Generation of Random Numbers in Calc

Tony Richardson

richardson.tony@gmail.com

University of Evansville

December 9, 2013

The LibreOffice/OpenOffice spreadsheet application Calc has a fairly extensive set of statistics functions. Unfortunately, it only has a uniform random number generator function. Many simulations require random number generators for other (non-uniform) probability distributions. This paper describes how to generate random numbers for several of the most common distribution functions.

I Generation of Discrete Random Numbers

I.A Uniformly Distributed Random Numbers

Calc does have a discrete uniform random number generator. The **RANDBETWEEN(A, B)** function returns an integer valued number between A and B inclusive. For example, **RANDBETWEEN(1,6)** could be used to simulate the roll of a die or **RANDBETWEEN(1, 52)** might be useful when simulating the draw of cards from a deck.

I.B Binomially Distributed Random Numbers

The number of successes in a sequence of N independent success/failure experiments in which the probability of success is p is a binomially distributed random variable. The inversion method can be used to generate binomially distributed random variables. Since there is no built-in quantile function, it is implemented using a binomial CDF and a lookup function. Use one sheet (the sheet is named CDF in subsequent formulas) to calculate the desired binomial CDF as shown in Figure 1. The first two columns define the N and p parameters of the distribution. The third (C) column list values of n from 0 to N (set the formula in the first cell as **ROW()-2** and then copy and paste this formula into the remaining cells in this column.) The fourth (D) column contains 0 in the first cell (D2), cell D3 contains the formula **BINOMDIST($C2,$A$2,$B$2,1)**. This is the binomial CDF for the value in the cell in the previous row and column. This formula should be copied and pasted into the remaining cells in column D. Note that the CDF values are one row below the corresponding n values in the previous column. For example CDF(0) = P(X ≤ 0) = 0.0000009537 and CDF(4) = P(X ≤ 4) = 0.0197277069. Also note that a larger (smaller) value of N would require that the n and *CDF* columns be extended (shortened).

On a separate sheet, the **RAND()** and **LOOKUP()** functions can be used to generate binomially distributed random numbers. Use the following formula in any cell where you want a random number to appear **LOOKUP(RAND(),CDF.$D$2:$D$65536,CDF.$C$2:$C$65536)**. This formula uses the uniform random number generator to generate CDF values between 0 and 1 and then return the corresponding n values from the CDF table. (The lookup reference region is actually much larger than the region occupied by the CDF table. This allows the table to be extended without needing to also change random number generation formula.) Figure 2 shows a 10 row by 4 column of cells that each contain this formula.

N	p	n	CDF(n)
10	0.75	0	0.0000000000
		1	0.0000009537
		2	0.0000295639
		3	0.0004158020
		4	0.0035057068
		5	0.0197277069
		6	0.0781269073
		7	0.2241249084
		8	0.4744071960
		9	0.7559747696
		10	0.9436864853
			1.0000000000

Figure 1: The Binomial CDF

6	7	7	4	Sample	Population
8	7	7	7	Average	Average
7	9	8	8	6.8750	7.5
8	7	7	10		
7	7	9	7	Variance	Variance
8	6	4	6	2.1122	1.875
7	6	6	9		
7	7	8	9		
7	6	5	6		
5	8	4	4		

Figure 2: Binomially Distributed Random Numbers

I.C Random Binary Data

Random binary data can be generated using the binomial generator. Just set N equal to 1. If p is equal to 0.5 then the number of zeros generated will be equal to (on average) the number of ones generated. Reduce (increase) p to generate more (fewer) zeros than ones.

I.D Poisson Distributed Random Numbers

The number of events occurring in a fixed interval of time is often assumed to have a Poisson distribution. The average number of events in the interval is a parameter of the distribution and is typically denoted by the Greek letter λ (Lamda). Poisson distributed random numbers can be generated in a similar manner to that used to generate binomial numbers. However, Poisson random numbers have no theoretical upper limit on their value. The CDF table is extended until values as close to one as desired are reached. This is illustrated in Figure 3 for a Poisson distribution with distribution parameter of $\lambda = 2$. (Rows corresponding to values of n between 4 and 12 are hidden in order to reduce the size of the figure.)

The cells in the column labeled n contains values from 0 to 17. (A larger value of λ would require more values.) The first cell in the CDF column contains 0. The other cells in that column contain the formula: **POISSON(NREF,$A$2,1)** where NREF contains a reference to the value of n in the preceding row and column. (Theoretically the CDF is not equal to 1 unless n is infinite. Numerical

precision result in the CDF being rounded to 1 when *n* is 17.) Poisson distributed random numbers are generated using the formula: **LOOKUP(RAND(),CDF.$D$2:$D$65536,CDF.$C$2:$C$65536)**. Larger values of λ would require lengthier *n* and CDF columns.

Lambda	*n*	CDF(*k*)
2	0	0.0000000000
	1	0.1353352832
	2	0.4060058497
	3	0.6766764162
	13	0.9999997927
	14	0.9999999707
	15	0.9999999961
	16	0.9999999995
	17	0.9999999999
		1.0000000000

Figure 3: The Poisson CDF

I.E Negative Binomial Distributed Random Numbers

There are a couple of differing definitions of the negative binomial distribution. As defined by Calc, if we repeat a success/fail experiment until there are *r* failures, then the number of successes *k* that has occurred has a negative binomial distribution. The distribution parameter *p* is the probability of failure. (In some definitions *p* is the probability of success, in others the total number of trials is said to have a negative binomial distribution instead of the number of successes.) Calc does not have a built-in CDF for the negative binomial distribution, but does have a probability mass function (PMF). An extra step is needed to calculate CDF values from the PMF. This is shown in Figure 4. (Again some rows are hidden to reduce the size of the Figure.)

r	*p*	*k*	PDF(*k*)	CDF(*k*)
8	0.8	0	0.000000	0.000000
		1	0.167772	0.167772
		2	0.268435	0.436208
		3	0.241592	0.677800
		4	0.161061	0.838861
		13	0.000035	0.999985
		14	0.000011	0.999995
		15	0.000003	0.999999
		16	0.000001	1.000000

Figure 3: The Negative Binomial CDF

The first cell in the PDF column contains 0 while the other cells contain the formula: **NEGBINOMDIST(NREF,$A$2,$B$2)** where NREF contains a reference to the value of *n* in the preceding row and column and **$A$2** and **$B$2** are references to the *r* and *p* parameters of the distribution. Cells in the CDF column contain the formula **SUM($D$2:PREF)** where PREF is a reference to the PDF cell in the previous column, but in the same row. Random numbers are generated using the formula: **LOOKUP(RAND(),CDF.$E$2:$E$65536,CDF.$C$2:$C$65536)**.

II Generation of Continuous Random Variables

Calc has a uniform random number generator function (**RAND()**) that generates random numbers in the range 0 to 1. To generate uniformly distributed random numbers in the range A to B use the cell formula: **(B-A)*RAND()+B**.

Calc has quantile (inverse CDF) functions for most common continuous random variable distributions making it quite simple to generate the corresponding random numbers via the inversion method. Given an inverse CDF function of the form fooinv(x), you can generate random numbers with the corresponding distribution by using the cell formula: fooinv(RAND()). For example, to generate Gaussian (normal) random numbers with a mean of 0 and a standard deviation of one use the cell formula: **NORMINV(RAND(),0,1)**. Similarly to generate random numbers that have a Chi Squared distribution with 10 degrees of freedom use: **CHISQINV(RAND(),10)**. Inverse functions are available for the following distributions: Beta, Chi Squared, *F*, Gamma, Lognormal, Normal, and *t*. There are distribution functions for the Exponential and Weibull distributions, but no corresponding inverse functions. However, the inverse functions for these distributions can be expressed fairly simply in terms of other built-in functions. To generate Exponential random numbers (with distribution parameter LAMBDA) use: **-LN(1-RAND(),LAMDA)**. To generate Weibull random numbers with parameters ALPHA and BETA use: **BETA*POWER(-LN(1-RAND()),1/ALPHA)**.

Libre/Open Office Calc Distribution Functions

	Calc Notation			
Distribution	**PDF**	**CDF**	**Inverse CDF (percentiles)**	**RV Generation**
Uniform (A≤X≤B				randbetween(A, B)
Binomial	binodist(x; n; p; 0)	binodist(x; n; p; 1)		
Neg. Binomial (Pascal)	negbinomdist(x; r; p)			
Poisson	poisson(x; λ; 0)	poisson(x; λ; 1)		

Discrete RV Distribution Functions

	Calc Notation			
Distribution	**PDF**	**CDF**	**Inverse CDF (percentiles)**	**RV Generation**
Uniform (0≤X<1)	1	x	η	rand()
Uniform (A≤X<B)	1/(B-A)	(x-A)/(B-A)	(B-A)*η+A	(B-A)*rand()+B
Beta (0≤X<1)	betadist(x;α;β;0;1;0)	betadist(x;α;β;0;1;1)	betainv(η;α;β;0;1)	See Note 1
Chi Squared	chisqdist(x; *v*; 0)	chisqdist(x; *v*; 1)	chisqinv(η; *v*)	See Note 1
Exponential	expondist(x; λ; 0)	expondist(x; λ; 1)	-ln(1-η)/λ	See Note 1
F	fdist(f; v_1; v_2; 0)	fdist(f; v_1; v_2; 1)	finv(η; v_1; v_2)	See Note 1
Gamma	gammadist(x;α;β;0)	gammadist(x;α;β;1)	gammainv(η;α;β)	See Note 1
Lognormal	lognormdist(x;μ;σ;0)	lognormdist(x;μ;σ;1)	loginv(η;μ;σ)	See Note 1
Normal (Gaussian)	normdist(x;μ;σ;0)	normdist(x;μ;σ;0)	norminv(η, μ, σ)	See Note 1
t	tdist(x; *v*; 0)	tdist(x; *v*; 1)	tinv(η; *v*)	See Note 1
Weibull	weibull(x;α;β;0)	weibull(x;α;β;1)	β*power(-ln(1-η),1/α)	See Note 1

Continuous RV Distribution Functions

Notes:

1. You can generate random variables with the desired distribution from the inverse CDF function. Replace η with a call to the **rand** function. For example, to generate a normally distributed random variable that has a mean of 10 and a standard deviation of 4 use the cell formula: norminv(rand();4; 10).