Subject: SMOOTH or TS_SMOOTH
Posted by markjamie on Tue, 20 Nov 2012 23:36:39 GMT
View Forum Message <> Reply to Message

I'm looking to calculate moving / windowed averages for a given time series using the last N values to calculate the mean.

Am I right in thinking ts_smooth with the /backward keyword is the right way of doing this? The documentation for ts_smooth doesn't give any formulae in the same way it does for smooth.

Any advice would be greatly appreciated. Many thanks

Subject: Re: SMOOTH or TS_SMOOTH
Posted by lecacheux.alain on Wed, 21 Nov 2012 10:38:52 GMT
View Forum Message <> Reply to Message

Le mercredi 21 novembre 2012 00:36:39 UTC+1, mark...@gmail.com a écrit :
> I'm looking to calculate moving / windowed averages for a given time series using the last N values to calculate the mean.
>
>
>
> Am I right in thinking ts_smooth with the /backward keyword is the right way of doing this? The documentation for ts_smooth doesn't give any formulae in the same way it does for smooth.
>
>
>
> Any advice would be greatly appreciated. Many thanks

TS_SMOOTH(/BACKWARD) uses a N-th order autoregressive model for calculating the (moving) average of the past N points, i.e. doing a weighted average in which weights are computed such that they minimize the difference with the present point. This is usually different from the simple average of the past N-points. It depends on what kind of "moving average" you actually want.
alx.

Subject: Re: SMOOTH or TS_SMOOTH
Posted by markjamie on Wed, 21 Nov 2012 13:01:16 GMT
View Forum Message <> Reply to Message

Thanks for your reply.

I've been playing with TS_SMOOTH today and I don't think it's quite what I'm after. As you mention it calculates the average based on a forward or backward forecast of the data. This gives a similar answer to using a mean of the last N-values but not exactly the same.

I was hoping there would be an IDL function to give the average of the last N points in a data series. In essence, a boxcar smoothing where the window is not centered.

For example:

```
N = 3
arrayData = dindgen(X)
arrayBackwardMovingMean = dblarr(X)
for i = 0, X do arrayBackwardMovingMean[i] = mean(arrayData[min([0, i - N] : i)])
```

The min function is to avoid using negative indices and incurring an array out of bounds error.

The code above is ok for what I'm after but it didn't feel very 'IDL'.

Mark

Subject: Re: SMOOTH or TS_SMOOTH
Posted by markjamie on Wed, 21 Nov 2012 13:09:46 GMT
View Forum Message <> Reply to Message

Typing code too quick....corrected version

```
X = 50
N = 3
arrayData = dindgen(X)
arrayBackwardMovingMean = dblarr(X)
for i = 0, X - 1 do arrayBackwardMovingMean[i] = mean(arrayData[min([0, i - N] : i)])
```

Subject: Re: SMOOTH or TS_SMOOTH
Posted by lecacheux.alain on Wed, 21 Nov 2012 14:28:29 GMT
View Forum Message <> Reply to Message

Le mercredi 21 novembre 2012 14:09:46 UTC+1, mark...@gmail.com a écrit :
> Typing code too quick....corrected version
>
>
>
> X = 50
>
> N = 3
>
> arrayData = dindgen(X)
>
> arrayBackwardMovingMean = dblarr(X)
>

> for i = 0, X - 1 do arrayBackwardMovingMean[i] = mean(arrayData[min([0, i - N] : i)])

A more "IDL" way (untested) might be :

```
pad = !Values.F_NAN + dindgen(N)
print,  mean(reform([pad,reform(rebin([arrayData,pad],X+N,N),(X*N)*N )],X+N+1,N),DIM=2,/NAN)
```

or, more explicitely:

```
print,mean( $
  reform( $
  [pad,reform(rebin([arrayData,pad],X+N,N),(X*N)*N)] $  ;replicate and pad your array N times
  ,X+N+1,N), $         ;do the shift
  DIM=2, $             ;do the average
  /NAN)               ;handle the bounds
```

alx.
