
Subject: Efficiently perform histogram reverse indices like procedure on a string array?

Posted by [Matt Francis](#) on Wed, 25 Jul 2012 23:39:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have an array of a data structure, one tag of which is a string identifier indicating which location the data belongs to. There are many thousands of data points, but only about a dozen or so unique locations.

I make frequent use of the HISTOGRAM function with the reverse_indices in order to carve up data by some identifier, most commonly the time. In this case, I want to divide out the data by site efficiently. I can't use HISTOGRAM on strings, so I need some other approach. There are plenty of ways this can be done, but I'd like some views on the better and most efficient ways to do it.

Take an example, say we have a simple string array

```
foo=['a','b','c','b','b','a','a','c']
```

To determine the list of unique strings we could do

```
sfoo = foo[sort(foo)]  
print,sfoo[uniq(sfoo)]
```

We can then repeatedly use WHERE to find the indices in the data array(s) corresponding to each site.

Is there a quicker/better way to do this? Repeatedly calling WHERE seems inefficient (certainly HISTOGRAM is way faster when it is usable)
