Bootcamp modeling exercises

In class we saw how to simulate discrete-time models (such as the discrete logistic growth model) in R, using a for loop. Now we will work with an alternative model for density-dependent population growth in discrete time, the Ricker model:

$$n_{t+1} = n_t \exp\left[r\left(1 - \frac{n_t}{K}\right)\right]$$

where $K > 0$ and r can take positive or negative values.

(a) Write a function that runs the Ricker model, plots the result, and returns the time series as an output. At minimum, your function should take all parameter values and initial conditions as input arguments.

(b) Explore the dynamics of the model. Try to find combinations of parameter values that yield the following patterns:

- Population decreases to $n = 0$.
- Population approaches stable equilibrium at $n^* = K$, without oscillations.
- Decaying oscillations around $n^* = K$.
- Persistent, regular oscillations.
- Crazy, random-looking fluctuations (chaos).

Which parameter is the key driver of these patterns?

(c) Choose six interesting values of this parameter. Write a script that makes an array of six plots showing the model dynamics for each of these values.

(d) Imagine $n_0 = 20$ and $K = 1000$ for a certain population of deer that is growing according to the Ricker model. You are a wildlife manager, and are concerned about how long it will take for the population to reach half of its carrying capacity. That is, you want to know $t_{K/2}$, the first year that $n_t \geq K/2$. Suppose your output time series is called `nVec`. Write an R command that will determine the index of the first element of `nVec` that is $\geq K/2$.

(e) Write a script that runs the necessary simulations and collects the necessary data to plot how $t_{K/2}$ depends on r, for the range of r from 0.1 to 0.9.

(f) Write pseudo-code to do a joint sensitivity analysis for two parameters. That is, choose a vector of values you'd like to consider for both r and K, and choose a simple output from your model (e.g. population size at $t = 10$). Run the model for all possible combinaioins of these values and collect the results in a matrix with appropriate dimensions. Plot the results in some way. Hint: the straight-forward way to do this uses a 'nested loop'.

(g) Convert your pseudo-code into an R script to do a two-dimensional sensitivity analysis, and create a visually appealing plot to summarize the results. (You will need to google for plotting commands, e.g. contour or surface plots).

(h) (BONUS, OPTIONAL) Write a script to make a bifurcation plot for this model, showing how its long-term dynamics change as the parameter r is varied. You will need to collect a set of values reflecting the long-term dynamics of N for each value of r, where r falls between 0 and 4. Plot these N-values as points on the y-axis, versus the corresponding value of r on the x-axis. Hint: you may need to look up the R command `matplot`. If you get done with this and still have hunger for more, do it again using the `apply` function.