Subject: Re: Plot multiple axes with log and linear scales
Posted by Conor on Mon, 10 Aug 2009 13:29:11 GMT
View Forum Message <> Reply to Message

On Jul 30, 12:21 pm, JDS <jdtsmith.nos...@yahoo.com> wrote:
> On Jul 29, 11:59 am, David Fanning <n...@dfanning.com> wrote:
>
>
>
>> Conor writes:
>>> I need to make a plot with multiple x-axes. I've done this before
>>> without any trouble when both axes had a linear scale, and so I would
>>> use the axis command to create the second plot, and just set the
>>> xrange keyword to be whatever I needed it to be. Now however things
>>> are more complicated, because one axis is a linear scale, and the
>>> other axis is sorta kinda logarithmic scale.
>
>> Here is how you add a second, logarithmic axis:
>
>> http://www.dfanning.com/tips/another_yaxis.html
>
>>> There is a one-to-one
>>> relationship between the axes, but the relationship is not in the
>>> least bit simple. Basically, I need a way to tell IDL, for these
>>> values on the first x-axis, plot these values for the second x-axis.
>>> Anyone know how to do this?
>
>> I don't have the foggiest idea of what this means. Sorry. :-(
>
> I think this means there is a nonlinear relationship between the first
> axis values, and the second axis. Classic example: redshift, and
> lookback time: related, but not (at all!) linearly. The way you do
> this is to create another axis with the same range as the first, but
> give it an [XYZ]TICKFORMAT function which does the nonlinear
> conversion for you, ala:
>
> IDL> plot,indgen(10),YRANGE=[1,12],YTITLE='first linear
> axis',YSTYLE=9,POSITION=[.1,.1,.9,.9],CHARSIZE=2
> IDL>
> axis,YSTYLE=1,YAXIS=1,YTICKFORMAT='conv_axis',CHARSIZE=2,YTI TLE='second
> non-linear axis'
>
> where 'conv_axis' is the name of the function which does the
> conversion (e.g.):
>
> function conv_axis,axis,index,value
> return,string(FORMAT='(F0.1)',value^1.5*exp(-value^2/100))
> end

>
> You'll notice (as in this case), you don't even need to maintain
> bijection (this is just a made up conversion function). If instead of
> matching the first axis' tick locations and ending up with random non-
> round values, you can use YTICKV and YTICKS to pass those value which
> *before non-linear conversion* work out to the correct converted
> (usually round) numbers. This of course requires you to invert the
> conversion equation, which sometimes you can do by hand, but sometimes
> you'll have to do numerically (FX_ROOT is your friend here).
>
> JD

Perfect, thanks! In fact redshift-age was exactly what I was doing :)
