Two variable Linear Regression Model:

Following Assumptions were taken:

1) The regression model is linear in parameters.

2) Zero mean value of disturbance.

3) Zero covariance between the independent term and disturbance term.

4) Homoscedasticity is present i.e. variance of disturbance term is constant.

5) No autocorrelation between the disturbance terms.

Setting Hypothesis:

Null Hypothesis, Ho:There is no significant difference between two variables i.e. there is no relation exist between dependent and independent variables.

Alternative Hypothesis, H1:There is significant difference between two variables i.e. there is some relation exist between dependent and independent variables.

Test Statistic:

Importing Libraries:

In [1]:

```
import math
import matplotlib.pyplot as plt
import pandas as pd
```

Assigning values to the variables:

In [2]:

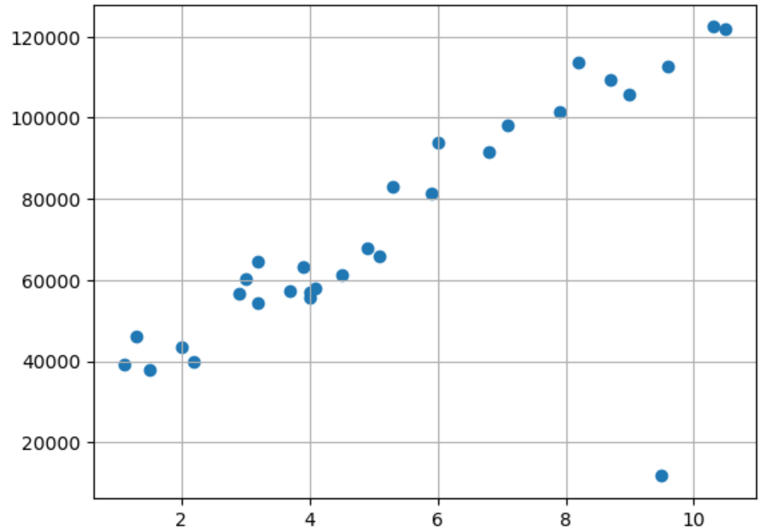
```
with open("file.txt","r") as f:  #To read data from .txt file
    data=f.readlines()
    l=[]
    for i in data:
       l.append(i.strip().split(","))
    X = [eval(i) for i in l[0][1:]] #Independent Variable
    Y = [eval(i) for i in l[1][1:]] #Dependent Variable
    print("Y =",Y)
    print("X =",X)
```

```
Y = [39343, 46205, 37731, 43525, 39891, 56642, 60150, 54445, 64445, 57189, 63218, 55794, 56957, 57891, 61111, 67938, 66029,
83088, 81363, 93940, 91738, 98273, 101302, 113812, 109431, 105582, 11696, 112635, 122391, 121872]
X = [1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.2, 3.7, 3.9, 4.0, 4.0, 4.1, 4.5, 4.9, 5.1, 5.3, 5.9, 6.0, 6.8, 7.1, 7.9, 8.
2, 8.7, 9.0, 9.5, 9.6, 10.3, 10.5]
```

Scatter plot of given data:

In [3]:

```
plt.scatter(X,Y)
plt.grid()
plt.show()
```



Calculating mean of X(Independent variable):

In [4]:

```
sum_X=0
for i in range(len(X)):
    sum_X=sum_X+X[i]
mean_X=sum_X/len(X)
print("Mean of X(X̄)=",mean_X)
```

```
Mean of X(X̄)= 5.3133333333333335
```

Squaring and Summing over X i.e. ΣX^2:

In [5]:

```
sum_X2=0
for i in range(len(X)):
    sum_X2=sum_X2+(X[i]*X[i])
print(" ΣX^2 =",sum_X2)
```

```
 ΣX^2 = 1080.5
```

Calculating mean of Y(Dependent variable):

In [6]:

```
sum_Y=0
for j in range(len(Y)):
    sum_Y=sum_Y+Y[j]
mean_Y=sum_Y/len(Y)
print("Mean of Y(Ȳ) =",mean_Y)
```

```
Mean of Y(Ȳ) = 72520.9
```

Squaring and Summing over Y i.e.ΣY^2:

In [7]:

```
sum_Y2=0
for j in range(len(Y)):
    sum_Y2=sum_Y2+(Y[j]*Y[j])
print("ΣY^2 =",sum_Y2)
```

```
ΣY^2 = 181636834897
```

Calculating deviation of X from it's mean i.e. x:

In [8]:

```
x=[]
for i in range(len(X)):
    dx=X[i]-mean_X
    x.append(dx)
print("(X-X̄) =",x)
```

```
(X-X̄) = [-4.213333333333333, -4.013333333333334, -3.8133333333333335, -3.3133333333333335, -3.1133333333333333, -2.41333333
33333336, -2.3133333333333335, -2.1133333333333333, -2.1133333333333333, -1.6133333333333333, -1.4133333333333336, -1.31333
33333333335, -1.3133333333333335, -1.2133333333333338, -0.8133333333333335, -0.4133333333333331, -0.21333333333333382, -0.0
13333333333333641, 0.5866666666666669, 0.6866666666666665, 1.4866666666666664, 1.7866666666666662, 2.586666666666667, 2.886
666666666666, 3.386666666666666, 3.6866666666666665, 4.1866666666666665, 4.286666666666666, 4.986666666666667, 5.1866666666
666665]
```

Square of Deviation of X i.e. x^2:

In [9]:

```
x2=[]
sum_x2=0
for k in range(len(x)):
    dx2=x[k]**2
    x2.append(dx2)
    sum_x2=sum_x2+dx2
print("Σ(X-X̄)^2 =",sum_x2)
```

```
Σ(X-X̄)^2 = 233.55466666666663
```

Calculating deviation of Y from it's mean i.e. y:

In [10]:

```
y=[]
for j in range(len(Y)):
    dy=Y[j]-mean_Y
    y.append(dy)
print("(Y-Ȳ) =", y)
```

```
(Y-Ȳ) = [-33177.899999999994, -26315.899999999994, -34789.899999999994, -28995.899999999994, -32629.899999999994, -15878.89
9999999994, -12370.899999999994, -18075.899999999994, -8075.899999999994, -15331.899999999994, -9302.899999999994, -16726.8
99999999994, -15563.899999999994, -14629.899999999994, -11409.899999999994, -4582.899999999994, -6491.899999999994, 10567.1
00000000006, 8842.100000000006, 21419.100000000006, 19217.100000000006, 25752.100000000006, 28781.100000000006, 41291.10000
0000006, 36910.100000000006, 33061.100000000006, -60824.899999999994, 40114.100000000006, 49870.100000000006, 49351.1000000
00006]
```

Square of Deviation of Y i.e. y^2:

In [11]:

```
y2=[]
sum_y2=0
for l in range(len(y)):
    dy2=y[l]**2
    y2.append(dy2)
    sum_y2=sum_y2+dy2
print("Σ(Y-Ȳ)^2 =",sum_y2)
```

```
Σ(Y-Ȳ)^2 = 23858406792.7
```

Product of deviation of X and Y i.e. xy and it's sum i.e. Σxy:

In [12]:

```
xy=[]
sum_xy=0
for m in range(len(x)):
    dxy=x[m]*y[m]
    xy.append(dxy)
    sum_xy=sum_xy+dxy
print("Σxy =",sum_xy)
```

```
Σxy = 1765357.0399999996
```

Calculating β2_cap:

In [13]:

```
B2_cap=sum_xy/sum_x2
print("Estimated value of β2 =", B2_cap)
```

```
Estimated value of β2 = 7558.645970108353
```

Calculating β1_cap:

In [14]:

```
B1_cap=mean_Y-(B2_cap*mean_X)
print("Estimated value of β1 =",B1_cap)
```

```
Estimated value of β1 = 32359.294412157607
```

Calculating estimated Y i.e. Y_cap:

In [15]:

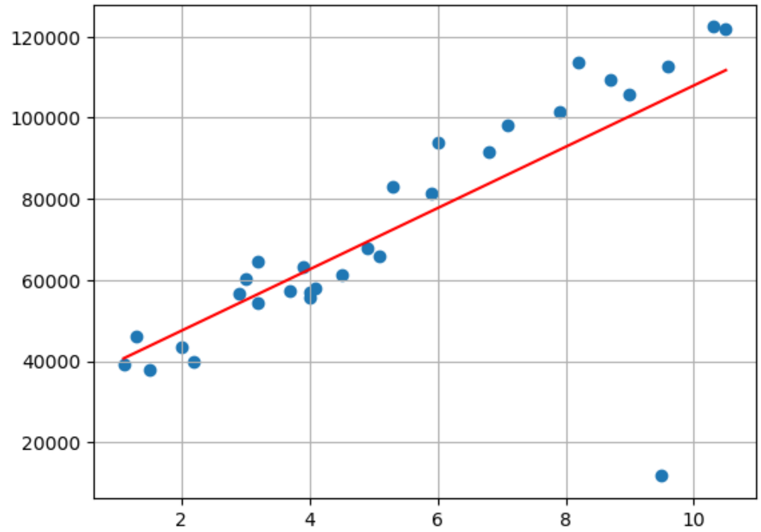
```
Y_cap=[]
for i in range(len(X)):
    Y_=B1_cap+(B2_cap*X[i])
    Y_cap.append(Y_)
print("Estimated value of Y =",Y_cap)
```

```
Estimated value of Y = [40673.8049792768, 42185.53417329847, 43697.263367320134, 47476.58635237432, 48988.315546395985, 542
79.367725471835, 55035.23232248267, 56546.961516504336, 56546.961516504336, 60326.28450155852, 61838.013695580186, 62593.87
829259102, 62593.87829259102, 63349.742889601854, 66373.20127764519, 69396.65966568854, 70908.3888597102, 72420.1180537318
9, 76955.30563579689, 77711.17023280772, 83758.0870088944, 86025.68079992692, 92072.59757601359, 94340.19136704609, 98119.5
1435210026, 100387.10814313279, 104166.43112818696, 104922.2957251978, 110213.34790427366, 111725.07709829533]
```

Plotting Regression Line:

In [16]:

```
plt.scatter(X,Y)
plt.plot(X,Y_cap,'r')
plt.grid()
plt.show()
```



Calculating Stochastic Residual term i.e. μ_cap:

In [17]:

```
u_cap=[]
for n in range(len(Y)):
    e=Y[n]-Y_cap[n]
    u_cap.append(e)
print("Estimated value of error(μ) =",u_cap)
```

```
Estimated value of error(μ) = [-1330.8049792767997, 4019.465826701533, -5966.263367320134, -3951.5863523743174, -9097.31554
6395985, 2362.632274528165, 5114.767677517331, -2101.961516504336, 7898.038483495664, -3137.284501558519, 1379.986304419813
7, -6799.87829259102, -5636.87829259102, -5458.742889601854, -5262.201277645188, -1458.6596656885376, -4879.388859710205, 1
0667.881946268113, 4407.694364203111, 16228.829767192277, 7979.912991105593, 12247.319200073078, 9229.402423986408, 19471.8
08632953907, 11311.485647899739, 5194.891856867209, -92470.43112818696, 7712.704274802207, 12177.652095726342, 10146.922901
704675]
```

Calculating Square of Stochastic Residual term i.e. μ^2 _cap and it's Sum i.e. Σμ^2_cap :

In [18]:

```
u_cap2=[]
sum_u2=0
for c in range(len(u_cap)):
    e_2=u_cap[c]**2
    u_cap2.append(e_2)
    sum_u2=sum_u2+e_2
print("Σμ^2_cap =",sum_u2)
```

```
Σμ^2_cap = 10514697916.501587
```

Calculating Sigma cap Square i.e. σ^2_cap:

In [19]:

```
sigma_2=sum_u2/(len(X)-2)
print("σ^2_cap =",sigma_2)
```

```
σ^2_cap = 375524925.5893424
```

Calculating Variance of β2_cap:

In [20]:

```
var_B2_cap=sigma_2/sum_x2
print("Variance of β2_cap =",var_B2_cap)
```

```
Variance of β2_cap = 1607867.361200272
```

Calculating Standard Error of β2_cap:

In [21]:

```
SD_B2_cap=math.sqrt(var_B2_cap)
print("Standard Error of β2_cap =",SD_B2_cap)
```

```
Standard Error of β2_cap = 1268.017098149813
```

T-Statistic for β2:

In [22]:

```
T_B2=B2_cap/SD_B2_cap
print("T-test of β2_cap =",T_B2)
```

```
T-test of β2_cap = 5.960996883352214
```

Calculating Variance of β1_cap:

In [23]:

```
var_B1_cap=sigma_2*(sum_X2/(len(X)*sum_x2))
print("Variance of β1_cap =",var_B1_cap)
```

```
Variance of β1_cap = 57910022.792563125
```

Calculating Standard Error of β1_cap:

In [24]:

```
SD_B1_cap=math.sqrt(var_B1_cap)
print("Standard Error of β1_cap =",SD_B1_cap)
```

```
Standard Error of β1_cap = 7609.863519969535
```

T-Statistic for β1:

In [25]:

```
T_B1=B1_cap/SD_B1_cap
print("T-test of β1_cap =",T_B1)
```

```
T-test of β1_cap = 4.252283149000174
```

Calculating Explained Sum of Squares i.e. ESS:

In [26]:

```
ESS=B2_cap*B2_cap*sum_x2
print("Explained Sum of Squares =",ESS)
```

```
Explained Sum of Squares = 13343708876.198406
```

Calculating Residual Sum of Squares i.e. RSS:

In [27]:

```
RSS=sum_u2
print("Residual sum of Squares =",RSS)
```

```
Residual sum of Squares = 10514697916.501587
```

Calculating Total Sum of Squares i.e. TSS:

In [28]:

```
TSS=ESS+RSS
print("Total Sum of Squares =",TSS)
```

```
Total Sum of Squares = 23858406792.699993
```

Calculating Coefficient of Determination i.e. r^2:

In [29]:

```
r_2=ESS/TSS
print("Regression Coefficient(r^2) =",r_2)
```

```
Regression Coefficient(r^2) = 0.559287507843198
```

ANOVA Table:

In [30]:

```
data={"Source of Variation":["Due to Regression(ESS)","Due to Residuals(RSS)","Total sum of Squares(TSS)"],
      "Sum of Squares":[ESS,RSS,TSS],
      "Degree of Freedom":[1,len(X)-2,len(X)-1],
      "Mean Sum of Squares":[ESS/1,RSS/(len(X)-2),TSS/(len(X)-1)]}
df=pd.DataFrame(data)
df.head()
```

Out[30]:

	Source of Variation	Sum of Squares	Degree of Freedom	Mean Sum of Squares
0	Due to Regression(ESS)	1.334371e+10	1	1.334371e+10
1	Due to Residuals(RSS)	1.051470e+10	28	3.755249e+08
2	Total sum of Squares(TSS)	2.385841e+10	29	8.227037e+08

F-Statistic:

In [31]:

```
F=(ESS/1)/(RSS/(len(X)-2))
print("F-test =",F)
```

```
F-test = 35.533483843334814
```

Level of significance: ¶

Take level of significance as 0.05 or 5%.

Critical Values of T-statistic and F-statistic:

Tabulated Values of T-statistic:

In [32]:

```
t_tabulated=[12.71,4.303,3.182,2.776,2.571,2.447,2.365,2.306,2.262,2.228,2.201,2.179,2.160,2.145,2.131,2.120,2.110,2.101,2.093,2.086,2.08
```

Tabulated Values of F-statistic:

In [33]:

```
f_tabulated=[161.4476,18.5128,10.1280,7.7086,6.6079,5.9874,5.5914,5.3177,5.1174,5.9646,4.8443,4.7472,4.6672,4.6001,4.5431,4.4940,4.4513,4
```

Confidence Interval:

In [34]:

```
CI_1=B2_cap+(t_tabulated[(len(X)-2)+1]*SD_B2_cap)
CI_2=B2_cap-(t_tabulated[(len(X)-2)+1]*SD_B2_cap)
if (CI_1>CI_2):
    print("Range of confidence interval of B1_cap will be from ",CI_2," to ",CI_1)
else:
    print("Range of confidence interval of B1_cap will be from ",CI_1," to ",CI_2)
```

```
Range of confidence interval of B1_cap will be from  4969.355055686436  to  10147.93688453027
```

Result:

In [35]:

```
print(f"Y_cap  = ({B1_cap}) + ({B2_cap}) X_cap \t r_2 = {r_2}")
print(f"S.E.   = ({SD_B1_cap})   ({SD_B2_cap}) \t\t d.f. ={len(X)-2} ")
print(f"T-test = ({T_B1})   ({T_B2}) \t\t t_tab = {t_tabulated[(len(X)-2)+1]}")
```

```
Y_cap  = (32359.294412157607) + (7558.645970108353) X_cap        r_2 = 0.559287507843198
S.E.   = (7609.863519969535)   (1268.017098149813)               d.f. =28
T-test = (4.252283149000174)   (5.960996883352214)               t_tab = 2.042
```

Interpretation:

In [36]:

```
print("1)")
print("For β1_cap:")
print("Calculated t-value = ",T_B1," and Tabulate t-value = ",t_tabulated[(len(X)-2)+1])
if (B1_cap>t_tabulated[(len(X)-2)+1]):
                       print("Null Hypothesis is Rejected for β1_cap i.e. There is significant difference between two variables")
else:
                       print("Null Hypothesis is Accepted for β1_cap i.e. there is no significant difference between two variables")
print()
print()
print("2)")
print("For β2_cap:")
print("Calculated t-value = ",T_B2," and Tabulate t-value = ",t_tabulated[(len(X)-2)+1])
if (B2_cap>t_tabulated[(len(X)-2)+1]):
                       print("Null Hypothesis is Rejected for β2_cap i.e. There is significant difference between two variables")
else:
                       print("Null Hypothesis is Accepted for β2_cap i.e. there is no significant difference between two variables")

print()
print()
print("3)")
print("All over")
print("Calculated f-value = ",F," and Tabulate f-value = ",f_tabulated[(len(X)-2)+1])
if (F>f_tabulated[(len(X)-2)+1]):
                       print("Null Hypothesis is Rejected for i.e. There is significant difference between two variables")
else:
                       print("Null Hypothesis is Accepted for i.e. there is no significant difference between two variables")

print()
print()
print("4)")
print("Point Estimation")
print("If there is a change of one unit in the value of X variable then the change in Y variable will be = ",B2_cap)

print()
print()
print("5)")
print("Interval Estimation")
if (CI_1>CI_2):
    print("If there is a change of one unit in the value of X variable then the change in Y variable will be in range from: \n",CI_2," to
else:
    print("If there is a change of one unit in the value of X variable then the change in Y variable will be in range from; \n ",CI_1," to


print()
print()
print("6)")
print("Coefficient of Determination is ",r_2," i.e. our model is ",100*r_2," explained.")

print()
print()
print("7)")
print("We assumed 5% Level of Significance i.e. there are 95% chances that our estimated model is correct.")
```

```
1)
For β1_cap:
Calculated t-value =  4.252283149000174  and Tabulate t-value =  2.042
Null Hypothesis is Rejected for β1_cap i.e. There is significant difference between two variables


2)
For β2_cap:
Calculated t-value =  5.960996883352214  and Tabulate t-value =  2.042
Null Hypothesis is Rejected for β2_cap i.e. There is significant difference between two variables


3)
All over
Calculated f-value =  35.533483843334814  and Tabulate f-value =  4.1709
Null Hypothesis is Rejected for i.e. There is significant difference between two variables


4)
Point Estimation
If there is a change of one unit in the value of X variable then the change in Y variable will be =  7558.645970108353


5)
Interval Estimation
If there is a change of one unit in the value of X variable then the change in Y variable will be in range from:
 4969.355055686436  to  10147.93688453027


6)
Coefficient of Determination is  0.559287507843198  i.e. our model is  55.9287507843198  explained.


7)
We assumed 5% Level of Significance i.e. there are 95% chances that our estimated model is correct.
```