```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared
```

```
df=pd.read_csv('/content/height_weight.csv')
print(df.head())
print('*'*50)
print(df.tail())
```

```
   Height  Weight
0    1.47   52.21
1    1.50   53.12
2    1.52   54.48
3    1.55   55.84
4    1.57   57.20
**************************************************
    Height  Weight
10    1.73   66.28
11    1.75   68.10
12    1.78   69.92
13    1.80   72.19
14    1.83   74.46
```

```
df.describe()
```

	Height	Weight
count	15.000000	15.000000
mean	1.650667	62.078000
std	0.114235	7.037515
min	1.470000	52.210000
25%	1.560000	56.520000
50%	1.650000	61.290000
75%	1.740000	67.190000
max	1.830000	74.460000

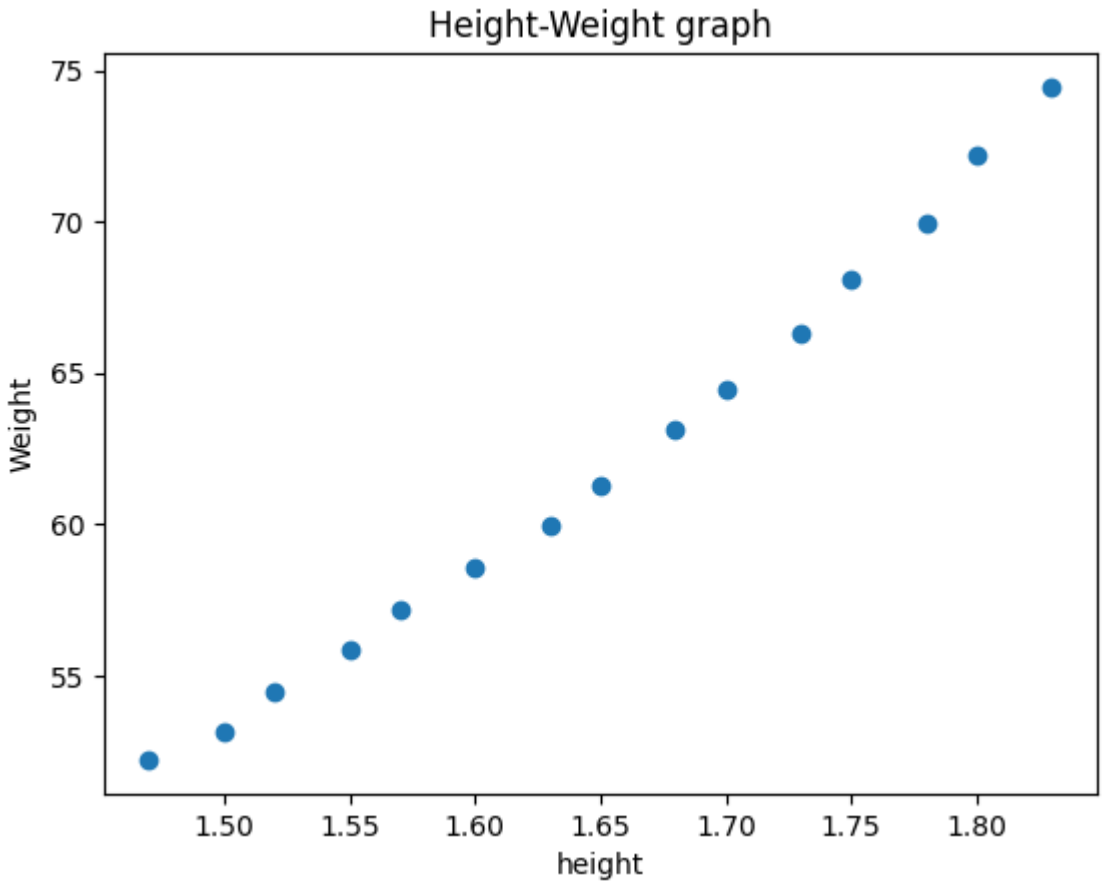
```
df.isna().sum()
```

```
Height    0
Weight    0
dtype: int64
```

```
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
xtr,xts,ytr,yts=train_test_split(x,y,test_size=0.30,random_state=42)
```

```
plt.scatter(x,y)
plt.xlabel('height')
plt.ylabel('Weight')
plt.title('Height-Weight graph')
```

```
Text(0.5, 1.0, 'Height-Weight graph')
```



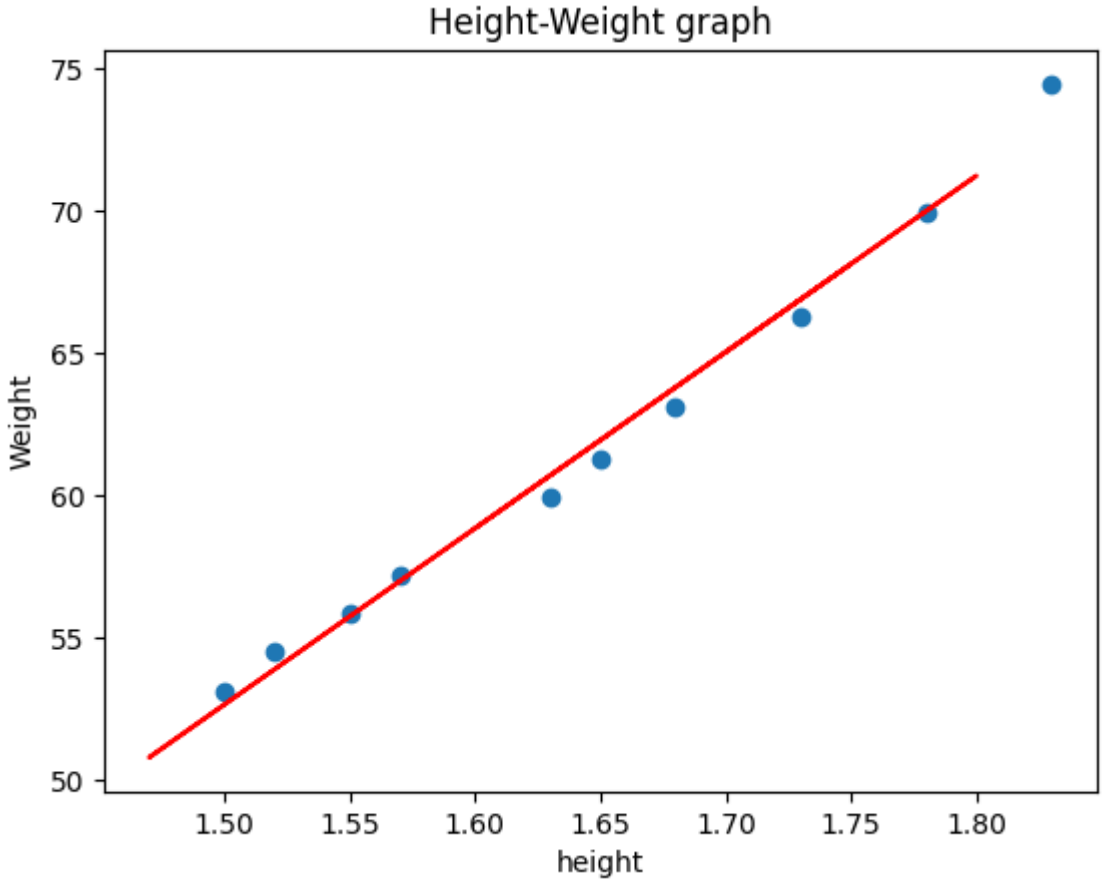

```
model=LinearRegression()
model.fit(xtr,ytr)
ypr=model.predict(xts)
```

```
df1=pd.DataFrame({'actual':yts,'predicted':ypr,'difference':yts-ypr})
df1
```

	actual	predicted	difference
0	64.47	65.035797	-0.565797
1	68.10	68.136508	-0.036508
2	52.21	50.772524	1.437476
3	72.19	71.237220	0.952780
4	58.57	58.834374	-0.264374

```
plt.scatter(xtr,ytr)
plt.plot(xts,ypr,color='r')
plt.xlabel('height')
plt.ylabel('Weight')
plt.title('Height-Weight graph')
```

```
Text(0.5, 1.0, 'Height-Weight graph')
```




```
# y=mx+c
print('slope is ',model.coef_)
print('constant is ',model.intercept_)
```

```
slope is  [62.01422981]
constant is  -40.388393810032014
```

```
# mean absolute error
print('MAE is',mean_absolute_error(yts,ypr))
```

```
MAE is 0.6513870508715769
```

```
# mean absolute error percentage
print('MAE percentage is ',mean_absolute_percentage_error(yts,ypr))
```

```
MAE percentage is  0.010911368997744855
```

```
# mean squared error
print('mse is ',mean_squared_error(yts,ypr))
```

```
mse is  0.6730959475011951
```

```
# rmse
rmse=np.sqrt(mean_squared_error(yts,ypr))
print('RMSE is ',rmse)
```

```
RMSE is  0.8204242484844015
```

```
#r2 score
print('r2 score ',r2_score(yts,ypr))
```

```
r2 score  0.986463232472019
```