

Week 10 Exercises: EXIOBASE introduction

Objectives

- Understand and explain the structure of EXIOBASE
- Use EXIOBASE in a Python environment
- Develop a code using Pandas

Part 1: Understanding and explaining MRIO results

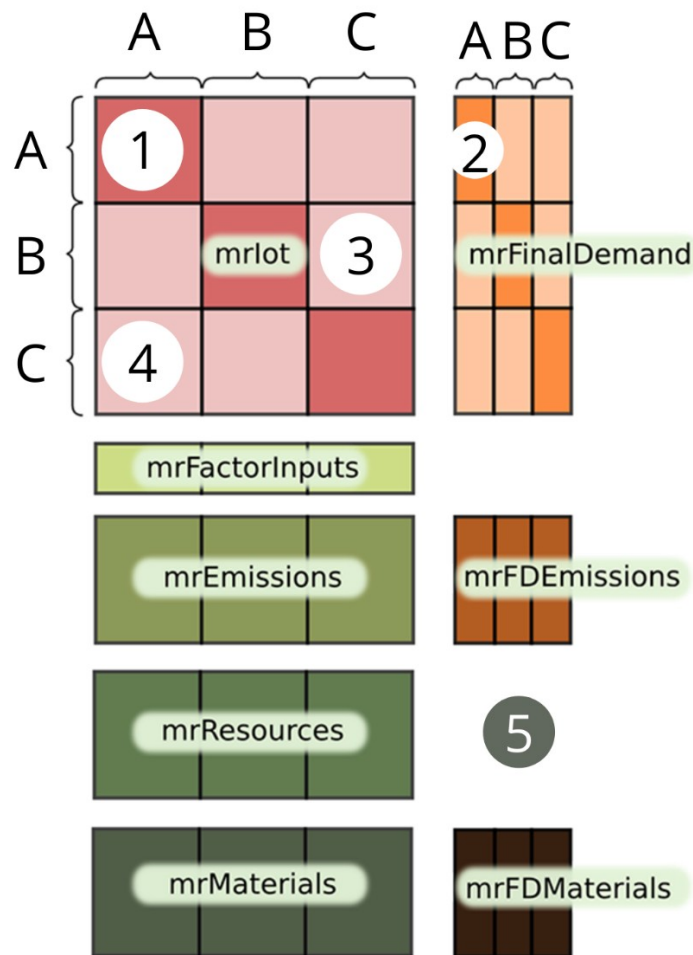


Figure 1. Structure of EXIOBASE MRIO

From figure 1:

1. What do matrices 1 and 2 represent for region A? Which are the units in matrix 1?
2. What does represent matrices 3 and 4 for region C?
3. Why is there any matrix in block 5 for all the regions?

Part 2: Python exercise with Pandas

Download EXIOBASE

1. Go to [EXIOBASE3 Data Download](#)
2. Before retrieve any data, you need to create an account
3. Download zip folder **EXIOBASE 3.4 - IOT - 2011 - pxp** . This is the latest EXIOBASE version for 2011 with a product-by-product structure
4. Unzip the data in a folder where you will store your code

Import EXIOBASE to Python using Pandas dataframe

1. In Python, import A, Y, F, and F_hh matrix using the following code lines:

```
path = 'IOT_2011_pxp' # add name of folder where data is stored
A = pd.read_csv(path + '/A.txt', sep='\t',
                index_col=[0, 1], header=[0, 1]) # Z matrix
Y = pd.read_csv(path + '/Y.txt', sep='\t',
                index_col=[0, 1], header=[0, 1]) # y matrix
F = pd.read_csv(path + '/satellite/F.txt', sep='\t',
                index_col=[0], header=[0, 1]) # satellite matrix
F_hh = pd.read_csv(path + '/satellite/F_hh.txt', sep='\t',
                   index_col=[0], header=[0, 1]) # satellite for FD matrix
```

Note: A matrix is quite large (9800x9800), thus, it might take a few seconds to run depending on your computer's memory

2. Open A and Y matrix in the Variable Explorer. Is the structure like the figure 1 from Exercise 1?

Select specific countries with Pandas

EXIOBASE uses an alpha-2 code for country labels. For example, the Netherlands is represented as 'NL' and China as 'CN'.

1. Make a variable country, for example:

```
country = 'NL'
```

2. Using [pd.DataFrame.loc](#), select the final demand of the Netherlands using the following code line:

```
y_c = Y.loc[:, country]
```

3. Calculate the sum of the Dutch final demand categories per commodity (i.e., sum columns) using Pandas

Select specific commodities with Pandas

Now, we want to know how much the Dutch final demand expenditures in office equipment and computers are. For doing so:

1. Find the label for office equipment and computers. You can search on the labels from the final demand vector calculated previously, or go to the 'products.txt' in EXIOBASE
2. Make a variable country:

```
product = '[Insert commodity label]'
```

3. Using [pd.DataFrame.loc](#), select the final demand of the Netherlands for the specific commodity using the following code lines:

```
y_c_total = y_c.sum(1) # Sum of final demand per commodity
```

```
y_c_p = y_c_total.loc[:, product]
```

4. Open vector y_c_p in the Variable Explorer. What are the top-3 direct contributors to the Dutch final demand for office equipment and computers?

Calculate L matrix

A matrix is pre-calculated in EXIOBASE. Thus, there are only a few steps to obtain the L-matrix:

1. Create an identity matrix I:

```
import numpy as np
```

```
I = np.identity(len(A))
```

2. Calculate L matrix:

```
L = np.linalg.inv(I - A)
```

Note 1: As A-matrix, the inverse is quite large. Thus, this might require a few seconds (or minutes) to run depending on your computer's memory

Note 2: Pandas package does not contain functions for calculating identity or matrix inverse. Thus, we use NumPy as in the previous lectures. Keep in mind that you might need transform NumPy arrays to Pandas dataframes (or vice versa) to make further calculations. Using NumPy or Pandas would depend on your own preferences.

Calculate total output vector (X)

1. Calculate the sum of all final demand categories (*y_total*)
2. Calculate total output vector, as:

```
X = L @ y_total
```

Notice that X is an NumPy array. If you prefer to see X as a Pandas dataframe, then:

```
df_X = pd.DataFrame(X) # Total outputs as dataframe
```

```
df_X.index = Y.index # Adding labels to rows
```