INSTRUCTION MANUAL
PROGRAM PACKAGE FOR ANALYZING CONVERGENCE WITH THE USE OF ASYMPTOTICALLY SUBSIDING TRENDS

Developed by **Konstantin Gluschenko**
glu@nsu.ru
http://nsu-ef.ru/teachers/Gluschenko_KP/
First version: 2010.07
This version: 2021.07

1. Introduction

The programs deal with a time series models containing nonlinear asymptotically subsiding trends. Such models describe convergence of incomes, outputs, prices, etc. in the time series context.

The programs are applicable in EViews 7 and later versions. They are provided in ASCII format with extension *.prg* and are ready for use with EViews; all you need is to copy programs to your computer. The programs use file *TauStatistics-NonlinearTrends.xls* which is also provided. You have to copy this file as well and place it to the default (for your EViews) directory. You may modify this file or create such a file by yourself. To do so, you should know its structure which is described in Section 5 of this document.

The programs are licensed to be available to users to download, copy, use, and modify (except for a commercial use). I hope that in doing so you will acknowledge me as the original creator.

The package consists of six programs:

Testing_Convergence.prg estimates models for a single time series or the difference between two series and test them for unit root;

Testing_Convergence (Batch).prg estimates models for the differences between every time series from a specified group and a series specified as a benchmark; the program also tests these models as well as the Dickey-Fuller models with and without constant for unit root;

Testing_Convergence (Total).prg estimates models for differences between two series in all pairs of series from a specified group; the program also tests these models as well as the Dickey-Fuller models with and without constant for unit root;

CDF of UR statistics with Log-exponential trend.prg, **CDF of UR statistics with Exponential trend.prg**, and **CDF of UR statistics with Fractional trend.prg** are auxiliary programs; they estimate cumulative distribution functions (CDF) of the unit root test statistics (tau-statistics) under the null hypothesis for equations with the respective trend and a specified sample size. These results are used to fill file *TauStatistics-NonlinearTrends.xls* file (for example, to replace CDF with that obtained with more replications, or to add CDF for an additional sample size).

2. The models

The starting point is the following time-series definition of convergence: Economic entities r and s described by an indicator Y converge over time, t, if

$$\lim_{t\to\infty} \log(Y_{rt} / Y_{st}) = 0 . \quad (1)$$

(See Bernard and Durlauf, 1995, or Durlauf and Quah, 1999, for a more evolved definition.)

Such a process can be described by an asymptotically subsiding trend $h(t)$ such that $d|h(t)|/dt < 0$ and

$h(t) \to 0$ as $t \to \infty$. The package deals with three kinds of such trends, namely,

- log-exponential trend $h(t) = \log(1 + \gamma e^{\delta t})$, $\delta < 0$;
- exponential trend $h(t) = \gamma e^{\delta t}$, $\delta < 0$;
- fractional trend $h(t) = \dfrac{\gamma}{1+\delta t}$, $\delta > 0$.

A time series $y_{rst} = \log(Y_{rt}/Y_{st})$ is modeled by a first-order autoregressive process, AR(1), with trend $h(t)$:

$$y_{rst} = h(t) + \nu_t,\ \nu_t = (\lambda + 1)\nu_{t-1} + \varepsilon_t, \tag{2}$$

where $\lambda + 1$ is the autoregressive coefficient and ε_t is a white noise. Applying the Cochrane-Orcutt transformation (i.e., substituting the second equation in (2) into the first one), we get a nonlinear econometric model to be estimated and tested:

$$\Delta y_{rst} = \lambda y_{rs,t-1} + h(t) - (\lambda + 1)h(t-1) + \varepsilon_t. \tag{3}$$

The specific models with the above trends have the following forms, respectively ($t = 2,\ldots,T$):

$$\Delta y_{rst} = \lambda y_{rs,t-1} + \log(1 + \gamma e^{\delta t}) - (\lambda + 1)\log(1 + \gamma e^{\delta(t-1)}) + \varepsilon_t; \tag{3a}$$

$$\Delta y_{rst} = \lambda y_{rs,t-1} + \gamma e^{\delta t} - (\lambda + 1)\gamma e^{\delta(t-1)} + \varepsilon_t = \lambda y_{t-1} + \gamma' e^{\delta t} + \varepsilon_t, \text{ where } \gamma' = \gamma(1 - (\lambda + 1)e^{-\delta}); \tag{3b}$$

$$\Delta y_{rstt} = \lambda y_{rs,t-1} + \frac{\gamma}{1+\delta t} - (\lambda + 1)\,\frac{\gamma}{1+\delta(t-1)} + \varepsilon_t. \tag{3c}$$

If hypotheses $\lambda = 0$ (against $\lambda < 0$), $\gamma = 0$ (against $\gamma \neq 0$), and $\delta = 0$ (against $\delta \neq 0$) can be rejected, y_{rst} is a trend stationary process about the given nonlinear trend. With the "right" sign of δ, convergence between r and s takes place; the "wrong" sign of δ ($\delta > 0$ in the log-exponential and exponential trends or $\delta < 0$ in the fractional trend) implies divergence.

Note that there is confusion regarding the term "convergence" in the literature. Model (3) in fact describes a superposition of two processes that can be called long-run, or deterministic, convergence, and stochastic, or short-run, convergence. Long-run convergence is a deterministic path of y_{rst} that tends to zero over time (catching-up, or, so to say, "genuine convergence"). Short-run convergence is an autocorrelated stochastic process containing no unit root (i.e., a stationary process), $\nu_t = (\lambda + 1)\nu_{t-1} + \varepsilon_t$. Intuitively, short-run convergence characterizes the behavior of transient random shocks. A unit shock deviates the process from its long-run path, dying out over time with half-life $\theta = \ln(0.5)/\ln(\lambda + 1)$, so that y_{rst} eventually returns to its long-run path. It is the later process that is sometimes called "convergence" in the literature.

An advantage of the log-exponential trend is the ease of interpretation. Parameter γ is the initial (at $t = 0$) disparity, i.e., the deviation of Y_{r0}/Y_{s0} from 1. Parameter δ characterizes the convergence rate which can be simply expressed in terms of the half-life time of the (deterministic) disparity, or semi-convergence (see Table 1). A shortcoming of this trend is in that is has no symmetry properties with respect to a permutation of the economy indices. Albeit $y_{rst} = -y_{srt}$, the permutation changes values of γ and δ (and may change the estimate of λ in regression (3a), hence, p-value of unit root test). Contrastingly, the exponential and fractional trends have symmetry properties. A permutation of r and s changes only the sign of γ, leaving its absolute value and the value of δ as well as λ in (3b), (3c) intact. However, while the initial disparity can be easily calculated from γ, the half-lives of the deterministic disparity involve a mixture of γ and δ; besides, they depend on t. This results in hardly interpretable expressions.

Table 1 reports initial disparities, $Y_{r0}/Y_{s0} - 1$, and half-life times of the (deterministic) disparity $Y_{rt}/Y_{st} - 1$ (the time the disparity takes to halve) in terms of the model parameters. The half-life time $\Theta_{(1/2)}$ is such

that $Y_{r,t+\Theta}/Y_{s,t+\Theta} - 1 = (Y_{rt}/Y_{st} - 1)/2$. In the case of divergence, its rate is characterized by the doubling time $\Theta_{(2)}$, i.e., the time the disparity takes to double: $Y_{r,t+\Theta}/Y_{s,t+\Theta} - 1 = 2(Y_{rt}/Y_{st} - 1)$. As the half-life times and doubling times for the exponential and fractional trends depend on t, the table reports these times for halving/doubling the initial disparity. If $e^{\gamma} \leq 0.5$ with diverging exponential or fractional trend, the doubling time is computed with γ replaced by its absolute value.

Table 1. Interpretation of the model parameters

Trend	Initial disparity, $Y_{r0}/Y_{s0} - 1$	Half-life time $\Theta_{(1/2)}$	Doubling time $\Theta_{(2)}$
log-exponential	γ	$\frac{\log(0.5)}{\delta}$	$\frac{\log(2)}{\delta}$
exponential	$e^{\gamma} - 1$	$\frac{1}{\delta}\log\left(\frac{\log(0.5(e^{\gamma}+1))}{\gamma}\right)$	$\frac{1}{\delta}\log\left(\frac{\log(2e^{\gamma}-1)}{\gamma}\right)$
fractional	$e^{\gamma} - 1$	$\frac{1}{\delta}\left(\frac{\gamma}{\log(0.5(e^{\gamma}+1))}-1\right)$	$\frac{1}{\delta}\left(\frac{\gamma}{\log(2e^{\gamma}-1)}-1\right)$

3. Estimation and testing

3.1. Estimation

The programs estimate models (3a)–(3b) with the use of nonlinear least squares, benefiting from the built-in EViews procedure. They set the maximum number of iterations M to 1000, and the convergence tolerance C (see, e.g., Quantitative Micro Software, 2010b, p. 627) to 10^{-7}. The operators that call the estimation procedure look like <Equation name>.ls(**M**=1000,**C**=1e-07) <Specification>.

Since the sum of squared residuals can be multiextremal, the estimations are repeated with the use of different sets of initial values of λ, γ, and δ. These are estimates in simplified versions of Equation (3). Table 2 reports the ways of obtaining initial values of the coefficients. The estimation of regression (3a) applies all seven sets of initial values; estimating regression (3b) applies six sets; and regression (3c) is estimated with the use of five sets. Out of all estimates obtained with different initial values of the coefficients, an estimate that yields the best fit (minimal sum of squared residuals) is selected.

In some cases, the iterative procedure of estimating will not converge under the specified number of iterations and specified convergence tolerance. Some other errors may occur as well, such as non-positive number in a logarithm, overflow, division by zero, etc. (for instance, when EViews tries values beyond the domain of parameters). A part of such errors can be reported during the execution of programs, not interrupting the execution if the number of errors does not exceed the user-specified (in the Run window of the program) value of Maximum errors before halting. (Therefore, ignore such messages.) While trying different sets of initial values, programs simply skip unsuccessful estimations. If estimations with all sets of initial values fail, different programs process such an event in their own way; see the description of specific programs in Section 4.

Programs *Testing_Convergence (Batch)* and *Testing_Convergence (Total)* also estimate linear AR(1) models, that with constant,

(4) $$\Delta y_{rst} = \lambda y_{rs,t-1} + \gamma + \varepsilon_t,$$

and without constant,

(5) $$\Delta y_{rst} = \lambda y_{rs,t-1} + \varepsilon_t.$$

Table 2. Estimating initial values of coefficients

Assumption	Auxiliary regression(s)	Initial values of coefficients in (3)
Zero parameters		
$\lambda = 0,\ \gamma = 0,\ \delta = 0$		$\lambda_0 = 0,\ \gamma_0 = 0,\ \delta_0 = 0$
$\gamma = 0,\ \delta = 0$	$\Delta y_t = \lambda y_{t-1} + \varepsilon_t$	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = 0,\ \delta_0 = 0$
$\delta = 0$	$\Delta y_t = \lambda y_{t-1} + \alpha + \varepsilon_t$	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = \exp(-\hat{\alpha} / \hat{\lambda}) - 1$ for (3a), $\gamma_0 = -\hat{\alpha} / \hat{\lambda}$ for (3b) and (3c), $\delta_0 = 0$
Linear approximations		
$\ln(1 + \gamma e^{\delta t}) \approx \gamma(1 + \delta t)$, $\gamma e^{\delta t} \approx \gamma(1 + \delta t)$, $\gamma/(1 + \delta t) \approx \gamma(1 - \delta t)$	$\Delta y_t = \lambda y_{t-1} + \alpha + \beta t + \varepsilon_t$, where $\alpha = (\lambda + 1)\gamma\delta - \lambda\gamma$, $\beta = -\lambda\gamma\delta$ for (3a) and (3b) or $\beta = \lambda\gamma\delta$ for (3c)	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = -(\hat{\lambda}\hat{\alpha} + (\hat{\lambda}+1)\hat{\beta}) / \hat{\lambda}^2$, $\delta_0 = -\hat{\beta} / \hat{\lambda}\gamma_0$ for (3a) and (3b), $\delta_0 = \hat{\beta} / \hat{\lambda}\gamma_0$ for (3c)
Semi-linear approximation (for (3a) only)		
$e^{\delta t} \approx 1 + \delta t$	$\Delta y_t = \lambda y_{t-1} + \ln(1 + \gamma(1 + \delta t)) -$ $-(\lambda + 1)\ln(1 + \gamma(1 + \delta(t-1))) + \varepsilon_t$	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = \hat{\gamma}$, $\delta_0 = \hat{\delta}$
Semi-linear approximation for (3a) and simplified equation for (3b)		
$\ln(1 + \gamma e^{\delta t}) \approx \gamma e^{\delta t}$ for (3a), $\gamma e^{\delta t} - (\lambda + 1)\gamma e^{\delta(t-1)} = \gamma' e^{\delta}$ for (3b)	$\Delta y_t = \lambda y_{t-1} + \gamma' e^{\delta t} + \varepsilon_t$, where $\gamma' = \gamma(1 + (\lambda + 1)e^{-\delta})$	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = \hat{\gamma}' / (1 - (\hat{\lambda} + 1)e^{-\hat{\delta}})$, $\delta_0 = \hat{\delta}$
Detrending	$y_t = \ln(1 + \gamma e^{\delta t}) + \nu_t$ for (3a), $y_t = \gamma e^{\delta t} + \nu_t$ for (3b), $y_t = \gamma/(1 + \delta t) + \nu_t$ for (3c), and $\Delta\hat{\nu}_t = \lambda\hat{\nu}_{t-1} + \varepsilon_t$	$\lambda_0 = \hat{\lambda}$, $\gamma_0 = \hat{\gamma}$, $\delta_0 = \hat{\delta}$

These models describe cases of non-convergence with the disparity remaining time-invariant (up to random shocks). In equation (4), $y_{rst} = -\gamma/\lambda$ or $Y_{rt}/Y_{st} - 1 = e^{-\gamma/\lambda} - 1$. Equation (5) corresponds to the case of the absence of disparity: $y_{rst} = 0$. The programs use the built-in EViews procedure to estimate these equations. In program outputs and elsewhere, model (4) is called ARc, and model (5) is called ARo.

It can occur that time series y_{rst} satisfies no one of models (3), (4), and (5). Two reasons of this are possible. First, process y_{rst} follows a random walk. Second, both the null hypothesis of random walk and the alternative hypothesis are wrong. That is, the process y_{rst} has some regularity (for example, has a U-shape trend); however, the models applied are not able to describe it.

3.2. Testing for unit root

The programs exploit two unit root tests, the augmented Dickey-Fuller (ADF) test and Phillips-Perron (PP) test. These tests make it possible to take account of possible autocorrelation of a form other than AR(1). Testing hypotheses $\lambda = 0$ (against $\lambda < 0$), both tests use the *t*-ratio of λ (hereafter, tau-statistic) $\tau = \hat{\lambda} / \hat{\sigma}_{\lambda}$. However, they modify it in different ways. To find *p*-value of a (modified) τ, the programs use file *TauStatistics-NonlinearTrends.xls* that contains the cumulative distribution functions of τ for equations with the above trends and different sample sizes.

The programs perform the ADF test in the following manner. At first, they find the optimal lag length. To

do so, a detrended series is generated as $\tilde{y}_{rst} = y_{rst} - \hat{h}(t)$, where $\hat{h}(t)$ is the estimate of trend in (3): $\hat{h}(t) = \log(1 + \hat{\gamma}\exp(\hat{\delta}t))$, or $\hat{h}(t) = \hat{\gamma}\exp(\hat{\delta}t))$, or $\hat{h}(t) = \hat{\gamma}/(1 + \hat{\delta}t)$. Then the built-in EViews procedure of the ADF test is applied to the detrended series $\tilde{y}$ with the use of operator URoot(ADF, none, info=MSIC, ...) ỹ, which yields the optimal lag length *k**. Note that the modified Schwartz (Bayesian) information criterion with a sample-dependent penalty factor is used, and not a standard criterion. This modification is due to Ng and Perron (2001), who note that the standard information criteria tend to select lag lengths that are generally too small for unit root tests to have good sizes. Having found the optimal number of lags *k**, the programs estimate the following auxiliary regression:

$$\Delta y_{rst} = \lambda_0 y_{rs,t-1} + h(t) - (\lambda + 1)h(t-1) + \sum_{l=1}^{k^*} \lambda_l \Delta y_{rs,t-l} + \varepsilon_t . \quad (6)$$

This provides the ADF test statistic $\tau_{\mathrm{ADF}} = \hat{\lambda}_0 / \hat{\sigma}_{\lambda_0}$. Then programs draw *p*-value of τ_{ADF} for equation with the respective trend from the file *TauStatistics-NonlinearTrends.xls*. The programs estimate regression (6) in the same manner as they do to estimate regression (3); see Section 3.1. The difference is that the initial values of λ_l are always $\lambda_l = 0$ for all $l > 0$.

Note that the role of regression (6) is merely technical. It serves for obtaining the value of τ_{ADF} only. The programs take estimates of λ and other regression parameters from the original regression (3).

To perform the PP rest, programs compute its statistic with the *autoregressive spectral density estimator* at frequency zero ω_0^2 (in contrast to commonly used kernel-based estimators), benefiting from results obtained while performing the ADF test:

$$\tau_{\mathrm{PP}} = \tau \sqrt{\frac{\sigma^2(\hat{\varepsilon})}{\omega_0^2}} - \frac{(T-1)(\omega_0^2 - \sigma^2(\hat{\varepsilon}))\hat{\sigma}_{\lambda}}{2\omega\hat{\sigma}_{\varepsilon}},$$

where $\omega_0^2 = \sigma^2(\hat{\varepsilon})/(1 - \sum_{l=1}^{k^*} \hat{\lambda}_l)^2$ (Perron and Ng, 1998) and $\hat{\sigma}_{\varepsilon}$ is the standard error of regression: $\hat{\sigma}_{\hat{\varepsilon}}^2 = \sum_{t=2}^{T} \hat{\varepsilon}_t^2 / (T-4)$, $T - 4$ being the degrees of freedom. Estimates $\hat{\lambda}$, $\hat{\sigma}_{\lambda}$, $\sigma^2(\hat{\varepsilon})$, and $\hat{\sigma}_{\varepsilon}$ are those from regression (3), while $\hat{\lambda}_l$ are estimates from regression (6). The application of the OLS autoregressive spectral method bases on Perron and Ng (1996) in order to avoid size distortions peculiar to the PP test with kernel-based estimators. The programs draw *p*-value of τ_{PP} for equation with the respective trend also from the file *TauStatistics-NonlinearTrends.xls*.

When there are no additional lags, i.e. $k^* = 0$, $\tau_{\mathrm{ADF}} = \tau_{\mathrm{PP}} = \tau$.

To test regressions (4) and (5) for unit root, programs *Testing_Convergence (Batch)* and *Testing_Convergence (Total)* use the built-in procedures of EViews with the same options as above, namely, the use of the modified Schwartz criterion and the OLS (not-detrended) autoregressive spectral method.

The operators URoot(ADF, const, info=MSIC, ...) and URoot(PP, const, hac={%AR}, info=MSIC, ...) for (4), and the operators URoot(ADF, none, info=MSIC, ...) and URoot(PP, none, hac={%AR}, info=MSIC, ...) for (5) in these programs call the respective procedures. There is a mistake in EViews 7: it misinterprets parameter hac=AR as hac=ARdt (OLS detrened). Therefore, to force executing the desirable versions of the PP test by EViews, formally incorrect value of hac is set by operator %AR="ARdt". If the mistake is corrected in a later of EViews, this operator in the above programs should be changed to %AR="AR".

Note that the applied versions of the tests are more severe in rejecting unit root than commonly used versions. The latter may cause over-rejections of the unit root hypothesis by the ADF test due to the choice of the lag length on the basis of standard information criteria (Ng and Perron, 2001). They may also cause over-rejections of the unit root by the PP test due to kernel-based spectral density estimators (Perron and Ng, 1996).

4. Instructions to the programs

4.0 Executing a program

This section is valid for EViews 7. Later versions can contain some additional options or changes in dialog boxes, settings, etc. However, these are hardly fundamental; I think the instructions below can be easily applied to a newer version of EViews.

To run a program, the standard way is used; see, e.g. Quantitative Micro Software (2010a). To load a program, click on `File` → `Open` → `Program`... in the EViews main menu, choose a program you need and open it.

To run the program, click on the `Run` button on the program window. On the `Reporting` tab of the `Run Program` dialog box, use the radio button to choose `Quiet` mode. In the `Log messages` section, specify `Program lines: Always off`, `Status line messages: Program controlled`, `User log messages: Program controlled`, `Program errors: Program controlled`.

On the `Settings` tab, set the `Maximum errors before halting` to a maximum possible number (in EViews 7, 10,000). Then EViews will continue running the program until this number of errors is reached. Note that even 10,000 may prove insufficient for programs *CDF of UR statistics with Log-exponential trend*, *CDF of UR statistics with Exponential trend*, and *CDF of UR statistics with Fractional trend*, especially for the first of them, when they should execute a great number of replications (e.g. 1 million). Section 4.4 explains what to do in this case.

On the `Settings` tab, you should also specify `Program arguments`. They depend on a specific program; Sections 4.1–4.4 describe arguments of respective programs. After all, push `OK` button on the `Run Program` dialog window to start the program.

Figure 1 shows examples of starting a program in EViews 7.

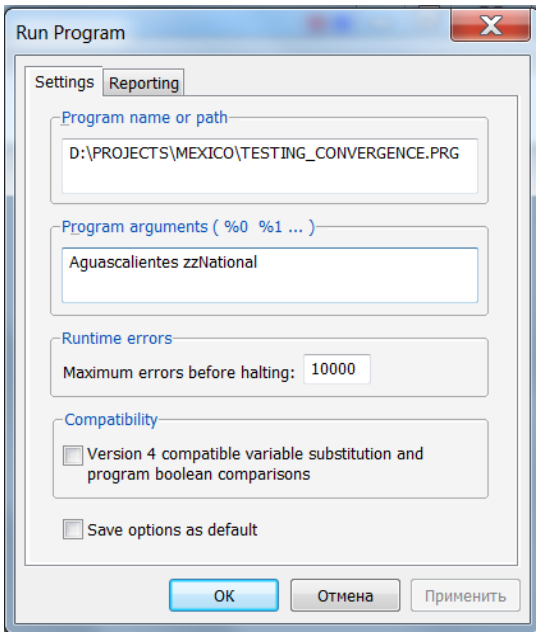


Figure 1. Starting program *Testing_Convergence*

Programs *Testing_Convergence*, *Testing_Convergence (Batch)*, and *Testing_Convergence (Total)* load file *TauStatistics-NonlinearTrends.xls* from the current data directory of EViews. To set it, go through `Options` → `General Options`... → `File locations` in the EViews main menu and fill out the `Current Data Path` section. While loading file *TauStatistics-NonlinearTrends.xls*, these programs check it and halt if they encounter a non-recoverable error in the file, reporting the error in the status line (at the lower left corner of the EViews window). If *TauStatistics-NonlinearTrends.xls* does not contain information for a sample size T specified

in the workfile, the programs take the nearest sample size smaller than T (or greater, if there is no smaller sample size), reporting this in a program-specific way.

Programs *Testing_Convergence*, *Testing_Convergence (Batch)*, and *Testing_Convergence (Total)* need a workfile with time series to be analyzed. Therefore, you should have such a file opened before starting a program.

4.1 Testing_Convergence

Function. This program analyzes models (3a), (3b), and (3c) for a single time series or the difference of two series (the series are supposed then to be logarithms of an indicator under consideration).

Arguments. Arguments of the program have the form <*series1*> [<*series2*>], containing the name/s of time series to be analyzed. Hereafter, <...> means some value of argument.

Source data. The program uses an EViews workfile with time series to be analyzed as well as file *TauStatistics-NonlinearTrends.xls* with the distributions of τ-statistics.

Execution. At first, the program checks whether *series1* (and *series2*, if specified) exists in an opened workfile. If one (or both) does not, the program reports this in the status line and stops. When two series are specified, the program generates series *series1* – *series2* with the name *series1_series2* (truncating it to 24 symbols if needed) and saves it to the opened workfile. (**Note** that the truncated name can coincide with a name of existing series which will be overwritten. In such a case you need to make a relevant renaming/s.) Then the program estimates models with the given series and tests them for unit root as described in Section 3.

Results. The program puts on the screen three tables `LOG_EXPONENTIAL_TREND_`, `EXPONENTIAL_TREND_`, and `FRACTIONAL_TREND_`. These tables are similar to standard EViews outputs for equations; however, they contain results of testing for unit root. The program saves these tables as well as standard outputs named `LOG_EXPONENTIAL_TREND`, `EXPONENTIAL_TREND`, and `FRACTIONAL_TREND` to the opened workfile. If the program is unable to estimate some model, it outputs the program log with message `Estimation of equation <EQUATION NAME> is impossible` instead of the equation output.

Table: LOG_EXPONENTIAL_TREND_ Workfile: MEXICAN STATES+::Untitled\

View | Proc | Object | Print | Name | Edit+/- | CellFmt | Grid+/- | Title | Comments+/-

Dependent Variable: D(AGUASCALIENTES_CAMPECHE)
Method: Least Squares
Date: 07/27/21 Time: 04:00
Sample (adjusted): 1941 2015
Included observations: 75 after adjustments
Convergence achieved after 1 iteration
D(AGUASCALIENTES_CAMPECHE)=C(1)*AGUASCALIENTES_CAMPECH
E(-1)+LOG(1+C(2)*EXP(C(3)*@TREND))-(C(1)+1)*LOG(1+C(2)
EXP(C(3)(@TREND -1)))

	Coefficient	Std. Error	t-Statistic	Prob.
C(1) = lambda	-0.024126	0.031992	-0.754121	0.4532
C(2) = gamma	2.00E-05	0.000222	0.090021	0.9285
C(3) = delta	0.151928	0.140975	1.077696	0.2848

R-squared	0.056137	Mean dependent var	0.006116
Adjusted R-squared	0.029919	S.D. dependent var	0.152384
S.E. of regression	0.150087	Akaike info criterion	-0.916025
Sum squared resid	1.621880	Schwarz criterion	-0.823326
Log likelihood	37.35094	Hannan-Quinn criter.	-0.879011
Durbin-Watson stat	0.877339		

UNIT ROOT TESTS

	tau-Statistic	p-Value	Bandwidth/lag length
Phillips-Perron test	-2.787112	0.165	1
Augmented Dickey-Fuller test	-2.944848	0.126	1

There is no tau-statistic for T=76 in TauStatistics-NonlinearTrends.xls. This for T=75 is used instead

Figure 2. One of output tables reported by program *Testing_Convergence*

Figure 2 provides an example of an output table. The reported results of testing for unit roots are estimated tau-statistics τ_{PP} and τ_{ADF}, their p-values, as well as the bandwidth for the PP test and the lag length for the ADF test (under the applied method of unit root testing – see Section 3.2, the bandwidth and lag length coincide). In this example, you can see the way of reporting the case of lacking sample size.

4.2 Testing_Convergence (Batch)

Function. This program analyzes models (3a), (3b), (3c), (4), and (5) for the differences of every series in the specified group and a series specified as the benchmark.

Arguments. Arguments of the program have the form <*group*> <*benchmark series*> containing the name of the series group to be analyzed and the series serving as a benchmark. The benchmark series may be a member of the specified group or be beyond the group.

Source data. The program uses an EViews workfile with time series to be analyzed as well as file *TauStatistics-NonlinearTrends.xls* with the distributions of τ-statistics.

Execution. At first, the program checks whether *group* and *benchmark series* exist in an opened workfile. If one (or both) does not, the program reports this in the status line and stops. Then the program analyzes models with given series (as described in Section 3) in the following order: (3a) → (3b) → (3c) → (4) and (5). Every model is analyzed for all generated series $y_r - y_*$, where y_* is the benchmark series (the generated series are not saved). Series y_r are processed in their order in the specified group (that not necessary coincides with their alphabetical order in the workfile). The program reports the progress of execution in the status line, indicating the model being analyzed and current number of series, e.g., exp trend: r=17.

Results. The program stores the results obtained in the output table named `ESTIMATES_<benchmark>`. If such a name exceeds 24 symbols, the program reports this in the program log as ESTIMATES_<benchmark> is an illegal or reserved name in "RENAME ESTIMATES ESTIMATES_<benchmark>" and names the table simply ESTIMATES. Then you need to rename it as you wish.

Figure 3 shows an example of the output table.

Table: ESTIMATES_BAJA_CALIF Workfile: MEXICAN STATES::Untitled\

View | Proc | Object | Print | Name | Edit+/- | CellFmt | Grid+/- | Title | Comments+/-

Benchmark: BAJA_CALIFORNIA

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No.	Location name	Specification	lambda	se	t	PP-t	p-value	Ba...	ADF-t	p-value	lags	gamma	se	p-value	delta
2	1	AGUASCALI...	log	-0.007122	0.028005	-0.254316	-0.694546	0.893	1	-0.700373	0.892	1	0.254859	2.047590	0.901292	0.031013
3			exp	-0.093799	0.055834	-1.679957	-2.312953	0.314	1	-0.737361	0.955	1	-9.582421	29.88440	0.749405	-0.069661
4			fract	-0.031557	0.007112	-4.437406	-3.887673	0.053	1	-3.222677	0.121	1	5.063246	2.458537	0.043068	0.068890
5			ARc	-0.000108	0.014452	-0.007481	-0.007481	0.954	0	-0.007481	0.954	0	0.022169	0.013807	0.112657	
6			ARo	-0.017062	0.009973	-1.710845	-1.533900	0.117	1	-1.882761	0.057	1				
7	2	BAJA_CALIF...	Benchmark location													
8																
9																
10																
11																
12	3	BAJA_CALIF...	log	-0.341656	0.086689	-3.941158	-3.279243	0.067	7	-2.308130	0.333	7	-0.915742	0.203149	2.48E-05	-0.047894
13			**exp**	-0.342619	0.086671	-3.953104	-3.953104	0.010	0	-3.953104	0.010	0	-2.455887	0.686835	0.000629	-0.094433
14			fract	-0.161848	0.072457	-2.233720	-1.214000	0.801	4	-1.977914	0.513	4	-1.117420	6.532364	0.864657	0.087855
15			ARc	-0.104532	0.043760	-2.388746	-2.260284	0.187	4	-2.706679	0.078	4	-0.010756	0.027588	0.697760	
16			**ARo**	-0.096988	0.039025	-2.485292	-2.603820	0.010	4	-3.058350	0.003	4				
17	4	CAMPECHE	log Estimation impossible													
18			exp	-0.039975	0.040931	-0.976653	-2.134180	0.395	1	-1.650106	0.637	1	0.243206	5.574018	0.965318	-0.012311
19			fract	-0.041151	0.029645	-1.388128	-2.035821	0.489	1	-2.465392	0.315	1	0.605879	2.983711	0.839659	0.047041
20			ARc	-0.041729	0.024226	-1.722489	-1.900600	0.330	1	-1.610194	0.472	1	0.003033	0.019345	0.875857	
21			**ARo**	-0.043222	0.022129	-1.953148	-1.989628	0.045	1	-1.627503	0.097	1				
22	Benchmark location: BAJA_CALIFORNIA															
23	Started: 12/07/2020 17:29; elapsed time 0 min.															
24	Log-exponential trend: tau-statistic for T=76															
25	Exponential trend: tau-statistic for T=76															
26	Fractional trend: tau-statistic for T=76															
27																

Figure 3. An example of the output table generated by program *Testing_Convergence (Batch)*

The output table contains coefficient estimates (the column named lambda, gamma, and delta), their standard errors (se) and *p*-values, information on unit root tests, namely, τ-statistics τ, τ_{PP}, and τ_{ADF} (named t, PP-t, and ADF-t), *p*-values of the PP and ADF tests (p-value), bandwidth and lag length (Bandwidth and lags), adjusted R^2 (AdjR2), and sum of squared residuals (SSR). Besides, the table contains initial disparity in percentage terms (Disparity, %) and halflife/doubling time (Halflife) computed as indicated in Table 1. If it is impossible to compute these indicators (because of too great γ), they are displayed as NA.

There are five rows with the results for every series; each row corresponds to a respective model. If the benchmark series is a member of the group, these rows are empty for it (containing indication Benchmark location).

Cells with *p*-values of the unit root tests and coefficients exceeding 0.1 are colored in yellow. Cells with δ corresponding to divergence and the respective doubling times are colored in red. If estimation of some model has failed, the relevant row contains indication Estimation impossible. Specifications of valid models (i.e. with all *p*-values not more than 0.1) are marked in bold font. If more than one model with nonlinear trend is valid, bold font marks a model with the minimal SSR.

After rows with the results, the program indicates the benchmark location, execution time, and sample sizes of τ-statistics drawn from *TauStatistics-NonlinearTrends.xls*.

4.3 Testing_Convergence (Total)

Function. This program analyzes models (3a), (3b), (3c), (4), and (5) for the differences between series in every pair of series from the specified group.

Arguments. The first arguments of the program is <*group*>, specifying the name of the series group to be analyzed. Optional arguments may follow in any order. Argument **`ADF=no`** or **`PP=no`** switches off the respective unit root test; both tests are performed by default. Arguments **`log=no`**, **`exp=no`**, and **`fract=no`** make it possible to switch off analyzing models with log-exponential trend, exponential trend, ad fractional trend, respectively (only one or two such arguments are possible); all three models are analyzed by default.

Source data. The program uses an EViews workfile with time series to be analyzed as well as file *TauStatistics-NonlinearTrends.xls* with the distributions of τ-statistics.

Execution. At first, the program checks whether *group* exists in an opened workfile. If it does not, the program reports this in the status line and stops. Then the program checks the correctness of optional arguments (if they are specified), halting in the case of error. After that the program analyzes models with given series (as described in Section 3) in the following order: (3a) → (3b) → (3c) → (4) and (5). Every model is analyzed for all generated series $y_{rs} = y_r - y_s$, $s > r$ (the generated series are not saved). Thus, there are $N(N-1)/2$ pairs of series, where N is the number of series in the specified group; $\{r\}$ and $\{s\}$ are the series numbers in the specified group (that not necessary correspond to their order in the workfile). Since unit root can be accepted in model (3a) for y_{rs} and rejected for y_{sr} (see Section 2), this model is additionally estimated with y_{sr} in the case of non-rejection of unit root for y_{rs}.

The program reports the progress of execution in the status line, indicating the model being analyzed and current numbers of series, e.g., exp trend: r=17, s=21.

Results. The program stores the results obtained in the output table `PAIRWISE_SUMMARY` and matrix `PAIRWISE_MATRIX`.

Table `PAIRWISE_SUMMARY` contains summarized results of the analysis. For every time series r it reports the number of other series s (s = 1, …, N, and $s \neq r$) for which the difference series $y_{rs} = y_r - y_s$ is described by model (5) – ARo, by model (4) – ARc, at least by one version of model (3) with the correct sign of δ – Converging, at least by one version of model (3) with incorrect sign of δ – Diverging, and described by no one model – No model. The total equals $N - 1$. These results relate to two approaches: specific to general and general to specific. In the first case, the first valid model in the sequence (5) → (4) → (3) is taken as describing process y_{rs}. (A model is deemed valid if *p*-values of the unit root test/s and all coefficients are no more than 0.1.) The general-to-specific approach considers the reverse sequence: (3) → (4) → (5).

Note that the total of all rows and columns under a given approach equals $N(N-1)$, i.e., twice the number of series pairs, as every series enters in pairs (r, s) and (s, r).

After rows with the results, the program reports the execution time and sample sizes of τ-statistics drawn from *TauStatistics-NonlinearTrends.xls*.

Figure 4 shows an example of table `PAIRWISE_SUMMARY`.

Table: PAIRWISE_SUMMARY Workfile: MEXICAN STATES::Untitled\

View | Proc | Object | Print | Name | Edit+/- | CellFmt | Grid+/- | Title | Comments+/-

	A	B	C	D	E	F	G	H	I	J	K	L
1	No.	Location ...	SPECIFIC TO GENERAL					GENERAL TO SPECIFIC				
2			ARo	ARc	Converging	Diverging	No model	ARo	ARc	Converging	Diverging	No model
3	1	AGUASCA...	2	0	15	2	12	1	0	15	3	12
4	2	BAJA_CA...	25	1	1	0	4	13	4	10	0	4
5	3	BAJA_CA...	5	8	2	0	16	4	8	3	0	16
6	4	CAMPECHE	17	0	0	0	14	17	0	0	0	14
7	5	COAHUILA	10	3	3	0	15	8	3	5	0	15
8	6	COLIMA	10	4	2	1	14	5	6	5	1	14
9	7	CHIAPAS	2	1	0	2	26	2	1	0	2	26
10	8	CHIHUAH...	4	4	3	0	20	4	3	4	0	20
11	9	CIUDAD_...	3	4	3	0	21	2	4	4	0	21
12	10	DURANGO	16	4	3	0	8	5	9	9	0	8
13	11	GUANAJU...	11	0	4	0	16	4	1	9	1	16
14	12	GUERRE...	9	3	2	0	17	4	4	6	0	17
15	13	HIDALGO	3	4	4	1	19	1	5	5	1	19
16	14	JALISCO	10	1	4	0	16	8	2	5	0	16
17	15	MEXICO	9	1	6	0	15	8	1	7	0	15
18	16	MICHOAC...	9	2	3	1	16	3	4	7	1	16
19	17	MORELOS	10	0	3	0	18	5	0	8	0	18
20	18	NAYARIT	8	1	3	1	18	6	1	5	1	18
21	19	NUEVO_L...	2	3	2	0	24	1	3	3	0	24
22	20	OAXACA	10	6	2	0	13	4	11	3	0	13
23	Started: 12/07/2020 17:3; elapsed time 3 min.											
24	Log-exponential trend: tau-statistic for T=76											
25	Exponential trend: tau-statistic for T=76											
26	Fractional trend: tau-statistic for T=76											
27												

Figure 4. Table `PAIRWISE_SUMMARY` generated by program *Testing_Convergence (Total)*

Matrix `PAIRWISE_MATRIX` contains details of the estimations. Its (r, s)-th element is a 4-digit figure; a digit equaling 1 means that the relevant model is valid for series y_{rs}:

```
XXXX
| | | |_ ARo (model (5) is valid)
| | |__ ARc (model (4) is valid)
| |___ Convergence (at least one version of model (3) is valid)
|____ Divergence (at least one version of model (3) is valid with "wrong" sign of δ).
```

Zero value of an element implies that no one model describes the respective series.

4.4 CDF of UR statistics with … trend

Function. These programs estimate cumulative distribution function (CDF) of the unit root test statistics (τ) under the null hypothesis of unit root for a respective version of model (3) and user-specified sample size. Program *CDF of UR statistics with Log-exponential trend.prg* do this for model (3a); program *CDF of UR statistics with Exponential trend.prg* do this for model (3b); and program *CDF of UR statistics with Fractional trend.prg* do this for model (3c). The results obtained are used to form or complement file *TauStatistics-NonlinearTrends.xls* that is used by programs *Testing_Convergence*, *Testing_Convergence*

(Batch), and *Testing_Convergence (Total)*.

For possible comparisons by user, the programs can also estimate CDF of the Dickey-Fuller statistics for equation with constant (4), with constant and trend, $\Delta y_t = \lambda y_{t-1} + \alpha + \beta t + \varepsilon_t$, and with constant and quadratic trend, $\Delta y_t = \lambda y_{t-1} + \alpha + \beta t + \xi t^2 + \varepsilon_t$.

Arguments. Arguments of the programs may follow in any order. The only optional argument is **`DF`**. If the workfile already exists on disc (when the execution of a program is continued after some break), only the filename should be specified (then keyword **`file=`** may be omitted).

The programs use the following arguments (keywords are in bold; [...] means an optional part of argument; | means a choice, i.e. 'or'; except for the filename, no blanks are tolerated within arguments):

`file=`[<*path*>]<*file name*> is the workfile name (if there are blanks in the path or/and file name, enclose the full name in quotes, e.g., `file="d:\my path\my file.wf1"`);

`T=`<*number*> is the sample size;

`N=`<*number*> is the number of replications;

`DF=`<**`yes`**|**`no`**> is the mode of estimating the Dickey-Fuller distributions; 'yes' means that they must be estimated, and 'no' means no estimating. By default (if this argument is omitted), **`DF=no`**.

Examples. Arguments

```
file=50 DF=no T=50 N=200000
```

imply that the CDF of τ-statistic for a model (3) will be estimated for sample of size 50, using 200,000 replications; no Dickey-Fuller distributions will be estimated. The results will be saved to a new EViews workfile named *50.wf1*.

If you specify

```
file="test 1" N=10000 T=100 DF=yes
```

then the program estimates CDF of τ-statistic for sample of size 100; in doing so, the program uses 10,000 replications. It also estimates the Dickey-Fuller test statistics, and stores results in file *test 1.wf1*.

Specification

```
file="D:\Projects\UR statistics.wf1"
```

means that the file *UR statistics.wf1* already exists on disk and contains incomplete work on estimation (parameters of estimation as well as the number of replications performed being stored in this file); the program will continue the estimation until reaches *N* specified at the very first start of this work.

Source data. The programs do not need source data.

Execution. At first, the programs checks arguments specified, halting the program in the case of error and reporting the error in the status line. Then the specified number (*N*) of replications is performed. In each replication, the programs generate $\{\varepsilon_t\}_{t=1,\ldots,T} \sim$ iid N(0,1) and construct a random walk $y_t = y_{t-1} + \varepsilon_t$ ($t = 2, \ldots, T$) with $y_1 = 0$. Then a relevant version of equation (3) is estimated with the use of essentially the same algorithm as described in Section 3.1 (if there is argument **`DF=yes`**, the programs also estimate the Dickey-Fuller test equations), and store the obtained value(s) of the test statistic(s), *t*-ratio of λ. While executing, the programs display the current number of replication and the estimated time left to complete the work in the status line. Having reached the specified number of replications, the programs compute CDF(s), using 1000 quantiles plus one more for probability 0.

Note that the programs do not check accordance between the numbers of quantiles and replications. Thus, having specified a small number of replications, you can obtain unsatisfactory results. Given that the number of the distribution quantiles equals 1001, the number of replications should be no less than 10,000.

It may happen that you make a mistake in arguments, while formally they are correct. In order that you can check correctness of your parameters, the programs show table `Description` that contains interpreted values of parameters (see Figure 5). When you find a mistake, you can halt the program (pushing `F1` or `Esc`), and rerun it with correct arguments. If the program has already saved its results on disc (i.e., if it has worked more than 30 minutes), delete this file from disc.

The programs can take much time to execute, several hours or even days, depending on a specified number of replications, sample size, as well as on the power of computer, operating system, and version of EViews. Therefore, the programs save the EViews workfile with results obtained every 30 minutes. Thus, if some problem occurs with your computer, you lose, at the worst, a half-hour work. To continue the work, start the program with the name of the workfile with the incomplete work as a sole argument. You may do the work in parts yourself. Push `F1` or `Esc` key to halt the program and save the relevant workfile to disc. You can continue the interrupted work at any time as above.

Results. The programs generate and save to disc an EViews workfile with a user-specified name. The main results are contained in the 1001 × 2 matrix `cdf_t_nlt` which is the CDF of the unit root test statistic for a relevant equation and specified sample size. The first column of this matrix is *p*-values from 0 through 1 with increment 0.001; the second column is values of the test statistic for the corresponding *p*-values.

If the estimation of the Dickey-Fuller statistics is specified, three additional 1001 × 2 matrices `cdf_df_c`, `cdf_df_t`, and `cdf_df_t2` with CDFs of these statistics (for the Dickey-Fiuller equation with constant, trend, and quadratic trend, respectively) are generated. They can be used, e.g., to compare a CDF of τ for a nonlinear model with the CDFs of the Dickey-Fuller statistics.

Until a program completes, it saves in the workfile additional objects that are necessary to continue the work if it has been interrupted. Do not remove or change these objects!

The programs also save raw results, i.e. all N estimated t-ratios of λ, in vector `t_nlt` (if **`DF=yes`** is specified, the programs create similar vectors `t_df_c`, `t_df_t`, and `t_df_t2`). They may prove useful for some analytical purposes, e.g., to plot the probability density function of a statistic.

The programs put on the screen and save table `Description`. It reports conditions of a given estimation, namely, the sample size and number of replications (if the work is not completed, the number of replications done is reported as well). Besides, it reports the time when program has started and finished, and elapsed time, as well as the list of arguments as they have been specified. Figure 5 shows an example.

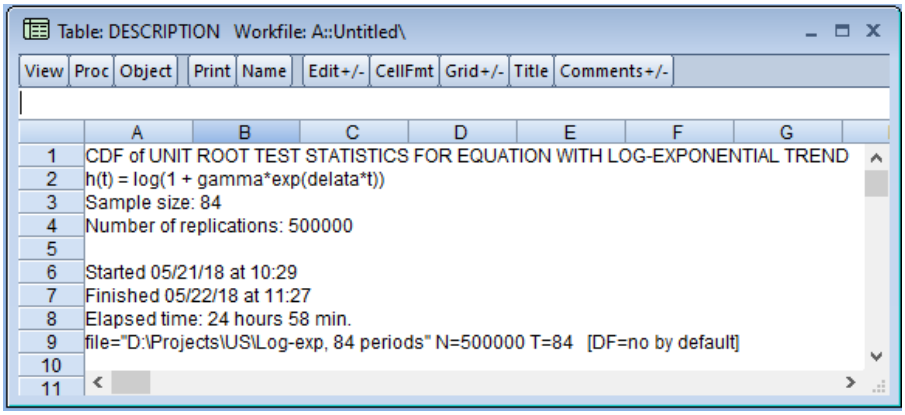


Figure 5. Table `Description` reported by program *CDF of UR statistics with Log-exponential trend*

When the file of CDFs, *TauStatistics-NonlinearTrends.xls*, is being created, copy the first column of matrix `cdf_t_nlt` and paste it starting from a relevant cell of the first column in a necessary sheet **(Log-exp**, or **Exp**, or **Fract)** of the file; see Section 5. Then copy the second column of matrix `cdf_t_nlt` and paste it from a relevant cell of the column for a given sample size in a necessary sheet of the file. You also

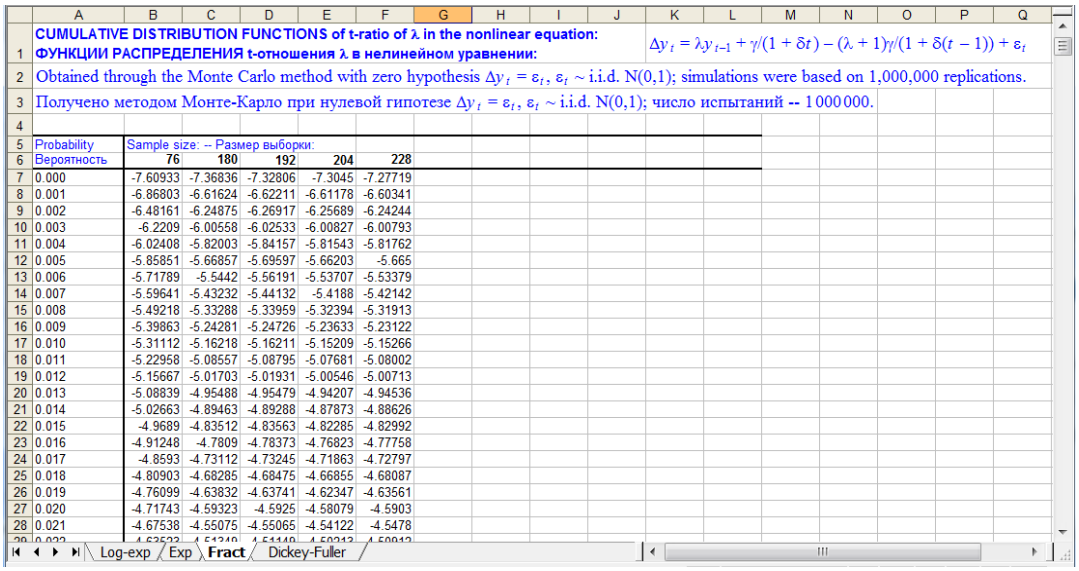
have to input the sample size, if it is not indicated. To copy, open the matrix and highlight cells in a column to be copied, from the first to the last one (and not the entire column!), press Ctrl+C, and choose option "`Unformatted - Copy numbers at highest precision`." To paste, open the necessary sheet of *TauStatistics-NonlinearTrends.xls*, locate cursor at a required position (line 7 and a proper column) and press Ctrl+V or click button Paste in the Excel menu. If you modify an existing file *TauStatistics-NonlinearTrends.xls*, you need to copy and paste only the second column of `cdf_t_nlt`.

5. The file of the unit root test statistics

This is an Excel file having a fixed name *TauStatistics-NonlinearTrends.xls*. It contains cumulative distribution functions of the unit root test statistics, τ, for equations (3a), (3b), and (3c) for different sample sizes. The data are obtained with the use of programs *CDF of UR statistics with ... trend*. This file is used by programs *Testing_Convergence*, *Testing_Convergence (Batch)*, and *Testing_Convergence (Total)* to find *p*-values of the unit root tests.

Note that these programs use the file with extension .xls, and are not able to work with the newer format .xlsx.

The CDFs are stored in sheets named **Log-exp**, **Exp**, and **Fract** intended for model (3a), (3b), and (3c), respectively. (Other sheets with arbitrary information can be included in the file). Figure 6 shows an example. The data used by the programs are in black print. The blue font marks optional comments. You may insert your own comments and store other information in this sheet outside the statistic block. Graphs are also allowable.



	A	B	C	D	E	F
1	CUMULATIVE DISTRIBUTION FUNCTIONS of t-ratio of λ in the nonlinear equation: $\Delta y_t = \lambda y_{t-1} + \gamma/(1+\delta t) - (\lambda+1)\gamma/(1+\delta(t-1)) + \varepsilon_t$					
2	Obtained through the Monte Carlo method with zero hypothesis $\Delta y_t = \varepsilon_t$, $\varepsilon_t \sim$ i.i.d. N(0,1); simulations were based on 1,000,000 replications.					
3	Получено методом Монте-Карло при нулевой гипотезе $\Delta y_t = \varepsilon_t$, $\varepsilon_t \sim$ i.i.d. N(0,1); число испытаний -- 1 000 000.					
4						
5	Probability	Sample size: -- Размер выборки:				
6	Вероятность	76	180	192	204	228
7	0.000	-7.60933	-7.36836	-7.32806	-7.3045	-7.27719
8	0.001	-6.86803	-6.61624	-6.62211	-6.61178	-6.60341
9	0.002	-6.48161	-6.24875	-6.26917	-6.25689	-6.24244
10	0.003	-6.2209	-6.00558	-6.02533	-6.00827	-6.00793
11	0.004	-6.02408	-5.82003	-5.84157	-5.81543	-5.81762
12	0.005	-5.85851	-5.66857	-5.69597	-5.66203	-5.665
13	0.006	-5.71789	-5.5442	-5.56191	-5.53707	-5.53379
14	0.007	-5.59641	-5.43232	-5.44132	-5.4188	-5.42142
15	0.008	-5.49218	-5.33288	-5.33959	-5.32394	-5.31913
16	0.009	-5.39863	-5.24281	-5.24726	-5.23633	-5.23122
17	0.010	-5.31112	-5.16218	-5.16211	-5.15209	-5.15266
18	0.011	-5.22958	-5.08557	-5.08795	-5.07681	-5.08002
19	0.012	-5.15667	-5.01703	-5.01931	-5.00546	-5.00713
20	0.013	-5.08839	-4.95488	-4.95479	-4.94207	-4.94536
21	0.014	-5.02663	-4.89463	-4.89288	-4.87873	-4.88626
22	0.015	-4.9689	-4.83512	-4.83563	-4.82285	-4.82992
23	0.016	-4.91248	-4.7809	-4.78373	-4.76823	-4.77758
24	0.017	-4.8593	-4.73112	-4.73245	-4.71863	-4.72797
25	0.018	-4.80903	-4.68285	-4.68475	-4.66855	-4.68087
26	0.019	-4.76099	-4.63832	-4.63741	-4.62347	-4.63561
27	0.020	-4.71743	-4.59323	-4.5925	-4.58079	-4.5903
28	0.021	-4.67538	-4.55075	-4.55065	-4.54122	-4.5478

Log-exp | Exp | **Fract** | Dickey-Fuller

Figure 6. Example of content of *TauStatistics-NonlinearTrends.xls*

The structure of these sheets is rigid. Lines 6 through 1007 contain data on CDFs (while other lines can contain arbitrary information, e.g., explanations, titles, etc.). Column A, lines 7 through 1007, contain *p*-values from 0 through 1 with increment 0.001. (Cell A6 may contain arbitrary information.) Further columns contain values of τ, a value in a given line corresponds to the same line of column A. Each column corresponds to some sample size which is placed in line 6 of this column. Such columns should be arranged in *ascending* sample sizes; no empty columns are permitted.

Note that the programs that use this file do not entirely check the validity of its format (the order of sample sizes, number of rows, etc.). Therefore be careful when modifying the file.

References

Bernard, A. B. and S. N. Durlauf (1995). Convergence in International Output. *Journal of Applied Econometrics*, 10 (2), 97–108.

Durlauf, S. N. and D. Quah (1999). The New Empirics of Economic Growth. In: Taylor, J. and M. Woodford, eds., *Handbook of Macroeconomics*, Vol. 1, 235–308.

Ng, S., and Perron, P. (2001). Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, 69 (6), 1519–1554..

Perron, P., and Ng, S. (1996) Useful modifications to some unit root tests with dependent errors and their local asymptotic properties. *Review of Economic Studies*, 63 (3), 435–463.

Perron, P., and Ng, S. (1998) An autoregressive spectral density estimator at frequency zero for nonstationarity tests. *Econometric Theory*, 14 (5), 560–603.

Quantitative Micro Software, LLC (2010a). *EViews 7 User's Guide I*.

Quantitative Micro Software, LLC (2010b). *EViews 7 User's Guide II*.