Simple population model

The birds do it, the bees do it,
let’s model it and run it past Matlab …
Our agenda

- Build a simple model
- Put model in computable form
- Create a simulation model
- Add additional features to the model
A simple model

- Consider a population in a given region.
- Assume no in- or out-migration.
- Consider a period of time (e.g., year)
- Why might the population increase?
- Why might it decrease?
Model factors defined

- Consider a population in a given region: $P$
- If $P$ represents a human population, what time scale is reasonable?
- Track the model behavior by $P(t)$.
- Why might the population increase?
- Why might it decrease?
Model factors defined (2)

Population increase effect:
\[ \frac{dP(t)}{dt} = BR \times P(t) \]

Population decrease effect:
\[ \frac{dP(t)}{dt} = -DR \times P(t) \]
Simple population model

Dependent vbl: \( P \), number of people

Independent vbl: \( t \), time (in years)

\[
dP(t)/dt = BR \cdot P(t) - DR \cdot P(t)
\]

Model parameters:
- \( BR \), birth rate (births per person per year)
- \( DR \), death rate (deaths per person per year)
dP(t)/dt = BR*P(t) – DR*P(t)

What do you predict for P(t) if …
(1) BR > 0 and DR = 0?
(2) DR > 0 and BR = 0?
(3) BR = DR?
(4) BR > DR?
(5) BR < DR?
Build a simulation model

What do we know at time 0 (start time)?

\[ \frac{dP(t)}{dt} = BR*P(t) - DR*P(t) \]

becomes

\[ \frac{dP(0)}{dt} = BR*P(0) - DR*P(0) \]

In other words, we know the slope of \( P \), provided we know \( P(0) \).
Express this in a computable form

Approximate the derivative term ‘$dP(t)/dt$’ by

$$\frac{\Delta P}{\Delta t} \rightarrow \frac{P(0+\Delta t) - P(0)}{\Delta t}$$
Steps in computation

Use ‘delt’ to replace $\Delta t$, for convenience.

The first step in time looks like this:

$$\frac{P(\text{delt}) - P(0)}{\text{delt}} = BR*P(0) - DR*P(0)$$
Steps in computation

Time step 1:

\[
P(delt) - P(0) = \frac{BR \cdot P(0) - DR \cdot P(0)}{delt}
\]

Time step 2:

\[
P(2\cdot delt) - P(delt) = \frac{BR \cdot P(delt) - DR \cdot P(delt)}{delt}
\]
How to save the results

**Time variable:**
The time values are simply

\[ 0, \text{delt}, 2\times\text{delt}, 3\times\text{delt}, \ldots \]

Save them in an array

\[ t = [ 0, \text{delt}, 2\times\text{delt}, 3\times\text{delt}, \ldots ] \]

What is \( t(1) \)? \( t(3) \)? \( t(i) \)?
Population variable:

Represent $P(t)$ as a set of values at the time step points $t_i$:

$$P = [P(1), P(2), P(3), \ldots]$$
A more convenient computing form

Convert this ~

\[
P(i+1) - P(i) = \frac{BR \cdot P(i) - DR \cdot P(i)}{\text{delt}}
\]

to this ~

\[
P(i+1) = P(i) + \text{delt} \cdot (BR \cdot P(i) - DR \cdot P(i))
\]
Standard simulation form

This form is commonly used for simulation:

\[ P(i+1) = P(i) + \delta t( BR*P(i) - DR*P(i) ) \]

It is usually referred to as the Euler form of a model.
Additional info for computation

Set initial time: \( t_{init} = 0 \)

Set final time, \( t_{final} \) = ? yrs

Choose nbr of time steps: \( n_{steps} \) = ?

Set initial population: \( P_{init} = 20000 \)
It’s Matlab time!

Key concept:
Organize the computations in a loop and step the time trajectory of \( P(t) \) out.

See
popdyn_1_demo … script file
popdyn_1_fcn_demo … function file
How to show dominance

Four subparts, in order of complexity:

- `popdyn_1`: *no immigration effect*
- `popdyn_2`: *constant immigration effect*
- `popdyn_3`: *delayed immigration effect*
- `popdyn_4`: *delayed cyclic immigration effect*
Evaluation scheme

Submit all the parts that run.

(1) Save a valid file 1 pts total
(2) File runs 2 pts total
(3) popdyn_1 correct 4 pts total
(4) popdyn_2 “ 6 pts total
(5) popdyn_3 “ 8 pts total
(6) popdyn_4 “ 10 pts total

We will evaluate the highest part we find.