NDK_ADFTEST

Last Modified on 01/09/2017 9:33 pm CST

- C/C++
- .Net

```
int __stdcall NDK_ADFTEST(double * X,
                          size_t   N,
                          WORD     K,
                          WORD     options,
                          BOOL     testDown,
                          double   alpha,
                          WORD     method,
                          WORD     retType,
                          double * retVal
                         )
```

Returns the p-value of the Augmented Dickey-Fuller (ADF) test, which tests for a unit root in the time series sample.

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 length of the autoregressive process. If missing, an initial value equal to the cubic root of the input data size is used.
[in]	**options**	is the model description flag for the Dickey-Fuller test variant (1=no constant, 2=contant-only, 3=trend only, 4=constant and trend, 5=const, trend and trend squared).
[in]	**testDown**	is the mode of testing. If set to TRUE (default), ADFTest performs a series of tests. The test starts with the input length lag, but the actual length lag order used is obtained by testing down.
[in]	**alpha**	is the statistical significance level. If missing, a default of 5% is assumed.
[in]	**method**	is the statistical test to perform (1=ADF).
[in]	**retType**	is a switch to select the return output:

Method	Value	Description
TEST_PVALUE	1	P-Value
TEST_SCORE	2	Test statistics (aka score)
TEST_CRITICALVALUE	3	Critical value.

[out] **retVal** is the calculated test statistics.

Method	Value	Description

Remarks

1. The testing procedure for the ADF test is applied to the following model: $$\Delta y_t = \alpha + \beta_1 t + \beta_2 t^2 + \gamma y_{t-1} + \phi_1 \Delta y_{t-1} + \cdots + \phi_{p-1} \Delta y_{t-p+1} + \varepsilon_t$$
Where:

- Δ is the first different operator
- α is a constant
- β_1 is the coefficient on a time trend
- β_2 is the coefficient on a squared time trend

2. This model can be estimated, and testing for a unit root is equivalent to testing that $\gamma = 0$.
3. In sum, the Augmented Dickey-Fuller Test in Excel test hypothesis is as follows: $$H_{o}: \gamma = 0$$ $$H_{1}: \gamma < 0$$
Where:

- H_{o} is the null hypothesis (i.e. y_t has a unit-root)
- H_{1} is the alternate hypothesis (i.e. ${y_t}$ does not have a unit-root)

4. The test statistics (τ) value is calculated as follows: $$\tau = \frac{\hat{\gamma}}{\sigma_{\hat\gamma}}$$
where:

- $\hat{\gamma}$ is the estimated coefficient
- $\sigma_{\hat\gamma}$ is the standard error in the coefficient estimate

5. The test statistics value (τ) is compared to the relevant critical value for the Dickey-Fuller Test. If the test statistic is less than the critical value, we reject the null hypothesis and conclude that no unit-root is present.
6. The ADFTest does not directly test for stationarity, but indirectly through the existence (or absence) of a unit-root. Furthermore, Augmented Dickey-Fuller Test in Excel incorporates a deterministic trend (and trend squared), so it allows a trend-stationary process to occur.
7. The main difference between the ADFTest and a normal Dickey Fuller test is that ADFTest allows for higher-order autoregressive processes.
8. For the test-down approach, we start with a given maximum lag length and test down by running several tests; in each, we exaimine the high-order coefficients t-stat for significance.
9. It is not possible to use a standard t-distribution to provide critical values for this test. Therefore this test statistic (i.e. τ) has a specific distribution simply known as the Dickey's-Fuller table.
10. The time series must have at least 10, and no more than 10,000 non-missing observations.
11. The time series is homogeneous or equally spaced.
12. The time series may include missing values (e.g. NaN) at either end.

Requirements

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

Examples

```
// (optional) NaN : quiet NaN (Not-A-Number) value of type double (initialization
)
const double NAN = std::numeric_limits::quiet_NaN();
....
double data[100] = {-2.213600965,0.205653805,0.536560947,...};
WORD maxOrder=5;
double alpha = 0.05;
WORD method=1;


double fValue = NAN;

// Scenario: No deterministic component
nRet = NDK_ADFTEST(
        data,   // is the univariate time series data (a one dimensional array)
        100,   // is the number of observations
        maxOrder,         // is the lag length of the autoregressive process.
        ADFTEST_DRIFT_ONLY, // Model 1: A stochastic drift
        TRUE,   // is the mode of testing
        alpha,   // is the statistical significance level
        1,   // is the statistical test to perform (1=ADF).
        TEST_PVALUE,       // is a switch to select the return output
        &fValue           // is the calculated test statistics
                  );
if( nRet >= NDK_SUCCESS)
{
  double fScore = NAN;
  double fCriticalVal = NAN;

  NDK_ADFTEST(data, 100, maxOrder, ADFTEST_DRIFT_ONLY, TRUE, alpha, 1, TEST_SCORE
, &fValue);
  NDK_ADFTEST(data, 100, maxOrder, ADFTEST_DRIFT_ONLY, TRUE, alpha, 1, TEST_CRITI
```

```
CALVALUE, &fValue);
}

// Scenario 2: A deterministic constant and stochastic drift
fValue = NAN;
nRet = NDK_ADFTEST(
        data,   // is the univariate time series data (a one dimensional array)
        100,   // is the number of observations
        maxOrder,         // is the lag length of the autoregressive process.
        ADFTEST_DRIFT_N_CONST, // Model II: A deterministic constant and stochast
ic drift
        TRUE,   // is the mode of testing
        alpha,   // is the statistical significance level
        1,   // is the statistical test to perform (1=ADF).
        TEST_PVALUE,      // is a switch to select the return output
        &fValue          // is the calculated test statistics
        );
// Scenario 3: A deterministic trend and stochastic drift
fValue = NAN;
nRet = NDK_ADFTEST(
        data,   // is the univariate time series data (a one dimensional array)
        100,   // is the number of observations
        maxOrder,         // is the lag length of the autoregressive process.
        ADFTEST_DRIFT_N_TREND, // Model III: A deterministic trend and stochastic
 drift
        TRUE,   // is the mode of testing
        alpha,   // is the statistical significance level
        1,   // is the statistical test to perform (1=ADF).
        TEST_PVALUE,      // is a switch to select the return output
        &fValue          // is the calculated test statistics
        );

// Scenario 4: A deterministic constant, trend and stochastic drift
fValue = NAN;
nRet = NDK_ADFTEST(
        data,   // is the univariate time series data (a one dimensional array)
        100,   // is the number of observations
        maxOrder,         // is the lag length of the autoregressive process.
        ADFTEST_DRIFT_N_CONST_N_TREND, // Model IV: A deterministic constant, tre
nd and stochastic drift
        TRUE,   // is the mode of testing
        alpha,   // is the statistical significance level
        1,   // is the statistical test to perform (1=ADF).
```

```
        1,    // is the statistical test to perform (1=ADF);
        TEST_PVALUE,      // is a switch to select the return output
        &fValue          // is the calculated test statistics
        );
```

int NDK_ADFTEST(double[]	**pData,**		**Namespace:** NumXLAPI
UIntPtr	**nSize,**		**Class:** SFSDK
UInt16	**maxOrder,**		**Scope:** Public
UInt16	**option,**		**Lifetime:** Static
BOOL	**testDown,**		
double	**alpha,**		
UInt16	**argMethod,**		
UInt16	**retType,**		
out double	**retVal**		
)			

Returns the p-value of the Augmented Dickey-Fuller (ADF) test, which tests for a unit root in the time series sample.

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]	**maxOrder**	is the lag length of the autoregressive process. If missing, an initial value equal to the cubic root of the input data size is used.
[in]	**option**	is the model description flag for the Dickey-Fuller test variant (1=no constant, 2=contant-only, 3=trend only, 4=constant and trend, 5=const, trend and trend squared).
[in]	**testDown**	is the mode of testing. If set to TRUE (default), ADFTest performs a series of tests. The test starts with the input length lag, but the actual length lag order used is obtained by testing down.
[in]	**alpha**	is the statistical significance level. If missing, a default of 5% is assumed.
[in]	**argMethod**	is the statistical test to perform (1=ADF).
[in]	**retType**	is a switch to select the return output:

Method	Value	Description
TEST_PVALUE	1	P-Value
TEST_SCORE	2	Test statistics (aka score)
TEST_CRITICALVALUE	3	Critical value.

[out] **retVal** is the calculated test statistics.

Remarks

1. The testing procedure for the ADF test is applied to the following model: $$\Delta y_t = \alpha + \beta_1 t + \beta_2 t^2 + \gamma y_{t-1} + \phi_1 \Delta y_{t-1} + \cdots + \phi_{p-1} \Delta y_{t-p+1} + \varepsilon_t$$

Where:

- Δ is the first different operator
- α is a constant
- β_1 is the coefficient on a time trend
- β_2 is the coefficient on a squared time trend

2. This model can be estimated, and testing for a unit root is equivalent to testing that $\gamma = 0$.

3. In sum, the Augmented Dickey-Fuller Test in Excel test hypothesis is as follows: $$H_{o}: \gamma = 0$$ $$H_{1}: \gamma < 0$$

Where:

- H_{o} is the null hypothesis (i.e. y_t has a unit-root)
- H_{1} is the alternate hypothesis (i.e. ${y_t}$ does not have a unit-root)

4. The test statistics (τ) value is calculated as follows: $$\tau = \frac{\hat{\gamma}}{\sigma_{\hat\gamma}}$$

where:

- $\hat{\gamma}$ is the estimated coefficient
- $\sigma_{\hat\gamma}$ is the standard error in the coefficient estimate

5. The test statistics value (τ) is compared to the relevant critical value for the Dickey-Fuller Test. If the test statistic is less than the critical value, we reject the null hypothesis and conclude that no unit-root is present.

6. The ADFTest does not directly test for stationarity, but indirectly through the existence (or absence) of a unit-root. Furthermore, Augmented Dickey-Fuller Test in Excel incorporates a deterministic trend (and trend squared), so it allows a trend-stationary process to occur.

7. The main difference between the ADFTest and a normal Dickey Fuller test is that ADFTest allows for higher-order autoregressive processes.

8. For the test-down approach, we start with a given maximum lag length and test down by running several tests; in each, we exaimine the high-order coefficient's t-stat for significance.

9. It is not possible to use a standard t-distribution to provide critical values for this test. Therefore this test statistic (i.e. τ) has a specific distribution simply known as the Dickey's-Fuller table.

10. The time series must have at least 10, and no more than 10,000 non-missing observations.
11. The time series is homogeneous or equally spaced.
12. The time series may include missing values (e.g. 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

Hull, John C.; Options, Futures and Other DerivativesFinancial Times/ Prentice Hall (2011), ISBN 978-0132777421

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")]