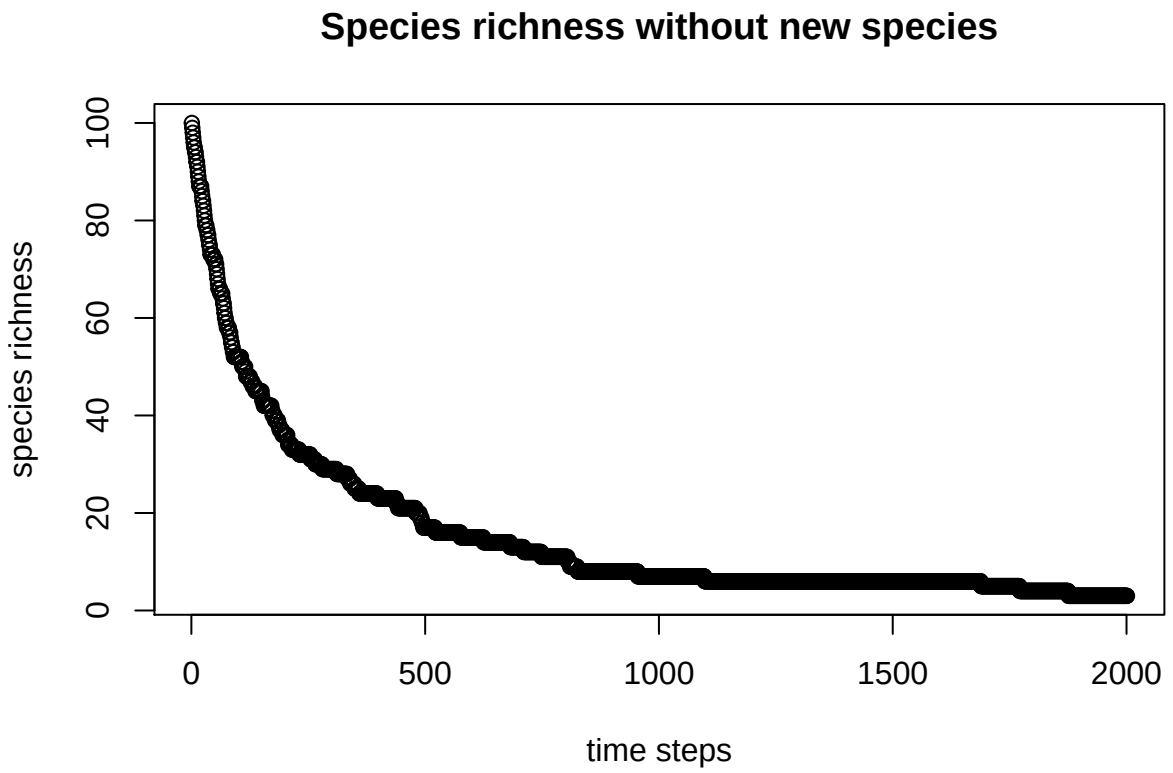
HPC Coursework

PetraGuy

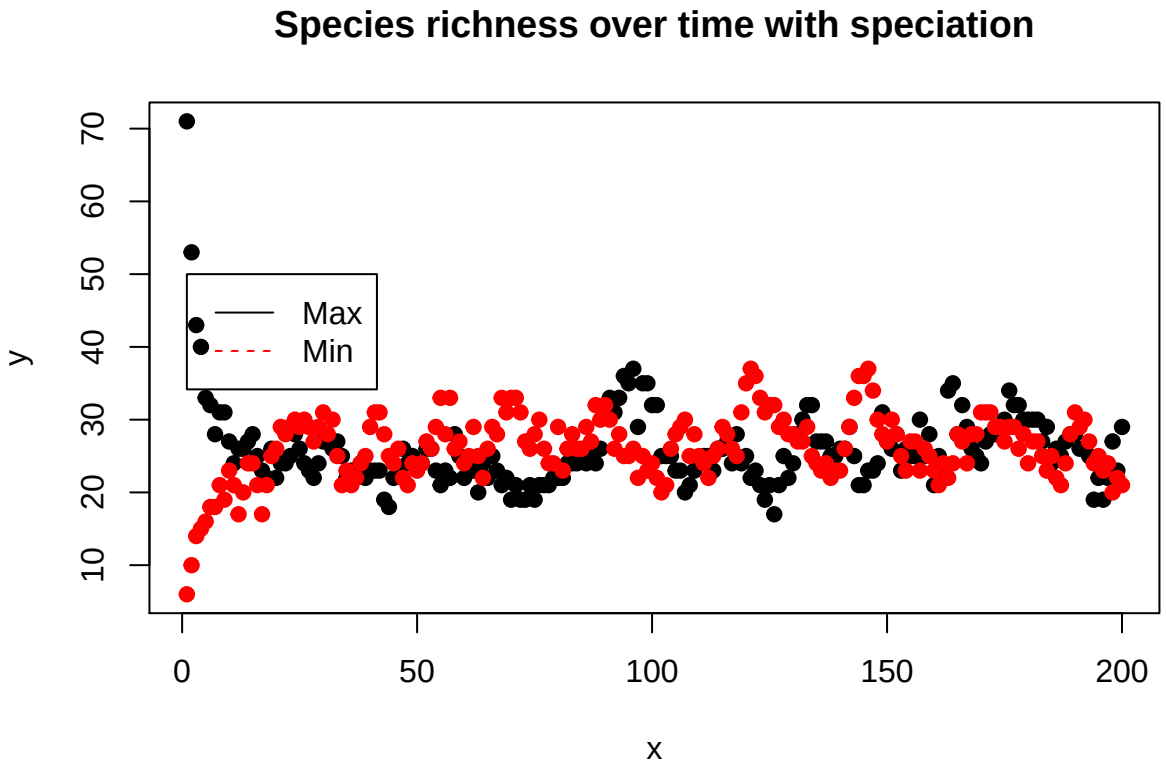
4 December 2017

```
question_8()
```



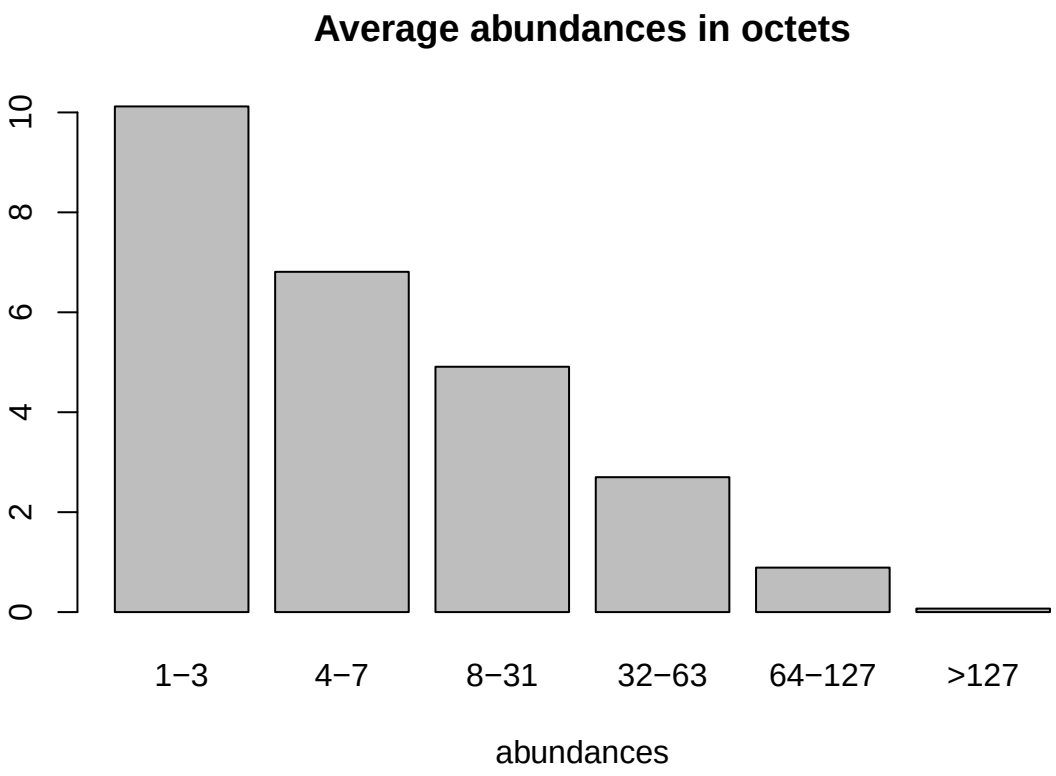
The graph plots species richness of a community starting with 100 different species over 2000 generations. In this model new species are not introduced, so each step wither maintains the richness or reduces it. For example, if you begin with a community (1,2,3,4,5), one step could change this to (2,2,3,4,5) At the next step he community will either remain unchanged, or be reduced, for example (2,2,3,3,5). Since there is no mechanism to increase species richness, eventually, given enough steps, the richness will become 1.

```
question_12()
```



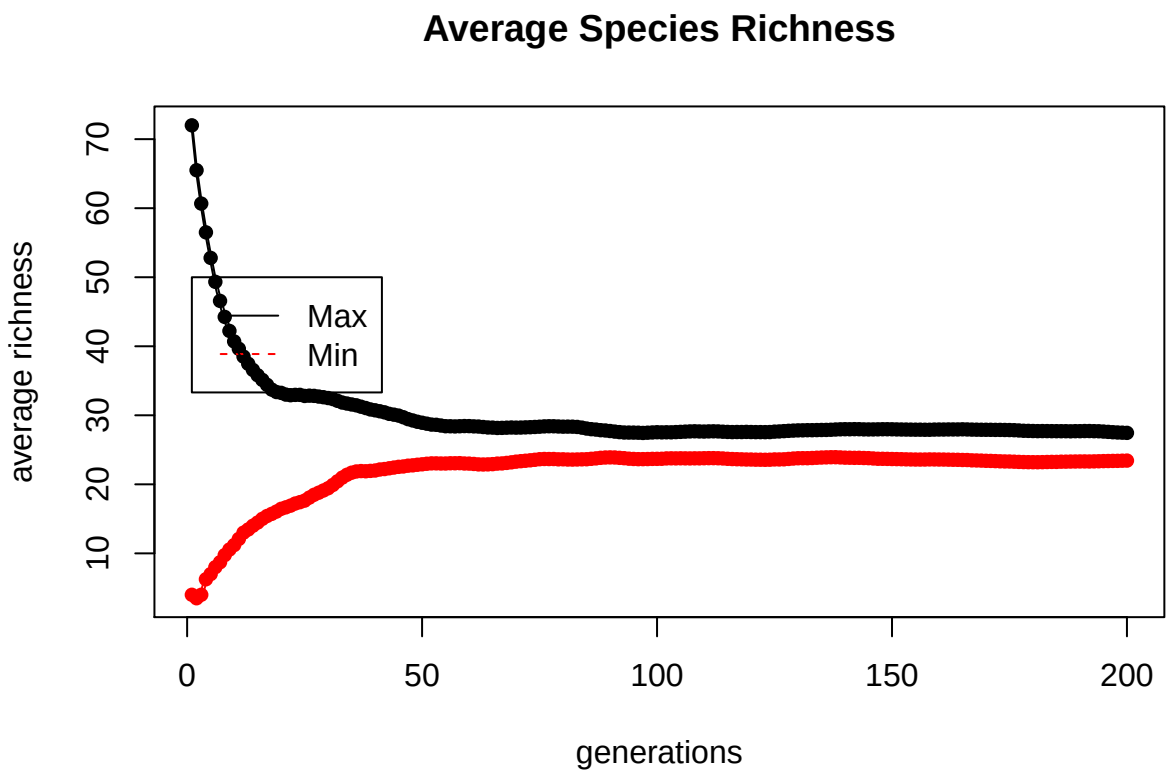
The graph shows the species richness for two communities, one with maximum richness, one with minimum, both with 100 individuals and a speciation reate of 0.1. Because the speciation rate is the same both communities they tend to the same richness after sufficient generations. A richness of around 30 after 50 generations in this case. The higher the speciation rate, the greater the richness of the final community, because the algorithm will more refrequently follow the step of generating a new individual.

```
question_16()
```



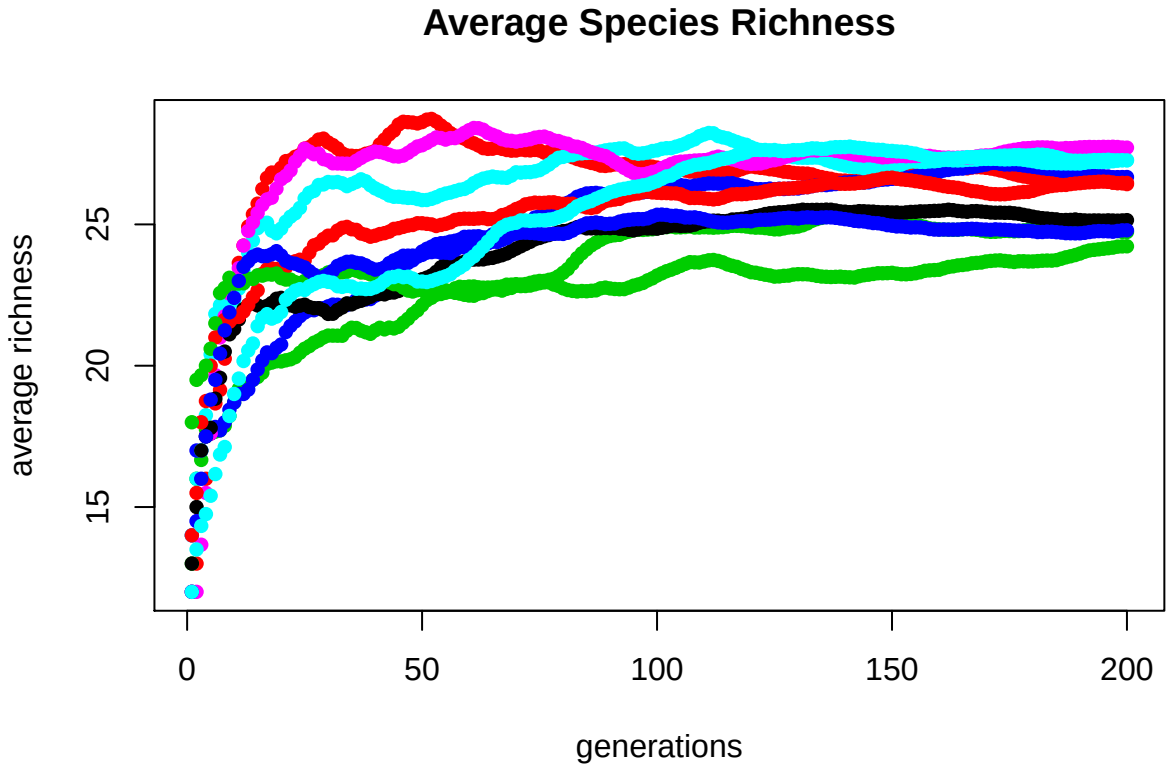
The graph shows the average abundances of the community after the burn-in of 200 generations. The distribution of abundances would change if the speciation rate changed. A higher speciation rate would result in a richer community and therefore the first octet would have a higher frequency. A low speciation rate results in a community of low richness and therefore the frequencies of the larger octets increase at the expense of the smaller.

```
challenge_A()
```



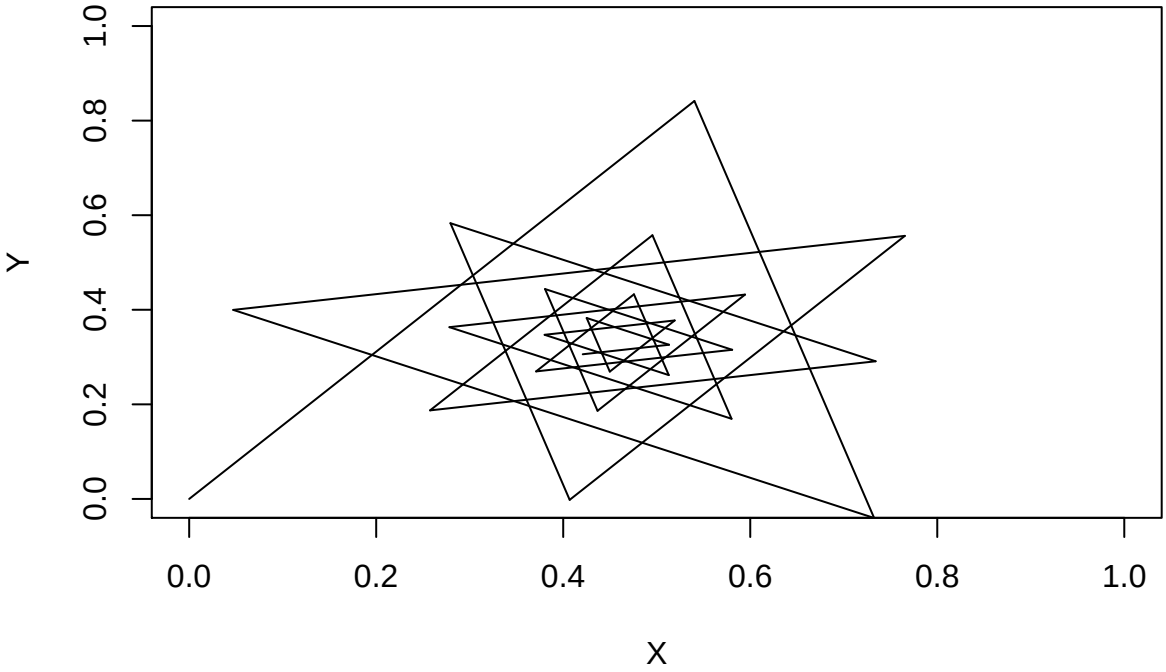
The graph is a plot of average richness values as the times increments. The confidence inervals were calcualted using the variance of the values from 50 to 100 increments. If the entire series was used, the confidence interval would be larger.

```
challenge_B()
```



```
chaos_game()
```

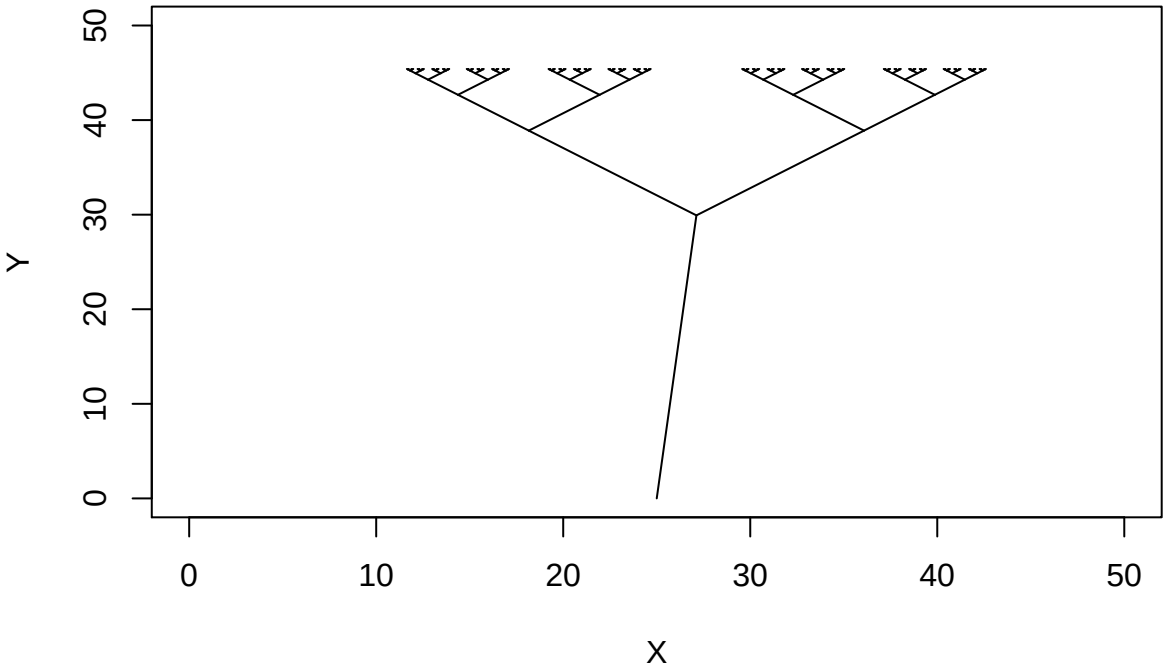
The code produces a Sierpinski Gasket type picture.



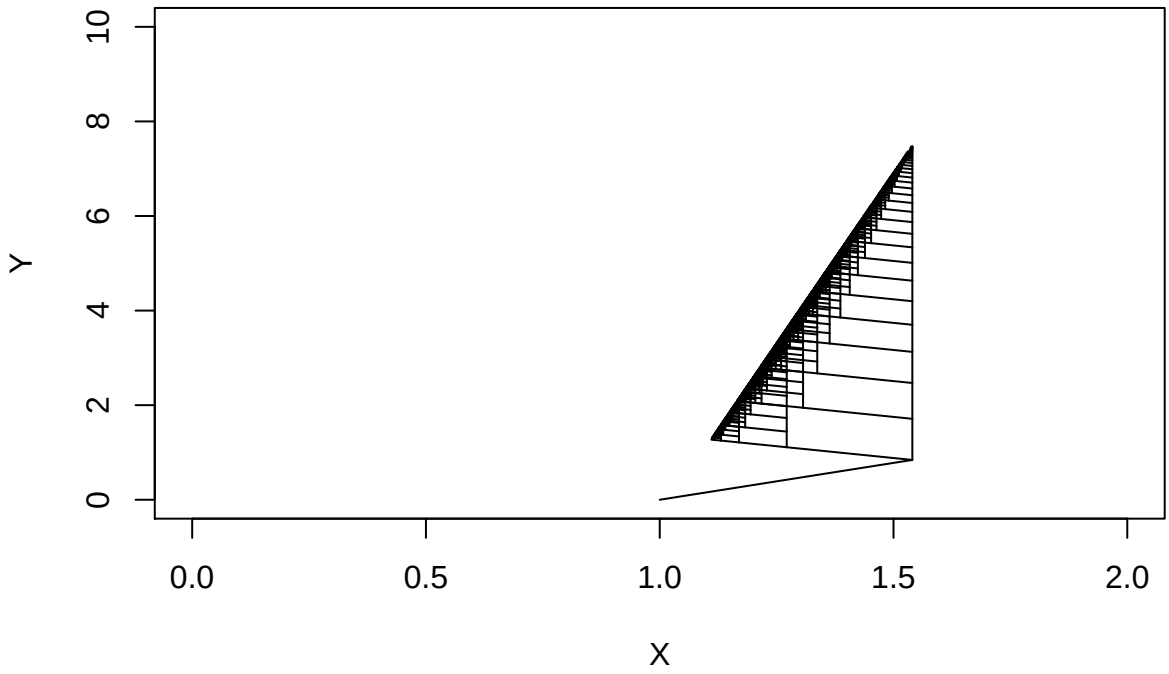
Q22

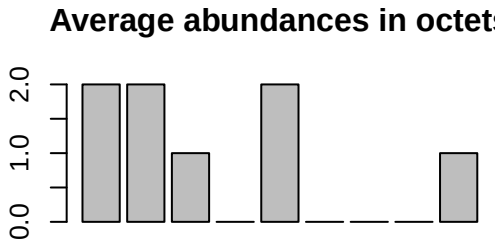
Fractals. The code produces a spiral, but includes a nested, recurssive loop because it continually calls itself. Putting an if (distance > 0.1) statement before calling spiral within spiral ensure that the programme will stop.

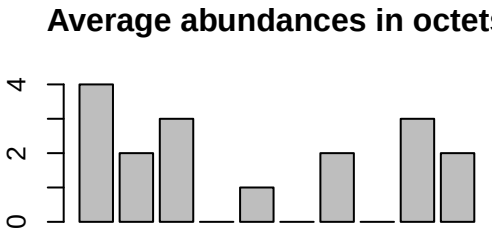
```
plot(NA, xlim=c(0,50), ylim=c(0,50), xlab="X", ylab="Y")
tree(c(25,0),30,1.5)
```

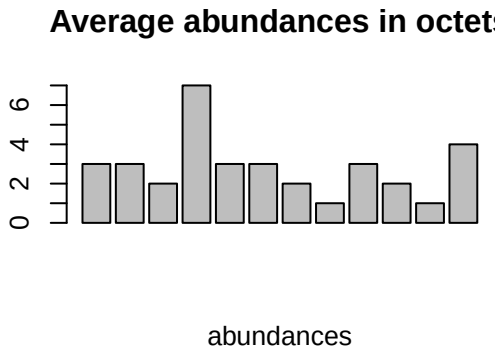


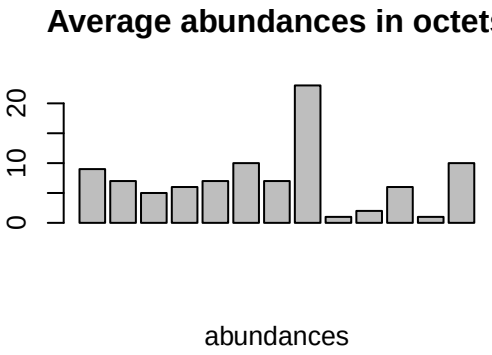

```
plot(NA, xlim=c(0,2), ylim=c(0,10), xlab="X", ylab="Y")
fern(c(1,0),1,1)
```



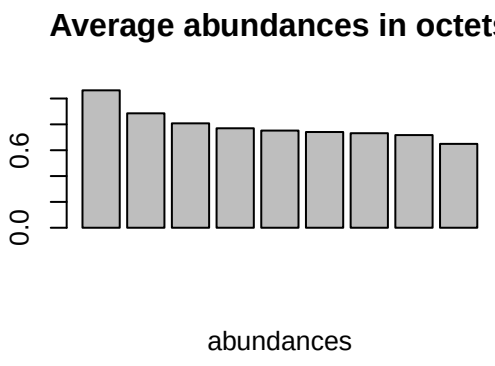


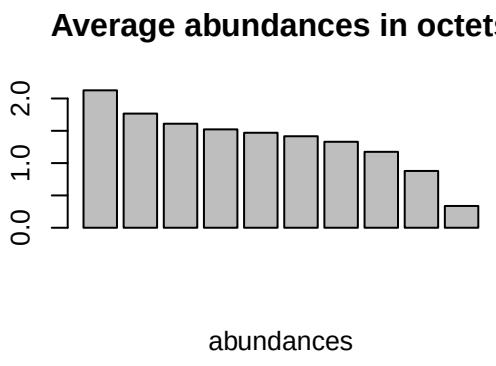


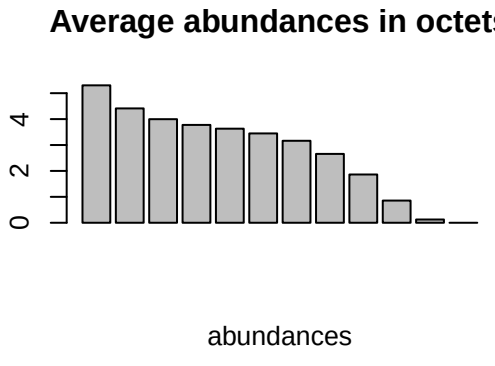


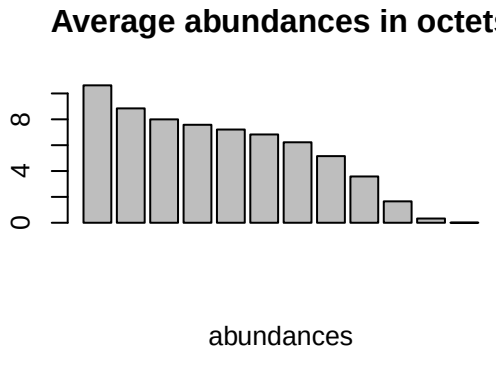


These are the octets from the coalescence simulation, which look very similar to the results of the simulation from the HPC (!). Not totally sure why one is quiker - except that in the coalescence model you generate one community, then do the size of the community calculations, so for our sizes,that's a maximum of 10,000 cyles in the loop. For the neutral model you generate a new community of size N many times during the burn in, then you continue recreating new communities for the presest time in order to take an average over all the cyles. Therefore there are many more calculations.


Average abundances in octets
0.0
0.6
abundances


Average abundances in octets
0.0
1.0
2.0
abundances


Average abundances in octets
0
2
4
abundances


Average abundances in octets
0
4
8
abundances