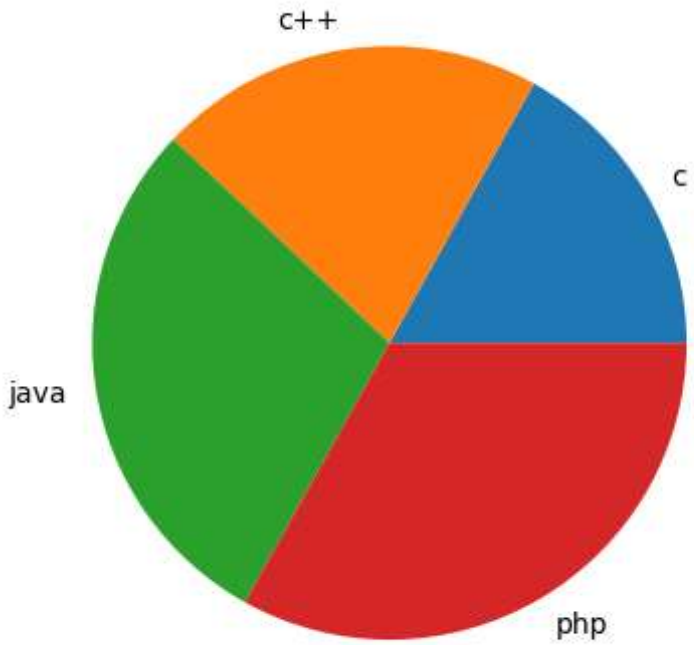
In [29]:

```
#Create two lists, one representing subject names and the other representing marks obtained
#Display the data in a pie chart
import numpy as np
import matplotlib.pyplot as plt
name=['om','sai','ram','sham']
marks=[45,56,77,88]
plt.pie(marks, labels=name)
plt.show()
```

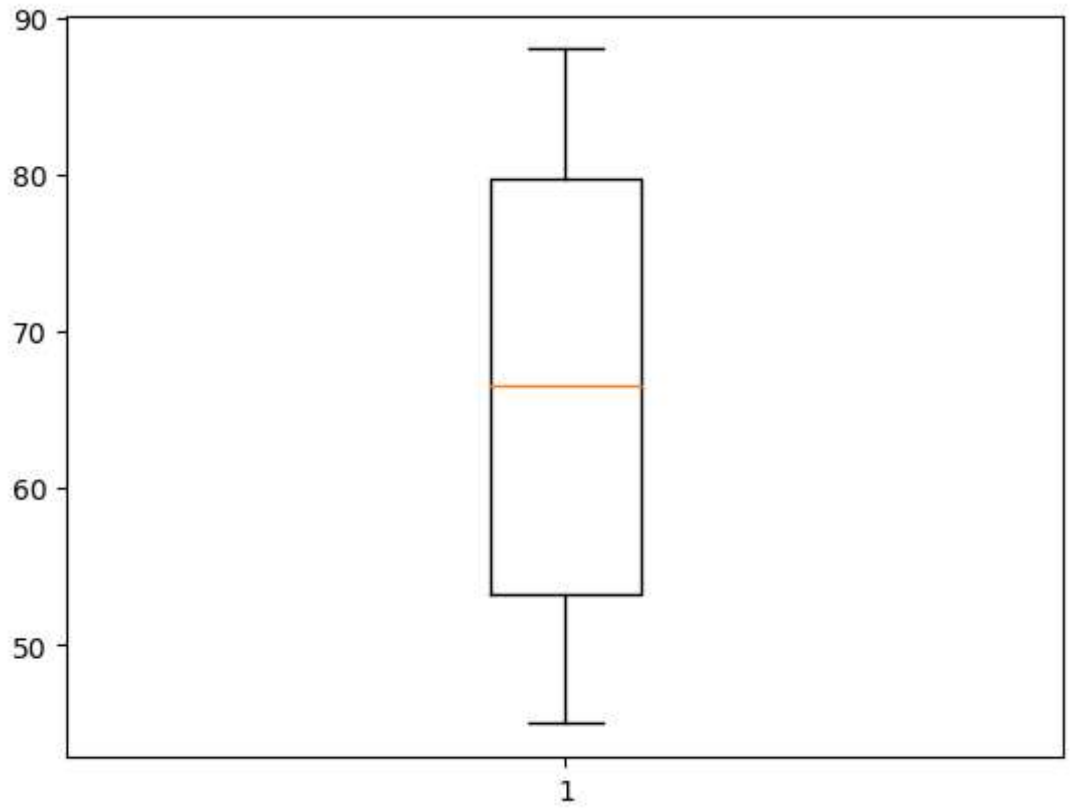


In [30]:

```
#Write a python program to create two lists, one representing subject names and the other r
#those subjects. Display the data in a pie chart and bar chart.
import numpy as np
import matplotlib.pyplot as plt
name=['c','c++','java','php']
marks=[45,56,77,88]
plt.pie(marks, labels=name)
plt.show()
print(plt.boxplot(marks))
```




```
{'whiskers': [<matplotlib.lines.Line2D object at 0x00000246D88AE820>, <matpl
otlib.lines.Line2D object at 0x00000246D88AEAF0>], 'caps': [<matplotlib.line
s.Line2D object at 0x00000246D88AEDC0>, <matplotlib.lines.Line2D object at 0
x00000246D88C20D0>], 'boxes': [<matplotlib.lines.Line2D object at 0x00000246
D88AE550>], 'medians': [<matplotlib.lines.Line2D object at 0x00000246D88C23A
0>], 'fliers': [<matplotlib.lines.Line2D object at 0x00000246D88C2670>], 'me
ans': []}
```



In [31]:

```
#Write a python program to create a data frame for students' information such as name, grad
#Display average age of students, average of graduation percentage.\
import pandas as pd
import numpy as np
df=pd.DataFrame({'name':['om','sai','ram'],
                 'age':[21,21,22],
                 'per':[45,67,98]})
print(df)
print("avrage age=",df['age'].mean())
print("avrage per=",df['per'].mean())
```

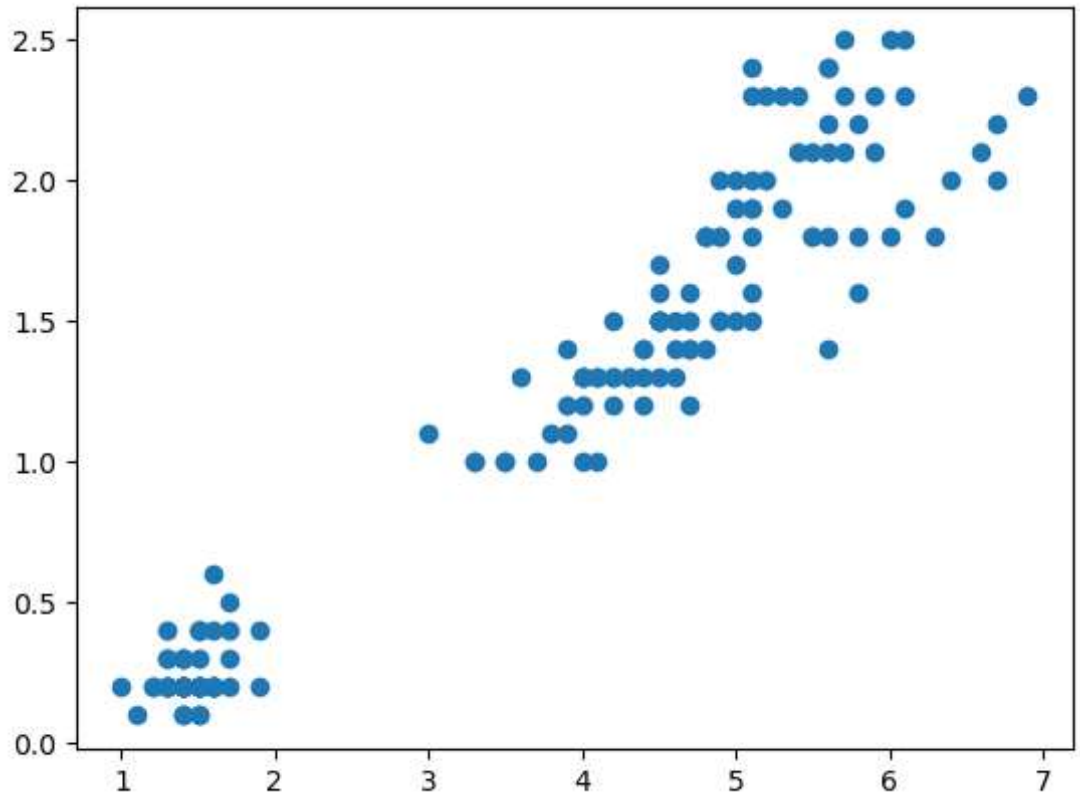
```
  name  age  per
0   om   21   45
1  sai   21   67
2  ram   22   98
avrage age= 21.333333333333332
avrage per= 70.0
```

In [34]:

```
#Write a Python program to draw scatter plots to compare two features of the iris dataset
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\iris.csv");
plt.scatter(data[["petal.length"]],data[["petal.width"]])
```

Out[34]:

```
<matplotlib.collections.PathCollection at 0x246d88d5d60>
```



In [35]:

```
#Write a Python program to create a data frame containing columns name, age , salary, depar
#Add 10 rows to the data frame. View the data frame.
import pandas as pd
dt=pd.DataFrame(columns=['name','age','sal','dept'])
dt.loc[0]=['om',21,89333,"comp"]
dt.loc[1]=['om',21,89333,'sci']
dt.loc[2]=['om',21,8932,'math']
dt.loc[3]=['om',21,8912,'comp']
dt.loc[4]=['om',21,8921,'math']
dt.loc[5]=['om',21,8232,'commerce']
dt.loc[6]=['om',21,89323,'ghg']
dt.loc[7]=['om',21,892,'com']
dt.loc[8]=['om',21,89897,'arts']
dt.loc[9]=['om',21,89,'dhg']
dt.loc[10]=['om',21,89,'hhj']
dt
```

Out[35]:

	name	age	sal	dept
0	om	21	89333	comp
1	om	21	89333	sci
2	om	21	8932	math
3	om	21	8912	comp
4	om	21	8921	math
5	om	21	8232	commerce
6	om	21	89323	ghg
7	om	21	892	com
8	om	21	89897	arts
9	om	21	89	dhg
10	om	21	89	hhj

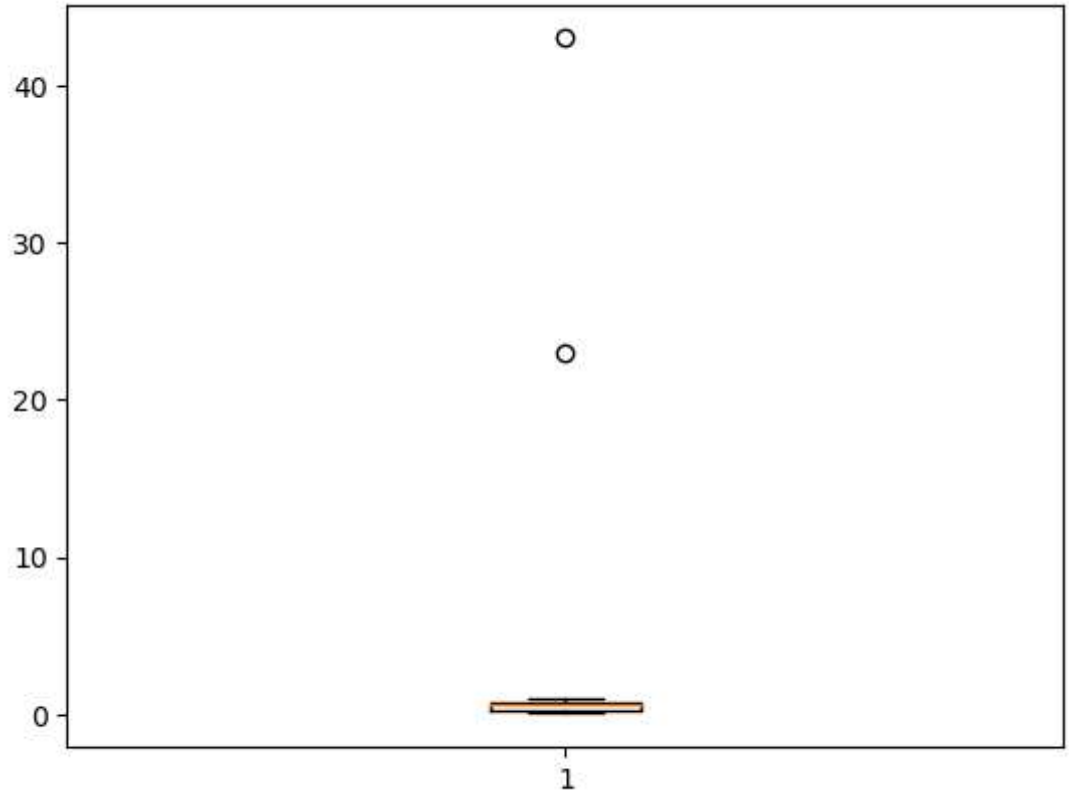
In [36]:

```
#Use the heights and weights dataset and load the dataset from a given csv file into a data
#Print the first, last 5 rows and random 10 row
import pandas as pd
import numpy as np
data=pd.read_csv("Documents\SOCR-HeightWeight.csv");
print(data.head(5))
print(data.tail(5))
print(data.sample(10))
```

```
   Index  Height(Inches)  Weight(Pounds)
0      1        65.78331        112.9925
1      2        71.51521        136.4873
2      3        69.39874        153.0269
3      4        68.21660        142.3354
4      5        67.78781        144.2971
       Index  Height(Inches)  Weight(Pounds)
24995  24996        69.50215        118.0312
24996  24997        64.54826        120.1932
24997  24998        64.69855        118.2655
24998  24999        67.52918        132.2682
24999  25000        68.87761        124.8742
       Index  Height(Inches)  Weight(Pounds)
1897    1898        65.19685        135.1309
23612  23613        67.18784        110.9939
12318  12319        68.51827        128.0545
11286  11287        67.58689        118.6697
12030  12031        67.02131        120.5036
5363    5364        70.07833        132.5733
7130    7131        71.77010        143.3808
2516    2517        67.80728        121.4194
5305    5306        67.06250        126.7332
21743  21744        66.63163        127.9701
```

In [37]:

```
#Add two outliers to the above data and display the box plot.
x=np.random.rand(50)
x1=np.append(x,[23,43])
plt.boxplot(x1)
plt.show()
```

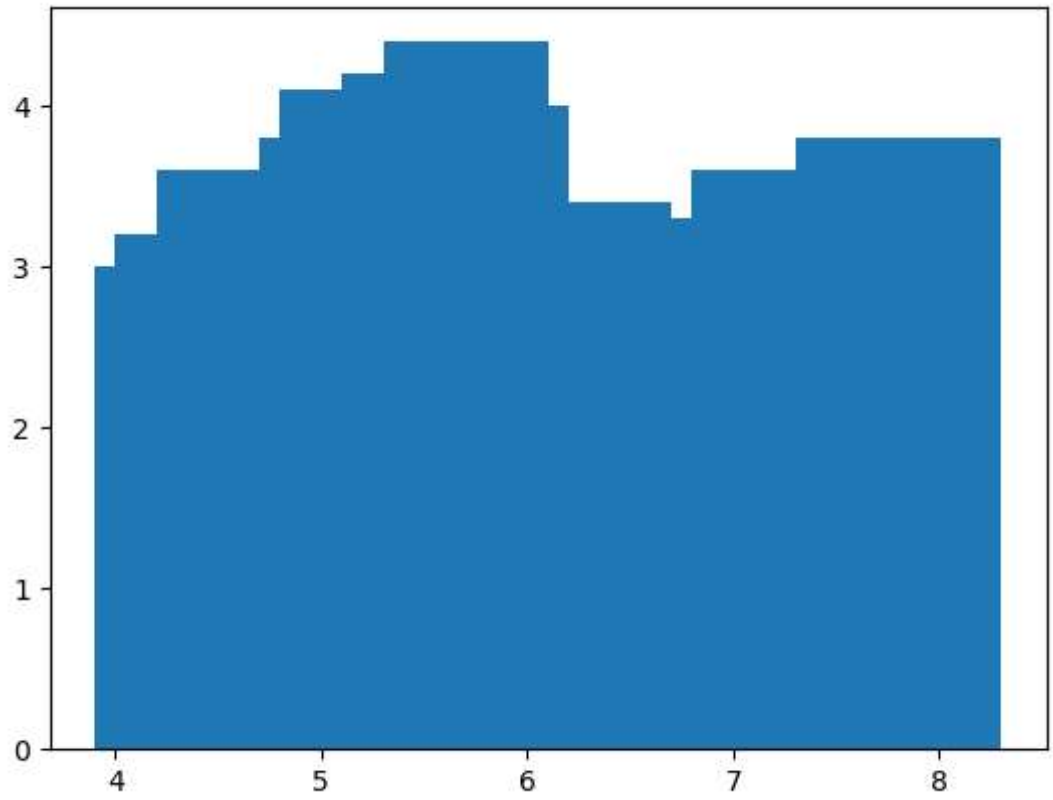


In [45]:

```
#Import dataset “iris.csv”. Write a Python program to create a Bar plot to get the
#frequency of the three species of the Iris data.
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv("Documents\iris.csv");
plt.bar(data["sepal.length"],data["sepal.width"])
```

Out[45]:

```
<BarContainer object of 150 artists>
```

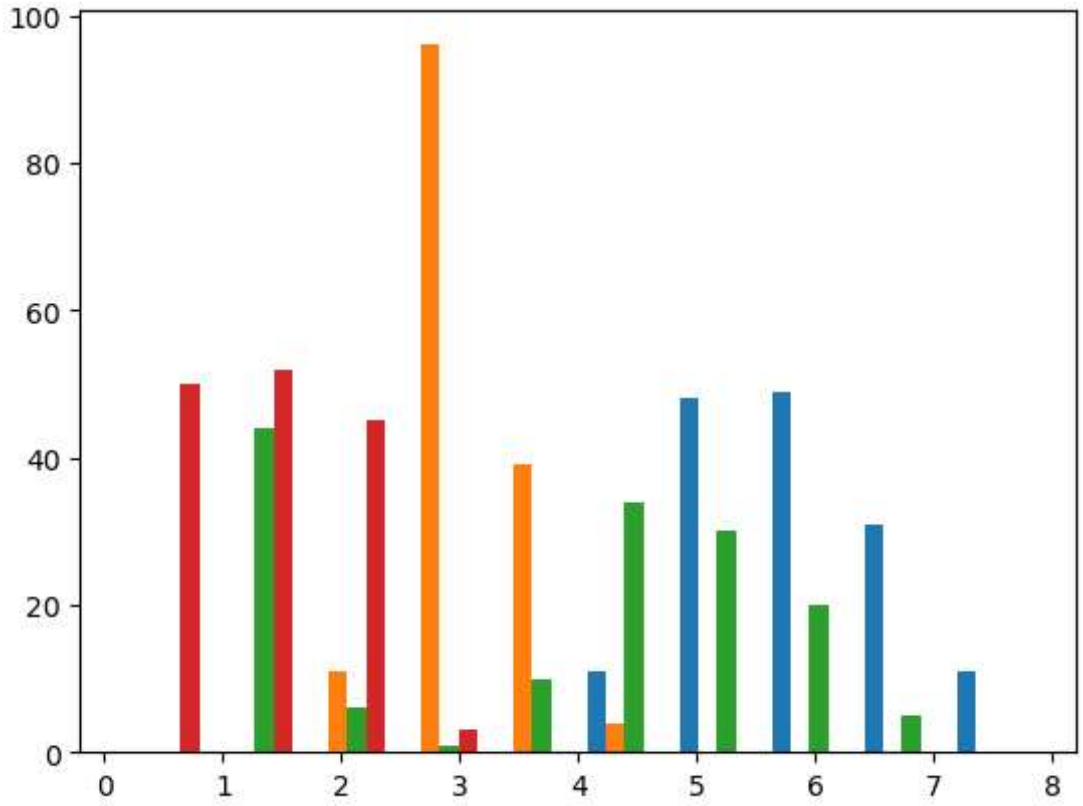


In [44]:

```
#Write a Python program to create a histogram of the three species of the Iris data.
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv("Documents\iris.csv");
newdata=data[["sepal.length","sepal.width","petal.length","petal.width"]]
plt.hist(newdata)
```

Out[44]:

```
(array([[ 0.,  0.,  0.,  0.,  0., 11., 48., 49., 31., 11.],
        [ 0.,  0., 11., 96., 39.,  4.,  0.,  0.,  0.,  0.],
        [ 0., 44.,  6.,  1., 10., 34., 30., 20.,  5.,  0.],
        [50., 52., 45.,  3.,  0.,  0.,  0.,  0.,  0.,  0.]]),
 array([0.1 , 0.88, 1.66, 2.44, 3.22, 4.  , 4.78, 5.56, 6.34, 7.12, 7.9 ]),
 <a list of 4 BarContainer objects>)
```



In [52]:

```
#Write a Python program [15]
#1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the d
#2. To print the shape, number of rows-columns, data types, feature names and the descripti
#3. To view basic statistical details of the data.
#4. To Add 5 rows with duplicate values and missing values. Add a column ‘remarks’ with emp
import pandas as pd
dt=pd.DataFrame(columns=['name','age','per'])
dt.loc[0]=['om',21,89]
dt.loc[1]=['sai',21,87]
dt.loc[2]=['ram',21,89]
dt.loc[3]=['sham',21,89]
dt.loc[4]=['aman',21,89]
dt.loc[5]=['anvesh',21,99]
dt.loc[6]=['gauri',21,79]
dt.loc[7]=['ganu',21,89]
dt.loc[8]=['nikkk',21,99]
dt.loc[9]=['ruts',21,79]
dt.loc[10]=['hari',21,89]
print(dt)
print("shape=",dt.shape)
print("description=",dt.describe)
dt.loc[11]=['gauri',21,79]
dt.loc[12]=['ganu',21,89]
dt.loc[13]=['nikkk',21,99]
dt.loc[14]=['ruts',21,79]
dt.loc[15]=['hari',21,89]
dt['remark']=None
print(dt)
```

```
      name  age  per
0       om   21   89
1      sai   21   87
2      ram   21   89
3     sham   21   89
4     aman   21   89
5   anvesh   21   99
6    gauri   21   79
7     ganu   21   89
8    nikkk   21   99
9     ruts   21   79
10    hari   21   89
shape= (11, 3)
description= <bound method NDFrame.describe of       name  age  per
0       om   21   89
1      sai   21   87
2      ram   21   89
3     sham   21   89
4     aman   21   89
5   anvesh   21   99
6    gauri   21   79
7     ganu   21   89
8    nikkk   21   99
9     ruts   21   79
10    hari   21   89>
      name  age  per remark
0       om   21   89   None
1      sai   21   87   None
2      ram   21   89   None
3     sham   21   89   None
```

```
4     aman   21   89   None
5   anvesh   21   99   None
6    gauri   21   79   None
7     ganu   21   89   None
8    nikkk   21   99   None
9     ruts   21   79   None
10    hari   21   89   None
11   gauri   21   79   None
12    ganu   21   89   None
13   nikkk   21   99   None
14    ruts   21   79   None
15    hari   21   89   None
```

In [ ]:

```
#Dataset Name: winequality-red.csv [15]
#Write a program in python to perform following task
#a. Rescaling: Normalised the dataset using MinMaxScaler class
#b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of
#c. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possi
```

In [ ]:

```

```