

# PharmaSUG 2013 – PO05

## ADaM Datasets for Graphs

Kevin Lee, Cytel, Inc., Chesterbrook, PA

Chris Holland, Amgen, Rockville, MD

### ABSTRACT

The paper is intended for clinical trial SAS® programmers who create graphs using ADaM (Analysis Data Model) datasets. ADaM datasets are analysis-ready, so programmers need to apply CDISC principles and guidelines to create them, while at the same time being mindful of their intent as plotting datasets. The paper will discuss the basic principles of graph creation using ADaM datasets.

We propose the use of PARAM, AVAL, AVISIT and AVISITN for plotting. The paper will show in an example how to use AVAL for the y-axis and AVISITN for the x-axis for simple plotting. We will discuss the role of AVISITN as a numeric representation of AVISIT and propose the use of AVISITN for plotting purposes.

We will also discuss some of the graphs created by features of SAS/STAT, ODS GRAPH and Graph Template Language (GTL). This will eliminate some of the intermediate SAS datasets that SAS programmers need to create for graph creations, which means that the graphs can be created directly from ADaM without the intermediate datasets. Examples using SAS/STAT, ODS GRAPH and GTL will be provided.

### INTRODUCTION

The CDER Common Data Standards Common Issues document (<http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM254113.pdf>) states the following with regard to ADaM data:

*One of the goals of standardizing analysis datasets using ADaM is to simplify the programming steps necessary for performing an analysis. Analysis datasets with repeated measures data should include a numeric time variable (adequately documented in the define file) that measures time from randomization in real time (e.g. Week 8 would be represented as an 8). The reason for this request is related to a common analysis, the portrayal of data over the duration of a study. This data is often shown as means or medians by treatment group by time on study. This time on study is defined by windowed visits which may be represented in ADaM datasets as AVISIT, a character variable. AVISIT is adequate for subsetting the data but this variable is not a measure of time in calendar time so it is not sufficient for plotting data where the x-axis measures real time.*

In addition to this CDER-specific requirement, the FDA Study Data Specifications document (<http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM312964.pdf>) includes the following similar piece of guidance:

*When a dataset contains multiple records per subject, a variable for relative day of measurement or event and variables for visit should be included. In addition to a protocol-scheduled visit variable, include at least two timing variables; a character variable describing the visit (e.g. WEEK 8) and a corresponding numeric variable (e.g. 8). These two variables are measures of time from randomization.*

A logical ADaM variable that can be used to address this common data issue is AVISITN, which is a numeric representation of AVISIT. The ADaM Implementation Guide v1.0 provides the following notes regarding AVISITN:

*A numeric representation of AVISIT. This may be a protocol visit number, a week or cycle number, an analysis visit number, or any other number logically related to AVISIT or useful for sorting that is needed for analysis. Within a parameter, there is a one-to-one mapping between AVISITN and AVISIT so that AVISITN has the same value for each distinct AVISIT. In the event that a record does not fall within any predefined analysis timepoint window, AVISITN can be populated in any way that the sponsor chooses to indicate this fact (e.g., may be null). Values of AVISITN are sponsor-defined.*

As a means of addressing this need of reviewers, statistical programmers, and study statisticians alike, we propose the use of AVISITN to represent a unit of time associated with the planned corresponding visit. In addition to representing a unit of time that can be used for plotting, the values can be used for sorting and have the added benefit of an intuitive meaning without the need for a protocol, a corresponding AVISIT value, or some other documentation that would be needed to interpret the meaning of coded values. Using an existing ADaM variable for this purpose obviates the need for a new, non-standard variable that would require additional documentation, programming, and quality control. Note however that although the use of AVISITN for plotting purposes may work for a number of clinical trial settings, it may not work for all situations.

It should be pointed out that the use of ADY is appropriate for patient-level plots only, such as scatter plots. The current “unmet need” with the ADaM standard is a time variable that can be used for grouping records from the same analysis visit from which descriptive statistics for each treatment group, cohort, subgroup, etc. can be calculated.

### EXAMPLE USING AVISITN AS A PLOTTING VARIABLE

The following example is partially based on a real-life Phase 3 study that can be read about at: <http://www.macrogenics.com/content/file/Lancet28JUN2011F.pdf>

Consider a randomized study with daily dosing of an experimental therapy and placebo for 14 days followed by an observation period out to one year after randomization. The visit schedule includes assessments for safety on Study Days 1-7, 10, 14, 28, 91, 182, 273, and 365. There is also a Screening visit that should occur 2-3 weeks before the randomization visit and an End of Study visit where safety data are recorded. Efficacy data are also captured at some of these visits. The values for VISIT and VISITNUM from the raw (SDTM) data for this study are displayed in Table 1.

Table 1: VISITNUM and VISIT values

VISITNUM	VISIT
-1	SCREENING
1	DAY 1
2	DAY 2
3	DAY 3
4	DAY 4
5	DAY 5
6	DAY 6
7	DAY 7
8	DAY 10
9	DAY 14
10	MONTH 1
11	MONTH 3
12	MONTH 6
13	MONTH 9
14	YEAR 1
99	END OF STUDY

In order to make the visit number usable for plotting purposes, AVISITN should be assigned to represent the study day. The values of AVISIT should be assigned to represent whatever is preferred for display purposes. In this case we will assume that it is preferred to have AVISIT display both the target study day (if applicable) and the month and/or year. Table 2 displays the same information as Table 1 plus values for AVISITN and AVISIT:

Table 2: VISITNUM, VISIT, AVISITN, and AVISIT values

VISITNUM	VISIT	AVISITN	AVISIT
-1	SCREENING	-21	Screening
1	DAY 1	1	Day 1
2	DAY 2	2	Day 2
3	DAY 3	3	Day 3
4	DAY 4	4	Day 4
5	DAY 5	5	Day 5
6	DAY 6	6	Day 6
7	DAY 7	7	Day 7
8	DAY 10	10	Day 10
9	DAY 14	14	Day 14
10	MONTH 1	28	Day 28
11	MONTH 3	91	Day 91/Month 3
12	MONTH 6	182	Day 182/Month 6
13	MONTH 9	273	Day 273/Month 9
14	YEAR 1	365	Day 365/Month 12
99	END OF STUDY	999	End of Study

The value of -21 was chosen for AVISITN and the Screening visit because this represents the expected mode (most common value) for the timing of the screening visit and was considered logical for plotting purposes.

The value of 999 was chosen for the End of Study visit so that the value would not be confused with a study day. Plotting this value might require a re-assignment of the value if it is desired to have it appear immediately after the last displayed visit on a time scale. Since the last displayed visit can depend on what is being plotted, there is no universal value that would work for all plots on a time scale.

Examples of some actual plots from the study that this example is based on are presented in Figure 1 and Figure 2 below.

Figure 1: Sample Plot of Safety Data Using AVISITN for the x-axis of time

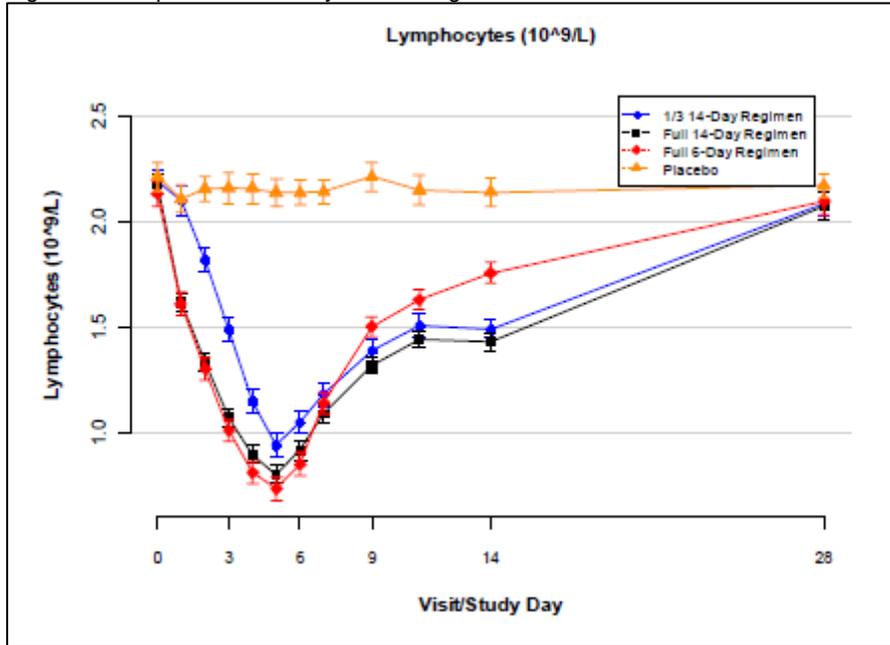
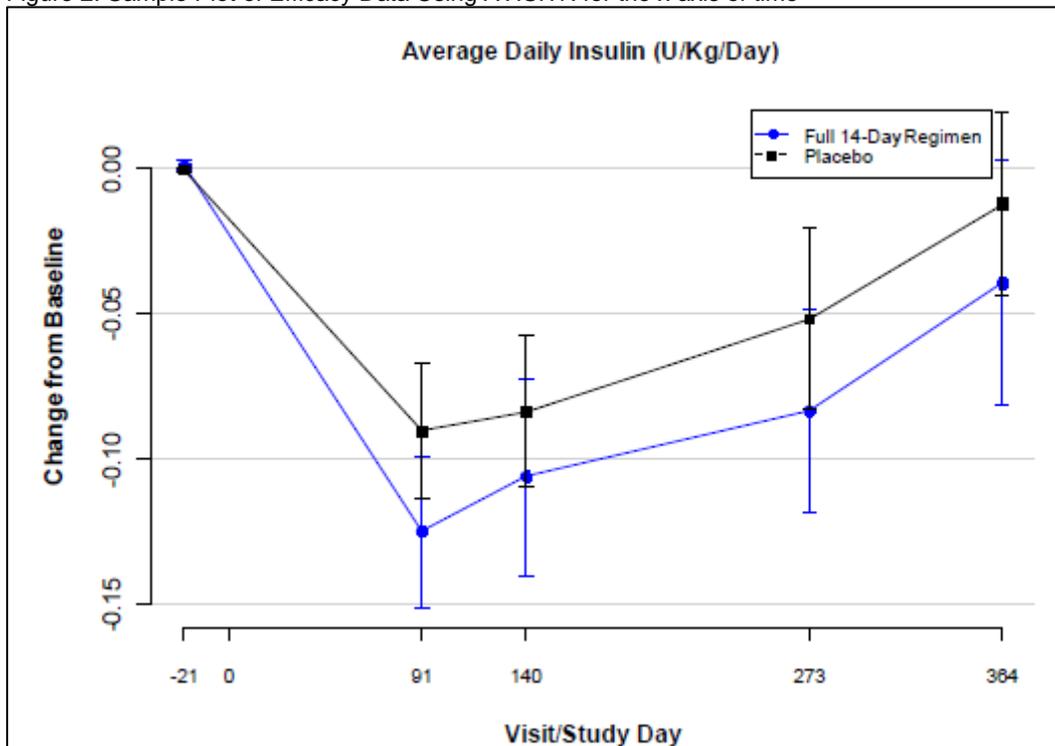


Figure 2: Sample Plot of Efficacy Data Using AVISITN for the x-axis of time



## ADVANTAGES AND DISADVANTAGES

Advantages and disadvantages to this approach relative to one where a new variable is instead applied at all times are provided in the table below:

Using AVISITN Whenever Feasible		Using a New Variable for all Situations	
Advantages	Disadvantages	Advantages	Disadvantages
Works for the vast majority of situations	May not work for all studies or situations, in which case an additional variable is needed	Can be used for all situations where the concept is relevant	Creates additional work and documentation
Applies an intuitive value to AVISITN rather than a coded value that can be hard to remember or interpret	Meets resistance with those who use AVISITN according to certain company-specific standards (e.g. equal to VISITNUM)		Allows continued use of AVISITN in a non-standard way
Reduces the need to add additional variables to a data set			Adds additional non-standard variables to a data set
Creates a standard for the use of AVISITN			

Despite the many advantages of using AVISITN as a plotting variable, the fact that this will not work for all situations does warrant consideration of a new set of ADaM variables that are specifically designed for plotting purposes. To this end, there is currently an ADaM sub-team working on a solution to this. It remains to be seen whether AVISITN will be considered an acceptable option for plotting on a time scale after this new set of variables have been proposed

## HOW TO USE ADaM DATASET FOR GRAPH USING SAS ODS AND SAS/STAT

The below Time to Event ADaM dataset can create a Kaplan Meier Curve without the intermediate datasets. The table 3 and 4 are the metadata of ADaM dataset of ADTTEOS. Table 5 is the final ADaM dataset of ADTTEOS.

Table 3: Analysis Dataset Metadata for ADTTEOS

Dataset Name	Dataset Description	Dataset Location	Dataset Structure	Key Variables of Dataset	Class of Dataset	Documentation
ADTTEOS	Overall Survival Analysis Dataset	<a href="#">adtteos.xpt</a>	one record per subject per parameter per analysis visit	USUBJID, PARAMCD, AVISITN	BDS	c-adtteos.txt

Table 4: Analysis Variable Metadata including Analysis Parameter Value-Level Metadata for ADTTEOS

Dataset Name	Parameter Identifier	Variable Name	Variable Label	Variable Type	Display Format	Codelist / Controlled Terms	Source / Derivation
ADTTEOS	*ALL*	USUBJID	Unique Subject Identifier	text	\$20		ADSL.USUBJID
ADTTEOS	*ALL*	SITEID	Site ID	text	\$20		ADSL.SITEID
ADTTEOS	*ALL*	SEX	Sex	text	\$20	M, F	ADSL.SEX
ADTTEOS	*ALL*	ITTFL	Intent-to-Treat Population Flag	text	\$1	Y, N	ADSL.ITTFL
ADTTEOS	*ALL*	TRTPN	Planned Treatment (N)	integer	1.0	1 = Placebo, 2 = Study Drug	ADSL.TRTPN
ADTTEOS	*ALL*	TRTP	Planned Treatment	text	\$20	Placebo, Study Drug	ADSL.TRTP
ADTTEOS	PARAMCD	PARAMCD	Parameter Code	text	\$8	DEATH	
ADTTEOS	*ALL*	PARAM	Parameter	text	\$50	Days to Death	
ADTTEOS	*ALL*	PARAMTYP	Parameter Type	text	\$50	DERIVED	
ADTTEOS	*ALL*	AVISITN	Analysis Visit (N)	integer	3.0	1 = Treatment Period	
ADTTEOS	*ALL*	AVISIT	Analysis Visit	text	\$20	Treatment Period	
ADTTEOS	*ALL*	AVAL	Analysis Value	float	8.2		ADT – STARTDT + 1
ADTTEOS	*ALL*	STARTDT	Time to Event Origin Date for Subject	integer	DDMONYY		SAS Numeric version of DM.RFSTDTC
ADTTEOS	*ALL*	ADT	Analysis Date	integer	DDMONYY		SAS numeric version of DS.DSDTC, last event or finding date
ADTTEOS	*ALL*	CNSR	Censor	integer	8.0		

Dataset Name	Parameter Identifier	Variable Name	Variable Label	Variable Type	Display Format	Codelist / Controlled Terms	Source / Derivation
ADTTEOS	*ALL*	EVNTDESC	Event or Censoring Description	text	\$40		DS.DSDECOD

Table 5: Final ADaM dataset of ADTTEOS

USUBJID	SITEID	ITTFL	TRTP	PARAMCD	PARAM	AVAL	STARTDT	ADT	CNSR	EVNTDESC
001-01-001	01	Y	Study Drug	DEATH	Days to Death	157	2011-01-04	2011-06-10	1	COMPLETED STUDY
001-01-002	01	Y	Placebo	DEATH	Days to Death	116	2011-02-01	2011-05-28	1	LOST TO FOLLOW- UP
001-01-003	01	Y	Placebo	DEATH	Days to Death	88	2011-02-25	2011-05-04	0	DEATH
001-01-004	01	Y	Study Drug	DEATH	Days to Death	102	2011-03-20	2011-06-03	1	ONGOING
001-01-005	01	Y	Study Drug	DEATH	Days to Death	101	2011-03-26	2011-07-05	1	ONGOING
.....										

In the normal setting, a programmer creates the intermediate SAS datasets of \_SURV like below SAS codes.

```
proc lifetest data=param out=_surv;
  time aval*cnsr(1);
  strata trtp;

  **** Title & footnote;
  title1 "Survival Plot";
run;
```

Table 6: The Intermediate dataset of \_SURV for a Kaplan Meier Curve

TRP	AVAL	_CENSOR_	SURVIVAL	SDF_LCL	SDF_UCL	STRATUM
Placebo	0	.	1	1	1	1
Placebo	1	0	0.939189	0.886397	0.967887	1
Placebo	3	0	0.898649	0.837527	0.937622	1
.....						
Study Drug	0	.	1	1	1	2
Study Drug	2	0	0.960526	0.914251	0.98207	2
Study Drug	3	1	0.960526	.	.	2
.....						

A programmer can create a Kaplan Meier Curve using this intermediate temporary dataset using stepwise function in SAS/GRAPH. But, the new ODS function can help the programmer to create a Kaplan Meier Curve directly from ADaM dataset. The following SAS codes will provide a simple way to create a Kaplan Meier Curve.

### SAS CODES AND LOG

A programmer can write below SAS codes for a simple Kaplan Meier Curve utilizing ODS GRAPH and SAS/STAT.

```
**** Prepare Graph Environment;
options orientation=landscape missing="" nonumber nodate;

**** ODS Option;
ods trace on;
ods listing close;
ods path sashelp.tmplmst(read);
ods graphics on ;
ods rtf file="C:\zother\paper\13 ADaM datasets for Graphs\SAS\program\Survival Graph.rtf"
bodytitle ;
ods noptitle;
ods select survivalplot;

**** Graph;
proc lifetest data=param ;
    time aval*cnsr(1);
    strata trt01p;

    **** Title & footnote;
    title1 "Survival Plot";
run;

**** Closing ODS;
options orientation=landscape ;
ods path sashelp.tmplmst(read);
ods rtf close;
ods graphics off;
ods listing;
ods trace off;
```

A programmer selects ODS SELECT SURVIVALPLOT as highlighted. The below SAS log shows that SURVIVALPLOT was added to the ODS rtf output. It also shows that template is Stat.Lifetest.Graphics.ProductLimitSurvival.

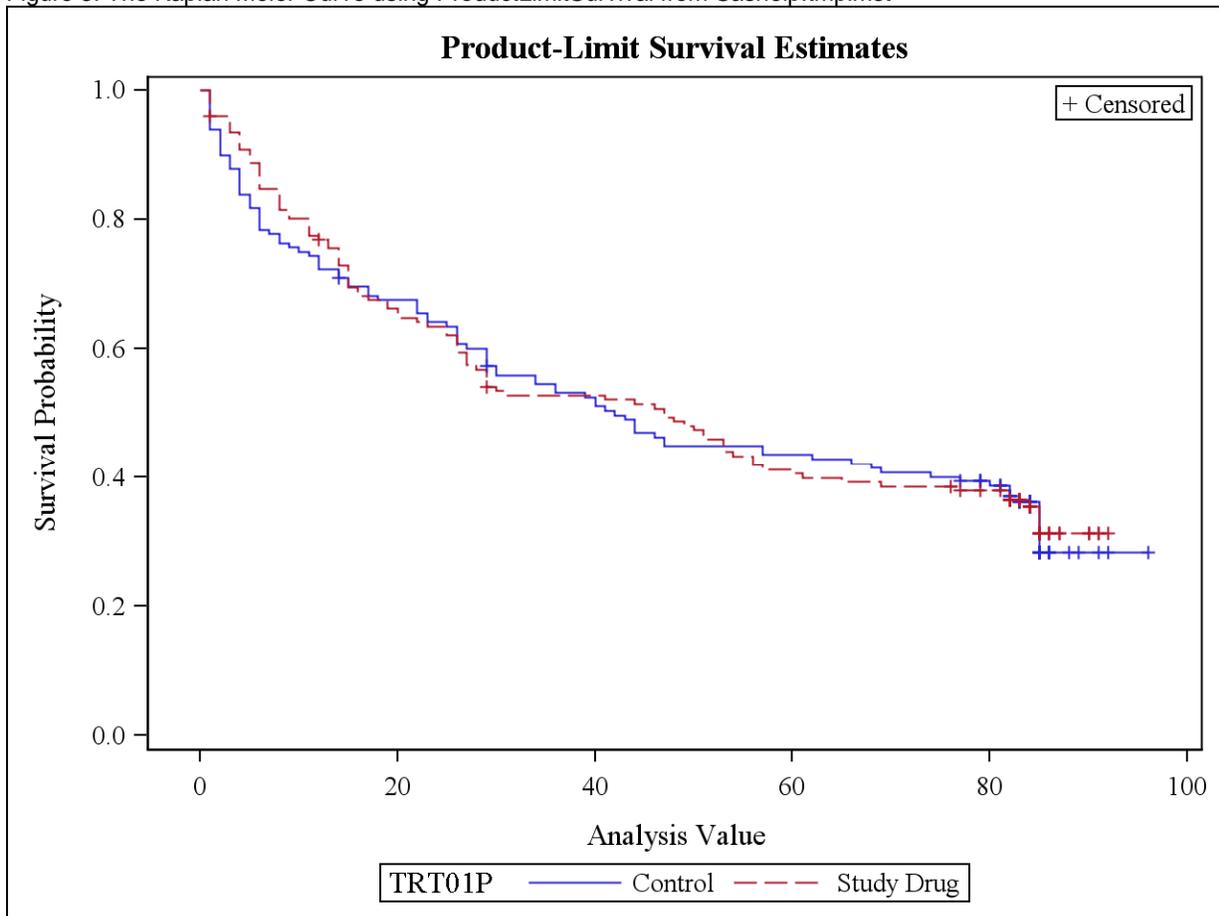
Output Added:

-----  
Name: SurvivalPlot  
Label: Survival Curves  
Template: Stat.Lifetest.Graphics.ProductLimitSurvival  
Path: Lifetest.SurvivalPlot  
-----

## OUTPUT

The above SAS codes create a simple Kaplan Meier Curve as seen in Figure 3. Figure 3 is a default Kaplan Meier Curve from the template of ProductLimitSurvival. If a programmer wants to create more customized graphs, the programmer needs to create the new template, which will be discussed in next sections.

Figure 3: The Kaplan Meier Curve using ProductLimitSurvival from Sashelp.tmplmst

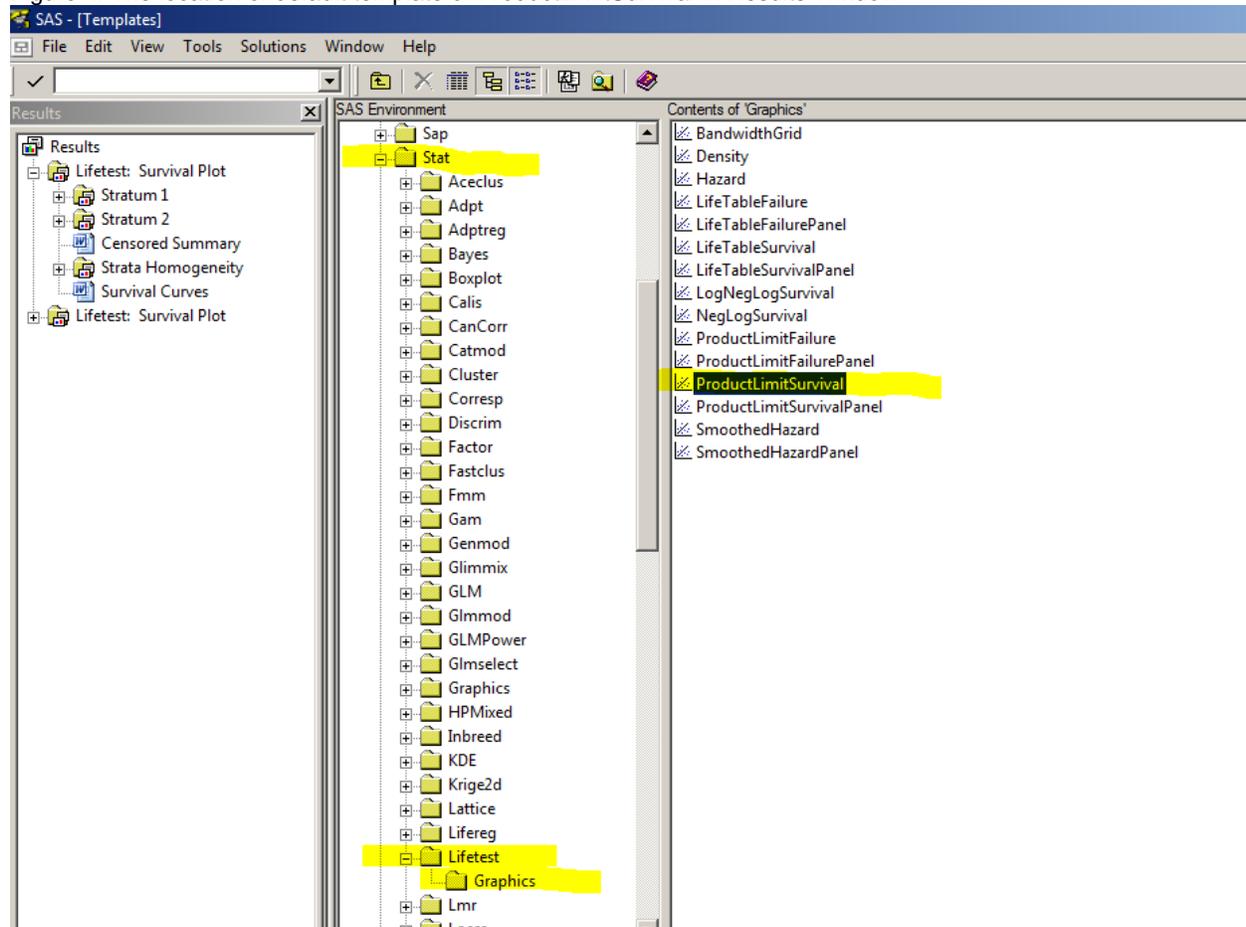


## GRAPH TEMPLATE LANGUAGE IN RESULTS AREA

In the yellow-highlighted, a programmer can find the location of template of ProductLimitSurvival template.

First, go to "Results" Window, right-click on "Results" and select "Template". Then, as shown below, go to "Sashelp.Tmplmst", "Stat", "Lifetest", "Graphics" and "ProductLimitSurvival".

Figure 4: The location of default template of ProductLimitSurvival in Results Window



Left-click on "ProductLimitSurvival", a programmer can find the GTL(Graph Template Language) for the template (Please see the Appendix 1). The programmer is able to modify this default template for more customized graphs.

The programmer can create the new template using GTL. Below template is created by modifying GTL from the default template.

```
**** Prepare Graph Environment;
options orientation=landscape missing="" nonumber nodate;

** Set margins for RTF destination ;
proc template;

**** New Survival Plot Template;
define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
dynamic NStrata xName plotAtRisk plotCensored plotCL plotHW plotEP
labelCL labelHW labelEP maxTime method StratumID classAtRisk plotBand
plotTest GroupName yMin Transparency SecondTitle TestName pValue;
BeginGraph;
entrytitle "" ; ** missing titles;
layout overlay /
axisopts=(label="Time(Days) on Study" shortlabel=XNAME offsetmin=.05
linearopts=(viewmax=MAXTIME))
yaxisopts=(label="Probability of Number of Events"
linearopts=(viewmin=0 viewmax=1
tickvaluelist=(0.0 0.2 0.4 0.5 0.6 0.8 1.0))); ** New tickets and label;

** Add the new lines;
lineparm x=0 y =0.5 slope=0 / extend = true lineattrs=(color=gray thickness=0.5px);
```

```

** New title in the legend;
DiscreteLegend "Survival" / title="Treatment Group" location=outside;

stepplot y=SURVIVAL x=TIME / group=STRATUM index=STRATUMNUM name=
  "Survival" rolename=( _tip1=ATRISK _tip2=EVENT) tip=(y x Time _tip1 _tip2);

if (PLOTCESTORED)
  scatterplot y=CENSORED x=TIME / group=STRATUM index=STRATUMNUM
  markerattrs=(symbol=circle); ** circle symbol;
if (PLOTTEST)
  else
    layout gridded / rows=1 autoalign=(TOPRIGHT BOTTOMLEFT TOP
    BOTTOM) border=false BackgroundColor=GraphWalls:Color Opaque=
    true; ** No border;
    entry "o Censored"; ** change legend ;
    endlayout;
  endif;
endif;
endlayout;
EndGraph;
end;
run;

**** ODS Option;
ods trace on;
ods listing close;
ods path sasuser.templat(read) sashelp.tmplmst(read);
ods graphics on ;
ods rtf file="C:\zother\paper\13 ADaM datasets for Graphs\SAS\program\Survival Graph.rtf"
bodytitle ;
ods noprt;
ods select survivalplot;

**** Graph;
proc lifetest data=param ;
  time aval*cnsr(1);
  strata trt01p;

  **** Title & footnote;
  title1 h=&height f=&font "Survival Plot";
run;

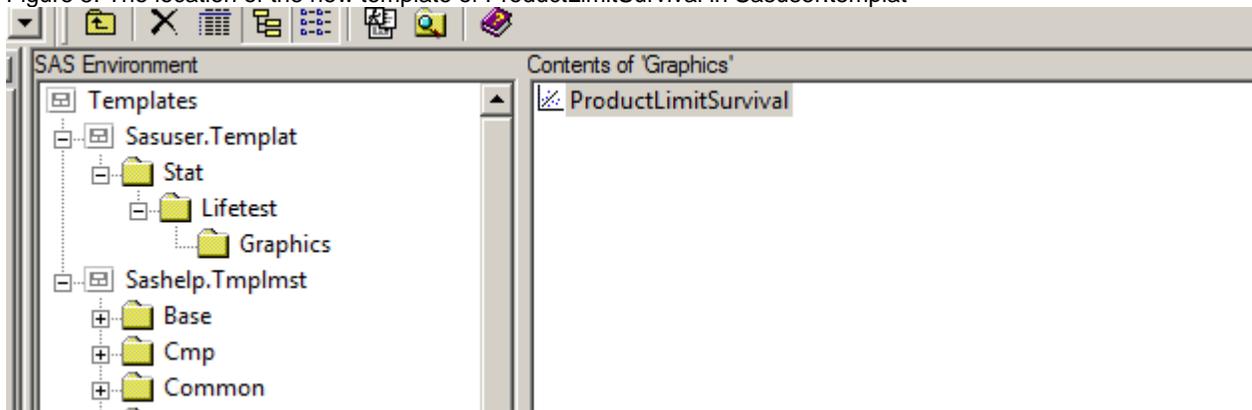
**** Closing ODS;
options orientation=landscape ;
ods path sashelp.tmplmst(read);
ods rtf close;
ods graphics off;
ods listing;
ods trace off;

```

The new template is created in proc template step as shown in SAS LOG (below yellow-highlighted) and in Figure 5.

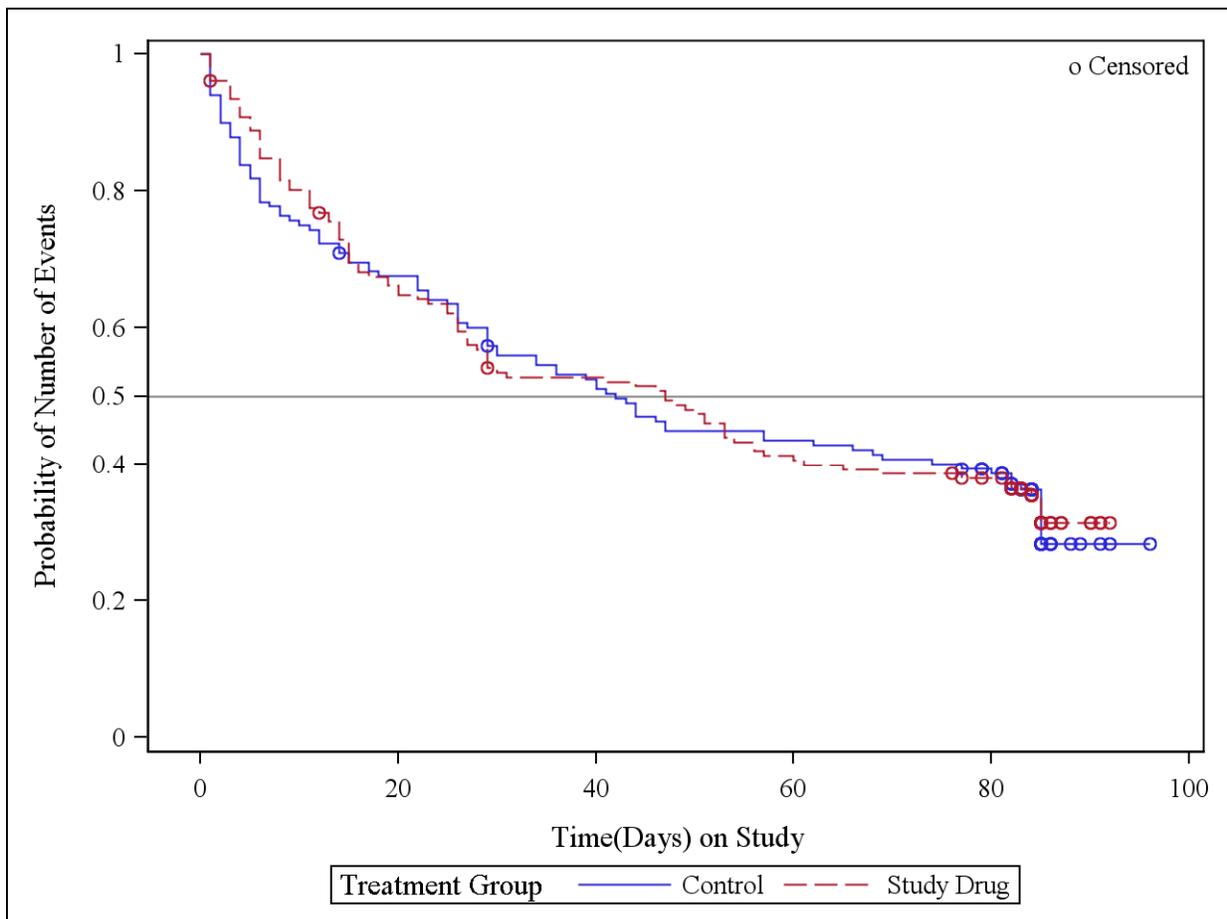
NOTE: STATGRAPH 'Stat.Lifetest.Graphics.ProductLimitSurvival' has been saved to: SASUSER.TEMPLAT

Figure 5: The location of the new template of ProductLimitSurvival in Sasuser.templat



ODS PATH(green-highlighted) calls the new template in Sasuser.Templat first before the default template in Sashelp.Tmplmst. Figure 6 is the customized Kaplan Meier Curve that the new template creates.

Figure 6: Kaplan Meier Curve from modified Graph Template Language  
Survival Plot



## **CONCLUSION**

The programmer can use AVISITN as a plotting variable in the graphs, but as discussed in the section of ADVANTAGES AND DISADVANTAGES, there might be a need to create new variables that can work for all situations. An ADaM Graph sub-team is looking at other variables that can help the programmers prepare more graph-ready ADaM datasets.

For certain plotting, the programmers need to create the intermediate SAS datasets from ADaM datasets. Using ODS GRAPH and GTL, the programmer can create some graphs directly from ADaM datasets without creating the intermediate SAS datasets.

## **REFERENCES**

The ADaM Implementation Guide, Version V 1.0 (ADaMIG v1.0)

The Analysis Data Model, Version 2.1 (ADaM 2.1)

The ADaM Basic Data Structure for Time-to-Event Analyses

CDER Common Data Standards Issues Document (Version 1.1/December 2011)

Study Data Specifications (July 18, 2012/Version 2.0)

Teplizumab for treatment of type 1 diabetes (Protégé study): 1-year results from a randomised, placebo-controlled trial

## **ACKNOWLEDGEMENT**

We would like to acknowledge and extend my heartfelt gratitude to ADaM Graph subteam – Kent Letoureau, Mario Widel, Elizabeth Li and Behrang Vali.

## **CONTACT INFORMATION**

Your comments and questions are valued and welcomed. Please contact the author at

Kevin Lee  
Cytel, Inc.  
Chesterbrook, PA  
(610) 994 - 9840  
Email:Kevin.lee@cytel.com

Chris Holland  
Amgen Inc.  
Email:cholland@amgen.com

## **TRADEMARKS**

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 registered trademarks or trademarks of their respective companies.

## APPENDIX 1. GTL of Stat.Lifetest.Graphics.ProductLimitSurvival in SASUSER.TMPLMST

```
proc template;
  define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
    dynamic NStrata xName plotAtRisk plotCensored plotCL plotHW plotEP
      labelLCL labelHW labelEP maxTime method StratumID classAtRisk plotBand
      plotTest GroupName yMin Transparency SecondTitle TestName pValue;
    BeginGraph;
      if (NSTRATA=1)
        if (EXISTS(STRATUMID))
          entrytitle METHOD " " "Survival Estimate" " for " STRATUMID;
        else
          entrytitle METHOD " " "Survival Estimate";
      endif;
      if (PLOTATRISK)
        entrytitle "with Number of Subjects at Risk" / textattrs=
          GRAPHVALUETEXT;
      endif;
      layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05 linearopts
        =(viewmax=MAXTIME)) yaxisopts=(label="Survival Probability"
        shortlabel="Survival" linearopts=(viewmin=0 viewmax=1
        tickvaluelist=(0 .2 .4 .6 .8 1.0)));
      if (PLOTBW=1 AND PLOTEP=0)
        bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME / modelname
          ="Survival" fillattrs=GRAPHCONFIDENCE name="HW" legendlabel=
          LABELHW;
      endif;
      if (PLOTBW=0 AND PLOTEP=1)
        bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME / modelname
          ="Survival" fillattrs=GRAPHCONFIDENCE name="EP" legendlabel=
          LABELEP;
      endif;
      if (PLOTBW=1 AND PLOTEP=1)
        bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME / modelname
          ="Survival" fillattrs=GRAPHDATA1 datatransparency=.55 name="HW"
          legendlabel=LABELHW;
        bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME / modelname=
          "Survival" fillattrs=GRAPHDATA2 datatransparency=.55 name="EP"
          legendlabel=LABELEP;
      endif;
      if (PLOTCL=1)
        if (PLOTBW=1 OR PLOTEP=1)
          bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME /
            modelname="Survival" display=(outline) outlineattrs=
            GRAPHPREDICTIONLIMITS name="CL" legendlabel=LABELCL;
        else
          bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME /
            modelname="Survival" fillattrs=GRAPHCONFIDENCE name="CL"
            legendlabel=LABELCL;
        endif;
      endif;
      stepplot y=SURVIVAL x=TIME / name="Survival" rolename=( _tip1=
        ATRISK _tip2=EVENT) tip=(y x Time _tip1 _tip2) legendlabel=
        "Survival";
      if (PLOTCEM=1)
        scatterplot y=CENSORED x=TIME / markerattrs=(symbol=plus) name=
          "Censored" legendlabel="Censored";
      endif;
      if (PLOTCL=1 OR PLOTBW=1 OR PLOTEP=1)
        discretelegend "Censored" "CL" "HW" "EP" / location=outside
          halign=center;
      endif;
    endgraph;
  enddefine;
run;
```

```

else
  if (PLOTCESTORED=1)
    discretelegend "Censored" / location=inside autoalign=(topright
    bottomleft);
  endif;
endif;
if (PLOTATRISK=1)
  innermargin / align=bottom;
  blockplot x=TATRISK block=ATRISK / repeatedvalues=true display=
  (values) valuealign=start valuefitpolicy=truncate
  labelposition=left labelattrs=GRAPHVALUETEXT valueattrs=
  GRAPHDATATEXT (size=7pt) includemissingclass=false;
endinnermargin;
endif;
endlayout;
else
  entrytitle METHOD " " "Survival Estimates";
if (EXISTS(SECONDTITLE))
  entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
endif;
layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05 linearopts
=(viewmax=MAXTIME)) yaxisopts=(label="Survival Probability"
shortlabel="Survival" linearopts=(viewmin=0 viewmax=1
tickvaluelist=(0 .2 .4 .6 .8 1.0)));
if (PLOTBW)
  bandplot LimitUpper=HW_UCL LimitLower=HW_LCL x=TIME / group=
  STRATUM index=STRATUMNUM modelname="Survival" datatransparency=
  Transparency;
endif;
if (PLOTBP)
  bandplot LimitUpper=EP_UCL LimitLower=EP_LCL x=TIME / group=
  STRATUM index=STRATUMNUM modelname="Survival" datatransparency=
  Transparency;
endif;
if (PLOTCL)
  if (PLOTBAND)
    bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME / group=
    STRATUM index=STRATUMNUM modelname="Survival" display=(outline)
;

else
  bandplot LimitUpper=SDF_UCL LimitLower=SDF_LCL x=TIME / group=
  STRATUM index=STRATUMNUM modelname="Survival" datatransparency=
  Transparency;
endif;
endif;
stepplot y=SURVIVAL x=TIME / group=STRATUM index=STRATUMNUM name=
"Survival" rolename=( _tip1=ATRISK _tip2=EVENT) tip=(y x Time
_tip1 _tip2);
if (PLOTCESTORED)
  scatterplot y=CENSORED x=TIME / group=STRATUM index=STRATUMNUM
markerattrs=(symbol=plus);
endif;
if (PLOTATRISK)
  innermargin / align=bottom;
  blockplot x=TATRISK block=ATRISK / class=CLASSATRISK
  repeatedvalues=true display=(label values) valuealign=start
  valuefitpolicy=truncate labelposition=left labelattrs=
  GRAPHVALUETEXT valueattrs=GRAPHDATATEXT (size=7pt)
  includemissingclass=false;
endinnermargin;
endif;

```

```

DiscreteLegend "Survival" / title=GROUPNAME location=outside;
if (PLOTCEASURED)
  if (PLOTTEST)
    layout gridded / rows=2 autoalign=(TOPRIGHT BOTTOMLEFT TOP
    BOTTOM) border=true BackgroundColor=GraphWalls:Color Opaque=
    true;
    entry "+ Censored";
    if (PVALUE < .0001)
      entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
    else
      entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
    endif;
  endlayout;
else
  layout gridded / rows=1 autoalign=(TOPRIGHT BOTTOMLEFT TOP
  BOTTOM) border=true BackgroundColor=GraphWalls:Color Opaque=
  true;
  entry "+ Censored";
endlayout;
endif;
else
  if (PLOTTEST)
    layout gridded / rows=1 autoalign=(TOPRIGHT BOTTOMLEFT TOP
    BOTTOM) border=true BackgroundColor=GraphWalls:Color Opaque=
    true;
    if (PVALUE < .0001)
      entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
    else
      entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
    endif;
  endlayout;
endif;
endif;
endlayout;
EndGraph;
end;
run;

```