Assignment #10:
Random Processes and populations

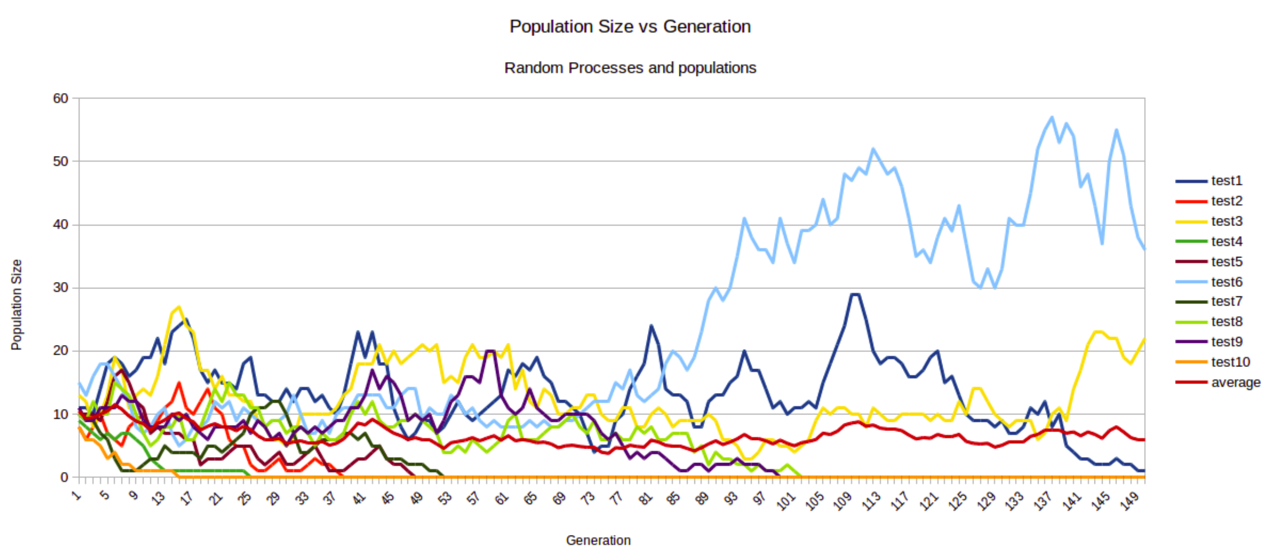
Timmy Nguyen

April 24, 2017

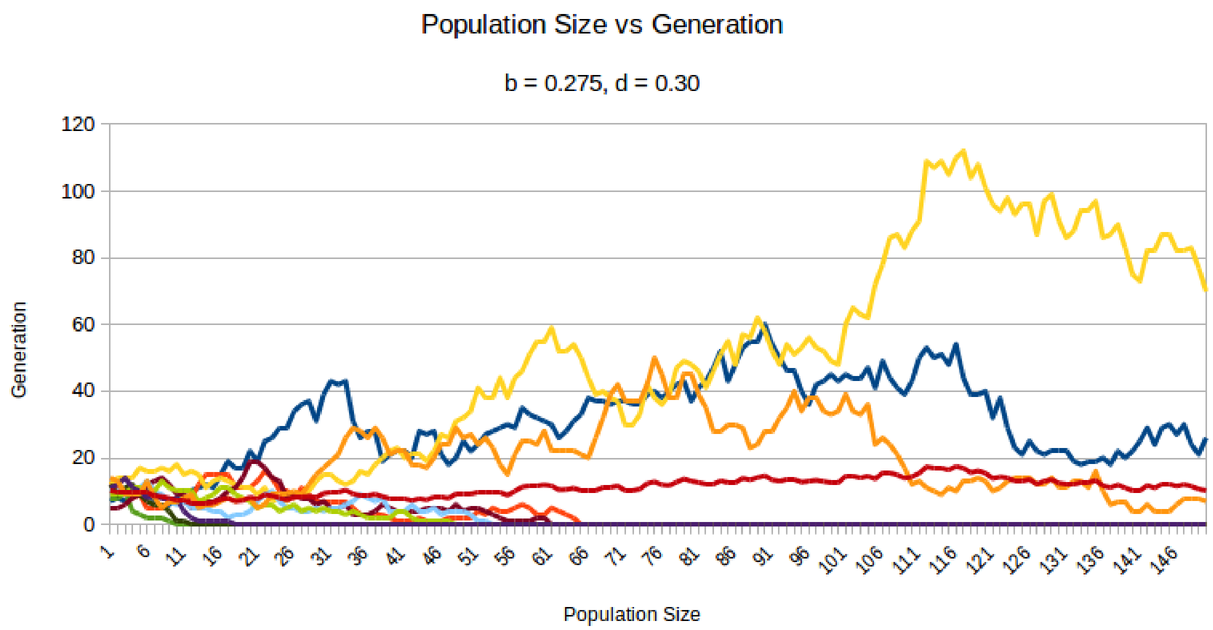
1 Overview

In this lab, I am to develop a stochastic model approach to population dynamics. Here, I am to analyze results from my model output. Learn the difference between deterministic vs stochastic. Learn effects on small populations vs large populations.

2 Simulation 1: b = 0.30, d = 0.275



3 Simulation 2: b = 0.275, d = 0.30



3.0.1 Summary

Unlike a deterministic model, the stochastic model introduces randomness as evident in the numerous upwards and downwards trends in the graphs. However, both shows a similarity when it comes to long term results.

When looking at the two graph results, I notice that when the birth rate coefficient is higher than the death rate, the population have a lower likely hood to die off. As a result, there are more large populations then small ones. Small differences in our birth and death coefficients can impacts our results, which can lead a population to be able to propagate or die off.

Individuals have a greater impact on earler generations due to the population being small. When comparing the average run to the others, the average run tends to stabalizes at a given population size.

4 C Code

```c

/**
* Name: Timmy Nguyen
* Assignment #10: Random Processes
* BIOL 480 Spring 2017
* Date: 4/19/2017
*/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

#define b 0.275    // Birth Rate Coefficient
#define d 0.30     // Death Rate Coefficient

int  main()
{
  // Seed
  srand(time(NULL));

  // Init Variables
  int pop_size;
  int births, deaths, gen;
  double birth_probability, death_probability, x;


  // Creating Array
  double N[150];

  pop_size = 10;


  // Main Code
  for (gen = 0; gen < 150; gen++) {

    births = 0;    deaths = 0;

    birth_probability = d * (1 - b);  death_probability = d * (1 - b);

    for (int i = 1; i <= pop_size; ++i) {

```

```
        // Random Number Generator
        x = (double)rand() / (double)(RAND_MAX);

        // X is between 0 and P+
        if ( x > 0 && x < birth_probability )

          {
            births++;
          }

        // X is between P+ and (P+ - P-)
        else if ( x > birth_probability && x < (birth_probability + death_probability) )
          {
            deaths++;
          }
      }


      // Current Population Size
      pop_size += (births - deaths) ;
      N[gen] = pop_size;
    }

    // File Output for Display
    for (int i = 0; i < 150; ++i)
    {
      printf("%d     %lf\n", i, N[i] );
    }

    FILE *outfile;


    outfile = fopen("output.txt", "w+");
    for (int i = 0; i < 150; ++i)
    {
      /* code */
      fprintf(outfile, "%7.6f\n", N[i] );
    }

    fclose(outfile);


}
```