Rotinas em GAUSS.
Simulação, Testes e Estimação de Processos Estocásticos

João Nicolau
ISEG/ULlisboa & CEMAPRE
(Janeiro de 2018)

Conteúdo

1 Introdução

GAUSS é uma linguagem de programação similar às linguagens C e Pascal mas concebida especialmente para trabalhar com matrizes. Estão disponíveis inúmeros procedimentos e programas (muitos deles cedidos gratuitamente, via Internet), a maioria na área da estatística e econometria. O programa é comercializado pela Aptech Systems, Inc. (http://www.aptech.com).

Nota Importante: todas as rotinas foram criadas pelo autor; embora testadas, não se oferece garantia de qualquer espécie. Deverá o utilizador testar e verificar se as rotinas funcionam adequadamente. A utilização das rotinas em trabalhos publicados implica a referência a este documento. Está vedado o uso destas rotinas para efeitos comerciais.

2 Análise Espectral [ana_esp]

2.1 estima_espectro

▶ Objectivo

Estimar o espectro através da seguinte expressão

$$\hat{f}_{\mathcal{W}}(\omega_j) = \sum_{k=-h}^{h} \left(\frac{h+1-|k|}{(h+1)^2} \right) \hat{f}(\omega_j)$$

onde $\hat{f}(\omega_j)$ é a estimativa do periodograma.

▶ Formato

{w,fw}=estima_espectro(w,I,h,graf);

▶ Input

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ (n é o número de observações)

I: Periodograma $I(\omega_k)$

h: bandwitdth {1,2,...}

graf: se graf = 1 apresenta-se o gráfico de X e o do periodograma

▶ Output

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ (n é o número de observações)

fw: Espectro Estimado

▶ Library

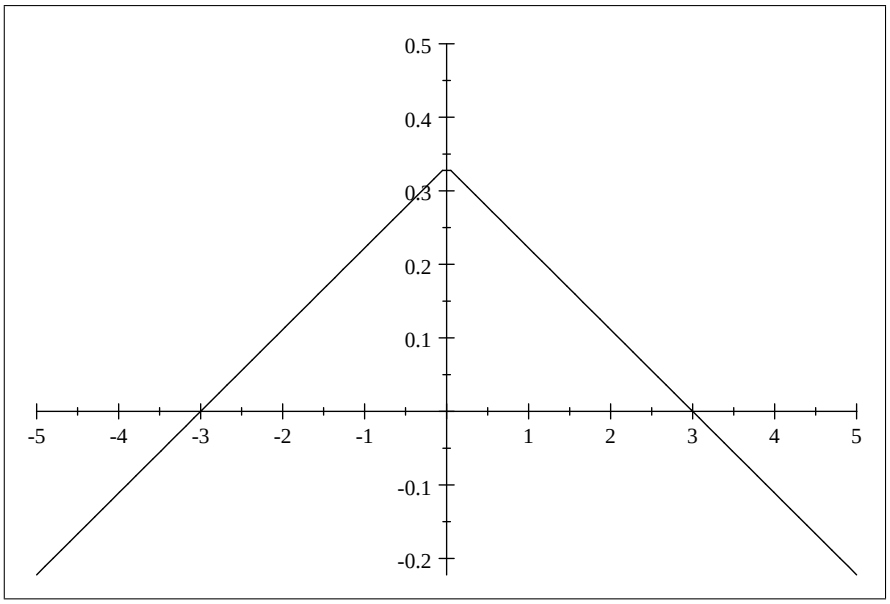
library ana_esp;

▶ Fonte

c:\gauss\srcnic\ana_esp\estimacao.src

▶ Observações

Apresenta-se na figura seguinte a estrutura de ponderações $\left(\frac{h+1-|k|}{(h+1)^2}\right)$ no caso $h = 2$.



Caso $h = 2$

2.2 estima_espectro1

▶ Objectivo

Apresentar graficamente o espectro estimado através da seguinte expressão

$$\hat{f}_{\mathcal{W}}(\omega_j) = \sum_{k=-h}^{h} \left(\frac{h+1-|k|}{(h+1)^2} \right) \hat{f}(\omega_j)$$

onde $\hat{f}(\omega_j)$ é a estimativa do periodograma, usando os valores $h = 0, 2, 4, 6, 8, 10, 12, 14, 16$.

▶ Formato

call estima_espectro1(w,I);

▶ Input

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ (n é o número de observações)

I: Periodograma $I(\omega_k)$

▶ Output

Apresenta nove gráficos numa única folha.

▶ Library

library ana_esp;

▶ Fonte

c:\gauss\srcnic\ana_esp\estimacao.src

2.3 estima_periodograma

▶ **Objectivo**

O nome diz tudo.

▶ **Formato**

{w,I}=estima_periodograma(x,graf);

▶ **Input**

x: vector das observações do processo.

graf: se graf = 1 apresenta-se o gráfico de X e o do periodograma

▶ **Output**

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ (n é o número de observações)

I: Periodograma $I(\omega_k)$ equação 12.1.14 de W.Wei)

▶ **Library**

library ana_esp;

▶ **Fonte**

c:\gauss\srcnic\ana_esp\estimacao.src

2.4 picos_sign

▶ **Objectivo**

Avaliar a significância dos maiores picos do periodograma (usa-se a expressão 12.1.21).

▶ **Formato**

call picos_sign(p,w,I);

▶ **Input**

p: número de picos que se pretende analisar

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$

I: Periodograma $I(\omega_k)$, $k = 0, ..., [n/2]$

▶ **Output**

Apresenta-se uma tabela com as seguintes colunas: Freq, Periodograma, Periodo, pvalue (aprox.)

▶ **Library**

library ana_esp;

▶ **Fonte**

c:\gauss\srcnic\ana_esp\estimacao.src;

3 Binary Choice [bc]

3.1 probit_01

▶ **Objectivo**

▶ **Formato**

{beta,f,cov,retcode}=probit_01(x,b0);

▶ **Input**

x: matriz de ordem $n \times (K+1)$. Primeira coluna: variável binaria dependente; restantes variáveis: variáveis explicativas.

b0 b0=0.1*ones(cols(x),1);

▶ **Output**

▶ **Library**

library bc,cml;

▶ **Observações**

▶ **Fonte**

c:\gauss\srcnic\binary_choice\models.src

4 Bull & Bear[bull_bear]

4.1 break_ET

▶ Purpose

From Nicolau (2016). Structural Change Test in Duration of Bull and Bear Markets:

> "We propose a recursive test, derived from the fluctuations test of Ploberger-Kramer-Kontrus, with a finite sample adjustment, to test possible structural changes in duration of bull and bear markets."

▶ Format

{Q2,sup_Q2,ET}=break_MC(S,estado,w);

▶ Input

S: $n \times 1$ state space $\{1, 2\}$ or $\{0, 1\}$

estado: indicates the state in which is test is applied

w: start-up value

▶ Output

Q2: $n \times 1$ sequence $Q_{i,n}([rn]) = \sqrt{\frac{[rn]-w}{n-w}} \frac{\sqrt{[rn]}}{\hat{\sigma}_i} \left(\hat{\theta}_{i,[rn]} - \hat{\theta}_{i,n}\right),\ i = 0, 1$ (the first w values are zero)

sup_Q2: $\max Q_{i,n}([rn])$

ET: $n \times 1$ expected time (assuming a fisrt order MC)

▶ Library

library bull_bear;

▶ Fonte

c:\gauss\srcnic\bull_bear\rules.src

4.2 bull_bear_cycles_Lunde_Timmerman

▶ Purpose

Estimate Bull and Bear markets using the methodology of Lunde, A., & Timmermann, A. (2012). Duration dependence in stock prices. Journal of Business & Economic Statistics.

▶ Format

S= bull_bear_cycles_Lunde_Timmerman(p,lambda1,lambda2);

▶ Input

p: $n \times 1$ prices (not the log-prices)

lambda1: See Lunde and Timmermann. Common value 0.20

lambda2: See Lunde and Timmermann. Common value 0.15

▶ Output

S: $S = 2$ bull, $S = 1$ bear

▶ Library

library bull_bear;

▶ Fonte

c:\gauss\srcnic\bull_bear\rules.src

4.3 identify_financial_crisis

▶ **Objective**

VER create_dummies_financial_crisis(p,min0);

▶ **Formato**

{t0,t1,min,rA,x}=identify_financial_crisis (begins,ends,p);

▶ **Input**

▶ **Output**

▶ **Library**

library bull_bear;

▶ **Fonte**

c:\gauss\srcnic\bull_bear\rules.src

4.4 create_dummies_financial_crisis

▶ **Objective**

▶ **Format**

{w,d}=create_dummies_financial_crisis(p,min);

▶ **Input**

p: vector of prices $n \times 1$

min: scalar. The dummy variable takes on 1 in all the periods in which the return in percentage is lower than min

▶ **Output**

w: 4 columns: begins ~ends ~return ~return anualized. This matriz presents all-time low prices (begin date, end date and the associated returns)

d: dummy variable $n \times 1$

▶ **Library**

library bull_bear,pgraph;

▶ **Source**

c:\gauss\srcnic\bull_bear\rules.src

4.5 create_dummies_financial_crisis2

▶ **Objective**

▶ **Format**

{w,d,d_}=create_dummies_financial_crisis2(p,min,m);

▶ **Input**

p: vector of prices $n \times 1$

min: scalar. The dummy variable takes on 1 in all period in which the return in percentage is lower than min

m: m={1 2, 3 3,4 4} d_=d[.,1]+d[.,2] ~d[.,3] ~d[.,4]

▶ **Output**

w: 4 columns: begins ~ends ~return ~return anualized. This matriz presents all-time low prices (begin date, end date and the associated returns)

d: dummy variable $n \times 1$

d_: dummy variable $n \times \text{rows}(m)$

▶ Library

library bull_bear,pgraph;

▶ Source

c:\gauss\srcnic\bull_bear\rules.src

5 Cadeias de Markov Multivariadas

5.1 MMC_3D_eq

▶ Purpose

Estimates the parameters η_{ji} of the equation

$$P\left(S_{jt}=k|\, S_{1,t-1}=i_1, S_{2,t-1}=i_s\right)$$
$$=\frac{\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{jt}=k|\, S_{1,t-1}=i_1\right)+\eta_{j1}P\left(S_{jt}=k|\, S_{1,t-1}=i_2\right)+\eta_{j3}P\left(S_{jt}=k|\, S_{3,t-1}=i_3\right)\right)}{\sum_{i=1}^{m}\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{jt}=i|\, S_{1,t-1}=i_1\right)+\eta_{j1}P\left(S_{jt}=i|\, S_{1,t-1}=i_2\right)+\eta_{j3}P\left(S_{jt}=i|\, S_{3,t-1}=i_3\right)\right)}$$

using the maximum likelihood method (three categorical data). j is chosen by us.

▶ Format

{b,cov,logfv}= MMC_3D_eq(S,P,equation);

▶ Input

S: $n\times 3$ matrix, where n is the number of observations. Each element of S take values in the set $\{1,2,...,m\}$.

P: P is obtained after running the code {f,p,x0}=multivariate_Markov_Chain_01(S);

equation: scalar j, one element of the set $\{1,2,3\}$

▶ Output

b: Estimates of η_{j0},η_{j1}, η_{j2}, η_{j3} where j is defined according to input "equation"

cov:

logfv:

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima4.src

5.2 MMC_5D_eq

▶ Purpose

Estimates the parameters η_{ji} of the equation

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,...,S_{s,t-1}=i_s\right)$$
$$=p_{jk}=\frac{\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=k|\,S_{s,t-1}=i_s\right)\right)}{\sum_{i=1}^{m}\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=i|\,S_{s,t-1}=i_s\right)\right)}$$

using the maximum likelihood method (five categorical data). j is chosen by us.

▶ Format

{b,cov,logfv}= MMC_5D_eq(S,P,equation,b0);

▶ Input

S: $n\times 5$ matrix, where n is the number of observations. Each element of S take values in the set $\{1,2,...,m\}$.

P: P is obtained after running the code {f,p,x0}=multivariate_Markov_Chain_01(S);

equation: scalar j, one element of the set $\{1,2,3,4,5\}$

b0: 6×1 vector. If b0=0 all parameters are initialized with zero

▶ Output

b: Estimates of $\eta_{j0},\eta_{j1},\ \eta_{j2},...,\ \eta_{j5}$ where j is defined according to input "equation"

cov:

logfv:

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima4.src

5.3 MMC_preve_cat2_D

▶ **Purpose**

To calculate

$$P\left(S_{j,t+h}=i_0|\,S_{1t}=i_1,S_{2t}=i_2\right)$$
$$=\frac{\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{j,t+h}=i_0|\,S_{1t}=i_1\right)+\eta_{j2}P\left(S_{j,t+h}=i_0|\,S_{2,t}=i_2\right)\right)}{\sum_{k=1}^{m}\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{j,t+h}=i_0|\,S_{1t}=i_1\right)+\eta_{j2}P\left(S_{j,t+h}=i_0|\,S_{2,t}=i_2\right)\right)}$$

This expression requires

$$P\left(S_{j,t+h}=i_0|\,S_{kt}=i_k\right)=\sum_{\alpha=1}^{m}P\left(S_{j,t+h}=i_0|\,S_{k,t+h-1}=\alpha\right)P\left(S_{k,t+h-1}=\alpha|\,S_{kt}=i_k\right).$$

This expression is equal to the element (i_0,i_k) of the matrix product $P^{(jk)}\left(P^{(kk)}\right)^{h-1}$ where $P^{(jk)}$ is a matrix with elements $P\left(S_{jt}=i_0|\,S_{k,t-1}=i_k\right)$.

▶ **Format**

{preve1,preve2,s_preve1,s_preve2}=MMC_preve_cat2_D(b1,b2,P,s0,h);

▶ **Input**

b1: Is obtained from the code {b1,cov1,b2,cov2,prob}=mult_MC_estima_cat_2_D(S,P);

b2: Is obtained from the same code.

P: P is obtained after running {f,p,x0}=multivariate_Markov_Chain_01(S);

s0: vector (i_1,i_2), where these elements are such that $S_{1,t}=i_1,S_{2,t}=i_2$ $(i_1,i_2=1,2,...,m)$

▶ **Output**

preve1: $h\times m$ matrix associated with the forecasting of $S_{1,t+h}$. For example, the element $(5,2)$ represents

$$P\left(S_{1,t+5}=2|\,S_{1,t}=i_1,S_{2,t}=i_2\right)$$

preve2: $h\times m$ matrix associated with the forecasting of $S_{2,t+h}$. For example, the element $(10,3)$ represents

$$P\left(S_{2,t+10}=3|\,S_{1,t}=i_1,S_{2,t}=i_2\right)$$

s_preve1: $h\times 1$ gives the most likely state of $S_{1,t+h}$ (1 or 2 or ...m) for each h

s_preve2: $h\times 1$ gives the most likely state of $S_{2,t+h}$ (1 or 2 or ...m) for each h

▶ **Library**

library cmm;

▶ **Fonte**

c:\gauss\srcnic\cmm\preve.src

5.4 MMC_preve_cat5_D

▶ Purpose

Similar to MMC_preve_cat2_D, but for 5 categories.

▶ Format

{preve,s_preve}=MMC_preve_cat5_D(b1,P,s0,h);

▶ Input

b1: Is obtained from the code {b1,cov,logfv}= MMC_5D_eq(S,P,1);

P: P is obtained after running {f,p,x0}=multivariate_Markov_Chain_01(S);

s0: vector $(i_1, i_2, i_3, i_4, i_5)$, where these elements are such that $S_{1,t} = i_1, S_{2,t} = i_2, ..., S_{5,t} = i_5$ $(i_1, i_2 = 1, 2, ..., m)$

h:

▶ Output

preve: $h \times m$ matrix associated with the forecasting of $S_{1,t+h}$. For example, the element $(5, 2)$ represents

$$P\left(S_{1,t+5} = 2 | \, S_{1,t} = i_1, S_{2,t} = i_2, ..., S_{5,t} = i_5\right)$$

s_preve1: $h \times 1$ gives the most likely state of $S_{1,t+h}$ (1 or 2 or ...m) for each h

▶ Library

library cmm;

▶ Fonte

c:\gauss\srcnic\cmm\preve.src

5.5 mult_MC_estima_cat_2

▶ **Objectivo**

Estimar o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\right)=\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+\lambda_{j2}\mathbf{P}^{(j2)}\mathbf{x}_{t-1}^{(2)},\qquad j=1,2$$

▶ **Formato**

{b1,b2,f,cov1,cov2}=mult_MC_estima_cat_2(S0,P);

▶ **Input**

S0: Matrix de tipo $n\times 2$. A matrix $S0$ assume valores no conjunto $M=\{1,2,...,m\}$. (usar a rotina categoriza_01).

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

▶ **Output**

b1: estimativas de λ_{11} e λ_{12}

b2: estimativas de λ_{21} e λ_{22}

f:

cov1:

cov2.

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima0.src

5.6 mult_MC_estima_cat_3

▶ **Objectivo**

Estimar o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\right)=\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+\lambda_{j2}\mathbf{P}^{(j2)}\mathbf{x}_{t-1}^{(2)}+\lambda_{j3}\mathbf{P}^{(j3)}\mathbf{x}_{t-1}^{(3)},\qquad j=1,2,3$$

▶ **Formato**

{b1,b2,b3,f,cov1,cov2,cov3,corr}=mult_MC_estima_cat_3(S0,P);

▶ **Input**

S0: Matrix de tipo $n \times 3$. A matrix $S0$ assume valores no conjunto $M = \{1, 2, ..., m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

► **Output**

► **Library**

library cmm,cml;

► **Fonte**

c:\gauss\srcnic\cmm\estima1.src

5.7 mult_MC_estima_cat_4

► **Objectivo**

Estimar o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)} \middle| \mathcal{F}_{t-1}\right) = \lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)} + ... + \lambda_{j4}\mathbf{P}^{(j4)}\mathbf{x}_{t-1}^{(4)}, \qquad j = 1, ..., 4$$

► **Formato**

{b1,b2,b3,b4,f,cov1,cov2,cov3,cov4}=mult_MC_estima_cat_4(S0,P);

► **Input**

S0: Matrix de tipo $n \times 4$. A matrix $S0$ assume valores no conjunto $M = \{1, 2, ..., m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

► **Output**

► **Library**

library cmm,cml;

► **Fonte**

c:\gauss\srcnic\cmm\estima1.src

5.8 mult_MC_estima_cat_5

► **Objectivo**

Estimar o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)} \middle| \mathcal{F}_{t-1}\right) = \lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)} + ... + \lambda_{j5}\mathbf{P}^{(j5)}\mathbf{x}_{t-1}^{(5)}, \qquad j = 1, ..., 5$$

► **Formato**

{b1,b2,b3,b4,b5,y1,y2,y3,y4,y5,cov1,cov2,cov3,cov4,cov5}=mult_MC_estima_cat_5(S0,P);

► **Input**

S0: Matrix de tipo $n \times 5$. A matrix $S0$ assume valores no conjunto $M = \{1, 2, ..., m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

► **Output**

y1: Vetor $nm \times 2$ (observações versus estimativas)

► **Library**

library cmm,cml;

► **Fonte**

c:\gauss\srcnic\cmm\estima1.src

5.9 mult_MC_estima_cat_5_B

▶ **Objectivo**

Estimar o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\right)=\Phi\left(\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+...+\lambda_{j5}\mathbf{P}^{(j5)}\mathbf{x}_{t-1}^{(5)}\right),\qquad j=1,...,5$$

assumindo

$$\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\sim Bernoulli\left(\Phi\left(\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+...+\lambda_{j5}\mathbf{P}^{(j5)}\mathbf{x}_{t-1}^{(5)}\right)\right).$$

ATENÇÃO PROBABILIDADES NÃO SOMAM 1 ...

▶ **Formato**

{b1,b2,b3,b4,b5,y1,y2,y3,y4,y5,cov1,cov2,cov3,cov4,cov5}=mult_MC_estima_cat_5_B(S0,P);

▶ **Input**

S0: Matrix de tipo $n\times 5$. A matrix $S0$ assume valores no conjunto $M=\{1,2,...,m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

▶ **Output**

y1: Vetor $nm\times 2$ (observações versus estimativas)

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima2.src

5.10 mult_MC_estima_cat_2_C

▶ **Objectivo**

▶ **Formato**

▶ **Input**

S0: Matrix de tipo $n \times 5$. A matrix S assume valores no conjunto $M = \{1, 2, ..., m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

▶ **Output**

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima3.src

5.11 mult_MC_estima_cat_3_C

▶ Purpose

Estimates the parameters λ_{ji} of the equation

$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,S_{2,t-1}=i_s\right)=\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+\lambda_{j2}P\left(S_{jt}=k|\,S_{2,t-1}=i_2\right)+\lambda_{j3}P\left(S_{jt}$

using the maximum likelihood method (three categorical data).

▶ Format

{a1,cov1,a2,cov2,a3,cov3}= mult_MC_estima_cat_3_C(S,P);

▶ Input

S: $n\times 3$ matrix, where n is the number of observations. Each element of S take values in the set $\{1,2,...,m\}$.

P: P is obtained after running the code {f,p,x0}=multivariate_Markov_Chain_01(S);

▶ Output

a1: Estimates of λ_{11}, λ_{12}, λ_{13}

cov1: Covariance Matrix of $\hat{\lambda}_{11}$, $\hat{\lambda}_{12}$, $\hat{\lambda}_{13}$

a2: Estimates of λ_{21}, λ_{22}, λ_{23}

cov2: Covariance Matrix of $\hat{\lambda}_{21}$, $\hat{\lambda}_{22}$, $\hat{\lambda}_{23}$

a3: Estimates of λ_{31}, λ_{32}, λ_{33}

cov3: Covariance Matrix of $\hat{\lambda}_{31}$, $\hat{\lambda}_{32}$, $\hat{\lambda}_{33}$

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima3.src

5.12 mult_MC_estima_cat_3_C_preve1

▶ Objectivo

Preve

$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,S_{2,t-1}=i_s\right)=\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+\lambda_{j2}P\left(S_{jt}=k|\,S_{2,t-1}=i_2\right)+\lambda_{j3}P\left(S_{jt}$

dada a informação

$$\text{S0}=\begin{bmatrix}S_{1,t-1}\\S_{2,t-1}\\S_{3,t-1}\end{bmatrix}=\begin{bmatrix}i_1\\i_2\\i_3\end{bmatrix},\qquad i_1,i_2,i_3\in\{1,...,m\}$$

▶ Formato

prob=mult_MC_estima_cat_3_C_preve1(b,P,s,equation);

▶ Input

b: vector $\hat{\lambda}_{j1},...,\hat{\lambda}_{jm}$ obtido através da rotina "mult_MC_estima_cat_3_D"

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

S: Vector de tipo 3×1. A matrix S assume valores no conjunto $M=\{1,2,...,m\}$.

equation: escalar pertencente ao conjunto $\{1,2,3\}$ (3 equações). Deve ser compatível com a informação **b**

▶ Output

▶ Library

library cmm;

▶ Fonte

c:\gauss\srcnic\cmm\estima3.src

5.13 mult_MC_estima_cat_5_C

▶ Objectivo

Estima MMC sob a hipótese

$$P\left(S_{jt}=k|\, S_{1,t-1}=i_1,...,S_{s,t-1}=i_s\right)=\lambda_{j1}P\left(S_{jt}=k|\, S_{1,t-1}=i_1\right) \\ +...+\lambda_{js}P\left(S_{jt}=k|\, S_{s,t-1}=i_s\right)$$

i.e.

$$\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\,\mathcal{F}_{t-1}\right)=\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+...+\lambda_{j5}\mathbf{P}^{(j5)}\mathbf{x}_{t-1}^{(5)},\qquad j=1,...,5$$

através do método da máxima verosimilhança: (confirmar****)

$$\max_{\lambda_{1j}\in(0,1)}\sum_{i_5i_4i_3i_2i_1i_0}^{m}n_{i_5i_4i_3i_2i_11}\log\left(\sum_{j=1}^{5}\lambda_{1j}P\left(S_{jt}=1|\, S_{j,t-1}=j\right)\right)$$
$$...$$
$$\max_{\lambda_{5j}\in(0,1)}\sum_{i_5i_4i_3i_2i_1i_0}^{m}n_{i_5i_4i_3i_2i_15}\log\left(\sum_{j=1}^{5}\lambda_{5j}P\left(S_{jt}=5|\, S_{j,t-1}=j\right)\right)$$

▶ Formato

{b1,cov1,b2,cov2,b3,cov3,b4,cov4,b5,cov5}=mult_MC_estima_cat_5_C(S,P);

▶ Input

S0: Matrix de tipo $n\times 5$. A matrix S assume valores no conjunto $M=\{1,2,...,m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

▶ Output

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima3.src

5.14 mult_MC_estima_cat_2_D

▶ **Objectivo**

▶ **Formato**

{b1,cov1,b2,cov2,prob}=mult_MC_estima_cat_2_D(S,P);

▶ **Input**

S:

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

▶ **Output**

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima4.src

5.15 mult_MC_estima_cat_3_D

▶ Purpose

Estimates the parameters η_{ji} of the equation

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,S_{2,t-1}=i_s\right)=\frac{\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+\eta_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_2\right)\right.}{\sum_{i=1}^{m}\Phi\left(\eta_{j0}+\eta_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=i_1\right)+\eta_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=\right.\right.}$$

using the maximum likelihood method (three categorical data).

▶ Format

{a1,cov1,a2,cov2,a3,cov3,prob}= mult_MC_estima_cat_3_D(S,P);

▶ Input

S: $n\times 3$ matrix, where n is the number of observations. Each element of S take values in the set $\{1,2,...,m\}$.

P: P is obtained after running the code {f,p,x0}=multivariate_Markov_Chain_01(S);

▶ Output

a1: Estimates of η_{10}, η_{11}, η_{12}, η_{13}

cov1: Covariance Matrix of $\hat{\eta}_{10}$, $\hat{\eta}_{11}$, $\hat{\eta}_{12}$, $\hat{\eta}_{13}$

a2: Estimates of η_{20}, η_{21}, η_{22}, η_{23}

cov2: Covariance Matrix of $\hat{\eta}_{20}$, $\hat{\eta}_{21}$, $\hat{\eta}_{22}$, $\hat{\eta}_{23}$

a3: Estimates of η_{30}, η_{31}, η_{32}, η_{33}

cov3: Covariance Matrix of $\hat{\eta}_{30}$, $\hat{\eta}_{31}$, $\hat{\eta}_{32}$, $\hat{\eta}_{33}$

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima4.src

5.16 mult_MC_estima_cat_3_D_preve1

▶ **Objectivo**

Preve

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,...,S_{s,t-1}=i_s\right)$$
$$=p_{jk}=\frac{\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=k|\,S_{s,t-1}=i_s\right)\right)}{\sum_{i=1}^{m}\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=i|\,S_{s,t-1}=i_s\right)\right)}$$

dada a informação

$$\text{S0}=\begin{bmatrix}S_{1,t-1}\\S_{2,t-1}\\S_{3,t-1}\end{bmatrix}=\begin{bmatrix}i_1\\i_2\\i_3\end{bmatrix},\qquad i_1,i_2,i_3\in\{1,...,m\}$$

▶ **Formato**

prob=mult_MC_estima_cat_3_D_preve1(b,P,s,equation);

▶ **Input**

b: vector $\hat{\lambda}_{j1},...,\hat{\lambda}_{jm}$ obtido através da rotina "mult_MC_estima_cat_3_D"

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

S0: Vector de tipo 3×1. A matrix S assume valores no conjunto $M=\{1,2,...,m\}$.

equation: escalar pertencente ao conjunto $\{1,2,3\}$ (3 equações). Deve ser compatível com a informação **b**

▶ **Output**

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima4.src

5.17 mult_MC_estima_cat_5_D

▶ **Objectivo**

Estima MMC sob a hipótese

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,...,S_{s,t-1}=i_s\right)$$
$$=p_{jk}=\frac{\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=k|\,S_{s,t-1}=i_s\right)\right)}{\sum_{i=1}^{m}\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=i|\,S_{s,t-1}=i_s\right)\right)}$$

através do método da máxima verosimilhança:

$$\max_{\lambda_{1j}}\sum_{i_5i_4i_3i_2i_1i_0}^{m}n_{i_5i_4i_3i_2i_1}\log\left(p_{1i_0}\right)$$
$$...$$
$$\max_{\lambda_{5j}}\sum_{i_5i_4i_3i_2i_1i_0}^{m}n_{i_5i_4i_3i_2i_1}\log\left(p_{5i_0}\right)$$

▶ **Formato**

{b1,cov1,b2,cov2,b3,cov3,b4,cov4,b5,cov5,prob,FV}=mult_MC_estima_cat_5_D(S,P);

▶ **Input**

S0: Matrix de tipo $n\times 5$. A matrix S assume valores no conjunto $M=\{1,2,...,m\}$.

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

▶ **Output**

prob: Matriz

$$\begin{bmatrix}p_{11}&\cdots&p_{51}\\p_{12}&\cdots&p_{52}\\\vdots&&\vdots\\p_{1m}&\cdots&p_{5m}\end{bmatrix}$$

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima4.src

5.18 mult_MC_estima_cat_5_D_preve1

▶ **Objectivo**

Preve

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,...,S_{s,t-1}=i_s\right)$$
$$=p_{jk}=\frac{\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=k|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=k|\,S_{s,t-1}=i_s\right)\right)}{\sum_{i=1}^{m}\Phi\left(\lambda_{j0}+\lambda_{j1}P\left(S_{jt}=i|\,S_{1,t-1}=i_1\right)+...+\lambda_{js}P\left(S_{jt}=i|\,S_{s,t-1}=i_s\right)\right)}$$

dada a informação

$$\text{S0}=\begin{bmatrix}S_{1,t-1}\\S_{2,t-1}\\S_{3,t-1}\\S_{4,t-1}\\S_{5,t-1}\end{bmatrix}=\begin{bmatrix}i_1\\i_2\\i_3\\i_4\\i_5\end{bmatrix},\qquad i_1,...,i_5\in\{1,...,m\}$$

▶ **Formato**

prob=mult_MC_estima_cat_5_D_preve1(b,P,S,equation);

▶ **Input**

b: vector $\hat{\lambda}_{j1},...,\hat{\lambda}_{jm}$ obtido através da rotina "mult_MC_estima_cat_5_D"

P: obter P através da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

S: Vector de tipo 5×1. A matrix S assume valores no conjunto $M=\{1,2,...,m\}$.

equation: escalar pertencente ao conjunto $\{1,2,...,5\}$ (5 equações) Deve ser compatível com a informação **b**

▶ **Output**

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima4.src

5.19 multivariate_Markov_Chain_01

▶ Purpose

Estimates the following expressions of Ching e Ng (2006):

$F^{(jk)}$ (page 145)
$P^{(jk)}$ (page 145)
$\hat{x}$ (page 145).

Note-se

$$P_{ab}^{(jk)} = P\left(S_{jt} = a | \, S_{k,t-1} = b\right).$$

Reference: Ching W. e Ng M. (2006) Markov chains: models, algorithms and applications. New York: Springer.

▶ Formato

{f,p,x0}=multivariate_Markov_Chain_01(S);

▶ Input

S: $n \times s$ matrix where n is the numberr of observations and s is the number of categorical sequences. Each element of S take values in the set $M = \{1, 2, ..., m\}$. Example:

$$S0 = \begin{bmatrix} 1\,2\,1\,1 \\ 1\,3\,1\,2 \\ 2\,2\,1\,3 \\ 3\,3\,3\,2 \\ 1\,2\,3\,3 \end{bmatrix}$$

$n = 5$, $s = 4$, $M = \{1, 2, 3\}$.

▶ Output

F: $s \times s$ matrix of matrices.Each element of F is a $m \times m$ matrix. See page 145 of Ching e Ng (2006).

P: $s \times s$ matrix of matrices. Each element of P is a $m \times m$ matrix. See page 145 of Ching e Ng (2006). For example,

if $s = 2$ one has

$$\begin{bmatrix} \text{Plane[1,.,.]} & = P^{11} & \text{Plane[3,.,.]} & = P^{12} \\ \text{Plane[2,.,.]} & = P^{21} & \text{Plane[4,.,.]} & = P^{22} \end{bmatrix};$$

if $s = 3$ one has

$$\begin{bmatrix} \text{Plane[1,.,.]} & = P^{11} & \text{Plane[4,.,.]} & = P^{12} & \text{Plane[7,.,.]} & = P^{13} \\ \text{Plane[2,.,.]} & = P^{21} & \text{Plane[5,.,.]} & = P^{22} & \text{Plane[8,.,.]} & = P^{23} \\ \text{Plane[3,.,.]} & = P^{31} & \text{Plane[6,.,.]} & = P^{32} & \text{Plane[9,.,.]} & = P^{33} \end{bmatrix};$$

if $s = 5$ one has

$$\begin{bmatrix} [1,.,.] = P^{11} & [6,.,.] = P^{12} & [11,.,.] = P^{13} & [16,.,.] = P^{14} & [21,.,.] = P^{15} \\ [2,.,.] = P^{21} & [7,.,.] = P^{22} & [12,.,.] = P^{23} & [17,.,.] = P^{24} & [22,.,.] = P^{25} \\ [3,.,.] = P^{31} & [8,.,.] = P^{32} & [13,.,.] = P^{33} & [18,.,.] = P^{34} & [23,.,.] = P^{35} \\ [4,.,.] = P^{41} & [9,.,.] = P^{42} & [14,.,.] = P^{43} & [19,.,.] = P^{44} & [24,.,.] = P^{45} \\ [5,.,.] = P^{51} & [10,.,.] = P^{52} & [15,.,.] = P^{53} & [20,.,.] = P^{54} & [25,.,.] = P^{55} \end{bmatrix}$$

x0: See page 145 of Ching e Ng (2006).

► Library

library cmm;

► Fonte

c:\gauss\srcnic\cmm\estimao.src

5.20 prediction_accuracy

▶ Objectivo

Calcula a % de vezes em que a maior probabilidade indica corretamente o evento futuro.

▶ Formato

{prob,pred}=prediction_accuracy3(&f,b,p,s,equation);

▶ Input

&f: hipóteses:

&mult_MC_estima_cat_2_D_preve1 ou

&mult_MC_estima_cat_3_C_preve1 ou

&mult_MC_estima_cat_3_D_preve1 ou

&mult_MC_estima_cat_5_D_preve1

b:

p: estimativa obtida a partir da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

s: vector de dimensão $n \times 3$ ou $n \times 5$

▶ Output

prob: probabilidades estimadas dos vários estados ao longo do tempo (pode ser comparado com S[2:n,equation])

pred: % de vezes em que a maior probabilidade indica corretamente o evento futuro

▶ Library

library cmm,cml;

▶ Fonte

c:\gauss\srcnic\cmm\estima4.src

5.21 preve_cat_3

▶ **Objectivo**

Prevê o valor de $\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\right)$ (são probabilidades) tendo em conta o modelo

$$\mathrm{E}\left(\mathbf{x}_t^{(j)}\middle|\mathcal{F}_{t-1}\right)=\lambda_{j1}\mathbf{P}^{(j1)}\mathbf{x}_{t-1}^{(1)}+\lambda_{j2}\mathbf{P}^{(j2)}\mathbf{x}_{t-1}^{(2)}+\lambda_{j3}\mathbf{P}^{(j3)}\mathbf{x}_{t-1}^{(3)},\qquad j=1,2,3$$

▶ **Formato**

xp=preve_cat_3(b1,b2,b3,p,s);

▶ **Input**

b1,b2,b3: Estimativas obtidas a partir da rotina

{b1,b2,b3,f,cov1,cov2,cov3,corr}=mult_MC_estima_cat_3(S0,P);

p: estimativa obtida a partir da rotina {f,p,x0}=multivariate_Markov_Chain_01(S0);

s: vector de dimensão 3×1. Representa as categorias em que S_1, S_2 e S_3 se encontram no período anterior ao da previsão. Por exemplo, supondo que a previsão é para o período t, s=1|4|2 significa

$$S_{1t-1}=1, S_{2t-1}=4, S_{3t-1}=2$$

ou (assumindo que $m=4$)

$$\mathbf{x}_{t-1}^{(1)}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix},\ \mathbf{x}_{t-1}^{(2)}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix},\ \mathbf{x}_{t-1}^{(3)}=\begin{bmatrix}0\\1\\0\\0\end{bmatrix}.$$

▶ **Output**

xp: vector $m\times 3$. Primeira coluna de xp fornece as probabilidades de S_1 se encontrar em cada uma das m categorias (valores de previsão). De igual forma para as outras colunas.

▶ **Library**

library cmm,cml;

▶ **Fonte**

c:\gauss\srcnic\cmm\estimacao1.src

5.22 standard_MMC

▶ Purpose

Given $S_{1t}, ..., S_{st}$ with state space $\{1, ..., m_1\}, ..., \{1, ..., m_s\}$ the procedure calculates

$$P\left(S_{1t} = i_0 | \, S_{1t-1} = i_1, ..., S_{st-1} = i_s\right)$$

for $i_0 \in \{1, ..., m_1\}$, $i_j \in \{1, ..., m_j\}$.

			S_{1t}		
$S_{1,t-1}$		$S_{s,t-1}$	1	$\cdots$	m_1
1	$\cdots$	1	$P\left(S_{1t} = 1 \mid S_{1t-1} = 1, ..., S_{st-1} = 1\right)$		$P\left(S_{1t} = m_1 \mid S_{1t-1} = 1, ..., S$
$\vdots$	$\cdots$	$\vdots$			$\vdots$
1	$\cdots$	m_s	$P\left(S_{1t} = 1 \mid S_{1t-1} = 1, ..., S_{st-1} = m_s\right)$	$\cdots$	$P\left(S_{1t} = m_1 \mid S_{1t-1} = 1, ..., S_s$
$\vdots$		$\vdots$			$\vdots$
m_1	$\cdots$	m_s	$P\left(S_{1t} = 1 \mid S_{1t-1} = m_1, ..., S_{st-1} = m_s\right)$	$\cdots$	$P\left(S_{1t} = m_1 \mid S_{1t-1} = m_1, ..., S$

Let Q be the matrix associated with the table

$S_{1,t-1}$		$S_{s,t-1}$
1	$\cdots$	1
$\vdots$	$\cdots$	$\vdots$
1	$\cdots$	m_s
$\vdots$		$\vdots$
m_1	$\cdots$	m_s

and $P = [P\left(S_{1t} = i_0 | \, S_{1t-1} = i_1, ..., S_{st-1} = i_s\right)]$

▶ Formato

{Q,P,SE}=call standard_MMC(S,mostrar);

▶ Input

S: $n \times s$ matrix where n is the numberr of observations and s is the number of categorical sequences. The state space of each column may be different.

mostrar: if 1 then Q, P and t-ratios are printed

▶ Output

Q: $(m_1 m_2 ... m_s) \times s$ matrix

P: $(m_1 m_2 ... m_s) \times m_1$

SE: Standar Errors $SE[i,j] = \sqrt{P[i,j] * (1 - P[i,j]) / \hat{P}\left(S_{1t-1} = i_1, ..., S_{st-1} = i_s\right)} * \sqrt{1/n}$

▶ **Library**

library cmm;

▶ **Fonte**

c:\gauss\srcnic\cmm\estima0.src

5.23 test_independ_cat5

▶ **Objectivo**

Testa

$$P\left(S_{jt}=k|\,S_{1,t-1}=i_1,...,S_{5,t-1}=i_s\right)=P\left(S_{jt}=k|\,S_{j,t-1}=i_j\right)$$

▶ **Formato**

{logfv_h0,logfv_H1,rv}=test_independ_cat5(S,P,logfv_H1);

▶ **Input**

▶ **Output**

▶ **Library**

library cmm;

▶ **Fonte**

c:\gauss\srcnic\cmm\test.src

5.24 testa_nulidade_CMM

▶ Objectivo

Testa a nulidade de parâmetros num modelo CMM através do teste rácio de verosimilhanças.

▶ Formato

call testa_nulidade_CMM(&f,logfv1,par_num,S,P,equation);

▶ Input

&f: Opções:

&MMC_3C_eq
&MMC_3D_eq
&MMC_5D_eq

logfv1: Valor máximo da função de verosimilhança do modelo sob hipótese alternativa (modelo sem restrições).

par_num: Escalar ou vector. Exemplo: par_num=2 testa a nulidade do segundo parâmetro; par_num=2|4 testa a nulidade do segundo e quarto parâmetro.

S: Matriz das observações da cadeia.

P: Estimativa obtida a partir da rotina {f,p,x0}=multivariate_Markov_Chain_01(S);

equation: número da equação (está relacionado com a ordem das colunas de S).

▶ Output

Fornece o valor da estatística rácio de verosimilhanças e o respetico p-value

▶ Library

library cml, cmm;

▶ Fonte

c:\gauss\srcnic\cmm\test.src

6 Diagnóstico [diagnos]

6.1 diagn

▶ Purpose

Analyses the time series or analyses the estimated model.

▶ Format

diagn(x,media,var);

▶ Input

x: observations X.

media: scalar or conditional mean (same length as the dimension of X).

var: scalar or conditional variance.

▶ Output

If media is a scalar (one assumes that there wasn't estimation - the focus is only on the time series X) the procedures presents:

- *kurtosis*.
- *skewness*.
- standardized *kurtosis* (under H0:kurt=3 the standardized kurtosis has N(0,1) distribution).
- standardized *skewness* (under H0:skew=0 the standardized skewness has N(0,1) distribution).
- Bera-Jarque p-value (tests normality of X).
- Ljung-Box p-value (tests serial correlation of X). The autocorrelation order can be controlled through ordem_aut (default = 2).
- ARCH test p-value. (LM test). The order of the ARCH process can be controlled through ordem_ARCH (default = 1).
- Dickey-Fuller test:

  "ADF statistic for random walk without drift"

  "ADF statistic for random walk with drift"

  "ADF statistic for random walk with drift and trend"
- Graph of X over time.
- Graph - Autocorrelation function of X (the order of autocorrelation can be controlled through ordem_aut).

- Graph - Partial Autocorrelation function of X (the order of autocorrelation can be controlled through ordem_aut).
- Graph - kernel density versus density of $N\left(\bar{X}, \hat{\sigma}^2\right)$.

If media has the same dimension as X (one assumes a estimated model) the procedures presents the following statistics and graph of

$$\varepsilon_t = \frac{X_t - \hat{\mu}_t}{\hat{\sigma}_t}$$

- R squared
- *kurtosis* of ε.
- *skewness* of ε.
- standardized *kurtosis* of ε (under H0:kurt=3 the standardized kurtosis has N(0,1) distribution)
- standardized *skewness* de ε (under H0:skew=0 the standardized skewness has N(0,1) distribution).
- Bera-Jarque p-value (tests normality of ε).
- Ljung-Box p-value (tests serial correlation of X). The autocorrelation order can be controlled through ordem_aut (default = 2).
- ARCH test p-value. (LM test). The order of the ARCH process can be controlled through ordem_ARCH (default = 1).
- Graph of X and μ_t over time.
- Graph of ε over time.
- Graph of $\hat{\sigma}_t^2$ over time.
- Graph - Autocorrelation function of ε (the order of autocorrelation can be controlled through ordem_aut).
- Graph - Partial Autocorrelation function of ε (the order of autocorrelation can be controlled through ordem_aut).
- Graph - kernel density of ε versus density of $N(0,1)$.

▶ Library

Library pgraph,diagnos;

▶ Remarks

Note: The graphics may not be displayed if graf is set to 0 (i.e. graf=0) (graf is a global variable). Other global variables: coef_cor, ordem_aut (default: 5), ordem_arch (default: 5), ordem_lag (default: 5) controls the number of lags of the dependent variable in the ADF test.

▶ Source

c:\gauss\srcnic\diagnos\diagnos.src

6.2 teste_RU_ao_longo_amostra

▶ Objectivo

Calcular e apresentar graficamente as estatísticas rácio-t, valor crítico e parâmetro autoregressivo no instante t, respectivamente, r_t, vc_t e ϕ_t, associadas ao teste Dickey-Fuller (DF). Dada uma amostra de n observações de Y e fixado um valor n_inicio, aquelas estatísticas são calculadas considerando-se (n-n_inicio+1) regressões do teste DF: a primeira regressão inclui os valores de $Y_1, ..., Y_{n\_inicio}$ que fornece r_{n_inicio}, vc_{n_inicio} e ϕ_{n_inicio}, a segunda regressão inclui os valores de $Y_1, ..., Y_{n\_inicio+1}$ que fornece $r_{n\_inicio+1}$, $vc_{n\_inicio+1}$ e $\phi_{n\_inicio+1}$, etc., até, finalmente $Y_1, ..., Y_n$ que fornece r_n, vc_n e ϕ_n.

▶ Formato

{raciot,vc,par_ar}=teste_RU_ao_longo_amostra(y,n_inicio,des,p,ns);

▶ Input

y: vector das observações do processo

n_inicio: valor inteiro entre 5 e n (ver explicação em "objectivo").

des: número de desfasamentos de Y a incluir na regressão do teste DF.

p: ordem do polinómio a incluir na regressão do teste DF (fixar p=-1 no caso de não existir parte determinística).

ns: nível de significância do teste DF. Valores possíveis: 1, 5, 10, 90, 95, 99.

▶ Output

raciot: vector de tipo $(n - n\_inicio + 1) \times 1$ dos rácio-t.

vc: vector de tipo $(n - n\_inicio + 1) \times 1$ dos valores críticos.

par_ar: vector de tipo $(n - n\_inicio + 1) \times 1$ dos parâmetros autoregressivos.

▶ Library

library pgraph,diagnos,coin;

▶ Fonte

c:\gauss\srcnic\diagnos\diagnos.src

7 Estimação de Equações Diferenciais Estocásticas [est_ede]

7.1 bootstrap1

▶ Objectivo

Dado um conjunto de observações $(X_{t_1}, ..., X_{t_n})$ e a EDE $dX_t = a(X_t; \theta_1)\, dt + b(X_t; \theta_2)\, dW_t$ determinar através de um procedimento bootsrap os vectores nc_{1,t_i} e nc_{2,t_i} tais que

$$nc_{1,t_i} : P[X_{t_i} \leq nc_{1,t_i}] = \frac{1-ic}{2} \quad \text{(NC)}$$

$$nc_{2,t_i} : P[X_{t_i} \geq nc_{2,t_i}] = \frac{1+ic}{2} \quad \text{(NC)}$$

para $i = 1, ..., n$, ou determinar os vectores c_{1,t_i} e c_{2,t_i} tais que

$$c_{1,t_i} : P\left[X_{t_i} \leq c_{1,t_i} | X_{t_{i-1}}\right] = \frac{1-ic}{2} \quad \text{(C)}$$

$$c_{2,t_i} : P\left[X_{t_i} \geq c_{2,t_i} | X_{t_{i-1}}\right] = \frac{1+ic}{2} \quad \text{(C)}$$

para $i = 1, ..., n$.

Nota: Numa aplicação concreta, depois de se estimar o modelo $dX_t = a(X_t; \theta_1)\, dt + b(X_t; \theta_2)\, dW_t$, tem interesse verificar se a trajectória observada de X se encontra delimitada, por exemplo, pelas trajectórias de nc_1 e nc_2 (ou c_1 e c_2). Como nc_1 e nc_2 (ou c_1 e c_2) são determinadas pelo modelo estimado $dX_t = a(X_t; \theta_1)\, dt + b(X_t; \theta_2)\, dW_t$, se ocorre com frequência observar que $X_{t_i} > nc_{2,t_i}$ ou $X_{t_i} < nc_{1,t_i}$ pode-se concluir que o modelo está mal especificado. Por exemplo, $nc_{2,t_i} = 100$ (para um certo i) significa que $(1+ic)/2 \times 100\%$ dos valores de X_{t_i} simulados a partir de $dX_t = a(X_t; \theta_1)\, dt + b(X_t; \theta_2)\, dW_t$ se encontram acima do valor 100. Se o modelo especificado está correcto, deveremos esperar que a probabilidade de X_{t_i} se encontrar acima de 100 seja exactamente $(1+ic)/2$. Sendo esta probabilidade, em princípio pequena, se de facto se observa $X_{t_i} > 100$, poderemos duvidar da especificação do modelo. Se entretanto ocorre com frequência observar $X_{t_i} > nc_{2,t_i}$ ou $X_{t_i} < nc_{1,t_i}$ (para vários i) teremos fortes razões para rejeitar o modelo.

A simulação de X é feita de acordo com a equação (**??**) onde o parâmetro N permite controlar a precisão do esquema de simulação.

▶ Formato

{liminf,limsup}=bootstrap1(&f1,&f2,par1,par2,x,d,s,nn,ic,cond);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x; \theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x; \theta_2)$.

par1: valor do vector θ_1.

par2: valor do vector θ_2.

x: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

s: número de simulações.

nn: valor do parâmetro N [ver equação (**??**)].

ic: parâmetro ic (0<ic<1).

cond: se cond=0 consideram-se as equações (NC); se cond=1 consideram-se as equações (C).

▶ Output

liminf: vector nc_1 de tipo $n \times 1$ no caso cond=0; vector c_1 de tipo $n \times 1$ no caso cond=1.

limsup: vector nc_2 de tipo $n \times 1$ no caso cond=0; vector c_2 de tipo $n \times 1$ no caso cond=1.

▶ Library

library pgraph,diagnos,est_ede;

▶ Fonte

c:\gauss\srcnic\diagnos\bootstra.src

7.2 bootstrap2

▶ Objectivo

Determinar, através de um procedimento bootstrap, os intervalos de confiança das estimativas não paramétricas $\hat{a}(x)$ e $v(x)$ (ver "drift_kernel2" e "var_kernel3") associado ao processo $dX_t = a(X_t;\theta)\,dt + b(X_t;\theta)\,dW_t$.

▶ Formato

{liminf_med,limsup_med,liminf_var,limsup_var}=bootstrap2(&f1,&f2,&f3,par1,par2,d,n,x0,s,nn,ic

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

&f3: ponteiro para um procedimento que define o esquema de discretização para simular X. Pode assumir &f3=&simulaNC1 - X é simulado de acordo com a equação (**??**) ou &f3=&simulaNC2 - X é simulado de acordo com o esquema de Milstein.

par1: valor do vector θ_1.

par2: valor do vector θ_2.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações de cada trajectória de X.

x0: valor inicial de X.

s: número de simulações

nn: valor do parâmetro N [ver a equação (**??**)].

ic: intervalo de confiança (0<ic<1).

▶ Output

liminf_med: limite inferior do intervalo de confiança para $\hat{a}(x)$

limsup_med: limite superior do intervalo de confiança para $\hat{a}(x)$

liminf_var: limite inferior do intervalo de confiança para $v(x)$

limsup_var: limite superior do intervalo de confiança para $v(x)$

Library

library est_ede,diagnos;

▶ Observações

A variável global pontos permite definir o número de pontos x_i (por defeito assume-se 15 pontos). A variável global _ic permite definir o parâmetro ic nas rotinas drift_kernel2 e var_kernel3 (por defeito _ic=0.95; não confundir este parâmetro com o parâmetro o ic da presente rotina).

▶ Fonte

c:\gauss\srcnic\diagnos\bootsra.src

7.3 drift_kernel1

▶ Objectivo

Estimar não parametricamente o coeficiente de tendência infinitesimal $a(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$\hat{a}(x) = \frac{\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}} \frac{(X_{t_{i+1}}-X_{t_i})}{\Delta}}{\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}}.$$

▶ **Formato**

{x,med,medy,k,lim_inf,lim_sup} = drift_kernel1(y,d,pontos,h,interpola,controle,z);

▶ **Input**

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

d: Δ, intervalo entre duas observações consecutivas, constante.

pontos: número de pontos x_i para os quais se calcula $\hat{a}(x_i)$. Isto é $i = 1, 2, ...,$pontos. Aconselha-se um valor entre 10 e 15.

interpola: assume $\{0, 1\}$. Se interpola = 1 calcula, por interpolação, $a(x_i)$, onde x_i assume todos os valores observados do processo X.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $a(x)$ apenas quando

$$\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}} \geq \text{controle}.$$

Aconselha-se um valor entre 10 e 20.

z: quantil para o cálculo do intervalo de confiança de a (por exemplo, z = 1.96 - tem associado um intervalo de confiança de 95%)

▶ **Output**

x: vector x de tipo $k \times 1$ onde $k \leq$ pontos.

med: vector $\hat{a}(x)$ de tipo $k \times 1$

medy: igual a zero se interpola = 0 e igual a $\hat{a}(x_i)$ de tipo $n \times 1$ onde n é o número de observação de X se interpola = 1.

k: vector $\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}$ de tipo $k \times 1$.

lim_inf: vector de tipo pontos×1 referente ao limite inferior do intervalo de confiança para $\hat{a}(x_i)$.

lim_sup: vector de tipo pontos×1 referente ao limite superior do intervalo de confiança para $\hat{a}(x_i)$.

▶ **Library**

library est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.4 drift_kernel2

▶ Objectivo

Estimar não parametricamente o coeficiente de tendência infinitesimal $a(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$\hat{a}(x) = \frac{\sum_{i=1}^{n-1} K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_i}-X_{t_{i+1}}\right)}{\Delta}}{\sum_{i=1}^{n-1} K\left(\frac{X_{t_i}-x}{h}\right)}$$

onde $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5}\,\hat{\sigma}_X n^{-1/5}$.

▶ Formato

{x,m}=drift_kernel2(y,pontos,ic,d);

▶ Input

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

d: Δ, intervalo entre duas observações consecutivas, constante.

pontos: número de pontos x_i para os quais se calcula $\hat{a}(x_i)$. Isto é $i = 1, 2, ...,$pontos. Se pontos $= 0$, calcula-se $\hat{a}(x_i)$ onde x_i assume todos os valores da amostra $(X_{t_1}, ..., X_{t_n})$ que satisfazem o parâmetro ic (ver a seguir).

ic: assume um valor no intervalo $]0, 1]$. Se ic $\in]0, 1[$ o procedimento é apenas executado para os valores $(X_{t_1}, ..., X_{t_n})$ que se encontram entre o percentil $(1 - ic)/2$ e o percentil $(1 + ic)/2$. As estimativas de $a(x)$ são portanto aparadas. Se ic $= 1$ o procedimento é executado para todos os valores da amostra.

▶ Output

x: vector x.

med: vector $\hat{a}(x)$.

▶ Library

library est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.5 est_dens_Dacunha_Florens1

▶ Objectivo

Estimar, através de simulação, a densidade de transição $p(\Delta, x, y)$ associada à EDE $dX_t = a(X_t; \theta)\, dt + \sigma dW_t$ considerando [ver Dacunha-Castelle e Florens-Zmirou (1986)]:

$$\hat{p}(\Delta, x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-y)^2}{2\sigma^2\Delta} + \frac{G(y) - G(x)}{\sigma^2} \right\} \hat{\psi}$$

(ver as notações no capítulo **??**).

▶ Formato

p=est_dens_Dacunha_Florens1(&drift,&derivada,&funcao_G,S,delta,d,x,y,b,seed);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x; \theta)$.

&f2: ponteiro para um procedimento que define a especificação funcional de $da(x; \theta)/dx$.

&f3: ponteiro para um procedimento que define a especificação funcional de $G(y)$.

S: valor do parâmetro n_s.

delta: valor entre 0 e 1 (tal que 1/delta seja inteiro) associado à precisão da integração numérica $\int_0^1 g\left(z_u(x,y) + \sqrt{\sigma^2\Delta} B_u(\omega)\right) du$. A precisão aumenta quando delta diminui. Concretamente delta tem a ver com a forma como o processo de Wiener é simulado para se obter $B_t = W_t - uW_1$. Fixado delta, tem-se $t_i = i \times delta$, $i = 0, 1, ..., 1/delta$ e $W_{t_i} = W_{t_{i-1}} + \sqrt{delta}\varepsilon_i$ onde ε_i é $N(0,1)$ i.i.d. (notar: delta $= N^{-1}$).

d: Δ, intervalo de tempo entre $X_\Delta = y$ e $X_0 = x$.

x: valor inicial.

y: valor final.

b: vector $\left[\theta\, \sigma\right]'$.

seed: valor *seed* (inteiro) associado à simulação dos valores aleatórios.

▶ Output

p: estimativa para $p(\Delta, x, y)$.

▶ Library

library est_ede;

► **Exemplo**

Considere-se $dX_t = \beta(\tau - X_t)\,dt + \sigma dW_t$. Identificando $(\beta, \tau, \sigma) = (b[1], b[2], b[3])$, tem-se:

```
library est_ede;

proc drift(b,x);

    retp(b[1]*(b[2]-x));

endp;

proc derivada(b,x);

    retp(-b[1]);

endp;

proc funcao_G(b,x);

    retp(b[1]*b[2]*x-b[1]*x^2/2);

endp;

/* Introduzir os valores de S, delta,d, x,y,b,seed */

p=est_dens_Dacunha_Florens1(&drift,&derivada,&funcao_G,S,delta,d,x,y,b,seed);
```

► **Fonte**

c:\gauss\srcnic\est_ede\est_fdpc.src

7.6 est_dens_fmogeneous

► **Objectivo**

$$\hat{p}(s, x, t, y) =$$

► **Formato**

```
p=est_dens_inhomogeneous(&sigma,&f,&A1,&B1,s,x,t,y,b,NN,simula,seed);
```

► **Input**

&sigma:

&f:

&A1:

&B1:

s: t_0

x:

t: scalar

y:

b:

NN:

simula:

seed:

► **Output**

p:

► **Library**

library est_ede;

► **Exemplo**

► **Fonte**

c:\gauss\srcnic\est_ede\est_fdpc.src

7.7 est_dens_Nao_Param

► **Objectivo**

Estimar, através de simulação, a densidade de transição de $X_\Delta = y$ dado $X_0 = x$, $p(\Delta, x, y)$, onde X é solução da EDE $dX_t = a(X_t; \theta_1)\, dt + b(X_t; \theta_2)\, dW_t$. O estimador é

$$\hat{p}(\Delta, x, y) = \frac{1}{n_s h} \sum_{j=1}^{n_s} K\left(\frac{y - Y_j^{(N)}}{h}\right)$$

onde, $Y_j^{(N)}$ é a j-ésima aproximação para X_Δ dado $X_0 = Y_0 = x$, $K(u) = (2\pi)^{-1/2} e^{-u^2/2}$ e $h = (4/3)^{1/5} \hat{\sigma}_Y n_s^{-1/5}$.

► **Formato**

p=est_dens_Nao_Param(&f1,&f2,&f3,par1,par2,NN,ns,d,x,y);

► **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x; \theta_1)$ (se &f1=0 o drift é nulo).

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

&f3: ponteiro para um procedimento que define a simulação do processo (se &simula o processo é simulado de acordo com a equação (**??**). Se &f3 é diferente de "&simula" os inputs "nn" e "ns" não são considerados, embora se devam escrever números nas entradas respeitantes a estes inputs.

par1: valor inicial para o vector θ_1.

par2: valor inicial para o vector θ_2.

nn: valor (inteiro) do parâmetro N [ver equação (**??**)] (valores altos aumentam a precisão da simulação e, por essa via, da estimação).

ns: valor do parâmetro n_s.

d: Δ, intervalo de tempo entre $X_\Delta = y$ e $X_0 = x$.

x: valor inicial.

y: valor final.

▶ Output

p: estimativa para $p(\Delta, x, y)$.

▶ Library

library cml,est_ede;

▶ Exemplo

Exemplo de um procedimento para &f3:

```
proc simula_flvsmn(b,x);

    local ys,e;

    e=rndn(1,_ns);

    ys=b[2]+(x-b[2])*exp(-b[1]*d)+sqrt((b[3]^2)/(2*b[1])*(1-exp(-2*b[1]*d))).*e;

    retp(ys);

endp;
```

Nota: as variáveis _ns, _d, respectivamente n_s e Δ, são variáveis "globais" cujo valor é passado de procedimento para procedimento. Não alterar esta notação.

▶ Fonte

c:\gauss\srcnic\est_ede\est_fdpc.src

7.8 flvsmn

▶ Objectivo

Estimar uma EDE através do método da função log-verosimilhança simulada não parametricamente, i.e., considerando o problema

$$\max_{\theta} \frac{1}{n} \sum_{i}^{n} \log \left[\frac{1}{n_s h} \sum_{j=1}^{n_s} K \left(\frac{X_{t_i} - Y_{t_i,j}^{(N)}(\theta)}{h} \right) \right].$$

▶ Formato

{b,f0,g,cov,retcode}=flvsmn(&f1,&f2,&f3,seed,par1,res1,par2,res2,y,d,nn,ns);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$ (se &f1=0 o drift é nulo).

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

&f3: Se zero (i.e., &f3=0) X é simulado de acordo com a equação (**??**). No caso em que &f3 é efectivamente um ponteiro, definir o procedimento que simula X (ver exemplo abaixo). Neste último caso os inputs "nn" e "ns" não são considerados, embora se devam escrever números nas entradas respeitantes a estes inputs.

seed:

par1: valor inicial para o vector θ_1.

res1: vector de zeros e uns de dimensão igual à do vector θ_1. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

par2: valor inicial para o vector θ_2.

res2: vector de zeros e uns de dimensão igual à do vector θ_2. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

nn: valor do parâmetro N [ver equação (**??**)].

ns: valor do parâmetro n_s.

▶ Output

b: vector das estimativas de θ_1 e θ_2.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

▶ Library

library cml,est_ede;

▶ Exemplo

(1) Estimar a EDE $dX_t = b_1 (b_2 - X_t)\, dt + b_3 X_t^{b_4} dW_t$

```
...

/* ler o vector y */

...

/*introduzir valores para seed, par1, etc., por exemplo:*/

seed=123456;

par1={1,10}; /* guess */

rest1={0,1}; /* b1 ≥ 0 */

par2={1,.5};

rest2={0,0}; /* b3, b4 ≥ 0 */

d=.05;

nn=10;

ns=30;

{beta,f,g,cov,retcode}=flvsmn(&linear1,&potencia,0,seed,par1,res1,par2,res2,y,d,nn,ns);

/* nota &f3=0 */
```

(2) Exemplo de um procedimento para &f3:

```
proc simula_flvsmn(b,x);

    local ys,e;

    e=rndn(1,_ns);

    ys=b[2]+(x-b[2])*exp(-b[1]*d)+sqrt((b[3]^2)/(2*b[1])*(1-exp(-2*b[1]*d))).*e;

    retp(ys);
```

endp;

Nota: as variáveis _ns, _d, respectivamente n_s e Δ, são variáveis "globais" cujo valor é passado de procedimento para procedimento. Não alterar esta notação.

▶ Observações

A variável global _mostrar permite controlar a apresentação do output. Se _mostrar = 0 o resultado da estimação não é apresentado (conveniente para simulações Monte Carlo).

▶ Fonte

c:\gauss\srcnic\est_ede\flvsmn.src

7.9 inf_ind

▶ Objectivo

(Rotina em desenvolvimento - o comando abaixo funciona mas é necessário melhorar vários aspectos)

Estimar uma EDE através do método da inferência indirecta.

▶ Formato

b_inf_ind=inf_ind(&f1,&f2,&f3,par1,rest1,par2,rest2,y,d,nn,ns,seed);

7.10 kernel_fdp_G

▶ Objectivo

Estimar a densidade de probabilidade $f(\cdot)$ no ponto x. Estimador:

$$\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

onde $K(\cdot)$ é o kernel Gaussiano.

▶ Formato

p = kernel_fdp_G(vectory,x);

▶ Input

vectory: vector de observações X_i.

▶ Output

p: escalar, estimativa para $f(x)$.

▶ **Library**

library est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.11 kernel_fdp_U

▶ **Objectivo**

Estimar a densidade de probabilidade $f(\cdot)$ no ponto x. Estimador:

$$\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

onde $K(\cdot)$ é o kernel uniforme.

▶ **Formato**

p = kernel_fdp_U(vectory,x);

▶ **Input**

vectory: vector de observações X_i.

▶ **Output**

p: escalar, estimativa para $f(x)$.

▶ **Library**

library est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.12 mv_inhomogeneous_BKM

▶ **Objectivo**

Estimates the parameters of the EDE

$$dX_t = X_t\left(e^{\tau t} + \frac{\sigma^2}{2} - \alpha \log X_t\right) + \sigma X_t dW_t$$

through the maximum likelihood method.

▶ **Format**

{bmv,f0,grad,cov,retcode,media,var}=mv_inhomogeneous_BKM(b0,s,d,y);

▶ **Input**

b0: (τ, α, σ)

s: The initial value of time t_0

d:

y:

▶ **Output**

b:

f0:

gra:

cov:

retcode:

▶ **Library**

library cml,est_ede;

▶ **Source**

c:\gauss\srcnic\est_ede\mv_inhomogeneous.src

7.13 mv_inhomogeneous_GBM

▶ **Objectivo**

Estimates the parameters of the EDE

$$dX_t = \beta X_t dt + \sigma e^{\alpha t} X_t dW_t$$

through the maximum likelihood method.

▶ **Format**

{bmv,f0,grad,cov,retcode,media,var}=mv_inhomogeneous_GBM(b0,s,d,y);

▶ **Input**

b0: (β, σ, α)

s: The initial value of time t_0

d:

y:

▶ **Output**

b: $\left(\hat{\beta}, \hat{\sigma}, \hat{\alpha}\right)$

f0:

gra:

cov:

retcode:

▶ **Library**

library cml,est_ede;

▶ **Source**

c:\gauss\srcnic\est_ede\mv_inhomogeneous.src

7.14 mv_inhomogeneous_ornstein

▶ **Objectivo**

Estimates the parameters of the EDE

$$dX_t = -\beta X_t dt + \sigma e^{\alpha t} dW_t$$

through the maximum likelihood method.

▶ **Format**

{bmv,f0,grad,cov,retcode,media,var}=mv_inhomogeneous_ornstein(b0,s,d,y);

▶ **Input**

b0:

s: The initial value of time t_0

d:

y:

▶ **Output**

b: $\left(\hat{\beta}, \hat{\alpha}, \hat{\sigma}\right)$

f0:

gra:

cov:

retcode:

▶ **Library**

library cml,est_ede;

▶ **Source**

c:\gauss\srcnic\est_ede\mv_inhomogeneous.src

7.15 mv_mbg

▶ **Objectivo**

Estimar a EDE $dX_t = \alpha X_t dt + \beta X_t dW_t$ através do método da máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode}=mv_mbg(b0,d,y);

▶ **Input**

b0: valor inicial para o vector (α, β).

d: Δ, intervalo entre duas observações consecutivas, constante.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas para (α, β).

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\mv_mbg.src

7.16 mv_ornstein

▶ Purpose

Estimates the SDE $dX_t = \beta(\tau - X_t)\,dt + \sigma dW_t$ using the ML method.

▶ Format

{b,f0,grad,cov,retcode,media,var}=mv_ornstein(b0,d,y);

▶ Input

b0: starting values (β, τ, σ).

d: Δ interval between two consecutive observations (constant)

y: observations

▶ Output

b: estimates of (β, τ, σ).

f0: average likelihood function

gra: *score*

cov: Cov (b)

retcode: convergence. If retcode = 0 convergence is reached.

media: vector of the conditional mean

var:

▶ Library

library cml,est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\mv_ornstein.src

7.17 mv_simulado_ede1

▶ Purposes

Estimates a SDE

$$dX_t = a(X_t;\theta)\,dt + \sigma dW_t$$

through a simulated based method. The optimization problem **is**

$$\max_{\theta} \frac{1}{n}\sum_{i}^{n} \log\left[\frac{1}{n_s}\sum_{j=1}^{n_s}\hat{p}(\Delta, x, y)\right].$$

▶ **Format**

{b,f0,grad,cov,retcode}=mv_simulado_ede1(&drift,&derivada,&funcao_G,b0,rest,d,y,S,delta);

▶ **Input**

&f1: pointer to a procedures that defines $a(x;\theta)$. Note: the output is a vector (has the same number of rows as that of x).

&f2: pointer to a procedures that defines $da(x;\theta)/dx$.Note: the output is a vector (has the same number of rows as that of x).

&f3: pointer to a procedures that defines $G(y)$. Note: the output is a vector (has the same number of rows as that of x).

b0: starting values of $\left[\theta\,\sigma\right]'$.

rest: vector of ones and zeros of the same dimension of $\left[\theta\,\sigma\right]'$. The zeros indicate a nonnegative restrictions; the ones indicate no restriction.

d: Δ,interval between observations (constant

y: observations of the process

S: scalar n_s.

delta: scalar between 0 and 1 related to the precision of the numerical integration $\int_0^1 g\left(z_u(x,y)+\sqrt{\sigma^2\Delta}B_u(\omega)\right)du$. The lower the delta, the higher the precision (you may start with delta = .001).

▶ **Output**

b: estimates of $\left[\theta\,\sigma\right]'$

f0: mean of the likelihood function

gra: *score*

cov: Cov (b)

retcode: convergence. If retcode = 0 convergence is reached.

▶ **Library**

library cml,est_ede;

▶ **Example**

Take $dX_t=\beta(\tau-X_t)\,dt+\sigma dW_t$ and $(\beta,\tau,\sigma)=(b[1],b[2],b[3])$.

```
library cml,est_ede;

proc drift(b,x);
```

```
    retp(b[1]*(b[2]-x));
endp;
proc derivada(b,x);
    retp(-b[1]);
endp;
proc funcao_G(b,x);
    retp(b[1]*b[2]*x-b[1]*x^2/2);
endp;
{b,f0,grad,cov,retcode}=mv_simulado_ede1(&drift,&derivada,&funcao_G,1|1|1,1|1|1,.1,y,5,.001);
```

► **Source**

c:\gauss\srcnic\est_ede\mv_simu.src

7.18 mv_simulado_ede2

► **Objectivo**

Estimates θ in the SDE

$$dX_t = a\left(t, X_t; \theta\right) dt + b\left(t, X_t; \theta\right) dW_t$$

using the simulated maximum likelihood method. The optimization problem is

$$\max_{\theta} \frac{1}{n} \sum_{i}^{n} \log \left[\frac{1}{n_s} \sum_{j=1}^{n_s} \hat{p}\left(s, x, t, y\right) \right]$$

where $\hat{p}\left(s, x, t, y\right)$ is estimated using the routine:

```
p=est_dens_inhomogeneous(&sigma,&f,&A1,&B1,s,x,t,y,b,NN,simula,seed);
```

► **Formato**

```
{b,f0,grad,cov,retcode}= mv_simulado_ede2(&sigma,&f,&A1,&B1,s,d,y,b0,NN,simula,rest);
```

► **Input**

s: initial value of t

► **Output**

► **Library**

```
library cml,est_ede;
```

▶ **Exemplo**

▶ **Fonte**

c:\gauss\srcnic\est_ede\mv_simu.src

7.19 mvs_pederson

▶ **Objectivo**

Estimar a EDE com coeficiente de difusão constante,

$$dX_t = a\left(X_t;\theta\right)dt + b\left(X_t;\theta\right)dW_t$$

através do método da máxima verosimilhança simulada de acordo com Pedersen(1995). O problema de optimização é

$$\max_{\theta}\frac{1}{n}\sum_{i}^{n}\log\left[\frac{1}{n_s}\sum_{j=1}^{n_s}\hat{p}\left(\Delta,X_{t_{i-1}},X_{t_i};\theta\right)\right]$$

onde

$$\hat{p}\left(\Delta,x,y;\theta\right)=\frac{1}{S}\sum_{s=1}^{S}h\left(\frac{\Delta}{N},Y^{(N)}_{(N-1)\Delta/N,s},y;\theta\right).$$

▶ **Formato**

{b,f0,grad,cov,retcode}=mvs_pederson(&f1,&f2,&f3,seed,par1,rest1,par2,rest2,y,d,nn,ns);

▶ **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a\left(x;\theta\right)$. Nota: o output é um vector com o mesmo número de linhas de x.

&f2: ponteiro para um procedimento que define a especificação funcional de $b\left(x;\theta\right)$.Nota: o output é um vector com o mesmo número de linhas de x.

&f3: ponteiro para um procedimento que define o esquema de simulação. Existem dois procedimentos disponíveis: &simula1 e &simulaSM1. Ambos baseiam-se no esquema de Euler com uma precisão associada de Δ/N. No segundo assume-se $a\left(x;\theta\right)=0$. No caso &f3=0 o procedimento de simulação é &simula1 e se simulataneamente &f1=0 o procedimento é &simulaSM1.

seed: semente.

par1: valor inicial para o vector de parâmetros definidos em $a\left(x;\theta\right)$

rest1: vector de zeros e uns de dimensão igual à do vector θ definido em $a\left(x;\theta\right)$. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

par2: valor inicial para o vector de parâmetros definidos em $b\left(x;\theta\right)$

rest2: vector de zeros e uns de dimensão igual à do vector θ definido em $b(x;\theta)$. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

nn: valor de N.

ns: valor de n_s.

▶ Output

b: vector das estimativas de θ.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

▶ Library

library cml,est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\mvs_pede.src

7.20 pmv_cir

▶ Objectivo

Estimar a EDE $dX_t = \beta(\tau - X_t)\,dt + \sigma\sqrt{X_t}dW_t$ através do método da pseudo máxima verosimilhança.

▶ Formato

{b,f0,grad,cov,retcode,media,var}=pmv_cir(b0,d,y);

▶ Input

b0: valor inicial de (τ, β, σ).

d: Δ, intervalo entre duas observações consecutivas, constante.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas de (τ, β, σ).

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_ede;

▶ **Observações**

O *score* é escrito analiticamente.

▶ **Fonte**

c:\gauss\srcnic\est_ede\pmv_cir.src

7.21 pmv_mom

▶ **Objectivo**

Estimar uma EDE através do método de pseudo máxima verosimilhança.

▶ **Formato**

{beta,f0,grad,cov,retcode,media,var}=pmv_mom(&f,b0,rest,y,d);

▶ **Input**

&f: ponteiro para um procedimento que define a especificação da média e variância condicional.

b0: valor inicial para o vector θ

rest: vector de zeros e uns de dimensão igual à do vector θ. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

► **Output**

beta: vector das estimativas θ.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

► **Library**

library est_ede, cml;

► **Exemplo**

(1) Estimação dos parâmetros da EDE $dX_t = \theta X_t dt + dW_t$ através do estimador baseado numa discretização de ordem 1.5 (ver exemplo **??**, p. **??**) (nota: os parâmetros desta EDE devem ser estimados pelo método da máxima verosimilhança - serve este exemplo apenas para ilucidar a aplicação do procedimento). A média condicional aproximada é $X_{t_{i-1}} + \theta X_{t_{i-1}}\Delta + \theta^2 X_{t_{i-1}}\frac{\Delta^2}{2}$. O programa a considerar é

```
library est_ede,cml; /* localização da rotinas */

...

/* ler o vector das observações y */

...

proc (2)=momentos(b,x);

   media=x + d*b[1]*x +d^2/2*b[1]^2*x;      /* em lugar de b[1] pode-se escrever,
simplesmente, b */

   var=d;

   retp(media,var);

endp;

...

/* introduzir o valor inicial para b (guess), por exemplo: */

b=1;
```

rest = 0; /* impomos $\theta \geq 0$ */

d = 0.05;

{beta,f0,grad,cov,retcode,media,var}=pmv_mom(&momentos,b0,rest,y,d);

(2) Estimação dos parâmetros da EDE

$$dX_t = b_1 (b_2 - X_t)\, dt + b_3 X_t^{b_4} dW_t.$$

Os momentos condicionais são aproximados através de uma expansão de Ito-Taylor.

```
library est_ede,cml; /* localização da rotinas */

...

ler o vector das observações y

...

proc (2)=momentos(b,x);

   media=x + d*b[1]*(-x + b[2]) - (d^2*b[1]^2*(-x + b[2]))/2 + (d^3*b[1]^3*(-x
+ b[2]))/6 -

         (d^4*b[1]^4*(-x + b[2]))/24 + (d^5*b[1]^5*(-x + b[2]))/120;

   var=d*x^(2*b[4])*b[3]^2 - d^2*x^(2*b[4])*b[1]*b[3]^2 +

         (2*d^3*x^(2*b[4])*b[1]^2*b[3]^2)/3 - (d^4*x^(2*b[4])*b[1]^3*b[3]^2)/3 +

         (2*d^5*x^(2*b[4])*b[1]^4*b[3]^2)/15;

   retp(media,var);

endp;

...

/* introduzir o valor inicial para b (guess), por exemplo: */

b0 = {5,100,1,.3};

rest = {1,1,0,0}; /* impomos b3 e b4 maiores ou iguais a zero */

d = 0.05;

{beta,f0,grad,cov,retcode,media,var}=pmv_mom(&momentos,b0,rest,y,d);
```

Nota Final: O procedimento está construído para que "x" definido no "ponteiro" "proc (2)=momentos(b,x);" seja um vector. Se na média ou var definido em "proc (2)=momentos(b,x);" existir um produto do tipo $g(x)x$ onde $g(x)$ e x são vectores é necessário escrever g(x).*x e não g(x)*x. Neste último caso o programa GAUSS tentaria, sem êxito, realizar um produto matricial quando na verdade o produto deve ser entre elementos homólogos.

▶ **Fonte**

c:\gauss\srcnic\est_ede\pmv_mom.src

7.22 pmv_s

▶ **Objectivo**

Estimar uma EDE através do método da pseudo máxima verosimilhança simulada [a média e a variância condicional são estimados através de simulações do processo - ver equação (**??**)]

▶ **Formato**

{b,f0,g,cov,retcode}=pmv_s(&f1,&f2,par1,rest1,par2,rest2,y,d,nn,ns);

▶ **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

par1: valor inicial para o vector θ_1.

res1: vector de zeros e uns de dimensão igual à do vector θ_1. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

par2: valor inicial para o vector θ_2.

res2: vector de zeros e uns de dimensão igual à do vector θ_2. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

nn: valor do parâmetro N [ver equação (**??**)].

ns: valor do parâmetro n_s.

▶ **Output**

b: vector das estimativas de θ_1 e θ_2.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\pmv_s.src

7.23 pmv_vol_quad

▶ **Objectivo**

Estimar a EDE $dX_t = \beta(\tau - X_t)\,dt + \sqrt{\sigma + \lambda(X_t - \mu)^2}dW_t$ através do método da pseudo máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode,media,var}=pmv_vol_quad(b0,d,y);

▶ **Input**

b0: valor inicial de $(\beta, \tau, \sigma, \lambda, \mu)$.

d: Δ, intervalo entre duas observações consecutivas, constante.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas de $(\beta, \tau, \sigma, \lambda, \mu)$.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\pmv_vq.src

7.24 pmvEuler

▶ Objectivo

Estimar uma EDE através do método de máxima verosimilhança da equação discretizada de acordo com o esquema de Euler (pode também ser considerado um método de pseudo máxima verosimilhança).

▶ Formato

{b,f0,grad,cov,retcode,media,var} = pmvEuler(&f1,&f2,par1,res1,par2,res2,y,d);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$ (se &f1 = 0 o drift é nulo).

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

par1: valor inicial para o vector θ_1.

res1: vector de zeros e uns de dimensão igual à do vector θ_1. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

par2: valor inicial para o vector θ_2.

res2: vector de zeros e uns de dimensão igual à do vector θ_2. Os zeros indicam que deve ser considerada a restrição de não negatividade; os uns indicam ausência de restrição.

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

▶ Output

b: vector das estimativas de θ_1 e θ_2

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

▶ **Library**

library est_ede, cml;

▶ **Exemplo**

Estimação dos parâmetros da EDE $dX_t = (\alpha_1 + \alpha_2 X_t)\, dt + \sqrt{\beta_1 + \beta_2 (X_t - \beta_3)^2}\, dW_t$.

library pmveuler,cml; /* localização da rotinas */

...

ler o vector das observações y

...

par1 = {10,-0.1};

res1 = {1,1}; /* α_1 e α_2 não estão sujeitos a restrições /*

par2 = {0.1,0.1,100};

res2 = {0,0,1}; /* β_1 e β_2 estão sujeitos a restrições de não negatividade /*

d = 0.05;

{b,f0,grad,cov,retcode,media,var} = pmveuler(&linear,&quadratica,par1,res1,par2,res2,y,d);

Nota: os "ponteiros" estão já definidos (são "conhecidos" pelo GAUSS) - ver FUNÇÕES no apêndice **??**.

▶ **Fonte**

c:\gauss\srcnic\est_ede\pmvEuler.src

7.25 rs_tc_1

▶ **Objectivo**

Estimar o modelo com alterações de regime, $dX_t = a(t, X_t)\, dt + b(t, X_t)\, dW_t$ onde $a(t,x) = \beta_1 (\tau_1 - X_t) I_1(t) + \beta_2 (\tau_2 - X_t) I_2(t)$, $b(t,x) = \sigma_1 I_1(t) + \sigma_2 I_2(t)$, $I_1(t) = 1$ se em t, $S = 1$ (zeros noutros casos) e $I_2(t) = 1 - I_1(t)$. $S_i = S_{i\Delta}$ é uma cadeia de Markov homogénea em tempo discreto, $i \in \{1, 2, ..., n\}$, com espaço de estados $\{1, 2\}$ e com probabilidades de transição

$$P = \begin{bmatrix} \alpha_{11} & 1 - \alpha_{11} \\ 1 - \alpha_{22} & \alpha_{22} \end{bmatrix}.$$

O modelo é estimado através do procedimento de Hamilton (1994) de acordo com as hipóteses A e B2 do ponto **??**.

▶ **Formato**

{b,cov,m1,m2,media,var,prob1}=rs_tc_1(y,d,b0);

▶ **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P\left[S_1 = 1 | X_{t_1}\right]$ (começar por considerar $p_0 = 0.5$).

▶ Output

b: vector das estimativas de $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional do regime 1 $(S = 1)$ de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 $(S = 2)$ de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1 | X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ $(i = 2, ..., n$ onde n é o número de observações).

▶ Library

library cml,est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\rs_tc.src

7.26 rs_tc_1a

▶ Objectivo

Igual a RS_TC1 com a diferença de que agora $\beta_1 = \tau_1 = 0$ (o regime 1 é um processo de Wiener).

▶ Formato

{b,f0,cov,f1,f2,m1,m2,media,var,prob1,retcode}=rs_tc_1a(y,d,b0);

▶ Input

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P[S_1 = 1|\, X_{t_1}]$ (começar por considerar $p_0 = 0.5$).

▶ **Output**

b: vector das estimativas de $(\sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

f1: vector de tipo $n \times 1$ densidade associada ao regime 1

f2: vector de tipo $n \times 1$ densidade associada ao regime 2

m1: vector da média condicional do regime 1 $(S = 1)$ de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 $(S = 2)$ de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1|\, X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ $(i = 2, ..., n$ onde n é o número de observações).

retcode:

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\rs_tc.src

7.27 rs_tc_2

▶ **Objectivo**

Estimar o modelo com alterações de regime, $dX_t = a(t, X_t)\, dt + b(t, X_t)\, dW_t$ onde $a(t,x) = \beta_1(\tau_1 - X_t) I_1(t) + \beta_2(\tau_2 - X_t) I_2(t)$, $b(t,x) = \sigma_1\sqrt{X_t} I_1(t) + \sigma_2\sqrt{X_t} I_2(t)$ (ver notações em RS_TC_1).

▶ **Formato**

{b,cov,m1,m2,media,var,prob1}=rs_tc_2(r,d,b0);

▶ **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P[S_1 = 1|\, X_{t_1}]$ (começar por considerar $p_0 = 0.5$).

▶ Output

b: vector das estimativas de $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional do regime 1 ($S = 1$) de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 ($S = 2$) de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1|\, X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ ($i = 2, ..., n$ onde n é o número de observações).

▶ Library

library cml,est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\rs_tc.src

7.28 rs_tc_3

▶ Objectivo

Estimar o modelo com alterações de regime, $dX_t = a(t, X_t)\, dt + b(t, X_t)\, dW_t$ onde $a(t,x) = \beta_1(\tau_1 - X_t) I_1(t)$, $b(t,x) = \sigma_1\sqrt{X_t} I_1(t) + \sigma_2\sqrt{X_t} I_2(t)$ (ver notações em RS_TC_1).

▶ Formato

{b,cov,m1,m2,media,var,prob1}=rs_tc_3(y,d,b0);

▶ Input

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\tau_1, \beta_1, \sigma_1, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P[S_1 = 1|\, X_{t_1}]$ (começar por considerar $p_0 = 0.5$).

▶ **Output**

b: vector das estimativas de $(\tau_1, \beta_1, \sigma_1, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional do regime 1 ($S = 1$) de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 ($S = 2$) de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1 | X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ ($i = 2, ..., n$ onde n é o número de observações).

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\rs_tc.src

7.29 rs_tc_4

▶ **Objectivo**

Estimar o modelo com alterações de regime, $dX_t = a(t, X_t)\, dt + b(t, X_t)\, dW_t$ onde $a(t,x) = \beta_1(\tau_1 - X_t) I_1(t)$ e $b(t,x) = \sigma_1 X_t I_1(t) + \sigma_2 \sqrt{X_t} I_2(t)$ (ver notações em RS_TC_1).

▶ **Formato**

{b,cov,m1,m2,media,var,prob1}=rs_tc_4(y,d,b0);

▶ **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\tau_1, \beta_1, \sigma_1, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P[S_1 = 1 | X_{t_1}]$ (começar por considerar $p_0 = 0.5$).

▶ **Output**

b: vector das estimativas de $(\tau_1, \beta_1, \sigma_1, \sigma_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional do regime 1 ($S = 1$) de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 ($S = 2$) de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1 | X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ ($i = 2, ..., n$ onde n é o número de observações).

▶ Library

library cml,est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\rs_tc.src

7.30 rs_tc_5

▶ Objectivo

Estimar o modelo com alterações de regime, $dX_t = a(t, X_t)\, dt + b(t, X_t)\, dW_t$ onde $a(t,x) = \beta_1(\tau_1 - x) I_1(t) + \beta_2(\tau_2 - x) I_2(t)$, $b(t,x) = \sqrt{\sigma_1^2 + \lambda_1 (x - \mu_1)^2} I_1(t) + \sqrt{\sigma_2^2 + \lambda_2 (x - \mu_2)^2} I_2(t)$ (ver notações em RS_TC_1).

▶ Formato

{b,cov,m1,m2,media,var,prob1}=rs_tc_5(y,d,b0);

▶ Input

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas (pode ser um vector).

b0: valor inicial para o vector dos parâmetros $(\beta_1, \tau_1, \sigma_1, \lambda_1, \mu_1, \beta_2, \tau_2, \sigma_2, \lambda_2, \mu_2, \alpha_{11}, \alpha_{22}, p_0)$, onde $p_0 = P\left[S_1 = 1 | X_{t_1}\right]$ (começar por considerar $p_0 = 0.5$).

▶ Output

b: vector das estimativas de $(\beta_1, \tau_1, \sigma_1, \lambda_1, \mu_1, \beta_2, \tau_2, \sigma_2, \lambda_2, \mu_2, \alpha_{11}, \alpha_{22}, p_0)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional do regime 1 ($S = 1$) de tipo $(n-1) \times 1$ onde n é o número de observações.

m2: vector da média condicional do regime 2 ($S = 2$) de tipo $(n-1) \times 1$ onde n é o número de observações.

media: vector da média condicional do processo de tipo $(n-1) \times 1$.

var: vector da variância condicional do processo de tipo $(n-1) \times 1$.

prob1: vector das probabilidades $P\left[S_{i-1} = 1 | X_{t_{i-1}}\right]$ estimadas de tipo $(n-1) \times 1$ ($i = 2, ..., n$ onde n é o número de observações).

▶ **Library**

library cml,est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\rs_tc.src

7.31 var_kernel1

▶ **Objectivo**

Estimar não parametricamente o coeficiente $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$S_n(x) = \frac{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i} - x| < h\}} \frac{\left(X_{t_{i+1}} - X_{t_i}\right)^2}{\Delta}}{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i} - \xi| < h\}}}.$$

▶ **Formato**

{x,v,vary,k,liminf,limsup} = var_kernel1(y,d,pontos,h,interpola,controle);

▶ **Input**

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

d: Δ, intervalo entre duas observações consecutivas, constante.

pontos: número de pontos x_i para os quais se calcula $S_n(x_i)$. Isto é $i = 1, 2, ...,$pontos.

h: bandwith

interpola: assume $\{0, 1\}$. Se interpola = 1 calcula, por interpolação, $S_n(x_i)$, onde x_i assume todos os valores observados do processo X.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $b^2(x)$ apenas quando

$$\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i} - x| < h\}} \geq \text{controle}.$$

Aconselha-se um valor entre 10 e 20.

▶ **Output**

x: vector x de tipo $k \times 1$ onde $k \leq$ pontos.

v: vector $S_n(x)$ de tipo $k \times 1$

vary: igual a zero se interpola = 0 e igual a $S_n(x_i)$ de tipo $n \times 1$ onde n é o número de observações de X se interpola = 1.

k: vector $\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}$ de tipo $k \times 1$.

liminf: vector de tipo $k \times 1$ que representa o limite inferior do intervalo de confiança a 95% para $b^2(x)$.

limsup: vector de tipo $k \times 1$ que representa o limite superior do intervalo de confiança a 95% para $b^2(x)$.

▶ **Library**

library est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.32 var_kernel2

▶ **Objectivo**

Estimar não parametricamente o coeficiente $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$V_n(\xi) = \frac{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i}-x|<h\}} \frac{\left(X_{t_{i+1}}-X_{t_i}-a\left(X_{t_i}\right)\right)^2}{\Delta}}{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i}-x|<h\}}}.$$

onde se admite que a é do tipo $a(x;\theta) = \theta_1 + \theta_2 x$.

▶ **Formato**

{x,v,vary,k,liminf,limsup} = var_kernel2(y,d,pontos,h,interpola,controle);

▶ **Input**

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

d: Δ, intervalo entre duas observações consecutivas, constante.

pontos: número de pontos ξ_i. Isto é $i = 1, 2, ...,$pontos.

interpola: assume $\{0, 1\}$. Se interpola = 1 calcula, por interpolação, $b^2(x_i)$, onde x_i assume todos os valores observados do processo X.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $b^2(x_i)$ apenas quando

$$\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}} \geq \text{controle}.$$

Aconselha-se um valor entre 10 e 20.

▶ Output

x: vector x de tipo $k \times 1$ onde $k \leq$ pontos.

v: vector $V_n(x)$ de tipo $k \times 1$

vary: igual a zero se interpola = 0 e igual a $V_n(x_i)$ de tipo $n \times 1$ onde n é o número de observações de X se interpola = 1.

k: vector $\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}$ de tipo $k \times 1$.

liminf: vector de tipo $k \times 1$ que representa o limite inferior do intervalo de confiança a 95% para $b^2(x)$.

limsup: vector de tipo $k \times 1$ que representa o limite superior do intervalo de confiança a 95% para $b^2(x)$.

▶ Library

library est_ede;

▶ Observações

Os parâmetros do coeficiente a são estimados pelo método dos mínimos quadrados.

▶ Fonte

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.33 var_kernel3

▶ Objectivo

Estimar não parametricamente o coeficiente $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$v(x) = \frac{\sum_{i=1}^{n-1} K\left(\frac{X_{t_i}-x}{h}\right) \frac{\left(X_{t_{i+1}}-X_{t_i}\right)^2}{\Delta}}{\sum_{i=1}^{n-1} K\left(\frac{X_{t_i}-x}{h}\right)}$$

onde $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$.

▶ Formato

{x,v,liminf,limsup}=var_kernel3(y,pontos,ic,d);

▶ Input

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

pontos: número de pontos x_i para os quais se calcula $v(x_i)$. Isto é $i = 1, 2, ...,$pontos. Se pontos $= 0$, calcula-se $v(x_i)$ onde x_i assume todos os valores da amostra $(X_{t_1}, ..., X_{t_n})$ que satisfazem o parâmetro ic (ver a seguir).

ic: assume um valor no intervalo $]0, 1]$. Se ic$\in]0, 1[$ o procedimento é apenas executado para os valores $(X_{t_1}, ..., X_{t_n})$ que se encontram entre o percentil $(1 - ic)/2$ e o percentil $(1 + ic)/2$. As estimativas de $b(x)$ são portanto aparadas.

d: Δ, intervalo entre duas observações consecutivas, constante.

▶ Output

x: vector x.

v: vector $v(x)$.

liminf: vector com dimensão igual à do vector x que representa o limite inferior do intervalo de confiança a 95% para $b^2(x)$.

limsup: vector com dimensão igual à do vector x que representa o limite superior do intervalo de confiança a 95% para $b^2(x)$.

▶ Library

library est_ede;

▶ Fonte

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.34 var_kernel4

▶ Objectivo

Estimar não parametricamente o coeficiente $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). O estimador é:

$$V_n^*(x) = \frac{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i} - x| < h\}} \frac{\left(X_{t_{i+1}} - X_{t_i} - T_n(X_{t_i})\Delta\right)^2}{\Delta}}{\sum_{i=1}^{n-1} \mathcal{I}_{\{|X_{t_i} - x| < h\}}}.$$

onde T_n é o estimador DRIFT_KERNEL1.

▶ Formato

{xM,med,medy,kM,liminfM,limsupM,xV,v,vary,kV,liminfV,limsupV} =

var_kernel4(y,d,pontosM,pontosV,h,interpola,controle);

▶ Input

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

d: Δ, intervalo entre duas observações consecutivas, constante.

pontosM: (referente a T_n) número de pontos x_i. Isto é $i = 1, 2, ...$,pontos. Aconselha-se um valor alto.

pontosV: (referente a V_n^*) número de pontos x_i. Isto é $i = 1, 2, ...$,pontos.

h: bandwith.

interpola: assume $\{0, 1\}$. Se interpola = 1 calcula, por interpolação, $b^2(x_i)$, onde x_i assume todos os valores observados do processo X.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $b^2(\xi_i)$ apenas quando

$$\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}} \geq \text{controle.}$$

Aconselha-se um valor entre 10 e 20.

▶ **Output**

xM: vector x de tipo $k \times 1$ onde $k \leq$ pontos.

med: vector $\hat{a}(x)$ de tipo $k \times 1$

medy: igual a zero se interpola = 0 e igual a $\hat{a}(x_i)$ de tipo $n \times 1$ onde n é o número de observação de X se interpola = 1.

kM: vector $\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}$ de tipo $kM \times 1$ (relativo ao estimador T_n).

lim_infM: vector de tipo pontos×1 referente ao limite inferior do intervalo de confiança a 95% para $\hat{a}(x_i)$.

lim_supM: vector de tipo pontos×1 referente ao limite superior do intervalo de confiança a 95% para $\hat{a}(x_i)$.

xV: vector x de tipo $kV \times 1$ onde $kV \leq$ pontosV.

v: vector $V_n^*(x)$ de tipo $k \times 1$

vary: igual a zero se interpola = 0 e igual a $V_n^*(x_i)$ de tipo $n \times 1$ onde n é o número de observações de X se interpola = 1.

kV: vector $\sum_{i=1}^{n} \mathcal{I}_{\{|X_{t_i}-x|<h\}}$ de tipo $kV \times 1$ (relativo ao estimador V_n^*).

liminfV: vector de tipo $kV \times 1$ que representa o limite inferior do intervalo de confiança a 95% para $b^2(x)$.

limsupV: vector de tipo $kV \times 1$ que representa o limite superior do intervalo de confiança a 95% para $b^2(x)$.

▶ **Library**

library est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.35 var_kernel5

▶ **Objectivo**

Estimar não parametricamente os coeficientes $a(x)$ e $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). Os estimador são:

$$T_n(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}\right)}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)},\ S_n(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}\right)^2}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)}$$

onde K é o kernel gaussiano.

▶ **Formato**

{x,B,drift,ic_drift,diffusion,ic_diffusion}=var_kernel5(y,xi,pontos,d,ic,controle);

▶ **Input**

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

xi: vector $x = (x_1, ...)$ em relação aos quais se calculam as estimativas para $a(x_1)$, $a(x_2), ...$ e $b^2(x_1)$, $b^2(x_2), ...$ Seja rows(x)=p. Se p>1 então pontos=p (qualquer que seja o valor atribuído a pontos) e x (output) vem igual a xi. Se p=1 o procedimento calcula internamente xi de acordo com o valor atribuído em "pontos". Neste caso o output x resulta do procedimento. A variável controle é posta em zero.

pontos: número de pontos x_i. Isto é $i = 1, 2, ...,$pontos. (ver xi)

d: Δ, intervalo entre duas observações consecutivas, constante.

ic: nível de confiança do intervalo de confiança.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $b^2(\xi_i)$ apenas quando

$$\sum_i K\left(\frac{X_{t_i}-x}{h}\right) > \text{ controle.}$$

(ver xi)

▶ **Output**

x: vector x de tipo $k \times 1$ onde $k \leq$ pontos. (ver xi em input)

B: vector $B_n(x) = \sum_i K\left(\frac{X_{t_i}-x}{h}\right)$ de tipo $k \times 1$ (estimativa da fdp estacionária).

drift:

ic_drift:

ic_diffusion:

▶ Library

library est_ede;

▶ Observações

A bandwith é definida como h=_factor_h*(4/3)^.2*sigmay*(n-1)^(-1/5). Por defeito, _factor_h=1. Esta variável é global.

▶ Fonte

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.36 var_kernel6

▶ Purpose

Estimates $a(x)$ and $b^2(x)$ (in the SDE $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). The estimators for $a(x)$ and $b^2(x)$ are respectively

$$T_n(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}\right)}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)},\ V_n^*(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}-T_n(x)\Delta\right)^2}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)}$$

where K is the Gaussian kernel.

▶ Format

{x,B,drift,ic_drift,diffusion,ic_diffusion}=var_kernel6(y,xi,points,d,ic,control);

▶ Input

y: vector of observation $(X_{t_1}, ..., X_{t_n})$

xi: vector or scalar. Vector $x = (x_1, ...)$ from which the estimates $a(x_1)$, $a(x_2)$, ... and $b^2(x_1)$, $b^2(x_2)$, ... are calculated. If xi is a scalar (regardless its value) the procedure calculates x_i (using seqa function) according to the value defined in points. In this case "control" takes on the value zero.

points: scalar, number of points x_i.

d: scalar, Δ, interval between two consecutive observations

ci: confidence interval (e.g. 0.95)

controle: scalar. The procedure estimates $b^2(\xi_i)$ if

$$\sum_i K\left(\frac{X_{t_i}-x}{h}\right) > \text{ control.}$$

▶ **Output**

x: $k \times 1$ vector x_i ($k \leq$ points).

B: $k \times 1$ vector $B_n(x) = \sum_i K\left(\frac{X_{t_i}-x}{h}\right)$

drift:

ic_drift:

ic_diffusion:

▶ **Library**

library est_ede;

▶ **Remark**

The bandwidth is defined as h=_factor_h*(4/3)^.2*sigmay*(n-1)^(-1/5). By default _factor_h=1. This variable is global.

▶ **Source**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.37 var_kernel7

▶ **Objectivo**

Estimar não parametricamente os coeficientes $a(x)$ e $b^2(x)$ (da equação $dX_t = a(X_t)\,dt + b(X_t)\,dW_t$). Os estimador são:

$$T_n(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}\right)}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)},\ V_n^{**}(x) = \frac{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)\frac{\left(X_{t_{i+1}}-X_{t_i}-a(x;\theta)\Delta\right)^2}{\Delta}}{\sum_i K\left(\frac{X_{t_i}-x}{h}\right)}$$

onde K é o kernel gaussiano.

▶ **Formato**

{x,B,drift,driftV(par1,x),ic_drift,diffusion,ic_diffusion}=var_kernel7(&driftV,par1,y,pontos,d,ic,con

▶ **Input**

driftV: ponteiro para $a(x;\theta)$. Ver V_n^{**}.

par1: θ

y: vector de observações $(X_{t_1}, ..., X_{t_n})$ do processo X.

pontos: número de pontos x_i. Isto é $i = 1, 2, ...,$pontos.

d: Δ, intervalo entre duas observações consecutivas, constante.

ic: nível de confiança do intervalo de confiança.

controle: assume $\{1, 2, ...\}$. O procedimento calcula estimativas para $b^2(\xi_i)$ apenas quando

$$\sum_i K\left(\frac{X_{t_i} - x}{h}\right) > \text{ controle.}$$

▶ **Output**

x: vector x de tipo $k \times 1$ onde $k \leq$ pontos.

B: vector $B_n(x) = \sum_i K\left(\frac{X_{t_i}-x}{h}\right)$ de tipo $k \times 1$ (estimativa da fdp estacionária).

drift: vector $T_n(x)$ de tipo $k \times 1$ onde $k \leq$ pontos.

driftV: vector $a(x; \theta)$ de tipo $k \times 1$ onde $k \leq$ pontos.

ic_drift:

ic_diffusion:

▶ **Library**

library est_ede;

▶ **Observações**

A bandwith é definida como h=_factor_h*(4/3)^.2*sigmay*(n-1)^(-1/5). Por defeito, _factor_h=1 Esta variável é global.

▶ **Fonte**

c:\gauss\srcnic\est_ede\nao_p_tc.src

7.38 vqe_tc1

▶ **Objectivo**

Estimar o modelo

$$dX_t = \sqrt{\alpha_0^2 + \alpha_1 (X_t - \mu_t)^2} dW_t,$$
$$d\mu_t = \lambda (X_t - \mu_t) dt + \sigma dW_{t,2}$$

através do problema de optimização $\min_\theta Q_n(\theta)$ onde $\theta = (\alpha_0, \alpha_1, \lambda, \sigma, \mu_0)$,

$$Q_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\log \overline{Var}\left[X_{t_i}|\, X_{t_{i-1}}, \tilde{\mu}; \theta\right] + \frac{1}{2}\frac{\left(X_{t_i} - X_{t_{i-1}}\right)^2}{\overline{Var}\left[X_{t_i}|\, X_{t_{i-1}}, \tilde{\mu}; \theta\right]}\right),$$

$$\overline{Var}\left[X_{t_i}|\, X_{t_{i-1}}, \tilde{\mu}; \theta\right] = \frac{1}{S}\sum_{h=1}^{S} Var\left[X_{t_i}|\, X_{t_{i-1}}, \mu^h_{t_{i-1}}; \theta\right], \tag{a}$$

$$Var\left[X_{t_i}|\, X_{t_{i-1}}; \theta\right] = \frac{2\alpha_0^2\lambda + \alpha_1\sigma^2}{2\lambda - a_1}\Delta + \frac{\alpha_1\left(a_0^2 + \sigma^2\right)\left(-1 + e^{(\alpha_1 - 2\lambda)\Delta}\right)}{\left(2\lambda - a_1\right)^2} + \frac{\alpha_1\left(1 - e^{(\alpha_1 - 2\lambda)\Delta}\right)}{2\lambda - \alpha_1}\left(X_{t_{i-1}} - \mu^h_{t_{i-1}}\right)^2.$$

Dado $\mu_{t_0} = \mu_0$, o processo μ_t pode ser simulado por uma das seguintes hipóteses

$$\mu^h_{t_i} = \mu^h_{t_{i-1}} + \lambda\left(X_{t_{i-1}} - \mu^h_{t_{i-1}}\right)\Delta + \sigma\sqrt{\Delta}\varepsilon_{t_i}, \tag{i}$$

$$\mu^h_{t_i} = e^{-\lambda\Delta}\lambda\left(\frac{X_{t_i}e^{\lambda\Delta} + X_{t_{i-1}}}{2}\right)\Delta + e^{-\lambda\Delta}\mu^h_{t_{i-1}} + \sqrt{\frac{\sigma^2}{2\lambda}\left(1 - e^{-2\lambda\Delta}\right)}\varepsilon_{t_i}, \tag{ii}$$

$$\mu^h_{t_i} = \left(1 - e^{-\lambda\Delta}\right)X_{t_{i-1}} + e^{-\lambda\Delta}\mu^h_{t_{i-1}} + \sqrt{\frac{\sigma^2}{2\lambda}\left(1 - e^{-2\lambda\Delta}\right)}\varepsilon_{t_i} \tag{iii}$$

$\varepsilon_{t_i} \sim N(0,1)$.

► **Formato**

{b,f0,grad,cov,retcode,miu,media,var}=vqe_tc1(y,d,aprox,s,seed,b0);

► **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

aprox: Define a aproximação a considerar para μ_t, (i), (ii) ou (iii), respectivamente, aprox=1, aprox=2, aprox=3.

s: número de simulações S.

b0: vector dos parâmetros iniciais.

► **Output**

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

miu: vector das estimativas de μ_t.

media: vector da média condicional.

var: vector da variância condicional obtida de acordo com (a).

▶ **Library**

library cml, est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_etd\vqe_tc.src

7.39 vqe_tc2

▶ **Objectivo**

Estimar o modelo

$$dX_t = \sqrt{\alpha_0^2 + \alpha_1 (X_t - \mu_t)^2} dW_t$$
$$d\mu_t = \lambda (X_t - \mu_t) dt,$$

através do problema de optimização $\min_\theta Q_n(\theta)$ onde $\theta = (\alpha_0, \alpha_1, \lambda, \mu_0)$,

$$Q_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\log Var\left[X_{t_i}|\, X_{t_{i-1}}, \mu_{t_{i-1}}; \theta\right] + \frac{1}{2}\frac{\left(X_{t_i} - X_{t_{i-1}}\right)^2}{Var\left[X_{t_i}|\, X_{t_{i-1}}, \mu_{t_{i-1}}; \theta\right]}\right),$$

$$Var\left[X_{t_i}|\, X_{t_{i-1}}; \theta\right] = \frac{2\alpha_0^2\lambda + \alpha_1\sigma^2}{2\lambda - a_1}\Delta + \frac{\alpha_1\left(a_0^2 + \sigma^2\right)\left(-1 + e^{(\alpha_1 - 2\lambda)\Delta}\right)}{(2\lambda - a_1)^2} + \frac{\alpha_1\left(1 - e^{(\alpha_1 - 2\lambda)\Delta}\right)}{2\lambda - \alpha_1}\left(X_{t_{i-1}} - \mu_{t_{i-1}}\right)^2.$$

Dado $\mu_{t_0} = \mu_0$, o processo μ_t é aproximado por uma das seguintes hipóteses

$$\mu_{t_i} = \mu_{t_{i-1}} + \lambda\left(X_{t_{i-1}} - \mu_{t_{i-1}}\right)\Delta, \tag{i}$$

$$\mu_{t_i} = e^{-\lambda\Delta}\lambda\left(\frac{X_{t_i}e^{\lambda\Delta} + X_{t_{i-1}}}{2}\right)\Delta + e^{-\lambda\Delta}\mu_{t_{i-1}}, \tag{ii}$$

$$\mu_{t_i} = \left(1 - e^{-\lambda\Delta}\right)X_{t_{i-1}} + e^{-\lambda\Delta}\mu_{t_{i-1}}. \tag{iii}$$

▶ **Formato**

{b,f0,grad,cov,retcode,miu,media,var}=vqe_tc2(x,d,aprox,b0);

▶ **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

aprox: Define a aproximação a considerar para μ_t, (i), (ii) ou (iii), respectivamente, aprox=1, aprox=2, aprox=3.

b0: vector dos parâmetros iniciais.

▶ **Output**

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

miu: vector das estimativas de μ_t.

media: vector da média condicional.

var: vector da variância condicional.

▶ **Library**

library cml, est_ede;

▶ **Fonte**

c:\gauss\srcnic\est_etd\vqe_tc.src

7.40 vq_tc3

▶ **Objectivo**

Estimar o modelo

$$dX_t = (k_0 + k_1 X_t)\, dt + \sqrt{\alpha_0^2 + \alpha_1 (X_t - \mu_t)^2}\, dW_t$$

$$d\mu_t = \lambda (X_t - \mu_t)\, dt,$$

através do problema de optimização $\min_\theta Q_n(\theta)$ onde $\theta = (k_0, k_1, \alpha_0, \alpha_1, \lambda, \mu_0)$ e

$$Q_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \log v_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon \right) + \frac{1}{2} \frac{\left(X_{t_i} - m_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon \right) \right)^2}{v_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon \right)} \right)$$

sendo que $m_{t_i}\left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon\right) = m_{t_i}\left(\theta, X_{t_{i-1}}\right)$ é a média condicional exacta

$$m_{t_i}\left(\theta, X_{t_{i-1}}\right) = -\frac{k_0}{k_1} + \frac{e^{k_1\Delta}\left(k_0 + k_1 X_{t_{i-1}}\right)}{k_1}$$

e $v_{t_i}\left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon\right)$ é a variância condicional obtida por simulação. O estimador para $Var\left[X_\Delta \middle| X_0, \mu_0; \theta\right]$ (supondo conhecida a média condicional):

$$v_\Delta\left(\theta, X_0, \mu_0, \varepsilon\right) = \frac{1}{S}\sum_{k=1}^{S}\left(Y_{k,\Delta}^{(N)} - m_{t_i}\left(\theta, X_{t_{i-1}}\right)\right)^2$$

onde $Y_{k,\Delta}^{(N)} = Y_\Delta^{(N)}$ (k-ésima aproximação para o valor X_Δ) é obtido de acordo com o esquema de Euler (ver o ponto **??**; note-se que $\lim_{N\to+\infty} E\left[v_\Delta\left(\theta, X_0, \mu_0, \varepsilon\right) \middle| X_0, \mu_0\right] = Var\left[X_\Delta \middle| X_0, \mu_0; \theta\right]$).

► **Formato**

{b,f0,grad,cov,retcode,media,var}=vq_tc3(y,d,N,s,seed,b0);

► **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

N: valor maior ou igual a um.

s: Número de simulações S.

b0: vector dos parâmetros iniciais.

► **Output**

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional.

var: vector da variância condicional.

► **Library**

library cml, est_ede;

▶ **Observações**

O vector μ encontra-se em memória. Basta invocar miu.

▶ **Fonte**

c:\gauss\srcnic\est_etd\vqe_tc.src

7.41 vq_tc4

▶ **Objectivo**

Estimar o modelo

$$dX_t = \sqrt{\alpha_0^2 + \alpha_1 \left(X_t - \mu_t\right)^2} dW_t$$
$$d\mu_t = \lambda \left(X_t - \mu_t\right) dt,$$

através do problema de optimização $\min_\theta Q_n(\theta)$ onde $\theta = (\alpha_0, \alpha_1, \lambda, \mu_0)$ e

$$Q_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \log v_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon\right) + \frac{1}{2} \frac{\left(X_{t_i} - X_{t_{i-1}}\right)^2}{v_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon\right)} \right)$$

sendo $v_{t_i} \left(\theta, X_{t_{i-1}}, \mu_{t_{i-1}}, \varepsilon\right)$ a variância condicional obtida por simulação. O estimador para $Var\left[X_\Delta | X_0, \mu_0; \theta\right]$ (supondo conhecida a média condicional):

$$v_\Delta(\theta, X_0, \mu_0, \varepsilon) = \frac{1}{S} \sum_{k=1}^{S} \left(Y_{k,\Delta}^{(N)} - m_{t_i} \left(\theta, X_{t_{i-1}}\right) \right)^2$$

onde $Y_{k,\Delta}^{(N)} = Y_\Delta^{(N)}$ (k-ésima aproximação para o valor X_Δ) é obtido de acordo com o esquema de Euler (ver o ponto **??**; note-se que $\lim_{N \to +\infty} E\left[v_\Delta(\theta, X_0, \mu_0, \varepsilon) | X_0, \mu_0\right] = Var\left[X_\Delta | X_0, \mu_0; \theta\right]$).

▶ **Formato**

{b,f0,grad,cov,retcode,media,var}=vq_tc4(y,d,N,s,seed,b0);

▶ **Input**

y: vector das observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

N: valor maior ou igual a um.

s: Número de simulações S.

b0: vector dos parâmetros iniciais.

▶ **Output**

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional.

var: vector da variância condicional.

▶ Library

library cml, est_ede;

▶ Observações

O vector μ encontra-se em memória. Basta invocar miu.

▶ Fonte

c:\gauss\srcnic\est_etd\vqe_tc.src

8 Estimação de Equações em Tempo Discreto[est_etd]

8.1 empirical_CDF

▶ Objectivo

Dado $\{y_1, ..., y_n\}$ estima $\{F_n(y_1), ..., F_n(y_n)\}$ onde

$$\hat{\tau}_i \equiv F_n(y_i) = \frac{1}{n}\sum_{k=1}^{n} \mathcal{I}_{\{y_k \leq y_i\}}$$

▶ Formato

tau=empirical_CDF(y);

▶ Input

y: vector das observações do processo.

▶ Output

tau: $\{F_n(y_1), ..., F_n(y_n)\}$.

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\nao_p_td.src

8.2 estima_arma11_hetero_1

▶ Objectivo

Estima ARMA(1,1) com heterocedasticidade dependente da matriz X e com distribuição normal ou t-student

▶ Formato

{b,f0,grad,cov,retcode,media,var}=estima_arma11_hetero_1(b0,y,x,g,gl,p,q);

▶ Input

b0: se b0=0 os valores iniciais são calculados pelo procedimento

y:

x: matriz das variáveis explicativas na média

g: matriz das variáveis explicativas na variância condicional

gl: se gl=0 $\Rightarrow$ distribuição normal; se gl $>$ 0 $\Rightarrow$ distribuição tStudent e gl funciona como valor incial para o n^o de graus de liberdade

p: se p=0 não estima o AR1

q: se q=0 não estima o MA1

▶ **Output**

▶ **Library**

library est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\arma_hetero.src

8.3 estima_brw_01

▶ Purpose

Estimates the model

$$\begin{aligned} y_t &= y_{t-1} - \beta_1 F\left(y_{t-1} - \mu_t\right) + \beta_1 F\left(-y_{t-1} + \mu_t\right) + \sigma\varepsilon_t \\ \mu_t &= \lambda y_{t-1} + (1-\lambda)\,\mu_{t-1} \\ F\left(x\right) &= \frac{1}{1+\exp\left(-\gamma\left(x-\alpha\right)\right)} \end{aligned}$$

▶ Formato

{b,f0,grad,cov,retcode,media}=estima_brw_01(b0,rest,y);

▶ Input

b0: initial values of $(\beta_1, \beta_2, \lambda, \alpha, \gamma, \mu_0)$

rest: example: rest=-999|-999|0|-999|-999|2 means that there are two resctritions: $\lambda = 0$ and $\mu_0 = 2$

y:

▶ Output

Along with the usual results, the output includes

****

SSR and var_res BRW

SSR and var_res RW

lnFV BRW

lnFV RW

****

********** AR MODEL **********

SSR AR

****

»

▶ Library

library cml,est_etd;

▶ Remarks

Imposing the restriction $\lambda = 0$, leads to the model

$$y_t = y_{t-1} - \beta_1 F\left(y_{t-1} - \mu\right) + \beta_1 F\left(-y_{t-1} + \mu\right) + \sigma\varepsilon_t$$
$$F\left(x\right) = \frac{1}{1 + \exp\left(-\gamma\left(x - \alpha\right)\right)}.$$

The estimate of μ is μ_0 (last element of b).

► **Fonte**

c:\gauss\srcnic\est_etd\nao_linear.src

8.4 estima_brw_01B

▶ Purpose

Estimates the model

$$y_t = y_{t-1} - \beta_1 F_1\left(y_{t-1} - \mu_t\right) + \beta_2 F_2\left(-y_{t-1} + \mu_t\right) + \sigma\varepsilon_t$$
$$\mu_t = \lambda y_{t-1} + (1-\lambda)\,\mu_{t-1}$$
$$F_1(x) = \frac{1}{1+\exp\left(-\gamma_1\left(x-\alpha\right)\right)}, F_2(x) = \frac{1}{1+\exp\left(-\gamma_2\left(x-\alpha\right)\right)}$$

▶ Formato

{b,f0,grad,cov,retcode,media}=estima_brw_01B(b0,rest,y);

▶ Input

b0: initial values of $(\beta_1, \beta_2, \lambda, \alpha, \gamma_1, \gamma_2, \mu_0)$

rest: example: rest=-999|-999|0|-999|-999|-999|2 means that there are two restrictions: $\lambda = 0$ and $\mu_0 = 2$

y:

▶ Output

Along with the usual results, the output includes the following:

****

SSR and var_res BRW

SSR and var_res RW

lnFV BRW

lnFV RW

****

********** AR MODEL **********

SSR AR

****

»

▶ Library

library cml,est_etd;

▶ Remarks

Imposing the restriction $\lambda = 0$, leads to the model

$$y_t = y_{t-1} - \beta_1 F_1 (y_{t-1} - \mu) + \beta_2 F_2 (-y_{t-1} + \mu) + \sigma\varepsilon_t$$
$$F_1 (x) = \frac{1}{1 + \exp(-\gamma_1 (x - \alpha))}, F_2 (x) = \frac{1}{1 + \exp(-\gamma_2 (x - \alpha))}$$

The estimate of μ is μ_0 (last element of b).

▶ **Fonte**

c:\gauss\srcnic\est_etd\nao_linear.src

8.5 estima_brw_02

▶ Purpose

Estimates the model

$$\begin{aligned} y_t &= y_{t-1} - \beta_1 F\left(y_{t-1} - \mu_t\right) + \beta_1 F\left(-y_{t-1} + \mu_t\right) + u_t, \qquad u_t = \sigma_t \varepsilon_t \\ \mu_t &= \lambda y_{t-1} + (1-\lambda)\,\mu_{t-1} \\ \sigma_t^2 &= v_0 + v_1 u_{t-1}^2 + v_2 \sigma_{t-1}^2 \\ F(x) &= \frac{1}{1 + \exp\left(-\gamma\left(x - \alpha\right)\right)} \end{aligned}$$

▶ Formato

{b,f0,grad,cov,retcode,media,var}=estima_brw_02(b0,rest,y);

▶ Input

b0: initial values of $(\beta_1, \beta_2, \lambda, \alpha, \gamma, \mu_0, v_0, v_1, v_2)$

rest: example: rest=-999|-999|0|-999|-999|2|-999|-999|-999 means that there are two restrictions: $\lambda = 0$ and $\mu_0 = 2$

y:

▶ Output

▶ Library

library cml,est_etd;

▶ Remarks

Imposing the restriction $\lambda = 0$, leads to the model

$$\begin{aligned} y_t &= y_{t-1} - \beta_1 F\left(y_{t-1} - \mu\right) + \beta_1 F\left(-y_{t-1} + \mu\right) + \sigma_t \varepsilon_t \\ \sigma_t^2 &= v_0 + v_1 u_{t-1}^2 + v_2 \sigma_{t-1}^2 \\ F(x) &= \frac{1}{1 + \exp\left(-\gamma\left(x - \alpha\right)\right)}. \end{aligned}$$

The estimate of μ is μ_0.

▶ Fonte

c:\gauss\srcnic\est_etd\nao_linear.src

8.6 estima_categorical_TS_1

▶ Purpose

Let

$$y_{tj} = \begin{cases} 1 \text{ if the } j\text{th category is observed at time } t \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

The procedure estimates the model

$$\begin{cases} P\left(y_{t1} = 1 | \mathcal{F}_{t-1}\right) = \frac{\exp(\beta_{11}x_{t1}+...+\beta_{1k}x_{tk})}{1+\sum_{i=1}^{m-1}\exp(\beta_{i1}x_{t1}+...+\beta_{ik}x_{tk})} \\ ... \\ P\left(y_{t,m-1} = 1 | \mathcal{F}_{t-1}\right) = \frac{\exp(\beta_{m1}x_{t1}+...+\beta_{mk}x_{tk})}{1+\sum_{i=1}^{m-1}\exp(\beta_{i1}x_{t1}+...+\beta_{ik}x_{tk})} \\ P\left(y_{tm} = 1 | \mathcal{F}_{t-1}\right) = 1 - P\left(y_{t1} = 1 | \mathcal{F}_{t-1}\right) - ... - P\left(y_{tm} = 1 | \mathcal{F}_{t-1}\right) \end{cases}$$

through the ML method,

$$\max_{\boldsymbol{\beta}} \sum_{t=1}^{n}\sum_{j=1}^{m} y_{tj} \log P\left(y_{t1} = j | \mathcal{F}_{t-1}; \boldsymbol{\beta}\right).$$

▶ Formato

{b,p}=estima_categorical_TS_1(y,x);

▶ Input

y: $n \times m$ matrix (m categories)

x: $n \times k$ matrix (k explanatory variables)

▶ Output

b: $(m-1) * k \times 1$ vector.

$$b = \begin{bmatrix} \beta \text{ first equation} \\ \beta \text{ second equation} \\ ... \\ \beta \text{ last equation} \end{bmatrix} = \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ ... \\ \beta_{21} \\ \beta_{22} \\ ... \\ \beta_{m-1,1} \\ ... \\ \beta_{m-1,k} \end{bmatrix}$$

▶ Library

library cml,cm;

▶ Fonte

c:\gauss\srcnic\est_etd\cm.src

8.7 estima_CM

▶ **Purpose**

Estimates the transition probability matrix from a Markov chain.

▶ **Format**

{c,p,se,sp}=estima_CM(s);

▶ **Input**

s: Observations of S.

▶ **Output**

c: absolute frequencies table

p: transition probability matrix

se: standard errors (p./se → gives the t ratios). See Basawa, p.55.

sp: stationary probabilities

▶ **Library**

library est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\cm.src

8.8 estima_CM1

▶ Purpose

Estimates the transition probability matrix from a Markov chain. Calculates log-likelihood and BIC

▶ Format

{c,p,st_erros,stat_prob,ll,bic,par_independentes}=estima_CM1(s);

▶ Input

s: Observations of S.

▶ Output

c: absolute frequencies table

p: transition probability matrix

se: standard errors (p./se → gives the t ratios). See Basawa, p.55.

sp: stationary probabilities

ll: log-likelihood

BIC: $BIC = -2ll + q \ln(n)$ where q is the number of independent parameters.

par_independentes:

▶ Remark

The terms in which $\hat{p}_{ij} = 0$ have no contribution to the log-likelihood. Also, the value q excludes the cases where $\hat{p}_{ij} = 0$.

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\cm.src

8.9 estima_estar_1

▶ Purpose

Estimates the model

$$y_t = \mu + \phi_1 (y_{t-1} - \mu) + \phi_2 (y_{t-1} - \mu) \left(1 - \exp\left(-\theta^2 (y_{t-1} - \mu)^2\right)\right) + \sigma\varepsilon_t$$

assuming that $\{\varepsilon_t\}$ is a Gaussian white noise.

▶ Formato

{b,f0,grad,cov,retcode,media}= estima_estar_1(b0,y);

▶ Input

b0: initial values of $(\phi_1, \phi_2, \mu, \theta, \sigma)$

y: vector of observations.

▶ Output

Along with the usual results, the output includes

****

SSR and var_res BRW

SSR and var_res RW

lnFV BRW

lnFV RW

****

********** AR MODEL **********

SSR AR

****

▶ Library

library cml,est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\nao_linear.src

8.10 estima_fdpKG

▶ Objectivo

Estimar a fdp de um vector de observações através do kernel Gaussiano

▶ Formato

{x,p,ic,pg}=estima_fdpKG(y,pontos,h,int_conf);

▶ Input

y: vector das observações do processo.

pontos: escalar = número de pontos equidistantes para os quais se obtém a correspondente estimativa. Caso pontos $=\{x_1, ..., x_k\}$ é calculada uma estimativa para cada um destes pontos.

h: Igual 1 bandwith usual (valor maior (menor) maior (menor) alisamento).

int_conf: nível de confiança

▶ Output

x: $\{x_1, ..., x_k\}$ onde $k =$ pontos.

p: estimativas de $\{f(x_1), ..., f(x_k)\}$.

pg: fdp gaussiana com média $\bar{y}$ e variância $\hat{\sigma}_y^2$.

ic

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\nao_p_td.src

8.11 misture_normal_01

▶ Objectivo

Estima os parâmetros α, μ_1, σ_1, μ_2 e σ_2 da mistura

$$f(x) = \alpha f_{x|u}(x|0) + (1-\alpha) f_{x|u}(x|1), \ 0 \leq \alpha \leq 1$$

sendo $f_{x|u}(x|0)$ a fdp da distribuição $N\left(\mu_1, \sigma_1^2\right)$ e $f_{x|u}(x|1)$ a dfp da distribuição $N\left(\mu_2, \sigma_2^2\right)$.

▶ Formato

{b,f0,grad,cov,retcode}=misture_normal_01(b0,y);

▶ Input

b0: Se b0=0 => b0=0.4|meanc(y)|meanc(y)|stdc(y)|stdc(y);

y: vector das observações;

▶ Output

▶ Library

library est_etd, cml;

▶ Fonte

c:\gauss\srcnic\est_etd\misture_normal_01.src

8.12 estima_nao_linear_01

▶ Objectivo

Estima o modelo

$$\begin{aligned} y_t = & \beta' \mathbf{x}_t + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} \\ & + \xi_1 u_{t-1} + ... \\ & + \xi_{11} u_{t-1} u_{t-2} + ... \\ & + \xi_{11} u_{t-1} u_{t-2} u_{t-3} + ... \\ & + ... + u_t \end{aligned}$$

▶ Formato

{b,f0,grad,cov,retcode,media,var}=estima_nao_linear_01(y,x,ar,par);

▶ Input

y:

x:

ar: vector de valores inteiros. ar=1|3|5⇒ estima o $\phi_1 y_{t-1} + \phi_3 y_{t-3} + \phi_5 y_{t-5}$. Se ar={} não estima a componente AR

par: matriz. Exemplo:

$$par = \begin{bmatrix} 1\,2\,0 \\ 1\,2\,3 \end{bmatrix}$$

Estima $\xi_{12} u_{t-1} u_{t-2} + \xi_{123} u_{t-1} u_{t-2} u_{t-3}$. Se

$$par = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

Estima MA(3). Se

$$par = \begin{bmatrix} 1\,3\,4\,5 \end{bmatrix}$$

Estima $\xi_{1345} u_{t-1} u_{t-3} u_{t-4} u_{t-5}$. NOTA: n^o de linhas de par corresponde ao número de parâmetros ξ a estimar

▶ **Output**

▶ **Library**

library est_etd,util;

▶ **Fonte**

c:\gauss\srcnic\est_etd\nao_linear.src

8.13 estima_TAR_2Regimes

▶ **Purposes**

Estimates

$$y_t = \begin{cases} \phi_{10} + \phi_{11} y_{t-1} + ... + \phi_{1p} y_{t-p} + u_t \; q_{t-d} \leq \gamma \\ \phi_{20} + \phi_{21} y_{t-1} + ... + \phi_{2p} y_{t-p} + u_t \; q_{t-d} > \gamma \end{cases}$$

and AR(p).

▶ **Format**

{d_opt,g_opt,min_sqr_mean,Fobs,resid}=estima_TAR_2Regimes(y,q,dmax,p,alfa);

▶ **Input**

y:

q:

dmax: Maximum value of d (the routine finds the optimal value in the set $\{1, 2, ..., \text{dmax}\}$).

p: Order of p.

alfa: scalar between 0 and 1. Proportion of observations trimmed in the set $\tilde{\Gamma}$ (see Hansen, B. E. (2000), Sample Splitting and Threshold Estimation. *Econometrica*, 68: 575–603). You may start with alfa=0.1 (a value too high makes the procedure faster, but the possibility of discarding the optimal γ value increases).

▶ **Output**

d_opt: $\hat{d}$ from the solution $\left(\hat{\gamma}, \hat{d}\right) = \arg\min_{\gamma \in \tilde{\Gamma}, d \in D} \hat{\sigma}^2(\gamma, d)$

g_opt: $\hat{\gamma}$ from the solution $\left(\hat{\gamma}, \hat{d}\right) = \arg\min_{\gamma \in \tilde{\Gamma}, d \in D} \hat{\sigma}^2(\gamma, d)$

min_sqr_mean: minimum $\hat{\sigma}^2(\gamma, d)$

Fobs: $F_n = n\left(\tilde{\sigma}_n^2 - \hat{\sigma}_n^2\right)/\hat{\sigma}_n^2$ (see Hansen, 2000)

resid:

▶ **Library**

library pgraph, est_etd,util;

▶ **Fonte**

c:\gauss\srcnic\est_etd\tar.src

▶ **Observações**

Run the routine test_TAR1 to test the AR against the TAR.

8.14 estima_TAR_3Regimes

▶ **Objectivo**

Estimar

$$y_t = \begin{cases} \phi_{10} + \phi_{11} y_{t-1} + ... + \phi_{1p} y_{t-p} + u_t & q_{t-d} < \gamma_1 \\ \phi_{20} + \phi_{21} y_{t-1} + ... + \phi_{2p} y_{t-p} + u_t \, \gamma_1 \leq q_{t-d} \leq \gamma_2 \\ \phi_{30} + \phi_{31} y_{t-1} + ... + \phi_{3p} y_{t-p} + u_t & q_{t-d} > \gamma_2 \end{cases}$$

e o AR(p)

▶ **Formato**

{d_opt,g1_opt,g2_opt,min_sqr_mean,Fobs,resid}=estima_TAR_3Regimes(y,q,dmax,p,alfa,min2)

▶ **Input**

y:

q:

dmax: Valor máximo a considerar para d.

p: Ordem do AR

alfa: valor entre 0 e 1. Fracção das observações aparadas no conjunto $\tilde{\Gamma}$ (ver o manual "Séries Temporais Financeiras"). Considerar por exemplo alfa=0.1 (um alfa mais alto torna o procedimento mais rápido, mas a possibilidade de descartar o $\hat{\gamma}$ óptimo amenta).

min2: n^o mínimo de obs. no segundo regime

▶ Output

d_opt: $\hat{d}$ da solução $\left(\hat{\gamma}_1, \hat{\gamma}_2, \hat{d}\right) = \arg\min_{\gamma_1,\gamma_2\in\tilde{\Gamma},d\in D} \hat{\sigma}^2(\gamma_1,\gamma_2,d)$

g1_opt: $\hat{\gamma}_1$ da solução $\left(\hat{\gamma}_1, \hat{\gamma}_2, \hat{d}\right) = \arg\min_{\gamma_1,\gamma_2\in\tilde{\Gamma},d\in D} \hat{\sigma}^2(\gamma_1,\gamma_2,d)$

g2_opt: $\hat{\gamma}_2$ da solução $\left(\hat{\gamma}_1, \hat{\gamma}_2, \hat{d}\right) = \arg\min_{\gamma_1,\gamma_2\in\tilde{\Gamma},d\in D} \hat{\sigma}^2(\gamma_1,\gamma_2,d)$

min_sqr_mean: $\hat{\sigma}^2(\gamma, d)$ minimo

Fobs: $F_n = n\left(\tilde{\sigma}_n^2 - \hat{\sigma}_n^2\right)/\hat{\sigma}_n^2$ (ver o manual "Séries Temporais Financeiras")

resid:

▶ Library

library pgraph, est_etd,util;

▶ Fonte

c:\gauss\srcnic\est_etd\tar.src

▶ Observações

Não está ainda implementado um teste AR versus STAR

8.15 EWMA_Multivariado

▶ Objectivo

Estima a matriz de variâncias-covariâncias condicionais através do modelo

$$\mathbf{H}_t = (1-\lambda)\,\mathbf{y}_{t-1}\mathbf{y}'_{t-1} + \lambda\mathbf{H}_{t-1}$$

e no caso $m = 3$ apresenta informação gráfica.

▶ Formato

G=EWMA_Multivariado(y,lam);

▶ Input

y: matriz das observações do processo (y1~y2 ~...) reportados todos à mesma data

▶ Output

G: $vech\,(H)$, isto é,

$$\begin{aligned} G[.,1] &\rightarrow \text{série da } \operatorname{Var}[y_{1t}|\,\mathcal{F}_{t-1}] \\ G[.,2] &\rightarrow \text{série da } \operatorname{Cov}(y_{1t}, y_{2t}|\,\mathcal{F}_{t-1}) \\ G[.,3] &\rightarrow \text{série da } \operatorname{Var}[y_{3t}|\,\mathcal{F}_{t-1}] \\ G[.,4] &\rightarrow \text{série da } \operatorname{Cov}(y_{1t}, y_{3t}|\,\mathcal{F}_{t-1}) \\ &\ldots \end{aligned}$$

▶ Library

library est_etd;

▶ Observações

Acertar primeiro as datas, por exemplo, com:

{data,y1,y2,y3}=acerta_datas_3(data1,y1,data2,y2,data3,y3);

Incompleto (aumentar a informação grafica para outros valores de m)

▶ Fonte

c:\gauss\srcnic\est_etd\mult_volati.src

8.16 hac_estimator

► **Objective**

(1) Estimate

$$\begin{aligned}\mathbf{S} &= \operatorname{Var}\left(\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i u_i\right) = \frac{1}{n}\operatorname{Var}\left(\sum_{i=1}^{n}\mathbf{x}_i u_i\right) \\ &= \mathrm{E}\left(u_t^2\mathbf{x}_t\mathbf{x}_t'\right) + \frac{1}{n}\sum_{j=1}^{n-1}\sum_{t=j+1}^{n}\left(\mathrm{E}\left(u_t u_{t-j}\mathbf{x}_t\mathbf{x}_{t-j}'\right) + \mathrm{E}\left(u_{t-j}u_t\mathbf{x}_{t-j}\mathbf{x}_t'\right)\right)\end{aligned}$$

using the HAC estimator (see below), $\mathbf{S}_{HAC}$. Note that this estimator can also estimate the long run variance of $\{\mathbf{y}_t\}$, i.e

$$\frac{1}{n}\operatorname{Var}\left(\sum_{i=1}^{n}\mathbf{y}_i\right)$$

$(m \times 1)$.

(2) Gives the diagonal of the matrix

$$\widehat{\operatorname{Var}}(\mathbf{b}) = \hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}\mathbf{S}_{HAC}\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}/n$$

where

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1} = \left(\frac{1}{n}\sum_{t=1}^{n}\mathbf{x}_t\mathbf{x}_t'\right)^{-1} = \left(\frac{\mathbf{X}'\mathbf{X}}{n}\right)^{-1}.$$

Context: Given

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) = \left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i'\right)^{-1}\left(\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i u_i\right)$$

and

$$\left(\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i u_i\right) \xrightarrow{d} N(\mathbf{0}, \mathbf{S})$$

we have

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) = \left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i'\right)^{-1}\left(\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i u_i\right) \xrightarrow{d} N\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{xx}}^{-1}\mathbf{S}\mathbf{\Sigma}_{\mathbf{xx}}^{-1}\right).$$

To estimate $\mathbf{S}$ in the presence of heteroskedasticity and autocorrelation of unknown form one considers

$$\hat{\mathbf{S}} = \mathbf{S}_{HAC} = \frac{1}{n}\sum_{t=1}^{n}\hat{u}_t^2\mathbf{x}_t\mathbf{x}_t' + \frac{1}{n}\sum_{j=1}^{L}\sum_{t=j+1}^{n}\omega(j)\left(\hat{u}_t\hat{u}_{t-j}\mathbf{x}_t\mathbf{x}_{t-j}' + \hat{u}_{t-j}\hat{u}_t\mathbf{x}_{t-j}\mathbf{x}_t'\right)$$

where the weighting function $\omega(j)$ is

$$\omega(j) = 1 - \frac{j}{L+1}.$$

Note

$$\text{Avar}\,(\mathbf{b}) = \mathbf{\Sigma}_{\mathbf{xx}}^{-1}\mathbf{S}\mathbf{\Sigma}_{\mathbf{xx}}^{-1}$$

and

$$\widehat{\text{Var}}\,(b_i) = \left[\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}\mathbf{S}_{HAC}\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}/n\right]_{ii}$$

▶ **Format**

{S,SE}=hac_estimator(e,x,L);

▶ **Input**

e: residuals from an econometric model

x: matrix of explanatory variables

L: See above

▶ **Remark**

It is also possible to estimate the long-run variance of a $m \times 1$ vector $\mathbf{y}_t$, i.e.

$$\frac{1}{n}\,\text{Var}\left(\sum_{i=1}^{n}\mathbf{y}_i\right).$$

In this case use the instruction S=hac_estimator(1,y,L);

Note also that "S=hac_estimator(e,x,L);" and "S=hac_estimator(1,x.*e,L);" lead to the same S matrix.

▶ **Output**

S: $\hat{\mathbf{S}}_{HAC}$

SE: Vector with the standard errors $\sqrt{\widehat{\text{Var}}\,(b_i)}$, where $\widehat{\text{Var}}\,(b_i) = \left[\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}\mathbf{S}_{HAC}\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}/n\right]_{ii}$.

▶ **Library**

library est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\estima_cov.src

8.17 hac_estimator_whitening

▶ Objective

See Andrews and Monahan (1992).

▶ Format

SE=hac_estimator_whitening(x,e,desf,L);

▶ Input

e: residuals from an econometric model

x: matrix of explanatory variables

desf:

L:

▶ Output

SE: Vector with the standard errors $\sqrt{\widehat{\text{Var}}(b_i)}$.

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\estima_cov.src

8.18 med_cond

▶ Objectivo

Estimar não parametricamente a média condicional $E[X_t|\, X_{t-k}]$, $k \geq 1$, a partir das observações $(X_1, X_2, ..., X_n)$. O estimador é

$$\mu(x) = \frac{\sum_{i=1}^{n-1} K\left(\frac{X_{t-k}-x}{h}\right) X_t}{\sum_{i=1}^{n-1} K\left(\frac{X_{t-k}-x}{h}\right)}$$

onde $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$.

▶ Formato

{x,m}=med_cond(y,pontos,des);

▶ Input

y: vector de observações $(X_1, X_2, ..., X_n)$ do processo X.

pontos: número de pontos x_i para os quais se calcula $\mu(x_i)$. Isto é $i = 1, 2, ...,$pontos. Se pontos = 0, calcula-se $\mu(x_i)$ onde x_i assume todos os valores da amostra $(X_1, X_2, ..., X_n)$.

des: valor de k.

▶ Output

x: vector x.

m: vector $\mu(x)$.

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\nao_p_td.src

8.19 med_cond1

▶ Objectivo

Estimar não parametricamente a média condicional $E[X_t|\, X_{t-1}]$ e obter um intervalo de confiança a partir das observações $(X_1, X_2, ..., X_n)$. O estimador é

$$\mu(x) = \frac{\sum_{i=1}^{n-1} K\left(\frac{X_{t-1}-x}{h}\right) X_t}{\sum_{i=1}^{n-1} K\left(\frac{X_{t-1}-x}{h}\right)}$$

onde $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$. O intervalo de confiança é dado por

$$\mu(x) \pm z\sqrt{\frac{kv(x)}{f(x)nh}}$$

onde z é um quantil da distribuição normal, $v(x)$ é a variância condicional (estimada) de X_t dado $X_{t-1} = x$ (ver VAR_COND), $k = \int K(u)^2 du$ e $f(x)$ é a densidade estacionária (estimada) de X (ver KERNEL_FDP_G).

▶ Formato

{x,m,var,lim_infM,lim_supM,densidade}=med_cond1(y,pontos,aparar,int_conf);

▶ Input

y: vector de observações $(X_1, X_2, ..., X_n)$ do processo X.

pontos: número de pontos x_i para os quais se calcula $\mu(x_i)$. Isto é $i = 1, 2, ...,$pontos. Se pontos = 0, calcula-se $\mu(x_i)$ onde x_i assume todos os valores da amostra $(X_1, X_2, ..., X_n)$.

aparar: assume um valor no intervalo $]0, 1]$. Se aparar $\in]0, 1[$ o procedimento é apenas executado para os ponto x_i que se encontram entre o percentil $(1 - \text{aparar})/2$ e o percentil $(1 + \text{aparar})/2$ da distribuição amostral de $(X_1, X_2, ..., X_n)$. Se aparar = 1 então $x_1 = \min_{i=1,...,n}(X_i)$ e $x_{pontos} = \max_{i=1,...,n}(X_i)$.

int_conf: valor entre 0 e 1 referente ao nível de confiança do intervalo.

▶ Output

x: vector x.

m: vector $\mu(x)$.

var: vector $v(x)$.

lim_inf: limite inferior do intervalo de confiança para $\mu(x)$

lim_inf: limite superior do intervalo de confiança.

densidade: vector $f(x)$.

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\nao_p_td.src

8.20 med_cond2

▶ Objectivo

Estimar não parametricamente $E\left[X_t - X_{t-1} | X_{t-1}\right]$ e obter um intervalo de confiança a partir das observações $(X_1, X_2, ..., X_n)$. O estimador é

$$\mu(x) = \frac{\sum_{i=1}^{n-1} K\left(\frac{X_{t-1}-x}{h}\right)(X_t - X_{t-1})}{\sum_{i=1}^{n-1} K\left(\frac{X_{t-1}-x}{h}\right)}$$

onde $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = factor\_h\,(4/3)^{1/5}\,\hat{\sigma}_X n^{-1/5}$. O intervalo de confiança é dado por

$$\mu(x) \pm z\sqrt{\frac{kv(x)}{f(x)\,nh}}$$

onde z é um quantil da distribuição normal, $v(x)$ é a variância condicional (estimada) de $X_t - X_{t-1}$ dado $X_{t-1} = x$, $k = \int K(u)^2\,du$ e $f(x)$ é a densidade estacionária (estimada) de X (ver KERNEL_FDP_G).

▶ Formato

{x,m,var,lim_infM,lim_supM,densidade}=med_cond2(y,pontos,aparar,int_conf,factor_h);

▶ Input

y: vector de observações $(X_1, X_2, ..., X_n)$ do processo X.

pontos: número de pontos x_i para os quais se calcula $\mu(x_i)$. Isto é $i = 1, 2, ...,$pontos. Se pontos = 0, calcula-se $\mu(x_i)$ onde x_i assume todos os valores da amostra $(X_1, X_2, ..., X_n)$.

aparar: assume um valor no intervalo $]0, 1]$. Se aparar $\in]0, 1[$ o procedimento é apenas executado para os ponto x_i que se encontram entre o percentil $(1 - \text{aparar})/2$ e o percentil $(1 + \text{aparar})/2$ da distribuição amostral de $(X_1, X_2, ..., X_n)$. Se aparar = 1 então $x_1 = \min_{i=1,...,n}(X_i)$ e $x_{pontos} = \max_{i=1,...,n}(X_i)$.

int_conf: valor entre 0 e 1 referente ao nível de confiança do intervalo.

factor_h: escalar presente na expressão $h = factor\_h\,(4/3)^{1/5}\,\hat{\sigma}_X n^{-1/5}$.

▶ Output

x: vector x.

m: vector $\mu(x)$.

var: vector $v(x)$.

lim_inf: limite inferior do intervalo de confiança para $\mu(x)$

lim_inf: limite superior do intervalo de confiança.

densidade: vector $f(x)$.

▶ **Library**

library est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\nao_p_td.src

8.21 Mult_GARCH11

▶ **Objectivo**

Estimar o modelo bivariado AR(r)+GARCH(p,q)

$$Y_t^A = \phi_1^A Y_{1t-1} + ... + \phi_{r_1}^A Y_{1t-r_1} + \beta_1^A X_{1t}^A + ... + \beta_{k_1}^A X_{k_1 t}^A + u_t^A$$

$$Y_t^B = \phi_1^B Y_{1t-1} + ... + \phi_{r_2}^B Y_{1t-r_2} + \beta_1^B X_{1t}^A + ... + \beta_{k_2}^B X_{k_2 t}^B + u_t^B$$

onde se admite,

$$Var\begin{bmatrix} u_t^A \\ u_t^B \end{bmatrix} = \begin{bmatrix} h_{1t} & h_{12t} \\ h_{12t} & h_{2t} \end{bmatrix}$$

com,

$$\begin{bmatrix} h_{1t} \\ h_{12t} \\ h_{2t} \end{bmatrix} = \begin{bmatrix} k^A \\ k^{AB} \\ k^B \end{bmatrix} + \begin{bmatrix} \alpha_1^A & \alpha_2^A & \alpha_3^A \\ \alpha_1^{AB} & \alpha_2^{AB} & \alpha_3^{AB} \\ \alpha_1^B & \alpha_2^B & \alpha_3^B \end{bmatrix} \begin{bmatrix} u_{1t-1}^2 \\ u_{1t-1}u_{2t-1} \\ u_{2t-1}^2 \end{bmatrix} + \begin{bmatrix} \delta_1^A & \delta_2^A & \delta_3^A \\ \delta_1^{AB} & \delta_2^{AB} & \delta_3^{AB} \\ \delta_1^B & \delta_2^B & \delta_3^B \end{bmatrix} \begin{bmatrix} h_{1t-1} \\ h_{12t-1} \\ h_{2t-1} \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^s \psi_i g_{it} \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

▶ **Formato**

{beta,f0,grad,cov,retcode,mediaA,mediaB,varA,varB,varAB}

=Mult_GARCH11(b0,rest,yA,desA,xA,g,yB,desB,xB);

▶ **Input**

b0: valor inicial para o vector de parâmetros a estimar:

$$\left(\phi^A, \beta^A, k^A, \alpha^A, \delta^A, \psi^A, \phi^B, \beta^B, k^B, \alpha^B, \delta^B, k^{AB}, \alpha^{AB}, \delta^{AB}\right)$$

rest: vector de tipo $R \times 1$ com $R \geq 1$ (e menor do que o número de parâmetros a estimar). Cada linha indica o número do parâmetro que deve ser nulo. Por exemplo, "rest={4,7,9};" indica que os parâmetros 4, 7 e 9 são nulos.

yA: vector de tipo $n \times 1$ das observações do processo yA.

desA: ordem $r_1 > 0$.

xA: matriz de tipo $k_1 \times n$ das variáveis explicativas X^A ($k_1 > 0$).

g: matriz de tipo $s \times n$ das variáveis explicativas de h_{1t} ($s > 0$).

yB: vector de tipo $n \times 1$ das observações do processo yB.

desB: ordem $r_2 > 0$.

xB: matriz de tipo $k_2 \times n$ das variáveis explicativas X^B ($k_2 > 0$).

▶ Output

b: vector das estimativas associadas a b0.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

mediaA: média condicional de Y^A.

mediaB: média condicional de Y^B.

varA: variância condicional de Y^A.

varB: variância condicional de Y^B.

varAB: covariância condicional entre Y^A e Y^B

▶ Library

library cml,est_etd;

▶ Observações

Como o programa exige que todos os parâmetros k_1,s, k_2 sejam maiores do que zero é necessário fornecer como inputs matrizes X^A, g e X^B compatíveis (pelo menos de tipo $n \times 1$). Se não se incorporar esta informação ao modelo, deve-se utilizar o input "rest" que permite impor restrições de nulidade. Esta observação é também válida no caso em que não se deseje considerar esquemas AR para as variáveis dependentes. Naturalmente, o número de linhas do input b0 deve ser igual ao número de parâmetros a estimar que está implícito nos demais inputs. A estimação do modelo é geralmente muito demorada - dever-se-á considerar sempre várias restrições de nulidade (pelo menos 5 ou 6).

▶ Fonte

c:\gauss\srcnic\est_etd\garch.src

8.22 normal_garch

▶ Objectivo

Estimar o modelo AR(r)+GARCH(p,q)

$$Y_t = \phi_1 Y_{t-1} + ... + \phi_r Y_{t-r} + \beta_1 X_{1t} + ... + \beta_k X_{kt} + u_t, \qquad u_t = \sigma_t \varepsilon_t, \qquad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + ... + \alpha_q u_{t-q}^2 + \delta_1 \sigma_{t-1}^2 + ... + \delta_p \sigma_{t-p}^2 + \psi_1 g_{1t} + ... + \psi_s g_{st}.$$

▶ Formato

{b,f0,grad,cov,retcode,media,var}=normal_garch(b0,y,des,x,p,q,g);

▶ Input

b0: valor inicial para o vector de parâmetros a estimar: $(\phi, \beta, \alpha, \delta, \psi)$

y: vector de tipo $n \times 1$ das observações do processo.

des: ordem r ($r \geq 0$).

x: matriz de tipo $k \times n$ das variáveis explicativas X. Se x = escalar a matriz X é considerada nula.

p: ordem p ($p > 0$).

q: ordem q ($q > 0$).

g: matriz de tipo $s \times n$ das variáveis explicativas de σ_t^2. Se g = escalar a matriz g é considerada nula.

▶ Output

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: média condicional de Y.

var: variância condicional de Y.

▶ Library

library cml,est_etd;

▶ Observações

O número de linhas do input b0 deve ser igual ao número de parâmetros a estimar que está implícito nos demais inputs, des, x, p, q e g. Se se desejar fixar q ou p igual a zero considerar as instruções _cml_A e _cml_B (consultar o manual do GAUSS).

Exemplo: AR(1)+GARCH(1,1)

des=1;

b0=.1|.1|.1|.1|.1;

p=1;

q=1;

g=0;

{b,f0,grad,cov,retcode,media,var}=normal_garch(b0,y,des,ones(n,1),p,q,g);

► **Fonte**

c:\gauss\srcnic\est_etd\garch.src

8.23 pal_exponencial

► **Objectivo**

Estimar o modelo

$$X_t = X_{t-1} + e^k \left(e^{-\alpha_1 (X_{t-1} - \tau)} - e^{\alpha_2 (X_{t-1} - \tau)} \right) + \sigma_t \varepsilon_t, \qquad \varepsilon_t \sim N(0,1)$$

$$\sigma_t^2 = \exp\left\{ \beta + \lambda (X_{t-1} - \mu)^2 \right\}$$

através do método da máxima verosimilhança.

► **Formato**

{b,f0,grad,cov,retcode,media,var}=pal_exponencial(b0,y);

► **Input**

b0: valor inicial de $(\alpha_1, \alpha_2, \tau, k, \beta, \lambda, \mu)$.

y: vector das observações do processo.

► **Output**

b: vector das estimativas de $(\alpha_1, \alpha_2, \tau, k, \beta, \lambda, \mu)$

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\pal.src

8.24 pal_linear0

▶ **Objectivo**

Estimar o modelo

$$X_t = X_{t-1} + e^k\left(e^{-\alpha_1(X_{t-1}-\tau)} - e^{\alpha_2(X_{t-1}-\tau)}\right) + \sigma\varepsilon_t, \qquad \varepsilon_t \sim N(0,1)$$

através do método da máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode,mediar}=pal_linear0(b0,y);

▶ **Input**

b0: valor inicial de $(\alpha_1, \alpha_2, \tau, k, \sigma)$.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas de $(\alpha_1, \alpha_2, \tau, k, \sigma)$

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_etd;

▶ **Observações**

O *score* é avaliado a partir da sua expressão exacta.

▶ **Fonte**

c:\gauss\srcnic\est_etd\pal.src

8.25 pal_linear1

▶ **Objectivo**

Estimar o modelo

$$X_t = X_{t-1} + e^k \left(e^{-\alpha_1 (X_{t-1} - \tau)} - e^{\alpha_2 (X_{t-1} - \tau)} \right) + \sigma_t \varepsilon_t, \qquad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = \sigma_0 + \sigma_1 |X_{t-1}|$$

através do método da máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode,media,var}=pal_linear1(b0,y);

▶ **Input**

b0: valor inicial de $(\alpha_1, \alpha_2, \tau, k, \sigma_0, \sigma_1)$.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas de $(\alpha_1, \alpha_2, \tau, k, \sigma_0, \sigma_1)$

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\pal.src

8.26 pal_garch1

▶ **Objectivo**

Estimar o modelo

$$X_t = X_{t-1} + e^k \left(e^{-\alpha_1 (X_{t-1} - \tau)} - e^{\alpha_2 (X_{t-1} - \tau)} \right) + u_t, \qquad u_t = \sigma_t \varepsilon_t, \qquad \varepsilon_t \sim N(0,1)$$
$$\sigma_t^2 = \phi_0 + \phi_1 u_{t-1}^2 + \delta_1 \sigma_{t-1}^2$$

através do método da máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode,media,var}=pal_garch1(b0,y);

▶ **Input**

b0: valor inicial de $\left(\alpha_1, \alpha_2, \tau, k, \phi_0, \phi_1, \delta_1, \sigma_0^2\right)$.

y: vector das observações do processo.

▶ **Output**

b: vector das estimativas de $\left(\alpha_1, \alpha_2, \tau, k, \phi_0, \phi_1, \delta_1, \sigma_0^2\right)$

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\pal.src

8.27 rs_td_1

▶ Objectivo

Estimar o seguinte modelo:

$$X_t = \alpha(S_t) + \beta(S_t) X_{t-1} + u_t \quad \text{(i)}$$
$$X_t - X_{t-1} = \alpha(S_t) + \beta(S_t) X_{t-1} + u_t \quad \text{(ii)}$$
$$u_t = \sigma(S_t)\varepsilon_t \quad \text{(iii)}$$

onde, ε_t é i.i.d. com distribuição $N(0,1)$,

$$\alpha(S_t) = \begin{cases} \alpha_1 \text{ se } S_t = 1 \\ \alpha_2 \text{ se } S_t = 2 \end{cases} \qquad \beta(S_t) = \begin{cases} \beta_1 \text{ se } S_t = 1 \\ \beta_2 \text{ se } S_t = 2 \end{cases} \qquad \sigma(S_t) = \begin{cases} \sigma_1 \text{ se } S_t = 1 \\ \sigma_2 \text{ se } S_t = 2 \end{cases}$$

e S_t é uma cadeia de Markov homogénea em tempo discreto com espaço de estados $\{1,2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}.$$

(ver **rs_td_4).**

▶ Formato

{b,f0,g,cov,retcode,prob1,media,var}=rs_td_1(y,b0,prim_dif);

▶ Input

y: vector das observações do processo.

b0: valor inicial de $(\alpha_1, \beta_1, \sigma_1, \alpha_2, \beta_2, \sigma_2, p_{11}, p_{22}, pa)$ (considerar pa=0.5).

prim_dif: se prim_dif=0 considera o modelo (i); no caso contrário considera o modelo (ii)

▶ Output

b: vector das estimativas de $(\alpha_1, \sigma_1, \alpha_2, \sigma_2, p_{11}, p_{22}, pa)$.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

prob1: vector das probabilidades $P[S_t = 1|\mathcal{F}_{t-1}]$ estimadas de tipo $(n-1) \times 1$ $(t = 2, ..., n)$.

► **Library**

library cml,est_etd;

► **Fonte**

c:\gauss\srcnic\est_etd\rs_td.src.

8.28 rs_td_2

► **Objectivo**

Estimar um modelo regime-switching $ARMA_{12}(1,0)(0,1)$ (modelo ARMA sazonal com periodicidade 12):

$$X_t = \alpha(S_t) + \beta(S_t) X_{t-1} + \theta(S_t) u_{t-12} + u_t$$
$$u_t = \sigma(S_t) \varepsilon_t$$

onde, ε_t é i.i.d. com distribuição $N(0,1)$,

$$\alpha(S_t) = \begin{cases} \alpha_1 \text{ se } S_t = 1 \\ \alpha_2 \text{ se } S_t = 2 \end{cases} \qquad \beta(S_t) = \begin{cases} \beta_1 \text{ se } S_t = 1 \\ \beta_2 \text{ se } S_t = 2 \end{cases}$$
$$\theta(S_t) = \begin{cases} \theta_1 \text{ se } S_t = 1 \\ \theta_2 \text{ se } S_t = 2 \end{cases} \qquad \sigma(S_t) = \begin{cases} \sigma_1 \text{ se } S_t = 1 \\ \sigma_2 \text{ se } S_t = 2 \end{cases}$$

e S_t é uma cadeia de Markov homogénea em tempo discreto com espaço de estados $\{1, 2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{bmatrix}.$$

► **Formato**

{b,cov,m1,m2,media,var,prob1}=rs_td_2(y,b0);

► **Input**

y: vector das observações do processo.

b0: valor inicial de $(\alpha_1, \beta_1, \theta_1, \sigma_1, \alpha_2, \beta_2, \theta_2, \sigma_2, p_{11}, p_{22}, pa)$

(considerar pa=0.5)

► **Output**

b: vector das estimativas de $(\alpha_1, \beta_1, \theta_1, \sigma_1, \alpha_2, \beta_2, \theta_2, \sigma_2, p_{11}, p_{22}, pa)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional associada ao regime 1 de tipo $(n-12)\times 1$.

m2: vector da média condicional associada ao regime 2 de tipo $(n-12)\times 1$.

media: vector da média condicional do processo de tipo $(n-12)\times 1$.

var: vector da variância condicional do processo de tipo $(n-12)\times 1$.

prob1: vector das probabilidades $P\left[S_t=1|\,\mathcal{F}_{t-1}\right]$ estimadas de tipo $(n-12)\times 1$ $(t=13,...,n)$.

▶ Library

library cml,est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\rs_td.src.

8.29 rs_td_2AR

▶ Objectivo

Estimar um modelo regime-switching $ARMA_{12}(1,0)(1,0)$ (modelo ARMA sazonal com periodicidade 12):

$$X_t = \alpha(S_t) + \beta(S_t) X_{t-1} + \theta(S_t) X_{t-12} + u_t$$
$$u_t = \sigma(S_t)\varepsilon_t$$

onde, ε_t é i.i.d. com distribuição $N(0,1)$,

$$\alpha(S_t) = \begin{cases} \alpha_1 \text{ se } S_t = 1 \\ \alpha_2 \text{ se } S_t = 2 \end{cases} \qquad \beta(S_t) = \begin{cases} \beta_1 \text{ se } S_t = 1 \\ \beta_2 \text{ se } S_t = 2 \end{cases}$$
$$\theta(S_t) = \begin{cases} \theta_1 \text{ se } S_t = 1 \\ \theta_2 \text{ se } S_t = 2 \end{cases} \qquad \sigma(S_t) = \begin{cases} \sigma_1 \text{ se } S_t = 1 \\ \sigma_2 \text{ se } S_t = 2 \end{cases}$$

e S_t é uma cadeia de Markov homogénea em tempo discreto com espaço de estados $\{1,2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}.$$

▶ Formato

{b,cov,m1,m2,media,var,prob1}=rs_td_2AR(y,b0);

▶ Input

y: vector das observações do processo.

b0: valor inicial de $(\alpha_1, \beta_1, \theta_1, \sigma_1, \alpha_2, \beta_2, \theta_2, \sigma_2, p_{11}, p_{22}, pa)$

(considerar pa=0.5)

▶ Output

b: vector das estimativas de $(\alpha_1, \beta_1, \theta_1, \sigma_1, \alpha_2, \beta_2, \theta_2, \sigma_2, p_{11}, p_{22}, pa)$.

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

m1: vector da média condicional associada ao regime 1 de tipo $(n-12)\times 1$.

m2: vector da média condicional associada ao regime 2 de tipo $(n-12)\times 1$.

media: vector da média condicional do processo de tipo $(n-12)\times 1$.

var: vector da variância condicional do processo de tipo $(n-12)\times 1$.

prob1: vector das probabilidades $P[S_t = 1 | \mathcal{F}_{t-1}]$ estimadas de tipo $(n-12)\times 1$ ($t = 13, ..., n$).

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\rs_td.src.

8.30 rs_td_3

▶ Objectivo

Estimar o seguinte modelo:

$$X_t = \alpha(S_t) + \beta(S_t) X_{t-1} + u_t \quad \text{(i)}$$
$$u_t = \sigma(S_t) \varepsilon_t \quad \text{(ii)}$$

onde, ε_t é i.i.d. com distribuição $N(0,1)$,

$$\alpha(S_t) = \begin{cases} \alpha_1 \; S_t = 1 \\ \alpha_2 \; S_t = 2 \\ \alpha 3 \; S_t = 3 \end{cases} \qquad \beta(S_t) = \begin{cases} \beta_1 \; S_t = 1 \\ \beta_2 \; S_t = 2 \\ \beta_3 \; S_t = 3 \end{cases} \qquad \sigma(S_t) = \begin{cases} \sigma_1 \; S_t = 1 \\ \sigma_2 \; S_t = 2 \\ \sigma_3 \; S_t = 3 \end{cases}$$

e S_t é uma cadeia de Markov homogénea em tempo discreto com espaço de estados $\{1,2,3\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} p_{11} \, p_{12} \, p_{13} \\ p_{21} \, p_{22} \, p_{23} \\ p_{31} \, p_{32} \, p_{33} \end{bmatrix}$$

(notar que os comandos _cml_A e _cml_B são úteis para impor restrições sobre os parâmetros - consultar o manual do GAUSS).

▶ Formato

{beta,f0,gra,cov,retcode,prob1,prob2,prob3,media,var}=rs_td_3(x,b0);

▶ Input

y: vector das observações do processo.

b0: valor inicial de $(\alpha_1, \beta_1, \sigma_1, \alpha_2, \beta_2, \sigma_2, \alpha_3, \beta_3, \sigma_3, p_{11}, p_{21}, p_{22}, p_{33}, pa, pb)$ (considerar pa=0.5,pb=0.5).

▶ Output

b: vector das estimativas de $(\alpha_1, \beta_1, \sigma_1, \alpha_2, \beta_2, \sigma_2, p_{11}, p_{21}, p_{22}, p_{33}, pa, pb)$.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

prob1: vector das probabilidades $P[S_t = 1 | \mathcal{F}_{t-1}]$.

prob2: vector das probabilidades $P[S_t = 2 | \mathcal{F}_{t-1}]$.

prob3: vector das probabilidades $P[S_t = 3|\mathcal{F}_{t-1}]$.

media: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1) \times 1$ onde n é o número de observações.

▶ Library

library cml,est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\rs_td.src.

8.31 rs_td_4

▶ Objectivo

Estimar o seguinte modelo

$$y_t = \beta_1(S_t)\, x_{t1} + \beta_2(S_t)\, x_{t2} + ... + \beta_1(S_t)\, x_{tk} + u_t$$
$$u_t = \sigma(S_t)\, \varepsilon_t$$

onde, ε_t é i.i.d. com distribuição $N(0,1)$,

$$\beta_i(S_t) = \begin{cases} \beta_{i1} \text{ se } S_t = 1 \\ \beta_{i2} \text{ se } S_t = 2 \end{cases} \qquad \sigma(S_t) = \begin{cases} \sigma_1 \text{ se } S_t = 1 \\ \sigma_2 \text{ se } S_t = 2 \end{cases}$$

e S_t é uma cadeia de Markov homogénea em tempo discreto com espaço de estados $\{1,2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}.$$

▶ Formato

{beta,f,g,cov,retcode,prob1,media,var}=rs_td_4(y,x,b0);

▶ Input

y: vector das observações do processo.

x:

b0: se b0=0 ⇒ b0=OLS; caso contrário, b0 vector linha de dimensão $2k+5$

▶ Output

b:

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

var: vector da média condicional de tipo $(n-1)\times 1$ onde n é o número de observações.

prob1: vector das probabilidades $P\left[S_t = 1 | \mathcal{F}_{t-1}\right]$ estimadas de tipo $(n-1)\times 1$ ($t = 2, ..., n$).

▶ Library

library cml,est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\rs_td.src.

8.32 rs_td_5

▶ Purpose

Estimates the model

$$y_t = c\left(S_t\right) + \phi\left(S_t\right) y_{t-1} + u_t$$
$$u_t = \sigma\left(S_t\right)\sqrt{y_{t-1}}\varepsilon_t$$

where, ε_t is i.i.d. with $N\left(0,1\right)$ distribution,

$$c\left(S_t\right) = \begin{cases} c_1 \text{ se } S_t = 1 \\ c_2 \text{ se } S_t = 2 \end{cases} \qquad \phi\left(S_t\right) = \begin{cases} \phi_1 \text{ se } S_t = 1 \\ \phi_2 \text{ se } S_t = 2 \end{cases} \qquad \sigma\left(S_t\right) = \begin{cases} \sigma_1 \text{ se } S_t = 1 \\ \sigma_2 \text{ se } S_t = 2 \end{cases}$$

and S_t is a Markov chain with transition probabilities given by

$$P = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}.$$

▶ Format

{b,f0,g,cov,retcode,prob1,media,var}=rs_td_1(y,b0);

▶ Input

y: observations

b0: starting values of $(\alpha_1, \beta_1, \sigma_1, \alpha_2, \beta_2, \sigma_2, p_{11}, p_{22}, pa)$ (considerar pa=0.5).

▶ **Output**

b: estimates of $(\alpha_1, \sigma_1, \alpha_2, \sigma_2, p_{11}, p_{22}, pa)$.

f0:

gra: *score*

cov:

retcode: convergence. If retcode = 0 convergence is reached

media: conditional mean ($(n-1) \times 1$ vector)

var: conditional variance ($(n-1) \times 1$ vector)

prob1: $(n-1) \times 1$ vector $P\left[S_t = 1 | \mathcal{F}_{t-1}\right]$ $(t = 2, ..., n)$.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\rs_td.src.

8.33 smoothed_probabilities

▶ **Objectivo**

Calcula $P\left(S_t = 1 | \mathcal{F}_T\right)$ para $t = 1, ..., T$, supondo um modelo regime-switching com dois regimes e matriz de probabilidades de transição

$$P = \begin{bmatrix} \alpha_{11} & 1 - \alpha_{11} \\ 1 - \alpha_{22} & \alpha_{22} \end{bmatrix}$$

▶ **Formato**

{ptt, sprob,p1} = smoothed_probabilities(alfa11,alfa22,f1,f2,prob1);

▶ **Input**

alfa11:

alfa22:

f1: vector de tipo $n \times 1$ densidade associada ao regime 1

f2: vector de tipo $n \times 1$ densidade associada ao regime 2

prob1: vector $n \times 1$ $P\left(S_t = 1 | \mathcal{F}_{t-1}\right)$

▶ **Output**

ptt: diagonal de p1

sprob: última coluna de p1

p1:

$$p1 = \begin{bmatrix} P\left(S_1 = 1|\,\mathcal{F}_1\right) & P\left(S_1 = 1|\,\mathcal{F}_2\right) & \cdots & P\left(S_1 = 1|\,\mathcal{F}_n\right) \\ 0 & P\left(S_2 = 1|\,\mathcal{F}_2\right) & \cdots & P\left(S_2 = 1|\,\mathcal{F}_n\right) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P\left(S_n = 1|\,\mathcal{F}_n\right) \end{bmatrix}$$

▶ Library

library est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd.src.

8.34 tStudent_1

▶ Objectivo

Estimar o modelo

$$Y_t = \beta_1 X_{1t} + ... + \beta_k X_{kt} + u_t, \qquad u_t \sim t\left(v\right).$$

(X pode incluir endógenas desfasadas).

▶ Formato

{b,f0,grad,cov,retcode,media}=tStudent_1(gl,y,x);

▶ Input

gl: valor inicial para v

y: vector de tipo $n \times 1$ das observações do processo.

x: matriz de tipo $k \times n$ das variáveis explicativas X (inclui endógenas desfasadas).

▶ Output

b: vector das estimativas.

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: média condicional de Y.

▶ **Observações**

Vector dos valores iniciais de b = estimativas OLS

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\garch.src

8.35 tstudent_garch

▶ **Objectivo**

Estimates the model AR(r)+GARCH(p,q)

$$y_t = \phi_1 y_{t-1} + ... + \phi_r y_{t-r} + \beta_1 x_{1t} + ... + \beta_k x_{kt} + u_t, \qquad u_t = \sigma_t \varepsilon_t, \qquad \varepsilon_t \sim t\,(v)$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + ... + \alpha_q u_{t-q}^2 + \delta_1 \sigma_{t-1}^2 + ... + \delta_p \sigma_{t-p}^2 + \psi_1 g_{1t} + ... + \psi_s g_{st}.$$

▶ **Format**

{b,f0,grad,cov,retcode,media,var}=tstudent_garch(b0,y,des,x,p,q,g);

▶ **Input**

b0: initial value for $(\phi, \beta, \alpha, \delta, \psi, v)$; if b0=0 the procedure selects b0=.1|.1|...|.1|10;

y:

des: order r ($r \geq 0$).

x: $k \times n$ matrix of explanatory variables. If x is a scalar, one assumes that the model does not have explanatory variables.

p: order p ($p > 0$).

q: order q ($q > 0$).

g: $s \times n$ matrix composed by the variables that explain σ_t^2 . If g is a scalar, one assume that there are not such variables.

▶ **Output**

b: estimate of $(\phi, \beta, \alpha, \delta, \psi, v)$

f0: value of likelihood function at maximum divided by the number of observations.

gra: *score*

cov: $\widehat{\text{Cov}\,(b)}$

retcode: if retcode = 0 convergence is reached.

media: Conditional mean of y.

var: Conditional variance of y.

▶ Library

library cml,est_etd;

▶ Fonte

c:\gauss\srcnic\est_etd\garch.src

8.36 var_cond

▶ Objectivo

Estimar não parametricamente a variância condicional $Var\left[X_t\right| X_{t-k}=x]$, $k\geq 1$, a partir das observações $(X_1, X_2, ..., X_n)$. O estimador é

$$v\left(x\right)=\frac{\sum_{t=k+1}^{n} K\left(\frac{X_{t-k}-x}{h}\right) X_t^2}{\sum_{t=k+1}^{n} K\left(\frac{X_{t-k}-x}{h}\right)}-\mu\left(x\right)^2$$

onde, $\mu\left(x\right)$ é a estimativa não paramétrica da média condicional $E\left[X_t\right| X_{t-k}=x]$, $K\left(u\right)=\left(2\pi\right)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h=\left(4/3\right)^{1/5}\hat{\sigma}_X n^{-1/5}$.

▶ Formato

{x,m,v}=var_cond(y,pontos,des);

▶ Input

y: vector de observações $(X_1, X_2, ..., X_n)$ do processo X.

pontos: número de pontos x_i para os quais se calcula $v\left(x_i\right)$. Isto é $i=1,2,...,$pontos. Se pontos = 0, calcula-se $v\left(x_i\right)$ onde x_i assume todos os valores da amostra $(X_1, X_2, ..., X_n)$.

des: valor de k.

▶ Output

x: vector x.

m: vector $\mu\left(x\right)$.

v: vector $v\,(x)$.

▶ **Library**

library est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\nao_p_td.src

8.37 vq_td

▶ **Objectivo**

Estimar o modelo (iv) e (v) juntamente com (i), (ii) ou (iii)

$$X_t = c + \phi_1 X_{t-1} + ... + \phi_r X_{t-r} + \sigma_t \varepsilon_{t,1} \quad \text{(i)}$$

$$\Delta X_t = c + \phi_1 X_{t-1} + ... + \phi_r X_{t-r} + \sigma_t \varepsilon_{t,1} \quad \text{(ii)}$$

$$\Delta X_t = c + \phi_1 \Delta X_{t-1} + ... + \phi_r \Delta X_{t-r} + \sigma_t \varepsilon_{t,1} \quad \text{(iii)}$$

$$\sigma_t^2 = \alpha_0^2 + \sum_{i=1}^{q} \alpha_i \left(X_{t-i} - \mu_{t-i}\right)^2 + \sum_{i=1}^{p} \delta_i \sigma_{t-i}^2 \quad \text{(iv)}$$

$$\mu_t = (1-\lambda)\,\mu_{t-1} + \lambda X_{t-1}. \quad \text{(v)}$$

através do método de máxima verosimilhança.

▶ **Formato**

{b,f0,grad,cov,retcode,media,miu,var}=vq_td(y,r,q,p,b0,prim_dif);

▶ **Input**

y: vector das observações do processo.

r: ordem do processo AR.

q: ordem q do desfasamento $\left(X_{t-i} - \mu_{t-i}\right)^2$

p: ordem AR da variância.

b0: vector dos parâmetros iniciais, b0=(lam,const,fi1,...,fir,a0,a1,...,aq,delta1,...,deltap).

prim_dif: Com prim_dif=1,2 e 3 considera-se respectivamente o modelo (i), (ii) e (iii).

▶ **Output**

b: vector das estimativas de (lam,const,fi1,...,fir,a0,a1,...,aq,delta1,...,deltap,var01,...,var0p).

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional.

miu: vector das estimativas de (v).

var: vector das estimativas de (iv).

▶ Library

library cml, est_etd;

▶ Observações

Comparar com as instruções VQE_TD e VQE_TD1. Com o comando VQ_TD o processo μ não depende de uma v.a. residual.

▶ Fonte

c:\gauss\srcnic\est_etd\vqe_td.src

8.38 vqe_td

▶ Objectivo

Estimar o modelo (i), (iii) e (iv) ou (ii), (iii) e (iv)

$$X_t = c + \phi_1 X_{t-1} + ... + \phi_r X_{t-r} + \sigma_t \varepsilon_{t,1} \tag{i}$$

$$X_t - X_{t-1} = c + \phi_1 X_{t-1} + ... + \phi_r X_{t-r} + \sigma_t \varepsilon_{t,1} \tag{ii}$$

$$\sigma_t^2 = \alpha_0^2 + \sum_{i=1}^{q} \alpha_i \left(X_{t-i} - \mu_{t-i} \right)^2 + \sum_{i=1}^{p} \delta_i \sigma_{t-i}^2 \tag{iii}$$

$$\mu_t = (1 - \lambda)\, \mu_{t-1} + \lambda X_{t-1} + \sigma_v \varepsilon_{t,2}. \tag{iv}$$

através do método da função de verosimilhança simulada:

$$\hat{\theta}_{n,S} = \arg\max_{\theta} \sum_{t}^{n} \log \left[\frac{1}{S} \sum_{h}^{S} f\left(X_t | \, X_{t-1}, ..., X_{t-r}, \mu_{t-1}^{h} \right) \right]$$

onde f é a função densidade de probabilidade normal.

▶ Formato

{b,f0,grad,cov,retcode,media,miu,var}=vqe_td(y,r,q,p,b0,s,seed,prim_dif);

▶ Input

y: vector das observações do processo.

r: ordem do processo AR.

q: ordem q do desfasamento $\left(X_{t-i} - \mu_{t-i}\right)^2$

p: ordem AR da variância.

b0: vector dos parâmetros iniciais, b0=(miu0,sigma,lam,const,fi1,...,fir,a0,a1,...,aq,delta1,...,deltap,var01, ...,var0p).

s: valor S.

prim_dif: se prim_dif=0 considera o modelo (i); no caso contrário considera o modelo (ii)

▶ **Output**

b: vector das estimativas de (miu0,lam,const,fi1,...,fir,a0,a1,...,aq,delta1,...,deltap,var01,...,var0p).

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional.

miu: vector das médias das S trajectórias de μ [equação (iv)] no valor maximizante.

var: vector das média das S trajectórias de σ_t^2 [equação (iii)] no valor maximizante.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\vqe_td.src

8.39 vqe_td1

Igual ao procedimento VQE_TD com excepção de que com VQE_TD1 os parâmetros miu0 e var0 são inicializados, através do cálculo dos dois primeiros momentos do processo relativamente às primeiras observações. O vector de inicialização é agora b0=(lam,sigma,const,fi1,...,fir,a0,a1,...,aq, delta1,...,deltap).

8.40 vqe_td2

▶ Objectivo

Estimar o modelo

$$\Delta Y_t = \beta_1 X_{t1} + ... + \beta_k X_{tk} + \phi_1 \Delta Y_{t-1} + ... + \phi_r \Delta Y_{t-r} + \sigma_t \varepsilon_{t,1}$$

$$\sigma_t^2 = \alpha_0^2 + \sum_{i=1}^{q} \alpha_i \left(Y_{t-i} - \mu_{t-i}\right)^2 + \sum_{i=1}^{p} \delta_i \sigma_{t-i}^2$$

$$\mu_t = \lambda_0 + (1-\lambda)\,\mu_{t-1} + \lambda Y_{t-1}.$$

através do método da função de verosimilhança (assumindo normalidade). Nota: assume-se que $Y_t \sim I(1)$.

▶ Formato

{beta,f0,grad,cov,retcode,media,miu,varVQE}=vqe_td2(y,x,r,q,p,b0);

▶ Input

y: vector das observações do processo.

x: matriz x (devendo-se incluir sempre o vector de **1**'s para o termo independente). Considerar eventualmente as instruções _cml_A e _cml_B se não se quiser o termo independente).

r: ordem do processo AR.

q: ordem q do desfasamento $\left(X_{t-i} - \mu_{t-i}\right)^2$

p: ordem AR da variância.

b0: vector dos parâmetros iniciais. No caso $r = 1, q = 1$ e $p = 1$ b0 é
b0=(beta1,fi1,a0/10^6,alfa1,delta1,lam0,lam1);

▶ Output

b: vector das estimativas de (miu0,lam,const,fi1,...,fir,a0,a1,...,aq,delta1,...,deltap,var01,...,var0p).

f0: média do logaritmo da função de verosimilhança no maximizante.

gra: *score* no maximizante

cov: matriz das estimativas das variâncias-covariâncias de b (de pseudo máxima verosimilhança).

retcode: convergência. Se retcode = 0 convergência atingida.

media: vector da média condicional.

miu: vector $\hat{\mu}$.

var: vector de $\hat{\sigma}_t^2$.

▶ **Library**

library cml,est_etd;

▶ **Fonte**

c:\gauss\srcnic\est_etd\vqe_td.src

9 Expected Time [ET]

9.1 bootstrap_block

▶ Purpose

To generate one sequence of bootstrap observations from a MC based on the regeneration-based bootstrap procedure of Athreya and Fuh (1992) (see also Kreiss and Lahiri, 2012). For Markov chains satisfying the Harris recurrence condition, successive returns to a recurrent state gives a decomposition of the chain into i.i.d. cycles (of random lengths). The regeneration-based bootstrap resamples these i.i.d. cycles to generate the bootstrap observations. Let A be an "accessible atom", in our case $A \in \mathcal{S} = \{1, 2, 3\}$. Define the successive return times to A by

$$\tau_1 = \inf\{t : S_t = A\}, \qquad \tau_{k+1} = \inf\{t : t > \tau_k, S_t = A\}, \ k \geq 1.$$

By strong Markov property, the blocks $B_k = \{S_t : \tau_k + 1 \leq t \leq \tau_{k+1}\}$, $k \geq 1$ are i.i.d. variables in the space $\cup_{k\geq 1}\mathcal{S}^k$. Therefore the regeneration-based bootstrap resamples the collection of blocks $\{B_k : B_k \subset \{S_0, ..., S_{n-1}\}\}$ with replacement to generate the bootstrap observations. Therefore, the size of B_k may de obtained counting the average successive returning times to $S_t = 1$.

▶ Formato

SS=bootstrap_block(S,n_block);

▶ Input

S: vector of observations of the MC

n_Block:

▶ Output

SS: bootstrap observations

▶ Library

library ET;

▶ Source

c:\gauss\srcnic\ET\?

9.2 bootstrap_ET

► Purpose

To calculate confidence intervals for the expected time derived form a rth Markov chain. We consider the regeneration-based bootstrap procedure of Athreya and Fuh (1992) (see also Kreiss and Lahiri, 2012). For Markov chains satisfying the Harris recurrence condition, successive returns to a recurrent state gives a decomposition of the chain into i.i.d. cycles (of random lengths). The regeneration-based bootstrap resamples these i.i.d. cycles to generate the bootstrap observations. Let A be an "accessible atom", in our case $A \in \mathcal{S} = \{1, 2, 3\}$. Define the successive return times to A by

$$\tau_1 = \inf\{t : S_t = A\}, \qquad \tau_{k+1} = \inf\{t : t > \tau_k, S_t = A\}, \ k \geq 1.$$

By strong Markov property, the blocks $B_k = \{S_t : \tau_k + 1 \leq t \leq \tau_{k+1}\}$, $k \geq 1$ are i.i.d. variables in the space $\cup_{k\geq 1}\mathcal{S}^k$. Therefore the regeneration-based bootstrap resamples the collection of blocks $\{B_k : B_k \subset \{S_0, ..., S_{n-1}\}\}$ with replacement to generate the bootstrap observations. Therefore, the size of B_k may de obtained counting the average successive returning times to $S_t = 1$.

This routine needs to run two other routines:

SS=bootstrap_block(S,n_block);

{ET[i,j],S0,se1}=expected_time2(0,SS,x0[j],x1,0,L,0);

► Formato

{media,SE}=bootstrap_ET(S,n_block,r,replicas,x0,x1);

► Input

S: vector of observations of the MC

n_Block:

r: order of the MC

replicas:

x0:

x1:

► Output

media: mean of ET generated by bootstrap

SE: SE of ET generated by bootstrap

► Library

library ET;

► **Source**

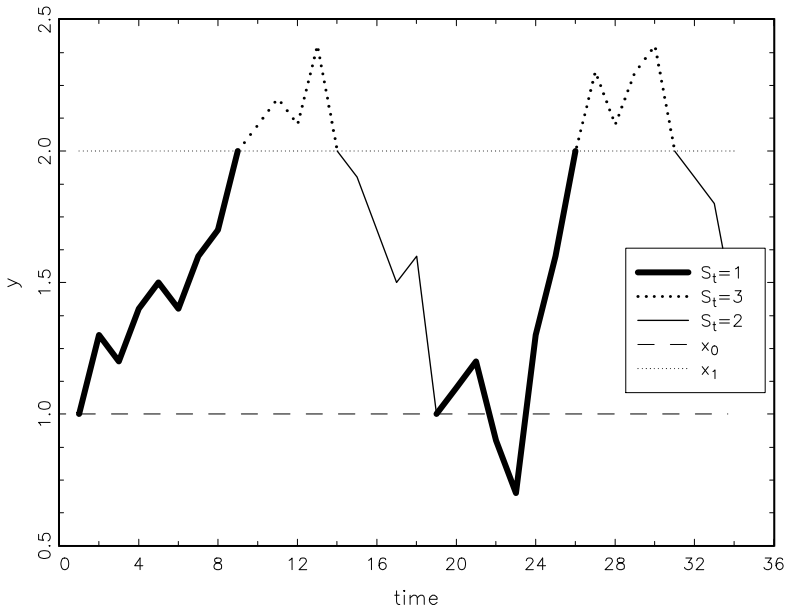
c:\gauss\srcnic\ET\?

9.3 build_seq_123

► Purpose

Given the thresholds x_0 and x_1, the procedure transform y into a sequence of 1,2 and 3 according to the rule

$$S_t = \begin{cases} 1 \text{ if } y_t < x_1, y_{t-1} < x_1, ..., y_{t-k+1} < x_1, y_{t-k} \leq x_0 \\ 2 \text{ if } x_0 < y_t \leq x_1, x_0 < y_{t-1} \leq x_1, ..., x_0 < y_{t-k+1} \leq x_1, y_{t-k} \geq x_1 \\ 3 \text{ otherwise.} \end{cases}$$



► Format

{S,t0}=build_seq_123(y,x0,x1,aparar);

► Input

x0:

x1:

y: vector de observations

aparar: If aparar=1 the procedure trims the last sequence of ones (because $S_t = 1$ could have continued - to avoid the problem of censored data)

► Output

t0: The sequence S starts at $S = 1$ when y_t crosses x_0. That period is t_0

S:

▶ **Library**

library ET;

▶ **Source**

c:\gauss\srcnic\ET\estima.src

9.4 expected_time

▶ **Purpose**

1- To estimate the ET for y_t to cross x_1 given that $y_t = x_0 < x_1$.

2- To estimate the ET for y_t to cross x_1 given that $y_t = x_0 > x_1$.

To decide between 1 and 2 write $x_0 < x_1$ or $x_0 > x_1$, respectively.

The procedure first finds x_0 and then calculate how many periods it takes to cross x_1. After crossing x_1 it finds the next x_0 and the calculation is repeated.

▶ **Formato**

{t0,t1,et}=expected_time(x0,x1,y,mostrar);

▶ **Input**

x0:

x1:

y: vector de observations

mostrar: show t0,t1,et

▶ **Output**

t0: periods where x_0 were crossed

t1: periods where x_1 were first crossed after x_0

ET: expected time (empirical mean of all value t1-t0)

▶ **Library**

library ET;

▶ **Fonte**

c:\gauss\srcnic\ET\estima.src

9.5 expected_time2

▶ **Purpose**

To calculate

$$\mathrm{E}(T)=\sum_{t=1}^{r} t\,(1-p_t)\prod_{i=1}^{t-1}p_i+\left((1-p_r)\prod_{i=1}^{r-1}p_i\right)\frac{p_r\,(1+r-rp_r)}{(1-p_r)^2}$$

This expression reduces to the following formulas:

$$r=1\Rightarrow \mathrm{E}(T)=\frac{1}{1-p_1}=\frac{1}{1-P_{11}}$$
$$r=2\Rightarrow \mathrm{E}(T)=\frac{1+p_1-p_2}{1-p_2},etc.$$

r is the order of the Markov chain.

▶ **Formato**

{ET,S,se1}=expected_time2(y,S,x0,x1,aparar,r,mostrar);

▶ **Input**

y: scalar or $n\times 1$ vector of observations. The case $y=$ scalar means that the S sequence was already obtained and it will be provided in the next input

S: scalar or $n\times 1$ vector of observations the state space may be $\{1,2\}$, or $\{1,2,3\}$, $\{0,1\}$, etc. Must include $\{1\}$. The expected time is calculated in relation to $\{1\}$

x0:

x1:

aparar: If aparar=1 the procedure trims the last sequence of ones (because $S_t=1$ could have continued - to avoid the problem of censored data)

r: order of the Markov chain (maximum = 5)

mostrar: If mostrar =1 it presents some results related to the case $L=1$

▶ **Output**

ET:

S:

SE1: SE associated with $L=1$. It is based on the result $\sqrt{n}\left(\widehat{\mathrm{E}(T)}-\mathrm{E}(T)\right)\xrightarrow{d} N\left(0,P_{11}/\left((1-P_{11})^3\pi_1\right)\right)$

▶ **Library**

library ET;

▶ **Source**

c:\gauss\srcnic\ET\estima.src

9.6 expected_time3

▶ Purpose

Same as expected_time2, but faster. The output is just ET.

▶ Formato

ET=expected_time3(S,L);

▶ Input

S: scalar or $n \times 1$ vector of observations the state space may be $\{1, 2\}$, or $\{1, 2, 3\}$, $\{0, 1\}$, etc. Must include $\{1\}$. The expected time is calculated in relation to $\{1\}$.

L: order of the Markov chain (maximum = 5)

▶ Output

ET:

▶ Library

library ET;

▶ Source

c:\gauss\srcnic\ET\estima.src

10 Histogram Time Series [hts]

10.1 bins_same_width

▶ Purpose

Builds an equivalent histogram where the bins have the same width (the output generaly creates different bins).

▶ Format

{l,u,pr}=bins_same_width(xL,xU,p,bars);

▶ Input

xL: $m \times 1$ matrix. Lower bounds

xU: $m \times 1$ matrix. Upper bounds

p: $m \times 1$ probabilities associated with each bin

bars: number of bins

▶ Output

l: $bars \times 1$ Lower bounds

u $bars \times 1$ Upper bounds

pr: $bars \times 1$ probabilities

▶ Remarks

RV_CTS_1 procedure calls CTS_1

▶ Library

library hts;

▶ Source

c:\gauss\srcnic\hts\basic1.src

10.2 barycentric_histogram_v1

▶ Purpose

Builds the barycentric histogram

▶ Format

{lB,uB,p,z}=barycentric_histogram_v1(xL,xU,pr,weights);

▶ Input

xL: $n \times m$ matrix. Lower bounds

xU: $n \times m$ matrix. Upper bounds

pr: $n \times m$ matrix. Probabilities

weights: weights of the n histograms

▶ Output

lB: $m' \times 1$ Lower bounds of the barycentric histogram

uB: $m' \times 1$ Upper bounds of the barycentric histogram

p: $m' \times 1$ probabilities associated with the barycentric histogram

z: cumulative distribution: 0 ~p[1] ~(p[1]+p[2]) ~etc.

▶ Remarks

This procedure calls the following other procedures: obtain_z_from_n_histograms and standardise_intervals.

▶ Library

library hts;

▶ Source

c:\gauss\srcnic\hts\basic1.src

▶ Example

```
xL={1 2 3,
 11 12 13};
xU={2 3 4,
 12 13 14};
pr={0.7 .2 .1,
 0.1 .2 .7};
weights=.5|.5;
```

Output:

print `lb|ub;`

```
 6.0000000 6.5714286 7.2142857 7.7857143 8.4285714
 6.5714286 7.2142857 7.7857143 8.4285714 9.0000000
```

print p;

```
 0.10000000 0.20000000 0.40000000 0.20000000 0.10000000
```

10.3 barycentric_histogram_v2

▶ Purpose

Build the barycentric histogram

▶ Format

{lB,uB,p,z}=barycentric_histogram_v1(xL,xU,pr,weights,cumprob);

▶ Input

xL: $n \times m$ matrix. Lower bounds

xU: $n \times m$ matrix. Upper bounds

pr: $n \times m$ matrix. Probabilities

weights: weights of the n histograms

cumprob:

▶ Output

lB: $m' \times 1$ Lower bounds of the barycentric histogram

uB: $m' \times 1$ Upper bounds of the barycentric histogram

p: $m' \times 1$ probabilities associated with the barycentric histogram

▶ Remarks

This procedure calls the following other procedures: obtain_z_from_n_histograms and standardise_intervals.

▶ Library

library hts;

▶ Source

c:\gauss\srcnic\hts\basic1.src

10.4 cdf_histogram

▶ Purpose

See expression 4 of Arroyo et al. (2009).

▶ Format

p=cdf_histogram(x,xL,xU,pX);

▶ Input

x: scalar $x \in \mathbb{R}$

xL: $1 \times m$ vector. Lower bounds

xU: $1 \times m$ vector. Upper bounds

pX: $1 \times m$ vector. Probabilities

▶ **Output**

p:

▶ **Library**

library hts;

▶ **Source**

c:\gauss\srcnic\hts\basic1.src

10.5 inverse_cdf_histogram

▶ **Purpose**

See formula A.2 of Arroyo et al. (2009).

▶ **Format**

inv_CDF=inverse_cdf_histogram(p,xL,xU,pX);

▶ **Input**

p: scalar $p \in [0, 1]$

xL: $1 \times m$ vector. Lower bounds

xU: $1 \times m$ vector. Upper bounds

pX: $1 \times m$ vector. Probabilities

▶ **Output**

inv_CDF:

▶ **Library**

library hts;

▶ **Source**

c:\gauss\srcnic\hts\basic1.src

▶ **Example**

```
xL={10 20 30};

xU={20 30 40};

pX={0.7 .2 .1};
```

p=0;

```
inv_CDF=inverse_cdf_histogram(p,xL,xU,pX);
```

print `inv_CDF;`

Output:

10

10.6 kernel_density_HTS

▶ **Purpose**

Calculates

$$\begin{aligned}\tilde{f}_{n+1}(\xi) &= \sum_{t=1}^{n}\sum_{i=1}^{m}\frac{1}{mh_t}K\left(\frac{X_{ti}-\xi}{h_t}\right)\frac{\alpha^{n-t}(1-\alpha)}{1-\alpha^n}\\ &= \sum_{t=1}^{n}\hat{f}_t(\xi)\,\frac{\alpha^{n-t}(1-\alpha)}{1-\alpha^n}\end{aligned}$$

▶ **Format**

p=kernel_density_HTS(y,x0,alfa,hh);

▶ **Input**

y: matrix $n \times m$ (n observations)

x0: scalar

alfa: α

hh: scalar $\bar{h}$ in the expression $h_t = \bar{h}(4/3)\hat{}.2 * stdc(y[t,.]\prime) * m^{-1/5}$;

▶ **Output**

p: scalar $\tilde{f}_{n+1}(\xi)$

▶ **Library**

library hts;

▶ **Source**

c:\gauss\srcnic\hts\kernel_density.src

10.7 Mallows_distance

▶ **Purpose**

To measure the distance between the histograms {xL[1,.],xU[1,.],pr[1,.]} and {xL[2,.],xU[2,.],pr[2,.]}

▶ **Format**

d=Mallows_distance(xL,xU,pr);

▶ **Input**

xL: $2 \times m$ vector. Lower bounds

xU: $2 \times m$ vector. Upper bounds

pr: $2 \times m$ vector. Probabilities

▶ **Output**

d:

▶ **Library**

library hts;

▶ **Source**

c:\gauss\srcnic\hts\basic1.src

10.8 Mallows_distance_n

▶ **Purpose**

To measure the distance between n pairs of histograms

$$xL = \begin{bmatrix} 1^{st}hist \\ 2^{nd}hist \\ ... \\ n^{th}hist \end{bmatrix} \text{ (lower bounds)} \qquad xU = \begin{bmatrix} 1^{st}hist \\ 2^{nd}hist \\ ... \\ n^{th}hist \end{bmatrix} \text{ (upper bounds)}$$

$$yL = \begin{bmatrix} 1^{st}hist \\ 2^{nd}hist \\ ... \\ n^{th}hist \end{bmatrix} \text{ (lower bounds)} \qquad yU = \begin{bmatrix} 1^{st}hist \\ 2^{nd}hist \\ ... \\ n^{th}hist \end{bmatrix} \text{ (upper bounds)}$$

It is measured the distance between

$$h_{X1} = \{xL[1,.], xU[1,.]\} \text{ and } h_{Y1} = \{yL[1,.], yU[1,.]\}$$

then, between

$$h_{X2} = \{xL[2,.], xU[2,.]\} \text{ and } h_{Y2} = \{yL[2,.], yU[2,.]\}$$

and so on.

▶ Format

{d,dist}=Mallows_distance_n(xL,xU,pX,yL,yU,pY);

▶ Input

xL: $n \times m$ vector. Lower bounds

xU: $n \times m$ vector. Upper bounds

pX: $n \times m$ vector. Probabilities

yL: $n \times m$ vector. Lower bounds

yU: $n \times m$ vector. Upper bounds

pY: $n \times m$ vector. Probabilities

▶ Output

d: $n \times 1$ gives

$$\begin{bmatrix} D(h_{X1}, h_{Y1}) \\ \vdots \\ D(h_{Xn}, h_{Yn}) \end{bmatrix}$$

where $D(h, g)$ is the Mallows distance between f and g

dist: sums up all the distances, i.e., gives $\sum_{i=1}^{n} D(h_{Xi}, h_{Yi})$.

▶ Library

library hts;

▶ Source

c:\gauss\srcnic\hts\basic1.src

10.9 obtain_z_from_n_histograms

▶ **Purpose**

See the example

▶ **Format**

z=obtain_z_from_n_histograms(p);

▶ **Input**

p: $n \times m$ matrix. Probabilities

▶ **Output**

z: cumulative distribution: 0 ~p[1] ~(p[1]+p[2]) ~etc.

▶ **Library**

library hts;

▶ **Source**

c:\gauss\srcnic\hts\basic1.src

▶ **Example**

pr={0.7 .2 .1,

0.1 .2 .7};

First step

$$a = \underbrace{0|0.7|0.9|1}_{\text{cum 1st dist.}} \underbrace{|0|0.1|.3|1}_{\text{cum 2nd dist.}}$$

Second step: sort a :

$$0|0|0.1|0.3|0.7|0.9|1|1$$

Finally: eliminate the repetions

10.10 standardise_intervals

▶ **Purpose**

See the example

▶ **Format**

{xL1,xU1,p}=standardise_intervals(xL,xU,piX,z);

▶ Input

xL: $1 \times m$ vector. Lower bounds

xU: $1 \times m$ vector. Upper bounds

piX: $1 \times m$ vector. Probabilities

z: $k \times 1$ cumulative distribution function

▶ Output

xL1: $1 \times k$ vector. Lower bounds

xU1: $1 \times k$ vector. Upper bounds

p: $1 \times k$

▶ Library

library hts;

▶ Source

c:\gauss\srcnic\hts\basic1.src

▶ Example

Suppose you are given the cumulative distribution $z = 0|.1|.3|.4|.9|1$

>From the original histogram {XL,xU,piX}, the issue is to define 5 bins (since z is a 5×1 vector) with cumulative distribution given by z. Obviously the width of the bins will be defined according to the z vector.

This procedure allows to compare two or more histograms since not only we have the same number of bins across all histograms, but also the probabilities associated with each bin are the same.

```
xL={1 2 3,
 11 12 13};
xU={2 3 4,
 12 13 14};
p={0.7 .2 .1,
 0.1 .2 .7};
cls;
```

```
z=obtain_z_from_n_histograms(p);
print "z";;z;
print;
{l,u,p1}=standardise_intervals(xL[1,.],xU[1,.],p[1,.],z);
l|u;
{l,u,p2}=standardise_intervals(xL[2,.],xU[2,.],p[2,.],z);
l|u;
```

Output:

```
z
 0.00000000
 0.10000000
 0.30000000
 0.70000000
 0.90000000
 1.0000000
l|u
 1.0000000 1.1428571 1.4285714 2.0000000 3.0000000
 1.1428571 1.4285714 2.0000000 3.0000000 4.0000000
l|u
 11.000000 12.000000 13.000000 13.571429 13.857143
 12.000000 13.000000 13.571429 13.857143 14.000000
```

11 QORegression [qor]

11.1 QOR_II

▶ **Purpose**

- To estimate $\mathrm{E}(\tau|\,\mathbf{x}=\mathbf{x}_0)$, where $\tau := F(y)$ is the CDF of y;
- To provide confidence intervals for $\mathrm{E}(\tau|\,\mathbf{x}=\mathbf{x}_0)$.

One assumes

$$\begin{aligned} y_i &= \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i \\ F(y_i) &= F\left(\mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i\right). \end{aligned}$$

Two estimators are offered for $\mathrm{E}(\tau|\,\mathbf{x}=\mathbf{x}_0)$:

1. $\hat{\mu}_1 = \frac{1}{S_1}\sum_{j=1}^{S_1} F_n\left(\mathbf{x}_0'\hat{\boldsymbol{\beta}} + \varepsilon_j^*\right)$, where $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, ε_j^* is obtained by resampling (bootstrap) $\{\hat{\varepsilon}_1, ..., \hat{\varepsilon}_n\}$, $\hat{\varepsilon}_i = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}$, and
2. $\hat{\mu}_2 = \frac{1}{S_1}\sum_{j=1}^{S_1}\left(\frac{1}{S_2}\sum_{i=1}^{S_2}\hat{\theta}_i\right)$, where $\hat{\boldsymbol{\beta}}^{(i)} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}^{(i)}$, $\hat{\theta}_i = F_n\left(\mathbf{x}_0'\hat{\boldsymbol{\beta}}^{(i)} + \varepsilon_j^*\right)$, $\left\{\mathbf{y}^{(1)}, ..., \mathbf{y}^{(S_1)}\right\}$ are generated according to $\mathbf{y}^{(i)} = \mathbf{X}\hat{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}^{**}$, and $\boldsymbol{\varepsilon}^{**}$ is obtained by resampling the residuals, keeping the values $\varepsilon_j^{*\prime}s$ fixed.

The confidence intervals are obtained from the sequence

$$\left\{\hat{\theta}_1, ..., \hat{\theta}_{S_2}\right\}$$

etc. etc.

▶ **Format**

{miu1, miu2, ci}= QOR_II(y,x,x0,replicas1,replicas2,conf_level,shows);

▶ **Input**

y: $n \times 1$

x: $n \times K$

x0: $L \times 1$

replicas1: scalar S_1

replicas2: scalar S_2

conf_level: confidence level $\in [0, 1)$. If this level is zero $\hat{\mu}_2$ is not computed and the procedure runs much faster

shows: if shows=1 some graphs are presented.

▶ **Output**

miu1: $\hat{\mu}_1$

miu2: $\hat{\mu}_2$

CI: $L \times 2$ matrix (confidence interval)

▶ **Remarks**

▶ **Library**

12 Portfolio [portf]

12.1 portfolio_01

▶ **Purpose**

▶ **Format**

{w,varP,miuP,miu_min,var_min,wT,miuT,varT,s}=portfolio_01(r,miuf,miu,H,pontos);

▶ **Purpose**

▶ **Input**

r: m time-series of returns ($n \times m$ matrix)

miuf: risk-free return. If miuf=0 portfolio doe not involve the risk-free return

miu: $m \times 1$ (mean of returns). If h=0 the unconditional mean is calculated.

H: $m \times m$ (variance-covariance matrix of returns). If h=0 the unconditional variance-covariance is calculated.

pontos: number of points to be used to build the efficient frontier

▶ **Output**

w: matriz de pesos de ordem pontos$\times$ n^o de activos (linha i: pesos dos vários activos associados à rendibilidade miuP[i])

varp: vector pontos $\times$ 1. Variância da carteira associada às rendibilidade miuP

miuP: miuP=seqa(minc(miu),(maxc(miu)-minc(miu))/(pontos-1),pontos);

miu_min: rendibilidade associada à variância minima do portfolio

var_min: variância mínima do portfolio

wT: pesos do *tangency portfolio, T*

miuT: μ_T

varT: σ_T^2

s: racio de Sharp, $s = \frac{\mu_T - \mu_f}{\sigma_T}$

▶ **Remarks**

Se _mostrar=1 (default) mostra os graficos.

▶ **Library**

library portf, pgraph;

13 Realized Volatility [realvol]

13.1 CTS_1

▶ Purpose

Defines equidistant time periods from one day tick-by-tick data, and gets the associated prices.

▶ Format

{p,s}=CTS_1(z,interval);

▶ Input

z: $n \times 4$ matrix where: first column of z: prices; second column: hours; third column: minutes; fourth column: seconds

interval: interval in seconds between two consecutive observations.

▶ Output

p: new time series built from z, with smaller sampling frequency than z.

s: time in seconds associated with p.

▶ Remarks

RV_CTS_1 procedure calls CTS_1

▶ Library

library realvol;

13.2 CTS_1

▶ Purpose

Define equidistant time periods and gets the associated prices. Adequate for one day of observations. The original observations must be also equidistant. This procedure is much quicker than RV_CTS_1.

▶ Format

p=CTS_2(z,k);

▶ Input

z: equidistant time prices (for example every 15 seconds).

k: integer that governs the sampling frequency of the new time series. See remarks below.

▶ Output

p: new time series built from z, according to the parameter m, with smaller sampling frequency than z. See remarks below.

▶ Remarks

To illustrate the procedure suppose that $z = (z_1, z_2, ...)$. If $m = 1$ then $p = z$. If $m = 2$ then $p = (z_1, z_3, ...)$. If $m = 3$ then $p = (z_1, z_4, ...)$.

▶ Library

library realvol;

13.3 dummies_int_rates

▶ Purpose

Creates a set of dummies that capture the following events: maintenance period, Governing Council, week days, end of month and end of year.

▶ Format

{d0,d1,d2,d3,d4,d5,d6,gc,mp_before,mp_after,tuesday_mp, end_month,end_year, monday, tuesday, wednesday,thursday, friday}

=dummies_int_rates(dmy,gc_dates);

▶ Input

dmy: $n \times k$ vector ($k \geq 4$). It should be provided the following information in the first four columns

$$day \quad month \quad year \quad weekday$$

where $day = 1, ..., 31$, $month = 1, ..., 12$, $year = 2003$ (for example), $weekday = 2, 3, 4, 5, 6$. $Weekday = 2 =$ Monday, $Weekday = 3 = Tuesday$, etc.

gc_dates: $j \times 3$ matrix where j is the number of governing council meetings. The governing council meetings dates from 14/12/2000 to 4/9/2003 are:

gc_dates={14 12 2000, 1 2 2001, 1 3 2001, 11 4 2001, 10 5 2001, 7 6 2001, 5 7 2001, 2 8 2001, 13 9 2001, 11 10 2001,

8 11 2001, 6 12 2001, 3 1 2002, 7 2 2002, 4 3 2002, 2 4 2002, 6 6 2002, 4 7 2002, 1 8 2002, 12 9 2002, 10 10 2002,

7 11 2002, 5 12 2002, 9 1 2003, 6 2 2003, 6 3 2003, 3 4 2003, 8 5 2003, 5 6 2003, 10 7 2003, 4 9 2003};

Note: you can just copy this instruction into your GAUSS program (please confirm the dates).

▶ **Output**

d0,d1,...,d6:

$$d0_t = \begin{cases} 1\, t = 23 \\ 0\, t \neq 23 \end{cases}, \quad d1_t = \begin{cases} 1\, t = 22 \\ 0\, t \neq 22 \end{cases}, ..., d6_t = \begin{cases} 1\, t = 17 \\ 0\, t \neq 17 \end{cases}$$

For example, $d4$ = four days before the end of maintenance period

gc:

$$gc_t = \begin{cases} 1\, t \text{ coincides with a GC meeting} \\ 0 \qquad\qquad \text{other cases} \end{cases}$$

mp_before:

$$mp\_before_t = \begin{cases} 1\, t = 19 \text{ or } 20 \text{ or } ... \text{ or } 23 \\ 0 \qquad\qquad \text{other cases} \end{cases}$$

mp_after:

$$mp\_after_t = \begin{cases} 1\, t = 24 \text{ or } 25 \text{ or } ... \text{ or } 28 \\ 0 \qquad\qquad \text{other cases} \end{cases}$$

tuesday_mp:

$$tuesday\_mp = \begin{cases} 1\, t \text{ coincides with a day between last Tuesday before the 23th and the 23th} \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{other cases} \end{cases}$$

end_month, end_year: Obvious. End_year is only the event 31/December. It is easy to alter the procedure in order to define other days around the 31/December.

monday, tuesday, wednesday,thursday, friday: Obvious

▶ **Remarks**

After running the program it is useful to define the last day of the maintenance period that falls on Saturday or Sunday. To do that consider the instruction

friday1=friday.*(d1+d2);

▶ **Library**

library realvol;

▶ **Source**

c:\gauss\srcnic\realvol\useful.src

13.4 get_time_series_CTS_1

▶ **Purpose**

Obtains equidistant log prices with a discretization step of m seconds

▶ **Format**

z=get_time_series_CTS_1(first_year,last_year,interval,x);

▶ **Input**

first year:

last year:

interval: intraday interval between observations from which the realized volatility is calculated (m).

x: tick-by-tick data (raw data)

▶ **Output**

z: $n \times 4$ matrix where:

first column of z: year

second column of z: month

third column of z: day

fourth column of z: $\log P_{t_i}$, such that $t_i - t_{i-1} = m$ seconds, $i = 1, ..., n$.

▶ **Library**

library realvol;

13.5 out_detection

▶ **Purpose**

This is an **ad-hoc** procedure to detect outliers in a time series under regime switching structure. The procedure detects an outlier if $|e_{t,h}| > k$, where

$$e_{t,h} = \frac{X_t - m_{t,h}}{s},\ t = 1, 2, ..., T$$

$$m_{t,h} = \begin{cases} \frac{X_1 + ... + X_h}{h} & t = 1, 2, ..., h \\ \frac{X_{t-h} + ... + X_t + ... + X_{t+h}}{2h+1} & t = h, h+1, ..., T-h \\ \frac{X_{T-h+1} + ... + X_T}{h} & t = T-h+1, T-h+2..., T \end{cases}$$

$s =$ standard deviation of the data $\left\{X_{(.025)}, ..., X_{(.975)}\right\}$

where $X_{(q)}$ is the quantile of order q

▶ **Format**

y=out_detection(x,h,k);

▶ **Input**

x: $n \times 1$ vector of observations

h:

k:

▶ **Output**

y: x sequence cleaned from outliers

▶ **Observation**

(1) The procedure doesn't automatically eliminate the outliers. The user is asked about what to do with them. The program requests input from the keyboard (-999 interrupt the execution)

(2) An improvement of this procedure consist in evaluating the hypothesis H_0 : $\omega_i = 0$ in the regression

$$e_{t,h} = c + \omega_i I_{\{t=i\}} + \varepsilon_{i,t}$$

where $I_{\{t=i\}} = 1$ if $t = i$, for different values of i (for example, $i = h, h+1, ..., T-h$).

▶ **Library**

Library realvol;

▶ **Source**

c:\gauss40\srcnic\useful.src

13.6 real_covariance

▶ **Purpose**

Estimates the realized variance, covariance and correlations from a set of time series.

▶ **Format**

{rcov,rcorr,r_h,dmy_h,r_standardized,rsd}=real_covariance(m,freq,x,dmy,print_graphs,rvz,q);

▶ **Input**

m: m is the number of observations in each period (see REAL_VOLATILITY procedure)

freq: see REAL_VOLATILITY

x: $n \times k$ matrix of observations (k times series)

dmy: $n \times c$ matrix of date information. For instance, a row of the dmy matrix can be like $\begin{bmatrix} 25 & 9 & 2003 & 5 \end{bmatrix}$ which identifies the date 25/9/2003 (Thursday =5)

print_graphs if equals 1 then a set of figures are displayed

rvz: if equals 1 then all the periods in which the realized volatility of the first time series is zero are removed. The procedure gives information on the removed observations (rvz = 1 is useful if one wants to compute the log volatility. Moreover it can help to identify holidays). In the case of interest rates it is advisable to consider as the first time series the overnight rate (first column of x).

q: 2×1 vector. It trims the vector of quadratic variation. If q=0|1; the procedure considers all quadratic variations.

▶ Output

rcov: $(n/m) \times ((1+k)k/2)$ matrix of realized (e.g. daily) variance-covariance (the procedure explains the sequence of columns)

rcorr: $(n/m) \times ((1+k)k/2-k)$ matrix of realized (e.g. daily) correlations (the procedure explains the sequence of columns)

r_h: $(n/m) \times k$ matrix of (e.g. daily interest rates) observations

dmy_h: $(n/m) \times k'$ matrix. It gives the date associated with the rcov estimates

r_standardized: $(n/m) \times k$ matrix of (e.g. daily) observations standardized by the realized standard deviation

rsd: $(n/m) \times k$ matrix of realized standard deviation (associated with the k time series).

▶ Remarks

In principle, the ratio n/m should be an integer number. For example, if the step of discretization of the original data is 5 minutes and $m = 108$, the procedure calculates daily realized covariance and variance ((9 hours $\times$ 60)/5=108). In the last day, if we do not have 9 hours of observations the ratio n/m won't be an integer number. In this case the procedure eliminates all the observations corresponding to the last day. Specifically, the procedure will consider the first $n-d$ observations where $d = n - int\,(n/m) \times m$ and $int\,(x)$ represents the integer of x. **To obtain reliable results it is absolutely necessary that all periods (with the possible exception of the last one) have the same number of observations. For example, with 5 minutes intraday observations and** $m = 108$ **all days should have 108 observations. With** $m = 540$ (180×5) **all weeks should have 5 days of observations (if you have eliminated at least one holiday then the REAL_VOL_WMY procedure shoud be ran instead, in order to obtain weekly or monthly realized volatility).**

▶ Library

library realvol, pgraph;

▶ Source

c:\gauss\srcnic\realvol\realvol.src

13.7 real_vol_01

▶ Objectivo

Estimates the realized variance, covariance and correlations from a set of time series.

▶ Formato

{data,rv}=real_vol_01(t,y);

▶ Input

t: vector de dimensão $n \times 1$ com datas repetidas (a realized volatility é calculada para as datas repetidas)

y: matriz $n \times k$

▶ Output

data: vector de dimensão $m \times 1$, $m < n$

rv: matriz de dimensão $m \times k'$, onde $k' = k\,(k+1)\,/2$

▶ Library

library realvol;

▶ Fonte

c:\gauss\srcnic\realvol\realvol.src

13.8 real_vol_WMY

▶ Purpose

Aggregates high frequency realized volatility to weekly, monthly and annual realized volatility.

▶ Format

{date1,r1,y}=real_vol_WMY(date,r,x,option);

▶ Input

date: $n \times 5$ vector. It should be provided the following information

$$day \quad month \quad year \quad weekday \quad week\_in\_the\_year$$

where $day = 1, ..., 31$, $month = 1, ..., 12$, $year = 2003$ (for example), $weekday = 2, 3, 4, 5, 6$. $Weekday = 2 =$ Monday, $Weekday = 3 = Tuesday$, etc $week\_in\_the\_year = 1, 2, ...53$

Note: the weekday and week of the year are easily created trough the EXCEL. See remark (1)

r: $n \times k$ matrix of interest rate observations. See remark (2)

x: $n \times k$ matrix of high frequency realized volatility

option: scalar.

option	Computes
1	weekly realized volatility
2	monthly realized volatility
3	annual realized volatility

▶ Output

date1: $n_0 \times k$ matrix of date associated with y ($n_0 < n$)

r1: $n_0 \times k$ matrix of weekly, monthly or annual interest rates ($n_0 < n$)

y: $n_0 \times k$ matrix of weekly, monthly or annual realized volatility ($n_0 < n$)

▶ Remarks

(1) The date matrix must have five columns. If one wants to pass from daily realized volatility to (say) weekly realized volatility only the 5th column of matrix date is necessary. Thus, one can attribute arbitrary values to the first four columns. However, it is advisable to supply complete date information since one can obtain the date associated with the calculated realized volatility (output "date1")

(2) The r input is not strictly necessary. We can supply a $n \times 1$ vector of arbitrary values. However, it is advisable to supply correct information since one can obtain the weekly, monthly or annual interest rate associated with the selected frequency (output "r1") (for example, with option = 1 one obtains weekly interest rates).

(3) The procedure can also compute daily realized volatility from high frequency interest rates if one sets option = 4 and if the matrix x represents quadratic variations, that is

$$x = \begin{bmatrix} (r_{1,2} - r_{1,1})^2 & \cdots & (r_{N,2} - r_{N,1})^2 \\ (r_{1,3} - r_{1,2})^2 & \cdots & (r_{N,3} - r_{N,2})^2 \\ \vdots & & \vdots \\ (r_{1,n} - r_{1,n-1})^2 & \cdots & (r_{N,n} - r_{N,n-1})^2 \end{bmatrix}$$

where $r_{i,t}$ is the ith interest rates at time t.

▶ Library

library realvol;

▶ Source

c:\gauss\srcnic\realvol\realvol.src

13.9 real_volatility

▶ Purpose

Estimates the realized volatility of an univariate time series. Consider the following time series (for simplicity, we calculate realized daily volatility)

$$
\begin{array}{ll}
X_{\frac{1}{m}\Delta} & \\
X_{\frac{2}{m}\Delta} & \\
\vdots & \\
X_{\frac{m}{m}\Delta} = X_{\Delta} & \text{1th day} \\
X_{\Delta+\frac{1}{m}\Delta} & \\
X_{\Delta+\frac{2}{m}\Delta} & \\
\vdots & \\
X_{2\Delta} & \text{2th day} \\
\vdots &
\end{array}
$$

m is the number of observations in each day.

If the input "freq"is 1 then the realized volatility of day $j+1$ is computed as follows

$$\sum_{i=2}^{m} \left(X_{j\Delta+\frac{i}{m}\Delta} - X_{j\Delta+\frac{i-1}{m}\Delta} \right)^2.$$

In the case freq=2 realized volatility of day $j+1$ is computed as follows

$$\sum_{i=2}^{\left[\frac{m-1}{2}\right]} \left(X_{j\Delta+\frac{2i-1}{m}\Delta} - X_{j\Delta+\frac{2(i-1)-1}{m}\Delta} \right)^2.$$

For example, if the original time series has a step of discretization of 5 minutes then freq=1 means that the realized daily volatility is based on 5 minutes intraday observations. If freq=2 the realized daily volatility is based on 10 minutes intraday observations, and so on.

All periods where the realized volatility is zero are removed.

▶ Format

{rv,conf_int_rv,log_rv,conf_int_log_rv}=real_volatility(m,freq,x,print_ graphs);

▶ Input

m: m is the number of observations in each period. Suppose that we have 5 minutes intra day observations. Then:

$m = 12$ (60/5=12) $\rightarrow$ computes hourly realized volatility

$m = 108$ ((9 hours $\times$ 60)/5=108) $\rightarrow$ computes daily realized volatility

etc. To obtain reliable results it is absolutely necessary that all periods have the same number of observations (see "Remarks in the next procedure").

freq: see "purpose"above

x: $n \times 1$ vector of observations

print_graphs if equals 1 a set of figures are displayed

▶ Output

rv: $n/m \times 1$ vector of realized (e.g. daily) variance

conf_int_rv: 95% intervals confidence for realized (e.g. daily) volatility (see Barndorff-Nielsen and Shephard, 2002)

log_rv: $n/m \times 1$ vector of realized (e.g. daily) log variance

conf_int_log_rv: 95% intervals confidence for log realized (e.g. daily) volatility (see Barndorff-Nielsen and Shephard, 2002)

▶ Library

library realvol,pgraph;

▶ Source

c:\gauss\srcnic\realvol\realvol.src

13.10 RV_CTS_1

▶ Purpose

Calculates the realized volatility $RV_t^{(m)}$ and the *adjusted* $RV_t^{*(m)}$ based on m seconds (interval) where t goes from "first year" to "last year". The prices are obtained from the tick-by-tick price data. We use the *previous-tick* method, i.e the price at time $\tau \in [a, b)$ is p_a

▶ Format

z=RV_CTS_1(first_year,last_year,interval,x);

▶ Input

first year:

last year:

interval: interval in seconds between consecutive observations from which the realized volatility is calculated.

x: tick-by-tick data (raw data).

▶ Output

z: $n \times 5$ matrix where: first column of z: year; second column of z: month; third column of z: day; fourth column of z: realized volatility $RV_t^{(m)}$; fifth column of z: realized volatility $RV_t^{*(m)}$

▶ **Remarks**

This procedure calls the procedure CTS_1. It is very time-consuming.

▶ **Library**

library realvol;

13.11 RV_CTS_2

▶ **Purpose**

Calculates the realized volatility $RV_t^{(m)}$ and the *adjusted* $RV_t^{*(m)}$ based on equidistant observations.

▶ **Format**

z=RV_CTS_2(first_year,last_year,k,x);

▶ **Input**

first year:

last year:

k: integer that governs the sampling frequency of the new time series. See remarks below.

x: raw data with equidistant observations.

▶ **Output**

z: $n \times 5$ matrix where: first column of z: year; second column of z: month; third column of z: day; fourth column of z: realized volatility $RV_t^{(m)}$; fifth column of z: realized volatility $RV_t^{*(m)}$

▶ **Remarks**

Needs the procedure CTS_2. If the original time series has a step of discretization of 15 seconds, then $k = 1$ means that the realized daily volatility is based on 15 seconds intraday observations. In general, the realized daily volatility is based on $\Delta \times k$ seconds intraday observations, where Δ is the step of discretization of the original observations. This procedure is much quicker than RV_CTS_1.

▶ **Library**

library realvol;

14 Simula Equações Diferenciais Estocásticas [sim_ede]

14.1 cir

▶ Objectivo

Simular uma trajectória da EDE

$$dX_t = \beta\left(\tau - X_t\right)dt + \sigma dW_t, \qquad X_0 = x_0$$

através das probabilidades de transição.

▶ Formato

y =CIR(beta,tau,sigma,n,d,y0,seed);

▶ Input

beta:

tau: parâmetro associado ao coeficiente de difusão.

sigma:

n: número de observações a simular.

d: Δ, intervalo entre duas observações consecutivas, constante.

y0: valor inicial do processo.

seed: se seed=0 não fixa a trajectória.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.2 euler

▶ Objectivo

Simular uma EDE do tipo $dX_t = a\left(X_t;\theta_1\right)dt + b\left(X_t;\theta_2\right)dW_t$ através do esquema de Euler.

▶ Formato

y = euler (&f1,&f2,par1,par2,y0,d,n);

► **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

► **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

► **Library**

library sim_ede;

► **Observações**

A maior parte das especificações de interesse dos coeficientes infinitesimais a e b estão já escritas (ver neste apêndice "Funções" em "Utilidades"). Se se pretender escrever uma nova especificação para a ou b, por exemplo, $a(x;\theta) = \theta_1 + \theta_2 \log x$, devem-se escrever as seguintes instruções

```
proc fun_log(par,x);
    local z;
    z = par[1]+par[2]*log(x);
    retp(z);
endp;
```

Escreve-se agora &fun_log no lugar de &f1.

► **Exemplo**

/* Simular uma trajectória do processo $dX_t = (10 - 0.1X_t)\,dt + 0.2X_t dW_t$ com um tempo de discretização $\Delta = 0.05$ */

```
par1 = {10,-0.1};
```

```
par2 = {0,0.2};

y0 = 100;

d = 0.05;

n = 1000;

y = euler (&linear,&linear,par1,par2,y0,d,n);
```

► **Fonte**

c:\gauss\srcnic\sim_ede\sim_ede.src

14.3 euler1

► **Objectivo**

Simular uma EDE através do esquema de Euler (os erros aleatórios podem ser controlados).

► **Formato**

```
y = euler1 (&f1,&f2,e,par1,par2,y0,d);
```

► **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(\theta_1; x)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(\theta_2; x)$.

e: vector dos erros aleatórios de tipo $n \times 1$ (previamente simulado).

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

► **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

► **Library**

```
library sim_ede;
```

► **Fonte**

c:\gauss\srcnic\sim_ede\sim_ede.src

14.4 euler2

▶ Objectivo

Simular uma EDE através do esquema de Euler com um tempo de discretização Δ, mas com uma precisão associada a um tempo de discretização Δ/k.

▶ Formato

y = euler2 (&f1,&f2,par1,par2,y0,d,n,k);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$. Se &f1=0 assume que o drift é zero. A variável par1 deve ser inicializada com um valor qualquer.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

k: inteiro.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X. O intervalo de discretização é de Δ (embora a simulação do processo seja feita com um tempo de discretização Δ/k).

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.5 euler_M2

▶ Objectivo

Simular um sistema de duas EDEs

$$dX_t = a_1\left(X_t, Y_t, \theta\right) dt + b_1\left(X_t, Y_t, \theta\right) dW_{t,1}$$
$$dY_t = a_2\left(X_t, Y_t, \theta\right) dt + b_2\left(X_t, Y_t, \theta\right) dW_{t,2},$$

onde $dW_{t,2} = \rho dW_{t,1} + \sqrt{1-\rho^2} dZ_t$ e Z é independente de $W_{t,1}$, através do esquema de Euler com um tempo de discretização Δ, mas com uma precisão associada a um tempo de discretização Δ/k.

▶ **Formato**

{x,y} = euler_M2(&a1,&b1,&a2,&b2,par_drift1,par_diffusion1,par_drift2,par_diffusion2,rho,x0,y0

▶ **Input**

&a1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a\left(x, y, \theta_1\right)$ associado à primeira EDE

&b1:

&a2:

&b2:

par_drift1:

par_diffusion1:

par_drift2:

par_diffusion2:

rho:

x0:

y0:

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

k: inteiro.

▶ **Output**

x: vector de tipo $n \times 1$ representativo de uma trajectória de X. O intervalo de discretização é de Δ (embora a simulação do processo seja feita com um tempo de discretização Δ/k).

y: vector de tipo $n \times 1$ representativo de uma trajectória de Y. O intervalo de discretização é de Δ (embora a simulação do processo seja feita com um tempo de discretização Δ/k).

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.6 mbg

▶ Objectivo

Simular uma trajectória do movimento Browniano geométrico, através da sua solução exacta.

▶ Formato

y = mbg (a,b,y0,d,n);

▶ Input

a: parâmetro associado ao coeficiente de tendência.

b: parâmetro associado ao coeficiente de difusão.

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.7 mbg1

▶ Objectivo

Simular uma trajectória do movimento Browniano geométrico através da solução exacta do processo (os erros aleatórios podem ser controlados)..

▶ Formato

y = mbg1 (a,b,e,y0,d);

► **Input**

a: parâmetros associado ao coeficiente de tendência.

b: parâmetros associado ao coeficiente de difusão.

e: vector dos erros aleatórios de tipo $n \times 1$ (previamente simulado).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

► **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

► **Library**

library sim_ede;

► **Fonte**

c:\gauss\srcnic\sim_ede\sim_ede.src

14.8 milstein

Objectivo

Simular uma EDE através do esquema de Milstein

► **Formato**

y = milstein (&f1,&f2,par1,par2,y0,d,n);

► **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X

▶ Fonte

sim_ede.src

14.9 milstein1

▶ Objectivo

Simular uma EDE através do esquema de Milstein (os erros aleatórios podem ser controlados).

▶ Formato

y = milstein1 (&f1,&f2,e,par1,par2,y0,d);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

e: vector dos erros aleatórios de tipo $n \times 1$ (previamente simulado).

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.10 milstein2

▶ Objectivo

Simular uma EDE através do esquema de Milstein. Em cada passo Y_{t_i} é simulado dado $X_{t_{i-1}}$ observado (e não, como habitualmente, dado $Y_{t_{i-1}}$). O valor inicial é X_{t_1}.

▶ Formato

y = milstein2 (&f1,&f2,e,x,par1,par2,d);

▶ Input

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x;\theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x;\theta_2)$.

e: vector dos erros aleatórios de tipo $n \times 1$ (previamente simulado).

x: vector das observações X_{t_i}

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

d: Δ, intervalo entre duas observações consecutivas, constante.

▶ Output

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

14.11 ornstein

▶ Objectivo

Simular a EDE $dX_t = \beta(\tau - X_t)\,dt + \sigma dW_t$ através da solução exacta do processo.

▶ Formato

y=ornstein(b,y0,n,d);

▶ Input

b: vector (β, τ, σ) (em coluna).

y0: valor inicial do processo.

n: número de observações do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

▶ **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_ede.src

14.12 platen

▶ **Objectivo**

Simular uma EDE através do esquema de Platen-Wagner

▶ **Formato**

y = platen (&f1,&f2,par1,par2,y0,d,n);

▶ **Input**

&f1: ponteiro para um procedimento que define a especificação funcional do coeficiente $a(x; \theta_1)$.

&f2: ponteiro para um procedimento que define a especificação funcional do coeficiente $b(x; \theta_2)$.

par1: vector θ_1 (em coluna).

par2: vector θ_2 (em coluna).

y0: valor inicial do processo.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações a simular.

▶ **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_ede.src

14.13 simula_b_bridge

▶ **Purpose**

Simulates a Brownian Bridge path

$$B_t = x + W_t - \frac{t}{T}\left(W_T - y + x\right), 0 \leq t \leq T.$$

where W_t is a Brownian motion. We use the formula

$$B_{t_i} = x + W_{t_i} - \frac{t_i}{t_n}\left(W_{t_n} - y + x\right)$$

where $t_i = i\Delta$ $(i = 0, 1, ..., n)$, $\Delta = T/n$, $W_{t_i} = W_{t_{i-1}} + \sqrt{\Delta}\varepsilon_i$, $W_0 = 0$ and ε_i are r.v. with $N(0, 1)$ distribution.

▶ **Format**

{t,y} = simula_b_bridge(n,T,x,y);

▶ **Input**

n: number of points used to simulate W_t.

T: instante final.

x: valor inicial.

y: valor final.

▶ **Output**

t:

y: vector de tipo $(n + 1) \times 1$ representativo de uma trajectória de B.

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_bb.src

14.14 simula_b_bridge2

▶ Purpose

Simulates a Brownian Bridge path

$$B_t = W_t - \frac{t}{T} W_T, \qquad t_0 \leq t \leq T.$$

where W_t is a Brownian motion. We use the formula

$$B_{t_i} = x + W_{t_i} - \frac{t_i}{t_n} \left(W_{t_n} - y + x \right)$$

where $t_i = i\Delta$ $(i = 0, 1, ..., n)$, $\Delta = (T - t_0)/n$, $W_{t_i} = W_{t_{i-1}} + \sqrt{\Delta}\varepsilon_i$, $W_0 = 0$ and ε_i are r.v. with $N(0, 1)$ distribution.

▶ Format

{t,y} = simula_b_bridge2(n,T,t0);

▶ Input

n: number of points used to simulate W_t.

T: instante final.

t0: valor inicial.

▶ Output

t:

y: vector de tipo $(n + 1) \times 1$ representativo de uma trajectória de B.

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_bb.src

14.15 simula_inhomogeneous_BKM

▶ **Objectivo**

Simulate X where X satisfies the SDE (proposed by Black and Karasinski, 1991)

$$dX_t = X_t \left(e^{\tau t} + \frac{\sigma^2}{2} - \alpha \log X_t \right) + \sigma X_t dW_t.$$

We use the fact

$$\ln X_t | X_s \sim N \left(e^{-\alpha(t-s)} \log (X_s) + \frac{e^{\tau t} - e^{-\alpha t + s(\alpha+\tau)}}{\alpha + \tau}, \frac{\left(1 - e^{-\alpha 2(t-s)}\right) \sigma^2}{2\alpha} \right)$$

▶ **Format**

y=simula_inhomogeneous_BKM(b,y0,n,s,d);

▶ **Input**

b: vector (τ, α, σ)

y0:

n:

s: t_0

d: Δ

▶ **Output**

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_inhomogeneous.src

14.16 simula_inhomogeneous_GBM

▶ **Objectivo**

Simulate X where X satisfies the SDE

$$dX_t = \beta X_t dt + \sigma e^{\alpha t} X_t dW_t.$$

We use the fact

$$\ln X_t|\, X_s \sim N\left(\log\left(X_s\right) + \Delta\beta + \frac{e^{2s\alpha} - e^{2t\alpha}}{4\alpha}\sigma^2, \frac{e^{2st} - e^{2s\alpha}}{2\alpha}\right)$$

▶ **Format**

y=simula_inhomogeneous_GBM(b,y0,n,s,d);

▶ **Input**

b: vector (β, σ, α)

y0:

n:

s: t_0

d: Δ

▶ **Output**

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_inhomogeneous.src

14.17 simula_inhomogeneous_ornstein

▶ **Objectivo**

Simulate X where X satisfies the SDE

$$dX_t = -\beta X_t dt + \sigma e^{\alpha t} dW_t.$$

We use the fact

$$X_t|\, X_s \sim N\left(e^{-\beta(t-s)}, \frac{\sigma^2}{2\left(\beta+\alpha\right)}\left(e^{2\alpha t} - e^{2(\alpha s - \beta(t-s))}\right)\right)$$

▶ **Format**

y= simula_inhomogeneous_ornstein(b,y0,n,s,d);

▶ **Input**

b: vector (β, α, σ)

y0:

n:

s: t_0

d: Δ

▶ **Output**

▶ **Library**

library sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_inhomogeneous.src

14.18 simula_rs_ornstein_ornstein

▶ **Objectivo**

Simular uma trajectória do processo com alterações de regime, $dX_t = a(t, X_t)\,dt + b(t, X_t)\,dW_t$ onde $a(t,x) = \beta_1(\tau_1 - X_t)\,I_1(t) + \beta_2(\tau_2 - X_t)\,I_2(t)$, $b(t,x) = \sigma_1 I_1(t) + \sigma_2 I_2(t)$, $I_1(t) = 1$ se em t, $S = 1$ (zero nos outros casos) e $I_2(t) = 1 - I_1(t)$. $S_{i\Delta}$ é uma cadeia de Markov homogénea em tempo discreto, $i \in \{1, 2, ..., n\}$, com espaço de estados $\{1, 2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} \alpha_{11} & 1-\alpha_{11} \\ 1-\alpha_{22} & \alpha_{22} \end{bmatrix}.$$

▶ **Formato**

{y,S} = simula_rs_ornstein_ornstein(d,n,alfa11,alfa22,x0,tau1,beta1,sigma1,

tau2,beta2,sigma2,mostrar);

▶ **Input**

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações do processo.

alfa11: parâmetro α_{11}.

alfa22: parâmetro α_{22}.

x0: Valor inicial

tau1,beta1,sigma1,tau2,beta2,sigma2: $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2)$ nota: β_1 e β_2 podem ser zero

mostrar: assume $\{0, 1\}$. Se mostrar = 1 apresenta-se estimativas para a matriz de probabilidade de transição e um gráfico de pares ordenados (t, S), onde $t = 1, 2, ..., n$.

▶ **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

S:

▶ **Library**

library pgraph,sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_rsw.src

14.19 simula_rs_ornstein_ornstein_1

▶ **Objectivo**

Simular várias réplicas do processo com alterações de regime, $dX_t = a(t, X_t)\,dt + b(t, X_t)\,dW_t$ onde $a(t,x) = \beta_1(\tau_1 - X_t)\,I_1(t) + \beta_2(\tau_2 - X_t)\,I_2(t)$, $b(t,x) = \sigma_1 I_1(t) + \sigma_2 I_2(t)$, $I_1(t) = 1$ se em t, $S = 1$ (zero nos outros casos) e $I_2(t) = 1 - I_1(t)$. $S_{i\Delta}$ é uma cadeia de Markov homogénea em tempo discreto, $i \in \{1, 2, ..., n\}$, com espaço de estados $\{1, 2\}$ e com matriz de probabilidades de transição

$$P = \begin{bmatrix} \alpha_{11} & 1-\alpha_{11} \\ 1-\alpha_{22} & \alpha_{22} \end{bmatrix}.$$

▶ **Formato**

y =simula_rs_ornstein_ornstein_1(d,n,alfa11,alfa22,x0,tau1,beta1,sigma1,tau2,beta2,sigma2,mostra

▶ **Input**

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações do processo.

alfa11: parâmetro α_{11}.

alfa22: parâmetro α_{22}.

x0: Valor inicial

tau1,beta1,sigma1,tau2,beta2,sigma2: $(\tau_1, \beta_1, \sigma_1, \tau_2, \beta_2, \sigma_2)$ nota: β_1 e β_2 podem ser zero

replicas:

▶ **Output**

y: vector de tipo $n \times replicas$

▶ **Library**

library pgraph,sim_ede;

▶ **Fonte**

c:\gauss\srcnic\sim_ede\sim_rsw.src

14.20 simula_Wiener

▶ Purpose

Simulates a Wiener process W_t in the interval $(0, T)$. The procedures is based on the following recursive equation

$$W_{t_i} - W_{t_{i-1}} = \sqrt{\Delta}\varepsilon_i, \qquad W_0 = 0, \qquad \varepsilon_i \sim i.i.d.N(0,1)$$

where

$$\Delta = T/n$$
$$t_i = i\Delta, \ (i = 0, 1, ..., n) \text{ i.e. } t_0 = 0, \ t_1 = \frac{T}{n}, t_2 = 2\frac{T}{n}, ..., t_n = T$$

▶ Format

{t,w} = simula_Wiener(T,n);

▶ Input

T:

n:

▶ Output

t:

w:

▶ Library

library sim_ede;

▶ Fonte

c:\gauss\srcnic\sim_ede\sim_ede.src

15 Simula Outros Processos Estocásticos [sim_ope]

15.1 cad_markov_TC

▶ Objectivo

Simular uma cadeia de Markov homogénea a tempo contínuo com estado de espaços $\{1, 2\}$ e matriz de probabilidades de transição

$$P(\Delta) = \begin{bmatrix} \frac{\mu+\lambda e^{-(\lambda+\mu)\Delta}}{\lambda+\mu} & \frac{\lambda\left(1-e^{-(\lambda+\mu)\Delta}\right)}{\lambda+\mu} \\ \frac{\mu\left(1-e^{-(\lambda+\mu)\Delta}\right)}{\lambda+\mu} & \frac{\lambda+\mu e^{-(\lambda+\mu)\Delta}}{\lambda+\mu} \end{bmatrix}.$$

Observe-se que

$$\begin{aligned} A &= \lim_{h \downarrow 0} \left(\frac{P(h) - I}{h} \right) \\ &= \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix} \end{aligned}$$

é a matriz dos parâmetros infinitesimais (cujas linhas somam zero).

▶ Formato

s = cad_markov_TC(miu,lam,d,n,mostrar);

▶ Input

miu: parâmetro μ.

lam: parâmetro λ.

d: Δ, intervalo entre duas observações consecutivas, constante.

n: número de observações.

mostrar: assume $\{0, 1\}$. Se mostrar = 1 apresentam-se estimativas para a matriz de probabilidade de transição e um gráfico de pares ordenados (t, s), onde $t = 1, 2, ..., n$

▶ Output

s: vector de tipo $n \times 1$ com valores $\{1, 2\}$.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_cm.src

15.2 cad_markov_TD

▶ Objectivo

Simular uma cadeia de Markov homogénea a tempo discreto com estado de espaços $\{1, 2\}$.

▶ Formato

s = cad_markov_TD(p11,p22,n,mostrar,s0);

▶ Input

p11: probabilidade de o processo se encontrar no estado 1 dado que no momento anterior se encontrava no estado 1.

p22: probabilidade de o processo se encontrar no estado 1 dado que no momento anterior se encontrava no estado 2.

n: número de observações.

mostrar: assume $\{0, 1\}$. Se mostrar = 1 apresentam-se estimativas para a matriz de probabilidade de transição e um gráfico de pares ordenados (t, s), onde $t = 1, 2, ..., n$

s0: valor inicial, s0 = 1 ou s0 = 2. Se s0 assumir outro qualquer valor a inicialização faz-se a partir das probabilidades estacionárias.

▶ Output

s: vector de tipo $n \times 1$ com valores $\{1, 2\}$.

▶ Observação

A cadeia é inicializada de acordo com a distribuição estacionária.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_cm.src

15.3 cad_markov_TD_2

▶ Objectivo

Simular uma cadeia de Markov homogénea a tempo discreto com estado de espaços $\{1, 2, ..., m\}$.

▶ Formato

s = cad_markov_TD_2(p,n);

▶ Input

p: matriz de probabilidades de transição de ordem $m \times m$

n: número de observações.

▶ Output

s: vector de tipo $n \times 1$ com valores $\{1, 2, ..., m\}$.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_cm.src

15.4 graph_BRW_01

▶ **Purposes**

Presents a figure including the following graphs:

- y together with $(l,u) = \left(\mu - \alpha + \frac{\log(19)}{\gamma}, \mu + \alpha - \frac{\log(19)}{\gamma}\right)$ → "random walk inner regime"
- $a(x) = -\beta_1 F_1(x-\mu) + \beta_2 F_2(x+\mu), \qquad F_i(x) = \left(1 + e^{-\gamma_i(x-\alpha)}\right)$
- $f(x) = x + a(x)$
- Conditional mean

$$\mathrm{E}\widehat{(y_t|\, y_{t-1})} = \frac{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right) y_t}{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right)}$$

where $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$.

▶ **Formato**

Call graph_BRW_01(b,y);

▶ **Input**

b: beta1|beta2|lamda|alfa|gama1|gama2|miu; Note: lamda=0

y: time series

▶ **Output**

▶ **Library**

library sim_ope,est_etd,pgraph;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\sim_nonlinearAR.src

15.5 graph_BRW_vs_ESTAR_1

▶ **Purposes**

Presents a figure including the following graphs:

- y
- $a_{bwr}(x) = -\beta_1 F_1(x-\mu) + \beta_2 F_2(x+\mu)$, $F_i(x) = \left(1 + e^{-\gamma_i (x-\alpha)}\right)$ vs. $a_{estar}(x) = -x + \mu + \phi_1 (x-\mu) + \phi_2 (x-\mu)\left(1 - \exp\left(-\theta^2 (x-\mu)^2\right)\right)$
- $f_{bwr}(x) = x + a_{bwr}(x)$ vs. $f_{estar}(x) = x + a_{estar}(x)$ vs. Conditional mean

$$\widehat{\mathrm{E}(y_t | y_{t-1})} = \frac{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right) y_t}{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right)}$$

where $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$.

▶ **Formato**

{x0,f1,f2,mc}=graph_BRW_vs_ESTAR_1(b_brw,b_estar,y);

▶ **Input**

b_brw: beta1|beta2|lamda|alfa|gama1|gama2|miu; Note: lamda=0

b_estar: fhi1|fhi2|miu|theta;

y: time series

▶ **Output**

▶ **Library**

library sim_ope,est_etd,pgraph;

▶ **Fonte**

c:\gauss\srcnic\sim_nonlinearAR.src.src

15.6 graph_ESTAR_01

▶ **Purposes**

Presents a figure including the following graphs:

- y
- $a(x) = -x + \mu + \phi_1 (x - \mu) + \phi_2 (x - \mu) \left(1 - \exp\left(-\theta^2 (x - \mu)^2\right)\right)$
- $f(x) = x + a(x)$
- Conditional mean

$$\mathrm{E}\left(\widehat{y_t | y_{t-1}}\right) = \frac{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right) y_t}{\sum_{i=1}^{n-1} K\left(\frac{y_{t-1}-x}{h}\right)}$$

where $K(u) = (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}}$ e $h = (4/3)^{1/5} \hat{\sigma}_X n^{-1/5}$.

▶ **Formato**

Call graph_ESTAR_01(b,y);

▶ **Input**

b: beta1|beta2|miu|theta;

y: time series

▶ **Output**

▶ **Library**

library sim_ope, pgraph;

▶ **Fonte**

c:\gauss\srcnic\sim_nonlinearAR.src.src

15.7 sim_setar_2R

▶ Objectivo

Simular o processo

$$X_t = \begin{cases} a_1 + b_1 X_{t-1} + \sigma_1 \varepsilon_t \; X_{t-1} < k \\ a_2 + b_2 X_{t-1} + \sigma_2 \varepsilon_t \; X_{t-1} \geq k \end{cases}$$

▶ Formato

{y,II,e}=sim_setar_2R(par1,par2,k,n,y0);

▶ Input

par1: vector de tipo 3×1: (a_1, b_1, σ_1)

par2: vector de tipo 3×1: (a_2, b_2, σ_2)

k:

n: número de valores simulados

y0: valor inicial

▶ Output

y: vector $n \times 1$ valores simulados

II: vector 2×1 $\left(\sum I_{\{X_{t-1}<k\}}, \sum I_{\{X_{t-1}\geq k\}}\right)$

e: vector $n \times 1$ inovações

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_setar.src

15.8 sim_setar_3R

▶ Objectivo

Simular o processo

$$X_t = \begin{cases} a_1 + b_1 X_{t-1} + \sigma_1 \varepsilon_t & X_{t-1} > k2 \\ a_2 + b_2 X_{t-1} + \sigma_2 \varepsilon_t & k1 \leq X_{t-1} \leq k2 \\ a_3 + b_3 X_{t-1} + \sigma_3 \varepsilon_t & X_{t-1} < k1 \end{cases}$$

▶ Formato

{y,II,e}=sim_setar_3R(par1,par2,par3,k1,k2,n,y0);

▶ Input

par1: vector de tipo 3×1: (a_1, b_1, σ_1)

par2: vector de tipo 3×1: (a_2, b_2, σ_2)

par3: vector de tipo 3×1: (a_3, b_3, σ_3)

k1:

k2:

n: número de valores simulados

y0: valor inicial

▶ Output

y: vector $n \times 1$ valores simulados

II: vector 3×1 $\left(\sum I_{\{X_{t-1}<k1\}}, \sum I_{\{k1\leq X_{t-1}\leq k2\}}, \sum I_{\{X_{t-1}>k1\}}\right)$

e: vector $n \times 1$ inovações

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_setar.src

15.9 simula_AR1_beta

▶ Objectivo

Simular uma trajectória do processo

$$y_t|\,\mathcal{F}_{t-1} \sim Beta\,(\alpha_t, \beta_t)\,, \qquad \alpha_t > 0, \beta_t > 0$$

onde α_t e β_t satisfazem as seguintes equações:

$$\mathrm{E}\,[y_t|\,\mathcal{F}_{t-1}] = \frac{\alpha_t}{\alpha_t + \beta_t},$$

$$\mathrm{Var}\,[y_t|\,\mathcal{F}_{t-1}] = \frac{\alpha_t\beta_t}{(\alpha_t+\beta_t)^2\,(\alpha_t+\beta_t+1)} = \frac{\mathrm{E}\,[y_t|\,\mathcal{F}_{t-1}]\,(1-\mathrm{E}\,[y_t|\,\mathcal{F}_{t-1}])}{\alpha_t+\beta_t+1}.$$

Assume-se:

$$\mathrm{E}\,[y_t|\,\mathcal{F}_{t-1}] \equiv \mu_t = c + \phi y_{t-1}$$
$$\mathrm{Var}\,[y_t|\,\mathcal{F}_{t-1}] = \sigma^2.$$

Note-se:

$$\alpha_t = -\frac{\sigma^2\mu_t - \mu_t^2 + \mu_t^3}{\sigma^2}, \quad \beta_t = \frac{(-1+\mu_t)\,(\sigma^2 - \mu_t + \mu_t^2)}{\sigma^2}.$$

▶ Formato

{y,m,alfa,beta}=simula_AR1_beta(n,c,fhi,sigma);

▶ Input

n: n^o de observações

c:

fhi:

sigma:

▶ Output

y: vector de tipo $n \times 1$

m: média condicional

alfa:

beta:

▶ Library

library sim_ope;

▶ Observações

y[1] = c/(1-fhi);

▶ Fonte

c:\gauss\srcnic\sim_ope\garch.src

15.10 simula_ARFIMA

▶ Objectivo

Simular uma trajectória do processo

$$\phi(L)(1-L)^d X_t = \Theta(L)\varepsilon_t$$

sendo $d \in \mathbb{R}$ e $\{\varepsilon_t\}$ variáveis aleatórias i.i.d. com distribuição $N\left(0, \sigma^2\right)$.

▶ Formato

x = simula_ARFIMA(n,d,coef_fi,coef_theta,sigma,x0);

▶ Input

n: n^o de observações

d:

coef_fi: vector $\left(\phi_1, ..., \phi_p\right)$. Se $\phi(B) = 1$ escrever "coef_fi = {};"

coef_theta: vector $(\theta_1, ..., \theta_q)$. Se $\Theta(B) = 1$ escrever "coef_theta = {};"

sigma: desvio padrão de ε_t

x0: valor inicial

▶ Output

x: vector de tipo $n \times 1$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_mlonga.src

15.11 simula_Burr

▶ Objectivo

Simular um conjunto de observações com fd Burr:

$$F(x) = 1 - \left(1 + x^{\beta}\right)^{\frac{1}{\rho}}, \qquad \beta = -\rho\alpha$$

▶ Formato

x =simula_Burr(n,rho,alfa);

▶ Input

n: número de observações a simular.

rho:

alfa:

▶ Output

x: vector nx1.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.12 simula_brw_01

▶ Purpose

To Simulate

$$y_t = y_{t-1} - \beta_1 F\left(y_{t-1} - \mu_{t-1}\right) + \beta_2 F\left(-y_{t-1} + \mu_{t-1}\right) + \sigma\varepsilon_t$$
$$\mu_t = \lambda y_{t-1} + (1-\lambda)\,\mu_{t-1}$$

where $\varepsilon_t iidN\,(0,1)$ and $F\,(x) = \left(1 + e^{-\gamma(x-\alpha)}\right)$. Also, it presents a random walk path with u_t innovations, for comparison purposes.

▶ Formato

{y,z}=simula_BRW_01(b,y1,miu1,e,mostrar);

▶ Input

b: beta1|beta2|alfa|gama|lam|sigma

y1: initial value of y

miu1: initial value of μ

e: erros, ε_t

mostrar: if mostrar =1 GAUSS presents some graphs.

▶ Output

y:

z:

If $\lambda \neq 0$ and mostrar $\neq 1$ GAUSS presents xy(0,y~z). If $\lambda = 0$ and mostrar $\neq 1$ GAUSS presents:

```
min=minc(y)*1.5;

max=maxc(y)*1.5;

x=seqa(min,(max-min)/(100-1),100);

begwind;

_pnumht=.23;

_pdate=;

window(2,2,0);

a=-beta1*cdf_logistic(x-miu1,alfa,gama)+beta2*cdf_logistic(-x+miu1,alfa,gama);
```

xy(x,a~(-2*(x-miu1)));

nextwind;

xy(x,(x+a)~x~(-x+2*miu1));

nextwind;

xy(0,y~miu~z);

endwind;

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_nonlinearAR.src.src

15.13 simula_chauchy

▶ Objectivo

Simular um conjunto de observações com distribuição de Chauchy:

$$f(x) = \frac{1}{\pi\left(\lambda + \left(\frac{x-\mu}{\lambda}\right)^2\right)}.$$

▶ Formato

x = simula_chauchy(n,par);

▶ Input

n: número de observações a simular.

par: vector 2x1; par[1]: λ, par[2]: μ.

▶ Output

x: vector nx1.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.14 simula_diagonal_VECH

▶ Objectivo

Simulate

$$\mathbf{y}_t = \mathbf{\Phi}\mathbf{y}_{t-1} + \mathbf{u}_t$$

where

$$\mathbf{u}_t = \mathbf{H}_t^{1/2}\boldsymbol{\varepsilon}_t$$

$$\begin{aligned}\operatorname{vech}(\mathbf{H}_t) = \begin{pmatrix} h_{11,t} \\ h_{12,t} \\ h_{22,t} \\ \vdots \\ h_{mm,t} \end{pmatrix} &= \begin{pmatrix} w_{11} \\ w_{12} \\ w_{22} \\ \vdots \\ w_{mm} \end{pmatrix} + \begin{pmatrix} \alpha_{11} & 0 & 0 & \cdots & 0 \\ 0 & \alpha_{12} & 0 & \cdots & 0 \\ 0 & 0 & \alpha_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{mm} \end{pmatrix} \begin{pmatrix} u_{1,t-1}^2 \\ u_{1,t-1}u_{2,t-1} \\ u_{2,t-1}^2 \\ \vdots \\ u_{m,t-1}^2 \end{pmatrix} \\ &+ \begin{pmatrix} \beta_{11} & 0 & 0 & \cdots & 0 \\ 0 & \beta_{12} & 0 & \cdots & 0 \\ 0 & 0 & \beta_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{mm} \end{pmatrix} \begin{pmatrix} h_{11,t-1} \\ h_{12,t-1} \\ h_{22,t-1} \\ \vdots \\ h_{mm,t-1} \end{pmatrix}\end{aligned}$$

or

$$\begin{aligned}\begin{pmatrix} h_{11,t} & h_{12,t} & \cdots & h_{1m,t} \\ h_{12,t} & h_{22,t} & \cdots & h_{2m,t} \\ \vdots & \vdots & & \vdots \\ h_{1m,t} & h_{2m,t} & \cdots & h_{mm,t} \end{pmatrix} &= \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{12} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & & \vdots \\ w_{1m} & w_{2m} & \cdots & w_{mm} \end{pmatrix} \\ &+ \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1m} \\ \alpha_{12} & \alpha_{22} & \cdots & \alpha_{2m} \\ \vdots & \vdots & & \vdots \\ \alpha_{1m} & \alpha_{2m} & \cdots & \alpha_{mm} \end{pmatrix} \circ \begin{pmatrix} u_{1,t-1}^2 & u_{1,t-1}u_{2,t-1} & \cdots & u_{1,t-1}u_{m,t-1} \\ u_{1,t-1}u_{2,t-1} & u_{2,t-1}^2 & \cdots & u_{2,t-1}u_{m,t-1} \\ \vdots & \vdots & & \vdots \\ u_{1,t-1}u_{m,t-1} & u_{2,t-1}u_{m,t-1} & \cdots & u_{m,t-1}^2 \end{pmatrix} + \\ &\begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{12} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & & \vdots \\ \beta_{1m} & \beta_{2m} & \cdots & \beta_{mm} \end{pmatrix} \circ \begin{pmatrix} h_{11,t-1} & h_{12,t-1} & \cdots & h_{1m,t-1} \\ h_{12,t-1} & h_{22,t-1} & \cdots & h_{2m,t-1} \\ \vdots & \vdots & & \vdots \\ h_{1m,t-1} & h_{2m,t-1} & \cdots & h_{mm,t-1} \end{pmatrix}\end{aligned}$$

$$\mathbf{H}_t = \mathbf{w} + \mathbf{a} \circ \mathbf{u}_{t-1}\mathbf{u}_{t-1}' + \mathbf{b} \circ \mathbf{H}_{t-1}$$

▶ Formato

{y,mat_h}=simula_diagonal_VECH(n,fhi,w1,a1,b1);

▶ Input

n: number of observations.

par: $m \times m$ matrix

w1: $m \times m$ matrix such that $\mathbf{w} = \mathbf{w1}.\mathbf{w1}'$

a1: $m \times m$ matrix such that $\mathbf{a} = \mathbf{a1}.\mathbf{a1}'$

b1: $m \times m$ matrix such that $\mathbf{b} = \mathbf{b1}.\mathbf{b1}'$

▶ Output

y: $n \times m$ matrix

mat_h: $n \times ((1+m) * m/2)$ matrix:

$$\begin{pmatrix} h_{11,t} & h_{12,t} & h_{22,t} & h_{13,t} & h_{23,t} & h_{33,t} & \cdots & h_{m,t} \end{pmatrix}_{t=1,\ldots,n}$$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.15 simula_disc_dist

▶ Objectivo

Simular uma sucessão de v.a. i.i.d. com distribuição $\{p_1, ..., p_m\}$ e espaço de estados $\{1, 2, ..., m\}$.

▶ Formato

S = simula_disc_dist(n,p);

▶ Input

n: número de observações a simular.

p: vector $m \times 1$

▶ Output

s: vector $n \times 1$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.16 simula_estar_1

▶ Purpose

Simulates

$$y_t = \mu + \beta_1 (y_{t-1} - \mu) + \beta_2 (y_{t-1} - \mu) \left(1 - \exp\left(-\theta^2 (y_{t-1} - \mu)^2\right)\right) + \sigma\varepsilon_t$$

assuming that $\{\varepsilon_t\}$ is a Gaussian white noise. Also, it presents a random walk path with $\sigma\varepsilon_t$ innovations, for comparison purposes.

▶ Format

{y,z}=simula_estar_1(b,y1,e,mostrar);

▶ Input

b: beta1|beta2|miu|theta|sigma

y1: initial value of y

miu1: initial value of μ

e: errors, ε_t

mostrar: if mostrar =1 GAUSS presents some graphs.

▶ Output

y: ESTAR path

z: Random Walk path

If mostrar $\neq$ 1 GAUSS presents

```
min=minc(y)*1.5;
max=maxc(y)*1.5;
x=seqa(min,(max-min)/(100-1),100);
begwind;
_pnumht=.23;
_pdate=;
window(2,2,0);
a=-x+miu+beta1*(x-miu)+beta2*(x-miu).*(1-exp(-theta^2*(x-miu)^2));
xy(x,a~(-2*(x-miu)));
```

nextwind;

xy(x,(x+a)~x~(-x+2*miu));

nextwind;

xy(0,y~z);

endwind;

► **Library**

library sim_ope;

► **Source**

c:\gauss\srcnic\sim_nonlinearAR.src.src

15.17 simula_F

▶ Purpose

To compute F r.v. One of the parameters is related to the tail index.

▶ Format

F=simula_F(n,gl1,alfa);

▶ Input

n:

gl1:

alfa: Note that gl2=alfa/2

▶ Output

x:

▶ Library

library sim_ope;

▶ Source

c:\gauss\srcnic\sim_ope\distrib.src

15.18 simula_GARCH11

▶ **Objectivo**

▶ **Formato**

{y,media,var}=simula_GARCH11(c,fi,a0,a1,d1,e,y0);

▶ **Input**

y0: if y0={} then y0=0 if fi=1 and y0=c/(1-fi) if |fi|<1

▶ **Output**

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\garch.src

15.19 simula_GARCHpq

▶ **Objectivo**

Simular o modelo

$$
\begin{aligned}
y_t &= c + \phi y_{t-1} + u_t \\
u_t &= \sigma_t \varepsilon_t, \\
\sigma_t^2 &= a_0 + \sum_{i=1}^{q} a_i u_{t-i}^2 + \sum_{i=1}^{p} b_i \sigma_{t-i}^2
\end{aligned}
$$

onde ε_t é uma sequência de v.a. i.i.d. de média zero e variância um com distribuição normal ou distribuição t-Student (normalizada).

▶ **Formato**

{y,media,var}=simula_GARCHpq(c,fi,a0,a,b,n,dist);

▶ **Input**

c:

fi:

a0:

a: vector de dimensão não superior a 10 ($q \leq 10$).

b: vector de dimensão não superior a 10 ($p \leq 10$).

n: escalar ou vector; se n é escalar define-se a dimensão do vector a simular. Se n for vector define-se o vector das inovações ε. Neste caso o input dist não é considerado.

dist: escalar; se dist = 0 ε_t tem distribuição normal; se dist = gl > 0 com gl inteiro então ε_t tem distribuição t-Student com gl graus de liberdade. Input activado apenas se n for escalar.

▶ Output

▶ Library

library sim_ope;

▶ Observações

Para simular por exemplo, um processo com variância condicional

$$\sigma_t^2 = 1 + .2u_{t-1} + .4u_{t-4} + .1\sigma_{t-2}^2$$

considerar,

a = .2|0|0|.4;

b = 0|.2;

▶ Fonte

c:\gauss\srcnic\sim_ope\garch.src

15.20 simula_MMC1

▶ Objectivo

Simula uma cadeia de MMC com s categorias ($2 \leq s \leq 3$) e m ($m \geq 2$) estados. No caso $s = 2$, o processo S_{2t} é simulado de forma independente e o processo S_{1t} é simulado de acordo com as probabilidades de transição forncecidas. S_{1t} depende de S_{1t-1} e S_{2t-1}. No caso $s = 3$, os processos S_{2t} e S_{3t} são simulados de forma independente e S_{1t} é simulado de acordo com as probabilidades de transição fornecidas. S_{1t} depende de S_{1t-1}, S_{2t-1} e S_{3t-1}.

▶ Formato

S=simula_MMC1(cat,m,n,prob);

▶ Input

cat: n^o de categorias (cat=2 ou cat=3). Parâmetro s.

m: n^o de estados que cada categoria pode assumir, $m \geq 2$

n: n^o de observações

prob: matriz de probabilidades de transição de dimensão $m^s \times m$ (ver exemplo). Se prob=0 (escalar) cada célula da matriz de probabilidades de transição assumirá o valor $1/m$.

▶ Output

s: matriz de dimensão $n \times s$

▶ Exemplo

Seja $P\left(i_0 | \, i_1, i_2\right) = P\left(S_{1t} = i_0 | \, S_{1t-1} = i_1, S_{2t-1} = i_2\right).$

Caso $m = 2$ e $s = 2$:

		prob	
1	1	$P\left(1 \mid 1,1\right)$	$P\left(2 \mid 1,1\right)$
1	2	$P\left(1 \mid 1,2\right)$	$P\left(2 \mid 1,2\right)$
2	1	$P\left(1 \mid 2,1\right)$	$P\left(2 \mid 2,1\right)$
2	2	$P\left(1 \mid 2,2\right)$	$P\left(2 \mid 2,2\right)$

Caso $m = 2$ e $s = 3$:

		prob		
1	1	$P(1\|1,1)$	$P(2\|1,1)$	$P(3\|1,1)$
1	2	$P(1\|1,2)$	$P(2\|1,2)$	$P(3\|1,2)$
1	3	$P(1\|1,3)$	$P(2\|1,3)$	$P(3\|1,3)$
2	1	$P(1\|2,1)$	$P(2\|2,1)$	$P(3\|2,1)$
2	2	$P(1\|2,2)$	$P(2\|2,2)$	$P(3\|2,2)$
2	3	$P(1\|2,3)$	$P(2\|2,3)$	$P(3\|2,3)$
3	1	$P(1\|3,1)$	$P(2\|3,1)$	$P(3\|3,1)$
3	2	$P(1\|3,2)$	$P(2\|3,2)$	$P(3\|3,2)$
3	3	$P(1\|3,3)$	$P(2\|3,3)$	$P(3\|3,3)$

Nota: cada linha soma 1.

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_mmc.src

15.21 simula_MMC2

▶ Objectivo

Igual a simula_MMC1, com a diferença que todos os processos são simulados de acordo com a matriz de probabilidades.

▶ Formato

S=simula_MMC2(cat,m,n,prob);

▶ Input

cat: n^o de categorias (cat=2 ou cat=3). Parâmetro s.

m: n^o de estados que cada categoria pode assumir, $m \geq 2$

n: n^o de observações

prob: matriz de probabilidades de transição de dimensão $cat \times (m^s \times m)$ (ver exemplo). Se prob=0 (escalar) cada célula da matriz de probabilidades de transição assumirá o valor $1/m$.

▶ Output

s: matriz de dimensão $n \times s$

▶ Exemplo

Seja $P_j\left(i_0 | \, i_1, i_2\right) = P\left(S_{jt} = i_0 | \, S_{1t-1} = i_1, S_{2t-1} = i_2\right).$

Caso $m = 2$ e $s = 2$:

		prob		prob	
1	1	$P_1\left(1 \mid 1,1\right)$	$P_1\left(2 \mid 1,1\right)$	$P_2\left(1 \mid 1,1\right)$	$P_2\left(2 \mid 1,1\right)$
1	2	$P_1\left(1 \mid 1,2\right)$	$P_1\left(2 \mid 1,2\right)$	$P_2\left(1 \mid 1,2\right)$	$P_2\left(2 \mid 1,2\right)$
2	1	$P_1\left(1 \mid 2,1\right)$	$P_1\left(2 \mid 2,1\right)$	$P_2\left(1 \mid 2,1\right)$	$P_2\left(2 \mid 2,1\right)$
2	2	$P_1\left(1 \mid 2,2\right)$	$P_1\left(2 \mid 2,2\right)$	$P_2\left(1 \mid 2,2\right)$	$P_2\left(2 \mid 2,2\right)$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_mmc.src

15.22 simula_palgarch

▶ Objectivo

Simula o processo

$$
\begin{aligned}
X_t &= X_{t-1} + e^k \left(e^{-a_1(X_{t-1}-\tau)} - e^{a_2(X_{t-1}-\tau)} \right) + u_t \\
u_t &= \sigma_t \varepsilon_t, \qquad \varepsilon_t \text{ i.i.d. } N(0,1) \\
\sigma_t^2 &= \alpha_0 + \alpha_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2
\end{aligned}
$$

▶ Formato

{y,media,var,yrw}=simula_palgarch(n,y0,k,a1,a2,tau,alfa0,alfa1,beta1);

▶ Input

n: número de valores simulados

y0:

a1:

a2:

tau:

alfa0:

alfa1:

beta1:

▶ Output

y:

media:

var:

yrw: Passeio Aleatório com inovações $\sigma\varepsilon_t$, $\sigma^2 = \alpha_0 / (1 - \alpha_1 - \beta_1)$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\garch.src

15.23 simula_Pareto

▶ Objectivo

O nome diz tudo.

▶ Formato

y=simula_pareto(n,alfa);

▶ Input

n: número de valores simulados

alfa índice de cauda

▶ Output

y: vector de tipo $n \times 1$ onde cada elemento é i.i.d. e tem distribuição $Pareto\,(\alpha)$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.24 simula_paretoII

▶ Objectivo

Simula y com distribuição

$$f\left(y_i|\,\mathbf{x}_i\right) = \alpha\left(\mathbf{x}_i\right)\sigma^{-1}\left(1+\frac{y_i-x_0}{\sigma}\right)^{-(\alpha(\mathbf{x}_i)+1)}$$

▶ Formato

y=simula_paretoII(n,alfa,x0,sigma);

▶ Input

n: número de valores simulados

alfa: índice de cauda

x0:

sigma:

▶ Output

y: vector de tipo $n \times 1$ onde cada elemento é i.i.d. e tem distribuição $Pareto\left(\alpha\right)$

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\distrib.src

15.25 simula_Poisson

▶ **Objectivo**

O nome diz tudo

▶ **Formato**

y=simula_Poisson(n,lam);

▶ **Input**

n: número de valores simulados

lam: parâmetro λ.

▶ **Output**

y: vector de tipo $n \times 1$ onde cada elemento é i.i.d. e tem distribuição $Po(\lambda)$

▶ **Observações**

Se $\lambda > 30$ usa-se o Teorema do limite central.

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\distrib.src

15.26 simula_RS_GARCH

▶ **Objectivo**

Simular o modelo

$$y_t = (a_{01} + a_{11} y_{t-1} + \sigma_{t1}\varepsilon_t)\,\mathcal{I}_{\{S_t=1\}} + (a_{02} + a_{12} y_{t-1} + \sigma_{t2}\varepsilon_t)\,\mathcal{I}_{\{S_t=2\}}$$
$$\sigma^2_{t1} = b_{01} + b_{11} u^2_{t-1} + b_{21}\sigma^2_{t-1}, \qquad \sigma^2_{t2} = b_{02} + b_{12} u^2_{t-1} + b_{22}\sigma^2_{t-1}.$$

/* S=1 */

a01=1;a11=0.9; @ media @

b01=1;b11=0;b21=0; @ variancia @

P=0.9;

/* S=2 */

a02=2;a12=0.9; @ media @

b02=1;b12=0;b22=0; @ variancia @

Q=0.2;

▶ **Formato**

{y,p1,h1,h2,h}=simula_RS_GARCH(y0,n,b01,b11,b21,a01,a11,b02,b12,b22,a02,a12,P,Q);

▶ **Input**

y0:

▶ **Output**

y: vector de tipo $n \times 1$ representativo de uma trajectória de X.

p1: ...

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\sim_rsw.src

15.27 simula_stable_dist

▶ Purpose

To generate data from a stable distribution with index α (using the method of Samorodnitsky-Taqqu (1994), i.e.

$$X_t = \frac{\sin(\alpha\theta_t)}{[\cos(\theta_t)]^{1/\alpha}} \left(\frac{\cos[(1-\alpha)\theta_t]}{z_t} \right)^{(1-\alpha)/\alpha}$$

where θ_t is uniform on $(-\pi/2, \pi/2)$ and z_t is exponential with mean 1. Note that $\alpha = 2$ corresponds to the Gaussian distribution.

▶ Format

x=simula_stable_dist(n,alfa);

▶ Input

n: número de valores simulados

alfa:

▶ Output

x:

▶ Library

library sim_ope;

▶ Source

c:\gauss\srcnic\sim_ope\distrib.src

15.28 simula_stock_prices

▶ **Objectivo**

Simular o processo estocástico

$$\lambda_t = [c + \phi N_{t-1}] + 1$$
$$N_t \sim Po\,(\lambda_t)$$
$$S_t = S_{t-1} + \sum_{i=0}^{N_t} \sigma\varepsilon_i$$

onde S_t procura representar o preço de uma acção.

▶ **Formato**

{S,NN}=simula_stock_prices(s0,lam0,c,fi,sigma,n);

▶ **Input**

n:

▶ **Output**

y:

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\simula_AF.src

15.29 simula_tStudent

▶ **Objectivo**

O nome diz tudo

▶ **Formato**

t=simula_tStudent(n,gl);

▶ **Input**

n: número de valores simulados

gl: graus de liberdade.(inteiro n, maior do que 1)

▶ **Output**

t: vector de tipo $n \times 1$ onde cada elemento é i.i.d e tem distribuição t-Student e gl graus de liberdade.

▶ **Library**

library sim_ope;

▶ **Fonte**

c:\gauss\srcnic\sim_ope\distrib.src

15.30 simula_VAR1

▶ **Purpose**

Simulates $y_t = (y_{t1}, ..., y_{tm})$

$$\begin{bmatrix} y_{t1} \\ \vdots \\ y_{tm} \end{bmatrix} = \begin{bmatrix} \phi \cdots 0 \\ 0 \ddots 0 \\ 0 \cdots \phi \end{bmatrix} \begin{bmatrix} y_{t-1,1} \\ \vdots \\ y_{t-1,m} \end{bmatrix} + \begin{bmatrix} u_{t1} \\ \vdots \\ u_{t1} \end{bmatrix}$$

$$\begin{bmatrix} u_{t1} \\ \vdots \\ u_{t1} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{H})$$

where

$$\mathbf{H} = \begin{bmatrix} \sigma^2 & \rho\sigma\sigma \cdots \rho\sigma\sigma \\ \rho\sigma\sigma & \sigma^2 \cdots \rho\sigma\sigma \\ \vdots & \vdots \ddots \vdots \\ \rho\sigma\sigma & \rho\sigma\sigma \cdots \sigma_{mt}^2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \sigma & 0 \cdots 0 \\ 0 & \sigma \cdots 0 \\ \vdots & \vdots \ddots \vdots \\ 0 & 0 \cdots \sigma \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} 1 & \rho \cdots \rho \\ \rho & 1 \cdots \rho \\ \vdots & \vdots \ddots \vdots \\ \rho & \rho \cdots 1 \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} \sigma & 0 \cdots 0 \\ 0 & \sigma \cdots 0 \\ \vdots & \vdots \ddots \vdots \\ 0 & 0 \cdots \sigma \end{bmatrix}}_{\mathbf{D}}$$

▶ **Format**

y=simula_VAR1(n,m,var,rho,fhi);

▶ **Input**

n: number of observations

m: number of equations

var: σ

rho: ρ

fhi: scalar ϕ, or $(\phi_1, ..., \phi_m)$ $(1 \times m)$

▶ Output

y:

▶ Library

library sim_ope;

▶ Fonte

c:\gauss\srcnic\sim_ope\sim_var.src

15.31 simula_varp

▶ Objetivo

Simular $VAR(p)$

$$y_t = v + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t$$

onde y_t é um vector $K \times 1$ e $u_t \sim N(0, \Sigma_u)$

▶ Formato

y=simula_varp(y0,v,w,A,n);

▶ Input

y0: matriz de valores iniciais de dimensão $p \times K$

v: vector the dimensão $K \times 1$

w: vech (Σ_u) vetor de dimensão de dimensão $((1+K)K/2) \times 1$. Por exemplo

$$\Sigma_u = \begin{pmatrix} 1 & .1 & .2 \\ .1 & 2 & .3 \\ .2 & .3 & 3 \end{pmatrix}$$

w=1|.1|2|.2|.3|3

A: matriz $\begin{bmatrix} A_1 A_2 \cdots A_p \end{bmatrix}$ de dimensão $K \times Kp$ (não corresponde exatamente à matriz **A** de Lutkepohl).

n: dimensão da amostra a simular

▶ Output

- Valores próprios da matriz

$$\begin{pmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I_K & 0 & \cdots & 0 & 0 \\ 0 & I_K & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I_K & 0 \end{pmatrix}$$

  ou de A_1 no caso $p = 1$

- Em condições estacionárias apresenta Médias, Variâncias, Covariâncias e Correlações
- Gráficos (basta fazer **xy(0,y);)**

▶ Library

***

▶ Fonte

***

16 Testes [testes]

16.1 bds

▶ Objectivo

▶ Formato

{rbds,v,stat }=bds(x,eps,m);

▶ Input

x: Series to be tested for iid

eps: Threshold value of the indicator function

m: m-history vector: x(t,m)=(x(t),x(t+1),...,x(t+m-1))

▶ Output

rbds: Nx3 matrix containing:

** Vector of BDS statistics (One BDS for each r)

** The Correlation integral for x(t,m)

** The Correlation integral for x(t)

v: Vector of statistics needed to calculate the variance of the BDS stat: v={vm2,k,c}'

stat: 7x1 vector with the following stats:

Cramer-Von Mises; Kolmogorov-Smirnov; Kuiper; Mallows; Anderson-Darling; U2; BDS

▶ Observações

/* BDS - Brock, Dechert & Scheinkman Test Statistic

** (C), Copywright 1990 and 1995 by Pedro de Lima

** All rights reserved**

▶ Library

library testes;

▶ Fonte

c:\gauss\srcnic\testes\testes_nl.src

16.2 change_break

▶ Purpose

Finds structural breaks.

▶ Format

call change_break(x,p);

▶ Input

x: Series to be tested for structural break

p: See below

$$v = \text{Var}\left(x_t\right) + \sum_{s=1}^{p}\left(1 - \frac{s}{p+1}\right)\left(\text{Cov}\left(x_t, x_{t-s}\right) + \text{Cov}\left(x_t, x_{t-s}\right)'\right)$$

Long run variance of x_t using the Newey-West estimator

▶ Output

▶ Observações

See Kokoszka and R. Leipus (1999),Testing for parameter changes in ARCH models, 39(2) and Andreou, E. & Ghysels, E (2002). Detecting multiple breaks in financial market volatility dynamics Journal of Applied Econometrics, John Wiley & Sons, Ltd., 2002, 17, 579-600

▶ Library

library testes;

▶ Fonte

c:\gauss\srcnic\testes\others.src

16.3 goodnessfit

▶ Objectivo

Ensaiar a hipótese

$$H0 : X \sim f\left(x|\theta_1, ..., \theta_k\right)$$

▶ Formato

{chi_square,pvalue,b,freq,ve}=goodnessfit(&cdf,x,par,ni,criterio,print_resultados);

▶ Input

cdf: ponteiro para um procedimento que calcule probabilidades sob a hipótese nula (ver observações)

x: vector das observações

par: vector dos parâmetros. Se &cdf= cdfn_int considerar par=$\mu|\sigma$; Se &cdf =&cdfchauchy_int considerar $\lambda|\mu$.

ni: número de classes (ou intervalos) a considerar no teste.

criterio: escalar, eliminar as classses (ou intervalos) cujos os valores esperados sejam inferiores ao valor criterio.

print_resultados: se igual a 1 apresenta os resultados e gráfico.

▶ Output

chi-square: valor da estatística de teste.

pvalue:

b: x

freq: vector das frequências (pode ter dimensão inferior a ni se a variável criterio for superior a zero)

ve: vector dos valores esperados sob a hipótese nula ((pode ter dimensão inferior a ni se a variável criterio for superior a zero).

▶ Observações

Estão definidos três procedimentos que poderão ser usados na opção Input cdf: (a) cdfn_int (distribuição normal) ; (b) cdfchauchy_int (distribuição de Chauchy) e cdfGEV_int (Generalized Extreme Value Distribution),

$$G\left(x\right) = \exp\left\{-\left(1 + \frac{1}{\alpha}\left(\frac{x-\lambda}{\delta}\right)\right)^{-\alpha}\right\}, \qquad 1 + \frac{1}{\alpha}\left(\frac{x-\lambda}{\delta}\right) > 0$$

$$par = Alpha|Lambda|Delta$$

Exemplifica-se o procedimento cdfn_int caso se pretenda ensair outra distribuição:

```
proc (1)=cdfn_int(b,par);
    local p,ni;
    b=(b-par[1])/par[2]; /* b vector: limite inferior de cada classe */
    ni=rows(b); /* numero de classes */
    p=zeros(ni,1);
    p[1]=cdfn(b[1]);
    for i (2,ni,1);
        p[i]=cdfn(b[i])-cdfn(b[i-1]);
    endfor;
retp(p);
endp;
```

▶ Library

library util,testes,pgraph;

▶ Fonte

c:\gauss\srcnic\testes\nao_param.src

16.4 log_periodogram

▶ Purpose

Estimates the fractional parameter d using the log periodogram estimator (Geweke and Porter-Haundak) and evaluates the hypothesis $H_0 : d = 0$.

▶ Format

{w,I}=log_periodogram(x,w,I);

▶ Input

x: vector of observations

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ or scalar. If w = scalar the procedure calculates ω

I: Periodogram $I(\omega_k)$, $k = 0, ..., [n/2]$ or scalar. If I = scalar the procedure calculates I

► **Output**

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$

I: Periodogram $I(\omega_k)$, $k = 0, ..., [n/2]$

► **Library**

library testes,pgraph;

► **Observation**

Supply ω and I whenever possible. It reduces the computational time

► **Source**

c:\gauss\srcnic\testes\long_memory.src;

16.5 lm_robinson

▶ Purpose

Estimates the fractional parameter d (and H) using the LM estimator of Robinson and evaluates the hypothesis $H_0 : d = 0$.

▶ Format

{w,I}=lm_robinson(x,w,I);

▶ Input

x: vector of observations

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$ or scalar. If w = scalar the procedure calculates ω

I: Periodogram $I(\omega_k)$, $k = 0, ..., [n/2]$ or scalar. If I = scalar the procedure calculates I

▶ Output

w: $\omega_k = 2\pi k/n$, $k = 0, ..., [n/2]$

I: Periodogram $I(\omega_k)$, $k = 0, ..., [n/2]$

▶ Library

library testes,pgraph;

▶ Observation

Supply ω and I whenever possible. It reduces the computational time. See Cassola and Morana (2003).

▶ Source

c:\gauss\srcnic\testes\long_memory.src;

16.6 moses_test

▶ Purpose

Test H0: Extreme values are equally likely in both populations. See Siegel, "Estatística não paramétrica"

▶ Format

p=moses_test(x,y,h);

▶ Input

x:

y:

h: Número de scores de ambos os extremos que são removidos. Por exemplo, se h=1

Ordem	Grupo
1	A
2	B
3	A
4	B
5	A
6	A
7	A
8	B
9	A
10	B

In the case $h = 1$ the rank amplitude is $8 - 4 + 1 = 5$ (9 if $h = 0$).

▶ Output

p: p-value

▶ Library

library testes

▶ Source

c:\gauss\srcnic\testes\long_memory.src;

16.7 runs_test

▶ Objectivo

Testa a indepêndencia de uma sequência de zeros e uns.

▶ Formato

call runs_test(I);

▶ Input

I: Sequência de zeros e uns

▶ Output

Calcula: x(n^o runs), n0(n^o zeros), n1(n^o uns), z p-value

▶ Library

library testes,pgraph;

▶ Fonte

c:\gauss\srcnic\testes\nao_param.src;

16.8 test_TAR1

▶ Objectivo

Testa AR(p) contra TAR(p), concretamente $H_0 : \phi_{1i} = \phi_{2i}$ $(i = 0, 1, .., p)$ no contexto do modelo

$$y_t = \begin{cases} \phi_{10} + \phi_{11} y_{t-1} + ... + \phi_{1p} y_{t-p} + u_t \, q_{t-d} \leq \gamma \\ \phi_{20} + \phi_{21} y_{t-1} + ... + \phi_{2p} y_{t-p} + u_t \, q_{t-d} > \gamma \end{cases}$$

Procedimento (Hansen, 2000):

1. obter $F_n = n\left(\tilde{\sigma}_n^2 - \hat{\sigma}_n^2\right)/\hat{\sigma}_n^2$ onde $\tilde{\sigma}_n^2$ é a variância dos erros de regressão do modelo AR (sob H_0) e $\hat{\sigma}_n^2$ resulta de (**??**);
2. simular u_t^*, $t = 1, ..., n$ com distribuição i.i.d. $N(0, 1)$;
3. $y_t^* = u_t^*$;
4. fazer a regressão de y_t^* sobre $x_t = \left(1 \, y_{t-1} \, ... \, y_{t-p}\right)$ e obter $\tilde{\sigma}_n^{*2}$
5. obter $\hat{\gamma} = \arg\min_{\gamma \in \tilde{\Gamma}} \hat{\sigma}_n^{*2}(\gamma)$ onde $\hat{\sigma}_n^{*2}(\gamma)$ resulta da regressão de y_t^* sobre $\mathbf{x}_t(\gamma)$;
6. obter $F_n^* = n\left(\tilde{\sigma}_n^{*2} - \hat{\sigma}_n^{*2}\right)/\hat{\sigma}_n^{*2}$

7. repetir os passos 2-6 B vezes

8. valor-p = percentagem de vezes (em B) em que $F_n^* \geq F_n$.

▶ **Formato**

call test_TAR1(n,s,d,p,alfa,Fobs);

▶ **Input**

n:

s: n^o de simulações (B)

d: d óptimo obtido na estimação

p: ordem do AR(p)

alfa: usar o mesmo alfa definido na estimação: ver o procedimento estima_TAR1. Considerar, por exemplo, alfa=0.1

▶ **Output**

valor-p = percentagem de vezes (em B) em que $F_n^* \geq F_n$.

▶ **Library**

library est_etd,pgraph;

▶ **Fonte**

c:\gauss\srcnic\est_etd\tar.src

16.9 teste_media_binomial

▶ **Objectivo**

Dado $X \sim B(n, \alpha)$, testa $H_0 : \mathrm{E}[X] = \alpha_0$ através da estatística de teste (rácio de verosimilhanças)

$$RV = -2 \log \frac{L(\alpha | H_0)}{L(\hat{\alpha})} = -2 \log \frac{\alpha^{n_1} (1-\alpha)^{n_0}}{\hat{\alpha}^{n_1} (1-\hat{\alpha})^{n_0}}$$

que, sob H_0 tem distribuição assimptótica $\chi^2_{(1)}$. $\hat{\alpha}$ é o estimador de máxima verosimilhança,

▶ **Formato**

call teste_media_binomial(I);

▶ **Input**

I: Sequência de zeros e uns

▶ **Output**

Calcula: $\hat{\alpha}$, n0 (n^o de zeros), n1(n^o uns), RV e p-value.

▶ **Library**

library testes,pgraph;

▶ **Fonte**

c:\gauss\srcnic\testes\nao_param.src;

16.10 testing_independence_01

▶ **Objectivo**

Testar a hipótese de X não exibir qualquer dependencia

1) Dado um conjunto de probabilidades, e.g. quant=.2|.4|.6|.8 , determinam-se os respectivos quantis;

2) Categoriza-se X numa sequências de $1, 2, ..., m$ (m é igual ao numero de linhas de quant+1)

3) Seja S a sequencia. Testa-se

$$H_0 : p_{ij} = \pi_j, \ i, j = 1, ..., m$$

4) Sob H_0

$$Q = \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{(n_{ij} - n_i n_j/n)^2}{n_i n_j/n} \xrightarrow{d} \chi^2_{\left((m-1)^2\right)}$$

sendo

$$n_{ij} = \sum_{t=1}^{n} \mathcal{I}_{\{S_t=j, S_{t-1}=i\}}$$
$$n_i = \sum_{j=1}^{m} n_{ij}$$
$$n = total \text{ obs.}$$

▶ **Formato**

{Q,pvalue,C,expec_values,P,stat_prob}=testing_independence_01(x,quant);

▶ **Input**

x:

quant: (ver objectivo)

▶ **Output**

Q: Estatística Q

pvalue: $P\left(\chi^2_{((m-1)^2)} > Q\right)$

C: Matriz $[n_{ij}]_{i,j=1,...,m}$

expec_values: Matriz $[n_i n_j/n]_{i,j=1,...,m}$

P: Matriz $[p_{ij}]_{i,j=1,...,m}$, $p_{ij} = P\left(S_t = j \middle| S_{t-1} = i\right)$

Stat_prob: probabilidades estacionárias

► Library

library testes;

► Fonte

c:\gauss\srcnic\testes\markov_chain.src;

16.11 testing_linearity_01

► Objectivo (REFORMULAR!)

Testar a hipótese de X seguir um processo linear $\mathrm{E}\left(X_t \middle| X_{t-1}\right) = c + \phi X_{t-1}$. Procedimentos:

1) Dado um conjunto de probabilidades, e.g. quant=.2|.4|.6|.8 , determinam-se os respectivos quantis;

2) Categoriza-se X numa sequências de $1, 2, ..., m$ (m é igual ao numero de linhas de quant+1)

3) Seja S a sequencia. Testa-se

$$\mathrm{E}\left(S_t \middle| S_{t-1} = j\right) - \mathrm{E}\left(S_t \middle| S_{t-1} = j-1\right) = \mathrm{E}\left(S_t \middle| S_{t-1} = j-1\right) - \mathrm{E}\left(S_t \middle| S_{t-1} = j-2\right), \qquad j = 3, ..., m$$

(segundos acréscimos nulos).

4) A partir de 3) deduzem-se as restrições lineares associadas aos $p_{ij} = P\left(S_t = j \middle| S_{t-1} = i\right)$. Seja $\mathbf{R}$ essa matrix (vector no caso $m = 3$).

5) Seja $\phi = (p_{11}, p_{12}, ..., p_{1m}, ..., p_{mm})'$ um vector de dimensão $m^2 \times 1$. Dado

$$\sqrt{n}\left(\hat{\phi} - \phi\right) \xrightarrow{d} N\left(0, \mathbf{\Sigma}\right), \qquad \mathbf{\Sigma}: m^2 \times m^2$$

tem-se, sob $H_0 : \mathbf{R}\phi = \mathbf{0}$

$$n\left(\hat{\phi}\mathbf{R}\right)' \left(\mathbf{R\Sigma R}'\right)^{-1} \phi\mathbf{R} \xrightarrow{d} \chi^2_{(linhas\ \mathbf{R})}$$

► Formato

{w,pvalue,x0,cond_mean,cond_var}=testing_linearity_01(x,quant,branquear);

► Input

x:

quant: (ver objectivo)

branquear: X é subtituído pelos resíduos AR(1)

▶ **Output**

w:

pvalue:

x0: pontos médios de cada classe ou categoria (x0=.5*(minc(x)|q)+.5*(q|maxc(x)); onde q são os quantis)

cond_mean: Px0

cond_var: Px0^2 **–** (**Px0**)^2

▶ **Library**

library testes;

▶ **Fonte**

c:\gauss\srcnic\testes\markov_chain.src;

17 Teoria dos Valores Extremos [tve]

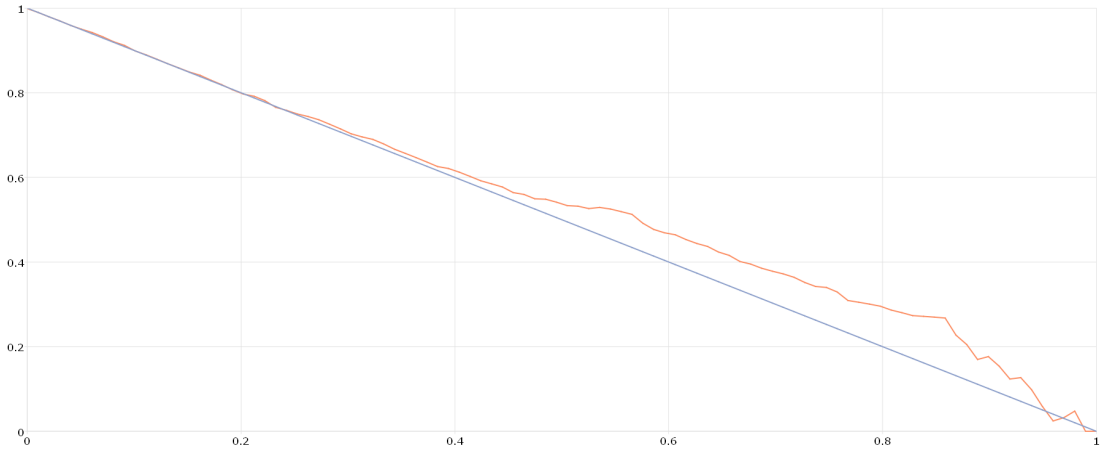
17.1 dep_measure1

▶ Objective

Estimate

$$Q(p) = P\left(Y > F_Y^{-1}(p) \middle| X > F_X^{-1}(p)\right)$$

and presents the figure



▶ Format

{q,prob}=dep_measure1(y,x,q0,figure);

▶ Input

y:

x:

q0: If q0=0 then q0 is specified as q0=seqa1(0,1,100);

figure: If figure =0 then the above graph is presented.

▶ Output

▶ Library

library tve;

▶ Fonte

c:\gauss\srcnic\tve\dependence.src

17.2 dep_measure2

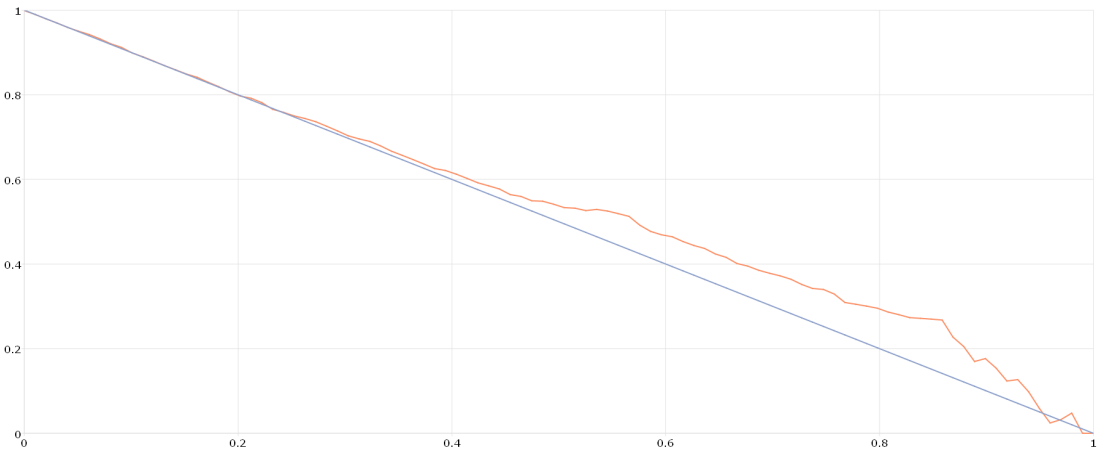
▶ Objective

Gives the same result as **dep_measure1**.

Estimate

$$Q(p) = P(F(T) > p | F(S) > p)$$

where $T = -1/\log(F_Y(Y))$, $S = -1/\log(F_X(X))$ and $F(s) = e^{-1/s}$ and presents the figure.



$F_Y(Y)$ and $F_X(X)$ are estimated using the empirical distribution.

Note

$$\begin{aligned} P(S \le y) &= P(-1/\log(F_X(X)) \le y) \\ &= P\left(F_X(X) \le e^{-1/y}\right) \\ &= P\left(X \le F_X^{-1}\left(e^{-1/y}\right)\right) \\ &= e^{-1/y}. \end{aligned}$$

▶ Format

{q,prob}=dep_measure2(y,x,q0,figure);

▶ Input

y:

x:

q0: If q0=0 then q0 is specified as q0=seqa1(0,1,100);

figure: If figure =0 then the above graph is presented.

▶ **Output**

▶ **Library**

library tve;

▶ **Fonte**

c:\gauss\srcnic\tve\dependence.src

17.3 estima_GEV

▶ Objectivo

Estima os parâmetros λ, δ, α da Generalized Extreme Value Distribution:

$$G(x) = \exp\left\{-\left(1+\frac{1}{\alpha}\left(\frac{x-\lambda}{\delta}\right)\right)^{-\alpha}\right\}, \qquad 1+\frac{1}{\alpha}\left(\frac{x-\lambda}{\delta}\right) > 0$$

.

▶ Formato

{b_GEV,f0,grad,cov,retcode}=estima_GEV(b0,y);

▶ Input

b0: Se b0=0 => b0=2|meanc(y)|stdc(y); b0=alpha|lamda|delta

y: vector das observações;

▶ Output

▶ Library

library est_etd, cml;

▶ Fonte

c:\gauss\srcnic\est_etd\estima_GEV.src

17.4 estimador_hill

▶ Purpose

Computes Hill's estimator:

$$\hat{\alpha}(q) = \frac{n(q)}{\sum_{t=1}^{n} \log(y_t/q)\, \mathcal{I}_{\{y_t>q\}}}, \qquad n(q) = \sum_{t=1}^{n} \mathcal{I}_{\{y_t>q\}}$$

where $\mathcal{I}_{\{y_t>q\}} = 1$ if $y_t > q$ and $\mathcal{I}_{\{y_t>q\}} = 0$ otherwise. The estimator is computed for both tails. Under some conditions one has

$$\sqrt{n(q)}\,(\hat{\alpha}(q) - \alpha(q)) \xrightarrow{d} N\left(0, \alpha^2\right)$$

when $n, n(q) \to +\infty$ and $n(q)/n \to 0$. Notice that $\mathrm{Var}(\hat{\alpha}(q)) = \alpha^2/n(q)$.

▶ Format

{nq,hill,racio_t}=estimador_hill(y,prob,mostrar);

▶ Input

y: $n \times 1$ vector

prob: scalar or vector with elements belonging to the interval $(0, 1)$. If *prob* is a scalar, one selects all observations such that

$$y_t > q_{prob}$$

where q_{prob} is a quantile. If *prob* is a vector, the procedure is repeted leading to various estimators $\hat{\alpha}(q)$ (as many as the dimension of *prob*). You may consider prob=.96|.97|.98; to start with.

mostrar:

▶ Output

nq: número de observações que efectivamente entram no cálculo do estimador.

hill: estimador

racio_t:

▶ Library

library tve;

▶ Fonte

c:\gauss\srcnic\tve\estimacao.src

17.5 reg_rank_size

▶ Purpose

Computes $\hat{\alpha}$ where α is the tail index from the regression

$$\log(t-\gamma) = c - \alpha \log X_{(t)} + error_t$$

where γ takes the values 0 and 1/2 and $X_{(1)} \geq X_{(2)} \geq ... \geq X_{(n)}$ (see Gabaix, X. and R. Ibragimov (2012), Log(Rank-1/2): A Simple Way to Improve the OLS Estimation of Tail Exponents. *Journal of Business Economics and Statistics.*)

▶ Format

{alfa_gamma0,alfagamma1_2,resid,x}=reg_rank_size(y,k);

▶ Input

y: $n \times 1$ vector

k: k is such that $m = [nk]$, or $(0 < k < 1)$ (fraction of observations used)

▶ Output

alfa_gamma0: $\hat{\alpha}$ with $\gamma = 0$

alfagamma1_2 $\hat{\alpha}$ with $\gamma = 1/2$

resid: Residuals from the regression $\log(t-1/2) = c - \alpha \log X_{(t)} + error_t$

x: $c - \alpha \log X_{(t)}$

▶ Library

library tve;

▶ Fonte

c:\gauss\srcnic\tve\estimacao.src

17.6 reg_1

▶ Purpose

Computes $\hat{\alpha}$ where α is the tail index $\bar{F}(x) = ax^{-\alpha}\left(1 + bx^{-\beta} + o\left(x^{-\beta}\right)\right)$ from the regression

$$y_i = \vartheta + \alpha \tilde{z}_i + \varepsilon_i, \qquad i = 1, 2, ..., m-1$$

where $y_i := \log \bar{F}_n(x_i)$, $x_i := (1-u_i)^{-\frac{1}{\tilde{\alpha}}} x_0$, $\hat{x}_0 := \hat{F}^{-1}(1-\kappa)$, $0 < \kappa < 1$, $\vartheta := (\alpha - 1)\log x_0$ and $z_i := \tilde{\alpha}^{-1}\log(1-u_i)$ and $\tilde{\alpha}$ is the Hill estimator (see Nicolau and Rodrigues, 2015).

▶ Format

{alfaHill,alfaReg1}=reg_1(y,k);

▶ Input

y: $n \times 1$ vector

k: k is such that $m = [nk]$, or $(0 < k < 1)$ (fraction of observations used)

▶ Output

alfaHill: Hill's estimator

alfaReg1: $\hat{\alpha}$

▶ Library

library tve;

▶ Fonte

c:\gauss\srcnic\tve\estimacao.src

17.7 ParetoX_01

▶ Objectivo

Estima $\boldsymbol{\beta}$ no modelo

$$\begin{aligned} y_i|\,\mathbf{x}_i &\sim Pareto\left(x_0, \alpha\left(\mathbf{x}_i\right)\right), \qquad y_i > x_0 \\ \alpha\left(\mathbf{x}_i\right) &= \exp\left(\mathbf{x}_i'\boldsymbol{\beta}\right) \end{aligned}$$

Admite-se que $y_i|\,\mathbf{x}_i$ possa ter apenas caudas de Pareto. Neste caso considerar $k < 1$ (ver abaixo). Para a aba esquerda fazer a transformação $-y_i$.

▶ Formato

{b,f0,grad,cov,retcode}=ParetoX_01(y,x,k);

▶ Input

y: vector das observações de dimensão $n \times 1$;

x: matriz das variavéis explicativas de dimensão $n \times K$, sendo K o número de variáveis explicativas. Incluir uma coluna de uns para estimar o termo independente.

k: fracção de observações a serem utilizadas na aba direita (ou esquerda) para efeitos de estimação (para a aba esquerda fazer a transformação $-y_i$). Se $y_i|\,\mathbf{x}_i$ tiver distribuição exacta de Pareto considerar $k = 1$ (todas as observações são usadas).

▶ Output

▶ Library

library cml,tve;

▶ Fonte

c:\gauss\srcnic\tve\estima_paretox.src

17.8 ParetoX_02

▶ Objectivo

Estima $\boldsymbol{\beta}$ e σ no modelo

$$\begin{aligned} y_i|\,\mathbf{x}_i &\sim ParetoII\left(x_0,\sigma,\alpha\left(\mathbf{x}_i\right)\right), \qquad y_i > x_0 \\ \alpha\left(\mathbf{x}_i\right) &= \exp\left(\mathbf{x}_i'\boldsymbol{\beta}\right) \end{aligned}$$

A fdp é

$$f\left(y_i|\,\mathbf{x}_i\right) = \alpha\left(\mathbf{x}_i\right)\sigma^{-1}\left(1+\frac{y_i-x_0}{\sigma}\right)^{-(\alpha(\mathbf{x}_i)+1)}$$

e a função log-verosimilhança é

$$L\left(\boldsymbol{\beta},\sigma\right) = \sum_i \mathbf{x}_i'\boldsymbol{\beta} - \log\sigma - \left(\exp\left(\mathbf{x}_i'\boldsymbol{\beta}\right)+1\right)\log\left(1+\frac{y_i-\min_i\left\{y_i\right\}}{\sigma}\right)$$

Admite-se que $y_i|\,\mathbf{x}_i$ possa ter apenas caudas de ParetoII. Neste caso considerar $k<1$ (ver abaixo)

▶ Formato

{b,f0,grad,cov,retcode}=ParetoX_02(y,x,k);

▶ Input

y: vector das observações de dimensão $n \times 1$;matriz das variavéis explicativas de dimensão $n \times K$, sendo K o número de variáveis explicativas. Incluir uma coluna de uns para estimar o termo independente.

k: fracção de observações a serem utilizadas na aba direita para efeitos de estimação (para a aba esquerda fazer a transformação $-y_i$). Se $y_i|\,\mathbf{x}_i$ tiver distribuição exacta de Pareto II considerar $k=1$ (todas as observações são usadas).

▶ Output

▶ Library

library cml,tve;

▶ Fonte

c:\gauss\srcnic\tve\estima_paretox.src

18 User [.]

18.1 binomial

▶ **Purpose**

$$\binom{a}{b} = \frac{a!}{b!\,(a-b)!}$$

▶ **Format**

v=binomial(a,b)

▶ **Input**

▶ **Output**

▶ **Library**

library não é necessário

▶ **Source**

c:\gauss\srcnic\funcoes.src

18.2 delta_kron

▶ **Purpose**

$$\delta_{ij} = \begin{cases} 1\, i = j \\ 0\, i \neq j \end{cases}$$

▶ **Format**

{delta} = delta_kron(i,j);

▶ **Input**

i: escalar

j: escalar

▶ **Output**

delta: zero ou um

▶ **Library**

library não é necessário

▶ **Source**

c:\gauss\srcnic\funcoes.src

18.3 gradp1

▶ **Objectivo**

Calculates the first derivative with respect to the first argument of the function.

▶ **Formato**

d=gradp1(&f,x0,y0));

▶ **Input**

▶ **Output**

▶ **Library**

▶ **Fonte**

c:\gauss\srcnic\user\derivada.src

▶ **Example**

```
x = 1|2|3|4;

b= 10|20;

dx=gradp1(&a,x,b);

dy=gradp2(&a,x,b);

fn a(x,b)=b[1]+b[2]*x^2;
```

$$dx = diag(40, 80, 120, 160) = \begin{bmatrix} \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial x_1} & 0 & 0 & 0 \\ 0 & \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial x_2} & 0 & 0 \\ 0 & 0 & \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial x_3} & 0 \\ 0 & 0 & 0 & \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial x_4} \end{bmatrix}$$

$$dy = \begin{bmatrix} 1 & 1 \\ 1 & 4 \\ 1 & 9 \\ 1 & 16 \end{bmatrix} = \begin{bmatrix} \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial b_1} & \frac{\partial a(\mathbf{x},\mathbf{b})}{\partial b_2} \end{bmatrix}$$

18.4 gradp2

▶ **Objectivo**

Calculates the first derivative with respect to the second argument of the function.

▶ **Formato**

d=gradp2(&f,x0,y0));

▶ **Input**

▶ **Output**

▶ **Library**

▶ **Fonte**

c:\gauss\srcnic\user\derivada.sr

18.5 seqa1

▶ **Objectivo**

▶ **Formato**

seqa1(x0,x1,n);

▶ **Input**

x0: initial value

x1: last value

n: vector's dimension

▶ **Output**

▶ **Library**

▶ **Fonte**

c:\gauss\srcnic\user\seqa1.sr

19 Utilidades [util]

19.1 acerta_datas_2

▶ Objectivo

Acertar as datas de duas séries. Os valores das séries passam a reportar-se aos mesmos momentos cronológicos.

▶ Formato

{data,y1,y2}=acerta_datas_2(data1,y1,data2,y2);

▶ Input

data1: vector das datas da primeira série.

y1: data numérica (por exemplo, 38343) associada a y1

data2: vector das datas da segunda série.

y2: data numérica (por exemplo, 38343) associada a y2

▶ Output

data: vector das datas comuns às duas séries

y1: vector y1 reformatado (são eliminados os valores cujas datas não constem de data2)

y2: vector y2 reformatado (são eliminados os valores cujas datas não constem de data1)

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\util1.src

19.2 acerta_datas_3

▶ Objectivo

Acertar as datas de três séries. Os valores das séries passam a reportar-se aos mesmos momentos cronológicos. ** confirmar rotina **

▶ Formato

{data,y1,y2,y3}=acerta_datas_3(data1,y1,data2,y2,data3,y3);

▶ Input

data1: vector das datas da primeira série.

y1:

data2: vector das datas da segunda série.

y2:

data3:

y3:

▶ **Output**

data: vector das datas comuns às duas séries

y1: vector y1 reformatado (são eliminados os valores cujas datas não constem de data2 e data3)

y2: vector y2 reformatado (são eliminados os valores cujas datas não constem de data1 e data3)

y3: vector y3 reformatado (são eliminados os valores cujas datas não constem de data1 e data2)

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.3 acerta_datas_5

▶ **Objectivo**

Acertar as datas de 10 séries. Os valores das séries passam a reportar-se aos mesmos momentos cronológicos.

▶ **Formato**

{data,y}=acerta_datas_5(data1,y1,data2,y2,data3,y3,data4,y4,data5,y5);

▶ **Input**

data1: vector das datas da primeira série.

y1:

data2: vector das datas da segunda série.

y2:

data3:

y3:

etc.

▶ Output

data: vector das datas comuns a todas as séries

y: y1~y2~y3~y4~y5

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\util1.src

19.4 acerta_datas_10

▶ Objectivo

Acertar as datas de 10 séries. Os valores das séries passam a reportar-se aos mesmos momentos cronológicos.

▶ Formato

{data,y}=acerta_datas_10(data1,y1,data2,y2,data3,y3,data4,y4,data5,y5

,data6,y6,data7,y7,data8,y8,data9,y9,data10,y10);

▶ Input

data1: vector das datas da primeira série.

y1:

data2: vector das datas da segunda série.

y2:

data3:

y3:

etc.

▶ Output

data: vector das datas comuns a todas as séries

y: y1~y2~y3~y4~y5~y6~y7~y8~y9~y10

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\util1.src

19.5 agrega_observ_media_01

▶ Objectivo

Agrega valores de frequência alta para valores de frequência mais baixa, calculando as respectivas médias.

▶ Formato

{dataF,z,c}=agrega_observ_media_01(data0,data,y,agr);

▶ Input

data0: data inicial para iniciar a agregação; se "data0=min(data);"a agregação inicia-se com o primeiro valor de y.

data: data numérica (por exemplo, 38343) associada a y

y: vector ou matriz

agr: para dados diários, agr=5, passa para médias semanais. Se agr=20 passa para médias mensais.

▶ Output

dataF:

z: médias agregadas

c: número de observações, em cada sub período, usadas para calcular a média.

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\util1.src

19.6 agrega_observ_media_02

▶ **Objectivo**

Calcula médias mensais.

▶ **Formato**

{ym,data}=agrega_observ_media_02(y,ano,mes);

▶ **Input**

y: vector ou matriz

ano:

mes:

▶ **Output**

ym: médias mensais

data: ano ~mes

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.7 arctanh

▶ **Objectivo**

Calcula o arco cuja a tangente hiperbólica é x, arctanh (x) para $|x| < 1$.

▶ **Formato**

y=arctanh(x);

▶ **Input**

x: vector

▶ **Output**

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.8 bootstrap

▶ **Objectivo**

Dado um vector x, de dimensão n, faz uma amostra de dimensão n com reposição de x.

▶ **Formato**

y=bootstrap(x);

▶ **Input**

x:

▶ **Output**

y:

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.9 bootstrap1

▶ **Objectivo**

Dado um vector x, de dimensão n, faz uma amostra de dimensão n com reposição de x.

▶ **Formato**

y=bootstrap(x,n);

▶ **Input**

x:

n: dimensão do bootstrap

▶ **Output**

y:

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.10 categoriza_01

▶ **Purpose**

Transforms the values of a matrix into categories.

▶ **Format**

S=categoriza_01(r,q);

▶ **Input**

r: vector $n \times K$

q: vector $(m-1) \times 1$, where m is the number of categories.

▶ **Output**

s: vector $n \times K$

▶ **Exemple**

Suppose

$$r = \begin{bmatrix} 0.5 \\ -2 \\ 3 \\ 0 \end{bmatrix}, \qquad q = 0$$

The data is divided into the following intervals $(-\infty, 0)$ e $(0, +\infty)$. Thus

$$S = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 2 \end{bmatrix}.$$

With the same r vector, suppose now that

$$q = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

The data is divided into the following intervals $(-\infty, -1)$, $(-1, 1)$ e $(1, +\infty)$. Hence,

$$S = \begin{bmatrix} 2 \\ 1 \\ 3 \\ 2 \end{bmatrix}.$$

The following instructions allow to discretize a continuous variable into a variable with 10 categories according to the percentiles.

```
quant=.1|.2|.3|.4|.5|.6|.7|.8|.9;

q=quantile(r,quant);

S=categoriza_01(r,q);
```

▶ **Library**

library cm; ou library est_etd; ou library util;

▶ **Fonte**

c:\gauss\srcnic\est_etd\cm.src

c:\gauss\srcnic\util\util2.src

19.11 categoriza_02

▶ **Objectivo**

Transforma os valores de uma variável em categorias usando dummies

▶ **Formato**

S=categoriza_02(r,q);

▶ **Input**

r: vector $n \times K$

q: vector $(m-1) \times 1$, sendo m o número de categorias.

▶ **Output**

s: vector $n \times K$

▶ **Exemplo**

Suponha-se

$$r = \begin{bmatrix} 0.5 \\ -2 \\ 3 \\ 0 \end{bmatrix}, \qquad q = 0$$

As classes a considerar são $-\infty, 0$ e $(0, +\infty)$.

Logo

$$S = \begin{bmatrix} 0\,1 \\ 1\,0 \\ 0\,1 \\ 0\,1 \end{bmatrix}.$$

Com o mesmo vector r suponha-se agora

$$q = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

As classes a considerar são $(-\infty, -1)$, $(-1, 1)$ e $(1, +\infty)$.

Logo

$$S = \begin{bmatrix} 0\,1\,0 \\ 1\,0\,0 \\ 0\,0\,1 \\ 0\,1\,0 \end{bmatrix}$$

► **Library**

library cm; ou library util;

► **Fonte**

c:\gauss\srcnic\est_etd\cm.src

c:\gauss\srcnic\util\util2.src

19.12 com

► **Objectivo**

Calcular

$$\binom{n}{p} = \frac{\Gamma(n+1)}{\Gamma(p+1)\,\Gamma(n-p+1)}$$

► **Formato**

y=com(n,p);

► **Input**

n: escalar

p: escalar

► **Output**

y: escalar

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.13 convert_daily_to_monthly1

▶ **Objectivo**

Passar de dados diários para dados mensais (último dia do mês)

▶ **Formato**

y=convert_daily_to_monthly1(z);

▶ **Input**

z: z=timeseries~day~month~year

▶ **Output**

y: y=timeseries~day~month~year (apenas ultimo dia de cada mês)

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.14 cross_corr

▶ **Purpose**

Estimates

$$Corr\left[X_{1t}, X_{2,t-p}\right], ..., Corr\left[X_{1t}, X_{2,t}\right], ..., Corr\left[X_{1t}, X_{2,t+p}\right] \qquad (*)$$

where

$$Corr\left[X_{1t}, X_{2,t-p}\right] = \frac{Cov\left(X_{1t}, X_{2,t-p}\right)}{\sqrt{Var\left[X_1\right]}\sqrt{Var\left[X_2\right]}}$$

▶ **Format**

{lags,cr} = cross_corr(x1,x2,p);

▶ **Input**

x1: $n \times 1$ vector of observations

x2: $n \times 1$ vector of observations

p: see "purpose"

▶ Output

lags: sequence $\{-p, -p+1, ...0, 1, ...p-1, p\}$

cr: sequence (*)

▶ Observation

X_1 and X_2 should be two stationary whitened processes. Note that $Corr\left[X_{1t}, X_{2,t+p}\right]$ can be interpreted as $Corr\left[X_{1,t-p}, X_{2t}\right]$

▶ Library

library util,pgraph;

▶ Source

c:\gauss\srcnic\util\cross_corr.src

19.15 cumulative_chi_square_noncentral

▶ Objectivo

Calcular $F\left(x; df, \lambda\right) = P\left(X \leq x\right)$ onde X segue uma distribuição do qui-quadrado não central com df graus de liberdade e com parâmetro não central λ

▶ Formato

F=cumulative_chi_square_noncentral(x,df,lam);

▶ Input

x: vector de tipo $n \times 1$.

df: graus de liberdade

lam: parâmetro não central

▶ Output

F: vector de tipo $n \times 1$.

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.16 desfas1

▶ Objectivo

Transformar um vector z de tipo $n \times 1$ numa matriz do tipo

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ z_1 & 0 & \cdots & 0 \\ z_2 & z_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ z_{n-1} & z_{n-2} & \cdots & z_{n-m} \end{bmatrix}$$

▶ Formato

y=desfas1 (z,m);

▶ Input

z: vector de tipo $n \times 1$.

m: escalar inteiro positivo tal que $n > m$.

▶ Output

y: matriz de tipo $n \times m$.

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\matrizes.src

19.17 desfas2

▶ Objectivo

Transformar um vector z de tipo $n \times 1$ numa matriz do tipo

$$y = \begin{bmatrix} \cdot & \cdot & \cdots & \cdot \\ z_1 & \cdot & \cdots & \cdot \\ z_2 & z_1 & \cdots & \cdot \\ \vdots & \vdots & \ddots & \vdots \\ z_{n-1} & z_{n-2} & \cdots & z_{n-m} \end{bmatrix}$$

Nota: "$\cdot$" são valores *missing*.

▶ Formato

y=desfas2(z,m);

▶ **Input**

z: vector de tipo $n \times 1$.

m: escalar inteiro positivo tal que $n > m$.

▶ **Output**

y: matriz de tipo $n \times m$.

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\matrizes.src

19.18 desfas3

▶ **Objectivo**

Desfasa o vector z de tipo $n \times 1$ de acordo com o vector m. Por exemplo "m=1|3|5;" Vem

$$y = lagn(z, 1)\tilde{}lagn(z, 3)\tilde{}lagn(z, 5)$$

y é um vector $n \times 1$ com valores *missing* no início.

▶ **Formato**

y=desfas3(z,m);

▶ **Input**

z: vector de tipo $n \times 1$.

m: vector de dimensão m

▶ **Output**

y: matriz de tipo $n \times m$.

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\matrizes.src

19.19 desfas4

▶ Objectivo

Multiplica valores desfasados de z de acordo com a matriz par. Por exemplo,

$$par = \begin{bmatrix} 1\,0\,0 \\ 1\,2\,0 \\ 2\,3\,5 \end{bmatrix}.$$

Dado z[1:i] vem y=z[i-1] ~z[i-1]*z[i-2] ~z[i-2]*z[i-3]*z[i-5] → escalar

▶ Formato

y=desfas4(z,m);

▶ Input

z: vector de tipo $n \times 1$.

par: matriz $k \times m$

▶ Output

y: vector linha de dimensão k

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\matrizes.src

19.20 est_desc

▶ Objectivo

Calcular e apresentar as seguintes estatísticas descritivas: média, variância, *skewness* e *kurtosis*.

▶ Formato

{m,var,sk,k,min,max}=est_desc(y);

▶ Input

y: vector das observações.

▶ Output

m: média de y.

var: variância de y.

sk: *skewness* de y.

k: *kurtosis* de y.

min: $\min_{i=1,..,n} \{y_i\}$

max: $\max_{i=1,..,n} \{y_i\}$

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

▶ **Observações**

Variável global: _mostrar=1. Se _mostrar<>1 não apresenta os resultados.

19.21 erf1

▶ **Objectivo**

Calcula o erf (x) para qualquer x real (GAUSS 5.0 assume $x \geq 0$)

▶ **Formato**

y=erf1(x);

▶ **Input**

▶ **Output**

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.22 erfi

▶ **Objectivo**

Calcula o erfi $(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2 dt}$

▶ **Formato**

y=erfi(x);

▶ **Input**

x: escalar

▶ **Output**

y: escalar

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.23 frequencias

▶ **Objectivo**

Calcular frequências

▶ **Formato**

{p,m,freq }=frequencias(x,v);

▶ **Input**

x: vector das observações.

v: número de classes ou Nx1 vector, the breakpoints to be used to compute the frequencies

▶ **Output**

p: vector vx1, limites inferiores e superiores em cada classe

m: vector vx1, pontos médios de cada classe

freq: vector vx1, frequência de cada classe

▶ **Observações**

$x \leq b[1]$	freq[1]
$b[1] < x \leq b[2]$	freq[2]
...	....
$b[v-1] < x \leq b[v]$	freq[v]

If a vector of breakpoints is specified, a final breakpoint equal to the maximum value of x will be added if the maximum breakpoint value is smaller.

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.24 funções

Para as rotinas definidas na tabela abaixo é válido o seguinte:

▶ **Input**

par: vector θ.

x: vector de tipo de $k \times 1$.

▶ **Output**

y: vector de tipo de $k \times 1$.

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes1.src

Algumas Funções Úteis

Objectivo (calcular:)	**Formato**
$\exp\left\{\theta_1/2 + \theta_2/2\,(x-\theta_3)^2\right\}$	y = exponencial(par,x);
$-\exp\{\theta_1 + \theta_2\,(x-\theta_5)\} + \exp\{\theta_3 - \theta_4\,(x-\theta_5)\}$	y = exponencial1(par,x);
$e^{\theta_1}\left(\exp\{-\theta_2\,(x-\theta_4)\} - \exp\{\theta_3\,(x-\theta_4)\}\right)$	y = exponencial2(par,x);
$\exp\left\{\theta_1/2\,\lvert x-\theta_2\rvert^3\right\}$	y = exponencial3(par,x);
$\theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4/x$	y=func1(par,x);
$\theta_1 + \theta_2 x + \theta_3 x^{\theta_4}$	y=func2(par,x);
$\sqrt{\theta_1 + \theta_2 x + \theta_3 x^2}$	y = geral(par,x);
$\theta_1 + \theta_2 x$	y = linear(par,x);
$\theta_1\,(\theta_2 - x)$	y = linear1(par,x);
$x\,(\theta_1 - \theta_2 x)$	y = logis(par,x);
$\theta_1\,\lvert x\rvert^{\theta_2}$	y = potencia(par,x),
$\theta\sqrt{x}$	y=potencia1(par,x)
$\sqrt{\theta_1 + \theta_2\,(x-\theta_3)^2}$	y = quadratica(par,x);
$\sqrt{\theta_1 + \theta_2 x}$	y=sqrtlinear(par,x);

Por exemplo,

```
proc linear1(a,x);

    local z;

    z=a[1]*(a[2]-x);

retp(z);

endp;
```

19.25 ierf

▶ Objectivo

Calcula o inverso da função erf (x)

▶ Formato

y=ierf(x);

▶ Input

▶ Output

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\funcoes2.src

19.26 cdf_logistic

▶ Objective

Calculates

$$G(x) = \frac{1}{1 + e^{-\gamma(x-m)}}.$$

Note: if $X \sim Logistic$ then $\mathrm{E}(X) = m$ and $\mathrm{Var}(X) = \pi / \left(3\gamma^2\right)$

▶ Formato

y=cdf_logistic(x,m,g);

▶ Input

x:

m:

s:

▶ Output

▶ Library

library util;

▶ Fonte

c:\gauss\srcnic\util\funcoes2.src

19.27 lngammaif

▶ Objective

Calculates $\log(gamma(a + bi))$ where $i = \sqrt{-1}$ using the formula (11), page 697 of Eic Weisstein (Concise encyclopedia of Mathematics)

▶ Formato

y=lngammaif(a,b,n);

▶ Input

a:

b:

n: precision. The higher is n the greater is the precision (choose $n > 100000$)

▶ **Output**

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.28 max_mensal

▶ **Objective**

Calculates the maximum of month from daily observations and registers the periods at which the maximums are observed.

▶ **Format**

{max,tmax}=max_mensal(x,mes,d);

▶ **Input**

x: daily observations

mes: month of the year

d: step of discretization

▶ **Output**

max:

tmax: periods at which the monthy maximums are observed

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util3.src

19.29 mod_bessel_1

▶ **Objectivo**

▶ **Formato**

y=mod_bessel_1(v,z);

▶ **Input**

▶ **Output**

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\funcoes2.src

19.30 momentos_moveis

▶ **Objectivo**

Calcular $m_t(k) = \frac{1}{h}\sum_{i=t-h+1}^{t} X_i$ e $\sigma(k) = \sqrt{\frac{1}{h-1}\sum_{i=t-h+1}^{t}(X_i - m_t(k))^2}$para $k = 0, 1, ..., n-h$.

▶ **Formato**

{m,s}=momentos_moveis(x,h);

▶ **Input**

x: vector das observações de X (de tipo $n \times 1$).

h: valor de alisamento $1 < h < n$.

▶ **Output**

m: vector das médias móveis de tipo $n \times 1$ (primeiros $h-1$ valores são missings).

s: vector dos desvios-padrão móveis de tipo $n \times 1$ (primeiros $h-1$ valores são missings).

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.31 rnd_int

▶ **Objectivo**

Simula um inteiro, com distribuição uniforme discreta, no intervalo $[a, b]$

▶ **Formato**

e=rnd_int(a,b);

▶ **Input**

a: escalar

b: escalar maior do que a

▶ **Output**

e:

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.32 select1

▶ **Objectivo**

Selecciona aleatoriamente um valor do vector x

▶ **Formato**

y=select1(x);

▶ **Input**

x: vector

▶ **Output**

y:

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

19.33 triang

▶ Objectivo

Transformar vectores em matrizes $M = [m_{ij}]$ com zeros nos elementos $i < j$.

▶ Formato

x = triang(vector,c);

▶ Input

vector: vector de tipo $k \times 1$

colunas: escalar inteiro positivo tal que $k \geq c$

▶ Output

x: matriz de tipo $k \times c$

▶ Exemplo

vector = {5,2,7};

x = triang(vector,3);

x é igual a

$$x = \begin{bmatrix} 5\,0\,0 \\ 2\,5\,0 \\ 7\,2\,5 \end{bmatrix}.$$

▶ Fonte

c:\gauss\srcnic\util\matrizes.src

19.34 var_dummy

▶ Objectivo

Criar uma variável dummy do tipo

$$\begin{matrix} 0 & \\ 1 & \rightarrow \text{inicio} = 2 \\ 0 & \\ 0 & \\ 1 & \rightarrow \text{periodo} = 3 \\ 0 & \\ 0 & \\ 1 & \rightarrow \text{periodo} = 3 \\ 0 & \\ 0 & \end{matrix}$$

▶ **Formato**

d=var_dummy(n,inicio,periodo);

▶ **Input**

n: numero de linhas (de tipo $n \times 1$).

inicio: ver exemplo

periodo: ver exemplo

▶ **Output**

d: ver exemplo

s:

▶ **Library**

library util;

▶ **Fonte**

c:\gauss\srcnic\util\util1.src

20 Value at Risk [var]

20.1 hybrid_approach_01

▶ Objectivo

Calcula o VaR de acordo com a metodologia de Boudoukh, Richardson e Whitelaw (1998).

▶ Formato

{q} = hybrid_approach_01(x,janela,lam,ordem);

▶ Input

x: vector das observações do processo.

janela: Escalar

lam: Escalar.

ordem: Escalar. Ordem do quantil

▶ Output

q: vector de dimensão igual a x e que corresponde ao quantis da ordem fixada em "ordem". As primeiras observações (de 1 a janela) são "missings".

▶ Library

library var;

▶ Fonte

c:\gauss\srcnic\var\met_np.src

21 Outros [...]

21.1 Qreg

▶ Objectivo

Linear quantile regression.

▶ Formato

{beta,u_plus,u_minus} = Qreg(Y,X,tau,w);

▶ Input

y: N*1 vector, dependent variable

x: N*K matrix, independent variables

tau: M*1 vector, quantile levels

w: Nx1 vector, containing weights – or – 0 for uniform weights

▶ Output

beta: (K+1)*M matrix, estimated parameters

u_plus: N*M matrix, positive part of residuals

u_minus: N*M matrix, negative part of residuals

▶ Library

library qreg;

▶ Observações

** Globals:

** __altnam - vector, variable names (default = 0)

** __output - scalar, (default = 1)

** 1 - print the statistics

** 0 - no printing

** _Qreg_algr - scalar, LP algorithm (default = 1)

** 1 for the interior-point method

** 2 for the QP method

Ver também o procedimento q = localQreg(Y,X,tau,xstar); Purpose: Local quantile regression.

Authors: D. Jacomy, J. Messines and T. Roncalli

http://gro.creditlyonnais.fr/content/rd/home_gauss.htm

Ver o manual em c:\...\srcqreg\

▶ Fonte

c:\...\srcqreg\qreg.src

21.2 kpss

▶ Objectivo

Compute the KPSS nu and tau statistics.

Kwiatkowski, Phillips, Schmidt and Shin [1992], Testing the null hypothesis of stationarity against the alternative of a unit root: how sure are we that economic series have a unit root, Journal of Econometrics, 54, 159-178

▶ Formato

{nu,tau} = kpss(x,order);

▶ Input

x: Nobs*1 vector, data.

order: scalar, maximum lag tested.

▶ Output

nu: (order+1)*1 vector, the NU statistic values.

order: (order+1)*1 vector, the TAU statistic values.

▶ Library

library tsm;

▶ Fonte

c:\...\srctsm\kpss.src

21.3 gera_AR

▶ Objectivo

```
*
***
*** gera_AR(x,const,ordem_x,par_x,y0,par_y,sigma,s);
***
sigma=-999 —>não existe componente aleatória
–>      se p>=q,
    –>      para t=1,...,p
                y(1)=y0[1]
                ...
                y(p)=y0[p]
    –>      par t=p+1,...,n
                y(t)=par_x[1]*x(t-ordem_x[1])+...+par_x[q]*x(t-ordem_x[q])+
                    +par_y[1]*y(t-1)+...+par_y[p]*y(t-p)+sigma*rndn(n,s)
–>      se q>p
    –>      para t=1,...,q
                y(1) =0
                ...
                y(q-p) =0
                y(q-p+1)=y0[1]
                ...
                y(q) =y0[p]
    –>      para t=q+1,...,n
                y(t)=par_x[1]*x(t-ordem_x[1])+...+par_x[q]*x(t-ordem_x[q])+
                    +par_y[1]*y(t-1)+...+par_y[p]*y(t-p)+sigma*rndn(n,s)
        nota: neste último caso apresenta-se apenas y(t) para t=q-p+1,...,n
    */
```

► **Formato**

```
y=gera_AR(x,const,ordem_x,par_x,y0,par_y,sigma,s);
```

► **Input**

▶ **Output**

▶ **Library**

?

▶ **Fonte**

22 Anexos

22.1 Gráficos pgraph

EXEMPLO:

```
graphset;
begwind;
_pdate=;
fonts ("simplex complex microb simgrma");
window(3,2,0);
_ptitlht=.3; /* dimensão titulos*/
_pnumht=.2; /* dimensão dos valores dos eixos */
_plegctl={2,5,.5,3.6}; /* dimensão das legendas e posição*/
title ("Condit. Density given x]0[=-1 (\204\68 \201= 0.5)");
_plegstr="True Dens. \000"\
"Proposed Dens. n000"\
"Euler Dens. \000";
xy(x[.,1],x[.,2:4]);
nextwind;
title ("Condit. Distribution Function given x]0[=-1 (\204\68 \201= 0.5)");
_plegstr="True DF \000"\
"Proposed DF \000"\
"Euler DF \000";
xy(x[.,1],x[.,5:7]);
nextwind;
title ("Condit. Density given x]0[=0 (\204\68 \201= 0.5)");
_plegstr="True Dens. \000"\
"Proposed Dens. \000"\
"Euler Dens. \000";
xy(x[.,8],x[.,9:11]);
```

```
nextwind;
title ("Condit. Distribution Function given x]0[=0 (\204\68 \201= 0.5)");
_plegstr="True DF \000"\
"Proposed DF \000"\
"Euler DF \000";
xy(x[.,8],x[.,12:14]);
nextwind;
title ("Condit. Density given x]0[=1 (\204\68 \201= 0.5)");
_plegstr="True Dens. \000"\
"Proposed Dens. \000"\
"Euler Dens. \000";
xy(x[.,15],x[.,16:18]);
nextwind;
title ("Condit. Distribution Function given x]0[=1 (\204\68 \201= 0.5)");
_plegstr="True DF \000"\
"Proposed DF \000"\
"Euler DF \000";
xy(x[.,15],x[.,19:21]);
endwind;
```

Comandos uteis:

```
**************************************************
min=0;max=20;step=2;minordiv=4;xtics(min,max,step,minordiv);
min=85;max=135;step=5;minordiv=4;ytics(min,max,step,minordiv);
*****************************************
_plwidth=10|1|2;
_pltype=6|1|1;
```

INSERIR TEXTO

```
_pmsgstr ="x]1\000 x]0\000 x‘]0";
_pmsgctl = { 2 0.1 .2 0 1 1 0,
```

2 -0.9 .2 0 1 1 0,

2 -1.9 .2 0 1 1 0};

INSERIR CIRCULO NA PRIMEIRA SERIE (de 4)

_pltype=1|6|6|6;

_plctrl=1|0|0|0;

_pstype=8;

PASSAR NUM CICLO FOR TITULO STRING PARA COMANDO TITLE

graphset;

_ptek = "temp.tkf";

No fim:

tkf2eps("temp.tkf",nome[1,serie]$+".eps");

22.2 Gráficos plot

22.2.1 Hip 1

```
struct plotControl myPlot; // Declare the structure
    myPlot = plotGetDefaults("xy"); // Initialize the structure
    plotlayout(1,2,1);
    plotSetXLabel(&myPlot, "n", "verdana",15, "black");
    plotSetYLabel(&myPlot, "MISE", "verdana",15, "black");
    plotSetTicLabelFont(&myPlot, "verdana", 15);
    plotxy(myplot,x,y);
    plotlayout(1,2,2);
    plotxy(myplot,t,y);
    plotClearLayout();
```

22.2.2 Hip 2

```
struct plotControl myPlot; // Declare the structure
    myPlot = plotGetDefaults("xy"); // Initialize the structure
    plotSetLegend(&myplot,"off");
    plotlayout(2,2,1);
```

```
plotSetXLabel(&myPlot,"x");

plotSetYLabel(&myPlot,);

plotSetTitle( &myPlot, "Conditional Mean");

plotxy(myplot,x,mean(par1,x)~x);

plotlayout(2,2,2);

plotSetTitle( &myPlot, "Cond. Mean Increment");

plotxy(myplot,x,(-x+mean(par1,x))~zeros(rows(x),1));

plotCustomLayout(0.1,0,.8,.5);

plotSetTitle( &myPlot, "Simulated Paths - Random Walk Vs. ESTAR");

plotSetXLabel(&myPlot,"time");

plotSetLegend(&myplot,"ESTAR"$|"R.Walk");

plotxy(myplot,t,y~z);

plotClearLayout();
```