
Subject: Re: cool way to determine durations in time series
Posted by news.qwest.net on Fri, 20 Jan 2006 19:01:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Thomas Pfaff" <yael@gmx.net> wrote in message
news:43com4F1mh6c1U1@individual.net...
> Hello everyone,
>
> I'm doing some IDL-abuse in hydrology, so my question might seem a bit
> odd, maybe.
>
> A task that is occurring quite regularly is to determine the duration of
> certain events. For example "What is the longest contiguous duration of
> stream flow below/above a certain discharge"
>
> Assuming I have an equidistant time series (e.g. one value each day)
> this basically reduces to the question of how can I transform an array
> like this
>
> series = [1,1,0,0,0,0,1,0,1,1,1,0,0,1,1]
>
> into something like this
>
> durations = [2,1,3,2]
>
> which is I want to count all contiguous fields of '1's in an array.
>
> Somehow my brain wants to use HISTOGRAM for this, but I just can't see
> how to do it.
> At the moment I'm helping myself by using CONVOL(to highlight the edges)
> and WHERE(to get the differences between two adjacent edge indices) but
> as the data gets more, this becomes extremely tedious as well as memory
> consuming (see the example below). Besides, CONVOL wouldn't work if a
> series started or ended with '1's as it can't correctly apply the kernel
> to those elements.
>
> Any ideas? Anyone who has seen this problem in one of his/her maths
> textbooks? Hints to literature are also highly appreciated.
>
> Thanks in advance,
>
>
> Thomas

Hi Thomas,
here is a starting point for you.
Assuming series contains integers, this piece of code gives you the

durations of
each repeated value EXCEPT for the very end of the series. Should be easy
enough to
patch that up. Also, should be easy enough to remove the series of zeros,
and explicitly
make it do a specific number (like 1 in your example).
Also, one can apply this to your "above/below" by first applying "series =
data > threshold"
which is what I am assuming you are doing above to get your "series".

Anyways, here it is (and it is just a starting point suggestion, you will
have to
modify it a bit)

```
series = [1,1,0,0,0,0,1,0,1,1,1,0,0,1,1]
d = series - shift(series,1)
w = where(d ne 0)
newdur = w - shift(w,1)
newdur[0] = w[0]
print,newdur
```

```
> newdur = 2 4 1 1 3 2
```

Note, it is missing that last "2" points, so you have to patch that.

Cheers,
bob
