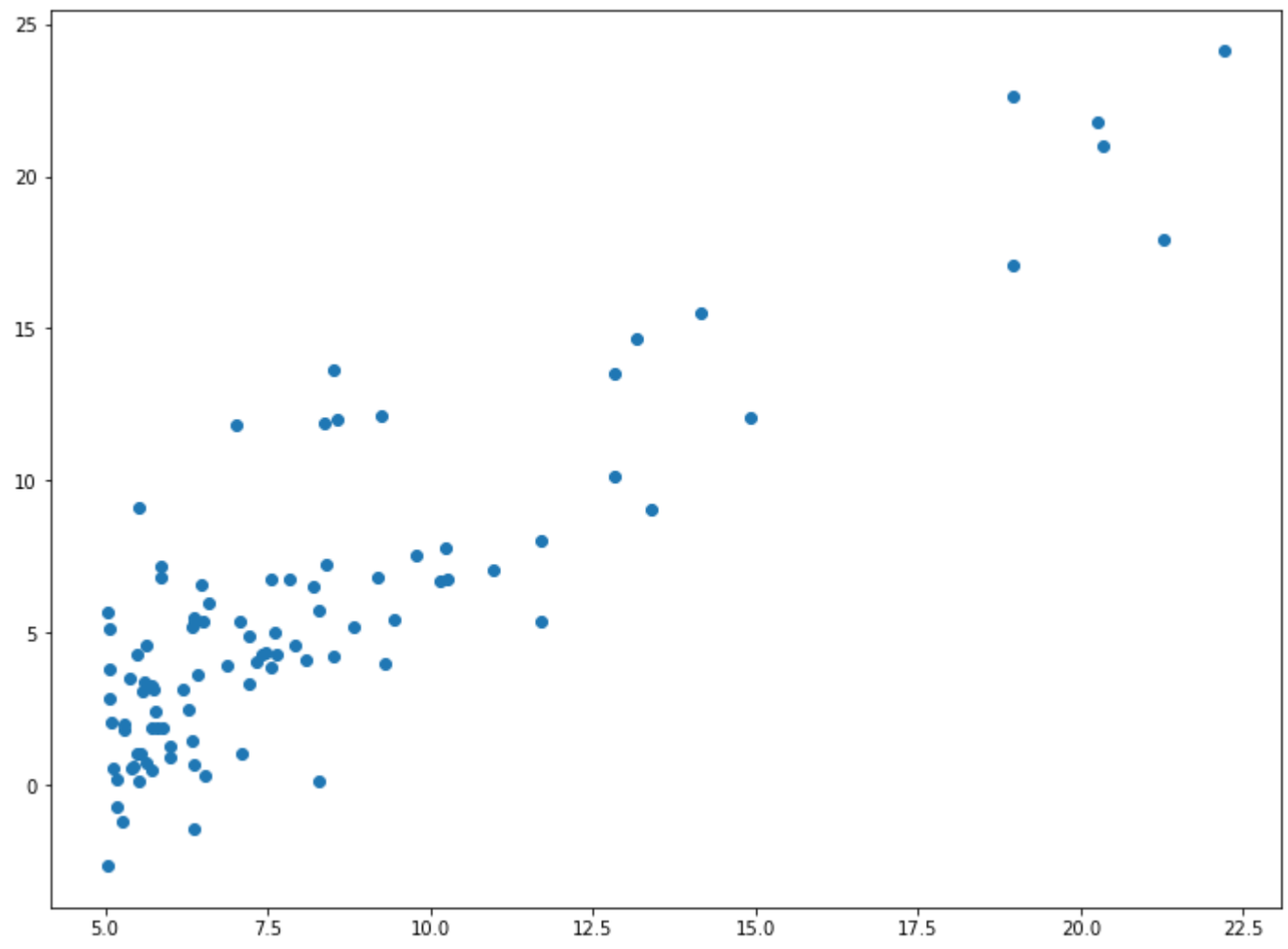
In [13]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (12.0, 9.0)
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [4]:
```
df = pd.read_csv("Downloads/ex1data1.csv", sep=",")
```

In [3]:
```
df.shape
```

Out[3]:
```
(96, 2)
```

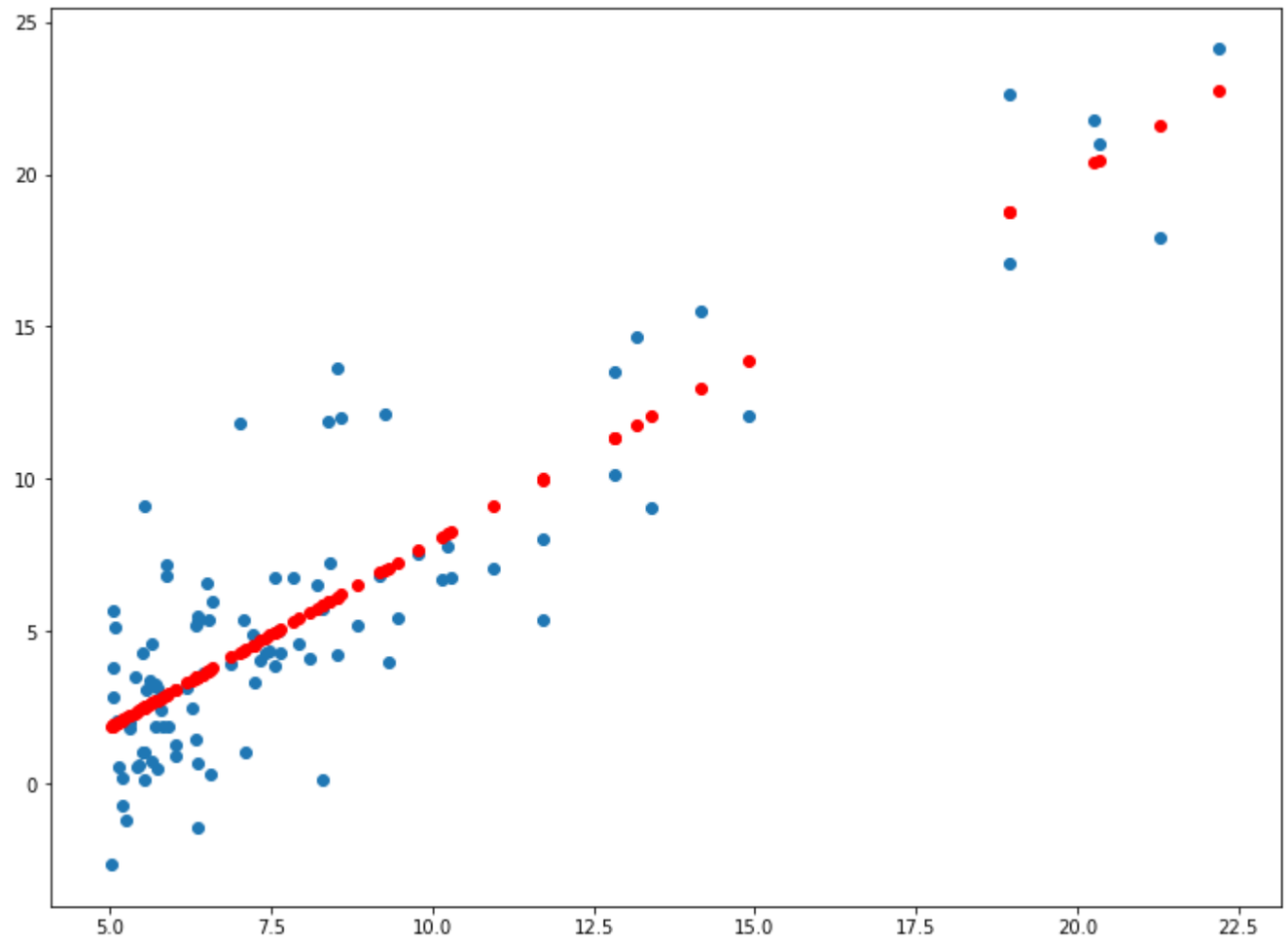
In [5]:
```
X = df.iloc[:, 0].values.reshape(-1, 1)
Y = df.iloc[:, 1].values.reshape(-1, 1)
plt.scatter(X, Y)
plt.show()
```



In [7]:
```
# building the model
X_mean = np.mean(X)
Y_mean = np.mean(Y)

num = 0
den = 0
for i in range(len(X)):
    num += (X[i] - X_mean)*(Y[i] - Y_mean)
    den += (X[i] - X_mean)**2
m = num / den
c = Y_mean - m*X_mean

print(m, c)
```

```
[1.21354725] [-4.21150401]
```

In [8]:
```
# Making Predictions
Y_pred = m*X + c

plt.scatter(X, Y)
plt.scatter(X,Y_pred, color='red')
plt.show()
```



In [15]:
```
print('Mean Absolute Error: ', metrics.mean_absolute_error(Y, Y_pred))
print('Mean Squared Error: ', metrics.mean_squared_error(Y, Y_pred))
print(' Root Mean Squared Error: ', np.sqrt(metrics.mean_squared_error(Y, Y_pred)))
```

```
Mean Absolute Error:  2.035022011375182
Mean Squared Error:  6.91916380631601
 Root Mean Squared Error:  2.630430346220179
```

In [ ]:
```

```