Subject: Re: Random testing
Posted by news.verizon.net on Tue, 02 Jan 2007 15:24:13 GMT
View Forum Message <> Reply to Message

Joshua wrote:
> Does anybody know how to test
> a binary mask for spatial randomness?
> All I'm trying to do is see whether the 1's are spatially random
> or systematic.

I assume that you are refering to a 2-d mask. You might want to look at Section 14.7 of "Numerical Recipes" titled "Do Two-Dimensional Distributions differ?", which provides code for the 2-d K-S test. Their example model function (quadv1) is a uniform 2-d random distribution. Unfortunately, I am not aware of an IDL implementations of this code. (Also check the bug fix at low number counts given in http://www.numerical-recipes.com/forum/showthread.php?t=576).

You could also just test the 1-d distributions of X and Y for randomness, for example, by using the ksone.pro procedure at http://idlastro.gsfc.nasa.gov/ftp/pro/math/. But this will not rule out any non-random correlation between X and Y. --Wayne

Subject: Re: Random testing
Posted by James Kuyper on Tue, 02 Jan 2007 16:20:45 GMT
View Forum Message <> Reply to Message

Joshua wrote:
> Does anybody know how to test
> a binary mask for spatial randomness?
> All I'm trying to do is see whether the 1's are spatially random
> or systematic.

There is, inherently, no such thing as a yes-no test for randomness. In principle, quantum uncertainty is truly random, but nothing else in nature is. Quantum randomness can only be measured with real-world devices that introduce a certain amount of non-random noise; the measured result always has a certain amount of non-randomness, even if the thing being measured is truly random. So-called "random" number generators are more correctly known as "pseudo-random", because there's always an inherently deterministic algorithm behind them. For any finite set of data, there's always a non-random pattern that fits it, even if it is actually random.

The best that you can do is check for various specific types of non-randomness that might cause you problems. Try fitting the mask to various different models of patterns that you do not want to see in the mask, using methods that provide estimates of the statistical significance of the fit. If the data is truly random, the model parameters will tend not be statistically significant. However, if you test a sufficiently large number of different models, sooner or later you'll find one that appears to be a good match, even though the mask is truly random.

One thing I'd strongly recommend for a 2-D bit-mask, is to perform 2-D fourier transform of the mask. Peaks in the magnitude of the transform indicate auto-correlations with a corresponding lag. With truly random data, you shouldn't see any statistically significant peaks (except, of course, the one corresponding to a lag of (0,0)).