Playing with Dice

Rolling Dice

Probability gives a scientific foundation to our guesses. It actually originated many years ago in trying to understand games of pure chance, like throwing dice. Many of the mathematicians involved in its original developments were either gamblers themselves or had an ongoing correspondence with well-known mathematicians. Probabilistic simulations is a very important area of research and is used not just for understanding your chances of winning at gambling, but for many other purposes like testing hypothesis, etcetera.

As said, probability started as an empirical science, geared toward answering question posed by gamblers: the typical concern was to understand if a wager was favorable. Most of their games were games with dice. We don't have enough dice to play with, so we will be simulating dice throws with out TI-84 calculators. We will simulate the throw of a die using the random number function `rand` of the TI. This is the function that you find by pressing [MATH], then pressing [◄] to select `[PRB]`, and then pressing [1] to select `[1:rand]` and place `rand` on the HOME screen.

When executed, this function `rand` produces a "random" number > 0 and < 1. It is a 10 digit decimal like `.7455607728` Since a throw of a die produces a whole number between 1 and 6, something must be done to this `rand` for it to be used in a game of dice.

¶ **1.** Modify the function `rand` so that its output is one of the numbers `1, 2, 3, 4, 5, 6`

One possible way is to use the combination

```
int(6*rand)+1
```

When we throw a die, each of the six possible outcomes is an elementary event. Typically, we assume that each of the elementary events is equally likely, and we assign to it a number $P(\text{event})$ between 0 and 1, a qualitative measure representing the chances of that event to happen; the closer the number $P(\text{event})$ is to 1 is interpreted as that event being more likely to occur in an experiment. In the case of a fair die, we have six elementary events, and each receives a probability $P = 1/6$. The sum of the probabilities of all the elementary events is 1, just because the outcome of any experiment must be one of the elementary events. Obviously, for a die,

$$\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}=\frac{6}{6}=1.$$

The probability of an event represents the frequency which which we should observe that event as outcome of multiple repetitions of the experiment. A mathematical theorem, the Weak Law of Large Numbers, give a precise content of this intuition.

But we will leave that for future occasions. We will test our TI-84 simulated die. For that, we will run a program that pick a random integer from 1 to 6 and then plot the frequencies in a histogram.
First we set up the graphing part: pressing [2nd] [STAT PLOT] take you to the STAT PLOTS configuration screen. Select `1:` and press [ENTER] . Select `On` and press [ENTER] . Move down to `Type:`, select the histogram and press [ENTER] . Move down to `Xlist` and press [2nd] [1] so that the list L_1 is the x-variable. Finally, move down to `Freq` and press [2nd] [2] so that the list L_2 is the list of frequencies (the heights of the bars of the histogram). Press [2nd] [QUIT] to returm to the HOME screen.
Next we set the display window dimensions. Press [WINDOW] and edit your current values as follows:

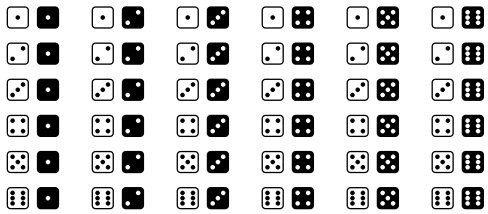
```
WINDOW
 ShadeRes=3
 Xmin=0
 Xmax=6
 Xscl=1
 Ymin=0
 Ymax=100
 Yscl=1
 Xres=1
```

Now we run the program that throws a die a number `T` of times and stores the results in a list L_2. Then the program will display the frequence with which each of the faces of the die appeared in the sequence of `T` throws. Pressing [ENTER] will make the program to display those frequencies in a histogram.

```
PROGRAM:DIE
:Input ‘‘THROWS=’’, T
:ClrList L1
:ClrList L2
:6►dim(L1)
:6►dim(L2)
:For(J,1,6)
:J-1/2►L1(J)
:End
:For(I,1,T)
:int(6*rand)+1►R
:L2(R)+1►L2(R)
:End
:ClrHome
:For(I,1,6)
:Disp L2(I)/T►Frac
:End
:Pause
:DispGraph
```

Two dice

It appears that our die simulation works quite well: each of the 6 possible outcomes appears about 16% of the times. Now we are ready to roll two dice. For each throw of two dice there is a pair of numbers from 1 to 6, one for each of the die. There is then a total of 36 possible elementary outcomes:



Often we do not want to just understand elementary event, but combinations of them: random variables. One such example is the sum of the two numbers when throwing a pair of dice. There are 11 possible values for that sum: `2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12`
If you throw a pair of dice 100 times, which value for the sum of the faces will you think occur more often? If you think naively, you will come up with 1/11 for each of the eleven possible sums, but that is the wrong answer. Some values for the sum actually occur more often than other: for example, there is only one way that you can get a sum of 12, and that is with a pair of 6. However, you can get a 7 with a 1 and a 6, with a 2 and a 5, or with 3 and a 4.

¶ **2.** Fill out the following table with the sums of the two dice. It will be easier if you think that you have a white die with black dots and a black die with white dots; the horizontal numbers represent the outcome of the white die, and the vertical those of the black die.

¶ 3. Fill in the following table of probabilities for the sum of the faces when rolling a pair of dice:

Result	2	3	4	5	6	7	8	9	10	11	12
Probability											

As you can see on this table, some values for the sum have higher probability than others. It is a great project to understand this by simulating an experiment with the TI-84. First we set the values for the [WINDOW] dimensions. The x-variable will range from `Xmin=1` to `Xmax=12` in increments of 1 with the bars centered at 1.5, 2.5, and so on; that is, the bar representing the sum 2 is over the interval $[1, 2]$, and so on. (This is not important, as we are only interested in the height of the bars).
The new program for the sum of throwing a pair of dice is similar to the one of rolling one die, and so we may just copy and edit it as follows. Edit a new program by pressing [PRGM] and [◄] to select `[NEW]` and [ENTER]. Call this new program `DICE` and hit [ENTER]. Your screen should display

```
PROGRAM:DICE
:
```

Press [2nd] [RCL] [PRGM], then select `[EXEC]` by pressing [◄], then press [▼] until the program `DIE` is selected. Press [ENTER] and your program `DIE` will be copied into the new program `DICE`. You should now edit it as indicated below. Note that the dimensions of the list are 11 because there are only 11 possible results for the sum of two dice.

```
PROGRAM:DICE
:Input ‘‘THROWS=’’, T
:ClrList L1
:ClrList L2
:11►dim(L1)
:11►dim(L2)
:For(J,2,11)
:J-1/2►L1(J)
:End
:For(I,1,T)
:int(6*rand)+1+int(6*rand)+1►S
:L2(S-1)+1►L2(S-1)
:End
:ClrHome
:For(I,1,11)
:Disp L2(I)/T
:End
:Pause
:DispGraph
```

Gambling with Dice

Chevalier the Mérè's (1607–1684) principal occupation was wagering in games of chance. He was well aware that with a single die, four throws suffice to make it worthwhile to bet on obtaining at least one 6, but with two dice, 4 times as many throws do not suffice to make it worthwhile to bet on obtaining a double 6. He corresponded on this with Blaise Pascal and with Pierre de Fermat, two of the greatest mathematicians of the 17th century and founders of combinatorial probability, to whom he posed the following problems.

¶ **4.** If one wagers to roll at least one 6 in four throws of a die, then a win is more likely than a loss.

The following TI-84 program simulates an instance of the game played by de Mérè. The player wins if at least one 6 is obtained in four throws of a die. At the end of the game, the variable `W` will be `1` if the player wins, and `0` if the player loses.

```
PROGRAM:MERE
:ClrHome
:Disp ‘‘FOUR THROWS’’
:0►W
:For(J,1,4)
:int(6*rand)+1►R
:Disp R
:If R=6
:1►W
:End
:If W=1
:Then
:Disp ‘‘YOU WIN’’
:Else
:Disp ‘‘YOU LOSE’’
:End
```

The math is as follows. With four throws of a die there are $6 \times 6 \times 6 \times 6 = 1296$ possible outcomes. Of these, a loss results if none of the throws is a 6, that is, if in every throw a 1, or a 2, or a 3, or a 4, or a 5 results. There is thus a total of $5 \times 5 \times 5 \times 5 = 625$ possible losing scenarios, hence $1296 - 625 = 671$ possible winning scenarios. Since $671/1296 = 0.517747$ is slightly greater than $1/2$, we conclude that we are more likely to win than to lose.
We can experiment that by making the TI-84 play a fairly large number of games and compute the frequency of wins. This program contains the program `MERE` as a subroutine. For this you do the following. Position the cursor on a new command line, press [PRGM] and then press [◄] to select `{EXEC]`. Then press [▼] until your program `MERE` is selected, and then press [ENTER]. Now you have `Rcl prgmMERE` written at the bottom of the screen. Press [ENTER] once more and you will have `:prgmMERE` in the current command line.

```
PROGRAM:MEREDIE
:Input ‘‘GAMES=’’, G
:0►X
:For(I,1,G)
:prgmMERE
```

```
:X+W►X
:End
:ClrHome
:Disp ‘‘FREQ WIN’’, X/G►Frac
```

The question that really concerned de Mérè was the following.

¶ 5. Suppose that you wager a double 6 in rolling a pair of dice in certain number of throws. What would be that number so that resulting game is favorable to you?

His math seemed to indicate that 24 games should make it worthwhile, but his results (that is, his loses) were not supporting that. The mathematicians worked out this problem and confirmed his suspicion that the game was to the player disadvantage. Indeed, with a pair of dice there are 36 possible outcomes in any one throw. Of these 36 only one results in a double 6, so there are 35 unfavorable outcomes in any one throw. When throwing a pair of dice 24 times, there are 36^{24} possible outcomes, of which 35^{24} are unfavorable. The chances of losing are then

$$(35/36)^{24} = \frac{1141913124207058038717508316040039062 5}{2245225770735455724008721112379267481 6} \approx 0.508596,$$

which is slightly larger than $1/2$, confirming de Mérè experience.
We will make the TI-84 play a such a game, and then another one that plays that game repeatedly.

```
PROGRAM:MERETWO
:ClrHome
:Disp ‘‘FOUR THROWS’’
:0►W
:For(J,1,24)
:int(6*rand)+1+int(6*rand)+1►R
:Disp R
:If R=12
:1►W
:End
:If W=1
:Then
:Disp ‘‘YOU WIN’’
:Else
:Disp ‘‘YOU LOSE’’
:End
```

Here is the program that plays Mérè’s game with a pair of dice repeatedly.

```
PROGRAM:MEREDICE
:Input ‘‘GAMES=’’, G
:0►X
:For(I,1,G)
:prgmMERETWO
:X+W►X
:End
:ClrHome
:Disp ‘‘FREQ WIN’’, X/G►Frac
```

Shooting Craps

One of the most popular casino games with dice is "Shooting Craps." In this game, a player throws a pair of dice. If the the total for the first throw is 7 or 11, the player wins at once. If the total is 2, 3 or 12, the player loses at once. Any other throw (that is, $4, 5, 6, 8, 9, 10, 11$) is called a point (2 is Snake eyes, 8 is Eighter from Decatur, 12 is Boxcars). If the first throw is a point, the player throws the dice repeatedly until throwing again the point (and the player wins) or a 7 (and the player loses).
The rules are complicated and are designed in such a way as to make the game slightly disadvantageous to the player, but not too much.

¶ **6.** What is the probability of wining on a game of craps? The answer is 0.49293, can you verify it?

Here is a TI-84 program that plays a game of craps.

```
PROGRAM:CRAPS
:ClrHome:0►W:0►L
:int(6*rand)+1+int(6*rand)+1►R
:If R=2 or R=3 or R=12
:Then
:Disp R, ‘‘YOU LOSE’’
:1►L:Goto N
:End
:If R=7 or R=11
:Then
:Disp R,’’YOU WIN’’
:1►W:Goto N
:End
:Disp ‘‘POINT IS’’, R
:0►P
:While P≠7 and P≠R
:int(6*rand)+1+int(6*rand)+1►P
:Disp ‘‘THROW’’:Pause:Disp P
:End
:If P=7
:Then
:Disp ‘‘7, YOU LOSE’’
:1►L
:Goto N
:End
:If P=R
:Disp ‘‘POINT, YOU WIN’’
:1►W
:Goto N
:End
:Lbl N
```

The next program plays as many games of craps as you wish and keeps a tally of the number of wins.

```
PROGRAM:SHOOTCRAPS
:Input ‘‘GAMES=’’,G
:0►X
:For(I,1,G)
:prgmCRAPS
:X+W►X
:End
:ClrHome
:Disp ‘‘FREQ WIN’’,X/G
```