In [1]:

```
from google.colab import drive     #mounting google drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

In [2]:

```
import pandas                          #to handle csv data
from pandas import DataFrame
import matplotlib.pyplot as plt        #for graph making
```

In [3]:

```
data=pandas.read_csv("/content/gdrive/MyDrive/cost_revenue_dirty.csv")   #importing csv file form Drive
data
```

Out[3]:

	production_budget_usd	worldwide_gross_usd
0	1000000.0	2.600000e+01
1	10000.0	4.010000e+02
2	400000.0	4.230000e+02
3	750000.0	4.500000e+02
4	10000.0	5.270000e+02
...	...	...
5029	225000000.0	1.519480e+09
5030	215000000.0	1.671641e+09
5031	306000000.0	2.058662e+09
5032	200000000.0	2.207616e+09
5033	425000000.0	2.783919e+09

5034 rows × 2 columns

In [4]:

```
data.describe()        #this describe data in brief
```

Out[4]:

	production_budget_usd	worldwide_gross_usd
count	5.034000e+03	5.034000e+03
mean	3.290784e+07	9.515685e+07
std	4.112589e+07	1.726012e+08
min	1.100000e+03	2.600000e+01
25%	6.000000e+06	7.000000e+06
50%	1.900000e+07	3.296202e+07
75%	4.200000e+07	1.034471e+08
max	4.250000e+08	2.783919e+09

In [5]:

```
5.034e3     #convert data from scientific notation to normal number
```

Out[5]:

```
5034.0
```

In [11]:

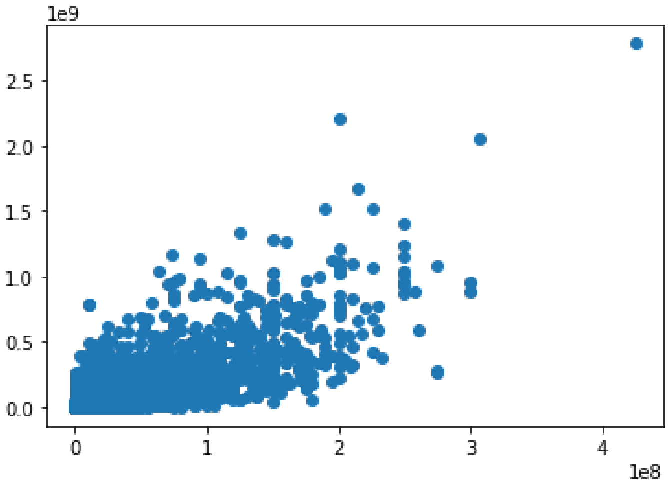
```
X=DataFrame(data,columns=['production_budget_usd'])     #use exact name from columns
y=DataFrame(data,columns=['worldwide_gross_usd'])
```

In [13]:

```
plt.scatter(X,y)  #make graph between X,Y  X->Independent variable(Feature) ,Y-> Dependent variable(Target)
```

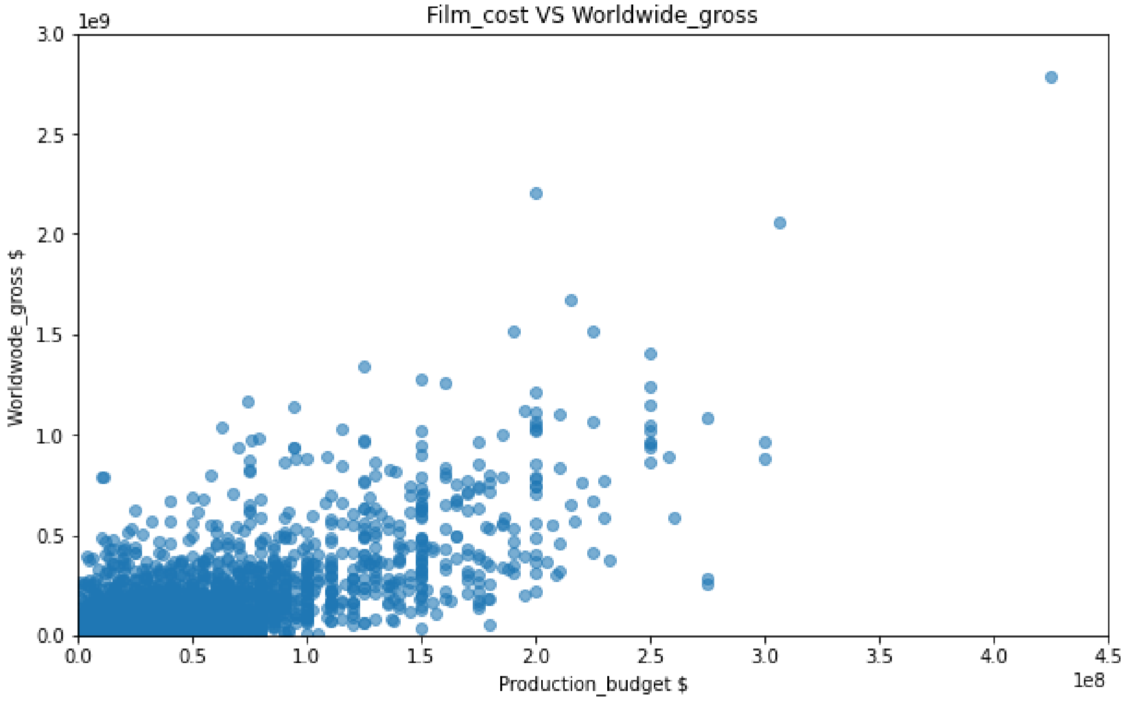
Out[13]:

```
<matplotlib.collections.PathCollection at 0x7f8a3d2cf7d0>
```



In [14]:

```
plt.figure(figsize=(10,6))
plt.scatter(X,y,alpha=0.6)  #make graph between X,y  X->Independent variable(Feature) ,y-> Dependent variable(Target)
plt.title("Film_cost VS Worldwide_gross")   #title to graph
plt.xlabel("Production_budget $")    #give name to x axis
plt.ylabel("Worldwode_gross $")
plt.xlim(0,450000000)      #remove blank space in x axis, set range for value of x
plt.ylim(0,3000000000)
plt.show()           #to show the graph
```



Linear Regression

In [15]:

```
from sklearn.linear_model import LinearRegression
```

In [16]:

```
regression=LinearRegression()
regression.fit(X,y)
```

Out[16]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [17]:

```
#theta 1 or slope
regression.coef_

#interpretation => for each dollar as budget you will earn 3 dollars in revenue
```

Out[17]:

```
array([[3.11150918]])
```

In [18]:

```
#theta 0 or  intercept
regression.intercept_
```

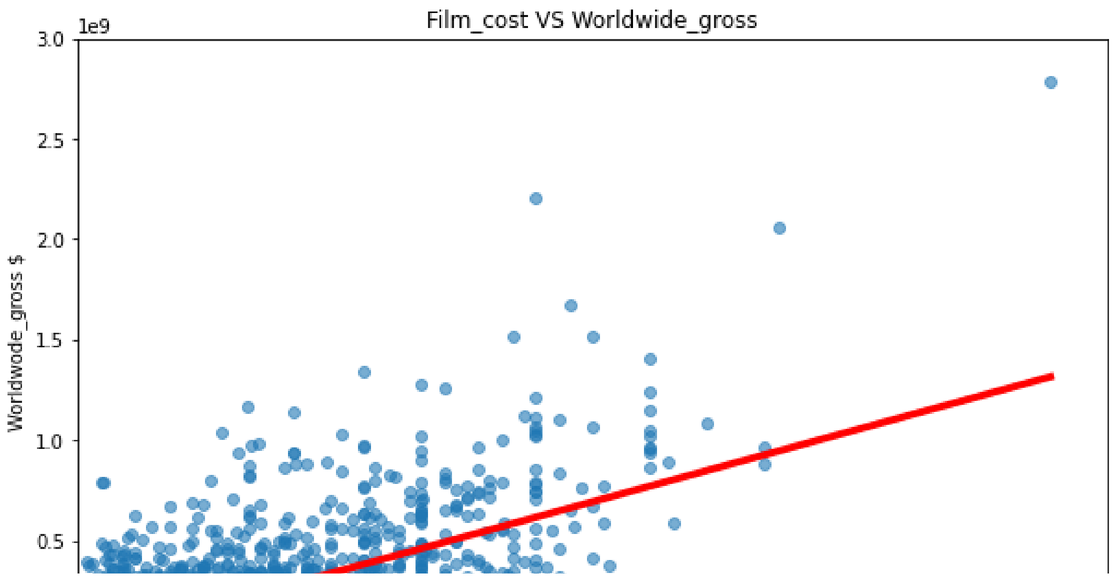
Out[18]:

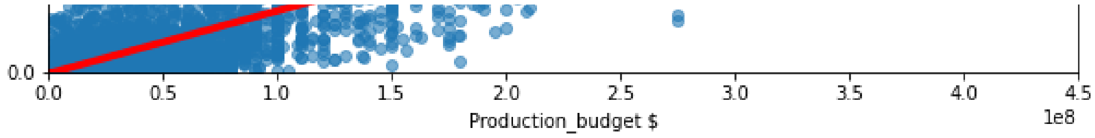
```
array([-7236192.72913958])
```

Model plotting

In [19]:

```
plt.figure(figsize=(10,6))
plt.scatter(X,y,alpha=0.6)                                  # make graph between X,y
X->Independent variable(Feature) ,y-> Dependent variable(Target)
plt.plot(X,regression.predict(X),color='red',linewidth=4)    # .plot is used to plot a
line
plt.title("Film_cost VS Worldwide_gross")                   # title to graph
plt.xlabel("Production_budget $")                           # give name to x axis
plt.ylabel("Worldwode_gross $")
plt.xlim(0,450000000)                                       # remove blank space in x
axis, set range for value of x
plt.ylim(0,3000000000)
plt.show()                                                  # to show the graph
```





Goodness of fit The goodness of fit of a statistical model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question

In [20]:

```
regression.score(X,y)
```

Out[20]:

```
0.5496485356985729
```