Subject: Re: Add one column based on the data in another column
Posted by Conor on Wed, 22 Aug 2007 12:20:11 GMT
View Forum Message <> Reply to Message

On Aug 22, 5:08 am, DirtyHarry <kim20...@gmail.com> wrote:
> G'day, Everyone!
>
> I have this data file. This was summarized from bigger file for test
> simulation.
>
> aaa.txt
> 01 -999.9
> 02 -999.9
> 03 -999.9
> 04 0.13
> 05 -999.9
> 06 -999.9
> 07 0.17
> 08 -999.9
> 09 -999.9
> 10 -999.9
> 11 -999.9
> 12 -999.9
> 13 -999.9
> 14 -999.9
> 15 -999.9
>
> I am making another column based on 2nd column with these criteria.
> 1) All five lines should have the same value.
> 2) This value is come from 2nd column and should be 0 < value <
> 1.
> 3) If no value is in this range, (i.e. 11~15th lines above), just copy
> the values from 2nd column.
>
> This is my target array.
>
> 01 -999.9 0.13
> 02 -999.9 0.13
> 03 -999.9 0.13
> 04 0.13 0.13
> 05 -999.9 0.13
> 06 -999.9 0.17
> 07 0.17 0.17
> 08 -999.9 0.17
> 09 -999.9 0.17
> 10 -999.9 0.17
> 11 -999.9 -999.9
> 12 -999.9 -999.9

> 13 -999.9 -999.9
> 14 -999.9 -999.9
> 15 -999.9 -999.9
>
> To get this array, I coded as follows. However, something must be
> wrong so far. Please take a look and give me any suggestions. Thanks.
>
> pro albedo_final
> close, /all
> data1 = 'D:\MODIS_ALL\aaa.txt'
> num_data = file_lines(data1)
> albedo_arr = fltarr(2, num_data)
> albedo_fin = fltarr(3, num_data)
> albedo_OK = 0.0
>
> openr, 2, data1
> readf, 2, albedo_arr
> close, 2
> c1 = 0
>
> openw, 1, 'bbb.txt'
> for i= 0, num_data-1 do begin
>
> dd = 5*(c1+1) +1
> if albedo_arr[0, i] lt DD then begin
> if (albedo_arr[1,i] gt 0 and albedo_arr[1,i] lt 1) then
> begin
> albedo_OK = albedo_arr[1,i]
> print, albedo_OK
> endif
> albedo_fin[0:1, i] = albedo_arr[0:1, i]
> albedo_fin[2, i] = albedo_OK
>
> endif
> c1 = c1+1
> endfor
> print, albedo_fin
> ;printf, 1, albedo_fin
> ;close, 1
> print, " It's done!"
> end
>
> This is the last result.
>
> 1.00000 -999.900 0.000000
> 2.00000 -999.900 0.000000
> 3.00000 -999.900 0.000000
> 4.00000 0.130000 0.130000

```
>  5.00000     -999.900     0.130000
>  6.00000     -999.900     0.130000
>  7.00000     0.170000     0.170000
>  8.00000     -999.900     0.170000
>  9.00000     -999.900     0.170000
>  10.0000     -999.900     0.170000
>  11.0000     -999.900     0.170000
>  12.0000     -999.900     0.170000
>  13.0000     -999.900     0.170000
>  14.0000     -999.900     0.170000
>  15.0000     -999.900     0.170000
```

The problem is that your algorith is all wrong. Your program is working fine, it's just doing the wrong thing. What it is doing is that it is always writing out the albedo_ok variable to the third column. Initially, the albedo_ok array has the value '0.0', which is why this is outputted to the first 3 lines. Then, the albedo_ok value changes only when you find a column with good data. When you do find such a value, then the albedo_ok value changes and that is printed out until you find another good value. So for instance if you had this file:

```
1 -999
2 -999
3 0.18
4 -999
5 0.16
6 0.84
7 0.95
```

you would get this output:

```
1 -999 0.00
2 -999 0.00
3 0.18 0.18
4 -999 0.18
5 0.16 0.16
6 0.84 0.84
7 0.95 0.95
```

The problem is that you need to process and output five rows at a time, you can't do them one at a time like you are doing now. What you need to do is read in five lines at one time and compare the values of the second column for all five lines. Then, you either extract the one good value for output or you set the output to be -999.9. Once you've figured out what you want to output, you output all five lines at once and output the same value to the third column for each line.