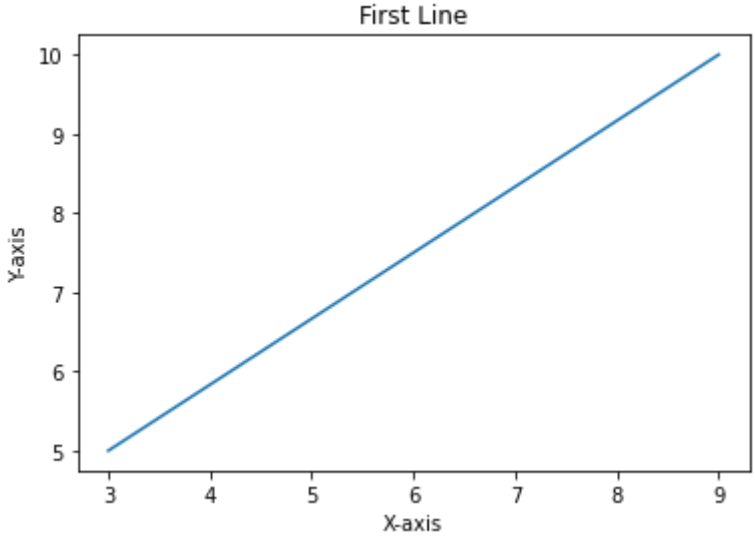
```
import matplotlib.pyplot as plt
import numpy as np

def eqnofline(p1x,p1y,p2x,p2y):
  m = (p2y-p1y)/(p2x-p1x)
  c = p1y-(m*p1x)
  print(f'Line of equation is: y={m:.2f}*x+{c}')
  x=np.array([p1x,p2x])
  y=np.array([p1y,p2y])
  plt.plot(x,y)
  plt.xlabel("X-axis")
  plt.ylabel("Y-axis")
  plt.title("First Line")
  plt.show()


eqnofline(3,5,9,10)
```

```
Line of equation is: y=0.83*x+2.5
```




```
import math
X = [171,151,124,134,156]
Y = [80,60,45,50,65]

sigmax=0
sigmay=0
for i in range(len(X)):
  sigmax+=X[i]
  sigmay+=Y[i]

meanx=sigmax/len(X)
meany=sigmay/len(Y)
X_mx=[]
Y_my=[]
for i in X:
  X_mx.append(i-meanx)
for i in Y:
  Y_my.append(i-meany)
```

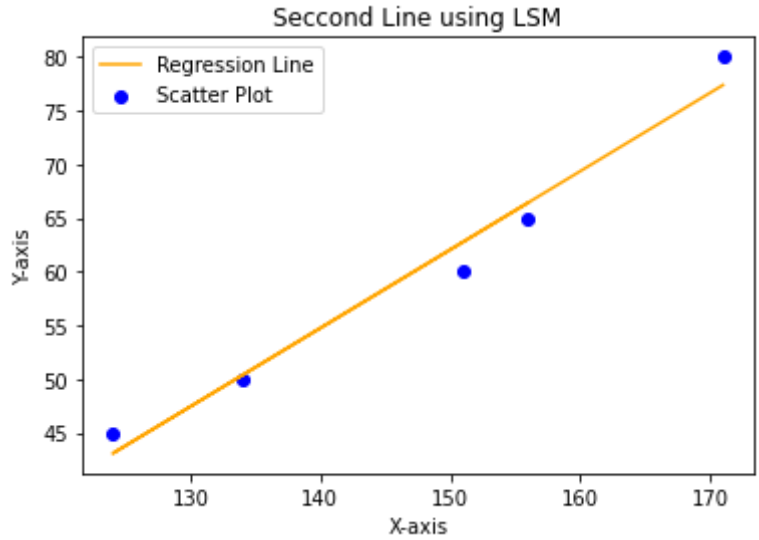
```
X_mX_sqr=[]
X_mX_Y_mY=[]
for i in X_mx:
  X_mX_sqr.append(i*i)
for i in range(len(X)):
  X_mX_Y_mY.append(X_mx[i]*Y_my[i])

s_X_mX_sqr=0
s_X_mX_Y_mY=0
for i in range(len(X)):
  s_X_mX_sqr+=X_mX_sqr[i]
  s_X_mX_Y_mY+=X_mX_Y_mY[i]
m = s_X_mX_Y_mY/s_X_mX_sqr
c = meany-(m*meanx)
print(f'Line of equation is: y={m:.2f}*x+{c:.2f}')
plt.scatter(X,Y, color='Blue', label='Scatter Plot')
plt.plot(X,m*np.array(X)+c, color='Orange', label='Regression Line')
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Seccond Line using LSM")
plt.legend()
plt.show()

pred = []
for i in X:
  y1 = m*i + c
  pred.append(y1)
s_error=0
for i in range(len(X)):
  s_error+= (pred[i]-Y[i])*(pred[i]-Y[i])

mean = s_error/len(X)
print(math.sqrt(mean))
```

```
Line of equation is: y=0.73*x+-47.38
```




```
2.02479523402097
```