Question 1

```
import numpy as np
import random
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```
random_data = np.random.normal(size=(1500, 2))
X = random_data[:,0]
y = 2*X + 5 + 0.4 * np.random.randn(1500)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3333333333, random_state=42)
len(X_train)
```

```
1000
```

```
X_train = X_train.reshape(-1, 1)
X_test = X_test.reshape(-1, 1)
# X_train
# X_test
```

```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
c = lr.intercept_
c
```

```
4.986147987363885
```

```
m = lr.coef_
m
```

```
array([1.9928823])
```

```
Y_pred_train = m*X_train + c
Y_pred_train.flatten()[:20]
```

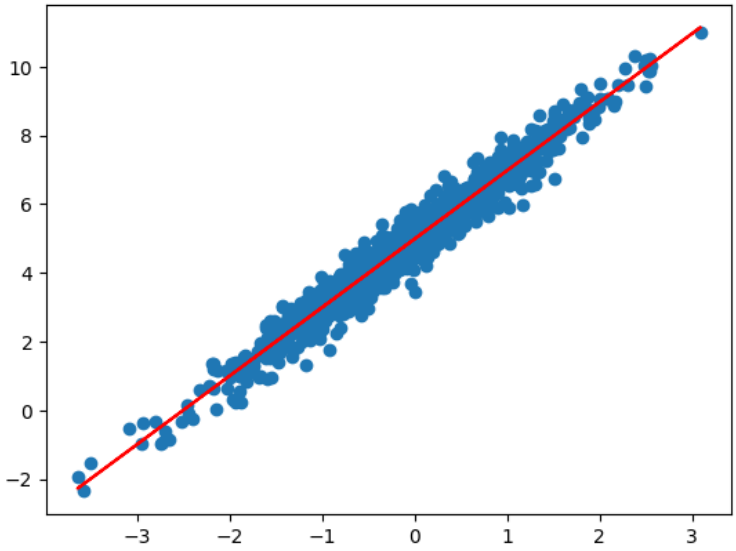
```
array([6.50050708, 5.8737251 , 3.81108637, 5.0792233 , 4.43445277,
       5.53667132, 1.73885698, 2.67689549, 6.96572459, 3.62124445,
       6.5739559 , 5.25672102, 4.89011565, 7.56042871, 2.92908586,
       5.18906399, 5.71391779, 2.53110805, 2.28432338, 3.44196924])
```

```
y_pred_train1 = lr.predict(X_train)
y_pred_train1[:20]
```

```
array([6.50050708, 5.8737251 , 3.81108637, 5.0792233 , 4.43445277,
       5.53667132, 1.73885698, 2.67689549, 6.96572459, 3.62124445,
       6.5739559 , 5.25672102, 4.89011565, 7.56042871, 2.92908586,
       5.18906399, 5.71391779, 2.53110805, 2.28432338, 3.44196924])
```

```
import matplotlib.pyplot as plt
plt.scatter(X_train, y_train)
plt.plot(X_train, y_pred_train1, color ='red')
```

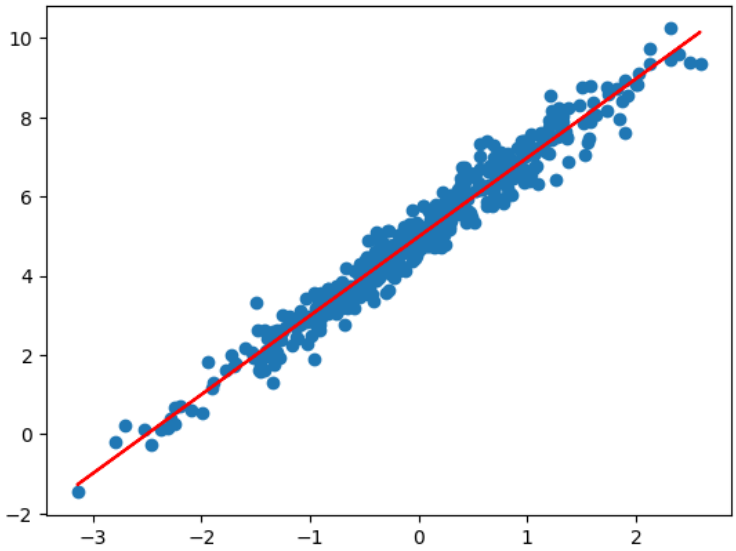
```
[<matplotlib.lines.Line2D at 0x78b63f4c75b0>]
```



```
y_pred_test1 = lr.predict(X_test)
y_pred_test1
```

```
import matplotlib.pyplot as plt
plt.scatter(X_test, y_test)
plt.plot(X_test, y_pred_test1, color ='red')
```

```
[<matplotlib.lines.Line2D at 0x78b63f5d9090>]
```



Question 2

```
import pandas as pd
```

```
data = pd.read_csv("/content/data1.csv")
data.info
```

```
<bound method DataFrame.info of       x   y
0    17  94
1    13  73
2    12  59
3    15  80
4    16  93
..   ..  ..
115  14  85
116  16  66
117  16  79
118  18  77
119  19  91

[120 rows x 2 columns]>
```

```
X = data.iloc[:,0].values
y = data.iloc[:,-1].values
```

```
X_train, X_test, y_train, y_test  = X[:84], X[84:], y[:84], y[84:] #(70:30 train test split, 70% of 120 = 84)
```

```
mean_x = np.mean(X)
mean_y = np.mean(y)
```

```
val_count = len(X)
```

```
numer = 0
denom = 0
for i in range(val_count):
  numer += (X[i] - mean_x) * (y[i] - mean_y)
  denom += (X[i] - mean_x) ** 2
m = numer / denom
c = mean_y - (m * mean_x)

print("m = ", m)
print("c = ", c)
```

```
    m =  3.1792452830188656
    c =  30.103773584905703
```

```
y_pred = m * X + c

rmse = sqrt(np.mean((y - y_pred) ** 2))
mae = np.mean(np.abs(y - y_pred))

print("RMSE =", rmse)
print("MAE =", mae)
```

```
    RMSE = 8.817810022046613
    MAE = 7.305660377358489
```

Verifing using Sklearn LR implementation

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,mean_absolute_error
from math import sqrt

X = X.reshape(val_count, 1)
lr = LinearRegression()
lr = lr.fit(X,y)
Y_pred = lr.predict(X)

rmse1 = sqrt(mean_squared_error(y,Y_pred))
mae1 = mean_absolute_error(y,Y_pred)

print("m = ", lr.coef_[0])
print("c = ", lr.intercept_)
print("RMSE =", rmse1)
print("MAE =", mae1)
```

```
    m =  3.1792452830188687
    c =  30.103773584905653
    RMSE = 8.81781002204661
    MAE = 7.305660377358489
```