Remember that for the *continuous* time logistic equation we found that a small deviation \( n \) from an equilibrium \( \hat{N} \) is given by:

\[
n(t) = n(0)e^{\lambda t}
\]

Where \( n(0) \) is the initial condition, \( \lambda = F'(\hat{N}) \), and \( F'(\hat{N}) \) is the slope of the production function \( F(N) \) at \( \hat{N} \). Now the small deviation \( n \) shrinks through time if \( \lambda < 0 \) giving a stable equilibrium, and \( n \) grows through time if \( \lambda > 0 \) giving an unstable equilibrium.

Is this the same for the *discrete* time logistic? Let’s derive what we need and find out. Although we have been writing the discrete time model as \( \Delta N = F(N) \), it is more customary to write it as:

\[
N_{t+1} = N_t + F(N)
= G(N_t)
\]

So an equilibrium is the root of \( N = G(N) \), remembering that \( N_{t+1} = N_t \) at equilibrium. So let’s see if an equilibrium is stable using the same approach we used for the continuous time case. A small deviation from an equilibrium is given by \( n = N - \hat{N} \) so:

\[
N_{t+1} = G(N_t)
\]

\[
n_{t+1} + \hat{N} = G(n_t + \hat{N})
\]

Now again we take advantage of the fact that we are considering small deviations from the equilibrium, so we use a Taylor series to the first order to expand \( G \).

\[
n_{t+1} + \hat{N} = G(\hat{N}) + G'(\hat{N})n_t
\]

Now \( G(\hat{N}) = \hat{N} \) so

\[
n_{t+1} = G'(\hat{N})n_t
\]

To simplify, let \( \lambda = G'(\hat{N}) \), which gives:

\[
n_{t+1} = \lambda n_t
\]
You’ll recognise this as the discrete time model for geometric growth. So, if the initial deviation from equilibrium is $n_0$, the deviation at time $t$ is:

$$n_t = n_0 \lambda^t$$

The test for stability is now clear. If $\lambda < 1$, any small deviation from equilibrium shrinks through time and the equilibrium is stable. If $\lambda > 1$, any small deviation from equilibrium grows through time and the equilibrium is unstable.

Now let’s apply these criteria to our discrete time logistic.

$$G(N_{t+1}) = N_t + rN_t(1 - \frac{N_t}{K})$$

$$G'(N_{t+1}) = 1 + r - \frac{2rN_t}{K}$$

Now we know that the two equilibrium points for the logistic are 0 and $K$, so that:

$$G'(0) = 1 + r$$

$$G'(K) = 1 + r - 2r = 1 - r$$

Thus if $r$ is positive, $N=0$ is always unstable because $\lambda = 1 + r > 1$. The equilibrium at $N=K$ is another matter. If $r$ is between 0 and 2 then $|1 - r|$ is less than 1 and $N=K$ is stable (i.e. $\lambda'$ will become smaller). However, if $r$ is too big (>2) then $N=K$ is unstable and the population never settles down to a steady state but keeps fluctuating instead.

Q1. Use the function logist_d that we covered in class to calculate values for the discrete logistic curve that can be run with various values of $r$, $K$, $n_0$ and the number of time steps (runlen). Run this function with $K=1000$, $n_0=2$, 30 time steps and $r$ values of 0.5, 1.5, 2.5, and 3.5. Plot on the same graph, population size vs time for each value of $r$. Explain what happens to the population with different values of $r$?

Now let’s do a more comprehensive assessment of what happens to population size for different values of $r$. We are going to produce a plot of population size vs $r$. The following commands will produce the graph.
Q2. Explain what happens to a population as \( r \) increases.

We can see that as \( r \) is tuned higher, the number of points in the oscillation increases. Above a value of 3 something special happens – the population is said to be “chaotic”.

Chaos in mathematical models is all the rage – it has even been mentioned in the movie “Jurassic Park”. Here’s what it is all about. When \( r=3 \), the population doesn’t generally settle down to any one steady state or to a particular oscillation between some fixed number of states. Instead, it wanders around the interval between 0 and \((4/3)K\), and comes arbitrarily close to any point within that interval. Furthermore, if two populations start out near to one another, they don’t stay together, but deviate from one another so completely that after a while one cannot tell that they were ever near one another to begin with.

Q3. Let’s show this by example. Assume that two populations with \( r=3 \) and \( K=100 \) are started close to each other, one at say \( N=101 \) and the other at \( N=99 \). Run each for 50 iterations and plot together on a graph. What is happening?

Because two populations that start near one another completely lose their closeness through time, in practice one cannot predict the long-term fate of a population if its trajectory is chaotic. This is because one cannot in practice ever know the initial conditions exactly, and any error in the initial condition is magnified through time, so that one cannot predict what the population’s size will be far off into the future. This inability to predict the future is true even though the model has no randomness. But don’t despair. It’s not hard at all to predict the fate of a chaotic population in a statistical sense, nor is it hard to falsify the claim that a real population is exhibiting the dynamics of discrete-time logistic chaos. Although two chaotic populations that are initially close to each other do lose their closeness, they also both converge to the same statistical distribution of population sizes through time.

Q4. Make a histogram of the population sizes through time for both of the chaotic populations we just computed. What do you notice?

How do we use what we have learnt to see if a real population exhibits chaos? First of all we can check whether \( r \) is near 3, and second we can check whether the statistical distribution of population sizes conforms to the U-shape seen in Q4. Let’s do this for the following data on the abundance of the titmouse, a bird species from Buckeye Lake, Ohio (USA).
Let’s see if this population exhibits chaotic behaviour. First we need to estimate the parameters of the discrete logistic equation for the two bird species by fitting the production function to the population size using non-linear curve fitting:

\[ n = \text{Titmouse}; \]
\[ \text{delta}_n = \text{diff}(n); \] % Gives a vector of delta n
\[ n = n(1: \text{length} (\text{delta}_n)); \] % Discards last element of n vector, so same length as delta_n
\[ \text{modelFun} = @(p, n) p(1) * n .* (1 - n ./ p(2)); \] % Equation of production function versus population size
\[ \text{Initial} = [1; 100]; \] % Initial estimates of r and K
\[ \text{paramEsts} = \text{nlinfit}(n, \text{delta}_n, \text{modelFun}, \text{Initial}); \] % Estimating r and K

Q5. What are the parameter estimates for \( r \) and \( K \) for the Titmouse?

Q6. What does the statistical distribution of population size look like for the Titmouse. Is it exhibiting chaotic fluctuations?