**Advance EEIOA course 21/22**

**Week 19 Exercises: Modelling Circular Economy**

**Objectives**

* Understand how to create functions for modelling CE
* Create functions in Python

This workshop will consist of two parts. **Part 1** will test and develop your conceptual understanding on how to create functions for implementing circular economy scenarios. **Part 2** will require you to create functions in Python to model multiple circular economy scenarios

**Part 1: Understanding functions for CE modelling**

The following IO system shows the direct requirements of an economy:



You are asked to explore the effects of two circular economy policies in this economy, which are:

1. Increasing the use of steel scrap in primary production of cars. This implies a reduction of 20% in the direct requirements of transport from the steel production, and an increase of 36% in the direct requirements of steel production from secondary steel.
2. Implementing subsidies for car sharing schemes, and use of second-hand bikes. This implies a final demand reduction of 35% in transport sector, but also and increase of repair transport equipment in 50%

Based on the lecture:

1a. What type of circularity interventions are considered in this economy?

1b. Build an alternative IOT that represents the scenarios of each circular economy policy

1c. Create a general equation that can be used to assess multiple scenarios for the circular economy policies

**Part 2: Creating functions for CE modelling in Python**

1a. Import "week\_19\_data.xlsx" as a pd.dataframe, using the following code:

|  |
| --- |
| import pandas as pd  import numpy as np  # A) Import data  A = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'A', index\_col = 0)  Y = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'Y', index\_col = 0)  f = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'F', index\_col = 0)  F\_hh = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'F\_hh', index\_col = 0) |

2. Create functions in Python

You can choose between 2.1 Python expert exercise (no code), and 2.2. Becoming a Python expert (with guiding code). In both cases, it is recommended to read section 2.3 below for further explanation about what is a Python function.

2.1. Python expert (no code)

2.1a Create a function to calculate environmental footprints using as main variable the indicators in the intensity matrix f

2.1b Create a function to calculate the environmental impacts of substitution in final demand categories considering at least 5 variables: as ori =original change in product category, k = change coefficient, sub = substituting product category, alpha= substitution weighting factor, impact= impact category (in b as CO2 or energy)

2.1c Let us assume that an economy (represented by the IOT) wants to implement a circular economy strategy for product lifetime extension. The government implements a strategy where vehicles are required to be used longer, which implies a reduction of 80% (k= -0.80) in 'Passenger motor cars' (i.e., ori) in final demand. This measure increases the services of 'Repair of motor vehicles and machine' (i.e., sub), where a reduction of ¥10 in 'Passenger motor cars' turns into ¥4 increases in 'Repair of motor vehicles and machine' (i.e., alpha = 0.4). What is the change in CO2 emissions for implementing this strategy?

2.2 Becoming a Python expert (with guiding code)

2.2a Create a function to calculate environmental footprints using as main variable the indicators in the intensity matrix f

|  |
| --- |
| def footprint(impact):  #Load the data files  A = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'A', index\_col = 0)  Y = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'Y', index\_col = 0)  f = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'f', index\_col = 0)  F\_hh = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'F\_hh', index\_col = 0)  # Calculating total impact  ID = np.identity(A.shape[1]) # Identity matrix  L = np.linalg.inv(ID-A) # L matrix  L = pd.DataFrame(L, index=A.index, columns=A.columns) # Making L-matrix a pd.DataFrame to use '.dot' in the matrix multplication below  r\_ = f.dot(L).dot(Y) # fLy calculation for all environmental indicators  r = r\_.loc[impact, :].sum() + F\_hh.loc[impact, :].sum() # Footprint calculation for the specific impact category  return r.round(1) # result rounded to one decimal |

2.2b Create a function to calculate the environmental impacts of substitution in final demand categories considering at least 5 variables: as ori =original change in product category, k = change coefficient, sub = substituting product category, alpha= substitution weighting factor, impact= impact category (in b as CO2 or energy)

|  |
| --- |
| def footprint\_alt(ori, k, sub, alpha, impact):  #Load the data files  A = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'A', index\_col = 0)  Y = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'Y', index\_col = 0)  f = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'f', index\_col = 0)  F\_hh = pd.read\_excel('week\_19\_data.xlsx', sheet\_name = 'F\_hh', index\_col = 0)  #Modelling substitution in final demand  Y\_new = Y.copy(deep=True) # copy of Y vector  Y\_new.loc[ori, :] = Y\_new.loc[ori, :].sum()\*(1 - k) # Selecting original sector and applying change coefficient  Y\_new.loc[sub, :] = (Y\_new.loc[sub, :].sum() +  (Y.loc[ori, :].sum() - Y\_new.loc[ori, :].sum())\*alpha) # Selecting substition sector and applying substitution weighting factor  #Calculating total impact  ID = np.identity(A.shape[1]) # Identity matrix  L = np.linalg.inv(ID-A) # L matrix  L = pd.DataFrame(L, index=A.index, columns=A.columns) # Making L-matrix a pd.DataFrame to use '.dot' in the matrix multplication below  r\_ = f.dot(L).dot(Y\_new) # fLy calculation with changes in final demand  r = r\_.loc[impact, :].sum() + F\_hh.loc[impact, :].sum() # Footprint calculation with changes in final demand  return r.round(1) # result rounded to one decimal |

2.3c Let us assume that an economy (represented by the IOT) wants to implement a circular economy strategy for product lifetime extension. The government implements a strategy where vehicles are required to be used longer, which implies a reduction of 80% (k= -0.80) in 'Passenger motor cars' (i.e., ori) in final demand. This measure increases the services of 'Repair of motor vehicles and machine' (i.e., sub), where a reduction of ¥10 in 'Passenger motor cars' turns into ¥4 increases in 'Repair of motor vehicles and machine' (i.e., alpha = 0.4). What is the change in CO2 emissions for implementing this strategy?

|  |
| --- |
| ori = 'Passenger motor cars' # origal sector label  sub = 'Repair of motor vehicles and machine' # substition sector  k = 0.80 # change coeffient  alpha = 0.4 # substitution weighting factor  fp = footprint('CO2') # original CO2 footprint  fp\_alt = footprint\_alt(ori, k, sub, alpha, 'CO2') # alternative CO2 footprint  change = (fp\_alt - fp) # Footprint change  change\_per = change/fp\*100 # Footprint change in percentage |

2.3 What is a Python function?

A Python function is a set of statements that take inputs, do some specific calculation to produce an output. In this way, it is possible to create a function that allows to try different inputs and analyse their outputs. A Python function consists of three main parts: input, statement, and output.

