

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 81, 1st Quarter 2026

This newsletter is the home of Foundation SAS and brings you information about making the most of SAS 9 and Altair SLC. This newsletter issue includes an article from LeRoy Bessler about distorted 3D pie charts, and we welcome a new *VIEWS* Consultant, Pablo Orosco Joerger, with an article about integrating SAS and R programming languages in Altair SLC. 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.

Did You Know?

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

Integration of the SAS Language and R within Altair SLC: A Survival Analysis Example

[Altair SLC 2026 with R-4.4.3 (survival, ggplot2, scales)] This article presents a concise yet expressive example of a survival analysis workflow implemented using the SAS Language within Altair SLC, seamlessly integrated with R via PROC R. The intent is to demonstrate how structured, transparent SAS Language data preparation can be combined with the statistical and graphical strengths of R, resulting in outputs that are both analytically sound and visually clear.

All figures and descriptions in this article are derived directly from the accompanying SLC SAS Language + R program. No results are invented or altered; the visualisations shown are exactly those produced by the program.

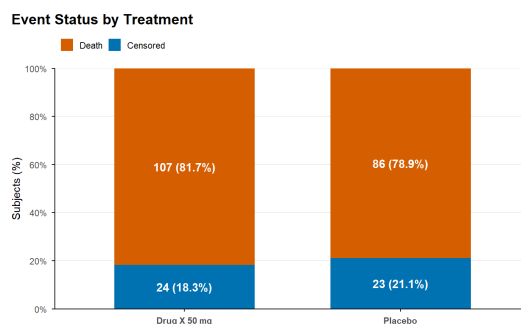
Event Status by Treatment

The first figure summarises event status by treatment arm using a stacked percentage bar chart. Subjects are grouped by treatment, and each bar is partitioned into deaths and censored observations.

Percentages are calculated within each treatment arm, ensuring that the relative contribution of events and censoring can be compared directly across treatments. Each segment is annotated with both the absolute number of subjects and the corresponding percentage, allowing the reader to interpret the figure without referring to external tables.

This visualisation is produced in R using **ggplot2**, while the underlying event data and treatment assignments are prepared in the SAS Language. The result illustrates how Altair SLC preserves the clarity and reproducibility of the SAS Language while leveraging R for expressive graphical output.

Figure 1. Event Status by Treatment - stacked percentage bar chart (R, ggplot2).

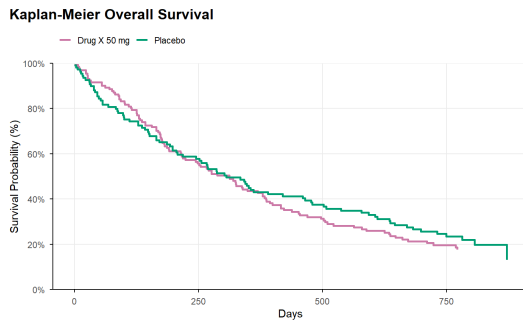


Kaplan-Meier Overall Survival

The second figure presents Kaplan-Meier overall survival curves by treatment. Time-to-event data are derived in the SAS Language, following an ADaM-like structure, and exported to R for survival modelling and visualisation.

Survival probability is displayed on a percentage scale, and treatment groups are distinguished by colour. The step functions reflect the underlying event times exactly as estimated by the Kaplan-Meier method.

Figure 2. Kaplan-Meier Overall Survival curves by treatment arm (R, survival, ggplot2, scales).



By separating data derivation and visualisation responsibilities between the SAS Language and R, the program maintains analytical transparency, while achieving a clean, publication-ready presentation.

Conclusion

This example highlights the elegance of combining the SAS language, as executed in Altair SLC, with R for statistical graphics. The SAS language provides a robust and well-structured foundation for clinical data handling, while R contributes flexible and high-quality visualisations.

Together, they form a coherent workflow that supports both regulatory-grade analysis and effective visual communication, without compromising either.

SLC SAS Language + R Program Code

The following listing contains the complete SAS Language program executed in Altair SLC to generate the analyses and figures presented in this article. The code is included verbatim to ensure full transparency and reproducibility.

```
/*=====
| Altair SLC 2026 (Windows 11)
| SAS + R:4.4.3 via PROC R
| R: survival:3.8.6
| ggplot2:4.0.2
| scales:1.4.0
|
| REPRODUCIBILITY - fixed seeds:
| ADL CALL STREAMINIT(20260207)
| ADTE CALL STREAMINIT(20260208)
| R set.seed(20260207)
| Do NOT change. Graphs are tied to
| these exact values.
|
| Output:
| - C:/test/r_bar_events.png
| - C:/test/r_km_os.png
|=====*/
OPTIONS NODATE NONUMBER
MPRINT MLOGIC SYMBOLGEN;

/*-----
| USER-DEFINED OUTPUT DIRECTORY
|-----*/
%LET outdir = C:/test;
/* Ensure directory exists */
OPTIONS DLCREATEDIR;
LIBNAME _chk "&outdir";
LIBNAME _chk CLEAR;
```

```
/*-----
| R CONFIGURATION
|-----*/
%LET RVER = R-4.4.3;
%LET RHOME = C:\Program Files\R\&RVER.;
OPTIONS SET = R_HOME "&RHOME.";
/* PATH warning >262 chars: harmless */
OPTIONS SET = PATH
"%sysget(PATH); &RHOME.\bin;&RHOME.\bin\x64";

/*-----
| ODS
|-----*/
ODS ALL CLOSE;
ODS ESCAPECHAR = '^';
ODS GRAPHICS ON / RESET = ALL;

/*=====
| 1) ADSL - Subject-Level Analysis
| Dataset (ADaM-like)
|=====*/
DATA work.adsl;
/* FIXED SEED - do NOT change */
/* (article graphs tied to value) */
CALL STREAMINIT(20260207);
LENGTH
STUDYID $20 USUBJID $40
SITEID $10 ARM $40
ARMCD $8 TRT01A $40
TRT01ACD $8 SEX $1
RACE $40;
DO SUBJIDN = 1 TO 240;
STUDYID = "ABC123";
SITEID = CATS("S",
PUT(CEIL(RAND("uniform") * 12), Z3.));
USUBJID = CATS(STUDYID, "-",
SITEID, "-", PUT(SUBJIDN, Z5.));
IF RAND("uniform") < 0.5 THEN DO;
ARM = "Placebo"; ARMCD = "PBO";
TRT01A = "Placebo";
TRT01ACD = "PBO";
END;
ELSE DO;
ARM = "Drug X 50 mg";
ARMCD = "DRUGX";
TRT01A = "Drug X 50 mg";
TRT01ACD = "DRUGX";
END;
AGE = FLOOR(18 + RAND("uniform") * 70);
SEX = IFC(RAND("uniform") < 0.52, "M", "F");
p = RAND("uniform");
IF p < 0.60 THEN RACE = "WHITE";
ELSE IF p < 0.75 THEN RACE = "ASIAN";
ELSE IF p < 0.88 THEN
RACE = "BLACK OR AFRICAN AMERICAN";
ELSE IF p < 0.95 THEN
RACE = "AMERICAN INDIAN OR ALASKA NATIVE";
ELSE RACE = "OTHER";
OUTPUT;
END;
DROP p SUBJIDN;
RUN;

/*=====
| 2) ADTE - Overall Survival
| (Time-to-Event)
|=====*/
DATA work.adte;
SET work.adsl;
/* FIXED SEED - independent of ADSL */
/* stream. Ensures AVAL/CNSR stable */
/* on every run. Do NOT change. */
IF _N_ = 1 THEN CALL STREAMINIT(20260208);
LENGTH PARAMCD $8 PARAM $40
EVNTDESC $40;
FORMAT RANDDT ADT date9.;
PARAMCD = "OS";
PARAM = "Overall Survival";
```

```

RANDDT = '01JAN2024'd
+ FLOOR(RAND("uniform") * 120);
IF TRT01ACD = "PBO" THEN lambda = 1 / 420;
ELSE lambda = 1 / 520;
true_t = RAND("exponential") / lambda;
censor_t = 700 + FLOOR(RAND("uniform") * 200);
AVAL = FLOOR(MIN(true_t, censor_t));
CNSR = (true_t > censor_t);
EVNTDESC = IFC(CNSR = 1, "Censored", "Death");
ADT = RANDDT + AVAL;
KEEP STUDYID USUBJID SITEID
      ARM ARMCD TRT01A TRT01ACD
      AGE SEX RACE
      PARAMCD PARAM
      RANDDT ADT
      AVAL CNSR EVNTDESC;

RUN;

/*=====
| 3) R ANALYSIS + GRAPHS (PROC R) |
*=====*/
DATA work._paths;
  LENGTH outdir png_bar png_km $300;
  outdir = "&outdir.";
  png_bar = CATS("&outdir.", "/r_bar_events.png");
  png_km = CATS("&outdir.", "/r_km_os.png");
RUN;

PROC R;
  EXPORT DATA = work.adtte R = adtte;
  EXPORT DATA = work._paths R = paths;
  SUBMIT;

  suppressPackageStartupMessages(library(survival))
  suppressPackageStartupMessages(library(ggplot2))
  suppressPackageStartupMessages(library(scales))

  outdir <- paths$outdir[1]
  png_bar <- paths$png_bar[1]
  png_km <- paths$png_km[1]
  setwd(outdir)

  # FIXED SEED - R-side reproducibility.
  # Do NOT change.
  set.seed(20260207)

  adtte$TRT01A <- factor(adtte$TRT01A)
  adtte$event <-
    ifelse(adtte$CNSR==1, 0, 1)

  # ---- QUIET WRAPPER ----
  tmp_out <- tempfile()
  tmp_msg <- tempfile()
  con_out <- file(tmp_out, open="wt")
  con_msg <- file(tmp_msg, open="wt")
  on.exit({
    try(sink(type="message"), silent=TRUE)
    try(sink(), silent=TRUE)
    try(close(con_msg), silent=TRUE)
    try(close(con_out), silent=TRUE)
    try(unlink(c(tmp_out,tmp_msg)), silent=TRUE)
  }, add=TRUE)
  sink(con_out)
  sink(con_msg, type="message")

```

```

# --- Bar chart ---
# --- % by Treatment, n (xx.x%) ---
ec <- aggregate(
  list(n=adtte$USUBJID),
  list(TRT01A=adtte$TRT01A, Event=adtte$EVNTDESC),
  length)
totals <- aggregate(n~TRT01A, ec, sum)
ec <- merge(ec, totals, by="TRT01A",
  suffixes=c("", "_tot"))
ec$pct <- ec$n / ec$n_tot
ec$Event <- factor(ec$Event,
  levels=c("Death", "Censored"))
ec$label <- paste0(ec$n, " (",
  percent(ec$pct,accuracy=0.1), "%)")

p1 <- ggplot(ec, aes(x=TRT01A, y=pct,fill=Event)) +
  geom_col(width=0.65) +
  geom_text(
    aes(label=label),
    position=position_stack(vjust=0.5),
    color="white", fontface="bold", size=5) +
  scale_fill_manual(values=c(
    "Death"="#D55E00",
    "Censored"="#0072B2")) +
  scale_y_continuous(
    limits=c(0,1),
    breaks=seq(0,1,0.2),
    labels=percent_format(accuracy=1),
    expand=c(0,0)) +
  labs(title="Event Status by Treatment",
    x=NULL, y="Subjects (%)", fill=NULL) +
  theme_minimal(base_size=14) +
  theme(
    plot.title.position="plot",
    plot.title=element_text(face="bold", size=18),
    axis.text.x=element_text(face="bold"),
    axis.line=element_line(color="black"),
    axis.ticks=element_line(color="black"),
    panel.grid.major.x=element_blank(),
    panel.grid.minor=element_blank(),
    legend.position="top",
    legend.justification="left")

invisible(png(png_bar,
  width=1400,height=850,res=150))
print(p1)
invisible(dev.off())

# --- Kaplan-Meier ---
# --- coloured + % y-axis ---
fit <- survfit(Surv(AVAL,event)~TRT01A, data=adtte)
ss <- summary(fit)
km <- data.frame(
  time=ss$time,
  surv=ss$surv,
  strata=ss$strata)
km$trt <- sub("^TRT01A=", "", km$strata)

```

```

p2 <- ggplot(km, aes(x=time,y=surv,color=trt)) +
geom_step(linewidth=1.2) +
labs(
  title="Kaplan-Meier Overall Survival",
  x="Days",
  y="Survival Probability (%)",
  color=NULL) +
scale_color_manual(values=c(
  "Placebo"="#009E73",
  "Drug X 50 mg"="#CC79A7")) +
scale_y_continuous(
  limits=c(0,1),
  breaks=seq(0,1,0.2),
  labels=percent_format(accuracy=1),
  expand=c(0,0)) +
theme_minimal(base_size=14) +
theme(
  plot.title.position="plot",
  plot.title=element_text(face="bold",size=18),
  axis.line=element_line(color="black"),
  axis.ticks=element_line(color="black"),
  panel.grid.minor=element_blank(),
  legend.position="top",
  legend.justification="left")

invisible(png(png_kn,
  width=1400,height=850,res=150))
print(p2)
invisible(dev.off())

# ---- end quiet wrapper ----
sink(type="message")
sink()
close(con_msg)
close(con_out)
unlink(c(tmp_out,tmp_msg))

ENDSUBMIT;
QUIT;

ODS GRAPHICS OFF;
%PUT NOTE: R bar chart -> &outdir./r_bar_events.png;
%PUT NOTE: R KM plot -> &outdir./r_km_os.png;

```

Pablo Orosco Joerger

Ask The Consultant

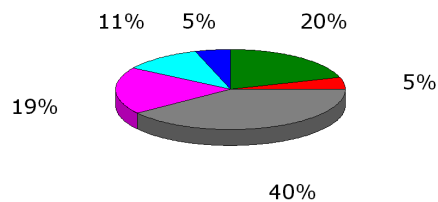
This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

What Is Better Than the ALWAYS Distorting 3D Pie Chart?

Q: What Is Better Than the ALWAYS Distorting 3D Pie Chart?

A: [SAS 9.4] Let me show you again how a 3D pie chart distorts. I first showed this 21 years ago, in Issue 29, 1st Quarter 2005.

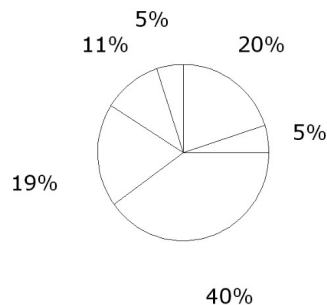
Figure 1. 3D Pie Chart
Distorts the Apparent Relative Size of Slices



11% slice is almost as wide as 19% slice
5% slice at rear is twice as big as 5% slice at the right

The only 3D pie chart like this which cannot distort is one for TWO slices of EQUAL size. As soon as I discovered this phenomenon, I crafted the obvious simple solution below.

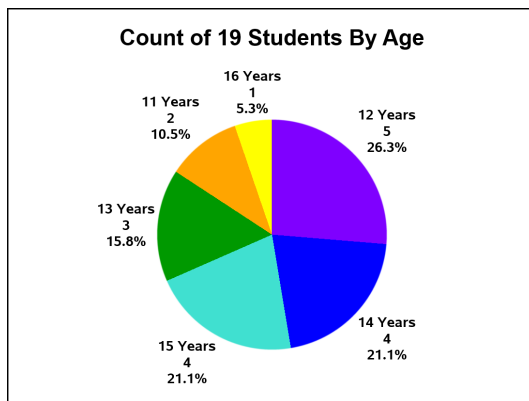
Figure 2. 2D Pie Chart
Always Presents Accurate Relative Size of Slices



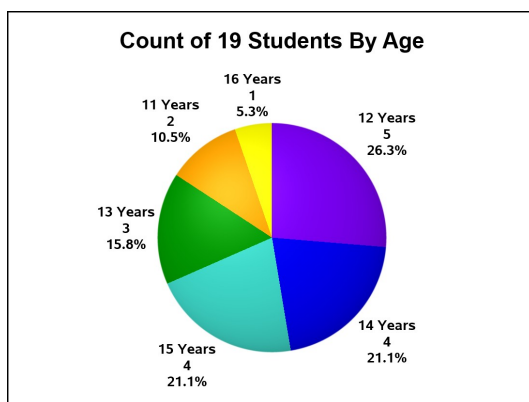
Both of these figures were created with SAS/GRAPH. You can find the code in Issue 29. Most SAS users and I have moved on to SAS ODS Graphics, which is built into Base SAS software. ODS Graphics has no PIE3D statement analogous to that which is in SAS/GRAPH.

With ODS Graphics PROC SGPIE and its PIE statement, it is impossible to create a pie chart with empty slices. I could paint all of the slices White, but there is no analogue to the SAS/GRAPH COUTLINE option to paint the outlines black. Such an "empty" pie chart disappears. One might try to create an empty pie chart outside of SAS/GRAPH, using Graph Template Language, which is outside the scope of this article. Also, I cannot assure you that GTL can be used to reproduce the layout below, differing only by having empty slices.

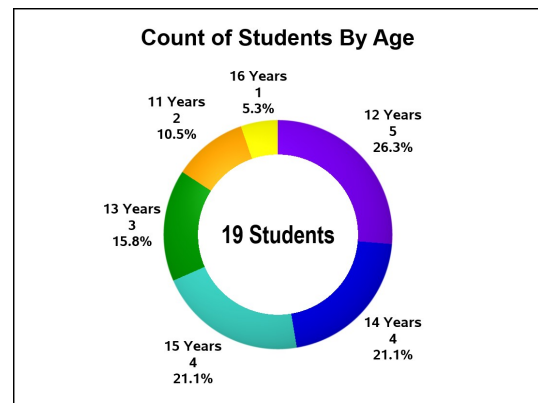
Below is a coloured 2D pie chart using ODS Graphics. The colour palette used below omits Red. The inability to distinguish Red and Green is the commonest form of colour blindness, as I have been reminding SAS users since 1995.



Below is an attempt create a Pseudo3D effect, using the PROC SGPIE PIE statement option DATASKIN=PRESSED. There are four other choices for DATASKIN. I recommend this one. It is the least bad option if one wishes to resort to this act of desperation for the sake of Pseudo3D. I say 'Pseudo3D', rather than "3D" or "like 3D" because many viewers might not regard this as like 3D.



Before I get to code, I wanted to first show you a Pseudo3D Donut Chart (aka Doughnut Chart). I had high hopes that DATASKIN=PRESSED on the DONUT statement would give me an image that REALLY looked like a doughnut. The problem is that the bulging up starts at the outer perimeter. Lack of bulging up at the inner perimeter makes this simply look like the PRESSED pie chart above, but with a hole and a label in the middle. The available space in the middle here is used to relocate the Total Count of Students from the chart title to the Hole Label. See the remarks about Hole Label options in the code. The Hole Label text can be non-Bold, but that still matches neither the text appearance of the title, nor that of the slice labels.



Here is code to create the three ODS Graphics images above.

```
DATA work.ToChart(KEEP=AgeInYears);
  LENGTH AgeInYears $8;
  SET sashelp.class(keep=Age) nobs=HowMany;
  IF _N_ EQ 1 THEN
    CALL SYMPUT('TotalCount', TRIM(LEFT(HowMany)));
  AgeInYears = TRIM(LEFT(Age)) || ' Years';
RUN;

ODS RESULTS OFF;
ODS _ALL_ CLOSE;
ODS LISTING GPATH="C:\temp" DPI=300;
ODS GRAPHICS / RESET=ALL NOSCALE WIDTH=3.65in
  IMAGENAME='PieChartWithOutsideLabels';
TITLE1 FONT='Arial/Bold' HEIGHT=11pt
  "Count of &TotalCount Students By Age";
TITLE2 HEIGHT=4pt COLOR=white 'White Space';
/* my preference */

PROC SGPIE DATA=work.ToChart;
  STYLEATTRS DATACOLORS=(
    CX7F00FF Blue Turquoise CX009900 Orange Yellow);
  /* Omit this STYLEATTRS statement
  to see the SAS defaults. */
  /* SAS Violet is too Pink */
  /* SAS Blue looks like CX0000FF */
  /* SAS Turquoise looks like CX40E0D0 */
  /* SAS Green looks like CX008000 */
  /* SAS Green is too dark. */
  /* Bessler Green is CX009900 */
  /* 100% RGB Green, CX00FF00 is too light */
  /* SAS Yellow looks like CXFFFF00 */
  /* SAS Orange looks like CXFFA500 */
  PIE AgeInYears /* the CATEGORY variable */
  /
  /* omitting to use RESPONSE= forces use
  of frequency of each CATEGORY value */
  OTHERPERCENT=0
  SLICEORDER=RESPDESC
  /* largest to smallest */
  DIRECTION=CLOCKWISE
  /* COUNTERCLOCKWISE is default */
  STARTANGLE=90 /* Twelve O'Clock,
  Three O'Clock is default */
  STARTPOS=EDGE /* Default is CENTER */
  DATALABELATTRS=(FAMILY=Arial WEIGHT=Bold SIZE=7pt)
  /* SIZE adjusted so that the label
  for smallest slice would fit within
  extended boundaries of the slice */
  DATALABELDISPLAY=(CATEGORY RESPONSE PERCENT)
  /* This is always the order
  top to bottom */
  /* But default omits PERCENT */
  DATALABELLOC=OUTSIDE;
RUN;

ODS GRAPHICS / RESET=INDEX
  /* retain NOSCALE WIDTH=3.65in */
  IMAGENAME='PieChart_Pseudo3D';
/* TITLE1 & TITLE2 inherited here */
```

```

PROC SGPIE DATA=work.ToChart;
  STYLEATTRS DATACOLORS=(
    CX7F00FF Blue Turquoise CX009900 Orange Yellow);
  PIE AgeInYears /
    DATASKIN=PRESSED /* for Pseudo3D effect */
    /* other options are NONE (the default),
      CRISP, GLOSS, MATTE, SHEEN */
    OTHERPERCENT=0
    SLICEORDER=RESPDESC
    DIRECTION=CLOCKWISE
    STARTANGLE=90
    STARTPOS=EDGE
    DATALABELATTRS=(FAMILY=Arial WEIGHT=Bold SIZE=7pt)
    DATALABELDISPLAY=(CATEGORY RESPONSE PERCENT)
    DATALABELLOC=OUTSIDE;
RUN;

ODS GRAPHICS / RESET=INDEX
  IMAGE=DonutChart_Pseudo3D';
  TITLE1 FONT=Arial/Bold HEIGHT=11pt
    "Count of Students By Age";
  /* new TITLE1 */
  TITLE2 HEIGHT=4pt COLOR=white 'White Space';

PROC SGPIE DATA=work.ToChart;
  STYLEATTRS DATACOLORS=(
    CX7F00FF Blue Turquoise CX009900 Orange Yellow);
  DONUT AgeInYears /
    /* HOLEVALUE=&TotalCount."
      HOLELABEL="Students" fails.
      HOLEVALUE must be hard-coded number */
    HOLELABEL=&TotalCount Students"
    HOLELABELATTRS=FAMILY=Arial Narrow WEIGHT=Bold)
    /* FAMILY=Arial WEIGHT=Bold
      Would cause the label to be omitted,
      without any WARNING in the SAS Log
      COLOR= supported, but SIZE= is NOT. */
    DATASKIN=PRESSED
    OTHERPERCENT=0
    SLICEORDER=RESPDESC
    DIRECTION=CLOCKWISE
    STARTANGLE=90
    STARTPOS=EDGE
    DATALABELATTRS=(FAMILY=Arial WEIGHT=Bold SIZE=7pt)
    DATALABELDISPLAY=CATEGORY RESPONSE PERCENT)
    DATALABELLOC=OUTSIDE;
RUN;

ODS LISTING CLOSE;

```

For other ways to create pie charts (excluding Pseudo3D with the DATASKIN option), using CallOut Labels or a Legend, or how to cope with readability of Inside-The-Slices labels, see Chapter 5 in “Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization”. More details can be found at link.springer.com/book/10.1007/978-1-4842-8609-8.

LeRoy Bessler, Bessler Consulting and Research

Formats, Options, and Functions

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

[SAS 9.3 and Altair SLC] The numeric format **BESTXw** is like the **Fw.** format, but with a different default width. It is used with **OPTIONS NOFMterr** to replace unknown numeric user formats.

[Altair SLC 2025] **NOWARNINGNOTESSETSYSERR** is the default system option, where some **WARNING** and **ERROR** messages do not alter the value of **&syscc** and **&syserr** macro variables. The **WARNINGNOTESSETSYSERR** system option can be used in these situations to alter the **&syserr** macro variable value.

[SAS 9.4 and Altair SLC] The **DOSUBL()** function enables you to run lines of code that form one or more executable SAS language code steps. The **DOSUB()** function enables you to run a SAS language program stored in an external file.

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 41,300 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 23,400 members. Everyone is welcome to request to join, but discussions here are also restricted to SAS-related topics: www.linkedin.com/groups?gid=805217.
- Altair SLC Personal Edition provides named users with free access to Altair SLC and Analytics Workbench 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. When the Personal Edition software is downloaded, it includes a licence for up to 12 months. At the end of this term, users should download a new copy of the software to renew their licence and/or to upgrade to the latest available full version. Download the software from altairone.com/Marketplace. No technical support service is offered with the Personal Edition, but assistance can be requested using the Community forums at community.altair.com. The current General Access version of Altair SLC is 5.26.0 (2026).

- 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 local SAS installations, SAS Viya, SAS 9 EBI, SASjs Server or local Altair SLC! Just set up the target, and click the running man to execute your SAS code. The SASjs Server could be your local SAS or Altair SLC 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.
 - The SAS Institute has also been developing its own extension called **SAS**. You can develop SAS applications and run code on local SAS installations, SAS Viya, or SAS OnDemand for Academics! Just set up the target, and click the running man to execute your SAS code. The extension can be obtained from the Visual Studio Code Marketplace, where the current version is v1.19.0.
 - Another Microsoft Visual Studio Code Editor extension has just been released called **Altair SLC**. This extension currently only allows you to connect to local 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 v0.7.1 (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.

May 2026

31 May-3 June **PharmaSUG 2026**, Boston, MA, USA

2.5 days of live presentations, hands-on training, e-posters and panel discussions. Pre- and post-conference in-person training seminars. Networking opportunities during the conference. Full details at www.pharmasug.org.

November 2026

15-18 **PhUSE EU Connect**, Glasgow, UK

The PhUSE Community allows you to enhance your learning and gain insights from our experts on today's most critical industry topics through a variety of engaging presentations, workshops, and networking opportunities. Individuals from all areas of the industry will be gathered in one place, providing an opportunity not only for you to advance your career but also to heighten the pharmaceutical industry's influence and findings. Full details at www.phuse.eu.

Contacts

LeRoy Bessler Email: besslerleroy2024@gmail.com

Pablo Orosco Joerger Email: pablo.joerger@icloud.com

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 Philip R Holland.