Package 'CADFtest'

October 12, 2022

Type Package

Title A Package to Perform Covariate Augmented Dickey-Fuller Unit Root
Tests

Version 0.3-3

Date 2017-05-31

Author Claudio Lupi

Maintainer Claudio Lupi <lupi@unimol.it>

Depends dynlm, sandwich, tseries, urca

Description Hansen's (1995) Covariate-Augmented
Dickey-Fuller (CADF) test. The only required argument is y, the
Tx1 time series to be tested. If no stationary covariate X is
passed to the procedure, then an ordinary ADF test is
performed. The p-values of the test are computed using the
procedure illustrated in Lupi (2009).

License GPL (>= 2)

URL http://www.jstatsoft.org/v32/i02

LazyLoad yes

LazyData yes

Repository CRAN

Date/Publication 2017-06-02 17:10:31 UTC

NeedsCompilation no

R topics documented:

1

CADFpvalues   *p-values of the CADF test for unit roots*

Description

The asymptotic p-values of the Hansen's (1995) Covariate-Augmented Dickey Fuller (CADF) test
for a unit root are computed using the approach outlined in Costantini et al. (2007). The function
can be used also to compute the p-values of the ordinary Dickey-Fuller distribution.

Usage

```
CADFpvalues(t0, rho2 = 0.5, type=c("trend", "drift", "none"))
```

Arguments

t0	the value of the test statistic.
rho2	the value of the long-run correlation. When rho2 = 1 is set, the p-values of the ordinary Dickey-Fuller are computed.
type	defines the deterministic kernel used in the test. It accepts the values used in package urca. It specifies if the underlying model must be with linear trend ("trend", the default), with constant ("drift") or without constant ("none").

Value

p.value, a scalar containing the estimated asymptotic p-value of the test.

Author(s)

Claudio Lupi

References

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

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

Examples

```
CADFpvalues(t0=-1.7, rho2=0.20, type="trend")
```

CADFtest	*Hansen's Covariate-Augmented Dickey Fuller (CADF) test for unit roots*

Description

This function is an interface to `CADFtest.default` that computes the CADF unit root test proposed in Hansen (1995). The asymptotic p-values of the test are also computed along the lines proposed in Costantini et al. (2007). Automatic model selection is allowed. A full description and some applications can be found in Lupi (2009).

Usage

```
CADFtest(model, X=NULL, type=c("trend", "drift", "none"),
    data=list(), max.lag.y=1, min.lag.X=0, max.lag.X=0,
    dname=NULL, criterion=c("none", "BIC", "AIC", "HQC",
    "MAIC"), ...)
```

Arguments

model	a formula of the kind y ~ x1 + x2 containing the variable y to be tested and the stationary covariate(s) to be used in the test. If the model is specified as y ~ 1, then an ordinary ADF is carried out. Note that the specification y ~ . here does not imply a model with all the disposable regressors, but rather a model with no stationary covariate (which correspons to an ADF test). This is because the stationary covariates have to be explicitly indicated (they are usually one or two). An ordinary ADF is performed also if model=y is specified, where y is a vector or a time series. It should be noted that model is not the actual model, but rather a representation that is used to simplify variable specification. The covariates are assumed to be stationary.
X	if model=y, a matrix or a vector time series of stationary covariates X can be passed directly, instead of using the `formula` expression. However, the `formula` expression should in general be preferred.
type	defines the deterministic kernel used in the test. It accepts the values used in package `urca`. It specifies if the underlying model must be with linear trend ("trend", the default), with constant ("drift") or without constant ("none").
data	data to be used (optional). This argument is effective only when model is passed as a formula.
max.lag.y	maximum number of lags allowed for the lagged differences of the variable to be tested.
min.lag.X	if negative it is maximum lead allowed for the covariates. If zero, it is the minimum lag allowed for the covariates.
max.lag.X	maximum lag allowed for the covariates.
dname	NULL or character. It can be used to give a special name to the model. If the NULL default is accepted and the model is specified using a formula notation, then dname is computed according to the used formula.

criterion	it can be either "none" (the default), "BIC", "AIC", "HQC" or "MAIC". If criterion="none", no automatic model selection is performed. Otherwise, automatic model selection is performed using the specified criterion. In this case, the max and min orders serve as upper and lower bounds in the model selection.
...	Extra arguments that can be set to use special kernels, prewhitening, etc. in the estimation of ρ^2. A Quadratic kernel with a VAR(1) prewhitening is the default choice. To set these extra arguments to different values, see kernHAC in package sandwich (Zeileis, 2004, 2006). If Hansen's results have to be duplicated, then kernel="Parzen" and prewhite=FALSE must be specified.

Value

The function returns an object of class c("CADFtest", "htest") containing:

statistic	the t test statistic.
parameter	the estimated nuisance parameter ρ^2 (see Hansen, 1995, p. 1150).
method	the test performed: it can be either ADF or CADF.
p.value	the p-value of the test.
data.name	the data name.
max.lag.y	the maximum lag of the differences of the dependent variable.
min.lag.X	the maximum lead of the stationary covariate(s).
max.lag.X	the maximum lag of the stationary covariate(s).
AIC	the value of the AIC for the selected model.
BIC	the value of the BIC for the selected model.
HQC	the value of the HQC for the selected model.
MAIC	the value of the MAIC for the selected model.
est.model	the estimated model.
estimate	the estimated value of the parameter of the lagged dependent variable.
null.value	the value of the parameter of the lagged dependent variable under the null.
alternative	the alternative hypothesis.
call	the call to the function.
type	the deterministic kernel used.

Author(s)

Claudio Lupi

References

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

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

Lupi C (2009). Unit Root CADF Testing with R, *Journal of Statistical Software*, **32**(2), 1–19. http://www.jstatsoft.org/v32/i02/

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

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

See Also

fUnitRoots, urca

Examples

```
##---- ADF test on extended Nelson-Plosser data ----
##--   Data taken from package urca
  data(npext, package="urca")
  ADFt <- CADFtest(npext$gnpperca, max.lag.y=3, type="trend")

##---- CADF test on extended Nelson-Plosser data ----
  data(npext, package="urca")
  npext$unemrate <- exp(npext$unemploy)     # compute unemployment rate
  L <- ts(npext, start=1860)                # time series of levels
  D <- diff(L)                              # time series of diffs
  S <- window(ts.intersect(L,D), start=1909) # select same sample as Hansen's
  CADFt <- CADFtest(L.gnpperca~D.unemrate, data=S, max.lag.y=3,
    kernel="Parzen", prewhite=FALSE)
```

| coeffs_ct | *Tables of coefficients to compute p-values* |

Description

These tables contain the response surface coefficients needed to compute the p-value of Hansen's CADF test (see Costantini et al., 2007; Lupi, 2009). coeffs_ct, coeffs_c, coeffs_nc are the relevant tables for the constant plus trend, constant, and no constant case, respectively.

Usage

```
data("coeffs_ct")
data("coeffs_c")
data("coeffs_nc")
```

Format

The tables are saved as binary data .rda objects. They are (1005×5) matrices, where the first column represents probabilities and the following four columns are $\beta_0, \ldots, \beta_3$ in eqn. (13) in Costantini et al. (2007) (see also Lupi, 2009).

Author(s)

Claudio Lupi

Source

Costantini et al. (2007).

References

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

Lupi C (2009). Unit Root CADF Testing with R, *Journal of Statistical Software*, **32**(2), 1–19. http://www.jstatsoft.org/v32/i02/

---

plot.CADFtest	*Function to plot CADFtest objects*

Description

This function conveniently plots the residuals of the Covariate Augmented Dickey-Fuller or the standard Augmented Dickey-Fuller regression carried out in CADFtest.

Usage

```
   ## S3 method for class 'CADFtest'
plot(x, plots=(1:4), ...)
```

Arguments

x	an object belonging to the class CADFtest.
plots	the plots to be produced (all the four plots by default): 1: standardized residuals plot; 2: density of the residuals, with an indication of the p-value of the Jarque-Bera test for normality; 3: ACF of the residuals; 4: partial ACF of the residuals.
...	currently not used.

Author(s)

Claudio Lupi

Examples

```
   data(npext, package="urca")
   ADFt <- CADFtest(npext$realgnp, type="trend")
   plot(ADFt, plots=c(3,4))
```

---

residuals.CADFtest *Function to extract the residuals from CADFtest objects*

Description

This function applies the residuals() method to an object of class CADFtest.

Usage

```
    ## S3 method for class 'CADFtest'
residuals(object, ...)
```

Arguments

object   an object belonging to the class CADFtest.

...   currently not used.

Author(s)

Claudio Lupi

Examples

```
    data(npext, package="urca")
    ADFt <- CADFtest(npext$realgnp, type="trend")
    residuals(ADFt)
```

---

summary.CADFtest *Function to print a summary of CADFtest objects*

Description

This function conveniently prints the detailed results of the Covariate-Augmented Dickey Fuller test carried out in CADFtest.

Usage

```
    ## S3 method for class 'CADFtest'
summary(object, ...)
```

Arguments

object   an object belonging to the class CADFtest.

...   currently not used.

Value

The function returns an object of class CADFtestsummary containing the main results of the test.

test.summary   a matrix, containing the t-test statistic, the estimated value of ρ^2, the p-value of
                the test, the max lag used for the differenced dependent variable, the max lag
                of the stationary covariate, the max lead of the stationary covariate. When a
                standard ADF test is performed, only the t-test statistic, the p-value and the max
                lag of the differenced dependent variable are reported.

model.summary   the summary of the test model, in the usual form. However, note that the p-value
                of the lagged dependent is computed under the null of a unit root. Furthermore,
                differently from the common practice, the F-statistic refers to the joint signifi-
                cance of the stationary regressors. If no stationary regressors are used (no lagged
                differences of the dependent, no stationary covariates) then the F-statistic is not
                computed and a NA value is returned.

Author(s)

Claudio Lupi

Examples

```
data(npext, package="urca")
ADFt <- CADFtest(npext$realgnp, type="trend")
summary(ADFt)
```

---

update.CADFtest   *Function to update the formula of CADFtest objects*

Description

This function updates the formula and/or the other arguments of CADFtest object and re-run the test
using the updated arguments. It can be useful if one wants to see the effect of adding/removing sta-
tionary covariates or the effect of changing lags, kernel, etc. If covariates have to be added/removed,
update() works only if model is passed as a formula.

Usage

```
   ## S3 method for class 'CADFtest'
update(object, change, ...)
```

Arguments

object   an object belonging to the class CADFtest.

change   list of charater describing the changes to be applied to the existing model.

...   currently not used.

Value

The function re-run the test and returns an object of class `CADFtest`. See `CADFtest()`.

Author(s)

Claudio Lupi

Examples

```
data(npext, package="urca")
npext$unemrate <- exp(npext$unemploy)     # compute unemployment rate
L <- ts(npext, start=1860)                # time series of levels
D <- diff(L)                              # time series of diffs
S <- window(ts.intersect(L,D), start=1909) # select same sample as Hansen's
CADFt <- CADFtest(L.gnpperca~D.unemrate, data=S, max.lag.y=3,
  kernel="Parzen", prewhite=FALSE)
CADFt.2 <- update(CADFt, change=list("+ D.indprod", "max.lag.X=3",
"criterion='BIC'"))
```

Index