

Trabalho de Python

- Os grupos são compostos por **4 elementos**.
- O ficheiro com a resolução deve ser **enviado, via Moodle até às 23h59min (hora portuguesa) do dia 6 de Maio de 2022**. Cada grupo envia apenas um ficheiro do tipo .py. O nome do ficheiro deve ser o número de aluno do aluno que submete o ficheiro, por exemplo, 12345.py.
- O ficheiro deve ter a estrutura:

```
# Group: <student number1, student number2; ... >

# a)
<código>

# b)
<código>

# c)
<código>

# d)
<código>
```

- Critérios gerais de correção:
 - Código não executa: 0 pontos
 - Envio do ficheiro depois do prazo: 0 pontos
 - Generalidade do código
 - Código bem organizado
 - Eficiência do código
 - Soluções criativas

1. Conectividade da economia do Reino Unido: 2000 - 2019

O estudo das ligações entre setores de uma economia pode ser feito com recurso a dados de input-output. Neste contexto, a intensidade da dependência de um determinado sector do "resto" da economia é medida a partir da matriz inversa de Leontief. Uma vez que a soma dos elementos da j -ésima coluna desta matriz nos dá o total das dependências direta e indiretas deste sector aos restantes setores da economia, podemos medir a conectividade da economia do Reino Unido através do *índice de ligações totais*:

$$\mathbf{b} = \frac{n\mathbf{1}^T L}{\mathbf{1}^T L \mathbf{1}}$$

onde

- $\mathbf{1}$ é uma matriz coluna de tamanho apropriado (e.g. $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$);
- L é a matriz inversa de Leontief;
- n é o número de setores da economia
- o símbolo T representa a operação, transposição de matrizes.

O resultado de $\mathbf{b} = [b_j]$ é uma matriz linha com n entradas, sendo b_j o índice de ligações totais do setor j .

Este índice tem sido amplamente usado para medir as ligações entre setores de uma economia e a sua estabilidade ao longo do tempo. Valores de b_j maiores do que 1, indicam sectores com uma forte dependência de inputs interindustriais. Neste caso, dizemos que o sector j está *fortemente ligado*. Caso contrário, dizemos que está *fracamente ligado*.

Um relatório de uma agência governamental sobre a economia do Reino Unido inclui as seguintes conclusões:

- 1) "Relativamente aos efeitos da recessão que terminou em 2009, a taxa de crescimento da produção após a recessão (2010-2019) tem sido, em média, aproximadamente igual à da taxa de crescimento antes da recessão (2000-2008)."
- 2) "Os sectores da Agricultura, Produção e Construção do Reino Unido têm estado fortemente ligados ao longo do período 2000-2019."
- 3) "Durante o período 2000-2019, o sector imobiliário é o sector mais fracamente ligado."

O vosso trabalho consiste em escrever código para verificar estas conclusões.

O Instituto Nacional de Estatística do Reino Unido produz quadros sumários de input-outputs com 10 sectores. Os sectores considerados são: Agricultura [S0], Produção [S1], Construção [S2], Distribuição, transportes, hotéis e restaurantes [S3], Informação e comunicação [S4],

Finanças e seguros [S5], Atividades imobiliárias [S6], Atividades profissionais e de apoio [S7], Governo, saúde & educação [S8], e Outros serviços [S9].

A partir do link:

<https://www.ons.gov.uk/economy/nationalaccounts/supplyandusetable/datasets/inputoutputsupplyandusetablesummarytables>

o ficheiro 'bb21a10summarytables.xlsx' pode ser descarregado. Este ficheiro contém dados de input-output para o Reino Unido, de 1997 a 2019, medidos em milhões de libras.

Descarregue o ficheiro e importe os dados para Python. O código abaixo cria a função `getio(year)` para importar os dados de input-output para um ano `year` específico. Este código utiliza o módulo Pandas para importar os dados do ficheiro e cria dois objetos do tipo array: um array 2D, `z`, com a tabela de input-output, e um array 1D (vetor), `x`.¹ Altere o variável `path`, para o local no seu computador onde foi descarregado o ficheiro.

Um exemplo para Windows é:

`path = 'C:/Users/somename/Downloads/bb21a10summarytables.xlsx'.`

```
import pandas as pd
import numpy as np

def getio (year):
    path = '/Users/somename/Downloads/bb21a10summarytables.xlsx'
    # Input output table
    df = pd.read_excel(path,
                       sheet_name = str(year),
                       usecols = "C:L",
                       header = None,
                       skiprows = 52,
                       nrows = 10)
    z = np.array(df, dtype = float) # £ million

    # Output per sector
    dfx = pd.read_excel(path,
                        sheet_name = str(year),
                        usecols = "C:L",
                        header = None,
                        skiprows = 75,
                        nrows = 1)
    x = np.array(dfx, dtype = float)[0] # £ million
    return [z, x]

# Example: Get input-output data for 2000
z, x = getio(2000)
```

a) Defina *output* num ano como a soma da produção de todos os sectores para esse ano.

Se o output no ano t é x_t , defina a taxa de crescimento de output no ano t como:

$$\frac{x_t - x_{t-1}}{x_{t-1}} \times 100.$$

Crie:

¹ O link https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html fornece mais informação sobre a função `read_excel`.

- i. a lista `outlist` com o output anual no período 1999 – 2019;
- ii. a função `growth(year)` que devolve a taxa de crescimento da produção para o ano especificado;
- iii. a lista `growthlist` com a taxa de crescimento anual do output para o período 2000 - 2019.

Escreva código para confirmar ou corrigir a conclusão 1.

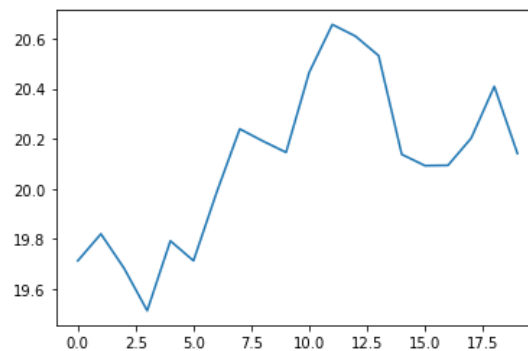
```
>>> round(growth(2009), 2)
-2.2
>>> round(growthlist[9], 2)
-2.2
```

b) Crie:

- i. a função `leon(year)` que devolve um array com a matriz inversa de Leontief para um dado ano;
- ii. a lista `leons`, com as matrizes inversas de Leontief para o Reino Unido no período 2000-2019.

A execução do seguinte código deverá devolver a figura abaixo.

```
import matplotlib.pyplot as plt
plt.plot([np.sum(leons[i]) for i in range(20)])
```

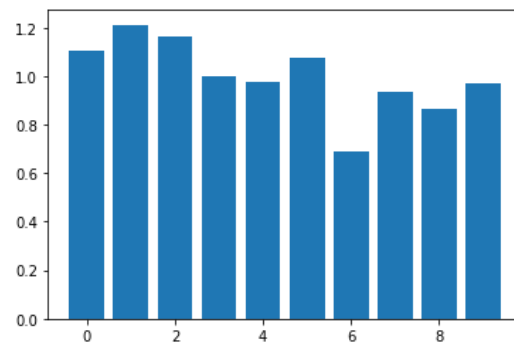


c) Crie:

- i. A função `bindex(year)`, que devolve um array com o índice de ligações totais de cada sector para um dado ano, `year`. Use a lista `leons` criada em b);
- ii. a função `blist(s)`, que devolve uma lista com o índice de ligações totais para o sector `s` para cada ano no período 2000-2019.

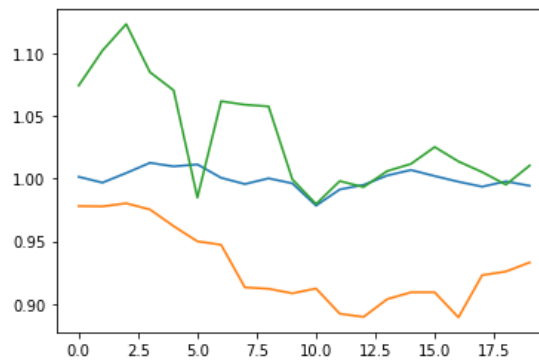
Abaixo estão duas figuras, e código associado, para ajudar a confirmar os resultados e dar algumas ideias sobre como explorar as afirmações.

```
plt.bar(x=range(10), height=bindex(2000))
```



Ambas as linhas de código abaixo devolvem a mesma figura.

```
>>> plt.plot(blist(3)); plt.plot(blist(4)); plt.plot(blist(5))
>>> [plt.plot(blist(i)) for i in range(3,6)]
```



d) Escreva código para confirmar ou corrigir as conclusões 2 e 3 do relatório.