Program Structure, Good Practices and First Program

AP Environmental Studies Lesson Summary

Nicholas Trunfio
NSF GK-12 Vibes and Waves in Action
AP Environmental Science

Summary of lesson:
This lesson is intended to provide the students with the good practices that are associated with programming. Emphasis is placed on making the code so readable that someone without a good deal of programming expertise can understand the code. Good program structure is covered, as well as good naming conventions. Proper places for comments are also discussed. Finally, the students are able to blend the coding techniques they've learned in the past with the practices learned today to create their first program that is geared towards understanding environmental science.

Program Structure, Good Practices and First Program

AP Environmental Science Lesson Plan

Objectives: Obtain and implement good programming practices in R.

Frameworks: Math: N-Q, A-all, A-SSE, A-APR, A-CED, A-REI, F-IF, F-LE. Inquiry: SIS1, SIS2, SIS3, SIS4

L-Side Activities: Teacher

At the bell: Write a short code segment that stores the value of 5 in x, the value of 12 in y and the value of x + y/x in z.

Agenda:

1. Program Structure
    - Program as an essay
2. Comments
    - Syntax
    - Where they're needed
3. Naming conventions

<

R-Side Notes: Students

Outline:

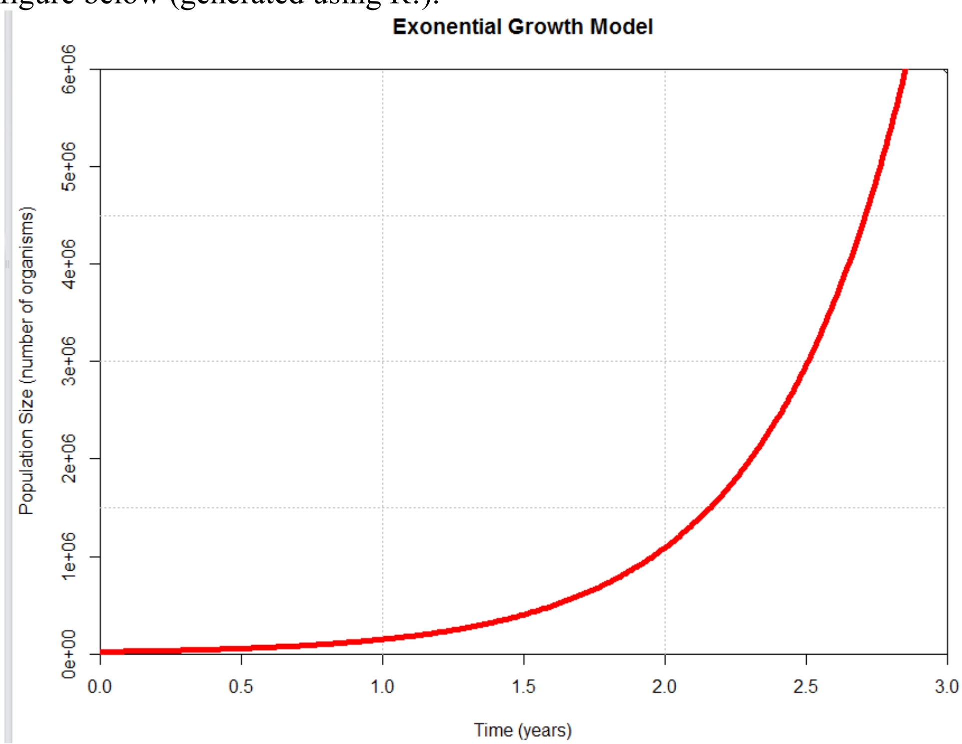
1. Program Structure
    a. As an Essay
        - Intro
            a. Prefaces Argument
            b. Provides background
        - Body
            a. Argument
            b. Your own Analysis
        - Outro
            a. Conclusions
    b. As a Program
        - Intro
            a. Define known variables
            b. Set range and scale of independent variables
        - Body
            a. Math operations to find needed parameters / unit conversions
            b. Simulation to find variables of interest
        - Outro
            a. Plot results
            b. Display any relevant data
2. Comments
    a. Syntax
    b. Where they're needed
        - Program Heading
        - Introduce body and outro sections of code
        - Detail code that may not be obvious
3. Naming conventions
    - Variables
    - Functions

Exponential Growth Model

AP Environmental Science – 4/2/14

The goal of this lesson is familiarize yourself with the way to structure a program properly. While you will implement this in the R language, it is universally applied to just about every language you might encounter.

This will be your first time using R to do something more than strictly programming. While you will see how to implement the good programming structure that was discussed in class, you will also be learning about the exponential growth model that is used to describe various species populations. In an exponential growth model, the species population continuously increases at an exponential rate. This is shown in the figure below (generated using R!).



The exponential growth model is characterized by the equation below. IP stands for the initial population of the species, GR stands for the specific growth rate of the species and t stands for time. For this model, we want to simulate the growth of the species over three years; that is, we want our time variable to span from 0 – 3 years. For parameters we are going to use an initial population of 20,000 and a specific growth rate of 2 $years^{-1}$. $Years^{-1}$ isn't a very intuitive unit, but you can think of specific growth rate as implying that each member of the population will generate 2 offspring per year.

$$Population = IP * e^{(GR * t)}$$

Getting Started

1. Open Rstudio
   a) Open Finder
   b) Open Applications
   c) Open RStudio

2. Create a new R Script
    a) File
    b) New File
    c) R Script
3. Add the title section to your script file using comments
    a) Name
    b) Title
    c) Brief description
    d) Date(s) created (and modified)

Program Intro

The program intro is where you can define your known parameters and set the ranges of your independent variables. We want to use the ones defined in the problem statement on page one.

1. Define a scalar variable named initPop that has the value 20,000.
2. Define a scalar variable named growRate that has the value 2.
3. Define a vector variable named time that takes on values between 0 and 3 with a spacing of 0.01.
    a) Hint: enter ?seq at the command line

Program Body

The program body is where all of the math simulations are performed. In this problem we want to use the exponential growth model to determine the population size as a function of time. To do this we make use of the exp( ) function and several of the mathematical operators.

1. Define a vector variable called popSize and give it the values determined by evaluating the model on page one.

Program Outro

The program output is where we output any data that is relevant to the model we are simulating. Since we wanted to determine the population size as a function of time in this problem, we will ouput a graph of population size as a function of time (this would be significantly easier to interpret than simply printing a data table).

1. Create a plot window.
    a) hint: xlim and ylim arguments
2. Add a grid to your plot.
    a) hint: Enter ?grid at the command line
3. Plot the population as a function of time.

a) hint: Enter ?lines at the command line

Comments

Comments are used to make it easier to understand what your code is doing. We have already added comments to the very beginning of our code that summarize who wrote the code, when it was written and what the overall aim of the program is. Comments should also be added to do the following:

1. Introduce code segments.
    1. Intro example:
        # Intro Segment
        # Define initial population, specific growth rate and time scale
    2. Body example:
        # Main segment
        # Simulate the exponential growth model over the specified time range
    3. Outro example:
        # Ouput segment
        # Output and format the data to be easy to interpret
2. Describe lines of code that are confusing to understand. This isn't too critical yet, but in the future as your code gets more complex you will want to add comments to lines of code that aren't understood as easily as simple math or plotting functions.

Extra Credit

If you finish early, rerun the simulation with the following changes:

1. Halve the growth rate (GR = 1). Plot the result on the same chart in a different color.
2. Halve the growth rate and double the initial population. Plot the result on the same chart in a different color.
3. Add a legend to your plot that indicates what the three curves represent.
    1. Hint: Enter ?legend at the command line