

Equilíbrio Geral Computável: Implementando o MINIMAL no R

Paulo Felipe Oliveira

 paulofelipe

Resumo

zzzz

Este documento tem o objetivo de discutir a implementação de um modelo de equilíbrio geral simples no R usando o pacote ‘emr’. Como exemplo, utiliza-se o MINIMAL que é um modelo desenvolvido com propósitos educacionais pelos criadores do GEMPACK. Este modelo é de um único país com múltiplos setores, múltiplas fontes de demanda e comércio internacional. A definição dos modelos segue a abordagem descrita em [Zhang \[2013\]](#). Os resultados obtidos com o ‘emr’ são consideravelmente próximos daqueles obtidos com o GEMPACK, ainda que aqui o modelo seja definido em variações relativas, diferentemente do GEMPACK que utiliza linearizações e métodos auxiliares para obter soluções mais exatas.

1. INTRODUÇÃO

Os modelos econômicos constituem uma ferramenta fundamental para o desenvolvimento de análises por economistas. Para análises *ex ante*, tradicionalmente, são utilizados os modelos de equilíbrio parcial ou de equilíbrio geral. Enquanto os modelos de equilíbrio parcial estão focados em um mercado específico, os modelos de equilíbrio geral focam na interação entre os diversos setores da economia. Conforme [Burfisher \[2017\]](#), o modelo de equilíbrio geral é economicamente amplo porque considera as motivações e o comportamento de consumidores e produtores e as suas ligações.

Apesar da microeconomia utilizada nos modelos tradicionais de equilíbrio geral ser relativamente simples, a implementação de um modelo completo pode ser uma tarefa não trivial. Para facilitar essa tarefa, duas linguagens específicas foram desenvolvidas: GEMPACK e GAMS. É incontestável que, muito provavelmente, especialistas em modelos de equilíbrio geral irão trabalhar com uma dessas duas ferramentas. Contudo, vale ressaltar que essas ferramentas necessitam da aquisição de licenças para que seja possível trabalhar com modelos mais complexos em termos do número de variáveis/equações.

O MINIMAL¹ ([Horridge and Powell \[2001\]](#)) é um modelo de equilíbrio geral simplificado utilizado em cursos introdutórios do GEMPACK. A partir desse modelo, é possível deixar claro como a teoria microeconômica do consumidor e do produtor podem ser combinadas para a construção de um modelo de equilíbrio geral.

¹<https://www.copsmodels.com/minimal.htm>

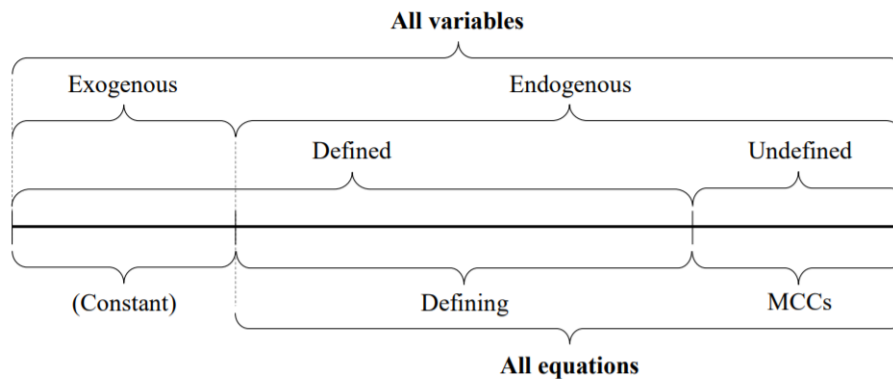
Este modelo considera 7 setores, um investidor agregado, um agente representativo das famílias, exportações agregadas e o governo. Cada produto pode ser obtido a partir de fontes domésticas ou importadas.

2. ESTRUTURAS DE MODELO PARA O PACOTE EMR

Para definirmos modelo com o pacote `emr` é preciso seguir uma determinada estrutura para a definição de parâmetros, variáveis e equações. No `emr`, é seguida a lógica descrita em [Zhang \[2013\]](#). Conforme o referido autor, a crescente complexidade dos modelos de equilíbrio geral torna-se uma barreira de entrada aos potenciais novos analistas que desejam trabalhar com esse tipo de ferramenta. [Zhang \[2013\]](#) ressalta que o sistema de equações que deve ser resolvido é desnecessariamente grande, uma vez que várias equações e variáveis são acessórias.

Para ajudar a evitar esse problema, [Zhang \[2013\]](#) define um sistema de estruturação de modelos de equilíbrio geral. Nesse sistema, existem apenas dois tipos de equações. O primeiro tipo define um conjunto de variáveis que descrevem a teoria associada ao modelo. Por outro lado, o segundo tipo de equações define condições de equilíbrio que devem ser satisfeitas. Adicionalmente, cada tipo de equação está associado a um tipo de variável, que chamaremos de definidas (*defined*) e não-definidas (*undefined*). O sistema é resumido na Figura 1.

Figura 1. Estrutura das variáveis e das equações ([Zhang \[2013\]](#))



Inicialmente, observa-se que, como usualmente é definido, as variáveis podem ser exógenas ou endógenas. No entanto, é apresentada uma nova classificação de variáveis: *defined* e *undefined*. As variáveis definidas são aquelas que são constantes, definidas fora do modelo, ou são definidas diretamente a partir dos valores das demais variáveis. Por outro lado, as variáveis não-definidas não têm uma equação que define o seu valor a partir das demais variáveis ou o seu valor é definido de forma implícita, o que implica que é preciso “chutar” um valor inicial pra essa variável e depois checar se o valor final da variável é igual ao chute inicial. Se não, o “chute” é atualizado. As variáveis definidas são associadas com equações definidoras (*defining*) e as não-definidas são associadas às equações de equilíbrio “mcc” (*market clearing codition*).

No pacote `emr`, tem-se três funções principais que serão usadas para criar as variáveis exógenas, as variáveis endógenas e as equações:

- `create_param()`: cria parâmetros que são, no sentido amplo, as variáveis exógenas e outras constantes do modelo;
- `create_variable()`: cria variáveis (*defined* e *undefined*) para o modelo;
- `create_equation()`: cria as equações (*defining* e *mcc*) para o modelo.

Além dos parâmetros, das variáveis e das equações, o modelo também é formado por outros componentes: *sets* e equações de atualizações (*update equations*). Os *sets* armazenam os índices de variáveis e parâmetros. Por exemplo, suponha que X_i seja uma variável e i possa assumir os valores [AGR, IND, SER], assim esse vetor de índices será chamado de *set*. Por sua vez, as equações de atualização servem para atualizar os valores iniciais de alguns parâmetros a partir dos resultados da simulação. Por exemplo, pode-se criar um parâmetro que guarde a informação da quantidade consumida no dados iniciais, $Q0$. Após a simulação, encontramos uma variação na quantidade consumida. Essa variação pode ser utilizada para atualizar o valor de $Q0$ no novo equilíbrio.

Apesar do nosso foco ser em modelos de equilíbrio geral, vale a pena introduzir o pacote com um modelo bastante simples de equilíbrio parcial.

2.1 Exemplo: Oferta e Demanda

Aqui, vamos apresentar como um modelo pode ser construído e simulações podem ser realizadas com o pacote `emr`. Para isso, iremos trabalhar com o seguinte modelo:

- Curva de demanda:

$$q^d = k^d [p(1 + t)]^\eta, \quad \eta < 0$$

- Curva de oferta:

$$q^s = k^s p^\epsilon, \quad \epsilon > 0$$

- Equilíbrio de mercado:

$$q^d = q^s$$

A Tabela 1 detalha as variáveis e equações do modelo. Nosso modelo tem 3 variáveis (q^d , q^s e p) e três equações. No entanto, perceba que as duas primeiras variáveis são definidas a partir do preço p . Dessa forma, o nosso “sistema” terá uma única equação de equilíbrio e que estará associado a variável p . Assim, q^d e q^s são definidas e p é não definida.

Tabela 1. Parâmetros e Variáveis do Modelo de Oferta e Demanda

	Classe	Tipo	Descrição
q^d	Variável	Definida	Quantidade demanda
q^s	Variável	Definida	Quantidade ofertada
p	Variável	Não-definida	Preço
t	Parâmetro	-	Imposto
k^d	Parâmetro	-	Deslocamento da curva de demanda
k^s	Parâmetro	-	Deslocamento da curva de oferta

Fonte: Elaboração própria.

Também é interessante introduzir a lógica de escrita de modelos em variações. Os modelos no GEMPACK são escritos em variações percentuais obtidas a partir das linearizações das equações do modelo. No emr, apesar de não ser obrigatório, em regra, iremos escrever o modelo em variações relativas, a chamada *exact-hat algebra*² introduzida em Dekle et al. [2008].

A variação relativa de uma variável x entre dois equilíbrios é denotada por $\hat{x} = x'/x$, em que x' é o valor de x no novo equilíbrio. A principal vantagem de utilizar o modelo em variações é eliminar a necessidade de se obter os valores iniciais de algumas variáveis e parâmetros. Por exemplo, uma constante desconhecida k , em variação, é reescrita como $\hat{k} = 1$, uma vez que, a priori, o seu valor é igual nos dois equilíbrios. Se os dados iniciais representam um equilíbrio, o valor inicial da variáveis também será 1.

Reescrevendo o modelo de oferta e demanda em variações, temos as seguintes equações:

- Curva de demanda:

$$\hat{q}^d = \hat{k}^d [\hat{p}\hat{\tau}]^\eta, \quad \eta < 0$$

- Curva de oferta:

$$\hat{q}^s = \hat{k}^s \hat{p}^\epsilon, \quad \epsilon > 0$$

- Equilíbrio de mercado:

$$\hat{q}^d = \hat{q}^s$$

Nessa formulação, $\hat{\tau}$ é variação no poder do imposto, isto é:

$$\hat{\tau} = \frac{1 + t'}{1 + t}.$$

Antes de começar a implementar o modelo no R, precisamos dos dados base que se referem a algum equilíbrio inicial. Usualmente, as bases de dados para modelos de equilíbrio geral consideram apenas fluxos monetários. Na prática, redefinindo-se a unidade de medida, assume-se preço inicial igual a 1.

²Na prática, reescrever as equações usando a *exact-hat algebra* é similar a *calibrated share form* que é usada em por alguns pesquisadores em CGE.

Tabela 2. Dados Base para o Modelo de Oferta e Demanda

Variável	Valor
Valor inicial	100
Imposto inicial	0%
Elasticidade-preço da demanda (η)	-1
Elasticidade-preço da oferta (ϵ)	0.5
Fonte: Elaboração própria.	

Agora, vamos definir o modelo. Antes de mais nada, precisamos carregar o pacote e inicializar alguns componentes do modelos:

```
library(emr)
params <- list()
variables <- list()
equations <- list()
```

Com o pacote carregado, iremos definir o bloco de demanda:

```
params[["K_d"]] <- create_param(
  value = 1,
  indexes = "k_d",
  desc = "Shift na curva de demanda em variação"
)

params[["ETA"]] <- create_param(
  value = -0.5,
  indexes = "eta",
  desc = "Elasticidade-preço da demanda"
)

params[["TAU"]] <- create_param(
  value = 1,
  indexes = "tau",
  desc = "Variação no poder do imposto"
)

# Vamos definir uma variável e uma equação
variables[["q_d"]] <- create_variable(
  value = 1,
  indexes = "q_d",
  type = "defined",
  desc = "Variação na quantidade demandada"
)

equations[["E_q_d"]] <- create_equation(
  "q_d = K_d * (p * TAU)^ETA",
  type = "defining",
  desc = "Variação na quantidade demandada"
)
```

O segundo bloco é o da oferta:

```
params[["K_s"]] <- create_param(
  value = 1,
  indexes = "k_s",
  desc = "Shift (em variação) na curva de oferta"
)

params[["EPS"]] <- create_param(
  value = 0.5,
  indexes = "eta",
  desc = "Elasticidade-preço da oferta"
)

variables[["q_s"]] <- create_variable(
  value = 1,
  indexes = "q_s",
  type = "defined",
  desc = "Variação na quantidade ofertada"
)

equations[["E_q_s"]] <- create_equation(
  "q_s = K_s * p^EPS",
  type = "defining",
  desc = "Variação na quantidade ofertada"
)
```

Para definição do preço de equilíbrio, utilizamos a equação de equilíbrio de mercado (*mcc*). O único detalhe é que, ao invés de usarmos a igualdade, iremos reescrevê-la de tal forma que o seu valor tenda a zero quando o equilíbrio for encontrado. Isto é:

$$\hat{q}^d - \hat{q}^s$$

```
variables[["p"]] <- create_variable(
  value = 1,
  indexes = "p",
  type = "undefined",
  desc = "Variação no preço"
)

equations[["E_p"]] <- create_equation(
  "q_d - q_s",
  type = "mcc",
  desc = "Equação de equilíbrio de mercado"
)
```

Apesar de não ser estritamente necessário, criaremos um parâmetro que guardará o valor do equilíbrio inicial. Chamaremos de V^0 . No novo equilíbrio, tem-se que

$$V^{0'} = V^0 \times \hat{p} \times \hat{q}^d.$$

Utilizaremos as equações de atualizações para atualizar esse valor:

```

params[["V0"]] <- create_param(
  value = 100,
  indexes = "V0",
  desc = "Valor do equilíbrio inicial"
)
update_equations <- list()
update_equations[["V0"]] <- create_equation(
  "V0 = V0 * p * q_d",
  desc = "Atualização do valor de equilíbrio"
)

```

Agora, vamos criar o objeto que conterá os elementos do modelo de oferta e demanda. Este objeto deve ser uma lista nomeada como os componentes³.

```

modelo_oferta_demanda <- list(
  params = params,
  variables = variables,
  equations = equations,
  update_equations = update_equations
)

```

Adicionalmente, vamos testar se o modelo encontra-se em equilíbrio. Para isso, iremos utilizar a função `solve_emr()`⁴.

```

solucao <- solve_emr(modelo_oferta_demanda)
# A função solve_emr retorna os seguintes elementos
names(solucão)

```

```

## [1] "sol"                "params"              "variables"
## [4] "updated_data"       "variables_descriptions" "params_descriptions"

```

```

# o objeto sol apresenta os detalhes da solução
solucao$sol

```

```

## $par
## p
## 1
##
## $residual
## [1] 0
##
## $fn.reduction
## [1] 0
##

```

³Os possíveis componentes são: sets, params, variables, equations e update_equations

⁴A função `solve_emr()` utiliza o pacote BB desenvolvido por [Varadhan and Gilbert \[2009\]](#). Também é fornecida a função `solve_emr_block()` que é baseada no método DF-SANE, porém em uma versão simplificada na qual o `stepsize` e a atualização das variáveis são definidos equação a equação, além de possibilitar a utilização de um `stepsize` fixo para cada variável não-definida.

```
## $feval
## [1] 1
##
## $iter
## [1] 0
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
##
## $cpar
## method      M      NM
##      2      50      0
```

Com o modelo finalizado, vamos utilizá-lo para o que realmente interessa: realizar simulações a partir de cenários. Como exemplo, vamos supor que a tarifa inicial de 0% foi alterada para 10%. Qual é o novo equilíbrio de mercado? O que aconteceria com a quantidade e com o preço? Assim, iremos fazer $\hat{\tau} = 1.1$. O seu valor inicial é igual 1.

```
modelo_oferta_demanda$params$TAU$value
```

```
## tau
## 1
```

Para criar o choque, precisamos alterar o parâmetro TAU. Assim, basta modificar o seu valor para 1.1 (1.10/1). Note que chamamos `value[]`, a chamada com `[]` garante que o nome do elemento não seja sobrescrito.

```
modelo_oferta_demanda$params$TAU$value[] <- 1.1
```

Após a criação do choque, iremos resolver o modelo novamente. Os novos valores das variáveis estarão no elemento `variables`. O novo valor de equilíbrio, atualização de V^0 , está no elemento `updated_data`.

```
solucao_cen1 <- solve_emr(modelo_oferta_demanda)
solucao_cen1$variables
```

```
## $q_d
##      q_d
## 0.9764541
##
## $q_s
##      q_s
## 0.9764541
##
## $p
##      p
## 0.9534625
```



```
solucao_cen1$updated_data
```

```
## $V0
##      V0
## 93.10124
```

O resultado para q_d foi de 0.9765, o que indica que a nova quantidade demandada é 2,4% $((0.9765 - 1) \times 100)$ menor do que no equilíbrio inicial. A Tabela 3 apresenta as variações para a quantidade, preço do produtor e preço do consumidor. Verifica-se que o custo do imposto foi suportado, quase na mesma proporção, pelos consumidores e produtores.

Tabela 3. Variações para um Choque de 10% no Imposto

Variável	Variação
\hat{q}^d	-2,4%
\hat{p}	-4,7%
$\hat{p}\hat{\tau}$	4,9%

Fonte: Elaboração própria.

Assim, concluímos o nosso primeiro exemplo. A lógica aplicada, em grande parte, é a mesma que será utilizada em um modelo mais “complexo” de equilíbrio geral. Começaremos, então, pelo MINIMAL.

3. IMPLEMENTANDO O MINIMAL NO R

Nesta seção iremos implementar o MINIMAL, que recebe esse nome em razão de ser uma versão em “miniatura” do modelo ORANI, desenvolvido por [Horridge \[2000\]](#). Os dados para esse modelo podem ser obtidos nas matrizes de insumo e produto disponibilizadas pelo órgãos oficiais de estatísticas nacionais. Nesse documento, iremos utilizar a base de dados para a Austrália. Primeiro trabalharemos os dados e depois iremos implementar o modelo.

3.1 Dados para o Modelo

Os dados para o modelo são esquematizados conforme a Figura 2. O esquema é similar ao de uma matriz de insumo produto, na qual os elementos das linhas vendem para os elementos das colunas.

Fica claro na Figura 2 que existem duas fontes (*sources*) de fornecimento de produtos: doméstica e importada. Esse produtos são demandados pelos I setores produtores, pelos investidores, pelas famílias, pelas exportações⁵ e pelo governo. As somas das linhas para produtos

⁵O fornecimento de produtos importados para exportação é igual a zero.

Figura 2. Base de Dados para o Minimal(Horridge and Powell [2001])

		Absorption Matrix					
		1	2	3	4	5	
		Producers	Investors	Household	Export	Government	Total Sales
		← I →	← 1 →	← 1 →	← 1 →	← 1 →	
Domestic Flows	↑ C ↓	USE(commodity,"dom",user)					
Imported Flows	↑ C ↓	USE(commodity,"imp",user)					
Labour	↑ 1 ↓	FACTOR (labour)	C = Number of Commodities = 7 I = Number of Industries = 7				
Capital	↑ 1 ↓	FACTOR (capital)					
Output tax	↑ 1 ↓	V1PTX					

		Tax on Imports
Size	← 1 →	
↑ C ↓	V0MTX	

(*commodities*) domésticos ou importados serão denominadas de vendas (*sales*). Note que a numeração das colunas serão utilizadas para identificar os usuários. Por exemplo, 1 para indústrias e 3 para as famílias.

Adicionalmente, os fatores trabalhos e capital são demandados pelos produtores, e há uma taxação sobre a produção. Por fim, independente do demandante, existe um imposto de importação por produto. O valor arrecadado de imposto de importação não é separado por usuário, o que resulta em uma única taxa de importação por produto.

A tabela apresentada na Figura 3 detalha um conjunto de valores para a Austrália a partir de dados de 1986–1987. Os valores estão a preços dos produtores. Ou seja, inclui qualquer imposto indireto que possa ter sido aplicada àquele fluxo. Para cada setor produtor, a sua produção (soma da respectiva coluna) tem que ser igual às suas vendas (soma da respectiva linha). Por exemplo, a produção do setor Agricultura-Mineração (AgricMining) foi igual 45.730, que é o mesmo valor de suas vendas.

Para as importações de produtos manufaturados (*Manufacture*), foi recolhido o montante de 5.787. O valor total importado desses produtos foi de 42.087 (este valor já inclui o imposto de

importação), o que implica em uma taxa de importação de 15,94%⁶.

3.2 Implementação

Nesta subseção, iremos detalhar a estrutura teórica do modelo e implementá-lo no R com o pacote `emr`.

3.2.1 Passos Iniciais

Inicialmente, é preciso carregar o pacote `emr`:

```
library(emr)
# Carregar o tidyverse para manipulação dos dados
library(tidyverse)
```

Adicionalmente, também é necessário ler os dados que servirão de base para o modelo. Os dados foram exportados do formato HAR do GEMPACK para csv. Nessa conversão, as diversas tabelas são empilhadas em um único csv, sendo separadas por uma linha denominada de *HEADER*.

Dessa forma, vamos inicialmente identificar os *headers*:

```
minimal_headers <- read_lines('../dados/minimal.csv')
minimal_headers[str_detect(minimal_headers, "Header")]

## [1] "!Header: USE , dimensions: COM*SRC*USER [7*2*11], description: USE matrix"
## [2] "!Header: 1FAC, dimensions: FAC*IND [2*7], description: Wages and profits"
## [3] "!Header: OTAR, dimensions: COM [7], description: Import tax revenue"
## [4] "!Header: 1PTX, dimensions: IND [7], description: Production tax revenue"
## [5] "!Header: ARM , dimensions: COM [7], description: Armington elasticities"
## [6] "!Header: P028, dimensions: IND [7], description: Primary factor substitution elasticity"
## [7] "!Header: P018, dimensions: COM [7], description: Export demand elasticities"
```

A Tabela 4 detalha em quais linhas as tabelas se iniciam de fato e quantas linhas de dados existem em cada tabela. Por exemplo, a tabela `USE`, que contém os dados de uso por produto, origem e usuário, inicia-se na linha 2 e encerra na linha 156, sendo 154 linhas de dados e uma com os títulos de cada coluna.

O código abaixo lê as diferentes tabelas, disponibilizando os dados que serão utilizados pelo modelo.

⁶5787/(42087 - 5787).

Figura 3. Base de Dados para Austrália (Milhões 1986-1987)

	All Users										
	Industries						Final Demands				
	Agric	Mining	Manufacture	Utilities	Construction	Trade	Transpt	Finan	Property	Services	Total
Domestic											
Agric	5502	14658	1839	689	143	52	641				45730
Mining	4587	30009	643	12486	10200	3061	6947				127264
Manufacture	1345	2045	3261	176	979	2814	2037				16401
Utilities	89	55	13	0	438	1708	381				40201
Construction	2958	11539	694	3353	8892	3052	5680				88793
Trade	1754	6545	622	1886	9623	9819	6111				74520
Transpt	403	1595	92	290	1316	1586	2210				80801
Finan											
Property											
Services											
Imported											
Agric	233	1677	1	49	7	3	145				2470
Mining	1305	12411	184	2518	2322	832	3232				42087
Manufacture	1	2	2	0	1	3	2				14
Utilities	0	1	0	0	3	0	8				82
Construction	104	259	11	34	703	142	258				2599
Trade	90	302	19	29	328	274	209				1470
Transpt	26	451	7	55	117	66	774				2312
Finan											
Property											
Services											
Labour	10779	22512	3594	15008	35532	17095	43346				147866
Capital	11337	6359	4293	2160	10409	28873	4612				68043
Production tax	5217	16844	1126	1468	7780	5140	4208				41783
Total Cost	45730	127264	16401	40201	88793	74520	80801	60839	150959	40112	782436
Tax on imports	497	5787	0	0	0	27	52				

Tabela 4. Posições dos Headers no Arquivo minimal.csv

Nome	Início	Fim	Nº de Linhas
USE	2	156	154
1FAC	158	172	14
0TAR	174	181	7
1PTX	183	190	7
ARM	192	199	7
P028	201	208	7
P018	210	217	7

Fonte: Elaboração própria.

```
# dados de uso (demanda)
use_df <- read_csv(
  file = '../dados/minimal.csv',
  skip = 1,
  n_max = 154,
  col_types = 'cccd'
)

# uso de fatores primários (capital e trabalho)
fac_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 157,
  n_max = 14,
  col_types = 'ccd'
)

# arrecadação de imposto de importação
tar_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 173,
  n_max = 7,
  col_types = 'cd'
)

# arrecadação de imposto sobre a produção
ptx_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 182,
  n_max = 7,
  col_types = 'cd'
)

# elasticidades de armington
arm_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 191,
```

```

n_max = 7,
col_types = 'cd'
)

# elasticidade substituição para os fatores primários
p028_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 200,
  n_max = 7,
  col_types = 'cd'
)

# elasticidade-preço das exportações
p018_df <- read_csv(
  file = "../dados/minimal.csv",
  skip = 209,
  n_max = 7,
  col_types = 'cd'
)

```

3.2.2 Conjuntos (Sets)

Aqui, iremos definir os conjuntos de índices que são utilizados pelas variáveis do modelo. Por exemplo, a variável de produção é definida por produto (*commodity*) pertencente ao conjunto COM, que é composto pela descrição de todos os produtos.

Para implementação do modelo, precisamos de uma lista nomeada *sets*, na qual cada elemento recebe o nome do conjunto e seus possíveis valores.

Os conjuntos do modelo são:

- IND: indústrias;
- SRC: origem (doméstica ou importada);
- COM: produtos;
- USER: usuários (fontes de demanda);
- IMPUSER: usuários que demandam produtos importados;
- FINALUSER: usuários que compõem a absorção final da economia;
- FAC: fatores primários (capital e trabalho).

Abaixo, o código para criar os conjuntos. O objeto *sets* deve ser uma lista nomeada com todos os conjuntos que o modelo utilizará.

```

IND <- c("AgricMining", "Manufacture", "Utilities", "Construction",
        "TradeTranspt", "FinanProppty", "Services")

COM <- c("AgricMining", "Manufacture", "Utilities", "Construction",
        "TradeTranspt", "FinanProppty", "Services")

SRC <- c("dom", "imp")

```

```

USER <- c("AgricMining", "Manufacture", "Utilities", "Construction",
         "TradeTranspt", "FinanProprty", "Services", "Investment",
         "Households", "Government", "Exports")

IMPUSER <- c("AgricMining", "Manufacture", "Utilities", "Construction",
            "TradeTranspt", "FinanProprty", "Services", "Investment",
            "Households", "Government")

FINALUSER <- setdiff(USER, IND)

FAC <- c("Labour", "Capital")

sets <- list(
  IND = IND,
  COM = COM,
  SRC = SRC,
  USER = USER,
  IMPUSER = IMPUSER,
  FINALUSER = FINALUSER,
  FAC = FAC
)

```

3.2.3 Preços

Inicialmente, vamos definir os preços dos produtos $c \in \text{COM}$ fornecidos pelas fontes $s \in \text{SRC}$:

$$P_{cs} = \begin{cases} \text{P1TOT}_c \times \text{PTX}_c & \text{se } s = \text{dom} \\ \text{PWORLD}_c \times \phi \times \text{mtx}_c & \text{se } s = \text{imp}, \end{cases} \quad (1)$$

em que P_{cs} é o preço do produto c de origem s , P1TOT_c é o custo (marginal) de produção do produto c , PTX_c é o poder da imposto sobre a produção ($1 + \text{imposto sobre a produção}$), PWORLD_c é o preço internacional do produto c , ϕ é a taxa de câmbio⁷ e mtx_c é o poder da tarifa sobre a importação do produto c .

Como no modelo de oferta e demanda, iremos utilizar as equações na forma de variação relativa. Assim, iremos reescrever 1 como:

$$\hat{P}_{cs} = \begin{cases} \text{P1}\hat{\text{TOT}}_c \times \text{P}\hat{\text{T}}\text{X}_c & \text{se } s = \text{dom} \\ \text{P}\hat{\text{W}}\text{ORLD}_c \times \hat{\phi} \times \hat{\text{m}}\text{tx}_c & \text{se } s = \text{imp}, \end{cases} \quad (2)$$

No MINIMAL, os bens domésticos e importados são agregados em um bem composto utilizando uma função do tipo CES. O índice de preço desse bem composto é denotado por P_{cu}^s , sendo calculado da seguinte forma:

$$\hat{P}_{cu}^s = \left[\sum_{s \in \text{SRC}} \text{SRC SHARE}_{csu} \hat{P}_{cs}^{1-\sigma_c} \right]^{\frac{1}{1-\sigma_c}}, \quad c \in \text{COM}, u \in \text{IMPUSER}$$

⁷A taxa de câmbio é usada como numerário no modelo

em que SRC SHARE_{csu} é participação no consumo do produto c da origem s no dispêndio do usuário u . O parâmetro σ_c é a elasticidade de substituição entre a origem doméstica e a origem importada para o produto c , também conhecida como elasticidade de Armington. Os valores dessas elasticidades estão no *header* ARM que importamos anteriormente.

```
head(arm_df)
```

```
## # A tibble: 6 x 2
##   COM      Value
##   <chr>    <dbl>
## 1 AgricMining    2
## 2 Manufacture    2
## 3 Utilities      2
## 4 Construction  2
## 5 TradeTranspt  2
## 6 FinanProprty  2
```

Com a definição desse bloco de preços, já podemos iniciar a implementação do modelo. Dessa forma, precisamos definir as listas que guardarão mais componentes do modelo:

```
params <- list()
variables <- list()
equations <- list()
```

Agora, vamos começar a definir os parâmetros das equações acima.

```
params[["PTX"]] <- create_param(
  value = 1,
  indexes = sets['IND'],
  desc = "Variação no poder do imposto sobre a produção"
)

params[["PWORLD"]] <- create_param(
  value = 1,
  indexes = sets['COM'],
  desc = "Variação no preço internacional do produto c"
)

params[["PHI"]] <- create_param(
  value = 1,
  indexes = "PHI",
  desc = "Variação na taxa de câmbio"
)

params[["MTX"]] <- create_param(
  value = 1,
  indexes = sets['COM'],
  desc = "Variação no poder da tarifa de importação do produto c"
)
```



```

params[["SIGMA"]] <- create_param(
  value = arm_df,
  indexes = sets[c("COM")],
  desc = "Elasticidade de Armington"
)

SRCSHARE <- use_df %>%
  filter(USER %in% IMPUSER) %>%
  group_by(COM, USER) %>%
  mutate(SRCSHR = Value/sum(Value),
         SRCSHR = ifelse(is.nan(SRCSHR), 0.5, SRCSHR)) %>%
  select(COM, SRC, IMPUSER = USER, SRCSHR)

params[["SRCSHARE"]] <- create_param(
  value = SRCSHARE,
  indexes = sets[c("COM", "SRC", "IMPUSER")],
  desc = "Participação da origem s no consumo do produto c pelo usuário u"
)

```

Consideramos que $PWORLD_c$ é um parâmetro, ou seja, é exógeno. Isto significa que foi assumido que os preços internacionais são dados e a demanda do país analisado não tem poder para alterar os preços internacionais.

O segundo passo para esse bloco é definir as variáveis. Iremos definir \hat{P}_{cs} e \hat{P}_{cu}^s . A variável $P1\hat{T}OT$ não será definida agora, pois não temos uma equação para definir o seu valor. Veremos adiante que o valor de $P1\hat{T}OT$ é encontrada a partir de um equação de equilíbrio de mercado (mcc) que define que $P1\hat{T}OT$ tem que ser igual à variação do custo de produção (lucro zero).

```

variables[["p"]] <- create_variable(
  value = 1,
  indexes = sets[c("COM", "SRC")],
  type = "defined",
  desc = "Variação no preço do produto c de origem s"
)

variables[["p_s"]] <- create_variable(
  value = 1,
  indexes = sets[c("COM", "IMPUSER")],
  type = "defined",
  desc = "Variação no índice de preço do bem composto c para o usuário u"
)

```

Por último, iremos definir as equações. Podemos, inclusive, as estruturas de controle do R (if()).

```

equations[["E_p"]] <- create_equation(
  'if(s == "dom"){
    p[c,s] = p1tot[c] * PTX[c]
  } else {
    p[c,s] = PWORLD[c] * PHI * MTX[c]
  }

```

```

}',
indexes = c("c in COM", "s in SRC"),
type = "defining",
desc = "Variação no preço do produto c de origem s"
)

equations[["E_p_s"]] <- create_equation(
  'p_s[c,u] = sum(SRCSHARE[c,,u] * p[c,]^(1-SIGMA[c])) ^
    (1/(1-SIGMA[c]))',
  indexes = c('c in COM', 'u in IMPUSER'),
  type = "defining",
  desc = "Variação no índice de preço do bem composto c para o usuário u"
)

```

Note que para o somatório usamos a função `sum()` do R. Além disso, tem-se que `SRCSHARE` tem 3 dimensões e queremos somar em relação a origem, que é a segunda dimensão. Então, omitimos o índice de origem (`SRC`) e a soma ocorrerá nessa dimensão. No entanto, nem sempre será possível utilizar diretamente a função `sum()` do R. Isto porque, a soma pode ocorrer em expressões mais complexas. Para esses casos, teremos que utilizar uma notação específica do pacote `emr`, na qual a expressão de soma é destacada entre duas `@`. Por exemplo, no caso da equação `E_p_s`, podemos reescrevê-la como:

```

equations[["E_p_s"]] <- create_equation(
  'p_s[c,u] = @sum_emr("SRCSHARE[c,s,u] * p[c,s]^(1-SIGMA[c])", "s", "SRC")@^(1/(1-SIGMA[c]))',
  indexes = c('c in COM', 'u in IMPUSER'),
  type = "defining",
  desc = "Variação no índice de preço do bem composto c para o usuário u"
)

```

A notação para esses casos é o seguinte `@soma_emr("expressao", "indice in set")@`. Vale ressaltar, conforme será visto adiante, que é possível incluir somas dentro de somas, o que ocorre quando a expressão possui múltiplos somatórios.

Além dos preços para os produtos c , existem os mercados de trabalho (*labor*) e de capital que possuem seus respectivos preços, $P1LAB_i$ (salário) e $P1CAP_i$ (remuneração do capital). Note que, inicialmente, esses preços estão indexados à indústria que utiliza o fator de produção. Isto ocorre quando assumimos que o fator de produção é específico da indústria. Se o fator de produção tem mobilidade entre os setores, haverá um único preço para o fator de produção e o índice i será removido. Isso dependerá do fechamento que será escolhido.

Além desses preços, o modelo considera índices de preços da cesta de consumos do investimento ($P2TOT$), das famílias ($P3TOT$) e do governo ($P5TOT$). Estes índices serão posteriormente definidos dentro do seu respectivo bloco.

3.2.4 Produção

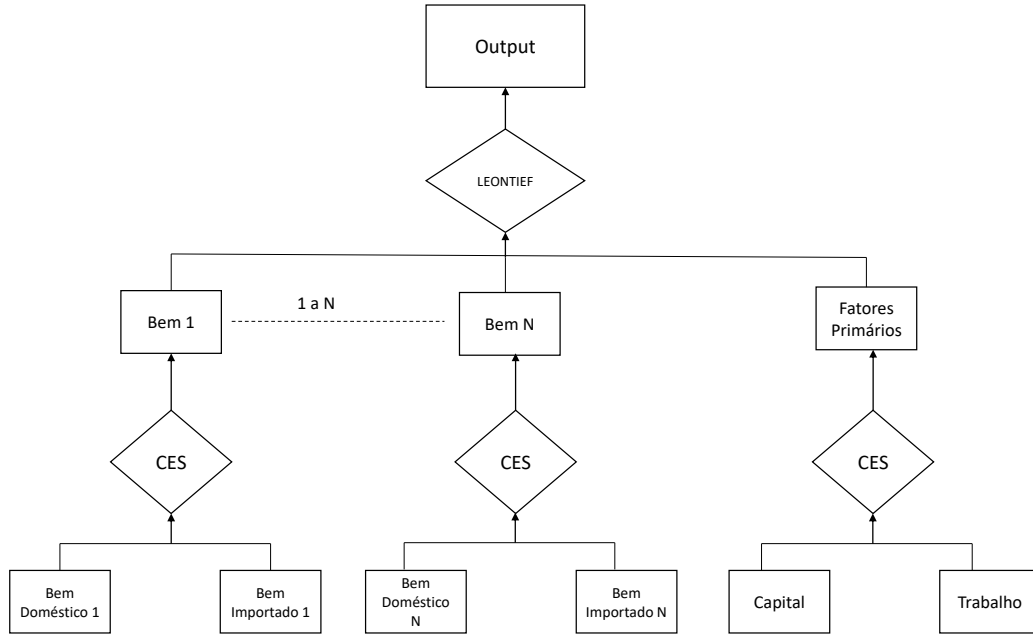
A estrutura de produção utilizada no minimal é apresentada na Figura 4. A estrutura adotada considera um primeiro nível em que o produtor demanda bens intermediários (*commodities*) e

o fator primário, que é uma combinação de capital e trabalho. É assumida uma tecnologia do tipo Leontief. Dessa forma, pode-se definir o primeiro nível da produção como:

$$X1TOT_i = \min \left\{ \frac{X_{c\{c \in COM\}i}}{A_{c\{c \in COM\}i}}, \frac{X1PRIM_i}{A1PRIM_i} \right\}, \quad i \in IND, \quad (3)$$

em que $X1TOT_i$ é a produção total da i -ésima indústria, X_{ci}^s é a demanda pelo produto c pela indústria i e $X1PRIM_i$ é a demanda por fatores primários pela indústria i . A_{ci} e $A1PRIM$ podem ser entendidos como coeficientes técnicos da matriz de insumo-produto. Isto é, necessita-se A_{ci} unidades do produto c para se produzir uma unidade de i .

Figura 4. Estrutura da Produção



Para essa tecnologia, tem-se as seguintes funções de demanda:

- Demanda por bens intermediários (compostos)⁸:

$$X_{ci}^s = A_{ci} \times X1TOT_i, \quad i \in IND, \quad c \in COM$$

- Demanda por valor adicionado:

$$X1PRIM_i = A1PRIM_i \times X1TOT_i$$

Essas equações, facilmente, podem ser reescritas em variações exatas:

⁸Os bens intermediários compostos são uma composição entre produtos domésticos e importados.

- Demanda por bens intermediários:

$$\hat{X}_{ci}^s = \hat{A}_{ci} \times X1\hat{T}OT_i, \quad i \in IND, \quad c \in COM$$

- Demanda por valor adicionado:

$$X1\hat{P}RIM_i = A1\hat{P}RIM_i \times X1\hat{T}OT_i$$

Na sequência, vamos definir os parâmetros \hat{A}_{ci} e $A1\hat{P}RIM_i$. Como estamos usando em variação, o valor inicial desses parâmetros é igual a 1.

```
params[["A"]] <- create_param(
  value = 1,
  indexes = sets[c('COM', 'IND')],
  desc = "Variação do coeficiente técnico para o produto c usado pela indústria i"
)

params[["A1PRIM"]] <- create_param(
  value = 1,
  indexes = sets['IND'],
  desc = "Variação do coeficiente técnico para o fator primário usado pela indústria i"
)
```

Agora, vamos definir as variáveis \hat{X}_{ci} (uso do produto composto c pela indústria i) e $X1\hat{P}RIM_i$ (uso do fator primário composto). Aqui, existe um detalhe, o uso do produto c pode ser feito pelas indústrias ou pelos demais usuários que demandam importações (IMPUSER). Dessa forma, na definição da variável \hat{X}_{ci} , vamos utilizar o conjunto IMPUSER ao invés do conjunto IND. Deixando mais claro, iremos definir

$$\hat{X}_{cu}^s, \quad c \in COM, u \in IMPUSER$$

$$\hat{X}_{cu}^s = \begin{cases} \hat{X}_{ci}^s & \text{se } u \in IND \\ \hat{X}_{c,HH}^s & \text{se } u = \text{Households} \\ \hat{X}_{c,GOV}^s & \text{se } u = \text{Government} \\ \hat{X}_{c,INV}^s & \text{se } u = \text{Investment} \end{cases}$$

Ambas as variáveis são do tipo *defined*, pois possuem equações que definem os seus valores.

```
# chamamos de x de x_s (composto de várias sources s)
variables[["x_s"]] <- create_variable(
  value = 1,
  indexes = sets[c("COM", "IMPUSER")],
  type = "defined",
  desc = "Variação no uso do composto c por impuser"
)

variables[["x1prim"]] <- create_variable(
```

```

value = 1,
indexes = sets[c('IND')],
type = "defined",
desc = "Uso do fator primário composto por indústria"
)

```

Finalmente, iremos definir as equações. Note que temos a equação de uso para o produto composto c para os usuários industriais. Assim, iremos utilizar o índice i in IND ($i \in IND$).

```

equations[["E_x_s_ind"]] <- create_equation(
  'x_s[c,i] = A[c,i] * x1tot[i]',
  indexes = c('c in COM', 'i in IND'),
  type = "defining",
  desc = "Variação do uso do composto c por indústria"
)

```

```

equations[["E_x1prim"]] <- create_equation(
  'x1prim[i] = A1PRIM[i] * x1tot[i]',
  indexes = 'i in IND',
  type = "defining",
  desc = "Uso do fator primário composto por indústria"
)

```

Perceba que ainda não especificamos a variável $X1TOT_i$. Ela será especificada em momento oportuno, mas vale antecipar que essa variável é do tipo *mcc*. Por quê? Perceba que, na Equação 3, $X1TOT_i$ é função dos usos de bens intermediários e fatores primários. No entanto, a quantidade demandada desses bens e fatores dependem de $X1TOT$. Dessa forma, $X1TOT$ não pode ser definida. Precisaremos de uma condição de equilíbrio de mercado para encontrar o seu valor no novo equilíbrio. Isto será feito posteriormente.

Adicionalmente, por enquanto, não entraremos nos detalhes sobre a demanda no segundo nível (escolha entre bens domésticos e importados), tendo em vista que a forma da demanda é comum independente do usuário (indústrias, família, governo etc.).

3.2.5 Famílias

Para as famílias, que serão representadas como HH , assume-se um agente representativo com preferências do tipo Cobb-Douglas sobre um conjunto de produtos (compostos) e uma restrição orçamentária. Isto é:

$$\begin{aligned}
 U &= \prod_{c \in \text{COM}} X_{c,HH}^s \alpha_c \\
 \text{s.a. } \sum_{c \in \text{COM}} P_{c,HH} X_{c,HH}^s &= W3TOT,
 \end{aligned}$$

em que $X_{c,HH}$ é quantidade demandada do bem composto c pelas famílias (HH), $P_{c,HH}$ é o índice de preço do bem composto c para as famílias e $W3TOT$ é renda nominal das famílias. Adicionalmente, $\sum_{c \in \text{COM}} \alpha_c = 1$.

Para esse tipo de preferência, sabe-se que, a partir da maximização de utilidade do consumidor, a função de demanda ótima é:

$$X_{c,HH}^s = \alpha_c \frac{W3TOT}{P_{c,HH}}, \quad c \in \text{COM}.$$

Em variações, a demanda das famílias é escrita da seguinte forma:

$$\hat{X}_{c,HH}^s = \frac{W3\hat{T}OT}{\hat{P}_{c,HH}}, \quad c \in \text{COM}.$$

Por fim, definimos o dispêndio real das famílias ($X3TOT$) como:

$$X3\hat{T}OT = \frac{W3\hat{T}OT}{P3\hat{T}OT},$$

em que $P3TOT$ é o índice de preços associado à cesta de consumo das famílias. Para uma função do tipo cobb-douglas, o índice $P3TOT$ pode ser definido como:

$$P3TOT = \prod_{c \in COM} \left(\frac{P_{c,HH}^s}{\alpha_c} \right)^{\alpha_c}.$$

Em variação, esse índice é reescrito como:

$$P3\hat{T}OT = \prod_{c \in COM} \left(\hat{P}_{c,HH}^s \right)^{\alpha_c},$$

em que α_c é a participação do bem c no dispêndio das famílias.

Especificado a estrutura das famílias no modelo, vamos definir, primeiramente, o parâmetro $SHARE_{c,HH}$.

```
# calcula os shares
SHARE_HH <- use_df %>%
  filter(USER == "Households") %>%
  group_by(COM) %>%
  summarise(Value = sum(Value)) %>%
  mutate(SHARE = Value/sum(Value)) %>%
  select(COM, SHARE)

params[["SHARE_HH"]] <- create_param(
  value = SHARE_HH,
  indexes = sets['COM'],
  desc = "Participação do bem c no dispêndio das famílias"
)
```

No fechamento do modelo que será adotado, o dispêndio real das famílias ($X3\hat{T}OT$) será exógeno. Portanto, o definiremos como um parâmetro do modelo.

```
params[["X3TOT"]] <- create_param(
  value = 1,
  indexes = "X3TOT",
  desc = "Variação no dispêndio real das famílias"
)
```

Com essa definição, temos que:

$$W3\hat{T}OT = X3\hat{T}OT \times P3\hat{T}OT.$$

Isto é, a variação da renda das famílias tem que ser igual a variação do dispêndio real vezes a variação dos preços para as famílias.

Lembrando que a variável $\hat{X}_{c,HH}$ já está incluída na variável x_s , vamos definir as demais variáveis desse bloco:

```
variables[["p3tot"]] <- create_variable(
  value = 1,
  indexes = "p3tot",
  type = "defined",
  desc = "Variação no índice de preços das famílias"
)

variables[["w3tot"]] <- create_variable(
  value = 1,
  indexes = "w3tot",
  type = "defined",
  desc = "Variação da renda nominal das famílias"
)
```

Finalmente, vamos definir as equações:

```
equations[["E_p3tot"]] <- create_equation(
  'p3tot = prod(p_s["Households"]^SHARE_HH['])',
  type = "defining",
  desc = "Variação do índice de preços das famílias"
)

equations[["E_w3tot"]] <- create_equation(
  'w3tot = X3TOT * p3tot',
  type = "defining",
  desc = "Variação na renda (dispêndio) nominal das família"
)

equations[["E_x_s_hh"]] <- create_equation(
  'x_s[c, "Households"] = w3tot/p_s[c, "Households"]',
  indexes = c('c in COM'),
  type = "defining",
  desc = "Variação do uso do composto c pelas famílias"
)
```

3.2.6 Investimento e Governo

No MINIMAL, não é assumida nenhuma estrutura específica para o dispêndio em investimento ou do governo. Será assumido, que essas duas fontes de demandas são exógenas. Ou seja, $\hat{X}_{c,INV}^s = 1$ e $\hat{X}_{c,GOV}^s = 1$. Portanto, iremos apenas defini-los como parâmetros, que poderão ser utilizados posteriormente como fontes de choques do modelo.

```
params[["X_S_INV"]] <- create_param(
  value = 1,
  indexes = sets["COM"],
  desc = "Variação na demanda de investimento por produto c"
)

params[["X_S_GOV"]] <- create_param(
  value = 1,
  indexes = sets["COM"],
  desc = "Variação na demanda do governo por produto c"
)
```

Também definimos as equações que capturarão esses choques.

```
equations[["E_x_s_inv"]] <- create_equation(
  'x_s[c, "Investment"] = X_S_INV[c]',
  indexes = 'c in COM',
  type = "defining",
  desc = "Variação no uso do composto c para investimento"
)

equations[["E_x_s_gov"]] <- create_equation(
  'x_s[c, "Government"] = X_S_GOV[c]',
  indexes = 'c in COM',
  type = "defining",
  desc = "Variação no uso do composto c pelo governo"
)
```

3.2.7 Demanda de segundo nível entre bens domésticos e importados

Até o momento, apresentamos a demanda⁹ das indústrias, das famílias, do governo e do investimento pelos bens compostos. Nessa parte, vamos definir a demanda do nível inferior. Nesse nível, o consumidor escolhe alocar o seu consumo total entre o produto doméstico e o produto importado.

É assumida uma função de agregação CES, com elasticidade de substituição σ_i , que combina os produtos domésticos e importados. Nesse caso, a variação na demanda por cada produto, por fonte e por usuário, \hat{X}_{csu} , é dada pela seguinte função de demanda:

$$\hat{X}_{csu} = \left(\frac{\hat{P}_{cs}}{\hat{P}_{cu}^s} \right)^{-\sigma_c} \hat{X}_{cu}^s, \quad c \in \text{COM}, s \in \text{SRC}, s \in \text{IMPUSER} \quad (4)$$

⁹O componente de exportação demanda apenas o bem doméstico.

em que \hat{P}_{cs} é a variação do preço do produto c fornecido pela fonte s . Já \hat{P}_{cu}^s é o índice de preço do bem composto c para o usuário u .

Abaixo, define-se a variável \hat{X}_{csu} e a equação para os usuários pertencentes ao conjunto IMPUSER.

```
variables[["x"]] <- create_variable(
  value = 1,
  indexes = sets[c("COM", "SRC", "USER")],
  type = "defined",
  desc = "Variação na demanda por produto, fonte e usuário"
)

equations[["E_x_impuser"]] <- create_equation(
  'x[c,s,u] = x_s[c,u]*(p[c,s]/p_s[c,u])^(-SIGMA[c])',
  indexes = c('c in COM', 's in SRC', 'u in IMPUSER'),
  type = "defining",
  desc = "Variação na demanda por produto, fonte e usuário"
)
```

Perceba que a variável foi definida para todos os usuários. No entanto, na Equação 4, está definida equação para $u \in \text{IMPUSER}$. Isto deve-se ao fato de que a demanda para o usuário *Exports* será definida de outra forma.

3.2.8 Exportações

Para as exportações, é assumida uma função de elasticidade constante com parâmetro EXP_ELAST_c para o produto c . A demanda externa pelo produto doméstico depende do preço relativo entre o preço doméstico e o preço internacional daquele produto:

$$\hat{X}_{csu} = F\hat{4}Q_c \left(\frac{\hat{P}_{cs}}{\hat{\phi} \text{PWORLD}_c} \right)^{-\text{EXP_ELAST}_c}, \quad c \in \text{COM}, s = \text{dom}, u = \text{Exports}$$

em que $F\hat{4}Q_c$ é um *shift* na demanda externa.

Abaixo, definimos este bloco.

```
params[["EXP_ELAST"]] <- create_param(
  value = p018_df,
  indexes = sets["COM"],
  desc = "Elasticidade da demanda por exportações"
)

params[["F4Q"]] <- create_param(
  value = 1,
  indexes = sets[c("COM")],
  desc = "Shift na demanda externa para o produto c"
)
```

```

equations[["E_x_exp"]] <- create_equation(
  'x[c,"dom","Exports"] = F4Q[c]*(p[c,"dom"]/(PHI*PWORLD[c]))^(-EXP_ELAST[c])',
  indexes = 'c in COM',
  type = "defining",
  desc = "Variação das exportações do produto c"
)

```

3.2.9 Demanda por Fatores Primários

Na parte da produção, vimos como cada indústria define a quantidade de fator primário que será utilizada para atingir uma determinada produção. Todavia, cada indústria pode escolher um mix diferente entre os fatores de produção capital e trabalho. Para essa alocação, também é utilizada uma função de agregação CES, com elasticidade substituição σ_i^{1PRIM} . Assim, pode-se definir a demanda (em variações) por trabalho e capital na indústria i como:

$$X1\hat{LAB}_i = \left(\frac{P1\hat{LAB}}{P1\hat{PRIM}_i} \right)^{-\sigma_i^{1PRIM}} X1\hat{PRIM}_i, \quad i \in IND \quad e$$

$$X1\hat{CAP}_i = \left(\frac{P1\hat{CAP}_i}{P1\hat{PRIM}_i} \right)^{-\sigma_i^{1PRIM}} X1\hat{PRIM}_i, \quad i \in IND,$$

em que $X1\hat{LAB}_i$ é a variação da demanda por trabalho pela indústria i , $P1\hat{LAB}$ é o salário nominal, $P1\hat{PRIM}_i$ é a variação do índice de preços dos fatores primários para a indústria i , $X1\hat{PRIM}_i$ é a variação da demanda por fatores primários da indústria i , $X1\hat{CAP}_i$ é a variação da demanda por capital pela indústria i e $P1\hat{CAP}_i$ é a remuneração do capital na indústria i .

Considerando que os fatores primários são agregados a partir de uma função do tipo CES, a variável $P1\hat{PRIM}_i$ é calculada da seguinte forma:

$$P1\hat{PRIM}_i = \left[SHAREPRIM_{lab,i} \times P1\hat{LAB}^{1-\sigma_i^{1prim}} + SHAREPRIM_{cap,i} \times P1\hat{CAP}_i^{1-\sigma_i^{1prim}} \right]^{\frac{1}{1-\sigma_i^{1prim}}}$$

Note que o índice da i foi retirado de $P1\hat{LAB}$ (salário), pois no fechamento adotado assume-se que o trabalho tem perfeita mobilidade entre os setores. Diferentemente, o capital será assumido fixo dentro de cada indústria. Ademais, consideraremos que no curto prazo a variação do salário real ($R\hat{W}$) é fixa (exógena) e a variação do nível de emprego (\hat{L}) é endógena, o que implica que $P1\hat{LAB}$ deve variar na mesma proporção de $P3\hat{TOT}$. Então, temos a seguinte equação:

$$P1\hat{LAB} = R\hat{W} \times P3\hat{TOT}$$

Primeiro, definimos os parâmetros desse bloco.

```

params[["SIGMA1PRIM"]] <- create_param(
  value = p028_df,
  indexes = sets["IND"],
  desc = "Elasticidade de substituição entre os fatores de produção"
)

SHAREPRIM <- fac_df %>%
  group_by(IND) %>%
  mutate(SHAREPRIM = Value/sum(Value)) %>%
  select(FAC, IND, SHAREPRIM)

params[["SHAREPRIM"]] <- create_param(
  value = SHAREPRIM,
  indexes = sets[c("FAC", "IND")],
  desc = "Part. de cada fator no uso do fator primário por indústria"
)

params[["RW"]] <- create_param(
  value = 1,
  indexes = "rw",
  desc = 'Variação no salário real'
)

```

Na sequência, definimos as variáveis.

```

variables[["x1lab"]] <- create_variable(
  value = 1,
  indexes = sets[c('IND')],
  type = "defined",
  desc = "Variação no emprego por indústria"
)

variables[["x1cap"]] <- create_variable(
  value = 1,
  indexes = sets[c('IND')],
  type = "defined",
  desc = "Variação no uso de capital por indústria"
)

variables[["p1lab"]] <- create_variable(
  value = 1,
  indexes = "p1lab",
  type = "defined",
  desc = "Variação no salário nominal"
)

variables[["p1prim"]] <- create_variable(
  value = 1,
  indexes = sets['IND'],
  type = "defined",
  desc = "Variação no índice de preço do fator primário composto por indústria i"
)

```

Por fim, vamos definir as equações.

```
equations[["E_x1lab"]] <- create_equation(
  'x1lab[i] = x1prim[i]*(p1lab/p1prim[i])^(-SIGMA1PRIM[i])',
  indexes = c('i in IND'),
  type = "defining",
  desc = "Variação no emprego por indústria"
)

equations[["E_p1lab"]] <- create_equation(
  'p1lab = RW * p3tot',
  type = "defining",
  desc = "Variação no salário nominal"
)

equations[["E_x1cap"]] <- create_equation(
  'x1cap[i] = (p1cap[i]/p1prim[i])^(-SIGMA1PRIM[i]) * x1prim[i]',
  indexes = "i in IND",
  type = "defining",
  desc = "Variação na demanda por capital na indústria i"
)

equations[["E_p1prim"]] <- create_equation(
  'p1prim[i] = (SHAREPRIM["Labour", i] * p1lab^(1 - SIGMA1PRIM[i]) +
    SHAREPRIM["Capital", i] * p1cap[i]^(1 - SIGMA1PRIM[i]))^(1/(1-SIGMA1PRIM[i]))',
  indexes = c('i in IND'),
  type = "defining",
  desc = "Variação no índice de preço do fator primário para indústria i"
)
```

3.2.10 Equilíbrios nos Mercados de Bens

Para começar a especificar as condições de equilíbrio nos mercados de bens, precisamos definir a demanda total por cada produto c para cada fonte s , \hat{X}_{cs}^0 :

$$\hat{X}_{cs}^0 = \sum_{u \in \text{USER}} \text{SHRSALES}_{csu} \hat{X}_{csu}, \quad c \in \text{COM}, s \in \text{SRC}$$

em que

$$\text{SHRSALES}_{csu} = \frac{\text{USE}_{csu}}{\sum_u \text{USE}_{csu}}, \quad c \in \text{COM}, s \in \text{SRC}, u \in \text{USER}.$$

USE_{csu} é o valor da demanda de produto c da fonte s pelo usuário u .

Adicionalmente, pode-se definir a variação no custo de produção como:

$$\hat{\text{COST}}_i = \frac{\sum_c \text{USE}_{ci}^s \times \hat{P}_{ci}^s + \text{FAC}_{labour,i} \times \text{P1LAB}_i + \text{FAC}_{capital,i} \times \text{P1CAP}_i}{\text{V1TOT}_i},$$

em que USE_{ci}^s é o consumo total do produto c pela indústria i , $FAC_{labour,i}$ é a remuneração total do fator trabalho utilizado na indústria i , $FAC_{capital,i}$ é a remuneração total do fator capital utilizado na indústria i e $V1TOT_i$ é o custo total de produção da indústria i nos dados base.

O código abaixo cria os dados e define os parâmetros SHRSALES_{csu} , $V1TOT_i$, $FAC_{labour,i}$ e $FAC_{capital,i}$.

```
SHRSALES <- use_df %>%
  group_by(COM, SRC) %>%
  mutate(SALES = sum(Value),
         SHRSALES = Value/SALES) %>%
  select(COM, SRC, USER, SHRSALES)

use_s_df <- use_df %>%
  group_by(COM, USER) %>%
  summarise(Value = sum(Value))

# Salários + Remuneração do Capital
V1PRIM <- fac_df %>%
  group_by(IND) %>%
  summarise(Value = sum(Value))

# Custos Intermediários
IC <- use_df %>%
  filter(USER %in% IND) %>%
  group_by(IND = USER) %>%
  summarise(ic = sum(Value))

# Custos totais
V1TOT <- V1PRIM %>%
  rename(va = Value) %>%
  left_join(IC, by = "IND") %>%
  mutate(Value = va + ic) %>%
  select(IND, Value)

params[["USE"]] <- create_param(
  value = use_df,
  indexes = sets[c("COM", "SRC", "USER")],
  desc = "Uso do produto c da fonte s pelo usuário u"
)

params[["USE_S"]] <- create_param(
  value = use_s_df,
  indexes = sets[c("COM", "USER")],
  desc = "Uso do produto c da fonte s pelo usuário u"
)

params[["FAC0"]] <- create_param(
  value = fac_df,
  indexes = sets[c("FAC", "IND")],
  desc = "Uso inicial dos fatores primários pela indústria i"
)
```

```

params[["V1TOT"]] <- create_param(
  value = V1TOT,
  indexes = sets[c("IND")],
  desc = "Custo total da indústria i"
)

params[["SHRSALES"]] <- create_param(
  value = SHRSALES,
  indexes = sets[c("COM", "SRC", "USER")],
  desc = "Participação do usuário u nas vendas de c de origem s"
)

```

Com os parâmetros definidos, iremos adicionar as variáveis e as equações.

```

variables[["x0"]] <- create_variable(
  value = 1,
  indexes = sets[c("COM", "SRC")],
  type = "defined",
  desc = "Variação na demanda total de c de origem s"
)

variables[["cost"]] <- create_variable(
  value = 1,
  indexes = sets[c("IND")],
  type = "defined",
  desc = "Variação no custo marginal de produção "
)

```

```

equations[["E_x0"]] <- create_equation(
  "x0[c,s] = sum(SHRSALES[c,s] * x[c,s])",
  indexes = c("c in COM", "s in SRC"),
  type = "defining",
  desc = "Variação na demanda total de c de origem s"
)

equations[["E_cost"]] <- create_equation(
  "cost[i] = (sum(USE_S[,i] * p_s[,i]) +
              sum(FACO[,i]) * p1prim[i]) / (V1TOT[i])",
  indexes = c("i in IND"),
  type = "defining",
  desc = "Variação no custo marginal de produção da indústria i"
)

```

Agora, vamos definir as equações para duas variáveis do tipo mcc: $P1\hat{TOT}_i$ e $X1\hat{TOT}_i$:

- Condição de lucro zero para a indústria i :

$$P1\hat{TOT}_i = C\hat{OST}_i, \quad i \in IND$$

- Oferta igual a demanda para a indústria i^{10} :

$$X1\hat{TOT}_i = \hat{X}_{i\text{ dom}}^0, \quad i \in \text{IND}$$

```
variables[["p1tot"]] <- create_variable(
  value = 1,
  indexes = sets[c("IND")],
  type = "undefined",
  desc = "Variação no preço da indústria i"
)

variables[["x1tot"]] <- create_variable(
  value = 1,
  indexes = sets[c("IND")],
  type = "undefined",
  desc = "Variação na produção da indústria i"
)

equations[["E_p1tot"]] <- create_equation(
  "p1tot[i] - cost[i]",
  indexes = c("i in IND"),
  type = "mcc",
  desc = "Condição de lucro zero"
)

equations[["E_x1tot"]] <- create_equation(
  "-x1tot[i] + x0[i,'dom']",
  indexes = c("i in IND"),
  type = "mcc",
  desc = "Equilíbrio no mercado i"
)
```

3.2.11 Equilíbrio nos Mercados de Fatores de Produção

No mercado de fatores, temos equilíbrios de para dois mercados: trabalho e capital. No entanto, a suposição escolhida sobre a mobilidade dos fatores irá determinar quantos preços de fato existirão. Para o fator trabalho, assume-se que a mão-de-obra não é específica de cada setor, o que resulta em um único mercado de trabalho com preço P1LAB. No entanto, para utilização de capital será assumido que o fator é específico da indústria e o total usado por cada indústria (X1CAP) é determinado exogenamente. Dessa forma, cada indústria terá um preço (remuneração) de capital que faz com que a variação no uso de capital seja igual àquela determinada exogenamente.

Adicionalmente, no exemplo apresentado, consideramos um fechamento de curto prazo no qual o salário real é fixo e o nível total de emprego é determinado endogenamente. Assim, no equilíbrio espera-se que

$$\hat{L} = \sum_{i \in \text{IND}} \text{SHARE LAB}_i \times X1\hat{\text{LAB}}_i,$$

¹⁰Lembre que o conjunto de produtos é o mesmo conjunto de indústrias. Dessa forma, utilizaremos o índice i para a variável \hat{X}_{cs}^0 .

em que \hat{L} é a variação no emprego total e o lado direito da equação mensura a variação na quantidade total demandada por trabalho de cada indústria ponderado pela participação inicial de cada indústria no uso do trabalho total (SHARE LAB_i).

```
SHARELAB <- fac_df %>%
  filter(FAC == "Labour") %>%
  mutate(SHARELAB = Value/sum(Value)) %>%
  select(IND, SHARELAB)

params[["SHARELAB"]] <- create_param(
  value = SHARELAB,
  indexes = sets["IND"],
  desc = "Part. de cada indústria no uso do fator trabalho"
)
```

```
variables[["l"]] <- create_variable(
  value = 1,
  indexes = "emprego",
  type = "undefined",
  desc = "Variação no emprego total"
)
```

```
equations[["E_1"]] <- create_equation(
  'l - sum(SHARELAB * x1lab)',
  type = "mcc",
  desc = "Variação no emprego total"
)
```

Para o mercado do fator capital, assume-se que ele é fixo na indústria. Assim, temos que a variação na demanda tem que ser igual a uma constante que tem valor inicial 1. Essa constante poderá ser usada para simular cenários de aumento do estoque de capital em alguma indústria. Considerando isso, em equilíbrio, a variação do preço do capital na indústria i , $\hat{P1CAP}_i$, tem que satisfazer a seguinte equação de equilíbrio:

$$\hat{X1CAP}_i = \hat{X1CAP}_i^{\text{EXO}}$$

```
# O capital é fixo na indústria (exógeno)
params[["X1CAP_EXO"]] <- create_param(
  value = 1,
  indexes = sets["IND"],
  desc = "Variação exógena no uso de capital por indústria"
)
```

```
variables[["p1cap"]] <- create_variable(
  value = 1,
  indexes = sets["IND"],
  type = "undefined",
  desc = "Variação na remuneração do capital por indústria i"
)
```



```
equations[["E_p1cap"]] <- create_equation(
  'X1CAP_EX0[i] - x1cap[i]',
  indexes = 'i in IND',
  type = "mcc",
  desc = "equilíbrio no mercado de capital para indústria i"
)
```

3.2.12 Variáveis Macroeconômicas

A primeira variável macroeconômica que vamos definir é o PIB (*GDP*) pela ótica da renda. Por essa ótica, o PIB é igual ao total das remunerações dos fatores mais a receita de impostos indiretos. O valor no cenário base pode ser facilmente calculado como:

```
v0gdpinc <- sum(fac_df$Value) +
  sum(tar_df$Value) +
  sum(ptx_df$Value)
```

No novo equilíbrio, a variação nominal do PIB pode ser computada como:

$$\begin{aligned} V0GDPINC \times WGD\hat{P}INC = & \sum_{i \in IND} FAC_{labour,i} \times P1\hat{L}AB \times X1\hat{L}AB_i + \\ & \sum_{i \in IND} FAC_{capital,i} \times P1\hat{C}AP_i \times X1\hat{C}AP_i + \\ & \sum_{c \in COM} V0CIF_c \times PW\hat{O}RLD_c \times \hat{\phi} \times ((TAR0_c + 1) \times M\hat{T}X_c - 1) \\ & \sum_{i \in IND} V1TOT_i \times P1\hat{T}OT_i \times \hat{\phi} \times ((PTX0_i + 1) \times P\hat{T}X_i - 1) \end{aligned}$$

O código abaixo cria os parâmetros necessários, a variável `wgdpinc` e a equação que calcula a variação do PIB pela ótica do dispêndio.

```
v0cif_tar <- use_df %>%
  filter(SRC == "imp") %>%
  group_by(COM) %>%
  summarise(sales = sum(Value)) %>%
  left_join(tar_df) %>%
  mutate(v0cif = sales - Value,
         tar = Value/v0cif)
```

```
## Joining, by = "COM"
```

```
v0cif <- v0cif_tar %>%
  select(COM, v0cif)

tarifas <- v0cif_tar %>%
```

```

select(COM, tar)

ptx0 <- left_join(V1TOT, ptx_df, by = "IND",
                  suffix = c("_V1TOT", "_PTX")) %>%
  mutate(ptx0 = Value_PTX/Value_V1TOT) %>%
  select(IND, ptx0)

params[["VOGDPINC"]] <- create_param(
  value = v0gdpinc,
  indexes = "v0gdpinc",
  desc = "Valor inicial do PIB"
)

params[["VOCIF"]] <- create_param(
  value = v0cif,
  indexes = sets["COM"],
  desc = "Valor CIF inicial das importações"
)

params[["MTX0"]] <- create_param(
  value = tarifas,
  indexes = sets["COM"],
  desc = "Tarifas Iniciais"
)

params[["PTX0"]] <- create_param(
  value = ptx0,
  indexes = sets["IND"],
  desc = "Taxa inicial de imposto sobre a produção"
)

variables[["wgdppinc"]] <- create_variable(
  value = 1,
  indexes = "wgdppinc",
  type = "defined",
  desc = "Variação no PIB nominal"
)

equations[["E_wgdppinc"]] <- create_equation(
  "wgdppinc = (sum(FACO['Labour'],] * p1lab * x1lab) +
  sum(FACO['Capital'],] * p1cap * x1cap) +
  sum(VOCIF * x0[, 'imp'] * PWORLD * PHI * ((1 + MTX0) * MTX - 1)) +
  sum(V1TOT * p1tot * x1tot * ((1 + PTX0) * PTX - 1)))/VOGDPINC",
  type = "defining",
  desc = "Variação no PIB nominal"
)

```

A segunda variável macro que iremos calcular é a variação nominal do PIB pela ótica do dispêndio, que pode ser calculada como:

$$\text{VOGDPEXP} \times \text{WGDPEXP} = \sum_{c \in \text{COM}} \sum_{s \in \text{SRC}} \sum_{u \in \text{FINALUSER}} \text{USE}_{csu} \times \hat{P}_{cs} \times \hat{X}_{csu} - \text{VOCIF}_c \times \hat{X}_{c,\text{imp}}^0 \times \text{PWORLD}_c \times \hat{\phi}$$

Primeiro, iremos calcular o valor do PIB pela ótica do dispêndio nos dados base e criar o parâmetro VOGDPEXP.

```
total_dispendio <- use_df %>%
  filter(USER %in% FINALUSER) %>%
  pull(Value) %>%
  sum()

total_importacoes <- sum(v0cif$v0cif)

v0gdpepx <- total_dispendio - total_importacoes

params[["VOGDPEXP"]] <- create_param(
  value = v0gdpepx,
  indexes = "v0gdpepx",
  desc = "Valor inicial do PIB pela ótica do dispêndio"
)

variables[["wgdpepx"]] <- create_variable(
  value = 1,
  indexes = "wgdpepx",
  type = "defined",
  desc = "Variação no PIB nominal pela ótica do dispêndio"
)

# equations[["E_wgdpepx"]] <- create_equation(
#   "wgdpepx = (sum(USE_CS) - sum(VOCIF * x0[, 'imp'] * PWORLD * PHI))/VOGDPEXP",
#   type = "defining",
#   desc = "Variação no PIB nominal pela ótica no dispêndio"
# )

equations[["E_wgdpepx"]] <- create_equation(
  "wgdpepx = (@sum_emr(sum_emr(sum_emr('USE[c,s,u] * p[c,s] * x[c,s,u]',
    'u', 'FINALUSER'), 's', 'SRC'), 'c', 'COM'))@
  - sum(VOCIF * x0[, 'imp'] * PWORLD * PHI))/VOGDPEXP",
  type = "defining",
  desc = "Variação no PIB nominal pela ótica no dispêndio"
)
```

Por sua vez, o deflator do PIB, PGDPEXP, é calculado a partir de um índice de preços de Fisher, que é a média geométrica dos índices de Paasche e Laspeyres. Isto é necessário para poder replicar a solução do GEMPACK. Os índices são calculados da seguinte forma:

- Índice de Preços de Laspeyres:

$$\text{PGDPEXP}^L = \frac{\left(\sum_{c \in \text{COM}} \sum_{s \in \text{SRC}} \sum_{u \in \text{FINALUSER}} \text{USE}_{csu} \times \hat{P}_{cs} - \sum_c \text{VOCIF}_c \times \text{PWORLD}_c \times \hat{\phi} \right)}{\left(\sum_{c \in \text{COM}} \sum_{s \in \text{SRC}} \sum_{u \in \text{FINALUSER}} \text{USE}_{csu} - \sum_c \text{VOCIF}_c \right)}$$

- Índice de Preços de Paasche¹¹:

$$\text{PGD}\hat{\text{PEXP}}^L = \frac{(\sum_c \sum_s \sum_u \text{USE}_{csu} \times \hat{P}_{cs} \times \hat{X}_{csu} - \sum_c \text{VOCIF}_c \times \text{PWORLD}_c \times \hat{\phi} \times \hat{X}_{c,\text{imp}}^0)}{(\sum_c \sum_s \sum_u \text{USE}_{csu} \times \hat{X}_{csu} - \sum_c \text{VOCIF}_c \times \hat{X}_{c,\text{imp}}^0)}$$

- Índice de Preços de Fisher:

$$\text{PGD}\hat{\text{PEXP}} = \sqrt{\text{PGD}\hat{\text{PEXP}}^L \times \text{PGD}\hat{\text{PEXP}}^P}$$

```
variables[["pgdpexp_l"]] <- create_variable(
  value = 1,
  indexes = "pgdpexp_l",
  type = "defined",
  desc = "Variação no deflator do PIB (Laspeyres)"
)

variables[["pgdpexp_p"]] <- create_variable(
  value = 1,
  indexes = "pgdpexp_p",
  type = "defined",
  desc = "Variação no deflator do PIB (Paasche)"
)

variables[["pgdpexp"]] <- create_variable(
  value = 1,
  indexes = "pgdpexp",
  type = "defined",
  desc = "Variação no deflator do PIB"
)

equations[["E_pgdpexp_l"]] <- create_equation(
  "pgdpexp_l = (@sum_emr(sum_emr(sum_emr('USE[c,s,u] * p[c,s]', 'u', 'FINALUSER'), 's', 'SRC'),
    'c', 'COM')@ - sum(VOCIF * PWORLD * PHI))/(sum(USE[,FINALUSER]) - sum(VOCIF))",
  type = "defining",
  desc = "Variação no deflator do PIB (Laspeyres)"
)

equations[["E_pgdpexp_p"]] <- create_equation(
  "pgdpexp_p = (@sum_emr(sum_emr(sum_emr('USE[c,s,u] * p[c,s] * x[c,s,u]',
    'u', 'FINALUSER'), 's', 'SRC'), 'c', 'COM')@ - sum(VOCIF * PWORLD * PHI * x0['imp']))/
    (sum(USE[,FINALUSER] * x[,FINALUSER]) - sum(VOCIF * x0['imp']))",
  type = "defining",
  desc = "Variação no deflator do PIB (Paasche)"
)

equations[["E_pgdpexp"]] <- create_equation(
  "pgdpexp = sqrt(pgdpexp_l * pgdpexp_p)",
  type = "defining",
  desc = "Variação no deflator do PIB"
)
```

¹¹Os conjuntos de cada índice são os mesmos que foram utilizados para calcular o índice de Laspeyres.

Calculados a variação nominal do PIB e o deflator, pode-se calcular a variação real do PIB de maneira simples:

$$XGD\hat{P}EXP = \frac{WGD\hat{P}EXP}{PGD\hat{P}EXP}$$

```
variables[["xgdpepx"]] <- create_variable(
  value = 1,
  indexes = "xgdpepx",
  type = "defined",
  desc = "Variação no PIB real"
)

equations[["E_xgdpepx"]] <- create_equation(
  "xgdpepx = wgdpepx/pgdpepx",
  type = "defining",
  desc = "Variação no PIB real"
)
```

Também pode-se calcular o índice de preços do investimento, $P2TOT$, como:

$$P2\hat{T}OT = \sum_{c \in COM} \sum_{s \in SRC} \frac{USE_{csINV}}{\sum_{c \in COM} \sum_{s \in SRC} USE_{csINV}} \hat{P}_{cs}.$$

```
variables[["p2tot"]] <- create_variable(
  value = 1,
  indexes = "p2tot",
  type = "defined",
  desc = "Variação no índice de preço do investimento"
)

variables[["p2tot_l"]] <- create_variable(
  value = 1,
  indexes = "p2tot_l",
  type = "defined",
  desc = "Variação no índice de preço do investimento (Laspeyres)"
)

variables[["p2tot_p"]] <- create_variable(
  value = 1,
  indexes = "p2tot_p",
  type = "defined",
  desc = "Variação no índice de preço do investimento (Paasche)"
)

equations[["E_p2tot_l"]] <- create_equation(
  "p2tot_l = sum(USE_S[, 'Investment'] * p_s[, 'Investment'])/sum(USE_S[, 'Investment'])",
  type = "defining",
  desc = "Variação no índice de preço do investimento (Laspeyres)"
)
```

```

equations[["E_p2tot_p"]] <- create_equation(
  "p2tot_p = sum(USE_S[, 'Investment'] * p_s[, 'Investment'] * x_s[, 'Investment']) /
  sum(USE_S[, 'Investment'] * x_s[, 'Investment'])",
  type = "defining",
  desc = "Variação no índice de preço do investimento (Laspeyres)"
)

equations[["E_p2tot"]] <- create_equation(
  "p2tot = sqrt(p2tot_l * p2tot_p)",
  type = "defining",
  desc = "Variação no índice de preço do investimento"
)

```

Também pode-se computar os índices de quantidade (X4TOT) e preços (P4TOT) das exportações. Para calcular o P4TOT:

$$P4\hat{T}OT = \sum_{c \in COM} \frac{USE_{c \text{ dom EXP}}}{\sum_{c \in COM} USE_{c \text{ dom EXP}}} \hat{P}_{c \text{ dom}}.$$

```

variables[["p4tot"]] <- create_variable(
  value = 1,
  indexes = "p4tot",
  type = "defined",
  desc = "Variação no índice de preços da exportação"
)

variables[["p4tot_l"]] <- create_variable(
  value = 1,
  indexes = "p4tot_l",
  type = "defined",
  desc = "Variação no índice de preços da exportação (Laspeyres)"
)

variables[["p4tot_p"]] <- create_variable(
  value = 1,
  indexes = "p4tot_p",
  type = "defined",
  desc = "Variação no índice de preços da exportação (Paasche)"
)

equations[["E_p4tot_l"]] <- create_equation(
  "p4tot_l = sum(USE[, 'dom', 'Exports'] * p[, 'dom']) / sum(USE[, 'dom', 'Exports'])",
  type = "defining",
  desc = "Variação no índice de preços da exportação (Laspeyres)"
)

equations[["E_p4tot_p"]] <- create_equation(
  "p4tot_p = sum(USE[, 'dom', 'Exports'] * p[, 'dom'] * x[, 'dom', 'Exports']) /
  sum(USE[, 'dom', 'Exports'] * x[, 'dom', 'Exports'])",
  type = "defining",
  desc = "Variação no índice de preços da exportação (Paasche)"
)

```

```
equations[["E_p4tot"]] <- create_equation(
  "p4tot = sqrt(p4tot_l * p4tot_p)",
  type = "defining",
  desc = "Variação no índice de preços da exportação"
)
```

E o X4TOT:

$$X4\hat{T}OT = \sum_{c \in COM} \frac{USE_{c \text{ dom } EXP}}{\sum_{c \in COM} USE_{c \text{ dom } EXP}} \hat{X}_{c \text{ dom } EXP}.$$

```
variables[["x4tot"]] <- create_variable(
  value = 1,
  indexes = "x4tot",
  type = "defined",
  desc = "Variação no índice de volume da exportação"
)

equations[["E_x4tot"]] <- create_equation(
  "x4tot = sum(USE[, 'dom', 'Exports'] * p[, 'dom'] * x[, 'dom', 'Exports']) /
  (sum(USE[, 'dom', 'Exports']) * p4tot)",
  type = "defining",
  desc = "Variação no índice de volume de exportação"
)
```

```
variables[["p0cif"]] <- create_variable(
  value = 1,
  indexes = "p0cif",
  type = "defined",
  desc = "Variação no índice de preços da importação"
)

variables[["p0cif_l"]] <- create_variable(
  value = 1,
  indexes = "p0cif_l",
  type = "defined",
  desc = "Variação no índice de preços da importação (Laspeyres)"
)

variables[["p0cif_p"]] <- create_variable(
  value = 1,
  indexes = "p0cif_p",
  type = "defined",
  desc = "Variação no índice de preços da importação (Paasche)"
)

equations[["E_p0cif_l"]] <- create_equation(
  "p0cif_l = sum(VOCIF * PWORDL * PHI) / sum(VOCIF)",
  type = "defining",
  desc = "Variação no índice de preços da importação (Laspeyres)"
)
```

```

equations[["E_p0cif_p"]] <- create_equation(
  "p0cif_p = sum(VOCIF * PWORLD * PHI * x0[, 'imp'])/sum(VOCIF * x0[, 'imp'])",
  type = "defining",
  desc = "Variação no índice de preços da importação (Paasche)"
)

equations[["E_p0cif"]] <- create_equation(
  "p0cif = sqrt(p0cif_l * p0cif_p)",
  type = "defining",
  desc = "Variação no índice de preços da importação"
)

```

Vamos também definir a variação no índice de volume das importações (X0CIF):

$$X0\hat{C}IF = \sum_{c \in COM} \frac{V0CIF_c}{\sum_{c \in COM} V0CIF_c} \hat{X}^0_{c \text{ imp}}.$$

```

variables[["x0cif"]] <- create_variable(
  value = 1,
  indexes = "x0cif",
  type = "defined",
  desc = "Variação no índice de volume da importação"
)

equations[["E_x0cif"]] <- create_equation(
  "x0cif = sum(VOCIF * PWORLD * PHI * x0[, 'imp'])/(sum(VOCIF) * p0cif)",
  type = "defining",
  desc = "Variação no índice de preços da importação"
)

```

Para finalizar, calculamos a razão entre a balança comercial e o PIB.

```

variables[["delB"]] <- create_variable(
  value = 1,
  indexes = "delB",
  type = "defined",
  desc = "Balança comercial como razão do PIB"
)

equations[["E_delB"]] <- create_equation(
  "delB = sum(sum(USE[, 'dom', 'Exports']) * p4tot * x4tot -
    sum(VOCIF[]) * p0cif * x0cif)/(VOGDPEXP * wgdpepx) -
    sum(USE[, 'dom', 'Exports'] - VOCIF)/(VOGDPEXP) ",
  type = "defining",
  desc = "Variação no índice de preços da importação"
)

```


3.3 Atualização dos Dados

Após a nova solução ser encontrada é possível atualizar os dados iniciais utilizando as variações obtidas. Por exemplo, temos o parâmetro USE que traz os valores iniciais do uso do produto c originário da fonte s pelo usuário u . É possível que se deseje saber quais seriam esses novos valores no contrafactual. Para isso, podemos criar um conjunto de equações de atualizações que servirão para computar esses novos valores.

Para atualizar os valores USE, precisamos das variações de preços e quantidades. A equação de atualização seria a seguinte:

$$USE'_{csu} = USE_{csu} \times \hat{P}_{cs} \times \hat{X}_{csu},$$

em que USE'_{csu} representa os novos valores de USE_{csu} .

Para definir essas equações de atualizações, precisamos primeiro criar uma lista chamada `update_equations`. Cada equação de atualização, ficará responsável por atualizar os valores de uma parâmetro. Para isso, precisamos que **a equação de atualização possua o mesmo nome do objeto que queremos atualizar**.

```
update_equations <- list()

update_equations[["USE"]] <- create_equation(
  "USE[c,s,u] = USE[c,s,u] * p[c,s] * x[c,s,u]",
  indexes = c("c in COM", "s in SRC", "u in USER"),
  desc = "Atualização dos dados de uso dos produto"
)
```

Para deixar mais uma exemplo, vamos atualizar os valores das remunerações (uso) de fatores primários. Esse parâmetro pode ser atualizado utilizando a seguinte equação:

$$FAC0'_{fi} = \begin{cases} FAC0_{fi} \times p1\hat{lab} \times x1\hat{lab}_i & \text{se } f = \text{labour} \\ FAC0_{fi} \times p1\hat{cap}_i \times x1\hat{cap}_i & \text{se } f = \text{capital}, \end{cases}$$

E definimos essa atualização da seguinte forma:

```
update_equations[["FAC0"]] <- create_equation(
  "if(f == 'Labour'){
    FAC0[f,i] = FAC0[f,i] * p1lab * x1lab[i]
  } else{
    FAC0[f,i] = FAC0[f,i] * p1cap[i] * x1cap[i]
  }",
  indexes = c("f in FAC", "i in IND"),
  desc = "Atualização dos dados de uso de fatores primários"
)
```

3.4 Exemplos de Simulações

Com o modelo definido, podemos, finalmente, partir para as simulações. O primeiro passo é criar um objeto que guarde os componentes do modelo (conjuntos, parâmetros, variáveis, equações etc.). Para isso, iremos criar uma **lista** que armazenará esses componentes:

```
minimal <- list(
  sets = sets,
  params = params,
  variables = variables,
  equations = equations,
  update_equations = update_equations
)
```

Com o objeto criado, vamos checar se os dados iniciais já representam uma solução do modelo, ou seja, um equilíbrio. Para resolver o modelo podemos utilizar a função `solve_emr()` ou `solve_emr_block()`. A diferença básica entre essas duas funções é que a função `solve_emr_block()` utiliza um método mais flexível para resolver o modelo. Contudo, em alguns casos será necessário que o usuário forneça o *step size*¹² para atualização dos valores das variáveis dos tipos undefined.

```
sol0 <- solve_emr(
  model = minimal
)

sol0$sol$message
```

```
## [1] "Successful convergence"
```

3.4.1 Experimento 1: Aumento de demanda das famílias

Nosso modelo está corretamente especificado. Agora, vamos ao experimento. Suponha um choque de 10% na demanda das famílias (X3TOT)¹³. Como estamos trabalhando com a variação exata, significa definir $X3\hat{T}OT = 1,1$. Dessa forma, vamos alterar esse parâmetro do modelo:

```
# o colchete [] serve para que o vetor nomeado não seja sobrescrito
# por um vetor não nomeado.
minimal$params$X3TOT$value[] <- 1.1
```

¹²De forma resumida, a atualização de variável x é feita como

$$x' = x + \alpha \times d,$$

em que $d = -F(x)$ e $F(x)$ é o resíduo da equação mcc associada à variável x . Nesse caso, α seria o *step size*.

¹³Lembre-se que no fechamento de curto prazo adotado esse componente da demanda é exógeno

Para encontrar a nova solução, utilizamos novamente a função `solve_emr_block()` com o objeto `minimal` alterado. Nesse caso, no modelo não está mais equilíbrio (ver o resíduo para `x1tot` na iteração inicial):

```
sol1 <- solve_emr(minimal)

sol1$sol$message
```

```
## [1] "Successful convergence"
```

Com a nova solução, podemos verificar as variações percentuais resultantes da alteração do parâmetro `X3TOT`. Apesar de não estar diretamente relacionado com o pacote `emr`, fica explicitado abaixo o código para a criação da tabela de resultados:

```
map_df(c("l", "wgdpepx", "wgdppinc", "pgdpepx", "xgdpepx", "delB", "p3tot",
        "w3tot", "p1lab", "p2tot", "x4tot", "p4tot", "x0cif"), ~{
  data.frame(
    variavel = sol1$variables_descriptions %>%
      filter(variable == .x) %>%
      pull(description) %>%
      paste0(" (", .x, ")"),
    stringsAsFactors = FALSE
  ) %>%
  mutate(
    value = case_when(
      .x == "delB" ~ round(sol1$variables[[".x"]] * 100, 2),
      TRUE ~ round((sol1$variables[[".x"]] - 1) * 100, 2)
    )
  )
}) %>%
kable(
  booktabs = TRUE,
  col.names = c("Variável", "Variação %"),
  caption = "Resultados da Simulação - Experimento 1",
  linesep = "",
  format.args = list(big.mark = ".", decimal.mark = ",")
) %>%
kable_styling(latex_options = "hold_position")
```

Verifica-se um impacto de +0,95% no PIB real e +1,24% no emprego. O aumento da demanda interna é atendido via redução das exportações, que tiveram redução no volume (-19,72%) e aumento nos preços (+4,49%), e aumento das importações (+12,51%).

Qual é a sensibilidade dos resultados em relação à elasticidade de Armington? O código abaixo realiza a simulação diversas vezes, alterando o parâmetro σ_c entre 0.5 e 7. Verifica-se que o efeito no PIB é tão maior quanto menor for essa elasticidade de substituição. Os resultados estão apresentados na Figura 5.

Tabela 5. Resultados da Simulação - Experimento 1

Variável	Variação %
Variação no emprego total (l)	1,24
Variação no PIB nominal pela ótica do dispêndio (wgdpexp)	9,23
Variação no PIB nominal (wgdpinc)	9,23
Variação no deflator do PIB (pgdpexp)	8,20
Variação no PIB real (xgdpexp)	0,95
Balança comercial como razão do PIB (delB)	-4,03
Variação no índice de preços das famílias (p3tot)	7,69
Variação da renda nominal das famílias (w3tot)	18,46
Variação no salário nominal (p1lab)	7,69
Variação no índice de preço do investimento (p2tot)	5,71
Variação no índice de volume da exportação (x4tot)	-19,72
Variação no índice de preços da exportação (p4tot)	4,49
Variação no índice de volume da importação (x0cif)	12,51

```
map_df(runif(20, 0.5, 7), ~{

  minimal <- list(
    sets = sets,
    params = params,
    variables = variables,
    equations = equations,
    update_equations = update_equations
  )

  minimal$params$SIGMA$value[] <- .x

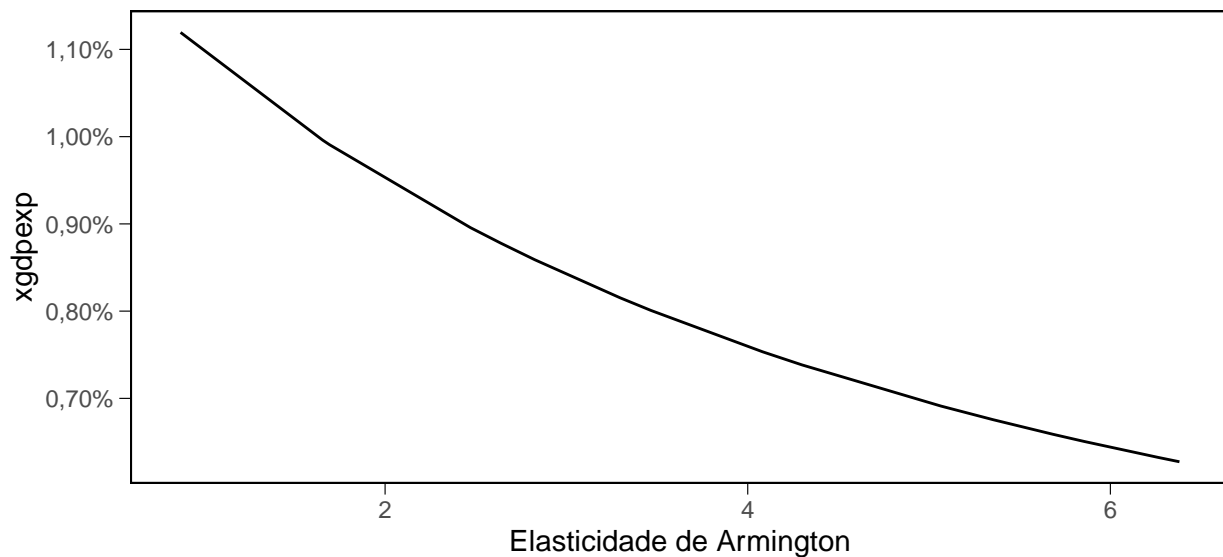
  minimal$params$X3TOT$value[] <- 1.1

  sol1 <- solve_emr(minimal)

  tibble(
    sigma = .x,
    xgdpexp = sol1$variables$xgdpexp - 1
  )
}) %>%
  ggplot(aes(x = sigma, y = xgdpexp)) +
  geom_line() +
  scale_y_continuous(labels = scales::percent_format(
    accuracy = .01,
    big.mark = ".",
    decimal.mark = ",",
  )) +
  labs(
    x = "Elasticidade de Armington"
  ) +
```

```
theme_minimal() +
theme(
  panel.border = element_rect(color = "black", fill = NA),
  panel.grid = element_blank(),
  axis.ticks = element_line(size = 0.2),
  axis.ticks.length=unit(.15, "cm")
)
```

Figura 5. Impacto % no PIB para um aumento exógeno em 10% na demanda das famílias (X3TOT)



3.4.2 Experimento 2: Aumento do salário real

No fechamento adotado, o salário real é definido exogenamente e o nível de emprego torna-se endógeno. Assim, nesse segundo experimento, verificamos quais seriam os impactos da elevação em 1% do salário real.

Em primeiro lugar, precisamos definir novamente o objeto `minimal`, uma vez que ele foi modificado no experimento anterior.

```
minimal <- list(
  sets = sets,
  params = params,
  variables = variables,
  equations = equations,
  update_equations = update_equations
)
```

O segundo passo é definir o choque:

```
minimal$params$RW$value[] <- 1.01
```

Finalmente, iremos resolver o modelo novamente:

```
sol1 <- solve_emr(minimal)
```

```
sol1$sol$message
```

```
## [1] "Successful convergence"
```

Tabela 6. Resultados da Simulação - Experimento 2

Variável	Variação %
Variação no emprego total (l)	-0,89
Variação no PIB nominal pela ótica do dispêndio (wgdpexp)	0,54
Variação no PIB nominal (wgdpinc)	0,54
Variação no deflator do PIB (pgdpexp)	1,15
Variação no PIB real (xgdpexp)	-0,61
Balança comercial como razão do PIB (delB)	-0,51
Variação no índice de preços das famílias (p3tot)	0,94
Variação da renda nominal das famílias (w3tot)	0,94
Variação no salário nominal (p1lab)	1,95
Variação no índice de preço do investimento (p2tot)	0,97
Variação no índice de volume da exportação (x4tot)	-3,06
Variação no índice de preços da exportação (p4tot)	0,62
Variação no índice de volume da importação (x0cif)	0,87

Para esse experimento, como esperado, ocorre uma redução no nível de emprego (-0,89%). Além disso, encontra-se um efeito negativo no PIB real (-0,61%), queda das exportações (-3,09%) e aumento das importações (+0,88%).

4. CONSIDERAÇÕES FINAIS

O objetivo desse documento foi apresentar a estrutura de para construção de modelos de equilíbrio parcial e geral no R, além de exemplificar com as simulações podem ser realizadas. Esses modelos podem ser bastante úteis para avaliações *ex ante* de políticas. Dessa forma, possibilitar a sua implementação em ambientes *open source* podem ajudar no ensino e disseminação desses modelos.

REFERÊNCIAS

- M.E. Burfisher. *Introduction to Computable General Equilibrium Models*. Cambridge University Press, 2017. ISBN 9781107132207. URL <https://books.google.com.br/books?id=mIHuDQAAQBAJ>.
- Robert Dekle, Jonathan Eaton, and Samuel Kortum. Global Rebalancing with Gravity: Measuring the Burden of Adjustment. *IMF Staff Papers*, 55(3):511–540, July 2008. URL <https://ideas.repec.org/a/pal/imfstp/v55y2008i3p511-540.html>.
- Mark Horridge. ORANI-G: A General Equilibrium Model of the Australian Economy. Centre of Policy Studies/IMPACT Centre Working Papers op-93, Victoria University, Centre of Policy Studies/IMPACT Centre, October 2000. URL <https://ideas.repec.org/p/cop/wpaper/op-93.html>.
- Mark Horridge and Alan Powell. Minimal – a simplified general equilibrium model. Technical report, Centre of Policies Studies and the Impact Project, 2001.
- Ravi Varadhan and Paul Gilbert. BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. *Journal of Statistical Software*, 32(4):1–26, 2009. URL <http://www.jstatsoft.org/v32/i04/>.
- Xiao-guang Zhang. A simple structure for cge models. Technical report, Australian Productivity Commission, 2013. URL <https://jgea.org/resources/download/6539.pdf>.