Transmissão de Preço no mercado de combustíveis

Caio, Lucas e Marcos

3 de fevereiro de 2017

Introdução

Estudos sobre a interação dos mercados e transmissão de preços são recorrentes na literatura econômica. Os primeiros estudos que se preocuparam com esse assunto analisaram apenas a correlação entre os preços, em cada mercado ou elo da cadeia produtiva, para explicar como se dava essa transmissão. O primeiro modelo que se preocupou com o caráter dinâmico dessas relações foi o proposto por Ravallion (1986), observando a diferença entre a relação de preços de custo prazo, da relação de equilíbrio de preços no longo prazo, como bem aponta Mattos (2009).

Conforme Garaffa (2016), modelos de forma reduzida (ou modelos financeiros) ganharam força ao longo dos anos 2000, devido ao processo de financeirização do mercado de commodities. Estes modelos diferem dos modelos estruturais ao focar na relação de interação entre os preços, não se preocupando com a estimação de parâmetros de oferta ou demanda. Desta forma, modelos na forma reduzida demandam apenas informações sobre as propriedades das séries históricas de preços, e se desenvolveram a partir dos modelos autorregressivos (AR), com posterior incorporação de dos modelos de correção de erro e análises de cointegração. (HUNTINGTON et al., 2013).

Neste trabalho será analisado a interação de preços de Gasolina em nível de distribuição e de revenda na cidade de Guarulhos-SP, para tal será utilizado método TVEC (*Threshold Vector Error Correction*), que é uma versão não linear do modelo do vetor de correção de erros (VEC) membro do grupo de modelos TAR, do inglês *Threshold Autoregressive Models*. Os modelos TAR apresentam um avanço em relação aos seus anteriores, lineares, no sentido de possibilitarem a incorporação de assimetrias e custos de transação ao arcabouço de estudo da integração de mercados, tais imperfeições geram não linearidades no movimento de adaptação de preços que não são captadas por modelos autorregressivos tradicionais. Vale apontar que esses modelos, apesar de serem compatíveis com o conceito de custos de transação, não são capazes de apontar a origem desses custos, apenas podem mensurar seus efeitos.

São exemplos de trabalhos que adotaram modelos com threshold: Serra e Gil (2006), que tratam os custos de transação no mercado de porco europeu por meio do modelo TAR; Bem-Kaabia e Gil (2007), que também por meio de um modelo TAR analisaram as assimetrias entre os preços ao produtor e varejista no mercado de carne de carneiro espanhol; Nick e Threschler (2014), que observaram a aplicabilidade da lei do preço único no mercado de gás natural europeu por meio de um modelo TVEC; Mattos (2009), que por meio de um modelo TVEC analisou os custos de transação no mercado do boi gordo brasileiro, e em trabalho semelhantes Mattos (2009), com uso de um modelo TAR, analisa a mesma questão no mercado de frango brasileiro; e Garaffa (2016), em consonância com o trabalho de Nick e Theschler (2014) avalia o mercado de gás europeu por meio de modelos TVEC e TVAR.

O objeto de estudo abordado neste trabalho será o mercado de combustíveis de Guarulhos. Podemos dizer que tal escolha se justifica pela importância da questão energética, tanto em âmbito nacional como internacional, e no fato de que Guarulhos, como segunda maior cidade do Estado de São Paulo, pode apresentar bons indícios de como este mercado se comporta em nível metropolitano ou mesmo estadual.

O mercado de Combustível

Em construção

Objetivos

O objetivo principal deste trabalho é analisar a relação de transmissão de preços entre distribuidores e revendedores de combustíveis - especificamente gasolina tipo C - no município de Guarulhos. Para tal, empregou-se a metodologia de Cointegração com *threshold* e testou-se a presença de assimetria na transmissão de preços. Nesse contexto uma pergunta interessante surge: como a entrada de veículos com a tecnologia *flexfuel* pode ter alterado (ou não) a transmissão vertical de preços do mercado análisado?

Dados

Estes dados foram disponibilizados pela Superintendência de Planejamento e Pesquisa no sítio da ANP e refletem o preço de venda médio (por litro) de gasolina tipo C realizadas pelas distribuidoras e revendedoras dos derivados combustíveis de petróleo no município de Guarulhos. O período de análise é de janeiro de 2003 à dezembro de 2012, entretanto, a fim de melhorar a precisão da análise e facilitar a comparação dos resultados, este período é dividido em dois subperíodos, em função da influência sofrida pela entrada da tecnologia *flex fuel* no mercado. Os valores mensais correntes foram deflacionados utilizando a série histórica do Índice Nacional de Preços ao Consumidor (IPCN), disponibilizada no sitio do Instituto Brasileiro de Geografia e Estatística, IBGE. Foi utilizado o mês de fevereiro de 2016 como referência para os cálculos.

Metodologia

Modelos que examinam a natureza da transmissão vertical de preços foram analisados em diferentes mercados, mas principalmente no setor alimentício. Esta relação começou a ser estudada pelo modelo de Houck(1977) e passou por diversas modificações com o passar do tempo. Seja o modelo estático na forma reduzida dado por:

$$\sum_{\tau=1}^{t} \Delta PR_\tau = \alpha_0 + \alpha_1 \sum_{\tau=1}^{t} \Delta PPI_\tau + \alpha_2 \sum_{\tau=1}^{t} \Delta PPF_\tau + \varepsilon_t \tag{1}$$

Em que PR_τ representa variações nos preços do revendedor, PPI_τ e PPF_τ são as variações positivas e negativas nos preços de distribuição/produção, respectivamente,α_0,$alpha_1$ e $\alpha 2$ são coeficientes a serem estimados, t é o tempo corrente e ε_t é o termo de erro aleatório. A hipótese nula de ajuste simétrico de preços é testada por meio das estimativas de α_1 e α_2. É comum o uso de técnicas de cointegração para estimas estes parâmetros, entretanto, von Cramon-Taubadel e Loy (1997) demonstraram que a especificação na equação 1 é inconsistente com o conceito de cointegração. Em seguida, Azzam (1999), em um trabalho seminal, mostrou que na presença de rigidez de preço o uso de funções não reversíveis, como é o caso da equação 1, o teste de assimetria não é apropriado.

Nesse sentido, este trabalho emprega um modelo de cointegração que reconhece o fato de que um choque pode ter que atingir um nível crítico antes que uma resposta significativa seja provocada. Considere a relação simples que é usada como base para várias análises de cointegração:

$$\Delta x_t = \pi x_{t-1} + \vartheta_t \tag{2}$$

Em que x_t é um vetor de variáveis estacionárias não aleatórias, π é uma matriz nxn e ϑ_t é um componente de erro normalmente distribuido. O procedimento de cointegração de Johansen consiste em estimar π e determinar seu rank. A idéia dessa abordagem é testar se o rank de π é ou não igual a zero. Em caso negativo, o sistema exibe ajustamento simetrico em torno de $x_t = 0$, ou seja, para qualquer $x_t \neq 0$, Δx_{t+1} será igual à πx_t.

A abordagem de dois passos de Engle-Granger (1987) também testa o ajuste simétrico. A abordagem usa OLS para estimar a relação de equilíbrio de longo prazo como:

$$x_{1t} = \beta_0 + \beta_2 x_{2t} + ... + \beta_n x_{nt} + \mu_t \tag{3}$$

Em que x_{it} são variáveis não estacionárias, β_i são parâmetros a serem estimados e μ_t é um termo de erro que pode ser serialmente correlacionados. O resíduos são utilizados para estimar a seguinte relação:

$$\Delta\mu_t = \rho\mu_{t-1} + \varepsilon_t \tag{4}$$

Em que ε_t é um ruído branco. A rejeição da hipótese nula de não cointegração (isto é, aceitando a hipótese alternativa de $2 < \rho < 0$) implica que os resíduos na Equação 3 são estacionários com média zero. De acordo com o teorema de Engle-Granger (1987), se $\rho \neq 0$, 3 e 4 implicam na existencia de um modelo de correção de erros que pode ser representado por:

$$\Delta x_{1t} = \delta_j \left(x_{1t-1} - \beta_0 - \beta_{2t-1} - ... - \beta_n x_{nt-1}\right) + \sum_{j=1}^{k} \beta_{2j} \Delta x_{2,t-j} + ... + \sum_{j=1}^{k} \beta_{nj} \Delta x_{n,t-j} + \upsilon_{1t} \tag{5}$$

Em que k determina a defasagem e υ_{1t} é um ruído branco. O termo dentro dos parênteses fornece o mecanismo de correção de erro. Enders e Granger (1998) argumentam que os testes de cointegração de Engle-Granger e Johansen são mal especificados se o ajuste é assimétrico. Quando esses testes são empregados para analisar a transmissão de preço de uma relação vertical de um mercado, a hipótese implicita é que as respostas as variações de preços são simétricas: choques no preço de produção/distribuição geram variações da mesma magnitude no preço dos revendedores, independente do choque ser positivo ou negativo.

Enders e Granger (1998) consideram um modelo alternativo de correção de erro denominado modelo autorregressivo com *threshold* (TAR), no qual a Equação 4 é representada como:

$$\Delta\mu_t = \begin{cases} \rho_1\mu_{t-1} + \varepsilon_t & \text{if} \quad \mu_{t-1} \geq 0 \\ \rho_2\mu_{t-1} + \varepsilon_t & \text{if} \quad \mu_{t-1} < 0 \end{cases} \tag{6}$$

A condição necessária para $\{\mu_t\}$ ser estacionária é $-2 < (\rho_1, \rho 2) < 0$. Enders e Granger (1998) mostram que Se a seqüência é estacionária, as estimativas por mínimos quadrados de ρ_1 e ρ_2 têm uma distribuição normal assintótica multivariada. O processo de ajuste é então formalmente quantificado como por meio da função indicadora:

$$\Delta\mu_t = I_t \rho_1 \mu_{t-1} + (1 - I_t)\, \rho_2 \mu_{t-1} + \varepsilon_t \tag{7}$$

Tal que:

$$I_t = \begin{cases} 1 & \text{if} \quad \mu_{t-1} \geq 0 \\ 0 & \text{if} \quad \mu_{t-1} < 0 \end{cases} \tag{8}$$

Nesse caso, zero representa o valor do *threshlod*. Modelos que utilizam as equações 7 e 8 são referidos como modelos de auto-regressão com *threshold* (TAR), enquanto o teste para comportamento de equilibrio de longo prazo com *threshold* é denominado teste de cointegração com *threshold*. Assumindo que o sistema é convergente, $\mu_t = 0$ éconsiderado o valor de equilibrio de longo prazo da série. Se μ_t está abaixo do valor de equilibrio, o ajustamento é de $\rho_1\mu_t$, por outro lado, Se μ_t está acima do valor de equilibrio, o ajustamento é de $\rho_2\mu_t$.

Dado que o ajuste é simétrico ou seja, $\rho_1 = \rho_2$, a abordagem de Engle-Granger é um caso especial das Equações 7 e 8.Dada a existência de um vetor de cointegração, a representação do modelo de correção de erro apresentada em 5 pode ser escrita como:

$$\Delta x_{1t} = \rho_{1.1} I_t \mu_{t-1} + \rho 2.1(1 - I_t)\mu_{t-1} + \sum_{j=1}^{k} \beta_{2j} \Delta x_{2,t-j} + ... + \sum_{j=1}^{k} \beta_{nj} \Delta x_{n,t-j} + \upsilon_{1t} \tag{9}$$

Em que $\rho_{1.1}$ e $\rho_{2.1}$ são os coeficientes de ajustamento para diferenças positivas e negativas, respectivamente. Enders e Granger (1998) mostraram que a equação 7 pode ser extendido para um modelo de defasagens em diferenças, como:

$$\Delta \mu_t = I_t \rho_1 \mu_{t-1} + (1 - I_t)\, \rho_2 \mu_{t-1} + \sum_{i=1}^{\rho-1} \gamma_i \delta \mu_{t-i} + \varepsilon_t \tag{10}$$

Em vez de estimar a equação 7 por meio da função indicadora 8, o *threshold* pode ser determinado pela variação de μ_t. Nesse caso, a função indicador fica:

$$I_t = \begin{cases} 1 & \text{if} \quad \Delta\mu_{t-1} \geq 0 \\ 0 & \text{if} \quad \Delta\mu_{t-1} < 0 \end{cases} \tag{11}$$

De acordo com Enders e Granger (1998), a substituição de 8 por 11 é especialmente valiosa quando o ajuste é assimétrico, de modo que a série exibe mais “momentos” em uma direção do que em outras. Modelos estimados usando as Equações 3, 14 e 11 são denominados modelos autorregressivos *momentum-threshold* (M-TAR).

No modelo TAR, se, por exemplo, $-2 < \rho_1 < \rho_2 < 0$ a fase negativa da sequênica $\{\mu_t\}$ deverá ser mais persistente que a fase positiva. Enquanto no modelo M-TAR, se, por exemplo $|\rho_1| < |\rho_2|$, então o modelo apresenta menos variações positivas que negativas para $\Delta\mu_{t-1}$.

As estatísticas de teste para a hipótese nula ($\rho_1 = \rho_2 = 0$) usando a especificação TAR e M-TAR são chamadas Φ_μ e Φ_μ^*, respectivamente. Três fatores principais determinam as distribuições de Φ_μ e Φ_μ^*. Estes incluem o número de defasagens de μ_t na Equação 10, o número de variáveis e o tipo de elementos determinísticos incluídos na relação de cointegração. Os valores críticos apropriados para Φ_μ e Φ_μ^* são apresentados em Enders e Siklos (1998) e Enders e Granger (1998).

Resultados

A hipótese de que as séries de preços analisadas são não estacionárias, é testada pelo método Dickey-Fuller aumentado (ADF). O AIC foi utilizado para determinar a defasagem apropriada das séries. Encontrou-se que uma defasagem é a mais apropriada para amabas as variáveis em análise. Os valores da estatística do teste para os preços de revenda são $-2,92$ e $-6,93$ para e série em nível e em primeira diferença, respectivamente, enquanto a estatística do teste para os preços de distribuição são $-3,01$ e $-8,05$ para e série em nível e em primeira diferença, respectivamente, O valor crítico de $-3,99$ a 1% de significância sugere que as duas séries tornam-se estacionárias após a primeira diferença.

Para assegurar que a série em primeira diferença é estacionária foi realizado, também, o teste DF-GLS. O teste DF-GLS (Elliot et al., 1996) é uma versão atualizada do teste ADF padrão (Dickey e Fuller 1979) para quando os dados apresentam média desconhecida e tendência.[1] Os resultados foram ao encontro do teste ADF indicando que todas as séries são integradas de primeira ordem. O teste DG-GLS, por ter mais poder que o teste ADF, fornece um argumento mais forte para a estacionaridade da primeira diferença. A regressão de cointegração é especificada como:

$$PR_t = \beta_0 + \beta_1 PD_t + \mu_t \tag{12}$$

[1] Os autores mostraram, por meio de um *Monte Carlo*, que o DF-GLS tem maior poder e performance em pequenas amostras.

Em que PR_t é o preço de revenda, PD_t representa o preço de distribuição e μ_t os resíduos que, caso sejam estacionários, garantem a relação de longo prazo entre os preços .Para a análise de cointegração *à la* Engle-Granger (1987), a equação 12 foi estimada por MQO. A estimativa da relação de equilibrio de longo prazo (com as estatísticas do teste t em parênteses) é dada por:

$$PR_t = \underset{-3.276}{-0.164} + \underset{54.829}{1.216} PD_t + \widehat{\mu}_t \tag{13}$$

Seguindo o procedimento de Engle Granger, os resíduos da Equação 13 são usados para estimar:

$$\Delta\widehat{\mu}_t = \rho_1\widehat{\mu}_{t-1} + \gamma_1\Delta\widehat{\mu}_{t-1} + \varepsilon_t \tag{14}$$

Como reportado na tabela *www*, o valor estimado de ρ_1 é de -0.239 e a estatística t para a hipótese nula, ρ_1=0, é de -4.116 os valores críticos para o procesimento de Engle-Granger são -1.62, -1.95, -2.58 para 10%, 5% e 1%, respectivamente. Portanto, procedimento de Engle-Granger sugere que as duas séries de preços são cointegradas. O p-valor do teste Ljung box também foi reportado na tabela *www* e indica que os resíduos da equação 14 são não autocorrelacionados.

Tanto o modelo TAR quanto o MTAR podem ser formulados para diferentes especificações de defasagem. A escolha de uma defasagem em ambos os casos foi feita pelo critério de informação AIC como pode ser obervado pela tabela *ttt*. O modelo de cointegração com *threshold* proporcinou uma estatística Φ_μ de 8.398 e 8.584 no modelo TAR e MTAR, respectivamente. Portanto, a hipótese nula de $\rho_1 = \rho_2 = 0$ pode ser rejeitada à um nível de significância de 1%, o que indica que as séries são cointegradas. Sendo assim, a hipótese nula de ajustamento assimétrico de preços ($\rho_1 = \rho_2$) pode ser testado por meio de um teste F padrão (Enders and Granger, 1998). A estatística F de 0.002 no modelo TAR e de 0.326 no modelo MTAR indicam que não se pode rejeitar a hipótese nula de ajustameto simétrico dos preços.

Enders e Granger (1998) mostram que em um modelo TAR, como 6 e 7, se $-2 < \rho_1 < \rho_2 < 0$, a série μ_t vai apresentar mais persistencia sempre que $\mu_{t-1} < 0$. Sendo esse o resultado obtido tanto no modelo TAR quanto no MTAR, o método de Chan (1993) foi, portanto, empregado para determinar uma estimativa consistente do *threshold*. O valor de 0.03 foi obtido no modelo TAR e -0.017 modelo no modelo MTAR. Aa estimativas dos modelos consistentes não variram muito, mas o modelo MTAR consistente, C.MTAR, sugere que é possível rejeitar a hipótese nula de ajustamento simétrico de preços à 5% de significância. Vale ressaltar que, pelos critério de informação AIC e BIC, o C.MTAR apresentou o melhor ajustamento aos dados. Os resultados dos quatro modelos podem sem observados pela tabela fff.

A possibilidade de um ajuste asSimétrico de preços encontrado pelo modelo c.MTAR implica que é incorreto examinar a dinâmica de curto prazo com um modelo simétrico de correção de erros. Um modelo simétrico de correção de erros não revelaria ajustes diferenciais das mudanças positivas e negativas (Enders e Granger, 1998). Assim, o modelo de correção de erro assimétrico (modelo C.MTAR) são empregados na análise. Eles podem ser representados como:

$$\Delta PR_t = \sum_{s=1}^{k} \alpha_s \Delta PR_{t-s} + \sum_{s=0}^{k} \beta_s \Delta PD_{t-s} + \gamma_1 Z_{t-1}^{pos} + \gamma_2 Z_{t-1}^{neg} \tag{15}$$

Em que k é a defasagem, Z_{t-1}^{pos} e Z_{t-1}^{neg} São os termos de correção de erro das regressões da cointegração com *threshold*, representando ajustes de choques positivos e negativos às variações na margem de comercialização. Eles podem ser representados como:

$$Z_{t-1}^{pos} = I_t \left(PR_{t-1} + 0.164 - 1.216 PD_{t-1}\right)$$
$$Z_{t-1}^{neg} = (1 - I_t)\left(PR_{t-1} + 0.164 - 1.216 PD_{t-1}\right)$$

Em que I é uma função indicadora. A tabela *qqq* apresenta os resultados do modelo de correção de erros. As estimativas do modelo simétrico e assimétrico foram apresentadas para a comparação. Vale ressaltar que as estatísticas t para Z_{t-1}^{neg} e Z_{t-1}^{pos} sugerem que o preço de revenda não responde a choques negativou ou positivos na margem de comercialização (modelo assimétrico).

Para avaliar o efeito da entrada dos carros *flexfuel* no contexto da transmissão de preços de gasolina entre revendedora e distribuidora vamos repetir as duas análises - cointegração com *threshold* e modelo de correção de erro assimétrico para modelo C.MTAR - em dois períodos distintos: de janeiro de 2003 à dezembro de 2007 e de janeiro de 2008 à dezembro de 2012.

Em ambos os períodos, os resultados sugerem que é possível rejeitar a hipótese nula de não cointegração ($\rho_1 = \rho_2 = 0$) à 1%. A hipótese nula de ajustamento assimétrico de preços ($\rho_1 = \rho_2$) também foi testada. No período em que haviam menos veiculos *flexfuel* no mercado, de 2003 a 2007, a estatística F encontrada foi de 5.496, ou seja, pode-se rejeitar a hipótese nula de ajustamento simétrico de preços à 5% de significância. No periódo seguinte, a estatistica F calculada subiu para 7.203, sendo assim, a hipótese nula pode ser rejeitada à 1%. Assim como no período completo, em nenhum dos subperídos o preço de revenda parece responder a choques positivos ou negativos na margem de comercialização.

Conclusão

Em construção

Bibliografia

Em construção

Códigos do R

```r
# 1. Data preparation
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(tseries)
library(fUnitRoots)
```

```
## Loading required package: urca

## Loading required package: timeDate

## Loading required package: timeSeries
```

```
##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

##

## Rmetrics Package fBasics

## Analysing Markets and calculating Basic Statistics

## Copyright (C) 2005-2014 Rmetrics Association Zurich

## Educational Software for Financial Engineering and Computational Science

## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.

## https://www.rmetrics.org --- Mail to: info@rmetrics.org

##
## Attaching package: 'fUnitRoots'

## The following objects are masked from 'package:urca':
##
##     punitroot, qunitroot, unitrootTable
```

```r
library(apt)
```

```
## Loading required package: erer

## Loading required package: lmtest

## Loading required package: gWidgets
```

```r
library(urca)
library(copula)
library(car)
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:fBasics':
##
##     densityPlot
```

```r
library(xts)


load("C:/Users/Lucas/Desktop/UFPR/2º Sem/Macroeconometria/ArtigoMacroeconometria/precos.RData")

SP <- precos[grep("GUARULHOS", precos$MUNICIPIO), ]

GSP <- SP[grep("GASOLINA COMUM", SP$PRODUTO), ]
View(GSP)



PDISTGSP<-ts(GSP$PRECOMEDIODISTRIBUICAO,frequency=12,start=c(2003,1),end=c(2012,12))
head(PDISTGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 1.8303 1.8862 1.9001 1.8958 1.8104
```

```r
PREVGSP<-ts(GSP$PRECOMEDIOREVENDA,frequency=12,start=c(2003,1),end=c(2012,12))
head(PREVGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 2.1238 2.1777 2.1545 2.1446 2.0335
```

```r
INPC <- (ts(GSP$INPCFEV2016100, frequency=12,start=c(2003,1),end=c(2012,12)))
head(INPC)
```

```
##       Jan  Feb  Mar  Apr  May
## 2003 87.0 86.4 86.1 85.7 85.4
```

```r
PREVGSP <- PREVGSP*100
head(PREVGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 212.38 217.77 215.45 214.46 203.35
```

```r
PREVGSP <- PREVGSP/INPC
head(PREVGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2003 2.441149 2.520486 2.502323 2.502450 2.381148
```

```r
PDISTGSP <- PDISTGSP*100
head(PDISTGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 183.03 188.62 190.01 189.58 181.04
```

```r
PDISTGSP <- PDISTGSP/INPC
head(PDISTGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2003 2.103793 2.183102 2.206852 2.212135 2.119906
```

```r
#2. Stationarity tests
precoadfGREV=ur.df(PREVGSP,type= "trend", selectlags= "AIC")
precoglsGREV=ur.ers(PREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGREV@lags
```

```
## [1] 1
```

```r
precoadfGREV@teststat[1]
```

```
## [1] -2.928053
```

```r
precoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
precoglsGREV@teststat[1]
```

```
## [1] -2.709104
```

```r
precoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.46 -2.93 -2.64
```

```r
precoadfGDIST=ur.df(PDISTGSP,type= "trend", selectlags= "AIC")
precoglsGDIST=ur.ers(PDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGDIST@lags
```

```
## [1] 1
```

```r
precoadfGDIST@teststat[1]
```

```
## [1] -3.011513
```

```r
precoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
precoglsGDIST@teststat[1]
```

```
## [1] -3.013861
```

```r
precoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.46 -2.93 -2.64
```

```r
diffPREVGSP <- diff(PREVGSP, lag = 1, differences = 1)
diffPDISTGSP <- diff(PDISTGSP, lag = 1, differences = 1)


diffprecoadfGREV=ur.df(diffPREVGSP,type= "trend", selectlags= "AIC")
diffprecoglsGREV=ur.ers(diffPREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGREV@lags
```

```
## [1] 1
```

```r
diffprecoadfGREV@teststat[1]
```

```
## [1] -6.936588
```

```r
diffprecoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
diffprecoglsGREV@teststat[1]
```

```
## [1] -5.145653
```

```r
diffprecoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.46 -2.93 -2.64
```

```r
diffprecoadfGDIST=ur.df(diffPDISTGSP,type= "trend", selectlags= "AIC")
diffprecoglsGDIST=ur.ers(diffPDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGDIST@lags
```

```
## [1] 1
```

```r
diffprecoadfGDIST@teststat[1]
```

```
## [1] -8.056968
```

```r
diffprecoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.99 -3.43 -3.13
```

```r
diffprecoglsGDIST@teststat[1]
```

```
## [1] -5.690091
```

```r
diffprecoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.46 -2.93 -2.64
```

```r
# 2. EG cointegration

LR <- lm(PREVGSP ~ PDISTGSP); summary(LR)
```

```
##
## Call:
## lm(formula = PREVGSP ~ PDISTGSP)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089685 -0.019867 -0.006965  0.017648  0.102410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16459    0.05024  -3.276  0.00138 **
## PDISTGSP     1.21675    0.02219  54.829  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03492 on 118 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9619
## F-statistic:  3006 on 1 and 118 DF,  p-value: < 2.2e-16
```

```r
(LR.coef <- round(summary(LR)$coefficients, 6))
```

```
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -0.164585   0.050239 -3.276062 0.001382
## PDISTGSP     1.216752   0.022192 54.828649 0.000000
```

```r
(ry <- ts(residuals(LR), start=start(PDISTGSP), end=end(PDISTGSP), frequency =12))
```

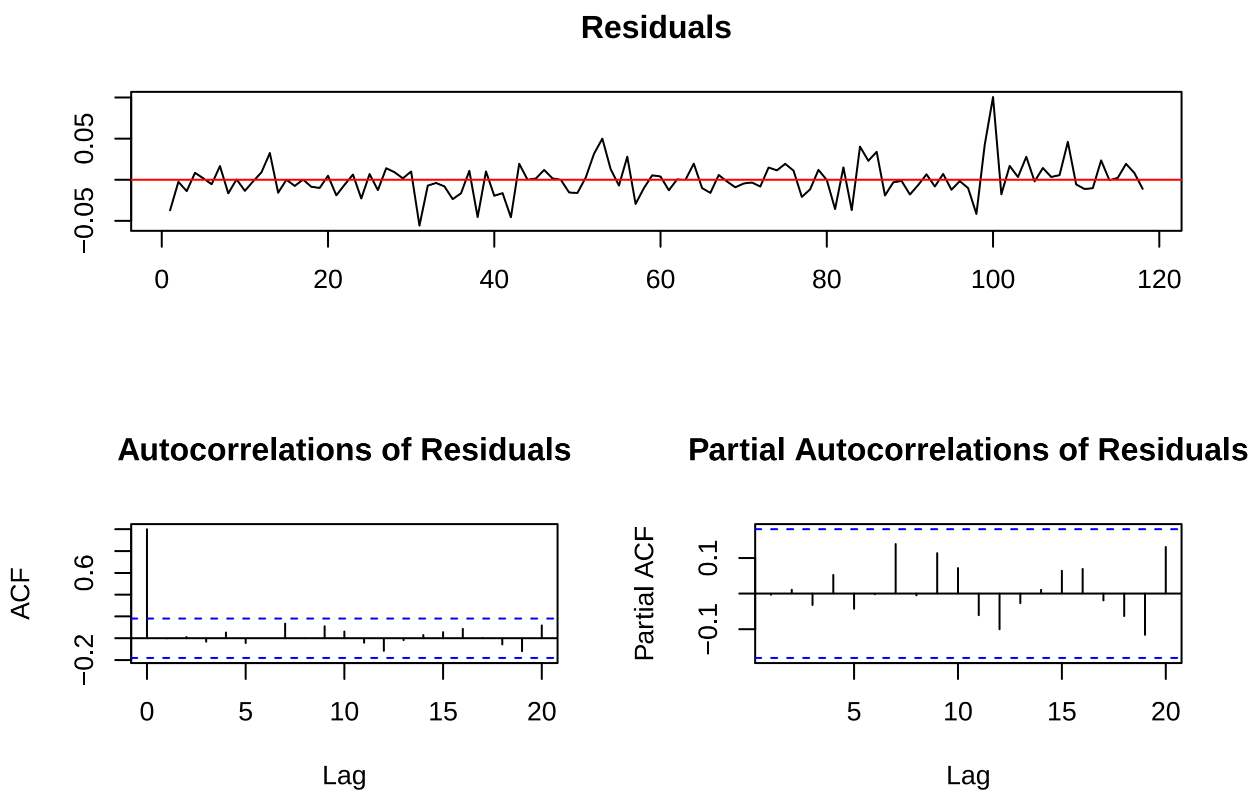
```
##                Jan           Feb           Mar           Apr           May
## 2003  0.0459394496  0.0287770348 -0.0182848477 -0.0245852502 -0.0336682316
## 2004 -0.0189083071 -0.0051792475  0.0307871281  0.0137938599  0.0075712554
## 2005 -0.0161369062 -0.0329243967 -0.0210509416 -0.0266579254 -0.0072832026
```

```
## 2006 -0.0528289613 -0.0590588045 -0.0352359797 -0.0681671564 -0.0474195266
## 2007  0.0028009018  0.0030993146 -0.0130630534 -0.0288712540 -0.0220437113
## 2008 -0.0003173658  0.0040306470 -0.0090567471 -0.0086635947 -0.0066460017
## 2009 -0.0169115417 -0.0211508895 -0.0019877520  0.0131203309  0.0318323269
## 2010 -0.0468183702 -0.0005563218  0.0304908028  0.0623974865  0.0337668935
## 2011 -0.0138878590 -0.0139048546 -0.0207554525 -0.0585841884 -0.0087301076
## 2012  0.0421650334  0.0360552462  0.0723660697  0.0555118787  0.0280679684
##                Jun           Jul           Aug           Sep           Oct
## 2003 -0.0188322949 -0.0101229336 -0.0117499335  0.0073340001 -0.0077731768
## 2004 -0.0028601041 -0.0037724606 -0.0117279125 -0.0202497728 -0.0120933209
## 2005  0.0068661305  0.0092868362  0.0174926519 -0.0411114405 -0.0484531092
## 2006 -0.0520481307 -0.0567219190 -0.0896847416 -0.0545288941 -0.0354108729
## 2007  0.0159389041  0.0685761664  0.0740212035  0.0501526331  0.0619729006
## 2008  0.0148046921  0.0046783734 -0.0141836174 -0.0083201825 -0.0075386289
## 2009  0.0384086945  0.0093775986 -0.0094796301  0.0015067452  0.0026600287
## 2010  0.0175909703  0.0090962475 -0.0125458049 -0.0196599008 -0.0096384268
## 2011  0.1024102649  0.0791350424  0.0729943654  0.0577031676  0.0690610598
## 2012  0.0063405744  0.0245979727  0.0212646530  0.0178208078  0.0322000241
##                Nov           Dec
## 2003 -0.0080819490 -0.0197391298
## 2004 -0.0268122952 -0.0289777699
## 2005 -0.0421064671 -0.0390504700
## 2006 -0.0222695545 -0.0028457955
## 2007  0.0197182959 -0.0024833204
## 2008 -0.0148867797 -0.0172218280
## 2009 -0.0334961249 -0.0166761740
## 2010 -0.0138888451 -0.0043537340
## 2011  0.0524576094  0.0512719380
## 2012  0.0352468848  0.0161751317
```

```r
summary(eg <- ur.df(ry, type=c("none"), lags=1)); plot(eg)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.055822 -0.011250 -0.000467  0.009784  0.100527
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.23926    0.05813  -4.116 7.25e-05 ***
## z.diff.lag  0.17100    0.09127   1.874   0.0635 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.02076 on 116 degrees of freedom
## Multiple R-squared:  0.1299, Adjusted R-squared:  0.1149
## F-statistic: 8.658 on 2 and 116 DF,  p-value: 0.0003128
##
##
## Value of test-statistic is: -4.1159
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```




```r
(eg4 <- Box.test(eg@res, lag = 4, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 0.48381, df = 4, p-value = 0.9751
```

```r
(eg8 <- Box.test(eg@res, lag = 8, type="Ljung") )
```

```
##
##  Box-Ljung test
##
```

```
## data:  eg@res
## X-squared = 3.0104, df = 8, p-value = 0.9337
```

```
(eg12 <- Box.test(eg@res, lag = 12, type="Ljung"))
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 7.08, df = 12, p-value = 0.8523
```

```
(eg16 <- Box.test(eg@res, lag = 16, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 8.6876, df = 16, p-value = 0.9257
```

```
(eg20 <- Box.test(eg@res, lag = 20, type="Ljung"))
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 13.144, df = 20, p-value = 0.8711
```

```
# 3. TAR + Cointegration
# best threshold

t3<-ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=0)
(th.tar <- t3$basic); plot(t3)
```

```
##           Item     tar
## 1          lag   0.000
## 2 thresh final   0.030
## 3 thresh range   0.150
## 4   sse.lowest   0.051
## 5    Total obs 120.000
## 6       CI obs 119.000
## 7    Lower obs  18.000
## 8    Upper obs 102.000
```

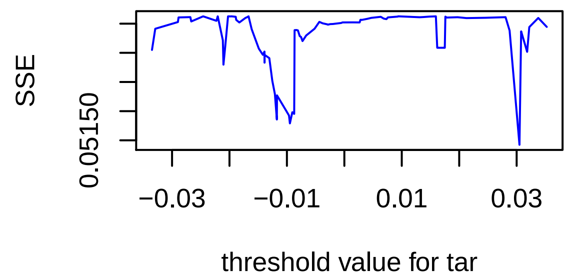
```
ttt<-t3$th.final
for (i in 1:12) { # 20 seconds
t3a <- ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=i)
th.tar[i+2] <- t3a$basic[,2]
}
th.tar
```

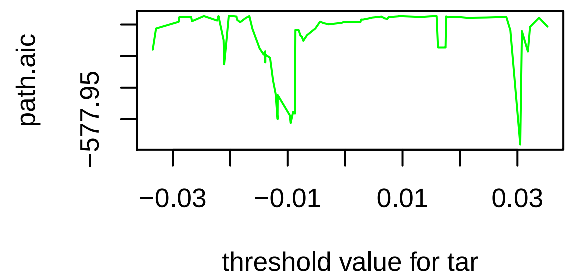
```
##           Item     tar     V3      V4      V5      V6      V7      V8
## 1          lag   0.000   1.00   2.000   3.000   4.000   5.000   6.000
## 2 thresh final   0.030   0.03  -0.021  -0.021  -0.021  -0.021  -0.021
## 3 thresh range   0.150   0.15   0.150   0.150   0.150   0.150   0.150
## 4   sse.lowest   0.051   0.05   0.048   0.048   0.048   0.047   0.047
## 5    Total obs 120.000 120.00 120.000 120.000 120.000 120.000 120.000
## 6       CI obs 119.000 118.00 117.000 116.000 115.000 114.000 113.000
## 7    Lower obs  18.000  18.00  18.000  18.000  18.000  18.000  17.000
## 8    Upper obs 102.000 101.00 100.000  99.000  98.000  97.000  97.000
##        V9     V10     V11     V12     V13     V14
## 1   7.000   8.000   9.000  10.000  11.000  12.000
## 2  -0.021  -0.021  -0.021  -0.021  -0.021  -0.021
## 3   0.150   0.150   0.150   0.150   0.150   0.150
## 4   0.047   0.046   0.045   0.045   0.045   0.045
## 5 120.000 120.000 120.000 120.000 120.000 120.000
## 6 112.000 111.000 110.000 109.000 108.000 107.000
## 7  17.000  17.000  17.000  17.000  17.000  17.000
## 8  96.000  95.000  94.000  93.000  92.000  91.000
```

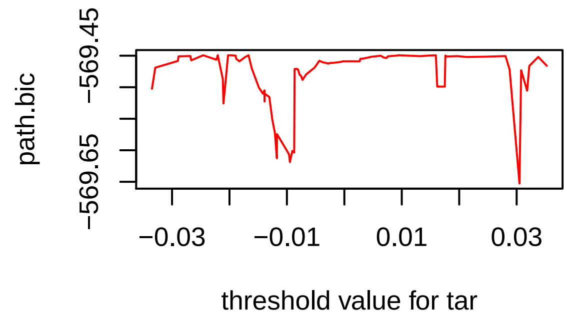
```
t4 <- ciTarThd(PREVGSP, PDISTGSP, model="mtar", lag=0); (th.mtar <- t4$basic)
```

```
##           Item    mtar
## 1          lag   0.000
## 2 thresh final  -0.017
## 3 thresh range   0.150
## 4   sse.lowest   0.050
## 5    Total obs 120.000
## 6       CI obs 118.000
## 7    Lower obs  18.000
## 8    Upper obs 101.000
```

```
plot(t4)
```





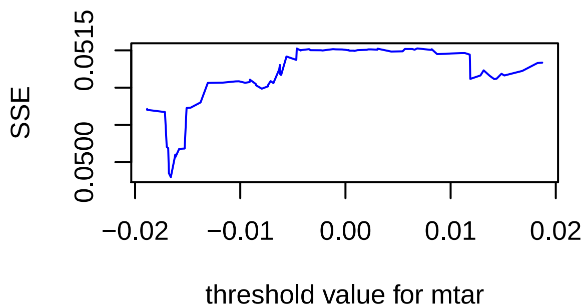


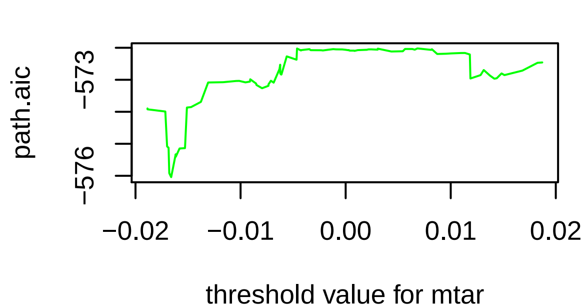

```
mttt<-t4$th.final
for (i in 1:12) {
t4a <- ciTarThd(PREVGSP,PDISTGSP, model="mtar", lag=i)
th.mtar[i+2] <- t4a$basic[,2]
}
th.mtar
```

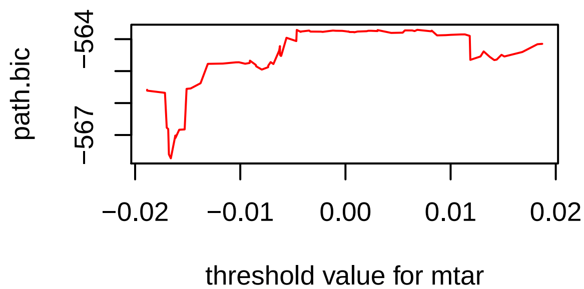
```
##           Item    mtar      V3      V4      V5      V6      V7      V8
## 1          lag   0.000   1.000   2.000   3.000   4.000   5.000   6.000
## 2 thresh final  -0.017  -0.017  -0.017  -0.017  -0.017  -0.017  -0.017
## 3 thresh range   0.150   0.150   0.150   0.150   0.150   0.150   0.150
## 4   sse.lowest   0.050   0.048   0.047   0.047   0.047   0.046   0.046
## 5    Total obs 120.000 120.000 120.000 120.000 120.000 120.000 120.000
## 6       CI obs 118.000 118.000 117.000 116.000 115.000 114.000 113.000
## 7    Lower obs  18.000  18.000  18.000  18.000  18.000  18.000  17.000
## 8    Upper obs 101.000 101.000 100.000  99.000  98.000  97.000  97.000
##         V9     V10     V11     V12     V13     V14
## 1   7.000   8.000   9.000  10.000  11.000  12.000
## 2  -0.017  -0.017  -0.017  -0.017  -0.017  -0.017
## 3   0.150   0.150   0.150   0.150   0.150   0.150
## 4   0.046   0.045   0.045   0.044   0.044   0.044
## 5 120.000 120.000 120.000 120.000 120.000 120.000
## 6 112.000 111.000 110.000 109.000 108.000 107.000
## 7  17.000  17.000  17.000  17.000  17.000  17.000
## 8  96.000  95.000  94.000  93.000  92.000  91.000
```

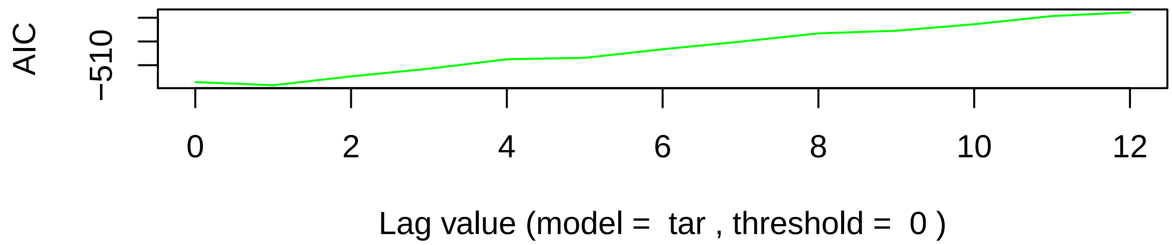
```
t.tar <- ttt; t.mtar <- mttt
#t.tar <- -8.041; t.mtar <- -0.451 # lag = 0 to 4
#t.tar <- -8.701 ; t.mtar <- -0.451 # lag = 5 to 12

mx <- 12
(g1 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh= 0)); plot(g1)
```

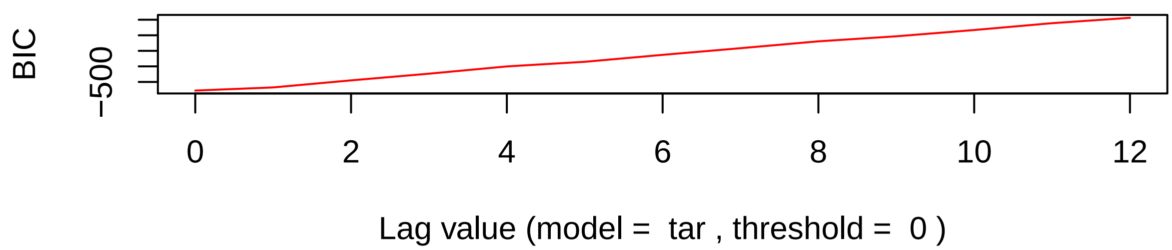
```
##              Item    Value
## 1           model      tar
## 2         max lag       12
## 3       threshold        0
## 4 BestLag.byAic        1
## 5 BestLag.byBic        0
## 6        Best AIC -514.227
## 7        Best BIC -505.545
```





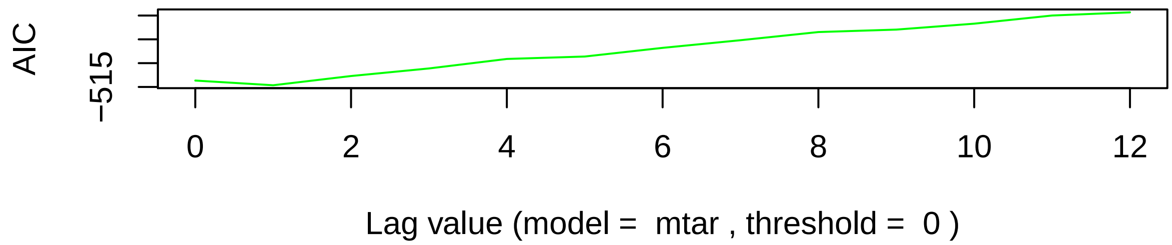


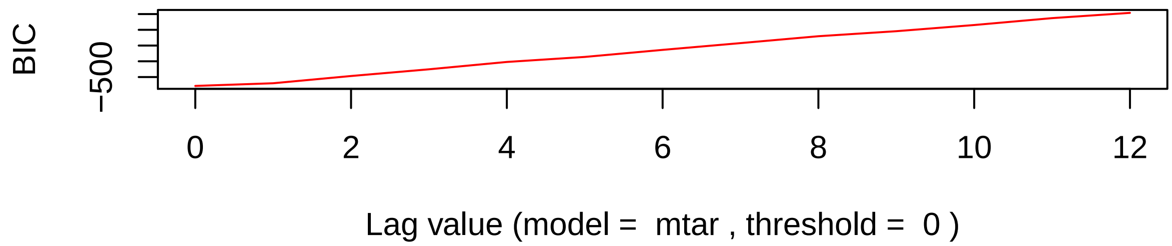





```
(g2 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh= 0)); plot(g2)
```

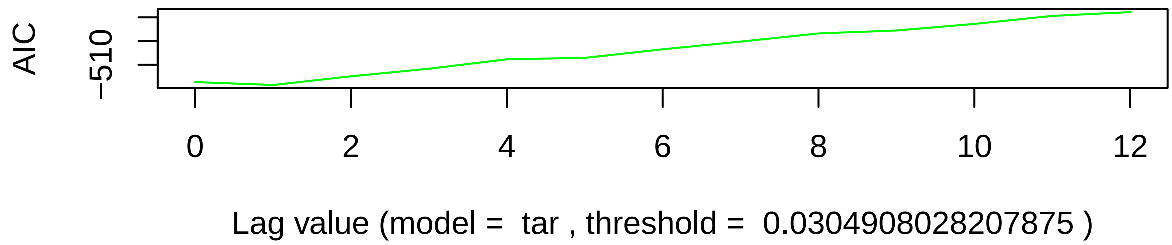
```
##             Item    Value
## 1          model     mtar
## 2        max lag       12
## 3      threshold        0
## 4 BestLag.byAic        1
## 5 BestLag.byBic        0
## 6       Best AIC -514.641
## 7       Best BIC -505.637
```

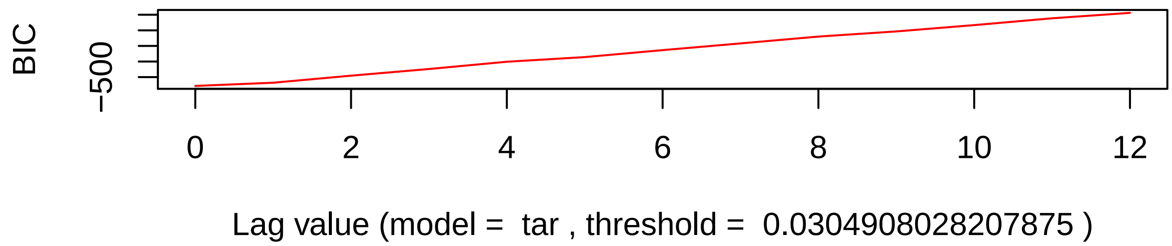





```
(g3 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh=t.tar)); plot(g3)
```

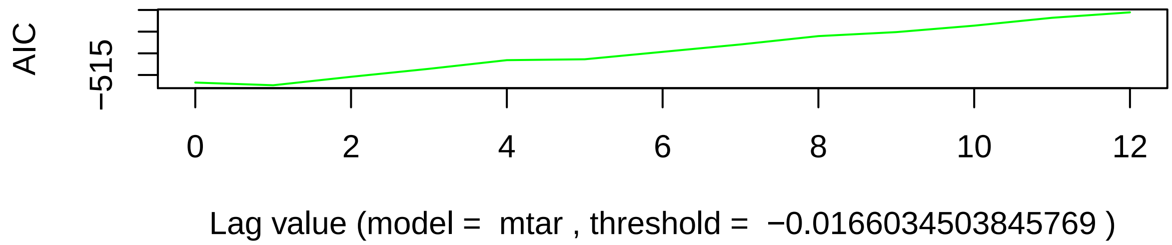
```
##                Item              Value
## 1             model                tar
## 2           max lag                 12
## 3         threshold 0.0304908028207875
## 4 BestLag.byAic                      1
## 5 BestLag.byBic                      0
## 6          Best AIC           -514.283
## 7          Best BIC            -505.65
```

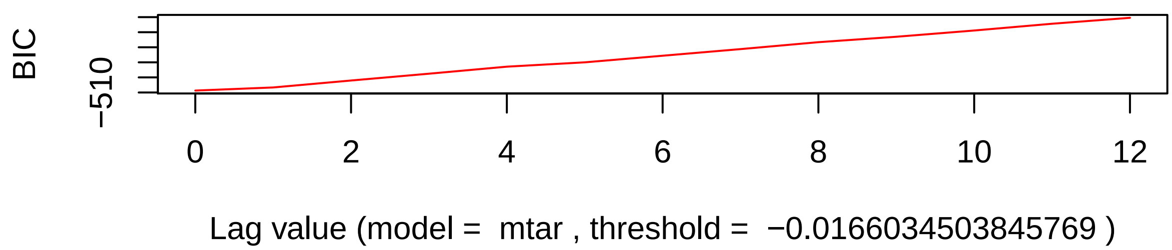





```
(g4 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh=t.mtar)); plot(g4)
```

```
##             Item               Value
## 1          model                mtar
## 2        max lag                  12
## 3      threshold -0.0166034503845769
## 4 BestLag.byAic                    1
## 5 BestLag.byBic                    0
## 6       Best AIC            -517.366
## 7       Best BIC             -508.73
```






```
vv <- 1
(f1 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089685 -0.019867 -0.006965  0.017648  0.102410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16459    0.05024  -3.276  0.00138 **
## PDISTGSP     1.21675    0.02219  54.829  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03492 on 118 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9619
## F-statistic:  3006 on 1 and 118 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
```

```
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.055779 -0.011126 -0.000435  0.009809  0.100556
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.24171    0.07568  -3.194  0.00181 **
## neg.resid.t_1  -0.23602    0.08647  -2.729  0.00734 **
## diff.resid.t_1  0.17098    0.09166   1.865  0.06468 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02085 on 115 degrees of freedom
## Multiple R-squared:  0.1299, Adjusted R-squared:  0.1072
## F-statistic: 5.724 on 3 and 115 DF,  p-value: 0.001096
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
## 1    117 0.057316
## 2    115 0.050011  2 0.0073048 8.3986 0.000394 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    116 0.050012
## 2    115 0.050011  1 1.125e-06 0.0026 0.9595
```

```
(f2 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=t.tar ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.089685 -0.019867 -0.006965  0.017648  0.102410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16459    0.05024  -3.276  0.00138 **
## PDISTGSP     1.21675    0.02219  54.829  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03492 on 118 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9619
## F-statistic:  3006 on 1 and 118 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.055399 -0.011799 -0.000732  0.008876  0.100417
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.21801    0.07885  -2.765  0.00664 **
## neg.resid.t_1  -0.26255    0.08235  -3.188  0.00184 **
## diff.resid.t_1  0.16913    0.09172   1.844  0.06776 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02084 on 115 degrees of freedom
## Multiple R-squared:  0.1311, Adjusted R-squared:  0.1084
## F-statistic: 5.784 on 3 and 115 DF,  p-value: 0.001017
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    117 0.057316
## 2    115 0.049943  2 0.0073734 8.4892 0.0003641 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
```

```
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1    116 0.050012
## 2    115 0.049943  1 6.9741e-05 0.1606 0.6894
```

```
(f3 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089685 -0.019867 -0.006965  0.017648  0.102410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16459    0.05024  -3.276  0.00138 **
## PDISTGSP     1.21675    0.02219  54.829  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03492 on 118 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9619
## F-statistic:  3006 on 1 and 118 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.055226 -0.011426 -0.000433  0.010110  0.099738
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.27762    0.08896  -3.121  0.00228 **
## neg.resid.t_1  -0.21292    0.07436  -2.863  0.00498 **
## diff.resid.t_1  0.18009    0.09291   1.938  0.05502 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02082 on 115 degrees of freedom
## Multiple R-squared:  0.1324, Adjusted R-squared:  0.1097
## F-statistic: 5.847 on 3 and 115 DF,  p-value: 0.0009404
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
```

```
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    117 0.057316
## 2    115 0.049871  2  0.007445 8.5839 0.0003353 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1    116 0.050012
## 2    115 0.049871  1 0.00014132 0.3259 0.5692
```

```
(f4 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=t.mtar))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.089685 -0.019867 -0.006965  0.017648  0.102410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.16459    0.05024  -3.276  0.00138 **
## PDISTGSP     1.21675    0.02219  54.829  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03492 on 118 degrees of freedom
## Multiple R-squared:  0.9622, Adjusted R-squared:  0.9619
## F-statistic:  3006 on 1 and 118 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.056964 -0.011720 -0.000697  0.008802  0.101276
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.17242    0.06649  -2.593 0.010745 *
## neg.resid.t_1  -0.41678    0.10601  -3.932 0.000145 ***
## diff.resid.t_1  0.16768    0.09014   1.860 0.065404 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0205 on 115 degrees of freedom
## Multiple R-squared:  0.1589, Adjusted R-squared:  0.137
## F-statistic: 7.243 on 3 and 115 DF,  p-value: 0.0001708
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    117 0.057316
## 2    115 0.048344  2 0.0089717 10.671 5.611e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1    116 0.050012
## 2    115 0.048344  1  0.001668 3.9678 0.04875 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
r0 <- cbind(summary(f1)$dia, summary(f2)$dia, summary(f3)$dia,
summary(f4)$dia)
diag <- r0[c(1:4, 6:7, 12:14, 8, 9, 11), c(1,2,4,6,8)]
rownames(diag) <- 1:nrow(diag); diag
```

```
##          item     tar   c.tar    mtar  c.mtar
## 1         lag   1.000   1.000   1.000   1.000
## 2      thresh   0.000   0.030   0.000  -0.017
## 3   total obs 120.000 120.000 120.000 120.000
```

```
## 4    coint obs  118.000  118.000  118.000  118.000
## 5          aic -573.541 -573.703 -573.872 -577.541
## 6          bic -562.458 -562.620 -562.789 -566.458
## 7   LB test(4)    0.975    0.973    0.972    0.974
## 8   LB test(8)    0.934    0.936    0.948    0.885
## 9  LB test(12)    0.852    0.858    0.845    0.948
## 10    H1: no CI    8.399    8.489    8.584   10.671
## 11   H2: no APT    0.003    0.161    0.326    3.968
## 12 H2: p.value    0.960    0.689    0.569    0.049
```

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

```
##     model reg       variable estimate st.error t.value p.value sign
## 1     tar  LR    (Intercept)   -0.165    0.050  -3.276   0.001  ***
## 2     tar  LR       PDISTGSP    1.217    0.022  54.829   0.000  ***
## 3     tar  CI  pos.resid.t_1   -0.242    0.076  -3.194   0.002  ***
## 4     tar  CI  neg.resid.t_1   -0.236    0.086  -2.729   0.007  ***
## 5     tar  CI diff.resid.t_1    0.171    0.092   1.865   0.065    *
## 6   c.tar  LR    (Intercept)   -0.165    0.050  -3.276   0.001  ***
## 7   c.tar  LR       PDISTGSP    1.217    0.022  54.829   0.000  ***
## 8   c.tar  CI  pos.resid.t_1   -0.218    0.079  -2.765   0.007  ***
## 9   c.tar  CI  neg.resid.t_1   -0.263    0.082  -3.188   0.002  ***
## 10  c.tar  CI diff.resid.t_1    0.169    0.092   1.844   0.068    *
## 11   mtar  LR    (Intercept)   -0.165    0.050  -3.276   0.001  ***
## 12   mtar  LR       PDISTGSP    1.217    0.022  54.829   0.000  ***
## 13   mtar  CI  pos.resid.t_1   -0.278    0.089  -3.121   0.002  ***
## 14   mtar  CI  neg.resid.t_1   -0.213    0.074  -2.863   0.005  ***
## 15   mtar  CI diff.resid.t_1    0.180    0.093   1.938   0.055    *
## 16 c.mtar  LR    (Intercept)   -0.165    0.050  -3.276   0.001  ***
## 17 c.mtar  LR       PDISTGSP    1.217    0.022  54.829   0.000  ***
## 18 c.mtar  CI  pos.resid.t_1   -0.172    0.066  -2.593   0.011   **
## 19 c.mtar  CI  neg.resid.t_1   -0.417    0.106  -3.932   0.000  ***
## 20 c.mtar  CI diff.resid.t_1    0.168    0.090   1.860   0.065    *
```

```
ee <- list(e1, e2, e3, e4); vect <- NULL
for (i in 1:4) {
ef <- data.frame(ee[i])
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) <- colnames(diag)
( res.CI <- rbind(diag, ve)[c(1:2, 13:16, 3:12), ] )
```

```
##           item       tar     c.tar      mtar    c.mtar
## 1          lag         1         1         1         1
## 2       thresh         0      0.03         0    -0.017
## 13   pos.coeff -0.242*** -0.218*** -0.278***  -0.172**
## 14 pos.t.value  (-3.194)  (-2.765)  (-3.121)  (-2.593)
```

```
## 15   neg.coeff -0.236*** -0.263*** -0.213*** -0.417***
## 16 neg.t.value  (-2.729)  (-3.188)  (-2.863)  (-3.932)
## 3    total obs       120       120       120       120
## 4    coint obs       118       118       118       118
## 5          aic  -573.541  -573.703  -573.872  -577.541
## 6          bic  -562.458   -562.62  -562.789  -566.458
## 7   LB test(4)     0.975     0.973     0.972     0.974
## 8   LB test(8)     0.934     0.936     0.948     0.885
## 9  LB test(12)     0.852     0.858     0.845     0.948
## 10   H1: no CI     8.399     8.489     8.584    10.671
## 11  H2: no APT     0.003     0.161     0.326     3.968
## 12 H2: p.value      0.96     0.689     0.569     0.049
```

```r
rownames(res.CI) <- 1:nrow(res.CI)




(sem <- ecmSymFit(y=PREVGSP, x=PDISTGSP, lag=1)); names(sem)
```

```
##
## ===============================================================
## ECM - Symmetric + linear cointegration -  "PDISTGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.161741 -0.017627 -0.001979  0.014581  0.178151
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.001079   0.003725   0.290   0.7725
## X.diff.PDISTGSP.t_1  0.428080   0.216878   1.974   0.0508 .
## X.diff.PREVGSP.t_1  -0.137791   0.208001  -0.662   0.5090
## X.ECT.t_1            0.101536   0.117393   0.865   0.3889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04039 on 114 degrees of freedom
## Multiple R-squared:  0.08341,    Adjusted R-squared:  0.05929
## F-statistic: 3.458 on 3 and 114 DF,  p-value: 0.01881
##
##
## ===============================================================
## ECM - Symmetric + linear cointegration -  "PREVGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.127123 -0.019776 -0.002289  0.015540  0.176282
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.001142   0.003674   0.311    0.757
## X.diff.PDISTGSP.t_1  0.316805   0.213932   1.481    0.141
## X.diff.PREVGSP.t_1  -0.018981   0.205175  -0.093    0.926
## X.ECT.t_1           -0.122307   0.115798  -1.056    0.293
##
## Residual standard error: 0.03984 on 114 degrees of freedom
## Multiple R-squared:  0.1248, Adjusted R-squared:  0.1017
## F-statistic: 5.417 on 3 and 114 DF,  p-value: 0.001609

## [1] "y"      "x"      "lag"    "data"   "IndVar" "name.y" "name.x" "ecm.y"
## [9] "ecm.x"
```

```r
aem <- ecmAsyFit(y=PREVGSP, x=PDISTGSP,lag=1, model="mtar", split=TRUE, thresh=t.mtar)
aem
```

```
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PDISTGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.136486 -0.021603  0.000531  0.015296  0.172493
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -6.176e-05  5.288e-03  -0.012  0.99070
## X.diff.PDISTGSP.t_1.pos  1.062e+00  3.727e-01   2.850  0.00522 **
## X.diff.PDISTGSP.t_1.neg -1.881e-02  3.046e-01  -0.062  0.95087
## X.diff.PREVGSP.t_1.pos  -6.390e-01  3.406e-01  -1.876  0.06322 .
## X.diff.PREVGSP.t_1.neg   2.685e-01  3.462e-01   0.776  0.43964
## X.ECT.t_1.pos            2.906e-03  1.337e-01   0.022  0.98269
## X.ECT.t_1.neg            4.725e-01  2.322e-01   2.035  0.04421 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03997 on 111 degrees of freedom
## Multiple R-squared:  0.1259, Adjusted R-squared:  0.07864
## F-statistic: 2.664 on 6 and 111 DF,  p-value: 0.01877
##
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PREVGSP"
## ===============================================================
##
```

```
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12959 -0.02144 -0.00360  0.01862  0.17281
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.0001042  0.0052681  -0.020   0.9842
## X.diff.PDISTGSP.t_1.pos   0.8469670  0.3713493   2.281   0.0245 *
## X.diff.PDISTGSP.t_1.neg -0.0479517  0.3034622  -0.158   0.8747
## X.diff.PREVGSP.t_1.pos  -0.4391675  0.3392857  -1.294   0.1982
## X.diff.PREVGSP.t_1.neg   0.2985251  0.3449000   0.866   0.3886
## X.ECT.t_1.pos           -0.1615212  0.1331530  -1.213   0.2277
## X.ECT.t_1.neg            0.0660819  0.2312790   0.286   0.7756
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03982 on 111 degrees of freedom
## Multiple R-squared:  0.1486, Adjusted R-squared:  0.1026
## F-statistic: 3.229 on 6 and 111 DF,  p-value: 0.005815
```

```
(ccc <- summary(aem))
```

```
##               DepVar                  IndVar   estimate error t.value
## 1  diff.PDISTGSP.t_0  |             (Intercept)    0.000 0.005  -0.012
## 2                     | X.diff.PDISTGSP.t_1.pos    1.062 0.373   2.850
## 3                     | X.diff.PDISTGSP.t_1.neg   -0.019 0.305  -0.062
## 4                     |  X.diff.PREVGSP.t_1.pos   -0.639 0.341  -1.876
## 5                     |  X.diff.PREVGSP.t_1.neg    0.269 0.346   0.776
## 6                     |           X.ECT.t_1.pos    0.003 0.134   0.022
## 7                     |           X.ECT.t_1.neg    0.472 0.232   2.035
## 8   diff.PREVGSP.t_0  -             (Intercept)    0.000 0.005  -0.020
## 9                     - X.diff.PDISTGSP.t_1.pos    0.847 0.371   2.281
## 10                    - X.diff.PDISTGSP.t_1.neg   -0.048 0.303  -0.158
## 11                    -  X.diff.PREVGSP.t_1.pos   -0.439 0.339  -1.294
## 12                    -  X.diff.PREVGSP.t_1.neg    0.299 0.345   0.866
## 13                    -           X.ECT.t_1.pos   -0.162 0.133  -1.213
## 14                    -           X.ECT.t_1.neg    0.066 0.231   0.286
##    p.value signif
## 1    0.991
## 2    0.005    ***
## 3    0.951
## 4    0.063      *
## 5    0.440
## 6    0.983
## 7    0.044     **
## 8    0.984
## 9    0.024     **
## 10   0.875
## 11   0.198
## 12   0.389
## 13   0.228
```

```
## 14   0.776
```

```r
(edia <- ecmDiag(aem, 3))
```

```
##          item PDISTGSP  PREVGSP
## 1   R-squared    0.126    0.149
## 2      Adj-R2    0.079    0.103
## 3      F-stat    2.664    3.229
## 4     Stat DW    1.924    1.889
## 5  p-value DW    0.608    0.416
## 6         AIC -416.172 -417.059
## 7         BIC -394.006 -394.893
## 8       LB(4)    0.820    0.632
## 9       LB(8)    0.789    0.325
## 10     LB(12)    0.930    0.627
```

```r
(tes <- ecmAsyTest(aem)$out)
```

```
##          Hypothesis description|
## 1 H1: Equ adjust path asymmetry|
## 2    H2: Granger causality test|
## 3    H2: Granger causality test|
## 4 H3: Distributed lag asymmetry|
## 5 H3: Distributed lag asymmetry|
## 6      H4: Cumulative asymmetry|
## 7      H4: Cumulative asymmetry|
##                                                     Expression
## 1                                  X.ECT.t_1.pos=X.ECT.t_1.neg
## 2                         PDISTGSP (x) does not Granger cause...
## 3                          PREVGSP (y) does not Granger cause...
## 4           X.diff.PDISTGSP.t_1.pos = X.diff.PDISTGSP.t_1.neg
## 5             X.diff.PREVGSP.t_1.pos = X.diff.PREVGSP.t_1.neg
## 6 Cumulative positive PDISTGSP = Cumulative negative PDISTGSP
## 7   Cumulative positive PREVGSP = Cumulative negative PREVGSP
##   PDISTGSP.F.Stat PREVGSP.F.Stat PDISTGSP.P.Value PREVGSP.P.Value
## 1           3.135          0.742            0.079           0.391
## 2           4.145          2.625            0.018           0.077
## 3           1.824          1.009            0.166           0.368
## 4           4.345          3.000            0.039           0.086
## 5           2.841          1.892            0.095           0.172
## 6           4.345          3.000            0.039           0.086
## 7           2.841          1.892            0.095           0.172
##   PDISTGSP.Sig PREVGSP.Sig
## 1            *
## 2           **           *
## 3
## 4           **           *
## 5            *
## 6           **           *
## 7            *
```

```r
###################################2003-2007#######################



load("C:/Users/Lucas/Desktop/UFPR/2º Sem/Macroeconometria/ArtigoMacroeconometria/precos1.RData")

SP <- precos1[grep("GUARULHOS", precos1$MUNICIPIO), ]

GSP <- SP[grep("GASOLINA COMUM", SP$PRODUTO), ]
View(GSP)



PDISTGSP<-ts(GSP$PRECOMEDIODISTRIBUICAO,frequency=12,start=c(2003,1),end=c(2007,12))
head(PDISTGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 1.8303 1.8862 1.9001 1.8958 1.8104
```

```r
PREVGSP<-ts(GSP$PRECOMEDIOREVENDA,frequency=12,start=c(2003,1),end=c(2007,12))
head(PREVGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 2.1238 2.1777 2.1545 2.1446 2.0335
```

```r
INPC <- as.zoo(ts(GSP$INPCFEV2016100, frequency=12,start=c(2003,1),end=c(2007,12)))
head(INPC)
```

```
## 2003(1) 2003(2) 2003(3) 2003(4) 2003(5) 2003(6)
##    87.0    86.4    86.1    85.7    85.4    85.5
```

```r
PREVGSP <- PREVGSP*100
head(PREVGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 212.38 217.77 215.45 214.46 203.35
```

```r
PREVGSP <- PREVGSP/INPC
```

```
## Warning: Métodos incompatíveis ("Ops.ts", "Ops.zoo") para "/"
```

```r
head(PREVGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2003 2.441149 2.520486 2.502323 2.502450 2.381148
```

```r
PDISTGSP <- PDISTGSP*100
head(PDISTGSP)
```

```
##         Jan    Feb    Mar    Apr    May
## 2003 183.03 188.62 190.01 189.58 181.04
```

```r
PDISTGSP <- PDISTGSP/INPC
```

```
## Warning: Métodos incompatíveis ("Ops.ts", "Ops.zoo") para "/"
```

```r
head(PDISTGSP)
```

```
##          Jan      Feb      Mar      Apr      May
## 2003 2.103793 2.183102 2.206852 2.212135 2.119906
```

```r
#2. Stationarity tests
precoadfGREV=ur.df(PREVGSP,type= "trend", selectlags= "AIC")
precoglsGREV=ur.ers(PREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGREV@lags
```

```
## [1] 1
```

```r
precoadfGREV@teststat[1]
```

```
## [1] -2.775685
```

```r
precoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
precoglsGREV@teststat[1]
```

```
## [1] -2.073077
```

```r
precoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
precoadfGDIST=ur.df(PDISTGSP,type= "trend", selectlags= "AIC")
precoglsGDIST=ur.ers(PDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGDIST@lags
```

```
## [1] 1
```

```r
precoadfGDIST@teststat[1]
```

```
## [1] -2.662713
```

```r
precoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
precoglsGDIST@teststat[1]
```

```
## [1] -2.344821
```

```r
precoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
diffPREVGSP <- diff(PREVGSP, lag = 1, differences = 1)
diffPDISTGSP <- diff(PDISTGSP, lag = 1, differences = 1)


diffprecoadfGREV=ur.df(diffPREVGSP,type= "trend", selectlags= "AIC")
diffprecoglsGREV=ur.ers(diffPREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGREV@lags
```

```
## [1] 1
```

```r
diffprecoadfGREV@teststat[1]
```

```
## [1] -4.41284
```

```r
diffprecoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
diffprecoglsGREV@teststat[1]
```

```
## [1] -3.841686
```

```r
diffprecoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
diffprecoadfGDIST=ur.df(diffPDISTGSP,type= "trend", selectlags= "AIC")
diffprecoglsGDIST=ur.ers(diffPDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGDIST@lags
```

```
## [1] 1
```

```r
diffprecoadfGDIST@teststat[1]
```

```
## [1] -4.849116
```

```r
diffprecoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
diffprecoglsGDIST@teststat[1]
```

```
## [1] -4.103865
```

```r
diffprecoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
# 2. EG cointegration

LR <- lm(PREVGSP ~ PDISTGSP); summary(LR)
```

```
##
## Call:
## lm(formula = PREVGSP ~ PDISTGSP)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056192 -0.020848 -0.008079  0.013024  0.090444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006219   0.051904   -0.12    0.905
## PDISTGSP     1.139056   0.023565   48.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03027 on 58 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9754
## F-statistic:  2336 on 1 and 58 DF,  p-value: < 2.2e-16
```

```r
(LR.coef <- round(summary(LR)$coefficients, 6))
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -0.006219   0.051904 -0.11982 0.905039
## PDISTGSP     1.139056   0.023565 48.33635 0.000000
```

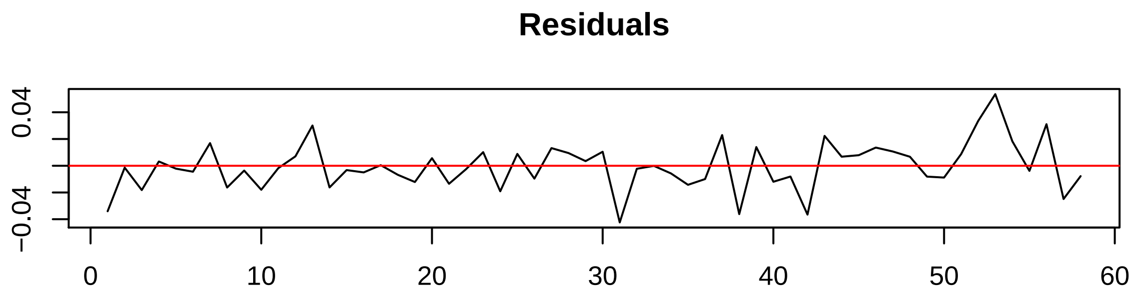
```r
(ry <- ts(residuals(LR), start=start(PDISTGSP), end=end(PDISTGSP), frequency =12))
```

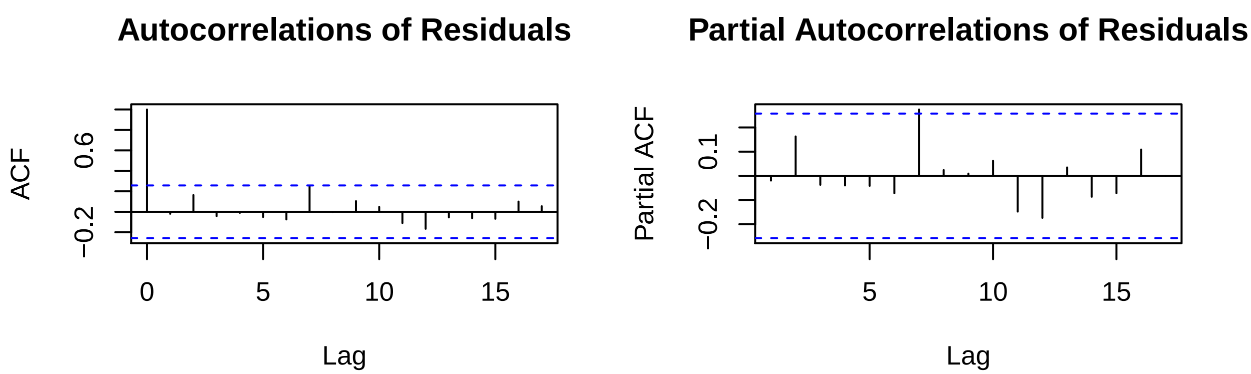
```
##               Jan           Feb           Mar           Apr           May
## 2003  0.0510297666  0.0400293215 -0.0051872318 -0.0110771774 -0.0273259825
## 2004 -0.0250757691 -0.0115434417  0.0230913358  0.0056128789 -0.0012280569
## 2005 -0.0044960597 -0.0206124683 -0.0087714045 -0.0147754223  0.0012508650
## 2006 -0.0299865895 -0.0338064627 -0.0031076336 -0.0346579735 -0.0162405681
## 2007  0.0263484899  0.0276569248  0.0129566765 -0.0009154679  0.0062532838
##               Jun           Jul           Aug           Sep           Oct
## 2003 -0.0195526459 -0.0159775793 -0.0160934159  0.0047521393 -0.0100616125
## 2004 -0.0067818428 -0.0053947351 -0.0106874523 -0.0209213261 -0.0114661907
## 2005  0.0124598904  0.0142903813  0.0214963259 -0.0252301217 -0.0272844248
## 2006 -0.0219299747 -0.0253783030 -0.0561915241 -0.0240762971 -0.0073865097
## 2007  0.0391840527  0.0873654169  0.0904442600  0.0649399766  0.0769503701
##               Nov           Dec
## 2003 -0.0131067494 -0.0282716390
## 2004 -0.0209696663 -0.0196099616
## 2005 -0.0209926726 -0.0208240492
## 2006  0.0044122768  0.0184152270
## 2007  0.0348298728  0.0132266706
```

```r
summary(eg <- ur.df(ry, type=c("none"), lags=1)); plot(eg)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.042401 -0.009885 -0.002228  0.009327  0.053558
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -0.2437     0.0836  -2.915  0.00511 **
## z.diff.lag   0.1267     0.1315   0.964  0.33925
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01792 on 56 degrees of freedom
## Multiple R-squared:  0.1319, Adjusted R-squared:  0.1009
## F-statistic: 4.254 on 2 and 56 DF,  p-value: 0.01906
##
##
## Value of test-statistic is: -2.9149
##
```

```
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```






```
(eg4 <- Box.test(eg@res, lag = 4, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 1.7988, df = 4, p-value = 0.7727
```

```
(eg8 <- Box.test(eg@res, lag = 8, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 6.7257, df = 8, p-value = 0.5665
```

```
(eg12 <- Box.test(eg@res, lag = 12, type="Ljung"))
```

```
##
```

```
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 10.656, df = 12, p-value = 0.5586
```

```
(eg16 <- Box.test(eg@res, lag = 16, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 12.415, df = 16, p-value = 0.7149
```

```
(eg20 <- Box.test(eg@res, lag = 20, type="Ljung"))
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 14.005, df = 20, p-value = 0.8302
```

```
# 3. TAR + Cointegration
# best threshold

t3<-ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=0)
(th.tar <- t3$basic); plot(t3)
```

```
##           Item    tar
## 1          lag  0.000
## 2 thresh final -0.011
## 3 thresh range  0.150
## 4   sse.lowest  0.018
## 5    Total obs 60.000
## 6       CI obs 59.000
## 7    Lower obs  9.000
## 8    Upper obs 51.000
```

```
ttt<-t3$th.final
for (i in 1:12) { # 20 seconds
t3a <- ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=i)
th.tar[i+2] <- t3a$basic[,2]
}
th.tar
```

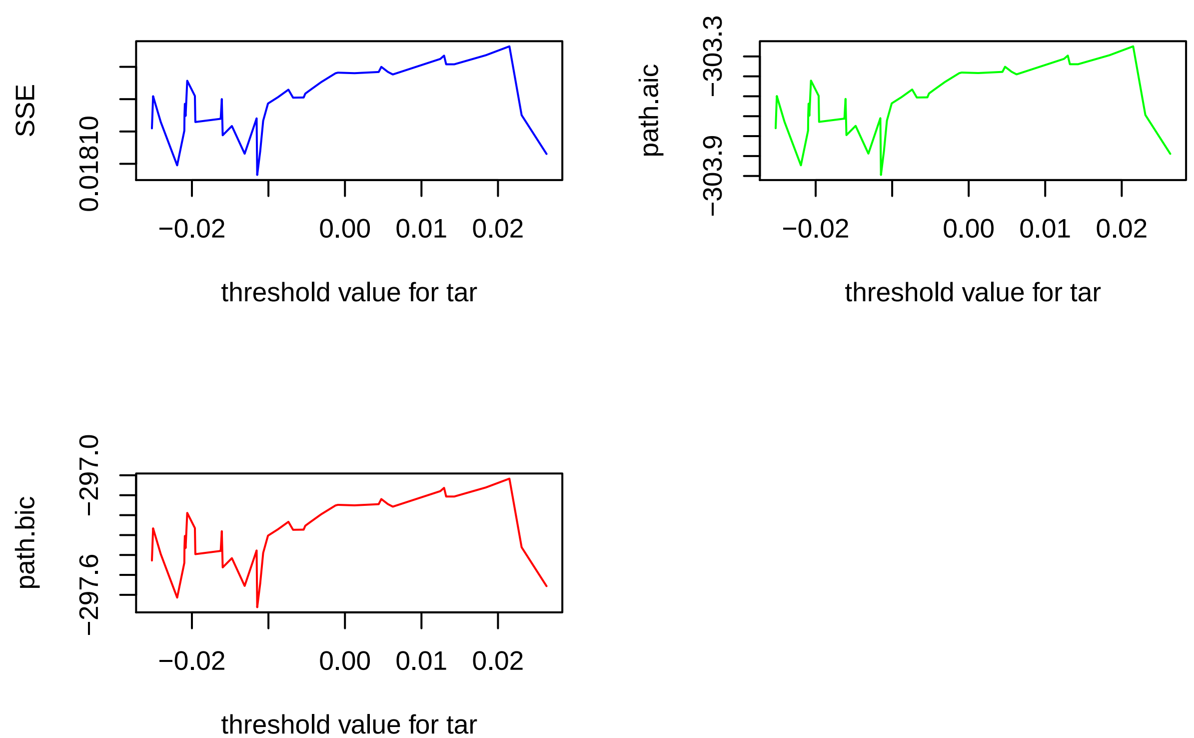
```
##           Item    tar     V3     V4     V5     V6     V7     V8     V9
## 1          lag  0.000  1.000  2.000  3.000  4.000  5.000  6.000  7.000
## 2 thresh final -0.011 -0.022 -0.022 -0.022 -0.022 -0.022 -0.025 -0.025
## 3 thresh range  0.150  0.150  0.150  0.150  0.150  0.150  0.150  0.150
## 4   sse.lowest  0.018  0.018  0.016  0.016  0.016  0.015  0.014  0.013
## 5    Total obs 60.000 60.000 60.000 60.000 60.000 60.000 60.000 60.000
## 6       CI obs 59.000 58.000 57.000 56.000 55.000 54.000 53.000 52.000
```

```
## 7    Lower obs  9.000  9.000  9.000  9.000  9.000  9.000  8.000  8.000
## 8    Upper obs 51.000 50.000 49.000 48.000 47.000 46.000 46.000 45.000
##       V10    V11    V12    V13    V14
## 1  8.000  9.000 10.000 11.000 12.000
## 2 -0.025 -0.025 -0.025 -0.025 -0.025
## 3  0.150  0.150  0.150  0.150  0.150
## 4  0.013  0.012  0.012  0.011  0.011
## 5 60.000 60.000 60.000 60.000 60.000
## 6 51.000 50.000 49.000 48.000 47.000
## 7  8.000  8.000  8.000  8.000  8.000
## 8 44.000 43.000 42.000 41.000 40.000
```

```
t4 <- ciTarThd(PREVGSP, PDISTGSP, model="mtar", lag=0); (th.mtar <- t4$basic)
```

```
##           Item   mtar
## 1          lag  0.000
## 2 thresh final  0.014
## 3 thresh range  0.150
## 4   sse.lowest  0.016
## 5    Total obs 60.000
## 6       CI obs 58.000
## 7    Lower obs  9.000
## 8    Upper obs 50.000
```

```
plot(t4)
```

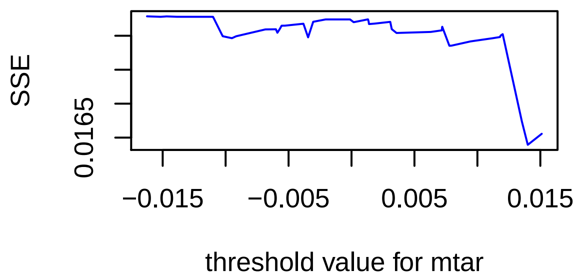


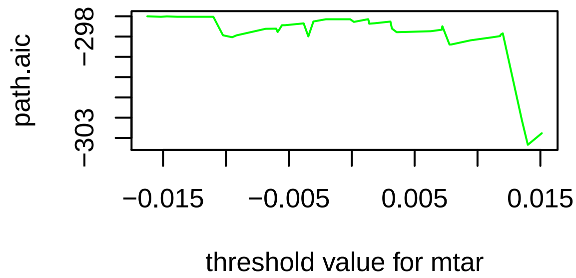
```r
mttt<-t4$th.final
for (i in 1:12) {
t4a <- ciTarThd(PREVGSP,PDISTGSP, model="mtar", lag=i)
th.mtar[i+2] <- t4a$basic[,2]
}
th.mtar
```

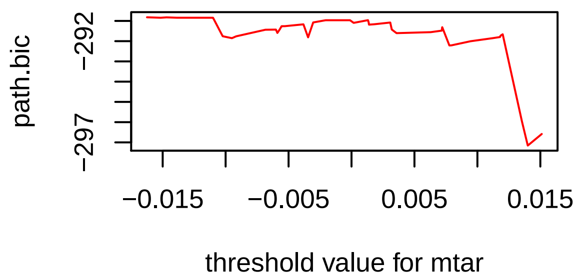
```
##           Item   mtar     V3     V4     V5     V6     V7     V8     V9
## 1          lag  0.000  1.000  2.000  3.000  4.000  5.000  6.000  7.000
## 2 thresh final  0.014  0.014  0.014  0.014  0.008  0.008  0.008  0.008
## 3 thresh range  0.150  0.150  0.150  0.150  0.150  0.150  0.150  0.150
## 4   sse.lowest  0.016  0.016  0.015  0.015  0.016  0.016  0.015  0.014
## 5    Total obs 60.000 60.000 60.000 60.000 60.000 60.000 60.000 60.000
## 6       CI obs 58.000 58.000 57.000 56.000 55.000 54.000 53.000 52.000
## 7    Lower obs  9.000  9.000  9.000  9.000  9.000  9.000  8.000  8.000
## 8    Upper obs 50.000 50.000 49.000 48.000 47.000 46.000 46.000 45.000
##      V10    V11    V12    V13    V14
## 1  8.000  9.000 10.000 11.000 12.000
## 2  0.001  0.001  0.001  0.001  0.001
## 3  0.150  0.150  0.150  0.150  0.150
## 4  0.013  0.013  0.012  0.011  0.011
## 5 60.000 60.000 60.000 60.000 60.000
## 6 51.000 50.000 49.000 48.000 47.000
## 7  8.000  8.000  8.000  8.000  8.000
## 8 44.000 43.000 42.000 41.000 40.000
```

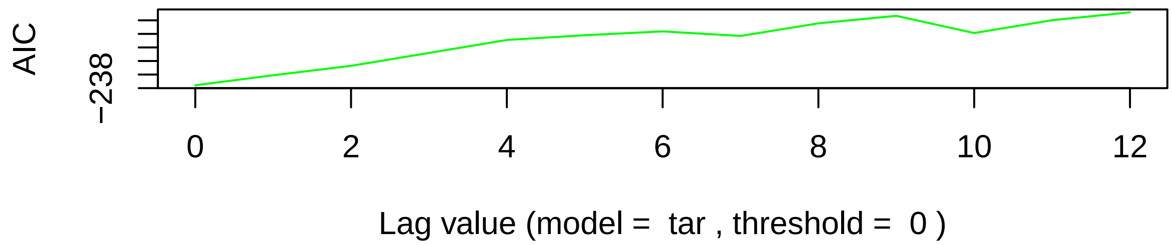
```r
t.tar <- ttt; t.mtar <- mttt
#t.tar <- -8.041; t.mtar <- -0.451 # lag = 0 to 4
#t.tar <- -8.701 ; t.mtar <- -0.451 # lag = 5 to 12

mx <- 12
(g1 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh= 0)); plot(g1)
```

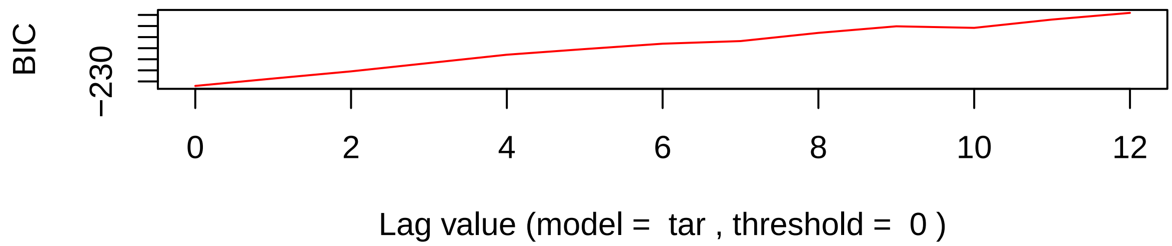
```
##             Item    Value
## 1          model      tar
## 2        max lag       12
## 3      threshold        0
## 4 BestLag.byAic        0
## 5 BestLag.byBic        0
## 6       Best AIC -237.575
## 7       Best BIC -232.024
```


SSE
0.0165
−0.015
−0.005
0.005
0.015
threshold value for mtar


path.aic
−298
−303
−0.015
−0.005
0.005
0.015
threshold value for mtar

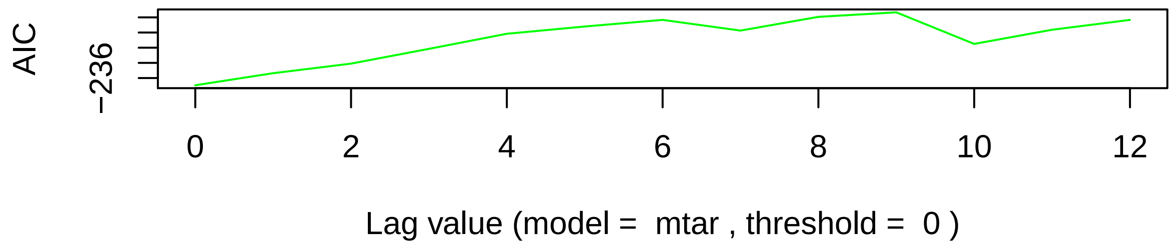

path.bic
−292
−297
−0.015
−0.005
0.005
0.015
threshold value for mtar

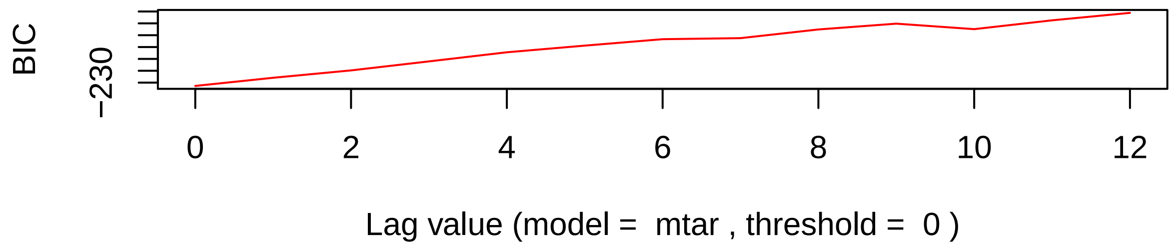





```
(g2 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh= 0)); plot(g2)
```

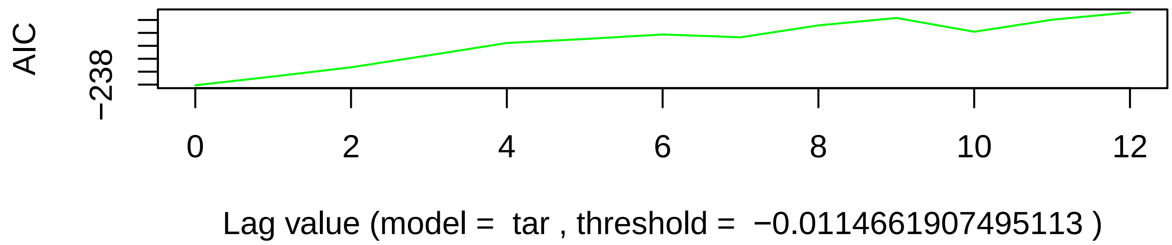
```
##              Item    Value
## 1           model     mtar
## 2         max lag       12
## 3       threshold        0
## 4 BestLag.byAic        0
## 5 BestLag.byBic        0
## 6        Best AIC -236.946
## 7        Best BIC -231.396
```

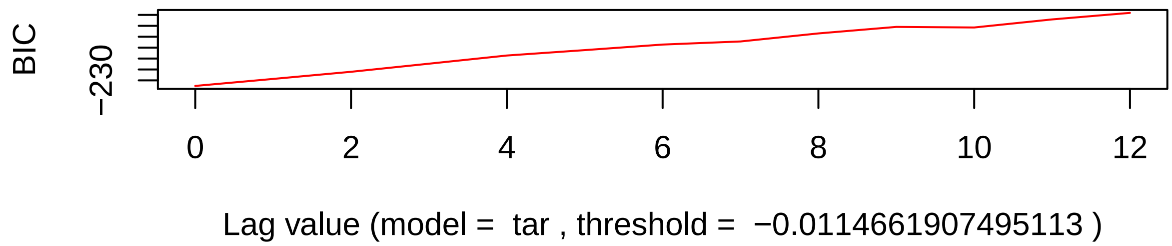





```
(g3 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh=t.tar)); plot(g3)
```

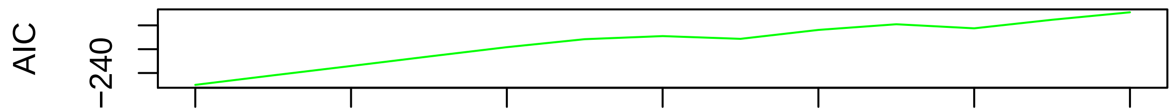
```
##               Item                Value
## 1            model                  tar
## 2          max lag                   12
## 3        threshold -0.0114661907495113
## 4 BestLag.byAic                       0
## 5 BestLag.byBic                       0
## 6         Best AIC             -238.096
## 7         Best BIC             -232.545
```

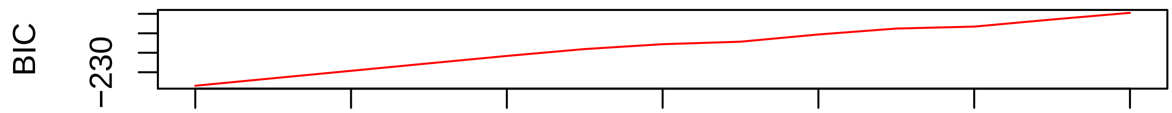





```
(g4 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh=t.mtar)); plot(g4)
```

```
##             Item              Value
## 1          model               mtar
## 2        max lag                 12
## 3      threshold 0.0140029502347804
## 4 BestLag.byAic                   0
## 5 BestLag.byBic                   0
## 6       Best AIC           -242.492
## 7       Best BIC           -236.941
```






```
vv <- 1
(f1 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056192 -0.020848 -0.008079  0.013024  0.090444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006219   0.051904   -0.12    0.905
## PDISTGSP     1.139056   0.023565   48.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03027 on 58 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9754
## F-statistic:  2336 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
```

```
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043032 -0.011303 -0.003347  0.008941  0.052257
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.21693    0.09997  -2.170   0.0343 *
## neg.resid.t_1  -0.30233    0.14507  -2.084   0.0418 *
## diff.resid.t_1  0.13437    0.13325   1.008   0.3177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01804 on 55 degrees of freedom
## Multiple R-squared:  0.1358, Adjusted R-squared:  0.08861
## F-statistic:  2.88 on 3 and 55 DF,  p-value: 0.04407
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     57 0.020718
## 2     55 0.017908  2 0.0028095 4.3142 0.01818 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1     56 0.017989
## 2     55 0.017908  1 8.0189e-05 0.2463 0.6217
```

```r
(f2 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=t.tar ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.056192 -0.020848 -0.008079  0.013024  0.090444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006219   0.051904   -0.12    0.905
## PDISTGSP     1.139056   0.023565   48.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03027 on 58 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9754
## F-statistic:  2336 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043541 -0.011297 -0.003844  0.008714  0.051179
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1  -0.19572    0.09836  -1.990   0.0516 .
## neg.resid.t_1  -0.35671    0.14768  -2.415   0.0191 *
## diff.resid.t_1  0.14185    0.13263   1.070   0.2895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01794 on 55 degrees of freedom
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.09867
## F-statistic: 3.117 on 3 and 55 DF,  p-value: 0.03338
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1     57 0.020718
## 2     55 0.017711  2 0.0030071 4.6693 0.0134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
```

```
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1     56 0.017989
## 2     55 0.017711  1 0.00027784 0.8628  0.357
```

```
(f3 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056192 -0.020848 -0.008079  0.013024  0.090444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006219   0.051904   -0.12    0.905
## PDISTGSP     1.139056   0.023565   48.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03027 on 58 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9754
## F-statistic:  2336 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043064 -0.010878 -0.003089  0.009271  0.052403
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   -0.2119     0.1073  -1.976   0.0532 .
## neg.resid.t_1   -0.2904     0.1291  -2.249   0.0285 *
## diff.resid.t_1   0.1240     0.1325   0.936   0.3534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01805 on 55 degrees of freedom
## Multiple R-squared:  0.1355, Adjusted R-squared:  0.08831
## F-statistic: 2.873 on 3 and 55 DF,  p-value: 0.04444
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
```

```
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     57 0.020718
## 2     55 0.017914  2 0.0028035 4.3037 0.01835 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1     56 0.017989
## 2     55 0.017914  1 7.4236e-05 0.2279  0.635
```

```
(f4 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=t.mtar))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056192 -0.020848 -0.008079  0.013024  0.090444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006219   0.051904   -0.12    0.905
## PDISTGSP     1.139056   0.023565   48.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03027 on 58 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9754
## F-statistic:  2336 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.040036 -0.010279 -0.004804  0.008619  0.041748
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   0.12392    0.17623   0.703 0.484906
## neg.resid.t_1  -0.32730    0.08799  -3.720 0.000469 ***
## diff.resid.t_1  0.04790    0.13087   0.366 0.715742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01724 on 55 degrees of freedom
## Multiple R-squared:  0.2108, Adjusted R-squared:  0.1677
## F-statistic: 4.896 on 3 and 55 DF,  p-value: 0.004358
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
## 1     57 0.020718
## 2     55 0.016354  2 0.0043636 7.3376 0.001497 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     56 0.017989
## 2     55 0.016354  1 0.0016343 5.4964 0.02269 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
r0 <- cbind(summary(f1)$dia, summary(f2)$dia, summary(f3)$dia,
summary(f4)$dia)
diag <- r0[c(1:4, 6:7, 12:14, 8, 9, 11), c(1,2,4,6,8)]
rownames(diag) <- 1:nrow(diag); diag
```

```
##            item     tar    c.tar     mtar   c.mtar
## 1           lag   1.000    1.000    1.000    1.000
## 2        thresh   0.000   -0.011    0.000    0.014
## 3     total obs  60.000   60.000   60.000   60.000
```

```
## 4    coint obs   58.000   58.000   58.000   58.000
## 5          aic -296.213 -296.857 -296.194 -301.479
## 6          bic -287.971 -288.615 -287.952 -293.237
## 7   LB test(4)    0.770    0.764    0.815    0.976
## 8   LB test(8)    0.584    0.615    0.617    0.702
## 9  LB test(12)    0.568    0.562    0.626    0.856
## 10   H1: no CI    4.314    4.669    4.304    7.338
## 11  H2: no APT    0.246    0.863    0.228    5.496
## 12 H2: p.value    0.622    0.357    0.635    0.023
```

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

```
##     model reg       variable estimate st.error t.value p.value sign
## 1     tar  LR    (Intercept)   -0.006    0.052  -0.120   0.905
## 2     tar  LR       PDISTGSP    1.139    0.024  48.336   0.000  ***
## 3     tar  CI  pos.resid.t_1   -0.217    0.100  -2.170   0.034   **
## 4     tar  CI  neg.resid.t_1   -0.302    0.145  -2.084   0.042   **
## 5     tar  CI diff.resid.t_1    0.134    0.133   1.008   0.318
## 6   c.tar  LR    (Intercept)   -0.006    0.052  -0.120   0.905
## 7   c.tar  LR       PDISTGSP    1.139    0.024  48.336   0.000  ***
## 8   c.tar  CI  pos.resid.t_1   -0.196    0.098  -1.990   0.052    *
## 9   c.tar  CI  neg.resid.t_1   -0.357    0.148  -2.415   0.019   **
## 10  c.tar  CI diff.resid.t_1    0.142    0.133   1.070   0.289
## 11   mtar  LR    (Intercept)   -0.006    0.052  -0.120   0.905
## 12   mtar  LR       PDISTGSP    1.139    0.024  48.336   0.000  ***
## 13   mtar  CI  pos.resid.t_1   -0.212    0.107  -1.976   0.053    *
## 14   mtar  CI  neg.resid.t_1   -0.290    0.129  -2.249   0.029   **
## 15   mtar  CI diff.resid.t_1    0.124    0.132   0.936   0.353
## 16 c.mtar  LR    (Intercept)   -0.006    0.052  -0.120   0.905
## 17 c.mtar  LR       PDISTGSP    1.139    0.024  48.336   0.000  ***
## 18 c.mtar  CI  pos.resid.t_1    0.124    0.176   0.703   0.485
## 19 c.mtar  CI  neg.resid.t_1   -0.327    0.088  -3.720   0.000  ***
## 20 c.mtar  CI diff.resid.t_1    0.048    0.131   0.366   0.716
```

```
ee <- list(e1, e2, e3, e4); vect <- NULL
for (i in 1:4) {
ef <- data.frame(ee[i])
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) <- colnames(diag)
( res.CI <- rbind(diag, ve)[c(1:2, 13:16, 3:12), ] )
```

```
##           item      tar    c.tar     mtar    c.mtar
## 1          lag        1        1        1         1
## 2       thresh        0   -0.011        0     0.014
## 13   pos.coeff -0.217**  -0.196*  -0.212*     0.124
## 14 pos.t.value  (-2.17)  (-1.99) (-1.976)   (0.703)
```

```
## 15   neg.coeff -0.302** -0.357**  -0.29** -0.327***
## 16 neg.t.value (-2.084) (-2.415) (-2.249)   (-3.72)
## 3    total obs       60       60       60        60
## 4    coint obs       58       58       58        58
## 5          aic -296.213 -296.857 -296.194  -301.479
## 6          bic -287.971 -288.615 -287.952  -293.237
## 7   LB test(4)     0.77    0.764    0.815     0.976
## 8   LB test(8)    0.584    0.615    0.617     0.702
## 9  LB test(12)    0.568    0.562    0.626     0.856
## 10   H1: no CI    4.314    4.669    4.304     7.338
## 11  H2: no APT    0.246    0.863    0.228     5.496
## 12 H2: p.value    0.622    0.357    0.635     0.023
```

```r
rownames(res.CI) <- 1:nrow(res.CI)


(sem <- ecmSymFit(y=PREVGSP, x=PDISTGSP, lag=1)); names(sem)
```

```
##
## ===============================================================
## ECM - Symmetric + linear cointegration -  "PDISTGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.093766 -0.019789 -0.004016  0.015060  0.156128
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0002043  0.0053874   0.038    0.970
## X.diff.PDISTGSP.t_1  0.4313869  0.3426264   1.259    0.213
## X.diff.PREVGSP.t_1  -0.0356217  0.3166346  -0.113    0.911
## X.ECT.t_1            0.0849969  0.1923334   0.442    0.660
##
## Residual standard error: 0.04096 on 54 degrees of freedom
## Multiple R-squared:  0.1593, Adjusted R-squared:  0.1126
## F-statistic:  3.41 on 3 and 54 DF,  p-value: 0.02383
##
##
## ===============================================================
## ECM - Symmetric + linear cointegration -  "PREVGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124643 -0.026403 -0.002823  0.019531  0.135503
##
## Coefficients:
```

```
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.0002564  0.0058701  -0.044    0.965
## X.diff.PDISTGSP.t_1   0.3325658  0.3733236   0.891    0.377
## X.diff.PREVGSP.t_1   -0.0010842  0.3450032  -0.003    0.998
## X.ECT.t_1            -0.1660583  0.2095653  -0.792    0.432
##
## Residual standard error: 0.04463 on 54 degrees of freedom
## Multiple R-squared:  0.1265, Adjusted R-squared:  0.078
## F-statistic: 2.607 on 3 and 54 DF,  p-value: 0.06096
```

```
## [1] "y"      "x"      "lag"    "data"   "IndVar" "name.y" "name.x" "ecm.y"
## [9] "ecm.x"
```

```
aem <- ecmAsyFit(y=PREVGSP, x=PDISTGSP,lag=1, model="mtar", split=TRUE, thresh=t.mtar)
aem
```

```
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PDISTGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.090802 -0.024957 -0.002867  0.015817  0.157527
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.0016105  0.0086790  -0.186    0.854
## X.diff.PDISTGSP.t_1.pos   0.2480386  0.5789470   0.428    0.670
## X.diff.PDISTGSP.t_1.neg   0.5153243  0.5571437   0.925    0.359
## X.diff.PREVGSP.t_1.pos    0.1827118  0.5412093   0.338    0.737
## X.diff.PREVGSP.t_1.neg   -0.1821365  0.5001431  -0.364    0.717
## X.ECT.t_1.pos             0.0009272  0.4659797   0.002    0.998
## X.ECT.t_1.neg             0.1037960  0.2238550   0.464    0.645
##
## Residual standard error: 0.042 on 51 degrees of freedom
## Multiple R-squared:  0.1652, Adjusted R-squared:  0.06695
## F-statistic: 1.682 on 6 and 51 DF,  p-value: 0.1446
##
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PREVGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124755 -0.022629 -0.003834  0.015744  0.137286
```

```
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.0012261  0.0094631  -0.130    0.897
## X.diff.PDISTGSP.t_1.pos  0.4523786  0.6312545   0.717    0.477
## X.diff.PDISTGSP.t_1.neg  0.3255092  0.6074813   0.536    0.594
## X.diff.PREVGSP.t_1.pos  -0.0950421  0.5901072  -0.161    0.873
## X.diff.PREVGSP.t_1.neg   0.0002818  0.5453307   0.001    1.000
## X.ECT.t_1.pos            0.0323015  0.5080807   0.064    0.950
## X.ECT.t_1.neg           -0.2076581  0.2440802  -0.851    0.399
##
## Residual standard error: 0.04579 on 51 degrees of freedom
## Multiple R-squared:  0.1314, Adjusted R-squared:  0.02925
## F-statistic: 1.286 on 6 and 51 DF,  p-value: 0.2802
```

```
(ccc <- summary(aem))
```

```
##               DepVar                  IndVar   estimate error t.value
## 1  diff.PDISTGSP.t_0  |            (Intercept)   -0.002 0.009  -0.186
## 2                     | X.diff.PDISTGSP.t_1.pos   0.248 0.579   0.428
## 3                     | X.diff.PDISTGSP.t_1.neg   0.515 0.557   0.925
## 4                     |  X.diff.PREVGSP.t_1.pos   0.183 0.541   0.338
## 5                     |  X.diff.PREVGSP.t_1.neg  -0.182 0.500  -0.364
## 6                     |           X.ECT.t_1.pos   0.001 0.466   0.002
## 7                     |           X.ECT.t_1.neg   0.104 0.224   0.464
## 8   diff.PREVGSP.t_0  -            (Intercept)   -0.001 0.009  -0.130
## 9                     - X.diff.PDISTGSP.t_1.pos   0.452 0.631   0.717
## 10                    - X.diff.PDISTGSP.t_1.neg   0.326 0.607   0.536
## 11                    -  X.diff.PREVGSP.t_1.pos  -0.095 0.590  -0.161
## 12                    -  X.diff.PREVGSP.t_1.neg   0.000 0.545   0.001
## 13                    -           X.ECT.t_1.pos   0.032 0.508   0.064
## 14                    -           X.ECT.t_1.neg  -0.208 0.244  -0.851
##    p.value signif
## 1    0.854
## 2    0.670
## 3    0.359
## 4    0.737
## 5    0.717
## 6    0.998
## 7    0.645
## 8    0.897
## 9    0.477
## 10   0.594
## 11   0.873
## 12   1.000
## 13   0.950
## 14   0.399
```

```
(edia <- ecmDiag(aem, 3))
```

```
##          item PDISTGSP  PREVGSP
## 1   R-squared    0.165    0.131
## 2      Adj-R2    0.067    0.029
```

```
## 3      F-stat    1.682    1.286
## 4     Stat DW    1.862    1.824
## 5  p-value DW    0.420    0.362
## 6         AIC -194.595 -184.561
## 7         BIC -178.112 -168.078
## 8       LB(4)    0.490    0.392
## 9       LB(8)    0.283    0.319
## 10     LB(12)    0.272    0.493
```

```
(tes <- ecmAsyTest(aem)$out)
```

```
##          Hypothesis description|
## 1 H1: Equ adjust path asymmetry|
## 2    H2: Granger causality test|
## 3    H2: Granger causality test|
## 4 H3: Distributed lag asymmetry|
## 5 H3: Distributed lag asymmetry|
## 6      H4: Cumulative asymmetry|
## 7      H4: Cumulative asymmetry|
##                                                      Expression
## 1                                  X.ECT.t_1.pos=X.ECT.t_1.neg
## 2                       PDISTGSP (x) does not Granger cause...
## 3                        PREVGSP (y) does not Granger cause...
## 4            X.diff.PDISTGSP.t_1.pos = X.diff.PDISTGSP.t_1.neg
## 5              X.diff.PREVGSP.t_1.pos = X.diff.PREVGSP.t_1.neg
## 6 Cumulative positive PDISTGSP = Cumulative negative PDISTGSP
## 7   Cumulative positive PREVGSP = Cumulative negative PREVGSP
##   PDISTGSP.F.Stat PREVGSP.F.Stat PDISTGSP.P.Value PREVGSP.P.Value
## 1           0.038          0.175            0.846           0.677
## 2           0.603          0.479            0.551           0.622
## 3           0.105          0.013            0.901           0.987
## 4           0.095          0.018            0.760           0.894
## 5           0.208          0.012            0.650           0.913
## 6           0.095          0.018            0.760           0.894
## 7           0.208          0.012            0.650           0.913
##   PDISTGSP.Sig PREVGSP.Sig
## 1
## 2
## 3
## 4
## 5
## 6
## 7
```

```
###################################2008-2012#######################


load("C:/Users/Lucas/Desktop/UFPR/2º Sem/Macroeoconometria/ArtigoMacroeconometria/precos2.RData")

SP <- precos2[grep("GUARULHOS", precos2$MUNICIPIO), ]

GSP <- SP[grep("GASOLINA COMUM", SP$PRODUTO), ]
View(GSP)
```

```r
PDISTGSP<-ts(GSP$PRECOMEDIODISTRIBUICAO,frequency=12,start=c(2008,1),end=c(2012,12))
head(PDISTGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 2.060536 2.059900 2.059302 2.060493 2.058098
```

```r
PREVGSP<-ts(GSP$PRECOMEDIOREVENDA,frequency=12,start=c(2008,1),end=c(2012,12))
head(PREVGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 2.356441 2.360765 2.348341 2.350146 2.347690
```

```r
INPC <- as.zoo(ts(GSP$INPCFEV2016100, frequency=12,start=c(2003,1),end=c(2007,12)))
head(INPC)
```

```
## 2003(1) 2003(2) 2003(3) 2003(4) 2003(5) 2003(6)
##    91.4    90.7    90.6    90.6    91.4    91.8
```

```r
PREVGSP <- PREVGSP*100
head(PREVGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 235.6441 236.0765 234.8341 235.0146 234.7690
```

```r
PREVGSP <- PREVGSP/INPC
```

```
## Warning: Métodos incompatíveis ("Ops.ts", "Ops.zoo") para "/"
```

```r
head(PREVGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 2.578163 2.602828 2.591988 2.593981 2.568588
```

```r
PDISTGSP <- PDISTGSP*100
head(PDISTGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 206.0536 205.9900 205.9302 206.0493 205.8098
```

```r
PDISTGSP <- PDISTGSP/INPC
```

```
## Warning: Métodos incompatíveis ("Ops.ts", "Ops.zoo") para "/"
```

```r
head(PDISTGSP)
```

```
##           Jan      Feb      Mar      Apr      May
## 2008 2.254416 2.271114 2.272960 2.274275 2.251748
```

```r
#2. Stationarity tests
precoadfGREV=ur.df(PREVGSP,type= "trend", selectlags= "AIC")
precoglsGREV=ur.ers(PREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGREV@lags
```

```
## [1] 1
```

```r
precoadfGREV@teststat[1]
```

```
## [1] -2.858102
```

```r
precoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
precoglsGREV@teststat[1]
```

```
## [1] -2.840048
```

```r
precoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
precoadfGDIST=ur.df(PDISTGSP,type= "trend", selectlags= "AIC")
precoglsGDIST=ur.ers(PDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
precoadfGDIST@lags
```

```
## [1] 1
```

```r
precoadfGDIST@teststat[1]
```

```
## [1] -3.466209
```

```r
precoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
precoglsGDIST@teststat[1]
```

```
## [1] -3.481878
```

```r
precoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
diffPREVGSP <- diff(PREVGSP, lag = 1, differences = 1)
diffPDISTGSP <- diff(PDISTGSP, lag = 1, differences = 1)


diffprecoadfGREV=ur.df(diffPREVGSP,type= "trend", selectlags= "AIC")
diffprecoglsGREV=ur.ers(diffPREVGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGREV@lags
```

```
## [1] 1
```

```r
diffprecoadfGREV@teststat[1]
```

```
## [1] -5.18176
```

```r
diffprecoadfGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
diffprecoglsGREV@teststat[1]
```

```
## [1] -5.179431
```

```r
diffprecoglsGREV@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
diffprecoadfGDIST=ur.df(diffPDISTGSP,type= "trend", selectlags= "AIC")
diffprecoglsGDIST=ur.ers(diffPDISTGSP, type = "DF-GLS", model = "trend",lag.max = 1)
diffprecoadfGDIST@lags
```

```
## [1] 1
```

```r
diffprecoadfGDIST@teststat[1]
```

```
## [1] -6.586498
```

```r
diffprecoadfGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -4.04 -3.45 -3.15
```

```r
diffprecoglsGDIST@teststat[1]
```

```
## [1] -6.630036
```

```r
diffprecoglsGDIST@cval[1,]
```

```
##  1pct  5pct 10pct
## -3.58 -3.03 -2.74
```

```r
# 2. EG cointegration

LR <- lm(PREVGSP ~ PDISTGSP); summary(LR)
```

```
##
## Call:
## lm(formula = PREVGSP ~ PDISTGSP)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096371 -0.017756 -0.003698  0.018978  0.081622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.38777    0.12536  -3.093  0.00304 **
## PDISTGSP     1.31815    0.05395  24.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 58 degrees of freedom
## Multiple R-squared:  0.9114, Adjusted R-squared:  0.9099
## F-statistic: 596.9 on 1 and 58 DF,  p-value: < 2.2e-16
```

```r
(LR.coef <- round(summary(LR)$coefficients, 6))
```

```
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -0.387770   0.125356 -3.093356 0.003043
## PDISTGSP     1.318148   0.053952 24.431934 0.000000
```

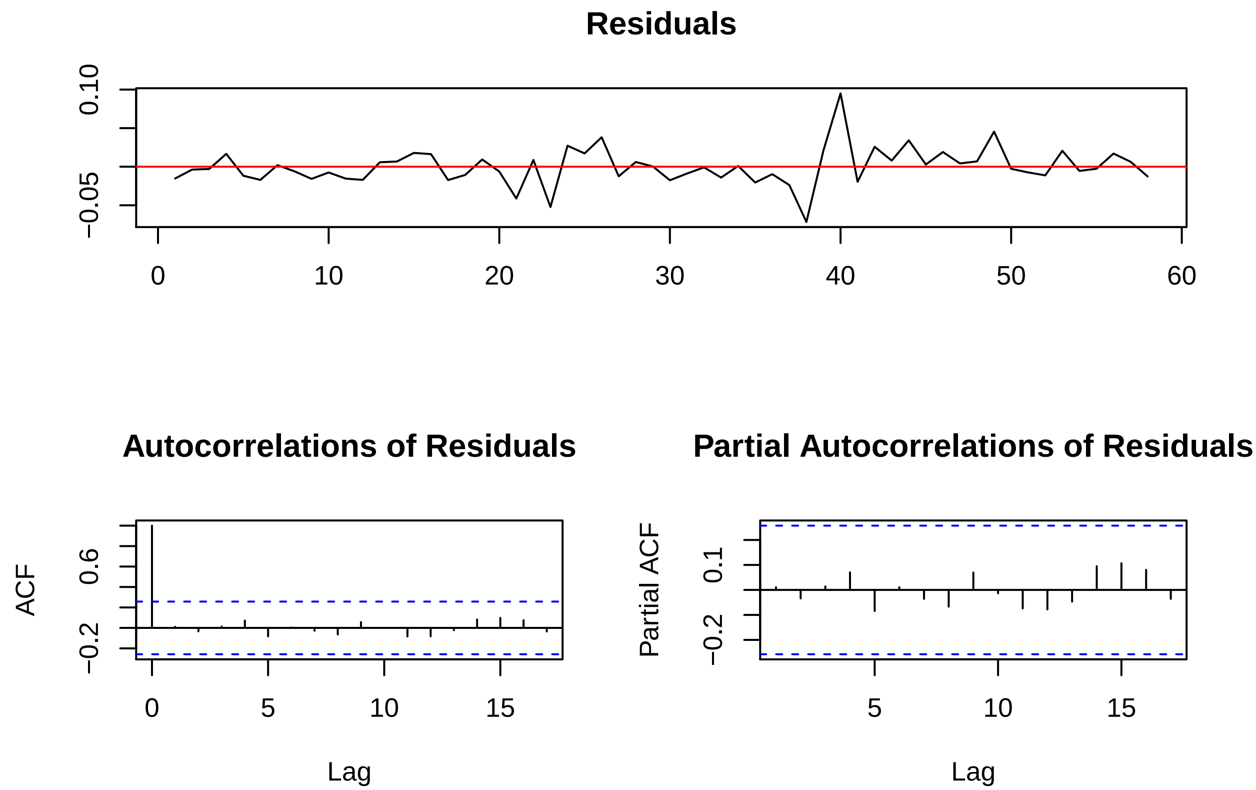
```r
(ry <- ts(residuals(LR), start=start(PDISTGSP), end=end(PDISTGSP), frequency =12))
```

```
##                Jan           Feb           Mar           Apr           May
## 2008 -0.0057215182 -0.0030665857 -0.0163412283 -0.0160813759 -0.0117796262
## 2009 -0.0278147631 -0.0339037486 -0.0145797822  0.0034591435  0.0243001420
## 2010 -0.0619214343 -0.0161011388  0.0197224898  0.0580459931  0.0293281575
## 2011 -0.0248827205 -0.0266979373 -0.0390533514 -0.0963709564 -0.0465791878
## 2012  0.0231153763  0.0170239206  0.0533859685  0.0359422290  0.0081807640
##                Jun           Jul           Aug           Sep           Oct
## 2008  0.0112215398  0.0002379844 -0.0197166244 -0.0139283601 -0.0123010409
## 2009  0.0349813185  0.0047737877 -0.0155801821 -0.0043288650 -0.0058499227
## 2010  0.0152335151  0.0052818155 -0.0171028593 -0.0239469739 -0.0158509952
```

```
## 2011  0.0816218439  0.0575766337  0.0518179005  0.0352654488  0.0496787059
## 2012 -0.0136083513  0.0076794129  0.0040912064 -0.0012605576  0.0149866560
##                Nov           Dec
## 2008 -0.0221956717 -0.0222793776
## 2009 -0.0449114236 -0.0259072992
## 2010 -0.0208685608 -0.0118943851
## 2011  0.0340136063  0.0340274071
## 2012  0.0187291829 -0.0012953445
```

```r
summary(eg <- ur.df(ry, type=c("none"), lags=1)); plot(eg)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.071626 -0.012568 -0.002683  0.009219  0.095080
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -0.4451     0.1094  -4.069  0.00015 ***
## z.diff.lag   0.2488     0.1302   1.911  0.06106 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02408 on 56 degrees of freedom
## Multiple R-squared:  0.2286, Adjusted R-squared:  0.2011
## F-statistic:   8.3 on 2 and 56 DF,  p-value: 0.0006968
##
##
## Value of test-statistic is: -4.0689
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```




```r
(eg4 <- Box.test(eg@res, lag = 4, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 0.42433, df = 4, p-value = 0.9804
```

```r
(eg8 <- Box.test(eg@res, lag = 8, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 1.2226, df = 8, p-value = 0.9964
```

```r
(eg12 <- Box.test(eg@res, lag = 12, type="Ljung"))
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 2.4813, df = 12, p-value = 0.9982
```

```
(eg16 <- Box.test(eg@res, lag = 16, type="Ljung") )
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 4.31, df = 16, p-value = 0.9983
```

```
(eg20 <- Box.test(eg@res, lag = 20, type="Ljung"))
```

```
##
##  Box-Ljung test
##
## data:  eg@res
## X-squared = 10.916, df = 20, p-value = 0.9484
```

```
# 3. TAR + Cointegration
# best threshold

t3<-ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=0)
(th.tar <- t3$basic); plot(t3)
```

```
##           Item    tar
## 1          lag  0.000
## 2 thresh final -0.026
## 3 thresh range  0.150
## 4   sse.lowest  0.033
## 5    Total obs 60.000
## 6       CI obs 59.000
## 7    Lower obs  9.000
## 8    Upper obs 51.000
```

```
ttt<-t3$th.final
for (i in 1:12) { # 20 seconds
t3a <- ciTarThd(PREVGSP, PDISTGSP, model="tar", lag=i)
th.tar[i+2] <- t3a$basic[,2]
}
th.tar
```

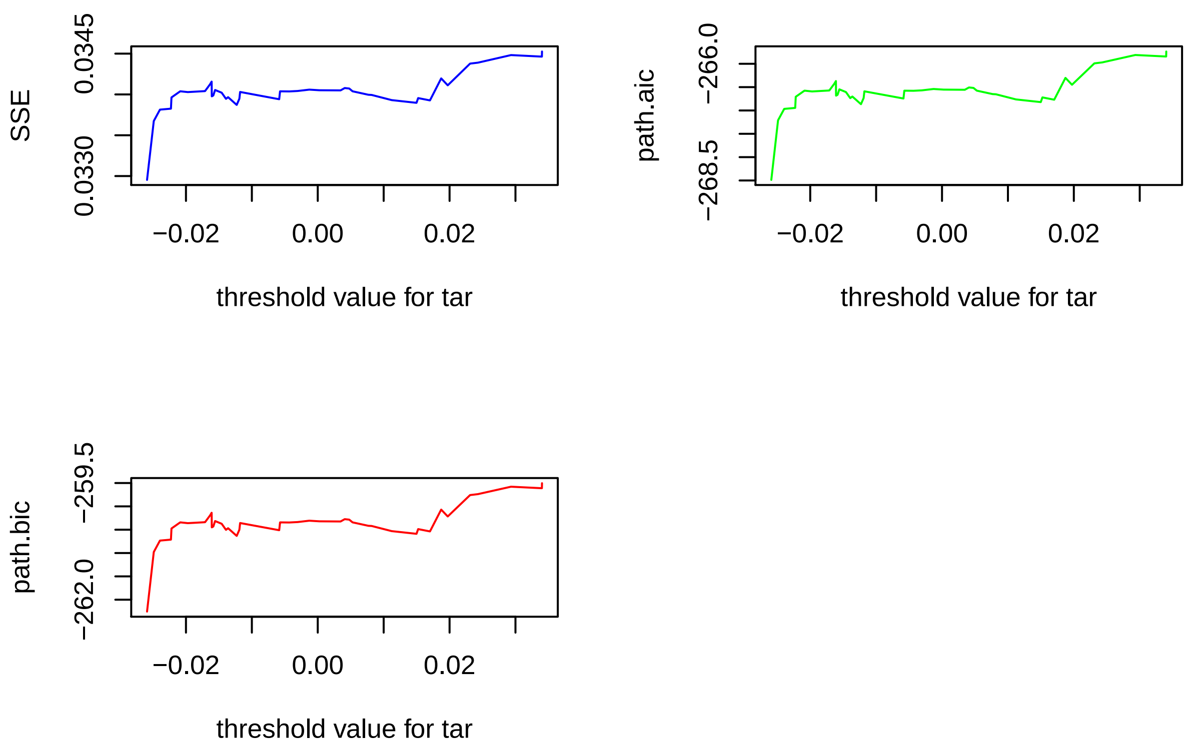
```
##           Item    tar     V3     V4     V5     V6     V7     V8     V9
## 1          lag  0.000  1.000  2.000  3.000  4.000  5.000  6.000  7.000
## 2 thresh final -0.026 -0.026 -0.026 -0.026 -0.026 -0.026 -0.027 -0.027
## 3 thresh range  0.150  0.150  0.150  0.150  0.150  0.150  0.150  0.150
## 4   sse.lowest  0.033  0.031  0.030  0.030  0.030  0.029  0.029  0.028
## 5    Total obs 60.000 60.000 60.000 60.000 60.000 60.000 60.000 60.000
## 6       CI obs 59.000 58.000 57.000 56.000 55.000 54.000 53.000 52.000
## 7    Lower obs  9.000  9.000  9.000  9.000  9.000  9.000  8.000  8.000
## 8    Upper obs 51.000 50.000 49.000 48.000 47.000 46.000 46.000 45.000
##      V10    V11    V12    V13    V14
## 1  8.000  9.000 10.000 11.000 12.000
## 2 -0.027 -0.027 -0.027 -0.027 -0.027
```

```
## 3  0.150  0.150  0.150  0.150  0.150
## 4  0.028  0.028  0.028  0.028  0.027
## 5 60.000 60.000 60.000 60.000 60.000
## 6 51.000 50.000 49.000 48.000 47.000
## 7  8.000  8.000  8.000  8.000  8.000
## 8 44.000 43.000 42.000 41.000 40.000
```

```
t4 <- ciTarThd(PREVGSP, PDISTGSP, model="mtar", lag=0); (th.mtar <- t4$basic)
```

```
##            Item   mtar
## 1           lag  0.000
## 2 thresh final  0.019
## 3 thresh range  0.150
## 4    sse.lowest  0.033
## 5     Total obs 60.000
## 6        CI obs 58.000
## 7     Lower obs  9.000
## 8     Upper obs 50.000
```

```
plot(t4)
```



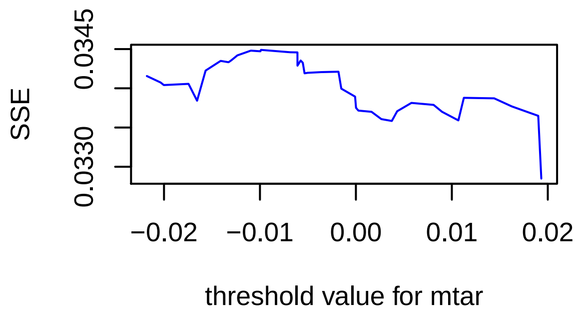

```
mttt<-t4$th.final
for (i in 1:12) {
t4a <- ciTarThd(PREVGSP,PDISTGSP, model="mtar", lag=i)
```

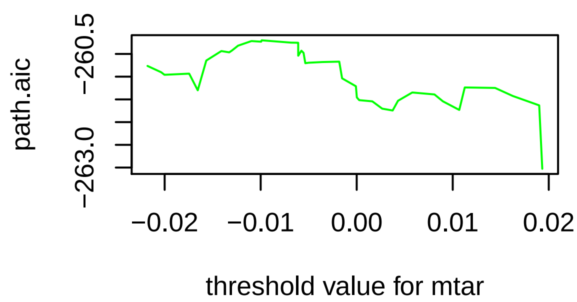
```
th.mtar[i+2] <- t4a$basic[,2]
}
th.mtar
```

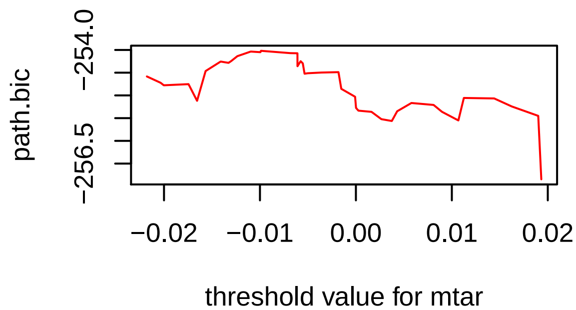
```
##            Item   mtar     V3     V4     V5     V6     V7     V8     V9
## 1           lag  0.000  1.000  2.000  3.000  4.000  5.000  6.000  7.000
## 2 thresh final  0.019  0.019  0.019  0.018  0.016  0.011  0.011  0.011
## 3 thresh range  0.150  0.150  0.150  0.150  0.150  0.150  0.150  0.150
## 4    sse.lowest  0.033  0.029  0.030  0.030  0.030  0.030  0.029  0.029
## 5     Total obs 60.000 60.000 60.000 60.000 60.000 60.000 60.000 60.000
## 6        CI obs 58.000 58.000 57.000 56.000 55.000 54.000 53.000 52.000
## 7     Lower obs  9.000  9.000  9.000  9.000  9.000  9.000  8.000  8.000
## 8     Upper obs 50.000 50.000 49.000 48.000 47.000 46.000 46.000 45.000
##      V10    V11    V12    V13    V14
## 1  8.000  9.000 10.000 11.000 12.000
## 2  0.011  0.009  0.000  0.000  0.000
## 3  0.150  0.150  0.150  0.150  0.150
## 4  0.029  0.029  0.028  0.028  0.028
## 5 60.000 60.000 60.000 60.000 60.000
## 6 51.000 50.000 49.000 48.000 47.000
## 7  8.000  8.000  8.000  8.000  8.000
## 8 44.000 43.000 42.000 41.000 40.000
```

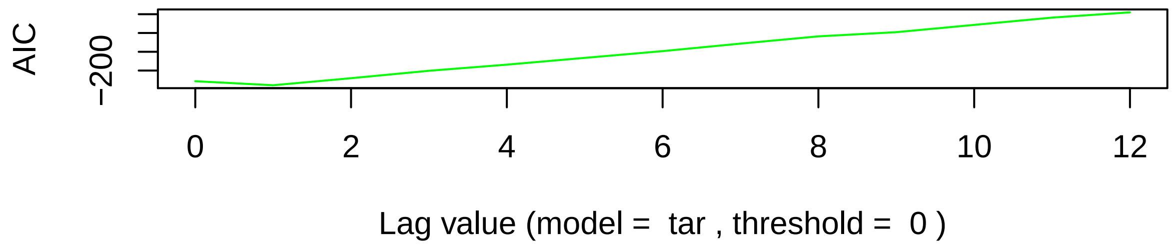
```
t.tar <- ttt; t.mtar <- mttt
#t.tar <- -8.041; t.mtar <- -0.451 # lag = 0 to 4
#t.tar <- -8.701 ; t.mtar <- -0.451 # lag = 5 to 12

mx <- 12
(g1 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh= 0)); plot(g1)
```

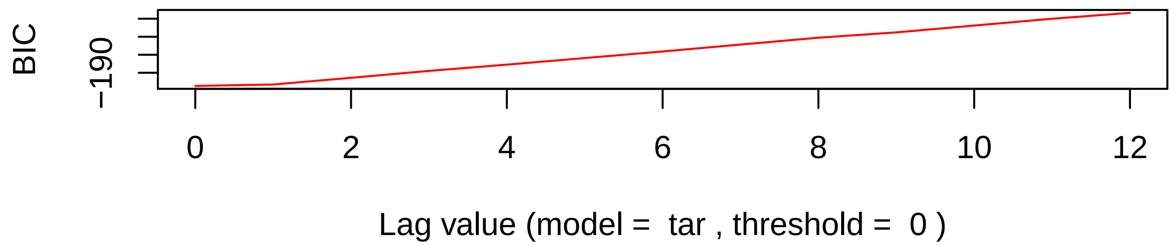
```
##             Item    Value
## 1          model      tar
## 2        max lag       12
## 3      threshold        0
## 4 BestLag.byAic        1
## 5 BestLag.byBic        0
## 6       Best AIC -203.912
## 7       Best BIC -197.294
```


SSE
0.0330
0.0345
−0.02
−0.01
0.00
0.01
0.02
threshold value for mtar


path.aic
−263.0
−260.5
−0.02
−0.01
0.00
0.01
0.02
threshold value for mtar

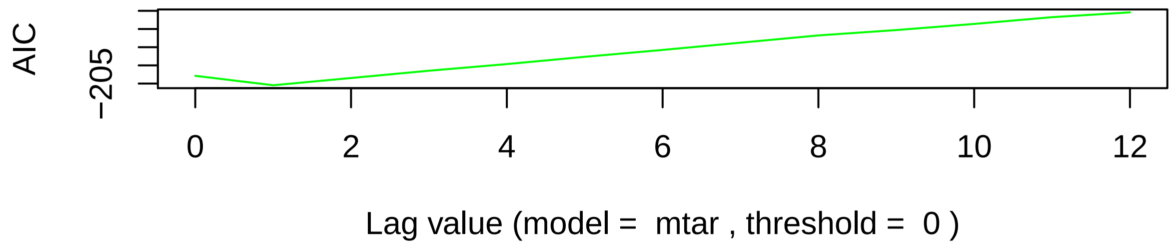

path.bic
−256.5
−254.0
−0.02
−0.01
0.00
0.01
0.02
threshold value for mtar

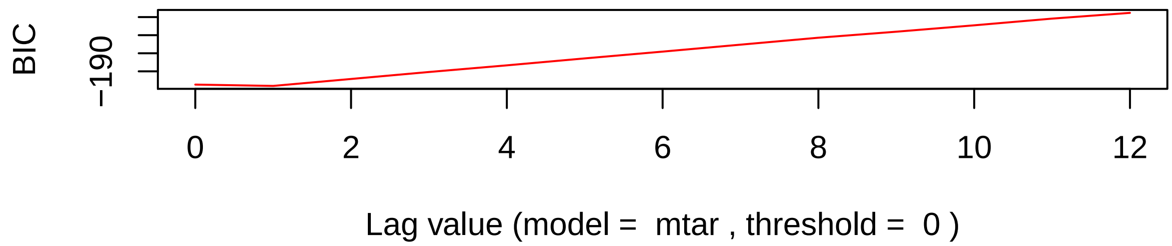





```
(g2 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh= 0)); plot(g2)
```

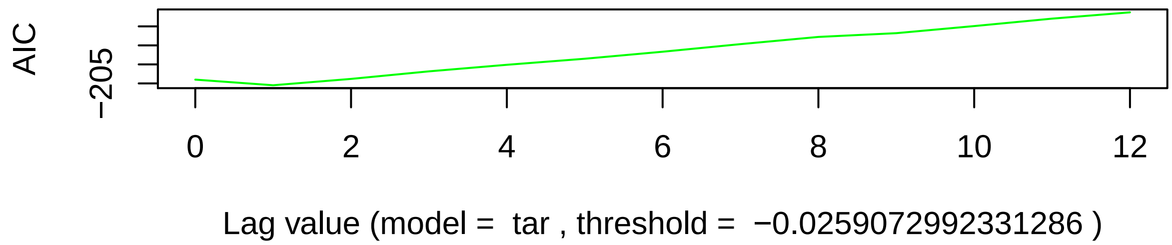
```
##             Item    Value
## 1          model     mtar
## 2        max lag       12
## 3      threshold        0
## 4 BestLag.byAic        1
## 5 BestLag.byBic        1
## 6       Best AIC -205.457
## 7       Best BIC -198.056
```

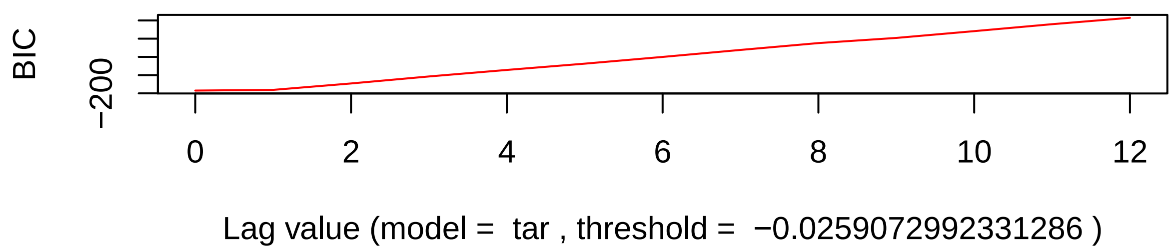





```
(g3 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="tar", maxlag=mx, thresh=t.tar)); plot(g3)
```

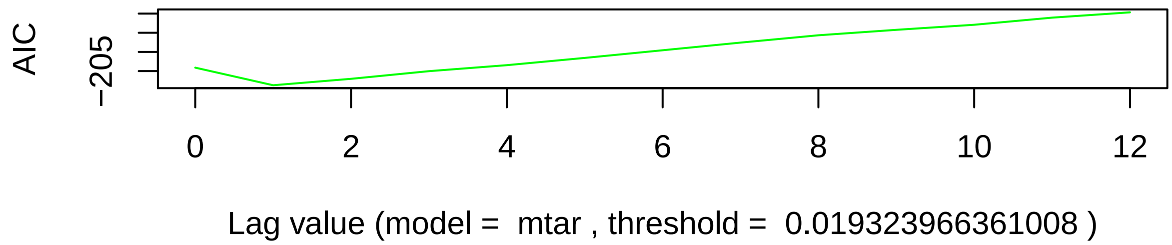
```
##              Item                Value
## 1           model                  tar
## 2         max lag                   12
## 3       threshold -0.0259072992331286
## 4 BestLag.byAic                     1
## 5 BestLag.byBic                     0
## 6        Best AIC             -205.483
## 7        Best BIC             -198.458
```

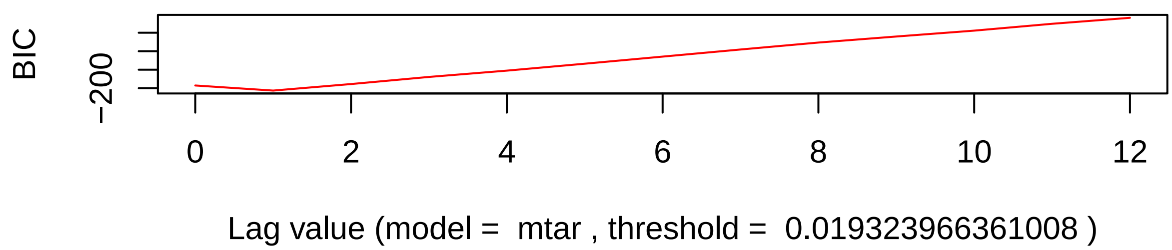





```
(g4 <-ciTarLag(y=PREVGSP, x=PDISTGSP, model="mtar",maxlag=mx, thresh=t.mtar)); plot(g4)
```

```
##            Item             Value
## 1         model              mtar
## 2       max lag                12
## 3     threshold 0.019323966361008
## 4 BestLag.byAic                 1
## 5 BestLag.byBic                 1
## 6      Best AIC          -208.695
## 7      Best BIC          -201.294
```






```
vv <- 1
(f1 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096371 -0.017756 -0.003698  0.018978  0.081622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.38777    0.12536  -3.093  0.00304 **
## PDISTGSP     1.31815    0.05395  24.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 58 degrees of freedom
## Multiple R-squared:  0.9114, Adjusted R-squared:  0.9099
## F-statistic: 596.9 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
```

```
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.075093 -0.015087 -0.003654  0.007323  0.091370
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   -0.3635     0.1462  -2.486 0.015986 *
## neg.resid.t_1   -0.5318     0.1503  -3.539 0.000825 ***
## diff.resid.t_1   0.2422     0.1307   1.853 0.069272 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02414 on 55 degrees of freedom
## Multiple R-squared:  0.2385, Adjusted R-squared:  0.197
## F-statistic: 5.742 on 3 and 55 DF,  p-value: 0.001718
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     57 0.042064
## 2     55 0.032051  2  0.010013 8.5917 0.0005662 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1     56 0.032466
## 2     55 0.032051  1 0.00041513 0.7124 0.4023
```

```r
(f2 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="tar", lag=vv, thresh=t.tar ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.096371 -0.017756 -0.003698  0.018978  0.081622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.38777    0.12536  -3.093  0.00304 **
## PDISTGSP     1.31815    0.05395  24.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 58 degrees of freedom
## Multiple R-squared:  0.9114, Adjusted R-squared:  0.9099
## F-statistic: 596.9 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.080427 -0.013082 -0.002528  0.007523  0.084289
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   -0.3180     0.1297  -2.452 0.017416 *
## neg.resid.t_1   -0.6719     0.1684  -3.990 0.000197 ***
## diff.resid.t_1   0.2534     0.1279   1.982 0.052542 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02365 on 55 degrees of freedom
## Multiple R-squared:  0.2693, Adjusted R-squared:  0.2294
## F-statistic: 6.756 on 3 and 55 DF,  p-value: 0.000583
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     57 0.042064
## 2     55 0.030756  2  0.011309 10.111 0.0001821 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
```

```
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     56 0.032466
## 2     55 0.030756  1 0.0017102 3.0583 0.08591 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(f3 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=0 ))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096371 -0.017756 -0.003698  0.018978  0.081622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.38777    0.12536  -3.093  0.00304 **
## PDISTGSP     1.31815    0.05395  24.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 58 degrees of freedom
## Multiple R-squared:  0.9114, Adjusted R-squared:  0.9099
## F-statistic: 596.9 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06623 -0.01027 -0.00094  0.01299  0.08223
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   -0.6679     0.1699  -3.932 0.000238 ***
## neg.resid.t_1   -0.3227     0.1296  -2.490 0.015837 *
## diff.resid.t_1   0.2985     0.1313   2.273 0.026981 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02368 on 55 degrees of freedom
## Multiple R-squared:  0.267,  Adjusted R-squared:  0.227
## F-statistic: 6.676 on 3 and 55 DF,  p-value: 0.0006338
##
```

```
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     57 0.042064
## 2     55 0.030853  2  0.011211 9.9923 0.0001987 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     56 0.032466
## 2     55 0.030853  1 0.0016124 2.8743 0.09566 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(f4 <- ciTarFit(y=PREVGSP, x=PDISTGSP, model="mtar", lag=vv, thresh=t.mtar))
```

```
## === Long Run Regression
##
## Call:
## lm(formula = formula.LR, data = data.LR)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096371 -0.017756 -0.003698  0.018978  0.081622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.38777    0.12536  -3.093  0.00304 **
## PDISTGSP     1.31815    0.05395  24.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 58 degrees of freedom
## Multiple R-squared:  0.9114, Adjusted R-squared:  0.9099
## F-statistic: 596.9 on 1 and 58 DF,  p-value: < 2.2e-16
##
## === Threshold Cointegration Regression
##
```

```
## Call:
## lm(formula = diff.resid.t_0 ~ 0 + ., data = data.CI)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.065222 -0.009659 -0.000041  0.014899  0.067037
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pos.resid.t_1   -0.9168     0.2041  -4.492 3.67e-05 ***
## neg.resid.t_1   -0.3197     0.1138  -2.809  0.00688 **
## diff.resid.t_1   0.3707     0.1316   2.817  0.00671 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02285 on 55 degrees of freedom
## Multiple R-squared:  0.318,  Adjusted R-squared:  0.2808
## F-statistic: 8.547 on 3 and 55 DF,  p-value: 9.427e-05
##
## === H1: No cointegration b/w two variables
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 = 0
## neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     57 0.042064
## 2     55 0.028706  2  0.013358 12.796 2.734e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## === H2: Symmetric adjustment in the long run
## Linear hypothesis test
##
## Hypothesis:
## pos.resid.t_1 - neg.resid.t_1 = 0
##
## Model 1: restricted model
## Model 2: diff.resid.t_0 ~ 0 + (pos.resid.t_1 + neg.resid.t_1 + diff.resid.t_1)
##
##   Res.Df      RSS Df Sum of Sq     F   Pr(>F)
## 1     56 0.032466
## 2     55 0.028706  1 0.0037595 7.203 0.009597 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
r0 <- cbind(summary(f1)$dia, summary(f2)$dia, summary(f3)$dia,
summary(f4)$dia)
diag <- r0[c(1:4, 6:7, 12:14, 8, 9, 11), c(1,2,4,6,8)]
rownames(diag) <- 1:nrow(diag); diag
```

```
##           item      tar    c.tar     mtar   c.mtar
## 1          lag    1.000    1.000    1.000    1.000
## 2       thresh    0.000   -0.026    0.000    0.019
## 3    total obs   60.000   60.000   60.000   60.000
## 4    coint obs   58.000   58.000   58.000   58.000
## 5          aic -262.454 -264.846 -264.662 -268.846
## 6          bic -254.212 -256.604 -256.420 -260.604
## 7   LB test(4)    0.961    0.792    0.999    0.999
## 8   LB test(8)    0.992    0.952    1.000    1.000
## 9  LB test(12)    0.997    0.983    0.999    0.997
## 10   H1: no CI    8.592   10.111    9.992   12.796
## 11  H2: no APT    0.712    3.058    2.874    7.203
## 12 H2: p.value    0.402    0.086    0.096    0.010
```

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

```
##     model reg       variable estimate st.error t.value p.value sign
## 1     tar  LR    (Intercept)   -0.388    0.125  -3.093   0.003  ***
## 2     tar  LR       PDISTGSP    1.318    0.054  24.432   0.000  ***
## 3     tar  CI  pos.resid.t_1   -0.363    0.146  -2.486   0.016   **
## 4     tar  CI  neg.resid.t_1   -0.532    0.150  -3.539   0.001  ***
## 5     tar  CI diff.resid.t_1    0.242    0.131   1.853   0.069    *
## 6   c.tar  LR    (Intercept)   -0.388    0.125  -3.093   0.003  ***
## 7   c.tar  LR       PDISTGSP    1.318    0.054  24.432   0.000  ***
## 8   c.tar  CI  pos.resid.t_1   -0.318    0.130  -2.452   0.017   **
## 9   c.tar  CI  neg.resid.t_1   -0.672    0.168  -3.990   0.000  ***
## 10  c.tar  CI diff.resid.t_1    0.253    0.128   1.982   0.053    *
## 11   mtar  LR    (Intercept)   -0.388    0.125  -3.093   0.003  ***
## 12   mtar  LR       PDISTGSP    1.318    0.054  24.432   0.000  ***
## 13   mtar  CI  pos.resid.t_1   -0.668    0.170  -3.932   0.000  ***
## 14   mtar  CI  neg.resid.t_1   -0.323    0.130  -2.490   0.016   **
## 15   mtar  CI diff.resid.t_1    0.299    0.131   2.273   0.027   **
## 16 c.mtar  LR    (Intercept)   -0.388    0.125  -3.093   0.003  ***
## 17 c.mtar  LR       PDISTGSP    1.318    0.054  24.432   0.000  ***
## 18 c.mtar  CI  pos.resid.t_1   -0.917    0.204  -4.492   0.000  ***
## 19 c.mtar  CI  neg.resid.t_1   -0.320    0.114  -2.809   0.007  ***
## 20 c.mtar  CI diff.resid.t_1    0.371    0.132   2.817   0.007  ***
```

```r
ee <- list(e1, e2, e3, e4); vect <- NULL
for (i in 1:4) {
ef <- data.frame(ee[i])
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) <- colnames(diag)
( res.CI <- rbind(diag, ve)[c(1:2, 13:16, 3:12), ] )
```

```
##           item       tar     c.tar      mtar    c.mtar
```

```
## 1          lag        1        1        1        1
## 2       thresh        0   -0.026        0    0.019
## 13   pos.coeff  -0.363**  -0.318** -0.668*** -0.917***
## 14 pos.t.value  (-2.486)  (-2.452)  (-3.932)  (-4.492)
## 15   neg.coeff -0.532*** -0.672***  -0.323**  -0.32***
## 16 neg.t.value  (-3.539)   (-3.99)   (-2.49)  (-2.809)
## 3    total obs       60       60       60       60
## 4    coint obs       58       58       58       58
## 5          aic  -262.454 -264.846 -264.662 -268.846
## 6          bic  -254.212 -256.604  -256.42 -260.604
## 7   LB test(4)    0.961    0.792    0.999    0.999
## 8   LB test(8)    0.992    0.952        1        1
## 9  LB test(12)    0.997    0.983    0.999    0.997
## 10   H1: no CI    8.592   10.111    9.992   12.796
## 11  H2: no APT    0.712    3.058    2.874    7.203
## 12 H2: p.value    0.402    0.086    0.096     0.01
```

```
rownames(res.CI) <- 1:nrow(res.CI)

(sem <- ecmSymFit(y=PREVGSP, x=PDISTGSP, lag=1)); names(sem)
```

```
##
## ================================================================
## ECM - Symmetric + linear cointegration -  "PDISTGSP"
## ================================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15245 -0.01593 -0.00433  0.01351  0.18830
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.001973   0.005419   0.364    0.717
## X.diff.PDISTGSP.t_1   0.421063   0.293328   1.435    0.157
## X.diff.PREVGSP.t_1   -0.345405   0.306788  -1.126    0.265
## X.ECT.t_1             0.012131   0.238485   0.051    0.960
##
## Residual standard error: 0.04102 on 54 degrees of freedom
## Multiple R-squared:  0.04067,    Adjusted R-squared:  -0.01262
## F-statistic: 0.7631 on 3 and 54 DF,  p-value: 0.5197
##
##
## ================================================================
## ECM - Symmetric + linear cointegration -  "PREVGSP"
## ================================================================
##
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
```

```
## -0.107044 -0.015103 -0.002802  0.012214  0.174780
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.002358   0.004722   0.499   0.6195
## X.diff.PDISTGSP.t_1  0.218023   0.255587   0.853   0.3974
## X.diff.PREVGSP.t_1  -0.126909   0.267316  -0.475   0.6369
## X.ECT.t_1           -0.373259   0.207801  -1.796   0.0781 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03574 on 54 degrees of freedom
## Multiple R-squared:  0.1656, Adjusted R-squared:  0.1193
## F-statistic: 3.573 on 3 and 54 DF,  p-value: 0.01973

## [1] "y"      "x"      "lag"    "data"   "IndVar" "name.y" "name.x" "ecm.y"
## [9] "ecm.x"
```

```
aem <- ecmAsyFit(y=PREVGSP, x=PDISTGSP,lag=1, model="mtar", split=TRUE, thresh=t.mtar)
aem
```

```
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PDISTGSP"
## ===============================================================
##
## Call:
## lm(formula = DepVar.x ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.092689 -0.016678 -0.000073  0.013800  0.180859
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.001280   0.007384   0.173   0.8631
## X.diff.PDISTGSP.t_1.pos  0.822362   0.479504   1.715   0.0924 .
## X.diff.PDISTGSP.t_1.neg  0.560649   0.502767   1.115   0.2700
## X.diff.PREVGSP.t_1.pos  -0.726338   0.430656  -1.687   0.0978 .
## X.diff.PREVGSP.t_1.neg  -0.260840   0.683206  -0.382   0.7042
## X.ECT.t_1.pos             0.871632   0.457008   1.907   0.0621 .
## X.ECT.t_1.neg            -0.215728   0.259506  -0.831   0.4097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03912 on 51 degrees of freedom
## Multiple R-squared:  0.1761, Adjusted R-squared:  0.07921
## F-statistic: 1.817 on 6 and 51 DF,  p-value: 0.1142
##
##
## ===============================================================
## ECM - Asymmetric + nonlinear threshold cointegration -  "PREVGSP"
## ===============================================================
```

```
##
## Call:
## lm(formula = DepVar.y ~ 1 + X.)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.063738 -0.013697 -0.000772  0.007734  0.168606
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.0002024  0.0065710   0.031   0.9755
## X.diff.PDISTGSP.t_1.pos  0.6081810  0.4266849   1.425   0.1601
## X.diff.PDISTGSP.t_1.neg  0.2651529  0.4473856   0.593   0.5560
## X.diff.PREVGSP.t_1.pos  -0.4417181  0.3832182  -1.153   0.2544
## X.diff.PREVGSP.t_1.neg  -0.1310444  0.6079485  -0.216   0.8302
## X.ECT.t_1.pos            0.1733381  0.4066673   0.426   0.6717
## X.ECT.t_1.neg           -0.5214260  0.2309204  -2.258   0.0283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03481 on 51 degrees of freedom
## Multiple R-squared:  0.2527, Adjusted R-squared:  0.1648
## F-statistic: 2.874 on 6 and 51 DF,  p-value: 0.01726
```

```
(ccc <- summary(aem))
```

```
##               DepVar                 IndVar  estimate error t.value
## 1  diff.PDISTGSP.t_0  |             (Intercept)    0.001 0.007   0.173
## 2                     | X.diff.PDISTGSP.t_1.pos    0.822 0.480   1.715
## 3                     | X.diff.PDISTGSP.t_1.neg    0.561 0.503   1.115
## 4                     |  X.diff.PREVGSP.t_1.pos   -0.726 0.431  -1.687
## 5                     |  X.diff.PREVGSP.t_1.neg   -0.261 0.683  -0.382
## 6                     |           X.ECT.t_1.pos    0.872 0.457   1.907
## 7                     |           X.ECT.t_1.neg   -0.216 0.260  -0.831
## 8   diff.PREVGSP.t_0  -             (Intercept)    0.000 0.007   0.031
## 9                     - X.diff.PDISTGSP.t_1.pos    0.608 0.427   1.425
## 10                    - X.diff.PDISTGSP.t_1.neg    0.265 0.447   0.593
## 11                    -  X.diff.PREVGSP.t_1.pos   -0.442 0.383  -1.153
## 12                    -  X.diff.PREVGSP.t_1.neg   -0.131 0.608  -0.216
## 13                    -           X.ECT.t_1.pos    0.173 0.407   0.426
## 14                    -           X.ECT.t_1.neg   -0.521 0.231  -2.258
##    p.value signif
## 1    0.863
## 2    0.092      *
## 3    0.270
## 4    0.098      *
## 5    0.704
## 6    0.062      *
## 7    0.410
## 8    0.976
## 9    0.160
## 10   0.556
## 11   0.254
## 12   0.830
```

```
## 13   0.672
## 14   0.028     **
```

```r
(edia <- ecmDiag(aem, 3))
```

```
##          item PDISTGSP  PREVGSP
## 1   R-squared    0.176    0.253
## 2      Adj-R2    0.079    0.165
## 3      F-stat    1.817    2.874
## 4     Stat DW    2.085    1.983
## 5  p-value DW    0.904    0.770
## 6         AIC -202.841 -216.379
## 7         BIC -186.358 -199.896
## 8       LB(4)    0.746    0.786
## 9       LB(8)    0.769    0.746
## 10     LB(12)    0.813    0.682
```

```r
(tes <- ecmAsyTest(aem)$out)
```

```
##          Hypothesis description|
## 1 H1: Equ adjust path asymmetry|
## 2    H2: Granger causality test|
## 3    H2: Granger causality test|
## 4 H3: Distributed lag asymmetry|
## 5 H3: Distributed lag asymmetry|
## 6      H4: Cumulative asymmetry|
## 7      H4: Cumulative asymmetry|
##                                                       Expression
## 1                                  X.ECT.t_1.pos=X.ECT.t_1.neg
## 2                      PDISTGSP (x) does not Granger cause...
## 3                       PREVGSP (y) does not Granger cause...
## 4           X.diff.PDISTGSP.t_1.pos = X.diff.PDISTGSP.t_1.neg
## 5             X.diff.PREVGSP.t_1.pos = X.diff.PREVGSP.t_1.neg
## 6 Cumulative positive PDISTGSP = Cumulative negative PDISTGSP
## 7   Cumulative positive PREVGSP = Cumulative negative PREVGSP
##   PDISTGSP.F.Stat PREVGSP.F.Stat PDISTGSP.P.Value PREVGSP.P.Value
## 1           4.442          2.290            0.040           0.136
## 2           2.687          1.470            0.078           0.240
## 3           1.735          0.786            0.187           0.461
## 4           0.115          0.250            0.736           0.620
## 5           0.275          0.155            0.602           0.696
## 6           0.115          0.250            0.736           0.620
## 7           0.275          0.155            0.602           0.696
##   PDISTGSP.Sig PREVGSP.Sig
## 1           **           .
## 2            *
## 3
## 4
## 5
## 6
## 7
```