

# 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 12 de Dezembro de 2021**. 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. Modelo input-output com gases de efeito de estufa (GEE)

O Instituto Nacional de Estatísticas Britânico (ONS) produz tabelas sumário input-output com 10 sectores. Os sectores são: Agricultura [A], Produção [B-E], Construção [F], Distribuição, transportes, hotelaria e restaurantes [G-I], Informação e comunicação [J], Serviços financeiros e seguros [K], Imobiliário [L], Serviços profissionais e de suporte [M-N], Governo, saúde e educação [O-Q], Outros serviços [R-T]. Entre parênteses rectos estão os códigos dos sectores, usados para agregar os dados.

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, em milhões de £, para os anos 1997 a 2019. Descarregue o ficheiro e adapte o código em baixo para importar os dados para trabalhar em Python. O código usa o módulo Pandas para importar a tabela input output de consumo intermédio por sector de 2019, e cria dois arrays: um array 2D com a tabela de input-output,  $z$ , e um vetor (array 1D) com o output por sector,  $x$ .<sup>1</sup> Mude a variável `path`, para o local no seu computador onde foi descarregado o ficheiro. Um exemplo em Windows é,

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

```
>>> import pandas as pd
>>> import numpy as np
>>> path = '/Users/somename/Downloads/bb21a10summarytables.xlsx'
>>> # Input output table
>>> df = pd.read_excel(path,
                        sheet_name = 22,
                        usecols = "C:L",
                        header = None,
                        skiprows = 52,
                        nrows = 10)
>>> z = np.array(df, dtype = float) # £ million

>>> # Output por sector
>>> dfx = pd.read_excel(path,
                        sheet_name = 22,
                        usecols = "C:L",
                        header = None,
                        skiprows = 75,
                        nrows = 1)
>>> x = np.array(dfx, dtype = float)[0] # £ million
```

Em muitos casos, o valor monetário da produção de cada sector, pode não ser o indicador mais adequado para quantificar o impacto de uma alteração da procura. Neste trabalho estamos também interessados nos impactos em termos de emissões de gases de efeito de estufa (GEE), medidos em emissões de CO<sub>2</sub>. Estas emissões podem ser incluídas no modelo

---

<sup>1</sup> Veja o link ([https://pandas.pydata.org/docs/reference/api/pandas.read\\_excel.html](https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html)) para uma descrição das opções da função `read_excel`.

input output, sabendo para cada sector a intensidade de emissões de GEE, i.e., a quantidade de CO2 emitida por unidade de produção de cada sector.

Os dados de intensidade de GEE de cada sector, para os anos 1997 a 2019, em milhares de toneladas de CO2 / milhões de £, podem ser obtidos do ficheiro 'ghgintensity\_uk.xlsx', disponível no moodle. Este ficheiro foi preparado a partir do ficheiro que pode ser obtido do link em rodapé.<sup>2</sup> Adapte o código seguinte para importar os dados para trabalhar em Python.

```
>>> pathe = '/Users/somename/Downloads/ghgintensity_uk.xlsx'
>>> dfem = pd.read_excel(pathe,
                        sheet_name = 0,
                        usecols = "B:X",
                        skiprows = 1,
                        header = None,
                        nrows = 10)
>>> E = np.array(dfem, dtype = float) #Thousand tonnes CO2 /£ million
```

Escreva código para justificar as respostas às perguntas seguintes. Escreva o código de modo a ser possível aplicar a matrizes input output com um qualquer número de sectores.

- a) Crie a função `coef_mat(z, x)`, que, dado um array com uma tabela input output  $z$ , e um array  $x$ , com o vetor de produção de cada sector, devolve um array com a matriz dos coeficientes técnicos. Encontre a matriz dos coeficientes técnicos de 2019. Em baixo é apresentado um exemplo para uma economia com dois sectores.

```
>>> z = np.array([[3, 2],
                  [1, 4]]) # £ million
>>> x = np.array([10, 10]) # £ million
>>> coef_mat(z, x)
array([[0.3, 0.2],
       [0.1, 0.4]])
```

- b) Se  $A$  é a matriz dos coeficientes técnicos, a matriz inversa de Leontief pode ser aproximada pela série de potências,

$$I + A + A^2 + A^3 + \dots + A^m.$$

Quando  $m$  aumenta,  $A^m$  tende para zero e a série converge para a matriz inversa de Leontief. Crie a função `leon(a, dif = 1e-6)`, que, dado um array com a matriz dos coeficientes técnicos  $a$ , devolve um array com a matriz inversa de Leontief. Esta matriz é obtida quando, adicionando mais um termo à séries de potências, a variação de cada entrada da matriz foi menor do que `dif`. Aplique a função `leon` e encontre a matriz inversa de Leontief para 2019.

Crie a função `impact_output(a)` que dada uma matriz dos coeficientes técnicos  $a$ , devolve uma lista em que cada elemento é a alteração de produção da economia que resulta de um aumento unitário da procura de um bem. Qual o sector para o qual um

---

<sup>2</sup> <https://www.ons.gov.uk/economy/environmentalaccounts/datasets/ukenvironmentalaccountsatmosphericemissionsgreenhousegasemissionsintensitybyeconomicsectorunitedkingdom>

aumento unitário pela procura do seu bem, gera o maior impacto na produção da economia?

```
>>> a = coef_mat(z, x)
>>> np.round(leon(a), 2)
array([[1.5 , 0.5 ],
       [0.25, 1.75]])

>>> np.round(impact_output(a), 2)
array([1.75, 2.25])

>>> # Qual o sector que gera o maior impacto na produção da economia?
>>> maximp(a)
'Sector 2'
```

No exemplo acima, criámos uma função `maximp(a)`, que dada uma matriz dos coeficientes técnicos `a`, devolve o sector com o maior impacto na produção da economia.

- c) Crie a função `impact_ghg(a, ghg)`, que, dado um array com a matriz dos coeficientes técnicos `a` e um vetor `ghg` com a intensidade de GEE de cada sector, em milhares de toneladas de CO<sub>2</sub> / milhões de £, devolve uma lista (em toneladas de CO<sub>2</sub> / £) com o aumento de emissões da economia que resulta de um aumento unitário na procura de cada bem. Do exemplo abaixo, um aumento de 1 £ na procura final pelo bem do sector 1, resulta na emissão de mais 0.003 toneladas de CO<sub>2</sub>.

```
>>> np.round(impact_ghg(a, E), 3)
array([0.003, 0.009])
```

- d) Defina a contribuição de cada sector para o bem estar da população como o output menos o custo das emissões de GEE. Admita que o custo da tonelada de emissões de GEE é o preço do CO<sub>2</sub> no mercado de emissões do Reino Unido (UK Emissions Trading System, i.e. UK ETS). Este mercado foi iniciado a 1 de Janeiro de 2021, e a tonelada de CO<sub>2</sub> tem sido transacionada a cerca de 50 £ / tonelada<sup>3</sup>. Crie a função `wellbeing(a, ghg, p = 50)`, que, dado um array com a matriz dos coeficientes técnicos `a`, um array `ghg` com a intensidade energética e cada sector, e o preço das emissões de CO<sub>2</sub>, `p`, em £ / tonelada, devolve uma lista com a contribuição para o bem-estar (em £) que resulta do aumento unitário de cada bem. Qual o sector cujo aumento unitário pela procura do seu bem, mais contribui para o bem-estar? Qual o intervalo de preços do CO<sub>2</sub> para o qual a resposta anterior não se altera? Assuma que os preços são números inteiros e o preço máximo é 200 £ / tonelada.

```
>>> E = np.array([1, 5]) # Thousand tonnes CO2 / £ million
>>> np.round(wellbeing(a, E), 2)
array([1.61, 1.79])
```

<sup>3</sup> <https://www.ft.com/content/56e02d3d-8c31-4937-be50-60d4bf9342f7>  
<https://ember-climate.org/data/carbon-price-viewer/>

```
>>> # Qual o sector que mais contribui para o bem-estar?
>>> maxwell(a, E)
'Sector 2'

>>> # Qual o intervalo de preços do CO2 para o qual a resposta anterior
não muda?
>>> price_range(a, E)
[0, 76]

>>> # Só para confirmar...
>>> maxwell(a, E, p = 77)
'Sector 1'
```

No exemplo acima, criámos a função `maxwell(a, ghg, p = 50)`, que, dado um array com a matriz dos coeficientes técnicos `a`, um vetor `ghg` com a intensidade energética de cada setor e o preço `p` das emissões de CO2 de 50 £ / tonelada, devolve o sector em que um aumento unitário pela procura final do seu bem resulta no maior impacto no bem-estar. Criámos também a função `price_range(a, ghg)`, para devolver uma lista com o intervalo de preços pretendido.

Do exemplo, vemos que um aumento unitário da procura final do bem 2 aumenta em 1.79 £ o bem estar da população. Vemos ainda que o sector 2 é o que mais contribui para o bem estar se o preço do CO2 for inferior a 77 £ / tonelada. Quando o preço do CO2 é 77 £ / tonelada, ou maior, o sector 1 é o que mais contribui para o bem-estar.