Modelo econômico de Leontief: Insumo-Produto MAP2110

Professor:	Nelson Mugayar Kuhl	NºUSP
Grupo:	Enzo Valentim Cappelozza	12556736
	Henrique Fujikawa Tokunaga	12675207
	Lucas Panfilo Donaire	12556552
	Marcos Martins Marchetti	11910868
	Ygor Peniche Maldonado	10271558

8 de junho de 2021

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução.

Denotaremos nesse presente estudo a produção total anual, por meio de uma equação linear, do $i - ésimo$ setor como:

$$\overline{x}_i = \overline{z}_{12} + \overline{z}_{22} + \ldots + \overline{z}_{in} + \overline{d}_i \tag{1}$$

Onde $\overline{z}_{ij}$ representa a venda intersetorial ($i = j$ incluso) e $\overline{d}_i$ a demanda final anual por produtos do setor i, do setor não produtivo. Tal modelo é chamado de *Economia Aberta*.

O modelo também pressupõe que para todo bem ou serviço de alguma classe i que seja produzido, necessariamente há um consumidor da classe j. Portanto, não há estoque ou desperdícios, ou seja, o modelo não considera prejuízos.

Em síntese, assumimos que todos os setores produtivos são interdependentes. Assim, para uma indústria (que pertence à um setor j) produzir uma quantidade x de produtos, em geral *demanda* consumir insumos de outros setores, inclusive de outras empresas do *mesmo setor* j a qual pertence.

Portanto, os setores produtivos são simultaneamente produtores e consumidores, apresentando uma *Demanda Intermediária*, enquanto que os setores não produtivos são exclusivamente consumidores, possuindo uma *Demanda do Setor Consumidor*.

Logo, nos modelos econômicos de Leontief abordados neste trabalho, um setor deve produzir uma quantidade x tal que essa quantidade seja o suficiente para suprir a *Demanda Intermediária* de todos os n setores (incluindo o seu próprio setor) ao longo de seus processos produtivos, paralelamente à *Demanda do Setor Consumidor*, de tal forma que:

$$\left\{\begin{matrix}\text{produção}\\ \text{x}\end{matrix}\right\} = \left\{\begin{matrix}\text{Demanda}\\ \text{Intermediária}\end{matrix}\right\} + \left\{\begin{matrix}\text{Demanda do Setor}\\ \text{Consumidor}\end{matrix}\right\}$$

Onde a produção é denotado por x, a "Demanda do Setor Consumidor"é denotada por $\overline{d}_i$, e a "Demanda Intermediária"é o produto matricial de $\overline{z}_{ij}$ pelo Vetor Produção x_i. Assim, podemos resumir os dados na seguinte tabela:

Tabela 1: Tabela Insumo-Produto

Setor	1	2	$\cdots$	n	**Demanda**	**Produção**
1	z_{11}	z_{12}	$\cdots$	z_{1n}	$\overline{d}_1$	$\overline{x}_1$
2	z_{21}	z_{22}	$\cdots$	z_{2n}	$\overline{d}_2$	$\overline{x}_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
n	z_{n1}	z_{n2}	$\cdots$	z_{nn}	$\overline{d}_n$	$\overline{x}_n$

Onde a coluna

$$z_j = \begin{bmatrix} z_{1j} \\ z_{2j} \\ \vdots \\ z_{nj} \end{bmatrix}$$

representa as compras realizadas pelo setor j dos outros setores produtivos, a fim de obter insumos necessários ao seu próprio processo produtivo. Dessa forma, a matriz acima nos dá não somente as fontes, como também as quantidades das compras realizadas pelo $j-\acute{e}simo$ setor.

Leontief parte do pressuposto que, para cada setor produtivo, *o valor dos insumos por unidade de valor produzido* é constante. Ou seja, ignora-se a variação dos preços de insumos para produção, inflação, custo de mão de obra, quebra de maquinário e conserto, variação salarial, entre outros.

De tal maneira, Leontief afirma que um modelo da forma matricial

$$x = D_{inter} + D_f \tag{2}$$

onde $x \in R^n$ e $D_f \in R^n$ é suficiente para prever a produção a fim de atender uma demanda final, essa que engloba todas as $i-\acute{e}simas$ demandas dos setores. Tendo em vista que noção de *Modelo Input-Output* se baseia, primeiro, nos pedidos feitos tanto pelo setor não produtivo quanto pelo produtivo , temos que a Matriz Produção x tem por finalidade prever e dar a resposta aos setores do quanto será necessário produzir, de cada item, para cada setor a fim de suprir a necessidade imposta por D_f.

Procura-se, então, uma maneira de representar o consumo de cada insumo, para cada setor, de uma maneira simplificada, a fim de sabermos qual é a importância (ou grandeza) do consumo de cada setor em relação ao total.

Note que a notação

$$z_{1j}/\overline{x}$$

denota o consumo do **Produto 1** pelo **Setor** j. De tal maneira, podemos dizer que o consumo total do setor j, em relação a tudo que é produzido, é dado pelo vetor de consumo

$$c_j = z_j/\overline{x}_j = \begin{bmatrix} z_{1j}/\overline{x}_j \\ z_{2j}/\overline{x}_j \\ \vdots \\ z_{nj}/\overline{x}_j \end{bmatrix}$$

Esse que, por sua vez, pode ser definido como uma das colunas c_j da **Matriz Consumo**. Onde cada c_j é uma coluna de C.

Definamos a Matriz Consumo como sendo

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_n \end{bmatrix}$$

De tal maneira obtemos:

$$C_{n\times n} = \begin{bmatrix} z_{11}/\overline{x}_1 & z_{12}/\overline{x}_2 & z_{13}/\overline{x}_3 & \dots & z_{1n}/\overline{x}_1 \\ z_{21}/\overline{x}_1 & z_{22}/\overline{x}_2 & z_{23}/\overline{x}_3 & \dots & z_{2n}/\overline{x}_2 \\ z_{31}/\overline{x}_1 & z_{32}/\overline{x}_2 & z_{33}/\overline{x}_3 & \dots & z_{3n}/\overline{x}_3 \\ z_{41}/\overline{x}_1 & z_{42}/\overline{x}_2 & z_{43}/\overline{x}_3 & \dots & z_{4n}/\overline{x}_4 \\ \vdots & \vdots & \dots & \ddots & \vdots \\ z_{n1}/\overline{x}_1 & z_2/\overline{x}_2 & z_{n3}/\overline{x}_3 & \dots & z_{nn}/\overline{x}_n \end{bmatrix} \tag{3}$$

Então de (2) e de (3) obtemos (4), onde $I_{n\times n}$ é a matriz identidade:

$$I - C = \begin{bmatrix} 1-c_{11} & -c_{12} & -c_{13} & \dots & -c_{1n} \\ -c_{21} & 1-c_{22} & -c_{23} & \dots & -c_{2n} \\ -c_{31} & -c_{32} & 1-c_{33} & \dots & -c_{3n} \\ -c_{41} & -c_{42} & -c_{43} & \dots & -c_{4n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_{n1} & -c_{52} & -c_{n3} & \dots & 1-c_{nn} \end{bmatrix} \tag{4}$$

Ou seja, a matriz $I - C$ denota a subtração entre a produção total do setor e o consumo pela demanda intermediária dos setores produtivos. Podemos inferir, portanto, que a Equação (4) representa as condições de consumo do setor intermediário, a fim de suprir as necessidades do setor não produtivo, denotado anteriormente por D_f. De tal maneira, temos que tudo que "sobra" do processo produtivo vai para a Demanda dos setores não produtivos, portanto:

$$D_f = (I - C)x$$

O futuro interesse será acerca da invertibilidade da matriz na Equação (4). Chamemos de **Matriz de Leontief** a matriz cuja forma é:

$$L = (I - C)^{-1} \tag{5}$$

Interpretemos agora os significados das Equações (4) e (5).
Pelas informações do problema descritos até agora, podemos afirmar:

- O custo de produção, e o rendimento em unidades de produto por unidade de valor é constante.
- Sabemos previamente o padrão de consumo do setor não produtivo.
- Há conhecimento prévio de tudo que o Setor Produtivo necessita para produção.

De tal maneira, devemos encontrar um método de isolar o *Vetor Demanda Final*, por meio da utilização de uma matriz inversa, a fim de que ele se torna a solução de um Sistema Linear.

No entanto, antes de qualquer manipulação prematura das equações, devemos analisar se (e quando) há de fato sentido em obter uma matriz inversa. Devemos, a partir de c_j obter alguma relação a fim de dar sentido à existência de uma possível matriz inversa. Note que pela definição dada acima da coluna da Matriz Consumo:

$$c_j = z_j / \overline{x}_j$$

Devemos, a fim de termos um regime necessariamente produtivo, ter que para qualquer j, o somatório das entradas da coluna c_j resulta, com efeito, em um valor numérico menor que a unidade. Ou seja:

$$\sum_{i=1}^{n} c_{ij} < 1 \tag{6}$$

Note que, caso a soma fosse igual à unidade, então teríamos pela própria definição de Consumo que o Setor Intermediário consumiria tudo aquilo que produz, isso implica dizer que a demanda nunca seria suprida. Caso a soma fosse igual a zero, teríamos que ou pelo menos um setor produziria a partir "do nada", não consumindo qualquer matéria prima, causando um "efeito dominó"nos outros setores, que passariam a não precisar produzir algum produto ou o setor não participaria economicamente, afetando, também, a produção de outros setores.

Teorema: 1 *Seja C uma matriz $n \times n$ com coeficientes reais não negativos e tal que (6) valha para todo $j = 1, 2, \ldots, n$. Então a matriz $I - C$ admite inversa, cujos coeficientes são todos não negativos.*

Sabemos então, pelo **Teorema 1** acima e pelas condições as quais a Matriz Consumo obedece, que de fato há uma matriz inversa coerente com o problema proposto. Ademais, como há a existência de uma matriz inversa, sabemos que as soluções serão únicas, visto que se trata de um Sistema Possível e Determinado.

Pois bem. Sabemos que de fato há uma única solução para $D_f = (I - C)x$. A pergunta mais natural que podemos fazer agora é: E daí? O que ganhamos com isso?

Como citado anteriormente através da interpretação das Equações (4) e (5), podemos, por meio da Matriz Inversa, isolar a Demanda. Isso é: O setor não produtivo pode meramente informar o quanto de produto será necessário, de cada uma das classes e, dessa maneira, através da matriz L, os setores poderão se coordenar a fim de projetar a produção necessária das j classes. Perceba:

$$D_f = (I - C)x \implies (I - C)^{-1} D_f = (I - C)^{-1}(I - C)x \implies$$

$$\implies (I - C)^{-1} D_f = Ix \implies (I - C)^{-1} D_f = x$$

Então:

$$L.D_f = x$$

Onde, agora, as soluções da equação matricial ditam qual deve ser a produção de cada setor.Em outras palavras: a Demanda ***(Input)*** do setor não produtivo dita qual deve ser a produção ***(Output)*** do setor produtivo. E, de tal maneira, pelos coeficientes da matriz L, cada setor saberá o quanto deve produzir.

2 Tarefas.

Tabela 2: Insumo-Produto

Setor	**Manufatura**	**Agrícola**	**Serviços**
Manufatura	0.10	0.60	0.60
Agrícola	0.30	0.20	0
Serviços	0.30	0.10	0.10

1. (a) De acordo com o enunciado, a matriz de consumo é:

$$C = \begin{bmatrix} 0.10 & 0.60 & 0.60 \\ 0.30 & 0.20 & 0 \\ 0.30 & 0.10 & 0.10 \end{bmatrix}$$

Para avaliar as demandas intermediárias, multiplicamos C_2 (vetor de consumo da agricultura) por 100, pois será o que a agricultura vai demandar, de cada setor, para produzir 100 unidades. Temos, portanto, que a demanda intermediária são 60 unidades da manufatura, 20 da própria agricultura, e 10 do setor de serviços.

(b) Para encontrar a matriz de Leontieff, invertemos a matriz

$$(I–C) = \begin{bmatrix} 0.90 & -0.60 & -0.60 \\ -0.30 & 0.80 & 0 \\ -0.30 & -0.10 & 0.90 \end{bmatrix}$$

Tal que $(I - C)^{-1} = L$

$$L = \begin{bmatrix} \frac{20}{9} & \frac{50}{27} & \frac{40}{27} \\ \frac{5}{6} & \frac{35}{18} & \frac{5}{9} \\ \frac{5}{6} & \frac{5}{6} & \frac{5}{3} \end{bmatrix}$$

Dado que os coeficientes da matriz L são o valor da produção de i para satisfazer 1 unidade da demanda final de j, é nítido que alguns padrões sobre a interdependência dos setores podem ser observados: todos valores na diagonal principal são maiores que um, pois um setor não produz só para sua própria demanda final (se um setor k fosse totalmente independente, L_{kk} teria valor 1), mas também para satisfazer as demandas intermediárias dos outros. Além disso, os valores da produção da manufatura (linha 1) são todos maiores que 1, pois os outros dois setores demandam muito dela (0,6 e 0,6 na matriz C). os coeficientes que não estão na diagonal principal e nem na primeira linha são todos menores de 1, pois são valores em que os setores não dependem tanto entre si (no máximo 0,3 na matriz C).

(c) Para determinar o nível de produção X, multiplicamos L por D
$L \times D = X$

$$L = \begin{bmatrix} \frac{20}{9} & \frac{50}{27} & \frac{40}{27} \\ \frac{5}{6} & \frac{35}{18} & \frac{5}{9} \\ \frac{5}{6} & \frac{5}{6} & \frac{5}{3} \end{bmatrix} \times \begin{bmatrix} 0 \\ 18 \\ 0 \end{bmatrix} = L = \begin{bmatrix} \frac{100}{3} \\ 35 \\ 15 \end{bmatrix}$$

Percebe-se mais claramente a interpretação dos coeficientes de Leontieff: como só a agricultura tem demanda final, a produção de cada setor foi $18 \times L_{i2}$ (Sendo L_{i2} o valor que i tem que produzir para 1 unidade de demanda final da agricultura).

2. Dado:
   (1) = Produtos domésticos e pessoais não metálicos
   (2) = Produtos metálicos finais (como veículos motorizados)
   (3) = Produtos metálicos básicos e mineração
   (4) = Produtos não metálicos básicos e agricultura
   (5) = Energia
   (6) = Serviços
   (7) = Produtos diversos

   Tabela 3: Insumo-Produto da economia norte-americana em 1958

Setor	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
(1)	.1588	.0064	.0025	.0304	.0014	.0083	.1594
(2)	.0057	.2645	.0436	.0099	.0083	.0201	.3413
(3)	.0264	.1506	.3557	.0139	.0142	.0070	.0236
(4)	.3299	.0565	.0495	.3636	.0204	.0483	.0649
(5)	.0089	.0081	.0333	.0295	.3412	.0237	.0020
(6)	.1190	.0901	.0996	.1260	.1722	.2368	.3369
(7)	.0063	.0126	.0196	.0098	.0064	.0132	.0012

   Temos que a matriz de consumo é dada por:

$$C = \begin{bmatrix} .1588 & .0064 & .0025 & .0304 & .0014 & .0083 & .1594 \\ .0057 & .2645 & .0436 & .0099 & .0083 & .0201 & .3413 \\ .0264 & .1506 & .3557 & .0139 & .0142 & .0070 & .0236 \\ .3299 & .0565 & .0495 & .3636 & .0204 & .0483 & .0649 \\ .0089 & .0081 & .0333 & .0295 & .3412 & .0237 & .0020 \\ .1190 & .0901 & .0996 & .1260 & .1722 & .2368 & .3369 \\ .0063 & .0126 & .0196 & .0098 & .0064 & .0132 & .0012 \end{bmatrix}$$

   A fim de satisfazer a demanda final de:

$$d = \begin{bmatrix} 74000 & 56000 & 10500 & 25000 & 17500 & 196000 & 5000 \end{bmatrix}^T$$

   (unidades em milhões de dólares), os níveis de produção devem ser tal que $(I - C)^{-1} \times X = d^T$:

$$\begin{bmatrix} 1.222119175 & 0.02708562 & 0.02256849 & 0.06770014 & 0.01354649 & 0.02265506 & 0.21674832 \\ 0.04324274 & 1.40455448 & 0.12438484 & 0.04658356 & 0.04036365 & 0.05162984 & 0.511031329 \\ 0.08055744 & 0.33874893 & 1.59274464 & 0.0555050 & 0.05077394 & 0.03262524 & 0.18095708 \\ 0.67324352 & 0.19045497 & 0.17627673 & 1.64480827 & 0.09483497 & 0.12663949 & 0.32647177 \\ 0.06357810 & 0.05312934 & 0.10097656 & 0.08971729 & 1.53927711 & 0.05751499 & 0.05899924 \\ 0.34094669 & 0.27106498 & 0.29527121 & 0.32529467 & 0.38417390 & 1.36736380 & 0.63713918 \\ 0.02134809 & 0.03032844 & 0.03924566 & 0.02311631 & 0.01746190 & 0.02111642 & 1.02455895 \end{bmatrix}$$

$$\times \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \overline{x}_3 \\ \overline{x}_4 \\ \overline{x}_5 \\ \overline{x}_6 \\ \overline{x}_7 \end{bmatrix} = \begin{bmatrix} 74000 \\ 56000 \\ 10500 \\ 25000 \\ 17500 \\ 196000 \\ 5000 \end{bmatrix}$$

Assim, os níveis de produção serão, respectivamente:

$$X \approx \begin{bmatrix} 99575,65 & 97703,02 & 51230,52 & 131569,92 & 49488,49 & 329554,45 & 13835,34 \end{bmatrix}^T$$

Para a demanda próxima ao ano de 1964:

$$d' = \begin{bmatrix} 99640 & 75548 & 14444 & 33501 & 23527 & 263985 & 6526 \end{bmatrix}^T$$

Temos que os níveis de produção serão dados por $(I-C)^{-1} \times X = d'^T$:

$$\begin{bmatrix} 1.222119175 & 0.02708562 & 0.02256849 & 0.06770014 & 0.01354649 & 0.02265506 & 0.21674832 \\ 0.04324274 & 1.40455448 & 0.12438484 & 0.04658356 & 0.04036365 & 0.05162984 & 0.511031329 \\ 0.08055744 & 0.33874893 & 1.59274464 & 0.0555050 & 0.05077394 & 0.03262524 & 0.18095708 \\ 0.67324352 & 0.19045497 & 0.17627673 & 1.64480827 & 0.09483497 & 0.12663949 & 0.32647177 \\ 0.06357810 & 0.05312934 & 0.10097656 & 0.08971729 & 1.53927711 & 0.05751499 & 0.05899924 \\ 0.34094669 & 0.27106498 & 0.29527121 & 0.32529467 & 0.38417390 & 1.36736380 & 0.63713918 \\ 0.02134809 & 0.03032844 & 0.03924566 & 0.02311631 & 0.01746190 & 0.02111642 & 1.02455895 \end{bmatrix}$$

$$\times \begin{bmatrix} \overline{x}'_1 \\ \overline{x}'_2 \\ \overline{x}'_3 \\ \overline{x}'_4 \\ \overline{x}'_5 \\ \overline{x}'_6 \\ \overline{x}'_7 \end{bmatrix} = \begin{bmatrix} 99640 \\ 75548 \\ 14444 \\ 33501 \\ 23527 \\ 263985 \\ 6526 \end{bmatrix}$$

Resultando em níveis de produção equivalentes, respectivamente, à:

$$X' \approx \begin{bmatrix} 134033,61 & 131686,64 & 69471,69 & 176912,00 & 66595,55 & 443772,90 & 18431,17 \end{bmatrix}^T$$

Dados que:

(1) = Produtos domésticos e pessoais não metálicos

(2) = Produtos metálicos finais (como veículos motorizados)

(3) = Produtos metálicos básicos e mineração

(4) = Produtos não metálicos e agricultura

(5) = Energia

(6) = Serviços

(7) = Produtos diversos

Vamos comparar a Demanda final e a produção de 1958 com a Demanda final e produção de 1964.

A Demanda aumentou de 1958 para 1964 em:

(1) 34,64%

(2) 34,90%

(3) 37,56%

(4) 34,00%

(5) 34,44%

(6) 34,69%

(7) 30,52%

Já a produção de 1958 para 1964 aumentou em:

(1) 34,60%

(2) 34,78%

(3) 35,60%

(4) 34,46%

(5) 34,57%

(6) 34,66%

(7) 33,22%

3. Retirado do site do IBGE, a matriz de consumo para o ano de 2015 é dada por:

Figura 1: Matriz Consumo do ano de 2015 para 12 setores da economia

	01 Agropecuária	02 Indústrias extrativas	03 Indústrias de transformação	04 Eletricidade e gás, água, esgoto e gestão de	05 Construção	06 Comércio	07 Transporte, armazenagem e correio	08 Informação e comunicação	09 Atividades financeiras, de seguros e serviços	10 Atividades imobiliárias	11 Outras atividades de serviços	12 Administração, defesa, saúde e educação
Agropecuária	0,040000	0,000008	0,075729	0,000102	0,001096	0,008983	0,000000	0,000000	0,000000	0,000000	0,004093	0,001363
Indústrias extrativas	0,000863	0,054875	0,042343	0,014855	0,009835	0,000064	0,000010	0,000000	0,000000	0,000582	0,000043	0,000061
Indústrias de transformação	0,211294	0,112176	0,278665	0,077120	0,209774	0,056695	0,186298	0,026816	0,008742	0,009331	0,071267	0,025694
Eletricidade e gás, água, esgoto e ges	0,023493	0,010865	0,015559	0,278018	0,001155	0,017458	0,005718	0,007179	0,004053	0,001247	0,016636	0,017357
Construção	0,000600	0,012810	0,000907	0,013016	0,095640	0,000939	0,003166	0,016649	0,002983	0,002977	0,003557	0,014197
Comércio	0,055685	0,027654	0,073310	0,018106	0,053831	0,025410	0,042389	0,022961	0,004513	0,003008	0,029879	0,012012
Transporte, armazenagem e correio	0,020341	0,084621	0,049872	0,018520	0,011928	0,050424	0,114175	0,008574	0,013583	0,000786	0,017800	0,011330
Informação e comunicação	0,000086	0,003769	0,005488	0,006511	0,002138	0,013016	0,007554	0,122009	0,039547	0,001412	0,038200	0,017405
Atividades financeiras, de seguros e s	0,015259	0,022101	0,017468	0,021561	0,014234	0,022972	0,024598	0,028425	0,123772	0,037877	0,016526	0,044452
Atividades imobiliárias	0,000040	0,001481	0,002001	0,004691	0,001811	0,036963	0,007358	0,013365	0,010510	0,002997	0,019127	0,003937
Outras atividades de serviços	0,003607	0,099139	0,048523	0,053922	0,024556	0,084810	0,061241	0,149943	0,108614	0,008492	0,096142	0,083457
Administração, defesa, saúde e educa	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000

Com esta tabela pode-se calcular a matriz de Leontief, $L = (I - C)^{-1}$. A demonstração do cálculo está no anexo Jupyter Notebook, item 3.

$$L = \begin{bmatrix} 1.0702 & 0.0196 & 0.1191 & 0.0164 & 0.0312 & 0.0202 & 0.0275 & 0.0081 & 0.0043 & 0.0016 & 0.0163 & 0.0075 \\ 0.0173 & 1.069 & 0.0675 & 0.030 & 0.0281 & 0.0063 & 0.0154 & 0.005 & 0.0022 & 0.0015 & 0.0069 & 0.0037 \\ 0.348 & 0.2334 & 1.4882 & 0.1947 & 0.3650 & 0.1259 & 0.3333 & 0.0860 & 0.0442 & 0.0190 & 0.139 & 0.0677 \\ 0.0465 & 0.0282 & 0.0441 & 1.3950 & 0.0157 & 0.0326 & 0.0226 & 0.0200 & 0.012 & 0.0031 & 0.0320 & 0.0298 \\ 0.0027 & 0.0173 & 0.0046 & 0.0219 & 1.1076 & 0.0032 & 0.0060 & 0.0228 & 0.0059 & 0.0037 & 0.0065 & 0.0175 \\ 0.0929 & 0.0612 & 0.1306 & 0.0500 & 0.0962 & 1.0453 & 0.082 & 0.0436 & 0.016 & 0.0059 & 0.0504 & 0.0249 \\ 0.0536 & 0.123 & 0.1046 & 0.0498 & 0.04 & 0.0716 & 1.158 & 0.0247 & 0.0254 & 0.0036 & 0.0363 & 0.0228 \\ 0.0079 & 0.016 & 0.019 & 0.0198 & 0.011 & 0.0242 & 0.0207 & 1.151 & 0.0594 & 0.0046 & 0.053 & 0.028 \\ 0.0325 & 0.0418 & 0.0443 & 0.0456 & 0.0328 & 0.038 & 0.046 & 0.046 & 1.1494 & 0.0447 & 0.0307 & 0.058 \\ 0.0061 & 0.009 & 0.0119 & 0.0122 & 0.0086 & 0.0427 & 0.0154 & 0.0220 & 0.0168 & 1.004 & 0.0254 & 0.0085 \\ 0.0453 & 0.1540 & 0.1186 & 0.1147 & 0.0720 & 0.1215 & 0.1165 & 0.209 & 0.1549 & 0.0178 & 1.1367 & 0.1143 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1 \end{bmatrix} \tag{7}$$

Figura 2: Matriz de Leontief - Matriz Consumo do ano de 2015

Descrição da atividade	01 Agropecuária	02 Indústrias extrativas	03 Indústrias de transformação	04 Eletricidade e gás, água, esgoto	05 Construção	06 Comércio	07 Transporte, armazenagem	08 Informação e comunicação	09 Atividades financeiras	10 Atividades imobiliárias	11 Outras atividades de serviços	12 Administração, defesa, saúde e educação
01 Agropecuária	1,070423	0,020221	0,119541	0,016951	0,032572	0,020605	0,027929	0,008638	0,004569	0,001716	0,016688	0,007821
02 Indústrias extrativas	0,017118	1,068457	0,066637	0,030737	0,029214	0,006457	0,015300	0,004882	0,002446	0,001611	0,007045	0,004035
03 Indústrias de transformaçã	0,338342	0,228313	1,475315	0,190108	0,354864	0,125413	0,324264	0,086666	0,045039	0,018839	0,137939	0,067116
04 Eletricidade e gás, água, esg	0,046079	0,028026	0,043750	1,390726	0,015665	0,032801	0,022582	0,020133	0,012474	0,003128	0,032043	0,029659
05 Construção	0,002713	0,017114	0,004714	0,021542	1,105148	0,003374	0,006093	0,022495	0,005938	0,003652	0,006547	0,017248
06 Comércio	0,098931	0,067164	0,139921	0,054462	0,102702	1,049128	0,089774	0,047267	0,019364	0,006476	0,054238	0,027237
07 Transporte, armazenagem	0,052697	0,122596	0,103238	0,049690	0,046314	0,071665	1,155888	0,026338	0,026436	0,003812	0,037013	0,023483
08 Informação e comunicação	0,008163	0,016810	0,020077	0,019914	0,011803	0,024248	0,020721	1,150185	0,058744	0,004658	0,052660	0,028402
09 Atividades financeiras, de s	0,032711	0,042425	0,044851	0,046220	0,033161	0,039227	0,046932	0,048274	1,150138	0,044830	0,031619	0,058562
10 Atividades imobiliárias	0,005779	0,008309	0,011106	0,011210	0,008089	0,039152	0,014229	0,020124	0,015367	1,003839	0,023276	0,007795
11 Outras atividades de serviç	0,043049	0,144534	0,111836	0,107755	0,068080	0,114687	0,109770	0,196537	0,145253	0,016832	1,128525	0,107208
12 Administração, defesa, saú	0,003040	0,007681	0,006545	0,008341	0,003652	0,005982	0,006806	0,009028	0,006548	0,000804	0,006158	1,004951

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Contas Nacionais.

Anexo

Algoritmos usados no Jupyter Notebook, para os cálculos das matrizes.

In [43]:

```
import numpy as np
import pandas as pd
```

5) Exemplo

Para testar o algoritmo, foram realizados os cálculos com a matriz do item 5 do projeto.

In [2]:

```
consumo = np.array([[0.50, 0.40, 0.20],
                    [0.20, 0.30, 0.10],
                    [0.10, 0.10, 0.30]])
```

In [ ]:

```

```

In [3]:

```
np.identity(consumo.shape[0]) - consumo
```

Out[3]:

```
array([[ 0.5, -0.4, -0.2],
       [-0.2,  0.7, -0.1],
       [-0.1, -0.1,  0.7]])
```

In [4]:

```
input_table = np.array([[0.50, -0.40, -0.20],
                        [-0.20, 0.70, -0.10],
                        [-0.10, -0.10, 0.70]])
```

In [5]:

```
total_output = np.array([50, 30, 20])
```

In [6]:

```
resultado = np.sum(np.linalg.inv(input_table) * total_output, axis=1)
```

In [7]:

```
resultado
```

Out[7]:

```
array([225.92592593, 118.51851852,  77.77777778])
```

In [ ]:

```

```

Tarefas

2)

a)

Criando a matriz:

In [36]:

```
input_table = np.array([[.1588, .0064, .0025, .0304, .0014, .0083, .1594],
                        [.0057, .2645, .0436, .0099, .0083, .0201, .3413],
                        [.0264, .1506, .3557, .0139, .0142, .0070, .0236],
                        [.3299, .0565, .0495, .3636, .0204, .0483, .0649],
                        [.0089, .0081, .0333, .0295, .3412, .0237, .0020],
                        [.1190, .0901, .0996, .1260, .1722, .2368, .3369],
                        [.0063, .0126, .0196, .0098, .0064, .0132, .0012]])
```

In [ ]:

```

```

Subtraindo a identidade pela matriz:

In [37]:

```
input_table = np.identity(input_table.shape[0]) - input_table
```

In [38]:

```
input_table
```

Out[38]:

```
array([[ 0.8412, -0.0064, -0.0025, -0.0304, -0.0014, -0.0083, -0.1594],
       [-0.0057,  0.7355, -0.0436, -0.0099, -0.0083, -0.0201, -0.3413],
       [-0.0264, -0.1506,  0.6443, -0.0139, -0.0142, -0.007 , -0.0236],
       [-0.3299, -0.0565, -0.0495,  0.6364, -0.0204, -0.0483, -0.0649],
       [-0.0089, -0.0081, -0.0333, -0.0295,  0.6588, -0.0237, -0.002 ],
       [-0.119 , -0.0901, -0.0996, -0.126 , -0.1722,  0.7632, -0.3369],
       [-0.0063, -0.0126, -0.0196, -0.0098, -0.0064, -0.0132,  0.9988]])
```

In [ ]:

```

```

Calculando a sua inversa, ou seja, a matriz de Leontief:

In [39]:

```
np.linalg.inv(input_table)
```

Out[39]:

```
array([[1.22119175, 0.02708562, 0.02256849, 0.06770014, 0.01354649,
        0.02265506, 0.21674832],
       [0.04324274, 1.40455448, 0.12438484, 0.04658356, 0.04036365,
        0.05162984, 0.51031329],
       [0.08055744, 0.33874893, 1.59274464, 0.0555055 , 0.05077394,
        0.03262524, 0.18095708],
       [0.67324352, 0.19045497, 0.17627673, 1.64480827, 0.09483497,
        0.12663949, 0.32647177],
       [0.0635781 , 0.05312934, 0.10097656, 0.08971729, 1.53927711,
        0.05751499, 0.05899924],
       [0.34094669, 0.27106498, 0.29527121, 0.32529467, 0.3841739 ,
        1.3673638 , 0.63713918],
       [0.02134809, 0.03032844, 0.03924566, 0.02311631, 0.0174619 ,
        0.02111642, 1.02455895]])
```

In [40]:

```
total_output = np.array([74000, 56000, 10500, 25000, 17500, 196000, 5000])
```

In [ ]:

Obtendo o Resultado de X:

In [41]:

```
resultado = np.sum(np.linalg.inv(input_table) * total_output, axis=1)
```

In [42]:

```
resultado
```

Out[42]:

```
array([ 99575.65339765,  97703.02286349,  51230.52316638, 131569.92192872,
        49488.49137236, 329554.45256999,  13835.33571501])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

b)

In [14]:

```
input_table = np.array([[.1588, .0064, .0025, .0304, .0014, .0083, .1594],
                        [.0057, .2645, .0436, .0099, .0083, .0201, .3413],
                        [.0264, .1506, .3557, .0139, .0142, .0070, .0236],
                        [.3299, .0565, .0495, .3636, .0204, .0483, .0649],
                        [.0089, .0081, .0333, .0295, .3412, .0237, .0020],
                        [.1190, .0901, .0996, .1260, .1722, .2368, .3369],
                        [.0063, .0126, .0196, .0098, .0064, .0132, .0012]])
```

In [15]:

```
input_table = np.identity(input_table.shape[0]) - input_table
```

In [16]:

```
input_table
```

Out[16]:

```
array([[ 0.8412, -0.0064, -0.0025, -0.0304, -0.0014, -0.0083, -0.1594],
       [-0.0057,  0.7355, -0.0436, -0.0099, -0.0083, -0.0201, -0.3413],
       [-0.0264, -0.1506,  0.6443, -0.0139, -0.0142, -0.007 , -0.0236],
       [-0.3299, -0.0565, -0.0495,  0.6364, -0.0204, -0.0483, -0.0649],
       [-0.0089, -0.0081, -0.0333, -0.0295,  0.6588, -0.0237, -0.002 ],
       [-0.119 , -0.0901, -0.0996, -0.126 , -0.1722,  0.7632, -0.3369],
       [-0.0063, -0.0126, -0.0196, -0.0098, -0.0064, -0.0132,  0.9988]])
```

In [ ]:

```
np.linalg.inv(input_table)
```

In [17]:

```
total_output = np.array([99640, 75548, 14444, 33501, 23527, 263985, 6526])
```

In [18]:

```
resultado = np.sum(np.linalg.inv(input_table) * total_output, axis=1)
```

In [19]:

```
resultado
```

Out[19]:

```
array([134033.61532755, 131686.642968  ,  69471.69907937, 176912.002023  ,
        66595.55670623, 443772.90207667,  18431.17768329])
```

In [ ]:

In [ ]:

```

```

3)

Importando a tabela com os dados do IBGE:

In [52]:

```
df_table = pd.read_excel('ibge.xlsx')
```

In [53]:

```
df_table.head()
```

Out[53]:

	01\nAgropecuária	02\nIndústrias extrativas	03\nIndústrias de transformação	04\nEletricidade e gás, água, esgoto e gestão de resíduos	05\nConstrução	06\nComé
0	0.040000	0.000008	0.075729	0.000102	0.001096	0.008
1	0.000863	0.054875	0.042343	0.014855	0.009835	0.000
2	0.211294	0.112176	0.278665	0.077120	0.209774	0.056
3	0.023493	0.010865	0.015559	0.278018	0.001155	0.017
4	0.000600	0.012810	0.000907	0.013016	0.095640	0.000

Convertendo os dados em Numpy Array:

In [54]:

```
input_table = np.array(df_table.values)
```

In [55]:

```
input_table
```

Out[55]:

```
array([[3.99995822e-02, 7.67539231e-06, 7.57288057e-02, 1.01978381e-04,
        1.09598487e-03, 8.98286007e-03, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 4.09262654e-03, 1.36279826e-03],
       [8.62699225e-04, 5.48752173e-02, 4.23431276e-02, 1.48548508e-02,
        9.83539667e-03, 6.35922537e-05, 9.89282118e-06, 0.00000000e+00,
        0.00000000e+00, 5.82493328e-04, 4.28511670e-05, 6.12679654e-05],
       [2.11294467e-01, 1.12175859e-01, 2.78664919e-01, 7.71203778e-02,
        2.09774350e-01, 5.66952196e-02, 1.86297651e-01, 2.68157253e-02,
        8.74156603e-03, 9.33088369e-03, 7.12669322e-02, 2.56944600e-02],
       [2.34934096e-02, 1.08645178e-02, 1.55586610e-02, 2.78017788e-01,
        1.15450065e-03, 1.74578906e-02, 5.71805064e-03, 7.17901309e-03,
        4.05317684e-03, 1.24741496e-03, 1.66357753e-02, 1.73570490e-02],
       [5.99502851e-04, 1.28102298e-02, 9.07270409e-04, 1.30161497e-02,
        9.56400994e-02, 9.39348434e-04, 3.16570278e-03, 1.66492613e-02,
        2.98288755e-03, 2.97657754e-03, 3.55732704e-03, 1.41967811e-02],
       [5.56848328e-02, 2.76544385e-02, 7.33102584e-02, 1.81057979e-02,
        5.38313607e-02, 2.54096477e-02, 4.23887602e-02, 2.29608578e-02,
        4.51261810e-03, 3.00771712e-03, 2.98788264e-02, 1.20118330e-02],
       [2.03413197e-02, 8.46212002e-02, 4.98721393e-02, 1.85198920e-02,
        1.19277314e-02, 5.04241149e-02, 1.14175028e-01, 8.57429822e-03,
        1.35831023e-02, 7.85816471e-04, 1.77995586e-02, 1.13296059e-02],
       [8.56432645e-05, 3.76861762e-03, 5.48828364e-03, 6.51116509e-03,
        2.13819847e-03, 1.30164259e-02, 7.55415825e-03, 1.22008982e-01,
        3.95467542e-02, 1.41227156e-03, 3.82001149e-02, 1.74050698e-02],
       [1.52591231e-02, 2.21012922e-02, 1.74675666e-02, 2.15607019e-02,
        1.42335697e-02, 2.29722474e-02, 2.45975106e-02, 2.84250113e-02,
        1.23772430e-01, 3.78767202e-02, 1.65262667e-02, 4.44523927e-02],
       [3.96883421e-05, 1.48135072e-03, 2.00110933e-03, 4.69100551e-03,
        1.81082637e-03, 3.69625433e-02, 7.35828039e-03, 1.33650625e-02,
        1.05097187e-02, 2.99672668e-03, 1.91272646e-02, 3.93688075e-03],
       [3.60746141e-03, 9.91392048e-02, 4.85225791e-02, 5.39218413e-02,
        2.45560708e-02, 8.48102634e-02, 6.12405202e-02, 1.49943218e-01,
        1.08614350e-01, 8.49194675e-03, 9.61423747e-02, 8.34569042e-02],
       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00]])
```

Subtraindo a matriz da sua identidade:

In [56]:

```
input_table = np.identity(input_table.shape[0]) - input_table
```

In [57]:

```
input_table
```

Out[57]:

```
array([[ 9.60000418e-01, -7.67539231e-06, -7.57288057e-02,
        -1.01978381e-04, -1.09598487e-03, -8.98286007e-03,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00, -4.09262654e-03, -1.36279826e-03],
       [-8.62699225e-04,  9.45124783e-01, -4.23431276e-02,
        -1.48548508e-02, -9.83539667e-03, -6.35922537e-05,
        -9.89282118e-06,  0.00000000e+00,  0.00000000e+00,
        -5.82493328e-04, -4.28511670e-05, -6.12679654e-05],
       [-2.11294467e-01, -1.12175859e-01,  7.21335081e-01,
        -7.71203778e-02, -2.09774350e-01, -5.66952196e-02,
        -1.86297651e-01, -2.68157253e-02, -8.74156603e-03,
        -9.33088369e-03, -7.12669322e-02, -2.56944600e-02],
       [-2.34934096e-02, -1.08645178e-02, -1.55586610e-02,
         7.21982212e-01, -1.15450065e-03, -1.74578906e-02,
        -5.71805064e-03, -7.17901309e-03, -4.05317684e-03,
        -1.24741496e-03, -1.66357753e-02, -1.73570490e-02],
       [-5.99502851e-04, -1.28102298e-02, -9.07270409e-04,
        -1.30161497e-02,  9.04359901e-01, -9.39348434e-04,
        -3.16570278e-03, -1.66492613e-02, -2.98288755e-03,
        -2.97657754e-03, -3.55732704e-03, -1.41967811e-02],
       [-5.56848328e-02, -2.76544385e-02, -7.33102584e-02,
        -1.81057979e-02, -5.38313607e-02,  9.74590352e-01,
        -4.23887602e-02, -2.29608578e-02, -4.51261810e-03,
        -3.00771712e-03, -2.98788264e-02, -1.20118330e-02],
       [-2.03413197e-02, -8.46212002e-02, -4.98721393e-02,
        -1.85198920e-02, -1.19277314e-02, -5.04241149e-02,
         8.85824972e-01, -8.57429822e-03, -1.35831023e-02,
        -7.85816471e-04, -1.77995586e-02, -1.13296059e-02],
       [-8.56432645e-05, -3.76861762e-03, -5.48828364e-03,
        -6.51116509e-03, -2.13819847e-03, -1.30164259e-02,
        -7.55415825e-03,  8.77991018e-01, -3.95467542e-02,
        -1.41227156e-03, -3.82001149e-02, -1.74050698e-02],
       [-1.52591231e-02, -2.21012922e-02, -1.74675666e-02,
        -2.15607019e-02, -1.42335697e-02, -2.29722474e-02,
        -2.45975106e-02, -2.84250113e-02,  8.76227570e-01,
        -3.78767202e-02, -1.65262667e-02, -4.44523927e-02],
       [-3.96883421e-05, -1.48135072e-03, -2.00110933e-03,
        -4.69100551e-03, -1.81082637e-03, -3.69625433e-02,
        -7.35828039e-03, -1.33650625e-02, -1.05097187e-02,
         9.97003273e-01, -1.91272646e-02, -3.93688075e-03],
       [-3.60746141e-03, -9.91392048e-02, -4.85225791e-02,
        -5.39218413e-02, -2.45560708e-02, -8.48102634e-02,
        -6.12405202e-02, -1.49943218e-01, -1.08614350e-01,
        -8.49194675e-03,  9.03857625e-01, -8.34569042e-02],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  1.00000000e+00]])
```

In [ ]:

```

```

In [ ]:

Calculando a Matriz de Leontief:

In [58]:

```
np.linalg.inv(input_table)
```

Out[58]:

```
array([[1.07021867, 0.01967333, 0.11913949, 0.01649366, 0.03127004,
        0.02024246, 0.02757532, 0.00811978, 0.00431414, 0.00163785,
        0.01633623, 0.00750962],
       [0.01735771, 1.0691786 , 0.06755098, 0.0309118 , 0.02817317,
        0.00631082, 0.01541208, 0.00444266, 0.00225996, 0.00156184,
        0.00690959, 0.00377236],
       [0.3483731 , 0.23340532, 1.48826527, 0.19475651, 0.36504546,
        0.12594738, 0.33335425, 0.08606937, 0.04421615, 0.01903134,
        0.1395327 , 0.06776518],
       [0.04658876, 0.02821867, 0.04416328, 1.39503294, 0.01575094,
        0.03264903, 0.02267191, 0.02005962, 0.0123883 , 0.00311106,
        0.03209834, 0.02987718],
       [0.00271298, 0.01738047, 0.00468648, 0.02195886, 1.10766586,
        0.00322741, 0.00608135, 0.02283331, 0.00591383, 0.00371557,
        0.00652446, 0.01755871],
       [0.09294423, 0.06123864, 0.13065574, 0.05000287, 0.09621484,
        1.04539139, 0.0825488 , 0.04362592, 0.0168242 , 0.00595811,
        0.05048524, 0.02495769],
       [0.05364207, 0.1239489 , 0.10467065, 0.04981436, 0.046754  ,
        0.07169284, 1.15828774, 0.02476849, 0.02542939, 0.00369411,
        0.03637255, 0.02289422],
       [0.00799788, 0.0168811 , 0.01996142, 0.01998918, 0.0116758 ,
        0.02425987, 0.02071228, 1.1518582 , 0.05940265, 0.00468672,
        0.0531014 , 0.0287025 ],
       [0.03253443, 0.04186978, 0.04437507, 0.04568207, 0.03285911,
        0.03822608, 0.0463884 , 0.0469479 , 1.14940414, 0.04474177,
        0.03074231, 0.05808392],
       [0.00615421, 0.0090313 , 0.01193658, 0.01221296, 0.00865816,
        0.04276788, 0.01542329, 0.02207385, 0.01685839, 1.00419342,
        0.02543754, 0.00854892],
       [0.04538018, 0.15409754, 0.11865022, 0.11471964, 0.07200748,
        0.12153775, 0.11659434, 0.2096626 , 0.15497403, 0.01788404,
        1.13671869, 0.11438985],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 1.        ]])
```

Referências.

- Howard Anton, Chris Rorres, *Algebra Linear (com aplicações)*, Bookman, Porto Alegre, 2012
- Wassily W. Leontief, *Input-Output Economics*, Scientific American, October 1951, pp. 15–21
- David C. Lay, Steven R. Lay, and Judi J. McDonald, *Linear Algebra and its Applications*, Pearson Education, Inc., 2016
- Ronald E. Miller and Peter D. Blair, *Input-Output Analysis*, Cambridge University Press, 2009