
Subject: Re: Colorbar Thinking in the Shower
Posted by [David Fanning](#) on Tue, 13 Sep 2011 16:30:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

> I believe the best place in the world to think is in the
> shower, don't you?

More typos than usual this morning, but I'm late for a class! ;-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Colorbar Thinking in the Shower
Posted by [Mark Piper](#) on Tue, 13 Sep 2011 20:05:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 9/13/2011 10:28 AM, David Fanning wrote:

> the problem of displaying
> two data sets, each with a different data range,
> with a single color bar to explain the colors in
> each?

Here's an example of displaying two plots with differing ranges, referenced by a single colorbar. It suffers from the same drawbacks as the other examples I've posted, but maybe it can be a starting point for discussion.

mp

```
pro ng_twoplots_singlecolorbar
  compile_opt idl2
```

```
  ; Example functions to plot. The first (f1) has a range of [0,100], the
  ; second (f2) a range of [0,70].
  d = dist(41)
  max1 = 100.0
  max2 = 70.0
  f1 = d / max(d) * max1
```

```

f2 = d / max(d) * max2

; Explicitly set 11 contour levels: [0, 10, 20, ... 100].
n_levels = 11
levels = findgen(n_levels)/(n_levels-1)*max1

; Make a step color table for the first contour plot. The color
table STEP_CT
; is a [256,3] array, but there are only n_levels=11 distinct colors (to
; check, load & view the color table in XPALETTE). The indices into
the color
; tables (both original and step) are contour levels interpolated to
the
; range of color table indices (i.e., the byte range).
ct_number = 4
ct_indices = bytscl(levels)
loadct, ct_number, rgb_table=ct, /silent
step_ct = congrid(ct[ct_indices, *], 256, 3)

; Display the first function using the step color table and the
; interpolated indices.
c1 = contour(f1, $
    layout=[2,1,1], $
    c_value=levels, $
    rgb_table=step_ct, $
    rgb_indices=ct_indices, $
    /fill, $
    title='Max = ' + strtrim(max1,2), $
    window_title='Discrete Colorbar Example')

; Display the second function using the original color table and the
; interpolated indices.
c2 = contour(f2, $
    layout=[2,1,2], $
    /current, $
    c_value=levels, $
    rgb_table=ct_number, $ ; compare with c1
    rgb_indices=ct_indices, $
    /fill, $
    title='Max = ' + strtrim(max2,2))

; Display colorbar with first contour plot as a target. It needs
n_levels + 1
; ticks to make labels line up correctly.
tick_labels = [strtrim(fix(levels), 2), ""] ; append empty string
cb = colorbar( $
    target=c1, $
    ticklen=0, $

```

```
major=n_levels+1, $
tickname=tick_labels, $
font_size=8, $
position=[0.2, 0.06, 0.8, 0.09])
end
```

Subject: Re: Colorbar Thinking in the Shower

Posted by [David Fanning](#) on Tue, 13 Sep 2011 20:30:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Mark Piper writes:

> Here's an example of displaying two plots with differing ranges,
> referenced by a single colorbar. It suffers from the same drawbacks as
> the other examples I've posted, but maybe it can be a starting point
> for discussion.

OK, perhaps I am just too handicapped by not being able to use the TICKNAME keyword, but I think you and I probably agree that the *default* labeling of the color bar is only suited for a particular data set, namely the target of the color bar.

But, we probably also agree that the colors in the color bar are also completely arbitrary and the labeling of those colors can be anything we want it to be. You label with TICKNAME, I have to do it in a different way because TICKNAME crashes my machine.

But, since the labeling is arbitrary, and thus there is no particular need to attach the color bar to a particular target (which in my view just causes massive user confusion), why don't you just allow the user to specify a color bar range with the color bar itself? Why "attach" it to something when that almost always causes you to have to modify the color bar range with the TICKNAME keyword? (At least when you are comparing two different data sets.) Using TICKNAME is onerous because you make the user format and create strings for the labels of the (nearly always?) incorrect tick mark values.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Colorbar Thinking in the Shower
Posted by [Mark Piper](#) on Wed, 14 Sep 2011 15:13:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 9/13/2011 2:30 PM, David Fanning wrote:

>
> But, since the labeling is arbitrary, and thus there
> is no particular need to attach the color bar to a
> particular target (which in my view just causes
> massive user confusion), why don't you just allow
> the user to specify a color bar range with the
> color bar itself? Why "attach" it to something when
> that almost always causes you to have to modify the
> color bar range with the TICKNAME keyword? (At least
> when you are comparing two different data sets.) Using
> TICKNAME is onerous because you make the user format
> and create strings for the labels of the (nearly
> always?) incorrect tick mark values.

Yes, I agree. Other users have also brought up this behavior, so we (that's me, Chris, Bill, others) have been talking about it over the past few months. I had to check with Chris about scheduling -- he told me that in the next few weeks he'll be modifying COLORBAR to allow it to be decoupled from data. I haven't seen how he'll do it, but I'd like to use your comments above as a guide.

mp
