Introduction to Python for Mathematics
A Crash Course

Simon Shaw

Mathematics

August 19, 2024

Contents

Getting started

- Log In
- Start Anaconda Navigator (use search box)
- untick box, click OK, or whatever you choose.
- Press 'LAUNCH' for jupyter notebook (not jupyter lab)
- Choose 'python 3' from 'new' menu on right
- In the first cell type 2+3 followed by SHIFT-RETURN.

Did you get 5?

Leontief Input-Output Problems in Economics

Input Output Problems in Economics

Consider the tourist economy in a small resort. The major industries are

A: Accommodation — rentals, hotels, B & B's, ...

F: Food & Drink — restaurants, kiosks, pubs, take away, ...

E: Entertainment — theatre, cinema, nightclubs, ...

T: Transportation — buses, trains, taxis, ferries, ...

The turnover of each of these industries will contain cash inputs from thenselves and the others, as well as from external demand like tourists, other industrial and commercial sectors, etc.

Reference

Mathematics for Economics and Business, Ian Jacques, Prentice Hall, 4 ed. 2003.

Let's consider just A and F. Suppose that ...

Each £1 of A's turnover requires an input of 10p of its own turnover plus 30p of F's.

Each £1 of F's turnover requires an input of 20p of its own turnover plus 50p of A's.

So, assuming these proportions are constant across all turnover levels, if we want A to turn over £50,000 and F to turnover £40,000 then:

For A: £50,000 requires £5000 (0.1 of £50,000) from itself, plus £15,000 (0.3 of £50,000) from F.

For F: £40,000 requires £20,000 (0.5 of £40,000) from A, plus £8000 (0.2 of £40,000) from itself.

Get the idea? Easily generalised to more industries. Look carefully:

You can see matrices at work here. Let's figure it out...

Look at it mathematically. . .

Each £1 of A's turnover requires an input of 10p of its own turnover plus 30p of F's.

Each £1 of F's turnover requires an input of 20p of its own turnover plus 50p of A's.

Let x_1 denote A's turnover and x_2 denote F's turnover.

Also d_1 (resp. d_2) denote external demand for A (resp. F). Then:

$$\begin{aligned} x_1 &= \overbrace{0.1x_1 + 0.5x_2 + d_1}^{\text{payments to A}} \\ x_2 &= \underbrace{0.3x_1 + 0.2x_2 + d_2}_{\text{payments to F}} \end{aligned}$$

Can you see the matrix now?

Leontief's input-output model

Let x_1 denote A's turnover and x_2 denote F's turnover.

Let d_1 (resp. d_2) denote external demand for A (resp. F). Then:

$$\begin{aligned} x_1 &= 0.1x_1 + 0.5x_2 + d_1 \\ x_2 &= 0.3x_1 + 0.2x_2 + d_2 \end{aligned}$$

This can be written as $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}$ for

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad \boldsymbol{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{A} = \begin{pmatrix} 0.1 & 0.5 \\ 0.3 & 0.2 \end{pmatrix}.$$

In this model $\boldsymbol{A}$ is called the matrix of technical coeffcients, or the technology matrix. (Developed by Wassily Leontief.)

$\boldsymbol{A}$'s columns give the inputs needed for one unit of output.

Using the model

The Leontief input-output model

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}$$

can be used to answer three key questions:

1. Given total output $\boldsymbol{x}$, how much is available for demand $\boldsymbol{d}$?
   Find $\boldsymbol{d} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x}$
2. How much total output $\boldsymbol{x}$ is required to satisfy a given level of demand $\boldsymbol{d}$?
   Solve $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{d}$
3. How should output $\boldsymbol{x}$ change if demand changes by $\Delta\boldsymbol{d}$?
   Solve $(\boldsymbol{I} - \boldsymbol{A})\Delta\boldsymbol{x} = \Delta\boldsymbol{d}$

We'll do these by hand first, and then with python.
Remember that $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{\det}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Use $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}$ with $\boldsymbol{A} = \begin{pmatrix} 0.1 & 0.5 \\ 0.3 & 0.2 \end{pmatrix}$ to answer these:

1. Given total output $\boldsymbol{x} = \begin{pmatrix} 50000 \\ 40000 \end{pmatrix}$, how much is available for demand $\boldsymbol{d}$?
   Find $\boldsymbol{d} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \begin{pmatrix} 0.9 & -0.5 \\ -0.3 & 0.8 \end{pmatrix}\begin{pmatrix} 50000 \\ 40000 \end{pmatrix} = \begin{pmatrix} 25000 \\ 17000 \end{pmatrix}$
2. How much total output $\boldsymbol{x}$ is required to satisfy a given level of demand $\boldsymbol{d} = \begin{pmatrix} 35000 \\ 29000 \end{pmatrix}$?
   Solve $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{d} \Longrightarrow \boldsymbol{x} = \frac{10}{57}\begin{pmatrix} 8 & 5 \\ 3 & 9 \end{pmatrix}\begin{pmatrix} 35000 \\ 29000 \end{pmatrix} \approx \begin{pmatrix} 74561 \\ 64210 \end{pmatrix}$
3. How should output $\boldsymbol{x}$ change if demand changes by $\Delta\boldsymbol{d} = \begin{pmatrix} 2500 \\ 1900 \end{pmatrix}$?
   Solve $(\boldsymbol{I} - \boldsymbol{A})\Delta\boldsymbol{x} = \Delta\boldsymbol{d} \Longrightarrow \boldsymbol{\Delta x} = \frac{10}{57}\begin{pmatrix} 8 & 5 \\ 3 & 9 \end{pmatrix}\begin{pmatrix} 2500 \\ 1900 \end{pmatrix} \approx \begin{pmatrix} 5175 \\ 4315 \end{pmatrix}$

Now let's do this in python...

The key commands

```
import numpy as np                            # import numerical python
A = np.array([[0.1, 0.5],[0.3, 0.2]])         # our technology matrix
Id = np.eye(2)                                # 2 by 2 identity matrix
d = np.array([[35000],[29000]])               # demand vector d
print(np.linalg.solve(Id-A, d))               # solve (I-A) x = d for x
Dd = np.array([[2500],[1900]])                # change in demand vector, Dd
print(np.linalg.solve(Id-A, Dd))              # solve (I-A) Dx = Dd for Dx
```

Note you can start a new cell whenever you like.

Do so frequently.

Exercise - use python

Given the technology matrix $\boldsymbol{A} = \begin{pmatrix} 0.3 & 0.2 \\ 0.1 & 0.6 \end{pmatrix} \ldots$

1. Given total output $\boldsymbol{x} = \begin{pmatrix} 70000 \\ 90000 \end{pmatrix}$, how much is available for demand $\boldsymbol{d}$?

   Find $\boldsymbol{d} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x}$

2. How much total output $\boldsymbol{x}$ is required to satisfy a given level of demand $\boldsymbol{d} = \begin{pmatrix} 52000 \\ 48000 \end{pmatrix}$?

   Solve $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{d}$

3. How should output $\boldsymbol{x}$ change if demand changes by $\Delta\boldsymbol{d} = \begin{pmatrix} 5200 \\ 4100 \end{pmatrix}$?

   Solve $(\boldsymbol{I} - \boldsymbol{A})\Delta\boldsymbol{x} = \Delta\boldsymbol{d}$

Exercise

We started with a tourist economy in a small resort with the major industries: A (Accommodation), F (Food & Drink), E (Entertainment) and T (Transportation).

The technology matrix for this economy is

$$\boldsymbol{A} = \begin{pmatrix} 0.15 & 0.12 & 0.05 & 0.03 \\ 0.17 & 0.16 & 0.04 & 0.04 \\ 0.03 & 0.08 & 0.18 & 0.22 \\ 0.07 & 0.18 & 0.03 & 0.19 \end{pmatrix}$$

1. How much is left for demand $\boldsymbol{d}$ with a total output $\boldsymbol{x} = \begin{pmatrix} 89000, & 55000, & 47000, & 76000 \end{pmatrix}^T$?
2. How much total output $\boldsymbol{x}$ is needed for demand $\boldsymbol{d} = \begin{pmatrix} 55000, & 24000, & 18000, & 40000 \end{pmatrix}^T$?
3. How should output $\boldsymbol{x}$ change if demand changes by $\Delta\boldsymbol{d} = \begin{pmatrix} -5000, & 350, & 2300, & -500 \end{pmatrix}^T$?

Finishing up

There is much much much more to python, and numpy.

We're just scratching the surface

As is the nature of a crash course

Let's look now at how to plot graphs

2D Plotting

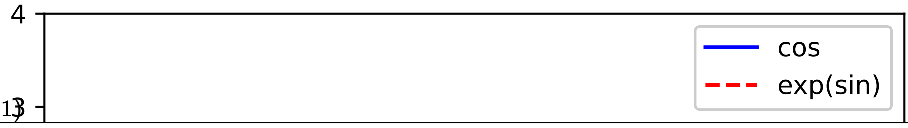
Section 1: Anaconda and Jupyter · Section 2: I/O Problems · Section 3: 2D plots · Section 4: Anonymity

Plotting in 2D

We jump straight in.

Plot $\cos(2\pi x)$ in solid blue and $\exp(\sin(4\pi x))$ in dashed red for $x \in [-1, 5]$

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-1,5,0.01)
y1, y2 = np.cos(2*np.pi*x), np.exp(np.sin(4*np.pi*x))
plt.plot(x,y1, 'b-')
plt.plot(x,y2, 'r--')
plt.axis([-2, 6, -2, 4])
plt.legend(['cos', 'exp(sin)'])
plt.xlabel(r'$x_1$'); plt.ylabel('$y_1$ and $y_2$')
plt.savefig('my2Dplot.png', dpi=600)
plt.savefig('my2Dplot.eps', dpi=600)
```



© Simon Shaw (2021)

Exercise

In python. . .

Plot $2^{3\sin(3\pi x)}$ in solid dash-dot blue and $\ln\big(1.2 + \sin(3\pi x)\big)$ in dotted red for $x \in [-4, 3]$

Hint: for the line-styles use

```
plt.plot(x,y1, 'b-.')
plt.plot(x,y2, 'r:')
```


How Anonymous is Anonymised Data?

How anonymous is anonymized data?

A university collects answers to personal questions from all of its students.

Each student's answer has their name, date of birth, gender and department.

On average a department has 250 students in each year.

We assume the UK setup where students attend for three years.

The names are erased: how anonymous are the resulting data?

Here's part of the dataset

NAME	D.O.B.	GENDER	DEPARTMENT
⋮	⋮	⋮	⋮
Ringo Starr	26/07/43	M	Music
Al Gebra	12/08/05	F	Maths
Sandie Shaw	16/02/38	F	Puppetry
Michael Mouse	17/04/92	F	Computing
Mr Pink	4/12/56	M	Criminology
L.O. Gear	11/9/23	F	Engineering
Donkey Kong	23/10/73	M	Video Games
⋮	⋮	⋮	⋮

The names are erased — can one line dentify the person? Is this enough information to identify the person?

How anonymous is a dataset like this?

Let's simulate

We assume that all students are born within a three year window, and that each department has 750 students across its three years.

- There are $365 \times 3 = 1095$ possible birthdays
- For each there are at least 2 possible genders
- So, for a given department, there are $d = 1095 \times 2 = 2190$ possible entries among $N = 750$ students.
- Can you see why anonymity might not be assured?

Think about a line of $N = 2190$ empty buckets. Now throw $N = 750$ balls at random into the buckets.

Most will stay empty. Some will have just one ball — the 'loners'.

The proportion of N having just one ball estimates the probabilty that line of anonymised data occurs just once in the department.

Planning the code

We are going to generate a list of $d = 2190$ zeros.

We'll then generate $N = 750$ random integers $z \in \{1, 2, \ldots, 2190\}$.

For each z we'll add one to the z^{th} item in the list, d.

In the end, the n^{th} item in the list, d, tells us how many students share that same data.

We want to find the 'loners' — the buckets with only one item in them.

Background and Exercise

- There is about a 70% probability that a student can be identified from this anonymised data.
- A 'near exact' solution is $\exp(-N/d) \approx 71\%$.

My main reference is John D Cook:

`www.johndcook.com/blog/2018/12/07/simulating-zipcode-sex-birthdate/`

Which itself references the paper *Only You, Your Doctor, and Many Others May Know*, Latanya Sweeney: `https://techscience.org/a/2015092903/`

Exercise: What is the probability that a line of anonymised data can be narrowed down to at most two students? Or three? Or four?

The End

That's it!

Thanks for listening

There's lots more to learn — as ever!

Good Luck!