

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information
for SAS® Users

Issue 75, 1st Quarter 2024

Supporting all Foundation SAS programmers, this newsletter continues to bring you information to make the most of SAS 9. This newsletter issue includes examples of how to make choropleth maps using ODS Graphics from LeRoy Bessler, and Richard Carson continues his discussion about configuring of SAS processes without editing them in the fifth and penultimate part of a series of articles about KT. As usual I have publicised some more SAS-related events for the next few months, and I have also added more SAS formats, options and functions that you may, or may not, have come across before, but you might find interesting, and even useful.

Note that LeRoy Bessler and I will be having an Authors Discussion Lunch in Milwaukee on 26Jun2024, or, more accurately, LeRoy and I will be having lunch in Milwaukee, and you are welcome to join us. A link for more details can be found in the Diary section below.

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

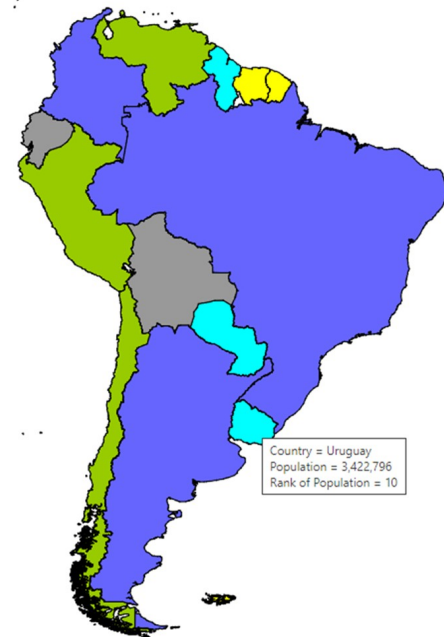
This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

Custom Bins (Ranges) for an InfoGeographic

Q: Can you add information to maps using ODS Graphics?

A: [SAS 9.4] A choropleth map is also called a thematic map, it tells a story or provides information beyond the geography. Even before the term “infographic” became popular for free-form overloaded composites of any information and data graphics, I had coined the term InfoGeographic. In SAS you can build one with PROC SGMAP (originally PROC GMAP) and the CHOROMAP statement. Below is a web-enabled InfoGeographic with the mouse resting on Uruguay.

2023 Ranked Population for 14 Countries in South America
Use mouse for Country Name, Population, Population Rank



Ranges:	3,801 - 618,046 (3 lowest)
	808,727 - 6,780,745 (below median)
	12,224,114 - 18,001,002 (median)
	19,603,736 - 34,049,588 (above median)
	45,510,324 - 215,313,504 (3 highest)

The challenge when creating an InfoGeographic is how to colour-code the areas. Not just which colours to use and how, but also how to break up the response variable values into ranges or bins. One of my longstanding graphic design mantras is “Show Them What’s Important”. A ranked bar chart is an excellent way to present categorical data. For a map of population, it’s interesting to show which countries are most or least populous. For a continent with more countries, 3 highest and 3 lowest could be 10 instead, or whatever you prefer.

The PROC SGMAP default way to break up the populations for this map would be 14 single-country bins. Other possibilities are CHOROMAP LEVELTYPE options of QUANTILE or INTERVAL. Experiment with them, if interested. They use colour differently, and boringly.

I use my brightest colours for the areas which are likely to be smallest (small areas tend to accommodate fewer people). I want small area fills to be detectable, even in a pie chart. You should be able to see some yellow peeking out of the black boundary of the Falkland Islands. At the end of the article I discuss the idea of annotating areas.

Below is the code for this example. It can be adapted to other data and other geographic entities. I actually created a macro version, but it is not shown here lest the code be longer and less readily understood. Wikipedia explains the ISOALPHA3

variable as “three-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest”.

The MAPDATA data set MAPSGFK.World, besides variables used internally by the software, includes ISOALPHA3, CONT for numeric Continent ID (91 to 97), and LONG (longitude). LONG is used in a WHERE clause below to exclude several outlying islands of South America. If included, there would have been much more white space at the sides of the continental mainland, shrinking the interesting part of the image in this 3.65-inch-wide column.

```
DATA work.MapInput;
  INFILE DATALINES;
  INPUT @1 Country $16.
        @18 ISOALPHA3 $3.
        @22 Population 9.;
  DATALINES;
Argentina      ARG 45510324
Bolivia        BOL 12224114
Brazil         BRA 215313504
Chile          CHL 19603736
Colombia       COL 51874028
Ecuador        ECU 18001002
Falkland Islands FLK 3801
French Guiana  GUF 304568
Guyana         GUY 808727
Paraguay       PRY 6780745
Peru           PER 34049588
Suriname       SUR 618046
Uruguay        URY 3422796
Venezuela      VEN 28301700
RUN;

%LET Continent = South America;
%LET N = 3; /*I usually prefer 10, but there are
only 14 countries in South America. The code
will work with N = 1. In any case, N must be
small enough so that there will be at least one
observation between the Top and Bottom ranges
and the Median. When the total observation count
is even, the Median will contain two
observations. For this data the maximum value
for N is 5.*/

PROC SORT DATA = work.MapInput;
  BY DESCENDING Population;
RUN;

DATA work.MapInput;
  SET work.MapInput;
  Rank = _N_;
RUN;

PROC SORT DATA = work.MapInput;
  BY Population;
RUN;

DATA _NULL_;
  SET work.MapInput NOBS = HowMany;
  CALL SYMPUT('MaxObsDisplay',
    TRIM(LEFT(PUT(HowMany, COMMA21.))));
  CALL SYMPUT('MaxObs', HowMany);
  IF MOD(HowMany, 2) EQ 0 THEN DO;
    CALL SYMPUT('MedianBottomObs',
      (HowMany / 2));
    CALL SYMPUT('MedianTopObs',
      (HowMany / 2) + 1);
  END;
END;
```

```
ELSE DO;
  CALL SYMPUT('MedianBottomObs',
    CEIL(HowMany / 2));
  CALL SYMPUT('MedianTopObs',
    CEIL(HowMany / 2));
END;
RUN;

DATA work.ToFormat;
  KEEP Start End Label FmtName Type HLO;
  LENGTH Start End 8 Label $100;
  RETAIN FmtName "PopFmt" Type 'N' Start 0;
  SET work.MapInput End = LASTONE;
  IF _N_ EQ 1 THEN Start = Population;
  IF _N_ LT &N THEN RETURN;
  IF _N_ EQ &N THEN DO;
    End = Population;
    Label = TRIM(LEFT(PUT(Start, COMMA13.)))
      || ' - '
      || TRIM(LEFT(PUT(End, COMMA13.)))
      || " (&N lowest)";
    OUTPUT;
    CALL SYMPUT('Nlowest', TRIM(LEFT(Label)));
  END;
  IF _N_ EQ %EVAL(&N + 1) THEN DO;
    Start = Population;
    IF _N_ EQ %EVAL(&MedianBottomObs - 1)
    THEN DO;
      End = Population;
      Label = TRIM(LEFT(PUT(Start, COMMA13.)))
        || ' - '
        || TRIM(LEFT(PUT(End, COMMA13.)))
        || ' (below median)';
      OUTPUT;
      CALL SYMPUT('BelowMedian',
        TRIM(LEFT(Label)));
    END;
    ELSE RETURN;
  END;
  ELSE IF _N_ EQ %EVAL(&MedianBottomObs - 1)
  THEN DO;
    End = POPULATION;
    Label = TRIM(LEFT(PUT(Start, COMMA13.)))
      || ' - '
      || TRIM(LEFT(PUT(End, COMMA13.)))
      || ' (below median)';
    OUTPUT;
    CALL SYMPUT('BelowMedian',
      TRIM(LEFT(Label)));
  END;
  ELSE IF _N_ EQ &MedianBottomObs
  OR
    _N_ EQ &MedianTopObs THEN DO;
    IF _N_ EQ &MedianBottomObs THEN DO;
      Start = Population;
      IF _N_ LT &MedianTopObs THEN RETURN;
    END;
    IF _N_ EQ &MedianTopObs THEN DO;
      End = Population;
      Label = TRIM(LEFT(PUT(Start, COMMA13.)))
        || ' - '
        || TRIM(LEFT(PUT(End, COMMA13.)))
        || ' (median)';
      OUTPUT;
      CALL SYMPUT('Median', TRIM(LEFT(Label)));
    END;
  END;
  ELSE IF _N_ EQ %EVAL(&MedianTopObs + 1)
  THEN Start = Population;
```

```

IF _N_ LT %EVAL(&MaxObs - &N) THEN RETURN;
ELSE IF _N_ EQ %EVAL(&MaxObs - &N) THEN DO;
  End = Population;
  Label = TRIM(LEFT(PUT(Start, COMMA13.)))
  || ' - '
  || TRIM(LEFT(PUT(End, COMMA13.)))
  || ' (above median)';
  OUTPUT;
  CALL SYMPUT('AboveMedian',
    TRIM(LEFT(Label)));
END;
ELSE IF _N_ EQ %EVAL(&MaxObs - &N + 1)
THEN Start = Population;
IF _N_ LT &MaxObs THEN RETURN;
ELSE IF _N_ EQ &MaxObs THEN DO;
  End = Population;
  Label = TRIM(LEFT(PUT(Start, COMMA13.)))
  || ' - '
  || TRIM(LEFT(PUT(End, COMMA13.)))
  || " (&N Highest)";
  OUTPUT;
  CALL SYMPUT('NHighest', TRIM(LEFT(Label)));
END;
IF LASTONE;
HLO = 'O';
Label = 'No Data';
OUTPUT;
RUN;

PROC FORMAT CNTLIN = work.ToFormat
  LIBRARY = work;
RUN;

PROC TEMPLATE;
  DEFINE STYLE STYLES.LeRB_FiveColorMap;
    PARENT = STYLES.HTMLBLUE;
    CLASS COLORS /
      'DOCBG' = CXFFFFFF /*white background for
        web page*/
      'SYSTITLEBG' = CXFFFFFF; /*white
        background for NOGTITLE title lines
        if used for LINK= capability.
        Such titles must be in web page
        frame around the graph*/
    CLASS GRAPHCOLORS /
      'GDATA1' = CXFFFFF0 /*N lowest*/
      'GDATA2' = CX00FFFF /*below median*/
      'GDATA3' = CX999999 /*median*/
      'GDATA4' = CX99CC00 /*above median*/
      'GDATA5' = CX6666FF; /*N highest*/
    ;
  END;
RUN;

%LET Title = 2023 Ranked Population for
  &MaxObsDisplay Countries in &Continent;

ODS RESULTS = OFF;
ODS _ALL_ CLOSE;

ODS HTML5 PATH = "C:\temp" (URL = NONE)
  OPTIONS(BITMAP_MODE = "INLINE") /*image is
    imbedded, not a separate file*/
  GTITLE GFOOTNOTE /*these are defaults*/
  STYLE = STYLES.LeRB_FiveColorMap
  BODY="%SYSFUNC(COMPRESS(&Title, ' ')).HTML"
  (TITLE = "&Title");
/*this optional TITLE= text appears at
  upper left of the browser window*/

```

```

ODS GRAPHICS ON / RESET = ALL SCALE = OFF
  WIDTH = 6.5IN HEIGHT = 9IN NOBORDER
  TIPMAX = 528 /*From SAS log as required to
    display data tips*/
  IMAGEMAP = ON /*needed for data tips*/
  IMAGENAME = 'NoFileNameLikeThisCanBeFound'
  OUTPUTFMT = PNG; /*SVG is not supported for
    PROC SGMAP */
TITLE1 COLOR = WHITE HEIGHT = 4PT "SPACE";
TITLE2 COLOR = BLACK HEIGHT = 15PT "&Title";
TITLE3 COLOR = WHITE HEIGHT = 4PT "SPACE";
TITLE4 COLOR = BLUE HEIGHT = 15PT
  "Use mouse for Country Name, Population,
  Population Rank";

PROC SGMAP MAPRESPDATA = work.MapInput
  MAPDATA=MAPSGFK.World
  (WHERE = (CONT EQ 92 /*South America*/
    AND
    ISOALPHA3 NE 'CUW' /*Correct an error.
      Curacao is an island in
      the Caribbean.*/
    AND (-82 LT LONG LT -29) /*trim off
      outlying islands of
      mainland countries*/
  ))
  DESCRIPTION = ' '; /*prevent mouseover pop-up
    of "The SGMAP Procedure", which is an
    annoying distraction when trying to raise
    data tips.*/
  FORMAT Population PopFmt.;
  CHOROMAP Population /
    MAPID = ISOALPHA3 ID = ISOALPHA3 DISCRETE
    TIP = (Country Population Rank)
    TIPFORMAT = ($69. COMMA13. $2.)
    TIPLabel = ('Country' 'Population'
      'Rank of Population')
    LINEATTRS = (THICKNESS = 1 COLOR = BLACK);
  KEYLEGEND / TITLE = 'Ranges:' AUTOITEMSIZE
  TITLEATTRS =
    (FAMILY = Arial SIZE = 15PT WEIGHT = Bold)
  VALUEATTRS =
    (FAMILY = Arial SIZE = 15PT WEIGHT = Bold)
  ;
RUN;

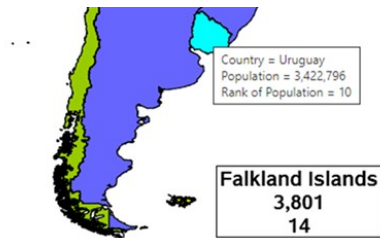
ODS HTML5 CLOSE;

```

NOTE: The SAS log for the PROC SGMAP step will include this message: WARNING: Variable POPULATION not found in data set MAPSGFK.WORLD. It will apply to any response variable that you choose to use on the CHOROMAP statement. The only variables required from MAPSGFK.WORLD are map variables. For an InfoGeographic with permanent annotation of three lines for each unit geographic area (rank, area name, and population see Chapter 15 of my book, “Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization”. There, to find where to place the annotation, the code relies on the SAS-provided **mapsgfk.uscenter** data set.

To annotate, say, Brazil and Falkland Islands, the most and least populous countries, there are no SAS-provided annotation locations. Coding can determine and adjust them. It is also possible to create a legend alternative with colour swatches that are the equivalent of the FILLASPECT=GOLDEN option for a KeyLegend. That option, as well as border suppression, and other customary KeyLegend features, are not available with the current release of PROC SGMAP.

Below is a snippet of what IS possible. Feel free to email request the code if interested.



Ranges:

- 3,801 - 618,046 (3 lowest)
- 808,727 - 6,780,745 (below median)
- 12,224,114 - 18,001,002 (median)
- 19,603,736 - 34,049,588 (above median)
- 45,510,324 - 215,313,504 (3 highest)
- No Data

This custom legend's "No Data" entry is not needed for THIS data, but shows that it can be made available when needed. The legend is really footnotes that use UNICODE rectangle symbols as a surrogate for FILLASPECT=GOLDEN colour swatches.

LeRoy Bessler, Bessler Consulting and Research

LeRoy's book "Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization" can be bought and downloaded from Apress, Amazon and all good book resellers.

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 5: Initial configurations for 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 parts 2 and 3 I demonstrated how the task of importing COVID 19 data into a project could be configured externally and encapsulated in individual **nodeConnect** processes, and in part 4 I introduced the **nodeBranch** macro to initiate each task in sequence

My implementation, KT, has 3 core processes: **nodeStem**, **nodeBranch** and **nodeConnect**. In this part I will introduce initial configuration tables and associated macros that will build the environment for running the project. The final part will introduce the **nodeStem** macro and additional setup statements to run the complete project.

As in previous examples for brevity the code for the process is a "Lite" version, without documentation, parameter validation and error handling. At this point all configuration files will exist as CSV files, and the configuration tables will be loaded into a work directory.

Example 4 – Managing the environment for global COVID 19 confirmed case data

First we need a macro to read the config files. In this case **configPath** is the fully referenced location of the CSV file, and **configData** is the name of the imported table.

readConfigFileLite.sas:

```
%MACRO readConfigFileLite
  (configPath=
  ,configData=
  ,configLib=WORK
  ,delim=', '
  );
  DATA &configLib..&configFile. (COMPRESS = NO);
    INFILE "&configPath." DLM = &delim. DSD
      MISOVER FIRSTOBS = 2;
    LENGTH Section Parameter $64
      Value $256
      Comment $64
      ;
    INPUT Section Parameter Value Comment &;
  RUN;
%MEND readConfigFileLite;
```

The **configOptions** file lists SAS system options for the project. Other options can be added: e.g. **mprint**, **mlogic** and **symbolgen**. The macro uses the **configOptions** table to set the options.

configOptions.csv:

Section	Parameter	Value	Comment
Lite	dlcreatedir		required
Lite	validvarname	any	
Lite	validmemname	extend	
Lite	compress	yes	

setOptionsLite.sas:

```
%MACRO setOptionsLite
  (configdata=, configlib=WORK);
  %GLOBAL processOptions;
  PROC SQL NOPRINT;
    SELECT CATX('=', parameter, value)
      INTO :processOptions SEPARATED BY ' '
      FROM &configlib..&configdata.
      ;
  QUIT;
  OPTIONS &processOptions;
%MEND setOptionsLite;
```

It is useful to have project parameters loaded as global macro variables.

configGlobals.csv:

Section	Parameter	Value	Comment
Lite	title	COVID-19 Confirmed Global Cases Time Series	
Lite	prefix	COVID19	
Lite	configlib	WORK	
Lite	branchConfig	COVID19branch.csv	
Lite	sourceLocn	C:\Projects\WorldStatistics\COVID-19\ sourcedata	
Lite	sourceRef	time_series_covid19_confirmed_global_narrow.csv	

setGlobalsLite.sas:

```
%MACRO setGlobalsLite
  (configdata=, configlib=WORK);
  DATA _NULL_;
    SET &configlib..&configdata.;
    IF NOT MISSING(value);
    PUT 'NOTE: ' parameter '=' +1 value;
    CALL SYMPUTX(parameter, value, 'G');
  RUN;
%MEND setGlobalsLite;
```

The **configLibraries** file lists SAS Libraries for the project data, and the macro implements them.

configLibraries:

Section	Parameter	Value	Comment
Lite	staging	C:\Projects\WorldStatistics\COVID-19\Lite\datastores\staging	
Lite	covid19	C:\Projects\WorldStatistics\COVID-19\Lite\datastores\covid19	

setLibrariesLite.sas:

```
%MACRO setLibrariesLite
  (configdata=, configlib=WORK);
  DATA _NULL_;
    SET &configlib..&configdata.;
    WHERE UPCASE(section) = 'LITE';
    IF NOT MISSING(value);
    rc = LIBNAME(parameter, RESOLVE(value));
    sysmsg = SYSMSG();
    PUT sysmsg;
  RUN;
%MEND setLibrariesLite;
```

A more advanced project might have additional initial configurations, e.g. for project specific formats, informats and task macros.

The **configProject** file lists the configuration files and associated macros to be implemented in the **nodeStem** process. Additional configurations need to be listed here if they are to be implemented in **nodeStem**.

Section	Parameter	Value	Comment
options	setOptionsLite	configOptions	required
globals	setGlobalsLite	configGlobals	required
libraries	setLibrariesLite	configLibraries	required

Richard C Carson

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 and Altair SLC] User-written SAS formats and informats can include pre-existing formats or informats, respectively, by enclosing them in square brackets, e.g. **OTHER=[DATE9.]**. This means that you do not have to re-invent what already exists.

[SAS 6 for Windows] When working in the SAS system on a Windows platform the initial folder can be specified with the **-SASINITIALFOLDER** configuration option on the command line or in the configuration file.

[SAS 9.4M3] The following Data Step functions carry out timezone-related operations based on the SAS system options **TIMEZONE=** and **LOCALE=**. **TZONENAME()** returns the current timezone name. **TZONEOFF()** calculates the offset

between the current timezone and the specified timezone. **TZONEID()** returns the current timezone ID. **TZONES2U()** converts a SAS datetime value into a UTC datetime value, and **TZONEU2S()** converts a UTC datetime value into a SAS datetime value.

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 40,400 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 21,200 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 new SAS Global Analytics Explorers hub, then please fill in your **name**, **company email address** and **company** at <https://infl.tv/npmv>.
- 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 integrated R and Python programming environments. A Community Edition or Personal 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 5.24.1 (2024).
- Do you use Microsoft Visual Studio Code Editor?
 - ◆ You may be interested in a SAS-related extension called **SASjs - DevOps for SAS**. 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, where the current version is v2.6.0.
 - ◆ Another Microsoft Visual Studio Code Editor extension has just been released called **Altair SLC**. This extension allows you to connect to local or remote Altair SLC installations, even if, until now, you have only been able to use Altair Analytics Workbench to access Altair SLC. The current version in the Visual Studio Code Marketplace is v0.5.0 (pre-release).

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.

April 2024

16–19 **SAS Innovate**, Las Vegas, NV, 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. Full details at innovate.sas.com.

May 2024

19–22 **PharmaSUG**, Baltimore, MD, USA

PharmaSUG is the premiere conference of the year for software professionals in the life sciences industry. For more details see www.pharmasug.org.

June 2024

26 **Authors Discussion Lunch**, Milwaukee, WI, USA

LeRoy Bessler and Philip R Holland will be discussing their latest and next books over lunch in Milwaukee. Join them for lunch and ask all of the SAS questions you have ever wanted to ask, with book signing possible too. If you are interested in attending, then look for more details which will be published on blog.hollandnumerics.org.uk, or email either of us directly.

November 2024

16-19 **MWSUG**, Waukesha, WI, USA

This will be the first post-pandemic conference of the MidWest SAS Users Group, which has been serving users in 12 states in the mid-western region of the United States since 1990. It provides educational opportunities and resources, and facilitates communication amongst SAS users and SAS. Full details at www.mwsug.org.

Contacts

LeRoy Bessler

Email: Le_Roy_Bessler@wi.rr.com

Richard C Carson

Email: richard@arcol-analytics.com

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.*