Residual Loss Development and the UPR

Richard L. Vaughan, FCAS, FSA, MAAA

Abstract. Traditional reserve estimators such as chain-ladder and Bornhuetter-Ferguson model unpaid losses as a function of accident period versus lag to payment or reporting. The result of primary interest is expected future losses; these are derived from intermediate results such as lag factors and loss ratios.

With certain adjustments, traditional estimators may also be used for the statutory Unearned Premium Reserve, or UPR, for long-duration contracts. This reserve is governed by SSAP 65, the most important requirement of which is "Test 2", that earnings be recognized in proportion to the expected emergence of losses and expenses. Here the estimators model unincurred losses by issue month versus lag to incurral. The result of greatest interest is now the set of lag, or earnings, factors.

Adjustments are necessary to accommodate the decline in exposure due to cancellations, the deficiency of recent diagonals in the issue-to-incurral lag triangle due to unreported losses, and the unusual shapes and weight of tails for immature business. In particular, it is convenient to develop not losses *per se* but partial loss ratios to premium remaining in force, thus "factoring out" the effect of cancellations and leaving the results correct on a no-cancellation basis.

In this paper we suggest taking this adjustment one step further, and developing loss ratios to the product of premium in force and a set of positive *a-priori* earnings factors, the final earnings factors being the product of the *a-priori* factors and these "residual" earnings factors. In the case of automobile extended service contracts, there exists an excellent model for such *a-priori* factors, published by Kerper and Bowron in 2007 [1], but the technique is not dependent on any particular underlying model.

We demonstrate that this procedure (a) improves the robustness of the estimators to lack of perfect homogeneity in the data, (b) greatly simplifies the specification and calculation of tail factors, and (c) facilitates the use of reference factors to improve the estimates at lags where the experience data is sparse.

1. INTRODUCTION

1.1 Statutory Requirements for the Unearned Premium Reserve

Statement of Statutory Accounting Principles 65 (SSAP 65) defines three tests for the adequacy of the Unearned Premium Reserve (UPR) for contracts longer than 12 months in duration: Test 1, provision for refunds, Test 2, earnings to be proportional to emergence of losses and expenses, and Test 3, provision for unincurred losses and expenses. Of these, Test 2, which requires that the UPR be no less than the gross premium multiplied by the ratio of expected future to total losses, including expenses, is normally dominant. More importantly, a UPR created to satisfy, and just satisfy, Test 2 will release earnings in such a way as to make immature inception-to-date loss ratios useful predictors of ultimate loss ratios, and thus give management an accurate measure of the performance of the business in question.

1.2 Satisfaction in Aggregate

SSAP 65 need only be satisfied in aggregate, for all of a company's long-duration contracts together. But it is the common practice of writers of long-duration contracts to calculate the UPR at the contract level, using either predefined "strings" (vectors of UPR factors by lag) or formulas. Only then do they accumulate this UPR over accounting segments of interest, and finally sum it across the entire company. The company's implied goal is that satisfaction of statutory requirements by most individual contracts will ensure satisfaction for most accounts and that satisfaction for most accounts will ensure satisfaction in aggregate.

The actuary testing the carried UPR typically starts by comparing average carried UPR factors with those indicated by experience, over reasonably homogeneous subdivisions of accounting segments, such as term groups. A close fit suggests, though it does not prove, that the carried UPR factors are satisfactory contract by contract, and will produce loss ratios that are reasonable predictors of ultimate experience. At the least it is, provisionally, a negative result in a management-by-exception sense, allowing attention to be focused elsewhere. The actuary also sums the indicated UPR's across each accounting segment, and finally across the entire company, to test the aggregate carried UPR for satisfaction of SSAP 65.

While the author believes this is a correct procedure it should be pointed out that it is really a matter of how the requirement that Test 2 be satisfied *in aggregate* is interpreted. Does this mean that the UPR must be no less than the aggregate inforce premium multiplied by aggregate future losses divided by aggregate ultimate losses (a "ratio of aggregates" definition), or does it mean that the UPR must be no less than the aggregate, over some exhaustive and mutually exclusive set of subdivisions, of premium multiplied by future losses divided by ultimate losses (an "aggregate of UPR's" definition)? We believe that the only sensible interpretation of the requirement is the latter. This is explained in Appendix A.

1.3 Estimation in Detail

Strings or formulas can give a UPR for each contract or for any collection of contracts, however small. But strings or formulas must be validated; this requires subdivisions that have enough experience to be credible, as well as being reasonably homogeneous.

Subdividing for homogeneity considers factors such as account, contract type (mechanical repair, GAP, etc.), insured product (automobile, boat, appliance, etc.), term in months, term in miles, and manufacturer's warranty. But using too many factors simultaneously may produce subdivisions with too few contracts for reliable estimates of UPR factors. So the actuary often must settle for reasonable, but not perfect, homogeneity.

Within a subdivision, we may use conventional methods to estimate UPR factors, loss ratios, and unincurred losses, provided certain adjustments are made to accommodate cancellations, unreported losses in policy month versus accident lag triangles, and the possibility of tail factors. For complete details of these adjustments see [2]. Here we propose a further adjustment to improve accuracy with imperfectly homogeneous data, to simplify greatly the projection of tails, and to weight the results against a simple set of reference factors to remove noise from the development at the later lags. The adjustment is similar to that used in the All-Terms Factors model, described in [2], to obtain residual

ATF's, and we call it "residual loss development". It requires a matrix of *a-priori* expected earnings factors, by issue month versus lag; for automobile service contracts; this may be derived contract by contract from exposure models such as that of Kerper and Bowron (KB) [1],[2].

2. BACKGROUND AND METHODS

2.1 Estimators of Earnings Factors

We consider estimators to be applied to experience data within a subdivision and to return earnings factors representing averages across the contracts in the subdivision. If these earnings factors resemble similar averages using the company's carried strings or formulas, we shall regard the strings or formulas as confirmed at that level of aggregation; otherwise, we shall take the results as evidence of a need for change.

Our suggested technique of residual loss development may be applied in conjunction with many underlying development methods. For concreteness in the discussion we assume that the estimators of earnings factors are in the chain-ladder family, and the estimators of future losses are in the Bornhuetter-Ferguson family, both of which have proven reliable in the context of extended service contracts. These families encompass variations in depth and weighting of average development factors, graduation of development or lag factors, and choice of expected loss ratios.

These estimators do require adjustment to cope with cancellations and with unreported losses in the issue-period-versus-incurral-lag triangle. For the purpose of this paper the most significant such adjustment is to develop, not losses *per se*, but partial loss ratios; this quite neatly produces earnings factors on a no-cancellation basis as required for SSAP 65 Test 2.

Losses include loss adjustment expenses where available; in practice the DCC component of such expenses may be negligible and the A&O component may be assumed to earn in parallel with the losses themselves. Issue periods and lags may be of any length but are normally (and will be assumed to be) of length one month.

The lag factors are normalized to total 1.00 and are then called earnings factors, since they represent the fraction of gross premium to be earned at each lag according to the principle of SSAP 65 Test 2. Their reverse cumulative sums represent SSAP 65 Test 2 UPR factors at the moment of issue and the beginning of each subsequent month; the first such factor will be 1.00. It is usually assumed that contracts are written uniformly throughout each month, so that there are potentially T+1 months with nonzero earnings factors (including the initial 1.00) if the term of the contracts is T. The experience may include some "goodwill" claims incurred even later; these may be handled by allowing the tail to extend beyond the term, or by treating all goodwill claims as if paid in the last month of the term and accepting the usually small misstatement of earnings in that month. Administrative systems usually require strings or formulas with exactly T factors after the initial 1.00.

2.2 Imperfect Homogeneity

It is not common for every subdivision of an account to be entirely homogeneous as regards factors that affect the earnings pattern. If all contracts in a subdivision have identical term in months, they still may differ in factors such as term in miles, manufacturers' basic warranties, manufacturers' power-train warranties, and odometer readings at issue. They may also differ in expected loss ratios, i.e. premium adequacy; while the loss ratio does not affect the earnings pattern of an individual contract, heterogeneity in loss ratios may affect the average earnings pattern of a subdivision.

Heterogeneity that is random over time, but stationary, presents little problem. Estimated earnings factors will be correct for the static mix, and, if it continues, will be correct on average for new contracts. Heterogeneity that changes over time is more serious. If contracts written in the early issue months of our lag triangle differ from those written later, then the resulting estimated earnings pattern may not be correct for either group of contracts, or even for a consistent mixture of the two, because each lag includes a different proportion of earlier and later contracts.

2.3 Residual Loss Development

By *residual loss development* we mean dividing the known cells of a loss triangle by an *a-priori* earnings pattern, developing the quotients, and multiplying the resulting earnings factors by the a-priori earnings pattern for both known and projected cells. The *a-priori* factors explain part of the observed pattern; we develop the residuals; the final estimate is the combination of a-priori and residual patterns.

Let the matrix $\boldsymbol{L} = [L_{ij}]$ represent losses for issue month i incurred at lag j, and reported through month $i+n$ (and therefore a "triangle" populated only for $i+j<=n+1$). Similarly, let $\boldsymbol{E} = [E_{ij}]$ represent exposures, such as premiums or numbers of contracts, for issue month i still in force, i.e. not cancelled, as of lag j. It is important to note that expirations, whether by months or miles, have no effect on the status of being "still in force"; the analysis is much simpler if the effect of expirations is allowed to emerge via the earnings factors rather than the exposure.

We are interested in completing $\boldsymbol{L}$ to a fully-populated "rectangle" $\boldsymbol{L}^c$ by estimating values for the future portion where $i+j>n+1$, as well as for the unreported portions of cells in the original $\boldsymbol{L}$. For SSAP 65 Test 2, we may be particularly interested in the earnings factors, or expected values of a row of $\boldsymbol{L}^c$, normalized to total 1. In completing $\boldsymbol{L}$ to $\boldsymbol{L}^c$ we may need future exposures; we assume that we have already completed $\boldsymbol{E}$ to a rectangle $\boldsymbol{E}^c$, for example by applying chain-ladder to $\boldsymbol{E}$ to estimate contract persistency rates, usually a straightforward procedure.

Finally we assume we have already estimated a vector $\boldsymbol{b} = (b_j)$ of cumulative *report* lag factors, for example by applying chain-ladder to a triangle of losses by incurral month versus report lag, again usually a straightforward procedure. In the absence of reliable report-date data we may let $\boldsymbol{b}$ represent cumulative incurral-to-payment lag factors, in which case $\boldsymbol{L}$ should contain only losses paid through the valuation date. This amounts to defining report date as payment date. In the absence of report or payment dates, it is possible to estimate $\boldsymbol{b}$ from a single issue-to-incurral triangle, simultaneously with the earnings factors, though we do not discuss this further here.

There are then several possible procedures for completing $\boldsymbol{L}$ to $\boldsymbol{L}^c$, and estimating earnings factors:

- *Conventional loss development* applied directly to $\boldsymbol{L}$ (and possibly involving $\boldsymbol{E}$) yields first a vector of lag factors $\boldsymbol{f}=(f_j)$, then a set of ultimate losses by month i, and finally the future values of L_{ij} for each cell i,j. The factors f_i will not usually be satisfactory for SSAP 65 Test 2 analysis because they confound the effects of cancellations and losses. Moreover, the method fails for the issue-versus-incurral lag triangle $\boldsymbol{L}$ unless all losses are reported in the month of incurral, as the recent diagonals of $\boldsymbol{L}$ will be deficient because of unreported losses. A crude but common workaround is to suppress the last few diagonals, which sacrifices potentially useful recent information. A better approach is to adjust the latest diagonals for the expected unreported fractions, setting $L^a_{ij} = L_{ij} \ / \ b_{(n+2-i-j)}$, but the resulting lag factors f^a_j still confound cancellations with losses.
- *Exposure-adjusted loss development* completes the triangle $\boldsymbol{R}$ of loss ratios to inforce exposure, adjusted for the expected fraction reported to date, to the rectangle $\boldsymbol{R}^c$, where $R_{ij} = L_{ij}/E_{ij}*b_{(n+2-i-j)}$. This yields a vector of lag factors $\boldsymbol{g}=(g_j)$, which now reflect the emergence pattern of losses on a no-cancellation basis. If $\boldsymbol{R}^c$ is eventually converted to $\boldsymbol{L}^c$, the cancellations are accounted for by the decline in the exposures $\boldsymbol{E}$. This step, which is not always necessary if the lag factors are the main result of interest, must include the previously unreported fraction of "known" cells as well as the entirety of "future" cells.
- *Residual loss development* completes the triangle $\boldsymbol{R*}$ of loss ratios to inforce exposure, adjusted for the expected fraction reported to date *and for the expected fraction earned in each period j*, to the rectangle $\boldsymbol{R*}^c$, where $R*_{ij} = L_{ij}/E_{ij} *b_{(n+2-i-j)}*A_{ij}$, where $\boldsymbol{A}=[A_{ij}]$ gives the fraction of losses incurred in month i expected, *a-priori*, to emerge at lag j. This yields a vector of residual lag factors $\boldsymbol{h}=(h_j)$, which is multiplied by A_{ij} to produce a *matrix* of final lag factors $\boldsymbol{G}=[g_{ij}]$, on a no-cancellation basis.

The *a-priori* earnings matrix $\boldsymbol{A}$ may be derived by applying a formula to each contract and taking a weighted sum over the contracts remaining in force in each cell. Normalization over each row may be deferred until inside the loss development calculations.

For automobile service contracts, the Kerper-Bowron (KB) exposure model [1],[2] is an excellent basis for deriving $\boldsymbol{A}$. For contracts of other types (usually simpler), any reasonable earnings pattern, such as pro-rata after manufacturer's warranty, may be used; normally subject to a minimum positive value so that some earnings are possible at any lag.

Note that the technique described here is independent of the method used to estimate the earnings factors $\boldsymbol{f}$ from the original $\boldsymbol{L}$ and $\boldsymbol{E}$, or $\mathbf{g}$ from $\boldsymbol{R}$, or $\boldsymbol{h}$ from $\boldsymbol{R*}$. Typically this will be some member of the chain-ladder family, with suitably-chosen depth of averaging, weighting, tail factors, and so forth, but it could be, e.g., a regression model. Similarly, the technique is independent of the method used to *complete* the triangles. Typically this will be Bornhuetter-Ferguson, with Cape Cod expected loss ratios, or with Gluck decay factors giving a spectrum of possibilities from pure chain-ladder through pure Cape Cod. These methods are usually very satisfactory for Warranty business, with its characteristic high frequency and narrow size-of-loss distribution.

The Test 2 UPR factors, to be applied to gross inforce premium, are the reverse cumulative sums of the vector $\mathbf{g}$ or of the rows of the matrix $\mathbf{G}$.

If a collection of contracts is perfectly homogeneous in earnings pattern, and losses emerge "noiselessly", then residual loss development will produce the same completed loss rectangle $\boldsymbol{L}^c$ as exposure-adjusted loss development, independent of $\boldsymbol{A}$, provided the rows of $\boldsymbol{A}$ are all proportional to each other. Each row of $\boldsymbol{G}$ from residual loss development will then be identical to the vector $\boldsymbol{g}$ from exposure-adjusted loss development. In the normal situation with random fluctuations in emerging losses this will no longer be true, in general, for the chain-ladder estimator. For any $\boldsymbol{L}$, a constant $\boldsymbol{A}$ will produce results from developing $\boldsymbol{R}^*$ identical to those from developing $\boldsymbol{R}$, so simple exposure-adjusted loss development is a special case of residual loss development, and the two methods may be handled by common program code.

The matrix $\boldsymbol{A}$ should normally be of full length to cover the terms of all contracts in the data, even if the data itself is immature, for reasons explained in Section 2.4.2 below.

If residual loss development is to have its expected beneficial effects, then the *a-priori* earnings factors in $\boldsymbol{A}$ should in fact reflect knowledge of the contracts included in each row of $\boldsymbol{L}$. Usually they will be averages over those contracts of formula earnings factors, or of earnings factors corresponding to UPR "strings" assigned to the contracts on an administrative system. Since they represent the expected emergence pattern of *losses* for that issue month, these averages ideally should be weighted not by contracts (which would ignore differences in expected loss costs) nor by actual premiums (which would ignore differences in expected loss ratios) but instead by theoretical premiums proportional to expected loss cost. Expected loss costs differ for two main reasons: the products insured (for example autos by manufacturer, make, and model) and the contract provisions (term, manufacturer's warranty, etc.). When the mix of products is not homogeneous over time, it may be necessary to derive an expected loss cost for each contract, using, for example, hierarchical credibility over the relevant characteristics. When just the contracts differ over time, it may be sufficient to use contract relativities estimated simultaneously with the KB or other *a-priori* earnings factors; this is part of the All-Terms Factors (ATF) model described in [2].

2.4 Advantages

Residual loss development has several useful features that improve or facilitate the estimation of earnings and UPR factors for long-duration contracts.

2.4.1 Handling of Heterogeneous Data.

Residual loss development circumvents the problem of imperfectly homogeneous data, with mix changing over time, precisely because $\boldsymbol{A}$ is a matrix, with a different *a-priori* emergence pattern for each issue month. If $\boldsymbol{A}$ correctly captures the changes in expected earnings patterns by issue month, these will be correctly propagated in the final development factors g_{ij} and the projected future losses L_{ij}. Each row of $\boldsymbol{G}$ will be a better representation of the earnings pattern of its contracts than the average vector of factors $\boldsymbol{g}$ that would have been produced by development of loss ratios to inforce premium alone. The *residual* factors h_j are still averages across issue months; the loss development process by itself cannot be expected to detect or compensate for heterogeneity. It is the matrix $\boldsymbol{A}$ that converts $\boldsymbol{h}$ into the matrix $\boldsymbol{G}$. Any positive matrix $\boldsymbol{A}$ will mechanically produce results, but only a reasonable $\boldsymbol{A}$ will produce reasonable results. Fortunately it is usually easy to construct a reasonable $\boldsymbol{A}$.

In this discussion we confine our attention to the estimation of earnings factors, not the completion of the loss triangle proper. But it goes without saying that residual loss development can improve the cell-by-cell accuracy of the completed loss triangle $\boldsymbol{L}^c$, since $\boldsymbol{G}$ contains a separate set of factors for each issue month, rather than a single average set.

2.4.2 Simplification of Tail Projections

Tails are necessary in projecting earnings factors for long-duration contracts whenever the available data is shorter than the longest term of the contracts in question. Projecting a tail usually requires considering its starting lag, its length, its shape, and its weight relative to the preceding known lags (or the number of such "lookback" lags to average to estimate the weight) [2]. Specifying length and shape requires information beyond that contained in the loss triangle, and this information must be passed as parameters to a program executing the loss development calculations. This complicates the logic of that program and of any programs calling it. Residual loss development, on the other hand, already captures all the information necessary to specify the shape of a tail in the *a-priori* earnings factor matrix $\boldsymbol{A}$, provided $\boldsymbol{A}$ includes factors for both known and projected lags. The tail of the *residual* earnings factors may simply be taken as constant (which gives pro-rata UPR factors). The result will be a separate tail for each row in the final matrix of earnings factors $\boldsymbol{G}$, and each such tail will have the shape specified by the corresponding row of $\boldsymbol{A}$.

2.4.3 Development of Sparse Triangles

When the data in a loss triangle is sparse, earnings factors estimated by loss development may be very irregular. A common solution to this problem is to take a weighted average of the raw earnings factors and a set of reference factors, usually derived by formula or from a larger volume of related experience. As with tail shapes, having to pass reference factors as separate parameters may complicate the logic of the loss development program and of all programs calling it. Residual loss development circumvents this problem by including all necessary reference factor information in the matrix $\boldsymbol{A}$. The specified "credibility" weight is applied to the measured residual earnings factors, and its complement to a set of constant factors. The result will be a separate set of smoothed factors in each row of the final matrix of earnings factors $\boldsymbol{G}$.

2.5 Examples

2.5.1 Construction of illustrative triangles

To illustrate the above concepts we start with the following 12 x 12 matrix $\boldsymbol{E}^c$ of inforce exposures, both known and (already) projected; $\boldsymbol{E}^c$ incorporates growth, shows the effect of a moderate amount of cancellations, and, for verisimilitude, includes some randomness in written premiums and cancellations. A simplifying assumption here is that each month's writings take place at the beginning of the month, so that the first column represents written exposure and the remaining eleven columns represent exposure still in force after cancellations to date:

	0	1	2	3	4	5	6	7	8	9	10	11
1	216054	194448	183646	177164	172843	170683	168522	166361	164201	162040	162040	162040
2	179896	161907	152912	147515	143917	142118	140319	138520	136721	134922	134922	134922
3	206126	185513	175207	169023	164901	162839	160778	158717	156656	154594	154594	154594
4	254353	228918	216200	208570	203483	200939	198396	195852	193309	190765	190765	190765
5	208020	187218	176817	170576	166416	164336	162255	160175	158095	156015	156015	156015
6	195239	175715	165954	160096	156192	154239	152287	150334	148382	146430	146430	146430
7	236256	212631	200818	193730	189005	186642	184280	181917	179555	177192	177192	177192
8	211880	190692	180098	173742	169504	167385	165267	163148	161029	158910	158910	158910
9	242908	218617	206472	199184	194326	191897	189468	187039	184610	182181	182181	182181
10	218130	196317	185411	178867	174504	172323	170142	167960	165779	163598	163598	163598
11	270039	243035	229534	221432	216032	213331	210631	207930	205230	202530	202530	202530
12	263241	236917	223755	215858	210593	207961	205328	202696	200063	197431	197431	197431

In the present context we have the luxury of specifying the exact parameters generating the loss triangle $\boldsymbol{L}$. In addition to the exposure (the known part $\boldsymbol{E}$ of $\boldsymbol{E}^c$) these include a loss ratio, a loss emergence pattern for each issue month, and the cumulative fractions of losses reported as of each lag from incurral. To illustrate the effect of heterogeneity in a particularly simple way, we initially take all of these remaining parameters as fixed or "noiseless": a loss ratio of 70% for all issue years, cumulative fractions reported of **(0.3 0.7 0.9 1 1 1 …)**, and two separate loss emergence patterns, one for the first six months and one for the last six months, respectively as follows:

1	2	3	4	5	6	7	8	9	10	11	12
0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525

The discontinuity in earnings patterns might exist because the manufacturer has lengthened its factory warranty. The resulting $\boldsymbol{L}$ becomes:

	1	2	3	4	5	6	7	8	9	10	11	12
1	2291	2062	1948	3758	9166	14482	17874	17644	17415	13921	6015	2062
2	1908	1717	1622	3129	7632	12058	14882	14692	13051	9015	2146	0
3	2186	1968	1858	3585	8745	13817	17052	15150	11630	4427	0	0
4	2698	2428	2293	4424	10791	17049	18938	14541	6151	0	0	0
5	2206	1986	1875	3618	8825	12549	12046	5096	0	0	0	0

6	2071	1864	1760	3396	7455	9161	4845	0	0	0	0	0
7	2803	2523	2383	2069	1570	1329	0	0	0	0	0	0
8	2514	2262	1923	1443	603	0	0	0	0	0	0	0
9	2882	2334	1715	709	0	0	0	0	0	0	0	0
10	2329	1630	660	0	0	0	0	0	0	0	0	0
11	2243	865	0	0	0	0	0	0	0	0	0	0
12	937	0	0	0	0	0	0	0	0	0	0	0

This $\boldsymbol{L}$ is of course much smaller than the typical triangle analyzed for Warranty business, which might be of size 120 x 120 with contracts of term 72 or 84 months, but this should fit more comfortably on the reader's screen. We assume here that all contracts contributing to $\boldsymbol{L}$ have term 12 months, though some may expire earlier because of "miling out". Because of our assumption that contracts are written at the beginnings of months, just 12 columns are needed here.

Our known discontinuity in the earning pattern of the contracts being written might be expected to have been accompanied by a discontinuity in expected loss costs and therefore in either rates or loss ratios or both. For simplicity in this illustration we assume that the rates were adjusted to keep the loss ratios constant and that the effect is therefore buried in the growth of written exposures and is of no consequence to the model.

Here we identify report date with payment date and assume that our triangle $\boldsymbol{L}$ contains paid losses only, and that we have a separate triangle $\boldsymbol{P}$ of paid losses by incurral month versus payment lag, from which we have already estimated payment lag factors of (**0.3 0.4 0.2 0.1 0 0 …**) and the corresponding cumulative reported fractions **(0.3 0.7 0.9 1 1 1 …).**

2.5.2 Conventional loss development

From $\boldsymbol{L}$ alone we may apply conventional loss development (chain-ladder, loss weighted, no judgment adjustments) to obtain the following vector $\boldsymbol{f}$ of conventional lag factors:

0.0364 0.0301 0.0269 0.0420 0.0969 0.1524 0.1731 0.1525 0.1296 0.0985 0.0425 0.0190

The reader may confirm this calculation (and those to follow) using his or her preferred loss reserving software, or may consult Appendix B for the algorithms expressed as J language code.

These factors confound the effects of cancellations with the effects of the underlying loss emergence pattern and of the lag in reporting losses; they will not do to derive UPR factors. It is possible to correct for the effect of unreported losses at this stage while leaving the effects of cancellations and loss emergence pattern commingled; this yields an improved vector $\boldsymbol{f}'$:

0.0218 0.0197 0.0186 0.0308 0.0739 0.1219 0.1520 0.1501 0.1481 0.1316 0.0731 0.0585

This may be useful for a quick projection of future incurred losses, but is still not suitable for deriving SSAP 65 Test 2 UPR factors.

2.5.3 Exposure-adjusted loss development

To adjust for declining exposure as well as for unreported losses, we convert $\boldsymbol{L}$ into the triangle $\boldsymbol{R}$ of loss ratios to expected reported fractions of inforce exposure:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0.1061	0.1061	0.0955	0.0530	0.0424
2	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0.1061	0.1061	0.0955	0.0530	0
3	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0.1061	0.1061	0.0955	0	0
4	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0.1061	0.1061	0	0	0
5	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0.1061	0	0	0	0
6	0.0106	0.0106	0.0106	0.0212	0.0530	0.0848	0.1061	0	0	0	0	0
7	0.0119	0.0119	0.0119	0.0119	0.0119	0.0237	0	0	0	0	0	0
8	0.0119	0.0119	0.0119	0.0119	0.0119	0	0	0	0	0	0	0
9	0.0119	0.0119	0.0119	0.0119	0	0	0	0	0	0	0	0
10	0.0119	0.0119	0.0119	0	0	0	0	0	0	0	0	0
11	0.0119	0.0119	0	0	0	0	0	0	0	0	0	0
12	0.0119	0	0	0	0	0	0	0	0	0	0	0

The "noiselessness" of $\boldsymbol{L}$ and the differences between its first and last six months are obvious here. Developing $\boldsymbol{R}$ yields a vector $\boldsymbol{g}$ of lag factors adjusted for both declining exposure and unreported losses; these may be taken as a reasonable candidate for average earnings factors:

0.0171	0.0171	0.0171	0.0294	0.0720	0.1201	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606

They differ from the previous vector $\boldsymbol{f}$ mainly in the early lags, where cancellations are concentrated.

Here, in applying chain-ladder, in averaging each column of development factors, the weights are the denominators (which are themselves loss ratios) multiplied by the numerator inforce exposures (see [2]). This properly recognizes the volume of experience contributed by each issue month, which otherwise would be flattened out when taking loss ratios.

2.5.4 Residual loss development

For residual loss development, we imagine that the actuary knows that the earnings pattern changes after six months – perhaps by applying a formula to the individual contracts – but that this knowledge is imperfect, so that the actuary's *a-priori* loss emergence patterns are not quite the same as the patterns underlying the actual data; we take his or her matrix $\boldsymbol{A}$ to be the following:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
2	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
3	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
4	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
5	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
6	0.0204	0.0204	0.0204	0.0408	0.0816	0.1224	0.1361	0.1361	0.1361	0.1224	0.0952	0.0680
7	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395
8	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395
9	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395
10	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395
11	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395
12	0.0233	0.0233	0.0233	0.0233	0.0233	0.0465	0.0930	0.1395	0.1550	0.1550	0.1550	0.1395

For residual loss development we divide ***R*** by **A** to obtain the following triangle ***R****:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0.7795	0.7795	0.7795	0.5568	0.6236
2	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0.7795	0.7795	0.7795	0.5568	0
3	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0.7795	0.7795	0.7795	0	0
4	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0.7795	0.7795	0	0	0
5	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0.7795	0	0	0	0
6	0.5197	0.5197	0.5197	0.5197	0.6496	0.6929	0.7795	0	0	0	0	0
7	0.5102	0.5102	0.5102	0.5102	0.5102	0.5102	0	0	0	0	0	0
8	0.5102	0.5102	0.5102	0.5102	0.5102	0	0	0	0	0	0	0
9	0.5102	0.5102	0.5102	0.5102	0	0	0	0	0	0	0	0
10	0.5102	0.5102	0.5102	0	0	0	0	0	0	0	0	0
11	0.5102	0.5102	0	0	0	0	0	0	0	0	0	0
12	0.5102	0	0	0	0	0	0	0	0	0	0	0

Developing ***R**** (with LDFs weighted as described above) yields the residual lag factors ***h***:

0.0680	0.0680	0.0680	0.0680	0.0824	0.0889	0.1010	0.1010	0.1010	0.1010	0.0721	0.0808

Multiplying ***h*** by ***A*** yields a matrix of lag factors ***G***, adjusted for declining exposure, unreported losses, and expected emergence patterns:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
2	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
3	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
4	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
5	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
6	0.0153	0.0153	0.0153	0.0307	0.0743	0.1204	0.1518	0.1518	0.1518	0.1366	0.0759	0.0607
7	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258
8	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258
9	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258
10	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258
11	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258
12	0.0176	0.0176	0.0176	0.0176	0.0214	0.0462	0.1048	0.1572	0.1747	0.1747	0.1248	0.1258

The average of these earnings factors, weighted by exposure and normalized to total 1, is:

0.0166	0.0166	0.0166	0.0237	0.0460	0.0807	0.1267	0.1547	0.16400	0.1569	0.102	0.0954

In this artificial example we know the earnings factors underlying the construction of ***L***:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
2	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
3	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
4	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
5	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
6	0.0152	0.0152	0.0152	0.0303	0.0758	0.1212	0.1515	0.1515	0.1515	0.1364	0.0758	0.0606
7	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525
8	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525

9	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525
10	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525
11	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525
12	0.0169	0.0169	0.0169	0.0169	0.0169	0.0339	0.0847	0.1356	0.1695	0.1695	0.1695	0.1525

with normalized average

0.0161	0.0161	0.0161	0.0232	0.0444	0.0746	0.1159	0.1430	0.1611	0.1540	0.1258	0.1097

from which it becomes apparent that the fact that the actuary chose an *a-priori* earnings factor matrix slightly different from the actual earnings patterns underlying $\boldsymbol{L}$ had some effect, but small, on the final estimated earnings pattern. If the actuary had in fact used the actual earnings pattern as the *a-priori* pattern, the final estimated earnings pattern would replicate it exactly.

2.5.5 Tails

Now suppose we have only the last eight rows and the first eight columns of $\boldsymbol{L}$, but have all twelve columns of the last eight rows of $\boldsymbol{E}^c$ and of $\boldsymbol{A}$. Residual loss development will automatically append a tail to each row of the final projected factors, bringing that matrix to shape 8 x 12:

	1	2	3	4	5	6	7	8	9	10	11	12
5	0.0164	0.0164	0.0164	0.0329	0.0761	0.1245	0.1603	0.1603	0.1280	0.1152	0.0896	0.0640
6	0.0164	0.0164	0.0164	0.0329	0.0761	0.1245	0.1603	0.1603	0.1280	0.1152	0.0896	0.0640
7	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330
8	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330
9	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330
10	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330
11	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330
12	0.0190	0.0190	0.0190	0.0190	0.0220	0.0479	0.1111	0.1666	0.1478	0.1478	0.1478	0.1330

2.5.6 Reference factors

Finally suppose that we replace our original $\boldsymbol{L}$ with a triangle of the same overall design but with random "noise" in each cell, so that the estimated earnings factors by residual loss development are somewhat erratic. By taking a weighted average of *residual* and *constant* earnings factors, we may obtain a weighted average of the *final* earnings factors from experience with the *a-priori* factors $\boldsymbol{A}$. It is convenient to define the weights given to experience as credibility-style factors $\boldsymbol{e}/(\boldsymbol{e}+k)$, where $\boldsymbol{e}$ is the total inforce exposure used in the estimation of each earnings factor and k is a constant selected, at the present time, by judgment. In the following example $k = 500000$. The "noisy" L is

	1	2	3	4	5	6	7	8	9	10	11	12
1	2853	1339	3536	475	7087	9807	3423	21225	9950	25801	3605	1722
2	2624	2877	2569	2911	9244	4584	21792	5848	17088	16688	1757	0
3	3899	3497	2663	7143	8355	26151	31550	24792	14522	1265	0	0
4	1585	3359	3849	6444	12068	16469	33179	12498	2681	0	0	0
5	2115	672	1858	2625	5589	20017	17969	610	0	0	0	0
6	3997	2477	298	1404	12537	10841	6107	0	0	0	0	0
7	5305	2551	3698	1272	824	1435	0	0	0	0	0	0
8	1341	3047	1572	245	1197	0	0	0	0	0	0	0
9	3674	2758	1665	1176	0	0	0	0	0	0	0	0
10	321	903	358	0	0	0	0	0	0	0	0	0
11	1146	807	0	0	0	0	0	0	0	0	0	0
12	596	0	0	0	0	0	0	0	0	0	0	0

The final earnings factors by pure residual loss development are:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
2	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
3	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
4	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
5	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
6	0.0173	0.0168	0.0175	0.0242	0.0676	0.1165	0.1746	0.1267	0.1215	0.2071	0.0524	0.0579
7	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199
8	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199
9	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199
10	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199
11	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199
12	0.0199	0.0193	0.0201	0.0139	0.0195	0.0447	0.1206	0.1313	0.1398	0.2649	0.0862	0.1199

and the normalized average of these factors is:

0.0187	0.0181	0.0189	0.0187	0.0419	0.0781	0.1457	0.1291	0.1313	0.2380	0.0704	0.0910

The final earnings factors by weighted-average residual loss development are:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
2	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
3	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
4	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
5	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
6	0.0181	0.0177	0.0184	0.0283	0.0721	0.1194	0.1635	0.1307	0.1275	0.1631	0.0773	0.0638
7	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310
8	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310
9	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310
10	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310
11	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310
12	0.0207	0.0202	0.0210	0.0162	0.0206	0.0454	0.1120	0.1342	0.1456	0.2069	0.1261	0.1310

with averages

0.0195	0.0191	0.0198	0.0218	0.0446	0.0799	0.1360	0.1326	0.1372	0.1865	0.1034	0.0997

2.4.3 UPR factors

The actuary will usually be interested in the incremental earnings factors $\boldsymbol{G}$ mainly for the fact that their reverse cumulative sums give factors to be applied to gross premium to obtain a UPR with desirable properties. In the context of UPR factors, I should mention the suggestion of John Sopkowicz that this technique might be even more usefully applied to claim counts versus inforce contract counts than to losses versus inforce premiums. This is an excellent point as it eliminates, as a source of noise, differences over time in rate adequacy, and we often do use analysis of claim count emergence patterns in related contexts, such as when using data with multiple terms to derive emergence patterns for each separate term in our All-Terms Factors model, or when analyzing frequency in ratemaking. The technique may indeed be applied whatever the definition of L and E.

Figure 1 compares the average UPR "curves" obtained from $\boldsymbol{L}$ by conventional loss development, conventional loss development adjusted only for unreported losses, exposure-adjusted loss development, and residual loss development, with the actual average UPR curve implicit in the construction of $\boldsymbol{L}$.

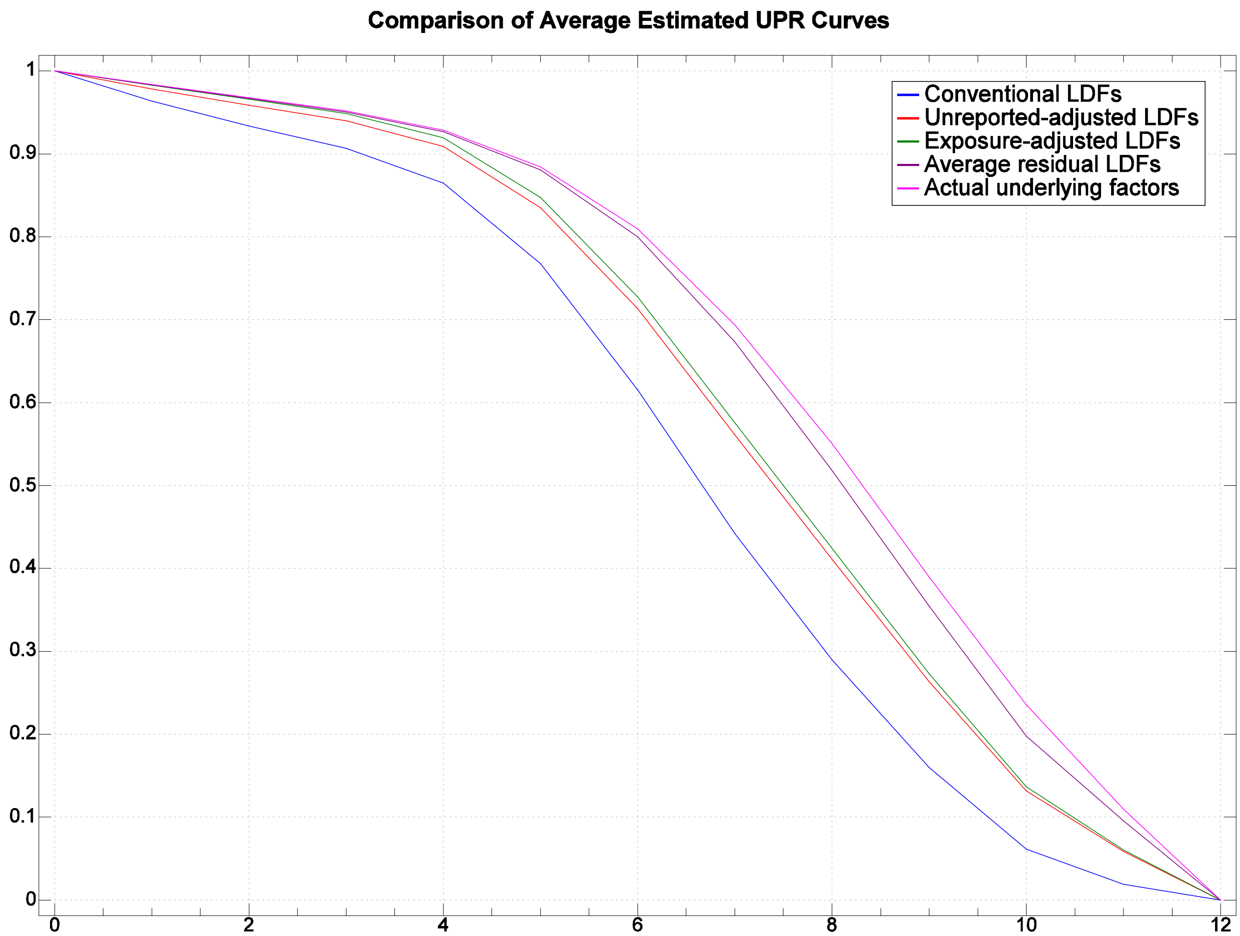

Fig. 1. UPR curves estimated from data in L and E

In this case residual loss development makes the UPR more conservative than simple exposure-adjusted loss development, reflecting the fact that the most recent six months' experience (which does not contribute to the estimated earnings factors by exposure-adjusted loss development at the later lags) is more conservative than the first six months' experience. The residual loss development shown here used the actuary's imperfect estimate of *a-priori* earnings factors; had it been perfect, the residual UPR curve would be identical to the actual curve.

Figure 2 compares the average UPR curves obtained from the last eight rows and first eight columns of $\boldsymbol{L}$, by conventional loss development with tails appended to each row to bring the projection to lag 12. Residual loss development facilitates such tails by providing the expected final shape in the a-priori earnings factor matrix, allowing the residual earnings factors to be extended with a constant tail.

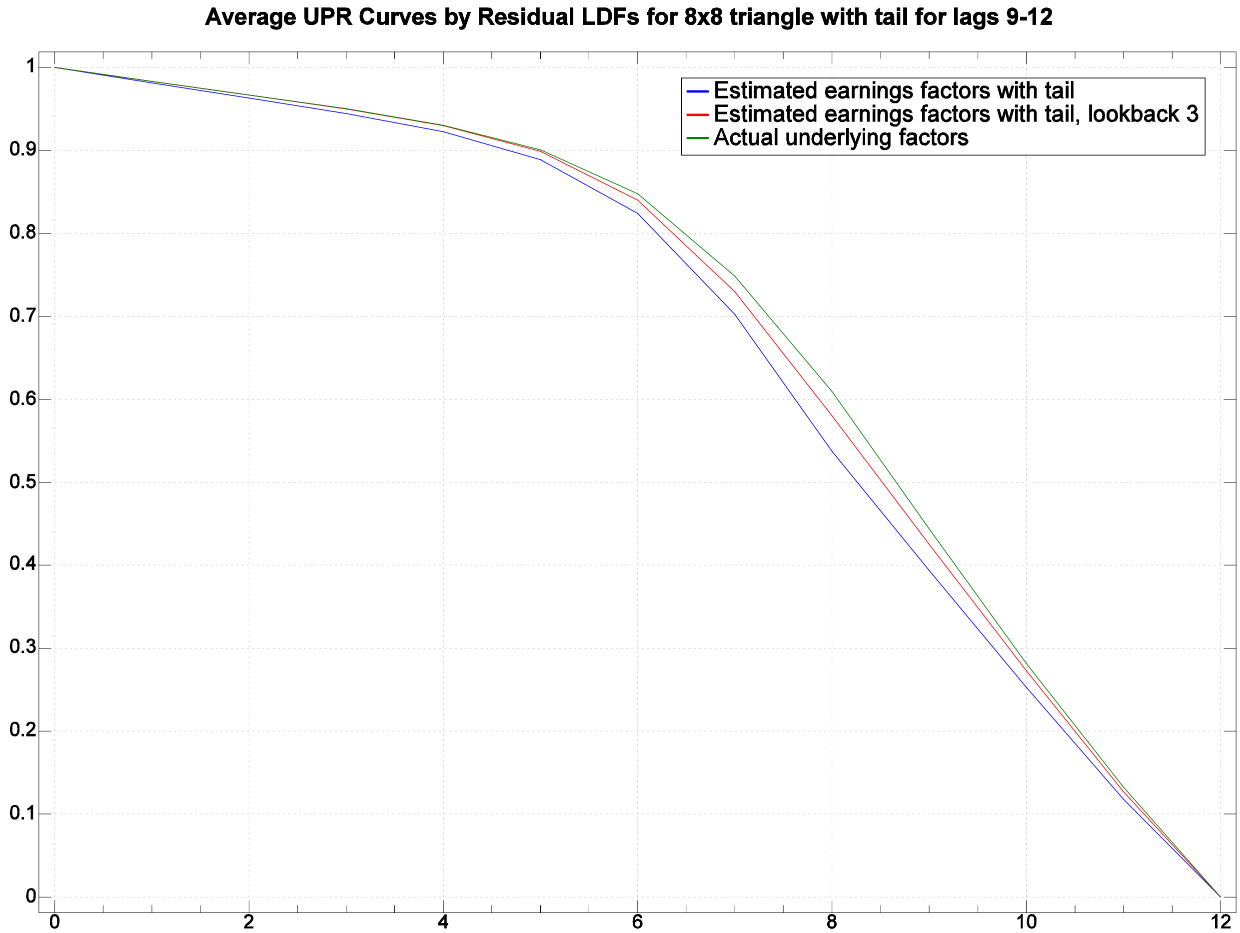

Fig. 2. UPR curve with tail after lag 8 months, compared with actual underlying UPR pattern

In the case at hand the tail is somewhat lighter than the actual last four lags (averaged across the last eight issue months), because the actuary's assumed *a-priori* earnings pattern for the last six months is lighter in the tail than the actual earnings pattern underlying $\boldsymbol{L}$. Because the earnings curve (the negative slope of the UPR curve) is normalized to total 1, the lighter tail causes earnings to increase at the earlier lags and makes the entire UPR curve less conservative. The differences between $\boldsymbol{A}$ and the actual earnings pattern of $\boldsymbol{L}$ were pronounced enough to show up in this illustration; in practice the actuary's formula-based $\boldsymbol{A}$ might be closer to the mark. It is also possible to adjust the weight in the tail to reflect the average residual earnings factors for a lookback period shorter than the entire known history of $\boldsymbol{L}$; we show this here with the "lookback 3" curve.

Figure 3 compares average UPR curves developed from a noisy triangle $\boldsymbol{L}^r$, together with the reference factors given by $\boldsymbol{A}$ and a normalized weighted average of the two. Residual loss development makes the use of reference factors particularly easy, since the weighting is done at the residual-factor stage and the complement of the weight is applied to constant factors.

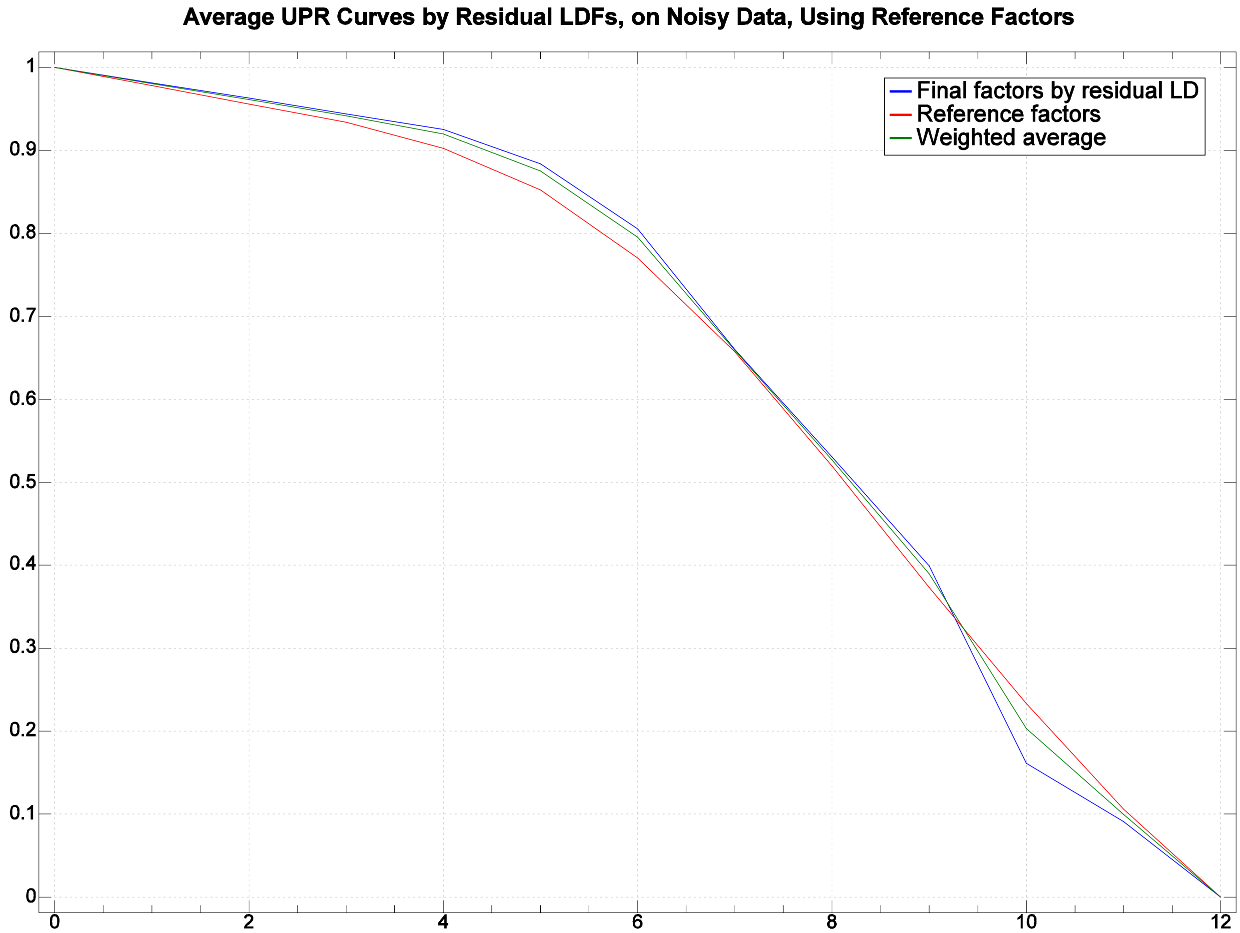

Fig. 3. UPR curve estimated from experience weighted against reference factors.

In this case the weighted average follows experience closely at the early lags, where several issue months contribute to the average development factors, and follows the reference factors at the later lags, where a smaller volume of exposure contributes to the averages.

3. CONCLUSIONS

Residual loss development is an enhancement to the adjustments required in the analysis of long-duration contracts for satisfaction of statutory UPR requirements and for providing accurate performance information to management, ownership, and regulators. It improves the analysis of imperfectly homogeneous segments, of immature segments requiring tail projections, and of segments with small volumes of experience. It may be possible to adapt this technique to policy or accident year analysis of other lines of business, particularly those for which the lag to settlement has a practical maximum.

4. REFERENCES

[1] Kerper, John and Lee Bowron, An Exposure-Based Approach to Automobile Warranty Ratemaking and Reserving, CAS *Forum* 2007: 29-43

[2] Vaughan, Richard, The Unearned Premium Reserve for Warranty Insurance, CAS *E-Forum*, Fall 2014

APPENDIX A. Meaning of Satisfaction in Aggregate for SSAP 65 Test 2

In Section 1.2 we remarked that the requirement that SSAP 65 Test 2 be satisfied in aggregate for a company's long-duration contracts is subject to two interpretations, the *ratio of aggregates*, in which future and ultimate losses are considered only at the aggregate level, or the *aggregate of UPR's*, in which future and ultimate losses are considered separately for a collection of reasonably homogeneous subdivisions of the company's business, and the aggregate Test 2 criterion is built up from the separate criteria for the subdivisions, so that, in particular, satisfaction in aggregate may be guaranteed by satisfaction in detail.

To illustrate the difference, consider two company's books of business with in-force premiums P_0 and P_1, with future expected losses L_0 and L_1, and with expected ultimate losses U_0 and U_1, with all these quantities assumed to be greater than zero. Suppose each company's book of business satisfies SSAP 65 Test 2, with carried UPR equal to P_0L_0/U_0 for the first company and P_1L_1/U_1 for the second. This amounts to treating each company's business as internally homogeneous, or otherwise regarding its UPR as satisfying the ratio-of-aggregates definition. It is certainly reasonable to expect that a merger of the two companies would leave the combined carried UPR in compliance.

But the ratio of aggregates for the combined companies will lead to the same UPR as the aggregate of UPR's if and only if

$$P_0L_0/U_0 + P_1L_1/U_1 = (P_0+P_1)(L_0+L_1)/(U_0+U_1)$$

A bit of algebra shows that this is true if and only if

$$(U_1L_0 - U_0L_1)(P_0U_1 - P_1U_0) = 0$$

which holds if and only if the UPR factors are equal or the expected loss ratios are equal:

$$L_0/U_0 = L_1/U_1 \quad \text{or} \quad U_0/P_0 = U_1/P_1$$

Similarly the aggregate of the carried UPR's will be less than the ratio-of-aggregates Test 2 if and only if both the UPR factors and the loss ratios differ in the *same* direction, and the aggregate of the carried UPR's will be greater than the ratio-of-aggregates Test 2 if and only if both the UPR factors and the loss ratios differ in *opposite* directions.

So the two definitions do not produce the same result except when the expected loss ratios are equal (a reasonable assumption when combining contract-by-contract UPR's within a homogeneous subdivision, but often not valid across subdivisions) or when the UPR factors are identical (not common). But the difference will not often be material. An fairly extreme example might be a book of mature business with average UPR factor about 50% and loss ratio about 70%, to which is added an incipient book of business with UPR factor 1.00 and loss ratio 100%; if the inforce premium for the new book is one-fifth that of the mature business, then the ratio-of-aggregates UPR will be about 4.8% greater than the aggregate of the separate UPR's. If the mature and incipient business were reversed, then the ratio-of-aggregates UPR would be about 3.8% less than the aggregate of the separate UPR's.

In principle the P_i's are known while the L_i's and U_i's must be estimated. Curiously, it may be harder to fix the value of P_i than to estimate L_i and U_i. More precisely, P_i depends on the definition of "in force"; which may attempt to exclude expired contracts or may exclude only cancelled contracts. Common estimators of L_i are indifferent to this choice, as long as each contract is considered in force at least as long as it is possible for a loss to emerge; they will simply return 0 for the estimated earnings factor at any later lag. If P_i is extended to P^*_i by inclusion of some expired contracts with 0 future losses, then $L^*_i = L_i$ while $U^*_i=(P^*_i/P_i)U_i$, so

$$(\text{Test 2})^*_i = P^*_i L^*_i / U^*_i = P_i L_i / U_i = (\text{Test 2})_i$$

But when we combine inhomogeneous blocks of business to get a ratio-of-aggregates UPR, the result will depend on the definition of inforce; for example

$$(\text{Test 2})^*_{agg} = (P^*_0 + P^*_1)(L^*_0 + L^*_1)/(U^*_0 + U^*_1) = (K_0P_0 + K_1P_1)(L_0+L_1)/(K_0U_0 + K_1U_1)$$

where $K_i = P^*_i/P_i$, and this will not, in general, equal

$$(P_0 + P_1)(L_0 + L_1)/(U_0 + U_1) = (\text{Test 2})_{agg}$$

For these reasons we conclude that the only reasonable interpretation of SSAP 65, where a company's business is subdivided into natural and reasonably homogeneous segments, is that the Test 2 criteria may be summed across such subdivisions to obtain the aggregate Test 2 criterion.

APPENDIX B. J Code

The illustrations in this paper were generated by a simple model of the procedure coded in the language J. To make the calculations replicable and to allow the reader to experiment with variations we include the code here.

A bit of a digression is in order here. About 60 years ago the late Kenneth Iverson, then teaching applied mathematics at Harvard, introduced the language APL in a small book entitled "A Programming Language"; this book later won Dr. Iverson the Turing Award. APL is basically a linearized mathematical notation convenient both for conveying algorithms in print and for parsing by a computer as a high level interpreted language. Dr. Iverson later joined IBM and in 1962 brought out the first APL interpreter, on an IBM mainframe. The language was extended by IBM and several other companies and was adopted by many users in the financial and actuarial communities. At one time it was not unusual for papers in the Proceedings of the Casualty Actuarial Society to include a page or two of APL code to illustrate their algorithms. This was especially convenient because of the extreme conciseness of APL.

APL takes arrays such as vectors and matrices as its fundamental objects and, partly for this reason, is admirably suited for many actuarial models such as life contingencies and P/C loss development. Many of APL's primitive functions are structural operations on arrays, not all of which are conveniently expressible in conventional mathematical notation. About 30 years after developing APL, Dr. Iverson, joined by Roger Hui, undertook to systematize the theory of operations on arrays and to create a new language, what APL would have been if he had it to do over again. The result is the language J. This language is a *tour de force*: elegant, concise, comprehensive, uncompromisingly systematic. The J interpreter and development environment are in the public domain, available free for all common computer platforms at the web site jsoftware.com, and are supported by Mr. Hui and by a large community of users.

The author recognizes that another language, R, has become a de facto standard for much work in CAS publications, and for good reason: R has an enormous library of contributed statistical packages, tested and validated, including several specifically actuarial packages. R also operates on arrays, and borrows some ideas from the original APL, but is not nearly as simple, consistent, or thorough in managing arrays as is J. For this reason actuaries looking for a language both to express their thoughts and to build libraries of models – or even just to prototype models eventually to be ported to other languages – would do well to consider J. A few days' experimentation, using the J interpreter interactively and writing small programs, will suffice to get started.

The following code is a simple model of residual loss development and at the same time of conventional loss development, loss development adjusted for unreported losses, and loss development adjusted for declining exposures, all of which may be treated as special cases. The code is in the form of a script – a simple text file – which may be edited by any text editor and parsed by the J interpreter. Comments are preceded by the J word NB.

This is only a fragment of a complete library for triangle analysis, which would also include functions for managing contract and claims data, producing printed reports, etc., along with additional stochastic and deterministic estimators. In anticipation of these purposes this model defines a triangle as a structure containing not only the matrix proper but also additional information such as cumulative and lag status, cell sizes, and dates. The script starts out with a description of the triangle structure, sets the print precision, and loads a couple of addon packages that will be needed.

```
NB. This script contains operations on triangles of actuarial data.

NB. By triangle we mean a matrix of losses or similar quantities, together with
NB. structural and date information, represented as a vector of boxes containing:
NB. a.  Numeric matrix proper
NB. b.  1 if lagged, 0 otherwise [default: 1]
NB. c.  1 if cumulative, 0 otherwise [default: 1]
NB. d.  cell size on first axis [default: 1]
NB. e.  cell size on last axis (must be <=(d)) [default: (d)]
NB. f.  earliest month on first axis, as yyyymm
NB. g.  latest known month on last axis, as yyyymm
NB. h.  latest known or projected month on last axis, as yyyymm

NB. It is assumed that the initial cells along the first axis and the final known
NB. cells along the second axis are complete; if the cell size is greater than 1,
NB. the last cell on the first axis and the first cell on the second axis may be
NB. fragments.

NB. The triangle is so called because its known portion has three "corners"; when
NB. completed to include future periods it takes the shape of a rectangle.

(9!:11)10
require 'plot'
require 'stats/distribs'
```

Next we define some small functions useful in an actuarial context. Notice that the primitive objects of J itself are spelled with ASCII punctuation marks or with one (or occasionally more) letters or punctuation marks followed by '.' or ':'. This makes it impossible to overwrite them with user-defined objects, the names of which cannot include punctuation. These small functions are defined tacitly, that is, with no explicit reference to their left and right arguments x and y, though we refer to the arguments using x and y in the comments. J has extraordinary flexibility in the composition of functions, which facilitates functional programming of this type.

```
NB. Trimming and extending arrays
TrimB=: ]}.~[:+/[:*./\e.~     NB. Trim leading items found in x from vector y
TrimE=: TrimB&.|.             NB. Trim trailing items found in x from vector y
TrimV=: [ TrimE TrimB         NB. Trim items in x from both ends of vector y
TrimX=: (_1{.]),~]TrimE~_1{.] NB. Trim extra copies of last item from end of y
ExtE=: [{.],[#_1{.]           NB. Extend y to length x with copies of last item
ExtB=: ([:-[){.([#1{.]),]     NB. Extend y to length x with copies of first item
MinL=: ]{.~[>.[:#]            NB. Overtake y to give it a minimum length x
RowMat=:,.&.|:                NB. Make vect y into 1-row mat; leave mat unchanged
```

```
NB. Delimited character vectors and arrays of boxed character vectors
UnD=:[:<;._2 }.,({.~:{:)#{.   NB. CV y delimited by first char to vector of boxes
UnD1=:[:UnD,                  NB. CV y delimited by x to vector of boxes
UnD2=: [:>[ UnD1&.> LF UnD1 ] NB. CV y delimited by x and LF to matrix of boxes
UnCSV=: ','&UnD2              NB. CV y from .csv format to matrix of boxes
ToD1=:[:}:[:;[,&.>~[:":&.>]   NB. Vector y of boxes to CV delimited by x
ToD2=:[:;LF,&.>~([:<[:;ToD1)"1  NB. Matrix y of boxes to CV delim by x and LF
ToCSV=: ','&ToD2              NB. Matrix of boxes to .csv format

NB. Operations on months represented as numeric yyyymm:
ThisM=: 3 : '10000 100 #. 2{.6!:0 '''''
IncrM=: 4 : '10000 100#.0 1+10000 12#:(x-1)+10000 12#.10000 100#:y'
PrevM=: _1&IncrM
FollM=: 1&IncrM
LastM=:PrevM@:ThisM
NextM=:FollM@:ThisM
DiffM=: -&(10000 12&#.@:(10000 100&#:)@:<:)

NB. Other supporting functions
Fill=:([:#[){.],[}.~[:#[:,]   NB. Fill vector y with defaults from vector x
Round=: ([:<.0.5+]) : ([*[:<.0.5+%~)  NB. Round y to nearest multiple of x
```

Now we come to some functions that manipulate triangle of the specified structure. These functions are defined explicitly, as a series of lines which may reference the arguments x and y explicitly. The expression 3 : 0 specifies that a monadic function (i.e., with right argument only) or an ambivalent function (may have a right argument only, or both arguments) is defined by the following code up to the first line consisting of a single right parenthesis. The meaning of various control structures should be evident.

```
TriDflts=: 3 : 0  NB. Complete triangle structure y with default values
if. 0=L.y do. y=.,<y end.
't lg cm c0 c1 e k p'=.((0 0$0);1;1;1;1;0;(LastM '');0) Fill y
c1=.c1<.c0
if. e=0 do. e=.>.(1-c1*{:$t) IncrM k end.
if. p=0 do.
    if. lg do.
        if. 3=#$t do. xx=.({:$t)<:((c0%c1)*i.0{$t)+/((c0%c1)*i.1{$t)+/i.2{$t
        else. xx=.({:$t)<:((c0%c1)*i.#t)+/i.{:$t
        end.
        p=.((-.*./0=,t*xx)*c1*({:$t)-1) IncrM k
              NB. If cells below diag all 0, dflt p for lagged tri equals k
    else. p=.(c1*({:$t)->.(1+k DiffM e)%c1) IncrM k
              NB. Default p for non-lagged tri is determined by its length
    end.
end.
z=.t;lg;cm;c0;c1;e;k;<p
)


LagTri=: 3 : 0  NB. Convert y from date to lag triangle
z=.'t lg cm c0 c1 e k p'=.TriDflts y
if. lg=0 do.
    if. 3=#$t do. z=.((i.#t)|.!.0"0 2 ((c0%c1)*i.1{$t)(|.!.0"0 1)"2
t);1;cm;c0;c1;e;k;<p
    else. z=.(((c0%c1)*i.#t)|.!.0"0 1 t);1;cm;c0;c1;e;k;<p
    end.
end.
)
```

```
UnlagTri=: 3 : 0  NB. Convert y from lag to date triangle
z=.'t lg cm c0 c1 e k p'=.TriDflts y
tt=.((>.1+p DiffM e)%c1){."1 t
if. lg=1 do.
    if. 3=#$t do. z=.((-(c0%c1)*i.1{$t)(|.!.0"0 1)"2 (-i.#t)|.!.0"0 2
t);0;cm;c0;c1;e;k;<p
    else. z=.((-(c0%c1)*i.#t)|.!.0"0 1 tt);0;cm;c0;c1;e;k;<p
    end.
end.
)


CumTri=: 3 : 0  NB. Make triangle y cumulative
tri=.'t lg cm c0 c1 e k p'=.TriDflts y
if. cm do. z=.tri
elseif. lg do. z=.(<1) 2}(<(1 KnownPart tri)*+/\"1 t) 0}tri
elseif. 1 do. z=.(<1) 2}(<+/\"1 t) 0}tri
end.
)


DiffTri=: 3 : 0  NB. Make triangle y incremental
tri=.'t lg cm c0 c1 e k p'=.TriDflts y
if. -.cm do. z=.tri
elseif. lg do. z=.(<0) 2}(<(1 KnownPart tri)*(}.-}:)"1 (0,.t)) 0}tri
elseif. 1 do. z=.(<0) 2}(<(}.-}:)"1 (0,"1 t)) 0}tri
end.
)


KnownTri=: 3 : 0  NB. Known, or known or projected, part of triangle y
NB. x is 0 for known part, 1 for known or projected part [default: 0]
0 KnownTri y
:
tri=.'t lg cm c0 c1 e k p'=.TriDflts y
if. lg do. z=.($t)(]0}~[:<[{.[:>0{]) LagTri x KnownTri UnlagTri tri
else. z=.((<.(0>.1+(x{k,p) DiffM e)%c1){."1 t);lg;cm;c0;c1;e;k;<x{k,p
end.
)


KnownPart=: 3 : 0  NB. Flags known, or known and projected, part of triangle y
NB. x is 0 for known part, 1 for known and projected part [default: 0]
0 KnownPart y
:
tri=.'t lg cm c0 c1 e k p'=.TriDflts y
($t){.>0{x KnownTri (<([=])t) 0}tri
)


BandTri=: 4 : 0  NB. Band of x diagonals of triangle y, lagged
tri=.'t lg cm c0 c1 e k p'=.LagTri TriDflts y
z=.t*(KnownPart tri)*.-.KnownPart (<(-(x<.#t)*c0) IncrM k) 6}tri
)
```

Now the main function for the purposes of this paper, and the source of most of the illustrated factors::

```
LDFs=: 3 : 0  NB. LDF's, etc, by residual (or simpler) loss development
NB. y is (loss triangle in standard format);(inforce exposures);(incremental
```

```
NB.   vector of report lag factors, OR triangle of losses by incurral month
NB.   versus report lag) [default for last two items: 1;,1]
NB. x is (depth);(vector or matrix proportional to a-priori lag factors);
NB.   (cred constant)(maximum tail lookback);(Boolean vector with flags for
NB.   unreported adjustment, inforce exposure adjustment, and residual adjustment);
NB.   (maximum length of vector of report lag factors, if calculated)
NB.   [default: _;(($%){:$>0{>0{y);0;_;0 0 0;6]
NB. z is (matrix with rows ldfs, uldfs, cfs, residual lag factors);(matrix
NB.   of final lag factors by accident period versus lag);<(vector of incremental
NB.   report lag factors)

NB. The credibility constant K produces credibility factor E%E+K, where E is
NB. the sum of the denominator exposures at each lag, before a-priori adjustment.
NB. For indications from experience only, use a-priori lag factors of length no
NB. greater than the experience and set the credibility constant to zero.
NB. To append a tail to the indications of experience, use a-priori lag factors
NB. extending to the end of the tail and set the credibility constant to zero.

NB. For a weighted average of experience and a-priori factors, set the credibility
NB. constant >0.  If the a-priori pattern includes factors beyond the end of the
NB. experience data, these will automatically be given full weight, thus appending
NB. a tail, while the earlier factors will be a weighted average.

'' LDFs y
:
'tri exp rpt'=.('';'';,1) Fill y
'd ap K lkb flags supp'=.(_;'';0;_;'';6) Fill x
tri=.'t lg cm c0 c1 e k p'=.DiffTri LagTri KnownTri TriDflts tri
'rptadj expadj apadj'=.0 0 0 Fill flags
'm n'=.$t
if. 0 e.$ap do. ap=.($%){:$>t end.
ap=.(%+/)"1 n MinL"1 m ExtE RowMat ap
N=.{:$ap                      NB. Length including tail
if. 0 e.$exp do. exp=.m$1 end.NB. Default exposure is constant
exp=.N ExtE"1 ,.exp           NB. Matrix of exposures
if. 2=#$>{.rpt do. rpt=.(%+/)supp{.RLFs rpt;'' end.  NB. Report lag factors
rfs=.(i.m)|.!.0"0 1 (m,n)$+/\.(-n){.|.rpt  NB. Matrix of reported fractions
E=.((m,n){.exp^expadj)*(rfs^rptadj)*(m,n){.ap^apadj  NB. Adjusted exposures
lrs=.t%E                      NB. Loss ratios to adjusted exposures
lrstri=.CumTri (<lrs)0}tri
w=.}."1 (n{."1 exp^expadj)*rfs^rptadj  NB. Adj to weights for avg ldf's
nums=.w*}."1 d BandTri lrstri NB. Numerators of ldf's
dens=.w*}:"1 d BandTri (<(-c1) IncrM k) 6}lrstri  NB. Denominators of ldf's
ldfs=.((+/nums)%+/dens),1     NB. Weighted average ldf's
uldfs=.*/\.ldfs               NB. Ultimate ldf's
cfs=.%uldfs                   NB. Completion factors
lfs=.(}.-}:)0,cfs             NB. Lag factors
xx=.d BandTri n{."1 exp       NB. Inforce exposure used in obtaining lag factors
Z=.N{.(+/xx)([%+)K            NB. Credibility factors by lag
rfs=.(n$(+/%#)lfs),(N-n)$((+/%#)(-lkb<.n){.lfs)  NB. Reference lfs
   NB.  These are constant at %#lfs for first n factors, then equal the
   NB.  average lfs over the lookback period for any tail.
LFS=.(%+/)(Z*N{."1 lfs)+"1 (1-Z)*"1 rfs  NB. Cred-adj lag factors
flfs=.(%+/)"1 LFS*"1 ap^apadj NB. Final lag factors
z=.(ldfs,uldfs,cfs,:lfs);flfs;<rpt
)
```

This carries the model through the earnings factor stage which is the focus of this paper. Additional functions may be included to estimate loss ratios by issue month, earnings factors from a single issue month versus incurral lag triangle, and future losses cell by cell, to project persistency, cancellations, and refunds, and so forth, and to use these results for such purposes as financial projections, estimates of ultimate loss and refund ratios, and tests of UPR factors for satisfaction of SSAP 65, beyond the scope of this paper. But the code shown above should serve to document the residual loss development algorithm for the actuary with some knowledge of J.