

Data Quality With Base SAS

Clarence Wm. Jackson, CSQA

Abstract

Attaining and maintaining Data Quality should be a hot item on everyone's to-do list. Data is an asset to an organization. The need for accurate and correct data is apparent to anyone that use the data for decision making, forecasting, and other uses that depend on the need for accurate information. The impact of bad data to organizations can be great when decisions made based on the data results in lost business, unhappy customers, lawsuits, and various levels of organizational failures.

This paper addresses how SAS can help an organization to

1. Understand the level of correctness and accuracy of their data thru data audits based on the organization's business rules
2. Correct the bad data
3. Keep the data clean

Introduction

The need for accurate data is obvious but the attention to ensuring data is accurate only comes up when building data warehouses, migrating data across data stores or to new systems, mergers that result and shared data, or when problems are noted either thru customer complaints, law suits, or other problems that make it an issue. Data should be the first thing that should be made accurate.

Remember the old saying "GIGO" or, "Garbage in, garbage out"? Well, relying on bad data going into a system or application will give you bad results coming out. This results in increased costs to find and fix errors, and the lost confidence in the reliability of the system or application.

The only problem with ensuring data quality is the cost and time associated with the activity, and providing ROI for the funding of the project. Although any manager or CEO will admit that inaccurate data is bad for business, they will not want to spend money to find bad data unless it impacts the bottom line. Companies have lived with bad data, and unless it can be shown that the problem is severe, a business case audit data will be hard to quantify as savings to the company.

Accurate Data Does Matter

I began my journey using SAS version 79 at the City of Dallas' Revenue & Taxation Department, Systems & Data Management Division (SDM). SDM was responsible for managing the data for property tax, financial, budget, and special collections systems. We had a great need to keep accurate data for all of these systems, and in being able to verify the correctness of the data. As a point of reference and examples, I'll describe how we did with SAS.

The need for accurate data in these systems was critical to the operations of the city. Bad data in any of these systems could result in the City not being able to function, and would cause various levels of disruptions in services.

Bad data was an unacceptable risk for the city, so making sure the data was OK was a high priority. Decisions by the City's leadership were made based on the numbers, i.e., data, from those systems, so accuracy was very important. Every decision must have accurate information (data) to make the best choice for the organization.

At one time, all data was stored and processed on the mainframe. Then came the personal computers and its applications like word processors and spreadsheets. As with other organizations during that time, the City put PCs in the hands of the decision makers, and they began transferring data from the mainframe reports to the PC applications. It soon became apparent that accurate data could become inaccurate because of typos, misunderstanding the data, and other reasons. This caused folks to start making it easier for the transferring of data thru electronic means to reduce errors.

But if the base data is bad, this will give rise to the rapid dissemination of bad data throughout the organization. With data being electronically passed and shared between various systems and environments, being summarized, transformed, and presented to the decision makers in spreadsheets,

dashboards, and scorecards, the chance that bad data can contaminate other systems is greater than at any time in IT. The risks of the data doing harm to the organization are greater also, so the need for good data should be very important. The data has to be accurate in all of its forms, or it is useless to the organization.

The risk and cost of bad data isn't realized until it is too late. The cost to clean data may be high, but the cost of using bad data can be higher. However, it is in the interest of organizations to make sure that their data is OK.

How accurate is the data in your organization? How do you know? How do you find and correct the data, and keep it clean? There are four basic steps to answer these questions:

1. Define the business rules
2. Audit or inventory the data
3. Clean, fix, or repair the bad data
4. Keep the data clean
 - a. Identify sources of bad data entry into the system
 - b. Correct the source of bad data entry
 - c. Enforce the business rules

Define the Business Rules

"Business rules" should be defined for all data elements, and all users and maintainers of the data should apply the rules when dealing with the data. These rules define what is 'good' data, and must be done in coordination with those that have a

stake in the data. Defining what is good data can be difficult if there aren't clearly defined business rules. These rules define what the valid values of each data element are, and what conditions the data should have a value.

Business rules should be enforced throughout the organization so that accurate data can be maintained. The owners and users of the data define business rules. Bad data is usually the result of bad rules and/or poor enforcement of the rules. If the business rules are bad and/or poorly enforced, data will never be reliable and will cost the organization.

Business rules help to set standards and establish baselines of accuracy and correctness for data. Clearly defined rules will leave no doubt about what the data should look like.

For instance, the four-character length data element 'LOT' should contain the geographic lot description of a parcel of real estate. An audit of the data shows that the value for "lot A" was in the following formats:

- ☞ 'LOTA'
- ☞ 'A '
- ☞ ' A'

Which format is correct and accurate? What is the rule? Can all three values be correct and accurate? Will a human and a computer read these the same?

A good business rule will limit variability and require that only one

format would be correct and accurate for use in the organization. Without clear business rules, the value of the data is suspect, and the integrity of the data is in question.

Audit and Inventory The Data

This is where SAS will provide the most bang for the buck, and save a few bucks in the process for the organization. An audit of data can be time consuming and require a fair amount of resources. It is best to define what data elements will be audited and inventoried based on the risks to the organization, the priority of the data being accurate, and the cost/benefit of correcting the data.

One method that can be deployed quickly to reduce the chance of bad data is to perform data quality audits. And a good tool to use in doing the audit is SAS' PROC FREQ. Why? PROC FREQ will find those combinations that don't follow business rules, and highlight the size of the problem to be fixed. And by programming the business rules to check for violations, SAS is the ideal tool for finding and fixing those data elements.

For example, let's say that business rule states that there should only be two values for 'status',

- ☞ 'F' for 'paid' and
- ☞ 'D' for 'due',

According to the rule, only 'D' status records should have a balance not equal to zero. How could you verify that the data is accurate based on the business rules? Write a SAS program that tests the data based on

the business rules, and summarize it using PROC FREQ. For example:

```
data chekdata;
  input status $ 1.
        Balance 5.2;
  If Balance = 0
  then
    BAL='0 $ due';
  else
    BAL='$$ due';
cards;
F 0.00
P 1.00
D 1.00
F 1.00
;
RUN;
```

```
PROC FORMAT;
  VALUE $ST
    'F'=PAID'
    'D'='DUE'
    OTHER='BAD';
RUN;
```

```
PROC FREQ;
  FORMAT status $FMT.;
  TABLE status*BAL;
RUN;
```

```
Data bad_data;
  Set chekdata;
  if (BAL='0' and status NE 'F')
  then status='F';
  if (BAL='$' and status NE 'D')
  then status='D';
Run;
```

This quick program should result in 2nd and 4th observations in step "A" being flagged as incorrect, and should be presented for correction to comply with the organization's business rules. The report from PROC FREQ step "B" will present a nice summary of the actual data, and provide understanding regarding the

level of correctness of the data elements being reviewed.

At this point, note that the requirement of knowing and auditing against a business rule is so important to the correctness of the data. What are the rules about the "F status" record with the balance? Is the status or balance correct? What is the business rule for this condition? Clear business rules define for the organization how the data should be treated from data entry to data warehouse to history archives.

Correct The Data

The previous SAS code example assumes that the rule regarding balances overrides the rule regarding the status code. The correction as applied in step "C" will rewrite the bad data for the "P status" record, and the "F status" record with the balance.

Keep The Data Clean

First, all points of data entry should be identified so that sources of bad data entry into the system can be reviewed. One thing about systems today is the ability to apply edits to data before it is added to the data stores. The rules should be so coded so that data doesn't enter the system, thereby correcting the source of bad data entry. Applying edits and constraints at data entry is one way of enforcing the business rules at the source.

All transformation processes should be reviewed to make sure that data

is not being changed in these processes. Audits at various points in these processes will uncover data transformation problems, and corrections should be applied in those processes.

Continued monitoring of the data will provide assurances that the data is staying clean. The same SAS jobs that were used to identify the bad data can be used to monitor the data stores, and should be run as needed.

A Case of Mistaken Identify

There are times when data needs to be centralized to form enterprise data stores, and data from two or more applications are required to be merged into a 'new' application. This was the case on a recent project to collect user info to be used in an accounting system. The data used was from "trusted" human resource and trouble tickets applications.

The thinking by management was that the data could be linked by names, but the reality was the data was so different, that the employee numbers had to be used, which resulted in a 95% match. But an interesting discussion began regarding how to apply business rules to the use of names.

One of the major areas of mismatched data was in the use of the rules on 'first' names. For instance, consider this small sampling of names and their common variants:

- ☞ Michael: Mike, Mickey, Mick
- ☞ Robert: Bob, Rob, Bobby, Robby

- ☞ William: Will, Willie, Bill, Billy
- ☞ Daniel: Dan
- ☞ Lawrence: Larry
- ☞ Patricia: Pat, Patty, Tricia
- ☞ Richard: Dick, Rich,
- ☞ David: Dave, Davy

As the QA manager, I was able to show the managers of the three systems what the impact of their decision would be, and also what the impact would be when the match criteria was changed to the employee number. This was done using Base SAS.

Reviewing the data with the managers, it looked like the data entry by one group was based on different business rules. One system required legal names, while the other allowed the operator to enter whatever first name the caller gave them. So those variations were there for the main file that would be used for usage accounting. I'm sure I gave the managers of all three systems something to think about in regards to the data:

- ☞ For the human resource system: How accurate was their data for those 5% of non-matches?
- ☞ For the trouble tickets app: How to clean the data correctly (duplicate customer records, use of nick-names, and other variations), and by what business rules?
- ☞ For the new applications: What rules should be used for the names, and which system should supply the name data?

Because of the lack of consistent rules on either data store for the two

systems, the name match would not work for loading the data.

This is information about the data matching by name report provided to the managers. But it only took an afternoon (mostly getting the data from the owning departments and two meetings with the managers) to do the analysis and review using Base SAS DATA and PROC steps. SAS was used to do the audit of each data store, create review reports for the managers, and create a 'clean' file for input into the new application.

The discovery of the problems in the data presented to the managers of the three systems information about their systems' business rules, especially those related to names. I used first names here, but there were problems with last names, and other items. This did lead to discussions between the managers, regarding the different business rules they used for names. Talk of an enterprise cleanup effort started, then deadlocked over vendor concerns and cleanup cost. But the discussion did lead each group to adopt clear rules, and to apply processes to cleanup their data stores. In the mean time, they have decided that using the employee number for interfacing between their systems in the future would be the best way to go.

Conclusions

There are issues with bad data, and there are plenty of examples where bad data has cost organizations either in lost business, lost

productivity, and other losses. The risks of using bad data depend on the value that is associated with it based on the organization. Some organizations cannot afford any bad data, whereas some organizations feel that the risks are low. However, it is up to each organization to ensure that the data used by their decision makers is correct and accurate.

Applying basic steps to find, fix, and stop bad data from being a part of their organization can minimize the cost of correcting the data. Basic SAS data steps and procedures can be used to find and fix the data, and monitor the data stores thru the reports that are generated with this review.

And the SAS programs are easy to code, and shouldn't be difficult.

Author Contact info:

Clarence Wm. Jackson, CSQA
City of Dallas, Communication &
Information Services
Manager, Change Management &
Quality Assurance
1500 Marilla 4DS
Dallas, TX 75201
cljacks@ci.dallas.tx.us (work)
CJac@compuserve.com (home)