MSDS7333 Case\_Study\_12

Simulation Study of a Branching Process

Scott Anderwald, Chandon Nahar, Celia Taylor

12/09/2017

# Introduction:

# In this assignment, we are trying to model the branching process using the Monte Carlo Method, which is just a way to generate a sample from a random distribution. The basic idea is that we start with a job which spawns other jobs. Upon job creation, that job can generate offspring jobs. The “parent” job takes a certain amount of time to finish. While this “parent” job is running, it can spawn other “child” jobs which cannot run until the parent job finishes.

# Background:

# This branching process is representative of a natural process as well and happens in computer programming. In addition, there is a recursive nature to this process. Therefore, the usual way of applying the probability distribution does not work. In essence, we will be generating these random jobs to fill in a high-dimensional space and our intent is to understand what the ultimate distribution turns out to be.

This case study is from Chapter 7, "Simulation Study of a Branching Process", of the book or .pdf by Deborah Nolan and Duncan Temple Lang, "Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving".

http://rdatasciencecases.org/ is the website address associated with this book. Code associated with Chapter 7 is at http://rdatasciencecases.org/BirthDeathProcess/code.R. Data for this case study is generated randomly by the code.

# Method:

In this case study, the project team will utilize a Monte Carlo Method to generate independent random results from its probability distribution. Then we use the properties (mean and Standard Deviation) of the observed results as approximations to the expected properties. Basically, we study the behavior of thousands of random variables in order to discern the probable behavior of the outcome.

The strengths of this process are that we can figure out the samples size to provide proper precision, use statistical principles to summarize and evaluate the results, and utilize experimental design to change parameters to better understand and study the process.

The cautionary aspects are that the study needs to be carefully designed so that the problem is actually reproducible and representative. The results will not be the same every time because it is a random process, but will be reproducible by remaining statistically coherent. Also, because the results are not the same every time, testing and debugging of the code must be done very carefully.

For this case study, the project team decided on answering Chapter 7, Question 10. Question 10 requires executing the simulation study with different values for the parameters. Question 10 suggests holding the kappa parameter constant at the value of 1 and running the simulation with various values of lambda.

Parameter lambda is the birth rate.

Parameter kappa is the completion rate or the lifetime.

In order to keep the run time of the algorithm at a reasonable level, the number of random outcomes studied was reduced from 400 to 40.

The relationship between lambda and kappa is very important to how long the process runs and how many generations occur. The simulation studies what happens when lambda/kappa ranges as less than one, close to one, and greater than one. To do this, we ran 5 different trials with kappa remaining constant at one.

Values below original or baseline run

Trial 1 for Plot 1

kappa = 1 and lambda ranges from 0.01 to 0.2 by increments of 0.1

kappa = 1 and lambda ranges from 0.4 to 1.6 by increments of 0.2

kappa = 1 and lambda ranges from 1.25 to 2.0 by increments of 0.25

---------------

Trial 2 for Plot 2

kappa = 1 and lambda ranges from 0.0015 to 0.4 by increments of 0.1

kappa = 1 and lambda ranges from 0.6 to 1.6 by increments of 0.2

kappa = 1 and lambda ranges from 1.75 to 2.75 by increments of 0.25

---------------

Original or baseline run

Trial 3 for Plot 3

kappa = 1 and lambda ranges from 0.1 to 0.6 by increments of 0.1

kappa = 1 and lambda ranges from 0.8 to 2.0 by increments of 0.2

kappa = 1 and lambda ranges from 2.25 to 3.0 by increments of 0.25

---------------

Values above original or baseline run

Trial 4 for Plot 4

kappa = 1 and lambda ranges from 0.2 to 0.8 by increments of 0.1

kappa = 1 and lambda ranges from 1.0 to 2.2 by increments of 0.2

kappa = 1 and lambda ranges from 2.50 to 3.25 by increments of 0.25

---------------

Trial 5 for Plot 5

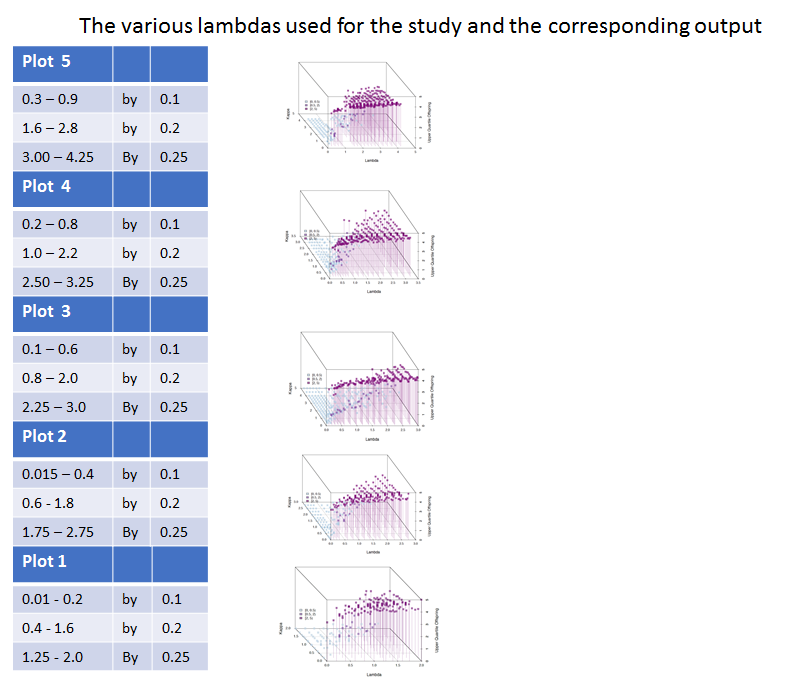
kappa = 1 and lambda ranges from 0.3 to 0.9 by increments of 0.1

kappa = 1 and lambda ranges from 1.6 to 2.8 by increments of 0.2

kappa = 1 and lambda ranges from 3.0 to 4.25 by increments of 0.25

# Analysis with Parameter Adjustment:

For case study 12 the project team decided to answer Question 10. Parameters selected for the baseline study (plot 3) are from the example presented in the book used for the class. The project team then decided to obtain sets of numbers for lambda that would be above and below the baseline study. The selection, in theory, should reveal any possible trends in the data. The diagram below is the parameter selection and the output.



# Results:

For the case study, the randomly picked lambda values do reveal a possible trend in the data. By keeping kappa, the lifetime value, constant and equal to one and increasing lambda numerically, the birth rates have a corresponding correlation with the lambda values. Starting with plot 1 (from above) the birth rate has the lowest density of the five trial runs. The smallest overall values of lambda are in trial 1. Revealed in the diagrams above as the lambda values change to higher values, there is a possible increase in birth rate seen in Plots 2 thru 5. Plot 5 has highest overall lambda values and the highest density for birth rate in the study. As Plot 1 is the most sparsely distributed output, one can see as you increase lambda there is an increase in density for the values in the Plots 1-5 for individual points in the diagram.

# Conclusions:

After comparing the base study to the randomly selected lambda values for Question 10, it appears that there is a possible trend in the birth rates. As lambda increases, the population density for birth rate has a corresponding result. After researching the definition for lambda, one should possibly expect to see this change in birth rates. By comparing the output from the Plots1 through Plot 5 above, one can see the possible trend in the increase of birth rates compared to the lambda values. Because kappa, the lifetime of a job, is kept constant at one, the value changes in lambda truly depict or highlight birth rate changes.