

CompAct

ISSUE 32 | JULY 2009



Cluster Modeling: A New Technique To Improve Model Efficiency

By Avi Freedman and Craig Reynolds

A new approach to actuarial modeling can solve a familiar problem for life insurance companies.

Life insurance companies around the world employ actuarial models to use in applications such as financial forecasting, product pricing, embedded value, risk management, and valuation. Depending on the company and the application, such models might be seriatim or might reflect some degree of compression.

Despite the enormous increase in processing power over the past decade or two, the ability of many companies to run models in a timely fashion has arguably worsened, primarily due to the large number of stochastic scenarios required to properly evaluate many types of insurance liabilities. For some purposes, nested stochastic scenarios need to be used, increasing runtime even more.

Typically, actuaries manage runtime in one of three ways:

- improving software efficiency;
- getting more or better hardware;
- reducing cell count via mapping techniques.

The first option can provide incremental payoffs but is unlikely to provide the order-of-magnitude performance improvements that the actuary might desire. The second option can help materially. Today it is not uncommon to see companies with grid farms of hundreds of computers. But the cost and maintenance efforts associated with such grid farms can be significant, and we have seen from experience that the complexity of the runs needed

- | | |
|--|--|
| <p>1 Cluster Modeling: A New Technique To Improve Model Efficiency
<i>Avi Freedman and Craig Reynolds</i></p> <p>2 Editor's Notes
<i>Howard Callif</i></p> <p>3 Letter from the Chair
<i>Tim Pauza</i></p> <p>9 Cool Tech
<i>Matthew Wilson</i></p> <p>14 R Corner¹—Model Formula Framework
<i>Steve Craighead</i></p> | <p>18 WANTED: Reviews and Articles on Life Insurance and Annuity Illustration, Needs Analysis, and Advanced Marketing Systems</p> <p>20 The End Users Justify the Means: IV The Journey Home
<i>Mary Pat Campbell</i></p> <p>25 In Praise of Approximations
<i>Carol Marler</i></p> |
|--|--|



CompAct

ISSUE 32 JULY 2009

Published quarterly by the Technology Section of the Society of Actuaries

2008-2009 SECTION LEADERSHIP

Tim A. Pauza
Chair
e: tim.pauza@ey.com

Jeffrey Pomerantz
Annual Meeting
e: jeff.pomerantz@qrm.com

Timothy Deitz
Vice-Chair/People
Coordinator
e: Tim.Deitz@omfn.com

David Minches
Communications
Coordinator
e: david.minches@ey.com

Frank Reynolds
Spring Meetings Education
Liaison
e: fgreynol@uwaterloo.ca

Carl Nauman
Scenario Manager/Webinar
Coordinator
e: Carl.Nauman@ggy.com

Holly L. Loberg
Web Coordinator/Actuarial
Club Program
e: holly_loberg@allianzlife.com

Carl Desrochers
"Other" Meetings
Coordinator /Speculative
Fiction Contest
e: carl_desrochers@berkshirelife.com

Phillip Gold
Board Partner
e: Phil.Gold@ggy.com

Joseph Liuzzo
Secretary/Treasurer
e: jliuzzo@tiaa-cref.org

OTHER VOLUNTEERS

Howard Callif
Newsletter Editor
e: howard@callif.org

Mary Pat Campbell
Social Networking Advisor
e: marypat.campbell@gmail.com

FRIEND OF THE COUNCIL

Kevin Pledge
Council Advisor

Robert LaLonde

SOA STAFF

Sam Phillips
Staff Editor
e: sphillips@soa.org

Susan Martz
Project Support Specialist
e: smartz@soa.org

Meg Weber
Staff Partner
e: mweber@soa.org

Julissa Sweeney
Graphic Designer
e: jsweeney@soa.org

Facts and opinions contained herein are the sole responsibility of the persons expressing them and shall not be attributed to the Society of Actuaries, its committees, the Technology Section or the employers of the authors. We will promptly correct errors brought to our attention.

Copyright © 2009 Society of Actuaries.
All rights reserved.
Printed in the United States of America

Editor's Notes

By Howard Callif

This edition has a wide variety of articles. Our feature for this month is on cluster modeling to save computing resources, and we would be very interested in hearing from people whether they'll consider this technique. We have a growing base of people submitting articles on a regular basis, and I want to thank all of them for their hard work! Please consider submitting an article too! I'd also like to call special attention to the second "Call for Illustration System Articles"—we are hoping to begin a series of articles on software used to sell and market insurance products.

The Editor's column in the last issue published a letter from Will Slade asking about information on source code control. I tried getting permission to reprint the article he suggested ("Where's the Real Bottleneck in Scientific Computing," *American Scientist*, by Gregory Wilson), but they do not allow this. I will try to find other resources or articles, because this is a relevant topic. In the meantime, they do allow posting of a link, so here it is: <http://www.americanscientist.org/issues/id.3473.y.0.no.,content.true.page.1.css.print/issue.aspx>

There is an additional letter I'd like to share:

From: Marler, Carol (GE, Corporate) [<mailto:Carol.Marler@ge.com>]

Subject: RE: April issue of the Technology section newsletter

To follow up, I am told that the Winchester House article is now a topic of conversation throughout our Corporate IT department. Thanks.

From: Marler, Carol (GE, Corporate)

Sent: Friday, May 01, 2009 11:05 AM

I love the letters to the editor column. And I have already shared the Winchester House article with some (non-actuarial) co-workers. Keep up the good work!

EDITOR'S REPLY: Thanks for your feedback, and your contributions! ■



Howard Callif, ASA, MAAA, is a product champion at Milliman IntelliScript. He can be contacted at howard@callif.org

Letter from the Chair

By Tim Pauza

Greetings! In this edition of my quarterly letter, I would like to highlight an exciting new opportunity for technology section members. With the help of our staff partners at the SOA, we have implemented a private LinkedIn group for section members. This online forum for networking and information sharing will focus on the challenges and interests of our members. As the technology demands grow for actuaries in an ever more complex and challenging economic environment, we believe LinkedIn can provide us an invaluable opportunity to keep abreast of the latest developments in arenas where technology is playing a major role in the future of our industry. Don't miss out on this opportunity to participate. As I write this letter, there are 73 members signed up for the group. The momentum is just starting to build, and I expect that number will have grown significantly by the time this edition of the newsletter is in your hands. If you have not already done so, join LinkedIn, join our group, and see what your fellow section members are up to. (www.linkedin.com)

In other important section news, section council elections are just around the corner, and we have an excellent slate of prospective council members who have volunteered to be on the ballot and take a leadership role for the sections. They will be introduced to you soon, as part of the SOA election process. Thanks to our volunteers for their willingness to get involved, and thanks to all of our current council members and friends of the council who work hard to keep our section going! The section council and friends, continues to make progress on several important initiatives. We are developing materials for upcoming meetings, soliciting software reviews for actuarial systems, researching webcast offerings in technologies of interest to our membership, and publishing this quarterly newsletter. The list goes on, and we are always ready to add to it, especially when we discover ways to serve you better. Have an idea for us? Let us know. Want to get involved? We can accommodate that too. Please reach out to any of the council members listed inside the front cover of this newsletter with your ideas or to volunteer.

Thanks and happy reading! ■

Tim Pauza

2009 Section Council Chair

Editor's Note: Just before the newsletter went to print, candidate profiles were posted on the LinkedIn site. Visit the site to review their statements.



Tim Pauza, ASA, is a manager at Ernst & Young, LLC. He can be contacted at tim.pauza@ey.com.

always seems to grow to match or exceed the capacity of the grid available.

Classic mapping techniques used to reduce cell counts and manage runtime can work reasonably well for some types of business, but they do not work as well when applied to products with many moving parts and heavy optionality, making it difficult to use them to create models even of moderate size, let alone of a smaller size to enable running against a large number of scenarios in a reasonable time.

Milliman has developed a technique called cluster modeling, a variant of cluster analysis techniques used in various scientific fields, to allow for an automated grouping of liabilities, assets or scenarios. This technique is described, with several liability examples, in our paper “Cluster analysis: A spatial approach to actuarial modeling.”¹ The paper includes a number of case studies that illustrate excellent fit from cluster techniques with cell compression ratios (actual policy count/model cell count) of 1,000–1 or more.

We believe that cluster modeling will be a valuable tool that will allow companies to reduce their runtime for stochastic models ...

In this article, we briefly describe the technique, and then offer some comments on implementation of the algorithm and on certain decisions that need to be made in considering how best to apply cluster modeling.

We believe that cluster modeling will be a valuable tool that will allow companies to reduce their runtime for stochastic models by orders of magnitude with minimal reduction in model accuracy. This process will generally be easier to implement than other processes for improving model efficiency and has material advantages over the use of replicating portfolios—another common technique to improve model efficiency.

DESCRIPTION OF THE TECHNIQUE

For purposes of exposition, we assume that we are using cluster modeling to compress a set of policies (perhaps 1,000,000) to a much smaller set of cells (perhaps 500).

The following are defined for each policy:

- An arbitrary number of *location variables*. A location variable is a variable whose value you would like your compressed model to be able to closely reproduce. Some location variables can be statically known items (e.g., starting reserves per unit); others can be projection results from a small number of *calibration scenarios* (typically one to three), such as:
 - reserves, cash value, account value, or premium per unit as of the projection date;
 - present value of GMB claims per unit;
 - sum of the premiums paid in the first five years of the projection per unit;
 - first-year liability cash flow per unit;
 - present value of profit (PVP) per unit.
- A size variable to represent the importance of a given policy. This ensures that large policies are not mapped away as readily as small policies, all other things being equal. For example, the size variable would typically be represented by face amount for life insurance or account value for deferred annuities.
- A *segment*; the program will not map across segment boundaries. Segments might be plan code, issue year, GAAP era, or any other dimension of interest. Reasons for using segments include:
 - To decrease calculation time, which is roughly proportional to the sum of the squares of the number of policies in each segment; a group run as one segment will take approximately 10 times as long as the same business split into 10 equal-sized segments. Assuming that the segments serve to separate policies that would be unlikely to be mapped together in any case, the results would be essentially the same.
 - For reporting, reconciliation, or similar reasons, where you might wish to keep policies from one

FOOTNOTES

¹ Available at <http://www.milliman.com/expertise/life-financial/publications/rr/pdfs/cluster-analysis-a-spatial-rr08-01-08.pdf>

segment of business from being mapped into another segment.

- Whenever the location variables by themselves do not, in your judgment, sufficiently distinguish policies in different segments.

The program then proceeds as follows:

1. The *distance* between any two policies is calculated using an n-dimensional sum-of-squares approach, as if the n location variables defined a location in n-dimensional space. Thus, as an example, with three location variables, *Var1*, *Var2* and *Var3*, the distance between policy 1 and policy 2 could be measured as:

$$\sqrt{(Var1_1 - Var1_2)^2 + (Var2_1 - Var2_2)^2 + (Var3_1 - Var3_2)^2}$$

In this definition, the location variables must be appropriately scaled. Each of the location variables is normalized by dividing each one by the size-weighted standard deviation of the associated variable. Users can also introduce weights to place different priorities on matching different distance variables.

2. The *importance* of each policy is defined by the cluster modeling process as the size times the distance to the nearest policy. Thus, a policy is unlikely to be mapped to another if it has a large size and is far away from others; however, a small policy or one that is very close to another is likely to be mapped to another policy.
3. At each step, the process finds the policy with the lowest importance and maps it to its nearest neighbor (*the destination policy*), adjusting the size, and hence the importance, of the destination policy in the process. This step is repeated until the model has the desired number of model points.
4. At this point, only the user-specified target number of clusters remains. In the next step, the program finds the most representative policy in each cluster, which is the policy in each cluster that is closest to the average location (centroid) of all cells in the cluster. In general, each cell in the compressed in-force file will consist of a policy from the original in-force file, scaled up (i.e., with all variables that are

logically additive grossed up by the size of the entire cell group over the size of the original policy, and all other variables taken from the original policy). For certain variables, you may prefer instead to sum the values from the various policies mapped into the cell, although this should be done sparingly because the distance methodology in essence assumes that cells will be scaled up.

The pictures below can help demonstrate the cluster modeling process. In it, we assume just two location variables that reflect two dimensions. The scatter plot in Figure 1 represents the value of each location variable by the point placement on the two-dimensional graph. The size of each dot represents the size of the policy. In Figure 2, each policy has been assigned to a cluster. Finally, the resulting four-point model is shown in Figure 3 with the size of the four model points appropriately grossed up:

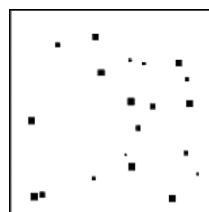


Figure 1

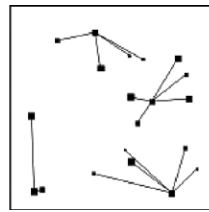


Figure 2

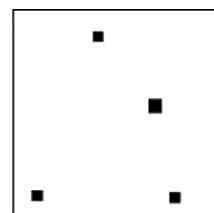


Figure 3

IMPLEMENTING THE TECHNIQUE

A naive implementation of the process, shown using R, might be as appears on page six. There we use an example involving six policies (from two segments), and three location variables representing nothing in particular.

CONTINUED ON PAGE 6



Avi Freedman, FSA, MAAA, is a consulting actuary with the life insurance consulting practice of Milliman in New York, NY. He may be reached at avi.freedman@milliman.com.



Craig W. Reynolds, FSA, MAAA, is a principal with the life insurance consulting practice of Milliman in Seattle, Wash. He may be reached at craig.reynolds@milliman.com.

1	#####setup
2	loc0 <- read.table(textConnection(“
3	23. 15. 13.
4	10. 20. 30.
5	24. 15. 13.
6	10. 26. 30.
7	25. 15. 13.
8	10. 20. 31.
9	“))
10	siz <- c(49, 25, 5, 50, 50, 100)
11	segments <- c(0, 1, 0, 1, 0, 1)
12	weights <- c(1, 1, 10)
13	endcells <- 3
14	#####calculations
15	loc1=loc0*siz
16	avgloc <- apply(loc1,2,sum)/sum(siz)
17	vars <- (apply(loc1*loc1/siz,2,sum)/sum(siz))-avgloc*avgloc
18	scalars <- weights/sqrt(vars)
19	loc <- t(scalars * t(loc1/siz))
20	elements <- length(siz)
21	mappings <- elements - endcells
22	listel <- 1:elements
23	z1 <- rep(listel, elements)
24	z2 <- sort(z1)
25	use <- segments[z1] == segments[z2] & z1 != z2
26	z1 <- z1[use]
27	z2 <- z2[use]
28	diff <- loc[z1,]-loc[z2,]
29	dist <- sqrt(apply(diff*diff, 1, sum))
30	siz2 <- siz
31	dest <- listel
32	for (tt in 1:mappings) {
33	keep <- dest[z1]==z1 & dest[z2]==z2
34	z1 <- z1[keep]
35	z2 <- z2[keep]
36	dist <- dist[keep]
37	importance <- siz2[z1]*dist
38	index <- order(importance)[1]
39	tempfrom <- z1[index]
40	tempto <- z2[index]
41	print(c(tempfrom, tempto))
42	dest[dest == tempfrom] <- tempto
43	siz2[tempto] <- siz2[tempfrom] + siz2[tempfrom]
44	siz2[tempfrom] <- 0
45	}
46	print(dest)

Location variable, size and other information is given in lines two through 13. Lines 15 through 19 perform the normalization. Lines 20 through 27 limit comparisons to pairs of cells in the same segment, and 28 and 29 calculate distances. Lines 30 through 42 correspond to steps two and three. The above script does not implement step four; this is left as an exercise for interested readers.

When run, the script will show cell three mapped into cell one, then five into one, then two into six. Examination of the data will show why this was done. Cells three and one are close, and cell three is very small. Cells five and one are close, and when cell three is mapped into cell one, cell one is larger than cell five, and so cell five is mapped into cell one. The other segment, consisting of cells two, four and six, is more spread out when the large weight on the third location variable is taken into account, so only now does a mapping occur in that segment. Understanding why two is closer to six than to four requires working out the standard deviations.

While the R code given above is helpful for understanding the algorithm, it is unsuitable for any real use, since it uses too much space and runs very slowly. In the production implementation, memory usage is reduced by not keeping around all of the distances, and recalculating missing distances as needed. Runtime is minimized by using C++, including compiler intrinsics for SSE instructions, instead of R; by distributing calculations over multiple cores where appropriate; and by giving careful consideration to when calculations may be deferred (and probably ultimately rendered unnecessary) without changing the results.

OPEN QUESTIONS

There are numerous questions that will need to be faced by companies adopting cluster modeling techniques. Because the techniques are still new, it is uncertain what choices companies will make in practice; these choices will depend on the specific circumstances.

How close is close enough?

We worked with a client on creating a cluster model to determine the market value of liabilities (MVL) averaged over multiple scenarios, both for a basic set of

scenarios and for sets of shocked scenarios. Differences (compared to a near-serial model) were very small (within about five bp of starting account value) for the baseline and liability shocks, but somewhat larger for economic shocks (up to about 20 bp for a large downward equity shock). (SEE TABLE 1), Pg. 8

A company in that situation might well choose to use cluster models for the insurance shocks but continue to use the larger model for economic shocks (while searching for a set of location variables and calibration scenarios that would work better). Or it might choose to use cluster models for all of the shocks, and use the computing resources freed up to run larger sets of scenarios and reduce the sampling error caused by scenario selection.

How should cell clustering and scenario clustering be mixed?

The clustering technique can also be used to select scenarios (in this case, there is only one segment, and all scenarios have weight one). Location variables can be based on the scenario inputs (e.g., wealth factors) and/or on results from a model. The model itself may be a very small cluster model; the resulting set of scenarios may then be run against the full model or a relatively large cluster model. Alternatively, of course, companies can rely entirely on cell clustering and ignore the scenario clustering functionality. More information as to the benefits of the various approaches will develop as companies gain experience with the technique.

What model sizes are best?

Cluster models of 2,500 cells will, of course, take longer to run through various scenarios than models of 250 cells. On the other hand, the relatively larger models have their advantages. The advantage is not so much in being able to serve usefully across a wide range of stochastic scenarios under a single set of assumptions, but more from the increased robustness in the face of changes either to liability assumptions or to the assumptions underlying the scenario generator. In order to determine the optimal size, companies will need to consider how broadly they intend the models to be used and how large of a model they are willing to have.

CONTINUED ON PAGE 8

Table 1				
Market Value of Liabilities				
Calculated using full model and cluster model				
Per \$10,000 of starting account value				
Under base assumptions and 11 shocks				
	Full	Cluster	Full	Cluster
	MVL	MVL	Shock	Shock
Base	9,896	9,899		
1	9,847	9,851	-49	-48
2	9,928	9,930	32	31
3	9,555	9,557	-340	-342
4	10,134	10,140	239	241
5	9,824	9,827	-72	-73
6	9,961	9,966	65	67
7	9,872	9,875	-24	-25
8	9,923	9,926	27	27
9	9,967	9,979	71	80
10	7,501	7,487	-2,395	-2,412
11	10,317	10,328	421	429

if at all, only for a quick assessment of the effect of liability assumption changes. Others will use cluster modeling for most work where model efficiency is a concern, using replicating portfolios, if at all, only for communication with investment personnel or where corporate staff wishes to assess liability results under different scenarios without using liability modeling software.

It should be noted that cluster modeling can enhance replicating portfolio techniques. Results from a cluster model for a large number of economic scenarios can be better as input to the replicating portfolio procedure than results from a full model against a smaller number of scenarios, as more data points should result in better replication fit. ■

What should the interplay be between clustering and replicating portfolios?

In recent years, replicating portfolio techniques have become popular avenues for addressing model efficiency concerns. In our view, this is something of a historical accident: Techniques that are a plausible solution to a different problem (communicating with people who do not understand and/or do not care about insurance liabilities) were, out of convenience, pressed into service to address a different concern (model efficiency) to which they are less well suited. Replicating portfolio techniques require a fairly large number of scenarios to be run with a large model; furthermore, for any sensitivity in the liability assumptions, another set of such scenarios must be rerun.

Our preference for cluster modeling techniques, however, may not be universally shared. Some companies may rely on replicating portfolios for most work where model efficiency is a concern, using cluster models,

Cool Tech

By Matthew Wilson

For this edition of Cool Tech I'm going to show you more than 200 free applications that you can download and try out. They're all free but not necessarily open source.

I start off with OpenDisc, which contains 40 free programs. You download an ISO image then burn a CD from the image. Making a CD from an ISO image is slightly different than copying a file to CD.

Next we go to PC Magazine's list of 173 free programs for you to try out. The programs range from application launchers, audio and backup all the way to networking and video applications.

Then there is BitNami. BitNami, or more correctly BitNami stacks, makes "it incredibly easy to install your favorite open source software," according to their Web site.

About a month ago I tried an experiment to see if I could improve traffic to my blog. I show you what I did and my results.

Finally, I briefly go over a Google custom search engine, a mail list manager, Feedblitz and a couple of Perl programs for scraping news articles off of your favorite news Web site.

MY VIEW

There are so many free applications available these days. It really pays to do a little investigating before purchasing a program or writing it yourself. Many Web hosts install open source applications for free.

Here are a few Web sites worth checking out:

www.Koders.com—Code snippets.

www.Hotscripts.com, www.Sourceforge.net—Open source software.

www.Bitnami.com, www.Apachefriends.org/en/xampp.html—Application packages.

www.Fantastico.com, www.Installatron.com—Typical Web host packages installed for free.

www.Elance.com—Hire help.

Recently, I was playing around with StumbleUpon (www.stumbleupon.com) and it served up a game that sucked me in. StumbleUpon serves up pages that might interest you based on your viewing history.

The game, Hell of Sand, has several streams of sand dropping into an area where you can draw walls to affect the flow. I probably spent 20 minutes drawing walls changing the sand flow. Ok, I know it sounds stupid. I got sucked in.

Hell of Sand:

<http://www.andyslife.org/games/sand.php>

If you're interested in games, then the *BUSINESS WEEK* Arcade has a cool set of independently produced games on the Web. And the games are free.

<http://tinyurl.com/dhlszw>

Did you notice that I'm using "tinyurl.com" for the URL. Check out <http://tinyurl.com> as a way to take those long URLs and shorten them for easy manual input.

It turns out that tinyurl.com is on *TIME* magazine's list of top 50 Web sites for 2008: <http://tinyurl.com/cc45dl>.

TIME magazine doesn't present the list in a convenient format, so I created my own list by using the advanced search features from Google:

http://www.google.com/advanced_search?hl=en

I searched the *TIME* Web site (Time.Com) for "50 Best Websites 2008." I included omitted results in order to get 144 links. I grabbed the HTML code and cleaned it up in a text editor. Later I show you how to clean up Google HTML using Regular Expressions (RegEx). My results are here: <http://tinyurl.com/cc45dl>.

Here's the search text: "50 Best Websites 2008," site time.com



Matthew Wilson,
ASA, MAAA,
can be reached at
matt_wilson@farmersinsurance.com

CONTINUED ON PAGE 10

Stratfor.com is an intelligence Web site that charges about \$300 per year for a subscription. Since I don't particularly want to spend that much for a subscription, I do the next best thing. I use the advanced search features in Google to see what I can grab for free.

Search the stratfor.com Web site and target filetype PDF. site: stratfor.com, filetype.pdf

The search results turn up over 50 reports. Here is their annual intelligence forecast for 2009:
<http://tinyurl.com/c75u49>.

Now you know how to use Google to grab interesting material from a Web site. What about developing some spiders to grab more interesting items? At the end of this article I share a couple of Perl programs on spidering. You can get more information on this topic by checking out these books:

Spidering Hacks—<http://tinyurl.com/c2qvn3>

Webbots, Spiders and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL—<http://tinyurl.com/dn4pck>

Spidering Hacks will introduce you to Perl and Regular Expressions for scraping purposes.

OpenDisc

OpenDisc is literally a disk or ISO image that you can download containing a lot of open source programs that work on Windows. I've summarized the list of programs below. You can get a better description of the programs on this page:

<http://www.theopendisc.com/programs/>

Design (Seven Programs)

Blender (3D graphics), Dia (Flow Charts), GIMP (Image Manipulation), Inkscape (Vector Graphics Editor), Nvu (Web page Editor), Scribus (Desktop Publishing) and Tux Paint (Children's Paint Program).

Games (Four Programs)

The Battle for Wesnoth, Enigma, Neverball and Neverputt.

Internet (10 Programs)

Azureus (bitTorrent Client), FileZilla (File Transfer), Firefox (Browser), HTTrack (Offline Browsing), Pidgin (Instant Messaging), RSSOwl (RSS Reader), SeaMonkey (Internet Suite), Thunderbird (E-Mail), TightVNC (PC Remote Control) and WinSCP (Secure File Transfer).

Multimedia (Six Programs)

Audacity (Audio), Celestia (Universe Simulation), Really Slick Screensavers, Stellarium (Planitarium), Sumatra PDF (PDF Reader) and VLC (Media Player).

Productivity (Five Programs)

GnuCash (Accounting), MoinMoinWiki (Wiki), Notepad2 (Replace Notepad), OpenOffice (Replace Microsoft Office) and PDFCreator (Create PDF Files).

Utilities (Eight Programs)

7-Zip (Replace WinZip), Abakt (Backup Tool), Clamwin (Anti-Virus), GTK+ (Create Graphical User Interfaces), HealthMonitor (Windows Monitoring Tool), TrueCrypt, Encryption and Workrave (Prevent Repetitive Strain Injury).

Burn a CD from an ISO image

If you want to burn a CD for the OpenDisc programs, then check out the following link for software capable of burning a CD from an ISO image:

http://en.wikipedia.org/wiki/List_of_ISO_image_software

The Best Free Software of 2009

PC Magazine's mother load of free software list:

<http://www.pcmag.com/article2/0,2817,2338803,00.asp>

Here's the print version of the article with everything on one page:

http://www.pcmag.com/print_article2/0,1217,a%253D235942,00.asp

There are 173 software packages listed below. The article contains links to each application. Also, some applications are classified as in the “Hall of Fame.” The application categories:

App Launchers (four), Audio/Music (11), Backup (six), Blogs (eight), Browsers (nine), Calendar/PIMs (10), Communication/E-Mail (10), Conferencing (four), File Transfer/Download (eight), File Viewers/Converters (eight), Finance (four), Fun/Home (seven), Graphics (17), Ims (four), Interface Enhancers (13), Local Search (three), Office (19), Operating Systems (three), Networking (nine), RSS Readers (six), Synchronization (five) and Video (five).

BitNami

BitNami stacks make it “easy to install your favorite open source software.” A stack includes a primary application plus all the dependencies necessary to run it. The installation wizard makes it very easy to complete the installation process quickly.

If you’ve ever tried to resolve dependency issues then you know that they can be a real black hole at times. They can suck up huge amounts of time if you let them. Here’s the link:

<http://bitnami.org>

Try to find the general category of software that interests you. Then go to the Bitnami Web site to find out a little more about a specific application:

- **Infrastructure**
 - DjangoStack, JRubyStack, LAMPStack, LAPPStack, MAMPStack, MAPPStack, RubyStack, SAMPStack, WAMPStack and WAPPStack
- **Blog**
 - Roller and WordPress
- **Bug-Tracking**
 - Mantis, Redmine and Trac
- **Business Intelligence**
 - JasperServer
- **Content Management System (CMS)**
 - Alfresco, Drupal, Enano CMS, eZ Publish, Joomla and KnowledgeTree
- **Client Relations Management (CRM)**

- SugarCRM
- **ECM**
 - Alfresco and KnowledgeTree
- **Forum**
 - phpBB
- **Photo Sharing**
 - Coppermine Photo Gallery
- **Planning**
 - Tracks
- **Poll Management**
 - Opina
- **Portal Server**
 - JasperServer and Liferay
- **Version Control**
 - Subversion
- **Wiki**
 - DokuWiki and MediaWiki
- **eLearning**
 - Moodle

HOW TO INCREASE YOUR BLOG TRAFFIC BY 30 PERCENT IN ONE MONTH

I’ve been running a blog since 2006. I have now posted thousands of articles. Recently, I started examining my Google Analytics which show how many hits specific pages got in the last month. I noticed that some older articles are still continuing to get a decent amount of traffic. I wondered if it was possible to improve these pages in order to get even more hits.

Google Analytics allows you to drill into the statistics for a specific page. For example, you can find out which keywords people used to find that page.

Once I knew what people were looking for, I set out to improve specific pages. I added more content specific to the top keywords for that page. For content I added excerpts from more news articles, content from Wikipedia and targeted videos from YouTube.

One article that I posted awhile back is called “What is Suter?” Suter is the name of a system that Israel used to hack into Syria’s attack air defense system as their jets were flying toward a secret Syrian nuclear facility. I guess that’s literally hacking on-the-fly. I thought

CONTINUED ON PAGE 12

people would find this page by searching for Suter. It turns out that most people were finding the page while searching for “Big Safari.” Suter is a part of the Big Safari project run by the U.S. Air Force. I had only briefly mentioned the Big Safari project in the original article.

Once I learned Big Safari was rather important, I added a paragraph explaining more about it.

I found out which tags are most popular. For example, the tag “global-financial-crisis” is more popular than “financial-crisis.” Unfortunately, I had been mostly using the tag financial-crisis. So I went back to many articles in the financial-crisis category and added the tag global-financial-crisis.

I found a few popular pages that surprised me. I posted one page on Google Earth secrets that was very popular. So I went back to this page and beefed up the content even more.

I found a few new categories that people were interested in. One category is on the powerful global elite. For example, the 50 most powerful people in the world or the world’s top billionaires. Another category is on Mexico, but this pertains more to the crisis in Mexico. These types of pages were very popular, so I created entirely new categories just for these topics.

I recently started something new as well. I started using Twitter (<http://www.twitter.com>). Of course, with a name like that it took someone hitting me over the head before I would give it a try. Now I’m hooked.

Think of twitter as a mini-blogging system. You have a maximum of 140 characters in each post. So why bother? I use it to post article titles and links. You can direct people back to your blog or to new articles on a topic.

With long URLs in use today, it can be difficult to squeeze an article title and link into 140 characters. Fortunately, you can use www.tinyurl.com to shorten the length of your links.

Sometimes I use content from Google in order to give

my readers more article choices. Unfortunately, I hate the HTML code that Google uses. So I need to spend a few minutes cleaning up the code. Here’s how I do it:

1. I use the ConText text editor which provides RegEx in its Find/Replace section.
2. I use Regular Expressions (RegEx) code for the clean-up.
 - a. Replace `<a .*?href` with `<a href`
 - b. Replace `<div.*?>` with blank
 - c. Replace `<td.*?>` with blank
3. “`.*?`” means any character 0 or more times. “`.*`” means any character. “`*`” means 0 or more times. The “`?`” in this context means stop as soon as possible.
4. You can clean up Wikipedia pages using `\[.*?\]` which removes those pesky edit and reference marks.
5. You can find a tutorial on RegEx here: http://www.1913intel.com/demo/RegEx_Tutorial.zip

MAIL LIST MANAGER

Here is a free e-mail list manager. Now you can manage your e-mail lists yourself. You will need to set up a MySQL database to get this working, but that’s usually pretty easy. Ask your host if you don’t know how to create a database.

<http://www.phplist.com>

<http://en.wikipedia.org/wiki/phplist>

I used this package for about a year with good results.

I now use Feedblitz (www.feedblitz.com) for convenience. Feedblitz automatically scans my Web site each night and sends out an e-mail with new articles to my list. It’s free, and it’s easy for people to sign up or remove one’s self from the list.

I also use Feedblitz for a newsletter where the content comes from Google. I feed Google news into Yahoo! Pipes (<http://pipes.yahoo.com/pipes/>). Feedblitz automatically checks my link (<http://tinyurl.com/ctbfn9>) and sends out an e-mail to the subscribers. There are about 200 subscribers to this newsletter.

SOME CUSTOM SEARCH ENGINES

Google has a custom search engine where you select the sites to be searched. Just enter the URLs that you want to be searched when configuring the search engine.

<http://www.google.com/coop/cse/>

Check out my instance of a Google custom search engine:

<http://www.1913intel.com/custom-search/>

Try entering “Russia” in the search box.

Since Google came up with this custom search engine, the following section is not really needed. However, the scrapers/spiders are still pretty cool to play around with.

DOWNLOAD THREE PERL PROGRAMS

The first program, `xoopsider4.cgi`, reads several news Web sites and grabs the links to news articles. The results are printed to the screen. The program is capable of writing out results to a file, but I’ve commented out that part of the code. Go into subroutine `writeFILES` and uncomment the code in order to write to files.

Now you can quickly scan articles from your favorite news sites.

Don’t forget to change the properties of `xoopsider4.cgi` so you can execute it. You’ll need to do this to all CGI programs.

This program reads one Web site at a time, so you’ll want to keep the number of Web Sites down to a manageable level.

`newsScaper4.cgi` is the hardcore version of `xoopsider4.cgi`. It uses parallel processing, so you can load up on the Web sites. It only writes to files. However, you’ll need to set up the `ParallelUserAgent` folder using the `ParallelUserAgent-2.57.zip` file.

In the past I developed `newsScaper4.cgi` for a custom search engine. I placed the results from scraping into a searchable MySQL database. Users simply entered keywords and pressed a button to get relevant articles for that day.

The second program, `parallelSpider.cgi`, grabs several Web pages in parallel and prints them to the screen.

<http://www.1913intel.com/demo/scrapers.zip>
<http://www.1913intel.com/demo/ParallelUserAgent-2.57.zip>

You will need to unzip `ParallelUserAgent-2.57.zip`. Place the unzipped folder on your Web site and point your program to the library.

This is a basic program showing how to do parallel processing. ■

New Report: Blue Ocean Strategies for Life Insurance Industry

A new study identifying and debating possible new approaches to acquiring business by life insurers is now available on the SOA Web site. Sponsored by the Futurism, Marketing and Distribution and Technology sections, this Delphi study gathered expert opinions as to whether there were any such “Blue Ocean Strategies” in technology for business acquisition that could affect the life insurance industry during the next 10 years.

To view the report, go to www.soa.org, click Research, Research Projects and Life Insurance.

R Cornerⁱ—Model Formula Framework

By Steven Craighead



Steve Craighead, ASA, MAAA, is an actuarial consultant at TowersPerrin in Atlanta, Ga. He can be reached at steven.craighead@towersperrin.com.

Editor's note: R Cornerⁱ is a series by Steve Craighead introducing readers to the “R” language used for statistics and modeling of data. The first column was published in the October 2008 issue, and explains how to download and install the package, as well as providing a basic introduction to the language. Refer to each CompAct issue since then for additional articles in the series. The introductory article can be found on p. 24 of the October 2008 on the SOA Web site: <http://soa.org/library/newsletters/compact/2008/october/com-2008-iss29.pdf>

In this article we will examine the Model Formula Framework within R¹ for linear and generalized linear models. You may use this framework to set up a large number of different statistical models.

Since we are going to concentrate on linear-like models, we will assume that both the predictors and the resultant variables are continuous. You may also use the framework to model Analysis of Variance (ANOVA) models on factor or categorical data (variables that take on discrete values), but that will be discussed in a future column.

The simplest formula will look like this:

$$\text{observed} \sim \text{predictor1} + \text{predictor2} + \text{predictor3} + \dots + \text{predictorN},$$

where *observed* is the variable that you wish to model, by determining some relationship with the various predictor variables. Note: The “+” convention is used to include a variable in the model in a linear fashion. For a linear regression, the actual model that is fit is of this form:

$$\text{observed} = (C1)\text{predictor1} + (C2)\text{predictor2} + \dots + (CN)\text{predictorN} + \text{residual_error}.$$

Here the *C_n* denote the separate coefficients of this model.

If you want to model the observed against all variables (except itself) in a dataset, you may use the “.” convention, such as:

$$\text{observed} \sim .$$

If you want to eliminate a specific term within a model, you may use the “-” convention,

$$\text{observed} \sim -1 + \text{predictor1} + \text{predictor2} + \dots + \text{predictorN}$$

Here the “-1” indicates that you do not want to have an intercept term calculated within your model.

For instance, if you want to model observed against all predictors except for predictor3, you can use the “.” and “-” in this way:

$$\text{observed} \sim . - \text{predictor3}$$

The symbol “.” with continuous variables indicates the actual product of variables. The symbol “*” denotes a factor crossing. So that *predictor1*predictor2* denotes *predictor1 + predictor2 + predictor1:predictor2*. Note how “*” is not a true product of the variable in the same way that “+” is used above. The “^” symbol denotes a factor crossing to a specific degree. For instance, *(predictor1 + predictor2 + predictor3)^2* is the same as *(predictor1 + predictor2 + predictor3)*(predictor1 + predictor2 + predictor3)*, which is a formula of the form:

$$(\text{predictor1} + \text{predictor2} + \text{predictor3} + \text{predictor1:predictor2} + \text{predictor1:predictor3} + \text{predictor2:predictor3})$$

Usually, you will just use the actual variable names, but you can use functions of the variables as well. For instance, *exp(observed) ~ exp(predictor1)* is a valid formula. Now, note however, that there appears to be a contradiction to this format, however, if you need to create formulae that actually need the normal arithmetic meaning of the operators. You overcome this by

FOOTNOTES

¹ R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.

using a $I()$ format to surround the model components that need to actually use these arithmetic operators. So if you wanted to create a formula that actually adds two predictors together, before creating the model, you would use a form like this:

```
observed ~ I(predictor1+predictor2)
```

Now, to further clarify the idea of the “.” format from above, $predictor1:predictor2$ is the same as $I(predictor1*predictor2)$.

If you want to create transformed observed and predictor models, you can use a formula like this:

```
I(1/Observed) ~ I(1/predictor1^2) + sqrt(predictor2)
```

Here the multiplicative inverse of the observed variable is fit against the multiplicative inverse of $predictor1$ squared and the square root of $predictor2$.

If you wish to create a polynomial model of a predictor, you may use the $poly(.)$ convention. For instance, suppose you want to model $predictor1$ as a cubic equation and $predictor2$ as quadratic and $predictor3$ as linear. The formula would look like this:

```
observed ~ poly(predictor1,3)+poly(predictor2,2) + predictor3
```

In generalized linear models, you may use the $s()$ convention. When you surround a variable by this convention, it tells R to fit a smoothed model. For instance:

```
observed ~ s(predictor1) + predictor2 + s(predictor3)
```

would fit $observed$ by creating non-parametric smoothed models of $predictor1$ and $predictor3$ and use the actual values of $predictor2$.

Let’s look at a couple of examples of a linear regression model using some of the formulas above. Define a dataset containing three predictors (say the variable names are x , y and z). In our model, we will let x contain 100 samples of the standard normal distribution, y will con-

tain 100 samples of the continuous uniform distribution on the interval (2,5), and z will be $(x+y)^3$. The observed variable will be r . Let r take on the values of $x^2 + 1/y + z$. To generate these, use the following commands:

```
x <- rnorm(100,mean=0,sd=1)
y <- runif(100,min=2,max=5)
z <- (x + y)^3
r <- x^2 + 1/y + z
```

Now, create a dataframe called $RTest$ containing these variables by executing this command:

```
RTest <- data.frame(cbind(r,x,y,z))
```

Here $cbind()$ will concatenate the four variables into a matrix whose columns are r , x , y and z . The $data.frame()$ function then converts the matrix into a dataframe. If you type

```
(names(RTest))
```

R will display the separate column names:

```
[1] “r” “x” “y” “z”
```

Using the “.” convention, create the following simple regression model:

```
(Model <- lm(r ~ ., data=RTest))
```

Note how the $data$ input parameter is used to reference the dataframe $RTest$. Now, R will display:

Call:
lm(formula = r ~ ., data = RTest)

Coefficients:

(Intercept)	x	y	z
4.217	-1.175	-1.303	1.028

So the linear model is:

```
r = 4.217 + -1.175x -1.303y+1.028z
```

CONTINUED ON PAGE 16

To observe additional information regarding the model, use this command:

```
(summary(Model))
```

R will display:

Call:

```
lm(formula = r ~ ., data = RTest)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-1.5187	-0.6738	-0.3520	0.4036	3.1998

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.217229	0.537224	7.850	5.89e-12 ***
x	-1.175125	0.195919	-5.998	3.52e-08 ***
y	-1.303079	0.199134	-6.544	2.93e-09 ***
z	1.027829	0.003924	261.915	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.044 on 96 degrees of freedom
Multiple R-squared: 0.9998, Adjusted R-squared: 0.9998
F-statistic: 1.354e+05 on 3 and 96 DF, p-value: < 2.2e-16

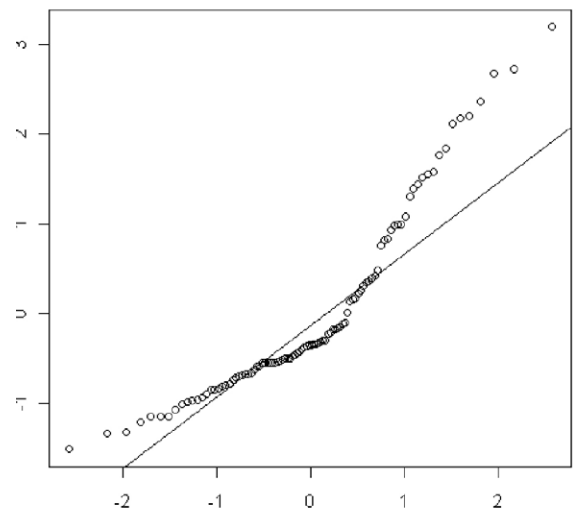
The F-statistic is very significant as well as each coefficient and the R-squared values are almost 1, so you may be tempted to just stop here and just use this simpler model. However, if you use the `plot()` function, you will see that the model's behavior is not that good. There are four separate graphs produced with the `plot()` function, but below we will only display the Quantile-Quantile (Q-Q) plot, to demonstrate that the residuals are not normal. We will use these commands to plot just Q-Q plot:

```
qqplot(resid(Model))
qqline(resid(Model))
```

R will display the graph above (right):

If the residuals were actually normal, the sorted residu-

Normal Q-Q Plot



als would follow the 45-degree line. However, this graph indicates that the left tails are actually heavier than a normal curve and the right tail is lighter, so since linear regression models require the residuals to be normal, we can say that this model is faulty. Also, note how the tails are not symmetric.

Let's examine the results of the actual formula that we used to model r, as a regression model:

```
(Model2 <- lm(r ~ I(x^2) + I(1/y) + z, data=RTest))
```

R displays this:

Call:

```
lm(formula = r ~ I(x^2) + I(1/y) + z, data = RTest)
```

Coefficients:

(Intercept)	I(x^2)	I(1/y)	z
6.879e-15	1.000e+00	1.000e+00	1.000e+00

Notice how the coefficients are all equal to one, but there is an intercept term, which isn't in the original model, so revise the regression to eliminate the intercept term:


```
(Model3 <- lm(r ~ -1 + I(x^2) + I(1/y) +
z,data=RTest))
```

R displays:

Call:
lm(formula = r ~ -1 + I(x^2) + I(1/y) + z, data = RTest)

Coefficients:

I(x^2)	I(1/y)	z
1	1	1

Now examine the results of summary:

```
(summary(Model3))
```

Call:

lm(formula = r ~ -1 + I(x^2) + I(1/y) + z, data = RTest)

Residuals:

Min	1Q	Median	3Q	Max
-1.298e-13	-3.347e-15	-1.798e-16	4.946e-15	3.037e-14

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
I(x^2)	1.000e+00	1.372e-15	7.289e+14	<2e-16 ***
I(1/y)	1.000e+00	7.584e-15	1.319e+14	<2e-16 ***
z	1.000e+00	2.278e-17	4.389e+16	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.655e-14 on 97 degrees of freedom

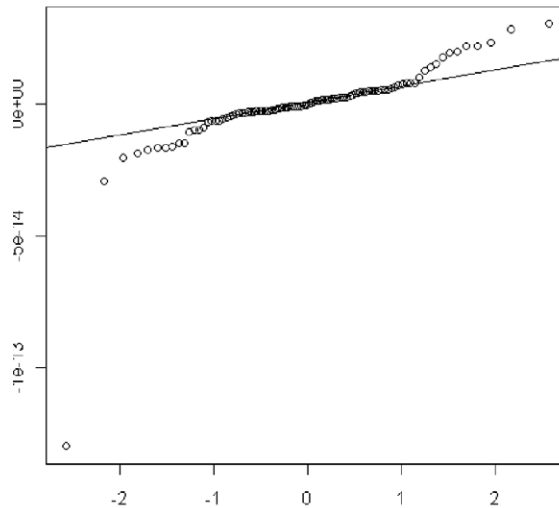
Multiple R-squared: 1, **Adjusted R-squared:** 1
F-statistic: 1.148e+33 on 3 and 97 DF, **p-value:** < 2.2e-16

Note how the F-statistic is larger than the first model above and that the t values have greater significance as well. Note that the R-square statistic is also equal to one. Now, look at the Q-Q plot:

```
qqnorm(resid(Model3))
```

```
qqline(resid(Model3))
```

Normal Q-Q Plot



Notice how the tails are symmetric like a normal distribution. Also, both tails are slightly lighter than normal. But, the largest residual in absolute value is $-1.298e-13$, which is effectively zero, so we can say that this model is definitely very good.

If you want to see how R explains how to use the formulae format, please use the help command:

```
help(formula)
```

Another good resource on how to use the formulae format is “The R Book” by Michael J. Crawley. This book is published by Wiley and its ISBN is 978-0-470-51024-7.

In the next article, I will actually use R to create an efficient actuarial modeling technique by using my two most favorite non-parametric models with R. These models are the CLARA clustering algorithm and the Projection Pursuit Regression (PPR) predictive model. I will use CLARA to extract a small set of representative scenarios from a collection of 10,000 scenarios, and then I will use PPR to create a predictive model of specific corporate surplus results. I will also demonstrate the effectiveness of this combined approach when trying to quickly model the Conditional Tail Expectation (CTE) on the surplus. ■

WANTED:

Reviews and Articles on Life Insurance and Annuity Illustration, Needs Analysis, and Advanced Marketing Systems

One of our goals is to have our readers or others submit articles on commercial software that is used to sell and market life insurance or annuities. Please consider submitting your own experiences and information, or contacting anyone you may know that would have potential input and useful information.

- **If your company has recently evaluated vendors and technologies, please share your knowledge!** We would love to know the process you followed, how you evaluated vendors, what you evaluated, and the results. Send any contact information you have to the editor of CompAct.
- **If your company uses commercial software to sell and market your products, we would love to have an article submitted on the process you follow to maintain and update it, or on your satisfaction with the solutions provided.**
- **If you use illustrations as part of your pricing process, those systems would be included in this call for reviews.** Most actuarial modeling systems are capable of producing illustrations, so we would love to hear your thoughts on the system you use. We are especially interested in how you selected the system, how easy it is to use and maintain, and the support that is offered.
- **If you use commercial software and have time to evaluate, or know of someone who can write an article on the features and functionality, please forward your name to the editor of CompAct.**

- **If you are a software vendor:** CompAct will be contacting vendors to provide general articles and information. If you are interested in submitting articles on the industry or the software process in general, please send a note to the editor of CompAct.

PROCESS

Ideally, we would love to do a roundup of the software available, including end-user evaluations and support ratings, much like you would see in a computer magazine or consumer report. Unfortunately, we do not have the resources for this, and are hoping to produce less complete, but more timely information. There will not be numerical or quantitative values assigned, because these will not be independent or complete reviews. Therefore, no editor's choice awards or rankings will be provided. The goal is to provide a resource for software available, how it is selected, and user experiences.

We intend to cover software areas of interest to actuaries (pricing, valuation, etc.), and will use this as a template for future software coverage. Although actuaries are not end users of illustration software, there is a large amount of interaction with these types of systems, and it is typically part of the pricing process. If you are interested in writing articles for a different software category, please feel free to submit those ideas or articles as well.

Software vendors may find champions to write articles, since there is little incentive for individuals to volunteer. Although this certainly results in bias, there is a lot of information to be gained from these articles. To ensure technical accuracy, vendors will be given the chance to respond and comment on articles submitted that pertain to their software.

We will not require a complete evaluation of all aspects of software packages. Sample topics are listed below, and we will be developing a way to index the articles based on what aspect is being reviewed. Articles can cover one topic, to encourage as many submitters as possible. Although this makes it difficult to compare vendors, and does not give us the ability to do a full assessment, we hope to be able to pull the information into one edition eventually. If time permits, we will include an overall features chart for system functionality.

SAMPLE TOPICS

- Reasons for change.
- Build vs. Buy.
- Process for selecting a system (RFI/RFP, etc.).
- Process for implementing a system.
- System functionality/features (Usability, Straight through processing, etc).
- Overall satisfaction, User Interface “Friendliness,”

Ease of Customizing, Integration.

- Vendor attributes (Responsiveness/Support, Cost, Delivery, Knowledge base, Process).

DISCLAIMERS

The editor of CompAct has extensive experience with this topic, so the decision to start with illustration software is a pragmatic one. Being previously employed by iPipeline, a large illustration software company, means potential conflicts of interest issues must be addressed. The articles will be written independently. Articles will be forwarded to the vendor for comment on content and accuracy. Any issues or concerns will be submitted to the section chair for resolution.

The articles will represent the opinions and experiences of the authors, and will not be substantiated or endorsed by the Technology Section, or the SOA. It needs to be clear that individual experiences may not be representative of a vendor’s current practices or software. Our hope is that articles will focus on the software functionality, features and vendor currently used, and that the submitters will be fair and honest in their assessments.

Although we hope to produce a features chart to help distinguish software features and functionality, numeric grades or evaluations will not be assigned, nor will editor’s choice or any other rankings be provided. ■

The End Users Justify the Means: IV The Journey Home

By Mary Pat Campbell



Mary Pat Campbell, FSA, MAAA, is a vice president at The Infinite Actuary. She can be contacted at marypat.campbell@gmail.com.

And so I come to the end of this series on spreadsheet design, concentrating on what I consider the most important set of end users: the maintainers of the spreadsheets.

I will admit, the reason I give this group primacy is from a purely selfish perspective: the likely maintainers of the spreadsheets you create will be other actuaries ... or yourself one year later. Nothing is more frustrating than wondering “What the heck were they thinking?” when “they” refers to one’s self. Also, I would rather not be on the receiving end of Byzantine messes of Excel files should I ever have the joy of being handed your spreadsheets; perhaps I will have saved myself a lot of trouble later by writing this article. (I highly recommend other people with similar motives also write for CompAct. Nothing motivates like self-interest.)

So let’s jump into it. While I generally have Excel in mind, most of the principles below belong to any spreadsheet setup, as well as computing projects in general.

1. IS A SPREADSHEET APPROPRIATE?

Obviously, this question should be asked before making any spreadsheet, for any use, but we’re thinking from the point of view of something that will need to be maintained—input updated, interfaces changed, techniques reprogrammed, etc. When it’s a one-off use, it’s not as crucial a question ... but so often those one-off, throwaway spreadsheets morph into something more permanent.

In “Spreadsheet Modelling Best Practices,” authors Nick Read and Jonathan Batson give a pro/con chart of using different software. Their paper was written 10 years ago, and Excel has evolved quite a bit since then, but the question of pros and cons can be used if one considers the options. As software features change along with the resources available, in terms of people’s skills, and software and hardware already owned (or budget available for the project), it may be more helpful to make a pro/con list, considering the following dimensions:

- How much might need to be changed in the future? And what would be changing?

This may be difficult to answer, as again, so many times we think we’re doing a one-time task and then come to find that we have to keep doing it every month. The below dimensions may be more or less important depending on the scope of change.

If the underlying structure won’t need to change, but numbers are updated regularly, it may make sense to develop something on a less flexible platform than Excel. If the structure and tasks change a great deal, and one may need to experiment a great deal, it may not make sense to write a program in C++ from scratch.

• Flexibility

The reason for Excel’s popularity is that it’s a general-purpose tool, with so many features, a relatively easy-to-use interface, and the ability to add on more functions as one needs. That said, this can also easily lead to a mess. It may make sense to do experiments in Excel in one’s models, and then put the result that will have to be maintained in a different software form.

• Cost

The cost here should be split into the development and maintenance phases. Too often people think of the development cost as the full cost, but forget the costs of updating and maintaining the project in question. It can be easier to point out the development costs (in terms of software, hardware and personnel) than maintenance in many cases, as one might not know exactly what’s involved until the dreaded maintenance comes.

According to Barry Boehm and Victor R. Baesili, in their article “Software Defect Reduction Top 10 List,” highly-dependable software tends to be more expensive in initially developing than low-dependability software—but that the operational and maintenance costs can easily swamp the “savings.”

• Development Time/Maintenance Time

Excel spreadsheets usually come together more quickly than other choices ... unless there’s a soft-

ware package already set up for the type of task you're trying. The development time is a function not only of the flexibility of the software, but also the years of knowledge most actuaries have with using spreadsheets.

- **Run Time**

There are ways to optimize Excel runs (such as turning off screen updates), but in general, if speed is what's foremost, Excel is suboptimal. I remember writing a Monte Carlo simulation in Excel about five years back ... pretty much, I commandeered a few workstations, set the spreadsheets to running, and then walked away. If I were doing it now, I would use R, given that I had to do a lot of experimentation. There are also software packages on the market, such as Crystal Ball, that are especially set up for this sort of thing.

- **Transparency/Complexity**

The reason I like using Excel compared with proprietary software packages is that the stuff I want to mess with, I can. In some packages, the underlying code is too much of a black box for me. However, not everyone wants or needs this level of control. If one is using a black box-type setup for important calculations, I recommend doing stress-testing to make sure it's giving you correct (or at least reasonable) results.

- **Computational power/optimization for the particular task**

This is a variant of some of the issues above, but the point is this: spreadsheets tend to be a general-purpose tool. There are software packages, such as R and SAS, that were developed specifically for statistical computation; database programs such as Access for slicing/categorizing large amounts of data; actuarial illustration software set up for various insurance products—while many of the same tasks can be done in Excel, they may optimally be done in other software environments.

Of course, you may have a bunch of disparate tasks to do, such as file manipulation, data processing, heavy calculation and visualization—then

the temptation is to do it all within one software setup. I think the better path may be to modularize the task and give each part to the most appropriate software. I have been known to program some file manipulation tasks in Perl, pull the resulting files into a C++ program to do calculations, and then take those results into Excel for graphing purposes.

Fit the tool to the task, rather than trying to keep everything within one file. Now, one may think this is asking for more trouble, but it is easier to debug a modularized setup than a monstrosity where all parts are rammed into the same file. It also makes it easier to split a task for group programming purposes.

2. OWNERSHIP

There needs to be a clear ownership of a spreadsheet, as well as a history of said ownership. In a corporate environment, one always needs to know who to blame (OK, not the best of motives), but more benignly, the maintainer needs to know of whom one can ask questions.

Also important, and considered in the next item, there needs to be clear ownership of the spreadsheet as one may find multiple versions of a spreadsheet flying about when there's no clear, single owner of a spreadsheet.

3. VERSION CONTROL

One should have one sheet of the spreadsheet dedicated to following version control. Also, all previous versions (at least major versions) should be saved (with different filenames, indicating the versions). You never know when you'll have to redo a calculation, using a previous version.

Read and Batson (*Spreadsheet Modelling Best Practices*) propose the following elements of version control and documentation:

The documentation should include:

- a short description of the model's purpose;
- who built the model;

CONTINUED ON PAGE 22

- how to contact the person responsible for the model; and
- the model version number and when it was written.

Depending on the model, other useful items to include on the documentation sheet are:

- details of the data which are currently in the model;
- some brief instructions, describing the layout of the model or how to use it;
- a list of recent changes to the model; and
- a summary of key logical assumptions in the model.

Now, they were writing when Excel 97 was the most recent version on the market, so the complexity of the models they had in mind may be a little lower than what people are using now. I think the general documentation (data sources, purpose of spreadsheet, etc.) as well as version control be on a separate sheet.

The details in a version control entry should be: version number, changes made from previous version, and the

Version	Author	Date	Notes
1.0	MP Campbell	10/1/2006	Original
1.01	MP Campbell	10/15/2006	Fixed VBA code for explanations of results form
1.5	MP Campbell	1/1/2007	Added: mortality improvement, explanations of methods, new Social Security table (not that different from before), printable explanations sheet
1.6	MP Campbell	1/8/2007	Looking at projection scale G
1.7	MP Campbell	1/15/2007	Implemented projection scale G, looking at calculation comparisons
1.8	MP Campbell	1/30/2007	Removed projection scale G stuff, explanation sheet wording edited, cleaning up VBA
1.8.1	MP Campbell	2/15/2007	Fixed text areas on "Printable Explanations" sheet and ExplanationForm user form

person or people who made the changes. Other information can be included, but those are the key items. An example is given in the chart below (left).

The dates in the chart are totally made up, but other than that, these are the kinds of notes I make on Version Control sheets. The particular version numbering is unimportant, but note that I kept moving forward with versions, even when I did something that looked like it undid a feature I had added previously.

Note that sometimes I gave general notes as to what I had been doing in changing a particular version, and sometimes I gave specific details relating to variables or named ranges. It would also be useful to put any runtime bugs discovered in such a version control sheet, which can give direction to fixes that may need to be made for future versions, and can serve as notes if one needs to revert to a previous version.

4. DOCUMENTATION

This is a more general category than version control. I have discussed documentation in the previous article in this series, "The End Users Justify the Means III: The Search for Problems." I will expand a bit more on this topic, as maintainers will have a perspective different from testers and auditors.

One of the key tasks of a maintainer is updating any inputs, and it may be useful to have a Maintenance or Updating Doc sheet, which would indicate which cells would need to be updated within the spreadsheet. Ideally, one would auto-update any information, but the problem often is that the maintainer has no control over the location of the information needed. Once I got a call from across the country, from a user I didn't even know I had, because the spreadsheet was looking for an external file and the file system structure had changed since the last time they updated the spreadsheet.

One thing I have tried, when the updating process was fairly predictable in terms of what needed updating, I made the update sheet a checklist. Something like the chart on page 23.

STEP 1: Check fund parameter sets	
TRUE	1.a Check fund management fees—named range “MERVector”
TRUE	1.b Check fund categories—range “FundClassVector”
TRUE	1.c Check fund margin offset—range “MarginOffsetVector”
STEP 2: Check product parameter sets	
TRUE	2.a Check product GMDB design flag—range “GMDBtype”
	2.b Check partial withdrawal option flag—range “PartialWithdrawaltype”
STEP 3: Populate policy information	
	3.a Clear previous policy information—macro “Policy_Info_Reset”
	3.b Paste in seriatim policy info—check with IA group
	3.c Paste in aggregate product info—check with IA group
	3.d Cross-check seriatim and agg info—ranges “GMDBcheck” and “AccValcheck”
STEP 4: Run Alternative Method	
	4.a Run macro “AltMethodCalculate” - DO NOT TOUCH ANYTHING WHILE RUNNING - runtime ~1 hr
	4.b Check aggregate result - range “TotAltMeth”
	4.c Do reasonability checks - run macro “AltMethReports”

Note that I gave references to the particular named cells that needed to be checked and/or updated, as well as which macros to run.

If there are items within VBA code that would need updating, generally it’s a good idea to keep all constants within a single module so they are easier to check and find.

Another thing I’ve found helpful is to do a guide to named ranges on a documentation page. One can paste a list of named ranges, which gives you the

range names as well as the references. Even though if one names a range well, the range name is its own documentation, it’s a good idea to make notes for the benefit of the maintainer so they know what the various named ranges are used for. Providing notes on which VBA macros will refer to those named ranges is also helpful.

Other things to consider including on documentation sheets: list of macros and their use (can be done within the VBA code itself, but if those are scattered through multiple modules, it becomes unwieldy); list of sheets within the spreadsheet and uses for each sheet; key assumptions made in the models; desired features for future versions. Having these “overview” kinds of documentation helps the maintainer get the big picture of the spreadsheet, and thus their particular learning curve is greatly shortened. Given that you may be the person using this documentation, one year after you last looked at the spreadsheet, think about the kinds of information you would want to know.

Of course, in addition to the big picture, the maintainer may need the “detail” view, in that they need documentation at the level of use/computation. By this, I mean having cell comments indicating what’s within a particular cell (or format conventions indicating an input cell, an intermediate calculation cell, a final result cell, etc.) and having comments within any VBA code itself.

5. SECURITY – DO NOT “PASSWORD PROTECT”

The reason I put the above phrase in sarcasm quotes is that there’s nothing secure about using most spreadsheets. Excel password “protection” is (relatively) easily cracked, and I’ve had to do it before when someone had locked a spreadsheet and subsequently left the company, or, even more annoyingly, the person forgot the password they used.

I have nothing against “locking” spreadsheets against changes, without using a password. This will keep most people who have no business messing with

CONTINUED ON PAGE 24

locked cells and code from doing anything, and the people who know what they're doing are only momentarily annoyed.

However, let us suppose that password protection is actually effective—this greatly complicates maintenance. Given how often people not only leave jobs, but also move around within an organization, if you have a spreadsheet that's run once a year and it's password-protected, the chances are high that the password will be forgotten. And if you have to write down the password somewhere ... that's not very secure, is it?

Again, these are just some general ideas to make the task of maintaining a spreadsheet easier, and I'm sure there are many that could be added to the above list.

If you have practices that help in maintenance of spreadsheets, or other programming packages, consider sharing them with the actuarial community. Too often we are thrown into various software practices as entry-level actuarial students, and good computing practices are picked up piecemeal, if at all.

I can be contacted at marypat.campbell@gmail.com.

REFERENCES:

Banham, Russ. "Up and Away," CFO.com, December 2008 <http://www.cfo.com/article.cfm/12665848>

Boehm, Barry and Basili, Victor R. "Software Defect Reduction Top 10 List," *Software Management*, January 2001. <http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.78.pdf>

Califf, Howard. "Spreadsheets and Specifications", *CompAct*, October 2007, <http://soa.org/library/newsletters/compact/2007/october/csn-0710.pdf>

Campbell, Mary Pat. "The End Users Justify the Means III: The Search for Problems," *CompAct*, April 2009, <http://soa.org/library/newsletters/compact/2009/april/com-2009-iss31.pdf>

European Spreadsheet Risks Interest Group. <http://eusprig.org/>

O'Beirne, Patrick. *Spreadsheet Check and Control*, Systems Publishing, 2005.

Read, Nick and Batson, Jonathan. "Spreadsheet Modelling Best Practices," April 1999 <http://www.eusprig.org/smbp.pdf> ■

In Praise of Approximations

By Carol Marler

When I was taking exams in the '70s, the article, "Analysis of Approximate Valuation Methods," was one of my favorite readings. It was written in 1955, by E. Allen Arnold. I found it both interesting and practical. It began, "Since Frank Shailer's paper 'Approximate Methods of Valuation' appeared in 1924, our actuarial literature has omitted any further development of this subject, except for occasional discussions." Not long after I took that exam, the syllabus was changed and the article was removed. Nothing comparable has replaced it. One purpose of this article is to begin some further discussions of when, how and why we need approximations.

Of course the environment has changed a lot over the years. Our personal computers have power exceeding many mainframes of earlier times. In fact, it has been said that with the computer power available today, approximations are no longer necessary. I disagree. The benefits of increasing computer power have led to significant changes in the way we do our work. Organizational structures are flatter. We no longer have an army of clerks to do routine calculations, and typing pools are an anachronism. We must produce results in compressed time frames, and more analysis is expected. The products we offer have become much more varied, more complex and more individualized, while our valuation methods are also growing more complex, reflecting a range of values rather than a single number result.

Before presenting my arguments for using approximations, it seems worthwhile to define a few terms and to provide some distinctions.

- Estimate/Approximation
 - o An estimate is an educated guess. My dictionary says, "Estimate ... implies a personal judgment" in a specific context.
 - o An approximation is a methodology for getting close enough. Generally this involves a model or formula.
- Accuracy/Precision
 - o Accuracy is a measure of how close one is to the correct answer.
 - o Precision relates to the possible range of results—more significant digits indicate higher precision.

Here are four reasons why approximations are still a very important part of actuarial work.

First, I believe that most companies have at least one

block of business that never grew big enough to justify making system modifications to handle all its unique features. An old term for this category is "shoe box" because all the administrative data was once kept in a box about the size of a shoe box. Even though these cases are probably administered on a computer now, the actuarial analysis is, of necessity, simplified in order to focus on other issues that are more material.

Cost/benefit analysis is always necessary. Good practice calls for putting in the amount of time commensurate with the accuracy that can be added. Experienced actuaries are able to recognize when a judgment call is better than another computer run.

Second, there are a lot of approximations used even in calculations often considered to be "exact." For example, there are two ways to express a person's age as an integer, and both methods are well accepted—age last birthday or age nearest birthday. Unless the calculation is actually done on the person's birthday, though, the integer age is only an approximation. Likewise the use of mean reserves or mid-terminal reserves is well-established. Some companies prefer to use interpolated terminal reserves, but even this is generally done only to the nearest month.

We use a lot of input assumptions that are only approximations. Our mortality tables may look exact, but they always involve some degree of smoothing. Interpolation and/or extrapolation are also necessary because of the sparseness of data, especially at the oldest and youngest ages.

Many companies use early cut-off for administrative systems in order to meet deadlines. Any adjustment to the actual month end-date is a form of approximation. There is often a trade-off between timeliness and accuracy, or a trade-off between the size of the potential error and the cost to make the results more accurate.

Third, the growing use of stochastic models has made it abundantly clear that all our actuarial calculations are merely a point estimate taken from a random distribution. The fact is, we know that the expected value we calculate is almost certain to be wrong, although the law of large numbers does tell us that we can get close enough. How close? A lot of work has gone into analysis of the error involved in various mathematical functions, particularly when these functions are included in a software package. Actuarial judgment is again the correct answer.

Carol A. Marler, FSA, MAAA, is an associate actuary with Employers Reassurance Corporation in Indianapolis. She can be reached at carol.marler@ge.com

CONTINUED ON PAGE 26

On the other side of the closeness question, consider a pension plan with only about five participants. Assuming pre-retirement mortality using any standard table will in most years result in a fractional short-fall in results because actual gains from mortality are less than expected. For this reason, it is common practice to assume zero pre-retirement deaths in small plans.

Fourth, when the underlying data is missing, inaccurate or otherwise flawed, a good enough calculation is really the most efficient choice. Various terms have been used to describe overexertion in such a situation: False precision, spurious precision or illusionary accuracy.

I once heard of an actuary who claimed that he got more accurate results when he ran his model with quarterly payment patterns. The problem was that he hadn't measured actual quarterly premium collections, but simply divided the annual premiums by four. Spurious precision. And because the input data was of low quality, illusionary accuracy.

Another story involves an actuary who presented a rounded result to his manager: about X thousand dollars. The manager wanted it more accurate, so the actuary went back to the computer output and gave an answer to the dollar. When the manager was still dissatisfied, the actuary pulled some change out of his pocket, counted it, and offered that result to provide dollars and cents. False precision. I wasn't there, but I do hope the manager laughed.

There are other times when approximations are valuable.

Checking for reasonableness: This might be for a complex calculation, such as scenario testing. An approximate calculation could show if the results are unreasonable, and may give some insight into where the problem might be.

Stochastic on stochastic: By this phrase, I refer to those cases where each year of each scenario requires an embedded stochastic model. This is a concern with regard to Embedded Value calculations, since one of the items to be projected is the required surplus, which is defined in terms of a conditional tail expectation (CTE), or in other words a stochastic calculation. The number of calculations is a linear function of the square of the product of the number of scenarios and the number of years projected. There are several methods for reducing the computational intensity. One of the most obvious is to replace the CTE with some approximate formula that does not require stochastic projections. Then the formula for time required becomes linear rather than quadratic.

Finally, some comments about incurred but not reported (IBNR) claim liabilities. Whatever you do for this liability, there will be some volatility that cannot be removed.

In other words, nothing will estimate it well. It can be helpful to remember that the objective is to estimate the eventual incurred claims, not the IBNR itself. Thus the error measurement ought to be with respect to the total current estimate of incurred claims.

Of course, you might be in the situation of a company president whose company had only recently begun writing life insurance. With just a few hundred policy holders, the president confidently explained, "I know all of our insured people and they haven't died." Sooner or later, though, there would be a situation in which, through sheer numbers, some death might not be noted in time. A consulting actuary was able to convince the president that he needed to establish a formula-based IBNR while it was small and then allow the provision to grow slowly over the years.

CONSIDERATIONS

Sometimes approximations are necessary, when no better alternative method exists. This is commonly the case when dealing with claim liabilities, including IBNR, as noted above.

Materiality is an important issue. For example, if the aggregate value of approximated item is small, a more complex or detailed approach is not justified. The goal should be substantial accuracy, or in other words, a minimum reasonable error. The method should also be unbiased, or at least have an acceptably small bias. Calculations that can be easily checked are always preferable. Caution should be used when results from one approximate method are used as input to other approximations, to avoid any compounding of errors—the snowball effect. Variations from period to period must also be considered. If a result is too large one time and too small the next, the distortion can have a bad effect on resulting earnings and/or surplus.

Saving time is helpful in meeting deadlines; however, sometimes an approximate method will result in a loss of additional information that was provided by a more detailed approach. This is another trade-off that must be taken into account.

Other issues that must be considered include appropriate utilization of technical personnel, acceptability to auditors if GAAP or to state insurance examiners for statutory, and the value of simplicity. The cost should not be disproportionate to the importance of a particular item.

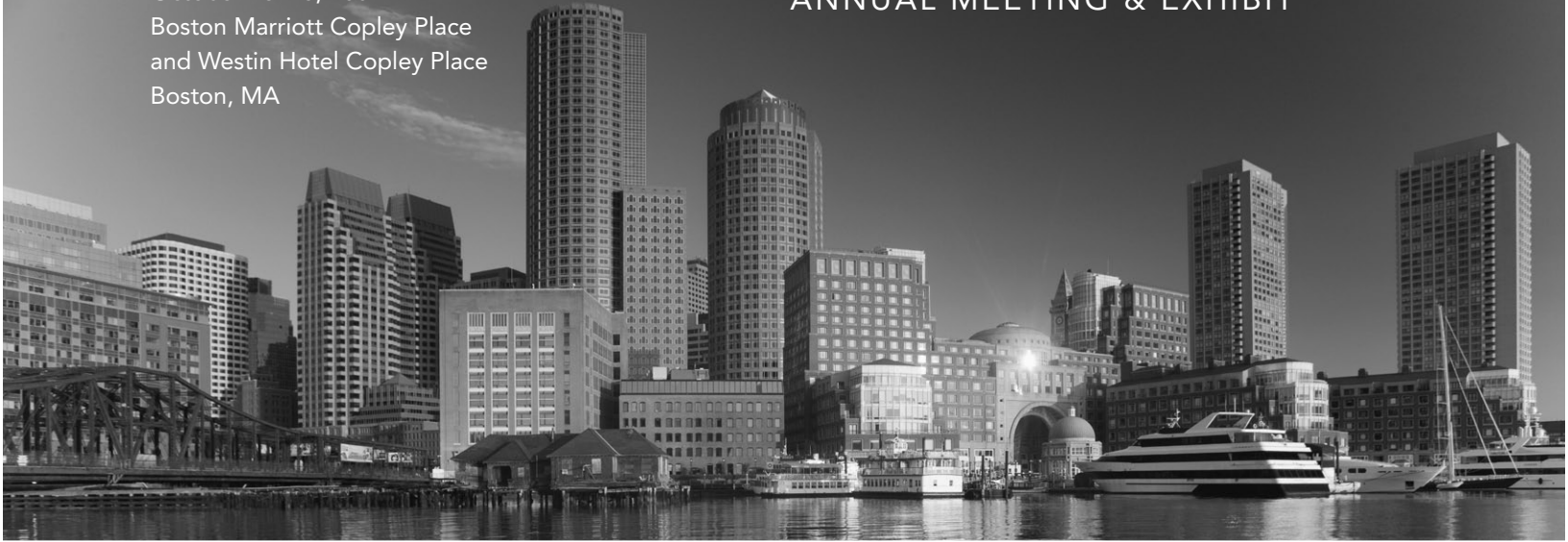
Mr. Arnold ended his paper with this sentence, "Modern business conditions virtually require that the actuary be continually alert to the opportunities for the extension and improvement of approximate methods of valuation." I think this statement is as true today as it was when he wrote it more than 50 years ago. ■



SOA⁰⁹

October 25–28, 2009
Boston Marriott Copley Place
and Westin Hotel Copley Place
Boston, MA

ANNUAL MEETING & EXHIBIT



Visit www.SOAAnnualMeeting.org to learn more about the SOA 09 Annual Meeting & Exhibit, where you can expect fresh ideas, innovative seminars and top-notch speakers, plus plenty of networking opportunities.

BE SURE TO SIGN UP FOR THESE INFORMATIVE SESSIONS:

Session 1

TECHNOLOGY SECTION WINE AND CHEESE
RECEPTION

Come and enjoy the company of your fellow members of the Technology Section. This will be an informal networking opportunity for Technology Section members and those desiring to see what the Technology Section has to offer. This reception is open to all meeting attendees.

Session 35 - Panel Discussion

ACTUARIAL AND IT DEPARTMENTS: MAKING THE
MARRIAGE WORK

Are there barriers between the actuarial and IT cultures that keep you from collaborating on technology issues? In this session, you will see recent survey results that reveal the current state of the relationship between actuarial and IT departments.

Actuaries
Risk is Opportunity.®

CompAct

475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
p: 847.706.3500 f: 847.706.3599
w: www.soa.org