

# Lab 11

## Simulations

In this lab you'll learn how to create simulations to provide approximate answers to probability questions. We'll make use of a particular kind of structure, called a box model, that can be used to simulate a wide range of probability problems.

### The Concept

Probabilities are long-run frequencies. When we say something like “the probability that event **A** will occur is 38%,” what we mean is that there is an experiment, and one of the possible outcomes of this experiment is the event **A**. If we were to repeat this experiment infinitely many times, the event **A** would occur in 38% of them.

For example, when we say that the probability of getting **heads** when we flip a coin is 50%, we mean that if we flip the coin infinitely many times, half of the flips would land **heads**.

No one has the patience to flip a coin infinitely many times, even if they could live long enough. The best we can hope for is to see the outcome of a large number of flips. Fortunately, computers make it possible to repeat an experiment many times, very quickly. Rather than flip a real coin 1000

times, which takes quite a bit of effort, we can program a computer to do the same in the blink of an eye. Or faster.

Suppose we program a computer to do virtual coin flips. Suppose everyone in the room flips a virtual coin 1000 times. Will everyone get the same number of heads? Almost certainly not. (But it's possible!) If everyone gets a different amount, how can we use our simulation to figure out a probability? The answer is that the best we can do is get an approximation.

Probabilities determined from simulations or from actual experiments are called **experimental probabilities** or synonymously, **empirical probabilities**. An experimental probability differs from person to person and from attempt to attempt. The important thing to keep in mind is that the more trials you do, the more likely you are to get an experimental probability that is close to the theoretical probability. To sum things up, your experimental probability is an approximation of the theoretical probability and the larger the sample size, the better the approximation.

In other words, if everyone in the room flips a real coin 20 times, the percentage of heads will vary quite a bit and might stray quite a ways (relatively speaking) from 50%. But, if everyone flips a real coin 1000 times, each person's percentage of heads will be closer to 50%.

To do our simulations, we need a simulation "machine." Our machine is called a box model. It is a mental abstraction, but it is also a routine programmed into *Stata*. This routine allows you to program the computer and order it to carry out certain types of simulations.

## What's a box model?

A box model is a box with slips of paper in it, like you might see in a raffle. The slips of paper have numbers on them. We reach into the box, pull out a slip and record the number. We could repeat this many times to get an idea of the approximate probability of seeing certain values.

Let's make a box model for estimating the probability of getting a "3" when rolling a fair die. The first step is to decide what values go on the tickets.

We need to make sure that all possible outcomes are represented. Therefore, we will need one ticket with a 1, one with a 2, another with a 3, and so on up to 6. The second step is to decide how many of each ticket. For a fair die, each outcome is equally likely, so each value in the box should have equal representation. Therefore, we should have the same number of tickets for each value. The simplest way to do this is to have each value appear exactly once. If we wanted to though, we could have 100 tickets for each value.

Once the box model is “created,” we are ready to use it. For this “thought” experiment, imagine mixing up the tickets and then reaching into the box to draw one out. If it’s a 3, we make a note of this. If not, we do nothing. No matter what, we put the ticket back, shuffle the box, and repeat. We repeat this many times, maybe 1000. When we’re done, our “experimental probability” is the number of 3’s divided by the number of trials.

Note that there’s another way of drawing out tickets. We could, if we wanted to, draw a ticket out, record its number, and then *throw it away*. This is called **sampling without replacement**. It is useful for experiments in which outcomes can only occur once, such as selecting lottery numbers.

## Using *Stata* to make and run box models

Before we get started, we need to do a couple things.

**\*\*\* Important \*\*\***

First, you need to change your working directory. Click on the File menu and select “Set Working Folder.” The default is your “Documents” folder. This is the correct location, so click on the “Choose” button.

Second, to ensure we truly are taking random samples, we want to randomly set a seed for *Stata* to start from.

```
. set seed YOUR STUDENT ID NUMBER
```

We are not going to use a real box model. Instead, we will use a virtual box model. A model box model, if you will. Our virtual box model exists within some *Stata* code. Don't worry, you don't have to write any code. You just have to execute the program that will then guide you to build a box model.

The program that does this is not part of the usual *Stata*,<sup>1</sup> and so if your computer does not already have this installed, you need to install it. If you are doing this in the Statistics lab, then it has probably already been installed. Type

```
. bxmodel
```

Does anything happen? If you get an error (*unrecognized command*), then you need to follow the instructions below to install it. Otherwise, hit *cancel* and skip past the installation steps.

*Commands needed for installing bxmodel*

```
. net from http://www.ats.ucla.edu/stat/stata/ado/teach  
. net install bxmodel
```

Using the *bxmodel* program requires three steps.

- Create the “box” with values and frequencies
- Choose the type of sampling and number of repetitions
- Analyze the output

To create the box, you will need to use the editor to create a data set with two columns or variables. The first variable represents the different values the experiment can take and the second variable represents the “number of tickets” each value should have in the box.

---

<sup>1</sup>Full directions for using the box model are available at <http://www.ats.ucla.edu/stat/stata/ado/teach/bxmodel.htm>

```
. edit
```

To correctly use the *bxmodel* program, the name of the first variable must be **value** and the name of the second variable must be **n**.

To simulate the roll of a fair die, put the numbers 1 through 6 in the **value** column. Each value should only appear once “in our box,” so put a 1 beside each value in the **n** column.

Click on “Preserve” and then close the editor. Finally, choose “Save as” under the File menu and save under a memorable name, such as, *fairdie.dta*.

Now you are ready to run the box model. In the command window, type

```
. bxmodel
```

A dialogue box will appear.

**Enter file name:** The name of the file where you saved your box model.

**Number:** How many tickets you want drawn out of the box.

**Type of Box Model:** The possible model types available. (We have briefly discussed the sampling with replacement and without replacement options. The birthday options will be discussed later.)

**Repetitions:** The number of times you want to repeat the experiment or how many samples you want to draw.

The program does not save the raw results. Instead, for each *repetition* or sample, it saves the sum of the tickets in the sample, the average of the tickets in the sample, and the standard deviation of the tickets in the sample. In other words, if you set up a box model and told it to do 100 repetitions, you would end up with 100 different sum totals, 100 different averages, and 100 different standard deviations.

Getting back to our dice problem. . .

*Question 1: What is the theoretical probability of rolling a 3 on a fair die?*

Remember, our goal is to simulate rolling a fair die. We want to “roll” the die one time, record the result, and repeat. For this experiment, we want to do this 100 times.

The *bxmodel* input should be the file we created previously, the fact that we only want to roll the die once (or analogously, we want to select one ticket from the box), sampling with replacement (the number 6 is not removed from a die once it has been rolled), and the number of times we want to repeat the experiment.

**Enter file name:** fairdie.dta

**Number:** 1

**Type of Box Model:** Sum, mean, sd - w/ replacement

**Repetitions:** 100

Look at the output that the *bxmodel* gives you.

```
. list
```

You should see three variables, **sum**, **mean**, and **sd**. The variables **sum** and **mean** contain the same information. This is because we only rolled the die once or drew only one “ticket.” The **sd** variable is full of missing values. You can not compute a standard deviation for a sample of size one. We have 100 observations, each representing a different roll of the die.

To summarize the output of the *bxmodel* program, we can use the **tabulate** command.

```
. tabulate sum
```

*Question 2: What is the experimental probability of rolling a 3? How far off is this from what you expected? Does this seem reasonable?*

The empirical probability density function should be similar to the theoretical probability density function. In this case, the probability of each outcome

is  $1/6$ , and so we expect each of the 6 outcomes (1,2,3,4,5,6) to occur about  $1/6$ th of the time. We can verify this by looking at a graph of the empirical probability density function.

```
. histogram sum, discrete width(1) xlabel(1 2 to 6)
yline(.16667)
```

*Question 3: What is the empirical probability density function you obtained for rolling a fair die?*

Empirical probabilities will change with every experiment we do. To examine this phenomenon, we will repeat the above experiment 10 times.

Note: Because of a quirk with the *bxmodel* program, we have to remove the file that the program created before it will correctly run again. You have to do this every time you want to run the *bxmodel* program.

```
. erase __000001.dta
```

*Question 4: Repeat the above experiment (100 repetitions of rolling a single die) 10 times. Each time, calculate (i) the experimental probability of rolling a 3 and (ii) the difference between the experimental probability and the theoretical probability.*

As you have seen, variation occurs in each repetition of the experiment. This is a direct result of it being a random process. We can change the experiment in a number of ways. For example, we can roll the die many more times. In a moment, we will repeat the experiment, but this time rolling the die 10,000 times!

*Question 5: You calculated the difference between the experimental probability and the theoretical probability of rolling a 3. How should the differences you compute for the 10,000 rolls compare to the differences you computed for the experiment with 100 rolls?*

*Question 6: Now do 10,000 repetitions of rolling a single die. Repeat 10 times. Each time calculate (i) the experimental probability of rolling a 3 and (ii) the difference between the experimental probability and the theoretical probability.*

To play the popular casino games of *craps*, you roll two dice. The outcome is based on the sum of the dice. The game places special significance on an outcome of 7. (If a 7 occurs on the first throw, this is usually good for the player. Otherwise 7's are usually bad news.)

*Question 7: What is the theoretical probability of rolling a 7? Simulate the experiment of throwing two dice together 1000 times. What is your experimental probability of rolling a 7? How does this differ from the theoretical probability?*

In the game of craps, the payoffs (the money the casino gives you if you win) are proportional to the probability of certain outcomes. Inversely proportional, of course! The less likely the outcome, the greater the payoff.

*Question 8: According to the experiment you just did, which outcomes have the biggest payoffs? Does this match with your theoretical expectations?*

In the card game *poker*, you are dealt a “hand,” that consists of 5 cards. To vastly oversimplify the game, the more rare your hand, the better of an advantage you have over your opponents. One rare hand is to have all of your cards the same suit. (There are four suits: hearts, diamonds, clubs, and spades.)

Create a new “box” with values and frequencies and use it to estimate the probability that all 5 cards will be spades.

*Hint: There are 52 cards in a deck and 13 in each suit. Assign the value “1” to spades and “0” to all the other suits. A hand consists of 5 cards dealt*

*without replacement.*

First, do the experiment with 1000 repetitions. Try this two or three times. Then try a few times with 10,000 repetitions.

*Question 9: What experimental probabilities did you find in your experiment? Based on your results, can you guess what the theoretical probability is? Why are so many repetitions necessary?*

*Question 10: Calculate the theoretical probability. Was your experimental probability close to this?*

## The Birthday Model

*In a classroom of 20 people, what are the chances that at least two will have the same birthday?*

We can use the *bmodel* program to estimate probabilities such as this. The first step is to imagine that each day has a number. (The second step is to ignore leap year day.) Next, we make a crucial assumption: that birthdays are uniformly distributed.<sup>2</sup> In other words, a person is just as likely to have his or her birthday on one day as any other. We can imagine that each person goes through life with a number between 1 and 365 assigned to them. When we assemble 20 people in the classroom, it's as if we just randomly selected, with replacement, 20 numbers from a box that has the numbers 1...365 in it.

A single repetition consists of drawing 20 tickets from the box, with replacement. But this time, it won't help us to record the sum or mean of the tickets. Instead, we want to know how many of the tickets are identical. If

---

<sup>2</sup>Is this assumption true? Check out the *Stata* Coding and Birthdays Lab.

two of the twenty people have the same number, then they have the same birthday!

The first step is to create a data file with two variables. The variable labeled **value** will have the numbers 1 through 365, and the variable labeled **n** will consist of a string of 1's. (Each value appears in the box exactly once.)

You may be thinking that it will be rather tedious to type all 730 numbers into the *Stata* editor. Luckily, *Stata* provides us with a better way. Type the following commands:

```
. clear
. set obs 365
. generate value = _n
. generate n = 1
```

The first command clears the memory. The second command tells *Stata* to save space for 365 observations. Next, the generate command creates a new variable named **value** that is given all of the values from 1 to n (where n is equal to the number of observations, 365 in this case.) The final command creates a new variable called **n** and gives it a value of 1 for all observations.

Save this data set as *bday.dta*

Now to answer the question proposed at the beginning of this section, run the *bxmodel* program, using the following input:

**Enter file name:** bday.dta

**Number:** 20

**Type of Box Model:** Birthday - w/ replacement

**Repetitions:** 1000

Look at the data:

```
. list
```

This time there are only two variables, **match** and **unique**. From glancing at the data, we can see that **match** seems to equal either 0 or 1, whereas

**unique** has values like 19, 20, and 18. For each repetition, **match** is 1 if any two or more tickets in the 20 selected match, 0 if there are no matches. The variable **unique** counts the number of tickets in those selected that are unique.

If there are two people with the same birthday, **match** will be 1 and **unique** will be 19 (since there are 19 unique numbers.) If 3 people share the same birthday, **match** will still be 1, but **unique** will be 18. If there are two pairs with the same birthday (i.e., two people were born on, say, day 34, and two more were born on day 165), then **match** will be 1 and **unique** will be 18. If none of the 20 shares a birthday, **match** will be 0 and **unique** will be 20.

We can estimate the probability of a match occurring by counting how many of the repetitions had a match. The experimental probability is the number of repetitions for which **match** is 1, divided by the number of repetitions. *Stata* calculates this as the mean of **match**.

```
. summarize match
```

*Question 11: In a classroom of 20 people, what are the chances that at least two people will have the same birthday?*

*Question 12: How many people are needed in a classroom for the chance that at least two have the same birthday to be over 90%*

The set up for the birthday problem can be used to solve other types of problems.

A children's cereal is having a special promotion. Each box of cereal has one of seven tokens inside. If you collect all 7 tokens, you send them in and receive a Big Prize! Assume the tokens are evenly distributed, so that if you select a box at random, tokens have an equal chance of appearing. Create a box to simulate this promotion.

*Question 13: Suppose you buy 7 cereal boxes. What is the probability you will get all 7 tokens and win the Big Prize?*

*Question 14: How many boxes must you buy for the probability of getting all 7 tokens to be at least 50%?*