Chapter 10 Exercise Hints and Solutions

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

Exercise 1

This exercise (or quiz questions similar to it) could be conducted either with or without the ability to use NetLogo to check the answers. It is common in programming classes to ask students to answer these kinds of questions on paper, without access to a computer, to see if they really know the syntax. However, answers to these kinds of questions can usually be tested by entering them into the code for a model (here, the Business Investor model version that uses links would be handy) and using the "check" button; or by typing them into the Command Center or the mini-command center in an Agent Monitor. Unless students are explicitly told not to use NetLogo, the answers should all be checked and mistake-free; it is also important to learn how to check code as it is written.

To check in the Command Center, replace the `let my-variable` with `show`. For example, for the third question enter this in the mini-command center of a turtle Agent Monitor: `show max [size] of turtles` and see whether an error statement or a plausible result appears in the main Command Center. (The `show` primitive is not needed in the Command Center: if you just enter `max [size] of turtles` NetLogo assumes you want to `show` the result.)

Solutions:

- `let my-size size`
  (*not* `let my-size [size] of self` or `let my-size [size] of myself`)
- `let current-patch-label plabel`
  (*not* `let current-patch-label [plabel] of patch-here` even though it works)
- `let biggest-turtle-size max [size] of turtles*`
- `let biggest-other-turtle-size max [size] of other turtles*`
- `let neighbor-color [color] of my-nearest-neighbor`
- `let a-linked-turtle one-of my-out-links`
- `let all-linked-turtles my-out-links`
- `let all-linked-turtles my-links`
- `let all-linked-turtles my-links with [ not is-directed-link? self ]`
  (The answer to this is in the NetLogo dictionary entry for `my-links`)
- `let num-turtles-up-and-over count turtles-on patch-at 1 1`
  (or, for the creative:
  `let num-turtles-up-and-over count turtles-on patch-at-heading-and-distance 45 1.41421`)
- `let red-wealth-list [wealth] of turtles with [pcolor = red]`
- `let wealth-SD standard-deviation [wealth] of turtles`

*In the first version of this book these were minimum instead of maximum sizes, so be suspicious if students submit `smallest-turtle-size min [size]` etc.

Exercise 2

If you try this statement in the mini-command center of a turtle Agent Monitor, it produces what looks like a plausible answer. (Enter the command `show patches in-radius 2 with [not any? other turtles-here]` and the result is something like: `(turtle 3): (agentset, 9 patches)`. But if you check more carefully, you will find that this command always includes *all* the patches within the sensing radius, even if some have turtles on them.

The problem is that `other` is in patch context: it is part of the `patches with ... []` statement, so everything inside the brackets is in patch context. But turtles-here always return a set of turtles, so `other turtles-here` means turtles here that are not this patch. No turtle is ever a patch, so `not any? other turtles-here` will always be true.

To obtain the desired agentset, use `patch-set` as illustrated in the Programming Note that starts on page 135.

Exercise 3

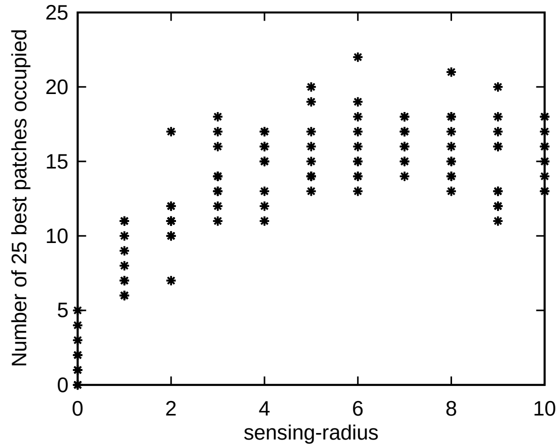
An implementation of the Business Investor model as described in Section 10.4.2 is provided with the instructor materials (`BusinessInvestors_Sect10-4-2_2ndEd.nlogo`).

The question here is why the standard deviation in investor wealth does not increase as rapidly as mean wealth as the radius over which investors see and choose among alternative investments increases. In other words, why does the increasing ability to sense business opportunities produce a wealth distribution that is a smaller fraction of total wealth?

The obvious hypothesis is that increasing the sensing radius lets more of the investors find the best business opportunities. One simple way to test this hypothesis is to see how many of the "best" patches are occupied by investors at different values of sensing radius. However, we must decide how to define "best" patches; obviously these are the ones providing highest value of the expected future wealth utility function, but the utility function depends on current wealth. We can, for example, define the best patches as the 25 providing highest utility for the wealthiest turtle. Adding the following reporter to the BehaviorSpace experiment used to produce Figure 10.1 (top) produces output saying how many of these 25 "best" patches are occupied by investors:

count (max-n-of 25 patches [utility-for (max-one-of turtles [wealth])]) with [any? turtles-here]

The result is a graph like this, with separate points for each of 10 replicates per sensing radius scenario:



The graph indicates that investors do indeed do much better at finding the best opportunities as the sensing radius increases.

Another way to investigate this question is to have each investor calculate and report the highest utility it could receive at any available patch, not just those it is allowed to move to. This is easily done by adding a turtle variable `best-utility`. This variable can be updated in the accounting procedure by adding the last two of these statements:

```
; For output, update the utility of the investor
set current-utility utility-for self

let all-available-patches (patch-set (patches with [not any? turtles-here]) patch-here)
set best-utility [utility-for myself] of max-one-of all-available-patches [utility-for myself]
```

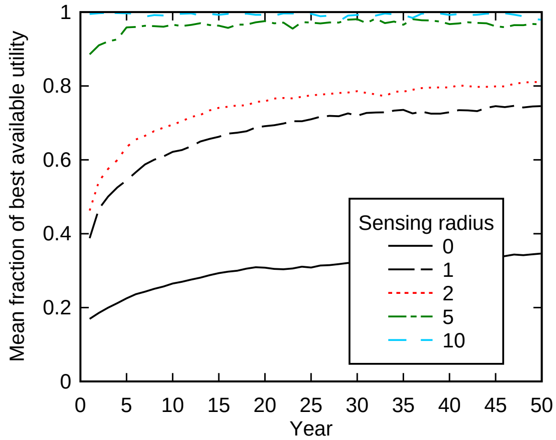
Then we can add reporters to the BehaviorSpace experiment used to create Figure 10-1 (bottom) to obtain two kinds of output. First is the mean ratio of actual utility to best-available utility, indicating how much their limited sensing ability is affecting investors:

```
mean [current-utility / best-utility] of turtles
```

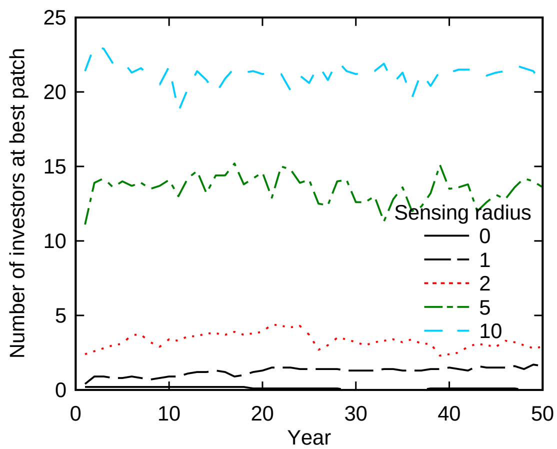
A second reporter tells us how many of the 25 investors are at their optimal patch:

```
count turtles with [current-utility = best-utility]
```

The result of the first reporter is a time series of the average fraction of best-available utility actually obtained by investors:



The second reporter produces a time series of how many investors are at the best-possible patch:



The graphs indicate several mechanisms causing higher wealth but not higher standard deviation in wealth at higher values of sensing radius. Higher sensing radius values result in investors obtaining utilities much closer to the best available, and taking less time to find high-utility patches. The last graph above indicates that even with sensing radiuses as high as 5 patches, only about half of investors find the best-available patches. More striking is that the number of investors at the best patch does not increase over time (after the first few time steps), at any sensing radius. Investors appear to get stuck in "local optima" instead of moving longer distances to find better opportunities. This result is confirmed by the View, which indicates that few investors move far over a full simulation.

The last graph suggests why the standard deviation in wealth does not increase much with sensing radius above 2 patches. As more investors find the best-possible patch, the upper limit on wealth is capped (there is no way for many of the investors to increase wealth because they are already in the patch providing best utility). As the sensing radius increases, fewer investors are poor because they were trapped in local optima, but the richest do not get richer because they are already in the globally optimal location.

Exercise 4

An implementation of the Business Investor model of Section 10.4.3 is also provided (`BusinessInvestors_Sect10-4-3_2ndEd.nlogo`).

The mean wealth of 25 investors increases *slightly* as the number of links increases from 0 to 2, but then increases no further for more links. Students may think they see patterns in how wealth changes with number of links higher than 2, but not if they (a) remember to scale the Y axis on their graph so it starts at zero (like Figure 10.1) and (b) include replicates. The following graph represents a good analysis.

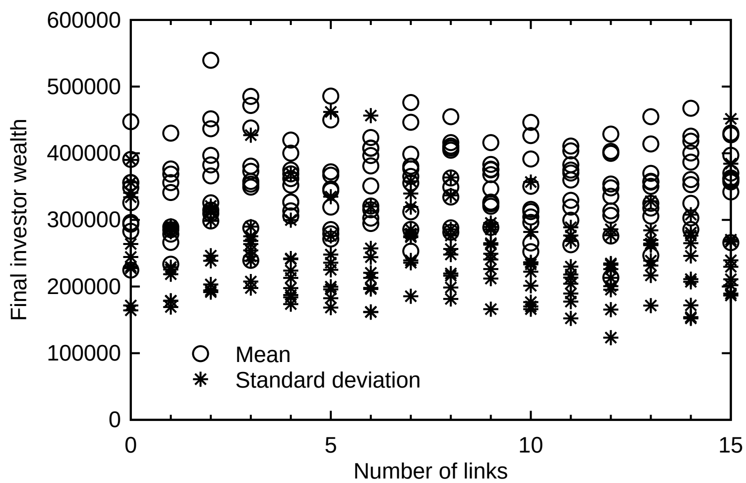


Figure 1. Results of simulation experiment varying number of links, using 10 replicates. No trend in wealth with number of links is apparent.

Why do links to other investors provide little benefit for finding better business opportunities, compared to a larger sensing radius? The simplest answer is that if good opportunities occurred near the investors you are linked to, those investors would have already taken them.

Exercise 5

The first challenge in this exercise is changing the NetLogo code so that the World's size is set by a parameter that BehaviorSpace can control. Students should readily find the primitive `resize-world`. Tricks to using it in `setup` so BehaviorSpace can set the world size include:

- Creating a slider with a global variable equal to the number of patches wide and high that the World will be; we call the variable `patches-per-side`.
- Changing the World settings, on the Interface, so that the origin is in a corner. We used the lower left corner.
- Reading the dictionary entry for carefully enough to see that it destroys all patches and turtles, so it must go before almost anything else in `setup`.

We started setup with this code:

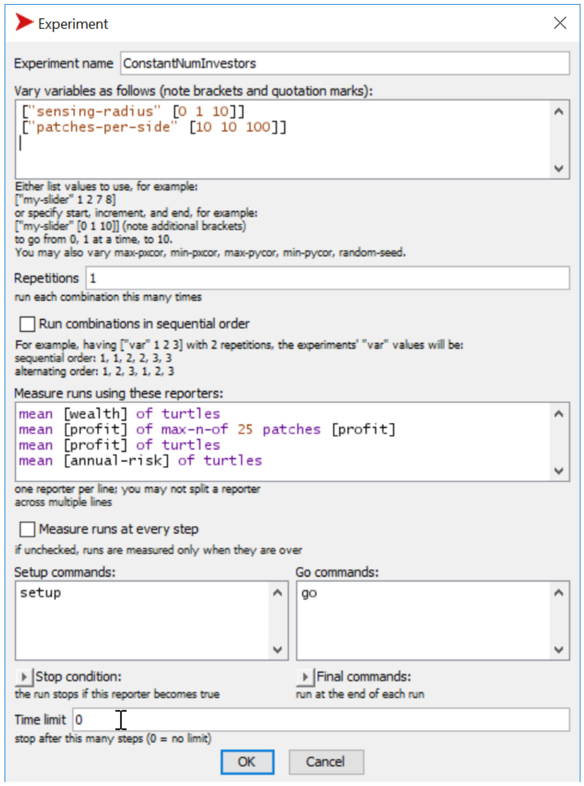
```
to setup

  clear-all
  reset-ticks

  ; Set world size
  resize-world 0 (patches-per-side - 1) 0 (patches-per-side - 1)
```

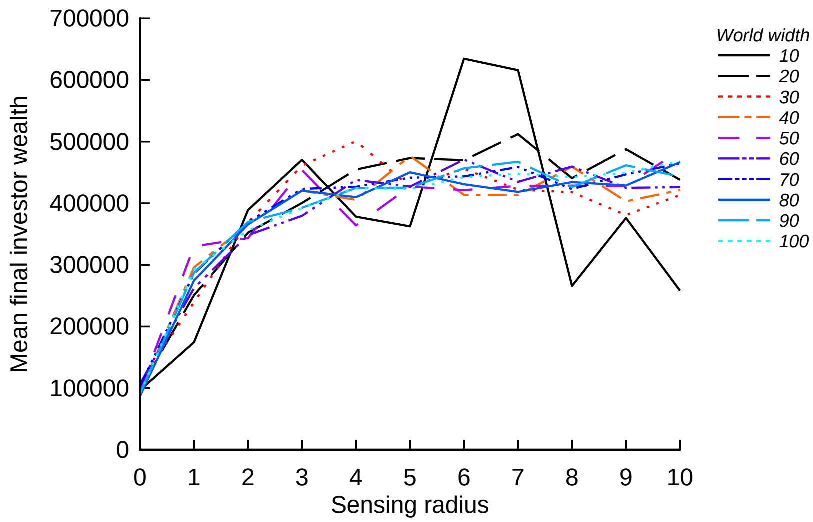
The second challenge is setting up and running BehaviorSpace experiment. Eventually, students should learn that they need several different kinds of output to understand their results. We used an experiment like this. Note that the outputs include:

- The mean final wealth of investors, the primary result we are trying to observe and explain;
- The mean profit of the 25 highest-profit patches, to see if the profit available to the 25 investors goes up with the space size; and
- The mean profit and risk of the patches actually used by the investors.



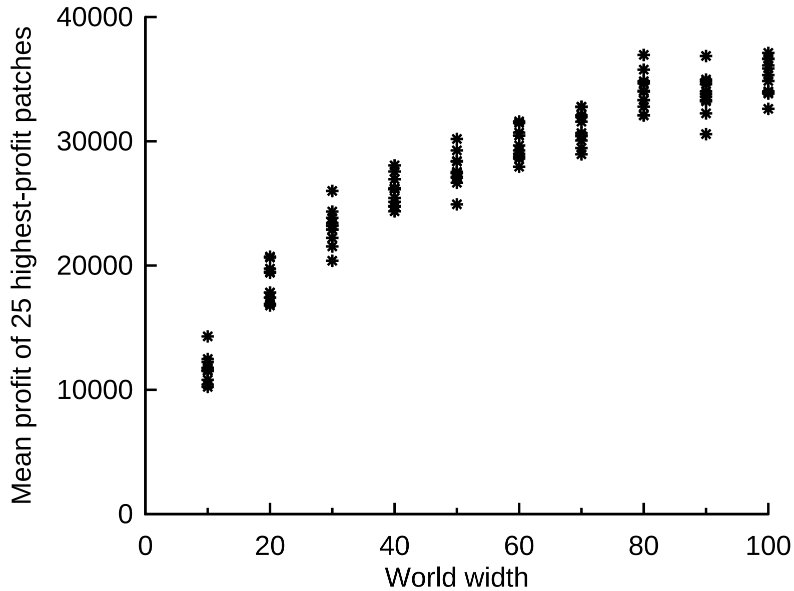
Students should produce graphs similar to the following.

First, the primary result, how the response of mean final wealth to sensing radius changes with world size:

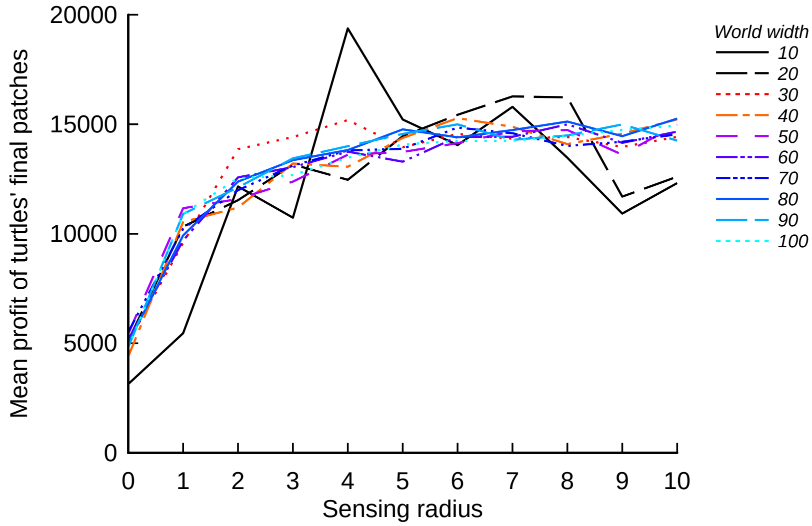


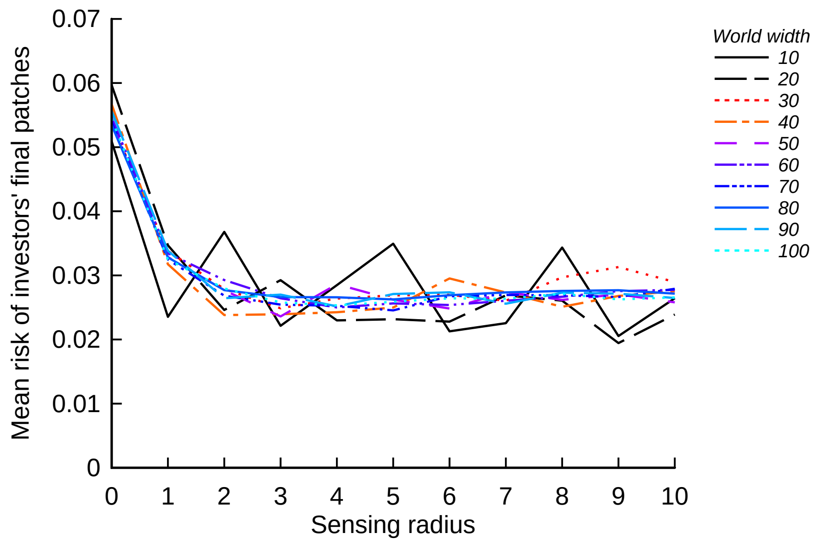
It is clear from this figure that world size has little effect except that results are more variable when world size is low.

Plotting the mean profit offered by the 25 highest-profit patches shows that increasing the space size does dramatically increase the profit available to the investors. (A similar result is obtain by plotting just the mean profit of all patches.)



So if higher-profit patches are available when the world is bigger, why does investor wealth not increase with world size? Let’s now look at the patches actually used by investors at their final location, and the profit and risk offered there:





These final plots show why investor wealth does not vary with space size: although bigger spaces offer patches with higher profit, the investors are not seeking just higher profit. The tradeoff they make between profit and risk, and its greater incentive to reducing risk over time, causes them to select patches with profit well below the highest available (compare the above plot of mean profit of patches used to the previous plot of highest-available profit).