NISS

Data Quality: A Statistical Perspective

Alan F. Karr, Ashish P. Sanil and David L. Banks

Technical Report Number 129
November, 2002

National Institute of Statistical Sciences
19 T. W. Alexander Drive
PO Box 14006
Research Triangle Park, NC 27709-4006
www.niss.org

Data Quality: A Statistical Perspective

Alan F. Karr and Ashish P. Sanil
National Institute of Statistical Sciences
{karr,ashish@niss.org}

David L. Banks
US Food and Drug Administration
BanksD@cber.FDA.gov

Abstract

We present the old-but-also-new problem of data quality from a statistical perspective, in part with the goal of attracting more statisticians, especially academics, to become engaged in research on a rich set of exciting challenges. The data quality landscape is described, and its research foundations in computer science, total quality management and statistics are reviewed. Two case studies based on an EDA approach to data quality are used to motivate a set of research challenges for statistics that span theory, methodology and software tools.

1 Introduction

Data quality is an old problem that has acquired urgent new dimensions. Once it was largely a scientific issue, with roots in measurement error and survey uncertainty. But for today's world of massive electronic data sets and difficult policy decisions, data quality (DQ) problems can create significant economic and political inefficiencies. Modern research on DQ improvement draws upon multiple disciplines and presents a rich set of scientific, technological, and process control challenges to statisticians.

DQ merits more attention from the statistical community, especially among academics. Faculty engagement in the problem has been virtually nil. At a National Institute of Statistical Sciences (NISS)-sponsored session on DQ at JSM 2002, only two of more than 70 attendees were from universities, and both of them had prior connection with NISS.

This paper is meant to help generate academic attention, in part by framing modern DQ from a statistical perspective. First, in §2, we present a conceptual overview of DQ. Next, in §2, we describe DQ as a multidisciplinary problem that entails ideas from computer science, quality control, human factors research, and the statistical sciences; in any application, these all linked by domain knowledge to the specific context of the problem. In §4 we use show that statistical visualization provides powerful tools for DQ improvement. This point is illustrated by applications in two case studies undertaken by NISS. Finally, based on our survey of the field, we outline in §5 DQ research challenges for statisticians that range from theory and methodology to new software tools.

2 What is Data Quality?

We begin with a definition:

> *Data quality is the capability of data to be used effectively, economically and rapidly to inform and evaluate decisions*. Necessarily, DQ is multi–dimensional, going beyond record-level accuracy to include such factors as accessibility, relevance, timeliness, metadata, documentation, user capabilities and expectations, cost and context-specific domain knowledge.

DQ concerns are problems of large-scale machine and human generation of data, the assembly of large data sets, data anomalies, and organizational influences on data characteristics such as accuracy, timeliness and cost. The impact of poor DQ and the potential benefit of good DQ have implications beyond the ambit of standard statistical analyses.

DQ issues can be dramatic. Some people blame the U.S. government's failure to avert the terrorist attacks of September 11, 2001 upon DQ problems that prevent the easy availability of prompt, accurate, and relevant information from key federal databases. In a different context, the Fatality Analysis and Reporting System discussed in §4 appears not to be of sufficient quality (vehicle make-model data were rife with errors) to support rapid identification of such problems as those associated with Firestone tires on Ford Explorers. At a more mundane level, nearly every major company loses significant income because of data errors—they send multiple mailings to a single person, mishandle claims, disaffect customers, suffer inventory shortfall, or simply spend too much on corrective data processing.

The impetus for improved DQ is especially strong in federal agencies. Managers there are struggling with declining survey response rates and consequent diminished quality. Simultaneously, Congress and the executive branch are using the Government Performance Results Act (GPRA) to require clear linkage between regulation and measurable outcomes. This confluence of a decreasing ability to obtain accurate measurements and increasing management accountability for achieving data-determined goals has compelled Federal managers to address DQ more directly than ever before.

Although the scientific community has developed improved measuring devices, such as automatic sensors that create computer records directly, DQ remains an issue in this setting. Indeed, technological advances may have created a false sense of security. Nonetheless, DQ in scientific research is a different problem from DQ in government and industry.

Industries stand between the Federal government and academic research in terms of DQ. While the quality of their scientific measurements may be generally good, significant problems arise in inventory management, customer service, billing, and marketing databases. Medical data are a particularly thorny problem: they are copious, complex, hard to verify, and entered by many uncoordinated hands. Nonetheless, many problems in industry data have bounded impact (Example: billing errors affect one customer at a time), and there are feedback mechanisms that can support correction (Example: customers report overcharges and wrong addresses).

Figure 1 presents a conceptual view of the DQ landscape. There are two coordinates: the "vertical" represents a data quality process with three principal steps: *Generation*, *Evaluation, Measurement and Improvement* and *Use*. The "horizontal" coordinate represents three levels at which the DQ process operates: on the data themselves (center), at a higher ("meta-") level (left), and at a mechanistic level of software tools.

The entire data handling process is linked to human factors issues that range from generation of the data (Examples: survey respondents, those who complete forms, operators of scientific instruments) through end use of the data, which depends sensitively upon the application and domain knowledge. Every part of the process and every relationship among the parts defines a new problem, one that is usually context-specific.

To explain this conceptual view, we work outward from the Data Level column, illustrating briefly how each step affects DQ. We believe that nowhere are all of these steps performed; indeed, in many DQ applications, some steps are simply impossible.

Design. Effects of design on DQ are perhaps best understood for survey data. Coverage, non-response bias, household bias, frame bias, and respondent bias all affect DQ. The design should enable some estimate of total survey error (or the analogous quantity for non-survey situations).

Collection. The Toxic Release Inventory (TRI) case study in §4.3 epitomizes problems with self-reported data, where human issues of disincentives for data collection are rampant. Possibly in this case (and surely in many others) data collection instruments request information that is not directly needed or easily available. When this burden becomes too great, the quality of reported data declines.

Verification and Review. Ideally, processes should be in place to review and verify data records. For example, in principle one could take a random sample of the records and independently check their accuracy. Although often impracticable, this kind of concern raises the need for software tools that detect and characterize anomalies in the data. Such tools are the focus of the TRI case study in §4.3.

Cleanup and Warehousing are one facet of DQ for which mature software tools exist; see §3.1.

Metrics and Measurement are discussed in §5. To the extent that quantification is the foundation of science, there is no science of DQ since there are no widely-accepted DQ metrics. A metric should capture several features:

- Accessibility: legacy database management systems that use outmoded software and run on old-fashioned platforms may be useless. Even systems with modern software may suffer from poorly designed interfaces.
- Relevance: inertia in data collection processes works against the need for relevant, useful data. As databases age, they often serve different (or more) purposes from those originally intended, or the problem that drove the initial use may change. Relevance might be assessed by checking that each item captured is needed by some clearly-defined class of customers.
- Currency: outdated data can be worse than no data, since it creates false certitude.
- Consistency: if data are reported in multiple ways, or if interacting databases use different definitions, then the utility of the data is reduced.

There are other aspects of DQ that a metric should capture, some of which depend upon the application.

Integration with other databases is increasingly vital but almost invariably difficult. Issues range from differing attribute definitions (Example: death from an automobile accident must occur within 30 days, but from a train wreck attributable death can occur up to a year later) to inability to join relational tables because neither contains a foreign key to the other. Tools for integration, especially relational database management systems (RDBMSs) and record linkage are rather well-developed; see §3.1.

Extraction of Information from data is a prerequisite for decisions. Statisticians ordinarily construe this as “inference,” but a broader perspective is necessary. One aspect is to develop inferential tools that are resistant to poor quality (§5.1). Another aspect is to ensure adequate metadata; without good documentation a database is unusable except by experts. Metadata provides the context that allows users to extract information from the database, and it is especially crucial when diverse customers trying to obtain information on-line.

Decisions drive both the generic need and context-specific requirements for DQ. Decisions are of two kinds: those *based on the data* (Example: government policies) and those *about the data*, which means that

Figure 1 could be filled in with feedback loops. Decisions of both types depend on cost, but despite its importance this has not yet stimulated significant research into the economics of DQ. For example, it would be good to know:

- What cost is needed to achieve a specified level of DQ;
- What are the financial benefits of improved DQ; and
- What are the costs of poor DQ.

To complicate the problem, note that costs may be shifted among multiple stakeholders (Example: more burden on survey respondents can reduce clean-up cost of survey data).

Even though the goal is remote, we believe that ultimately DQ must be given a decision-theoretic formulation (see §5.1), allowing use of tools based on statistical decision theory.

Human factors are a challenging part of DQ. People are the key links in many data generation processes, and the ultimate customers (even if decisions are made on their behalf by another entity). The case studies in §4 illustrate this.

Domain knowledge is central to DQ. Data can be useless for one purpose but adequate for others, and domain knowledge is necessary to distinguish these situations. Anomalies in data (a central theme of §4.3) make no sense in the absence of domain expertise. Possibly the most difficult challenge in creating the data quality toolkit (DQTK) outlined in §5.2 is to automate generic aspects for characterizing and measuring DQ but to build in the right “hooks” for domain knowledge.

Some organizations are trying to formalize aspects of the DQ process. For example, the Bureau of Transportation Statistics (BTS) has tried to evaluate databases using a data quality report card (DQRC), on which a database receives a rating of 2 (good performance), 1, or 0 (substantial failure) on eight criteria that span much of the DQ landscape in Figure 1. One important DQRC criterion concerns whether a database is subject to periodic review by data managers and suppliers. Another criterion checks whether the database is compatible with other databases in the U.S. Department of Transportation; this criterion is similar to the *Integration* box in Figure 1.

3 DQ as a Multi-Disciplinary Problem

Besides problem-specific domain knowledge, DQ rests on three disciplines, as shown in Table 1. Each brings its own perspective. We discuss them in order of increasing familiarity to statisticians.

3.1 Computer Science

Ever since organizations started collecting and storing their data electronically, information technology (IT) departments have served as custodians of the data. Consequently, computer scientists and information technologists have had considerable exposure to some DQ issues, and they have developed the most mature set of technologies to deal with them (from an IT/database perspective). We present a brief sketch of the computer science viewpoint of DQ under two broad categories: *database management* (ensuring correctness in the data collection phase) and *data warehousing* (improving consistency and correctness in the data consolidation phase).

Database Management. Database management is concerned with ensuring data correctness at the collection or entry stage. The general approach has been to devise sound database design guidelines and to

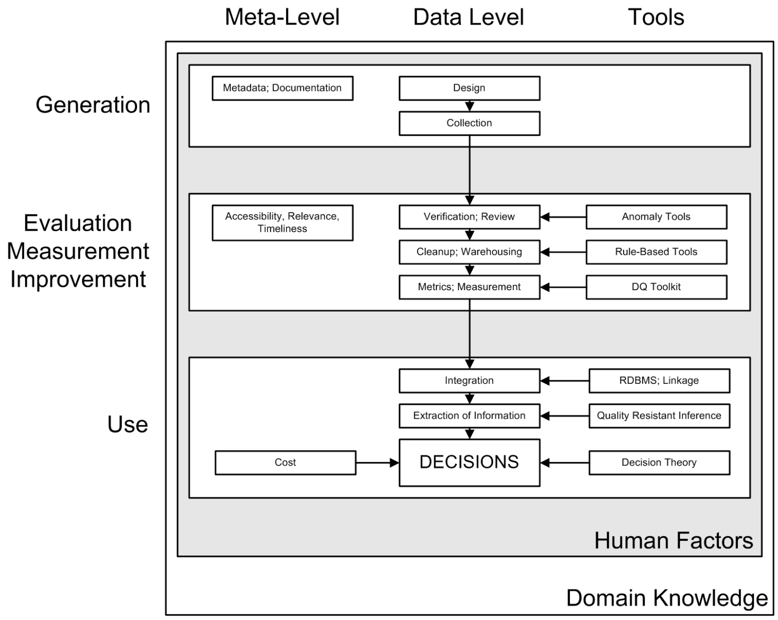


Figure 1: The DQ Landscape. The vertical coordinate represents the DQ process from data generation to data use, in decisions either about or based on the data. The horizontal coordinate, somewhat vaguely, represents different levels of abstraction, from the meta-level (most abstract) through the data themselves to (software) tools for dealing with DQ. The entire process is embedded in human factors and in domain knowledge.

establish correctness-enforcing application development practices. This thinking is often incorporated into development environments of RDBMSs.

Data Entry. Structured Query Language (SQL) [5, 18], the *lingua franca* of the database community, provides a powerful and flexible syntax to create data tables in which data type and other constraints on data attributes (from the metadata specification) are enforced. "Good practice" guidelines exist for designing user-entry forms for data-entry [25]. Example: when a person's contact information is being input, the user should be made to select an enumerated-type data element such as State only via a menu of allowable choices (eliminating mis-spellings, spelling variants and other errors).

Database Design. The central question that drives design considerations for relational databases is how the logical database should be divided physically into separate tables or flat files (as in spreadsheets) in order to minimize duplication and updating anomalies without information loss (i.e., one must be able to link data in individual tables in order to answer queries that span the entire logical database). The theory of "database normalization" provides methodology to solve the design problem [5]. This theory defines a set of increasingly complex "normal forms" (first normal form, second normal form, . . .) such that databases in higher normal forms have fewer sources of anomalies. Database designs that satisfy the third normal form requirements are generally considered adequate.

Transactions and Business Rules. Modern database management systems and, more generally, software frameworks and middleware for enterprise-wide information management include powerful features for ensuring correct and consistent data. For instance, sophisticated On-Line Transaction Processing (OLTP) systems handle extremely complex transactions and are able to enforce elaborate business rules and operational constraints. Recent versions of SQL implement constructs for incorporating non-trivial rules and constraints [18].

Data Warehousing. The warehousing process involves assembling data from a variety of sources and consolidating them into a central data store for future analysis or decision support. Often the data are of poor quality. Even if the data were all of high quality, integration from disparate sources (Example: payroll, manufacturing, and laboratory data), each with their own idiosyncracies and standards, is challenging. The need to incorporate quality checks in the traditional ETL (Extract, Transform, and Load) process is increasingly recognized, and a number of commercial software solutions in the form of add-on modules are emerging (such as Dataflux from SAS [4], Trillium [35] and Evoke [34]).

The DQ issues that the warehousing process attempts to address revolve around identifying data elements that are *invalid* because they violate physical, logical, or metadata-based constraints (which can be specified independently of the data actually observed). In contrast, the EDA-based approach used in § 4 complements the warehousing process since it is geared towards finding anomalies in possibly valid data.

Standardization and Duplicate Removal. Free format data (Example: addresses) may require complex parsing in order to be put into standardized form. Commercial DQ software tools contain procedures to do this parsing, and also metadata on commonly occurring data types, such as state-Zip code relationships, that allow automatic correction (or flagging) of some errors (Example: a mis-specified Zip code, assuming that the state is correct). Sophisticated string matching algorithms can detect, depending on the specificity/accuracy parameter settings, very subtle problems (Example: that the street name `Fxx Craft` should be `Foxcroft`). Following standardization, suspected duplicate records which may be processed manually or in a semi-automated manner, but the current tools are not very powerful.

Record Matching. Integrating data from multiple sources requires correctly linking each record in one database with the corresponding record in another. In many cases, the common (linking) attributes in the two sources are not exactly the same (Example: one database may contain full names and the other only initials and the last name). Record linkage has received attention from computer scientists as well as statisticians, and a number of methods that span a range of complexity and applicability have been devised [43, 44]. Record matching techniques that lie within the IT domain typically rely on key-generation and string matching (probabilistic methods are outlined in § 3.3). String matching methods of varying levels of complexity determine if string-valued attributes from the two sources are “close enough” to be declared a match. The key-generation approach is based on generating a surrogate key for string-valued data that is less ambiguous than the original, and then using the key for record matching.

Constraint Checking. This component of the warehousing process verifies that the data domain, type, and numeric range constraints specified in the metadata are satisfied. Some software tools also allow the user to specify domain-knowledge-based constraints that the data must satisfy.

3.2 Total Quality Management

Statistical quality control has had profound effects on industrial production. Some researchers and practitioners believe that the same ideas and techniques can be applied to DQ. However, inability to model (or even quantify) costs is one reason why the total quality management (TQM) paradigm may be difficult to implement for DQ.

The chain of reasoning is straightforward: Data are a product, with producers and customers; therefore data have both cost and value. In (conceptually) the same way as physical products, data have quality characteristics resulting from the processes by which they are generated. In principle, DQ can be measured and improved. Finally, the same financial incentives that lead to high quality products also apply to DQ.

One approach [9] relates DQ to information quality in a TQM setting: "quality in all characteristics of information, such as completeness, accuracy, timeliness, clarity of presentation" should "consistently meet knowledge worker and end-customer expectations." A key tenet, which is admirable if not always attainable, is that DQ should be present from the start, rather than created by re-work. The need to measure the costs and benefits associated with different levels of DQ is critical in this setting

An alternative approach is called Total Data Quality Management (TDQM). It involves concepts such as multi-dimensional data quality, data quality metrics, evaluation of the user's assessment of DQ, and data production maps [22, 39, 40].

TQM ideas may be applied at the data collection phase by monitoring such characteristics as the number of invalid or nonconforming records generated per day [26]. These can be tracked not only over time, but also against geography or the amount of money expended to acquire the data.

3.3 Statistics

Statisticians have always fought for better DQ. That is why we worry about outliers and long-range dependence and exploratory methods. It is why we study robust statistics, and stress the need to understand the science and to talk to domain experts before undertaking an analysis.

Some statistical contributions to DQ are substantial.

Data Editing. Statistical data editing is the automated process of stepping through the data records and correcting them if they violate pre-specified constraints (edit rules). There are many (sometimes *ad hoc*) methods based essentially on sets of if-then-else rules, as in the warehousing approach (§ 3.1). A formal framework for data editing [13] allows one to specify edit rules or constraints to identify invalid data records (Examples: lists of impossible attribute combinations for discrete data [42] and sets of linear inequality constraints on ratios of continuous variables [17]). Making the edits involves first generating all the implied edit rules from the explicitly specified rules and then solving optimization problems to determine the minimum-change edit for every record that violates some edit rule. There are formidable computational challenges in implementing an automatic edit system, and the development of efficient algorithms and heuristics remains an active area of research [45].

Probabilistic Record Linkage. Probabilistic approaches to the record linkage problem can be traced to [14, 29]. To link records in file **A** with those in file **B**, the Fellegi–Sunter framework provides a method for evaluating the likelihood that pairs of records in $\mathbf{A} \times \mathbf{B}$ are matches, as well as a corresponding optimal linkage rule for specified Type I and Type II errors. This basic idea combined with refinements and heuristics has formed the basis of several record linkage implementations. Extensions of the Fellegi–Sunter methods are still being developed [41, 43].

		Data Settings		
		Government	Business	Scientific
Discipline	Computer Science	⋆⋆	⋆ ⋆ ⋆	
	TQM		⋆	
	Statistics	⋆ ⋆ ⋆		⋆⋆

Table 1: Various disciplines contributing to DQ methodology and their focus on various application domains. (⋆ ⋆ ⋆ ⋆ ⋆ ⇒ frequently applied; “ ”⇒ almost never applied.)

Measurement Error and Survey Methodology. Statisticians involved with survey data have devoted enormous attention to DQ [19]. In this setting, DQ effects arise from modes of collection [6], interviewers [20] and survey design [36]. DQ concerns for surveys have focused primarily on non-response problems [21], coverage biases, [7] and measurement error [1].

4 Case Studies

Two case studies of DQ undertaken by NISS are described here. The first treats the version of the Fatility Analysis Reporting System (FARS) maintained in the Intermodal Transportation Database (ITDB) of the Bureau of Transportation Statistics, and the second addresses the TRI maintained by the US Environmental Protection Agency (EPA). The methods used in handling the case studies illustrate the power that modern visualization tools and a statistical perspective offer for finding, measuring, and correcting DQ problems.

Two case studies cannot span modern DQ in its entirety. Examples of DQ problems encountered by NISS in other contexts include inconsistencies between documentation and data files; cumbersome data representations (Example: tables converted to vectors in the National Transit Database (NTD) [12]); records in which the attributes in which the cannot be compared (Example: multiple paired attributes of the form “(Mode, Expenditure),” also in the NTD, such that bus expenditures by different organizations may in different attributes); and databases with attributes having identical names and meaning (Example: Number of Students Enrolled in Second Grade in 2000–01) but different values, because they came from different data sources at different times.

4.1 Approach and Rationale

The case studies employ an exploratory data analysis (EDA) [37] strategy with a strong component of visualization, augmented by consideration of issues involving relational databases. The emphases are on:

Metadata Characteristics, including information about the database, its structure, tables, and attributes, as well as missing and incomplete values.

Characteristics of Individual Attributes, such as the legitimacy and distribution of their values.

Relationships among Attributes, whose existence may be suggestive of either good or poor DQ. This also includes consistency checks between attributes, possibly in different relational tables.

Relational Characteristics, within the database, such as primary and foreign keys and joinability of tables.

The main message from these examples is that the EDA strategy produces useful insights into the data that can improve quality and increase value to the users.

4.2 Case Study 1: The ITDB Version of FARS

Background. The FARS [27] contains a census of fatal traffic accidents within the 50 States, the District of Columbia and Puerto Rico. Data for each year, which are derived from police and emergency medical system (EMS) reports, are in four files: **FARS-A**, the accident file, **FARS-V**, the vehicle file, **FARS-D**, the driver file, and **FARS-P**, the person file. The FARS is assembled and maintained by the National Highway Traffic Safety Administration (NHTSA)and is available from both the NHTSA Web site and the ITDB. As will be seen below, the two versions differ significantly, possible because "DQ problems are acute when [data] are collected in one organization for use by another" [16].

Metadata Issues. Inconvenient coding and poor data representations are not simply annoyances, but can inhibit analysis. For example, people who use FARS data in conjunction with a geographical information system (GIS) would be served better if the geographical attributes State and County were coded using FIPS codes rather than GSA codes, which are effectively unintelligible (see Figure 3) without a data dictionary. Similarly, in the ITDB, time values are coded as 0147 for 1:47 AM, whereas the NHTSA version of FARS treats hours and minutes as separate attributes. Without correct and cumbersome parsing, subtraction of these values to compute the time difference between two events produces erroneous results.

A pervasive DQ problem is the failure of documentation to distinguish clearly between "original data" attributes and *derived attributes* calculated from original data. At the very least, making this distinction informs users of what might be redundant (or informative!) consistency checks, and also flags some kinds of analyses as inappropriate (Example: those where both derived and original attributes are used as predictors in a regression).

Poor coding of missing and incomplete values, a venerable problem, seriously reduces the usability of the data. In the ITDB version of FARS, missing values are padded with "0" (but this usually means "not applicable"), "9" (which usually means "missing"),"*" and blanks. Aggravating this inconsistency, there are partially missing attributes in the ITDB version of FARS, such as dates of the form *10991999*; here the missing month is replaced by "99." (The NHTSA version of FARS has month, day and year as separate attributes, and so has no partially missing values.) This again unnecessarily burdens the user to parse data values correctly. Figure 2 shows the prevalence of these partially missing values in the **FARS-P** file for 1999. Moreover, legitimate data values become confounded with the missing value code (Example: in FARS, Age = 99 years is indistinguishable from Age = missing). To some degree, these are legacy problems associated with fixed-width attributes, and would vanish if data were maintained in a modern RBDMS.

Metadata should address possible systematic influences on missing data. Figure 3 plots the percentage of missing EMS arrival times by state. A state-to-state effect is apparent; some analyses would be flawed without accounting for it.

Even such seemingly simple metadata as file sizes can help identify DQ problems. Figure 4 shows the sizes of the four FARS files available for download on the ITDB. The small sizes of the files for 1989, 1991, 1992, 1993 1998 and 1999 clearly merit investigation and, as discussed in **Relational Database Issues** below, signal a dramatic DQ problem. Figure 5 reveals that the problem affects all states.

Single Attribute Analyses. A natural first check is to ensure that data values are legitimate (Examples:

(DeathDate, DeathTime)	Number of Records
(0,0)	29,701
(MM**1999,*)	29,091
(99991999,*)	188
(99999999,*)	3
(,*)	643

Figure 2: Summary of various forms of DeathDate in the 1999 FARS.

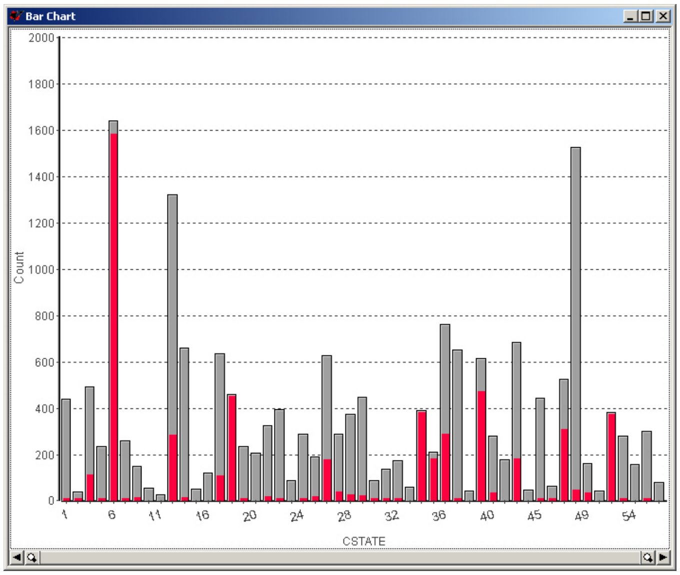


Figure 3: Bar chart showing missing EMS arrival times by state. Accidents having missing times are shown in dark gray. There are clear systematic state-to-state effects. Numerical state labels rather than USPS codes (It is not easy to know that 6 = CA) make the data harder to use.

date values should represent *bona fide* dates, and purportedly positive-valued data should indeed be positive). Consistency of attribute representation must also be checked, especially for databases such as FARSthat are collected from disparate sources with differing data-collection procedures. Problems do exist: for some records in **FARS-A**, the accident time and EMS arrival time have differing representations, usually with one on a 24-hour clock (1602 Hours) and other on a 12-hour clock (0402 PM).

A simple, effective strategy to find anomalous values of individual attributes consists of three steps. First, one should sort and list unique values, which reveals suspicious values and can also indicate subtle DQ problems. To illustrate, looking at extreme values of the derived attribute LTIME in **FARS-P**, the lag time between accident time and death time, reveals several anomalies. Two of the extreme values of LTIME are both 72000 (which means, although this is not obvious, 720 hours and 00 minutes). Closer examination reveals that both of these are associated with the same accident—in State = 54 having Case No. = 321—which took place at 19:27 hrs on 11/7/97. Both fatalities are recorded in the **FARS-P** file with death date

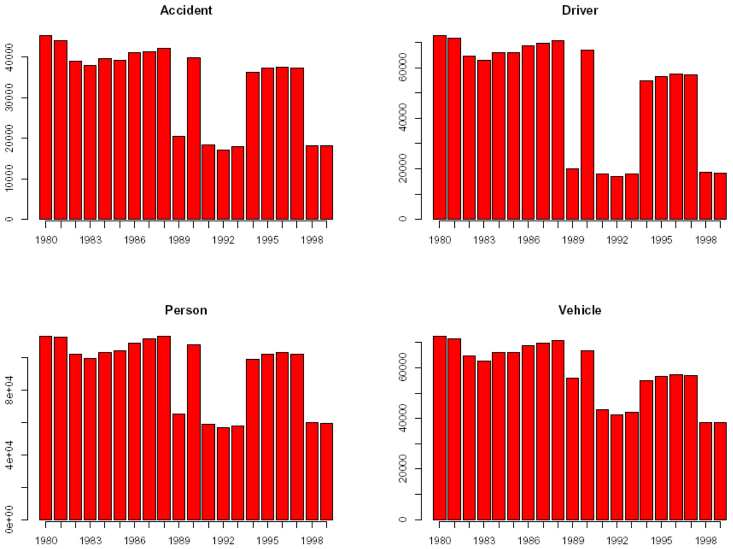


Figure 4: Sizes of FARS files downloaded from the ITDB Web site [3] for the years 1980–1999. The inexplicable deficiencies for some years clearly require explanation.

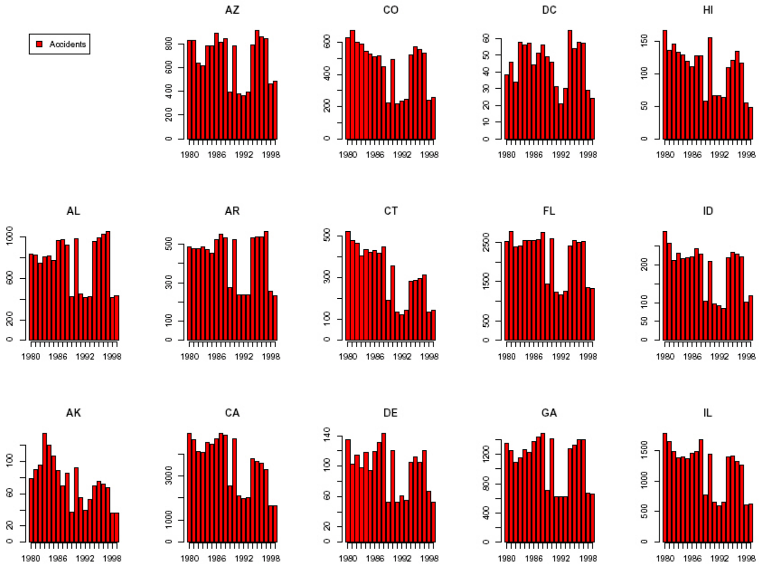


Figure 5: Accident counts by year in **FARS-A** files downloaded from the ITDB Web site [3] for selected states. It is clear that the deficiency problem in Figure 4 affects all states.

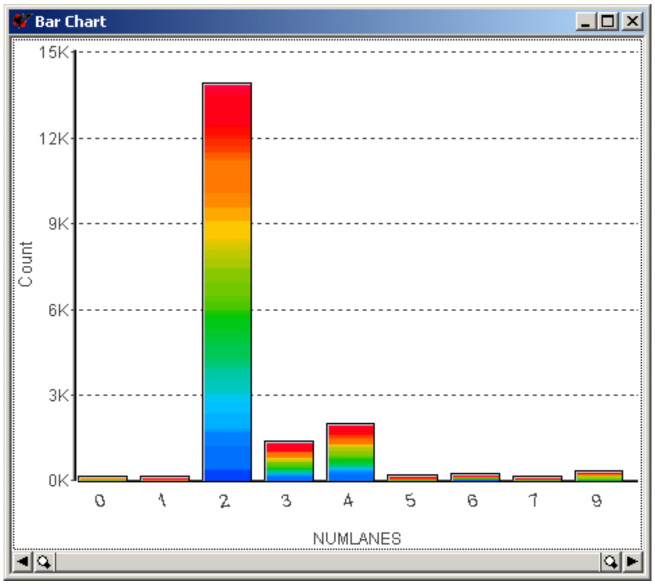


Figure 6: Bar chart showing distribution accident counts as a function of the number of lanes, from **FARS-A**. Colors (here translated to gray levels) correspond to states. As discussed in the text, at least two (7 = "≥ 7" and 9 = "Missing") and possibly three (0) of the seemingly numerical values are actually categorical.

12/7/97 and death time 19:27. This clear case of a mis-recorded month could not readily have been detected otherwise.

Second, one can visualize distributions of data values for each attribute. Figure 6 shows the distribution of the number of lanes on the road on which accidents occurred, taken from **FARS-A**. The relatively large number of zeroes ("not applicable") might lead to further investigation whether, for instance, "0" and "9" were both used to denote missing values. The category "7" in Figure 6 actually represents "seven or more," which cannot be determined without close reading of the documentation. An analysis that ignored this would be incorrect. The problem could readily have been avoided simply by using "7+" as the attribute value, which also prevents the attribute from being treated as a purely numerical quantity.

Third, for multi-year data, one can examine the longitudinal behavior of individual or aggregated values, which we do repeatedly in §4.3.

Multiple Attribute Analyses. Extending the notion of examining distributions of individual attributes, one can examine bivariate relationships. Looking at every possible pair of attributes may not be feasible or meaningful, but two special cases are relevant. Nor need the relationships be "systematic;" for example, (see Figure 8) variation in one attribute (Example: numbers of lanes associated with accidents by state) may reflect an "underlying" variation (Example: states' differing road inventories).

The first case is pairs of attributes for which there is a "physical" basis to expect a positive or negative relationship. An example is driver height and weight in **FARS-D**, which are shown in Figure 7, where we see the expected positive relationship. Moreover, outlying and boundary cases are readily visible and can be investigated further.

The second case concerns relationships based on domain knowledge. For example, data can be checked for "plausible" and "implausible" correlations. As examples, we present in Figures 8 and 9 visualizations

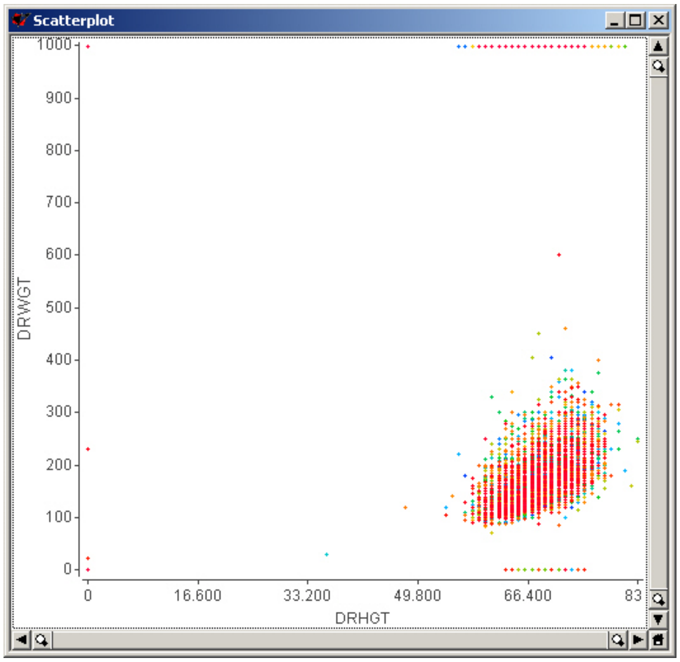


Figure 7: Scatterplot of driver height and weight, from **FARS-D**. As expected, these attributes are positively correlated. Missing weights, denoted by 999, are visually apparent, but would not be rejected or ignored by many statistical packages.

for pairs of attributes in **FARS-A**. Both plausible (for example, between state and number of lanes) and implausible (for example, between state and day of week) pairwise relationships are easily assessed from the visualizations.

Relational Database Issues. Checking RBDMS properties of the database is essential to appraising its quality. Principles of RBDMS design (cf. §3.1) provide a framework to ensure consistency and to identify or even resolve many data anomalies. The principal steps are:

Primary Keys: Check if the attributes that are stated to be or logically ought to be primary keys for a given file actually are. For example, in **FARS-A**, check that State and Case No. uniquely determine an accident record.

Normalization: Check if the database is properly normalized. If not, check that the non-normalized aspects do not harbor inconsistencies.

Foreign Keys: Verify that there are foreign keys or other attributes that allow files to be joined. For example, in FARS, verify that an accident record can be linked with the driver, vehicle, and person records involved in that accident.

Joins: Perform joins of tables to verify that records can be *correctly* linked.

To illustrate, attempts to join the Accident, Driver, Vehicle and Person files in the ITDB version of the FARS for those years with significantly less data (as shown in Fig. 4) reveals very serious problems,

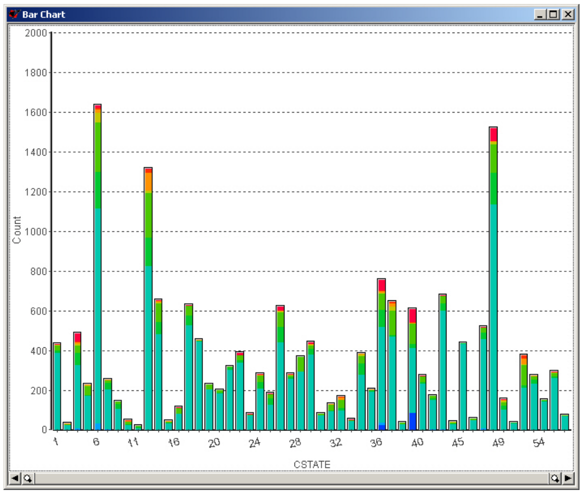


Figure 8: Bar chart of accident counts showing existence of a relationship between state and number of lanes. Colors (gray levels) correspond to the number of lanes. (This bar chart is the “transpose” of that in Figure 6.) This relationship is plausible since the distribution of the number of lanes varies by state.

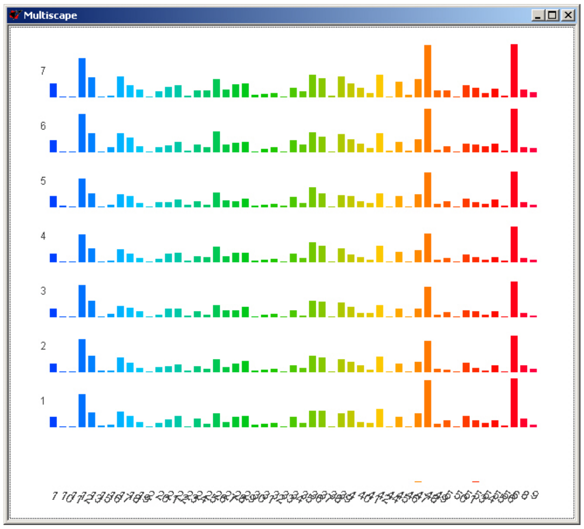


Figure 9: Multiscape [8, 38] of accident counts by state (horizontal axis) and day of week (vertical axis), showing no relationship.

State Code	Case Number	Vehicle Number in **FARS-D**	Vehicle Number in **FARS-V**	Vehicle Number in **FARS-P**
1	85	2,4,6,7	1,3,5	1,3,5,6,7
1	91	1	2	0,2
1	92	2	1,3	1,3
6	1345	2,3	1	1,2
6	1349	2,3	1	1,1,2,3,3

Table 2: Excerpt from Joins in FARS 1999 downloaded from the ITDB Web site. The vehicle numbers are those present in each of the files. The driver and vehicle files have *no records in common*.

which explain the anomalies discussed under **Metadata Characteristics.** Specifically, for a given accident we would expect a 1-to-1 correspondence between vehicle records and driver records. However, for 1999 FARS data downloaded from the ITDB, there is *no overlap* between the driver and vehicle files! In fact,

$$\begin{aligned} \#\{\mathbf{FARS\text{-}V}\} &= 38{,}268 \qquad \text{Key} = (\mathsf{CSTATE, CNUM, VNUM}) \\ \#\{\mathbf{FARS\text{-}D}\} &= 18{,}403 \qquad \text{Key} = (\mathsf{CSTATE, CNUM, VNUM}) \\ \#\{\mathbf{FARS\text{-}D} \bowtie \mathbf{FARS\text{-}V}\} &= 0 \end{aligned}$$

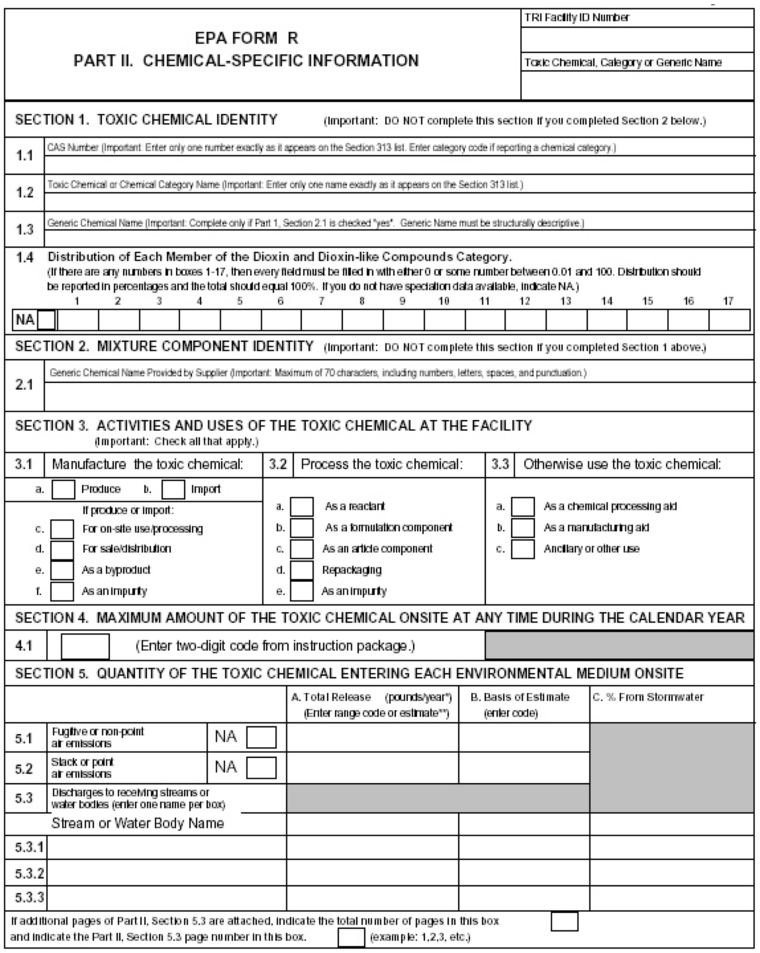
where $\{A\}$ denotes the number of records in database A and $A \bowtie B$ represents the linkage of records in databases A and B. Table 2 shows details of a few of the records. This diagnoses but does not explain the problem in Figures 4 and 5: no accident record appears in both the driver and vehicle files.

4.3 Case Study 2: The TRI

Background. The TRI database [10] was established by Federal law in 1986 to compel industrial and government facilities to disclose releases of any of more than 300 registered toxic chemicals into the air and water. Reports must be made for any facility that manufactures or processes more than 25,000 pounds (annually) of a registered chemical. (Starting 2001, the threshold for lead was reduced to 100 pounds, leading to 9800 additional facilities being required to report.) The TRI is used by the public and government agencies for purposes ranging from clean-up planning to lawsuits involving environmental justice.

Two characteristics of the TRI dominate assessment and amelioration of DQ problems. First, the TRI was created without identifying any specific uses of the data, which militates against user-centric approaches. Second, TRI data are self-reported, using a complex, five-page form (see Figure 10) that must be completed for *every facility and chemical* for which releases must be reported. There is no strong incentive even to complete the report, let alone do so accurately, which leads to a variety of data anomalies, most of which are not readily handled by standard statistical methods. In many cases, the form appears to have been completed by an engineer, often with substantial substitution of engineering judgement for “hard numbers.”

Rather than repeat all steps of the strategy outlined in §4.2, we focus here on detection and characterization of anomalies specific to one plausible use of the TRI data—to estimate regional-level trends. Both detection and characterization are addressed in the context of data viewed as facility-level time series. The data employed were downloaded using the TRI Explorer [10] and cover air, water, on-site land, and off-site land releases of lead in “Original Industries” (SIC 20xx–39xx) for the years 1988–1998. Since the TRI



EPA FORM R

PART II. CHEMICAL-SPECIFIC INFORMATION

TRI Facility ID Number

Toxic Chemical, Category or Generic Name

SECTION 1. TOXIC CHEMICAL IDENTITY (Important: DO NOT complete this section if you completed Section 2 below.)

1.1 CAS Number (Important: Enter only one number exactly as it appears on the Section 313 list. Enter category code if reporting a chemical category.)

1.2 Toxic Chemical or Chemical Category Name (Important: Enter only one name exactly as it appears on the Section 313 list.)

1.3 Generic Chemical Name (Important: Complete only if Part 1, Section 2.1 is checked "yes". Generic Name must be structurally descriptive.)

1.4 Distribution of Each Member of the Dioxin and Dioxin-like Compounds Category.
(If there are any numbers in boxes 1-17, then every field must be filled in with either 0 or some number between 0.01 and 100. Distribution should be reported in percentages and the total should equal 100%. If you do not have speciation data available, indicate NA.)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NA □																	

SECTION 2. MIXTURE COMPONENT IDENTITY (Important: DO NOT complete this section if you completed Section 1 above.)

2.1 Generic Chemical Name Provided by Supplier (Important: Maximum of 70 characters, including numbers, letters, spaces, and punctuation.)

SECTION 3. ACTIVITIES AND USES OF THE TOXIC CHEMICAL AT THE FACILITY
(Important: Check all that apply.)

3.1 Manufacture the toxic chemical:	3.2 Process the toxic chemical:	3.3 Otherwise use the toxic chemical:
a. □ Produce b. □ Import		
If produce or import:	a. □ As a reactant	a. □ As a chemical processing aid
c. □ For on-site use/processing	b. □ As a formulation component	b. □ As a manufacturing aid
d. □ For sale/distribution	c. □ As an article component	c. □ Ancillary or other use
e. □ As a byproduct	d. □ Repackaging	
f. □ As an impurity	e. □ As an impurity	

SECTION 4. MAXIMUM AMOUNT OF THE TOXIC CHEMICAL ONSITE AT ANY TIME DURING THE CALENDAR YEAR

4.1 □ (Enter two-digit code from instruction package.)

SECTION 5. QUANTITY OF THE TOXIC CHEMICAL ENTERING EACH ENVIRONMENTAL MEDIUM ONSITE

			A. Total Release (pounds/year*) (Enter range code or estimate**)	B. Basis of Estimate (enter code)	C. % From Stormwater
5.1	Fugitive or non-point air emissions	NA □			▨
5.2	Stack or point air emissions	NA □			▨
5.3	Discharges to receiving streams or water bodies (enter one name per box)		▨	▨	▨
	Stream or Water Body Name				
5.3.1					
5.3.2					
5.3.3					

If additional pages of Part II, Section 5.3 are attached, indicate the total number of pages in this box □
and indicate the Part II, Section 5.3 page number in this box. □ (example: 1,2,3, etc.)

Figure 10: Page 2 of Form R, on which TRI data are reported.

Explorer contains only separate files for each year, time series analyses require addition of Year attributes to each file, followed by merging of the annual files, a non-trivial impediment for some users.

Data Anomalies. The most striking characteristic of TRI data is extreme variation and gaps in the facility-level time series. Figure 11 shows several examples, all of which are both incomplete and exhibit dramatic year-to-year variability.

There is clear (but sometimes misleading) evidence of systematic constancy in the TRI data. Other than one aberrant year (a different completer of the form?), the facility at the top of Figure 12 seems to have constant, albeit suspiciously rounded releases. In fact, the values of “500” are categorical responses (corresponding to the range 100–999) to Question 5.1 on Form R (Figure 10) that appear in the TRI database as numerical values! Beyond this, multiple measurement methods may be employed by the form-completer, and while the method used must be entered on Form R, it is not contained in the downloadable data.

There is also evidence of “systematic change:” the facility at the bottom of that figure (in addition to the isolated 444,000 pound land release in 1990) has examples for which (with R_t the release in year t),

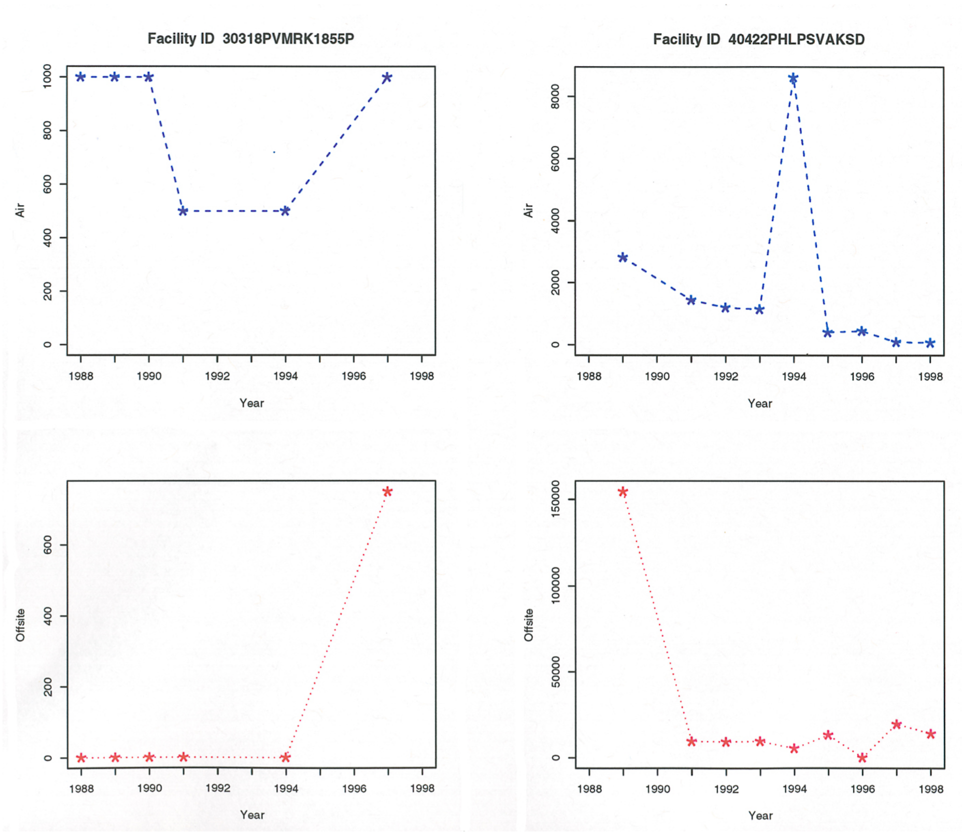


Figure 11: TRI Data showing annual air (top) and off-site releases of lead for two representative facilities.

$R_t = (2/3) * R_{t-1}$ and $R_t = (1/2) * R_{t-1}$. In these cases, the completer of Form R seems to have adjusted the previous year's values to approximate current releases.

Another human-engendered anomaly is disparities between on-site and off-site releases. Figure 13 illustrates this for several facilities; the same behavior is also present in Figure 12.

These anomalies, whose detection again requires use of visualization methods, have hard-to-model consequences. Most methods for time series analysis, for instance, are not designed to handle the kinds of error structures encountered above.

Regional-Level Trend. EPA administrators and others are concerned with pollution trends, at local and more aggregated geographical scales. The capability of TRI to provide useful information about regional trends, therefore, is one measure of its quality. Even though facility-level latitude and longitude are available from the TRI Explorer, without a GIS, true geographical aggregation is difficult. A simple approach aggregates facilities on the basis of the first digit of their Zip codes (the first 5 characters of the TRIFID in Figure 12), in effect dividing the country into ten regions and pooling releases within region.

This EDA-like strategy again illuminates the central issue. In many cases, the regional sum is effectively the maximum over facilities in the region. That is, regions are dominated by massive facilities. Therefore, facility-level anomalies are transferred to the regional level, and there is no error-cancellation from aggrega-

TRIFID	Year	Air	SurfH$_2$O	Land	OnSite	OffSite	Total
07029KRYNS936HA	1989	500	0	0	500	0	500
07029KRYNS936HA	1990	500	0	0	500	0	500
07029KRYNS936HA	1991	156	0	0	156	0	156
07029KRYNS936HA	1992	500	0	0	500	0	500
07029KRYNS936HA	1993	500	0	0	500	0	500
07029KRYNS936HA	1994	500	0	0	500	0	500
07029KRYNS936HA	1995	500	0	0	500	0	500
07029KRYNS936HA	1996	500	0	0	500	0	500
07029KRYNS936HA	1997	500	0	0	500	0	500
07029KRYNS936HA	1998	500	0	0	500	0	500

TRIFID	Year	Air	SurfH$_2$O	Land	OnSite	OffSite	Total
38109RFNDM257WE	1988	15300	0	0	15300	22163	237643
38109RFNDM257WE	1989	15300	0	0	15300	148151	164451
38109RFNDM257WE	1990	10200	0	440000	450200	19444	469644
38109RFNDM257WE	1991	3800	0	0	3800	76850	80650
38109RFNDM257WE	1992	1900	0	0	1900	313750	315650

Figure 12: Constancy and systematic changes in TRI data. *Top:* A facility with one seemingly anomalous year, which may be misleading because values of 500 actually correspond to the range 10–999. *Bottom:* A facility on which year t air releases appeared to be derived by judgement (Examples: two-thirds and one-half as much) from year $t - 1$ releases.

tion. Figure 14 illustrates this point for the regions defined by Zip codes 6**** (Midwest) and 7**** (south Central).

The TRI exercise shows that simple strategies—visualization and manual examination guided by domain knowledge—can work on a small scale. It also highlights the need to detect and characterize anomalies in data, raising in turn the need to characterize the impact of anomalies on inference from the data. Many of the same questions raised by the analysis of the ITDB arise again: What strategies scale to large problems? What strategies can be automated? These kinds of issues inform the research challenges described in the following section.

5 Research Challenges

Here we present statistical research challenges associated with DQ, grouped into theory and methodology, including modeling (§5.1) and the software tools (§5.2) that are necessary for the theory and methodology to address real DQ problems. Framing these challenges is the need to *integrate generic and problem-specific* components into the DQsystem. The case studies in §4 confirm that domain knowledge is an essential component of DQ. This is necessary at both the theory and methodology and software tools levels. How to do it, however, is not at all clear.

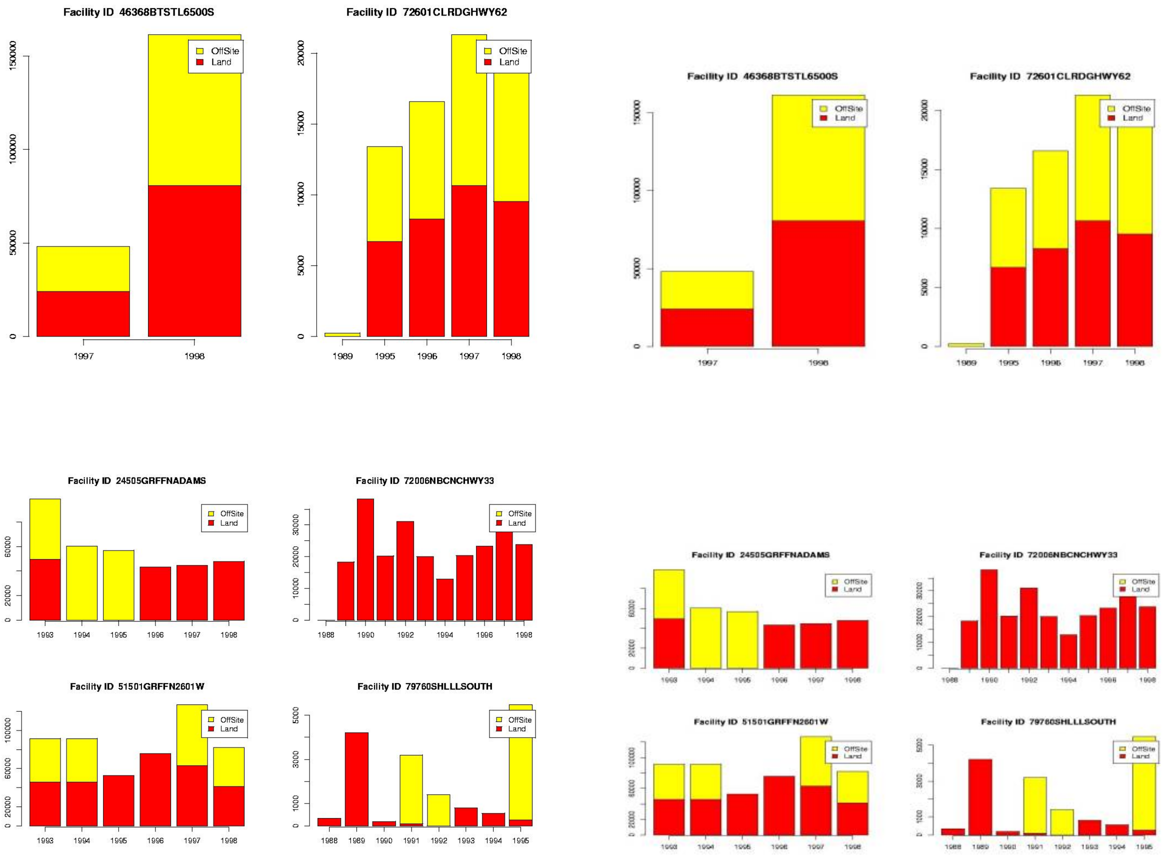


Figure 13: Stacked bar chart showing on-site and off-site disparities for selected facilities. Questionable aspects include Off-Site ≡ On–Site, Off-Site oscillating between 0% and 100%.

5.1 Statistical Theory and Methodology

DQ Metrics are necessary for both summary and decision purposes. A participant in the NISS Affiliates Technology Day on DQ [30] posed the question “Is there a science of data quality?” If there is, as noted in §2, it is rooted in quantification—metrics for DQ.

Here are two initial examples. First, the *Intactness* of a table is the fraction of records that pass completeness, consistency and plausibility checks of the sort described in §4. In general, low intactness is symptomatic of low DQ. Accounting only for missing values other than latitude/longitude and inconsistent times (Example: EMS arrives before the time of the accident), the intactness of the 1999 **FARS-A** file is 0.3. Were the latitude/longitude included, intactness would be less than one-half of one percent.

Second, given a table with D attributes, the *Dimensional Efficiency* D^* of D may be defined in both a simple way, as the number of attributes that are neither constant across all records (and hence contain no discriminatory or predictive information) nor missing to such an extent that they are useless nor derived from

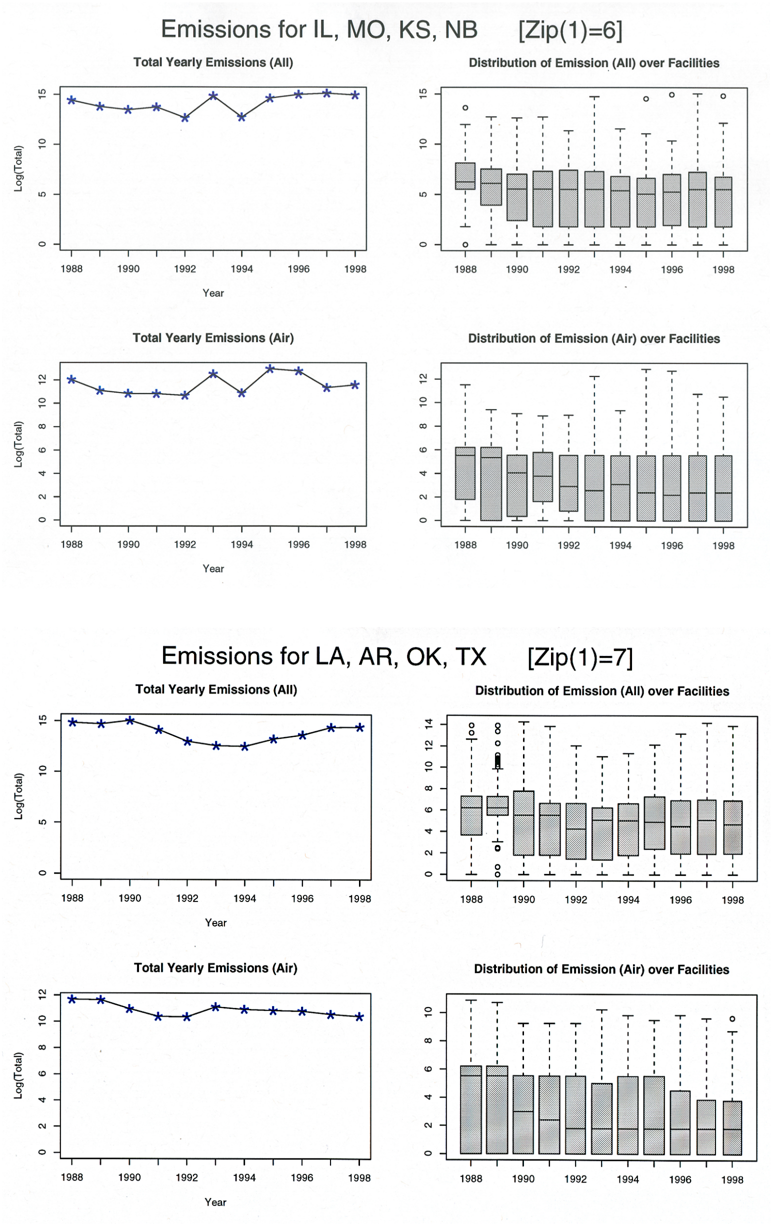


Figure 14: Facility-level distribution (Box plots) and regional totals (lines for Zip codes 6xxxxx (Top) and 7xxxx (Bottom).

other attributes, and in a complex way, as the number of dimensions needed to explain a fixed proportion of the variability in principal components analysis. The latter provides sharper information about the amount of independent information among the attributes.

Other, yet-to-be-developed metrics must accommodate such DQ characteristics as accuracy, credibility, accessibility, relevance, timeliness, and interpretability. These properties can all apply at multiple scales, i.e., individual records, databases and integrated databases; they can also reflect multiple uses of data.

Quality Resistant Inference. Despite attractions of the TQM approach to DQ, it will always be necessary to use poor quality data. In light of this, inference procedures are needed that mitigate poor DQ.

One strategy is to use robust estimators. Some procedures (Example: S-estimators [31, 32]), can perform well for single parameters even when just more than half of the data are correct, giving robust estimates of central tendency, dispersion, and association. (Of course, one would want to know first that the data are this bad.) These techniques perform less well, of course, for multivariate parameters such as covariance matrices: breakdown points depend on the dimension, and most methods cannot accommodate a very large percentage of poor-quality data.

Robust methods can also assist in detecting “outliers” and other data oddities, but as the TRI case study shows, real-world data anomalies correspond to error structures that even robust models seem unable to accommodate, so there is much room for innovation.

One specific research issue is that there is no robust procedure for estimating a proportion from survey data, a ubiquitous issue in federal statistics. The impact would be enormous: estimates of proportions arguably play a larger role in framing national policy than estimates of location or dispersion.

Crosswalks. Data collection processes are not static, and this introduces a special kind of DQ problem. For example, the Census Bureau recently changed the way in which race is reported: for many decades, people could report membership in only a single race, but in 2000 they were permitted to report any combination of six racial categories. Such changes are made for good reasons and are likely to improve DQ in the long run, but the immediate impact is to make it difficult to characterize trends. Crosswalks are meant to address such problems. A crosswalk is a plan to collect data by both the new and old methods for a period of time, so that the old time series and the new time series can be calibrated [2].

There are limits as to how successful a crosswalk can be. Suppose that at time c the process changes, and that the parallel collection for the crosswalk lasts for m time periods after the change. Thus the original data collection system captures $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_c, \mathbf{X}_{c+1}, \ldots, \mathbf{X}_{c+m}$, while the new collection system captures $\mathbf{Y}_c, \mathbf{Y}_{c+1}, \ldots$. Suppose that the original data collections were used to produce summary estimates (Example: the proportion of people whose primary racial identification falls into each of six categories). Denote that series of estimates by $\hat{\theta}_1, \ldots, \hat{\theta}_c$. Then the goal of the crosswalk is to find a function $f(\cdot)$ that operates upon the $\mathbf{Y}_t$ to produce estimates $\hat{\theta}_t$, thereby allowing the analyst to provide comparable estimates for time period $c+1$ and beyond.

The ideal statistical strategy is to ensure that the m is large, so that there is extensive overlap in the crosswalk. That would enable analysts to use nonparametric models such as MARS [15] to find the function that best extends the series. However, this ideal is almost never achieved in practice: typically, because of financial considerations, $m \leq 2$. Moreover, only rarely are both systems run at a full level.

As a result, in practice many organizations apply a simple proportional correction, assuming that correction factor is stable across future survey conditions. For cases such as race in the Census, this assumption is suspect: people’s attitudes about race are changing at the same time that the racial composition of the US is changing.

New research ideas are needed to attempt to resolve such multiple effects. Almost inevitably, the models will be Bayesian, in order to incorporate both prior information and hierarchical structure.

Predictive models for DQ that reflect the nature of the processes by which data are generated represent an immense opportunity for statisticians. The needs encompass modeling the role of people in data production processes (Example: Form R for TRI), new kinds of errors (Example: transposition of digits by a data entry clerk, or the incorrect linkage of records) with unusual dependences (Example: TRI time series), and novel concepts of "noise."

To illustrate, to the extent that DQ is a function of the quality of individual elements of a database, not all elements are equally likely to have poor quality. (Example: the accuracy of a particular item in a survey depends upon such factors as the complexity of the question, the sensitivity or delicacy of the information, the burden of the survey, and the demographics of the respondent.) One path would be to build a logistic regression model that predicts the probability that a data item is correct as a function of suitably quantified explanatory variables. These variables are context-specific, but such abstractions as complexity, sensitivity, and burden are significantly generic. Ultimately, this kind of approach will help in targeting quality improvement efforts.

Models for changes in data quality under transformations (Examples: aggregation, joins of relational tables within the same database, and integration of multiple databases) of data are also essential.

A Decision–Theoretic Formulation of DQ is a daunting research challenge for theory and methodology. For example, when decisions depend sensitively on factors other than data issues, it may not be worthwhile to improve DQ, but it might nevertheless be essential to characterize DQ. In other cases, where databases drive decisions that have enormous consequences, one could, with the right models, make a strong case for significant investment of resources to improve DQ.

The entry point is *quantification and models for costs and benefits* of DQ. The issues are complex: multiple costs (Examples: costs incurred by data generators, data maintainers, data users and data subjects), elusive costs (Examples: how much does it cost one to receive duplicate mailings? How much would it cost EPA to remove anomalies from the TRI? How much does it cost anyone if a person is wrongly suspected of terrorism on the basis of low quality data?) and elusive benefits (Examples: how much would a telephone operating company gain if it could link its cellular and landline customers accurately? How much would the public benefit if TRI were of higher quality?). Given the complexity of the issues, a meaningful starting point is to conduct cost-benefit case studies similar to those in §4.

Similarly, and linking to cost considerations, regression models could be used to predict the cost of raising DQ to a given level, at either the item or the database level. Such models, tailored to specific applications, can also help prioritize resource allocation for quality improvement.

5.2 Software Tools

The NISS DQ Workshop [24] identified as a central research need "Software tools that solve real data quality problems. Such tools (even in prototype form) can also be used to evaluate and improve new theory and methodology. Issues include algorithms that cope with complexity of the techniques as well as the scale of the data and problems of human–computer interaction such as presentation and visualization."

Such tools exist already, as discussed in § 3.1, but are oriented more to the computer science view of DQ than the statistical perspective. To make the challenges concrete, we frame them in terms of a data quality toolkit—an extensible set of software tools that automates the generic components of the strategy described in §4, performing both automatic and on-demand computations, including visualizations, but contains "hooks" for relevant domain knowledge. The DQTK has not yet been built, although NISS has worked to develop many of its components.

There are precedents: the ARIES software system [11] developed at the Bureau of Labor Statistics (BLS) is a graphical tool for examining DQ that provides means to visualize anomalous data. It incorporates a portion of the functionality of the DQTK in the setting of the Current Employment Statistics survey data.

Functionality. At a high level, a DQ evaluation using the DQTK would consist three phases. Importantly, the DQTK creates a complete log of the process that contains user-entered metadata and results.

The first phase is user metadata entry and characterization. For the database, this includes names of tables, file names, and formats. For tables, it includes dimensions (records × attributes), and for each attribute, such information as name, unit of measurement, data type (Examples: numerical, text, date, time, latitude/longitude, Boolean, "other" with parsing rule), constraints (Examples: "must be positive," "must be an integer"), whether the attribute is original or derived, and representation of missing and N/A values.

An essential functionality is to help users understand metadata such as survey response rates and respondent burden (§2). One very intriguing way to do this is to use data collection instruments such as the TRI's Form R (Figure 10) to visualize such metadata, for example, by coloring items according to the response rate. This makes not only the collection instrument but also key metadata about the data generation step immediately accessible to DQTK users.

In the second phase, DQ evaluation uses automatic computations at two stages: following metadata entry and at the end of the process to calculate DQ metrics. At the first stage, the files are read into the RBDMS. Then, for each attribute, the DQTK computes the percentages of missing and N/A values, and creates one key visualization, based on attribute type (Examples: histogram, bar chart, map). Consistency checks are then performed, first for user-selected individual attributes. For instance, the DQTK checks for "illegal values" relative to the attribute type (Example: time = 2585) or relative to bounds specified by user on the basis of domain knowledge (Example: weight $\geq$ 500 pounds).

In the second stage of the second phase, the DQTK checks inter-attribute consistency (Examples: temporal consistency, mathematical relationships such as $A_1 \geq A_2 + A_{15}$, and logical relationships such as A_1 = (Latitude, Longitude) belongs to A_{25} = State). Derived attributes are checked as well, with the user specifying the "formula" to compute a derived attribute from others. The DQTK also checks missingness, to detect systematic features such as those in Figure 3.

In the third phase, the user follows through EDA DQ strategy, seeing the results of selected automatic computations and directing the system to perform additional computations. As this process occurs, the user is provided text and visual summaries of problems, with lists containing details available as one drills down. All results, including visualizations, are written to the log file.

RDB characteristics are treated similarly. For instance, for each table the user specifies the attribute meant to be its primary key, and the DQTK provides either a verification or a summary of problems (with details via drill-down). Joins are performed, with the user specifying which tables are to be joined, the joining attributes, and attribute filters, and the DQTK returns the cardinality of the join and other summary information with lists of problematic records as drill-down.

Finally, the DQTK computes automatically DQ metrics (§5.1) for each table in the database.

Structure. The main components, which are shown pictorially in Figure 15, are:

User interface (UI), with the look and feel a graphical SQL client, allowing access by means of a standard Web browser.

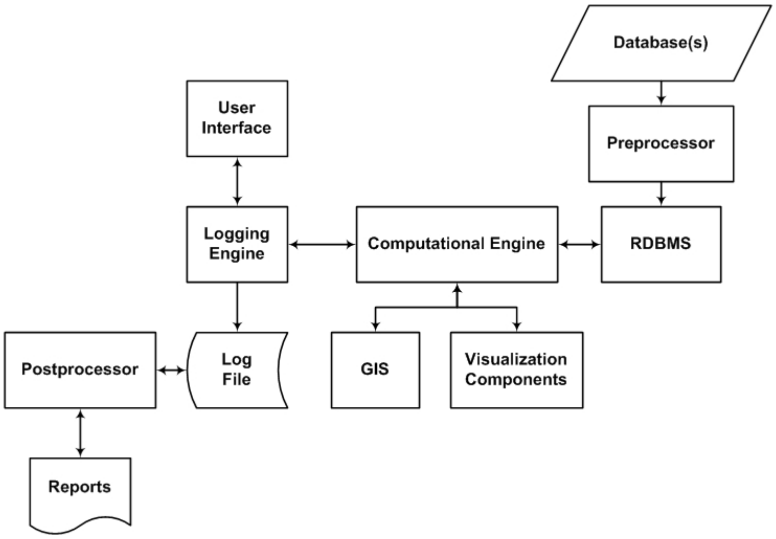


Figure 15: System architecture for the DQTK.

Logging engine to record the entire session.

Computational engine to perform necessary calculations, calling as necessary on (1) An RBDMS to access and manipulate the data, which are converted to relational form by a **preprocessor**; (2) A GIS for manipulating and displaying geographical data; and (3) Visualization components to provide graphical output to the user.

Postprocessor to generate reports from the logs.

Building even a prototype DQTK raises research issues that span the three contributing disciplines of DQ. Beyond those implicit in the discussion of its core functionality, these include selection of software and hardware platform, design of the user interface, and incorporation of additional statistical and data manipulation capabilities. In the long run, an expert system wrapper for the system, which would suggest analyses to the user on the basis of the results of other analyses, seems very desirable. Making the DQTK scalable to handle large data sets is a challenge in its own right.

5.3 DQ and Other Problems

The relationship of DQ to other problems provides a way to leverage research on DQ, and raises intriguing research challenges.

For example, software quality bears a strong resemblance to DQ. Both “products” are electronic rather than physical, so that quality is a characteristic of a class rather than instances. For both, quality is highly situational: in the same way that a database may be adequate for one purpose but not another, the same software may serve adequately in one setting, but not in another. Also for both, humans (data generators and software developers) are central to the production process as well as an essential source of variability. The protracted (and still incomplete) effort to produce metrics and standards for software quality [23] may provide insight into DQ metrics.

Data confidentiality, a long-standing problem for government agencies, is burgeoning in the world of E-commerce and has dramatic implications for homeland security. The relationship between data confidentiality (DC) and DQ is complementary: the same tools, such as those for record linkage, that increase DQ threaten DC. Conversely, tools that alter data but allow informative inference (Example: swapping some attributes between records [33]), or that maximize data utility subject to constraints on disclosure risk [28], suggest ways to characterize how much information can be extracted from low quality data.

6 Conclusions

Too many organizations run on poor quality. The appeal of TQM approaches notwithstanding, the error rates in most databases (with rare exceptions) are so great that it would be prohibitively expensive to attempt correction. Root cause analyses, although intriguing, seem problematic because of the diversity of causes, as illustrated in § 4.

Instead, organizations have developed informal coping mechanisms that limit the sensitivity of their key decisions to the influence of bad data. Business managers look to their databases for suggestive trends, but are skeptical of detailed conclusions. In government, policy is almost never framed solely on the basis of statistics, or even largely upon statistics, but rather arises from compromises among stakeholders, legal experts and pragmatists, all of whom are skeptical (perhaps for different reasons) about even their own experts' analyses.

The drawback is not so much that the present system is unworkable, but rather that we do not make the best use of our data. This is a heavy price to pay in a competitive world economy. Businesses suffer when there are errors in billing or service records, or when they fail to quickly discover emerging market trends. In the federal context, some have attributed failure to detect the September 2001 hijackers, tire problems with Ford Explorers, and the fen-phen drug interaction that caused heart damage (discovered by an alert nurse, not the Adverse Event Reporting System at the US Food and Drug Administration) to poor quality to our inability to use poor quality data effectively.

Statisticians, in collaboration with computer scientists and domain knowledge experts, have a major role to play in the process of using data effectively—which is ultimately what DQ is all about. That role ranges from characterizing DQ to methods of inference that are resistant to poor DQ to cost-benefit analyses that determine when specific investments in quality improvement pay off. We have attempted to lay the groundwork through a concrete path of case studies, our EDA approach to DQ metrics and measurement, and the DQTK. Ultimately, a decision-theoretic formulation of DQ will lead to tools that help managers to allocate resources to the most urgent, addressable problems.

Acknowledgements

Support for the case studies described in §4 was provided to NISS by the BTS and EPA. Many of the ideas here were shaped by the NISS Affiliates Workshop on DQ [24]. Jon Kettenring first brought DQ to the attention of NISS. Jerome Sacks participated in the TRI case study. The visualizations were produced using Visual Insights ADVIZOR [38], a Windows-based set of visualization components that support linked selection operations across multiple views.

References

[1] P. P. Biemer, R. M. Groves, L. E. Lyberg, N. A. Mathiowetz, and S. Sudman. *Measurement Errors in Surveys*. Wiley, 1991.

[2] D. Bradley and K. Earle. Developing and explaining the crosswalk between census 1990 and 2000 industry and occupation codes. In *Proceedings of the Joint Statistical Meetings*, Alexandria, VA, 2001. American Statistical Association.

[3] Bureau of Transportation Statistics. Intermodal Transportation Database. Available on-line at www.itdb.bts.gov.

[4] SAS Corporation. Dataflux, 2002. Information available on-line at www.dataflux.com.

[5] C. J. Date. *An Introduction to Database Systems, 7th Ed.* Addison-Wesley, Reading, MA, 1999.

[6] E. D. de Leeuw and J. van der Zouwen. Data quality in telephone and face to face surveys: A comparative meta-analysis. In *Telephone Survey Methodology*, pages 283–300, New York, 1988. Wiley.

[7] K. B. Duncan and E. A. Stasny. Using propensity scores to control coverage bias in telephone surveys. *Survey Methodology*, 27(2):121–130, 2001.

[8] S. G. Eick and A. F. Karr. Visual scalability. *J. Computational and Graphical Statist.*, 11(1):22–43, 2002.

[9] L. P. English. *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*. Wiley, New York, 1999.

[10] US Environmental Protection Agency. Toxic Release Inventory Database. Available on-line at www.epa.gov/triexplorer.

[11] R. Esposito, J. K. Fox, D. Lin, and K. Tidemann. Aries: A visual path in the investigation of statistical data. *Journal of Computational and Graphical Statistics*, 3:113–125, 1994.

[12] Federal Transit Administration. National Transit Database. Available on-line at www.fta.dot.gov/ntl/database.html.

[13] I. P. Fellegi and D. Holt. A systematic approach to automatic edit and imputation. *J. Amer. Statist. Assoc.*, 71:17–35, 1976.

[14] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *J. Amer. Statist. Assoc.*, 64:1183–1210, 1969.

[15] J. H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19:1–67, 1991.

[16] L. Galway and C. H. Hanks. *Data Quality Problems in Army Logistics: Classification, Examples & Solutions*. Rand Corporation, Santa Monica, CA, 1996.

[17] B. Greenberg and T. Petkunas. SPEER (structured Program for Economic Editing and Referrals). In *ASA Proceedings of the Section on Survey Research Methods*, pages 95–104, Alexandria, VA, 1990. American Statistical Association.

[18] J. R. Groff and P. N. Weinberg. *SQL: The Complete Reference*. Osborne/McGraw-Hill, Berkeley, CA, 1999.

[19] R. M. Groves. Research on survey data quality. *Public Opinion Quarterly*, 51:156–172, 1987.

[20] R. M. Groves, L. J. Magilavy, and N. A. Mathiowetz. The process of interviewer variability: Evidence from telephone surveys. In *ASA Proceedings of the Section on Survey Research Methods*, pages 438–443, Alexandria, VA, 1981. American Statistical Association.

[21] M. A. Hidiroglou, J. D. Drew, and G. B. Gray. A framework for measuring and reducing nonresponse in surveys. *Survey Methodology*, 19:81–94, 1993.

[22] K.-T. Huang, Y. W. Lee, and R. Y. Wang. *Quality Information and Knowledge Management*. Prentice Hall, Upper Saddle River, NJ, 1999.

[23] S. H. Kan. *Metrics and Models in Software Quality Engineering*. Addison–Wesley, Reading, MA., 1994.

[24] A. F. Karr, A. P. Sanil, J. Sacks, and E. Elmagarmid. Workshop report: Affiliates workshop on data quality. Technical Report, National Institute of Statistical Sciences. Available on-line at www.niss.org/affiliates/dqworkshop/report/dq-report.pdf, 2001.

[25] K. E. Kendall and J. E. Kendall. *Systems Analysis and Design, 5th Ed.* Prentice Hall, Upper Saddle River, NJ, 2001.

[26] D. Loshin. *Enterprise Knowledge Management*. Morgan Kaufmann, San Francisco, 2001.

[27] National Highway Traffic Safety Administration. Fatility Analysis Reporting System. Available on-line at www.nhtsa.dot.gov/people/ncsa/fars.html.

[28] National Institute of Statistical Sciences. Digital Government Project Web Site. Available on-line at www.niss.org/dg.

[29] H. B. Newcombe and J.M. Kennedy. Record Linkage: Making Maximum Use of the Discriminating Power of Identifying Information. *Communications of the ACM*, 5, 1962.

[30] National Institute of Statistical Sciences. Niss affiliates technology day on data quality, 2002. Information available on-line at www.niss.org/affiliates/techday200202.html.

[31] P. J. Rousseeuw and V. Yohai. Robust regression by means of s-estimators. In J. Franke, W. Hardle, and R. D. Martin, editors, *Robust and Nonlinear Time Series*, volume 26 of *Lecture notes in Statistics*, pages 256–272. Springer-Verlag, New York, 1984.

[32] S. Sakata and H. White. Breakdown point. In S. Kotz, C. Read, and D. Banks, editors, *Encyclopedia of Statistical Sciences*, volume Update Volume 2, pages 84–89. Wiley, New York, 1998.

[33] A. P. Sanil, S. Gomatam, and A. F. Karr. NISSWebSwap: A Web Service for data swapping. *J. Statist. Software*, 2002. Submitted for publication.

[34] Evoke Software. Evoke, 2002. Information available on-line at www.evokesoft.com.

[35] Trillium Software. Trillium Data Quality, 2002. Information available on-line at www.trilliumsoft.com.

[36] C. Tucker. The estimation of instrument effects on data quality in the Consumer Expenditure Diary Survey. *J. Official Statist.*, 8:41–61, 1992.

[37] J. Tukey. *Exploratory Data Analysis*. Addison–Wesley, Reading, MA, 1977.

[38] Visual Insights, Inc. Visual Insights ADVIZOR. Information available on-line at www.visualinsights.com/advizor.

[39] R. Wang. A product perspective on total data quality management. *Comm. ACM*, 41(2), 1998.

[40] R. Y. Wang, M. Ziad, and Y. W. Lee. *Data Quality*. Kluwer, Amsterdam, 2000.

[41] W. E. Winkler. Advanced Methods for Record Linkage. *ASA Proceedings of the Section on Survey Research Methods*, pages 467–72, 1994.

[42] W. E. Winkler. Editing discrete data. In *ASA Proceedings of the Section on Survey Research Methods*, pages 108–113, Alexandria, VA, 1995. American Statistical Association.

[43] W. E. Winkler. Machine learning, information retrieval, and record linkage. In *ASA Proceedings of the Section on Survey Research Methods*, pages 20–29, Alexandria, VA, 2000. American Statistical Association.

[44] W. E. Winkler. Machine learning, information retrieval and record linkage. Available on-line at www.niss.org/affiliates/dqworkshop/papers.html, 2000.

[45] W. E. Winkler and B.-C. Chen. Extending the Fellegi–Holt model of statistical data editing. In *ASA Proceedings of the Section on Survey Research Methods*. American Statistical Association, 2001.