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.

A calling program, **UnevenAxDemo.sas**, is included in the paper. Also included are relevant graphics examples such as needle plots, box plots, uneven width histograms, and enhanced line plots that accommodate several class values. Associated macros and input data are stored separately in the NESUG proceedings.

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].**

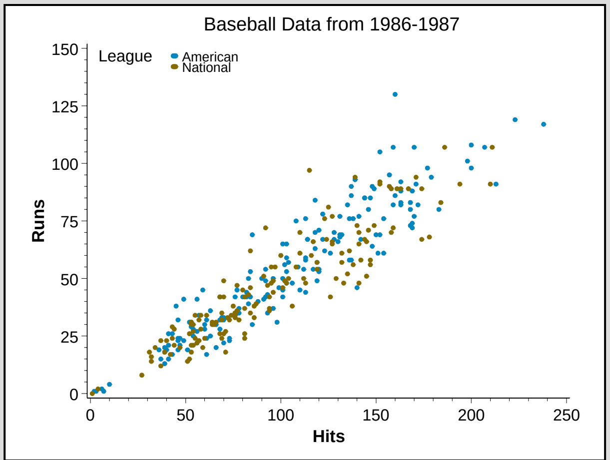
Figure 1. The axis ORDER= option is ideally suited for the plots below. A single minor tick mark references intermediate days for the mouse study.

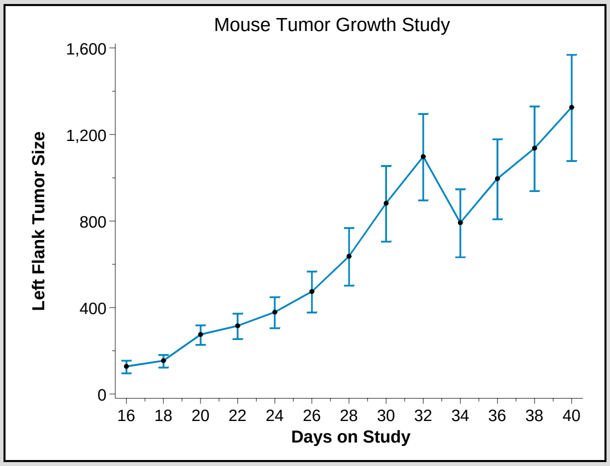
Baseball Data Axis Statement

```
axis2 w=3 order=(0 to 250 by 50)
  label=(f=HWCGM002 "Hits") minor=(n=4);
```

Mouse Data Axis Statement

```
axis2 w=3 order=(16 to 40 by 2) offset=(2,2)
  label=(f=HWCGM002 "Days on Study")
  minor=(n=1);
```





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**.

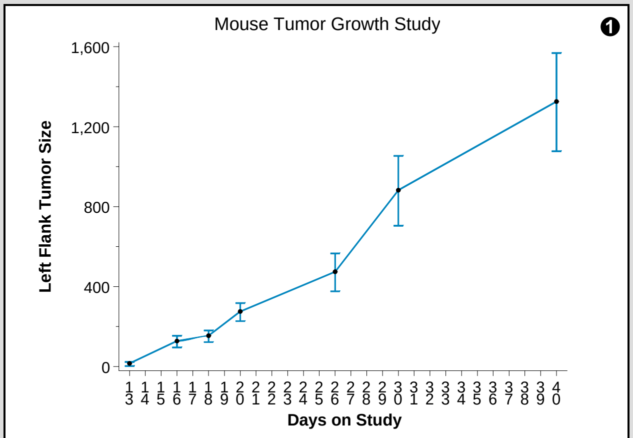
Even Unit Intervals

```
axis2 w=3 order=(13 to 40 by 1)
  offset=(2pct, 2pct)
  label=(f=HWCGM002 "Days on Study")
  major=(h=.8) minor =(n=1);
```

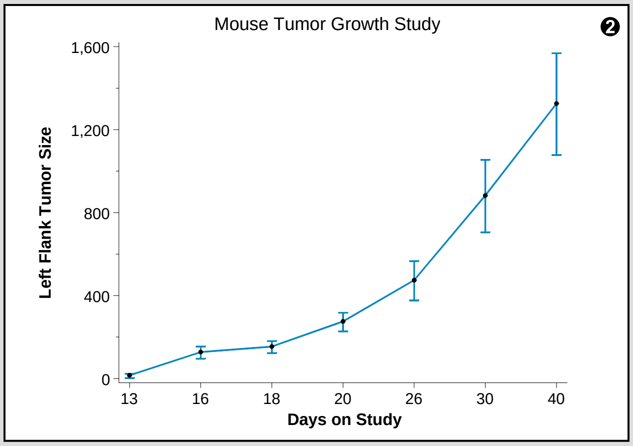
Restricted Order Statement

```
axis3 w=3 order=(13,16,18,20,26,30,40)
  offset=(2pct, 2pct)
  label=(f=HWCGM002 "Days on Study")
  major=(h=.8) minor = none;
```

❶



❷



❸

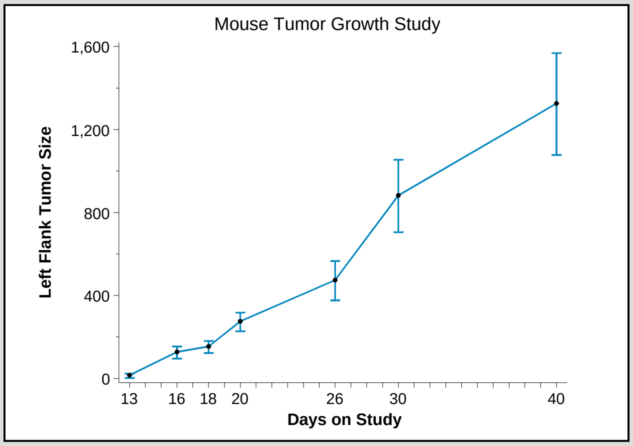
Do NOT use the tick clause associated with the VALUE= option when values for the variable being plotted are coincident with major tick marks. Too much hard-coding is required, and the visible tick labels will go vertical. Use a format instead.

```
axis6
 w=3 order=(13 to 40 by 1) offset=(2pct, 2pct)
 label=(f=HWCGM002 "Days on Study")
 major=(h=.8) minor = none
 value=( h=3
   tick=1 justify=c  '13'   tick=2 justify=c  ' '
   tick=3 justify=c  ' '    tick=4 justify=c  '16'
   tick=5 justify=c  ' '    tick=6 justify=c  '18'
   tick=7 justify=c  ' '    tick=8 justify=c  '20'
   tick=9 justify=c  ' '    tick=10 justify=c ' '
   tick=11 justify=c ' '    tick=12 justify=c ' '
   tick=13 justify=c ' '    tick=14 justify=c '26'
   tick=15 justify=c ' '    tick=16 justify=c ' '
   tick=17 justify=c ' '    tick=18 justify=c '30'
   tick=19 justify=c ' '    tick=20 justify=c ' '
   tick=21 justify=c ' '    tick=22 justify=c ' '
   tick=23 justify=c ' '    tick=24 justify=c ' '
   tick=25 justify=c ' '    tick=26 justify=c ' '
   tick=27 justify=c ' '    tick=28 justify=c '40');
```

❹

Problem Solved with a format

```
proc format;
 value vfmt 13,16,18,20,26,30,40=[2.] other=' ';
run;
axis2 w=3 order=(13 to 40 by 1)
  offset=(2pct, 2pct)
  label=(f=HWCGM002 "Days on Study")
  major=(h=.8) minor =(n=1);
proc gplot data=unevenMeans annotate=annoUnEven;
 plot meanLT*day=1 /vaxis=axis1 haxis=axis2
                   noframe;
 format day vfmt. meanLT comma6.;
run;
```



REMOVING INTERMEDIATE TICKS FROM DITHERED PLOTS FOR DISCRETE ORDINAL DATA

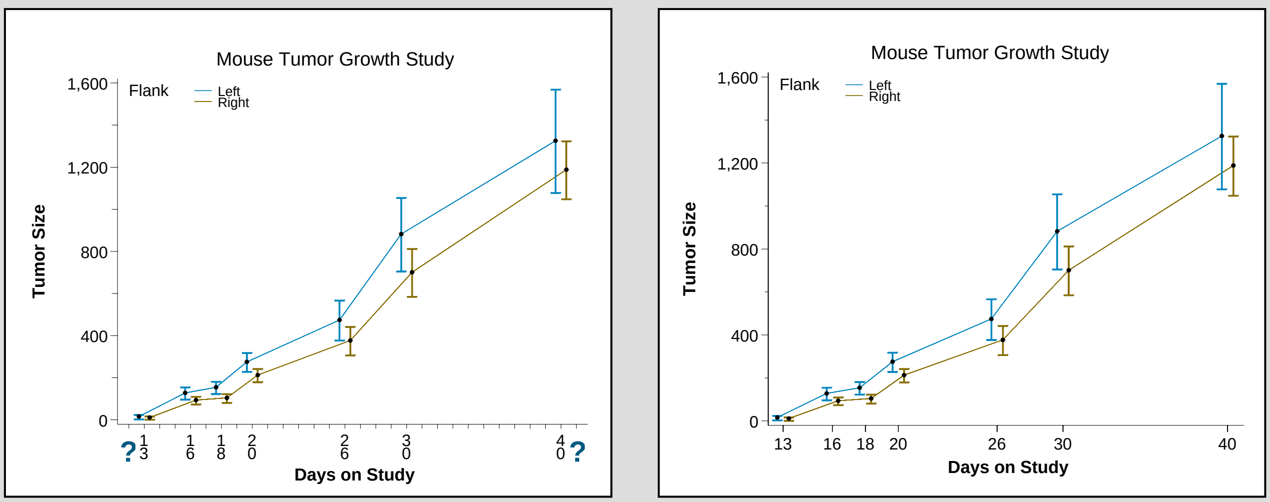
Unequal intervals in the final plot in Figure 2 are highlighted by drawing intermediate ticks at unit distances. However, unit ticks don't work for pharmaceutical data that show how patients react to various therapies over time. Multiple therapies require dithering which messes up the time line. Programmers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

Continuous vs. Discrete Ordinal Data

Multiple-therapy data from the pharmaceutical industry are identical in structure 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 looks 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.

Figure 3. DAY is incorrectly treated as a continuous variable in the first panel, whereas the removal of the intermediate tick marks in the second panel highlights DAY as a discrete ordinal variable.



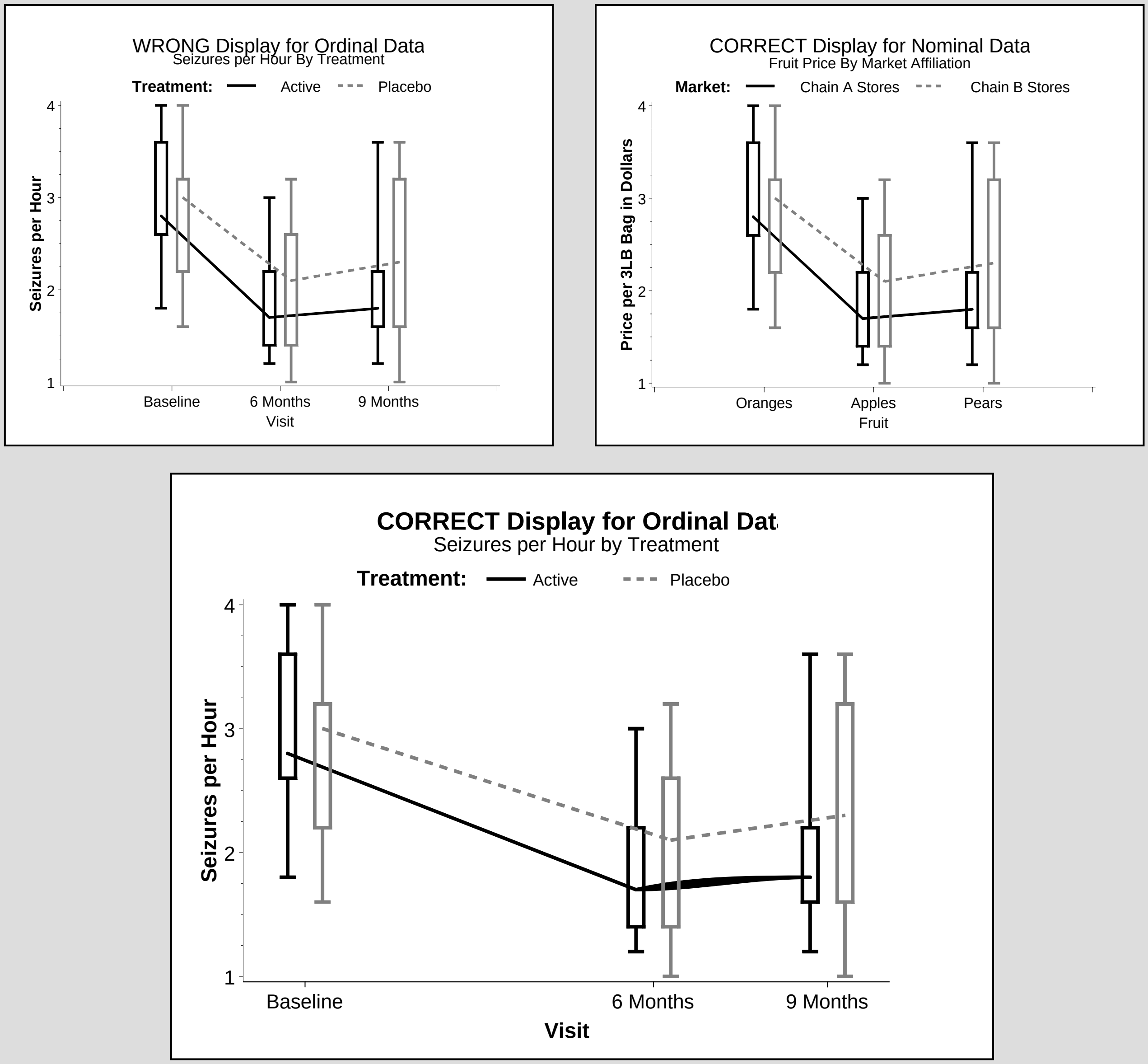
Discrete Nominal vs. Discrete Ordinal Data

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.

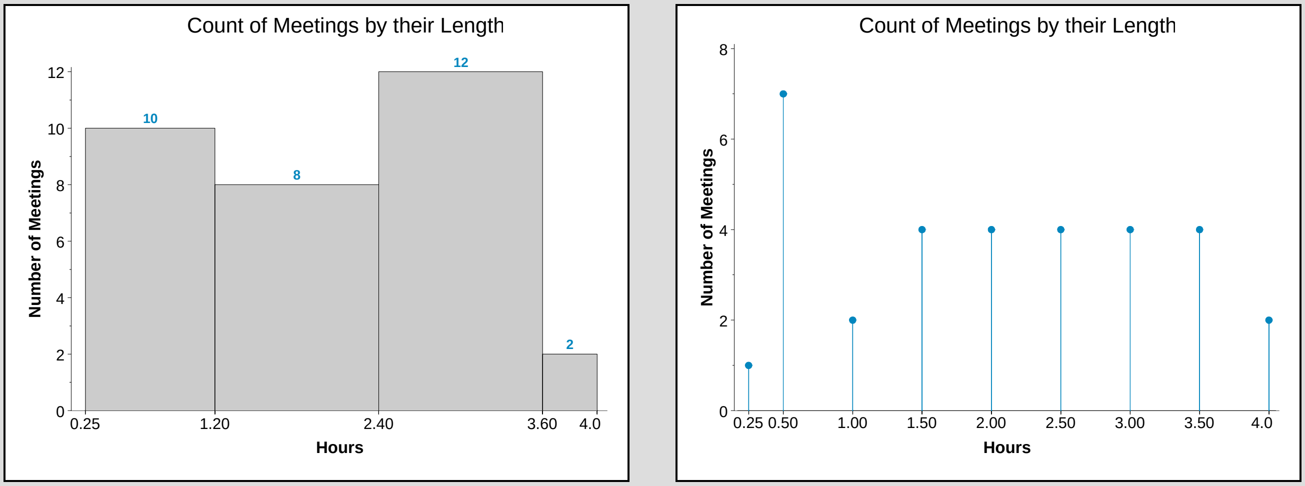


REMOVING INTERMEDIATE TICKS FROM NEEDLE PLOTS AND HISTOGRAMS

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-pass algorithm described in the next section can be combined with the symbol statement below to produce the needle plot in the second panel:

```
symbol1 V=dot C=CX0386BE L=1 W=3 h=3pct i=needle;
```

Figure 5. An uneven-width histogram and needle plot summarize data obtained from *The How-To Book for SAS/GRAPH Software* by Thomas Miron **[2, p. 90]**. Exact counts from the needle plot easily confirm frequency calculations in the histogram when intermediate ticks are removed from both graphs.



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 are taken from **`UnevenAxDemo.sas`** and 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 proceedings 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 ***`%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 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 ***`%unevenIntervalAxis`*** 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 the value label "Follow-up" in the second panel.

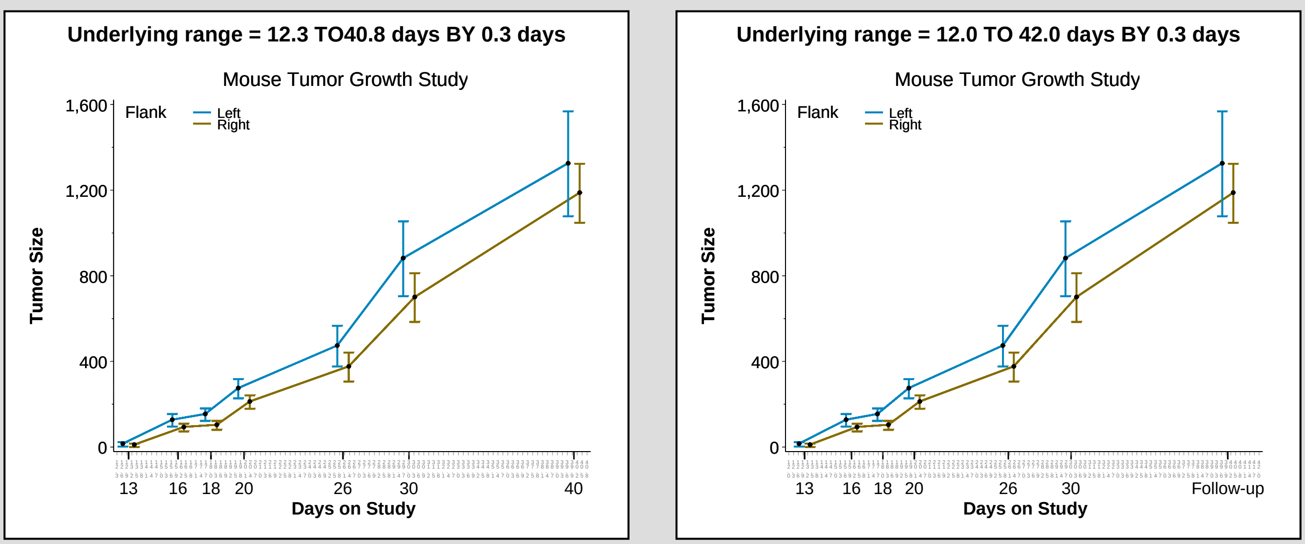
Step #2: Generate an Uneven Interval Axis with `%UnevenIntervalAxis`

The output from an execution of the ***%unevenIntervalAxis*** 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 can be found in the NESUG proceedings.

```
%macro UnevenIntervalAxis(inDS=, xvar=, pctSize=, xlabel=, yOrigin=, XvalFmt=);
  %annomac;
  %local tickLength labelYpos axisLabelPos;
  %let tickLength=1;    /* Ticks are 1 percent in length. They start at YORIGIN */
  %let labelYpos=3.5;  /* Tick values are listed 3.5% below YORIGIN.           */
  %let axisLabelPos=9; /* The axis label is 9% below YORIGIN.                   */
  < Create data set DISTINCTXTICK with SQL here. See full listing in the Appendix. >
  data annoAxisX;
    %dclanno;
    length text $30;    /* %dclanno takes care of the rest */
    set distinctXtick end=last;
      %system(2,3,3);
      %move(xtick, &yOrigin);
      %draw(xtick, &yOrigin - &tickLength., black, 1, 0.04);
      %label(xtick, &yOrigin - &LabelYpos., DisplayX, black, 0, 0, &pctSize, Hwcgm001,5);
      if last then do;
       %system(1,3,3);
       %label(50, &yOrigin - &axisLabelPos., "&xLabel", black, 0, 0, &pctSize, Hwcgm002,5);
      end;
  run;
%mend UnevenIntervalAxis;
```

- `inDS=`, `xvar=`, `pctSize=`, `xlabel=`, `yOrigin=`, `XvalFmt=` See source code for parameter definitions.
- highlights hard-coded values that could be converted to parameters that would extend the macro.
- ***`%system(2,3,3)`*** is a built-in ANNOTATE macro that translates positional parameters to XYS, YSYS and HSYS coordinate systems. For axis ticks and value labels an XSYS value of '2' uses absolute values from the data area whereas a value of '3' for YSYS and HSYS translates assigned numbers to percentages of the graphics output area. With YSYS set to '3', TICKLENGTH and LABELYPOS can be accurately subtracted from YORIGIN which is also defined as a percent.
- ***`%system(1,3,3)`*** XSYS, here is changed from '2' to '1' (percent of data area) so that the axis label is centered on the horizontal axis when the corresponding X coordinate is set to 50.

Figure 6. The display axis from ***%UnevenIntervalAxis*** 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.

COPYRIGHT STATEMENT

REFERENCES

[1] Lin, Qin. *Tips and Tricks in Creating Graphs Using PROC GPLOT*. Proceedings of the 20th 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 21st Annual Northeast SAS Users Group Conference. Pittsburgh, PA 2008, paper #NP??.

SAS Institute Reference:

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

WHAT'S ALSO IN THE NESUG PROCEEDINGS:

1) The mouse data set MouseUnevenInterval.sas7bdat
2) SAS source code

   All macros used to generate an uneven axis without intermediate ticks:
   - mkUnderlyingScale.sas
     - getIncr.sas
     - getAxisMax.sas
     - getAxisMin.sas
   - UnevenIntervalAxis.sas

   The calling program:
   - UnevenAxDemo.sas (also in the appendix)

TRADEMARK CITATION

CONTACT INFORMATION

The author welcomes feedback via email at perryWatts@comcast.net

APPENDIX: THE CALLING PROGRAM, UNEVENAXDEMO.SAS

```
/*  -----------------------------------------------------------------------
    Program  :  UnevenAxDemo.sas

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

    Input    :  psd.MouseUnevenInterval
    Output   :  demo1.cgm and demo2.cgm for distribution
    ----------------------------------------------------------------------  */
    options nodate number pageno=1 fmtsearch=(work) mautosource nomprint nosymbolgen;
    options sasautos=(sasautos 'c:\SASPapers\CustomAxis\Mac');
    libname psd 'c:\N08\CustomAxis\Data';
    title1 move=(+10pct,+0pct) 'Mouse Tumor Growth Study';
    %let output=C:\N08\CustomAxis\grf;

/******
 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 'HWCGM001' 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'; style='none'; text='dot'; size=1.8; y=Tmean; output;
    run;

/* NEED CALCXMIN AND CALCXMAX MACRO VARIABLES FOR MACRO MKUNDERLYINGSCALE 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=(,12.5pct);
/* 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=(,12.5pct);
```

```
/* 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=l h=2.5 "Left"
            t=2 j=l 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.
*****/
    %UnevenIntervalAxis(inDS=verticalLines, xvar=day, pctSize=3, xlabel=Days on Study, yOrigin=12.5)
    filename demo1 "&output.\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;
    %UnevenIntervalAxis(inDS=verticalLines, xvar=day, pctSize=3, xlabel=Days on Study,
                  yOrigin=12.5, xvalfmt=xaxisFM.)
    filename demo2 "&output.\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;
```