Chapter 15 Exercise Hints and Solutions

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

Exercise 1

An example NetLogo code that writes the random draws for exercises 1 and 2 is provided: `Ch15-Ex1-2_RandDraws_2ndEd.nlogo`.

The primitive `random-float` is a uniform real number (or "continuous uniform") distribution. The characteristics of such distributions are readily found in statistical references (including Wikipedia). The mean is $\frac{(a+b)}{2}$ where *a* and *b* are the lower and upper limits (here, 0 and 100). The variance is $\frac{(b-a)^2}{12}$ so the standard deviation (square root of variance) is $\frac{b-a}{\sqrt{12}}$. Hence, the theoretical mean for this exercise is 50 and the standard deviation is 28.9. The calculated mean and standard deviation over 1000 draws from `random-float` should be close to these theoretical values.

Students should create the equivalent of a histogram for each set of 1000 random draws. They could, for example, use Excel's "histogram" tool under Data > Data Analysis (which requires the optional Data Analysis package to be installed) to produce results like these:

	Run 1	Run 2	Run 3
Bin	*Frequency*	*Frequency*	*Frequency*
10	94	113	110
20	103	87	102
30	109	101	91
40	86	84	102
50	101	105	103
60	100	99	94
70	92	106	113
80	108	102	94
90	117	100	91
100	90	103	100

Exercise 2

The Poisson distribution should produce results similar to these with a parameter of 0.8:

	Run 1	Run 2	Run 3
Bin	*Frequency*	*Frequency*	*Frequency*
0	467	446	462
1	313	363	352
2	164	153	142
3	45	30	33
4	8	7	9
5	1	1	2
6	2	0	0
7	0	0	0

The theoretical mean of a Poisson distribution is the distribution's parameter, 0.8. The theoretical variance is equal to the mean, so the standard deviation over the 1000 random values for this exercise should be near $\sqrt{0.8}$.

Exercise 3

Figure 12.1 shows the mean and standard deviation over 10 replicates of mean final investor wealth, for time horizon values from 1 to 25 years. For comparison, this figure is reproduced below (left). The standard deviations are big enough to indicate that, except for an overall trend of decreasing final wealth as time horizon exceeds 10 or so years, variation among time horizon scenarios is due to randomness.

For the second part of the exercise, there are several ways to generate the same initial conditions, using `random-seed` or `with-local-randomness` as described in the chapter (as in the provided file `BusinessInvestors_Ch15-Ex3-StaticSetup.nlogo`). With this change, standard deviations in mean final wealth are lower by about 20% (following graph), indicating that initialization causes some but not most of the randomness in results.

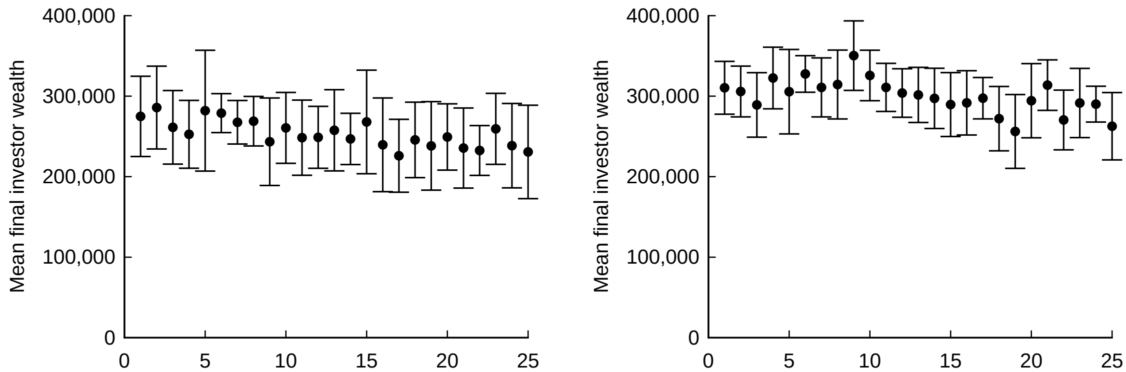

Figure 1. Mean and standard deviation over 10 replicate runs in mean investor wealth, for decision time horizons from 1 to 25 years. Left: standard model with stochastic variation among replicates in initialization. Right: Model with the no variation in initialization among runs.

Exercise 4

The program can be as simple as:

```
to go
  reset-ticks
  while [random-normal 5 1.5 > 0.0] [tick]
end
```

This code will, when executed *once* (not with a 'forever' button), make NetLogo add ticks until the `while` statement is false—a random-normal draw is less than zero. The number of ticks until this happens can simply be read from the tick counter on the Interface. Ten typical values show that results are extremely variable: 3073, 1129, 6662, 204, 1118, 4896, 1554, 2952, 3860, 1076. (From probability theory, this event should happen, on average, one time in 2330.)

Exercise 5

Here is a simple random-binomial distribution. (It is also included in the NetLogo file for exercises 1 and 2.)

```
to-report random-binomial [probability-true num-trials]

  ; First, do some defensive programming to make sure "probability-true"
  ; has a legal value
  if (probability-true < 0.0 or probability-true > 1.0)
    [
      type "Warning in random-binomial: probability-true equals "
      print probability-true
    ]

  let num-true 0
  repeat num-trials
    [
      if random-float 1.0 < probability-true [set num-true num-true + 1]
    ]
```

```
    report num-true
  end
```

(Ideally, students would note that there are algorithms in the numerical recipes literature that are much more efficient when the number of trials is high.)

Trying to compare results of this distribution to the random Poisson distribution results obtained in Exercise 2 should reinforce the differences between the distributions. If you exercise the binomial distribution with a probability true of 0.8 and number of trials equal to one, you get results of zero or one, with about 80% of values being one. The clear difference is that the binomial distribution cannot produce results higher than the number of trials; if you simulate 1000 trials, you cannot get more than 1000 "true" events. But with the Poisson distribution, if you simulate an event that happens on average 0.8 times per tick, the distribution can produce results of 2 and more events for any given tick.

The two kinds of distribution produce quite similar results when their parameters are small: when the mean rate of occurrence (Poisson) and probability true (binomial) are much less than one. The statements `random-poisson 0.08` and `random-binomial 0.08 1` produce results like these when repeated 1000 times:

Poisson:

Bin	*Frequency*
0	917
1	81
2	2
3	0

Binomial:

Bin	*Frequency*
0	936
1	64
2	0

Exercise 6

The `grim-reaper` method should be a binomial distribution: it models how many turtles die when the probability of each turtle dying is known. That probability of dying is set to `death-rate`, the fraction that needs to die to maintain a constant population size.

Another indicator that the distribution should be binomial, not Poisson, is that the number of deaths cannot exceed the number of turtles, which would be possible with a Poisson distribution.

Exercise 7

A NetLogo program illustrating this exercise is provided: `Ch15-Ex7_LogNormalReporter.nlogo`. Note use of the `set-plot-x-range` primitive to scale the histogram X axis nicely.

Exercise 8

An implementation of the stochastic Business Investor model is available: `BusinessInvestors_Ch15-Ex8_2ndEd.nlogo`.

In comparing this version with the baseline version (Section 10.4), students should find that investor wealth is a little lower. Investors continue moving to higher return-higher risk investments over time, whereas in baseline version they take fewer risks as time proceeds. The reason for that difference is that this empirical model is not state-based: decisions do not depend on how wealthy the individuals already are.