Normal distribution of random numbers

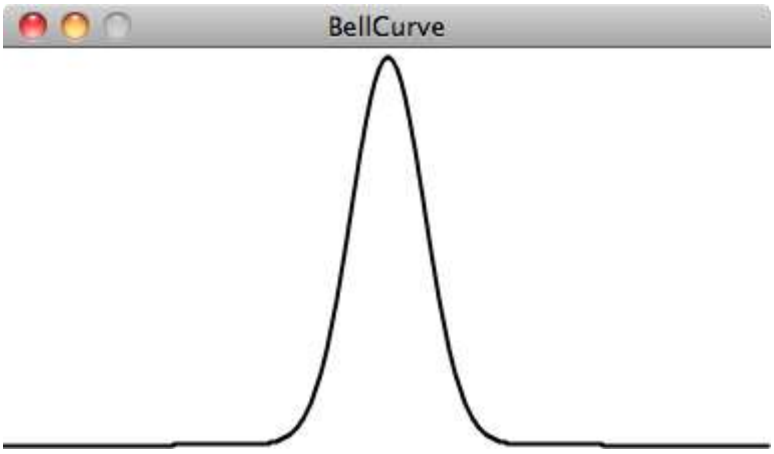
Let's say we want to make that program that generates a world of monkeys. Your program could generate a thousand Monkey objects, each with a height value between 200 and 300 (as this is a world of monkeys that have heights between 200 and 300 pixels).

```
var randomHeight = random(200, 300);
```

Does this accurately depict the heights of real-world beings? Think of a crowded sidewalk in New York City. Pick any person off the street and it may appear that their height is random. Nevertheless, it’s not the kind of random that `random()` produces. People’s heights are not uniformly distributed; there are a great deal more people of average height than there are very tall or very short ones. To simulate nature, we may want it to be more likely that our monkeys are of average height (250 pixels), yet still allow them to be, on occasion, very short or very tall.

A distribution of values that cluster around an average (referred to as the “mean”) is known as a “normal” distribution. It is also called the Gaussian distribution (named for mathematician Carl Friedrich Gauss) or, if you are French, the Laplacian distribution (named for Pierre-Simon Laplace). Both mathematicians were working concurrently in the early nineteenth century on defining such a distribution.

When you graph the distribution, you get something that looks like the following, informally known as a bell curve:

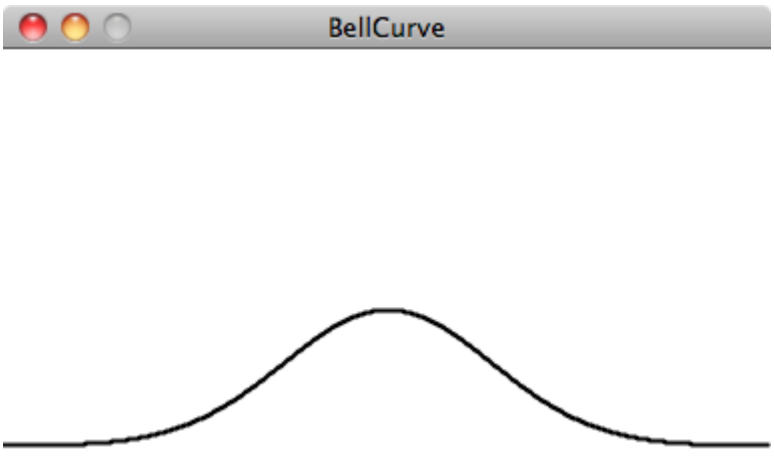


Graph of a standard bell curve

A standard bell curve

The curve is generated by a mathematical function that defines the probability of any given value occurring as a function of the mean (often written as μ, the Greek letter *mu*) and standard deviation (σ, the Greek letter *sigma*).

The mean is pretty easy to understand. In the case of our height values between 200 and 300, you probably have an intuitive sense of the mean (i.e. average) as 250. However, what if I were to say that the standard deviation is 3 or 15? What does this mean for the numbers? Looking at graphs can give us a hint. The graph above shows us the distribution with a very low standard deviation, where the majority of the values cluster closely around the mean. The graph below shows us a higher standard deviation, where the values are more evenly spread out from the average:



Graph of a bell curve with a higher standard deviation

A bell curve with a higher standard deviation

Not familiar with the concept of "standard deviation"? Don't worry! You can study Variance and standard deviation separately on Khan Academy before continuing.

The numbers work out as follows: Given a population, 68% of the members of that population will have values in the range of one standard deviation from the mean, 98% within two standard deviations, and 99.7% within three standard deviations. Given a standard deviation of 5 pixels, only 0.3% of the monkey heights will be less than 235 pixels (three standard deviations below the mean of 250) or greater than 265 pixels (three standard deviations above the mean of 250).

Calculating Mean and Standard Deviation

Consider a class of ten students who receive the following scores (out of 100) on a test:

85, 82, 88, 86, 85, 93, 98, 40, 73, 83

The mean is the average: 81.3

The standard deviation is calculated as the square root of the average of the squares of deviations around the mean. In other words, take the difference from the mean for each person and square it (variance). Calculate the average of all these values and take the square root as the standard deviation.

Score	Difference from Mean	Variance
85	85 - 81.385−81.385, minus, 81, point, 3 = 3.7	(3.7)^2(3.7)2left parenthesis, 3, point, 7, right parenthesis, start superscript, 2, end superscript = 13.69
82	82 - 81.382−81.382, minus, 81, point, 3 = 0.7	(0.7)^2(0.7)2left parenthesis, 0, point, 7, right parenthesis, start superscript, 2, end superscript = 0.49
88	88 - 81.388−81.388, minus, 81, point, 3 = 6.7	(6.7)^2(6.7)2left parenthesis, 6, point, 7, right parenthesis, start superscript, 2, end superscript = 44.89
etc.	...	...
	Average Variance:	228.81

The standard deviation is the square root of the average variance: 15.13

Want to understand standard deviation better? You can study Variance and standard deviation in more depth here on Khan Academy.

Luckily for us, to use a normal distribution of random numbers in a program here, we don't have to do any of these calculations ourselves. Instead, we can make use of the `Random` object provided by ProcessingJS.

To use `Random`, we must first instantiate a new `Random` object, passing in 1 as the parameter. We call that variable "generator" because what we've created can be basically thought of as a random number generator.

```
var generator = new Random(1);
```

If we want to produce a random number with a normal (or Gaussian) distribution each time we run through `draw()`, it's as easy as calling the function `nextGaussian()`.

```
var num = generator.nextGaussian();
println(num);
```

So, now, what are we supposed to do with this value? What if we wanted to use it, for example, to assign the x-position of a shape we draw on screen?

The `nextGaussian()` function returns a normal distribution of random numbers with the following parameters: *a mean of zero* and *a standard deviation of one*. Let's say we want a mean of 200 (the centre horizontal pixel in a window of width 400) and a standard deviation of 60 pixels. We can adjust the value to our parameters by multiplying it by the standard deviation and adding the mean.

```
var standardDeviation = 60;
var mean = 200;
var x = standardDeviation * num + mean;
```

Now, we can create our program that draws semi-transparent circles according to a normal distribution. The darkest spot will be near the centre, where most of the values cluster, but every so often circles are drawn farther to the right or left of the centre.