In [1]:

```
import numpy as np
```

In [2]:

```
def linear_regression(x, y):
    n = np.size(x)
    m_x = np.mean(x)
    m_y = np.mean(y)
    SS_xy = np.sum(y*x) - n*m_y*m_x
    SS_xx = np.sum(x*x) - n*m_x*m_x
    b = SS_xy / SS_xx
    a = m_y - b*m_x

    return (round(a,4), round(b,4))
```

In [3]:

```
x_lin = np.array([30, 28, 32, 25, 25, 25, 22, 24, 35, 40])
y_lin = np.array([25, 30, 27, 40, 42, 40, 50, 45, 30, 25])

a, b = linear_regression(x_lin, y_lin)
print("Linear Regression")
print("Y = "+str(a)+" + ("+str(b)+")*X ")
```

```
Linear Regression
Y = 74.1151 + (-1.3537)*X
```

In [4]:

```
def multiple_regression(x1,x2,y):
    n = np.size(x1)
    x1_x1 = x1*x1
    x2_x2 = x2*x2
    y_y = y*y
    x1_x2 = x1*x2
    x1_y = x1*y
    x2_y = x2*y

    X1_square =  np.sum(x1_x1) - (np.sum(x1)*np.sum(x1))/n
    X2_square =  np.sum(x2_x2) - (np.sum(x2)*np.sum(x2))/n
    y_square =  np.sum(y_y) - (np.sum(y)*np.sum(y))/n

    X1_X2 = np.sum(x1_x2) - (np.sum(x1)*np.sum(x2))/n
    X1_y = np.sum(x1_y) - (np.sum(x1)*np.sum(y))/n
    X2_y = np.sum(x2_y) - (np.sum(x2)*np.sum(y))/n

    b1 = ((X2_square*X1_y) - (X1_X2*X2_y))/(X1_square*X2_square - X1_X2*X1_X2)
    b2 = ((X1_square*X2_y) - (X1_X2*X1_y))/(X1_square*X2_square - X1_X2*X1_X2)

    a = np.mean(y) - b1* np.mean(x1) - b2* np.mean(x2)

    return(round(a, 4), round(b1, 4), round(b2, 4))
```

In [6]:

```
# x_new_1 = np.array([3,4,5,6,2])
# x_new_2 = np.array([8,5,7,3,1])
# y_new = np.array([-3.7,3.5,2.5,11.5,5.7])

x_new_1 = np.array([60,62,67,70,71,72,75,78])
x_new_2 = np.array([22,25,24,20,15,14,14,11])
y_new = np.array([140,155,159,179,192,200,212,215])

a,b1,b2 = multiple_regression(x_new_1, x_new_2, y_new)
print("Y = "+str(a)+" + ("+str(b1)+")*X1 "+" + ("+str(b2)+")*X2 ")
```

```
Y = -6.8675 + (3.1479)*X1  + (-1.6561)*X2
```