Creating a Great Loss-Model without Losing a Wink of Sleep

September 17, 2019
David Odell PhD.
Senior Statistician
Insureware

Creating a Great Loss-Model without Losing a Wink of Sleep

How simulations can help bring models and data together

Models for Loss Reserving

The problem is about matching data and model – making a productive relationship between them. There is a purpose, and it is so that you can make a forecast about future data. An important side-benefit is that a good model gives you insight into what's driving your data.

Mmm, I like that model!

153,638	342,050	476,584	564,040	624,388	666,792	698,030	719,282	735,904	750,344	762,544
178,536	404,948	563,842	668,528	739,976	787,966	823,542	848,360	871,022	889,022	
210,172	469,340	657,728	780,802	864,182	920,268	958,764	992,532	1,019,932		
211,448	464,930	648,300	779,340	858,334	918,566	964,134	1,002,134			
219,810	486,114	680,764	800,862	888,444	951,194	1,002,194				
205,654	458,400	635,906	765,428	862,214	944,614					
197,716	453,124	647,772	790,100	895,700						
239,784	569,026	833,828	1,024,228							
326,304	798,048	1,173,448								
420,778	1,011,178									
496,200										

Wow, I like that data!

$$y(w,d) = \alpha_w + \sum_{j=1}^{d} \gamma_j + \sum_{t=2}^{w+d} \iota_t + \varepsilon$$

Models for Loss Reserving

So you want to match data and model in such a way that it empowers you to make a good guess, an educated guess about future outcomes.
An actuarial prediction, or forecast, is a very different thing from an investor's forecast. There's very little room for inspired hunches. Do we talk about the risk appetite of an actuary? You are supposed to account for risks but not take them!

What we usually call models or methods I prefer to think of as structures for deriving models. Once you've entered all the parameters you have a model. So, two different sets of parameters in the same structure are two different models. For example, when you use age-to-age ratios there are many different ways of choosing them

Method	0~1	1~2	2~3	3~4	4~5	5~6
Standard Chain Ladder	2.30320	1.42097	1.20149	1.11491	1.07412	1.04784
Arithmetic Average	2.28806	1.41494	1.19832	1.11307	1.07234	1.04741
Geometric Average	2.28656	1.41462	1.19820	1.11301	1.07229	1.04740
Average Without Min/Max	2.27951	1.41098	1.19697	1.11129	1.06840	1.04720
Wtd. Average of Last 4	2.38317	1.44078	1.20870	1.11892	1.07664	1.04802
Average of Last 4	2.37843	1.43814	1.20704	1.11771	1.07532	1.04755

Models for Loss Reserving

Method	0~1	1~2	2~3	3~4	4~5	5~6
Standard Chain Ladder	2.30320	1.42097	1.20149	1.11491	1.07412	1.04784
Arithmetic Average	2.28806	1.41494	1.19832	1.11307	1.07234	1.04741
Geometric Average	2.28656	1.41462	1.19820	1.11301	1.07229	1.04740
Average Without Min/Max	2.27951	1.41098	1.19697	1.11129	1.06840	1.04720
Wtd. Average of Last 4	2.38317	1.44078	1.20870	1.11892	1.07664	1.04802
Average of Last 4	2.37843	1.43814	1.20704	1.11771	1.07532	1.04755
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Each of these choices can be regarded as a different model, and if you think about them you'll find that each of them embodies certain assumptions about the data (or about the modeler or their company). You might also have certain rules of thumb as to when to use one model rather than another – these considerations ought properly to be regarded as a part of your model. This is because we want to be able to compare models in terms of quality, or, if you'll forgive the pun, 'fit for purpose'.

In this example there are always the same number of parameters (for a given size and shape of data array) and that's dictated by your structure, but it seems more natural that the number of required parameters should also depend on what's in the data.

Models for Loss Reserving: Further Considerations

These are some of the things you need to think about, in terms of achieving a good model:

The smaller the number of parameters the better, although this is one consideration among others. In more sophisticated contexts you can use the AIC or BIC to grade various models.
Take away point: redundant parameters harm a model – they improve the fit but add uncertainty to the forecast.

In the table above we showed the age-to-age ratios, but we should also include a 'to Ultimate' ratio – that is really part of the forecast, but the parameters used in a forecast should be regarded as part of the model as well. How do you go about choosing a 'to Ultimate ratio'? Via smoothing, perhaps?

To get a grip on all relevant considerations it is useful to have a sandbox for model testing.

A Sandbox for model QC: Real vs Simulated data

This just means that you have a rich set of sample data that you can use for testing models.
These could be sets of real data where you know the long-term outcome.
You know the data well and use it again and again.
This has advantages over using simulated data, but also has certain drawbacks.
What are the issues or real vs simulated data?
Knowing the outcome with real data can leads to a bias in choice of model. We want to know what the best estimate of the outcome was at the time, and this might differ from what actually happened.
Generally it is good to have access to both kinds of data.

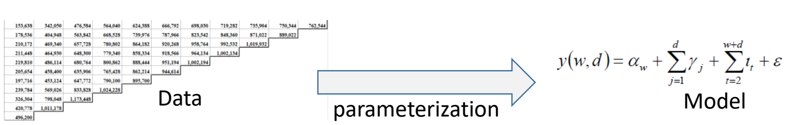
Simulation of data might sound tricky, but having a model for the data and being able to simulate data are almost identical, or should be.

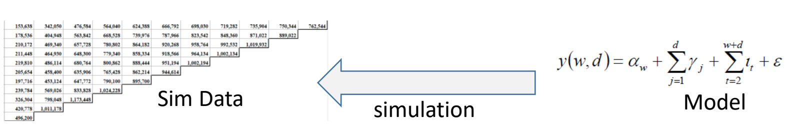
This is an important point and bears repeating.

Modeling and Simulating

Having a model for the data and being able to simulate data are almost identical.

What makes a good model? What makes a good simulation?
Is the answer to these two questions the same? Not really.
More like two ways of looking at the same thing.
A good simulation should be a “deep fake”.



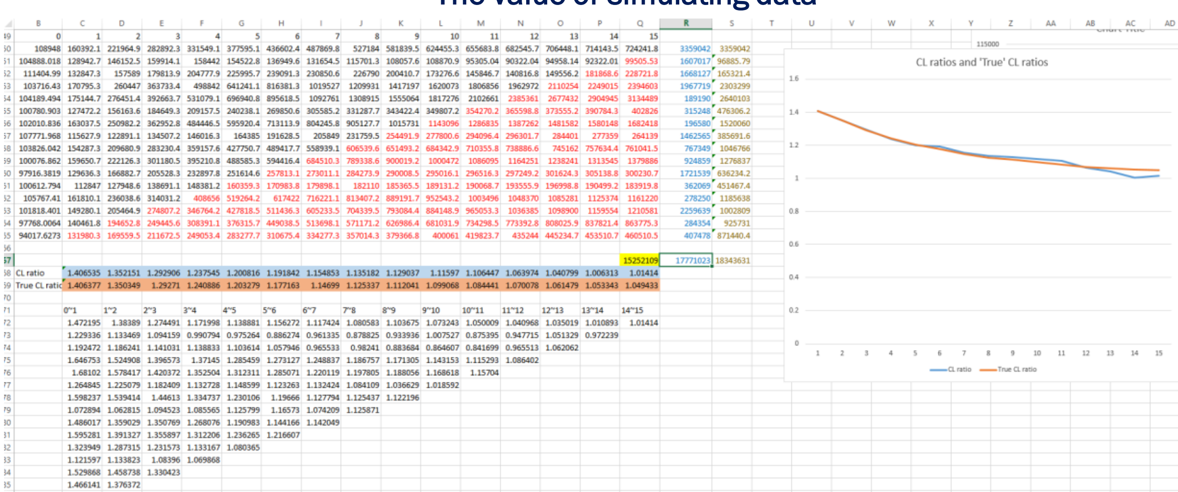


The value of simulating data

Does anyone know of a good chain-ladder based data simulator?

The next few slides show a simple one that I created in Excel. This is an exercise I recommend, especially if you use this method a lot.

The value of simulating data



B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
108948	160392.1	221964.9	282892.3	331549.1	377595.1	436602.4	487869.8	527184	581839.5	624455.3	655683.8	682545.7	706448.1	714143.5	724241.8	3359042	3359042
104888.018	128942.7	146152.5	159914.1	158442	154522.8	136949.6	131654.5	115701.3	108057.6	108870.9	95305.04	90322.04	94958.14	92322.01	99505.53	1607017	96885.79
111404.99	132847.3	157589	179813.9	204777.9	225995.7	239091.3	230850.6	226790	200410.7	173276.6	145846.7	140816.8	149556.2	181868.6	228721.8	1668127	165321.4
103716.43	170795.3	260447	363733.4	498842	641241.1	816381.3	1019527	1209931	1417197	1620073	1806856	1962972	2110254	2249015	2394603	1967719	2303299
104189.494	175144.7	276451.4	392663.7	531079.1	696940.8	895618.5	1092761	1308915	1555064	1817276	2102661	2385361	2677432	2904945	3134489	189190	2640103
100780.903	127472.2	156163.6	184649.3	209157.5	240238.1	269850.6	305585.2	331287.7	343422.4	349807.2	354270.2	365598.8	373555.2	390784.3	402826	315248	476306.2
102010.836	163037.5	250982.2	362952.8	484446.5	595920.4	713113.9	804245.8	905127.7	1015731	1143096	1286835	1387262	1481582	1580148	1682418	196580	1520060
107771.968	115627.9	122891.1	134507.2	146016.3	164385	191628.5	205849	231759.5	254491.9	277800.6	294096.4	296301.7	284401	277359	264139	1462565	385691.6
103826.042	154287.3	209680.9	283230.4	359157.6	427750.7	489417.7	558939.1	606539.6	651493.2	684342.9	710355.8	738886.6	745162	757634.4	761041.5	767349	1046766
100076.862	159650.7	222126.3	301180.5	395210.8	488585.3	594416.4	684510.3	789338.6	900019.2	1000472	1086095	1164251	1238241	1313545	1379886	924859	1276837
97916.3819	129636.3	166882.7	205528.3	232897.8	251614.6	257813.1	273011.1	284273.9	290008.5	295016.1	296516.3	297249.2	301624.3	305138.8	300230.7	1721539	636234.2
100612.794	112847	127948.6	138691.1	148381.2	160359.3	170983.8	179898.1	182110	185365.5	189131.2	190068.7	193555.9	196998.8	190499.2	183919.8	362069	451467.4
105767.41	161810.1	236038.6	314031.2	408656	519264.2	617422	716221.1	813407.2	889191.7	952543.2	1003496	1048370	1085281	1125374	1161220	278250	1185638
101818.401	149280.1	205464.9	274807.2	346764.2	427818.5	511436.3	605233.5	704339.5	793084.4	884148.9	965053.3	1036385	1098900	1159554	1210581	2259639	1002809
97768.0064	140461.8	194652.8	249445.6	308391.1	376315.7	449038.5	513698.1	571171.2	626986.4	681031.9	734298.5	773392.8	808025.9	837821.4	863775.3	284354	925731
94017.6273	131980.3	169559.5	211672.5	249053.4	283277.7	310675.4	334277.3	357014.3	379366.8	400061	419823.7	435244	445234.7	453510.7	460510.5	407478	871440.4
															15252109	17771023	18343631
CL ratio	1.406535	1.352151	1.292906	1.237545	1.200816	1.191842	1.154853	1.135182	1.129037	1.11597	1.106447	1.063974	1.040799	1.006313	1.01414		
True CL ratic	1.406377	1.350349	1.29271	1.240886	1.203279	1.177163	1.14699	1.125337	1.112041	1.099068	1.084441	1.070078	1.061479	1.053343	1.049433		
	0~1	1~2	2~3	3~4	4~5	5~6	6~7	7~8	8~9	9~10	10~11	11~12	12~13	13~14	14~15		
	1.472195	1.38389	1.274491	1.171998	1.138881	1.156272	1.117424	1.080583	1.103675	1.073243	1.050009	1.040968	1.035019	1.010893	1.01414		
	1.229336	1.133469	1.094159	0.990794	0.975264	0.886274	0.961335	0.878825	0.933936	1.007527	0.875395	0.947715	1.051329	0.972239			
	1.192472	1.186241	1.141031	1.138833	1.103614	1.057946	0.965533	0.98241	0.883684	0.864607	0.841699	0.965513	1.062062				
	1.646753	1.524908	1.396573	1.37145	1.285459	1.273127	1.248837	1.186757	1.171305	1.143153	1.115293	1.086402					
	1.68102	1.578417	1.420372	1.352504	1.312311	1.285071	1.220119	1.197805	1.188056	1.168618	1.15704						
	1.264845	1.225079	1.182409	1.132728	1.148599	1.123263	1.132424	1.084109	1.036629	1.018592							
	1.598237	1.539414	1.44613	1.334737	1.230106	1.19666	1.127794	1.125437	1.122196								
	1.072894	1.062815	1.094523	1.085565	1.125799	1.16573	1.074209	1.125871									
	1.486017	1.359029	1.350769	1.268076	1.190983	1.144166	1.142049										
	1.595281	1.391327	1.355897	1.312206	1.236265	1.216607											
	1.323949	1.287315	1.231573	1.133167	1.080365												
	1.121597	1.133823	1.08396	1.069868													
	1.529868	1.458738	1.330423														
	1.466141	1.376372															
	1.436685																

Here is a screen cap. The data is the black part of the array at the top. The red are the future values that I got for free as a result of my method.
Underneath is the age-to-age ratio triangle, which tells me how plausible looking my sims are.
Light blue are the chain-ladder ratios and the C-L ultimates. The orange or light-brown are the 'best' ratios.

The value of simulating data

1. The aim is to simulate data that ‘looks like’ real data.
2. Because a uniform process is used we produce consistent ‘future’ values at the same time. We can use these to test the forecast.

4	5	6	7	8	9	10	1
331549.1	377595.1	436602.4	487869.8	527184	581839.5	624455.3	655683.
158442	154522.8	136949.6	131654.5	115701.3	108057.6	108870.9	95305.0
204777.9	225995.7	239091.3	230850.6	226790	200410.7	173276.6	145846.
498842	641241.1	816381.3	1019527	1209931	1417197	1620073	180685
531079.1	696940.8	895618.5	1092761	1308915	1555064	1817276	210266
209157.5	240238.1	269850.6	305585.2	331287.7	343422.4	349807.2	354270.
484446.5	595920.4	713113.9	804245.8	905127.7	1015731	1143096	128683
146016.3	164385	191628.5	205849	231759.5	254491.9	277800.6	294096.
359157.6	427750.7	489417.7	558939.1	606539.6	651493.2	684342.9	710355.
395210.8	488585.3	594416.4	684510.3	789338.6	900019.2	1000472	108609
232897.8	251614.6	257813.1	273011.1	284273.9	290008.5	295016.1	296516.
148381.2	160359.3	170983.8	179898.1	182110	185365.5	189131.2	190068.
408656	519264.2	617422	716221.1	813407.2	889191.7	952543.2	100349
346764.2	427818.5	511436.3	605233.5	704339.5	793084.4	884148.9	965053.
308391.1	376315.7	449038.5	513698.1	571171.2	626986.4	681031.9	734298.

3~4	4~5	5~6	6~7	7~8	8~9	9~10
1.171998	1.138881	1.156272	1.117424	1.080583	1.103675	1.073243
0.990794	0.975264	0.886274	0.961335	0.878825	0.933936	1.007527
1.138833	1.103614	1.057946	0.965533	0.98241	0.883684	0.864607
1.37145	1.285459	1.273127	1.248837	1.186757	1.171305	1.143153
1.352504	1.312311	1.285071	1.220119	1.197805	1.188056	1.168618
1.132728	1.148599	1.123263	1.132424	1.084109	1.036629	1.018592
1.334737	1.230106	1.19666	1.127794	1.125437	1.122196	
1.085565	1.125799	1.16573	1.074209	1.125871		
1.268076	1.190983	1.144166	1.142049			
1.312206	1.236265	1.216607				
1.133167	1.080365					

3. A table of cumulatives doesn’t reveal much to the eye, so the idea was to produce a plausible looking age-to-age ratios table.

The value of simulating data

4. The CL method is deterministic so there is a need to ‘inject’ randomness into the simulation. There is no obvious consistent way to do this, so it’s a matter of choice. The underlying structure is therefore CL-skeleton with randomness added on.

In my example I put random fluctuations in the individual age-to-age ratios and added some normally distributed ‘noise’ on top of that. I wanted a small minority of individual age-to-age ratios to be less than 1.

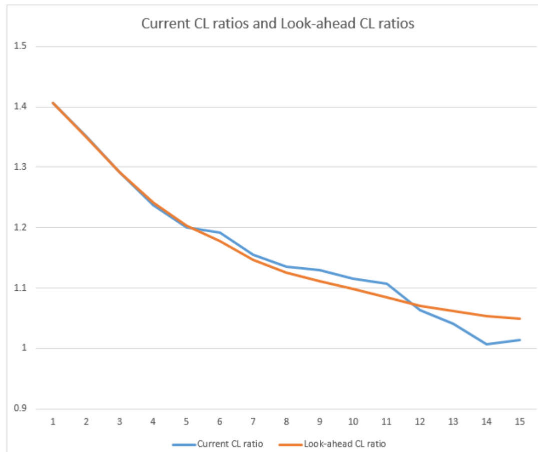
fx =D4*(D4/C4+(D4/C4-1)*(0.25*RAND()-0.3))+AVERAGE(D4:D19)*NORM.S.INV(RAND())*E$1/50

C	D	E	F	G	
7*RAND()	=(0.8+0.25*RAND())*	=(0.8+0.25*RAND())*	=(0.8+0.25*RAND())*	=(0.8+0.25*RAND())*	=
1+0.7*RAND())	=C4*(C4/B4+(C4/B4-	=D4*(D4/C4+(D4/C4-	=E4*(E4/D4+(E4/D4-1	=F4*(F4/E4+(F4/E4-1	=
1+0.7*RAND())	=C5*(C5/B5+(C5/B5-	=D5*(D5/C5+(D5/C5-	=E5*(E5/D5+(E5/D5-1	=F5*(F5/E5+(F5/E5-1	=
1+0.7*RAND())	=C6*(C6/B6+(C6/B6-	=D6*(D6/C6+(D6/C6-	=E6*(E6/D6+(E6/D6-1	=F6*(F6/E6+(F6/E6-1	=
1+0.7*RAND())	=C7*(C7/B7+(C7/B7-	=D7*(D7/C7+(D7/C7-	=E7*(E7/D7+(E7/D7-1	=F7*(F7/E7+(F7/E7-1	=
1+0.7*RAND())	=C8*(C8/B8+(C8/B8-	=D8*(D8/C8+(D8/C8-	=E8*(E8/D8+(E8/D8-1	=F8*(F8/E8+(F8/E8-1	=
1+0.7*RAND())	=C9*(C9/B9+(C9/B9-	=D9*(D9/C9+(D9/C9-	=E9*(E9/D9+(E9/D9-1	=F9*(F9/E9+(F9/E9-1	=
*(1+0.7*RAND()	=C10*(C10/B10+(C10	=D10*(D10/C10+(D10	=E10*(E10/D10+(E10/	=F10*(F10/E10+(F10/	=
*(1+0.7*RAND()	=C11*(C11/B11+(C11	=D11*(D11/C11+(D11	=E11*(E11/D11+(E11/	=F11*(F11/E11+(F11/	=
*(1+0.7*RAND()	=C12*(C12/B12+(C12	=D12*(D12/C12+(D12	=E12*(E12/D12+(E12/	=F12*(F12/E12+(F12/	=
*(1+0.7*RAND()	=C13*(C13/B13+(C13	=D13*(D13/C13+(D13	=E13*(E13/D13+(E13/	=F13*(F13/E13+(F13/	=
*(1+0.7*RAND()	=C14*(C14/B14+(C14	=D14*(D14/C14+(D14	=E14*(E14/D14+(E14/	=F14*(F14/E14+(F14/	=

The value of simulating data

5. Because I had the 'true' forecast values I also had access to 'true' or 'look-ahead' CL ratios, based on all the data instead of just the current data.
The look-ahead ratios when graphed were smoother than the current ratios – which might make a case for ratio smoothing.

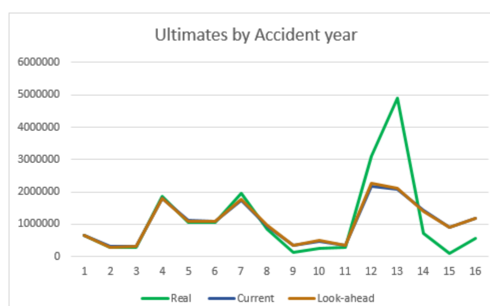
14	101818.4011	149280.09	205464.9	274807.2	346764.2
15	97768.00644	140461.82	194652.8	249445.6	308391.1
16	94017.62731	131980.27	169559.5	211672.5	249053.4
	Current CL ratio	1.4065347	1.352151	1.292906	1.237545
	Look-ahead CL ratio	1.4063774	1.350349	1.29271	1.240886



The value of simulating data

6. Naturally, part of the simulator was to run the CL method forecast with the two sets of ratios, and compare the ultimates with the 'true' values that were consistently simulated.

.6	552765.589	564919	1170456	1170506
	Ultimates:	Real	Current	Look-ahead
		18038852	16814808	16969278
)8	1.09971743	1.095544		



7. How did we do? Typically with this set-up the CL forecasts differ significantly from the true result. And it's clear why. Models are good in the early accident years but poor in the later ones. They under-estimate the volatility.
8. This gives you exactly the motivation behind Bornhuetter-Ferguson. But the problem it seeks to correct is not in the data, but in the model.

Take-away point #1: Having a data model and being able to simulate data are almost identical

1. If your model is derived from (parameterized by) real data, then the datasets you simulated ought to 'look-like' the original data.

2. This is an important idea: **if your model really captures the features in the original data then you should be able to run it backwards and produce datasets that are similar to the original data. This is a powerful way of testing the model.**

3. It leads to an important question: **What does it mean to be similar to the data?**

4. In my example I wasn't trying to simulate a particular dataset, but I judged the plausibility of the data by the plausibility of the age-to-age ratios.

5. So one answer: Every result of transforming the data that arises in the course of modeling it should 'look like' the same result for the real data. (The transformations should make the 'looks like' easier to judge.) A good simulation should be a "deep fake".

6. Corollary: If it's easy to tell that data simulated from a model is not real data then it's probably not a good model.

Take-away point #1: Having a data model and being able to simulate data are almost identical

7. A further point: My Excel CL-reversing simulator was poor because the randomness was introduced in a completely ad hoc manner. A better model is going to have some way of measuring the volatility inherent in the data and incorporating that, so that when you run the model backwards your simulations should come out with pretty much the same volatility.

Enter Mack and the Bootstrap!

But first a digression...

Digression: Simulating Insurance

Before we go on, one reason that I recommend creating your own CL data simulator is that it's fun. It very quickly becomes a game where you can play around with the formulas and parameters and see how good you can get at forecasting. It's simple enough to do this easily.

In general 'game-ification' is a bit of a buzzword these days. We're reliably informed that people learn skills faster if it is turned into a game. How many of you learned to type in this way?

Someone might create a good game to learn loss-reserving, putting in all the pitfalls.

A while back I had the idea of creating a P&C Insurance simulator game. The idea was to turn it into an app that we could give away at conferences like this. People might enjoy playing it for a day or two and it would have our logo plastered over it.
It never got beyond a prototype in Python which I want to show you. It doesn't have any logos on it, but if you think it might be fun to play come and see me at our booth and I'll give you a copy of the Python script. (The Python language is free and the script is just an easy to understand text file.)

The P&C Insurance Game

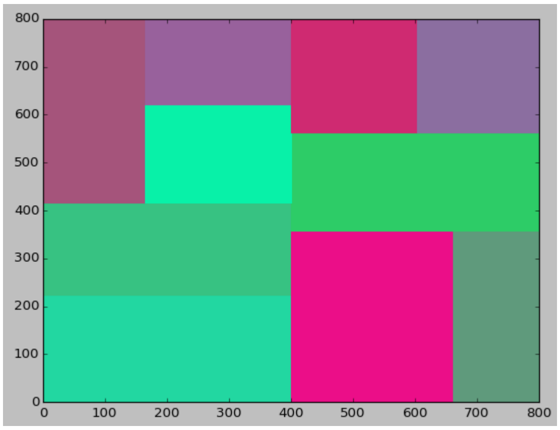
Two players: You and your Competitor (= the computer)

When the game round starts you get a list of properties seeking insurance:

```
>>>
Start? y
Number of properties? 10
property # 1 : insured value =  1081791.0  event probability =  3.0 %
property # 2 : insured value =  2112635.0  event probability =  6.0 %
property # 3 : insured value =  1157874.0  event probability =  17.0 %
property # 4 : insured value =  1136942.0  event probability =  24.0 %
property # 5 : insured value =  1800431.0  event probability =  10.0 %
property # 6 : insured value =  2216953.0  event probability =  5.0 %
property # 7 : insured value =  1129078.0  event probability =  1.0 %
property # 8 : insured value =  1486219.0  event probability =  15.0 %
property # 9 : insured value =  987038.0  event probability =  21.0 %
property # 10 : insured value =  1930853.0  event probability =  14.0 %
```

Each property want to be insured to a certain value and has a certain probability of a loss event in the insured period. The loss, if it happens, can be anything up to the insured value.
The game model includes *randomized environmental factors* affecting severity and event likelihood. These change with each round.

The P&C Insurance Game



It generates a little picture of the properties. Size = value, color = event likelihood.

When you close the picture window you are prompted for your premium-setting strategy:

```
------------------------------------

What is your strategy? [eg. OS3R2 or C+S-R] OS0R-2
```

Strategy = how you set your premiums. Whether you or your competitor get the account is premium driven, but only probabilistically.

So depending on the client you will be more or less competitive than the base rate which is designed for LR = 90%
S means discounting for high value accounts; R means discounting for high risk ones.
OS2R-1 means you prefer low risk and strongly prefer high value accounts.
C+S-R means you look at what your competitor is charging and whatever it is you try do outdo him for high value low risk.

The P&C Insurance Game

OS0R-2 was chosen in response to the picture. In this case the competitor chose a similar strategy, but discounted more on large accounts. Split was about equal, but we raised a bit more premium. The ELR for each policy indicates discounting. Here the Competitor did more of that.

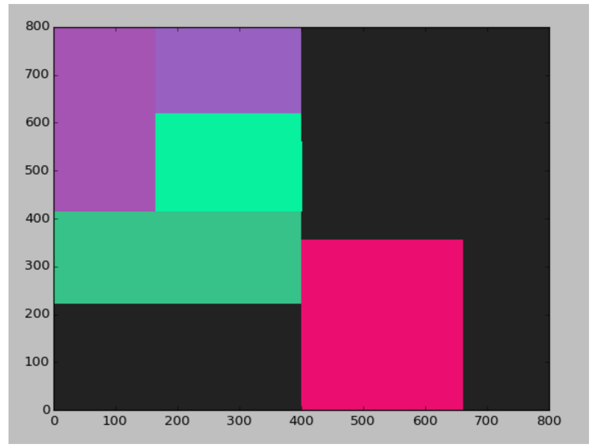
```
Competitor strategy is OS1R-1
[0, 1, 1, 1, 0, 0, 1, 1, 0, 0]
MY PORTFOLIO
--------------------
property # 2  Premium =  126981.0  Value =  2,112,635  Risk level =  0.055 ELR =  92 %
property # 3  Premium =  211432.0  Value =  1,157,874  Risk level =  0.166 ELR =  91 %
property # 4  Premium =  298153.0  Value =  1,136,942  Risk level =  0.236 ELR =  90 %
property # 7  Premium =  10382.0  Value =  1,129,078  Risk level =  0.009 ELR =  92 %
property # 8  Premium =  250263.0  Value =  1,486,219  Risk level =  0.153 ELR =  91 %
--------------------------
Total Premium =  897,211

COMPETITOR PORTFOLIO
-----------------------
property # 1  Premium =  39501.0  Value =  1,081,791  Risk level =  0.034 ELR =  93 %
property # 5  Premium =  184633.0  Value =  1,800,431  Risk level =  0.096 ELR =  93 %
property # 6  Premium =  108132.0  Value =  2,216,953  Risk level =  0.046 ELR =  94 %
property # 9  Premium =  222309.0  Value =  987,038  Risk level =  0.207 ELR =  92 %
property # 10  Premium =  289307.0  Value =  1,930,853  Risk level =  0.14 ELR =  93 %
--------------------------
Total Premium =  843,882
```

The P&C Insurance Game

The competitor's properties have been blacked out.
We only got two of the 5 low risk properties, and we got the big high risk one.
Not as good a position as we'd hoped for.

Salesmanship counts for something, but at least we didn't discount as much as the competitor.

ready? y



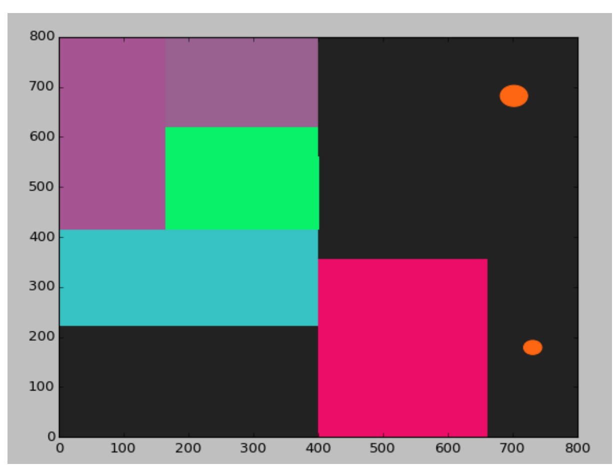
The P&C Insurance Game

property # 5 is hit.
property # 10 is hit.

The size of the orange dot shows the severity.

All of our properties were event free!

Bonus to the sales team!!

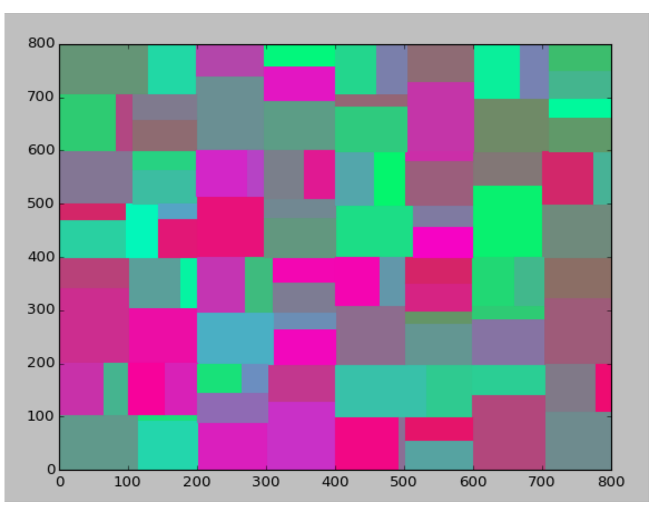


The P&C Insurance Game

```
My losses =  0
Change = 897,211
My total funds =  897,211
------------------
Competitor losses = 2,213,494
Change = -1,369,611
Competitor total funds =  -1,369,611

=======================================
```

```
Another round? y
Number of properties? 100
```

The P&C Insurance Game



Back to the main thread: The value of simulating data

One moral we can draw from all of that is when you simulate you need to choose your random inputs carefully, both as to distribution and where in the structure you inject them. The original Excel CL model failed on both counts: the random bits were chosen ad hoc from easy distributions, they were used to make the individual ratios table look good, but with no further insight into the structure.

The Bootstrap Method (which is ultimately nothing more than a way to run the Chain-Ladder backwards*) attempts to solve both of these problems in a single stroke.

1. We'll extract the random inputs from the deviations between model and data (i.e. the 'residuals')
2. We'll inject them in exactly the same places as we took them from.
3. Randomness will come from randomly shuffling them.

There are numerous technical problems that have to be solved to make this work, of which I'll only mention the first:

If you are going to shuffle the residuals they need to all be on the same basis – differences between big/small numbers are typically big/small – they need to be scaled before you can swap them.

The natural way to do this is via a regression, where part of the process is to compute the standard deviation of the residuals; this ought to be the natural scaling factor.

So, we'll follow Thomas Mack and express each step of the C-L as a regression.

*When applied to the Mack model. Other bootstraps invert other models.

The Chain-Ladder as a set of regressions

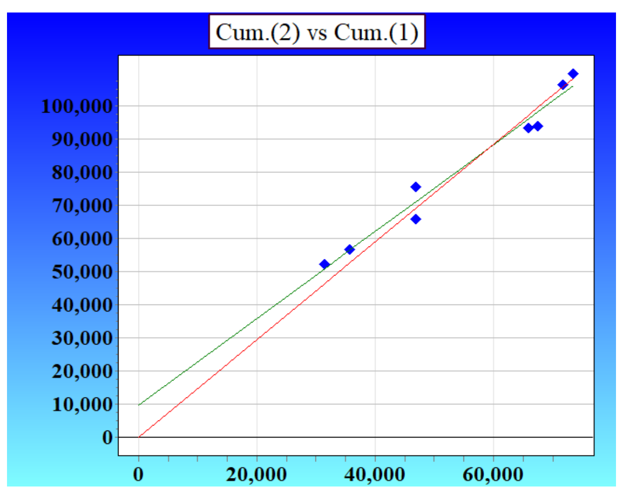
Here is what a single step in the C-L looks like as a regression. (Real data.)

The red line is the C-L ratio, and its gradient has been selected so as to minimize the sum of the squares of the distances from the blue dots, but with a weighting so that the same deviation counts more lower down the line than higher.

It looks pretty good, doesn't it?

Actually there's an optimistic bias here, since *Cumul2 = Cumul1 + Incr2*

The true regression underneath what you see here is of *Incr2* on *Cumul1*:

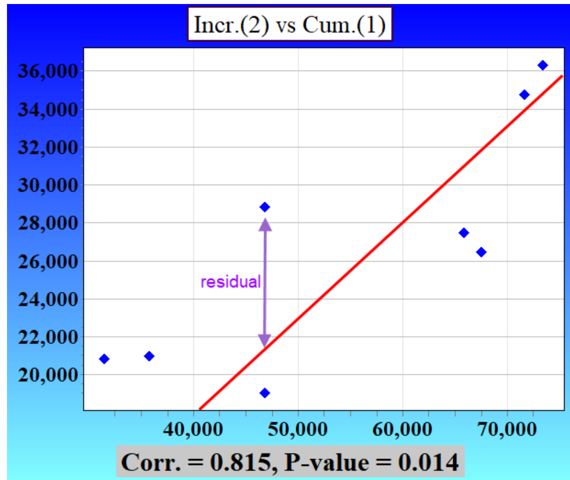


The Chain-Ladder as a set of regressions

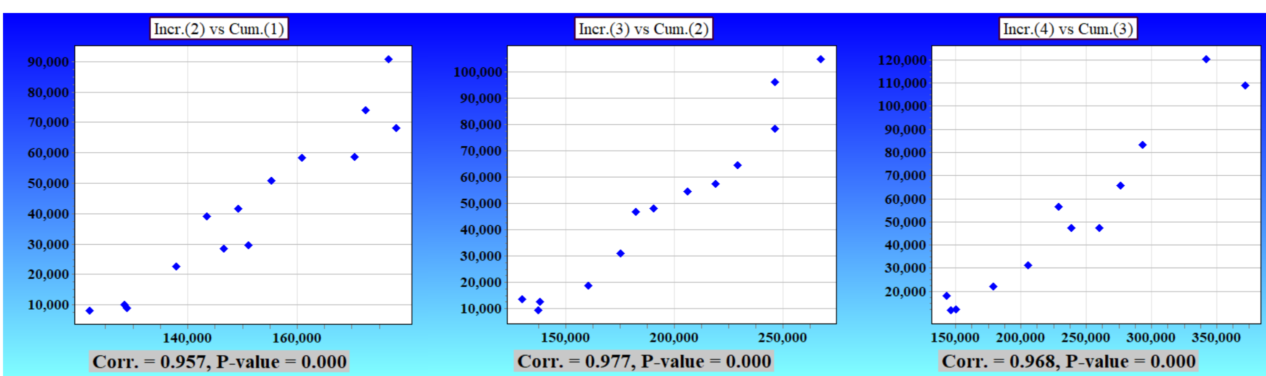
The residuals from this regression, the differences between observed and predicted, suitably reweighted and standardized will provide us with a completely interchangeable set of random-like values expressive of the inherent volatility in the data.

To simulate a dataset we can sprinkle these residuals randomly over our data array, unweighting and un-standardizing according to the cell they fall into, and then add them onto the regression value (red line) that belongs with that cell.

Incidentally, these regressions form a powerful fake data detector. If I apply them to data produced by my Excel C-L simulator, I get…



Chain-Ladder regressions as fake data detector



Here are three successive age-to-age regressions from the data generated by my primitive C-L reverser. The correlations are insanely high, such as are never seen in real data! They completely expose my fabrication.

(Note that Bootstrap generally abbreviates as 'BS'.)

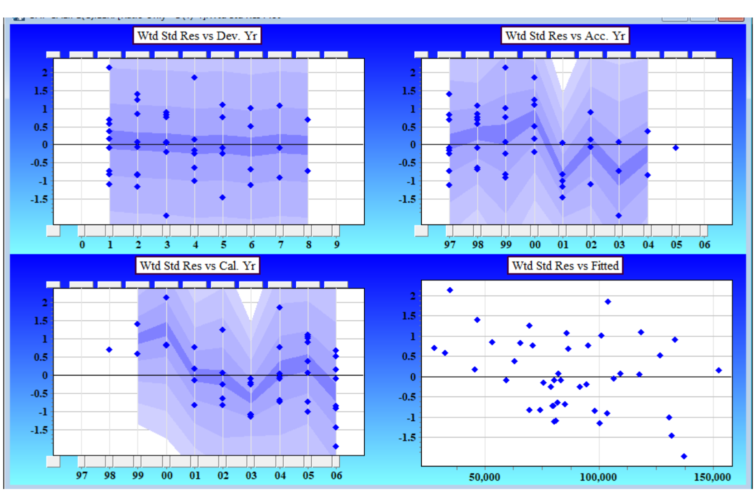
The Chain-Ladder as a set of regressions

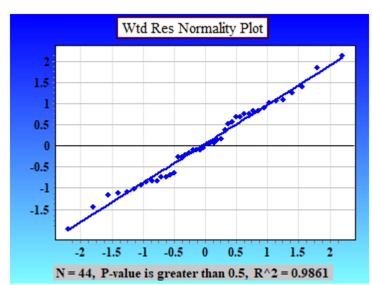
Now that we have a full set of standardized residuals we can do three things:
(1) Plot them in different views, and this will give us a sort of x-ray of **how well the model fits the data**.
(2) Create **as many Bootstrap pseudo-data sets as we want** and use them to draw out distributional measures for our C-L model.
And
(3) Combine (1) and (2) by comparing the X-rays of the real data with those of the pseudo-data to **again assess the quality of the model**.

I'll finish by quickly going through each of these in turn.

Evaluating the fit of model to data

(1) Plot them in different views.
Here are five different ways of looking at the residuals:

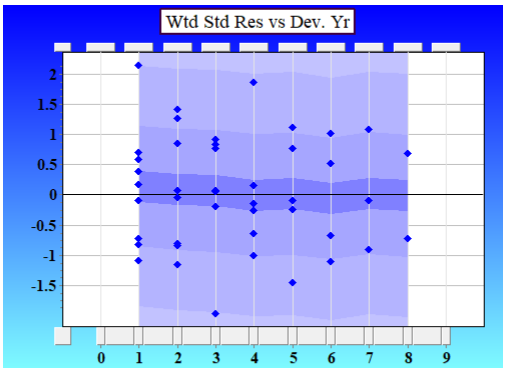




Left clockwise:
1. By development year
2. By accident year
3. By Size of underlying value
4. By Calendar year

Above:
1. Normality plot – quantile vs. Normal quantile.

Evaluating the fit of model to data: the view from development

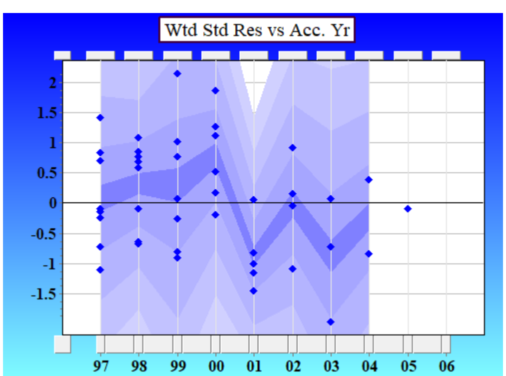


Along the development axis the residuals ought to appear nicely balanced. This is the age-to-age direction; each residual comes out of its respective regression and the weighting and standardization mean that the shade bars ought to be close to horizontal.

Weighting = adjusting the Std. Deviation by size of the predictor
Standardizing = adjusting the residual by the Std. Deviation.

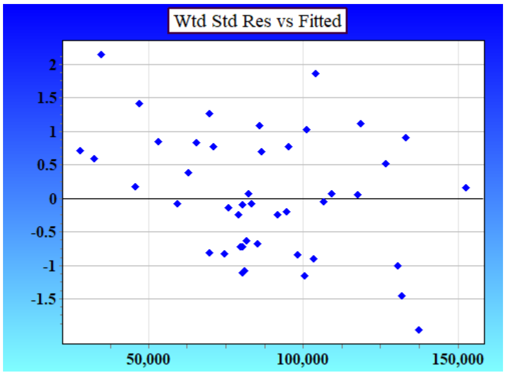
This shows only that Mack was done correctly.

Evaluating the fit of model to data: the view from accident



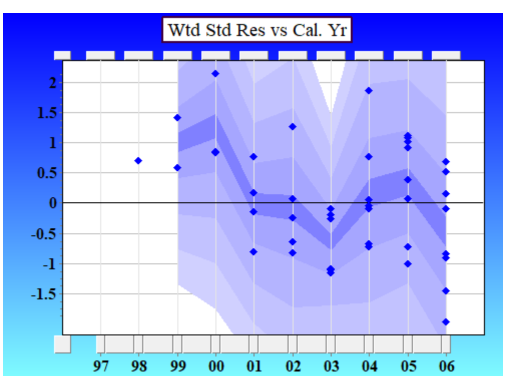
Along the accident axis we can see some structure that was not captured by the model.
The early year residuals lean positive, the later ones lean strongly negative.
Does this matter?
Yes, it does.
It means our forecasts come in too low in the early years and too high in the later years, and the division is sharp at 2000~2001.
It's something you have to look into. You might need to split the data, or remove early years from your calculations.
(This data is CAL.)

Evaluating the fit of model to data: the view from size



Arrayed according to the size of the fitted value the division between positive and negative residuals appears again. The big numbers have negative residuals while the small ones have positive ones. This suggests a systematic bias in our model derived values and that the forecast is likely to come in on the high side.
The conclusion is not certain, but again we see that there is structure in the data that is not removed by the model.

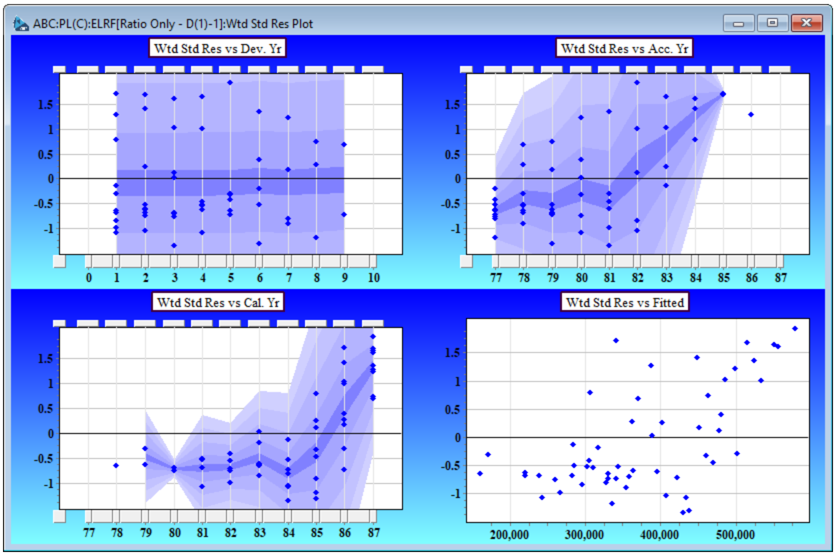
Evaluating the fit of model to data: the view from calendar



In the calendar direction this unaccounted for structure appears as something like a constant trend. The further in time we go the more out of kilter the fitted values are.
Calendar trends (aka inflation) are not picked up by this kind of model. They can have severe effects on the forecast. If you know they are there you might be able to develop strategies to compensate – but first you need to know they are there.

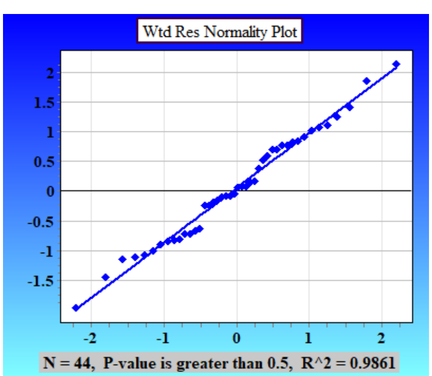
In this case the downward trend means the model likely overestimates future losses. The next slide shows a more dangerous example.

Evaluating the fit of model to data: a synoptic view



This is WC data.
In this case there was a radical change in trend at about 1984.
We go from overfitting the earlier years to seriously underfitting the later ones.
Trusting this model without further adjustment would lead to drastic under-reserving and eventual insolvency.
(Which I believe is what happened.)

Evaluating the fit of model to data: the view from Normal

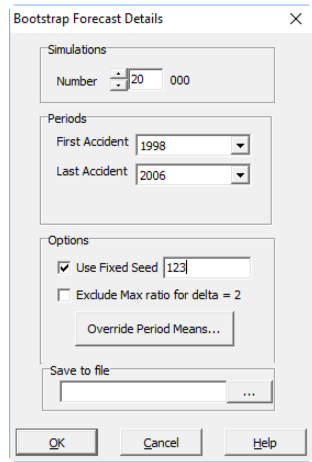


The fifth 'x-ray' picture is the Normality plot, which for this example looks good. There are reasons why we want the weighted and standardized residuals to be Normal.
In fact this speaks to Mack's original motivation in developing the regression formulation of the C-L.
Of all the ways that could have been used to come up with an average age-to-age ratio, the C-L formula is best if these residuals are Normal. It is a hidden assumption in the method.

Create as many Bootstrap pseudo-data sets as we want and use them to draw out distributional measures for your C-L model

As they say about sausages: There is no problem as long as we don't go into the details of how it is done.

Depending on your software you press a few buttons to set it up.



And voilà!

Bootstrap distribution

Sample-Based Statistics

Accident Year	Sample Mean	ELRF Mean	Sample Median	Sample S.D.	ELRF S.D.
1998	507	989	491	947	478
1999	932	914	929	1,385	1,065
2000	2,725	2,976	2,713	1,970	1,857
2001	8,964	6,536	8,906	3,672	2,274
2002	18,633	16,360	18,592	7,250	6,624
2003	36,648	28,916	36,517	9,945	9,154
2004	59,203	50,009	58,955	12,004	10,598
2005	82,725	75,373	82,011	18,791	13,343
2006	92,666	86,679	92,116	20,677	19,662
Total	303,003	268,752	302,153	37,952	33,501

20,000 Simulations. 1 Unit = 1,000 €

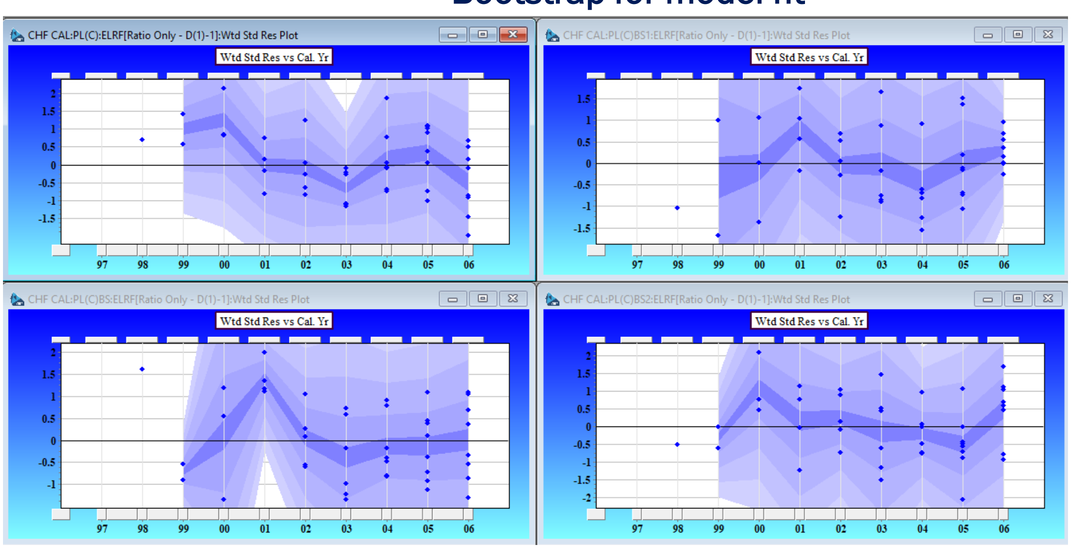
%	Sample			
	Quantile	# S.D.'s	VaR	T-VaR
99.7	409.503	2.806	106.500	118.402
99.6	406.963	2.739	103.959	115.168
99.5	403.711	2.654	100.708	112.664
99.4	401.117	2.585	98.114	110.484
99.3	398.778	2.524	95.774	108.568
99.2	397.495	2.490	94.492	106.882
99.1	395.702	2.443	92.699	105.399
99.0	394.295	2.405	91.292	104.068
98.0	383.416	2.119	80.413	94.615
97.0	376.138	1.927	73.134	88.595
96.0	371.186	1.797	68.183	84.104
95.0	366.585	1.675	63.582	80.455
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

The entire distribution is available. However, if you look closely you can see that the Bootstrap sample mean is about 13% bigger than the ELRF = Mack = C-L mean, and the sample S.D. is about 13% bigger than the Mack S.D.
If you like the C-L mean value better you can shift the distribution over to equalize the means.

Combine Bootstrap and model-fit by comparing the residual 'X-rays' of the real data with those of the pseudo-data to assess the quality of the model

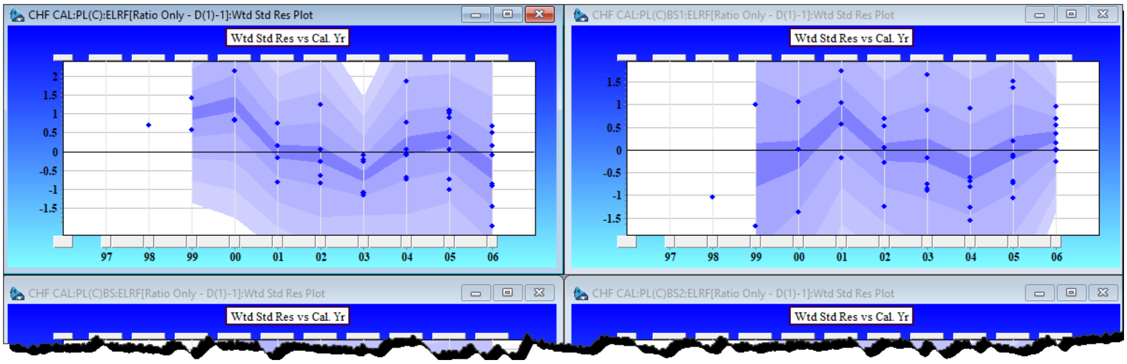
Bootstrap simulated pseudo-datasets are computationally very cheap. We just used 20,000 of them to derive a loss distribution. For each such dataset we computed its C-L forecast and threw it away.
Now we'll use the Bootstrap to make just three pseudo-datasets, but we'll hang on to them in full detail. It's not the forecasts we want to look at but the residual plots from applying Mack.

Bootstrap for model fit



Here they the calendar direction residual plots. Can you see the odd one out?

Bootstrap for model fit



If you went for the upper left you were right. The downward trend in the residuals in the calendar direction disappears in the Bootstrapped samples. Which is just what you'd expect when you think about it.
In one way you could think of it as smoothing out the idiosyncrasies in the data.
But if you regard the residual plots as valuable diagnostic tools…
… you might better call it, "shredding the evidence."

Casualty Actuarial Society
4350 North Fairfax Drive, Suite 250
Arlington, Virginia 22203

www.casact.org