
Subject: Re: plotting routine to handle missing data
Posted by [thompson](#) on Thu, 15 Apr 1993 13:42:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

sharma@climate.gsfc.nasa.gov (Om Sharma) writes:

> Does anyone have a plotting routine that can handle missing data ?

> Om Sharma

> sharma@climate.gsfc.nasa.gov

This has been asked before, so this time I'll post the routines instead of simply mailing them. There are two routines, PLOT_DROP and OPLOT_DROP.

Bill Thompson

P.S. A better newsgroup would be comp.lang.idl-pvwave.

```
=====
=====
PRO PLOT_DROP,X_ARRAY,Y_ARRAY,DROP_VALUE,TITLE=TITLE,XTITLE=XTITLE,$
  YTITLE=YTITLE,XTYPE=XTYPE,YTYPE=YTYPE,PSYM=PSYM,COLOR=COLOR,$
  LINESTYLE=LINESTYLE,XRANGE=XRANGE,YRANGE=YRANGE
;+
; NAME:
; PLOT_DROP
; PURPOSE:
; Plots arrays where some points have data dropout. Only those segments
; with data are plotted. If !PSYM is not positive, then lines will
; connect the data points only within complete segments and will not
; cross over segments with data dropout.
; CATEGORY:
; CALLING SEQUENCE:
; PLOT_DROP, [ X_ARRAY, ] Y_ARRAY [, DROP_VALUE ]
; INPUTS:
; Y_ARRAY - Array of y values, including data dropouts.
; OPTIONAL INPUT PARAMETERS:
; X_ARRAY - Array of x values, defaults to indices of Y_ARRAY.
; DROP_VALUE - Values of Y_ARRAY equal to DROP_VALUE signal the
; data dropout. Defaults to 32000.
; OPTIONAL KEYWORD PARAMETERS:
; TITLE = Main plot title, default is !P.TITLE.
; XTITLE = X axis title, default is !X.TITLE.
; YTITLE = Y axis title, default is !Y.TITLE.
; XTYPE = If set, then X axis is logarithmic.
; YTYPE = If set, then Y axis is logarithmic.
; PSYM = Plotting symbol.
```

```

; COLOR = Plotting color. Used only for the data itself. The axes
; and titles are plotted using !COLOR.
; LINSTYLE = Line style to be used for plotting.
; X RANGE = Range of data values in the X direction.
; Y RANGE = Range of data values in the Y direction.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; PLOT is called using the /NODATA keyword to draw the axes and titles,
; and then OPLOT_DROP is called to overplot the data.
;
; The normal system variables, such as !XMIN, etc., can be used to
; control the plot, or one can allow the plot to autoscale in the usual
; manner. In the latter case all of X_ARRAY is used to scale the plot
; in the X direction, even the parts which have data dropout.
;
; MODIFICATION HISTORY:
; William Thompson Applied Research Corporation
; September, 1987 8201 Corporate Drive
; Landover, MD 20785
;
; William Thompson, December 1991, added keywords XTYPE, YTYPE, PSYM,
; COLOR, and LINSTYLE.
; William Thompson, May 1992, added keywords X RANGE and Y RANGE.
;-
;
ON_ERROR,2
;
; Parse the input parameters.
;
IF N_PARAMS(0) EQ 1 THEN BEGIN ;Only Y_ARRAY passed.
  Y = X_ARRAY
  X = FINDGEN(N_ELEMENTS(Y))
  DROP = 32000
END ELSE IF N_PARAMS(0) EQ 2 THEN BEGIN
  N = N_ELEMENTS(Y_ARRAY)
  IF N EQ 1 THEN BEGIN ;Y_ARRAY and DROP_VALUE passed.
    Y = X_ARRAY
    X = FINDGEN(N_ELEMENTS(Y))
    DROP = Y_ARRAY
  END ELSE BEGIN ;X_ARRAY and Y_ARRAY passed.
    X = X_ARRAY
    Y = Y_ARRAY
    DROP = 32000
  
```

```

ENDELSE
END ELSE IF N_PARAMS(0) EQ 3 THEN BEGIN ;All variables passed.
X = X_ARRAY
Y = Y_ARRAY
DROP = DROP_VALUE
END ELSE BEGIN
PRINT,' This procedure must be called with 1-3 parameters:'
PRINT,' [ X_ARRAY ,] Y_ARRAY [, DROP_VALUE ]'
RETURN
ENDELSE
;
;
; Save the variables !C and !XMIN, !XMAX, !YMIN, !YMAX to be restored at the
; end of the procedure. If not already defined, set the scaling in X.
;
BANG_C = !C
XMIN = !XMIN & XMAX = !XMAX & X1 = XMIN & X2 = XMAX
YMIN = !YMIN & YMAX = !YMAX & Y1 = YMIN & Y2 = YMAX
IF X1 EQ X2 THEN BEGIN
X1 = MIN(X)
X2 = MAX(X)
ENDIF
;
; Find all the points not equal to DROP_VALUE.
;
W = WHERE(Y NE DROP)
IF W(0) EQ -1 THEN BEGIN
PRINT,' All points equal to ',DROP
GOTO,DONE
ENDIF
;
; If not already defined, set the scaling in Y. Ignore any data dropouts.
;
IF Y1 EQ Y2 THEN BEGIN
Y1 = MIN(Y(W))
Y2 = MAX(Y(W))
ENDIF
SET_XY,X1,X2,Y1,Y2
;
; First create the axes without plotting any data points.
;
COMMAND = "PLOT,X,Y,/NODATA"
IF N_ELEMENTS(XTITLE) EQ 1 THEN COMMAND = COMMAND + ",XTITLE=XTITLE"
IF N_ELEMENTS(YTITLE) EQ 1 THEN COMMAND = COMMAND + ",YTITLE=YTITLE"
IF N_ELEMENTS(TITLE) EQ 1 THEN COMMAND = COMMAND + ",TITLE=TITLE"
IF N_ELEMENTS(XTYPE) EQ 1 THEN COMMAND = COMMAND + ",XTYPE=XTYPE"
IF N_ELEMENTS(YTYPE) EQ 1 THEN COMMAND = COMMAND + ",YTYPE=YTYPE"
IF N_ELEMENTS(XRANGE) EQ 2 THEN COMMAND = COMMAND + ",XRANGE=XRANGE"
IF N_ELEMENTS(YRANGE) EQ 2 THEN COMMAND = COMMAND + ",YRANGE=YRANGE"

```

```

TEST = EXECUTE(COMMAND)
;
; Call OPLOT_DROP to plot the data.
;
OPLOT_DROP,X,Y,DROP,PSYM=PSYM,COLOR=COLOR,LINESTYLE=LINESTYL E
;
; Exit point; restore system variables.
;
DONE:
!C = BANG_C
!XMIN = XMIN & !XMAX = XMAX & !YMIN = YMIN & !YMAX = YMAX
RETURN
END
=====
=====
PRO OPLOT_DROP,X_ARRAY,Y_ARRAY,DROP_VALUE,PSYM=PSYM,COLOR=COLOR, $
LINESTYLE=LINESTYLE
;+
; NAME:
; OPLOT_DROP
; PURPOSE:
; Overplots arrays where some points have data dropout. Only those
; segments with data are plotted. If !PSYM is not positive, then lines
; will connect the data points only within complete segments and will
; not cross over segments with data dropout.
; CATEGORY:
; CALLING SEQUENCE:
; OPLOT_DROP, [ X_ARRAY, ] Y_ARRAY [, DROP_VALUE ]
; INPUTS:
; Y_ARRAY - Array of y values, including data dropouts.
; OPTIONAL INPUT PARAMETERS:
; X_ARRAY - Array of x values, defaults to indices of Y_ARRAY.
; DROP_VALUE - Values of Y_ARRAY equal to DROP_VALUE signal the
; data dropout. Defaults to 32000.
; OPTIONAL KEYWORD PARAMETERS:
; PSYM = Plotting symbol.
; COLOR = Plotting color.
; LINESTYLE = Line style to be used for plotting.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; A series of overplots is done for each segment.
; MODIFICATION HISTORY:
; William Thompson Applied Research Corporation

```

```

; September, 1987 8201 Corporate Drive
; Landover, MD 20785
;
; William Thompson, December 1991, added keywords XTYPE, YTYPE, PSYM,
; COLOR, and LINESYLE.
;-
;
; Parse the input parameters.
;
IF N_PARAMS(0) EQ 1 THEN BEGIN ;Only Y_ARRAY passed.
  Y = X_ARRAY
  X = FINDGEN(N_ELEMENTS(Y))
  DROP_VALUE = 32000
END ELSE IF N_PARAMS(0) EQ 2 THEN BEGIN
  N = N_ELEMENTS(Y_ARRAY)
  IF N EQ 1 THEN BEGIN ;Y_ARRAY and DROP_VALUE passed.
    Y = X_ARRAY
    X = FINDGEN(N_ELEMENTS(Y))
    DROP_VALUE = Y_ARRAY
  END ELSE BEGIN ;X_ARRAY and Y_ARRAY passed.
    X = X_ARRAY
    Y = Y_ARRAY
    DROP_VALUE = 32000
  ENDELSE
END ELSE IF N_PARAMS(0) EQ 3 THEN BEGIN ;All variables passed.
  X = X_ARRAY
  Y = Y_ARRAY
END ELSE BEGIN
  PRINT,' This procedure must be called with 1-3 parameters:'
  PRINT,' [ X_ARRAY ,] Y_ARRAY [, DROP_VALUE ]'
  RETURN
ENDELSE
;
; Save !C to be restored at end of routine.
;
BANG_C = !C
;
; Find all the points not equal to DROP_VALUE.
;
W = WHERE(Y NE DROP_VALUE)
IF W(0) EQ -1 THEN BEGIN
  PRINT,' All points equal to ',DROP_VALUE
  GOTO,DONE
ENDIF
;
; Set the values of PSYM, COLOR, and LINESYLE.
;
IF N_ELEMENTS(PSYM) EQ 0 THEN PSYM = !PSYM

```

```

IF N_ELEMENTS(COLOR) EQ 0 THEN COLOR = !COLOR
IF N_ELEMENTS(LINESTYLE) EQ 0 THEN LINESTYLE = !LINESTYLE
;
; Define the variables XX and YY to be the remaining data points. I is the
; position of the last point with a dropout.
;
XX = X
YY = Y
I = -1
;
; Do the next segment.
;
NEXT:
I = I + 1
IF I EQ N_ELEMENTS(XX) THEN GOTO,DONE
XX = XX(I:*)
YY = YY(I:*)
;
; Find the first point without a dropout.
;
I = MIN(WHERE(YY NE DROP_VALUE))
IF I EQ -1 THEN GOTO,DONE
XX = XX(I:*)
YY = YY(I:*)
;
; Find the next data dropout, and plot up to that point. If only one point is
; to be plotted, double it to fool the OPLLOT procedure.
;
I = MIN(WHERE(YY EQ DROP_VALUE)) - 1
!C = 0
IF I EQ -2 THEN BEGIN
IF N_ELEMENTS(XX) GT 1 THEN BEGIN
OPLLOT,XX,YY,PSYM=PSYM,COLOR=COLOR,LINestyle=LINESTYLE
END ELSE BEGIN
OPLLOT,[XX,XX],[YY,YY],PSYM=PSYM,COLOR=COLOR, $
LINestyle=LINESTYLE
ENDELSE
GOTO,DONE
END ELSE IF I GT 0 THEN BEGIN
OPLLOT,XX(0:I),YY(0:I),PSYM=PSYM,COLOR=COLOR,LINestyle=LINESTYLE
END ELSE BEGIN
OPLLOT,[XX(0),XX(0)],[YY(0),YY(0)],PSYM=PSYM,COLOR=COLOR, $
LINestyle=LINESTYLE
ENDELSE
GOTO,NEXT
;
; Exit point; restore !C.
;

```

```
DONE:  
!C = BANG_C  
RETURN  
END
```
