

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 70, 4th Quarter 2022

Wishing you a very Happy and Healthy New Year from the UK! Now we are in January I have a feeling 2023 is going to be a very productive year! This newsletter issue includes yet another useful SAS programming tip from Nina L Werner, and some thoughts on how to develop SAS ODS Graphics programs with different SAS maintenance versions based on a recent discussion in the SAS Programming Forum on my blog site. As usual I have publicised some SAS-related events for the next few months, and I have also added some more SAS formats, options and functions that you may or may not have come across before, but you might find interesting, and even useful.

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 feel free to send an email to me at views-uk@hollandnumerics.org.uk.

Philip R Holland (Newsletter Editor)

Membership of VIEWS is **free** and you can register for notification of all future VIEWS News issues by registering at blog.hollandnumerics.org.uk, and then emailing views-uk@hollandnumerics.org.uk, or using the blog site menu, to request access to the VIEWS UK forum. Members will receive an email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please inform us when you change your email address, otherwise your registration may be cancelled.

Ask The Consultant

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

How to Draw a Butterfly Plot

Q: How can I draw a butterfly plot with this data?

```
INPUT question $
      dislike dislike_strongly like like_strongly;

Cheese, 10, 20, 2, 11
Meat, 9, 7, 5, 12
Salad, 3, 4, 7, 2
Bread, 12, 10, 2, 6
Milk, 9, 10, 9, 11
```

A:[SAS 9.4M3, SAS 9.4M7] Let us start by using PROC SGPLOT, and create 2 bar charts that will be overlaid on each other. In order to use a common x-axis, the values in the left-hand panel will be negative, and in the right-hand panel will be positive. The program below creates the Butterfly plot data

for both PROC SGPLOT and the ODS Graphics Template, which will be discussed later. The **&maxcount** macro variables are created to force the left and right x-axes to show the same ranges.

```
%LET dsn = work.butterfly;

%LET var1 = dislike;
%LET var2 = dislike_strongly;
%LET var3 = like;
%LET var4 = like_strongly;

PROC FORMAT;
  INVALUE catorder
    "%sysfunc(PROPCASE(&var1.))" = 1
    "%sysfunc(PROPCASE(&var2.))" = 2
    "%sysfunc(PROPCASE(&var3.))" = 3
    "%sysfunc(PROPCASE(&var4.))" = 4;
  PICTURE abs
    LOW -< 0 = "0009" (MULT = -1)
    0 - HIGH = "0009"
    OTHER = "*****";
RUN;

DATA &dsn.;
  RETAIN maxcount .;
  INFILE DATALINES DLM = ',';
  INPUT question $ &var1. &var2. &var3. &var4.;
  ** Find the maximum size of stacked bar **;
  IF SUM(&var1., &var2.) GT maxcount
    THEN maxcount = SUM(&var1., &var2.);
  IF SUM(&var3., &var4.) GT maxcount
    THEN maxcount = SUM(&var3., &var4.);
  CALL symput('maxcount', maxcount);
  CALL symput('maxcount1', -maxcount);
  DATALINES;
Cheese, 10, 20, 2, 11
Meat, 9, 7, 5, 12
Salad, 3, 4, 7, 2
Bread, 12, 10, 2, 6
Milk, 9, 10, 9, 11
;
RUN;

PROC SORT DATA = &dsn.;
  BY question;
RUN;

%PUT maxcount1=&maxcount1.
      maxcount=&maxcount.;

PROC TRANSPOSE DATA = &dsn. OUT = plotdata;
  BY question;
  VAR &var1. &var2. &var3. &var4.;
RUN;

DATA plotdata;
  SET plotdata;
  LENGTH category category1 categoryr $50;
  category = PROPCASE( _NAME_ );
```

```

IF UPCASE( _name_ ) EQ:
  UPCASE(SUBSTR("&var1.", 1, 1)) THEN DO;
    countl = -coll;
    countlg = coll;
    categoryl = category;
  END;
ELSE DO;
  countr = coll;
  countrg = coll;
  categoryr = category;
END;
LABEL category = 'Category'
      categoryl = 'Category'
      categoryr = 'Category';
RUN;

DATA plotdata;
  SET plotdata;
  catorder = INPUT(category, catorder.);
RUN;

PROC SORT DATA = plotdata;
  BY question catorder category;
RUN;

```

Butterfly plot data

Obs	question	_NAME_	COL1	category	categoryl	categoryr	countl	countlg	countr	countrg	catorder
1	Bread	dislike	12	Dislike	Dislike		-12	12	.	.	1
2	Bread	dislike_strongly	10	Dislike_strongly	Dislike_strongly		-10	10	.	.	2
3	Bread	like	2	Like		Like	.	.	2	2	3
4	Bread	like_strongly	6	Like_strongly		Like_strongly	.	.	6	6	4
5	Cheese	dislike	10	Dislike	Dislike		-10	10	.	.	1
6	Cheese	dislike_strongly	20	Dislike_strongly	Dislike_strongly		-20	20	.	.	2
7	Cheese	like	2	Like		Like	.	.	2	2	3
8	Cheese	like_strongly	11	Like_strongly		Like_strongly	.	.	11	11	4
9	Meat	dislike	9	Dislike	Dislike		-9	9	.	.	1
10	Meat	dislike_strongly	7	Dislike_strongly	Dislike_strongly		-7	7	.	.	2
11	Meat	like	5	Like		Like	.	.	5	5	3
12	Meat	like_strongly	12	Like_strongly		Like_strongly	.	.	12	12	4
13	Milk	dislike	9	Dislike	Dislike		-9	9	.	.	1
14	Milk	dislike_strongly	10	Dislike_strongly	Dislike_strongly		-10	10	.	.	2
15	Milk	like	9	Like		Like	.	.	9	9	3
16	Milk	like_strongly	11	Like_strongly		Like_strongly	.	.	11	11	4
17	Salad	dislike	3	Dislike	Dislike		-3	3	.	.	1
18	Salad	dislike_strongly	4	Dislike_strongly	Dislike_strongly		-4	4	.	.	2
19	Salad	like	7	Like		Like	.	.	7	7	3
20	Salad	like_strongly	2	Like_strongly		Like_strongly	.	.	2	2	4

Using the PLOTDATA data set above, the following program creates the attributes map data set and the graph below. The PROC SGPLOT only uses the following variables from PLOTDATA: y value=QUESTION, group=CATEGORY, response=COUNTL (left-hand panel) or COUNTR (right-hand panel). The response variables are also displayed as absolute values, so they both appear to be positive. As the input data PLOTDATA has already been sorted, the DISCRETEORDER=DATA options honours that order in the y-axis labels.

```

DATA attrmap (KEEP = id value fillcolor);
  LENGTH id value fillcolor $50;
  SET plotdata;
  id = 'bars';
  value = category;
  SELECT (value);
    WHEN ("%sysfunc(PROPCASE(&var1.))") DO;
      fillcolor = 'lightblue';
    END;
    WHEN ("%sysfunc(PROPCASE(&var2.))") DO;
      fillcolor = 'lightred';
    END;
    WHEN ("%sysfunc(PROPCASE(&var3.))") DO;
      fillcolor = 'lightgreen';
    END;
  END;

```

```

  WHEN ("%sysfunc(PROPCASE(&var4.))") DO;
    fillcolor = 'yellow';
  END;
  OTHERWISE;
  END;
RUN;

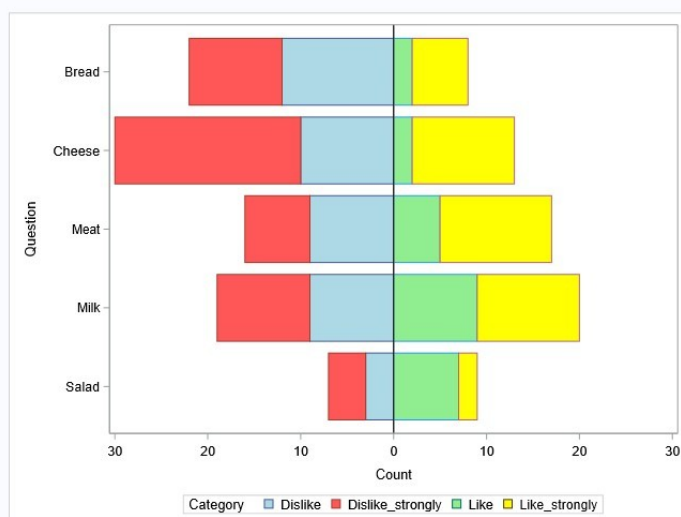
PROC SORT DATA = attrmap NODUPKEY;
  BY id value;
RUN;

PROC SGPLOT DATA = plotdata DATTRMAP = attrmap;
  TITLE 'Butterfly plot from SGPLOT';
  XAXIS LABEL = 'Count' MIN = &maxcountl.
    MAX = &maxcountr. TYPE = LINEAR;
  YAXIS LABEL = 'Question' DISCRETEORDER = DATA;
  HBAR question / GROUP = category
    GROUPDISPLAY = STACK RESPONSE = countl
    ATTRID = bars;
  HBAR question / GROUP = category
    GROUPDISPLAY = STACK RESPONSE = countr
    ATTRID = bars;
  REFLINE 0 / AXIS = X
    LINEATTRS = (COLOR = BLACK);
  FORMAT countl countr abs.;
RUN;

```

Note that the PROC SGPLOT step above contains no post-9.4M3 statements and options, but crashes in 9.4M3 with an exception. In 9.4M7 this step runs as expected.

Butterfly plot from SGPLOT



The following program defines the ODS Graph Template that is rendered by PROC SGRENDER. The PROC SGRENDER only uses the following variables from PLOTDATA, but there are completely separate variables for the left-hand and right-hand panels: y value=QUESTION, group=CATEGORYL or CATEGORYR, response=COUNTLG or COUNTRG. As the response variables are already both positive values, there is no need to display them as positive, and these values can be added inside the segments with SEGMENTLABEL=TRUE options on both BARCHART statements in the Butterfly template. The central y-axis labels are controlled by the DISPLAY, REVERSE and TICKVALUEHALIGN sub-options within YAXISOPTS= options.

```

PROC TEMPLATE;
  DEFINE STATGRAPH Butterfly;
    DYNAMIC _question _countl _categoryl
    _question _countr _categoryr;
    BEGINGRAPH;
      LAYOUT LATTICE / ROWDATARANGE = DATA
        ROW2DATARANGE = DATA
        COLUMNDATARANGE = DATA COLUMNS = 2
        ROWGUTTER = 10 COLUMNGUTTER = 10
        COLUMNWEIGHTS = (0.5 0.5);
      LAYOUT OVERLAY / XAXISOPTS =
        (REVERSE = TRUE TYPE = LINEAR
        DISPLAY = (TICKVALUES LINE)
        LINEAROPTS = (VIEWMAX = &maxcountr.))
        YAXISOPTS = (REVERSE = TRUE
        DISPLAY = (LINE) GRIDDISPLAY = OFF
        DISCRETEOPTS =
        (TICKVALUEFITPOLICY = NONE));
      BARCHART CATEGORY = _question
        RESPONSE = _countl / STAT = MEAN
        NAME = 'barleft'
        ORIENT = HORIZONTAL
        GROUP = _categoryl
        GROUPORDER = DATA
        GROUPDISPLAY = STACK
        CLUSTERWIDTH = 1.0
        SEGMENTLABEL = TRUE;
    ENDLAYOUT;
    LAYOUT OVERLAY / XAXISOPTS =
      (REVERSE = FALSE TYPE = LINEAR
      DISPLAY = (TICKVALUES LINE)
      LINEAROPTS = (VIEWMAX = &maxcountr.))
      YAXISOPTS = (REVERSE = TRUE
      DISPLAY = (TICKVALUES LINE)
      GRIDDISPLAY = OFF DISCRETEOPTS =
      (TICKVALUEFITPOLICY = NONE)
      TICKVALUEHALIGN = LEFT);
      BARCHART CATEGORY = _question
        RESPONSE = _countr / STAT = MEAN
        NAME = 'barright'
        ORIENT = HORIZONTAL
        GROUP = _categoryr
        GROUPORDER = DATA
        GROUPDISPLAY = STACK
        CLUSTERWIDTH = 1.0
        SEGMENTLABEL = TRUE;
    ENDLAYOUT;
    SIDEBAR / ALIGN = bottom;
      DISCRETELEGEND 'barleft' 'barright' /
        OPAQUE = TRUE BORDER = TRUE
        HALIGN = CENTER VALIGN = CENTER
        EXCLUDE = (" ")
        SORTORDER = ASCENDINGFORMATTED
        DISPLAYCLIPPED = TRUE;
    ENDSIDEBAR;
  ENDLAYOUT;
ENDGRAPH;
END;
RUN;

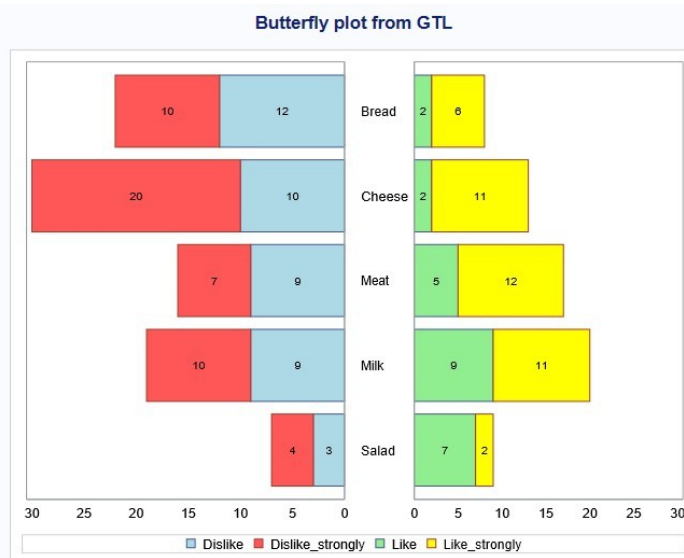
PROC SGRENDER DATA = plotdata
  TEMPLATE = Butterfly DATTRMAP = attrmap;
  TITLE "Butterfly plot from GTL";
  DATTRVAR categoryl = "bars"
  categoryr = "bars";

```

```

DYNAMIC _question = "question"
_countl = "countlg"
_categoryl = "categoryl"
_countr = "countrg"
_categoryr = "categoryr";
RUN;

```



What could still be improved?

- The left and right panels are not the same size! This is because there are only 2 windows in the LAYOUT LATTICE, and they are the same size, but the right-hand windows includes the central labels. This could be resolved by making 3 windows, and only displaying the labels in the central window.
- It would be nice to have the legend values in the same order as the bar segments are drawn. Unfortunately this is quite complicated to achieve, as the sort order impacts both the bar segments and the legend order, and the DISCRETELEGEND statement only has more limited SORTORDER values than that in the BARCHART statement.

It is recommended that you use as up-to-date a version and maintenance level of SAS software as possible. This will widen your choice of options, but also fix bugs that can hinder your programming!

Philip R Holland (Newsletter Editor)

Did You Know?

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

Using SYSPARM to Pass Parameters

[SAS] Instead of editing a series of %LET statements to set current values for parameters into macro variables, you can use SYSPARM and a clever macro, % read_sysparm from Martha Messineo's wonderful book "Practical and Efficient SAS Programming: The Insider's Guide", Appendix A: Utility Macros, p. 190, Creating Macro Variables from SYSPARM. You can download all her code zipped on support.sas.com [SAS Downloads: Book title: Practical and Efficient SAS® Programming: The Insider's Guide]. The link is: support.sas.com/downloads/package.htm?pid=2134.

```

/* ----- */
/* Macro:      _read_sysparm
/*
/* Description: This macro will read the
/* contents of the sysparm option and create
/* macro variables from the contents. This
/* assumes that the contents are space or comma
/* delimited and the values are either in name
/* value pairs (name=value) or just a value.
/* If it is just a value then the macro
/* variables will be called parm#
/* If any of the macro variables don't have
/* names, then an additional macro variable
/* called __macroCount will be created to tell
/* you how many parm# variables were created.
/*
/* Parameters:  notes - set the parameter to
/*                nonotes to keep the
/*                notes about which macro
/*                variables were created
/*                from being printed in
/*                the SAS Log.
/*
/* Author:      Martha Messineo
/* ----- */

%macro _read_sysparm(notes);

    /* turn off the log options so that this */
    /* macro doesn't fill up the SAS log      */
    %global __debug;
    %if (&__debug le 0) %then
    %do;
        %local l_mlogic l_symbolgen l_mprint;
        %let l_mlogic =
            %sysfunc(getoption(mlogic));
        options nomlogic;
        %let l_symbolgen =
            %sysfunc(getoption(symbolgen));
        options nosymbolgen;
        %let l_mprint =
            %sysfunc(getoption(mprint));
        options nomprint;
        %local l_source l_notes;
        %let l_source =
            %sysfunc(getoption(source));
        %let l_notes = %sysfunc(getoption(notes));
        options nosource nonotes;
    %end; /* if - not debug */

    %global __macroCount;
    %let __macroCount = 0;

    data _null_;

        parms = sysparm();
        count = countw(parms, " ,");
        do i = 1 to count;
            parm = scan(parms, i, " ,");
            if (index(parm, "=")) then
            do;
                parmName = scan(parm, 1, "=");
                parmValue = scan(parm, 2, "=");
            end;
            else
            do;
                macroCount + 1;
                parmName = cats("parm", macroCount);
                parmValue = parm;
            end;
        end;

```

```

        call symputx(strip(parmName),
                      strip(parmValue), "G");
        %if (%upcase(&notes) ne NONOTES) %then
        %do;
            putlog 'Macro variable created: &'
                parmName "=" parmValue;
        %end;
    end;
    if (macroCount gt 0) then
    do;
        call symputx("__macroCount",
                      macroCount, "G");
    end;
run;

/* reset log options */
%if (&__debug le 0) %then
%do;
    options &l_source &l_notes &l_mprint;
    options &l_mlogic;
    options &l_symbolgen;
%end; /* if - not debug */

%mend _read_sysparm;

```

Store your compiled macro in a shared or local SAS macro library or save and %INCLUDE the SAS macro code. Additionally, you can store your SYSPARM in an external text file and call it with %INCLUDE so you never have to touch your SAS program code just to update parameter values each time you run.

The following SAS code is contained in the file **thisjob_sysparm.txt**:

```

OPTIONS
    SYSPARM="me=Nina, today=20220826, a=2.5, ABCD4680";

/* instead of:
%LET me = Nina;
%LET today = 20220826;
%LET a = 2.5;
%LET parm1 = ABCD4680; */

```

My SAS program contains the following SAS statements to use the %_read_sysparm() macro:

```

* load the macro from text (if not already
  loaded into a macro library);
%INCLUDE "_read_sysparm.sas";

* load the SYSPARM values;
%INCLUDE "thisjob_sysparm.txt";

* run the macro to parse all the parameters into
  macro variables;
%_read_sysparm();

* optional: shows created macro variables and
  their values;
%PUT _GLOBAL_;

--the rest of my code which needs those
parameters--

```

Nina L Werner, Wisconsin

"From Practical and Efficient SAS® Programming: The Insider's Guide (ISBN 978-1-63526-023-6) by Martha Messineo. Copyright 2017, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC"

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions. Please read the SAS documentation to find out more about them.

[SAS 9.4M4] The SAS format **ODDSRw.d** displays SAS values as odds ratios.

[SAS 9.4M3] In the STATGRAPH template in PROC TEMPLATE, the LAYOUT OVERLAY statement now includes the feature to adjust the SAS format for the tick values on an axis using the **TICKVALUEFORMAT=** sub-option in the XAXISOPTS= and YAXISOPTS= options.

[SAS 9.4M3] The SAS function **COT()** returns the cotangent, **CSC()** returns the cosecant and **SEC()** returns the secant.

In Brief

- Back issues of *VIEWS News* are available from the VIEWS UK forum at **blog.hollandnumerics.org.uk**. Please register for free membership on the blog, and then request free access to the VIEWS UK forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 38,500 members. Everyone is welcome to request to join, but discussions here are restricted to SAS-related topics: **www.linkedin.com/groups?gid=70702**.
- The SAS Opportunities group on LinkedIn, which currently has over 16,300 members. Everyone is welcome to request to join, but discussions here are also restricted to SAS-related topics: **www.linkedin.com/groups?gid=805217**.
- In the UK and Ireland the SUGUKI group had been holding regular meetings in London, and also occasional meetings in Bristol, Dublin, Edinburgh, Leeds and Manchester. For more details and to register for future events please go to **meetup.com/SUGUKI**.
- Fancy joining the SAS EMEA Analytics Explorers, then please send your **country, name, email address** and **company** with your permission to pass these details on to SAS to **views-uk@hollandnumerics.org.uk**. Other regions include Asia Pacific, Canada, Latin America and US, but to join these communities based on where you live, you will now need to provide this information to a current member of these communities, instead of me.
- Do you use Microsoft Visual Studio Code? You may be interested in a SAS-related extension called SASjs. You can develop SAS applications and run code on SAS Viya, SAS 9 EBI or SASjs Server! Just set up the target, and click the running man to execute your SAS code. The SASjs Server could be your local SAS installation. SASjs is an open source project, and can be obtained from the Visual Studio Code Marketplace.
- Altair Analytics Workbench Community Edition provides named users with free access to Altair Analytics Workbench software (formerly WPS) including a SAS language compiler with coding and workflow tools on unlimited data volumes. A Community Edition licence key will be issued that will activate the software for up to 12 months. At the end of this term, users may be invited to renew their licence and/or to upgrade to the latest available full version of the software. Download the software from **www.altair.com/altair-slc**. No

technical support service is offered with the Community Edition, but assistance can be requested using the Community forums at **community.altair.com/community**.

- Altair, a global leader in computational science and artificial intelligence, paid and fully satisfied the 2019 North Carolina judgment SAS Institute obtained against recently acquired World Programming (WPL), a UK-based technology company that specializes in data analytics software. Shortly thereafter on 03Mar2022, the North Carolina court lifted the injunction that had previously prevented WPL from licensing its solutions to new customers in the United States. Note that Altair can now license WPL solutions to customers in the United States and around the world.

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.

March 2023

5-8 **PhUSE US Connect**, Orlando, FL, USA

PhUSE Connect events bring together our global Community to collaborate on hot topics, share perspectives and discuss different approaches to guiding the healthcare industry landscape, both now and in the future. at **www.phuse-events.org**.

May 2023

8-10 **SAS Innovate**, Orlando, FL, USA

SAS Innovate is a conference dedicated to analytics and AI for executives and how to transform existing data into faster, more effective decisions in an era of digital transformation, economic uncertainty, consumer demands and fierce competition. More details at **www.clocate.com/sas-innovate**.

14-17 **PharmaSUG**, San Francisco, CA, USA

Join us for the premier event of the year for statistical programmers, statisticians, data managers, researchers, and others who analyze data in the life sciences - PharmaSUG 2023! Full details at **www.pharmasug.org**.

Contacts

Nina L Werner,
Wisconsin

Email: **Nina.Werner@dot.wi.gov**

SAS UK

Tel: +44-(0)1628-486933

Web: **www.sas.com/uk**

VIEWS

Web Site

Web: **blog.hollandnumerics.org.uk**

Newsletter Editor

Email:
views-uk@hollandnumerics.org.uk

***VIEWS News** is published quarterly for the VIEWS International SAS Programmer Community by Holland Numerics Limited.*