In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import *
```

In [17]:

```
dict={

    'x':np.arange(1,50,4),
    'y':np.random.choice([1,0],size=13)
}
```

In [19]:

```
dataFrame=pd.DataFrame(dict)
dataFrame
```

Out[19]:

	x	y
0	1	1
1	5	1
2	9	1
3	13	0
4	17	1
5	21	0
6	25	1
7	29	0
8	33	1
9	37	0
10	41	0
11	45	1
12	49	0

$$xi = mx + c$$

$$sigmoid = \frac{1}{1 + e^{-xi}}$$

In [24]:

```
class Logistic_Regression():
    def __inint__(self,dataframe,m,c):
        self.dataframe=dataframe
        self.y_org=0
        self.y_pred=0

    def sigmoid(self):
        self.m=3.4
        self.c=-0.6
        X=[]
        for value in dataframe['x'].values:
            X.append(self.m*(value)+self.c)
        pre=[]
        for v in X:
            pre.append(1/(1+np.exp(-v)))
        self.y_pred=[]
        for pri in pre:
            if pri > 0.5:
                self.y_pred.append(1)
            else:
                y_pred.append(0)
        self.y_org=dataframe['y'].values
        return self.y_org , self.y_pred

    def accuracy(self):
        self.correct=0
        for Y_org,Y_pre in zip(self.y_org,self.y_pred):
            if Y_org==Y_pre:
                self.correct+=1
        print(f"Total Correct {self.correct}, Total length {len(self.y_pred)}")
        return f"{(self.correct/len(self.y_pred))*100}%"



obj=Logistic_Regression()
obj.sigmoid()
obj.accuracy()
```

```
Total Correct 12, Total length 23
```

Out[24]:

```
'52.17391304347826%'
```

In [ ]:

In [ ]: