Subject: Fill Missing values with NaN
Posted by idlusr on Thu, 10 Aug 2017 19:00:39 GMT
View Forum Message <> Reply to Message

How can one do the following in IDL:

time = [1, 4, 7, 8]
dens =[80,60,30,50]

alltimes = [0,1,2,3,4,5,6,7,8,9]
densnew = [NaN, 80, NaN, NaN, 60, NaN, NaN, 30, 50, NaN]

i.e., compare time to all times, allowing densnew to have same # of elements as all times but ONLY the densities in dens. Missing elements in densnew (depending on time) are given NaN value.

Thank you!

Subject: Re: Fill Missing values with NaN
Posted by idlusr on Thu, 10 Aug 2017 19:03:59 GMT
View Forum Message <> Reply to Message

On Thursday, August 10, 2017 at 12:00:41 PM UTC-7, idlusr wrote:
> How can one do the following in IDL:
>
> time = [1, 4, 7, 8]
> dens =[80,60,30,50]
>
> alltimes = [0,1,2,3,4,5,6,7,8,9]
> densnew = [NaN, 80, NaN, NaN, 60, NaN, NaN, 30, 50, NaN]
>
> i.e., compare time to all times, allowing densnew to have same # of elements as all times but ONLY the densities in dens. Missing elements in densnew (depending on time) are given NaN value.
>
> Thank you!

Forgot to mention, the values in my data set are type double for time & float for density. Thanks again!

Subject: Re: Fill Missing values with NaN
Posted by Helder Marchetto on Fri, 11 Aug 2017 06:51:21 GMT
View Forum Message <> Reply to Message

I don't get the question. So I will just suggest the use of the function finite() to check for not-NaN

values in densnew. This will give you the valid indices.

valInd = finite(densnew)

times_fromAlltimes = alltimes[valInd]
dens_fromDensnew   = densnew[valInd]

Hope it helps.
Helder

Subject: Re: Fill Missing values with NaN
Posted by Matthew Argall on Fri, 11 Aug 2017 12:31:53 GMT
View Forum Message <> Reply to Message

The example below works well with integer times. For floats, you should define "alltimes" as the beginning of the sampling interval and "times" as the center of the sampling interval. This is in-part because of how Value_Locate works and in-part because of the discreteness of floating point values.

```
;Time Arrays
time     = [1, 4, 7, 8]
alltimes = IndGen(10)

;Data Arrays
dens    = [80, 60, 30, 50]
densnew = Replicate( !Values.D_NaN, N_Elements(alltimes) )

;Locate "time" within "alltimes"
iTime    = Value_Locate(alltimes, time)

;Fill the new density array
densnew[iTime] = dens

;Result
print, densnew, FORMAT='(10(f5.2, 2x))'
  NaN  80.00    NaN    NaN  60.00    NaN    NaN  30.00  50.00    NaN
```
