DSGE モデルの推定 2

DSGE モデルのコマンド構文

本セクションでは dsge コマンドを利用して DSGE モデルを推定するためのコマンドの用法を具体的に説明します. [1]DSGE モデルは連立方程式の形で記述し, その際に利用する変数を大きく分けて 3 種類に区別しますので, その事に留意して以下の内容を読み進めてください.

intro 2 の構成は次の通りです.

- dsge コマンド構文の概略
- 連立方程式の記述
- 制御変数
- 状態変数とショック
- 制御変数の将来の期待値
- 置換表現を用いてパラメータを推定する

dsge コマンド構文の概略

初回の DSGE 入門 (intro 1) をまだ読んでいない方は, 先にそちらに目を通してください. 特に後半に示した“DSGE モデルの構造形と誘導形”の項目は重要ですので, ご一読ください. ここでは例として次に示すように連立方程式が誘導形として与えられていることを前提として話を進めます.

$$
\begin{aligned}
p_t &= \beta\,\mathrm{E}_t\left(p_{t+1}\right) + \kappa y_t \\
y_t &= \mathrm{E}_t\left(y_{t+1}\right) - \left(r_t - \mathrm{E}_t\left(p_{t+1}\right) - \rho z_t\right) \\
\beta r_t &= p_t + \beta u_t \\
z_{t+1} &= \rho z_t + \epsilon_{t+1} \\
u_{t+1} &= \delta u_t + \xi_{t+1}
\end{aligned}
$$

このモデルは次のように入力します.

```
. dsge  (p          = {beta}*E(F.p) + {kappa}*y)                    ///
        (y          = E(F.y) -(r - E(F.p) - {rho}*z), unobserved)   ///
        ({beta}*r   = p + {beta}*u)                                 ///
        (F.z        = {rho}*z, state)                               ///
        (F.u        = {delta}*u, state)
```

[1]Stata 15 の DSGE マニュアル intro 2 の要約です。

数式とコマンドを比べてみると, 直感的に数式と同じだと簡単に分かります. したがって, ここからはユーザが実際に自分の数式を記述するときに気をつけるべき事柄を説明します.

intro 1 では DSGE モデルで利用する変数を制御 (control)[2]変数 (観測可能な変数), 状態変数 (観測不可能な変数), ショックの 3 つに分けました. したがって, この 3 つの種類の変数の記述方法を具体的に解説します.

連立方程式の記述

DSGE モデルでは各制御変数と各状態変数について, それぞれ対応する方程式を記述します. その式は約束事としてカッコで囲みます. ここでは dsge コマンドの基本的な書き方は次のようになります.

```
. dsge   (eq1)                    ///
         (eq2)                    ///
         ...,...
```

ここで *eq1* と *eq2* にはモデルを構成する推定式を入力します. もちろん, 推定式として 3 つ以上の式を利用できます.

推定式は基本的にカッコで囲みます.

```
term = term [ + term [ + term [:  :  :  ]]]
```

term とある所には変数名を入力します. 変数名の前にはパラメータ, または, パラメータの非線形な組み合わせを記述します. 適切な記述例を次に示します.

```
(y = {kappa}*z)
```

または,

```
(y = {kappa}*z + {kappa}*{beta}*x)
```

または,

```
(1/{beta}*y = {gamma}*z + x)
```

記述方法に関する詳細は後述します. 直感的にも分かるように DSGE モデルでの推定式の記述方法は普通の数式とほぼ同じような形式となっています. ただし, パラメータだけは大カッコ {} で囲みます. Stata の他のコマンドで推定式を入力する場合とは, 若干, 構文が異なりますので注意してください. dsge コマンドの場合, 他のコマンドに比べ, 特殊な形式で推定式を入力します.

dsge コマンドによる推定式の記述方法を説明する前に, ここでは各種変数の記述方法を解説します.

制御変数

制御変数は他の制御変数, 制御変数の将来の値の期待値, 状態変数の関数としてモデル化します. ただし, 制御変数を表現する式でショックは利用できません.

先のセクションで示したように基本的な dsge コマンドの書式は次の通りです.

```
. dsge   (eq1)                    ///
         (eq2)                    ///
         ...,...
```

[2]control variable を intro 1 ではコントロール変数と訳しましたが、これは経済学でいう所のコントロール変数とは異なりますので、今回から制御変数と呼び方を変えます。

eq1 として観測可能な制御変数 y を z_t と x_t の関数としてモデル化する場合, dsge コマンドは次のようになります.

```
. dsge  (y=pexp*z+pexp*x)                    ///
        (eq2)                                ///
        ...,...
```

ここで *pexp* にはパラメータの非線形式を利用することも可能です.

制御変数には観測できない変数も利用できます. この例のようにオプションなしで記述した場合, y は観測可能であることになります. y が観測できない変数である場合は次に示すようにオプション unobserved を追加します.

```
(y=pexp*z+pexp*x,unobserved)
```

モデルで利用する制御変数は, 必ず一度は左辺に誘導形の被説明変数として記述する必要があります.

状態変数とショック

状態変数をモデル化する場合, 左辺には状態変数の 1 期先の変数を被説明変数として記述します. そして右辺には状態変数, 制御変数, 制御変数の将来の期待値, そしてショックを利用します.

状態変数を表現する式ではオプション state を利用します.

先の例を用いて, *eq2* で状態変数 x を利用する場合, x_{t+1} を x_t の関数として表現します. dsge コマンドは次のようになります.

```
. dsge  (y=pexp*z+pexp*x)                    ///
        (F.x=pexp*x,state)                   ///
        ...,...
```

x の一期先を示す変数を表現する場合はリードオペレータ F. を利用して F.x とします. オペレータの詳細は [U] 11.4.4 Time series varlists を参照してください. ただし, 時系列の演算子がすべて DSGE モデルで利用できる訳ではありませんので, ご注意ください. リード項として推定式で利用できるのは一期先の値までです. したがって, F.x のところを F2.x のように変更することはできません. しかし, この制約によって, 表現する推定式に限界があると考えるのは誤りです. 詳細は [DSGE] intro 4c を参照してください.

デフォルトでは, 状態変数の推定式は観測できないショックを含みます. もちろん, 状態変数にショックが必須であるという訳ではありません. DSGE モデルの連立式において, ショックの数と, 観測可能な制御変数の数は一致させる必要があります. 例えば, x_{t+1} の式でショックを利用したくない時は次に示すように noshock オプションを利用します.

```
(F.x = pexp*x, state noshock)
```

先にも触れましたが, 状態変数の 1 期先の変数を被説明変数とする式を, 状態変数の個数分だけ記述する必要があります.

制御変数の将来の期待値

制御変数の一期先の値を示すリード項の期待値は制御変数の式, または, 状態変数の式で利用可能です. t 時点での期待値を $\mathrm{E}_t\,()$ で表現することにします. そうすると, t 時点の y_{t+1} の期待値は $\mathrm{E}_t\,(y_{t+1})$ と表現できます. dsge のコマンド文では E(F.y) となります. もちろん, E() は F.y の期待値であり, これは y のリー

ド項の期待値を意味します．リードを伴う E() による期待値は通常の変数と同じ要領でモデルで利用できます．

例えば，y_t が $\mathrm{E}_t\left(y_{t+1}\right), z_t, x_t$ の関数である場合，先の dsge コマンドは次のようになります．

```
. dsge  (y=pexp*E(F.y)+pexp*z+pexp*x)          ///
        (F.x=pexp*x,state)                     ///
        ...,...
```

厳密に言えば，E() による期待値は制御変数の一期先の予測値になります．しかし，$t+2$ による 2 期先の期待値を E(F2.y) として表現することはできません．しかし，実際のモデルでは他の方法を用いて，二期以上先の期待値も利用できます．2 期先以上の制御変数の期待値の記述方法に関する詳細は [DSGE] intro 4d を参照してください．

置換表現を用いてパラメータを指定する

DSGE モデルで利用する変数の種類と記述方法についておわかりいただけたでしょうか．ここでは推定したいパラメータを設定する方法を説明します．

ここでもう一度，推定式の基本的なモデルを示しておきます．

term = *term* [+ *term* [+ *term* [: : :]]]

ここで *term* には変数名を用います．変数名の前にはパラメータや非線形式を記述することができます．また，右辺の *term* にはパラメータに関する仕様を追記できます．

観測可能な制御変数は既にデータセットに用意されています．一方，観測できない制御変数や状態変数はデータセットには存在していないはずです．

ここで，*term* にスカラー置換表現という特殊な置換表現を利用します．スカラー置換表現は大カッコ { } でパラメータを囲みます．そして，モデルを線形または，非線形の形で入力します．ただし，*term* には一つの変数だけを利用します．そして各 *term* は各式でスカラー置換表現を伴って線形に結合しなければなりません．

仮に，y が次のような式に従うとします．

$$y_t = \beta\,\mathrm{E}_t\left(y_{t+1}\right) + \kappa z_t + \gamma x_t$$

この時，dsge コマンドは次のようになります．

```
. dsge  (y={beta}*E(F.y)+{kappa}*z+{gamma}*x)   ///
        (F.x=pexp*x,state)                      ///
        ...,...
```

y の式においてパラメータと線形関係を維持する必要はありません．例えば，次のようなモデルも推定可能です．

$$y_t = (1/\beta)\,\mathrm{E}_t\left(y_{t+1}\right) + \kappa z_t + (\gamma/\beta)\,x_t$$

コマンドは次のようになります．

```
. dsge  (y=1/{beta}*E(F.y)+{kappa}*z+({gamma}/{beta})*x)   ///
        (F.x=pexp*x,state)                                 ///
        ...,...
```

パラメータを左辺に利用することもできます. 例えば, y について次のような式を考えます.

$$(1/\beta)\, y_t = \mathrm{E}_t\,(y_{t+1}) + \kappa z_t + (\gamma/\beta)\, x_t$$

dsge コマンドは次のようになります.

```
. dsge  ((1/{beta})*y=E(F.y)+{kappa}*z+({gamma}/{beta})*x)          ///
        (F.x=exp*x,state)                                          ///
        ...,...
```

最後に推定式を記述する上で便利な機能を紹介しておきます. 線形モデルにおける共通の係数は, 次に示すようにカッコの外に共通項として記述することもできます.

$$y_t = (1/\beta)\,\mathrm{E}_t\,(y_{t+1}) + (\gamma/\beta)\, x_t + \kappa z_t$$

これは,

$$y_t = (1/\beta)\,\{\mathrm{E}_t\,(y_{t+1}) + \gamma x_t\} + \kappa z_t$$

これを dsge コマンドで記述すると次のようになります.

```
(y = (1/{beta}) * (E(F.y) + {gamma}*x) + {kappa}*z)
```

dsge コマンドの基本的な文法はここに説明したとおりです.

以上の文法を利用した古典的な DSGE モデルの推定例は [DSGE] intro 3 で紹介します. 制御変数の推定式でショックや, 状態変数のラグ項を利用する場合は [DSGE] intro 4 を参照してください. これらのモデルを記述するためには若干のテクニックが必要です.

研究者向けの統計解析ソフトウェア **Stata®** **評価版のお申込み**

機能制限なし！30日間無料で使える評価版でまずはお試しください！

https://www.lightstone.co.jp/stata/evaluate.html

※学生の方への評価版の提供はございません。大学の講義で使用するなど単位取得に関わる形でのご利用となる場合は学生版のご購入をご検討ください。

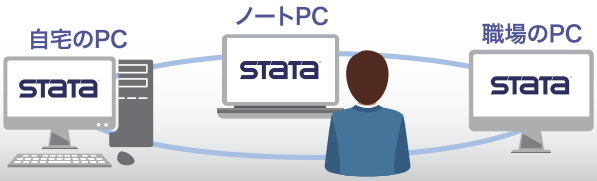
ライセンスはサブスクリプションがおすすめ！

サブスクリプションライセンスを選ぶメリット

- 常にStataの最新バージョンが利用できます
- 技術サポートの対象となります
- 毎年の経費としてStataを導入できます
- 利用者の増減に柔軟に対応可能
- 初期導入費を抑えたい場合にもおすすめ

シングルライセンス（サブスクリプション）

職場のPC・ノートPC・自宅のPC等、個人所有のPCにインストール可能です。
同時に使用できるのは1台のみとなります。



マルチユーザライセンス（サブスクリプション）　2ユーザ以上

ボリュームライセンス

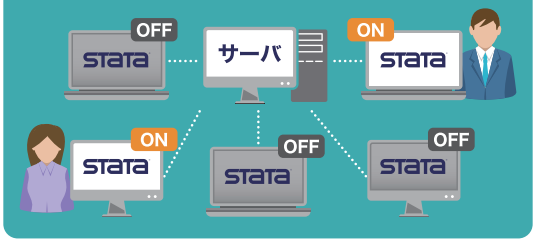
シングルライセンスを複数人でまとめてご購入いただく際のボリュームディスカウント商品です。

同時起動ライセンス

ご利用環境により下記3通りの運用方法からご選択いただけます。

PC毎にスタンドアロンインストールをして運用
例:2ライセンスの場合

クライアント&サーバ環境によるネットワーク運用
例:2ライセンスの場合



1台の仮想マシンにユーザがアクセスする運用
例:2ライセンスの場合

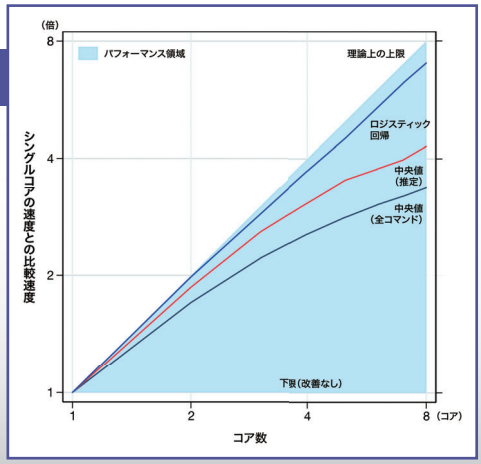
＼データ処理時間の削減ならMP／

Stata／MPの演算能力

Stata MPはPCの持つマルチコアの特性を活かして、処理を分散・並列する機能を備えます。およそ85%以上のコマンドで処理速度が向上し、コア数に応じた計算時間の短縮が期待できます。コマンドごとの処理速度の向上の度合いについては以下のページや資料をご覧ください。

コア数	全てのコマンド	推定コマンド	ロジスティック回帰
2	1.7倍	1.8倍	1.9倍
4	2.6倍	3.1倍	3.8倍
8	3.3倍	4.2倍	6.8倍

弊社Webページ内の『Stata／MP』
https://www.lightstone.co.jp/stata/statamp.html



株式会社 ライトストーン
〒101−0031
東京都千代田区東神田 2−5−12　龍角散ビル 7F
TEL：03−3864−5211　e-mail：sales@lightstone.co.jp
FAX：03−3865−0050　WEB：https://www.lightstone.co.jp/