

STUDY SERIES
(*Statistics #2002-01*)

**The GenBounds Software for Generating a Complete
Set of Ratio Edits: %Implied User's Guide**

Roger Goodwin, Maria Garcia, and
Katherine Thompson

Statistical Research Division
U.S. Bureau of the Census
Washington D.C. 20233

Report Issued: November 18, 2002

Disclaimer: This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a Census Bureau review more limited in scope than that given to official Census Bureau publications. This paper is released to inform interested parties of ongoing research and to encourage discussion of work in progress.

The GenBounds Software for Generating a Complete Set of Ratio Edits: %Implied User's Guide

Roger Goodwin, Maria Garcia, and Katherine Thompson

Abstract

We present documentation for running the GenBounds software, new SAS programs for generating the implicit edits from a given set of explicit ratio edits. We have developed two separate programs: one written in SAS/IML and one written in SAS/OR. The first program is designed for production processing and the second is designed for edit development. Both programs are submitted using the same macro driver routine, %Implied, and are available from ESMPD and SRD.

Keywords: **explicit edits, implicit edits, shortest path**

Table of Contents

1. Introduction	1
2. Input file (SAS Dataset).....	2
3. Program Overview	3
4. How to Run %Implied as a Standalone Program.....	4
5. Interpreting the Output	5
5.1 GenBndsIML	5
5.2 GenBndsOR (NETFLOW).....	6
5.3 Final Print Out.....	7
5.4 Changed Edits Report.....	8
6. References	8

1. Introduction

Key data items collected by the Economic Census are subjected to ratio edits as part of the overall review process. A ratio edit compares the ratio of two highly correlated items to upper and lower bounds. Reported items that fall outside of the tolerances are considered edit failures, and one or both of the items in an edit-failing ratio are either imputed or flagged for analyst's review. The Economic Census' Plain Vanilla (PV) Ratio module utilizes the Fellegi-Holt model of editing (Fellegi-Holt, 1976). The program determines the minimum number of reported data fields that need to be changed to satisfy the complete set of edits (Greenberg, 1986). The complete set of edits is defined as the user-specified edits provided in the script (the explicit edits), plus the other ratio tests **implied** by the explicit set. The Fellegi-Holt model requires the complete set of explicit and implicit edits. [Note: any pair of ratio edits with a common data item implies

another ratio edit]. This methodology has been used successfully at the Census Bureau by other economic programs since the early 1980s (Greenberg *et al*, 1990).

The GenBnds Edit Generation software generates the needed implicit ratio edits for a given (explicit) set of ratio edits using a shortest path algorithm (Fagan (1999), Garcia and Goodwin (2002)). Starting with the user-provided set of explicit edits, the programs check the logical consistency of the explicit edits, determine if there are any specified redundancies, and generate the implicit ratio edits.

This User's Guide describes two programs: GenBndsOR.sas and GenBndsIML.sas. GenBndsOR.sas (NETFLOW) is designed for **interactive** implied edit review and evaluation. The core algorithm uses SAS/OR. GenBndsOR.sas (NETFLOW) offers several useful printed analytical outputs, but is not designed for processing large datasets. GendbndsIML.sas is designed for **batch (or large scale)** processing (generally performed after the more exact micro-review is completed). The core algorithm uses SAS/IML. Although this program can process large data sets far more quickly than NETFLOW, its analytical outputs are limited. Both programs are executed using the same macro call (%Implied). Both programs have been tested in Windows 2000 and UNIX environments using SAS, v8.2.

2. Input file (SAS Dataset)

The GenBnds software requires a SAS data set (version 8.2) that contains the following variables:

- a) Column **CLASS** contains the classification variable code¹.
- b) Column **E_RATIO** contains the explicit ratio test (i.e. the mnemonic name of the ratio).
- c) Column **L_FENCE** contains the explicit ratio edit's lower bound.
- d) Column **U_FENCE** contains the explicit ratio edit's upper bound.

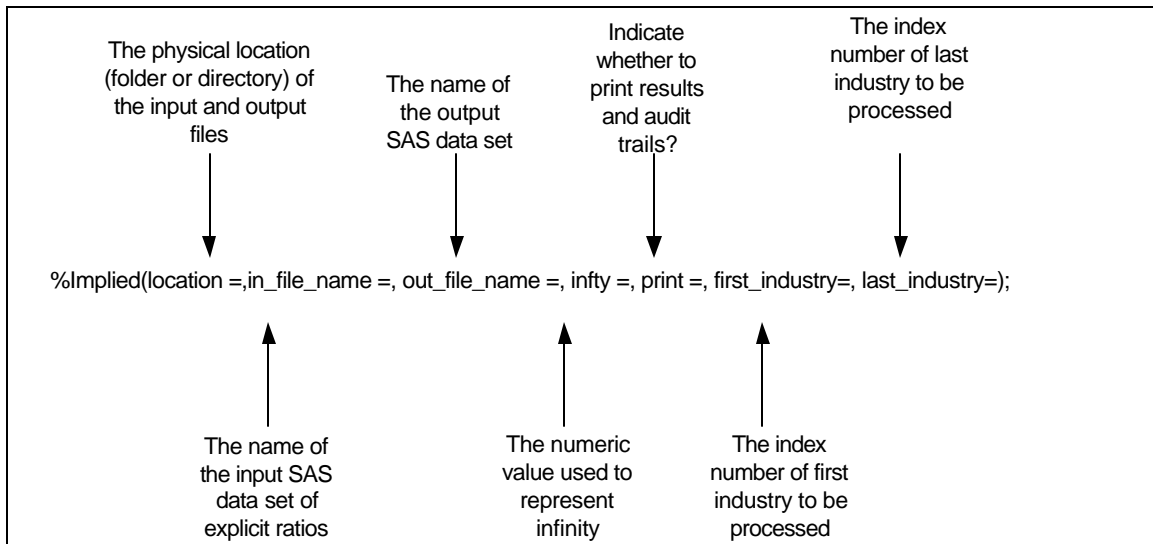
[Note: This SAS data set can be created using RESBND_I.sas or RESBND_B.sas (Thompson and Sigman (1996)).

¹ The user can specify any character variable as a classification variable. Since the Economic Census' edits are industry based, we use the terms "industry" and "classification variable" interchangeably. This value can also be blank (missing for all items).

3. Program Overview

GenBndsOR.sas and GenBndsIML.sas have the same keyword macro parameters (see Figure 1): `location`, `in_file_name`, `out_file_name`, `infty`, `print`, `first_industry` and `last_industry`.

Figure 1: Keyword Parameters



where

- **LOCATION (required):** Specifies the directory (UNIX) or folder (PC) of the input file. The location must be in quotes. For example, `LOCATION = "c:\2002rnd\"`.
- **IN_FILE_NAME (required):** Specifies the (unquoted) member name of the SAS data set that contains the explicit edits.
- **OUT_FILE_NAME (not required):** Specifies the (unquoted) member name of the SAS data set that contains the complete set of edits (explicit edits and implied edits). The default is `OUT_FILE_NAME = ALL_EDITS`.
- **INFTY (not required):** Contains the value of infinity used by `%Implied`. The default is `INFTY = 107` which is also the default value used in the PV RATIO module.
- **PRINT (not required):** Specifies whether to send any output produced by `%Implied` to the SAS Output Window (YES/NO). The default is `PRINT = YES`.
- **FIRST_INDUSTRY (not required):** Contains a sequential number for each unique classification variable value in the input file (i.e. 1 for the first industry, 2


for the second industry, 3 for the third industry, etc). If the user wishes to process a sequentially ordered subset of “industries,” then use the FIRST_INDUSTRY and LAST_INDUSTRY values (e.g. FIRST_INDUSTRY = 1 and LAST_INDUSTRY = 10 will produce implied edits for the first 10 industries in the input file). The default is FIRST_INDUSTRY = 1.

- **LAST_INDUSTRY (not required):** Used in conjunction with FIRST_INDUSTRY to process a subset of industry codes (see above). The default is LAST_INDUSTRY = 1,162.

4. How to Run %Implied as a Standalone Program

Both programs have been tested in a UNIX environment and on a PC with Windows 2000. To run either program interactively, include the desired program into the Program Manager window. At the bottom of the program(s), type a call to %Implied similar to:

```
%Implied(location='/2002rnd/',  
         in_file_name= spv_eratio);
```

using your own data directory and file name. To execute, click on run icon  from the SAS Program Manager window.

To run either program in batch mode, modify the appropriate program file (GenBndsIML.sas or GenbndsOR.sas) with the %Implied call as described above, then use the appropriate operating system level command to submit the file to a batch queue. For instance, to execute the GenBndsIML code in a UNIX operating system, type:

```
$nohup sas82 GenBndsIML.sas &
```

SAS will create an output file, GenBndsIML.lst, that contains the output ordinarily sent to the SAS Output window. To execute the GenBndsOR.sas program in batch mode, type:

```
$nohup sas82 GenBndsOR.sas &
```

SAS will create an output file, GenBndsOR.lst. Note that UNIX commands, paths, and file names are case sensitive.

5. Interpreting the Output

This section describes the available analytic outputs (printouts) from both programs using sample data from two industries: XXXX1 and XXXX3.

5.1 GenBndsIML Outputs

Figure 2 shows the explicit edits and the implied edits for industries XXXX1 and XXXX3. GenBndsIML prints each complete set of ratio edits separately by classification variable (industry). These edits are printed in *alphabetical order*, and some may appear as the reciprocal of the given explicit edits.

Figure 2: Partial IML Output

Explicit Edits and Implied Edits for Industry XXXX1					09:34
Obs	Industry Code	Lower Bound	Mnemonic Name	Upper Bound	
1	XXXX1	1.9011972	REP_APR / REP_EMP	200.9521091	
2	XXXX1	2.5627607	REP_APR / REP_QPR	7.0724513	
3	XXXX1	0.0298806	REP_APR / REP_SLS	2.7109327	
4	XXXX1	0.0127531	REP_EMP / REP_QPR	3.7199988	
5	XXXX1	0.0001487	REP_EMP / REP_SLS	1.4259082	
6	XXXX1	0.0042249	REP_QPR / REP_SLS	1.0578174	

Explicit Edits and Implied Edits for Industry XXXX3					09:34
Obs	Industry Code	Lower Bound	Mnemonic Name	Upper Bound	
1	XXXX3	3.1058753	REP_APR / REP_EMP	26.8782981	
2	XXXX3	3.1057553	REP_APR / REP_QPR	5.7816754	
3	XXXX3	0.1195074	REP_APR / REP_SLS	0.7329608	
4	XXXX3	0.1155488	REP_EMP / REP_QPR	1.8615285	
5	XXXX3	0.0044462	REP_EMP / REP_SLS	0.2359917	
6	XXXX3	0.0206700	REP_QPR / REP_SLS	0.2360008	

Figure 3 displays a sample of the audit trail output produced by GenBndsIML (the output window has been edited to fit the screen capture).

Figure 3: GenBndsIML Audit Trail

TRAIL								
v_i	v_k	u_ik	v_k	v_j	u_kj	v_i	v_j	u_ij
REP_EMP / REP_APR	0.5259843586	REP_APR / REP_QPR	7.0724513251	REP_EMP / REP_QPR	3.7199987743			
REP_EMP / REP_APR	0.5259843586	REP_APR / REP_SLS	2.7109327401	REP_EMP / REP_SLS	1.4259082186			
REP_QPR / REP_APR	0.3902042111	REP_APR / REP_EMP	200.9521091000	REP_QPR / REP_EMP	78.4123591930			
REP_QPR / REP_APR	0.3902042111	REP_APR / REP_SLS	2.7109327401	REP_QPR / REP_SLS	1.0578173711			
REP_SLS / REP_APR	33.4665717880	REP_APR / REP_EMP	200.9521091000	REP_SLS / REP_EMP	6725.1781851000			
REP_SLS / REP_APR	33.4665717880	REP_APR / REP_QPR	7.0724513251	REP_SLS / REP_QPR	236.6906999900			

This audit trail output lists the edits bounds as they are updated. The column headings v_i, v_j, and v_k, are used to represent the data fields and the column headings U_ik, U_kj, and U_ij are used to represent the ratio edits upper bounds. The first row of the audit trail output shows that the upper bound for the ratio REP_EMP/REP_QPR (3.7199987743) was obtained using the upper bounds for the ratios REP_EMP/REP_APR and REP_APR/REP_QPR, that is $0.525984356 \times 7.0724513251 = 3.7199987743$.

5.2 GenBndsOR (NETFLOW)

Figure 4 presents a portion of the standard GenBndsOR analytical output for one industry.

Figure 4: Partial NETFLOW Output for Industry XXXX1

Explicit and Implied Edits for Industry XXXX1					
Obs	Lower Bound	Mnemonic Name	Upper Bound		
1	1.9011972	REP_APR / REP_EMP	200.9521091		
2	2.5627607	REP_APR / REP_QPR	7.0724513		
3	0.0298806	REP_APR / REP_SLS	2.7109327		
4	0.0127531	REP_EMP / REP_QPR	3.7199988		
5	0.0001487	REP_EMP / REP_SLS	1.4259082		
6	0.0042249	REP_QPR / REP_SLS	1.0578174		
Audit Trail for Industry XXXX1					
Lower Bound	Mnemonic Name	Upper Bound	Lower Path Bounds	Edits Traversed	Upper Path Bounds
1.9011972	REP_APR / REP_EMP	200.9521091	1.9011972	REP_APR / REP_EMP	200.9521091
2.5627607	REP_APR / REP_QPR	7.0724513	2.5627607	REP_APR / REP_QPR	7.0724513
0.0298806	REP_APR / REP_SLS	2.7109327	0.0298806	REP_APR / REP_SLS	2.7109327
0.0127531	REP_EMP / REP_QPR	3.7199988	2.5627607 0.0049763	REP_APR / REP_QPR REP_EMP / REP_APR	7.0724513 0.5259844
0.0001487	REP_EMP / REP_SLS	1.4259082	0.0298806 0.0049763	REP_APR / REP_SLS REP_EMP / REP_APR	2.7109327 0.5259844
0.0042249	REP_QPR / REP_SLS	1.0578174	0.0298806 0.1413937	REP_APR / REP_SLS REP_QPR / REP_APR	2.7109327 0.3902042

GenBndsOR produces two printouts for each industry: a listing of the complete set of edits (explicit and implied edits) and an audit trail specifying the derivation of each implied edit. For the sample industry displayed in Figure 4, the lower bound of implied edit REP_EMP/REP_QPR is obtained by multiplying the lower bounds from the two edits REP_APR/REP_QPR and REP_EMP/REP_APR ($2.5627607 \times 0.0049763 = 0.0127531$). Similarly, the upper bound of this implied edit is the product of the upper bounds for the same two explicit edits (REP_APR/REP_QPR and REP_EMP/REP_APR, $7.0724513 \times 0.5259844 = 3.7199988$). This audit trail is extremely useful for edit development, because the user can see (at a glance) which edits contributed to the lower and/or upper bounds of an implied edit.

NOTE: In some instances the Lower Path Bounds or the Upper Path Bounds variable is represented as a missing value (denoted by “.”) in the audit trail. In such cases, an edit contributes to either the upper bound or lower bound of an implied edit, but

not both. Figure 5 shows such instances. For the implied edit REP_QPR/REP_SLS, the edit REP_APR/REP_EMP contributes to the lower bound but does not contribute to the upper bound. The edit REP_APR/REP_SLS contributes to the upper bound but not the lower bound.

Figure 5: Edits Contributing to One Bound and Not the Other

Explicit and Implied Edits for Industry XXXX1					
Obs	Lower Bound	Mnemonic Name		Upper Bound	
1	1.9011972	REP_APR / REP_EMP		18.9765292	
2	2.5627607	REP_APR / REP_QPR		7.0724513	
3	0.2715996	REP_APR / REP_SLS		2.7109327	
4	0.1350490	REP_EMP / REP_QPR		3.7199988	
5	0.1428571	REP_EMP / REP_SLS		1.0000000	
6	0.0384025	REP_QPR / REP_SLS		1.0578174	

Audit Trail for Industry XXXX1					
Lower Bound	Mnemonic Name	Upper Bound	Lower Path Bounds	Edits Traversed	Upper Path Bounds
1.9011972	REP_APR / REP_EMP	18.9765292	1.9011972	REP_APR / REP_EMP	.
			.	REP_APR / REP_SLS	2.7109327
			.	REP_SLS / REP_EMP	7.0000000
2.5627607	REP_APR / REP_QPR	7.0724513	2.5627607	REP_APR / REP_QPR	7.0724513
0.2715996	REP_APR / REP_SLS	2.7109327	1.9011972	REP_APR / REP_EMP	.
			.	REP_APR / REP_SLS	2.7109327
			0.1428571	REP_EMP / REP_SLS	.
0.1350490	REP_EMP / REP_QPR	3.7199988	2.5627607	REP_APR / REP_QPR	7.0724513
			.	REP_EMP / REP_APR	0.5259844
			0.1428571	REP_EMP / REP_SLS	.
			0.3688767	REP_SLS / REP_APR	.
0.1428571	REP_EMP / REP_SLS	1.0000000	0.1428571	REP_EMP / REP_SLS	1.0000000
0.0384025	REP_QPR / REP_SLS	1.0578174	1.9011972	REP_APR / REP_EMP	.
			.	REP_APR / REP_SLS	2.7109327
			0.1428571	REP_EMP / REP_SLS	.
			0.1413937	REP_QPR / REP_APR	0.3902042

5.3 Final Print Out

Both programs produce the complete set of edits titled “Implied Edits and Explicit Edits for All Industries” which contains the complete set of edits for every industry. Again, notice that the edits are printed in *alphabetical order* and some may appear as the reciprocal of specified explicit edits.

Figure 6: Final Output for Both Programs

Implied Edits and Explicit Edits for All Industries				
Obs	Industry Code	Lower Bound	Mnemonic Name	Upper Bound
1	XXXX1	1.9011972	REP_APR / REP_EMP	200.9521091
2	XXXX1	2.5627607	REP_APR / REP_QPR	7.0724513
3	XXXX1	0.0298806	REP_APR / REP_SLS	2.7109327
4	XXXX1	0.0127531	REP_EMP / REP_QPR	3.7199988
5	XXXX1	0.0001487	REP_EMP / REP_SLS	1.4259082
6	XXXX1	0.0042249	REP_QPR / REP_SLS	1.0578174
7	XXXX3	3.1058753	REP_APR / REP_EMP	26.8782981
8	XXXX3	3.1057553	REP_APR / REP_QPR	5.7816754
9	XXXX3	0.1195074	REP_APR / REP_SLS	0.7329608
10	XXXX3	0.1155488	REP_EMP / REP_QPR	1.8615285
11	XXXX3	0.0044462	REP_EMP / REP_SLS	0.2359917
12	XXXX3	0.0206700	REP_QPR / REP_SLS	0.2360008

In addition to the calculations sent to the SAS Output Window, the information in Figure 6 is stored in a permanent SAS dataset with the member name specified in the keyword parameter &out_file_name.

Both programs create an **entirely new** data set each time that they are submitted. The programs do not append or update existing data sets. Users who are processing portions of an input file should be careful to change the output file name on each run.

Note: GenBndsIML produces implied ratio edits for *all* possible ratio-edit-variable pairs in the explicit edit file. In the case where there is no implied relationship between a pair of variables, GenBndsIML will create a record with bounds of (0, ∞). GenBndsOR does not create records in the implied edit file in these cases. So the complete set of edits obtained using GenBndsIML may be larger than the corresponding file obtained using GenBndsOR for the same input explicit edit file. Because the PV Ratio module requires that all industries processed with the same input file contain the same ratio edit pairs, it is safest to use GenBndsIML to produce the production-ready file. Of course, it is easier to review the smaller (and more relevant) set of edits produced by GenBndsOR during development.

5.4 Changed Edits Report

An implied edit can change an explicit edit. When this happens, both programs print a Changed Explicit Edits Report (See Figure 7). The report contains the explicit edits whose bounds were modified along with their corresponding reciprocals. For example, the original upper bound of 200.952109100 for REP_APR/REP_EMP was revised to 18.976529181 after generating all implied edits. The lower bound of REP_SLS/REP_APR was revised from 0.029880563 to 0.271599603.

Figure 7: Changed Explicit Edits Report

Changed Explicit Edits Report								
Obs	Industry Code	Original Lower Bound	Original Upper Bound	Mnemonic Name	Revised Lower Bound	Revised Upper Bound		
1	XXXX1	1.901197219	200.952109100	REP_APR / REP_EMP	1.901197219	18.976529181		
3	XXXX1	0.029880563	2.710932740	REP_APR / REP_SLS	0.271599603	2.710932740		
4	XXXX1	0.004976310	0.525984359	REP_EMP / REP_APR	0.052696675	0.525984359		
7	XXXX1	0.368876728	33.466571788	REP_SLS / REP_APR	0.368876728	3.681890511		

6. References

1. Brassard, G. and Bratley, P., *Algorithmics: Theory and Practice*, Prentice Hall, NJ, 1988.
2. Fagan, J., "Generating Implied Ratio Edits", unpublished Census manuscript. October 1999.
3. Fellegi, I. P. and Holt, D., "A Systematic Approach to Automatic Edit and Imputation," *The Journal of the American Statistical Association*, No. 71, 1976.

4. Garcia, Maria and Goodwin, Roger, "Developing SAS Software for Generating a Complete Set of Ratio Edits," SRD Research Report RR-Statistics #2002-06.
5. Greenberg, B. and Petkunas, T., "SPEER (Structured Program For Economic Editing and Referrals)", *Proceedings of the Section on Survey Research Methods*, ASA, 1990.
6. Hillier, Frederick S. and Lieberman, Gerald J, *Introduction to Operations Research 5th Edition*, McGraw-Hill Publishing Co, NY, NY, 1990.
7. SAS Institute Inc., *SAS/OR User's Guide: Mathematical Programming, Version 8*, Cary, NC: SAS Institute Inc., 1999. 566pp.
8. Thompson, K. J. and Sigman, R., "Statistical Methods for Developing Ratio Edit Tolerances for Economic Data", *J. Official Statistics*, 2000.
9. Winkler, W. and Draper, L., "The SPEER Edit System", *Proceedings of the Conference of European Statisticians, Section on Statistical Data Editing, UNECE, 1997*.