GA_f_k

February 9, 2021

[1]:
```
%pylab inline
import numpy as np
import matplotlib.pyplot as plt
from sko.GA import GA
import warnings
import xlrd
warnings.filterwarnings('ignore')
```

```
Populating the interactive namespace from numpy and matplotlib
```

[48]:
```
xl=xlrd.open_workbook(r'C:\Users\29691\Documents\WeChat␣
↪Files\wxid_xnxilu4ixtb211\FileStorage\File\2021-02\Fungi_temperature_curves(1).
↪xlsx')
```

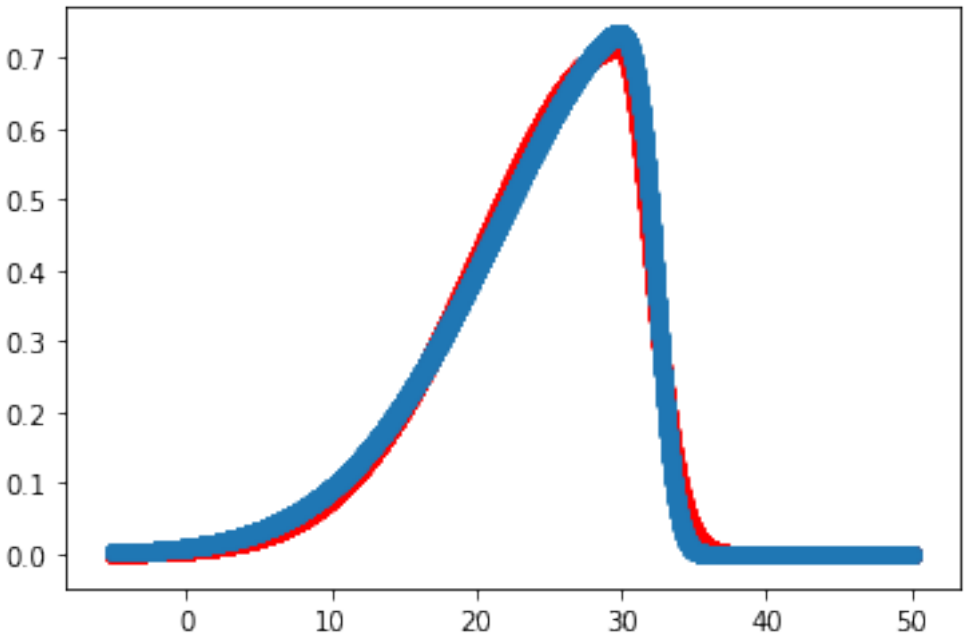
[3]:
```
def f_fun(t,a,b,c,d):
    y=[0.0]*len(t);
    for i in range(len(t)):
        if (t[i]<=d):
            y[i]=a*exp(-1/b*(d-t[i])**2)
        else:
            y[i]=a*exp(-1/c*(t[i]-d)**2)
    return y
```

[4]:
```
def obj_fun(p):
    global x_true
    global y_true
    a,b,c,d=p
    res=np.square(f_fun(x_true,a,b,c,d)-y_true).sum()
    return res
```

[16]:
```
def printf(i):
    table=xl.sheets()[i]
    global x_true
    global y_true
    x_true=np.array(table.col_values(1,1))
    y_true=np.array(table.col_values(2,1))
    plot(x_true,y_true,'o')
```

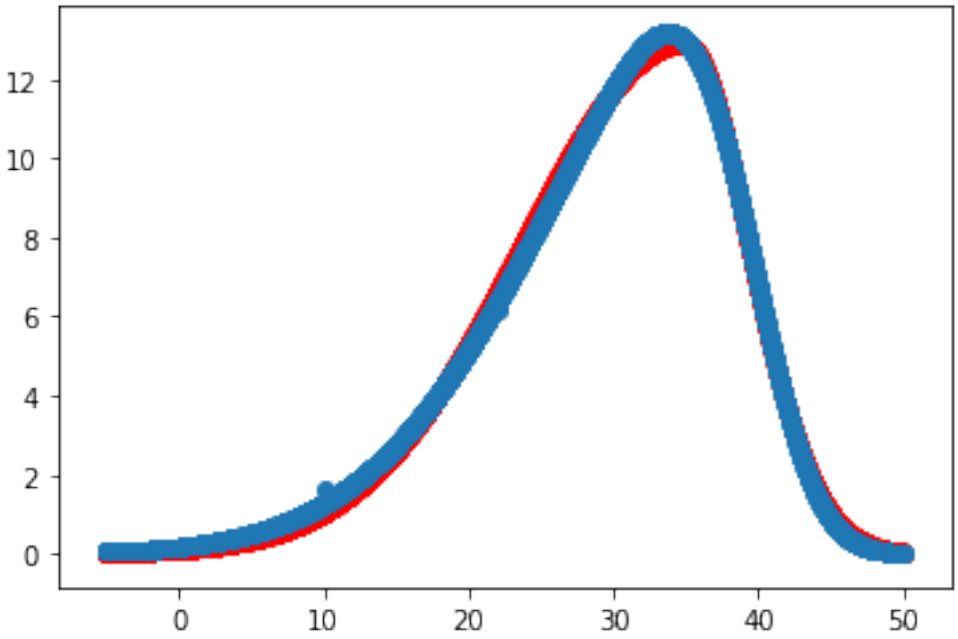
[90]:
```
def GA_f(i,aa,l,u):
    table=xl.sheets()[i]
    global x_true
    global y_true
    x_true=np.array(table.col_values(1,1))
    y_true=np.array(table.col_values(2,1))
    ga = GA(func=obj_fun,n_dim=4,size_pop=50,max_iter=500,lb=[0,50,0,l],␣
 ↪ub=[aa,500,500,u])
    best_params,residuals=ga.run()
    y_predict=f_fun(x_true,*best_params)
    fig,ax=plt.subplots()
    ax.plot(x_true,y_true,'o')
    ax.scatter(x_true,y_predict,c='r')
    plt.show()
    return best_params,residuals
```

[57]:
```
GA_f(1,1,20,30)
```



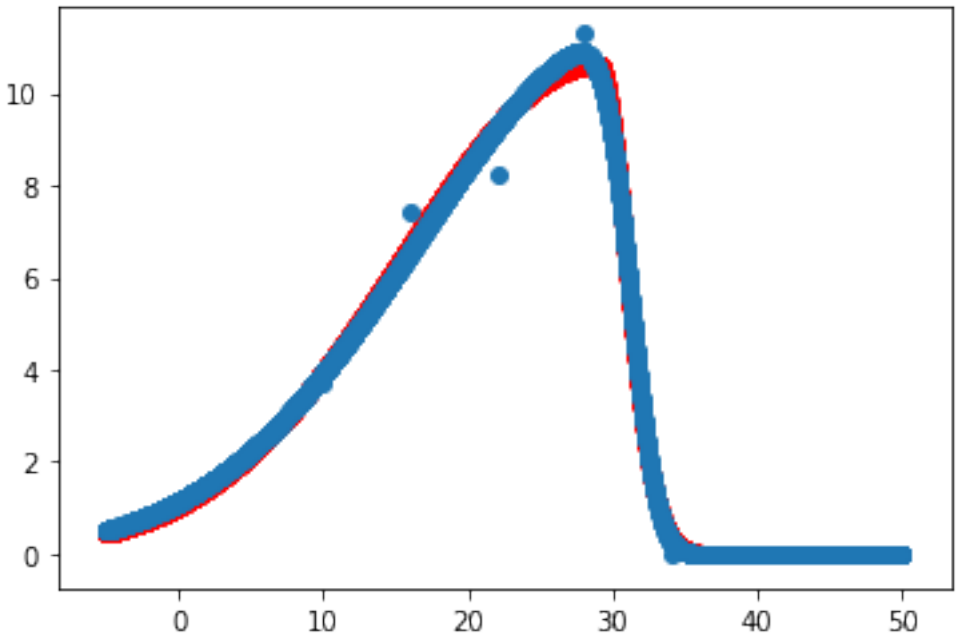
[57]:
```
(array([  0.71571426, 182.12862694,   8.05326589,  29.98048499]),
 array([1.23609018]))
```

[62]:
```
GA_f(2,14,30,38)
```



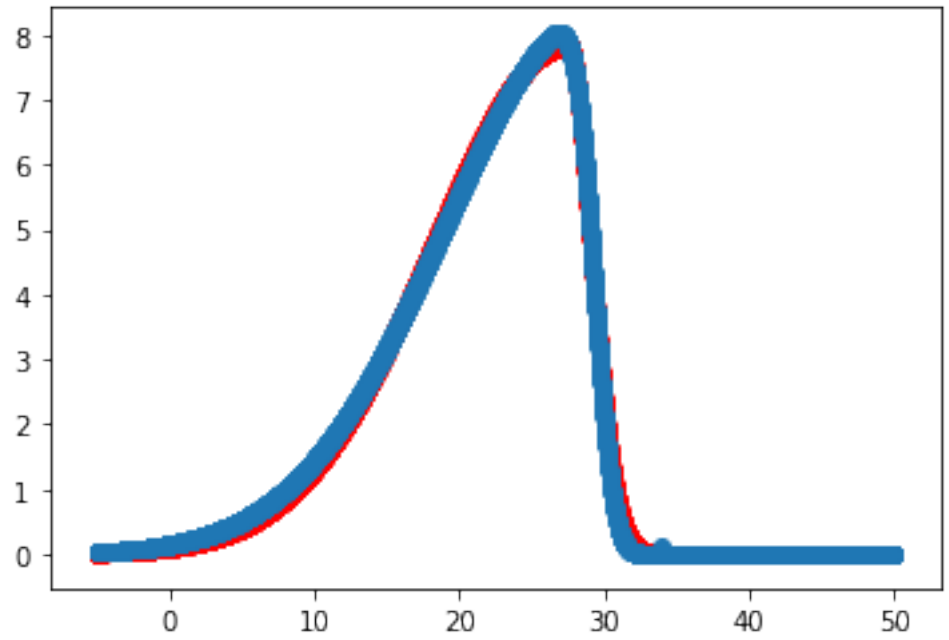
[62]:
```
(array([ 12.9064354 , 246.82643087,  40.5293823 ,  34.86284653]),
 array([157.55832229]))
```

[61]:
```
GA_f(3,12,20,30)
```



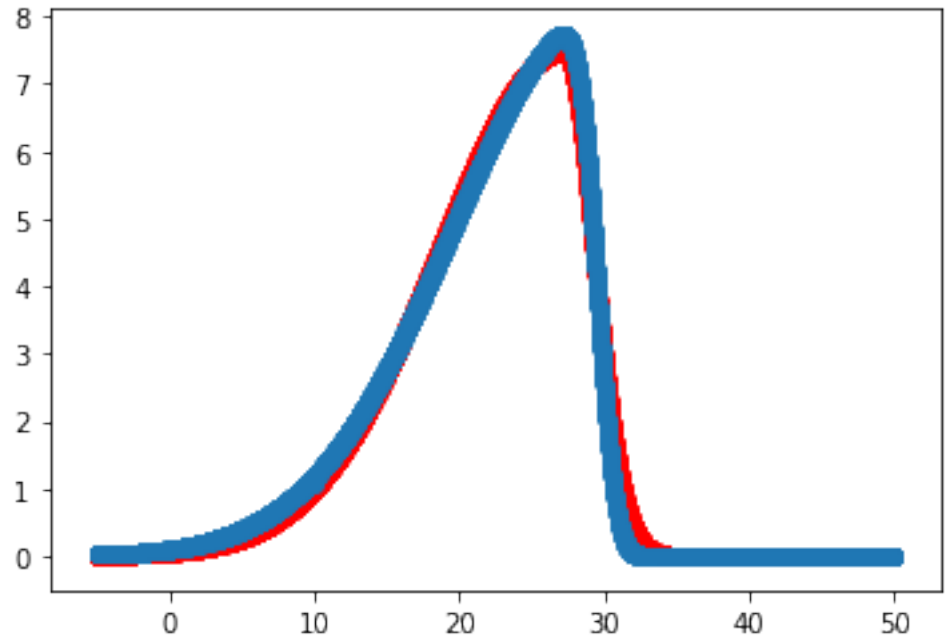
[61]:
```
(array([ 10.63191282, 362.34952665,   7.44863466,  28.96692804]),
 array([64.11856047]))
```

[64]:
```
GA_f(4,9,20,30)
```



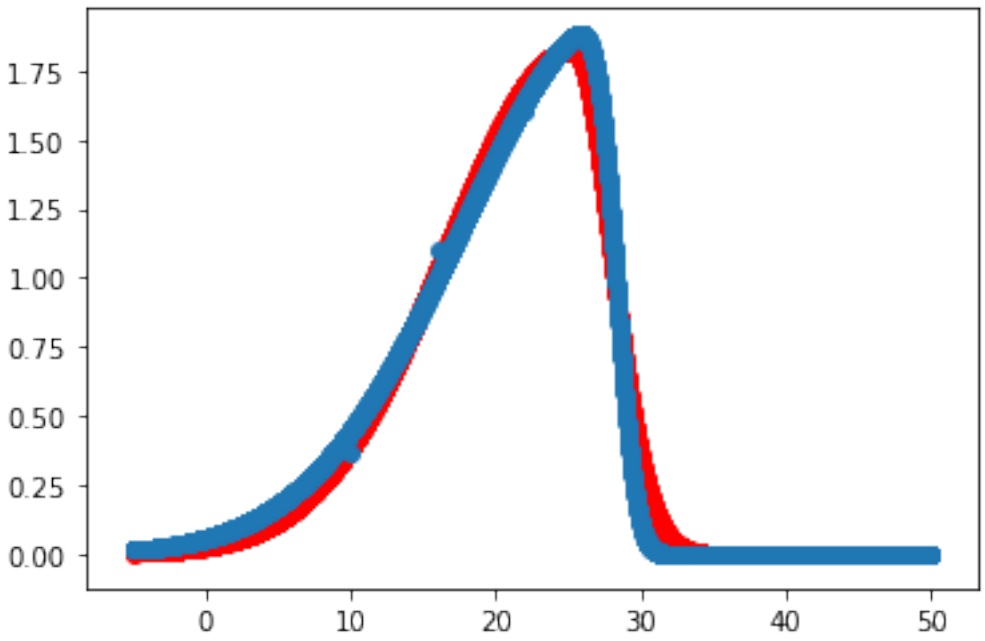
[64]:
```
(array([  7.84422406, 171.35991344,   5.24534856,  27.46413102]),
 array([46.38328671]))
```

[67]:
```
GA_f(5,9,20,30)
```



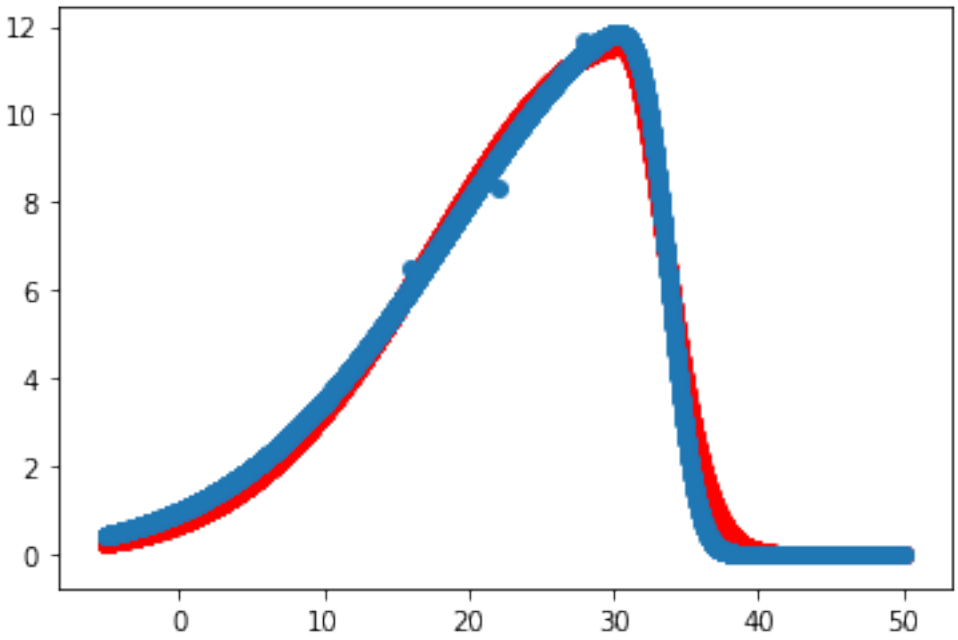
[67]:
```
(array([  7.49470104, 149.13721239,   7.93957413,  27.26698039]),
 array([177.420267]))
```

[69]:
```
GA_f(6,2,20,30)
```



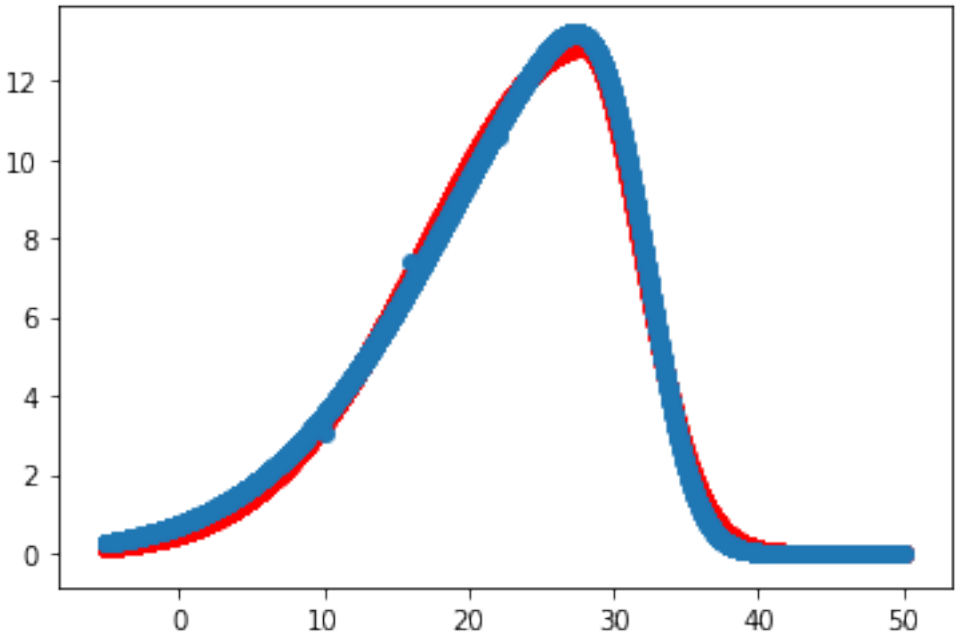
[69]:
```
(array([  1.82684707, 153.31242259,  12.87844725,  25.37952606]),
 array([25.27951812]))
```

[71]:
```
GA_f(7,13,25,35)
```



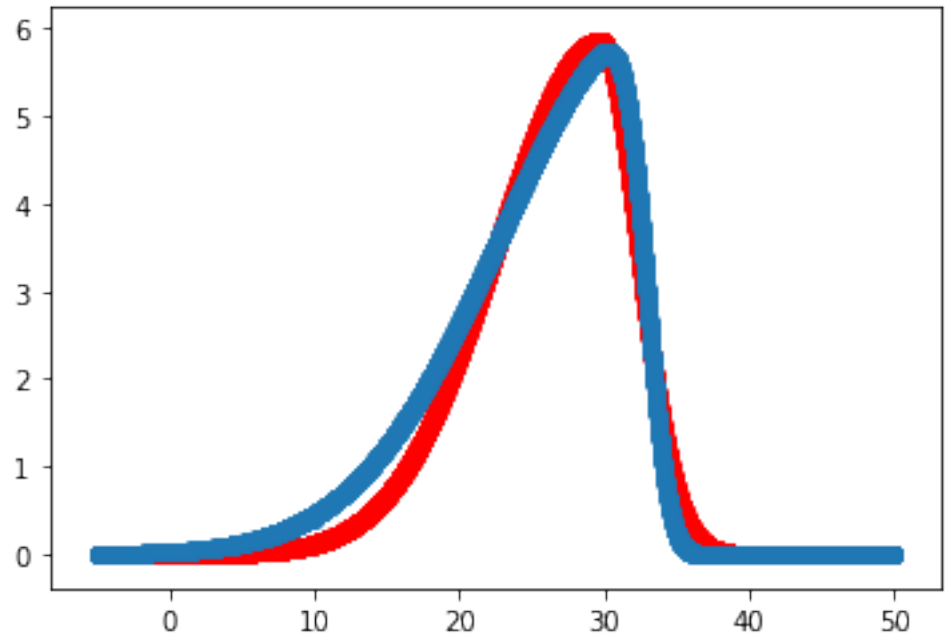
[71]:
```
(array([ 11.54940983, 333.66591196,  17.21532777,  30.61543074]),
 array([607.42716106]))
```

[73]:
```
GA_f(8,14,20,30)
```



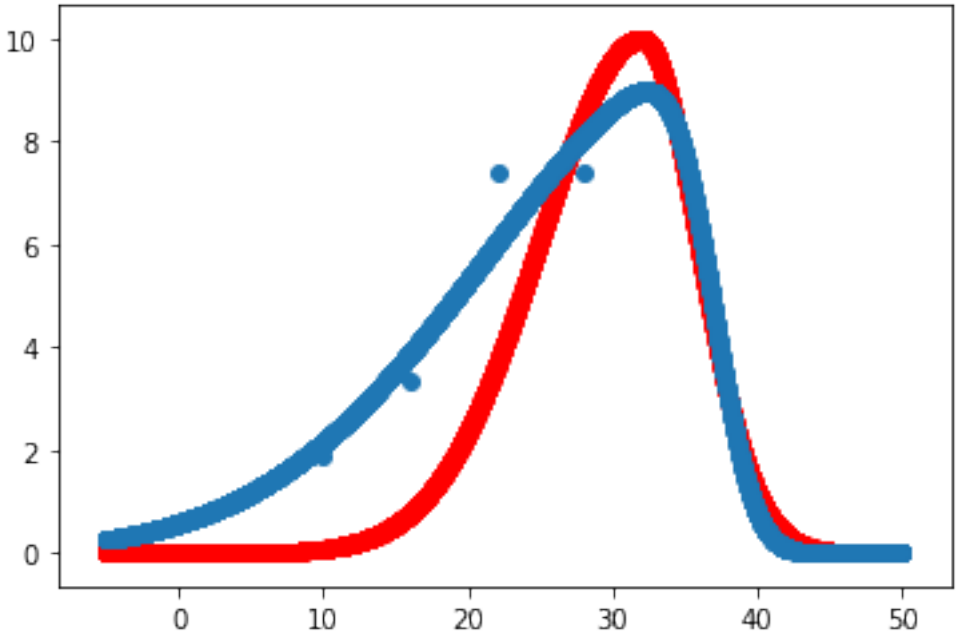
[73]:
```
(array([ 12.86646103, 240.45321616,  29.54091155,  28.03856841]),
 array([287.59017307]))
```

[77]:
```
GA_g(9,6,25,35)
```



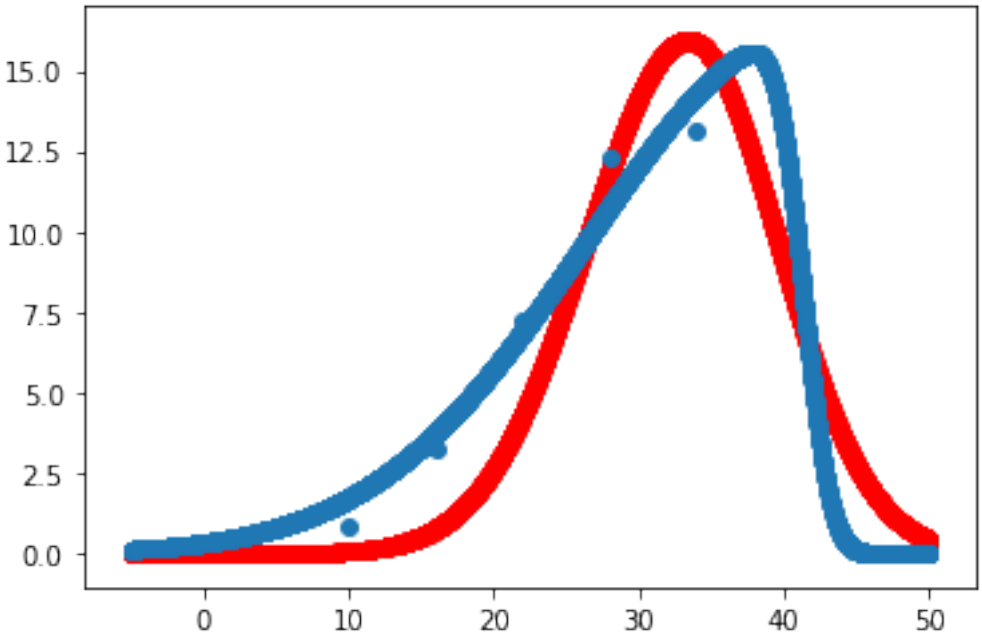
[77]:
```
(array([ 5.85330024, 99.99999963, 13.4759226 , 29.70687087]),
 array([466.19703601]))
```

[91]:
```
GA_g(10,10,32,38)
```



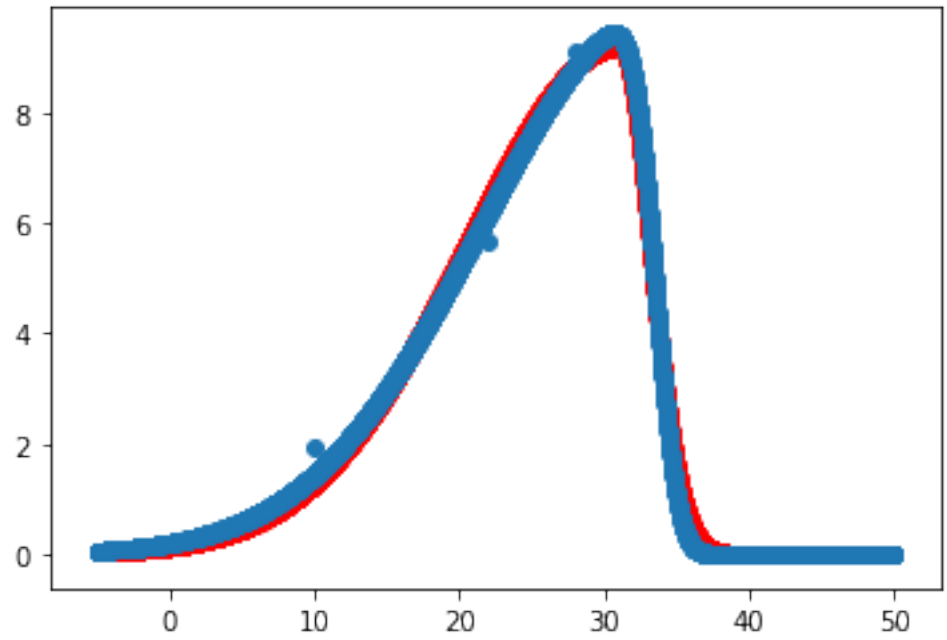
[91]:
```
(array([10.        , 99.99999395, 29.62449894, 32.        ]),
 array([12891.11573093]))
```

[82]:
```
GA_g(11,16,30,40)
```



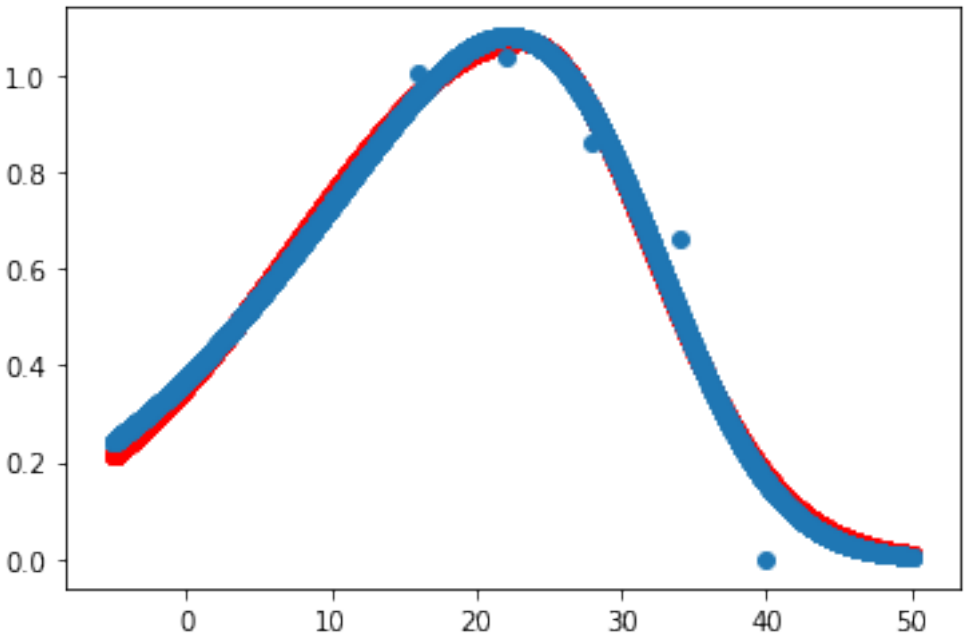
[82]:
```
(array([ 15.95139164, 100.        ,  76.03595832,  33.42404547]),
 array([22662.46231109]))
```

[86]:
```
GA_f(12,10,25,35)
```



[86]:
```
(array([  9.21875759, 227.88289466,   8.33661994,  31.13574651]),
 array([123.6505132]))
```

[87]:
```
GA_f(13,1.2,20,30)
```



[87]:
```
(array([  1.06962232, 496.17292552, 152.58942389,  23.25600828]),
 array([0.90301934]))
```