Subject: Re: Removing and Replacing Nan values in IDL
Posted by adhdunn on Thu, 19 Jul 2012 18:52:11 GMT
View Forum Message <> Reply to Message

On Thursday, July 19, 2012 9:47:39 AM UTC-4, adh...@gmail.com wrote:
> Hello,
>
> I am working with MODIS Land Surface Temperature and need to clean my data to remove and replace the NaN values. I currently have a little code that I have written to scale the data, convert from Kelvin to Celsius, and average the night and day temp observations, but I am having trouble writing the (probably very simple) lines that will accomplish this last goal of NaN removal.
>
> What I would like to do is to set up some kind of statement where the program looks for any NaN values in 'image1' and checks if the corresponding location in 'image2' also has NaN values.
>
> The next step would be to say if image1 has NaN values, but image2 does not then replace that pixel in image1 with image2. Then you would also do this for image2 replacing with the image1 value. If they both have NaN values you do nothing.
>
> Does this make sense?
>
> If anyone would like to help me I can post my code and give more details.
> Any guidance would definitely be appreciated!!
>
> Thank you!
> Allisyn

Hi Alain,

Thank you for the suggestion! I am not sure if you saw the reply post from an earlier suggestion, but I am in need of figuring out where to insert these code lines. I currently have it written as follows, and I will need to retain the component that creates the average between image1 and image2 (assuming the pixel is not being replaced due to its NaN value). Any thoughts?

```
openr, lun, 'F:\PhD\Data\MODIS\LST\h9v4_h10v4_LST_daynight', /get_lun

band = fltarr(2401, 1200)
image = assoc(lun, band)

openw, lun, 'F:\PhD\Data\MODIS\LST\h9v4_h10v4_LST_mnthavgTST_2010', /get_lun
filter = assoc(lun, band)

imageout = image[0]
imagetwo = image[1]


a= image[0]/image[1]
```

```
; evaluate division result to make sure it's a valid #
; then examine all indices that represent valid numbers to see if they're GT 1.2
; i.e., find places where image[0] values are 1.2x greater than corresponding
 image[1] values

removefirst = where(a[where(FINITE(a) eq 1)] gt 1.2, count)

; if division result gt 1.2
; replace all values in imageout with values from image[1] that are 1.2x less
   than image[0]

if count GT 0 then imageout[removefirst] = imagetwo[removefirst]

for i = 0,23,2 do begin

   image1 = image[i]*0.02-273.15
   image2 = image[i+1]*0.02-273.15

        filter[i/2] = image1>image2

create an image1-sized matrix of all NaNs (c)
; find where both image1 and image2 have valid values
; replace corresponding indices in c with the larger of image1 or image2
; all other values remain as NaNs

C = REPLICATE( !VALUES.F_NAN, N_ELEMENTS(image1) )

good = WHERE( FINITE(image1) and FINITE(image2), ngood )
IF ( ngood GT 0 ) THEN C[good] = (image1[good] + image2[good])/2

filter[i/2] = c

end for
end
```