動態マクロ経済学

Week 4

佐藤健治

sato@eco.osakafu-u.ac.jp

2020/5/29

準備運動：IPython を起動してください

- いつものインポート文

```
import numpy as np
```

- np.arange() は等差数列

```
A = np.arange(24)
A
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9,
       10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
       20, 21, 22, 23])
```

配列（ベクトル）の shape

- shape を確認する習慣を身に着けよう。
  - (24,) は 1 個の数字からなる「タプル」。
  - 24 個の数字が 1 列に並んだ配列，ということ。

```
A.shape
```

```
(24,)
```

shape の変更

- shape を上書きすると配列の形を変形できる。
  - 次のコードの結果はどうなる？

```
A.shape = (3, 8)
```

配列（行列）の shape

- shape = (3, 8) は 3×8 行列を作る。
  - 行方向に 3 つ，列方向に 8 つ数字が並ぶ。
  - 数字の並びは「行指向」（行方向に数字を埋めていく）

```
A
```

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7],
       [ 8,  9, 10, 11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20, 21, 22, 23]])
```

- **「次元」** = shape に並ぶ数字の個数（ここでは 2）
  - 各数字はその「軸 (axis)」のサイズ
  - (axis0, axis1)

もっと大きな次元

```
A.shape = (2, 3, 4)
A
```

```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],

       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

- 2 個の 3×4 行列

軸ごとの sum()

行和，列和の計算で使う。

```
A.shape = (4, 6)
A.sum()             # (4, 6) -> ()
```

```
276
```

```
A.sum(axis=0)    # (4, 6) -> (6,)
```

```
array([36, 40, 44, 48, 52, 56])
```

```
A.sum(axis=1)    # (4, 6) -> (4,)
```

```
array([ 15,  51,  87, 123])
```

for ループ

- 同じ処理を繰り返し実行するときに使う。
  - Python の for は「できるだけ使うな」と書いている。
- どうしても必要なとき以外は for を使わず NumPy のベクトル計算を使う。
- どうしても必要なとき？
  - 「漸化式」で定義される数字の計算にはループが必要
  - 動学モデルのほとんど

基本の for

```
for i in range(5):
    print(i ** 3)
```

```
0
1
8
27
64
```

```
for␣i␣in␣[0, 1, 2]:
␣␣␣␣f()

g()
```

リストのようなオブジェクトを in の後ろに書く。ループの各回が処理されるとき，in の前に書いた変数（i）には，in の後ろのオブジェクトの各要素が順に代入される

コロンは for 文の本体（繰り返される処理）が始まる合図

繰り返し実行するコードは半角スペース4つ右にインデントする

繰り返し実行しないコードはインデントの解除で判別される

- range(5) は [0, 1, 2, 3, 4] と同じ数字の並びを表す。
- range(1, 8) はどんな数列か？

本日の目標

- テキスト 第 3 章
- やること
  - GDP の構成
  - 成長率
  - 成長率と寄与度

実質 GDP，成長率の計算は価格指数の計算とほぼ同じ。今回は for ループを使ってもう少しスマートにやってみる。

GDP

- $\boldsymbol{p}_t = t$ 期の価格ベクトル
- $\boldsymbol{x}_t = t$ 期の数量ベクトル（すべての最終財）
- 名目 GDP

$$Y_t = \boldsymbol{p}_t \cdot \boldsymbol{x}_t$$

実質 GDP

- $t = 0$ を基準年とする。
- t 期の実質 GDP（連鎖方式） y_t を次のように定義する。

$$
\begin{aligned}
y_0 &= Y_0 \\
y_t &= y_{t-1} \times \frac{\boldsymbol{p}_{t-1} \cdot \boldsymbol{x}_t}{\boldsymbol{p}_{t-1} \cdot \boldsymbol{x}_{t-1}}
\end{aligned}
$$

  - 実質 GDP はラスパイレス式の数量指数である。

GDP デフレーター

- t 期の GDP デフレータ（インプリシット・デフレーター）DFL_t は次のように定義される。
$$\mathrm{DFL}_t = \frac{Y_t}{y_t}$$
  - これは一種の価格指数である。
  - 実際，DFL_t は前回説明したパーシェ式の連鎖価格指数と一致する。

練習問題

DFL_t がパーシェ式価格指数（連鎖方式）の定義式を満たすことを証明しなさい。

例

産業 A, B, C それぞれの平均価格と取引数量を以下のように表す。

価格	A	B	C
2000 年	100	100	100
2001 年	101	99	103
2002 年	100	98	104
2003 年	99	99	106

数量	A	B	C
2000 年	1000	2000	500
2001 年	980	1980	510
2002 年	1010	1990	520
2003 年	1005	2005	530

NumPy

```
price = np.array([[100, 100, 100],
                  [101, 99, 103],
                  [100, 98, 104],
                  [99, 97, 106.]])

quantity = np.array([[1000, 2000, 500],
                     [980, 1980, 510],
                     [1010, 1990, 520],
                     [1005, 2005, 530.]])
```

取引額と名目 GDP

```
price * quantity
```

```
array([[100000., 200000.,  50000.],
       [ 98980., 196020.,  52530.],
       [101000., 195020.,  54080.],
       [ 99495., 194485.,  56180.]])
```

各年度の名目 GDP を計算するにはどうすればよいか？

```
NGDP
```

```
array([350000., 347530., 350100., 350160.])
```

実質 GDP

```
GDP = np.empty(price.shape[0])  # 行数 = 年数

GDP[0] = np.sum(price[0, :] * quantity[0, :])
t = 1
GDP[t] = GDP[t-1] * (np.sum(price[t-1, :] * quantity[t, :])
                    / np.sum(price[t-1, :] * quantity[t-1, :]))
t = 2
# .... continue
```

……としてもいいのだけど，年数が増えると大変なので for ループを使う。

実質 GDP（つづき）

```
T = price.shape[0]  # 行数 = 年数
GDP = np.empty(T)

GDP[0] = np.sum(price[0, :] * quantity[0, :])
for t in range(1, T):
    GDP[t] = (GDP[t-1]
              * (np.sum(price[t-1, :] * quantity[t, :])
               / np.sum(price[t-1, :] * quantity[t-1, :])))

GDP
```

```
array([350000.        , 347000.        ,
       352042.2985066 , 354063.44966341])
```

GDP の分解

- GDP（国内総生産）の支出面は次のように加法的に分解される

$$Y_t = C_t + I_t + G_t + NX_t$$

  - C_t: 消費
  - I_t : 粗投資
  - G_t : 政府購入
  - NX_t : 純輸出 = 輸出 (EX_t) から輸入 (IM_t) を引いたもの。
- GDP の成長において，どの項目の成長が GDP の成長にどれだけ寄与したか？
  - 話を簡単にするために，名目 GDP の成長への寄与を計算しよう。

寄与度

$$Y_t = C_t + I_t + G_t + NX_t$$

$$\downarrow$$

$$Y_{t+1} = C_{t+1} + I_{t+1} + G_{t+1} + NX_{t+1}$$

例えば，C の寄与度は

$$\frac{\Delta C_{t+1}}{Y_t} = \frac{C_{t+1} - C_t}{Y_t}$$

寄与度（続き）

次の性質に注意する。

$$\begin{aligned}\frac{\Delta Y_{t+1}}{Y_t} &= \frac{\Delta C_{t+1}}{Y_t} + \frac{\Delta I_{t+1}}{Y_t} + \frac{\Delta G_{t+1}}{Y_t} + \frac{\Delta NX_{t+1}}{Y_t} \\ &= \frac{\Delta C_{t+1}}{Y_t} + \frac{\Delta I_{t+1}}{Y_t} + \frac{\Delta G_{t+1}}{Y_t} + \frac{\Delta EX_{t+1}}{Y_t} + \frac{\Delta(-IM_{t+1})}{Y_t}\end{aligned}$$

- 成長率を項目ごとに分解した。
- 控除項目は負数にしておくとよい。

例

各支出項目の寄与度を計算しなさい。

	C	I	G	EX	IM
2017 年度	298.88	101.56	132.33	91.43	92.62
2018 年度	299.05	102.28	139.39	92.87	94.62

```
expenditure = np.array([[298.88, 101.56, 132.33, 91.43, -92.62],
                        [299.05, 102.28, 139.39, 92.87, -94.62]])
```

差分と GDP

- 列ごとの差分計算が必要。次のようにする。

```
diff = np.diff(expenditure, prepend=np.nan, axis=0)
diff
```

```
array([[  nan,   nan,   nan,   nan,   nan],
       [ 0.17,  0.72,  7.06,  1.44, -2.  ]])
```

- GDP は横方向に足すだけ。

```
gdp = expenditure.sum(axis=1)
gdp
```

```
array([531.58, 538.97])
```

寄与度

- GDP の準備：ちょっと小細工がすぎる気がするけどデータが大きくなっても大丈夫なように。

```
gdp_shift = np.r_[np.nan, gdp[:-1]]
gdp_shift.shape = (2, 1)
gdp_shift
```

```
array([[    nan],
       [531.58]])
```

- 寄与度は次のように計算

```
contribution = diff / gdp_shift
```

確認問題

練習問題

寄与度の総和が GDP の変化率に（ほぼ）一致することを確認しなさい。
2 つの実数が近いことを確認するには，`np.allclose()` を用いる。