Modelo estimado utilizando R

Gabriel Petrini da Silveira

2019

Carregando pacotes

```r
library(zoo)
library(xts)
library(tsDyn)
library(urca)
library(vars)
library(dplyr)
library(stargazer)
library(lmtest)
```

Carregando dados

```r
df <- read.csv(
  "/dados/Dissertacao/Modelo/SeriesTemporais/Dados_completos.csv",
  encoding="UTF-8",
  stringsAsFactors=FALSE
  )
df <- ts(data = df, start = c(1987,01), frequency = 4)
#df <- as.xts(df)
df <- zoo::na.locf0(df)
```

Quebra estrutural

Taxa de crescimento do investimento residencial

```r
result = breakpoints(gZ~1, data=df)
result$breakpoints %>% unique() %>% na.omit() %>% c() -> breaks

for(i in breaks){
  print(paste0("Testando para i = ", index(df)[i]))
  strucchange::sctest(gZ~1, data=df, point=i, type="Chow") %>% print()
}
```

```
## [1] "Testando para i = 1991.5"
##
##  Chow test
##
## data:  gZ ~ 1
## F = 5.1147, p-value = 0.0254
##
## [1] "Testando para i = 2005.75"
##
##  Chow test
##
## data:  gZ ~ 1
## F = 7.286, p-value = 0.007881
##
## [1] "Testando para i = 2010.5"
##
##  Chow test
##
## data:  gZ ~ 1
## F = 6.1013, p-value = 0.01481
```

Taxa Própria

```
result = breakpoints(Taxa.Própria~1, data=df)
result$breakpoints %>% unique() %>% na.omit() %>% c() -> breaks

for(i in breaks){
  print(paste0("Testando para i = ", index(df)[i]))
  strucchange::sctest(Taxa.Própria~1, data=df, point=i, type="Chow") %>% print()
}
```

```
## [1] "Testando para i = 1991.75"
##
##  Chow test
##
## data:  Taxa.Própria ~ 1
## F = 63.453, p-value = 7.487e-13
##
## [1] "Testando para i = 1996.5"
##
##  Chow test
##
## data:  Taxa.Própria ~ 1
## F = 107.47, p-value < 2.2e-16
##
## [1] "Testando para i = 2001.25"
##
##  Chow test
##
## data:  Taxa.Própria ~ 1
## F = 78.378, p-value = 5.662e-15
##
## [1] "Testando para i = 2006"
##
##  Chow test
##
## data:  Taxa.Própria ~ 1
## F = 20.68, p-value = 1.236e-05
##
## [1] "Testando para i = 2011"
##
##  Chow test
##
## data:  Taxa.Própria ~ 1
## F = 78.969, p-value = 4.663e-15
```

Taxa de juros

```
result = breakpoints(Taxa.de.juros~1, data=df)
result$breakpoints %>% unique() %>% na.omit() %>% c() -> breaks

for(i in breaks){
  print(paste0("Testando para i = ", index(df)[i]))
  strucchange::sctest(Taxa.de.juros~1, data=df, point=i, type="Chow") %>% print()
}
```

```
## [1] "Testando para i = 1991.5"
##
##  Chow test
##
## data:  Taxa.de.juros ~ 1
## F = 124.35, p-value < 2.2e-16
##
## [1] "Testando para i = 1997"
##
##  Chow test
##
## data:  Taxa.de.juros ~ 1
## F = 199.25, p-value < 2.2e-16
##
## [1] "Testando para i = 2002"
##
##  Chow test
##
## data:  Taxa.de.juros ~ 1
## F = 301.18, p-value < 2.2e-16
##
## [1] "Testando para i = 2009.75"
##
##  Chow test
##
## data:  Taxa.de.juros ~ 1
## F = 172.97, p-value < 2.2e-16
```

Inflação

```r
result = breakpoints(Inflação~1, data=df)
result$breakpoints %>% unique() %>% na.omit() %>% c() -> breaks

for(i in breaks){
  print(paste0("Testando para i = ", index(df)[i]))
  strucchange::sctest(Inflação~1, data=df, point=i, type="Chow") %>% print()
}
```

```
## [1] "Testando para i = 1997.5"
##
##  Chow test
##
## data:  Inflação ~ 1
## F = 1.5508, p-value = 0.2153
##
## [1] "Testando para i = 2005.75"
##
##  Chow test
##
## data:  Inflação ~ 1
## F = 23.49, p-value = 3.569e-06
##
## [1] "Testando para i = 2011.5"
##
##  Chow test
##
## data:  Inflação ~ 1
## F = 4.4981, p-value = 0.03586
```

Teste de Johansen

gZ e Taxa Própria

```r
vars::VARselect(
  y = df[,c("gZ", "Taxa.Própria")] %>%  na.omit(),
  type="both"
)$selection[1] %>% as.numeric() -> p
urca::ca.jo(
  x = df[,c("gZ", "Taxa.Própria")],
  ecdet = "const",
  #ecdet = "trend",
  K = p-1,
  spec = "longrun",
  type = "trace"
) %>% summary()
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
```

```
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1]  1.434138e-01  2.268023e-02 -2.445838e-18
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  2.91  7.52  9.24 12.97
## r = 0  | 22.57 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                       gZ.l4 Taxa.Própria.l4   constant
## gZ.l4             1.0000000       1.0000000  1.0000000
## Taxa.Própria.l4   0.1690159       2.2308504  1.1763123
## constant         -0.0233424      -0.1392448 -0.1647254
##
## Weights W:
## (This is the loading matrix)
##
##                       gZ.l4 Taxa.Própria.l4      constant
## gZ.d            -0.35607981     0.009256991 -1.159899e-18
## Taxa.Própria.d   0.03810164    -0.022321492 -1.185584e-17
```

gZ, Inflação e Taxa de juros

```r
vars::VARselect(
  y = df[,c("gZ", "Inflação", "Taxa.de.juros")] %>%  na.omit(),
  type="both"
)$selection[1] %>% as.numeric() -> p

urca::ca.jo(
  x = df[,c("gZ", "Inflação", "Taxa.de.juros")],
  ecdet = "const",
  #ecdet = "trend",
  K = p-1,
  spec = "longrun",
  type = "trace"
) %>% summary()
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1] 2.178996e-01 6.630650e-02 4.981228e-02 2.586886e-17
```

```
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 2 |  6.44  7.52  9.24 12.97
## r <= 1 | 15.08 17.85 19.96 24.60
## r = 0  | 46.05 32.00 34.91 41.07
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                        gZ.l4  Inflação.l4 Taxa.de.juros.l4   constant
## gZ.l4             1.00000000    1.0000000        1.0000000  1.0000000
## Inflação.l4      -1.39478996 -100.6726782      -12.6616050 -1.0202930
## Taxa.de.juros.l4  0.06699596   18.8950268       -8.3629159  3.9302585
## constant         -0.00438352    0.1929623        0.3533404 -0.3079164
##
## Weights W:
## (This is the loading matrix)
##
##                        gZ.l4   Inflação.l4 Taxa.de.juros.l4      constant
## gZ.d            -0.376111685  0.0012306816      0.009272629 -4.432163e-18
## Inflação.d       0.072690737  0.0004891931      0.003259871 -1.669739e-17
## Taxa.de.juros.d  0.002131398 -0.0006204737      0.001203076  1.002505e-17
```

gZ, Inflação (Taxa de juros exog)

```
df <- df[,c("gZ", "Inflação", "Taxa.de.juros")] %>% na.omit()

vars::VARselect(
  y = df[,c("gZ", "Inflação")],
  type="both",
  exogen = df[,c("Taxa.de.juros")]
)$selection[1] %>% as.numeric() -> p
urca::ca.jo(
  x = df[,c("gZ", "Inflação")],
  ecdet = "const",
  #ecdet = "trend",
  K = p-1,
  spec = "longrun",
  type = "trace"
) %>% summary()
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1] 2.056307e-01 6.055571e-02 5.551115e-17
```

```
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  7.87  7.52  9.24 12.97
## r = 0  | 36.88 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                   gZ.l4  Inflação.l4  constant
## gZ.l4        1.00000000   1.00000000  1.000000
## Inflação.l4 -1.28768282 -12.74465206 -2.106929
## constant    -0.00429196   0.09533979  0.252638
##
## Weights W:
## (This is the loading matrix)
##
##                  gZ.l4 Inflação.l4      constant
## gZ.d       -0.41733552  0.03276165 -7.607639e-18
## Inflação.d  0.06182504  0.01037346 -9.496784e-19
```