Subject: Multiple axes and plots with NG
Posted by Matthew Argall on Sat, 11 Jan 2014 19:17:45 GMT
View Forum Message <> Reply to Message

Hi,

I am trying to reproduce the following plot from the coyote library using function graphics, as it similar to a plot of my own that I need to make.

http://www.idlcoyote.com/gallery/#ADDITIONAL_AXES_PLOT

My attempt is below.

Everything seems fine, except for the blue axis, which did not obey the axis range I gave it. I was hoping to use "emptyPlot" as the target for the y-axes, then use the axes as the targets for the plots. This did not work, though, since each additional axis reset the axis ranges of the others...

Can anyone help improve this?.... I was hoping to use the Target and Overplot keywords more, and not just position one plot on top of the other, then hide their axes...

Thanks in advance

```
data_1 = cgScaleVector(cgDemodata(17), 0.0, 1.0)
data_2 = cgScaleVector(cgDemodata(17), 0.0, 1000.0)
data_3 = (Findgen(101)+1) / 5

position = [0.15, 0.15, 0.7, 0.820]
thick = 2

;Create the empty plot. Hide the y-axes.
emptyPlot = Plot(data_1, Position=position, /NoData)
emptyAxes = emptyPlot.AXES
emptyAxes[1].HIDE = 1
emptyAxes[3].HIDE = 1

;Add the first set of data and the left axes.
location = emptyPlot.ConvertCoord(position[0:1], /Normal, /To_Data)
Plot1 = Plot(data_1, /Current, Overplot=emptyPlot, Color='red', Thick=thick)
Axis1 = Axis('Y', Location=location, Target=emptyPlot, Color='red', Title='Data 1')

;Add the second set of data. Turn off the axes.
Plot2 = Plot(data_2, /Current, Position=position, Color='green', LineStyle=2, Thick=thick)
Plot2Axes = Plot2.AXES
ForEach ax, Plot2Axes do ax.HIDE=1

;Add the right axis
```

```
location = Plot2.ConvertCoord(position[2:3], /Normal, /To_Data)
Axis2 = Axis('Y', Location=location, Target=Plot2, TextPos=1, Color='green', Title='Data 2',
Axis_Range=[0,1000])

;Add the third set of data. Turn off the axes
Plot3 = Plot(data_3, /Current, Position=position, Color='blue', LineStyle=1, Thick=thick)
Plot3Axes = Plot3.AXES
ForEach ax, Plot3Axes do ax.HIDE=1

;Add a third axis
location = Plot3.ConvertCoord([0.85, 0.15], /Normal, /To_Data)
Axis3 = Axis('Y', Location=location, Target=Plot3, TextPos=1, Color='blue', Title='Data 3',
Axis_Range=[0.1,100], /Log)
```

Subject: Re: Multiple axes and plots with NG
Posted by chris_torrence@NOSPAM on Mon, 13 Jan 2014 03:20:23 GMT

View Forum Message <> Reply to Message

Hi Matthew,

Well, first of all, your final plot looks really nice. Just like David Fanning's. Second, I don't think you can really improve the code much. By definition, once a plot is created, then all other plots that go on top of it (using say /overplot) must share the same "data space". In other words, they are all assumed to have the same data units. So your method of doing a new plot with /current is more or less correct. I was able to simplify your code a little bit by avoiding the original "empty" plot, and also by suppressing the axes for the later plots:

```
data_1 = cgScaleVector(cgDemodata(17), 0.0, 1.0)
data_2 = cgScaleVector(cgDemodata(17), 0.0, 1000.0)
data_3 = (Findgen(101)+1) / 5

position = [0.15, 0.15, 0.7, 0.820]
thick = 2

;Add the first set of data and the left axes.
Plot1 = Plot(data_1, Color='red', Thick=thick, Position=position, Ytitle='Data 1')
ay1 = Plot1['Axis 1']
ay1.Color = 'red'
ay3 = Plot1['Axis 3']
ay3.hide = 1

;Add the second set of data. Turn off the axes.
Plot2 = Plot(data_2, /Current, AXIS_STYLE=0, Position=position, Color='green', LineStyle=2,
Thick=thick)

;Add the right axis
Axis2 = Axis('Y', Location=[(Plot2.xrange)[1], 0, 0], Target=Plot2, TextPos=1, $
  Color='green', Title='Data 2')

;Add the third set of data. Turn off the axes
Plot3 = Plot(data_3, /Current, AXIS_STYLE=0, Position=position, Color='blue', LineStyle=1,
Thick=thick)

;Add a third axis
xr = Plot3.xrange
Axis3 = Axis('Y', Location=[xr[1]+0.25*(xr[1]-xr[0]),0,0], Target=Plot3, TextPos=1, $
  Color='blue', Title='Data 3', /Log)
```

Hope this helps!

Cheers,
Chris
ExelisVIS

Subject: Re: Multiple axes and plots with NG
Posted by Matthew Argall on Mon, 13 Jan 2014 13:52:08 GMT

View Forum Message <> Reply to Message

> Well, first of all, your final plot looks really nice. [...] By definition, once a plot is created, then all other plots that go on top of it (using say /overplot) must share the same "data space". In other words, they are all assumed to have the same data units. [...]

Great! Thanks! I had an idea of what /Overplot does, but it is good to know for sure. Could you clarify what TARGET does, too?

In the following example, "theAxis" can have a different range from "thePlot", but they share the same scale, so "theAxis" does not become completely visible until I expand the axis range of "thePlot". I suppose if I wanted the axes to be the same length and have different scales, I would have to use the Coord_Transform keyword, right? (any chance of getting an exponential transform for linear/log-scaled axes?)...

```
thePlot = Plot(/Test)
theAxis = Axis('Y', LOCATION=[200, -1], Axis_Range=[-2,2], Color='Blue', TextPos=1,
Target=thePlot)
thePlot.XRange = [-5,5]
```

Subject: Re: Multiple axes and plots with NG
Posted by chris_torrence@NOSPAM on Mon, 13 Jan 2014 15:16:36 GMT

View Forum Message <> Reply to Message

On Monday, January 13, 2014 6:52:08 AM UTC-7, Matthew Argall wrote:
>> Well, first of all, your final plot looks really nice. [...] By definition, once a plot is created, then all other plots that go on top of it (using say /overplot) must share the same "data space". In other words, they are all assumed to have the same data units. [...]
>
>
>
> Great! Thanks! I had an idea of what /Overplot does, but it is good to know for sure. Could you clarify what TARGET does, too?
>
>
>
> In the following example, "theAxis" can have a different range from "thePlot", but they share the same scale, so "theAxis" does not become completely visible until I expand the axis range of "thePlot". I suppose if I wanted the axes to be the same length and have different scales, I would have to use the Coord_Transform keyword, right? (any chance of getting an exponential transform for linear/log-scaled axes?)...
>
>
>
> thePlot = Plot(/Test)
>
> theAxis = Axis('Y', LOCATION=[200, -1], Axis_Range=[-2,2], Color='Blue', TextPos=1, Target=thePlot)
>
> thePlot.XRange = [-5,5]

Hi Matthew,

Yes, the TARGET just says that this axis "belongs" to this dataspace, and should share the same range as that dataspace. So, you are correct, in your example, that axis is going to have a "range" of the original plot (probably [-1,1]), regardless of what you put in for the axis_range. The axis_range is really only useful for creating an axis that only extends partway. For example:

```
thePlot = Plot(/Test)
theAxis = Axis('Y', LOCATION=[200, -1], Axis_Range=[0,1], Color='Blue', TextPos=1, Target=thePlot)
```

I'd have to think carefully about the exponential transform. I'm not sure what that would mean for the actual data scaling within the dataspace. It's supposed to be tied 1-1 with the axis.

-Chris

Subject: Re: Multiple axes and plots with NG
Posted by Matthew Argall on Mon, 13 Jan 2014 16:43:15 GMT
View Forum Message <> Reply to Message

Thanks again!

Subject: Re: Multiple axes and plots with NG
Posted by Gompie on Tue, 14 Jan 2014 03:05:17 GMT
View Forum Message <> Reply to Message

Hi,
This plot works well but varying the limits of Axis_range in plot2 does not scale the plot2.
Any hints on how to achieve that,
GlanPlon

Subject: Re: Multiple axes and plots with NG
Posted by Matthew Argall on Tue, 14 Jan 2014 20:11:35 GMT
View Forum Message <> Reply to Message

> This plot works well but varying the limits of Axis_range in plot2 does not scale the plot2.

Plots do not have an "Axis_Range" keyword. Instead, they use "[XYZ]Range". In the above example, if you can do, for example,

Plot2.YRange = [100,1000]

If instead you are using Axis2, then

Axis2.Axis_Range = [100,1000]

will scale the axis (the axis shortens to be in the proper range), but leaves the actual data space alone. The plot itself does not zoom to the range [100,1000].