T3.1 / N2 e Presença em aula 24

Atividade de Presença para Probest.

Para nossa análise, X são os casos e Y são as mortes.

Primeiramente iremos pegar os dados das 20 primeiras semanas epidemiológica.

In [33]:
```python
import numpy as np
```

In [3]:
```python
import pandas as pd
dados = pd.read_csv('./Dados semana 1 a 20 - Covid 2021 - Página1.csv')
```

Amostragem dos 5 primeiros dados:

In [4]:
```python
dados.head()
```

Out[4]:

	Semana	Casos	Obitos
0	1	359593	6906
1	2	379061	6665
2	3	361195	7149
3	4	360721	7500
4	5	320820	7067

2: Encontre a reta de regressão linear Y em função de X (para as semanas ímpares das 20 semanas), se ela existir.

Vamos primeiramente pegar os dados das semanas ímpares.

In [5]:
```python
impares = dados.loc[dados.Semana % 2 != 0]
# 5 primeiros dados.
impares.head()
```

Out[5]:

	Semana	Casos	Obitos
0	1	359593	6906
2	3	361195	7149
4	5	320820	7067
6	7	329394	7445
8	9	421604	10104

Temos que encontrar a fórmula da regressão: Ya = a + b*X

Onde:
a = (média de Y) - b * (média de X)
b = Covariância(X,Y) / (desvio padrão(X))^2

Vamos descobrir B:

B = Covariância(X,Y) / (desvio padrão(X))^2

Como todos os dados são diferentes, a esperança de X e Y é igual à média aritimética.

Covâriancia (X, Y):

Cov(X,Y) = E(XY) - [E(X) E(Y)]

In [6]:
```python
# E (X), onde X são os casos.
E_X = impares.Casos.sum() / len(impares.Casos)
print('Esperança/Média de  X (Casos)', E_X)
```

```
Esperança/Média de  X (Casos) 408086.5
```

In [7]:
```python
# E(Y), onde Y são os Obitos
E_Y = impares.Obitos.sum() / len(impares.Obitos)
print('Esperança/Média de  Y (Óbitos)', E_Y)
```

```
Esperança/Média de  Y (Óbitos) 12466.3
```

In [8]:
```python
# E(X*Y)
X_Y = impares.Casos * impares.Obitos
print("Valores de X*Y:", X_Y.values)
E_X_Y = X_Y.sum() / len(X_Y)
print('\nEsperança/Média de E*Y:', E_X_Y)
```

```
Valores de X*Y: [2483349258 2582183055 2267234940 2452338330 4259886816 8010977364
 9099325105 9258249240 7078943200 5904336345]

Esperança/Média de E*Y: 5339682365.3
```

Então temos:

In [9]:
```python
cov_X_Y = E_X_Y - (E_X * E_Y)
print('Covâriancia de X e Y:', E_X_Y)
```

```
Covâriancia de X e Y: 5339682365.3
```

(Desvio Padrão(X))^2

Desvio Padrão(X) = Raiz de Variância (X) = raiz de E(X^2) - E(X)^2
Logo, (Desvio Padrão(X))^2 = (Raiz de Variância (X))^2 = Variância(X) = E(X^2) - E(X)^2

In [10]:
```python
# Calcular E(X²), onde X são os casos.
x_ao_quadrado = impares.Casos**2
print('Valores de X²:', x_ao_quadrado.values)
E_X_2 = x_ao_quadrado.sum() / len(x_ao_quadrado)
print('\n E(X²) = ', E_X_2)
```

```
Valores de X²: [129307125649 130461828025 102925472400 108500407236 177749932816
 261656802576 214586665225 207102357225 174523417600 194176829025]

 E(X²) =  170099083777.7
```

In [11]:
```python
from math import sqrt
# Variância = E(x²) - E(x)²
var_X = E_X_2 - (E_X**2)
print('Variância de X:', var_X)
dp_X = sqrt(var_X)
print('Desvio padrão de X:', dp_X)
```

```
Variância de X: 3564492295.450012
Desvio padrão de X: 59703.36921355454
```

Agora podemos calcular: B = Covariância(X,Y) / (desvio padrão(X))^2

In [12]:
```python
B = cov_X_Y / (dp_X**2)
print('Temos B:', B)
```

```
Temos B: 0.07079651446353907
```

Agora calculamos A: a = (média de Y) - b * (média de X)

In [13]:
```python
A = E_Y - (B * E_X)
print('Valor de A:', A)
```

```
Valor de A: -16424.80179962504
```

Por fim, a fórmula da reta: Ya = a + b*X*

Temos: ***Y = -16424.80 + 0.070*** **X**

In [14]:
```python
# Definição da nossa formula Y = A + B*X
def previsao(X, a = A, b = B):
    Y = a + b * X
    return Y
```

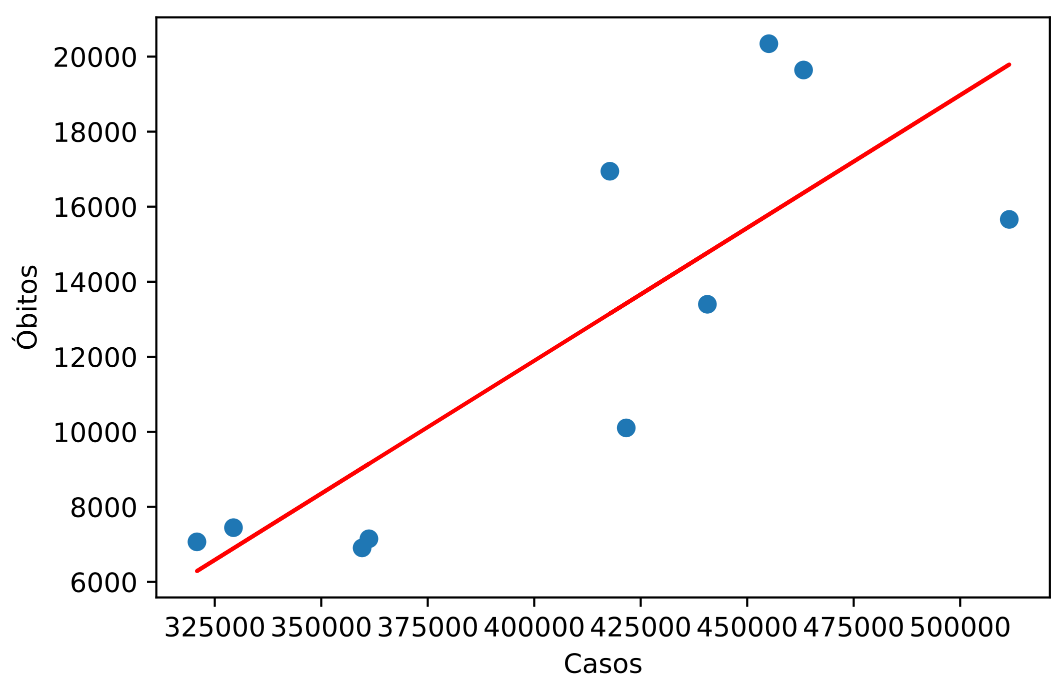
In [15]:
```python
# Prever usando a regressão linear.
previsao(X=450_000)
```

Out[15]:
```
15433.629708967546
```

Vamos agora plotar os dados em um Gráfico de dispersão e a Reta de Regressão

In [16]:
```python
import matplotlib.pyplot as plt
plt.scatter(impares.Casos, impares.Obitos)
plt.plot(impares.Casos,[previsao(i) for i in impares.Casos], color='red')
plt.ylabel('Óbitos')
plt.xlabel('Casos')
```

Out[16]:
```
Text(0.5, 0, 'Casos')
```



Coeficiente de Determinação

Agora vamos calcular o Coeficiente de Determinação

R^2 = Soma(Yia - E(Y))^2 / Soma(Yi - E(Y))^2

In [62]:
```python
previsoes = [previsao(i) for i in impares.Casos]
previsoes_soma_2 = [(p - E_Y)**2 for p in previsoes]

real = impares.Obitos.values
real_soma_2 = [(r - E_Y)**2 for r in real]
```

In [66]:
```python
R_2 = sum(previsoes_soma_2) / sum(real_soma_2)
R_2
```

Out[66]:
```
0.6771566077941553
```

In [54]:
```python
R_2 = soma_prevista / soma_real
print('Coeficiente de Determinação:', R_2)
```

```
Coeficiente de Determinação: 0.6771566077941553
```

In [55]:
```python
from sklearn.linear_model import LinearRegression
model = LinearRegression()
x_li = np.array(impares.Casos).reshape(-1, 1)
model.fit(x_li, impares.Obitos)
model.predict([[420_000]])
```

Out[55]:
```
array([13309.73427506])
```

In [56]:
```python
```

Out[56]:
```
array([0.07079651])
```

In [ ]:
```python
```