

VIEWS

International SAS Programmer Community

News, Hints, Tips and Information for SAS® Users

Issue 46, 2nd Quarter 2009

In this issue of *VIEWS News* I have some more articles from new and familiar sources. I would like to welcome 3 new *VIEWS* Consultants Sanjay Matange with his article on the new SG Procedures in SAS 9.2, Murphy Choy for his article on text editors for SAS programming and SAS Global Forum report and Ted Feuerbach for his report on 2 PhUSE Single Day Events in Frankfurt and Brussels. Our regular contributor LeRoy Bessler demonstrates how to link Excel spreadsheets to SAS graphs, and I have replaced my regular Formats, Options, and Functions section with an article on some SAS functions from the Amadeus Software Consultancy Team.

I presented at the SAS Global Forum in Washington DC and 2 PhUSE Single Day Events in Brussels and Frankfurt earlier this year, and thought you would prefer to read unbiased reviews of these events. New *VIEWS* Consultants Murphy Choy and Ted Feuerbach have kindly provided their personal views on the events. You will also be able to see me at the PhUSE/PSI One-Day Event in Marlow in June, so please come and tell me what you would like to see in future issues of *VIEWS News*.

I'm hoping Murphy has started a series of articles about your favourite text editor for SAS programming. If you would like to contribute an article, to re-visit and improve an existing article, or just discuss the possibility of doing so, please send an email to me at newsletter@views-uk.org. A list of subject suggestions for your articles can be found on the *VIEWS* web site, and anyone is very welcome to add to that list by sending emails to newsletter@views-uk.org with your own questions.

Philip R Holland (Newsletter Editor)

Membership of *VIEWS* is **free** and you can register for notification of all future *VIEWS* events by emailing us at membership@views-uk.org, making certain you include your name and postal address. If you would like to receive email notification whenever a new issue of the free quarterly *VIEWS News* is published on the web site, please remember to also include your email address.

Did You Know?

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

Visual + Detail = Effective Communication: Web-Enabled SAS/GRAPH + Spreadsheet

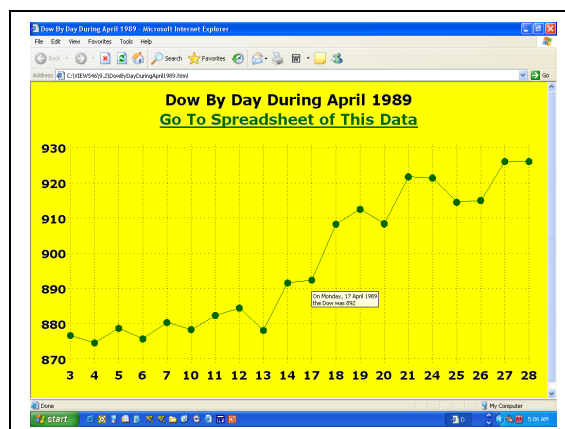
[SAS 9.2] Graphs facilitate and accelerate communication and decisions, but detail is needed for reliable communication and decisions. When data is prepared with SAS software, the ultimate user often insists on having it delivered via an Excel

spreadsheet. Then, its next destination might be a regrettable Excel graph.

Here I will show you how to package your results so that the user can view it as a maximally communication-effective web graph, but can toggle to a spreadsheet to inspect the supporting data as a unit, and manipulate it further non-graphically with Excel. For someone wanting data in Excel, but appreciating a better and finished graph, this is the best deliverable.

Web deployment permits embedding of hyperlinks in a graph, plus ALT text, also variously called pop-up text, flyover text, float over text, hover text, or a tool tip. A graph most likely to benefit from ALT text is a plot. Even with tick marks and/or reference/grid lines, it is impossible to know the precise y value for a plot point. With ALT text, simply resting the mouse on a plot point displays its x and y values to any precision you choose to provide, in any format you like, and with whatever description you like for the x and y variables.

Here is the graph web page, with the mouse resting on a point:



Here is an enlargement of the ALT text that popped up:

On Monday, 17 April 1989
the Dow was 892

HTML does not support specification of font, size, or colour for this ALT text, only line breaks.

Here is the spreadsheet after being linked to from the graph:

Date	Date of Observation	Dow
1	Monday, 3 April 1989	871
2	Tuesday, 4 April 1989	875
3	Wednesday, 5 April 1989	876
4	Thursday, 6 April 1989	879
5	Friday, 7 April 1989	883
6	Saturday, 8 April 1989	883
7	Sunday, 9 April 1989	883
8	Monday, 10 April 1989	883
9	Tuesday, 11 April 1989	883
10	Wednesday, 12 April 1989	884
11	Thursday, 13 April 1989	879
12	Friday, 14 April 1989	892
13	Saturday, 15 April 1989	892
14	Sunday, 16 April 1989	892
15	Monday, 17 April 1989	892
16	Tuesday, 18 April 1989	908
17	Wednesday, 19 April 1989	913
18	Thursday, 20 April 1989	908
19	Friday, 21 April 1989	922
20	Saturday, 22 April 1989	922
21	Sunday, 23 April 1989	922
22	Monday, 24 April 1989	924
23	Tuesday, 25 April 1989	915
24	Wednesday, 26 April 1989	915
25	Thursday, 27 April 1989	928
26	Friday, 28 April 1989	928

Here is an enlarged picture of the upper left corner:

Address  C:\VIEWS46\9.2\DowByDayDuringApril1989.xls			
IDX 			
	A	B	C
1		Dow By Day During April 1989	
2		Go To Graph of This Data	
3			
4	Obs	Date of Observation	Dow
5	1	Monday, 3 April 1989	877
6	2	Tuesday, 4 April 1989	875

Here is the code used:

```
PROC TEMPLATE;
  EDIT Styles.Minimal AS MinimalYellow;
  STYLE Document / BACKGROUND = CXFFFF00;
  STYLE Container / BACKGROUND = CXFFFF00;
END;
RUN;

DATA work.ToReport
  (KEEP = date snydjcm Day AltTextVar);
  LENGTH AltTextVar $ 200 Day $ 2;
  SET sashelp.citiday
    (KEEP=date snydjcm
     WHERE=(YEAR(date) EQ 1989
            AND MONTH(date) EQ 4));
  AltTextVar = 'alt="On ' ||
    TRIM(LEFT(PUT(date, WEEKDATX.))) ||
    '0D'X ||
    'the Dow was ' ||
    TRIM(LEFT(PUT(snydjcm, 5.))) || "'";
  /* '0D'X forces a line break */
  Day = PUT(DAY(date), 2.);
RUN;

%LET Path = C:\VIEWS46\9.2;
%LET Title = Dow By Day During April 1989;
%LET GraphDescription = &Title.;
%LET BodyFileName =
  %SYSFUNC(COMPRESS(&Title.));
%LET GraphFileName =
  %SUBSTR(&BodyFileName, 1, 8);
/* See note later about filename length */

ODS NORESULTS;
ODS LISTING CLOSE;

GOPTIONS RESET = ALL;
GOPTIONS DEVICE = GIF;
/* In V9.2,
  ODS HTML would select PNG as default. */
GOPTIONS TRANSPARENCY;
/* LET web page background show through. */
/* PNG does not support transparency. */
GOPTIONS XPIXELS = 980 YPIXELS = 560;
/* size to fit in a 1024 X 768 screen
  with an extra tool bar */
GOPTIONS FTEXT = 'Verdana/Bold' HTEXT = 4PCT;

PROC CATALOG CAT = work.gseg KILL;
RUN;
QUIT;

ODS HTML PATH = "&Path." (URL = NONE)
  BODY = "&BodyFileName..html"
  (TITLE=&Title)
  STYLE = MinimalYellow;
PROC GPLOT DATA = ToReport;
```

```
TITLE FONT = 'Verdana/Bold' HEIGHT = 5PCT
  "&Title."
JUSTIFY = CENTER COLOR = CX006600
UNDERLIN=2 LINK="&BodyFileName..xls"
'Go To Spreadsheet of This Data';
SYMBOL1 INTERPOL = JOIN LINE = 1 WIDTH = 1
  VALUE = DOT HEIGHT = 1.5
  COLOR = CX006600;
AXIS1 LABEL = NONE MAJOR = NONE MINOR = NONE
  STYLE = 0;
PLOT snydjcm * Day / NAME = "&GraphFileName."
  DESCRIPTION = "&GraphDescription."
  HTML = AltTextVar NOFRAME
  VAXIS = AXIS1 HAXIS = AXIS1
  AUTOVREF LAUTOVREF = 33 WAUTOVREF = 1
  AUTOHREF LAUTOHREF = 33 WAUTOHREF = 1;
/* WAUTOVREF & WAUTOHREF new in SAS 9.2 */
RUN;
QUIT;

ODS HTML CLOSE;

GOPTIONS RESET = ALL;
/* Prior GOPTIONS can affect PROC PRINT. */

/* Use COLSPAN= so that TITLE lines do not
  get put only into column 1 and expand its
  width to length of longest TITLE line.
  Spread them across four columns. */

ODS HTML PATH = "&Path." (URL = NONE)
  BODY = "&BodyFileName..xls";

TITLE JUSTIFY = LEFT
  "<td COLSPAN=4>&Title.</td>"
JUSTIFY=LEFT "<td COLSPAN=4>
<a href='\" " &BodyFileName..html" '\">
Go To Graph of This Data</a></td>";

PROC PRINT DATA = ToReport LABEL;
  VAR date snydjcm;
  LABEL snydjcm = 'Dow';
  FORMAT date WEEKDATX. snydjcm 5.;
RUN;

ODS HTML CLOSE;
```

SAS 9.2 was used here. The results for SAS 9.1.3 would be identical, except that the fonts would not be as thickly drawn, and these HEIGHT=1 plot symbols would be smaller. If you do not specify DEVICE=, then ODS HTML in SAS 9.1.3 selects GIF, but in SAS 9.2 selects PNG, which does not support transparency, which is used to let the web page background show through the graph. Also, SAS 9.2 permits thickening reference lines. However, line type 33 becomes very thick short dashes at a thickness of 2.

NOTE: In SAS 9.1.3, the graph file name assigned in NAME= on the PLOT (or HBAR, or PIE, etc.) statement is limited to eight characters. In SAS 9.2, the limit is expanded to 256 characters. However, if in the same program/session you use more than one long filename that all start with the same initial eight characters (e.g., "First8ch"), the second and subsequent file names are automatically renamed to the following names: First8chRestOf2ndFileName1, First8chRestOf3rdFileName2, etc. This problem is fixed in SAS 9.2 TS2M0.

LeRoy Bessler, Assurant Health

Correction: In Issue 45 of VIEWS News, I incorrectly stated that in SAS 9.2 the PNG device driver supports transparency. It does not. – LeRoy Bessler, Assurant Health

Data masking in SAS

Sensitive information is commonly found in large databases which deal with consumer information. However, with more and more data being collected, sensitive personal information can now be found in many places such as banks and even your local gym. The presence of such sensitive data has made it very difficult for analysis of data without revealing sensitive information to the analysts. To combat this problem, data masking has been used frequently to allow sensitive information to be encrypted, avoiding the risk of data leakage.

Sensitive information in consumer data has always been a problem in analytics and is the target of many legislations and acts. With the Internet becoming more integral to our life, sensitive information is found even in social network databases and places like gyms. The nature of such data makes it critical that some form of protection is available. This problem is compounded by the fact that analysts often have direct access to these database and information. To avoid potential legal issues, data masking is one of the ways to allow for data analysis without compromising consumer confidentiality.

Data protection is a critical component of the analytics business and it is sometimes a requirement for companies in certain countries where information laws are very strict. Examples of such data protection acts and regulations include Data Protection Act 1998 (UK), EC directive 94/95(EU) and HIPEA (US). To ensure that sensitive information are not leaked and used by unauthorized parties in possibly illegal activities, it is essential that protection in the form of data masking be applied to all data. Data masking makes it more difficult for unauthorized parties who might have accessed the data illegally to obtain critical information. Data masking also allow companies which require analytics services to be sure that their data will not be used by the service providers for other purposes.

There are three major techniques of data masking that are widely used; nullification, substitution and encryption.

The following data set will be used in all of the examples below:

```
DATA credit_card;
  INPUT card_no :$20. name $ 17-35;
  DATALINES;
4211569111000001 AMY LEVINE
4211569111000002 SHANE LENNINGS
4211569111000003 EVANS TEDDINGTON
4211569111000004 MARC RODRIGUEZ
;
RUN;
```

Nullification is the process of neutralizing certain sections of the data by way of substituting the information with a single null value. Usually the null value used is X or an empty space, but there are other choices. Below is a snapshot of a program that does nullification:

```
/******
NULLIFICATION EXAMPLE
******/
DATA credit_card1;
  SET credit_card;
  new_card_no = card_no;
  DO i = 1 TO 12;
    SUBSTR(new_card_no, i, 1) = 'X';
  END;
```

```
DROP i;
PUT card_no = new_card_no =;
RUN;
```

Output:

```
CARD_NO=4211569111000001
NEW_CARD_NO=XXXXXXXXXXXX0001
CARD_NO=4211569111000002
NEW_CARD_NO=XXXXXXXXXXXX0002
CARD_NO=4211569111000003
NEW_CARD_NO=XXXXXXXXXXXX0003
CARD_NO=4211569111000004
NEW_CARD_NO=XXXXXXXXXXXX0004
```

We can see from the output on the log that the first 12 digits have been nullified. Without the first 12 digits, the credit card information is essentially useless for anyone. Thus the value of the information has been fully neutralized by the process of nullification.

Substitution is the process of replacing a section of the information with pseudo information. Usually a fixed length of the information will be replaced a randomly generated number; this number is fixed throughout for all the values. Below is a program showing how this is done:

```
/******
SUBSTITUTION EXAMPLE
******/
DATA credit_card2;
  SET credit_card;
  new_card_no = card_no;
  SUBSTR(new_card_no, 1, 8) = '12345678';
  PUT card_no = new_card_no =;
RUN;
```

Output:

```
CARD_NO=4211569111000001
NEW_CARD_NO=1234567811000001
CARD_NO=4211569111000002
NEW_CARD_NO=1234567811000002
CARD_NO=4211569111000003
NEW_CARD_NO=1234567811000003
CARD_NO=4211569111000004
NEW_CARD_NO=1234567811000004
```

We can see from the output on the log that the first 8 digits have been substituted by a random value. Without the first 8 digits, the credit card information is essentially useless for anyone. Thus the value of the information has been fully neutralized by substitution. However, we have to exercise caution as an insider who knows the structure of the credit card might be able to reverse the substitution.

Encryption is the process of masking the information via an algorithm that renders it impossible to retrieve the original information without the key. The original information is usually transformed by a bit operation with a key. Below is a snapshot of the program:

```
/******
ENCRYPTION EXAMPLE
******/
DATA credit_card3;
  SET credit_card;
```

```

new_card_no = COMPRESS(SUBSTR(card_no, 1, 8)
|| BXOR(INPUT(SUBSTR(card_no, 9, 8), 8.),
'12345678'));
PUT card_no = new_card_no =;
RUN;

```

Output:

```

CARD_NO=4211569111000001
NEW_CARD_NO=421156911816975
CARD_NO=4211569111000002
NEW_CARD_NO=421156911816972
CARD_NO=4211569111000003
NEW_CARD_NO=421156911816973
CARD_NO=4211569111000004
NEW_CARD_NO=421156911816970

```

From the results, we can see that the card numbers look similar to the original ones. However, the operation has actually reduced the length by 1 digit. Thus the information is now useless even though we have the first 8 digits. The neutralization of the information is an excellent protection as a careless criminal will not notice the difference.

Data masking is a simple and yet crucial process in data warehousing and analytics. SAS provides many different options for users who wish to mask their data.

Murphy Choy, University College Dublin

Why should I use SG Procedures?

[SAS 9.2] SAS version 9.2 includes a new set of procedures to create modern statistical graphs. These include the SGPLOT, SGPANEL and the SGSCATTER procedures.

Each procedure is specialized for specific use cases:

- SGPLOT – Single-cell graphs
- SGPANEL – Classification panel graphs
- SGSCATTER – Comparative graphs & matrices

These procedures can be valuable tools in the graph programmer's toolbox, and can provide the ability to create graphs that otherwise may need a lot of coding. This series of short articles will present the capabilities of these new procedures by example, starting with the SGPANEL procedure.

The SGPANEL procedure has the following syntax:

```

PROC SGPANEL DATA = data-set;
  PANELBY class-vars / options;
  plotname reqd-params / options;
RUN;

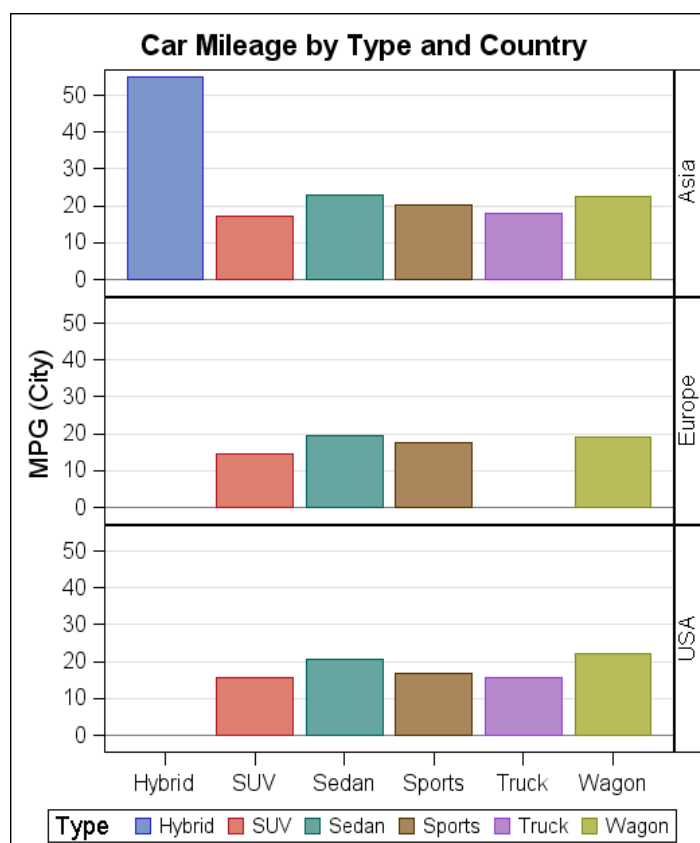
```

Let us use this procedure to create a graph of car mileage by type and country of origin. The entire code required to create this graph is:

```

TITLE 'Car Mileage by Type ...';
PROC SGPANEL DATA = sashelp.cars;
  PANELBY origin / LAYOUT = ROWLATTICE;
  VBAR type /
    RESPONSE = mpg_city
    GROUP = type STAT = MEAN;
  ROWAXIS GRID;
  COLAXIS DISPLAY = (NOLABEL);
RUN;

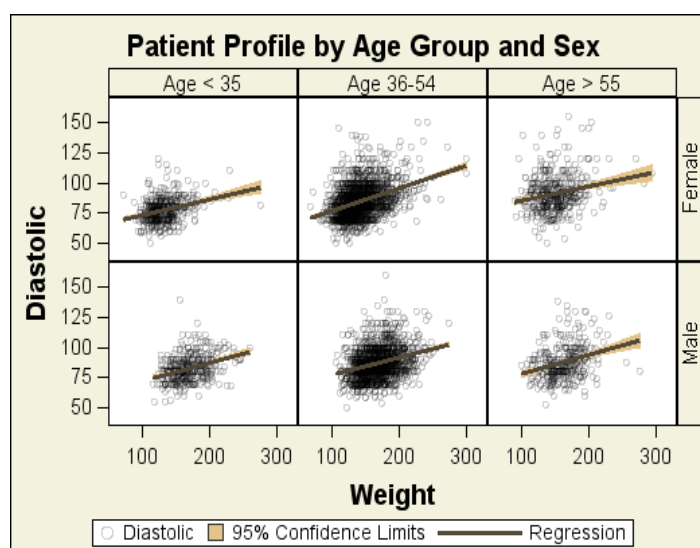
```



A similar graph could be done using the traditional SAS/GRAPH procedures, requiring multiple steps including creation of the graphs by country of origin, defining a GREPLAY template, and rendering the graphs in the template. However, the graphs created by SGPANEL give you a few important benefits for free:

- Automatic uniform axes.
- A uniform and common legend
- A clean, crisp and modern look.

The SGPANEL procedure automatically creates the number of cells needed based on the unique values of the class variables. You can use multiple class variables and plot overlays as shown in this graph:



```

TITLE 'Patient Profile by Age Group ...';
PROC SGPANEL DATA = sashelp.heart;
    FORMAT ageatstart age.;
    PANELBY ageatstart sex /
        LAYOUT = LATTICE NOVARNAME;
    SCATTER X = weight Y = diastolic /
        TRANSPARENCY = 0.8;
    REG X = weight Y = diastolic /
        CLM NOMARKERS;
RUN;

```

For details, please see procedure documentation. Examples can be found at: support.sas.com/sassamples/graphgallery.

Sanjay Matange, SAS Institute

News

SAS Global Forum 2009

This year's SAS Global Forum was held in Maryland, just outside Washington DC, from 22-25 March with a new twist to it.

- An online social network
- Real time updates and video
- Daily announcements
- 25% of the participants were not from US

This year's SAS Global Forum was full of interesting and exciting new concepts and ideas. As compared to previous years, the content of the presenters were in greater depth and sophistication. There were a number of new techniques applied to real life issues using SAS discussed in the SAS Global Forum. Generalized additive models, structural analysis and general estimating equations were some of the more familiar names. Although there were many new presenters, the classic presenters like Art Carpenter and Andrew Karp were also presenting their papers covering other interesting areas such as traffic lighting in tables and linking HTML files together. Paul Dorfman's presentation together with Koen Vyerman on DOW loop was fascinating and presented the complex loop in a simple palatable way. Paul Sherman's presentation on password protection was intriguing and challenged other SAS users to look into their password protection.

There were a couple of panel discussion where users were allowed to voice their opinions. The demo room was filled with the latest innovations from SAS. I was particularly impressed by Peggy Haas' demonstration of SAS manuals on Kindle and BlackBerry. Although they are currently in experimental stage, the prototypes were very interesting and easy for SAS users to use.

Although this was my first time here at SAS Global Forum, I am excited at the prospect of going to Seattle in 2010 for the next conference. The birds of feather (BOF) for SAS-L posters will definitely be something to look forward to.

Murphy Choy, University College Dublin

PhUSE Single Day Events: Frankfurt and Brussels

The Frankfurt PhUSE Single Day Event (SDE) was held 6 May. It was another great success due to the hard work of Beate Hientzsch and Dirk Spruck, both from Accovion, the sponsor of

this event, and all the presenters. There were about 50 delegates for this one, so it appears that we are growing.

The morning session consisted of several important CDISC topics followed by a delicious lunch. Who says there ain't no free lunch! During the lunch break, Ann Martin of UCB held a discussion about open source software and the OS3A open source clinical programming project. The afternoon session consisted of a great series of more traditional programming papers. By the end of the day, both my stomach and my brain were full. Well worth the couple of hours train ride from Aachen.

The previous SDE in Brussels that was sponsored by UCB Pharma and organized by Pierre Mayeur of UCB and Heidi Curinckx of J&J PRD, was every bit as good. Lots of important topics were discussed and everyone found something of value (and also another great free lunch).

These are excellent events for learning, the sharing ideas and general networking. But, while you can't put a price on knowledge anyway, they are included in your membership fee and cost nothing.

I keep mentioning the "Free Lunch". That is because, with recent corporate cutbacks on travel and conference spending, the events in your region are really easy on the departmental budget. I highly recommend the SDEs to all European pharmaceutical programmers.

The next PhUSE Single Day Event will be held on 27 May 2009 at SAS UK in Marlow from 0900-1600hr. It will be a co-production with the CDISC English Speaking User Group.

Ted Feuerbach, Feuerbach and Associates

In Brief

- Back issues of VIEWS News are available from the VIEWS web site, and also from the VIEWS News page at www.sasCommunity.org/wiki/VIEWS_News
- Amadeus is starting courses and seminars in Frankfurt am Main, Germany, later this year. Some of the seminars are listed in the Diary section below. A full list of courses and seminars can be found at: www.sascommunity.org/wiki/VIEWS_Events_Diary.
- More SAS hints and tips can be found on the SAS FAQ pages www.hollandnumerics.com/sasfaq and also on the Amadeus Software Tips pages at www.amadeus.co.uk/TipsList.aspx?ST=1
- There is a social networking groups for SAS users on Ning:
 - **SAS Professionals**, intended for SAS users in the UK, at www.sasprofessionals.net, but currently totalling more than 800 members worldwide.
- Phil Holland has also started a group page on Squidoo for other pages related to SAS, and would like to make it a central resource for SAS hints, tips, links and books at www.squidoo.com/groups/SAS_users
- SAS users who use Twitter can now join a "twibe" at twibes.com/SAS, which filters your tweets down to those containing "SAS", "EnterpriseGuide" and "SGF".
- There is a new forum system for PhUSE members, including SAS-related streams at www.phenix-mtk.com/forums

- Phil Mason has a SAS tips newsletter, which can be requested by going to www.woodstreet.org.uk and then clicking on Tips. Previous tips can also be found on the web site.

Ask The Consultant

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

Text editors for SAS programmers

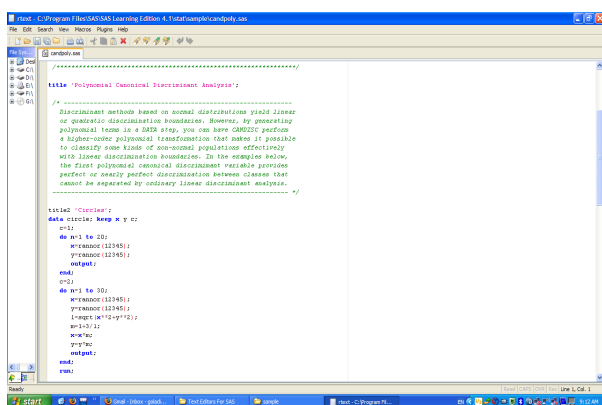
Q: What text editors can be used to write SAS programs?

A: Most SAS programmers use the in built SAS enhanced editor for their daily works. Sometimes, this editor is replaced by the code editor of SAS Enterprise Guide. However, some SAS users like their text editor to be very customizable and full of features which may or may not be in the enhanced editor. Sometimes, the SAS programmers may not always work in SAS environment and they are involved in other programming languages such as Perl, C, C++ and Java. In this article, I will introduce a number of different editors which are useful for various programmers utilizing different styles.

Programming work is always done in a text editor which caters to the taste and style of the programmer. Each programmer will have their favourite code editor which differs with individuals. Most of the time, the first few editors used by the programmer will likely to be the ones that they stick to. Occasionally, some users will find their current editors to be insufficient in handling their work and starts searching for alternative editors. Below are some of the text editors that the author likes and used in his job as a SAS programmer.

Rtext

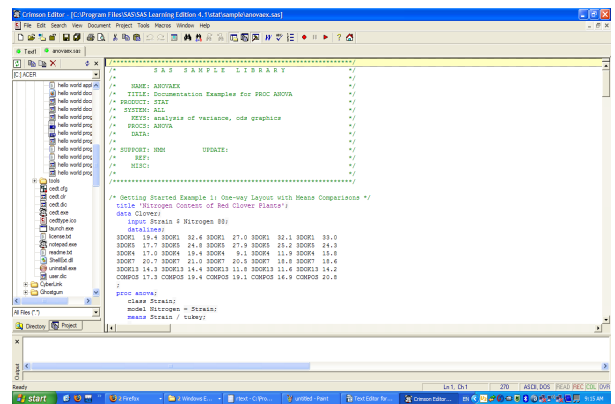
Rtext is a small open source code editor with a very nice user interface and provides highlighting for SAS programs. Rtext is available on most of the platform that SAS operates on, with the exception of Z/OS and openVMS. Below is a screen shot of the editor:



The author likes the simple layout of Rtext with its excellent code highlight functions. The software is also very small and simple to install. The explorer window also allows multiple source programs to be opened and referred to simultaneously.

Crimson Editor

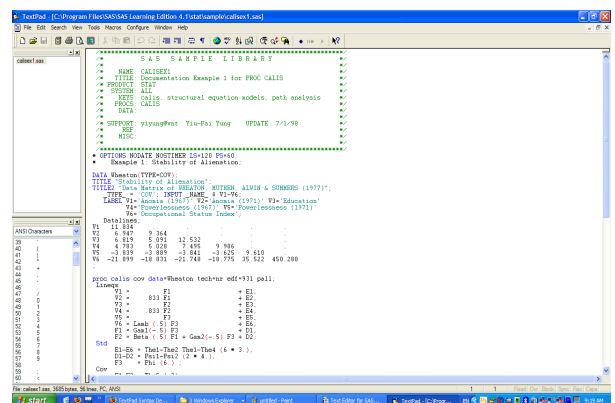
Crimson Editor is one of the most popular editor available for programmers to use. It is open source and has many options that users can try. Crimson Editor also allow users to install schematics that will highlight sections of the SAS programs. It is sometimes know as Emerald Editor due to a rewrite by a group of open source programmers. Below is a screen shot of the editor:



The editor has a very crisp look to it and is very useful for programming with other languages beside SAS.

Textpad

Textpad is a proprietary software by Helios Software company. This editor is the personal favourite of the author and is used everyday in his work. Textpad has the usual highlighting syntax feature, like many editors, and it has many indentation and block select features. These features are excellent for SAS macro programming and SCL programming. Besides these, Textpad has a built in compiler for Java which allows for rapid switching to Java coding that is occasionally required. Textpad also features multiple line searching and replacement, which is extremely useful when a simple variable name needs to be replaced. Below is a screen shot of the editor:



Textpad has many macro features that allows for repetitive actions to be recorded and recycled easily.

Conclusion

The style of programming strongly influence the type of editors that the programmer use. It is acknowledged that very famous editor such as emacs and Vim are not shown here. However, the author does not have any experience utilizing them and thus cannot comment on them properly.

Murphy Choy, University College Dublin

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions.

[SAS 9] The **MAX()** and **MIN()** functions allow you to easily identify the maximum and minimum values from a selection of variables. However, in order to find the second highest value or the bottom 3 values, you will need to make use of the SAS 9 functions **LARGEST()** and **SMALLEST()**.

The following examples are based on the data listed below:

```
DATA balances;
  INPUT id ball-bal6;
  DATALINES;
13245  911.00 863.30  896.98 1627.26 1672.20 1973.21
13246   62.27 10.28 1377.79 1290.92 602.13 1905.32
13247  336.61 29.78 196.32   3.40      .    592.76
13248  691.90 593.42 1598.80 4170.23  72.70      .
13249 1765.04      .      .    1032.22      .    1325.36
RUN;
```

The syntax for the **MAX()** and **MIN()** functions is as follows:

```
function(list of variables);
```

The **LARGEST()** and **SMALLEST()** functions have similar syntax:

```
function(value, list);
```

Where *value* is the *n*th item you are looking for (e.g. 2 for the second largest/smallest), and *list* is a list of variables that will be processed (as with the **MAX()** and **MIN()** functions).

The following code demonstrates the use of the **MAX()**, **MIN()**, **LARGEST()** and **SMALLEST()** functions:

```
DATA balances2;
  SET balances;
  maxbal = MAX(OF ball-bal6);
  minbal = MIN(OF ball-bal6);
  large1 = LARGEST(1, OF ball-bal6);
  large2 = LARGEST(2, OF ball-bal6);
  small11 = SMALLEST(1, OF ball-bal6);
  small12 = SMALLEST(2, OF ball-bal6);
  small13 = SMALLEST(3, OF ball-bal6);
  DROP bal;
RUN;
```

The following results are produced:

maxbal	large1	large2	minbal	small11	small12	small13
1973.21	1973.21	1672.20	863.30	863.20	896.98	911.00
1905.32	1905.32	1377.79	10.28	10.28	62.27	602.13
592.76	592.76	336.61	3.40	3.40	29.78	196.32
4170.23	4170.23	1598.80	72.70	72.70	593.42	691.90
1765.04	1765.04	1325.36	1032.22	1032.22	1325.36	1765.04

Note that where the first argument in the **LARGEST()** and **SMALLEST()** functions is set to '1', the result is the same as the **MAX()** and **MIN()** functions. Also note that in this example when evaluating the minimum, the missing values have been ignored.

The **MAX()** and **MIN()** functions completely exclude missing data from the evaluation. However, the **LARGEST()** and **SMALLEST()** functions handle missing values in the data by placing any missing values present at the end of the sorted list of variables.

So the **LARGEST()** function will return the maximum data value, where a '1' is specified in the first argument, the second largest for a '2', and so on. Once all of the non-missing data are referenced, the missing values will be returned.

The **SMALLEST()** function will return the lowest value, when '1' is specified, the second lowest for a '2', and so on. Again, once all the non-missing values have been referenced, the missing values will be returned. The last non-missing value, in this case, will be the highest data value.

The following example demonstrates how the **LARGEST()** and **SMALLEST()** functions deal with missing data values:

```
DATA balances3;
  SET balances;
  large2 = LARGEST(2, OF ball-bal6);
  large5 = LARGEST(5, OF ball-bal6);
  small13 = SMALLEST(3, OF ball-bal6);
  small15 = SMALLEST(5, OF ball-bal6);
  DROP bal;
RUN;
```

Generates the following data:

large2	large5	small13	small15
1672.20	896.98	911.00	1672.20
1377.79	62.27	602.13	1377.79
336.61	3.40	3.40	592.76
1598.80	72.70	691.90	4170.23
1325.36	.	1765.04	.

Amadeus Software Consultancy Team

PhUSE News

Call for Papers for PhUSE 2009 Conference

I am very pleased to announce that we have our first papers submitted to the website. As always this can be marked as the beginning of our Conference.

As the expectations for the conference are high, we are striving to share new material to keep the content fresh, interesting and innovative. All stream chairs and co-chairs are working closely together, looking pro-actively for good and high qualitative papers for their streams.

Abstracts for papers are to be submitted to our paper portal, which can be directly accessed from the PhUSE website. I want to thank Anup Patel for having coordinated the efforts to improve this portal in order to facilitate the work for the stream (co-)chairs, the conference committee, the PhUSE back-office and the authors.

We will have twelve streams and a poster session for which abstracts can be submitted: Applications and Software Development, CDISC & Industry Standards, Coder's Corner, Coding Standards, Data Handling, Management, Professional Development and Training, Regulatory, Statistics and Pharmacokinetics, Tutorials, Technical Solutions and Posters.

As the venue in Basel has room for novelties, discussions with the organizing committees have led to the expansion of the program in 2009. We will have a separate Discussion club in which we will organize themed sessions where we hope to have lively and enthusiastic discussions/debates with you and experts within the industry.

Besides debates on Industry standards, Open Source and others, we will use this room to elaborate on the Main Conference theme, being Accelerating Clinical Development. If you are interested in participating in these panel discussions, please contact me at benjamin.szilagyi@phuse.eu.

Finally, I would like to thank everyone who is contributing to the success of Basel 2009, especially my Co-Chair Heidi Curinckx and the organizing committee for their continued commitment in order to organize the best event possible.

Plan now to attend PhUSE 2009 in Basel, register soon and be aware that papers can be submitted until June 15th. It is for the interest of us all and for the interest of the industry that you share your good ideas and your experiences with your colleagues from around the world.

Looking forward to seeing you in Basel!

Benjamin Szilagyi, PhUSE Conference Chair

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.

May 2009

27 **PhUSE/CDISC Single Day Event**, Marlow

See the PhUSE web site www.phuse.eu for further details.

June 2009

1 **SAS Programming Tips and Tricks**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

11 **PhUSE/PSI Single Day Event**, Marlow

See the PhUSE web site www.phuse.eu for further details.

25 **New and Enhanced Features of SAS 9.2**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

July 2009

14-16 **SAS Professionals Convention**, Marlow

See www.sasprofessionals.net for further details.

20 **SAS Programming Tips and Tricks**, Frankfurt am Main, Germany

See the Amadeus Software web site www.amadeus.co.uk for further details.

22 **From Data to Knowledge: A Journey into SAS9**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

September 2009

24 **SAS Programming Tips and Tricks**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

October 2009

19-20 **Best Practices in SAS Statistical Programming in Regulatory Submission**, Hoofddorp, The Netherlands

These courses are intended for anyone directly or indirectly responsible for the creation, content or validation of summary tables, data lists and graphs used to support research, drug or medical device efficacy and safety in a regulatory submission. See tinyurl.com/5ayrmq for more details of the courses from Sunil Gupta.

19-21 **PhUSE Conference**, Basel, Switzerland

The Pharmaceutical Users Software Exchange (PhUSE) conference is the premier programming event for the pharmaceutical industry in Europe. See www.phuse.eu for more details.

28 **New and Enhanced Features of SAS 9.2**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

November 2009

2 **SAS Programming Tips and Tricks**, Frankfurt am Main, Germany

See the Amadeus Software web site www.amadeus.co.uk for further details.

19 **From Data to Knowledge: A Journey into SAS9**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

December 2009

1 **SAS Programming Tips and Tricks**, Oxfordshire

See the Amadeus Software web site www.amadeus.co.uk for further details.

All VIEWS event information will be posted and/or emailed to registered members of VIEWS, and will include an event application form. Please send your queries about any VIEWS events to event@views-uk.org, and don't forget to look at the web site for the latest news.

Contacts

Amadeus Software

Email: info@amadeus.co.uk

Tel: +44-(0)1993-878287

Web: www.amadeus.co.uk

LeRoy Bessler
Assurant Health

Email: le_roy_bessler@wi.rr.com

Murphy Choy
University College Dublin

Email: goladin@gmail.com

Ted Feuerbach
Feuerbach and Associates

Email: feuer@feuerbach.com

Web: www.feuerbach.com

Sanjay Matange
SAS Institute

Email: sanjay.matange@sas.com

PhUSE

Email: office@phuse.eu

Web: www.phuse.eu

SAS UK

Tel: +44-(0)1628-486933

Web: www.sas.com/uk

VIEWS

Web Site

Web: www.views-uk.org

Chairman

Email: chair@views-uk.org

Newsletter Editor

Email: newsletter@views-uk.org

Membership secretary

Email: membership@views-uk.org

***VIEWS News** is published quarterly by VIEWS International SAS Programmer Community. VIEWS is a not-for-profit organisation. Our Mission Statement and Rules can be viewed on our web site together with a list of current committee members.*