NDK_ACFCI

Last Modified on 06/24/2016 10:54 am CDT

- C/C++
- .Net

```
int __stdcall NDK_ACFCI(double * X,
                        size_t   N,
                        size_t   K,
                        double   alpha,
                        double * ULCI,
                        double * LLCI
                       )
```

Calculates the confidence interval limits (upper/lower) for the autocorrelation function.

Returns

status code of the operation

Return values

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

Parameters

[in]	**X**	is the univariate time series data (a one dimensional array).
[in]	**N**	is the number of observations in X.
[in]	**K**	is the lag order (e.g. k=0 (no lag), k=1 (1st lag), etc.).
[in]	**alpha**	is the statistical significance level. If missing, a default of 5% is assumed.
[out]	**ULCI**	is the upper limit value of the confidence interval
[out]	**LLCI**	is the lower limit value of the confidence interval.

Remarks

1. The time series is homogeneous or equally spaced.
2. The time series may include missing values (NaN) at either end.
3. The lag order (k) must be less than the time series size, or else an error value (**NDK_FAILED**) is returned.
4. The ACFCI function calculates the confidence limits as:

- $\hat\rho_k - Z_{\alpha/2}\times \sigma_{\rho_k} \leq \rho_k \leq \hat\rho_k+ Z_{\alpha/2}\times \sigma_{\rho_k}$
  />, where:
  - ρ_k is the population autocorrelation function.
  - σ_{ρ_k} is the standard error of the sample autocorrelation.
  - $\hat{\rho_{k}}$ is the sample autocorrelation function for lag k.

- $Z\sim N(0,1)$
- $P(\left|Z\right|\geq Z_{\alpha/2}) = \alpha$

5. For the case in which the underlying population distribution is normal, the sample autocorrelation also has a normal distribution:

- $\hat \rho_k \sim N(\rho_k,\sigma_{\rho_k}^2)$, where:
  - $\hat \rho_k$ is the sample autocorrelation for lag k.
  - ρ_k is the population autocorrelation for lag k.
  - σ_{ρ_k} is the standard error of the sample autocorrelation for lag k.

6. Bartlett proved that the variance of the sample autocorrelation of a stationary normal stochastic process (i.e. independent, identically normal distributed errors) can be formulated as:

- $\sigma_{\rho_k}^2 = \frac{\sum_{j=-\infty}^{\infty}\rho_j^2+\rho_{j+k}\rho_{j-k}-4\rho_j\rho_k\rho_{i-k}+2\rho_j^2\rho_k^2}{T}$

7. Furthermore, the variance of the sample autocorrelation is reformulated:

- $\sigma_{\rho_k}^2 = \frac{1+\sum_{j=1}^{k-1}\hat\rho_j^2}{T}$, where:
  - σ_{ρ_k} is the standard error of the sample autocorrelation for lag k.
  - T is the sample data size.
  - $\hat\rho_j$ is the sample autocorrelation function for lag j.
  - k is the lag order.

Requirements

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

Example

```
#include "SFMacros.h"
#include "SFSDK.h"

// Input time series: 110 observation
double data[110]={0.23, 0.24, 0.45, ..., 0.95}

int nRet = NDK_FAILED;
double alpha = 0.05f;
double UL = -2.0f;
double LL = -2.0f;

nRet = NDK_ACFCI(data, 110, 1, alpha, &UL,  &LL);
if( nRet < NDK_SUCCESS){
    // Error occured
    // Call NDK_MSG to retrieve description of the error, and write it to th
e log file
    ....
}
```

int NDK_ACFCI(double[] pData, UIntPtr nSize, int nLag, double alpha, out double retUpper, out double retLower)

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

Calculates the confidence interval limits (upper/lower) for the autocorrelation function.

Return Value

a value from **NDK_RETCODE** enumeration for the status of the call.

NDK_SUCCESS	operation successful
Error	Error Code

Parameters

[in]	**pData**	is the univariate time series data (a one dimensional array).
[in]	**nSize**	is the number of observations in pData.
[in]	**nLag**	is the lag order (e.g. nLag=0 (no lag), nLag=1 (1st lag), etc.).
[in]	**alpha**	is the statistical significance level. If missing, a default of 5% is assumed.

[out] **retUpper** is the upper limit value of the confidence interval
[out] **retLower** is the lower limit value of the confidence interval.

Remarks

1. The time series is homogeneous or equally spaced.
2. The time series may include missing values (NaN) at either end.
3. The lag order (nLag) must be less than the time series size, or else an error value (**NDK_FAILED**) is returned.
4. The ACFCI function calculates the confidence limits as:

- $\hat\rho_k - Z_{\alpha/2}\times \sigma_{\rho_k} \leq \rho_k \leq \hat\rho_k+ Z_{\alpha/2}\times \sigma_{\rho_k}$
  />, where:
  - rho_k is the population autocorrelation function.
  - σ_{ρ_k} is the standard error of the sample autocorrelation.
  - $\hat{\rho_{k}}$ is the sample autocorrelation function for nlag.
  - $Z\sim N(0,1)$
  - $P(\left|Z\right|\geq Z_{\alpha/2}) = \alpha$

5. For the case in which the underlying population distribution is normal, the sample autocorrelation also has a normal distribution:

- $\hat \rho_k \sim N(\rho_k,\sigma_{\rho_k}^2)$, where:
  - $\hat \rho_k$ is the sample autocorrelation for lag k.
  - ρ_k is the population autocorrelation for lag k.
  - σ_{ρ_k} is the standard error of the sample autocorrelation for lag k.

6. Bartlett proved that the variance of the sample autocorrelation of a stationary normal stochastic process (i.e. independent, identically normal distributed errors) can be formulated as:

- $\sigma_{\rho_k}^2 = \frac{\sum_{j=-\infty}^{\infty}\rho_j^2+\rho_{j+k}\rho_{j-k}-4\rho_j\rho_k\rho_{i-k}+2\rho_j^2\rho_k^2}{T}$

7. Furthermore, the variance of the sample autocorrelation is reformulated:

- $\sigma_{\rho_k}^2 = \frac{1+\sum_{j=1}^{k-1}\hat\rho_j^2}{T}$, where:
  - σ_{ρ_k} is the standard error of the sample autocorrelation for lag k.
  - T is the sample data size.
  - $\hat\rho_j$ is the sample autocorrelation function for lag j.
  - k is the lag order.

Exceptions

Exception Type	Condition
None	N/A

Requirements

Header	SFSDK.H
Library	SFSDK.LIB
DLL	SFSDK.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")]