

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 49, 1st Quarter 2010

Figure 1. Simple Annotation

A new year and I have another newsletter issue for you filled with SAS news, hints and tips that I hope you will find interesting. In this issue of *VIEWS News* there are articles from our regular contributors LeRoy Bessler, Murphy Choy and the Amadeus Software Consultancy Team, who have written articles on graphical impact, encryption algorithms and format merges, and Excel functions available in SAS 9.2, respectively. Finally, my regular Formats, Options, and Functions section has more unusual features of SAS software.

I will be presenting later this year at the SAS Global Forum in Seattle, as well as at the PhUSE Single Day Events in Copenhagen and Hertfordshire, so please come and talk to me about *VIEWS News*.

As an added bonus Amadeus Software is running a Prize Draw especially for *VIEWS News* readers to win free places at their webinars. For details please read all about it in the Stop Press!! section near the end of this issue, and good luck!

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 newsletter@views-uk.org. A list of subject suggestions for your articles can be found on the *VIEWS* web site, and anyone is very welcome to add to that list by sending emails to newsletter@views-uk.org with your own questions.

Philip R Holland (Newsletter Editor)

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

Did You Know?

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

Axis-Free Horizontal Bar Charts: Visual Impact + Precise Numbers

In *VIEWS News*, I have previously shown minimalist horizontal bar charts, but they always involved supplying bar names and bar values collinearly with, and outside of, their respective bars. Here are ways to instead put the information inside the bars or above them.

Top Twelve African Nations by Population (Millions)

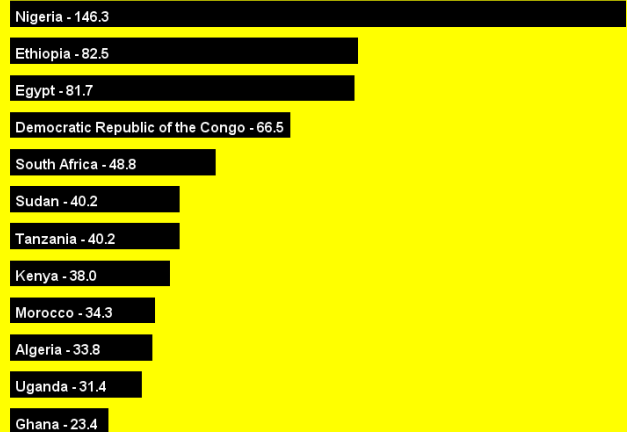
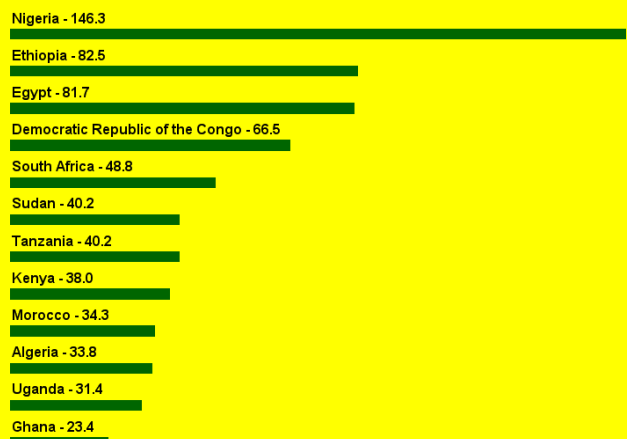


Figure 2. Requires TXT2CNTL Annotate Function

Top Twelve African Nations by Population (Millions)



These graphs combine Visual for quick, easy inference and Precise Numbers for correct, reliable inference. That is the power of graphs designed and constructed for Communication Effectiveness. Use of the one-word option DESCENDING on the HBAR statement of PROC GCHART implements my design principle of Show Them What Is Important. The only time that you would not use DESCENDING or ASCENDING to help most effectively convey your message/information might be for a comprehensive look-up of the populations of all 54 African nations, when a sort by nation name could be preferred. Such alphabetical ordering would certainly be mandatory for graphic list of populations for all 240 nations of the world.

I have long recommended horizontal bar charts in place of vertical bar charts in order to avoid the problem of fitting bar chart labels. There is no excuse for relying on the desperate

remedy of the unreadable vertical labels applied to vertical bar charts. With a horizontal bar chart, you can always fit long labels. The set of bars is simply drawn shorter to accommodate the longest label. However, the solution above allows you to both fit long labels (such as Democratic Republic of Congo) and use the full graph area width for your longest bar.

Before I present the code, I want to mention a few things. Using the NOAXES option on the HBAR statement would perplexingly shrink the available space to draw the graph. Therefore, use my method for axis suppression. For more about the SAS/GRAPH Annotate facility, please see the SAS/GRAPH documentation or Art Carpenter's book "Annotate: Simply the Basics" available from SAS Press. This code will run the same in both SAS 9.1.3 and 9.2, but the text will be drawn slightly smaller in SAS 9.1.3.

Here is the code used to create the graphs above:

```
/* Input data set WORK.ToChart contains two
variables: Nation which is 32 bytes of
character data and Population which is
numeric. */

GOPTIONS RESET = ALL;
GOPTIONS DEVICE = GIF;
/* Put the graph in a box. */
GOPTIONS BORDER;
/* Make background color RGB Yellow. */
GOPTIONS CBACK = CXFFFF00;
GOPTIONS GSFMODE = REPLACE GSFNAME = anyone;
TITLE1 FONT = 'Arial/Bold'
/* Insert space at top. */
HEIGHT = 3 PCT ' '
/* Force a new line. */
JUSTIFY = CENTER
HEIGHT = 5 PCT
/* Make text RGB blue. */
COLOR = CX0000FF
"Top Twelve African Nations by Population
(Millions)";
FOOTNOTE1 ANGLE = +90 FONT = NONE
HEIGHT = 1 ' ' ;
/* This is a blank "right-side-note" to
increase space at the side of the image. */
FOOTNOTE2 ANGLE = -90 FONT = NONE HEIGHT = 1
' ' ;
/* This a blank "left-side-note". */
/* Suppress all AXIS paraphernalia: */
AXIS1 LABEL = NONE MAJOR = NONE MINOR = NONE
STYLE = 0 VALUE = NONE;
/* In PROC GCHART MAXIS & RAXIS are assigned
the AXIS1 parameters. */

DATA AnnoInside;
LENGTH function $5 style $32
/* length for two-blank prefix */
text $42;
RETAIN function 'LABEL' when 'A'
xsys ysys '2' hsys 'B' x 0
position '6' size 3
/* Use RGB White for the text. */
color 'CXFFFFFF' style "'Arial/Bold'";
SET ToChart;
text = ' ' || TRIM(LEFT(Nation)) || ' - ' ||
TRIM(LEFT(PUT(Population / 1000000, 5.1)));
yc = Nation;
RUN;
```

```
FILENAME anyone "C:\HbarAnnoInside.gif";
/* Use RGB Black for the solid bars. */
PATTERN1 VALUE = SOLID COLOR = CX000000;

PROC GCHART DATA = ToChart
ANNOTATE = AnnoInside;
HBAR Nation / SUMVAR = Population NOSTATS
WIDTH = 2 SPACE = 1 DESCENDING
MAXIS = AXIS1 RAXIS = AXIS1;

RUN;
QUIT;

DATA AnnoAbove;
LENGTH function $8 style $32
/* length for one-blank prefix */
text $41;
RETAIN xsys '2' hsys 'B' x 0 position '3'
size 3 style "'Arial/Bold'"
/* Use RGB Black for the text. */
color 'CX000000';

SET ToChart;
function = 'MOVE';
ysys = '2';
midpoint = Nation;
OUTPUT;
function = 'MOVE';
ysys = '7';
y = 0.4;
OUTPUT;
function = 'CNTL2TXT';
OUTPUT;
function = 'LABEL';
text = ' ' || TRIM(LEFT(Nation)) || ' - ' ||
TRIM(LEFT(PUT(Population / 1000000, 5.1)));
OUTPUT;

RUN;

FILENAME anyone "C:\HbarAnnoAbove.gif";
/* Use RGB Medium Dark Green for the bars. */
PATTERN1 VALUE = SOLID COLOR = CX006600;
PROC GCHART DATA = ToChart
ANNOTATE = AnnoAbove;
HBAR Nation / SUMVAR = Population NOSTATS
WIDTH = 0.8 SPACE = 2.2 DESCENDING
MAXIS = AXIS1 RAXIS = AXIS1
/* Override default bar outline colour,
which is black, to match the bar fill
colour. */
COUTLINE = CX006600;

RUN;
QUIT;
```

LeRoy Bessler

Classical Cipher in SAS: Alberti's cipher

Alberti's cipher is a cipher first described by Leon Battista Alberti in the mid-15th century. The method uses a wheel with 2 circles with the inner circle marking the original alphanumeric list to be substituted, and the outer circle being the one that indicates the value that is to be used for substitution. After the initial shift, the substitution wheel changes for every Nth word by a shift of K. Due to this ever changing shifts, it is a relatively stronger encryption compared to our Caesar's cipher (which was described in *VIEWS News* issue 48).

```

/*****
Alberti's Cipher
-----
Parameter  Description
-----
input      Input data
var        Variable to be encrypted/decrypted
shift_1    Preliminary starting position
shift_2    Next position to shift
shift_3    Subsequent shifts
mode       ENCRYPT/DECRYPT
*****/

%MACRO alberti_cipher(input=, shift_1=
                      ,shift_2=, var=, mode=);

  DATA &input.;
    SET &input.;
    /* Retaining cypher text */
    RETAIN cipher _CHAR_;
    RETAIN count;

    /* ENCRYPTION SECTION */
    %IF &mode. EQ ENCRYPT %THEN %DO;
      /* Declaring the original string */
      original = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
      /* First time initialization */
      IF _N_ = 1 THEN DO;
        cipher =
          COMPRESS(SUBSTR(original, &shift_1. + 1)
        %DO I = 1 %TO &shift_1.;
          || SUBSTR(original, &i., 1)
        %END;
      );
      encrypted_&var. =
        TRANSLATE(&var., original, cipher);
      count = 1;
    END;

    /* Every Nth time reinitialization */
    ELSE IF count = &shift_2. THEN DO;
      cipher =
        COMPRESS(SUBSTR(cipher, &shift_3. + 1)
      %DO I = 1 %TO &shift_3.;
        || SUBSTR(cipher, &i., 1)
      %END;
    );
      encrypted_&var. =
        TRANSLATE(&var., original, cipher);
      count = 1;
    END;

    /* Encryption between every
       reinitialization */
    ELSE DO;
      encrypted_&var. =
        TRANSLATE(&var., original, cipher);
      count = count + 1;
    END;
  %END;

%ELSE %IF &mode. EQ DECRYPT %THEN %DO;
  /* Declaring the original string */
  original = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  /* First time initialization */
  IF _N_ = 1 THEN DO;
    cipher =
      COMPRESS(SUBSTR(original, &shift_1. + 1)
    %DO I = 1 %TO &shift_1.;
      || SUBSTR(original, &i., 1)
    %END;
  );
  decrypted_&var. =
    TRANSLATE(&var., cipher, original);
  count = 1;
END;

/* Every Nth time reinitialization */
ELSE IF count GE &shift_2. THEN DO;
  cipher =
    COMPRESS(SUBSTR(cipher, &shift_3. + 1)
  %DO I = 1 %TO &shift_3.;
    || SUBSTR(cipher, &i., 1)
  %END;
);
  decrypted_&var. =
    TRANSLATE(&var., cipher, original);
  count = 1;
END;

/* Decryption between every
   reinitialization */
ELSE DO;
  decryption_&var. =
    TRANSLATE(&var., cipher, original);
  count = count + 1;
END;
%END;
DROP count cipher original;
RUN;
%MEND;

/*-----*/
%alberti_cipher(input=class, shift_1=5
               ,shift_2=6, shift_3=4, var=name
               ,mode=ENCRYPT);

```

```

decrypted_&var. =
  TRANSLATE(&var., cipher, original);
count = 1;
END;

/* Every Nth time reinitialization */
ELSE IF count GE &shift_2. THEN DO;
  cipher =
    COMPRESS(SUBSTR(cipher, &shift_3. + 1)
  %DO I = 1 %TO &shift_3.;
    || SUBSTR(cipher, &i., 1)
  %END;
);
  decrypted_&var. =
    TRANSLATE(&var., cipher, original);
  count = 1;
END;

/* Decryption between every
   reinitialization */
ELSE DO;
  decryption_&var. =
    TRANSLATE(&var., cipher, original);
  count = count + 1;
END;
%END;
DROP count cipher original;
RUN;
%MEND;

/*-----*/
%alberti_cipher(input=class, shift_1=5
               ,shift_2=6, shift_3=4, var=name
               ,mode=ENCRYPT);

```

Conclusion

This is one of the more complicated cipher that can be implemented in SAS. Its complexity in both resisting brute force and frequency methods makes it a stronger cipher. This cipher provide a strong cipher to encrypt information.

Murphy Choy, University College Dublin

Ask The Consultant

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

Excel functions in SAS 9.2

Q: Can you use Excel functions in SAS?

A: [SAS 9.2] In SAS 9.2, many Excel functions are generally available in Base SAS, including in the data step. These functions were originally implemented for use by PROC SYLK, which was available in earlier versions of SAS.

SAS 9.2 has a new SASHELP view called VFUNCS, which looks at DICTIONARY.FUNCTIONS. You might expect to find the Excel functions among those displayed by this view, but you would be wrong. If you have problems with the SASHELP views under SAS 9.2, SAS Note 32190 may help.

You need instead to look at the data set SASHELP.SLKWXL. For SAS 9.2, the Excel functions appear to have been implemented using PROC FCMP, and the SLKWXL dataset is, in effect, a function library. You should find that it defines about

40 functions, all of them with names ending in “_SLK”, each corresponding to an Excel function whose name lacks that suffix.

You will need to use the CMPLIB option to make the functions defined in the SLKWXL library available. Once this has been done, they can be used in data steps in the same way as any other SAS function. This example demonstrates COUPDAYS_SLK, the SAS implementation of Excel’s COUPDAYS function.

```
OPTIONS CMPLIB = sashelp.slkwxl;
DATA _NULL_;
  /* Settlement date */
  settledate = '15Oct2009'D;
  /* Maturity date */
  maturedate = '31Dec2009'D;
  /* Frequency = quarterly */
  freq = 4;
  /* Basis = 'Actual/actual' */
  basis = 1;
  days = COUPDAYS_SLK(settledate, maturedate
                      ,freq, basis);
  PUT 'Number of days in coupon period=' days;
RUN;
```

In common with other functions defined using PROC FCMP, the Excel functions can be used in data steps and in certain other SAS procedures, including PROC SQL. There are however other procedures which do not support them, and they are not available via %SYSFUNC in the SAS macro language.

Amadeus Software Consultancy Team

Formatting a merge with SAS macro

Q: How would you reduce the chance of programming errors in a merge?

A: High speed data set merges can be achieved easily using either hash tables or format merges. However both techniques are very complicated and requires careful programming to ensure that the output is exactly the way you want it to be. By combining the codes into a SAS macro, we can then control our output in a more consistent manner. The SAS macro below specifically deals with data set merges using formatting.

High speed data set merge has always been one of the main selling point of SAS especially in data management tools. Typically the normal sort-merge and SQL merges are very effective in handling data within a certain size. Once the size is breached, the merges are typically quite slow and arduous for the SAS programmers who likes to do things quickly. This is especially painful for those merges where a single field is tagged to a bigger data set. To solve this problem, many SAS programmers have used a SAS format step combined with a Data Step to merge and create the new flag. This approach is more commonly known as the Format-Merge approach.

The format-merge approach begins as a regular Data Step where the variables are sorted by the merging key and any duplicates removed via the use of NODUPKEY option in the PROC SORT step. After this step, a few additional variables have to be created to make sure the data set is in a usable form for PROC FORMAT. Once this is done, the data set is now ready to be used in the PROC FORMAT step. Below is an example of how the format data set is being created.

```
DATA keyfile;
  SET group1;
```

```
  fmtname = 'KEY';
  RENAME id = start;
RUN;

PROC SORT DATA = keyfile NODUPKEY;
  BY start;
RUN;
```

After this, the dataset is directly sent to the PROC FORMAT statement which then generates the format. Below is an example of this step.

```
PROC FORMAT CNTLIN = keyfile;
RUN;
```

Typically after this step, the programmer will then use a data step with a PUT function to tag the values.

```
DATA new_overall;
  SET group2;
  LENGTH g 8;
  g = PUT(id, key.);
  IF g = id THEN g = .;
RUN;
```

After this, the whole merge process is over.

However, occasionally such steps can be done incorrectly resulting in wrong information being tagged on the data set. To prevent this, it is possible to achieve the Format-Merge process by using a macro.

```
/******
Format-Merge macro
-----
Parameter      Description
-----
main_file      Main file to have values tagged
side_file      Tag value data set
merge_key      Merge key
var            Variable to be tagged
rename_var     New name for tagged variable
*****/

%MACRO fmt_merge(main_file=, side_file=
                ,merge_key=, var=
                ,rename_var=);

/* Format data set */
DATA keyfile;
  SET &side_file. (KEEP = &merge_key &var.);
  RETAIN fmtname 'KEY';
  RENAME &merge_key. = start &var. = label;
RUN;

/* De-duplication */
PROC SORT DATA = keyfile NODUPKEY;
  BY start;
RUN;

/* Creating the format */
PROC FORMAT CNTLIN = keyfile;
RUN;

/* Applying the tagged values */
DATA &main_file.;
  SET &main_file.;
  &rename_var. = PUT(&merge_key., key.);
  IF &rename_var. = &merge_key.
    THEN &rename_var. = .;
RUN;
%MEND;
```

```

/*-----*/
%fmt_merge(main_file=dsn1, side_file=dsn2
,merge_key=col1, var=col2
,rename_var=col3);

```

By using a macro, we can then reduce human errors. This macro essentially combines all the steps to make this type of merge simpler for beginners to use without errors.

Murphy Choy, University College Dublin

Formats, Options, and Functions

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

[SAS 9] The PCTL() function returns the percentile of a list of arguments that corresponds to the percentage.

[SAS 9.2] The \$BASE64Xw. informat and format read and write Base 64 encoded ASCII character values.

[SAS 9.2] The ODS EXCLUDE statement can be instructed to exclude objects based on an arithmetic or logical expression using the WHERE= option. This option requires one or more of the following subsetting variables: _LABEL_ (object label), _LABELPATH_ (object label path), _NAME_ (object name) or _PATH_ (full or partial object path).

News

SAS Spoken Here — Enjoy the Gemütlichkeit



Milwaukee is known for its Gemütlichkeit, or comfortable ambiance and friendliness. As a warm-up or rehearsal for PhUSE in Berlin, you are cordially invited to attend, and optionally present a paper, at the 2010 MidWest SAS Users Group Conference in Milwaukee, on October 10-12.

Milwaukee was once known as the "Deutsches Athen" (German Athens) of America. Even in the early twentieth century, this popular immigration destination on the western shore of one of America's great inland seas, Lake Michigan, had more German speakers and German-language newspapers than

English speakers and English-language newspapers. Nevertheless, English and SAS will this year again be the official languages of the MidWest SAS Users Group Conference. ☺

For more information or to discuss a paper proposal, please contact LeRoy Bessler at le_roy_bessler@wi.rr.com or +1-414-351-6748.

LeRoy Bessler

ODS Book Review

The following reviews can be found inside the front cover of the new ODS book by Lauren E Haworth, Cynthia L Zender and Michele M Burlew, entitled "Output Delivery System: The Basics and Beyond", ISBN 978-1-59994-660-3.

LeRoy Bessler:

"Lauren Haworth, author of the previous pioneering book on ODS, has been joined by SAS's own Cynthia Zender, expert on all aspects of SAS reporting, and Michele Burlew, the versatile and prolific author of several other SAS books. With a team like this SAS users could expect an outstanding resource, and we got it."

Philip R Holland, Holland Numerics:

"I haven't enjoyed a technical book as much as I enjoyed reading this one. I've been impressed with the writing styles of all 3 authors, and this book is another easy read. What impressed me was the detail incorporated into every example, and, just when I had a burning question about one example, there was an example there immediately to answer it. I've used ODS a lot myself, but I'm learning more in every chapter. This is good!!"

PhUSE News

PhUSE 2010 Single Day Events (SDE)

The PhUSE Society is delighted to announce that 4 "Single Day Events" have been scheduled in to the calendar to run in the first half of 2010:

- Brussels, Belgium - 23 February 2010
- Basel, Switzerland - 9 March 2010
- Copenhagen, Denmark - 15 March 2010
- Welwyn Garden City, UK - 11 May 2010

The 2010 Single Day Events further demonstrate PhUSE's commitment to fulfilling the primary objective of promoting and facilitating "Software Exchange" within the pharmaceutical industry across Europe.

See www.phuse.eu for more details.

PhUSE Committee

In Brief

- Back issues of *VIEWS News* are available from the *VIEWS* web site, and also from the *VIEWS News* page at www.sasCommunity.org/wiki/VIEWS_News
- Andrew Ratcliffe's excellent email newsletter "NOTE:" has been converted into a blog at notecolon.info.

- More SAS hints and tips can be found on the SAS FAQ pages www.hollandnumerics.com/sasfaq and also on the Amadeus Software Tips pages at www.amadeus.co.uk/TipsList.aspx?ST=1
- There is a social networking groups for SAS users on Ning called **SAS Professionals**, originally intended for SAS users in the UK, at www.sasprofessionals.net, but currently totalling more than 2,200 members worldwide.
- Phil Holland has started a group page on Squidoo for other pages related to SAS, and would like to make it a central resource for SAS hints, tips, links and books at www.squidoo.com/groups/SAS_users
- Phil Holland has also created the SAS Contractors group on LinkedIn specifically for independent SAS consultants in the UK and Europe. Requests to join the group should be made at www.linkedin.com/groups?gid=2481512, and the associated SAS Contractor Resources page, which is open to all, can be found at www.squidoo.com/sas_contractors
- SAS users who use Twitter can now join a “twibe” at twibes.com/SAS, which filters your tweets down to those containing “SAS”, “EnterpriseGuide” and “PROC”.
- Phil Mason has a SAS tips newsletter, which can be requested by going to www.woodstreet.org.uk and then clicking on Tips. Previous tips can also be found on the web site.

Stop Press!

SAS Programming webinars

Amadeus, SAS Business Intelligence specialists, have announced their new series of Webinars for 2010, and there are **free places available to VIEWS News readers**.

Amadeus have built up a reputation for providing quality fundamental to advanced training in SAS software for two decades, including their popular Tips & Tricks Seminars, and now the expertise which many companies have enjoyed for so long is available on-line. The first 2 Webinar titles are: “SAS Programming Tips and Tricks for Base SAS” and “SAS Programming Tips & Tricks for the Output Delivery System”, which will run from 1000-1200 GMT, the standard cost will be £80 plus VAT.

SAS Programming Tips and Tricks for Base SAS

Audience

Whilst not for complete beginners, this webinar will offer exposure to a variety of Data Step, Proc Step and macro examples used in everyday SAS programming.

Prerequisites

The most benefit will be gained from this webinar by those with some previous experience of SAS programming.

Objectives

Being Productive with Base SAS.

Demonstrates automating the enhanced editor using abbreviations and editor macros, SAS9 Data Step Functions – a select few from the 500+, manipulating data in the Data Step,

joining tables and robust programming when implementing business rules.

Making the Most from Macros

Various uses, best practices for writing, bullet proofing, storing and sharing SAS macro libraries are demonstrated.

Next date: 18 March 2010

SAS Programming Tips & Tricks for the Output Delivery System

Audience

This webinar is perfect for those who export results from SAS to external formats such as Microsoft Excel, Word and PDF formats, amongst others.

Prerequisites

The most benefit will be gained by those with some previous experience of SAS programming.

Objectives

Writing Professional Reports

Shows techniques for creating reports and graphs with SAS procedures and the Output Delivery System which not only look professional and of publication quality, but also make use of some output delivery system features to add visual clarity and impact to all formats of reports.

Sharing SAS Results with Microsoft Office

Probably the question asked most frequently of our consultants is that of transferring data from SAS to Excel; this webinar will show you how.

Next date: 30 March 2010

To be entered into the Amadeus draw for a free Webinar place please email: info@amadeus.co.uk with the subject title: free VIEWS webinar place.

Visit www.amadeus.co.uk/Webinars.aspx for more details.

Louise Harvey, Amadeus Software

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 2010

9 **PhUSE Single Day Event**, Basel, Switzerland

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

15 **PhUSE Single Day Event**, Copenhagen, Denmark

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

18 **SAS Programming Tips and Tricks for Base SAS**

This is a webinar. For further details see the Amadeus Software web site www.amadeus.co.uk/Webinars.aspx

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

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

30 **SAS Programming Tips and Tricks for the Output Delivery System**

This is a webinar. For further details see the Amadeus Software web site www.amadeus.co.uk/Webinars.aspx

April 2010

11 **Preparing SAS Programmers for the Pharmaceutical Industry**, Seattle, WA, USA

This is a pre-conference class by Sunil Gupta at the SAS Global Forum. See tinyurl.com/yhazq5o for more details.

11-14 **SAS Global Forum**, Seattle, WA, USA

The SAS Global Forum (SGF, formerly called SUGI) is held each year and focuses on the technical aspects of SAS software. See www.sasglobalforum.org for more details.

May 2010

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

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

11 **PhUSE Single Day Event**, Hertfordshire

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

June 2010

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

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

October 2010

10-12 **MWSUG Conference**, Milwaukee, WI, USA

The 2010 MWSUG (MidWest SAS Users Group) Conference is open to all. Further details will be available shortly at www.mwsug.org.

18-20 **PhUSE Conference**, Berlin, Germany

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

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

Contacts

Amadeus Software

Email: info@amadeus.co.uk
Tel: +44-(0)1993-878287
Web: www.amadeus.co.uk

LeRoy Bessler

Email: le_roy_bessler@wi.rr.com

Murphy Choy

University College Dublin

Email: goladin@gmail.com

PhUSE

Email: office@phuse.eu
Web: www.phuse.eu

SAS UK

Tel: +44-(0)1628-486933
Web: www.sas.com/uk

VIEWS

Web Site

Web: www.views-uk.org

Chairman

Email: chair@views-uk.org

Newsletter Editor

Email: newsletter@views-uk.org

Membership secretary

Email: membership@views-uk.org

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