

Passo a passo de como estimar elasticidades de grupos de receita usando o pacote ElastH

Caio Figueiredo¹

¹Secretaria de Política Econômica do Ministério da Fazenda

30 de maio de 2017

Sumário

Introdução

O pacote ElastH foi desenvolvido com a intenção de disponibilizar método de cálculo de elasticidades de grupos de receitas conforme descrito na metodologia de cálculo do Resultado Fiscal Estrutural da Secretaria de Política Econômica do Ministério da Fazenda - SPE/MF.

O pacote permite a estimação de diversos modelos de componentes não observados de maneira simples, sem exigir conhecimentos mais profundos do funcionamento do Filtro de Kalman. Adicionalmente são disponibilizados testes e critérios que permitam a avaliação do modelo estimado.

Este tutorial parte do pressuposto de que o leitor já tem conhecimento básico sobre o funcionamento do software estatístico R, que consiga importar e exportar dados bem como criar variáveis. Também é recomendável que se saiba trabalhar com séries de tempo.

Estrutura do Modelo de Componentes não Observados

Apesar de não ser necessário ter conhecimento profundo sobre os modelos de componentes não observados, ou sobre o filtro de Kalman, é interessante uma breve introdução do modelo que é utilizado pela metodologia da SPE:¹

$$y_t = \mu_t + \beta_t \cdot X_t + \gamma_t + \varepsilon_t$$

$$\mu_t = \mu_{t-1} + \nu_{t-1} + \xi_t$$

$$\nu_t = \nu_{t-1} + \zeta_t$$

$$\gamma_t = \gamma_{1,t} + \gamma_2,$$

$$\gamma_{1,t} = -\gamma_{1,t-2} + \omega_{1,t}$$

$$\gamma_{2,t} = -\gamma_{2,t-1} + \omega_{2,t}$$

¹Modelos de componentes não observados podem assumir formas mais abrangentes, que poderiam ser aplicados utilizando as funções deste pacote. Porém, o tutorial se atém ao modelo utilizado pela metodologia de SPE.

$$\beta_t = \beta_{t-1} + \eta_t$$

onde y_t é a série de tempo a ser decomposta e X_t um grupo de variáveis explicativas. No caso de interesse, y_t é uma das séries de grupos de receitas definidas na metodologia e X_t é o Hiato do Produto ou o Hiato do Petróleo². Já μ_t , ν_t , γ_t e β_t são os componentes não observados estimados pelo Filtro de Kalman, respectivamente, nível, inclinação, sazonalidade e coeficiente(s).

Os choques destas equações seguem as seguintes distribuições:

$$\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

$$\xi_t \sim \mathcal{N}(0, \sigma_\xi^2)$$

$$\zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

$$\omega_{1,t} \sim \mathcal{N}(0, 2\sigma_\omega^2)$$

$$\omega_{2,t} \sim \mathcal{N}(0, \sigma_\omega^2)$$

$$\eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$$

Algumas hipóteses podem ser feitas sobre o comportamento dos componentes, que podem, por exemplo, ser estocásticos (padrão) ou fixos³. Um componente estocástico segue a estrutura de um passeio aleatório, e tem a variância de seu choque estimada pelo pacote. Um componente fixo apresentará o mesmo valor em todos os períodos, ou seja, tem a variância de seu choque determinada e igual a 0.

Este pacote, então, se propõe a estimar os componentes sob diferentes hipóteses e prover as ferramentas de testes que permitam comparar os modelos, subsidiando a escolha da melhor forma funcional sob a qual as elasticidades devem ser estimadas.

Instalação e Carregamento

Para instalar o pacote de estimação de elasticidades da SPE/MF, é necessário estar conectado à internet e digitar o seguinte comando no console do R:

```
> install.packages("ElastH")
```

²Como destacado na metodologia de cálculo do Resultado Fiscal Estrutural da SPE, apesar de utilizarmos durante o tutorial os nomes Hiato de Produto e Hiato de Petróleo, isto não é estritamente correto. Na verdade o que é utilizado é $\ln(Y/Y^*)$ e $\ln(Pet/Pet^*)$ onde Y , Y^* , Pet , Pet^* são respectivamente, o PIB efetivo, o PIB tendencial, o preço de petróleo efetivo e o preço de petróleo tendencial.

³Outra possibilidade é que os componentes sejam ignorados, ou seja, excluídos do modelo. Nesta metodologia, porém, apenas o componente sazonal é ignorado, e somente com as receitas de petróleo (Royalties e Participação Especial).

Alternativamente é possível baixar o pacote no site da SPE⁴, e utilizar o seguinte comando para instalar:

```
> install.packages("Caminho/ElastH_0.3.1.tar.gz", repos=NULL, type="source")
```

Por favor note que para separar os diretórios é necessário utiliza a barra padrão `"/`, ainda que sistemas windows tradicionalmente exibam seu endereços utilizando a barra invertida `"\"`, pois a barra invertida é um caractere especial no software estatístico R. Outra alternativa seria utilizar barras invertidas duplas `"\"`.

Para carregar o pacote no ambiente padrão do R é necessário chamar o seguinte comando no início de toda sessão.

```
> library(ElastH)
```

Estrutura dos dados de entrada

O primeiro passo para estimar as elasticidades é importar as séries de receitas para o R. Para fins deste tutorial, será usada uma série fictícia de exemplo disponibilizada neste pacote. Para ter acesso a ela, basta digitar o seguinte comando no console do R:

```
> data(Exemplo)
```

Assim, a variável `Exemplo` estará disponível. Esta variável consiste em uma lista fictícia de dados de interesse, incluindo séries de tempo similares (mas não iguais) aos grupos de receitas de interesse⁵. Para visualizá-las, digite:

```
> Exemplo$y
```

	Qtr1	Qtr2	Qtr3	Qtr4
2005	8.468764	8.456690	8.599470	8.594502
2006	8.589788	8.697490	8.809763	8.834902
2007	8.799565	8.886277	9.074360	9.134457
2008	9.199340	9.299899	9.485294	9.471351
2009	8.934018	8.804211	9.028303	9.219753
2010	9.271527	9.317269	9.441083	9.486182
2011	9.447879	9.526106	9.651847	9.654834
2012	9.482889	9.560783	9.490462	9.511258
2013	9.382203	9.497319	9.541160	9.354080
2014	9.260405	9.183220	9.229239	9.206761

⁴ <http://www.spe.fazenda.gov.br/assuntos/politica-fiscal-e-tributaria/resultado-fiscal-estrutural>

⁵É importante destacar que para a estimação das elasticidades é utilizado a série das receitas em log.

```
> Exemplo$HpiB
```

```
      Qtr1      Qtr2      Qtr3      Qtr4
1997 -0.0021561715  0.0072766419 -0.0097881772 -0.0064660835
1998 -0.0161704444 -0.0149841336 -0.0034022772 -0.0046802549
1999  0.0097203565  0.0080387997  0.0048600295  0.0017400091
2000  0.0535383046  0.0033112412  0.0727705217 -0.0403442535
2001 -0.0133474694 -0.0139414899 -0.0157975711  0.0030958321
2002  0.0011423274  0.0073170240  0.0120591076  0.0134010693
2003 -0.0106558455  0.0118463839  0.0089797693  0.0089602312
2004  0.0065060722  0.0024627787  0.0283609871 -0.0051798943
2005 -0.0021561715  0.0072766419 -0.0097881772 -0.0064660835
2006 -0.0161704444 -0.0149841336 -0.0034022772 -0.0046802549
2007  0.0097203565  0.0080387997  0.0048600295  0.0017400091
2008  0.0535383046  0.0033112412  0.0727705217 -0.0403442535
[ reachedgetOption("max.print") -- omitted 7 rows ]
```

Funções do ElastH

O pacote disponibiliza quatro funções públicas que serão explanadas em suas respectivas seções, a saber:

1. `decompor`
2. `decompor.todos`
3. `calcular.elasticidades`
4. `exportar`

decompor - Estimando um modelo de componentes não observados

Para estimar os componentes não observados, é necessário utilizar a função `decompor`. Em sua forma mais simples, é preciso informar, como argumento, a série de tempo a ser usada como variável a ser decomposta:

```
> modelo <- decompor(Exemplo$y)
```

É importante destacar que, para funcionar corretamente, a variável a ser decomposta seja uma série de tempo, ou seja de classe "ts":

```
> class(Exemplo$y)
```

```
[1] "ts"
```

A função retorna uma longa lista de variáveis, e que agora está salva sob o nome de `modelo`. Uma das variáveis desta lista é a `comp`, que exibe o valor estimado para todos os componentes. Para visualizá-la, digite⁶:

```
> modelo$comp

      Valor  Nivel  Inclinação  Sazon
1  8.468764 8.507514 0.044671440 -0.03875099
2  8.456690 8.495584 0.046973232 -0.03889367
3  8.599470 8.549551 0.048970157 0.04991870
4  8.594502 8.566609 0.052358190 0.02789302
5  8.589788 8.629090 0.055304934 -0.03930228
6  8.697490 8.735747 0.056013184 -0.03825800
7  8.809763 8.760313 0.058092240 0.04945007
8  8.834902 8.806554 0.060687935 0.02834742
9  8.799565 8.839107 0.064510069 -0.03954201
10 8.886277 8.924557 0.067419381 -0.03828045
11 9.074360 9.025070 0.068886090 0.04928918
12 9.134457 9.105930 0.069830868 0.02852751
[ reached getOption("max.print") -- omitted 28 rows ]
```

Entre as diversas outras variáveis retornadas (vide `str(modelo)` para uma lista completa), destacam-se: os testes de independência de resíduos (variável `q`), o teste de homocedasticidade (variável `h`) e teste de normalidade dos resíduos (variável `nt`). Para visualizá-los, utilize:

```
> modelo$q

      Q.valor Q.critico  pvalor lags
X-squared 5.37734 12.59159 0.496404 8

> modelo$h

      H.valor H.critico  pvalor
1 6.976872 2.81793 0.001598265

> modelo$nt

      Normality.test Normality.critico  pvalor
1 2.364778 5.991465 0.3065455
```

Os valores críticos são calculados a um nível de significância de 5%. Para mais informações sobre os resultados retornados por `decomp`, leia a página de ajuda dessa função com: `help(decomp)`.

⁶Os resultados exibidos abaixo podem apresentar pequena variação a depender da versão do software R e da arquitetura (32/64 bits) do computador utilizado

Inclusão de variáveis explicativas

Para incluir variáveis explicativas, basta definir o argumento `X`:

```
> modelo <- decompor(Exemplo$y, X=Exemplo$Hpib)

> modelo$comp

      Valor  Nivel  Inclinação  Sazon  Coef1
1  8.468764 8.511197 0.0450381474 -0.04156619 0.4023497
2  8.456690 8.488695 0.0475332442 -0.03493222 0.4023504
3  8.599470 8.559267 0.0491148171 0.04414068 0.4023521
4  8.594502 8.564744 0.0524267067 0.03235966 0.4023549
5  8.589788 8.637867 0.0549179533 -0.04157288 0.4023571
6  8.697490 8.738441 0.0555988391 -0.03492351 0.4023585
7  8.809763 8.766999 0.0573519593 0.04413324 0.4023618
8  8.834902 8.804418 0.0598954747 0.03236690 0.4023651
9  8.799565 8.837232 0.0635127929 -0.04157853 0.4023684
10 8.886277 8.917962 0.0664474238 -0.03491961 0.4023724
[ reached getOption("max.print") -- omitted 30 rows ]
```

Note que agora temos um componente⁷ a mais: o coeficiente da variável explicativa, que, no caso particular deste pacote, representa a elasticidade de um grupo de receita ante o Hiato do PIB e é aproximadamente igual a 0.40.

Controle de comportamento dos componentes

Para controlar as hipóteses sobre os componentes, utilizamos os argumentos `nivel`, `inclinacao`, `sazon` e `regres`, que controlam, respectivamente, os componentes de: nível, inclinação, sazonalidade e coeficientes. Os argumentos podem ser definidos como "S", para componentes estocásticos (padrão), "F" para componentes fixos⁸. Por exemplo, para fixar a inclinação, mantendo os outros parâmetros estocásticos, digite:

```
> modelo <- decompor(Exemplo$y, X=Exemplo$Hpib, inclinacao="F")

> modelo$comp
```

```
      Valor  Nivel Inclinação  Sazon  Coef1
1  8.468764 8.512409 0.03006796 -0.04276414 0.4088652
```

⁷O termo componente é utilizado neste trabalho de forma abrangente, de modo que tanto a sazonalidade, que na realidade é a combinação de três variáveis de estado, como visto anteriormente, quanto o coeficiente que é apenas parte daquilo que afeta a equação principal são tratados como "componentes".

⁸Além de "N" para componentes ignorados.

```

2  8.456690  8.487756  0.03006796 -0.03404081  0.4088655
3  8.599470  8.558418  0.03006796  0.04505383  0.4088662
4  8.594502  8.565395  0.03006796  0.03175112  0.4088673
5  8.589788  8.639164  0.03006796 -0.04276414  0.4088681
6  8.697490  8.737656  0.03006796 -0.03404081  0.4088686
7  8.809763  8.766101  0.03006796  0.04505383  0.4088699
8  8.834902  8.805064  0.03006796  0.03175112  0.4088711
9  8.799565  8.838355  0.03006796 -0.04276414  0.4088723
10 8.886277  8.917031  0.03006796 -0.03404081  0.4088738
[ reached getOption("max.print") -- omitted 30 rows ]

```

Perceba que, agora, o componente de inclinação apresenta o mesmo valor em todos os períodos. Para consultar informações, vide: `help(decompor)`.

Definição de período de análise

Por fim, é possível restringir o escopo do modelo definindo o período de início e fim da análise. Isso é especialmente útil no caso de interesse, pois algumas receitas têm quebras estruturais importantes em suas elasticidades, sendo interessante, então, resumir o escopo da estimação somente ao período mais recente. Para isso utilize os parâmetros `comeco` e `fim`:⁹

```

> modelo <- decompor(Exemplo$y, X=Exemplo$Hpib, comeco=2007,
+                   fim=c(2014,4))

```

decompor.todos - Testando os componentes

Considerando as várias possíveis hipóteses sobre o comportamento dos componentes, faz-se necessário testar quais destas hipóteses se encaixam melhor no grupo de receita em questão. A fim de facilitar a estimação de diversos modelos e possibilitar a sua comparação, é possível usar a função `decompor.todos`. A função estimará os componentes não observados para oito conjuntos de hipóteses diferentes.¹⁰ Para utilizar esta função, basta digitar o seguinte comando:

```

> lista.dlm <- decompor.todos(Exemplo$y, X=Exemplo$Hpib,
+                             comeco=2005, fim=c(2014,4))

```

Para visualizar a estrutura do resultado digite:

⁹Cabe destacar que os valores da variável `comeco` e `fim` devem ter a seguinte estrutura: `c(Ano,Trimestre)`. Alternativamente é possível definir apenas o `ano`, caso em que será utilizado o primeiro trimestre daquele Ano como parâmetro.

¹⁰As hipóteses são as oito diferentes combinações de componentes fixos ou estocásticos para: nível, inclinação e sazonalidade.


```

> str(lista.dlm,1)

List of 8
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"
 $ :List of 15
  ..- attr(*, "class")= chr "mee"

```

Perceba que a função retorna uma lista com oito modelos estimados, e, para cada um deles, uma elasticidade é estimada e pode ser acessada pela mesma variável `comp`. Para visualizar os componentes estimados de um desses oito modelos, digite:

```

> lista.dlm[[1]]$comp

      Valor  Nivel  Inclinação  Sazon  Coef1
1  8.468764 8.511197 0.0450381474 -0.04156619 0.4023497
2  8.456690 8.488695 0.0475332442 -0.03493222 0.4023504
3  8.599470 8.559267 0.0491148171 0.04414068 0.4023521
4  8.594502 8.564744 0.0524267067 0.03235966 0.4023549
5  8.589788 8.637867 0.0549179533 -0.04157288 0.4023571
6  8.697490 8.738441 0.0555988391 -0.03492351 0.4023585
7  8.809763 8.766999 0.0573519593 0.04413324 0.4023618
8  8.834902 8.804418 0.0598954747 0.03236690 0.4023651
9  8.799565 8.837232 0.0635127929 -0.04157853 0.4023684
10 8.886277 8.917962 0.0664474238 -0.03491961 0.4023724
[ reached getOption("max.print") -- omitted 30 rows ]

```

Note também que cada um dos modelos estimados é de mesma classe, e tem as mesmas variáveis que os modelos estimados pela função `decomp`.

```

> class(lista.dlm[[1]]) == class(modelo)

[1] TRUE

```

Para mais informações, vide `help(decomp.todos)`.

calcular.elasticidades - Estimação em série

Caso se tenha em mãos todas as séries de grupo de receita de interesse para o cálculo do resultado estrutural, é possível usar a função `calcular.elasticidades` para calcular todos os modelos de uma única vez. A função retornará os oito conjuntos de hipóteses diferentes para cada grupo de receita (do mesmo modo que `decompor.todos`). Ainda restará ao analista a escolha de qual conjunto de hipóteses apresenta os melhores testes e resultados.

Para utilizar essa função, digite:

```
> resultado <- calcular.elasticidades(Exemplo$receitas,  
+                                     Hpib=Exemplo$Hpib, Hpet=Exemplo$Hpet, fim=c(2015,4))
```

Não estranhe se demorar para se ter resultados. Devido a grande gama de modelos estimados, este comando pode demorar vários minutos para completar.

Perceba que é necessário informar o argumento `fim`, embora o começo da série seja definido automaticamente¹¹. Também é necessário informar os dois tipos de variáveis explicativas: `Hpib`, a série de Hiato do PIB; e `Hpet`, a série de Hiato de Petróleo, a função usará cada série quando for conveniente. Além destas, é possível definir a variável `tce`, taxa de cambio efetiva, que será utilizada como variável de controle na estimação da elasticidade das receitas de importação. Caso esta variável não seja definida, como no caso acima, o processo controle é ignorado.

É muito importante observar a estrutura da variável `receitas`, que precisa ter um formato específico para este comando funcionar. A mesma deve ser uma tabela de dados, com propriedades de série de tempo, e conter todos os grupos de receitas para os quais se deseja calcular as elasticidades. Além disso, cada coluna da tabela precisa conter o nome que identifica aquele grupo de receita. Ou seja, um dos seguintes: TRT, TFP, TRC, TI, TM, TGC, ROY, PE, TRAN, ICMS, ISS¹². Observe a estrutura da variável usada no exemplo:

```
> Exemplo$receitas
```

	TRT	TFP	TRC	TI	TM	TGC
1997 Q1	8.468764	9.260405	9.382203	9.482889	9.447879	9.271527

¹¹A função sempre assumirá como período inicial aqueles utilizados pela metodologia da SPE/MF.

¹²Respectivamente: Tributos sobre Renda do Trabalho, Tributos sobre Folha de Pagamento, Tributos sobre Renda Corporativa, Tributos Indiretos, Tributos de Importação, Tributos sobre Ganho de Capital, Royalties de Petróleo, Participação Especial de Petróleo, Imposto sobre Circulação de Mercadorias e Serviços, Imposto sobre Serviços. Como definidos na metodologia de SPE/MF.

```

1997 Q2 8.456690 9.183220 9.497319 9.560783 9.526106 9.317269
1997 Q3 8.599470 9.229239 9.541160 9.490462 9.651847 9.441083
1997 Q4 8.594502 9.206761 9.354080 9.511258 9.654834 9.486182
      ROY      PE      TRAN      ICMS      ISS
1997 Q1 8.934018 9.199340 8.799565 8.589788 8.468764
1997 Q2 8.804211 9.299899 8.886277 8.697490 8.456690
1997 Q3 9.028303 9.485294 9.074360 8.809763 8.599470
1997 Q4 9.219753 9.471351 9.134457 8.834902 8.594502
[ reached getOption("max.print") -- omitted 72 rows ]

```

Por fim, a estrutura do resultado:

```
> str(resultado,1)
```

```

List of 33
 $ trt  :List of 8
 $ trt1 :List of 8
 $ trt2 :List of 8
 $ tfp  :List of 8
 $ tfp1 :List of 8
 $ tfp2 :List of 8
 $ trc  :List of 8
 $ trc1 :List of 8
 $ trc2 :List of 8
 $ ti   :List of 8
 $ ti1  :List of 8
 $ ti2  :List of 8
 $ tm   :List of 8
 $ tm1  :List of 8
 $ tm2  :List of 8
 $ tgc  :List of 8
 $ tgc1 :List of 8
 $ tgc2 :List of 8
 $ roy  :List of 4
 $ roy1 :List of 4
 $ roy2 :List of 4
 $ pe   :List of 4
 $ pe1  :List of 4
 $ pe2  :List of 4
 $ tran :List of 8
 $ tran1:List of 8
 $ tran2:List of 8
 $ icms :List of 8
 $ icms1:List of 8

```

```

$ icms2:List of 8
$ iss :List of 8
$ iss1 :List of 8
$ iss2 :List of 8

```

Como se vê, o resultado é uma longa lista com todos os dados estimados, sendo que para cada grupo de receita são estimado 24 modelos, oito¹³ para cada um dos três casos seguintes: Grupo de Receita contra Hiato, contra Hiato com lag e contra Hiato e Hiato com lag. Para acessar um resultado específico, digite:

```
> resultado$trt[[4]]$comp
```

	Valor	Nivel	Inclinacao	Sazon	Coef1
1	9.199340	9.243788	1.188945e-08	-0.06098927	0.3089673
2	9.299899	9.336369	1.188945e-08	-0.03749278	0.3089714
3	9.485294	9.412189	1.188944e-08	0.05062022	0.3089755
4	9.471351	9.435954	1.188947e-08	0.04786183	0.3089754
5	8.934018	8.999131	1.188947e-08	-0.06098927	0.3089755
6	8.804211	8.846012	1.188947e-08	-0.03749278	0.3089782
7	9.028303	8.982564	1.188947e-08	0.05062022	0.3089764
8	9.219753	9.170934	1.188946e-08	0.04786183	0.3089738
9	9.271527	9.332162	1.188948e-08	-0.06098927	0.3089710
10	9.317269	9.352501	1.188941e-08	-0.03749278	0.3089681

[reachedgetOption("max.print") -- omitted 54 rows]

```
> resultado$tg[[7]]$comp
```

	Valor	Nivel	Inclinacao	Sazon	Coef1
1	9.199340	9.206964	-0.008554567	-0.10043170	0.02369907
2	9.299899	9.198409	-0.008554567	-0.06948689	0.02369907
3	9.485294	9.189855	-0.008554567	0.08705765	0.02369907
4	9.471351	9.181300	-0.008554567	0.08286093	0.02369907
5	8.934018	9.172746	-0.008554567	-0.10043170	0.02369907
6	8.804211	9.164191	-0.008554567	-0.06948689	0.02369907
7	9.028303	9.155637	-0.008554567	0.08705765	0.02369907
8	9.219753	9.147082	-0.008554567	0.08286093	0.02369907
9	9.271527	9.529652	-0.008554567	-0.10043170	0.02369907
10	9.317269	9.521098	-0.008554567	-0.06948689	0.02369907

[reachedgetOption("max.print") -- omitted 34 rows]

¹³Exceção de ROY e PE para quem são estimados apenas quatro modelos, já que o componente de sazonalidade é ignorado.

exportar - Exportando resultados

O resultado final da função `calcular.elasticidades` é de complexa interpretação. Para facilitar a compreensão deste resultado, pode-se usar a função `exportar`. Esta função resume os resultados da função `calcular.elasticidades` aos valores mais importantes de modo a facilitar o acesso:

```
> resultado.exportacao <- exportar(resultado)
```

O resultado é uma grande tabela detalhando os resultados, com 28 colunas:

1. As Colunas 3, 7, 10, 13, 17, 22, 25 estão vazias para facilitar leitura em caso de exportação.
2. A Coluna 1 descreve o modelo que foi estimado.
3. A Coluna 2 descreve o período de estimação.
4. As Colunas 4, 5 e 6 descrevem as hipóteses feitas para os componentes, em que S indica comportamento estocástico e F indica comportamento fixo.
5. As Colunas 8 e 9 indicam o valor médio dos coeficientes.
6. As Colunas 11 e 12 indicam os p-valores dos testes t de cada coeficiente. O teste t é realizado apenas no último período.
7. As Colunas 14 e 15 indicam os p-valores dos testes de correlação serial dos erros de predição.
8. A Coluna 16 diz quantos lags foram usados para realizar o teste da Coluna 14. Já o teste da Coluna 15 utiliza 2 vezes esse valor.
9. As Colunas 18 e 19 indicam, respectivamente, os p-valores dos testes de homocedasticidade e normalidade.
10. As Colunas 20 e 21 são os critérios de informação de Akaike (AIC) e bayesiano (BIC).
11. As Colunas 23 e 24 indicam os desvios padrão dos choques da equação do coeficiente. Quanto menor esse valor mais estável é a elasticidade ao longo do tempo.
12. A Coluna 26 detalha as intervenções utilizadas.

13. As Colunas 27 e 28 indicam, de forma similar às colunas 8 e 9, os valores dos coeficientes no último período. Se a elasticidade é constante (colunas 23 ou 24 igual a 0), essas colunas são numericamente iguais às coluna 8 e 9.

A fim de facilitar a análise dos resultados, recomenda-se salvar a tabela gerada em CSV, e abri-la em Excel, o que pode ser feito com o seguinte comando¹⁴

```
> write.csv2(resultado.exportacao, file="Caminho/Desejado/Nome.csv")
```

Script

Alternativamente à função `calcular.elasticidades`, é possível baixar no site da SPE/MF arquivo de script R que auxilia a estimação das elasticidades. O script é, em grande parte, o código-fonte da função `calcular.elasticidades` e faz-se útil para estimar subconjuntos do total de modelos possíveis, evitando estimações desnecessárias. Uma vez que estas podem ter alto custo computacional e demorar vários minutos para serem interpretadas, e para adicionar flexibilidade caso se deseje alterar os períodos iniciais de estimação.

Segue o Script:

```
> arquivo <- "elasticidades.csv"
> Receitas <- NULL
> Hpib <- NULL
> Hpet <- NULL
> fim <- c(2016,4)
> tce <- NULL
> comeco <- list(
+   trt = 2000,
+   tfp = 1997,
+   trc = 1997,
+   ti = 2001,
+   tm = 2005,
+   tgc = 2005,
+   roy = 2000,
+   pe = 2000,
+   tran = 2002,
+   icms = 1997,
```

¹⁴O “2” em `write.csv2` força a utilização do padrão CSV europeu, ou seja vírgulas separam decimais, pontos separam milhares e pontos e virgulas separam colunas, que é padrão mais comum no Brasil. Porém, se o leitor desejar utilizar o padrão americano é possível omitir o “2”, e usar apenas `write.csv`.

```

+   iss = 2002)
> resultado <- list(
+   trt = decompor.todos(Receitas[, "TRT"] , X = Hpib,
+                       comeco = comeco$trt , fim = fim),
+   trt1 = decompor.todos(Receitas[, "TRT"] , X = lag(Hpib,-1),
+                       comeco = comeco$trt , fim = fim),
+   trt2 = decompor.todos(Receitas[, "TRT"] , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$trt , fim = fim),
+
+   tfp = decompor.todos(Receitas[, "TFP"] , X = Hpib,
+                       comeco = comeco$tfp , fim = fim),
+   tfp1 = decompor.todos(Receitas[, "TFP"] , X = lag(Hpib,-1),
+                       comeco = comeco$tfp , fim = fim),
+   tfp2 = decompor.todos(Receitas[, "TFP"] , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$tfp , fim = fim),
+
+   trc = decompor.todos(Receitas[, "TRC"] , X = Hpib,
+                       comeco = comeco$trc , fim = fim),
+   trc1 = decompor.todos(Receitas[, "TRC"] , X = lag(Hpib,-1),
+                       comeco = comeco$trc , fim = fim),
+   trc2 = decompor.todos(Receitas[, "TRC"] , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$trc , fim = fim),
+
+   ti  = decompor.todos(Receitas[, "TI"]  , X = Hpib,
+                       comeco = comeco$ti  , fim = fim),
+   ti1 = decompor.todos(Receitas[, "TI"]  , X = lag(Hpib,-1),
+                       comeco = comeco$ti  , fim = fim),
+   ti2 = decompor.todos(Receitas[, "TI"]  , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$ti  , fim = fim),
+
+   tm  = decompor.todos(Receitas[, "TM"]  , X = cbind(Hpib, log(tce)),
+                       comeco = comeco$tm , fim = fim),
+   tm1 = decompor.todos(Receitas[, "TM"]  , X = cbind(lag(Hpib,-1), log(tce)),
+                       comeco = comeco$tm , fim = fim),
+   tm2 = decompor.todos(Receitas[, "TM"] ,
+                       X = cbind(Hpib, lag(Hpib, -1), log(tce)),
+                       comeco = comeco$tm , fim = fim),
+
+   tgc = decompor.todos(Receitas[, "TGC"] , X = Hpib,
+                       comeco = comeco$tgc , fim = fim),
+   tgc1 = decompor.todos(Receitas[, "TGC"] , X = lag(Hpib, -1),
+                       comeco = comeco$tgc , fim = fim),
+   tgc2 = decompor.todos(Receitas[, "TGC"] , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$tgc , fim = fim),

```

```

+
+   roy = decompor.todos(Receitas[, "ROY" ] , X = Hpet,
+                       comeco = comeco$roy , fim = fim),
+   roy1 = decompor.todos(Receitas[, "ROY" ] , X = lag(Hpet, -1),
+                       comeco = comeco$roy , fim = fim),
+   roy2 = decompor.todos(Receitas[, "ROY" ] , X = cbind(Hpet, lag(Hpet, -1)),
+                       comeco = comeco$roy , fim = fim),
+
+   pe  = decompor.todos(Receitas[, "PE" ] , X = Hpet,
+                       comeco = comeco$pe , fim = fim),
+   pe1 = decompor.todos(Receitas[, "PE" ] , X = lag(Hpet, -1),
+                       comeco = comeco$pe , fim = fim),
+   pe2 = decompor.todos(Receitas[, "PE" ] , X = cbind(Hpet, lag(Hpet, -1)),
+                       comeco = comeco$pe , fim = fim),
+
+   tran = decompor.todos(Receitas[, "TRAN"], X = Hpib,
+                       comeco = comeco$tran, fim = fim),
+   tran1= decompor.todos(Receitas[, "TRAN"], X = lag(Hpib, -1),
+                       comeco = comeco$tran, fim = fim),
+   tran2= decompor.todos(Receitas[, "TRAN"], X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$tran, fim = fim),
+
+   icms = decompor.todos(Receitas[, "ICMS"], X = Hpib,
+                       comeco = comeco$icms, fim = fim),
+   icms1= decompor.todos(Receitas[, "ICMS"], X = lag(Hpib,-1),
+                       comeco = comeco$icms, fim = fim),
+   icms2= decompor.todos(Receitas[, "ICMS"], X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$icms, fim = fim),
+
+   iss  = decompor.todos(Receitas[, "ISS" ] , X = Hpib,
+                       comeco = comeco$iss , fim = fim),
+   iss1 = decompor.todos(Receitas[, "ISS" ] , X = lag(Hpib, -1),
+                       comeco = comeco$iss , fim = fim),
+   iss2 = decompor.todos(Receitas[, "ISS" ] , X = cbind(Hpib, lag(Hpib, -1)),
+                       comeco = comeco$iss , fim = fim)
+ )
> ret <- exportar(resultado)
> write.csv2(ret, file=arquivo)

```

No arquivo disponibilizado, ainda existem comentários explicando diversos trechos que foram omitidos das linhas acima por questões de simplicidade.

Para se utilizar o script, é imprescindível que se defina as seguintes variáveis: `arquivo`, `Receitas`, `Hpib`, `Hpet`, `comeco`, `fim` e `tce`. Todas estão

localizadas na parte inicial do script e algumas como `comeco` e `fim` já possuem valores pré-estabelecidos.

A variável `arquivo` contém o caminho e o nome do arquivo onde serão salvos os resultados. As variáveis `Hpib` e `Hpet` correspondem, respectivamente, às séries de tempo do Hiato do Produto e do Hiato do Petróleo. A variável `tce` define uma taxa de câmbio efetiva¹⁵ utilizada como variável de controle para estimar a elasticidade das receitas de importação, caso não se queira utilizar essa variável. é possível defini-la como `NULL`.

As variáveis `comeco` e `fim` definem os períodos iniciais e finais de estimação, sendo que a variável `fim` é vetor ordinário que será aplicado a todos os grupos de receita. Já a variável `comeco` é uma lista que define um período inicial de estimação para cada grupo de receita, devidamente identificada pela sigla do grupo de receita em questão.

O objeto `Receitas` deve ser uma tabela com todos os 11 grupos de receita utilizados na metodologia da SPE/MF devidamente identificados com as siglas de cada um desses grupos, como definidos anteriormente¹⁶. Caso não se queira estimar algum dos grupos de receita é possível não incluí-lo na variável, porém, neste caso será necessário excluir ou comentar a linha do grupo de receita correspondente do comando de criação da variável `resultado`.

O resultado do script original será idêntico ao resultado da função `calcular.elasticidades`, desde que com os mesmos parâmetros.

¹⁵Foi utilizado, na metodologia da SPE, a série 11752 do Sistema Gerenciador de Séries Temporais do Banco Central do Brasil.

¹⁶Vide seção `calcular.elasticidades`