6 Analyzing Agent-Based Models

My answer to him was, "… when people thought the Earth was flat, they were wrong. When people thought the Earth was spherical they were wrong. But if you think that thinking the Earth is spherical is just as wrong as thinking the Earth is flat, then your view is wronger than both of them put together."

—Isaac Asimov

Measure twice, cut once.

—Proverb

Types of Measurements

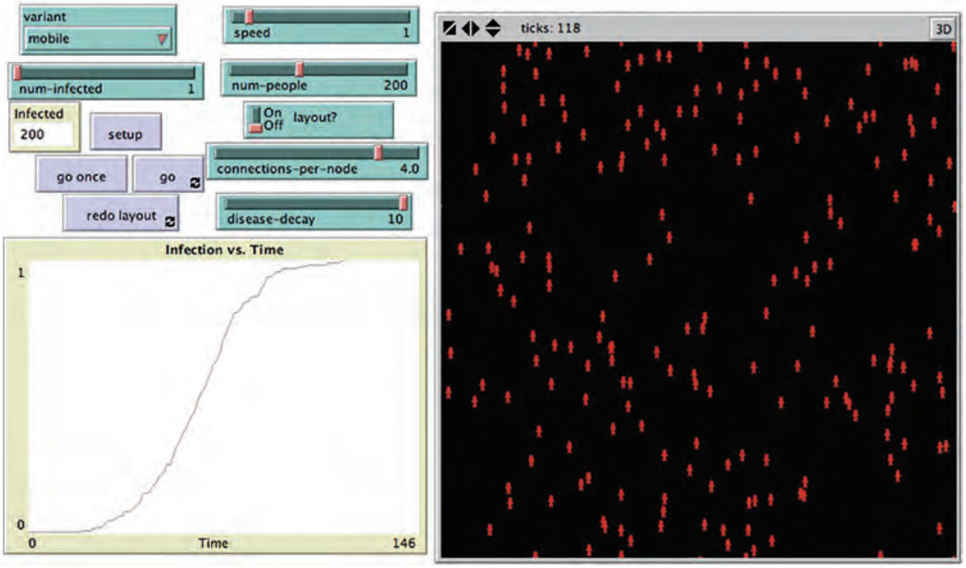
Heretofore, we have examined ABMs, modified them, built them from scratch, and analyzed their behavior. In this chapter, we will learn to employ ABM to produce new and interesting results about the domain that we are investigating. What kinds of results can ABMs produce? There are many different ways of examining and analyzing ABM data. Choosing just one of these techniques can be limiting; therefore, it is important to know the advantages and disadvantages of a variety of tools and techniques. It is often useful to consider your analysis methods *before* building the ABM, to enable you to design output that is conducive to your analysis.

Modeling the Spread of Disease

If someone catches a cold and is coughing up a storm, he might infect others. Those that he comes into contact with—his friends, co-workers, and even strangers—may catch the cold. If a cold virus infects someone, that person might spread that disease to five other people (six now infected) before they recover. In turn, those five other people might spread the cold to five more people each (thirty-one are now infected), and those twenty-five people might spread the cold to five additional people (a hundred and fifty-six people are now infected). In fact, the rate of infection initially rises exponentially.

However, since this infection count grows so quickly, any population will eventually reach the limit of the number of people who can be infected. For instance, imagine that the 156 people mentioned above all work for the same company of 200 individuals. It is impossible for the remaining 125 people to each infect five new people, and thus the number of infected people will tail off because there is no one left to infect. As we have described it so far, this simple model assumes that each person infects the same number of people, which is manifestly not the case in real contexts. As a person moves through their workspace, it might be the case that, they happen to not see many people in one day, whereas another individual might see many people. Also, our initial description assumes that if one person infects five people and another person infects five, there will not be any overlap. In reality, there is likely to be substantial overlap. Thus, the spread of disease in a workplace is not as straightforward as our initial description suggests. Suppose that we are interested in understanding the spread of disease, and we want to build an ABM of such a spread. How should we go about doing it?

First, we need some agents that keep track of whether they are infected with a cold or not. Additionally, these agents need a location in space and the ability to move. Finally, we need the ability to initialize the model by infecting a group of individuals. That is exactly how the NetLogo model we will be discussing in this chapter behaves (See figure 6.1). Individuals move around randomly on a landscape and infect other individuals whenever they come into contact with them.



variant: mobile

num-infected 1

Infected 200

speed 1

num-people 200

layout? On Off

connections-per-node 4.0

disease-decay 10

ticks: 118

Figure 6.1
Spread of Disease model.

Table 6.1
Infection Data

Population	50	100	150	200
Time to 100% Infection	419	188	169	127

Though this model is simple, it exhibits interesting and complex behavior. For instance, what happens if we increase the number of people in the model? Does the disease spread quicker throughout the population, or does it take a longer time because there are more people? Let us run the model at population sizes of 50, 100, 150, and 200, and examine the results. We will keep the size of the world constant, so that, as we increase the number of individuals, we are also increasing the population density. Along the way we will write down at what time the entire population becomes infected (see table 6.1).

Based on these results, we conclude that as the population density increases, the time to full infection dramatically decreases. This makes sense upon further contemplation. In the beginning, when the first person becomes infected, if there are not many other people around, the person has no one to infect, and thus the infection rate increases slowly. However, if there are many people around then there will be plenty of infection opportunities. Moreover, at the end of the run, when there are only one or two uninfected agents, they will be more likely to run into someone with an infection if the population count is high. This is true despite the fact that the total number of people that need to be infected increases.

Box 6.1
Language Change

In the Spread of Disease model, we discuss how diseases can spread from one individual to another. However, there is no reason to restrict this model to just disease spread. Ideas can also spread from one individual to another. One classic example of this is language change. There are many different types of language change that can occur but one clear example is the introduction of new words into a language (Labov, 2001). To view the Spread of Disease model in this way, we assume that when any two individuals come into contact, they talk to each other. If one of the speakers uses a new word, then he or she provides the other speaker with the ability to use that word in future encounters. In this way, we can see how the introduction of a new word into a language spreads through a population in much the same way that an infection does (Enfield, 2003). One difference between the current Spread of Disease model and language change, is that usually in language change there is a resistance to change, and that also needs to be included in the model. In the Explorations for this section you can explore how to change the Spread of Disease model to make it a more robust model of language change. See also the Language Change model (*Troutman & Wilensky, 2007*) in the Social Sciences section of the NetLogo models library.

Table 6.2
Your Friend's Data

Population	50	100	150	200
Time to 100% Infection	305	263	118	126

Table 6.3
Raw Data

Population	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
50	419	365	305	318	323	337	432	380	430	359
100	188	263	256	205	206	205	201	181	202	231
150	169	118	163	146	143	167	137	121	140	140
200	127	126	113	111	133	129	109	101	105	133

Let us suppose that we show this data to a friend of ours and she does not believe it. She believes that the time to 100 percent infection should grow linearly with the population. She looks over the code and determines that it seems to match the description (a process called verification, which we will discuss in the next chapter). After that, she runs the model and collects the same data that we did (a form of replication which we also discuss in the next chapter). Her data is in table 6.2.

These results do not support our friend's prediction that the time to 100 percent infection will grow as the population increases, but, on the other hand, they are quite different from the results that were originally collected. In fact, the time to 100 percent infection for 150 and 200 increases in our friend's data, seemingly contradicting our original results. Moreover, if we run the model several more times you might again get different results. We need some methods for determining if there are trends in the data.[1]

The data is inconsistent because most ABM models employ randomness in their algorithms—i.e., the code makes use of a random number generator. How the agents move around on the landscape is not specifically determined, but instead is the result of several calls to the random number generator at each time step that determine the actions of any particular agent. Moreover, these random decisions occur at least once for each agent in the population for each time step. Clearly, then, one set of runs is not enough to characterize the behavior of this model. Suppose, then, that we collect additional data for a set of population densities for ten different model runs as in table 6.3. Though most of these runs look more like our original results than our friend's results, it might be difficult to see

[1] In the particular case of this simple model, it is relatively straightforward to determine what the rate of increase in the model will be by using heuristic methods or by generating a closed-form solution. However, we will explore how to discover this rate directly from the data as that will often be the only method available.

clear trends, and to analyze the results overall. Thus, to describe these patterns of behavior it makes sense to turn to some statistics.

Statistical Analysis of ABM: Moving beyond Raw Data

Statistical results are the most common way of looking at any kind of scientific data whether it is computational models, physical experiments, sociological surveys, or other methods that generate data. The general methodology behind descriptive statistics is to provide numerical measures that summarize a large data set and describe the data set in such a way that it is not necessary to examine every single value. For instance, suppose we are interested in determining whether a coin is fair (i.e., it is as likely to turn up heads when flipped as it is to turn up tails) then we can conduct a series of experiments where we flip the coin and observe the results. One way to determine if the coin was fair would be to simply examine all of the observations: HHHHTTHTTT and, based upon our examination, make a determination if the coin was fair or not. However, if we wanted to examine a thousand, or ten thousand, or even a million such observations, it would take too much time to examine all of them. A better way is to simply count the frequency with which heads occurs, i.e., the observed probability of success, and the standard deviation of this observed probability. It is much easier to look at means and standard deviations than it is to examine large series of data (e.g., for HHHHTTHTTT, the observed probability is 0.5, and the expected outcome for ten trials is to observe five heads with a standard deviation of 1.58).

To apply this technique to our Spread of Disease model in more depth, we can create summary statistics of the table 6.3 results, which we display in table 6.4.

From these summary results, we see that the mean time to 100 percent infection declines as the population density increases. Another interesting result is that as the population density goes up, the standard deviation goes down. This means that the data is less varied. In other words, more trials are closer to the mean than farther away from it. This happens because when there are few agents, there is a possibility that individuals might not run into each other to transmit the disease for quite a while, but when there is a high density of individuals, there is less probability of this occurring, which means that the time to 100 percent infection remains closer to the mean time.

Table 6.4
Summary Statistics

Population	Mean	Std. Dev.
50	366.8	47.39385802
100	213.8	27.40154091
150	144.4	17.65219533
200	118.7	12.12939497

These results seem to confirm our original hypothesis that as population density increases the mean time to infection declines. Within ABM, statistical analysis is a common method of confirming or rejecting hypotheses. When initially examining an ABM we may start by exploring the space of possibilities (the parameter space) observing the variations in the results, but over time we will start to create hypotheses about how the inputs to the ABM generate various outputs. Devising an experiment like the one above, and analyzing the results is how we begin to confirm or reject these hypotheses.

If you are conversant with basic statistical methods, it is straightforward to carry out further statistical analysis on this data and attempt to describe the rate at which the time decreases as the population density increases. We could also carry out a statistical test to prove that changes in population density lead to different times to achieving 100 percent infection. Statistical tests for analyzing ABMs will be touched upon briefly in the next chapter. A detailed discussion of statistical analysis is beyond the scope of this textbook, but for a more in-depth introduction to statistical analysis, see an introductory statistics text.

Because of the natural “compression” of data that occurs when conducting statistical analysis, this is a useful technique when examining ABMs. ABMs create large amounts of data (the Spread of Disease model is just a small example), and if we can summarize that data we can examine large amounts of output in an efficient manner.

Numerous easily available tools can facilitate the conduction of statistical analysis. For instance, Microsoft Excel, the open source R package, SAS, Mathematica, and Matlab, all have packages and sets of functions that assist in the analysis of large data sets. It is usually quite easy to take data from an ABM and import it into one of these packages. Most ABM toolkits allow you to export your data to a CSV (Comma Separated Value) file, and all of the preceding tools can import such data. You can then carry out any statistical analysis necessary in these toolkits. In addition, most ABM toolkits give you the basic ability to carry out simple statistical analysis within the package itself (e.g., in NetLogo there are MEAN and STANDARD-DEVIATION primitives). Thus, while the model is running, the ABM itself can generate summary statistics. Finally, some ABM toolkits provide the ability to connect to statistical packages while the model is running (e.g., in NetLogo you can use Mathematica Link (*Bakshy & Wilensky, 2007*) to control NetLogo from within Mathematica, which allows you to retrieve any results from your model within Mathematica. Similarly, you can use the NetLogo R extension to conduct analyses with the R statistical package).

The Necessity of Multiple Runs within ABM

As illustrated earlier, when you are trying to collect statistical results from an ABM you should run the model multiple times and collect different results at different points. Most ABM toolkits will provide you with a way to collect the data from these runs automatically (e.g., in NetLogo there is a tool called BehaviorSpace[2]) and it is important to know how to access these features. However, even when these features do not exist, ABM toolkits are often full-featured programming languages, allowing you to write your own tools for creating experiments to produce the data sets you want to analyze. In fact, in the analysis described before, that is what we did to generate the data. We simply wrote the following code:

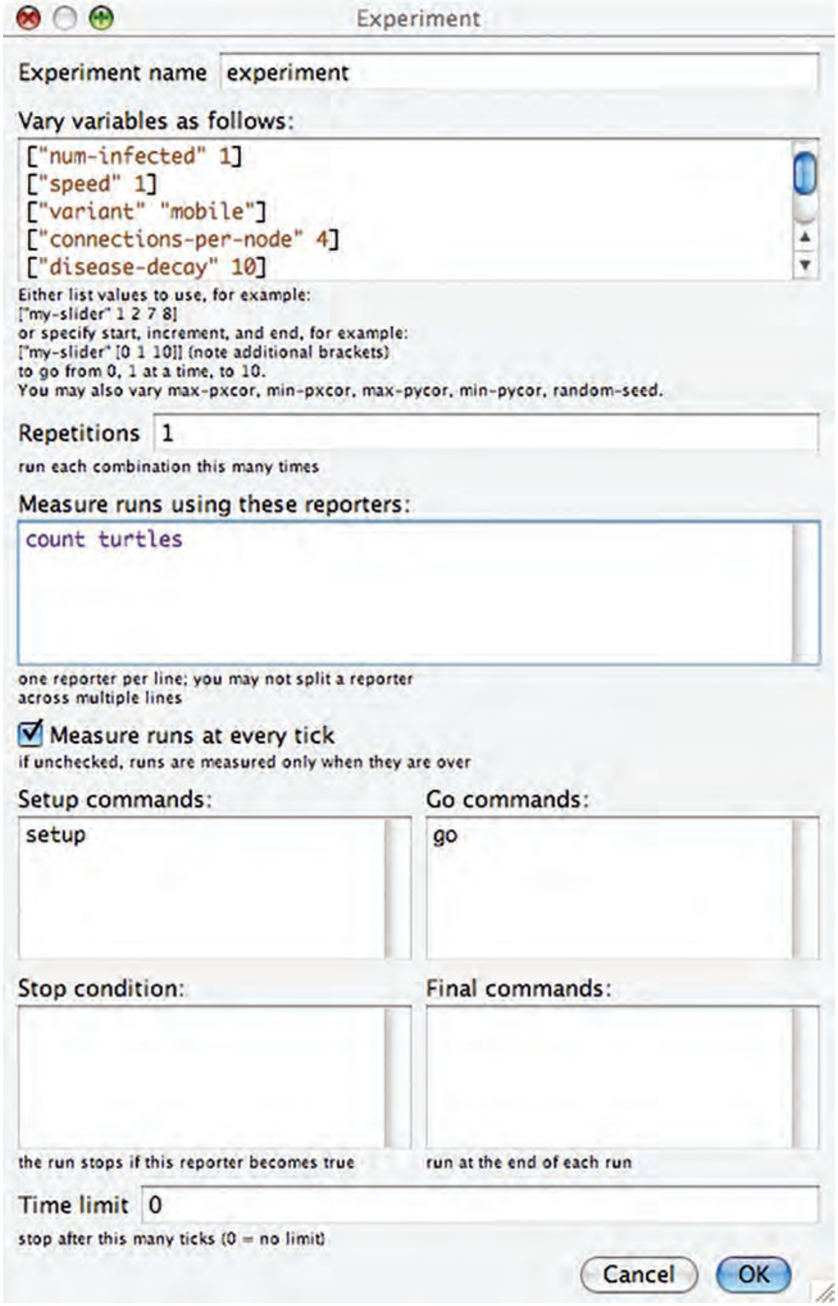
```
repeat 10 [
    set num-people 50
    setup
    while [count turtles with [ not infected? ] > 0 ]
       [ go ]
    print ticks[3]
]
```

By modifying the value of NUM-PEOPLE, we were able to generate the four sets of data presented in table 6.3. This technique can become tedious if you want to explore a large number of variables, or a large number of settings for one variable. Consequently, to aid in this process, most ABM toolkits have a *batch experiment tool*. These tools will automatically run the model multiple times with multiple different settings and collect the results in some easy to use format like the CSV files mentioned earlier. For instance, if we wanted to run the same experiment described before using NetLogo's BehaviorSpace, we would begin by starting BehaviorSpace and selecting a new experiment. The resulting dialog is illustrated in figure 6.2.

It is instructive to go through the steps of setting up a BehaviorSpace experiment. We start by giving a name to the experiment. Let us call this experiment "population density," which will correspond to the name of the output file that BehaviorSpace generates to hold the results of our experiment. We can then select the parameters and parameter ranges for BehaviorSpace to "sweep" so as to recreate the same experiment within BehaviorSpace. We can then set each parameter using the BehaviorSpace syntax. For instance, ["num-people" 50] sets NUM-PEOPLE to 50, or we can set NUM-PEOPLE to a range of values, and BehaviorSpace will automatically run the model for each value in the range. If we want to recreate the results above we need to vary NUM-PEOPLE by using ["num-people" [50 50 200]]. This tells BehaviorSpace to start at 50 and increment that number by 50 until it reaches 200. Additionally, we can modify two parameters at the same time. For instance, if we wanted to modify the number of people originally infected (NUM-INFECTED) at the same time as NUM-PEOPLE, we could specify ["num-infected" [1 1 5]]. This will

[2] Wilensky & Shargel (2001).

[3] The print primitive will print the data to the command center. It is often useful to write the data to a file using the file-open, file-print, and file-close primitives.



Experiment name: experiment

Vary variables as follows:

```
["num-infected" 1]
["speed" 1]
["variant" "mobile"]
["connections-per-node" 4]
["disease-decay" 10]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions: 1
run each combination this many times

Measure runs using these reporters:

```
count turtles
```

one reporter per line; you may not split a reporter
across multiple lines

☑ Measure runs at every tick
if unchecked, runs are measured only when they are over

Setup commands:

```
setup
```

Go commands:

```
go
```

Stop condition:
the run stops if this reporter becomes true

Final commands:
run at the end of each run

Time limit: 0
stop after this many ticks (0 = no limit)

Figure 6.2
BehaviorSpace, a batch experiment tool in NetLogo.

then run the various values of num-people and num-infected at the same time (i.e., (num-infected, num-people) = (1, 50), (2, 50), (3, 50), (4, 50), (5, 50), (1, 100), (2, 100), (3, 100), (4, 100), (5, 100), (1, 150), (2, 150), (3, 150), (4, 150), (5, 150), (1, 200), (2, 200), (3, 200), (4, 200), (5, 200)) or 20 different parameter settings). Next we can look at the number of times we want each set of parameters to repeat. In our experiment, we collected the results of 10 repetitions, so we set this to 10. After that we can specify the values that we are interested in, since we want to examine the time it takes until 100 percent infection occurs, and we can put the special reporter "ticks," which reports the current time, in this box. We can then turn off "Measure runs at every tick" and all we will collect is the final number of ticks. The SETUP and GO commands allow you to specify any additional NetLogo code that you need to make the model start and go. "Stop condition" allows you to specify special stop conditions for each run, and "final commands" allows you to insert any commands that you want executed between runs of the model. Finally, the time limit box allows you to set a limit to the number of ticks for which the model will run if no stop conditions are reached. After we make these changes, the BehaviorSpace dialog looks like figure 6.3.

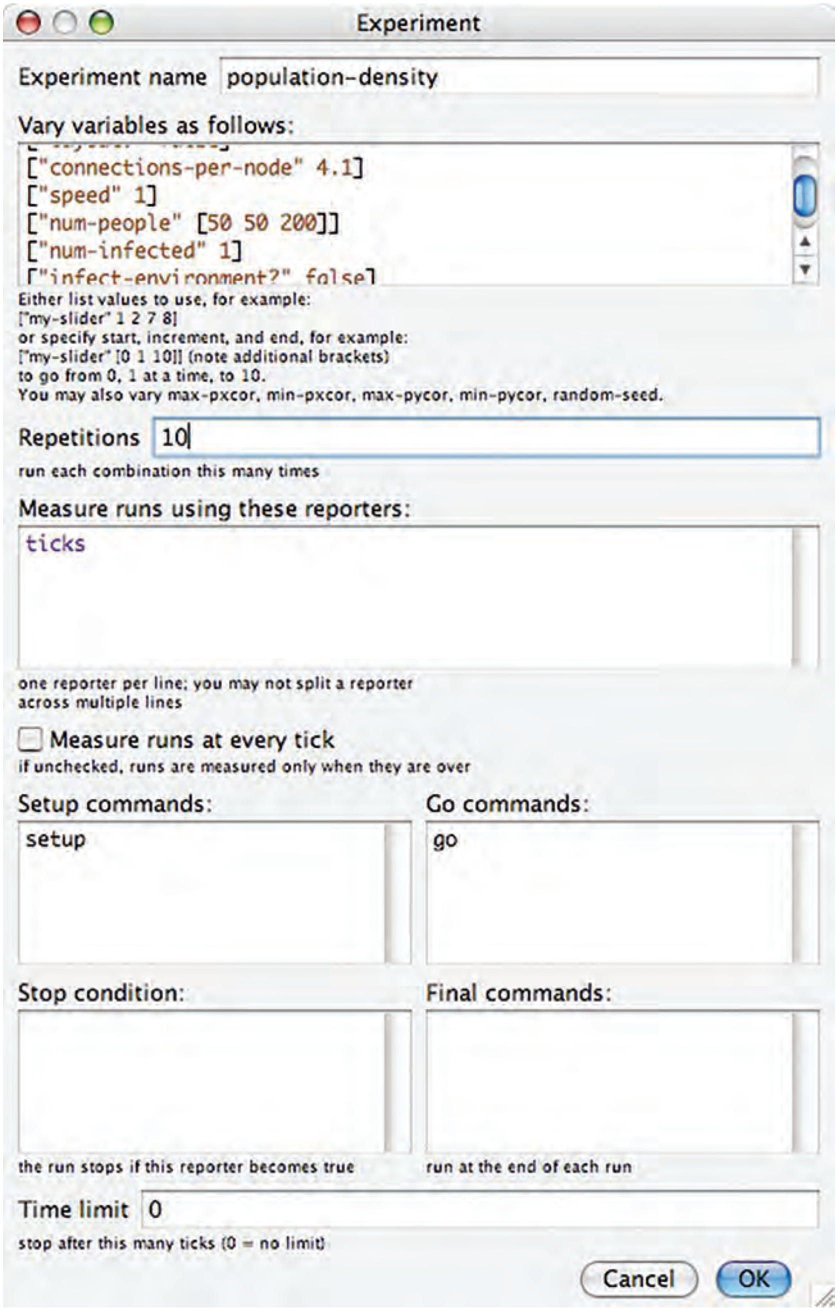
After we click OK, we go back to the experiment selection dialogue. From here we can run the experiment and select whether we want it in table (row-oriented) or spreadsheet (column-oriented) format (or both or neither), and we specify where we want the file saved. If, after we run the experiment, we want to look at the results, we can start up a spreadsheet or another statistical package and load the CSV file. The results of our experiment are illustrated in table format in table 6.5. In addition to the actual data we are interested in (NUM-PEOPLE vs. TICKS), BehaviorSpace also displays all of the additional parameters, such as NUM-INFECTED and SPEED.[4] These results are similar to the results described earlier, but they do not correspond exactly, since different random number seeds were used.

In general in ABM, it is important to carry out multiple runs of your experiments so that you can determine if some result is truly a pattern or just a one-time occurrence. One common way is to start by manually running your model multiple times, but to get a better sense of the results it is usually much easier to use a batch experiment tool. We have illustrated the BehaviorSpace tool, which is the batch experiment tool for NetLogo but most ABM toolkits have a similar method of sweeping parameters and collecting multiple runs.

Using Graphs to Examine Results in ABM

As you look over the summary statistics in table 6.4 (or, even worse, the full data set in table 6.3), you may become aware of a deficiency in such information. The summary

[4] In fact, one of the additional variables that BehaviorSpace automatically reports every time it prints out a row of this spreadsheet is the tick count, so our ticks reporter was actually superfluous.



Experiment

Experiment name population-density

Vary variables as follows:

```
["connections-per-node" 4.1]
["speed" 1]
["num-people" [50 50 200]]
["num-infected" 1]
["infect-environment?" false]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions 10

run each combination this many times

Measure runs using these reporters:

```
ticks
```

one reporter per line; you may not split a reporter
across multiple lines

□ Measure runs at every tick

if unchecked, runs are measured only when they are over

Setup commands:

```
setup
```

Go commands:

```
go
```

Stop condition:

the run stops if this reporter becomes true

Final commands:

run at the end of each run

Time limit 0

stop after this many ticks (0 = no limit)

Figure 6.3
Final settings for BehaviorSpace.

Table 6.5
BehaviorSpace Data Imported into a Spreadsheet

BehaviorSpace Table data

population-density

DATE
TIME

[run number]	network?	layout?	connections-per-node	speed	num-people	num-infected	infect-environment?	[tick]	ticks
1	FALSE	FALSE	4.1	1	50	1	FALSE	299	299
2	FALSE	FALSE	4.1	1	50	1	FALSE	432	432
3	FALSE	FALSE	4.1	1	50	1	FALSE	444	444
4	FALSE	FALSE	4.1	1	50	1	FALSE	400	400
5	FALSE	FALSE	4.1	1	50	1	FALSE	467	467
6	FALSE	FALSE	4.1	1	50	1	FALSE	397	397
7	FALSE	FALSE	4.1	1	50	1	FALSE	337	337
8	FALSE	FALSE	4.1	1	50	1	FALSE	280	280
9	FALSE	FALSE	4.1	1	50	1	FALSE	366	366
10	FALSE	FALSE	4.1	1	50	1	FALSE	257	257
11	FALSE	FALSE	4.1	1	100	1	FALSE	268	268
12	FALSE	FALSE	4.1	1	100	1	FALSE	165	165
13	FALSE	FALSE	4.1	1	100	1	FALSE	183	183
14	FALSE	FALSE	4.1	1	100	1	FALSE	200	200
15	FALSE	FALSE	4.1	1	100	1	FALSE	151	151
16	FALSE	FALSE	4.1	1	100	1	FALSE	206	206
17	FALSE	FALSE	4.1	1	100	1	FALSE	217	217
18	FALSE	FALSE	4.1	1	100	1	FALSE	234	234
19	FALSE	FALSE	4.1	1	100	1	FALSE	197	197
20	FALSE	FALSE	4.1	1	100	1	FALSE	209	209
21	FALSE	FALSE	4.1	1	150	1	FALSE	131	131
22	FALSE	FALSE	4.1	1	150	1	FALSE	127	127
23	FALSE	FALSE	4.1	1	150	1	FALSE	173	173
24	FALSE	FALSE	4.1	1	150	1	FALSE	179	179
25	FALSE	FALSE	4.1	1	150	1	FALSE	203	203
26	FALSE	FALSE	4.1	1	150	1	FALSE	124	124
27	FALSE	FALSE	4.1	1	150	1	FALSE	128	128
28	FALSE	FALSE	4.1	1	150	1	FALSE	190	190
29	FALSE	FALSE	4.1	1	150	1	FALSE	170	170
30	FALSE	FALSE	4.1	1	150	1	FALSE	141	141
31	FALSE	FALSE	4.1	1	200	1	FALSE	103	103
32	FALSE	FALSE	4.1	1	200	1	FALSE	124	124
33	FALSE	FALSE	4.1	1	200	1	FALSE	124	124
34	FALSE	FALSE	4.1	1	200	1	FALSE	138	138
35	FALSE	FALSE	4.1	1	200	1	FALSE	211	211
36	FALSE	FALSE	4.1	1	200	1	FALSE	137	137
37	FALSE	FALSE	4.1	1	200	1	FALSE	115	115
38	FALSE	FALSE	4.1	1	200	1	FALSE	110	110
39	FALSE	FALSE	4.1	1	200	1	FALSE	142	142
40	FALSE	FALSE	4.1	1	200	1	FALSE	130	130

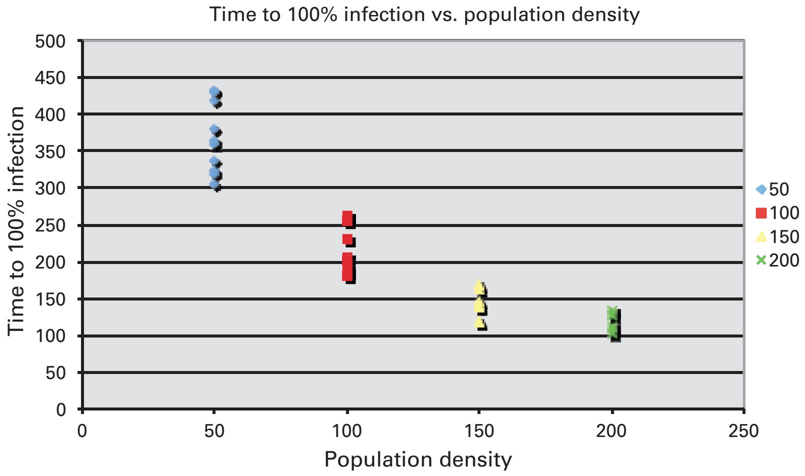


Figure 6.4
All raw data in a graph-based form.

statistics provide a check that you were correct when you thought that time to 100 percent infection decreased as population increases, but they do not tell the whole story. They do not provide you with a total description of the distribution of the data. Moreover, looking at the data itself is difficult, since it is hard to look at a series of numbers, as in table 6.3, and discern any meaningful pattern quickly. It would be nice to present the data in a way that you could quickly understand the full data set. Graphs, which embed the full set of data in a pictorial representation, facilitate understanding while still providing all the data available. For instance, we can take the data from table 6.3, and create a graph of the four population densities versus the time to 100 percent infection, as illustrated in figure 6.4. In NetLogo creating simple graphs is easy to do and will often suffice for simple data analysis. But with complex data sets, designing a useful and immediately informative graph can be challenging and is the subject of an extensive body of literature (Tufte, 1983; Bertin, 1983).

From figure 6.4, we can quickly see how the data is distributed and how the data changes with population density. This data seems to indicate that as population increases the differences between times to infection decreases. If we kept looking at higher and higher population values, it might be that the time to infection starts leveling off, i.e., regardless of whether there are 2,000 or 3,000 individuals, it still takes 100 time steps to fully infect the population. For instance, it appears that the times to 100 percent infection is not very different whether the population is 150 or 200.

However, this data might still be too complicated to understand. We can use the same technique to make the summary data easier to understand as is illustrated in figure 6.5.

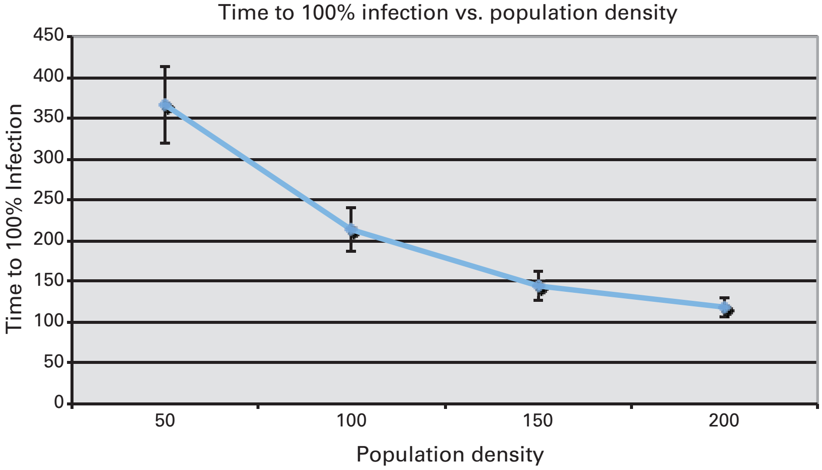


Figure 6.5
Summary data.

In this figure, the mean of each data series is plotted as well as error bars of one standard deviation. This new figure might not be easier to understand than figure 6.4, but if there were one hundred data points in figure 6.4 rather than ten data points, then a figure like figure 6.5 might be very helpful.

Graphs are not only useful to help clarify your data after a model run, but they are also useful during the running of a model. Many ABM toolkits include capabilities for continually updating graphs and charts during the running of a model and thus enable you to see the progress of the model temporally. This can be very useful for understanding the dynamics and temporal evolution of the model. For instance, in the Spread of Disease model there is a graph that illustrates the change in the fraction of infected agents as time proceeds (see figure 6.6). If you were to examine one run from table 6.3 for one value, then you would be looking at the final x-value on this graph, since table 6.3 records the x-value of this graph when the y-value hits 1.0.

This graph (figure 6.6) is an example of a *time series* since it is data collected over time. Time series analysis is very important in agent-based modeling because much of the data generated by ABMs is temporal in nature. One way to analyze a time series is to determine if there are particular phases that data goes through during the course of a run. For instance, in the preceding results there seem to be three very different phases of behavior. In the beginning, the number of infected agents grows very slowly. After around fifty time steps, the number of infected agents grows very quickly. Finally, after around one hundred time steps, the number of infected agents increases slowly. Usually, it is useful to compare these three phases of data to the behavior in the model. During the first phase, there are only a

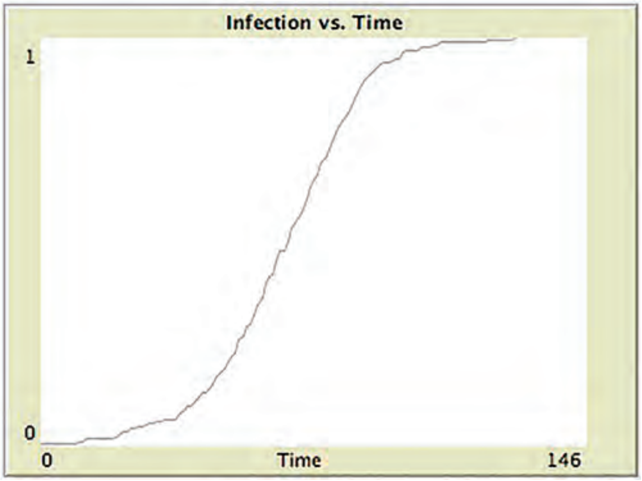


Figure 6.6
Fraction infected versus time for 200 agents.

few infected agents spreading the disease, and so the spread is very slow. During the second phase, the number of infected agents grows fast because there are a lot of infected agents and a lot of agents being infected. During the final phase, there simply are no longer that many agents who can be infected and so the number of infections slows considerably. This is one example of how we can use time series to help understand the behavior of a model.

Many times within ABM, it is useful to look not only at one particular run like in figure 6.6, but also at multiple runs overlaid on each other. This can help you see not only the general trend of the model, but the possible paths that the model usually takes. For instance, it might be possible that a model always winds up taking one of two paths: either the disease spreads throughout the population in the S-shaped curve above, or it spreads very slowly and then quickly spreads to all members of the population. Such a bifurcation in the path of the model would show up in a graph of overlaid characteristic runs. Looking at a set of characteristic runs allows you to observe the characteristic behavior of the system rather than just a particular run.

Analyzing Networks within ABM

As we mentioned in chapter 5, having agents walking around and interacting is useful if you want to examine physically proximate interactions, but many types of interactions do not occur in physical space, but rather across social networks. The Spread of Disease model we have explored relied on contact between moving agents to spread the infection. Diseases do indeed spread in part by people walking around and infecting other people, but

Box 6.2
Time Series Analysis

Time series analysis is a research area in and of itself. The basic question in this area is how to characterize a data set that is temporal in nature. Many times the goal of time series analysis is to predict what will be the next set of data points in the series. One way to do this is to decode the past time series experience in to both short term and long term components. By differentiating between these two, it is possible to predict the long-term behavior of the time series. There are many subject areas for which times series analysis is useful, such as ecology, evolution, political science, and sociology. Time-series analysis also has applications to stock market analysis, where if it is possible to predict the long-term behavior of a stock, it is possible to make money off of that stock. For more information on time series analysis see (Box, Jenkins, & Reinsel 1994).

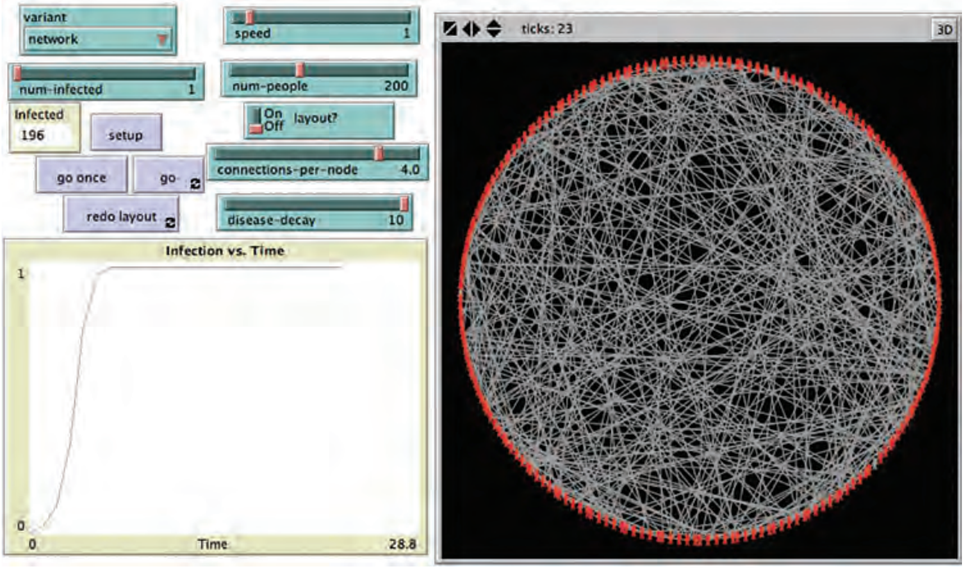
the spread of diseases relies heavily on the social network of individuals. In fact, some diseases such as sexually transmitted diseases are not spread by casual contact at all but only through certain kinds of social networks. The Spread of Disease model was designed to explore that possibility as well. There is an interface element, the chooser, which allows you to select different variants of this model. When the chooser is set to "network," instead of agents moving around on the plane, they are connected via a network, and the disease spreads over the network. The network that is created in this model is a particular type of network, a random graph,[5] which we saw in the previous chapter (in this case an Erdös-Rényi random graph [1959]).

In a network model, the location of the agents in physical space does not matter—what matters is who is connected to whom. Herein we analyze the effect of varying an important network property, the connections-per-node (i.e., average degree of the network). This factor determines how many connections, on average, each individual has to other individuals in the network. Let us analyze this case.

We begin by setting the VARIANT chooser to "network," and the connections-per-node to a reasonable number such as 4.0. Then we can run the model and see the results as illustrated in figure 6.7.

We now need to create an experiment to explore how connections-per-node affects disease spread. If you play around with the model, you will notice that often the disease does not spread to all of the individuals. For instance, in figure 6.7, the disease has only infected 197 of the 200 individuals. As you adjust the sliders back and forth, you will realize that there seems to be a critical point near 1.0. If the connections-per-node is less than 1.0, then the disease does not infect very many individuals, but if it is greater than

[5] This is not as realistic a social network as other types of well known networks such as small world networks or scale-free networks (Barabasi & Albert, 1999; Watts & Strogatz, 1998), but it is a good base case to analyze.


variant: network

speed 1

num-infected 1

num-people 200

layout? On Off

Infected 196

connections-per-node 4.0

disease-decay 10

ticks: 23

Figure 6.7
Network variant of the Spread of Disease model.

1.0, then it infects a much larger proportion of the population. Let us proceed to create an experiment to explore this.

There is no guarantee that the entire population will become infected, and consequently it is necessary to create a new termination criteria for the experiment. One way to do this would be to look at the number of agents that are infected at a particular time step, allowing the model to run for a while so that there is as much spread as possible. We can do this in BehaviorSpace by setting a time limit for our run. Here we set this time limit to 50. We create a BehaviorSpace experiment that varies the connections-per-node from 0.5 to 4.0 at 0.5 increments. Importing these results into a spreadsheet yields the data summarized in table 6.6.

From these results, the average number of individuals infected grows substantially as the connections-per-node exceeds 1.0. This is in fact a well-known property of random graphs, as the average degree (connections-per-node) exceeds 1.0, a giant component (a large connected subset of the nodes) forms in the network (Janson et al., 1993). If an infection occurs inside this giant component, then the disease will infect a large percentage of the population. This result corresponds exactly to a result in classical (non-network-based) epidemiology models. In these classical models, if the rate of infection over the rate of recovery of a disease exceeds 1.0 then an epidemic will occur in the population, if it is below 1.0 the disease will die out. You can think of the links in the network-based

Table 6.6
Number of Agents Infected after 50 Ticks

connections-per-node	mean # infected	std. dev.
0.5	1.8	1.229272594
1	15.1	21.75852323
1.5	68	59.11946474
2	145.3	50.64922946
2.5	181.1	3.348299734
3	189.3	3.093002856
3.5	174.5	61.03050239
4	196.4	2.170509413

model as infections that the individual will transmit during the time that the individual carries the disease. Given this way of thinking of links, the average degree and the ratio of the rate of infection to rate of recovery are equivalent. In other words if the average individual will at some point infect at least one individual then you will have an epidemic, i.e., all of the individuals in the model will become infected. Consequently, you are likely to have an epidemic if the connections-per-node parameter is greater than 1.0.

Average degree (connections-per-node) is just one property of a network. There are many more. For instance, the property of average path length is a measure of the average distance between any two nodes in the network. This property affects the spread of disease, since if a network has a high average path length, it will take a long time for the disease to reach everyone. However, if the network has a low average path length, the vast majority of people will be infected very quickly. Another widely used property is the clustering coefficient of a network. A clustering coefficient is a measure of how tightly clustered the network is—that is, a measure of how many of the nodes an agent is linked to are also linked to each other (or how many friends of your friends are also your friends). These measures are just two examples of a wide variety of metrics and analysis tools that have been created within the field of Social Network Analysis (SNA). SNA and ABM often work well together. ABM provides a rich model of the process of a phenomenon, while SNA provides a detailed model of the patterns of interaction. Together they allow you to model both the pattern and process that exist within these complex systems.

Each of these network properties can be analyzed as to their effect on the spread of disease. You would need to create reporters to measure these properties, but you could also export the network that you are examining and import it into a standard network analysis toolkit like UCINet or Pajek for further examination.[6]

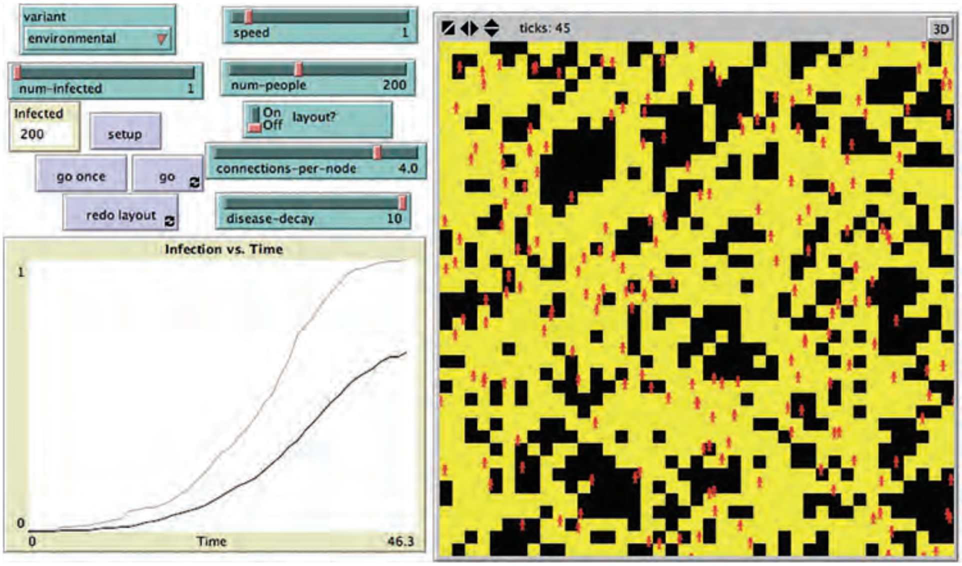
[6] Some of these reporting functions may be provided as primitives in the ABM toolkit. NetLogo does include many of these. NetLogo's network extension also provides a more comprehensive set of primitives.

Box 6.3
Diffusion of Innovation

We have discussed this model as a spread of disease, and we have also mentioned how it shares commonalities with models of language change. Another way to view this model is as a diffusion of innovation model. One agent becomes “infected” or adopts an innovation, like a new audio device, a new business process, or a new movie that the agent likes (Rogers, 2003). This agent then spreads this innovation by word of mouth to their friends and co-workers. The network-based version of the Spread-of-Disease model is particularly good for modeling diffusion of innovation since innovations usually spread across social networks and not necessarily across physical space (Valente, 1995). One commonly discussed topic in innovation diffusion is the role of influentials, i.e., are there some people in social networks who are better able to spread innovations than others (Watts, 1999). In the Explorations we will propose extending this model to more explicitly model the diffusion of innovation.

Box 6.4
Social Network Analysis

Social Network Analysis (SNA) is a burgeoning field of research. The basic premise of SNA is that, within social systems, the structure of interactions is at least as important as the type of interactions that occur. In other words, it’s not just how *you* interact, but it’s also *who* you interact with and *who* they interact with. One of the major findings of this area of research is that most people are connected through only a few intermediate connections. This is colloquially known as the Six Degrees hypothesis: that everyone in the world is only six connections away from everyone else. This hypothesis is based on a Stanley Milgram experiment where it was found that on average it took six letters for someone from Iowa to contact someone in New York City. Recently, this has been formalized as the idea of Small World networks in the book Six Degrees by Duncan Watts (2003), building upon work he carried out with Strogatz (1998). For a set of classical papers in Social Network Analysis, see Newman, Watts and Strogatz (2006) and for an introductory textbook, see Newman (2010).



variant: environmental
speed: 1
num-infected: 1
num-people: 200
Infected: 200
layout?: Off
connections-per-node: 4.0
disease-decay: 10
ticks: 45

Figure 6.8
Environmental variant of the Spread of Disease model.

Environmental Data and ABM

After looking at the model for some time, we might realize that for the diseases we are interested in studying, such as the common cold, the mobile agent model makes more sense. However, the mobile agent model did not go far enough. Besides agent-to-agent transmission of cold germs, there can also be agent-to-environment and environment-to-agent transmission. For instance, if someone has a bad cold and wipes his nose with his hand, and then opens a door, someone else who comes through that door shortly after the person with the disease might catch the cold. Of course, germs do not live very long outside the body so these environmental infections might decrease after time; so we would want the model to reflect this.

In fact, the Spread of Disease model allows us to examine this scenario. There is an environmental interaction effect. This variant can be seen in figure 6.8. If you turn this variant on (by choosing "environmental" in the VARIANT chooser) then the patch below any infected agent will become yellow. For a limited amount of time (DISEASE-DECAY), this patch will infect any other agent who steps on it. Let's investigate how disease-decay affects the time to 100 percent infection. We carry out an investigation in a similar manner to the one described before. This experiment is very similar to the one described for the original variant, but instead of changing the NUM-PEOPLE, which we fix at 200, we change the rate of DISEASE-DECAY from 0 (which exactly represents the original variant) to 10 at single time step intervals. We create a BehaviorSpace experiment to carry this out and import the results into Excel summarizing our findings in table 6.7.

Table 6.7
Time to 100% Infection, Environmental Variant

disease-decay	Average	Std. Dev
0	126.4	12.2854928
1	71	4.988876516
2	62	7.363574011
3	51	4.242640687
4	51.2	2.780887149
5	49.4	2.716206505
6	49.9	2.643650675
7	46.5	2.758824226
8	48.5	3.341656276
9	47.4	3.062315754
10	47.3	2.213594362

We can see that as the environmental decay parameter increases (i.e., the disease remains in the environment for a longer period of time) the time to 100 percent infection decreases. In the original model or when DISEASE-DECAY is set to 0, the only thing that can infect agents is other agents, but when the DISEASE-DECAY is positive then patches and agents can infect other agents. As DISEASE-DECAY increases, the number of patches that can infect other agents increases as well, meaning there are simply more objects in the environment, which can cause the spread of the disease.

It might be possible that the effect of DISEASE-DECAY is not completely separate from the population density. After all, if there are not many individuals in the model then a long DISEASE-DECAY might have a negligible effect over having a small DISEASE-DECAY. This can also be investigated using BehaviorSpace. We set it up to vary both DISEASE-DECAY and NUM-PEOPLE at the same time and report the number of ticks until full infection. To make sure we do not get anomalies, we average the data over ten runs. This results in a large number of runs, since we are looking at four values for NUM-PEOPLE and eleven values for DISEASE-DECAY, resulting in 440 runs. We can display this data using a three-dimensional graph (e.g., figure 6.9). As we see from this chart, both NUM-PEOPLE and DISEASE-DECAY have an effect on the results. However, it is clear that it is only when both of these values are very small that the model is sensitive to their results. The sharp peak in the results of the values as we decrease each variable indicates this. If NUM-PEOPLE is small and so is DISEASE-DECAY we will see a dramatic change in the time to full infection if we alter either of these variables.

A powerful aspect of ABMs, is that in addition to showing us the dynamics of how a disease spreads, it also shows us the pattern of infection. The yellow patches indicate

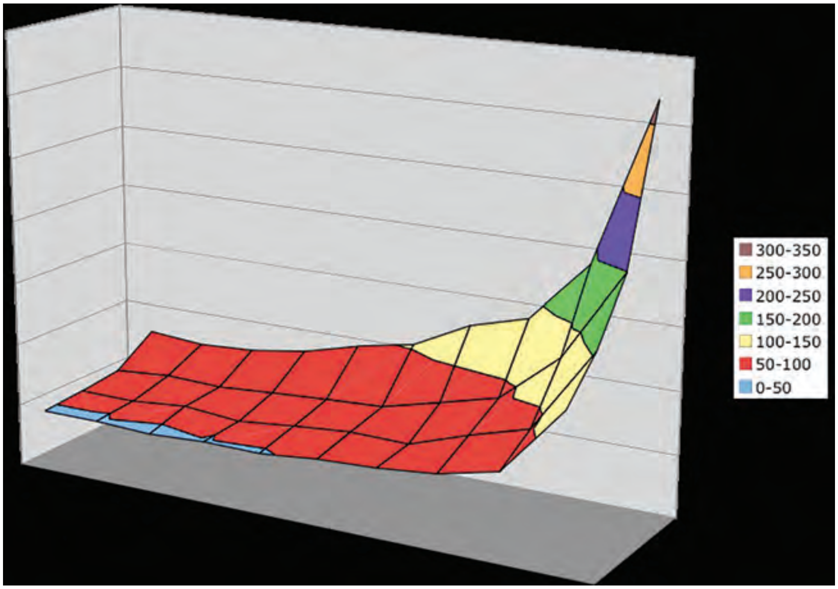


Figure 6.9

3D Chart of NUM-PEOPLE and DISEASE-DECAY versus time to 100 percent infection.

where the disease is still alive and how it might infect other agents. This could be useful if we are trying to control a disease. This disease model leaves long, stringy patterns of environmental infection, which is a product of how the individuals move through the landscape. Since these patterns of disease are not well clustered, then it does not make much sense to try and quarantine an area of the world to prevent the disease from spreading. Instead, we have to track down each individual in the world that is infected and attempt to cure them of the disease they have. Yet, if the disease spreads out from a single point, then if we were aware that there was an outbreak of a disease in an area it would make sense to create a ring of immunity around the disease through vaccination, for example, and slowly move toward the center of the ring, curing individuals along the way. This can be a successful method for smallpox control (Kretzschmar et al., 2004).

These qualitative stories of patterns in the environment are interesting, but there are also quantitative ways to attempt to capture the same results. For instance, we can take a map of yellow disease trails and measure what is called, in landscape metrics (McGarigal & Marks, 1995), the *edge density* of the yellow trails.[7] The edge density, as described in

[7] Typically, landscape metrics are analyzed utilizing a computational tool like FRAGSTATS (McGarigal & Marks, 1995).

Box 6.5
Geographic Information Systems (GIS)

The central idea of the field of Geographic Information Systems (GIS) is that most data in the world can be indexed via a spatial location. Moreover, by representing this data in spatial ways we can gain a better understanding of complex phenomena. GIS is used to analyze everything from the spread of disease as described in this chapter, to the environmental effects of suburban sprawl (Brown et al., 2008), to the flow of commuters in a transportation network. Since GIS has developed a powerful model of the spatial pattern of data, it is useful to combine GIS with ABM, which has a powerful model of the process of data transformation. Together, GIS and ABM can provide rich pictures of both the pattern and process of complex spatial systems. For an introduction to GIS, please see (Longley, Goodchild, Maguire, & Rhind, 2005).

chapter 5, is the ratio of the number of edges between two different states of the environment (in this case infected and noninfected patches) to the overall area. If the edge density is low, then similar patch states are highly clustered and have very few shared edges with different patch states; when studying the spread of disease, this would suggest that you should use a ring type of intervention. If the edge density is high, which is the case in the preceding environmental interaction model, then the different location types are interwoven; when studying the spread of disease, this would suggest the use of a targeted intervention.

Edge density is just one of many types of landscape metrics that have been created. Geographic Information Systems (GIS) is an entire field devoted to the study of patterns in the environment. GIS provides a rich description of the pattern within an environment when those patterns are geographically related. This is similar to the model of pattern that SNA provides within network-related environments. GIS combines well with ABM because together they enable the creation of elaborate models of both pattern and process.

We could implement some of the measures such as edge density within our ABM, and some ABM toolkits provide tools to make this easier. However, it might be better to simply export the data from your ABM to a GIS tool like ArcView Grass or MyWorld (Edelson, 2004), which have been developed specifically for understanding geographic patterns. This can be done in a number of ways from passing text files, to a tightly coupled application-programming interface (API) (Brown et al., 2005). We saw an example of a loosely coupled ABM model in chapter 5, the Grand Canyon model (*Wilensky, 2006*). This model reads in a digital elevation map that was created in a standard GIS software package, and it uses the elevation map to predict how rain flows across the terrain of the Grand Canyon.

Summarizing Analysis of ABMs

Measuring and analyzing ABMs presents different challenges from the analysis of equation-based models and even from other types of computational models. This is because of the large number of inputs and outputs that are normally associated with ABMs. Because ABMs enable the model user to control so many aspects of the agents, there are often a large number of inputs that must be specified. For instance, in a model of commuter patterns in an urban area, each agent or commuter may have characteristics such as age, wealth, environmental group membership, children, and ethnicity, not to mention nondemographic parameters such as personal preferences for waiting times and aesthetic qualities. On the output side since we are modeling the micro-level behavior of the system, it is possible not only to generate aggregate patterns of data, but also individual patterns. For example, in the commuter model mentioned earlier, we could observe the average commuting time over all individuals, but we could also break that down by any other characteristic. We can examine data along orthogonal dimensions as well. For example, we could examine the commuting time for anyone who takes a particular highway to work.

The large number of possible inputs and outputs gives researchers the ability to precisely control and measure their models. However, trying to control for them all can cause combinatorial explosion. If the commuting model has one thousand agents (commuters) that each have five characteristics (e.g., wealth, eco-friendly, children, age, work location) each of which can take just two values (e.g., high/low, yes/no, yes/no, old/young, north/south), then there are 2^{10} or 1,024 possible different types of individuals and $1{,}024^{1000}$ possible populations of individuals. This is one of the core reasons behind the ABM Design principle we presented in chapter 4—when building an agent-based model, start as simple as possible, and add complexity only as it is necessary to improve the model. The example we just gave is actually a relatively small parameter space since most agent characteristics will be real-valued not binary, and most models will have not only agent parameters but also environmental and global parameters. Despite its relatively small size, this number is much too large to exhaustively examine, especially when we are interested in a set of results for each run and not just one number. For example, in the commuter model imagine that there are five measures we are interested in, the average commuting time for all individuals and the commuting times for five subclasses of the individual types described before (e.g., wealthy individuals, individuals heading north). Essentially our model becomes a mapping from $1{,}024^{1000}$ possible inputs to five real-valued outputs. However, even this conceptualization is limiting, because often when examining an ABM we are interested not only in the final value of a particular output but also the dynamic patterns of the model, like the time series we examined. Thus, we are not really interested in just five outputs, but the dynamics of five outputs over many time steps. If in the commuter model we assume we are observing the results for a year of working days, then we are talking about $5 \cdot 20 \cdot 12 = 1{,}200$ real-valued outputs.

Thus, though the vast number of ABM inputs gives a model author a very precise level of control and the vast number of outputs gives the author a lot of detail, it also presents some challenges. First, the vast number of inputs means that there are that many more parameters of the model to validate. Each parameter must be examined carefully and either tested against real-world data or explored well enough to show that within a reasonable set of choices the model is robust to changes. Second, the vast number of outputs makes it easy for a model author to become lost in all the data that the model generates. Moreover, it makes it difficult to extract clear patterns of behavior. Often model authors will need to look at many different relationships between the input and output data before they are able to find a salient pattern of behavior that is compelling.

The four distinct formats of ABM data that we have talked about in this chapter are: (1) statistical, (2) graphical, (3) network-based, and (4) spatial. Statistical results are standard model output: means, standard deviations, medians, and other methods of analyzing the values of a variable. Graphical results are an outgrowth of statistical results; they transform statistical results into graphs that can be more easily examined by the observer. Network-based results, like cluster analysis and path length examinations, are another particular way of analyzing data that is often useful in ABMs. Finally, spatial results address the analysis of patterns of variables in a one-, two-, or higher dimensional space, and they frequently address questions regarding the pattern of data in the space.

However, these four formats of data output can also be used as data input. As we have seen, this is very clear in the case of network and spatial data. In the network variant of the Spread of Disease model we initialized the model using network properties (e.g., the number of connections per node). We then ran the model and used the output of the network data in combination with the model measures, such as time to 100 percent infection, to describe the model. In the spatial case, we initially start with a world in which there is no infection except in one location, but it would be a simple extrapolation to “seed” the world with multiple pockets of environmental infection. We can also use statistical and graphical data as input to the model. For instance, when we set the parameter of DISEASE-DECAY to 4, we are really setting the parameter to a mean of 4 with a standard deviation of 0. We could add another parameter to control the standard deviation as well, and then whenever a new agent becomes infected they might have a slightly slower or faster DISEASE-DECAY because it would be generated using the “random-normal” primitive instead of being set to exactly 4. This could indicate whether that individual practices hygiene habits or participates in infectious behavior. Finally, we can also use graphical data as model input. Graphs can embody equations, and we might place equations within agents to govern their behavior. For instance, rather than having the DISEASE-DECAY be constant based on the agent, we could make it a variable based on the time since they have been infected (e.g., DISEASE-DECAY = $e^{\text{TIME-SINCE-INFECTION}}$), indicating that the longer an individual has been infected, the more infectious he becomes. By using all of these results and inputs together, we can obtain a better understanding of how any model works, and by understanding the model we gain a deeper understanding of the phenomenon we are modeling.

Explorations

1. In all of the Spread of Disease models that we discussed, a contact results in an infection, but in reality diseases do not often spread based on one contact. Instead, there is usually a probability of a contact resulting in a spread. How would you modify the model and its variants described earlier to account for the probability of disease spread?
2. In the experiments discussed in this chapter, we discussed how density of individuals affects the Spread of Disease; however, there is also the speed with which individuals move throughout a landscape. One might hypothesize that if individuals move faster, that is the same as there being more individuals. What arguments exist for and against this hypothesis? How can you construct an experiment to test this hypothesis?
3. In the environmental model variant, the disease impact left behind by an individual dissipates at a constant rate (10 time steps). One could instead imagine that the disease diffuses through the local environment and as the concentration falls below some critical level, it becomes impossible for the disease to still be infectious. How would you model the phenomenon instead of the current constant rate of dissipation?
4. The social network variant, the mobile/spatial variant, and the environmental variant described earlier were all separate variants. However, in many cases of the spread of real diseases, all of these factors interplay. How would you modify the models described so that they took into account social networks, random meetings, and environmental effects?
5. Throughout this chapter, we have modeled the spread of disease. Researchers have hypothesized that innovations spread in similar ways to disease. How would you modify our model to model the diffusion of innovation instead of disease? What about the spread of rumors or urban legends? Carry out one of these modifications and analyze the results.
6. This class of disease-spread models is also related to percolation that we discussed before. Compare and contrast the Spread of Disease model and the Fire model that we discussed in chapter 3. How can you modify the Spread of Disease model so that it represents a forest fire instead of the spread of an infection?
7. The focus of this chapter has been on measuring the results of ABM. Why is measuring a result different in ABM as compared to classical science? What advantages and disadvantages does the multitude of results and inputs have for ABM as compared to classical experiments?
8. *Dynamic networks* In the Spread of Disease model, when we switched from a spatial relationship between agents to a network-based relationship, we also switched our output measure. In the spatial case we were measuring the time to full infection given the population size, but since in the case of a network we may not have full infection, we had to

switch to measuring the number infected after a certain time period for a given average degree. How can we compare these two numbers? Why is the degree of a node in the static case not the same as the number of individuals contacted every time step in spatial case? Design a measure that describes a dynamic degree. Use this measure to compare the results of the two models. Does the Network model or the Spatial model result in faster infection? Why?

9. *Testing parameter spaces* The Fur model in the Biology section of the models library can generate a lot of different patterns. For instance, you can create both horizontal and vertical stripes by manipulating the four parameters that control the repulsion and attraction radii. Find all the sets of parameters that will create at least one strip that of the nondominant color that goes all the way around the world. Keep in mind that this model is nondeterministic. Hint: It might be easier to first create a new measure then create a BehaviorSpace experiment to explore this space.

10. *Spread runs* Sometimes it can be useful to examine a group of runs at once. Run the Spread-of-Disease model one hundred times and plot the number of infected versus time on the same graph. What does this Spread Run graph tell you that a single instance does not? What does this graph tell you that graphing the mean of the number infected does not?

11. *Batch runs* We have discussed how you can run a model multiple times from BehaviorSpace, but you can also run a model multiple times without ever opening up the NetLogo application. Read in the NetLogo documentation about the Controlling API about "headless" running. Run the simple Spread of Disease model this way multiple times and collect the results in a single graph showing time versus number infected. What are the advantages to running your simulation in this manner?

12. *Language change* Rewrite the Spread of Disease model as a Language Change model. Think of the infection as a new way of pronouncing a word. Whether or not an individual adopts the new pronunciation depends on how many of the people he or she interacts with use the same pronunciation. This is in contrast to an infection model where there is a probabilistic chance of infection based on every contact. How does this new infection method affect the results?

13. *Birth and death* Some diseases are fatal. Add birth and death to the Spread of Disease model, but make the death of an individual dependent on how long he or she has had the disease. Is it possible to adjust these birth and death rates such that the disease persists but does not kill off all of its hosts?

14. *Recovery* The model that we have created is what is called an SI (Susceptible and Infected) model. Modify this model to create an SIR (Susceptible, Infected, and Recovered) model. Add a third state to the agents where after individuals become infected they have a chance of becoming recovered. Recovered agents are immune to the disease and cannot become re-infected. Describe the results of the new model.

15. *Different distributions* In the results of the Spread of Disease model that we have presented herein, we have described the statistical distributions of the results using a mean and a standard deviation. This is fine if the results are normally distributed, that is, they all fall around a central mean, with most of them being closer to that mean and fewer of them being far from the mean. However, some results are better described as two groupings of data instead of just one, and thus the results can be more naturally divided into subgroups. What parameters would you use to describe these types of results? When looking at data, how would you know to split them into multiple groups? Describe a general-purpose method that will enable you to take a raw set of data and determine the number of means that it takes to adequately describe the data.

16. *Different thresholds of infection* In the current model every individual has the same threshold of infection. In fact, this threshold is a constant and cannot be changed by a parameter. In the current model, as soon as an individual is in contact with an infected individual, he becomes infected himself. Change this so that different individuals have different thresholds of infection. There are at least two different ways to do this. One way would be to have each individual have a probability of becoming infected every time he comes in contact with an infected individual. Another possibility is to have individuals have to contact at least *x* infected individuals before they become infected themselves. Implement both of these methods. Is there any difference in the results? Describe why this difference is or is not meaningful.

17. *Time series analysis* We discussed how time series analysis can be used to examine data that is time-dependent. The typical way this is done is by describing a relationship between time and some input parameters. For instance, fraction-infected(t) = population/ $(1 + e^{-at})$ creates a graph that looks somewhat like the increase in the fraction-infected over time, but this function must be tuned to more closely approximate the results of our model. Create a function that represents the change in fraction-infected over time as a function of the population, for the original “mobile” Spread of Disease model. Describe your function. Highlight which areas of the graph it more closely matches and which areas it does not.

18. *Clustering coefficient and average path length* We discussed how the clustering coefficient and average path length of a network model can help us to analyze a network. Create reporters for both of these in the network variant of the Spread of Disease model. Examine the relationship between these values and the mean number infected after fifty ticks.

19. *Mean patch size and edge ratio* Similar to the previous exploration, the mean patch size and edge ratio of geographical systems can contribute to an understanding of the system. Implement these reporters in the environmental variant of the Spread of Disease model. Do these measures change as you vary the disease-decay time? If so, describe how they change. If they do not change, describe why they do not change.

20. *Exponential decay of disease* At the end of the chapter, we talked about how you can give agents rules that are equations. Implement an exponential decay of disease model in the patches. How does this change compare with the original environmental variant of the Spread of Disease model?

21. *Other types of measurements* Throughout this chapter, we measured the number of individuals infected and the time to 100 percent infection. Create another measure that may be of interest to someone studying the spread of disease. Why did you choose this measure? Explain why it would be useful to someone interested in this subject.