Subject: Re: Box Axes with Map Function
Posted by David Fanning on Sun, 02 Dec 2012 17:04:30 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> ; Something to see on the map.
> c = MapContinents(Color=cgColor('tomato', /Triple, /Row))

Whoops! Didn't mean to stick some Coyote Graphics in there!
Try this, instead:

c = MapContinents(Color=!Color.red)

Cheers,

David

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

Subject: Re: Box Axes with Map Function
Posted by David Fanning on Sun, 02 Dec 2012 18:35:48 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I *finally* discovered a way to create box axes with
> the function graphics Map function that are labeled
> in the way we expect box axes to be labeled.

I've written an article that describes this in
more detail:

http://www.idlcoyote.com/ng_tips/boxaxes.php

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Subject: Re: Box Axes with Map Function
Posted by lecacheux.alain on Sun, 02 Dec 2012 20:35:30 GMT
View Forum Message <> Reply to Message

Le dimanche 2 décembre 2012 19:35:48 UTC+1, David Fanning a écrit :
> David Fanning writes:
>
>
>
>> I *finally* discovered a way to create box axes with
>
>> the function graphics Map function that are labeled
>
>> in the way we expect box axes to be labeled.
>
>
>
> I've written an article that describes this in
>
> more detail:
>
>
>
> http://www.idlcoyote.com/ng_tips/boxaxes.php
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

I think that you may also draw *only one* map projection, but by adding a second grid to the map:

```
mp1 = map('Equirectangular', CENTER_LONGITUDE=180, $
   POSITION=[0.1,0.1,0.90,0.75], $
   LABEL_POSITION = 0, BOX_AXES=1, $
   GRID_LATITUDE = 30, GRID_LONGITUDE = 45, $
   ASPECT_RATIO=0, LIMIT=[-89.99, 0, 89.99, 360])
mp1['Latitudes'].label_angle=90
mp1['Longitudes'].label_angle=0

grid = MAPGRID( $
  LONGITUDE_MIN=0, LONGITUDE_MAX=360, $
  LATITUDE_MIN=-90, LATITUDE_MAX=90, $
  GRID_LONGITUDE=45, GRID_LATITUDE=30, $
  LABEL_POSITION=1)
foreach g,grid.latitudes do g.label_angle=270
foreach g,grid.longitudes do g.label_angle=0
```

alain.

Subject: Re: Box Axes with Map Function
Posted by David Fanning on Sun, 02 Dec 2012 21:53:43 GMT
View Forum Message <> Reply to Message

Alain writes:

> I think that you may also draw *only one* map projection, but by adding a second grid to the map:
>
> mp1 = map('Equirectangular', CENTER_LONGITUDE=180, $
>   POSITION=[0.1,0.1,0.90,0.75], $
>   LABEL_POSITION = 0, BOX_AXES=1, $
>   GRID_LATITUDE = 30, GRID_LONGITUDE = 45, $
>   ASPECT_RATIO=0, LIMIT=[-89.99, 0, 89.99, 360])
> mp1['Latitudes'].label_angle=90
> mp1['Longitudes'].label_angle=0
>
> grid = MAPGRID($
>   LONGITUDE_MIN=0, LONGITUDE_MAX=360, $
>   LATITUDE_MIN=-90, LATITUDE_MAX=90, $
>   GRID_LONGITUDE=45, GRID_LATITUDE=30, $
>   LABEL_POSITION=1)
> foreach g,grid.latitudes do g.label_angle=270
> foreach g,grid.longitudes do g.label_angle=0

I'm detecting a pattern here with function graphics.
I try something and can't get it to work. I ask for
help. No one helps me. I muck around for several weeks
and finally cobble something together that works.
I write an article to save everyone else the trouble
of the two week muck-about, and within seconds someone
writes in with the correct answer!

I guess I should probably think about writing the
articles sooner. :-(

Thanks for the help!

Cheers,

David

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

Subject: Re: Box Axes with Map Function
Posted by Paul Van Delst[1] on Tue, 04 Dec 2012 15:18:03 GMT
View Forum Message <> Reply to Message

Hello,

On 12/02/12 16:53, David Fanning wrote:
> Alain writes:
>
>> I think that you may also draw *only one* map projection, but by adding a second grid to the map:
>>
>> mp1 = map('Equirectangular', CENTER_LONGITUDE=180, $
>> POSITION=[0.1,0.1,0.90,0.75], $
>> LABEL_POSITION = 0, BOX_AXES=1, $
>> GRID_LATITUDE = 30, GRID_LONGITUDE = 45, $
>> ASPECT_RATIO=0, LIMIT=[-89.99, 0, 89.99, 360])
>> mp1['Latitudes'].label_angle=90
>> mp1['Longitudes'].label_angle=0
>>
>> grid = MAPGRID($
>> LONGITUDE_MIN=0, LONGITUDE_MAX=360, $

>>   LATITUDE_MIN=-90, LATITUDE_MAX=90, $
>>   GRID_LONGITUDE=45, GRID_LATITUDE=30, $
>>   LABEL_POSITION=1)
>> foreach g,grid.latitudes do g.label_angle=270
>> foreach g,grid.longitudes do g.label_angle=0
>
> I'm detecting a pattern here with function graphics.
> I try something and can't get it to work. I ask for
> help. No one helps me. I muck around for several weeks
> and finally cobble something together that works.
> I write an article to save everyone else the trouble
> of the two week muck-about, and within seconds someone
> writes in with the correct answer!

If it makes you feel any better, I don't see how Alain's answer is any more "correct" compared to yours.

My take away from this: you have to jump through a bunch of arcane hoops to get a result that previously was simple to achieve.

Reminds me of when I was talking with a Qantas pilot many years ago - he'd been flying the latest, greatest 747's and was telling me about the continual training and re-qualification he had to go through. his take was something like "If they want me to jump through all those hoops to fly those planes, that's fine with me".

This function graphics example is equivalent to all that hoop jumping, but with the reward being the beat up two-seat cessna at the local municipal airport.

> I guess I should probably think about writing the
> articles sooner. :-(

Additionally, it also been my experience with various OO languages (even IDL pre-OO-ish-ness) that there are several ways to achieve the same thing. What ends up being the idiomatic example can take a while to determine as the community cogitates over the issue.

cheers,

paulv

p.s. I'm a little bit down on my Function Graphics boosterism because lately I've had to plot many spectral and interfergram plots consisting of 2^19 points. My original DG code (using my own zooming widget) does it in a second or two - and I can zoom in and redisplay quickly at will inspecting the data. The FG replacement? Fuhgeddaboutit.