LAB 3 SET 2

E21CSEU0054

Vashu Agarwal

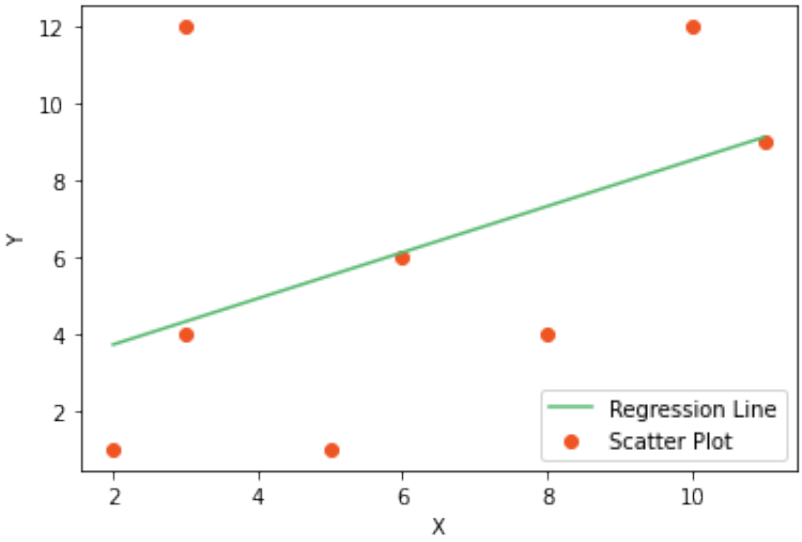
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Q1

In [2]:
```
X = np.array([8,3,2,10,11,3,6,5])
Y = np.array([4,12,1,12,9,4,6,1])

mean_x = np.mean(X)
mean_y = np.mean(Y)

n = len(X)

numer = 0
denom =0

for i in range(n):
    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("Coefficients")
print(m,c)
```

```
Coefficients
0.6 2.5250000000000004
```

In [3]:
```
# Plotting the line of best fit
# Plotting Values and Regression Line

max_x = np.max(X)
min_x = np.min(X)

# Calculating line values x and y
x = np.linspace(min_x, max_x, 1000)
y = c + m * x

# Ploting Line
plt.plot(x, y, color='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.show()
```

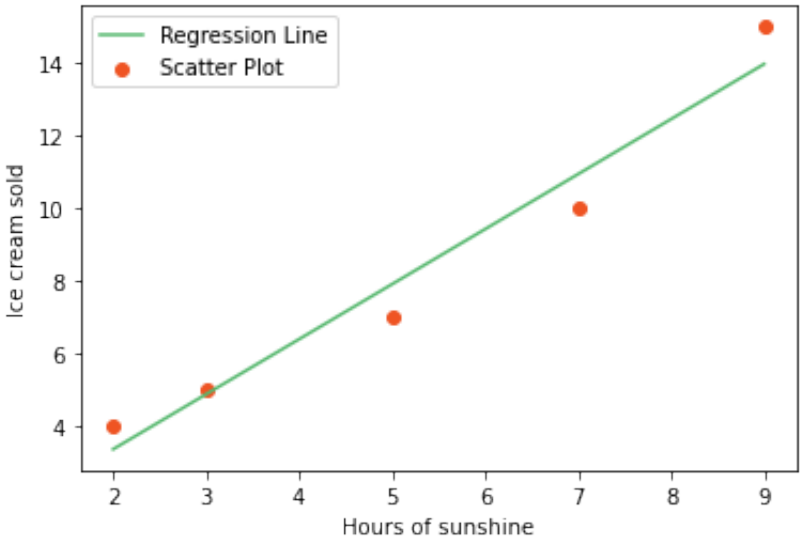


Q2

In [4]:
```
X = np.array([2,3,5,7,9])
Y = np.array([4,5,7,10,15])
mean_x = np.mean(X)
mean_y = np.mean(Y)

n = len(X)

numer = 0
denom =0

for i in range(n):
    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("Coefficients")
print(m,c)
```

```
Coefficients
1.518292682926829 0.30487804878048763
```

In [5]:
```
# Plotting the line of best fit
# Plotting Values and Regression Line

max_x = np.max(X)
min_x = np.min(X)

# Calculating line values x and y
x = np.linspace(min_x, max_x, 1000)
y = c + m * x

# Ploting Line
plt.plot(x, y, color='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('Hours of sunshine')
plt.ylabel('Ice cream sold')
plt.legend()
plt.show()
```



In [6]:
```
f = c+m*(8)
print(f)
```

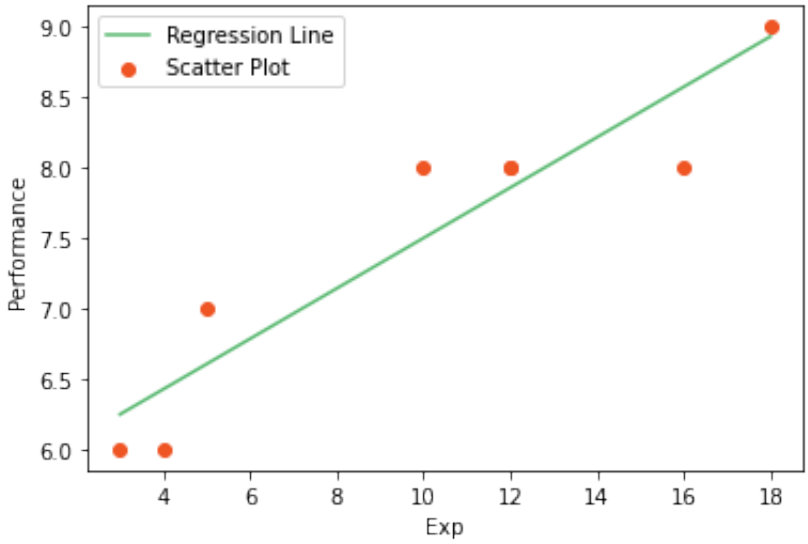
```
12.45121951219512
```

Q3

In [17]:
```
X = np.array([16,12,18,4,3,10,5,12])
Y = np.array([8,8,9,6,6,8,7,8])
mean_x = np.mean(X)
mean_y = np.mean(Y)

n = len(X)

numer = 0
denom =0

for i in range(n):
    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("Coefficients")
print(m,c)
```

```
Coefficients
0.17889908256880735 5.7110091743119265
```

In [18]:
```
# Plotting the line of best fit
# Plotting Values and Regression Line

max_x = np.max(X)
min_x = np.min(X)

# Calculating line values x and y
x = np.linspace(min_x, max_x, 1000)
y = c + m * x

# Ploting Line
plt.plot(x, y, color='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('Exp')
plt.ylabel('Performance')
plt.legend()
plt.show()
```



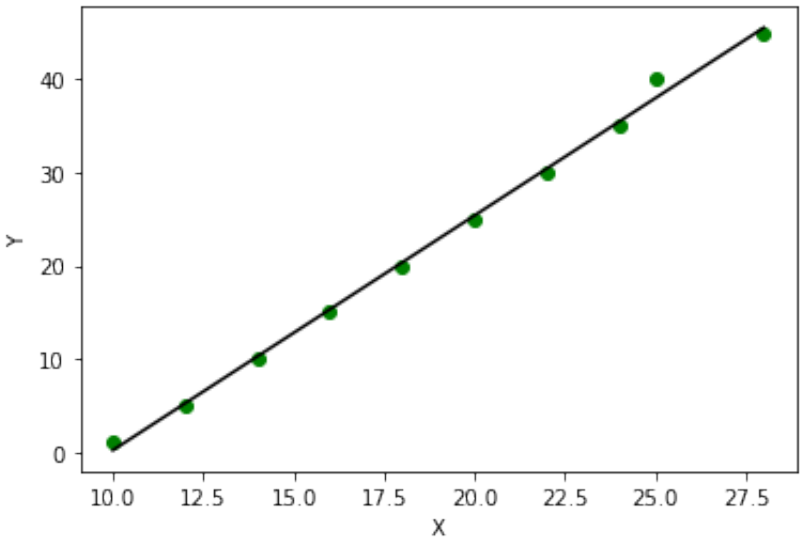
In [19]:
```
per = c+m*25
print(per)
```

```
10.18348623853211
```

Q4

In [16]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt

from sklearn.linear_model import LinearRegression

# LinearRegression will expect an array of shape (n, 1)
# for the "Training data"
X = np.array([10, 12, 14, 16, 18, 20, 22, 24, 25, 28]).reshape((-1,
# target data is array of shape (n,)
y = np.array([1, 5, 10, 15, 20, 25, 30, 35, 40, 45])


model2 = LinearRegression()
model2.fit(X, y)

plt.scatter(X, y,color='g')
plt.plot(X, model2.predict(X),color='k')
m = model2.coef_[0]
c = model2.intercept_
print(m,c)
plt.xlabel('X')
plt.ylabel('Y')

plt.show()
```

```
2.5200378668349632 -25.0287156831808
```



In [ ]:
```

```