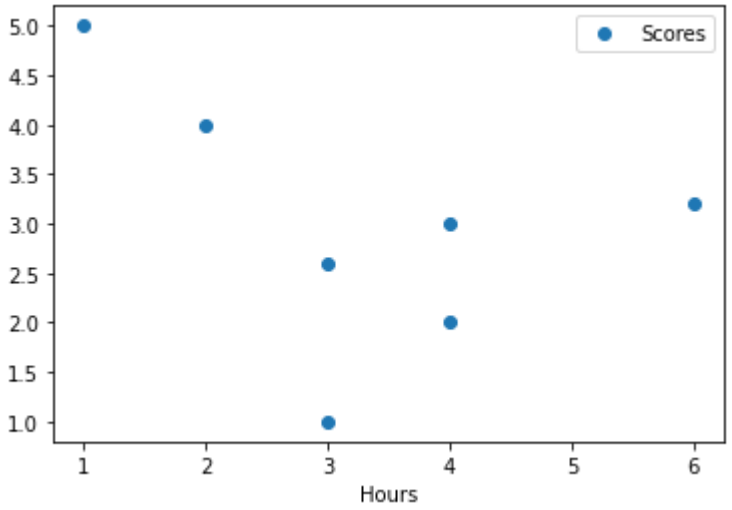
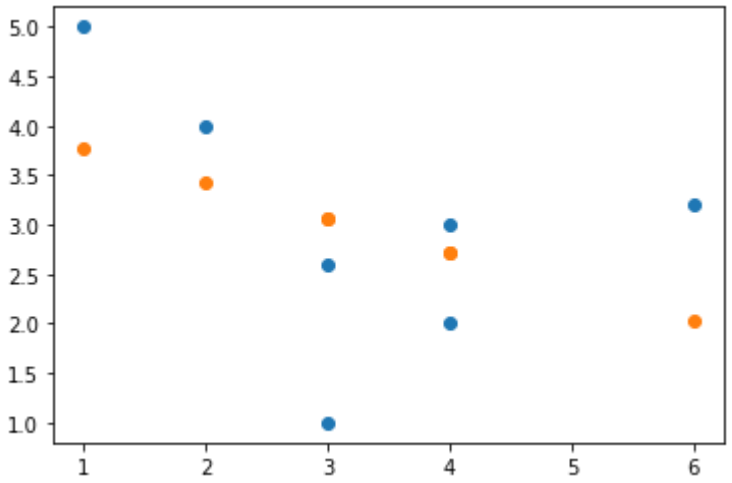
In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt


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_squa
correlation


def getMean(numbers):
    if len(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):
    if len(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]:

	Hours	Scores	x	y	x*y	x^2	y^2	y_prediction	erro
0	3	1.0	-0.285714	-1.971429	0.563265	0.081633	3.886531	3.070370	-5.04179
1	4	3.0	0.714286	0.028571	0.020408	0.510204	0.000816	2.724074	-2.69550
2	1	5.0	-2.285714	2.028571	-4.636735	5.224490	4.115102	3.762963	-1.73439
3	2	4.0	-1.285714	1.028571	-1.322449	1.653061	1.057959	3.416667	-2.38809
4	4	2.0	0.714286	-0.971429	-0.693878	0.510204	0.943673	2.724074	-3.69550
5	6	3.2	2.714286	0.228571	0.620408	7.367347	0.052245	2.031481	-1.80291
6	3	2.6	-0.285714	-0.371429	0.106122	0.081633	0.137959	3.070370	-3.44179

In [6]:
```
df["SSE"]=df["error"]**2
df
```

Out[6]:

	Hours	Scores	x	y	x*y	x^2	y^2	y_prediction	erro
0	3	1.0	-0.285714	-1.971429	0.563265	0.081633	3.886531	3.070370	-5.04179
1	4	3.0	0.714286	0.028571	0.020408	0.510204	0.000816	2.724074	-2.69550
2	1	5.0	-2.285714	2.028571	-4.636735	5.224490	4.115102	3.762963	-1.73439
3	2	4.0	-1.285714	1.028571	-1.322449	1.653061	1.057959	3.416667	-2.38809
4	4	2.0	0.714286	-0.971429	-0.693878	0.510204	0.943673	2.724074	-3.69550
5	6	3.2	2.714286	0.228571	0.620408	7.367347	0.052245	2.031481	-1.80291
6	3	2.6	-0.285714	-0.371429	0.106122	0.081633	0.137959	3.070370	-3.44179

In [9]:
```
df["y_mean"]=df["y_prediction"].mean()
```

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
0	3	1.0	-0.285714	-1.971429	0.563265	0.081633	3.886531	3.070370	-5.04179
1	4	3.0	0.714286	0.028571	0.020408	0.510204	0.000816	2.724074	-2.69550
2	1	5.0	-2.285714	2.028571	-4.636735	5.224490	4.115102	3.762963	-1.73439
3	2	4.0	-1.285714	1.028571	-1.322449	1.653061	1.057959	3.416667	-2.38809
4	4	2.0	0.714286	-0.971429	-0.693878	0.510204	0.943673	2.724074	-3.69550
5	6	3.2	2.714286	0.228571	0.620408	7.367347	0.052245	2.031481	-1.80291
6	3	2.6	-0.285714	-0.371429	0.106122	0.081633	0.137959	3.070370	-3.44179

In [11]:
```
df["SSR"]=(df["SST"]**2).sum()
df
```

Out[11]:

	Hours	Scores	x	y	x*y	x^2	y^2	y_prediction	erro
0	3	1.0	-0.285714	-1.971429	0.563265	0.081633	3.886531	3.070370	-5.04179
1	4	3.0	0.714286	0.028571	0.020408	0.510204	0.000816	2.724074	-2.69550
2	1	5.0	-2.285714	2.028571	-4.636735	5.224490	4.115102	3.762963	-1.73439
3	2	4.0	-1.285714	1.028571	-1.322449	1.653061	1.057959	3.416667	-2.38809
4	4	2.0	0.714286	-0.971429	-0.693878	0.510204	0.943673	2.724074	-3.69550
5	6	3.2	2.714286	0.228571	0.620408	7.367347	0.052245	2.031481	-1.80291
6	3	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
0	3	1.0	-0.285714	-1.971429	0.563265	0.081633	3.886531	3.070370	-5.04179
1	4	3.0	0.714286	0.028571	0.020408	0.510204	0.000816	2.724074	-2.69550
2	1	5.0	-2.285714	2.028571	-4.636735	5.224490	4.115102	3.762963	-1.73439
3	2	4.0	-1.285714	1.028571	-1.322449	1.653061	1.057959	3.416667	-2.38809
4	4	2.0	0.714286	-0.971429	-0.693878	0.510204	0.943673	2.724074	-3.69550
5	6	3.2	2.714286	0.228571	0.620408	7.367347	0.052245	2.031481	-1.80291
6	3	2.6	-0.285714	-0.371429	0.106122	0.081633	0.137959	3.070370	-3.44179

In [ ]:
```

```