

Defining Test Data Using Population Analysis

Clarence Wm. Jackson, CQA - City of Dallas CIS

Abstract

Defining test data that provides complete test case coverage requires the tester to accumulate data that will satisfy all of the test cases in the requirements. Having access to production data eases the accumulation somewhat, but then any smart tester knows that testing should not be done in the production data. The next best situation is to completely copy production data, but this can be expensive depending on the size of the files and the amount of testing being committed. Another option would be to build test data based on the requirements, which may or may not match the varied data found in production data. This paper proposes using population analysis of production data as a tool to defining test data coverage. The tools available to provide statistical analysis, such as frequency of data element values and percentages to the total population will provide measurable verification of the test data to the production data, ensuring better results of testing scenarios.

Introduction

Test data is very important in ensuring that software performs as required. The testers using software specifications usually create test data, which usually means that the software will meet the specifications. This approach may not test the total requirements of the software or how the software will be used in production. If the tester has no knowledge of the data and how it's used, vital requirements can be missed in testing. Since the test data is an important link in the software testing cycle, ways to ensure that the test data is adequate should be explored. Population Analysis of production data will reduce the risk of missing critical data types, and increase the reliability of the software when moved to production.

Population Analysis

Population Analysis is simply creating reports that describe the type, frequency, and characteristics of data used by the application being tested. These reports will help in locating data and in defining the

ranges and limits of data elements, lengths of fields, and the number of data values within each data element. The benefits of using Population Analysis are many, but the major benefit is to the tester in the following ways:

- Identification of codes and values being used in production which were not indicated in the specification or requirement document
- Unusual data conditions, like special codes, trigger values
- Provides a model for use in creating test transactions and test cases
- Provides a model for the type and frequency of transactions that should be created
- Helps identify incorrect transactions for testing error processing
- Documentation for use later in the project life cycle

There are three types of population analysis that can be performed to gather identifying information for testing. All the information gathered from the analysis should be documented. The three types of population analysis are:

1. File/data stores - to identify files and other input/output used by the software being tested
2. Screen - to identify screen displays that will be used or generated by the software
3. Field/data element - to identify characteristics and frequencies of fields/data elements

Usually the file/data stores are known, and the screens are straightforward. Of the three types, the Field/data element analysis is the most complex and time-consuming part of population analysis. That's where SAS comes in. Population analysis is best performed using software, and SAS can deal with any type of data encountered in any shop. SAS is designed for population analysis, and can compute all the statistics needed. There are so many PROCedures that will give you the information that it is a natural tool for population analysis. Other software that will provide analysis information are Audit Software, DataBase utilities, and CASE tools. But for field/data element analysis, SAS is the choice.

The steps for population analysis as presented in this paper are based on the Deming/Shewhart PDCA Quality Circle for continuous process improvement, and the Quality Assurance Institute's (QAI) process "WORKBENCH" model. The steps are:

1. Define the inputs for analysis (Plan)

- Existing production files being used "as is"
- Existing production files for which there will be minor changes to the file
- Production files that contain the same or similar fields/data elements that will be included
- Existing manual files, such as paper invoices, forms, etc.
- Files from other systems that contain the same or similar data elements

2. Implement procedures (Do)

- Identify file/data stores location, medium, organization of data, and other details
- Identify the record types on the file/data stores, number of records, and sizes
- Document the file/data stores characteristics
- Identify each data element/field in the file/data stores
- Document the characteristics of each data element
- Analyze the population of data recorded in this data element area
- Identify and document abnormal conditions

3. Check procedures (Check)

4. Produce deliverables (Act)

Before beginning, define what information (deliverable) is required from the analysis, create forms or other documentation templates. The documentation requirements for the three types of population analysis should be established in the planning stage. The documentation will assist you in evaluating the completeness of your analysis, and allow you to measure your progress.

Implement Procedures

The major task is to identify the files that will be used for your analysis. Files and data stores must be located and defined. Flowcharts, data flow diagrams, system charts, and other documents can be used to identify files within the scope of the testing project.

Field/data element analysis is the most complex of the analysis of population data. The analysis of the data is the most important, and time spent in this analysis will have a direct payback in improving the test data. The objective of this type of analysis is to describe the type, frequency, and characteristics of the data elements. It is this analysis that will be used to produce the test data.

The analysis of the data should include the frequency, maximum, minimum, percentage of the population, and total values for each data element in each file/data store. This information should be documented.

Check Procedures

You should have mostly everything needed at this point, but it's a good time to review. Check your progress using your documentation template. Are you getting the information needed to perform a good test, and create good test data? Review abnormal conditions with those involved with the data for verification. All abnormal conditions should be tracked and resolved.

Produce Deliverables

The deliverables are the documentation which describe the population of the files, screens, and data elements. With the correct analysis, a question concerning the percentage of records having certain values is known and documented. The number of values within a given data element is known and documented. You now know what data values are required, and a much better set of tests can be performed. With this information, the test project should proceed with more assurance that the correct items will be tested. The documentation can be used for any of the following purposes:

- Supplement the requirements and specification documentation
- Development of the test plan
- Creation of test cases and scripts
- Creation of the test data
- Stress testing

How SAS Can Be Utilized To Perform Population Analysis

SAS can be used in many ways to provide for all types of population analysis reports through the many tools that are a part of the SAS System. SAS can be used for every type, or just for the more labor-intensive analysis. In some cases, part of the file/data stores, screen, and data element analysis may be known, and provided to the tester. However, the analysis of the data element values should still be performed, and using SAS to do it will save many hours of manual checklisting. Also, SAS can be used to store the results of each type of analysis and report it.

For data element analysis, SAS will be able to read the data, and using your favorite PROC, such as FREQ, MEANS, TABULATE, etc, you can produce a listing of the frequencies of any and all data elements that are of concern. SAS can also load the test data stores using the values from the analysis, with a little coding using the DATA step programming statements.

Screen analysis can also be done using SAS, especially if it is in some format that can be read directly. Screens are such that tools may only be required to store the data from analysis.

The files and data stores can be analyzed either manually or using SAS. On the mainframe, PROC SOURCE, PDS, and other methods of getting file information from the operating system will give you plenty of information about the files. If you are working with SAS data, PROC DATASETS, CONTENT, and other SAS PROCedures will provide everything from file information to data element metadata. For instance, PROC CONTENT will provide you with most of the identification information from SAS data files regarding each data element.

Conclusion

The use of population analysis for testing is a valuable tool in ensuring that software will function properly when moved to production. The best source of data for testing is production data. Why? It is data that is being used in the live process. To analyze the production data for the purpose of building test data is a sound method to verify software. Population analysis is a method to avoid the risk of missing obvious transaction types during the testing phase. The benefits of using the method are many.

SAS offers many tools that can be used to perform population analysis of production data. SAS can readily compute mean, median, mode, standard deviation, and other statistics by using a few BASE SAS PROCedures for field/data element population analysis.

References

Jackson, Clarence Wm., *Using SAS To Help Manage Data*
Proceedings of the Sixth Annual South Central Regional SAS Users Group Conference, 1996

Perry, William E., *A Structured Approach to Software Testing*,
Q.E.D Information Sciences, Inc., Wellesley, MA, 1983

Quality Assurance Inst., *QA/QC Solutions – Population Analysis: Creating Test Data From Production Data*, The Process Warehouse, Quality Assurance Inst (QAI), Orlando, FL, 1997

Beizer, Boris, *Software Testing and Quality Assurance*, Van Nostrand Reinhold, NY, NY, 1984

SAS and the SAS System are registered trademarks of SAS Inst, Inc, in the USA and other countries.

QAI and the Process Warehouse are registered trademarks of the Quality Assurance Inst, in the USA and in other countries.

The sample SAS program code used to support this paper is located in the Posters Area. The author can be contacted via email as follows:

Clarence Wm. Jackson, CQA
QA Change Manager
City of Dallas
Communication and Information Services
QA Change Management
1500 Marilla 4DS
Dallas, TX 75201

cljacks@ci.dallas.tx.us (City of Dallas)

CJac@compuserve.com (home)

<http://ourworld.compserve.com/homepages/CJac>

SAMPLE PROJECT

Test Project involves Salary and Jobcode changes for Company A

SAMPLE WORKSHEET #1 FILE/SUBSCHEMA POPULATION ANALYSIS

1. FILE/SUBSCHEMA IDENTIFIER:
2. FILE/SUBSCHEMA NAME:
3. RECORD TYPE(S)
4. IDENTIFIER FIELD
 - a. Yes No
 - b. NAME
5. VOLUME OF RECORDS (COUNT)
6. DESCRIPTION
7. VOLUME DENSITY OF RECORDS ON FILE:
 - a. Average: Maximum: Minimum:
8. FILE MEDIA
9. FILE ORGANIZATION
10. FILE OWNER
11. SECURITY CLASSIFICATION
12. DATE LAST CREATED/MODIFIED
13. DAYS SAVED
14. STORAGE LOCATION

SAMPLE WORKSHEET #2 FIELD/DATA ELEMENT POPULATION ANALYSIS

1. FIELD IDENTIFIER
2. FIELD NAME
3. FIELD TYPE
 - a. Numeric
 - b. Alphabetic
 - c. Alphanumeric
 - d. Special Symbols
 - e. Other (please specify)
4. FIELD DATA DICTIONARY DESCRIPTION ATTACHED?
Yes No
5. FIELD OWNER

APPLICABLE	FIELD POPULATION ANALYSIS
Yes No _____	1. VOLUME/FREQUENCY OF USE
Yes No _____	2. NUMBER ALL BLANKS IN FIELD
Yes No _____	3. NUMBER ALL ZEROS IN FIELD
Yes No _____	4. HIGH VALUE
Yes No _____	5. LOW VALUE
Yes No _____	6. MEAN VALUE
Yes No _____	7. MEDIAN VALUE
Yes No _____	8. MODE VALUE
Yes No _____	9. STANDARD DEVIATION
Yes No _____	10. CODES FREQUENCY
Yes No _____	11. MAXIMUM NUMBER NONBLANK ZEROS FOUND IN FIELD
Yes No _____	12. MINIMUM NUMBER NONBLANK ZEROS FOUND IN FIELD

Data to complete the worksheets are in **BOLD** in the following SAS log and output.

SAS LOG

NOTE: Copyright (c) 1989-1996 by SAS Institute Inc., Cary, NC, USA.

NOTE: SAS (r) Proprietary Software Release 6.12 TS020

Licensed to CITY OF DALLAS, Site 0001457002.

NOTE: AUTOEXEC processing beginning; file is C:\SAS\AUTOEXEC.SAS.

NOTE: AUTOEXEC processing completed.

```
1
2  * Sample program to get basic Population
  Analysis data.
3  The data used is the SAS supplied sample
  data "SALARY".
4
5  The Questions for analysis relates to
6      (1) retention rates for employees
7      (2) pay rates
8      a. by job type
9      b. by rate groups
10
11  Clarence Wm. Jackson, CQA
12  ;
13
14  title1 "SSU 2001, A Joint Conference of SCSUG
  and SESUG";
15  title2 "Sample Set of Reports to Support
  Population Analysis";
16
17  proc format;
18  value salrange
19      0 - 25000      = "$25,000 or Less  "
20      25001 - 50000  = "$25,001 - 50,000  "
21      50001 - 80000  = "$50,001 - 80,000  "
22      80001 - 110000 = "$80,001 - 110,000 "
23      110001 - 200000 = "$110,001 - 200,000"
24      200001 - 250000 = "$200,001 - 250,000"
25      250001 - HIGH  = "$250,001 Up    "
26      other          = "Not Paid - Error  ";
NOTE: Format SALRANGE has been output.
27  run;
NOTE: The PROCEDURE FORMAT used 0.44 seconds.
28
29  title4 "The SAS LOG Will Provides Information
  About The File";
30
31  libname samples "C:\SAS\CORE\SAMPLE\";
NOTE: Libref SAMPLES was successfully assigned as
  follows:
```

Engine: V612
Physical Name: C:\SAS\CORE\SAMPLE

```
32
33  title4 "The PROC CONTENTS Provides Information
  About The Data Set";
34
35  proc contents data=samples.salary;
36  run;
NOTE: The PROCEDURE CONTENTS used 1.32 seconds.
```

```

37
38 title4 "The PROC MEANS Provides Information
About Variables In Question";
39 title5 "Output data used in PROC FORMAT for
RANGES in SALARY variable";
40
41 proc means data=samples.salary MAXDEC=2;
42   var salary;
43
44   output out = salnums;
45 run;
NOTE: The data set WORK.SALNUMS has 5 observations
and 4 variables.
NOTE: The PROCEDURE MEANS used 0.82 seconds.
46
47 data sumrec (keep= maxsal minsal smean stdev
ucl lcl);
48   set _last_ end=alldone;
49   if _STAT_='MAX' then do;
50       maxsal=salary;
51       retain maxsal;
52   end;
53   if _STAT_='MIN' then do;
54       minsal=salary;
55       retain minsal;
56   end;
57   if _STAT_='MEAN' then do;
58       smean=salary;
59       retain smean;
60   end;
61   if _STAT_='STD' then do;
62       stdev=salary;
63       retain stdev;
64   end;
65   if alldone then do;
66       ucl=stdev+smean;
67       lcl=stdev-smean;
68       output;
69   end;
70   else do;
71       delete;
72       return;
73   end;
74 run;
NOTE: The data set WORK.SUMREC has 1 observations
and 6 variables.
NOTE: The DATA statement used 1.47 seconds.
75
76 proc print;
77 run;
NOTE: The PROCEDURE PRINT used 0.22 seconds.
78
79 title4 "At This Point, Some Reordering is
Needed on Variables to Make Sense";
80
81 data popanly;
82   if not allsum then do;
83       set sumrec end=allsum;
84       retain maxsal minsal smean stdev ucl lcl;
85       delete;

```

```

86       return;
87   end;
88   else do;
89       set samples.SALARY;
90       if salary gt ucl then pay="Above UCL ";
91       if salary lt lcl then pay="Below LCL ";
92       if salary gt smean and salary lt ucl then
pay="Above Avg ";
93       if salary lt smean and salary gt lcl then
pay="Below Avg ";
94       if enddate=. then current="Yes";
95       else current="No ";
96       jobtype=substr(jobcode,1,3);
97       if jobcode=" " then put _all_;
98   end;
99 run;
NOTE: The data set WORK.POPANLY has 319
observations and 14 variables.
NOTE: The DATA statement used 0.44 seconds.
100
101 proc sort;
102   by salary;
103 run;
NOTE: The data set WORK.POPANLY has 319
observations and 14 variables.
NOTE: The PROCEDURE SORT used 0.28 seconds.
104
105 proc freq data=_last_;
106   format salary salrange.;
107   tables jobcode;
108   tables jobtype;
109   tables salary;
110   tables current;
111   tables pay;
112   tables jobtype*salary;
113   tables jobtype*pay;
114 run;
NOTE: For table location in print file, see
page 4 for JOBCODE
page 9 for JOBTYP
page 9 for SALARY
page 10 for CURRENT
page 10 for PAY
page 11 for JOBTYP*SALARY
page 16 for JOBTYP*PAY
NOTE: The PROCEDURE FREQ used 0.7 seconds.

```

CONTENTS PROCEDURE OUTPUT

Data Set Name: SAMPLES.SALARY

Observations: 319

Member Type: DATA

Variables: 5

Engine: V612

Indexes: 0

Created: 17:08 Thursday, April 25, 1996

Observation Length: 40

Last Modified: 17:08 Thursday, April 25, 1996

Deleted Observations: 0

Protection:

Compressed: NO

Data Set Type:

Sorted: NO

Label:

-----Engine/Host Dependent Information-----

Data Set Page Size: 8192

Number of Data Set Pages: 2

File Format: 607

First Data Page: 1

Max Obs per Page: 203

Obs in First Data Page: 180

----Alphabetic List of Variables and Attributes----

#	Variable	Type	Len	Pos	Format	Informat	Label
3	BEGDATE	Num	8	16	DATE7.	DATE7.	
4	ENDDATE	Num	8	24	DATE7.	DATE7.	
1	IDNUM	Num	8	0	SSN11.	F11.	Identification Num
5	JOBCODE	Char	8	32			
2	SALARY	Num	8	8	DOLLAR12.	DOLLAR12.	Salary

MEANS PROCEDURE OUTPUT

Analysis Variable : SALARY Salary

N	Mean	Std Dev	Minimum	Maximum
319	46936.83	39929.42	12000.00	500000.00

PRINT PROCEDURE OUTPUT

OBS	MAXSAL	MINSAL	SMEAN	STDEV	UCL	LCL
1	500000	12000	46936.83	39929.42	86866.25	-7007.42

FREQUENCY PROCEDURE OUTPUT

CURRENT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	10	3.1	10	3.1
Yes	309	96.9	319	100.0

JOBCODE	Frequency	Percent	Cum Frequency	Cum Percent
SIS004	1	0.3	1	0.3
ACT001	1	0.3	2	0.6
APP001	1	0.3	3	0.9
APP002	6	1.9	9	2.8
.....(stuff in the middle deleted).....				
VID002	2	0.6	318	99.7
VID003	1	0.3	319	100.0

SALARY	Frequency	Percent	Cum Freq	Cum Percent
\$25,000 or Less	53	16.6	53	16.6
\$25,001 - 50,000	179	56.1	232	72.7
\$50,001 - 80,000	64	20.1	296	92.8
\$80,001 - 110,000	7	2.2	303	95.0
\$110,001 - 200,000	12	3.8	315	98.7
\$200,001 - 250,000	3	0.9	318	99.7
\$250,001 Up	1	0.3	319	100.0