

Using the SAS/GRAPH® Annotate Facility to Create Timeline Plots

Erin L. Abner, Marta S. Mendiondo, University of Kentucky, Lexington, KY

ABSTRACT

A timeline plot is a simple but effective way of graphically representing events over time. Unlike survival curves, which can only be used to depict one-time events at the group-level, a timeline plot can be used to show transitions in and out of disease states at the individual level. Although SAS® does not automatically create timeline plots, the SAS/GRAPH® Annotate facility provides a convenient means of producing these graphs. This presentation will demonstrate how, using relatively simple syntax, an Annotate data set allows you to use multiple line types, colors, and symbols within the same plot line to depict any number of events over time.

Keywords: SAS/GRAPH, Annotate, timeline plot

INTRODUCTION

Survival curves graphically depict group trajectories to a common endpoint (e.g., time to death, disease onset, initiation of risky behaviors, etc.). That is, points on the curve represent what proportion of the group has yet to reach the endpoint at time t . But what if you wanted to graphically depict this endpoint at the individual level instead of the group level? Or what if the endpoint in question is impermanent, i.e., individuals may revert to earlier states or progress to later stages of disease? In such cases, you may wish to create a timeline plot, which graphically represents a chronological sequence of events.

Though SAS/GRAPH does not automatically create such plots, the Annotate facility, which allows you to generate custom graphics or enhance graphics output, provides a convenient means of producing timeline plots. Through simple manipulation of Annotate data set variables FUNCTION, X, Y, COLOR, LINE, SIZE, STYLE, and TEXT, you can use multiple line types, colors, and symbols within the same plot line to depict any number of events over time.

BUILDING AN ANNOTATE DATA SET

An Annotate data set is nothing more than a SAS data set, though it contains either a combination of user-defined variables and special Annotate variables or Annotate variables only. Using these special variables, each observation in the Annotate data set provides SAS graphics procedures with instructions on what to draw or what actions to perform. The Annotate variable FUNCTION tells SAS what to do, coordinate variables X and Y (and Z) tell SAS where to do it, and Annotate attribute variables tell SAS how to do it.

An Annotate data set is created through the SAS DATA step. After all the variable values for an observation have been assigned, an OUTPUT statement must be used to write the observation to the data set.

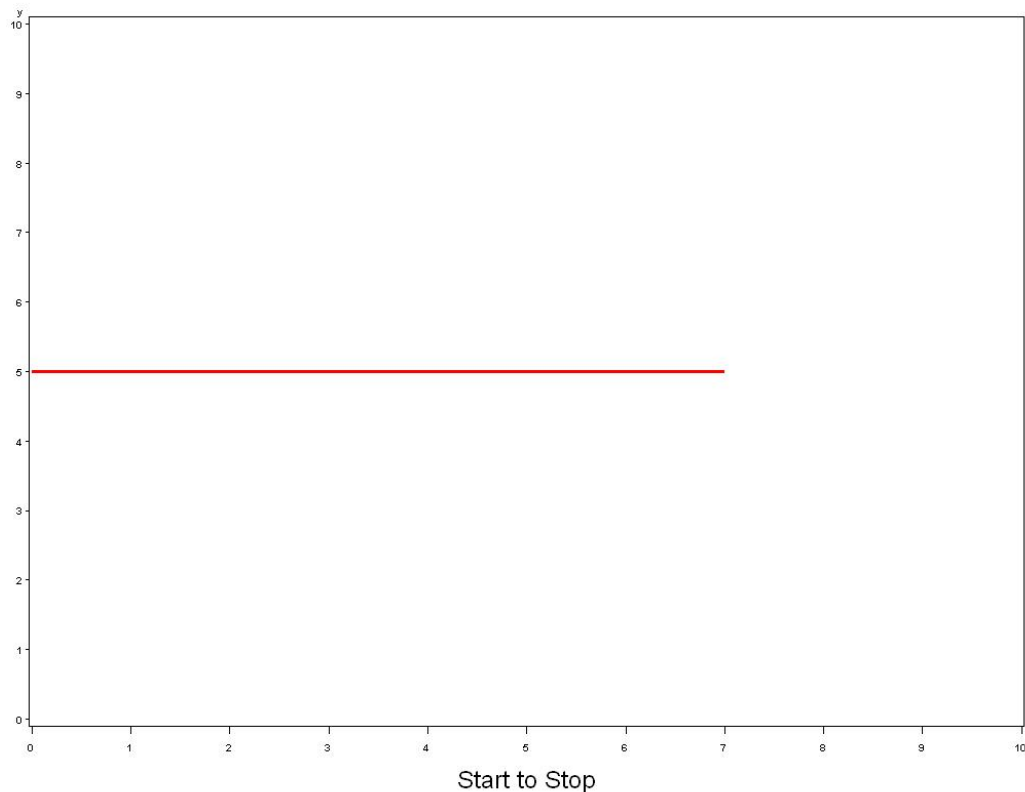
```
data anno;
  length function color $ 8;
  retain xsys ysys '2';
  function = 'move'; x = 0; y = 5; output;
  function = 'draw'; x = 7; y = 5; color = 'red'; size = 3; output;
run;
```

The above DATA step builds a simple Annotate data set that instructs SAS, via PROC GPLOT in this example, to draw a red line from 0 to 7 units on the horizontal axis that is 5 units up the vertical axis (Figure 1).

```
axis1 order = (0 to 10) minor = none label = (f = 'arial' h = 2 'Start to Stop');
axis2 order = (0 to 10) minor = none;
symbol i = none;

proc gplot;
  plot y*x / annotate = anno haxis = axis1 vaxis = axis2;
run;
quit;
```

FIGURE 1. START TO STOP



USING THE ANNOTATE FACILITY TO CREATE A TIMELINE PLOT

The above syntax forms the foundation of the timeline plot, but you will likely want to create a plot using an existing data set. In such cases it will not make sense to hard code the values for x and y one row at a time. Consider a data set (MYDATA) that contains information on a subset of fifty volunteers in a longitudinal study of aging and dementia. Cognitively normal elderly volunteers enroll and agree to annual examinations, which detect cognitive decline, and autopsy upon death (see Schmitt, 2006). The data set contains dates of enrollment, examination dates, and, where applicable, any diagnosis dates (of clinically significant cognitive impairment and/or dementia) and dates of death. Using these dates we can depict the trajectory of each volunteer from enrollment to the current state of cognition or death. If we want to see at a glance how long each volunteer has been followed relative to the other volunteers, we create a variable that starts all volunteers at time zero. We might call this variable START and give it a value of zero for all volunteers. Next we wish to find out how long each volunteer was followed from enrollment (START) to the last examination. This is simple matter of subtracting the enrollment date from the last examination date. We might store this value (which is measured in months in this example) in a variable called STOP. Each volunteer is uniquely identified by a variable called ID. At this point we could create an Annotate data set as above with the following modifications (in bold):

```
data anno;
length function color $ 8 ;
retain xsys ysys '2' size 2;
set mydata;
function = 'move'; x = start; y = id; output;
function = 'draw'; x = stop; y = id; color = 'black'; output;
run;

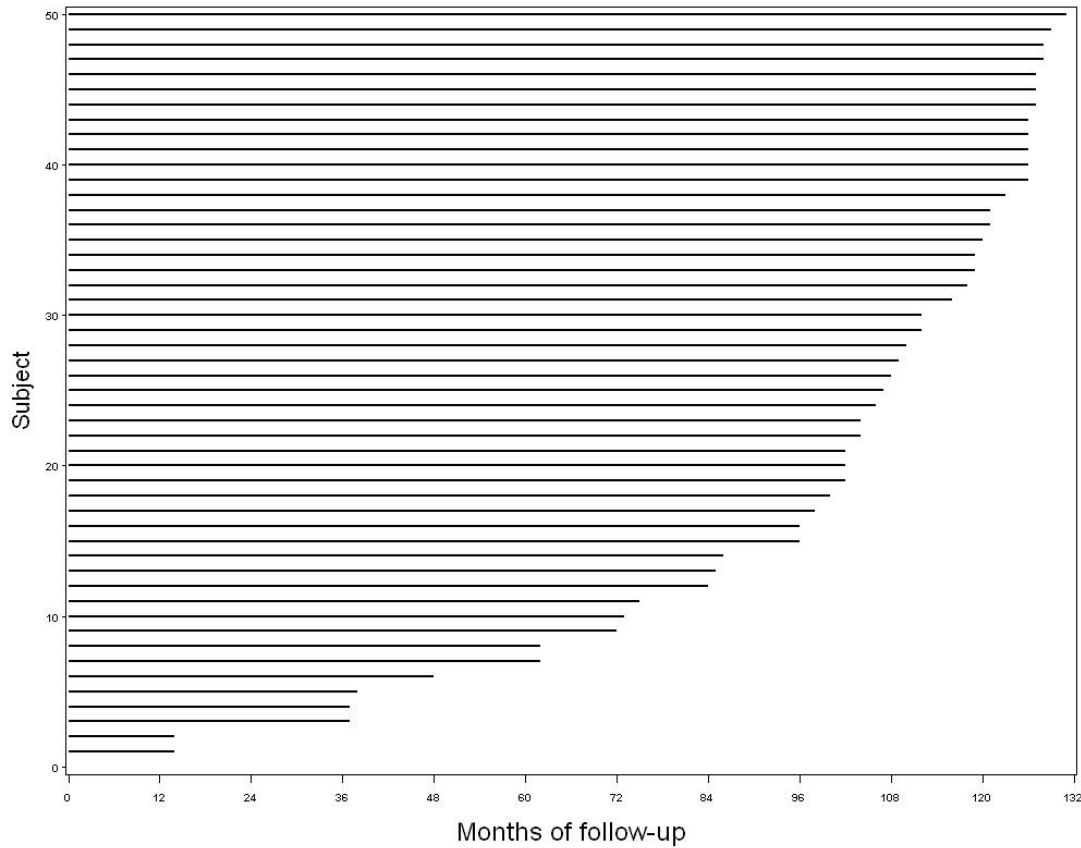
axis1 order = (0 to 10) minor = none label = (f = 'arial' h = 2 'Months of follow-
up');
axis2 order = (0 to 10) minor = none label = (a = 90 f = 'arial' h = 2 'Subject');
symbol i = none;

proc gplot;
plot id*start / annotate = anno haxis = axis1 vaxis = axis2;
```

```
run;
quit;
```

In this case, PROC GPLOT will produce a graph where each line represents a volunteer's time in the study (Figure 2). The volunteer's unique identifier is on the y-axis and time on study in months is on the x-axis. SAS processes observations in the Annotate data set one row at a time beginning with the first observation. In this example, it was desired to order the lines in the plot by the volunteers' length of time spent in the study. This was achieved by sorting the data set by STOP, and then assigning ID based on position in the data set. That is, the volunteer with the least time in the study was assigned ID = 1, and the volunteer with the most time in the study was assigned ID = 50.

FIGURE 2. BASIC TIMELINE PLOT: MONTHS ENROLLED IN THE STUDY



It may seem that a horizontal bar chart would have done the job just as well, and that may be true if we were to stop here. But, the Annotate facility allows us to do much more. Suppose we want to show which subjects have died. We can create a variable, which we'll call DECEASED, that contains the interval in months from the last examination to the volunteer's death. Suppose in addition to showing which volunteers have died, we also want to show which volunteers have transitioned from a cognitively normal state into a cognitively impaired or demented state. In the same way, we can create variables, we'll call them STOP_MCI and STOP_AD, that contain the interval in months from cognitive impairment or dementia to last examination, or from cognitive impairment to dementia. To clearly depict these different states on the graph we'll need to use different colors and line types.

The syntax needed to achieve this plot is a bit longer, and only slightly more complicated than the above example.

```
data anno;
length function color $8.;
retain xsys ysys '2' size 2;
set mydata;

function = 'move'; x = start; y = id; color = 'black'; output;
function = 'draw'; x = stop; y = id; output;

if stop_mci ne . then do;
function = 'move'; x = stop_mci; y = id; output;
```

```

        function = 'draw'; x = stop;      y = id; color = 'blue'; output;
end;

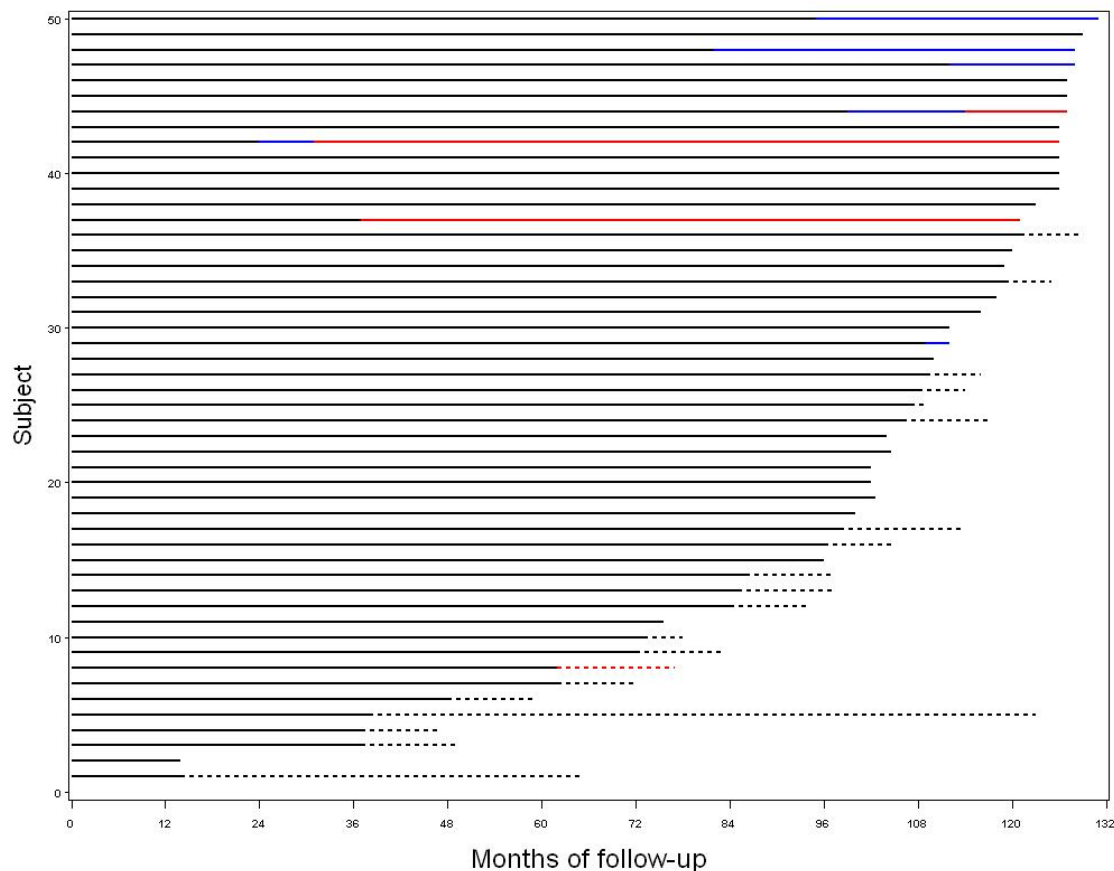
if stop_AD ne . then do;
    function = 'move'; x = stop_ad;  y = id; output;
    function = 'draw'; x = stop;      y = id; color = 'red'; output;
end;

if deceased ne . then do;
    function = 'move'; x = stop;      y = id; ; output;
    function = 'draw'; x = deceased; y = id; line = 2; output;
end;
run;

```

Conditional logic is used to ensure observations with instructions to change line colors and types are only output when appropriate. That is, for example, if it is true that STOP_MCI is not missing, then that volunteer transitioned to clinical impairment, thus the color of the line should change. Using the same syntax for PROC GLOT as above, Figure 3 is produced.

FIGURE 3. TIMELINE PLOT WITH MULTIPLE COLORS AND LINE TYPES



Here the time spent in a cognitively normal state is represented by black lines, time spent in the cognitively impaired state is represented by blue lines, and time spent in a demented state is represented by red lines. Dashed lines indicate the time from the volunteer's last in-person examination until death.

But, there is still more information we can add to the graph. It is possible, for example, that a diagnosis of impairment or dementia was made at a volunteer's last examination and would therefore not be depicted by a change in line color. In other words, STOP and STOP_MCI or STOP_AD would have the same value so there would be no line drawn between them. We could instead represent this transition with a symbol, with the color of the symbol indicating the diagnosis (blue for impaired, red for demented). We might also add a symbol to mark the time when death occurred. Again, the required syntax remains relatively simple:

```

data anno;
length function color $8.;
retain xsys ysys '2';
set mydata;
size = 2;

function = 'move'; x = start; y = id; color = 'black'; output;
function = 'draw'; x = stop; y = id; output;

if stop_mci ne . then do;
    function = 'move'; x = stop_mci; y = id; output;
    function = 'draw'; x = stop; y = id; color = 'blue'; output;
end;

if stop = stop_mci then do;
    function = 'move'; x = stop; y = id; ; output;
    function = 'symbol'; x = stop; y = id; ; text = 'C'; style = 'marker';
    color = 'blue'; size = 1; output;
end;

if stop_ad ne . then do;
    function = 'move'; x = stop_ad; y = id; color = 'red'; output;
    function = 'draw'; x = stop; y = id; color = 'red'; output;
end;

if stop = stop_ad then do;
    function = 'move'; x = stop; y = id; ; color = 'red'; output;
    function = 'symbol'; x = stop; y = id; ; color = 'red'; text = 'C';
    style = 'marker'; size = 1; output;
end;

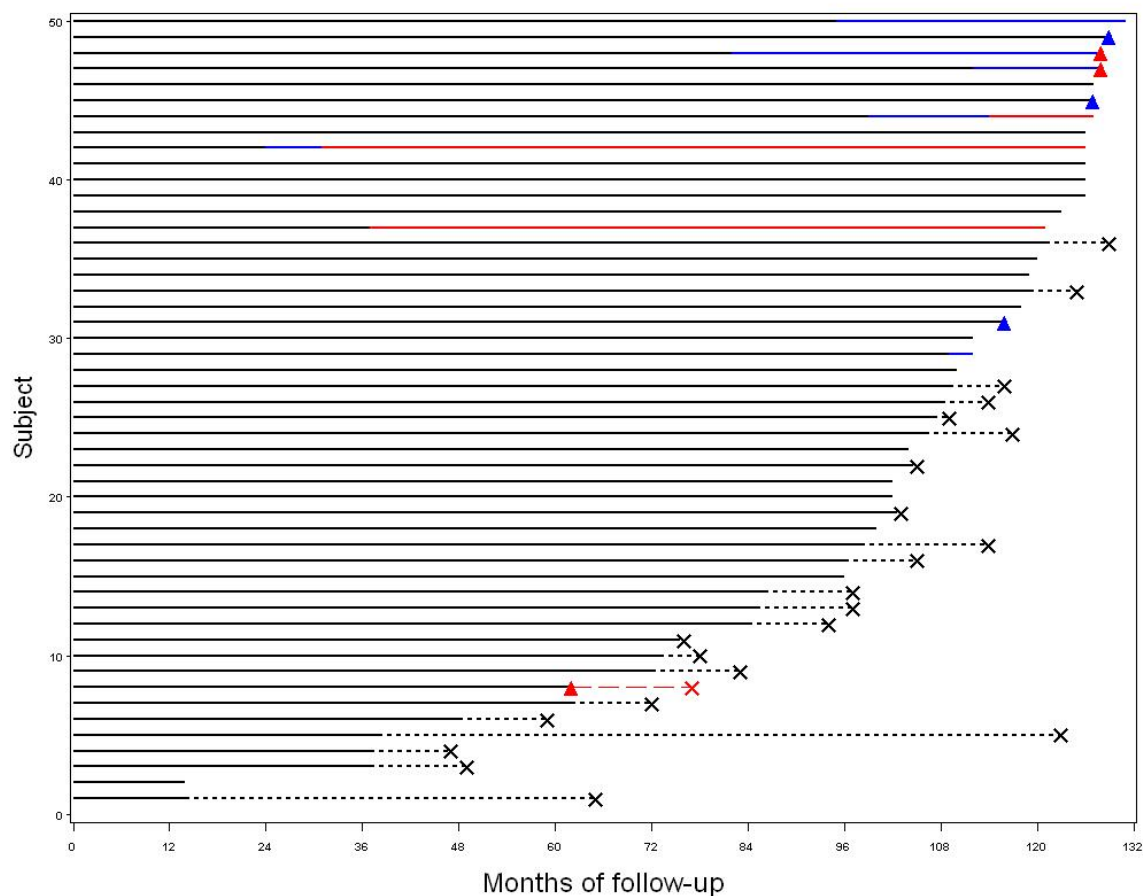
if deceased ne . then do;
    function = 'move'; x = stop; y = id; ; output;
    function = 'draw'; x = deceased; y = id; line = 2; output;
    function = 'move'; x = deceased; y = id; output;
    function = 'symbol'; x = deceased; y = id; text = 'X'; style = 'marker';
    size = 1; output;
end;
run;

```

In the Annotate data set, symbol instructions are first created by assigning FUNCTION the value SYMBOL. The attributes of the symbol are determined by the values of COLOR, SIZE, TEXT, and STYLE, among others. MARKER is one of several special SAS graphics fonts that include additional graphics symbols. Notice that SIZE has been moved from the RETAIN statement to an assignment statement. Assigning the value of SIZE in the RETAIN statement is not appropriate in this case because it will cause the value of SIZE to be retained whenever it is changed. That is, instead of changing the value of SIZE to 1 for only those observations where STOP = STOP_MCI, the value of 1 would be retained for all subsequent observations until the value was changed again.

Again using the same syntax for PROC GPLOT, Figure 4 is produced. Here a diagnosis of impairment at the last examination is represented by blue triangles, dementia by red triangles, and death by Xs. If an individual died in a state other than normal cognition, the color of the X reflects the diagnosis. Notice that subjects 49, 46, and 31 have blue triangles and the end of solid black lines, indicating a diagnosis of clinically significant impairment at their last evaluation. Subject 8 was diagnosed with dementia at his or her last clinical evaluation, and was not seen again in-person before death, which is reflected by a red triangle at the end of a solid black line, followed by a red dashed line and X.

FIGURE 4. TIMELINE PLOT WITH MULTIPLE COLORS, LINE TYPES, AND SYMBOLS



CONCLUSION

The SAS/GRAPH Annotate facility provides a powerful means of producing custom graphics or enhancing graphics output. With a simple Annotate data set, a timeline plot can depict events over time with multiple colors, symbols, and line types. Timeline plots should be considered whenever multiple events over time need to be graphically represented, although very large samples (over 250) are not well-suited to such plots.

REFERENCES

Mink, D. and Pasta, D.J. "Improving Your Graphics Using SAS/GRAPH® Annotate Facility". *Proceedings of the Thirty-first Annual SAS® Users Group International Conference*. Cary, NC: SAS Institute Inc., 2006.

"SAS Code Fragments: How to Create a Timeline Plot". UCLA: Academic Technology Services, Statistical Consulting Group. http://www.ats.ucla.edu/stat/sas/code/timeline_plot.htm (accessed March 31, 2008).

SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

Schmitt FA, Davis DG, Wekstein DR, Smith CD, Ashford JW, Markesbery WR. "Preclinical" AD revisited: neuropathology of cognitively normal older adults. *Neurology*. 2000;55:370-376

ACKNOWLEDGMENTS

Special thanks to all the research volunteers, past and present, at the Sanders-Brown Center on Aging.

CONTACT INFORMATION

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

Erin L. Abner, MPH
Sanders-Brown Center on Aging
University of Kentucky
800 S. Limestone St.
Lexington, KY 40536
Work Phone: 859-257-1412
Fax: 859-323-2866
E-mail: elabne0@email.uky.edu

Marta S. Mendiondo, Ph.D.
Sanders-Brown Center on Aging
University of Kentucky
800 S. Limestone St.
Lexington, KY 40536
Work Phone: 859-257-1412
Fax: 859-323-2866
E-mail: marta@email.uky.edu

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.