Subject: Re: Help with Histogram...
Posted by peter.albert@gmx.de on Wed, 28 Sep 2005 06:43:14 GMT
View Forum Message <> Reply to Message

Hi,

```
>  index=Where((profile GE base[0] AND profile LE top[0]) OR $
>                  (profile GE base[1] AND profile LE top[1]) OR $
>                  ...
>                  (profile GE base[7] AND profile LE top[7]))
>  profile[index]=1
```

before I get things wrong from the beginning, I'd better ask: "profile"
is the array which should just contain the zeros and ones, depending on
the cloud base and top heights, right? In that case, comparing
"profile" itself to base[i] and top[i] does not make sense. There has
to be another variable actually giving the height levels. And I guess
those are constant for all profiles, are they not?

Regards,

Peter

Subject: Re: Help with Histogram...
Posted by peter.albert@gmx.de on Wed, 28 Sep 2005 08:26:05 GMT
View Forum Message <> Reply to Message

Hi again,

assuming that the above assumption is correct, I could propose
something to start with. It's just a start, as there is still one
for-loop, but you can get rid of the ugly where statements:

let's first create some test data:

```
;I'll use 8 cloud layers, each 300 meters thick:
base = findgen(8) * 1000.
top = base + 300.

;Here is the profile variable:
profile = intarr(60)

; And here come the height levels, just an arbitrary set
; of numbers, unevenly spaced.
; Mind that this array has 61 elements, as you need one upper
; and lower boundary for each layer in "profile"
```

```
height_level = [0., exp((findgen(60)-30) / 15.)] * 1300.

; I'll use interpol for finding the correct indices, so we need
; the abscissa values for "height_levels":

x = indgen(61)

; Now, with interpol, we can find those  layers of "height_level"
; which will be completely covered by each combination
; of cloud base and top value; using ceil and floor
; is equivalent to the "...GE base[0] AND ... LE top[0] ..."
; in your example:

base_idx = ceil(interpol(x, height_level, base))
top_idx = floor(interpol(x, height_level, top))

; Fine so far, but now I still need a for-loop for filling
; "profile":

for i = 0, 7 do $
  if base_idx[i] le top_idx[i] then $
    profile[base_idx[i]:top_idx[i]] = 1
```

Note that the "interpol" way also works if "base" and "top" are not simple 8-element vectors but rather arrays of size(number_of_profiles, 8). In that case, "base_idx" and "top_idx" are of the same size, but here the for-loops becomes even more uglier.

Regards,

Peter

Subject: Re: Help with Histogram...
Posted by peter.albert@gmx.de on Wed, 28 Sep 2005 08:47:46 GMT
View Forum Message <> Reply to Message

Me again :-)

here is a solution for n_profiles at once, which at the end just needs one loop over all profile levels, which is probably acceptible ...

```
; Let's try 10 profiles:
  n = 10

; Arbitrary clouds
```

```
  base = (findgen(8) * 1000.) ## (intarr(n) + 1)
  top = base + 300.

; Let's make one different from the others:
  base[5, *] = base[5, *] * .5
  top[5, *] = top[5, *] * .5

; This is the same as above
  profile = intarr(n, 60)
  height_level = [0., exp((findgen(60)-30) / 15.)] * 1300.
  x = indgen(61)
  base_idx = ceil(interpol(x, height_level, base))
  top_idx = floor(interpol(x, height_level, top))

; Now, this is the loop over the profile levels:
; For each level we check for all profiles at once
; whether level "i" is within the borders as given
; by base_idx and top_idx:

  for i = 0, 59 do $
     profile[*, i] = $
     total($
       base_idx le i and top_idx ge i $
       , 2) $
     gt 0
```

I gave it a quick check and the result looked right to me, but feel
free to tell me if it does not what you expect.

Best regards,

Peter

Subject: Re: Help with Histogram...
Posted by btt on Wed, 28 Sep 2005 12:34:57 GMT
View Forum Message <> Reply to Message

MA wrote:
> Sorry, some people here are surely starting to get sick of Histogram by
> now...
> I've been reading the tutorial a couple of times in the last few weeks,
> and I've actually managed to get rid of a lot of unneccessary loops and
> stuff. I have a problem here, and I'm almost sure there must be a way
> to do it with histogram, but can't figure out how.
> I have an (as yet empty) array (profile=IntArr(60)), each index
> corresponding to a height level (e.g. surface to 20km, unevenly

> spaced), and two arrays (base, top, each FltArr(8)) containing the
> height of eight cloud bases and cloud tops. I want to set the value of
> the levels in 'profile' that fall between a cloud base and a cloud top
> (i.e. are "inside" a cloud) with a 1. Here's an attempt at an
> illustration:
>
> profile value of
> levels profile
> .
> .
> .
> - 0
> - 0
> - ------ top[1]=8000m 1
> - 1
> - 1
> - ------base[1]=7300m 1
> - 0
> - 0
> - 0
> - ------top[0]=500m 1
> - 1
> - ------base[0]=370m 1
> - 0
> - 0
> ----------
> ////////surface////////////////
>
> How can I do that without looping? If I could specify the histogram
> bins by hand, I'd set them to the cloud base and top levels, and let
> histogram sort 'profile' into those bin. At fist I thought the keyword
> 'Locations' would let me do that, but I guess I got that wrong.
> The only thing I can think of is something along the lines of
>
> index=Where((profile GE base[0] AND profile LE top[0]) OR $
> (profile GE base[1] AND profile LE top[1]) OR $
> ...
> (profile GE base[7] AND profile LE top[7]))
> profile[index]=1
>
> Any ideas? As an aside, I got a couple of thousand profiles like that,
> and 'profile' is really an array of (number_of_profiles x 60). If
> there's any way to do this problem without looping over the individual
> profiles, that would be even better.
>
> Thanks!
>

Hi,

Would the follwoing work? It mixes the base and top values - then searched for each level within using VALUE_LOCATE. A binary flag is used to mark the 0/1 of the profile. I think it should be pretty fast.

Cheers,
ben

```
****START
PRO cloudlevel
n = 8
top = findgen(n) * 100.0 + 30.0
base = top - 28.0

all = [base, top]
s= SORT(all)
all = all[s]

bProfile = BytArr(n*2)
bProfile[0:*:2] = 1B

nLevel = 10
Level = findgen(nLevel)/(nLevel-1) * (MAX(top)-1)

index = VALUE_LOCATE(all, level)

iProfile = bProfile[index]


plot, top, psym = 6, /noclip
oplot, base, psym = 5, /noclip

for i = 0, nLevel-1 Do Plots, [0,8], [level[i], level[i]], linestyle =
2, thick = 2
for i = 0, nLevel-1 Do XYOUTS, 7.5, level[i] + 10, STRTRIM(i,2), /DATA

for i = 0, nLevel-1 Do $
 if iProfile[i] EQ 1 Then XYOUTS, 7.6, level[i]+10, '*' , /DATA


end

***FINI
```

Subject: Re: Help with Histogram...

Posted by MA on Wed, 28 Sep 2005 16:44:42 GMT
View Forum Message <> Reply to Message

Wow! Thanks for all the responses. I just got all your messages, give me some time to check them out. BTW, you are right, of course, and in addition to 'profile' I have another array that holds the actual level heights corresponding to the 60 indices.
Peter, I didn't even know floor and ceil existed.... learn something new everyday.
I'll let you know how your tips work.
Thanks!

Subject: Re: Help with Histogram...
Posted by MA on Wed, 28 Sep 2005 18:54:59 GMT
View Forum Message <> Reply to Message

Hello Peter,
I've been playing around with your second suggestion for a while. There are a couple of added difficulties with my data.
For one, the actual height levels (corresponding to 'height_level' in your example) vary from profile to profile. The other problem is that the arrays containing the cloud bases and tops are not always full (i.e. there are not always 8 stacked clouds). Most of the time, only the first 1-3 places in the array actually contain data, the rest is set to a default value. I have another array (length=number of profiles) that holds the number of clouds in each profile.
On the first problem:
I have to alter the line from your example
  base_idx = ceil(interpol(x, height_level, base))
; in dimensions: [10,8]=ceil(interpol([60],[60],[10,8]))
to dimensions
[10,8]=ceil(interpol([60,10],[60,10],[10,8]))
x and height_level are now 60x10 arrays, to account for the changing height levels in different profiles.
BUT, the resulting array from this line (base_idx, top_idx) doesn't give me the right answer.
I think it is because it probably does the interpolation now in both dimensions. Is there a way to tell Interpol to only interpolate in the one direction (between height levels, but not between profiles)? I haven't found such an option on the help, but maybe I overlooked it.
Otherwise I'm back to looping over the profiles:
For i=0,num_profiles-1 do $
    base_idx[*,i] = ceil(interpol(x[*,i], height_level[*,i], base[i,*]))

The second problem (fill values) is probably not as bad.
I'll look a bit into Ben's suggestion, see if that one does do better

with the varying profiles.
Thanks again.

Subject: Re: Help with Histogram...
Posted by peter.albert@gmx.de on Thu, 29 Sep 2005 10:31:41 GMT
View Forum Message <> Reply to Message

Good morning,

there is always something new for me too; I did not know about VALUE_LOCATE, for instance. Replacing "ceil(interpol(x, height_level, base)) " by "(value_locate(height_level, base)-1)> 0" saves you a factor of 10! But you can drop that, anyway, as I'd argue that Ben's approach is better. It took me a while to get the point, however. But looking the other way round, i.e. not to check each cloud layer but instead checking each height level is quite appealing. And faster.

However, having different height levels with each profile *is* a problem. You can't concatenate the different levels to one large array, as the first array for value_locate must be monotonically increasing.

Regards,

Peter

Subject: Re: Help with Histogram...
Posted by MA on Thu, 29 Sep 2005 18:16:40 GMT
View Forum Message <> Reply to Message

Hello Peter,
I've decided to keep the one loop (over the number of profiles) in the program, and use Ben's idea for the rest. It works quite nice, and I can use the Value_Locate the other way around to find clouds that are so thin that both top and base fall between model levels. Nifty.
Thanks for all your help.
