Name: Long Nguyen
Student ID: 801235507
Homework #: 1

Using a dataset "D3.csv", let the first three columns of the data set be separate explanatory variables x1, x2, x3. Let the fourth column be the dependent variable Y.

Problem 1:

Develop a code that runs linear regression with gradient descent algorithm for each of the explanatory variables in isolation. In this case, assume that in each iteration, only one explanatory variable (either X1, or X2, or X3) is explaining the output. Basically, do three different training, one per each explanatory variable. For the learning rate, explore different values between 0.1 and 0.01. Initialize the parameters to zero (theta to zero).

Link to Github repository: https://github.com/lnguye782/ECGR-4105-Intro-to-ML/tree/main/HW1

1. Report the linear model for each explanatory variable.

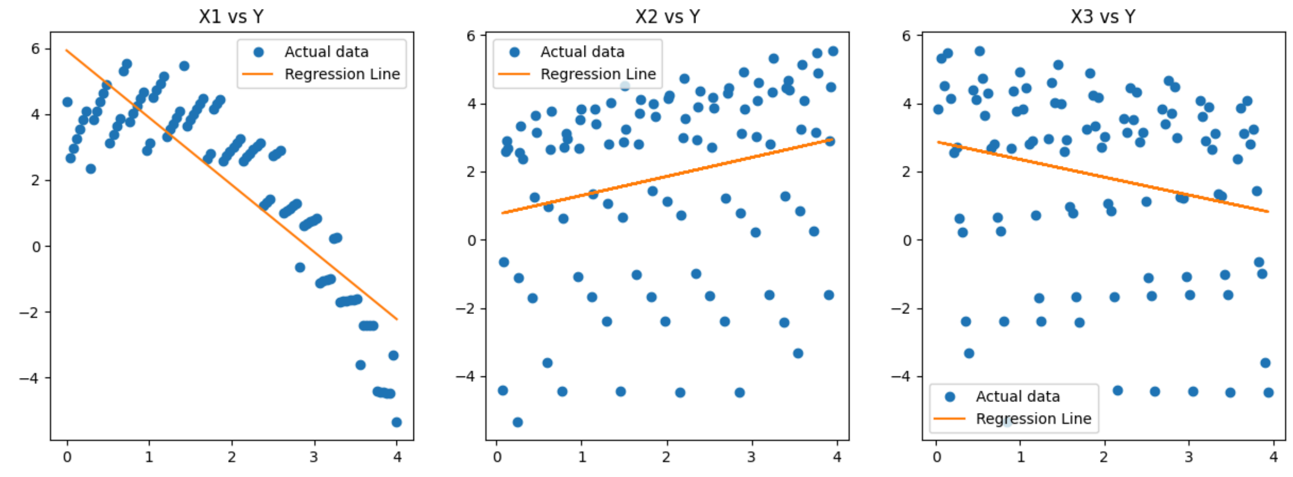
```
{'X1': (-2.038336633229477, 5.927948916979075 6),
 'X2': (0.5576076103651677, 0.7360604300111252),
 'X3': (-0.5204828841600003, 2.8714221036339524)}
```

X1: $Y = 5.928 - 2.038 \cdot X1$

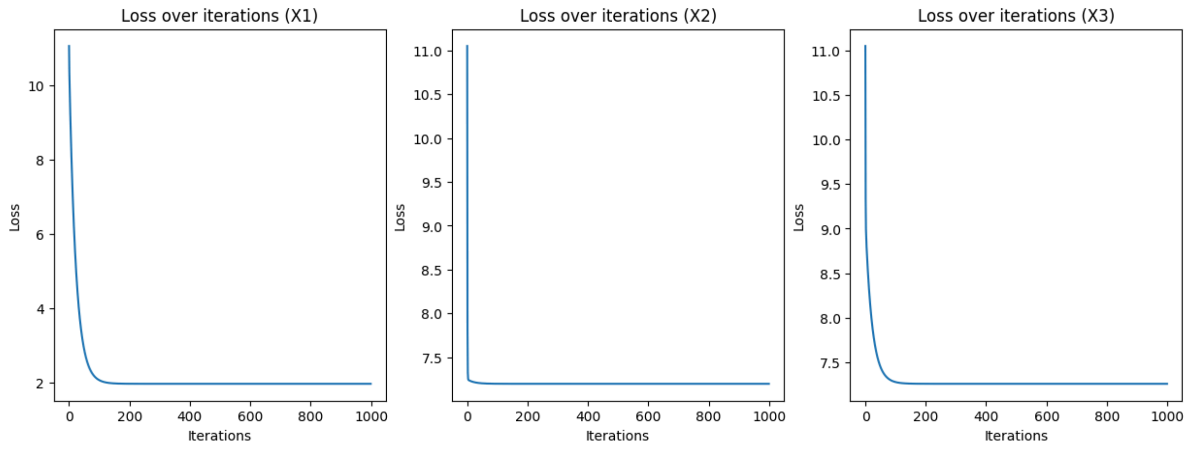
X2: $Y = 0.736 + 0.558 \cdot X2$

X3: $Y = 2.871 - 0.520 \cdot X3$

2. Plot the final regression model and loss over the iteration per each explanatory variable.



Name: Long Nguyen
Student ID: 801235507
Homework #: 1



3. Which explanatory variable has the lower loss (cost) for explaining the output (Y)?

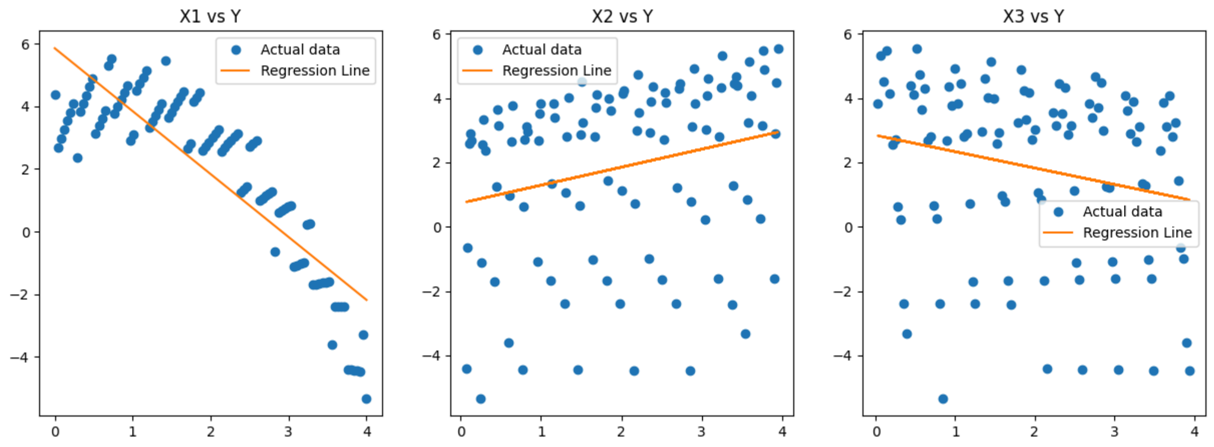
```
{'X1': 1.9699861650811892, 'X2': 7.198732036336083, 'X3': 7.258902249215831}
```

X1 has the lowest final loss (cost), with a value of 1.970, for explaining the output (Y).

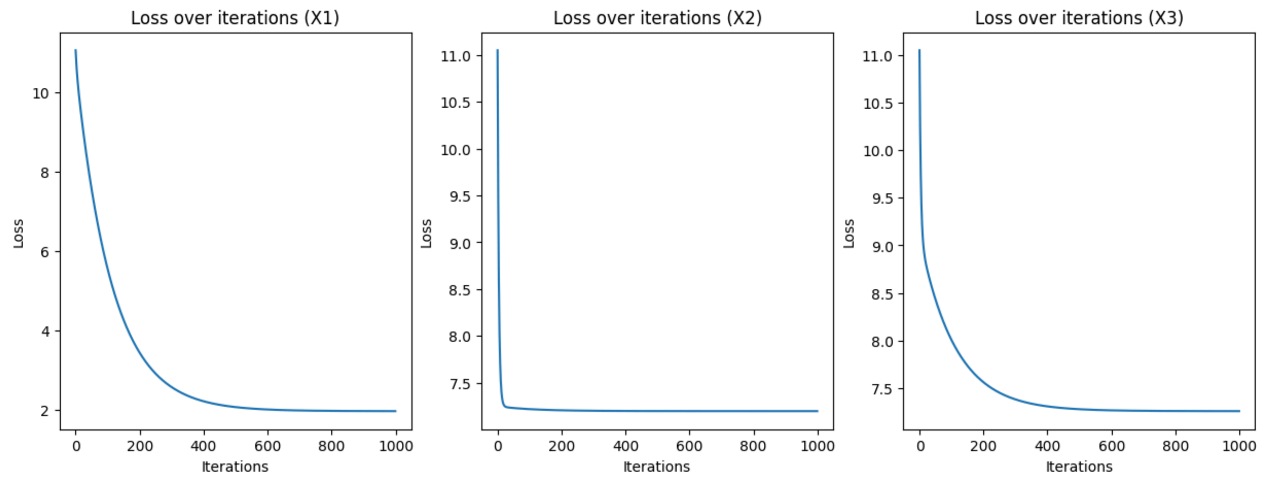
4. Based on the training observations, describe the impact of the different learning rates on the final loss and number of training iterations.

Learning rate = 0.01:

```
{'X1': (-2.0115825039221686, 5.859208469587321),
 'X2': (0.5596740020661444, 0.7307512257200968),
 'X3': (-0.5088702809869871, 2.8420598533051726)}
```



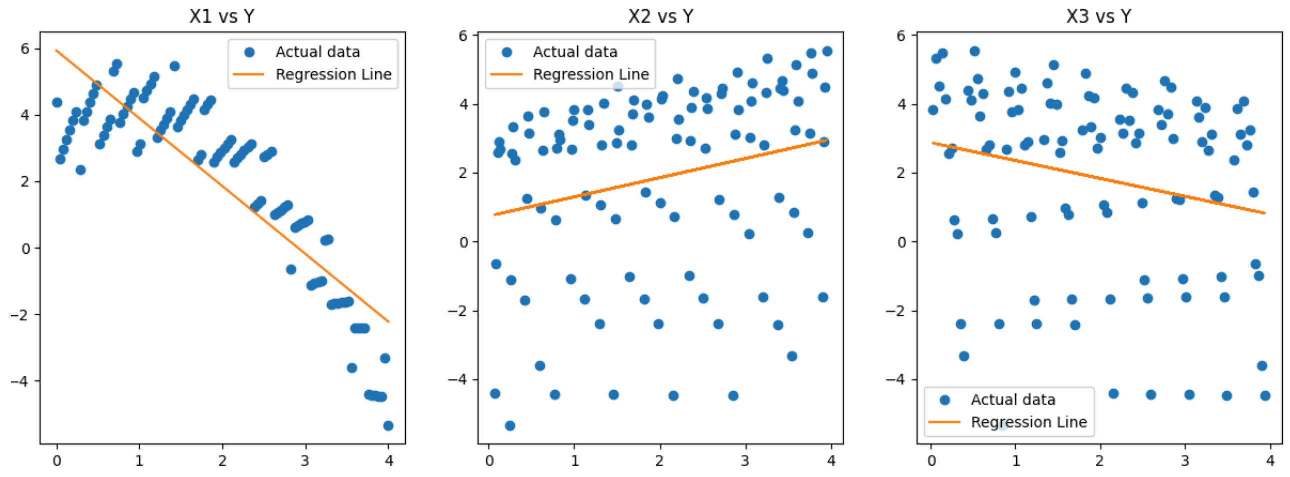
Name: Long Nguyen
Student ID: 801235507
Homework #: 1



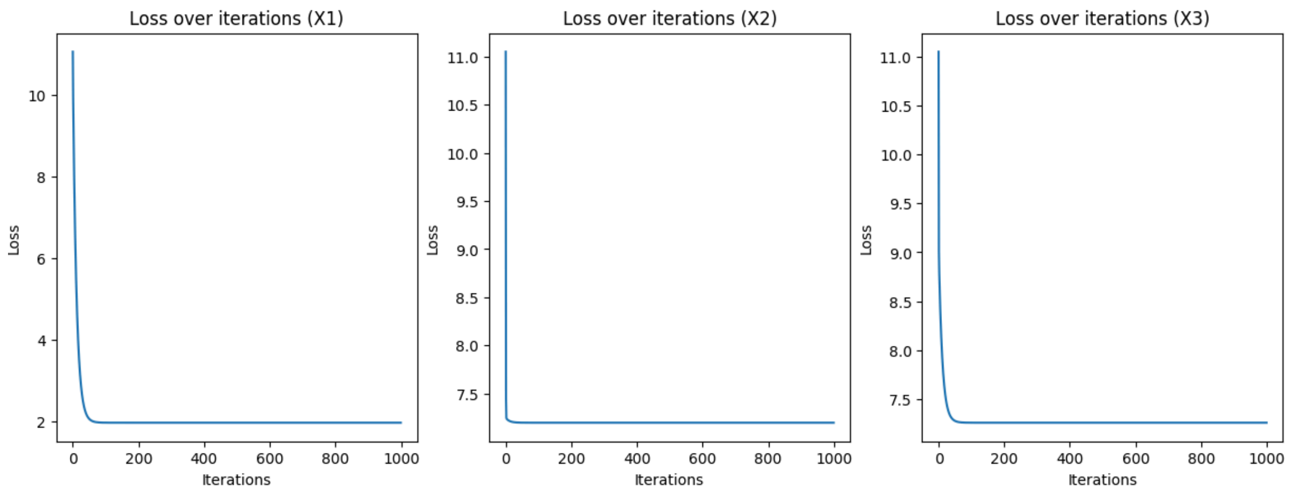
```
{'X1': 1.9712026010746704, 'X2': 7.198739292512929, 'X3': 7.25912843414523}
```

Learning rate = 0.1:

```
{'X1': (-2.0383366336508018, 5.927948918061602),
 'X2': (0.5576076103326035, 0.7360604300947927),
 'X3': (-0.520482884300106, 2.871422103988207)}
```



Name: Long Nguyen
Student ID: 801235507
Homework #: 1



```
{'X1': 1.9699861650811892, 'X2': 7.198732036336083, 'X3': 7.258902249215829}
```

The learning rate of 0.05 was effective in converging the loss. The learning rate that is too high, closer to 0.1, could cause oscillations or divergences in the loss. The learning rate that is lower, closer to 0.01, would require a lot more iterations to converge.

Problem 2:

This time, run linear regression with gradient descent algorithm using all three explanatory variables. For the learning rate, explore different values between 0.1 and 0.01. Initialize the parameters (theta to zero).

1. Report the best final linear model.

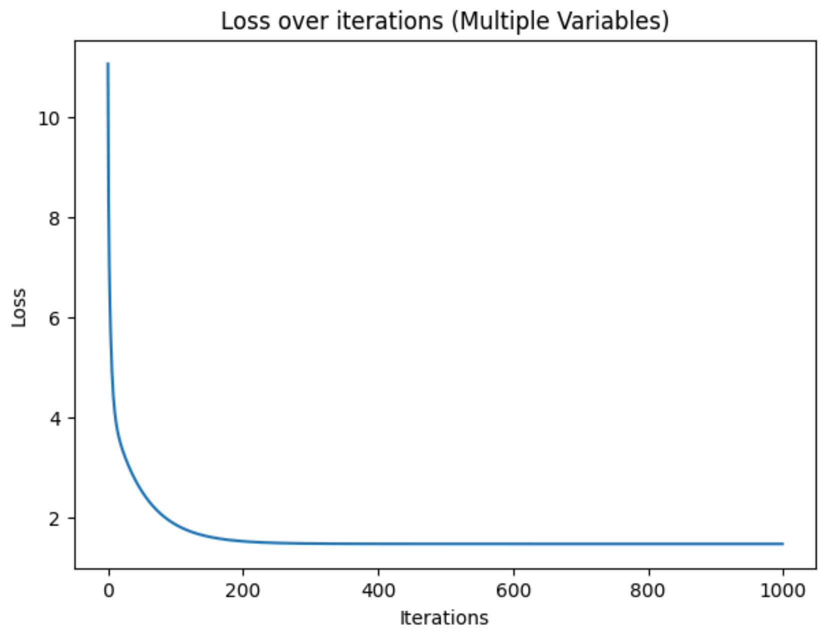
```
array([ 5.31393577, -2.00368658,  0.53260157, -0.26556795])
```

The best final linear model using all three explanatory variables:

$$Y = 5.314 - 2.004 \cdot X1 + 0.533 \cdot X2 - 0.266 \cdot X3$$

2. Plot loss over the iteration.

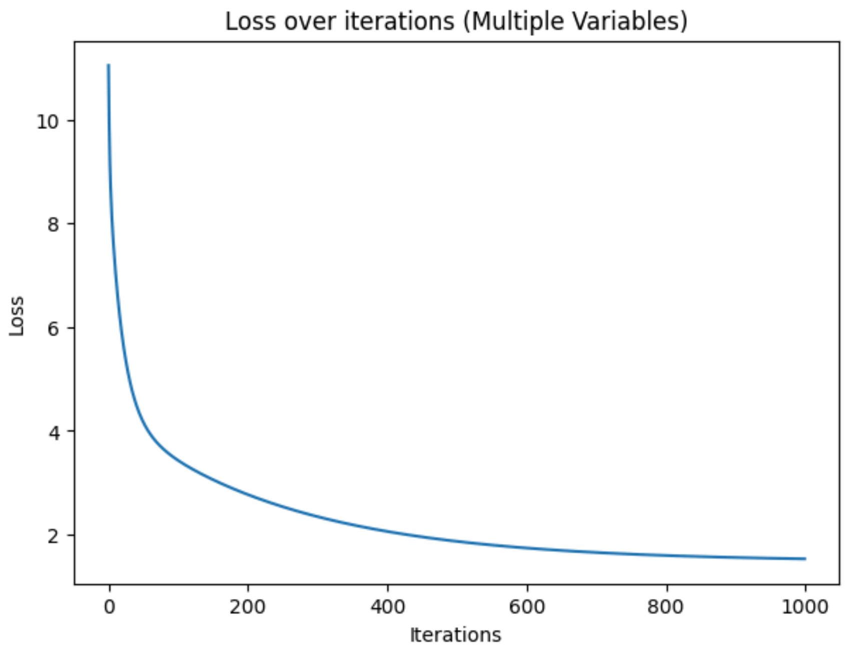
Learning rate = 0.05:



3. Based on training observations, describe the impact of the different learning rates on the final loss and number of training iterations.

Learning rate = 0.01:

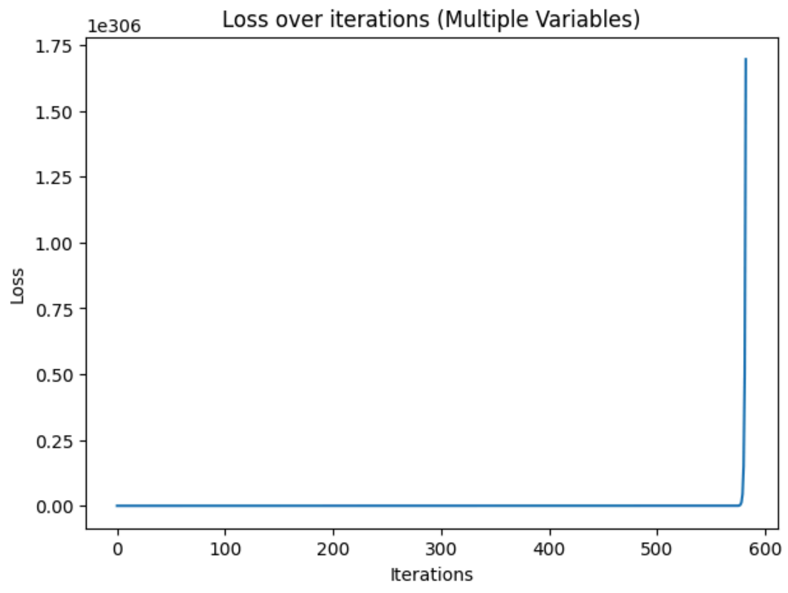
```
array([ 4.60854423, -1.90403835,  0.64916316, -0.16217188])
```



Name: Long Nguyen
Student ID: 801235507
Homework #: 1

Learning rate = 0.1:

```
/usr/local/lib/python3.10/dist-packages/numpy/core/fromnumeric.py:88: RuntimeWarning: overflow encountered in reduce
  return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
<ipython-input-47-4fee1eaab75e>:13: RuntimeWarning: overflow encountered in square
  loss = (1/n) * np.sum((Y - Y_pred) ** 2)
array([-6.55334854e+261, -1.46634665e+262, -1.44308866e+262,
       -1.43080993e+262])
```



In this case, the learning rate of 0.05 shows to be effective, leading in a smooth reduction of the loss over 1000 iterations. A higher learning rate (e.g. 0.1) has resulted in errors (overflow, instability, slower convergence). To get the same outcome with the lower learning rate, more iterations would be needed.

4. Predict the value of y for new (X1, X2, X3) values (1, 1, 1), for (2, 0, 4), and for (3, 2, 1).

```
array([3.57728282, 0.24429082, 0.10251123])
```

(X1, X2, X3) = (1, 1, 1): The predicted value of y is **3.577**.

(X1, X2, X3) = (2, 0, 4): The predicted value of y is **0.244**.

(X1, X2, X3) = (3, 2, 1): The predicted value of y is **0.103**.