Subject: Re: contiguous stacked plots
Posted by sjt on Mon, 09 Jan 1995 09:57:57 GMT
View Forum Message <> Reply to Message

Vincent E. Kargatis (vek@spacsun.rice.edu) wrote:
: I want to make a plot of contiguous stacked plots, like so:
:  |----------|
:  |  |
: Y1 |  |
:  |  |
:  |--|--|--|--|--|--|--|--|--|--|---|
:  |  |
: Y2 |  |
:  |  |
:  |--|--|--|--|--|--|--|--|--|--|---|
:  |  |
: Y3 |  |
:  |  |
:  |--|--|--|--|--|--|--|--|--|--|---|
:  |  |
: Y4 |  |
:  |  |
:  |--|--|--|--|--|--|--|--|--|--|---|
:  0  1  2  3  4  (etc.)

:  X label

: Any hints? I know it probably is a rather convoluted series of AXIS
: statements, but if I can save some time...

: Thanks,
: --
: Vincent Kargatis -- Space Physics & |
:  Astronomy, Rice U., Houston, TX | A penny saved is ridiculous.
:  [vek@spacsun.rice.edu] |
: [http://spacsun.rice.edu/~vek/vek.html] |

Your best bet for contiguous plots is to use either the position keyword
or the !P.POSITION system variable. (For stacked plots with gaps the
!P.REGION is more suitable). You'll also need to remember for all except
the bottom plot to set the tick names to spaces.

e.g. for 4 plots:

plot, x, y1, position=[.1,.1,.9,.3] ; bottom plot has labels.
plot, x, y2, position=[.1,.3,.9,.5], xtickn=replicate(' ',30), /noerase
plot, x, y3, position=[.1,.5,.9,.7], xtickn=replicate(' ',30), /noerase

plot, x, y4, position=[.1,.7,.9,.9], xtickn=replicate(' ',30), /noerase

--
+----------------------+-------------------------------- --+---------+
| James Tappin,          | School of Physics & Space Research |  O__    |
| sjt@star.sr.bham.ac.uk | University of Birmingham       | --  \/` |
| Ph: 021-414-6462. Fax: 021-414-3722         |  |
+-------------------------------------------------------- --+---------+

---

Subject: Re: contiguous stacked plots
Posted by knight on Mon, 09 Jan 1995 16:29:16 GMT
View Forum Message <> Reply to Message

In article <3el9e7$iiq@larry.rice.edu>, vek@spacsun.rice.edu (Vincent E. Kargatis) writes:
|> I want to make a plot of contiguous stacked plots, like so:
|>     |------------------------------|
|>     |                              |
|> Y1  |                              |
|>     |                              |
|>     |--|--|--|--|--|--|--|--|--|---|
|>     |                              |
|> Y2  |                              |
|>     |                              |
|>     |--|--|--|--|--|--|--|--|--|---|
|>     |                              |
|> Y3  |                              |
|>     |                              |
|>     |--|--|--|--|--|--|--|--|--|---|
|>     |                              |
|> Y4  |                              |
|>     |                              |
|>     |--|--|--|--|--|--|--|--|--|---|
|>     0     1     2     3     4   (etc.)
|>
|>               X label
|>
|>
|> Any hints?  I know it probably is a rather convoluted series of AXIS
|> statements, but if I can save some time...
|>
|> Thanks,
|> --
|> Vincent Kargatis  --  Space Physics &   |
|>    Astronomy, Rice U., Houston, TX      | A penny saved is ridiculous.
|>        [vek@spacsun.rice.edu]           |
|> [http://spacsun.rice.edu/~vek/vek.html] |

Here's one I use.  See the examples in the header for usage.
Note that multiplot,/help will print the header.

```
;+
; Name:
; MULTIPLOT
; Purpose:
; This procedure makes a matrix of plots with *SHARED AXES*, either using
; parameters passed to multiplot or !p.multi in a non-standard way.
; It is good for data with one or two shared axes and retains all the
; versatility of the plot commands (e.g. all keywords and log scaling).
; The plots are connected with the shared axes, which saves space by
; omitting redundant ticklabels and titles.  Multiplot does this by
; setting !p.position, !x.tickname and !y.tickname automatically.
; A call (multiplot,/reset) restores original values.
;
; Note: This method may be superseded by future improvements in !p.multi
; by RSI.  For now, it's a good way to gang plots together.
; Examples:
; multiplot,/help   ; print this header.
; ; Then copy & paste, from your xterm, the following lines to test:
;
; x = findgen(100)  ;    MULTIPLOT
; t=exp(-(x-50)^2/300)  ;  -------------------------
; erase    ;  |          |          |
; u=exp(-x/30)   ;  |          |          |
; y = sin(x)   ;  |  UL plot  |  UR plot  |
; r = reverse(y*u)  ;  |          |          |
; !p.multi=[0,2,2,0,0]  ;  |          |          |
; multiplot     ; y-------------------------
; plot,x,y*u,title='MULTIPLOT' ; l|          |          |
; multiplot & plot,x,r   ; a|          |          |
; multiplot    ; b|  LL plot  |  LR plot  |
; plot,x,y*t,ytit='ylabels' ; e|          |          |
; multiplot    ; l|          |          |
; plot,x,y*t,xtit='xlabels' ; s-------------------------
; multiplot,/reset  ;          xlabels
;
; wait,2 & erase   ;   TEST
; multiplot,[1,3]   ; H-------------------------
; plot,x,y*u,title='TEST'  ; E| plot #1  |
; multiplot   ; I-------------------------
; plot,x,y*t,ytit='HEIGHT' ; G| plot #2  |
; multiplot   ; H-------------------------
; plot,x,r,xtit='PHASE'  ; T| plot #3  |
; multiplot,/reset  ;  -------------------------
;     ;   PHASE
```

```
;
; multiplot,[1,1],/init,/verbose ; one way to return to single plot
; % MULTIPLOT: Initialized for 1x1, plotted across then down (column major).
; Usage:
;  multiplot[pmulti][,/help][,/initialize][,/reset][,/rowmajor]
; Optional Inputs:
; pmulti = 2-element or 5-element vector giving number of plots, e.g.,
;   multiplot,[1,6]  ; 6 plots vertically
;   multiplot,[0,4,2,0,0]  ; 4 plots along x and 2 along y
;   multiplot,[0,4,2,0,1]  ; ditto, except rowmajor (down 1st)
;   multiplot,[4,2],/rowmajor  ; identical to previous line
; Optional Keywords:
; help = flag to print header
; initialize = flag to begin only---no plotting, just setup,
;   e.g., multiplot,[4,2],/init,/verbose & multiplot & plot,x,y
; reset = flag to reset system variables to values prior to /init
; default = flag to restore IDL's default value for system variables
; rowmajor = flag to number plots down column first (D=columnmajor)
; verbose = flag to output informational messages
; Outputs:
; !p.position = 4-element vector to place a plot
; !x.tickname = either '' or else 30 ' ' to suppress ticknames
; !y.tickname = either '' or else 30 ' ' to suppress ticknames
; !p.noerase = 1
; Common blocks:
; multiplot---to hold saved variables and plot counter.  See code.
; Side Effects:
; Multiplot sets a number of system variables: !p.position, !p.multi,
; !x.tickname, !y.tickname, !P.noerase---but all can be reset with
; the call: multiplot,/reset
; Restrictions:
; 1. If you use !p.multi as the method of telling how many plots
; are present, you have to set !p.multi at the beginning each time you
; use multiplot or call multiplot with the /reset keyword.
; 2. There's no way to make an xtitle or ytitle span more than one plot,
; except by adding spaces to shift it or to add it manually with xyouts.
; 3. There is no way to make plots of different sizes; each plot
; covers the same area on the screen or paper.
; Procedure:
; This routine makes a matrix of plots with common axes, as opposed to
; the method of !p.multi where axes are separated to allow labels.
; Here the plots are joined and labels are suppressed, except at the
; left edge and the bottom.  You tell multiplot how many plots to make
; using either !p.multi (which is then reset) or the parameter pmulti.
; However, multiplot keeps track of the position by itself because
; !p.multi interacts poorly with !p.position.
; Modification history:
; write, 21-23 Mar 94, Fred Knight (knight@ll.mit.edu)
```

```
; alter plot command that sets !x.window, etc. per suggestion of
;   Mark Hadfield (hadfield@storm.greta.cri.nz), 7 Apr 94, FKK
; add a /default keyword restore IDL's default values of system vars,
;   7 Apr 94, FKK
; modify two more sys vars !x(y).tickformat to suppress user-formatted
;   ticknames, per suggestion of Mark Hadfield (qv), 8 Apr 94, FKK
;-
pro multiplot,help=help,pmulti $
  ,initialize=initialize,reset=reset,default=default $
  ,rowmajor=rowmajor,verbose=verbose
;
; =====>> COMMON
;
common multiplot $
  ,nplots $ ; [# of plots along x, # of plots along y]
  ,nleft $ ; # of plots remaining---like the first element of !p.multi
  ,pdotmulti $ ; saved value of !p.multi
  ,margins $ ; calculated margins based on !p.multi or pmulti
  ,pposition $ ; saved value of !p.position
  ,colmajor $ ; flag for column major order
  ,noerase $ ; saved value of !p.noerase
  ,xtickname $ ; Original value
  ,ytickname $ ; Original value
  ,xtickformat $; Original value
  ,ytickformat  ; Original value
;
; =====>> HELP
;
;on_error,2
if keyword_set(help) then begin & doc_library,'multiplot' & return & endif
;
; =====>> RESTORE IDL's DEFAULT VALUES (kill multiplot's influence)
;
if keyword_set(default) then begin
  !p.position = 0
  !x.tickname = ''
  !y.tickname = ''
  !x.tickformat = ''
  !y.tickformat = ''
  !p.multi = 0
  !p.noerase = 0
  nleft = 0
  nplots = [1,1]
  pdotmulti = !p.multi
  margins = 0
  pposition = !p.position
  noerase = !p.noerase
  xtickname = !x.tickname
```

```
  ytickname = !y.tickname
  xtickformat = !x.tickformat
  ytickformat = !y.tickformat
  if keyword_set(verbose) then begin
    message,/inform,'Restore IDL''s defaults for affected system variables.'
    message,/inform,'Reset multiplot''s common to IDL''s defaults.'
    endif
  return
  endif
;
; =====>> RESTORE SAVED SYSTEM VARIABLES
;
if keyword_set(reset) then begin
  if n_elements(pposition) gt 0 then begin
    !p.position = pposition
    !x.tickname = xtickname
    !y.tickname = ytickname
    !x.tickformat = xtickformat
    !y.tickformat = ytickformat
    !p.multi = pdotmulti
    !p.noerase = noerase
    endif
  nleft = 0
  if keyword_set(verbose) then begin
    coords = '['+string(!p.position,form='(3(f4.2,","),f4.2)')+']'
    multi = '['+string(!p.multi,form='(4(i2,","),i2)')+']'
    message,/inform,'Reset.  !p.position='+coords+', !p.multi='+multi
    endif
  return
  endif
;
; =====>> SETUP: nplots, MARGINS, & SAVED SYSTEM VARIABLES
;
if n_elements(nleft) eq 1 then init = (nleft eq 0) else init = 1
if (n_elements(pmulti) eq 2) or (n_elements(pmulti) eq 5) then init = 1
if (n_elements(!p.multi) eq 5) then begin
  if (!p.multi(1) gt 0) and (!p.multi(2) gt 0) then init = (!p.multi(0) eq 0)
  endif
if init or keyword_set(initialize) then begin
  case n_elements(pmulti) of
  0:begin
    if n_elements(!p.multi) eq 1 then return ; NOTHING TO SET
    if n_elements(!p.multi) ne 5 then message,'Bogus !p.multi; aborting.'
    nplots = !p.multi(1:2) > 1
    if keyword_set(rowmajor) then colmajor = 0 else colmajor = !p.multi(4) eq 0
    end
  2:begin
    nplots = pmulti
```

```
    colmajor = not keyword_set(rowmajor) ; D=colmajor: left to rt 1st
    end
  5:begin
    nplots = pmulti(1:2)
    if keyword_set(rowmajor) then colmajor = 0 else colmajor = pmulti(4) eq 0
    end
  else: message,'pmulti can only have 0, 2, or 5 elements.'
  endcase
  pposition = !p.position   ; save sysvar to be altered
  xtickname = !x.tickname
  ytickname = !y.tickname
  xtickformat = !x.tickformat
  ytickformat = !y.tickformat
  pdotmulti = !p.multi
  nleft = nplots(0)*nplots(1)   ; total # of plots
  !p.position = 0    ; reset
  !p.multi = 0
  plot,/nodata,xstyle=4,ystyle=4,!x.range,!y.range,/noerase ; set window & region
  margins = [min(!x.window)-min(!x.region) $ ; in normlized coordinates
    ,min(!y.window)-min(!y.region) $
    ,max(!x.region)-max(!x.window) $
    ,max(!y.region)-max(!y.window)]
  noerase = !p.noerase
  !p.noerase = 1    ; !p.multi does the same
  if keyword_set(verbose) then begin
    major = ['across then down (column major).','down then across (row major).']
    if colmajor then index = 0 else index = 1
    message,/inform,'Initialized for '+strtrim(nplots(0),2) $
    +'x'+strtrim(nplots(1),2)+', plotted '+major(index)
    endif
;  print,margins,'=margins'
  if keyword_set(initialize) then return
endif
;
; =====>> Define the plot region without using !p.multi.
;
cols = nplots(0)   ; for convenience
rows = nplots(1)
nleft = nleft - 1   ; decrement plots remaining
cur = cols*rows - nleft   ; current plot #: 1 to cols*rows
idx = [(1.-margins(0)-margins(2))/cols $
  ,(1.-margins(1)-margins(3))/rows] ; normalized coords per plot
if colmajor then begin   ; location in matrix of plots
  col = cur mod cols
  if col eq 0 then col = cols
  row = (cur-1)/cols + 1
endif else begin   ; here (1,2) is 1st col, 2nd row
  row = cur mod rows
```

```
  if row eq 0 then row = rows
  col = (cur-1)/rows + 1
endelse
pos =  [(col-1)*idx(0),(rows-row)*idx(1),col*idx(0),(rows-row+1)*id x(1)] $
  + [margins(0),margins(1),margins(0),margins(1)]
;print,row,col,rows,cols,pos
;
; =====>> Finally set the system variables; user shouldn't change them.
;
!p.position = pos
onbottom = (row eq rows) or (rows eq 1)
onleft = (col eq 1) or (cols eq 1)
if onbottom then !x.tickname = xtickname else !x.tickname = replicate(' ',30)
if onleft then !y.tickname = ytickname else !y.tickname = replicate(' ',30)
if onbottom then !x.tickformat = xtickformat else !x.tickformat = ''
if onleft then !y.tickformat = ytickformat else !y.tickformat = ''
if keyword_set(verbose) then begin
  coords = '['+string(pos,form='(3(f4.2,","),f4.2)')+']'
  plotno = 'Setup for plot ['+strtrim(col,2)+','+strtrim(row,2)+'] of ' $
    +strtrim(cols,2)+'x'+strtrim(rows,2)
  message,/inform,plotno+' at '+coords
  endif
;stop
return
end
```

--

=Fred Knight (knight@ll.mit.edu) (617) 981-2027
C-483\\MIT Lincoln Laboratory\\244 Wood Street\\Lexington, MA  02173

---

## Subject: Re: contiguous stacked plots

Posted by mark_cadwell on Mon, 09 Jan 1995 17:27:34 GMT

View Forum Message <> Reply to Message

Save yourself a *lot* of work and pick whichever plot has the largest X
axis range and plot it.  Then overplot the rest of them.

--

---

mark_cadwell@qmail4.trw.sp.com

---