

## MAS212 Population dynamics

1. Attempts to model population dynamics using mathematical methods go back to the 18th century, and the writings of Robert Malthus. Malthus argued that, at any point in time, the rate of population growth is proportional to the population itself. This leads to exponential growth.

Malthus idea is equivalent to the differential equation  $\dot{x} = ax$ , where  $x$  is the population,  $\dot{x} \equiv dx/dt$  is its rate of change with respect to time  $t$ , and  $a$  is a constant parameter. The general solution is  $x(t) = x_0 e^{at}$ , where  $x_0$  is the initial population. Malthus' writings influenced the work of both the naturalist Charles Darwin and the economist Adam Smith.

– **Q.** Why is this model unrealistic?

2. In 1838, Verhulst refined Malthus' model to include the effect of resource limitations, by introducing the **logistic equation**

$$\dot{x} = ax(1 - bx),$$

where  $a$  and  $b$  are positive parameters. This differential equation has a general solution  $x(t) = (b + Ae^{-at})^{-1}$ , where  $A$  is any positive constant.

– **Q.** What happens to the population as  $t \rightarrow \infty$ ?

3. Now suppose a population varies annually according to the **logistic map**

$$x_{n+1} = rx_n(1 - x_n),$$

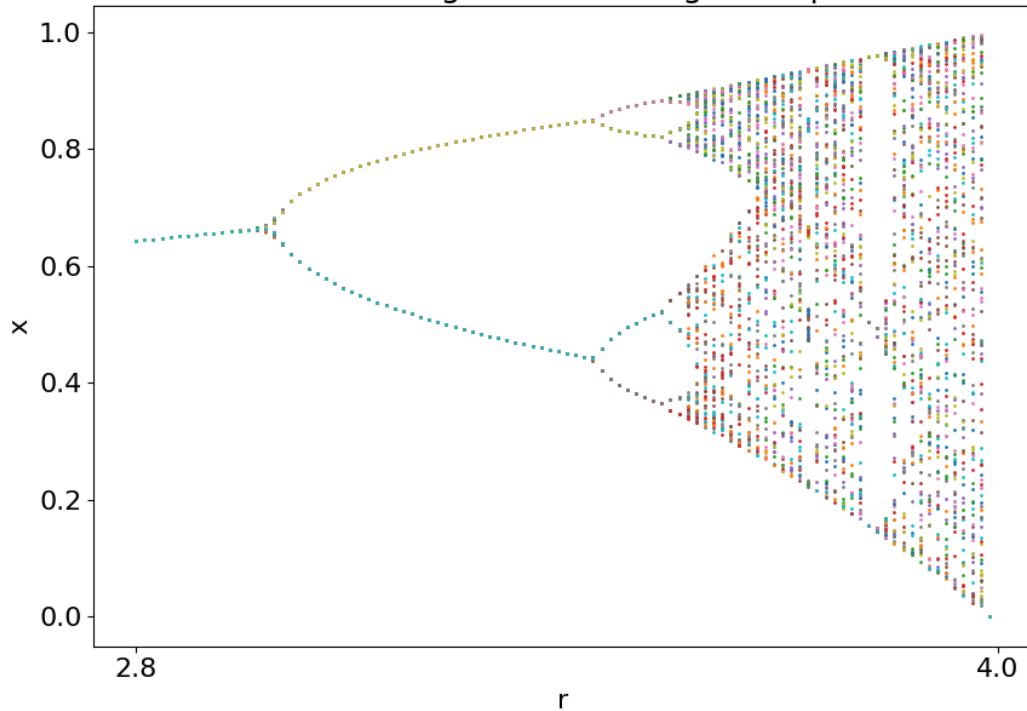
where  $x_n$  is the population in year  $n$ , and  $r$  is a positive real parameter. (The logistic map is a discrete version of the logistic equation.)

– **Q.** What happens to  $x_n$  as  $n \rightarrow \infty$ ? Are you sure?

4. Write a function `logisitic(r, n=100)` that returns an array of values  $x_0, x_1, \dots, x_{n-1}$  for the parameter  $r$ .

5. Plot the values  $[x_{n_1}, x_{n_1+1}, \dots, x_{n_2}]$  as pixels (with e.g.  $n_1 = 100, n_2 = 200$ ), for parameters  $r$  in the range  $[2.8, 4.0]$ .

Bifurcation diagram for the logistic equation



```
import numpy as np
import matplotlib.pyplot as plt
N = 100
rmin, rmax = 2.8, 4.0
rs = np.linspace(rmin,rmax,N)
npts = 100
nmin = 100
data = np.zeros((N,npts))
for i in range(N):
    x = 0.5
    for k in range(nmin+npts):
        x = rs[i]*x*(1-x)
        if k >= nmin:
            data[i, k-nmin] = x

plt.rcParams.update({'font.size': 16}) # larger font
plt.title("Bifurcation diagram for the logistic equation")
plt.gcf().set_size_inches(10,7) # make bigger
plt.xlabel("r"); plt.ylabel("x")
plt.xticks([0,N], [rmin,rmax])
plt.plot(data, 'o', markersize=1)
plt.savefig("logistic_bifurcation.png", dpi=100)
```