Genetics and Evolution
Assignment Two

Hadis Ahmadian
400211524

Libraries :

first we need to import the libraries that we need:

matplotlib for drawing graphs.

In [1]:
```
import matplotlib.pyplot as plt
```

Drwing the logistic map :

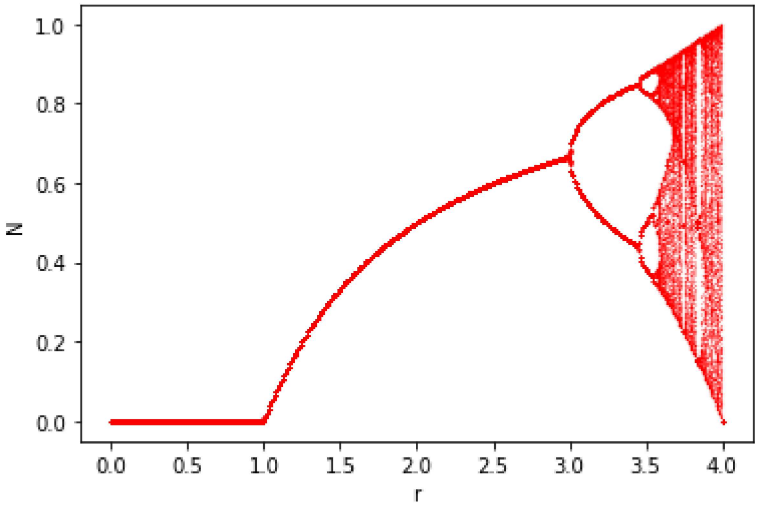
the below code calculates the value that N converges to in 500 generations for all 0<r<4 and N=0.5

the logistic map shows the value wich N converges to, per each r. we can see that for r>=3 N does not converge to a certain value and here is were chaos stars. the chaos will get more as the value of r gets closer to 4.

I will explain more detail about the value of convergence for each range of r next parts.

below you can see the logistic map.

In [2]:
```
n=0.5

x=[]
y=[]

for j in range (401):
    r=j/100
    n=0.5
    for i in range(500):
        n=r*(n-(n*n))
        if i>100:
            x.append(r)
            y.append(n)


plt.scatter(x, y,s=0.01,color="red")

plt.xlabel('r')
plt.ylabel('N')

plt.show()
```



pop_itter_plot function :

this function recives N and r and draws the graph of N per generation for 60 generation, we'll use it to see what does N converge to for diffrent amounts for r. the function calculates N of each generation in a loop (using n=r*(n-(n^2))) and saves N of each generation and draws the graph.

In [3]:

```
def pop_itter_plot(n,r):

    x=[]
    y=[]

    for i in range(60):
        x.append(i+1)
        y.append(n)
        n=r*(n-(n*n))
    plt.plot(x, y,color="blue")
    plt.xlabel('itteration')
    plt.ylabel('population')
    return plt
```

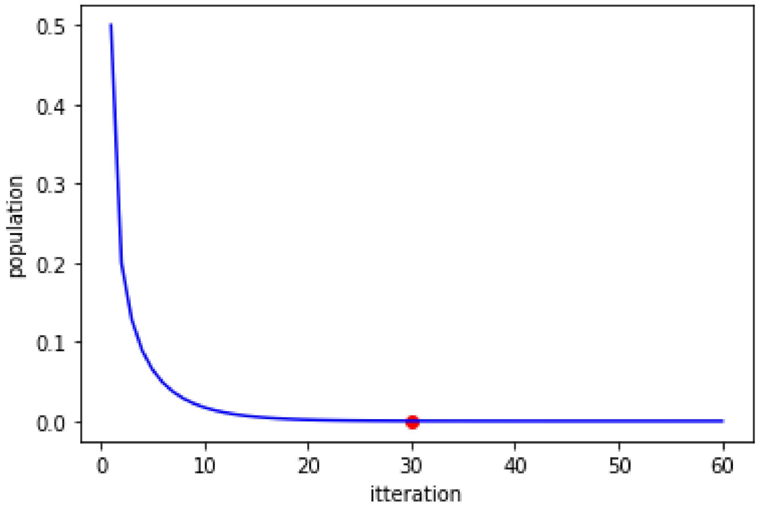
we're going to test above function with N=0.5 for 1st generation and diffrent amounts of r.

1) 0<r<1

we can see that for r=0.8 (and all r if 0<r<1) the N will converge to 0.

In [4]:

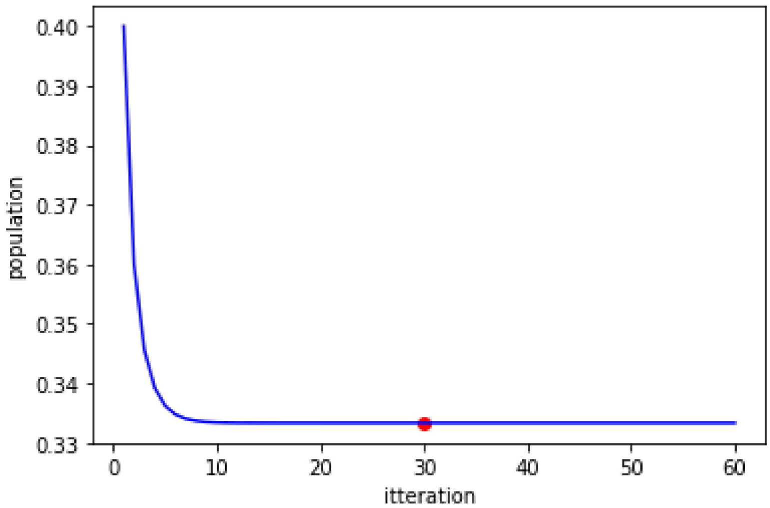
```
n=0.5
r=0.8
p=pop_itter_plot(n,r)
p.scatter([30],[0],color="red")
p.show()
```



2) 1<r<2

we can see that for r=1.5 (and all r if 1<r<2) the N will converge to a number. it is proven that the number is (r-1)/r

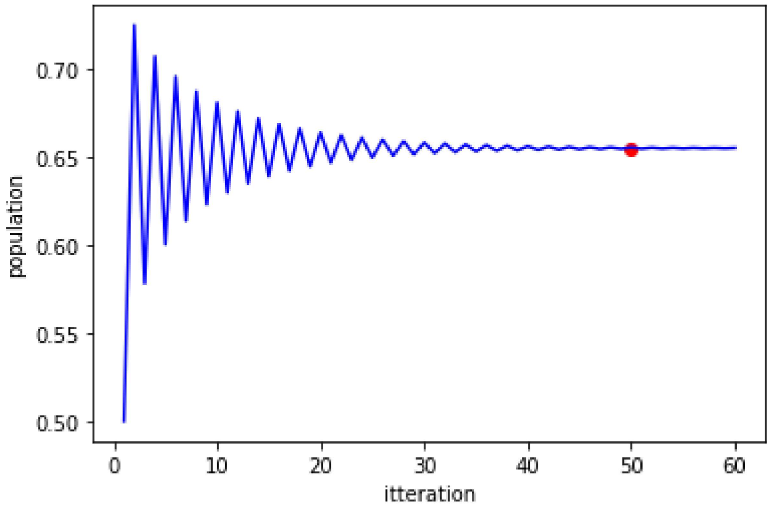
In [5]:
```
n=0.4
r=1.5
p=pop_itter_plot(n,r)
p.scatter([30],[(r-1)/r],color="red")
p.show()
```



3) 2<r<3

we can see that for r=2.9 (and all r if 2<r<3) the N will converge to a number, however not in the same shape as the last part, it increases and decreases for a while until it finally converges. it is proven that the number witch it converges to is (r-1)/r

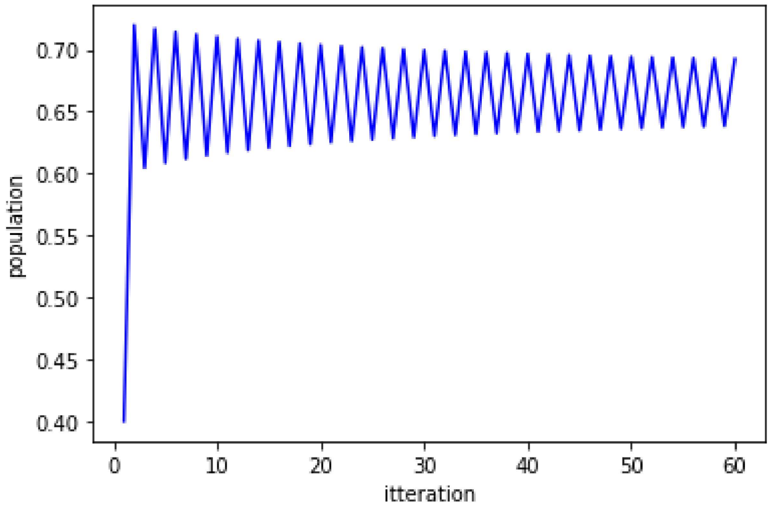
In [6]:
```
n=0.5
r=2.9
p=pop_itter_plot(n,r)
p.scatter([50],[(r-1)/r],color="red")
p.show()
```



4) r>=3

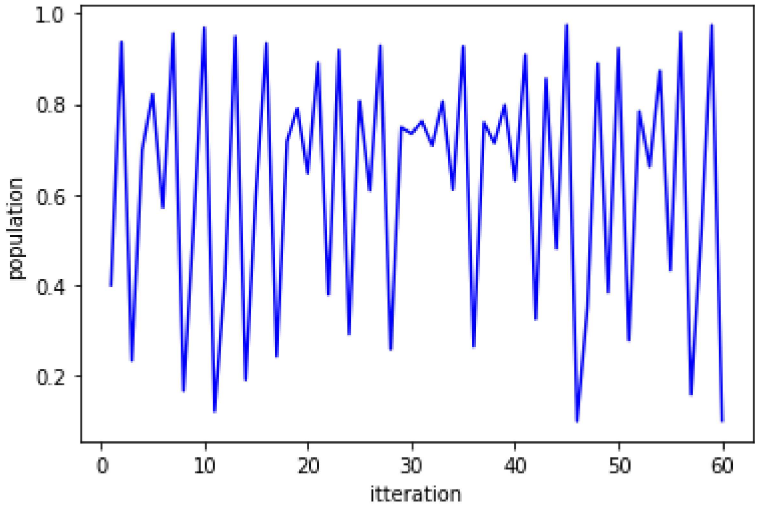
for r>=3 we can see that N will not converge to a certain value over generations. so **r=3** is the **threshhold**. and it's when the chaos begins.

In [7]:
```
n=0.4
r=3
pop_itter_plot(n,r).show()
```



it is also important to say that the higher the r (in range 3 to 4) the more fluctuation N will have (its value will be less stable over generations, for bigger r values)

In [8]:
```
n=0.4
r=3.9
pop_itter_plot(n,r).show()
```


1.0
0.8
0.6
0.4
0.2
population
0
10
20
30
40
50
60
itteration