
Subject: Removing unwanted data from a structure
Posted by [cab581](#) on Fri, 26 Apr 2013 16:34:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I tried searching for this, but nothing seemed to match my question, so sorry if this has been asked before.

I have a structure with data from an experiment that contains error values. These are given as (for example, but always less than -100) -999 or -888. I want to keep the structure in tact for analysis, but I'm looking simply to replace these values with NaN. I tried and failed (see below) to use WHERE, but it doesn't work and I'm told that you can't use IF statements with structures.

My thoughts:

I know with an array I can just use

```
B = WHERE(A LT -100, count)
IF (count GT 0) THEN A[B] = !VALUES.F_NAN
```

but I think that it is probably inefficient (I have a lot of data) to pull out arrays from my structures, then place them all back in to a new structure to continue my work. Is there any way that I can assign the parts of the structure as scalars for the purpose of an IF statement but replace them back as part of a loop?

Any help is appreciated.

Subject: Re: Removing unwanted data from a structure
Posted by [cab581](#) on Wed, 08 May 2013 18:49:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, May 8, 2013 2:47:20 PM UTC-4, cab581 wrote:

> I'm back for more!

>

>

>

> I'm trying to sift through my data to remove unwanted values. Previously I was removing data as it read out of a file, so the code worked fine. Now It's a bit more complex.

>

>

>

> FOR I = 0, 17 DO BEGIN

>

> FOR J = 0, 99 DO BEGIN

>

```

> ind = where(STRUCT.(6)[J] GT MED[i,j], count)
>
> IF count NE 0 THEN STRUCT.(6)[ind] = !values.F_Nan
>
> MEAN[I,J] = MEAN(STRUCT.(6)[J], /double, /NAN)
>
> ENDFOR
>
> ENDFOR
>
>
>
> I have lots of data. I have two dimensions (altitude, J and 10 degree latitude bins, I) within each
data set. I have calculated the median (MED) for each I,J section in the structure. I want run
through every single point in all of my files and remove those that are greater than that median.
The problem with what I have written is that (I think!) it's looking for where every value across the
whole of STRUCT.(6) are greater than the median, not the individual values within STRUCT.(6)
>
>
>
> Thanks in anticipation

```

I should add that the 'files' that I want to compare to the median are in a structure, the same dimensions as that of the median.

Subject: Re: Removing unwanted data from a structure
 Posted by [Jeremy Bailin](#) on Thu, 09 May 2013 23:13:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 5/8/13 1:47 PM, cab581 wrote:

```

> I'm back for more!
>
> I'm trying to sift through my data to remove unwanted values. Previously I was removing data
as it read out of a file, so the code worked fine. Now It's a bit more complex.
>
> FOR I = 0, 17 DO BEGIN
> FOR J = 0, 99 DO BEGIN
> ind = where(STRUCT.(6)[J] GT MED[i,j], count)
> IF count NE 0 THEN STRUCT.(6)[ind] = !values.F_Nan
> MEAN[I,J] = MEAN(STRUCT.(6)[J], /double, /NAN)
> ENDFOR
> ENDFOR
>
> I have lots of data. I have two dimensions (altitude, J and 10 degree latitude bins, I) within each
data set. I have calculated the median (MED) for each I,J section in the structure. I want run
through every single point in all of my files and remove those that are greater than that median.
The problem with what I have written is that (I think!) it's looking for where every value across the

```

whole of STRUCT.(6) are greater than the median, not the individual values within STRUCT.(6)
>
> Thanks in anticipation
>

What are the dimensions of struct.(6)? I can't quite figure out what you're trying to do.

-Jeremy.

Subject: Re: Removing unwanted data from a structure
Posted by [cab581](#) on Fri, 10 May 2013 17:41:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, I'll back track and try to explain more clearly.
I have a structure called STRUCT, it is made from thousands of files, each of these has a measurement at a particular location. The first six tags STRUCT.(0) to STRUCT.(5) are identifiers, STRUCT.(6) is the data. First, I put the data in to 18 latitudinal bins;

```
FOR I = 0,17 DO BEGIN  
A = 0.0  
A = WHERE(STRUCT.LAT GT LAT[I] AND STRUCT.LAT LE LAT[I]+10)
```

STRUCT[A].(6) is therefore the data that I want to analyze in each of the 18 bins.

I found the medians by using;

```
FOR J = 0,99 DO BEGIN  
  
MEDIAN[I,J] = MEDIAN(STRUCT[A].(6)[J], /double, /even)
```

Where I is the latitude (0,17) and J is the altitude (0,99), so I get an array of the median measurement at each of my 18x100 points.

What I want to do is sift through the data from which I calculated the median and then compare those to the median to exclude erroneous values, so I thought that I could use the following;

```
ind = where(STRUCT[A].(6)[J] GT (MEDIAN+1000))  
IF ind[0] GT 0 then STRUCT[A].(6)[ind] = !values.F_Nan
```

but it doesn't work, I end up with NaNs all over the place where they shouldn't be.

Subject: Re: Removing unwanted data from a structure
Posted by [Jeremy Bailin](#) on Fri, 10 May 2013 22:10:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 5/10/13 12:41, cab581 wrote:

```
> OK, I'll back track and try to explain more clearly.
> I have a structure called STRUCT, it is made from thousands of files, each of these has a
measurement at a particular location. The first six tags STRUCT.(0) to STRUCT.(5) are identifiers,
STRUCT.(6) is the data. First, I put the data in to 18 latitudinal bins;
>
> FOR I = 0,17 DO BEGIN
> A = 0.0
> A = WHERE(STRUCT.LAT GT LAT[I] AND STRUCT.LAT LE LAT[I]+10)
>
> STRUCT[A].(6) is therefore the data that I want to analyze in each of the 18 bins.
>
> I found the medians by using;
>
> FOR J = 0,99 DO BEGIN
>
> MEDIAN[I,J] = MEDIAN(STRUCT[A].(6)[J], /double, /even)
>
> Where I is the latitude (0,17) and J is the altitude (0,99), so I get an array of the median
measurement at each of my 18x100 points.
>
> What I want to do is sift through the data from which I calculated the median and then compare
those to the median to exclude erroneous values, so I thought that I could use the following;
>
> ind = where(STRUCT[A].(6)[J] GT (MEDIAN+1000))
> IF ind[0] GT 0 then STRUCT[A].(6)[ind] = !values.F_Nan
>
> but it doesn't work, I end up with NaNs all over the place where they shouldn't be.
>
```

Aha! I think I understand. I was missing the part about binning in latitude.

The three problems I see with the code as you've written it are:

1. You're using the names of language elements ("median", "struct") as variable names. While this is not forbidden and won't cause errors, it is very confusing and will almost certainly lead you into writing buggy code. So I'm going to pretend that your "median" variable is actually called "med".

2. In this comparison:
ind = where(STRUCT[A].(6)[J] GT (MEDIAN+1000))
you want med[i,j].

3. You're indexing the wrong part of the struct here:
IF ind[0] GT 0 then STRUCT[A].(6)[ind] = !values.F_Nan
I think you actually want:
if ind[0] gt 0 then struct[ind].(6)[j] = !values.f_nan

-Jeremy.
