

PythonとQuantEcon.pyへの 入門

Syzygyをつかって

MKobayashi23m

Table of Contents

- 環境構築をせず, ブラウザからPythonを使う-> Syzygy
- QuantEcon.pyの使い方- ナッシュ均衡の計算まで
- Appendix:ローカルにPythonのための環境構築を行う->Anaconda

Table of Contents

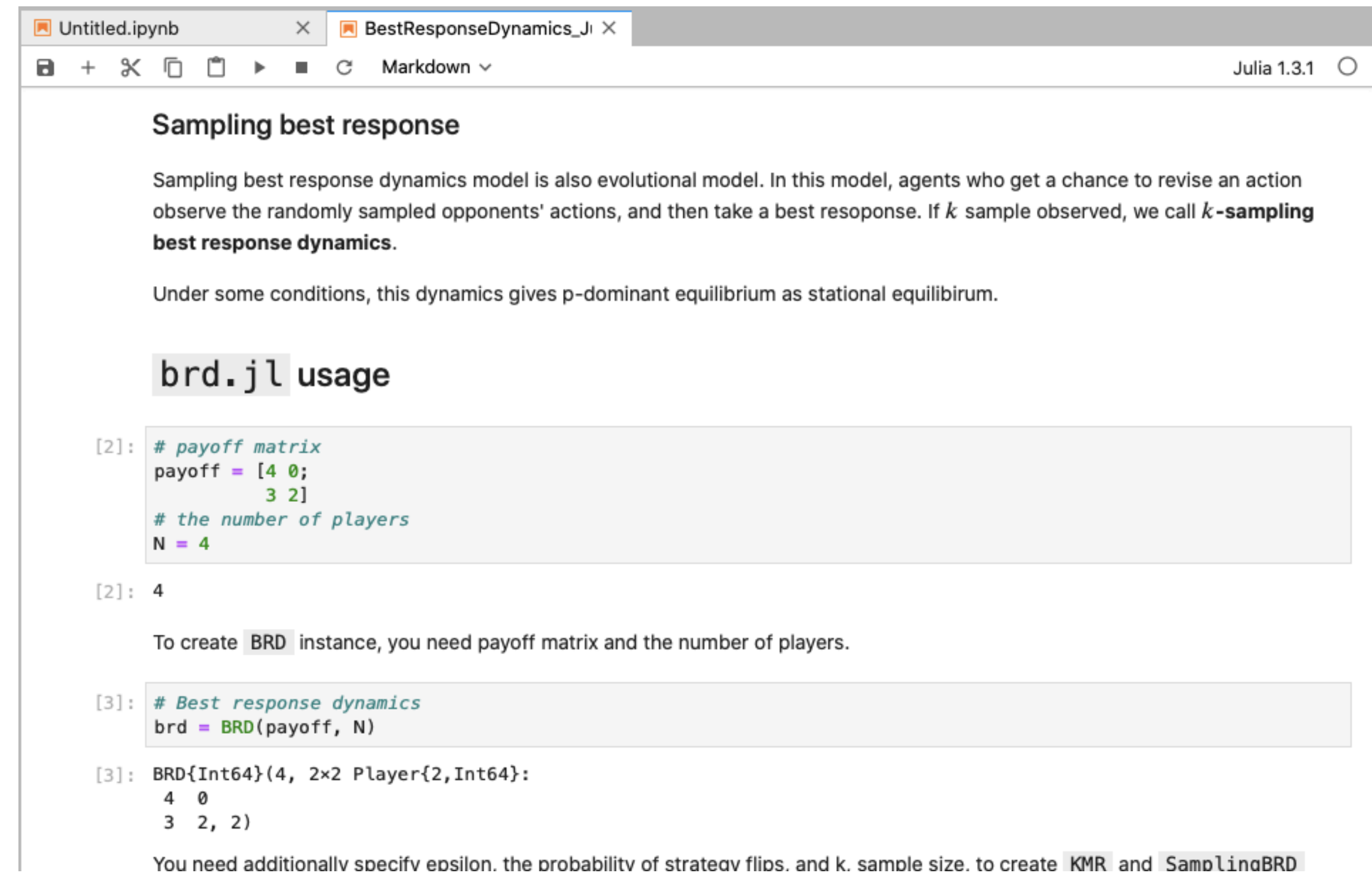
- 環境構築をせず, ブラウザからPythonを使う-> Syzygy
- QuantEcon.pyの使い方- ナッシュ均衡の計算まで
- Appendix:ローカルにPythonのための環境構築を行う->Anaconda

Pythonを使い始めるには

- ・ 2つのやり方- 自分のPC(ローカル)に環境構築を行う or webブラウザ上(クラウド)で行う
- ・ ローカルに環境構築を行う: PCを持っている人はこっちの方が良い
- ・ クラウドサービスの利用: PCがなくてスマホやタブレットからしかPythonが使えない人

Jupyter Notebook

- 文章・グラフ・コードを一括で管理するipynbファイルを開覧, 編集, 実行する
- エディタ兼ノートという感じ
- 今回使用する



The screenshot shows a Jupyter Notebook window with two tabs: 'Untitled.ipynb' and 'BestResponseDynamics_Jl'. The active tab is 'BestResponseDynamics_Jl'. The notebook content includes a title 'Sampling best response', a paragraph explaining the model, a code cell for defining the payoff matrix and number of players, and a cell for creating a BRD instance.

```
Sampling best response

Sampling best response dynamics model is also evolutionary model. In this model, agents who get a chance to revise an action observe the randomly sampled opponents' actions, and then take a best response. If  $k$  sample observed, we call  $k$ -sampling best response dynamics.

Under some conditions, this dynamics gives p-dominant equilibrium as stationary equilibrium.

brd.jl usage

[2]: # payoff matrix
payoff = [4 0;
          3 2]
# the number of players
N = 4

[2]: 4

To create BRD instance, you need payoff matrix and the number of players.

[3]: # Best response dynamics
brd = BRD(payoff, N)

[3]: BRD{Int64}{4, 2x2 Player{2, Int64}}:
      4  0
      3  2, 2)

You need additionally specify epsilon, the probability of strategy flips, and k, sample size, to create KMR and SamplingBRD
```

Syzygy

- ・ 環境構築なしでブラウザからJupyter Notebookが操作できる！
- ・ 使用方法
 1. <https://pims.syzygy.ca> の赤い家のアイコンをクリックしてgoogleアカウントでログイン
 2. 右上の'New'ボタンを押してPython 3を選択
 3. You are ready!

Google Colaboratory

- Syzygyのサーバが落ちた場合のバックアップ
- 使用方法
 1. <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja> に入ると, チュートリアル用ノートブックが開く
 2. 自分のグーグルアカウントでログイン
 3. 今回はチュートリアルを飛ばして, 左上の'ファイル'タブから'ノートブックを新規作成'を選択

Microsoft Azure Notebooks

- Syzygyのサーバが落ちた場合のバックアップ2
- 使用方法
 1. <https://notebooks.azure.com> の右上の'Sign In'から入る(アカウントがなければ登録)
 2. 'My Profile' の 'My Projects'下, 'create one now?' をクリックして, 新規プロジェクトを作成
 3. 新しく作ったプロジェクトの画面で'+ 'タブから'Notebook'を選び, 言語設定で'Python3.x'を選ぶ(最新バージョン)
 4. 作成したipynbファイルをクリックして選ぶ

Table of Contents

- 環境構築をせず, ブラウザからPythonを使う->Syzygy
- QuantEcon.pyの使い方- ナッシュ均衡の計算まで
- Appendix:ローカルにPythonのための環境構築を行う->Anaconda

QuantEcon.pyとは？

- GitHub上で開発されている, Quantitive economics のためのパッケージ
- 開発ページ: <https://github.com/QuantEcon/QuantEcon.py>
- 計量経済学に出てくる最尤法や, 繰り返しゲームの均衡計算, 動的計画法などをサポートしている
- 解説: <https://quantecon.org/notebooks/>
- 今回はインストールからナッシュ均衡の計算まで



(参考)quanteconパッケージのインストール

- Jupyter Notebook上で新しく作ったipynbファイルを開いて, セルに以下を入力, Shiftキーを押しながらEnterで実行

```
!pip install quantecon
```

- quanteconがインストールされる
- Syzygyにはquanteconがプリインストールされているので必要なし

quanteconを使ってみる

- ・ 尾山先生が書いているnotebookを参考に, quanteconを使ってみよう

- ・ https://nbviewer.jupyter.org/github/QuantEcon/QuantEcon.notebooks/blob/master/game_theory_py.ipynb

- ・ まず, さっき作ったipynbファイルのセルに以下のコードを入力

```
import quantecon.game_theory as gt
```

- ・ Shiftキーを押しながらEnterキーを押して実行. game_theory モジュールを読み込む. (*が数字に変わったら完了)

復習: 標準形ゲーム

- 標準形ゲーム G は以下のような要素の組として定義されるのだった

$$G = \langle I, (A_i)_{i \in I}, (u_i)_{i \in I} \rangle$$

ただし

I : プレイヤーの集合

A_i : プレイヤー $i \in I$ の純戦略の集合

$u_i : A \rightarrow \mathbb{R} \ i \in I$ の利得関数 (なお $A = \times_{i \in I} A_i$)

標準形ゲームを作成する

- 以下の行列で表現されるジャンケンゲームを例に、標準形ゲームの作成を学ぶ

P1\P2	グー	チョキ	パー
グー	0,0	1,-1	-1,1
チョキ	-1,1	0,0	1,-1
パー	1,-1	-1,1	0,0

標準形ゲームを作成する

1. 利得行列を作成する (対称2人ゲームならプレイヤー1の利得のみで良い)

```
RPS_matrix = [[ 0, -1, 1],  
               [ 1,  0, -1],  
               [-1,  1,  0]]
```

2. 利得行列を引数として, NormalFormGame インスタンスを作成する

```
g_RPS = gt.NormalFormGame(RPS_matrix)
```

print(g_RPS) で作成した NormalFormGame インスタンスを表示

ナッシュ均衡の計算

- ・ナッシュ均衡の計算アルゴリズムは複数存在する
 - ・尾山先生のノート参照(URLは前述)
- ・今回は, **support enumeration** という方法を使って, 混合戦略ナッシュ均衡を求める
- ・以下のコードを実行すれば良い

```
gt.support_enumeration(g_RPS)
```

- ・プレイヤー1,2が各純戦略に割り振る確率が, グー, チョキ, パーの順に表示される

演習問題 次のゲームの混合戦略ナッシュ均衡を求めよ

- 非対称ゲームであることに注意

P1\P2	L	R
T	0,0	1,2
B	3,1	0,0

P1\P2	R	P	S
R	0,0	-6,1	3,-1
P	6,-1	0,0	-6,1
S	-3,1	6,-1	0,0

演習問題 次のゲームの混合戦略ナッシュ均衡を求めよ

- ・ ヒント: 非対称ゲームの場合, P1 と P2, 両方の利得を入力する必要がある

```
asy_matrix = [[(0,0),(1,2)],  
               [(3,1),(0,0)]]
```

- ・ 括弧の使い分けに注意

おわりに

- ・今日のまとめ
 - ・Syzygyを使ってPythonを使ったナッシュ均衡の計算を学んだ
- ・今後の勉強に関して
 - ・Quantecon.pyを使って卒論や宿題で必要な計算, あるいは検算を試みては
 - ・PCがある人は是非Anacondaのインストールを！

Table of Contents

- 環境構築をせず, ブラウザからPythonを使う-> Syzygy
- QuantEcon.pyの使い方- インストールからナッシュ均衡の計算まで
- Appendix:ローカルにPythonのための環境構築を行う->Anaconda

ローカルに環境構築を行う- Anaconda

- Anacondaを使ったやり方が最も簡単
- ECCSにはすでにインストールされている.
- インストールの手順
 1. <https://www.anaconda.com/distribution/#download-section> に行き, 'Python 3.x version' の下の 'Download' ボタンをクリックしてダウンロード (今は多分3.7)
 2. ダウンロードしたpkg(mac. windowsならexeか) ファイルを実行してインストーラを起動, インストール (設定はそのまま基本的に良い)

Anaconda Navigator

- Anacondaインストール後, 追加されたアプリケーション, ‘Anaconda-Navigator’を実行
- Jupyter Notebookを使う… Jupyter lab か Notebook
- IDE(統合開発環境)… Spyder(使ったことはない, 立ち上げが遅い?)
 - 別にVisual Studio CodeとかAtomを入れたほうがいいのかも