```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

Pandas

1. read_csv()
2. head()
3. tail()
4. isnull()
5. sum()
6. describe()

```
df = pd.read_csv('Real estate.csv')
```

```
df.head()
```

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8

```
df.tail()
```

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
409	410	2013.000	13.7	4082.01500	0	24.94155	121.50381	15.4
410	411	2012.667	5.6	90.45606	9	24.97433	121.54310	50.0
411	412	2013.250	18.8	390.96960	7	24.97923	121.53986	40.6
412	413	2013.000	8.1	104.81010	5	24.96674	121.54067	52.5
413	414	2013.500	6.5	90.45606	9	24.97433	121.54310	63.9

```
df.isnull().sum()
```

```
No                                      0
X1 transaction date                     0
X2 house age                            0
X3 distance to the nearest MRT station  0
X4 number of convenience stores         0
X5 latitude                             0
X6 longitude                            0
Y house price of unit area              0
dtype: int64
```

```
df.describe()
```

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
count	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000
mean	207.500000	2013.148971	17.712560	1083.885689	4.094203	24.969030	121.533361	37.980193
std	119.655756	0.281967	11.392485	1262.109595	2.945562	0.012410	0.015347	13.606488
min	1.000000	2012.667000	0.000000	23.382840	0.000000	24.932070	121.473530	7.600000

Numpy

1. argmax()
2. sort()
3. median()
4. mean()
5. average()
6. std()

```
np.argmax(df['No'])
```

```
413
```

```
a=df['X1 transaction date']
np.sort(a)
print(a.head(10))
print(a.tail(10))
```

```
0    2012.917
1    2012.917
2    2013.583
3    2013.500
4    2012.833
5    2012.667
6    2012.667
7    2013.417
8    2013.500
9    2013.417
Name: X1 transaction date, dtype: float64
404    2013.333
405    2012.667
406    2013.167
407    2013.000
408    2013.417
409    2013.000
410    2012.667
411    2013.250
412    2013.000
413    2013.500
Name: X1 transaction date, dtype: float64
```

```
print('Median of a is :',np.median(a))
print('Mean of a is :',np.mean(a))
print('Average of a is :',np.average(a))
print('Standard Deviation of a is :',np.std(a))
```

```
Median of a is : 2013.167
Mean of a is : 2013.1489710144926
Average of a is : 2013.1489710144926
Standard Deviation of a is : 0.2816264942288487
```

scikitlearn

1. train_test_split
2. LinearRegression()
3. r2_score()
4. fit()
5. predict()

```
X=df.drop(['Y house price of unit area'], axis=1)
y=df['Y house price of unit area']


X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=42)


X_test.shape
```

```
(83, 7)
```

```
X_train.shape
```

```
(331, 7)
```

```
reg = LinearRegression()
reg.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred = reg.predict(X_test)


reg.coef_
```

```
array([-5.61695287e-03,  5.40743502e+00, -2.67827999e-01, -4.81543315e-03,
        1.08114445e+00,  2.26048799e+02, -3.01254914e+01])
```

```
reg.intercept_
```

```
-12824.256569928497
```

```
r2_score(y_test, y_pred)
```

```
0.6745228670350882
```

Matplotlib

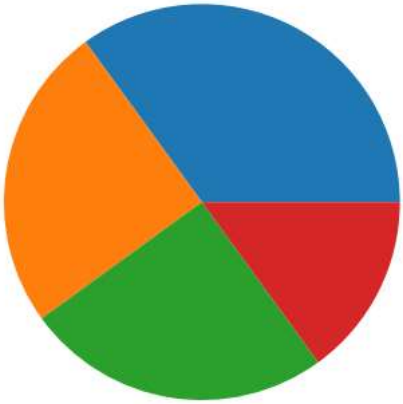
1. bar()
2. pie()
3. plot()
4. show()
5. scatter()

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```

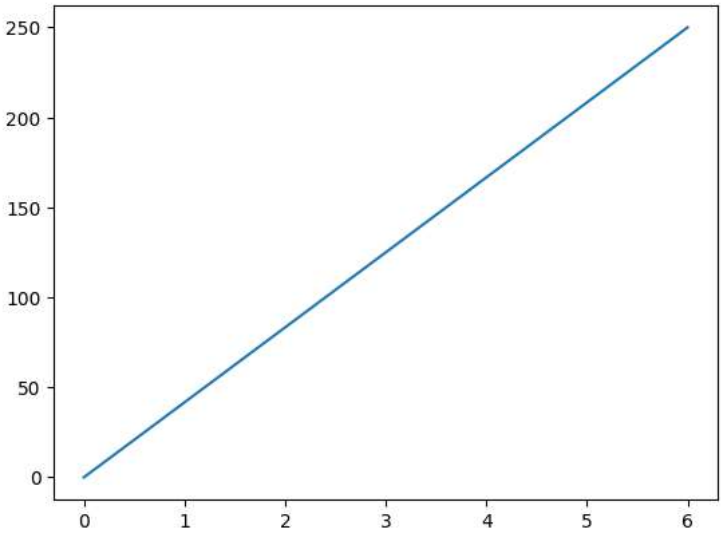

```
y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```



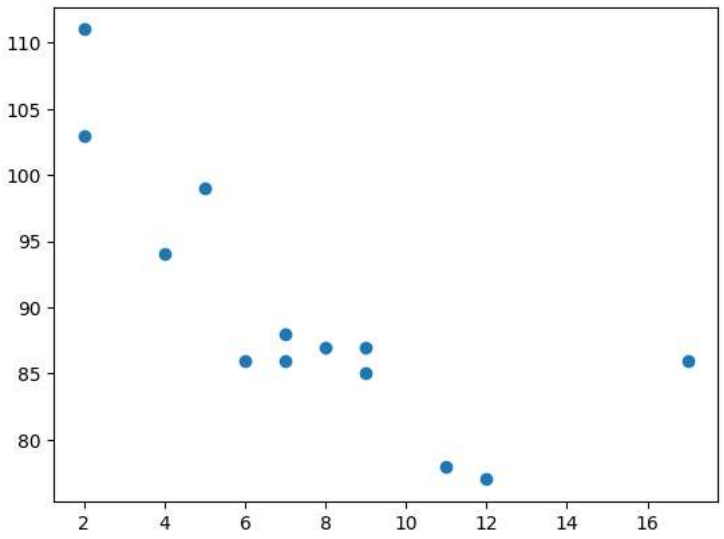
```
xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```




```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```



Colab paid products - Cancel contracts here