



HOT Tips from SAS-L

- OK, I KNOW WHAT A "HOT TIP" IS
- I KNOW WHAT SAS IS
- "**WHAT'S THE -L**"



HOT Tips from SAS-L

from: <http://www.listserv.uga.edu/archives/sas-l.html>



www.listserv.uga.edu
The University of Georgia

[Home](#) | [Browse](#) | [Manage](#) | [Request](#) | [Manuals](#) | [Register](#)

Archives of SAS-L@LISTSERV.UGA.EDU

SAS(r) Discussion

- Search the archives
- Post to the list
- Join or leave the list (or change settings)
- Manage the list (list owners only)

- [August 2008, week 3](#)
- [August 2008, week 2](#)
- [August 2008, week 1](#)
- [July 2008, week 5](#)
- [July 2008, week 4](#)
- [July 2008, week 3](#)
- [July 2008, week 2](#)
- [July 2008, week 1](#)
- [June 2008, week 5](#)
- [June 2008, week 4](#)



HOT Tips from SAS-L

Howard Schreier: *Ask This Old Newsgroup*

<http://www2.sas.com/proceedings/sugi29/247-29.pdf>

- SAS-L is a worldwide online community of SAS software users
- participants discuss various aspects of SAS and help one another solve SAS-related problems
- participants include many of the world's most experienced and knowledgeable SAS users



HOT Tips from SAS-L

- others are newcomers to SAS

MSZ addendum: students looking for homework answers

users with experience in one area who are moving into new SAS products or usages

- customarily, “newbies” are treated well
- their questions are taken seriously, and it’s not unusual to see responses providing tested code and extensive explanations



HOT Tips from SAS-L

Ian Whitlock: *A Personal View of SAS-L as a Teaching Tool*

<http://www2.sas.com/proceedings/sugi25/25/ps/25p249.pdf>

SAS-L is an on-line SAS Users Group with over 2000 members around the world, plus a huge number of silent or nearly silent listeners who follow the discussions via the associated news group, comp.soft-sys.sas.



HOT Tips from SAS-L

- Can you learn from asking questions?

Or is there too much misinformation?

- Can you learn by giving answers?

Or do you need to be an "expert"?

- Is SAS-L sufficiently recognized by training and support people?

Or should you add SAS-L to your learning kit?



HOT Tips from SAS-L

Mike Zdeb: I think ...

it's a bit like calling tech support and having 2000+ people answer the phone rather than just one

it's a bit like 'support via SPAM' ... you post your question and assume that out of all the list subscribers, at least one other person has had a similar problem and is willing to answer (or claim that inheritance in Nigeria)

it's a bit like getting up from your desk and walking into another office to ask a SAS-related question, but you find 2000+ other SAS users



HOT Tips from SAS-L

■ WHAT TYPES OF QUESTIONS ARE ASKED

I am working with a large epidemiological study and I am trying to create a variable that is dichotomous and tells if they have received treatment during any of these six visits.

outcome is a 1 if they have answered 1 on any of the questions, regardless of missing values ... outcome2 is 0 based on the fact all of the observed variables are 0 ... missing values should only be produced if they have a missing values on all the variable as the third line states.

How can this problem be solved?



HOT Tips from SAS-L

code supplied by user ...

```
if y1=1 or y2=1 or y3=1 or y4=1 or y5=1 or y6=1  
then blodbeh=1;  
if y1=0 and y2=0 and y3=0 and y4=0 and y5=0 and  
y6=0 then blodbeh=0;  
if y1= . and y2= . and y3= . and y4= . and y5= . and  
y6= . then blodbeh= .;
```

*not working since second IF requires all y variables (**sp**) to be 0
to make the value of blodbeh 0, not any y*



HOT Tips from SAS-L

■ SUGGESTION #1

```
sumy = sum(of y1-y6);  
if sumy ge 1 then blodbeh = 1;  
else if sumy = 0 then blodbeh = 0;  
else if sumy = . then blodbeh = .;
```

must know ... how do functions work



HOT Tips from SAS-L

■ SUGGESTION #2

```
if y1=1 or y2=1 or y3=1 or y4=1 or y5=1 or y6=1  
then blodbeh=1;  
else  
  if y1=. and y2=. and y3=. and y4=. and y5=. and  
  y6=. then blodbeh=.;  
else  blodbeh=0;
```

add ELSE and change the order of the statements



HOT Tips from SAS-L

- SUGGESTION #3 (from original poster of question)

```
if y1=1 or y2=1 or y3=1 then x1=1;  
if (y1=0 or y1=.) and (y2=0 or y2=.) and  
(y3=0 or y3=.) then x1=0;  
if y1=. and y2=. and y3=. then x1=.;
```

also does not work (2 for 2 !!!)



HOT Tips from SAS-L

- SUGGESTION #4 (DUH ...)

```
blodbeh = max (of y1-y6);
```

again, must know ... how do functions work



HOT Tips from SAS-L

■ ANOTHER QUESTION

What SAS code will help me find the difference between first and last height for each id, please? I have 70 different ids to find the difference between the first and last height.



HOT Tips from SAS-L

- CONVENTIONAL
FIRST. AND LAST. VARIABLES
- CONVENTIONAL WITH A 'TWEAK'
FIRST. AND LAST. VARIABLES WITH A DOW LOOP
- FROM 'OUT OF THE BLUE'
PROC SUMMARY + A DATA STEP



HOT Tips from SAS-L

```
data diff;
retain firstht;
set x;
by id;
if first.id then firstht=ht;
if last.id then do;
  diff = ht - firstht;
  output;
end;
run;
```

CONVENTIONAL

```
data diff;
do until(last.id);
  set x;
  by id;
  if first.id then
    firstht=ht;
  end;
  diff = ht - firstht;
run;
```

WITH A 'TWEAK (DOW)



HOT Tips from SAS-L

■ MORE ABOUT THE DOW

Paul Dorfman: *The DoW-Loop Unrolled*

<http://analytics.ncsu.edu/sesug/2007/SD08.pdf>
Hands-On Workshop at NESUG 2008

The DOW-loop (Whitlock DO-loop) is a nested repetitive DATA step programming structure, intentionally organized in order to allow for programmatically and logically natural isolation of DO-loop instructions related to a certain break-event from actions performed before and after the loop, and without resorting to superfluous conditional statements.



HOT Tips from SAS-L

```
Data ... ;  
<Stuff done before break-event> ;  
Do <Find Specs> Until ( Break-Event ) ;  
Set A ;  
<Stuff done for each record> ;  
End ;  
<Stuff done after break-event... > ;  
Run ;
```

THE DOW ... basic idea, take all the stuff that's done once and get it out-of-the-loop ... often eliminates both the need to RETAIN variables and the use of IF-THEN checking (IF _N_ EQ 1 THEN DO ..., IF LAST THEN DO ...)



HOT Tips from SAS-L

```
proc summary data=x nway;
  class id;
  output out=diff (drop= _:)
    idgroup(out(ht)=first)
    idgroup(last out(ht)=last);
  run;

  data diff;
    set diff;
    diff = last-first;
  run;
```

OUT OF THE BLUE



HOT Tips from SAS-L

■ ANOTHER QUESTION

i'm looking for a solution for the following problem. i have a data set with a variable OLD and want the variable NEED.

old	need
AA	1
AA	1
AA	1
AD	2
AE	3
.	
zz	100

my problem is that the number or variations of old can differ.



HOT Tips from SAS-L

■ DOW VERSUS DUH

```
data want;
need = _n_;
do until (last.old);
  set have;
  by old;
  output;
end;
run;
```

DOW

```
data want;
set have;
by old;
need + first.old;
run;
```

DUH (Howard
Schreier)



HOT Tips from SAS-L

■ CHANGE THE QUESTION

i'm looking for a solution for the following problem. i have a data set with a variable OLD and want to output only those groups with at least 3 members



HOT Tips from SAS-L

```
data want;
do _n_=1 by 1 until (last.old);
  set have;
  by old;
end;
do until (last.old);
  set have;
  by old;
  if _n_ ge 3 then output;
end;
run;
```

DOUBLE DOW WITH `_N_` USED TO COUNT
MEMBERS (NO DROP `_N_` IS NEEDED)



HOT Tips from SAS-L

```
data want;
  set have (in=first) have (in=second);
  by old;
  if first.old then count = 0;
  count + 1*one;
  if two and count ge 3 then output;
  drop count;
run;
```

SELF-INTERLEAVE

SCHREIER: *Interleaving a Dataset with Itself: How and Why*

<http://www.nesug.org/proceedings/nesug03/cc/cc002.pdf>



HOT Tips from SAS-L

■ ANOTHER QUESTION

I have the following example I created:

```
1234 01021969 2345 Y
1234 01301969 3456 N
3157 02031969 2345 N
3157 02201969 2897 N
3157 04151969 2345 Y
1011 02051969 2345 N
1011 02211969 2345 N
1011 05201969 2897 N
2468 03211969 2234 Y
2468 07151969 2255 Y
```

What I want to do is to create a new datafile that has all of the records for the ID's that have a passed = 'Y' for any of the tests



HOT Tips from SAS-L

```
data new;
do until (last.id);
  set practice_dates;
  by id;
  flag = sum(flag,(passed eq 'Y'));
end;
do until (last.id);
  set practice_dates;
  by id;
  if flag then output;
end;
drop flag;
run;
```

DOUBLE DOW



HOT Tips from SAS-L

- DIGRESSION: SEVEN DIFFERENT FLAG STATEMENTS,
DO THEY HAVE DIFFERENT RESULTS ... IF SO, WHY

```
do until (last.id);
  set practice_dates;
  by id;
  if passed eq 'Y' then flag = 1;
  if passed eq 'Y' then flag = sum(flag,1);
  if passed eq 'Y' then flag = flag + 1;
  if passed eq 'Y' then flag + 1;
  flag = sum(flag,(passed eq 'Y'));
  flag = flag + (passed eq 'Y');
  flag + (passed eq 'Y');
end;
```



HOT Tips from SAS-L

```
data new;
set practice_dates (in=first)
    practice_dates (in=second);
by id;
if first.id then flag = 0;
flag + (passed eq 'Y')*first;
if second and flag gt 0 then output;
drop flag;
run;
```

SELF-INTERLEAVE



HOT Tips from SAS-L

```
proc sort data=have;
by id descending passed;
run;

data want (drop=passed_test);
set have;
by id;
if first.id then passed_test=0;
passed_test + (first.id and passed eq 'Y');
if passed_test then output;
run;
```

OUT OF THE BLUE



HOT Tips from SAS-L

■ ANOTHER QUESTION

Have an intriguing (again, it's not my spelling) coding problem I'm yet to solve, hoping someone may be able to assist. I have data set of the form:

Cust Manager

111	A
222	A
333	B
444	C
555	C
666	C
777	D



HOT Tips from SAS-L

I want to output individual data set for each Manager containing only the custs for them, ie:

Manager_A.sas would contain:

111

222

Manager_B.sas would contain:

333

Manager_C.sas would contain:

444

555

666

Manager_D.sas would contain:

777



HOT Tips from SAS-L

- POSTING INCLUDED A DATA-SPECIFIC SOLUTION, BUT THE POSTER WANTED A GENERAL SOLUTION

```
data a b c d;
set have;
if manager eq 'A' then output a;
else
if manager eq 'B' then output b;
else
if manager eq 'C' then output c;
else
if manager eq 'D' then output d;
run;
```



HOT Tips from SAS-L

```
data _null_;
set have;
by manager;
if first.manager then call execute(cat(
'data manager_',manager,';',',
'set have;',
'where manager eq "','',manager,'"';',
'keep cust;',
'run;'

));
run;
```

CALL EXECUTE ... DATA STEP WRITES SAS
CODE THAT EXECUTES AFTER THE DATA
STEP IS COMPLETED



HOT Tips from SAS-L

```
filename mgr temp;

data _null_;
set have;
by manager;
file mgr;
if first.manager then put
'data manager_' manager +(-1) ';' /
'set have;' /
'where manager eq "' manager +(-1) '"';
'keep cust;' /
'run;' /;
run;

%include mgr;
```

SIMILAR TO PUT IN DATA STEP, INCLUDE AFTER
THE DATA STEP



HOT Tips from SAS-L

```
data manager_A;  
set have;  
where manager eq 'A';  
keep cust;  
run;
```

```
data manager_B;  
set have;  
where manager eq 'B';  
keep cust;  
run;
```

```
data manager_C;  
set have;  
where manager eq 'C';  
keep cust;  
run;
```

```
data manager_D;  
set have;  
where manager eq 'D';  
keep cust;  
run;
```



HOT Tips from SAS-L

eliminate need for
FIRST.MANAGER
check

create a data set
with one
observation per
manager

```
proc sql noprint;
create table mgrs as
select distinct manager
from have;
quit;

data _null_;
set mgrs;
call execute(cat(
'data manager_',manager,';',
'set have;',
'where manager eq ''',manager,'"';',
'keep cust;',
'run;'

));
run;
```



HOT Tips from SAS-L

```
data _null_;
length dsets $200;
do until(last);
  set mgrs end=last;
  dsets = catt(dsets,' manager_',manager,
    '(where=(manager eq "','manager,'"))');
end;
call execute(cat('data ',dsets,'; set have;run;'));
run;
```

get more creative and write one data step that separates the observations with WHERE data set options



HOT Tips from SAS-L

EVEN
MORE
CREATIVE

USE A DATA
STEP HASH
OBJECT

```
data _null_;
declare
hash hid      ();
hid.definekey ('_n_');
hid.definedata ('cust');
hid.definedone ();

do _n_=1 by 1 until(last.manager);
  set have;
  by manager;
  hid.add();
end;

hid.output(dataset: cats('manager_',manager));
run;
```



HOT Tips from SAS-L

Secosky/Bloom: *Getting Started with the DATA Step Hash Object*

<http://www2.sas.com/proceedings/forum2007/271-2007.pdf>

Bloom/Secosky: *Getting Started with the DATA Step Hash Iterator*

<http://support.sas.com/rnd/base/dastep/dot/iterator-getting-started.pdf>

Ray/Secosky: *Better Hashing in SAS® 9.2*

<http://support.sas.com/rnd/base/dastep/dot/better-hashing-sas92.pdf>



HOT Tips from SAS-L

Loren: *How Do I Love Hash Tables? Let Me Count The Ways!*

<http://www2.sas.com/proceedings/forum2008/029-2008.pdf>

Dorfman/Shajenko: *Hash Crash and Beyond*

<http://www.nesug.info/Proceedings/nesug07/ff/ff03.pdf>

Hamilton: *Creating Data-Driven Data Set Names in a Single Pass Using Hash Objects*

<http://analytics.ncsu.edu/sesug/2007/SD04.pdf>

SAS 9 Hash Object Tip Sheet

<http://support.sas.com/rnd/base/datastep/dot/hash-tip-sheet.pdf>



HOT Tips from SAS-L

■ ANOTHER QUESTION

I want to infile (is this really a VERB) a text file with the format

```
var1424 var889 var1058 var0091  
1 2 1 2  
0 6 0 6  
2 9 4 2  
4 2 2 9
```

where the first line is the variable names and data starts on second line. I want to tell SAS to treat the 1st line as the variable names. The actual text file has many variables and is part of a macro program. The variable names embedded in the text file is essential. So I was wondering if there is an option in SAS that can automatically read-in the 1st line as variables names?



HOT Tips from SAS-L

```
* place variable list in a macro variable;
data _null_;
infile xyz obs = 1 ;
input;
call symput ( "varlist" , _infile_ ) ;
run ;

* make sas data set using the macro
variable;
data want;
infile xyz firstobs = 2 ;
input &varlist ;
run;
```

MACRO SOLUTION



HOT Tips from SAS-L

```
data _null_;
infile xyz obs=1;
input;
call execute(cat('data want; infile xyz
                  firstobs=2; input ',_infile_,';run;');
run;
```

CALL EXECUTE SOLUTION



HOT Tips from SAS-L

```
data _null_;
infile x obs=1;
input;
call execute(cat(
'data want;
infile x firstobs=2 dsd lrecl=500 pad;
input ',translate(_infile_," ",",",""),';
run;');
run;
```

READ A CSV FILE WITH A HEADER
RECORD (PROC IMPORT ALTERNATIVE
FOR NUMERIC DATA)



HOT Tips from SAS-L

■ ANOTHER QUESTION

I have a question to pull out some specific subject from the data set. I have a data set, the variables name started with same word like: "abc_aaa" "abc_bbb" "abc_ccc"..... "xyz_aaa" "xyz_bbb" "xyz_ccc".....

Right now, I want to pull out the subjects if abc_ equal to something and xyz_* equal to something. How can I write a simple code instead of writing all the variable name and if else sentence in my code? I want to test any of the variable equal to 2 or 3, how can I select it?*



HOT Tips from SAS-L

```
data want;
set have;
array x(*) _all_ ;
do i=1 to dim(x);
  if x(i) in (2,3) then
    filter=1;
end;
if filter;
drop i filter;
run;
```

USE AN ARRAY

```
data want;
set have;
if find(cats(of _all_), '2') +
  find(cats(of _all_), '3') ne 0;
run;
```

NO ARRAY, NO LOOP, NO
DROP, JUST CATS (EVEN WITH
NUMERIC DATA, NO LOG
MESSAGES)



HOT Tips from SAS-L

APPLICATION ...
FIND
OBSERVATIONS
WITH A SPECIFIC
DIAGNOSIS

```
data diabetes;
set all;
array dx(8);
do j=1 to 8 until (dia eq 1);
  dia = (dx(j) eq : '250');
end;
if dia;
drop j dia;
run;
```

```
data diabetes;
set all;
if find(catx(' ', ' ', of dx:), '*250') ne 0;
run;
```

THE 'EXTRA STUFF' IS USED TO FIND ONLY
DX CODES THAT BEGIN WITH 250



HOT Tips from SAS-L

LOOK FOR DIABETES, ASTHMA, HEART ATTACK AND
CREATE THREE INDICATOR VARIABLES

```
data dia_ast_ami;
set all;
dia = (find(catx('*', '*', of dx:), '*250') ne 0);
ast = (find(catx('*', '*', of dx:), '*493') ne 0);
ami = (find(catx('*', '*', of dx:), '*410') ne 0);
run;
```



HOT Tips from SAS-L

EASIER IF DIAGNOSES ARE ONLY THREE CHARACTERS

```
data dia_ast_ami;
set all;
dia = (findw(catx(' ',of dx:),'250') ne 0);
ast = (findw(catx(' ',of dx:),'493') ne 0);
ami = (findw(catx(' ',of dx:),'410') ne 0);
run;
```



HOT Tips from SAS-L

■ ANOTHER QUESTION

*I have a dataset set with two variables: date and return.
Altogether 20,000 obs.*

*I want to calculate rolling Standard deviation of returns on a
window of 750 observations, i.e. 1 to 750 obs; 2 to 751 obs,...*

Can you please assist me in writing a rolling SAS macros.

FIRST POSTED ANSWER: What makes you think a macro is needed? See PROC EXPAND (if you have SAS/ETS licensed).



HOT Tips from SAS-L

though a
macro
solution
is not
ideal

it is
possible

works,
but
takes a
bit of
time

```
%macro rollingstd;
proc datasets lib=work nolist;
delete stats;
quit;

%do start=1 %to 19251;
proc summary
data=test (firstobs=&start obs=%eval(&start + 749));
var x y;
output out=temp (drop=_:) 
std=stdx stdy;
run;
proc append base=stats data=temp;
run;
%end;
%mend;
```



HOT Tips from SAS-L

FIRST IDEA ... DO THE CALCULATIONS "MANUALLY"

PART 1 ... COMPUTE SUMS AND SUMS OF SQUARES FOR THE FIRST 750 OBSERVATIONS

```
data stds;
* first 750;
do j = 1 to 750;
set test point=j;
sumx + x; sumxsq + (x**2);
sumy + y; sumysq + (y**2);
end;
stdx = sqrt((sumxsq - (sumx**2/750))/749);
stdy = sqrt((sumysq - (sumy**2/750))/749);
output;
```



HOT Tips from SAS-L

PART 2 ...

REMOVE
ONE OBS,
ADD
ANOTHER

COMPUTE

< 1 SEC
CPU TIME

```
* increment sums (subtract 1 obs, add 1 obs);
do j = 2 to 19251;
* remove an observation;
jj = j-1; set test point = jj;
sumx + (-x); sumxsq + (-(x**2));
sumy + (-y); sumysq + (-(y**2));
* add an observation;
jj = j+749; set test point=jj;
sumx + x; sumxsq + (x**2);
sumy + y; sumysq + (y**2);
stdx = sqrt((sumxsq - (sumx**2/750))/749);
stdy = sqrt((sumysq - (sumy**2/750))/749);
output;
end;
stop;
keep stdx stdy;
run;
```



HOT Tips from SAS-L

ALTERNATIVE ...

PLACE VALUES IN AN
ARRAY USING THE MOD
FUNCTION TO KEEP
'REFRESHING' THE 750
VALUES IN THE ARRAY

```
data std(keep = std_x);
array xx(750);
retain xx: ;
set test;
xx(mod(_n_ - 1, 750) + 1) = x;
if _n_ >= 750 then do;
  std_x = std(of xx1-xx750);
  output;
run;
```

NEAT AND A LOT LESS SAS CODE (USES THE STD FUNCTION)

MUCH SLOWER THAN FIRST SOLUTION WHEN DATA SET IS
LARGE



HOT Tips from SAS-L

ALTERNATIVE COMBINING THE TWO IDEAS ... USE AN ARRAY, BUT AVOID THE FUNCTION AND DO THE CALCULATIONS "MANUALLY" ... NEAT AND FAST

```
data c(keep = std_x);
array xx(750) (750*0);
retain xx: ;
x2remove = xx(mod(_n_ - 1, 750) + 1);
set test;
xx(mod(_n_ - 1, 750) + 1) = x;
sumx + x - x2remove ;
sumxsq + x**2 - x2remove**2;
if _n_ >= 750 then do;
  std_x = sqrt((sumxsq - (sumx**2/750))/749);
  output;
end;
run;
```



HOT Tips from SAS-L

■ JUST TWO MORE (RECENT)

Is there a way to make a label that is a blank? I would like for the variable amd_flag below just show an empty box in my proc tabulate, but setting the label to a blank is actually removing the label altogether and showing "amd_flag" in my proc tabulate, whereas I actually want " "; just an empty box.

```
data mike;
x = 10;
* use a NULL hex value;
label x='00'x;
run;

proc print data=mike label;
run;

Obs
 1      10
```



HOT Tips from SAS-L

■ LAST ONE

Hi, I wanted to create two sets of variables from a SAS data set. One group is all char variables, the other is numeric. Can I do that with Proc Contents and how? Or is there other way to do it?

Later ... It appears from the later part of the thread that Peter wants comma-separated lists in macro variables.

DO YOU KNOW ABOUT THE VARIABLE `_FILE_` ?



HOT Tips from SAS-L

SOME 'POTENTIALLY NEW' CODE ... FILENAME DUMMY,
COMPRESS WITH A 3RD ARGUMENT, _FILE_

```
filename noseee dummy;
data _null_;
file noseee;
set sashelp.zipcode (obs=1);
call missing(of _all_);
put (_numeric_) (= ',' ) @;
call symput('nm',compress(substr(_file_,2),",_","kda"));
put;
put (_character_) (= ',' ) @;
call symput('ch',compress(substr(_file_,2),",_","kda"));
run;

1298 %put &nm;
ZIP,Y,X,STATE,COUNTY,MSA,AREACODE,GMTOFFSET
1299 %put &ch;
ZIP_CLASS,CITY,STATECODE,STATENAME,COUNTYNM,AREACODES,TIMEZONE,
DST,PONAME,ALIAS_CITY
```



HOT Tips from SAS-L

- OTHER QUESTIONS (SINCE I ANSWER A LOT OF SAS/GPLOT QUESTIONS, I HAVE TO INCLUDE A FEW)

I've a plot where x is a time variable (in days) and y is a certain metric that assumes values between -100 and 100. I'd like to create a plot line with GPLOT and I need to fill with a pattern any areas where y is below 0. In other words, I'd like no fill for points above y=0 and a pattern of vertical lines for the closed area between 0 and negative values of y. Not sure if this matters, but the plotted line may intersect y=0 in multiple instances.

ANSWER: data tweak plus AREAS option with GPLOT



HOT Tips from SAS-L

I have the following PC SAS 9.1.3 code:

```
symbol1 interpol=join;  
<more>  
proc gplot data=temp_history;  
plot &var1*.*sasdate/haxis=axis1 vaxis=axis2 vref=&lclp &p  
&uclp;  
run;
```

But what I really want to do is fill in with light grey between the upper and lower reference lines. Is this possible? How?

ANSWER: annotate data set with BAR



HOT Tips from SAS-L

I need to get rid of the tick marks in the "HILO" symbol interpolation option, so that I just have a line joining the upper and lower values. I just can't see how to.....

ANSWER: don't use HILO, change the PLOT statement



HOT Tips from SAS-L

 sasCommunity.org

search

main menu

- [Main Page](#)
- [Blogger's Corner](#)
- [Events](#)
- [Forums](#)
- [Geographies](#)
- [Opportunity Registry](#)
- [Presentations](#)
- [Provider's Registry](#)
- [Quick Links](#)
- [SAS Books](#)
- [Sasopedia](#)
- [Talk to Us](#)
- [User Groups](#)

article discussion edit history

SAS-L

NOTE: This article is a work in progress and the changes here may be rolled back and moved to another page. The intent of this update is to provide some insights on how to post to SAS-L and how to leverage sasCommunity.org in a way that benefits both forms of communication/collaboration.

Contents [\[hide\]](#)

- [1 Overview of SAS-L](#)
- [2 Posting a Question/Problem to SAS-L](#)
- [3 Answering Questions Posted on SAS-L](#)
- [4 Ideas from the Original Note](#)

Overview of SAS-L [\[edit\]](#)

As Howard Schreier ([User:Howles](#)) once put it, "SAS-L is a worldwide online community of SAS software users; see [Ask This Old Newsgroup: Using SAS-L Effectively \(PDF\)](#). Participants discuss various aspects of SAS and help one another solve SAS-related problems."



HOT Tips from SAS-L

MY EMAIL ADDRESS ...

msz03@albany.edu

MY WEB PAGE ...

<http://www.albany.edu/~msz03>



STATUS SYMBOL



HOT Tips from SAS-L

MY SASCOMMUNITY.ORG PAGE ...

<http://www.sascommunity.org/wiki/User:Msz03>





HOT Tips from SAS-L

!!! THANKS FOR THE INVITATION !!!



HOT Tips from SAS-L

USE PROC SUMMARY AND IDGROUP TO TRANSPOSE TWO VARIABLES

Obs	id	month	chol	weight
1	1	1	200	180
2	1	2	190	175
3	1	3	185	160
4	2	1	180	150
5	2	2	179	155
6	2	3	175	140

```
proc summary data=one nway;
class id;
output out=two (drop=_type_ _freq_) idgroup(out[3](chol weight)=);
run;
```

id	chol_1	chol_2	chol_3	weight_1	weight_2	weight_3
1	200	190	185	180	175	160
2	180	179	175	150	155	140



HOT Tips from SAS-L

COMPARE TWO DATA SETS ... KEEP ONLY
OBSERVATIONS THAT ARE NOT IN BOTH DATA SETS

```
proc sql noprint;
  create table data12 as
    select * from data1 union select * from data2
  except
    select * from data2 intersect select * from data1;
quit ;
```