The Virtual Mailing List

By

Elizabeth M. Pierce
empierce@grove.iup.edu

Eberly College of Business
Indiana University of Pennsylvania
Indiana, PA 15705 USA

Abstract

Demonstrating to students how the quality of an actual database can change over time can be difficult. In this paper, the author has created a Visual Basic program that permits students to experiment with a virtual mailing list. In this simulated environment, students can specify a variety of data processing parameters for their mailing list and then run the simulation to see how the percentage of mailing list records with deliverable addresses will change over time.

Introduction

For individuals trying to study the dynamics of data quality, there are several obstacles. First, getting access to an actual database is often difficult because data managers are reluctant to let someone from outside the company look at their data. There are several reasons for this. The data may be confidential or proprietary in nature. The data manager may also be concerned that the outsider might accidentally or purposely delete or alter the contents of the database. If the outsider issues many queries in conducting the data quality study, the updating performance of the database might be degraded. Finally, if the outsider's objective is to study data quality, the data manager might worry about the publication of negative results concerning the data's quality.

A second obstacle concerns the logistics of studying data quality in an actual database. Most production databases are changing over time. Not only may the data be changing, but the way records are being maintained in the database may be changing as well. A quality measurement taken on one day will probably be different than a quality measurement taken on another day. Another logistical problem is how to check the data. While some errors like missing fields, bad codes, or improper formatting may be checked by computer, other types of quality indicators like the accuracy of an address must be checked against an independent data source. For large databases, this could involve considerable time and money.

To assist students in understanding the dynamics of data quality, I have created a Visual Basic program to allow students to observe the data quality patterns in a virtual mailing list. Using a GUI interface, students are able to determine the data processing conditions for their virtual mailing list and then run the simulation for any desired period of time. The program allows students to see how the quality of the mailing list is changing and to see how modifications to the data environment might impact the quality of their mailing list. The simulation also gives students an opportunity to discuss how their assumptions concerning the way errors in the mailing list are introduced, detected, and corrected should be incorporated into their virtual data processing environment.

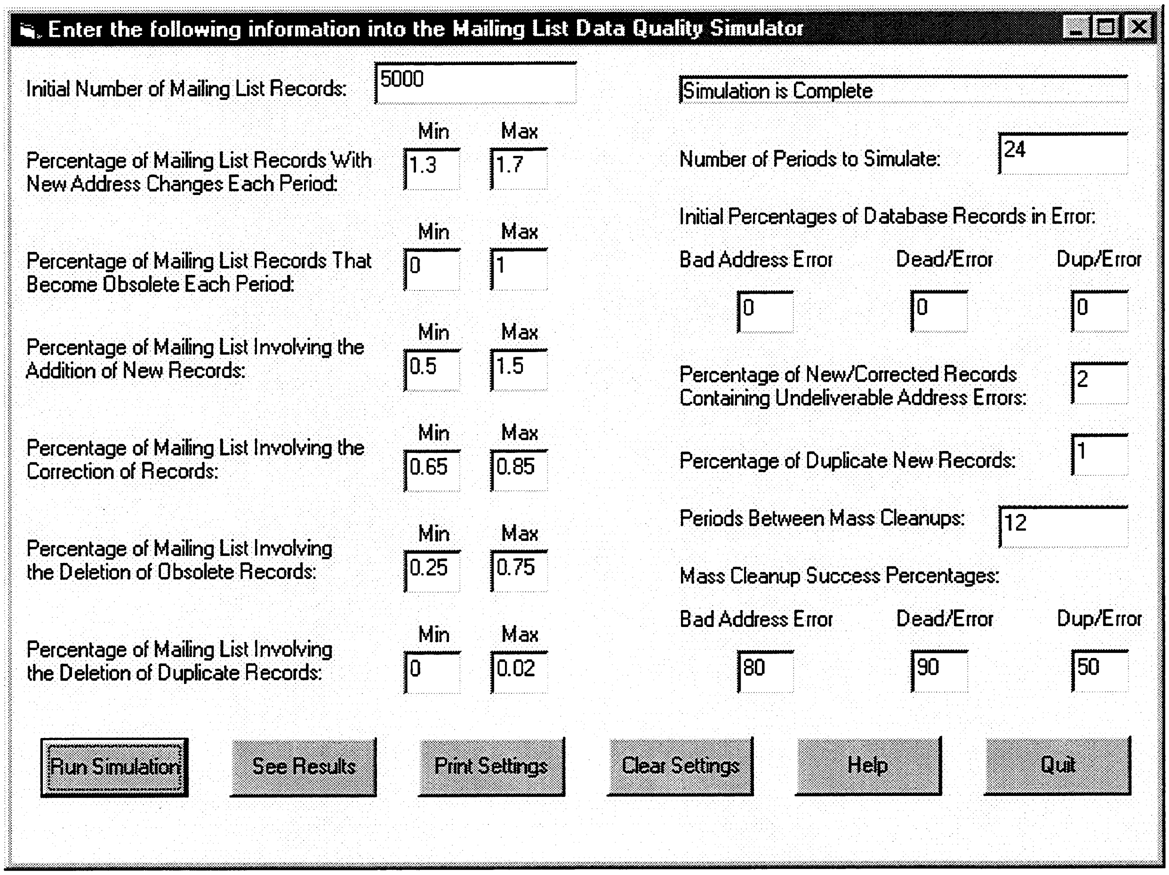
The mailing list was chosen for several reasons. First, it is an easy database for students to visualize. The program assumes that one is dealing with a simple mailing list based on a standard address format of the recipient's name, delivery address, city, state and zip+4. Secondly, the mailing list was chosen because it represents a significant business data quality problem. Standard A advertising mail has grown nearly 15% in the past five years (over 7% from 1996 to 1997) to more than 77 billion mailpieces (USPS, 1999). According to the Postal Service National Address Information Center, more than 7% of Standard A advertising mail is not deliverable as addressed (USPS, 1998). In fact, mailers' address lists on average are only about 67% accurate (USPS, 1998). The biggest problems with addresses are an incorrect or missing directional suffix (8.9%), customer has moved (8.6%), wrong street name or number (6.2%), wrong zip code, city, or state (4.7%), wrong or missing rural route or box number (2.5%), and wrong or missing apartment number (2.2%) (USPS, 1998).

The savings from cleaning up a dirty mailing list are significant. Working with the U.S. Postal Service, Dow Jones Inc. reported that they were able to raise the accuracy of their mailing list from 86% to 98% correct. Dow Jones believes it will save about $250,000 annually in printing and mailing costs. (USPS, 1998). Surprisingly, fewer than half (47%) of Standard A and only 19% of government mailers use the U.S. Postal Services' National Change of Address (NCOA) program for updating their mailing lists (USPS, 1999). The primary reason cited for not using the NCOA is lack of information on how to use it (USPS, 1999). This simulation program demonstrates; however, the impact that these cleanup programs can have on a mailing list and can help to convince students why companies should spend the time investigating how to use a mailing list management service to improve the quality of their address data.

The remainder of this paper explains how the simulation program works. A copy of the pseudo code is included in the appendix. For those interested, a soft copy of the simulation program itself is available from the author, simply e-mail the request to empierce@grove.iup.edu.

The Virtual Mailing List Interface

To Install the Program: Create a folder called "C:\DQSIM". Unzip the file, "dqsim.zip", that comes on the install diskette and extract all the files into the "C:\DQSIM" directory. Once installed, simply execute the file: dqsimulate.exe and the following menu should appear.



Enter the following information into the Mailing List Data Quality Simulator

Initial Number of Mailing List Records: 5000

Percentage of Mailing List Records With New Address Changes Each Period: Min 1.3 Max 1.7

Percentage of Mailing List Records That Become Obsolete Each Period: Min 0 Max 1

Percentage of Mailing List Involving the Addition of New Records: Min 0.5 Max 1.5

Percentage of Mailing List Involving the Correction of Records: Min 0.65 Max 0.85

Percentage of Mailing List Involving the Deletion of Obsolete Records: Min 0.25 Max 0.75

Percentage of Mailing List Involving the Deletion of Duplicate Records: Min 0 Max 0.02

Simulation is Complete

Number of Periods to Simulate: 24

Initial Percentages of Database Records in Error:

Bad Address Error	Dead/Error	Dup/Error
0	0	0

Percentage of New/Corrected Records Containing Undeliverable Address Errors: 2

Percentage of Duplicate New Records: 1

Periods Between Mass Cleanups: 12

Mass Cleanup Success Percentages:

Bad Address Error	Dead/Error	Dup/Error
80	90	50

Note: Sample data has already been added for illustration purposes. Normally the menu's text boxes would be blank.

Control Parameters for the Simulation: The following paragraphs describe the parameters that can be specified for the simulated mailing list. All examples are based on the sample data shown in the Virtual Mailing List's menu screen.

Initial Database Size: In this text box, students can enter the number of initial records in the simulated database. The simulation will use this number as the starting number of records in the database. This number should be entered without any punctuation (i.e. no commas or decimal points). Example: The sample data depicts a scenario involving a company that has just purchased a mailing list containing 5,000 address records.

Address Change Percentage Rate: Each period, the simulation will assume that a certain percentage of individuals who are alive will have their address record become undeliverable due to changes in their name and/or address. The simulation allows students to specify 2 percentages: minimum and maximum. Using the uniform distribution, the simulation will make random draws each period. This random draw reflects the number of address changes that were added to the database during that period. The simulation will assume that individuals move once during the period. If multiple moves per individual occur, the simulation will over count this as two individuals that moved rather than one individual who moved twice. Example: Based on historical records, the company believes that between 1.3% and 1.7% of its customers will change their address each month.

Note: The uniform distribution was chosen because it is a useful distribution when a variable is known to be random, but no information is readily available about the shape of the distribution. Where additional information is available about the random behavior of the variables, this simulation could be improved by switching to a more informative distribution such as the triangular or beta distributions.

Note: All percentages entered into the simulation program should be between 0 and 100 percent.

Death Percentage Rate: Each period, the simulation will assume that a certain percentage of individuals who are alive and whose records are in the database will die or in some other way cease to be valid customers. The simulation allows students to specify 2 percentages: minimum and maximum. Using the uniform distribution, the simulation will make random draws each period. This random draw reflects another source of undeliverable addresses that occurred in the database during that period. The simulation will assume that removals due to death or other reasons come equally to customers with valid and invalid records. Hence when a percentage of records are marked for death, a corresponding proportion of erroneous duplicate and non-duplicate records are moved to their respective death categories. Example: The company assumes that between 0% and 1% of its customers will become undeliverable each period because of death or other problems.

Note: The simulation will also assume that address changes and deaths are disjoint events. That is, it is unlikely that someone both dies and changes their address during the same period. If this does occur, the simulation will count this as two erroneous records rather than one record with two different types of error. This can lead to over counting in the simulation program.

Percentage of New Records Added: Each period, a certain number of records are added to the simulated database. The simulation will calculate the number of new records based on a percentage of the existing records in the mailing list at the start of the simulated period. Example: Based on the mailing list's size, the company plans to add between 0.5% and 1.5% new customers to the mailing list each month.

Percentage of Records Updated: Each period, a certain number of records are updated in the simulated database. The simulation will calculate the number of updated records based on a percentage of the existing records in the mailing list at the start of the simulated period. The number of records updated will not affect the number of records in the database. This simulation assumes that records are updated either to correct in advance for an upcoming name or address change or to correct for some existing name or address error in the record. Because of data entry errors, not all updates are successful. The simulation will assume that sometimes in correcting an error, another error will be introduced in the same record. The simulation will also assume that first erroneous records are updated (this includes duplicates as well as non-duplicate records). If there are more updates than erroneous records, the simulation will then apply the updates to the non-erroneous records. The simulation allows one update per record per period. Multiple updating of the same record within the same period is not supported in the program. Example: Through address correction returns and customer correspondence, the company feels confident they can detect and correct about half of the address chances (0.65% to 0.85% of the mailing list) that occur each month.

Note: The reason the simulation does not support multiple updating is for simplification reasons. If multiple updating is allowed and say 10 updates occur, then how does one differentiate between a case of one record being updated 10 times or 10 records each receiving one update or something in between? To simulate multiple updating, more information would be needed to generate a probability distribution to describe the occurrence of multiple updating.

Percentage of Obsolete Records Deleted: Each period, a certain percentage of the invalid customer records are deleted in the simulated database because the customers are no longer valid customers (i.e. customer may be deceased or may have asked to be removed from mailing list). The simulation will calculate the number of deleted records based on a percentage of the existing records in the mailing list at the start of each simulated period. This simulation will assume that only customers who are dead or no longer valid customers will be deleted. If the number of deletions exceed this point then no further deletions are performed. Example: The company estimates based on past experiences with mailing lists that between 0.25% and 0.75% of the mailing list will be deleted each month because those customers were reported deceased or asked to be removed from the mailing list for other reasons.

Percentage of Duplicate Records Deleted: Each period, a certain percentage of the duplicate customer records are detected and deleted from the simulated database. The simulation will calculate the number of deleted duplicate records based on a percentage of existing records in the mailing list at the start of each simulated period. This simulation will assume that duplicate records are just as likely to contain errors as non-duplicate records. As a result, when duplicate records are deleted, a corresponding proportion of errors is deleted as well. The simulation will only delete the duplicate records when this percentage is specified. Example: A few delete transactions (between 0% and 0.02% of the mailing list records) will be deleted each month because they are known duplicate records.

Number of Periods: In this text box, students can specify the number of periods that they wish the simulation to run. Students should not use any punctuation such as commas or decimal points when specifying this number. Example: For this simulation, the sample data specifies 24 simulated monthly periods.

Initial Error Percentage Rate: Few databases start off completely clean. The simulation allows students to specify 3 initial error percentages. The Bad Address text box reflects the percentage of records in the database that pertain to living customers whose records are undeliverable due to name or address errors. The Dead/Error text box reflects the percentage of records in the database that pertain to customers whose address is undeliverable primarily because they are dead (or otherwise unavailable). These records may have other address problems as well. The Dup/Error text box reflects the percentage of duplicate records in the database. These duplicate records may contain other errors (bad address, dead customer) as well. Example: For this scenario, the company will assume that their initial mailing list of 5,000 address records is 100% accurate (no bad records, duplicates, or deceased individuals).

Note: The simulation assumes that mail is undeliverable because of two main reasons, (a) customer is alive but has a bad address or (b) customer is dead (address may have other problems as well). These two percentages should sum to the total percentage of undeliverable mail in the population. Records where the customer is both dead and address is in error should be counted in the Dead category. Records where the customer is living and address is in error should be counted in the Bad Address category.

Data Entry Error Percentage Rates: Each time a record is either added or changed in the database, there is the possibility that an error could occur somewhere in the address. This simulation assumes that the chance of an error is the same whether the record is being added for the first time or updated for the 100th time. Example: This company assumes that data entry personnel make mistakes about 2% of the time when entering or changing records.

Percentage of Duplicates: A certain percentage of the new records that are added each period to the database may in fact be duplicates of existing records. The duplicate records may or may not be error free. In this text box, students should enter a percentage between 0 and 100 to reflect the percentage of incoming database records that are redundant. If no duplicates, then students can simply indicate this with a zero. The simulation assumes that duplicate records can only occur through the entry of new records. Example: The company suspects that 1% of the new records being added each month are actually duplicates of existing customer records in the mailing list.

Periods between Cleanups: In addition to routine updates, it is possible that students may wish to schedule a mass cleanup of the database. For instance, many mailers use the U.S. Post Office's National Change of Address Service on a periodic basis to clean their records. In this text box, students can specify the number of periods that elapse between cleanups. If NCOA is used every period, students should specify 1. If mass cleanups are never employed, students should specify a number larger than the number of simulated periods. Students should not use any commas or decimal points when specifying this number. Example: This company plans to conduct a special cleanup on the mailing list every 12 months.

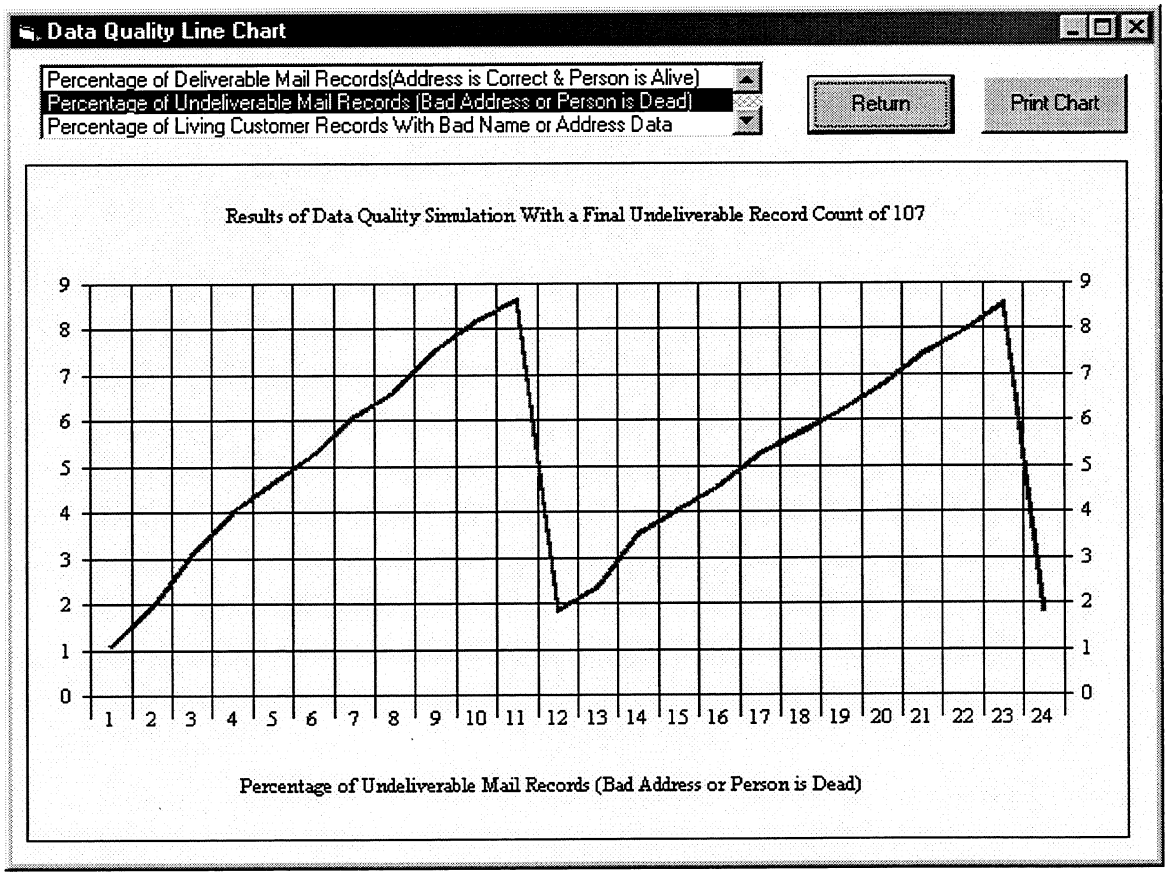
Percentage Success of Cleanups: It is quite possible that mass cleanups are not 100% successful. The simulation allows students to specify the percentage of existing errors in the database that students believed were detected and removed from the database during the mass cleanup effort. The simulation allows students to specify the percentage of records with bad names or addresses that were cleaned, the percentage of dead customers that were removed, and the percentage of duplicate records that were removed. Example: The sample data specifies that corporate personnel can detect and correct 80% of the records with bad addresses, 90% of the records belonging to deceased customers, and 50% of the duplicate records during a mass cleanup.

Run Simulation: This button runs the simulation for the specified parameters for the given number of periods. If the simulation completes successfully, a message should appear to that effect above the command buttons. If an error occurred, an error message will be displayed and the simulation will not run until the corrections are made.

See Results: This button takes students to a screen that displays a line chart of the data quality results. From that screen, students can either print a copy of the chart or return to the main screen. Students can also use the list box to see other data quality diagnostic charts. The charts that can be displayed include:

- Percentage of Deliverable Mailing Records Over Time
- Percentage of Non-Deliverable Mailing Records Over Time
- Percentage of Living Customers Mailing Records with Bad Addresses Over Time
- Percentage of Deceased Mailing Records Over Time
- Percentage of Duplicate Mailing Records Over Time
- Number of Mailing List Records Over Time

Using the sample data inputted on the main menu screen, a sample chart that displays the percentage of records in the mailing list that have non-deliverable addresses is shown below. The chart demonstrates that the percentage of undeliverable address records start just below 1% and gradually rises over the course of the year to nearly 9%. During month 12, a special mass cleanup of the mailing list dramatically reduces the percentage of undeliverable mail; however, the effects are short lived. The quality of the mailing list gradually deteriorates over the next 12 months to its prior levels.


Data Quality Line Chart
Percentage of Deliverable Mail Records(Address is Correct & Person is Alive)
Percentage of Undeliverable Mail Records (Bad Address or Person is Dead)
Percentage of Living Customer Records With Bad Name or Address Data
Return
Print Chart
Results of Data Quality Simulation With a Final Undeliverable Record Count of 107
9
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Percentage of Undeliverable Mail Records (Bad Address or Person is Dead)

By experimenting with the simulation parameters such as the cleanup schedule, the sources of data pollution, and the level of ongoing detection and correction of records, students can see how much they can make their mailing list's quality vary. They can experiment to see under which conditions the quality of the mailing list will stabilize to acceptable levels and under which conditions the mailing list's quality will remain in flux. In addition, by taking postage and printing costs into consideration, students can use the virtual mailing list model to make decisions as to how best to match the amount spent on quality improvement with the amount saved by increasing the percentage of deliverable mailing records.

Remaining Buttons: The remaining buttons provide some additional features.

- **Print Settings:** This button allows students to print a copy of their simulation settings.
- **Clear Settings:** This button allows students to clear their simulation settings.
- **Help:** This button takes students to the Help Screen.
- **Quit:** This button ends the simulation program.

Summary

In summary, the virtual mailing list is meant to be a teaching tool to help students understand the dynamics of data quality. Students can use the software to visualize what will happen to their data's quality over time based on a given set of assumptions and parameters. Students should also be encouraged to change the model's code so they can incorporate different assumptions about the way errors are introduced, detected, and corrected in their virtual mailing list world to further their understanding of data quality dynamics.

References

U.S. Postal Service (1998), "Dow Jones Saved $250,000 by Cleaning Address Lists, Memo to Mailers, Vol. 33, No. 5.

U.S. Postal Service (1999), Greening the Mail: Recommendations of the National Task Force on Greening the Mail, Final Report (January), pp. 13-14.

Appendix – Pseudo Code

1. **1.** Read input from screen into variables. Check that percentages, numbers are in the correct ranges.

2. Set Initial Starting Conditions

- Let Number of Duplicate Records (Dup_No) = Number of Records * Initial Duplicate Percentage.
- Let Number of Non Duplicate Records (NonDup_No) = Number of Records – Dup_No. Note: There should be at least one non duplicate record in the database. The code will insert one and reduce the duplicate number by one in the rare case that someone specifies a database composed of 100% duplicate records.
- Let Number of Dead, Duplicate Records (DeadDup_No) = Dup_No * Initial Dead Percentage.
- Let Number of Dead, Non Duplicate Records (Dead_No) = NonDup_No * Initial Dead Percentage.
- Let Number of Bad Address Duplicate Records (BadAddrDup_No) = Dup_No * Initial Bad Address Percentage.
- Let Number of Bad Address Non Duplicate Records (BadAddr_No) = NonDup_No * Initial Bad Address Percentage.

Summary of Relationships for Key Program Variables		
Total Number of Records in the Mailing List **(Record_No = NonDup_No + Dup_No)**	Total Number of Non Duplicate Records **(NonDup_No)**	Total Number of Duplicate Records **(Dup_No)**
Total Number of Records pertaining to Customers who are Dead. **(Dead_No + DeadDup_No)**	Total Number of Non Duplicate, Dead Records **(Dead_No)**	Total Number of Duplicate, Dead Records **(DeadDup_No)**
Total Number of Records pertaining to Living Customers with Bad Addresses. **(BadAddr_No + BadAddrDup_No)**	Total Number of Non Duplicate, Living, Bad Records **(BadAddr_No)**	Total Number of Duplicate, Living, Bad Records **(BadAddrDup_No)**
Total Number of Records pertaining to Living Customers with Correct Addresses. **(A + B)**	**(A) NonDup_No – Dead_No – BadAddr_No**	**(B) Dup_No – DeadDup_No – BadAddrDup_No**

3. Repeat the following for the specified Number of Simulated Periods

A. Generate Changes that Occur During the Period

- Let Moving Percentage (Movers) be generated from Uniform Distribution using screen inputs.
- Let New Duplicate Movers (Dup_NewMovers) = A percentage of the Living, Correct Duplicate Records, i.e. (Dup_No – DeadDup_No – BadAddrDup_No) * Movers.
- Let New Non Duplicate Movers (NonDup_NewMovers) = A percentage of the Living, Correct Non Duplicate Records, i.e. (NonDup_No – Dead_No – BadAddr_No) * Movers.
- Let Delete Percentage (Deceased) be generated from Uniform Distribution using screen inputs.
- Let New Duplicate Deceased (Dup_Deceased) = A percentage of the Living Duplicate Records, i.e. (Dup_No – DeadDup_No) * Deceased.
- Let New Non Duplicate Deceased (NonDup_Deceased) = A percentage of the Living Non Duplicate Records, i.e. (NonDup_No – Dead_No) * Deceased.
- Let New Records (New_Records) = Number of Records * New Record Percentage generated from Uniform Distribution using screen inputs.
- Let New Duplicates Records (New_Dup) = New_Records * Duplicate Data Entry Percentage.
- Let New Non Duplicates Records (New_NonDup) = New_Records – New_Dup.
- Let Updated Records (Updated_Records) = Number of Records * Updated Record Percentage generated from Uniform Distribution using screen inputs.
- Let Deleted Non Duplicate Records (Deleted_Records) = Number of Records * Deleted Record Percentage generated from Uniform Distribution using screen inputs. Restrict the number of Deleted Records to the number of existing and new Non Duplicated Deceased records (Dead_No + NonDup_Deceased).
- Let Duplicate Records Deleted (Dup_Deleted) = Record Number * Deleted Duplicate Record Percentage generated from Uniform Distribution using screen inputs. Restrict the number of Duplicate Records Deleted to number of existing and new Duplicate Records (Dup_No + New_Dup).
- If there are Duplicate Records (Dup_No > 0) then Let Number of Deleted Duplicate Records with Bad Addresses (Addr_Dup) = Dup_Deleted * (BadAddrDup_No / Dup_No). Otherwise let Addr_Dup be set to 0.
- Let New Non Duplicate Bad Addresses (New_BadAddr) = New_NonDup * Data Entry Percentage for Bad Addresses.
- Let New Duplicate Bad Addresses (NewDup_BadAddr) = New_Dup * Data Entry Percentage for Bad Addresses.
- If there are living, non duplicate customers left (NonDup_No – Dead_No > 0), using the number of newly deceased records, switch a proportion of living, non duplicate customers with bad addresses to the dead, non duplicate category. That is, Let New Non Duplicate Deceased with Bad Addresses

(BadAddr_Deceased) = NonDup_Deceased * (BadAddr_No / (NonDup_No – Dead_No)). Otherwise Let BadAddr_Deceased = 0.

- If there are living, duplicate customers left (Dup_No – DeadDup_No > 0), using the number of newly deceased duplicate records, switch a proportion of living, duplicate customers with bad addresses to the dead, duplicate category. That is, Let New Duplicate Deceased with Bad Addresses (BadAddrDup_Deceased) = Dup_Deceased * (BadAddrDup_No / (Dup_No – DeadDup_No)). Otherwise Let BadAddrDup_Deceased = 0.

B. Apply Changes to Mailing List Statistics

- Update Number of Duplicate Records using the formula: Dup_No = Dup_No + New_Dup – Dup_Deleted. Check and adjust for any anomalies like negative results.
- Update Number of Non-Duplicate Records using the formula: NonDup_No = NonDup_No + New_NonDup – Deleted_Records. Check and adjust for any anomalies like negative results.
- Let Number of Records = Dup_No + NonDup_No.
- If there are Duplicate Records then Update the Number of Duplicate Records where Customer is Dead using the formula: DeadDup_No = DeadDup_No + Dup_Deceased – Dup_Deleted * (DeadDup_No / Dup_No). Check and adjust for any anomalies like negative results.
- Update Number of Non Duplicate Records where Customer is Dead using the formula: Dead_No = Dead_No + NonDup_Deceased – Deleted_Records. Check and adjust for any anomalies like negative results.
- Update Number of Duplicate Records where Address is Bad using the formula: BadAddrDup_No = BadAddrDup_No + NewDup_BadAddr + Dup_NewMovers – BadAddrDup_Deceased – Addr_Dup. – Corrected, Duplicate Records. The Corrected Duplicate Records is based on the Number of Updated Records * (1 – Data Entry Percentage for Bad Addresses) * Proportion of Bad Addresses that are Duplicates. If the number of records updated exceeds the number of erroneous records, then the number of Duplicate Records where Address is bad is adjusted upwards by the number of extra Duplicates corrections * Data Entry Percentage for Bad Addresses. Check and adjust for any anomalies like negative results.
- Update Number of Non Duplicate Records where Address is Bad using the formula: BadAddr_No = BadAddr_No + New_BadAddr + NonDup_NewMovers – BadAddr_Deceased - Corrected, Non Duplicate Records. The Corrected Non Duplicate Records is based on the Number of Updated Records * (1 – Data Entry Percentage for Bad Addresses) * Proportion of Bad Addresses that are Non Duplicates. If the number of records updated exceeds the number of erroneous records, then the number of Non Duplicate Records where Address is bad is adjusted upwards by the number of extra Non Duplicates corrections * Data Entry Percentage for Bad Addresses. Check and adjust for any anomalies like negative results.

C. Check if it is time for a Mass Cleanup. If yes, then reduce the number of errors as follows:

- Multiply Dup_No, DeadDup_No, and BadAddrDup_No by (1 - Mass Cleanup Dup. %).
- Multiply Dead_No, DeadDup_No by (1 - Mass Cleanup Dead Percentage). Adjust Dup_No, NonDup_No, and Record_No downwards by the number of records removed.
- Multiply BadAddr_No, BadAddrDup_No by (1 – Mass Cleanup Address Percentage).

D. Store results in a data array for graphing 2 dimensional line charts. The arrays include:

- Number of Records
- Number of Duplicate and Non Duplicate Records with Bad Addresses
- Number of Duplicate and Non Duplicate Records where Customer is Dead
- Number of Duplicate Records