

**EVERYTHING IN ONE RECORD:
CREATING ONE-LINERS WITH MERGES AND RETAINS**

Sharon W. Stroud, Medical Review of North Carolina, Inc., Cary, NC
David P. Biggs, Medical Review of North Carolina, Inc., Cary, NC

ABSTRACT

In our hospital claims relational Oracle database, a patient with multiple diagnoses for one hospital stay appears in the dataset multiple times, each diagnosis distinguished by a sequence field. When analyzing this data it is best to have one record per patient stay, with all diagnosis codes in that record.

The genius program converts a 'long' dataset with multiple related records into a shorter, wider dataset using a series of merges. The new dataset has fewer records and more variables, making analysis much easier. The genius2 program performs the same conversion more efficiently, making use of the retain statement.

INTRODUCTION

At Medical Review of North Carolina, Inc., much of the data analyzed come from Medicare health claims. The data are stored in multiple tables in an Oracle database. Analysts extract data and convert to SAS datasets for analysis. The Medicare claims diagnosis table includes, among others, the following variables: hic (health identification code) number, admit date, discharge date, diagnosis code and sequence. Hic, admit date, and discharge date together uniquely identify a patient hospital stay. The sequence variable distinguishes the numerous diagnosis codes; a sequence of 00 identifies the primary diagnosis code, sequences 01 to 08 identify secondary diagnoses, and a sequence of AD identifies the admitting diagnosis.

For any inpatient hospital stay a patient is given a minimum of two diagnoses: an admitting and a primary diagnosis. In addition, there may be up to eight additional secondary diagnoses. In the Oracle table, each diagnosis code is a separate entry - so a single hospital stay is in the dataset between two and ten times, depending on the number of secondary diagnosis codes assigned.

For analysis, it is preferable to have a dataset with only one record per hospital stay, with admission diagnosis (dxadm), primary diagnosis (dx1), and all secondary diagnoses (dx2, dx3, etc.) contained in that record:

<u>hic</u>	<u>admit</u>	<u>disch</u>	<u>dx1</u>	<u>dx2</u>	<u>dx3</u>	<u>dx4</u>	<u>dx5</u>	<u>dx6</u>
9999999999A	5/5/97	5/10/97	1983	5990	2761	.	.	.

<u>dx7</u>	<u>dx8</u>	<u>dx9</u>	<u>dxadm</u>
.	.	.	1950

rather than

<u>hic</u>	<u>admit</u>	<u>disch</u>	<u>diag</u>	<u>seq</u>
99999999999A	5/5/97	5/10/97	1950	AD
99999999999A	5/5/97	5/10/97	1983	00
99999999999A	5/5/97	5/10/97	5990	01
99999999999A	5/5/97	5/10/97	2761	02

THE GENIUS PROGRAM

The genius program does such a conversion using a series of merges.

```
*****
*   genius.pgm   *
*****
;

data new;
  set query;
  if seq='00' then dx1=dx;
  if seq='01' then dx2=dx;
  if seq='02' then dx3=dx;
  if seq='03' then dx4=dx;
  if seq='04' then dx5=dx;
  if seq='05' then dx6=dx;
  if seq='06' then dx7=dx;
  if seq='07' then dx8=dx;
  if seq='08' then dx9=dx;
  if seq='AD' then dxadm=dx;
run;

data uno (drop= dx dx2-dx9 dxadm);
  set new;
  if seq='00';
run;

data dos (drop=dx dx1 dx3-dx9 dxadm);
  set new;
  if seq='01';
run;

data tres (drop=dx dx1 dx2 dx4-dx9 dxadm);
  set new;
  if seq='02';
run;

data cuatro (drop=dx dx1-dx3 dx5-dx9
               dxadm);
  set new;
  if seq='03';
run;

data cinco (drop=dx dx1-dx4 dx6-dx9
              dxadm);
  set new;
  if seq='04';
run;

data seis (drop=dx dx1-dx5 dx7-dx9 dxadm);
  set new;
  if seq='05';
run;

data siete (drop=dx dx1-dx6 dx8 dx9
             dxadm);
  set new;
  if seq='06';
run;
```

```

data ocho (drop=dx dx1-dx7 dx9 dxadm);
  set new;
  if seq='07';
run;

data nueve (drop=dx dx1-dx8 dxadm);
  set new;
  if seq='08';
run;

data diez (drop=dx dx1-dx9);
  set new;
  if seq='AD';
run;

proc sort data=uno; by hic admit disch; run;
proc sort data=dos; by hic admit disch; run;
proc sort data=tres; by hic admit disch; run;
proc sort data=cuatro; by hic admit disch; run;
proc sort data=cinco; by hic admit disch; run;
proc sort data=seis; by hic admit disch; run;
proc sort data=siete; by hic admit disch; run;
proc sort data=ocho; by hic admit disch; run;
proc sort data=nueve; by hic admit disch; run;
proc sort data=diez; by hic admit disch; run;

data final (drop=seq);
  merge uno dos tres cuatro cinco seis siete ocho
        nueve diez;
  by hic admit;
run;

```

The program creates ten new variables , dxadm and dx1 to dx9, and assigns the appropriate diagnosis code to each variable. A diagnosis sequence of 00 is the primary diagnosis code, so the associated code becomes dx1. The sequence 01 code becomes dx2, and so forth. Each record still contains only one diagnosis code, but now there are nine variables containing blanks. At this point the data layout is:

hic	admit	disch	dx1	dx2	dx3	dx4	dx5
9999999999A	5/5/97	5/10/97	1983	.	.	.	.
9999999999A	5/5/97	5/10/97	.	5990	.	.	.
9999999999A	5/5/97	5/10/97	.	.	2761	.	.
9999999999A	5/5/97	5/10/97	.	.	.	.	.

dx6	dx7	dx8	dx9	dxadm	seq
.	.	.	.	.	00
.	.	.	.	.	01
.	.	.	.	.	02
.	.	.	.	1950	AD

Ten datasets are then created, each one containing the identifiers (hic, admit and disch) and the corresponding diagnosis code - uno contains dx1, dos contains dx2, etc. The other variables that held blank values are dropped, as well as the original diagnosis code variable (dx). Finally the ten datasets are sorted and merged, creating one dataset with the desired layout, and the seq variable is dropped.

THE GENIUS2 PROGRAM

When working with very large datasets, the genius program may take several hours to run, or SAS can run out of temporary space when trying to create and sort the ten new datasets. To avoid these problems, the genius2 program was developed, using the retain statement.

```

*   genius2.pgm   *
*****

proc sort data=query;
by hic admit disch seq;
run;

data new (drop=seq dx);
  set query;
  by hic admit disch;
  if seq='00' then do; dx1=dx;
    retain dx1; end;
  if seq='01' then do; dx2=dx;
    retain dx2; end;
  if seq='02' then do; dx3=dx;
    retain dx3; end;
  if seq='03' then do; dx4=dx;
    retain dx4; end;
  if seq='04' then do; dx5=dx;
    retain dx5; end;
  if seq='05' then do; dx6=dx;
    retain dx6; end;
  if seq='06' then do; dx7=dx;
    retain dx7; end;
  if seq='07' then do; dx8=dx;
    retain dx8; end;
  if seq='08' then do; dx9=dx;
    retain dx9; end;
  if seq='AD' then do; dxadm=dx; end;
  if last.disch then do;
    output;
    dx1='';
    dx2='';
    dx3='';
    dx4='';
    dx5='';
    dx6='';
    dx7='';
    dx8='';
    dx9='';
  end;
run;

```

The data are first sorted by the three identifiers and the sequence field. Note that SAS will sort seq='AD' to the last entry for a hospital stay. Each diagnosis code is assigned to a new variable and then retained as the next observation is read. There is no need to retain the admission diagnosis, dxadm, since this will always be the last in a group. At this point the data layout is:

hic	admit	disch	dx1	dx2	dx3
9999999999A	5/5/97	5/10/97	1983	.	.
9999999999A	5/5/97	5/10/97	1983	5990	.
9999999999A	5/5/97	5/10/97	1983	5990	2761
9999999999A	5/5/97	5/10/97	1983	5990	2761

dx4	dx5	dx6	dx7	dx8	dx9	dxadm	seq
.	.	.	.	.	.	.	00
.	.	.	.	.	.	.	01
.	.	.	.	.	.	.	02
.	.	.	.	.	.	1950	AD

Since only the last observation for a group contains all the diagnosis codes, we output only for last.disch. (SAS uses the first. and last. prefixes to identify the beginning and ending of by groups.) It is necessary to output at this point and set all the dx variables to missing so that no diagnosis codes will be

carried over to another hospital stay.

CONCLUSION

Genius2 is a more efficient program, creating only one new dataset and requiring only one sort. By running both programs on the same datasets, we found that genius2, using retains, ran in less than half the time of the first genius program.

The genius2 program is one of the few where we have been able to make use of the retain statement. The genius program is a good alternative if a user is unfamiliar or uncomfortable with the retain statement.

CONTACT INFORMATION

Sharon Stroud
David Biggs
Medical Review of North Carolina, Inc.
5625 Dillard Drive
Suite 203
Cary, NC 27511
919-851-2955
919-851-8457 (fax)
ncpro.sstroud@sdps.org
ncpro.dbiggs@sdps.org