MAKING HEALTH ANALYSTS INDEPENDENT: A USER-FRIENDLY MANAGEMENT INFORMATION SYSTEM

Leslie L. Roos, University of Manitoba
Andre Wajda, University of Manitoba
J. Patrick Nicol, University of Manitoba

INTRODUCTION

Continuing concerns about the cost and quality of health care suggest the importance of making it easier to analyze data relevant to questions of utilization, new programs, and technology assessment. Health Applications System (HAS) provides a fast, low-cost management information system for analyzing data available on existing hospital discharge abstracts, medical claims, or cancer registries.

The information system outlined here is designed to be responsive to the perceived needs of an important group of end users—health policy analysts. Analysts should be able to access necessary information as easily as possible with minimal help from programming staff. To the extent that programmers are involved, the software is designed to improve programmer productivity.

The Health Applications System is a set of modules written in the SAS System's macro language; these modules allow the nonprogrammer to put in his own control statements and run applications which would otherwise involve complicated programming. Thus, SAS' modular structure facilitates users' doing their own data analysis at their own level of sophistication.

The Health Applications System, in combination with the SAS System, permits the user to select appropriate options and link predesigned modules into a single program stream. These modules include:

- A linkage module: LINKS is a series of procedures which perform record linkage between datasets, thus enabling matching across files.

Command Language

A simple command language provides control over:

- the selection and specification of background characteristics, diagnoses, and procedures which define records to be entered into the reports or statistics;
- the selection of time periods for which reports are to be created; and
- the presentation and detailed breakdown of statistics by various classifications.

Each module in the Health Applications System begins with the program call statements and ends with the _RUN statement. In order to distinguish between SAS statements and statements written using the macro processor, each statement in the Health Applications System starts with an underscore.

A number of commands are common to two or more procedures:

_ID - identifies individuals according to the criteria in each data set.
_SELECT - selects individuals when used with _COMBINE, _BUILD, and _HISTORY.
_SELECT - selects records in _RATES. This command operates as a filter to select variables and values which will be used subsequently in the run.
_DEFINE - defines cases for analysis or record management. This command is used to define the variables and values for specific analyses.
_RUN - indicates the end of the program statements.

Both _SELECT and _DEFINE use the SAS conventions for comparison operators (equal to, not equal to, greater than, etc.) for choosing appropriate values of the variables. HAS programs can use any number of _DEFINE statements. In addition to the main statements, _ID, _SELECT, and _DEFINE, each procedure has optional statements for specific
data and record definitions.

The Health Applications System allows the user to identify individuals, select records, and define cases for analysis in a manner understandable to nonprogrammers. For example:

```
 data HISTORY DATA=DATA1 OUT=DATA2;
  ID SSNUMBER;
  select SEX=1 Age>=25;
  define TEST DIAG1-DIAG4='001'
    TARIFF=370-373,375;
 run;
```

specifies that individuals are identified according to the variable SSNUMBER; only records for males (SEX=1) aged 25 years or older are to be analyzed. For each individual, counts of the number of records with at least one diagnosis out of the DIAG1-DIAG4 equal to 001 and tariff equal to one of 370-373 or 375 are generated. The records are read in from data set DATA1; the results are to be saved to a data set DATA2 which consists of one record per individual.

### RATES

The analysis module RATES:

- generates area-wide and hospital-specific information for evaluation of utilization and for cost control.
- highlights high and low use areas.
- produces data on rates and length of stay to aid in dealing with requests for new programs and facilities.
- permits the use of hospital and physician identifiers (admitting physician, physician performing surgery) to examine patient frequency, length of stay, and mortality.
- can generate morbidity and mortality statistics.

The RATES procedure combines hospital admission/separation records or other claims-type data with population information to compute rates of hospitalization and other statistics for user-defined regions. Such "small area analysis" is particularly important because admission rates for most medical and surgical hospitalizations differ markedly among hospital market areas. Since "differences in illness rates cannot provide an adequate explanation for the differences in hospitalization rates seen in small-area studies" (Wennberg et al., 1984), this variation is increasingly the focus of both research and policy-makers' interest (AMA, 1986).

Length of stay has become a major focus of attention since the implementation of the U.S. reimbursement system based on diagnosis-related groups (DRGs). This DRG classification system for hospitalized cases has been implemented by the American Medicare program to control costs by reimbursing hospitals on a cost-per-case basis (Fetter and Freeman, 1986). If length of stay, total charges, number of lab tests, or any other such data on the hospital record identifying the case are available, HAS facilitates calculating length of stay and cost per case for each area and per 10,000 residents. If there is a code identifying the hospital, surgeon, or attending physician, the user can request separate tables indicating number of admissions, average length of stay, and number of deaths for each of these codes. The program can also analyze data and print statistical summaries for several medical procedures or diagnoses in the same run.

To implement the RATES procedure, the user must supply the names of two input data sets:

- a data file (hospital or medical claims, cancer registry records);
- a population file.

The data in both files are used to produce tables of statistics for each specified procedure. The first two tables contain "Summary of Rates" by age and sex and "Summary of Rates by Region" information. A third optional table provides a variety of statistics associated with total days of hospitalization or total costs per patient by region. Length of stay and mortality summaries by a specified variable, such as hospital number of physician number, can also be generated and listed in a fourth optional table.

### COMBINE

- provides the capacity to look at important indicators for quality of care assessment.
- facilitates quality of care audits by providing committee members with lists of relevant hospital readmissions suspected to be complications.

The COMBINE procedure tracks records for two related user-defined events. For example, COMBINE will search files of hospital discharge abstracts for a specific type of admission followed by a specific type of readmission for the same patient. If both admission and readmission records are found, the COMBINE procedure automatically saves all relevant data from both records into an output file. This file can be conveniently analyzed by SAS.
COMBINE also counts the number of input records read, the number of admissions, and the number of readmissions selected. All of these counts are again saved to the output file.

BUILD saves selected information from a combination of individual records. This is particularly useful in longitudinal studies.

At the option of the user, the output from the COMBINE procedure may contain several records for one individual, i.e., if an individual had two readmissions, this information would be stored on two separate records. BUILD rearranges the data from the COMBINE output into one record per individual. The user need only specify the input variables that would uniquely identify an individual, for example, registry number, year of birth, or sex, and the variables to be included in the new output dataset. BUILD then produces output which lists the individual identifiers, the admission data requested, and all the readmissions on one record. This data is again saved to an output file for further analysis. Thus, COMBINE and BUILD save and rearrange particular data elements.

HISTORY uses diagnostic data on the hospital admission/separation abstract and patient histories of prior usage to help control for different outcomes among hospitals.

- identifles problems for which care has been obtained before a given admission occurred.

The HISTORY procedure is designed to facilitate building person-based records from files with multiple records per individual. Diagnostic data on the hospital discharge abstract can be used to study comorbidity in general (single versus multiple diagnoses) and to identify specific risk factors (coronary problems, diabetes) (Luft and Hunt, 1986; Wennberg et al., 1987). Such data have been instrumental in developing the widely-used DRG system (Fetter and Freeman, 1986). This information has the disadvantage of being collected during, rather than before, a given hospital stay. Thus, the results of the treatment may inappropriately affect the case-mix data (myocardial infarct occurring after surgery). Consistent patient identifiers are not required and no effort is made to trace individual usage through time.

When consistent identifiers are available, the HISTORY procedure provides the capability of linking claims through time to develop more sophisticated measures of patient case severity. Thus, when the analyst is able to determine if patients have been hospitalized in the prior six months for cardiovascular disease or cancer, a powerful case-mix adjustment is possible.

More generally, "before" and "after" histories from claims data or hospital abstracts can be used to study outcomes of common medical conditions and surgical procedures (Wennberg et al., 1987). The program allows creation of "before" covariates specified by the user. Such covariates can be incorporated into outcomes analyses using variables generated in the "after" histories. Standard statistical analyses using the SAS System can be performed on the output file generated.

LINKS

- assists in the use of administrative records for research purposes by organizing the facts together.
- facilitates cross checks between separately maintained datasets to highlight quality problems.
- links individual records with mortality information to study long-term survival.

LINKS is comprised of a series of procedures which perform record linkage functions. The statistical methodology for performing this record linkage is well-developed (Smith, 1984; Newcombe et al., 1983). The first two procedures, TESTPWR and TESTPKT, permit the user to easily generate information about the structure of the datasets. TESTPWR prescreens the datasets to tell the user which variables would be the most useful for the linkage. An output table is produced which lists the frequency of all the variables, the number of missing values, the discriminating power, and the Shannon Entropy. TESTPKT assists in finding the combination of variables that will give a unique identification for each individual. The procedure splits and sorts the datasets into "pockets" defined by a sequence of variables. An output table lists the variables that define each pocket, along with the number of pockets and the minimum, maximum, and average number of records in each pocket.

As outlined in Figure 1, the actual linkage is performed by four procedures. LINKEXC compares the two datasets for records that agree on all comparison variables, i.e., it finds all the "perfect" matches. These matches are saved to an output file. The remaining unmatched records from the two datasets are also saved in their own output files. This reduces the number of records...
that need to be considered for the probabilistic matching.

The first phase of this matching, LINKPKT, links records within pockets if they agree on a minimum number of variables as specified by the user. Again, three output files are saved: one with the new paired records; a second with the records remaining from the first dataset; and the third with the remaining records from the second dataset. LINKWGT then attaches probabilistic weights to the records paired by LINKPKT. The user specifies the variables, the weighting factors, and any conditions on the weights. The last procedure, LINKRES, then resolves the links using the weight variable. Two output files are produced: one with the resolved links and the other with the unresolved links. Normal SAS System procedures can, of course, be applied to the output files.

BENCHMARKS

 Besides allowing nonprogrammers to make relatively complicated runs, the Health Applications System has improved the efficiency of our computer operations. Well-designed macros run considerably faster than complicated SAS programs developed by people without professional programming experience. HAS is relatively fast computationally, thus economical of computer time. For example, on the Amdahl 5850, eighty thousand hospital claim records containing fourteen variables were read in approximately twenty seconds of CPU time. Using the RATES module, analysis of utilization, length of stay, and mortality for a single hospital procedure required an additional sixteen seconds. When three procedures simultaneously analyzed, approximately twenty-seven additional seconds of CPU time were needed.

 Runs using the COMBINE and HISTORY modules were somewhat longer if a large input file needed to be sorted by patient number. BUILD is very fast since it uses a dataset already created by the COMBINE procedure. To complete the entire LINKS procedures, working with two datasets of forty thousand records and eight variables, approximately one hundred and seventy seconds of CPU were required.

USES of HEALTH APPLICATIONS SYSTEM

 Insurance carriers such as Medicare in the United States and the Canadian provincial insurers are natural users of the software described here; they need both area-wide and hospital-specific information to evaluate existing utilization. Because some changes in utilization can occur fairly rapidly (Barnes et al., 1984; Etheredge and Juba, 1984), a management information system for health analysts should help produce timely reports. Using recent data is important, since the questions: "where and how are we spending all that money?" are ones which policy-makers especially want answered.

 Identifying high use areas and analyzing data on rates and length of stay can aid in dealing with requests for new medical care programs and expansion of health care facilities. Depending upon the cost-containment strategies adopted, there is the danger that high-use areas may continue to be favoured by policy makers (Wennberg, 1982). Indeed, high-use areas may be among those most vigorous in pressing for additional funding. Monitoring both overall usage and specific types of procedures seems important. Thus, utilization review has been noted in the press as "an idea whose time has come" (Paris, 1985), but reducing the costs of such review depends on the kind of software outlined here.

REFERENCES


Paris E: Hold that scalpel? Forbes 1985(May); 35-36.


*SAS is the registered trademark of SAS Institute Inc., Cary, NC, USA.

Acknowledgements
The authors gratefully acknowledge the help of the Manitoba Health Services Commission. This research was supported by National Health Research and Development Project No. 6607-1197-44 and by Career Scientist Award No. 6607-1314-48, Health and Welfare, Canada. Interpretations and viewpoints contained in this paper are the authors' own and do not necessarily represent the opinion of either the Manitoba Health Services Commission or Health and Welfare Canada.

Contact Author:
Leslie L. Roos
Faculty of Management
University of Manitoba
Winnipeg, Manitoba
Canada R3T 2N2