

### Lesson Overview

Students engage in a dice game with the goal of finding the optimum strategy for winning, in this case, maximizing the possible number of points when tossing dice as long as no sixes show up in a toss. After experimenting with actual dice, students use technology to generate random integers representing the situation and explore how technology can facilitate their investigation. They investigate finding an efficient way to eliminate certain subsets of each set of random numbers (those with a six) and finding a way to accumulate the results of each simulation (typically with some guidance from their teacher). As they try different strategies, students should recognize the tradeoff between the potential for getting a large sum by tossing many dice and the increased risk of getting a six, which results in no points.

#### ***About the Lesson and Possible Course Connections:***

The activity can be used whenever students have a background in finding measures of center and elementary probability, typically in upper middle school. The problem can also be done in an algebra two or year two high school mathematics course where students think about expected value and/or examine variability within and across samples in a probability and statistics unit.

### Learning Goals

Students will be able to:

1. Recognize and capture the variability in tossing dice using simulated samples of the same size (i.e., repeated samples for the same number of dice in a toss)
2. Identify possible approaches for maximizing outcomes of a chance game
3. Recognize the value of simulation in developing understanding of a problem
4. Identify and use conditionals and accumulation commands in creating a simulation

### CCSS Standards



#### ***Statistics and Probability Standards:***

- 7.SP.A.1
- 7.SP.A.2
- 7.SP.C.6
- 7.SP.C.7B
- 7.SP.C.8C
- HSS.SP.MD.A.1 +
- HSS.SP.MD.A.2 +

#### ***Mathematical Practice Standards***

- SMP.4

### Lesson Materials

- Compatible TI Technologies:  
TI-84 Plus\*; TI-84 Plus Silver Edition\*;  TI-84 Plus C Silver Edition;  TI-84 Plus CE  
*\* with the latest operating system (2.55MP) featuring MathPrint™ functionality.*  
TI-Smartview CE software
- The\_Game\_of\_Hog\_Student.doc
- The\_Game\_of\_Hog\_Student.pdf
- The\_Game\_of\_Hog\_Teacher Notes.doc
- The\_Game\_of\_Hog\_Teacher Notes.pdf

### Background

Some games of chance involve dice. One such game is Hog. In this version of Hog, each player may toss any number of dice from one up to the total number of dice available. (In the example that follows, each player can choose from 1 to 7 dice). The number of dice a player chooses to toss can vary from turn to turn. The player's score for a turn is zero if at least one of the dice comes up with the value six. Otherwise, the player's score for the turn is the sum of the faces showing on the dice. (Tossing a six gets a zero just for that toss not for the total cumulative score for the player.) A cumulative running total of the scores is kept for each player.

If you can choose to toss anywhere from 1 to 7 dice, what number of dice should you choose to win the game?



### Facilitating the Lesson

Give students sets of dice and have them spend five or ten minutes tossing different numbers of dice and reflecting on the number of points each toss would earn. The idea is not to develop a mathematical approach to solving the problem but rather do so experimentally, by actually simulating different strategies. Have students in random groups of three brainstorm about possible strategies for optimizing the chance of winning the game.

Students' initial thinking is likely to be based on empirical evidence, actually tossing a given number of dice and finding the sum. Eventually they may want to replicate the situations in ways that are less time consuming by using random number generators and writing some simple steps to facilitate the process. Several examples are described below.

#### **1) Open-Ended Approach:**

The problem is not well defined- how many tosses, how long should the game run, etc. Some groups might decide to set a limit on the number of points, say whoever reaches 100 points first wins. Others might decide to limit the number of tosses, say to 25, where the winner is the person with the most points after 25 tosses. Others might decide to look for the mean number of points per toss for a given number of dice. Be sure students recognize that the number of points will vary per toss for a given number of dice, and they need to find some way to estimate what is likely to happen over the long run.

#### **2) More-Structured Approach to Finding a Model:**

If students need more guidance, structure the problem by prescribing the total number of points to win the game, e.g., the first to reach 100, or by defining the number of tosses, e.g., who is ahead after 25 tosses for each participant. Encourage the students to share the work, either with different students repeatedly tossing a different number of dice or by all tossing the same number of dice and pooling their results.

What to Expect: Example Student Approaches

**No Technology:**

One approach is to investigate what is likely for each number of tosses from 1 to 7. One student might toss one die 25 times, do this repeatedly, then find the mean of the sums for one die. Another might do the same for two dice and so on. Another approach is to have everyone in class toss one die and find the mean of the sums earned by the students in the class; repeat the process with everyone in class tossing two dice and so on. Some students might think alternating the number of dice would be a good strategy, particularly if they are behind and want to risk tossing a larger number of dice to potentially get a larger sum.

After students have tossed different numbers of dice by hand and recorded the results, they should realize they need to do many simulations to see any pattern. This is where technology comes in. Note however, that technology can easily become a black box, where inputting several numbers produces a result that might not be understood. Ensuring that students actually understand what the technology is doing often means taking small steps to lay the groundwork. Some students might be content with primarily using technology to generate the random numbers and working from there to eliminate sixes from the sums. Several possible pathways are described below.

**Example 1 Best Score out of 25 tosses:**

**Simulation Strategy #1 for one die:**

A list can be used to simulate the number of points that would be earned if a player took 25 turns and tossed only one die each turn by going to MATH, PROB,randInt(, enter (Figure 1). Choose 1 as the lower bound, 6 as the upper bound and n=25, enter, sto L1, enter.

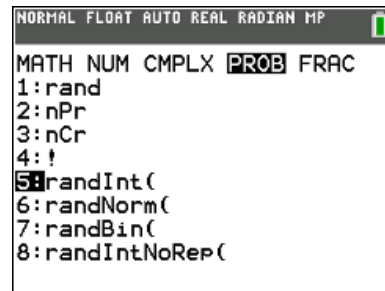


Figure 1: Generating random integers

Figure 2 shows 25 random integers from 1 to 6 that have been stored in List 1. Note: students may use random integers from 0 to 5 realizing that the 6's count zero. This works well for 1 die but will present a challenge for multiple dice.

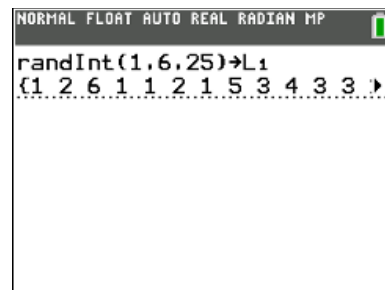


Figure 2: Simulate tossing one die 25 times

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

Choose list, MATH, sum(L1), enter to sum the 25 numbers (Figures 3 & 4). Be aware though, that this sum includes all the sixes.



Figure 3: Sum the faces for the 25 tosses

Students will need to devise a method to eliminate any toss that was a six. For example, using list, Sort D, L1 enter will make it easy to see the number of times the toss resulted in a six (four tosses in Figure 4). Subtracting 6 times the number of sixes from the original sum gives the total score for the 25 tosses, 58.

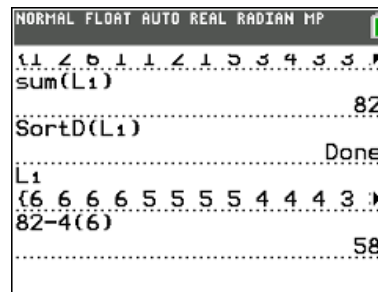


Figure 4: Subtract 4 x 6 from sum

Students can record the sums in another list manually by selecting Stat, edit, and choosing a list, say L2 (Figure 5).

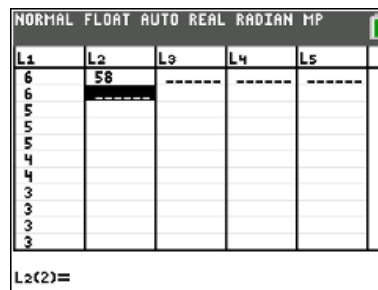


Figure 5: Capture the score in another list

They might also store the result in L2 as illustrated in Figure 6, by storing each sum in a specific row in L2 beginning with L2(1).

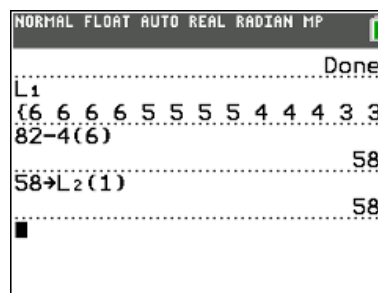


Figure 6: Storing the sum in a list

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

To repeat the process, arrow up to the line in which the random integers were generated (Figure 7). Enter will copy and paste that line at the cursor. Using enter again will execute the command.

```
HISTORY
randInt(1,6,25)→L1
{1 2 6 1 1 2 1 5 3 4 3 3 2
sum(L1)
..... 82
SortD(L1)
..... Done
L1
{6 6 6 6 5 5 5 5 4 4 4 3 3
82-4(6)
..... 58
```

Figure 7: Generating a new set of random integers

The commands to generate the set of random integers, to sort them in descending order and to display the sorted list can be put on one line and separated by a colon (Alpha, period) as in Figure 8 (randInt(1,6,25)→L1: SortD(L1): L1). Counting  $n$ , the number of 6's in the list, subtracting those from the sum and storing to the next row in L2 can be done by list, MATH, Sum(L1)- $n(6)$  sto L2(2) (Figure 8).

```
NORMAL FLOAT AUTO REAL RADIAN MP
{6 6 6 6 5 5 5 5 4 4 4 3 3
82-4(6)
..... 58
58→L2(1)
..... 58
randInt(1,6,25)→L1:SortD(L1)
{6 6 5 5 5 4 4 3 3 3 3 3 3
sum(L1)-2(6)→L2(2)
..... 62
```

Figure 8: Combining commands to find the sum when tossing a die 25 times

Students should repeat the process at least 30 times or collect the score from everyone in class to get a sampling distribution for the possible scores when tossing one die 25 times.

To graph the results, set the window with Xmin =40, Xmax = 90, Xscl =5, Ymin= -1, Ymax = 15 and Yscl = 1. Select statplot, Plot 1 on, histogram, choose the list with the scores, frequency1 (Figure 9) and then graph. Figure 10 displays a sampling distribution of the scores from 50 simulations tossing 1 die 25 times. Trace will show the width of each bin and the frequency of scores in that interval.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRESS [◀] OR [▶] TO SELECT AN OPTION
Plot1 Plot2 Plot3
On Off
Type: [ ] [ ] [ ] [ ] [ ]
Xlist:L2
Freq :1
Color: BLUE [ ]
```

Figure 9: Setting up a graph

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

Using list, MATH, mean(List containing scores) will find the mean, about 62 for this simulation. Draw, vertical line and the arrow keys can be used to display the mean. Thus, a player who tosses one die 25 times is likely to end up with a score somewhere around 62.

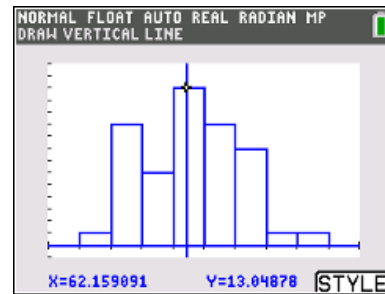


Figure 10: Sampling distribution of HOG scores when tossing one die 25 times

### Strategy #1 for two dice:

To estimate a score when tossing two dice 25 times, generate two columns of 25 random numbers as before, storing one set in L1 and the other in L2 (lines 1 and 2 in Figure 11). The command  $(L1+L2)*(L1 \neq 6 \text{ and } L2 \neq 6)$  (test will display  $\neq$ ) will produce the sum of the numbers in a row, where, any 6 is replaced by a 0 (note the product in the command will be 0, see line 6 in Figure 11). (Note also that with this command for adding the lists, it is not necessary to sort the lists to see how many 6s were in each.) Sum the 25 numbers in L3 and store the sums in consecutive rows in L4 beginning with L4(1) (line 7 in Figure 11).

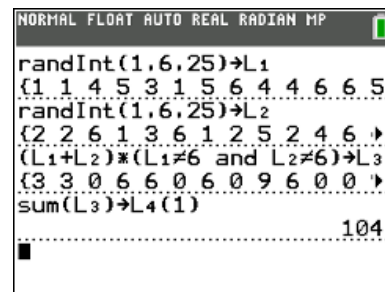


Figure 11: Generating scores for tossing two dice, neither of which is a 6

Arrowing up will highlight the command to generate a new set of random numbers for L1, enter will paste the line at the cursor and enter again will generate the new set of random numbers. Repeat the process for L2. Arrow up to copy and paste the sum command and store the new sum to the next row in L4. STAT edit displays the lists containing the random numbers (L1 and L2), the sum of each row (L3), and the total sums for each of 25 tosses of one die (L4) (Figure 12).

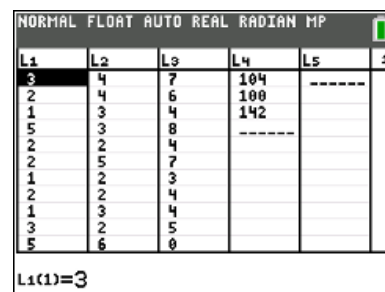


Figure 12: Cumulating Hog scores when tossing two dice

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

It is important for students to experience the copy and paste steps to actually visualize the numbers in L1, L2 and the sums in L3. Once they have worked through the process enough to be familiar with what is happening, they can join all of the commands for generating the random numbers and finding the sums in one line separated by colons. Enter will produce the sums for each row, and sum(L3) can be stored in the next row in L4 (Figure 13).

```
NORMAL FLOAT AUTO REAL RADIAN MP
10.88 / 0.75070420
sum(L3)→L4(29)
89
randInt(1,6,25)→L1:randInt(1,6,25)→L2
{1 1 2 3 1 4 5 3 4 5 1 3 2
(L1+L2)*(L1≠6 and L2≠6)→L3
{5 0 6 0 2 8 10 7 8 10 0 0
sum(L3)→L4(30)
122
```

Figure 13: Combining commands to find the Hog sum of tossing two dice

Another method for cumulating the sums is to use the augment command. On the home screen create 25 random integers from 1 to 6 in both L1 and L2, find  $(L1+L2)*(L1 \neq 6 \text{ and } L2 \neq 6)$  sto L3, separating the commands by a colon. Sum L3 and store in in the first row in L4 (Figure 14)

```
NORMAL FLOAT AUTO REAL RADIAN MP
randInt(1,6,25)→L1:randInt(1,6,25)→L2
{6 5 8 9 2 5 4 6 8 0 5 8 6
sum(L3)→L4(1)
106
```

Figure 14: Generating two sets of 25 random integers and finding the Hog sum

Arrow up to the line generating the random integers, select enter, enter to generate two new sets of 25 tosses (Figure 15).

```
HISTORY
randInt(1,6,25)→L1:randInt(1,6,25)→L2
{6 5 8 9 2 5 4 6 8 0 5 8 6
sum(L3)→L4(1)
106
```

Figure 15: Generating new sets of 25 tosses of two dice

To add the sum of L3 to L4 use list, OPS, augment(L4, {(sum(L3))} store L4) enter and the new sum of L3 (99 in this example) is appended to the list of sums (Figure 16).

```
NORMAL FLOAT AUTO REAL RADIAN MP
randInt(1,6,25)→L1:randInt(1,6,25)→L2
{6 5 8 9 2 5 4 6 8 0 5 8 6
sum(L3)→L4(1)
106
randInt(1,6,25)→L1:randInt(1,6,25)→L2
{0 0 5 7 0 5 4 3 10 6 6 6
augment(L4, {sum(L3)})→L4
{106 99}
```

Figure 16: Augmenting L4 with a new sum



# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

At some point, students might want to combine the commands for generating random numbers and the augment command to shorten the process (Figure 17).

```

NORMAL FLOAT AUTO REAL RADIAN MP
{106 99 109 88 84 134 144
randInt(1,6,25)→L1:randIn▶
{106 99 109 88 84 134 144
randInt(1,6,25)→L1:randIn▶
{106 99 109 88 84 134 144
randInt(1,6,25)→L1:randIn▶
{106 99 109 88 84 134 144
randInt(1,6,25)→L1:randIn▶
{106 99 109 88 84 134 144

```

Figure 17: Combining the steps

Figure 18 displays the distribution of 50 scores in L4 when tossing two dice 25 times.

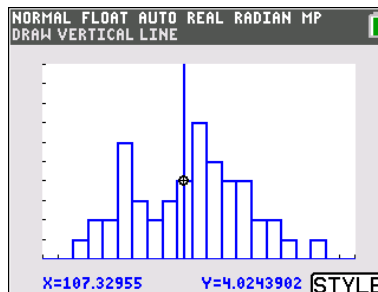


Figure 18: Distribution of Hog mean scores tossing two dice 25 times.

The same procedure can be used to estimate the total scores in 25 tosses of three, four, five, six or seven dice. Because only six lists are displayed, to find the mean when tossing six dice, insert a new list by arrowing to the top line of L6 and selecting 2<sup>nd</sup> ins. Use the alpha key to name the new list. Two new lists, L7 and Score, have been added in Figure 19.

L4	L5	L6	L7	SCORE	8
5	2	1	3		
3	3	2	4		
4	6	2	3		
2	6	1	3		
2	3	4	6		
5	2	2	5		
6	5	4	4		
6	1	3	6		
6	5	6	5		
6	4	5	2		

SCORE(L1)=

Figure 19: Adding new lists

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

#### Example #2: Finding the mean sum for tossing a given number of dice one time

Another approach is to consider what a typical score will be when tossing a specified number of dice, say three.

#### Simulation Strategy #2 for three dice:

Generating three random integers and storing them in L1 displays the integers (line 2 in Figure 20). If the set does not contain a 6, the list can be summed and stored in consecutive rows in L2. If the set contains a 0, store 0 in the appropriate row in L2. In Figure 20, 7 is stored in row 8 of L2 and 0 is stored in row 9 of L2.

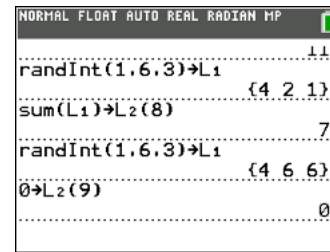


Figure 20: Finding the sum of three random integers

Continuing this process produces a distribution of sums from tossing three dice, none of which is a 6 (Figure 21).

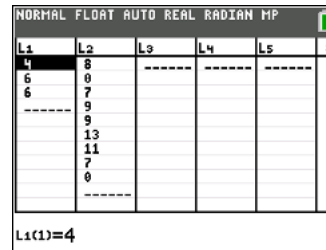


Figure 21: Generating a distribution of sums of three random numbers none of which is 6

The mean (list, MATH, mean(L2)) can be displayed on the graph by drawing a vertical line (Figure 22). The same process can be used to find the mean sum for tossing any number of dice by adjusting the number of random integers generated.

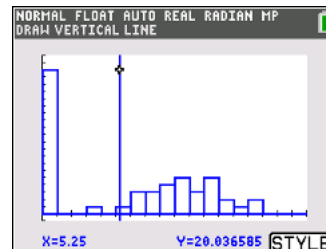


Figure 22: Hog mean sum tossing three dice

#### Simulation Strategy #3 for four dice:

Another approach is to define each list as a set of random numbers and make use of the augment command to cumulate the sums in one list. To find a typical score for tossing four dice define L1 with the formula to generate random numbers using quotes to lock in the formula: Type stat, Edit, arrow up to the list name, enter. Type "math, Prob, randInt, enter. Set the parameters 1,6, with n= 4, arrow to paste and enter, " (Figure 23).

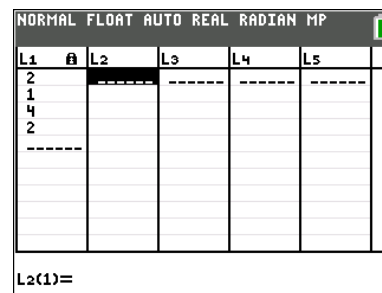


Figure 23: Defining a list with random numbers

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

On the Home screen, type the command for finding the sum of the list and storing either the sum or a zero for when the max of the list is 6 in L2  
 $\text{sum}(L1) * (\text{max}(L1) \neq 6) \rightarrow L2$  (Figure 24).

To add the next iteration to L2, type  
 $\text{augment}(L2, \{\text{sum}(L1) * (\text{max}(L1) \neq 6)\}) \rightarrow L2$  (Figure 25),  
then enter. Continuing to press enter will simulate more rolls of four dice.

Continuing this process produces a distribution of sums from tossing four dice. Figure 26 shows a graph of the simulated distribution of the sums of tossing four dice. The mean (list, MATH, mean(L2)), about 5.77 can be displayed on the graph by drawing a vertical line .

The same process can be used to find the typical score using the mean for tossing any number of dice by changing the number of dice generated in the formula for L1.

### Simulation Strategy #4 for five dice:

To find the typical score for tossing five dice, on the home screen, define each list with a formula to generate 100 random numbers (math, Prob, randInt, enter). Set the parameters 1,6, with n= 100, arrow to paste and enter. Connect the definition for each list with a colon (Figure 35). Enter will display the set of random numbers in L5, but L1, L2, L3, L4, and L5 will each have 100 random integers representing tossing five dice 100 times.(Figure 27)

## TEACHER NOTES

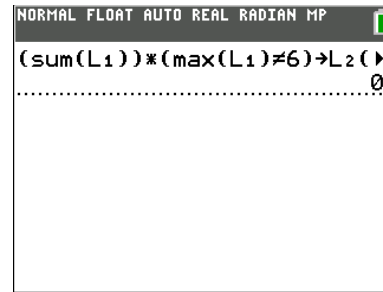


Figure 24: Finding the Hog sum of tossing four dice and storing to L2

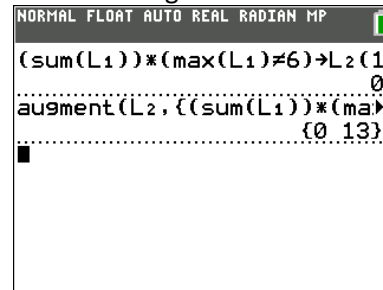


Figure 25: Augmenting L2 by a new iteration

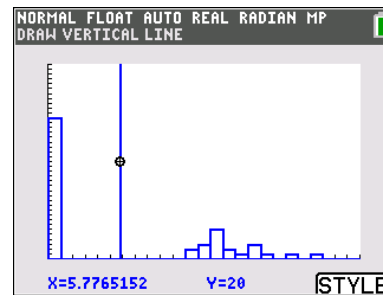


Figure 26: Simulated distribution of Hog sums tossing four dice

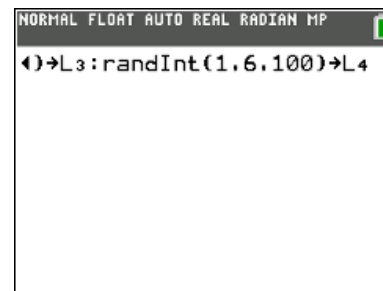


Figure 27: Defining random numbers for L1, L2, L3, L4, and L5

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

### TEACHER NOTES

Go to stat EDIT, insert a new column next to L5 and label it SCORE. Arrow up and enter to define the rule for SCORE as “(L1+L2+L3+L4+L5)\*(L1≠0 and L2≠0 and L3≠0 and L4≠0 and L5≠0)” then enter (Figure 28).

L2	L3	L4	L5	SCORE	6
1	2	4	6	0	
3	6	6	5	0	
3	6	3	5	0	
2	1	5	5	18	
2	6	4	2	0	
1	4	5	4	0	
3	4	1	3	0	
6	3	3	3	0	
2	5	5	5	22	
5	4	5	1	17	
5	4	6	4	0	

SCORE= "(L1+L2+L3+L4+L5)\*(L1≠0 and L2≠0 and L3≠0 and L4≠0 and L5≠0)"

Figure 28: Defining SCORE as the Hog sum of each toss of five dice

On the home screen use list, MATH, mean(SCORE), enter to see the mean (Figure 29).

randInt(1,6,100)→L1:randI▶  
 {3 1 2 6 1 3 3 5 2 6 2 6 4  
 mean(LSCORE  
 ..... 6.68

Figure 29: Mean Hog score for 100 tosses of five dice

Use statplot to see the distribution of SORES when tossing five dice 100 times (Figure 30).

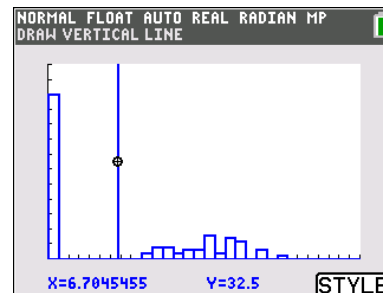



Figure 30: Distribution of Hog scores when tossing five dice

Table 1 shows the results for simulating the mean sum when tossing 1 to 7 dice 50 times as well as the mean score in 25 tosses.

Table 1

Number of dice	Estimated mean score per toss	Estimated mean score for 25 tosses
1	2.49	62.16
2	4.22	107.4
3	5.25	131.25
4	5.77	144.25
5	6.68	172.0
6	6.72	168.00
7	6.31	157.75

 **Validating the Models**

***Students should validate their models either by asking whether the models make sense in different scenarios related to the context or by finding other information to reflect against the model. The suggestions below might be useful in helping students think about whether their model was reasonable:***

1. Students should compare their results to those others found. Note that the values might vary as much as 3 points. If the results are quite different, they should reexamine what they did.
2. If students have a background in thinking about probability, they might use expected value to compare the theoretical results to the experimental results. Expected value is the sum of the probability of each outcome times the value of that outcome. For example, the formula for the expected value for rolling one die would be:

$$\frac{1}{6}(1) + \frac{1}{6}(2) + \frac{1}{6}(3) + \frac{1}{6}(4) + \frac{1}{6}(5) + \frac{1}{6}(6) = \frac{15}{6} = 2.5$$

For two dice:

$$\frac{1}{36}(2) + \frac{1}{36}(3)(2) + \frac{1}{36}(4)(3) + \frac{1}{36}(5)(4) + \frac{1}{36}(6)(5) + \frac{1}{36}(7)(4) + \frac{1}{36}(8)(3) + \frac{1}{36}(9)(2) + \frac{1}{36}(10)(1) + \frac{1}{36}(11)(0) + \frac{1}{36}(12)(0) = \frac{150}{36} = 4.17$$

And so on. (Note that as the number of dice increases, the complexity of counting the possible number of ways to obtain a sum also increases.)

 **Extension**

1. Answer each of the following:
  - a) How would your analysis change if the number of dice allowed was from 1 to 10?
  - b) Is the probability of getting a sum of 25 tossing five dice greater than, less than or the same as the probability of getting a sum of 36 tossing six dice? Explain your reasoning
  
2. The discrete graph of the expected value of the Hog score for a toss of a given number  $N$  of dice can be generated by making L1 (or other named list) = seq(K,K,1,24) (this generates a sequence of  $N$  values from 1 to 24), and L2 (or another named list) =  $3 * L1 * (5/6)^{L1}$  (this generates the corresponding expected scores on Hog for a toss of  $N$  dice) and then plotting the scatterplot for L1 and L2. (The continuous version of the graph can be obtained by graphing  $Y1 = 3 * x * (5/6)^x$ .) Make a plot of (number of dice, expected value). Describe how the plot can be used to inform a strategy for winning in Hog.
  
3. Design a strategy for winning the games below.
  - a) A New Game: For each turn, a player repeatedly rolls a die until either a 6 is rolled or the player decides to "hold":
    - If the player rolls a 6, the player scores nothing, and it becomes the next player's turn.
    - If the player rolls any other number, it is added to the player's turn total and the player's turn continues.
    - If a player chooses to "hold", the player's turn total is added to the player's score, and it becomes the next player's turn.
    - The first player to score 100 or more points wins.
  - b. A variation: For each turn, a player repeatedly rolls one or two dice until a 6 is rolled or the player decides to hold.
    - If neither face is a six, the sum of the faces is added to the player's score and the player's turn continues.
    - If a player chooses to "hold", the total the player has earned from the round is added to the player's score and it becomes the next player's turn.
    - The first player to score 100 or more points wins.

# Modeling: Simulating the Game of Hog

## T-84 Plus CE Technology

## TEACHER NOTES

4. A fair game is when every opponent has an equal chance to win the game. Is the following game fair?
- Player 1 rolls a die and records the number.
  - Player 2 rolls two dice. If one of the numbers is the same as the number player 1 rolled, player 2 gets a point. If neither number is the same, player 1 gets a point
  - The turn passes to player 1 again.
  - First player to 11 points wins.

### Resources

Resource: See <http://jse.amstat.org/v11n2/feldman.html>