

Animate Your Safety Data

Kriss Harris, SAS Specialists Limited, Hertfordshire, United Kingdom

ABSTRACT

When reporting your safety data do you ever feel sorry for the person who has to read all of the laboratory listings and/or summaries? Or do you ever wonder that there might be a better way to visualize the safety data. Let's help make the reviewer's life easier and also understand the safety data better with the use of animation.

This paper shows how you can use animation in SAS 9.4 to report your safety data. Such as visualizing a patient's laboratory results, vital sign results and electrocardiogram results and seeing how those safety results change over time. In addition you will learn how to make animate adverse events over time, and how to show the relationships between adverse events and laboratory results using animation. Animating your data will bring your data to life and help to better more lives!

INTRODUCTION

Animating your safety data over time and/or for each subject can help you to understand your safety data more easily and put things into context. Such as, you can quickly get an idea of the number of subjects that had abnormal laboratory results. You are also able to easily determine the adverse events each subject had and the duration of the adverse event. Using SAS 9.4 you can create animations with the new animation OPTIONS in conjunction with the PRINTER destination.

ANIMATION AND GTL

You can animate figures that you have produced using GTL or the SG procedures: SGPLOT, SGPANEL and SGSCATTER by encapsulating those procedures within the animation OPTIONS and the ODS PRINTER statements. Please see the example below. The code highlighted in yellow is an example of how to use the animation OPTIONS and the ODS PRINTER statements.

```
options printerpath=gif animation=start animduration=0.5 animloop=yes  
noanimoverlay;  
ods printer file="C:\animation.gif";
```

```
proc template;  
  define statgraph <template-name>;  
    beginngraph / <options>;  
      <GTL statements>;  
    endngraph;  
  end;  
run;
```

```
proc sgrender data=<data-set-name> template=<template-name>;  
  <other optional statements>;  
run;
```

```
options printerpath=gif animation=stop;  
ods printer close;
```

To animate figures, you need to have more than one image, and so you will need to have the right data, and also use the BY statement in your SGRENDER procedure, or alternatively use multiple SGRENDER statements.

ANIMATING LABORATORY RESULTS

Typically when assessing safety within laboratory results a summary table of the abnormal results for each parameter is produced, and a listing of every observation for each parameter is produced. These tables and listings are very large – especially the listings, however the outputs are currently required by the regulatory authorities. Occasionally a box plot is produced of the laboratory parameters, which show the distribution of the measurements by treatment and if any of the measurements are above or below the standard normal limits. These box plots tend to be done for each visit, and therefore as well as producing the static graphs it will be useful to easily see if and how the distribution of the measurements changes over time, and this can be done with animation.

The code below shows how to animate your laboratory results, and to view the animation click on the link below:

<http://www.krissharris.co.uk/pharmasugchina/2016/83/output/boxplotM1.gif>

The dataset used in this paper is from the CDISC SDTM / ADAM Pilot Project and this was obtained from the CDISC website (CDISC, 2013). The dataset has been further filtered down to only contain the Creatinine laboratory parameter and only Male subjects. The reason why there are currently only Male subjects is because the normal limits for the Creatinine parameter vary by gender, and so for this example, only using one gender allows us to determine when the values are outside the standard normal limits. The counts of the total number of subjects with observations for each treatment and visit have been added on to the dataset too. This enables us to put the number of subjects on each plot.

When producing animations you are going to want to know what each image is. The DYNAMIC statement along with the ENTRYTITLE statement in GTL can help you to place identifiers on the images, in particular when you are using the BY variable in SGRENDER. In the example below the DYNAMIC, ENTRYTITLE and BY statement allows us to know the laboratory parameter, gender and visit that is being plotted. Figure 1, is an example of the Boxplots that are produced at one of the visits.

```
* Creating template;
proc template;
  define statgraph boxplot_template_sex;
    dynamic _byval_ _byval2_ _byval4_ upperlim lowerlim;
    begingraph;
      entrytitle halign = center _byval_ ":" " Sex = " _byval2_ " and
        Visit = " _byval4_ ;
      layout overlay / yaxisopts=(linearopts=(viewmin=60 viewmax=180
        tickvaluesequence=(start=60 end=180 increment=30)));
      boxplot x = trtan y = aval / group = trtan groupdisplay = cluster;
      referenceline y=lowerlim / lineattrs=(pattern=2);
      referenceline y=upperlim / lineattrs=(pattern=2);
      innermargin;
        axistable x = trtan value = nobs_subject_visit / stat=mean
          label= "n" valueattrs=(size=9);
      endinnermargin;
    endlayout;
  endgraph;
end;
run;
* Counting distinct subjects numbers in each treatment group at each
timepoint;

* ADLBC - Source dataset;
data adlbc_all;
  set source.adlbc;
  where param in ("Creatinine (umol/L)") and ablf1 ne "Y";
  avisit=strip(avisit);
run;
```

```

* Adding on n numbers;
proc sql;
  create table adlbc_all_n_gen as
  select *, count(distinct usubjid) as nobs_subject_visit
  from adlbc_all
  group by trtan, avisitn, sex;
quit;

proc sort data = adlbc_all_n_gen;
  by param sex avisitn avisit trtan;
run;

* Producing animation;
options nobyline;

options papersize=('8 in', '4.8 in') printerpath=gif animation=start
animduration=0.5 animloop=yes noanimoverlay;
ods printer file="&outpath\boxplotM1.gif";

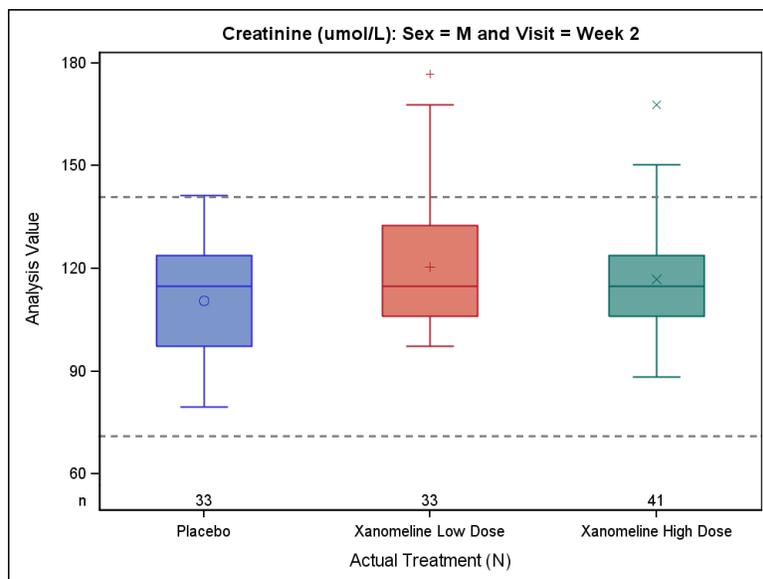
ods graphics / width=8in height=4.8in imagefmt=gif;

proc sgrender data = adlbc_all_n_gen template = boxplot_template_sex;
  where avisitn ne . and sex = "M";
  by param sex avisitn avisit;
  dynamic upperlim = "alhi" lowerlim = "allo";
  format trtan trtfmt.;
run;

options printerpath=gif animation=stop;
ods printer close;

```

Figure 1: Boxplot of Creatinine by Treatment for Male subject at Visit 2



SCATTER PLOT OF LABORATORY RESULTS

Scatter plots can be useful at explaining laboratory results, especially when the number of observations in each treatment group is quite small and/or you want to see the amount of results that fall outside the standard normal limits. The code below was used to produce an animation of the scatterplots which showed the Creatinine measurements by the treatment and also showed the number of subjects that had results outside the standard normal limits. Figure 2, is an example of the scatterplots that are produced at one of the visits. To see the animation, click on the link below.

<http://www.krissharris.co.uk/pharmasugchina/2016/83/output/scatterplotM1.gif>

```
* For Males;
```

```
proc template;
```

```
  define statgraph scatterplot_template_sex;
    dynamic _byval_ _byval2_ _byval4_ upperlim lowerlim;
    begingraph;
      entrytitle halign = center _byval_ ":" " Sex = " _byval2_ " and
        Visit = " _byval4_;
      layout overlay / yaxisopts=(linearopts=(viewmin=60 viewmax=180
        tickvaluesequence=(start=60 end=180 increment=30)));
      scatterplot x = trtan y = aval / group = trtan groupdisplay =
        cluster jitter=auto jitteropts=(axis=x);
      referenceline y=lowerlim / lineattrs=(pattern=2);
      referenceline y=upperlim / lineattrs=(pattern=2);
      innermargin;
      axistable x = trtan value = nobs_abn_subject_visit / stat=mean
        label= "n" valueattrs=(size=9);
      endinnermargin;
    endlayout;
  endgraph;
end;
```

```
run;
```

```
* Counting distinct subjects numbers with abnormal observations in each
treatment group at each timepoint;
```

```
proc sql;
```

```
  create table abnormal_n as
  select distinct param, trtan, avisitn, avisit, sex, count(distinct usubjid)
  as nobs_abn_subject_visit
  from adlbc_all (where=(LBNRIND ne "NORMAL"))
  group by param, trtan, avisitn, sex;
```

```
quit;
```

```
* Merging abnormal n numbers onto adlbc_all dataset, for plotting;
```

```
proc sql;
```

```
  create table adlbc_all_abnormal_n_gen as
  select a.*, b.nobs_abn_subject_visit
  from adlbc_all as a left join abnormal_n as b
  on a.param = b.param and a.sex = b.sex and a.avisitn = b.avisitn and
  a.avisit = b.avisit and a.trtan = b.trtan;
```

```
quit;
```

```
proc sort data = adlbc_all_abnormal_n_gen;
```

```
  by param sex avisitn avisit trtan;
```

```

run;
options nobyline;
goptions reset=all;
options papersize=('8 in', '4.8 in') printerpath=gif animation=start
animduration=0.5 animloop=yes noanimoverlay;
ods printer file="&outpath\scatterplotM1.gif";

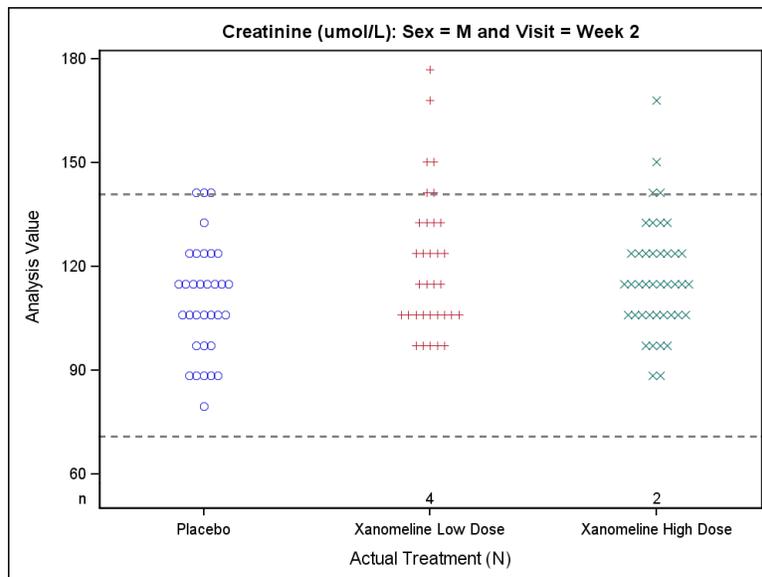
ods graphics / width=8in height=4.8in imagefmt=gif;

proc sgrender data = adlbc_all_abnormal_n_gen template =
scatterplot_template_sex;
  where avisitn ne . and sex = "M";
  by param sex avisitn avisit;
  dynamic upperlim = "alhi" lowerlim = "allo";
  format trtan trtfmt.;
run;

options printerpath=gif animation=stop;
ods printer close;

```

Figure 2: Scatterplot of Creatinine by Treatment for Male subject at Visit 2



ANIMATING ADVERSE EVENTS

In clinical trials Adverse Events (AEs) are typically assessed in the following ways:

- examining the incidence of the number of subjects with AE summarised by treatment group;
- examining the time to first occurrence of AE summarised by treatment group;
- examining the duration of AE summarised by treatment group;

Usually the reporting of AEs is presented descriptively, such as when the incidence rate of the number of subjects with AE is examined. The common way of reporting AE's descriptively within a table is predominantly due to the ICH guideline "Structure and Content of Clinical Study Reports E3" (ICH, 1995), which is adhered to by organisations conducting clinical trials. The ICH recommends in section 12.2 of

the guidelines that, for the evaluation of adverse events, a detailed record of adverse events be made available to regulatory authorities such as the FDA when requested. The ICH also recommends that all adverse events occurring after initiation of study treatments should be displayed in summary tables.

Kaplan-Meier figures of time to first occurrence of adverse events is becoming increasingly popular when reporting adverse, however the most popular adverse event figure is the AE timeline. The benefits of the AE timeline plots are that it allows you to see the events that each patient has and any multiple occurrences. It shows when the events started and stopped and the duration of the events, and also shows the severity of the event. The drawbacks of the AE timeline plots are that often the stop date of the event is missing and therefore imputation methods need to be decided on the best way to handle the missing end dates, and another drawback is that the AE timeline plots needs to be done for every subject, however this is where animating the AE's is a benefit.

The code below shows how to animate AE data. Figure 3, is an example of the AE timeline plot which is produced for one patient. The animation is provided in the link below:

<http://www.krissharris.co.uk/pharmasugchina/2016/83/output/AE.gif>

```
* ADAE;
data adae_trtemfl;
  set source.adae;
  where TRTEMFL = "Y";
run;

proc sort data = adae_trtemfl;
  by usubjid;
run;

* Template for AE;
proc template;
  define statgraph ae_timeline;
    dynamic _byval_;
    beginingraph;
      DiscreteAttrMap name="Severity";
      Value "MILD" /fillattrs=(color= GraphData3:color)
        lineattrs=(color=GraphData3:contrastcolor pattern=solid)
        markerattrs=(color= GraphData3:color);
      Value "MODERATE" /fillattrs=(color= GraphData1:color)
        lineattrs=(color=GraphData1:contrastcolor pattern=solid)
        markerattrs=(color= GraphData1:color);
      Value "SEVERE" /fillattrs=(color= GraphData2:color)
        lineattrs=(color=GraphData2:contrastcolor pattern=solid)
        markerattrs=(color= GraphData2:color);
    EndDiscreteAttrMap;
    DiscreteAttrVar attrvar=id_severity var=aesev attrmap="Severity";

    entrytitle halign = center "Unique Subject ID = " _byval_;
    layout overlay / xaxisopts=(label="Relative Start Day");
      highlowplot y=aedecod low=astdy high=aendy / type = bar barwidth =
        0.05 highcap=none lowcap = none group = id_severity;

      scatterplot y =aedecod x=astdy /
        markerattrs=(symbol=trianglerightfilled size =15)
        markeroutlineattrs=(color=black) group = id_severity;
      scatterplot y =aendy x=aendy /
        markerattrs=(symbol=triangleleftfilled size =15)
```

```

markeroutlineattrs=(color=black) group = id_severity;

discretelegend "Severity" / type = markercolor location = outside
  valign = bottom across = 3 down = 1;
endlayout;
endgraph;
end;
run;

options nobyline;
option reset=all;

options papersize=('8 in', '4.8 in') printerpath=gif animation=start
animduration=0.5 animloop=yes noanimoverlay;
ods printer file="&outpath\AE.gif";

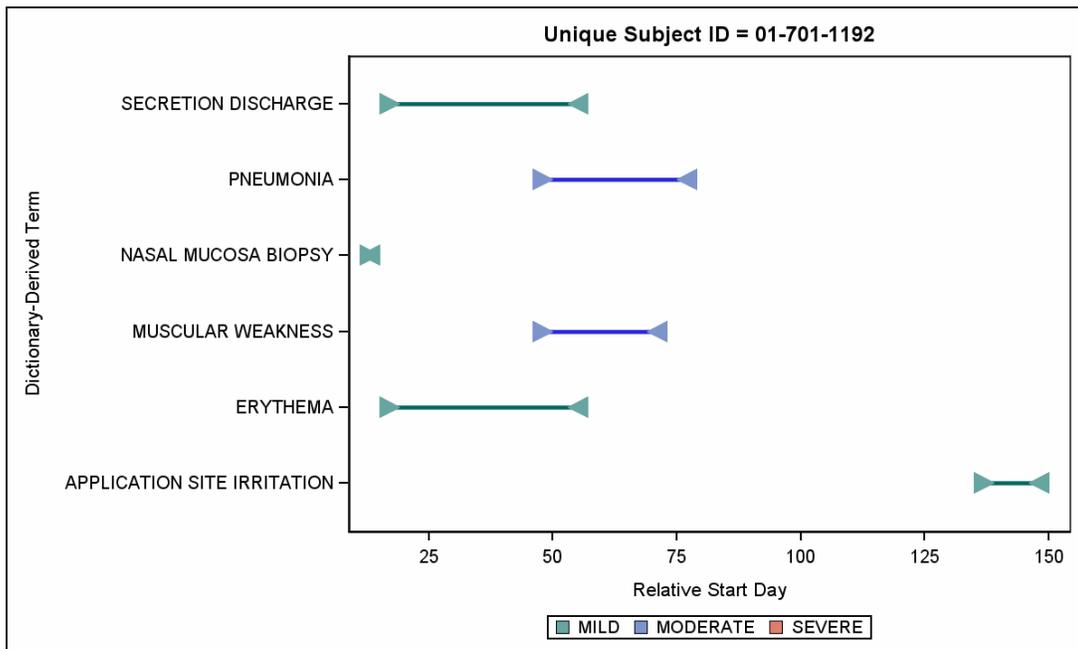
ods graphics / width=8in height=4.8in imagefmt=gif;

title "Treatment Emergent Adverse Events";
*ods graphics / imagefmt=png;
proc sgrender data = adae_trtemfl template=ae_timeline;
  where astdy ne . and aendy ne .;
  by usubjid;
run;

options printerpath=gif animation=stop;
ods printer close;

```

Figure 3: Adverse Event Timeline for subject 1192



ANIMATION AND RELATIONSHIPS

Animation can be used to visualise relationships. Such as, the relationship between laboratory measurements and adverse events. The animation OPTIONS and the PRINTER statement in conjunction with ODS LAYOUT GRIDDED and ODS REGION can help to produce animations that show multiple plots using multiple datasets. The code below shows you how to show a relationship between laboratory measurements and adverse events, and an example of this can be seen in Figure 4. The animation is provided in the link below:

<http://www.krissharris.co.uk/pharmasugchina/2016/83/output/AE2.gif>

The dataset used in the example below only contain subjects that are both in the laboratory datasets and the adverse events datasets. In the example below, BY usubjid cannot be used as in the examples above, otherwise you will get unusual results, for the below example to work, you need to explicitly select the subject that you want plotted in both regions. A macro has been used to select the subjects, however the macro can be made to be more generalised so the results of all the common subjects can be displayed within the animation.

```
proc template;
  define statgraph scatterplot_template_sex;
    dynamic _byval_ _byval2_ _byval4_ upperlim lowerlim;
    begingraph;
      entrytitle halign = center "Unique Subject ID = " _byval_;
      layout overlay / yaxisopts=(linearopts=(viewmin=60 viewmax=180
        tickvaluesequence=(start=60 end=180 increment=30)));
      scatterplot x = ADY y = aval / group = trtan groupdisplay = cluster
        jitter=auto jitteropts=(axis=x);
      referenceline y=lowerlim / lineattrs=(pattern=2);
      referenceline y=upperlim / lineattrs=(pattern=2);
    endlayout;
  endgraph;
end;
run;

* Macrolised;

options nobyline;

options papersize=('8 in', '4.8 in') printerpath=gif animation=start
animduration=0.5 animloop=yes noanimoverlay;
ods printer file="&outpath\AE2.gif";

ods graphics / width=8in height=4.2in imagefmt=gif ;

%macro subject(subject=);

ods layout gridded rows=2 column_gutter=1 advance=bygroup;

ods region width=4.2in height=2in;
title1 "Treatment Emergent Adverse Events";
proc sgrender data = adae_trtemf1 template=ae_timeline;
  where astdy ne . and aendy ne . and sex = "M" and usubjid in ("&subject.");
  by usubjid;
run;

ods region width=4.2in height=2in;
```

```

proc sgrender data = adlbc_all_abnormal_n_gen_ae template =
scatterplot_template_sex;
  where avisitn ne . and sex = "M" and usubjid in ("&subject.");
  by usubjid;
  dynamic upperlim = "alhi" lowerlim = "allo";
  format trtan trtfmt.;
run;

ods layout end;

%mend;

%subject(subject=%str(01-701-1023));
%subject(subject=%str(01-701-1097));

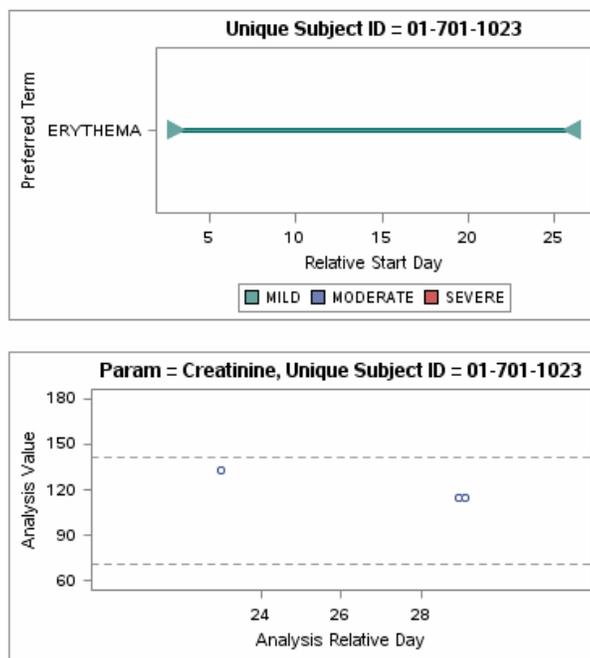
options animation=stop;
ods printer close;

```

Figure 4: Adverse Event and Laboratory Measurements

Treatment Emergent Adverse Events

08:35 Sunday, August 14, 2016 1



You will notice that within the animations above, there is no way to pause the animation and look further into an individual visit or subject. This is because the above outputs have been outputted in the .GIF format. If you want to be able to start, stop or pause your output then you can use the .SVG output, although with .SVG output you do need to give your browser permission to use the controls. To output to .SVG only requires a few modifications to the code. These changes are:

- Change the first `printerpath=gif` to `printerpath=svg`.
- Change `ods printer file="&outpath\AE2.gif"` to `ods printer file="&outpath\AE2.svg"`.
- Change `options printerpath=gif animation=stop` to `options animation=stop`;

CONCLUSIONS

Animations can be done quite easily with SAS 9.4. It does take a while getting used to the PAPERSIZE option and how that interacts with the ODS GRAPHIC sizing options, and also determining how ODS LAYOUT GRIDDED and ODS REGION can work within the animations, however this paper does give you a few examples of the methods to use.

If you want to control your animations more, then .SVG output should be produced instead of .GIF. Please see the example below:

<http://www.krissharris.co.uk/pharmasugchina/2016/83/output/AE2.svg>

REFERENCES

- CDISC. (2013). *CDISC*. Retrieved June 2015, from SDTM/ADaM Pilot Project:
http://www.cdisc.org/system/files/members/article/application/zip/updated_pilot_submission_package.zip
- Matange, S. (n.d.). *Animation using SGPLOT*. Retrieved August 12, 2016, from Graphically Speaking:
<http://blogs.sas.com/content/graphicallyspeaking/2013/05/23/animation-using-sgplot/>

ACKNOWLEDGMENTS

I would like to thank Adrienne Bonwick for reviewing this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kriss Harris
SAS Specialists Limited
italjet125@yahoo.com
<http://www.krissharris.co.uk>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.