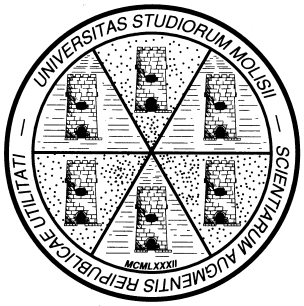
Università degli Studi del Molise
Facoltà di Economia
Dipartimento di Scienze Economiche, Gestionali e Sociali
Via De Sanctis, I-86100 Campobasso (Italy)



ECONOMICS & STATISTICS DISCUSSION PAPER
No. 51/09

Covariate Augmented Dickey-Fuller Tests with R

by

Claudio Lupi
University of Molise, Dept. SEGeS

Covariate Augmented Dickey-Fuller Tests with R

Claudio Lupi
University of Molise

Abstract

This paper describes **CADFtest**, a R (R Development Core Team 2008) package for testing for the presence of a unit root in a time series using the Covariate Augmented Dickey-Fuller (CADF) test proposed in Hansen (1995). The procedures presented here are user friendly, allow fully automatic model specification, and allow computation of the asymptotic p-values of the test.

Keywords: unit root, stationary covariates, asymptotic p-values, R.

1. Introduction and statistical background

Testing for unit roots is a frequent problem in macroeconomic modeling and forecasting. However, many commonly used tests have low power (see e.g. Campbell and Perron 1991; DeJong, Nankervis, Savin, and Whiteman 1992b,a; Phillips and Xiao 1998). In recent years, an important strand of research has been devoted to finding unit root tests with better power properties (see Haldrup and Jansson 2006, for a recent survey of some of these approaches).

The strategy suggested in a seminal paper by Hansen (1995) in order to increase the power of unit root tests is to consider stationary covariates in an otherwise conventional Augmented Dickey-Fuller (ADF) framework (Dickey and Fuller 1979, 1981; Said and Dickey 1984). The intuition lying behind the approach is very simple: if the stationary covariates have explanatory power within the ADF equation, then the regression parameters will be more precisely estimated and the test statistic will be more powerful. Indeed, also the estimate of the autoregressive parameter can be affected by the inclusion of the stationary covariates (see Caporale and Pittis 1999; Costantini, Lupi, and Popp 2007). More precisely, Hansen (1995, p. 1152) shows that, unless the variable of interest is independent of the stationary covariates, the most powerful test makes use of the information embodied in the stationary covariates themselves. As a consequence, not considering the covariates leads to a loss in the power of the test.

Formally, Hansen (1995) considers a univariate time series y_t composed of a deterministic and a stochastic component such that

$$\begin{aligned} y_t &= d_t + s_t && (1)\\ a(L)\Delta s_t &= \delta s_{t-1} + v_t && (2)\\ v_t &= b(L)'\left(\Delta x_t - \mu_{\Delta x}\right) + e_t && (3) \end{aligned}$$

where d_t is the deterministic term (usually a constant or a constant and a linear trend), $a(L) := (1 - a_1 L - a_2 L^2 - \ldots - a_p L^p)$ is a polynomial in the lag operator L, Δx_t is a vector of *stationary* covariates, $\mu_{\Delta x} := \mathsf{E}(\Delta x)$, $b(L) := (b_{q_2} L^{-q_2} + \ldots + b_{q_1} L^{q_1})$ is a polynomial where

both leads and lags are allowed for. Furthermore, the long-run (zero-frequency) covariance matrix

$$\Omega := \sum_{k=-\infty}^{\infty} \mathsf{E}\left[\begin{pmatrix} v_t \\ e_t \end{pmatrix} \begin{pmatrix} v_{t-k} & e_{t-k} \end{pmatrix}\right] = \begin{pmatrix} \omega_{vv} & \omega_{ve} \\ \omega_{ve} & \omega_{ee} \end{pmatrix} \tag{4}$$

can be defined on which the long-run squared correlation between v_t and e_t can be derived as

$$\rho^2 := \frac{\omega_{ve}^2}{\omega_{vv}\,\omega_{ee}}\,. \tag{5}$$

When Δx_t has no explicative power on the long-run movement of v_t, then $\rho^2 \approx 1$. On the contrary, when Δx_t explains nearly all the zero-frequency variability of v_t, then $\rho^2 \approx 0$. The case $\rho^2 = 0$ is ruled out for technical reasons (see Hansen 1995, p. 1151).[1]

As with the ADF test (Said and Dickey 1984), Hansen's Covariate-Augmented Dickey-Fuller (CADF) test is based on different models, according to the different deterministic kernels that the investigator may wish to consider:

$$a(L)\Delta y_t = \delta y_{t-1} + b(L)'\Delta x_t + e_t\,; \tag{6}$$

$$a(L)\Delta y_t = \mu + \delta_\mu y_{t-1} + b(L)'\Delta x_t + e_t\,; \tag{7}$$

$$a(L)\Delta y_t = \mu^* + \theta\, t + \delta_\tau y_{t-1} + b(L)'\Delta x_t + e_t\,. \tag{8}$$

Similarly to the conventional ADF test, the CADF test is based on the t-statistic for δ, $\widehat{t(\delta)}$, with the null hypothesis being that a unit root is present, i.e. $H_0:\ \delta = 0$ against the one-sided alternative $H_1:\ \delta < 0$. Hansen (1995) refers to the test statistic derived from (6)-(8) as the CADF(p, q_1, q_2) statistic.

Hansen (1995, p. 1154) proves that under the unit-root null, $\widehat{t(\delta)}$ in (6) is such that

$$\widehat{t(\delta)} \Rightarrow \rho\,\frac{\int_0^1 W\,dW}{\left(\int_0^1 W^2\right)^{1/2}} + \left(1-\rho^2\right)^{1/2} N(0,1) \tag{9}$$

where $\Rightarrow$ denotes weak convergence, W is a standard Wiener process, and $N(0,1)$ is a standard normal independent of W.[2] It is interesting to note that (9) is the distribution of a weighted sum of a Dickey-Fuller and a standard normal random variable. If a model with constant ($\widehat{t(\delta_\mu)}$) or a model with constant and trend ($\widehat{t(\delta_\tau)}$) are considered, the standard Wiener process W in (9) has to be replaced by a demeaned or a detrended Wiener process, respectively. Note that the asymptotic distribution of the test statistic depends on the nuisance parameter ρ^2 but, provided ρ^2 is given, it can be simulated using standard techniques.[3]

Hansen (1995, p. 1155) provides the asymptotic critical values of the test. However, while procedures are readily available to compute the p-values from standard ADF tests,[4] no procedure is available to compute the p-values of Hansen's CADF test.[5] In many cases this can

[1] This excludees that the variable to be tested is cointegrated with the cumulated covariate(s).

[2] The asymptotic distribution of the test statistic is derived under conventional weak dependence and moment restrictions (see Hansen 1995, p. 1151).

[3] See, e.g., Hatanaka (1996).

[4] For example, p-values of the standard ADF test can be computed using the function `punitroots()` implemented in the R package **fUnitRoots** (Wuertz 2008).

[5] To the best of our knowledge, there are only two other procedures that compute Hansen's CADF test that have been witten by Bruce Hansen himself in Gauss and Matlab. These procedures can be freely downloaded from http://www.ssc.wisc.edu/~bhansen/progs/et_95.html. However, Hansen's code does not allow automatic model identification and the computation of the p-value of the test.

be a serious limitation to the application of the test. This is especially true if meta-analytic techniques have to be applied that combine p-values from individual tests (see e.g. Costantini *et al.* 2007).

Indeed, Elliott and Jansson (2003) show that Hansen's CADF test is not the point optimal test and that it can be "nearly efficient" only in the absence of deterministic terms. Feasible point optimal tests in the presence of deterministic components are developed in Elliott and Jansson (2003). However, Monte Carlo simulations reported in Elliott and Jansson (2003) show that power gains can be obtained with respect to the CADF test at the cost of slightly worse size performances.

In this paper we present the R (R Development Core Team 2008) package **CADFtest** that allows to perform the unit root CADF test easily.[6] The main test function, `CADFtest()`, computes the CADF test in a very flexible way, allowing the user to apply different criteria and even to select the model to be used from within a set of models via the AIC or the BIC information criterion automatically. The main function `CADFtest()` returns a `CADFtest` class object that not only contains the test statistic, but also its asymptotic p-value and many other useful details. In fact, the class `CADFtest` inherits from the class `htest`,[7] so that no special `print()` method is needed. However, a dedicated `summary()` method has been developed to report the detailed test results.

The algorithm to compute the p-values, implemented in the function `CADFpvalues()`, has been originally proposed in Costantini *et al.* (2007) and it is described here in more detail, along with the instructions on how to use the function `CADFpvalues()` as a stand-alone function. In fact, the function `CADFpvalues()` can also be used separately from the main function `CADFtest()`.

The remainder of this paper is structured as follows: Section 2 discusses the way the CADF test has been implemented in the function `CADFtest()`, also illustrating some applications. In Section 3, the method to compute the asymptotic p-values is illustrated in detail along with the use of the function `CADFpvalues()`. A summary is offered in Section 4.

2. Implementation and use of the function `CADFtest()`

In principle, carrying out a CADF test is no more complicated than carrying out an ordinary ADF test. What makes things more complicated is the presence of the nuisance parameter ρ^2 in the asymptotic distribution (9). In fact, a consistent estimate of ρ^2 has to be used to choose the correct asymptotic critical value and/or to compute the correct asymptotic p-value of the test. The problem is solved into two steps. First, $\hat{e}_t$ and $\hat{v}_t$ are derived; then, their long-run covariance matrix Ω is estimated using a HAC covariance estimator (see e.g. Andrews 1991; Zeileis 2004, 2006; Kleiber and Zeileis 2008).

Once the kind of model (no constant, with constant, with constant and trend) has been chosen, using `CADFtest()` the investigator can either set the polynomial orders p, q_2 and q_1 to fixed values, or can decide the maximum value for each and let the procedure to select and estimate the model according to the AIC or the BIC.

In order to estimate ρ^2 it is necessary to estimate e_t and v_t first. For example, if the model

[6] The present paper describes version 0.1-0 of the package. Other versions of the package are planned which will include further extensions and refinements.

[7] A fairly detailed description of the `htest` class can be gathered from within R by typing `?t.test`.

with constant and trend (8) is used, then e_t and v_t are estimated as

$$\hat{e}_t = \widehat{a(L)}\Delta y_t - \hat{\mu}^* - \hat{\theta}\, t - \hat{\delta}_\tau y_{t-1} - \widehat{b(L)}'\Delta x_t \quad (10)$$

$$\hat{v}_t = \widehat{b(L)}'\left(\Delta x_t - \overline{\Delta x}\right) + \hat{e}_t \quad (11)$$

where "$\widehat{\ }$" denote parameters estimated by ordinary least squares and $\overline{\Delta x}$ is the sample average of Δx_t. Once $\hat{e}_t$ and $\hat{v}_t$ have been computed, a kernel-based HAC covariance estimator (Andrews 1991) is used to estimate Ω and hence ρ^2. In order to estimate ρ^2 in a rather flexible way, in `CADFtest()` the `kernHAC()` function included in the **sandwich** package (Zeileis 2004, 2006) is used. This allows the investigator to chose the kernel to be applied and if and how prewhitening should be performed. A Parzen kernel without prewhitening is the default choice as in Hansen (1995). The bandwidth is always adaptively chosen using the method proposed in Andrews (1991).

The usage of the function is extremely simple:

```
CADFtest(model, X=NULL, trend=c("c", "nc", "ct", "none", "drift", "trend"),
         data=list(), max.lag.y=1, min.lag.X=0, max.lag.X=0, dname="",
         Auto=FALSE, criterion=c("BIC", "AIC"), prewhite=FALSE,
         kernel = c("Parzen", "Quadratic Spectral", "Truncated",
         "Bartlett", "Tukey-Hanning"))
```

The minimal required input is `CADFtest(y)`, where `y` can be either a vector or a time series. However, if no stationary covariate is specified, an ordinary ADF test is performed. In fact, the ordinary ADF test can be carried out with R using other existing packages such as **fUnitRoots** (Wuertz 2008), **tseries** (Trapletti and Hornik 2008), **urca** (Pfaff 2008), **uroot** (López-de Lacalle and Díaz-Emparanza 2005). In this respect there is no need to add one further package. However, given that the ADF test can be seen as a particular case of the more general CADF test, it seems logical to leave the user the opportunity to carry out both tests in the same framework, using the same conventions and allowing for the computation of the test p-values. In fact, as far as the computation of the p-values for the ADF case is concerned, `CADFtest()` exploits the facilities offered by `punitroots()` implemented in the package **fUnitRoots** that uses the method proposed in MacKinnon (1994, 1996). In principle, it would have been possible (and easy) also to use the function `CADFpvalues` described in the next section, but given that MacKinnon (1996) describes what is probably the state-of-the-art of the computation of the ADF test p-values, it seems fair to refer directly to a function that implements this procedure. On the other hand, the package **tseries** allows instead the computation of the p-values of the ADF test using a simpler method based on the interpolation of the empirical cumulative distribution of the test statistic reported in Table 4.2 of Banerjee, Dolado, Galbraith, and Hendry (1993).

If a proper CADF test has to be performed, at least a stationary covariate should be passed to the procedure. The covariates are passed in a very simple way, using a formula (model) statement. For example, suppose we want to test the variable y using x_1 and x_2 as stationary covariates: if the other default options are accepted, then it is sufficient to specify `CADFtest(y ~ x1 + x2)`. Note that the formula that is passed as argument to the `CADFtest()` function is not the complete model to be used, but it just indicates which variable has to be tested for a unit root (`y`) and which are to be used as stationary covariates in the test (`x1` and `x2`).

The following arguments specify the deterministic kernel to be used in the model, the lead and lag orders, and if the model has to be selected using the AIC or the BIC information criterion. In particular, `trend` specifies the deterministic components to be used in the model, using the conventions utilized either in the R package **urca** (Pfaff 2008) or in the package **fUnitRoots** (Wuertz 2008). The default value (`trend="c"`) implies that the model with constant is used. The same result can be obtained by specifying `trend="drift"`. If `trend="nc"` or `trend="none"` is specified, then the model without constant is used. In order to use the model with constant and trend, `trend="ct"` or `trend="trend"` must be passed to the procedure. `max.lag.y` corresponds to p, the lag order of $a(L)$ in (6)-(8), and it is set to 1 by default: it can be set equal to 0 or to a positive integer. `min.lag.X` corresponds to q_2, the maximum lead in $b(L)$ in (6)-(8), and it is equal to 0 by default: if modified, it must be set equal to a negative integer (a negative lag is a lead). `max.lag.X` correspond to q_1, the maximum lag in $b(L)$ in (6)-(8), and the default choice is 0: if modified, it must be set equal to a positive integer. `data` is the data set to be used and `dname` is the name of data: in general there is no need to change `dname`, given that it is automatically computed by the function itself. `Auto=FALSE` indicates that no model search is performed. If `Auto` is set to `TRUE`, then all the models included in the maximum and minimum lag orders are estimated and the final model to be used is selected on the basis of the chosen `criterion` that can be either `criterion="BIC"` (the default) or `criterion="AIC"`. The remaining two arguments are used to select the kernel to be used in the HAC covariance estimation. These two arguments are the same as in the package **sandwich** and can take all the values allowed for in `kernHAC()` (see the package **sandwich**: Zeileis 2004, 2006). In fact, though the default is `kernel="Parzen"`, the kernel can be any of `"Quadratic Spectral"`, `"Truncated"`, `"Bartlett"`, `"Parzen"`, `"Tukey-Hanning"`. Also `prewhite` is defined as in `kernHAC()`. The default (`prewhite=FALSE`) is that no prewhitening is performed. Alternatively, `prewhite` can either be set to `prewhite=TRUE`, in which case a VAR(1) prewhitening is used, or it can be set to a positive number, in which case a VAR of order `as.integer(prewhite)` is used for prewhitening.

The function `CADFtest()` returns an object of class `CADFtest` containing the test statistic (`statistic`), the p-value of the test (`p.value`), the lag structure of the model (`max.lag.y`, `min.lag.X`, `max.lag.X`), the value of the information criteria (`AIC`, `BIC`), the estimated value of ρ^2 (`parameter`), and the full estimated model (`est.model`). Other information returned regards the nature of the test (either CADF or ADF) stored in `method`, the name of data used (`data.name`), the value of δ under the null (`null.value`), the description of the alternative (`alternative`) and the estimated value of δ (`estimate`).

A summary of the test can be obtained just by using a `print()` command. Given that the class `CADFtest` inherits from the class `htest`, the `print()` command produces the standard R output of the `htest` class. However, the `summary()` command is also allowed that returns a more detailed account of the test results.

We provide here a simple example of application of the function `CADFtest()`. Data are taken from the R package **urca** (Pfaff 2008) and refer to the extended Nelson and Plosser (1982) data set used in Schotman and Van Dijk (1991). These are the same data used in Hansen (1995), so that we will be able to replicate one of the empirical applications proposed there.

First, we load the required package **CADFtest** using the usual `library()` command. All the following examples assume that **CADFtest** has been loaded.

```
> library(CADFtest)
```

The data are easily loaded from the package **urca**

```
> data(npext, package = "urca")
```

A complete description of the data can be retrieved simply by typing `?npext` in R.

We replicate the analysis carried out in Hansen (1995, p. 1165) by testing first for the presence of a unit root in the log per capita US real GNP using a standard ADF test with constant and trend and three lags (the convention `trend="trend"` of **urca** is used here, but we could have used `trend="ct"` as in **fUnitRoots** instead):

```
> ADFt <- CADFtest(npext$gnpperca, max.lag.y = 3, trend = "trend")
```

The p-value of the test is stored in `ADFt$p.value` and it is easily accessible:

```
> ADFt$p.value

[1] 0.07292127
```

Here a standard Dickey-Fuller test is performed so that equation (8) is simply estimated using the package **dynlm** (Zeileis 2008) and the test p-value is computed using `punitroots()` implemented in the package **fUnitRoots** (Wuertz 2008).[8] We cannot reject the null of the presence of a unit root in the log-real GNP series using the standard 5% significance level, although we can reject the null at the 10% level.

Even if all the results are readily accessible, a summary of the test can be obtained just by typing

```
> print(ADFt)

        ADF test

data:  npext$gnpperca
ADF(3) = -3.2606, p-value = 0.07292
alternative hypothesis: true delta is less than 0
sample estimates:
     delta
-0.2014652
```

The function correctly warns the user that a conventional ADF test has been performed and reports the main results along with the number of lags used in the test.

If we want to obtain a more detailed summary that includes the details of the estimated model, we can just type

```
> summary(ADFt)
```

[8] This function is directly derived from MacKinnon (1994, 1996) and returns *finite sample*, rather than asymptotic, p-values.

```
Covariate-Augmented Dickey Fuller (CADF) test.
No covariate used in the analysis: standard ADF is performed.
Test statistic t:                        -3.2606
Test p-value:                             0.0729
Max lag of the diff. dependent variable: 3

Call:
dynlm(formula = formula(model))

Residuals:
      Min        1Q    Median        3Q       Max
-0.163620 -0.025697  0.007439  0.026647  0.147798

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.201825   0.370695   3.242  0.00182
t            0.004016   0.001203   3.339  0.00135
L(y, 1)     -0.201465   0.061788  -3.261  0.00172
L(d(y), 1)   0.391840   0.110751   3.538  0.00072
L(d(y), 2)   0.060429   0.119135   0.507  0.61358
L(d(y), 3)  -0.052543   0.115921  -0.453  0.65176

Residual standard error: 0.05309 on 70 degrees of freedom
Multiple R-squared: 0.2586,        Adjusted R-squared: 0.2057
F-statistic: 4.884 on 5 and 70 DF,  p-value: 0.0006861
```

The model output uses the same conventions utilized in the package **dynlm** (Zeileis 2008): `t` is the deterministic trend, `L(y, 1)` stands for y_{t-1} and `L(d(y), i)` represents Δy_{t-i}.

In order to replicate the analysis developed in Hansen (1995), we need now to carry out a few data transformations:

```
> npext$unemrate <- exp(npext$unemploy)
> L <- ts(npext, start = 1860)
> D <- diff(L)
> S <- window(ts.intersect(L, D), start = 1909)
```

Data are now interpreted as annual time series starting in 1860. The sample ends in 1988 (this is easy to verify by invoking the `tsp()` function). Given that `unemploy` is the log of the unemployment rate, while we need the unemployment rate, the series in levels used by Hansen (1995) is computed. The time series in levels are stored in `L`, while `D` stores the first differences of the original variables, that will be used as stationary covariates in the CADF tests. `S` contains all the series over a common sample that starts in 1909, as in Hansen (1995).

We now run Hansen's CADF test on the log-real GNP per capita by using the first difference of the unemployment rate as stationary covariate. The test is carried out with constant and trend and allowing 3 lags for the (differences of the) dependent variable and 0 lags of the covariate, without automatic model selection and using the default Parzen kernel. This is the same model as the one reported in Hansen (1995, Table 8, column 2, p. 1166).

```
> CADFt <- CADFtest(L.gnpperca ~ D.unemrate, data = S, trend = "ct",
+     max.lag.y = 3)
> print(CADFt)

        CADF test

data:  L.gnpperca ~ D.unemrate
CADF(3,0,0) = -3.413, rho2 = 0.064, p-value = 0.001729
alternative hypothesis: true delta is less than 0
sample estimates:
      delta
-0.08720302
```

Consistently with Hansen's results (see Hansen 1995, Table 8, column 2, p. 1166), the estimated value of ρ^2 is very low and we can now strongly reject the null. However, differently from Hansen (1995), we not only verify that the test is significant at the asymptotic 1% level, but we can also give a precise assessment of the test p-value. From the detailed results we can also easily check that the stationary covariate is highly significant:

```
> summary(CADFt)

Covariate-Augmented Dickey Fuller (CADF) test.
Test statistic t:                         -3.413
Estimated rho^2:                           0.0635
Test p-value:                              0.0017
Max lag of the diff. dependent variable: 3
Max lag of the stationary covariate(s):  0
Max lead of the stationary covariate(s): 0

Call:
dynlm(formula = formula(model))

Residuals:
      Min        1Q    Median        3Q       Max
-0.096966 -0.010025  0.001116  0.010832  0.037384

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.6142646  0.1766271   3.478 0.000880
t            0.0017930  0.0004973   3.605 0.000584
L(y, 1)     -0.0872030  0.0255505  -3.413 0.001079
L(d(y), 1)  -0.0117603  0.0492906  -0.239 0.812130
L(d(y), 2)   0.1658117  0.0482098   3.439 0.000993
L(d(y), 3)  -0.0675385  0.0466073  -1.449 0.151840
L(X, 0)     -0.0211035  0.0011059 -19.083  < 2e-16

Residual standard error: 0.02134 on 69 degrees of freedom
```

```
Multiple R-squared: 0.8819,        Adjusted R-squared: 0.8716
F-statistic: 85.88 on 6 and 69 DF,  p-value: < 2.2e-16
```

Besides the CADF(3,0,0) test, Hansen's original analysis includes some other CADF tests, namely the CADF(3,2,0), CADF(3,2,2), CADF(3,0,2). Instead of using different tests in this way, we rather specify the maximum lead and lag, and leave the model to be selected by using the BIC:

```
> CADFt <- CADFtest(L.gnpperca ~ D.unemrate, data = S, max.lag.y = 3,
+     max.lag.X = 3, min.lag.X = -3, Auto = TRUE, criterion = "BIC",
+     trend = "ct")
> print(CADFt)

        CADF test

data:  L.gnpperca ~ D.unemrate
CADF(0,2,0) = -4.4157, rho2 = 0.012, p-value = 8.114e-05
alternative hypothesis: true delta is less than 0
sample estimates:
     delta
-0.1023180
```

The selected model is CADF(0,2,0) and clearly rejects the null of a unit root for any reasonable confidence level.

The test can be easily carried out using more than just one covariate. In fact, we can easily perform a more general test by using two stationary covariates and leaving the model to be selected by the BIC, after the maximum lag orders have been again set to 3:

```
> CADFt <- CADFtest(L.gnpperca ~ D.unemrate + D.indprod, data = S,
+     max.lag.y = 3, min.lag.X = -3, max.lag.X = 3, Auto = TRUE,
+     criterion = "BIC", trend = "ct")
> print(CADFt)

        CADF test

data:  L.gnpperca ~ D.unemrate + D.indprod
CADF(0,2,0) = -4.0793, rho2 = 0.002, p-value = 0.0001236
alternative hypothesis: true delta is less than 0
sample estimates:
     delta
-0.0850728
```

The two covariates explain a great deal of the zero-frequency variation of the dependent variable (see the value of $\hat{\rho}^2$). The p-value of the test is again extremely low and we can reject the null at any reasonable significance level.

In the next section we describe in detail how the asymptotic p-values of the test are computed.

3. P-values computation and the function `CADFpvalues()`

The possibility of computing the p-values of a test greatly increases the chances that the test is effectively used by practitioners. This is *a fortiori* true when the test procedure requires the use of non-standard tables available only in few specialized papers. There are even instances where computation of the p-values is necessary for further investigations, as is the case for some panel unit root tests (see e.g. Maddala and Wu 1999; Choi 2001; Costantini *et al.* 2007). The R function `CADFpvalues()` presented here allows the computation of asymptotic p-values of the CADF test proposed in Hansen (1995). `CADFpvalues()` is used within the `CADFtest()` function to compute the p-values of the test along with the other test results already discussed. However, `CADFpvalues()` can also be used separately from the main testing procedure.

The method used to compute the p-values is similar to that proposed in MacKinnon (1994, 1996) for the p-values of the ADF test and has been proposed in Costantini *et al.* (2007). Differently to what happens with reference to the Dickey-Fuller distribution, the asymptotic distribution of the CADF test statistic depends on the nuisance parameter $0 < \rho^2 \leq 1$, so that the asymptotic distribution (9) has to be simulated over a grid of values for ρ^2. In order to obtain fairly good approximations, here a grid of 40 values $\rho^2 \in \{0.025, 0.050, 0.0725, \ldots, 1\}$ is considered.

For each of the three models (6)-(8), 100,000 replications[9] have been used for each value of ρ^2. The Wiener functionals have been simulated using a standard approach (see e.g. Hatanaka 1996, p. 67) with $T = 5,000$ (for the "no constant", "constant" and "constant plus trend" case, standard, demeaned and detrended Wiener processes have been used). On the basis of the simulated values, for each value of ρ^2 1,005 asymptotic quantiles q_ρ are derived corresponding to the probabilities $p =$ (0.00025, 0.00050, 0.00075, ..., 0.001, 0.002, ..., 0.998, 0.999, 0.99925, 0.99950, 0.99975). As a result, we obtain a $1,005 \times 40$ matrix of estimated quantiles. Along the rows of the matrix it is possible to read how a given quantile varies with ρ^2. Indeed, the estimated quantiles vary very smoothly with ρ^2 (see Costantini *et al.* 2007).

For each row of the quantile matrix the model

$$q_\rho(p) = \beta_0 + \beta_1\rho^2 + \beta_2\left(\rho^2\right)^2 + \beta_3\left(\rho^2\right)^3 + \epsilon \tag{12}$$

is estimated and the $\hat{\beta}$'s are saved in a $1,005 \times 4$ table. Tables of estimated coefficients for the "no constant", "constant" and "constant plus trend" case, respectively are used by the function `CADFpvalues()` in order to compute the asymptotic p-values for any value of $0 < \rho^2 \leq 1$ for the relevant model.

The way the computation of the p-values proceeds in `CADFpvalues()` is essentially the following:

1. The relevant table of parameters is read, depending on the specific model used ("no constant", "constant" or "constant plus trend").

2. For any desired value ρ_0^2 of ρ^2, the estimated parameters are used to compute for all the 1,005 probability values p the fitted quantiles $\widehat{q_{\rho_0}}(p)$ as

$$\widehat{q_{\rho_0}}(p) = \widehat{\beta_0} + \widehat{\beta_1}\rho_0^2 + \widehat{\beta_2}\left(\rho_0^2\right)^2 + \widehat{\beta_3}\left(\rho_0^2\right)^3 . \tag{13}$$

[9] Simulations have been carried out using R.

3. The approach suggested in MacKinnon (1994, 1996) can now be used on $\widehat{q_{\rho_0}}$ to derive the p-value. First, given the value $\widehat{t(\delta)}$ of the test statistic, it is necessary to find the fitted quantile $\widehat{q_{\rho_0}}$ that is closest to $\widehat{t(\delta)}$ and the corresponding probability $\tilde{p}$.

4. The regression

$$\Phi^{-1}(p) = \gamma_0 + \gamma_1 \widehat{q_{\rho_0}}(p) + \gamma_2 \widehat{q_{\rho_0}}^2(p) + \gamma_3 \widehat{q_{\rho_0}}^3(p) + \nu_p \tag{14}$$

   where $\Phi^{-1}(p)$ is the inverse of the cumulative standard normal distribution is estimated locally on an interval of p centered on $\tilde{p}$. In `CADFpvalues()` local interpolation takes place using 11 values centered on $\tilde{p}$.

5. The p-value associated with the estimated test statistic $\widehat{t(\delta)}$ is finally obtained from

$$\Phi\left(\hat{\gamma}_0 + \hat{\gamma}_1 \widehat{t(\delta)} + \hat{\gamma}_2 \widehat{t(\delta)}^2 + \hat{\gamma}_3 \widehat{t(\delta)}^3\right). \tag{15}$$

The usage of the function is extremely simple:

```
CADFpvalues(t0, rho2=0.5, trend="c")
```

where `t0` is the value of the test statistic $\widehat{t(\delta)}$, `rho2` is the estimated value of ρ^2, and `trend` assumes the values `"nc"` (or `"none"`), `"c"` (or `"drift"`), and `"ct"` (or `"trend"`) as above when a model without constant, with constant, or with constant plus trend is considered.

For example, suppose that we want to know the p-values of the tests reported in Hansen (1995, Table 10). The tests are carried out using models with constant and trend. Specifically, consider the CADF(3,0,0) and CADF(3,2,0) whose test statistics are -2.2 and -1.7, with $\hat{\rho}^2$ equal to 0.53 and 0.20, respectively. The computation of the p-values of these tests is immediate:

```
> CADFpvalues(t0 = -2.2, rho2 = 0.53, trend = "ct")

          [,1]
[1,] 0.2447352

> CADFpvalues(t0 = -1.7, rho2 = 0.2, trend = "ct")

          [,1]
[1,] 0.2189253
```

It is now clear that both tests do not reject the null.

If desired, `CADFpvalues()` can be used also to compute the asymptotic p-values of the ordinary ADF test. In fact, it is sufficient to set `rho2=1` to obtain the p-values of the Dickey-Fuller distribution. For example

```
> CADFpvalues(-0.44, trend = "drift", rho2 = 1)
```

```
          [,1]
[1,] 0.9018844
```

computes a p-value that can be compared directly with the values reported in Table 4.2 in Banerjee *et al.* (1993).

4. Summary

This paper presents the R package **CADFtest** that allows unit root testing using the Covariate-Augmented Dickey Fuller (CADF) test advocated in Hansen (1995).

Differently from the already available routines written in Gauss and in Matlab (dowloadable from Bruce Hansen's home page at http://www.ssc.wisc.edu/~bhansen/progs/et_95.html), the present functions are easy to use, do not require the user to modify the programs, and allow the computation of the asymptotic p-values of the tests. Beside being extremely useful in general, p-values computation opens to the possibility of using the CADF tests in unit root combination tests, for example in the context of macro panels (see e.g. Maddala and Wu 1999; Choi 2001; Costantini *et al.* 2007).

CADFtest can be downloaded from the Comprehensive R Archive Network (CRAN) at http://CRAN.r-project.org/package=CADFtest.

Computational details

The functions illustrated in this paper use the following R (R Development Core Team 2008) packages, listed in alphabetical order:

- **dynlm**: Zeileis (2008)
- **fUnitRoots**: Wuertz (2008)
- **sandwich**: Zeileis (2004, 2006)

Acknowledgements

I would like to thank Achim Zeileis for his many comments and suggestions that helped me in improving on previous versions of the package. I am grateful to Mauro Costantini and Stephan Popp for comments and discussion. Of course, none of them is responsible for any remaining error. I owe a special thank to the authors of the R packages used in the development of the functions described in this paper.

This text was typeset using MiKTEX(Schenk 2009), R (R Development Core Team 2008) and `Sweave()` (Leisch 2002).

References

Andrews DWK (1991). "Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation." *Econometrica*, **59**(3), 817–858.

Banerjee A, Dolado J, Galbraith JW, Hendry DF (1993). *Co-Integration, Error-Correction, and the Econometric Analysis of Non-Stationary Data*. Advanced Texts in Econometrics. Oxford University Press, Oxford.

Campbell J, Perron P (1991). "Pitfalls and Opportunities:What Macroeconomists Should Know about Unit Roots." In O Blanchard, S Fischer (eds.), "NBER Macroeconomics Annual," volume 6. MIT Press, Cambridge, MA.

Caporale GM, Pittis N (1999). "Unit Root Testing Using Covariates: Some Theory and Evidence." *Oxford Bulletin of Economics and Statistics*, **61**(4), 583–595.

Choi I (2001). "Unit Root Tests for Panel Data." *Journal of International Money and Finance*, **20**(2), 249–272.

Costantini M, Lupi C, Popp S (2007). "A Panel-CADF Test for Unit Roots." *Economics & Statistics Discussion Paper 39/07*, University of Molise. URL http://econpapers.repec.org/paper/molecsdps/esdp07039.htm.

DeJong DN, Nankervis JC, Savin NE, Whiteman CH (1992a). "Integration Versus Trend Stationary in Time Series." *Econometrica*, **60**(2), 423–433.

DeJong DN, Nankervis JC, Savin NE, Whiteman CH (1992b). "The Power Problems of Unit Root Tests in Time Series with Autoregressive Errors." *Journal of Econometrics*, **53**(1-3), 323–343.

Dickey DA, Fuller WA (1979). "Distributions of the Estimators for Autoregressive Time Series With a Unit Root." *Journal of the American Statistical Association*, **74**(366), 427–431.

Dickey DA, Fuller WA (1981). "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root." *Econometrica*, **49**(4), 1057–1072.

Elliott G, Jansson M (2003). "Testing for Unit Roots With Stationary Covariates." *Journal of Econometrics*, **115**(1), 75–89.

Haldrup N, Jansson M (2006). "Improving Size and Power in Unit Root Testing." In TC Mills, K Patterson (eds.), "Econometric Theory," volume 1 of *Palgrave Handbook of Econometrics*, chapter 7, pp. 252–277. Palgrave MacMillan, Basingstoke.

Hansen BE (1995). "Rethinking the Univariate Approach to Unit Root Testing: Using Covariates to Increase Power." *Econometric Theory*, **11**(5), 1148–1171.

Hatanaka M (1996). *Time-Series-Based Econometrics: Unit Roots and Cointegration*. Advanced Texts in Econometrics. Oxford University Press, Oxford.

Kleiber C, Zeileis A (2008). *Applied Econometrics with R*. Use R! Springer, New York.

Leisch F (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), "Compstat 2002 — Proceedings in Computational Statistics," pp. 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9, URL http://www.stat.uni-muenchen.de/~leisch/Sweave.

López-de Lacalle J, Díaz-Emparanza I (2005). *uroot: Unit Root Tests and Graphics for Seasonal Time Series*. R package version 1.4.

MacKinnon JG (1994). "Approximate Asymptotic Distribution Functions for Unit-Root and Cointegration Tests." *Journal of Business and Economic Statistics*, **12**(2), 167–176.

MacKinnon JG (1996). "Numerical Distribution Functions for Unit Root and Cointegration Tests." *Journal of Applied Econometrics*, **11**(6), 601–618.

Maddala GS, Wu S (1999). "A Comparative Study of Unit Root Tests with Panel Data and a New Simple Test." *Oxford Bulletin of Economics and Statistics*, **61**(Supplement 1), 631–652.

Nelson CR, Plosser CR (1982). "Trends and Random Walks in Macroeconmic Time Series: Some Evidence and Implications." *Journal of Monetary Economics*, **10**(2), 139–162.

Pfaff B (2008). *Analysis of Integrated and Cointegrated Time Series with R*. Use R! Springer, New York, second edition. URL http://www.pfaffikus.de.

Phillips PCB, Xiao Z (1998). "A Primer on Unit Root Tesing." *Journal of Economic Surveys*, **12**(5), 423–470.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Said SE, Dickey DA (1984). "Test for Unit Roots in Autoregressive-Moving Average Models of Unknown Order." *Biometrika*, **71**(3), 599–607.

Schenk C (2009). "MiKTeX Project Page." URL http//:miktex.org.

Schotman PC, Van Dijk HK (1991). "On Bayesian Routes to Unit Roots." *Journal of Applied Econometrics*, **6**(4), 387–401.

Trapletti A, Hornik K (2008). *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-16., URL http://CRAN.R-project.org/.

Wuertz D (2008). *fUnitRoots: Rmetrics - Trends and Unit Roots*. R package version 260.74, URL http://www.rmetrics.org.

Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software*, **11**(10), 1–17. URL http://www.jstatsoft.org/v11/i10/.

Zeileis A (2006). "Object-oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. URL http://www.jstatsoft.org/v16/i09/.

Zeileis A (2008). *dynlm: Dynamic Linear Regression*. R package version 0.2-0.

Affiliation:

Claudio Lupi
Department of Economics, Management and Social Sciences (SEGeS)
University of Molise
Via De Sanctis I-86100 Campobasso, Italy
E-mail: lupi@unimol.it