

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 73, 3rd Quarter 2023

Supporting all Foundation SAS programmers, this newsletter continues to bring you information to make the most of SAS 9. This newsletter issue includes more ideas about plotting multiple values and a book review from LeRoy Bessler, Nina L Werner gives us another SAS tip, and Richard Carson continues his discussion about configuring of SAS processes without editing them in the third part of a series of articles about KT. 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

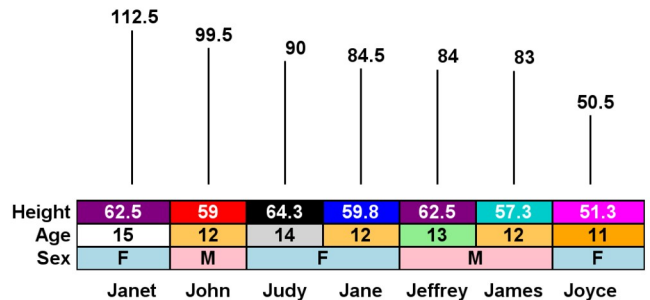
This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

How to Plot Five Variables in Two Dimensions

Q: Can you plot 5 variables in 2 dimensions?

A: [SAS 9.4] The graph below is a derivative and enhancement of the examples in Figures 4-42 and 4-43 in my book "Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization". There, I showed how one can add information to a Needle Plot with one Block Chart overlaid below the needles. The block charts below are an enhanced alternative to three x-axis tables, but with colour coding to make any identical values of each block variable readily apparent.

Student Ranked Weight (pounds) with Height (inches), Age, and Sex



Three block charts with custom colour palettes for fills use all white or all black for block values depending on readability.

This article demonstrates design guidelines for graphs and colour use. It includes a macro to simplify assignment of uniform text attributes (font, size, and weight, which by macro default is Bold) for **all** text in the graph. It uses a Discrete Attribute Map to specify the three colour palettes. In two dimensions, it presents precise values for all five variables. It ranks the values of the response variable, and in the block charts uses colour coding. The White values in the Height colour blocks are due to an override of the default colour Black.

Five relevant design guidelines from my book are as follows:

- Provide precise numbers to deliver correct and certain knowledge to the graph viewer. No guessing based on axis values.
- Rank values of the response variable (here the weights) for the categories (here the names).
- Provide sufficient contrast between background colour and text colour to assure readability.
- Make all text the same font, size, and weight. Weight choices are Bold or Normal. Varying text characteristics adds no value. All text should be important. **Omit** the unimportant.
- For maximum readability make all text Bold. Bold never does harm. One text weight is simpler to implement.

Before showing the code to build the result, I share a macro that I developed when writing the book. To control all of the text attributes for its hundreds of examples with direct controls would have expanded the book and the coding task. Direct controls were used, if needed, as overrides.

Direct controls for TITLE and FOOTNOTE statements are FONT= and HEIGHT=, as well as the BOLD option. One can instead imbed the Bold weight with, e.g. FONT='Arial/Bold'.

Direct controls for other graph parts are sub-options of, for example, the VALUEATTRS and LABELATTRS options for XAXIS and YAXIS statements. The sub-options are FAMILY=

(for font choice), SIZE=, and WEIGHT=. There are numerous other “ATTRS” options, not used or discussed here.

In the book, I created the **AllGraphTextSetup** macro.

My most frequent invocation of the macro was **%AllGraphTextSetup(11)**; To use the custom ODS style that this creates, your program must start with:

```
ODS LISTING GPATH="C:\TEMP"
STYLE=GraphFontArial11ptBold;
```

The LISTING destination sends to the folder C:\TEMP. With this macro invocation, all text in the image is 11pt Arial Bold by default. All direct text controls are available to customize text parts that need different attributes, typically to make text smaller to cure problems, such as for XAXIS and YAXIS statements. 11pt x-axis values might not fit unless tilted or omitted, in part, with “thinning”. See the documentation for more information about thinning and the FITPOLICY options for AXIS statements. When the default font size from the ODS style causes axis value fit problems, an override with VALUEATTRS=(SIZE=Npt) is the solution, with the appropriate value of N.

For this example 7pt Arial Bold poses no text fit problems.

In the macro below the other option for Weight is Normal. The number N specified for Size is used as Point Size, Npt. This number is the macro's positional parameter. A macro may have at most one positional parameter, and it must be first in the invocation of the macro when the other parameters are used. Family may be used for font names with imbedded blanks, such as Times New Roman, but the style created omits blanks, as in style name **GraphFontTimesNewRoman10ptBold**. The blanks are removed with the %LET statement below.

```
%MACRO AllGraphTextSetup(
    Size,Family=Arial, Weight=Bold);
%LET FAMILYFORSTYLE =
    %SYSFUNC (COMPRESS(&Family,' '));
PROC TEMPLATE;
    DEFINE STYLE
        GraphFont&FamilyForStyle.&Size.pt&Weight;
    PARENT=STYLES.LISTING;
    CLASS GraphFonts /
        'GraphValueFont' =
            ("&Family",&Size.pt,&Weight)
        'GraphLabelFont' =
            ("&Family",&Size.pt,&Weight)
        'GraphDataFont' =
            ("&Family",&Size.pt,&Weight)
        'GraphTitleFont' =
            ("&Family",&Size.pt,&Weight)
        'GraphFootnoteFont' =
            ("&Family",&Size.pt,&Weight);
    END;
RUN;

%MEND AllGraphTextSetup;

%AllGraphTextSetup(7);

PROC SORT DATA=SASHELP.CLASS OUT=work.ToPlot;
    WHERE Name =: 'J';
    /* Name must start with 'J' */
    BY DESCENDING Weight;
RUN;
```

```
DATA work.DattrMapForThreeColorPalettes;
    LENGTH ID $ 12 Value $ 4 FillColor $ 11;
    ID='SexColors';
    Value='F'; FillColor='LightBlue'; OUTPUT;
    Value='M'; FillColor='Pink'; OUTPUT;
    ID='AgeColors';
    Value='11'; FillColor='Orange'; OUTPUT;
    Value='12'; FillColor='LightBrown'; OUTPUT;
    Value='13'; FillColor='LightGreen'; OUTPUT;
    Value='14'; FillColor='LightGray'; OUTPUT;
    Value='15'; FillColor='White'; OUTPUT;
    ID='HeightColors';
    Value='51.3'; FillColor='CXFF00FF'; OUTPUT;
    Value='57.3'; FillColor='CX00CCCC'; OUTPUT;
    Value='59'; FillColor='Red'; OUTPUT;
    Value='59.8'; FillColor='Blue'; OUTPUT;
    Value='62.5'; FillColor='Purple'; OUTPUT;
    Value='64.3'; FillColor='Black'; OUTPUT;
RUN;

ODS RESULTS OFF;
ODS _ALL_ CLOSE;

ODS LISTING GPATH="C:\temp" DPI=300
/* DPI=300 makes lines & text smoother */
STYLE=GraphFontArial7ptBold;

ODS GRAPHICS ON / RESET=ALL SCALE=OFF
WIDTH=3.65in HEIGHT=2in
IMAGENAME="NeedlePlotWithThreeBlockCharts";

TITLE JUSTIFY=CENTER 'Student Ranked Weight
(pounds) with Height (inches), Age, and Sex';

PROC SGPLOT DATA=ToPlot NOBORDER
    DATTRMAP=DattrMapForThreeColorPalettes;
    NEEDLE X=Name Y=Weight / DISPLAYBASELINE=OFF
    DATALABEL DATALABELPOS=TOP ;
    BLOCK X=Name BLOCK=Height /
    POSITION=BOTTOM ATTRID=HEIGHTCOLORS
    VALUEATTRS=(COLOR=White);
    BLOCK X=Name BLOCK=Age /
    POSITION=BOTTOM ATTRID=AGECOLORS;
    BLOCK X=Name BLOCK=Sex /
    POSITION=BOTTOM ATTRID=SEXCOLORS;
    XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS);
    YAXIS DISPLAY=NONE;
RUN;

ODS LISTING CLOSE;
```

LeRoy Bessler, Visual Data Insights™

Read a review of LeRoy's book later in this issue.

Did You Know?

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

KT – Part 3: Adding tasks to the COVID-19 project

What this is about

In part 1 of this series I outlined how a standard framework for recurring SAS processes could be controlled in entirety with external configuration files. In part 2 I demonstrated how the task of importing COVID-19 data into a project could be configured externally and encapsulated in a nodeConnect process.

My implementation, KT, has 3 core processes – nodeStem, nodeBranch and nodeConnect. In this part I will describe how an additional process task – extracting dimensional data – can be configured as a separate nodeConnect process. Later I will show how multiple instances of nodeConnect and different tasks can be controlled through nodeBranch. Finally I will show how the project as a whole is configured using nodeStem.

Example 2 – Adding another process for global COVID-19 confirmed case data

Examination of the global COVID-19 confirmed case data (ultimately derived from the collection maintained by the Johns Hopkins University Center for Systems Science and Engineering) reveals that the varying information is contained in 2 columns (Date and Value, i.e. the cumulative count of cases in the indicated locality), the primary key is a combination of 2 columns (Country/Region and Province/State) and 'ISO 3166-1 Alpha 3-Codes' is a foreign key. The remaining 5 columns are dimensional (reference) data and can be placed in a separate table.

Here is the configuration data in table **work.covid19confirmed_extractDims**:

Section	Parameter	Value
process	macroname	extractDimensionDataLite
sourcedata	library	work
sourcedata	tablename	covid19_confirmed_global
outputdata	library	work
outputdata	tablename	covid19_dimensions_global
columns	primary	'Country/Region'n
columns	primary	'Province/State'n
columns	foreign	'ISO 3166-1 Alpha 3-Codes'n
columns	dimension	'Region Code'n
columns	dimension	'Sub-region Code'n
columns	dimension	'Intermediate Region Code'n
columns	dimension	Lat
columns	dimension	Long

Macro **extractDimensionDataLite** is a generic macro so its non-Lite version will be included in the KT toolkit. It is an example of a data consolidation task, where the whole source table is processed but there is no longer a one to one correspondence between rows.

Here is the nodeConnect invocation which will run the macro:

```
%nodeConnectLite (
    configlib=work
    ,configdata=covid19confirmed_extractDims
);
```

The code for **extractDimensionDataLite** is:

```
%MACRO extractDimensionDataLite (
    configlib=
    ,configdata=
);
%* Convert parameter values
into local macro variables;
%LOCAL sourcetable outputtable
extractcols ordercols;
```

```
PROC SQL NOPRINT;
SELECT value
INTO :sourcetable SEPARATED BY ','
FROM &configlib..&configdata.
WHERE UPCASE(section) = 'SOURCEDATA'
;

SELECT value
INTO :outputtable SEPARATED BY ','
FROM &configlib..&configdata.
WHERE UPCASE(section) = 'OUTPUTDATA'
;

SELECT value
INTO :extractcols SEPARATED BY ','
FROM &configlib..&configdata.
WHERE UPCASE(section) = 'COLUMNS'
;

SELECT value
INTO :ordercols SEPARATED BY ','
FROM &configlib..&configdata.
WHERE UPCASE(parameter) = 'PRIMARY'
;

%* Create dimension table;
CREATE TABLE &outputtable. AS
SELECT DISTINCT &extractcols.
FROM &sourcetable.
ORDER BY &ordercols.
;
QUIT;
%MEND extractDimensionDataLite;
```

Suggestions for additional data preparation tasks

The Johns Hopkins University team have been very careful to preserve data as presented to them, but this has resulted in some unusual features. Some clean-up is recommended.

Create configuration tables and task macros to:

1. Create subsets of regional data for Australia, Canada and China (the United States regional data is in a separate downloadable file).
2. Create a special events subset with cruise ship and Olympics records.
3. Create a Locality column with the logic:
 - When ('Province/State'n is NULL) then 'Country/Region'n
 - When ('Country/Region'n in ('Australia', 'Canada', 'China')) then 'Country/Region'n
 - Else 'Province/State'n
4. Sum the Value data, grouped by Locality, 'ISO 3166-1 Alpha 3-Codes'n, Date (this will elevate Australia, Canada and China confirmed cases to national level).
5. Turn the cumulative data into a daily or weekly difference table (i.e. additional cases).
6. Remove records with null values for 'ISO 3166-1 Alpha 3-Codes'n (North Korea will be affected, but there is no credibility in the value reported).
7. To avoid errors when merging with population data, set 'ISO 3166-1 Alpha 3-Codes'n to null for dependencies of France and the United Kingdom where the locality does not have a different code from the country of jurisdiction.

Richard C Carson

SAS Tip - %PUT: A new syntax choice

[SAS 9.4] The classic %PUT writes characters and substitutes in values of macro variables.

%PUT; writes a blank line to the Log.

```
%PUT >>>My userID is >&SYSUSERID.< on &SYSDAY.;  
%PUT;
```

The new syntax &= writes characters and the macro variable name = the value of the macro variable.

```
%PUT --> &=SYSUSERID. &=SYSDAY. <--;  
%PUT;
```

Change the colour of the text in the Log by inserting a keyword before your text.

```
%PUT NOTE: --> &=SYSDAY. <-- print in blue;  
%PUT WARNING: --> &=SYSDAY. <-- print in green;  
%PUT ERROR: --> &=SYSDAY. <-- print in red;  
%PUT;
```

You can always use the special keywords to display all the current macro variables and values.

```
%PUT _ALL_ ; %PUT;  
%PUT _GLOBAL_ ; %PUT;  
%PUT _LOCAL_ ; %PUT;
```

Interesting that

```
%PUT NOTE: &=SYSDAY. ;
```

includes the word “NOTE:” in the Log, but

```
%PUT NOTE: --> &=SYSDAY. ;
```

doesn't.

Try it yourself. Do you like the new results better?

Nina L Werner, Wisconsin

Book Review

Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization

Find the book at link.springer.com where the 20% coupon code VizData23 is redeemable until 31 December 2023 for unlimited times use for ISBN 978-1-4842-8608-1 (paperback) and for ISBN 978-1-4842-8609-8 (eBook).

This is the book for which there is no substitute. It is, in effect, a course on how to do ANYTHING that ODS Graphics can do, and how to do it best. Exception: No vector plots. Learn by example how to deliver graphs with clarity, certainty, and comprehensibility. Viewers of your work will never need to wonder “What is it all about?”.

The examples deliver both a visual for quick, easy understanding, and precise values for the graphic elements, making unreliable estimates from axis values unnecessary. For the case when neither data labels, nor axis tables, nor any other method can provide enough, the book shows how to use ODS HTML5 to deliver precise numbers as data tips (mouseover text) and optionally a link to web table or spreadsheet.

The book takes you way beyond the defaults to the best choice, avoiding both the challenge of Options OverChoice and the sad reality of options that yield anti-usable results. For heat maps, it provides alternatives to colour gradient legends that provide no genuine knowledge as to the numeric values for the colour-filled areas. For bubble plots, it demonstrates the problem of default bubble sizing.

Also, it includes unique creations: sparsely annotated time series, enhanced multi-line time series, maximally informative bar charts, better box plots (the standard one requires prior explanation, there is even an annotated diagram provided in the SAS documentation to decode it), the tree chart, the flag chart, and the crossroads signpost chart. It even offers surrogates for bar charts and simple time series without bars, lines, or markers, visuals with text and numbers only, they really work! Histograms can be more than under-informative summary images, see ones that are truly illuminating, or purpose-built (i.e. with bins that are meaningful, not arbitrary).

Not only are standalone graphs covered, but also all the ways to assemble composites (as panel, lattice, or matrix) are shown in multiple chapters, including one devoted to PROC SGPanel. There is also a chapter on ODS LAYOUT, the tool that enables Anything Anywhere All At Once, graphs, tables, and text as any composite you wish!

Based on my experience-based design principles, 48 for graphics and 23 for colour use, 327 examples take SASHELP data sets as input. The code (optionally downloadable) can be explored, experimented with, or adapted to repurpose the design to your needs and application.

I hope you enjoy the book. I enjoyed writing it, having spent 42 years in a quest to get the best out of SAS graphics software, and finally sharing my ideas in a book. Let me know what you think (email). Post a review if you wish. Thanks for your interest.

LeRoy Bessler, Visual Data Insights™

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 9.2] The SAS format **VMSZNw.d** generates VMS and MicroFocus COBOL zoned numeric data, and the SAS informat **VMSZNw.d** reads VMS and MicroFocus COBOL zoned numeric data.

[SAS 9.4] The system option **DECIMALCONV=** specifies the binary and decimal conversion methodology for formatting numeric data. There are 2 possible values: **COMPATIBLE** converts and formats decimal values by using the conversion methodology that is compatible with releases prior to SAS 9.4, and **STIEEE** converts and formats decimal values by using the IEEE Standard for Floating-Point Arithmetic 754-2008. Using the **STIEEE** argument improves the accuracy and readability of floating-point numbers. In some cases, more significant digits can be displayed in the same field width.. Use this option when SAS output might be processed by existing applications that are sensitive to the details of formatting.

[SAS] The function **REPEAT()** returns a character string of duplicated characters.

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 39,900 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 19,400 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 Editor? 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 SLC software (formerly WPS) including a SAS language compiler with coding and workflow tools on unlimited data volumes, with interfaces to a wide range of databases, as well as R and Python programming environments. 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**. The current General Access version of Altair SLC is 5.23.0 (2023).

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.

November 2023

5-8 **PhUSE EU Connect**, Birmingham, UK

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. Full details at **www.phuse-events.org**.

February 2024

25-28 **PhUSE US Connect**, Bethesda, MA, 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. Full details at **www.phuse-events.org**.

April 2024

28-30 **MWSUG**, Milwaukee, WI, USA

MidWest SAS Users Group (MWSUG) is an official SAS Users Group which exists to serve users of SAS software in the midwestern region by providing educational opportunities and resources and by facilitating communication among SAS users and SAS.

Contacts

LeRoy Bessler , Visual Data Insights™	Email: Le_Roy_Bessler@wi.rr.com
Richard C Carson	Email: richard@arcol-analytics.com
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.*