

Using SAS to Auto-code Occupation and Industry Fields

Gail Wisan, Westat, Inc., Rockville, MD

ABSTRACT

In survey research, a typical question asked of respondents concerns their occupation and industry. Traditionally, coding has been approached as an effort performed one at a time: each respondent's occupation and industry coded separately. While at first glance this may appear to be the best approach to quality control, it can be argued that in a world of limited resources the quality approach to coding will involve automating the coding process. This paper discusses how SAS can and has been used to help automate the process of coding. Several approaches to auto-coding will be discussed including: use of FIRST.byvariable; exact matching to standard occupational and industrial databases; exact matching to earlier cycles of a survey's occupational and industrial databases; matching to another survey's (same coding system) occupational and industrial databases; use of the UPCASE function to facilitate matching; use of the SUBSTR function to match to a core part of an occupation or industry word; use of SAS MERGE to get information from several existing databases and thus enlarge the extant coding database; use of SAS MERGE for quality control of coding and use of the INDEX function for examining different strings for the same occupation at the same time.

INTRODUCTION

In survey research, demographic and socioeconomic questions are standard items which provide information for analysis regardless of a survey's substantive focus. Questions frequently included ask respondents about their occupation and industry or, for younger respondents, their parents' occupations and industries. Typically these questions have been structured as open-ended questions. Consequently, responses required coding into categories or codes such as those provided by the Standard Occupational Classification Manual, the Standard Industrial Classification Manual, the Census, or some abbreviated coding system based upon one of the detailed standard coding systems. Traditionally codes have been attached to these occupations and industries by individual coders, working one by one with a single respondent's questionnaire data. Auto-coding with SAS offers an approach which can streamline the person-intensive coding process. What is meant by the term auto-coding? The term combines coding (here referring to attaching symbols according to a specified classification system, occupational and industrial in this case) and auto. Auto is a prefix which, in one of its two basic meanings, indicates an automatic or self-operating system which does not require continual manual adjustment or intervention. The SAS system with

its matching capabilities and string manipulation functions offers an opportunity to at least partially automate the occupational and industrial coding process.

PROC FREQ: AN OVERVIEW

You should first begin by running PROC FREQ on your occupational and industrial responses or, to switch to SAS data processing terminology, character strings. By examining the frequency or percentages of your survey's occupational and industrial responses, you can get an overview of the distribution of your occupational and industrial strings and will begin to be able to assess how much help auto-coding can provide to your study. The larger the concentration of responses, the bigger the frequencies and the percentages associated with certain strings, and the greater the return you can expect from auto-coding. For example, if the occupation "teacher" appears five hundred times in your database of occupations, it is preferable to code it one time with a coder and then merge it onto each duplicate string in your occupational strings database rather than use a set of coders to code each occurrence of the string "teacher" five hundred times. There are a couple of strategies for using SAS to produce an output file for coders to edit. One approach would be to produce an output file within PROC FREQ. The following SAS code will produce an output file which could be used by coders.

```
PROC FREQ;
  TABLES occup / OUT= io.freqoc;
  TITLE 'Example of Output from Freq';
  RUN;
```

```
DATA _NULL_;
  SET io.freqoc;
  FILE file-specification;
  PUT
    @1 occup      $16.;
  RUN;
```

The disadvantage of this approach is that the longer occupational and industrial strings would be truncated to sixteen characters by PROC FREQ which would create problems for matching longer strings.

HELPFUL SAS DATA STEPS

Rather than creating an output file with PROC FREQ, a better approach is to use several SAS data steps to

subset the occupational and industrial database files and then create a file with only one of each occupational or industrial code which coders would then use for coding occupation and industry. First you should sort the full file of occupational strings by occupation, variable "occup" in this example. The sorted occupational file can then be processed in occupational BY groups.

FIRST.byvariable

When a BY statement is used, SAS automatically keeps track of which observation is the first and last in the BY group. This FIRST.occup variable can be used to subset the occupational string file to the first occurrence of an occupational string. The following code can be used to remove duplicate strings from the full occupational string file:

```
PROC SORT DATA=allstrng;
  BY occup;
  Run;
```

```
DATA Subocc;
  SET allstrng;
  BY occup;
  IF FIRST.occup;
  RUN;
```

The output file Subocc will contain only unique occupational string so that coders can assign a code to "practical nurse" one time rather than, for example, the eighty seven times that it appears in the occupational strings file.

Match-Merging

Once the subset occupational file has had codes entered, SAS can be used to match-merge the subset file with codes back to the original file containing all observations (i.e., all persons with their occupational and industrial string variables). The following SAS code can be used to merge the occupational codes (e.g., SOC or CENSUS codes) onto the full file with all records and duplicate occupational and/or industrial strings. You will use the occupational string as the variable by which to merge the two files: the coding file composed of unique strings (and codes at the end of the coding process) and the full file with duplicate strings.

```
DATA Mergcode ;
  MERGE Allstrng
        Subocc;
  BY occup;
  RUN;
```

File Allstrng

Obs	Id	Occup
1	11	beautician
2	22	plumber
3	05	plumber
4	01	practical nurse
5	14	practical nurse
6	02	practical nurse
7	45	practical nurse
8	32	principal

File Subocc

Obs	Occup	Code
1	plumber	645
2	practical nurse	366
3	principal	1282

File Mergcode

Obs	ID	Occup	Code
1	11	beautician	
2	22	plumber	645
3	05	plumber	645
4	01	practical nurse	366
5	14	practical nurse	366
6	02	practical nurse	366
7	45	practical nurse	366
8	32	principal	1282

It is much more efficient in terms of both time and money to code an occupational string one time and then merge the code onto to all other duplicates of that occupational or industrial string on the full occupations and industry file rather than have a coder attach a code separately to fifteen, fifty, or five hundred records. Note the output file created for manual code entry could be, among other choices, an ASCII file created in a SAS data step or a DBF file.

Creating an Output File for Coders

When creating the subset file, you will only keep the relevant string (occupation or industry). No record/observation identification numbers (i.e., IDs) will be kept for two reasons. First, since you are not keeping most observations or records in this subset file, the IDs would be confusing. Secondly, the IDs are not needed for merging because merging will be performed by occupational or industrial string, not by ID. An ASCII file is a useful output file that can be utilized directly in an editing system or loaded into some other software such as dBase for coding and data entry. The following SAS code can be used to create an ASCII file for use by coders.

FILENAME out 'file-specification';

DATA Flat;

```
FILE out;  
SET Subocc;  
IF Occup NE '' THEN PUT @1 Occup $25.;  
RUN;
```

Coders could then enter the correct occupational code for the listed occupation string .

Match-Merge Data Sources Prior to Manual Coding

In the above example, we are using just one data source file of strings for occupations and one for strings of industries. A subset coding file is created from the original file with the process involving merging the codes back with the source file via a merge by the string variable. Match merging can additionally be used prior to any manual coding to reduce the time and money spent on coding.

Electronic versions of coding systems can be used if your study's coding scheme is using the same level of detail as the full coding system. Some other sources of strings and codes to match to would be: previous waves of the existing survey. For example, if twenty percent of the survey codes have been manual coded, this file can be used as a database to match to the remaining codes. Other sources of codes and strings are previous studies which you have access to which used the same coding scheme that you intend to use. Another overlooked database which may be present in your own survey is other coding questions. For example if both a respondent and her or his parents' occupations are coded, the second variable can initially be automatically coded from the first variable and the third variable can be auto-coded using SAS from the first and second variable prior to any manual coding being conducted . Substantial portions of coding can be accomplished using auto-coding prior to use of manual coders.

In addition to the use of match-merging, other SAS tools can be used to help automate the coding process. SAS functions can also be used to speed up and aid the manual coding process.

QUALITY CONTROL

Match -merging can be used for quality control process. Manually it would be difficult for coders or supervisors to carefully check for consistency, but SAS's match-merging data step makes consistency checking a relatively easy task to accomplish with substantial payoffs in consistency and quality.

Codes attached to strings from different occupational variables in the same study (e.g., mother's occupation, father's occupation, student's occupation , former occupation) or in different waves of the same research project can be match-merged into the same database. If the occupational or industrial string matches from the

different file, a comparison of the code should be performed. If the strings match and the occupational or industrial codes do not match, then a new file should be output which can be examined by master occupational and industrial coders. Thus, the most skilled coders can be used to resolve any discrepancies. Major embarrassment can be caused by inconsistent codes attached to the same string. A file can be created which merges mother's occupation, mother's occupational code, father's occupation, and father's occupational code. The file can be merged by the occupational string (with the SAS RENAME statement used in the DATA step to rename variables from different files that have the same name—e.g., momocc and dadocc instead of occup.). The occupational code attached to mother's and father's occupation can be compared. If the codes are found to be inconsistent then the observation can be output to a new file which can be examined by a supervisor or "expert" coder.

HELPFUL SAS FUNCTIONS

There are a number of SAS character functions which can aid in the process of coding occupations and industry. Some functions which can prove useful in automating the coding process are discussed below. During the first wave of any coding project, it may not be possible to match merge to an existing occupational or industrial strings and codes database. Moreover, existing coding systems may be much too specific or too detailed to match to the scheme your project needs. Moreover, no previous studies may have used the coding system that your project will be using. Given this situation, SAS still has tools to offer a project to help with the coding. A new subset file of unique occupational and industrial strings can still be used. Moreover, other SAS character functions offer help to partially automate the coding task.

UPCASE

The function UPCASE is particularly useful when data is going to be edited, coded, sorted, or compared in a system that does not automatically convert case. Before creating output files from the occupational and/or industrial strings, the occupational and industrial string files should be cleaned of case problems. Strings should be produced that are all in the same case.

```
Industup = UPCASE(indust);
```

SUBSTR

The function SUBSTR can also be used to quickly search through a number of occupational of related variables for a root string. For example, if industry is a twenty character string, and you are interested in searching for references to public, you could write the following SAS statements:

DATA govt;

```
SET occupdb;
subocc = SUBSTR(occup,1,6);
IF subocc = "public" THEN OUTPUT govt;
RUN;
```

Combining the UPCASE and SUBSTR functions can yield substantially improved results when string matching. A file containing industry strings was used to test the usefulness of two different SAS statements, one without character functions and one with the SAS character functions UPCASE and SUBSTR. The file containing industrial strings included the following entries with the word "primary" in the string entry:

```
primary school
PRIMARY EDUCATION
PRIMARY
primary ed
PRIMARY SCHOOL.
```

The following SAS statement yields no output records:

```
IF indus = 'primary' THEN OUTPUT d3;
```

However, by adding the UPCASE and SUBSTR functions to the statement, as shown below, all five records were output to file d3.

```
IF UPCASE((SUBSTR(indus,1,7)))= 'PRIMARY' THEN
  OUTPUT d3;
```

COMPRESS

The compress function can help in the auto-coding process by eliminating the problem of blanks located in different locations causing the same industry to fail a test of equality. By removing blanks, the COMPRESS function can aid the auto-coding process by finding more string matches among files of occupational or industrial strings. The SAS statement without COMPRESS fails to find matches if extra blanks are embedded in the string. For example, "sec school" will be output by both of the following SAS statements but " sec school" will only be output by the second SAS data statement which is using the COMPRESS function.

```
IF indus = 'sec school' THEN OUTPUT educ;

IF COMPRESS(indus) = 'secschool' THEN
  OUTPUT educ;
```

Besides the actual industrial string being output to a file, the full complement of related variables for a person could also be output: industry, job duties, job title, salary. SAS can help automate the coding process so

that coders can see the "big picture" of the pattern of responses.

INDEX

Another useful SAS character function is INDEX. INDEX can be used to search for strings associated with specific occupations or industries. INDEX searches the first argument specified for the character string specified in the second argument. A 0 is returned if INDEX fails to find the string in the second argument. A file can be output from records that return a value greater than 0. The following code is useful in producing a file of industry strings related to education.

DATA indlook;

```
INFILE 'file-specification';
```

```
INPUT
  @1 indus      $25.;
```

```
a='school';
b='educ';
c='elem';
d='secondary';
e='primary';
f='college';
g='university';
h='academ';
```

```
inda=INDEX(indus,a);
indb=INDEX(indus,b);
indc=INDEX(indus,c);
indd=INDEX(indus,d);
inde=INDEX(indus,e);
indf=INDEX(indus,f);
indg=INDEX(indus,g);
indh=INDEX(indus,h);
```

```
ARRAY ind inda indb indc indd inde
        indf indg indh;
```

```
DO OVER ind;
```

```
IF ind > 0 THEN OUTPUT indlook;
END;
```

```
RUN;
```

```
PROC PRINT DATA=indlook;
RUN;
```

Appendix A has a data file used in the above SAS program. Appendix B has the output from PROC PRINT which when viewing variables inda-indh provides a clear picture of how the INDEX function can be used to create a subset file related to an industry and/or occupation which can be used to help automate the coding process.

The INDEX function combined with the OUTPUT SAS data statement is helpful in creating a file that quickly

brings together records that may fall into a specific category of industry and occupation and permits the coder to see the "forest" instead of just looking at the trees. After seeing all the related occupation and industry variables together, it may be possible to code at least a portion of them in an automated fashion.

AUTO-CODING AND QUALITY CONTROL

Using SAS to create an automated approach to coding is consistent with producing a quality product. Auto-coding, when managed properly, can often provide higher quality data than a non-automated approach which uses individual coders to enter codes one by one. What advantages can be found in auto-coding?

Consistency

Auto-coding is a tool which can improve the consistency in occupational and industrial coding far beyond what is possible at the manual level. Even with a large number of supervisors, experts and specialists, it is very difficult to assure consistency across occupational variables, industrial variables, different waves of the same survey, and variables in different instruments. However, what is difficult at the manual level can actually be an easy job when using automated coding.

Increased Expert or Supervisor Checking

Resources are always limited. If less time and money has to be spent on the basic repetitive tasks, more money and time can be spent on quality control which includes increased supervisory checking and increased resources for expert coders who can then focus on more difficult coding questions.

View of The Big Picture

Use of auto-coding helps liberate coders and coding supervisors from a one by one approach to coding. Auto-coding permits coders to view the distributions of their strings and their codes. Moreover, management decisions about labor distribution can be influenced by the overview of the scope of the industry and occupation coding task.

Access To Multiple Databases

Auto-coding permits use of more than one database at the same time. Strings and codes from multiple databases can be merged and, if the same coding system is being used, coding performed on different databases can contribute to current projects.

Access To Codes From Different Waves

Similarly, coding done on earlier waves of the same study can be used to provide a database of occupational and industrial strings and codes for the current wave of the study. Match-merging can be performed on strings with codes from earlier waves

being attached to the current waves. Coding which is performed automatically can then be reviewed by expert coders.

Improved Speed and Quality

Auto-coding should lead to improvements in both the speed and quality of coding.

CONCLUSION

Auto-coding occupational and industrial character string variables with SAS has much to recommend it. Improved processing speed and a reduction in hours needed for manual occupational and industrial coding are improvements which can be expected from auto-coding. It has been argued in this paper that auto-coding with SAS would not reduce the quality of occupational and industrial coding and, in fact, could be expected to actually improve the quality of a project's occupational or industrial coding. Auto-coding would improve the consistency of the application of codes. Auto-coding would permit speedy and accurate review of codes to assure that the same occupational string, when appearing in different occupational or industrial variables, was not given discrepant codes. Moreover, the limited resource of the expert coder or of the project's most experienced coders could be used to code occupational and industrial strings which were not handled at a automated level, and also to review the codes of other coders. More money and time would be available for review and quality control checks because a portion of the base occupational and industrial coding task, the simpler or more obvious coding applications, could be performed through automation.

REFERENCES

- Executive Office of the President, Office of Management and Budget, *Standard Industrial Classification Manual*, National Technical Information Service, Springfield, VA, 1987.
- SAS Institute Inc., *SAS Language: Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1990
- SAS Institute Inc., *SAS Programming Tips: A Guide to Efficient SAS Processing*, Cary, NC: SAS Institute Inc., 1990
- U.S. Department of Commerce, Office of Federal Statistical Policy and Standards, *Standard Occupational Classification Manual*, U.S. Government Printing Office, Washington, D.C., 1980.
- Westat, Inc. *Literal Strings and Codes Data File: 1987 National Postsecondary Student Aid Study*, Westat, Inc., Rockville, MD, November, 1988;

APPENDIX A

school
 work in school
 lunch counter
 beauty parlor
 construction
 education
 primary school
 PHARMACEUTICAL
 PRIMARY EDUCATION
 PRIMARY
 primary ed
 PRIMARY SCHOOL
 university
 college
 electrical industry
 data processing
 sec school
 SOCIAL SERVICES
 sec education
 academia
 government
 city government
 school
 sec school

APPENDIX B

Example of Use of INDEX Function to Create File
 of Teaching Related Industries from Industry Strings Database

OBS	INDUS	A	B	C	D	E	F
1	school	school	educ	elem	university	college	primary
2	work in school	school	educ	elem	university	college	primary
3	education	school	educ	elem	university	college	primary
4	primary school	school	educ	elem	university	college	primary
5	primary school	school	educ	elem	university	college	primary
6	primary ed	school	educ	elem	university	college	primary
7	university	school	educ	elem	university	college	primary
8	college	school	educ	elem	university	college	primary
9	sec school	school	educ	elem	university	college	primary
10	sec education	school	educ	elem	university	college	primary
11	school	school	educ	elem	university	college	primary
12	sec school	school	educ	elem	university	college	primary

OBS	G	H	INDA	INDB	INDC	INDD	INDE	INDF	INDG	INDH
1	secondary	teach	1	0	0	0	0	0	0	0
2	secondary	teach	9	0	0	0	0	0	0	0
3	secondary	teach	0	1	0	0	0	0	0	0
4	secondary	teach	9	0	0	0	0	1	0	0
5	secondary	teach	9	0	0	0	0	1	0	0
6	secondary	teach	0	0	0	0	0	1	0	0
7	secondary	teach	0	0	0	1	0	0	0	0
8	secondary	teach	0	0	0	0	1	0	0	0
9	secondary	teach	8	0	0	0	0	0	0	0
10	secondary	teach	0	5	0	0	0	0	0	0
11	secondary	teach	1	0	0	0	0	0	0	0
12	secondary	teach	5	0	0	0	0	0	0	0