

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 59, 1st Quarter 2020

This newsletter issue contains an article from LeRoy Bessler about a new ODS Graphics procedure, an article from Scott Thomson about an alternative analytics environment, and a Base SAS macros article from myself. I have again included some SAS-related events publicised for the next few months, and I've also added some more SAS formats, options and functions that you 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.

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 the New ODS Graphics PROC SGPIE

You can create communication-effective pie charts, with ranking (ordering), category, value, and percent.

The new ODS Graphics SGPIE procedure, as of February 2020, is still in Pre-Production status in SAS Version 9.4 TS01M6.

Four images created here have the ODS Graphics default width of 640 pixels. These results were shrunk upon insertion into this document. The default height is 480 pixels. Case 5 uses a width in inches to match the width of this newsletter column, and a minimum height in inches to fit the legend.

All of the examples suppress use of the "Other" slice that could consolidate small slices. Use of an "Other" category just prompts a question, "What is in 'Other'?" I have always contended that a well-designed graph, chart, or plot should ANSWER all anticipatable questions, not prompt any. The only purpose of consolidating small slices is to avoid slice label collisions, which are more likely with small slices adjacent to one

another. Since I recommend ordering slices by size (Show them what's important), grouping the smaller slices together does increase the probability of label collisions. An always reliable way to prevent label collisions is the maximally informative legend in Case 5.

Though some ODS Graphics procedures allow the user to specify how much space is reserved to fit a legend, PROC SGPIE does not. If the content and font size of the legend require more than some threshold percent of the image area, the legend is omitted.

CALLOUT data labels are always outside of the pie. Case 5 has no data labels. All other cases use DATALABELLOC = OUTSIDE to assure maximum readability. Text overlaid on a coloured background is often less readable. When using a KEYLEGEND, assure that colour samples are big enough to assure colour distinguishability.



Case 1. Fully Labelled Pie Chart

Here is the code to create Case 1:

```
ODS RESULTS = OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE = LISTING
  GPATH = "C:\VIEWS_Feb2020\Results";

ODS GRAPHICS / RESET = ALL
  SCALE = OFF IMAGENAME = 'Case_1';
/* the above statement is also used for
  Cases 3, 4, & 5, with revised case number */

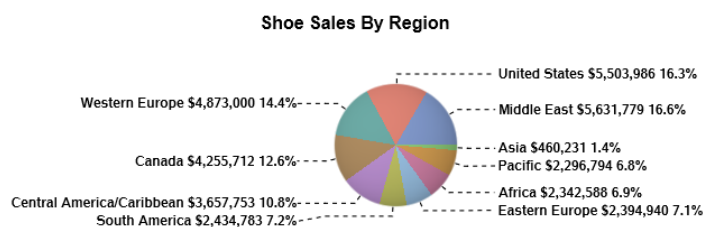
TITLE1 FONT = 'Arial/Bold' HEIGHT = 12PT
  'Shoe Sales By Region';
TITLE2 COLOR = WHITE FONT = 'Arial/Bold'
  HEIGHT = 8pt 'white space, not a title';
```

```

PROC SGPIE DATA = SASHELP.SHOES;
  PIE Region /
    RESPONSE = Sales
    SLICEORDER = RESPDESC
    DIRECTION = COUNTERCLOCKWISE
    /* COUNTERCLOCKWISE is the default */
    STARTANGLE = 0 /* This is the default */
    STARTPOS = EDGE /* Default is CENTER */
    OTHERPERCENT = 0 /* This is the default */
    DATALABELDISPLAY = ALL /* Default is only
    CATEGORY and RESPONSE, omitting PERCENT */
    DATALABELLOC = OUTSIDE /* Default is a mix
    of labels outside and inside, as deemed
    by the preference of the software.
    You can overrule it, as is done here. */
    DATALABELATTRS = (FAMILY = Arial
    WEIGHT = Bold
    SIZE = 9pt
    /* 9pt is Default */
    );
RUN;
ODS LISTING CLOSE;

```

The DIRECTION, STARTANGLE, and STARTPOS specified above duplicate the default actions of SAS/GRAPH. The concept of STARTPOS = CENTER is not supported in SAS/GRAPH. If you take the ODS Graphics defaults for DIRECTION and STARTPOS, the labels for Asia and Pacific collide. SIZE = 8pt can remove that collision.



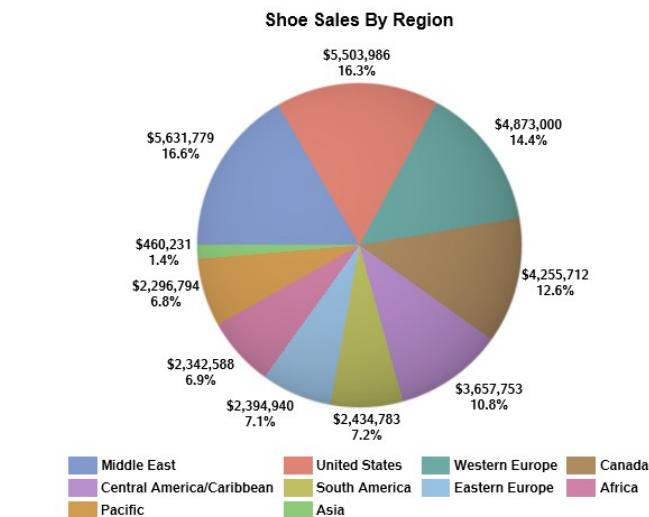
Case 2. Pie Chart with CallOut Labels with Image Height Optimized to Eliminate Wasted Vertical White Space

Here is the “inner” code to create Case 2:

```

ODS GRAPHICS / RESET = ALL
  SCALE = OFF IMAGENAME = 'Case_2'
  WIDTH = 640px HEIGHT = 240px;
  /* Defaults are 640px and 480px */
  /* Height 240px is Best for Width 640px.
  Increased height does not increase
  pie size, but increases useless white
  space above and below the pie. */
TITLE1 FONT = 'Arial/Bold' HEIGHT = 12PT
  'Shoe Sales By Region';
PROC SGPIE DATA = SASHELP.SHOES;
  PIE Region /
    RESPONSE = Sales
    SLICEORDER = RESPDESC
    STARTPOS = EDGE
    OTHERPERCENT = 0
    DATALABELDISPLAY = ALL
    DATALABELLOC = CALLOUT
    DATALABELATTRS =
      (FAMILY = Arial WEIGHT = Bold);
RUN;

```



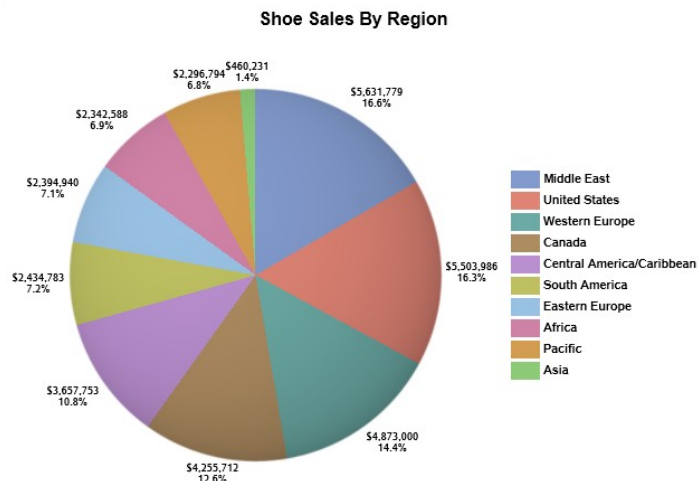
Case 3. Pie Chart with Legend at Bottom

Here is the “inner” code to create Case 3:

```

TITLE2 COLOR = WHITE FONT = 'Arial/Bold'
  HEIGHT = 8pt 'white space, not a title';
PROC SGPIE DATA = SASHELP.SHOES;
  PIE Region /
    RESPONSE = Sales
    SLICEORDER = RESPDESC
    STARTANGLE = 180 DIRECTION = CLOCKWISE
    /* Display slices to match directional
    order of values in the legend */
    STARTPOS = EDGE
    OTHERPERCENT = 0
    DATALABELDISPLAY = (RESPONSE PERCENT)
    /* If you omit DATALABELDISPLAY,
    NO Legend is provided, and
    the Data Labels are Region Sales. */
    DATALABELLOC = OUTSIDE
    DATALABELATTRS =
      (FAMILY = Arial WEIGHT = Bold);
  KEYLEGEND / NOBORDER TITLE=''
  /* POSITION=BOTTOM is the default.
  ACROSS=Integer is the default.
  The default integer is maximized
  based on lengths of the values
  and font Family, Size, & Weight.
  DOWN=Integer is the alternative. */
  FILLHEIGHT = 12PT
  FILLASPECT = GOLDEN
  VALUEATTRS =
    (FAMILY = Arial WEIGHT = Bold);
RUN;

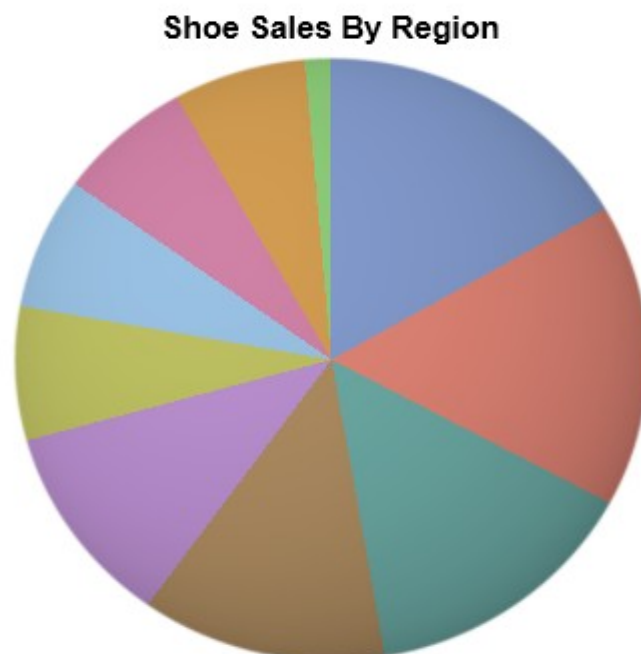
```



Case 4. Pie Chart with Legend at Right Side

Changes to Case 3 code to create Case 4:

```
TITLE2; /* White space is not needed. */
/* In PIE statement: */
STARTANGLE = 90 DIRECTION = CLOCKWISE
/* Display slices to match directional
order of values in the legend */
DATALABELATTRS =
(FAMILY = Arial WEIGHT = Bold SIZE = 7pt);
/* Prevents Data Labels collision
for Pacific & Asia */
/* In KEYLEGEND statement: */
POSITION = RIGHT
/* ACROSS = 1 is the default
when POSITION = RIGHT */
VALUEATTRS =
(FAMILY = Arial WEIGHT = Bold
SIZE = 8pt /* To fit legend */
);
```



01 - Middle East - 16.6% - \$5,631,779
02 - United States - 16.3% - \$5,503,986
03 - Western Europe - 14.4% - \$4,873,000
04 - Canada - 12.6% - \$4,255,712
05 - Central America/Caribbean - 10.8% - \$3,657,7
06 - South America - 7.2% - \$2,434,783
07 - Eastern Europe - 7.1% - \$2,394,940
08 - Africa - 6.9% - \$2,342,588
09 - Pacific - 6.8% - \$2,296,794
10 - Asia - 01.4% - \$460,231

Case 5. Maximally Informative Legend
Avoids any possible data label or callout collisions

Here is the pre-processing needed to create Case 5:

```
PROC SUMMARY DATA = SASHELP.SHOES;
CLASS Region;
VAR Sales;
OUTPUT OUT = Summary SUM = Sales;
RUN;

PROC SORT DATA = SUMMARY;
BY DESCENDING Sales;
RUN;

/* concatenate Rank, Percent, & Value
with the input Category (Region) */
DATA ToChart;;
RETAIN TotalSales 0;
LENGTH Rank_Category_Percent_Value $ 49;
SET Summary;
IF _N_ EQ 1
THEN DO;
TotalSales = Sales;
DELETE;
END;
```

```

Percent = (Sales / TotalSales) * 100;
Rank_Category_Percent_Value =
TRIM(LEFT(PUT(_N_ - 1, Z2.)))
|| ' - ' ||
TRIM(LEFT(Region)) || ' - ' ||
TRIM(LEFT(PUT(Percent, Z4.1))) || '%'
|| ' - ' ||
TRIM(LEFT(PUT(Sales, DOLLAR10.)));
RUN;

```

Here is the “inner” code to create Case 5:

```

ODS GRAPHICS / RESET = ALL
SCALE = OFF IMAGENAME = 'Case_5'
WIDTH=3.65in HEIGHT=5.41in;
/* minimum height to get the legend
   for this width of the image */
TITLE1 FONT = 'Arial/Bold' HEIGHT = 12PT
'Shoe Sales By Region';
PROC SGPIE DATA = ToChart;
  PIE Rank_Category_Percent_Value /
  RESPONSE = Sales
  SLICEORDER = RESPDESC
  STARTANGLE = 90
  DIRECTION = CLOCKWISE
  STARTPOS = EDGE
  OTHERPERCENT = 0
  DATALABELDISPLAY = NONE;
KEYLEGEND / NOBORDER TITLE=''
FILLHEIGHT = 8.2PT
/* maximum height to get the legend */
FILLASPECT = GOLDEN
VALUEATTRS =
  (FAMILY = Arial WEIGHT = Bold);
/* SIZE = 9pt is the default */
RUN;

```

LeRoy Bessler, Visual Data Insights™

Modernising SAS language applications and why WPS Analytics works so well

“Do I really get value from my SAS language environment?”

Let me try and answer this... there is no doubt that many SAS language users are considering modernising the analytics tooling they use. If you can relate to this there could be various reasons: maybe you have commercial and pricing pressures - you feel you are paying more than you should be, or you don't want to be tied to a particular software vendor; maybe you have technical drivers - you may be wanting to incorporate open-source technologies, new visual tools, or a move to cloud platforms.

The decision to make a change is not simple. Many businesses depend on analytical tools and processes for mission-critical, line-of-business processing. Down-time comes at huge risk and expense: even staying purely with SAS language and moving from one SAS language compiler vendor to another is not completely risk-free.

Moving to entirely different technologies brings the greatest risks, but there can be great benefits too. You may consider changing or incorporating:

- Programming languages
- Visual/graphical data science and analytics tools
- Compute and storage platforms

- Private or public cloud infrastructure
- Managed services or outsourcing

When weighing up the best potential courses of action, there are many factors to keep in mind:

- Cost and complexity of code and data migration
- Robustness of open-source package management and support
- Maintainability and reliability of free software
- Interoperability
- Performance and scalability
- Platform and hosting cost control
- Data security

Balancing risk, cost and benefit is complex.

The practical difficulties of maintaining 20+ years of SAS language applications and workflows created by business users can be daunting. You know the ones I mean? “multiple extracts from operational systems that get merged, filtered and summarised, passed to the finance team who tweak in large Excel files which end up in complex actuarial calculations for the latest price review session”. If this resonates, the WPS Analytics platform is the answer.

Why listen to me? Since leaving university as an applied statistician in the early 1990s, most of my professional career has involved using SAS language tools in some shape or form – whether working for tier-one insurers, SAS Institute, World Programming or one of the big four management consultancies. My time has been spent variously installing, developing, supporting and presenting on most aspects of SAS language tools ranging from installing SAS System software version 5.18 on mainframes back in 1995, to using the latest versions of WPS Analytics software on desktops and servers, AWS and Azure. I have witnessed the introduction and rise of analytics programming languages, graphical tools, workflows, enterprise data integration, business intelligence suites, web-based data science tools, end-to-end solutions in fraud management, financial risk monitoring, customer marketing, high-performance visualisation and in-memory analytics. I can't say I've seen it all, but I've seen quite a lot of it!

So what, I hear you say. I frequently hear people say that they love the power and flexibility of the SAS language but due to rising costs and internal demands to reduce IT spend, they are looking for ways to control costs associated with running their SAS language programs. At the same time there is an interest in moving at least some data and processes from on-premises systems to public cloud-based hosted platforms and incorporating “free” open-source technology.

What can you or should you do? I believe you need three components for a successful and effective data science and analytics environment:

[1] One development tool for both visual workflows and coding

Consider a full life-cycle analytics scenario: You want to build a complex analytical process (workflow) using a high-productivity GUI tool using your own logic for data acquisition from Excel, databases, Snowflake and S3, prepare data to shape and cleanse it ready for analysis, develop advanced predictive models and explain those models with decision trees, seamlessly share data between R, Python and SAS code for modelling tasks,

publish results to Power BI, Tableau or Qlik and make results available for external processing via APIs. WPS Analytics consolidates these capabilities into a single toolset. With its own fully-integrated SAS language compiler, you can combine R and Python with the SAS language and with other commercial and non-commercial tools and technologies to create single executable programs or visual workflows.

With the WPS Analytics approach, it's simple to see how this differs from the complex approach of needing multiple products offered by other vendors. The open interoperability philosophy of WPS Analytics puts the power of commercial tools together with open source technology at the fingertips of data scientists and business analysts for ease of use and better productivity. All this at a highly-competitive price.

[2] One tool for deployment, integration and management

Let's consider "productionisation": Having developed a valuable analytics program or predictive model, to get real value back from it you need to deploy it to a managed production environment that gets maximum exposure and use out of your predictive model. This should likely involve Restful APIs so that other systems, business users and external services can easily consume the deployed program or model. WPS Analytics Hub supports clear and easy deployment, operation and governance of programs and models into production, with version control integration and clear separation of development, test and production environments.

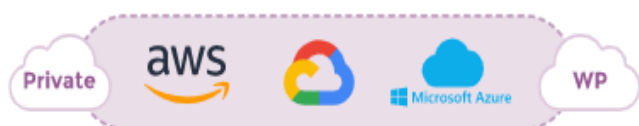
[3] One pricing structure that controls cost whilst you create value

Modern public and private cloud infrastructure enables businesses to scale analytics processes on-demand for development, test and production. This can be a difficult fit with traditional licensing models that typically involve an annual subscription for a fixed size of compute environment or number of "seats" (users). Your data scientists and analysts may, for example, want to work on a large analytical program for just two months. Paying a full annual software licence subscription fee just to cover this time-limited project may seem poor value, although this is almost universally still the situation. In addition, increasingly a problem with cloud deployments is internal governance controlling who within an organisation is permitted to leverage a cloud environment. Without a governance programme, businesses risk racking up unexpected cloud platform and/or software licensing bills and falling out of licensing compliance.

For enterprise customers WPS Analytics is available with new flexible enterprise cloud pricing models that are designed to accommodate the issues arising with elastic compute and storage infrastructure. WPS Analytics enterprise cloud pricing works by charging only for production analytics, for data and users. We understand that this may not suit everyone, and our traditional pricing model is still available.

WPS Analytics for enterprise cloud computing is delivered in 3 easy components

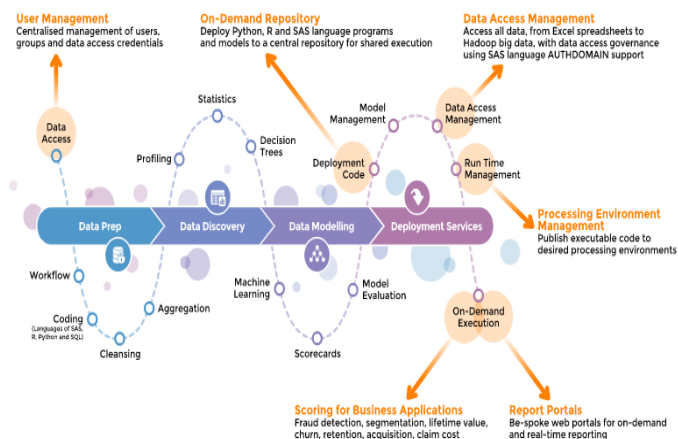
1. Choose an infrastructure cloud (AWS, Google, Azure).



2. Subscribe to a package for production jobs and transactions.



3. Unlimited analytics for development and testing.



Modernising your analytics environment doesn't have to be a step into the unknown and can bring huge benefits.

Migrating your existing SAS language programs to WPS Analytics is easily achieved in a simple three step approach:

1. Run the WPS Analytics code analyser on all your existing SAS language programs to generate a detailed compatibility report. Code analysis of thousands of programs takes just seconds to generate a heads-up view on the functional fit of WPS Analytics to your needs.
2. Deploy WPS Analytics for free and run some of your selected SAS language programs to gain confidence in the software's capability.
3. Migrate your applications, programs and data with full support from experts at World Programming if necessary. Migration projects are only signed-off when you are 100% satisfied.

And finally...

Maybe consider WPS Analytics and how they might relate to your current needs and how you plan to future-proof and retain the investment you've made in your analytics environment. Don't underestimate how valuable your data science team and analytics applications are to your business!

For more information, get in touch with our sales team (sales@worldprogramming.com) and let us show you how we can support your investment in analytics or simply sign-up for a free evaluation of WPS Analytics.

Scott Thomson, World Programming

Adapted from www.worldprogramming.com/blog/why_wps

Ask The Consultant

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

Managing SAS Options in Macros

Q: How do you make certain SAS options changed in macros don't impact the calling program?

A:[SAS] The current settings of all SAS options are stored in a SAS dictionary table called **dictionary.options** (for PROC SQL), or **sashelp.voptions** (for PROC SQL and Data Steps). The SAS-supplied function **GETOPTION()** can then be used to extract the current setting. The new option settings can then be specified and used, and then the **OPTION** statement will allow the previous settings to be restored from the saved macro variables values:

```
OPTIONS PAGESIZE=60 LINESIZE=132
        ORIENTATION=landscape;

%MACRO papermac(in=, ps=, ls=,
               orient=landscape);

    %LOCAL ps_in ls_in orient_in;
    %LET ps_in =
        %sysfunc(GETOPTION(pagesize, keyword));
    %LET ls_in =
        %sysfunc(GETOPTION(linesize, keyword));
    %LET orient_in =
        %sysfunc(GETOPTION(orientation, keyword));

    OPTIONS PAGESIZE=&ps. LINESIZE=&ls.
            ORIENTATION=&orient.;

    PROC PRINT DATA = &in.;
    RUN;

    OPTIONS &ps_in. &ls_in. &orient_in.;

%MEND papermac;

%papermac(in=sashelp.cars, ps=80, ls=72,
         orient=portrait)
```

There are no complicated programming techniques used here, just common sense. Before writing a new SAS macro think about what it will create and use, including SAS options, and make sure that only the ones that the user can specify can "escape". Writing macros that "play nice" will keep you popular and your macro users productive.

Philip R Holland (Newsletter Editor)

Extract from "Writing Reusable Macros" by Philip R Holland

Formats, Options, and Functions

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

The **FLOATw.d** format displays a numeric value as text, and the corresponding informat reads text into a numeric value. The text value is a hexadecimal representation of a binary number that is stored in IEEE form. The text value is equivalent to that produced by other programming languages:

- SAS: **FLOAT4.**
- Fortran: **REAL+4**
- C: **float**
- IBM 370 ASM: **E**
- PL/I: **FLOAT BIN(21)**

[Windows] The **-TOOLDEF** configuration option controls the position of the Toolbox in interactive SAS for Windows, with the default setting being **TOP RIGHT**. However, it has no effect at all if you are using a toolbar instead the Toolbox.

[Windows] The routine **CALL MODULE()** and functions **MODULEC()** and **MODULEN()** allow SAS to run external compiled routines and functions in Windows Dynamic Link Libraries (*.dll).

In Brief

- Back issues of *VIEWS News* are available from the VIEWS UK forum at **blog.hollandnumerics.org.uk**. Please register on the blog, and then request free access to the forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 33,830 members. Everyone is welcome to join, but discussions are restricted to SAS-related topics: **www.linkedin.com/groups?gid=70702**.
- In the UK and Ireland the SUGUKI group is 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**.
- WPS Community Edition provides users with free access to WPS Analytics software 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 six months. At the end of this term, users may be invited to renew their licence and to upgrade to the latest available version of the software. No technical support service is offered with the Community Edition. Download from **www.worldprogramming.com/try-or-buy/editions**. Note that the software is made available without warranty or indemnity.

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 2020

18 **PhilaSUG**, Malvern, PA, USA

See the PhilaSUG web site at philasug.org for further details. Please register by 16 March.

26 **BSUG**, Tervuren, Belgium

2nd Anniversary meeting at SAS Belgium offices. See the BSUG web site at meetup.com/BeLux-SAS-User-Group for further details.

May 2020

7 **SUGUKI**, London, UK

See the SUGUKI web site at meetup.com/SUGUKI for more details. Speakers to be announced.

10-13 **PharmaSUG US**, San Francisco, CA, USA

See the PharmaSUG web site at pharmasug.org for further details.

June 2020

4 **SUGUKI**, London, UK

See the SUGUKI web site at meetup.com/SUGUKI for more details. Speakers to be announced.

22-23 **SAS Forum UK**, Birmingham, UK

For more details see www.sas.com/uk/events.

July 2020

2 **SUGUKI**, London, UK

See the SUGUKI web site at meetup.com/SUGUKI for more details. Speakers to be announced.

August 2020

28-29 **PharmaSUG China**, Shanghai, China

For further details see the PharmaSUG China web site at pharmasug.org/china.

November 2020

8-11 **PhUSE EU Connect**, Liverpool, UK

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

Stop Press!

COVID-19 Impact on SAS Events

This week a number of SAS events have been cancelled due to the novel coronavirus COVID-19, which is why the above Diary section is not as full as expected.

PhUSE US Connect in Orlando FL on 8 to 11 March, and SAS Global Forum in Washington DC on 29 March to 1 April, have now both been cancelled. The organisers of PharmaSUG US in San Francisco CA on 10 to 13 May are also watching developments, but at this time it is still going ahead.

However, the SUGUKI London meeting on 2 April had been cancelled, as it conflicted with the dates of SAS Global Forum, but may now be reinstated. Please watch meetup.com/SUGUKI for more details.

Philip R Holland (Newsletter Editor)

Contacts

LeRoy Bessler

Email: le_roy_bessler@wi.rr.com

Scott Thomson

Web:

www.worldprogramming.com

Email:

sales@worldprogramming.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.