ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

LETTER FROM THE CHAIR

by Tim Deitz

I am pleased to introduce myself as the new chairman of the Technology Section Council. Last year, I served as the vice-chair in preparation for this year. I feel we have a great group on the council and would like to take this opportunity to introduce you to our 2010 council and some of the projects they will be working on this year.

[full article]

EDITOR'S NOTES

by Howard Callif

Welcome to the new on-line version of CompAct! This exciting new format enables us to provide content in a much more accessible format than PDF.

[full article]

COMMON SOFTWARE DESIGN ISSUES

by Andrew Chan

When I developed my first business system 20 years ago, I was very proud that I took a business process needing multiple man-months to run and managed to migrate it to an automated computer process requiring only a few key strokes to execute.

[full article]

EUSPRIG MEETING FOCUSES ON DOING IT RIGHT THE FIRST TIME

by Mary Pat Campbell

This past July, EuSpRIG [European Spreadsheet Risks Interest Group] held their annual meeting in Paris, under the theme "The Role of Spreadsheets in Organizational Excellence."
[full article]

HPC SERVER REDUCES COSTS OF ACTUARIAL MODELING

by By Windows In Financial Services

David Dorfman, a specialist in computational modeling for Microsoft, recently shared his thoughts on using HPC Server to improve performance and reduce actuarial modeling costs.
[full article]

MODELING EFFICIENCY RESEARCH PROJECT: THE ACADEMY NEEDS YOUR HELP!

by the American Academy of Actuaries' Modeling Efficiency Working Group

Over the past 10 to 15 years, the number of projections mandated for reporting has increased tremendously. Future demands after the adoption of PBR will be even greater. The Modeling Efficiency Working Group (MEWG) of the American Academy of Actuaries is searching for techniques to make these requirements more manageable.
[full article]

R CORNER–FUNCTIONS

Steve Craighead

To design your own functions in R, you will need to follow a simple format ...
[full article]

Actuaries

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173

Risk is Opportunity.®

Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

LETTER FROM THE CHAIR

by Tim Deitz

I am pleased to introduce myself as the new chairman of the Technology Section Council. Last year, I served as the vice–chair in preparation for this year. I feel we have a great group on the council and would like to take this opportunity to introduce you to our 2010 council and some of the projects they will be working on this year.

Mary Pat Campbell is a first–year council member from Scor Re, headquartered in France. You may recognize her name from her frequent CompAct article contributions. Mary Pat will serve as our Webinar Coordinator. We have planned a couple of webinars for the upcoming year, but are constantly looking for ideas. If you have any good webinar ideas, please feel free to contact Mary Pat or me.

Carl Desrochers from Berkshire Life in Pittsfield, Mass. is in the third and final year of his term on the council. As he did last year, Carl will serve as the Other meetings coordinator. Carl will also help by serving as our Section liaison. Additionally, Carl will continue to head up the Speculative Fiction Contest.

Dave Kester is another first–year council member from SALT Solutions Inc., in Des Moines, Iowa. Dave is excited about invigorating our involvement in local actuarial clubs as the Actuarial Club program coordinator. We see this as a great

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,
Section Specialist

Sam Phillips, Staff Editor

opportunity to get involved with and share ideas with actuaries in a smaller group setting.

Lisa Lefkowitz is also a first–year council member. Lisa works at PolySystems Inc. in Chicago, Ill. She has many ideas to make the Technology Section Web site a better resource and will serve as our Web coordinator. With the recent SOA redesign of the Section Web pages, Lisa will work with them to make future improvements. If you have any ideas, please let Lisa or me know.

Holly Loberg is another third–year council member. She comes to us from Allianz Life Inc. Company of North America in Minneapolis, Minn. Holly served as the Web coordinator last year, working closely with the SOA on the Web page redesign. This year Holly will serve as the Secretary/Treasurer.

Tim Pauza is from the Philadelphia office of Ernst & Young. Tim is the outgoing Section chair and I want to thank him for all of the guidance he has given to me in preparing to take over the role and for the fantastic job he has done. This year, Tim will serve his third and final year on the council as the Annual Meeting coordinator and also manage our partnerships with various analyst groups.

Jeff Pomerantz is a second–year council member from Quantitative Risk Management in Chicago, Ill. Last year, Jeff did a great job as our Annual Meeting coordinator. There were a couple of very informative sessions with excellent speakers that Jeff pulled together for the meeting in Boston. This year Jeff will serve as vice–chair and the People coordinator.

Frank Reynolds is another of our second–year council members and is from the University of Waterloo. Last year Frank served as the Spring Meetings coordinator and will continue with that role again this year. Also, with his background in education, Frank will again serve as our education liaison.

There are a couple of others that need special recognition for service

to the council. Howard Callif is entering his third year as editor of the newsletter, but the role as editor is typically a two–year commitment! J. Eddie Smith has offered to co–edit CompAct, so for the rest of the year, Howard and Eddie will be co–editors. Phil Gold, of GGY AXIS in Toronto, ON, served this past year as the council's board partner and I want to thank him for that service. Our board partner this year is Jim Toole of MBA Actuaries Inc. in Winston–Salem, N.C. I look forward to working with Jim in this role. In closing, I would like to also thank the outgoing council members for their three years of service, Joe Liuzzo, David Minches and Carl Nauman. Thank you very much for your contributions and support!! I'm looking forward to the upcoming year and want to encourage all of the members to feel free to contact me with any ideas how we, as a council, can better serve you, our members.

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

EDITOR'S NOTES

by Howard Callif

Welcome to the new on–line version of CompAct! This exciting new format enables us to provide content in a much more accessible format than PDF. You can print each article using the "print" link on the top right corner, if you prefer. However, I think most readers will appreciate the ability to view articles interactively, as well as the ability to click on links and references. Please let us know what you think by clicking on the "poll" on the bottom left of the screen, or click on the "e–mail the editor" link and provide your feedback directly.

This edition has two articles touching on financial modeling. We have a new author submission as well, discussing some basics on programming, with very useful tips and suggestions. Again, R-Corner author Steven Craighead has also requested some feedback on the direction he should take in future articles. Please let us know whether you are finding the column useful, and what other information you would like to see.

Happy New Year!

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,
Section Specialist

Sam Phillips, Staff Editor

POLL

What do you think of the new CompAct format?

- I love it!
- It's better than the printed version
- The print version was better / more convenient
- I'm not sure - what's CompAct?

View Results vote
Share This

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

COMMON SOFTWARE DESIGN ISSUES

by Andrew Chan

When I developed my first business system 20 years ago, I was very proud that I took a business process needing multiple man-months to run and managed to migrate it to an automated computer process requiring only a few key strokes to execute.

However, my first business application had a lot of software design issues which made the system difficult to debug and maintain. Improper software design could reduce your development team's productivity and your actuarial system could become difficult to adapt to competitive market and statutory requirements.

I am going to discuss some of the more common design issues such as: blob class, copy–and–paste programming, spaghetti code, and Swiss army knife classes. I'll review their characteristics, problems and explain how you can avoid them.

Blob Class

The blob class is also known as the God class. It is a huge class that contains many attributes and operations. It is not a cohesive object and it often tries to serve multiple purposes. Its existence eliminates all of the object oriented design benefits, so it is difficult to reuse, modify and test. It can also compromise the scalability and performance of your system since it is huge and consumes a lot of system resource even for simple operations.

Copy–and–Paste Programming

This was one of my favorites 20 year ago, and describes the process

of copying code from one project to another as a form of "reuse." It is simple and easy to operate this way until you want to add a new feature or fix a bug. You have to spend a lot more effort to get the same work done across the whole system (i.e. identify the bugs, add new features, code review, and test), because you need to modify each system independently. Not to mention inserting useful comments in the right places every time—providing you can find all the pasted code!

Spaghetti Code

Spaghetti code has many forms, I will just go through some of the common ones:

- Long function—if you have the opportunity to read a function with more than 1,000 lines (believe me, they do exist), by the time you are half way through you may forget what it is supposed to do.
- Mystery variable name or condition—x1, x2, x3 or if (prem > prem1 && prem2 + prem3 > prem4). I wonder if anyone, including the original developer, can remember what they represent.
- Global variables–which seem very convenient! You can use or change a global variable anywhere, anytime. However, when you have to maintain or debug your system, and you have to find out why its value is suddenly changed and where the change comes from, the problem becomes clear. You may have similar problems if member variables in your classes are public. Note that this is also not scalable for classes that are used in Web systems.
- GOTO/BREAK—it is a developer's nightmare that the code suddenly jump from one place to another or just stop in an loop or break out from the function.

Spaghetti code is very difficult to reuse and update. Future maintenance can be costly, and fixing a bug may generate more bugs. In the end, you may find your development team spends more time fixing bugs than adding new features. It would eventually reach the point of diminishing returns to update or add to a system.

Swiss Army Knife

Swiss Army Knife class typically has a lot of data, functions and interfaces. It tries to provide a solution to every possible use of the class. Again, whenever something is big and complex, it becomes difficult to understand, modify and debug.

Solution

There is no simple solution. You have to understand SOLID object oriented (OO) design principles.

A class should be cohesive and have a single, clearly stated responsibility. If it has more than 60 attributes and operations then it is time to examine your class and refactor it. Simplicity is beauty!

It is every developer's job to constantly review and optimize the code. Refactor the code once a defect is identified. Eliminate global variables and write accessor functions for member variables. Remove obsolete code. Classes, functions, data types and variables must have meaningful names.

Every time you read or modify a piece of code, you have to understand what it does and ask yourself if it makes sense. Comments in code should explain what is being done, and why.

I will discuss SOLID OO design in more detail and demonstrate how modern development tools can help you refactor the code in another paper.

Andrew Chan is an independent consultant, and can be reached at andrew.chan@actuariallink.com

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

EUSPRIG MEETING FOCUSES ON DOING IT RIGHT THE FIRST TIME

by Mary Pat Campbell

This past July, EuSpRIG [European Spreadsheet Risks Interest Group] held their annual meeting in Paris, under the theme "The Role of Spreadsheets in Organizational Excellence."

From basic research in the sources of spreadsheet error to very concrete, practical tips for the daily user, to policy and auditing discussions, this year's conference adds to the already considerable resources produced by the members of EuSpRIG. Let me highlight a few of the presentations:

Keynote: Technical Standards for Modeling

This presentation was given by Deniz Sumengen, from the Board of Actuarial Standards [BAS] at the Financial Reporting Council. BAS was created after the Morris Review in the UK, and is tasked with setting up independent actuarial technical standards for that country.

In Sumengen's presentation, she highlighted an exposure draft on technical actuarial standards in modeling [TAS M], where modeling is defined very broadly, and is indeed a large part of actuarial work.

She noted there were a variety of general problems with models:

- Lack of testing
- Poor documentation
- Misunderstanding [of what the model covers, or what the

results mean]

- Over-reliance on an established view
- Unrealistic assumptions

Below, I draw out some of the main points from the draft exposure of TAS M:

- Documentation should be sufficiently detailed, include statement of purpose of the model, and be clear, unambiguous, and complete
- Models shall represent all phenomena relevant to their purpose
- Models shall be no more complex than can be justified
- Documentation shall include assumptions used in the model
- Model results shall be reproducible
- Checks will be constructed, performed, and documented to test theoretical, implementation, and end result issues
- Model limitations shall be disclosed

This is a rather robust, rigorous set of requirements. Of course, judgment on the part of the modelers plays a large role in these standards, but the principles are good for normal practice. The FRC is inviting comment, and some comment can already be seen on the current draft report. While this would apply only to the UK, I have been told by Sumengen's colleague Louis Pryor that they invite comment from anybody.

If nothing else, look over the report [they have the comments at the beginning, and TAS M itself can be found at the end of the document], and consider incorporating these practices in your own work. I think getting into these sorts of practices will definitely help with dealing with the more complex modeling that is becoming part of the standard actuarial toolkit.

Presentation | Exposure draft of TAS M

Self-checks and Controls in Spreadsheets

This presentation by Patrick O'Beirne focused on very concrete practices to check one's spreadsheets. These practices are:

1. Cross foot
2. Balance
3. Proportion

4. Multiple plus ungood
5. Room for expansion
6. Other sources of information
7. Expectations
8. Top 10 spreadsheet questions checklist

Let me talk about a few of these items. The first, cross foot, involves doing column sums and row sums on the same information, and making sure the overall total is equal for both. This is one of the oldest spreadsheet checks extant. He recommends having this crosscheck cell flagged with conditional formatting, so that the difference pops out to your attention if the difference is beyond a certain tolerance [the difference is unlikely to be zero, just from floating–point arithmetic issues].

The fifth item, room for expansion, relates to a common problem with formulas over ranges: what happens if you insert or delete cells in that range? Often there are issues of missed cells in sums because one has inserted new data at the beginning or end of the range [a problem, I'll note, that is caught by cross–footing.] O'Beirne recommends having sums start and end with empty cells, so if you insert cells/rows/columns at the beginning or end of the ranges containing numbers, Excel will properly update.

And the final item, is a checklist of questions, which I highly recommend. I would make an analogy to the preflight checklist pilots perform. Once you have this routine, you won't have to worry about particular issues being forgotten. Many professionals in other areas have complained about institutionalized checklists, as being demeaning of their great professionalism and intellect [pilots originally complained, and similar systems have become part of pre-surgery in hospitals, but not without complaint], but this has been a very effective tool in reducing operational risk.

Check out the links below to see descriptions of the other items in O'Beirne's list. I previously reviewed O'Beirne's book Spreadsheet Check and Control for CompAct, and these items do show up there. But if you want a free, short list of tips you can apply right away, check out O'Beirne's paper below.
Presentation | Paper

An Exploratory Analysis of the Impact of Named Ranges on the Debugging Performance of Novice Users
This paper, presented by Ruth McKeever, Kevin McDaid, and Brian

Bishop of the Dundalk Institute of Technology won the Student Prize from the conference, as the judges noted it was a " well–designed and thoroughly executed piece of research."

One of the simple good practices in spreadsheet design has been to use named ranges as opposed to opaque references as AC4 when building formulas. That's the conventional wisdom, and the experimenters set out to investigate this, as many spreadsheet best practices have been developed through individual experience and common sense, but no real scientific investigation. A small group of college students, who had been trained on spreadsheets the year previously, and who were given a little training on named ranges in Excel, were asked to debug a simple accounting spreadsheet. One group got spreadsheets using named ranges, and the other got one without.

The types of errors that had been entered ranged from non–material typos [e.g., misspelled header], rule violations [items contrary to written company policy], data entry errors [wrong numbers], and formula errors [wrong logic, wrong calculations]. In their results, they found little difference between the correction rate for the first three categories, but a noticeable difference for the final category of formula errors—the most serious type of error to occur in a spreadsheet, usually, and awfully common.

Those given the spreadsheets with named ranges found fewer formula errors than did the control group. The researchers posited a few explanations: high cognitive load [students did not develop the spreadsheets, and would have to keep checking the names and what cells they referred to], misplaced confidence in names [would do spot check, see expected named range, and move on without seeing error], or just plain lack of understanding of the error or how to correct it. Also, some of the range names were very long, and it could have been a function of poor naming conventions.

I cannot say that I am much surprised by the results. In previous research, different behavior of novices vs. experts has been shown when it comes to spreadsheet error and debugging. It would be interesting to see what the results were for experts, but it may require more complex spreadsheets in order to discover differences in debugging results.

There are limitations to this study, as freely noted by the researchers themselves, but it points out the important lesson that we should put our assumptions of risk management techniques [here, reducing spreadsheet error, as an operational risk] to the test. Paper

For more papers and presentations from the EuSpRIG conference, check out the group's Web site. You can find capsule reviews of the presentations at Patrick O'Beirne's site.

You can find the research papers at the archive site using the search term "eusprig," which will bring up this year's papers as well as papers from previous conferences.

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

This article originally appeared in the Spring 2009 issue of "Windows In Financial Services"

HPC SERVER REDUCES COSTSOF ACTUARIAL MODELING

by Windows In Financial Services

David Dorfman, a specialist in computational modeling for Microsoft, recently shared his thoughts on using HPC Server to improve performance and reduce actuarial modeling costs.

WFS: Why is the insurance industry facing increasing demands for computational modeling? DD: A number of factors are driving demand... new regulatory requirements, support for equity risk hedging and Enterprise Risk Management (ERM) programs and the increased emphasis on accurately assessing reserve requirements.

WFS: Can you elaborate on the regulatory demands? DD: Sure. Regulatory requirements for a principles-based approach to reserves are about to reach required implementation. The VACARVM actuarial guidelines become effective on December 31, 2009. Large variable annuity providers (VA) writers have been using a principles-based approach for determining capital requirements for several years, and many have invested in hardware to support these computationally intensive valuations. However, reserve calculations have to be done much more frequently and under tighter timelines, so the infrastructure in place in many organizations to support the C3 Phase II capital analysis will not be sufficient to support reserving.

WFS: Can you also give us more information on hedging program demands? DD: I think Ken Mungan, FSA, MAAA, and Principal at the Milliman Financial Risk Management Practice provides the best explanation: Research at Milliman has shown that life insurers hedging programs were 93% effective during the financial crisis. These hedging programs were implemented to offset risk exposures created by guaranteed minimum payments on variable annuities, a popular retirement savings vehicle. Hedge assets, owned by life insurers, are estimated to have generated approximately $40 billion of cash due to market declines. This capital strengthens life insurers at a critical time of financial turbulence. Life insurer hedging programs rely on large scale technology platforms and are extremely computationally intensive.

WFS: What are the demands in ERM programs and financial reporting? DD: The combination of data and programs required to build accurate corporate risk models creates a significant computational challenge. Companies that invested heavily in developing these programs need computational support to effectuate them. As for financial reporting, in today's economic climate, access to capital is a challenge and the need for computational models to accurately assess reserve and capital requirements is more critical than ever.

WFS: How does Microsoft's HPC Server (HPCS) help insurers meet these challenges? DD: HPCS combines previously isolated dedicated compute resources into larger combined compute clusters, efficiently providing support for multiple modeling applications. Hedging, ALM, seriatim valuation, stochastic projections, product pricing, cash flow testing, and other compute–intensive financial projections can all share a single, large cluster.Each application has access to a larger pool of shared resources, providing more flexibility to meet constantly changing business demands for modeling and simulation results at lower cost or faster time to solution than with separate smaller clusters.

WFS: How does HPCS help reduce costs? DD: HPCS can support all the required applications. In the past, typically, three different clusters and schedulers were required to run ALM, Hedging, and Policy Valuation. With industry-wide support for HPCS only one cluster may be required, and in comparison to other commercial schedulers, HPCS reduces software costs significantly. Insurers using HPCS can reduce operating budgets or invest these savings in building and running better models.

WFS: What do you see as the next cost–effective

improvement in risk modeling? DD: Service providers offering compute resources for investigative modeling on a massive scale. For example, Milliman currently offers this option to actuaries with constrained modeling resources. As other insurers gain experience with HPCS, this type of service will become attractive to more insurers.

David Dorfman is a Solution Specialist, High Performance Computing at Microsoft

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

MODELING EFFICIENCY RESEARCH PROJECT: THE ACADEMY NEEDS YOUR HELP!

by the American Academy of Actuaries' Modeling Efficiency Working Group

Over the past 10 to 15 years, the number of projections mandated for reporting has increased tremendously. Future demands after the adoption of PBR will be even greater. The Modeling Efficiency Working Group (MEWG) of the American Academy of Actuaries is searching for techniques to make these requirements more manageable.

We have already compiled a bibliography of modeling and technological methods that may produce statistically valid results while reducing the time and labor needed (see Actuary.org/risk). This list is not exhaustive, and we hope to add more useful information to assist practicing actuaries. We welcome information about additional methods.

MEWG needs your help to test these techniques with real data and on a variety of insurance products. We are looking for companies that are willing to run two sets of projections, one with a detailed model and all scenarios, and the other using your choice of scenario or modeling reduction techniques from the bibliography or from other sources. Perhaps you are already using an effective technique and would be willing to share it with your peers.

If your company agrees to participate, we need only a brief description of the technique used, along with a summary of the results and some basic specifications about the business modeled.

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,
Section Specialist

Sam Phillips, Staff Editor

For the data to be compiled, please view the response template on the MEWG Web page .

This is a voluntary survey. Do not provide any business confidential or proprietary information in response to this survey or any company-specific information without the company's permission. The Academy cannot receive such information or maintain its confidentiality.

This effort will complement the SOA research project, Analysis of Proposed Principle-Based Approach (LP157), conducted under the guidance of a Milliman research team. More information on the SOA project can be found here: SOA.org/research

For more information, please contact mewg@actuary.org (or visit the MEWG Web page on the Academy's Web site).

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 34 | JANUARY 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

R CORNER–FUNCTIONS

by Steve Craighead

Editor's note: R Corner[1] is a series by Steve Craighead introducing readers to the "R" language used for statistics and modeling of data. The first column was published in the October 2008 issue, and explains how to download and install the package, as well as providing a basic introduction to the language. Refer to each CompAct issue since then for additional articles in the series. The introductory article can be found on p. 24 of the October 2008 issue on the SOA Web site.

To design your own functions in R, you will need to follow a simple format:

```
functionname <- function( param1=default1,
param2=default2,...,paramN=defaultN)
{
# your function's calculations
return_value
}
```

For instance, a function of a single variable (with no defaults) that squares the input would be:

```
Square<-function(x)
{
x^2
}
```

You would use this function as
Square(4)
[1] 16

What is interesting about this function is that it can be processed against vectors or matrices (and other data structures).

Vector:
Square(1:10)
[1] 1 4 9 16 25 36 49 64 81 100

Matrix:
(M <- matrix(c(1:9),nrow=3))

	[,1]	[,2]	[,3]
[1,]	1	4	7
[2,]	2	5	8
[3,]	3	6	9

and

Square(M)

	[,1]	[,2]	[,3]
[1,]	1	16	49
[2,]	4	25	64
[3,]	9	36	81

So in a sense, what appears to be a function in a single variable, actually can be treated as having both multivariate input and output.

Now, if you wanted the function to have a default value, you would write your function as so:

```
Square2(x=0)
{
x^2
}
```

Now, if you invoke the function Square2 as follows, the default value will be used.

Square2()
[1] 0

Let's say that you need to do a large number of log–log plots of various data, You could create your own plot function by

LLplot<-function(x,y){plot(log(x),log(y))}

LLplot(1:10,101:110) would produce this graph:

I have been using the command area of R to create all of these functions, but as the function size grows, you will want to use a full screen editor on them. Just use the fix() function. For instance fix(LLplot), would display the function within the fix editor. I've already mentioned the fix() editor when we were examining how to manipulate data frames.

Let's revise your function so that your function would place a heading on the graph.

LLplot <- function(x,y,main="My Plot"){plot(log(x),log(y),main=main)}

Now if you just use the LLplot(1:10,101:110) as before, you will obtain the default header of "My Plot." However, if you invoke LLplot(1:10,101:110,"My Log Log Plot"), the header will use your specified header.

You have seen how to expand the input of a function to allow for multiple parameters. Now you need to see how to control the output of the function. Returning a list object as output in the last statement of the function does this. You specify the list object with both the name and the content of each output.

For instance, the function bellows squares your x values and returns the values in alpha and cubes the y values and returns them in beta.

```
Test<- function(x,y)
{
a<-x^2
b<-y^3
list( alpha = a, beta = b)
}
```

Now:

```
(answer <- Test(1:10,4:9))
$alpha [1] 1 4 9 16 25 36 49 64 81 100

$beta
```

```
[1] 64 125 216 343 512 729
```

To reaccess the values in beta, you would type answer$beta, or answer[[2]], which are basic access techniques for list objects.

Notice how the data format of x and y was maintained within the list object. If you used the above function on a vector for x and a matrix for y, the output list would contain a vector and a matrix.

All of the above examples are very simple, but you can enhance your functions by using conditional statements in them. For example, let's construct a function that squares all values less than 10 and cubes all values above 10.

```
Test2<-function(x)
{
if (x < 10) y <- x^2 else y<- x^3
y
}
```

If you test on a single value for x, the obvious transformation will be returned. However, if you used a vector input for x, with one value for x beginning greater or equal to 10, you will get the following result with a warning:

```
Test2(1:10)
[1] 1 4 9 16 25 36 49 64 81 100
Warning message:
In if (x < 10) y <- x^2 else y <- x^3 :
the condition has length > 1 and only the first element will be used
```

Also, note how the return value for 10 is 10^2 and not 10^3!

Now, let's change the function so that it maintains the same structure. Now, a conditional structure such as (x<10), is a structure that is the same as x, but the values of x are replaced by values of TRUE and FALSE.

```
Test2<-function(x)
{
(x<10)*^2+(x>=10)*x^3
}
```

The above uses conditionals on the index of x. When the conditional index value (e.g., (x<10)) is true, all values will be TRUE (effectively = 1) or FALSE (effectively = 0).

Testing Test2 on a matrix, you will see:
(x<-matrix(c(5:22),nrow=3))

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	5	8	11	14	17	20
[2,]	6	9	12	15	18	21
[3,]	7	10	13&	16	19	22

Test2(x)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	25	64	1331	2744	4913	8000
[2,]	36	81	1728	3375	5832	9261
[3,]	49	1000	2197	4096	6859	10648

I have barely scratched the surface on how to use functions. If you would like to learn more, please read Chapter 10 in "An introduction to R" in the R environment. Access this by choosing the "Manuals (in PDF)" feature under the "Help" option dropdown list.

1Craighead, S. (2000), "Insolvency Testing: An Empirical Analysis of the Generalized Beta Type 2 Distribution, Quantile Regression, and a Resampled Extreme Value Technique," ARCH, pp. 13–149.

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org