Chapter 13 Exercise Hints and Solutions

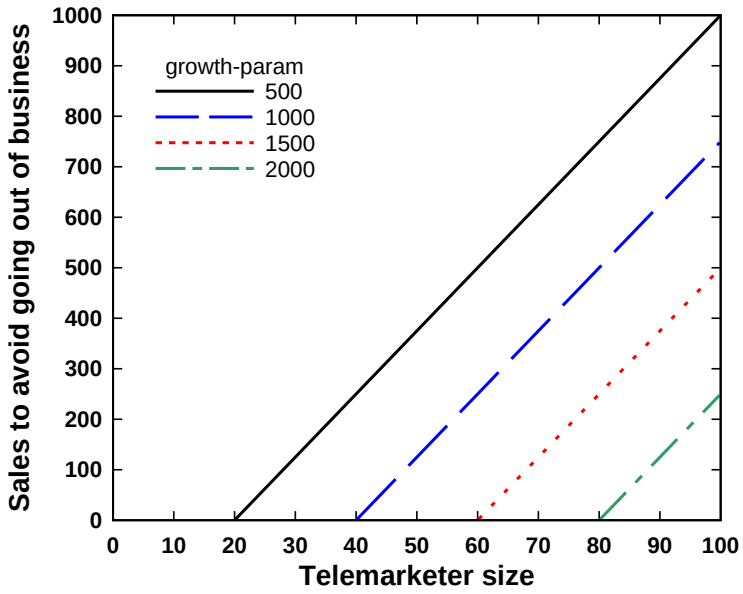
Agent-based and Individual-Based Modeling: *A Practical Introduction, 2nd Edition*

Exercise 1

The point of this exercise is to reiterate a main point of Chapter 12: the importance of understanding characteristics of a submodel before trying to understand the full model. However, the question could have been stated more specifically. In particular, we are are looking for the number of sales a telemarketer must make in one time step to keep its bank balance from going to zero on that same time step.

This question can be answered from the Telemarketer accounting submodel. A telemarketer goes out of business if its previous bank balance, plus income from sales, minus the weekly cost of staying in business, is negative. Therefore, to just avoid going out of business, bank balance plus sales income must equal the weekly cost of business. *If* the telemarketer has been growing, its previous bank balance equals the value of `growth-param`. In that case, bank balance (= `growth-param`) plus sales income (= number of sales × 2) equals weekly cost of business (= size × 25).

Reorganizing this relationship produces: $N = \frac{25S - g}{2}$ where N is the number of sales to avoid going out of business, S is telemarketer size, and g is `growth-param`. This equation produces a negative value of N when bank balance is high enough, compared to size, that the telemarketer cannot go out of business in one time step even if it makes no sales. In such cases, N can be set to zero. The equation produces the following graph for four values of `growth-param`.



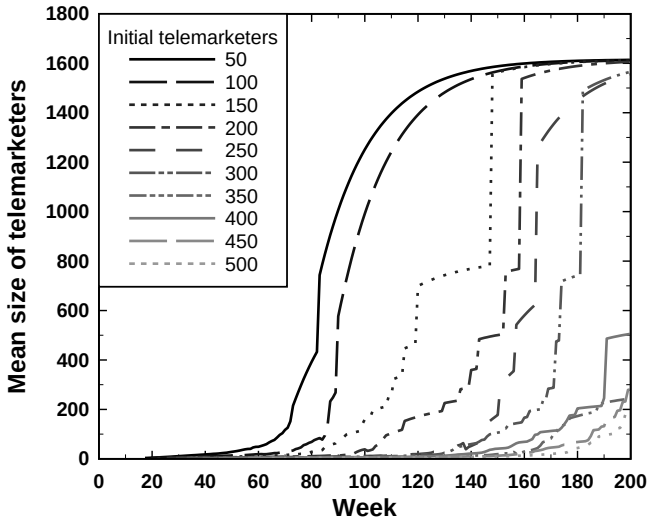
The graph should be accompanied with some interpretation. It indicates that until telemarketers reach a certain size (40, if `growth-param` is 1000), they can have at least one day of zero sales

and survive. But as they grow, they must make an increasing number of sales per day or go out of business immediately.

(If the bank balance has already been depleted, then the minimum sales to stay in business is just enough for income to equal the weekly cost of business: $N = 25S$, independent of `growth-param`.)

Exercise 2

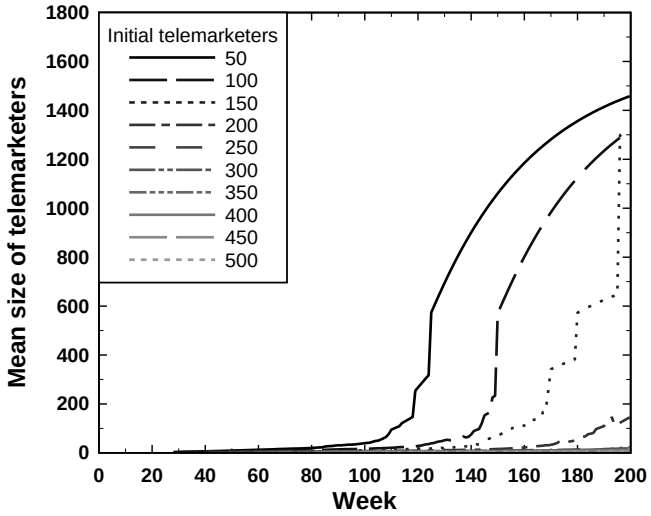
The question is whether the reason that telemarketers stay in business longer when there are more of them is that they go out of business because they grow too quickly. This question is based on two presumptions. The first presumption is that when we start with more telemarketers, they grow more slowly. It is best to start by testing this presumption, e.g., by creating a graph similar to the left panel of Figure 13-2 except showing the mean size of telemarketers instead of the number of them. This analysis confirms that telemarketers are smaller when there are more of them:



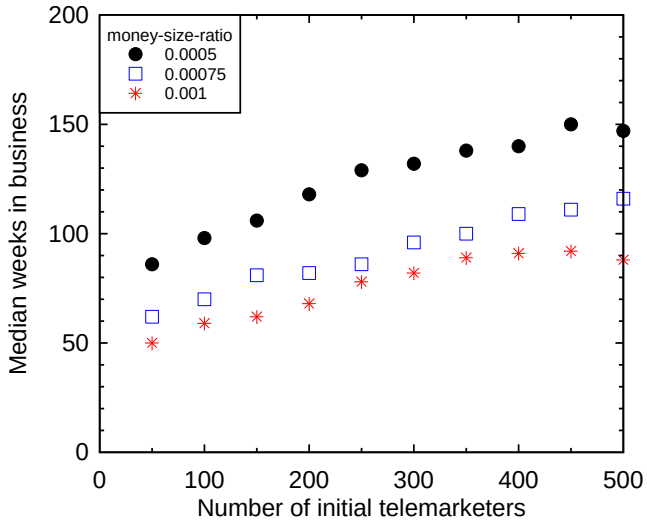
The second presumption is that telemarketers stay in business longer when they grow less, because the smaller they are, the less competition there is for the limited number of customers and the fewer sales are needed to meet the weekly cost of business. This presumption can be tested by conducting experiments to see how model results change in response to the parameters that control growth: `growth-param` and the parameter `money-size-ratio` that converts bank balance to increase in size. The most useful to analyze here is `money-size-ratio` because it controls only the rate of growth, whereas `growth-param` controls both rate of growth and the size of the bank balance. (The bank balance could be important because it could provide a buffer against short periods of losses, but the pattern we are trying to explain here does not involve variation in bank balance.)

We can vary `money-size-ratio` to see its effect on both telemarketer size and median time in business. The rate at which telemarketers grow, when they have excess bank balance, is

proportional to this parameter. With `money-size-ratio` set to 0.0005 (half its standard value), the mean size results confirm that telemarketers are much smaller:



We can also produce results such as those in the right panel of Fig. 13-2 with several values of `money-size-ratio`. (For these graphs, median lifespan of telemarketers is calculated simply by finding the week at which number of telemarketers still in business is half the initial number.) These results confirm that telemarketers would stay in business longer if they grew more slowly.



Provided with the instructor materials is a version of the Telemarketer model code with BehaviorSpace experiments to produce the above results (`Telemarketer_Ch13-Ex2_2ndEd.nlogo`), and an Excel workbook that calculates median time in business (`Ch13-Ex2_MoneySizeRatioEffect_2ndEd.xlsx`).

Exercise 3

The key factor for this exercise is for students to start by saying clearly what they are trying to get the telemarketers to be more successful at: staying in business, making more total sales over 200 time steps, getting bigger, etc. This is especially important now that we know that growing in size conflicts with staying in business.

Exercise 4

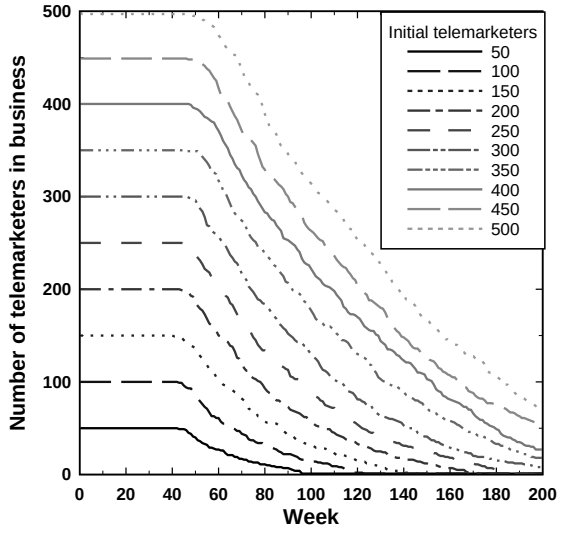
This change in the model is very simple to implement; just remove the restriction that potential customers must be within a radius of the telemarketer. For example, change this:

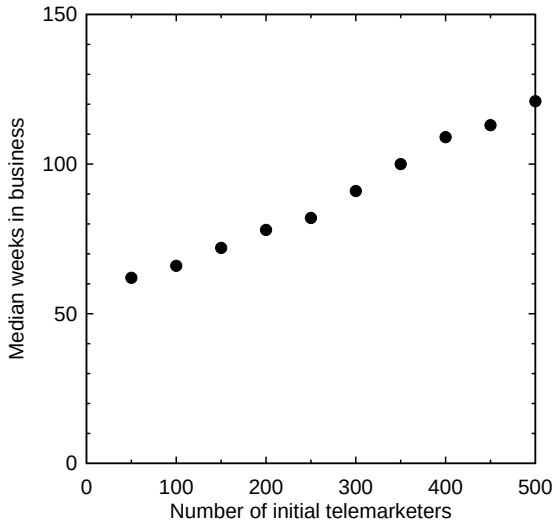
```
to do-sales

   let max-calls size * 100
   let potential-customers patches in-radius (10 * size ^ 0.5)
   let customers-called potential-customers
   if max-calls < count potential-customers
    [set customers-called n-of max-calls potential-customers]
```

To this:

```
to do-sales

   let max-calls size * 100
   let potential-customers patches  ; The only change is here
   let customers-called potential-customers
   if max-calls < count potential-customers
    [set customers-called n-of max-calls potential-customers]
```

With this change, the following graphs show that more telemarketers last longer, and no longer do some some telemarketers fail immediately when the initial number of them is above 300.

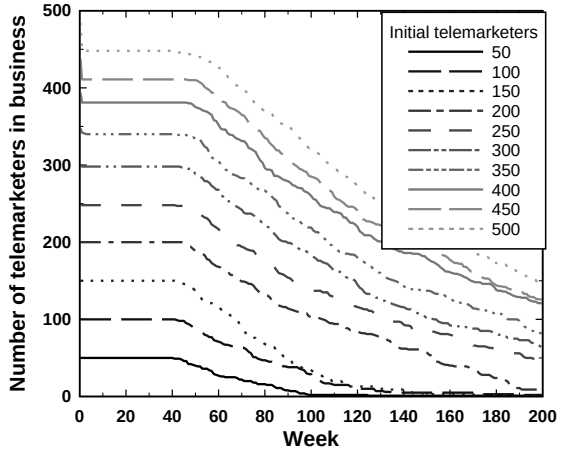


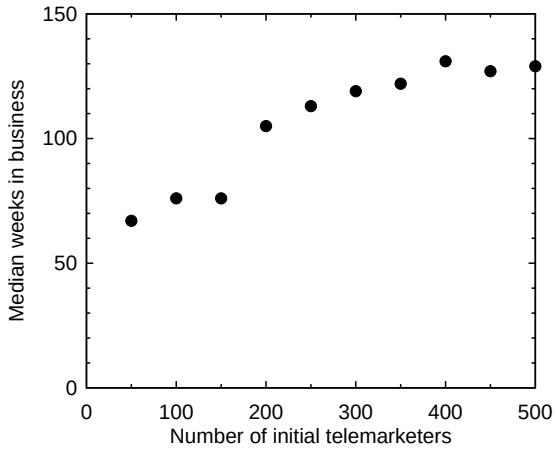


Exercise 5

A NetLogo program implementing the model version of Section 13.5 is provided (`Telemarketer-Mergers-Links_Ch13-Ex5_2ndEd.nlogo`). It includes test output written to the Command Center in several places; these outputs are commented out. (Un-comment the test output for one procedure at a time or they will be hard to interpret.)

Results (below) show that the mergers do in fact substantially increase the persistence of telemarketers. (However, see the instructor materials for Exercise 2 of Chapter 14!)

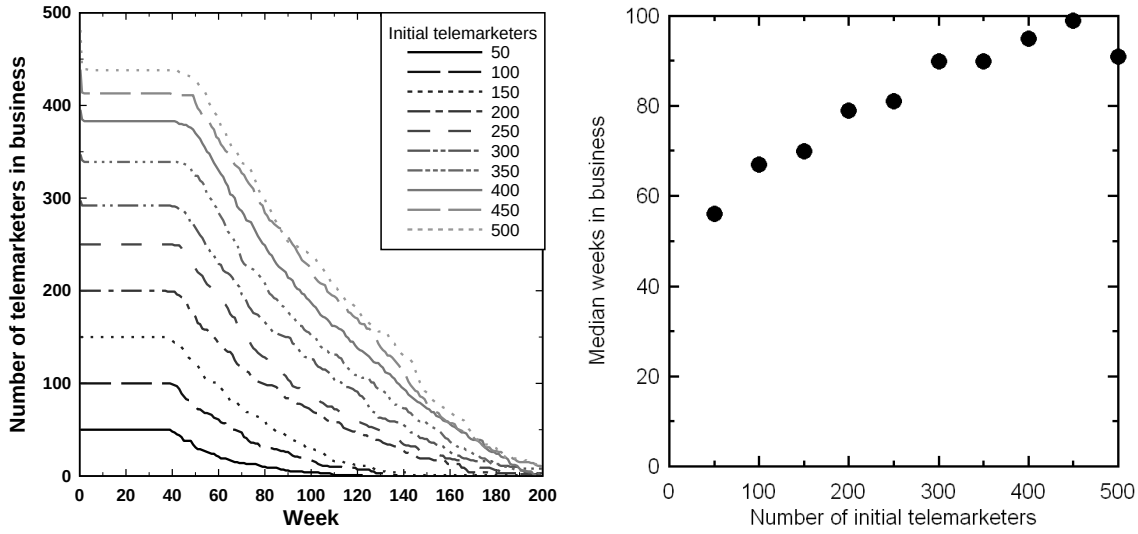




Exercise 6

The model version from Section 13.6 is available as: `Telemarketer-Memory_Ch13-Ex6_2ndEd.nlogo`.

Results comparable to those of Fig.13-2 show that the change has little clear effect over 200 weeks (graphs below). However, you should suspect that when customers now have a limited tolerance for each telemarketer, it will be difficult for the last big telemarketers to stay in business. In fact, if you run this version of the model for more than 200 ticks, all the telemarketers fail because there are not enough customers still accepting their calls.



Exercise 7

Both of these do-not-call list approaches are quite easy to implement. They require making an assumption about whether customers on the list are or are not among the maximum calls each telemarketer can make per week (neither way is clearly right or wrong). These modifications also require making up some assumptions about how and when customers decide to put themselves on the lists.

For the "national" do-not-call list, an important trick is to implement this "list" not as a NetLogo list or agentset, but simply as another patch color. Instead of patches being (for example) green (ready to take a call) or brown (already called this tick), they can also change themselves permanently to red, meaning they never accept a call any more. An example implementation is: `Telemarketer_Ch13-Ex7-National_2ndEd.nlogo`.

When each telemarketer has its own separate do-not-call list, this list can be a turtle variable. It is equally easy to make the list either a NetLogo list or an agentset. The primitive `member?` is important. The example implementation `Telemarketer_Ch13-Ex7-Separate_2ndEd.nlogo` uses NetLogo list variables. See its procedure `do-sales` for an example statement that excludes members of the telemarketer's do-not-call list from its agentset of potential customers; the same statement works whether the do-not-call list is programmed as an agentset instead of a list.

Exercise 8

The only change needed to limit the number of calls the customers remember is to limit the length of their caller memory list to five. Add new calls to the beginning of the list and, if the list is longer than five, remove the last member. The primitive `but-last` is helpful.

```
; Then remember this call by adding it to the memory
set caller-memory-list fput a-marketer caller-memory-list
; Limit memory to five newest calls
if length caller-memory-list > 5
[ set caller-memory-list but-last caller-memory-list ]
```