

Side-by-Side Dot Box Plots in SAS^R

Alan Davis, Mary Lane, Andy Cucchiara, Don Parker
University of Oklahoma Health Science Center, Oklahoma City

ABSTRACT

For visual comparison of groups of data, Side-by-side Dot Box Plots are an important tool in exploratory data analysis. This paper shows how they can be constructed in SAS^R, and some examples, including their advantage over standard displays such as PROC CHART and PROC UNIVARIATE.

INTRODUCTION

In a recent article on data presentation, Wilkinson¹ has shown the value of dot box-plots in displaying data. Dot box plots display individual data points as well as a stem-and-whiskers plot on the same graph. In SAS^R, PROC UNIVARIATE produces box plots, stem and leaf plots, and summary box plots which are side-by-side for each level of a BY variable.

Side-by-side plots are valuable in comparing groups, and our desire is to enhance the side-by-side plot with dots, thus making side-by-side dot box plots.

Previous SAS^R users have used the ANNOTATE procedure to produce specialized graphical display. In 1985, Olmsted² describes a macro for Producing Box Plots. Benoit³ has several types of graphical displays which are of interest, also using ANNOTATE.

In 1990, Bergstralk⁴ wrote a macro for adding correlation and regression statistics to scatter plots, to make a more convenient one-page summary of data. Also, Friendly⁵ presented a macro for correlation matrix scatter plots.

COMPARING GROUPS

A common task facing the statistician or data analyst is the presentation of distributions from a large number of variables, compared over several groups. SAS^R contains a number of procedures for reporting the statistics for multiple variables, including MEANS, UNIVARIATE, SUMMARY, TABULATE, and CORR.⁶ These procedures are useful when the groups being compared are known to be characterized by simple unimodal distributions, so that they can be adequately described by summary statistics such as mean, median, standard deviation, and skewness. Particularly useful in the presentation of summary statistics is the TABULATE procedure, which is capable of producing tailor-made reports by accepting statements which describe the format of the table by type of variable, row and column definition, and type of statistic.

For example, Table 1 is the output from TABULATE which presents statistics side-by side for 6 months, for ease of comparison. The TABULATE commands follow:

```
PROC TABULATE F=8.3;
*(this invokes the procedure and requests a default output of
8-digit fields with 3 decimals.)
;
CLASS MONTH;
*The classification variable will be month;

VAR pap cer sut inj;
*The variables to be analyzed must be specified in the var
statement;

TABLE (pap cer sut inj )*(MEAN MIN MAX STD), ALL MONTH/RTS=12;
*      (row variables      )*(  statistics      ), column vars/ row
title spacing;
```

GRAPHICAL TECHNIQUES

When distributions are not known to be unimodal within group it is best to examine distributions graphically. When many variables are examined, and observations are not plentiful, a parsimonious plot is desirable, such as a stem-and-leaf plot.

The following summary compares the home-run careers of Babe Ruth and Roger Maris (Moore⁷):

	0	8	
	1	346	
52	2	368	
54	3	39	
9766611	4		
944	5		
0	6	1	
Ruth		Maris	

The stem is the ten's digit of no. of homeruns per season. Ruth's graph centers in the 40's, and Maris' indicates that 61 home-runs is much higher than the other season totals.

BOX PLOTS

The SAS^R UNIVARIATE procedure will draw stem-and-leaf plots for groups of 48 or less. Box plots are drawn for larger size groups. Bottom and top edges of the box correspond to the 25th and 75th percentile, respectively. Center line of the box is the median, and the mean is indicated with a plus sign. A vertical line extends above and below the box to indicate the distance to the smallest and largest observation, when these are within 1.5 interquartile ranges ($Q_{75} - Q_{25}$). A zero beyond this line indicates a data point within 3 interquartile ranges of the box. An asterisk is used to indicate an even more extreme point. Using the BY option causes UNIVARIATE to produce side-by-side box plots on the last page of the statistical analysis. Figure 1 shows the distributions of a test result grouped by months.

SIDE-BY-SIDE DOT BOX PLOTS

Side-by-side box plots can be produced by SAS/GRAPH using the ANNOTATE facility, from a dataset which contains selected output of PROC UNIVARIATE run on the input dataset. The box plots tell much about the distributions of the response variable within and between groups (medians, quartiles, ranges), but it is still helpful to see all of the individual values. Inspection of groups of individual values provides insights about possible multi-modal distributions, locations of multiple outliers, and relative group sample sizes.

PROC GPLOT can be used to produce dot plots of the individual values. To get around the problem of multiple dots being hidden one behind another, a small amount of random "jitter", or noise, may be introduced into the horizontal or vertical, or both, components to spread the points by a small amount for better visualization.

The two types of plots may easily be combined by creating a box plot ANNOTATE dataset to be used with the proc GPLOT procedure which produces the dot plot. The program BOXDOTS.SAS produces such overlaid side-by-side box and dot plots, and the main program is generalizable for any user dataset and variables.

The program first reads in (with %include) the files ANNOMAC.SAS and PLOTMAC.SAS. ANNOMAC.SAS is a slightly modified version of the SAS file of the same name which is provided with SAS/GRAPH, which streamlines some of the annotate file creation.

PLOTMAC.SAS provides the macros which the main program calls to create the box plot/dot plot of the user data. The data input section of BOXDOTS.SAS may be replaced by any user statements for producing the input data. The parameters in the call to PLOTDAT may be replaced with any user-specified dataset name, variable names, and format name.

The main program (BOXDOTS.SAS) reads in a dataset which has one or more "response" variables and at least one grouping variable. This program expects the grouping variable to be numeric, and requires a user format to label the groups with group names. The call to macro plotdat requires the specification of the input dataset name, the name of the variable to be plotted, the name of the grouping variable, and the name of the format for the grouping variable.

EXAMPLES OF DOT BOX PLOTS

The examples show Dot Box plots of response variables across four groups. In each graph, the distribution of the plot variable is displayed vertically. Groups are arranged along the horizontal axis, for side-by-side comparison. The value for every subject is plotted as a dot, and the corresponding box-plot for the distribution is overlaid on the

dots. At a glance, differences (or similarities) in central tendencies between the groups are apparent. The larger sample size of the first group is immediately apparent, and locations of extreme values are easily observable. In example 3, similar central tendencies but different ranges are observed. Example 4 shows the same situation but also highlights the lack of all values for group 2. Note that horizontal variation within groups is strictly random "noise", introduced for the purpose of spreading out data points so that they may be more easily seen.

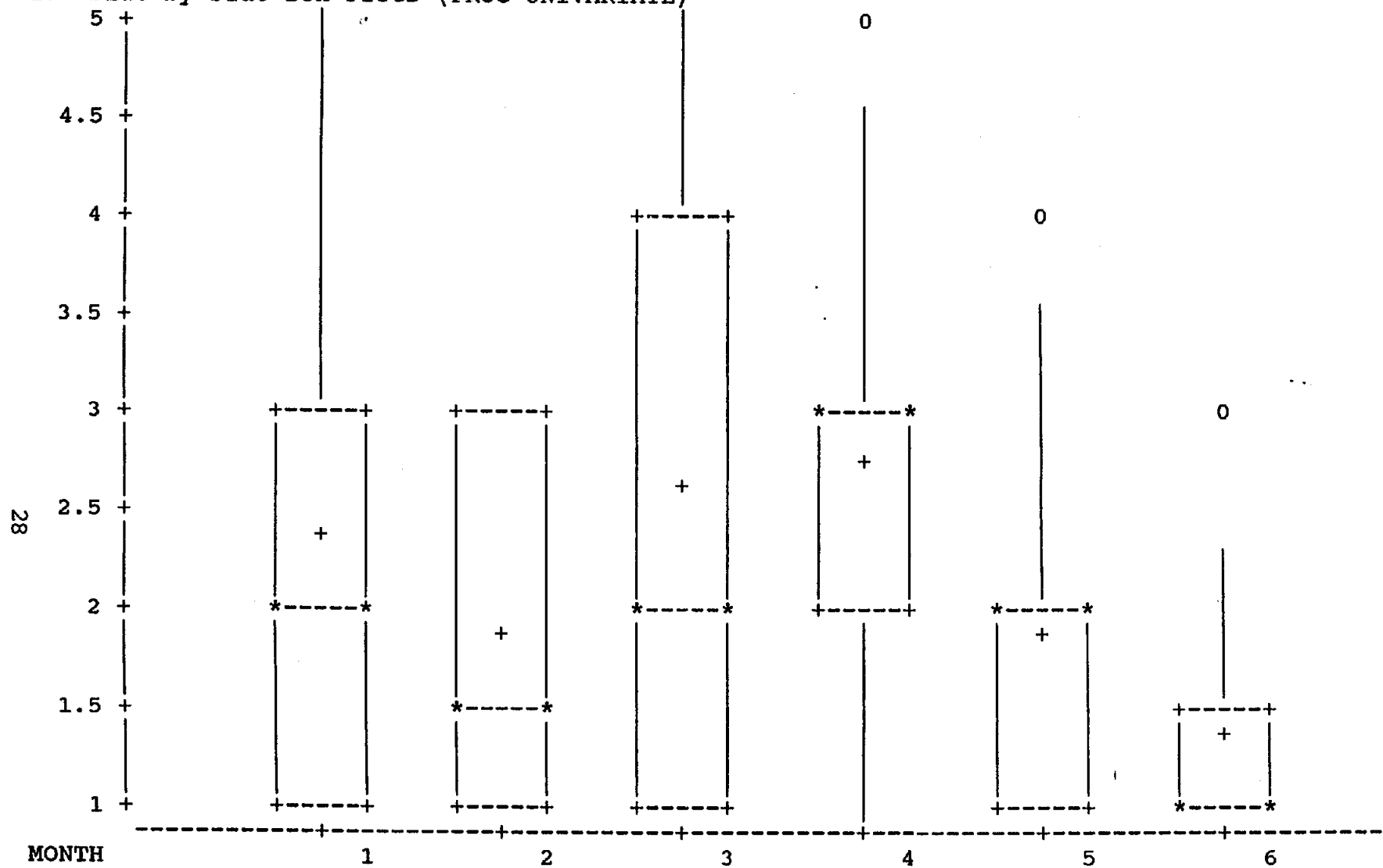
BIBLIOGRAPHY

1. Wilkinson, "Graphical Displays", Statistical Methods in Medical Research 1992 1:3-25
2. Olmsted, Proceedings of the 10th Annual SAS^R Users Group, 1985, 888-892.
3. Benoit, Proceedings of the 10th Annual SAS^R Users Group, 1985, 232-238.
4. Bergstralk, Proceedings of the 15th Annual SAS^R Users Group, 1990, 1094-1099.
5. Friendly, Proceedings of the 15th Annual SAS^R Users Group, 1990, 1425-1428.
6. SAS Institute Inc. SAS^R Procedures Guide, Release 6.03 Edition. Cary, NC: SAS Institute Inc., 1988.
7. Moore, "Uncertainty", On the Shoulders of Giants, pg. 105, L.A. steenl, Ed., National Academic Press, 1990.

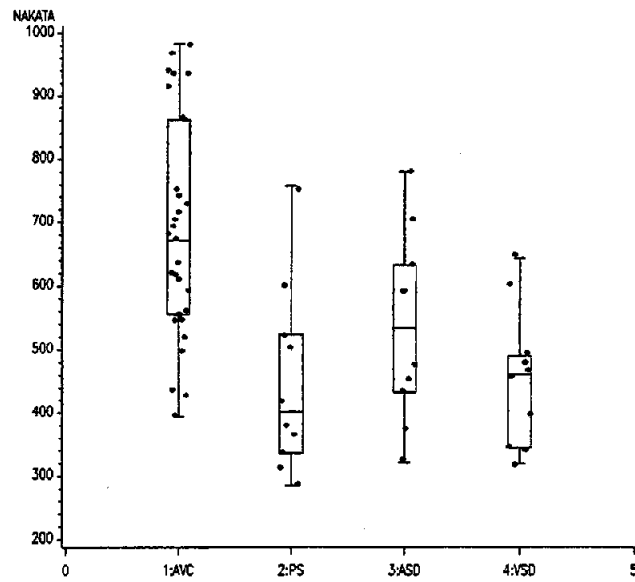
Table 1. Proc Tabulate output comparing variables across 6 months.

		ALL	MONTH					
			1	2	3	4	5	6
PAP1	MEAN	3.500	4.400	4.100	3.286	4.111	4.250	1.417
	N	56.000	10.000	10.000	7.000	9.000	8.000	12.000
	MIN	1.000	1.000	2.000	1.000	3.000	3.000	1.000
	MAX	6.000	6.000	6.000	6.000	5.000	6.000	2.000
	STD	1.584	1.430	1.197	1.799	0.782	1.035	0.515
CER1	MEAN	2.107	2.400	1.900	2.571	2.778	1.875	1.417
	N	56.000	10.000	10.000	7.000	9.000	8.000	12.000
	MIN	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	MAX	5.000	5.000	3.000	5.000	5.000	4.000	3.000
	STD	1.216	1.506	0.994	1.512	1.202	0.991	0.793
SUT1	MEAN	2.536	2.100	2.800	3.429	3.111	3.375	1.167
	N	56.000	10.000	10.000	7.000	9.000	8.000	12.000
	MIN	1.000	1.000	1.000	3.000	1.000	2.000	1.000
	MAX	5.000	5.000	5.000	4.000	4.000	5.000	3.000
	STD	1.293	1.197	1.476	0.535	1.054	0.916	0.577
INJ1	MEAN	3.286	3.400	3.900	4.429	3.667	3.625	1.500
	N	56.000	10.000	10.000	7.000	9.000	8.000	12.000
	MIN	1.000	2.000	1.000	1.000	2.000	2.000	1.000
	MAX	6.000	6.000	6.000	6.000	6.000	5.000	3.000
	STD	1.637	1.578	1.595	1.718	1.225	1.302	0.798

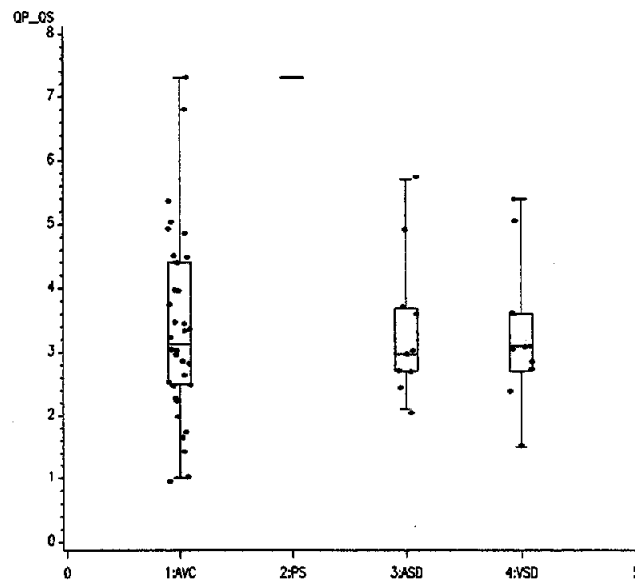
Figure 1. Side-by-Side Box Plots (PROC UNIVARIATE)



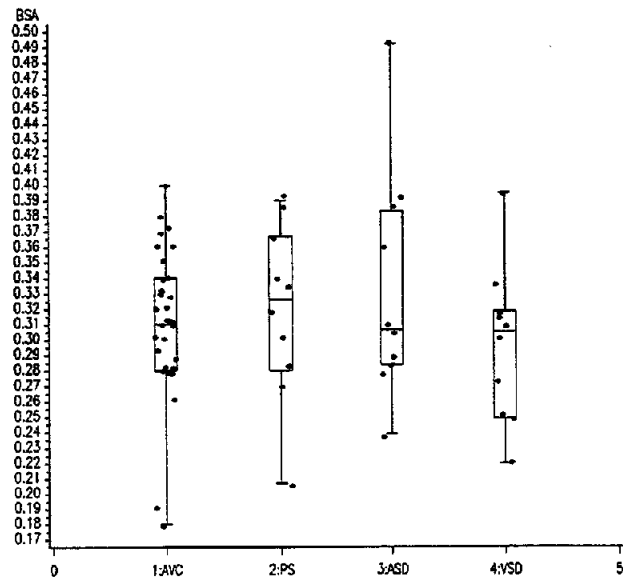
Dots + Box-Plot Example 1



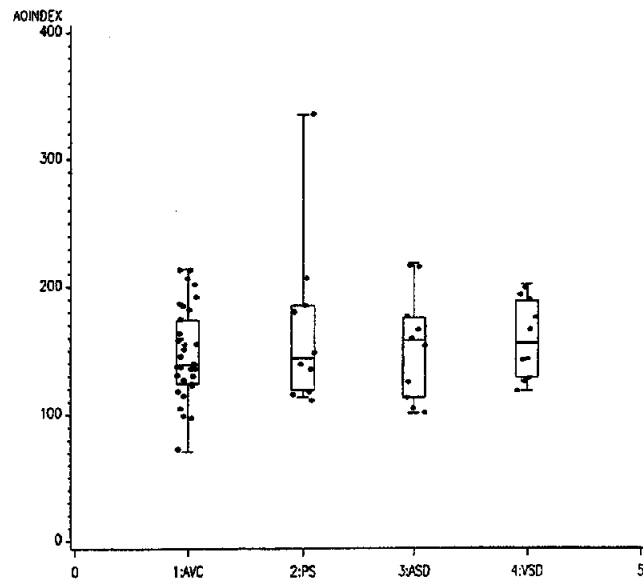
Dots + Box-Plot Example 2



Dots + Box-Plot Example 3



Dots + Box-Plot Example 4



```

/*****
/*      S A S   S A M P L E   L I B R A R Y      */
/*
/*      NAME: ANNOMAC
/*      TITLE: ANNOTATE SUPPORT MACROS
/*      PRODUCT: GRAPH
/*      SYSTEM: OS
/*      KEYS: MACRO GDEVICE
/*      PROCS:
/*      DATA:
/*
/*      SUPPORT: GRAPHICS STAFF          UPDATE: 16NOV84
/*      REF:
/*      MISC: ON CMS--use annocms instead of annomac beacuse of
/*            the length of the macro names.
/*****
/*      NOTE: Modified 9/92, Mary M. Lane
/*            to allow for non-specified parameters to be
/*            set to missing.
/*****
/* option dquote; */

%MACRO sequence( seq );
%*-----;
%* Definition of ANNOTATE generation sequence.
%*-----;
      IF UPCASE("&seq")='AFTER' OR UPCASE("&seq")='A' THEN WHEN = "A";
      ELSE WHEN = "B";
%MEND sequence;

%MACRO system( xs, ys, hs );
%*-----;
%* Definition of ANNOTATE reference systems.
%*-----;
      XSYS = "&xs"; YSYS = "&ys"; HSYS = "&hs";
%MEND system;

%MACRO comment( txt );
%*-----;
%* Places descriptive text string into ANNOTATE observation stream.
%*
%* NOTE: This function will cause a LENGTH to be assigned to the TEXT ;

```

```

%*      variable, due to the nature of the DATA step. Insert a
%*      LENGTH TEXT $ nnn. ;
%*      statement into the DATA step if this is undesirable.
%*-----;
      TEXT=&txt;
      FUNCTION = "COMMENT "; output;
      TEXT=" ";
%MEND comment;

%MACRO dclanno;
%*-----;
%* Set up required variable lengths.
%* Assure X and Y variables are in the dataset.
%* Assure all variables in LENGTH statements are referenced.
%*-----;
      LENGTH FUNCTION COLOR STYLE $ 8;
      LENGTH XSYS YSYS HSYS $ 1;
      LENGTH WHEN POSITION $ 1;
      RETAIN XSYS YSYS HSYS;

      X = . ;
      Y = . ;
      STYLE = " ";
      POSITION = "5";
      COLOR = " ";
      FUNCTION = " ";

      %system( 4, 4, 4 );
      %sequence( BEFORE );

%MEND dclanno;

%MACRO move( x1, y1 );
%*-----;
%* MOVE to the requested ( X1,Y1 ) coordinate.
%*-----;
      X = &x1;
      Y = &y1;
      FUNCTION = "MOVE "; output;
%MEND move;

```

```

%MACRO draw( x1, y1, colin, lintyp, width );
%*-----;
%* DRAW a line to the requested ( X1,Y1 ) coordinate.
%*-----;
  X = &x1;
  Y = &y1;
  LINE   = &lintyp;
  SIZE   = &width;
  IF "&colin" =: '*' THEN ; ELSE color = "&colin" ;
  FUNCTION = "DRAW " ; output;
%MEND draw;

```

```

%MACRO label( x1, y1, txt, coltxt, ang, rot, hgt, font, pos );
%*-----;
%* Place the TXT string at ( X1,Y1 ).
%*
%* NOTE: Literal text MUST be enclosed in quotes, otherwise the TEXT
%*       variable will be assigned the value of the variable named in
%*       the symbolic macro parameter. DATA step errors will occur if
%*       one of these two conditions is not met.
%*-----;
  X = &x1;
  Y = &y1;
  if "&ang" =: '' then angle=.; else ANGLE   = put("&ang",$char.);
  if "&rot" =: '' then rotate=.; else ROTATE  = put("&rot",$char.);
  if "&hgt" =: '' then size=.; else SIZE     = put("&hgt",$char.);
  STYLE   = "&font";
  TEXT    = &txt;
  IF "&pos" =: '*' THEN ; ELSE POSITION = "&pos" ;
  IF "&coltxt" =: '*' THEN ; ELSE color = "&coltxt";
  FUNCTION = "LABEL " ; output;
%MEND label;

```

```

%MACRO rect( x1, y1, x2, y2, colin, lintyp, width );
%*-----;
%* Draw a rectangle. Non-fillable definition.
%*
%* NOTE: ( X1,Y1 ) and ( X2,Y2 ) are opposing corners.
%*-----;
  %move ( &x1, &y1 );
  %draw ( &x2, &y1, &colin, &lintyp, &width );
  %draw ( &x2, &y2, &colin, &lintyp, &width );
  %draw ( &x1, &y2, &colin, &lintyp, &width );
  %draw ( &x1, &y1, &colin, &lintyp, &width );

```

```
%MEND rect;
```

```

%MACRO bar( x1, y1, x2, y2, color, bartyp, pattern );
%*-----;
%* Draw a rectangle. Fillable definition.
%*
%* NOTE: ( X1,Y1 ) and ( X2,Y2 ) are opposing corners.
%*-----;
  %move ( &x1, &y1 );
  X = &x2;
  Y = &y2;
  LINE   = &bartyp;
  STYLE  = "&pattern";
  IF "&color" =: '*' THEN ; ELSE color = "&color" ;
  FUNCTION = "BAR " ; output;
%MEND bar;

```

```

%MACRO circle( x1, y1, rad, colin );
%*-----;
%* Draw a circle with center at ( X1,Y1 ) of radius RAD.
%*-----;
  X = &x1;
  Y = &y1;
  LINE   = 0;
  ANGLE  = 0.00;
  ROTATE = 360.00;
  SIZE   = &rad;
  STYLE  = "EMPTY";
  IF "&colin" =: '*' THEN ; ELSE color = "&colin";
  FUNCTION = "PIE"; output;
%MEND circle;

```

```

%MACRO slice( x1, y1, ang, rot, rad, color, pattern, lttyp );
%*-----;
%* Define a pie slice.
%* Center at ( X1,Y1 ), radius of RAD, starting angle of ANG, with a
%* traverse angle of ROT. LTYP defines type of boundary to draw.
%*
%* NOTE: LTYP must be defined by literal strings as listed below
%*
%*       Setting LTYP to WHOLE is a special case in the macro,
%*       and causes the overriding of specified ANG and ROT values.

```

```

%*
%*      See documentation for special ANG/ROT value handling.
%*
%*-----
%LET  lntp= %UPCASE("&ltyp");
      %IF %INDEX(&lntp,WHOLE) > 0 %THEN %STR( LINE = 0; ) ;
%ELSE %IF %INDEX(&lntp,LEAD) > 0 %THEN %STR( LINE = 1; ) ;
%ELSE %IF %INDEX(&lntp,TRAIL) > 0 %THEN %STR( LINE = 2; ) ;
%ELSE %IF %INDEX(&lntp,BOTH) > 0 %THEN %STR( LINE = 3; ) ;
%ELSE %IF %INDEX(&lntp,NONE) > 0 %THEN %STR( LINE = 0; ) ;
%ELSE                                %STR( LINE = 3; ) ;

X = &x1;
Y = &y1;

%IF %INDEX(&lntp,WHOLE) > 0
%THEN %DO;
      %STR( ANGLE = 0; );
      %STR( ROTATE = 360; );
%END;
%ELSE %DO;
      %STR( ANGLE = &ang; );
      %STR( ROTATE = &rot; );
%END;

SIZE = &rad;
STYLE = "&pattern";
      IF "&color" =: '*' THEN ; ELSE color = "&color";
FUNCTION = "PIE " ; output;
%MEND slice;

%MACRO piexy( ang, mul );
%*-----
%* Return coordinates along the radius of a previously defined pie.
%*-----
X = .;
Y = .;
SIZE = &mul;
ANGLE = &ang;
FUNCTION = "PIEXY " ; output;
%MEND piexy;

%MACRO poly ( x1, y1, color, pattern, lntyp );
%*-----

```

```

%* Begin definition of a polygon.
%*-----
X = &x1;
Y = &y1;
LINE = &lntyp;
STYLE = "&pattern";
      IF "&color" =: '*' THEN ; ELSE color = "&color" ;
FUNCTION = "POLY " ; output;
%MEND poly;

%MACRO polycont ( x1, y1, colin );
%*-----
%* Continue definition of a polygon.
%*-----
X = &x1;
Y = &y1;
      IF "&colin" =: '*' THEN ; ELSE color = "&colin";
FUNCTION = "POLYCONT"; output;
%MEND polycont;

%MACRO frame ( colin, lntyp, width, pattern );
%*-----
%* Continue definition of a polygon.
%*-----
X = . ;
Y = . ;
      IF "&colin" =: '*' THEN ; ELSE color = "&colin";
STYLE = "&pattern";
LINE= &lntyp;
SIZE= &width;
FUNCTION = "FRAME " ; output;
%MEND frame;

%MACRO scale( ptx,pty, xmin,ymin, xmax,ymax, vxmin,vymin, vxmax,vymax
);
%*-----
%*
%* scales input coordinates ( ptx,pty ) into a requested range
%*
%* vxmin,vymin are the output range minimums ( target )
%* vxmax,vymax are the output range maximums ( target )
%*

```

```

%*      xmin,ymin      are the input range minimums ( source )
%*      xmax,ymax      are the input range maximums ( source )
%*
%* NOTE: ABS(minimum) MAY NOT EQUAL ABS(maximum).
%*
%*      NO OBSERVATION IS OUTPUT BY THIS MACRO.
%*
%*-----
%LET f1 = ( (&vxmax-&vxmin) / (&xmax-&xmin) );
%LET f2 = ( (&vymax-&vymin) / (&ymax-&ymin) );
      X = &f1 * (&ptx-&xmin);
      Y = &f2 * (&pty-&ymin);
%MEND scale;

%MACRO push;
%*-----
%* PUSH values (XLAST,YLAST),(XLSTT,YLSTT) onto a LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      FUNCTION="PUSH      "; output;
%MEND push;

%MACRO pop;
%*-----
%* POP values (XLAST,YLAST),(XLSTT,YLSTT) from the LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      FUNCTION="POP      "; output;
%MEND pop;

%MACRO swap;
%*-----
%* SWAP values (XLAST,YLAST) and (XLSTT,YLSTT).
%* Does NOT affect values on LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      FUNCTION="SWAP      "; output;
%MEND swap;

```

```

%MACRO txt2cntl;
%*-----
%* COPY values (XLSTT,YLSTT) into (XLAST,YLAST).
%* Does NOT affect values on LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      FUNCTION = "TXT2CNTL"; output;
%MEND txt2cntl;

%MACRO cntl2txt;
%*-----
%* COPY values (XLAST,YLAST) into (XLSTT,YLSTT).
%* Does NOT affect values on LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      FUNCTION = "CNTL2TXT"; output;
%MEND cntl2txt;

%MACRO draw2txt ( colin, lintyp, width );
%*-----
%* DRAW a line from (XLAST,YLAST) to (XLSTT,YLSTT).
%* Does NOT affect values on LIFO system stack.
%*-----
      X = . ;
      Y = . ;
      SIZE      = &width;
      LINE      = &lintyp;
      IF "&colin" =: '*' THEN ; ELSE color = "&colin" ;
      FUNCTION = "DRAW2TXT"; output;
%MEND draw2txt;

%MACRO line( x1, y1, x2, y2, colin, lintyp, width );
%*-----
%* DRAW a line from (X1,Y1) to (X2,Y2).
%* Simplified version supplying the invisible move instruction.
%*-----
      %move( &x1, &y1 );
      %draw( &x2, &y2, &colin, &lintyp, &width );
%MEND line;

```

```

*****;
%macro univ1(indata, group, vname, outdata);
proc sort data=&indata; by &group; run;
proc univariate data=&indata noprint; by
&group;
var &vname;
output out=&outdata
nobs=nobs n=n min=min max=max
median=median q1=q1 q3=q3;
run;
data &outdata; set &outdata;
length vname$ 8 grpvar$ 8;
vname=upcase("&vname");
grpvar=upcase("&group");
grpval=&group;
run;
%mend;
*****;
%macro ginit;
goptions reset=goptions reset=symbol
device=vga16 rotate=landscape
ftext=simplex;
proc greplay igout=gseg nofs; delete _all_;
run;
%mend;
*****;
%macro boxdat(indata, plotvar, grpvar);
%univ1(indata=&indata, group=&grpvar,
vname=&plotvar, outdata=udat1);

data boxes; set udat1;
%dcianno;
drop barw;
%system(xs=2,ys=2);
barw=.2;* barwidth;
x0=grpval;

%bar(x1=x0-barw/2, y1=q1, x2=x0+barw/2,
y2=q3,
color=white, bartyp=0, pattern=none);

%move(x1=x0-barw/2, y1=median);
%draw(x1=x0+barw/2, y1=median, colin=*,
lintyp=1, width=1);

%move(x1=x0, y1=q1);
%draw(x1=x0, y1=min, colin=*, lintyp=1,
width=1);

```

```

%move(x1=x0-barw/4, y1=min);
%draw(x1=x0+barw/4, y1=min, colin=*,
lintyp=1, width=1);

%move(x1=x0, y1=q3);
%draw(x1=x0, y1=max, colin=*, lintyp=1,
width=1);
%move(x1=x0-barw/4, y1=max);
%draw(x1=x0+barw/4, y1=max, colin=*,
lintyp=1, width=1);
run;
%mend;
*****;
%macro lbdatt(indata, grpvar);
proc freq data=&indata(keep=&grpvar); tables
&grpvar / noprint out=g1; run;
data lbdatt; set g1;
%dcianno;
%system(xs=2,ys=3);
xval=input(substr(&grpvar,1,1),1.);
%label(x1=xval,y1=5,txt=&grpvar,pos=5);
run;
%mend;
*****;
%macro plotdat(indata, plotvar, grpvar, gfmt);
%ginit;

%boxdat(indata=&indata, plotvar=&plotvar, grpvar=&grpvar);
%*lbdatt(indata=&indata, grpvar=&grpvar);
data anotes; set boxes; run;

%plot2a(indata=&indata, plotvar=&plotvar, grpvar=&grpvar, gfmt=&gfmt);

%mend;
*****;
%macro plot2a(indata, plotvar, grpvar, gfmt);
proc univariate data=&indata(keep=&plotvar)
noprint;
var &plotvar;
output out=pv1 range=pvrange;
run;
data _null_; set pv1;
maxfuzz=.01*pvrange;
call symput('maxfuzz',maxfuzz);
run;

```

```

axis1 origin=(10 pct) order=(0 to 5 by 1)
label=none minor=none;
axis2 label=("&plotvar");
%let symsize=.5;
%let symv=dot;
symbol1 v=dot i=none c=red h=&symsize;
symbol2 v=dot i=none c=white h=&symsize;
symbol3 v=dot i=none c=blue h=&symsize;
symbol4 v=dot i=none c=orange h=&symsize;
*add more symbol statements here if more
than four groups;
data t1; set patient;
xw=.1;
xfuzz=ranuni(0)*2*xw-xw;
xdum=&grpvar+xfuzz;
yfuzz=ranuni(0)*2*&maxfuzz-&maxfuzz;
pv=&plotvar+yfuzz;
run;
proc gplot anno=anotes;
plot pv*xdum=&grpvar / vaxis=axis2
haxis=axis1 nolegend;
format xdum &gfmt..;
run;
%mend;
*****;

```

```
%include 'annomac.sas';* load in sas annotate macros;
%include 'plotmac.sas';* load in macros to do plots;
*****Main program begins here*****;

proc datasets lib=work memtype=data kill; run;* clear previous datasets;

libname data '.';* assumes permanent dataset is in current directory;
data patient; set data.examp1;* read permanent data into a temporary
                        dataset;

run;
proc print; run;

* set group format;
proc format;
  value grpfmt 1='1:AVC' 2='2:PS' 3='3:ASD' 4='4:VSD';
run;

options mprint;* take this out if printout of macro steps not needed;

* call the plot routine, giving input dataset name, the response variable
to plot, the name of the grouping variable, and the name of the format
for the grouping variable;

title 'Dots + Box-Plot Example 1';
%plotdat(indata=patient,plotvar=NAKATA,grpvar=groupnb,gfmt=grpfmt);

title 'Dots + Box-Plot Example 2';
%plotdat(indata=patient,plotvar=QP_QS,grpvar=groupnb,gfmt=grpfmt);

title 'Dots + Box-Plot Example 3';
%plotdat(indata=patient,plotvar=BSA,grpvar=groupnb,gfmt=grpfmt);

title 'Dots + Box-Plot Example 4';
%plotdat(indata=patient,plotvar=AOINDEX,grpvar=groupnb,gfmt=grpfmt);

options nomprint;* turns off macro step logging;

*****;
*****;
```