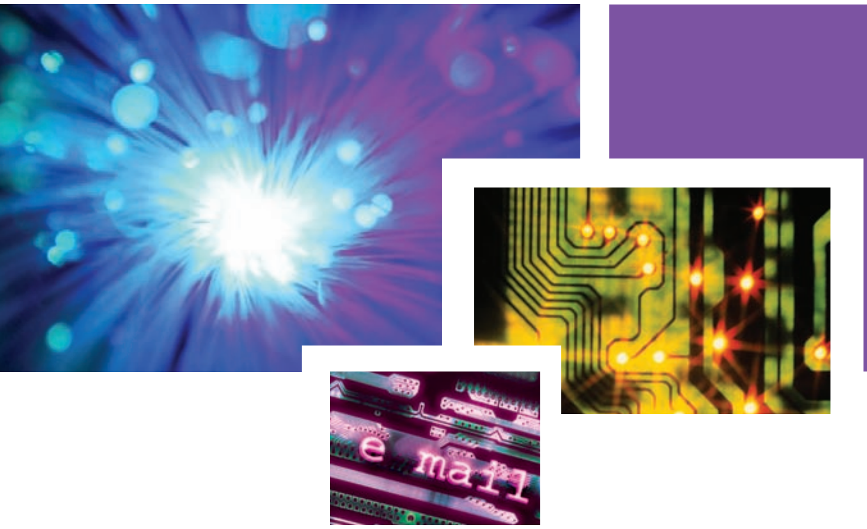
TECHNOLOGY SECTION

"A KNOWLEDGE COMMUNITY FOR THE SOCIETY OF ACTUARIES"

CompAct

Issue No. 29 • October 2008



Inside

Letter from the Editor

by Howard Callif

It's hard to believe this issue completes my first year as editor. There have been many people stepping up and contributing to help make this an informative publication. Thanks to all of the authors! A print edition will be distributed at the annual SOA meeting in October 2008, which will showcase some of the articles published in the last year, in addition to the new content for our October issue. If you are attending the meeting, be sure to come to the section reception (002M) and get information on where you can get a copy. If you are reading the print edition, be sure to visit the SOA Web site to see the other newsletter editions.

Steven Craighead has offered to publish a regular column on the R programming language. This is a popular modeling language, and has been discussed in the education section newsletters. Shyamal Kumar published an article titled, Using R for Actuarial Science, some time ago, which is posted in the technology section newsletter link on the SOA Web site (*soa.org/news-and-publications/newsletters/technology/compact-tech-section-details.aspx*).

Howard Callif is a senior system architect at iPipeline in the Illustrations unit. He can be reached at howard@callif.org.

Matthew Wilson will continue his Cool Tech features, and keep everyone informed on the latest tools and utilities and will give valuable details on how to best use the technology. It would be wonderful to have a column for Excel and VB, as well as any other common technology (such as pricing tools, etc). Please volunteer to share your expertise!

A special thanks to Mary Pat Campbell, who has written many informative articles in the last year. Please consider following her lead, and submit your own article!

Please keep the comments and articles coming! If there are any issues or questions you have, feel free to e-mail me, or seek me out at the SOA meeting by leaving a note on the message board. I'd love to hear any ideas you have!

Howard Callif

Letter from the Chair

by Kevin Pledge and Tim Pauza

Just over three years ago we set out to revitalize the CompAct newsletter. Under the guidance of my predecessors Phil Gold and Paula Hodges, with editorial leadership from Shyamal Kumar, CompAct became a regular quarterly newsletter. We are proud to have continued this effort this year, under the editorial leadership of Howard Callif. Examples of some of our articles from the year are included in this publication. (Past editions of CompAct can be found online at *http://www.soa.org/news-and-publications/newsletters/technology/compact-details.aspx*
.

The section has been busy on a number of initiatives, these include:

- Rate Manager: Many of you have found the Rate Manager, originally developed by Steve Strommen, a popular and useful tool. Responding to concerns regarding the accuracy of the rates, Joseph Liuzzo has been leading the initiative to review and update rates.

- Scenario Manager: The Scenario Manager will provide a tool to store and exchange economic scenarios using a standard format. The tool has been developed, tested and documented, and will be formally released soon.

As Technology Council enters a new year, the position of Chair will transition to Tim Pauza. Tim has been very active in supporting me in this role over the last year and I am sure he will do great. Tim Deitz will take over as Vice-Chair and Joe Liuzzo will continue as Secretary and Treasurer. Your council continues to meet monthly via teleconference. If you have something for us to consider as we evolve our section, please let us know.

Kevin Pledge (Chairperson)
kpledge@insightdecision.com

Tim Pauza (incoming Chairperson)
Tim.Pauza@ey.com

Kevin Pledge, FIA, FSA, is president and CEO of Insight Decision Solutions in Markham , Ontario. He can be contacted at kpledge@insightdecision.com.

Tim Pauza is a manager in Ernst & Young LLP's Insurance and Actuarial Advisory Services practice, and is based in Philadelphia , Pennsylvania. He can be reached at 215.448.5836 or tim.pauza@ey.com.

Where are the Actuaries?

by Tia Goss Sawhney

There are good arguments as to why actuaries should have an integral role within nearly every IT and business intelligence (BI) company serving the insurance industry:

1. **Data.** Actuaries have always been the ultimate power users of insurance company data. As a result we understand the supply chain of insurance data: from what gets/should get captured, to how it gets consolidated and sometimes corrupted along the way, to how it ultimately gets used to make critical decisions.

   a. We understand the peculiarities of insurance data in ways that people outside the industry do not understand the data.

   b. Our perspective is broader and often deeper than any of the other insurance industry business users.

   c. We focus more on the goals, than the process. IT people, even those with insurance experience, often get the mechanics of data management and lose sight of business decisions enabled by the data.

2. **Analysis.** Actuaries have long been pioneers in the analysis of insurance data. New business intelligence tools are being designed for the analysis of insurance data. These tools will be stronger if they incorporate what actuaries already know about insurance data analysis.

3. **Credibility.** Actuaries have incredible credibility within the insurance industry. An IT/BI team that includes actuaries will be perceived by insurance company customers as stronger than a team without actuaries.

4. **Minimal skill gap.** Of all the business users in an insurance company, actuaries are the most technical. It is easy for us to learn both the technical skills and clear, linear thought processes, and communication skills required in the IT/BI world.

While these arguments might sound good to us actuaries, they are, unfortunately, ineffective. I know this from personal experience. Being a gregarious person, not interested in working on the nth actuarial valuation, over the years I have initiated conversations with several IT and BI companies serving the insurance industry. Each time I have been told that there is no role for actuarial skills in their work.

Although the lack of interest from the IT/BI companies that I experienced was with respect to both technical software development roles and soft skill roles such as selling software products to actuaries and other insurance industry business users, for the remainder of the article I will focus on actuaries in development roles.

Perhaps a better salesperson than me could have made more progress in these discussions. But it does not look like many actuaries actually have. Consider the following:

- *Of the 20,106 members on the SOA roster as of July 2008, only 194 declared their "employment type" to be "software developer/vendor"—less than 1 percent.*

- *The 194 is admittedly understated somewhat. There are actuaries working primar-*

ily on software development for consulting firms (example: Milliman) that list themselves as consulting actuary. Furthermore, there are some actuaries employed by software companies who list their employment as consulting actuary (or less frequently one of the other options)—many of these actuaries, however, are probably more involved in sales, implementation, or management rather than in software development.

- *Most companies that employ multiple actuaries in development roles are companies focused exclusively on actuarial software (examples: PolySystems, Actuarial Resources Corporation, GGY AXIS, and Winklevoss Technologies).*

- *Almost every large IT/BI company serving multiple industries has an insurance vertical.*

- *The following are the actuaries employed in development roles by some of the largest multi-industry IT/BI companies:*

  o *CSC*—17 (give them a gold star!)*
  o *Accenture*—4*
  o *SAS—2*
  o *Fiserv*—2*
  o *IBM (including Cognos)*, Microsoft, Oracle, SAP (including BusinessObjects), HCL, Symantec, Computer Associates, EDS*, CapGemini, Adobe, First Data, Infosys, Wipro, and NCR/Teradata—0*
  * *In addition there are a handful of actuaries (in total) employed by these companies in roles other than development.*

Get the picture? IT/BI companies very occasionally employ actuaries. As a rule, we are not part of their insurance vertical teams. Do we care? We should care. A lot.

The reasons why we should care parallel the reasons that we should be part of the IT/BI teams:

1. **Data.** Every form of actuarial work depends on data. We need to ensure that the data that we need is captured, properly managed, and available to us when and how we need it. We need to partner with IT/BI teams to ensure that this will happen, especially since they are actively working at replacing the legacy systems upon which we have long relied.

2. **Analysis.** Actuaries are skilled at analysis. But we are not the only people with analytic skills. IT/BI companies commonly employ statisticians, accountants and other analytic professionals. These professionals are using the tools of their professions to build analysis algorithms, algorithms that claim to be better than ours, and sometimes really are. Furthermore their algorithms are getting better all the time. Actuaries are posed to be displaced as the preferred partners for insurance data analysis.

3. **Credibility.** If we lose our strong role in insurance company data and analysis, we will also lose our highly credible position within insurance companies—a position that has propelled many of us into management and leadership roles outside of purely actuarial functions. Although actuaries will still be needed for numerous statutory functions, the actuarial function will become smaller and more marginalized.

(continued on page 6)

Where are the Actuaries? • continued from page 5

IT/BI companies very occasionally employ actuaries. As a rule, we are not part of their insurance vertical teams. Do we care? We should care. A lot.

4. **Less Relevant Skills.** Is there any doubt today that technology has an integral role in every business endeavor? That technology is at the forefront of nearly every business transformation? If we want to truly be part of the action, and not a marginal end user, we need strong IT/BI skills.

So why aren't we deeply integrated into insurance industry IT/BI software development? I have my theories. Topping the list is the actuarial exam syllabus.

The preliminary actuarial exams do not test us on everything that we might need to know for insurance risk analysis. Instead they cover the essential core topics. The syllabus does not cover data capture and management in any way, not even the rudiments of database design. The syllabus also does not cover the predictive modeling tools that long ago proved their value for risk analysis in other industries, but which are only now being brought into life and health companies–mainly by statisticians working for IT/BI companies.

We work hard to pass actuarial exams. The exams consume much of the available time for learning. We trust that the exams give us the skills we need for our job. Although the exam syllabus is falling short of this goal, part of the responsibility lies beyond the syllabus. We are so cocky about having passed actuarial exams, that when we are done, many of us feel that our days of serious learning and test taking are over. In contrast, continuous learning and additional certification are the norm in the IT/BI industry. We don't accept that an older IT person should have no knowledge of Internet-based tools; but we accept that older actuaries will have no knowledge of predictive modeling tools.

There are, however, probably other issues at work. Although actuaries with outgoing personalities have a long tradition of moving to non-actuarial roles within insurance companies, less outgoing, more technical actuaries, the ones perhaps most suited for IT/BI work, have a tradition of staying put. This does not need to be the tradition. What if actuaries in training had to rotate through in-house IT/BI teams, including on-site vendor IT/BI teams?

Price and teamwork are other issues. Actuaries traditionally command salaries that are higher than many IT workers. This hurdle is not, however, insurmountable. Price works out if we demonstrate value. Another hurdle is team work. IT/BI work is almost always team-based, much more so than actuarial work. Working in team environments may be another skill that we need to add to our skill acquisition list, perhaps by restructuring actuarial departments or the solo-approach to exam taking.

At the beginning of the article I told you that I have never been successful in convincing an IT or BI company to employ me. I have worked only at the edges of these fields. Therefore, I have prepared this article with the intent of stimulating your thinking and discussion, rather than giving definitive answers. I look forward to the discussion.

Tia Goss Sawhney, FSA, a health insurance actuary, is the owner of TS Strategic Consulting and can be reached at tia@tsstrategic.com. *She is joining a public health doctoral program in the fall 2008 where she will learn more about health care data and analysis–from the perspective of statisticians, epidemiologists, and other professionals.*

How to Reinvigorate Your Career in Technology

by John West Hadley

If I want to have an outstanding career path in technology, then obviously I should work hard to keep myself up-to-date in all the latest technology areas, shouldn't I?

Not necessarily.

The absolutely most important skill you can work on to reinvigorate your career is your ability to express your accomplishments in a concise, compelling, results-oriented manner. I'm not saying that technology skills aren't important, but they will only get you so far. It's your ability to communicate what you offer that will make you stand head and shoulders above others.

Case in point: I finished out my traditional actuarial career as Chief Actuary, a role to which I never really aspired. I knew I was ready for a change, and set out to carve out a systems role for myself. My first step wasn't to secure my PMP, or to take a course in JAVA, or to become an expert in Web design and search engine optimization, or to seek any of the myriad of systems certifications. I set out to:

1. **Examine my past accomplishments and roles, and get really clear on what I had done, what challenges I had overcome, and the results I had achieved** for my employers. (Naturally, systems-related work was particularly critical to emphasize.)
2. **Turn that into a compelling marketing message** I could present that would get conversations started.
3. Sit down (by phone or in person) with everyone I used to work with, knew from SOA and actuarial club meetings, had socialized with, etc**. to make sure they had a clear picture of what I could do for my target employer.**

Within six months, I had a systems consulting practice well into six figures.

Here's the difficulty:

Most people, especially very technically-oriented professionals like actuaries and systems professionals, find it very difficult to market themselves. In fact, they often think of marketing as a dirty word, something they shouldn't have to do. On top of that, our profession tends to attract relatively introverted people, so that effective networking generally doesn't come naturally.

So how do you develop this ability? Look at your achievements, and ask yourself these questions about each one:

- Why did I or why was I asked to take on this project?
  - o Why me and not someone else?
  - o Why was it even necessary for ANYONE to take it on?
  - o What were the expected benefits, or the significant problems to be solved?
- What challenges did I have to overcome along the way?
- What actions did I take to overcome those?

(continued on page 8)

- What specifically did I do to bring the project to its conclusion?

- What was the result of my efforts?
  - o Did I complete the project more quickly than usual, freeing up resources for other critical tasks?
  - o Was I able to take a project that was behind schedule, and put it back on target?
  - o Are some of the techniques I employed going to serve as templates that will improve efficiency on future projects?
  - o Will successful completion of the project lead to increased productivity, lower expenses, higher profitability, etc.?

Watch out for this trap: While you want to get really clear in your own mind about the actions you took, you don't necessarily want to relate all of those to others unless they specifically ask. What will really grab people's attention are the challenges solved and the results achieved. Actions taken should be left at the high level, like this:

"I took three primary steps to put the project back on track. First, I interviewed every key player to uncover any obstacles in the critical path. Second, I negotiated with senior management to prioritize the project into Day 1 vs. Day 2 objectives. Third, I instituted a new incentive program that re-energized the team members and ensured that they all gave 110 percent effort."

Your goal is to keep the listener engaged, wanting to know more, not to tell them all the details of what you did. When they show sufficient interest to ask how, for example, you developed and implemented the incentive program, then you can go into more detail.

Now let's look at a concrete example of how to do this, in the context of a résumé, where you need to boil the entire story down to concise, compelling bullet points.

Here's the information "Jim" and I started with:
"Responsible for all data discrepancies resolution in equity research universe and communication with individual analysts. Created and propagated efficient communication, reporting and process changing network between sell-side analysts, research data vendors, and institutional buy-side/sell-side clients aimed at resolution of data issues."

One quick observation: Never use words like responsible for. This is very passive and boring, and it's totally unnecessary. The first sentence is substantially improved with just this simple revision:

"Resolved all data discrepancies in equity research universe and communicated results to individual analysts."

I interviewed Jim on what he had accomplished in this job, using some of the questions I outlined above. Here are just two of the resulting bullets we came up with:

- Executed account management and customer service functions for 20 percent of the world's sell-side analysts in 15 research houses internationally, contributing to 100 percent renewal rate and four-fold increase in sales over two years.

- Effected improvements in delivery of financial information from brokers to data vendors by instituting data quality control routines. This reduced the error rate from 10- to 20-percent to below 1 percent.

By the way, once Jim started presenting his accomplishments effectively, he secured a job offer that increased his salary 30 percent, which his current company then agreed to match. ■

John West Hadley is a career search counselor. He can be reached at John@JHACareers.com *or 908.725.2437. His free Career Tips newsletter and other career resources are available at* www.JHACareers.com.

The End Users Justify the Means
Part 1: The General Public as End User

by Mary Pat Campbell

"Make something idiot-proof, and they will build a better idiot."

I first realized I had an end user problem when I got a call from Denver, complaining that my annuity payment calculator spreadsheet wasn't working any more. As I listened, I thought: who is this person, and how did my spreadsheet end up in Denver?

Many times, spreadsheets we create for one purpose and one audience end up getting re-purposed and re-audienced. In trying to make user-friendly spreadsheets, we can end up with very unfriendly users. In previous articles (To Err is Human; To Correct, Divine and Spreadsheet Check and Control: A Review) I covered the issues of preventing and correcting errors in your spreadsheets; here, I assume you have that part nailed, and are now readying to hand over your work of Excel art over to a bunch of people who will ruin it … or at the very least, will put your spreadsheets through very hard use.

My antagonism with end users is a bit over-the-top, but often spreadsheet makers and end users seem to be in a war. The makers work very hard to get the moving bits working correctly, and the end users complain about the interfaces and that it doesn't deliver what they need. While it is very important to have the technical aspects of one's spreadsheet work, it is even more important that we take user needs into consideration.

Of course, not all users are created equal, and some people, such as those who have to maintain the spreadsheet (which could very well be us, the spreadsheet creators), don't seem like users at all. Users will have different needs depending on what they want out of the spreadsheets, and they will have different capabilities. Here I start a series of articles on some of the major types of end users, and some general principles and specific techniques in dealing with them.

My intended schedule of articles:
1. The General Public
2. Numerate decision-makers
3. Auditors and testers
4. Maintainers

These articles are not intended to be exhaustive, and are based on my own direct experience as well as research into this area. At the end of this article are several references I've used, touching on different aspects of this subject, whether it's the visual design aspect or a systems perspective. I highly recommend other people write articles on this subject as well, or e-mail me at *marypat.campbell@gmail.com* with your ideas, so I can follow up on these articles.

First up: the toughest set of end users—the General Public a.k.a. the Great Unwashed Innumerate.

Granted, actuaries are not often called upon to write spreadsheets for use by the general public, though one may come close by being asked to make spreadsheet tools for the marketing or sales departments (OK, OK, many of them are numerate … now.) This is not something to be proud about. How many times have you heard outsiders describe actuaries as subject matter experts, but horrible at non-technical communication? This is our own fault.

Actuaries are experts on many topics related to risk, and many people are poorly educated in this realm—we should be out there educating people on the various risks they face and how to manage or ameliorate them. Spreadsheets are but one tool we can use in this goal of communicating with and educating the public. As an example, I created a spreadsheet (which can be found on the SOA Web site at: *http://soa.org/research/pension/research-simple-life-calculator.aspx.* Editor's note: you must enable macros to use the calculator) to give people a better feel for longevity in retirement when most life expectancy numbers are quoted from birth (and you're only given expectancies, as opposed to percentiles as well). As of this writing, the first version of the spreadsheet is up, but I am assuming my second version is up by time of publication, which has some features described in the items below.

A. Is a spreadsheet appropriate?

Spreadsheets may be our general utility, but they're not always the best tool for the job. I like to think of Excel as a fancy calculator, allowing me to try different inputs to get a feel for a situation, or to let me enter information specific to my needs so as not to be distracted by irrelevant information.

If there is no interactivity, but a simple static display of information, then there's no need for a spreadsheet. This is especially true if you don't want anybody tampering with anything in the spreadsheet. Copy and paste into PowerPoint or Word if you just want to show a table of results, or a graph.

Tip: if you are copying a graph, make sure when you paste that you go into the *Edit > Paste Special* menu in Word or PowerPoint, and paste the graph as a picture as opposed to an Excel object. I learned the hard way that when you blithely CTRL-C CTRL-V any graphic from Excel into Word, the entire spreadsheet is embedded within the Word document—for each graph you copy over. I crashed a few people's mailboxes by sending them my overloaded Word documents before I wised up.

If you really don't want people tampering with your information, it will be even better to get it into PDF format. Too often a graph in PowerPoint or Word can be moved to where its context is destroyed, and tables of numbers can be inadvertently—or purposefully—altered.

Another situation where a spreadsheet is not appropriate for a general audience is when you have sensitive or confidential information inside the spreadsheet. You may hide the sheets, lock the workbook, and password protect, but the security features of Microsoft Office are not totally secure. I have cracked many spreadsheet passwords with free add-ins you can find on the Internet. There is nothing secure about any Excel workbook. If you must use secret information to do your calculations, use another method, such as a Web interface to a secure server.

Tip: If you want to make a spreadsheet for general use where you strip out sensitive information from another spreadsheet, it's best to start out with an empty spreadsheet and copy over the parts you want rather than take the original spreadsheet and delete the sensitive parts. I'm probably being overly cautious here, but I can never be sure how much information Excel holds of previous versions of a spreadsheet; obviously it holds some information, or there would be no "Undo" functionality. In the life expectancy calculator, the

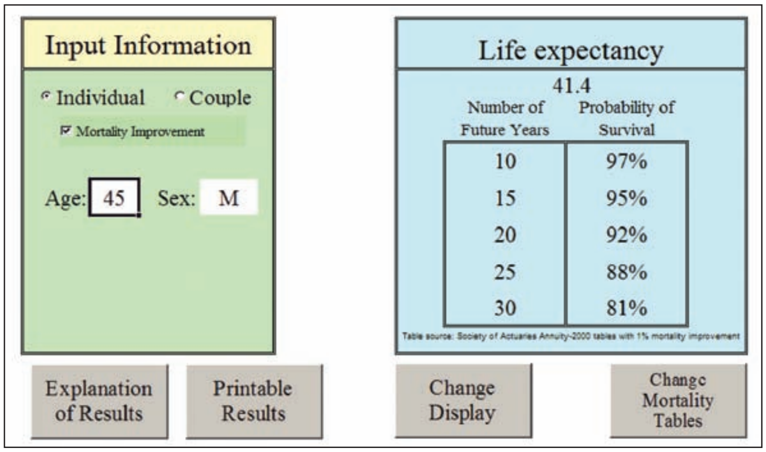
(continued on page 12)

original version was for use within TIAA-CREF and had some TIAA-CREF-specific mortality assumptions. In creating a version for the SOA, I started with an empty spreadsheet and copied over only a few of the mortality tables and nothing specific to TIAA-CREF.

B. Good interface design: think iPod

Apple has been the design leader of the 21st century, in my opinion (yes, I'm a Mac person, but I promise I'm not insufferable about it—much.). Where they have excelled is in clean, simplified interfaces, where a new user can pick up an object (such as an iPod), or open up an application (such as iPhoto), and get right to using it without having to muck about with user manuals … and they're pretty, to boot!

When I started to design the interface for life expectancy calculator, I had Apple's design in mind. Here's a screen shot of the front page:



Input Information

● Individual ○ Couple

☑ Mortality Improvement

Age: 45 Sex: M

Life expectancy

41.4

Number of Future Years	Probability of Survival
10	97%
15	95%
20	92%
25	88%
30	81%

Table source: Society of Actuaries Annuity-2000 tables with 1% mortality improvement

It's not the sleekest thing around, but it's very clear—large fonts for easy reading (except for the info in itty-bitty type, which is not going to be important to most general public users), clearly defined and separate input and output areas. There's no paper clip yabbering at you, asking what you'd like to do. There's a box for you to type in, buttons to push and an option to click on and off. Every button and option is clearly labeled with the least amount of words. (I do think Input Information is clunky, though—I'd be happy to hear suggestions for my next version.)

Here are some general principles or issues I've considered in designing user interfaces for the general public:

i. Visual structure: Having well-defined areas on the page for various functions, whether input, output, explanatory or other. Also, always turn off the grid and row and column labels—only designers/maintainers will want to look at that.

ii. Uniformity of style: You don't want different elements clashing with each other. To go back to the Apple example, the iPod aesthetic is immediately recognizable, though it has expanded from the original model to include the Shuffle, the Nano, and the Touch—all with very different capabilities, but a commonality of design.

iii. Readability: Large fonts, dark text on light backgrounds, and space around different elements help improve readability of an interface. Edit wording down as much as possible—the fewer words used, the easier it is to read and the more likely it will be read. As Joel Spolsky of Joel on Software proclaims: "In fact, users don't read anything." A bit of hyperbole, but a good warning to prolix designers.

iv. Color: Excel has a limited default palette, which is why my spreadsheets tend to end up looking like an Easter window at Macy's. Color should be used with a specific purpose in mind. Here, I use color to set different regions apart in the spreadsheet. I used black text on a yellow background, as it provides one of the most eye-catching contrasts for reading (why do you think warning signs are yellow and black?), so that it would grab visual focus when this screen is first presented.

v. Font: Some fonts are easier to read than others. They all become less readable if you mix them up or lard on too many features, such as underlining, italics, boldface, all caps, colors other than black, and so forth. Generally, I use just one font—Times New Roman, a serifed font, at a sufficiently large size. Sometimes I use the sans serif Arial (the default for Excel most of the time) if I need to write something very small; serifed fonts do not work well at small size as the text-to-space ratio approaches illegibility.

C. Hide what isn't needed

I hide a lot of things in my spreadsheet. First, I turn off sheet tabs, the grid, and row and column headers. I hide input fields when you click on particular options and hide ridiculous results. For example, I do not show probabilities for survival beyond age 95, because the mortality tables beyond 95 are unreliable. But the end user doesn't need to know that. If you click between Individual and Couple, you will note that the age and sex input areas disappear and reappear. I do this by using VBA to set the background and font colors of the pertinent cells to the background color for the area.

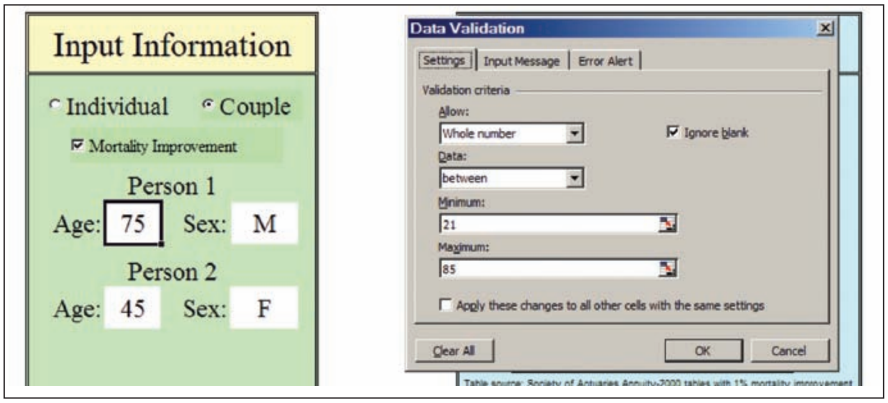
If you want to see what lies behind the calculation, feel free to unlock the spreadsheet (I will talk about that below) and see what I have floating around. Calculating life expectancy is pretty simple, so there's not much to it. You will see some features I will talk about in later articles, which should be of use to auditors, testers and maintainers, but is of no interest to my target audience.

Hiding what the user doesn't need or want to see provides for a cleaner interface and reduces confusion. Also, in hiding the tabs and using buttons to force the user to navigate between two possible pages, I am controlling how the user navigates their way through the tool. I have done this many times for multi-paged spreadsheets, to make sure that all input pages are gone through before a final calculation.

D. Constrain inputs and have sensible defaults

As a designer, I plan on whatever can be entered wrong, will be. You should have Data Validation on every input cell. Sure, the input prompt says Age, but just watch—someone will type in Forty-two in that field instead of 42.

If you've never used Data Validation before, let me give you a quick walkthrough. I use two types of Data Validation in my spreadsheet. The first one relates to entering the age. To check this out, you need to unprotect the sheet (Go to Tools > Protection > Unprotect Sheet ... I don't use passwords). Highlight one of the age entry areas and pull down Data > Validation. You should see this:



Input Information

○ Individual ● Couple

☑ Mortality Improvement

Person 1

Age: 75 Sex: M

Person 2

Age: 45 Sex: F

Data Validation

Settings | Input Message | Error Alert

Validation criteria

Allow: Whole number

☑ Ignore blank

Data: between

Minimum: 21

Maximum: 85

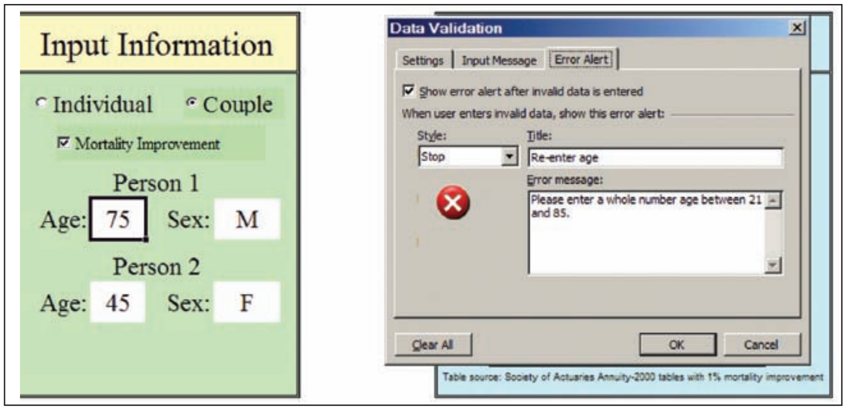
☐ Apply these changes to all other cells with the same settings

Table source: Society of Actuaries Annuity-2000 tables with 1% mortality improvement

I restrict entry of age to whole numbers between 21 and 85. There are some reasons for this—first, the point is to help people gauge retirement longevity or longevity from their current age. This is not intended for children; that explains the lower bound. The upper bound is due to the scarcity of data at high ages and lack of reliability in the extreme ages. Ideally, I expect the tool to be used for people age 45-70, but I expanded it to a more reasonable range.

(continued on page 14)

When someone tries to enter something that does not fulfill the validation criteria, they get an error message. There is a generic one, but you should never allow that. You should always customize the error message to let the user know what you are looking for in that field:



Input Information

○ Individual ● Couple

☑ Mortality Improvement

Person 1

Age: 75 Sex: M

Person 2

Age: 45 Sex: F

☑ Show error alert after invalid data is entered

When user enters invalid data, show this error alert:

Style: Stop

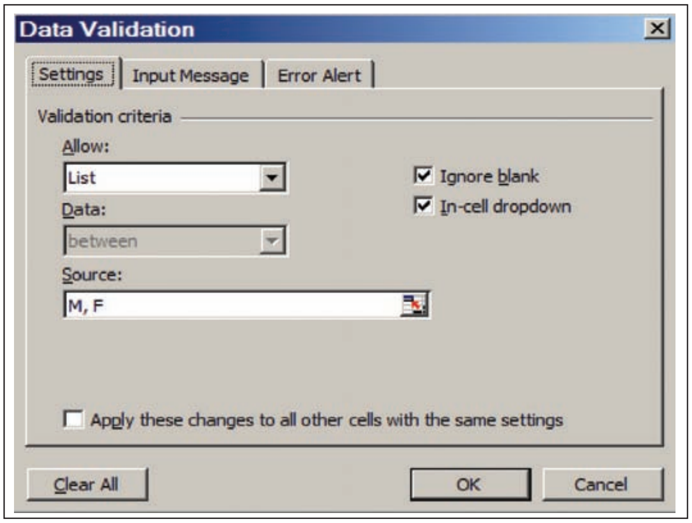
Title: Re-enter age

Error message: Please enter a whole number age between 21 and 85.

Table source: Society of Actuaries Annuity-2000 tables with 1% mortality improvement

Notice there is a third tab, which calls for an Input Message. I usually keep this empty, so as not to annoy knowledgeable users. I figure I should correct the users only if they have erred. You might think an Input Message would be appropriate when the input prompt does not supply enough information as to how you want the cell filled. I say that's a failure of design and the interface should be redesigned.

The other Data Validation I use is for sex:



Validation criteria

Allow: List

Data: between

Source: M, F

☑ Ignore blank

☑ In-cell dropdown

□ Apply these changes to all other cells with the same settings

The List criterion is a very useful option in Data Validation—if you have a small list of options to choose between. You can enter the list (as I did here), or refer to a range of cells (if you use cells from a different sheet, it must be a named range). The most useful feature of this option is the In-cell dropdown list, which automatically pops up when the cell is selected. The user will then automatically pick one of the items on the list. Note that I do not take this for granted—I still specify an error message, just in case the user is unusually creative.

While Data Validation is crucial to ensure that all inputs are such that calculation errors won't occur, another item to add is reasonable defaults to all input cells. I did not actually do this with my life expectancy calculator spreadsheet as the inputs are so simple. The reason to populate input cells with default values is to give the end user an idea of the range of values one is expecting to be entered. As well, many times people will not want to change defaults if they are not knowledgeable about the item entered (such as an expected interest rate, as an example).

E. Information at the point of need, only to the extent needed

Notice that I keep the explanation of what these numbers mean tucked away. I expect non-experts to need lengthy explanations and concrete examples to interpret the numerical results of my life expectancy calculator. However, I want to keep a clean interface and don't want to bother them with an explanation when they don't need it. There's a reason people got annoyed at Mr. Paperclip.

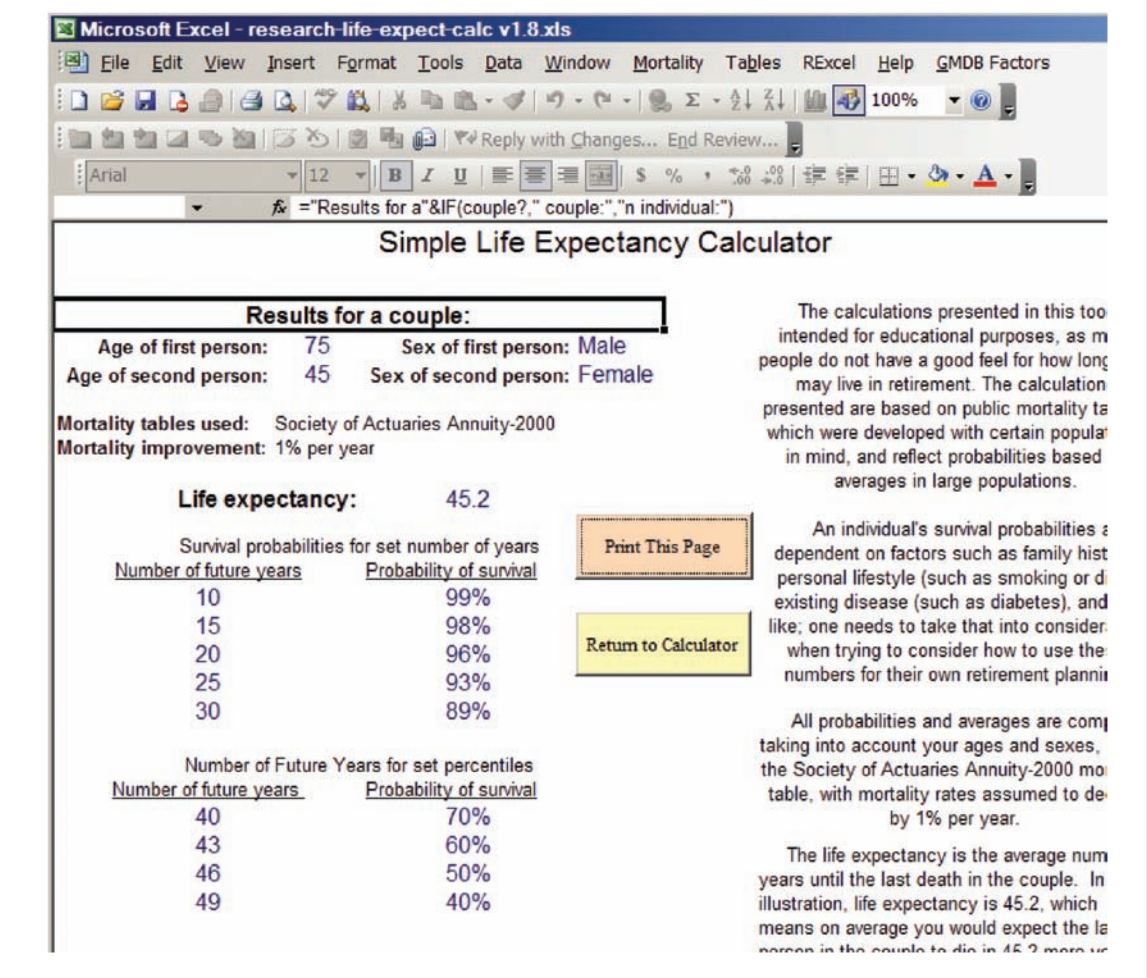
On the input page, I have a button simply labeled Explanation of Results. If you click that, an explanation of the numbers in terms of life expectancies and survival probabilities comes up—and it's directly related to what is currently in the input fields. Once one has read the explanation once, that will probably be enough for many users, and they can go back to playing with the calculator without any distracting explanations.

Note, also, that I don't go deeply into the assumptions behind the mortality tables used (that comes later). When people are playing around with the simple interface, they just need to be able to interpret what the numbers mean. People generally don't like reading a bunch of text off a computer screen, so I try to keep the explanation as succinct as possible.

F. Printable results

The clean interface and simple explanations are all very good, but they don't print out well. A computer screen is very different from a piece of paper. The empty space and large fonts that help with interactivity become low-density information when printed out. Also, the pop-up boxes with explanations do not appear in a print out.

It is a good idea to have a printable results page that has all inputs and outputs, as well as more detailed explanations than were on the input page. Sometimes your interface design will lend itself well to both computer screen and paper, but it's unlikely. The design concerns with a printable page are somewhat different from the computer interface—you will still want to have well-defined regions, carefully pick a font, and make sure the overall design is consistent. However, you will not


Microsoft Excel - research-life-expect-calc v1.8.xls
File Edit View Insert Format Tools Data Window Mortality Tables RExcel Help GMDB Factors
100%
Reply with Changes... End Review...
Arial
12
Print This Page
Return to Calculator

fx ="Results for a"&IF(couple?," couple:","n individual:")

Simple Life Expectancy Calculator

Results for a couple:

Age of first person: 75 Sex of first person: Male
Age of second person: 45 Sex of second person: Female

Mortality tables used: Society of Actuaries Annuity-2000
Mortality improvement: 1% per year

Life expectancy: 45.2

Survival probabilities for set number of years

Number of future years	Probability of survival
10	99%
15	98%
20	96%
25	93%
30	89%

Number of Future Years for set percentiles

Number of future years	Probability of survival
40	70%
43	60%
46	50%
49	40%

The calculations presented in this too
intended for educational purposes, as m
people do not have a good feel for how long
may live in retirement. The calculation
presented are based on public mortality ta
which were developed with certain popula
in mind, and reflect probabilities based
averages in large populations.

An individual's survival probabilities a
dependent on factors such as family hist
personal lifestyle (such as smoking or d
existing disease (such as diabetes), and
like; one needs to take that into consider
when trying to consider how to use the
numbers for their own retirement plannin

All probabilities and averages are comp
taking into account your ages and sexes,
the Society of Actuaries Annuity-2000 mo
table, with mortality rates assumed to de
by 1% per year.

The life expectancy is the average num
years until the last death in the couple. In
illustration, life expectancy is 45.2, which
means on average you would expect the la
person in the couple to die in 45.2 more ye

(continued on page 16)

want to be using much color, as you can't count on your user having a color printer.

On a printable page, one can include far more results, and wordier explanations. The idea is that a person will print out the results and will be able to contemplate them at their leisure. Note still that I know the important part for most people will be the numerical results, so I used larger fonts for these items.

Users should be able to immediately focus on the meat of the page in looking at the printed product. People (who read English, at least) focus on the top left before anything else—it's a habit from decades of reading. So that's where I put the prime information.

G. Bulletproof your code

Simply put: the user should never see an error message, and you should never allow a user to do something that breaks the spreadsheet.

This is difficult to do, as you may think you have covered every possible screw-up, only to find that the user has found a new way to screw up. Hooray.

I expect non-experts to need lengthy explanations and concrete examples to interpret the numerical results of my life expectancy calculator.

As noted above, one method of bulletproofing your spreadsheet is to add Data Validation to input cells. Another thing to consider (which I did not do in my own spreadsheet) is to add catch-all error handling in your VBA (get to know On Error) and in Excel itself (get to know ISNA and ISERR).

A quick tip: make sure the spreadsheet recalculates after any change in inputs. My default for my Excel is manual calculation (as I'm often dealing with large spreadsheets or volatile functions), and your user may also have such an option chosen when they open your spreadsheet. You can set the mode to automatic calculation upon the opening of the spreadsheet, but this could annoy the user (they may have good reason not to want automatic calculation), or the user may change it back to manual while the spreadsheet is open. You do not want the input and the results to be out of sync with each other.

H. Own your spreadsheet and protect the original

There should be contact information in the spreadsheet itself, and clear ownership of the spreadsheet. If you did your job well in following the rules above, you shouldn't have too many people contacting you about problems. Note in my life expectancy calculator, I have my name (people might have to know to look me up in the SOA directory, but if they search for my name online, they will have an easy time finding me). In addition, there's a link to the download page, which has the e-mail of an SOA staff member, as well as a link to download the current version of the tool.

The main reason to do this is to get feedback on your spreadsheet. There may be errors or missing features desired by users that you didn't catch, and allowing for users to contact you directly (or someone else who can contact you directly) will most efficiently get this information to you.

As well, you want to protect your original spreadsheet. In the method of dissemination shown here, the original is on a Web server, so the end user cannot tamper with it in any way. Sometimes, though, your spreadsheet will be sitting in a shared directory where anybody can use it. Some thoughts on handling this situation:

1. **BACKUP!** You should always have more than one copy of your spreadsheet lying around. I not only have my most impor-

tant spreadsheets in a shared directory, but also in my hard drive and my personal network directory. If it's that crucial, I will even e-mail myself a copy, to have a copy sitting on the mail server.

2. Automatic copy: You can write some VBA code that will check the name/location of the file to see if it's the original. If it is the original, then it automatically saves the file under a different name. Here's some sample VBA code, which would be in the code under the VBA object ThisWorkbook:

```
Private Sub Workbook_Open()

If
(ThisWorkbook.FullName=OrigName) Then
ThisWorkbook.SaveAs  Filename:=BaseName
& "TEMP"
End If
End Sub
```

OrigName and BaseName are defined as constants in a module. This is just an example. I think having the spreadsheet save a new copy every time it is opened is overkill.

As I said before, these ideas are nowhere near exhaustive. These issues come up in a variety of situations, so check out the references below as a place to start thinking about interface issues (generally, they are thinking in terms of other contexts, but it still works for spreadsheets).

References:

Joel on Software—online series on user interface design for programmers. Starts with this post: *http://www.joelonsoftware.com/uibook/chapters/fog0000000057.html*

Wikipedia article: "Typography," subsection on Readability and Legibility: *http://en.wikipedia.org/wiki/Typography#Readability_and_legibility*

Color rules of thumb: *http://www.writedesignonline.com/resources/design/rules/color.html*

Jakob Nielsen—Ten Usability Heuristics: *http://www.useit.com/papers/heuristic/heuristic_list.html* (Jakob Nielsen is a usability guru. I highly recommend his publications: *http://www.useit.com/jakob/*)

Bruce Tognazzini—First Principles of Interaction Design: *http://www.asktog.com/basics/firstPrinciples.html*

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

Get In or Get Out Of The Way

by Van Beach

The financial modeling process is set to dramatically change as new regulations become effective and enterprise products begin to hit the market.

Over the last 20 years, the advances enabled by financial modeling are impressive. Insurers have made improvements in risk management, capital optimization, product development and other important aspects of their business with the help of financial models. Regulators, rating agencies, analysts, etc. all utilize disclosures based on financial models for their analysis so the need to manage, measure and report risk continues to grow. Now with principle-based reserves becoming a reality at the end of 2009, the critical importance of financial models has never been more evident. Most will admit, however, that keeping pace with the volume of these demands is becoming increasingly difficult.

Actuaries will be more productive as technology will provide an automated, managed environment for creating a more sophisticated financial modeling process.

The picture that comes to mind is the farmer who brings his products to town using his horse and wagon. Over time, the farm becomes more productive, but demand increases as well. The load gets bigger, the farmer needs to get to town faster, and he's making more frequent trips. The farmer considers his options: Add more horses? Train the horses to get stronger and faster? Get a bigger wagon?

The farmer should be getting a truck.

The horses of the past—financial modeling systems that are coded and deployed in desktop environments—are struggling to keep up. As an example, AG 42 (VA-CARVM) will require stochastic analysis and company-specific assumptions in an auditable, compliant process. No longer just a formulaic calculation, the principle-based reserve will require deterministic and stochastic calculations, a wide array of assumptions and inputs that need to be audited, controlled and documented. More calculations, more data, more assumptions, more iterations, etc.—the size of the process will be significantly larger. Added to the burden, reproducibility, traceability and version control will be essential. The load is heavy.

The good news is that later this year, a new generation of enterprise financial modeling products will start hitting the market. These products will utilize current technology and are designed specifically for the production-style processes, tight timelines, and heavy volumes of models, data, users, iterations, etc., demanded by principle-based reserves and other key modeling-based calculations. These systems will allow server deployment for collaborative, centralized modeling. Security, controls, automation and integration will ensure that the results generated are valid, timely and of high quality. Sequenced processes will allow movement analysis and other orchestrated calculations to be performed automatically. Actuaries will be more productive as technology will provide an automated, managed environment for creating a more sophisticated financial modeling process. Speed, power, control, automation, integration, collaboration—all at levels well beyond what is available today. Yes, these systems are a bit more than legacy desktop applications. These are trucks!

Insurers will need to embrace organizational changes as well. Actuaries will be asked to relinquish some control of the modeling systems and process to IT departments as these systems become more closely linked with other key enterprise IT components. In many cases this will involve a major shift in mindset as actuaries have traditionally owned the entire financial modeling realm. IT will take on increased responsibility for development, testing, deployment and integration of modeling systems. Model development will become more centralized and controlled to comply with Sarbanes-Oxley and other governance requirements. Driving is not the same as riding.

The benefits are real. Insurers that embrace this step-change in technology will be able to meet the increased modeling demands with greater confidence and fewer resources. Reserves and regulatory capital levels will be reduced. Regulatory compliance will be more easily demonstrated and audit costs will be smaller. Processes will be more stable, repeatable and automated resulting in less risk of error and lower staffing costs. The product design and implementation processes will be improved as collaboration, process memory and access to computation resources are all enhanced. The partnership between IT and actuaries will allow each to do what they do best—IT will be managing the technology; actuaries will be managing risk. The technology will be enabling. These are NOT horses!

It is time to get in a truck. Financial models are central to competitive advantage and new solutions are here. And for those insurers that stick with their horses, their legacy desktop environments, make sure you mount a slow-moving vehicle sign on the back of your wagon and stay to the side of the road!

Van Beach, FSA, MAAA, is a senior consultant at Towers Perrin. He can be contacted at van.beach@towersperrin.com.

Cool Tech

by Matthew Wilson

Welcome to another edition of Cool Tech. In this edition I talk more about Content Management Systems, running a small business, XAMPP, Yahoo!Pipes and displaying YouTube videos, Flickr Images and Amazon.Com books on your Web site.

This edition is loaded with links that I think you'll find interesting. In the small business section I explore outsourcing your life. That's right. We're going to a whole new level with outsourcing.

Content Management Systems (CMSs)

Last time we looked at two popular Content Management Systems (CMSs): Wordpress and Joomla. If you're interested in setting up a professional Web site then these two choices are definitely the way to go.

Both Wordpress and Joomla have programs that extend the capability of the base program. For example, you can add a forum and photo gallery to Joomla if you wish. Wordpress has hundreds of plug-ins that do all kinds of things such as automatically submitting content to Google or displaying Youtube videos.

I wrote my own simple PHP program to display Youtube videos in Wordpress. Check out this page: *http://www.1913intel.com/videos*. This runs my YouTube play list. Get the code for this program at end of this article.

There are quite a few open-source CMSs available today. You can demo many of them at this Web site:

http://www.opensourcecms.com/

Two other CMSs worth taking a look at are Drupal and XOOPs. Installation of both packages is fairly easy.

How to install Drupal (video): *http://www.volkanrivera.com/eng/?p=13*

How to install XOOPS: *http://www.xoops.org/misc/docs/Xoops-for-Dummies/install.htm*

All of these packages allow you to easily build database-driven Web sites with Web pages created dynamically on the fly. Make sure to stick with PHP and MySQL initially, because they're easier to set up than PERL programs.

Top 10 mistakes when selecting a CMS:

http://www.opensourcecms.com/index.php?option=com_content&task=view&id=2291&Itemid=1

Running a Small Business

Let's say that you've decided to start a small actuarial consulting firm. You could set up a professional looking Web site using either Wordpress or Joomla. Typically, you can find a host that will install these packages for free.

If you need graphic design or other types of Web site work, then I suggest you check out www.elance.com. At Elance, people from around the world can bid on your job. A programmer in India can probably do the job for much less than a local programmer. I've used Elance in the past, and it worked well for me.

Don't stop at just outsourcing your Web site work. What about outsourcing your life?

http://finance.yahoo.com/family-home/article/103092/Outsourcing-Your-Life

Check out this Business Week article titled, "Mom-and-Pop Multinationals." It shows how an African-American couple outsourced many tasks for several of their businesses.

http://www.businessweek.com/magazine/content/08_28/b4092077027296.htm

I'm in the process of hiring *http://www.globetask.com/* as my Virtual Assistant. I run a Web site on international conflict and need an assistant to help with various tasks such as tracking down YouTube videos and developing a strategy to attract advertisers to my Web site. The pay is $10/hour. I'll let you know how it works out.

Information about outsourcing comes from the book, "The 4-Hour Workweek." The Web site for this is *http://www.fourhourworkweek.com.* Why do the work yourself when someone in India can do it for much less? Also, they'll do it while you sleep. When you wake up in the morning your tasks are done. Your work is getting done 24 hours a day!

Instead of buying "The 4-Hour Workweek," try locating it at your local library. I live in the Seattle area, and when I want a book or audio CD I go online to track it down (*http://www.kcls.org*). I order it online and have it delivered to the local library near where I work. I get an e-mail notifying me when it has arrived. I then go to the local library to pick it up and spend only a couple of minutes in the actual library building.

You're going to need sales in your new job as an actuarial consultant. I was absolutely blown away at the sales concepts described at www.highprobsell.com. Get ready to have your sales world turned upside down using High Probability Selling concepts. High Probability Selling is a no-nonsense approach that focuses on efficiently finding people that already want to buy what you have to sell. You quickly learn to disqualify people that aren't ready to buy now. Also, you will learn how to stay in touch with people so that when they're ready to buy they'll think of you.

XAMPP

In the last edition of Cool Tech, I left off with a reference to XAMPP. XAMPP is a series of programs that will allow you to easily set up a Web server on your PC. Additionally, it comes with PHP, Perl, MySQL and a whole lot more. So let's get it running on your PC.

Go here to find the right version for you:

http://www.apachefriends.org/en/xampp.html

I like to use the zip version. After downloading the package, unzip it to c:\xampp. Read the readme_en.txt file. Go to the DOS command

(continued on page 22)

prompt and launch the setup_xampp.bat file. That's it. There is no messing around with the registry.

Go to the DOS command prompt and run apache_start.bat. Do not close this window or you will shut down the Apache server.

Launch a Web browser such as Internet Explorer or Firefox. Type in http://localhost

You should see the following statement:

Congratulations:
You have successfully installed XAMPP on this system!

In the small business section I explore outsourcing your life. That's right. We're going to a whole new level with outsourcing.

Check the status. Everything should be active except MySQL and FTP programs.

Let's check out PHP. Write the following program in a text editor. Call it hw.php. Put it in the htdocs subdirectory.

```
<?PHP
   Echo "Hello World";
?>
```

Type this in browser address line: *http://localhost/hw.php*

If you have a problem, go back to the XAMPP site and snoop around for answers. Another option is to search the Web for answers. I experienced no problems on my Windows XP operating system. I have also installed this package on the Linux operating system (Redhat 8) with no problems.

Let's check out Perl. Call the following program hw.cgi. Put it in the htdocs subdirectory.

```
#!"C:\xampp\perl\bin\perl.exe"

print "Content-type: text/html\n\n";
print "<H1>Hello World</H1>\n";
```

Type this in browser address line: *http://localhost/hw.cgi*

Hopefully it worked. If not, you're on your own.

Check out these Web sites if you want some cool open-source programs to run:

http://www.hotscripts.com
http://www.sourceforge.net
http://www.freshmeat.net

When you're done playing around with the server you can close it by shutting down the DOS window.

Using Yahoo!Pipes

Most of you have heard about RSS feeds. Many of you use RSS feeds. Now there's a way to gather RSS feeds, remove duplicates, filter them, sort them and more. Check out *http://pipes.yahoo.com.*

I run a Web site located at *http://pipes.1913intel.com*. This is a fully automated system. News from several RSS feeds originating from Google News goes into my Pipes program. Once the RSS feeds are sliced and diced, the end result automatically feeds into my Wordpress Web site at *http://pipes.1913intel.com*. This allows me to automatically keep up with events in Russia, China, Iran and Israel. Check out this link:

http://pipes.yahoo.com/pipes/pipe.info?_id=3EzWPqy_2xG0Bc9c00qv4w

This Pipes program points to YouTube videos based on the article titles located on my front page at *http://www.1913intel.com*. Click on the individual links to see the YouTube video: *http://pipes.yahoo.com/pipes/pipe.info?_id=a36132c04d387dc6ee953f305be74831*

How did I get my Pipes programs running? I borrowed some code from other people and modified it. You can snoop through programs written by other people and make a copy for yourself.

Displaying YouTube Videos, Flickr Pictures and Amazon.Com Books

The concept for the displaying YouTube videos, Flickr pictures and Amazon.Com books is the same in a PHP program:

1. Read an RSS file: any read file function works here.
2. Parse the RSS file into an array: simplexml_load_string($file)
3. Loop through the array: foreach()

Looping through the array is the hard part, because you need to know the array elements, which change for each application. I searched Google for YouTube and simplexml to find the RSS/XML elements that I needed to display YouTube pictures. Also, looking at the RSS feed elements directly helps. The simplexml_load_string() function is already built into PHP.

The Amazon.Com program can be conceptually upgraded (ramped up) to replicate most of Amazon.Com content. I know this because I tried to do it a few years ago using the browse node codes from *http://www.browsenodes.com*. I also wrote a PHP program to capture my own node codes using Regular Expressions (RegEx). That was before Google kicked me out of their search engine results.

Here are the programs and source code for displaying books, videos and pictures:

Books: *http://www.1913intel.com/demo/amazon-xml.php*

Get the code here: *http://www.1913intel.com/demo/amazon-xml.txt*

Just change the txt extension to php in order to run the program. This program returns two random books from a list.

Videos: *http://www.1913intel.com/demo/videos.php?tags=Paris+Tourism*. Change the tags to anything you wish.

Get the source code here: *http://www.1913intel.com/demo/videos.txt*

Pictures: *http://www.1913intel.com/demo/flickr.php?tags=Russia,River*

This program displays thumbnails so that there are no copyright problems. One photographer complained to me about displaying thumbnails of his photographs. He wanted to charge me $300 per thumbnail. I modified the program to exclude his photos.

I wanted to put a sticker on his forehead that says, "Logic Does Not Apply Here!" Is it a good idea to add photos to Flickr knowing that all of your material is freely available via the Flickr API? Perhaps a more secure location might be in order if you're going to panic over thumbnails? Anyway, the courts have already made clear that displaying thumbnails is NOT a copyright violation. We can thank Google for getting that decision in court.

Get the source code here: *http://www.1913intel.com/demo/flickr.txt*

If you have any cool ideas or links for Cool Tech then shoot me an e-mail at *matt_wilson@farmersinsurance.com.* 💻

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

R Corner

by Steven Craighead

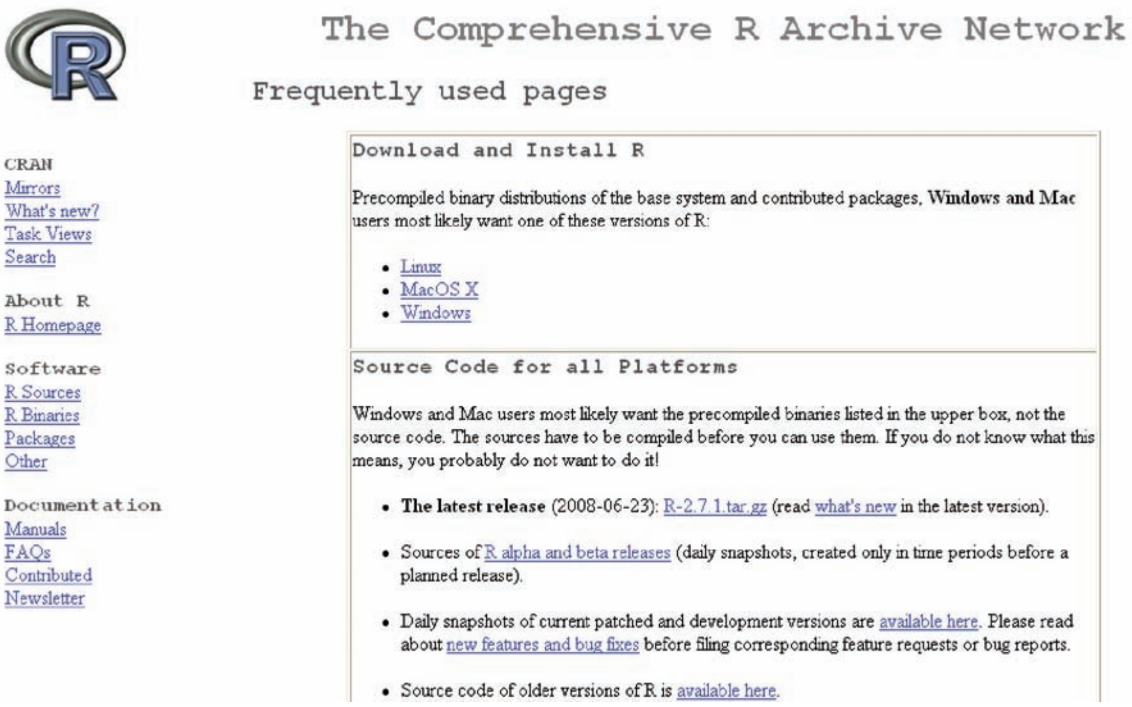
R[1] has become the lingua franca of the statistical world. Personally, I believe that it also has a lot to offer us actuaries. Most of our models can be constructed within Excel, but there are still areas where other modeling languages can be useful. R is a good candidate because:

1. It is open source.
2. It runs on multiple platforms.
3. It is free.
4. It has over 1000 various packages available.
5. It can easily be integrated into multiple packages including Excel.
6. It is constantly improving with new statistical tools constantly being developed for R by researchers.

I am going to write a series of columns on R for the readers of CompAct and Expanding Horizons explaining how to use R as well as describing various packages available.

By the end of this article you should know how to locate, download and install R. I will also tell you how to start and end R and how to access the help features. The next article I will dedicate to how to get data into and out of R and how to access the data from a dataframe, which is the most common data structure within R. Other articles will follow outlining how to create modeling formulae and using R to model not just linear regression but other predictive models such as GLM and GAM.

Figure 1. The cran.r-project.org web page.



The Comprehensive R Archive Network

Frequently used pages

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Linux
- MacOS X
- Windows

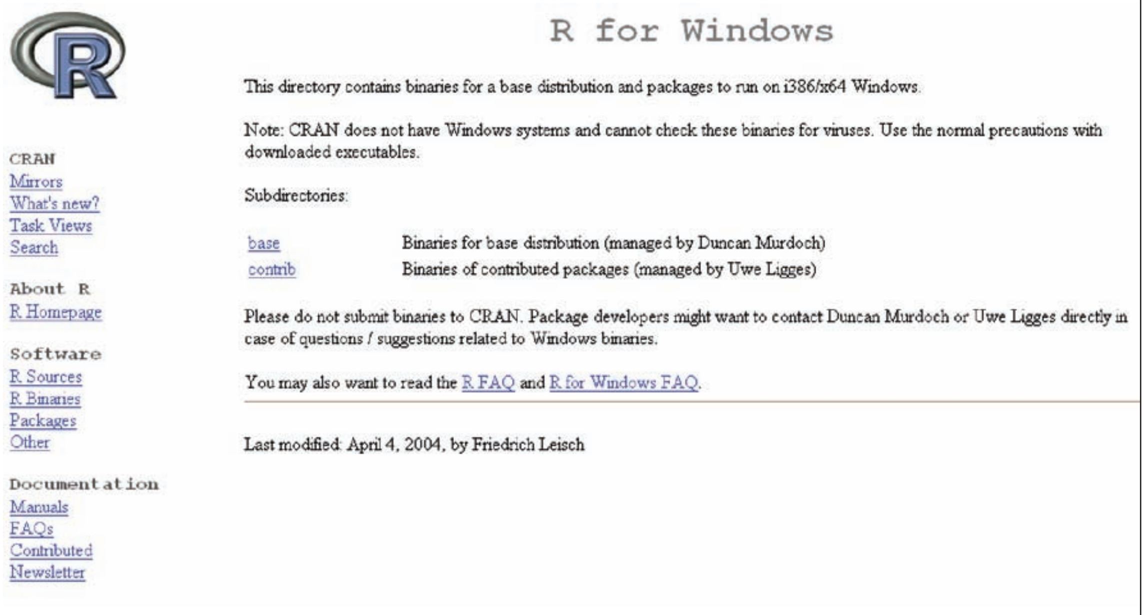
Source Code for all Platforms

Windows and Mac users most likely want the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- **The latest release** (2008-06-23): R-2.7.1.tar.gz (read what's new in the latest version).
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.

R is located on "The Comprehensive R Archive Network," whose Web site is *http://cran.r-project.org/*. When you go to the site, the homepage displays the frequently used pages. See Figure 1. Note that the first set of choices is on how to download and install R from existing pre-compiled binaries for Linux, Mac and Windows. Choose your favorite operating system platform and you will go to a Web page that allows you to choose between base and contributed packages. Due to my limited access to testing other operating systems, in the remainder of this article I will only describe how to install the Windows binary. Notice in Figure 2, you will have the two choices of base and contrib. Initially, choose base and download the most current release. At the time of the writing of this article, version 2.7.1 is available.

Figure 2. The R for Windows Page



CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed
Newsletter

R for Windows

This directory contains binaries for a base distribution and packages to run on i386/x64 Windows.

Note: CRAN does not have Windows systems and cannot check these binaries for viruses. Use the normal precautions with downloaded executables.

Subdirectories:

base	Binaries for base distribution (managed by Duncan Murdoch)
contrib	Binaries of contributed packages (managed by Uwe Ligges)

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

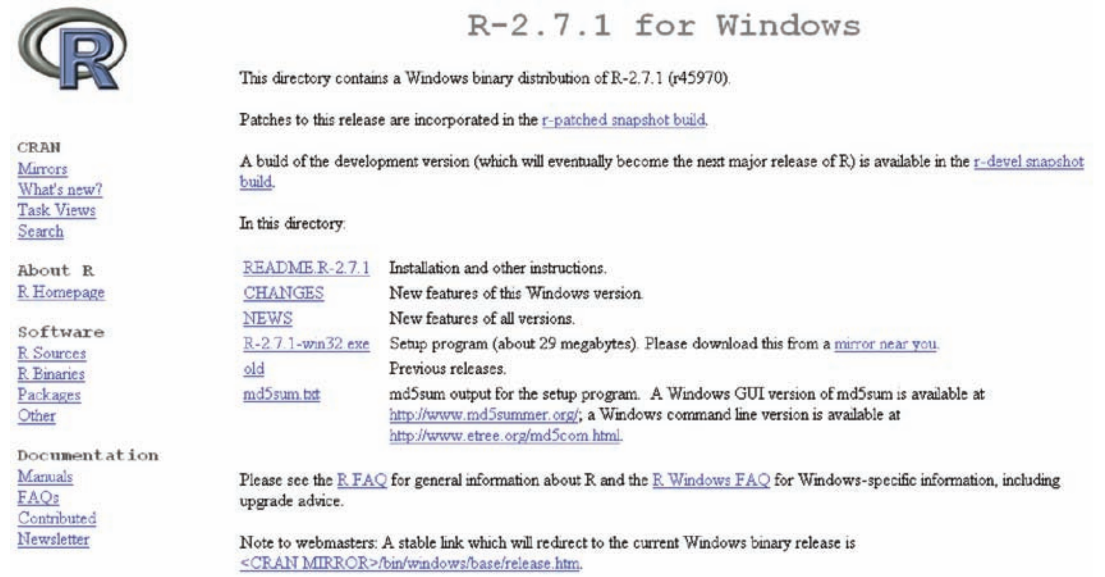
Last modified: April 4, 2004, by Friedrich Leisch

Observe from Figure 3, the availability of the file R-2.7.1-win-32.exe (or whatever the latest version is). Download this file to some location on your hard disk. Next run the executable and begin the installation process.

(continued on page 26)

R Corner • continued from page 25

Figure 3. R-2.7.1 for Windows



R-2.7.1 for Windows

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed
Newsletter

This directory contains a Windows binary distribution of R-2.7.1 (r45970).

Patches to this release are incorporated in the r-patched snapshot build.

A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.

In this directory:

README.R-2.7.1	Installation and other instructions.
CHANGES	New features of this Windows version.
NEWS	New features of all versions.
R-2.7.1-win32.exe	Setup program (about 29 megabytes). Please download this from a mirror near you.
old	Previous releases.
md5sum.txt	md5sum output for the setup program. A Windows GUI version of md5sum is available at http://www.md5summer.org/; a Windows command line version is available at http://www.etree.org/md5com.html.

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information, including upgrade advice.

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm.

If you choose the default installation directory, R will be installed in C:\Program Files\R\R-2.7.1. Note how the release is installed under a general R folder. I suggest that you make sure to keep your releases separate because of potential future conflicts with new releases.

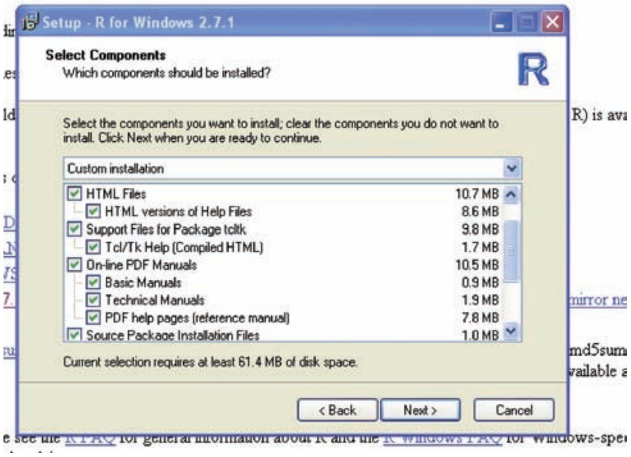
The installer will then ask what support files you want to have installed with the base installation. I suggest that you should include the technical manuals as well as the reference manual. These will become handy as you expand your modeling skills. See Figure 4.

I also suggest that you use the default startup options as well.

To run R from your Start button, choose the Programs tab and select the R selection and then choose the current release under that selection.

Figure 5 displays an empty R console that will then appear. R will suggest that you should run demo() for some examples, or help() for on-line help or help-start() for an html based help brower for R. It also states that to exit, you type q() and press return.

Figure 4. PDF Manual choice in R installation



Select Components
Which components should be installed?

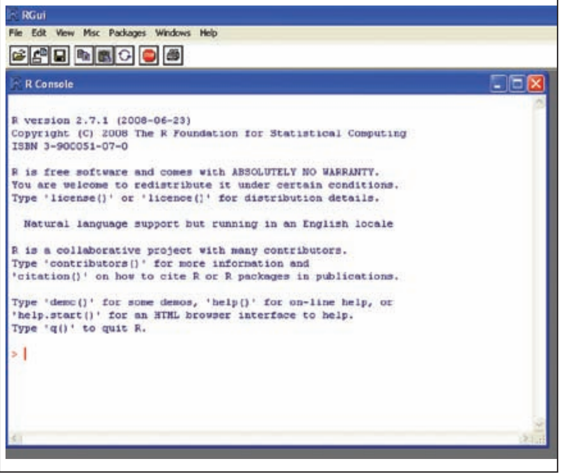
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom installation

☑ HTML Files	10.7 MB
☑ HTML versions of Help Files	8.6 MB
☑ Support Files for Package tcltk	9.8 MB
☑ Tcl/Tk Help (Compiled HTML)	1.7 MB
☑ On-line PDF Manuals	10.5 MB
☑ Basic Manuals	0.9 MB
☑ Technical Manuals	1.9 MB
☑ PDF help pages (reference manual)	7.8 MB
☑ Source Package Installation Files	1.0 MB

Current selection requires at least 61.4 MB of disk space.

Figure 5. Opening R Screen.



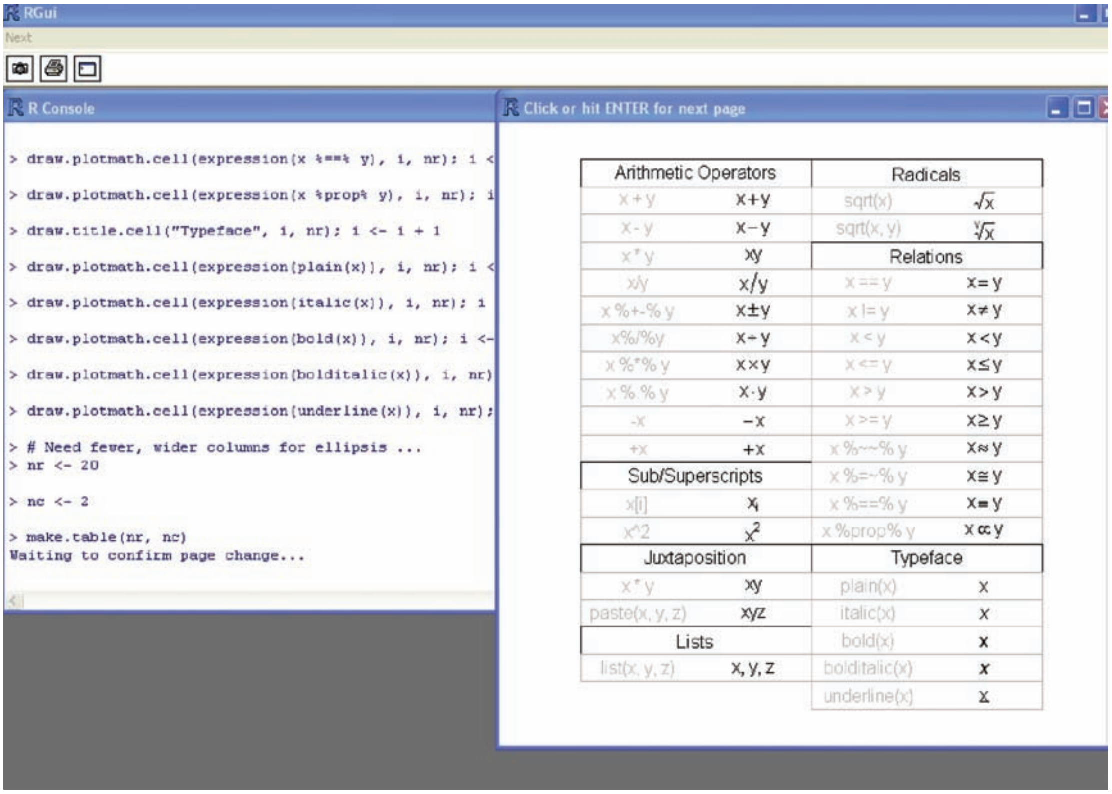
```
R version 2.7.1 (2008-06-23)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

The demos() will give a list of various demonstrations available. You run these demonstrations by type "demo ("demoprogram") in R. For instance to look at the "plotmath" demo, type:

demo("plotmath")

You will see multiple displays of various mathematical formulas and symbols available, displayed in a separate graphics window. See Figure 6 for an example. The window may pause and you need to press enter to continue to the next example within the demo. Note the various commands that appear in the R environment.

Figure 6. Plotmath demo screen



```
> draw.plotmath.cell(expression(x %==% y), i, nr): i <
> draw.plotmath.cell(expression(x %prop% y), i, nr); i
> draw.title.cell("Typeface", i, nr); i <- i + 1
> draw.plotmath.cell(expression(plain(x)), i, nr); i <
> draw.plotmath.cell(expression(italic(x)), i, nr); i
> draw.plotmath.cell(expression(bold(x)), i, nr); i <-
> draw.plotmath.cell(expression(bolditalic(x)), i, nr)
> draw.plotmath.cell(expression(underline(x)), i, nr);
> # Need fewer, wider columns for ellipsis ...
> nr <- 20
> nc <- 2
> make.table(nr, nc)
Waiting to confirm page change...
```

Arithmetic Operators		Radicals	
x + y	x+y	sqrt(x)	$\sqrt{x}$
x - y	x−y	sqrt(x, y)	$\sqrt[y]{x}$
x * y	xy	Relations	
x/y	x/y	x == y	x=y
x %+-% y	x±y	x != y	x≠y
x%/%y	x÷y	x < y	x<y
x %*% y	x×y	x <= y	x≤y
x %.% y	x·y	x > y	x>y
-x	−x	x >= y	x≥y
+x	+x	x %~~% y	x≈y
Sub/Superscripts		x %=~% y	x≅y
x[i]	x_i	x %==% y	x≡y
x^2	x^2	x %prop% y	x∝y
Juxtaposition		Typeface	
x * y	xy	plain(x)	x
paste(x, y, z)	xyz	italic(x)	*x*
Lists		bold(x)	**x**
list(x, y, z)	x, y, z	bolditalic(x)	***x***
		underline(x)	$\underline{x}$

To use help, either type help(command) or ?command in the R environment. For example, to find out help on the linear regression method, you can type:
?lm
or
help(lm)

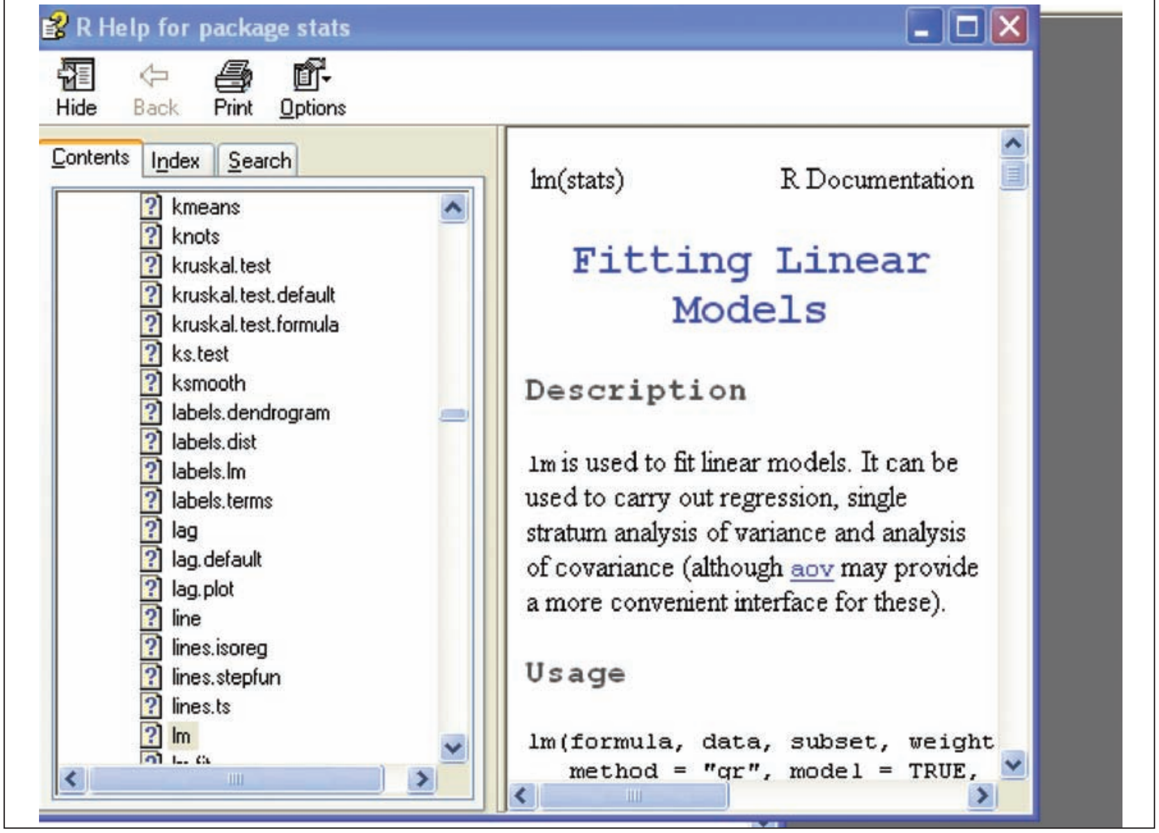
Figure 7 displays an example of the Web browser that pops up with a description of Fitting Linear Models. If you page down through various help articles, various R commands will be displayed, which you can copy and paste into R.

(continued on page 28)

Steven Craighead, ASA, MAAA, CERA, is a consultant at Towers Perrin. He can be contacted at steven.craighead@towersperrin.com.

R Corner • continued from page 27

Figure 7. Help Browser display lm function.



kmeans
knots
kruskal.test
kruskal.test.default
kruskal.test.formula
ks.test
ksmooth
labels.dendrogram
labels.dist
labels.lm
labels.terms
lag
lag.default
lag.plot
line
lines.isoreg
lines.stepfun
lines.ts
lm

lm(stats) R Documentation

Fitting Linear Models

Description

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weight
   method = "qr", model = TRUE,
```

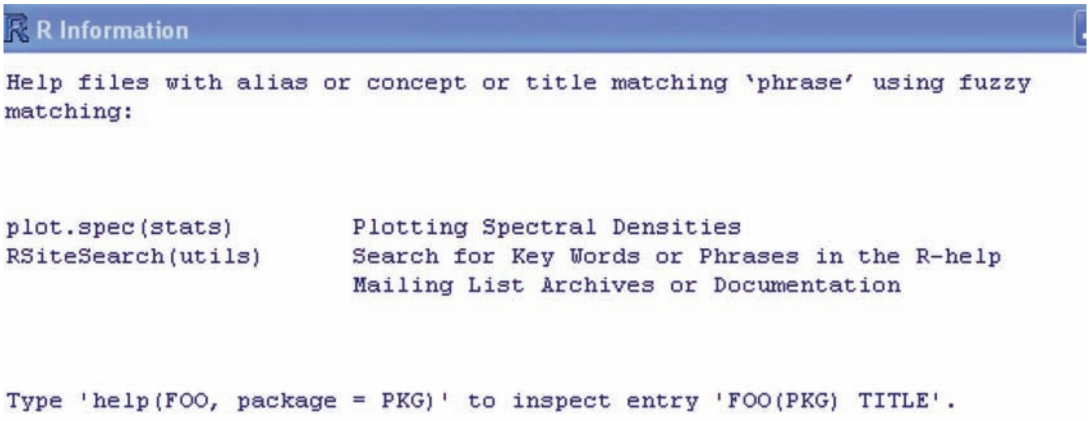
Also, in the help browser you are able to index and search for various other objects or methods available within the base system.

Another helpful help command is "help.search("phrase"), where in the R environment you can look for various packages and methods and objects that contain the phrase that you enter.

For example, if you enter: help.search("phrase")

Look at Figure 8, for the R information screen that results from the help.search command.

Figure 8. help.search results



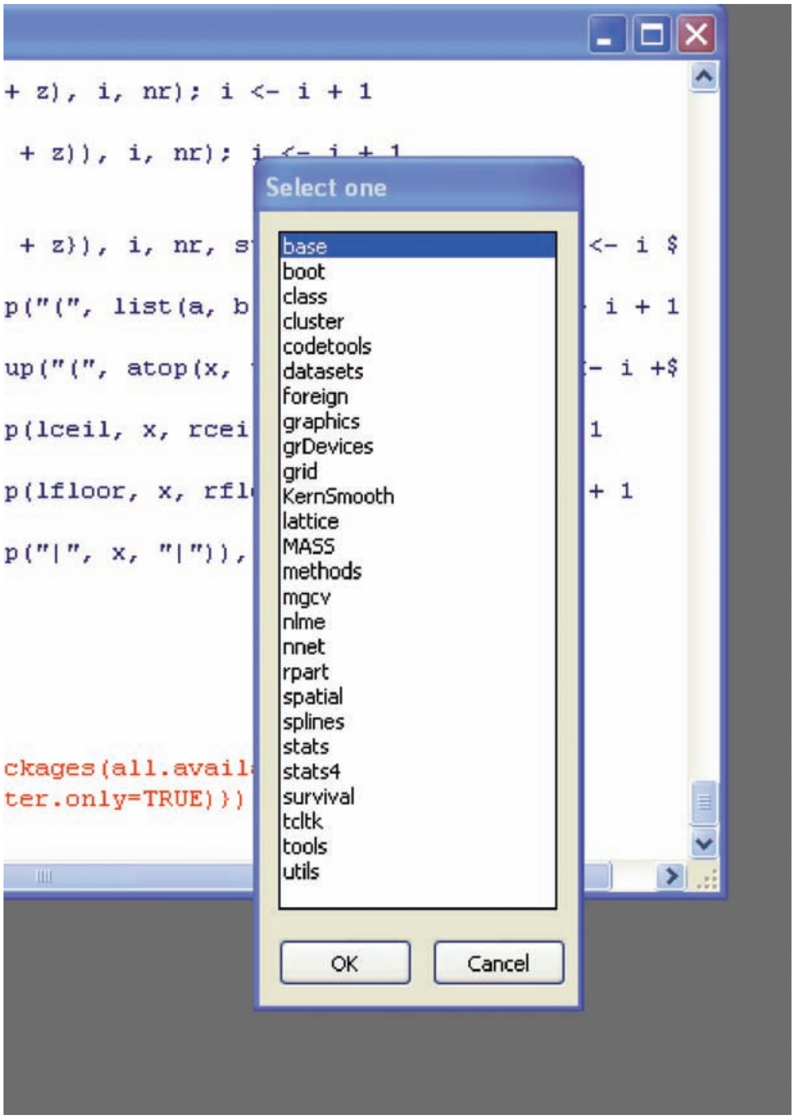
```
Help files with alias or concept or title matching 'phrase' using fuzzy
matching:



plot.spec(stats)        Plotting Spectral Densities
RSiteSearch(utils)      Search for Key Words or Phrases in the R-help
                        Mailing List Archives or Documentation



Type 'help(FOO, package = PKG)' to inspect entry 'FOO(PKG) TITLE'.
```

Figure 8 is the R information screen that results from the help.search command.

Observe from Figure 8 that if you load the stats package and invoke help on the plot.spec function, you will find some use of the word "phrase."

By the way, you load packages by choosing the Load Package option off of the main menu Package choice. See Figure 9.

Figure 9. Loading Packages selection screen



```
+ z), i, nr); i <- i + 1

 + z)), i, nr); i <- i + 1

 + z)), i, nr, s            <- i $

p("(", list(a, b           i + 1

up("(", atop(x,           <- i +$

p(lceil, x, rcei          1

p(lfloor, x, rfl          + 1

p("|", x, "|")),

ckages(all.avail
ter.only=TRUE))
```

If you want to specify the subdirectory in which you want to work in Windows, choose the Change Directory option off of the File option of the main menu. From the main menu, you can observe how to save and load workspaces and history files. The workspace will contain your data and various functions and objects that you build, but the history files will store the command you enter in the R environment. As you develop your own models you will need to exploit these features.

One final thing, if you want to stop R from running some long process, press the Stop Sign Icon at the top of the environment.

Remember to type q() to exit.

I'm going to stop for now. Next column, we will look at various ways to get data into and out of R. ■

References:

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.*

Predictive Modeling

by Barry Senensky

[Editor's Note: This article originally appeared in the July 2008 edition of CompAct.]

Recently I attended the 4th annual conference on predictive modeling for property and casualty insurers. There were approximately 300 casualty actuaries in attendance. Many, to my surprise, were experienced predictive modelers, eagerly looking for new ideas to enhance and expand their capabilities. From being used by almost no one six years ago, I learned, predictive modeling is now being used for pricing by many property and casualty companies.

Yet in the life insurance industry, almost no one uses it. Why?

Further discussions with my P&C colleagues revealed that there had been a critical catalyst, only a few years ago. The catalyst was an innovative P&C firm, who discovered that an individual's credit score was highly predictive of their likelihood for filing an automobile claim.

They discovered this by experimenting with predictive modeling.

Incorporating credit scores into their underwriting and pricing of automobile insurance immediately, measurably, and visibly increased this firm's profitability. They had an instant, significant advantage over their competitors. Of course, their advantage did not last long. Other P&C insurers leapt to incorporate predictive modeling in their pricing—eager to re-level the playing field.

After the conference I began to wonder why life insurers—specifically life insurance actuaries—have yet to take to predictive modeling with the same vigor as their P&C counterparts. I thought back to when I had begun my own predictive modeling company …

In 2001, along with another actuary, Jonathan Polon, I started Claim Analytics, a company dedicated to bringing modern predictive modeling to the life insurance industry.

To us, the time seemed ripe. Predictive modeling had been successfully used for years, in an ever-widening range of industries. The ability of modern computing to crunch massive amounts of data had radically changed analytical methodology. The traditional methods of statistics, constrained by the fact that they were developed in the days when all computations had to be done by hand, were beginning to look limited—maybe even dated.

What exactly is predictive modeling? Where better to find a definition than Google, which itself is powered by predictive models:

"Predictive modeling is a process used in predictive analytics to create a statistical model of future behavior. Predictive analytics is the area of data mining concerned with ***forecasting probabilities and trends.*** *A predictive model is made up of a number of predictors, variable factors that are likely to influence future behavior of results. In marketing, for example, a customer's gender, age and purchase history might predict the likelihood of a future sale.*

To create a predictive model, data is collected for the relevant predictors, a statistical model is formulated, predictions are made, and the model is validated. The model may employ a simple linear equation or a complex neural network or genetic algorithm."

Jonathan, an actuary who had followed a non-traditional career path, had worked in the credit card industry building predictive models. I had many years experience working in the life insurance industry doing a range of traditional actuarial work. We thought we had an ideal fit for a new company, building predictive models. We would take the life insurance industry by storm.

That was seven years ago. We now have clients throughout North America. Our flagship product, a predictive model that "scores" disability claims on the likelihood of recovery in a given timeframe, has produced terrific results. We have built predictive models for pricing, reserving, and fraud detection, all of which have been received extremely favorably by our clients.

Yet, while we are very pleased by our progress, we wonder why predictive modeling hasn't "rocked" the industry. Why isn't every life insurance company using predictive modeling?

Predictive models are everywhere! Financial institutions use them to determine your credit score when you want a loan; the post office uses them to decipher your handwriting; meteorologists use them to predict weather; retailers use them to decide what to put on their shelves; marketers use them to improve their products; concessionaires use them to figure out how many hot dogs they need for the "big game." They are even used by sports teams to pick player personnel.

Is it just chance that the P&C actuaries found predictive modeling first? Is it inevitable that life insurance actuaries will soon starting using it? Or is there something fundamentally different in the two businesses that makes predictive modeling a fit for P&C insurance, but not for life?

(continued on page 32)

Predictive Modeling • continued from page 31

On reflection I came up with a greater understanding of how life actuaries operate. And why they have been reluctant to adopt predictive modeling.

Actuaries are trained in mathematical techniques developed by mathematicians and statisticians. Predictive modeling, which was born in the realm of computer science, is not part of an actuary's natural toolkit.

Take the example of a mortality study. Life actuaries will use the tools and techniques they learned studying for actuarial exams. They are unlikely to use a new predictive modeling technique such as "Boosted Trees," even though the new technique may produce a better result.

Predictive modeling, by contrast, was developed by computer scientists, schooled in the methods of if/then/loop analysis and massive data-crunching. The methods of analysis they have developed, rather than using elegant complex mathematical shorthand to reveal trends, are, in concept, more primitive. These methods look at every bit of data, using the brute-force power of modern computing.

The insight yielded by predictive modeling are impressive and daily more respected. Yet the method is foreign to the classically-trained actuary.

Still—this was at one point true of the P&C industry. But then came the catalyst, the one innovative firm that used predictive modeling. The visibly profit-generating discovery they came up with made all the difference.

It put me in mind of the early 1980s, when some bright spark came up with the idea of offering lower life insurance rates to non-smokers than smokers. The benefits were immediate. Life insurance actuaries adjusted their approach almost overnight.

So what will the future bring?

It is only a matter of time before life actuaries make predictive modeling tools (generalized linear models, neural networks, genetic algorithms, classification and regression trees, etc.) best practice for pricing, underwriting, experience studies, underwriting and many other applications. These new tools are just too powerful to stay on the shelf forever. Life insurance will come around to predictive modeling, as have other industries dealing with statistics and prediction.

Will there be the one sudden catalyst, as there was for P&C Insurance? Or will there be a steady increase over time—more like a snowball effect—as life actuaries gradually discover the advantages of predictive modeling?

I don't know. But I am confident that the change will come. And my advice to ambitious actuaries is, find out as much as you can about predictive modeling. You might find it comes in handy—over time, or immediately.

Barry Senensky is president of Claim Analytics. He can be contacted- at bsenensky@claimanalytics.com.

Effectively Mitigating Spreadsheet Risk in Insurance

by Eric Perry

[Editor's Note: This article originally appeared in the July 2008 edition of CompAct.]

The power and flexibility of Microsoft Office Excel have made it the ubiquitous analysis tool of choice for insurance companies in determining premiums and the risk value of clients. Complex spreadsheet models are often developed by actuaries and then run thousands of times to determine risk under thousands of scenarios for each insurance policy. According to a 2006 study led by Louise Pryor, a well known actuarial consultant, 732 respondents across the global insurance market concluded that: "Excel appears to be almost universally used, with 98 percent of all respondents saying they use it."

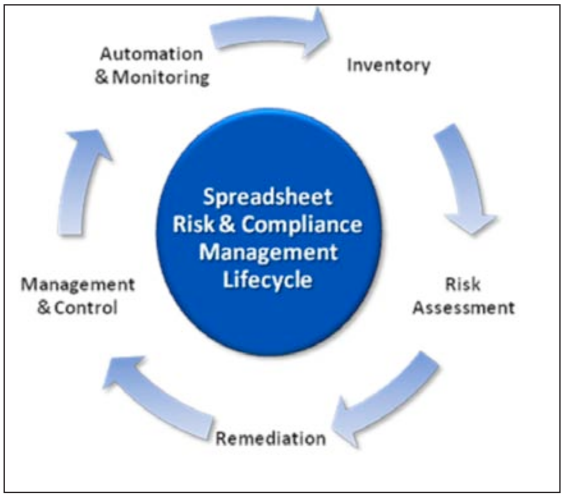
However, despite their ubiquity, stand alone spreadsheets have proven difficult to control. They are often stored on employee desktops and corporate file shares; an uncontrolled environment that is absent many of the IT controls expected from enterprise class software. In addition, the complexity of actuarial spreadsheets makes them error prone. Even one small mistake in an actuarial spreadsheet model—incorrect input data, a formula error, a bug in a macro, or a false assumption—can produce the wrong results and eventually affect an entire business.

Compliance Mandates and Auditor Guidance

In addition, actuarial spreadsheet models have to comply with ever-changing regulatory mandates such as FAIS, ERISA, PBGC, FAS 87, FAS 106/158, and IAS 19. Leading tax and audit firms have begun to scrutinize the use and dependency on actuarial spreadsheets, and are recommending the application of spreadsheet controls to help insurance firms effectively mitigate risk and improve compliance.

Establishing Effective Spreadsheet Controls

Applying a lifecycle approach to establishing spreadsheet controls has proven to be effective while satisfying the latest guidance from leading tax and audit firms.



First, organizations should discover and inventory all critical actuarial (and financial) spreadsheets across all corporate data sources, including corporate file shares, employee desktops, and document repositories.

Second, assess each spreadsheet for risk, complexity and impact on key financial account balances.

Next, remediate all critical spreadsheets to correct errors, apply best practices, and improve performance.

Then, establish a controlled environment to manage critical spreadsheets that incorporate key IT controls recommended by auditors, including: version control, access control, change control, documentation, input control, security and data integrity, documentation, backup and archival, software development lifecycle, logic inspection, segregation of duties/roles/procedures, and overall analytics. In most cases, a controlled environment should be established with appropriate spreadsheet management software.

Finally, automate the controlled environment and monitor the process from end-to-end. For example, review and approval of critical spreadsheets can be automated via workflow, proactive alerts can be sent to users about key spreadsheet changes to formulas and macros, and management reports can be sent via e-mail to highlight potential risk and compliance gaps. Management reports can include a review of user access levels to critical spreadsheets, approval status, or cell level change reports.

Driving Business Value

By managing critical actuarial spreadsheets like enterprise software applications, insurance firms can improve visibility and control, improve compliance with regulatory mandates, mitigate the risk of errors, and improve productivity for actuaries. Leveraging the new emerging breed of enterprise spreadsheet management software tools and applications can be a catalyst to helping firms accomplish these goals by automating the spreadsheet risk and compliance management lifecycle. ■

Prodiance provides spreadsheet governance software and is the winner of the Microsoft 2007 Innovator of the Year award. Further information can be found at *www.prodiance.com.*

Eric Perry is vice president, Marketing at Prodiance. He can be contacted at eric.perry@prodiance.com *or 866-569-5678.*

The Actuary in an Information Technology World—Planning for Project Success

by Stewart Shay

[Editor's Note: This article originally appeared in the April 2008 edition of CompAct.]

As companies move away old desktop and mainframe-based systems, the traditional role of actuary is expanding to supply the insurance knowledge and numerical quality assurance required on these complex projects. The actuary is often a critical stakeholder in any insurance sales or operations support project.

Of the many critical success factors in project management, the capture of precise and correct business rules and requirements is considered the most crucial. Regardless of the Systems Development Lifecycle (SDLC) used, true requirements will emerge and will be far less expensive if found sooner rather than later. Successful projects involving IT will have requirements identified and documented by a Business Analyst (BA) in a formal Business Requirements Document (BRD).

Problems found early in the project life cycle are much less costly to fix. A common rule of thumb is that the project's analysis phase should take 25 percent of the total project's elapsed time to complete but only cost approximately 5 percent of the total project expense. Some studies show that a defect found or avoided early (due to good specifications) can cost 20 times less to fix than one found later in User Acceptance Testing.

For life insurance applications, the gathering of user requirements will normally be communicated to the BA from three sources: sales, operations and the actuary. By contributing to the BRD, providing thorough product specifications, and supporting the QA effort with sound test cases, the actuary can significantly and positively affect the success of any project involving the issue and administration of life insurance products.

Often, when planning a project, the actuary's time is glossed over and possibly not even included at all. However, the actuary often becomes a key consultant to the project manager. The actual time spent by the actuary in support of the project may ultimately equal the project manager's (or business analyst's) effort, which typically amounts to 5- to 10-percent of the overall project.

Depending on the corporate culture and the SDLC used, project managers should consider and include time for actuarial resources for the following activities:

Initiation Phase:

- The actuary may be asked to provide input for a cost/benefit analysis to evaluate whether or not it is worthwhile to take on the project from a financial standpoint.

Analysis Phase:

- The actuary is the owner and producer of the insurance product specifications.
- The actuary is a consultant to business analysts who develop the Business Requirements Document.
- The actuary is a provider of test case scenarios and expected results to be used in the QA process for checking illustrations (new business and reproposals), gross premium and month-i-versary calculations on non-traditional products, policy cost disclosures, policy annual reports, and 7702/7702a tax calculations.

- The actuary is a consultant and quality assurer on specifications for policy conversions.

Design and Coding Phases:

- The actuary is a consultant for specifications on change requests found after the initial Business Requirements Document is completed.
- The actuary is involved in any negotiations to alter product specifications to fit into an existing system's infrastructure. This involves a tradeoff between saving money by reusing existing technology versus adding new functionality that requires new technology.

Quality Assurance Phase (User Acceptance Testing):

- The actuary is a primary stakeholder in assuring that the new business illustration and administration systems match. This usually involves the reconciliation of the new system's calculations (e.g., policy values, tax premiums, etc.) against an existing legacy system or a trusted spreadsheet.

Implementation Phase:

- Exception handling—while we'd like to think that any new system handles everything thrown its way, oftentimes there is some functionality that does not make it to the first phase of introduction. In such cases, going live means being able to handle some functions outside of the system manually. Actuaries may be involved in planning who, what, when, where and how such processes are to be handled until sufficient automation arrives.

Post-Implementation:

- As a stakeholder of the project, the actuary will be asked to contribute to lessons learned sessions designed to improve future project implementations.
- A systems project that has multiple phases will require a repeat of some of the tasks mentioned above.

Whether a company is large or small, all of these activities should be considered in the project planning phase to ensure actuarial resources are allocated appropriately and are available when needed. For many life insurance IT projects, the actuary is just as key a stakeholder as any other member of the project team and his/her participation should not be overlooked.

Stewart Shay is senior business analyst with iPipeline. He can be contacted at sshay@ipipeline.com.

To Err Is Human; To Correct, Divine

by Mary Pat Campbell

[Editor's Note: This article originally appeared in the January 2008 edition of CompAct.]

How error-riddled are your spreadsheets? How much can a simple Excel flub cost your company? When you do make a mistake, how likely are you to catch it?

Though we often work with specialized software, Excel is the central tool for most of us, easily adaptable and giving us results faster than much more complicated and targeted programs. However, the danger for material error is there, and the bad news is it's well-nigh impossible to escape error. Even worse, Excel errors can cause large financial damage: at the European Spreadsheet Risks Web site, they've got a page of spreadsheet error horror stories. As an example: a Canadian power company took a 24-million-dollar loss in 2003 when a cut-and-paste error led to a mispriced bid. This is not a one-off event: as of September 2007, the EUSPRIG has 89 news stories, dating back to 1995, of substantial spreadsheet errors. Minor mistakes don't show up in newspapers.

Of course, we're experts, so our spreadsheet errors are rare—right? Let's go to the research and see.

Ray Panko, a professor at the University of Hawaii and a researcher into error rates in spreadsheets, has found that in audit research of spreadsheet errors, 94 percent of spreadsheets reviewed had errors, and about 5 percent of cells in the reviewed spreadsheets contained errors. Some of these errors are immaterial, such as minor typos, but the most insidious type of errors are the ones least likely to be found: omission errors, where something is missing; and logic errors, where the model or calculation is just plain wrong. We are very unlikely to discover the errors in our own thinking (the source of logic errors), and it's very hard to see that something is not there without explicitly looking for it.

We may think, "Sure, those studies show high error rates, but that's because they're looking at the spreadsheets of a bunch of schmoes ... most likely MBA students." OK, yes, some of the research subjects were MBA students, but in controlling for expertise level, error rates were similar for novices and experts. Even when the spreadsheet task was greatly simplified, the cell error rate was 2 percent for a very artificial situation, as opposed to operational spreadsheets from real businesses. Spreadsheets used in business often were much more complicated, involving links to other files and macros doing a great deal of the calculations.

The reason for novice error rates is simple: they don't know what they're doing. But what about us experts? The problem there is we may underestimate the likelihood of error.

It's hard to detect what you don't expect. If you do only the most cursory of error checks, because you are confident about what you've done, it's highly unlikely that you'll discover that you set up your model incorrectly. Someone else may not catch the error because they don't know enough about the spreadsheet to understand when you've made a mistake. So it's a bind: other people might be more inclined to consider the possibility of an error, but they are unable to find it from ignorance; you would be able to find it, but you're overconfident about your work.

The good news is that there are ways to mitigate the errors, and to reduce your chances for error. At the end of the article, I've provided links to resources I've found to be very helpful. The most helpful of all the articles is Philip Bewig's article "How do you know your spreadsheet is right?" If you do nothing else, check that one article out. Some of the sites and articles focus on error rates and types of error; don't discount these—if you know how material errors are most likely to occur, you're more likely to catch them or prevent them.

From reading these papers, there seem to be three main ideas that greatly reduce error rate initially and improve error detection:

- Think before you create. Plan your model structure in advance, and consider extreme values that should break your model (you can use those for testing later). Work out the logic in advance, not on the fly. This will result in better structured spreadsheets and reduce the likelihood of logic errors.

- Expect errors as you work. In lab research, when people were made aware of how common spreadsheet errors were, they were much more likely to catch their errors, especially material ones. People who expect errors examine spreadsheets more carefully, perform more stress tests, and make error-checking part of their routine. Keeping your "spreadsheet ego" in check is a must.

- Work in groups. Research has found that there is a great improvement in error detection when spreadsheets are reviewed in groups. We can be blind to our own errors but very able to see the mistakes of others. As well, different people may be apt to find different types of errors, so that in combination they improve the overall error-correction rate. Panko's research has found a statistically significant improvement in error detection when people work in groups of four (a two-thirds improvement in the error-detection rate); working in pairs did not improve detection to a significant extent.

The sources listed have even more practical and technical ideas (data validation, cell protection, named ranges, R1C1 notation, and more), but the central concept is to be mindful and to be humble.

Spreadsheets have become part of the quantitative sea we swim in—let's make sure they're our Queen Mary, and not our Titanic.

Helpful tips:
"How do you know your spreadsheet is right?" *http://www.eusprig.org/hdykysir.pdf*

"52 ways to prevent spreadsheet problems" *http://www.mailbarrow.com/services_excel_prevent.php*

"Block that Spreadsheet Error!" *https://www.aicpa.org/PUBS/jofa/aug2002/callahan.htm*

Other Sources:
European Spreadsheet Risks Interest Group: *http://www.eusprig.org/*

Ray Panko's Spreadsheet Research: *http://panko.shidler.hawaii.edu/SSR/index.htm*

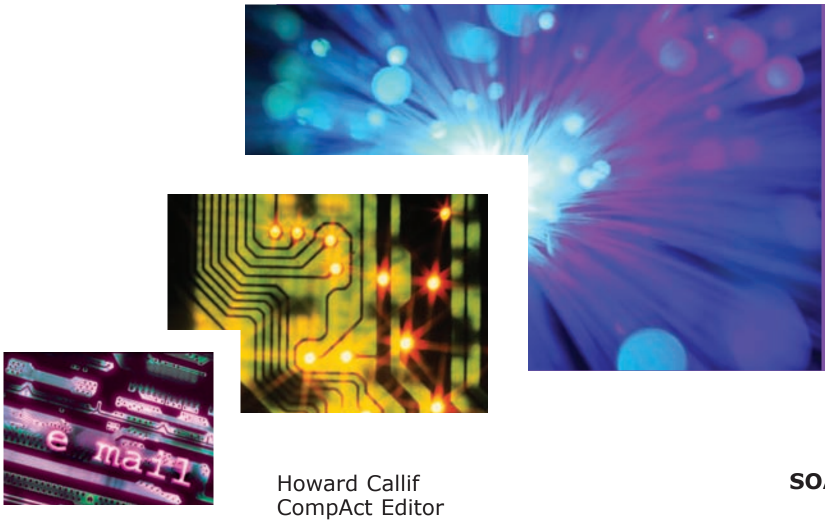
Other spreadsheet error news stories: *http://www.eusprig.org/stories.htm*

Ray Panko, "What We Know About Spreadsheet Errors", January 2005. *http://panko.shidler.hawaii.edu/SSR/Mypapers/whatknow.htm*

Tuck School at Dartmouth: Spreadsheet Engineering Research Project *http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs.html*

Systems Modelling Ltd., Spreadsheet Research Resources *http://www.sysmod.com/sslinks.htm#Research*

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



Technology Section Newsletter
Issue Number 29
October 2008

Published quarterly by the
Technology Section of the Society
of Actuaries

World Wide Web: www.soa.org

Howard Callif
CompAct Editor
iPipeline,
10600 N. Port Washington Rd.,
Mequon, WI 53092
e-mail: howard@callif.org

Technology Section Council
Kevin Pledge, **Chairperson**
Tim Pauza, **Vice Chairperson**
Joseph Liuzzo, **Secretary/Treasurer**

Council Members
Van Beach, **Web Site Coordinator**
Carl Nauman, **Scenario Generator**
Timothy Rozar, **Communications Coordinator**
Carl Desrochers, **2008 Annual Meeting program Comittee Coordinator**
Holly Loberg
David Minches

Other Volunteers
Howard Callif, **CompAct Editor (Web Technologies)**
Robert LaLonde, **2008 Spring Meeting Program Committee Coordinator**
Paula Hodges, **Past Section Chairperson and Education Subcommittee Coordinator**

SOA Staff

Staff Partner
Meg Weber
mweber@soa.org

Section Specialist
Susan Martz
smartz@soa.org

Staff Editor
Sam Phillips
sphillips@soa.org

Graphic Designer
Angie Godlewska
angie@kredodesign.com

Facts and opinions contained in these pages are the responsibility of the persons who express them and should not be attributed to the Society of Actuaries, its committees, the Technology Section or the employers of the authors. Errors in fact, if brought to our attention, will be promptly corrected.

SOCIETY OF ACTUARIES

475 N. Martingale Road Suite 600
Schaumburg, Illinois 60173
www.soa.org