```
#Practice Questions
import numpy as np
from scipy import linalg
import pandas as pd
import matplotlib.pyplot as plt
```

```
mylist = list(range(1,13))
print(mylist)
print("Type of mylist : ",type(mylist))
myarr = np.array(mylist)
print(myarr)
print("Type of myarr : ",type(myarr))
arr_2d = myarr.reshape(2,-1)
print("\n2D array\n",arr_2d)
arr_3d = myarr.reshape(2,2,-1)
print("\n3D array\n",arr_3d)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Type of mylist :  <class 'list'>
[ 1  2  3  4  5  6  7  8  9 10 11 12]
Type of myarr :  <class 'numpy.ndarray'>

2D array
 [[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]]

3D array
 [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [10 11 12]]]
```

```
arr = np.random.randint(0,10,(3,3))
print("\nRandom Matrix\n",arr)
```

```

Random Matrix
 [[4 5 3]
 [9 5 2]
 [7 4 5]]
```

```
det = linalg.det(arr)
print(det)
```

```
-84.0
```

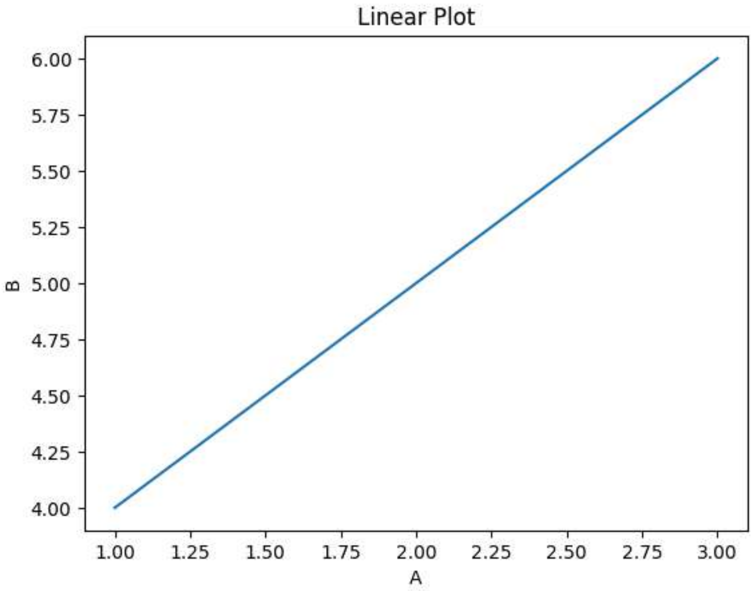
```
eigvals, eigvecs = linalg.eig(arr)
print("\nEigen Values\n",eigvals)
print("\nEigen Vectors\n",eigvecs)
```

```

Eigen Values
 [14.21746305+0.j -2.54184673+0.j  2.32438368+0.j]

Eigen Vectors
 [[-0.48262945 -0.65470702 -0.04202229]
 [-0.6080223   0.72164069 -0.50364885]
 [-0.63037902  0.22492984  0.86288583]]
```

```
data = {'A': [1, 2, 3], 'B': [4, 5, 6]}
df = pd.DataFrame(data)
plt.plot(df['A'], df['B'])
plt.title("Linear Plot")
plt.xlabel("A")
plt.ylabel("B")
plt.show()
```



```
mylist = list(range(10,26))
my_series = pd.Series(mylist)
print(my_series)
print(type(my_series))
```

```
0     10
1     11
2     12
3     13
4     14
5     15
6     16
7     17
8     18
9     19
10    20
11    21
12    22
13    23
14    24
15    25
dtype: int64
<class 'pandas.core.series.Series'>
```

```
print("Index of the series",my_series.index)
print("Value of the series",my_series.values)
```

```
Index of the series RangeIndex(start=0, stop=16, step=1)
Value of the series [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
```

```
list1 = list(range(1,11))
print(type(list1))
print(list1)
np_arr = np.array(list1)
print(type(np_arr))
print(np_arr)
pd_series = pd.Series(np_arr)
print(type(my_series))
print(my_series)
```

```
<class 'list'>
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
<class 'numpy.ndarray'>
[ 1  2  3  4  5  6  7  8  9 10]
<class 'pandas.core.series.Series'>
0     10
1     11
2     12
3     13
4     14
5     15
6     16
7     17
```

```
8     18
9     19
10    20
11    21
12    22
13    23
14    24
15    25
dtype: int64
```

```
list = ['apple','banana','mango']
pd_ser = pd.Series(list,index=[1,2,3])
print(pd_ser)
```

```
1     apple
2    banana
3     mango
dtype: object
```

```
print(pd_ser[1])
print(pd_ser[2])
print(pd_ser[3])
```

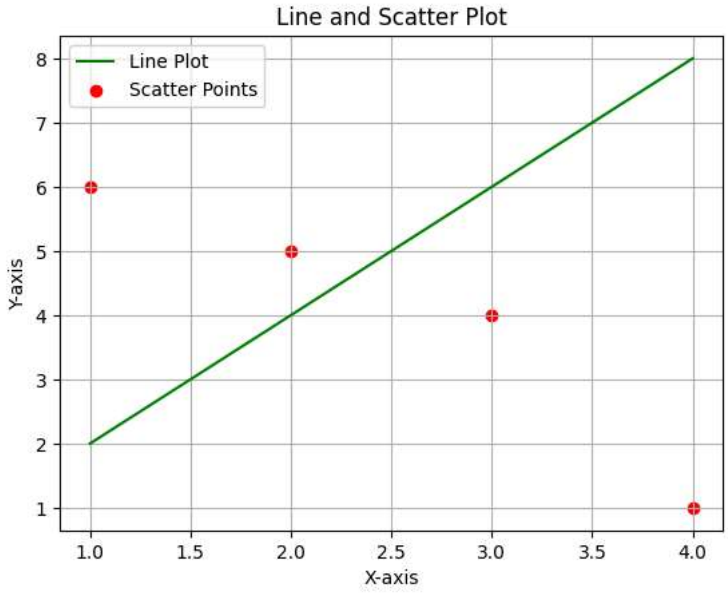
```
apple
banana
mango
```

```
data_read = pd.read_csv("BostonHousing.csv")
print(data_read.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

   LSTAT  MEDV  CAT. MEDV
0   4.98  24.0          0
1   9.14  21.6          0
2   4.03  34.7          1
3   2.94  33.4          1
4   5.33  36.2          1
```

```
plt.figure()
plt.plot([1, 2, 3, 4], [2, 4, 6, 8], color='green', label="Line Plot")
plt.scatter([1, 2, 3, 4], [6, 5, 4, 1], color='red', label="Scatter Points")
plt.title("Line and Scatter Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.grid(True)
plt.show()
```



```
#Next Question Set
```

```
#Practice Session Questions
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'Salary': [50000, 60000, 70000]
}

df = pd.DataFrame(data)
print(df)
```

```
      Name  Age  Salary
0    Alice   25   50000
1      Bob   30   60000
2  Charlie   35   70000
```

```
from sklearn.datasets import load_iris
import pandas as pd

iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2   
1                4.9               3.0                1.4               0.2   
2                4.7               3.2                1.3               0.2   
3                4.6               3.1                1.5               0.2   
4                5.0               3.6                1.4               0.2   

   target  
0       0  
1       0  
2       0  
3       0  
4       0  
```

```
#from google.colab import files
#uploaded = files.upload()
df = pd.read_csv('BostonHousing.csv')
print(df.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3   
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8   
```

```
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

   LSTAT  MEDV  CAT. MEDV
0   4.98  24.0          0
1   9.14  21.6          0
2   4.03  34.7          1
3   2.94  33.4          1
4   5.33  36.2          1
```

```
import pandas as pd
from scipy import stats

data = {
    'Score': [85, 90, 88, 92, 85, 95, 89]
}
df = pd.DataFrame(data)

mean = df['Score'].mean()
median = df['Score'].median()
mode = df['Score'].mode()[0]
variance = df['Score'].var()
std_dev = df['Score'].std()

print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")
print(f"Variance: {variance}")
print(f"Standard Deviation: {std_dev}")
```

```
Mean: 89.14285714285714
Median: 89.0
Mode: 85
Variance: 13.14285714285714
Standard Deviation: 3.6253078686998625
```

```
import numpy as np

data = np.array([[1, 2], [3, 4], [5, 6]])
print(data)
print(data.shape)
reshaped = data.reshape((2, 3))
print(reshaped)
print(reshaped.shape)
```

```
[[1 2]
 [3 4]
 [5 6]]
(3, 2)
[[1 2 3]
 [4 5 6]]
(2, 3)
```

```
filtered_df = df[df['Score'] > 90]
print(filtered_df)
```

```
   Score
3     92
5     95
```

```
df1 = pd.DataFrame({'ID': [1, 2], 'Name': ['Alice', 'Bob']})
df2 = pd.DataFrame({'ID': [1, 2], 'Score': [90, 85]})

merged_df = pd.merge(df1, df2, on='ID')
print(merged_df)
```

```
   ID   Name  Score
0   1  Alice     90
1   2    Bob     85
```

```
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, None, 35]
}
```

```
df = pd.DataFrame(data)

df['Age'] = df['Age'].fillna(df['Age'].mean())

print(df)
```

```
      Name   Age
0    Alice  25.0
1      Bob  30.0
2  Charlie  35.0
```

```
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, None, 35]
}
df = pd.DataFrame(data)
df = df.dropna()

print(df)
```

```
      Name   Age
0    Alice  25.0
2  Charlie  35.0
```

```
from sklearn.preprocessing import MinMaxScaler

data = pd.DataFrame({'Score': [60, 70, 80, 90, 100]})

scaler = MinMaxScaler()
data['Normalized'] = scaler.fit_transform(data[['Score']])
```