TECHNOLOGY PARTNERZ

The Need for Speed

A detailed comparison of the top Monte-Carlo Simulation Tools for Microsoft Excel

With over 20 individual tests and 64 million trials, we have prepared a comprehensive comparison of the top Monte-Carlo Tool (Crystal Ball, ModelRisk, @Risk and Risk Solver) in terms of speed, accuracy and precision.

Eric Torkia
Executive Partner, Analytics Practice

September 2011

© Copyright 2011 by Technology Partnerz Ltd

This document is destined to be a discussion piece and Technology Partnerz Ltd. encourages its free distribution to all who might benefit from its contents.

The information contained in this document is proprietary to Technology Partnerz Ltd. and its authorized partners.

All rights reserved. *All* other *trademarks* are the property of their respective owners. No portion of the materials, including the models may be reproduced for commercial purposes. Any commercial application or reproduction of this material, other than in its original electronic format, cannot be undertaken without prior written consent from Technology Partnerz Ltd.

For more information on obtaining reproduction rights or customized educational white papers :

CONTACT US

Research & Development
Technology Partnerz Ltd.

550 Sherbrooke St. W., West Tower, Suite 1650
Montreal, Qc., Canada
H3A 1B9

- **888-879-8440** (Toll-Free)
- 514-278-2221 (Montreal)
- 416-607-6925 (Toronto)

Please send all faxes to 514-278-5060

Please send general inquiries to :

- info@technologypartnerz.com orinfo@crystalballservices.com

©Technology Partnerz Ltd, 2011.

Table of Contents

Introduction 3
Tests and Analysis 4
Test Environment 4
Our Test Model 4
The impact of correlation on the results 5
Test 1: Fitting Simulated Data 6
Test 2: Fitting Real Data 10
Test 3: No Fitting, Just Normal Distributions 15
Who's the fastest- analyzing the speed tests 18
Speed and correlation 19
Speed and the Number of Trials 20
Who is the fastest overall – Total Test Duration 21
Tips to improve simulation speeds 22
Conclusion 23
Accuracy and precision 23
Speed 23
PSI versus Excel-based simulation 23
64-bit 23
Downloadable Test results 24
Acknowledgments 24
Comments and feedback 24
Commentary on the test results by David Vose: 25
Follow-Up Notes by Eric Torkia 28

©Technology Partnerz Ltd, 2011.

Introduction

In this article, we are going to cover an aspect of Monte Carlo tools which we are sure will be of interest to many – speed, accuracy and precision. Our curiosity was piqued when we ran several simulations and noticed that the standard deviation was moving around at certain percentiles at different rates than others. Therefore, we started wondering how much volatility would naturally occur at various percentiles across different packages.

Our testing focused on two different aspects: accuracy and precision. We used the correlated returns model from our previous article as our test model. We ran the model 20 times at 10 000, 50 000 and 100 000 trials, resulting in 60 simulations per package. Furthermore, we ran three sets of tests, each of which removed an element that could throw the results off. The first test consisted of fitting data that had been generated using a Monte Carlo tool. The second test consisted of fitting data from actual data to eliminate fitting error that could potentially arise from using simulated data. The third test used normal distributions for all the assets in order to completely eliminate potential fitting error and put all the applications on the same footing.

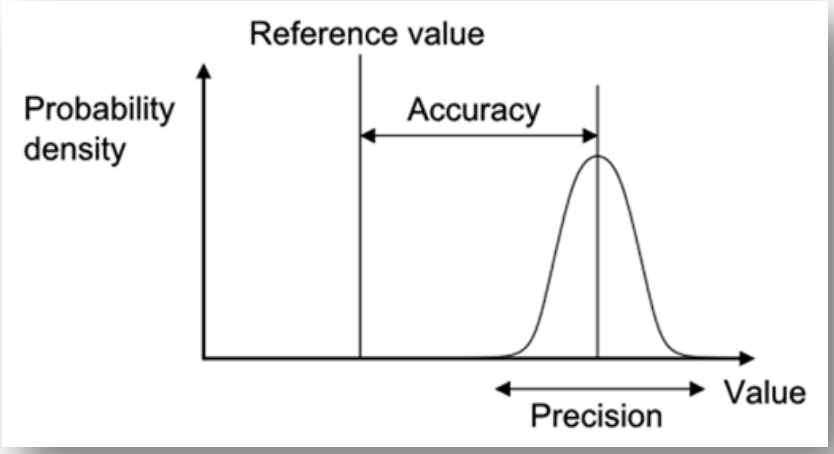


Figure 1: Precision Vs. Accuracy (source: http://en.wikipedia.org/wiki/Accuracy_and_precision)

In order to test for *accuracy* we **averaged out** the values of all 20 simulations for each percentile/package and compared the results against the calculated form using the Markowitz mean variance approach. As for *precision*, we looked at the **standard deviation** of each percentile/package at a given number of trials for all 20 simulations. For example, we would look at the standard deviation at the 99th percentile using a sample of 20 simulations at 10,000 trials each (200,000 trials total).

Our basic objective when analyzing a package for precision is whether it follows a certain set of statistical rules. A good example would be the expectation for a reduction of standard deviation as we increase the number of trials at a given percentile.

When we set up the script to test the accuracy of each tool we realized that it would be easy to do a performance comparison as well. This turned out to be an added bonus! During the scripted tests, we also took into account the execution time of the whole script and that of each simulation. We also evaluated the average trials/sec for both the test and the individual simulations.

©Technology Partnerz Ltd, 2011.

Tests and Analysis

Test Environment

Our test machine consisted of a Dell Dimensions XPS system with a 3.2 GHz dual core Pentium D, 4 GB of RAM, Windows XP and Microsoft Office 2010

Our Test Model

We are an investor that seeks to evaluate the risk in his portfolio. We have $250,000 to invest and we would like to make sure we have a better chance at making money than losing it. For purposes of the exercise, we have allocated our investment equally among each asset class at 25% a piece. We analyzed the rank correlation (using the Excel based method presented Copulas vs. Correlation) and prepared a correlation matrix that was used to correlate the returns distributions for each class. Please refer to our previous article on **correlating distributions** where we have prepared a video on how to do this with each package.

Portfolio Allocation

Aggregate Returns = Sum (Simulated Returns by asset class * Exposure by asset class)

Distributions fitted to historical returns data

Model Parameters	
Initial Value of Portfolio	$250,000.00
Aggregate Portfolio Returns	14.41%
% funds invested	100%

Portfolio Composition	Industrial	Energy	HighTech	Bonds
Assets Returns	12.51%	17.13%	21.93%	6.06%
Simulated Returns	12.51%	17.13%	21.93%	6.06%

Allocation Decisions				
% Invested / Exposure	25%	25%	25%	25%
Min Invest Req.	0	0	0	0
Max Invest Req.	100%	100%		

= Initial Value * (1+Aggregate Returns)

Final Portfolio Value	$286,019.58

	Industrial	Energy	HighTech	Bonds
Industrial	1.0000			
Energy	0.5694	1.0000		
HighTech	0.5409	0.7444	1.0000	
Bonds	0.0568	0.0616	0.0335	1.0000

Figure 2: portfolio returns model

©Technology Partnerz Ltd, 2011.

Since the applications fit the data using mostly normal distributions, we chose to use the Markowitz mean variance optimization method as a benchmark for the results at various numbers of trials (10k, 50k, and 100k).

With the Markowitz benchmarking data in hand, the first thing to do is contrast correlated versus uncorrelated results.

Table 1: Markowitz portfolio returns values at various percentiles

Covariance

	Industrial	Energy	HighTech	Bonds
Industrial	0.020396777	0.019290891	0.024015655	0.000126758
Energy	0.019290891	0.051079388	0.052452310	0.000103142
HighTech	0.024015655	0.052452310	0.089890878	0.000082357
Bonds	0.000126758	0.000103142	0.000082357	0.000102357

Benchmark Portfolio Returns

Mean	14.41%
Variance	2.21%
Std. Dev.	14.87%

Percentile	Returns	Calculated Value	Uncorrelated
1.00%	-20.2%	$199,558.74	$227,474
5.00%	-10.0%	$224,887.01	$244,159
10.00%	-4.6%	$238,389.42	$253,324
50.00%	14.4%	$286,019.28	$286,130
90.00%	33.5%	$333,649.15	$318,131
95.00%	38.9%	$347,151.56	$326,955
99.00%	49.0%	$372,479.82	$343,878
99.90%	60.3%	$400,870.19	$361,382
99.99%	69.7%	$424,239.44	$376,424

Clearly, the difference between the correlated and uncorrelated percentile values increases as we move away from the 50th percentile or Median. If we run this model without taking into account the correlation, we would be underestimating both the lower bound (loss) by almost 30 000$ and the upper bound by almost 50 000$. The consequence is that we are making decisions on the wrong range and risk profile – which could potentially be very good but mostly very bad because we are underestimating the downside risk.

The impact of correlation on the results

We ran each test with correlation turned on and off on all packages and tracked the results. This enabled us to both look at the results and the performance differences when handling correlation. The initial test was run only once with the correlation turned on while the two tests were run with it both on and off.

©Technology Partnerz Ltd, 2011.

Test 1: Fitting Simulated Data

The distributions were fitted using historical returns, which are included in the test sheet (For more information on how to fit distributions please refer to Excel Simulation Show-Down Part 2: Distribution Fitting). The data used for fitting both the distributions and correlations in the model was normally distributed because it was a doctored data set generated with a Monte-Carlo Engine. Note that normal distributions are unbounded and their tails can extend into infinity. Thus, the Min and the Max values of the output distribution generated by multiple simulations can vary greatly. Lastly, we made sure to run the scripts using the Monte Carlo method to ensure all the packages can be tested using the same approach.

Comparing Accuracy (Mean Results) by Percentile/Package: Fictional Fitted Historical Data

Table 2: Mean results by percentile for 20 simulations (by package) using simulated data

Test	AppEval
Trials	10000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$200,150.97	$225,382.95	$238,899.90	$286,003.52	$333,206.04	$346,596.29	$371,628.52	$399,590.57	$418,923.17
Crystal Ball	$199,735.04	$224,483.71	$238,196.75	$286,440.49	$333,345.31	$346,447.57	$370,685.45	$398,082.98	$416,993.22
ModelRisk	$198,063.66	$224,747.14	$238,657.86	$285,870.77	$333,816.22	$346,773.66	$371,582.34	$393,401.26	$402,206.98
RiskSolver	$200,235.86	$225,395.68	$238,807.40	$285,928.53	$333,026.46	$346,352.03	$371,277.51	$398,860.73	$419,971.28
Group Average	**$199,546.38**	**$225,002.37**	**$238,640.48**	**$286,060.83**	**$333,348.51**	**$346,542.39**	**$371,293.45**	**$397,483.88**	**$414,523.66**
Benchmark	$ 199,558.74	$ 224,887.01	$ 238,389.42	$ 286,019.28	$ 333,649.15	$ 347,151.56	$ 372,479.82	$ 400,870.19	$ 424,239.44

Test	AppEval
Trials	50000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$200,211.47	$225,405.17	$238,792.23	$286,053.88	$333,261.85	$346,707.28	$371,693.38	$399,390.72	$421,802.79
Crystal Ball	$199,250.39	$224,336.44	$238,053.69	$286,318.09	$333,206.55	$346,301.54	$370,999.63	$399,346.11	$422,167.46
ModelRisk	$199,092.04	$225,203.74	$238,443.17	$285,793.68	$333,625.44	$346,476.85	$370,491.60	$391,620.41	$404,340.59
RiskSolver	$200,487.66	$225,569.96	$238,892.24	$286,028.87	$333,129.85	$346,465.95	$371,536.35	$399,703.15	$421,148.97
Group Average	**$199,760.39**	**$225,128.83**	**$238,545.33**	**$286,048.63**	**$333,305.92**	**$346,487.91**	**$371,180.24**	**$397,515.10**	**$417,364.95**
Benchmark	$ 199,558.74	$ 224,887.01	$ 238,389.42	$ 286,019.28	$ 333,649.15	$ 347,151.56	$ 372,479.82	$ 400,870.19	$ 424,239.44

Test	AppEval
Trials	100000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$200,348.65	$225,384.05	$238,830.21	$286,004.80	$333,218.85	$346,622.69	$371,707.84	$400,078.28	$422,346.34
Crystal Ball	$199,329.99	$224,530.27	$238,153.39	$286,290.18	$333,330.31	$346,515.68	$371,166.04	$398,760.30	$422,316.01
ModelRisk	$198,692.23	$225,108.49	$238,529.40	$285,628.09	$333,477.94	$347,502.95	$372,037.54	$395,424.88	$408,044.56
RiskSolver	$200,441.75	$225,497.47	$238,887.20	$286,007.54	$333,137.63	$346,462.08	$371,484.36	$399,538.59	$422,585.54
Group Average	**$199,703.16**	**$225,130.07**	**$238,600.05**	**$285,982.65**	**$333,291.18**	**$346,775.85**	**$371,598.94**	**$398,450.51**	**$418,823.12**
Benchmark	$ 199,558.74	$ 224,887.01	$ 238,389.42	$ 286,019.28	$ 333,649.15	$ 347,151.56	$ 372,479.82	$ 400,870.19	$ 424,239.44

When fitting simulated data, we can see that ModelRisk is a clear outlier in terms of accuracy against the benchmark values.

This issue seems odd and could pose some serious issues because an analyst, from time to time, will be confronted with fitting simulated data. What is stranger still is that as we increase the number of trials -

©Technology Partnerz Ltd, 2011.

we only see moderate improvement in the outlier percentiles such as 99.9 and 99.99. Fortunately, as we will see in further tests the accuracy problem does not persist with ModelRisk.

It turns out, that using real data will have an impact on the final fitting result. Actually ModelRisk struggled with fitting simulated data (potentially) because of its implementation of information criteria versus goodness of fit methods. The other three packages use goodness of fit to estimate and rank potential distribution fits. Anecdotally, ModelRisk seems to have a very high degree of accuracy when picking out the right distribution and it is this very sensitivity that could make it inappropriate to fit simulated data with a low number of samples. You will be less prone to these issues with the other packages regarding simulated data.

Analyzing precision using simulated data

Test	AppEval
Correlation	On
Package	@RISK

Values	Column Labels 10000	50000	100000
StdDev of 1	$656.86	$310.23	$269.13
StdDev of 5	$511.19	$256.25	$155.48
StdDev of 10	$314.56	$122.23	$72.92
StdDev of 50	$179.09	$88.62	$51.86
StdDev of 90	$252.71	$114.62	$114.29
StdDev of 95	$502.11	$236.76	$159.61
StdDev of 99	$713.21	$364.13	$225.41
StdDev of 999	$2,412.86	$1,165.08	$650.81
StdDev of 9999	$5,526.26	$3,293.16	$2,174.65

Test	AppEval
Correlation	On
Package	Crystal Ball

Values	Column Labels 10000	50000	100000
StdDev of 1	$1,064.11	$676.18	$482.24
StdDev of 5	$574.51	$378.22	$194.73
StdDev of 10	$414.94	$279.53	$221.97
StdDev of 50	$461.73	$162.75	$148.58
StdDev of 90	$738.94	$318.02	$197.91
StdDev of 95	$895.03	$363.91	$196.64
StdDev of 99	$1,485.22	$443.09	$394.01
StdDev of 999	$3,206.43	$1,236.09	$997.94
StdDev of 9999	$6,832.71	$4,977.88	$2,566.46

Test	AppEval
Correlation	On
Package	ModelRisk

Values	Column Labels 10000	50000	100000
StdDev of 1	$5,389.21	$3,125.44	$4,964.51
StdDev of 5	$2,164.04	$1,746.06	$2,467.70
StdDev of 10	$1,454.23	$1,490.79	$1,208.47
StdDev of 50	$1,714.82	$1,646.39	$1,784.32
StdDev of 90	$1,758.51	$2,215.37	$1,753.26
StdDev of 95	$2,268.58	$2,691.36	$1,753.54
StdDev of 99	$3,754.09	$2,856.45	$3,684.09
StdDev of 999	$7,328.62	$7,106.30	$6,199.80
StdDev of 9999	$10,520.03	$10,696.90	$10,146.50

Test	AppEval
Correlation	On
Package	RiskSolver

Values	Column Labels 10000	50000	100000
StdDev of 1	$1,009.38	$467.54	$297.59
StdDev of 5	$577.10	$301.96	$156.60
StdDev of 10	$409.88	$228.10	$144.04
StdDev of 50	$222.52	$133.90	$95.94
StdDev of 90	$469.00	$219.28	$118.37
StdDev of 95	$473.42	$255.25	$153.53
StdDev of 99	$459.38	$465.53	$229.36
StdDev of 999	$2,329.80	$829.85	$675.45
StdDev of 9999	$4,626.72	$2,012.76	$1,695.16

Table 3: Standard deviation/package/number of trials (correlation on) - real data test

Note on how to analyze the tables

Each one of the tables presented above covers one package and presents the standard deviation of 20 samples for selected percentiles at specific number of simulation trials. In the rightmost column, you will find a spark line histogram which presents a graphical of the results. The light blue column represents the highest value in the data set.

©Technology Partnerz Ltd, 2011.

Theoretically, we should expect a consistent reduction in standard deviation as we increase the number of trials. Based on conventional statistical wisdom, the light blue bars (or highest standard deviation) should be the first bar in the series given that that is the standard deviation at the lowest number of trials tested. The second column should have the second highest value and the third the lowest.

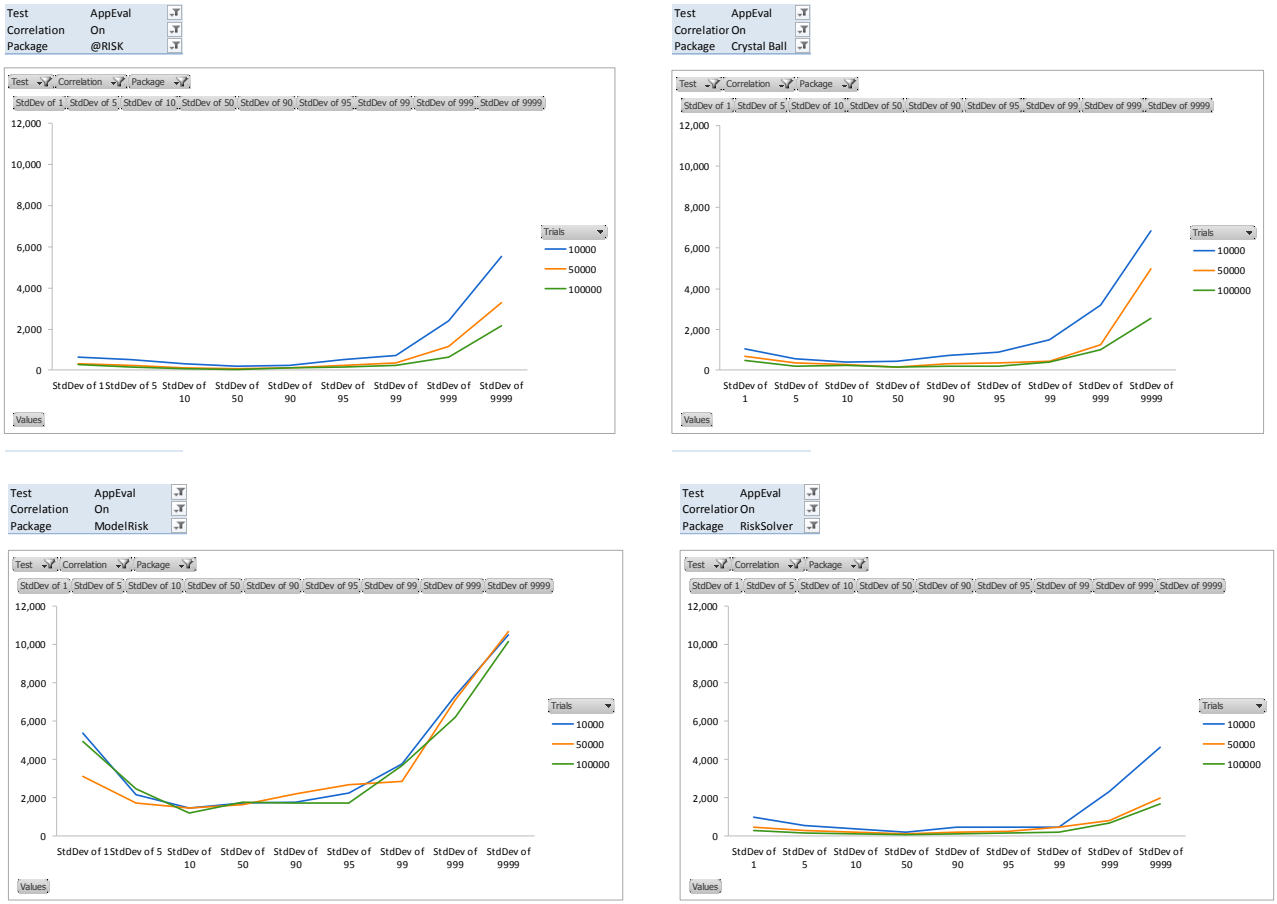


Figure 3: standard deviation by percentile/package - simulated data test

When analyzing the following graphs, one will notice that all the graphs (except one) have one common feature: the standard deviation goes down as we increase the number of trials. Again, ModelRisk by Vose software is a clear outlier in this regard. We find it very bizarre that there is no reduction in standard deviation as you increase the number of trials at certain specific outlier percentiles such as those at 99 and beyond. We have submitted this concern to Vose software and we look forward to hearing from them back soon. On the other hand, when we look at the other packages such as Crystal Ball, Risk Solver, @RISK -they all demonstrate a consistent reduction in standard deviation as you go further and further out in the tail.

> *"The Central Limit Theorem tells us that the expected variation of the* ***mean*** *is about ±2 times standard deviation divided by square root of number of iterations. But the relationships are not so simple for percentiles. And the further out you get, toward the zeroth or 100th percentile, the more variation you expect to see.*

©Technology Partnerz Ltd, 2011.

> *Why is that? Well, most of the data are close to the center of the distribution, which means that out in the tails it is relatively sparse. When you get out to the min or max, you are looking at one iteration. That might be nominally a statistic of the simulated distribution, but really it's just whichever iteration happened to be lowest or highest in a particular run. In a normal distribution, or any other distribution with a tail, these are essentially meaningless numbers because the true min and max are ±infinity.*
>
> *Take a normal distribution such as your Normal(286000,36800) with 100000 iterations. The way Latin Hypercube works, you would expect one iteration, give or take, in each of the 100000 equally probable intervals. The lowest .0001 of probability is between -infinity and 149140, so you would expect about 10 iterations in that interval. But that's a huge interval, so huge variation would be expected in whatever iteration happens to be the lowest from run to run. By contrast, the .0001 of probability near the 40th %ile is 276677 to 27686, so you would expect to find about ten iterations in that small range, and very little variation in the 40th %ile from run to run."- **Palisade Corporation.***

Risk Solver handled the simulated data in a very impressive way - starting at 50,000 trials you get a significant improvement in standard deviation similar to the levels you would get at 100,000 trials. Crystal ball and @RISK both share similar patterns where precision improves tremendously between 50,000 and 100,000 trials.

©Technology Partnerz Ltd, 2011.

Test 2: Fitting Real Data

When analyzing the results from the simulated data, we were somewhat surprised at how some of the packages responded. It was therefore recommended by one of the vendors that we use actual data to perform further fitting tests. We sourced data a series of industrial indices from a stock exchange and calculated the log returns as well as the volatility for the entire data set. As with the previous test, we made sure to run all the packages using the Monte Carlo method and not Latin hypercube.

Comparing Accuracy (Mean Results) by Percentile/Package: Real Fitted Historical Data

Table 4: Mean results by percentile for 20 simulations (by package) using real data

Test	RealData
Trials	10000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 99	Average of 9999
@RISK	$226,398.76	$234,986.73	$239,169.68	$251,832.53	$263,160.83	$266,493.81	$272,972.27	$281,209.84	$288,045.14
Crystal Ball	$226,958.64	$235,284.08	$239,357.55	$251,879.75	$263,366.94	$266,755.91	$273,506.28	$282,206.39	$289,149.49
ModelRisk	$224,393.46	$233,683.50	$238,093.53	$251,683.35	$263,637.15	$267,056.23	$273,668.35	$281,405.36	$288,437.83
RiskSolver	$223,921.86	$233,972.43	$238,572.05	$251,898.65	$262,565.13	$265,408.48	$270,914.66	$277,275.11	$282,640.16
Group Average	**$225,418.18**	**$234,481.69**	**$238,798.20**	**$251,823.57**	**$263,182.51**	**$266,428.61**	**$272,765.39**	**$280,524.17**	**$287,068.15**
Benchmark	$ 227,185.30	$ 234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$268,875.20	$276,029.57	$284,048.87	$ 290,649.88

Test	RealData
Trials	50000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 99	Average of 9999
@RISK	$226,594.00	$235,014.52	$239,109.04	$251,822.71	$263,164.47	$266,449.10	$273,004.73	$281,554.96	$288,602.34
Crystal Ball	$227,076.47	$235,319.03	$239,351.73	$251,864.25	$263,331.22	$266,706.49	$273,637.44	$282,360.45	$289,741.37
ModelRisk	$224,619.03	$233,696.03	$238,119.49	$251,730.38	$263,647.07	$267,075.69	$273,549.24	$281,034.01	$287,670.73
RiskSolver	$224,065.13	$233,871.02	$238,485.25	$251,849.89	$262,478.07	$265,347.52	$270,826.14	$277,699.84	$283,005.18
Group Average	**$225,588.66**	**$234,475.15**	**$238,766.37**	**$251,816.81**	**$263,155.21**	**$266,394.70**	**$272,754.39**	**$280,662.32**	**$287,254.90**
Benchmark	$ 227,185.30	$ 234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$268,875.20	$276,029.57	$284,048.87	$ 290,649.88

Test	RealData
Trials	100000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 99	Average of 9999
@RISK	$226,472.31	$234,994.25	$239,122.30	$251,814.39	$263,169.87	$266,460.52	$272,997.73	$281,189.80	$288,690.98
Crystal Ball	$227,124.52	$235,341.54	$239,342.22	$251,858.22	$263,324.91	$266,706.17	$273,583.34	$282,295.73	$290,361.37
ModelRisk	$224,393.40	$233,715.16	$238,152.39	$251,734.47	$263,729.17	$267,053.31	$273,613.34	$281,458.77	$287,070.46
RiskSolver	$224,157.71	$233,981.56	$238,555.45	$251,904.49	$262,504.35	$265,339.92	$270,976.17	$277,445.13	$282,657.38
Group Average	**$225,536.98**	**$234,528.46**	**$238,793.09**	**$251,830.29**	**$263,168.05**	**$266,389.98**	**$272,771.60**	**$280,597.36**	**$287,198.24**
Benchmark	$ 227,185.30	$ 234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$268,875.20	$276,029.57	$284,048.87	$ 290,649.88

In the second round of tests, we can see that Risk Solver is now the one that is having issues with fitting. Though the differences are not tremendous, they are enough to be different from the others. This reinforces how much care an analyst must take when fitting data in order to get consistent results.

©Technology Partnerz Ltd, 2011.

Analyzing precision using real data

Let's take a closer look at the spark line graphics in the last column of each table. As we mentioned before, this test required the fitting of real data from a real source. Therefore, we have to bear in mind that we are introducing potential fitting error.

Contrary to their peers, Risk Solver and ModelRisk both exhibit erratic patterns in the real data test and do not gain added precision by adding trials. Overall, even with the strange behavior, Risk Solver has less standard deviation in the tail percentiles than ModelRisk.

Test	RealData
Correlation	On
Package	@RISK

Values	Column Labels 10000	50000	100000
StdDev of 1	$379.67	$161.24	$134.54
StdDev of 5	$217.47	$55.05	$50.87
StdDev of 10	$111.13	$49.61	$32.61
StdDev of 50	$68.32	$30.14	$19.10
StdDev of 90	$99.87	$38.39	$20.90
StdDev of 95	$97.81	$62.26	$24.06
StdDev of 99	$248.55	$151.29	$71.93
StdDev of 999	$873.82	$358.83	$224.08
StdDev of 9999	$3,008.61	$1,231.85	$748.63

Test	RealData
Correlation	On
Package	Crystal Ball

Values	Column Labels 10000	50000	100000
StdDev of 1	$397.17	$167.67	$94.47
StdDev of 5	$213.15	$149.47	$70.89
StdDev of 10	$194.72	$85.86	$53.80
StdDev of 50	$133.65	$51.69	$42.90
StdDev of 90	$156.94	$66.84	$49.07
StdDev of 95	$232.21	$61.97	$66.38
StdDev of 99	$471.33	$202.27	$73.44
StdDev of 999	$770.19	$629.70	$309.66
StdDev of 9999	$2,765.35	$1,193.69	$1,038.13

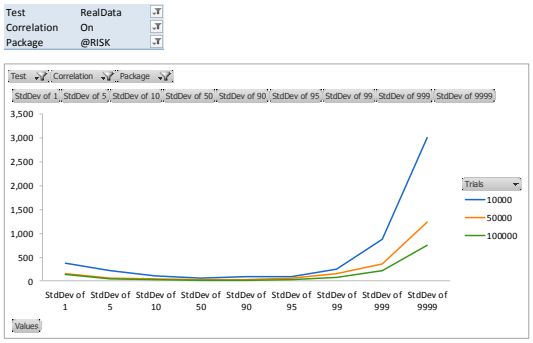
Test	RealData
Correlation	On
Package	ModelRisk

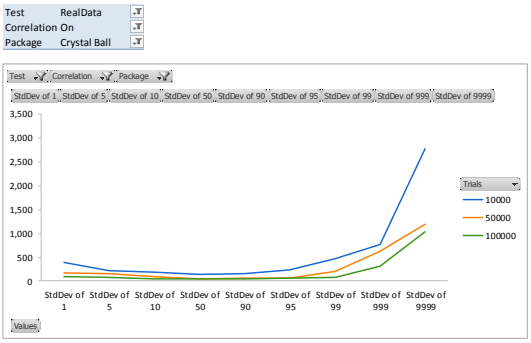
Values	Column Labels 10000	50000	100000
StdDev of 1	$506.88	$582.54	$521.36
StdDev of 5	$288.13	$333.80	$296.02
StdDev of 10	$176.53	$246.22	$206.40
StdDev of 50	$105.34	$103.66	$168.47
StdDev of 90	$181.57	$176.24	$137.47
StdDev of 95	$198.13	$236.94	$214.67
StdDev of 99	$355.43	$459.30	$432.14
StdDev of 999	$918.07	$1,090.24	$878.94
StdDev of 9999	$2,855.84	$1,981.54	$2,252.37

Test	RealData
Correlation	On
Package	RiskSolver

Values	Column Labels 10000	50000	100000
StdDev of 1	$484.59	$397.34	$413.46
StdDev of 5	$255.51	$198.35	$231.52
StdDev of 10	$190.76	$192.43	$201.67
StdDev of 50	$72.60	$87.74	$97.43
StdDev of 90	$106.07	$114.94	$88.48
StdDev of 95	$131.73	$149.48	$126.67
StdDev of 99	$261.02	$225.69	$236.05
StdDev of 999	$839.83	$676.46	$619.80
StdDev of 9999	$1,813.84	$2,352.78	$1,779.84

Table 5: Standard deviation/package/number of trials (correlation on) - real data test





©Technology Partnerz Ltd, 2011.

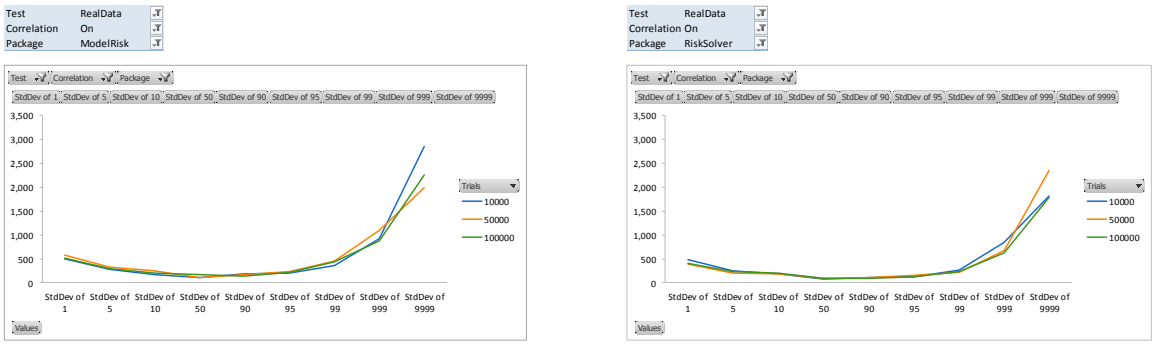


Figure 4: Standard deviation graphs by percentile/package - real data test

Alternatively when perusing the uncorrelated results we can see that the erratic pattern goes away with Risk Solver but not with ModelRisk. One could interpret the erratic behavior in Risk Solver is being contributed by their implementation of correlation.

Even though we could write off the precision error easily in a financial context, this may not be the case for other applications such as science and engineering where lives and safety might be at risk. For this reason, more research will need to be conducted with the vendors to more fully understand what factors could contribute to the precision results presented in the table below and how to mitigate against them.

Test	RealData
CorrelatiOff	Off
Package	@RISK

	Column Labels		
Values	10000	50000	100000
StdDev of 1	$336.59	$114.08	$104.19
StdDev of 5	$149.72	$57.24	$49.85
StdDev of 10	$106.72	$41.85	$31.60
StdDev of 50	$57.44	$21.93	$19.80
StdDev of 90	$99.27	$34.35	$21.45
StdDev of 95	$152.66	$38.72	$30.90
StdDev of 99	$237.79	$136.96	$63.28
StdDev of 999	$568.54	$398.01	$247.10
StdDev of 9999	$1,743.68	$1,064.22	$747.07

Test	RealData
Correlation	Off
Package	Crystal Ball

	Column Labels		
Values	10000	50000	100000
StdDev of 1	$346.24	$191.14	$91.84
StdDev of 5	$213.30	$98.18	$54.51
StdDev of 10	$134.64	$76.19	$39.65
StdDev of 50	$89.98	$42.27	$18.68
StdDev of 90	$114.96	$48.18	$29.56
StdDev of 95	$130.68	$61.33	$37.98
StdDev of 99	$300.80	$98.24	$65.48
StdDev of 999	$906.03	$384.51	$269.04
StdDev of 9999	$1,629.70	$1,198.69	$783.91

Test	RealData
Correlation	Off
Package	ModelRisk

	Column Labels		
Values	10000	50000	100000
StdDev of 1	$359.35	$425.58	$415.54
StdDev of 5	$157.03	$203.93	$192.80
StdDev of 10	$143.91	$142.77	$147.08
StdDev of 50	$88.30	$108.96	$103.46
StdDev of 90	$96.07	$127.92	$124.39
StdDev of 95	$129.16	$143.75	$155.33
StdDev of 99	$211.87	$252.69	$281.38
StdDev of 999	$985.11	$1,025.96	$694.75
StdDev of 9999	$2,256.58	$2,238.96	$1,970.54

Test	RealData
Correlation	Off
Package	RiskSolver

	Column Labels		
Values	10000	50000	100000
StdDev of 1	$519.20	$155.24	$137.82
StdDev of 5	$231.85	$64.34	$46.72
StdDev of 10	$118.98	$65.32	$38.40
StdDev of 50	$103.73	$37.64	$31.36
StdDev of 90	$131.72	$44.11	$51.61
StdDev of 95	$152.09	$56.77	$41.36
StdDev of 99	$216.20	$93.86	$53.18
StdDev of 999	$729.07	$291.69	$234.19
StdDev of 9999	$2,073.40	$905.45	$829.69

Table 6: Standard deviation/package/number of trials (correlation off) - real data test

©Technology Partnerz Ltd, 2011.

Given the sheer difference between the standard deviation generated at certain percentiles by ModelRisk versus its peers, we felt it would be constructive to compare it on a ratio basis. To be fair, we should have prepared similar tables for the other tests but they would've indicated very similar patterns anyways. For those who want to check, all the test results will be made available at the end of this post

ModelRisk also has very high standard deviation in almost all the percentiles as compared to its peers at both 50,000 and 100,000 trials. Theoretically, we would expect a more consistent reduction in standard deviation as we increase the number of trials yet this is not the case with ModelRisk. The standard deviation can also be looked at in terms of multiples against its peers - again this is something we would like to better understand because we are not sure if this is intentional (or not) nor if it is good or bad.

Real Data MR/FS Ratio No Correl

	10K	50K	100K	Std Dev
1	1.00	2.31	3.48	
5	0.72	2.22	3.95	
10	0.90	1.75	3.09	
50	0.85	2.54	3.35	
90	0.88	2.39	2.85	
95	0.88	2.50	3.22	
99	0.81	2.45	4.31	
99.9	1.21	3.34	3.34	
99.99	1.02	1.96	2.14	

Real Data MR/FS W/ Correl

10K	50K	100K	Std Dev
1.26	3.06	4.32	
1.21	2.88	4.09	
1.09	3.33	5.00	
1.19	2.27	5.36	
2.04	3.67	3.10	
1.70	3.89	3.96	
1.22	3.78	5.27	
1.09	3.48	2.95	
0.97	2.46	2.93	

Real Data MR/CB Ratio No Correl

	10K	50K	100K	Std Dev
1	1.04	2.23	4.52	
5	0.74	2.08	3.54	
10	1.07	1.87	3.71	
50	0.98	2.58	5.54	
90	0.84	2.66	4.21	
95	0.99	2.34	4.09	
99	0.70	2.57	4.30	
99.9	1.09	2.67	2.58	
99.99	1.38	1.87	2.51	

Real Data MR/CB W/ Correl

10K	50K	100K	Std Dev
1.28	3.47	5.52	
1.35	2.23	4.18	
0.91	2.87	3.84	
0.79	2.01	3.93	
1.16	2.64	2.80	
0.85	3.82	3.23	
0.75	2.27	5.88	
1.19	1.73	2.84	
1.03	1.66	2.17	

Real Data MR/AR Ratio No Correl

	10K	50K	100K	Std Dev
1	1.07	3.73	3.99	
5	1.05	3.56	3.87	
10	1.35	3.41	4.65	
50	1.54	4.97	5.23	
90	0.97	3.72	5.80	
95	0.85	3.71	5.03	
99	0.89	1.84	4.45	
99.9	1.73	2.58	2.81	
99.99	1.29	2.10	2.64	

Real Data MR/AR W/ Correl

10K	50K	100K	Std Dev
1.34	3.61	3.88	
1.32	6.06	5.82	
1.59	4.96	6.33	
1.54	3.44	8.82	
1.82	4.59	6.58	
2.03	3.81	8.92	
1.43	3.04	6.01	
1.05	3.04	3.92	
0.95	1.61	3.01	

Real Data MR/Group Ratio No Correl

	10K	50K	100K	Std Dev
1	1.04	2.61	3.95	
5	0.81	2.48	3.78	
10	1.08	2.15	3.71	
50	1.05	3.05	4.48	
90	0.89	2.82	3.94	
95	0.90	2.74	3.98	
99	0.80	2.24	4.35	
99.9	1.29	2.82	2.88	
99.99	1.21	1.97	2.41	

Real Data MR/Group W/ Correl

10K	50K	100K	Std Dev
1.29	3.37	4.47	
1.29	3.13	4.57	
1.13	3.53	4.85	
1.09	2.44	5.41	
1.58	3.45	3.61	
1.33	3.84	4.45	
1.06	2.90	5.70	
1.11	2.51	3.17	
0.98	1.84	2.64	

Figure 5: ModelRisk versus its peers.

©Technology Partnerz Ltd, 2011.

Nonetheless, ModelRisk exhibits counterintuitive behavior when compared to its peers at numbers of trials exceeding 10,000. Though some may argue that the standard deviation is negligible for the amounts we are working with, it appears that in both correlated and uncorrelated versions of this test, Oracle Crystal Ball and Palisade @RISK both seem to do just a little better when it comes to reducing the standard deviation around percentiles deep in the tails at higher numbers of trials.

In the next test, we account for the potential fitting error by removing it altogether. The reason for this stems from each application's different implementation of distribution fitting.

©Technology Partnerz Ltd, 2011.

Test 3: No Fitting, Just Normal Distributions

Of course, no matter how you cut it, if you are fitting data then you will introduce fitting error because of the different methods each packages employs. In our tests, when we fitted the data we noticed that some packages ranked different distributions as being the best fit than others. It became logical to apply normal distributions to all the asset classes just as we would when using the calculated Markowitz method. Essentially, we calculated both the mean and the standard deviation for each data set and used those parameters to configure their [asset class] respective normal distribution. By doing this all the packages are now on equal footing.

Comparing Accuracy (Mean Results) by Percentile/Package: Parameterized normal distributions

Table 7: Mean results by percentile for 20 simulations (by package) using Normal Distributions using parameters

Test	Normal
Trials	10000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$228,349.03	$235,071.56	$238,616.77	$251,055.94	$263,558.14	$267,127.10	$273,668.61	$281,004.85	$286,111.48
Crystal Ball	$228,099.46	$234,816.28	$238,430.68	$251,059.42	$263,671.30	$267,249.90	$274,063.98	$281,426.26	$286,400.70
ModelRisk	$227,434.42	$234,406.26	$238,132.02	$251,055.02	$263,971.73	$267,636.09	$274,563.81	$281,880.51	$286,598.49
RiskSolver	$225,125.52	$232,755.39	$236,781.04	$251,106.24	$265,372.56	$269,432.64	$276,986.43	$285,316.12	$291,248.84
Group Average	**$227,252.11**	**$234,262.38**	**$237,990.13**	**$251,069.15**	**$264,143.43**	**$267,861.43**	**$274,820.71**	**$282,406.94**	**$287,589.88**
Benchmark	This is the idea th	$234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$ 268,875.20	$ 276,029.57	$ 284,048.87	$ 290,649.88

Test	Normal
Trials	50000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$228,372.01	$234,990.18	$238,556.38	$251,051.46	$263,567.80	$267,119.10	$273,784.66	$281,295.33	$287,071.60
Crystal Ball	$228,182.49	$234,873.72	$238,439.37	$251,055.55	$263,645.63	$267,247.09	$274,005.25	$281,561.37	$287,618.23
ModelRisk	$227,481.42	$234,451.56	$238,123.41	$251,047.42	$263,937.73	$267,586.55	$274,444.25	$281,948.01	$286,449.30
RiskSolver	$225,079.16	$232,701.60	$236,766.49	$251,070.90	$265,373.91	$269,417.51	$277,014.28	$285,538.10	$292,347.61
Group Average	**$227,278.77**	**$234,254.27**	**$237,971.41**	**$251,056.33**	**$264,131.27**	**$267,842.56**	**$274,812.11**	**$282,585.70**	**$288,371.68**
Benchmark	$ 227,185.30	$234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$ 268,875.20	$ 276,029.57	$ 284,048.87	$ 290,649.88

Test	Normal
Trials	100000
Correlation	On

Row Labels	Average of 1	Average of 5	Average of 10	Average of 50	Average of 90	Average of 95	Average of 99	Average of 999	Average of 9999
@RISK	$228,326.84	$234,995.77	$238,551.19	$251,042.89	$263,583.25	$267,129.21	$273,812.99	$281,064.14	$287,204.55
Crystal Ball	$228,128.16	$234,847.12	$238,423.70	$251,070.89	$263,678.11	$267,272.92	$273,972.29	$281,566.83	$287,853.16
ModelRisk	$227,455.54	$234,388.96	$238,089.64	$251,003.61	$264,037.24	$267,660.67	$274,668.16	$282,439.45	$288,849.99
RiskSolver	$225,088.30	$232,696.10	$236,755.81	$251,061.14	$265,382.72	$269,435.89	$277,057.18	$285,476.22	$292,292.80
Group Average	**$227,249.71**	**$234,231.99**	**$237,955.09**	**$251,044.63**	**$264,170.33**	**$267,874.67**	**$274,877.65**	**$282,636.66**	**$289,050.12**
Benchmark	$ 227,185.30	$234,339.66	$ 238,153.63	$ 251,607.43	$ 265,061.23	$ 268,875.20	$ 276,029.57	$ 284,048.87	$ 290,649.88

When using normal distributions across the board instead of relying on each individual package's fit selections, we end up with far more consistent results. However, it appears that Risk Solver is much

©Technology Partnerz Ltd, 2011.

closer to the values calculated using Excel 2010' s new statistical functions. It is my understanding that Frontline Systems (the makers of Risk Solver) had a hand in designing these new functions and this could potentially account for the close proximity between Risk Solver's simulated results and Excel 2010's calculated results.

Of note is the accuracy of the three other packages (Crystal Ball, @RISK and ModelRisk) is relatively consistent across the group. The only way to know for sure who is right would be to model the very same problem using a tool such as MatLab in order to establish a golden standard. Nevertheless, it is also my understanding that the statistical functions have truly improved in the latest version of Excel and are now considered reliableprecision using real many experts.

Analyzing precision using parameterized normal distributions

Once again, when compared to its peers, ModelRisk has the highest amount of standard deviation in the tails. We are not certain if this is by design or if this is a bug in the math or even the simulation tool's number generation method. Vose software has often stated that most tools and analysts tend to underestimate the real amount of risk related to a specific problem. In fact, they have added a built-in feature to account for uncertainty because most underestimate risk. These are questions definitely worth asking because as a risk analyst, it is very important to understand the implications of the math and the tools you use - as they will invariably have an impact on the final outcome of a decision. *Please refer to the* ***Commentary by David Vose*** *to compare results with his simultaneous method.*

Test	Normal
Correlation	On
Package	@RISK

Values	Column Labels 10000	50000	100000
StdDev of 1	$328.70	$149.15	$139.08
StdDev of 5	$162.70	$66.38	$51.81
StdDev of 10	$173.41	$65.69	$37.70
StdDev of 50	$94.24	$44.80	$30.24
StdDev of 90	$150.00	$54.22	$43.33
StdDev of 95	$174.15	$78.99	$57.12
StdDev of 99	$287.84	$145.59	$108.11
StdDev of 999	$898.07	$328.97	$272.31
StdDev of 9999	$2,118.28	$755.22	$827.99

Test	Normal
Correlation	On
Package	Crystal Ba

Values	Column La 10000	50000	100000
StdDev of 1	$380.23	$142.89	$90.00
StdDev of 5	$213.94	$63.85	$68.94
StdDev of 10	$171.17	$58.05	$59.88
StdDev of 50	$132.64	$56.96	$38.20
StdDev of 90	$127.11	$86.53	$48.81
StdDev of 95	$184.94	$100.81	$55.64
StdDev of 99	$325.45	$229.19	$124.70
StdDev of 999	$928.29	$428.12	$332.96
StdDev of 9999	$1,820.66	$928.46	$730.76

Test	Normal
Correlation	On
Package	ModelRisk

Values	Column Labels 10000	50000	100000
StdDev of 1	$377.14	$537.51	$460.54
StdDev of 5	$200.35	$223.77	$187.42
StdDev of 10	$145.91	$142.05	$134.11
StdDev of 50	$152.33	$119.66	$97.22
StdDev of 90	$169.87	$194.73	$175.65
StdDev of 95	$201.46	$230.47	$207.69
StdDev of 99	$351.36	$374.97	$403.93
StdDev of 999	$992.86	$848.05	$1,180.82
StdDev of 9999	$2,139.46	$2,113.98	$2,699.45

Test	Normal
Correlation	On
Package	RiskSolver

Values	Column La 10000	50000	100000
StdDev of 1	$299.73	$96.99	$75.91
StdDev of 5	$122.76	$77.34	$25.03
StdDev of 10	$98.40	$50.44	$33.50
StdDev of 50	$92.16	$38.51	$27.54
StdDev of 90	$133.35	$42.05	$41.16
StdDev of 95	$142.89	$72.10	$43.67
StdDev of 99	$234.30	$131.66	$87.24
StdDev of 999	$568.24	$215.55	$174.40
StdDev of 9999	$1,237.21	$607.96	$515.36

Table 8: Standard deviation/package/number of trials (correlation on) - normal test

©Technology Partnerz Ltd, 2011.

Test	Normal
Correlation	Off
Package	@RISK

Values	Column Labels 10000	50000	100000
StdDev of 1	$266.74	$102.46	$60.83
StdDev of 5	$151.48	$60.16	$42.27
StdDev of 10	$156.94	$52.07	$37.09
StdDev of 50	$81.15	$37.49	$32.80
StdDev of 90	$116.29	$51.32	$31.21
StdDev of 95	$150.98	$55.29	$40.68
StdDev of 99	$233.14	$122.39	$96.01
StdDev of 999	$588.34	$349.87	$243.56
StdDev of 9999	$1,494.95	$626.79	$615.05

Test	Normal
Correlation	Off
Package	Crystal Ball

Values	Column Labels 10000	50000	100000
StdDev of 1	$282.36	$143.48	$80.89
StdDev of 5	$137.19	$89.09	$57.29
StdDev of 10	$123.22	$68.78	$30.09
StdDev of 50	$101.09	$44.70	$25.98
StdDev of 90	$123.09	$56.50	$49.17
StdDev of 95	$143.32	$72.08	$58.90
StdDev of 99	$268.46	$101.08	$77.27
StdDev of 999	$533.62	$312.73	$219.54
StdDev of 9999	$1,888.71	$656.87	$509.16

Test	Normal
Correlation	Off
Package	ModelRisk

Values	Column Labels 10000	50000	100000
StdDev of 1	$819.32	$1,117.20	$894.97
StdDev of 5	$457.70	$487.66	$612.37
StdDev of 10	$436.70	$437.26	$470.54
StdDev of 50	$288.17	$409.57	$280.17
StdDev of 90	$276.38	$327.00	$397.01
StdDev of 95	$433.61	$466.57	$472.55
StdDev of 99	$760.93	$1,116.15	$807.27
StdDev of 999	$1,340.61	$2,242.48	$1,656.17
StdDev of 9999	$2,008.89	$3,054.79	$2,248.54

Test	Normal
Correlation	Off
Package	RiskSolver

Values	Column Labels 10000	50000	100000
StdDev of 1	$324.50	$136.47	$72.31
StdDev of 5	$139.92	$92.11	$52.69
StdDev of 10	$114.02	$70.69	$44.66
StdDev of 50	$73.56	$49.67	$31.50
StdDev of 90	$98.14	$47.55	$42.95
StdDev of 95	$163.37	$41.92	$50.82
StdDev of 99	$235.58	$117.68	$94.27
StdDev of 999	$734.71	$233.60	$215.28
StdDev of 9999	$1,394.94	$954.39	$597.75

Table 9: Standard deviation/package/number of trials (correlation off) -normal test

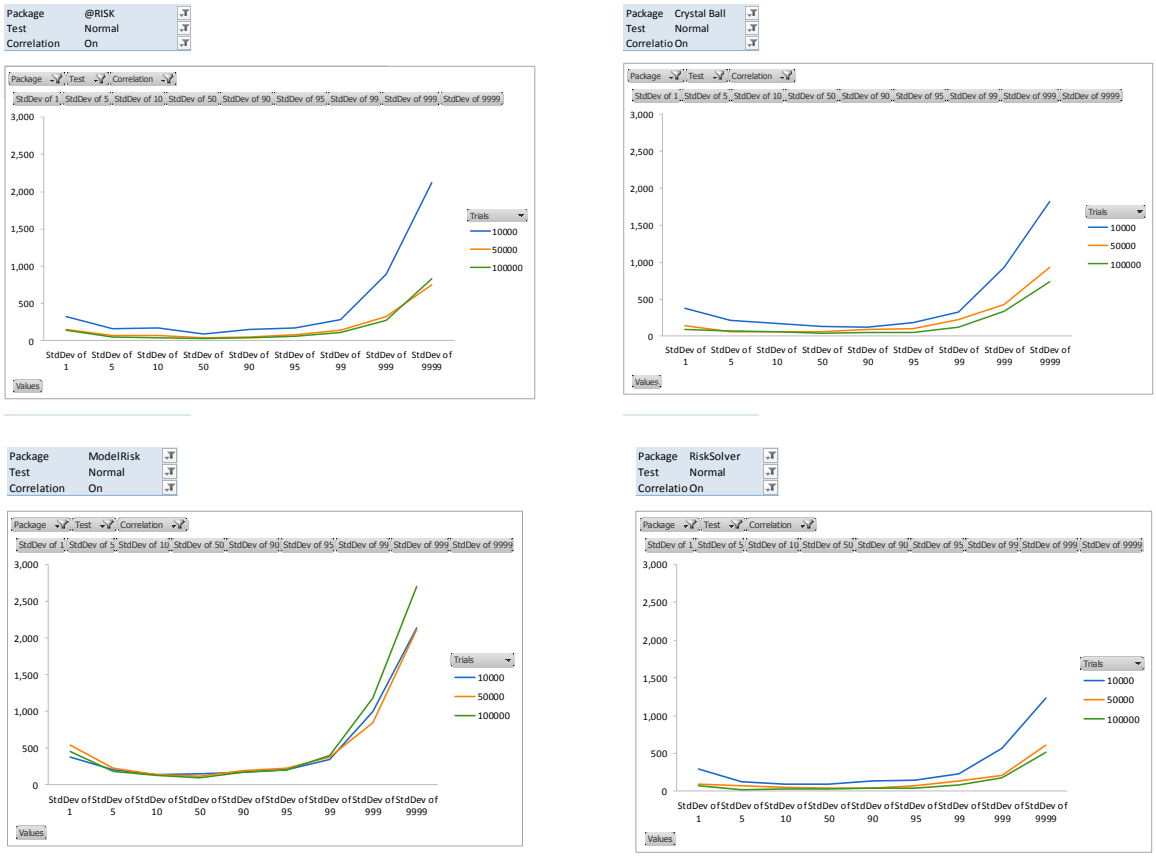


Figure 6: standard deviation by percentile/package (correlation on) - normal test

©Technology Partnerz Ltd, 2011.

Who's the fastest- analyzing the speed tests

In order to properly conduct the testing of our favorite applications, we started with a very simple script. Our script took the correlated returns model that we used in our previous article and looped it 20 times at various numbers of trials i.e. 10,000, 50,000, and 100,000.

For each test run, an application ran the equivalent of 3.2 million trials. We ran each test twice to account for the correlation being on and off therefore generating 6.4 million trials per package. All in all we ran 20 individual tests totaling 64 million trials.

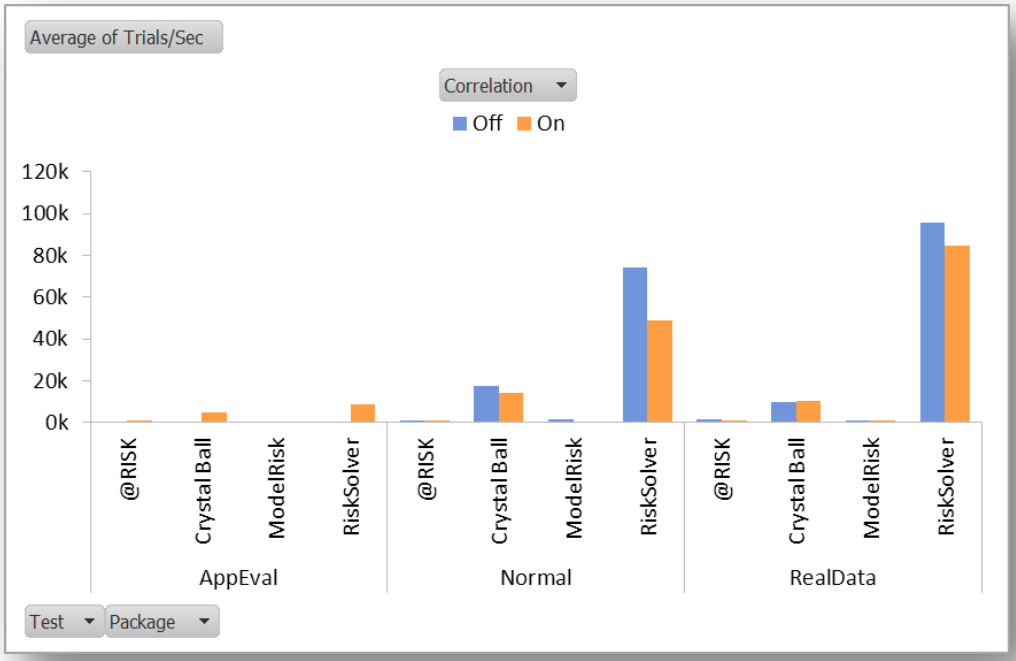


Figure 7: trials/sec by package with and without correlation

©Technology Partnerz Ltd, 2011.

Speed and correlation

Average of Trials/Sec			
Row Labels	Off	On	Group Average
AppEval			
@RISK		799.61	799.61
Crystal Ball		4,716.50	4,716.50
ModelRisk		429.37	429.37
RiskSolver		8,679.59	8,679.59
Normal			
@RISK	986.13	678.51	832.32
Crystal Ball	17,319.44	14,363.10	15,841.27
ModelRisk	1,375.14	452.60	913.87
RiskSolver	74,151.23	48,682.10	61,416.67
RealData			
@RISK	1,318.24	966.12	1,142.18
Crystal Ball	9,580.09	10,055.56	9,817.82
ModelRisk	788.12	727.52	757.82
RiskSolver	95,450.62	84,389.99	89,920.31
Group Average	**25,121.13**	**14,578.38**	**18,795.48**

Correlation and copulas tend to be very computationally intensive, regardless of the tool or package. ModelRisk uses another set of tools called objects that are equally computationally intensive and can slow down a model considerably. It is for these reasons that we believe there are such performance discrepancies between ModelRisk and its peers regarding speed. Both initial test and the real data test required some sort of fitting on the part of the applications. ModelRisk's fitting objects are very powerful and useful tools for the advanced modeler but carry with them a heavy performance price. When you combine copulas, fitting objects and data objects you can end up with a much slower model than when you started, therefore it is very important to weigh the relative importance of these features when designing your analytical solution.

©Technology Partnerz Ltd, 2011.

Speed and the Number of Trials

Correlation	On

Average of Trials/Sec				
Row Labels	**10000**	**50000**	**100000**	**Group Average**
AppEval				
@RISK	647.02	858.41	893.40	799.61
Crystal Ball	3,125.00	5,305.56	5,718.95	4,716.50
ModelRisk	413.95	434.79	439.37	429.37
RiskSolver	1,861.81	8,119.33	16,057.63	8,679.59
Normal				
@RISK	562.98	697.16	775.39	678.51
Crystal Ball	6,250.00	16,041.67	20,797.62	14,363.10
ModelRisk	430.82	467.03	459.96	452.60
RiskSolver	15,259.26	53,287.04	77,500.00	48,682.10
RealData				
@RISK	788.37	1,038.89	1,071.10	966.12
Crystal Ball	3,416.67	11,750.00	15,000.00	10,055.56
ModelRisk	683.13	746.50	752.93	727.52
RiskSolver	13,250.00	75,083.33	164,836.65	84,389.99
Group Average	**3,890.75**	**14,485.81**	**25,358.58**	**14,578.38**

One of the interesting findings of our speed tests is that as you increase the number of trials you should find an increase in average speed (trials per second). Once again, Risk Solver on a simulation per simulation basis ranks fastest and is seconded by Crystal Ball. The reason why these two packages are so much faster than their peers is their implementation of PSI (polymorphic spreadsheet interpretation).

PSI technology compiles the spreadsheet and runs it in the background as code resulting in significant performance improvements. One of the major drawbacks of this technology is that it will not run models using special lookup functions such as INDEX or MATCH as well as certain high-end recursive functions. When this happens, you must gear down to an Excel-based method such as employed by @RISK and ModelRisk. Under these conditions both Risk Solver and Crystal Ball will run at speeds comparable to their peers.

©Technology Partnerz Ltd, 2011.

Who is the fastest overall – Total Test Duration

AppEval	Start	Finish	Total Duration:	Total Trials Run:	Avg. Trials
@RISK	8/6/2011 18:01	8/6/2011 19:03	1:02:18	3,200,000	856.07
Crystal Ball	8/6/2011 20:10	8/6/2011 20:20	0:10:21	3,200,000	5,152.98
ModelRisk	8/6/2011 14:11	8/6/2011 16:14	2:02:29	3,200,000	435.43
RiskSolver	8/11/2011 16:24	8/11/2011 16:37	0:12:17	3,200,000	4,341.93
Normal - Correlation					
@RISK	8/13/2011 23:05	8/14/2011 0:18	1:13:19	3,200,000	727.44
Crystal Ball	8/13/2011 11:27	8/13/2011 11:31	0:03:36	3,200,000	14,814.81
ModelRisk	8/12/2011 17:49	8/12/2011 19:47	1:58:41	3,200,000	449.38
RiskSolver	8/23/2011 20:56	8/23/2011 20:57	0:01:31	3,200,000	35,164.84
Normal - No Correlation					
@RISK	8/14/2011 10:47	8/14/2011 11:38	0:51:08	3,200,000	1,043.02
Crystal Ball					
ModelRisk	8/13/2011 11:16	8/13/2011 11:54	0:37:54	3,200,000	1,407.21
RiskSolver	8/23/2011 21:02	8/23/2011 21:04	0:01:10	3,200,000	45,714.29
RealData - Correlation					
@RISK					
Crystal Ball	8/12/2011 12:47	8/12/2011 12:53	0:05:25	3,200,000	9,846.15
ModelRisk	8/11/2011 19:49	8/11/2011 21:00	1:11:40	3,200,000	744.19
RiskSolver					
RealData - No Correlation					
@RISK					
Crystal Ball	8/12/2011 12:32	8/12/2011 12:38	0:05:51	3,200,000	9,116.81
ModelRisk	8/11/2011 18:14	8/11/2011 19:20	1:06:24	3,200,000	803.21
RiskSolver					

As a note to the reader, due to some bugs in the test script we were not always able to capture the start and finish times of all the tests therefore our table is somewhat incomplete but indicative. If you average the number of trials executed over the actual duration of the 60 simulations the results are more accurate. This is especially true for the applications that use PSI technology.

As we can see in the various performance tests Risk Solver is bar none the fastest simulation tool on a simulation per simulation basis on the market. But this is not to say that Risk Solver took the less time to run through the entire test.

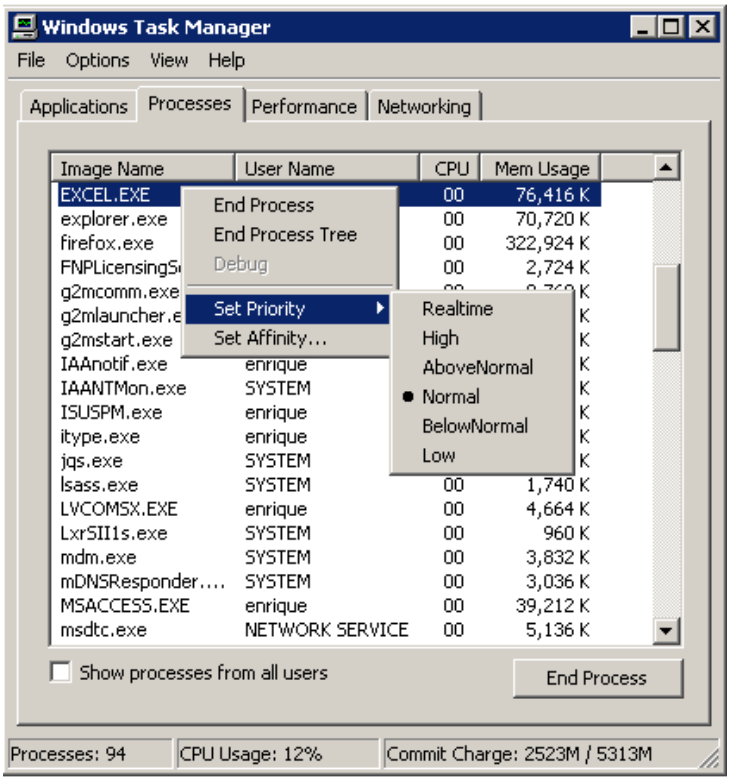
The other aspect of speed is the efficiency of the VBA toolkit when running the routines in the background. In this regard, Oracle Crystal Ball had the most robust and efficient handling of the VBA. When running a script, many things can happen in the background which can impact the total duration of the test script. For example, when looking at the total duration of the tests, Oracle Crystal Ball has finished in less time and with fewer trials per second than Risk Solver on two out of three tests. Speed

©Technology Partnerz Ltd, 2011.

notwithstanding, all of the other packages had some tedious issue - some serious, some not. For example:

- @RISK had a conflict with Risk Solver resulting in a dialog box popping up after each simulation. The only way to deal with it was to shut down Risk Solver's Excel add-in and COM add-in and restart Excel. I am sure that this is not an issue most people would run in however these types of conflicts can happen with other Excel-based add-ins.
- ModelRisk has a default feature (which can easily be shut off but not easy to find) that prompts the user to save his results at the end of each simulation. When looping through 60 simulations, you don't want to have to deal with this. I spent some time trying to figure out how to shut this off using VBA only to realize that this was part of the default run parameters set.
- Risk solver would hang on certain simulations for up to 45 seconds and execute others in two seconds. After some analysis I realized that I had some registry issues and I believe a potential application conflict that I inadvertently resolved. These actions seem to have corrected the issue and enabled Risk Solver to perform consistently.

Tips to improve simulation speeds



When researching and designing our test script we stumbled across a few techniques that can add anywhere between 15 to 30% in performance (increased trials per second) when combined.

- Freeze screen updating in VBA using Application.ScreenUpdating = False
- Use the xlMinimized / xlMaximized functions in VBA (Application.WindowState = xlMinimized). These functions, like freezing screen updating, reduce the workload in Excel by eliminating the need for refreshes.
- Close unnecessary programs
- Set the Excel process to real-time and make sure multiprocessor support is active both in Windows and Excel

©Technology Partnerz Ltd, 2011.

Conclusion

So far, we have tested our four favorite Monte Carlo packages along three dimensions: speed, accuracy and precision. During this process, we discovered some very interesting idiosyncrasies among the packages that are worth further examination on two levels: by the vendors who make the tools and by the analysts who use them.

Accuracy and precision

When testing for accuracy and precision, our goal was not to get the right number but to understand how each package to behaved under specific run conditions and if it was consistent with what should normally be expected. We urge the reader to analyze the data and spark line bar charts of each package in each test with the correlation both on and off.

Speed

As for speed, it is very important to look at both elements: simulation by simulation and test cycle time. If we looked at each one of these tests separately, we might come up with different conclusions than when we look at the answers together.

PSI versus Excel-based simulation

It is very important to note that we compared simulation within Excel (@Risk and ModelRisk) versus Polymorphic Spreadsheet Interpretation (Crystal Ball and Risk Solver) that are two entirely different approaches to running simulation. If we had compared all the tools using Excel-based simulation, it is our understanding that we would have obtained comparable performance results. In sum, there are advantages and inconveniences to both methods :

- PSI is extremely fast but has issues with certain types of functions (recursive and specialized lookups) that will prevent it to run
- Excel simulation is slow but will run almost anything, including functions from any package.

Nevertheless, technologies exist to close this performance gap. Providers such as Spreadsheet Gear or Calc For Web (formerly Turbo Excel) offer solutions that could add new levels of performance and flexibility for @Risk and ModelRisk. If you are developing a custom solution using Monte-Carlo simulation to run as a standalone application or Web service, than these solutions are even more interesting.

64-bit

Though we did not test it, another major contributor to performance is 64-bit capability. A 64-bit machine will enable you to install much more physical memory than 4 GB as well as handle it better. When working with correlation or other computationally intensive applications within Excel, the ability to address memory as well as having a bigger pipe to push data will invariably improve performance and enhance overall capability.

©Technology Partnerz Ltd, 2011.

Oracle Crystal Ball, Palisade @Risk and Frontline Risk Solver all support Excel 64-bit and it's enhanced calculation capabilities. Vose software has indicated that a 64-bit version is being considered for a future release, but no dates or commitments were offered. If you are building a large-scale solution requiring large amounts of data and memory, then you cannot discount the necessity of moving towards a 64-bit solution.

Downloadable Test results

Generally, it is wiser to let the data speak for itself. For those who would like to take a closer look at the test results, please make sure to download the consolidated test results file (Excel format) from our website (www.technologypartnerz.com). It contains all the simulation results as well as graphs and pivot tables that will enable you to slice and dice the results.

Acknowledgments

I would like to thank all the vendors for their continued support in this series of articles. I would like to thank David Vose for sharing his extremely novel and efficient implementation of the Markowitz model and Excel. It is with this technique, that we established the benchmark for all of the other packages. Without his intervention, it would have been very difficult to do this analysis.

I would also like to thank my friend and colleague Dr. William McKibbin for his time and advice on how to design the tests and interpret the results in the fairest possible way.

Comments and feedback

If you have any questions regarding this white paper, please do not hesitate to drop me a line at etorkia@technologypartnerz.com.

©Technology Partnerz Ltd, 2011.

Commentary on the test results by David Vose:

If the new test results aren't like the ones I sent you, run my model and increase the number of lines. *[This model is available for download off the Technology Partnerz website]* You are dealing with highly variable measures, so it is very difficult to arrive at solid conclusions without a lot of testing. You are just simulating a multivariate normal – which, by the way, ModelRisk has available immediately and would have run many times faster as a result.

Vose Simultaneous Simulation Test

This test was developped by David Vose to demonstatrate that ModelRisk will produce precise numbers using a model that generates all 20 simlations simultaneously

Values	10000	50000	100000
StdDev of 1	$398.42	$206.95	$115.74
StdDev of 5	$258.56	$124.74	$60.47
StdDev of 10	$159.27	$82.83	$62.28
StdDev of 50	$144.69	$44.93	$43.63
StdDev of 90	$166.07	$63.36	$40.24
StdDev of 95	$123.65	$80.13	$72.29
StdDev of 99	$294.73	$120.74	$126.39
StdDev of 999	$1,044.53	$374.96	$277.08
StdDev of 9999	$1,913.77	$723.41	$795.78

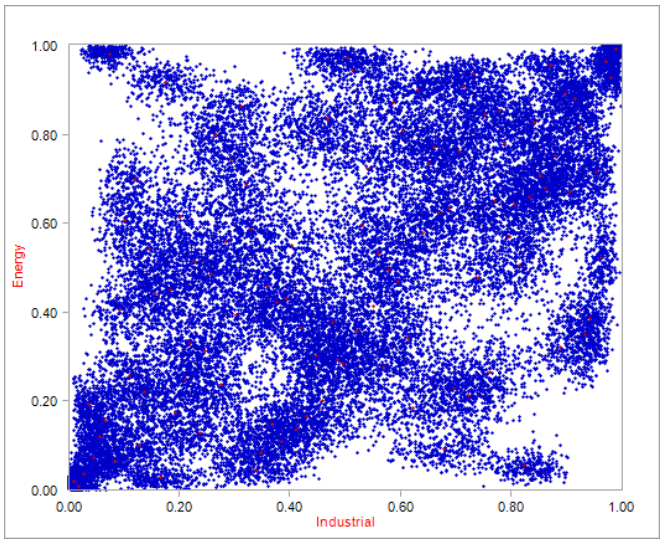
Figure 8: New Results for the Norma Test

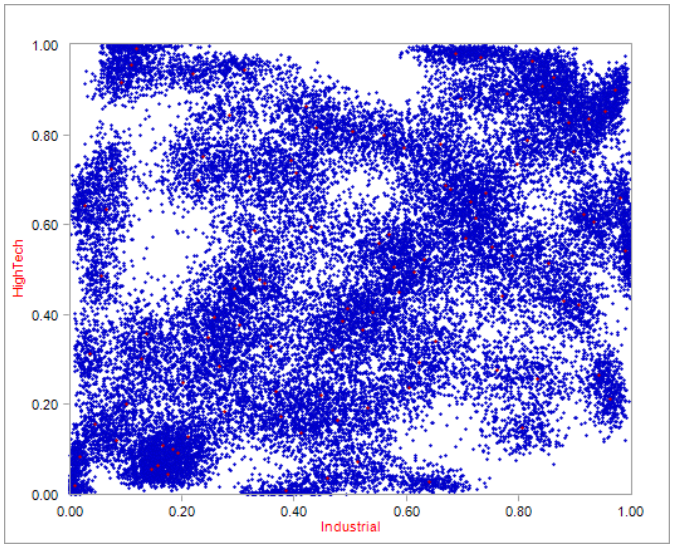
Test	Normal		
Correlation	On		
Package	ModelRisk		
Values	**10000**	**50000**	**100000**
StdDev of 1	$377.14	$537.51	$460.54
StdDev of 5	$200.35	$223.77	$187.42
StdDev of 10	$145.91	$142.05	$134.11
StdDev of 50	$152.33	$119.66	$97.22
StdDev of 90	$169.87	$194.73	$175.65
StdDev of 95	$201.46	$230.47	$207.69
StdDev of 99	$351.36	$374.97	$403.93
StdDev of 999	$992.86	$848.05	$1,180.82
StdDev of 9999	$2,139.46	$2,113.98	$2,699.45

Figure 9: Original Test Results for ModelRisk using the sequential test script

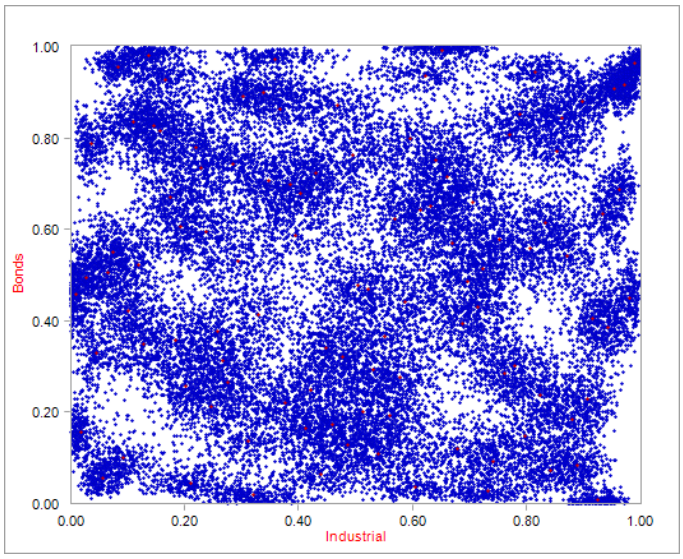
©Technology Partnerz Ltd, 2011.

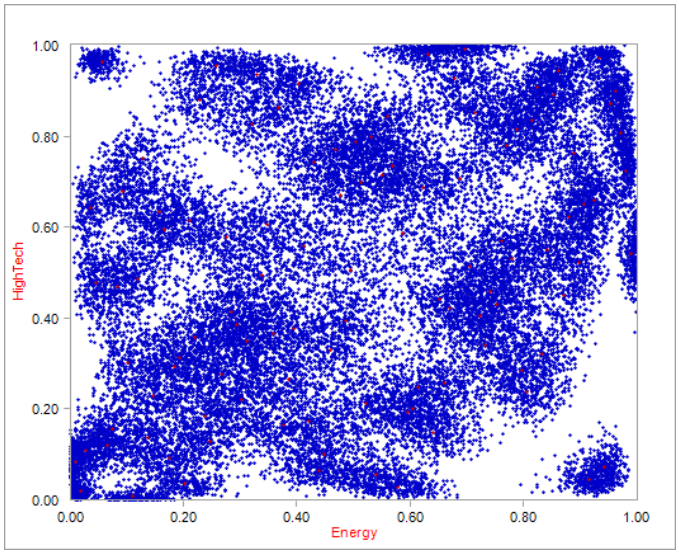
My point of using real data [versus simulated data] is that you would also try the empirical copula – a unique ModelRisk feature that is extremely powerful. It's slow, which seems to be a primary focus for you, but it doing a far better job (see charts below for the implied correlation structure for the real data example). For most risk analysts, it really doesn't matter if a model takes 1 minute or 10 – what matters is that they get the best possible answers. That means having a choice of distributions and a choice of correlation structures.





©Technology Partnerz Ltd, 2011.





It is extremely unfair that you compare how long the products take to simulate when they are absolutely not doing the same thing: for ModelRisk you could swap out the Normal copula for a T, Gumbel, Frank, etc – and for the others there is NO other choice because they use this one fast but inflexible method that statistically cannot be correctly fit to data! […] and ask yourself why financial modelers NEVER use rank order correlation, always use copulas.

[…] Finally, if you really want to test accuracy, try using the U parameter and comparing results against Wolfram Alpha:

VoseNormalProb(-6,0,1,1) = 9.86587645037694E-10
Wolfram Alpha = 9.866E-10

VoseNormalProb(-10,0,1,1) = 7.61985302416047E-24
Wolfram Alpha = 7.62E-24

©Technology Partnerz Ltd, 2011.

VoseNormalProb(-20,0,1,1) = 2.75362411860615E-89
Wolfram Alpha = 2.754E-89

Or, better still, try comparing what each software product returns as the mean (by calculation, not simulation, eg RiskTheoMean, VoseMean) for say a Pareto distribution truncated way into its right tail. Then you will see a really big difference.

We compare all the functions we can against Wolfram Mathematica (Alpha is a cut-down online version of some of Mathematica's features). We take a lot of pride in trying to get things accurate….

- **David Vose**

Follow-Up Notes by Eric Torkia

We are not quite sure why our testing methodology did not work as intended with ModelRisk. When we ran the tests sequentially, the results we obtain are strange and erratic compared to its peers. Whereas, when we run a variation of the model where all the simulations happen at once - we obtain 2 benefits:

- Firstly, by structuring the model in this way, we can obtain an almost 50% improvement in the time it takes to generate the answers.
- Secondly, the results are bang on with its peers. This suggests that those who did or are considering buying ModelRisk should not worry.

In order to see how ModelRisk performs, we suggest you download a copy of ModelRisk from our website and test out the model David Vose kindly provided.

We are intent on working with Vose software to flush out this inconsistency and we will keep you posted as we progress.

©Technology Partnerz Ltd, 2011.