Random Numbers

Exercise: Describe in your own words the meaning of "chance", "deterministic", "correlation", and "random". Give an example of the proper use of each word.

Shodor Tutorial on Random Number Generators

http://www.shodor.org/interactivate/discussions/RandomNumberGenerato/

Exercise: Use $\boldsymbol{r_1}$ = 3, *c* = 1, *a* = 4, *M* = 9 in Equation (4) to generate a sequence of pseudorandom numbers. We will do this for the first three numbers in the sequence and then let you have some fun and finish the job:

(6) r_2 = remainder $(\frac{4\times3+1}{9})$ = remainder $(\frac{13}{9})$ = remainder $(1\,\frac{4}{9})$ = 4,

(7) r_3 = remainder $(\frac{4\times4+1}{9})$ = remainder $(\frac{17}{9})$ = remainder $(1\,\frac{8}{9})$ = 8,

(8) r_4 = remainder $(\frac{4\times8+1}{9})$ = remainder $(\frac{33}{9})$ = remainder $(3\,\frac{6}{9})$ = 6,

(9) r_{5-10} = 7, 2, 0, 1, 5, 3.

We see from this example that that we get a sequence of length *M* = 9, after which the entire sequence repeats in exactly the same order. Since repetition is anything but random, using a very large value of *M* helps hide the lack of randomness, and is one of the tricks used in pseudorandom number generators.

Scaling and Uniformity of Random Numbers (Not Optional)

It is always a good idea, and sort of fun, to test a random number generator before using it.

As see from the above example, the linear congruent method generates random numbers in the range 0--*M*. If we want numbers in the more common range of 0--1, then we need only divide by the endpoint *M* = 9. By doing that, the sequence in Equations 5 -- 7 becomes

(10) 0.333, 0.444, 0.889, 0.667, 0.778, 0.222, 0.000, 0.111, 0.555, 0.333

This is still a sequence of length 9, but is no longer a sequence of integers.

Uniform Distributions

Exercise: Use your favorite random number generator to generate a sequence of 50 random numbers. (You can do this in Excel by using the *rand* function under *Formulas*, in Python using *random* function, and in Vensim using the *RANDOM UNIFORM* function; see examples in the Spontaneous Decay Module.) In case you are not able to generate those numbers at this instant, here are 50 random numbers that we have generated between 0 and 100:

(11) 21 24 43 88 78 35 38 0 13 50 66 45 6 98 99 95 4 76 89 2 34 86 85 15 96
1 86 33 90 48 44 9 88 27 8 91 17 37 4 3 79 79 31 48 72 26 64 24 71 4

Plot a histogram with 10 columns and decide if this distribution looks uniform. You can use Excell or some other program to plot a histogram of these numbers. We have gone to a Web-based interactive histogram plotter from the Shodor Foundation at

http://www.shodor.org/interactivate/activities/Histogram/

and used it to make the two histograms in Figure 1.

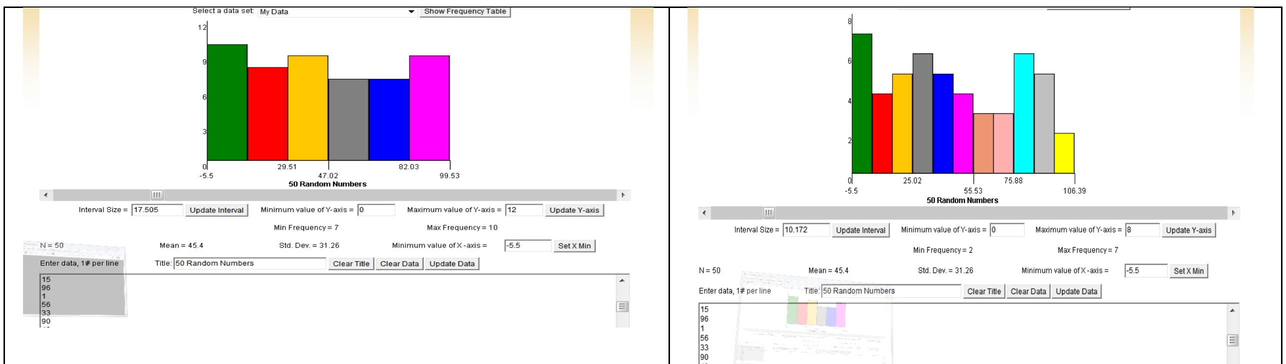
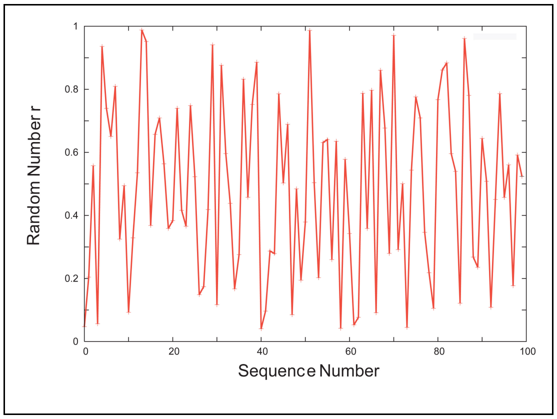


Figure 1: *Left:* A histogram of 50 random number using bin size of 17.5. *Right:* The same numbers using a bin size of 10.2.

While these histograms do *not* tell us anything about the randomness of the distribution, they do tell us about its uniformity. The large bin sizes on the left show a fairly uniform distribution. Since the numbers are random, some statistical fluctuation is to be expected. Note that is we plot these same numbers using a smaller bin size, as we do on the right, then there is more fluctuation and the uniformity appears less evident. In general, statistical fluctuations are less evident as the sample size increases.

Exercise: Testing a Sequence for Randomness

Figure 2. A sequence plotted as $(x_i, y_i) = (r_i, i)$.



OK, now that we have a way of testing a distribution for uniformity, let's get on to testing for randomness. You do not have to understand how the numbers are being generated or do anything involving math, but rather just plot some graphs. Your visual cortex is quite refined at recognizing patterns, and will tell you immediately if there is a pattern in your random numbers. While you cannot "prove" that the computer is generating random numbers in this way, you might be able to prove that the numbers are not random. In any case, it is an excellent exercise to help you "see" and learn what randomness looks like.

Probably the first test you should make of a random number generator, is to see what the numbers look like. You can do this by printing them out, as we did in Equation (11), which tells you immediately the range of values they span and if they seem uniform. Another test is to plot the random number versus its position in the sequence, that is, to plot $(x_i, y_i) = (r_i, i)$ with the points connected. We do this in Figure 2 for another sequence we have generated, and you should do it now for your sequence. Here we see that the numbers lie between 0 and 1 and jump around a lot. This is what random numbers tend to look like!

A better test of the randomness of a sequence is one in which your brain can help you discern patterns in the numbers that may indicate correlations (nonrandomness). This is essentially a scatter plot as shown in Figure 3. Here we plot successive pairs of random numbers as the *x* and *y* values of data points, that is, plot $(x, y) = (r_i, r_{i+1})$, without connecting the points. For example, (r_1, r_2), (r_3, r_4), (r_5, r_6). On the left of Figure 3 you see results from a random number generator that has made a "bad" choice of internal parameters [(a, c, M, r_1) =(57, 1, 256, 10) in Equation (4)]. It should be clear that the sequence represented on the left is not random.

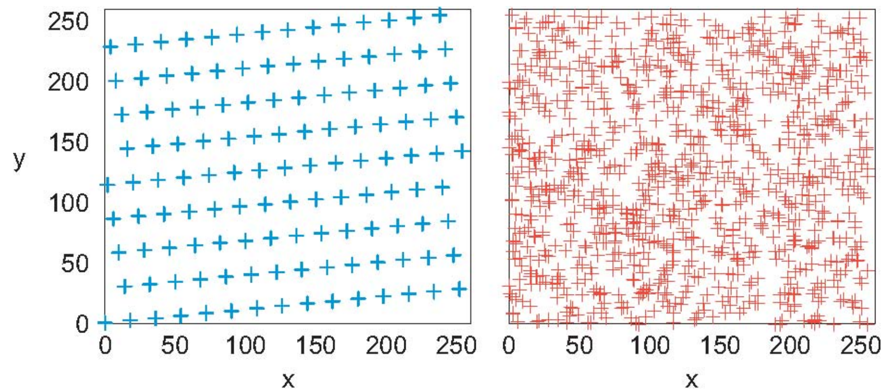


Figure 4 *Left*: Random numbers from a generator with correlations. *Right:* Random numbers from a properly functioning generator.

On the right of Figure 3 you see $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{r_i}, \boldsymbol{r_{i+1}})$ results from a random number generator that has made a "good" choice of internal parameters [*a* = 25214903917, c = 11, *M* = 281,474,976,710,656 in Equation (4)].

Make up your own plot and compare it to Figure 4 *right*. Be warned, your brain is very good at picking out patterns, and if you look at Figure 4 *right* long enough, you may well discern some patterns of points clumped near each other or aligned in lines. Just as there variations in the uniformity of the distribution, a random distribution often shows some kind of clumping --- in contrast to a completely uniform "cloud" of numbers.

There are more sophisticated and more mathematical tests for randomness, but we will leave that for the references. We will mention however, that one of the best tests of a random number generator is how will computer simulations that use that generator are able to reproduce the nature of natural processes containing randomness (like spontaneous decay), or how well they can reproduce known mathematical results. The modules on *Random Walk* and *Spontaneous Decay* contain such simulations, and the module on *Stone Throwing* contains such a mathematical result.