NDK_TESMTH

Last Modified on 07/07/2016 11:45 am CDT

- C/C++
- .Net

```
int __stdcall NDK_TESMTH(double * pData,
                         size_t   nSize,
                         BOOL     bAscending,
                         double * alpha,
                         double * beta,
                         double * gamma,
                         int      L,
                         int      nHorizon,
                         BOOL     bOptimize,
                         double * retVal
                        )
```

Returns the (Winters's) triple exponential smoothing estimate of the value X at time T+m.

Returns

status code of the operation

Return values

NDK_SUCCESS Operation successful

NDK_FAILED Operation unsuccessful. See **Macros** for full list.

Parameters

[in] **pData** is the univariate time series data (a one dimensional array).

[in] **nSize** is the number of elements in pData.

[in] **bAscending** is the time order in the data series (i.e. the first data point's corresponding date (earliest date=1 (default), latest date=0)).

[in] **alpha** is the data smoothing factor (alpha should be between zero and one (exclusive)).

[in] **beta** is the trend smoothing factor (beta should be between zero and one (exclusive)).

[in] **gamma** is the seasonal change smoothing factor (Gamma should be between zero and one (exclusive)).

[in] **L** is the season length.

[in] **nHorizon** is the forecast time horizon beyond the end of pData. If missing, a default value of 0 (latest or end of pData) is assumed.

[in] **bOptimize** is a flag (True/False) for searching and using optimal value of the smoothing factor. If missing or omitted, optimize is assumed false.

[out] **retVal** is the calculated value of this function.

Remarks

1. The triple exponential smoothing function F_{T+m} is defined as follows: $$S_1=x_1$$ $$b_1=\frac{1}{L}(\frac{x_{L+1}-x_1}{L}+\frac{x_{L+2}-x_2}{L}+\frac{x_{L+3}-x_3}{L}+...+\frac{x_{L+L}-x_L}{L})$$ $$S_{t>1}=\alpha \times \frac{x_t}{c_{t-L}}+(1-\alpha)(S_{t-1}+b_{t-1})$$ $$b_{t>1}=\beta\times (s_t-s_{t-1})+(1-\beta)b_{t-1}$$ $$c_{t>1}= \gamma \times \frac{x_t}{c_t}+(1-\gamma)c_{t-L}$$ $$F_{t+m}=(s_t+m\times b_t)c_{t-L + (m-1)\pmod L} $$ Where:
   - X_t is the value of the time series at time t
   - T is the time of the latest observation in the sample data
   - α is the smoothing factor
   - β is the trend smoothing factor
   - γ is the seasonal change smoothing factor
   - F_{T+m} is the output of the algorithm at m steps past the end of the sample
2. To search for the optimal values of the smoothing factors (alpha, beta and gamma), the number of non-missing observations should be greater than on seasonal length (L).
3. The time series is homogeneous or equally spaced.
4. The time series may include missing values (NaN) at either end.

Requirements

Header	SFSDK.H
Library	SFSDK.LIB
DLL	SFSDK.DLL

Examples

```
int NDK_TESMTH(double[]  pData,
               int       nSize,
               BOOL      bAscending,
               ref double alpha,
               ref double beta,
               ref double gamma,
               int       seasonLength,
               int       nHorizon,
               BOOL      bOptimize,
               ref double retVal
```

Namespace: NumXLAPI
Class: SFSDK
Scope: Public
Lifetime: Static

)

Returns the (Winters's) triple exponential smoothing estimate of the value of X at time T+m.

Returns

status code of the operation

Return values

NDK_SUCCESS Operation successful
NDK_FAILED Operation unsuccessful. See **Macros** for full list.

Parameters

[in]	**pData**	is the univariate time series data (a one dimensional array).
[in]	**nSize**	is the number of elements in pData.
[in]	**bAscending**	is the time order in the data series (i.e. the first data point's corresponding date (earliest date=1 (default), latest date=0)).
[in]	**alpha**	is the data smoothing factor (alpha should be between zero and one (exclusive)).
[in]	**beta**	is the trend smoothing factor (beta should be between zero and one (exclusive)).
[in]	**gamma**	is the seasonal change smoothing factor (Gamma should be between zero and one (exclusive)).
[in]	**seasonLength**	is the season length.
[in]	**nHorizon**	is the forecast time horizon beyond the end of pData. If missing, a default value of 0 (latest or end of pData) is assumed.
[in]	**bOptimize**	is a flag (True/False) for searching and using optimal value of the smoothing factor. If missing or omitted, optimize is assumed false.
[out]	**retVal**	is the calculated value of this function.

Remarks

1. The triple exponential smoothing function F_{T+m} is defined as follows: $$S_1=x_1$$ $$b_1=\frac{1}{L}(\frac{x_{L+1}-x_1}{L}+\frac{x_{L+2}-x_2}{L}+\frac{x_{L+3}-x_3}{L}+...+\frac{x_{L+L}-x_L}{L})$$ $$S_{t>1}=\alpha \times \frac{x_t}{c_{t-L}}+(1-\alpha)(S_{t-1}+b_{t-1})$$ $$b_{t>1}=\beta\times (s_t-s_{t-1})+(1-\beta)b_{t-1}$$ $$c_{t>1}= \gamma \times \frac{x_t}{c_t}+(1-\gamma)c_{t-L}$$ $$F_{t+m}=(s_t+m\times b_t)c_{t-L + (m-1)\pmod L} $$ Where:
   - X_t is the value of the time series at time t
   - T is the time of the latest observation in the sample data
   - α is the smoothing factor
   - β is the trend smoothing factor
   - γ is the seasonal change smoothing factor
   - F_{T+m} is the output of the algorithm at m steps past the end of the sample
2. To search for the optimal values of the smoothing factors (alpha, beta and gamma), the number of non-missing observations should be greater than on seasonal length (L).
3. The time series is homogeneous or equally spaced.

4. The time series may include missing values (NaN) at either end.

Exceptions

Exception Type	Condition
None	N/A

Requirements

Namespace	NumXLAPI
Class	SFSDK
Scope	Public
Lifetime	Static
Package	NumXLAPI.DLL

Examples

References

Hamilton, J .D.; Time Series Analysis , Princeton University Press (1994), ISBN 0-691-04289-6

Tsay, Ruey S.; Analysis of Financial Time Series John Wiley & SONS. (2005), ISBN 0-471-690740

See Also

[template("related")]