ERROR DETECTION IN NON-UNIFORM RANDOM VARIATES

Presented By:
Sougata Chaudhuri,Indian Institute of Technology,Kharagpur.
Under the guidance of:
Prof. Josef Leydold, Wirtschafts Universität, Wien

Project Outline

- Non-Uniform Random Variate Generation
- Errors in Random Variates
- Theory of Error Detection
- Graphical and Statistical Tests
- Introduction of Artificial Errors in Random Variate Generators
- Simulation Examples
- Development of Error Testing Package in R

How are non-uniform random NUMBERS generated?

Non-Uniform Random Variate Generation

- Usually generated by transforming sequence of independent U(0,1) random numbers into sequence of independent random variates of desired distribution.

- The basic assumption of such algorithms is that there is an ideal source of uniform random numbers available.

- Some of the well known transformation methods are *inversion*, *acceptance-rejection* and *decomposition* methods.

- Various of these algorithms have been used to build universal generators for fairly large distribution of families [1].

WHAT KIND OF ERRORS CAN BE CONFRONTED ?

Errors in Random Variate Generators

- Random variate generators might not produce random numbers from the desired distribution.

- Most of the non-conformation with the theoretical concepts are caused by:

  a. *Implementation errors*:- Mistakes in computer programs.

  b. *Error in design of algorithm*:- The proof of the theorem that claims the correctness of the algorithm is wrong.

  c. *Limitations of floating point numbers* and *Round off errors* in implementation of these algorithms in real world computers.

Examples of Errors

- A relevant example is the *Kinderman-Ramage* generator for normal distribution, in R, prior to version 1.6.
- In this, a line of code was overlooked by the programmer. On further research, it was detected that the in algorithm, a *rejection* line was missing. [2]
- Error in F distribution with df_1=1 and df_2 ~ 0.001, where:

  pf (1e100,1,.001)= 0.112, pf(1e200,1,.001)= 0.21,

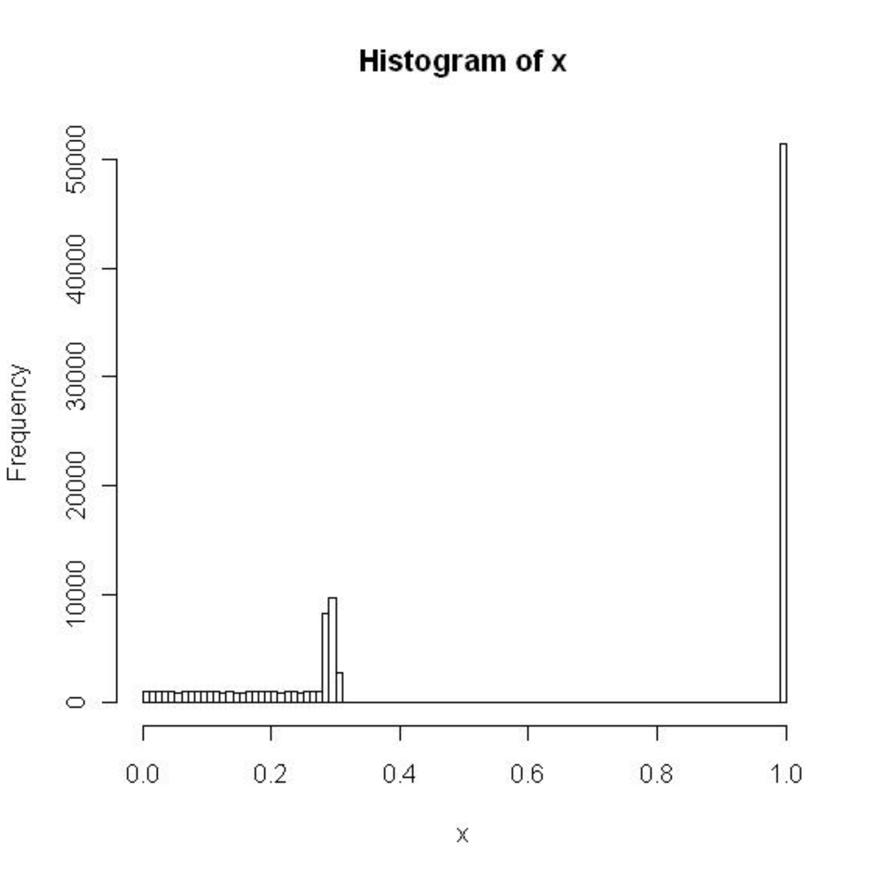
  pf(1e308,1,.001)= 0.30 and pf(>1e308,1,.001)= 1.

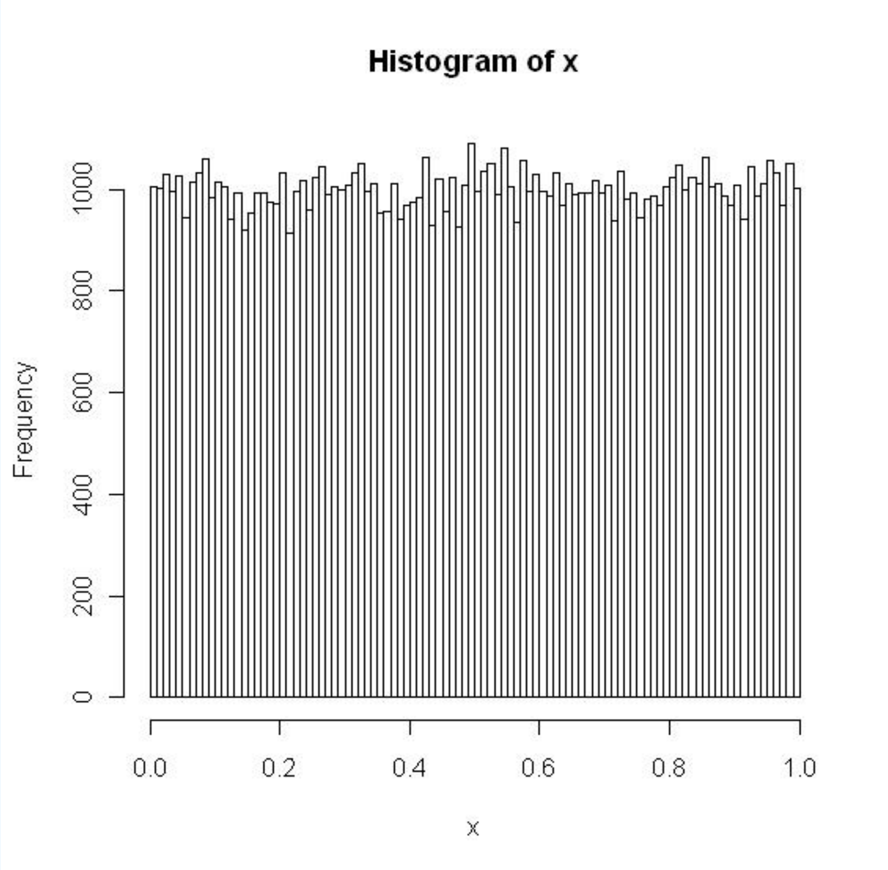
  Numbers greater than 10^308 cannot be handled due to limitation of floating point numbers.

Examples continued..

```
u<-runif(1e5)
X<-pbeta(qbeta(u,1,.01),1,.01)
hist(x,breaks=100)
```

```
u<-runif(1e5)
x<-pbeta(qbeta(u,1,1),1,1)
hist(x,breaks=100)
```





Potential Hazards of Errors in Random Variate Generators.

- In *Monte-Carlo simulations*, which depends on the quality of random variates generated, there might be serious errors due to error in generated random variates.

- In *Transformed Density Rejection* technique [1], computed hat function might not remain a valid hat function, especially in the tails of the distribution, due to round off errors in random variates.

THEORY OF ERROR DETECTION

1. Fundamental Property of Random Variates:

If a random point (X,Y) is uniformly distributed in the region G_f *between the graph of the density function f and the x-axis then X has density f*

2. Theory of Probability Integral Transform:

Let F(x) be a continuous cumulative distribution function(cdf) and U be a uniform U(0,1) random number. Then the random variate $X=F^{-1}(U)$ *has cdf F. Furthermore, if X has cdf F, then F(X) is uniformly distributed.*

Clustering of Random Variates in Histogram bins

There are two ways in which generated random variates can be clustered into bins of histogram:

- Transformation of generated random variates using cumulative distribution function.
- Transformation of uniform(0,1) scale by application of inverse cumulative distribution function.

Application of Cumulative Distribution Function F

- Cumulative distribution function F is applied on random variates.
- Theoretically, the transformed variates should follow U(0,1) distribution.
- The (0,1) scale is divided into equispaced bins of histogram and transformed variates clustered into it.
- Every bin should have *equal frequency count*, as probability of random variates entering a bin is equal.
- Very expensive due to large number of variates generated.

Application of Inverse Cumulative Distribution Function F^{-1}

- The (0,1) scale is divided into intervals which is equal to the specified number of bins of histogram.
- F^{-1} is applied to the limits of the intervals, which generates random variates having distribution F.
- Generated random variates are clustered into the bins of varying width.
- Every bin should have *equal frequency count*. as probability of random variates entering a bin is equal.
- Greatly reduces computational expense but exact inverse distribution not always available

Advantage of Clustering Random Variates into Histogram Bins

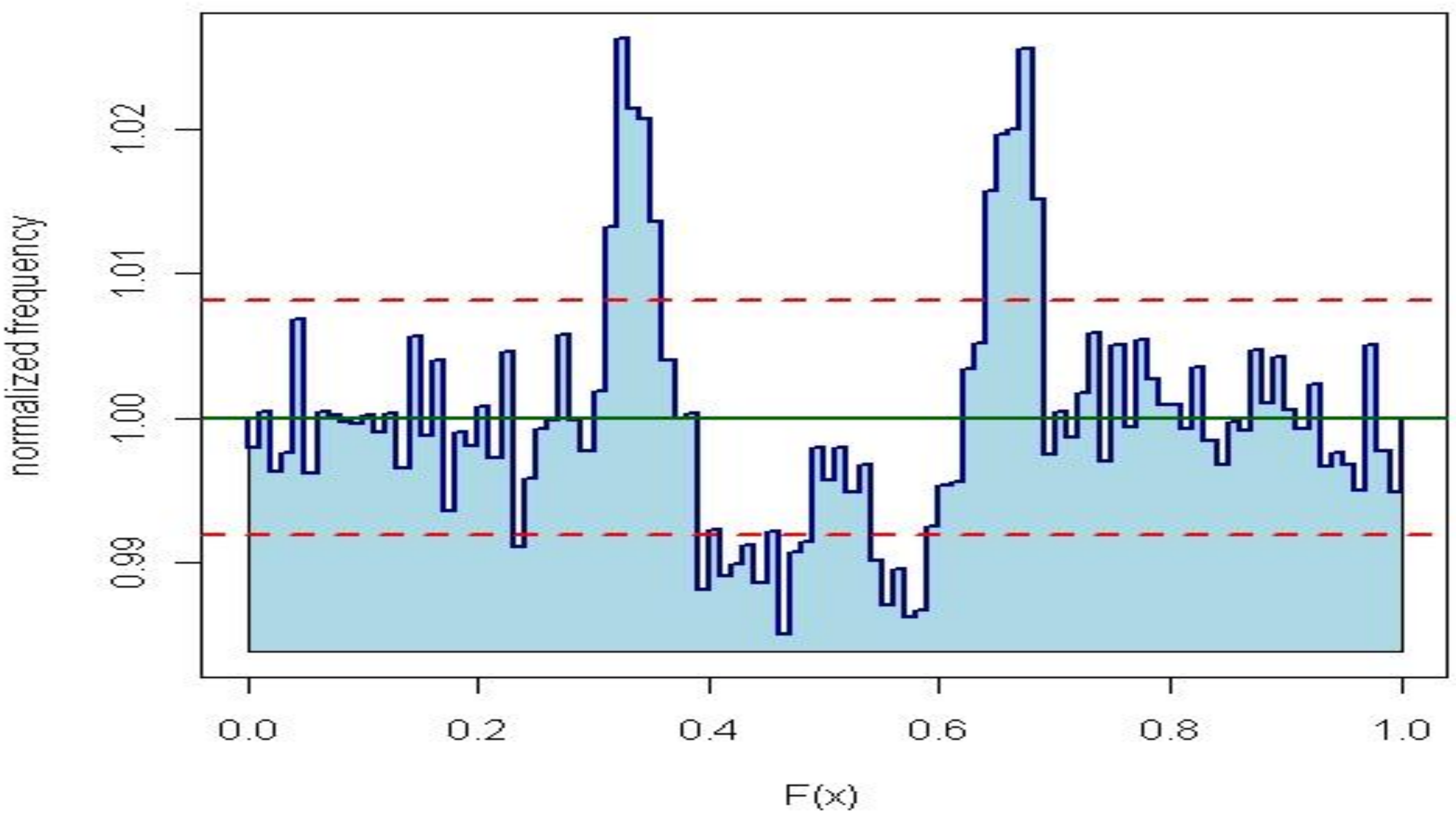
- Efficiency of execution.
- Can be visually inspected for significant bin deviation by plotting of histogram.
- Testing of errors effectively reduces to testing for equality of bins.
- Can handle very large samples, of more than 100 million points.

WHAT TESTS HAVE BEEN CONSIDERED?

Graphical Test

- Visual inspection of random variates histogram is a quick yet efficient technique for detection of errors.
- Plot zooms on the unit line of the normalized frequency histogram and draws the confidence lines for a specified significance level.
- It can be visually inspected whether there is any significant bin deviation from equality, indicating error in the random variates.

Histogram plot of normalized frequency of random variates generated by Buggy Kinderman-Ramage generator in R.



Statistical Tests

- The following statistical tests were considered in this project:

  1. Chi-Square Goodness of Fit Test.
  2. Adjusted Residual or M-test.[3]
  3. Kolmogorov-Smirnov Test.
  4. Anderson-Darling Goodness of Fit Test.
  5. Test of Uniformity by Fisher or Level-2 chi-Square Test.[4]

Chi-Square Goodness of Fit Test

- Popular and efficient test.
- Used to check whether a sample of data comes from a population with specified distribution.
- Since frequency of each histogram bin is supposed to be equal, chi-square test was applied to check for any significant bin deviation.
- The test statistic is calculated as $\sum(O_i - E_i)^2/Ei$, which follows chi-square distribution; O_i being observed frequency of bin i, E_i being expected frequency, which is 1 (normalized frequency).
- p-value for test is reported.

Adjusted Residual or M-test

- Test for detecting outlying cells in the multinomial distribution.
- Developed by Fuchs,C. And Kenett, R. [3].
- Let n be a random vector from a multinomial distribution,

  $n=\{n_i : 1<=i<=k\}\sim mult(N,p)$, $N= \sum n_i$, $p_i >=0$, $\sum p_i =1$.
- In our case, n_i represents normalized frequency of bin i and k represents the number of bins.
- We test $H_o : p=p^{(o)}$ against $H_1 : p \neq p^{(o)}$, where $p^{(o)}$ is prespecified frequency vector. In our case, $p_i^{(o)}$ is equal to 1/k, for all p_i .

- Under the null hypothesis, n_i is asymptotically normally distributed with mean $N\,p_i^{(o)}$ and variance $N\,p_i^{(o)}(1- p_i^{(o)})$.

- The adjusted residuals Z_i are defined as:

$$Z_i = (n_i - N\,p_i^{(o)})/(N\,p_i^{(o)}(1- p_i^{(o)}))^{½}, \quad i=1,2,\ldots.k$$

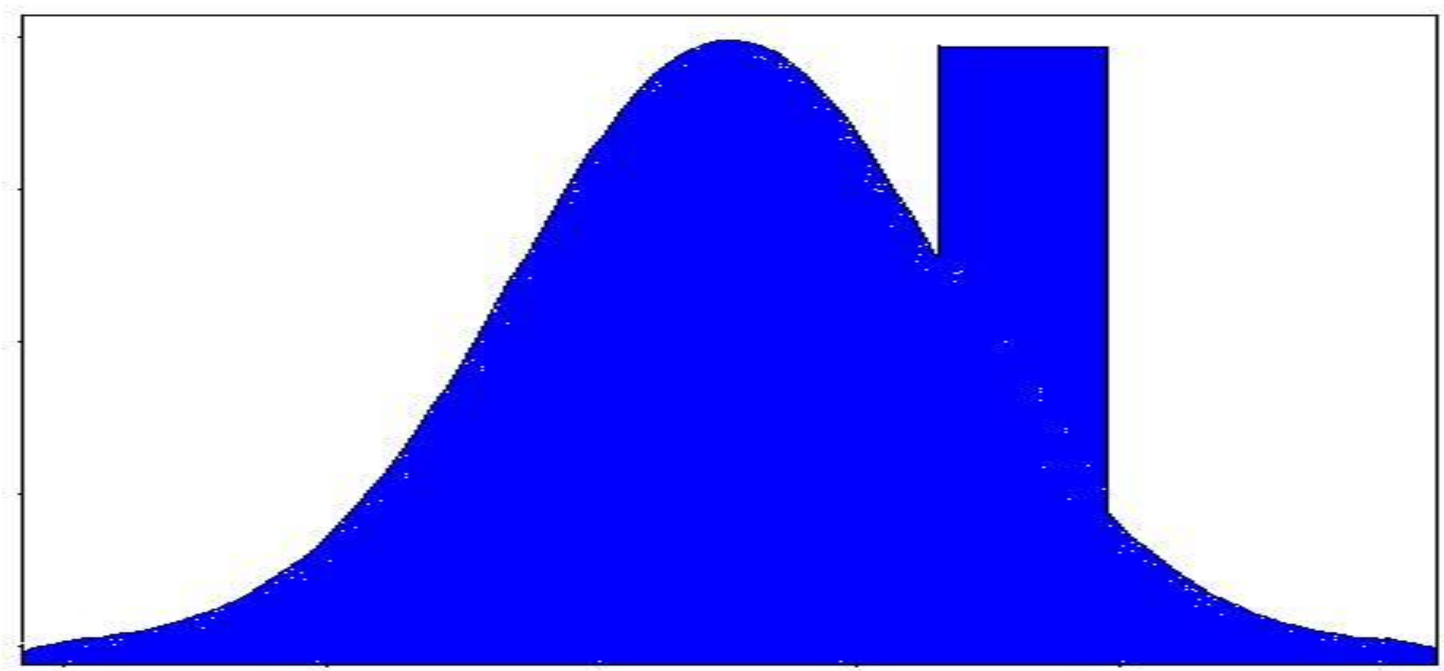
- The proposed M test for two-sided alternative, at significance level α, rejects the null hypothesis if $\max|Z_i| > M$, where $\{\Pr \max|Z_i| > M | H_o\}=\alpha$.

- The upper bound on M is calculated as $\Phi^{-1}\{1- \alpha/2k\}$.

- To maintain consistency with result generated from chi-square test, in our project, we calculate *p-value* from this test as $2*k*\{1-\text{pnorm}(\max|Z_i|)\}$

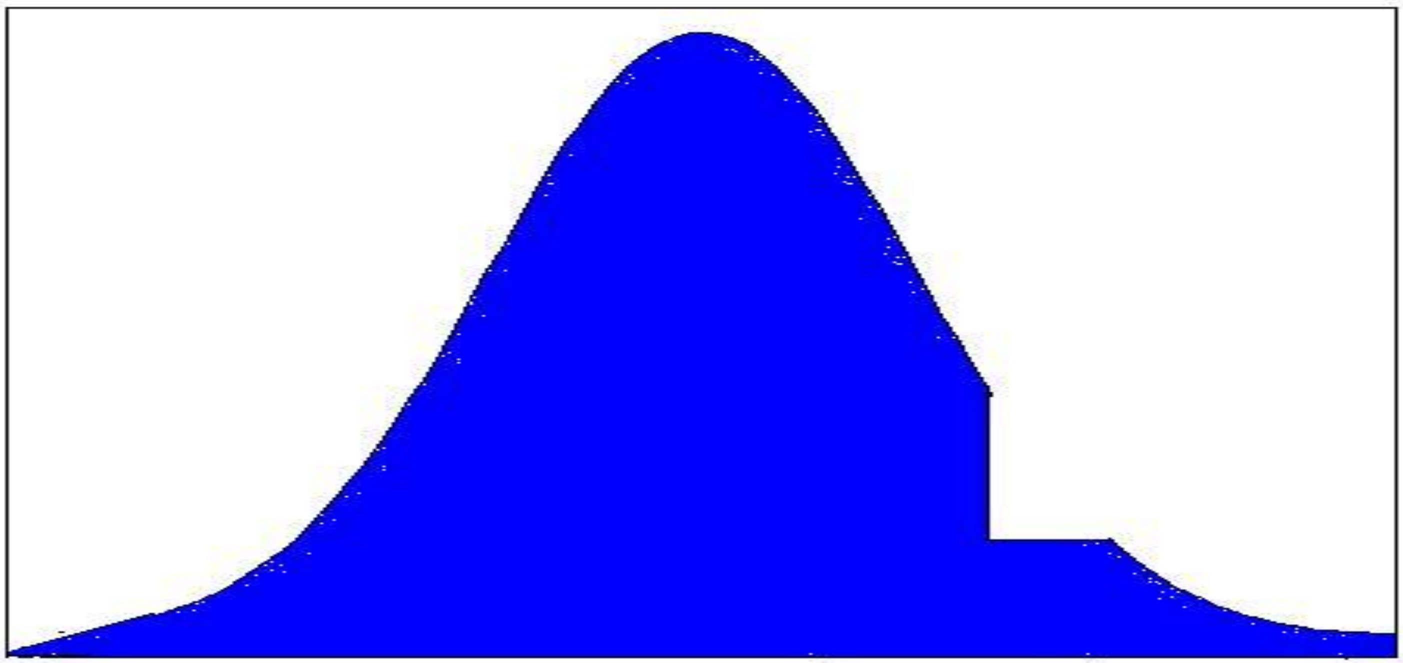
ARTIFICIALLY INTRODUCED ERRORS IN RANDOM VARIATE GENERATORS FOR TESTING PURPOSE:

Perturbating Parent Distribution with Uniform Distribution(Additive)



- Sample of random variates generated from mixture of a parent distribution and uniform distribution.
- Probability of error is specified as *p*.
- Random variate generator of parent distribution is used to generate random variates with probability 1-p.
- Uniform distribution of varying width and placement forms the error distribution.
- Random variates are drawn from the uniform distribution with probability p.
- Total number of random variates generated is equal to specified sample size *n*.

Removing Part of Parent Distribution Uniformly



- Random variate generator of parent distribution is used to generate *n* random variates, where *n* is sample size.

- Uniform distribution of varying width and placement forms the error distribution.

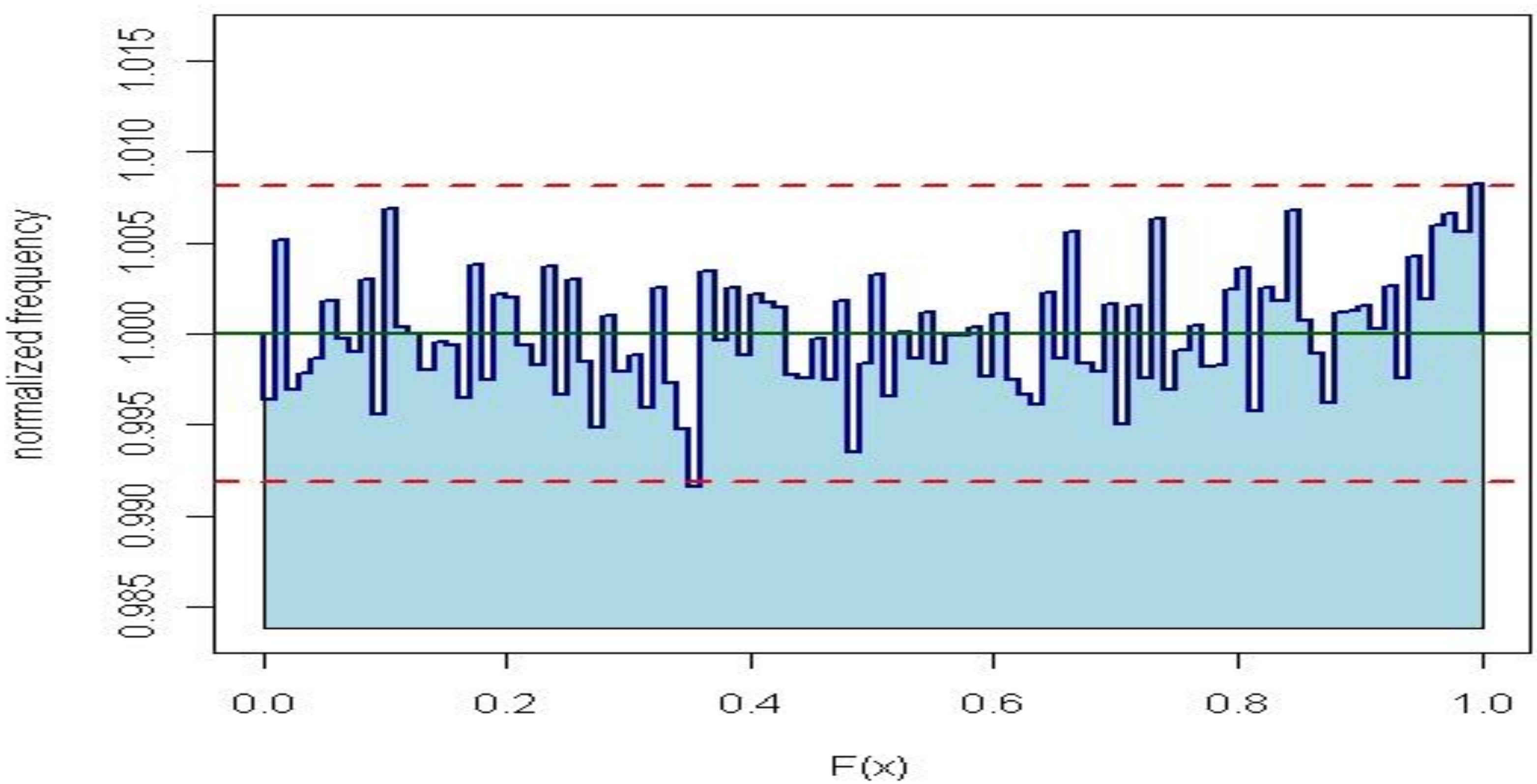
- Random variates which fall in the uniform distribution range are rejected with probability *p*.

- For the rejected random variates, new random variates are generated from the parent distribution.

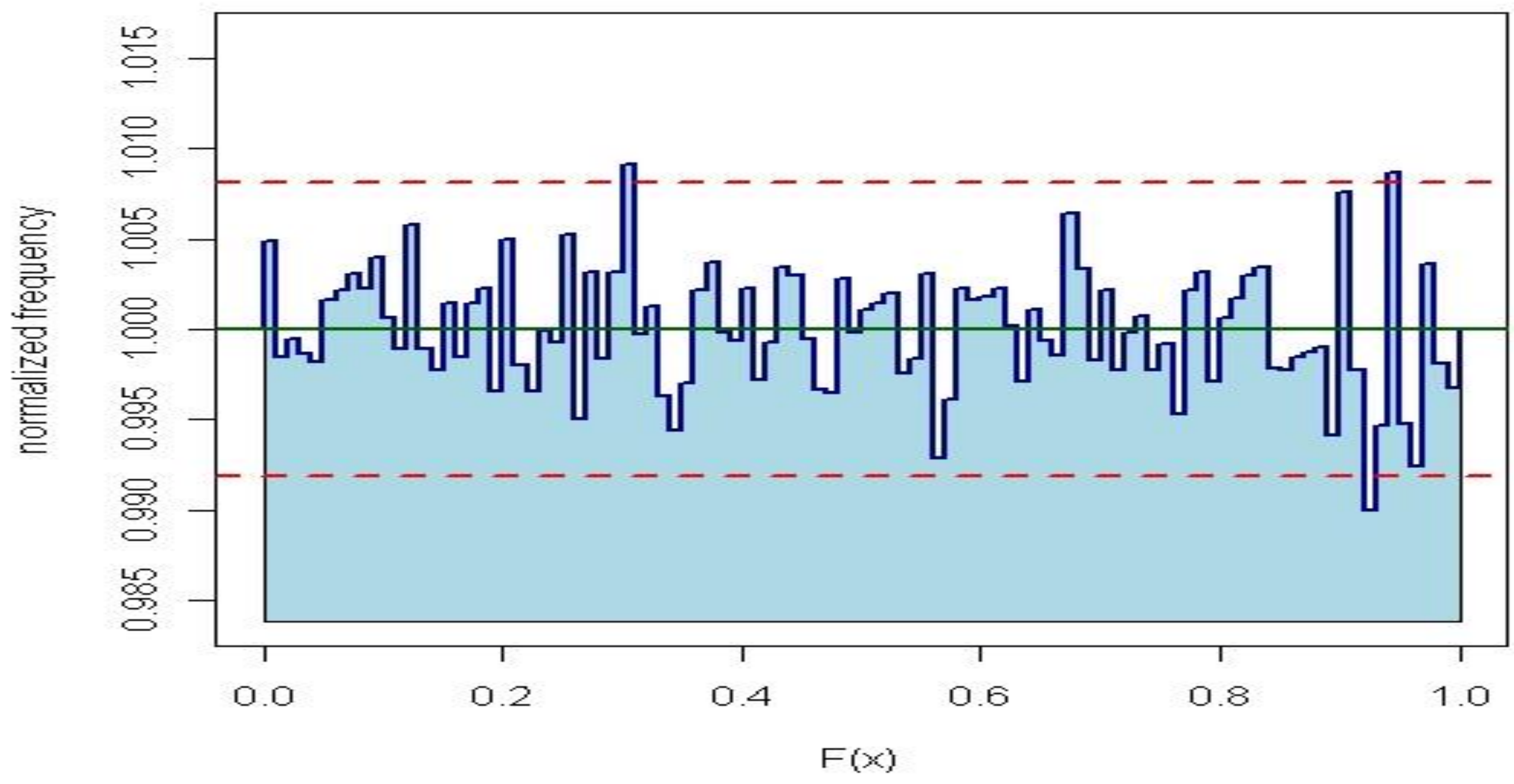
SIMULATION EXAMPLES

GRAPHICAL EXAMPLES

Standard Normal Distribution Perturbated Additively with Uniform(0,2.5), Probability of Error=0.001, bins=100, α=0.01,n=1e7



Standard Normal Distribution Perturbated Negatively with Uniform(1,2), Probability of Error=0.001, bins=100, α=0.01,n=1e7.



STATISTICAL TEST EXAMPLES

- The statistical tests that were conducted focused on standard normal distribution as parent distribution, perturbated *additively* by uniform distribution, of varying width, arbitrarily placed along the normal distribution.

- Two interesting observations were made:

1. Effect of Histogram breaks on the efficieny of the test.
2. Sensitivity of tests to error width and placement.

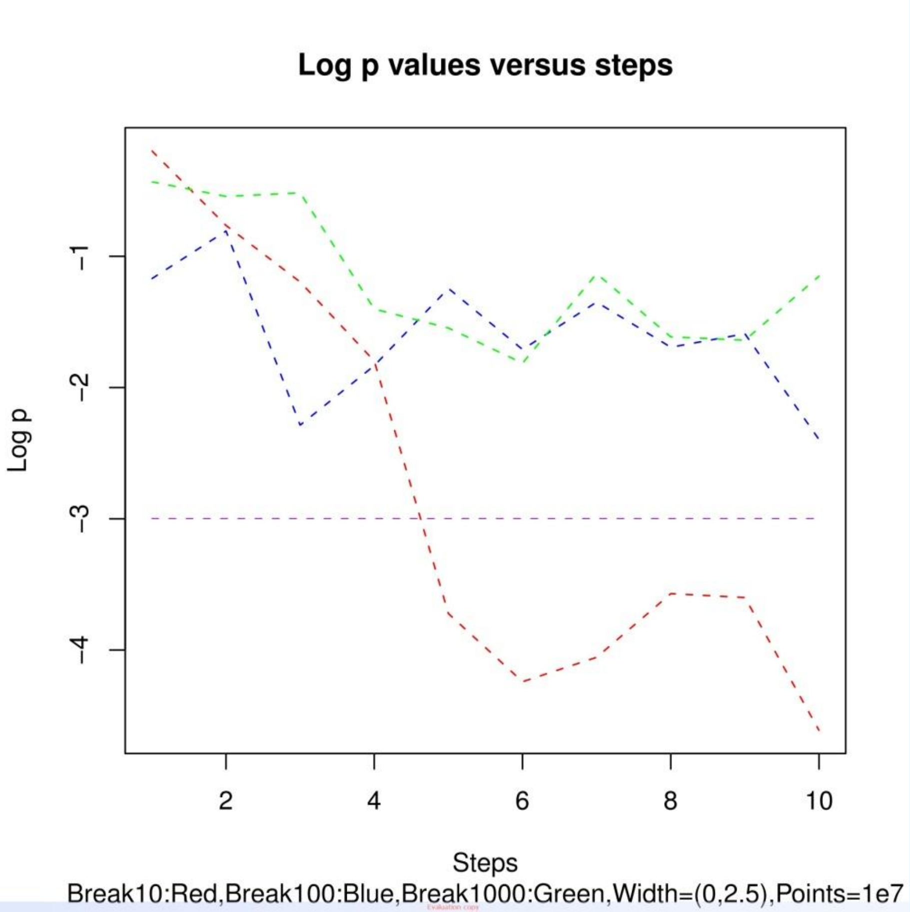
Effect of Histograms Breaks on Efficiency of Chi-Square Test

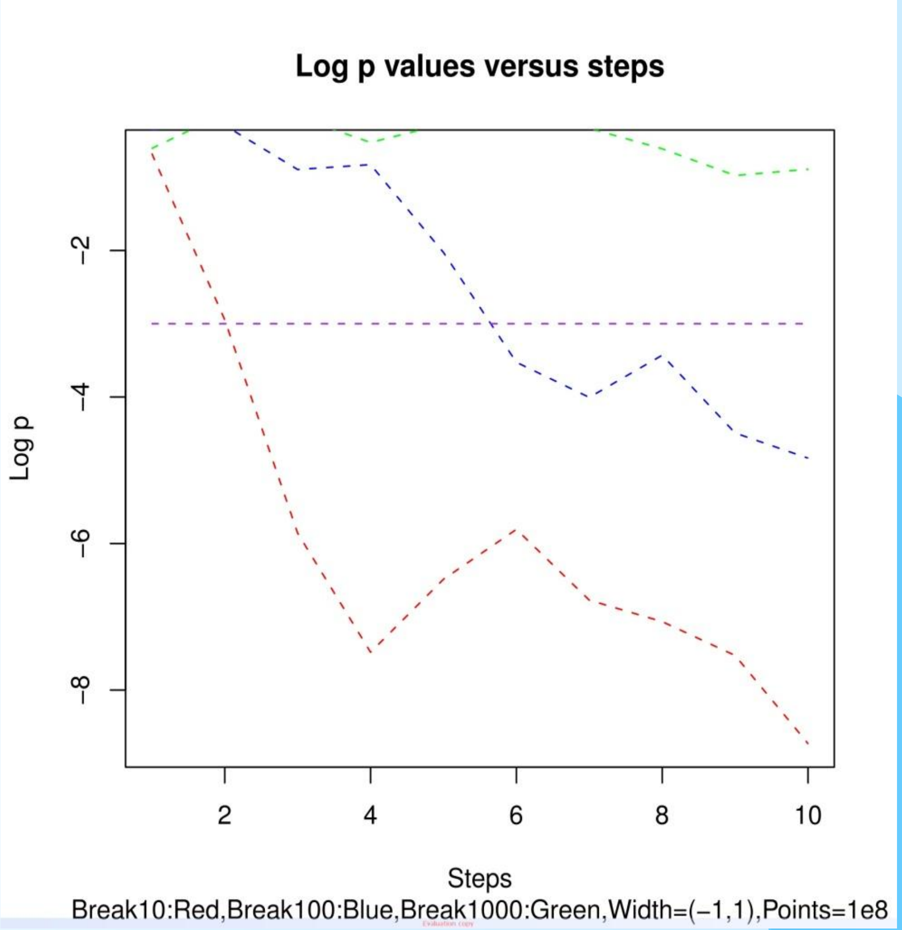
- Chi-square test was conducted on generated random vairates, with histogram breaks of 11,101,1001. Probability of error was kept fixed at p=0.001, and significance level α=0.001.

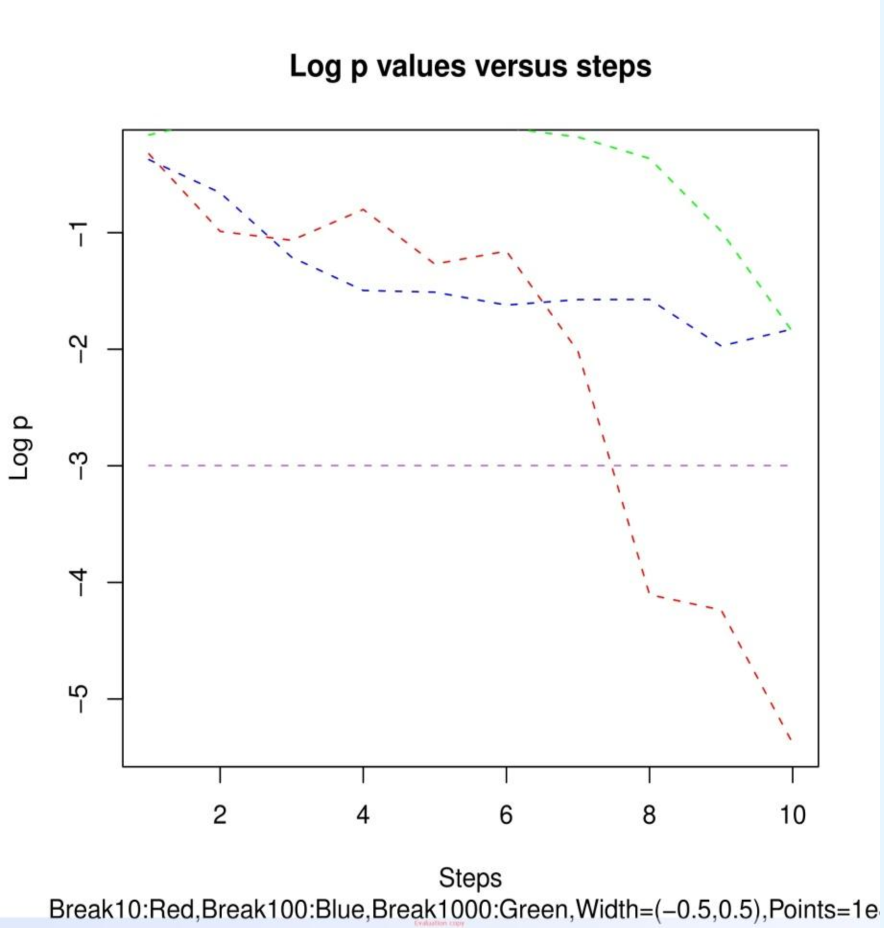
- The following tables and graphs will give examples of some of the select experiments.

- It was observed throughout that decreasing the number of breaks made the test more efficient in detecting errors.

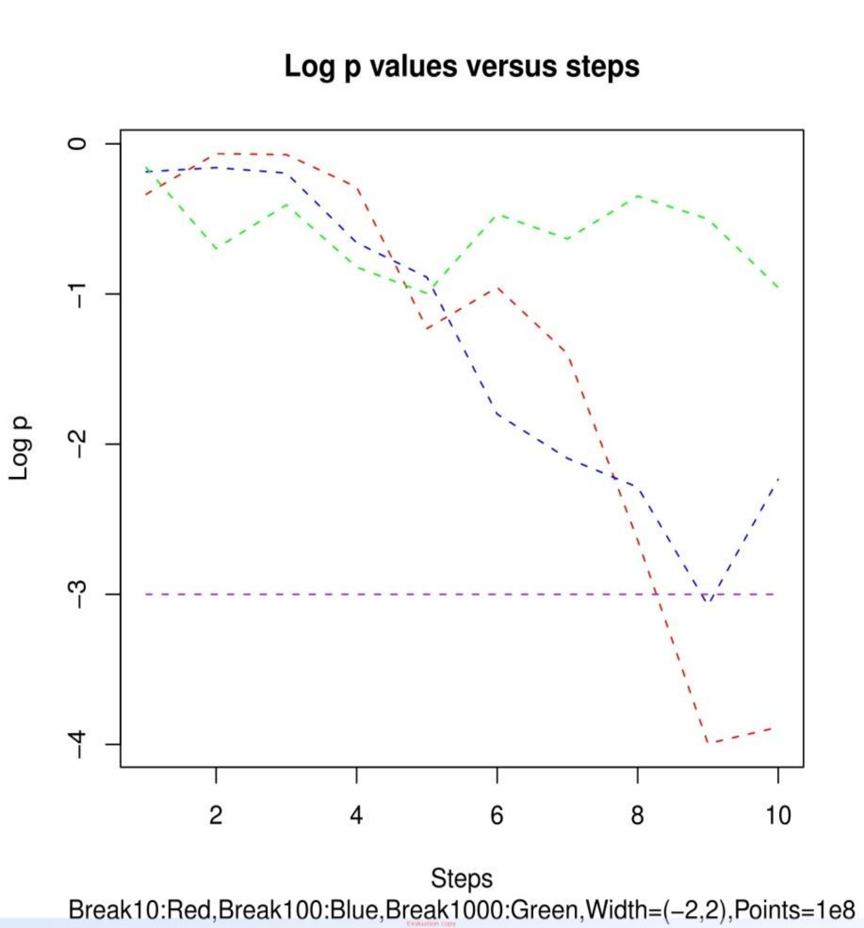
Width(0,2.5)	1	2	3	4	5	6	7	8	9	10
Breaks:10	0.636	0.172	0.063	0.016	1e-4	6e-5	9e-5	2e-4	2e-4	3e-5
Breaks:100	0.067	0.155	0.005	0.014	0.056	0.019	0.044	0.021	0.025	0.004
Breaks:1000	0.37	0.286	0.30	0.04	0.039	0.015	0.074	0.024	0.022	0.07
Width(-1,1)	1	2	3	4	5	6	7	8	9	10
Breaks:10	0.209	0.001	1.3e-6	3e-8	3e-7	1.5e-6	1.7e-7	9e-8	3e-8	2e-9
Breaks:1010	0.443	0.509	0.127	0.148	0.009	2e-4	1e-4	3e-4	3e-5	1e-5
Breaks:1000	0.247	0.737	0.658	0.298	0.529	0.758	0.464	0.244	0.105	0.128
Width(-0.5,0.5)	1	2	3	4	5	6	7	8	9	10
Breaks:10	0.477	0.103	0.086	0.158	0.054	0.069	0.009	8e-5	5e-5	4e-6
Breaks:100	0.424	0.219	0.062	0.032	0.031	0.024	0.027	0.027	0.011	0.015
Breaks:1000	0.687	0.949	0.924	0.953	0.919	0.792	0.661	0.433	0.103	0.014
Width(-2,2)	1	2	3	4	5	6	7	8	9	10
Breaks:10	0.461	0.861	0.847	0.515	0.059	0.111	0.039	0.002	1e-4	1e-4
Breaks:100	0.651	0.694	0.638	0.219	0.129	0.016	0.008	0.005	8e-4	0.006
Breaks:1000	0.697	0.200	0.392	0.151	0.101	0.339	0.232	0.448	0.315	0.110


Log p values versus steps
Log p
Steps
Break10:Red,Break100:Blue,Break1000:Green,Width=(0,2.5),Points=1e7


Log p values versus steps
Log p
Steps
Break10:Red,Break100:Blue,Break1000:Green,Width=(−1,1),Points=1e8



Log p values versus steps
Log p
−1
−2
−3
−4
−5
2
4
6
8
10
Steps
Break10:Red,Break100:Blue,Break1000:Green,Width=(−0.5,0.5),Points=1e



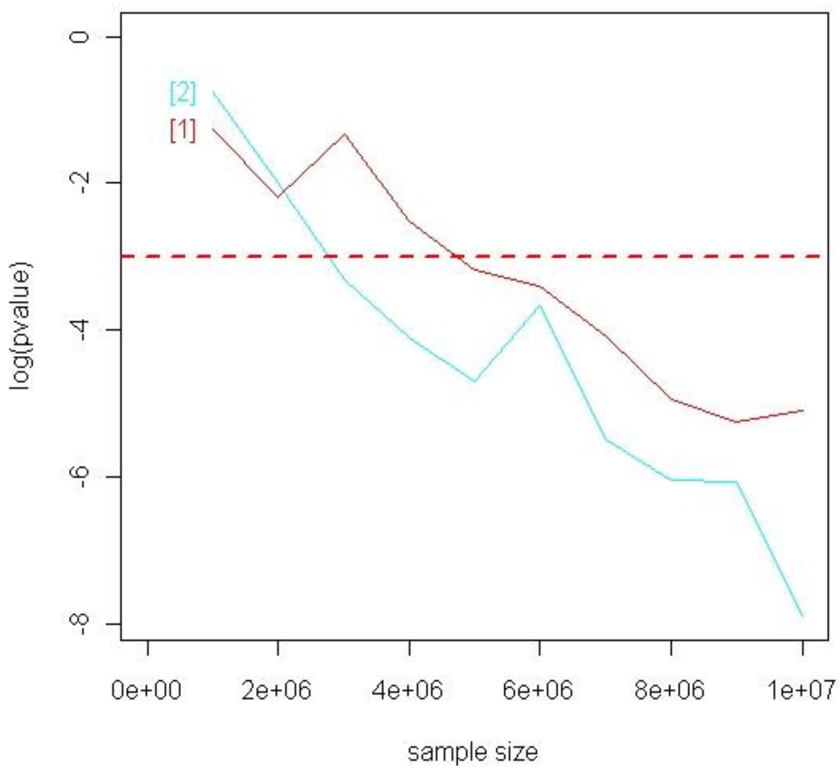
Log p values versus steps
Log p
0
−1
−2
−3
−4
2
4
6
8
10
Steps
Break10:Red,Break100:Blue,Break1000:Green,Width=(−2,2),Points=1e8

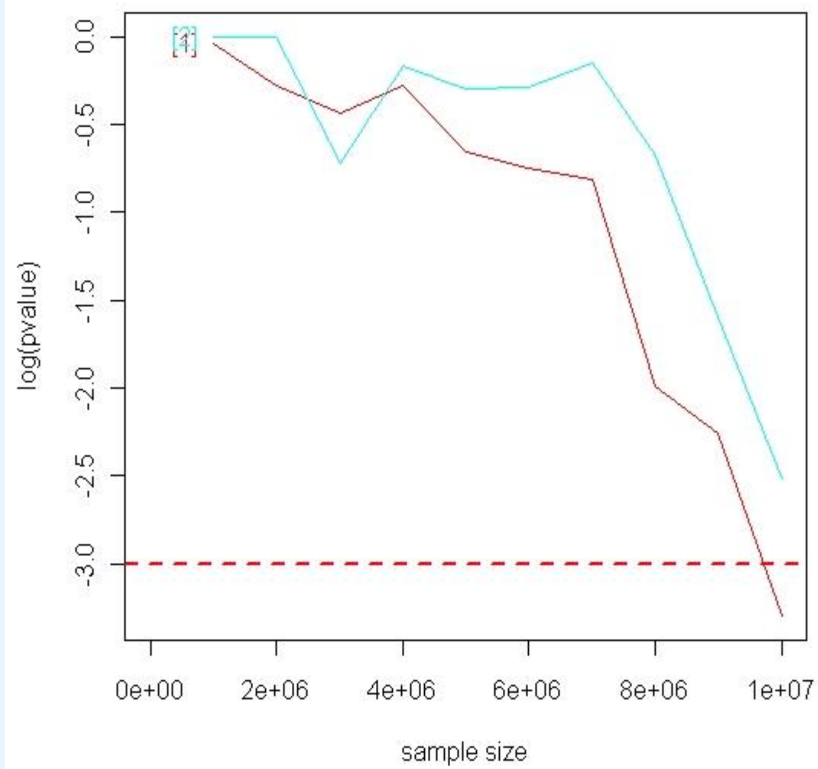
Sensitivity of Tests to error Width and Placement

- Both chi-square and M test was conducted to check the sensitivity of tests when uniform distribution was arbitrarily moved along the normal distribution.
- Probability of error was kept fixed at p=0.001.
- Around 0 point, as width of uniform distribution was increased, both tests became less effective in detecting errors.
- Tests were extremely efficient in detecting error when uniform distribution was placed in the tails of the normal distribution.

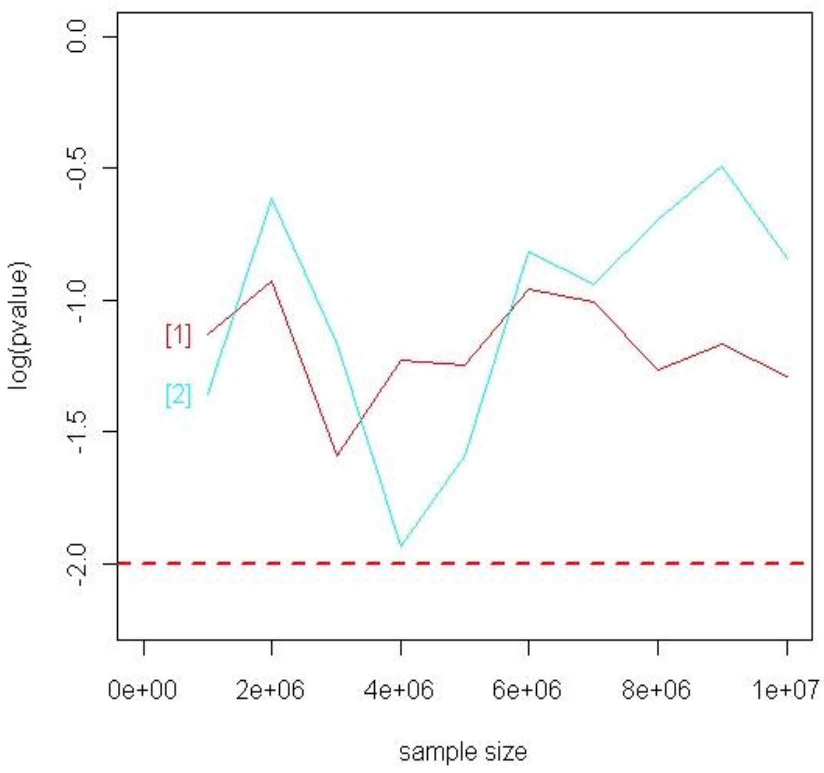
Width(-.2,.2)	1	2	3	4	5	6	7	8	9	10
Test 1: Chi-sq	0.054	0.006	0.047	0.003	6e-4	3e-4	8e-5	1e-5	5e-6	8e-6
Test 2: M	0.174	0.010	4e-4	8e-5	2e-5	2e-4	3e-6	9e-7	8e-7	1e-8
Width(-.4,.4)	1	2	3	4	5	6	7	8	9	10
Test 1: Chi-sq	0.911	0.529	0.365	0.522	0.219	0.179	0.155	0.010	0.005	5e-4
Test 2: M	1.00	1.00	0.188	0.674	0.506	0.511	0.710	0.209	0.025	0.003
Width(-.8,.8)	1	2	3	4	5	6	7	8	9	10
Test 1: Chi-sq	0.074	0.117	0.025	0.059	0.056	0.109	0.098	0.054	0.068	0.051
Test2: M	0.043	0.241	0.067	0.011	0.025	0.151	0.115	0.203	0.320	0.143
Width(1.96,4)	1	2	3	4	5	6	7	8	9	10
Test 1: Chi-sq	0.151	0.019	0.023	0.078	0.078	0.014	0.007	2e-4	4e-4	3e-5
Test2: M	0.782	0.003	0.001	3e-4	6e-7	2e-6	1e-6	2e-7	4e-7	2e-8
Width(0,2.5)	1	2	3	4	5	6	7	8	9	10
Test 1: Chi-sq	0.671	0.221	0.088	0.183	0.219	0.063	0.021	0.007	0.003	5e-4
Test 2: M	1.000	1.000	0.506	0.937	0.131	0.017	0.004	0.003	3e-4	3e-4

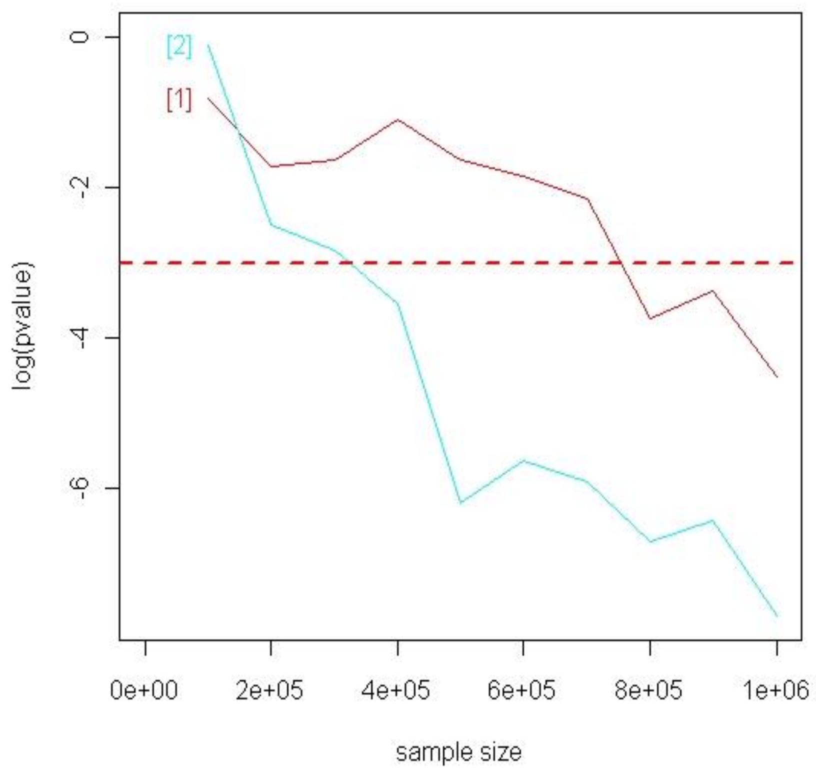
Red Line: Chi-square test
Blue Line: M test



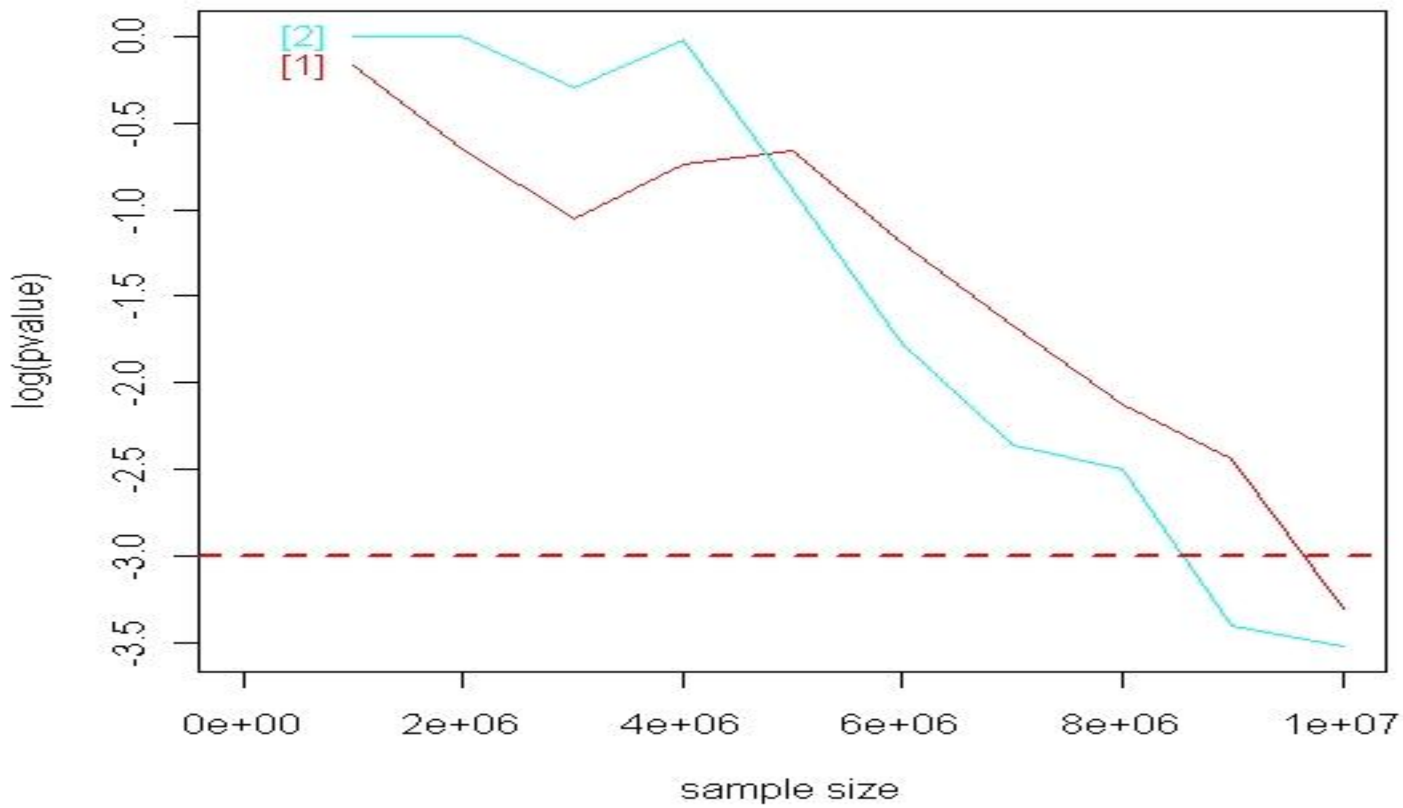


Red Line: Chi-square test
Blue Line: M test

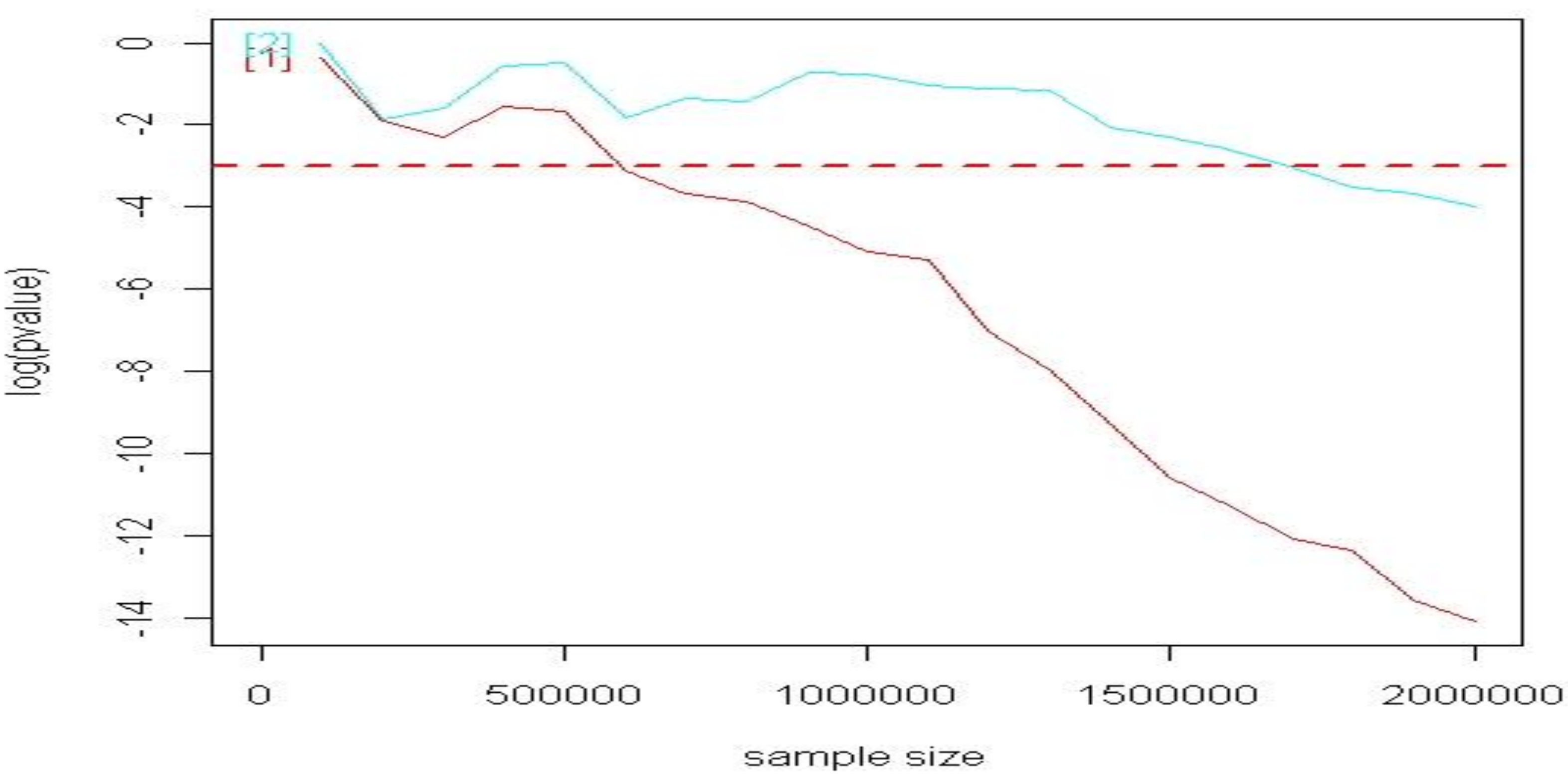




Red Line: Chi-square test
Blue Line: M test



Tests conducted with Buggy Kinderman-Ramage Generator.



DEVELOPMENT OF ERROR TESTING PACKAGE IN R

- Package Name : rvgtest
- Version : 0.1
- Title : Test suite for pseudo-random variate generators
- AUTHOR : Sougata Chaudhuri, Josef Leydold
- MAINTAINER : Josef Leydold
- LICENSE : GPL-2

FUNCTIONS AVAILABLE IN THE PACKAGE

rvgt.ftable(n, r = 10, rvg = rnorm, qdist = qnorm, ..., breaks = 101)

- Creates frequency table for random variate generator.
- Each row represents a histogram and each cell represents a bin of histogram.
- Break points of bins are uniformly distributed in u-scale,i.e, break points are calculated as u_i= i/(breaks-1), for i=0,1,2...(breaks-1) and points transformed into x-scale using qdist(i).
- The bins have equal probabilities.
- The frequency table can be now used to run tests or visualize possible errors in random variate generator.

rvgt.fhistplot(ftable, row = 1, alpha = 0.01)

- Plots normalized counts of the frequency table.
- The plot range is the union of 2 times the confidence intervals and the range of the normalized counts.
- The display zooms in on the expected value for the normalized counts.
- Also plots the confidence intervals calculated using alpha.
- Helps in visualizing significant bin deviations at certain significance level.

rvgt.rvghistplot(n, rvg = rnorm, qdist = qnorm, ..., breaks = 101, alpha = 0.01)

- Clusters random variates generated by *rvg* into histogram bins and plots normalized counts of the bins.
- No need to separately create frequency table.
- The plot range is the union of 2 times the confidence intervals and the range of the normalized counts.
- The display zooms in on the expected value for the normalized counts.
- Also plots the confidence intervals calculated using alpha.
- Helps in visualizing significant bin deviations at certain significance level.

rvgt.chisq(table)

- Performs chi-square test on *rvg* frequency table.
- A stepwise cumulation of row frequencies is performed (columnwise), and chi-square test is done on the columns, at every step.
- Each of the p-values is reported
- This allows for getting an idea of the power of the test.
- A list is returned, which contains information about the test and p-values calculated at every step.

rvgt.Mtest(table)

- Performs M-test on *rvg* frequency table.
- A stepwise cumulation of row frequencies is performed (columnwise), and M-test test is done on the columns, at every step.
- Each of the p-values is reported
- This allows for getting an idea of the power of the test.
- A list is returned, which contains information about the test and p-values calculated at every step.

rvgt.write(result, file)

- Function to write *result* to a file.
- *result* is a list generated from chi-square or M-test.
- A large number of tests can be done in batch mode and results can be written in one particular file.
- Subsequent reading back of files allows further calculation and plotting of p-values.

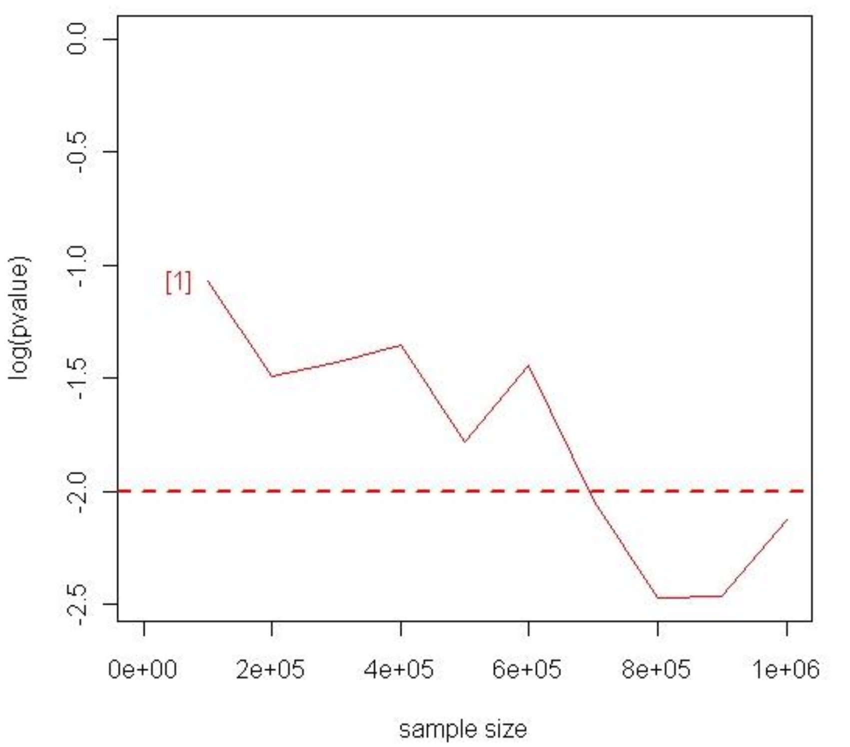
rvgt.read(file)

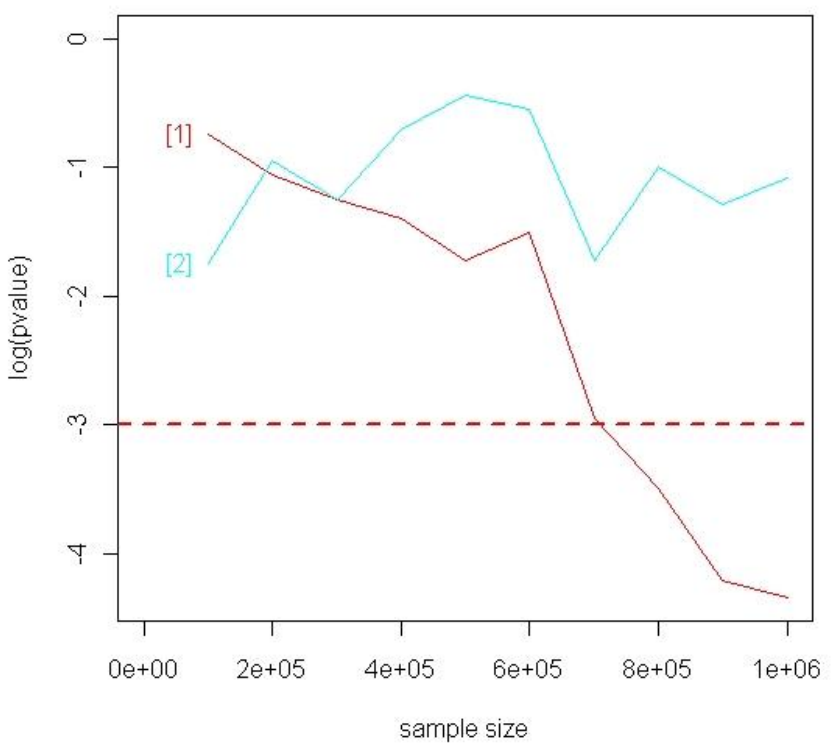
- Function to read data from a file.
- A list of list(s) is created, where every individual list contains the information and p-values of each experiment written in the file.
- *file* should be a valid name of an existing file.
- p-values can now be plotted or further calculations done.

rvgt.plot(result, alpha = 0.001)

- Plots log(10)*p-values* against sample size.
- *result* can be a list or list of lists, containing information about single/multiple experiments and corresponding p-values.
- For multiple experiments, p-values will be plotted in the same graph, with different colours.
- A line corresponding to log10(*alpha*)is displayed, for visually inspecting whether test has been able to detect errors at a given significance level.

Examples of plot function
Left: Single Experiment.
Right: Multiple experiment





rvgt.pertadd(n, rvg = rnorm, ..., min = 0, max = 1, p = 0.001)

- Generates random variates from a mixture of *rvg* and uniform distribution on the interval *(min,max)*.
- The uniform distribution is chosen with a probability *p*.
- By varying the width of uniform distribution *(min,max)* and probability of error *p*, different levels of *artificial* errors can be introduced.
- Allows to investigate power of test in detecting errors in random variate generators.
- A vector of size *n*, of random variates from the perturbated distribution, is returned.

rvgt.pertsub(n, rvg = rnorm, ..., min = 0, max = 1, p = 0.001)

- Generates random variates from *rvg* but rejects all points in the interval *(min,max)*, with probability *p*.
- By varying the width of uniform distribution *(min,max)* and probability of error *p*, different levels of *artificial* errors can be introduced.
- Allows to investigate power of test in detecting errors in random variate generators.
- A vector of size *n*, of random variates from the perturbated distribution, is returned.

REFERENCES

- Hörmann, W., Leydold, J. and Derflinger, G. (2004): *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin Heidelberg.
- Tirler, G., Dalgaard, P., Hormann, W. and Leydold, J. (2004): *An error in Kinderman-Ramage method and how to fix it*. Computational Statistics. 433-440.
- Fuchs, C. and Kenett,R. (1980). *A Test for Detecting Outlying Cells in the Multinomial distribution and Two-Way Contingency Tables.* Journal of American Statistical Association, Vol 75. 395-398.
- Fisher, R.A. (1967). *Statistical Methods for Research Workers.* 4th edition, New York.
- Stephens, M.A. (1986). Tests for the uniform distribution, p.358. in: R. B. D'Agostino and M. A. Stephens (eds.), Goodness-of-fit techniques, New York: Dekker.