Diehard Tests

The Diehard Tests

- The diehard tests are a battery of statistical tests for measuring the quality of a random number generator.
- They were developed by George Marsaglia over several years and first published in 1995 on a CD-ROM of random numbers.

RDieHarder R Package

RDieHarder: R interface to the dieharder RNG test suite

The RDieHarder packages provides an R interface to the dieharder suite of random number generators and tests that was developed by Robert G. Brown and David Bauer, extending earlier work by George Marsaglia and others.

Version:	0.1.3
Depends:	R (≥ 2.5.0)
Published:	2014-02-21
Author:	Dirk Eddelbuettel
Maintainer:	Dirk Eddelbuettel <edd at debian.org>
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL:	http://code.google.com/p/rdieharder/
NeedsCompilation:	yes

Vignette on CRAN

RDieHarder: An R interface to the Die*Harder* suite of Random Number Generator Tests

Dirk Eddelbuettel
Debian
edd@debian.org

Robert G. Brown
Physics, Duke University
rgb@phy.duke.edu

Initial Version as of May 2007
Rebuilt on February 20, 2014 using RDieHarder 0.1.3

1 Introduction

Random number generators are critically important for computational statistics. Simulation methods are becoming ever more common for estimation; Monte Carlo Markov Chain is but one approach. Also, simulation methods such as the Bootstrap have long been used in inference and are becoming a standard part of a rigorous analysis. As random number generators are at the heart of the simulation-based methods used throughout statistical computing, 'good' random numbers are therefore a crucial aspect of a statistical, or

Diehard Tests

Birthday spacings: Choose random points on a large interval. The spacings between the points should be asymptotically exponentially distributed. The name is based on the birthday paradox.

Overlapping permutations: Analyze sequences of five consecutive random numbers. The 120 possible orderings should occur with statistically equal probability.

Diehard Tests

Ranks of matrices: Select some number of bits from some number of random numbers to form a matrix over 0,1, then determine the rank of the matrix. Count the ranks.

Monkey tests: Treat sequences of some number of bits as "words". Count the overlapping words in a stream. The number of "words" that don't appear should follow a known distribution. The name is based on the *infinite monkey* theorem.

Diehard Tests

Count the 1s: Count the 1 bits in each of either successive or chosen bytes.
Convert the counts to "letters", and count the occurrences of five-letter "words".

Parking lot test: Randomly place unit circles in a 100×100 square. If the circle overlaps an existing one, try again.
After 12,000 tries, the number of successfully "parked" circles should follow a certain normal distribution.

Diehard Tests

Minimum distance test: Randomly place 8,000 points in a 10,000 x 10,000 square, then find the minimum distance between the pairs.
The square of this distance should be exponentially distributed with a certain mean.

Random spheres test: Randomly choose 4,000 points in a cube of edge 1,000.
Center a sphere on each point, whose radius is the minimum distance to another point.
The smallest sphere's volume should be exponentially distributed with a certain mean.

Diehard Tests

The squeeze test: Multiply 231 by random floats on (0,1) until you reach 1. Repeat this 100,000 times. The number of floats needed to reach 1 should follow a certain distribution.

Overlapping sums test: Generate a long sequence of random floats on (0,1).
Add sequences of 100 consecutive floats.
The sums should be normally distributed with characteristic mean and sigma.

Diehard Tests

Runs test: Generate a long sequence of random floats on (0,1). Count ascending and descending runs. The counts should follow a certain distribution.

The craps test: Play 200,000 games of craps, counting the wins and the number of throws per game.
Each count should follow a certain distribution.