

## ▼ Anexo 1:

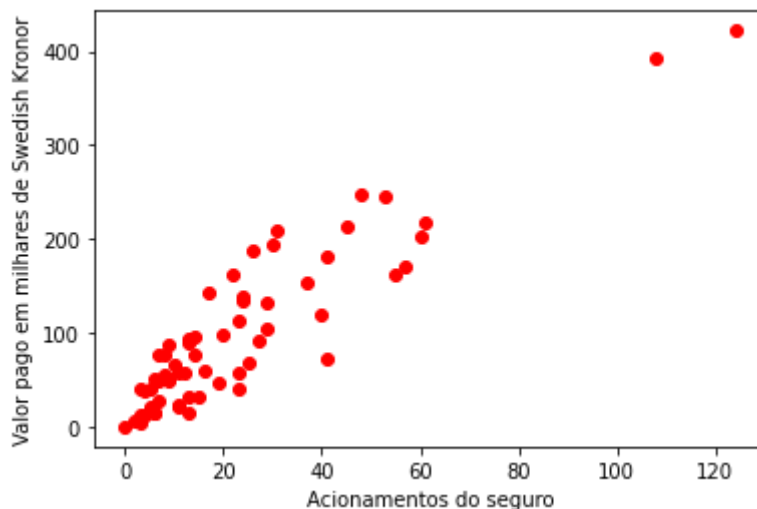
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def produto_interno(array1, array2):
    return sum([a * b for a,b in zip(array1, array2)])

data = pd.read_csv("dataComplete.csv")
values = data.values
x = values[:,0]
y = values[:,1]

plt.plot(x,y, 'ro')
plt.xlabel("Acionamentos do seguro")
plt.ylabel("Valor pago em milhares de Swedish Kronor")

plt.text(0, 0.5, 'Valor pago em milhares de Swedish Kronor')
```



```
a00 = produto_interno(np.ones(63), np.ones(63))
a01 = produto_interno(np.ones(63), x)
a10 = produto_interno(x, np.ones(63))
a11 = produto_interno(x, x)
f1 = produto_interno(y, np.ones(63))
fx = produto_interno(y, x)

print("Dados:")
print("<1,1>:", a00)
print("<1,x>:", a01)
print("<x,1>:", a10)
print("<x,x>:", a11)
print("<f,1>:", f1)
print("<f,x>:", fx)
```



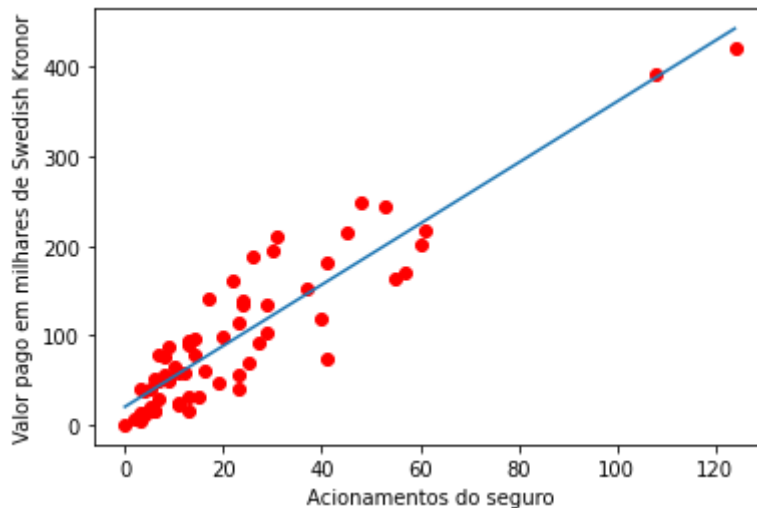
Dados:

```
<1,1>: 63.0
<1,x>: 1443.0
<x,1>: 1443.0
<x,x>: 66861.0
<f,1>: 6185.800000000001
```

```
a = np.array([[a00,a01],[a10,a11]])
b = np.array([f1,fx])
res = np.linalg.solve(a,b)
print("coeficiente linear:", res[0])
print("coeficiente angular:", res[1])
print()
eixo_x = np.arange(0,max(x),.1)
curve = lambda n: res[1]*n + res[0]
plt.plot(x,y, 'ro')
plt.plot(eixo_x, curve(eixo_x))
plt.xlabel("Acionamentos do seguro")
plt.ylabel("Valor pago em milhares de Swedish Kronor")
```

```
↪ coeficiente linear: 19.9944857591148
coeficiente angular: 3.413823560066368
```

```
Text(0, 0.5, 'Valor pago em milhares de Swedish Kronor')
```



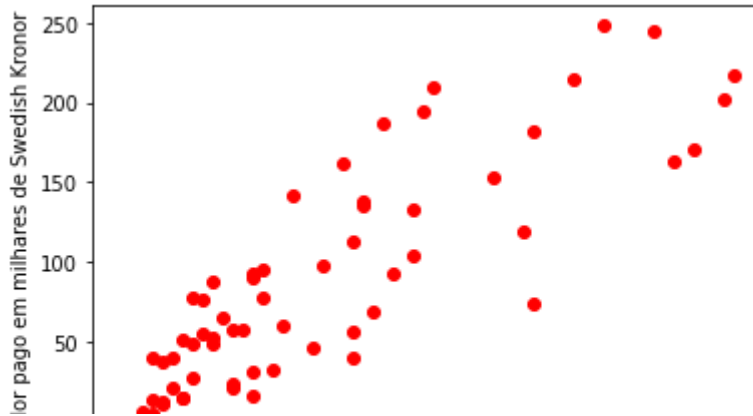
Podemos perceber que existem dois pontos distantes dos demais, visando analisar a influencia desses pontos no calculo do mqm abaixo temos o mesmo calculo mas sem esses dois pontos

```
data = pd.read_csv("data.csv")
values = data.values
x = values[:,0]
y = values[:,1]

plt.plot(x,y, 'ro')
plt.xlabel("Acionamentos do seguro")
plt.ylabel("Valor pago em milhares de Swedish Kronor")
```

```
↪
```

Text(0, 0.5, 'Valor pago em milhares de Swedish Kronor')



```
a00 = produto_interno(np.ones(63), np.ones(63))
a01 = produto_interno(np.ones(63), x)
a10 = produto_interno(x, np.ones(63))
a11 = produto_interno(x, x)
f1 = produto_interno(y, np.ones(63))
fx = produto_interno(y, x)
```

```
print("Dados:")
print("<1,1>:", a00)
print("<1,x>:", a01)
print("<x,1>:", a10)
print("<x,x>:", a11)
print("<f,1>:", f1)
print("<f,x>:", fx)
```

```
↳ Dados:
<1,1>: 63.0
<1,x>: 1211.0
<x,1>: 1211.0
<x,x>: 39821.0
<f,1>: 5371.1
<f,x>: 162360.9
```

```
a = np.array([[a00, a01], [a10, a11]])
b = np.array([f1, fx])
res = np.linalg.solve(a, b)
print("coeficiente linear:", res[0])
print("coeficiente angular:", res[1])
print()
eixo_x = np.arange(0, max(x), .1)
curve = lambda n: res[1]*n + res[0]
plt.plot(x, y, 'ro')
plt.plot(eixo_x, curve(eixo_x))
plt.xlabel("Acionamentos do seguro")
plt.ylabel("Valor pago em milhares de Swedish Kronor")
```

```
↳
```

coeficiente linear: 16.564469459855225  
coeficiente angular: 3.5735247101809424

Text(0, 0.5, 'Valor pago em milhares de Swedish Kronor')

