GAMING SIMULATIONS IN PROBABILITY

Julie Belock
Salem State College
Department of Mathematics
Salem, MA 01970
julie.belock@salemstate.edu

Games such as Bingo and craps provide a rich source of problems for a probability course. These games are familiar to many students, and are easily described to the uninitiated. I have developed a series of activities based on these games to reinforce a number of topics for the students. These activities combine simulations, theoretical computations, research into published results and a project consisting of a report on the students' findings.

Many of the questions would involve tedious computations to solve analytically, so simulations are introduced. I have used simulations on the TI-83, Excel and Minitab, depending on the problem. Solutions obtained by simulation are compared to the corresponding theoretical result whenever possible. Sometimes the students obtain these theoretical results; at other times, we rely on published solutions if the theoretical computations are beyond the scope of the course.

Below I describe the games of craps and Bingo; for each, I list typical probability questions one could ask and give the code of a TI-83 program for simulations to answer those questions. The theoretical computations are left as exercises for the reader (I give references for them as well).

Craps

In the game of craps, a pair of dice is rolled. If the sum of the numbers on the pair is a 7 or 11, the player wins. If the sum is a 2, 3 or 12, the player loses. If the sum is any other value, that value is designated the "point" and the player rolls again, until either the point comes up, resulting in a win, or until 7 comes up, resulting in a loss.

Students quickly observe that it is possible for the game to continue infinitely. Some natural questions that arise include:

- What is the probability of winning?
- What is the probability that the game ends after only one toss of the dice?
- On average, how many tosses does it take to end the game?

A TI-83 program for simulating the game of craps is given below. The program takes as input the number of games to play, and it reports number of wins and losses so that the students can compute the experimental probability of winning based on the entered number of trials. Also, the number of rolls required for each game to end is exported to a list, so that the average number of rolls can be computed.

Program CRAPS

```
ClrList L6
Disp "NO. OF GAMES"
Prompt N                          (Enter the number of games)
N →dim(L6)
0 → W                             (Initialize the # of wins and losses to zero)
0 → L

For(J,1,N)                        (Start of the loop for N games)
0 → K
randInt(1,6) → X
randInt(1,6) → Y                  (Roll the dice and take the sum)
X + Y → S
K + 1 → K                         (Count 1 roll)
If S = 7 or S = 11
Then
W + 1 → W                         (Win on the first roll)
Else
        If S = 2 or S = 3 or S = 12
        Then
        L+1 → L                   (Lose on the first roll)
        Else
                S → P             (Otherwise, set the sum as the point)
                0 →D              (Set counter to 0)
                While D < 1
                randInt(1,6) → X  (Roll again)
                randInt(1,6) → Y
                X + Y → S
                K + 1 → K         (Count another roll of the dice)
                        If S = P  (If you roll the point, you win.)
                        Then
                        W +1 → W
                        D + 1 → D
                        Else
                                If S = 7
                                Then
                                L + 1 → L   (If you roll 7, lose)
                                D + 1 → D
                                End
                        End       (Otherwise keep rolling)
                End
        End
End
K + 1 → L6(J)                     (Record the number of rolls for game J in L6)
End
Disp "WINS", W
Disp "LOSSES", L
```

(end of program)

The output screen for a simulation of 10 games of craps is shown below:

```
NO. OF GAMES
N=?10
WINS
                 4
LOSSES
                 6
              Done
█
```

Computing the theoretical probability of winning a game of craps involves infinite geometric sums. The expected number of tosses required to end the game is computed using conditioning and the geometric distribution, as described in [2], among other places. If the students in the course have the background, these computations may be assigned to the students or worked out in class, and then compared to the simulated probabilities.

Bingo

A Bingo card is a 5x5 array of positive integers, in which the first column (B) contains integers between 1 and 15, the second (I) between 16 and 30, the third (N) between 31 and 45, the fourth (G) between 46 and 60 and the fifth (O) between 61 and 75. The entry in the third row, third column is designated a "free space." A caller selects numbers between 1 and 75 at random and without replacement and announces them to the players, and if a player's card contains the number, he covers it. A Bingo occurs (and the game ends) when a player has covered all the positions in a row, column or diagonal on his card.

Some natural questions that arise include:

- How many calls does it take, on average, for a Bingo to occur?
- What is the probability that after *k* calls, your card has *n* squares covered?
- What is the probability that after *k* calls, there is at least one Bingo?

A TI-83 program given below simulates the probability of at least one Bingo occurring in a given number of calls. The program takes as input the number of trials (games) and the number of calls and returns the number of games with at least one Bingo for a game with only one card. The card must be entered into matrix [A] in advance. To generate a sample from {1, ….,75} *without replacement,* calls program SAMPLEB as a subroutine. This program is a slight alteration of a program found in [4]. To use that program, integers 1, . . . , 75 must be entered into L1.

Program "BINGO"

```
Input "HOW MANY TRIALS?", M
0 → D
Input "NO. OF CALLS", C
```

```
For (G, 1, M)                  (Starts the loop for M trials)
prgm SAMPLEB                   (calls the program to generate a random sample w/out
                               replacement)
Fill (0,[B])                   (Fills matrix B with 0's)
1→[B](3,3)                     (Fills the free space with a 1)

For (K, 1, C)          (for each number called)
        If L2(K) ≤ 15
        Then
                For(I,1,5)
                        If L2(K)=[A](I,1)
                        1 → [B](I,1)
                End
        Else
                If L2(K)>15 and L2(K) ≤ 30
                Then
                        For(I,1,5)
                                If L2(K)=[A](I,3)
                                1 → [B](I,2)
                        End
                Else
                        If L2(K)>30 and L2(K) ≤ 45
                        Then
                                For(I,1,5)
                                        If L2(K)=[A](I,3)
                                        1 → [B](I,3)
                                End
                        Else
                                If L2(K)>45 and L2(K) ≤ 60
                                Then
                                        For(I,1,5)
                                                If L2(K)=[A](I,4)
                                                1 → [B](I,4)
                                        End
                                Else
                                        For(I,1,5)
                                        If L2(K)=[A](I,1)
                                        1 → [B](I,5)
                                        End
                                End
                        End
                End
        End
End

0 → X
0 → Y
1 → J
```

```
While J < 6 and(X < 5 and Y < 5
        [B](1,J) + [B](2,J) + [B](3,J) + [B](4,J) + [B](5,J) →X   (adds all ones in column J)
        [B](J,1) + [B](J,2) + [B](J,3) + [B](J,4) + [B](J,5) →Y   (adds all ones in row J)
        J + 1 →         J
End

        If X = 5 or Y = 5
        Then
        D +1 → D
        Else
                [B](1,1) + [B](2,2) + [B](3,3) + [B](4,4) + [B](5,5) →P
                [B](1,5) + [B](2,4) + [B](3,3) + [B](4,2) + [B](5,1) →Q
                        If P = 5 or Q = 5
                        D+1 → D
        End
End             (ends the loop for all M trials)
Disp D
Float           (return to floating point decimal)
Disp D/M        (computes the experimental prob. of at least 1 bingo in C calls for M trials.)
```

(end of program)

Below is an output screen for an example for a simulation of 10 games. The probability of at least one bingo in 35 calls for a single card is 0.3 in this example.

```
prgmBINGO
HOW MANY TRIALS?
10
NO. OF CALLS?35
               3
              .3
            Done
```

The theoretical probability of at least one bingo in 35 calls is 0.272 (for a single card) as computed on [5]. Since the computation of theoretical probabilities in Bingo tend to be tedious, I usually refer students to published sources, such as [1] and [5]. When multiple Bingo cards are involved computing the probabilities becomes even more complicated, since the cards are not independent; a discussion is included in [1].

References:

1. Agard, D.B., and M.W. Shackleford. A new look at the probabilities in Bingo. *The College Mathematics Journal* **33** (4) 2002. pp. 301-305.
2. Catlin, D. How long is a craps roll? *Casino City Times* http://catlin.casinocitytimes.com/articles/1240.html 2000. (10/25/03)
3. Ross, S. *A First Course in Probability,* 6th ed., Prentice Hall, Upper Saddle River NJ, 2002.
4. Starnes, D. Technology Tips. *Mathematics Teacher* **93** (8) 2000. pp. 720 – 721.
5. "Durango Bill's Bingo Probabilities". http://www.durangobill.com/BingoHowTo.html (10/13/03)