High Performance Computing Programming Exercises

Question 8

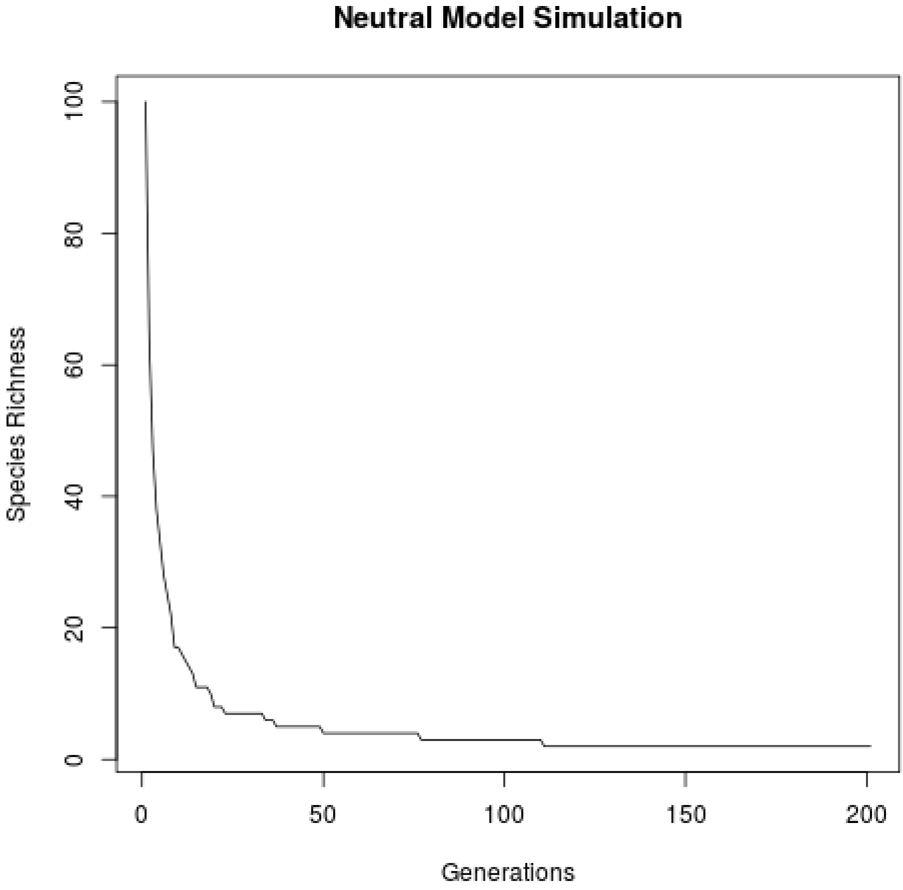


Figure 1. Graph showing Species Richness over time, when simulated using a neutral model. Initial species richness is set to 100, but will always converge to 1, as there is no speciation parameter included in the model. Over time, species will go extinct and can only be replaced by currently existing species, eventually, this leads to the existence of just a single species.

Question 12

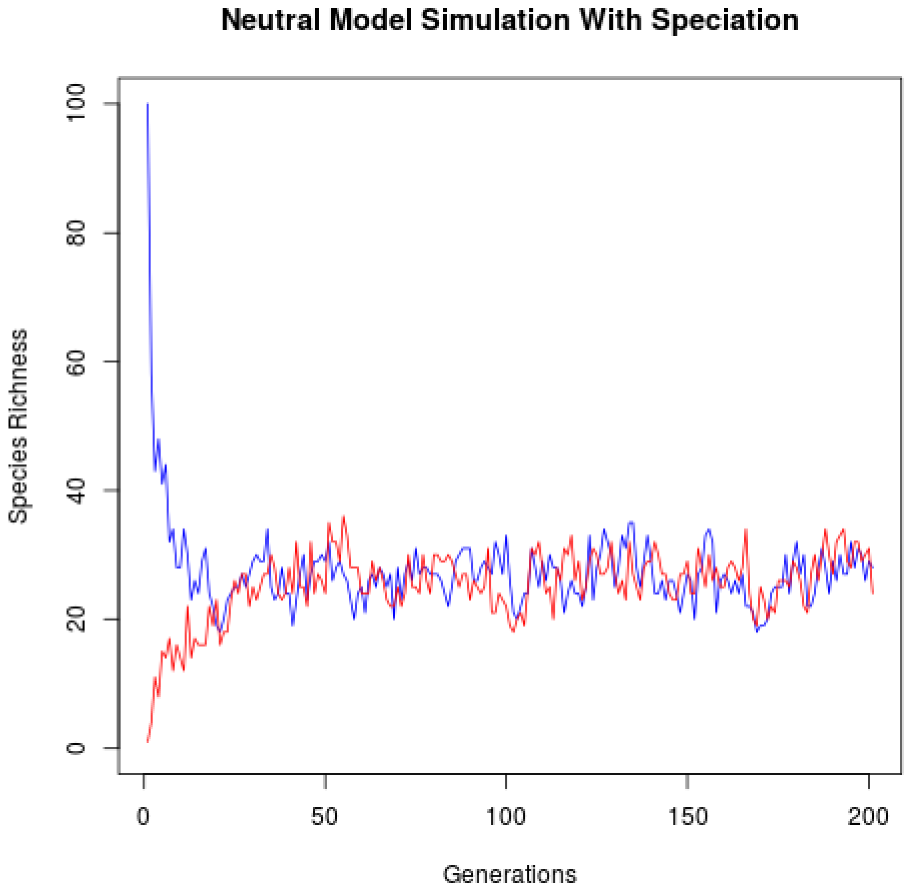


Figure 2. Graph showing Species Richness over time when simulated using a neutral model for two different initial community structures; a speciation rate of 0.1 was included in the model. The blue line represents the change in species richness over time for a community of size 100 with maximal initial diversity (i.e. species richness = community size). The red line represents the change in species richness over time for a community of size 100 with minimal intitial diversity (i.e. species richness = 1). The two lines converge to the same dynamic equilibrium, as the speciation rate balances with the extinction rate.

Question 16

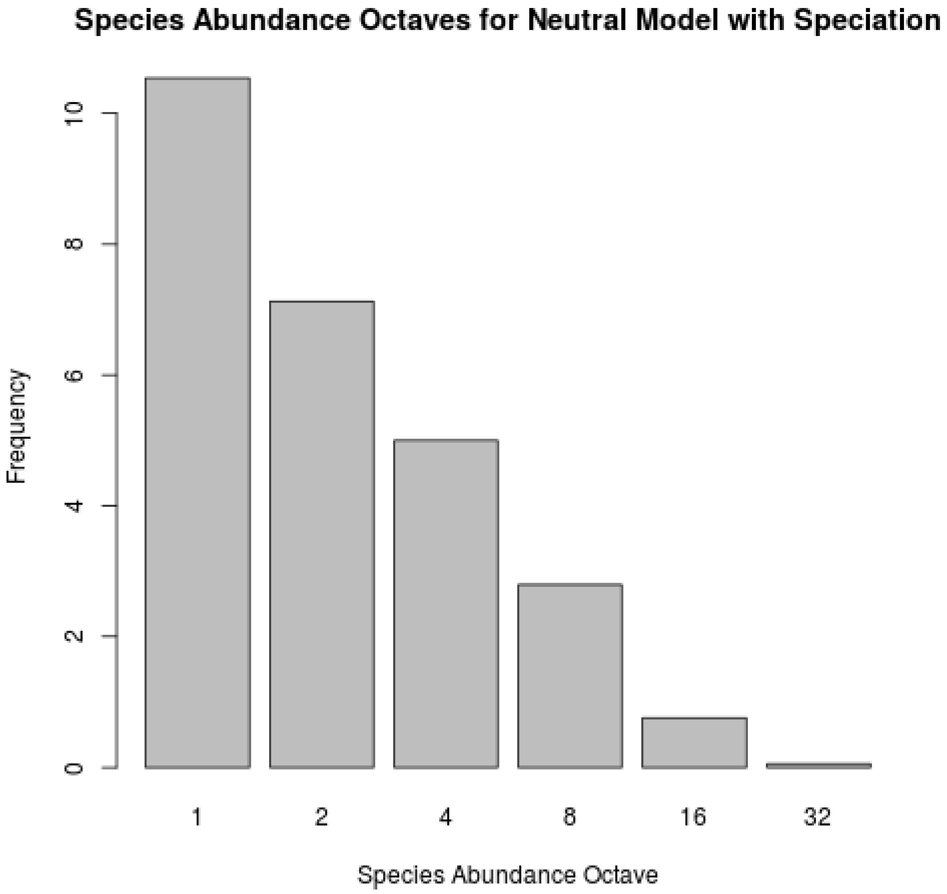


Figure 3. Bar plot showing mean species abundance distribution as octaves, for a neutral model simulation with a speciation rate of 0.1. A burn in period of 200 generations was used to allow for a dynamic euqilibrium to be reached. After this period, the model was run for a further 2000 generations, with abundance octaves being recorded every 20 generations. The initial species richness of the community does not affect the outcome of this simulation, as can be seen in Figure 2.

Challenge A

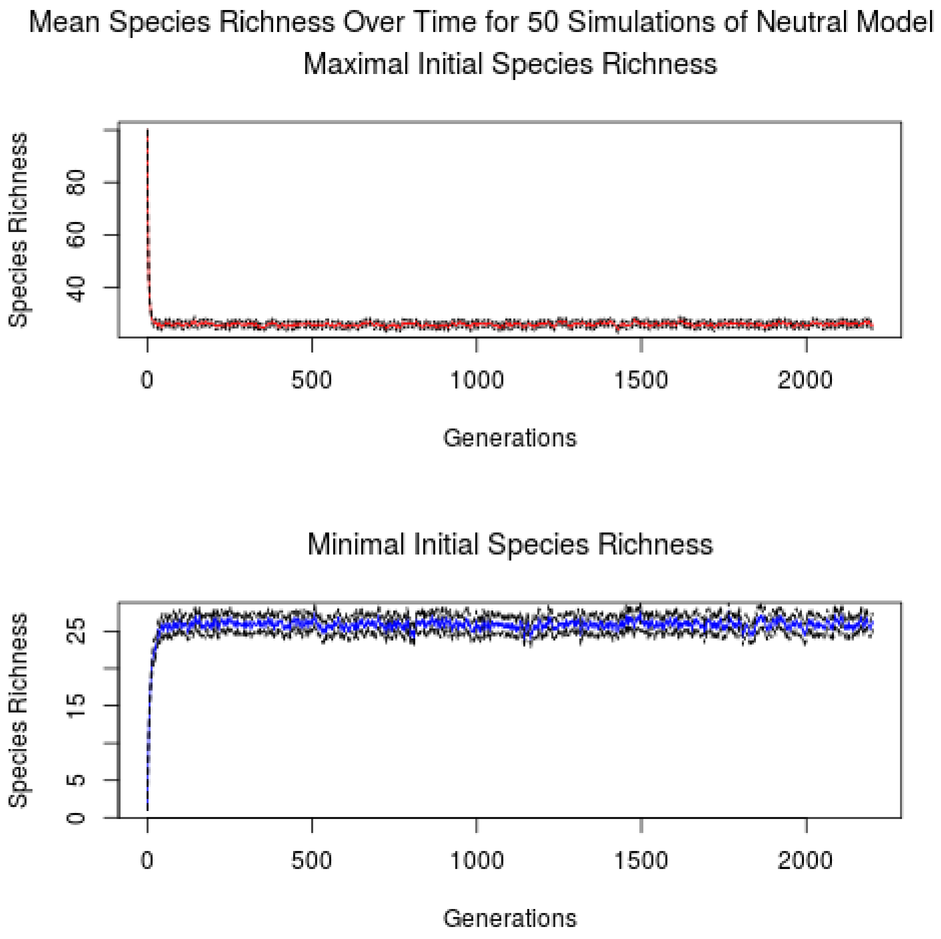


Figure 4. Mean species richness over time for 50 simulations of a neutral model with a speciation rate of 0.1. The dotted black lines represent the 97.2% confidence intervals for the calculation of the mean. When the initial species richness is set to 100, the community reaches dynamic equilibrium after about 15 generations. The community reaches a dynamic equilibrium after about 25 generations whith an initial species richness of 1. This is because the difference between the extinction rate and speciation rate when starting from higher diversity is larger than when starting from a lower diversity, therefore the community is altered at a higher rate.

Challenge B

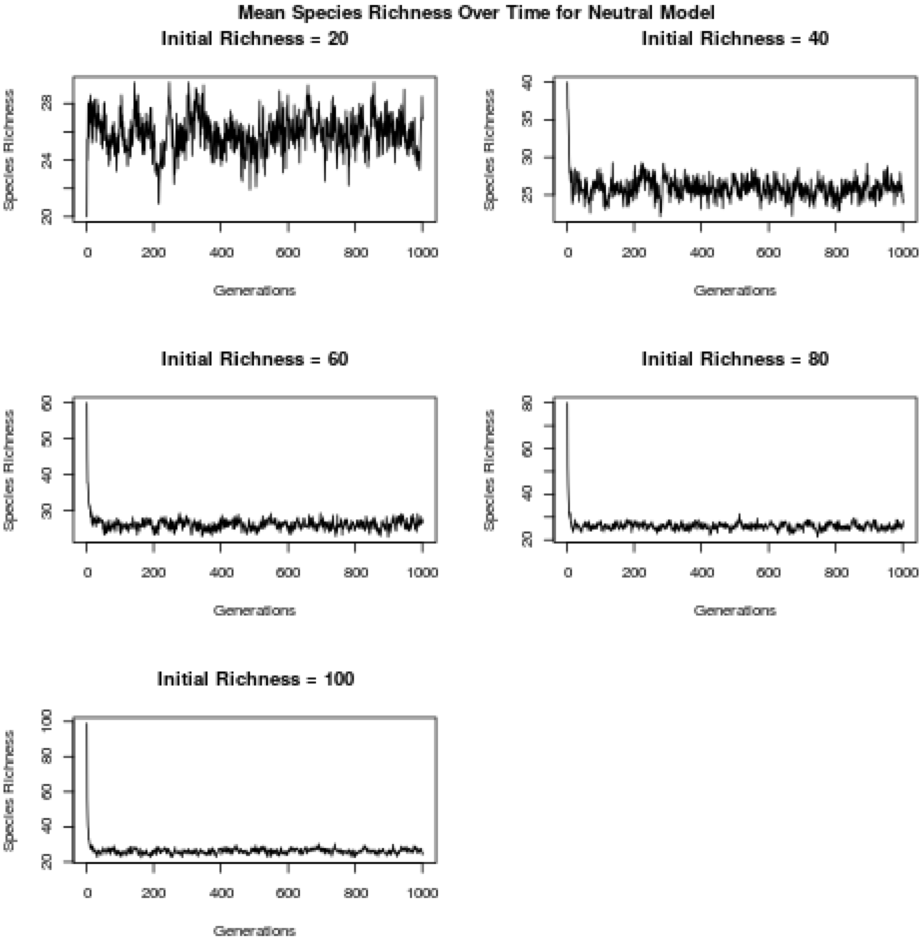


Figure 5. Mean species richness over time for communities of various initial diversity simulated using a neutral model with a speciation rate of 0.1. Each initial speceis richness was simulated 10 times.

Question 20

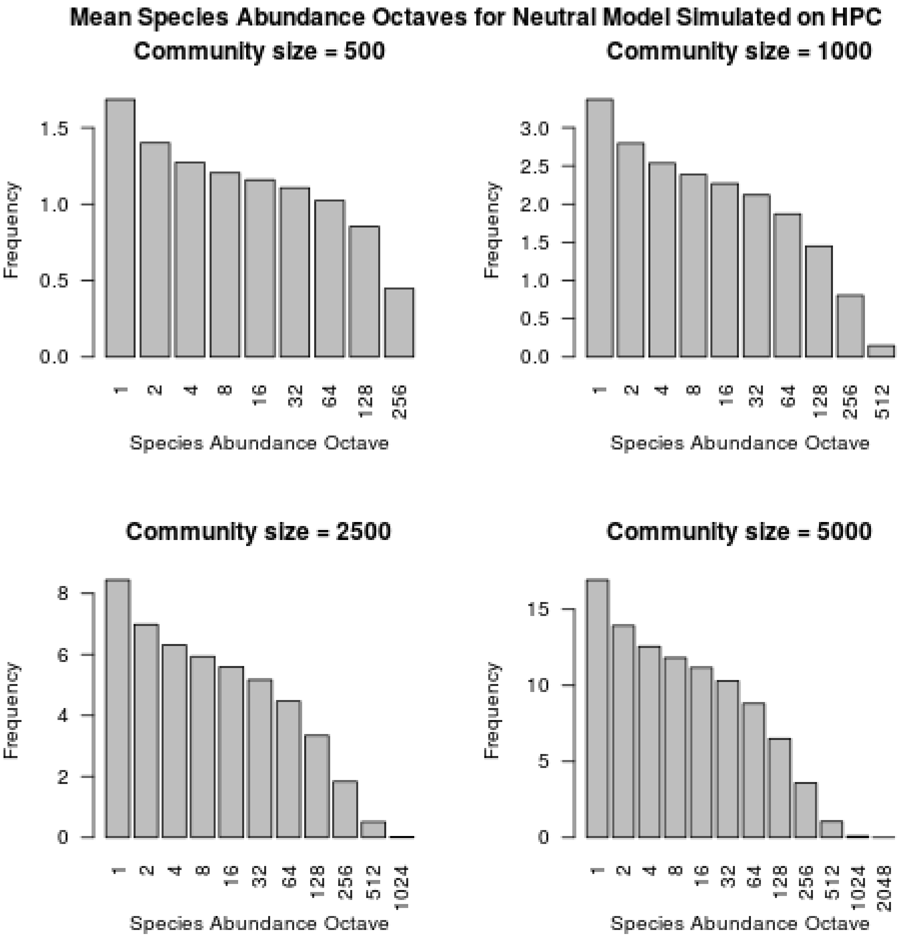


Figure 6. Mean species abundance octaves, after a burn in period, for multiple simulations of a neutral model with a specation rate of 0.003369. 25 simulations were run for each community size. In all simulations, communities contain more rare species than common species.

```
[1] "Mean octaves for J = 500"
[1] 1.6844065 1.4005617 1.2723002 1.2045483 1.1585608 1.1065548 1.0227578
[8] 0.8539114 0.4472141

[1] "Mean octaves for J = 1000"
 [1] 3.3680107 2.7949303 2.5360329 2.3842639 2.2675221 2.1196566 1.8714663
 [8] 1.4501992 0.8048670 0.1474153

[1] "Mean octaves for J = 2500"
 [1] 8.44114418 6.98120234 6.31358120 5.93501677 5.59338002 5.16736892
 [7] 4.46615827 3.34221536 1.83665173 0.50955461 0.02499926

[1] "Mean octaves for J = 5000"
 [1] 1.693528e+01 1.393268e+01 1.256032e+01 1.181413e+01 1.116928e+01
 [6] 1.029987e+01 8.814542e+00 6.501815e+00 3.586246e+00 1.056115e+00
[11] 8.692044e-02 2.074474e-04
```

Challenge C

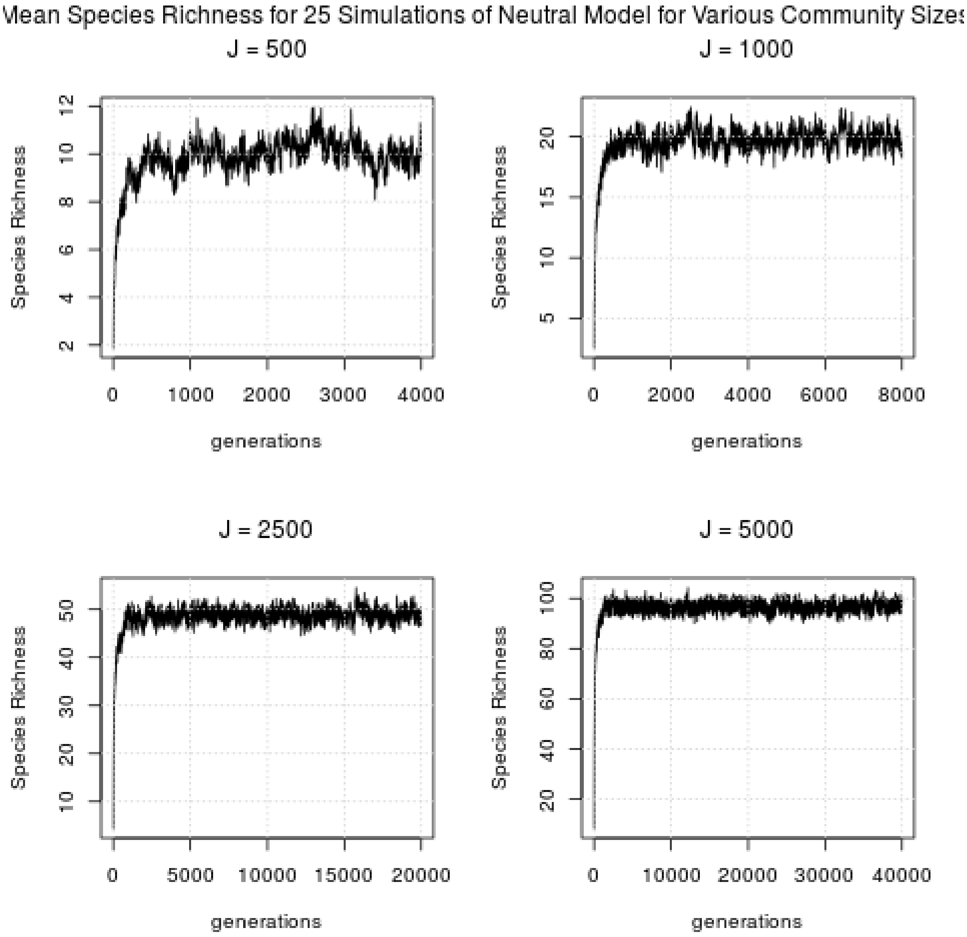


Figure 7. Species richness measured over a burn in period of 200 generations, for a neutral model simulated with a speciation rate of 0.003369. Each simulation reaches a dynamic equilibrium after about 1000 generations, although the burn in period used for the simulations was much larger than this.

Challenge D

The resuts of the coalescence simulations are generally in agreement with those of the simulations run on the HPC cluster. The species richness of the communities produced by the coalesence simulations (indicated by the length of the abundance vector) are in line with the dynamic equilibriums shown in Figure 7.

The coalesence simulations took only 4.781 seconds to run, as opposed to the 1150 cpu hours spent on the neutral models on the HPC cluster. This reduced time is due to the fact that the coalesence method does not waste time simulating lineages that will not be present in the final community; rather, it extrapolates from a sample of an initial community. Another way the coalesence method saves time is by not simulating a burn in period.

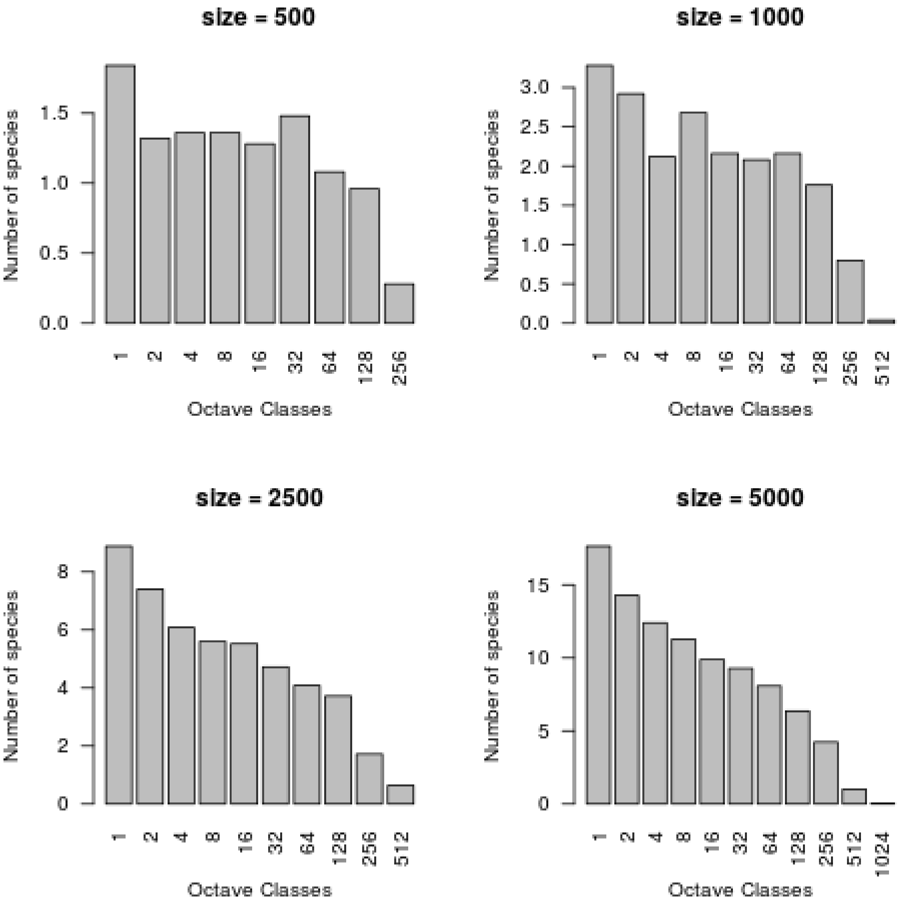


Figure 8. Mean species abundance octaves, of communities simulated with a neutral model using a coalesence method. A specation rate of 0.003369 was used. 25 simulations were run for each community size.

Fractals

<u>Question 18</u>

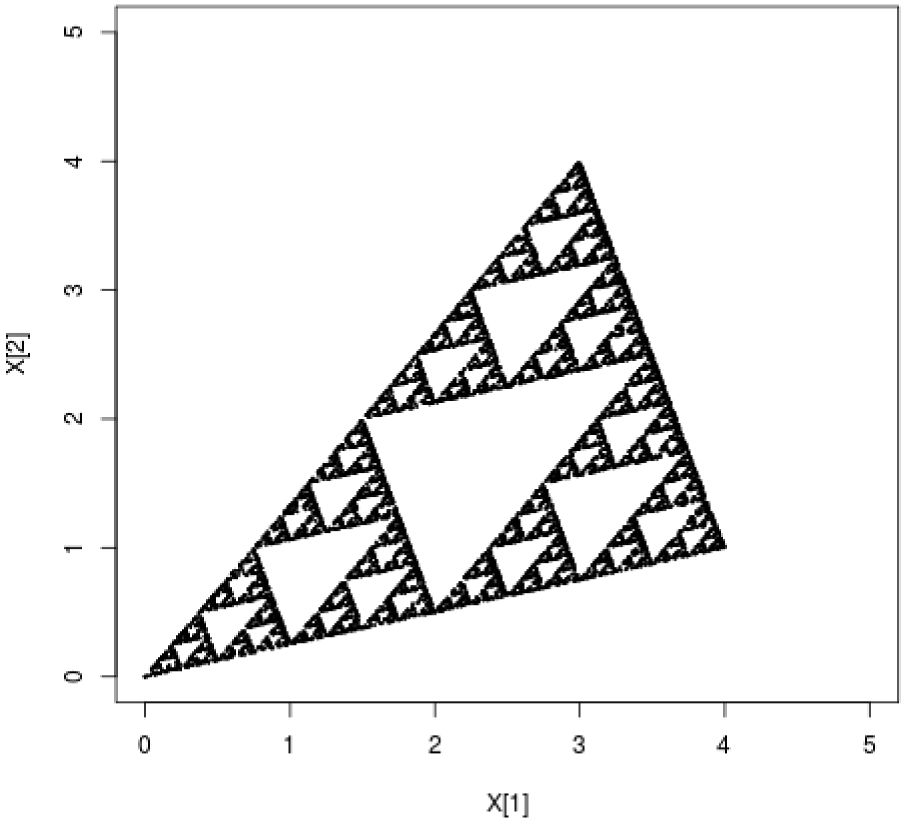
The fractal dimension of the first object;
log(8)/log(3) = 1.892789

The fractal dimension of the second object;
log(20)/log(3) = 2.726833

The formula used to calculate fractal dimension of a self similar fractal pattern is D = $\frac{\log N}{\log S}$, where D is the fractal dimension, N is the number of self-similar units needed to make up the entire object, and S is the factor by which the length of one side of a self-similar unit must be multiplied to be made equal to the length of one side of the entire object. Since N = S^D, if we take logs of both sides of the equation we get; log N = D log S. Dividing both sides of the equation by log S, gives D = $\frac{\log N}{\log S}$.

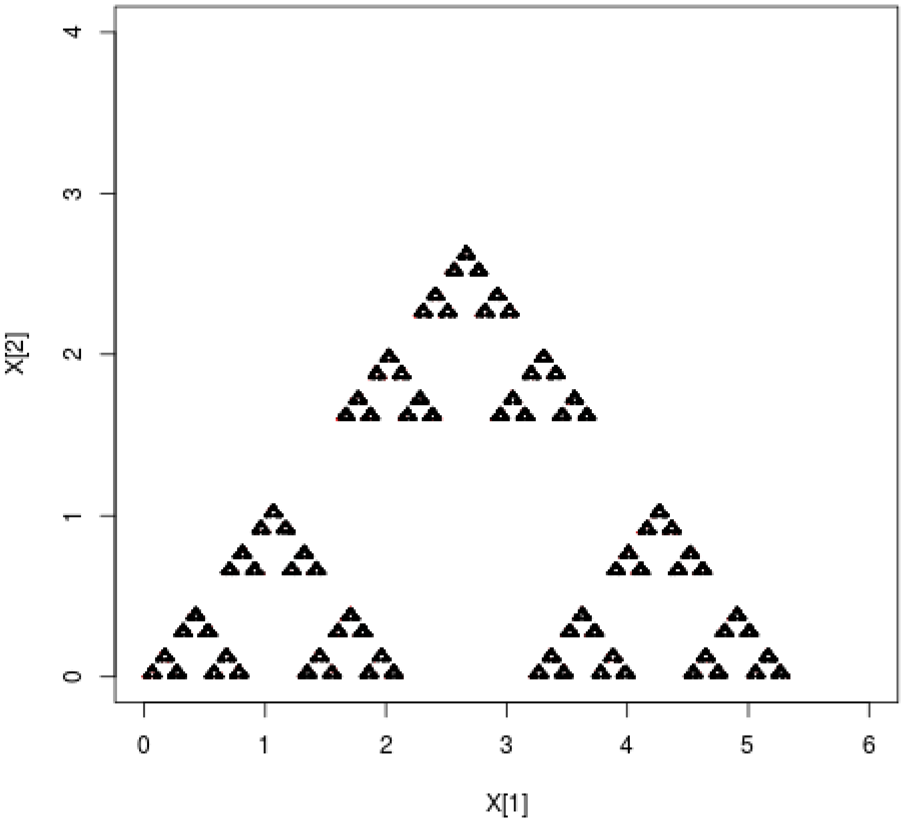
In the example, the first object is made up of 8 self-similar units, and the length of each side of the entire object is equal to the length of 3 of the self-similar units, hence the fractal dimension can be found using the formula, log(8)/log(3) = 1.893.

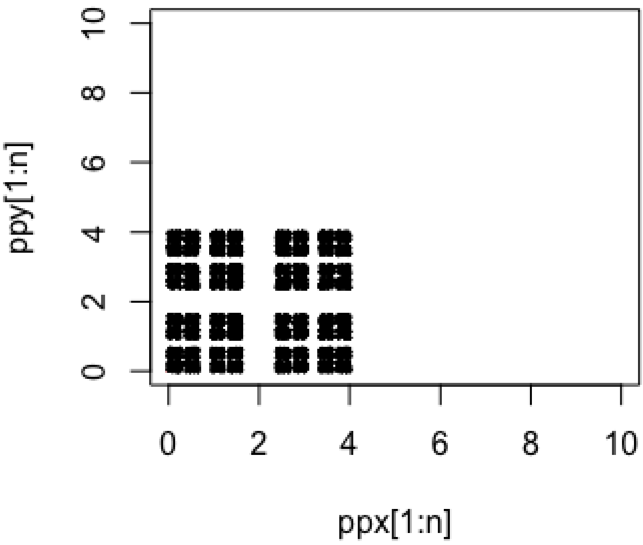
Question 19



Challenge E

Classic
Sierpinski
Gasket



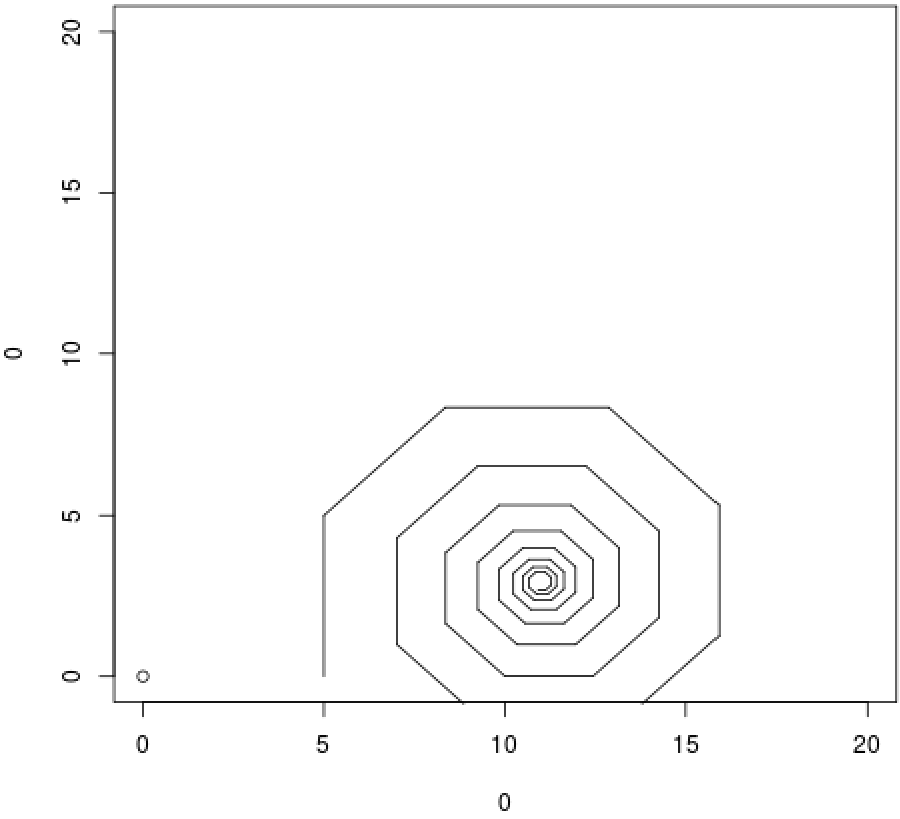


The starting position in chaos_game() doe not have any major impact on the shape drawn by the function, as after just a few points have been drawn, all new points will be within the limits set by vectors a, b, and c.

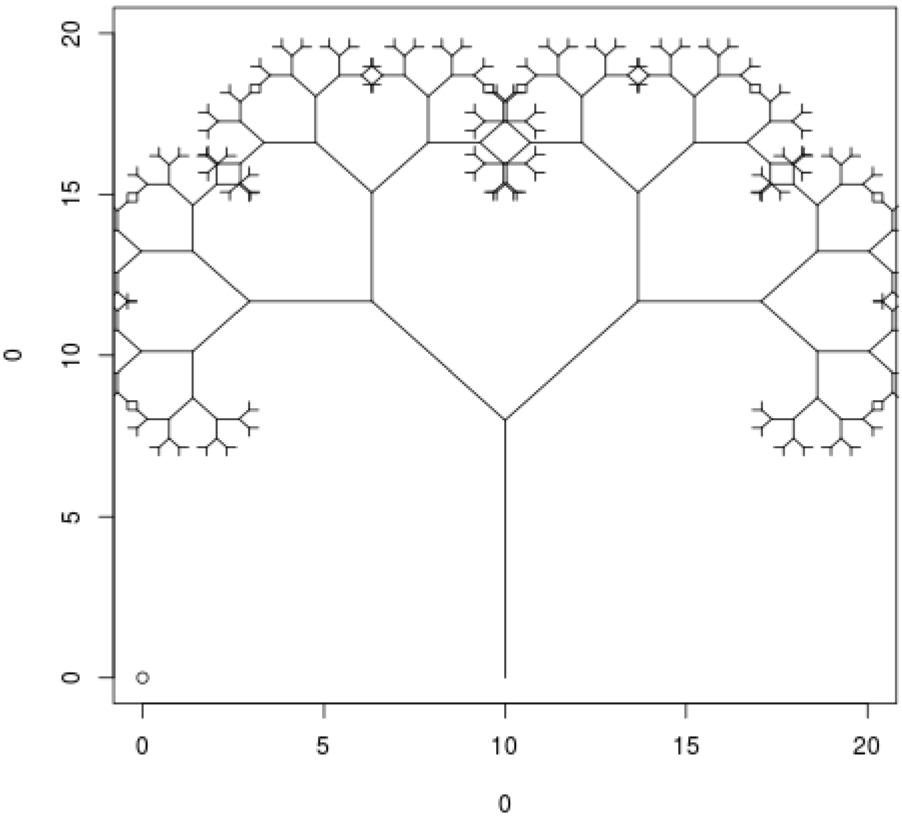
Question 25

The new function spiral calls itself an infinite number of times, this produces the following error message:
Error: evaluation nested too deeply: infinite recursion / options(expressions=)?

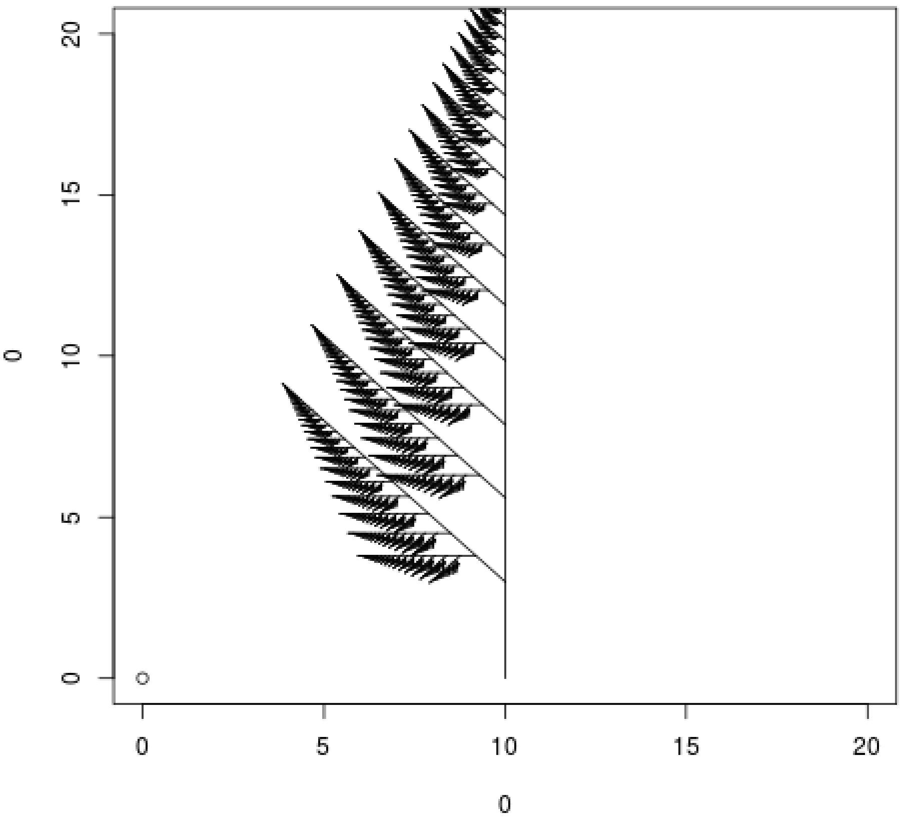
Question 26



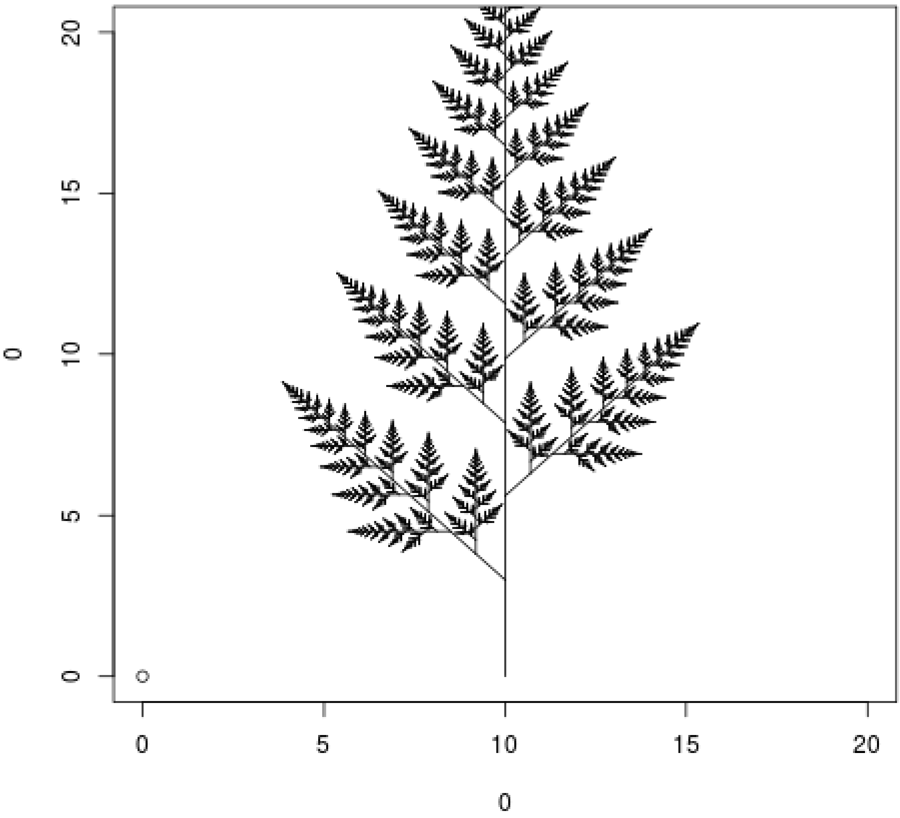
Question 27



Question 28



Question 29



Challenge F

As the value of e gets smaller, the function challenge_F() takes longer to run, this is because each branch of the fractal pattern continues for longer, with more detail added to each branch.