Package 'apt'

February 4, 2011

Title Asymmetric Price Transmission (apt)

Version 1.0

Date 2010-11-03

Author Changyou Sun <csun@cfr.msstate.edu>

Maintainer Changyou Sun <csun@cfr.msstate.edu>

Depends R (>= 2.10.0), car, erer

Description This package focuses on asymmetric price transmission
(APT) between two time series. This package contains functions
for linear and nonlinear threshold cointegration, and
furthermore, symmetric and asymmetric error correction model.

License GPL (version 2 or later)

LazyLoad yes

R topics documented:

---

apt-package	*Asymmetric Price Transmission*

Description

This package focuses on asymmetric price transmission between two time series. The name of functions and datasets reveals the categories they belong to. A prefix of `da` is for datasets, `bs` for basic data manipulation, `ci` for cointegration, and `ecm` for error correction model.

This package focuses on the price transmission between *two* price variables. Therefore, objectives like fitting an error correction model for five variables are beyond the scope of this package.

Details

Package:	apt
Type:	Package
Version:	1.0
Date:	2010-11-03
Depends:	R (>= 2.10.0), car, erer
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Changyou Sun <csun@cfr.msstate.edu>

---

ciTarFit	*Fitting Threshold Cointegration*

Description

Fit a threshold cointegration regression between two time series.

Usage

```
ciTarFit(y, x, model = c("tar", "mtar"), lag, thresh,
    small.win, ...)
```

Arguments

y	dependent or left-side variable for the long-run regression.
x	independent or right-side variable for the long-run regression.
model	a choice of two models: tar or mtar.
lag	number of lags for the threshold cointegration regression.
thresh	a threshold value.
small.win	value of a small window for fitting the threshold cointegration regression.
...	additional arguments to be passed.

Details

This is the main function for threshold autoregression regression (TAR) in assessing the nonlinear threshold relation between two time series variables. It can be used to estimate four types of threshold cointegration regressions. These four types are TAR with a threshold value of zero; consistent TAR with a nonzero threshold; MTAR (momentum TAR) with a threshold value of zero; and consistent MTAR with a nonzero threshold. The option of small window is used in model selection because a comparison of AIC and BIC values should be based on the same number of regression observations.

Value

Return a list object of class "ciTarFit" with the following components:

y	dependend variable
x	independent variable
model	model choice
lag	number of lags
thresh	threshold value
data.LR	data used in the long-run regression
data.CI	data used in the threshold cointegration regression
z	residual from the long-run regression
lz	lagged residual from the long-run regression
ldz	lagged residual with first difference from the long-run regression
LR	long-run regression
CI	threshold cointegration regression
f.phi	test with the null hypothesis of no threshold cointegration
f.apt	test with the null hypothesis of no asymmetric price transmission in the long run
sse	value of sum of squared errors
aic	value of Akaike Information Criterion
bic	value of Bayesian Information Criterion.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Balke, N.S., and T. Fomby. 1997. Threshold cointegration. International Economic Review 38(3):627-645.

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

Enders, W., and P.L. Siklos. 2001. Cointegration and threshold adjustment. Journal of Business and Economic Statistics 19(2):166-176.

See Also

print.ciTarFit; summary.ciTarFit; ciTarLag for lag selection; and ciTarThd for threshold selection;

Examples

```
data(daVi); data(daCh)
t.tar <- -8.041; t.mtar <- -0.451
mya <- myb <- 3

(f1 <- ciTarFit(y=daVi, x=daCh, model="tar", lag=mya, thresh=0))
(f2 <- ciTarFit(y=daVi, x=daCh, model="tar", lag=mya, thresh=t.tar))
(f3 <- ciTarFit(y=daVi, x=daCh, model="mtar",lag=myb, thresh=0))
(f4 <- ciTarFit(y=daVi, x=daCh, model="mtar",lag=myb, thresh=t.mtar))

r0 <- cbind(summary(f1)$dia, summary(f2)$dia, summary(f3)$dia,
    summary(f4)$dia)
(r1 <- r0[, c(1,2,4,6,8)])
col.name <- c("item","tar","c.tar","mtar","c.mtar")
diag <- r1[c(1:4, 6:7, 12:14, 8, 9, 11), col.name]
rownames(diag) <- 1:nrow(diag); diag

e1 <- summary(f1)$out; e2 <- summary(f2)$out
e3 <- summary(f3)$out; e4 <- summary(f4)$out
(r2 <- rbind(e1, e2, e3, e4))

ee <- list(e1, e2, e3, e4); vect <- NULL
for (i in 1:4) {
  ef <- data.frame(ee[i]); ef
  vect2 <- c(paste(ef[3, "estimate"], ef[3, "sign"], sep=""),
         paste("(", ef[3, "t.value"], ")", sep=""),
         paste(ef[4, "estimate"], ef[4, "sign"], sep=""),
         paste("(", ef[4, "t.value"], ")", sep=""))
  vect <- cbind(vect, vect2)
}
item <- c("pos.coeff","pos.t.value", "neg.coeff","neg.t.value")
ve <- data.frame(cbind(item, vect)); colnames(ve) <- col.name
( tab3.b <- rbind(diag, ve)[c(1:2, 13:16, 3:12), ])
```

---

ciTarLag	*Lag Selection for Threshold Cointegration Regression*

Description

Select the best lag for threshold cointegration regression by AIC and BIC

Usage

```
ciTarLag(y, x, model = c("tar", "mtar"), maxlag, thresh,
    adjust = TRUE, ...)
```

Arguments

y	dependent or left-side variable for the long-run regression.
x	independent or right-side variable for the long-run regression.
model	a choice of two models, either tar or mtar.
maxlag	maximum number of lags allowed in the search process.

thresh	a threshold value.
adjust	logical value (default of TRUE) of whether to adjust the window widths so all regressions by lag have the same number of observations
...	additional arguments to be passed.

Details

Estimate the threshold cointegration regressions by lag and then select the best regression by AIC or BIC value. The longer the lag, the smaller the number of observations availabe for estimation. If the windows of regressions by lag are not ajusted, the maximum lag is usually the best lag by AIC or BIC. Theorectially, AIC and BIC from different models should be compared on the basis of the same observation numbers (Ender 2004). `adjust` shows the effect of this adjustment on the estimation window. By default, the value of `adjust` should be TRUE.

Value

Return a list object of class "ciTarLag" with the following components:

path	a data frame of model criterion values by lag, including `lag` for the current lag, `totObs` for total observations in the raw data, `coinObs` for observations used in the cointegration regression, `sse` for the sum of squared errors, `aic` for AIC value, `bic` for BIC value, `LB4` for the p-value of Ljung_Box Q statistic with 4 autocorrelation coefficients, `LB8` with 8 coefficients, `LB12` for Q statistic with 12 coefficients
out	a data frame of the final model selection, including the values of model, maximum lag, threshold value, best lag by AIC, best lag by BIC

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Enders, W. 2004. Applied Econometric Time Series. John Wiley & Sons, Inc., New York. 480 P.

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

See Also

`plot.ciTarLag` for the graphical trend; `ciTarFit`; and `ciTarThd`;

Examples

```
data(daVi); data(daCh)
t.tar <- -8.041; t.mtar <- -0.451
mx <- 4
(g1 <-ciTarLag(y=daVi, x=daCh, model="tar", maxlag=mx, thresh= 0))
(g2 <-ciTarLag(y=daVi, x=daCh, model="mtar",maxlag=mx, thresh= 0))
(g3 <-ciTarLag(y=daVi, x=daCh, model="tar", maxlag=mx, thresh=t.tar))
(g4 <-ciTarLag(y=daVi, x=daCh, model="mtar",maxlag=mx, thresh=t.mtar))
plot(g4)

(g5 <-ciTarLag(y=daVi, x=daCh, model="mtar",maxlag=mx,
    adjust=FALSE, thresh=t.mtar))
plot(g5)
```

---

ciTarThd   *Threshold Selection for Threshold Cointegration Regression*

Description

Select the best threshold for threshold cointegration regression by sum of squared errors

Usage

```
ciTarThd(y, x, model = c("tar", "mtar"), lag, th.range = 0.15,
    digits = 3, ...)
```

Arguments

y	dependent or left-side variable for the long-run regression.
x	independent or right-side variable for the long-run regression.
model	a choice of two models, either tar or mtar.
lag	number of lags.
th.range	the percentage of observations to be excluded from the search.
digits	number of digits used in rounding outputs.
...	additional arguments to be passed.

Details

The best threshold is determined by fitting the regression for possible threshold values, sorting the results by sum of squared errors (SSE), and selecting the best with the lowest SSE. To have sufficient observations on either side of the threshold value, certain percentage of observations on the top and bottoms are excluded from the search path. This is usually set as 0.15 by the th.range (Chan 1993).

Value

Return a list object of class "ciTarThd" with the following components:

model	model choice
lag	number of lags
th.range	the percentage of observations excluded
th.final	the best threshold value
ssef	the best (i.e., lowest) value of SSE
obs.tot	total number of observations in the raw data
obs.CI	number of observations used in the threshold cointegration regression
basic	a brief summary of the major outputs
path	a data frame of the search record (number of regression, threshold value, SSE, AIC, and BIC values).

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Chan, K.S. 1993. Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model. The Annals of Statistics 21(1):520-533.

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

See Also

`plot.ciTarThd` for the graphical trend; `ciTarFit`; and `ciTarLag`

Examples

```
data(daVi); data(daCh)

# 1. best threshold value for tar
t3 <- ciTarThd(y=daVi, x=daCh, model="tar", lag=0);
(th.tar <- t3$basic)
plot(t3)

# 2. best threshold value for mtar
t4 <- ciTarThd(y=daVi, x=daCh, model="mtar", lag=0)
(th.mtar <- t4$basic)
plot(t4)

# 3. best threshold by lag;
# The following two loops are executable.
# But it may take about 5 to 10 minutes to run
# because there are over 1,000 linear regressions in the loops.

## Not run:
for (i in 1:12) {
    t3a <- ciTarThd(y=daVi, x=daCh, model="tar", lag=i)
    th.tar[i+2] <- t3a$basic[,2]
 }
th.tar

for (i in 1:12) {
    t4a <- ciTarThd(y=daVi, x=daCh, model="mtar", lag=i)
    th.mtar[i+2] <- t4a$basic[,2]
 }
th.mtar

## End(Not run)

# 4. The final result of threshold selection are as follows.
# The best threshold values do not vary much by lag.
# This is not suprising because more lagged values do not
# reduce SSE much.

t.tar <- -8.041; t.mtar <- -0.451    # lag = 0 to 4
t.tar <- -8.701; t.mtar <- -0.451    # lag = 5 to 12
```

---

daCh   *Import price of wooden beds from China*

Description

This data set contains the unit import price of wooden beds from China to the United States.

Usage

```
data(daCh)
```

Format

A monthly time series from Janurary 2002 to Janurary 2010 with 97 observations.

Details

Under the Harmonized Tariff Schedule (HTS) system, the commodity of wooden beds is classified as HTS 9403.50.9040. The monthly cost-insurance-freight values in dollar and quantities in piece are reported by country from U.S. ITC (2010). The unit price (\$/piece) is calculated as the ratio of value over quantity.

Source

U.S. ITC, 2010. Interactive tariff and trade data web. http://dataweb.usitc.gov (Assecced on March 1, 2010).

References

Sun, C. 2011. Price dynamics in the import wooden bed market of the United States. Under preparation.

Examples

```
data(daCh)
plot(daCh)
```

---

daVi   *Import price of wooden beds from Vietnam*

Description

This data set contains the unit import price of wooden beds from Vietnam to the United States.

Usage

```
data(daVi)
```

Format

A monthly time series from Janurary 2002 to Janurary 2010 with 97 observations.

Details

Under the Harmonized Tariff Schedule (HTS) system, the commodity of wooden beds is classified as HTS 9403.50.9040. The monthly cost-insurance-freight values in dollar and quantities in piece are reported by country from U.S. ITC (2010). The unit price (\$/piece) is calculated as the ratio of value over quantity.

Source

U.S. ITC, 2010. Interactive tariff and trade data web. http://dataweb.usitc.gov (Assecced on March 1, 2010).

References

Sun, C. 2011. Price dynamics in the import wooden bed market of the United States. Under preparation.

Examples

```
data(daVi)
plot(daVi)
```

---

daVich   *Import price of wooden beds from Vietnam anc China*

Description

This data set contains two unit import prices of wooden beds from Vietnam and China to the United States.

vietnam	Monthly price over Janurary 2002 to Janurary 2010 from Vietnam.
china	Monthly price over Janurary 2002 to Janurary 2010 from China.

Usage

```
data(daVich)
```

Format

A monthly time series from Janurary 2002 to Janurary 2010 with 97 observations for each of the two series.

Details

Under the Harmonized Tariff Schedule (HTS) system, the commodity of wooden beds is classified as HTS 9403.50.9040. The monthly cost-insurance-freight values in dollar and quantities in piece are reported by country from U.S. ITC (2010). The unit price ($/piece) is calculated as the ratio of value over quantity by country.

Source

U.S. ITC, 2010. Interactive tariff and trade data web. http://dataweb.usitc.gov (Assecced on March 1, 2010).

References

Sun, C. 2011. Price dynamics in the import wooden bed market of the United States. Under preparation.

Examples

```
data(daVich)
plot(daVich)
```

---

ecmAsyFit   *Fitting Asymmetric Error Correction Model*

Description

Estimate an asymmetric error correction model (ECM) for two time series.

Usage

```
ecmAsyFit(y, x, lag = 1, split = TRUE,
    model = c("linear", "tar", "mtar"), thresh, ...)
```

Arguments

y	dependent or left-side variable for the long-run regression.
x	independent or right-side variable for the long-run regression.
lag	number of lags for variables on the right side.
split	a logical value (default of TRUE) of whether the right-hand variables should be split into positive and negative parts.
model	a choice of three models: linear, tar , or mtar cointegration.
thresh	a threshold value; this is only required when the model is specified as 'tar' or 'mtar.'
...	additional arguments to be passed.

Details

There are two specficiations of an asymmetric ECM. The first one is how to calculate the error correction terms. One way is through linear two-step Engle Granger approach, as specificied by model="linear". The other two ways are threshold cointegration by either 'tar' or 'mtar' with a threshold value. The second specification is related to the possible asymmetric price transmission in the lagged price variables, as specified in split = TRUE. Note that the linear cointegration specification is a special case of the threshold cointegration. A model with model="linear" is the same as a model with model="tar", thresh = 0.

Value

Return a list object of class "ecm" and "ecmAsyFit" with the following components:

`y`	dependend variable
`x`	independent variable
`lag`	number of lags
`split`	logical value of whether the right-hand variables are split
`model`	model choice
`IndVar`	data frame of the right-hand variables used in the ECM
`name.x`	name of the independent variable
`name.y`	name of the dependent variable
`ecm.y`	ECM regression for the dependent variable
`ecm.x`	ECM regression for the independent variable
`data`	all the data combined for the ECM
`thresh`	thresh value for TAR and MTAR model

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

See Also

`print.ecm`; `summary.ecm`; `ecmDiag`; and `ecmAsyTest`.

Examples

```
data(daVi); data(daCh)
t.mtar <- -0.451
aem <- ecmAsyFit(y=daVi, x=daCh, lag=4, model="mtar",
    split=TRUE, thresh=t.mtar)
aem
summary(aem)
ecmDiag(aem, 3)
(tes <- ecmAsyTest(aem)$out)
```

---

ecmAsyTest   *Hypothesis Tests on Asymmetric Error Correction Model*

Description

Conduct several F-tests on the coefficients from asymmetric ECM.

Usage

```
ecmAsyTest(w, digits = 3, ...)
```

Arguments

w	an object of 'ecmAsyFit' class.
digits	number of digits used in rounding outputs.
...	additional arguments to be passed.

Details

There are two ECM equations for the two price series. In each equation, four types of hypotheses are
tested; equilibrium adjustment path symmetry on the error correction terms (H1), Granger causality
test (H2), distributed lag symmetry at each lag (H3), and cumulative asymmetry of all lags (H4).
The latter two tests are only feasible and availabe for models with split variables. The number of
H3 tests is equal to the number of lags.

Value

Return a list object with the following components:

H1ex	H01 in equation x: equilibrium adjustment path symmetry
H1ey	H01 in equation y: equilibrium adjustment path symmetry
H2xx	H02 in equation x: x does not Granger cause x
H2yx	H02 in equation y: x does not Granger cause y
H2xy	H02 in equation x: y does not Granger cause x
H2yy	H02 in equation y: y does not Granger cause y
H3xx	H03 in equation x: distributed lag symmetry of x at each lag
H3yx	H03 in equation y: distributed lag symmetry of x at each lag
H3xy	H03 in equation x: distributed lag symmetry of y at each lag
H3yy	H03 in equation y: distributed lag symmetry of y at each lag
H4xx	H04 in equation x: cumulative asymmetry of x for all lags
H4yx	H04 in equation y: cumulative asymmetry of x for all lags
H4xy	H04 in equation x: cumulative asymmetry of y for all lags
H4yy	H04 in equation y: cumulative asymmetry of y for all lags
out	summary of the four types of hypothesis tests

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Frey, G., and M. Manera. 2007. Econometric models of asymmetric price transmission. Journal of Economic Surveys 21(2):349-415.

See Also

`ecmAsyFit` and `ecmDiag`.

Examples

```
data(daVi); data(daCh)
t.mtar <- -0.451
aem <- ecmAsyFit(y=daVi, x=daCh, lag=4, model="mtar", split=TRUE, thresh=t.mtar)
aem
summary(aem)
ecmDiag(aem, 3)
(tes <- ecmAsyTest(aem)$out)
```

---

ecmDiag	*Diagnostic Statitics for Symmetric or Asymmetric ECMs*

Description

Report a set of diagnostic statistics for symmetric or asymmetric error correction models

Usage

```
ecmDiag(m, digits = 2, ...)
```

Arguments

m	an object of class `ecm` from the function of `ecmAsyFit` or `ecmSymFit`.
digits	number of digits used in rounding outputs.
...	additional arguments to be passed.

Details

Compute several diagnostic statistics for each ECM equation. This is mainly used to assess the serial correlation in the residuals and model adequacy.

Value

Return a data frame object with the following components by equation: R-squared, Adjusted R-squared, F-statistic, Durbin Watson statistic, p-value for DW statistic, AIC, BIC, and p-value of Ljung_Box Q statistics with 4, 8, 12 autocorrelation coefficients.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Enders, W. 2004. Applied Econometric Time Series. John Wiley & Sons, Inc., New York. 480 P.

See Also

ecmAsyFit; ecmSymFit; and ecmDiag.

Examples

```
data(daVi); data(daCh)
t.mtar <- -0.451
aem <- ecmAsyFit(y=daVi, x=daCh, lag=4, model="mtar",
    split=TRUE, thresh=t.mtar)
ecmDiag(aem, 3)

sem <- ecmSymFit(y=daVi, x=daCh, lag=4)
ecmDiag(sem)
```

---

ecmSymFit   *Fitting symmetric Error Correction Model*

Description

Estimate a symmetric error correction model (ECM) for two time series.

Usage

```
ecmSymFit(y, x, lag = 1, ...)
```

Arguments

y	dependent or left-side variable for the long-run regression.
x	independent or right-side variable for the long-run regression.
lag	number of lags for variables on the right side.
...	additional arguments to be passed.

Details

The package apt focuses on price transmission between two series. This function estimates a standard error correction model for two time series. While it can be extended for more than two series, it is beyond the objective of the package now.

Value

Return a list object of class "ecm" and "ecmSymFit" with the following components:

y	dependend variable
x	independent variable
lag	number of lags
data	all the data combined for the ECM

IndVar	data frame of the right-hand variables used in the ECM
name.x	name of the independent variable
name.y	name of the dependent variable
ecm.y	ECM regression for the dependent variable
ecm.x	ECM regression for the independent variable

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Enders, W. 2004. Applied Econometric Time Series. John Wiley & Sons, Inc., New York. 480 P.

See Also

`print.ecm`; `summary.ecm`; `ecmDiag`; and `ecmAsyFit`.

Examples

```
data(daVi); data(daCh)
sem <- ecmSymFit(y=daVi, x=daCh, lag=4)
sem; names(sem)
summary(sem)
ecmDiag(sem)
```

| `plot.ciTarLag` | *Plot of AIC and BIC for Threshold Cointegration by Lag* |

Description

Plot the values of AIC and BIC for threshold cointegration regressions by lag length.

Usage

```
## S3 method for class 'ciTarLag'
plot(x, ...)
```

Arguments

| x | an object of class 'ciTarLag'. |
| ... | additional arguments to be passed. |

Details

This demonstrates the trend of AIC and BIC changes of threshold cointegration regressions by lag. It facilitates the selection of the best lag for a threshold cointegration model.

Value

A panel of two graphs.

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

See Also

ciTarLag; ciTarThd; and ciTarFit.

Examples

```
data(daVi); data(daCh)
t.tar <- -8.041; t.mtar <- -0.451
mx <- 4
(g4 <-ciTarLag(y=daVi, x=daCh, model="mtar",maxlag=mx,
    adjust=TRUE, thresh=t.mtar))
plot(g4)
```

---

plot.ciTarThd	*Plot for Threshold Value Selection*

Description

Plot the values of SSE, AIC and BIC for threshold cointegration regressions by threshold value.

Usage

```
## S3 method for class 'ciTarThd'
plot(x, ...)
```

Arguments

x	an object of class 'ciTarThd'.
...	additional arguments to be passed.

Details

This function generates three plots in one window. They reveals the relationship between SSE (sum of squared errors), AIC, BIC and threshold value. The best threshold value is associated with the lowest SSE value.

Value

A plot of three graphs (SSE, AIC, and BIC versus threshold value).

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Chan, K.S. 1993. Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model. The Annals of Statistics 21(1):520-533.

Enders, W., and C.W.J. Granger. 1998. Unit-root tests and asymmetric adjustment with an example using the term structure of interest rates. Journal of Business & Economic Statistics 16(3):304-311.

See Also

ciTarThd and ciTarFit.

Examples

```
data(daVi); data(daCh)
t4 <- ciTarThd(y=daVi, x=daCh, model="mtar", lag=0)
(th.mtar <- t4$basic)
plot(t4)
```

---

print.ciTarFit	*Printing Outputs from Threshold Cointegration Regression*

Description

Show main outputs from threshold cointegration regression on screen.

Usage

```
## S3 method for class 'ciTarFit'
print(x, ...)
```

Arguments

x	an object of class 'ciTarFit'.
...	additional arguments to be passed.

Details

This is the print method for ciTarFit to show main outputs from the threshold cointegration regression.

Value

Four main outputs from threshold cointegration regression are shown: long-run regression between the two price variables, threshold cointegration regression, hypothesis test of no cointegration, and hypothesis of no asymmetric adjustment.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

ciTarFit.

Examples

```
data(daVi); data(daCh)
f1 <- ciTarFit(y=daVi, x=daCh, model="tar", lag=3, thresh=0)
f1
```

---

print.ecm	*Printing Outputs from Error Correction Models*

Description

Show main outputs from symmetric `ecmSymFit` or asymmetric `ecmAsyFit` error correction models.

Usage

```
## S3 method for class 'ecm'
print(x, ...)
```

Arguments

x   an object of class `ecm` from the function of `ecmAsyFit` or `ecmSymFit`.

...   additional arguments to be passed.

Details

This is the print method for `ecmAsyFit` or `ecmSymFit` to show main model outputs.

Value

Summary results of the two ECM equations are shown for the two focal price series.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

`ecmSymFit` and `ecmAsyFit`.

Examples

```
data(daVi); data(daCh)
sem <- ecmSymFit(y=daVi, x=daCh, lag=4); sem
aem <- ecmAsyFit(y=daVi, x=daCh, lag=4, model="mtar",
    split=TRUE, thresh=-0.451)
aem
```

---

summary.ciTarFit *Summary of Results from Threshold Cointegration Regression*

Description

This summarizes the main results from threshold cointegration regression.

Usage

```
## S3 method for class 'ciTarFit'
summary(object, digits=3, ...)
```

Arguments

object   an object of class 'ciTarFit'.

digits   number of digits for rounding outputs.

...   additional arguments to be passed.

Details

This wraps up the outputs from threshold cointegration regression in two data frames, one for diagnostic statistics and the other for coeffients.

Value

A list with two data frames. dia contains the main model specifications and hypothesis test results. out contains the regression results for both the long run (LR) and threshold cointegration (CI).

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

ciTarFit.

Examples

```
data(daVi); data(daCh)
(f4 <- ciTarFit(y=daVi, x=daCh, model="mtar",lag=3, thresh=-0.451))
f4.sum <- summary(f4)
names(f4.sum)
f4.sum$dia
f4.sum$out
f4.sum
```

---

summary.ecm   *Summary of Results from Error Correction Model*

Description

This summarizes the main results from error correction models.

Usage

```
## S3 method for class 'ecm'
summary(object, digits=3, ...)
```

Arguments

object	an object of class `ecm` from the function of `ecmAsyFit` or `ecmSymFit`.
digits	number of digits for rounding outputs
...	additional arguments to be passed.

Details

This wraps up the coefficents and statistics from ECM by equation.

Value

A data frame object with coefficients and related statistics by equation.

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

See Also

`ecmSymFit` and `ecmAsyFit`.

Examples

```
data(daVi); data(daCh)
sem <- ecmSymFit(y=daVi, x=daCh, lag=4)
aem <- ecmAsyFit(y=daVi, x=daCh, lag=4, model="mtar",
    split=TRUE, thresh=-0.451)
summary(sem)
summary(aem)
```

Index