SAS:
Proc GPLOT

Computing for Research I

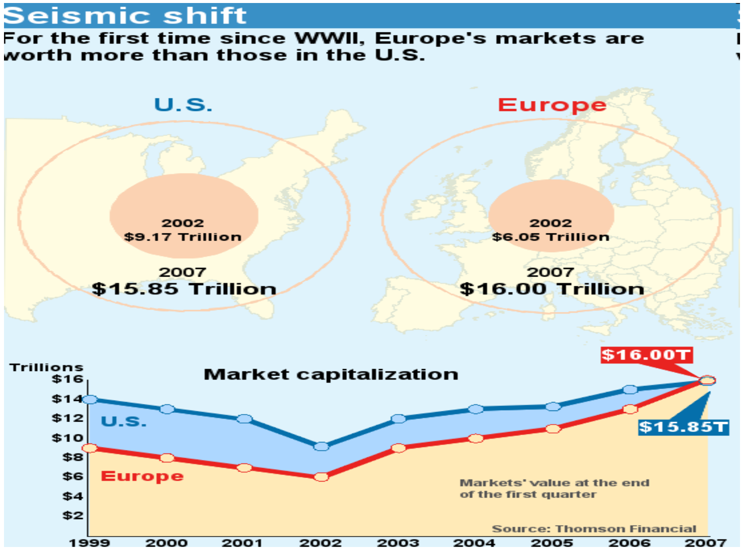
01/26/2011

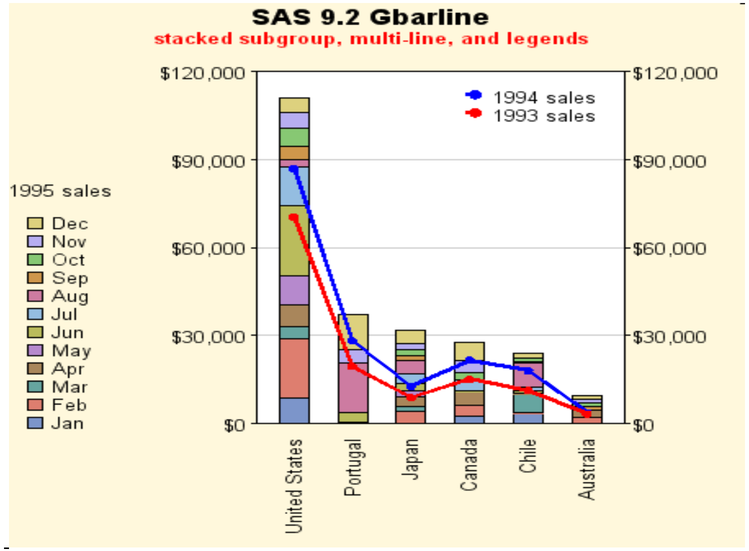
N. Baker

Introduction to SAS/GRAPH

- Graphics component of SAS system.
- Includes charts, plots, and maps in both 2 and 3 dimensions.
- Procedures included GCHART, GPLOT, GMAP, GCONTOUR etc…
- We will focus on PROC GPLOT

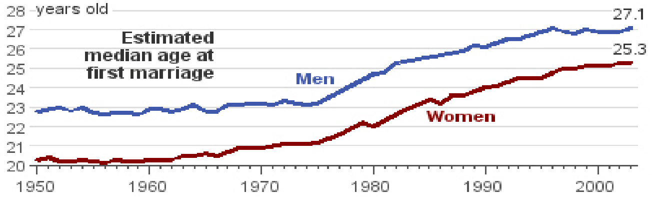
What can be done with SAS/GRAPH?





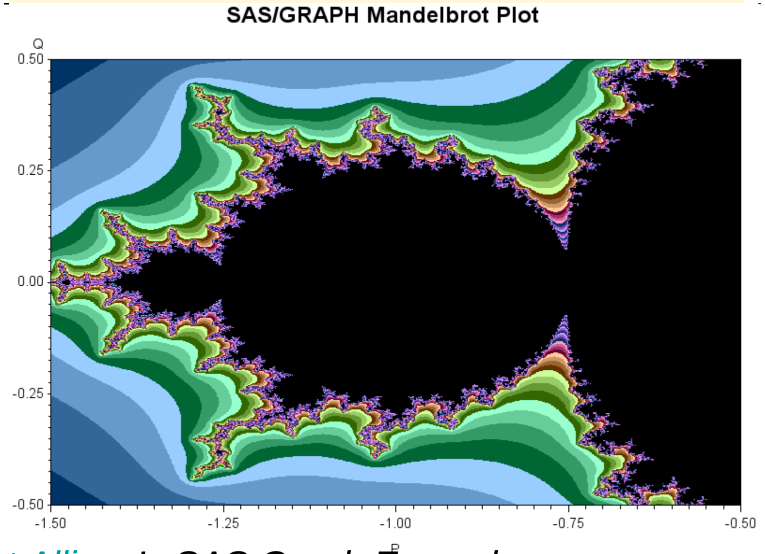
Waiting longer to tie the knot

Americans are waiting longer to get married than they did decades ago, with larget percentages of both men and women in their 20s and 30s who have yet to say "I do".



Percentage of men and women never married, by age...

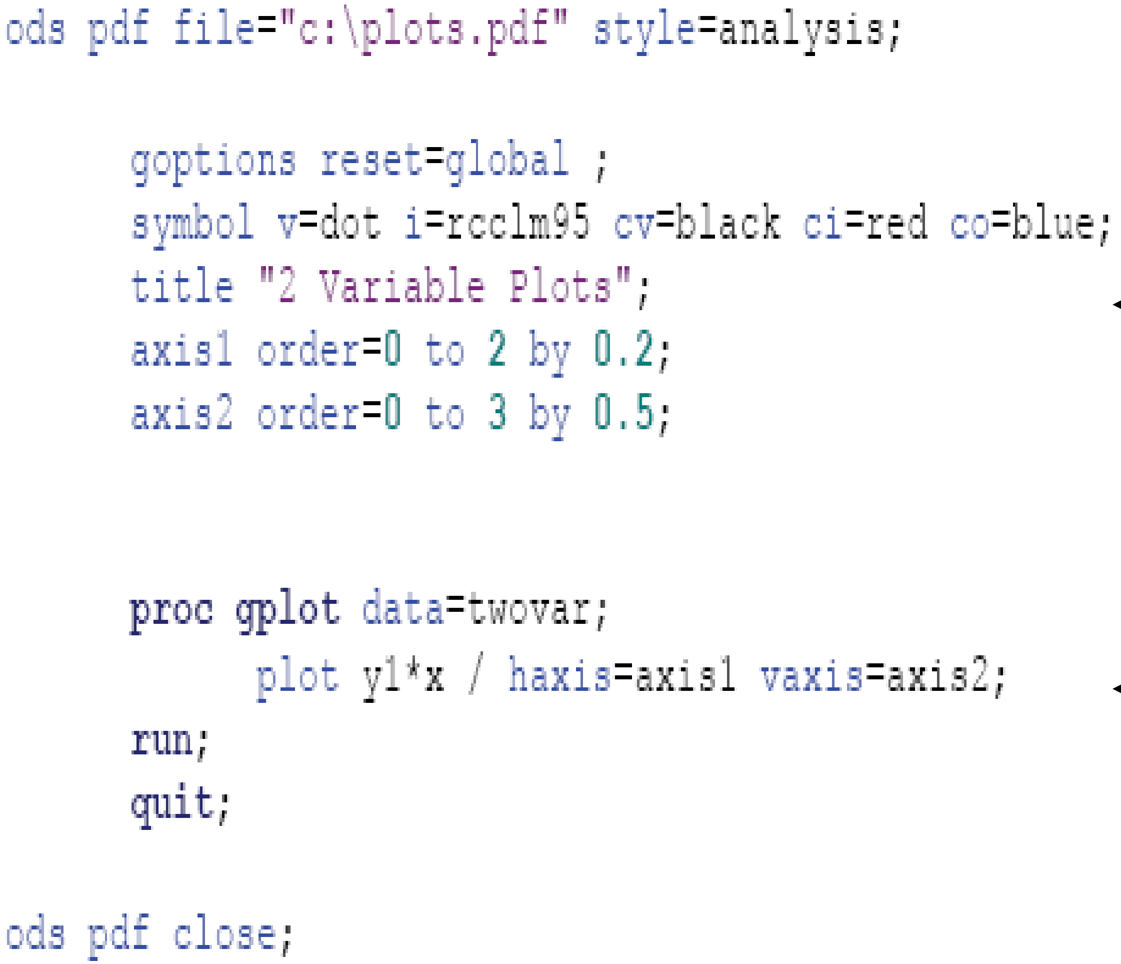
		1970			2003	
AGE	MEN		WOMEN	MEN		WOMEN
20-24	54.7		35.8	86.0		75.4%
25-29	19.1		10.5	54.6		40.3
30-34	9.4		6.2	33.1		22.7
35-44	6.7		5.2	19.5		13.2



These samples courtesy of Robert Allison's SAS Graph Examples

Elements of SAS/GRAPH

Overview



```
ods pdf file="c:\plots.pdf" style=analysis;

     goptions reset=global ;
     symbol v=dot i=rcclm95 cv=black ci=red co=blue;
     title "2 Variable Plots";
     axis1 order=0 to 2 by 0.2;
     axis2 order=0 to 3 by 0.5;


     proc gplot data=twovar;
          plot y1*x / haxis=axis1 vaxis=axis2;
     run;
     quit;

ods pdf close;
```

ODS

Destination

Elements

Global Statements

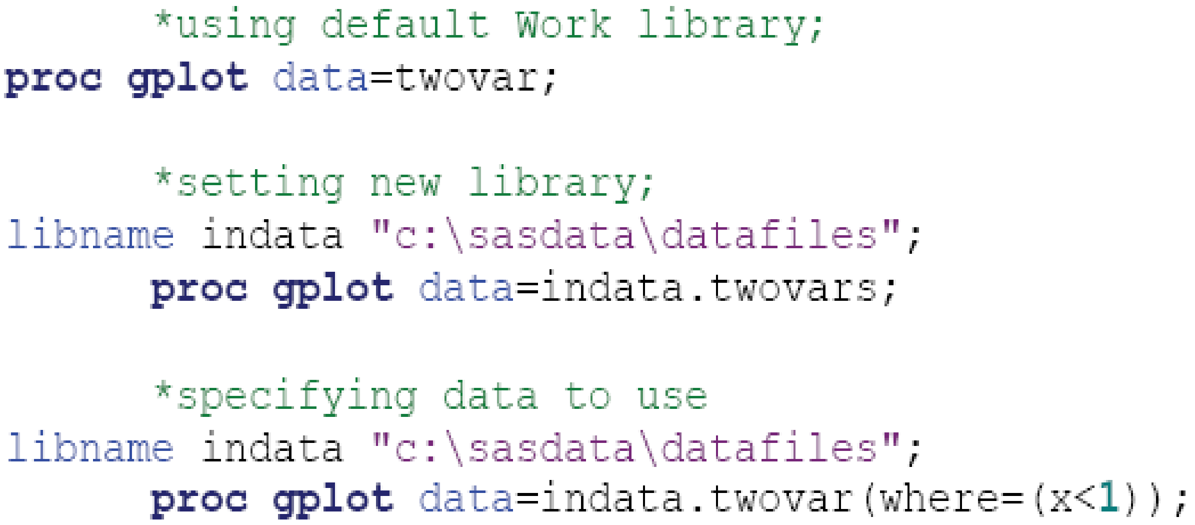
Procedure Step

Taken from SAS 9.2 documentation

Elements of SAS/GRAPH
PROC GPLOT: *Specifying an input data set*

Similar to all other SAS PROC’s

– Proc gplot data=<libname>.<data set><options>;



```
      *using default Work library;
proc gplot data=twovar;

      *setting new library;
libname indata "c:\sasdata\datafiles";
      proc gplot data=indata.twovars;

      *specifying data to use
libname indata "c:\sasdata\datafiles";
      proc gplot data=indata.twovar(where=(x<1));
```

Options include setting annotate data sets, image mapping for drill-down plots in web applications, Creating Uniform axis across plots, and specifying SAS catalog for placement of output.

Elements of SAS/GRAPH

PROC GPLOT: *Plotting*

- You can use up to 2 plots statements at a time, however, at least one Plot statement is required.

- The plot statement is used to control the axis, plotting points, labels, tick marks, and the plot legend.

- **The only required arguments are…**
  - **Plot <*Y Variable*>*<*X Variable*> / <options>;**

Elements of SAS/GRAPH

PROC GPLOT: *Plotting Options*

- Options for plotting
  - Plot options
    - Legend= or nolegend: specifies figure legend options
    - Overlay: allows overlay of more than one Y variable
    - Skipmiss: breaks the plotting line where Y values are missing
  - Appearance option
    - Axis: Specifies axis label and value options
    - Symbol: Specified symbol options
    - href, vref: Draws vertical or horizontal reference lines on plot
    - frame/fr or noframe/nofr: specifies whether or not to frame the plot
    - caxis/ca, cframe/cfr, chref/ch, cvref/cv, ctext/c: specifies colors used for axis, frame, text or reference lines.

Introduction to SAS/GRAPH

- We will begin with rather simple code and let SAS decide how our graph will look.

- Then we will step through a few options that allow us to control and adjust the graphic output.

Examples

2 Variable Plotting / Scatter plots

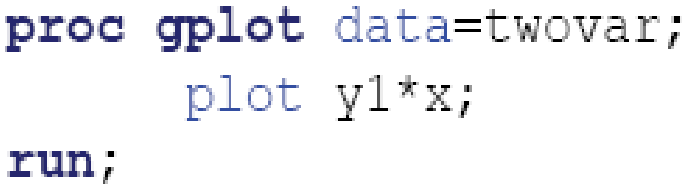
Examples

2 Variables

- Suppose subjects are given a doses of experimental medication based on body weight over a 24 hour period (mg/24hrs). Variable X

- On the following day, each subject had their Vascular Cell Adhesion Molecule (µg/ml) levels measured. Variable Y1

- The investigators are interested in seeing a plot of the dose given vs. the plasma VCAM levels to see if there may be an effect of the drug dose.

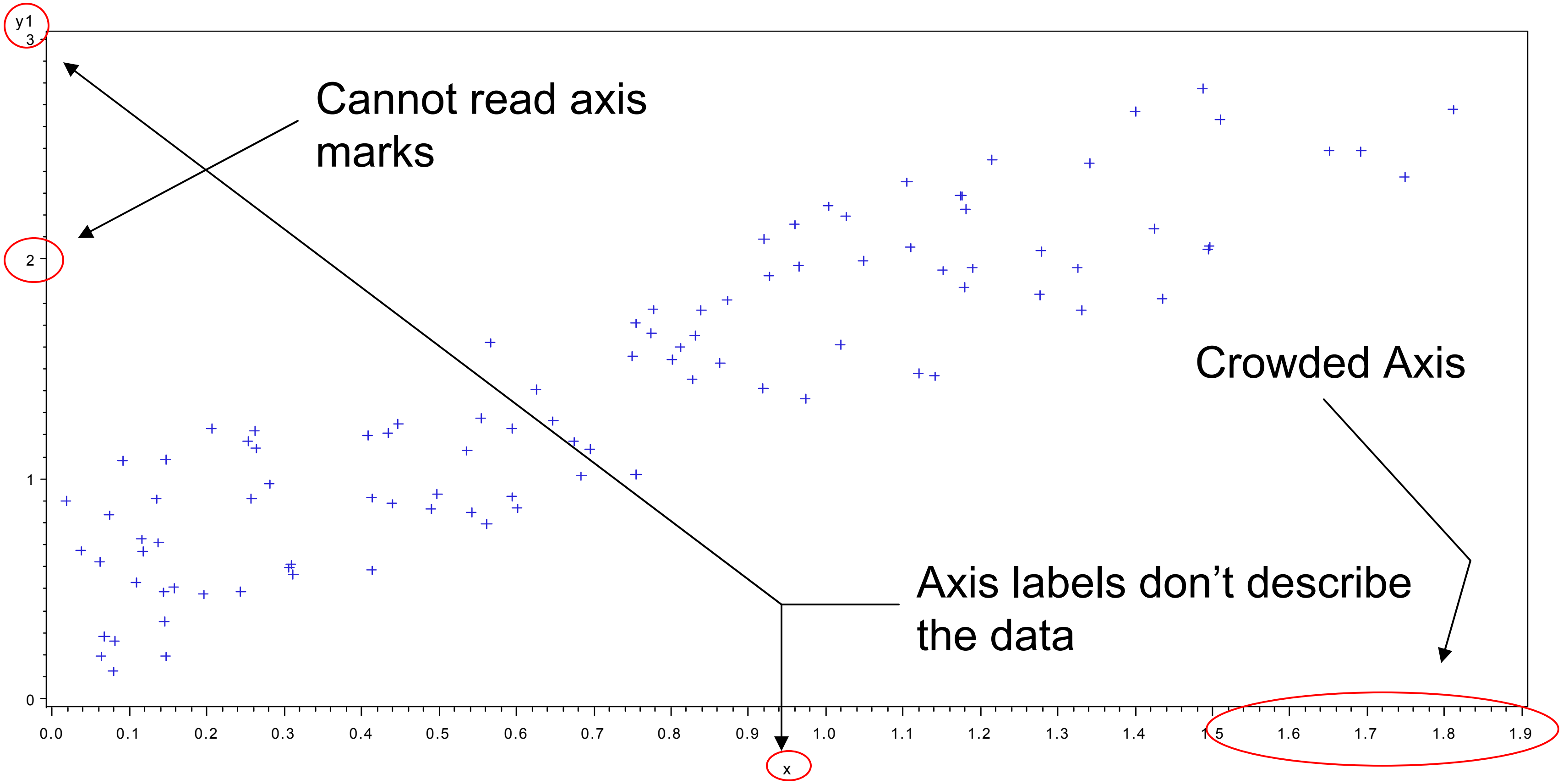


```
proc gplot data=twovar;
        plot y1*x;
run;
```

Examples
2 Variables

```
proc gplot data=twovar;
        plot y1*x;
run;
```

Very basic plot, below we get all of the default options. Not very exciting. Definitely not publication quality.



Cannot read axis marks

Crowded Axis

Axis labels don’t describe the data

Examples

2 Variables: AXIS Statements

- AXIS<1..99> <options>;
  - Label Option;
    - Angle/a=degrees (0-359)
    - Color/c=text color
    - Font/f=font
    - Height/h=text height (default=1)
    - Justify=(left/center/right)
    - Label=“text string”
      - Options precede label

- axis1 label=(a=90 c=black f=“arial” h=1.2 “time”
  a=90 c=black f=“arial” h=1.0 “hours”);

Examples

2 Variables: AXIS Statements

- AXIS<1..99> <options>;
  - Order Option
    - Order=(a to b by c): major tick marks will show up at intervals based on c.
      - Example order=(0 to 3 by 1);
  - Value Option
    - value=(“” “” “”): applies text label to each major tick.
      - Example Value=(“Start” “Middle” “End”)

Examples

2 Variables: AXIS Statements

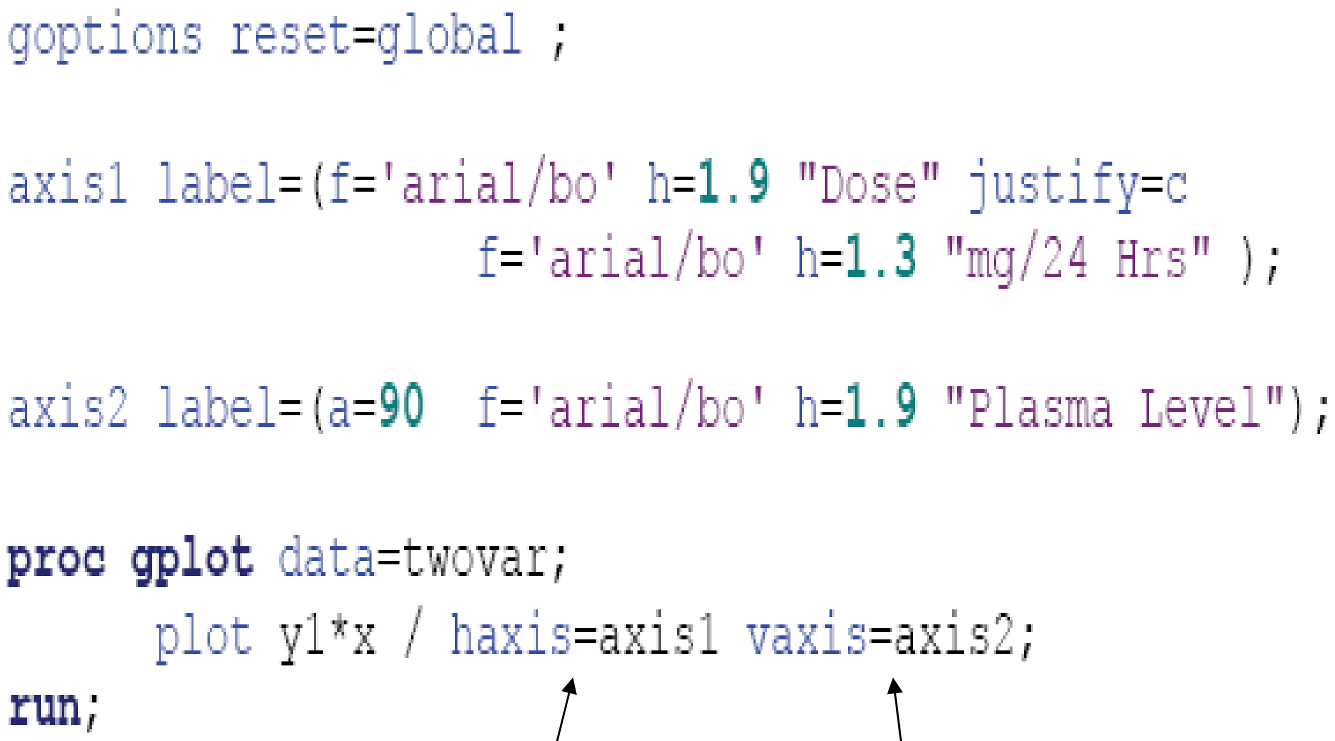
Resets previous options

Horizontal axis

(X Variable)

Vertical axis

(Y Variable)



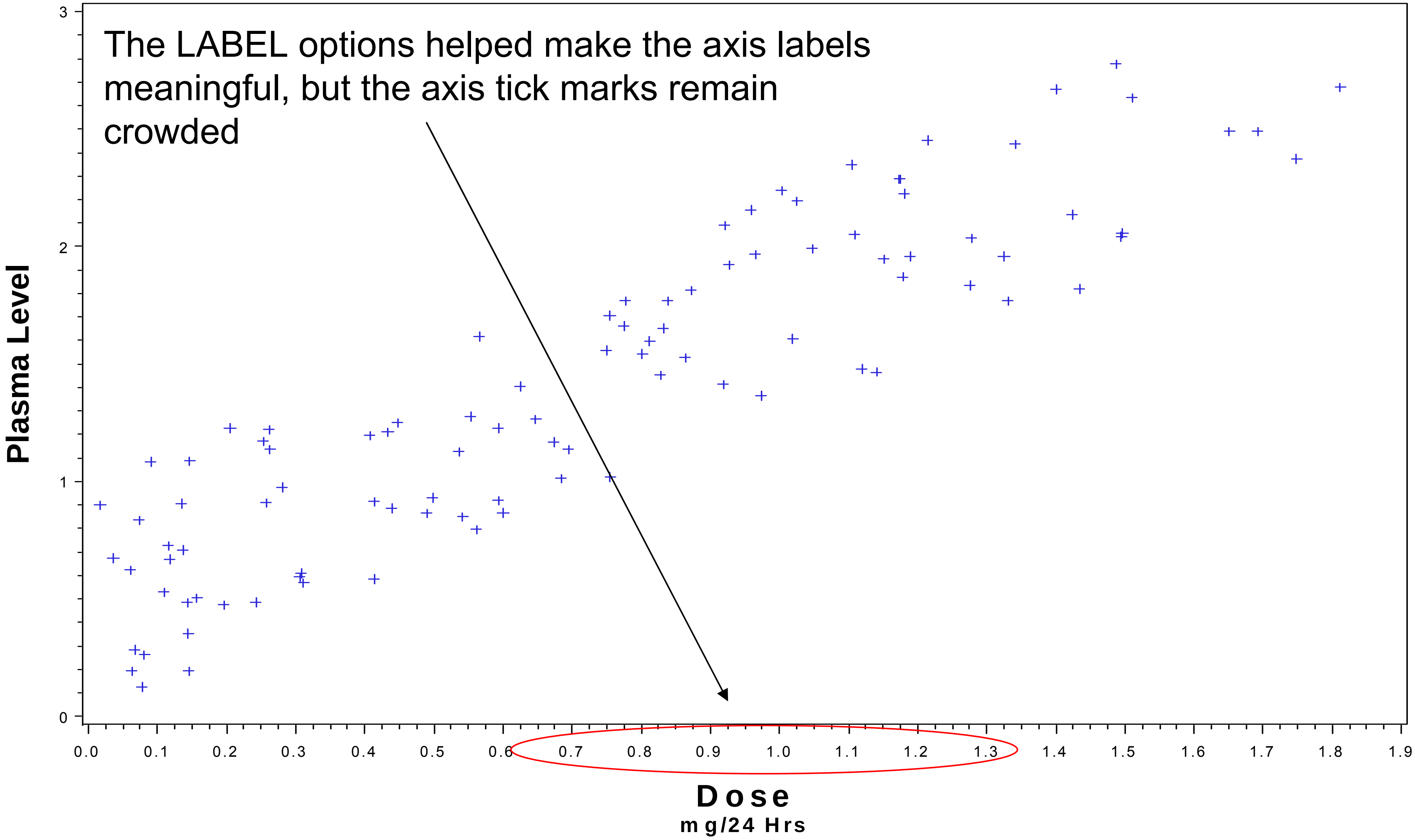
```
goptions reset=global ;

axis1 label=(f='arial/bo' h=1.9 "Dose" justify=c
                         f='arial/bo' h=1.3 "mg/24 Hrs" );

axis2 label=(a=90  f='arial/bo' h=1.9 "Plasma Level");

proc gplot data=twovar;
      plot y1*x / haxis=axis1 vaxis=axis2;
run;
```

Call Axis statements

NOTE: you can also place the AXIS statements within the gplot proc

Examples

2 Variables: AXIS Statements

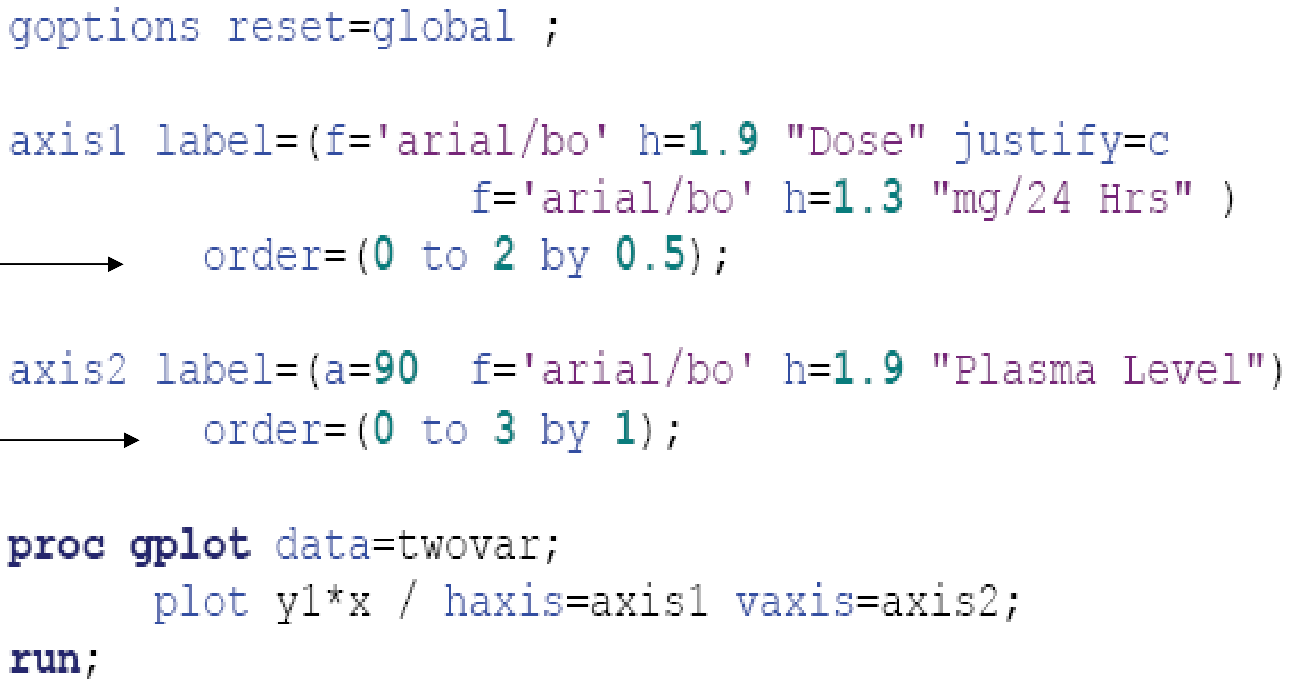


The LABEL options helped make the axis labels meaningful, but the axis tick marks remain crowded

Examples

2 Variables: AXIS Statement

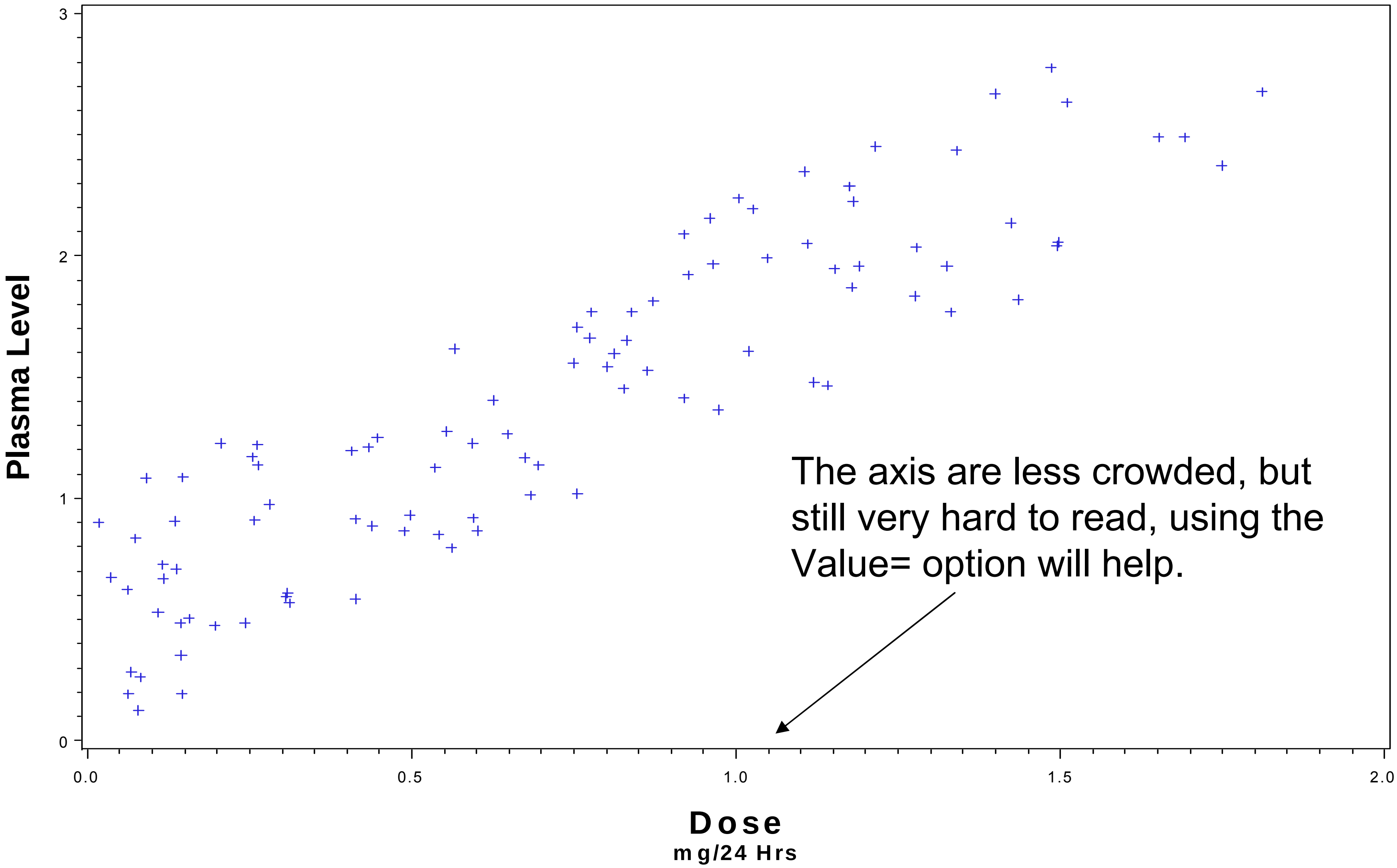
Added ORDER option to Axis statement



```
goptions reset=global ;

axis1 label=(f='arial/bo' h=1.9 "Dose" justify=c
                       f='arial/bo' h=1.3 "mg/24 Hrs" )
          order=(0 to 2 by 0.5);

axis2 label=(a=90  f='arial/bo' h=1.9 "Plasma Level")
          order=(0 to 3 by 1);

proc gplot data=twovar;
       plot y1*x / haxis=axis1 vaxis=axis2;
run;
```

Examples

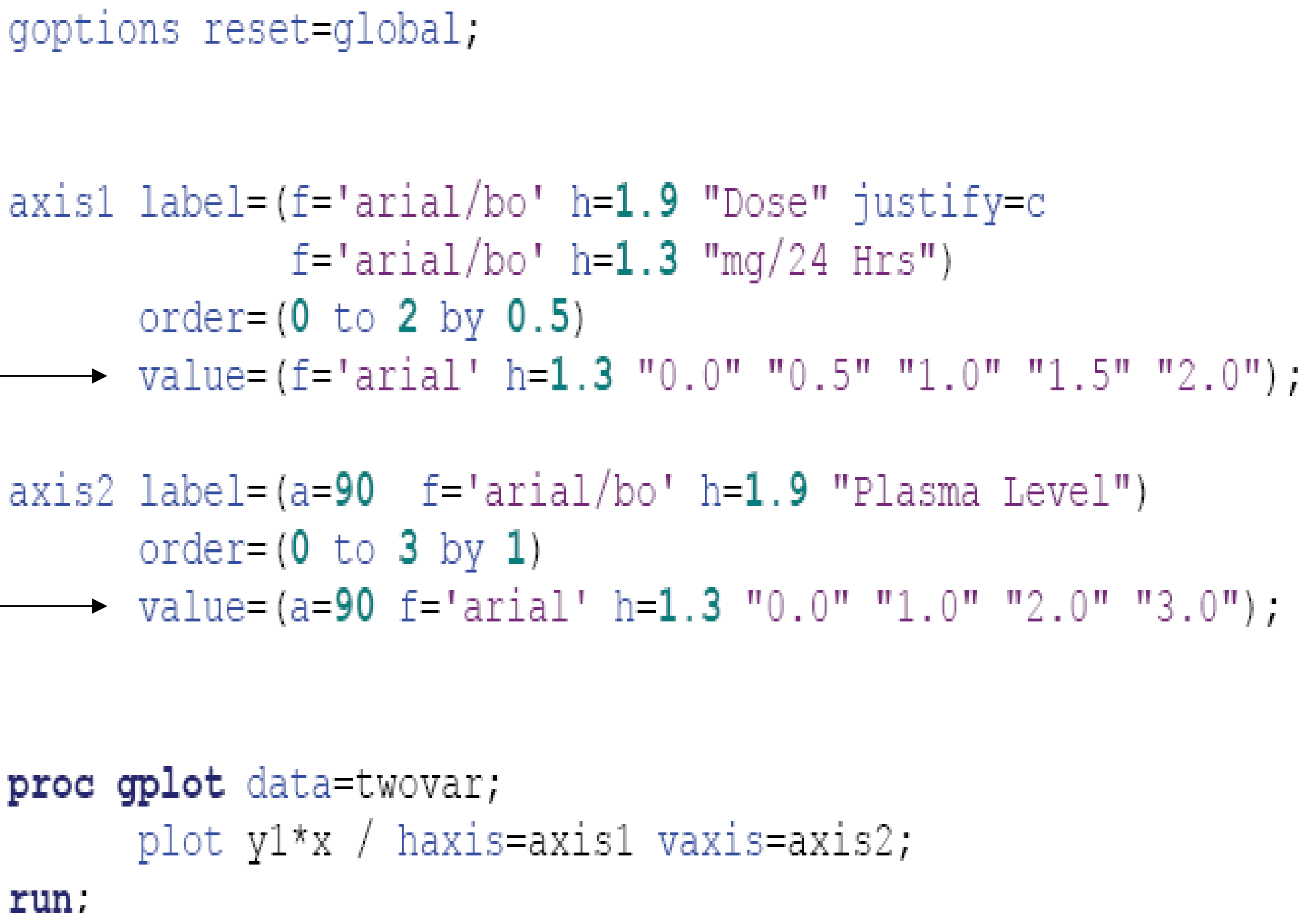
2 Variables: AXIS Statement



The axis are less crowded, but still very hard to read, using the Value= option will help.

Examples

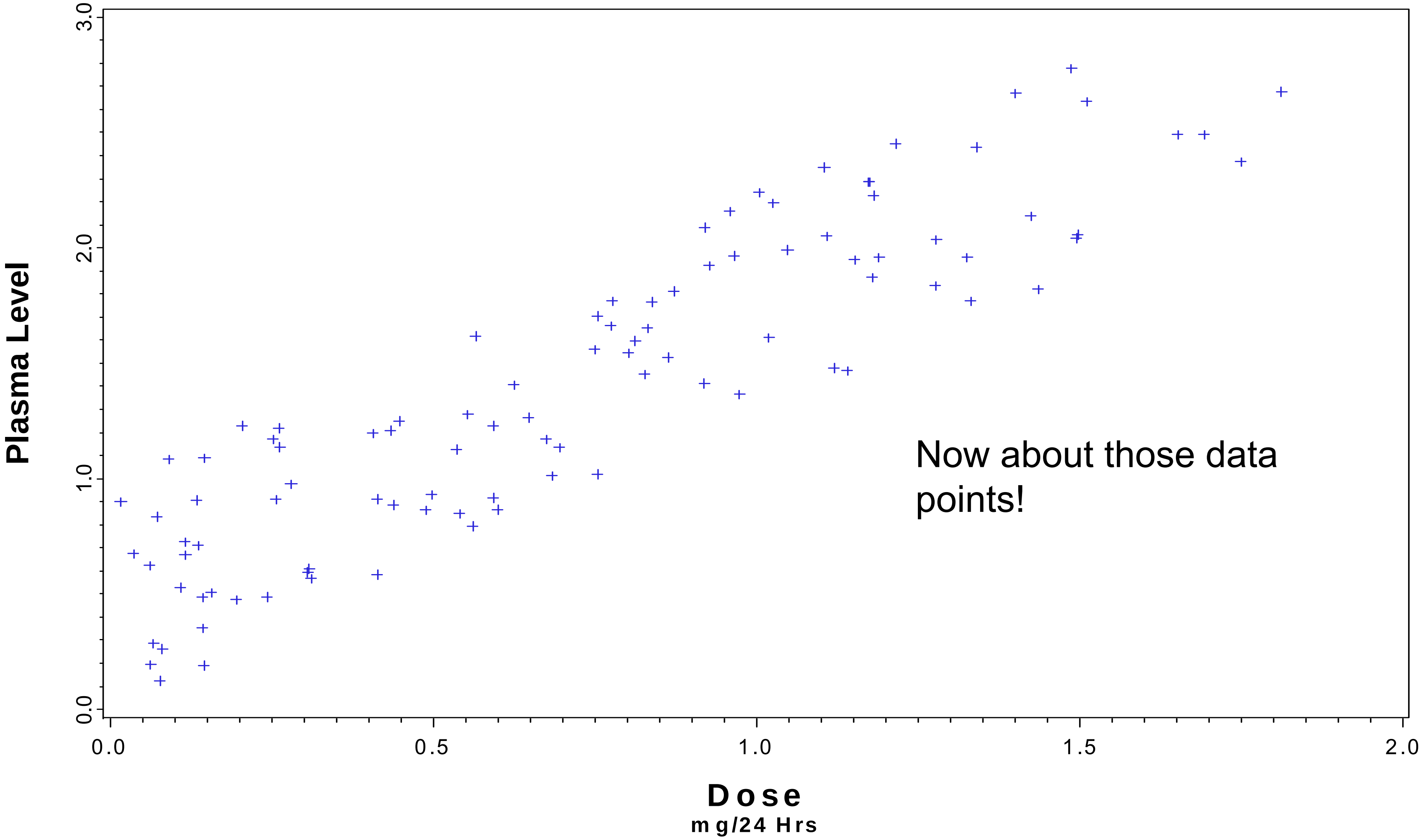
2 Variables: AXIS Statement



```
goptions reset=global;


axis1 label=(f='arial/bo' h=1.9 "Dose" justify=c
             f='arial/bo' h=1.3 "mg/24 Hrs")
      order=(0 to 2 by 0.5)
      value=(f='arial' h=1.3 "0.0" "0.5" "1.0" "1.5" "2.0");

axis2 label=(a=90  f='arial/bo' h=1.9 "Plasma Level")
      order=(0 to 3 by 1)
      value=(a=90 f='arial' h=1.3 "0.0" "1.0" "2.0" "3.0");


proc gplot data=twovar;
      plot y1*x / haxis=axis1 vaxis=axis2;
run;
```

Added VALUE option to Axis statement

Examples
2 Variables



Now about those data points!

Examples

2 Variables: Symbol Statement

- Symbol<1…255> <options>;
  - Symbol options
    - Color= value color
    - Ci=line color
    - Height=symbol height
    - Line=line type
    - Value=symbol
    - Width=thickness factor
    - Interpol=point interpolations

Examples

2 Variables: Symbol Statement

- Symbol<1…255> <options>;
  - Symbol options
    - Interpolation options
      - Join, box, hilo interpolation, regression, spline, standard deviations.
    - value options
      - Dot, circle, star, square, plus, minus, “text value”.
    - Color options
      - 256 colors available,

www.devenezia.com/docs/SAS/sas-colors.html

Examples

2 Variables: Symbol Statement

Symbol options

- Interpolation options
  - None
  - Join: points connected by straight line
  - Needle: vertical line from horizontal axis to point
  - Stepx: (L,R,C) step function, stepxJ will add a verticle line to each step plot
  - stdkxxx: (M,P,J,B,T) k=1,2,3 (standard deviations) or
    - » stdM=SEM, stdp=uses pooled sample variance, stdj=joins the errors, T will give tops and bottoms to error lines, where B will request error bars.
  - HILOxxx: (T,B,C,J)

Examples

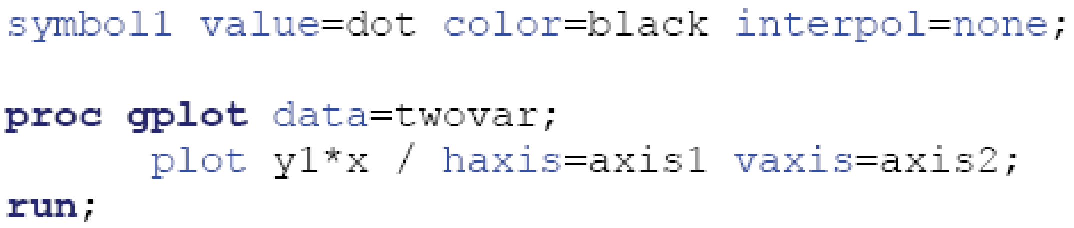
2 Variables: Symbol Statement

Symbol options

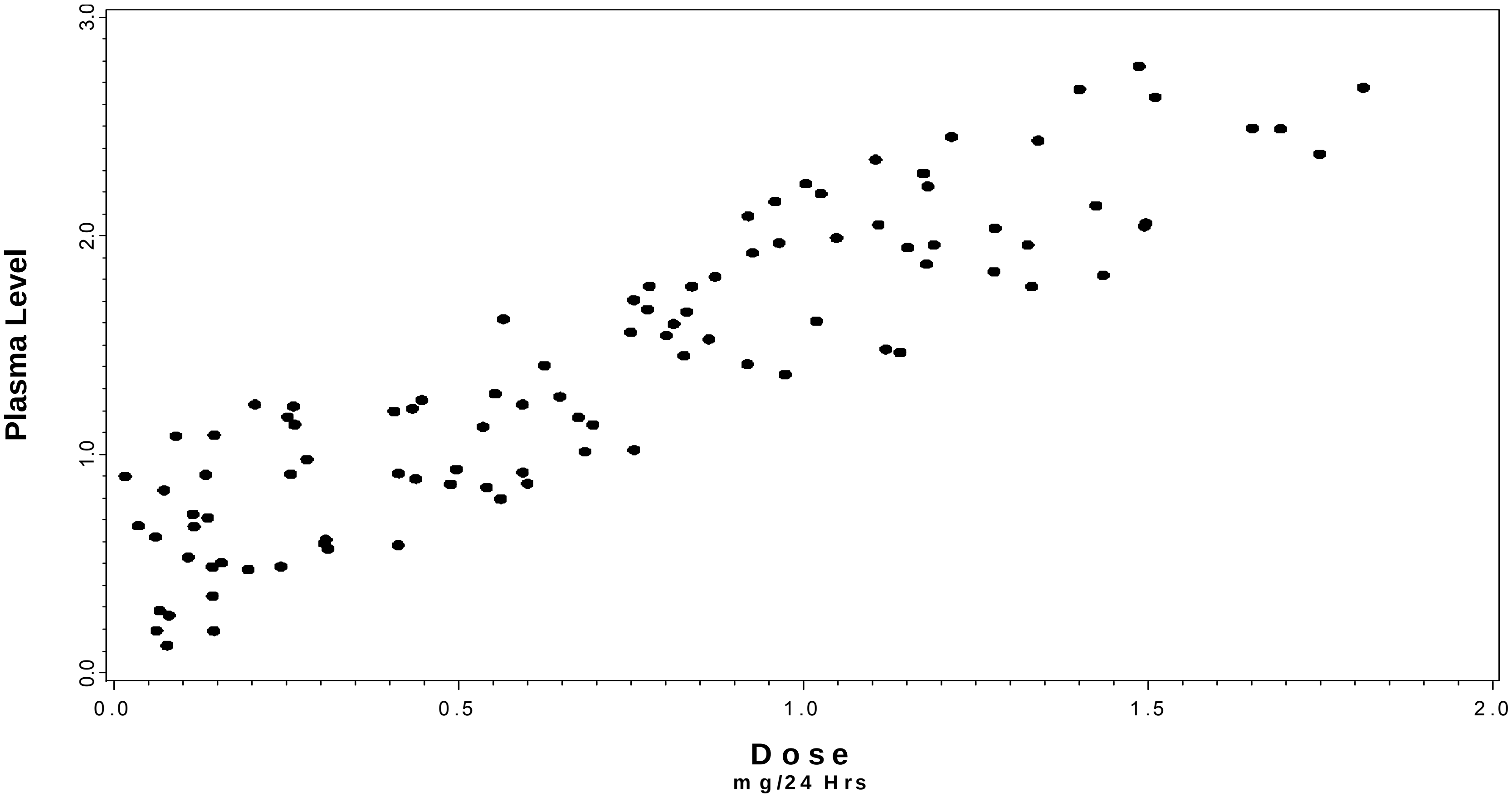
- Interpolation options
  - R-series interpolation
  - Rxxxxxxx
    - RL: linear regression
    - RQ: Quadratic Regression
    - RC: Cubic Regression
    - CLM: CI for mean predicted values
    - CLI: CI for Individual predicted values
    - 90, 95, 99: confidence limits

Examples

2 Variables: SYMBOL Statement



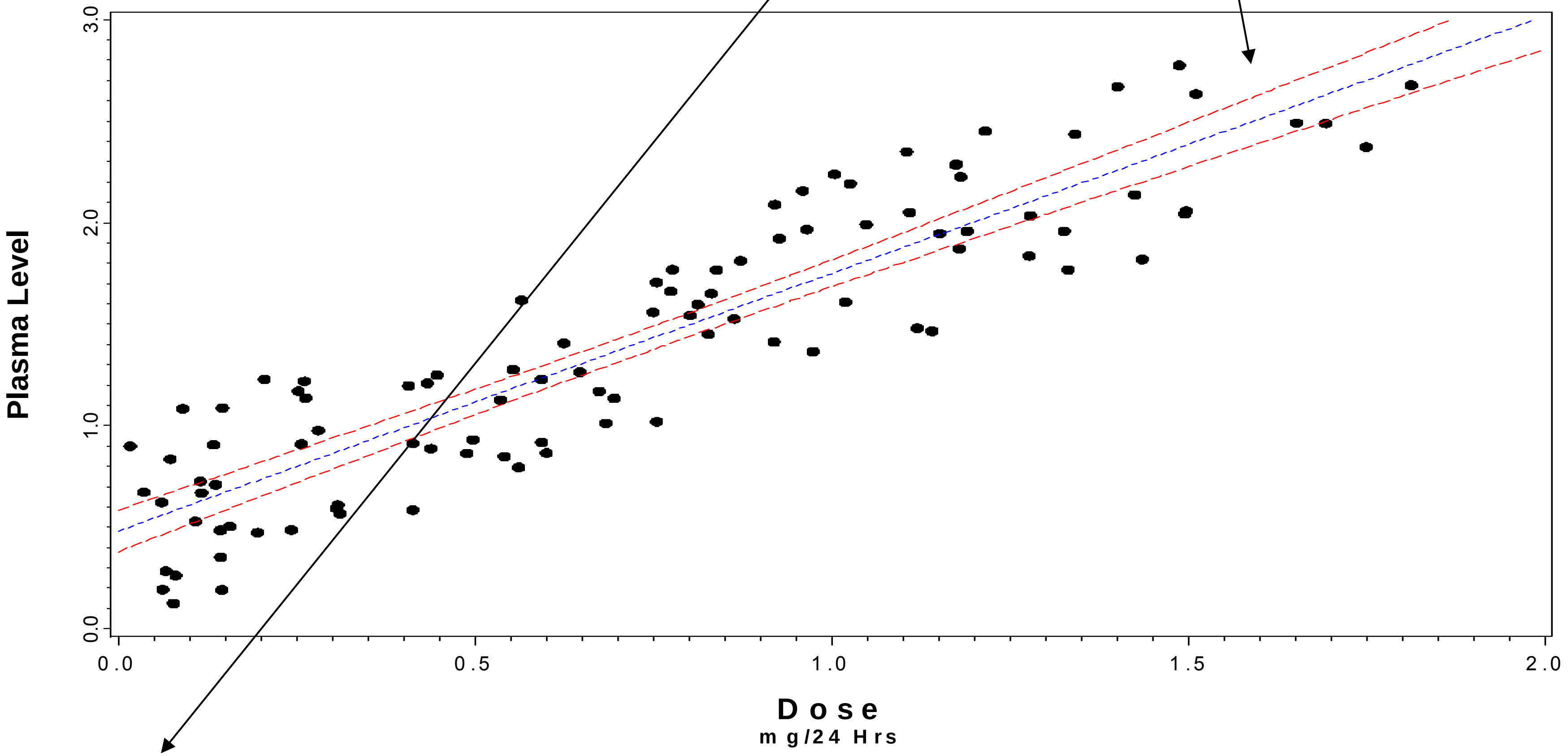
```
symbol1 value=dot color=black interpol=none;

proc gplot data=twovar;
      plot y1*x / haxis=axis1 vaxis=axis2;
run;
```



Examples

2 Variables: Adding Regression Lines

```
symbol1 value=dot color=black
                interpol=rlclm95 ci=blue co=red line=2;

proc gplot data=twovar;
     plot y1*x / haxis=axis1 vaxis=axis2 regeqn;
run;
```



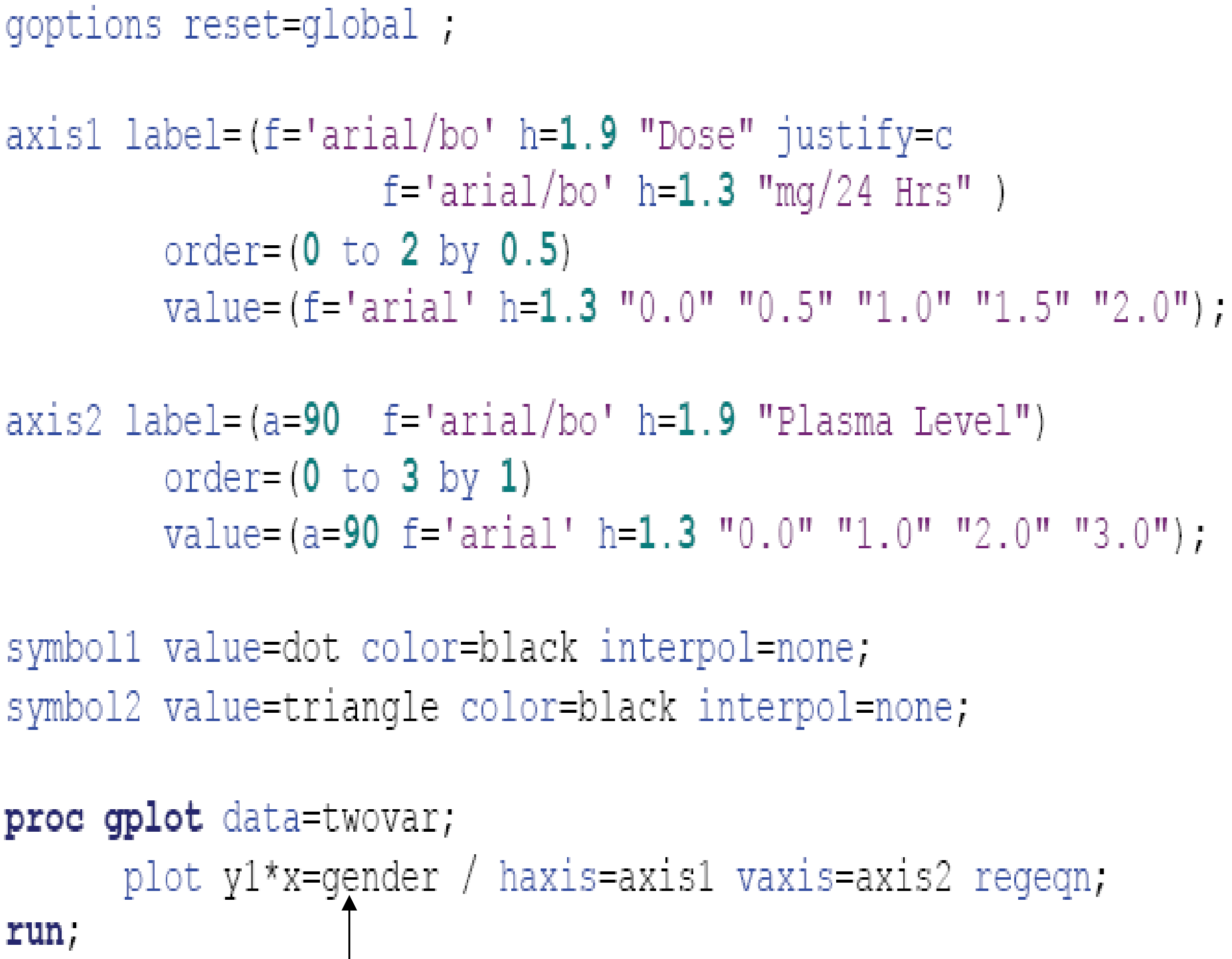
Regression Equation:
y1 = 0.481173 + 1.269433*x

Examples
Grouping Variables

- Many times we want to look at group differences.
- Demographic groups, treatment groups, etc…
- Grouping variable must be in the data file.

Examples

Grouping Variables



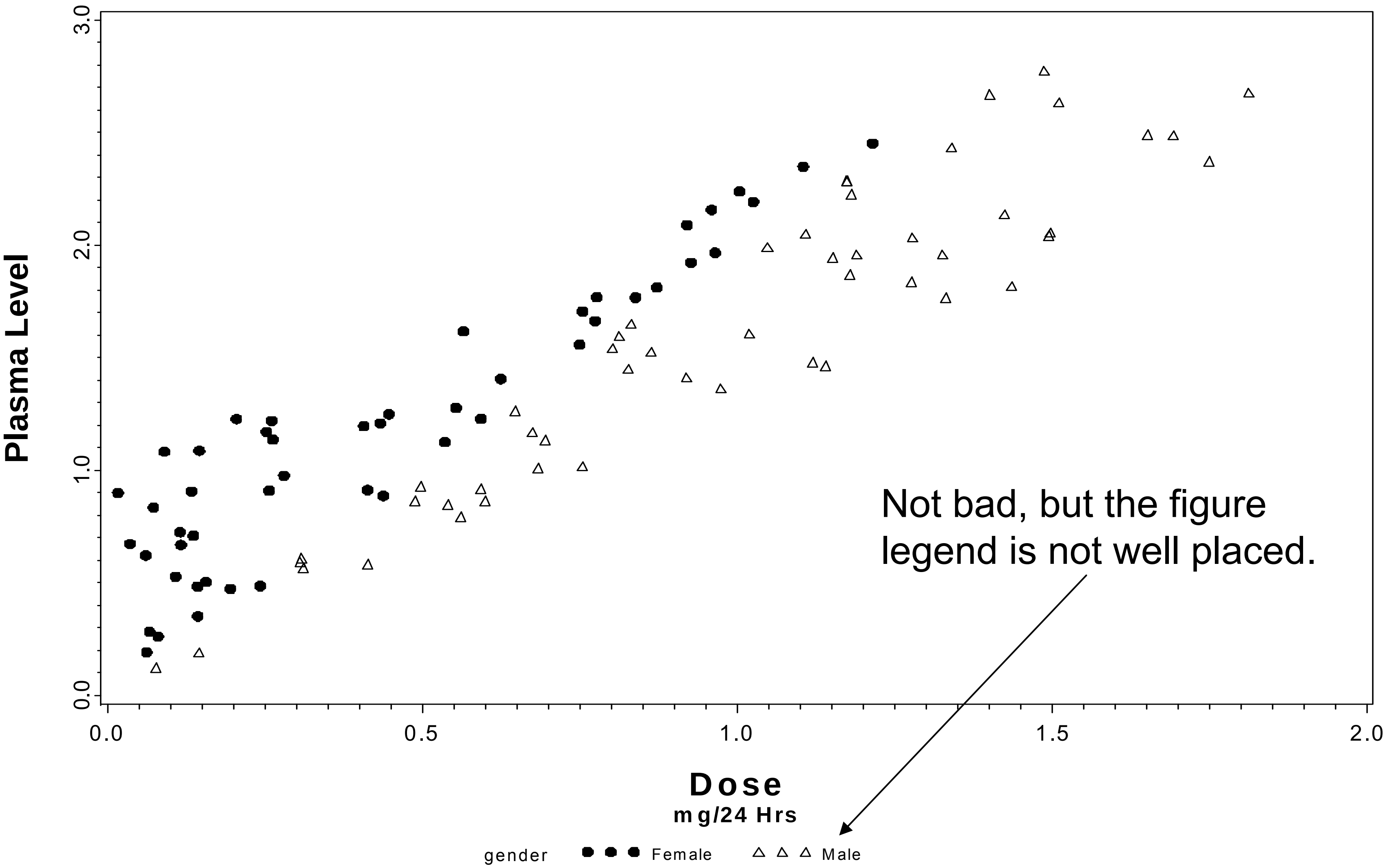
```
goptions reset=global ;

axis1 label=(f='arial/bo' h=1.9 "Dose" justify=c
                   f='arial/bo' h=1.3 "mg/24 Hrs" )
        order=(0 to 2 by 0.5)
        value=(f='arial' h=1.3 "0.0" "0.5" "1.0" "1.5" "2.0");

axis2 label=(a=90  f='arial/bo' h=1.9 "Plasma Level")
        order=(0 to 3 by 1)
        value=(a=90 f='arial' h=1.3 "0.0" "1.0" "2.0" "3.0");

symbol1 value=dot color=black interpol=none;
symbol2 value=triangle color=black interpol=none;

proc gplot data=twovar;
      plot y1*x=gender / haxis=axis1 vaxis=axis2 regeqn;
run;
```

You need to add a new SYMBOL statement for the each additional group.

Add the grouping variable to the PLOT statement

Examples

Grouping Variables



Not bad, but the figure legend is not well placed.

Examples

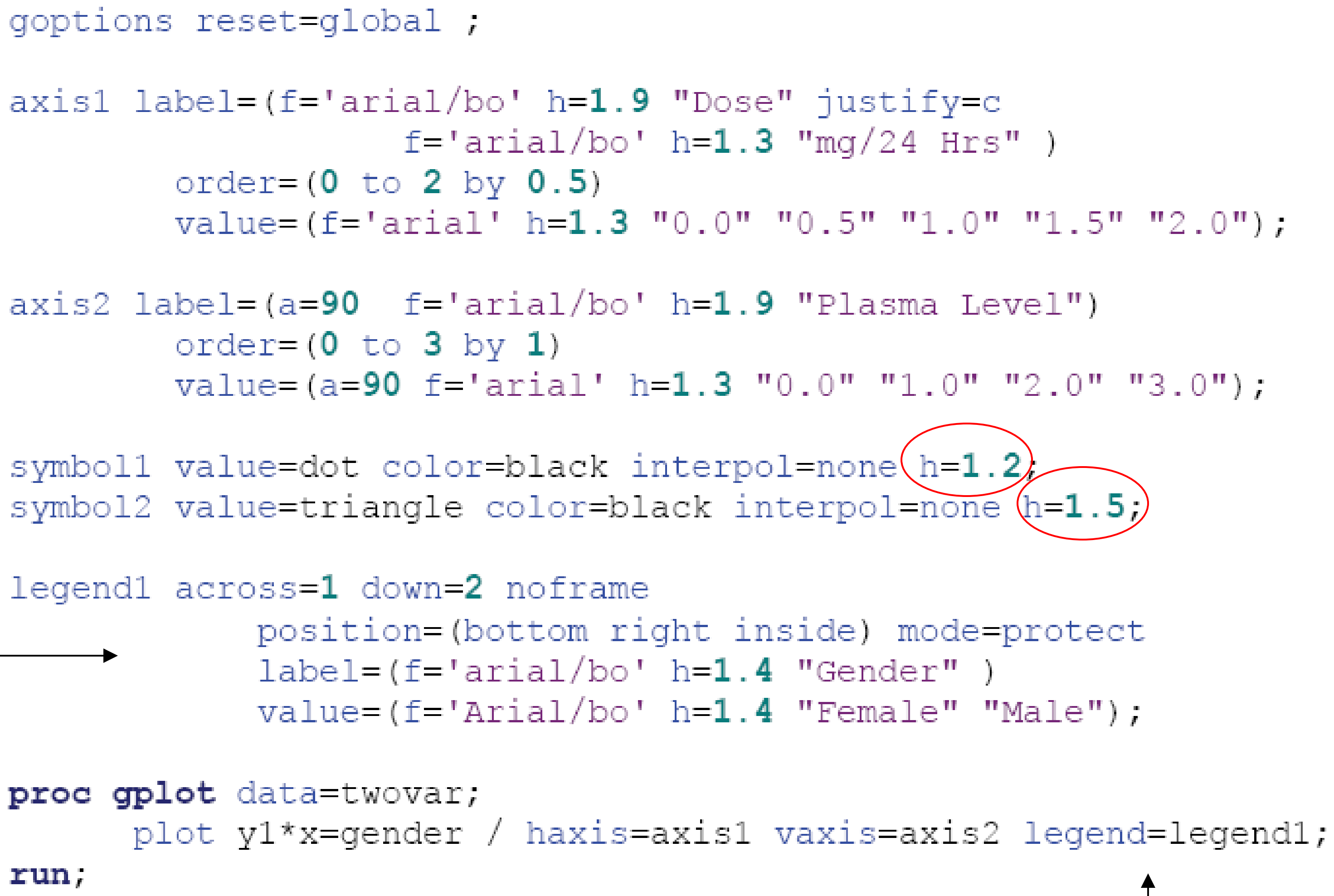
Grouping Variables: Legend Statement

- Legend<1…99> <options>;
  - Legend options
    - Across=: number of columns
    - Down=: number of rows
    - Frame/noframe
    - Position=(bottom, middle, top) (left, center, right) (inside, outside)
    - Origin=(x,y)
    - Label=
    - Order=
    - Value=

These options are the same as within the axis statement discussed earlier

Examples

Grouping Variables: Legend Statement



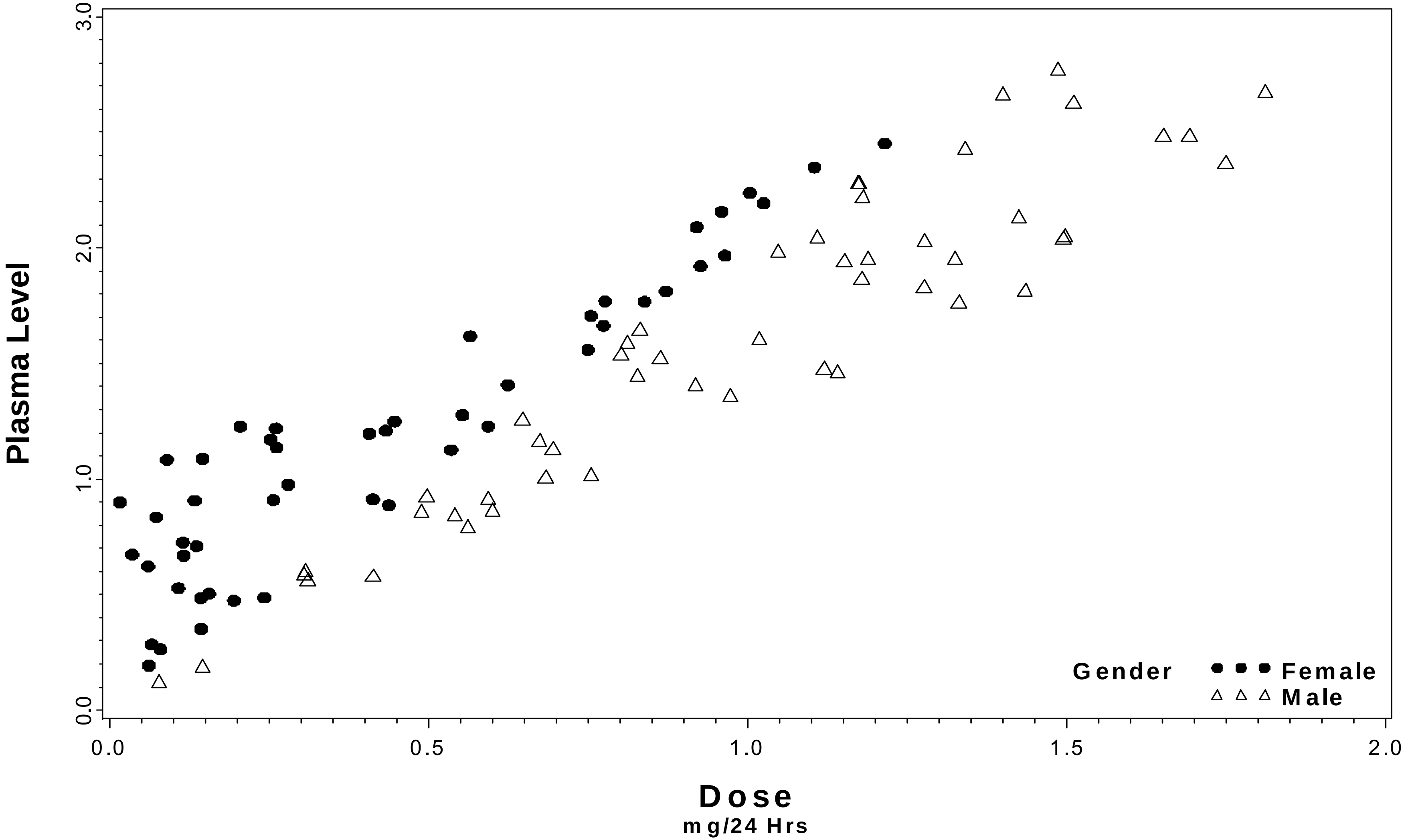
```
goptions reset=global ;

axis1 label=(f='arial/bo' h=1.9 "Dose" justify=c
                   f='arial/bo' h=1.3 "mg/24 Hrs" )
        order=(0 to 2 by 0.5)
        value=(f='arial' h=1.3 "0.0" "0.5" "1.0" "1.5" "2.0");

axis2 label=(a=90  f='arial/bo' h=1.9 "Plasma Level")
        order=(0 to 3 by 1)
        value=(a=90 f='arial' h=1.3 "0.0" "1.0" "2.0" "3.0");

symbol1 value=dot color=black interpol=none h=1.2;
symbol2 value=triangle color=black interpol=none h=1.5;

legend1 across=1 down=2 noframe
            position=(bottom right inside) mode=protect
            label=(f='arial/bo' h=1.4 "Gender" )
            value=(f='Arial/bo' h=1.4 "Female" "Male");

proc gplot data=twovar;
      plot y1*x=gender / haxis=axis1 vaxis=axis2 legend=legend1;
run;
```

Legend Statement

Call Legend Statement

Examples

Grouping Variables: Legend Statement



Examples

Repeated Measures/Longitudinal Plotting

Examples

Repeated Measures/Longitudinal Plotting

- Suppose that you have many observations on each subject taken at various time points.

- 40 subjects
- 2 treatments (Placebo and Active med)
- 5 time points (baseline plus 4 1-week intervals)
  - During the last week, both treatment groups receive Placebo
- Data should be in the Long format

At diagnosis, subjects are randomized to an experimental treatment or placebo. During the final week of treatment, all subjects will receive active medicatio.

Examples

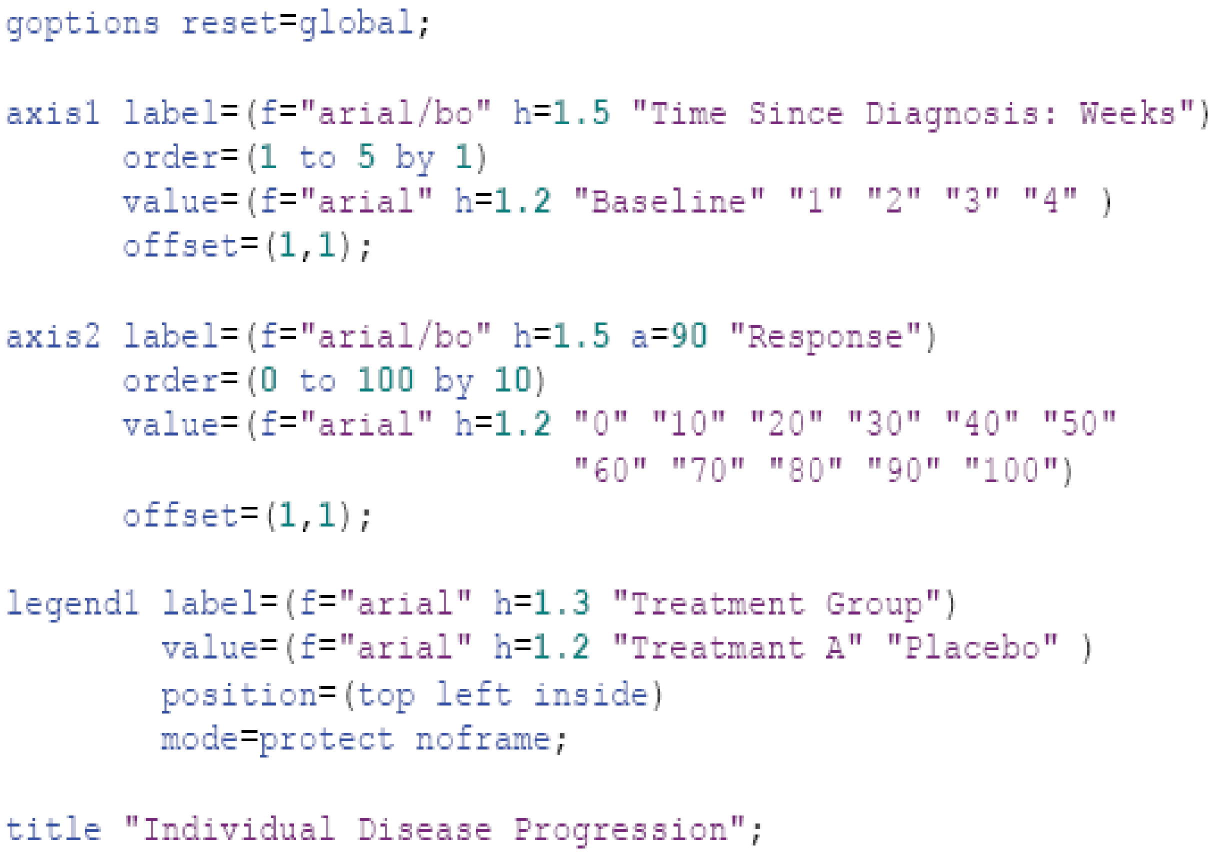
Repeated Measures/Longitudinal Plotting

Create appropriate axis and legend statements as before.

AXIS for X (time) variable

AXIS for Y (Response) variable

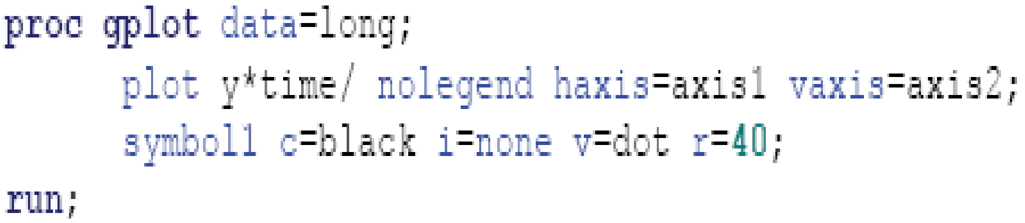
Added TITLE statement for plot



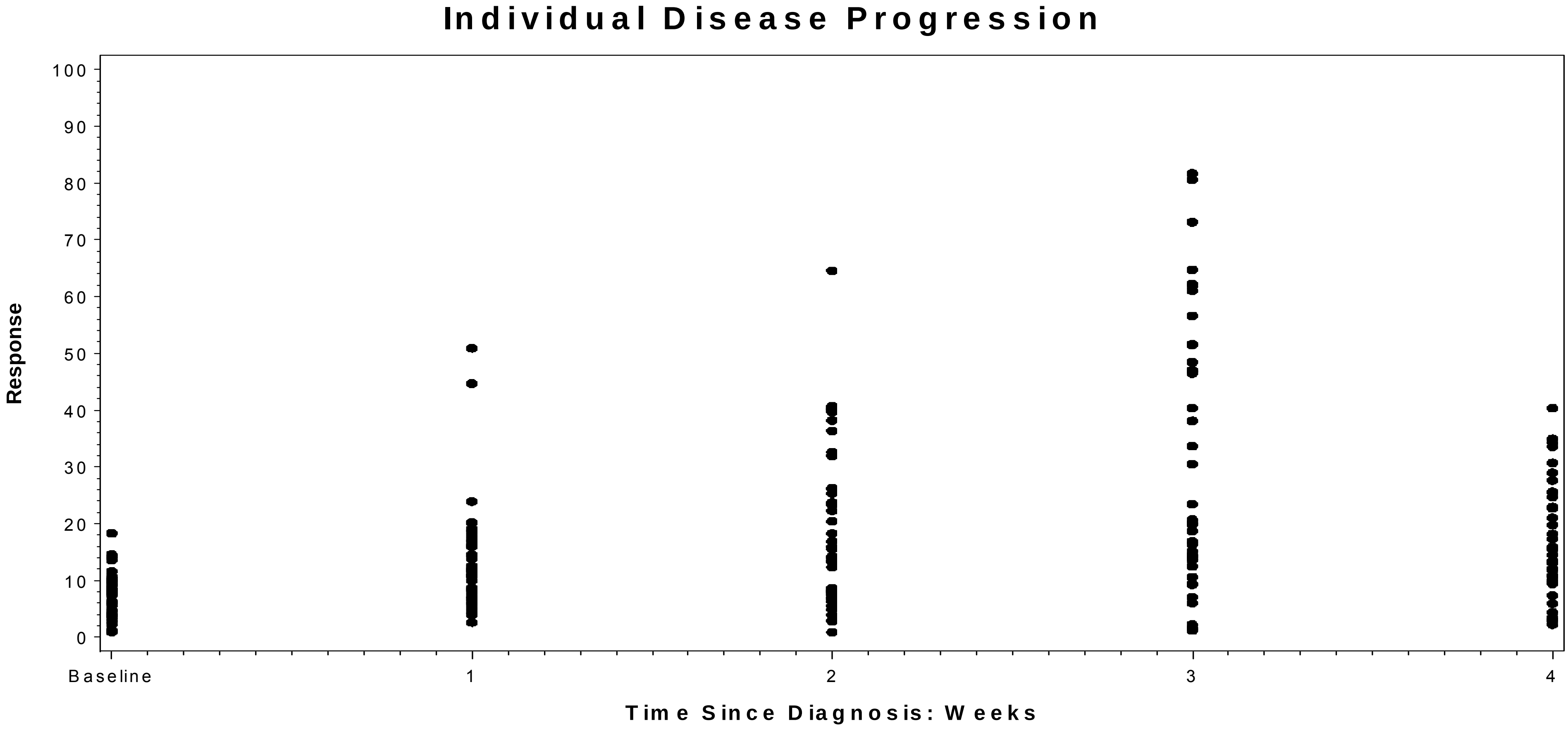
```
goptions reset=global;

axis1 label=(f="arial/bo" h=1.5 "Time Since Diagnosis: Weeks")
      order=(1 to 5 by 1)
      value=(f="arial" h=1.2 "Baseline" "1" "2" "3" "4" )
      offset=(1,1);

axis2 label=(f="arial/bo" h=1.5 a=90 "Response")
      order=(0 to 100 by 10)
      value=(f="arial" h=1.2 "0" "10" "20" "30" "40" "50"
                             "60" "70" "80" "90" "100")
      offset=(1,1);

legend1 label=(f="arial" h=1.3 "Treatment Group")
        value=(f="arial" h=1.2 "Treatmant A" "Placebo" )
        position=(top left inside)
        mode=protect noframe;

title "Individual Disease Progression";
```

Examples

Repeated Measures/Longitudinal Plotting

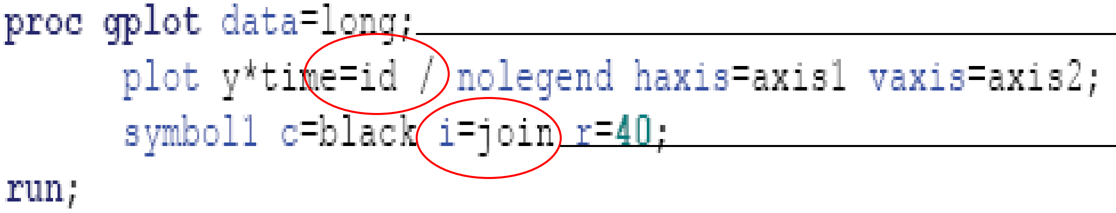


```
proc gplot data=long;
     plot y*time/ nolegend haxis=axis1 vaxis=axis2;
     symbol1 c=black i=none v=dot r=40;
run;
```



Examples

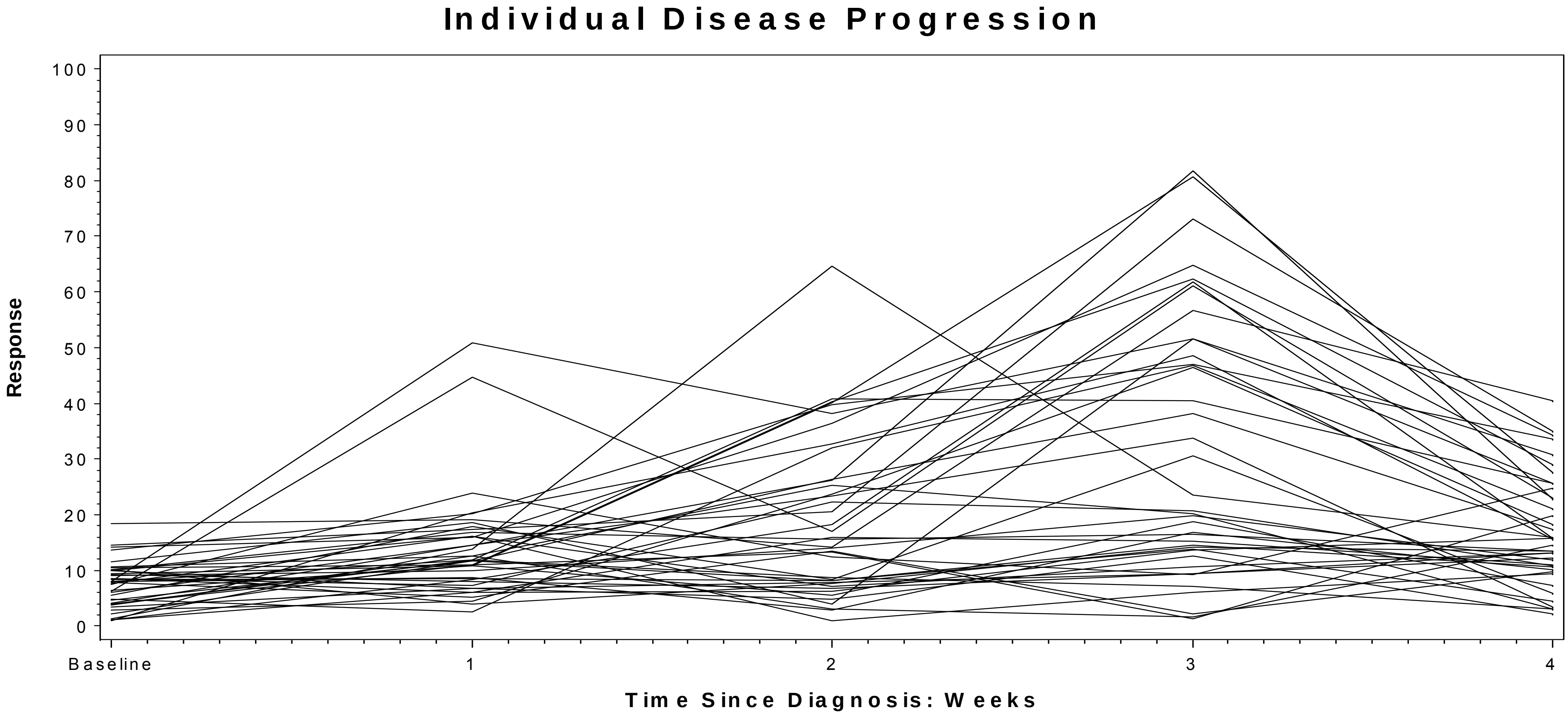
Repeated Measures/Longitudinal Plotting



```
proc gplot data=long;
     plot y*time=id / nolegend haxis=axis1 vaxis=axis2;
     symbol1 c=black i=join r=40;
run;
```

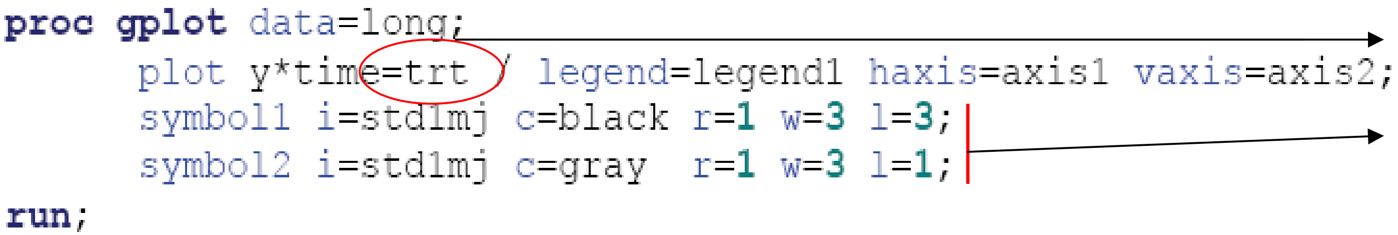
Joins the dots,

By ID



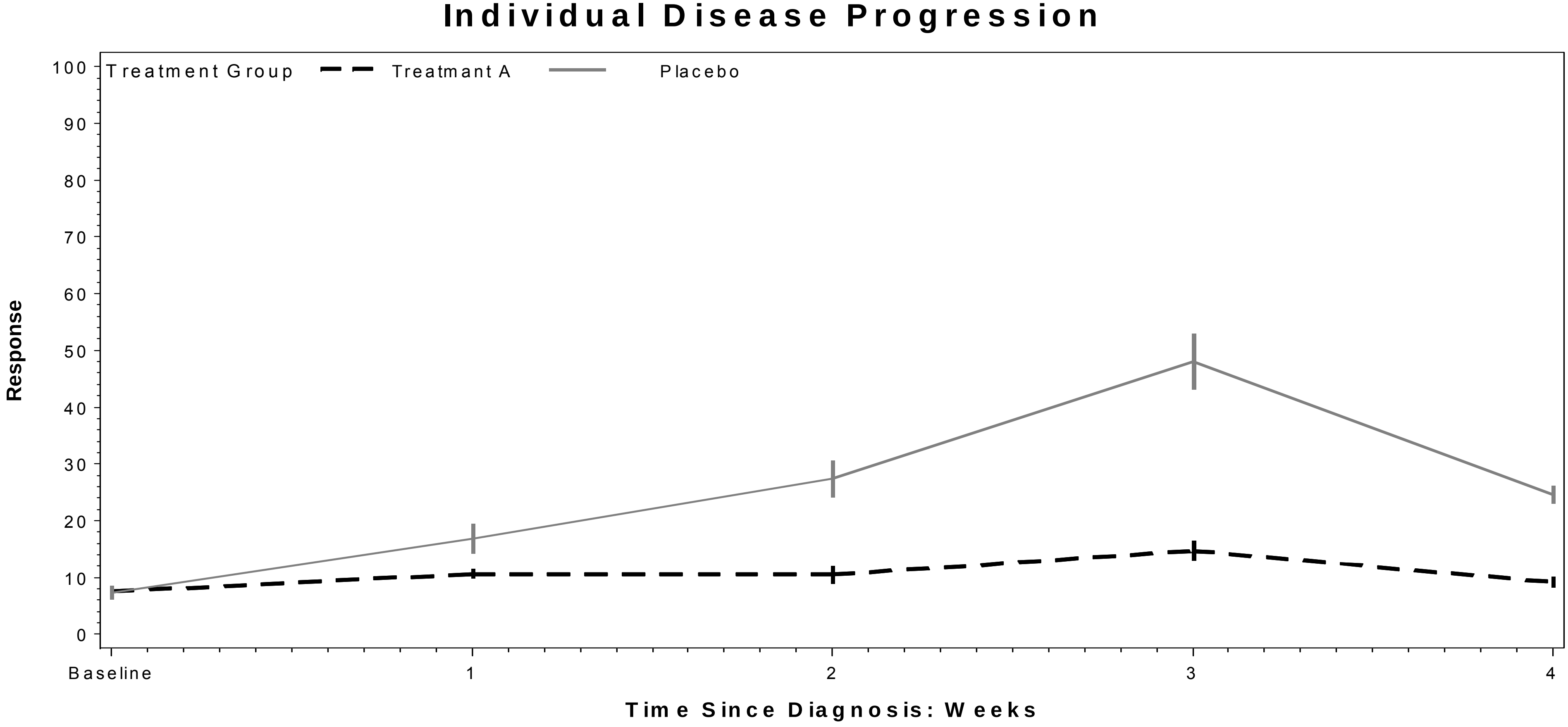
Examples

Repeated Measures/Longitudinal Plotting



```
proc gplot data=long;
      plot y*time=trt / legend=legend1 haxis=axis1 vaxis=axis2;
      symbol1 i=std1mj c=black r=1 w=3 l=3;
      symbol2 i=std1mj c=gray  r=1 w=3 l=1;
run;
```

Plot data by trt group and create a symbol statement for each group



Examples

Using the Overlay statement to stack plots

Examples

Overlay 2 plots w/ the same data

Suppose that you are asked to graphically show progression of tumor growth for a group of subjects and overlay the progression of each treatment group.

50 subjects randomized to either low or high dose medication.

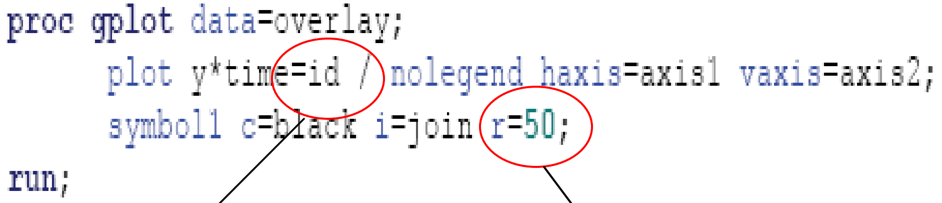
Tumor size is measured at baseline as well as the following 9 weeks.

The investigator would like an easy to present plot containing both pieces of information for a presentation to his peers.

Examples

Overlay 2 plots w/ the same data

Plot of individual values as before

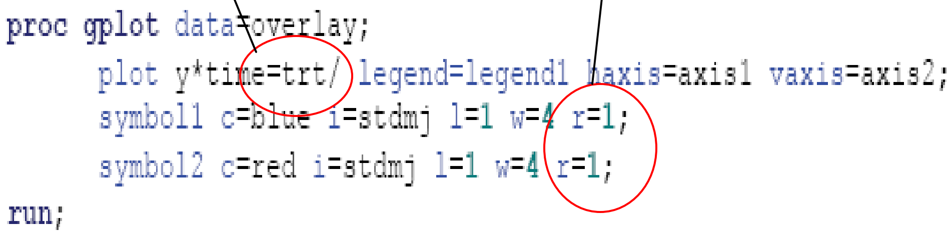


```
proc gplot data=overlay;
     plot y*time=id / nolegend haxis=axis1 vaxis=axis2;
     symbol1 c=black i=join r=50;
run;
```

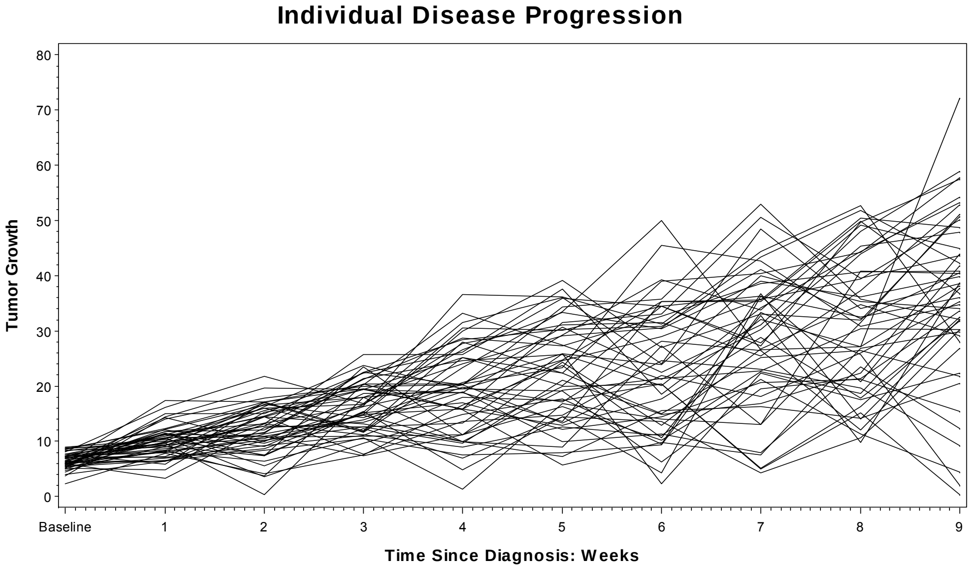
Grouping variable

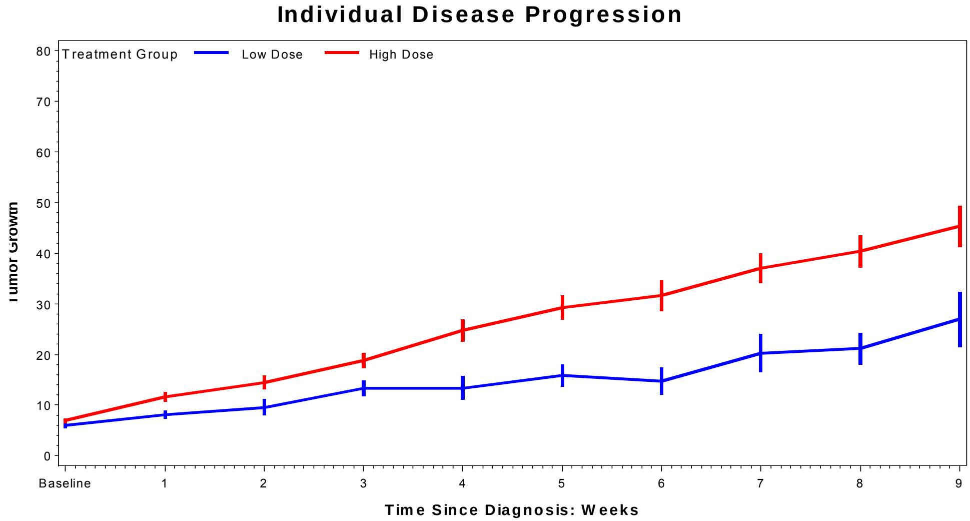
Symbol repeats

Plot of treatment group means and Standard errors as before



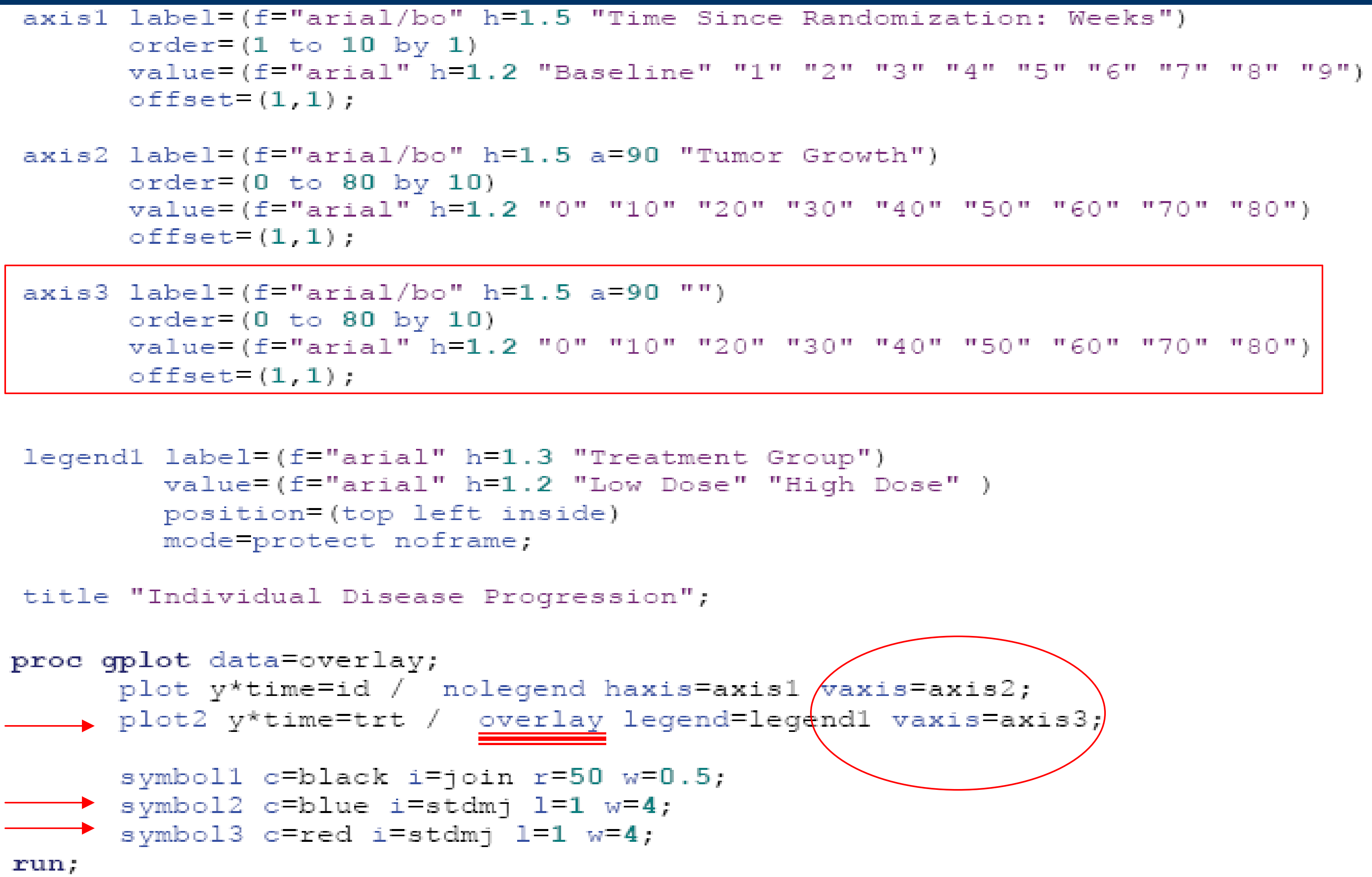
```
proc gplot data=overlay;
     plot y*time=trt/ legend=legend1 haxis=axis1 vaxis=axis2;
     symbol1 c=blue i=stdmj l=1 w=4 r=1;
     symbol2 c=red i=stdmj l=1 w=4 r=1;
run;
```





Examples

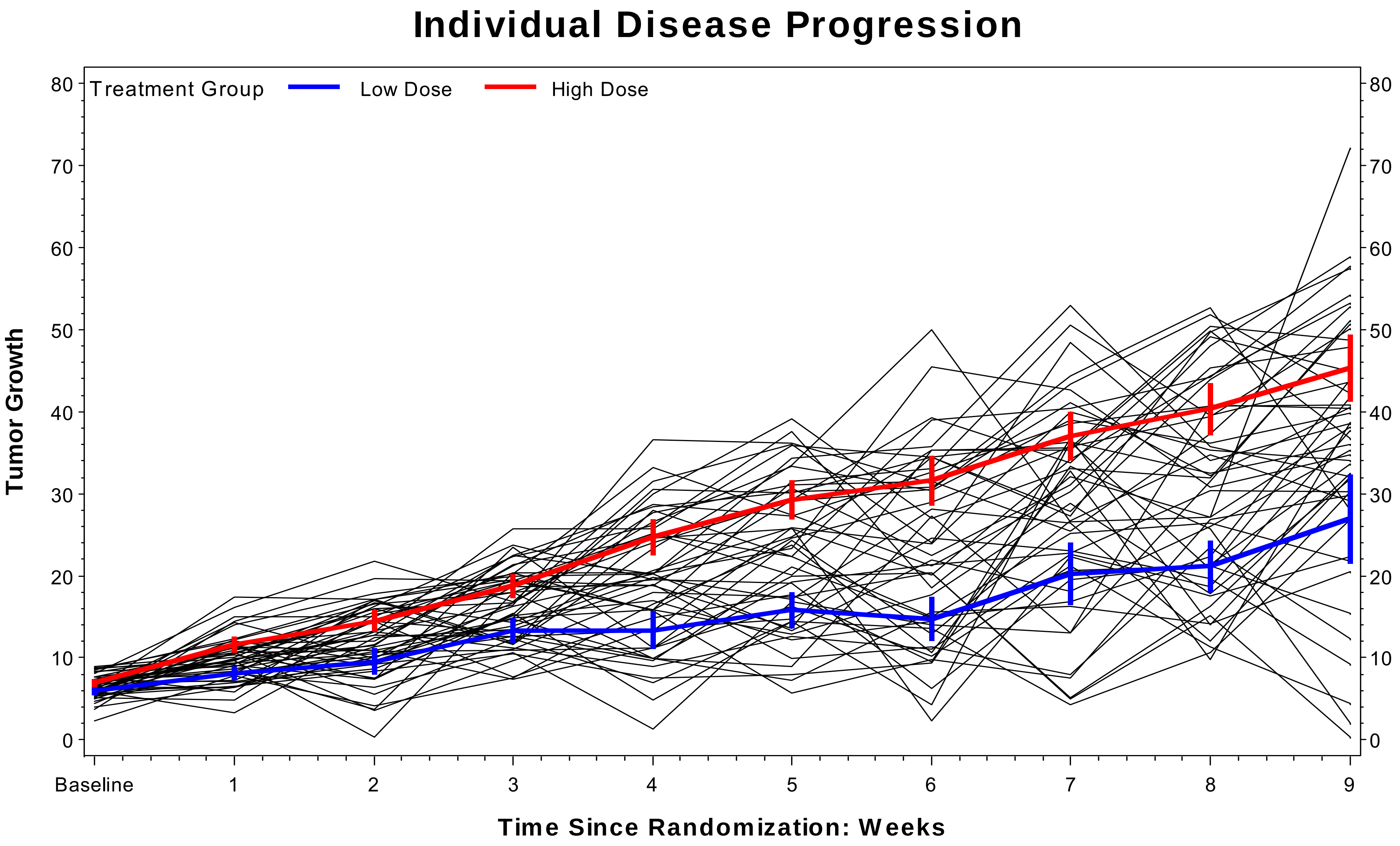
Overlay 2 plots w/ the same data



```
axis1 label=(f="arial/bo" h=1.5 "Time Since Randomization: Weeks")
      order=(1 to 10 by 1)
      value=(f="arial" h=1.2 "Baseline" "1" "2" "3" "4" "5" "6" "7" "8" "9")
      offset=(1,1);

axis2 label=(f="arial/bo" h=1.5 a=90 "Tumor Growth")
      order=(0 to 80 by 10)
      value=(f="arial" h=1.2 "0" "10" "20" "30" "40" "50" "60" "70" "80")
      offset=(1,1);

axis3 label=(f="arial/bo" h=1.5 a=90 "")
      order=(0 to 80 by 10)
      value=(f="arial" h=1.2 "0" "10" "20" "30" "40" "50" "60" "70" "80")
      offset=(1,1);


legend1 label=(f="arial" h=1.3 "Treatment Group")
        value=(f="arial" h=1.2 "Low Dose" "High Dose" )
        position=(top left inside)
        mode=protect noframe;

title "Individual Disease Progression";

proc gplot data=overlay;
      plot y*time=id /  nolegend haxis=axis1 vaxis=axis2;
      plot2 y*time=trt /  overlay legend=legend1 vaxis=axis3;

      symbol1 c=black i=join r=50 w=0.5;
      symbol2 c=blue i=stdmj l=1 w=4;
      symbol3 c=red i=stdmj l=1 w=4;
run;
```

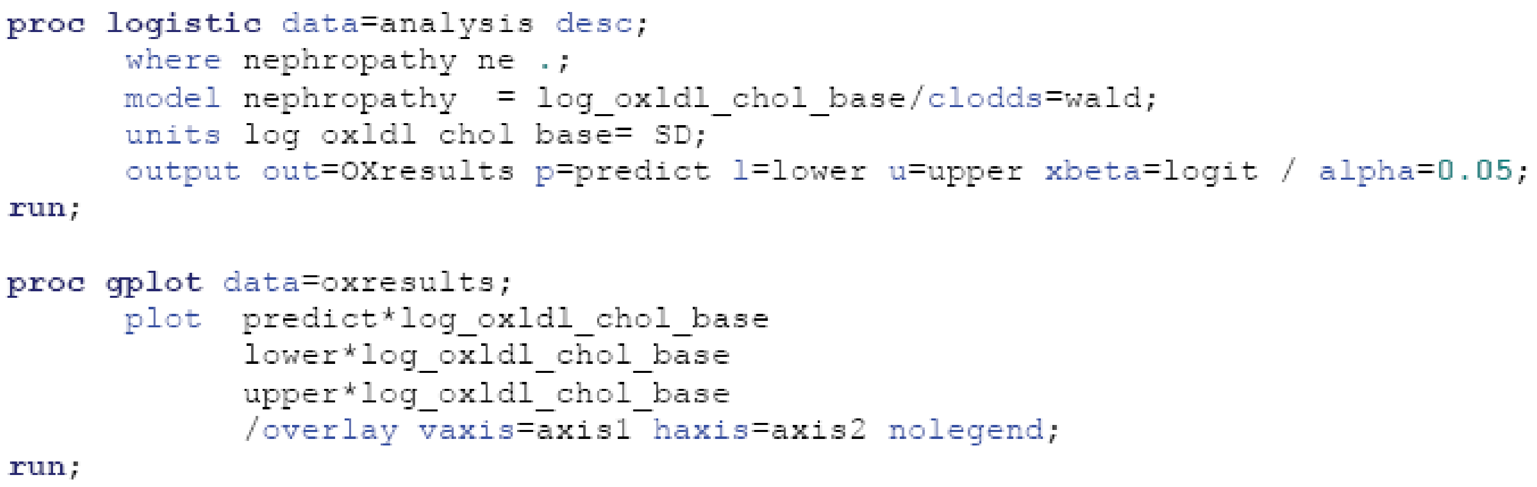
Examples

Overlay 2 plots w/ the same data



Examples

Overlay multiple plots from different variables



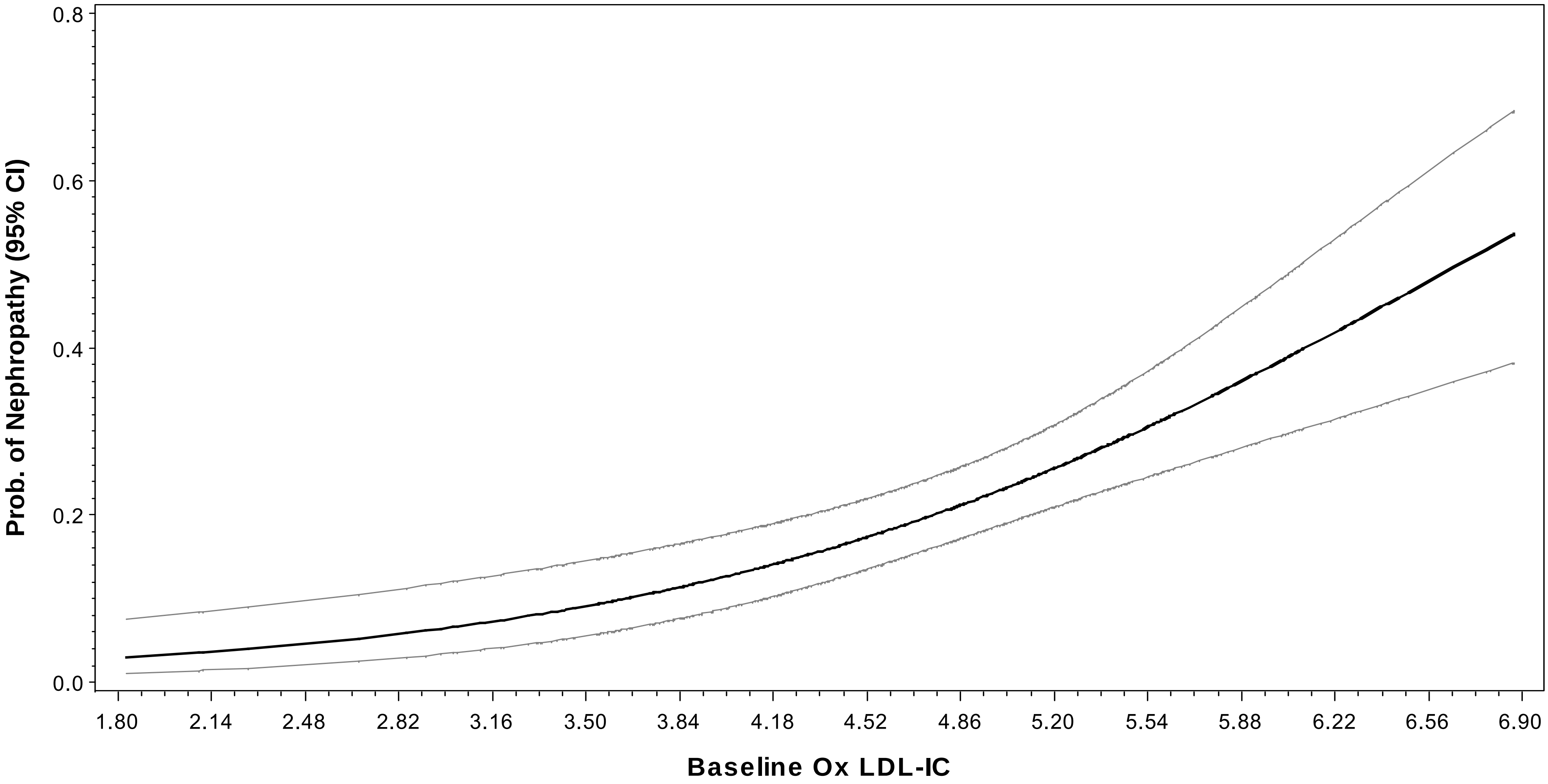
```
proc logistic data=analysis desc;
      where nephropathy ne .;
      model nephropathy  = log_oxldl_chol_base/clodds=wald;
      units log oxldl chol base= SD;
      output out=OXresults p=predict l=lower u=upper xbeta=logit / alpha=0.05;
run;

proc gplot data=oxresults;
      plot  predict*log_oxldl_chol_base
            lower*log_oxldl_chol_base
            upper*log_oxldl_chol_base
            /overlay vaxis=axis1 haxis=axis2 nolegend;
run;
```

Use proc logistic to output the predicted probability of developing nephropathy given the baseline Oxidized LDL immune complex level as well as the 95% confidence limits.

Many PROCs can output predicted values, adjusted means, along with point wise confidence values.

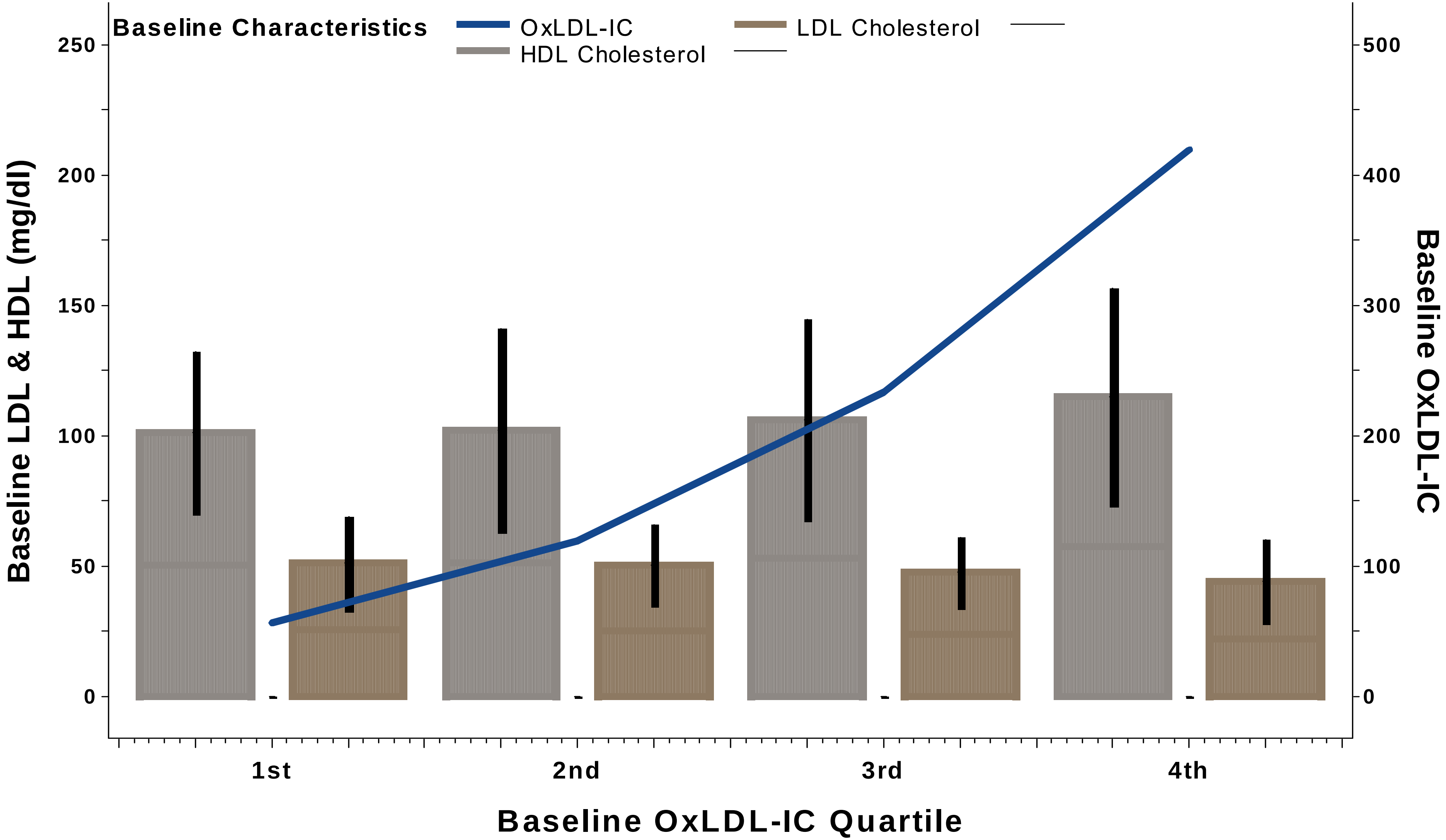
Examples

Overlay multiple plots from different variables



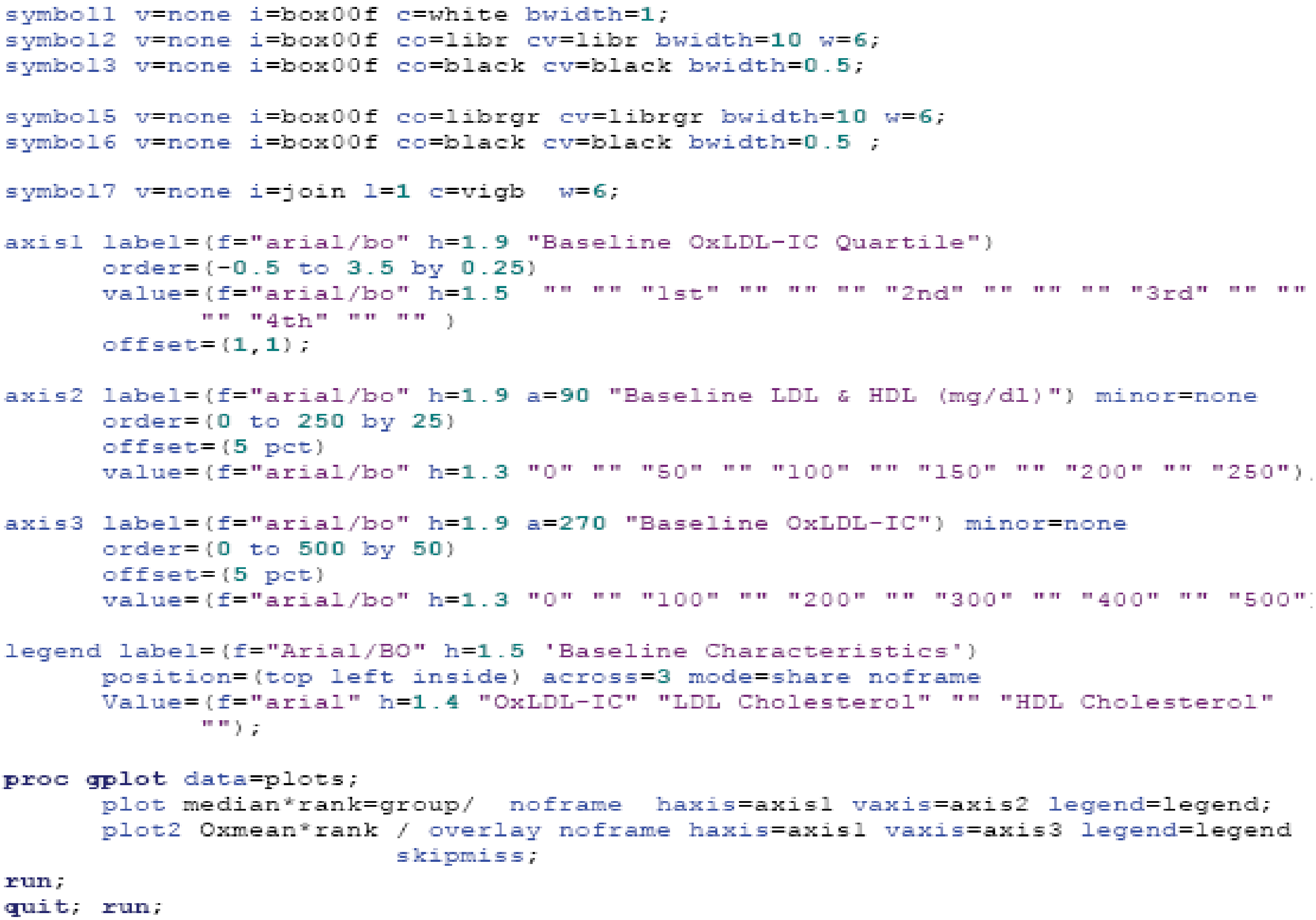
Examples

Overlay multiple plots from different variables



Examples

Overlay multiple plots from different variables



```
symbol1 v=none i=box00f c=white bwidth=1;
symbol2 v=none i=box00f co=libr cv=libr bwidth=10 w=6;
symbol3 v=none i=box00f co=black cv=black bwidth=0.5;

symbol5 v=none i=box00f co=librgr cv=librgr bwidth=10 w=6;
symbol6 v=none i=box00f co=black cv=black bwidth=0.5 ;

symbol7 v=none i=join l=1 c=vigb  w=6;

axis1 label=(f="arial/bo" h=1.9 "Baseline OxLDL-IC Quartile")
      order=(-0.5 to 3.5 by 0.25)
      value=(f="arial/bo" h=1.5  "" "" "1st" "" "" "" "2nd" "" "" "" "3rd" "" ""
            "" "4th" "" "" )
      offset=(1,1);

axis2 label=(f="arial/bo" h=1.9 a=90 "Baseline LDL & HDL (mg/dl)") minor=none
      order=(0 to 250 by 25)
      offset=(5 pct)
      value=(f="arial/bo" h=1.3 "0" "" "50" "" "100" "" "150" "" "200" "" "250")

axis3 label=(f="arial/bo" h=1.9 a=270 "Baseline OxLDL-IC") minor=none
      order=(0 to 500 by 50)
      offset=(5 pct)
      value=(f="arial/bo" h=1.3 "0" "" "100" "" "200" "" "300" "" "400" "" "500")

legend label=(f="Arial/BO" h=1.5 'Baseline Characteristics')
      position=(top left inside) across=3 mode=share noframe
      Value=(f="arial" h=1.4 "OxLDL-IC" "LDL Cholesterol" "" "HDL Cholesterol"
            "");

proc gplot data=plots;
      plot median*rank=group/  noframe  haxis=axis1 vaxis=axis2 legend=legend;
      plot2 Oxmean*rank / overlay noframe haxis=axis1 vaxis=axis3 legend=legend
                          skipmiss;
run;
quit; run;
```

The Annotate Facility

The Annotate Facility

Introduction

The Annotate Facility allows SAS users to customize graphical output. The customizations can be data driven or user specified. Text, shapes, lines and images can be added to output graphics

Step 1. Create an annotate data set

this data file will give commands to SAS/GRAPH

Specific variables must be in the annotate data set. Others are allowed but ignored

What, how, and where are defined by these variables.

Table 1 list important variables.

The Annotate Facility

Introduction

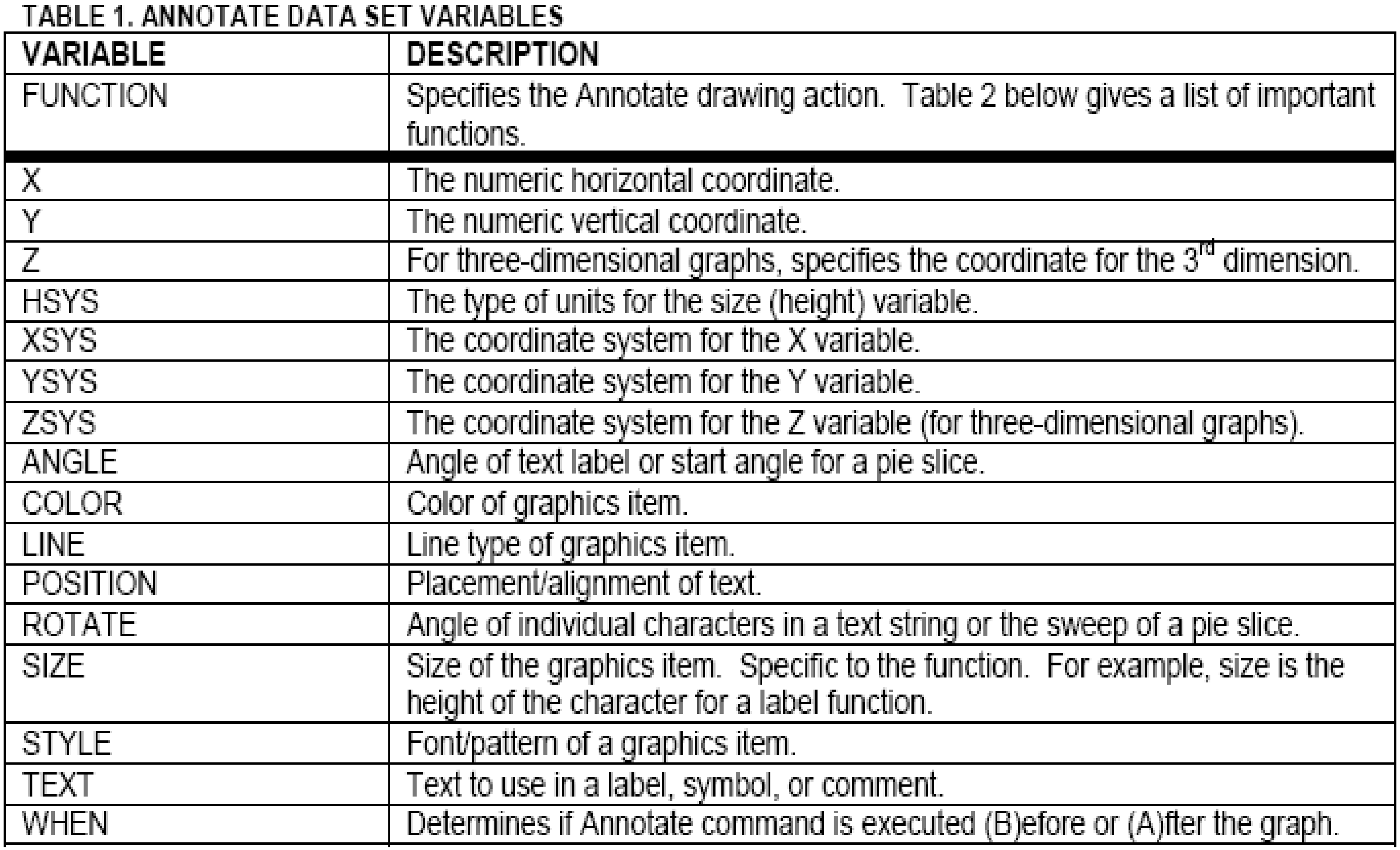


TABLE 1. ANNOTATE DATA SET VARIABLES

VARIABLE	DESCRIPTION
FUNCTION	Specifies the Annotate drawing action. Table 2 below gives a list of important functions.
X	The numeric horizontal coordinate.
Y	The numeric vertical coordinate.
Z	For three-dimensional graphs, specifies the coordinate for the 3^{rd} dimension.
HSYS	The type of units for the size (height) variable.
XSYS	The coordinate system for the X variable.
YSYS	The coordinate system for the Y variable.
ZSYS	The coordinate system for the Z variable (for three-dimensional graphs).
ANGLE	Angle of text label or start angle for a pie slice.
COLOR	Color of graphics item.
LINE	Line type of graphics item.
POSITION	Placement/alignment of text.
ROTATE	Angle of individual characters in a text string or the sweep of a pie slice.
SIZE	Size of the graphics item. Specific to the function. For example, size is the height of the character for a label function.
STYLE	Font/pattern of a graphics item.
TEXT	Text to use in a label, symbol, or comment.
WHEN	Determines if Annotate command is executed (B)efore or (A)fter the graph.

The Annotate Facility

Introduction

The Annotate FUNCTION variable tells SAS what to do

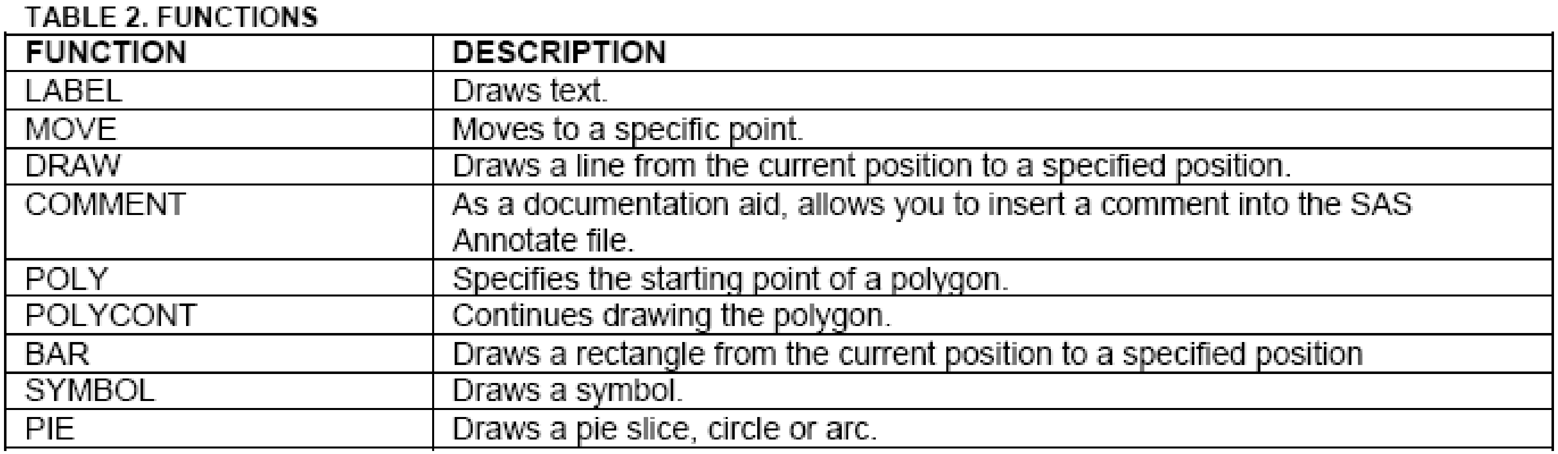


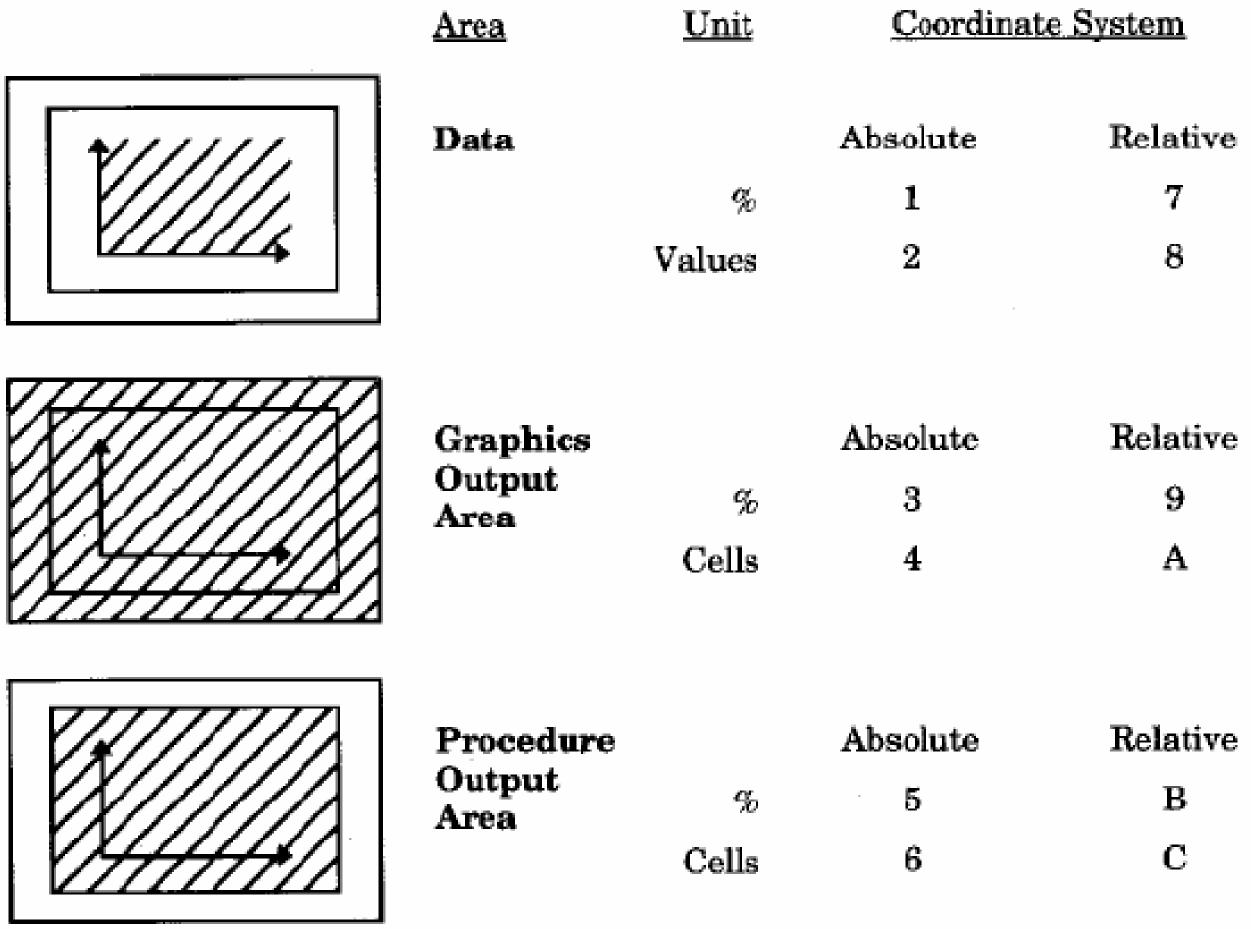
TABLE 2. FUNCTIONS

FUNCTION	DESCRIPTION
LABEL	Draws text.
MOVE	Moves to a specific point.
DRAW	Draws a line from the current position to a specified position.
COMMENT	As a documentation aid, allows you to insert a comment into the SAS Annotate file.
POLY	Specifies the starting point of a polygon.
POLYCONT	Continues drawing the polygon.
BAR	Draws a rectangle from the current position to a specified position
SYMBOL	Draws a symbol.
PIE	Draws a pie slice, circle or arc.

The annotate coordinate system allows for flexibility in placing objects within the output. There are 12 possible conditions.

The Annotate Facility

Introduction



Area	Unit	Coordinate System	
		Absolute	Relative
Data	%	1	7
	Values	2	8
Graphics Output Area	%	3	9
	Cells	4	A
Procedure Output Area	%	5	B
	Cells	6	C

Taken from Mink and Pasta, SUGI 31

The Annotate Facility

Introduction

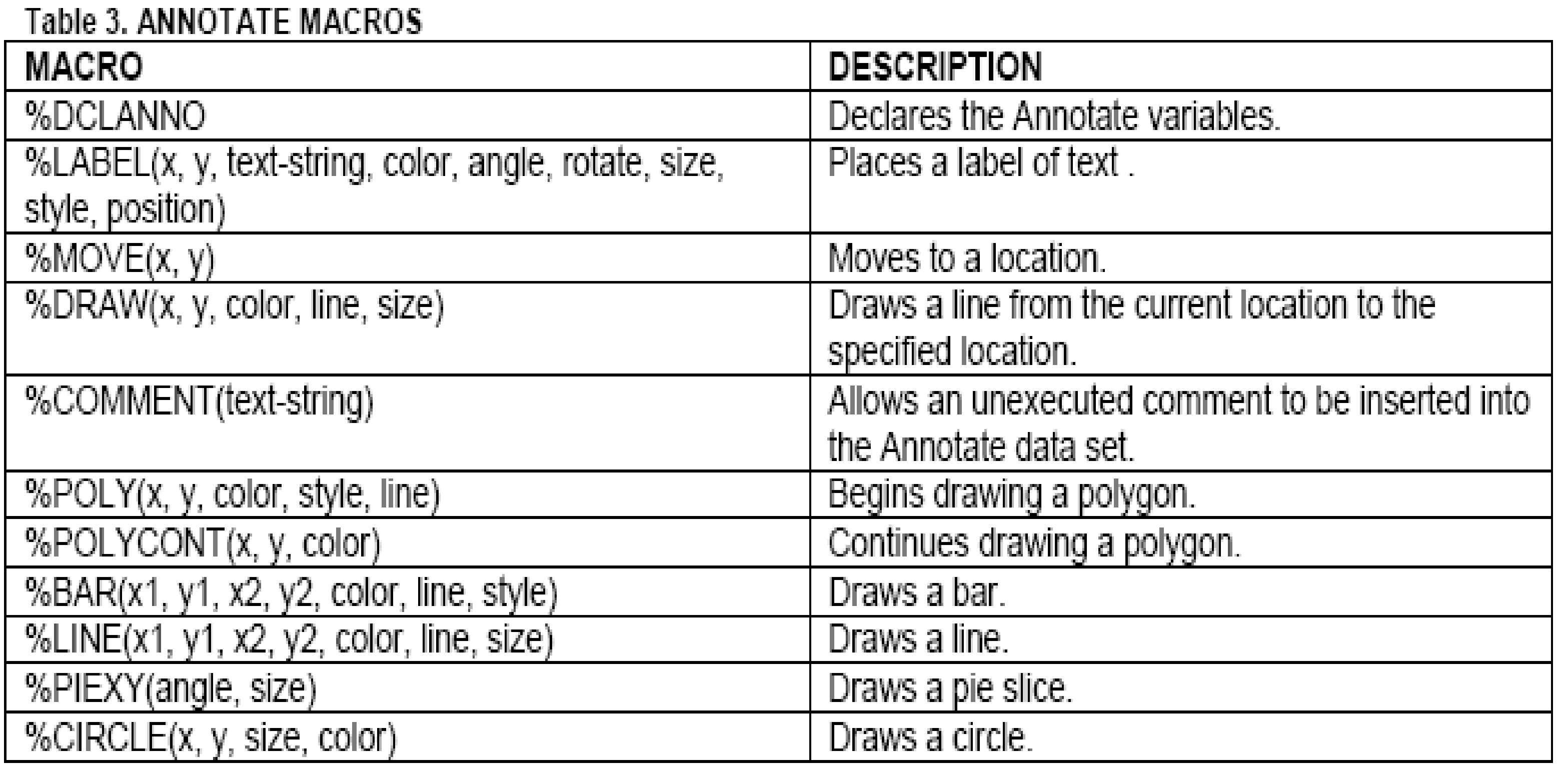
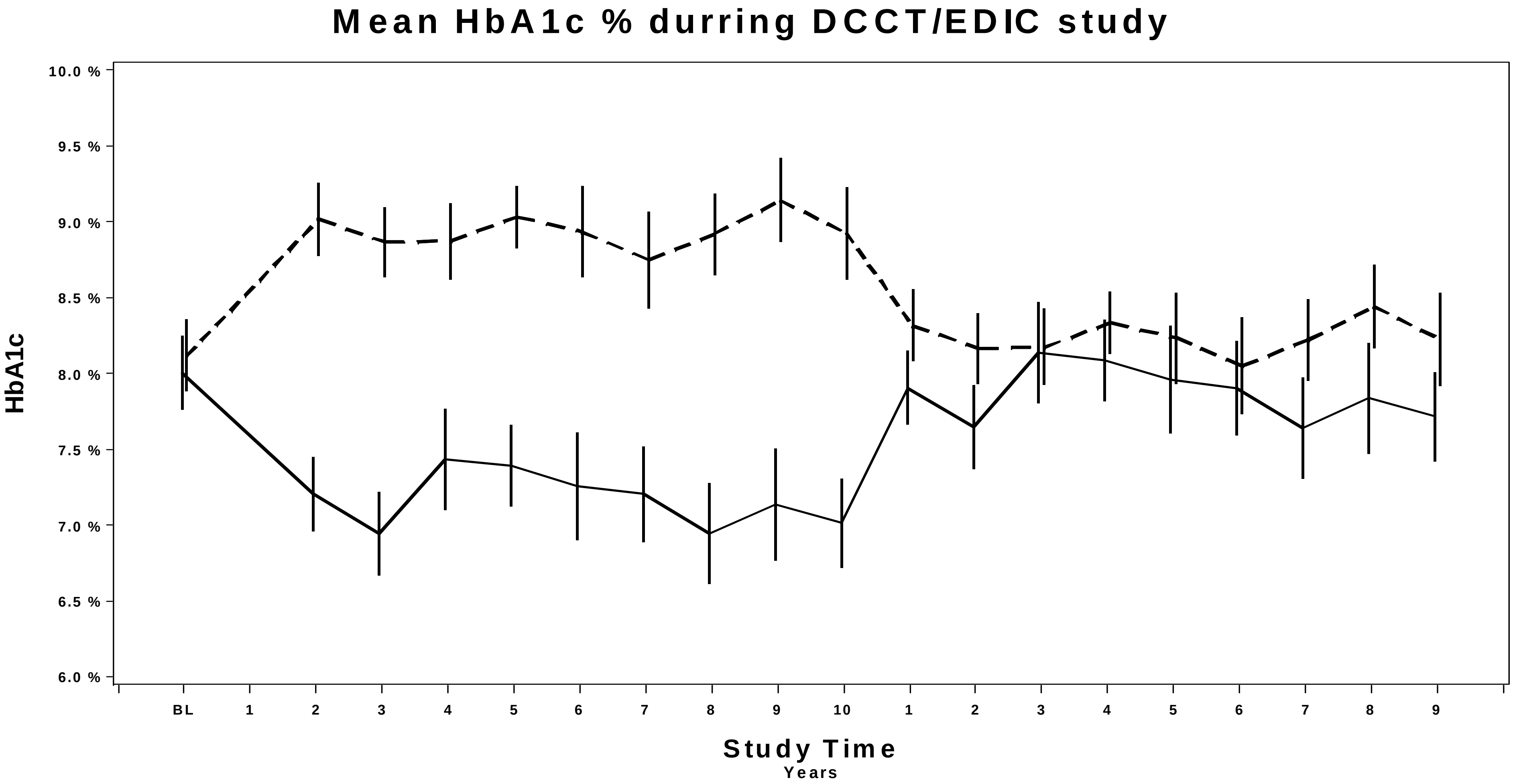


Table 3. ANNOTATE MACROS

MACRO	DESCRIPTION
%DCLANNO	Declares the Annotate variables.
%LABEL(x, y, text-string, color, angle, rotate, size, style, position)	Places a label of text .
%MOVE(x, y)	Moves to a location.
%DRAW(x, y, color, line, size)	Draws a line from the current location to the specified location.
%COMMENT(text-string)	Allows an unexecuted comment to be inserted into the Annotate data set.
%POLY(x, y, color, style, line)	Begins drawing a polygon.
%POLYCONT(x, y, color)	Continues drawing a polygon.
%BAR(x1, y1, x2, y2, color, line, style)	Draws a bar.
%LINE(x1, y1, x2, y2, color, line, size)	Draws a line.
%PIEXY(angle, size)	Draws a pie slice.
%CIRCLE(x, y, size, color)	Draws a circle.

The Annotate Facility

Proc GPLOT global options help make graphs more pleasing, however, there are cases where more work is needed to fully explain the data



The Annotate Facility

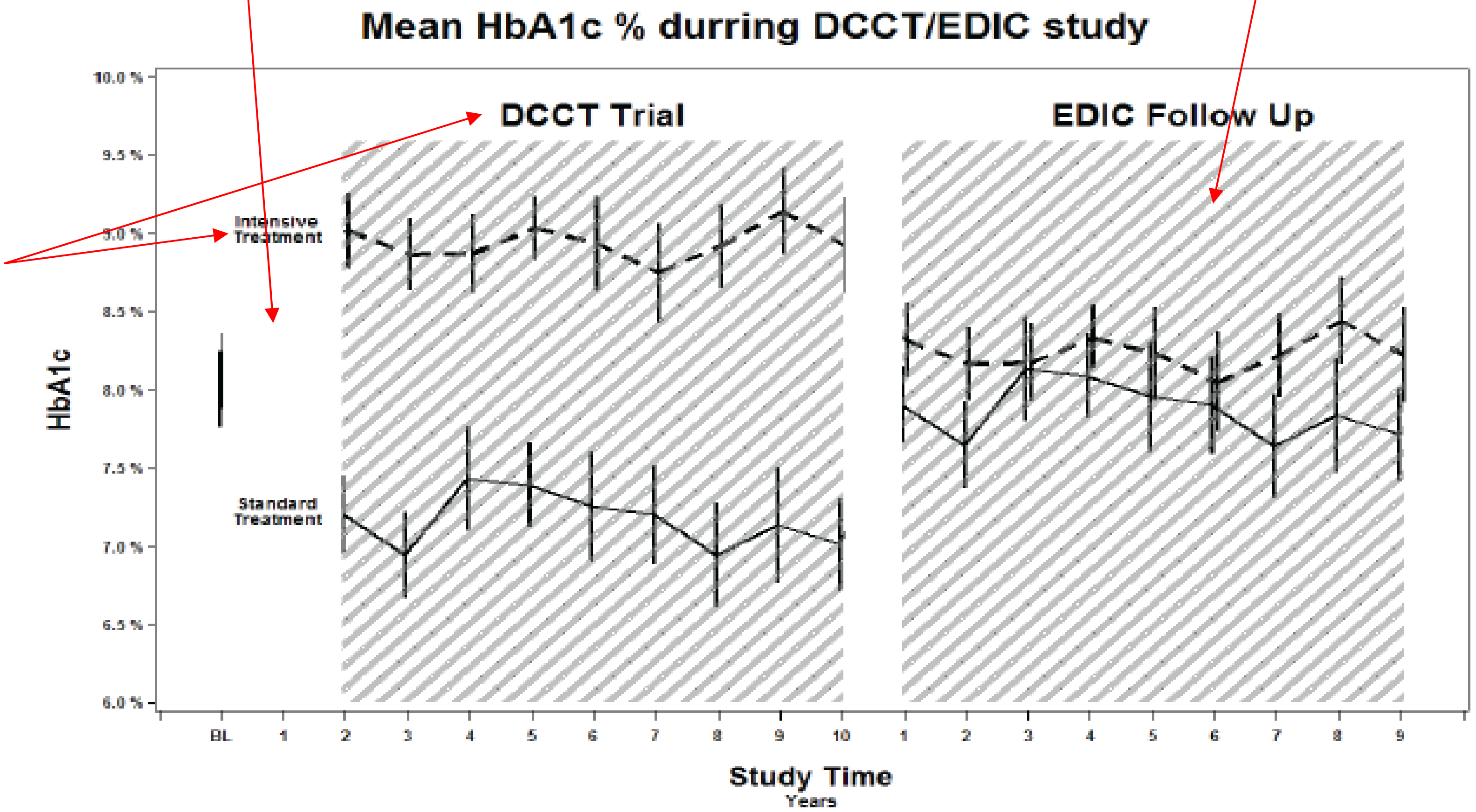
```
%annomac;
data anno bar;
      %dclanno; length text $30;
      xsys='2'; ysys='2'; hsys='2';
      when='A';

      %bar(1.021, 6, 2.935, 10, white, 3, solid);
      %bar(11.055, 6, 11.940, 10, white, 3, solid);
      %bar(2.935, 6, 11.02,9.6, CX808080, 3, r5);
      %bar(11.95, 6, 20.07,9.6, CX808080, 3, r5);

      function='label'; color='black'; x=1.9; y=9.1;  style='ARIAL/bo';
            text='Intensive';output;
      function='label'; color='black'; x=1.9; y=9.0;  style='ARIAL/bo';
            text='Treatment';output;

      function='label'; color='black'; x=1.9; y=7.3;  style='ARIAL/bo';
            text='Standard';output;
      function='label'; color='black'; x=1.9; y=7.2;  style='ARIAL/bo';
            text='Treatment';output;

      function='label'; color='black'; x=7; y=9.8; size=0.19; style='ARIAL/bo';
            text='DCCT Trial';output;
      function='label'; color='black'; x=16.5; y=9.8; size=0.19; style='ARIAL/bo';
            text='EDIC Follow Up';output;
run;
```

The Annotate Facility

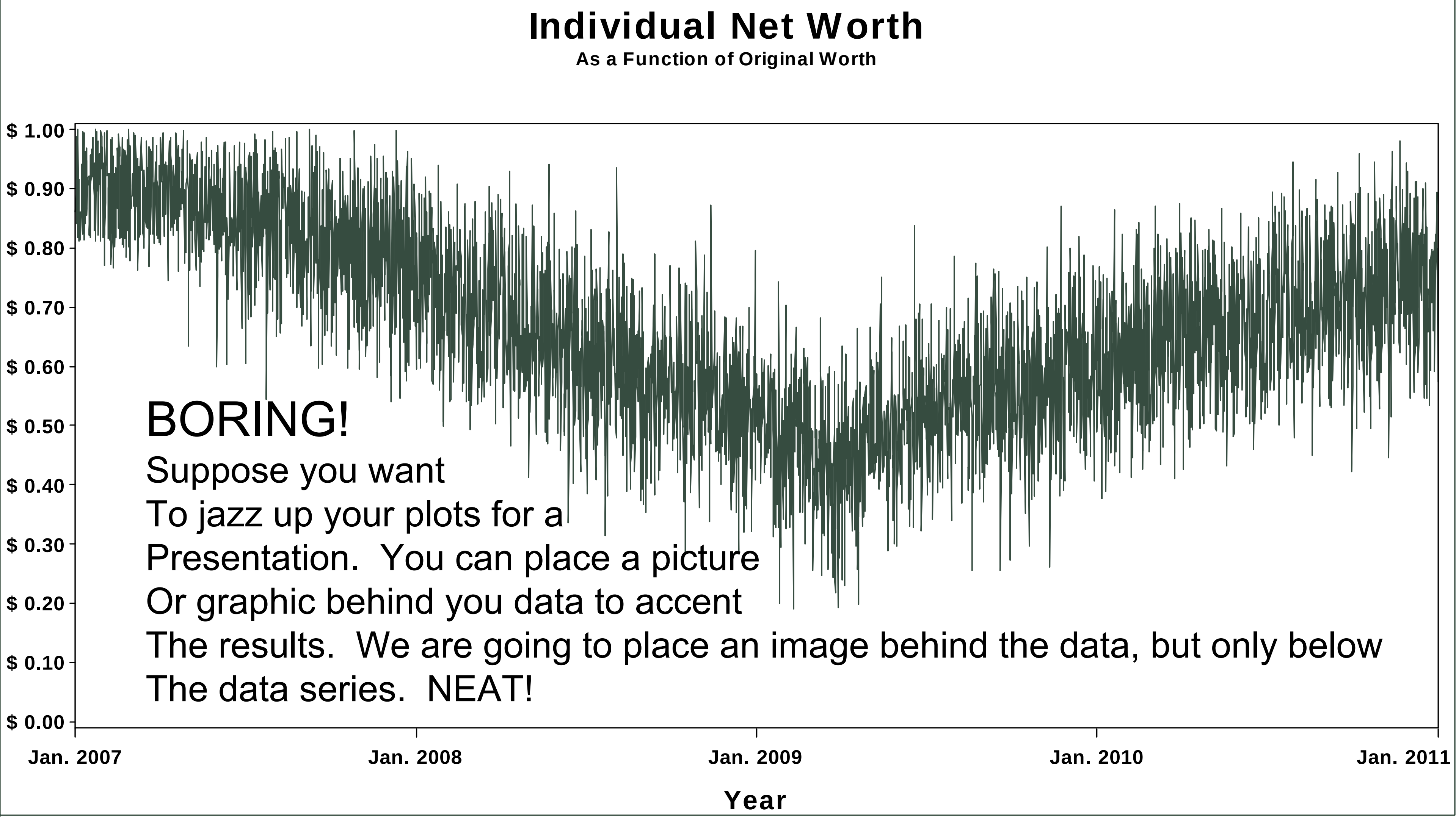
Created shaded regions to designate study sections

Deleted regions of non interest

Added treatment group and study section labels



The Annotate Facility

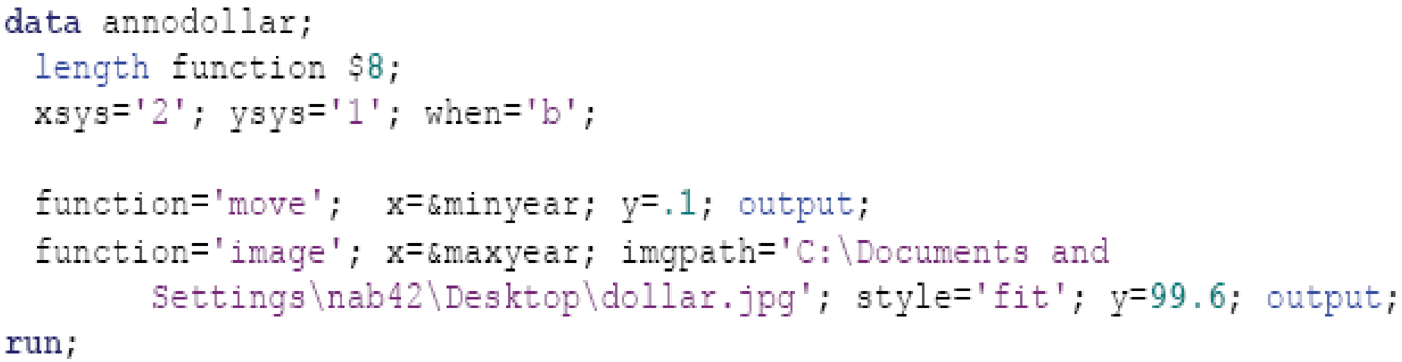


BORING!

Suppose you want
To jazz up your plots for a
Presentation. You can place a picture
Or graphic behind you data to accent
The results. We are going to place an image behind the data, but only below
The data series. NEAT!

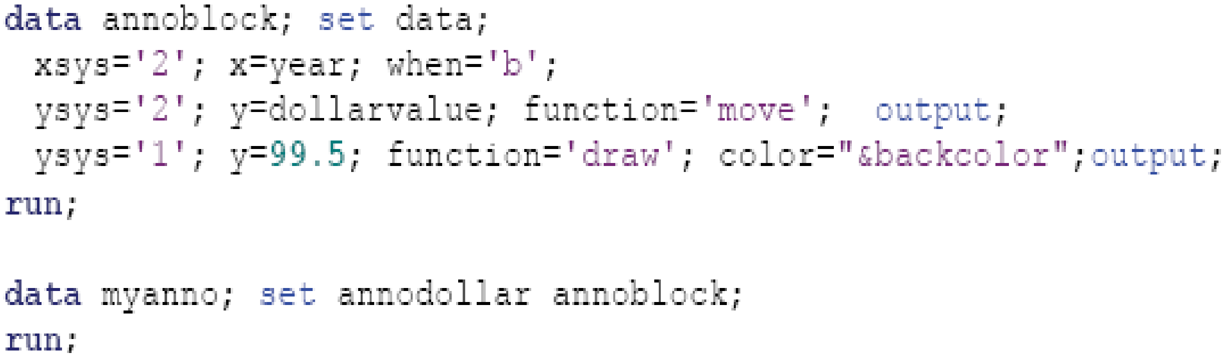
The Annotate Facility

Anno data set 1:
Will place the
image
of the dollar over
the plotting area.



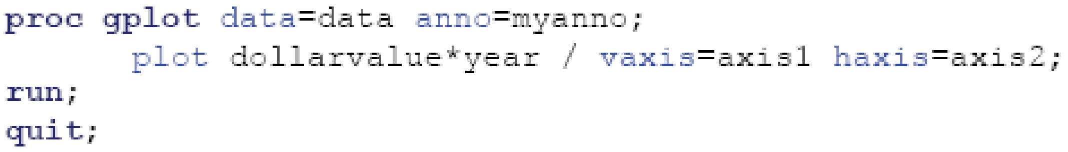
```
data annodollar;
 length function $8;
 xsys='2'; ysys='1'; when='b';

 function='move';  x=&minyear; y=.1; output;
 function='image'; x=&maxyear; imgpath='C:\Documents and
       Settings\nab42\Desktop\dollar.jpg'; style='fit'; y=99.6; output;
run;
```

Anno data set 2:
Will create white
Space above the
Plotted line over
time.



```
data annoblock; set data;
 xsys='2'; x=year; when='b';
 ysys='2'; y=dollarvalue; function='move';  output;
 ysys='1'; y=99.5; function='draw'; color="&backcolor";output;
run;

data myanno; set annodollar annoblock;
run;
```

SET the anno data
sets and call them
in the GPLOT
statement

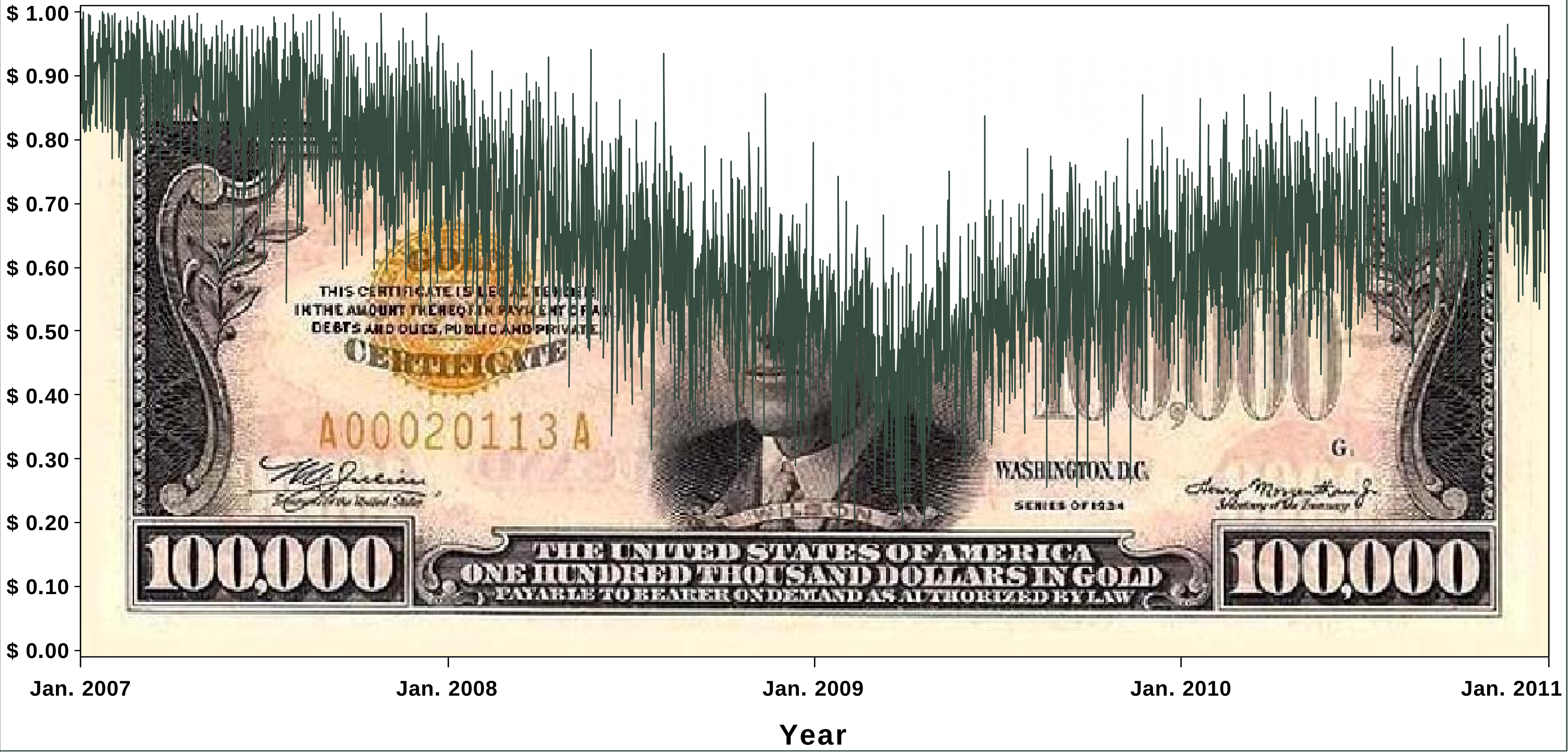


```
proc gplot data=data anno=myanno;
      plot dollarvalue*year / vaxis=axis1 haxis=axis2;
run;
quit;
```

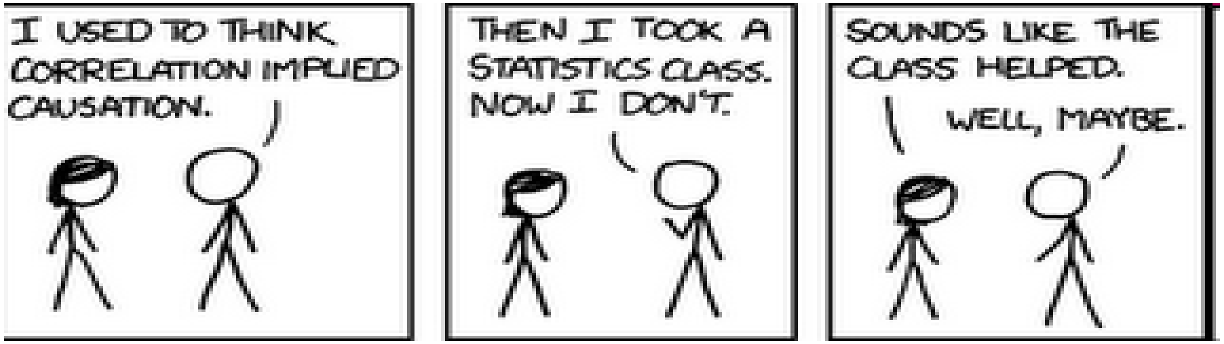
The Annotate Facility

Individual Net Worth

As a Function of Original Worth



The End



I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.