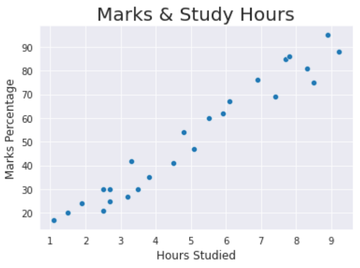
In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
```

In [4]:
```
data=pd.read_csv('http://bit.ly/w-data')
data.head()
```

Out[4]:

| | Hours | Scores |
|---|---|---|
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |

In [6]:
```
sns.set_style('darkgrid')
sns.scatterplot(y=data['Scores'],x=data['Hours'])
plt.title('Marks & Study Hours',size=20)
plt.ylabel('Marks Percentage',size=12)
plt.xlabel('Hours Studied',size=12)
plt.show()
```




In [7]:
```
sns.regplot(x=data['Hours'],y=data['Scores'])
plt.title('Regression Plot',size=20)
plt.ylabel('Mark Percentage',size=12)
plt.xlabel('Hours Studied',size=12)
plt.show()
print(data.corr())
```


```
            Hours    Scores
Hours    1.000000  0.976191
Scores   0.976191  1.000000
```

In [9]:
```
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
train_x,val_x,train_y,val_y=train_test_split(x,y,random_state=0)
regression=LinearRegression()
regression.fit(train_x,train_y)
```

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

In [11]:
```
pred_y=regression.predict(val_x)
prediction=pd.DataFrame({'Hours':[i[0] for i  in val_x],'predicted marks':[k for k in pred_y]})
prediction
```

Out[11]:

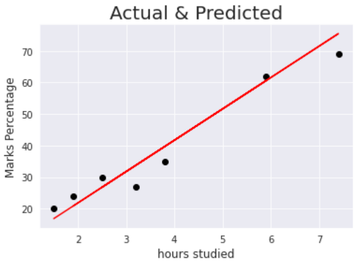
| | Hours | predicted marks |
|---|---|---|
| **0** | 1.5 | 16.844722 |
| **1** | 3.2 | 33.745575 |
| **2** | 7.4 | 75.500624 |
| **3** | 2.5 | 26.786400 |
| **4** | 5.9 | 60.588106 |
| **5** | 3.8 | 39.710582 |
| **6** | 1.9 | 20.821393 |

In [12]:
```
compare_score=pd.DataFrame({'Actual Marks':val_y,'Predicted Marks':pred_y})
compare_score
```

Out[12]:

| | Actual Marks | Predicted Marks |
|---|---|---|
| **0** | 20 | 16.844722 |
| **1** | 27 | 33.745575 |
| **2** | 69 | 75.500624 |
| **3** | 30 | 26.786400 |
| **4** | 62 | 60.588106 |
| **5** | 35 | 39.710582 |
| **6** | 24 | 20.821393 |

In [13]:
```
plt.scatter(x=val_x,y=val_y,color='black')
plt.plot(val_x,pred_y,color='red')
plt.title('Actual & Predicted', size=20)
plt.ylabel('Marks Percentage',size=12)
plt.xlabel('hours studied',size=12)
plt.show()
```




In [14]:
```
print('Mean Error:',mean_absolute_error(val_y,pred_y))
```

```
Mean Error: 4.130879918502482
```

In [18]:
```
hours=[9.25]
ans=regression.predict([hours])
print("Score=",format(round(ans[0],2)))
```

```
Score= 93.89
```

In [ ]:
```
```