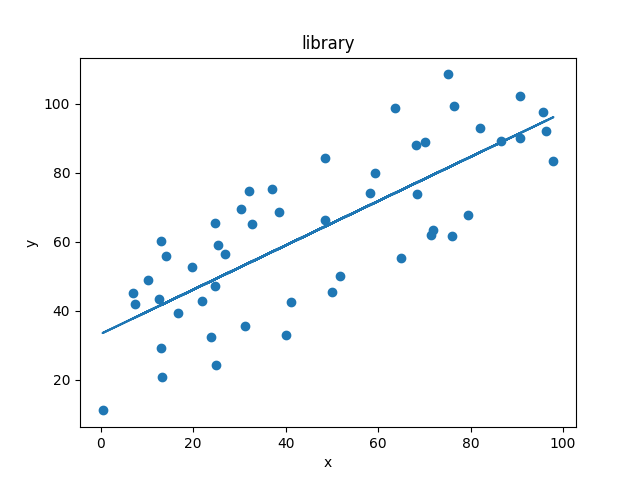
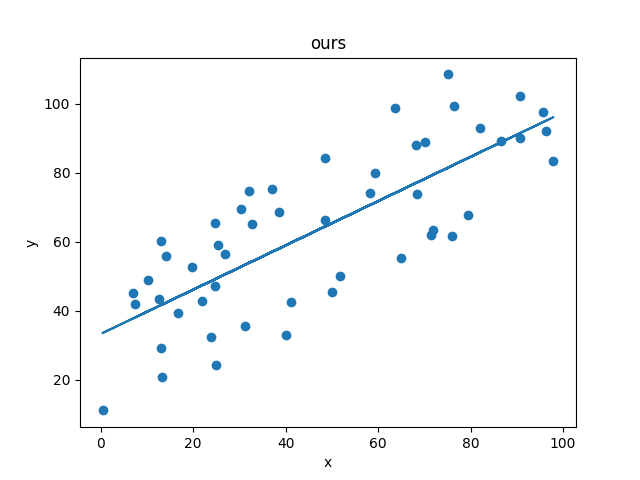
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn import linear\_model  
  
# getting the data  
data = pd.read\_excel("data.xlsx")  
x = data.values[1:, 1]  
y = data.values[1:, 2]  
  
# the linear regression from sklearn  
reg = linear\_model.LinearRegression()  
reg.fit(np.reshape(x, (-1, 1)), y)  
slope1, intercept1 = reg.coef\_, reg.intercept\_  
  
# drawing the graph  
plt.title("library")  
plt.xlabel("x")  
plt.ylabel("y")  
plt.scatter(x, y)  
plt.plot(x, slope1 \* x + intercept1)  
plt.show()  
  
# the linear regression we created  
slope2 = 0  
intercept2 = 0  
lr = 0.0005  
  
# training the dataset  
for i in range(100000):  
 sum1 = 0  
 sum2 = 0  
 for j in range(len(x)):  
 sum1 += (slope2\*x[j] + intercept2) - y[j]  
 sum2 += ((slope2\*x[j] + intercept2) - y[j]) \* x[j]  
 intercept2 = intercept2 - lr \* sum1/len(x)  
 slope2 = slope2 - lr \* sum2/len(x)  
  
# drawing the graph  
plt.title("ours")  
plt.xlabel("x")  
plt.ylabel("y")  
plt.scatter(x, y)  
plt.plot(x, slope2\*x+intercept2)  
plt.show()



עדיף להשתמש ברגרסיה הלינארית של הספרייה המוכנה משום שהוא נותן לנו מקדמים יותר מדויקים כך שהcost function תהיה קטנה יותר