

ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES
Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)
[Letter From The Chair](#)
[Editor's Notes](#)
[SOLID Object Oriented
Design](#)
[Rand-dumb or Random?](#)
[Web 2.0 Offers Many
Opportunities for
Actuaries](#)
[R Corner: A Toy Copula
ERM Model in R](#)
[Number Puzzle !\[\]\(95b425611cbd2b8716a140cf67c81822_img.jpg\)](#)
[Last Issue's Number](#)
[Puzzle Solved !\[\]\(4f6bf54ae7e4144a72d78316053e412d_img.jpg\)](#)
[Articles Needed](#)
[SOA 2010 Elections:
Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section](#)
[Web site](#)


Share



Print Article

Search
Back issues

LETTER FROM THE CHAIR

by Tim Deitz

I've been reflecting on Technology for the past few days, in trying to think about writing for this issue of CompAct. I guess I'm a middle-aged actuary, entering the profession in 1985. There are many actuaries older than me and certainly many younger than me.

[\[full article \]](#)

EDITOR'S NOTES

by Howard Callif

I am pleased to announce that this edition of CompAct was co-edited by Eddie Smith. Eddie brings energy and enthusiasm to all of the projects he works on for the Technology Section, so I am very happy is joining me.

[\[full article \]](#)

SOLID OBJECT ORIENTED DESIGN

by Andrew Chan

A lot of systems were developed using an Object Oriented (OO) programming language, e.g. C++, Java, C# or Visual Basic .Net. Why did so many system developers choose to use OO programming languages? What are the benefits of OO programming languages?

[\[full article \]](#)

[Council](#)

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

SOA Staff
[Meg Weber, Staff Partner](#)

[Sue Martz,](#)
[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

RAND-DUMB OR RANDOM?

by Mary Pat Campbell

First: you have no business doing Monte Carlo modeling in Excel. Clear? Fine. Now that that's out of the way, we're using Excel all the time for inappropriate things, so what's a few random trials between friends?

[\[full article \]](#)

WEB 2.0 OFFERS MANY OPPORTUNITIES FOR ACTUARIES

by J. Eddie Smith, IV

Two extremes bound the gradient of strategies you can take in this new era: you can ignore it all, or you can try to take it all in. Neither approach is in your best interest.

[\[full article \]](#)

R CORNER: A TOY COPULA¹ ERM MODEL IN R²

by Steve Craighead

A simple ERM model entails setting up various risk sub-models and creating a dependency relationship between these risks. After this is accomplished all one needs to do is to simulate for a given number of trials (say 1,000) and aggregate the dependent risk values.

[\[full article \]](#)

NUMBER PUZZLE

by Marc Whinston

A fun, challenging puzzle for the readers of CompAct.

[\[full article \]](#)

LAST ISSUE'S NUMBER PUZZLE SOLVED

by Marc Whinston

The solvers of last issue's puzzle are: Gregory Dreher (Conseco); Andy Gennarelli (Centene Corporation); Paula Hodges (Allstate); Ken Longerman (Modern Woodmen); Lee Michelson (Office of the Insurance Commissioner, Washington); Tim Swankey (Disability Management Services); and Cameron Williams (Segal).

[\[full article \]](#)

ARTICLES NEEDED

by CompAct Newsletter

Your help and participation are needed: Articles, book reviews and other information on technology useful to actuaries are needed!

[\[full article \]](#)

SOA 2010 ELECTIONS: LET YOUR VOICE BE HEARD!

The SOA 2010 election is just around the corner. Online voting will be open 24 hours a day from Aug. 9 through Sept. 3 at 5:00 p.m. Central Daylight Time.

Visit SOA.org/elections to learn more about candidates for SOA president-elect, vice president, board of directors and section council positions-and to vote.

Have questions? E-mail elections@soa.org.

[\[full article \]](#)



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

ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES

Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)[Letter From The Chair](#)[Editor's Notes](#)[SOLID Object Oriented
Design](#)[Rand-dumb or Random?](#)[Web 2.0 Offers Many
Opportunities for
Actuaries](#)[R Corner: A Toy Copula
ERM Model in R](#)[Number Puzzle !\[\]\(248b91fcdac4810ffd15cf33fb6aec6f_img.jpg\)](#)[Last Issue's Number
Puzzle Solved !\[\]\(899d8b7697d64725bf017d3296cfcf1b_img.jpg\)](#)[Articles Needed](#)[SOA 2010 Elections:
Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section
Web site](#)

Share



Print Article

Search
Back issues

LETTER FROM THE CHAIR

I've been reflecting on Technology for the past few days, in trying to think about writing for this issue of CompAct. I guess I'm a middle-aged actuary, entering the profession in 1985. There are many actuaries older than me and certainly many younger than me.

I recently had an interesting conversation with my 17-year-old son regarding technology. For him, it's hard to comprehend a world without TIVO, DVDs, iPods, cell phones, PCs, etc., but I remember it vividly. If he wants to watch a television show, he will TIVO it and watch it when it is convenient, like after homework is done, watch it online or download it to his iPod. I remember you actually had to schedule your time, so you could watch a show when it came on, or you didn't see it. The family gathered around the TV to watch *Rudolph, the Red-Nosed Reindeer* when it came on, not record it or watch it on tape or DVD. There weren't even all of the cable channels that constantly show re-runs.

My first actuarial position was in the financial reporting area. I had to do some work on green sheets (if you don't know what these are, ask an older actuary or accountant). We had shared PCs, you remember the dual-floppy ones, where you put the Lotus 5.25" program disk in drive A and your data disk in drive B. Windows were something you looked out of or opened, if you wanted some fresh air. Just think about how far we have advanced in the past 25 years. Do you ever think how much farther we will advance in the next 25 years?

In this financial reporting position, I was trained to type certain

[Council](#)

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

[SOA Staff](#)
[Meg Weber, Staff Partner](#)

[Sue Martz,](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

numbers from mainframe generated reports into spreadsheets, take those totals and type them into other spreadsheets and on so on. Basically, using the spreadsheets as adding machine tape (again, ask someone if you don't know what this is either). Fortunately, the person who previously did this job was still in the department, so he could transition the responsibility to me over a couple of quarters. As I learned what the process really was, I began to question it. This seems really tedious, isn't there an easier way? This question was basically ignored. When the work had been totally transitioned to me and I fully understood the process, I decided to simplify it. I was determined to streamline (I'm not sure if that was a term back then) it a little bit each quarter. If I tried to do too much at once, I risked the chance of getting shot down and being told to do it the way we always did it in the past. I decided to make slight changes each quarter that I could easily explain and "sell" to my supervisor.

Each quarter, I would transfer some green sheets to spreadsheets, combine or link spreadsheets, etc. The biggest hurdle was getting the information from the mainframe reports directly to my spreadsheets. This meant learning Easytrieve Plus, creating a report of only the data I wanted, dumping it to a data file and linking it to my spreadsheets. This was not an easy process, because the IT department considered themselves a "closed shop." I could request a data dump like I needed, but they would prioritize it and schedule it, etc. In other words, don't hold your breath! Fortunately, I had an "in." My wife was a systems analyst. The area she worked in couldn't help me, but she was friends with those programmers who could and one of them was more than happy to help me. He realized that if he shared his knowledge, it would help the company in the long run and he wasn't threatened by me learning and doing this.

To wrap up my story, eventually, at quarter close, I pushed a button and went home. I came in the next day and checked on my final spreadsheet. The entire process was automated! The rest of the department was working late nights, but not me. I utilized the available technology to my advantage. This allowed me more time to analyze and understand results, as well as take on further work responsibilities.

There are a couple of points I hope you get from this story. First, always push technology! If a process seems tedious or inefficient, question it, someone else probably has, too. If you think there is a better way, there probably is, you just need to discover it. You may be able to do this yourself or you may need to enlist the help of others. The rapid exchange of information over the Internet and the

ability to network , allows us to quickly find others who are posed with similar issues. Work this to your advantage, use LinkedIn connections to share and learn. We all need to take advantage of this technology to work "smarter" or more efficiently.

This brings me to my second point. The Actuarial department and IT department need to work together and not feel threatened by each other. In my experience, I have generally seen tension between these two areas of insurance companies and there does not need to be. However, this is sometimes easier said than done. For the past two years, and again this year, the Technology Section has hosted sessions at the Annual Meeting centered around this relationship (Actuarial and IT Departments: Making the Marriage Work). I encourage you to attend this year's session. It is important that we, as actuaries, make the effort to make the relationship to work.

Council elections are coming up and we have some excellent prospective council members who have volunteered to take a leadership role and be on the ballot. The SOA election process is approaching, so pay attention to your emails, learn about the candidates and make sure you vote. I want to thank our volunteers for their willingness to get involved, all of our current council members and friends of the council who work hard to keep our section going! The section council and friends, continues to make progress on several important initiatives. We are developing materials for upcoming meetings, soliciting software reviews for actuarial systems, researching webcast offerings in technologies of interest to our membership, and publishing this quarterly newsletter.

In closing, I would like to solicit any ideas from the membership. If you have any ideas, comments, concerns, or want to get involved, please let us know. The council members are listed inside the front cover of this newsletter, so feel free to contact any of us.

Tim Deitz
SOA Technology Section Chair - 2010

Tim Deitz can be contacted at Tim.Deitz@milliman.com.



ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES
Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)
[Letter From The Chair](#)
[Editor's Notes](#)
[SOLID Object Oriented
Design](#)
[Rand-dumb or Random?](#)
[Web 2.0 Offers Many
Opportunities for
Actuaries](#)
[R Corner: A Toy Copula
ERM Model in R](#)
[Number Puzzle !\[\]\(e9474ce1d70442456f8fe9c393ea149c_img.jpg\)](#)
[Last Issue's Number
Puzzle Solved !\[\]\(e3f255517d37bb309a3a931ec4849e6a_img.jpg\)](#)
[Articles Needed](#)
[SOA 2010 Elections:
Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section
Web site](#)


Share



Print Article

Search
Back issues

COMPACT EDITORIAL - JULY 2010

I am pleased to announce that this edition of CompAct was co-edited by Eddie Smith. Eddie brings energy and enthusiasm to all of the projects he works on for the Technology Section, so I am very happy he is joining me. Although he has contributed several articles before (including an article introduction "Web 2.0" in this edition), we thought it would be a good idea to provide a short "introduction," which is below.

The "Letter from the Chair" struck a chord with me. I am approximately the same age as Tim, and was shocked to hear he classified himself as "middle aged." I did feel the same way about the issues he raised regarding technology; but my first reaction was to check Wikipedia for the definition of "middle aged," because I was sure I had leeway to not be in this age range. Unfortunately for me, I found that I can argue to be at the lower end of the range, but there is no doubt that I am in the range. I can rationalize not having IM (I need to concentrate, and prefer talking to people), and ignoring Facebook, Twitter, and "Google Wave" (I have real work to do), but I do need to consider the reality that Tim has a point, and I may be doing a disservice to myself by ignoring these mediums.

That is where I hope CompAct can help. People like Eddie can help me keep up, and quickly bring me up to speed on important new technologies and techniques. We have been publishing "introduction" articles on a variety of topics, and this edition is no different. We hope you find the information relevant, and useful.

There are also several "Letters to the Editor" that we would like to

[Council](#)

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

[SOA Staff](#)

[Meg Weber, Staff Partner](#)

[Sue Martz,](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

present (edited for the newsletter). Please feel free to submit your comments and questions!



INTRODUCTION TO EDDIE SMITH

I would like to introduce myself as the newest co-editor of CompAct. Howard Callif has been doing a fantastic job with CompAct for years now, and I would like to help him continue to make it a great resource for SOA members interested in using technology to improve their work.

I am an assistant actuary with RBC Insurance, and I also work with The Infinite Actuary to instruct portions of two of the Canadian Individual Life and Annuity exams offered by the SOA. Though not Canadian by birth, I consider myself a "Canadian by profession," as I work heavily with Canadian GAAP in all of my professional roles.

My primary technology interests include actuarial/IT relations, helping actuaries find practical ways of achieving complex modeling tasks, social and collaborative (Web 2.0) technologies, and using technology to increase workflow efficiency. I hope to contribute future articles in these areas, but more importantly, I hope to find other members interested in writing articles on any technology topic that can impact actuarial work—or be of practical interest to the actuarial community within the SOA.

The new HTML version of our newsletter offers many possibilities. We've only scratched the surface in the first few issues. If you have feedback or ideas on how we can make our Web-based newsletter better and take advantage of this new medium, please let me know.

I look forward to working with Howard and meeting past and future CompAct contributors. With each passing year, actuarial work relies more and more on complex technology. This trend will continue indefinitely. I believe that it's critical for the SOA to have a quality Technology Section, and CompAct should continue to be a valuable resource for sharing technology knowledge among our members.

J. Eddie Smith, IV, FSA, MAAA

J. Eddie Smith, IV, can be contacted at j.eddie.smith.iv@gmail.com.

Letters to Editor

FROM: CHUCK RITZKE

You asked for interest in articles on VB.NET ...

I may be the only one interested (and I think we discussed this once when you were looking around) and I realize this is actually asp.net, but I would be interested in a primer on Web Services. How to set them up, how data gets passed back and forth, etc. I have some potential applications where I think Web Services would be the way to go (like Web-based illustrations and similar apps). In perusing Visual Studio documentation (very briefly), I was particularly confused about how arrays or objects (if possible) would typically get passed between the service and client.

Also, maybe a personal Web Servers article? I prefer to write apps for the Web, but in some cases (like illustrations that I know you are very familiar with), there is still a desire to have desktop versions. I don't want to be mired in maintaining desk-top apps for the various operating systems, etc. It seems to me, life would be easier if we could run Web-based applications on the desktop. Would it be practical to build an application installation that installs a personal server (if none was already installed) so that users could run your Web-apps on their desktop? What are the pitfalls, issues, etc? What free servers could be installed? What considerations/complications for different operating systems? Is this approach practical?

Lastly, primer articles on using .net (if that is possible) for apps on I-pods, I-pads, BlackBerries, etc, either downloaded apps or again Web-based apps? What are the issues/differences, etc?

I know there is tons of stuff about this on the Internet, but I think you'd be easier to understand and the actuarial background helps.

Editor Answer: Thanks for the ideas! We are not sure we will be able to find people to write the articles. We can publish the request, and see if we can get anyone to volunteer to write on any of the topics.

If you have any knowledge on the above topics, please submit it so we can provide useful information to our readers!

FROM: Yue Lu

Hi Tim Rozar,

I read your article "Calculation of Generalized IRR in Excel" published in CompAct issue no. 27, in April 2008. I am very interested in the Becker's function and examples you discussed in your article. Could you please send me a sample workbook with the function and the examples? Thank you very much and have a nice day.

Editor Reply: The spreadsheet was sent, and we are very happy that we are a resource! Thanks again to Tim Rozar for his contribution!

FROM: Steve Welander

In your April 2010 CompAct editorial, you mentioned the possibility of solutions to the same problem in a variety of programming languages.

One way to accomplish this would be to issue a challenge on the Actuarial Outpost in the Software/Technology sub-forum. There are enough of us out there with quite the range of skills to accomplish the challenge in a variety of languages. Currently when someone poses a problem there, usually an excel question, it is non-infrequently answered with "here is how I would do this in R" or "... in SAS."

I think it would be a fun challenge and then someone could cull the "best" answers for the July (or whatever) issue.

I'm sure there would be a lot of Excel-based answers, along with R and SAS. But there are enough of us that have dabbled (or more) in C, FORTRAN, COBOL, APL, PERL, etc. to provide quite the array of answers. And you'd have the benefit of the answers being somewhat peer reviewed.

Editor Reply: Great idea! I will bring this up at our next Section call. Thanks for your input!

If you would be willing to work on this, please contact either Eddie or Howard. There are a lot of details to work out, and we want to gauge the level of interest in this type of endeavor. It is a great idea, but does require volunteers and thought to ensure success.

We appreciate everyone's feedback! Please submit any ideas, or comments to us. We will do our best to answer your questions, and provide content that helps you succeed. Thanks!

Howard Callif

Howard Callif can be contacted at howard.callif@gmail.com.



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

ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES
Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)
[Letter From The Chair](#)
[Editor's Notes](#)
[SOLID Object Oriented Design](#)
[Rand-dumb or Random?](#)
[Web 2.0 Offers Many Opportunities for Actuaries](#)
[R Corner: A Toy Copula ERM Model in R](#)
[Number Puzzle !\[\]\(07549ea8c24e6a9587f5e27f215997c7_img.jpg\)](#)
[Last Issue's Number](#)
[Puzzle Solved !\[\]\(bbcc5d2e6bfdea06264cef1b81418bd0_img.jpg\)](#)
[Articles Needed](#)
[SOA 2010 Elections: Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section Web site](#)


Share



Print Article

Search
Back issues

SOLID OBJECT ORIENTED DESIGN

by Andrew Chan

A lot of systems were developed using an Object Oriented (OO) programming language, e.g. C++, Java, C# or Visual Basic .Net. Why did so many system developers choose to use OO programming languages? What are the benefits of OO programming languages?

Without proper object oriented design (OOD), OO programming languages do not offer any significant advantages. To illustrate, consider the following scenario:

Let's say we had an old actuarial system that we developed using Visual Basic 6 that had 20 program files. At some point, we decided to migrate to the latest Visual Basic.Net and create 20 classes. Moreover, since we couldn't think of a good name for each class, we simply named them actuar01, actuar02 ... actuar20, the same name as each of the program files. We then copied the content of each program file into the corresponding class. In doing this, we created a system that uses OO programming language. But what benefits does our system have from this migration? Not a lot unfortunately!

In order to realize the true benefits of an OO programming language, we must understand the OO concepts and principles.

Object Oriented Concepts

There are a few object oriented concepts:

- Abstraction

Council

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

SOA Staff

[Meg Weber, Staff Partner](#)

[Sue Martz,](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

- Encapsulation
- Inheritance
- Polymorphism

You can find some very good definitions and explanations from Wikipedia. I have posted the hyperlinks and highlighted the summary for your reference.

Abstraction

Abstraction is "the mechanism and practice of abstraction reduce and factor out details so that one can focus on a few concepts at a time." Since abstraction extracts key characteristics of an object and hides other immaterial complexity, your readers should easily understand and visualize what we want to discuss.

"I just bought a new Samsung 46-inch 1080p 120Hz LCD TV," my friend told once told me. Size, number of lines, refresh rate and type are the key characteristics of an HDTV. He can go on to tell me about its physical dimension, weight, power consumption, etc., but most people don't care, and would prefer being told "I just bought a new TV," abstracting the technical specifications to hide unnecessary details.

Encapsulation

Encapsulation is "the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior; encapsulation serves to separate the contractual interface of an abstraction and its implementation."

Encapsulation allows programmers to use any method without understanding the details of implementation. This can drastically reduce the learning curve or maintenance effort of an actuarial system.

For example, how many actuarial programmers understand the implementation of ADO .Net? With ADO .Net, most of us can learn in a couple of hours how to write a simple function to retrieve data.

If your actuarial system consists of 1,000+ classes, do you want all your actuarial developers to understand every single class and every method within? It would take forever to train a new actuarial developer. Encapsulation will shorten the learning curve of actuarial programmers and will provide better system manageability and stability.

Inheritance

Inheritance is "a way to form new classes (instances of which are called objects) using classes that have already been defined.

Inheritance is employed to help reuse existing code with little or no modification."

When you derive a new reserve calculator from its base class, you only have to implement the new features. It can save you enormous development, checking and testing time.

Inheritance also increases overall system stability and reduces quality assurance effort. It allows developers to reuse code, enhance and modify existing class.

Polymorphism

"Polymorphism in the context of object-oriented programming, is the ability of one type, A, to appear as and be used like another type, B."

Inheritance is required in order to achieve polymorphism.

Polymorphism can greatly simplify coding and allow extensibility for future enhancements. With polymorphism, you can call 10 different reserve calculators from a single line of code. When you want to add five more new reserve calculators, you don't need to modify the calling function.

SOLID Object Oriented Principles

- Single Responsibility Principle
- Open Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

Single Responsibility Principle (SRP)

"There should never be more than one reason for a class to change."—Robert Martin, SRP paper linked from [The Principles of OOD](#).

Simple is beautiful. Each class should have only one responsibility and focus to do one single thing.

Your reserve calculator classes are already very sophisticated. If you also implement policy projection, decrement calculation and cashflow projection within it, then it would be huge and all your actuarial

programmers may always work on this big class together.

SRP would promote the reuse of code, clarity and readability. Your system would also be easier to test, enhance and maintained. Developers would also find less contention for source code files.

Open Closed Principle (OCP)

"Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification."—Robert Martin paraphrasing Bertrand Meyer, OCP paper linked from [The Principles of OOD](#).

Most of us work on existing systems rather than build new systems from scratch. When you add new features to a system, you often feel more comfortable adding new functions than modifying an existing codebase. Why? You worry that your modifications would accidentally add new bugs to the systems, especially fragile systems. OCP recommends extending existing codebase, not modifying it.

If you already have 10 different reserve calculator classes, adding a new one should not modify any existing code.

Once you have followed SRP to build your system, it would be easier to implement OCP. Systems following OCP are often more stable because existing code does not change, and new changes are isolated. Deployment is also faster because existing features would not be accidentally modified.

Liskov Substitution Principle (LSP)

"Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it."—Robert Martin, LSP paper linked from [The Principles of OOD](#)

This principle is just an extension of the Open Close Principle, and it means that we must make sure that new derived classes are extending the base classes without changing their behavior so that the derived classes must be completely substitutable for their base class. Otherwise the new classes can produce undesirable effects when they are used in existing program modules.

Below is a classic example of LSP:

```
public class Rectangle
{
protected int _width;
```

```
protected int _height;
public int Width
{
    get { return _width; }
}

public int Height
{
    get { return _height; }
}

public virtual void SetWidth(int width)
{
    _width = width;
}

public virtual void SetHeight(int height)
{
    _height = height;
}

} public class Square: Rectangle
{
    public override void SetWidth(int width)
    {
        _width = width;
        _height = width;
    }

    public override void SetHeight(int height)
    {
        _height = height;
        _width = height;
    }

}

[TestFixture]
public class RectangleTests
{
    [Test]
    public void AreaOfRectangle()
    {
        Rectangle r = new Square();

        r.SetWidth(5);
```

```

r.SetHeight(2);

// Will Fail - r is a square and sets
// width and height equal to each other.
Assert.AreEqual(r.Width * r.Height,10);
}
}

```

Square class is derived from Rectangle class; so C++ allows a Square object to be cast into a Rectangle object. However, Square class has its own setter functions; so r.SetWidth and r.SetHeight would set width and height equal to each other. What do you expect r.Width * r.Height to be equal to? Is r a Rectangle or Square object?

LSP would make the system easier to test and provide a more stable design.

Interface Segregation Principle (ISP)

"Clients should not be forced to depend upon interfaces that they do not use."—Robert Martin, ISP paper linked from [The Principles of OOD](#)

Again, it is another "simple is beautiful" principle. We should have multiple slim interfaces rather than a giant interface. Each interface should serve one purpose only.

If you have both policy month and calendar month projections, put them in two separate interfaces. For example, use IPMCashflowProj and ICMCashflowProj rather than just one interface named ICashflowProj.

With ISP, design would be more stable and flexible; changes are isolated and do not cascade throughout the code.

Dependency Inversion Principle (DIP)

"A. High level modules should not depend upon low level modules. Both should depend upon abstractions.

B. Abstractions should not depend upon details. Details should depend upon abstractions."—Robert Martin, DIP paper linked from [The Principles of OOD](#).

Low-level classes implement basic and primary operations, and high-level classes often encapsulate complex logic and rely on the low-level classes. It would be natural to implement low-level classes first and then to develop the complex high-level classes. This seems logical as the high-level classes consume low-level classes.

However, this is not a flexible design. What happens if we need to add or to replace a low-level class?

If your reserve classes (high level) contain cashflow classes (low level) directly, and you want to introduce a new cashflow class, you will have to change the design to make use of the new cashflow class.

In order to avoid such problems, we can introduce an abstraction layer between the high-level classes and the low-level classes. Since the high-level modules contain complex logic, they should not depend on the low-level modules. The new abstraction layer should not be created based on the low-level modules. The low-level classes are created based on the abstraction layer.

Once you implement DIP, your actuarial system will be significantly easier to extend, and deploying new features will take less time.

Conclusion

Please keep in mind that all of these principles are just guidelines that would make your system more stable, easier to maintain and enhance; but they are not ironclad rules. You must apply your own judgement and experience.

Andrew Chan can be contacted at chanpangchi@rogers.com.



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

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)

[Letter From The Chair](#)

[Editor's Notes](#)

[SOLID Object Oriented Design](#)

[Rand-dumb or Random?](#)

[Web 2.0 Offers Many Opportunities for Actuaries](#)

[R Corner: A Toy Copula ERM Model in R](#)

[Number Puzzle](#) 

[Last Issue's Number](#)

[Puzzle Solved](#) 

[Articles Needed](#)

[SOA 2010 Elections: Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section Web site](#)



Share



Print Article



Search Back issues



RAND-DUMB OR RANDOM?

by Mary Pat Campbell

First: you have no business doing Monte Carlo modeling in Excel.

Clear?

Fine.

Now that that's out of the way, we're using Excel all the time for inappropriate things, so what's a few random trials between friends?

Even if you're using a tool that isn't fully appropriate for Monte Carlo modeling in terms of efficiency, there's no reason to make the situation worse: to wit, using Excel's built-in pseudorandom number generating function RAND().

Below I will give a few reasons why using this function is a bad idea, as well as a few options to use in place of RAND(). Along the way, we will see some principles in choosing a pseudorandom number generator (PRNG) for various purposes, as well as pointing out some resources within this realm.

1. RAND() is a volatile function.

Volatile functions are ones that recalculate every time Excel recalculates. For mundane volatile functions such as TODAY(), this is not such a big deal, as they rarely change their results upon recalculation (and generally, you're only calling them a few times

Council

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

SOA Staff
[Meg Weber, Staff Partner](#)

[Sue Martz,](#)
[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

within a spreadsheet—not a million times).

But every time Excel recalculates, you get a different number resulting from RAND(). And the slightest thing will make Excel recalculate. Change a font? Excel recalculates. Hit enter in an empty cell? Excel recalculates. Sneeze violently? Well, if you accidentally hit a key Excel will recalculate.

This can be a big pain if you have a lot of calls to RAND() in your spreadsheet. Every single one of the cells that contains that function will recalculate.

To be sure, there are a few ways to get around this problem. First, you can set your calculation mode to Manual, in which case the spreadsheet is recalculating only when you request it to.

Second, you can copy your "random number cells" and paste as values over those very same cells. Of course, if you need more "random" numbers, you're going to need to put the formulas back in.

2. RAND() is not seeded.

What do I mean by a seeded PRNG? I mean that you give the PRNG a specific number, a "seed," which initializes the algorithm underlying the routine and will give you the exact same sequence of pseudo-random numbers in a particular order each time you use that particular seed.

This makes it easy to reproduce a particular Monte Carlo run without needing to store all the pseudorandom variates that go into producing your Monte Carlo estimate.

Now you may wonder why it's necessary to be able to reproduce your random variates that go into your estimate:

- a. Auditability—as Monte Carlo calculations become used for regulatory purposes, an auditor will need to be able to follow your work from author inputs to final numbers. This will be especially important as many of these processes will be new, requiring more detailed investigation on the part of auditors. Of course, many Monte Carlo estimates are purely for exploration or quick estimate purposes, and do not need to be audited.
- b. Sensitivity Analysis/Analysis of Change—many times, even for exploratory purposes, you would like to determine the impact of model assumption changes. Say that you use a

different lapse assumption for a variable annuity with a GMDB. If you don't use the same PRN (pseudo-random number) sequences, how do you know if variance of results are from the new assumption or a new pseudorandom sequence?

- c. Testing goodness-of-PRNG—it's difficult to test how "good" a PRNG is if you can't even rely on what the starting sequence will be.

What should you look for in a PRNG for Monte Carlo purposes?

1. Reproducibility (i.e., seeded)
2. Good statistical behavior
3. Ease of use with your system

I already explained why reproducibility is useful, and of course ease of use is just common sense. So what do I mean by "good statistical behavior"?

In Monte Carlo modeling, you are trying to estimate something—an expected value, a standard deviation, Value at Risk or Conditional Tail Expectation, histogram of the distribution—all those sorts of things. If there is bad behavior on the part of your PRNG, your results can be biased and your estimates off in particular ways. The most obvious is when the PRNG does not give a uniform distribution of numbers (which can be tested using a chi-square goodness-of-fit test). But it goes well beyond just checking uniform distribution.

Another issue is independence between consecutive PRNs, or triplets, or quadruplets, and so forth. If there are correlations between the numbers produced, then if you are using the PRNs for a sequential purpose (say, the random walk of stock returns), you can end up with biased results.

There is a suite of statistical tests that one can apply, called Diehard, that checks for issues like those above and so much more. I provide a link to a particular version, called the Dieharder suite, that is a very rigorous battery of tests. I also link to a paper about even stronger tests.

The problem is that most PRNGs will fail at least one of these tests. Is that a problem? Not necessarily. The problem would be if you're using these algorithms for cryptographic purposes—failing one of those tests means that a "randomized" crypto algorithm may be easier for someone to crack. With regards to Monte Carlo modeling, we don't need such strength—just something that gives us confidence

our results will not be biased.

So let's look at some alternatives to RAND():

- A. Excel's VBA function Rnd(), which is seeded. Of course, this requires coding some in VBA, but it's not too bad to implement. Problem: it's a black box (I've seen claims of knowing what code goes into this, but I've not pinned it down), but you can run the results through the randomness test suite Dieharder and see how it does for what you want.
- B. The free Excel add-in NtRand. This is based on the popular-for-MC-PRNG [enough jargon yet?] Mersenne Twister, and is fairly easy to use with Excel. What's more, it will produce correlated normal variates, which is pretty handy if you don't want to be doing a Cholesky decomposition yourself. NtRand can be found here: <http://www.ntrand.com/>
- C. Use your own code in VBA. You can find standard code for PRNGs at Numerical Recipes (yes, they're not in VBA, but it's relatively easy to take the FORTRAN code and do a transfer). As well, sometimes one finds PRNGs explicitly coded in spreadsheets online (such as the Academy's last release of the interest rate generator). Reusing other people's code is a long-time practice of mature programmers; just be sure you test it.
- D. Random.org—now, interestingly, this does not give you pseudorandom sequences—it's based on observed noise in the atmosphere. This is not easily used with Excel, though.

By the way, I do use RAND() all the time—to create simple illustrative calculations to show how various formulas work in a stochastic cash-flow model. It's fine for a quick simulation or two, but not serious Monte Carlo estimation.

I also found a RAND() substitute that was, if anything, worse than the original. I leave it as an exercise for the reader of this article to see what's bad about the following:

`=(NOW()*100000 - INT(NOW()*100000))`

Information about why the author of this formula thought this was a good idea resides at this link:

<http://support.microsoft.com/kb/44738/en-us>

If you'd like to argue in favor of this approach, buy me a drink, and I'll listen to your defense of that monstrosity. But I'm going to need

more than just one drink to sit still for it.

Thanks to the community at the Actuarial Outpost for discussion of PRNG issues in Excel.

Links:

[Diehard\(er\) tests](#)

[More on the Diehard tests](#)

[NtRand](#)

[Numerical Recipes](#)

Random.org: come on, you can guess this one

[On some difficult-to-pass randomness tests.](#)

Mary Pat Campbell can be contacted at
marypat.campbell@gmail.com.



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

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)

[Letter From The Chair](#)

[Editor's Notes](#)

[SOLID Object Oriented Design](#)

[Rand-dumb or Random?](#)

[Web 2.0 Offers Many Opportunities for Actuaries](#)

[R Corner: A Toy Copula ERM Model in R](#)

[Number Puzzle](#) 

[Last Issue's Number](#)

[Puzzle Solved](#) 

[Articles Needed](#)

[SOA 2010 Elections: Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section](#)

[Web site](#)



Share



Print Article



Search Back issues



WEB 2.0 OFFERS MANY OPPORTUNITIES FOR ACTUARIES

by J. Eddie Smith, IV

Editor's note: The following article is also being published in the July issue of Stepping Stone.

Two extremes bound the gradient of strategies you can take in this new era: you can ignore it all, or you can try to take it all in. Neither approach is in your best interest.

Even if terms like "status update" and "tweeting" make your skin crawl, it is becoming increasingly imperative that you at least understand what is happening and how the social landscape is changing. Finding your personal middle ground is key, and will depend on your personality, your needs and your goals in life.

It's okay if you feel lost in the blizzard of tech jargon brought on by the Web 2.0 era. There is a lot to keep up with, and new websites seem to appear daily. Knowing which to invest time in and which to ignore is a challenge, and to some degree, a trial and error process.

What The Heck Is Web 2.0, Anyway?

Consider compact discs (CDs). The first generation was read-only, meaning that you could access the information on them, but you could not add to or modify them. Web 1.0 was the read-only version of the Internet. As with the first generation of CDs, a very small minority had the means of contributing and creating content, while the majority only had the ability of consuming information.

[Council](#)

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

[SOA Staff](#)

[Meg Weber, Staff Partner](#)

[Sue Martz,](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

Later, CDs became read/write, and inexpensive technology was developed that allowed anyone with a CD burner to put information on CDs. Web 2.0, which began in 2004, is the read/write version of the Web. In this second generation of the Web, virtually anyone can add, edit and share content very easily and inexpensively. From an anthropological perspective, this is no small transformation. For much of human existence, information has been controlled and distributed by a very small subset of the population because of political factors and technological limitations. As journalist A. J. Liebling once noted, "freedom of the press is guaranteed only to those who own one." We are now entering an era in which everyone owns a press.

The early success of popular Web 2.0 media like Wikipedia, Facebook, LinkedIn, Twitter, blogs and message boards indicates not only the hunger people have for information but also their willingness to add and share information in decentralized systems. The number of sites classified under the Web 2.0 umbrella has grown so fast that the term is becoming increasingly too generic to be useful. And so many previously static sites have added features that allow for user interaction and contribution that it's all blurring into a single thing. Web 2.0 is the Web today.

The following statistics highlight the traction being gained by Web 2.0 sites:

- Social media sites are growing three times faster than the overall Internet in terms of "time spent" according to Nielsen in 2009.
- Six-year-old Facebook now has over 400 million users (exceeding the population of the United States). Half of them log into Facebook every day, and 35 million of them update their statuses every day.
- As of March 2010, users of YouTube upload 20 hours of video every minute, a rate that equals 12 centuries of video per year!
- Twitterers tweet at a rate of 50 million tweets a day, or 600 per second. This figure is up 1400 percent from 2009 to 2010.
- From LinkedIn's blog in October 2009: "LinkedIn has 50 million users worldwide and we're growing that figure at roughly one new member per second. When LinkedIn launched in 2003, it took 477 days—almost a year and four months—to reach our first million members. This last million

took only 12 days."

Actuarial Awareness of Web 2.0

In 2009, the SOA Technology and Financial Reporting Sections participated with the research firm Gartner to assess the relationship between section members and their IT staffs. One of the survey questions also inquired about the popularity of Web 2.0 tools.

Results ranged from close to half at least talking about implementing instant messaging, to less than 15 percent even having heard of crowdsourcing. Let's look at a few of these .

Crowdsourcing

If you have ever turned to an online message board to ask a question, you've used crowdsourcing. Crowdsourcing is the act of "outsourcing" a project or problem to a crowd, the members of which evaluate and ultimately offer solutions. The technology forum of the Actuarial Outpost is perhaps one of the best examples in the actuarial community. Almost daily, people go there to ask questions about Excel and a variety of other common (and not-so-common) topics.

If used wisely, the productivity gains from crowdsourcing can be great. First, there is no cost. Second, solutions to problems can be found quickly—sometimes in just minutes. And third, the person seeking an answer can work on other things while the crowd chews on the problem. It's like having free access to a legion of knowledge workers just down the hall from you. At no time in history has such a dynamic, immediate and real-time resource been available.

Wikis

Another popular Web 2.0 tool not highly recognized by respondents in the Gartner survey was the wiki. Perhaps the odd-sounding name is a distraction. Here's a little history .

In 1994, computer programmer Ward Cunningham wanted to make it easier for programmers to communicate with each other. He came up with the idea of a Web-based system wherein users could edit pages themselves. He installed the system on his company's website in 1995, and it became known as WikiWikiWeb, based on the Hawaiian word "wiki," which means "fast."

About five years later, two men, Jimmy Wales and Larry Sanger, were troubled by the lack of success of their project to create a Web-based encyclopedia called Nupedia. Their idea was to impose a strict set of rules for adding, reviewing and changing content in

Nupedia. Only experts (mostly Ph.D.s) were allowed to contribute. The model wasn't working. After 18 months and \$250,000 of expenses, Nupedia had only a dozen articles.

When Wales and Sanger learned about Cunningham's WikiWikiWeb concept, they decided to give it a try and launched a site called Wikipedia in January 2001. Unlike Nupedia, anyone could contribute to and edit Wikipedia. By the end of the first month, Wikipedia amassed 617 articles; by the end of the year, 19,000. Presently, Wikipedia has over 3.2 million articles in English alone. Nupedia and its ultimate collection of 24 articles went offline in 2003.

Wikis are so prevalent today, it's hard to imagine that they were practically nonexistent a decade ago. But in that time, they've been put to use in a variety of settings—from software documentation to enterprise collaboration to government intelligence.

Wikis can be useful to actuaries for a variety of reasons. Here are a handful:

Versioning. Wikis back up everything. If someone breaks a page or overwrites something, restoring is very easy. On the other hand, if you overwrite a Word file on your network, it's gone unless some backup process ran. Wikis back up every change.

User tracking. Want to know who changed your documented per policy expense assumption from 10 to 20? A wiki keeps up with who changed pages. This increases accountability, and also directs you to people with information you may need. For example, suppose I'm a new hire and I notice that that Joe Actuary usually updates interest rates. I know to go to Joe for more information on those rates.

Procedures openly available and readily searchable. Suppose Joe becomes ill and can't come into work for a while. I'm pretty sure that he has documented his procedures, but I have no idea where they are because they are most likely in a Word or Excel file stored on the network somewhere, or worse, stored on his hard drive or e-mail. If Joe has them in a wiki, I can go to a single search field and probably find his procedures in seconds.

Project tracking. Users can update their project status simultaneously in one, open location. No more sending those e-mails saying, "Whoever is in Spreadsheet ABC, please let me know when you're done so I can update it."

Cross-referencing. You can quickly link to past projects or integrate previously disparate areas. For example, maybe the valuation area

needs to add assumptions for a new term product. If the pricing area has the product specs stored in a centralized wiki, the valuation area can link directly to them and probably avoid retyping tons of information into individual files.

Completely searchable. Suppose new hires are coming in and need answers related to software or any procedure. These files are probably difficult for someone brand new to find. Folder structures are often intuitive only to those who created them. However, a wiki could connect someone who needs information with that information in a matter of seconds because the user doesn't need to know exactly where the information is; he or she just needs to do a keyword search.

Social networking

Even if you have no desire to engage in online social networking in your personal life, you will increasingly find it useful, if not necessary, in your professional life. For example, by participating in group discussions on LinkedIn, you can meet people who are interested in the same subjects. LinkedIn's core premise is the forging of connections in your network. A connection is made on LinkedIn when one person invites another to connect. If accepted, the two individuals can see each other's activity in a "newsfeed" appearing on each other's home page.

You may notice that one of your connections is working on or just completed a new project in an area you're interested in. These little bits of information in your newsfeed may seem insignificant, but they are actually a very powerful and efficient mechanism for joining people who need information with those who can provide it. It's a knowledge/worker matchmaking service of sorts. And it will prove to be tremendously beneficial for actuaries or any other highly skilled professionals whose numbers are not great and are dispersed throughout the world.

Additionally, LinkedIn and similar sites provide an unprecedented opportunity for interdisciplinary collaboration. By engaging in groups of interest to you, you will naturally learn from and connect to non-actuaries. This could be enormously valuable to the actuarial profession, one that has much to offer outside its traditional sphere of work.

It's very important to understand that you don't even have to be a "social" person in the traditional sense to benefit from social media sites. These new media cut through the customary face-to-face inhibitions that roadblock the same connections at traditional

meeting venues like annual meetings and symposia. What's more, sites like LinkedIn are 24/7, context-based networks that naturally evolve with you, your interests, and the time you invest in them. Once started, your network will work for you—even when you aren't online.

A well-developed, active network can be an enormous asset to a job seeker. Credibility is very difficult to fake on a site like LinkedIn. Put yourself in a hiring manager's shoes. Suppose you're evaluating two candidates. One of them sent you a one-page paper resume. The other has a LinkedIn profile, multiple connections (some of which you share), public recommendations and is also active in group discussions on LinkedIn. Who do you know more about?

The quickest way of finding other actuaries on LinkedIn is by joining actuarial groups. The SOA itself has an official group: "Society of Actuaries," as do many of the Sections, including Technology, Financial Reporting and Entrepreneurial Actuaries just to name a few. If you do a group search for "actuaries" on LinkedIn, you will find many, many more groups with actuarial members.

LinkedIn is certainly not the only social media site of professional value to actuaries. Twitter is another social network where actuaries have been emerging in the last two years. Twitter is a free service that allows you to publicly broadcast 140-character messages. The openness of Twitter allows you to connect with people that you would never find through a more context-based network like LinkedIn. Twitter has benefits beyond networking, including being one of the best sources for real-time news on virtually anything taking place in the world at any given time.

One of the quickest ways of finding actuarial Twitterers is searching for the #actuary hashtag at <http://search.twitter.com>. Hashtags are a way of tagging tweets to give them more context and they also make it easier to find people interested in that topic. Another way of finding actuaries on Twitter is by looking at the followers of popular actuaries like current SOA President Michael McLaughlin, who goes by @PrezMike2010 on Twitter. I wrote an entire article explaining the basics of Twitter in the October 2009 issue of *CompAct*; you may want to check that out to learn more about Twitter.

If you have questions on using social media or other aspects of Web 2.0, feel free to contact me or someone on the Technology Section council. We are interested in helping actuaries find ways of utilizing these new tools. You can also find more of my personal writing on these subjects at my blog: <http://risk20.wordpress.com>.

References

- Enterprise 2.0 by Andrew McAfee
- <http://www.wikipedia.com>
- http://youtube-global.blogspot.com/2009/05/zinks-20-hours-of-video-uploaded-every_20.html
- <http://blog.twitter.com/2010/02/measuring-tweets.html>
- http://en-us.nielsen.com/main/news/news_releases/2009/march/social_networks
- <http://blog.linkedin.com/2009/10/14/linkedin-50-million-professionals-worldwide/>

J. Eddie Smith, IV, FSA, MAAA, is an actuarial associate with RBC Insurance. He is also an exam instructor with The Infinite Actuary. He can be reached at j.eddie.smith.iv@gmail.com and @eddie_smith on Twitter.



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

ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES
Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)
[Letter From The Chair](#)
[Editor's Notes](#)
[SOLID Object Oriented
Design](#)
[Rand-dumb or Random?](#)
[Web 2.0 Offers Many
Opportunities for
Actuaries](#)
[R Corner: A Toy Copula
ERM Model in R](#)
[Number Puzzle !\[\]\(bfc7ba4486737525edb6839028ede2e8_img.jpg\)](#)
[Last Issue's Number
Puzzle Solved !\[\]\(7490bbd2236eba0faf0313a0183fefb3_img.jpg\)](#)
[Articles Needed](#)
[SOA 2010 Elections:
Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section](#)
[Web site](#)


Share



Print Article

Search
Back issues

R CORNER: A TOY COPULA¹ ERM MODEL IN R²

by Steve Craighead

Editor's note: R Corner is a series by Steve Craighead introducing readers to the "R" language used for statistics and modeling of data. See footnote 2 at the end of the article for information on installing and learning more about R.

A simple ERM model entails setting up various risk sub-models and creating a dependency relationship between these risks. After this is accomplished all one needs to do is to simulate for a given number of trials (say 1,000) and aggregate the dependent risk values. From these aggregated results one can then either determine VaR or conditional tail expectation (CTE) at a specific percentile. Below, we will construct our ERM toy model, which will demonstrate how to link statistical sub-models with copula dependency models using the R copula package.

The R copula package models the Frank, Gumbel and Clayton copulas within the Archimedean family, as well as, the multivariate normal and Student t copulas.

The Archimedean copulas are limited to dimensions less than seven since the resultant multivariate probability density function (PDF) is not available due to difficulty in symbolically differentiating the associated multivariate cumulative distribution function (CDF).

In brief, the steps that the reader needs to follow are:

Council

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

SOA Staff

[Meg Weber, Staff Partner](#)

[Sue Martz](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)

- A. Specify the copula that will model the dependence.
- B. Specify the multivariate distribution using the copula defined in step A by using the `mvdc()` function. You must also define the marginal distributions associated with each risk. You will also specify the marginal parameters which will be discussed below.
- C. Supply the marginal data associated with each risk.
- D. Fit the model.
- E. Examine the results.

Before we actually create the toy ERM model, we need to discuss some basics in R. The R language has many different univariate distributions available for modeling the marginals. A statistical distribution in R uses three separate functions for modeling (with the possibility of a fourth). For example, if you wanted to model a univariate normal distribution, you would use either the `dnorm`, `pnorm`, `qnorm` (and `rnorm`) functions. The `dnorm` function models the density, the `pnorm` models the distribution (CDF) and the `qnorm` is the quantile function, which is the inverse function of the CDF. The fourth function is `rnorm`, which can be used to generate random normal variates. Other example distributions functions are `dt`, `pt`, `qt` and `rt` for the Student t, and `dexp`, `pexp`, `qexp` and `rexp` for the exponential distribution. Each of these functions requires specific model parameters. The normal distribution has the model parameters mean and sd for the mean and standard deviation. The Student t distribution requires the parameter df for the degrees of freedom. The exponential distribution uses the rate parameter.

One must also load the copula package after starting R. This is done in Windows by choosing the copula package when using the Load Package option under the Packages option on the command list at the top of R.

When carrying out step B above, the `mvdc()` function creates a multivariate distribution object in R. This function has three major inputs. The first is the copula. The second is a list of the specific marginals and the third is a list containing lists of associated marginal parameters.

For instance, say you want to model a bivariate CDF using the Gumbel copula (with parameter of three) and the first marginal is a normal distribution with a mean of 10 and a standard deviation of two and the second marginal is an exponential with a rate of two.

A. First, you need to use the command

```
gmb<-gumbelCopula(3,dim=2)
```

to create the copula object. The symbol "<-" is used as the assignment operator. The "dim=2" assures that we are creating a bivariate CDF. You can specify up to six dimensions within the R Copula library.

B. Next you specify the bivariate distribution by using the command:

```
myCDF<- mvdc(gmb, margins=c("norm","exp"),
paramMargins=list(list(mean=10,sd=2),list(rate=2)))
```

Notice the first parameter in mvdc is the gmb object defined in step A. The second parameter is a generic list of the two marginal distributions (produced using the c() function). The third parameter is much more complex, where the list() function is used three times. Here we have a list made of two separate lists. The first of these two separate lists are the parameters that model the normal marginal as discussed above. The second of these lists contains the parameter required to model the exponential marginal. The list() function is not as generic as the c() function, but the mvdc function expects the marginal parameter defined in this fashion.

C. Next you need to supply the risk data associated with each marginal. Since this is a toy, we will actually use the myCDF multivariate distribution as defined to generate the data (a bit incestuous, but OK for this demonstration). Here you will use the rmvdc() function which will generate random variates from myCDF. Use the command:

```
x <- rmvdc(myCDF,1000)
```

to generate 1000 samples from your bivariate CDF. You could also supply your own data x as in an R matrix with 1000 rows and two columns. For instance, if you were able to use identical scenarios and produce separate various risk values based on these scenarios, and collect these together into an R matrix, you could then design your copula multivariate distribution and use the fitting algorithm to fit your actual data.³

D. Next, take the data in the matrix x and determine the best parameters in myCDF by using the fitting function fitMvdc() (which uses maximum likelihood). This function requires the x

matrix, the myCDF object and a generic c() list of initial guesses to the parameters. Note, the first parameter in this list will be the parameter associated with the Gumbel copula model (which we assumed as three above). The remaining values are associated with the parameters of the marginals. Suppose that you use the command:

```
Fitted<-fitMvdc(x, myCDF, c(3,9,1,1))
```

Here the three is the Gumbel parameter, the nine is the initial guess for the normal mean, one for the normal sd parameter and one for the exponential rate parameter. Here starting values were different than the original model, so that in Step E, we can see the goodness of fit. Note: you may get an warning message stating that there are so many separate warnings and that you should enter the command "warnings()" to display these. These warnings arise when the fitting algorithm moves the parameter estimates outside the acceptable values. At this time, you may disregard these.

- E. Once the model fitting has completed, you can examine the results of the fit by just issuing the command "Fitted." This will display to the screen the results. For instance the following is the results obtained by the author. Your results may vary due to the random values that were generated in matrix x.

Fitted

The ML estimation is based on 1000 observations.

Margin 1 :

	Estimate	Std. Error
m1.mean	10.157765	0.05885397
m1.sd	1.984802	0.03243711

Margin 2 :

	Estimate	Std. Error
m2.rate	1.893372	0.05849316

Copula:

	Estimate	Std. Error
param	2.819898	0.08826639

The maximized loglikelihood is -1787.369

The convergence code is 0

Observe both the relative parameters and their standard error, and how close the estimates are to the original parameters originally set up.

Also, one can examine the contour plot of the bivariate density function by using the following commands:

```
contour(myCDF,dmvdc,xlim=c(4,15),ylim=c(0,1.7))
title("Gumbel Multivariate PDF")
```

These produce the following graph:

One obtains the graph of the normal marginal by using these commands:

Gumbel Bivariate PDF Graph

```
K <- density(x[,1]) #x[,1] gets risk 1 results
K$call <- "Normal Marginal"
plot(K)
```

Normal Marginal Graph

To obtain the exponential marginal, use these commands:

```
K <- density(x[,2]) #x[,2] obtains risk 2
K$call <- "Exponential Marginal"
plot(K)
```

Exponential Marginal Graph

Note how that the PDF is not symmetric, which is a characteristic of the Gumbel Copula since it is only defined in the right tail.

Also, one can examine the contour plot of the bivariate cumulative function by using the following commands:

```
contour(myCDF,pmvdc,xlim=c(7,15),ylim=c(0,1.7))
title("Gumbel Bivariate CDF")
```

This produces the following graph:

Gumbel Bivariate CDF Graph

Of course, if you add up the random samples, you will have the total

dependent risks! By using these following commands, R will display the density of the total risks:

```
Y<- apply(x,1,"sum") #sum across columns of x
K<-density(Y)
K$call <- "Aggregate Risks"
plot(K)
```

Aggregate Density Graph

With just the above commands you may examine other copulas such as the Frank copula using the commands:

```
frank <- frankCopula(param=2,dim=2)
myCDF<-
mvdC(frank,c("norm","exp"),list(list(mean=10,sd=2),list(rate=2)))
contour(myCDF,dmvdC,xlim=c(4,15),ylim=c(-0.5,1.5))
title("Frank Bivariate PDF")
```

You will obtain the following density contour:

Frank Bivariate Graph

Other things that you can do, is to increase the number of dimensions and use other marginals. Also, you can use the `rmvdc()` command as we did in step C above and simulate your dependency multivariate distribution.

Though this toy model only uses two risks, you can use up to six risks, by changing the definition of the copula as well as specifying the different sub-models. Hopefully, the examples above are explicit enough to let you do various what-ifs on your own data.

Bibliography

R Development Core Team (2006). 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>.

Footnotes:

¹This entire article assumes that the reader is acquainted with both the concept of copulas in the modeling of dependence in ERM modeling and how to use the R language with statistical modeling. However, for further information on copulas, you may read the excellent survey article "Understanding Relationships Using Copulas" by Frees and Valdez in NAAJ Volume 2, Number 1 of

1998. If you prefer to experiment with copulas in Excel instead of R, please refer to the workbook and articles created by Sam Cox and Don Behan on <http://www.behan.ws>.

²The first column of R Corner 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](#). The R language is a very popular open source statistical modeling environment, which you can obtain from the webpage <http://cran.r-project.org>. If you are interested in building the toy model in this article, you will need to download the base R Binary for the operating system of your choice, (Windows, Linux or MAC). Also, you will need to download the contributed copula package. You can do this initially when you download the base system, or you can use the package installation facility when you start R.

³The import function "read.csv" allows you to import your own numerical data, but the "read.csv" function stores the data in as a data.frame object. You must convert your data object to a matrix object by using the function "as.matrix"

Steve Craighead can be contacted at steven.craighead@towerswatson.com.



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

ISSUE 36 | JULY 2010

SOCIETY OF ACTUARIES Technology
Section

CompAct

ELECTRONIC NEWSLETTER



CONTENTS

[Table of Contents](#)

[Letter From The Chair](#)

[Editor's Notes](#)

[SOLID Object Oriented
Design](#)

[Rand-dumb or Random?](#)

[Web 2.0 Offers Many
Opportunities for
Actuaries](#)

[R Corner: A Toy Copula
ERM Model in R](#)

[Number Puzzle](#) 

[Last Issue's Number](#)

[Puzzle Solved](#) 

[Articles Needed](#)

[SOA 2010 Elections:
Let your Voice be Heard!](#)

QUICK LINKS

[Technology Section](#)

[Web site](#)



Share



Print Article



Search
Back issues

ARTICLES NEEDED FOR COMPACT NEWSLETTER

Your help and participation are needed: Articles, book reviews and other information on technology useful to actuaries are needed! All articles will include a byline to give you full credit for your effort. An annual prize is awarded for the best article of the year. If you would like to submit or discuss an idea for an article, please contact [Howard Callif](#) or [J. Eddie Smith](#).

Submissions for future editions are also welcome!
Submission deadline is August 5 for the October issue.

Thank you for your ideas!

J. Eddie Smith can be reached at j.eddie.smith.iv@gmail.com.

[Council](#)

[Links of Interest](#)

[Fiction Contest](#)

[Howard Callif, Editor](#)

SOA Staff
[Meg Weber, Staff Partner](#)

[Sue Martz](#)

[Section Specialist](#)

[Sam Phillips, Staff Editor](#)



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