

## Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically

Perry Watts, SAS® User, Elkins Park, PA

### ABSTRACT

The conventional order clause in the axis statement for SAS/GRAPH® software is well-suited for scatter plots with continuous data or line plots where discrete data are evenly spaced apart. However, accommodating discrete ordinal data with uneven intervals requires a two-step approach. First, a hidden conventional axis with narrow intervals needs to be generated where ticks and labeling are turned off. Then the hidden axis is overlaid by the uneven scale that is plotted from ANNOTATE. With ANNOTATE there is no need to hard-code formats or label ticks in an axis statement. Instead, full automation can be achieved by combining ANNOTATE with a few macros.

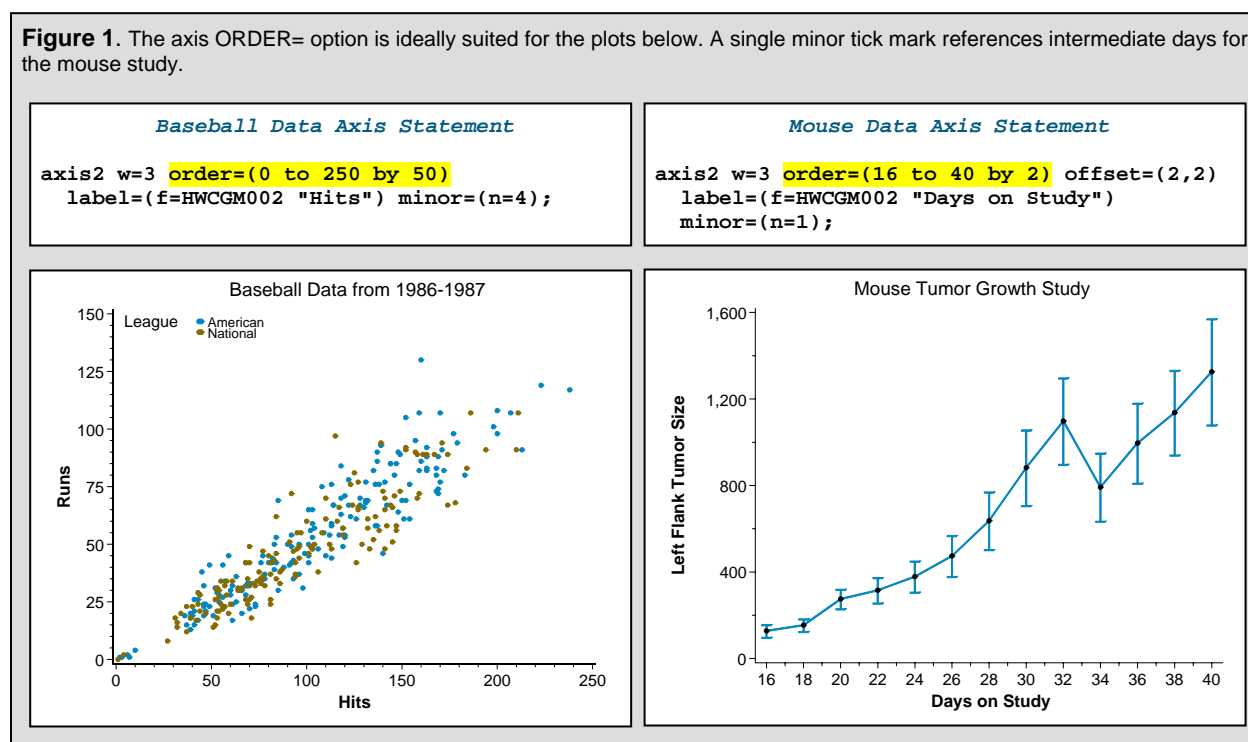
Included in the paper are relevant graphics examples such as needle plots, box plots, uneven width histograms, and embedded bar charts that rely on multiple horizontal axes for their display. Associated macros and a calling program can be found in the Appendix.

### PROBLEM DEFINITION

Often it is necessary to produce a graph with an axis containing major tick marks positioned at unevenly spaced intervals. Unfortunately nothing is set up in the axis statement of SAS/GRAPH software to automatically handle this situation. The ORDER= option always handles values as if they were evenly spaced apart, and using a small inclusive interval will clutter a graph and confuse the viewer since all major ticks are labeled by default. A format can sometimes be used to remove the unwanted value labels, but in the case of plots where dithering separates class values, ticks associated with the unwanted values should be removed as well. To remove ticks, the original axis needs to be replaced by an axis drawn in ANNOTATE. The paper shows step-by-step how the replacement is made with little or no hard-coding.

### HOW THE ORDER= OPTION IS TYPICALLY USED IN AN AXIS STATEMENT

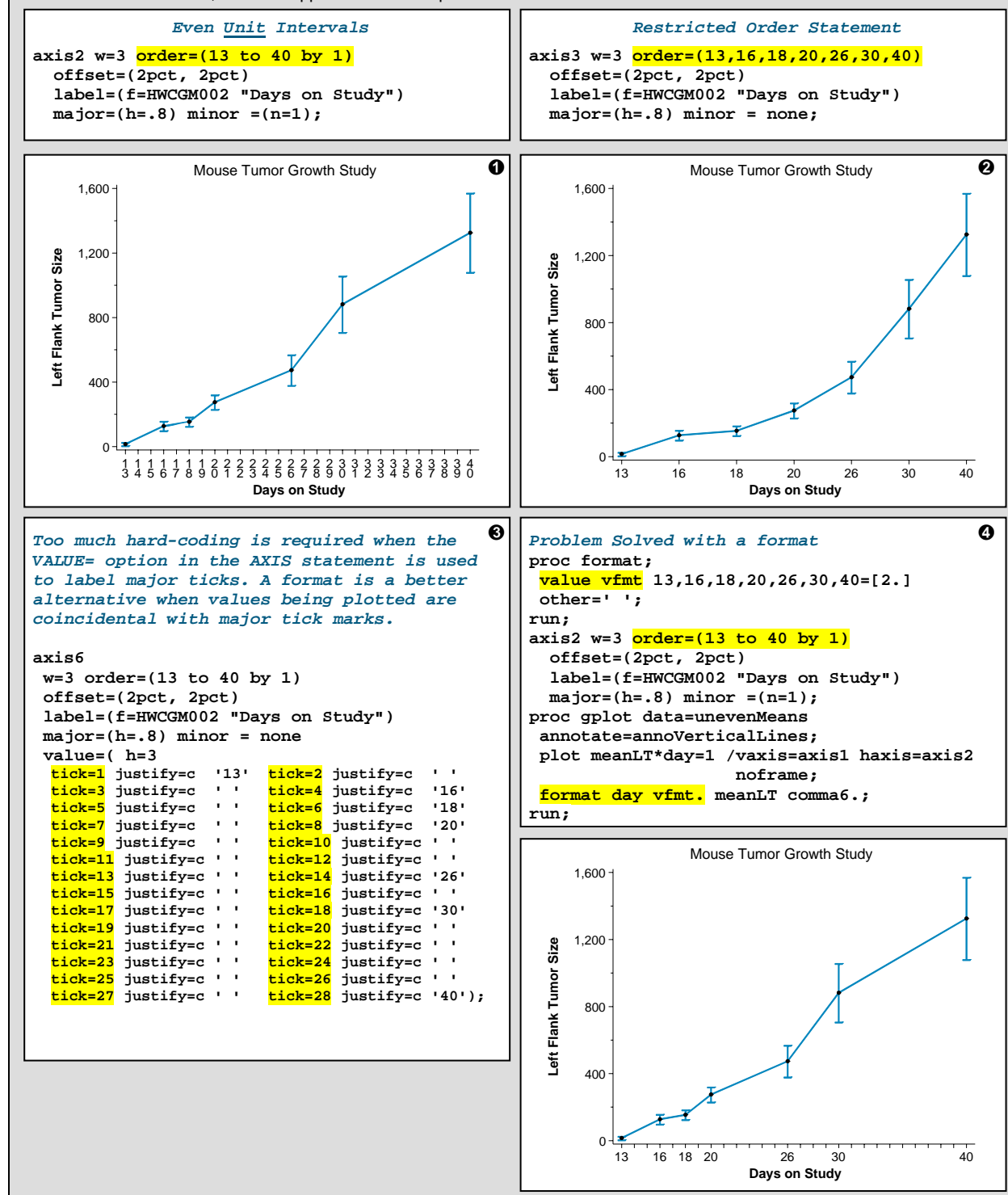
As demonstrated in Figure 1, the ORDER= option accommodates scatter plots with continuous data or plots where the discrete data along the horizontal axis are evenly spaced apart. The range syntax for the ORDER= option is straight-forward with *m* TO *n* <BY *increment*> [5, p. 131].



## ACCOMMODATING UNEVEN-INTERVAL AXIS LABELS WITH A FORMAT

The situation is not so straight-forward when data are collected at irregular intervals in time. To eliminate axis clutter from the first panel in Figure 2, for example, it might be tempting to limit ORDER= to the values listed in the data. However, when a value-list contains unequal intervals, a warning is written to the LOG, and a plot with equal-width intervals is incorrectly generated. (See panel 2). Using a format in the fourth panel provides a workable solution to the problem.

**Figure 2.** Uneven intervals in the mouse data are set at: 13-(2)-16-(1)-18-(1)-20-(5)-26-(3)-30-(9)-40. Vertical lines in the plots are drawn in ANNOTATE, because upper and lower "top" widths are not coordinated with the thicker vertical lines.



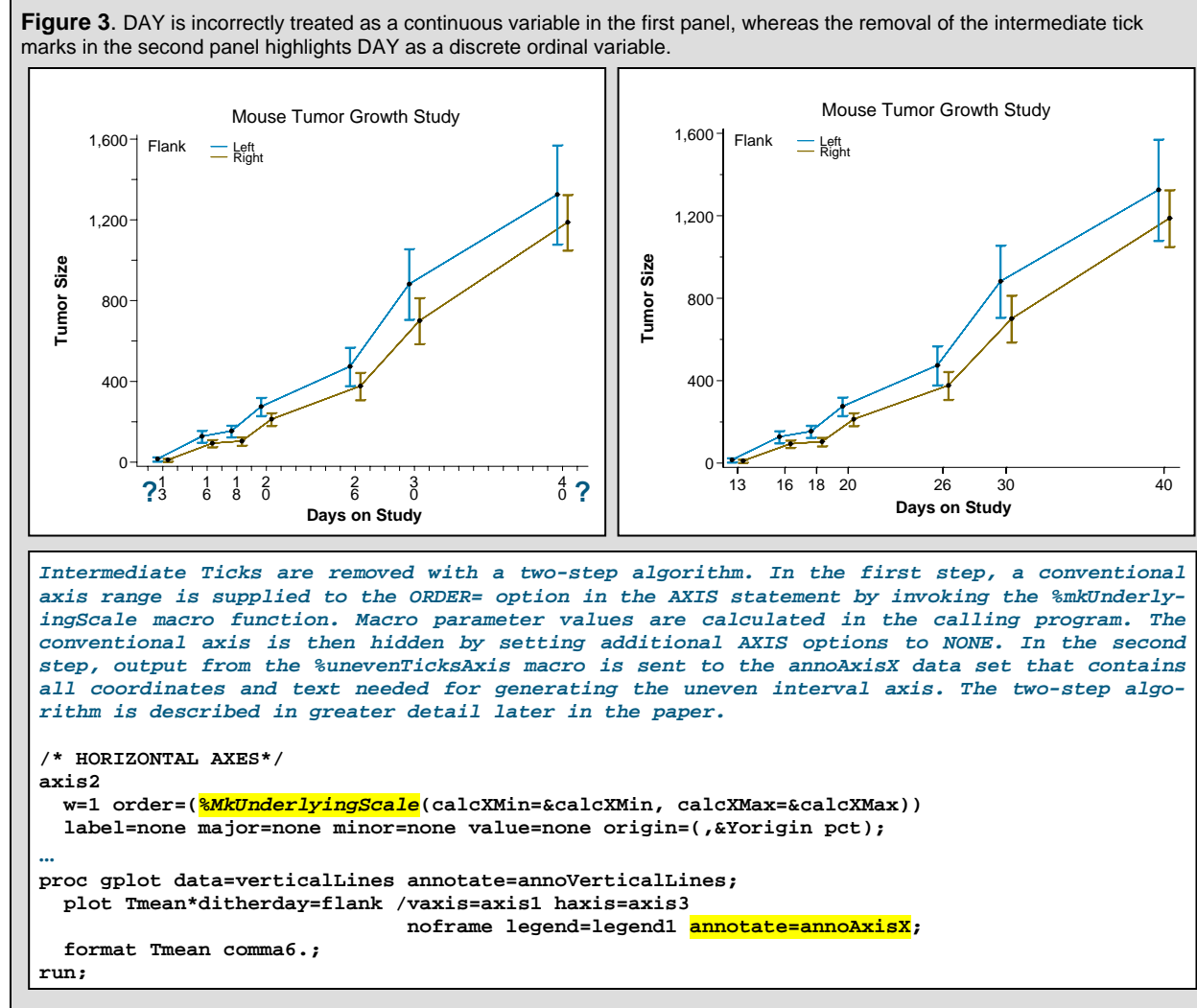
## REMOVING INTERMEDIATE TICKS FROM DITHERED PLOTS

Unequal intervals in the final plot in Figure 2 are highlighted by drawing intermediate ticks at unit distances. However, unit ticks don't work for pharmaceutical data that show how patients react to various therapies over time. Multiple therapies require dithering which messes up the time line. Programmers struggle to fit dithered data that are both discrete and ordinal into fixed continuous or nominal structures to produce a graph. However, both structures present problems that are only resolved with an axis where intermediate labels *and* associated ticks are removed.

### A CONTINUOUS STRUCTURE VS. DISCRETE ORDINAL DATA

Multiple-therapy data from the pharmaceutical industry are identical in composition to the mouse data shown in Figure 3. Right and left flanks are simply substituted for the multiple therapies. In the first panel, DAY is treated as if it were a continuous variable with the display of intermediate ticks along the horizontal axis. Unfortunately, ticks at Days 12 and 41 denoted by a question mark can't be found in the input data, and the dithering for increased visibility makes it look like left tumor sizes are calculated **before** right tumor sizes.

Lin solves the time problem by removing all ticks from the unevenly spaced axes displayed in *Tips and Tricks in Creating Graphs Using GPLOT* [1]. However, ticks add clarity to the graphics display of a discrete ordinal variable. They should only be removed when nominal data are plotted. SAS adheres to this convention by inserting ticks into histograms and removing them from bar charts. Both ticks and associated labels are displayed in the second graph in Figure 3 where DAY is correctly portrayed as a discrete ordinal variable. Note that the dithering is correct, since left and right flanks straddle the tick marks.



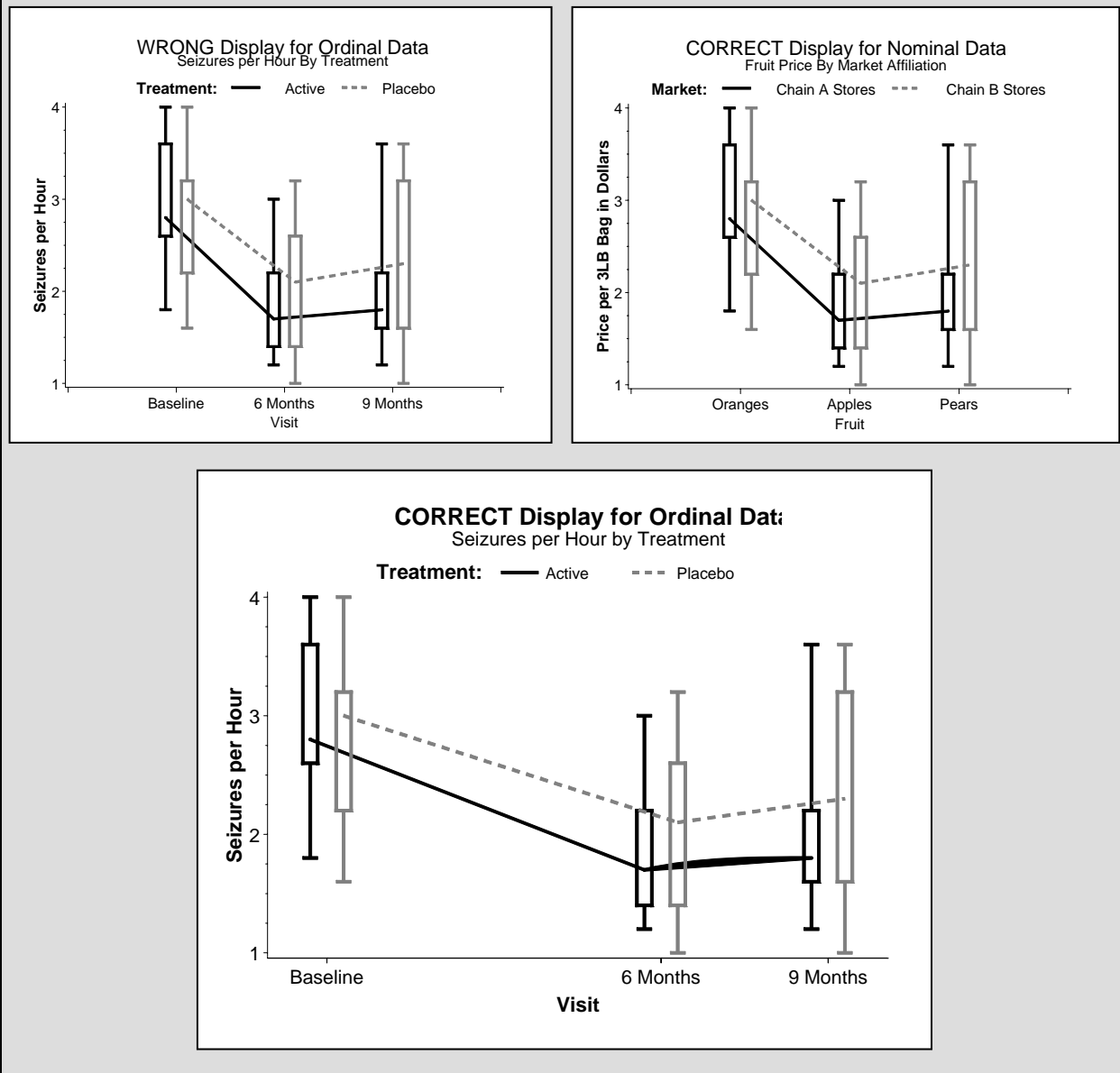
## A NOMINAL STRUCTURE VS. DISCRETE ORDINAL DATA

The seizure graph from *SAS® Programming in the Pharmaceutical Industry* is reproduced and adjusted in Figure 4 [3,218] to show what happens when the distance between displayed intervals does not match associated tick labels.

In the first panel, treatment visits are plotted at ticks 2, 3 and 4. Ticks 1 and 5, unobtrusively placed at the axis terminals, are added to accommodate the dithering that occurs between treatment groups. This graph is flawed, because *baseline* at the second tick should really be labeled *three months*. Slopes of the median connecting lines should be meaningful in displays of discrete ordinal data, but in this instance they have been distorted.

Evenly spaced intervals are well-suited for the display of discrete nominal data. In the second panel, VISITS have been replaced by FRUITS. Now the connecting lines are added simply for emphasis. In the third panel of Figure 4, the first graph is re-plotted correctly with uneven width intervals, correct slopes and no intermediate tick marks.

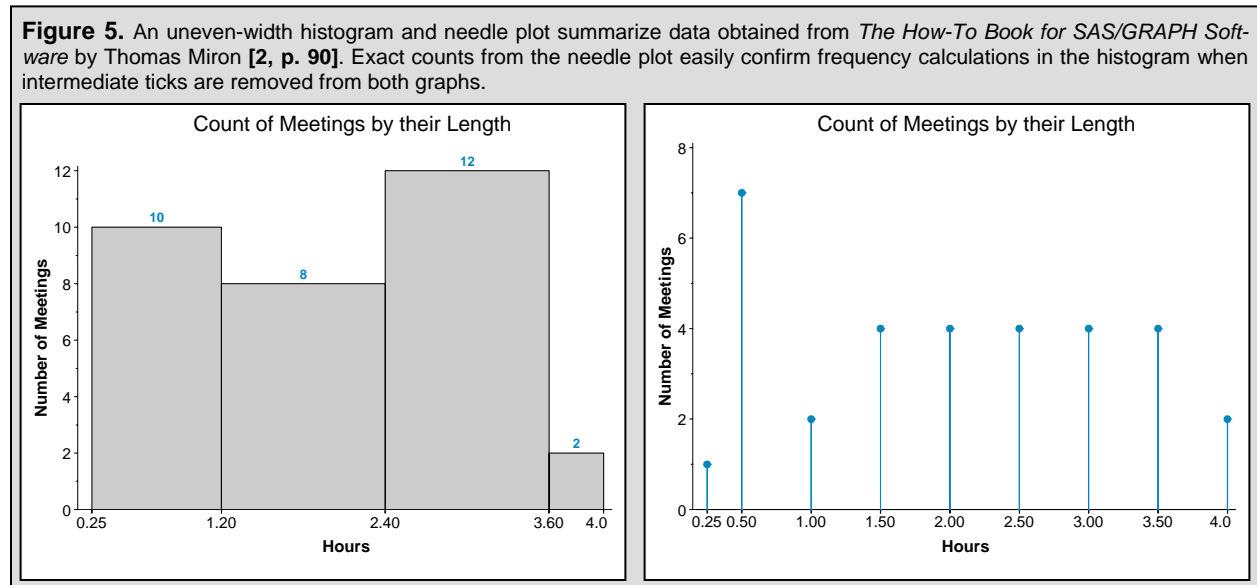
**Figure 4.** The first two graphs are identical. Only the titles and labels have been changed. For the first plot the drop rate in seizures per hour between baseline and 6 months is too sharp, whereas the same drop accurately shows that oranges are more expensive than apples in the second plot. The third plot accurately displays interval lengths in the seizure data.



## REMOVING INTERMEDIATE TICKS FROM NEEDLE PLOTS AND HISTOGRAMS

Besides dithered graphs, needle plots and histograms with uneven time intervals are enhanced with the removal of intermediate tick marks. Construction of the histogram in the first panel of Figure 5 is fully described in the paper, *Using SAS® Software to Generate Textbook-Style Histograms* [4], and the two-step algorithm described in the next section can be combined with the symbol statement below to produce the needle plot in the second panel:

```
symbol11 V=dot C=CX0386BE L=1 W=3 h=3pct i=needle;
```



## A TWO-STEP ALGORITHM FOR REMOVING INTERMEDIATE TICKS FROM A GRAPH

There is no way in SAS to directly remove selected *major* ticks from a graph. They can't even be assigned background colors to camouflage their existence. Therefore, a two-step algorithm has been designed to create the types of graphs pictured in Figures 3 - 5 above. First, a hidden conventional axis with an ORDER option is generated where all tick marks and labeling is turned off. Then the hidden axis is overlaid by the actual scale with coordinates derived from the input data via ANNOTATE.

### Step #1: Generate then Hide a Conventional Axis with %MkUnderlyingScale

The Axis2 and Axis3 statements below generate hidden axes that are revealed but grayed-out in Figure 6.

**axis2**

```
w=3 order=(%MkUnderlyingScale(calcXMin=&calcXMin, calcXMax=&calcXMax))
label=none major=none minor=none value=none origin=(,12pct);
```

- **%MkUnderlyingScale** is a macro function listed in the Appendix that returns an order statement in a range format with a BY clause. For the dithered mouse data, the range would be defined as 12.3 to 40.8 by 0.3
- **&calcXMin** **&calcXMax** represent the minimum and maximum dithered values in the input data. For the mouse data that would be 12.65 and 40.35 respectively. The values are altered twice in the macro after the BY value is calculated. For example, in the first minimum adjustment, the BY value (0.3) is subtracted from 12.65 leaving 12.35. Then the final minimum 12.3, a factor of 0.3, is returned from the nested **%getAxisMin** macro function. The maximum adjustment is made the same way with addition replacing subtraction and the macro **%getAxisMax** replacing **%getAxisMin**. The calculation for the BY value occurs in a third nested macro function, **%getIncr**.
- **label=none** **major=none** **minor=none** **value=none** erases the axis completely, leaving only a single horizontal line. Even though the axis is erased, the ORDER and ORIGIN options remain in effect. Otherwise the algorithm wouldn't work.
- **origin=(,12pct)** The YORIGIN in the hidden axis must match the parameter, YORIGIN, in the macro **%unevenTicksAxis** where the X-axis is redrawn via ANNOTATE..

**axis3**

```
w=3 order=(%MkUnderlyingScale(calcXMin=&calcXMin, calcXMax=&calcXMax, minOffset=1, maxOffset=4))
label=none major=none minor=none value=none origin=(,12pct);
```

- **minOffset=1**, **maxOffset=4** extends the axis range by 5 units or 1.5 days where a unit is defined as 0.3 in the **%getIncr** macro. The increase in range is needed for "Follow-up" in the second panel of Figure 6.

## Step #2: Generate an Uneven Interval Axis with %UnevenTicksAxis

The output from an execution of the %unevenTicksAxis macro is an ANNOTATE data set, ANNOAXISX. The macro accommodates summary or raw data in INDS, since PROC SQL is used to create an intermediate data set, DISTINCTXTICK with a *select distinct* on the XVAR variable. The portion of the macro that deals with the generation of the ANNOTATE data set is listed below. A full listing is available in the Appendix.

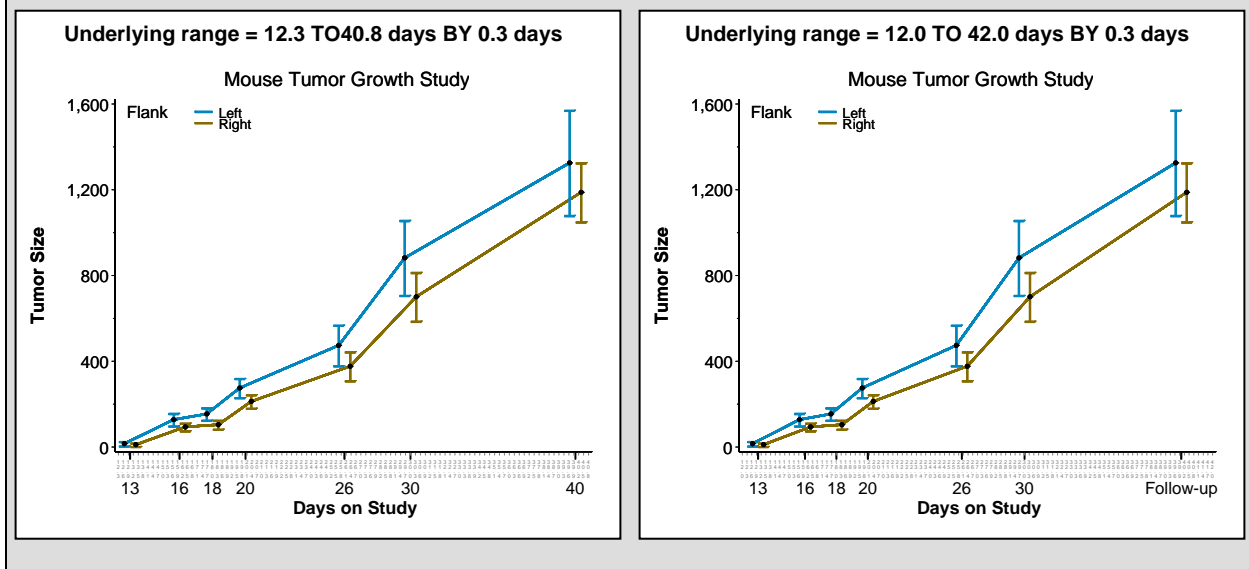
```
/* -----
Program : UnevenTicksAxis.sas
Parms   : Name           Description
-----
          INDS           Input data set name
          XVAR           Horizontal axis variable name
          PCTSIZE        Text Size in PCT
          XLABEL         X-axis label
          YORIGIN        Vertical origin in percent
          XVALFMT        Format used to display tick
                        marks along the X-Axis.
          TICKLENGTH     Length of Ticks (default=1.5 pct)
          TICKWIDTH      Width of Ticks (default= 0.2 pct)
          FONT           e.g. Hwcm001 (default), 'Arial'
          FONTBOLD       e.g. Hwcm002, (default) 'Arial/Bold'
Note    : GOPTIONS GUNIT must be set to PCT in the calling program.
Input   : &inDS
Output  : work.annoAxisX
----- */

%macro UnevenTicksAxis(inDS=, xvar=, pctSize=, xlabel=, yOrigin=, XvalFmt=,
                      tickLength=1.5, tickWidth=0.2, font=HWcm001, fontBold=HWcm002);
%annomac;
%local textSeparation labelYcoord;
%let textSeparation=%sysevalf(&tickLength * 2);
%let labelYcoord=%sysevalf(&textSeparation * 2);

< Create data set DISTINCTXTICK with SQL here. See full listing in the Appendix. >
data annoAxisX;
  retain xsys '2' hsys '3';
  length text $30 function color $8 style $15;
  set distinctXtick end=last;
  function='move'; ysys='1'; x=xtick; y=0; output;
  function='draw'; ysys='7'; y=-&tickLength; color='black'; line=1; size=&tickWidth;
  output;
  function='move'; y=-&textSeparation; output;
  function='cntl2txt'; output;
  function='label'; call missing(x,y); position='5'; text = displayX;
  size=&pctSize; style="&font"; output;
  if last then do;
    function='move'; xsys='1'; x=50; y=-&labelYcoord; output;
    function='cntl2txt'; output;
    function='label'; call missing(x,y); position='5'; text = "&xLabel";
    size=&pctSize; style="&fontBold"; output;
  end;
run;
%mend UnevenTicksAxis;
```

- `retain xsys '2'` interprets coordinates as absolute values from the data area, whereas `ysys='1'` and `ysys='7'` interprets coordinates as absolute or relative percentages of the data area, respectively.
- `y= -&textSeparation; ... function=cntl2txt; ... function='label'; call missing(x,y)...` makes it possible to assign relative percentages as label coordinates in an annotate data set. The solution comes from code written by Robert Allison and included as an attachment to an email message sent to the author by Mike Zdeb on September 1, 2008. See also [5, 617-619].

**Figure 6.** The display axis from `%UnevenTicksAxis` is juxtaposed over a grayed-out (usually invisible) axis where the `ORDER` option is filled in with an invocation of the `%MkUnderlyingScale` macro function. The underlying axis is extended in the second graph to accommodate the text "Follow-up".



## SUMMARY AND CONCLUSIONS

Techniques have been described for reformatting the horizontal axis so that it better accommodates discrete ordinal data. Uneven interval graphs with visible intermediate major ticks can be easily graphed with a judicious use of a format. However, if dithering is employed in the graph to offset data classifications then intermediate major tick marks must be removed with an application of the two-step process described in this paper.

## ACKNOWLEDGEMENTS

The author thanks Mike Zdeb for his review of the paper. He detected a significant error in notation for Figure 2 and provided the reference to Robert Allison's use of the `%cnt12txt` ANNOTATE macro that reduces the number of parameters needed for the `%UnevenTicksAxis` macro.

## COPYRIGHT STATEMENT

The paper, *Generate a Customized Axis Scale with Uneven Intervals in SAS® Automatically*, is protected by copyright law. This means if you would like to use part or all of the original ideas or text from these documents in a publication where no monetary profit is to be gained, you are welcome to do so. All you need to do is to cite the paper in your reference section. For ALL uses that result in corporate or individual profit, written permission must be obtained from the author. Conditions for usage have been modified from <http://www.whatiscopyright.org>.

## REFERENCES

- [1] Lin, Qin. *Tips and Tricks in Creating Graphs Using PROC Gplot*. Proceedings of the 20<sup>th</sup> Annual Northeast SAS Users Group Conference. Baltimore, MD, 2007, paper #CC25.
- [2] Miron, Thomas. *The How-To Book for SAS/GRAPH Software*. Cary, NC: SAS Institute Inc., 1995. Copyright 1995, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Data used with permission of SAS Institute Inc., Cary, NC.
- [3] Shostak, Jack. *SAS® Programming in the Pharmaceutical Industry*. Cary, NC: SAS Institute Inc., 2005. Copyright 2005, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC.
- [4] Watts, Perry. *Using SAS® Software to Generate Textbook-Style Histograms*. Proceedings of the SAS® Global Forum 2009 Conference. Washington, DC, 2008, paper #216-2009.

## SAS Institute Reference:

- [5] SAS Institute Inc. *SAS/GRAPH® 9.1 Reference, Volumes 1, 2, and 3*, Cary NC: SAS Institute Inc., 2004.

## **TRADEMARK CITATION**

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.

## **CONTACT INFORMATION**

The author welcomes feedback via email at [perryWatts@comcast.net](mailto:perryWatts@comcast.net). A text version of the source code is available upon request.

## APPENDIX: THE TWO-STEP UNEVEN AXIS MACROS AND CALLING PROGRAM

### Macros

```

/* -----
Program   :   mkUnderlyingScale.sas

Purpose   :   Macro function that creates an order statement in a hidden
                axis where the distance between major tick marks is very small.

Output    :   An explicit range in &XMIN to &XMAX by &Xby format.

Parms     :   NAME                DESCRIPTION                DEFAULT
                -----                -----                -----
                CALCXMIN           XMIN calculated from input data
                                in the calling program           none (required)
                CALCXMAX           XMAX calculated from input data
                                in the calling program           none (required)
                MINOFFSET          IF GT 0 then adjust CALCXMIN by
                                -&MINOFFSET * &XBY.                blank (not required)
                MAXOFFSET          IF GT 0 then adjust CALCXMAX by
                                &MAXOFFSET * &XBY.                blank (not required)

Macros
Called    :   GETINCR              Calculates the narrow increment used in the
                                order statement and used to stretch the
                                axis when desired.
                GETAXISMIN         Re-calculates Axis minimum with output from
                                GETINCR.
                GETAXISMAX         Re-calculates Axis maximum with output from
                                GETINCR.
----- */

%macro MkUnderlyingScale(calcXMin=, calcXMax=, minoffset=, maxoffset=);
  %local rangeX xBy xmin xmax;
  %let rangeX = %sysevalf(&calcXMax - &calcXmin);
  %let xBy= %getIncr(range=&rangeX);
  %let calcXmin = %sysevalf(&calcXmin - &xBy);
  %let calcXmax = %sysevalf(&calcXmax + &xBy);
  %if &minOffset gt 0 %then
    %let calcXmin=%sysevalf(&calcXmin - (&minOffset * &Xby) );
  %if &maxOffset gt 0 %then
    %let calcXmax=%sysevalf(&calcXmax + (&maxOffset * &Xby) );
  %let xmin=%getAxisMin(min=&calcXmin, by=&Xby);
  %let xmax=%getAxisMax(max=&calcXmax, by=&Xby);
  &XMIN to &XMAX by &Xby
%mend MkUnderlyingScale;

```

```

/* -----
Program   :   getIncr.sas

Purpose   :   Obtain small increment (BY-Value) for an underlying hidden
                axis from the RANGE parameter. The return value is used
                to stretch out the axis minimum and maximum so that value
                labels (created in annotate) are fully visible.
                The return value also becomes the factor if relative units are
                specified for the XMIN and XMAX parameters in MKUNDERLYINGSCALE.
                For example, XMIN=-3 tells MKUNDERLYINGSCALE to decrease the
                horizontal axis by 3*factor.

Parms     :   Name                Description                Default
                -----                -----                -----
                RANGE              Data MAX - Data MIN (where MAX
                                and MIN are internally defined
                                with any degree of precision)       none (required)

Usage     :   %let RangeX = %sysevalf(&calcXMax - &calcXmin);
                %let _Xby= %getIncr(range=&RangeX);

```

```

Notes      : From the manual: ROUND(argument, round-off-unit)
            The ROUND function returns a value rounded to the nearest
            round-off unit. So for example:
            Range=40.35(days) - 12.65(days) = 27.7
            %let x=%getincr(range=27.7);
            LOOP1 NOTFOUND=1: round range and result = 1 0.277 0
            LOOP2 NOTFOUND=0: round range and result = 0.1 0.277 0.3
            %put x=&x;
            x=0.3

```

```

----- */
%macro getINCR(Range=);
  %local notfound found result round;
  %let range=%sysevalf(&range/100);
  %let notFound = 1;
  %let round = 1;
  %do %while (&notFound);
    %let result=%sysfunc(round(&range, &round));
    %if &result gt 0 %then %let notFound=0;
    %else %do;
      %let round = %sysevalf(&round / 10.0);
    %end;
  %end;
  %result
%mend getIncr;

```

```

/* -----
Program   : getAxisMax.sas

Purpose   : Obtain maximum range value for an axis where the BY value
            can be either integer or real.

Parms     : Name           Description           Default
            -----
            Max           Maximum value derived from data   none (required)
            By           By value entered by user           none (required)

Examples  : %let XMax=%getAxisMax(Max=0.9321, by=0.02); XMax = 0.94
            %let XMax=%getAxisMax(Max=-13, by=2);       XMax = -12
----- */

```

```

%macro getAxisMax(Max=, by=);
  %local adjMax intBy intMax modX multiplier notFound;
  %let notFound = 1;
  %let multiplier = 1;
  %do %while (&notFound);
    %let intMax=%sysevalf(&multiplier * &Max);
    %let intBy=%sysevalf(&multiplier * &by);
    %if %sysfunc(floor(&intBy)) ne %sysfunc(ceil(&intBy)) %then %do;
      %let multiplier = %sysevalf(&multiplier * 10.0);
    %end;
    %else %do;
      %let notFound = 0;
    %end;
  %end;
  %let intMax=%sysfunc(ceil(&intMax));
  %let modX = %sysfunc(mod(&intMax,&intBy));
  %do %while (%eval(&modX) ne 0);
    %let intMax=%eval(&intMax +1);
    %let modX = %sysfunc(mod(&intMax,&intBy));
  %end;
  %let adjMax=%sysevalf(&intMax/&multiplier);
  %adjMax
%mend getAxisMax;

```

```

/* -----
Program   : getAxisMin.sas

Purpose   : Obtain minimum range value for an axis where the BY value
            can be either integer or real.

Parms     : Name           Description           Default
            -----
            Min           minimum value derived from data   none
            By           By value entered by user           none

```

```

examples : %let XMin=%getAxisMin(Min=0.9321, by=0.02); Xmin = 0.92
           %let Xmin=%getAxisMin(Min=-13, by=2);      Xmin = -14
           %let Xmin=%getAxisMin(Min=14.5, by=15);    Xmin = 0
           %let Xmin=%getAxisMin(Min=15.1, by=15);    Xmin = 15
           %let XMin=%getAxisMin(Min=9.321, by=0.5);  Xmin = 9
----- */
%macro getAxisMin(Min=, by=);
%local adjMin intBy intMin modX multiplier notFound ;
%let notFound = 1;
%let multiplier = 1;
%do %while (&notFound);
%let intMin=%sysevalf(&multiplier * &Min);
%let intBy=%sysevalf(&multiplier * &by);
%if %sysfunc(floor(&intBy)) ne %sysfunc(ceil(&intBy)) %then %do;
%let multiplier = %sysevalf(&multiplier * 10.0);
%end;
%else %do;
%let notFound = 0;
%end;
%end;
%let intMin=%sysfunc(floor(&intMin));
%let modX = %sysfunc(mod(&intMin,&intBy));
%do %while (%eval(&modX) ne 0);
%let intMin=%eval(&intMin -1);
%let modX = %sysfunc(mod(&intMin,&intBy));
%end;
%let adjMin=%sysevalf(&intMin/&multiplier);
&adjmin
%mend getAxisMin;
-----

/* -----
Program : UnevenTicksAxis.sas

Purpose : Create an uneven X axis where ticks coincide with a finite
number of X-coordinates in the data. Ticks are labeled with
values for the X-coordinates.

Parms : Name Description Default
-----
INDS Input data set name none (required)
XVAR Horizontal axis variable name none (required)
PCTSIZE Text Size in PCT none (required)
XLABEL X-axis label none (required)
YORIGIN Vertical origin in percent none (required)
XVALFMT Format used to display tick none (not required)
marks along the X-Axis.
TICKLENGTH Length of Ticks 1.5
TICKWIDTH Width of Ticks 0.2
FONT Hwcmg001 or 'Arial' Hwcmg001
FONTBOLD Hwcmg002 or 'Arial/Bold' Hwcmg002

Note : GOPTIONS GUNIT must be set to PCT in the calling program.
Input : &inDS
Output : work.annoAxisX
----- */

%macro UnevenTicksAxis(inDS=, xvar=, pctSize=, xlabel=, yOrigin=, XvalFmt=,
tickLength=1.5, tickWidth=0.2, font=Hwcmg001, fontBold=Hwcmg002);
%annomac;
%local textSeparation labelYcoord;
%let textSeparation=%sysevalf(&tickLength * 2);
%let labelYcoord=%sysevalf(&textSeparation * 2);

proc sql noprint;
create table distinctXtick as
select distinct &xvar as xtick from &inDS
order by &xvar;
quit;
data distinctXtick;
length DisplayX $15;
set distinctXtick;
%if &Xvalfmt eq %then %do;
fmtYvN=0;
DisplayX=left(put(xtick,best.));
%end;

```

```

%else %do;
  fmtYvN=1;
  displayX=left(trim(put(xtick, &xValFmt.)));
%end;
run;
data annoAxisX;
  retain xsys '2' hsys '3';
  length text $30 function color $8 style $15;
  set distinctXtick end=last;
  function='move'; ysys='1'; x=xtick; y=0; output;
  function='draw'; ysys='7'; y=-&tickLength; color='black'; line=1; size=&tickWidth; output;
  function='move'; y=-&textSeparation; output;
  function='cntl2txt'; output;
  function='label'; call missing(x,y); position='5'; text = displayX;
  size=&pctSize; style="&font"; output;
  if last then do;
    function='move'; xsys='1'; x=50; y=-&LabelYcoord; output;
    function='cntl2txt'; output;
    function='label'; call missing(x,y); position='5'; text = "&xLabel";
    size=&pctSize; style="&fontBold"; output;
  end;
run;
%mend UnevenTicksAxis;

```

---

## Calling Program

```

/* -----
Program   :   UnevenTicksAxDemo.sas

Purpose   :   Create a dithered line plot with an uneven axis where unlabeled
              ticks at regular intervals have been removed.

Input     :   psd.MouseUnevenInterval, available upon request.
Output    :   demo1.cgm and demo2.cgm
----- */
options nodate number pageno=1 fmtsearch=(work) maautosource mprint nosymbolgen;
options sasautos=(sasautos 'c:\SGF09\CustomAxis\Mac');
libname psd 'c:\SGF09\CustomAxis\Data';
title1 move=(+10pct,+0pct) 'Mouse Tumor Growth Study';
%let Yorigin=15;
%let PctSize=3.25;
%let tickLength=1.5;
%let tickWidth=0.3;

/*****
SUMMARY DATA ARE USED TO PRODUCE THE VERTICAL SEGMENTS OF THE LINE PLOT, BECAUSE THE
CGM GRAPHICS DEVICE DOES NOT COORDINATE LINE AND 'TOP' WIDTHS. (TOPS ARE MUCH TOO WIDE).
*****/
proc summary data=psd.MouseUnevenInterval NOPRINT;
  by day;
  var LeftT RightT;
  output out=VerticalLines MEAN=LTMean RTMean STDERR=LTSTD RTSTD;
run;

data VerticalLines(keep=day ditherDay flank TstdMinus Tmean TstdPlus);
  set VerticalLines;
  length flank $5;
  flank='Left'; Tmean=LTMean; TstdPlus=LTmean+LTstd; TstdMinus=LTmean-LTstd;
  ditherday=day-0.35; /*WHERE DITHERING OCCURS */
  output;
  flank='Right'; Tmean=RTMean; TstdPlus=RTmean+RTstd; TstdMinus=RTmean-RTstd;
  ditherday=day+0.35; /*WHERE DITHERING OCCURS */
  output;
run;

data annoVerticalLines;
  length style function $8;
  retain style 'none' line 1;
  retain when 'a';
  retain tickwidth 2.25 linewidth 0.3;
  set VerticalLines;
  hsys='3'; xsys='2';
  if flank eq 'Left' then color='CX0386BE'; else color='CX866C00'; /* NESUG 08 COLORS */
  ysys='2'; function='move'; y=TstdPlus; x=ditherDay; output;

```

```

    ysys='9'; function='draw'; y=-0.3; size=tickwidth; output;
    ysys='2'; function='draw'; y=TstdMinus; size=linewidth; output;
    ysys='9'; function='draw'; y=-0.3; size=tickwidth; output;
    color='black';
    ysys='2'; function='symbol'; text='dot'; size=1.8; y=Tmean; output;
run;

/* NEED CALCXMIN AND CALCXMAX MACRO VARIABLES FOR MACRO MKUNDERLYINGSSCALE INVOKED LATER ON */
proc sql noprint;
    select min(ditherday), max(ditherday) into :calcXmin, :calcXmax
    from verticalLines;
quit;

/* GRAPHICS OUTPUT */
goptions reset=goptions;
goptions device=cgmOf97L gunit=pct rotate=landscape ftext=HWCGM001
htext=3 htitle=3.5 display;

/* VERTICAL AXIS */
axis1 w=3 label=(f=HWCGM002 a=90 "Tumor Size")
    order=(0 to 1600 BY 400) major=(h=1.2) minor=(h=0.6 n=1);

/* HORIZONTAL AXES*/
axis2
    w=1 order=(%MkUnderlyingScale(calcXMin=&calcXMin, calcXMax=&calcXMax))
    label=none major=none minor=none value=none origin=(,&Yorigin pct);
/* MAKE ROOM FOR "FOLLOW-UP" WITH AN ADJUSTMENT TO %MkUnderlyingScale */
axis3
w=1 order=(%MkUnderlyingScale(calcXMin=&calcXMin, calcXMax=&calcXMax, minOffset=1, maxOffset=6))
    label=none major=none minor=none value=none origin=(,&Yorigin pct);

/* FOR HORIZONTAL DATA LINES ONLY */
symbol1 interpol=j l=1 w=3 c=CX0386BE value=none;
symbol2 interpol=j l=1 w=3 c=CX866C00 value=none;

/* LEGEND FOR THE DITHERED DATA */
legend1 noframe down=2
    position=(LEFT INSIDE TOP) offset=(+2pct,)
    label=(f=HWCGM001 "Flank")
    shape=line(3pct)
    value=(t=1 j=1 h=2.5 "Left"
    t=2 j=1 h=2.5 "Right");

/*****
CREATE ANNOTATE DATASET, WORK.ANNOAXIX INSIDE THE UNEVENINTERVALAXIS MACRO. &INDS STORES
COORDINATES AND LABELS FOR THE DISPLAYED, UNEVEN INTERVAL AXIS. THE UNDITHERED VARIABLE
'DAY' IS ASSIGNED TO THE PARAMETER, XVAR.
*****/
*****/
%UnevenTicksAxis(inDS=verticalLines, xvar=day, pctSize=&pctSize,
    tickwidth=&tickWidth, tickLength=&tickLength,
    xlabel=Days On Study, yOrigin=&Yorigin)
filename demo1 'C:\SGF09\CustomAxis\grf\demo1.cgm';
goptions gsfname=demo1;
proc gplot data=verticalLines annotate=annoVerticalLines;
    plot Tmean*ditherday=flank /vaxis=axis1 haxis=axis2 noframe
    legend=legend1 annotate=annoAxisX;
    format Tmean comma6.;
run;
quit;

/* FORMAT THE AXIS VALUES WITH XVALFMT FILLED IN ON %UNEVENINTERVAL*/
proc format;
    value xaxisFM 13,16,18,20,26,30=[2.] 40='Follow-up';
run;
%UnevenTicksAxis(inDS=verticalLines, xvar=day, pctSize=&pctSize, xvalfmt=xaxisFM.,
    tickwidth=&tickWidth, tickLength=&tickLength,
    xlabel=Days On Study, yOrigin=&Yorigin)
filename demo2 'C:\SGF09\CustomAxis\grf\demo2.cgm';
goptions gsfname=demo2;
proc gplot data=verticalLines annotate=annoVerticalLines;
    plot Tmean*ditherday=flank /vaxis=axis1 haxis=axis3 noframe
    legend=legend1 annotate=annoAxisX;
    format Tmean comma6.;
run;
quit;

```