

VIEWS

NEWS

Issue 4 Autumn 1998

News, Hints, Tips, and information for UK SAS Users

The SAS® System has its roots in statistical analysis and, subsequently, mainframe performance analysis. For many years, however, it has been a popular tool for application development. The desire to build front-ends to statistical procedures led to an industrial strength application development environment.

The subject of VIEWS' third sub-group conference is applications development. The conference is to be held in mid-November and was sold-out weeks before the event. As always, a Books By UsersSM (BBU) author will be the focus of the day. Don Stanley has been flown in from New Zealand especially for the event.

Given the fast pace of change in the applications development environment, it is especially appropriate for Don to share the day with SAS Institute's European Manager of Applications Development Products - Rob Owen.

It seems like a day doesn't go by without somebody saying to me that SCL is dead and Java will be the new application development language. Well, I beg to disagree. Whilst it may be true that Java (and its competitors) will begin to take a significant proportion of front-end development in the coming years, the back-end code will continue to be written in base SAS® software and SCL. That's where SAS software's computing strength comes to the fore and the likes of Java are in no position to compete.

Not all of the front-end development will be stolen either. There are countless cases where an application does not justify a) using a client/server solution with its associated costs, or b) using more than one language and supplier. The attendees of the November sub-group will be learning tips and techniques that will stand them in good stead for years to come. And that's what VIEWS is all about - sharing knowledge throughout the UK SAS community.

Andrew Ratcliffe (Editor)

The Consultant

This part of VIEWS is where you can get your technical questions answered. Send your questions to the Editor.

Split Axis

Q: I have response data (y-axis) in the range of 600 to 1000 units. I have been told that it is possible that there could be a '0' response and so I want the graph to indicate that. If I change the y-axis to a 0 to 1000 scale all of the data is clustered at the top of the graph. What I would like is to start the y-axis at 0 and then use a // break in the y-axis and then continue the axis from 600; so that the area between 0 and 600 would be eliminated.

A: Here's a solution that uses Annotate. No, don't skip to the next item! It works very simply by using an AXIS statement to define the Y axis and then using annotate to draw a pink line over the part of the axis we don't want. Then, we use annotate to draw two more lines as our slashes.

```
data fred;                                * Generate test data ;
do x=0 to 10;
  y=600+ranuni(0)*400;
output;
end;
run;

%annomac;                                * Declare/compile annotate macros ;
```

```
options cback=pink ftext=swissb ctext=yellow;

options device=imgjpeg gaccess=gsasfile; * JPEG o/p file ;
filename gsasfile 'd:\views\newsletter\issue04\split.jpg';

title1 'Demo of a Split Axis';

data anno;
%dcanno;                                * Declare annotate variables ;
%system(1,2,.);                        * X=%data, Y=abs data ;
when='a';                              * Draw anno after graph is drawn ;
%line(0,250,0,350,pink,.,.);           * Hide y-axis ;
%line(-3,300,3,200,white,.,.);         * Lower slash ;
%line(-3,400,3,300,white,.,.);         * Higher slash ;
run;

axis1 order=(0 50 550 to 1000 by 50);
symbol1 i=join v=plus;
proc gplot data=fred anno=anno;
  plot y*x/vaxis=axis1;
run;quit;
```

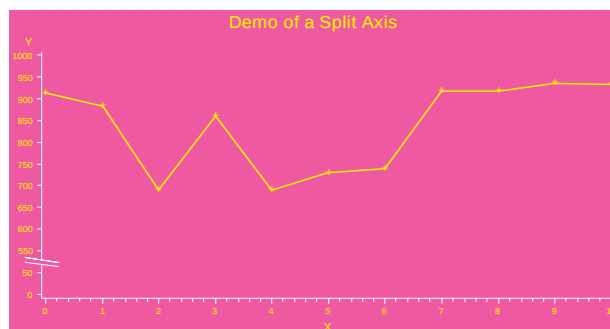
The SYSTEM macro sets the co-ordinate systems in the X and Y direction and the height (size) of things. Our invocation says we want to use percentages of the data area in the X direction, data values in the Y direction, and the default co-ordinate system for height.

Our lower slash consists of a line drawn from (-3%,300) to (+3%,200). Using percentages in the X direction allows our code to be independent of the data values, i.e. it doesn't matter what the range of the X axis is.

A few other things to note include the use of WHEN to tell annotate to draw its objects after the graph is complete - this allows annotate to place the pink line over the top of the vertical axis where required. The log will show notes about the negative percentages and the use of pink on a pink background, and a warning about the intervals on the axis not being evenly spaced.

If you do have zero values in your data the join line will still be continuous - you'll need to put more work into annotation.

Andrew Ratcliffe



SAS/CONNECT® Window Title

Q: When my users sign-on to a remote host using the SAS/CONNECT log-in screen, the title of the little screen is always "Type Window". Is there any way I can change this please?

A: Yes, you can. To change the text, add the following to your code

```
%let cstyle=Enter Login;
```

This will change the text to "Enter Login".

Andrew Ratcliffe

Did You Know?

If you have a useful hint or tip, send it to the Editor and share it with the VIEWS membership.

Not Running SAS Code

There are many occasions where you need *not* to run a large block of SAS code. Simple solutions include preceding it with a /* and following it with an */. However, if your code already includes a /*...*/ style comment then that solution won't work. If it's a big block of code (that maybe you didn't write

yourself) then visually checking the section can be unreliable.

An alternative to the above is to surround the block of code with a macro definition, e.g. precede the block with `%macro dummy;` and follow the code with `%mend dummy;`.

Here's one more tip on the subject. The RUN statement can be coded with the keyword CANCEL immediately following it. If you do this, SAS software will not run the DATA step or PROC, but it will check the syntax of your statements. An example follows.

```
data chewing;
  set gum;
  x = x**2;
run cancel;
NOTE: Data step not executed at user's request.

proc datasets lib=work nolist;
  delete gum;
run cancel;
NOTE: The procedure was not executed at user's request.
quit;
```

Andrew Ratcliffe

Recording SAS/INSIGHT® Sessions

In the latest versions of SAS/INSIGHT® software you are able to record and later resubmit commands from your SAS/INSIGHT® session. This was an undocumented feature in earlier versions such as V6.08 and V6.09. The FILE and INFILE options allow you to produce a file containing commands to document and reproduce a SAS/INSIGHT® session. It's great for exploratory analysis that you need to interrupt or repeat on different sets of data. An example follows.

```
filename note 'insight.txt';
proc insight file=note;
run;
```

After doing your analysis and then exiting, your file will contain the commands that were used to create and close your Insight session. Edit the file and delete the line(s) at the end containing FILE:END statements. You can begin a second Insight session from where you left your first session with the following code.

```
filename note 'insight.txt';
proc insight infile=note;
run;
```

Alternatively, just code the FILE keyword without a file name specified. If you do that then your commands will be recorded in the SAS Log window.

Andrew Ratcliffe

Footnote Alignment

Ever had a problem with lining up footnotes that contain macro variables? This is a problem whenever a procedure is used to produce some output that requires footnotes - PRINT or GLM for example. The following gets over the problem:

```
footnote1 "%sysfunc(putc(NOTE: (*) = Value outside the reference
range(not clinically significant),$102.))";

footnote2 "%sysfunc(putc((#) = Value outside the reference
range(clinically significant),$90.))";

footnote3 "%sysfunc(putc(NPF = No pathological findings,$90.))";

footnote5 "%sysfunc(putc(Data: &datemade &timemade,$90.))";

footnote6 "%sysfunc(putc(Program: &mdir\&progname
&sysdate2&tdnew,$90.))";
```

%SYSFUNC() allows SCL functions to be used in the macro language and therefore base SAS software. PUTC() is a base/SCL function, which puts out character strings. Using them together enables us to do a PUT statement within a title or footnote, giving control of the output.

If the line size of your page is 130 and you use a format of 90 (as in footnotes 2 to 6) then the footnotes will be indented by half the difference (20), as 90 will be centralised within the 130. In the above example the first footnote needs to start 6 spaces to the left to allow for "NOTE: ". In order to compensate for this

a format of 90 + double this difference needs to be used, i.e. 102.

Martin Johnson

Call Execute

You can use CALL EXECUTE to set-up SAS code that will be executed after a data step ends. This is similar to the traditional method of putting statements out to a temporary file and then %INCLUDE-ing them into the SAS stream. A simple example follows.

```
18 data _null_;
19   call execute('data y;set ');
20   call execute('sasuser.crime;run;');
21   call execute('proc options option=linesize;run;');
22
23   call execute('%put &message;');
24   call symput('MESSAGE','Hello readers');
25 run;
```

```
WARNING: Apparent symbolic reference MESSAGE not resolved.
&message
NOTE: The DATA statement used 0.04 seconds.
```

```
NOTE: CALL EXECUTE generated line.
1  +data y;set
NOTE: CALL EXECUTE generated line.
2  +sasuser.crime;run;
```

```
NOTE: The data set WORK.Y has 50 observations and 9 variables.
NOTE: The DATA statement used 0.12 seconds.
```

```
NOTE: CALL EXECUTE generated line.
3  +proc options option=linesize;run;
```

SAS (r) Proprietary Software Release 6.12 TS045

LINESIZE=96 Line size for printed output

```
NOTE: The PROCEDURE OPTIONS used 0.09 seconds.
```

```
NOTE: CALL EXECUTE generated line.
4  +
```

Notice that you may split statements and code as many steps as you wish. Note also that the generated lines begin a new numbering sequence in the log - this can be disconcerting when you first see it.

This technique can be useful in situations where you want to avoid writing a macro or creating a temporary external file, though it can make your code a little less easy to read.

One significant drawback I've found is that macro references are resolved immediately, not when the code is run. In the example above you can see that the %PUT statement was resolved before the first DATA step had completed.

Andrew Ratcliffe

Targeted Help

I thought I would like to share this tip for SAS V6.12 that I discovered the other month with help from who else but Phil Mason. I was becoming frustrated at having to view several help screens before arriving at what is now called the "Limited Index" or what used to be called the main help screen in versions before 6.12.

I posted messages to every Usenet group that had anything to do with SAS software or statistics. No Joy. I plucked-up enough courage to contact Phil and he promptly replied pointing me in the direction of the WINHELP() function.

It was with this help that I discovered that if I changed the code behind the F1 function key to:

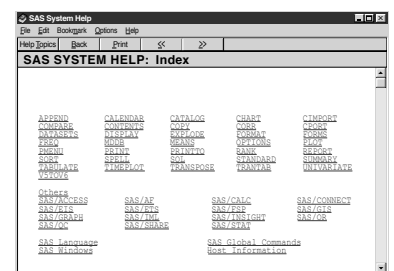
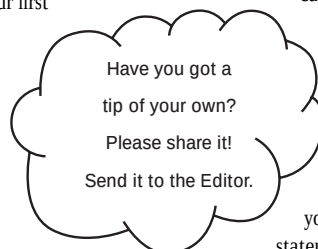
```
help c=sashelp.sasdoc.sasdoc.cbt frame=index
```

up popped my favourite help screen.

Andrew Davey

Incrementing A Character Value

When you have a lot of names to produce you may wish to create unique



alphanumeric names for each of them. The easiest way is to suffix them with a unique number, however you may want to use the full range of letters and numbers. The following code does this very simply.

```
data _null_;
  string='VIEWS99';

  do ptr=length(string) to 1 by -1
    until (substr(string,ptr,1) ne 'A');
    substr(string,ptr,1) = translate(substr(string,ptr,1)
                                     , 'BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A'
                                     , 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
                                     );
  end;
  put string=;
run;

STRING=VIEWTAA
```

The translate() function is used to increment the last character of the name: B is translated to C, etc. When appropriate, 9 is translated to A and this signifies that the penultimate letter must be incremented too. The loop with the UNTIL clause controls how many of the characters need to be incremented.

Andrew Ratcliffe

News

Master Class Training

Just over a year ago, SAS Institute introduced a new concept in training in Europe, namely Master Class.

Master Class is an intensive, goal-orientated training course for programming and applications development with the SAS System. In approximately six weeks, students submit themselves to theoretical training, extensive hands-on exercises, an in-depth case study, and examinations. The final exam consists of developing a prototype application

Master Class has been particularly popular in emerging market countries where there is a comparatively lower SAS software skill base. To date, this has been less of a requirement here in the UK.

However, SAS Training in the UK will be in a position to schedule Master Class courses for appropriate audiences from early 1999. If you would like to see this happen, contact Stephanie Penning (SAS UK Training Manager) to discuss your needs in detail (see *Contacts*).

Books Tout SAS Institute

Read any good books lately? Check-out *Contented Cows Give Better Milk* and *Good Company: Caring as Fiercely as You Compete*, both of which highlight the management philosophies and practices of SAS Institute.

Good Company co-authors Hal Rosenbluth and Diane McFerrin Peters selected 15 companies from the book, *The 100 Best Companies To Work For In America*, for inclusion in *Good Company*. Their book asserts that those companies who are "employers of choice" will be the most successful organisations, attracting and retaining the finest talent.

Similarly, *Contented Cows Give Better Milk*, written by Bill Catlette and Richard Hadden, establishes a convincing link between an organisation's employee relations practices and its bottom line. To illustrate their concept, Catlette and Hadden include a chart depicting 10 years of SAS Institute's solid revenue growth. Wow, never seen that before!

New BBUSM

Additions to the ever increasing Books By Users (BBU) library are:

- *Carpenter's Complete Guide to the SAS Macro Language* (Order No. AB56100) by Arthur L. Carpenter
- *Efficiency: Improving the Performance of Your SAS Applications* (Order No. AB55960) by Robert Virgile
- *Professional SAS Programmer's Pocket Reference, Second Edition* (Order No. AB56646) by Rick Aster
- *Programming Techniques for Object-Based Statistical Analysis with SAS Software* (Order No. AB55869) by Tanya Kolosova and Samuel Berestizhevsky
- *Strategic Data Warehousing Principles Using SAS Software* (Order No. AB56278) by Peter R. Welbrock

- *Your Guide to Survey Research Using the SAS System* (Order No. AB55688) by Archer R. Gravely

New Download and Demos

Recent additions and updates to SAS Institute's list of downloads and demos on its web site includes:

- Add-in tools for SAS/Warehouse Administrator™ software
- Java Tools and MetaSpace Explorer for SAS/IntrNet™ software
- Enterprise Miner™ documentation (version 2.0)

In Brief

- Use %SYSGET to get the value of DOS environment variables and -SET values from CONFIG
- If you want to display the value of all global macro variables, try %put _global_;
- *VIEWS News* is now available at the VIEWS web site as a PDF file
- The Dutch quality partner PW Consulting have a very useful web site at <http://www.pwcons.com>. It includes tips and a range of SAS/AF® shareware and freeware objects
- If *VIEWS News* doesn't provide sufficient information for you, try the SAS-L archive at <http://jse.stat.ncsu.edu:70/11/othergroups/sasl>
- The Summer issue of *VIEWS News* was distributed to over 700 UK SAS users

Diary

Are you organising an event that would be of interest to the VIEWS readership? Let us know, as we are interested in all non profit making events related to SAS software.

November 1998

18th The VIEWS Applications Development sub-group conference

February 1999

1st Fifth edition of *VIEWS News*

? The VIEWS Graphics sub-group conference (provisionally sometime in the 1st quarter of 1999)

April 1999

11-14 SUGI, Miami Beach, Florida

May 1999

1st Sixth edition of *VIEWS News*

June 1999

15-18 SEUGI, Den Haag, The Netherlands

July 1999

8th VIEWS 5 conference, Earls Court, London (provisional)

Contacts

Contributors and Contacts

Andrew Davey	Email: andy@andyweb.demon.co.uk
Martin Johnson	Email: martin.johnson@covance.com Covance, Leeds
Stephanie Penning SAS UK Training Manager	Tel: 01628-486933 Fax: 01628-483203 Email: suksxp@suk.sas.com

VIEWS

Web Site	http://www.views-uk.demon.co.uk
Phil Mason Chairman	Tel: 01491-834615 Fax: 01491-834615 Email: phil_mason@email.com 16 Wood Street, Wallingford, Oxfordshire. OX10 0AY
Bruce Bovill SAS Liaison	Tel: 01628-404347 Fax: 01628-404303 Email: sukbnb@suk.sas.com
Andrew Ratcliffe Newsletter Editor	Tel: 01322-525672 Fax: 01322-525672 Email: andrew@ratcliffe.demon.co.uk

SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASA**SAS**@**AS**ASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA
SASASASAS@ASASASASA

Looking for a contract? Or looking for quality people?
Call us now - we know SAS® software inside out!

Official sponsors of



<http://www.asaltd.co.uk>



ASA Ltd. First Floor, East Wing, Auckland House, New Zealand Avenue, Walton Upon Thames, Surrey KT12 1PL.
Contact Richard Field at ASA on Tel: 01932 701801 Fax: 01932 701800 Email: sas@asaltd.co.uk

SAS is a registered trademark of SAS Institute Inc. Cary, North Carolina, USA.