HPC Report

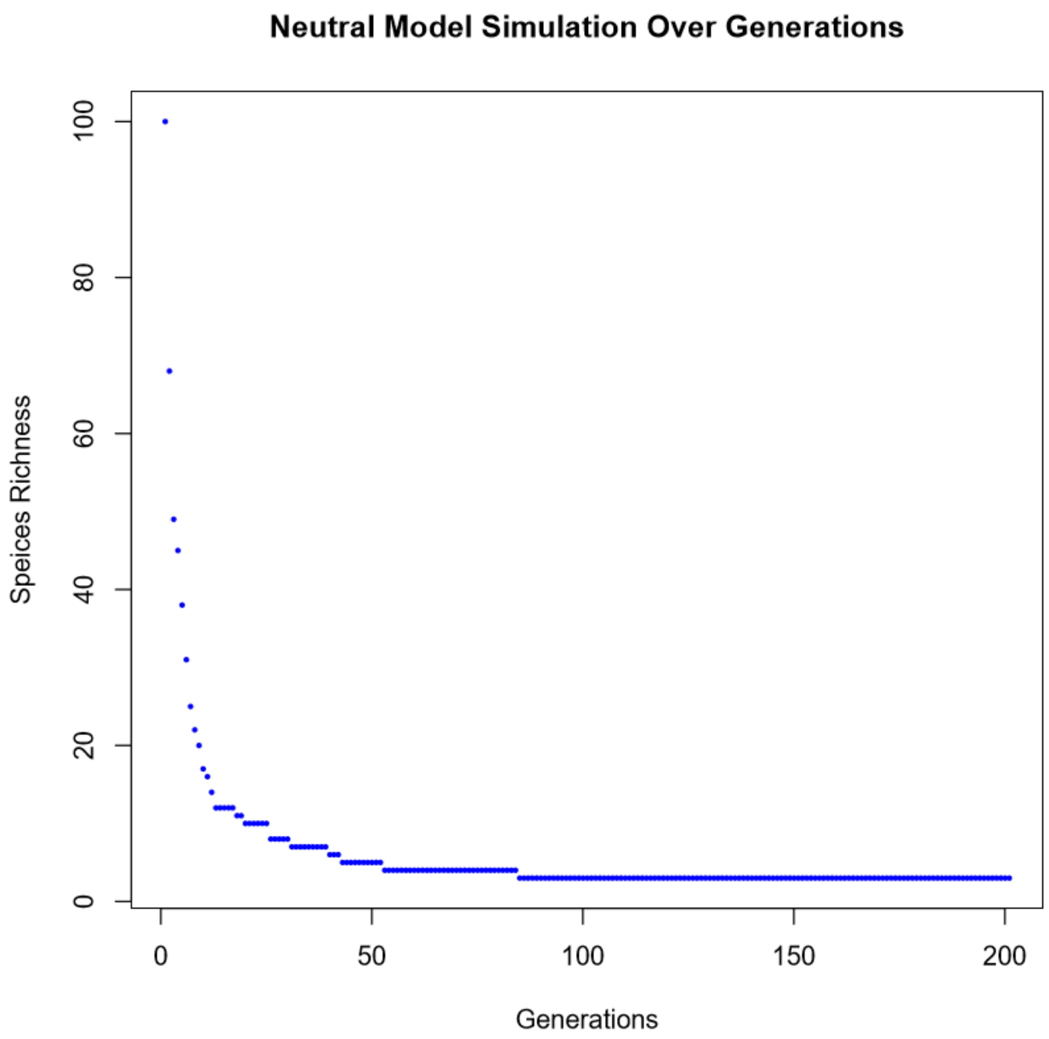
Author: Jiqiu Wu
01618530
j.wu18@imperial.ac.uk

Supervisor:
James L Rosindell
j.rosindell@imperial.ac.uk
Imperial College London

MRes in Computational Methods in Ecology and Evolution, 2018-2019
Department of Life Science
Silwood Park Campus
Imperial College London

1 Question 8:

Plot a time series graph of your neutral model simulation from an initial condition of maximal diversity in a system size of 100 individuals. Run the simulation for 200 generations. Make sure that the axes are properly labelled and that the write up contains a suitable title and caption for the graph. Include the code you wrote for this question in a function called ***'question_8'*** which should require no inputs to run.

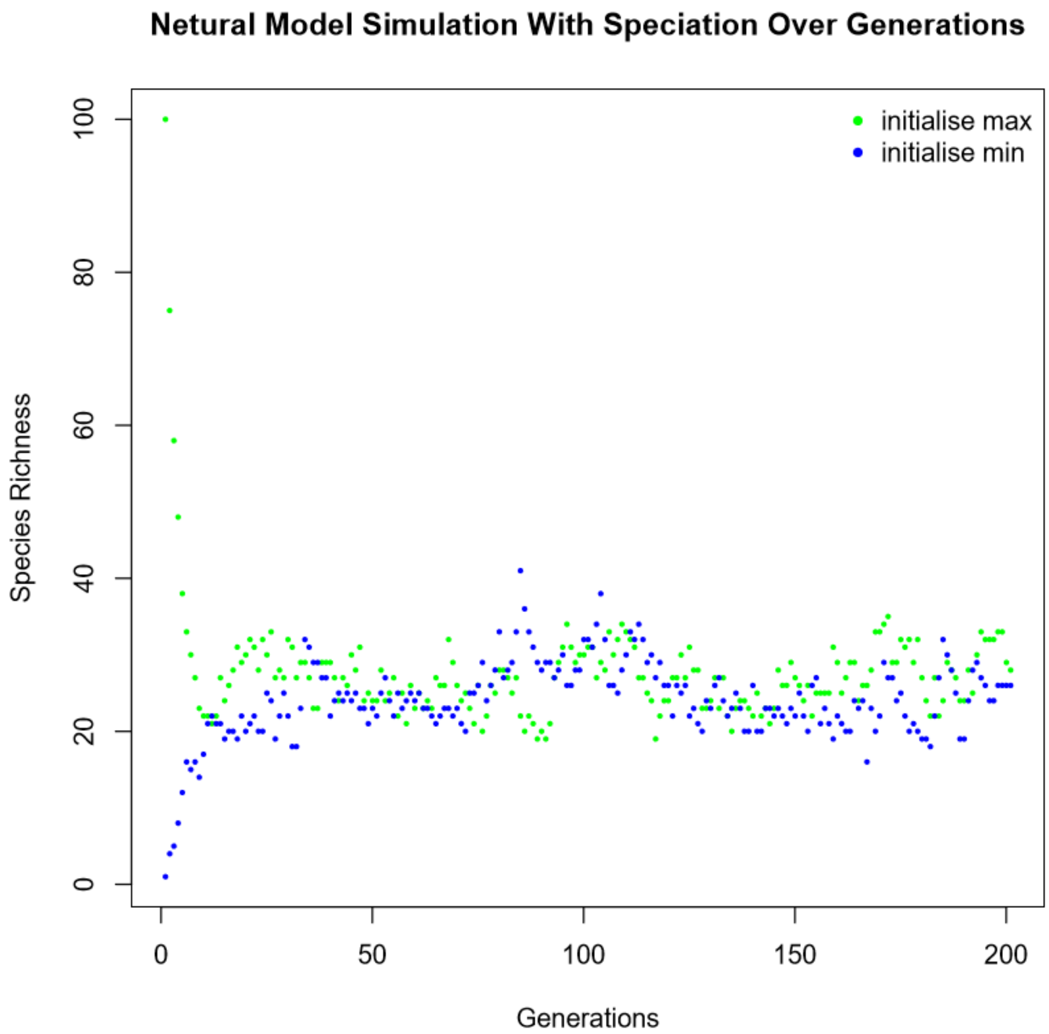


★ What state will the system always converge to if you wait long enough? Why is this? (Hint: use your own function ***neutral_time_series*** from above, as well as the plot command) [3 marks]

This system will converge to 1 if waiting for long enough. The reason is that every individual that dies is equivalent to a reduction of one species, and no new species appears. Therefore, only one species will remain after a long enough time.

2 Question12

★ Perform a neutral theory simulation with speciation and plot species richness against time as you above. Use a speciation rate of ν= 0.1, a community size of J = 100 and run your simulation for 200 generations. Plot two time series on the same axes in different colours showing how the simulation progresses from two different initial states given by ***initialise_max*** and ***initialise_min***. Include the code you wrote for this question in a function called ***'question_12'*** which should require no inputs to run.



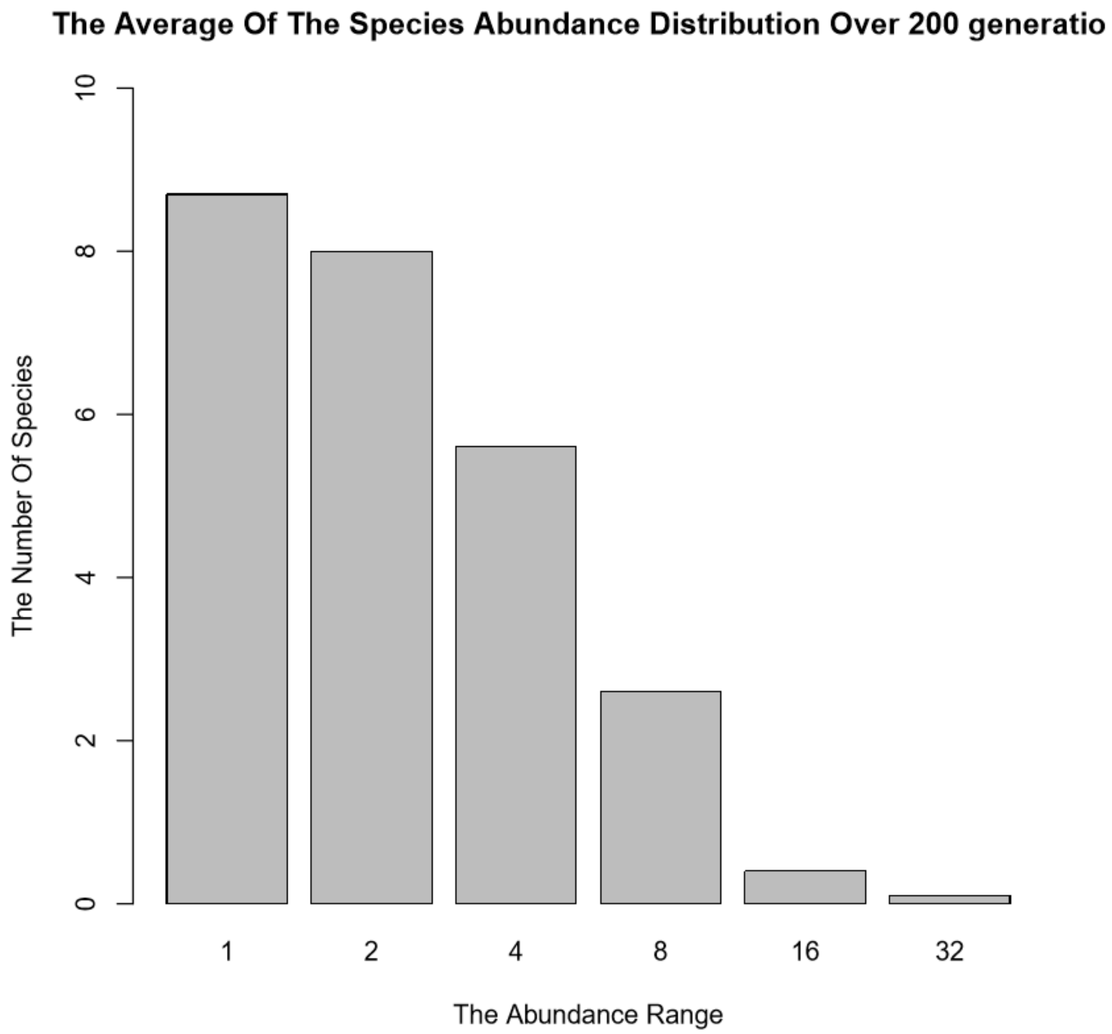
★ Explain what you found from this plot about the effect of initial conditions. Why does the neutral model simulation give you those particular results? [4 marks]

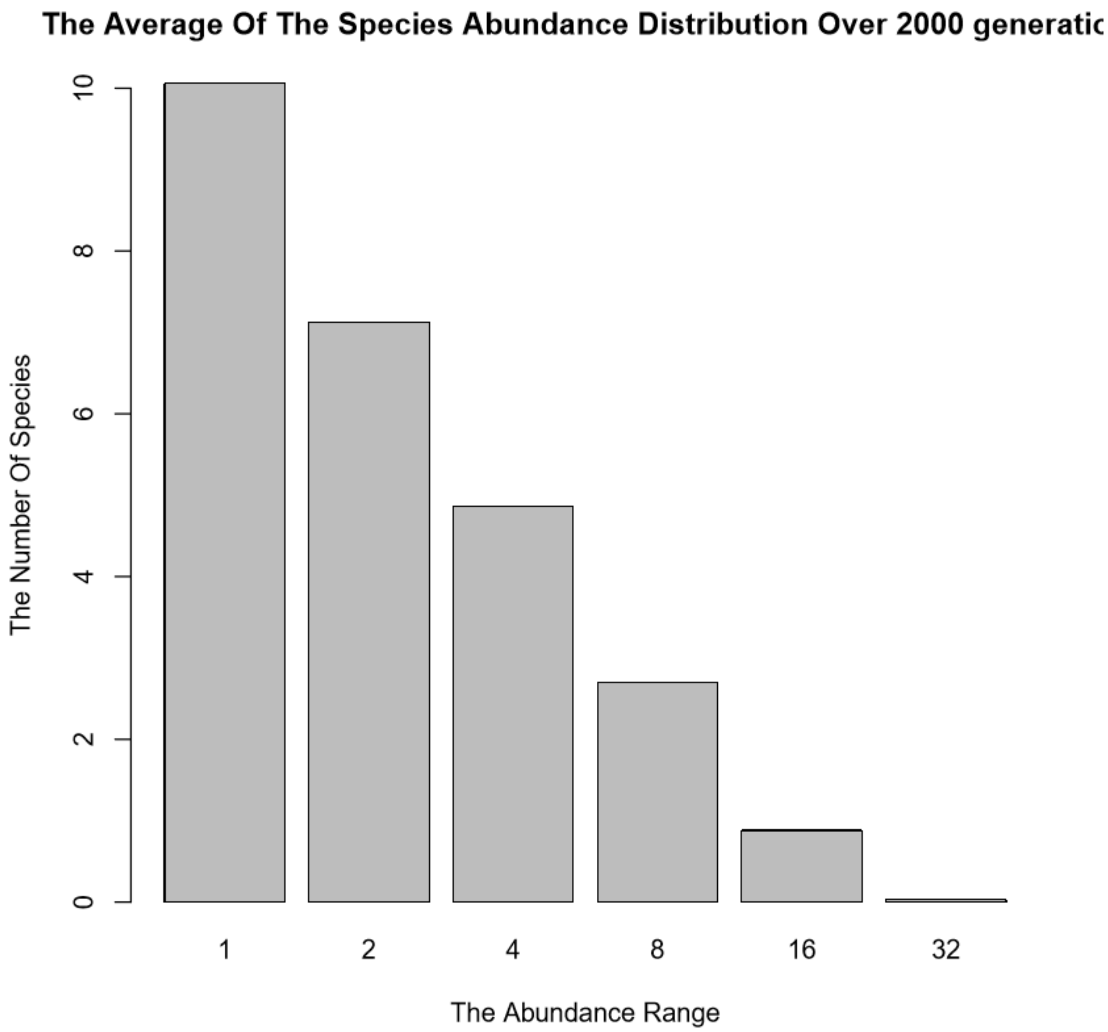
Starting from the two States, ecological equilibrium was reached in the end. The reason is that they have the same speciation rate, so if the community size is the same, ecological equilibrium will be reached. However, the further reason is still under consideration. I am still reading a paper[1] on this issue, to be continued ...

3 Question16

★ Run a neutral model simulation using the same parameters as in question 12 for

a ‘burn in’ period of 200 generations. Next record the species abundance octave vector. Then repeatedly continue the simulation from where you left off for a further 2000 generations and record the species abundance octave vector every 20 generations.
★ Produce a bar chart plot of the average species abundance distribution (as octaves). Include the code you wrote for this question in a function called ***‘question_16’*** which should require no inputs to run.





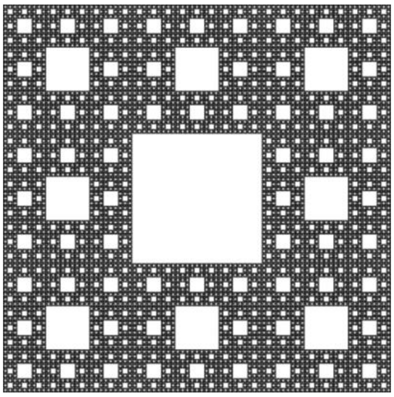
★ Does the initial condition of the system matter? Why is this? (Hint: it's OK to use a for loop here, it will also be helpful to use the ***sum_vect***, ***octaves*** and ***species_abundance*** and ***neutral_generation_speciation*** functions that you already wrote. You will also find the %% function useful). [4 marks]

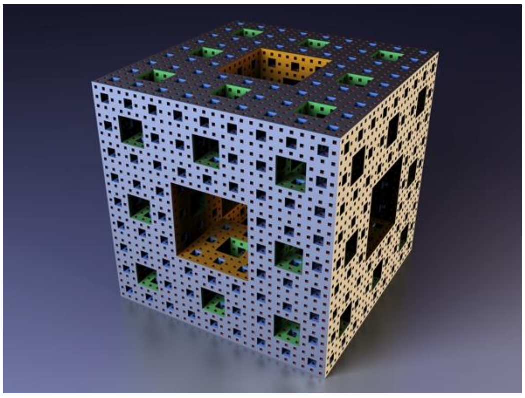
From the fiugre point of view without significant test, the final results are basically the same, so the initial state doesn't matter. The reason is that they have the same speciation rate, so if the community size is the same, ecological equilibrium will be reached. However, the further reason is still under consideration. I am still reading a paper[1] on this issue, to be continued ...

4 Question21

★ What are the fractal dimensions of these objects? Show and briefly explain your workings. [4 marks]

Hint: the object on the right looks the same from all six faces and is hollow in the *very center; it should have a dimension somewhere between 2 and 3.*





The definition of fractal dimension is: an index for characterizing fractal patterns or sets y quantifying their complexity as a ratio of the change in detail to the change in scale.[2]

Left one:
The stick size changes from 1 to 3
The area changes from 1 to 8
The dimension= log8 / log3 = 1.89

Right one:
The stick size changes from 1 to 3
The area changes from 1 to (3*9-6) =21
The dimension = log21 / log 3 = 2.27

5 Question22

★ The chaos game

a. Store the following three points that correspond to coordinates on a graph: A=(0,0), B=(3,4) and C=(4,1).
b. Initialize the point vector X to indicate the point (0,0).
c. Plot a **very small** point on the graph at X. (hint: use cex)
d. Choose one of the three points (A, B or C) at random and move X half way towards whichever of the three points you chose.

Write a loop to repeat the code of c. and d. 100 times – what do you see? Now try increasing the number of repeats to 1000 or more. The function that does this should be called ***'chaos_game'*** [8 marks]

6 Question25

★ Now copy and paste your 'elbow' function and rename it ***'spiral'***. Spiral will be an iterative function that draws a spiral. Instead of calling ***'turtle'*** twice to draw the first and second lines, spiral should call ***'turtle'*** to draw the first line and then call itself ***'spiral'*** instead of ***'turtle'*** to draw the second line.

```
> spiral(c(2,1), pi/2,1)
Error: evaluation nested too deeply: infinite recursion / options(expressions=)?
Error during wrapup: evaluation nested too deeply: infinite recursion / options(expressions=)?
> |
```

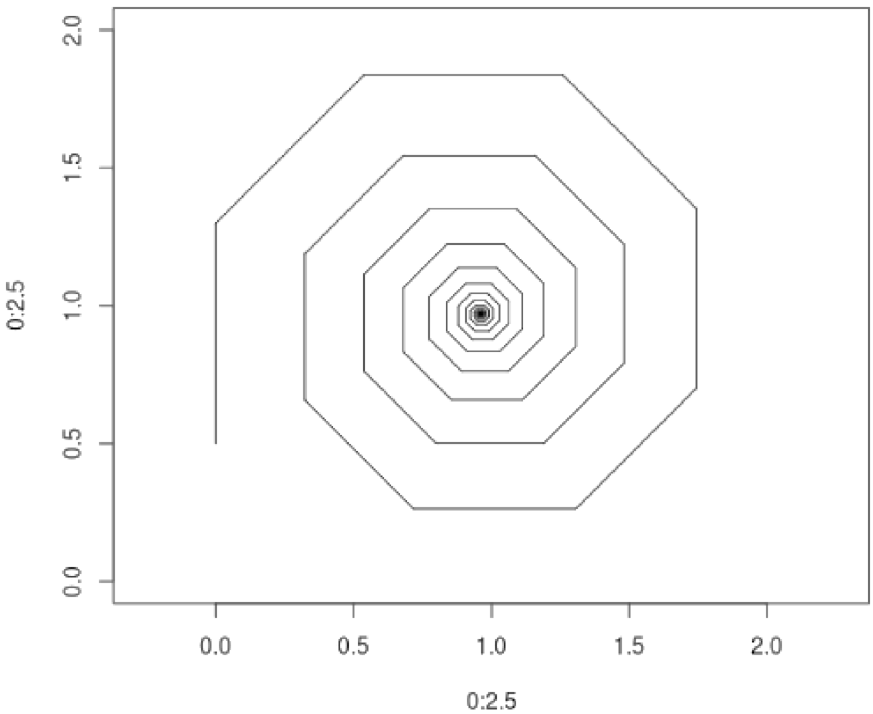
★ What happens and now and why? (Hint: if you get an error message that might be what's expected! Try and think about why you're getting it - think like a computer – run through the code you just wrote in your own head and see where it gets you) [2 marks]

The result is error messages:
Error: evaluation nested too deeply: infinite recursion / options(expressions=)?
Error during wrapup: evaluation nested too deeply: infinite recursion / options(expressions=)?

The reason is that this function will call itself indefinitely and will not stop, so the following function spiral_2 sets a limit on the length.

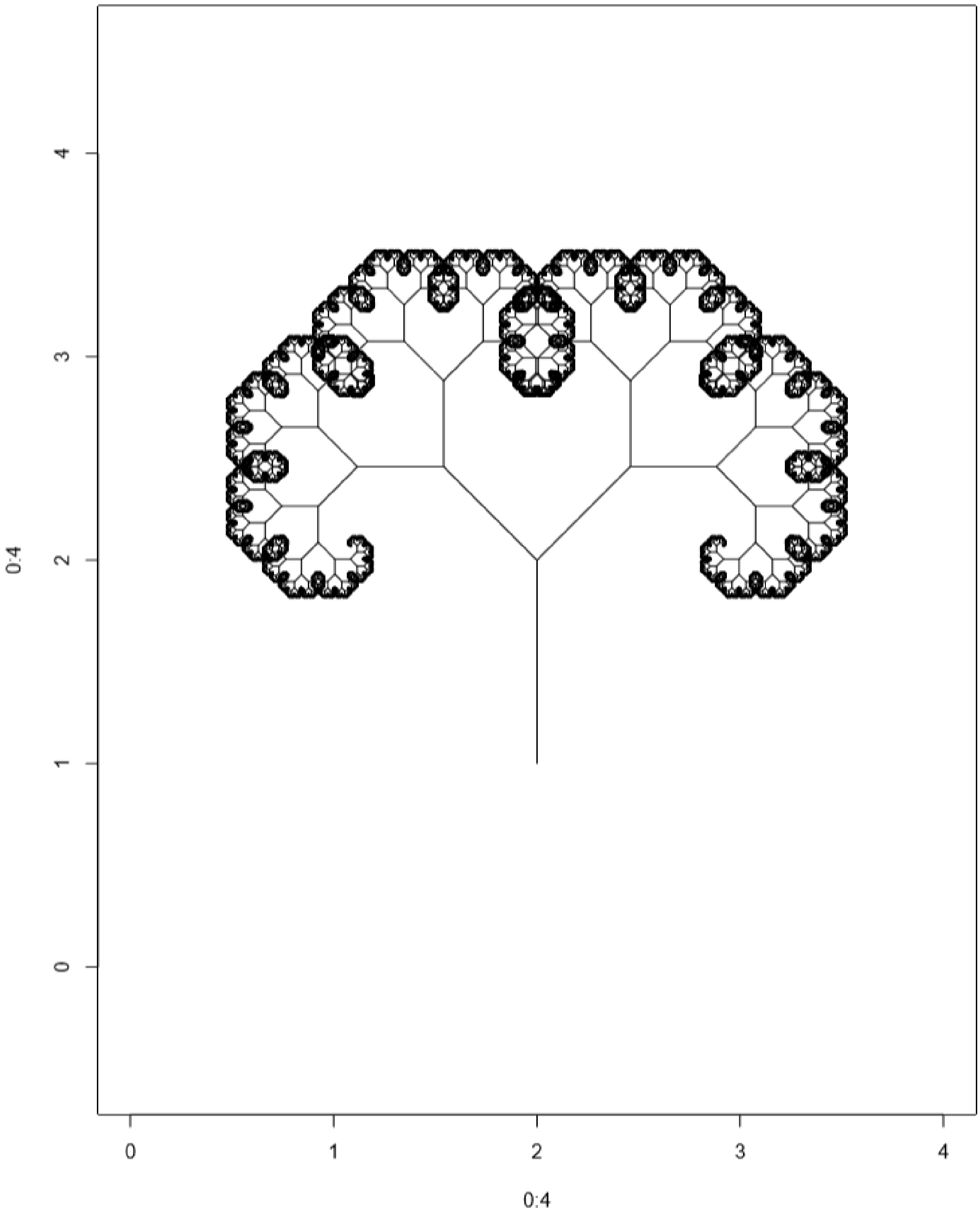
7 Question26

★ Edit the 'spiral' function calling it ***'spiral_2'***. The edit should make it so that 'spiral 2' will only act if it's called with a line length that's above a certain size (e, a variable that you can experiment with). Now your code will draw a spiral shape on the graph without crashing or giving any error messages. [3 marks for a working spiral plotting function]



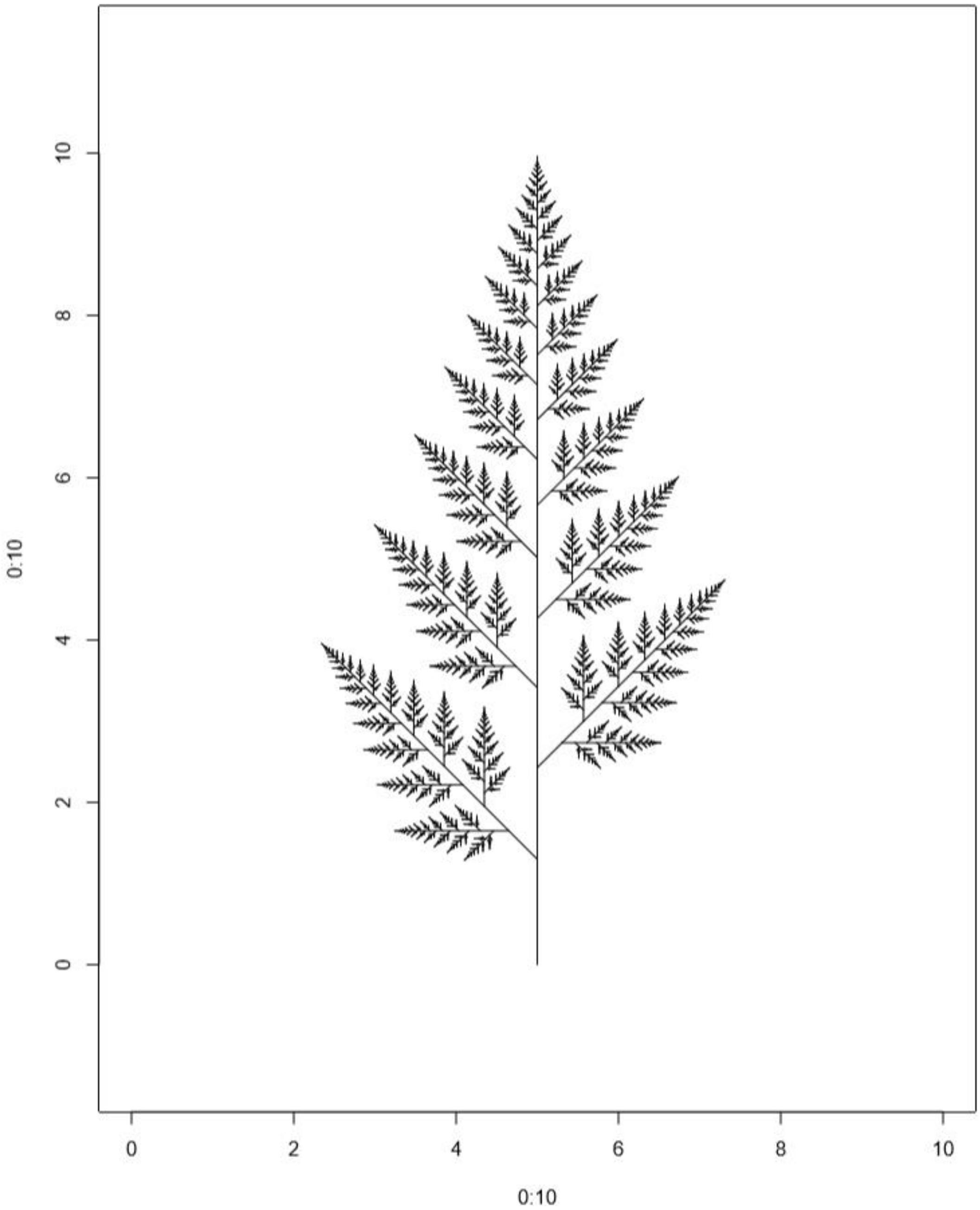
8 Question27

★ Now, copy and paste the ***'spiral_2'*** function and rename the copy ***'tree'***. Instead of having ***'tree'*** call itself only once (as ***'spiral_2'*** did), you should have it call itself twice: with directions that are 45 degrees to the right and 45 degrees to the left. Also, make the length of each subsequent call 0.65 times the length of the previous call (instead of 0.95 as it was for drawing the spiral). Don't forget that ***'tree'*** should still call ***'turtle'*** once as well as ***'tree'*** twice. You should get an attractive tree shape as your output plot. [4 marks for a working tree plotting function]



9 Question29

★ Now copy and paste the function 'fern' and rename it to ***'fern_2'***. This should have an input parameter 'dir' which will decide whether the side branch of the fern goes to the left or right (it's easiest to do this with a variable that takes the value of either -1 or +1). When calling ***'fern_2'*** iteratively from within itself allow the direction of the side branch to alternate by passing on the 'dir' variable that has been multiplied by -1 to revers the direction. You should now get an attractive fern picture. (Hint: spot the difference, look very carefully at your fern to check that it does look the same as the example in this worksheet, you will not get full marks unless they are really the same) [4 marks for a working fern plotting function]



---

[1] Wang, Shaopeng, et al. "Why abundant tropical tree species are phylogenetically old." *Proceedings of the National Academy of Sciences* 110.40 (2013): 16039-16043.
[2] https://en.wikipedia.org/wiki/Fractal_dimension