Using P-Charts to Track Data Quality

(Some observations based on a simulation study)

By

Elizabeth Pierce
MIS & Decision Sciences
Indiana University of Pennsylvania
203 Eberly College of Business
Indiana, PA 15705-1087

(412) 357-5773
(412) 357-4831 (Fax)

E-mail: EMPIERCE@IUP.EDU

Abstract

With the growing interest in data warehousing, many managers are concerned about the cleanliness of their data. The p-chart is a tool that is often used to track the proportion of defective items in a population. This article explores what kinds of data processing conditions are necessary for managers to effectively use p-charts to track their data quality metrics. Using a simulation package, I modeled several different data processing scenarios. From the simulated data process, I drew random samples of data, measured the error rate, and recorded the results on p-charts. I then examined the p-charts to determine under which conditions the p-charts were able to track the true level of errors in the data process and to spot out of control conditions.

Introduction

Many organizations want to make better use of their growing volumes of information by loading data from their legacy systems and external sources into a data warehouse. The data warehouse is a neutral data storage area specifically designed to allow different end-user applications and tools to quickly access those subsets of data that they require. By using data mining software to help analyze and uncover hidden patterns and factors in their data, end-users hope to improve tactical and strategic decision making. However, the success of the data mining operations is strongly dependent on the quality of data in the data warehouse. The cleaner the data; the more accurate the results produced by the data mining software.

To prevent dirty data from adversely affecting the quality of the data mining, the data must be cleansed. Removing dirty data involves filling in missing or unreadable fields, rectifying inconsistent data values, and removing duplicate records. In addition, information that is inaccurate or out-of-date must be corrected as well. In a typical data warehouse operation, each time a regularly scheduled fresh extraction is made from the legacy system, it must undergo an integrity-checking and data-cleansing routine. This can be both time-consuming and expensive. Thus it makes sense to improve the quality of data in the legacy system as much as possible so as to help minimize the amount of cleansing that must be done each time a fresh data extraction is made.

The Use of P-Charts for Tracking Data Quality

One way to improve the data quality of the legacy system is to apply process management and statistical quality control (SQC) to the data process. Originally applied to the manufacturing environment, the basic idea behind process management and SQC is the continuous improvement of a process based on a management style that focuses on teamwork, customers, and quick reactions to change combined with a strong statistical foundation. One of the most common statistical tools used in SQC are control charts which are designed to help managers study the variability of their systems. (Note: A full discussion of process management, statistical quality control, and control charts is beyond the scope of this paper, but for those interested, I have included some useful references at the end of this paper.)

Control charts can assist managers in differentiating a special or assignable cause of variation for dirty data from the common (random) variability in data quality levels that exists in a legacy system. By detecting and eliminating special causes for dirty data from a data process, the data process becomes more predictable (stable) in its data quality levels. In addition, control charts provide a visual display of a quality metric associated with a process. Managers can use control charts to track the effects of changes made to the legacy system to determine if the amount of common variation or average level of the quality metric has been improved.

There are several different types of control charts available. A manager should choose the type of control chart that matches the characteristic of interest that he/she would like to study in his/her process. For instance, quantitative variables are often tracked using control charts for the range and the mean while qualitative variables such as the proportion of defective items in a population are tracked using p-charts. For managers interested in tracking the proportion of defective values in their systems, the p-chart is one possible option to consider.

The p-chart is based upon the behavior of the binomial distribution. To use a p-chart, a manager must first pick a metric of interest, such as the percentage of defective addresses in a legacy system database. Next the manager must periodically sample n_t items from the legacy system database and count the number of defective addresses (x_t) found in the sample taken at time t. The manager then plots the proportion of defective addresses (x_t / n_t) found at time t on a p-chart containing the following control limits.

Upper Control (3σ) Limit = $\bar{\rho} + 3\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

Upper 2σ Limit = $\bar{\rho} + 2\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

Upper 1σ Limit = $\bar{\rho} + 1\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

Center Line = $\bar{\rho}$ = average proportion of nonconforming items in k samples

Lower 1σ Limit = $\bar{\rho} - 1\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

Lower 2σ Limit = $\bar{\rho} - 2\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

Lower Control (3σ) Limit = $\bar{\rho} - 3\sqrt{\frac{\bar{\rho}(1-\bar{\rho})}{\bar{n}}}$

By looking for unusual behavior in the p-chart pattern, the manager has a way of detecting when the stable quality level of his/her data has changed. Indications that the process may be out of control include any points beyond either 3σ limit. In addition, some out-of-control conditions may manifest themselves in patterns such as two out of three consecutive points beyond either 2σ limit, four out of five points beyond either 1σ limit, or eight consecutive points on the same side of the Center Line. It is important to note that control charts are not infallible. Just because a point falls outside of the 3σ limit does not guarantee the process is out of control. Using the 3σ control limits, you can expect on average 1 false alarm to occur in every 370 periods.

Design of Simulation Experiments

To examine under what conditions a p-chart would work well in a data processing environment, I decided to use the simulation package, GPSS/H, to model several different data processing scenarios. The general framework of each scenario can easily be altered to incorporate different batch sizes and schedules, as well as different assumptions about the type and frequency of the errors that occur. I chose simulation because the results are quick, repeatable, and inexpensive to obtain. However simulation has the disadvantage in that it is only as good as one's interpretation and translation from reality to simulation program. Thus the results from this study depend on how well these simulations replicate the real data that one would obtain from an actual data processing environment. The chosen scenarios were ones that I felt represented typical data processes. GPSS/H is powerful enough to handle additional scenarios given one's imagination and programming skills.

While each simulation scenario incorporates different elements, all the scenarios profiled contain a common logic pattern. The common underlying pattern used for each of the simulation scenarios follows these steps:

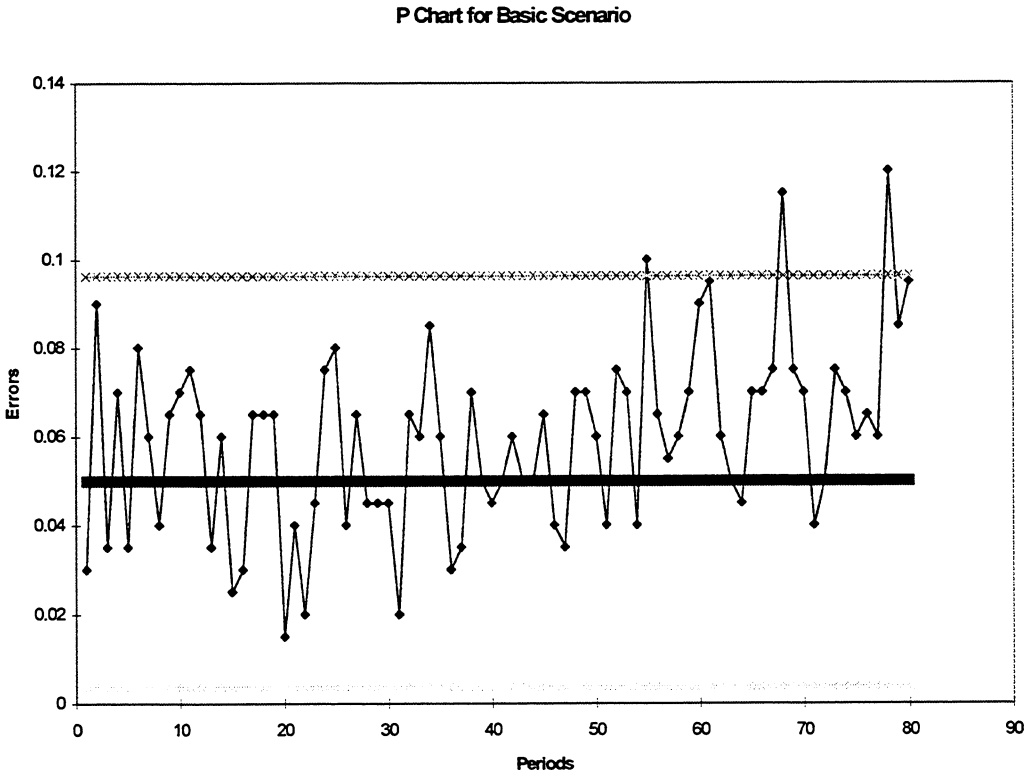
1. Initialize and set the starting conditions for the database.
2. Generate the incoming errors for a given database field based on a pre-specified error pattern.
3. Generate the incoming records based on a pre-specified arrival pattern.
4. Add the new records and any incoming errors to the database. Adjust the database statistics for any deletion of records, clean up of errors, or special processing as dictated by the scenario being simulated.
5. Calculate the current proportion of defective values in the database population, i.e. ρ = (total errors in database) / (total records in database).
6. Draw a random sample of size n from a simulated binomial random sample based on the value of ρ.
7. Print the results of the simulation for that iteration and repeat the process beginning at step 2. Repeat for as many iterations as desired.
8. For each simulation run, track the results of each random sample drawn from the simulation on a p-chart.

The Scenarios

To examine the performance of p-charts under a number of conditions, a number of different scenarios were simulated. These scenarios include both batch and on-line processing, periodic clean-ups of the files, periodic deletions of file records, and various error patterns. Along with each scenario, its respective p-chart is depicted along with an examination of the results.

Basic Scenario – "Monthly Batch/Sample from Batch"

In this scenario, a batch of records is added each period to the database. From past experience, it has been determined that the number of new records (n_i) is normally distributed with a mean of 1,000 and a standard deviation of 150. It has also been determined from past experience, that a particular field of interest contains an erroneous value approximately 4% to 6% of the time. It is also assumed that this type of error is usually generated through the input process and does not occur spontaneously once the record is in the database. This erroneous probability (p_e) will be modeled as a uniform distribution within the range of 4% to 6%. The number of erroneous field values (n_e) in a batch of new records (n_i) will be modeled as a binomial distribution with parameters n_i and p_e. If a manager decides to take a sample of 200 records (n = 200) from the incoming batch records, the number of erroneous values found in that sample (x_t) is modeled as a binomial distribution with parameters n and p_s where p_s = (n_e / n_i). The p-value that is plotted on the p-chart for that period t is calculated using the formula $\rho_t = (x_t / n)$. To make things interesting, at the 51st period assume that some changes are made in the batch process that causes the underlying erroneous probability (p_e) to shift from a uniform (4%, 6%) distribution to a uniform (6%, 8%) distribution. The simulation is run for 80 iterations and a p-chart based on a sample size of 200 is generated.

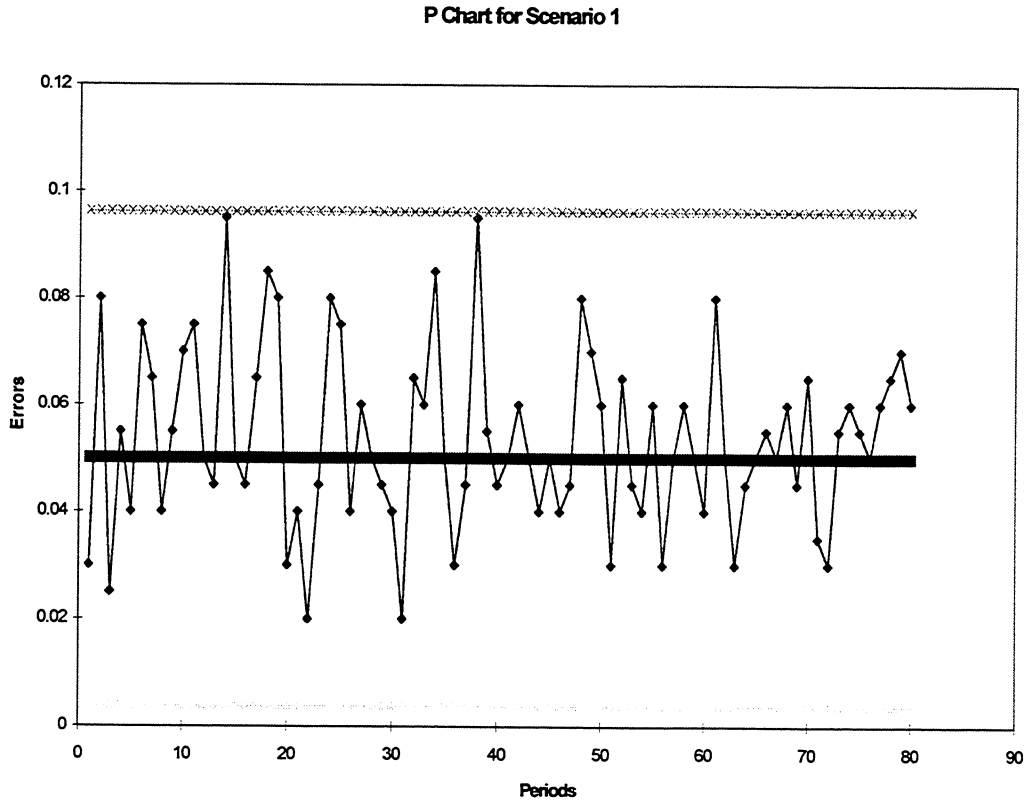


In the case of the Basic Scenario, the p-chart appears to perform well. The average level of error in the incoming records during the control period is 5%. There do not appear to be false alarms in the early periods when the quality level of the incoming records is stable. In addition when the quality level of the incoming transaction stream shifts in the 51st period, the p-chart exhibits a shift in its behavior soon after the change. Within a few

periods, one can see at least one point outside of the 3σ limits and the majority of the points are falling above the Center Line.

Scenario 1 – "Monthly Batch/Sample from Database"

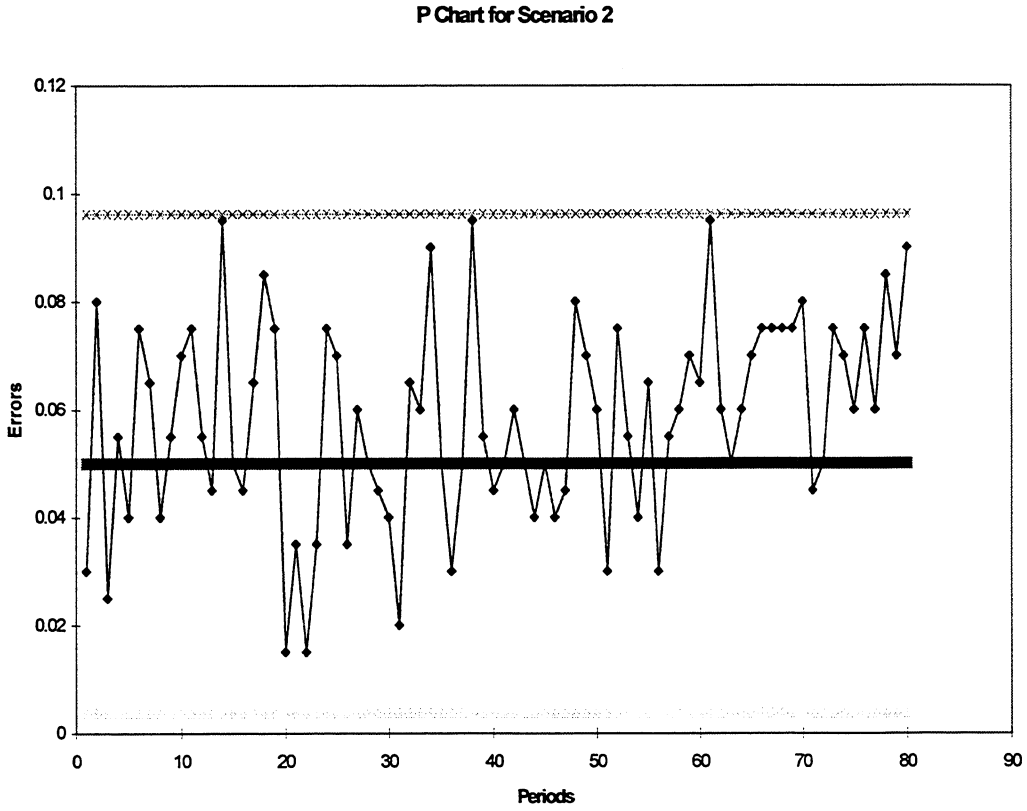
This scenario is identical to the Basic Scenario except now the manager decides to sample from the database population rather than from the incoming batch records. As a result, the sample of 200 records (n = 200) is now randomly selected from across the entire database of size Σn_i. For this scenario, assume that no further correction of errors is done once the records reach the database and no records are deleted. The number of erroneous values found in that sample (x_t) is modeled as a binomial distribution with parameters n and p_s where $p_s = \Sigma n_e / \Sigma n_i$ (i.e. the total number of erroneous values entered to date divided by the total number of records added to date). The simulation using the same random number seeds and quality shift at period 51 as the Basic Scenario is run for 80 iterations and a p-chart based on a sample size of 200 is generated.



For Scenario 1 the p-chart is less satisfactory. The average error rate for the database field during the control period is 5% just like the incoming records. In addition, the p-chart appears to function appropriately in the early periods. However as the volume of total records grow, the database becomes less sensitive to changes from the much smaller volume of the input stream. As a result, the p-chart gives no indication that a shift in the quality of the input stream occurs at period 51. Thus, a manager would be wise under this scenario to do his sampling as close to the source of input as possible.

Scenario 2 – "Monthly Batch/Sample from Database/Records Deleted"

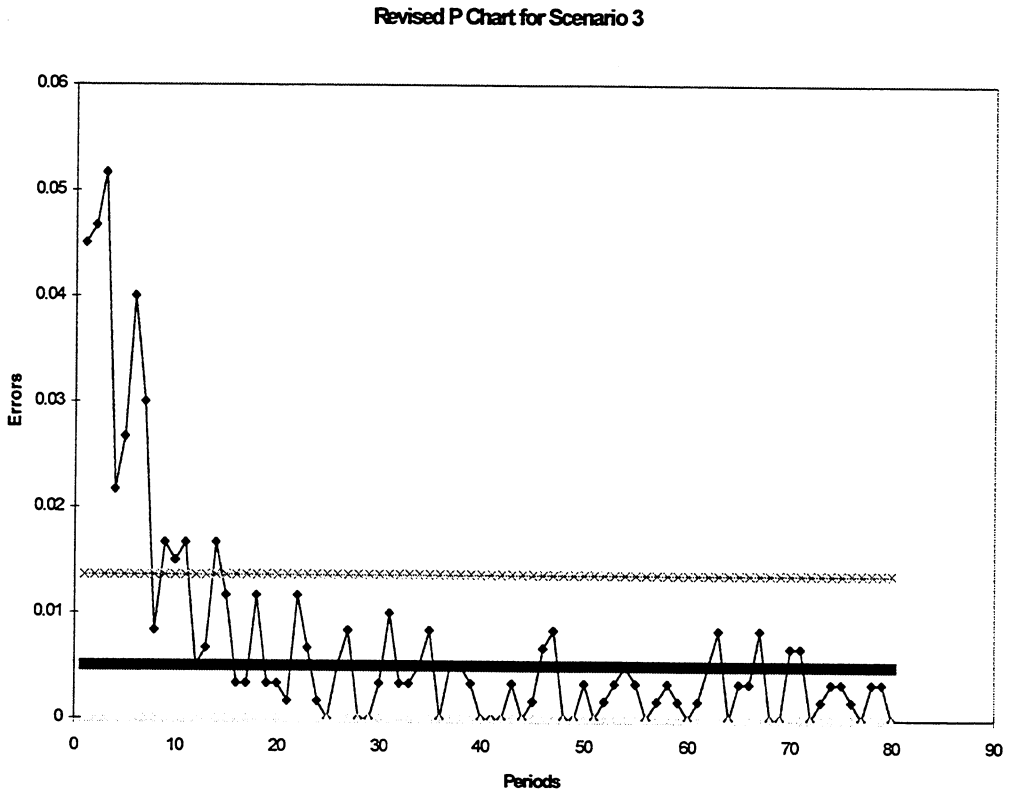
This scenario is very similar to scenario 1, but now assume that the database from which the manager is sampling is designed to hold only the 10 most recent periods. This type of scenario often occurs when storage space is limited or the usefulness of the information declines over time. For example, an on-line text retrieval service may only offer searches against publications issued within the last 20 quarters. Again no checking or correction of erroneous values is done once the records are loaded into the database. The simulation using the same random number seeds and quality shift at period 51 as scenario 1 is run for 80 iterations and a p-chart based on a sample size of 200 is generated.



For Scenario 2, the average error level for the database field during the control period is 5%, again matching what is occurring in the incoming records. The p-chart appears normal, tracking the stable quality levels in the early periods. Because this database is designed to hold only the 10 most recent periods, it is more sensitive to a change in the quality levels of the incoming records than Scenario 1 but not as sensitive as the Basic Scenario. After period 60, the p-chart is clearly indicating a change has occurred in the data quality level. One can expect that if a database holds k periods of data, there is roughly a k period of delay before the p-chart will indicate changes in the database quality levels.

Scenario 3 – "Monthly Batch/Sample from Database/Batch Checking"

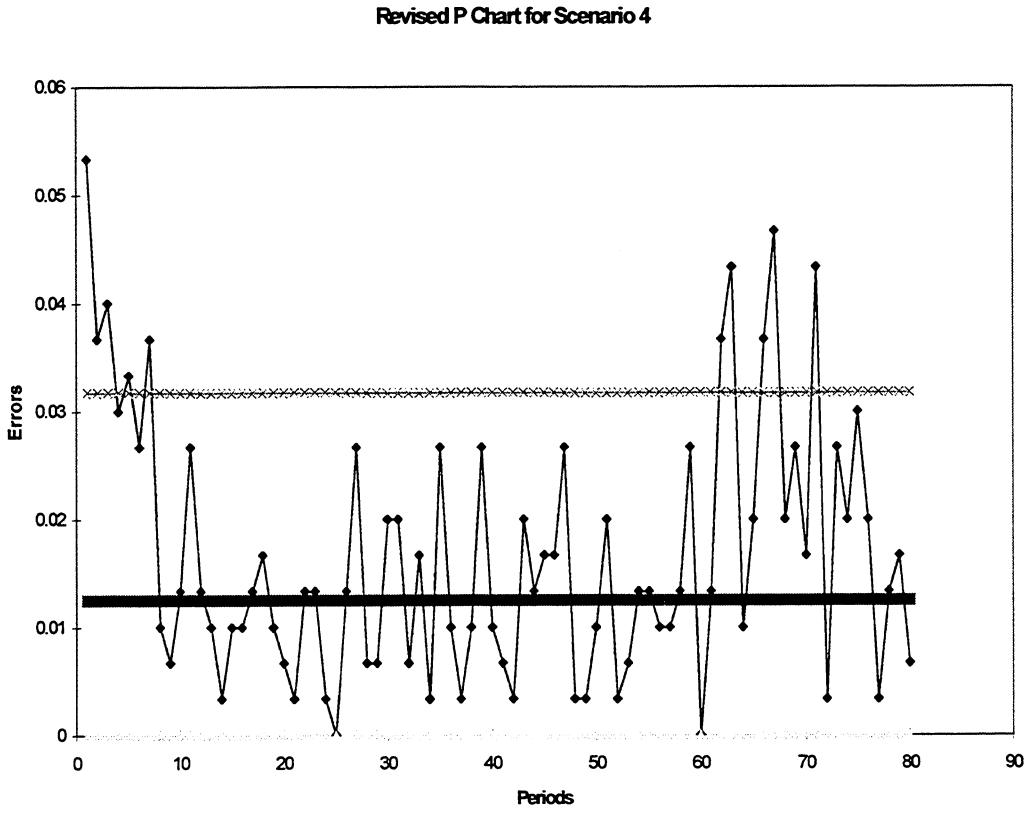
This scenario is similar to scenario 2 but it assumes that the manager is sampling from a database where batch corrections are applied every 4 periods. New records still arrive monthly in batch and no records are ever deleted from the legacy system. This might be the case where every 4 months, the data warehouse undergoes an extensive data cleansing operation. After the data cleansing phase takes place, a corrected file is back flushed to the legacy system. The simulation assumes that the cleansing process detects 100% of the errors. Again the simulation is run for 80 iterations and a p-chart is generated using the same random number seeds and quality shift at period 51 as the earlier scenarios.



For Scenario 3, the performance of the p-chart for detecting changes in the data quality levels is again poor. Because the database is growing in size combined with a clean-up occurring every 4 periods, the error percentages for that database field are gradually approaching zero. This renders the p-chart insensitive to quality changes in the incoming records. In addition, as the error rate approaches zero, the manager must continually increase his sample size to ensure a decent chance of finding one nonconforming unit per sample. For instance if the current erroneous value rate is ρ and we want the probability of at least one nonconforming unit in the sample to be at least 95%, then using the Poisson approximation to the binomial, we must choose the sample size, n, such that $\lambda = n\rho$ exceeds 3. In the case of an erroneous data value rate of 0.5%, this means $n > (3 / .005)$ or a sample size greater than 600. A revised p-chart for scenario 3 incorporating the larger sample size of 600 gives a clearer picture of what is happening in the database

Scenario 4 – "Monthly Batch/Sample from Database/Records Deleted/Batch Checking"

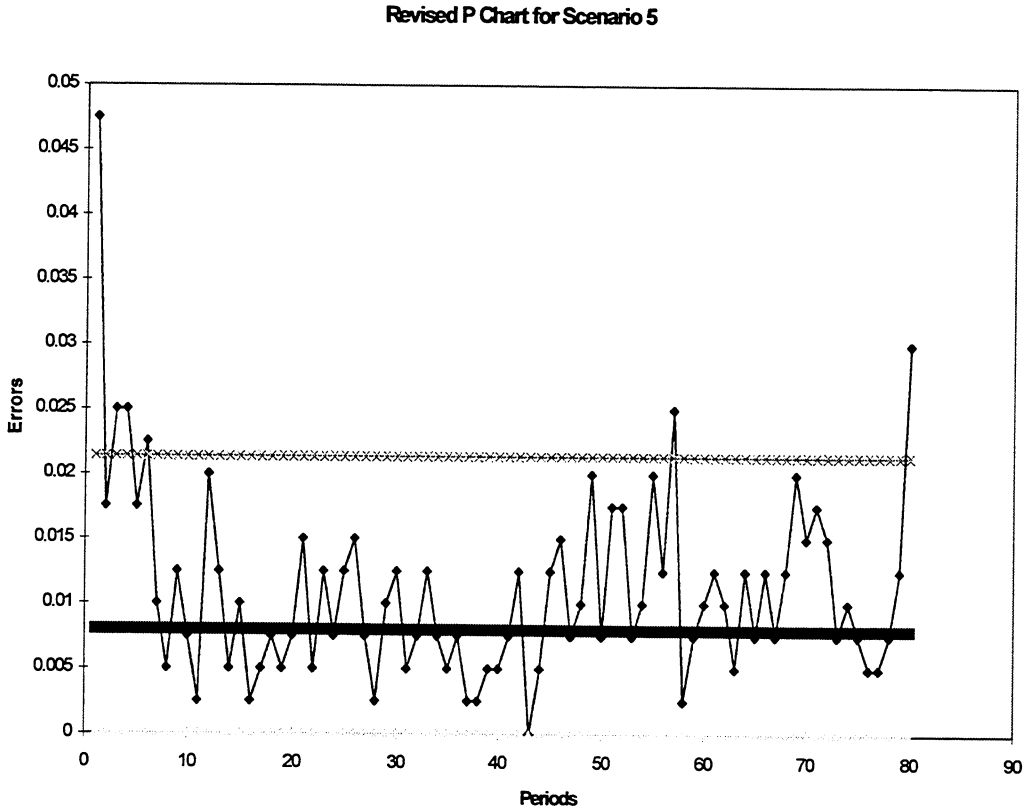
This scenario combines elements of both scenarios 2 and 3. Not only is a correction applied after every 4 periods, but this database is designed to hold only the 10 most recent periods as well. The simulation is also run for 80 iterations and a p-chart is generated using the same random number seeds and quality shift at period 51 as before.



For Scenario 4 when one combines keeping the last 10 periods of data along with clean-ups scheduled every 4 periods, the error level for the field averages 1.25% during the control period. The reason that the error level appears higher in scenario 4 than in scenario 3 is because the same number of incoming records with errors are being averaged over a smaller number of clean data records in the database. When I originally ran the p-chart using a sample size of 200, I found the chart to be insensitive to quality changes because the sample size was not big enough to effectively track such a small error rate. By increasing the sample size to 300, I got the p-chart that is included in this paper. The revised p-chart appears in control except for 2 false alarms in the first 10 periods. Since the simulation was starting the database from scratch, this may simply be due to random variation before the simulation settles down. Like the previous scenarios, this simulation was programmed to shift the quality of the incoming batch records at period 51. Although it takes about 10 periods, one can see the shift in the data quality levels taking hold in the later samples.

Scenario 5 – "Monthly Batch/Sample from Database/Records Deleted/On-Line Checking"

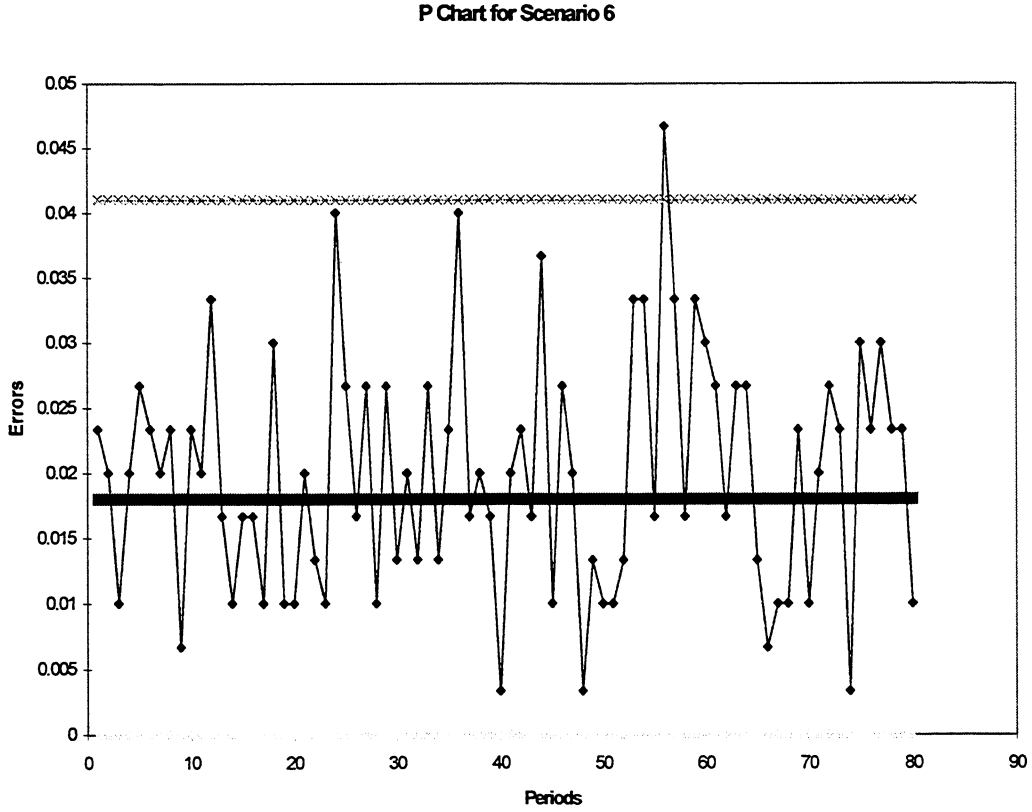
This scenario is similar to scenario 4 except now the simulation assumes that correction is done on-line as the erroneous values are discovered rather than in batch. Only records from the past 10 periods are kept. The simulation assumes that the time to detect and correct an erroneous data value follows a triangular distribution with a minimum of 0.05 period, an average of 2.1 periods, and a maximum of 6 periods. Each error that arrives in the database will be assigned a random time from this distribution. When the simulation determines that time has expired, the error is then removed from the database. This scenario is meant to mimic an error that stays in a database until someone discovers it and has the erroneous value corrected.



For Scenario 5, the average error level in the database is 0.8% during the control period. Although this is a lower error rate than Scenario 4, one cannot infer that on-line checking is better than batch since it all depends on the distribution of correction times chosen to model the on-line checking. Again because the erroneous data error rate is so low, a sample size of 200 is inadequate for tracking changes. As a result, I ran the simulation a second time using a larger sample size of 400. Examining the revised p-chart after the initial 10 periods when the simulation is warming up, it appears that the error levels in the database are stable. In addition, after period 51, one can observe a shift in the error levels in the database.

Scenario 6 – "On-Line Addition, Deletion & Checking of Records/Sample from Database

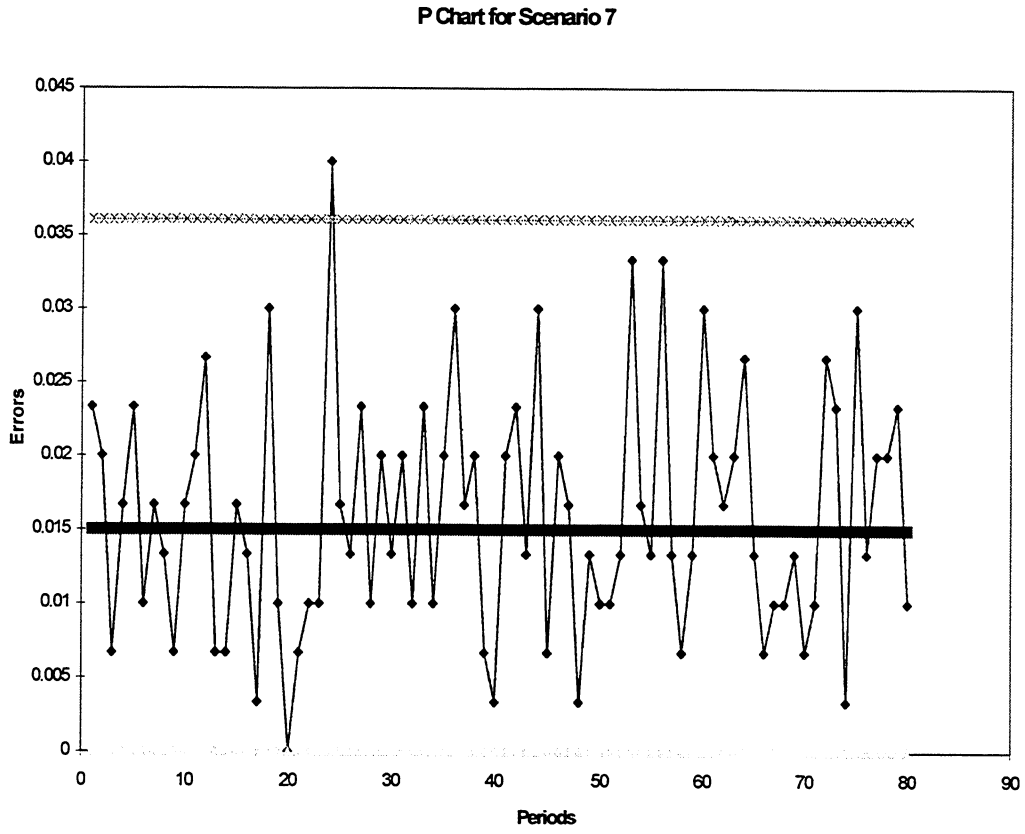
For some type of errors such as out-of-date information, the erroneous values do not come from the input stream. Rather they are based on factors affecting the database as a whole. Consider the scenario of a company's employee database containing 1,500 records. The employee's address field can be changed by such occurrences as an employee's move or even a re-addressing to accommodate a community's 911 implementation. Suppose from past experience, the manager feels comfortable modeling the number of new employees being hired each month as an approximately normal distribution with a mean of 12 and a standard deviation of 3. In addition, the manager feels that the number of employees that leave the company each month for whatever reason can be modeled as an approximately normal distribution with a mean of 8 and a standard deviation of 2. From studying the average number of moves each month, the manager feels that between 1.3% to 1.7% of the employee population moves each month. The simulation will model this by using an exponential distribution to simulate the intervals between moves occurring. In addition, suppose corrections are done on-line and the time to correct the address change once it has occurred follows a triangular distribution with a minimum of 0.25 month, an average of 1.25 months and a maximum of 2.25 months. Each month the manager plans to take a sample of 300 records from the database and calculate the proportion of out-of-date addresses found and plot this as a p-value on a p-chart. This simulation is repeated for 80 iterations. To see how well the p-chart detects changes, the simulation is programmed to shift the address change rate to a range of 1.5% to 1.9% of the current employee population, starting at the 51st period.



For Scenario 6, the p-chart appears adequate. Under this scenario, the average percentage of out-of-date addresses in the database during the control period is 1.8%. During the first 50 periods, the p-chart does not exhibit any false alarms or indications that the quality levels for the address field is unstable. After the 51st period, the p-chart does seem to shift upward indicating that the underlying quality level for the address field has changed. The change is more difficult to see in this chart because the in control move rate (1.3% to 1.7%) overlaps with the out of control move rate (1.5% to 1.9%). This type of situation demonstrates that p-charts can have difficulties detecting small shifts in the underlying error pattern.

Scenario 7 – "On-Line Addition, Deletion/Monthly Batch Checking/Sample from Database

This scenario is similar to scenario 6, but this time rather than correct errors as they come in, the manager decides to wait and do the corrections as a batch every month. This simulation is repeated for 80 iteration using the same random number seeds as scenario 6. Again, a p-chart based on a sample size of 300 is generated.

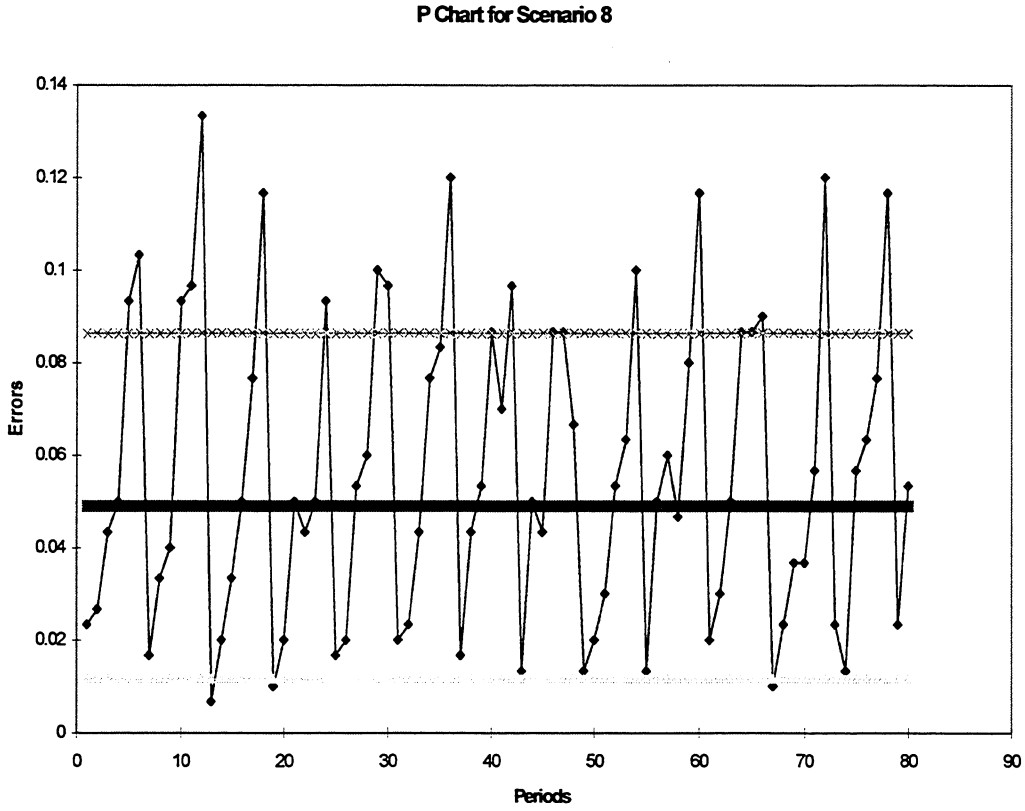


For Scenario 7 the average out-of-date address rate during the control period is 1.5% in the employee database. This p-chart displays one value outside of the 3σ limits during the period where the underlying simulation has not changed; otherwise the quality level appears stable. The problem with this p-chart is its inability to detect a shift in the move rate in the employee population. By doing a batch correction each month, the database is insensitive to at least minor changes in the process. I did experiment with larger, more distinct shifts in the

underlying move rates and found evidence that the p-chart did respond in these cases. This suggests a manager may need to resort to larger p-chart samples or to switch to a different type of control chart in order to detect small changes in data quality.

Scenario 8 – "On-Line Addition, Deletion/Biannual Batch Checking/Sample from Database

This scenario is similar to scenario 7, except now the error correction occurs in batch every 6 months. This simulation is repeated for 80 iterations using the same random number seeds as scenario 7. Again a p-chart based on a sample size of 300 is generated.



Scenario 8 demonstrates what happens when errors that are not restricted to the input process are permitted to build. With the earlier scenarios, the levels of errors were restricted to just the incoming records, thus limiting the scope of dirty data. The problem with an error like obsolescence is that it springs from the database itself. For example, over time the erroneous address rate in a database could approach 100% if there is no updating of information. In the case of corrections every 6 months, one can see that the average out-of-date address rate in the database is approximately 5% during the control period. However there is a cyclical pattern in effect. The error rates gradually build to a peak over 6 months before dropping off at the next scheduled cleanup and then beginning the pattern all over again. Because the underlying process is not a stable but cyclical, this makes the p-chart

difficult to use and interpret. In addition to the number of false alarms registered by the p-chart, the chart does not indicate any clear signs that the underlying move average has changed after period 51.

This may be an example of a mixture effect. Essentially there are 6 averages, one for the data quality level right after cleanup, another for the data quality level one month after the cleanup, another for the data quality level two months after clean-up and so on. Under this scenario, the manager may need to formulate multiple p-charts (one for each different level) to track this process in order to detect shifts in the data quality levels beyond the cyclical pattern.

Observations

Based on these simulations, I have these observations about using p-charts to track data quality.

(1) It is important to know the source for one's errors since that will dictate the optimal position for sampling for the p-chart. The closer you can place the p-chart to the source of the errors, the more sensitive the chart will be to detecting changes in the data quality levels. If the errors are coming from the incoming records, then that is where you should sample for your p-chart, not from the database itself. Particularly in a large database that is periodically cleaned, the effects of a data quality change in incoming records will be diluted in the database to the point that sampling from the database itself is ineffective in detecting any change in the input stream. This dilution occurs simply because the large number of clean records in the database will overwhelm the number of incoming dirty records. The end result is a very small database error rate.

Only in the case that errors are arising while the records are in the database, should you consider p-charts based on sampling from the database. Examples of errors arising from the database would include situations like out-of-date information or data damaged due to viruses, storage device failures, or software problems. Under these circumstances, a p-chart based on sampling from the database itself would be appropriate for detecting changes to the data's quality levels.

(2) The smaller the error rate you want to monitor, the larger the sample size you must pick. Even a relatively large number of errors, such as 15,000 incorrect addresses, may be difficult to detect when sampling from a database with over a million different records. One rule of thumb from Montgomery (1991) states that if you want a 50% chance of detecting a process shift of some specified amount, δ, with a 3σ limit p-chart then you must select a sample size, $n = \left(\frac{3}{\delta}\right)^2 \rho(1-p)$ where ρ is the mean proportion in your process. This means if your error rate is already low, say 1.5% and you want to detect a 0.5% increase to 2% then you would need a sample size of n =

$\left(\frac{3}{.005}\right)^2 (.015)(.985) = 5{,}319$. This means that unless you have an easy means to check such a large n on a frequent basis, the p-chart might be economically infeasible. In addition, if you have a relatively small database, you may simply not have enough records to detect such small shifts in data quality.

(3) Because the proportion, ρ, is the number of erroneous values divided by the total number of records, interesting things can happen when records are deleted. For the simulations modeled, I assumed records being deleted were similar in their error properties as the rest of the database; hence, their elimination was independent of the underlying error pattern. However, suppose the errors are concentrated in the records being discarded. Eliminating error prone records would cause ρ to shrink. On the other hand, if one is discarding records that are relatively clean compared to the rest of the database, one could cause a situation where ρ increases because now there is a greater number of errors per the remaining records.

(4) The greater the underlying stability of the process, the greater the effectiveness of the p-charts. Managers will probably find that p-charts are most useful in tracking data quality when the underlying quality levels are very stable. If the incoming batch records are usually 95% accurate, this is a process that is stable and predictable and the p-chart will track it with few false signals. On the other hand, if the quality of the incoming batch records fluctuates on a regular basis, some weeks 95% accurate, other weeks 80% accurate, and so on, this will make the p-chart less useful. In playing with the simulations, I modeled a few cases where there was 5 or more percentage point variations around a central mean. The resulting p-chart generated a high number of points outside of the 3σ limit which would have caused the manager to spend a lot of time investigating whether a true out-of-control condition exists or if the point is simply that week's wild variation.

In closing, I plan to further explore the use of p-charts and other types of control charts for tracking data quality metrics using simulation techniques as well as obtaining real data to help validate these observations. Eventually I hope this research will lead to additional insights into strategies for implementing a data quality tracking system that is both economically feasible and effective for monitoring data quality levels.

General References

Anderson, D., D. Sweeney, and T. Williams, "Statistics for Business and Economics", West Publishing Company, Minnesota, 1993.

Banks, J., J.S. Carson, and J. Ngo Sy, "Getting Started with GPSS/H", Wolverine Software Corporation, Virginia, 1989.

Levine, D.M., M.L. Berenson, and D. Stephan, "Statistics for Managers", Prentice Hall, New Jersey, 1997.

Mattison, R., "Data Warehousing: Strategies, Technologies and Techniques", McGraw-Hill, New York, 1996.

Montgomery, D.C., "Introduction to Statistical Quality Control (2nd Edition)", John Wiley & Sons, New York, 1991.

Pierce, E., "An Analysis of Data Error Rates Using a Stochastic Queuing Model", Doctoral Thesis at The University of Michigan, Ann Arbor, 1996.

Redman, T.C., "Data Quality: Management and Technology", Bantam Books, New York, 1992.

Ross, S.M., "Introduction to Probability Models (4th Edition)", Academic Press, California, 1989.

Schriber, T.J., "An Introduction to Simulation", John Wiley & Sons, New York, 1991.