In [2]: import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("/home/dslab/Downloads/std1.csv") df

df.plot(x="Hours", y="Scores", style="o") plt.show()

x\_mean = df["Hours"].mean() y\_mean = df["Scores"].mean() print(x\_mean, y\_mean)

df["x"] = df["Hours"] - x\_mean

df["y"] = df["Scores"] - y\_mean

df["x\*y"] = df["x"] \* df["y"] df["x^2"] = df["x"]\*\*2

df["y^2"] = df["y"]\*\*2 df

summation\_x\_y = df["x\*y"].sum() summation\_x\_squared = df["x^2"].sum() summation\_y\_squared = df["y^2"].sum()

print(summation\_x\_y, summation\_x\_squared, summation\_y\_squared)

correlation = summation\_x\_y / (summation\_x\_squared \* summation\_y\_squ correlation

def getMean(numbers):

iflen (numbers) == 0:

return None else:

current\_sum = 0

for i in numbers: current\_sum += i

current\_avg = current\_sum/len(numbers)

return current\_avg

def getStandardDeviation(numbers):

iflen (numbers) == 0:

return 0

else:

mean = getMean(numbers) std\_deviation = 0

for i in numbers:

std\_deviation += (i - mean)\*\*2

return (std\_deviation/len(numbers))\*\*0.5

std\_deviation\_x = getStandardDeviation(df["x"].tolist()) std\_deviation\_y = getStandardDeviation(df["y"].tolist())

print(std\_deviation\_x, std\_deviation\_y)

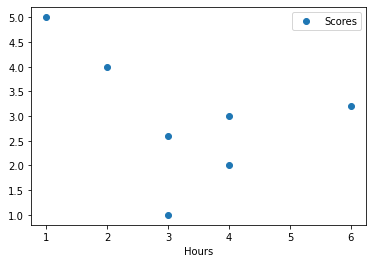
m = correlation \* (std\_deviation\_y / std\_deviation\_x) m

c = df["Scores"].mean() - m \* df["Hours"].mean() c

df["y\_prediction"] = m \* df["Hours"] + c df

plot1 = plt.scatter(df["Hours"], df["Scores"])

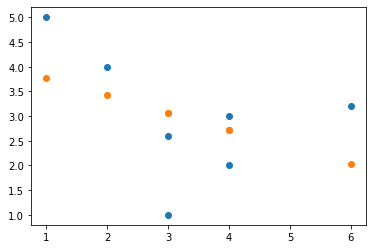
plot2 = plt.scatter(df["Hours"], df["y\_prediction"]) plt.show()



3.2857142857142856 2.9714285714285715

-5.3428571428571425 15.428571428571429 10.194285714285716

1.4846149779161806 1.206783547539593



In [3]:

df["error"] = df["y"]-df["y\_prediction"] df

Out[3]:

In [6]:

df["SSE"]=df["error"]\*\*2 df

Hours Scores x y x\*y x^2 y^2 y\_prediction erro

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 03 | 1.0 | -0.285714 | -1.971429 | 0.563265 | 0.081633 | 3.886531 | 3.070370 | -5.04179 |
| 14 | 3.0 | 0.714286 | 0.028571 | 0.020408 | 0.510204 | 0.000816 | 2.724074 | -2.69550 |
| 21 | 5.0 | -2.285714 | 2.028571 | -4.636735 | 5.224490 | 4.115102 | 3.762963 | -1.73439 |
| 32 | 4.0 | -1.285714 | 1.028571 | -1.322449 | 1.653061 | 1.057959 | 3.416667 | -2.38809 |
| 44 | 2.0 | 0.714286 | -0.971429 | -0.693878 | 0.510204 | 0.943673 | 2.724074 | -3.69550 |
| 56 | 3.2 | 2.714286 | 0.228571 | 0.620408 | 7.367347 | 0.052245 | 2.031481 | -1.80291 |
| 63 | 2.6 | -0.285714 | -0.371429 | 0.106122 | 0.081633 | 0.137959 | 3.070370 | -3.44179 |

Out[6]:

In [9]:

df["y\_mean"]=df["y\_prediction"].mean()

Hours Scores x y x\*y x^2 y^2 y\_prediction erro

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 03 | 1.0 | -0.285714 | -1.971429 | 0.563265 | 0.081633 | 3.886531 | 3.070370 | -5.04179 |
| 14 | 3.0 | 0.714286 | 0.028571 | 0.020408 | 0.510204 | 0.000816 | 2.724074 | -2.69550 |
| 21 | 5.0 | -2.285714 | 2.028571 | -4.636735 | 5.224490 | 4.115102 | 3.762963 | -1.73439 |
| 32 | 4.0 | -1.285714 | 1.028571 | -1.322449 | 1.653061 | 1.057959 | 3.416667 | -2.38809 |
| 44 | 2.0 | 0.714286 | -0.971429 | -0.693878 | 0.510204 | 0.943673 | 2.724074 | -3.69550 |
| 56 | 3.2 | 2.714286 | 0.228571 | 0.620408 | 7.367347 | 0.052245 | 2.031481 | -1.80291 |
| 63 | 2.6 | -0.285714 | -0.371429 | 0.106122 | 0.081633 | 0.137959 | 3.070370 | -3.44179 |

In [10]:

df["SST"]=(df["y"]-df["y\_mean"]).sum() df

Out[10]:

Hours Scores x y x\*y x^2 y^2 y\_prediction erro

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 03 | 1.0 | -0.285714 | -1.971429 | 0.563265 | 0.081633 | 3.886531 | 3.070370 | -5.04179 |
| 14 | 3.0 | 0.714286 | 0.028571 | 0.020408 | 0.510204 | 0.000816 | 2.724074 | -2.69550 |
| 21 | 5.0 | -2.285714 | 2.028571 | -4.636735 | 5.224490 | 4.115102 | 3.762963 | -1.73439 |
| 32 | 4.0 | -1.285714 | 1.028571 | -1.322449 | 1.653061 | 1.057959 | 3.416667 | -2.38809 |
| 44 | 2.0 | 0.714286 | -0.971429 | -0.693878 | 0.510204 | 0.943673 | 2.724074 | -3.69550 |
| 56 | 3.2 | 2.714286 | 0.228571 | 0.620408 | 7.367347 | 0.052245 | 2.031481 | -1.80291 |
| 63 | 2.6 | -0.285714 | -0.371429 | 0.106122 | 0.081633 | 0.137959 | 3.070370 | -3.44179 |

Out[11]:

Hours Scores x y x\*y x^2 y^2 y\_prediction erro

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 03 | 1.0 | -0.285714 | -1.971429 | 0.563265 | 0.081633 | 3.886531 | 3.070370 | -5.04179 |
| 14 | 3.0 | 0.714286 | 0.028571 | 0.020408 | 0.510204 | 0.000816 | 2.724074 | -2.69550 |
| 21 | 5.0 | -2.285714 | 2.028571 | -4.636735 | 5.224490 | 4.115102 | 3.762963 | -1.73439 |
| 32 | 4.0 | -1.285714 | 1.028571 | -1.322449 | 1.653061 | 1.057959 | 3.416667 | -2.38809 |
| 44 | 2.0 | 0.714286 | -0.971429 | -0.693878 | 0.510204 | 0.943673 | 2.724074 | -3.69550 |
| 56 | 3.2 | 2.714286 | 0.228571 | 0.620408 | 7.367347 | 0.052245 | 2.031481 | -1.80291 |
| 63 | 2.6 | -0.285714 | -0.371429 | 0.106122 | 0.081633 | 0.137959 | 3.070370 | -3.44179 |

In [12]:

df["r\_sqr"]=df["SSR"]/df["SST"] df

Out[12]:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Hours Scores | | | | x | y | x\*y | x^2 | y^2 | y\_prediction | erro |
|  | 03 | 1.0 | -0.285714 | -1.971429 | 0.563265 | | 0.081633 | 3.886531 | 3.070370 | -5.04179 |
|  | 14 | 3.0 | 0.714286 | 0.028571 | 0.020408 | | 0.510204 | 0.000816 | 2.724074 | -2.69550 |
|  | 21 | 5.0 | -2.285714 | 2.028571 | -4.636735 | | 5.224490 | 4.115102 | 3.762963 | -1.73439 |
|  | 32 | 4.0 | -1.285714 | 1.028571 | -1.322449 | | 1.653061 | 1.057959 | 3.416667 | -2.38809 |
|  | 44 | 2.0 | 0.714286 | -0.971429 | -0.693878 | | 0.510204 | 0.943673 | 2.724074 | -3.69550 |
|  | 56 | 3.2 | 2.714286 | 0.228571 | 0.620408 | | 7.367347 | 0.052245 | 2.031481 | -1.80291 |
| In [ ]: | 63 | 2.6 | -0.285714 | -0.371429 | 0.106122 | | 0.081633 | 0.137959 | 3.070370 | -3.44179 |