



Modelo Semiestructural de Pequeno Porte

Estimação conjunta das 4 equações

Vítor Wilher, MSc in Economics

17 de novembro, 2019

Abstract

Concluimos nessa subsecção a estimação de um modelo semiestructural de pequeno porte para a economia brasileira com a estimação conjunta das quatro equações vistas nas aulas anteriores.

Contents

1	Introdução	2
2	Pacotes	3
3	Importação e tratamento das variáveis	3
3.1	Importar expectativas de inflação	3
3.2	Importar e tratar inflação total e de preços livres	4
3.3	Importar e tratar IC-Br	5
3.4	Importar o hiato do produto	5
3.5	Importar o resultado estrutural do setor público	6
3.6	Construir o juro real	6
3.7	Importar a taxa Selic	6
3.8	Criar desvio entre inflação esperada e meta de inflação	6
3.9	Importar o swap pré-DI 360 dias	7
3.10	Criar selic esperada 4 trimestres à frente	7
3.11	Criar o prêmio entre o swap e a selic esperada	8
3.12	Importar o risco país	9
3.13	Reunir os dados	9
4	Escrever as equações	10
5	Estimar o modelo	11

1 Introdução

Nas aulas anteriores, vimos a estimação das quatro equações que compõem o modelo semiestrutural básico e representativo da economia brasileira. Para começar, estimamos uma Curva de Phillips como descrito abaixo:

$$\pi_t^{livres} = \sum_{i>0} \beta_{1i} E_t \pi_{t+i}^{total} + \sum_{j>0} \beta_{2j} \pi_{t-j}^{total} + \sum_{k \geq 0} \beta_{3k} \pi_{t-k}^{importada} + \sum_{l>0} \beta_{4l} h_{iato_{t-l}} + \sum_{m=1}^4 D_m + \epsilon_t \quad (1)$$

Basicamente, a inflação dos preços livres é uma função linear da **inflação passada**, das **expectativas de inflação**, do **hiato do produto** e da **inflação importada**. A seguir, estimamos uma Curva IS que representa o lado da demanda da economia e pode ser descrita como segue:

$$h_t = \gamma_0 + \sum_{i>0} \gamma_{1i} h_{t-1} + \sum_{j>0} \gamma_{2j} r_{t-j} + \sum_{k>0} \gamma_{3k} \Delta sup_{t-k} + u_t \quad (2)$$

Basicamente, a Curva IS estimada irá descrever a dinâmica do hiato do produto com base em **suas próprias defasagens**, da **taxa de juros real ex-ante** e da **variação do superávit primário**. Na sequência, estimamos uma regra de Taylor que descreve o comportamento da autoridade monetária e pode ser descrita como segue:

$$i_t = \alpha_0 + \alpha_1 i_{t-1} + \alpha_2 i_{t-2} + \alpha_3 (\pi_t^e - \pi^M) + \alpha_4 h_{t-1} + v_t \quad (3)$$

Basicamente, a regra de Taylor a ser estimada relaciona a taxa básica de juros às suas próprias defasagens - de modo a incorporar a suavização da taxa de juros ao longo do tempo e, econometricamente, prevenir autocorrelação nos resíduos -, a diferença entre a inflação projetada e a meta de inflação e uma medida de hiato do produto. Por fim, modelamos o prêmio do Swap pré-DI de 360 dias frente a taxa básica de juros, a Selic, por meio da seguinte equação:

$$premio_t = \delta_0 + \sum_{i>0} \delta_{1i} premio_{t-i} + \delta_2 rispa_t + x_t \quad (4)$$

Onde $premio_t$ é o diferencial entre a taxa swap pré-DI de 360 dias e a expectativa para a taxa Selic para o período do contrato do *swap*; $rispa_t$ é uma variável representativa do prêmio de risco do país (Embi ou CDS 5 anos, por exemplo); e x_t é um termo de erro.¹

Já aqui, vamos mostrar a estimação conjunta dessas quatro equações.

¹A ideia é que na construção de cenários, a trajetória do *swap* depende fortemente da trajetória para a Selic.

2 Pacotes

Abaixo, carregamos os pacotes necessários para o exercício.²

```
library(devtools)
library(tidyverse)
library(lubridate)
library(readxl)
library(restriktor)
library(RcppRoll)
library(broom)
library(scales)
library(rbcb)
library(xts)
library(dynlm)
library(systemfit)
```

3 Importação e tratamento das variáveis

Nessa parte do exercício, nós consolidamos a importação de todas as variáveis que utilizaremos para a estimação das equações.

3.1 Importar expectativas de inflação

O modelo representado por 1 será estimado em bases trimestrais. Desse modo, vamos precisar fazer algum tratamento dos dados. Para começar, importamos as expectativas de inflação em bases diárias e depois obtemos as expectativas diárias para os trimestres seguintes e, por fim, retiramos a média trimestral dessas expectativas.

```
## Importar e Tratar Expectativas de Inflação
exp_ipca <- get_monthly_market_expectations("IPCA",
                                             end_date = "2018-09-31")

diff_quarter <- function(end_date, start_date){

  year_to_quarters <- (floor(end_date)*4 + (end_date %% 1)*10) -
                      (floor(start_date)*4 + (start_date %% 1)*10)

  return(year_to_quarters)
}

exp_ipca_aux <- exp_ipca %>%
```

²O pacote `rbcb` não está mais disponível no CRAN, sendo necessário instalá-lo via `github` através do pacote `devtools`. Maiores detalhes, ver aqui.

```

dplyr::select(date, reference_month, median) %>%

dplyr::mutate(reference_month = lubridate::ymd(paste(reference_month,
                                                    "01",
                                                    sep = "-")) %>%

dplyr::mutate(date_year = lubridate::year(date),
              date_month = lubridate::month(date)) %>%

dplyr::group_by(date_year, date_month, reference_month) %>%

dplyr::summarise(median_month = mean(median)) %>%

dplyr::ungroup() %>%

dplyr::mutate(date = lubridate::make_date(year = date_year,
                                          month = date_month)) %>%

dplyr::select(date, reference_month, median_month) %>%

dplyr::filter(date > "2001-12-01") %>%

dplyr::mutate(ref_quarter = lubridate::quarter(reference_month,
                                              with_year = T)) %>%

dplyr::group_by(date, ref_quarter) %>%

dplyr::summarise(median_quarter = last(((cumprod(1+(median_month/100)))-1)*100)) %>%

dplyr::mutate(date_quarter = lubridate::quarter(date, with_year = T)) %>%

dplyr::group_by(date_quarter, ref_quarter, add = F) %>%

dplyr::summarise(median_quarter = mean(median_quarter)) %>%

dplyr::filter(ref_quarter > date_quarter) %>%

dplyr::mutate(diff = round(diff_quarter(ref_quarter, date_quarter),1) %>%

dplyr::select(-ref_quarter) %>%

tidyr::spread(key = diff, value = median_quarter)

colnames(exp_ipca_aux)[-1] <- paste("EInf_t+",
                                   colnames(exp_ipca_aux)[-1],
                                   sep = "")

```

3.2 Importar e tratar inflação total e de preços livres

```

acum_quarter <- function(x){

```

```

x_fac <- 1+(x/100)

x_cum <- RcppRoll::roll_prodr(x_fac, n = 3)

x_qr <- last((x_cum-1)*100)

return(x_qr)
}

dados_ipca <- read_csv2('ipca.csv',
  col_types =
    list(col_date(format='%d/%m/%Y'),
         col_double(),
         col_double())) %>%

dplyr::mutate(date_quarter = lubridate::quarter(date, with_year = TRUE)) %>%

dplyr::group_by(date_quarter) %>%

dplyr::summarise_at(vars(ipca_total, ipca_livres), funs(acum_quarter))

```

3.3 Importar e tratar IC-Br

```

acum_ic <- function(x){

  x_diff <- log(x/first(x))*100

  x_acum <- last(x_diff)

  return(x_acum)

}

dados_ic <- read_csv2('ic.csv',
  col_types =
    list(col_date(format='%d/%m/%Y'),
         col_double())) %>%

dplyr::mutate(date_quarter = lubridate::quarter(date, with_year = TRUE)) %>%

dplyr::group_by(date_quarter) %>%

dplyr::summarise_at(vars(ic_br), funs(acum_ic))

```

3.4 Importar o hiato do produto

O hiato do produto é divulgado pelo IPEA em frequência trimestral e, logo, não necessita de transformações.

```
hiato <- read_excel("hiato.xlsx") %>%

  dplyr::mutate(date_quarter = as.numeric(gsub(" T", ".", date_qr)),
               hiato = hiato*100) %>%

  dplyr::select(date_quarter, hiato)
```

3.5 Importar o resultado estrutural do setor público

O resultado fiscal estrutural é divulgado em bases trimestrais pela Secretaria de Política Econômica (SPE) do Ministério da Fazenda. Nós importamos como abaixo.

```
sup <- read_excel("sup.xlsx") %>%

  dplyr::mutate(date_quarter = as.numeric(gsub(" T", ".", date_qr)),
               sup = sup*100) %>%

  dplyr::select(date_quarter, sup)

dsup = diff(sup$sup)
dsup = tibble(date_quarter=sup$date_quarter[-1], dsup=dsup)
```

3.6 Construir o juro real

```
juro = read_excel('juro.xlsx',
                  col_types = c('date', rep('numeric',2)))
juroreal = (((1+(juro$swap/100))/(1+(juro$expinf/100)))-1)*100
juroreal = xts(juroreal, order.by=juro$date)
juroreal = apply.quarterly(juroreal, FUN=mean)
juroreal = tibble(date_quarter=sup$date_quarter, juroreal=juroreal)
```

3.7 Importar a taxa Selic

Abaixo, começamos importando a taxa básica de juros que é calibrada pelo Banco Central.

```
selic <- read_excel("selic.xlsx")
selic = xts(selic$selic, order.by = selic$date)
selic = apply.quarterly(selic, FUN=mean)
selic = window(selic, start='2002-03-01', end='2017-12-01')
selic = tibble(date_quarter=sup$date_quarter, selic=selic)
```

3.8 Criar desvio entre inflação esperada e meta de inflação

Assim como fizemos ao construir e estimar a Curva de Phillips, nós vamos importar e tratar as expectativas de inflação.

```
## Importar e Tratar Expectativas de Inflação
exp_ipca <- get_twelve_months_inflation_expectations("IPCA",
                                                    end_date = "2017-12-31")

exp_ipca_d = xts(exp_ipca$mean, order.by = exp_ipca$date)
exp_ipca_q = apply.quarterly(exp_ipca_d, FUN=mean)

meta_ahead <- read_excel("meta_ahead.xlsx")

meta_ahead = xts(meta_ahead$meta_ahead, order.by = meta_ahead$date)
meta_ahead = apply.quarterly(meta_ahead, FUN=mean)

exp_ipca_q = ts(exp_ipca_q, start=c(2001,4), freq=4)
meta_ahead = ts(meta_ahead, start=c(2000, 1), freq=4)

desvio = exp_ipca_q - meta_ahead
desvio = window(desvio, start=c(2002,01))
desvio = tibble(date_quarter=sup$date_quarter, desvio=desvio)
```

3.9 Importar o swap pré-DI 360 dias

Abaixo, começamos importando a taxa básica de juros que é calibrada pelo Banco Central.

```
swap <- read_excel("swap.xlsx")
swap = xts(swap$swap, order.by = swap$date)
swap = apply.quarterly(swap, FUN=mean)
```

3.10 Criar selic esperada 4 trimestres à frente

O maior trabalho para estimar a nossa equação é criar a selic esperada 4 trimestres à frente. O código abaixo implementa.

```
## Importar e Tratar Expectativas para Selic
exp_selic <- get_monthly_market_expectations("Meta para taxa over-selic", end_date = "2018-09-31")

diff_quarter <- function(end_date, start_date){

  year_to_quarters <- (floor(end_date)*4 + (end_date %% 1)*10) -
                      (floor(start_date)*4 + (start_date %% 1)*10)

  return(year_to_quarters)
}

exp_selic_aux <- exp_selic %>%

  dplyr::select(date, reference_month, median) %>%

  dplyr::mutate(reference_month = lubridate::ymd(paste(reference_month,
```

```

                                "01",
                                sep = "-")))) %>%

dplyr::mutate(date_year = lubridate::year(date),
              date_month = lubridate::month(date)) %>%

dplyr::group_by(date_year, date_month, reference_month) %>%

dplyr::summarise(median_month = mean(median)) %>%

dplyr::ungroup() %>%

dplyr::mutate(date = lubridate::make_date(year = date_year,
                                          month = date_month)) %>%

dplyr::select(date, reference_month, median_month) %>%

dplyr::filter(date > "2001-12-01") %>%

dplyr::mutate(ref_quarter = lubridate::quarter(reference_month,
                                              with_year = T)) %>%

dplyr::group_by(date, ref_quarter) %>%

dplyr::summarise(median_quarter = mean(median_month)) %>%

dplyr::mutate(date_quarter = lubridate::quarter(date, with_year = T)) %>%

dplyr::group_by(date_quarter, ref_quarter, add = F) %>%

dplyr::summarise(median_quarter = mean(median_quarter)) %>%

dplyr::filter(ref_quarter > date_quarter) %>%

dplyr::mutate(diff = round(diff_quarter(ref_quarter, date_quarter),1) %>%

dplyr::select(-ref_quarter) %>%

tidyr::spread(key = diff, value = median_quarter)

colnames(exp_selic_aux)[-1] <- paste("ESelic_t+",
                                   colnames(exp_selic_aux)[-1],
                                   sep = "")

```

3.11 Criar o prêmio entre o swap e a selic esperada

```

swap = ts(swap, start=c(1999,3), freq=4)
selic = ts(exp_selic_aux$`ESelic_t+4`, start=c(2002,01), freq=4)
data = ts.intersect(swap, selic)
premio = data[,1]-data[,2]
premio = window(premio, end=c(2017,4))

```



```
premio = tibble(date_quarter=sup$date_quarter, premio=premio)
```

3.12 Importar o risco país

Agora, importamos o EMBI, que irá medir o risco país.

```
embi <- read_excel("embi.xlsx")
embi = xts(embi$embi, order.by = embi$date)
embi = embi[complete.cases(embi)]
embi = apply.quarterly(embi, FUN=mean)
risco_pais = ts(embi, start=c(1994,2), freq=4)
risco_pais = window(risco_pais, start=c(2002,01), end=c(2017,04))
risco_pais = tibble(date_quarter=sup$date_quarter,
                    risco_pais=risco_pais)
```

3.13 Reunir os dados

```
dados_reg <- dplyr::inner_join(dados_ipca, dados_ic) %>%

  dplyr::inner_join(hiato) %>%

  dplyr::inner_join(exp_ipca_aux) %>%

  dplyr::inner_join(juroreal) %>%

  dplyr::inner_join(dsup) %>%

  dplyr::inner_join(selic) %>%

  dplyr::inner_join(desvio) %>%

  dplyr::inner_join(premio) %>%

  dplyr::inner_join(risco_pais) %>%

  dplyr::mutate(quarter = sub('.*\\.', '', date_quarter)) %>%

  dplyr::filter(date_quarter >= 2002.1) %>%

  dplyr::mutate(ipca_l1 = lag(ipca_total, 1),
                ipca_l2 = lag(ipca_total, 2),
                ipca_l3 = lag(ipca_total, 3),
                hiato_l1 = lag(hiato,1),
                hiato_l2 = lag(hiato,2),
                hiato_l3 = lag(hiato, 3),
                ic_l1 = lag(ic_br,1),
                ic_l2 = lag(ic_br,2),
                Einf_1 = `EInf_t+1`,
                juroreal_l1 = lag(juroreal,1),
                juroreal_l2 = lag(juroreal,2),
```

```

selic_l1 = lag(selic,1),
selic_l2 = lag(selic,2),
selic_l3 = lag(selic,3),
premio_l1 = lag(premio,1),
premio_l2 = lag(premio,2),
risco_pais_l1 = lag(risco_pais,1))

```

4 Escrever as equações

Uma vez que todas as variáveis foram importadas, podemos agora escrever as equações.

```

cp = ipca_livres ~ ipca_l1 + ipca_l2 + Einf_1 + hiato_l3 + ic_l1 +
  quarter

is = hiato ~ hiato_l1 + juroreal_l1 + dsup

taylor = selic ~ selic_l1 + selic_l2 + desvio + hiato_l1

premio_swap = premio ~ premio_l1 + risco_pais

inst = ~ipca_l1 + ipca_l2 + ipca_l3 + Einf_1 + hiato_l2 + ic_l2 +
  quarter + hiato_l1 + juroreal_l2 + dsup + selic_l1 + selic_l2 +
  selic_l3 + desvio + premio_l2 + risco_pais_l1

system = list(cp=cp, is=is, taylor=taylor, premioswap=premio_swap)

```

5 Estimar o modelo

```
# OLS estimation
```

```
modelo_ols = systemfit(system, data=dados_reg)
round(coef(summary(modelo_ols)), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
## cp_(Intercept)	1.060	0.295	3.598	0.001
## cp_ipca_l1	0.347	0.102	3.395	0.001
## cp_ipca_l2	-0.185	0.106	-1.743	0.087
## cp_Einf_1	0.451	0.310	1.457	0.151
## cp_hiato_l3	0.083	0.031	2.678	0.010
## cp_ic_l1	0.037	0.014	2.685	0.010
## cp_quarter2	-0.621	0.210	-2.964	0.005
## cp_quarter3	-0.848	0.241	-3.518	0.001
## cp_quarter4	-0.279	0.261	-1.070	0.290
## is_(Intercept)	-0.094	0.306	-0.308	0.760
## is_hiato_l1	0.873	0.058	15.112	0.000
## is_juroreal_l1	-0.017	0.036	-0.481	0.632
## is_dsup	-1.343	0.427	-3.146	0.003
## taylor_(Intercept)	0.613	0.353	1.735	0.088
## taylor_selic_l1	1.386	0.090	15.341	0.000
## taylor_selic_l2	-0.461	0.091	-5.078	0.000
## taylor_desvio	0.436	0.114	3.812	0.000
## taylor_hiato_l1	0.101	0.053	1.910	0.061
## premioswap_(Intercept)	-1.061	0.115	-9.208	0.000
## premioswap_premio_l1	0.279	0.053	5.286	0.000
## premioswap_risco_pais	0.005	0.000	13.240	0.000

```
# TSLS estimation
```

```
modelo_tsls = systemfit(system, method='2SLS', inst=inst,
                        data=dados_reg)
round(coef(summary(modelo_tsls)), 3)
```

	Estimate	Std. Error	t value	Pr(> t)
## cp_(Intercept)	0.984	0.315	3.122	0.003
## cp_ipca_l1	0.341	0.107	3.182	0.002
## cp_ipca_l2	-0.257	0.126	-2.035	0.047
## cp_Einf_1	0.666	0.369	1.806	0.077
## cp_hiato_l3	0.087	0.035	2.469	0.017
## cp_ic_l1	0.006	0.029	0.208	0.836
## cp_quarter2	-0.630	0.220	-2.868	0.006
## cp_quarter3	-0.938	0.263	-3.565	0.001
## cp_quarter4	-0.365	0.282	-1.295	0.201
## is_(Intercept)	0.199	0.340	0.585	0.561
## is_hiato_l1	0.861	0.058	14.870	0.000
## is_juroreal_l1	-0.061	0.042	-1.453	0.152
## is_dsup	-1.208	0.431	-2.803	0.007
## taylor_(Intercept)	0.842	0.352	2.390	0.020
## taylor_selic_l1	1.410	0.090	15.748	0.000
## taylor_selic_l2	-0.505	0.090	-5.594	0.000
## taylor_desvio	0.397	0.113	3.507	0.001
## taylor_hiato_l1	0.078	0.052	1.483	0.144

## premioswap_(Intercept)	-0.427	0.253	-1.689	0.097
## premioswap_premio_l1	0.443	0.098	4.517	0.000
## premioswap_risco_pais	0.003	0.001	2.449	0.017