

In [27]:



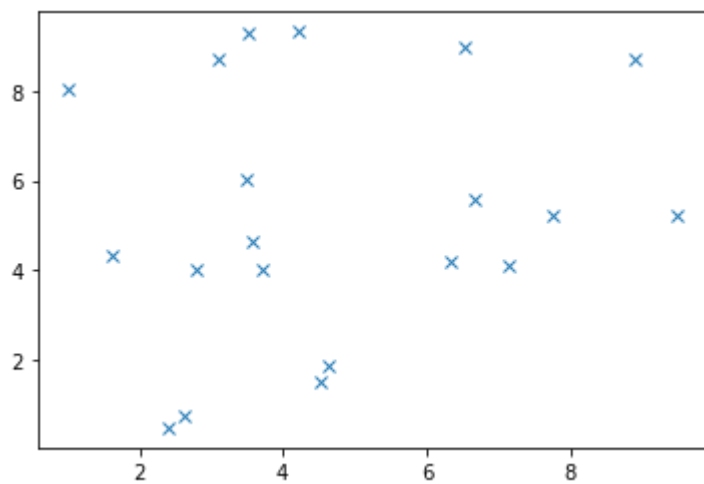
```
1 import numpy as np #Librería numérica
2 import matplotlib.pyplot as plt
3 from sklearn import linear_model #Regresión Lineal con scikit-learn
4
```

In [64]:



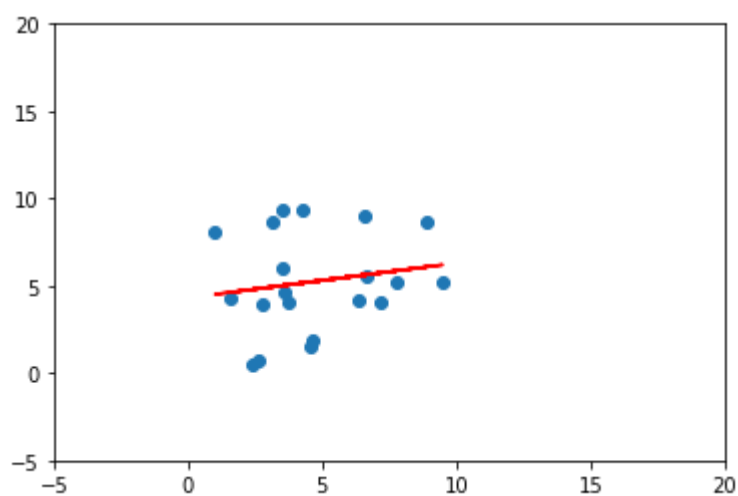
```
1 def generar_datos():
2     x = np.random.random(20)*10
3     y = np.random.random(20)*10
4     #return x,y
5     return x.reshape((20,1)), y.reshape((20,1))
6 x,y = generar_datos()
7
8 #print(x,y)
9 #print('-----')
10 plt.plot(x, y, 'x')
11 plt.show()
12
13
14 modelo = linear_model.LinearRegression()
15 modelo.fit(x, y)
16 y_pred = modelo.predict(x)
17
18
19 #VALOR DE PENDIENTE
20 print(modelo.coef_[0])
21
22 #VALOR DE B
23 b = -modelo.coef_[0][0]*x[0]+y_pred[0]
24
25
26 plt.scatter(x, y)
27 plt.xlim([-5, 20])
28 plt.ylim([-5, 20])
29 plt.plot(x, y_pred, color='red')
30 x_real = np.array([0, 100])
31 plt.show()
32
33 #fu=lambda x: regresion_lineal.coef_*x+regresion_lineal.intercept_
34 if (b < 0):
35     ecua = 'y = {}x {}'
36 else:
37     ecua = 'y = {}x + {}'
38
39 print('LA ECUACION DE LA RECTA ES')
40 print(ecua.format(modelo.coef_[0][0],b[0]))
41
42
43
44 numero = input('Ingrese el numero que desee predecir')
45 nuevo_x = np.array([int(numero)])
46 prediccion = modelo.predict(nuevo_x.reshape(-1,1))
47 print("LA PREDICCION ES IGUAL A: " , prediccion[0])
```





[0.19908261]

b es igual a: [4.31085698]



LA ECUACION DE LA RECTA ES

$y = 0.19908260713335849x + 4.310856984140261$

Ingrese el numero que desee predecir

LA PREDICCIÓN ES IGUAL A: [8.29250913]

In [ ]:



1