Subject: error bars
Posted by vek on Thu, 24 Nov 1994 03:14:26 GMT
View Forum Message <> Reply to Message

I'm looking for a simple way to make *both* X and Y error bars. The USERLIB error bar routines only supply Y error bars (why anyone would think that's sufficient I have no idea, grumble grumble). One program I'm using draws them by individually connecting each plus & minus value - is there no way to do the whole array at once?

Thanks,
Vince, who is completely bewildered by the lack of X error bar plotting capability in a multitude of packages. Does everyone besides me live in a perfect world?

Subject: Re: error bars
Posted by lrn on Mon, 28 Nov 1994 21:27:36 GMT
View Forum Message <> Reply to Message

vek@spacsun.rice.edu (Vincent E. Kargatis) writes:

> I'm looking for a simple way to make *both* X and Y error bars. The
> USERLIB error bar routines only supply Y error bars (why anyone would think
> that's sufficient I have no idea, grumble grumble). One program I'm using
> draws them by individually connecting each plus & minus value - is there no
> way to do the whole array at once?

Don't the USERLIB routines also draw error bars individually by connecting each plus and minus value? I have a program that not only plots error bars in both X and Y directions, but will also handle cases where the error-bars of individual points are not parallel to the overall plot axes nor to those of other points. (Correlated errors) If anyone is interested, I can post it.

Larry

--

Larry R. Nittler	Human beings were invented by water as
lrn@howdy.wustl.edu	a means of transporting itself from
Interstellar Dust Buster	one place to another. -- Tom Robbins

Subject: Re: Error bars
Posted by MA on Thu, 08 Sep 2005 19:55:36 GMT
View Forum Message <> Reply to Message

Sorry, typo. The piece of code shoud be
p=(s^(n-1.)*(t-s)^n*(1./y-1.)^ n*Factorial(2.*n-2.))/ $
((s/y+t-2.*s)^(2.*n-1.)*(y-1.) ^2.*Factorial(n)*Factorial(n-2.))

Subject: Re: Error bars
Posted by Craig Markwardt on Fri, 09 Sep 2005 07:45:57 GMT
View Forum Message <> Reply to Message

"MA" <ahlgrimm@lamar.colostate.edu> writes:
> Hello,
> I have a (scientific) problem and could use some help on how to best
> solve it with IDL. I am trying to calculate a confidence interval on a
> cloud fraction that can range between 0 and 1. I have a mathematical
> function that calculates the probability of occurrence of all cloud
> fractions between 0 and 1, given a couple of input parameters, e.g.
>
> t=250. ;; constant, number of samples
> s=120. ;; number of cloudy samples, can range between 0 and t
> n=3. ;; number of 'gaps' between continuously cloudy samples
> y=IndGen(1000)/999. ;; this is arbitrary, could choose more/less
> points between 0 and 1
> p=FltArr(1000)
> p=(s^(n-1.)*(t-s)^n*(1./y-1.)^n*Factorial(2.*n-1.))/ $
> ((s/y+t-2.*s)^(2.*n-1.)*(y-1.)^2.*Factorial(n)*Factorial(n-1 .))
>
> This curve can be a nice bell shape (as with the parameters above), but
> can also be flat (if probability is same everywhere) or be very skewed,
> to the point where the curve goes to infinity (in IDL world) (you can
> fiddle around with s and n for that, though n>=1). The first and last
> entry in the array are often either NaN (s/y=NaN for y=0) or Inf.
> To find the 90% error bar on the cloud fraction, I have to find the two
> cloud fractions between which 90% of the area under the curve lies.
> Is there a smart way to calculate this error bar? Graphically, I'd draw
> a horizontal line across the plot, see at what cloud fractions it
> intercepts the curve, calculate the area under the curve between those
> cloud fractions. If it's more/less than 90%, I'd lift/lower the
> horizontal line and repeat. I've tried to mimic this process in my
> program, but it takes forever and is not very accurate. Also, the
> infinity/NaN values are really annoying (though pysically correct,
> since the function only applies for fractions between 0,1, exclusive of
> those values), because the total area under the curve is no longer 1.
> Any suggestions on how to do this better? Maybe something with
> Histogram?

The easiest way to do something like this is to make a cumulative
probability distribution.

Since you have a uniform distribution of "Y" ordinates, it's very easy: just use TOTAL,

```
PCUM = TOTAL(P, /CUMULATIVE)  ;; Cumulative distribution
PCUM = PCUM / MAX(PCUM)       ;; Be safe: normalize to 1
```

Then it's just a matter of picking out whatever confidence interval you want. If you want 90% confidence, you might be satisfied by taking the 5% and 95% crossover points.

```
ISTART = WHERE(PCUM GE 0.05) & ISTART = ISTART(0)
ISTOP  = WHERE(PCUM LE 0.95) & ISTOP  = MAX(ISTOP)
```

And then your confidence interval is Y(ISTART:ISTOP).

You will probably have to do some additional error checking if you have NaNs. Also, you will need to ensure that the Y array is finely enough sampled to capture the 5% and 95% crossover points.

Good luck,
Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response