

## FAQ(3) Procedure 関連

### 【質問 1】

プログラム中に自分で作成した procedure を置いているのですが、それを呼び出そうとすると、Wrong number of returns というエラーが出てしまいます。

### 【解答】

procedure を作成したなら、retpl( )の丸括弧の中味の変数について着目してください。通常のリターンが1個（それが行列であろうとベクトルであろうと関係ありません。変数の数が問題です）の場合、冒頭は proc 関数名(引数);となります。しかしながら、リターンの個数が2つ以上の複数の場合、この方法ではいけません。なお、インプットの引数の個数は無関係です。

### 【対処方法】

リターンの数です。もし、2つ以上の変数がカンマで区切られてリターンとして retpl( )の中に書かれているのなら、その個数だけ proc(2)=関数名(引数);としないといけません。retpl( )の中のリターンの個数と冒頭の proc(数字)= のところの数が常に一致している必要があります。1つのリターンのときにのみ、

```
proc 関数名(引数);
```

という書き方が許されています。その他の個数のリターンの場合、特に0個のリターンの場合や retpl ; となっていたりリターン文自体がない場合も、

```
proc(0)=関数名(引数);
```

```
proc(2)=関数名(引数);
```

などというふうに、しなくてははいけません。なお、1個の場合も上の書き方ができます。このことは、リターンのとしてその個数が外部に返されるのだということをその procedure のブロックの最初で宣言していることに相当します。ただし、リターンの変数を水平方向か垂直方向にマージして1つの変数にしてあるケースは1変数と勘定します。

### 【質問 2】

procedure の中に使った変数が Undefined symbol というエラーを出してしまいます。あるいは、local 文がないことはわかるのですが、Undefined symbol というエラーを出さない変数もあります。これはなぜですか？

### 【解答】

procedure 内で、インプット引数の丸括弧の中に書かれた変数以外の変数を「変数 = 」の形で使うのには、その変数は原則 local 文でローカル宣言する必要があります。ただし、これとは別に、procedure の外部にも共通して使いたい変数を procedure 内外を問わずに同等に使いたいのであれば、その変数は procedure 外部で何か数値や行列を入れて定義して、それをローカル変数の宣言なしに、procedure 内部で使うことも許されています。

### 【対処方法】

例えば、次のような procedure f を考える場合、

```
new; cls;  
a=5;  
print f(2);  
  
proc f(x);  
    local y;  
    y=a*x^2;  
    retp(y);  
endp;
```

f(x)のローカル変数は、インプットのx以外の、yとaのように思えますが、a=5;と外部で定義していますので、これをプログラムを通して全体で使える変数と考えて、ローカル宣言なしで使えます。一方、yは外部には定義されていないので、必ずローカル宣言が必要になります。なお、aは外部でそう決まっているのですが、procedure 内部だけで違う意味を持たせるにはローカル宣言すればよいだけです。したがって、procedure の外部にも内部にも別々にbやxなどが存在するのに、混乱したくなければそれぞれをローカル宣言して使えば、procedure の内部は、外部の変数をまったく気にせず、インプットだけを用いてプログラムできます。このような仕組みがあるからこそ、他人が作った procedure をそのままインプットを与えるだけで利用できるわけです。なお、ローカル変数が多くて1行に書けない場合には、複数の local 文を書いてもかまいません。

【質問3】

retp( )というリターン文がprocedure内に複数存在するプログラムを見たことがあるのですが、それは誤りではないでしょうか？

【解答】

何個リターン文があっても、条件分岐がしっかりできていれば、それは誤りではありません。ただし、それぞれのリターン文のリターンの変数の個数が1個以外の場合、そのそれぞれの個数と冒頭のproc(数字)=の数字とは一致している必要があります。合計ではありませんし、リターン文の個数自体でもありません。それぞれのリターン文の変数の個数です。

【対処の仕方】

リターン文がprocedure 内部に複数ある場合には、そのそれぞれの変数の個数と冒頭の数字が一致していることを目視で必ず確認してください。間違った個数や、個数がリターンごとに違っていても、間違いは間違いなのですが、一方のリターンの方しか通常使わなくて、もう一方のリターンの方が間違っているにもかかわらずプログラムの流れが行かなくて、エラーを見つけられない場合もあります。インプットにもう一方のリターンに向わせる意地悪テストをして確認するの必要でしょう。

**【質問 4】**

私は MATLAB をよく扱うのですが、GAUSS をやっていると PROC without ENDP というエラーによく遭遇します。

**【解答】**

Matlab などでは関数の締めに対応する部分が存在しなかったり省略されることがありますが、GAUSS では proc としてプログラムする場合、endp; は必須です。

**【対処方法】**

proc と endp; が対になっているかを目視で必ず確認してください。ただし、1 行定義の fn の方法では、Matlab と同様に締めに相当する部分は存在しません。あくまで、1 行定義で最後はセミコロンで終わりです。

【質問 5】

最適化問題をオプションのライブラリや GPE2 でやりたりのですが、その部分でうまくいきません。定義した目的関数または log-Likelihood 自体に問題があるようですが、プログラム自体は完璧であるか他からコピーしたもので問題はないはずですが？

【解答】

おそらく、それは定義した目的関数のインプットの数に問題があるものと思われます。通常、自作の最適化（最大化あるいは最小化）アルゴリズムでは、1 値のインプット引数の関数の最適化を行ないます。オプションの最適化のライブラリは、maxlik と cml はパラメータのほかにデータ行列もついている 2 値のインプット引数の目的関数を最適化します。一方、同じオプションのライブラリでも optmum と co はパラメータだけの 1 値の目的関数を最適化します。GPE2 は上の両方ができますが、2 値の場合にはその変数の順序が逆転します。

【対処方法】

パラメータのほかにデータ行列もついている 2 値のインプット引数の目的関数の最適化を行なうには、データの方の変数を消去して、その変数の計算の部分を procedure 外部から全体の変数として串刺しにして「ローカル宣言せずに」内部で強引に計算させて動かすことも可能です。その反対に、1 値の目的関数を 2 値のインプット引数を必要とする maxlik や cml で最適化しようと思えば、データに相当する変数をダミーで作って、外部からダミーの変数を 2 値の一方に受け渡すようにしたものを最適化すれば、データを必要とする BHHH のアルゴリズムオプション関連以外は動くはずです。

【質問 6】

外部の procedure をインプットとして呼び出して処理するような、例えば最適化をするプログラムを procedure で作成したのですが動きません。

【解答】

それはローカル宣言が足りません。ローカル宣言の例外として、procedure や fnなどをインプットとして用いて別の procedure で処理する場合、その別の procedure のローカル宣言で:proc の文字をつけて明示することでその名前も宣言する必要があります。

【対処方法】

例えば、関数 f という procedure を最適化問題の procedure で「インプットとして」呼び出す場合には、local f:proc;と他のローカル変数とは別に宣言する必要があります。なお、この宣言は、通常のローカル変数の宣言と混ざって書いてもかまいません。また、fn で作成したものは通常は:fn としますが、:proc としてもかまいません。これは procedure の特殊なものが fn だからです。インプットとして procedure を呼び出す時のみ、この宣言は必要であって、インプットしてではない場合にはこの方法は使いません。

**【質問 7】**

procedure を入れ子にしたいのですがどうすればできますか？

**【解答】**

入れ子にできないことはありませんが、複数の procedure をプログラムに併記して、それを GAUSS の標準の組み込み関数と同じように使うことをイメージしてください。

**【対処方法】**

入れ子のことは忘れて、GAUSS 特有のプログラムの記述法である procedure をたくさん作って、それをプログラム上において、それを呼び出してください。プログラム上にある限り、procedure は標準組み込み関数と全く同等に呼び出して使うことができます。どれが上でどれが下であることは考えなくてもかまいません。

#### 【質問 8】

procedure を自分でも書いたのですが、Run しても何も動作しません。あるいは、ARMA や MA の procedure を手に他から入れたのですが、これを Run しても何も動作しません。

#### 【解答】

procedure だけをプログラム上において Run しても、何も動作しないばかりか、エラーがある procedure であるかもわかりません。そのインプットに何かの数値を入れて、外部で呼び出す必要があります。

#### 【対処方法】

呼び出し方には、インプット変数は外部で与えられているものとして、次のようにします。

( 1 ) `a=f(x);`

( 2 ) `{a,b,c}=f(x);`

( 3 ) `print f(x);`

リターンが 1 つだけの場合には、( 1 ) または ( 3 ) の方法ができます。リターンが複数の場合には { } 括弧の中に、任意の変数をそのリターンの個数だけカンマで区切って列挙して受け渡したものをその後で操作したり画面表示します。リターンが複数の場合には、`変数 =` の形には絶対できませんので注意してください。また、インプットを与えて呼び出したとしても、それは log-Likelihood 関数であったり、単なる残差（あるいは残差平方和）関数であったりするだけで、パラメータを計算するものではないことがあります。その場合には、最適化のアルゴリズムを自分でこしらえて、それを利用してその関数を最適化するパラメータを求めるか、オプションのライブラリを使って自動的に最適化パラメータを自動的に計算する方法とがあります。procedure 1 つだけでは何も動作しない、または意味あることをしないことがそのほとんどであることを理解してください。