Watch Your TPA: A Practical Introduction to Actuarial Data Quality Management

Aleksey S. Popelyukhin, Ph.D.

Watch your TPA:

A Practical Introduction to Actuarial Data Quality Management

Aleksey S. Popelyukhin, Ph.D.

> *"Dear Cardmember, the 1997 Year End Summary of your account regretfully contained an error: we discovered that one or more of your transactions were "double counted" – please, accept our sincerest apologies for the error and for any inconvenience it may caused you."*
>
> ***Major credit card issuer***

Introduction

We live in the era of information: an enormous amount of information. Information gets collected, stored, processed, summarized and distributed; there are too many opportunities for errors to sneak in. Data is translated, transformed and aggregated so often, that it is inevitable that some results of the data processing are imprecise.

We may experience this data infidelity elsewhere every day. Once in a while, some bank counts every withdrawal twice, some airline issues two tickets for the same reserved seat and some healthcare provider goes broke due to errors in its financial reports. And we are yet to witness the consequences of the "Year 2000 bug".

The actuarial field can not escape the effects of data errors, either. For example, the NCCI has to restate published LDF's every year (compare [1], [2], [3]) due to errors/restatements in the summaries from information providers.

With the proliferation of the Data Warehousing projects, Data Quality issues come into the spotlight: inaccuracies in data become very apparent. The Data Warehouse, as a source of quality data for analysis and the decision-making process ([4]), requires data to be cleaned up before entering the system.

There is extensive literature on the topics of Data Quality Management ([5]), measurement of the value of information ([6]) and data stewardship ([7]), which is highly recommended for reading. However, sources of information on particular problems with **actuarial** data are scarce, and usually not readily available to actuaries ([8]-[10]). This paper, in an attempt to correct that situation:

- *discusses Data Quality concepts and data clean-up processes addressing specific issues of actuarial analysis requirements,*
- *highlights the inevitability of actuarial involvement in data management procedures,*
- *provides practical examples of the Data Quality Shield's filters and routines derived from the study of the data samples from 43 TPA's and*

- *emphasizes that the quest for actuarial data quality does not stop once data are downloaded in the company-wide Data Warehouse or departmental Data Mart.*

Data Quality Shield

According to Andrew Ippilito (see [11]), data has a number of quality characteristics:

- *Accuracy: the measure of the degree of agreement between a data value and a source assumed to be correct.*
- *Completeness: the degree to which values are present in the attributes that require them.*
- *Consistency: the requirement that data be free from variation or contradiction and satisfy a set of constraints.*
- *Timeliness: the extent to which a data item or multiple items are provided at the time required or specified (a degree to which specified values are up to date)*
- *Uniqueness: the need for precise identification of a data record (and data key values).*
- *Validity: the property of maintained data to satisfy the acceptance requirements of classification criteria and the ability of the data values to pass tests for acceptability, producing desired results.*

Data sets which do not satisfy all the quality characteristics constitute a data quality problem. Often a data quality problem requires two separate efforts: a project to correct existing data and a project to correct the cause behind the data problem. In a typical situation, all data sources are accessible, (for example, mainframe legacy systems within one company) and once the faulty source is identified, the fix is feasible.

Unfortunately, the typical insurance/reinsurance company relies on multiple **external** sources for actuarial data. Third Party Claim Administrators (TPA) monthly summary reports (Loss Runs) are a primary examples of such sources (other examples are industry statistics from NCCI, ISO or RAA bulletins). For the purposes of this article, the company's own legacy systems can be considered as one more (self) TPA, as it is usually external to the actuarial departmental Data Mart and is (potentially) subject to the same types of errors.

There is a limited number of available options for eliminating the cause of data problems in an external data source:

- *External: certification of the TPA information systems.*
- *Internal: deployment of a Data Quality Shield.*

A **Data Quality Shield** is an integrated set of standardized routines optimized for every external data source and comprised from pre-load data filters and translators, along with post-load data analysis tools, statistical diagnostics and quality alarms. This type of integration is needed in order to address two specific distinctions of the actuarial data: multiple *external* sources of data (TPA's) and the *time-variant* nature of intended applications (actuarial methods).

The purpose of a Data Quality Shield is to:

- *Establish standards, (discovering and enforcing business rules, including time-variant business rules)*
- *Validate Input (checking that data values satisfy data definitions)*
- *Eliminate redundant data*
- *Resolve data conflicts (determining which piece of redundant, but not matching data is the correct one)*
- *Propagate corrections and adjustments to prior evaluation for the time-variant data*

The Data Quality Shield's goal is to discover business rules for the actuarial data which may serve as a foundation for the testing and certification of TPA systems.

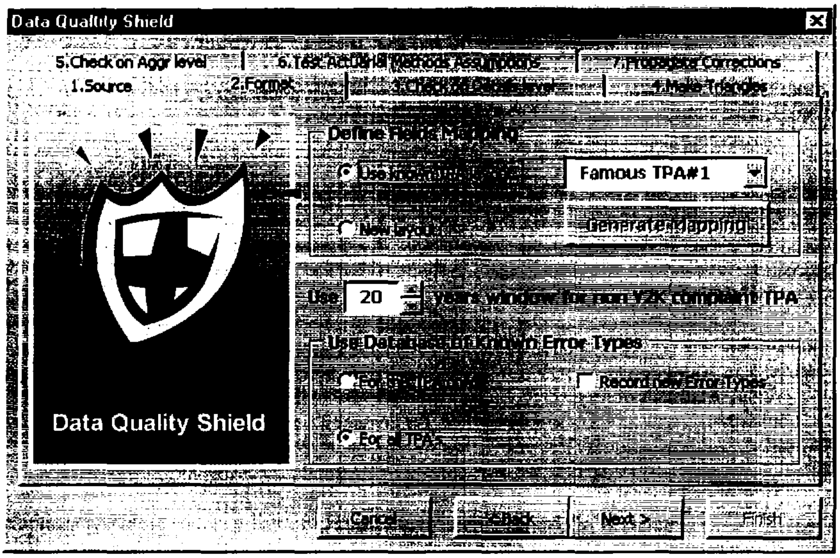

Figure1

In order to create a data quality shield for the actuarial Data Mart in his own company, the author analyzed Loss Runs from more than 40 TPA's and concluded that (currently) no TPA provides data which completely satisfies the Data Quality definition. As a result of his research, the author created a list of typical errors and potential problems and devised a set of routines to identify and fix them.

Typical Problems

As real life experience shows, nothing, not even the most evident data quality requirements, can be taken for granted – even the most obvious actuarial business rule has to be tested and enforced. Every single type of error or deficiency listed below has been detected in at least two TPA Loss Runs.

1. Fields availability

Quality data by definition has to satisfy the completeness and uniqueness requirement: enough fields have to be provided for the possibility to

> *check policy conditions.* For example, the Location field is required if deductible differs by state,
>
> *perform actuarial analysis.* For example, the Report Date field is required if the coverage is "claims-made",
>
> *uniquely identify each record.* For example, the Type of Coverage field is required if the same accident is covered by Worker's Compensation and Employer's Liability.

Of course, fields designated as required can not contain NULL values, that is, be empty for any particular record.

2. Duplicates ("double counting")

- *Source of the problem*

  There are several types of redundant records created with different causes:

  > *True duplicates (same ClaimID).* Possible cause – inaccurate join of the tables with "many-to-many" relationship (for example, the Payments and Recoveries tables with multiple records per claim in both of them joined prior to aggregation).
  >
  > *Duplicate files (different ClaimID, but same Accident Date and ClaimantID).* Possible cause – poor checking against existing records on entry (the TPA system erroneously treats the same claim with a slight variation in claimant name or with a supplied middle initial, as a different claim with its own ClaimID).
  >
  > *Insufficient number of key fields.* Possible cause – missing Claim Suffix or Type of Coverage fields – a deficiency of the Loss Run rather than a whole TPA system problem.

- *Detection*

  > Duplicates can be detected by a simple aggregation (GROUP BY) query with the application of the post-aggregation filtering (HAVING):

```
SELECT ClaimID
```

```
FROM LossRun
GROUP BY ClaimID
HAVING Count(ClaimID) > 1
```

To see all duplicate records rather than a single representative from each group, one can use an embedded query (a query within a query):

```
SELECT *
FROM LossRun
WHERE AccDate In
    (SELECT AccDate
    FROM LossRun
    GROUP BY AccDate, ClaimantID
    HAVING Count(*)>1 And ClaimantID =LossRun.ClaimantID)
ORDER BY Accdate, ClaimantID
```

Records with values matching in any number of fields can be found with the help of such embedded queries. For example, one can detect multiple claims from the same claimant reported on the same date (GROUP BY ReportDate, ClaimantID).

3. Unidentified Occurrences

Depending on the actuarial methodology used to count claims or reinsurance contract conditions, it is crucial to know which groups of claims constitute the same accident or occurrence.

- *Source of the problem*

  Some TPA's do not provide and frequently don't even maintain exact criteria (like Claim Suffix field) for determining occurences, others concatenate Claim Suffix into ClaimID.

- *Workaround*

  In the former case, one can use an embedded query, described above, grouping claims by Accident Date and Location to extract a list of claims, which potentially may constitute the same occurrence. Unfortunately, farther investigation with additional help from the TPA will be required.

  In the latter case, the use of built-in or user-defined string functions (e.g., left() and length()) in a GROUP BY clause of the query may help to break the ClaimID into an OccurrenceID and a Claim Suffix:

```
SELECT left(ClaimID, length(ClaimID)-3), count(ClaimID) AS Claimants...,
sum(Amount) AS TotalPerOccurence
FROM LossRun
GROUP BY left(ClaimID, length(ClaimID)-3)
ORDER BY left(ClaimID, length(ClaimID)-3)
```

4. Recoveries (SIF, salvage & subrogation).

Recoveries may be reported as a separate (from payments) table, may be reported late or may not be reported at all.

- *Source of the problem.*

  While loss payments are made through TPA system, recoveries usually are credited directly to the primary insurer. Thus, at least two sources of data have to be synchronized and related in order to generate net amounts correctly.

- *Workaround*

  To relate tables of payments and recoveries one can use left join (beware of SQL syntax variations in different RDBMSs) of pre-aggregated Loss and Recovery tables (joining non aggregated tables may lead to appearance of duplicates (see "2. Duplicates"):

```
SELECT p.ClaimID, ..., p.GrossLoss, r.Recovery
FROM LossRunPayments AS p, LossRunRecoveries AS r
WHERE p.ClaimID = r.ClaimID (+)
```

5. Consistency of the redundant fields

Some fields are interdependent, and when information in these fields is inconsistent, it is unclear which field to trust. Examples of dependent fields are too numerous to list here, but a few of the most common are:

closed and reopened claims have "last closing" date

open claims have non-zero reserves, closed claims have zero reserves

incurred amount equals paid amount plus outstanding reserve

total paid amount equals sum of indemnity, medical and expense payments (for Worker's Compensation line)

- *Source of the problem*

  Apparently some TPA systems do not have triggers on the closing claim event. Such a trigger is supposed to nullify reserves and insert closing date every time claim is closed.

  As for arithmetic inconsistencies, there are two possibilities: if the TPA system stores redundant amount fields, then system does not react adequately on the changes (adjustments) in the values in the fields; if TPA system stores only independent fields, then it is Report Generator that is broken.

- *Detection and Workaround*

Given "write" access to the data repository and information on which fields are correct, one can execute UPDATE SQL query to restore consistency:

```
UPDATE LossRun
SET Incurred = Paid + OSReserves
WHERE NOT(Incurred = Paid + OSReserves)
```

6. Dummy records

There are several types of redundant records, which do not belong in the LossRun in the first place. These records are filtered out by the TPA's internal tools, and thus remain practically invisible for insiders. However, with the proliferation of online access and digital exchange, these dummy records can be potentially accessed by outsiders, and there is nobody to warn the external user that, for example, record type "99" is a subtotal and has to be filtered out to avoid double counting.

- *Source of the problem*

Subtotals. This is "no-no" of the database design – subtotals should not be stored in the same table as original data: that is what Data Marts with their pre-summarized tables are for.

Dummy claims for "hard to allocate" ALAE. Similar to subtotals, this problem has two causes: one is the inflexibility of TPA system to accommodate all types of allocated payments; a second is the mismatch in the periodicity of summaries of such payments (for example, only quarterly reports from the outer source are available to the TPA)

Test claims – remains of database development projects. This is a development culture problem: systems have to be cleaned up before deployment.

7. Year 2000 compliance

Still a significant issue for many TPA's: 9 out of 43 still allocate just 2 digits for the year value either in their own systems or in the Loss Runs they generate. Another related problem is the handling of NULLs in date fields, for example, in the "Closed Date" field for open claims one can find anything from 01/01/01 to 0 to 11/01/1901 to 1/0/1900 (Excel's representation of 0 as a date).

8. Disappearing claims

Many actuarial methods assume – and not without reason – that the number of claims never decreases in time, or more precisely: a claim once reported will appear on all following Loss Runs. In reality, this assumption does not always hold true.

- *Source of the problem*

  Due to inevitable miscodings, some claims end up in the wrong Loss Run. Once identified as "voided", claims have to be removed from all past Loss Runs (see "13. Propagation of corrections") – that does not always happen.

- *Detection*

  A simple SQL query may help identify claims that "disappeared":

```
SELECT *
FROM LossRun
WHERE Evaluation = PreviousEvaluation
AND ClaimID Not In
    (SELECT ClaimID
    FROM LossRun
    WHERE Evaluation = CurrentEvaluation)
```

9. Non-monotonic losses

Another popular actuarial assumption is that cumulative direct (gross of reinsurance and recoveries) payments are non-decreasing in time.

- *Source of the problem*

  Some drafts that TPA's pay to claimants are voided for some reasons.

- *Detection*

  The so-called self-join SQL query helps to isolate unusual reductions in payments:

```
SELECT LossRun.*
FROM LossRun, LossRun As PrevLossRun
WHERE LossRun.ClaimID = PrevLossRun.ClaimID AND
LossRun.Evaluation = CurrentEvaluation AND PrevLossRun.Evaluation =
PreviousEvaluation AND LossRun.DirectPTD < PrevLossRun.DirectPTD
```

10. Consistent fields definitions

Before validating any business rules and running any tests on TPA data, one has to make sure that fields satisfy standard definitions (i.e., for Statutory Page 14 Data or the ISO statistical plan). Once consistency of field definitions is established, various constraints and validation rules can be tested. For example, one would expect losses to be positive; recoveries to be negative; accident date not to exceed report date, not to exceed closing date, not to exceed evaluation date, etc.

11. Online access and digital exchange

The proliferation of online access to TPA data has created one more type of problem – download integrity. The online session may result in the download of an incomplete set of data or, alternatively, undesirable auxiliary records (see "6. Dummy records"). One of the digital exchange formats, for example, specifies three records of different types for every claim. Thus, every download has to be tested for claim records integrity (every claim has all three records) as well as for completeness of the download (comparison to control subtotals info).

12. Data Entry human errors

An inevitable source of errors cured only by the accuracy of company employees and the system of database self-testing and data entry validation routines.

13. Propagation of corrections

Due to the time-variant nature of Data Warehouses and Data Marts, it is not enough to maintain data consistency in every given time slice – consistency through time is as important. It is crucial, that any adjustment due to miscoding or other error (see "8. Disappearing claims" and "9. Non-monotonic losses") be propagated back to previous evaluations.

Summary

Data sets with even single typical error fail to satisfy data quality definition cited above. Indeed, Loss Runs with error types 6, 8, 9, 10, 12 fail on the requirement for *accuracy*; 1, 2, 3, 7, 8, 11 – for *completeness*; 5, 13 – for *consistency*; 4 – for *timeliness*; 1, 2 – for *uniqueness*; 1, 2, 3, 6, 7, 9 – for *validity*. Unfortunately, in addition to typical problems some sources have their unique (but, nevertheless, malicious) errors.

Legacy systems

All the examples above contains snippets of code written in SQL – a Structured Query Language invented by IBM in order to standardize requests to the database management systems (DBMSs). While every modern DBMS supports SQL, mainframe-based legacy systems usually don't. Absence of SQL support, however, should not be a reason for allowing data errors to slip through.

As long as the reader understands that SQL is just a parsable set of instructions allowing the optimizer to perform a sequence of sorts, scans and lookups, it becomes clear that the same functionality can be achieved using Quick Sort combined with subroutines in PL/1, Cobol or SAS. For example, in order to find and display duplicate records, one would perform a sort placing potential duplicates one after another, and then scan record by record, comparing the previous record with the current one (if records don't match, the user would reset counter of duplicates to 0, otherwise incrementing it by 1; if resulting value of the counter equals 1, the previous record would be placed in the output set; in addition, a positive value of the counter would trigger output of the current record).

In fact, any traditional programming language, being computationally complete, is more capable than SQL. It is just that as an established standard, with it's ease of learning and use, database optimization, and wide availability, SQL has become such a popular language. As the examples above demonstrate, SQL is simple enough for an actuary to run quite a sophisticated query against Data Mart or Loss Run data, yet it so powerful and useful - it definitely deserves to be included in the actuarial syllabus (sometime in the future).

Quality Requirements for Certification process

The existing situation for TPA data quality is unacceptable. In contrast with the explicitly spelled out list of "Year 2000 (Y2K) compliance" requirements, there is no commonly accepted list of "TPA data quality" criteria. And while companies expend a great effort to ensure that all their data sources do satisfy these rigid Y2K requirements, the author is not aware of any significant centralized effort directed to the clean-up of data supplied by TPA's. Similar to the Y2K situation, TPA's have to provide clean data, but they (currently) don't.

It is possible, with the help of actuaries and data administrators, to compile a list of standard tests for the TPA system to satisfy in order to be certified as "actuarially compliant". The typical problems list above may serve as a starting point for such a compilation.

Data that ultimately end up in the actuarial Data Mart move through the following stages, all of which can serve as a source of errors:

- *collection,*
- *storage,*
- *report generation,*
- *communication/distribution.*

For a TPA system to be called "ideal", it has to pass error tests at every stage. Other requirements to the ideal TPA system would include:

- *Flexibility to accept changes: endorsements, adjustments.*
- *Availability of history (previous evaluations).*

As the only stage that involves both the TPA and data recipient, the communication (digital exchange) stage has to be examined most carefully. Any digital interchange standard along with the format should include a list of checks and balances. Introduction of the standard for information exchange without built-in safeguards and a list of testable quality criteria, while possibly eliminating one type of error (e.g., human errors on data re-entry), will inevitably lead to proliferation of other types of error (e.g., duplicates).

An argument for the companies – consumers of TPA data – to be involved in the fixing of TPA problems, even if errors are in their favor, is that errors in their favor are still errors. They are indicators of poor data quality and it's just a matter of time when inevitably they will affect these companies negatively.

Actuaries to the Rescue

While one can rely on the FDA for food quality certification, one should not completely disregard one's own immune system. The same rule of thumb applies to actuarial data quality. No matter how clean and consistent TPA data will become, or whether certification for TPA computer systems will be introduced, it is the data consumer's responsibility to run the last error check and, thus, actuaries will always remain the company's last line of defense against errors.

The list of the typical errors found in TPA's Loss Runs can be sharply divided into two major categories:

- *Violations of static business rules (those which need single Loss Run present to be identified and fixed) and*
- *Violations of time-variant business rules (those which track changes in time and need multiple Loss Runs for identification).*

Static, that is, time-invariant business rules, can be expressed in the Data Mart's metadata format and enforced by validation processes, while "dynamic", or more precisely, time-variant rules, can not. Also, "dynamic" errors require significantly different procedures for discovery vs. correction. While the correction of static data problems has to be and can be addressed by the TPA's, "dynamic" data problems belong to consumer of the information domain, because the level of sophistication, actuarial expertise and customization required for "dynamic" problems resolution is usually beyond TPA's core business – administration of claims.

Given that

> Data Marts provide ***time-variant*** data depository,
>
> TPA's provide data which violate ***time-variant*** business rules,
>
> people who study ***time-variant*** regularities in the insurance companies and, thus, require high quality ***time-variant*** data are called actuaries,

it is clear that they are the best suited professionals to discover *time-variant* business rules and develop routines for protection against *time-variant* errors.

The Data Mart created from TPA data can serve not only as a source of decision-support information, but also as a source of alarms about actuarial quality of the data. The time-variant property of a Data Mart makes it the ideal platform for identifying "dynamic" errors, and actuaries are the most qualified people for designing data quality shields against this type of errors. Once found on the aggregate level, adjustments to the data have to be propagated back in time and granularity. Business rules discovery is an iterative process, with the Data Mart improving after each iteration.

Testing Assumptions of the Actuarial Algorithms

Data quality issues can not be considered separately from the application of the data. Data accumulated in the actuarial Data Mart are supposed to be used in the pricing and reserving algorithms.

Any algorithm – an ordered sequence of operations – has assumptions (explicit or implicit) to be satisfied in order for the result to be correct and reliable. Thus, before starting any calculations, the algorithm's assumptions have to be tested. A good example would be checking whether a given number is non-negative prior to any attempt to extract a square root from it.

Despite the evident importance of the assumption testing and availability of testing routines (see, for example, [12] - [13]), an unacceptably large number of actuaries don't test assumptions. The use of results taken from calculations on untested data will inevitably lead to wrong decisions and misleading conclusions. While the determination of implicit assumptions of actuarial algorithms is an extremely fascinating topic by itself, deserving separate research, this paper is concerned with the data quality aspect of assumption testing.

It turns out that assumption testing is one of the main sources of time-variant business rules. Indeed, a monotonically increasing number of claims is both a time-variant rule and a requirement for the applicability of the Berquest-Sherman algorithm; the same for the assumption of lognormality in ICRFS [14] which coincides with the time-variant rule that requires incremental gross payments to be positive. The failure of the portion of data to satisfy an assumption test can be sometimes caused by data error and lead to discovery of the time-variant business rules, which were violated.

Precise measurement of the impact that data errors have on actuarial algorithm outcomes is beyond the scope of this paper. However, common sense and rough estimates suggest that erroneous claim counts may significantly distort Fisher-Lange method results and large loss frequencies used for pricing; incorrect amounts of losses may affect Chain-Ladder estimates of ultimates; and misreported recoveries may bend loss development patterns, which may result in many negative consequences. Errors in the data may render some of the more advanced actuarial methods inapplicable, potentially leaving actuaries without the best possible estimates. And in a cumulative world of Data Marts, errors do not disappear – they have an undesirable tendency to propagate forward: data points in every evaluation accumulate errors from the previous ones.

Thus, pre-analysis diagnostics of actuarial data, whose purpose essentially is assumption testing, can be viewed as a part of the data quality process and time-variant business rules enforcement, once again highlighting the importance and necessity of the actuarial involvement in it.

Outliers

Another area of actuarial attention should be determination and investigation of the sources of outliers.

Outliers are observations too distant from the expected values. Proper treatment of outliers is important, because the usual regression parameters are significantly affected by them. There are two major ways to treat outliers: robust algorithms and elimination (zero-weight approach).

Robust algorithms help not only avoid distortion of the output, but also determine outliers, which reflect unusual behavior and for which further investigation is necessary.

However, the origin of some outliers is just data error, and these outliers are usually thrown away. Detection, determination and prevention of that type of outliers consequently become an important data quality issue, because instead of throwing away outliers, clean data could provide one more useful observation.

Conclusion

In the world of imperfect external data sources and nontrivial time-variant business rules, the data quality shield's dual approach (pre-load filtering and post-load statistical analysis) is the only practical solution to actuarial data quality problems. Deployment of the data quality shield may significantly improve company's bottom line both directly and indirectly. Potential savings on overpayments to TPA's measured in millions of dollars with significant reduction in company's losses (and consequently, reserves) is not a bad payoff for the design and regular execution of several database queries and custom programs. A fresh review of performance in some business segments supported by correct data may lead to reevaluation of their profitability and may affect important business decisions (the author witnessed exactly that in his own company).

The author views the actuarial process as an inseparable trinity of input, analysis and report phases (see [15]). With this paper, the author tries to demonstrate that for high quality reports based on high quality analysis, actuaries need high quality data; and that nobody is better suited for the determination and enforcement of data quality tests and time-variant business rules than actuaries. Therefore, the author maintains that actuarial involvement in the data management process and data ownership and stewardship is not even a question – it is a tautology.

Clean external data provide a healthy start for the whole actuarial process. To ensure external data quality some type of governing body could to be established. Equipped with a battery of standard quality tests (both static and time-variant) provided by the actuaries, this organization could certify TPA computer systems for use in actuarial applications.

With or without system certification process in place, the situation is steadily improving:

- *Many TPA's, in order to prepare for Year 2000, are updating their systems addressing data quality problems as well.*
- *A proposed electronic data exchange standard (EDI) is now being implemented, requiring TPA's to maintain enough detail for actuarial analysis and accounting calculations.*
- *The move from mainframes to client-server solutions is providing an opportunity for significantly better data quality control.*

Still many problems with TPA data remain. The author hopes that this article will trigger papers from his colleagues from ISO, IDMA and NCCI, where they will share their thoughts on the topic.

Technology today allows more involved actuarial participation in the assurance of the data quality. Modern database management systems, Data Marts and Data Warehouses allow actuaries to access more detail in their data with the most powerful query and analysis tools ever. The author hopes that as a result of reading this paper, some actuaries will establish a standard set of queries, routines and alarms for data quality assurance procedures and will begin a constantly improving data monitoring and correction process.

Epilogue

As for the letter quoted as an epigraph, the author (with the help of his personal data quality shield) discovered duplicates himself, called the bank and triggered corrective action, which benefited everybody.

Acknowledgements

The author would like to thank Leigh M. Walker (ACAS) for his tremendous editing effort; Boris Privman (FCAS) and Jose-Antonio Sobrino-Reineke for initiation into actuarial programming; Karl Möller (FCAS) and Ginda K. Fisher (FCAS) for the opportunity to participate in every stage of the actuarial process; Michael Lipkin, Brian Zell and William Harris for creating an ideal environment for the pure research; and Bart Tesoriero for the insights into claim adjustment process and TPA operations.

Stamford, 1998

Bibliography

[1] NCCI Annual Statistical Bulletin, 1995 Edition

[2] NCCI Annual Statistical Bulletin, 1996 Edition

[3] NCCI Annual Statistical Bulletin, 1997 Edition

[4] Inmon W.H., Using the Data Warehouse, (QED, 1994)

[5] Thomas C. Redman, Data Quality for the Information Age (Artech, 1995).

[6] Michael Brackett, Data Sharing Using a Common Data Architecture

[7] Bonnie O'Neil, Oracle Data Warehousing Unleashed. (Sams, 1997)

[8] Standards and Guidelines for Data Elements. (ISO, 1981)

[9] Insurance Data Quality. Marr, Richard, ed. (IDMA, 1995)

[10] Data Quality Certification Model (Framework and Guidelines) for Insurance Data Management. (IDMA, ISO, 1995)

[11] Data Quality Methods. White paper based on Public Document prepared for US Gov't. (KISMET Analytic Corp., 1996)

[12] Thomas Mack, Measuring the Variability of Chain Ladder Reserve Estimates. (CAS, 1993)

[13] Gary G. Venter, Checking Assumptions of Age-to-Age Factors. (CLRS, 1994)

[14] Ben Zehnwirth, Probabilistic Development Factor Models with Applications to Loss Reserve Variability, Prediction Intervals, and Risk Based Capital

[15] Aleksey S. Popelyukhin. The Actuarial Process from the Data Management Point of View (CAS, 1997)