Subject: Re: All twisted up over reverse indices!
Posted by penteado on Mon, 19 Dec 2016 20:45:36 GMT
View Forum Message <> Reply to Message

In case anyone is wondering, the reverse indices can be converted into lists or hashes, which make their use much simpler, with

```
 h=histogram(array,_strict_extra=_ex,reverse_indices=ri,locat ions=locations)
if arg_present(rl) then begin
  rl=list(length=n_elements(h))
  foreach el,h,i do if (el gt 0L) then rl[i]=ri[ri[i]:ri[i+1]-1]
endif
if arg_present(rh) then begin
  rh=!version.release ge '8.3' ? orderedhash(locations) : hash(locations)
  foreach el,h,i do if (el gt 0L) then rh[locations[i]]=ri[ri[i]:ri[i+1]-1]
endif
```

This is used in my histogram wrapper, histogram_pp
(http://www.ppenteado.net/idl/pp_lib/doc/histogram_pp.html):

```
 vals=randomu(0L,15)*13d0
  h=histogram_pp(vals,reverse_indices=ri,reverse_list=rl,locat ions=loc,reverse_hash=rh)
 print,h
 ;1           3           0           4           2           1           0           4
 print,ri[ri[0]:ri[1]-1]
 ;13
 foreach el,rl,i do print,i,' : ',el
 ;0 :           13
 ;1 :            8          11          12
 ;2 : !NULL
 ;3 :            0           1           4           6
 ;4 :            9          10
 ;5 :            2
 ;6 : !NULL
 ;7 :            3           5           7          14
 foreach el,loc do print,el,' : ',rh[el]
 ;3.8679500 :           13
 ;4.8679500 :            8          11          12
 ;5.8679500 : !NULL
 ;6.8679500 :            0           1           4           6
 ;7.8679500 :            9          10
 ;8.8679500 :            2
 ;9.8679500 : !NULL
 ;10.867950 :            3           5           7          14
```

On Monday, December 19, 2016 at 9:37:46 AM UTC-8, Stephen Williams wrote:
>>
>> I found this helped (scroll down to the section on reverse indices):

>> http://www.idlcoyote.com/tips/histogram_tutorial.html
>
> For completeness, I figured out the problem and determined how to extract what I needed. My issue came primarily from improper referencing of the arrays. For example, the following line gave the incorrect indices when used with multiple matches:
>
> ra_i_m(i) = ra_i(match1[match1[part_two(i)]:match1[part_two(i+1)]-1])
>
> The problem is in the end, and should be this:
>
> ra_i_m(i) = ra_i(match1[match1[part_two(i)]:match1[part_two(i)+1]-1])
>
> The parentheses are critical here, as I want the next array element in "match1" and not the next array element in "part_two".
>
> Distances were obtained similarly via the following line:
>
> d(match1[match1[part_thr(i)]:match1[part_thr(i)+1]-1])
>
> That is to say, the above gives the distances for each element in the sub-array denoted by the ":".
>
> I hope this may help someone, I am sure relieved that my problem has been solved.