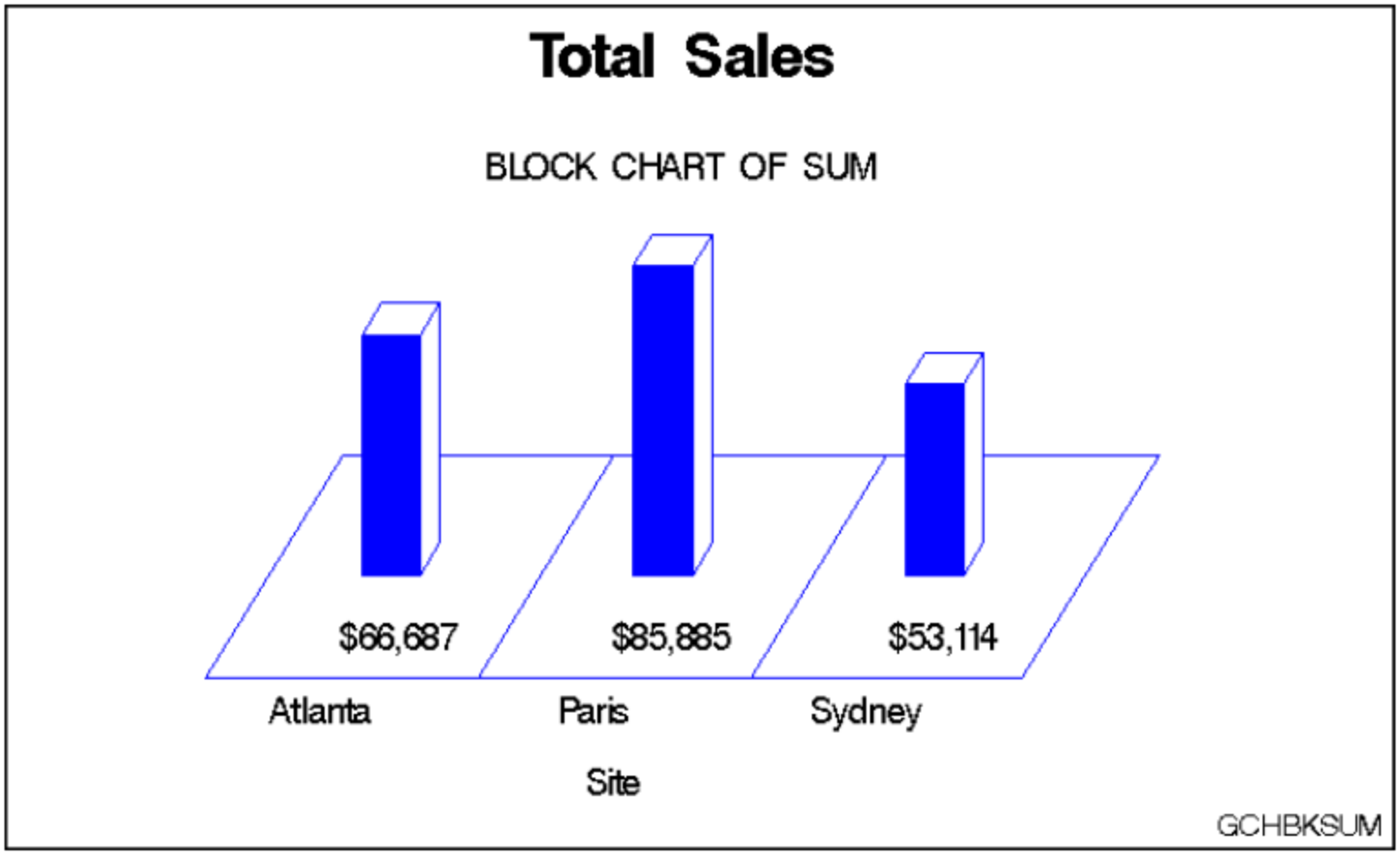
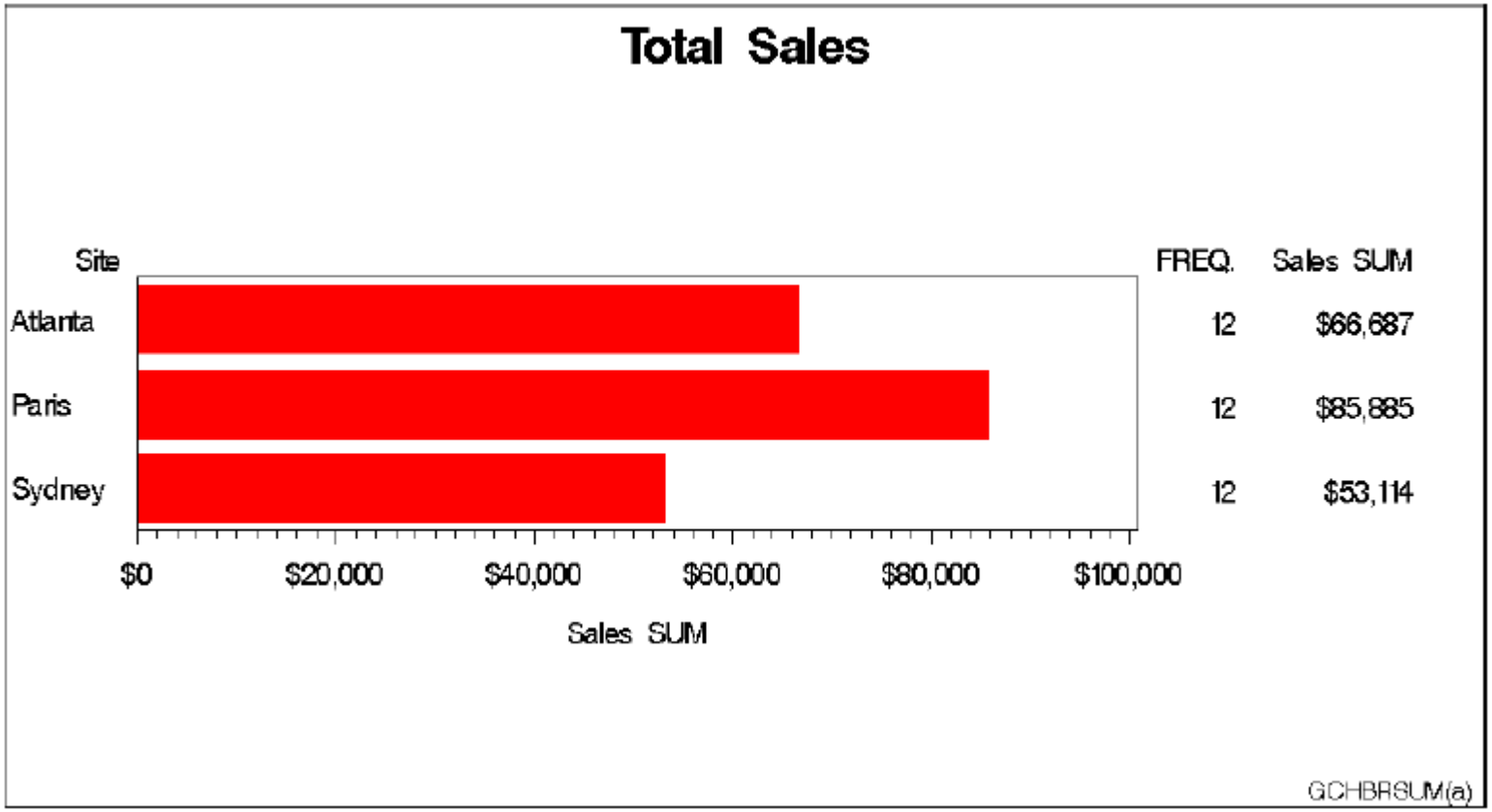
临床研究SAS高级编程
——Graph

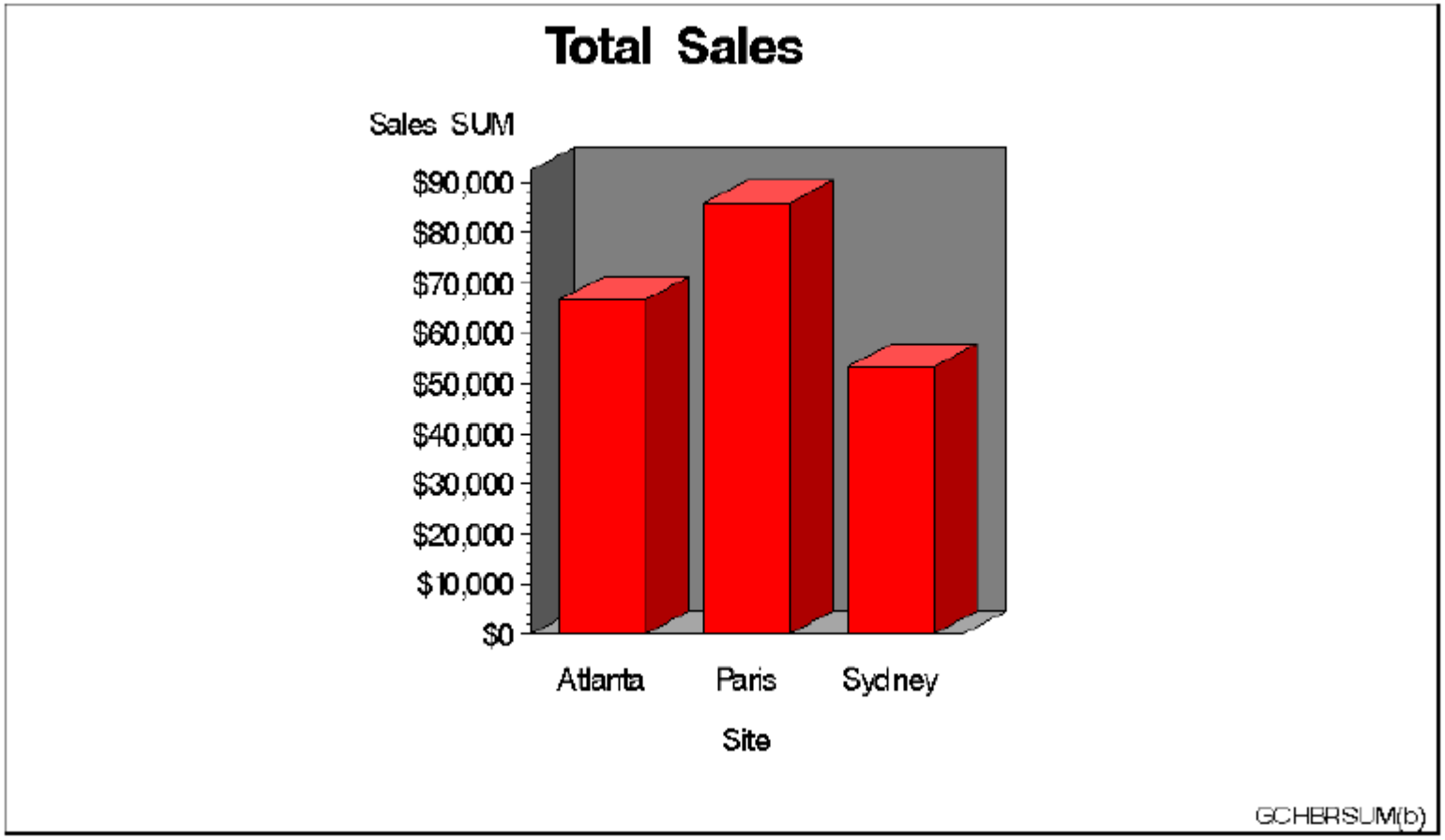
Block Chart


Total Sales
BLOCK CHART OF SUM
$66,687
$85,885
$53,114
Atlanta
Paris
Sydney
Site
GCHBKSUM

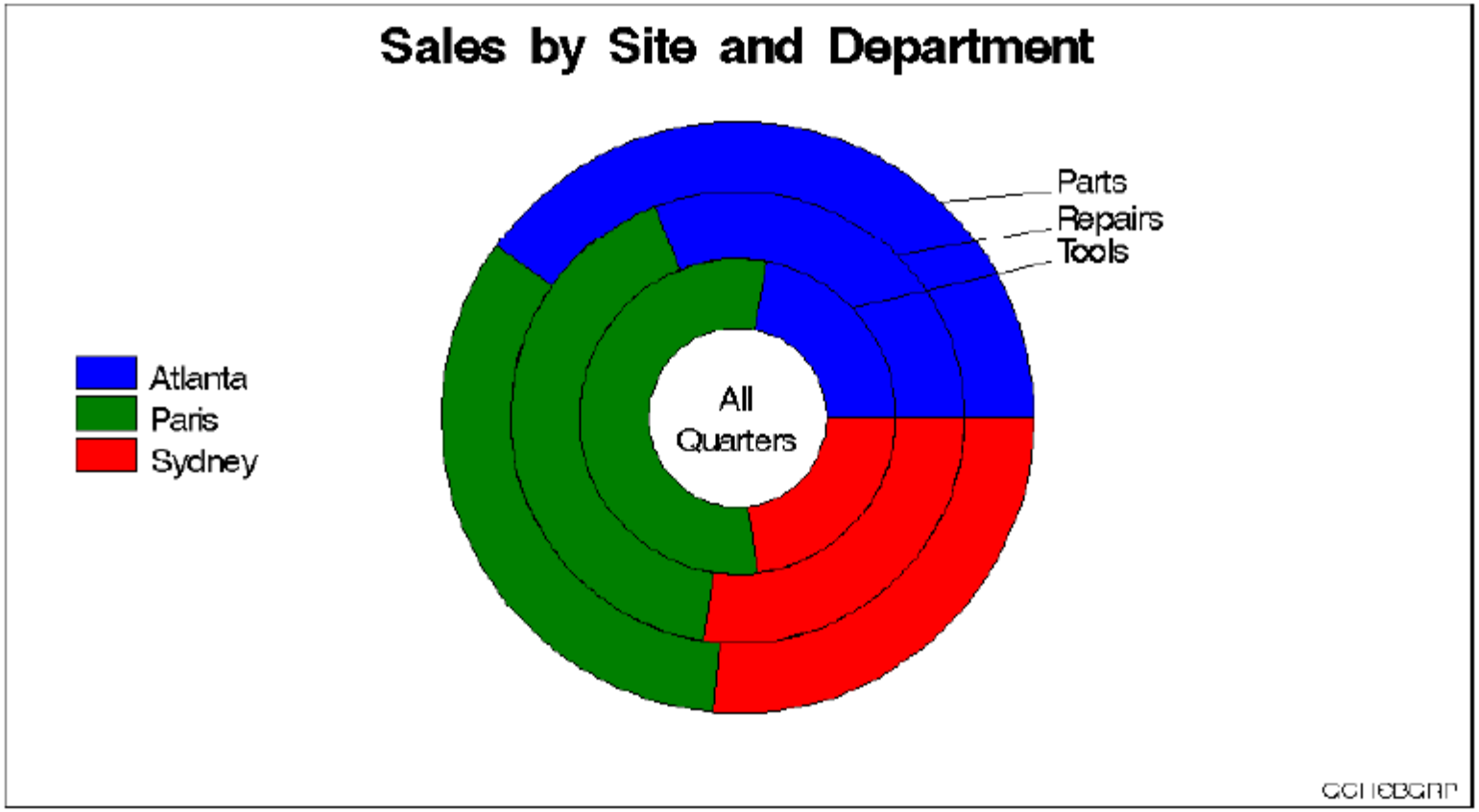
Horizontal Bar Chart



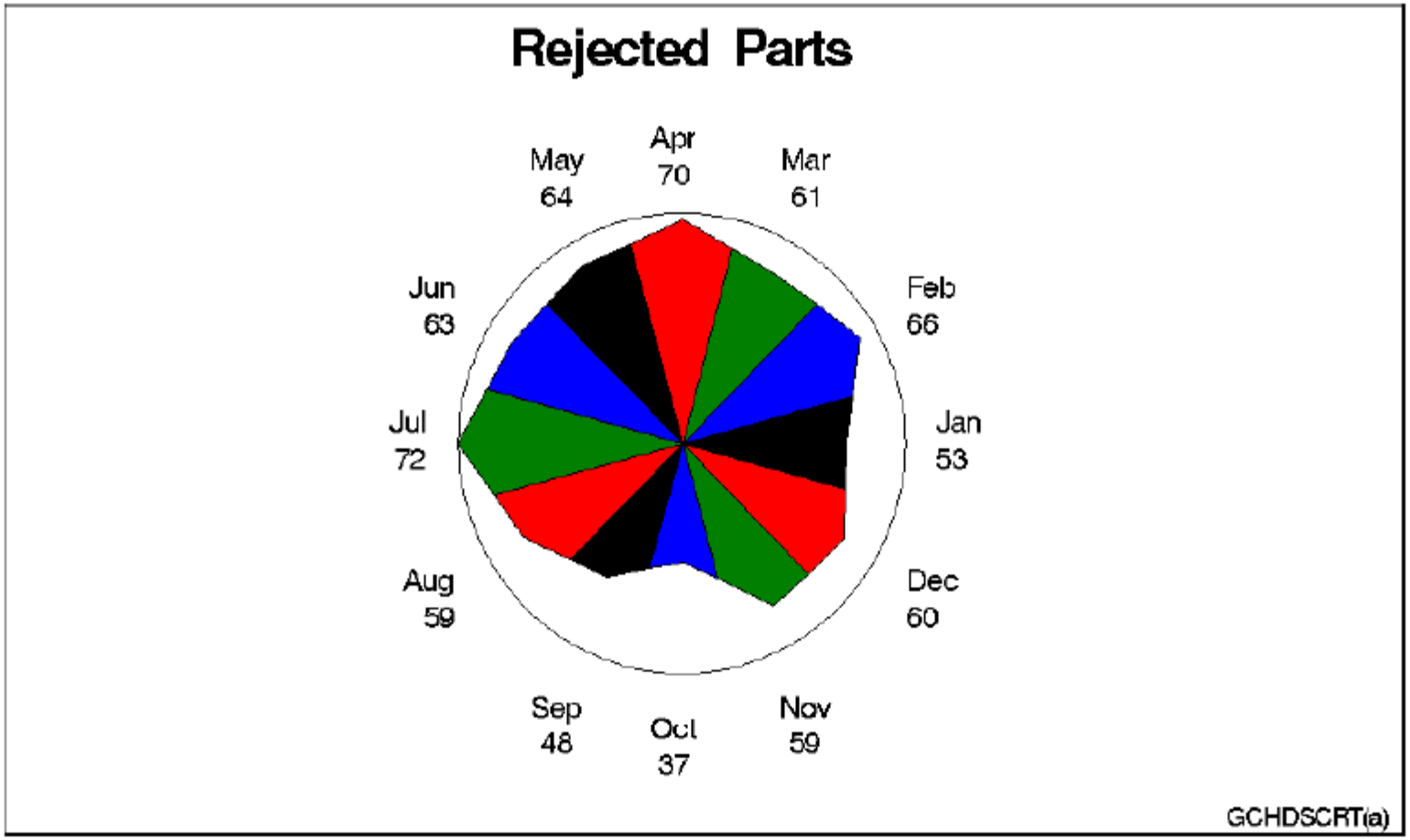
Vertical Bar Chart



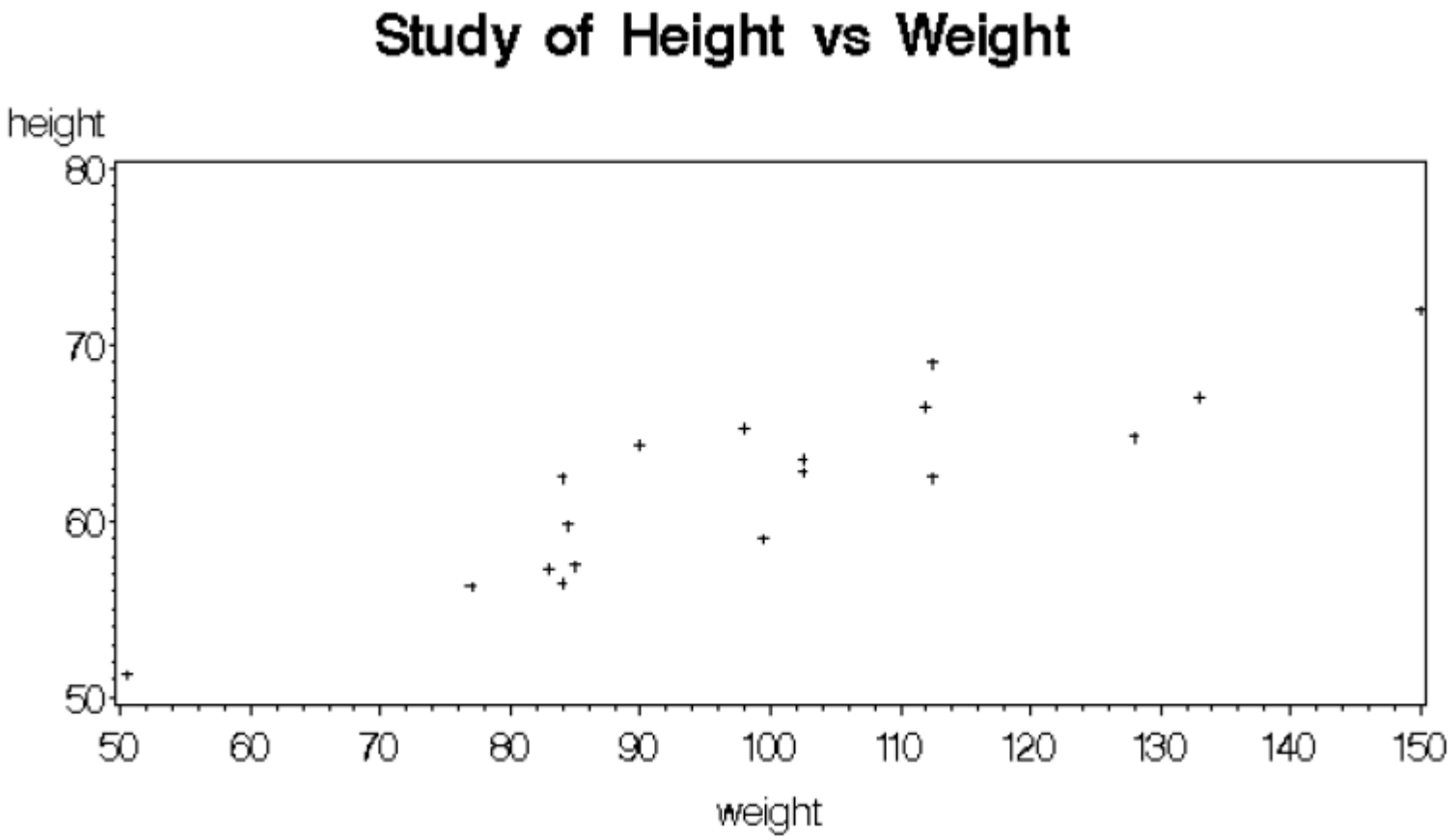
Pie Chart


Sales by Site and Department
Parts
Repairs
Tools
Atlanta
Paris
Sydney
All
Quarters

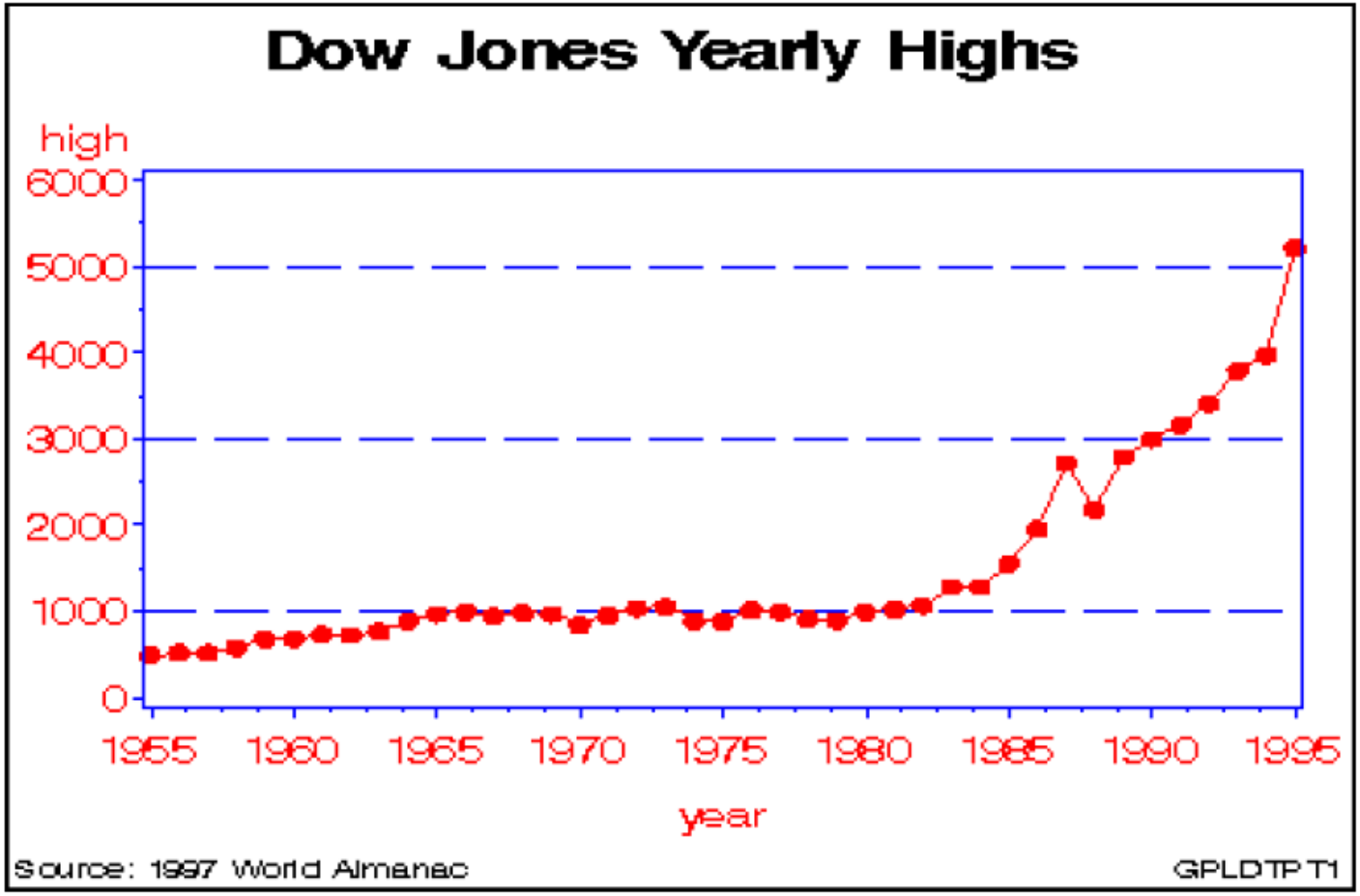
Star Chart


Rejected Parts
Apr
70
Mar
61
Feb
66
Jan
53
Dec
60
Nov
59
Oct
37
Sep
48
Aug
59
Jul
72
Jun
63
May
64
GCHDSCRT(a)

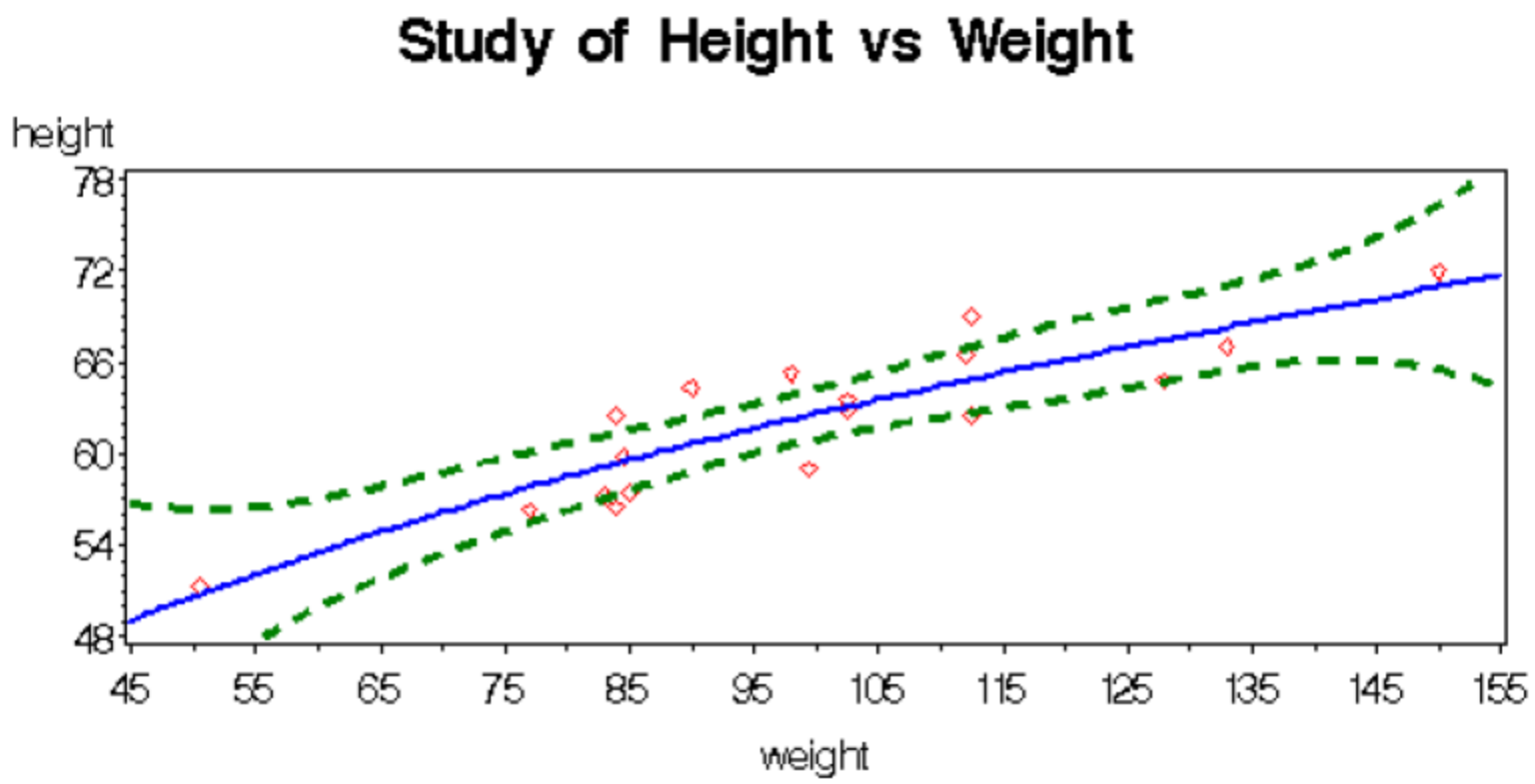
Scatter Plot



Line Plot

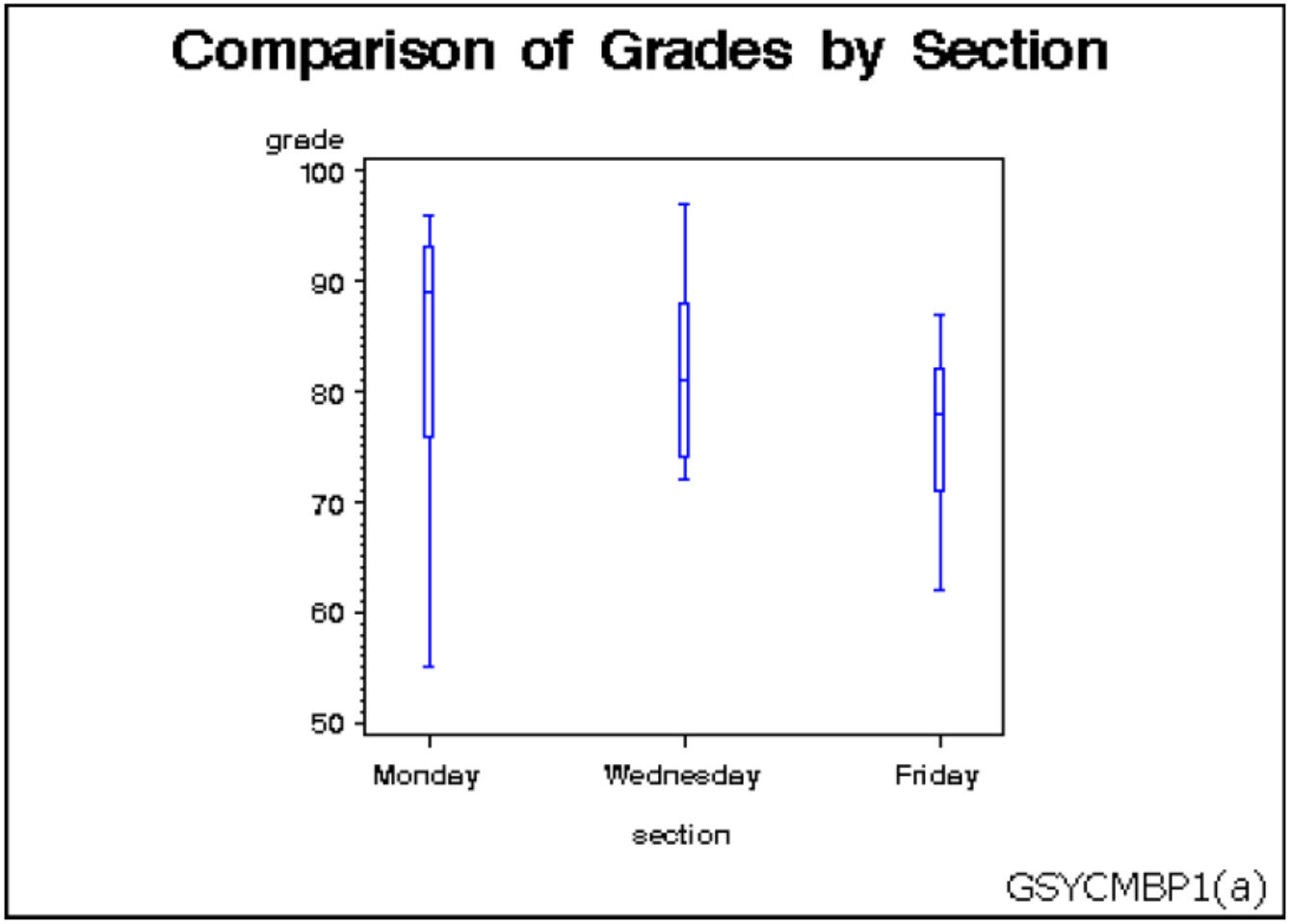

Dow Jones Yearly Highs
high
6000
5000
4000
3000
2000
1000
0
1955
1960
1965
1970
1975
1980
1985
1990
1995
year
Source: 1997 World Almanac
GPLDTPT1

Regression Plot

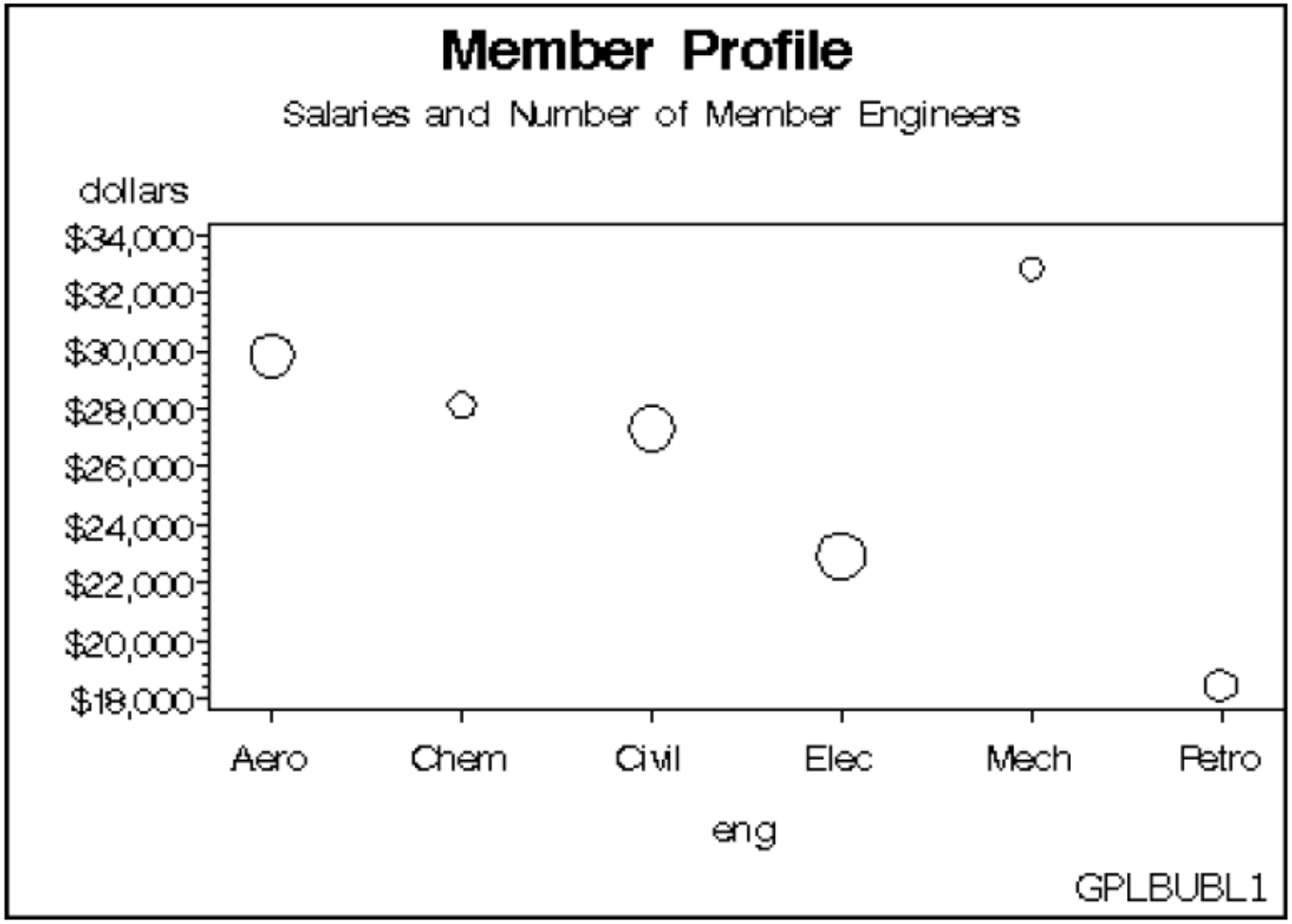


Regression Equation:
height = 30.77829 + 0.50121*weight − 0.002375*weight^2 + 5.448E−6*weight^3

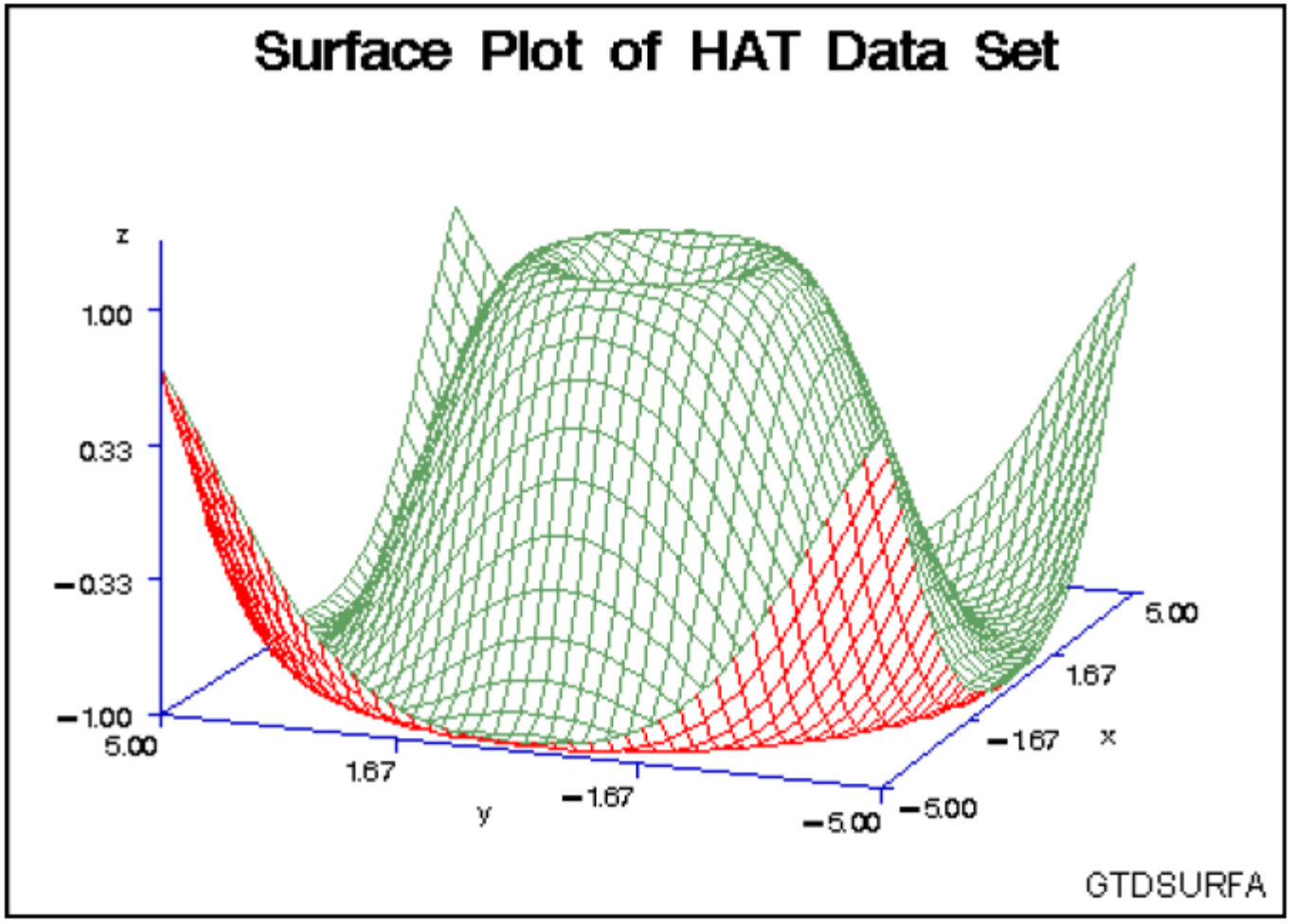
High-Low Plot



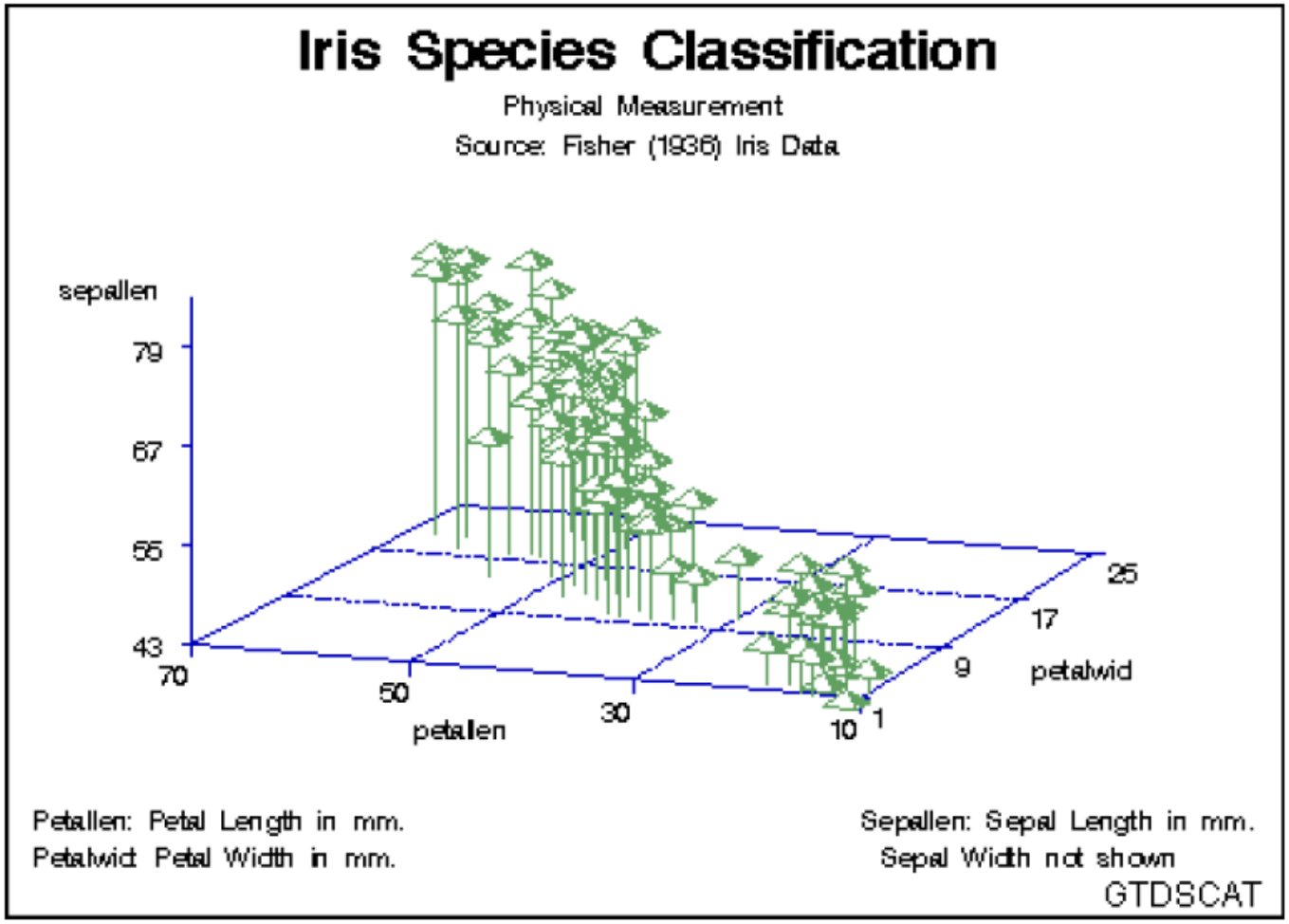
Bubble Plot


Member Profile
Salaries and Number of Member Engineers
dollars
$34,000
$32,000
$30,000
$28,000
$26,000
$24,000
$22,000
$20,000
$18,000
Aero
Chem
Civil
Elec
Mech
Petro
eng
GPLBUBL1

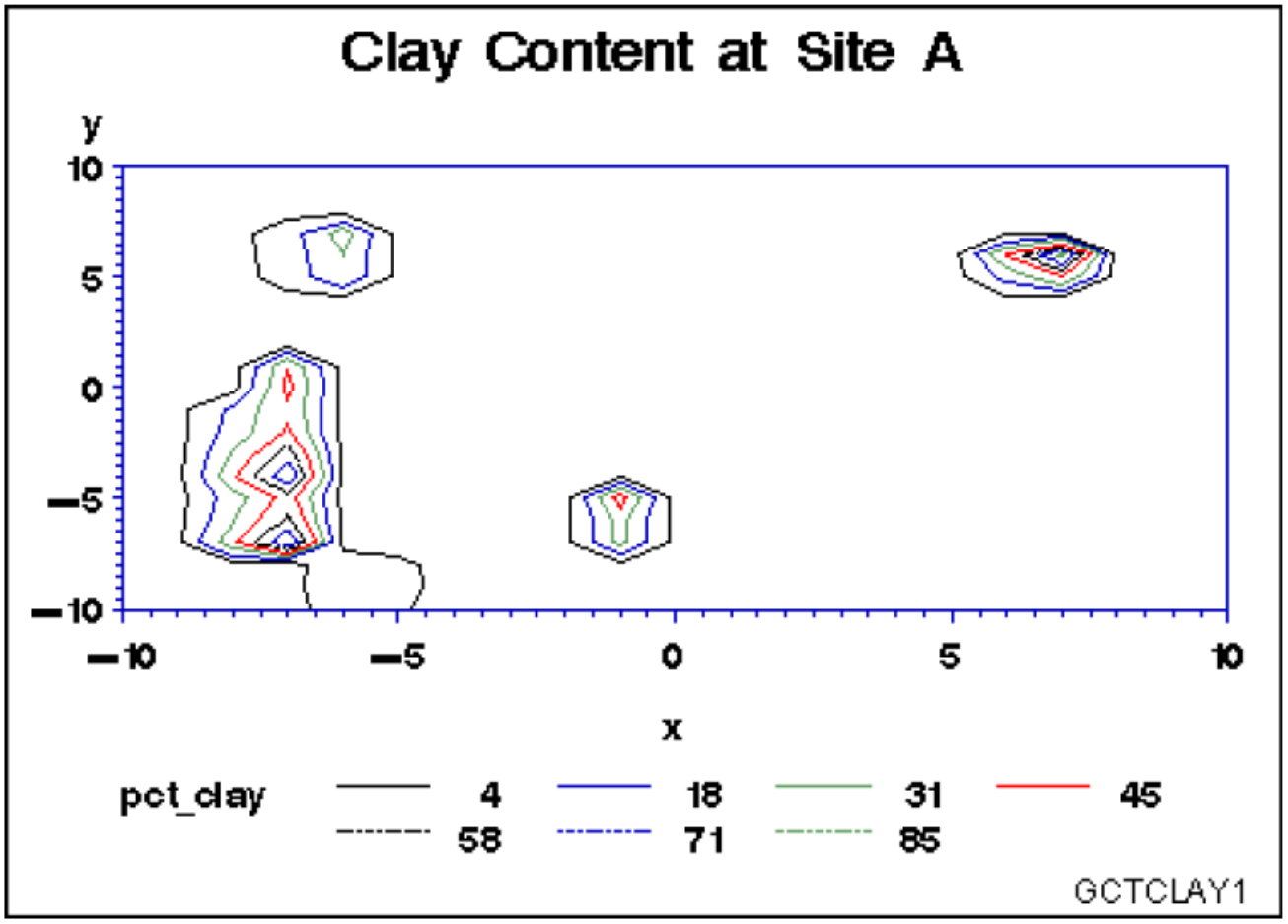
Surface Plot



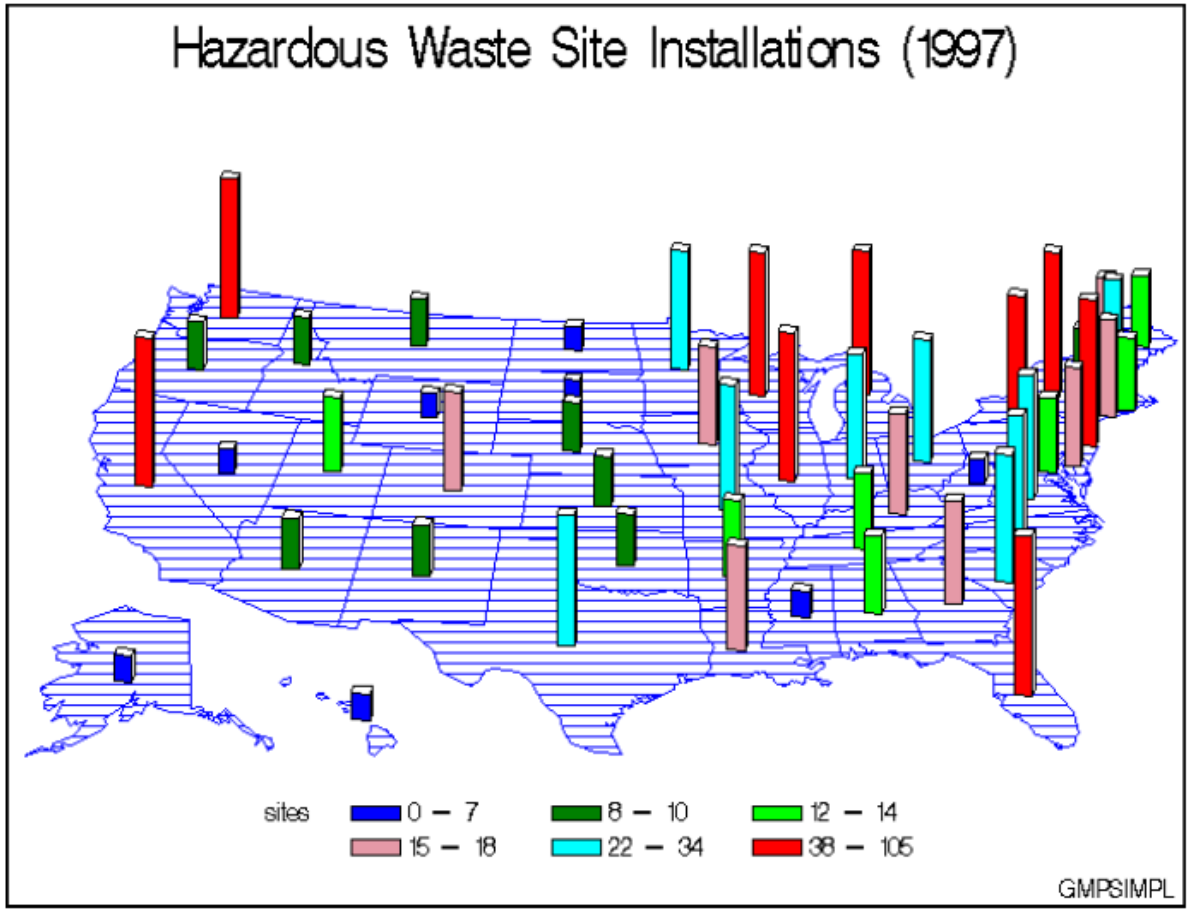
3-D Scatter Plot



Contour Plot

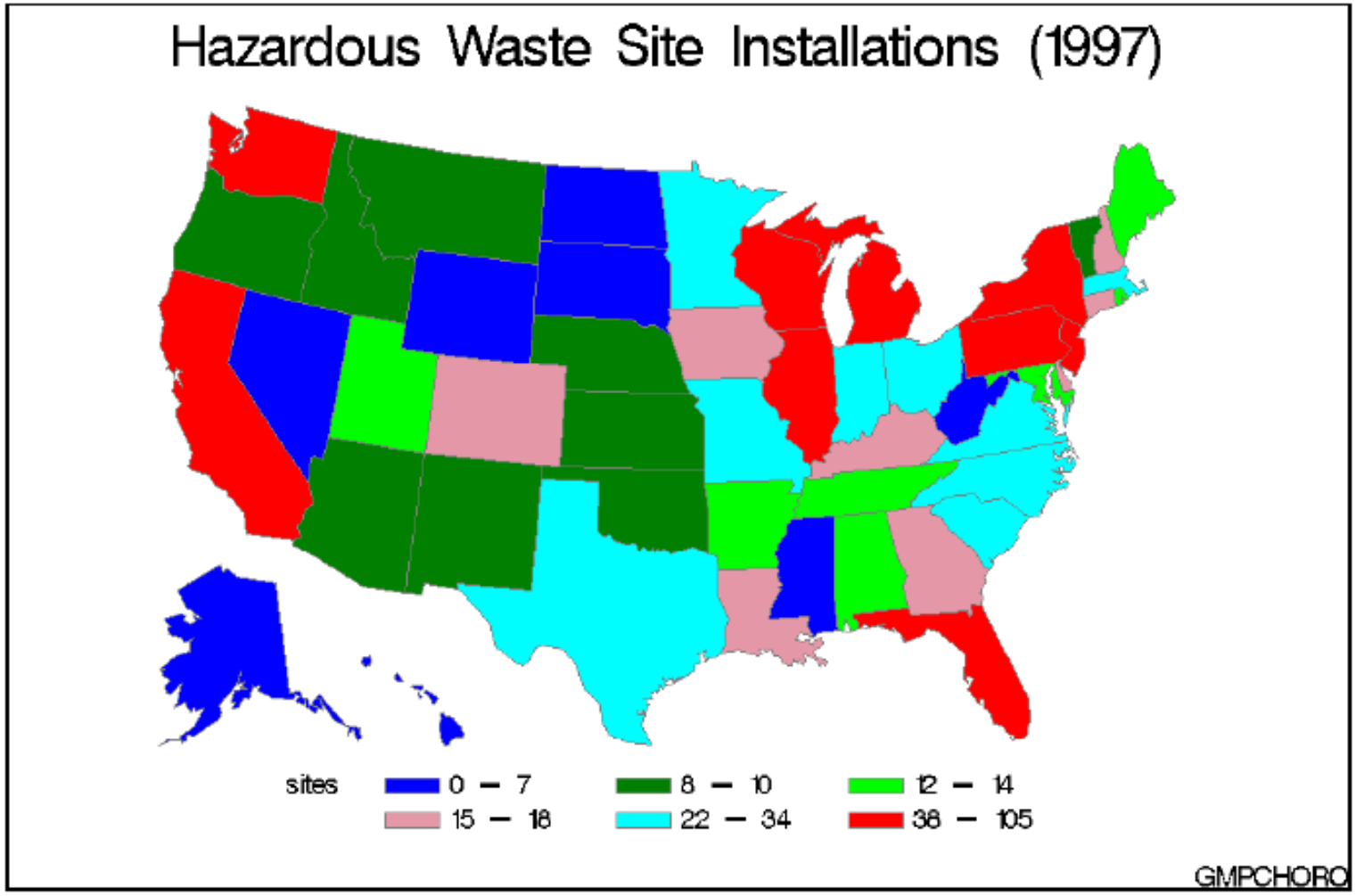


Block Map

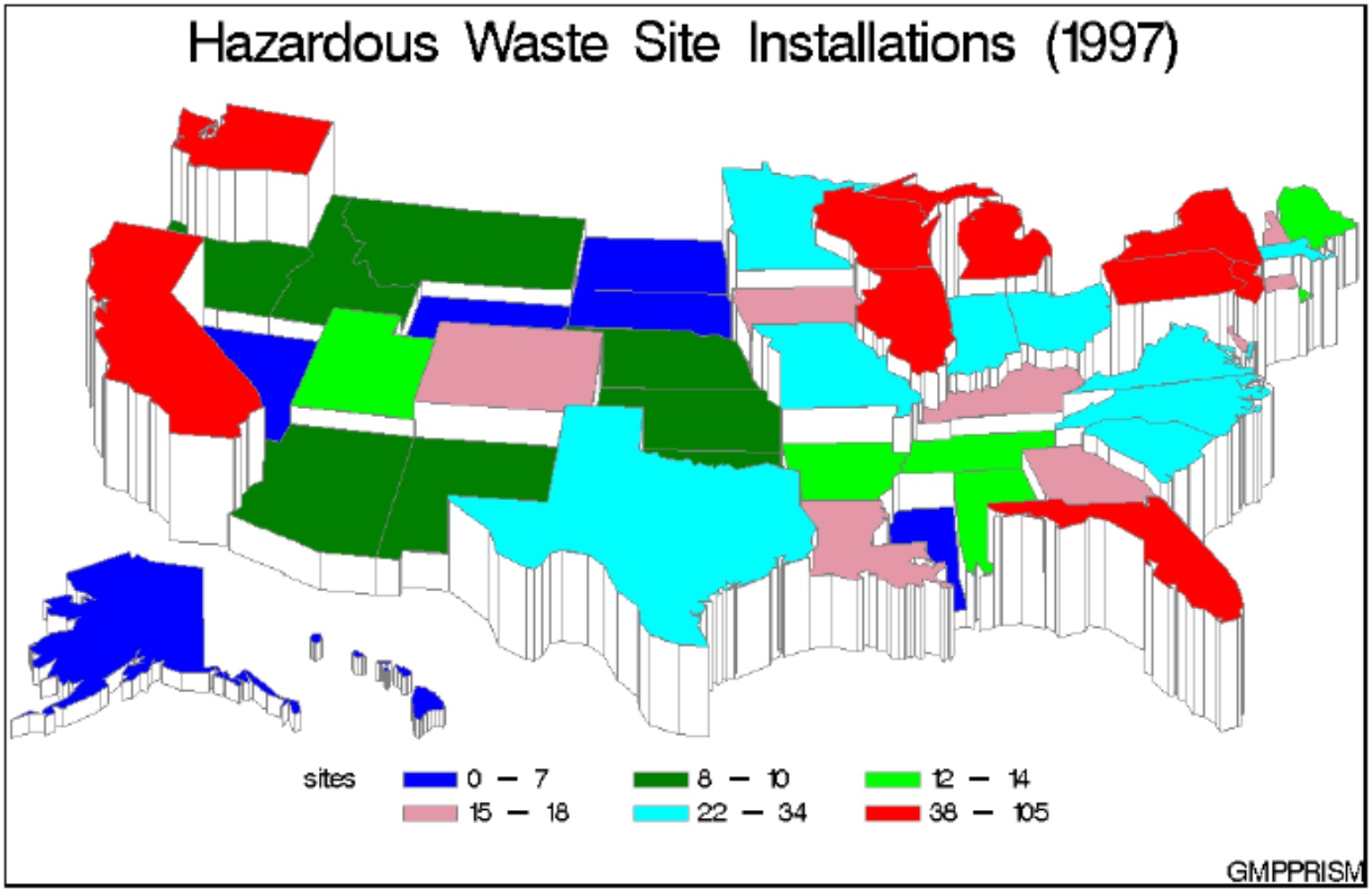

Hazardous Waste Site Installations (1997)
sites
0 – 7
8 – 10
12 – 14
15 – 18
22 – 34
36 – 105
GMPSIMPL

China Map

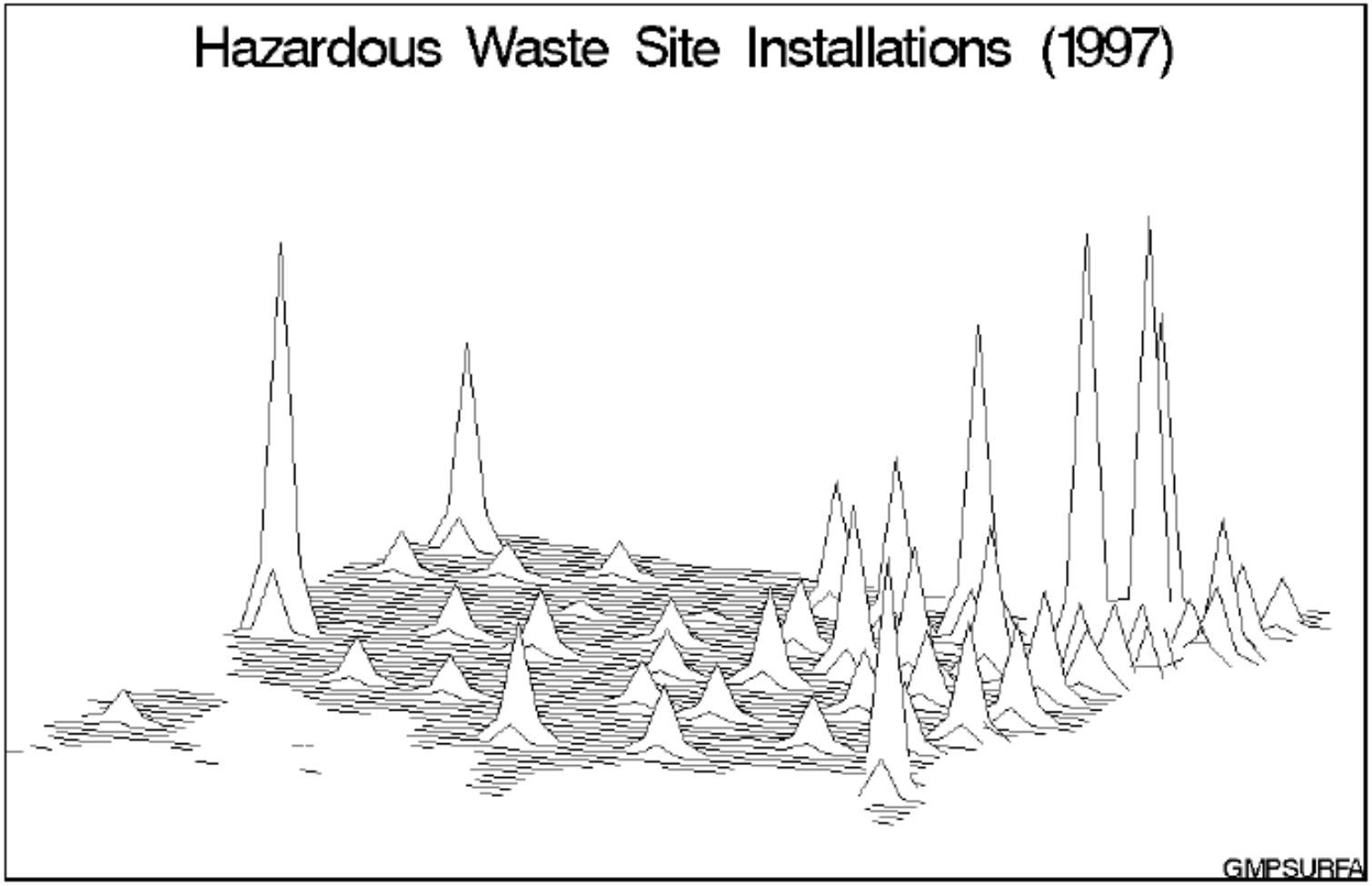
Choropleth Map



Prism Map


Hazardous Waste Site Installations (1997)
sites
0 – 7
8 – 10
12 – 14
15 – 18
22 – 34
38 – 105
GMPPRISM

Surface Map



Text Slide


New Directions
Goals and strategies
for the coming year
ABC Engineering, Inc
January 1998
GSI TEXTS

Graphs Combination

Monthly Sales Figures

January	$ 10,500
February	$ 26,780
March	$ 267,600
Total	$ 299,880

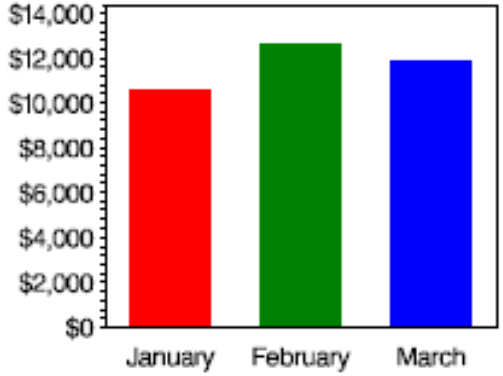
US Marketing Division

Inflation Rate

January	1.6 %
February	2.1 %
March	0.3 %

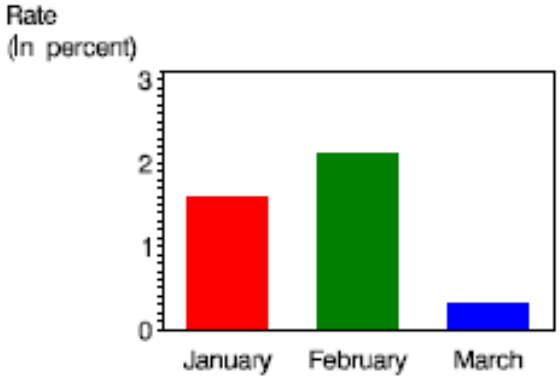
US Marketing Division

Monthly Sales Figures



US Marketing Division

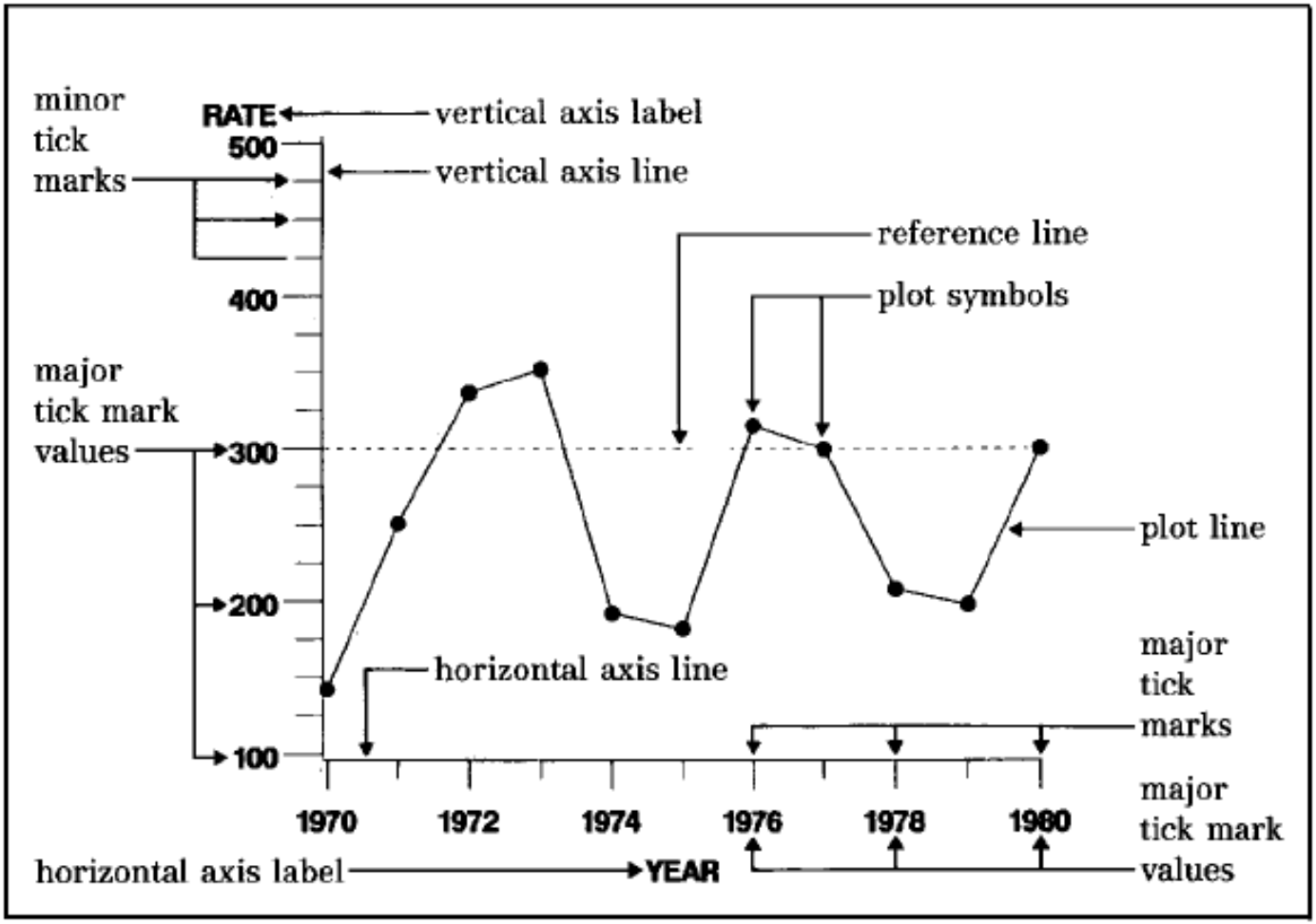
Inflation Rate



US Marketing Division

Annotated Graph

Graph Elements



Graph Procedure

- PROC statement
  - proc gplot data=demo;
- Subordinate statement
  - plot ht*wt;
- BY statement
- NOTE statement
  - Title;
  - Footnote;
- RUN statement
  - quit;

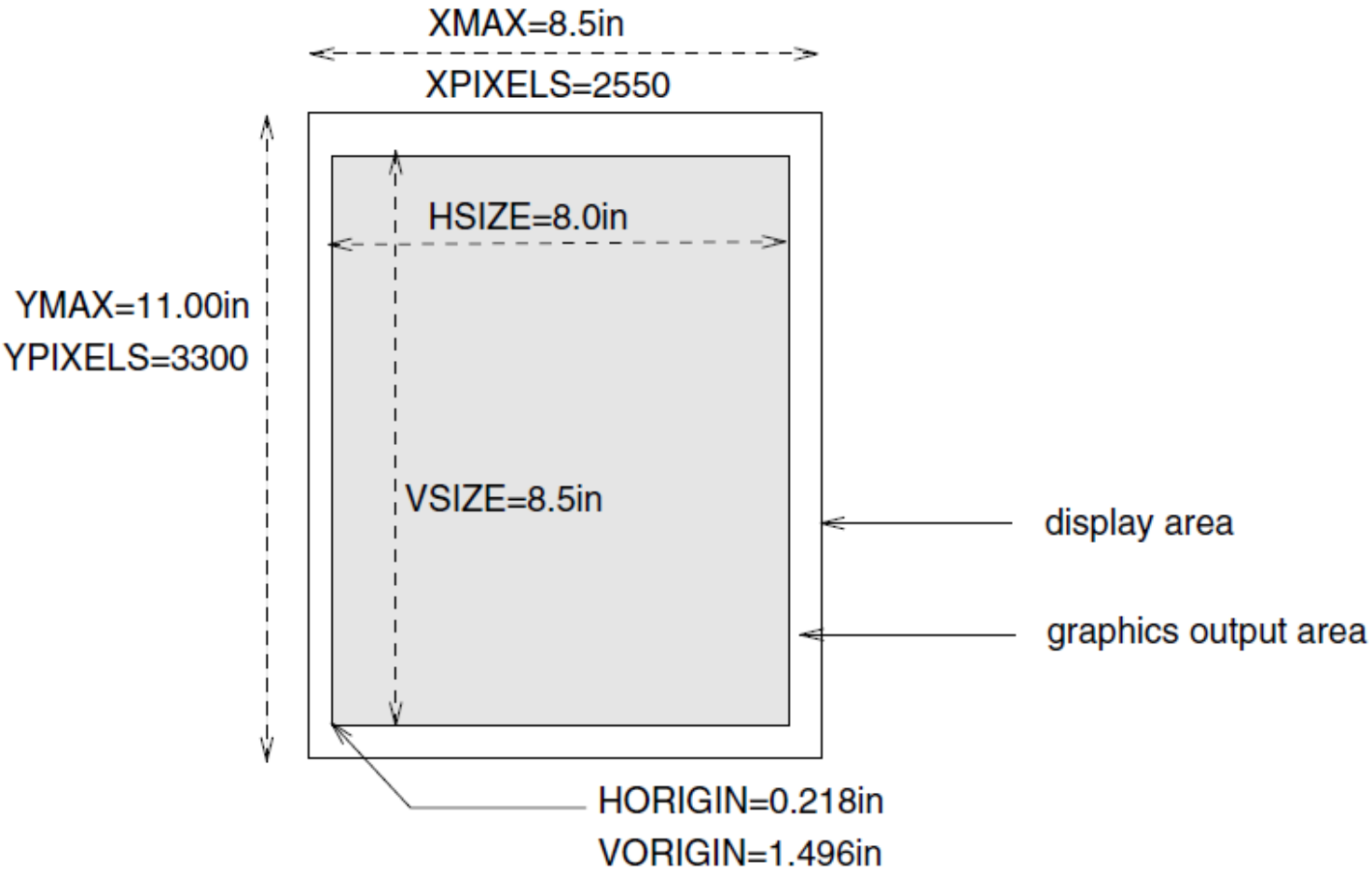
Global Statements

- GOPTIONS
- SYMBOL
- AXIS
- LEGEND
- PATTERN
- TITLE & FOOTNOTE

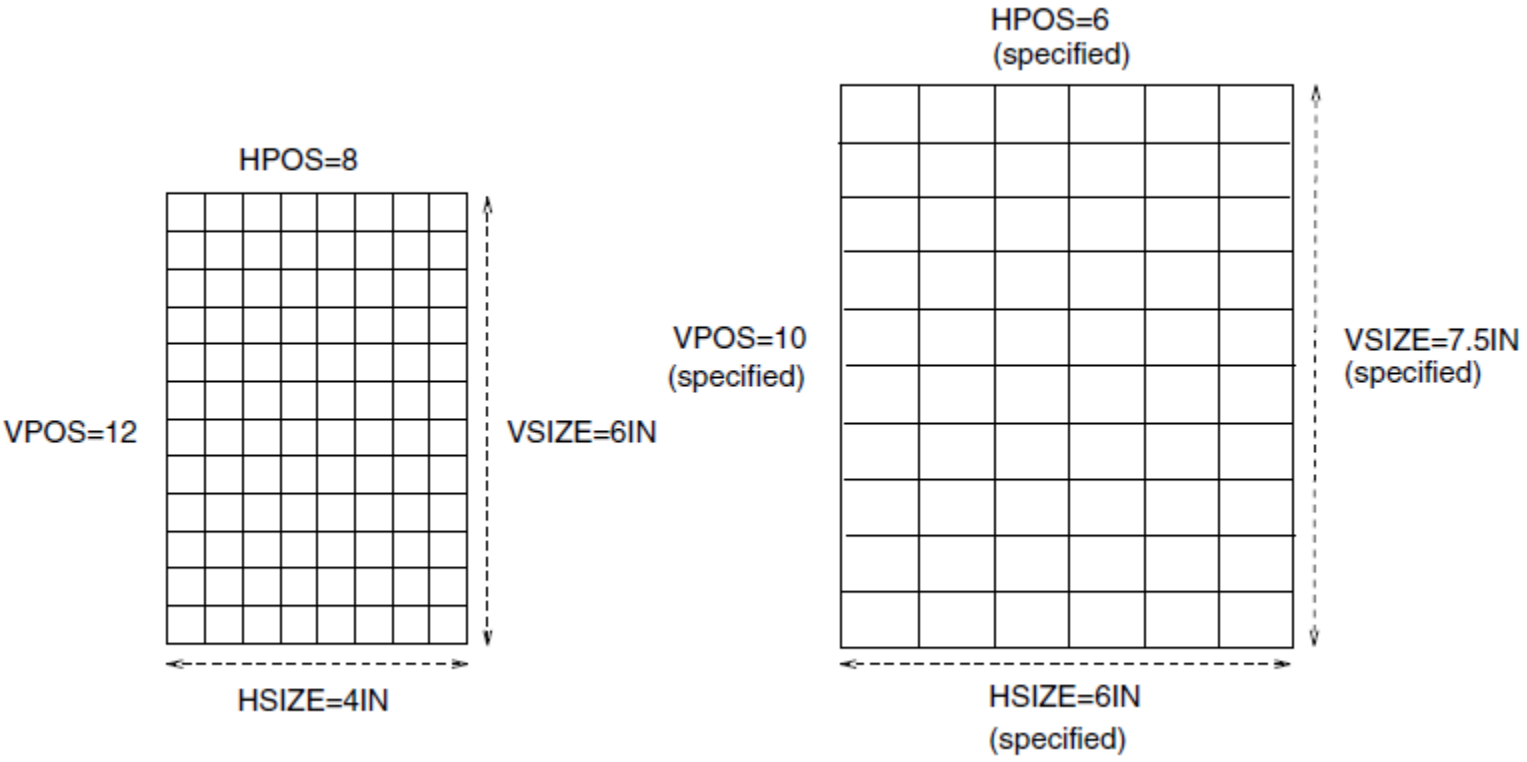
Other Statements

- LIBNAME
- FILENAME
- FORMAT
- LABEL
- ODS
- OPTIONS

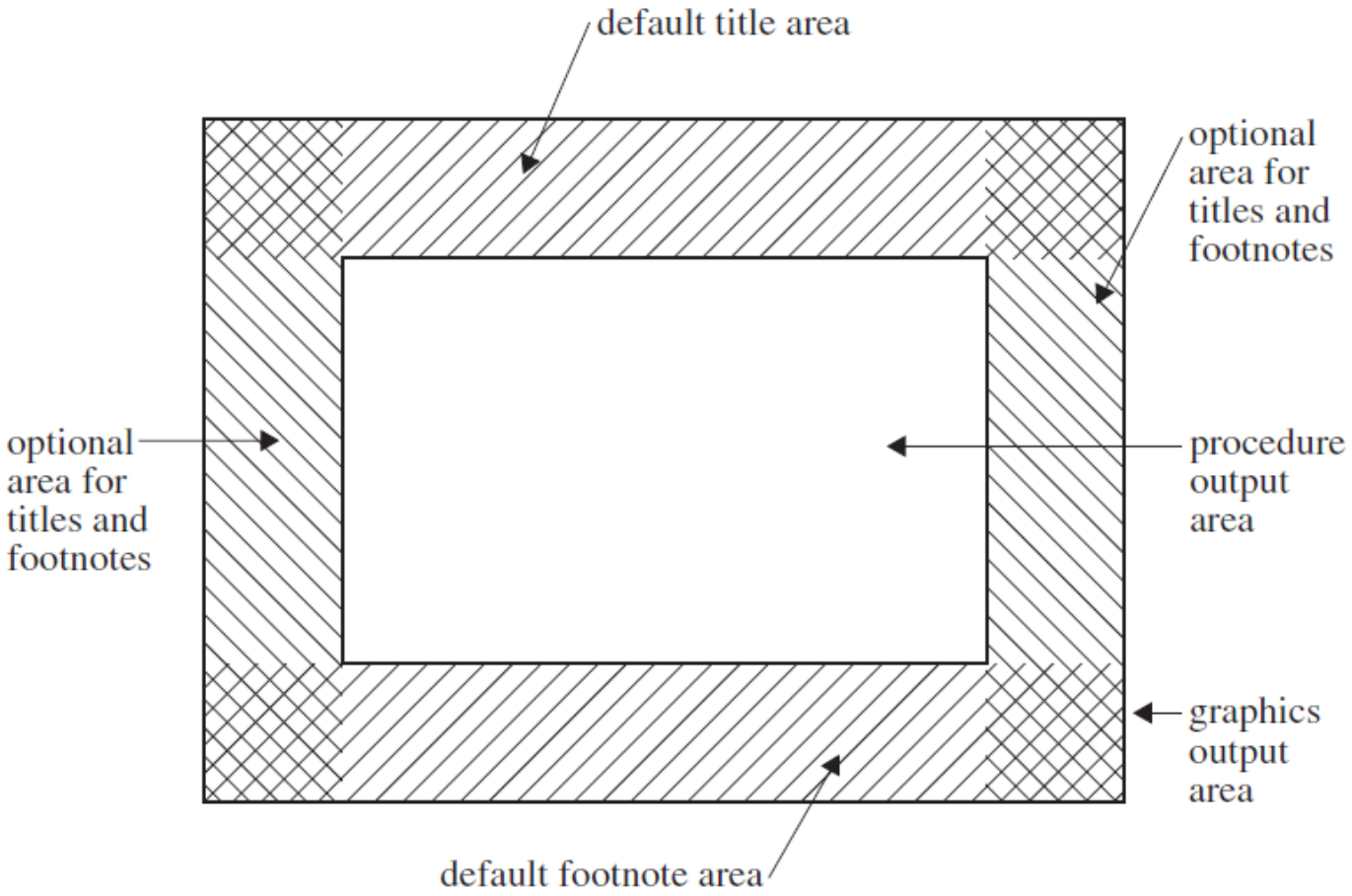
Output Area


XMAX=8.5in
XPIXELS=2550
HSIZE=8.0in
YMAX=11.00in
YPIXELS=3300
VSIZE=8.5in
display area
graphics output area
HORIGIN=0.218in
VORIGIN=1.496in

Size of Cell


HPOS=8
VPOS=12
VSIZE=6IN
HSIZE=4IN
HPOS=6
(specified)
VPOS=10
(specified)
VSIZE=7.5IN
(specified)
HSIZE=6IN
(specified)

Default Location for Graph Elements



Device Drivers

```
proc gdevice catalog=sashelp.devices nofs;
        list;
run;
quit;

goptions device=cgm;
```

Save Graph Files

- GSFNAME
- GSFMODE = Replace, Append
- GSACCESS

```
filename grafout "&cd.\slide.ps";

goptions reset=all
                                device=pscolor
                                gsfname=grafout
                                gsfmode=replace
                                ftext=swissb;

proc gslide border name='proposal';
        title1 h=4 'Proposed Design Improvements:';
        title2 h=3 ' * Increase Stability';
        title3 h=3 ' * Increase Speed';
        title4 h=3 ' * Reduce Weight';
        footnote h=2 j=l 'ABC Company';
run;
quit;
```

Colors in SAS/GRAPH

- SAS employs a number of different color schemes
  - HLS (hue lightness saturation)
  - RGB (red green blue)
  - HSV (hue saturation value)
  - Gray-Scale
  - CMY(K) (cyan magenta yellow(black))
  - SAS color names (from the SAS Registry)
  - the SAS Color Naming System (CNS).

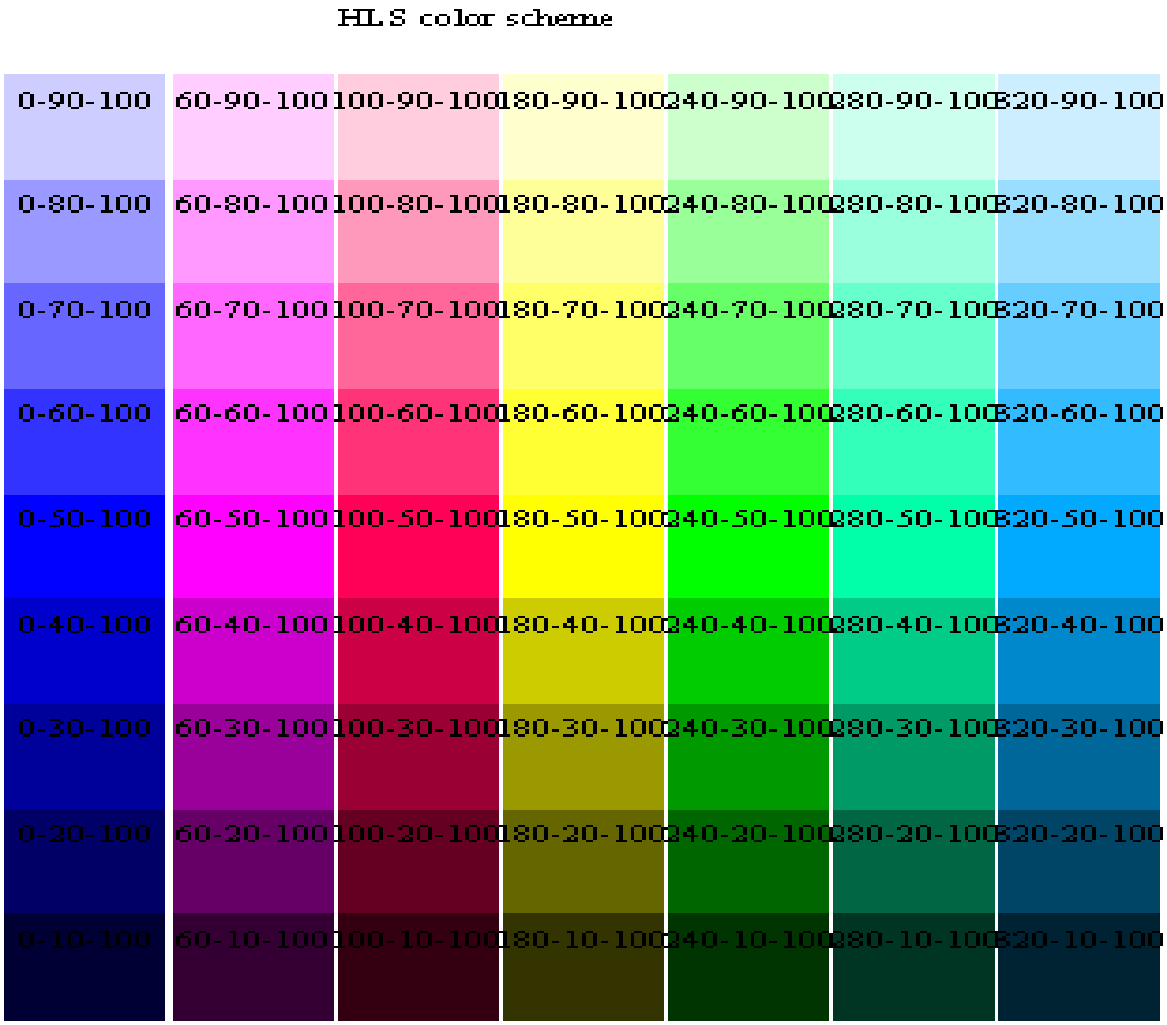
- Each color scheme uses different complex algorithms to construct a given color

HLS color scheme

- Any color in the HLS scheme can be constructed by specifying values for hue, lightness and saturation
- HLS color names are of the form H*hhhllss* (hexadecimal format)
  - *hhh* is the hue component (000-168)
  - *ll* is the lightness component (0-FF)
  - *ss* is the saturation component (0-FF).

- The %HLS converts values of hue, lightness and saturation into the corresponding HLS code and circumvents the need for hexadecimal representations of colors
  - E.g. %hls(360,50,100)=H16880FF, %hls(130,50,90)=H08280E6
  - Pattern color= %hls(130,50,90)

HLS color scheme



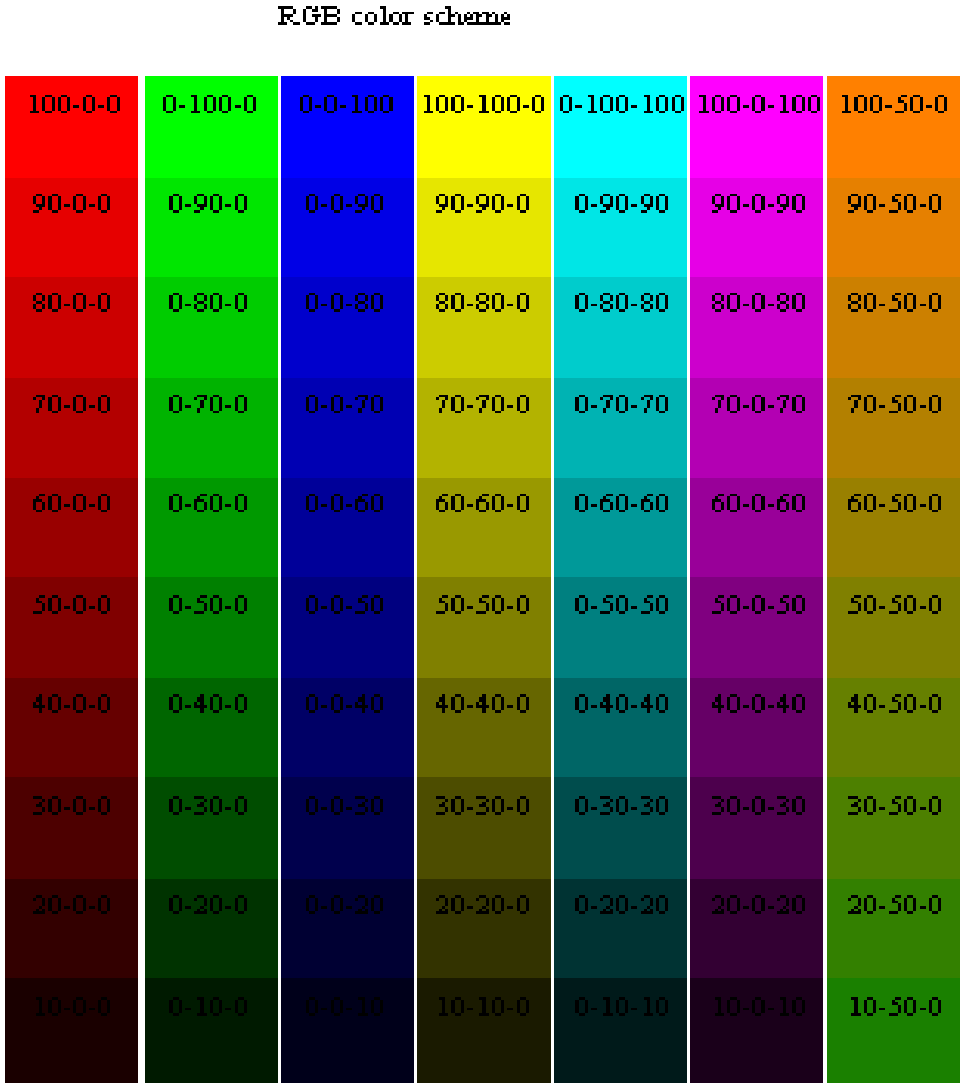
RGB color scheme

- Any given color can be defined as a combination of red, green and blue colors

- RGB color names are of the form CX*rrggbb* (hexadecimal format)
  - *rr* is the red component (00-FF)
  - *gg* is the green component (00-FF)
  - *bb* is the blue component (00-FF).

- Like HLS, %RGB converts percentages to a hexadecimal number
  - E.g. %RGB(30,50,100)=CX4D80FF, %RGB(80,50,90)=CXCC80E6
  - Pattern color= %RGB(30,50,100)

RGB color scheme

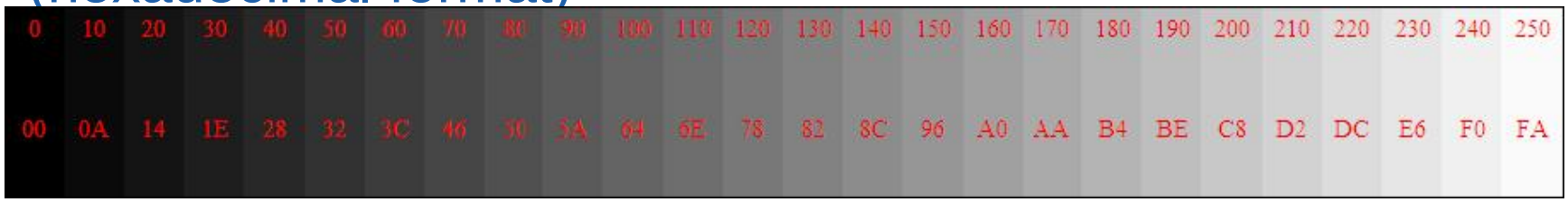


GRAY-Scale color scheme

- The Gray-scale color scheme may be used to create various shades of gray

- Gray-scale shades are formed by combining the word GRAY with the hexadecimal value of a color from 0 to 255

- Gray-scale color names are of the form GRAY*//* (hexadecimal format)

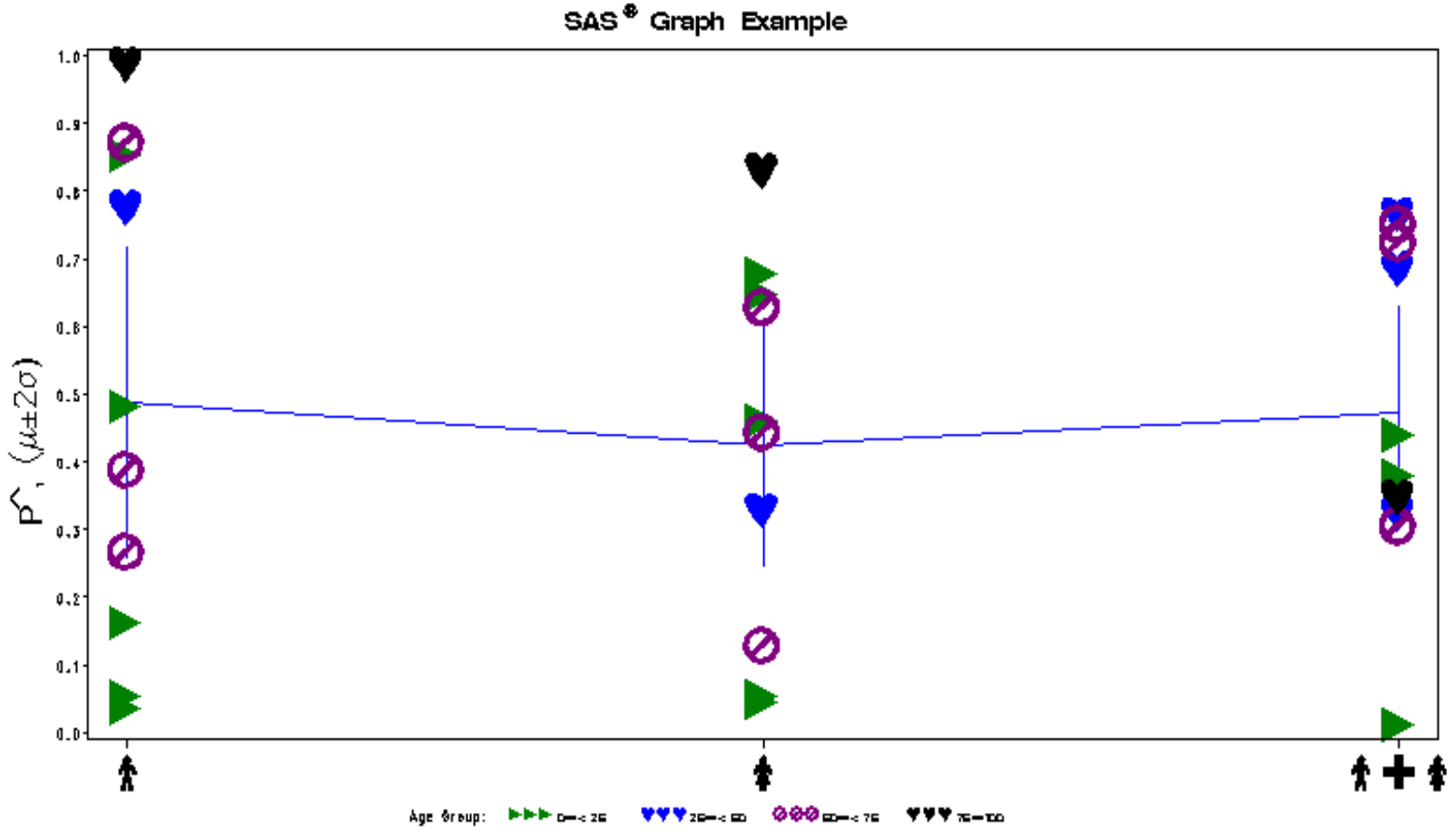


Fonts/Symbols in SAS/GRAPH

- All of the SAS/GRAPH fonts are stored in the catalog SASHELP.FONTS.
- The special symbols can be Greek letters, mathematical symbols, subscription,superscription, underline and custom-designed symbols
- To preview the fonts/symbols and its corresponding character codes, we can use thefollowing code:

```
proc gfont name=greek/marker/music/math/sperical
        nobuild
        height=3.7
        romcol=red
        romfont=swissl
                noromhex
        romht=2.7
        showroman;
run;
quit;
```

Fonts/Symbols in SAS/GRAPH



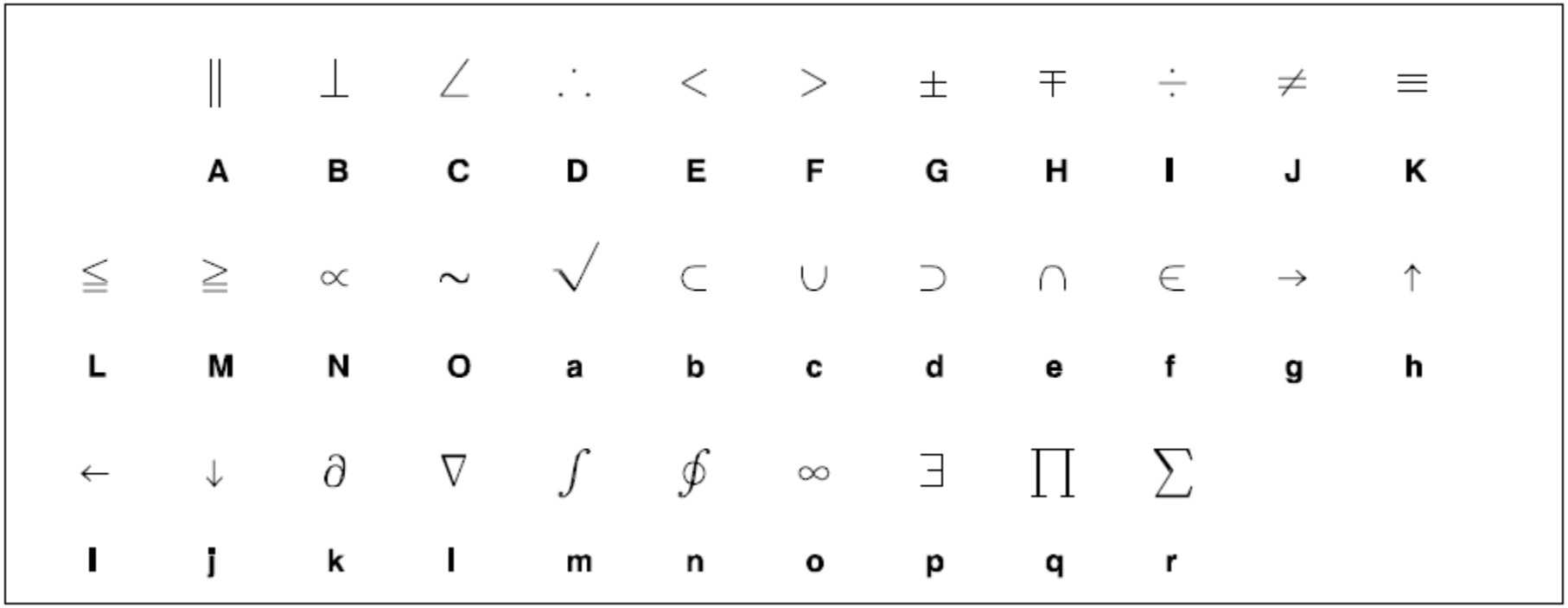
† This is only an example.

Front (1)

Title front=xxxx;

Type Style	Font Name	Type Sample	Uniform Font
Brush	BRUSH	*A B C a b c 1 2 3*	
Century			
Bold	CENTB	**A B C a b c 1 2 3**	CENTBU
Bold Empty	CENTBE	A B C a b c 1 2 3	
Bold Italic	CENTBI	***A B C a b c 1 2 3***	CENTBIU
Bold Italic Empty	CENTBIE	*A B C a b c 1 2 3*	
Expanded	CENTX	**A B C a b c 1 2 3**	CENTXU
Expanded Empty	CENTXE	A B C a b c 1 2 3	
Expanded Italic	CENTXI	***A B C a b c 1 2 3***	CENTXIU
Expanded Italic Empty	CENTXIE	*A B C a b c 1 2 3*	

Front (2)



GOPTIONS

RESET=ALL | GLOBAL | statements

GOPTIONS

options that affect the appearance of the display area and the graphics output

ASPECT=*scaling-factor*
AUTOSIZE=ON | OFF | DEFAULT
BORDER | NOBORDER
CELL | NOCELL
GSIZE=*lines*
HORIGIN=*horizontal-offset* <IN | CM>
HPOS=*columns*
HSIZE=*horizontal-size* <IN | CM>
IBACK= *fileref* | *'external-file'*
IMAGESTYLE = TILE | FIT
IMAGEPRINT | NOIMAGEPRINT
ROTATE=LANDSCAPE | PORTRAIT
ROTATE | NOROTATE
SWAP | NOSWAP
TARGETDEVICE=*target-device-entry*
VORIGIN=*vertical-offset* <IN | CM>
VPOS=*rows*
VSIZE=*vertical-size* <IN | CM>
XMAX=*width* <IN | CM>
XPIXELS=*width-in-pixels*
YMAX=*height* <IN | CM>
YPIXELS=*height-in-pixels*

GOPTIONS

options that affect color

CBACK=*background-color*

CBY=*BY-line-color*

COLORS=<(*colors-list* | NONE)>

CPATTERN=*pattern-color*

CSYMBOL=*symbol-color*

CTEXT=*text-color*

CTITLE=*title-color*

PENMOUNTS=*active-pen-mounts*

PENSORT | NOPENSORT

GOPTIONS

options that control font selection or text appearance

CHARTYPE=*hardware-font-chartype*
FASTTEXT | NOFASTTEXT
FBY=*BY-line-font*
FCACHE=*number-fonts-open*
FONTRES=NORMAL | PRESENTATION
FTEXT=*text-font*
FTITLE=*title-font*
FTRACK=LOOSE | NONE | NORMAL | TIGHT | TOUCH | V5
HBY=*BY-line-height <units>*
HTEXT=*text-height <units>*
HTITLE=*title-height <units>*
RENDER=APPEND | DISK | MEMORY | NONE | READ
RENDERLIB=*libref*
SIMFONT=*software-font*

GOPTIONS

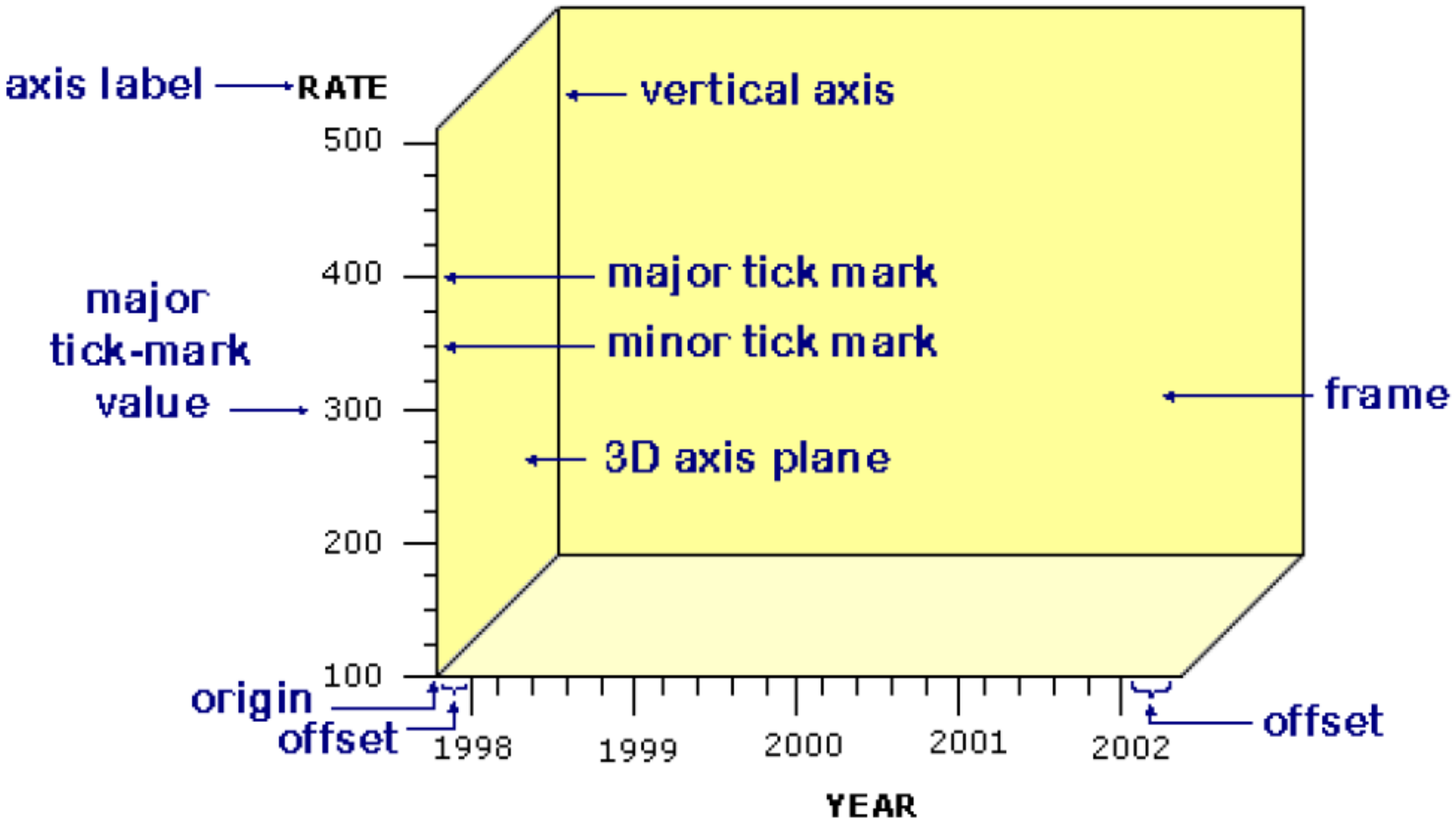
options that control how output is sent to devices or files

ADMGDF | NOADMGDF
DEVADDR=*device-address*
DEVICE=*device-entry*
DEVMAP=*device-map-name* | NONE
EXTENSION='*file-type*'
FILECLOSE=DRIVERTERM | GRAPHEND
FILEONLY | NOFILEONLY
GACCESS=*output-format* | '*output-format* > *destination*'
GEND='*string*' <...'*string-n*'>
GEPILOG='*string*' <...'*string-n*'>
GOUTMODE=APPEND | REPLACE
GPROLOG='*string*' <...'*string-n*'>
GPROTOCOL=*module-name*
GSFLEN=*record-length*
GSFMODE=APPEND | PORT | REPLACE
GSFNAME=*fileref*
GSFPROMPT | NOGSFPROMPT
GSTART='*string*' <...'*string-n*'>
HANDSHAKE=HARDWARE | NONE | SOFTWARE | XONXOFF
KEYMAP=*map-name* | NONE
POSTGEPILOG='*string*'

GOPTIONS

```
goptions reset=all
        cback=white
        colors=(black)
                gunit=pct
        rotate=landscape
                device=cgmof97l
        gsfmode=replace
        gsfname=plotgph;
```

Axis


axis label
RATE
500
400
300
200
100
major
tick-mark
value
vertical axis
major tick mark
minor tick mark
3D axis plane
frame
origin
offset
1998
1999
2000
2001
2002
offset
YEAR

Axis Syntax

axis scale options:

INTERVAL=EVEN | UNEVEN | PARTIAL

LOGBASE=*base* | E | PI

LOGSTYLE=EXPAND | POWER

ORDER=(*value-list*)

appearance options:

COLOR=*axis-color*

LENGTH=*axis-length* <*units* >

NOBRACKETS

Axis Syntax

NOPLANE
OFFSET=(*<n1 ><,n2 >)<units >* | (*<n1<units>><,n2<units >>*)
ORIGIN=<(*x*><,*y* >)*<units>* | (*<x<units >><,y<units>>*)
STYLE=*line-type*
WIDTH=*thickness-factor*

tick mark options:

MAJOR=(*tick-mark-suboption(s)*)| NONE
MINOR=(*tick-mark-suboption(s)*)| NONE

text options:

LABEL=(*text-argument(s)*)| NONE
REFLABEL=(*text-argument(s)*)| NONE
SPLIT="*split-char*"
VALUE=(*text-argument(s)*)| NONE

Axis Syntax

Unit

CELLS	character cells
CM	centimeters
IN	inches
PCT	percentage of the graphics output area
PT	points

If you omit *units*, a unit specification is searched for in this order:

1. GUNIT= in a GOPTIONS statement
2. the default unit, CELLS.

Axis Syntax

Color

Option	Items Affected
AXIS statement:	axis label
LABEL=(COLOR=*color*)	reference-line labels
REFLABEL=(COLOR=*color*)	major tick mark values
VALUE=(COLOR=*color*)	
calling procedure:	all axis text (AXIS label and major tick mark value descriptions)
CTEXT=	
CAXIS=	axis line and major and minor tick marks

Axis Syntax

Text Description Suboptions

ANGLE=*degrees*

COLOR=*text-color*

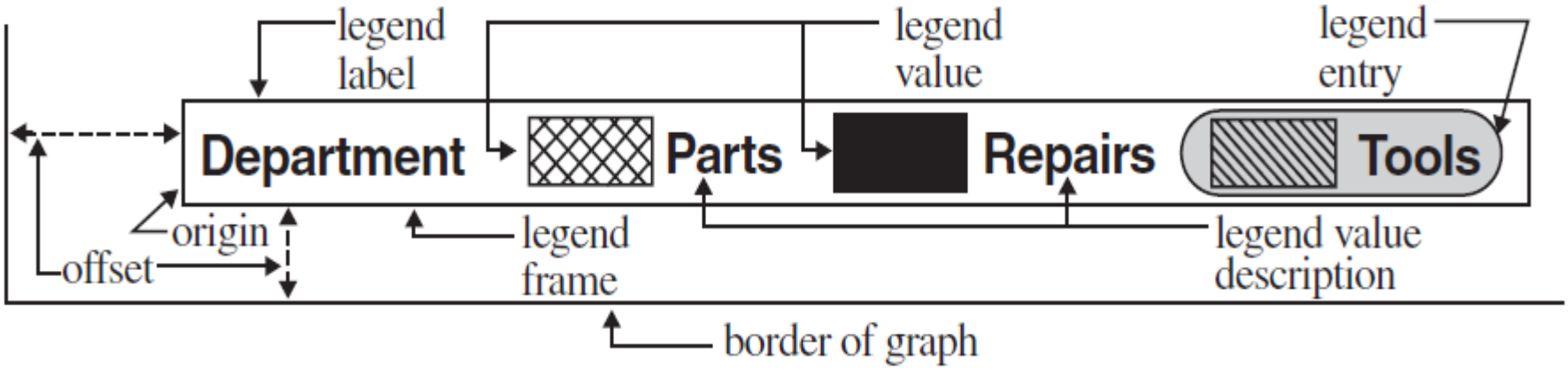
FONT=*font* | NONE

HEIGHT=*text-height* <*units* >

JUSTIFY=LEFT | CENTER | RIGHT

ROTATE=*degrees*

Legend


legend label
legend value
legend entry
Department
Parts
Repairs
Tools
origin
offset
legend frame
legend value description
border of graph

Legend

- LEGEND<1...99><options>;

option(s) can be one or more options from any or all of the following categories:

- appearance options
- ACROSS=number-of-columns
- CBLOCK=block-color
- CBORDER=frame-color
- CFRAME=background-color
- CSHADOW=shadow-color
- DOWN=number-of-rows
- FRAMEFWIDTH=thickness-factor
- SHAPE=BAR(width,height) <units> | LINE(length) <units> | SYMBOL(width,height) <units>

Legend

position-options

- MODE=PROTECT | RESERVE | SHAREOFFSET=(<x ><,y >)<units > | (<x <units >><,y <units >>)
- ORIGIN=(<x ><,y >)<units > | (<x <units >><,y <units >>)
- POSITION=(<BOTTOM | MIDDLE | TOP> <LEFT | CENTER | RIGHT> <INSIDE | OUTSIDE>)

text-options

- LABEL=(text-argument(s)) | NONEORDER=(value-list)
- VALUE=(text-argument(s)) | NONE

Legend

```
legend label=(justify=c 'Distribution'
              justify=c 'Centers')
value=(tick=1 justify=c 'Portland,'
              justify=c 'Maine'
       tick=2 justify=c 'Paris,'
              justify=c 'France'
       tick=3 justify=c 'Sydney,'
              justify=c 'Australia');
```

Symbol

SYMBOL<1...255>
<COLOR=*symbol-color*>
<MODE=EXCLUDE | INCLUDE>
<REPEAT=*number-of-times*>
<STEP=*distance*<*units*>>
<*appearance-option(s)*>
<*interpolation-option*>
<SINGULAR=*n*>;

appearance-options can be one or more of these:

BWIDTH=*box-width*

CI=*line-color*

CO=*color*

CV=*value-color*

FONT=*font*

HEIGHT=*symbol-height*<*units*>

LINE=*line-type*

POINTLABEL<=(*label-description(s)*) | NONE>

VALUE=*special-symbol* | *text-string* | NONE

WIDTH=*thickness-factor*

Symbol

interpolation-option can be one of these:

- general methods
  - INTERPOL=JOIN
  - INTERPOL=*map* / *plot-pattern*
  - INTERPOL=NEEDLE
  - INTERPOL=NONE
  - INTERPOL=STEP<*placement*><J><S>
- high-low interpolation methods
  - INTERPOL=BOX<*option(s)*><00...25>
  - INTERPOL=HILO<C><*option(s)*>
  - INTERPOL=STD<1 | 2 | 3><*variance*><*option(s)*>
- regression interpolation methods
  - INTERPOL=R<*type*><0><CLM | CLI<50...99>>
- spline interpolation methods
  - INTERPOL=L<*degree*><P><S>
  - INTERPOL=SM<*nn*><P><S>
  - INTERPOL=SPLINE<P><S>

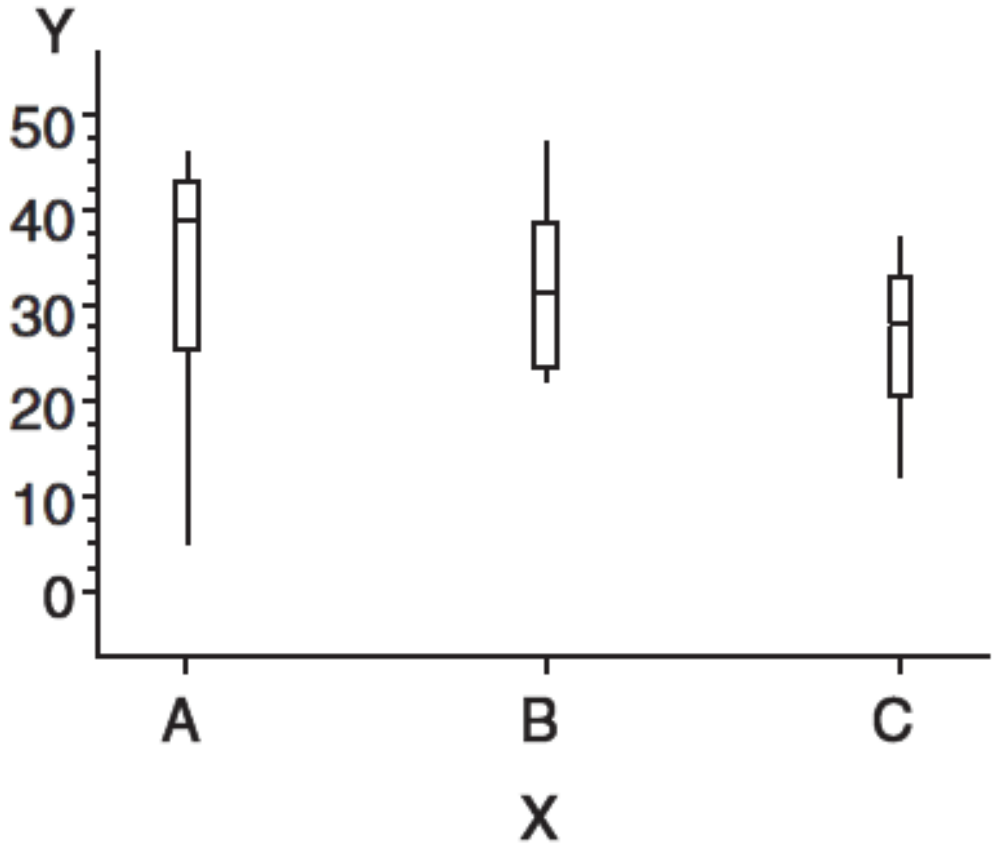
I=BOX <options> <00-25>

F	fills the box with the color specified by CV= and outlines the box with the color specified by CO=
J	joins the median points of the boxes with a line
T	draws tops and bottoms on the whiskers.

In addition, you can specify a percentile to control the length of the whiskers within the range 00 through 25. These are examples of percentile specifications and their effect:

00	high/low extremes. INTERPOL=BOX00 is *not* the same as the default, INTERPOL=BOX.
01	1st percentile low, 99th high
05	5th percentile low, 95th high
10	10th percentile low, 90th high
25	25th percentile low, 75th high; since the box extends from the 25th to the 75th percentile, no whiskers are produced.

I=BOX



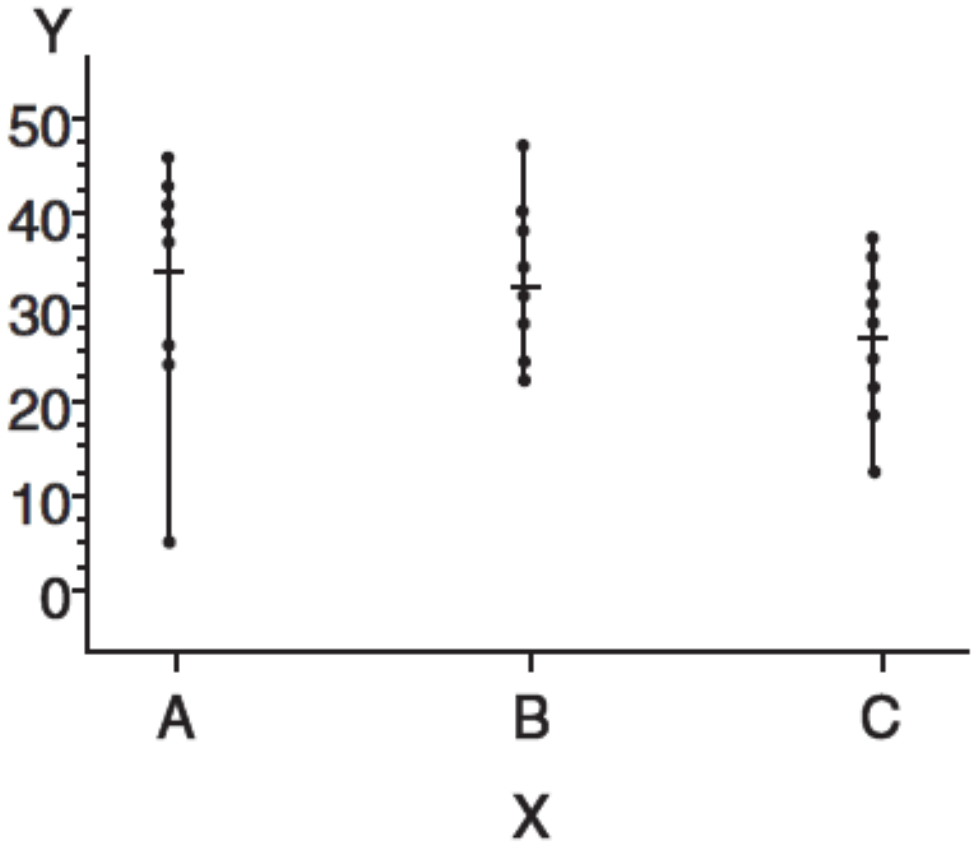
I=HILO <C><option>

C	draws tick marks at the close value instead of at the mean value. Specifying C assumes that there are three values of Y (HIGH, LOW, and CLOSE) for every value of X. If more or fewer than three Y values are specified, the mean is ticked. The Y values can be in any order in the input data set.

In addition, you can specify one of these values for *option*:

B	connects the minimum and maximum Y values with bars instead of lines. Use the BWIDTH= option to increase the width of the bars.
J	joins the mean values or the close values (if HILOC is specified) with a line. This point is not marked with a tick mark. You cannot use the PLOT statement option AREAS= with INTERPOL=HILOJ.
T	adds tops and bottoms to each line.
BJ	connects maximum and minimum values with a bar and joins the mean or close values.
TJ	adds tops and bottoms to the lines and joins the mean or close values.

I=HILO



I=JOIN

- Connects data points with straight lines. Points are connected in the order they occur in the input dataset. Therefore, the data should be sorted by the independent variable.

- If the data contain missing values, the observations are omitted. However, the plot line is not broken at missing values unless SKIPMISS option is used.

I=R <type> <0> <CLM | CLI<50…99>>

Type specifies the type of regression. Specify one of these values for *type*:

L requests linear regression representing the regression equation

$$Y = \beta_0 + \beta_1 X$$

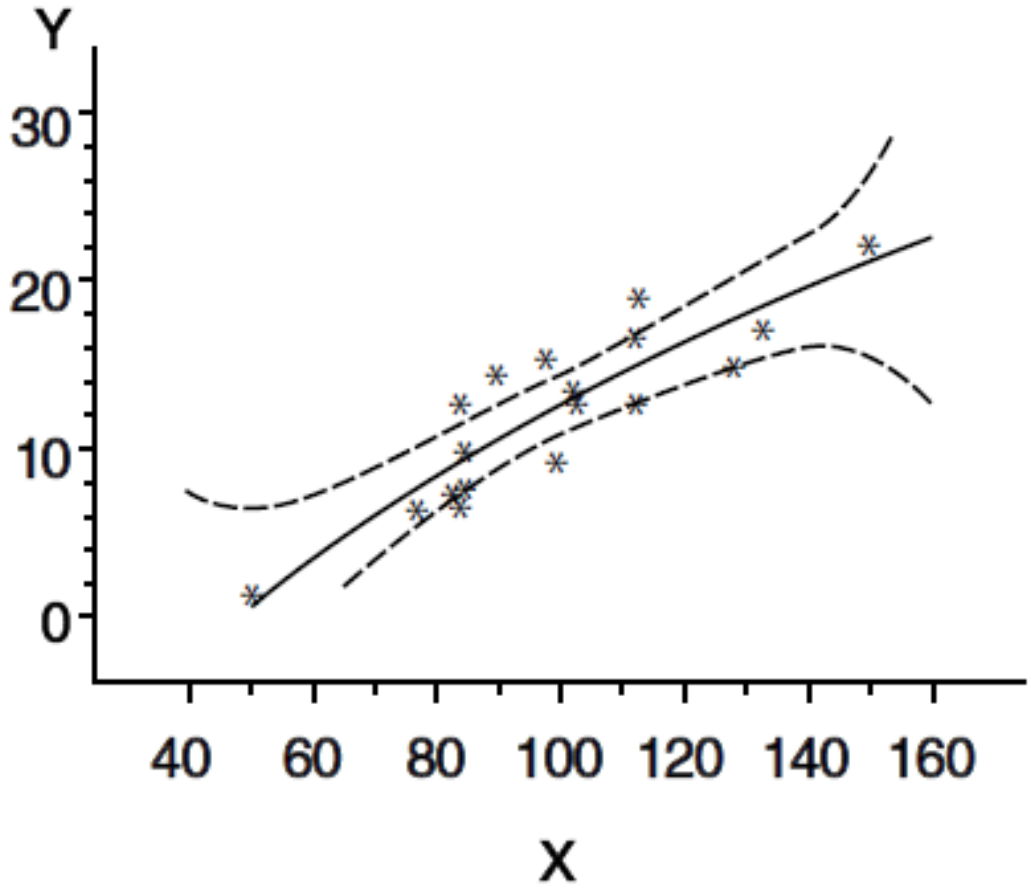
Q requests quadratic regression representing the regression equation

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2$$

C requests cubic regression representing the regression equation

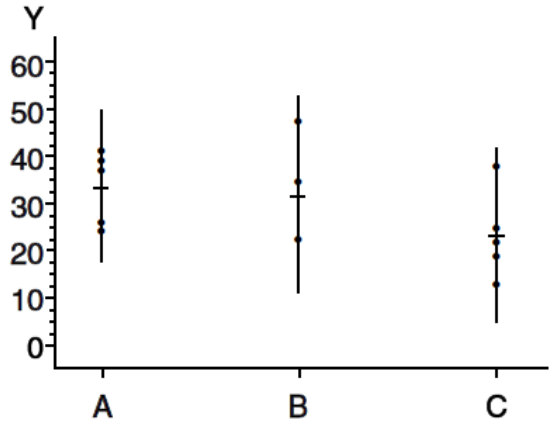
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$$

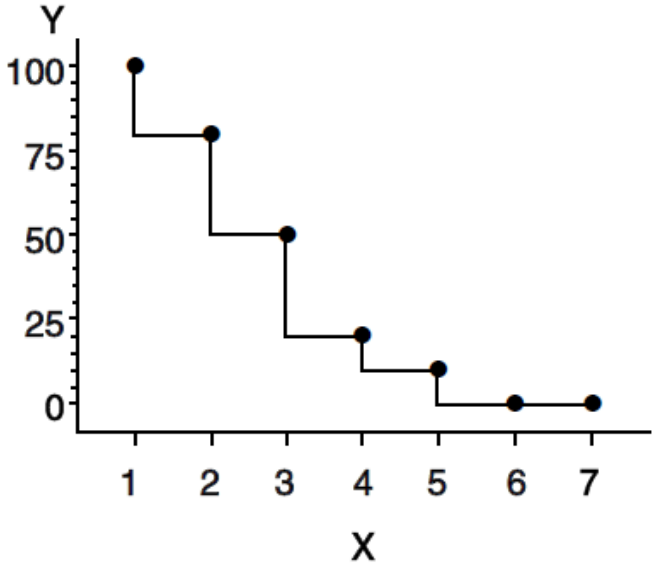
I=R



Other Interpol

- I=SM
- I=Needle
- I=STD <1 | 2 | 3>
- I=STEP

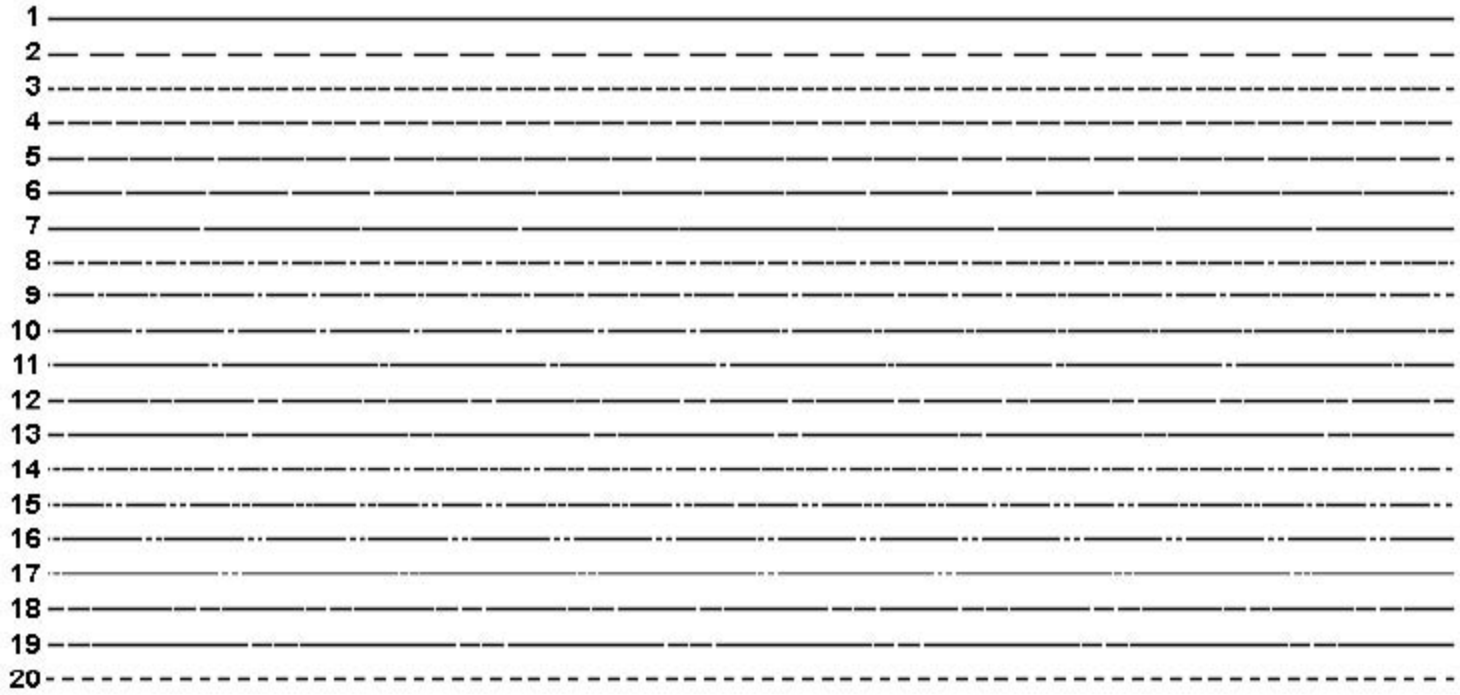




Values

VALUE=	Plot Symbol	VALUE=		Plot Symbol
PLUS	+	%	(percent)	♣
X	×	&	(ampersand)	♧
STAR	✳	'	(single quote)	⚜
SQUARE	□	=	(equals)	☆
DIAMOND	◇	-	(hyphen)	⊙
TRIANGLE	△	@	(at)	♉
HASH	#	*	(asterisk)	♀
Y	Y	+	(plus)	⊕
Z	Z	>	(greater than)	♂
PAW	⁘	.	(period)	♃
POINT	·	<	(less than)	♄
DOT	●	,	(comma)	☉

Line <1 … 46>



Samples

PROC GCHART<DATA=*input-data-set*>
<ANNOTATE=*Annotate-data-set*>
<GOUT=<*libref.*>*output-catalog*>
<IMAGEMAP=*output-data-set*>;

BLOCK *chart-variable(s)* </ *option(s)*>;

HBAR | HBAR3D | VBAR | VBAR3D*chart-variable(s)* </ *option(s)*>;

PIE | PIE3D | DONUT *chart-variable(s)* </ *option(s)*>;

STAR *chart-variable(s)* </ *option(s)*>;

Samples

PROC GPLOT <DATA=*input-data-set*>
<ANNOTATE=*Annotate-data-set*>
<GOUT=<*libref.*>*output-catalog*>
<IMAGEMAP=*output-data-set* >
<UNIFORM>;

BUBBLE *plot-request(s)* </*option(s)*>;
BUBBLE2 *plot-request(s)* </*option(s)*>;

PLOT *plot-request(s)* </*option(s)*>;
PLOT2 *plot-request(s)* </*option(s)*>;

Statistical Graphics Principles

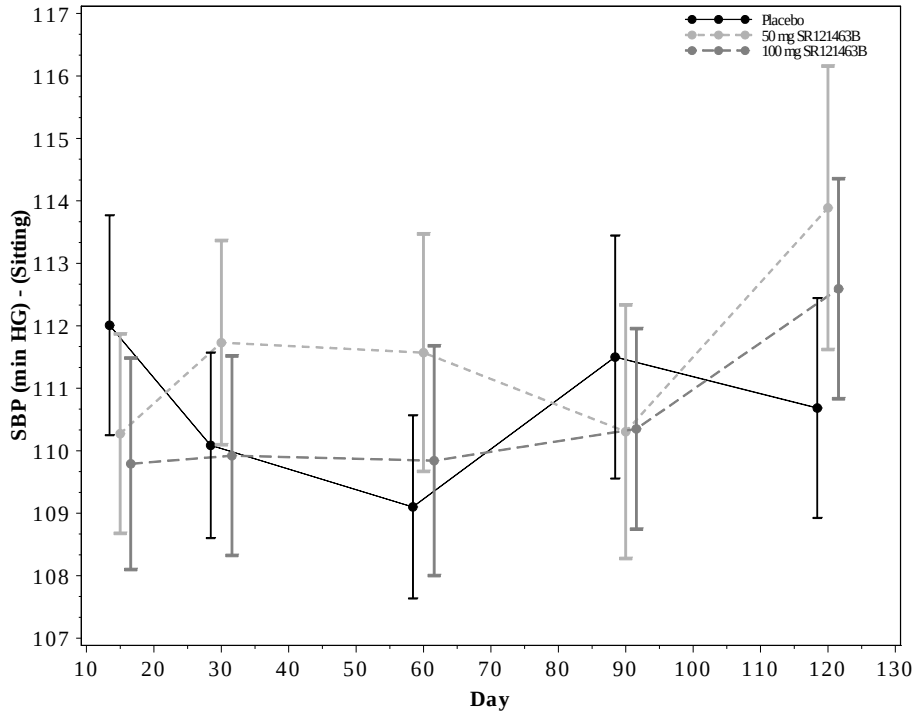
- Excellence in statistical graphics consists of complex ideas communicated with **clarity**, **precision** and **efficiency** (Tufte, 1983)

- Tufte's graphics principles for accurate and retained information extraction
  - Show all the data
  - Induce the viewer to think about the substance rather than the graphic design – maximize the data-to-ink ratio
  - Avoid distorting what the data are saying
  - Make large data sets coherent
  - Encourage the eye to compare different pieces of data
    - leverage investment by showing multiple plots of same type
  - Reveal the data at several levels of detail, from a broad overview to the fine structure
  - Serve a clear purpose: description, exploration, tabulation
  - Be closely integrated with the statistical and verbal descriptions of a data set
  - Use gray scale and color sparingly

Gold Standards of Good Graphics

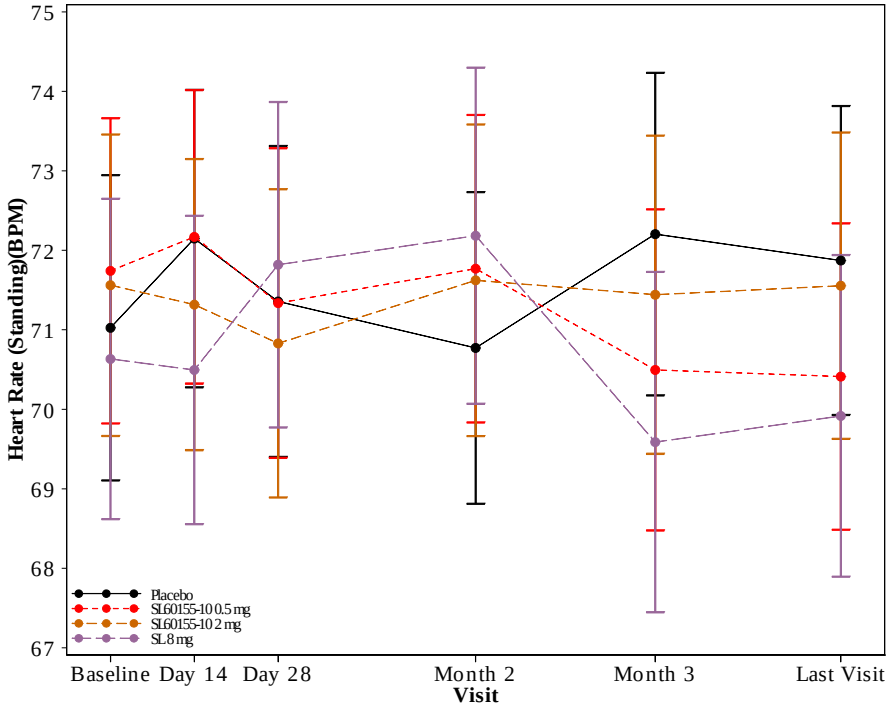
- Clarity
- Precision
- Efficiency

Clarity

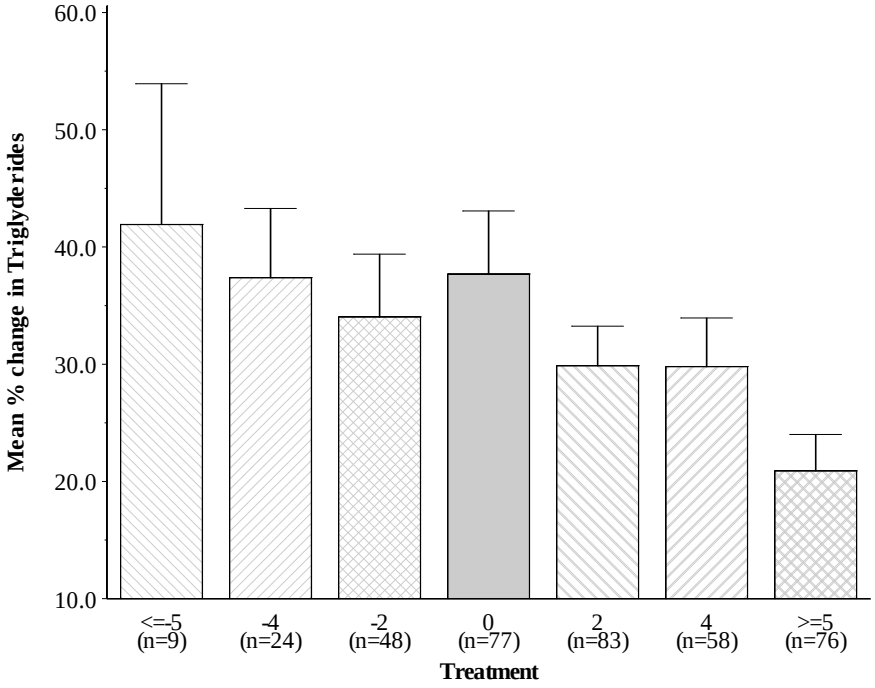


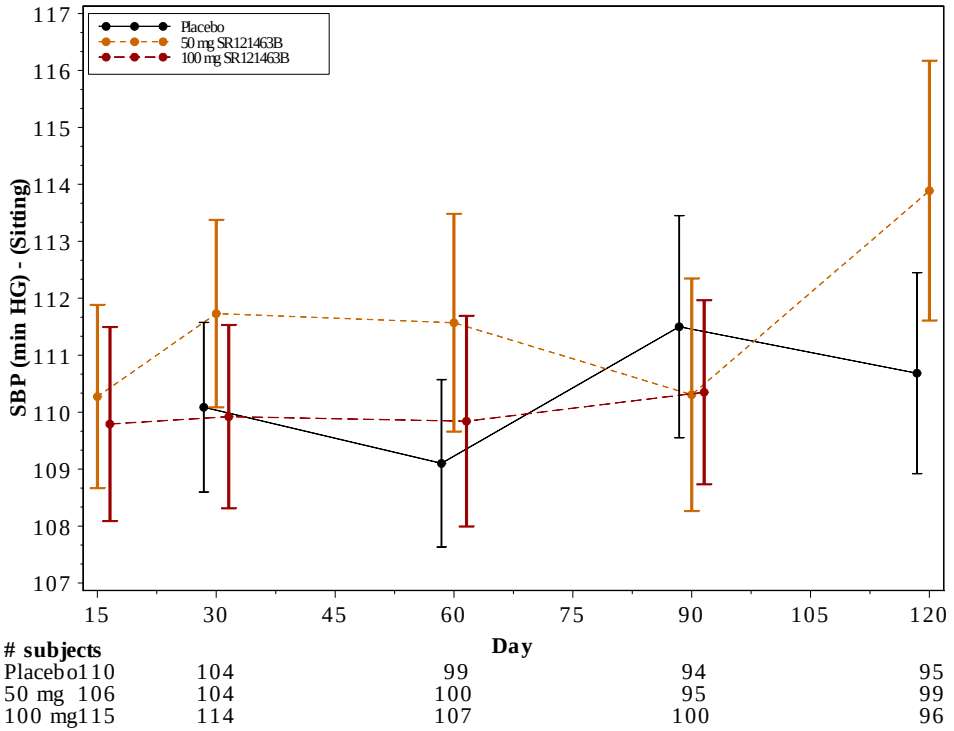
Which one is more clear?

Upper or Bottom?

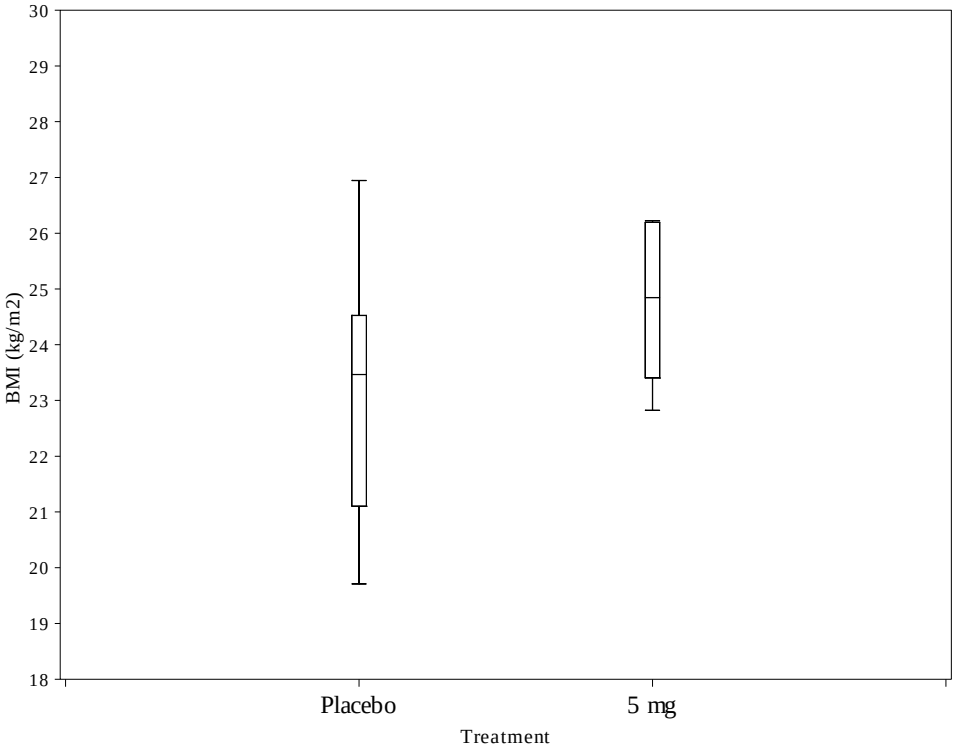


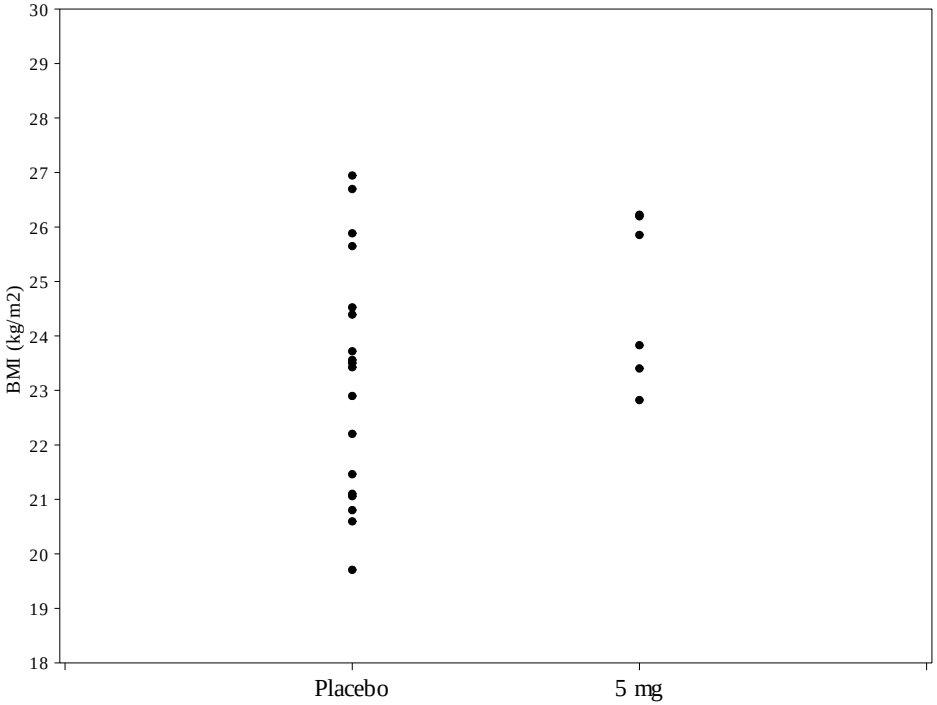
Precision


Mean % change in Triglycerides
60.0
50.0
40.0
30.0
20.0
10.0
<=-5 (n=9)
-4 (n=24)
-2 (n=48)
0 (n=77)
2 (n=83)
4 (n=58)
>=5 (n=76)
Treatment


Placebo
50 mg SR121463B
100 mg SR121463B
SBP (min HG) - (Sitting)
117
116
115
114
113
112
111
110
109
108
107
15
30
45
60
75
90
105
120
Day
subjects
Placebo 110 104 99 94 95
50 mg 106 104 100 95 99
100 mg 115 114 107 100 96

Efficiency


30
29
28
27
26
25
24
23
22
21
20
19
18
BMI (kg/m2)
Placebo
5 mg
Treatment


30
29
28
27
26
25
24
23
22
21
20
19
18
BMI (kg/m2)
Placebo
5 mg
Treatment

Good Plotting Practices

- Using two-dimensional graphs to display one-dimensional data (such as three-dimensional histograms) is inefficient and often results inperceptional ambiguity;
- Important events or data points of particular interest should be labeled on the graph and not just as some reference in the legend;
- Always use clear, concise, detailed labeling. Avoid computer labels;
- Maximize the data-ink ratio, i.e., the proportion of a graph’s ink that is devoted to the display of non-redundant information;

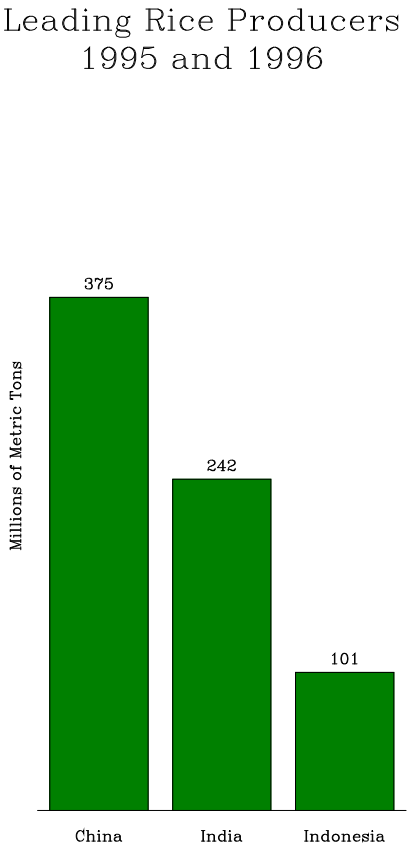
Good Plotting Practices (cont'ed)

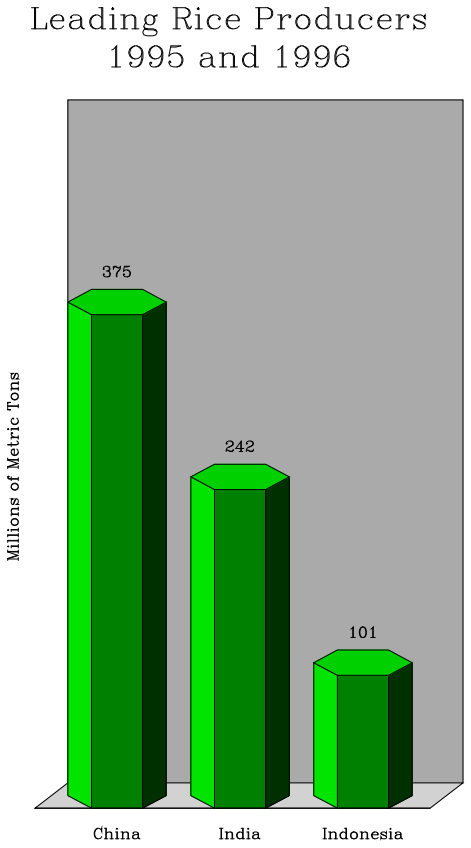
- If a grid is needed, do not make the grid color black or the line width the same width as other lines or symbols in the graph. Use a gray color instead to bring the data to the foreground;
- Avoid confusing legends;
- Plots should have greater length than height, the so-called golden rectangle, since our eye is naturally better at detecting trends on the horizontal than the vertical and it is easier to place labels in a wider graph than a thinner graph;
- Use colors to identify data properly.

Graphics Principles for Clinical Data

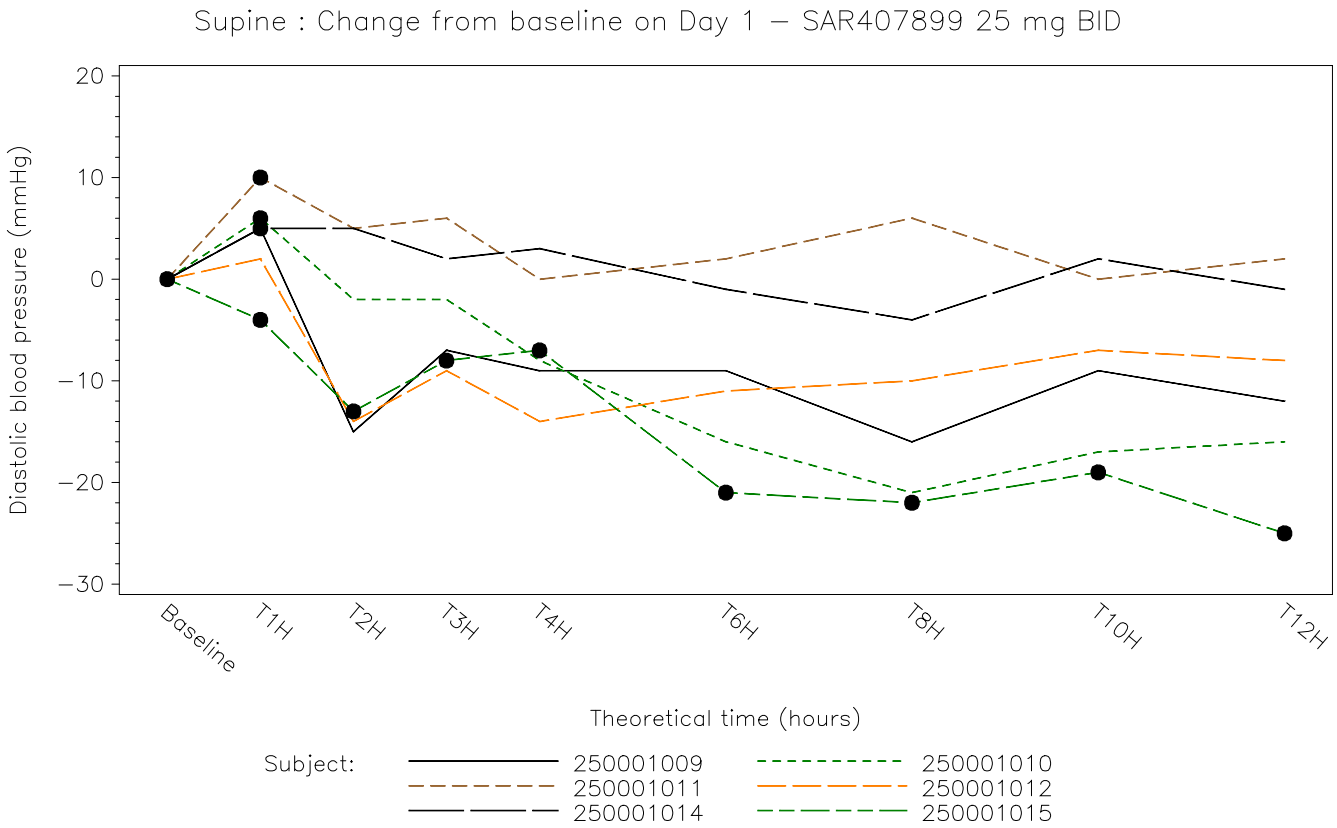
- Set a sound aspect ratio
- Include (feint) grid lines and reference lines as appropriate
- Sort categories e.g. on dot plots
- Use group and multi-panel displays (with same scale)
- Minimize use of legends by annotating the graph
- Don’t use more than 3 hues
- Use contrast hues (blue-orange/yellow); avoid color-blind hues (red-green)
- Use black and gray for 2 group comparisons (treatment and control)
- Don’t use to many dashed line types – they are difficult to distinguish

Good Plotting Practices





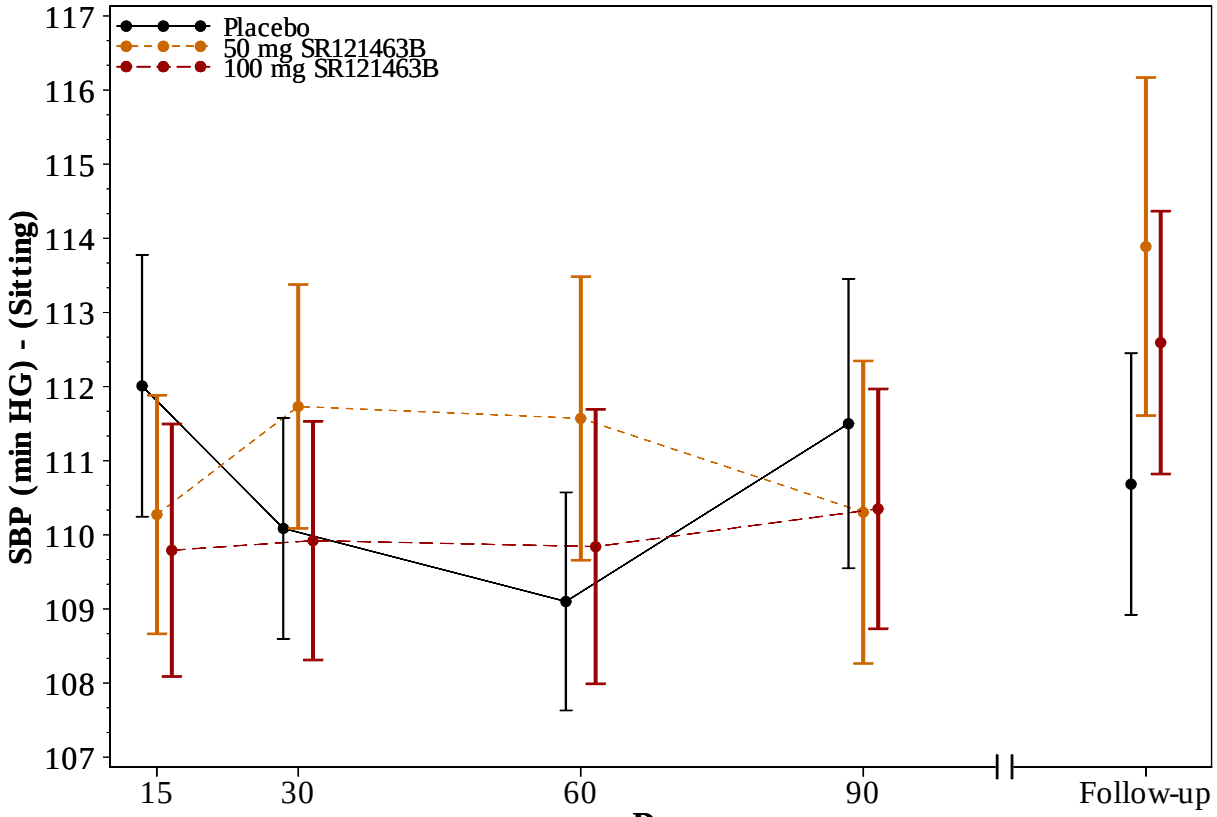
Good Plotting Practices (cont'ed)



Baseline is the Day 1 predose value.
Only planned values are included in the plot.

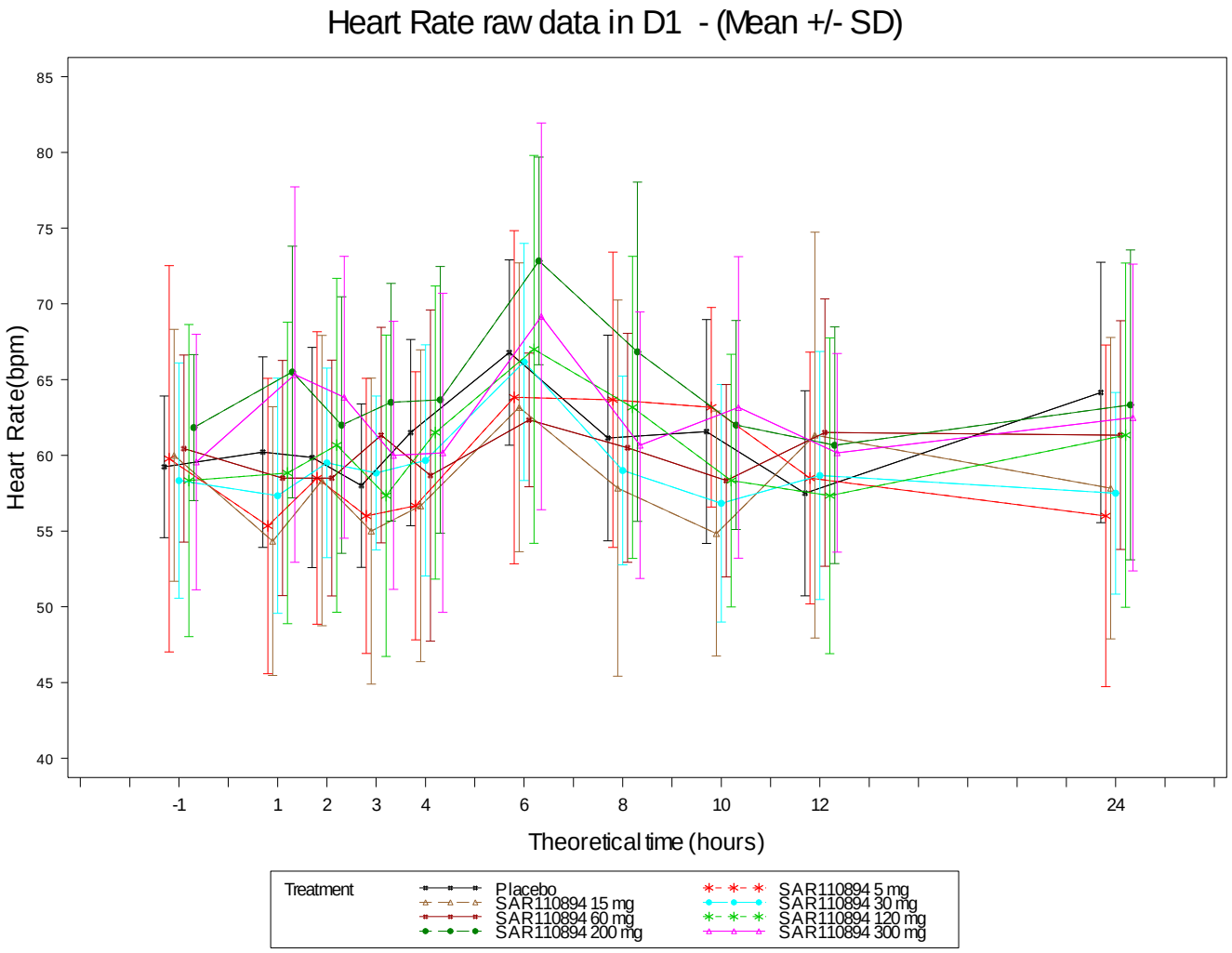
● : Value was measured with time deviations (<8 min in supine or not 2, 3 or 4 min in standing position).

Good Plotting Practices (cont'ed)

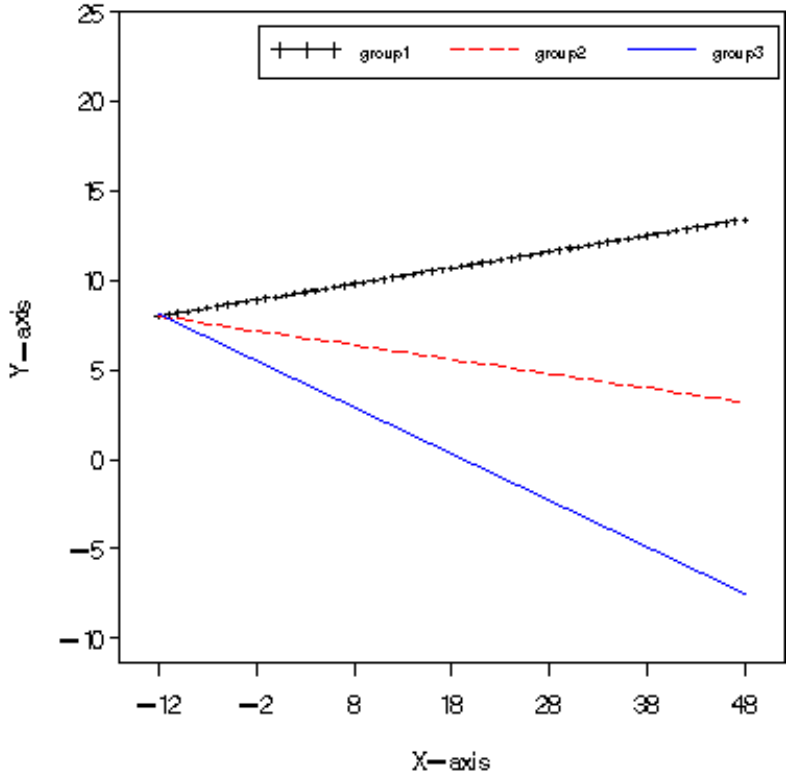


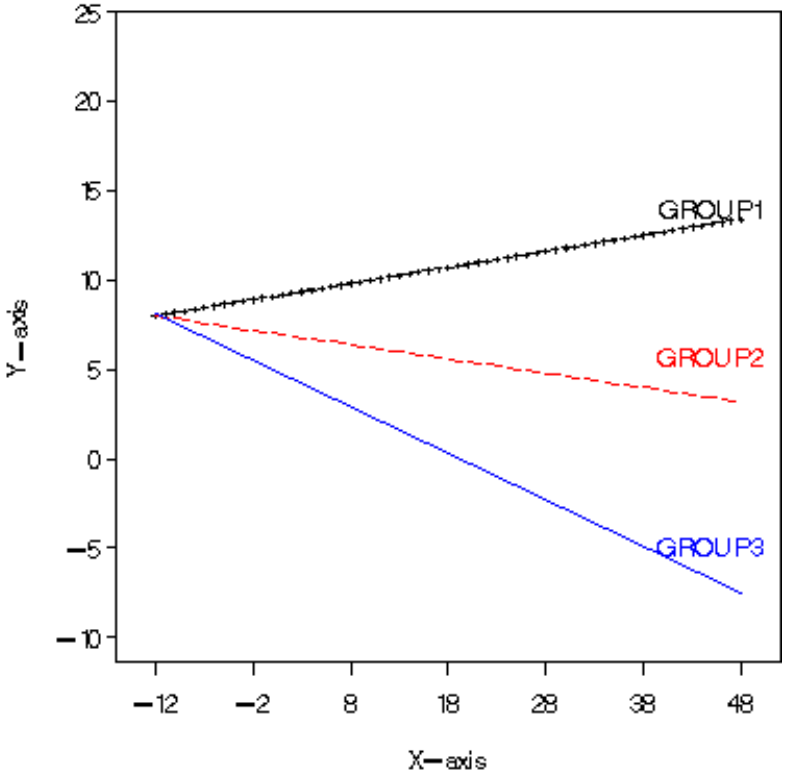
# subjects					
Placebo	110	104	99	94	95
50 mg	106	104	100	95	99
100 mg	115	114	107	100	96

Good Plotting Practices (cont'ed)

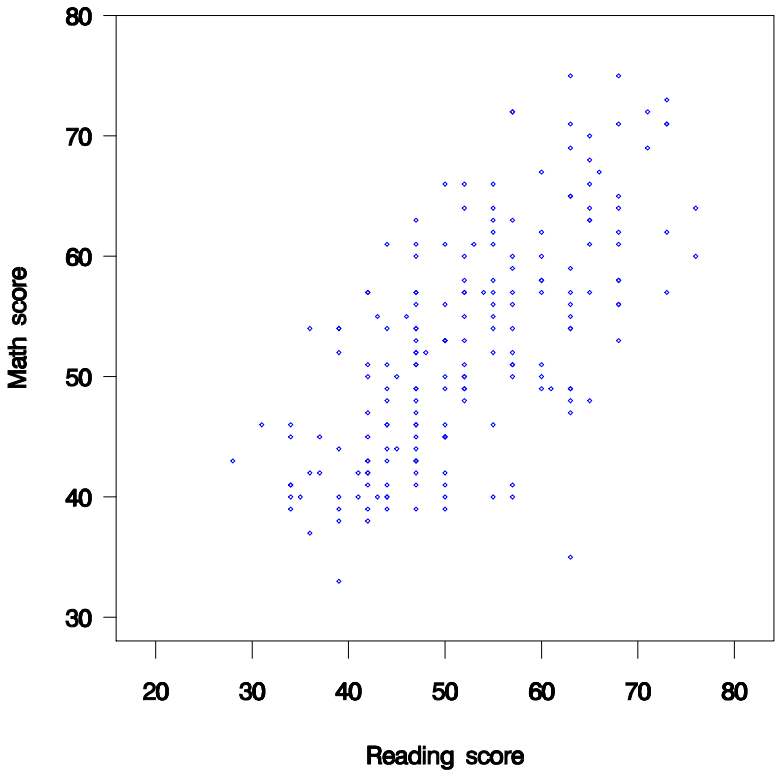


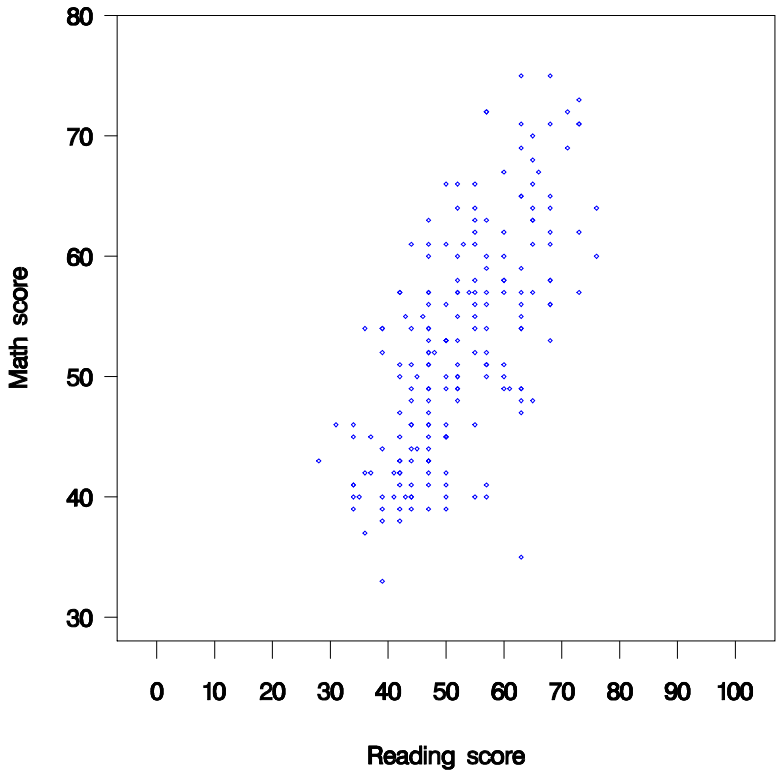
Good Plotting Practices (cont'ed)



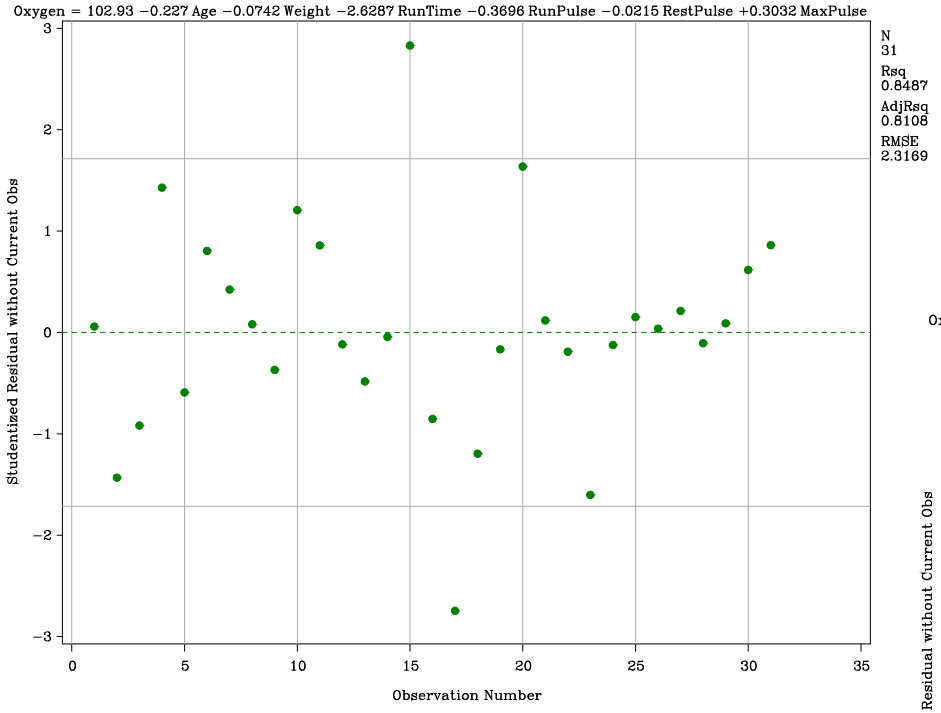


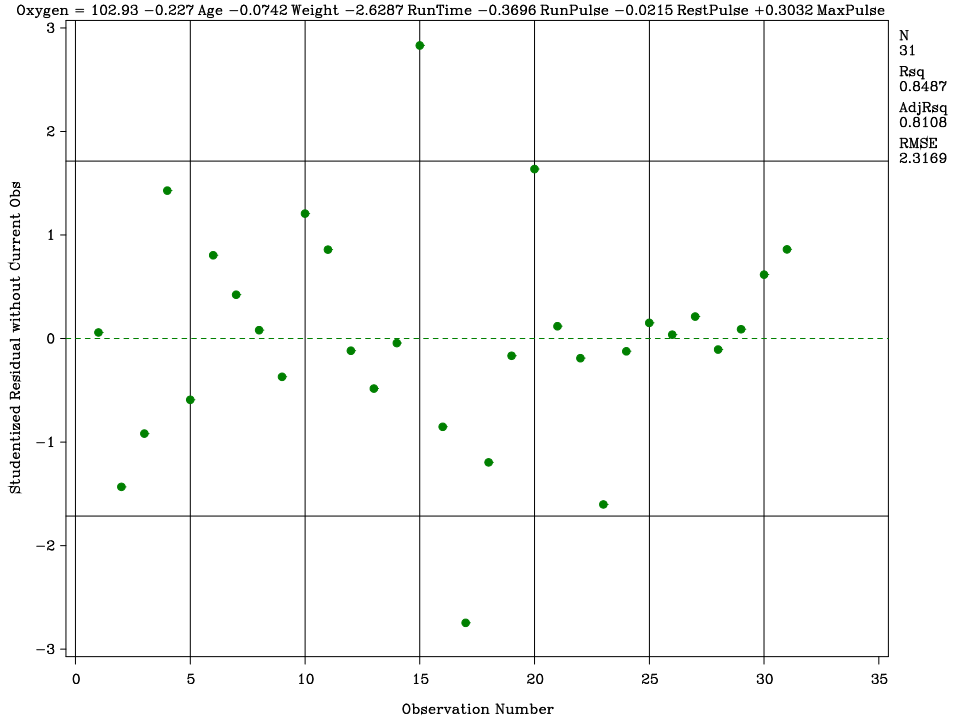
Good Plotting Practices (cont'ed)





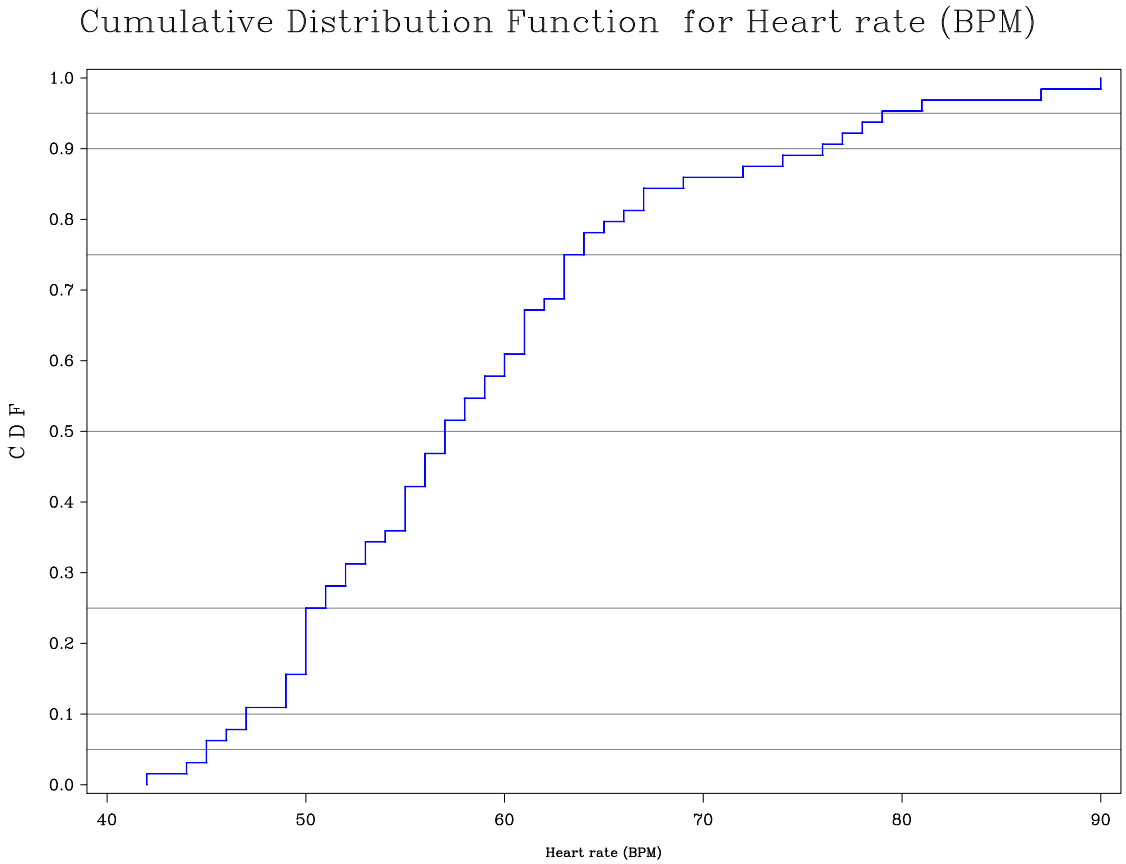
Good Plotting Practices (cont'ed)



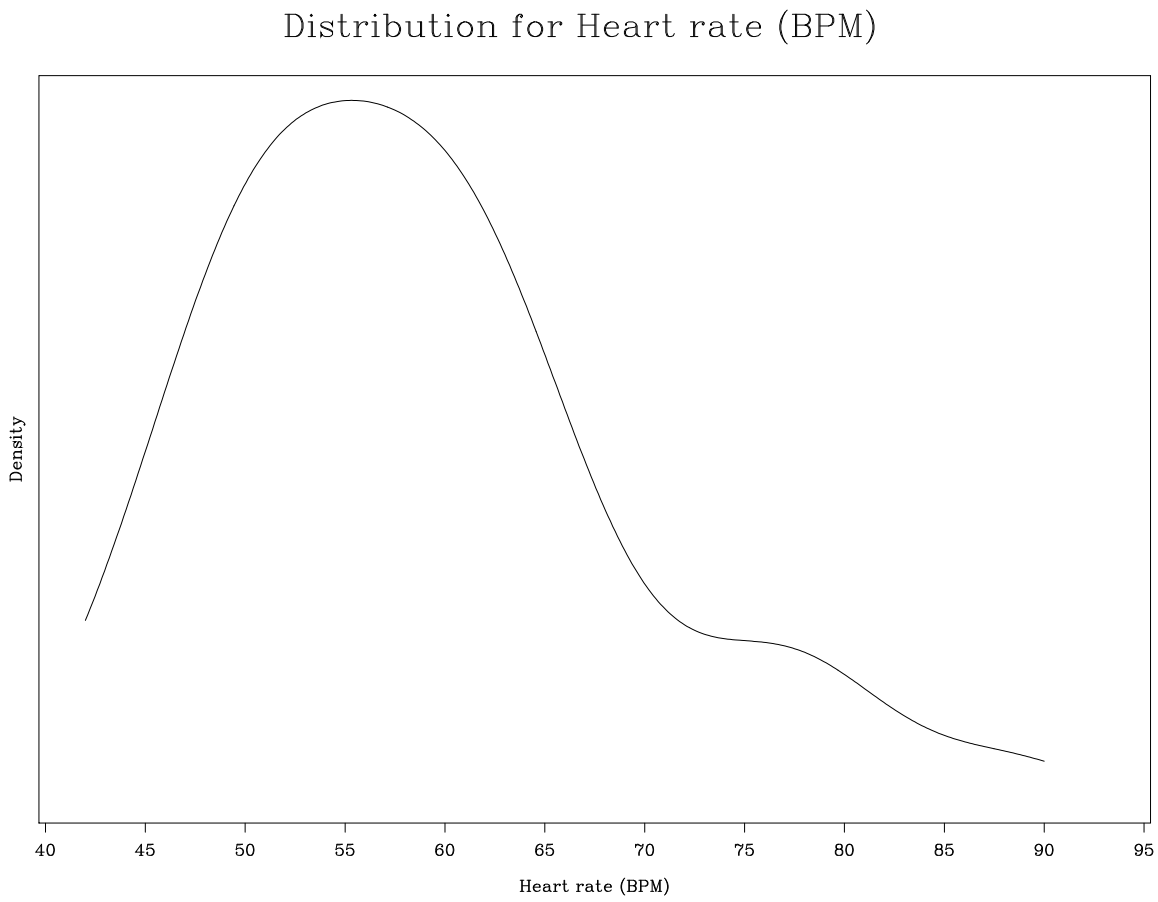


CDF plot using i = steplj in SYMBOL statement

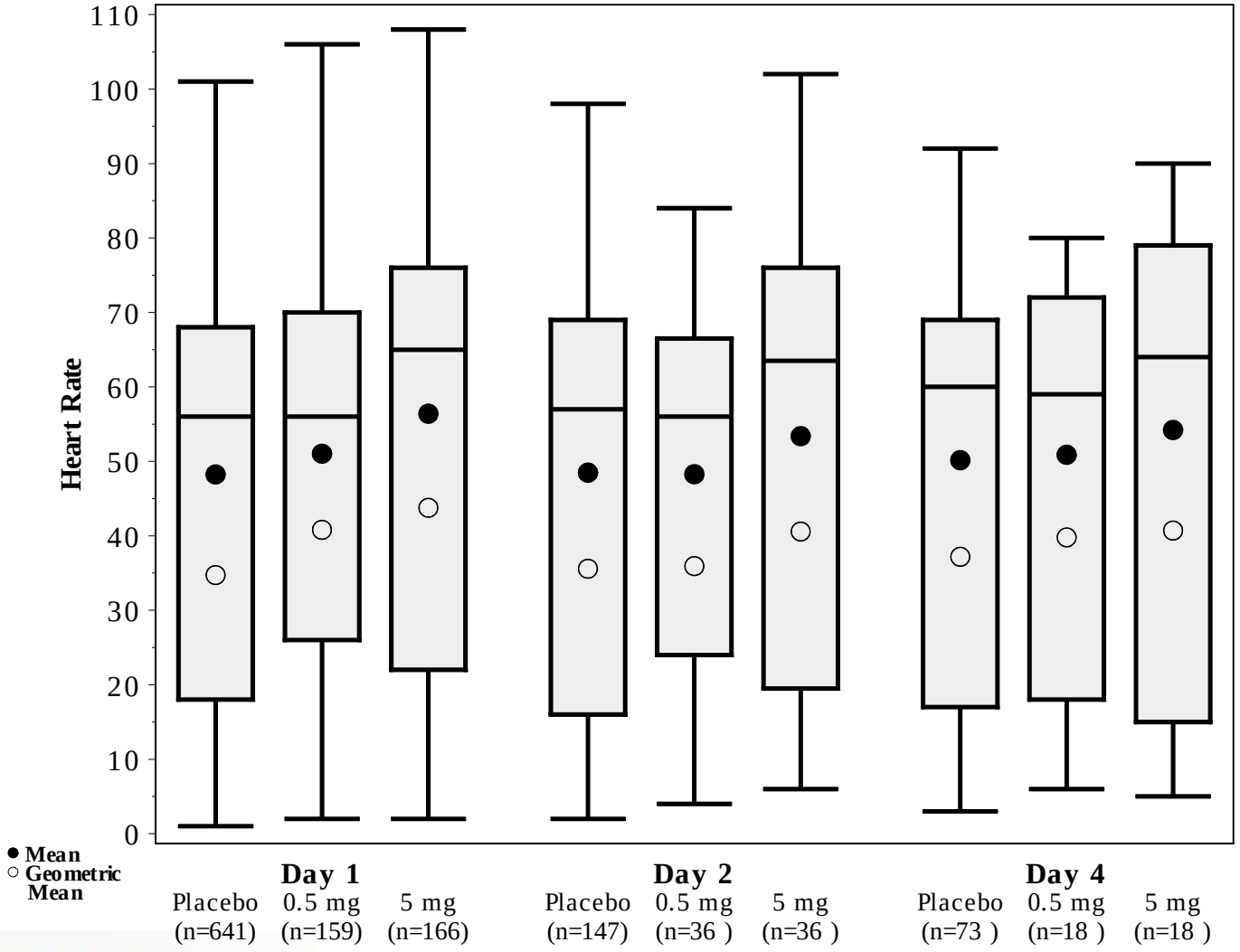
C:\Documents and
ettings\i0037128.RES



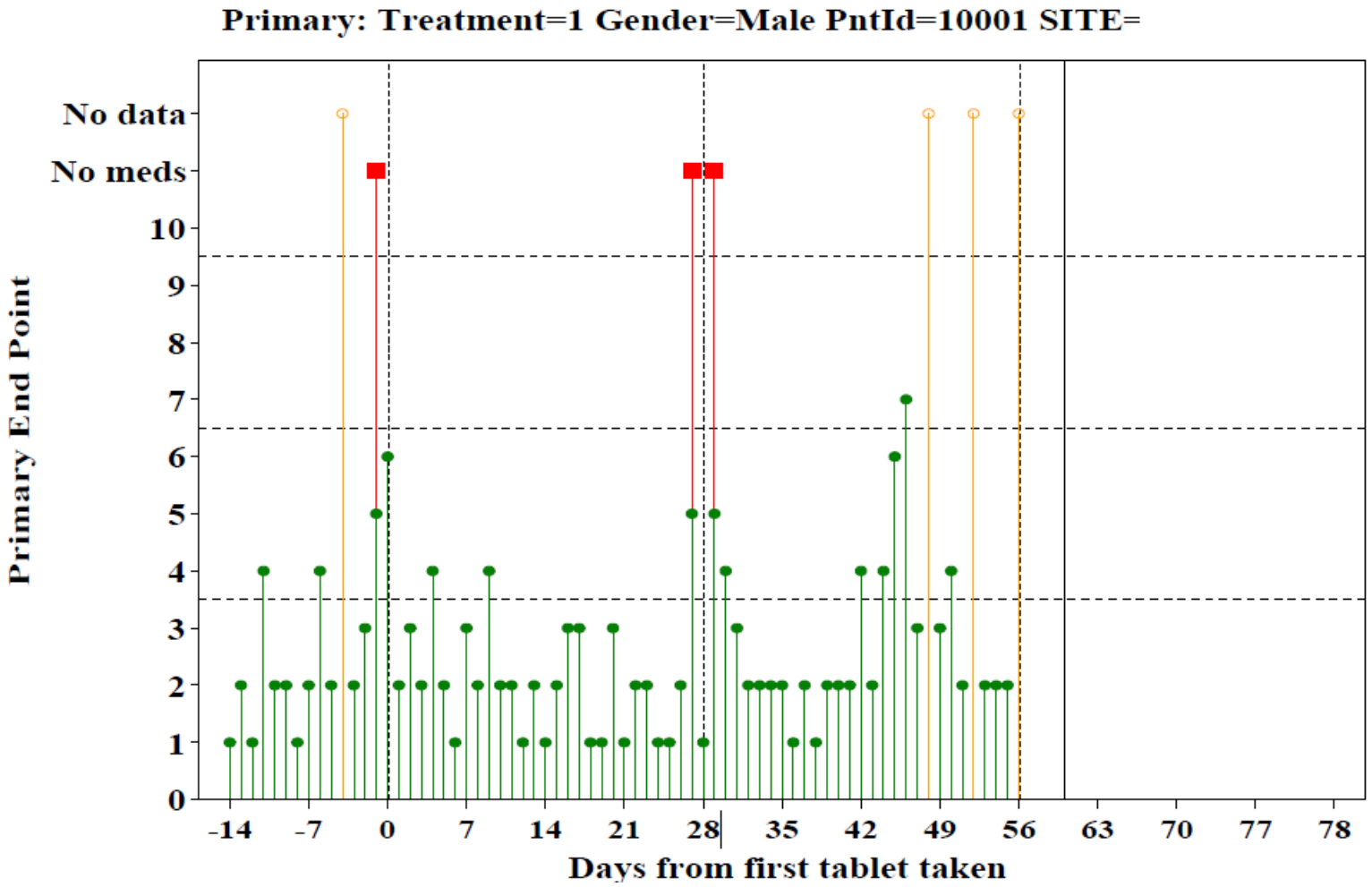
Density plot using i = join in SYMBOL statement



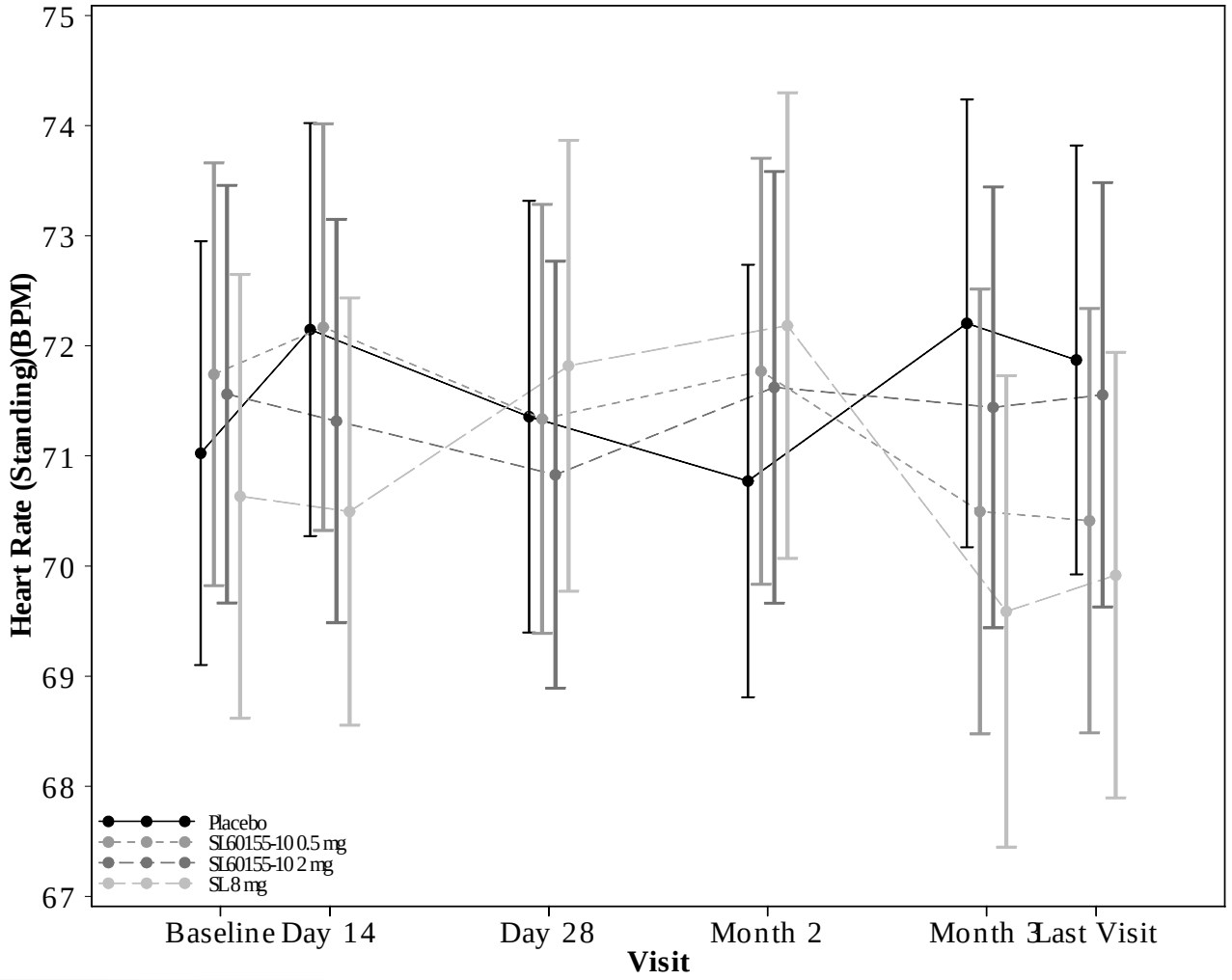
Boxplot using i = boxt in SYMBOL statement



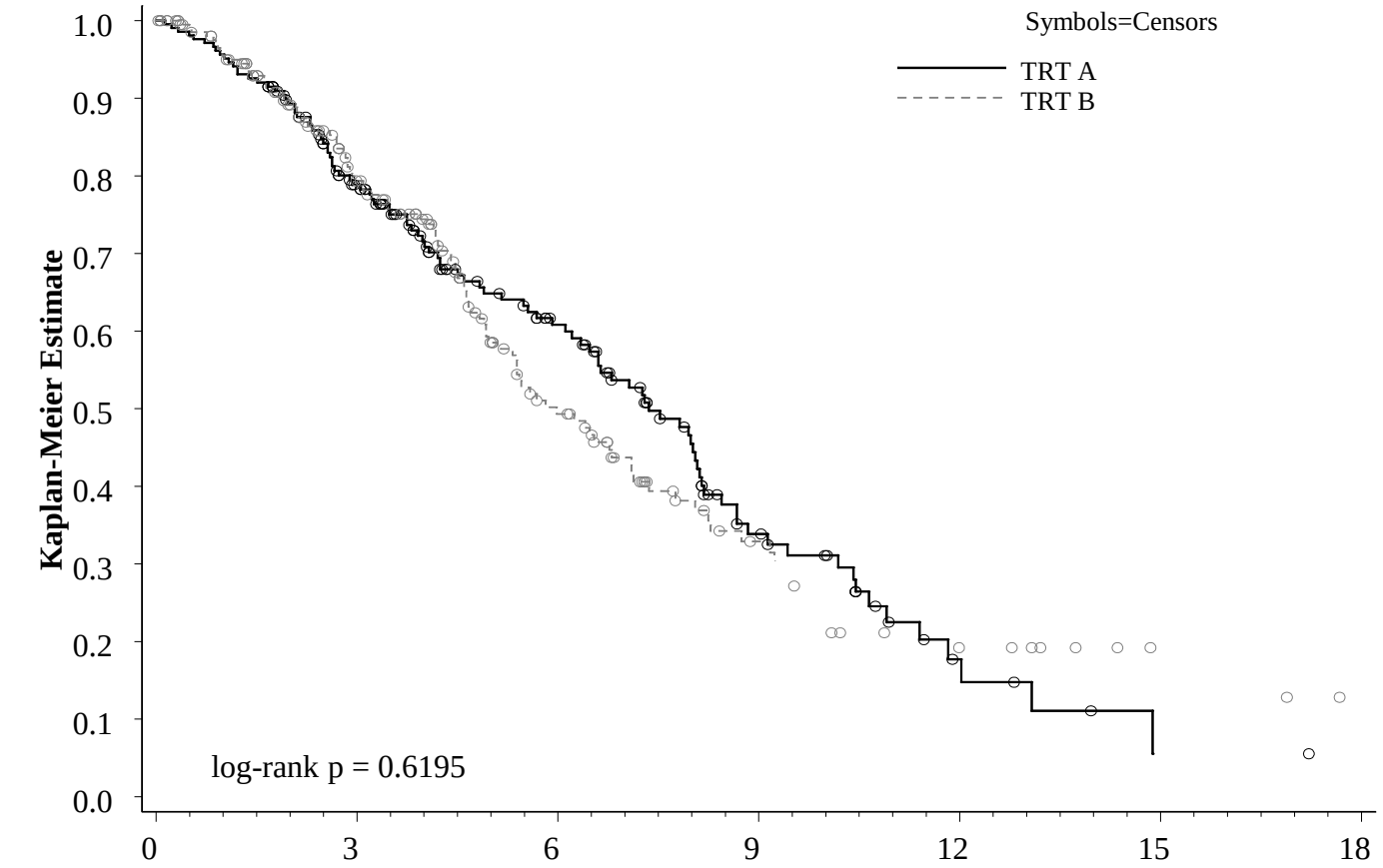
Needle plot using i = needle in SYMBOL statement



High-low plot using i = hilocjt in SYMBOL statement



KM plot using i = steplj in SYMBOL statement



What's the Annotate Facility

- The annotate facility enables you to generate a special data set of graphics commands from which you can produce graphics output
- This data set is referred to as an ***Annotate data set***
- Use it to generate custom graphics or to enhance graphics output from many SAS/GRAPH procedures, including GCHART, GCONTOUR, GMAP, GPLOT and so on.
- The Annotate Facility acts as a bridge between the procedure selected by the user and the user's desire to customize the graphics output.

The Annotate Data Table

- The Annotate data set is not different from any other SAS data set.
- In an Annotate data set, each observation represents a command to draw a graphics element or to perform an action. The graphic elements drawn by these commands can be added to SAS/GRAPH output or displayed with the GANNO or GSLIDE procedure as a custom graphic
- The observations in an Annotate data set use a set of predefined Annotate variables. The values of the variables in the observation determine what is done and how it is done.
- To create these observations, you assign values to the variables either explicitly with a DATA step or implicitly with Annotate macros.

OBS	FUNCTION	X	Y	HSYS	XSYS	YSYS	STYLE	COLOR	POSITION	SIZE	LINE	TEXT
1	label	20	85	3	3	3	swissb	green	6	6.0	.	Sample Annotate Graphics
2	move	28	30	3	3	3	swissb	green	6	6.0	.	Sample Annotate Graphics
3	draw	68	30	3	3	3	swissb	red	6	0.8	1	Sample Annotate Graphics
4	draw	48	70	3	3	3	swissb	red	6	0.8	1	Sample Annotate Graphics
5	draw	28	30	3	3	3	swissb	red	6	0.8	1	Sample Annotate Graphics

What can the Annotate Facility do

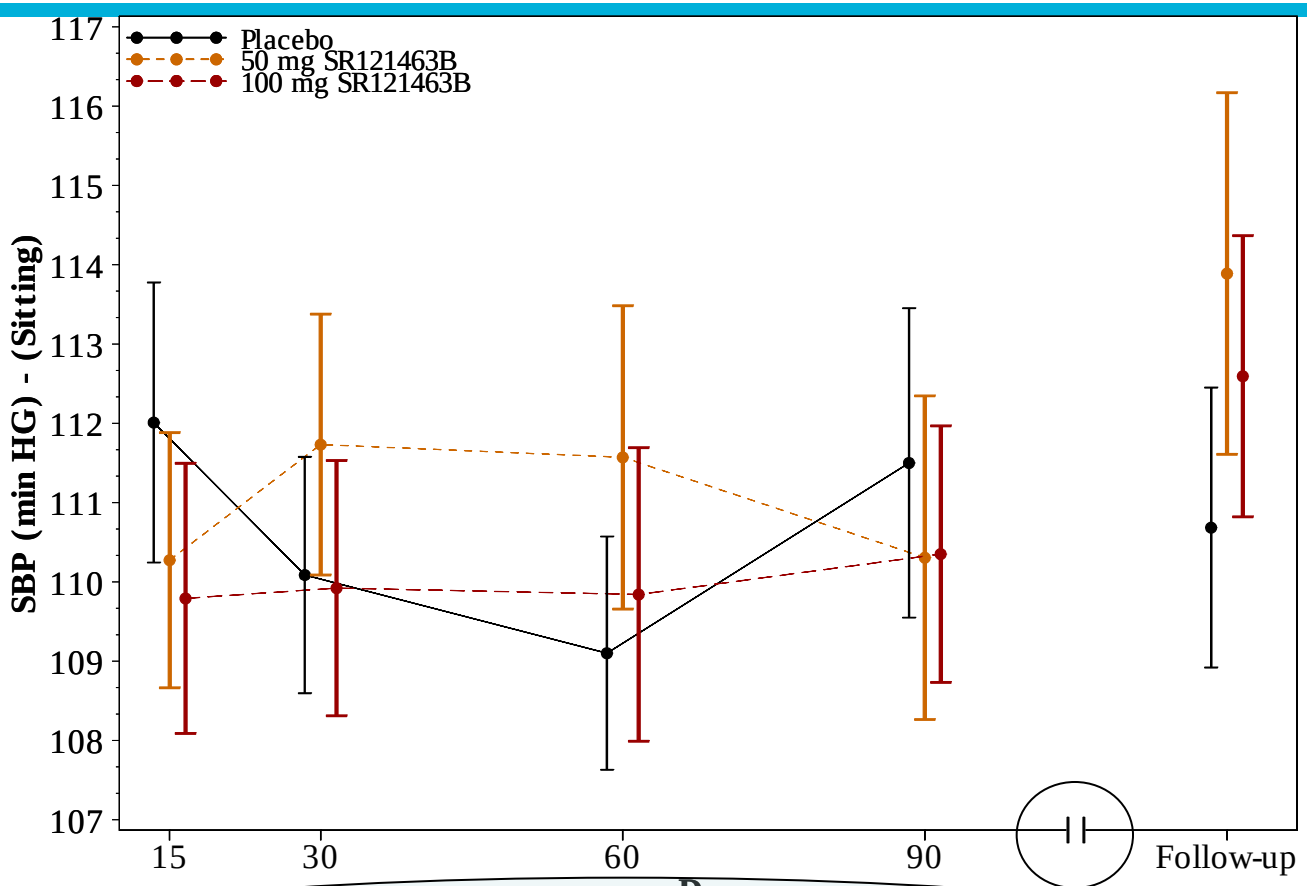
- Enhances output from SAS/GRAPH procedures by adding graphics elements to the output.
  - label points on a map using map coordinates
  - label bars on horizontal and vertical bar charts
  - label points on a plot
  - create a legend for a three-dimensional graph
- Use an Annotate data set to create custom graphics
  - create various types of graphs (including pie charts, bar charts, and plots)
  - draw graphics elements such as lines, polygons, arcs, symbols, and text

Examples(1)


Distribution Center Locations
Seattle
Chicago
Atlanta


Flag of Micronesia

Examples(2)



# subjects					
Placebo	110	104	99	94	95
50 mg	106	104	100	95	99
100 mg	115	114	107	100	96

The Annotate Process

- The annotate is used primarily to enhance a graph. First, we must know three questions for use of the annotate.
  - What is to be done?
  - Where is it to be done?
  - How is it to be done?
- The annotate process is the same as answering the three questions above
- Some of the variables that are used to answer these three questions include:
  - What : FUNCTION (this is the ONLY variable used to answer this question)
  - How: COLOR, SIZE, STYLE, POSITION
  - Where: X, Y, XSYS, YSYS

WHAT, HOW, and WHERE(1)

WHAT is to be done?

- An action variable FUNCTION tells what to do

- The character variable FUNCTION provides the information on WHAT is to be done. The FUNCTION variable accepts a set of predefined values (functions) that perform both graphics tasks and programming tasks

- Some of the common values of the variable FUNCTION
  - BAR draw a rectangle
  - DRAW draw a line
  - LABEL places text or symbols on the graphic
  - MOVE allows movement to a specific point on the graphic
  - SYMBOL places a symbol on the graphic
  - POINT places a single point

WHAT, HOW, and WHERE(2)

HOW is it to be done?

- Attribute variables tell how to do it

- The attribute variables specify the characteristics of the graphics element (for example, color, size, line style, text font)

- some of the attribute variables include
  - TEXT='*string*' add *string* to display
  - COLOR='*color*' specify the color of text
  - SIZE=*n* size of text
  - STYLE='*font*' select font for text string

WHAT, HOW, and WHERE(3)

WHERE is it to be done?

- Positioning variables tell where to do it

- The positioning variables specify the point at which to draw the graphics element

- some of the positioning variables include
  - X specifies a numeric horizontal coordinate
  - Y specifies a numeric vertical coordinate
  - XSYS specifies coordinate system for X or XC coordinates
  - YSYS specifies coordinate system for Y or YC coordinates

WHAT, HOW, and WHERE(4)

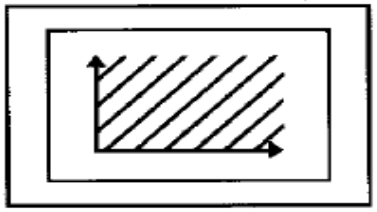
WHERE is it to be done?

- The coordinates are usually placed using the numeric variables X and Y

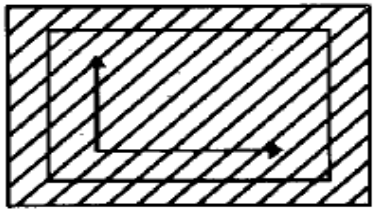
How these coordinates are interpreted depends on the coordinate system, which is specified by the XSYS and YSYS variables?

Coordinate System(1)

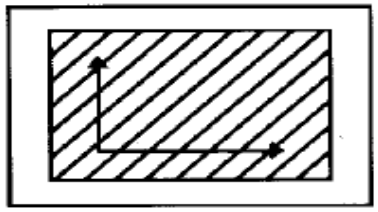
The coordinate system itself can be selected by using the character variables XSYS and YSYS, these variables can take on one of twelve 'system' values, two of these values for XSYS and YSYS will satisfy most of your Annotate needs



Area	Unit	Coordinate System	
Data		Absolute	Relative
	%	1	7
	Values	2	8



Area	Unit	Coordinate System	
Graphics Output Area		Absolute	Relative
	%	3	9
	Cells	4	A

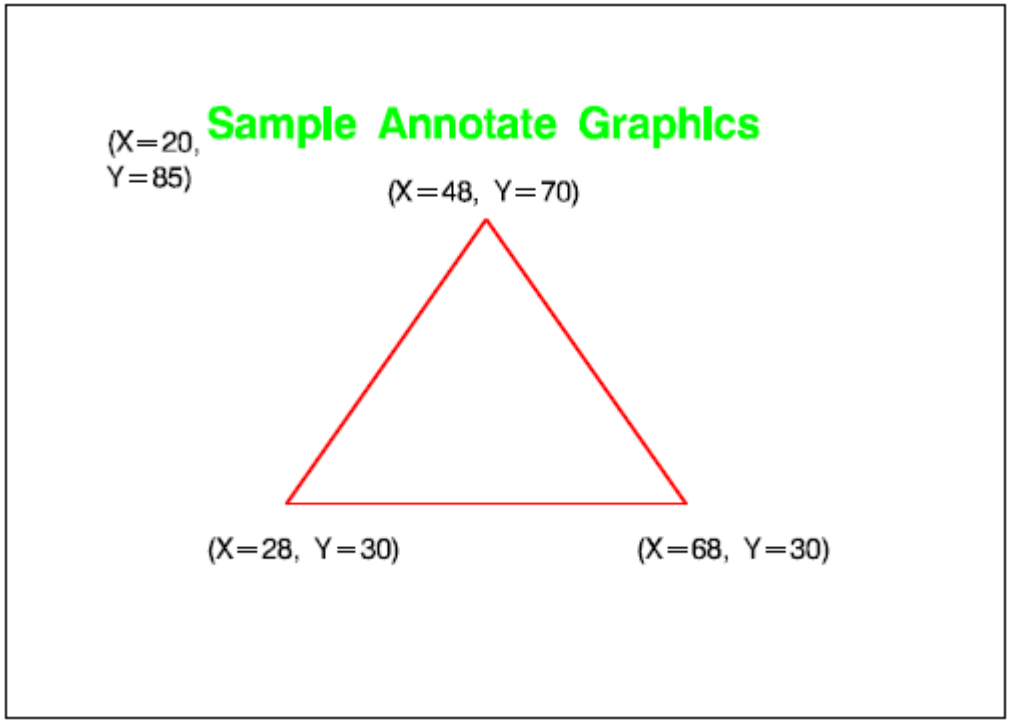


Area	Unit	Coordinate System	
Procedure Output Area		Absolute	Relative
	%	5	B
	Cells	6	C

Coordinate System(2)

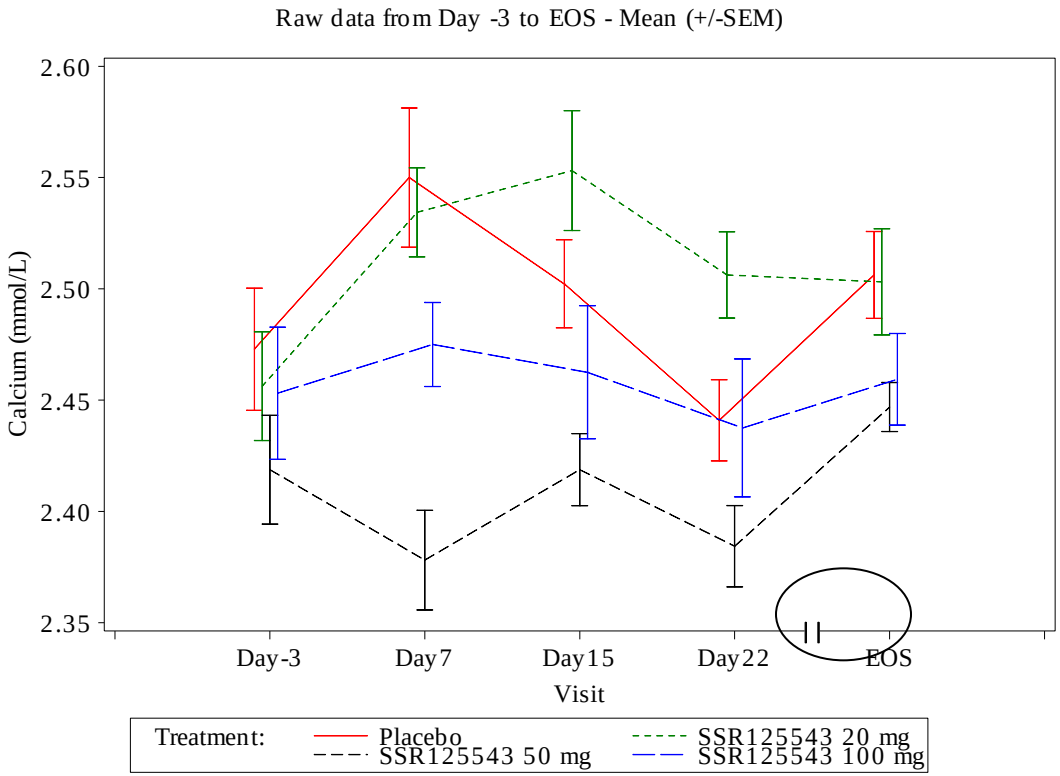
- A particular value of X or Y will be located depends on the value assigned to XSYS or YSYS
- The values of XSYS and YSYS need not be constant in the Annotate data set. XSYS and YSYS do not need to have the same values.
- E.g.
  - XSYS & YSYS='2' 'Absolute data value' places the point according to the values of the horizontal and vertical axes that are plotted on the graph.
  - XSYS & YSYS='3' 'Absolute Graphics Output Area percent' uses percentages of the entire graphics area, which are measured from the lower left corner.

Annotate Example(1)



	function	style	color	text	hsys	xsys	ysys	x	y	position	size	line
1	label	swissb	green	Sample Annotate Graphics	3	3	3	20	85	6	6	.
2	move	swissb	green	Sample Annotate Graphics	3	3	3	28	30	6	6	.
3	draw	swissb	red	Sample Annotate Graphics	3	3	3	68	30	6	0.8	1
4	draw	swissb	red	Sample Annotate Graphics	3	3	3	48	70	6	0.8	1
5	draw	swissb	red	Sample Annotate Graphics	3	3	3	28	30	6	0.8	1

Annotate Example(2)



D:\Work\MACRO\
annoBreakSym.sas

C:\Documents and
ettings\i0037128.RES

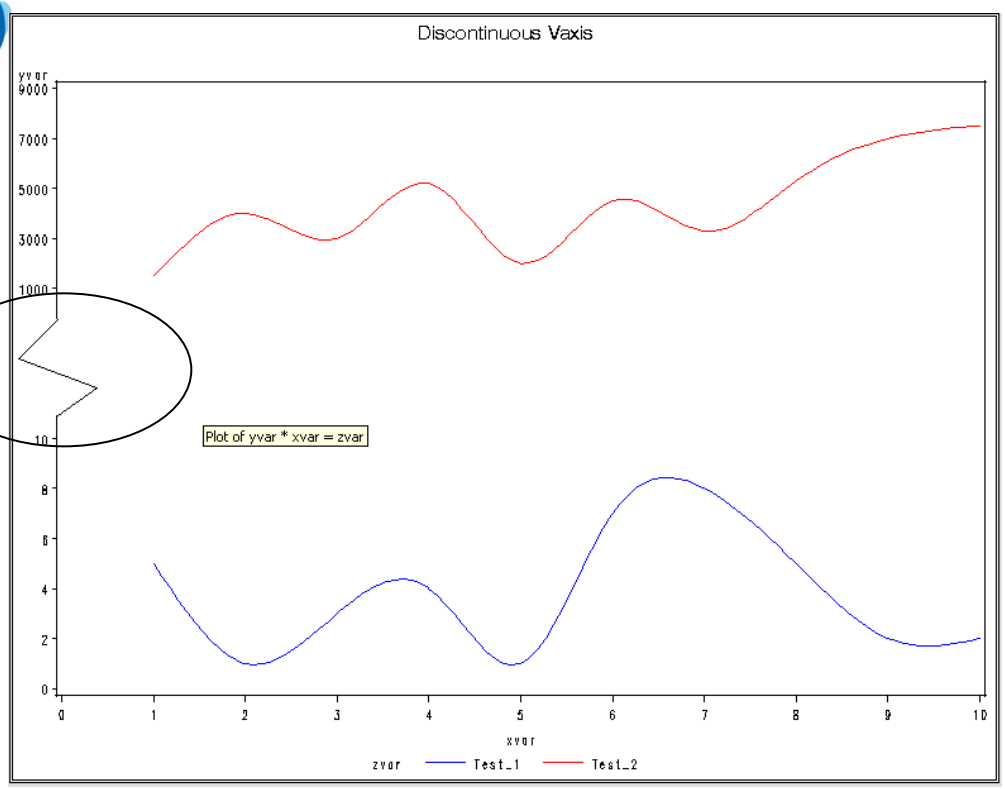
Baseline is Dav -3

	STYLE	FUNCTION	COLOR	XSYS	YSYS	HSYS	WHEN	POSITION	X	Y	LINE	SIZE
1		MOVE	black	2	3	3	a	5	22	7	.	.
2		MOVE	black	7	3	3	a	5	-0.75	7	.	.
3		DRAW	white	7	3	3	a	5	1.5	7	1	1
4		MOVE	white	7	3	3	a	5	-1.5	7	1	1
5		DRAW	black	7	3	3	a	5	0.15	7	1	2.5
6		MOVE	black	7	3	3	a	5	1.2	7	1	2.5
7		DRAW	black	7	3	3	a	5	0.15	7	1	2.5

Building the Annotate Dataset(1)

- The Annotate data set can be built:
  - by using assignment statements
    - The Annotate data set can be built by using assignment statements.
    - This technique is best employed when the Annotate data set does not depend on incoming data and only a few observations are needed.
  - from an existing SAS data set (or other data source)
    - When the graphics display depends on an established SAS data set, that data set can often be used to build the Annotate data set as well.
    - This technique is especially useful when you need to place labels or text strings at a location that is to be determined by the data itself.

Building the Annotate Dataset(2)

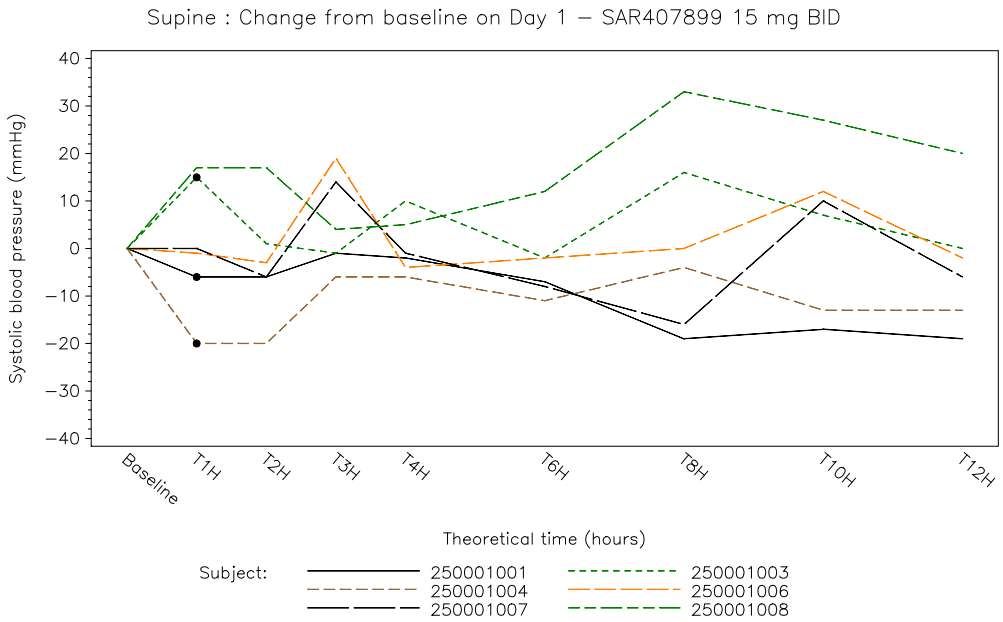


C:\Documents and
ettings\i0037128.RESt

Obs	function	style	color	xsys	ysys	when	x	y	size
1	move	solid		5	2	a	1	50	.
2	bar	solid	white	5	2	a	8	500	.
3	move	solid	black	1	2	a	0	50	2
4	draw	solid	black	B	B	a	4	4	2
5	draw	solid	black	B	B	a	-8	4	2
6	draw	solid	black	1	2	a	0	500	2

Building the Annotate Dataset(3)

Building on an Existing Data Table



C:\Documents and
ettings\i0037128.RES

Baseline is the Day 1 predose value.
Only planned values are included in the plot.

● : Value was measured with time deviations (<8 min in supine or not 2, 3 or 4 min in standing position).