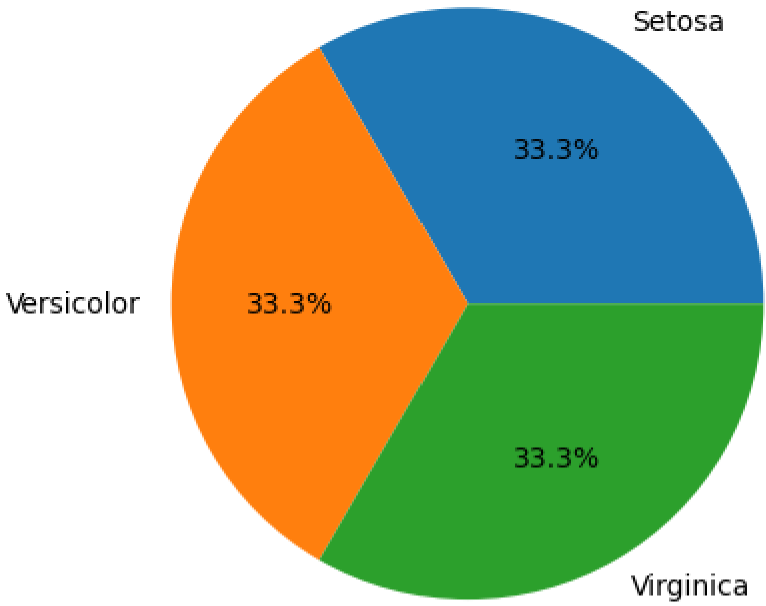
Slip 1

1 Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

In [14]:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('/home/anup/i/iris.csv')
s_count=data['variety'].value_counts()
plt.pie(s_count,labels=s_count.index,autopct='%1.1f%%')
plt.show()
data
```



Out[14]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

In [16]:

```
import pandas as pd
df = pd.read_csv("/home/anup/i/iris.csv")
df.describe()
```

Out[16]:

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Slip 2

1 Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

In [23]:
```
import pandas as pd
data = pd.read_csv('data.csv')
data
data['Skin_Fold'].fillna(data['Skin_Fold'].mean(),inplace=True)
data['Serum_Insulin'].fillna(data['Serum_Insulin'].mean(),inplace=True)
print(data)
```

```
     Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
0           6    148.0          72.0   35.00000     155.548223  33.6   
1           1     85.0          66.0   29.00000     155.548223  26.6   
2           8    183.0          64.0   29.15342     155.548223  23.3   
3           1     89.0          66.0   23.00000      94.000000  28.1   
4           0    137.0          40.0   35.00000     168.000000  43.1   
..        ...      ...           ...        ...            ...   ...   
763        10    101.0          76.0   48.00000     180.000000  32.9   
764         2    122.0          70.0   27.00000     155.548223  36.8   
765         5    121.0          72.0   23.00000     112.000000  26.2   
766         1    126.0          60.0   29.15342     155.548223  30.1   
767         1     93.0          70.0   31.00000     155.548223  30.4   

     Diabetes_Pedigree  Age  Class  
0                0.627   50      1  
1                0.351   31      0  
2                0.672   32      1  
3                0.167   21      0  
4                2.288   33      1  
..                 ...  ...    ...  
763              0.171   63      0  
764              0.340   27      0  
765              0.245   30      0  
766              0.349   47      1  
767              0.315   23      0  

[768 rows x 9 columns]
```

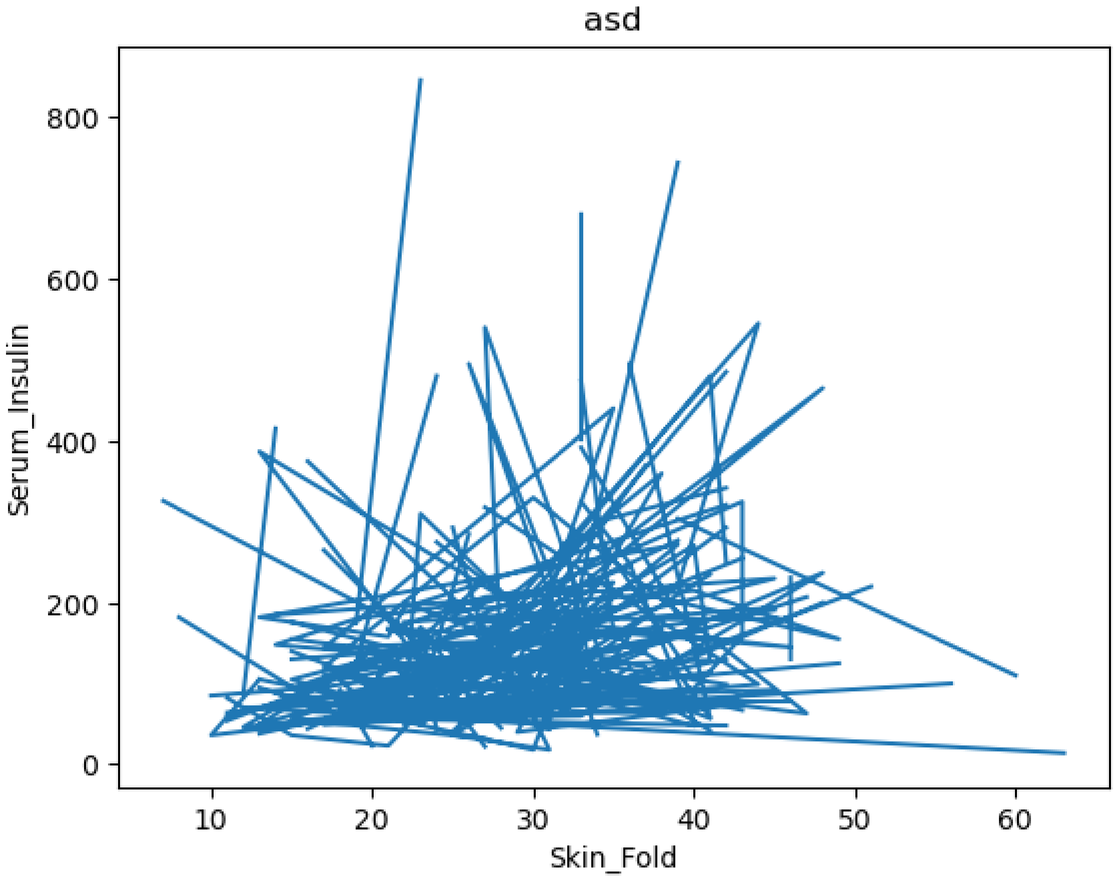
Write a Python program to generate a line plot of name Vs salary

In [29]:

```python
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv('data.csv')
data
plt.plot(df['Skin_Fold'],df['Serum_Insulin'])
plt.xlabel('Skin_Fold')
plt.ylabel('Serum_Insulin')
plt.title("asd")
```

Out[29]:

```
Text(0.5, 1.0, 'asd')
```



3 Download the heights and weights dataset and load the dataset froma given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display shape of the dataset.

In [36]:

```python
import pandas as pd
df = pd.read_csv('data.csv')
df
print("first 10 rows are :")
print(df.head(10))
print()
print("Last 10 rows are :")
print(df.tail(10))
print()
print("Random 20 rows are")
print(df.sample(20))
print()
print("The shape of dataset is")
print(df.shape)
```

```
first 10 rows are :
   Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
0         6    148.0          72.0       35.0            NaN  33.6   
1         1     85.0          66.0       29.0            NaN  26.6   
2         8    183.0          64.0        NaN            NaN  23.3   
3         1     89.0          66.0       23.0           94.0  28.1   
4         0    137.0          40.0       35.0          168.0  43.1   
5         5    116.0          74.0        NaN            NaN  25.6   
6         3     78.0          50.0       32.0           88.0  31.0   
7        10    115.0           NaN        NaN            NaN  35.3   
8         2    197.0          70.0       45.0          543.0  30.5   
9         8    125.0          96.0        NaN            NaN   NaN   

   Diabetes_Pedigree  Age  Class  
0              0.627   50      1  
1              0.351   31      0  
2              0.672   32      1  
3              0.167   21      0  
4              2.288   33      1  
5              0.201   30      0  
6              0.248   26      1  
7              0.134   29      0  
8              0.158   53      1  
9              0.232   54      1  

Last 10 rows are :
     Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
758         1    106.0          76.0        NaN            NaN  37.5   
759         6    190.0          92.0        NaN            NaN  35.5   
760         2     88.0          58.0       26.0           16.0  28.4   
761         9    170.0          74.0       31.0            NaN  44.0   
762         9     89.0          62.0        NaN            NaN  22.5   
763        10    101.0          76.0       48.0          180.0  32.9   
764         2    122.0          70.0       27.0            NaN  36.8   
765         5    121.0          72.0       23.0          112.0  26.2   
766         1    126.0          60.0        NaN            NaN  30.1   
767         1     93.0          70.0       31.0            NaN  30.4   

     Diabetes_Pedigree  Age  Class  
758              0.197   26      0  
759              0.278   66      1  
760              0.766   22      0  
761              0.403   43      1  
762              0.142   33      0  
763              0.171   63      0  
764              0.340   27      0  
765              0.245   30      0  
766              0.349   47      1  
767              0.315   23      0  

Random 20 rows are
     Pregnant  Glucose  Diastolic_BP  Skin_Fold  Serum_Insulin   BMI  \
595         0    188.0          82.0       14.0          185.0  32.0   
535         4    132.0           NaN        NaN            NaN  32.9   
41          7    133.0          84.0        NaN            NaN  40.2   
103         1     81.0          72.0       18.0           40.0  26.6   
51          1    101.0          50.0       15.0           36.0  24.2   
407         0    101.0          62.0        NaN            NaN  21.9   
170         6    102.0          82.0        NaN            NaN  30.8   
216         5    109.0          62.0       41.0          129.0  35.8   
321         3    112.0          74.0       30.0            NaN  31.6   
```

```
599         1    109.0          38.0       18.0          120.0  23.1
52          5     88.0          66.0       21.0           23.0  24.4
373         2    105.0          58.0       40.0           94.0  34.9
81          2     74.0           NaN        NaN            NaN   NaN
126         3    120.0          70.0       30.0          135.0  42.9
693         7    129.0          68.0       49.0          125.0  38.5
522         6    114.0           NaN        NaN            NaN   NaN
647         0    179.0          50.0       36.0          159.0  37.8
451         2    134.0          70.0        NaN            NaN  28.9
155         7    152.0          88.0       44.0            NaN  50.0
495         6    166.0          74.0        NaN            NaN  26.6

     Diabetes_Pedigree  Age  Class
595              0.682   22      1
535              0.302   23      1
41               0.696   37      0
103              0.283   24      0
51               0.526   26      0
407              0.336   25      0
170              0.180   36      1
216              0.514   25      1
321              0.197   25      1
599              0.407   26      0
52               0.342   30      0
373              0.225   25      0
81               0.102   22      0
126              0.452   30      0
693              0.439   43      1
522              0.189   26      0
647              0.455   22      1
451              0.542   23      1
155              0.337   36      1
495              0.304   66      0

The shape of dataset is
(768, 9)
```

Slip 3

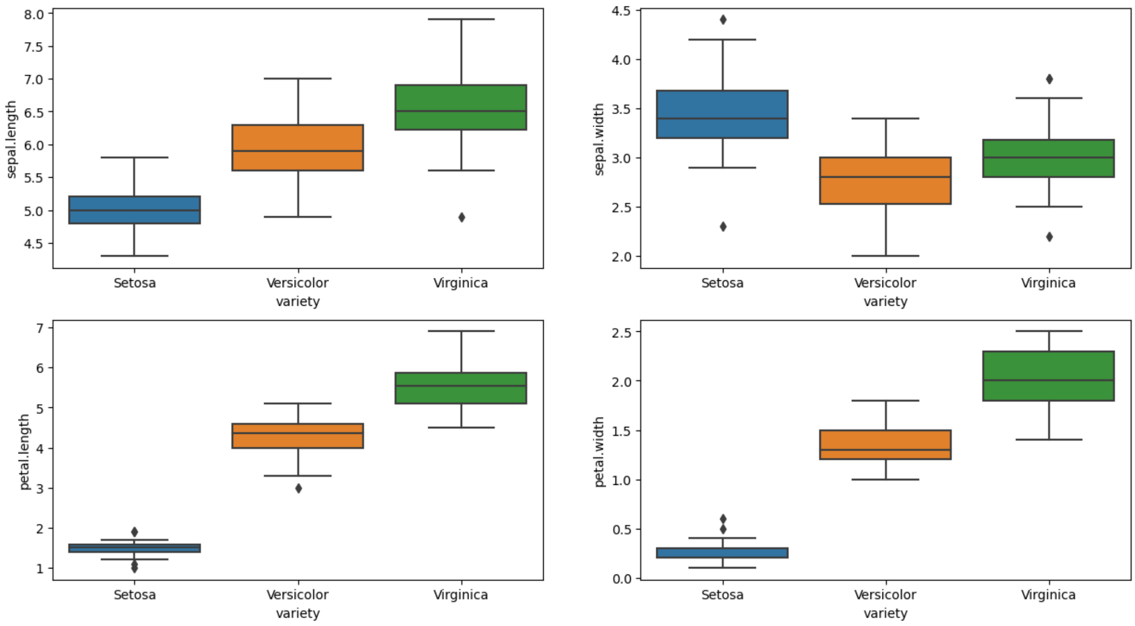
Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

In [45]:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
df = pd.read_csv('/home/anup/i/iris.csv')
df
#box plot for sepal length
plt.figure(figsize=(15,8))
plt.subplot(2,2,1)
sns.boxplot(x='variety',y='sepal.length',data=df)
#box plot for sepal width
plt.subplot(2,2,2)
sns.boxplot(x='variety',y='sepal.width',data=df)
#box plot for petal length
plt.subplot(2,2,3)
sns.boxplot(x='variety',y='petal.length',data=df)
#box plot for petal width
plt.subplot(2,2,4)
sns.boxplot(x='variety',y='petal.width',data=df)
```

Out[45]:

```
<AxesSubplot:xlabel='variety', ylabel='petal.width'>
```



Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

In [46]:

```
import pandas as pd
df = pd.read_csv("/home/anup/i/iris.csv")
df.describe()
```

Out[46]:

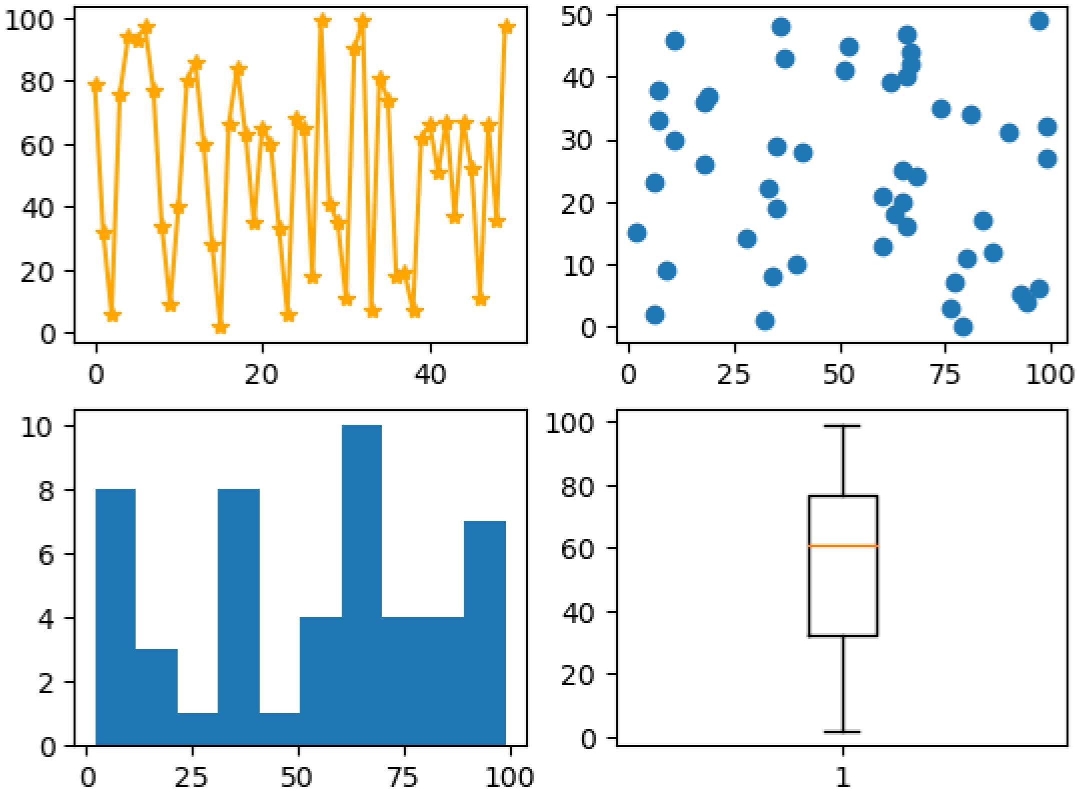
	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Slip 4

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.

In [55]:

```
import numpy as np
import matplotlib.pyplot as plt
arr = np.random.randint(0,100,50)

#line chart
plt.subplot(2,2,1)
plt.plot(arr,color='orange',label='abc',marker='*')
#scatterplot
plt.subplot(2,2,2)
plt.scatter(arr,range(50))
#histogram
plt.subplot(2,2,3)
plt.hist(arr)
#boxplot
plt.subplot(2,2,4)
plt.boxplot(arr)

plt.show()
```



) 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)

In [64]:

```
import pandas as pd
data=pd.read_csv('data.csv')
data
s=data.shape
num_row,num_col=s
data_type=data.dtypes
feature_name=data.columns
description=data.describe()
print(s)
print(num_row)
print(num_col)
print(data_type)
print(feature_name)
print(description)
```

```
(768, 9)
768
9
Pregnant               int64
Glucose              float64
Diastolic_BP         float64
Skin_Fold            float64
Serum_Insulin        float64
BMI                  float64
Diabetes_Pedigree    float64
Age                    int64
Class                  int64
dtype: object
Index(['Pregnant', 'Glucose', 'Diastolic_BP', 'Skin_Fold', 'Serum_Insuli
n',
       'BMI', 'Diabetes_Pedigree', 'Age', 'Class'],
      dtype='object')
         Pregnant     Glucose  Diastolic_BP   Skin_Fold  Serum_Insulin  \
count  768.000000  763.000000    733.000000  541.000000     394.000000   
mean     3.845052  121.686763     72.405184   29.153420     155.548223   
std      3.369578   30.535641     12.382158   10.476982     118.775855   
min      0.000000   44.000000     24.000000    7.000000      14.000000   
25%      1.000000   99.000000     64.000000   22.000000      76.250000   
50%      3.000000  117.000000     72.000000   29.000000     125.000000   
75%      6.000000  141.000000     80.000000   36.000000     190.000000   
max     17.000000  199.000000    122.000000   99.000000     846.000000   

              BMI  Diabetes_Pedigree         Age       Class  
count  757.000000         768.000000  768.000000  768.000000  
mean    32.457464           0.471876   33.240885    0.348958  
std      6.924988           0.331329   11.760232    0.476951  
min     18.200000           0.078000   21.000000    0.000000  
25%     27.500000           0.243750   24.000000    0.000000  
50%     32.300000           0.372500   29.000000    0.000000  
75%     36.600000           0.626250   41.000000    1.000000  
max     67.100000           2.420000   81.000000    1.000000  
```

slip 7

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).

In [70]:

```
import pandas as pd
data = pd.read_csv('data.csv')
#apply onehot coding
data = pd.get_dummies(data,columns=['Glucose'])
data
#apply lebel encoding
data['Skin_Fold']=data['Skin_Fold'].map({'Yes':1,'No':0})
data
```

Out[70]:

	Pregnant	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
0	6	72.0	NaN	NaN	33.6	0.627	50	1
1	1	66.0	NaN	NaN	26.6	0.351	31	0
2	8	64.0	NaN	NaN	23.3	0.672	32	1
3	1	66.0	NaN	94.0	28.1	0.167	21	0
4	0	40.0	NaN	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...
763	10	76.0	NaN	180.0	32.9	0.171	63	0
764	2	70.0	NaN	NaN	36.8	0.340	27	0
765	5	72.0	NaN	112.0	26.2	0.245	30	0
766	1	60.0	NaN	NaN	30.1	0.349	47	1
767	1	70.0	NaN	NaN	30.4	0.315	23	0

768 rows × 143 columns

Slip 8

Write a program in python to perform following task : [15] 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 [71]:

```python
import pandas as pd
import s
data = pd.read_csv('/home/anup/i/iris.csv')
print(data)
data=(data - data.mean())/data.std() # to standardizing data
print(data)
```

```
     sepal.length  sepal.width  petal.length  petal.width    variety
0             5.1          3.5           1.4          0.2     Setosa
1             4.9          3.0           1.4          0.2     Setosa
2             4.7          3.2           1.3          0.2     Setosa
3             4.6          3.1           1.5          0.2     Setosa
4             5.0          3.6           1.4          0.2     Setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
     petal.length  petal.width  sepal.length  sepal.width variety
0       -1.335752    -1.311052     -0.897674     1.015602     NaN
1       -1.335752    -1.311052     -1.139200    -0.131539     NaN
2       -1.392399    -1.311052     -1.380727     0.327318     NaN
3       -1.279104    -1.311052     -1.501490     0.097889     NaN
4       -1.335752    -1.311052     -1.018437     1.245030     NaN
..            ...          ...           ...          ...     ...
145      0.816859     1.443994      1.034539    -0.131539     NaN
146      0.703564     0.919223      0.551486    -1.278680     NaN
147      0.816859     1.050416      0.793012    -0.131539     NaN
148      0.930154     1.443994      0.430722     0.786174     NaN
149      0.760211     0.788031      0.068433    -0.131539     NaN

[150 rows x 5 columns]

/tmp/ipykernel_4199/696278818.py:4: FutureWarning: The default value of nu
meric_only in DataFrame.mean is deprecated. In a future version, it will d
efault to False. In addition, specifying 'numeric_only=None' is deprecate
d. Select only valid columns or specify the value of numeric_only to silen
ce this warning.
  data=(data - data.mean())/data.std()
/tmp/ipykernel_4199/696278818.py:4: FutureWarning: The default value of nu
meric_only in DataFrame.std is deprecated. In a future version, it will de
fault to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence
this warning.
  data=(data - data.mean())/data.std()
```

In [72]:

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

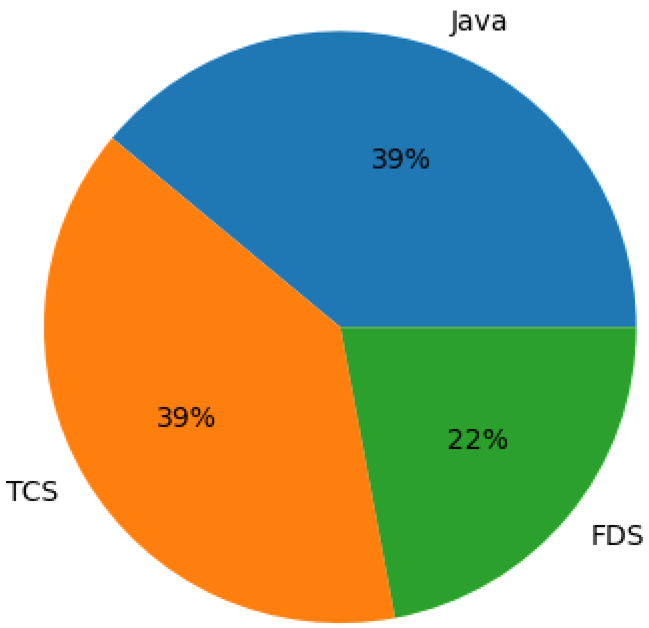
```
---------------------------------------------------------------------------
-
ModuleNotFoundError                       Traceback (most recent call las
t)
Cell In [72], line 1
----> 1 from sklearn.preprocessing import MinMaxScaler, StandardScaler, No
rmalizer

ModuleNotFoundError: No module named 'sklearn'
```

Slip 9

Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

In [75]:

```
import matplotlib.pyplot as plt
subjects =['Java','TCS','FDS']
marks=[99,99,56]
plt.pie(marks,labels=subjects,autopct='%1.f%%')
plt.show()
```



Slip 10

Write a python program to Display column-wise mean, and median for SOCR- HeightWeight dataset.

In [77]:
```
import pandas as pd
data = pd.read_csv('data.csv')
print(data.mean())
print(data.median())
```

```
Pregnant              3.845052
Glucose             121.686763
Diastolic_BP         72.405184
Skin_Fold            29.153420
Serum_Insulin       155.548223
BMI                  32.457464
Diabetes_Pedigree     0.471876
Age                  33.240885
Class                 0.348958
dtype: float64
Pregnant              3.0000
Glucose             117.0000
Diastolic_BP         72.0000
Skin_Fold            29.0000
Serum_Insulin       125.0000
BMI                  32.3000
Diabetes_Pedigree     0.3725
Age                  29.0000
Class                 0.0000
dtype: float64
```

Write a python program to compute sum of Manhattan distance between all pairs of points.|

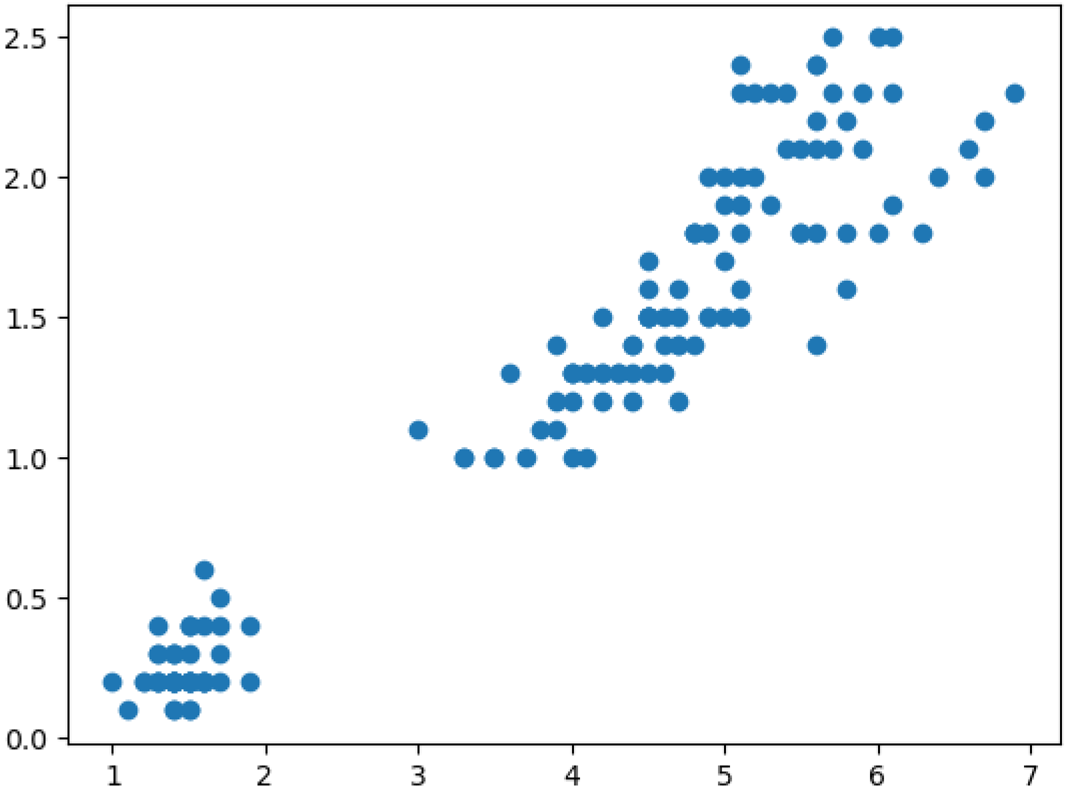
In [82]:
```
p=(0,2,1)
q=(2,5,6)
dist=abs(p[0]-q[0])+abs(p[1]-q[1])+abs(p[2]-q[2])
dist
```

Out[82]:
```
10
```

Write a Python program to create a graph to find relationship between the petal length and petal width.(Use iris.csv dataset)

In [89]:

```
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_csv('/home/anup/i/iris.csv')
plt.scatter(data['petal.length'],data['petal.width'])
plt.show()
```



Write a Python program to find the maximum and minimum value of a given flattened array

In [90]:

```
import numpy as np
arr = np.array([1,2,3,4,5,6,7,8,9])
max = np.max(arr)
min = np.min(arr)
print(max)
print(min)
```

```
9
1
```

Write a python program to create a data frame for students' information such as name, graduation percentage and age. Display average age of students, average of graduation percentage.

In [94]:

```
import pandas as pd
data = {'name':['asd','fgh','zxc'],'per':[12,56,89],'age':[56,98,5]}
df=pd.DataFrame(data)
df
avg=df['age'].mean()
avg
avg1=df['per'].mean()
avg1
```

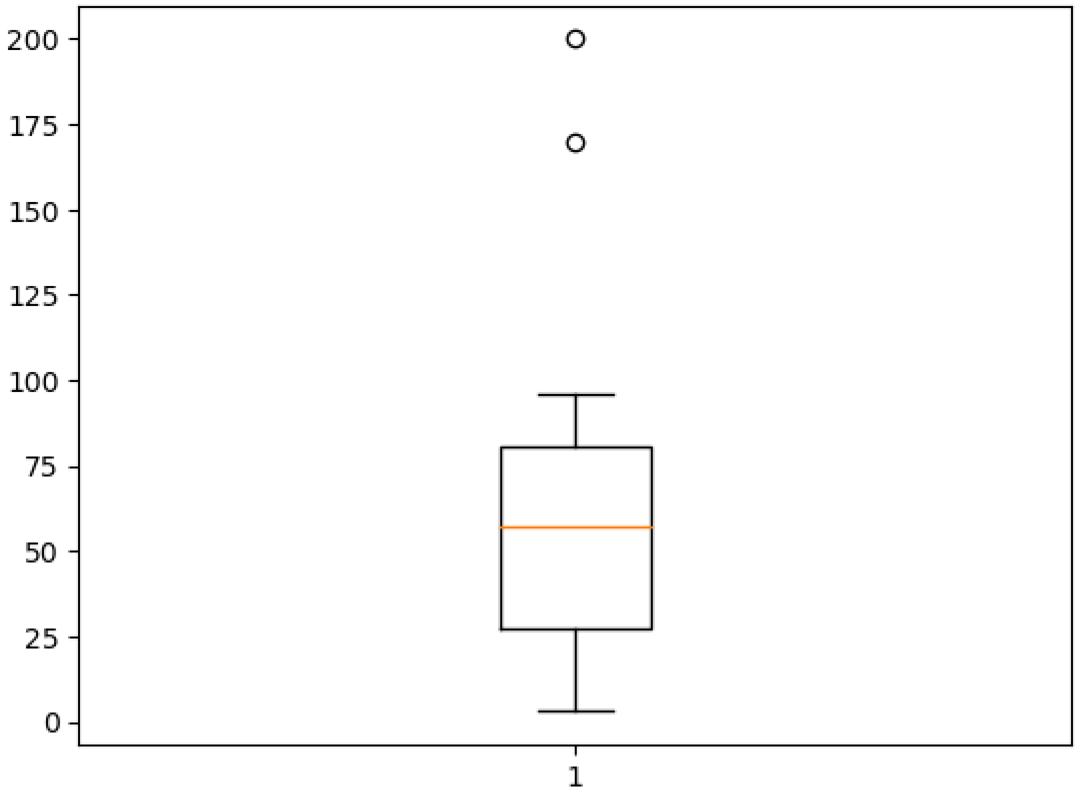
Out[94]:

```
52.333333333333336
```

Add two outliers to the above data and display the box plot.

In [101]:

```
import numpy as np
import matplotlib.pyplot as plt
arr = np.random.randint(0,100,50)
arr[0]=200
arr[1]=170

plt.boxplot(arr)
plt.show()
```



In [ ]: