Subject: Re: How to Sort/Uniq a list and keep its original index
Posted by David Fanning on Wed, 11 Oct 2006 23:12:07 GMT
View Forum Message <> Reply to Message

Dilkushi@gmail.com writes:

> I have to sort a file with 650,000 records in search of duplicate
> records.. and I need a list of duplicates (not a list without
> duplicates)...
> indexS=sort(testTotal)
> test=testTotal[indexS]
> indexU=uniq(test)
>
> indexU is an index with no duplicates..
> how do I get an index pertaining to the duplicates only?..

I haven't tested this, but just off the top of my
head:

```
  I = Where(Histogram(indexU, Min=0, Max=N_Elements(testTotal)) $
     EQ 0, count)
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to Sort/Uniq a list and keep its original index
Posted by Jean H. on Thu, 12 Oct 2006 16:19:01 GMT
View Forum Message <> Reply to Message

without the histogram, you could try:

```
tmp = lindgen(650000)
tmp[indexU] = -1
duplicate = tmp[where tmp ne -1)]
```

Jean

Dilkushi@gmail.com wrote:
> Dear all
> I have to sort a file with 650,000 records in search of duplicate

> records.. and I need a list of duplicates (not a list without
> duplicates)...
> indexS=sort(testTotal)
> test=testTotal[indexS]
> indexU=uniq(test)
>
> indexU is an index with no duplicates..
> how do I get an index pertaining to the duplicates only?..
> please help..
> thanks in advance
> dilkushi
>

Subject: Re: How to Sort/Uniq a list and keep its original index
Posted by JD Smith on Thu, 12 Oct 2006 18:16:13 GMT
View Forum Message <> Reply to Message

On Wed, 11 Oct 2006 17:12:07 -0600, David Fanning wrote:
>
> I haven't tested this, but just off the top of my head:
>
> I = Where(Histogram(indexU, Min=0, Max=N_Elements(testTotal)) $
> EQ 0, count)

I think that will leave out one of the duplicates of each set (since
one of them by definition is unique).

If you're going to use HISTOGRAM, you could use it to do the whole
thing:

```
h=histogram(testTotal,REVERSE_INDICES=ri)
wh=where(h gt 1,cnt) ;; bins with duplicates
for i=0,cnt-1 do do_something_with,ri[ri[wh[i]]:ri[wh[i]+1]-1]
```

since it's faster than SORT for well-behaved data. Notice that I didn't
explicitly test for empty bins, since I'm only looping over those bins
with 2 or more entries. If most of your duplicate counts are low (2x, 3x,
etc.), you can see another big speedup by binning the resulting histogram.
Standard sparse data warnings apply.

If you want to use SORT anyway (for simplicity, or for instance
because the data could be very sparse), your could just do the
opposite of what UNIQ does:

```
indexDUP=where((test eq shift(test,-1)) OR (test eq shift(test,1)))
```

JD

Subject: Re: How to Sort/Uniq a list and keep its original index
Posted by Dilkushi@gmail.com on Wed, 18 Oct 2006 18:36:31 GMT
View Forum Message <> Reply to Message

Thank you JD
this is waht i was looking for... perfect...
dilkushi

JD Smith wrote:

> On Wed, 11 Oct 2006 17:12:07 -0600, David Fanning wrote:
>>
>> I haven't tested this, but just off the top of my head:
>>
>> I = Where(Histogram(indexU, Min=0, Max=N_Elements(testTotal)) $
>> EQ 0, count)
>
> I think that will leave out one of the duplicates of each set (since
> one of them by definition is unique).
>
> If you're going to use HISTOGRAM, you could use it to do the whole
> thing:
>
> h=histogram(testTotal,REVERSE_INDICES=ri)
> wh=where(h gt 1,cnt) ;; bins with duplicates
> for i=0,cnt-1 do do_something_with,ri[ri[wh[i]]:ri[wh[i]+1]-1]
>
> since it's faster than SORT for well-behaved data. Notice that I didn't
> explicitly test for empty bins, since I'm only looping over those bins
> with 2 or more entries. If most of your duplicate counts are low (2x, 3x,
> etc.), you can see another big speedup by binning the resulting histogram.
> Standard sparse data warnings apply.
>
> If you want to use SORT anyway (for simplicity, or for instance
> because the data could be very sparse), your could just do the
> opposite of what UNIQ does:
>
> indexDUP=where((test eq shift(test,-1)) OR (test eq shift(test,1)))
>
> JD