Simple Linear Regression with CSV

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score
```

Ranomly generated 100 values for hours studied and marks scored using random function

In [2]:
```
hours_study = np.random.rand(100, 1) * 10
exam_score = 50 + 5 * hours_study + np.random.randn(100, 1) * 2
df = pd.DataFrame({'Hours_Study': hours_study.flatten(), 'Exam_Score': exam_score.flatt
df.head()
```

Out[2]:

	Hours_Study	Exam_Score
0	7.074082	86.095461
1	1.491955	58.284550
2	8.921078	93.399532
3	0.772967	51.624327
4	0.668732	51.455379

In [3]:
```
#convert the dataframe to csv file
csv_file = 'E:\MCA\sem_3\ML_Lab\programs\week4\HoursStudied_vs_ExamScore.csv'
df.to_csv(csv_file,index=False)
```

In [4]:
```
#reading the csv file
df1 = pd.read_csv('E:\MCA\sem_3\ML_Lab\programs\week4\HoursStudied_vs_ExamScore.csv')
df1.head()
```

Out[4]:

	Hours_Study	Exam_Score
0	7.074082	86.095461
1	1.491955	58.284550
2	8.921078	93.399532
3	0.772967	51.624327
4	0.668732	51.455379

In [5]:

```
x = df1[['Hours_Study']]
x
```

Out[5]:

	Hours_Study
0	7.074082
1	1.491955
2	8.921078
3	0.772967
4	0.668732
...	...
95	9.039741
96	7.268935
97	9.720008
98	9.939549
99	3.240635

100 rows × 1 columns

In [6]:

```
y = df1[['Exam_Score']]
y
```

Out[6]:

	Exam_Score
0	86.095461
1	58.284550
2	93.399532
3	51.624327
4	51.455379
...	...
95	94.679005
96	87.655903
97	96.237040
98	100.483146
99	66.298804

100 rows × 1 columns

Iteration 1

i. test_size = 0.2 random_state = 0

In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0
```

In [8]:
```
model = LinearRegression()
```

In [9]:
```
model.fit(x_train, y_train)
```

Out[9]:
```
▾ LinearRegression
LinearRegression()
```

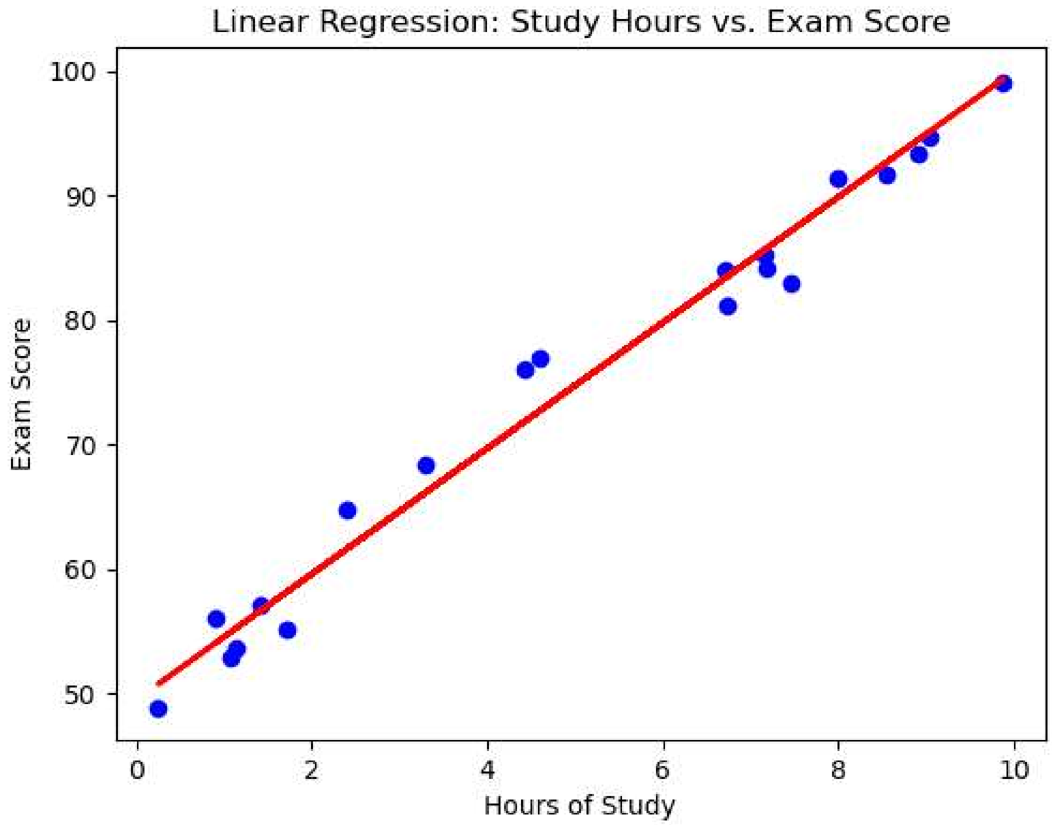
In [10]:
```
y_pred = model.predict(x_test)
```

In [11]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 5.26
R-squared: 0.98
```

In [12]:
```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [13]:
```
model.coef_
```

Out[13]:
```
array([[5.04522484]])
```

In [14]:
```
model.intercept_
```

Out[14]:
```
array([49.51061205])
```

In [15]:
```
model.score(x,y)
```

Out[15]:
```
0.9853069451039211
```

In [16]:
```
hours_new = [[5]]
marks_predicted = model.predict(hours_new)
marks_predicted
```

```
C:\Users\Dell\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not
have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[16]:
```
array([[74.73673627]])
```

ii. test_size = 0.2 random_state = 42

In [17]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=4
```

In [18]:
```
model = LinearRegression()
```

In [19]:
```
model.fit(x_train, y_train)
```

Out[19]:
```
▾ LinearRegression
LinearRegression()
```

In [20]:
```
y_pred = model.predict(x_test)
```

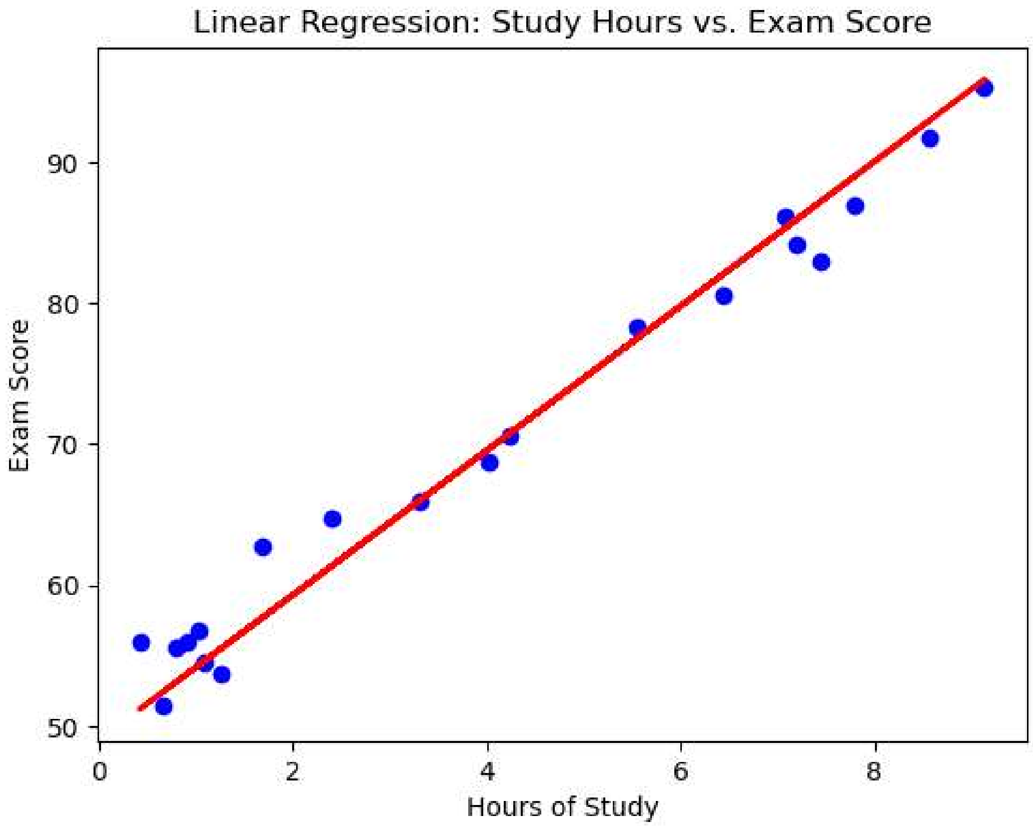
In [21]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 5.65
R-squared: 0.97
```

In [22]:

```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [23]:

```
model.coef_
```

Out[23]:

```
array([[5.12578263]])
```

In [24]:

```
model.intercept_
```

Out[24]:

```
array([49.02875415])
```

In [25]:

```
model.score(x,y)
```

Out[25]:

```
0.984918296389473
```

Iteration 2

i. test_split = 0.3 random_state = 0

In [26]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0
```

In [27]:

```
model = LinearRegression()
```

In [28]:

```
model.fit(x_train, y_train)
```

Out[28]:

```
▾ LinearRegression
LinearRegression()
```

In [29]:

```
y_pred = model.predict(x_test)
```

In [30]:

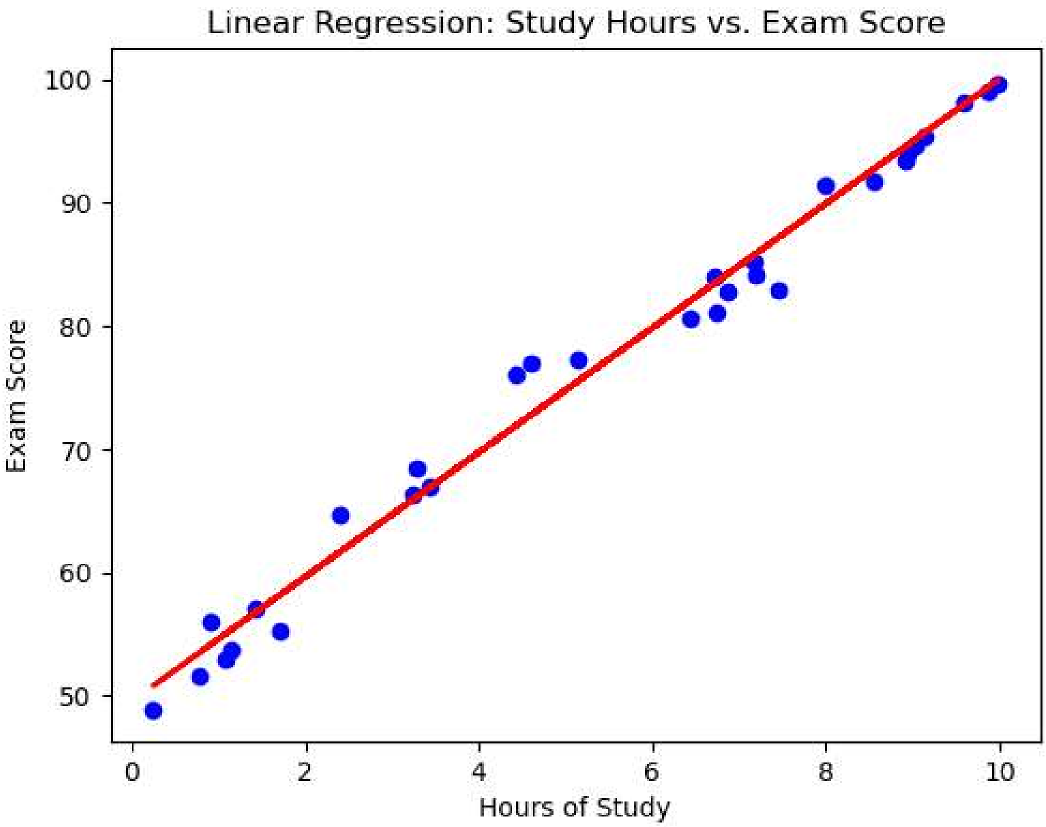
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 3.88
R-squared: 0.98
```

In [31]:

```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [32]:
```
model.coef_
```

Out[32]:
```
array([[5.0487867]])
```

In [33]:
```
model.intercept_
```

Out[33]:
```
array([49.53930025])
```

In [34]:
```
model.score(x,y)
```

Out[34]:
```
0.985285571128082
```

ii. test_split = 0.3 random_state=42

In [35]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=4
```

In [36]:
```
model = LinearRegression()
```

In [37]:
```
model.fit(x_train, y_train)
```

Out[37]:
```
▾ LinearRegression
LinearRegression()
```

In [38]:
```
y_pred = model.predict(x_test)
```

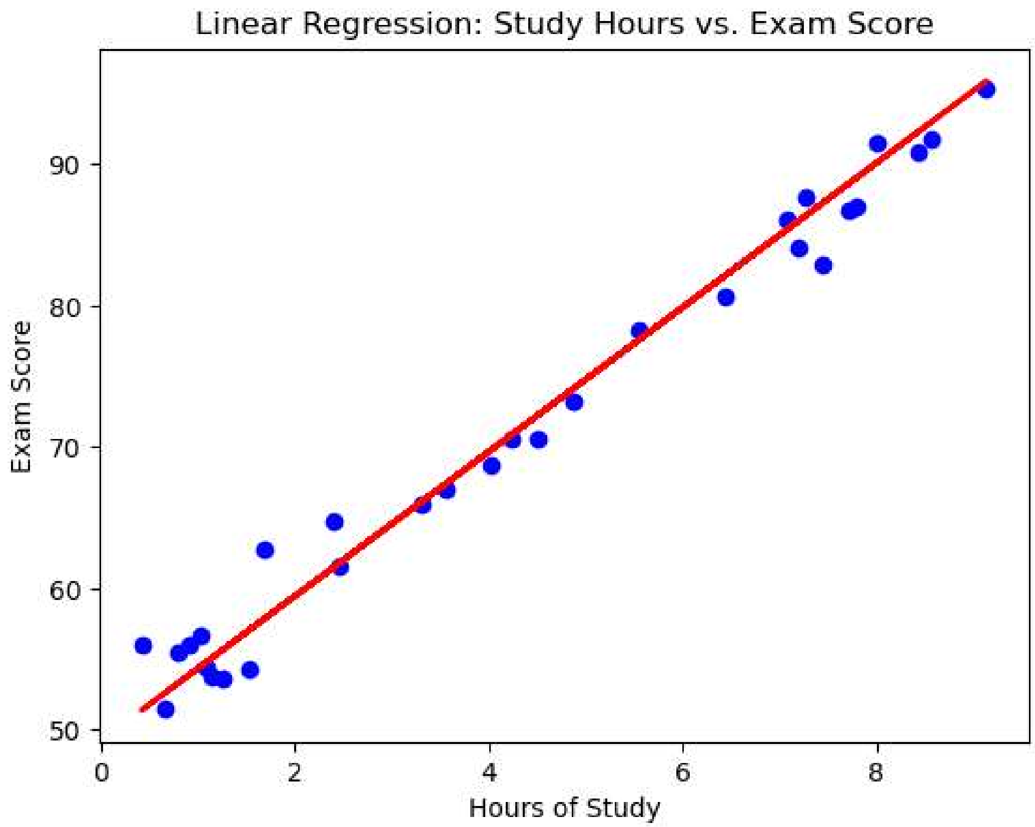
In [39]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 4.36
R-squared: 0.98
```

In [40]:

```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [41]:

```
model.coef_
```

Out[41]:

```
array([[5.10453039]])
```

In [42]:

```
model.intercept_
```

Out[42]:

```
array([49.23615891])
```

In [43]:

```
model.score(x,y)
```

Out[43]:

```
0.9850796729927146
```

Iteration 3

i. test_split = 0.4 random_state=0

In [44]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random_state=0
```

In [45]:
```
model = LinearRegression()
```

In [46]:
```
model.fit(x_train, y_train)
```

Out[46]:
```
▾ LinearRegression
LinearRegression()
```

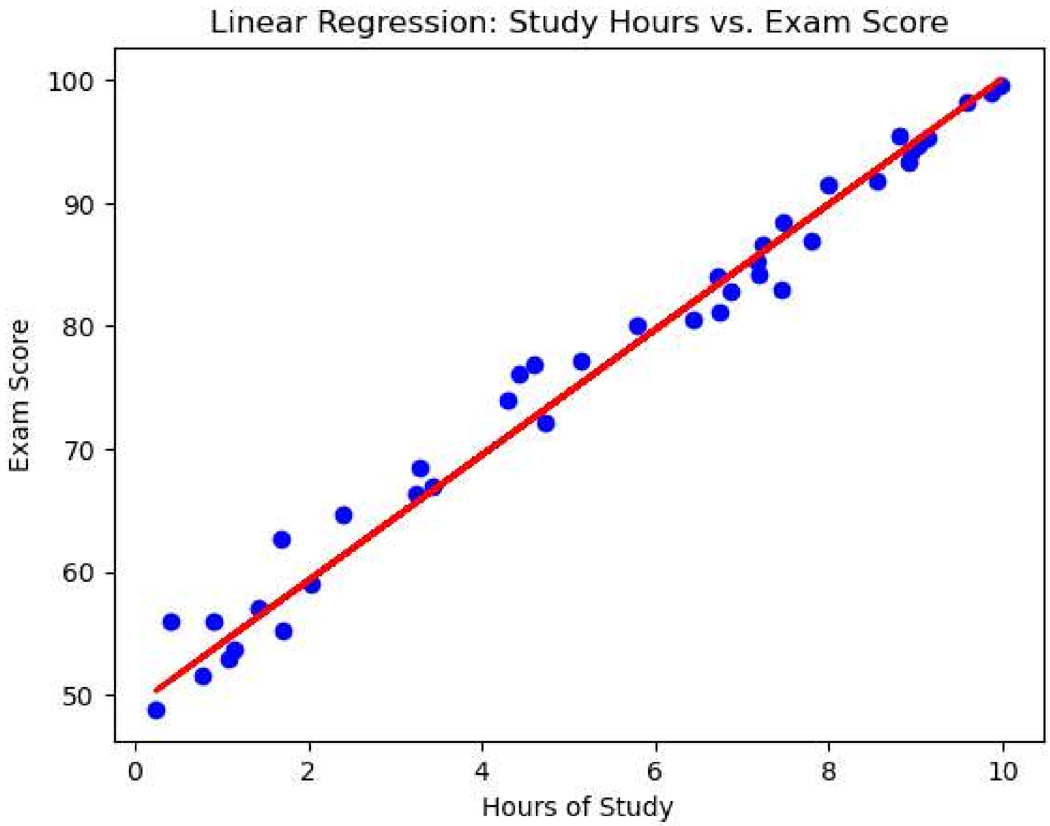
In [47]:
```
y_pred = model.predict(x_test)
```

In [48]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 4.61
R-squared: 0.98
```

In [49]:
```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [50]:
```
model.coef_
```

Out[50]:
```
array([[5.09853903]])
```

In [51]:
```
model.intercept_
```

Out[51]:
```
array([49.10463486])
```

In [52]:
```
model.score(x,y)
```

Out[52]:
```
0.9850510890023612
```

ii. test_split = 0.4 random_state = 42

In [53]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random_state=4
```

In [54]:
```
model = LinearRegression()
```

In [55]:
```
model.fit(x_train, y_train)
```

Out[55]:
```
▾ LinearRegression
LinearRegression()
```

In [56]:
```
y_pred = model.predict(x_test)
```

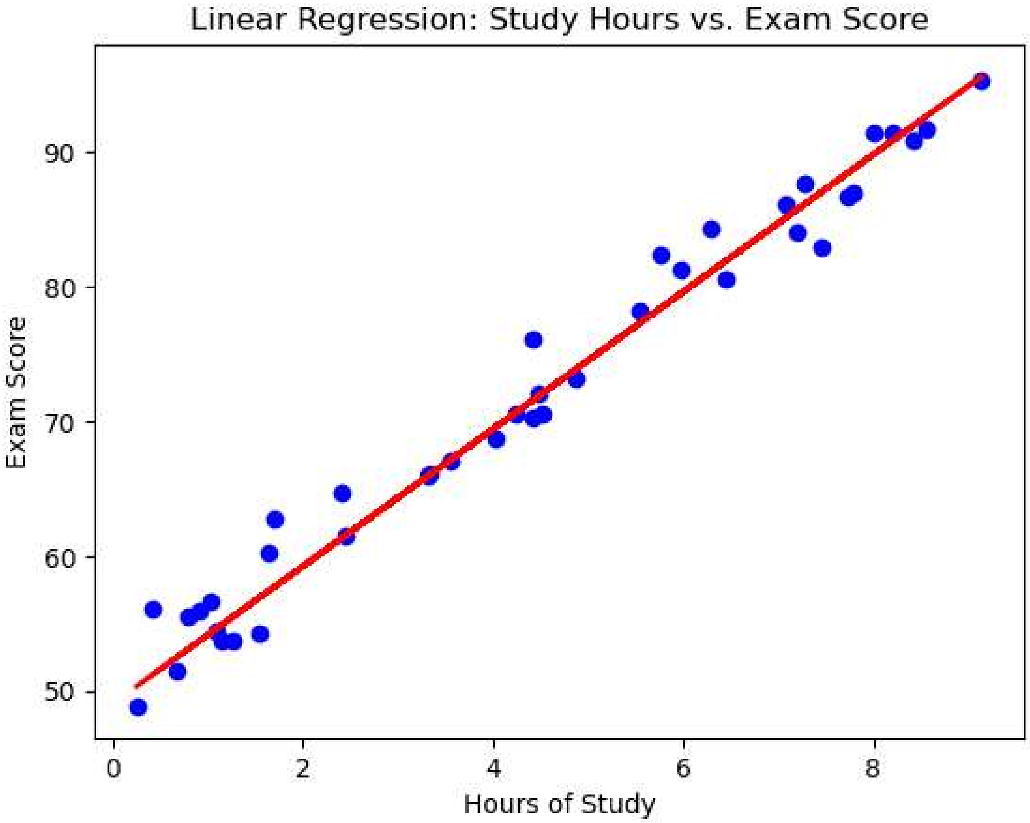
In [57]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 4.76
R-squared: 0.97
```

In [58]:

```
plt.scatter(x_test, y_test, color='blue')
plt.plot(x_test, y_pred, color='red', linewidth=2)
plt.xlabel('Hours of Study')
plt.ylabel('Exam Score')
plt.title('Linear Regression: Study Hours vs. Exam Score')
plt.show()
```



In [59]:

```
model.coef_
```

Out[59]:

```
array([[5.0986416]])
```

In [60]:

```
model.intercept_
```

Out[60]:

```
array([49.06285763])
```

In [61]:

```
model.score(x,y)
```

Out[61]:

```
0.9849982956855295
```