prova_1_questão_1

Questão 1

Tabela 1: Dados da Célula de Carga para Compressão

	Ida		Volta
Massa	Tensão [V]	Desvio [m V]	Tensão [V]
0	-35,1805*10^-3	0.2613	-33,2095*10^-3
0,490	-101,4555*10^-3	0.3293	-100,2115*10^-3
1,474	-0,237981	0.3854	-0,243285
2,457	-0,377301	0.2511	-0,377443
4,439	-0,656438	0.2431	-0,659317
6,959	-1,008969	0.2868	-1,008895
11,997	**-1,71**	0.4049	**-1,71**

Tabela 2: Dados da Célula de Carga para Tração

	Ida		Volta	
Massa	Tensão [V]	Desvio [mV]	Tensão [V]	Desvio [mV]
0,580	76,6392*10^-3	0.2842	73,1370*10^-3	0.9575
1,070	126,1190*10^-3	0.3573	128,3680*10^-3	0.5794
2,054	0,262307	0.3499	0,266097	0.6381
3,040	0,399873	0.3722	0,405097	0.4810
5,022	0,679279	0.4626	0,680896	0.3675
7,542	1,032813	0.2922	1,033632	0.5635
12,58	1,738543	0.3633	1,738543	0.3633

A calibração direta do sistema de medição piezoresistivo Kratos utiliza um conjunto de massas padrão. A célula de carga foi ligada no condicionador de sinais, a fonte DC foi ajustada e ligada no condicionador de sinais (para a alimentação dos strain gages da célula de carga) e o sinal do condicionador foi conectado no multímetro HP. A célula de carga foi calibrada para compressão e tração, de forma a verificar sua histerese. Os dados de compressão estão apresentados na Tabela 1 enquanto os dados de tração estão na Tabela 2. Pede-se:

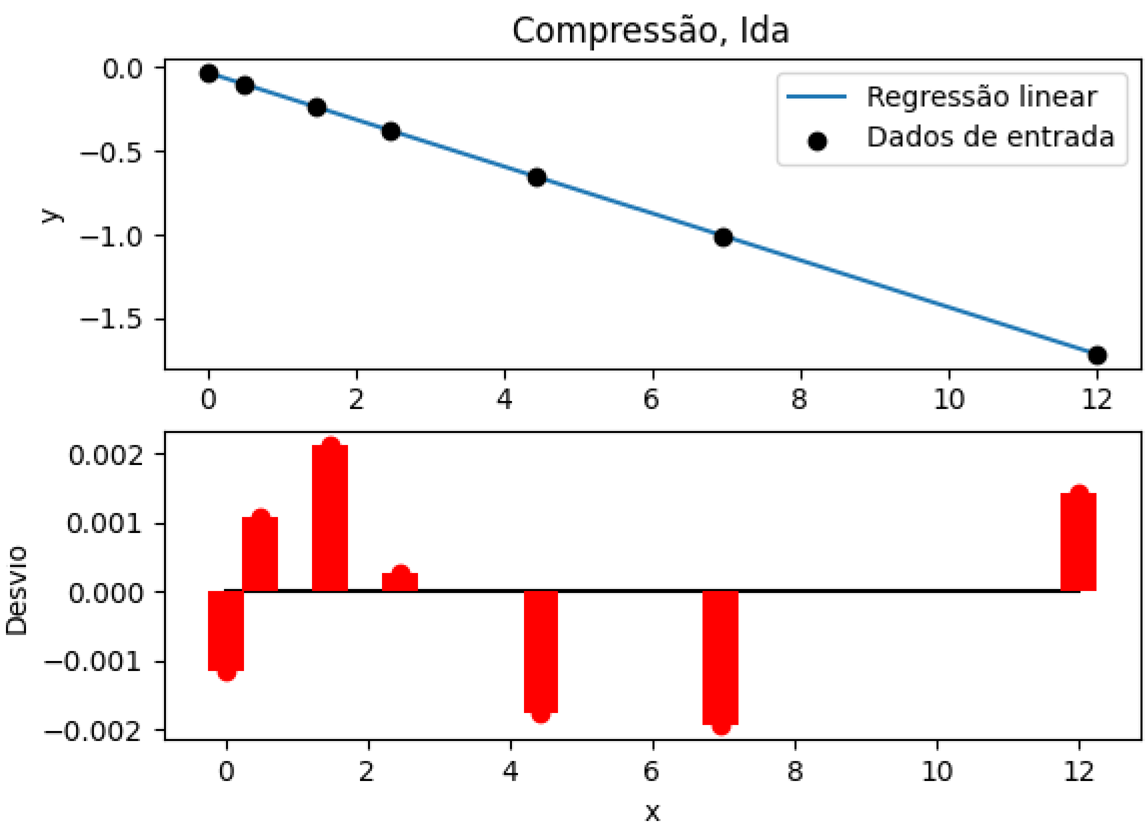
a) Apresentar os resultado da regressão linear para dos dados de carga e descarga; (5 pontos)

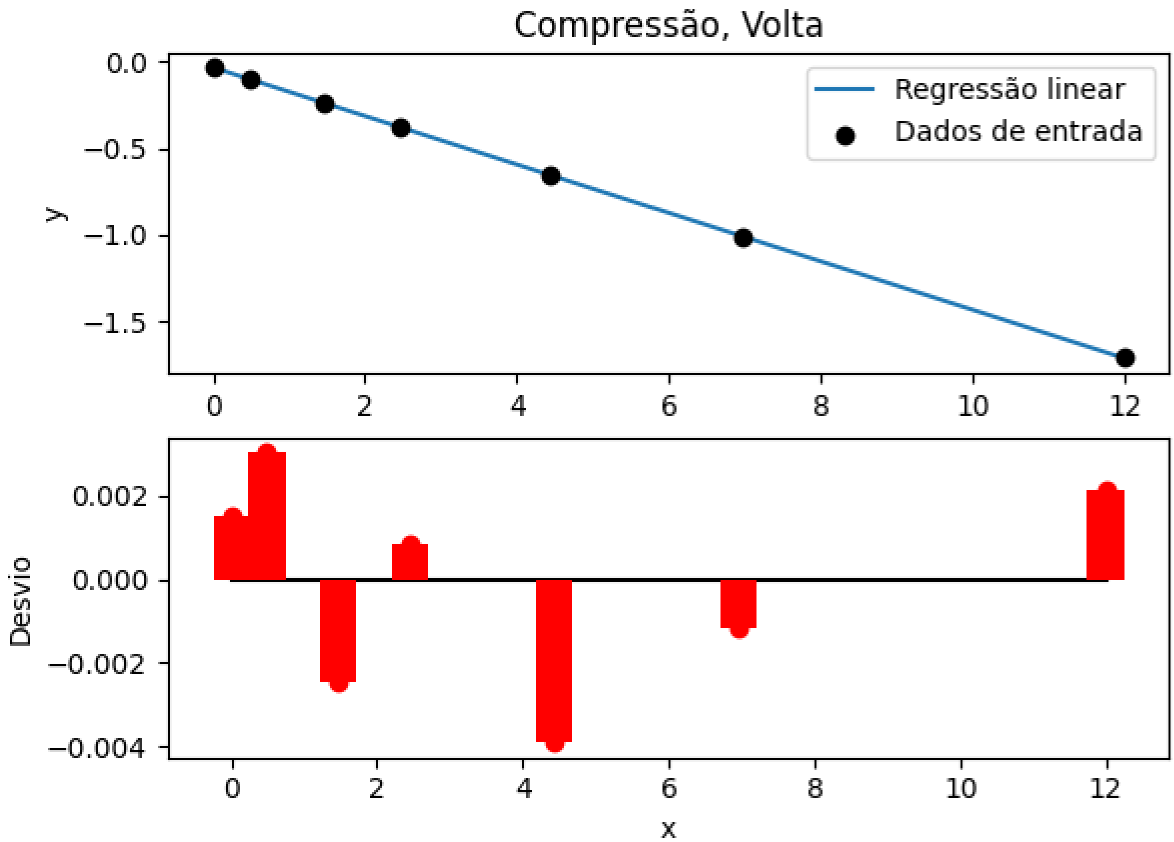
Usou-se o método de regressão linear em mínimos quadrados implementado em python para cada caso:

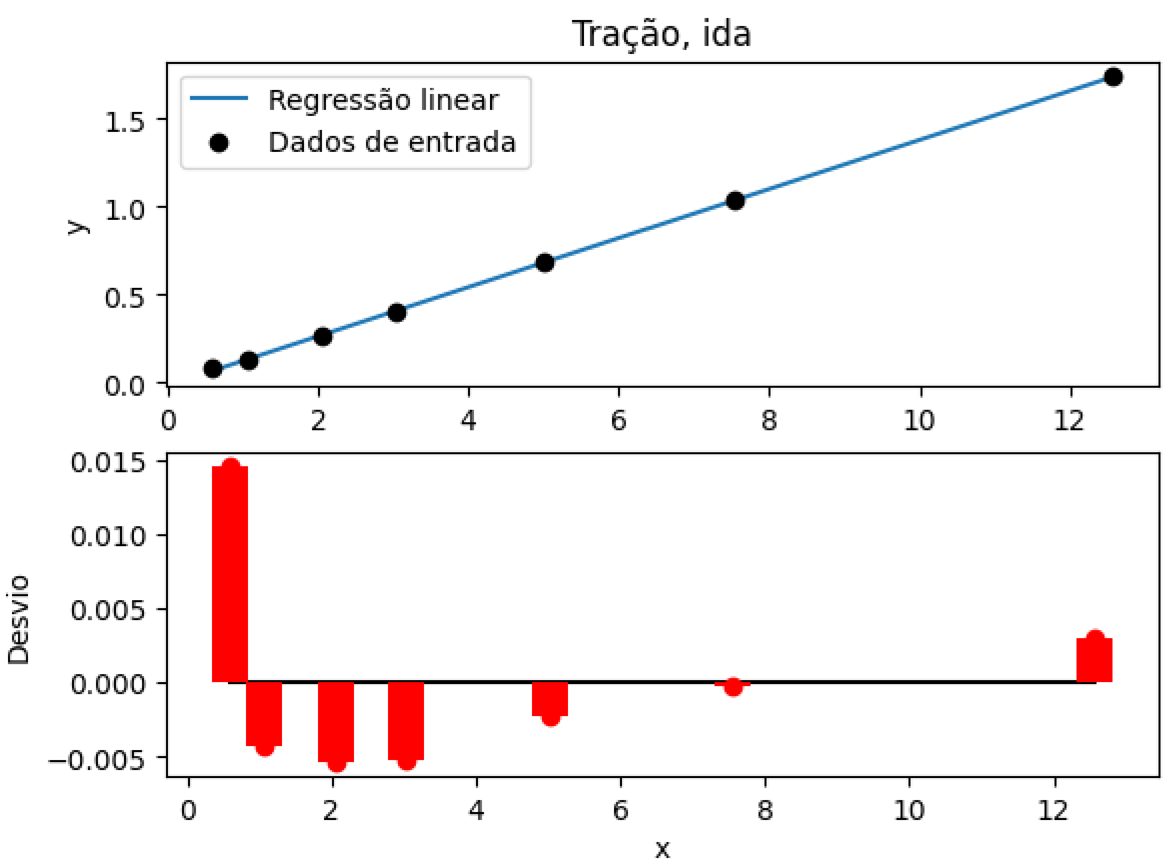
```python
def less_square_linear_reduction(vector_x, vector_y, show_graph):
    '''
    Executa a regressão linear em mínimos quadrados dos dois vetores de entrac
    '''

    # Initializing some local variables
    x_sum = 0
    y_sum = 0
    square_x_sum = 0
    square_y_sum = 0
    x_prod = 1
    y_prod = 1
    cross_prod_sum = 0
    length = len(vector_x)

    # Checking length from both input vectors
    if len(vector_y) != length :
        print("[ERROR] - Vectors must have same length")
        return

    # Calculating some usefull measures
    for i in enumerate(vector_x):
        x_sum += vector_x[i[0]]
        y_sum += vector_y[i[0]]
        square_x_sum += vector_x[i[0]]**2
        square_y_sum += vector_y[i[0]]**2
        x_prod = x_prod * vector_x[i[0]]
        y_prod = y_prod * vector_y[i[0]]
        cross_prod_sum += vector_x[i[0]] * vector_y[i[0]]

    # Determining angular and scalar constants for the linear regression.
    a_constant = (length * cross_prod_sum - x_sum * y_sum)/(length * square_x_
                                              x_sum**2)
    b_constant = (y_sum * square_x_sum - cross_prod_sum * x_sum)/(length * squ
                                                    x_sum**2)

    # Generating graph of the linear regression
    x_result = np.linspace(vector_x[0], vector_x[length - 1], 100)
    y_result = a_constant * x_result + b_constant

    # calculating losses
    losses = [vector_y[i] - (a_constant * vector_x[i] + b_constant) for i
              in range(length)]

    # Ploting results if convenient
    if show_graph:
        plt.figure()
        plt.subplot(211)
        plt.plot(x_result, y_result, label ="Regressão linear")
```

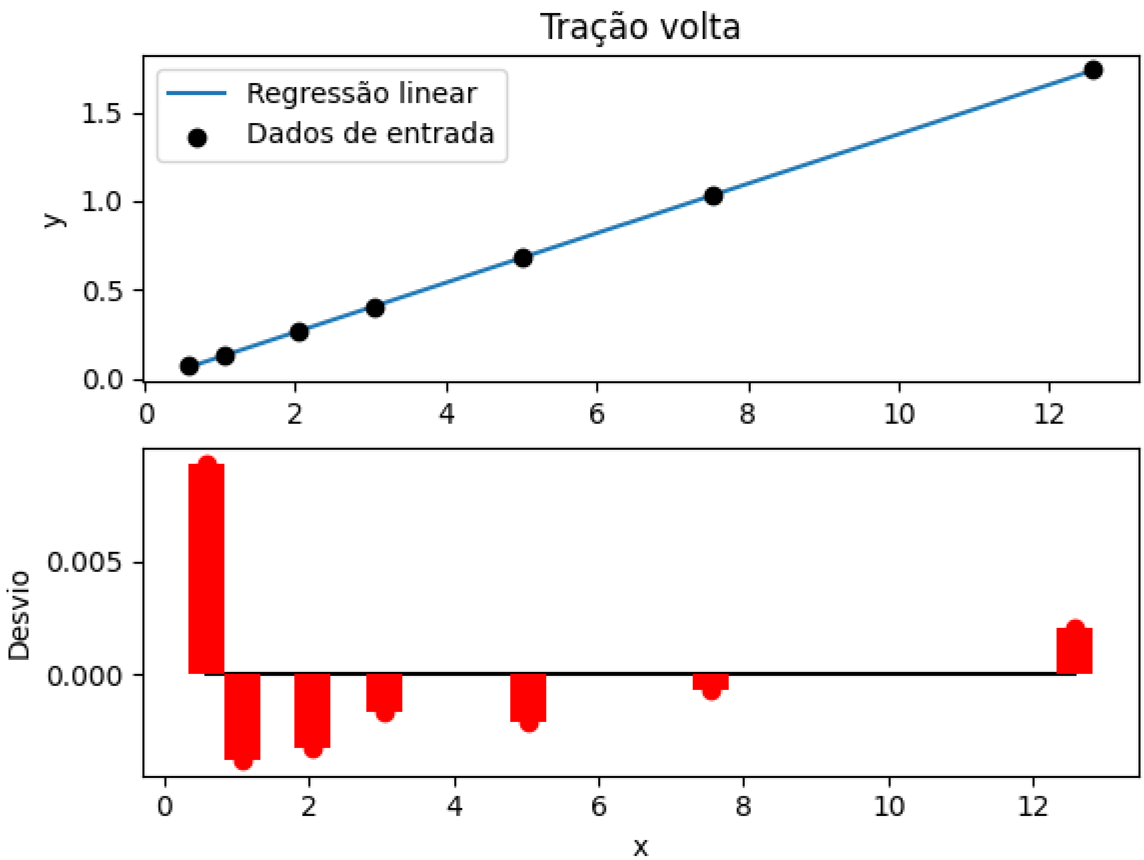
```
        plt.plot(x_result, y_result, label = "Regressão linear")
        plt.scatter(vector_x, vector_y, c="black", label = "Dados de entrada",
                    zorder=10)
        plt.xlabel("x")
        plt.ylabel("y")
        plt.legend()
        plt.subplot(212)
        plt.plot([vector_x[0], vector_x[len(vector_x) - 1]], [0, 0], c="black'
                 zorder = -1)
        plt.bar(vector_x, losses, width=2, color="red")
        plt.scatter(vector_x, losses, c="red")
        plt.xlabel("x")
        plt.ylabel("Desvio")
        plt.show()
        print("Linear regrecion ==> f(x) = {:.3f} x + {:.3f}".format(a_constar
    return [a_constant, b_constant, losses]
```

Os resultados seguem:




Compressão, Volta
Regressão linear
Dados de entrada
y
Desvio
x
0.0
−0.5
−1.0
−1.5
0.002
0.000
−0.002
−0.004
0
2
4
6
8
10
12

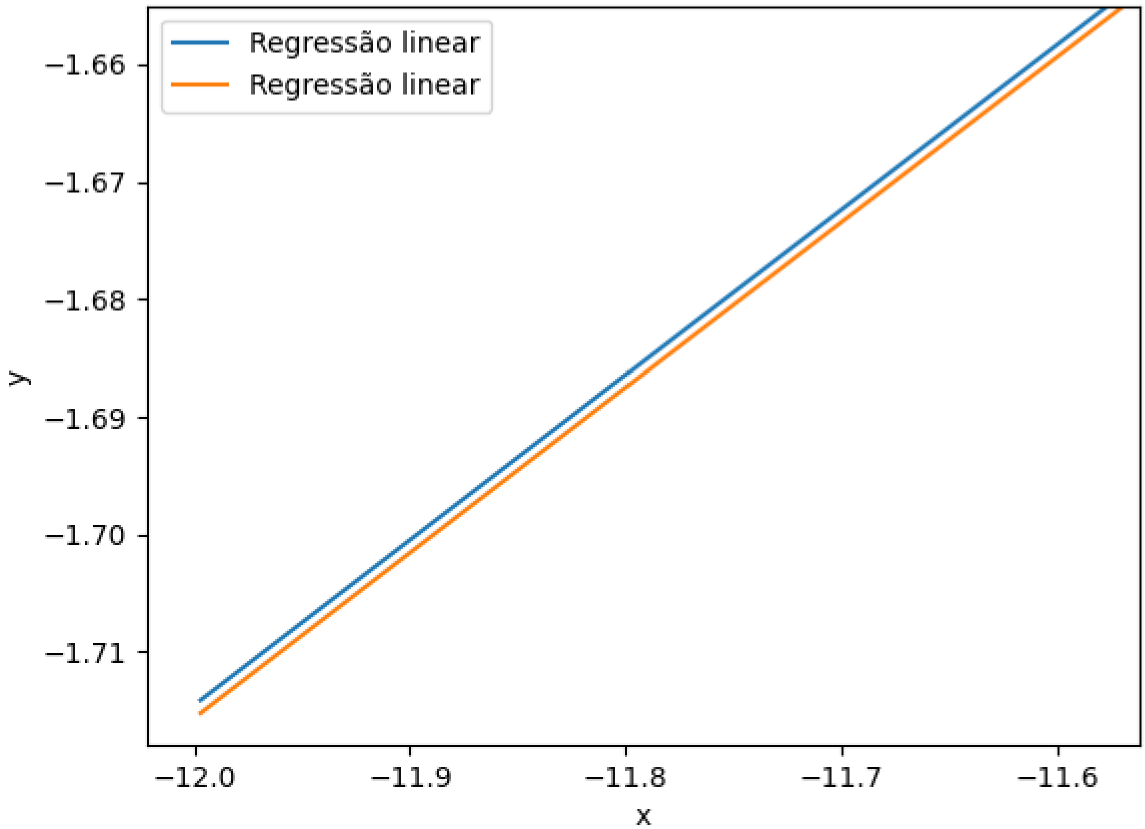

Tração, ida
Regressão linear
Dados de entrada
y
Desvio
x
1.5
1.0
0.5
0.0
0.015
0.010
0.005
0.000
−0.005
0
2
4
6
8
10
12



b) Mostrar graficamente a histerese da célula de carga; (5 pontos)

Como podemos ver no gráfico adiante, a regressão linear da ida e da vinda possui um offset que pode ser visto pela diferença no coeficiente linear de ambas as curvas:

- -0.02744
- -0.02711



c) Qual é a sensibilidade estática da célula de carga para tração, compressão e média? (5 pontos)

A sensibilidade é igual ao coeficiente linear da equação de ajuste.

- Para o caso da compressão, foi obtido um valor de 0.140.
- Para o caso da tração, foi obtido um valor de 0.139.
- Assim, obteve-se o valor médio de 0.1395