

# 資料篇：1

## 第1章（植村）

## 資料篇 1-1. 入力関数の符号条件判定プログラム (EViews)

本来 EViews のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。

```
10  workfile 20120925_chn a 1970 2009
20  smpl 1970 2009
30  '
40  !count=43
```

<<<一時変数格納のためのベクトル及び行列の宣言>>>

```
50  svector(!count) _tmdl
60  matrix(!count,1) _check
70  '
```

<<<入力関数「候補」を羅列 (Excel より貼り付け) >>>

```
80  _tmdl(1) = "equation eqJPN1N0002A00.ls m1jpn c gdp (下に続く)
      pm1lc d(jpnpx1/pc1jpn)"
      (略)
500  _tmdl(43) = "equation eqUSA3N1100R00.ls log(m3usa) c (下に続く)
      log(gdp) pm3lc/pgdp usapx3/pc3usa"
510  '
520  for !i = 1 to !count
530  '
540  %z = _tmdl(!i)
550  '
560      {%z}
```

<<<入力関数文字列より、必要な部分 (関数の名前部分) のみ切り出し>>>

```
570      !txb = @instr(%z, " ")
580      !txe = @instr(%z, ".")
```

判定する「候補」をすべて一旦ワークスペース内に保存しておく。

```
590         %tmpfilnam = @mid(%z, !txb+1, !txe-!txb-1)
600     ’
```

推定された係数の符号判定

```
610         !income = @coefs(2)
620         !price1 = @coefs(3)
630         !price2 = @coefs(4)
640     ’
```

符号条件を満たさない場合、先ほど一時保存した関数名をワークスペースから削除する。従って、条件を満たす候補のみがワークスペース内に残り、以下の作業の対象となる。

```
650         if (!income<=0) or (!price1>=0) or (!price2>=0) then
660             delete {%tmpfilnam}
```

条件を満たした場合、判定用のフラグ格納ベクトルにフラグ 1 を立てる。

```
670         else
680             _check(!i, 1) = 1
690         endif
700     ’
710 next ’ !i
720 ’
```

判定用フラグベクトルをテキストファイルに書き出す（確認用）。

```
730 _check.write(t=txt)    (下に続く)
    "R:¥Personal¥File_EViews¥ModelFiles¥20121225chn.txt"
(以上)
```

## 資料篇 1-2. (1) 組み合わせ自動生成のためのプログラム (Excel VBA)

本来 Excel VBA のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。ここに掲載したものは中国モデル作成支援のためのプログラムであるが、他の国についてもほぼ同様である（相手国が変わることは当然であるが、90~160行で指定する「候補」の数が多少変化する程度である）。

```

10    Sub SourceFiles()
20    '
30    Dim TempStr, SourceFileName As String
40    Const WorkDirectory = "C:\EViewsTemp\"
50
60    ' 輸入関数番号の組み合わせ自体をファイル名とする
70    Sheets("候補順").Select

```

セル AJ10 に移動し、以下に作成される「モデル名」を順番に書き出していく。モデル総数は 256 個（中国の場合）で、AJ10~AJ266 まで、1 行に 1 つのモデル名が書き出される。これを EViews のスクリプトにそのまま貼り付けられて用いる。

```

80    Range("aj10").Select

```

以下、90 行~160 行で順次 1 と 2 を取ることになる変数 (a~h) は、それぞれ以下の輸入関数「候補」の上位 2 番目までのすべての組み合わせを作成するための変数である。

a : 日本からの財 1 (一次産品) 輸入関数  
b : 日本からの財 3 (製造業品) 輸入関数  
. . .  
g : 米国からの財 1 (一次産品) 輸入関数  
h : 米国からの財 3 (製造業品) 輸入関数

```

90    For a = 1 To 2
100        For b = 1 To 2
110            For c = 1 To 2
120                For d = 1 To 2
130                    For e = 1 To 2

```

```

140             For f = 1 To 2
150                 For g = 1 To 2
160                     For h = 1 To 2
170

```

セル AG1 には、モデル名に対応した入力関数を格納したファイルにつけるファイル名が書き出される。これを 200 行でファイル名とし、310 行で宣言する。

```

180         Range("ag1").Value = "s" & a & b & c & d & e & f & g & h
190         TempStr = Range("ag1").Value
200         SourceFileName = TempStr & ".txt"
210

```

セル L4 から S4 には、90 行～160 行で設定された各入力関数の番号が順に入れられる。

```

220         Range("l4").Value = a
230         Range("m4").Value = b
240         Range("n4").Value = c
250         Range("o4").Value = d
260         Range("p4").Value = e
270         Range("q4").Value = f
280         Range("r4").Value = g
290         Range("s4").Value = h
300

```

同時に、これらの値を参照してセル AG5 から AG12 に、入力関数の番号に対応した関数名 (EViews でそのまま関数のラベルとして使われる) が出力され、310 行で開いたファイルに 320 行～390 行で書き出される。

```

310         Open WorkDirectory & SourceFileName For Output As #2
320             Print #2, ":" & Range("ag5").Value
330             Print #2, ":" & Range("ag6").Value
340             Print #2, ":" & Range("ag7").Value
350             Print #2, ":" & Range("ag8").Value
360             Print #2, ":" & Range("ag9").Value

```

```

370          Print #2, ":" & Range("ag10").Value
380          Print #2, ":" & Range("ag11").Value
390          Print #2, ":" & Range("ag12").Value
400      Close #2
410

```

例えば、プログラムを走らせた直後に作成される (a=b=・・・=h=1 に対応する) モデル名 (80 行) は「m11111111」であり、それに対応する関数を格納するファイル名 (200 行) は「s11111111.txt」、そして、s11111111.txt の中身は以下の通りとなる。

```

<s11111111.txt>
eqJPN1N0012R00
eqJPN3N1123A00
eqKOR1N0012R00
eqKOR3N1122R11
eqTWN1L1122R00
eqTWN3N3330R00
eqUSA1N0010R01
eqUSA3N1100R00

420          Next 'h
430          Next 'g
440          Next 'f
450          Next 'e
460          Next 'd
470          Next 'c
480          Next 'b
490      Next 'a
500
510      End Sub
(以上)

```

資料篇 1 - 2. (2) 自動生成された輸入関数のセット例

(例 1) モデル m11111111 の輸入関数群

(ファイル s11111111.txt の内容)

:eqJPN1N0012R00

:eqJPN3N1123A00

:eqKOR1N0012R00

:eqKOR3N1122R11

:eqTWN1L1122R00

:eqTWN3N3330R00

:eqUSA1N0010R01

:eqUSA3N1100R00

(モデルに挿入される輸入関数群)

eqJPN1N0012R00 :

(定式化)  $m1jpn = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), d(\text{jpnpx1} / \text{pc1jpn}) ]$

(推定式)  $m1jpn = -295224.34 + 164.63831 * \text{gdp} - 2181505.5 * \log(\text{pm1lc} / \text{pgdp})$   
 $-1863140.7 * d(\text{jpnpx1} / \text{pc1jpn})$

eqJPN3N1123A00 :

(定式化)  $\log(m3jpn) = f[ \log(\text{gdp}), d(\text{pm3lc}), d\log(\text{jpnpx3} / \text{pc3jpn}) ]$

(推定式) (以下省略)

eqKOR1N0012R00 :

(定式化)  $m1kor = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), d(\text{korpx1} / \text{pc1kor}) ]$

eqKOR3N1122R11 :

(定式化)  $\log(m3kor) = f [ \log(\text{gdp}), d(\text{pm3lc}(-1) / \text{pgdp}(-1)),$   
 $d(\text{korpx3}(-1) / \text{pc3kor}(-1)) ]$

eqTWN1L1122R00 :

(定式化)  $\log(m1twn) = f[ \log(\text{gdp}), d(\text{pm1lc} / \text{pgdp}), d(\text{twnpx1} / \text{pc1twn}),$   
 $\log(m1twn(-1)) ]$

eqTWN3N3330R00 :

(定式化)  $d\log(m3twn) = f[ d\log(\text{gdp}), d\log(\text{pm3lc} / \text{pgdp}), \text{twnpx3} / \text{pc3twn} ]$

eqUSA1N0010R01 :

(定式化)  $m1usa = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{usapx1}(-1) / \text{pc1usa}(-1) ]$

eqUSA3N1100R00 :

(定式化)  $\log(m3usa) = f[ \log(\text{gdp}), \text{pm3lc} / \text{pgdp}, \text{usapx3} / \text{pc3usa} ]$

(例2) モデル m12121212 の輸入関数群

(ファイル s12121212.txt の内容)

:eqJPN1N0012R00

:eqJPN3L0003A00

:eqKOR1N0012R00

:eqKOR3N1123R11

:eqTWN1L1122R00

:eqTWN3L3320R00

:eqUSA1N0010R01

:eqUSA3L1100R00

(モデルに挿入される輸入関数群)

eqJPN1N0012R00

(定式化)  $m1jpn = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), d(\text{jnpx1} / \text{pc1jpn}) ]$

eqJPN3L0003A00

(定式化)  $m3jpn = f[ \text{gdp}, \text{pm3lc}, d\log(\text{jnpx3} / \text{pc3jpn}), m3jpn(-1) ]$

eqKOR1N0012R00

(定式化)  $m1kor = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), d(\text{korpx1} / \text{pc1kor}) ]$

eqKOR3N1123R11

(定式化)  $\log(m3kor) = f[ \log(\text{gdp}), d(\text{pm3lc}(-1) / \text{pgdp}(-1)),$   
 $d\log(\text{korpx3}(-1) / \text{pc3kor}(-1)) ]$

eqTWN1L1122R00

(定式化)  $\log(m1twn) = f[ \log(\text{gdp}), d(\text{pm1lc} / \text{pgdp}), d(\text{twnpx1} / \text{pc1twn}),$   
 $\log(m1twn(-1)) ]$

eqTWN3L3320R00

(定式化)  $d\log(m3twn) = f[ d\log(\text{gdp}), d(\text{pm3lc} / \text{pgdp}), \text{twnpx3} / \text{pc3twn},$   
 $d\log(m3twn(-1)) ]$

eqUSA1N0010R01

(定式化)  $m1usa = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{usapx1}(-1) / \text{pc1usa}(-1) ]$

eqUSA3L1100R00

(定式化)  $\log(m3usa) = f[ \log(\text{gdp}), \text{pm3lc} / \text{pgdp}, \text{usapx3} / \text{pc3usa},$   
 $\log(m3usa(-1)) ]$



(例3) モデル m21212121 の輸入関数群

(ファイル s21212121.txt の内容)

:eqJPN1N0013R00

:eqJPN3N1123A00

:eqKOR1N0013R00

:eqKOR3N1122R11

:eqTWN1L1130R00

:eqTWN3N3330R00

:eqUSA1N0011R01

:eqUSA3N1100R00

(モデルに挿入される輸入関数群)

eqJPN1N0013R00

(定式化)  $m1jpn = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{dlog}(\text{jpnpx1} / \text{pc1jpn}) ]$

eqJPN3N1123A00

(定式化)  $\log(m3jpn) = f[ \log(\text{gdp}), \text{d}(\text{pm3lc}), \text{dlog}(\text{jpnpx3} / \text{pc3jpn}) ]$

eqKOR1N0013R00

(定式化)  $m1kor = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{dlog}(\text{korpx1} / \text{pc1kor}) ]$

eqKOR3N1122R11

(定式化)  $\log(m3kor) = f[ \log(\text{gdp}), \text{d}(\text{pm3lc}(-1) / \text{pgdp}(-1)),$   
 $\text{d}(\text{korpx3}(-1) / \text{pc3kor}(-1)) ]$

eqTWN1L1130R00

(定式化)  $\log(m1twn) = f[ \log(\text{gdp}), \text{dlog}(\text{pm1lc} / \text{pgdp}), \text{twnpx1} / \text{pc1twn},$   
 $\log(m1twn(-1)) ]$

eqTWN3N3330R00

(定式化)  $\text{dlog}(m3twn) = f[ \text{dlog}(\text{gdp}), \text{dlog}(\text{pm3lc} / \text{pgdp}), \text{twnpx3} / \text{pc3twn} ]$

eqUSA1N0011R01

(定式化)  $m1usa = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \log(\text{usapx1}(-1) / \text{pc1usa}(-1)) ]$

eqUSA3N1100R00

(定式化)  $\log(m3usa) = f[ \log(\text{gdp}), \text{pm3lc} / \text{pgdp}, \text{usapx3} / \text{pc3usa} ]$

(例4) モデル m22222222 の輸入関数群

(ファイル s22222222.txt の内容)

:eqJPN1N0013R00

:eqJPN3L0003A00

:eqKOR1N0013R00

:eqKOR3N1123R11

:eqTWN1L1130R00

:eqTWN3L3320R00

:eqUSA1N0011R01

:eqUSA3L1100R00

(モデルに挿入される輸入関数群)

eqJPN1N0013R00

(定式化)  $m1jpn = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{dlog}(\text{jpnpx1} / \text{pc1jpn}) ]$

eqJPN3L0003A00

(定式化)  $m3jpn = f[ \text{gdp}, \text{pm3lc}, \text{dlog}(\text{jpnpx3} / \text{pc3jpn}), m3jpn(-1) ]$

eqKOR1N0013R00

(定式化)  $m1kor = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \text{dlog}(\text{korpx1} / \text{pc1kor}) ]$

eqKOR3N1123R11

(定式化)  $\log(m3kor) = f[ \log(\text{gdp}), \text{d}(\text{pm3lc}(-1) / \text{pgdp}(-1)),$   
 $\text{dlog}(\text{korpx3}(-1) / \text{pc3kor}(-1)) ]$

eqTWN1L1130R00

(定式化)  $\log(m1twn) = f[ \log(\text{gdp}), \text{dlog}(\text{pm1lc} / \text{pgdp}), \text{twnpx1} / \text{pc1twn},$   
 $\log(m1twn(-1)) ]$

eqTWN3L3320R00

(定式化)  $\text{dlog}(m3twn) = f[ \text{dlog}(\text{gdp}), \text{d}(\text{pm3lc} / \text{pgdp}), \text{twnpx3} / \text{pc3twn},$   
 $\text{dlog}(m3twn(-1)) ]$

eqUSA1N0011R01

(定式化)  $m1usa = f[ \text{gdp}, \log(\text{pm1lc} / \text{pgdp}), \log(\text{usapx1}(-1) / \text{pc1usa}(-1)) ]$

eqUSA3L1100R00

(定式化)  $\log(m3usa) = f[ \log(\text{gdp}), \text{pm3lc} / \text{pgdp}, \text{usapx3} / \text{pc3usa},$   
 $\log(m3usa(-1)) ]$

(例示以上)

### 資料篇 1 - 3. モデルの自動構築プログラム (EViews)

本来 EViews のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。なお、ここで取り上げている中国モデルでは、輸入関数モデル数（格納ファイルも）が 256 個あるため、以下の紹介では 90 行～2610 行、2670 行から 5190 行を省略している。

```
10      workfile 20120925_chn a 1970 2009
20
30      !count = 256
```

40 行、50 行で \_\_tmn、\_\_tsn という名前のベクトルを宣言し、それぞれモデル名（m11111111～m22222222）、輸入関数格納ファイル名（s11111111～s22222222）を格納する。

```
40      svector(!count) __tmn
50      svector(!count) __tsn
60
70      __tmn(1) = "m11111111"
80      __tmn(2) = "m11111112"
      (略)
2620   __tmn(256) = "m22222222"
2630
2640
2650   __tsn(1) = "s11111111"
2660   __tsn(2) = "s11111112"
      (略)
5200   __tsn(256) = "s22222222"
5210
5220
```

プログラム本体部分はこちらから。一時変数!i を 1～256 まで変化させながらモデル名、ファイル名を呼び出す。5260 行でモデルの宣言、そこに 5270 行でモデル名に対応する輸入関数群をファイル (sxxxxxxx.txt) から呼び出してマージする。続く 5280

行でモデルの本体部分 (GDP 定義式や輸出関数等、すべてのモデルに共通している部分) がマージされ、完結したマクロモデルとなる。これらモデルはワークスペース内に「mxxxxxxx」というモデル名で保存され、使用可能な状態となっている。

```
5230   for !i = 1 to !count
5240       %z1 = __tmn(!i)
5250       %z2 = __tsn(!i)
5260   model {%z1}
5270   {%z1}.merge(t)   (下に続く)
           R:¥Personal¥File_EVIEWS¥ModelFiles¥M-Files¥{%z2}.txt
5280   {%z1}.merge(t)   (下に続く)
           R:¥Personal¥File_EVIEWS¥ModelFiles¥_modelbodytwn.txt
5290   next ' !i
```

(以上)

#### 資料篇 1 - 4. モデルパフォーマンス確認作業の自動化プログラム (EViews)

本来 EViews のプログラムは行番号に依存しないが、ここでは解説のために行番号をつけてある。これらの「コメント」を取り外せば（行番号も不要）、そのまま作動するプログラムとなっている。なお、ここで取り上げている中国モデルでは、輸入関数モデルが 256 個あるため、以下の紹介では 150 行～2670 行を省略している。

```
10      workfile 20120925_chn a 1970 2009
20      '-----
30      '--- sample period ---
40      !begyear = 2001
50      !endyear = 2009
60      !sampleperiod = !endyear - !begyear + 1
70      '-----
```

サンプル期間の設定を行っている。有効データ期間が変更になったり、シミュレーション期間を変えて実験したりする場合、プログラムのあちこちに開始年と終了年（の具体的な数字）を散りばめるのは、プログラムの管理上も問題があるし、そもそもケアレス・ミステイクの温床となる。従ってここでは、この 1 箇所だけを変更すればプログラム全体にわたってサンプル期間の変更が一斉に完了するようにした。

```
80      smpl !begyear !endyear
90      '
```

一時変数の設定。

```
100     !count = 256
110     svector(!count) __tmn
120     '
130     __tmn(1) = "m11111111"
140     __tmn(2) = "m11111112"
        (略)
2680    __tmn(256) = "m22222222"
2690    '
```

<<<計算した RMSPE を格納するベクトルと、ファイル名 (文字列) を格納するベクトルの宣言>>>

```
2700  vector(!count,1) _rmspes
2710  svector(!count) _rmspem
2720  _rmspes = 0
2730  '
2740  for !i = 1 to !count
2750  %z = __tmn(!i)
2760  '

```

ターゲット変数 (RMPSE を計算する対象となる) の設定。ここでは対世界財総輸入を取り上げている。この行を変更することにより、他の変数を対象とすることもできるし、行を増やして複数のターゲットを設定することも可能である。

```
2770  target ="m0wld"
2780  '

```

<<<モデルを解く>>>

```
2790  solve {%z}

```

<<<RMPSE の計算とベクトルへの格納>>>

```
2800      series _dtp = (({%target} - {%target}_0 )/{%target})
2810      !_dsq = @inner(_dtp)
2820      scalar _rmspe = (!_dsq/!sampleperiod)^(1/2)
2830      _rmspes(!i,1) = _rmspe
2840      _rmspem(!i) = %z
2850  next ' !i

```

(以上)