[web:reg] unit root DLL (BETA)

[web:reg] Kurt Annen

www.web-reg.de
annen@web-reg.de

Körner Str. 30
41464 Neuss
- Germany -

[web:reg] unit root DLL (BETA)

[web:reg] unit root DLL is a programmers' tool that lets you add the augmented Dickey Fuller test to your programs quickly and easily. You do not have to know anything about numerical methods to use the ADF test – just add it to your project like any other DLL and call its functions.

Requirements

This program requires Win95, Win98, NT 3.51, 4.0, Windows 2000 or Windows XP. This program also requires that you have any C++ Compiler, Visual Basic or Microsoft Office/OpenOffice.

Setup and Installation

To install the DLL you must download and execute the setup program from my webpage and follow than the installation instructions.

Functions

---

function	description	return value	parameter
pv_adf	calculates the p-value of the ADF Statistic (P-values based on MacKinnon 1996)	**p-value**	**t-tstat**: the t statistic **nobs**: number of observations (if nobs = 0 the asymp. cdf will be calculated) **model**: 0 = no constant / no trend 1 = constant / no trend 2 = constant / trend
cv_adf	calculates the critical value of the ADF Statistic (crit-values based on MacKinnon 1996)	**critical value**	**prob**: probability **nobs**: number of observations (if nobs = 0 then the asymtotic result will be calculated) **model**: 0 = no constant / no trend 1 = constant / no trend 2 = constant / trend
adf_p	calculates the ADF test by manual lag selection (pointer method)	**Error code:** 0 = Everything is OK 1 = Model is not 0,1 or 2 2 = Too few observations 3 = Covariance matrix is singular	**data**: pointer to data points **nobs**: number of observations model: 0 = no constant / no trend 1 = constant / no trend 2 = constant / trend **lag_length**: lagged difference **tstat**: reference t-Statistic **pvalue**: reference P-Value
adf_sa	calculates the ADF test by	**Error code**: 0 = Everything is	**data**: data as SAFEARRAY **model**:

	manual lag selection (SAFEARRAY method)	OK 1 = Model is not 0,1 or 2 2 = Too few observations 3 = Covariance matrix is singular 4 = No one-dimensional Safearray	0 = no constant / no trend 1 = constant / no trend 2 = constant / trend **lag_length**: lagged difference **tstat** = reference t-Statistic **pvalue** = reference P-Value
adf_auto_p	calculates the ADF test by auto lag selection (pointer method)	**Error code**: 0 = Everything is OK 1 = model is not 0,1 or 2 2 = too few observations 3 = covariance matrix is singular 5 = Infocrit. is not 0,1 or 2	**data**: pointer to data points **nobs**: number of observations **model**: 0 = no constant / no trend 1 = constant / no trend 2 = constant / trend **IC**: automatic selection method 0 = Akaike 1 = Schwarz 2 = Hannan-Quinn **max_lag**: reference maximum of lagged difference. if max_lag = 0 then max_lag will be calculated by *(int)(12*pow((nobs/100),0.25))* After calculation max_lag is the auto selected lag length **tstat**:reference t-Statistic **pvalue**: reference P-Value
adf_auto_sa	calculates the ADF test by auto lag selection (SAFEARRAY method)	**Error code**: 0 = Everything is OK 1 = model is not 0,1 or 2 2 = too few observations 3 = covariance matrix is singular 4 = No one-dimensional Safearray 5 = Infocrit. is not 0,1 or 2	**data**: data as SAFEARRAY **model**: 0 = no constant / no trend 1 = constant / no trend 2 = constant / trend **IC** = automatic selection method 0 = Akaike 1 = Schwarz 2 = Hannan-Quinn **max_lag**: reference maximum of lagged difference. if max_lag = 0 then max_lag will be calculated by *(int)(12*pow((nobs/100),0.25))* After calculation max_lag is the auto selected lag length **tstat**: reference t-Statistic **pvalue**: reference P-Value

further notes

There are also examples for C++ and VBA (Excel) included. This examples describes the using of the unitroot.dll. Any other declarations can be derived from the information in unit_root.h.
If you are using MSVC you need also the unitroot.lib. I do not use MSVC but you may easily produce the library by the Microsoft LIB and tool:

```
lib /machine:i386 /def:unitroot.def
```