In [9]:

```
sum1=0
sum2=0

def calculator(x,y,sum1,sum2):
    ax = 27
    ay =21.8
    result = (x-ax)*(y-ay)
    sum1+=result
    sum2+=(x-ax)**2
    print(result,(x-ax)**2)
    return sum1,sum2

sum1,sum2=calculator(5,7,sum1,sum2)
sum1,sum2=calculator(15,20,sum1,sum2)
sum1,sum2=calculator(25,12,sum1,sum2)
sum1,sum2=calculator(35,32,sum1,sum2)
sum1,sum2=calculator(55,38,sum1,sum2)

print(sum1)
print(sum2)
```

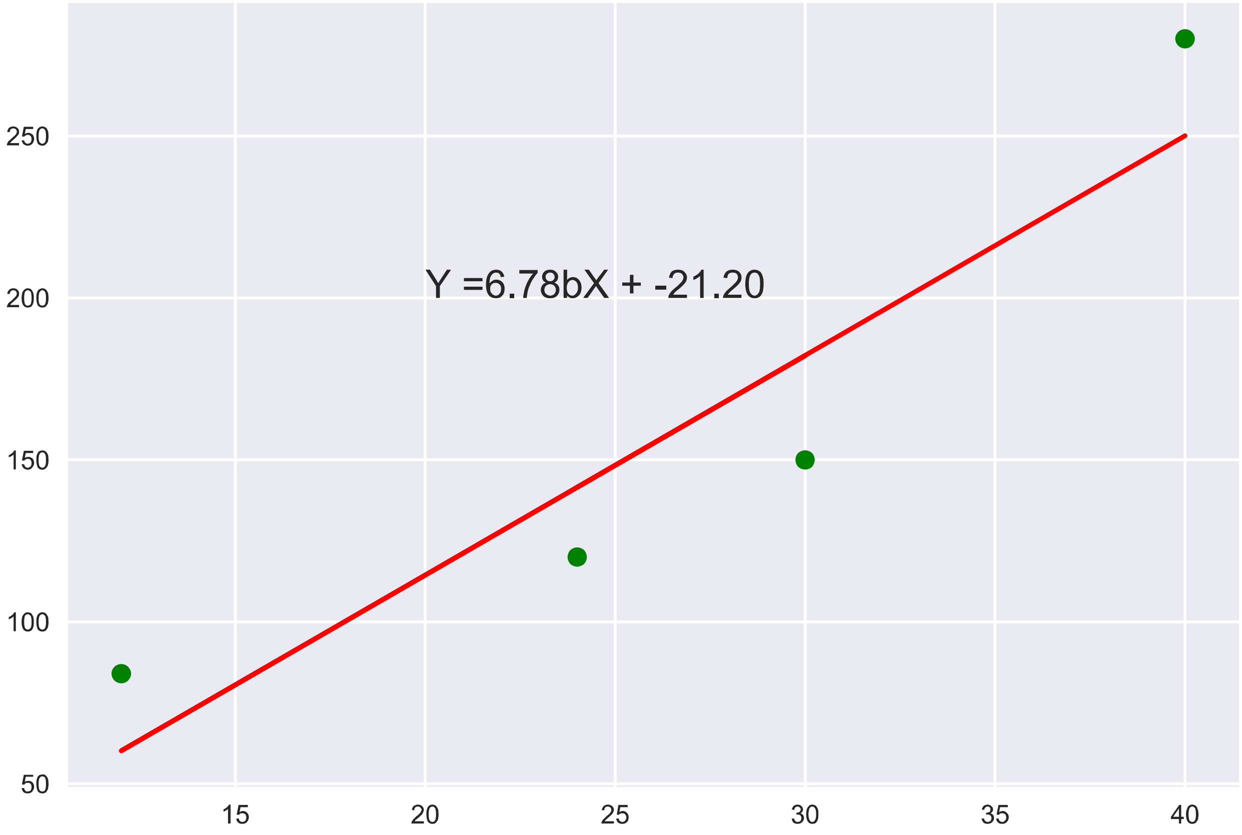
```
325.6 484
21.60000000000001 144
19.6 4
81.6 64
453.59999999999997 784
902.0
1480
```

In [10]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt
plt.style.use('seaborn')

#step2: read data
x_train = np.array([40,12,30,24]).reshape((-1,1))
y_train = np.array([280,84,150,120])


#step3: train Regression model
model = LinearRegression()
model.fit(x_train,y_train)

#step4: y=bX + a
b = model.coef_
a = model.intercept_
print("LinearRegession equation is: Y = %.2fX + %.2f"%(b,a))

#R2 MSE
y_pred = model.predict(x_train)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)

print("R2 : %.2f"%(r2))
print("MSE : %.2f"%(mse))

#step5:
x_new = np.array([200,300,500]).reshape((-1,1))
```

```
y_pred_new = model.predict(x_new)
print(y_pred_new)

#step6:
plt.scatter(x_train,y_train,color ="green")
plt.plot(x_train,y_pred,color = "red")
plt.text(20,200,"Y =%.2fbX + %.2f"%(b,a),fontsize = 15)
plt.show()
```

```
LinearRegession equation is: Y = 6.78X + -21.20
R2 : 0.86
MSE : 742.07
[1335.01 2013.11 3369.31]
```

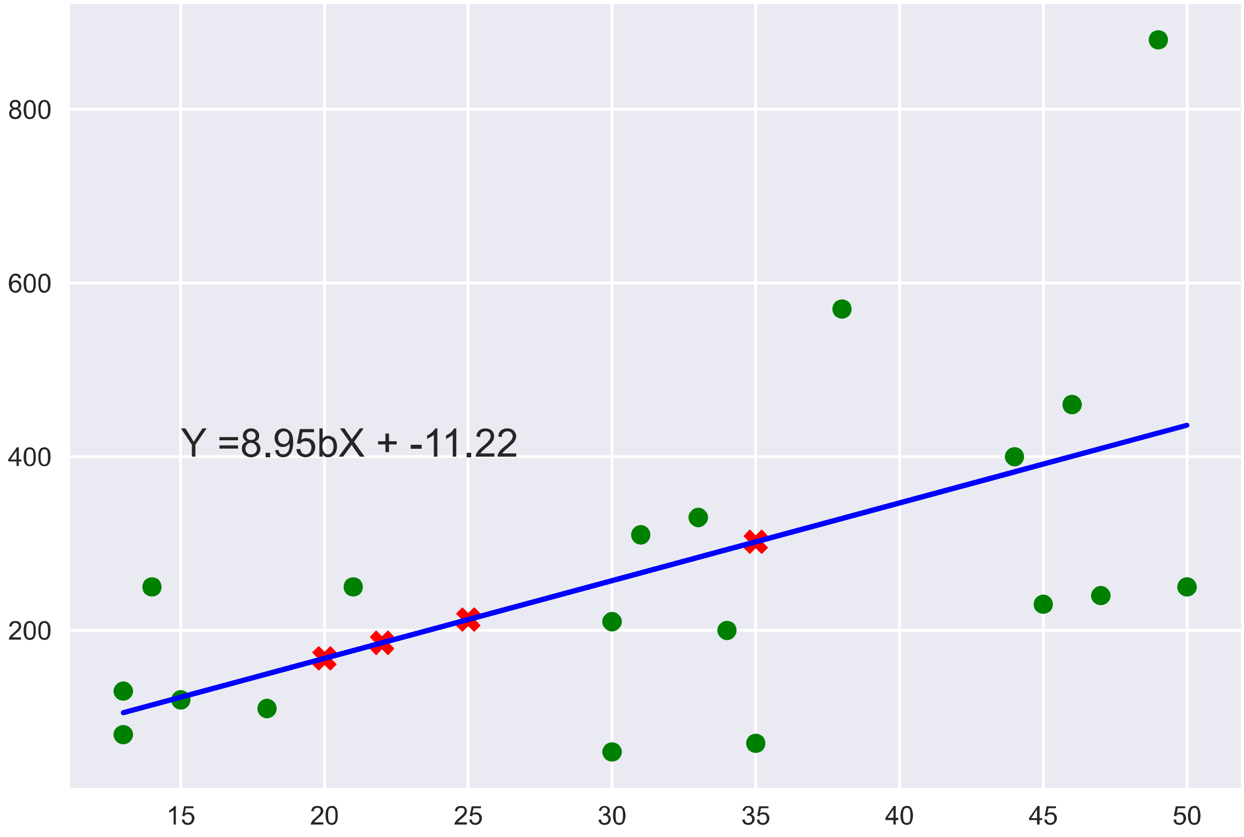


In [11]:

```
#step 1
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt
plt.style.use('seaborn')

#step 2:
x_train = np.array([34,44,18,31,13,45,15,46,13,47,46,30,30,50,38,14,21,35,33,49]).resha
y_train = np.array([200,400,110,310,130,230,120,460,80,240,460,210,60,250,570,250,250,7

#step 3:
model = LinearRegression()
model.fit(x_train,y_train)

#step 4:
b = model.coef_
a = model.intercept_
print('Coefficient(b)\t: %.2f'%(b))
```

```python
print('Intercept(a)\t: %.2f'%(a))
print("Linear equation\t: Y = %.2fX + %.2f"%(b,a))

#R2 MSE
y_pred = model.predict(x_train)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)

print("R2 : %.2f"%(r2))
print("MSE : %.2f"%(mse))

#step5:
x_new = np.array([20,22,25,35]).reshape((-1,1))
y_pred_new = model.predict(x_new)
print("\nPredicted response of X:")
print("20\t%.2f"%(y_pred_new[0]))
print("22\t%.2f"%(y_pred_new[1]))
print("25\t%.2f"%(y_pred_new[2]))
print("35\t%.2f"%(y_pred_new[3]))

#step6:
plt.scatter(x_train,y_train,color ="green")
plt.plot(x_train,y_pred,color = "blue")
plt.text(15,400,"Y =%.2fbX + %.2f"%(b,a),fontsize = 15)
plt.scatter(x_new,y_pred_new,color ="red" ,marker = "X", s=80)
plt.show()
```

```
Coefficient(b)  : 8.95
Intercept(a)    : -11.22
Linear equation : Y = 8.95X + -11.22
R2 : 0.34
MSE : 24715.26

Predicted response of X:
20      167.75
22      185.65
25      212.49
35      301.98
```

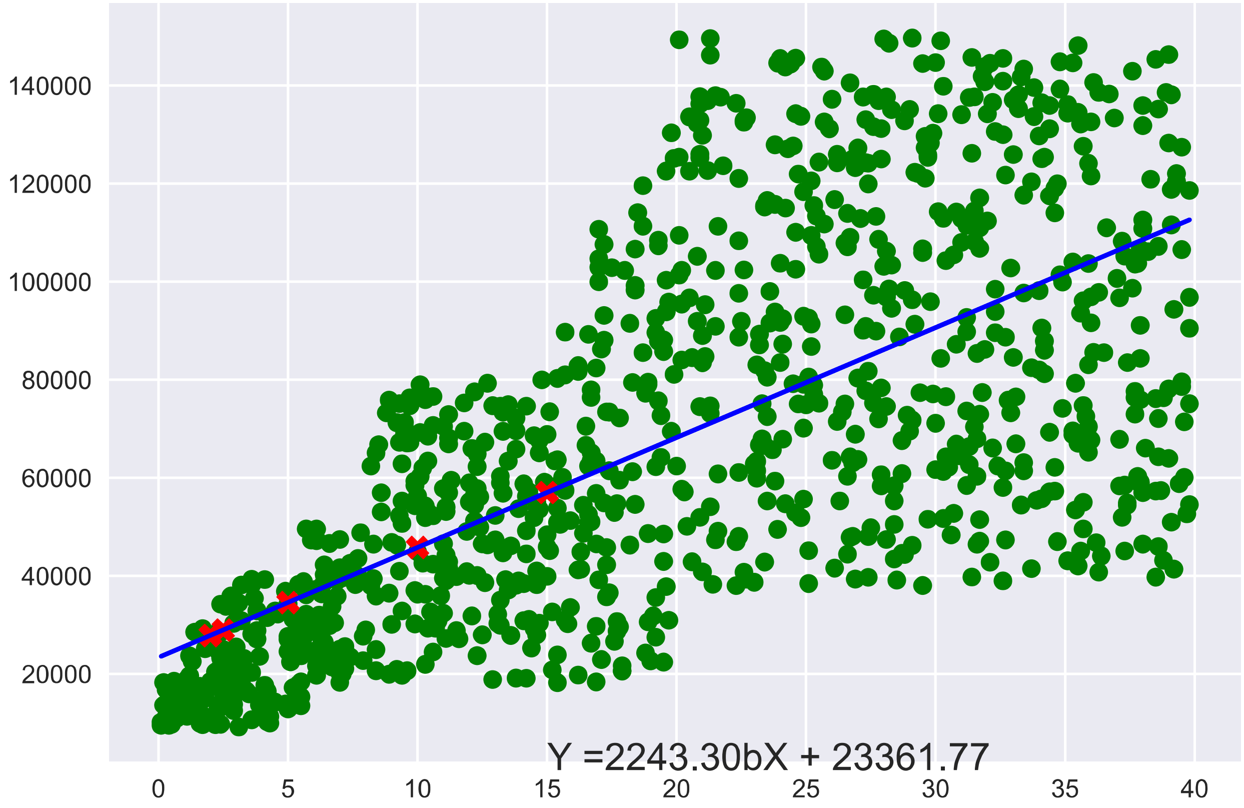


In [12]:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt
import pandas as pd
np.set_printoptions(precision=2)
plt.style.use('seaborn')

df = pd.read_csv("data/salary.csv")
print(df.head())

x_train = df[["experience"]]
y_train = df["salary"]
n = x_train.shape[0]
print("There are %i records."%(n))

#step 3:
model = LinearRegression()
model.fit(x_train,y_train)

#step 4:
b = model.coef_
a = model.intercept_
print('Coefficient(b)\t: %.2f'%(b))
print('Intercept(a)\t: %.2f'%(a))
print("Linear equation\t: Y = %.2fX + %.2f"%(b,a))

#R2 MSE
y_pred = model.predict(x_train)
r2 = r2_score(y_train,y_pred)
mse = mean_squared_error(y_train,y_pred)
```

```
print("R2 : %.2f"%(r2))
print("MSE : %.2f"%(mse))

#step5:
x_new = np.array([2,2.5,5,10,15]).reshape((-1,1))
y_pred_new = model.predict(x_new)
print("\nPredicted response of X:")
print("20\t%.2f"%(y_pred_new[0]))
print("22\t%.2f"%(y_pred_new[1]))
print("25\t%.2f"%(y_pred_new[2]))
print("35\t%.2f"%(y_pred_new[3]))
print("35\t%.2f"%(y_pred_new[4]))

#step6:
plt.scatter(x_train,y_train,color ="green")
plt.plot(x_train,y_pred,color = "blue")
plt.text(15,400,"Y =%.2fbX + %.2f"%(b,a),fontsize = 15)
plt.scatter(x_new,y_pred_new,color ="red" ,marker = "X", s=80)
plt.show()
```

```
   experience  salary
0        11.0   46576
1        16.2   82888
2        24.6  145595
3        29.6  121111
4        10.6   42415
There are 1000 records.
Coefficient(b)  : 2243.30
Intercept(a)    : 23361.77
Linear equation : Y = 2243.30X + 23361.77
R2 : 0.46
MSE : 788047186.88

Predicted response of X:
20      27848.37
22      28970.02
25      34578.27
35      45794.77
35      57011.27
```


140000
120000
100000
80000
60000
40000
20000
0
5
10
15
20
25
30
35
40
Y =2243.30bX + 23361.77

In [ ]: