Problem Set 3a

Observation Noise, Process Error, and Parameter Estimation

When we look at real-world survey data, we rarely see the smooth lines that we might expect from constant growth rates; instead, we generally see quite a bit of year to year variance. As we will discuss in class and in the readings, we must take different approaches to analyzing population time-series depending on whether we believe most of the variance in the recorded time series represents observer error or process error. To investigate this further, you will be generating some simulated data sets (for which you know the underlying population parameters) and seeing how well you can do at estimating these parameters under each scenario (i.e. either with process error assumed to be much greater than observer error or with observer error assumed to be much greater than process error). This exercise will require you to use most of the programming skills you have developed in the preceding weeks. (There's some hints and suggestions on the next page too.)

Part A

Simulate discrete density-independent growth over a 100-year period for a population that reproduces annually, has a starting population size of 10, and where $\lambda = 1.01$. Conduct 50 replications of two types of simulations:

- Growth with process error but no observation error, such that temporal variation in λ is log-normal with $\sigma_p = 0.1$. [HINT: This should sound familiar; it's the same as the last problem set.]
- Growth with no process error, but where our annual estimates of N include sampling error. That is, assume that variance in the estimated values of N (which we'll refer to as N_{est}) is also log-normal or, put another way, that $\log(N_{est})$ is normally distributed, with standard deviation $\sigma_o = 0.1$.

Part B

For both sets of simulations, make a plot of log population size (or log estimated population size) versus time, with each replication shown as a separate line. Overlay these with a line showing the expected population growth with the specified "true" value of λ. How do the two lines differ? Make an additional plot of $\log(N_t + 1)/N_t$ versus N_t for the process error simulations and overlay a line showing the expected growth rate.

Part C

Use the methods we discussed in class (and in the readings) to estimate λ and σ_p (for the first set of simulations performed above), and $N(0)$ and σ_o (for the second set of simulations) from their respective simulated time-series data. How close are your estimates to the actual values you used to generate the data? Do you think we would we have done this well if we had used the inappropriate methods (i.e. assumed observer error with the process error data set and vice versa)?

Hints, equations and commands you might find useful

- With process error: $N_{t+1} = \lambda e^{\epsilon}_{t} N_t$, where ϵ_t is a normally-distributed random variable with mean 0, and standard deviation σ_p.
- With observer error: $N_{t+1} = \lambda N_t$ and $\log(N_{t,est}) = \log(N_t) + \epsilon_t$, where ϵ_t is a normally-distributed random variable with mean 0, and standard deviation σ_o.
- Remember to include observer error in your estimated population size at $t = 0$.
- The function for creating a normally-distributed random variable is `rnorm()`.
- Remember to use the appropriate method for estimating variance depending on whether it is observer error (use trajectory matching) or process error (use growth rate per time step fitting).
- You may want to define some functions of your own to conduct your calculations, just like in the last problem set. You create a function like this:

  ```
  f2 < -function(x){ log(N0) + log(lambda) * x}
  ```

  [Note: It is conventional to use the symbol x as the independent variable.]
- Having created a function, you can call it up anywhere in the script by using it as an "argument" in other functions. For example, to plot your new function over the interval `1:T` you can use `curve(f1, 1, T)`.
- You may have noticed that function `f1` requires two parameters: `N0` and `lambda`. What if we did not know $N(0)$ and λ, but instead wanted to estimate them by fitting our function to a data set of observed x (= time) and y (= $\log(N_{est})$)? We would have to create a new function:

  ```
  f2 <- function(b, x){ log( b[1] ) + log( b[2] ) * x }
  ```
- Function `f2` can be called as an argument by an appropriate fitting function to estimate the two values of `b`, where `b[1]` represents $N(0)$ and `b[2]` represents λ. Code to fit a function using least squares regression is:

  ```
  fit <- nls(as.vector(y) ~ log(b1) + log(b2) * as.vector(x),
                              start = list(b1 = 1, b2 = 1))
  b <- coef(fit)
  ```

  Note that we specify the function we are trying to fit as an argument within the `nls` function and provide some starting values of `b` for the search routine. The second line extracts the two parameters from the fitted model.