Java Rolling two Dice Histogram

Project Definition

Dan McElroy
October 2018

This video is offered under a **Creative Commons Attribution Non-Commercial Share** license. Content in this video can be considered under this license unless otherwise noted.

Table of Contents

Rolling Dice Histogram Project Definition

Design a Java program that does the following:

1) Use a loop to simulate the rolling of a pair of dice 1000 times.
2) Count the number of times each value from 2 to 12 occurred using an array. The array is to hold counts for the number of times each value of 2 to 12 occurred. If the value is 2, then increment its counter. If the value is 3, then increment its counter, etc.
3) Display the result of each of the counters from the array, **using a for loop**.

Rolling Dice

Originally, **dice** is the plural form of **die**. In modern usage, the word **dice** can be used as both a single or plural noun. **Dice** can refer to one or more than one dice.

The most common die is six-sided with spots from 1 to 6. Rolling two dice can give a sum of 2 to 12.

Possible Combinations of Two Dice

Possible combinations for 2 die (Wikipedia)

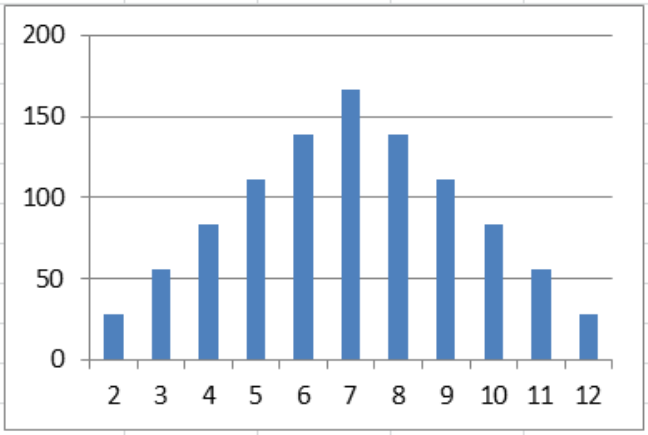
Dice Roll	Possible Dice Combinations
2	1-1
3	1-2, 2-1
4	1-3, 2-2, 3-1
5	1-4, 2-3, 3-2, 4-1
6	1-5, 2-4, 3-3, 4-2, 5-1
7	1-6, 2-5, 3-4, 4-3, 5-2, 6-1
8	2-6, 3-5, 4-4, 5-3, 6-2
9	3-6, 4-5, 5-4, 6-3
10	4-6, 5-5, 6-4
11	5-6, 6-5
12	6-6

+	Die A					
Die B	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

When rolling two dice, the most common total is 7. There are six ways to roll a 7, but only one way to roll a 2 or a 12.

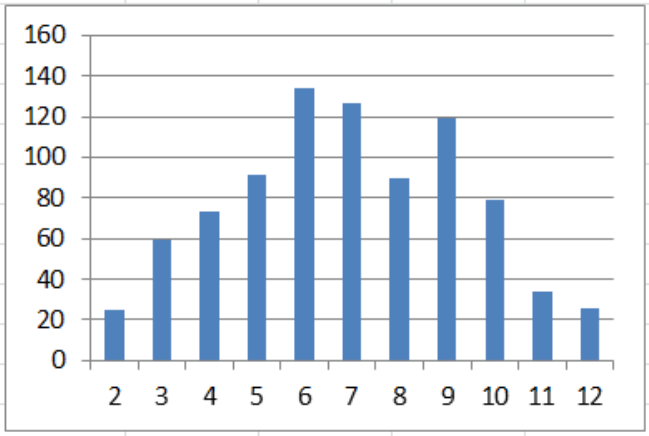
Rolling 2 Dice Histogram - Expected Counts for 1,000 rolls

Total of two dice	Ways to roll a number	Average 1000
2	1	28
3	2	56
4	3	83
5	4	111
6	5	139
7	6	167
8	5	139
9	4	111
10	3	83
11	2	56
12	1	28



Rolling 2 Dice Histogram Random Counts for 1,000 rolls

Total of two dice	Ways to roll a number	Average 1000
2	1	25
3	2	59
4	3	73
5	4	91
6	5	134
7	6	127
8	5	90
9	4	119
10	3	79
11	2	34
12	1	26



Program Organization

Main Program
- Roll 2 dice 1000 times
- Count the number of times each value occurred
- Display the counts

Use the **Math.random( )** function to simulate rolling two dice, add the two rolls up to get value of the 2 dice.

Use the value of the two rolls to increment a counter in an array

After 1000 rolls have been done, display the counts that were collected in the array.

Pseudo Random Number Generator

A **pseudorandom number generator** (**PRNG**) is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by an initial value, called the PRNG's *seed.* A PRNG can be started from an arbitrary initial state using a seed state.

-Wikipedia

Pseudo Random Number Generator in Java

The **Math.random( );** function returns an double in the range from 0.0 to just less than 1.0.

The result of multiplying Math.random( ) by 6 is a number from 0.0 to 5.99999. Take the integer of this and the result becomes 0 to 5. Add one and the new result is from 1 to 6.

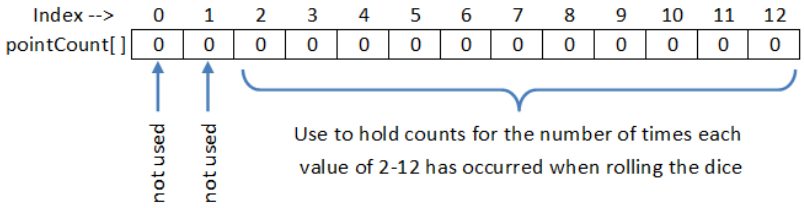
To get an integer from 2 to 12, simulating the roll of two dice

```
int die1 = (int)(Math.random( )*6) +1;  // roll the first die
int die2 = (int)(Math.random( )*6) +1;  // roll the second die
int roll2dice = die1 + die2;   // the total of rolling 2 dice
```

An Array of Counters

The position in an array is called the index. The index is 0 for the first position of all arrays in Java. It may be tempting to try to save a few bytes of memory by using index = 0 as the counter for a roll of 2 on the dice, but don't worry about wasting a couple of bytes in memory. All of the bytes in memory used by a program are returned back to the operating system when the program ends.

Use an Array as a Collection of Counters



In this example, an array of 13 integers has been created and named **pointCount**. The size of the array is set to 13 because the array starts from 0. This is because 0 to 12 is 13 elements when counting from zero.

Creating an Array in Java

To create an array named **pointCount** of 13 integers in Java:

1a) Declare the variable

```
int[ ] pointCount;
```

1b) Create the array 'object' and assign it to the variable

```
pointCount = new int[13];
```

Both steps can be combined when creating an array in Java

2) Declare the variable and create the array at the same time

```
int[ ] pointCount = new int[13];
```

Each element in an array of integers is automatically initialized to 0;

Accessing an Array Element

To access element 5 in the array:

```
int x = pointCount[5];   // x is set to the value from pointCount[5]
```

To place a 0 in element 5:

```
pointCount[5] = 0;
```

To increment element 5:

```
pointCount[5]++;
```

--- or ---

```
int i = 5;        // the index can be a variable or even an expression
pointCount[i]++;  // the variable i is being used as the index
```

Incrementing the Array of Counts

In the code below, **roll2dice** is the sum of the two dice that were rolled. Increment the appropriate element within the array based on the the the value rolled.

```
int die1 = (int)(Math.random( )*6) +1;  // roll the first die
int die2 = (int)(Math.random( )*6) +1;  // roll the second die
int roll2dice = die1 + die2;   // the total of rolling 2 dice
pointCount[roll2dice]++;      // increment the counter
```

Put these instructions in a loop that runs 1000 times

Display the Array of Counts

Remember that we are only concerned with elements 2 through 12.

```
for (int i=2; i<=12; i++)  {
    //   display the index i and the contents of the array
}
```

Sample output

```
1000 dice rolls
Value   Count
  2      29
  3      48
  4      91
  5     116
  6     135
  7     163
  8     137
  9     106
 10      88
 11      59
 12      28
```

Image Credits

TwoDice.jpg by Roland Scheicher - Wikipedia, Public domain

Dice combinations – Wikipedia article on Craps