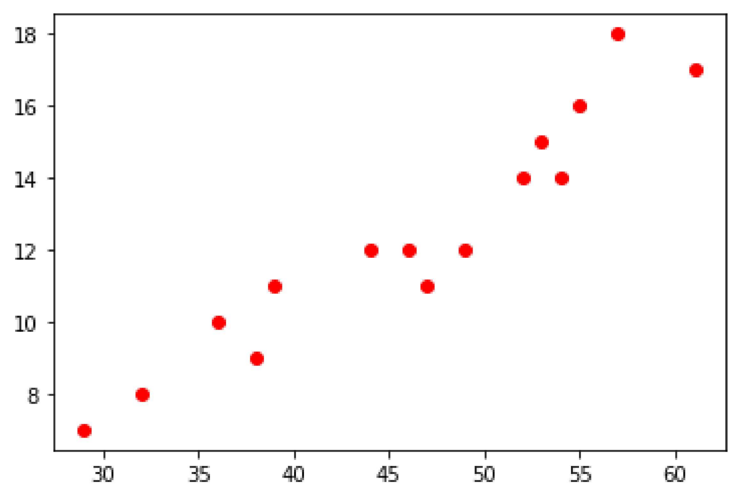
Least Squares Estimators of Regression Parameters

Example The raw material used in the production of a certain synthetic fiber is stored in a place without humidity control. The relative humidity measurements in the warehouse and the moisture content of a sample of the raw material are taken for 15 days (as a percentage) in the data list below.

- relative_humidity = [46, 53, 29, 61, 36, 39, 47, 49,52, 38, 55, 32, 57, 54, 44]
- content_of_moisture = [12, 15, 7, 17, 10, 11, 11, 12, 14, 9, 16, 8, 18, 14, 12]

In [53]:

```
from LinearRegression import RegressionParameters
import matplotlib.pyplot as plt

relative_humidity = [46, 53, 29, 61, 36, 39, 47, 49,52, 38, 55, 32, 57, 54, 44]
content_of_moisture = [12, 15, 7, 17, 10, 11, 11, 12, 14, 9, 16, 8, 18, 14, 12]

plt.scatter(relative_humidity,content_of_moisture,color='red')
plt.show()
```



Let's create a regression line against the scatter diagram we have drawn above. For this, let's obtain the regression parameters alpha and beta.

In [54]:

```
alpha = RegressionParameters.A(relative_humidity,content_of_moisture)
beta = RegressionParameters.B(relative_humidity,content_of_moisture)

alpha , beta
```

Out[54]:

```
(-2.510457651687677, 0.32320356181403925)
```

- Regression Equation --> alpha + beta * x

In [55]:

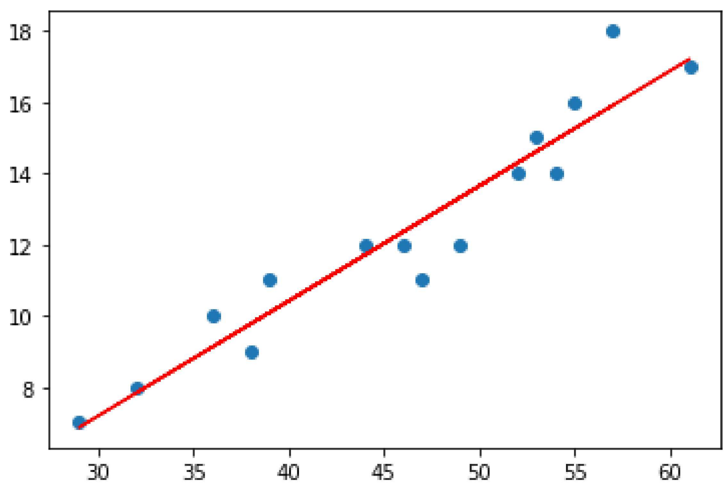
```
# They are output according to the regression equation.
reg_Y = [(i*beta+alpha) for i in relative_humidity]

reg_Y
```

Out[55]:

```
[12.356906191758128,
 14.619331124456405,
 6.862445640919461,
 17.20495961896872,
 9.124870573617736,
 10.094481259059854,
 12.680109753572168,
 13.326516877200246,
 14.296127562642365,
 9.771277697245814,
 15.265738248084482,
 7.832056326361579,
 15.912145371712562,
 14.942534686270442,
 11.71049906813005]
```

In [56]:

```
plt.scatter(relative_humidity,content_of_moisture)
plt.plot(relative_humidity,reg_Y,color='red',linewidth=1.3)
plt.show()
```



Sum of Squares for Residuals

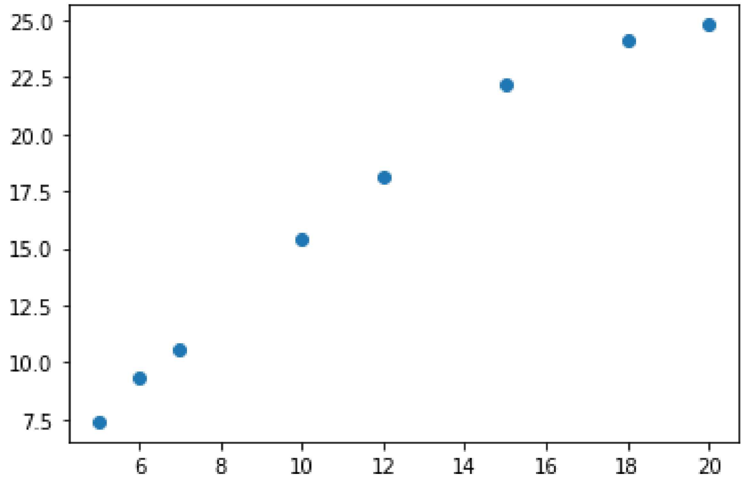
Example

There are data on x and Y. x is the humidity of the wet mix of a particular product. Y is dense of a finished product.

- x = [5, 6, 7, 10, 12, 15, 18, 20]
- Y = [7.4, 9.3, 10.6, 15.4, 18.1, 22.2, 24.1, 24.8]

In [57]:

```
x = [5, 6, 7, 10, 12, 15, 18, 20]
Y = [7.4, 9.3, 10.6, 15.4, 18.1, 22.2, 24.1, 24.8]

plt.scatter(x,Y)
plt.show()
```



In [58]:

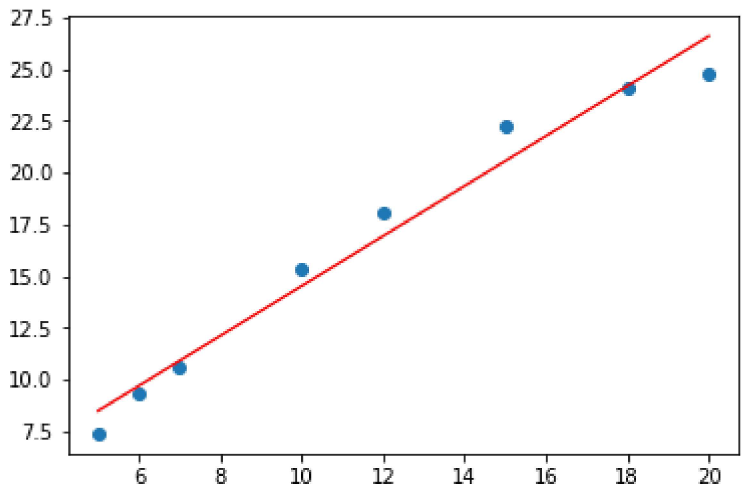
```
alpha = RegressionParameters.A(x,Y)
beta = RegressionParameters.B(x,Y)

# They are output according to the regression equation.
reg_Y =[(alpha +beta * i)for i in x]
reg_Y
```

Out[58]:

```
[8.495323943661976,
 9.701690140845074,
 10.908056338028173,
 14.527154929577467,
 16.939887323943662,
 20.558985915492958,
 24.17808450704225,
 26.59081690140845]
```

In [59]:

```
plt.scatter(x,Y)
plt.plot(x,reg_Y,color='red',linewidth=1.3)
plt.show()
```



In [60]:

```
# Sum of Squares for Residuals(SSR)

ssr = RegressionParameters.Ssr(x,Y)
```

```
ssr
```

Out[60]: 9.469757746479173

Hypothesis Testing for $H_0 : \beta = 0$

Example One person claims that the fuel consumption of his car is not dependent on the car's speeding. To test the plausibility of this hypothesis, the car was tested at various speeds between 45 and 75 miles per hour. The miles per gallon achieved at each of these speeds were determined.

- speed = [45, 50, 55, 60, 65, 70, 75]
- miles = [24.2, 25, 23.3, 22, 21.5, 20.6, 19.8]

In [61]:

```
speed = [45, 50, 55, 60, 65, 70, 75]
miles = [24.2, 25, 23.3, 22, 21.5, 20.6, 19.8]

ssr =RegressionParameters.Ssr(speed,miles)
ssr
```

Out[61]: 1.5271428571412253

- Test Statistics for Beta=0

In [62]:

```
from LinearRegression import CoefHypoTest

test_stat = CoefHypoTest.B_hypot(speed,miles,0)
test_stat
```

Out[62]: 8.138476446948063

After the result is compared with the t-table value, a conclusion can be obtained about the validity of the hypothesis.

Confidence Interval for β

$$\left(B - \sqrt{\frac{SS_R}{(n-2)S_{xx}}} t_{\alpha/2,n-2}, B + \sqrt{\frac{SS_R}{(n-2)S_{xx}}} t_{\alpha/2,\right.$$

Derive the 95% confidence interval estimate of beta for the above example.

In [63]:
```
from LinearRegression.ConfidenceInterval import *

conf = B_Conf_Interval(speed,miles,0.05)
conf
```
Out[63]:
```
[-0.22369542015304456, -0.11630457984696062]
```

It can be said with 95% confidence that the beta is between -.224 and -.116.

Example for $\beta = 1$

In [64]:
```
dad_length = [60, 62, 64, 65, 66, 67, 68, 70, 72, 74]
boy_length = [63.6, 65.2, 66, 65.5, 66.9, 67.1, 67.4, 68.3, 70.1, 70]

test_stat = CoefHypoTest.B_hypot(dad_length,boy_length,1)
test_stat
```
Out[64]:
```
-16.232797516332063
```

Confidence Interval Estimator for $\alpha + \beta x_0$

$$A + Bx_0 \pm \sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}} \sqrt{\frac{SS_R}{n-2}} t_{\alpha/2, n-2}$$

Example : Determine the 95% confidence interval for the average height of all men whose fathers are 68 inches tall, according to the example above.

In [65]:
```
from LinearRegression.ConfidenceInterval import Alpha_BetaX0_Conf_Interval

conf = Alpha_BetaX0_Conf_Interval(dad_length,boy_length,68,0.05)
conf
```
Out[65]:
```
[67.23944245539089, 67.89552257957403]
```

$$\alpha + \beta x_0 \in (67.239, 67.896)$$

In [66]:
```
alpha = A(dad_length,boy_length)
beta = B(dad_length,boy_length)

y = alpha + beta * 68
y # A + B *x0 regression output result for x0 = 68
```
Out[66]:
```
67.56748251748246
```

Response Prediction Interval For Input x_0

$$A + Bx_0 \pm t_{\alpha/2,n-2}\sqrt{\left[\frac{n+1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right]\frac{SS_R}{n-2}}$$

Based on the Yi response values corresponding to the Xi input values, the Yi response based on the x0 input will fall within the above range.

Example : In the example above, let's calculate a range that would be 95% reliable and would cover a certain man's height with a father who is 68 inches tall.

In [67]:

```
conf = X0_Conf_Interval(dad_length,boy_length,68,0.05)
conf
```

Out[67]:

```
[66.51848207926423, 68.61648295570069]
```

$$Y(68) \in (66.518, 68.616)$$

Coefficient of Determination (R^2)

Example : Let's look at the relationship of his son's height to his father's height.

In [68]:

```
from LinearRegression.R2 import R2

r2 = R2(dad_length,boy_length)

r2
```

Out[68]:

```
0.9612349635708113
```

96% of the change in height of 10 individuals is explained by their father's height. The remaining 4% change, which cannot be explained, is due to the change in his son's height, although the father's height is taken into account.

Weighted Least Squares

Example : The following data shows the transportation times of a particular city to the city center area. The independent variable (input) is the distance traveled.

- miles = [0.5, 1, 1.5, 2, 3, 4, 5, 6, 8, 10]
- minute = [15, 15.1, 16.5, 19.9, 27.7, 29.7, 26.7, 35.9, 42, 49.4]

In [69]:

```
from LinearRegression.WeightedLeastSquares import Weighted_Squares

miles = [0.5, 1, 1.5, 2, 3, 4, 5, 6, 8, 10]
minute = [15, 15.1, 16.5, 19.9, 27.7, 29.7, 26.7, 35.9, 42, 49.4]

ws = Weighted_Squares(miles,minute)

ws
```

```
alpha = 12.55448498004342 , beta = 3.715979273160141
Regression equation: Y = 12.55448498004342 + 3.715979273160141 * x
```

Out[69]:

```
array([[12.55448498],
       [ 3.71597927]])
```

In [70]:

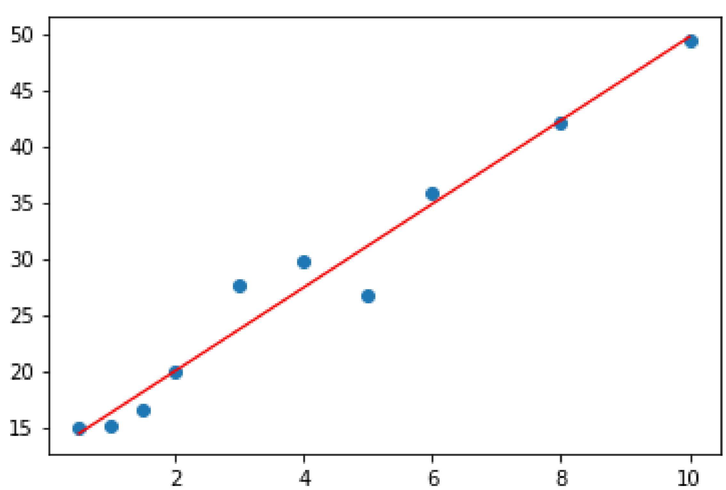
```
alpha, beta = ws[0,0] ,ws[1,0]
Y = [(alpha + beta*i) for i in miles]
Y
```

Out[70]:

```
[14.41247461662349,
 16.27046425320356,
 18.128453889783632,
 19.9864435263637,
 23.702422799523845,
 27.418402072683982,
 31.134381345844126,
 34.85036061900426,
 42.28231916532455,
 49.71427771164483]
```

In [71]:

```
plt.scatter(miles,minute)
plt.plot(miles,Y,color='red',linewidth=1.3)
plt.show()
```



The graph of the 12,554 + 3.716x estimated regression line with the data points is above. As a qualitative check of our result, the regression line best fits the data pairs as

it should when the input values are small, as the weights are inversely proportional to the inputs.