

Paper 089-2008

It's a Bird, It's a Plane, It's SQL Transpose!

Ted Conway, Chicago, IL

ABSTRACT

Ever wish you could easily transpose and summarize monthly data using just PROC SQL? Regardless of whether the data is stored in a SAS dataset or other relational databases? Even if scores of variables for dozen of months are involved? This paper tackles the task of data denormalization with a simple macro that uses PROC SQL to take a "vertical" table with multiple rows per subject and transform it into a "horizontal" table containing one row per subject with aggregated monthly values.

INTRODUCTION

Much to the chagrin of Data Modelers and DBAs everywhere, sometimes analysis and reporting is a lot easier with a one-row-per-subject table than a multiple-rows-per-subject table.

On his Statistical Computing weblog, Wensui Liu describes three ways of converting a "long" table to a "wide" table using SAS:

- PROC TRANSPOSE
- PROC SQL
- DATA STEP

Wensui's PROC SQL solution, repeated below, is a logically appealing technique that has the added bonus of being able to be used with SAS datasets as well as other relational databases:

```
*****
* METHOD 2: PROC SQL                               *
*****;
proc sql;
create table wide2 as
select id,
sum(case when time = 1 then x else 0 end) as x1,
sum(case when time = 2 then x else 0 end) as x2,
sum(case when time = 3 then x else 0 end) as x3
from long group by id;
```

While the above solution looks attractive, it loses some of its luster when you consider what you'd have to do to deal with more variables and time periods.

For example, what if you had to deal with 36 months instead of the 3 time periods above? And what if you were dealing with 200 variables instead of the 1 column above?

That, my friend, is where the **LONG2WIDE** macro comes in!

LONG2WIDE MACRO – SAMPLE USAGE

Before looking at the guts of the **LONG2WIDE** macro, let's first take a look at how one might use it:

```
%long2wide(data=sashelp.prdsale, by=product,
           months=month jan1993-dec1994, vars=actual predict,
           out=prdsalesum);
```

In the above example, the **data** keyword parameter is used to specify the input SAS data set (*sashelp.prdsale*, which is provided with Base SAS).

The **by** parameter is used to specify the level of summarization – in this instance, we want to create one row at the *product* level.

The **months** parameter is used to specify the name of the variable containing the date value (*month* in this example), as well as a list of individual month values (or hyphen-separated ranges) that are to be summarized and transposed.

The **vars** parameter is used to list the variables that are to be transposed and summarized.

The final parameter, **out**, is used to specify the name of the output dataset that will hold the output dataset. Here, *prdsalesum* will contain variables *product*, *actual199301-actual199412* and *predict199301-predict199412*. We've been able to denormalize the data into a new flat structure containing 49 variables with a mere three lines of code!

LONG2WIDE MACRO – SAMPLE INPUT – SASHELP.PRDSALE

NOTE: Table has been opened in browse mode.

LONG2WIDE MACRO – SAMPLE OUTPUT – WORK.PRDSALESUM

NOTE: Table has been opened in browse mode.

LONG2WIDE MACRO – CODE

```

%macro long2wide(data=_last_, by=, months=, vars=, out=_data_);
proc sql;
create table &out as select &by
%let dv=%scan(&months,1);
%let i=1;
%let v=%scan(&vars,1);
%do %while("&v" ^= "");
  %let j=2;
  %let mo=%scan(&months,2," ");
  %do %while("&mo" ^= "");
    %let lmo="01%scan(&mo,1,"-")"d;
    %let hmo="01%scan(&mo,2,"-")"d;
    %if &hmo="01"d %then %let hmo=&lmo;
    %let nmo=%sysfunc(intck(month,&lmo,&hmo));
    %do m2=0 %to &nmo;
      %let mo1=%sysfunc(intnx(month,&lmo,&m2),date9.);
      %let mo2=%sysfunc(intnx(month,&lmo,&m2),yymmnn6.);
      , sum(case when &dv="&mo1"d then &v end) as &v&mo2
    %end;
    %let j=%eval(&j+1);
    %let mo=%scan(&months,&j," ");
  %end;
  %let i=%eval(&i+1);
  %let v=%scan(&vars,&i);
%end;
from &data group by &by order by &by;
%mend;

```

LONG2WIDE MACRO – SAMPLE MACRO INVOCATION

```

%long2wide(data=sashelp.prdsale, by=product,
           months=month jan1993-dec1994, vars=actual predict,
           out=prdsalesum);

```

LONG2WIDE MACRO – SAMPLE DESIRED TRANSFORMATION

MONTH	PRODUCT	ACTUAL	PREDICT
1993-01	BED	\$31	\$29
1993-02	BED	\$40	\$39
1993-03	BED	\$50	\$40
1993-01	CHAIR	\$76	\$65
1993-02	CHAIR	\$78	\$72
1993-03	CHAIR	\$90	\$92



PRODUCT	ACTUAL199301	ACTUAL199302	ACTUAL199303	PREDICT199301	PREDICT199302	PREDICT199303
BED	\$31	\$40	\$50	\$29	\$39	\$40
CHAIR	\$76	\$78	\$90	\$65	\$72	\$92

LONG2WIDE MACRO – SAMPLE GENERATED SAS CODE

```

proc sql;
create table prdsalesum as select product ,
sum(case when month="01JAN1993"d then actual end) as actual199301 ,
sum(case when month="01FEB1993"d then actual end) as actual199302 ,
sum(case when month="01MAR1993"d then actual end) as actual199303 ,
sum(case when month="01APR1993"d then actual end) as actual199304 ,
sum(case when month="01MAY1993"d then actual end) as actual199305 ,
sum(case when month="01JUN1993"d then actual end) as actual199306 ,
sum(case when month="01JUL1993"d then actual end) as actual199307 ,
sum(case when month="01AUG1993"d then actual end) as actual199308 ,
sum(case when month="01SEP1993"d then actual end) as actual199309 ,
sum(case when month="01OCT1993"d then actual end) as actual199310 ,
sum(case when month="01NOV1993"d then actual end) as actual199311 ,
sum(case when month="01DEC1993"d then actual end) as actual199312 ,
sum(case when month="01JAN1994"d then actual end) as actual199401 ,
sum(case when month="01FEB1994"d then actual end) as actual199402 ,
sum(case when month="01MAR1994"d then actual end) as actual199403 ,
sum(case when month="01APR1994"d then actual end) as actual199404 ,
sum(case when month="01MAY1994"d then actual end) as actual199405 ,
sum(case when month="01JUN1994"d then actual end) as actual199406 ,
sum(case when month="01JUL1994"d then actual end) as actual199407 ,
sum(case when month="01AUG1994"d then actual end) as actual199408 ,
sum(case when month="01SEP1994"d then actual end) as actual199409 ,
sum(case when month="01OCT1994"d then actual end) as actual199410 ,
sum(case when month="01NOV1994"d then actual end) as actual199411 ,
sum(case when month="01DEC1994"d then actual end) as actual199412 ,
sum(case when month="01JAN1993"d then predict end) as predict199301 ,
sum(case when month="01FEB1993"d then predict end) as predict199302 ,
sum(case when month="01MAR1993"d then predict end) as predict199303 ,
sum(case when month="01APR1993"d then predict end) as predict199304 ,
sum(case when month="01MAY1993"d then predict end) as predict199305 ,
sum(case when month="01JUN1993"d then predict end) as predict199306 ,
sum(case when month="01JUL1993"d then predict end) as predict199307 ,
sum(case when month="01AUG1993"d then predict end) as predict199308 ,
sum(case when month="01SEP1993"d then predict end) as predict199309 ,
sum(case when month="01OCT1993"d then predict end) as predict199310 ,
sum(case when month="01NOV1993"d then predict end) as predict199311 ,
sum(case when month="01DEC1993"d then predict end) as predict199312 ,
sum(case when month="01JAN1994"d then predict end) as predict199401 ,
sum(case when month="01FEB1994"d then predict end) as predict199402 ,
sum(case when month="01MAR1994"d then predict end) as predict199403 ,
sum(case when month="01APR1994"d then predict end) as predict199404 ,
sum(case when month="01MAY1994"d then predict end) as predict199405 ,
sum(case when month="01JUN1994"d then predict end) as predict199406 ,
sum(case when month="01JUL1994"d then predict end) as predict199407 ,
sum(case when month="01AUG1994"d then predict end) as predict199408 ,
sum(case when month="01SEP1994"d then predict end) as predict199409 ,
sum(case when month="01OCT1994"d then predict end) as predict199410 ,
sum(case when month="01NOV1994"d then predict end) as predict199411 ,
sum(case when month="01DEC1994"d then predict end) as predict199412
FROM SASHELP.PRDSALE GROUP BY PRODUCT ORDER BY PRODUCT;

```

Imagine how “fun” it would be to hand-code this for two hundred variables instead of two!

CONCLUSION

The **LONG2WIDE** macro provides an easy way of transposing and aggregating relational data into a denormalized format for subsequent analysis and reporting without having to resort to more complicated techniques involving array manipulation and repetitive merges.

For the purposes of this paper, the feature set of the **LONG2WIDE** macro has been kept rather simple.

Feel free to extend it to include transposes of non-numeric data (you can use MAX instead of SUM), process other relational database sources like Oracle, DB2 or Teradata (use pass-through SQL with SAS/ACCESS), pivoting on variables other than months (including calculations) and perhaps even field-level pivot specifications.

Just remember to share your enhancements on SAS-L or in a paper!

REFERENCES AND ACKNOWLEDGEMENTS

In addition to the SAS documentation on PROC TRANSPOSE, there's a wealth of freely-available information on the web on the topic of denormalization.

My own Google searches brought me to Gerhard Svolba's *Efficient "One-Row-per-Subject" Data Mart Construction* (www2.sas.com/proceedings/sugi31/078-31.pdf) and Wensui Liu's *Blog In Statistical Computing* (statcompute.blogspot.com), both of which I referred to in the course of writing this short paper – *thanks Wensui and Gerhard!*

CONTACT INFORMATION

Ted Conway resides in Chicago, Illinois. He can be reached at tedconway@aol.com.

TRADEMARKS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.