R による時系列分析の方法 2[†]

以下の内容について説明する

1. **VAR モデル推定する**
2. **VAR モデルを用いて予測する**
3. **グレンジャーの因果性を検定する。**
4. **インパルス応答関数を描く**

1. VAR モデルを推定する。

ここでは VAR(p)モデル：

$$\mathbf{y}_t = \mathbf{c} + \mathbf{\Phi}_1 \mathbf{y}_{t-1} + \ldots + \mathbf{\Phi}_p \mathbf{y}_{t-p} + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathrm{W.N}(\mathbf{\Omega}),$$

を推定することを考える(ただし誤差項は場合によっては i.i.d を仮定)。

1.1 パッケージ “vars” をインスツールする。

VAR モデルを推定するために R のパッケージ vars をインスツールする。パッケージとは通常の R には含まれていない、追加的な R のコマンドの集まりのようなものである。R には追加的に 600 以上のパッケージが用意されており、それぞれ分析の目的に応じて標準の R にパッケージを追加していくことになる。

インターネットに接続してあるパソコンで R を起動させ、「パッケージ」→「パッケージのインストール...」→「Japan(Tokyo)」→「vars」→「OK」とクリックする。すると(いろいろとインストールの途中経過が表示されて)パッケージのインストールが自動的に終わる。(上記の作業は次回以降はやる必要はないが、以下の作業は R を起動するたびに毎回やる必要がある)。次にインストールしたパッケージを使うためにコマンドウィンドウ (R Consolde) に

```
> library(vars)
```

と入力すると(再びコマンドウィンドウ上にいろいろと表示され)パッケージ vars を使用できる様になる。

次に使用するデータを読み込む。教科書の著者のホームページからダウンロードした G7 国の日次の MSCI(Morgan Stanley Capital International)データ (これは MSCI が作成している株式指数) の(対数階差×100 によって計算した)収益率を用いる。MSCI データについて詳しくは教科書を参照。データはテキストファイルの形式に直したものを講義のホームページにアップしてある (msci_ret.txt ファイル)。「ファイル」→「ディレクトリの変更」によってデータ msci_ret.txt のあるディレクトリに移動。データを読み込み最初の 5 行だけ表示してみる。

```
> msci=read.table("msci_ret.txt",header=T)
> head(msci,5)
Date        ca        fr        ge        it        jp        uk        us
1 2003/1/2 2.573635  2.747145 5.790717  2.308201 -0.943505 0.848019  3.296573
2 2003/1/3 0.954365  0.225262 0.033972  0.602298  0.012482  0.295622 -0.033330
```

[†]この資料は私のゼミおよび講義で R の使用法を説明するために作成した資料です。ホームページ上で公開しており、自由に参照して頂いて構いません。ただし、内容について、一応検証してありますが、間違いがあるかもしれません。間違いがあった場合でもそれによって生じるいかなる損害、不利益について責任は負いかねますますのでご了承下さい。

```
3 2003/1/6 1.608301  1.464002 2.538144  1.500286  2.582230  0.472653  2.272284
4 2003/1/7 -0.646058 -2.058904 -1.806455 -1.790239 -1.790900 -1.294592 -0.635741
5 2003/1/8 -1.600249 -2.560138 -4.173460 -1.386237 -1.672348 -1.275576 -1.464613
```

ca はカナダ、fr はフランス、ge はドイツ、it はイタリア、jp は日本、uk はイギリス、us はアメリカを指している。

1.2. VAR(p) モデルを推定する

それではまずカナダと日本についての 2 変量 VAR(2)モデルを条件付き最尤法によって推定してみよう。このためにまず ca と jp のデータだけを含んだ新しいデータを作成する。例えばこのデータに cajp という名前を付けるとすると

```
> cajp=data.frame(msci$ca,msci$jp)
```

とする(msci$ca によって msci の ca を取り出している。msci$jp も同様)。最初の 5 行を見てみると

```
> head(cajp,5)
    msci.ca   msci.jp
1  2.573635 -0.943505
2  0.954365  0.012482
3  1.608301  2.582230
4 -0.646058 -1.790900
5 -1.600249 -1.672348
```

となっている。ca と jp だけから構成されていることがわかる(一番左の番号は R が自動的につけた番号)。これらのデータの 2 変量 VAR(2)モデルを推定するには VAR() 関数を用いる。推定結果に var2cajp という名前を付けるとすると、

```
> var2cajp=VAR(cajp,p=2,type="const")
```

とすればよい。ここで p は推定する VAR の次数、type="const" は定数項を含める時このようにする(含めないときは type = "none", トレンド項を含めるときは type = "trend", トレンド項と定数項を両方含めるときには type = "both" とする)。

推定結果は summary( ) 関数を使って見る事ができる(太字青字の部分が推定結果です。これは私が見やすいようにこうしたので、実際の出力ではこの部分は太字青字にはなっていません)。

```
> summary(var2cajp)

VAR Estimation Results:
=========================
Endogenous variables: msci.ca, msci.jp
Deterministic variables: const
Sample size: 1388
Log Likelihood: -4229.794
Roots of the characteristic polynomial:
0.2471 0.2471 0.1877 0.1877
Call:
VAR(y = cajp, p = 2, type = "const")
```

```
Estimation results for equation msci.ca:
========================================
msci.ca = msci.ca.l1 + msci.jp.l1 + msci.ca.l2 + msci.jp.l2 + const

            Estimate Std. Error t value Pr(>|t|)
msci.ca.l1  0.044230   0.027250   1.623  0.10479
msci.jp.l1  0.025374   0.023245   1.092  0.27520
msci.ca.l2  0.002142   0.028842   0.074  0.94082
msci.jp.l2 -0.032767   0.022041  -1.487  0.13734
const       0.081668   0.028009   2.916  0.00361 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.036 on 1383 degrees of freedom
Multiple R-Squared: 0.005056,   Adjusted R-squared: 0.002179
F-statistic: 1.757 on 4 and 1383 DF,  p-value: 0.1351

Estimation results for equation msci.jp:
========================================
msci.jp = msci.ca.l1 + msci.jp.l1 + msci.ca.l2 + msci.jp.l2 + const

           Estimate Std. Error t value Pr(>|t|)
msci.ca.l1  0.42559    0.03185  13.364  < 2e-16 ***
msci.jp.l1 -0.11482    0.02717  -4.227 2.53e-05 ***
msci.ca.l2  0.07060    0.03371   2.094  0.03640 *
msci.jp.l2 -0.07543    0.02576  -2.928  0.00346 **
const       0.01250    0.03273   0.382  0.70257
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 1.211 on 1383 degrees of freedom
Multiple R-Squared: 0.1196,     Adjusted R-squared: 0.117
F-statistic: 46.97 on 4 and 1383 DF,  p-value: < 2.2e-16

Covariance matrix of residuals:
        msci.ca msci.jp
msci.ca  1.0740  0.2091
msci.jp  0.2091  1.4668

Correlation matrix of residuals:
        msci.ca msci.jp
msci.ca  1.0000  0.1666
msci.jp  0.1666  1.0000
```

上記の推定結果は

$$ca_t = 0.081668 + 0.044230\, ca_{t-1} + 0.025374\, jp_{t-1} + 0.002142\, ca_{t-2} - 0.032767\, jp_{t-2} + \varepsilon_{1t}$$

$$jp_t = 0.01250 + 0.42559\, ca_{t-1} - 0.11482\, jp_{t-1} + 0.07060\, ca_{t-2} - 0.07543\, jp_{t-2} + \varepsilon_{2t}$$

$$\mathrm{var}\begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{bmatrix} = \begin{bmatrix} 1.0740 & 0.2091 \\ 0.2091 & 1.4668 \end{bmatrix}$$

という事を示している。

（補足： 誤差項の分散共分散行列の推定値について）
推定値より計算した残差（これはそれぞれの回帰式の OLS 残差と等しい）を

$$\hat{\boldsymbol{\varepsilon}}_t = \mathbf{y}_t - \hat{\mathbf{c}} - \hat{\boldsymbol{\Phi}}_1 \mathbf{y}_{t-1} - \ldots - \hat{\boldsymbol{\Phi}}_p \mathbf{y}_{t-p}\ ,\ t = p+1, \ldots, T$$

とすると、(誤差項に正規分布を仮定した)条件付き最尤法では $\boldsymbol{\Omega}$ の推定値は

$$\hat{\boldsymbol{\Omega}} = \frac{1}{T-p} \sum_{t=p+1}^{T} \boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t'$$

であるが、vars の VAR コマンドを用いた推定では、

$$\hat{\boldsymbol{\Omega}} = \frac{1}{T-p-1-np} \sum_{t=p+1}^{T} \boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t'\ ,$$

すなわち OLS 推定のように標本数から回帰式の説明変数（と定数項）の数を引いたもので割った推定が出力されていることに注意。

summary() 関数で結果を見るといろんなものがいっぺんに出力されてしまうので、係数の推定値だけを見たい場合は coef() 関数を使う。

```
> coef(var2cajp)
$msci.ca
              Estimate Std. Error   t value    Pr(>|t|)
msci.ca.l1  0.044230484 0.02725028  1.623120 0.104791573
msci.jp.l1  0.025374115 0.02324488  1.091600 0.275198973
msci.ca.l2  0.002141753 0.02884166  0.074259 0.940815042
msci.jp.l2 -0.032766827 0.02204106 -1.486626 0.137341454
const       0.081667982 0.02800909  2.915767 0.003605339

$msci.jp
              Estimate Std. Error    t value     Pr(>|t|)
msci.ca.l1  0.42559163 0.03184659 13.3638055 2.121305e-38
msci.jp.l1 -0.11482167 0.02716560 -4.2267310 2.526548e-05
msci.ca.l2  0.07059701 0.03370639  2.0944697 3.639975e-02
msci.jp.l2 -0.07543156 0.02575873 -2.9283876 3.463036e-03
const       0.01250178 0.03273339  0.3819275 7.025738e-01
```

また係数を行列表示で見たい場合は Acoef() 関数や Bcoef() 関数を用いると

```
> Acoef(var2cajp)
[[1]]
        msci.ca.l1  msci.jp.l1
msci.ca 0.04423048  0.02537411
msci.jp 0.42559163 -0.11482167

[[2]]
         msci.ca.l2  msci.jp.l2
msci.ca 0.002141753 -0.03276683
msci.jp 0.070597009 -0.07543156

> Bcoef(var2cajp)
        msci.ca.l1  msci.jp.l1  msci.ca.l2  msci.jp.l2      const
msci.ca 0.04423048  0.02537411 0.002141753 -0.03276683 0.08166798
```

```
msci.jp 0.42559163 -0.11482167 0.070597009 -0.07543156 0.01250178
```

と表示する事もできる(しかし t value や標準誤差は表示されない)。

次にアメリカ、イギリス、日本のデータについて 3 変量 VAR(2)モデルを推定する。

まず 3 変量のデータを作る。

```
> usukjp=data.frame(msci$us,msci$uk,msci$jp)
> head(usukjp,5)
    msci.us   msci.uk   msci.jp
1  3.296573  0.848019 -0.943505
2 -0.033330  0.295622  0.012482
3  2.272284  0.472653  2.582230
4 -0.635741 -1.294592 -1.790900
5 -1.464613 -1.275576 -1.672348
```

次に先ほどと同様に推定する

```
> var2usukjp=VAR(usukjp,p=2,type="const")
> coef(var2usukjp)
$msci.us
               Estimate Std. Error    t value     Pr(>|t|)
msci.us.l1 -0.108754632 0.03048408 -3.5675883 0.0003726023
msci.uk.l1 -0.003954436 0.02727541 -0.1449817 0.8847464770
msci.jp.l1  0.010163820 0.02029120  0.5008980 0.6165227567
msci.us.l2 -0.026411630 0.03321418 -0.7951913 0.4266388867
msci.uk.l2  0.009666319 0.02671985  0.3617655 0.7175825951
msci.jp.l2 -0.002539888 0.01850181 -0.1372778 0.8908312514
const       0.035254426 0.02316750  1.5217191 0.1283084162

$msci.uk
              Estimate Std. Error     t value     Pr(>|t|)
msci.us.l1  0.49423261 0.03417176  14.4631875 2.982181e-44
msci.uk.l1 -0.31404450 0.03057495 -10.2713019 6.700977e-24
msci.jp.l1 -0.02258831 0.02274584  -0.9930744 3.208477e-01
msci.us.l2  0.17167376 0.03723213   4.6109030 4.380243e-06
msci.uk.l2  0.01073443 0.02995217   0.3583859 7.201093e-01
msci.jp.l2 -0.02507635 0.02073999  -1.2090820 2.268383e-01
const       0.03894191 0.02597009   1.4994905 1.339750e-01

$msci.jp
              Estimate Std. Error    t value     Pr(>|t|)
msci.us.l1  0.51153875 0.04076436 12.5486759 2.772687e-34
msci.uk.l1  0.21918263 0.03647363  6.0093457 2.379395e-09
msci.jp.l1 -0.13672488 0.02713409 -5.0388595 5.302780e-07
msci.us.l2  0.15292309 0.04441516  3.4430384 5.924175e-04
msci.uk.l2  0.03291661 0.03573070  0.9212416 3.570852e-01
msci.jp.l2 -0.05940869 0.02474126 -2.4011988 1.647298e-02
const       0.02300597 0.03098038  0.7425981 4.578513e-01
```

のようになる。

1.3. 次数の選択

VAR()関数は最大次数を設定してやると、情報量基準によって自動的に最適な次数を選択し、推定してくれる。この場合最大次数 lag.max と情報量基準 ic を設定してやる必要がある。

lag.max は整数、情報量基準は AIC なら AIC, BIC なら SC とする(SC は BIC の別名のシュワルツ情報量基準 (SIC)を表している)。例えば、先ほどのカナダと日本のデータに対して、最大次数を 5, AIC によって次数を選択したとすると

```
> varcajp=VAR(cajp,type="const",lag.max=5,ic="AIC")
> summary(varcajp)

VAR Estimation Results:
=========================
Endogenous variables: msci.ca, msci.jp
Deterministic variables: const
Sample size: 1387
Log Likelihood: -4221.269
Roots of the characteristic polynomial:
0.4583 0.4354 0.4354 0.3065 0.2611 0.2611
Call:
VAR(y = cajp, type = "const", lag.max = 5, ic = "AIC")

Estimation results for equation msci.ca:
========================================
msci.ca = msci.ca.l1 + msci.jp.l1 + msci.ca.l2 + msci.jp.l2 + msci.ca.l3
+ msci.jp.l3 + const

            Estimate Std. Error t value Pr(>|t|)
msci.ca.l1  0.044660   0.027232   1.640  0.10124
msci.jp.l1  0.030625   0.023308   1.314  0.18907
msci.ca.l2 -0.005876   0.028967  -0.203  0.83927
msci.jp.l2 -0.021849   0.023411  -0.933  0.35083
msci.ca.l3 -0.028071   0.028865  -0.972  0.33098
msci.jp.l3  0.047828   0.022100   2.164  0.03062 *
const       0.080939   0.028069   2.884  0.00399 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.035 on 1380 degrees of freedom
Multiple R-Squared: 0.008511,   Adjusted R-squared: 0.0042
F-statistic: 1.974 on 6 and 1380 DF,  p-value: 0.06628

Estimation results for equation msci.jp:
========================================
msci.jp = msci.ca.l1 + msci.jp.l1 + msci.ca.l2 + msci.jp.l2 + msci.ca.l3
+ msci.jp.l3 + const

            Estimate Std. Error t value Pr(>|t|)
msci.ca.l1  0.424080   0.031844  13.318  < 2e-16 ***
msci.jp.l1 -0.117131   0.027254  -4.298 1.85e-05 ***
msci.ca.l2  0.070085   0.033872   2.069  0.03872 *
msci.jp.l2 -0.089128   0.027375  -3.256  0.00116 **
msci.ca.l3  0.052529   0.033753   1.556  0.11987
msci.jp.l3 -0.024841   0.025842  -0.961  0.33660
const       0.008423   0.032822   0.257  0.79750
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.21 on 1380 degrees of freedom
Multiple R-Squared: 0.1206,    Adjusted R-squared: 0.1167
```

```
F-statistic: 31.53 on 6 and 1380 DF,  p-value: < 2.2e-16

Covariance matrix of residuals:
        msci.ca msci.jp
msci.ca  1.0709  0.2101
msci.jp  0.2101  1.4643

Correlation matrix of residuals:
        msci.ca msci.jp
msci.ca  1.0000  0.1678
msci.jp  0.1678  1.0000
```

のようになる。次数 p=3 が自動的に選択されている事がわかる。同様に BIC を用いると次数 p=1 が選択される（試してみて下さい）。また実際に AIC や BIC の値を見たい場合は VARselect() 関数を使うと、それぞれの情報量基準でどのラグを選んだかとその値を表示してくれる。

```
> VARselect(cajp,lag.max=5,type="const")
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      1      1      3
$criteria
               1         2         3         4         5
AIC(n) 0.4323897 0.4299484 0.4288886 0.4330619 0.4350347
HQ(n)  0.4408696 0.4440815 0.4486749 0.4585015 0.4661275
SC(n)  0.4550617 0.4677351 0.4817899 0.5010779 0.5181654
FPE(n) 1.5409355 1.5371783 1.5355502 1.5419722 1.5450177
```

ここで HQ と FPE というのはそれぞれ Hannan-Quinn 情報量基準、forecast prediction error 基準と呼ばれる情報量基準である。

2. VAR モデルを用いて予測をする

cajp データにおいて ca.msci と jp.msci の n 期先の予測値、およびその区間予測を計算するには predict() 関数を使う。先ほど推定した cajp データの 2 変量 VAR(2) モデルの推定結果を用いて予測をする。推定結果は var2cajp にまとめられているのでこれに predict() 関数を適用する。95%区間予測を 8 期先まで（ n.ahead=8 ci=0.95 とする） するとすると、

```
> predict(var2cajp, n.ahead=8, ci=0.95)
$msci.ca
           fcst     lower    upper       CI
[1,] 0.04264256 -1.988499 2.073784 2.031141
[2,] 0.07870349 -1.955759 2.113166 2.034462
[3,] 0.09283888 -1.943298 2.128976 2.036137
[4,] 0.08567549 -1.950608 2.121959 2.036284
[5,] 0.08496877 -1.951324 2.121261 2.036292
[6,] 0.08523117 -1.951062 2.121524 2.036293
[7,] 0.08528420 -1.951009 2.121577 2.036293
[8,] 0.08528687 -1.951006 2.121580 2.036293

$msci.jp
            fcst     lower    upper       CI
[1,] -0.19116298 -2.564897 2.182571 2.373734
[2,]  0.05259975 -2.472800 2.577999 2.525399
[3,]  0.05738790 -2.471641 2.586417 2.529029
```

```
[4,]  0.04701241 -2.482613 2.576638 2.529626
[5,]  0.04579180 -2.483848 2.575432 2.529640
[6,]  0.04590810 -2.483733 2.575549 2.529641
[7,]  0.04604860 -2.483592 2.575689 2.529641
[8,]  0.04606479 -2.483576 2.575706 2.529641
```

と出力される。fcst が点予測の値であり lower と upper がそれぞれ区間予測の上限と下限の値である。CI は upper から fcst を引いた値(upper – fcst の値)が出力されている。

3. グレンジャーの因果性を検定する。

グレンジャーの因果性を検定する方法の 1 つとして causality() 関数を用いる方法がある(ただしこれは後述する理由により有用性が少し限定的である)。先ほどの cajp データについての 2 変量 VAR(2)モデルの推定結果 var2cajp を用いて jp から ca にグレンジャーの因果性があるかを検定するには(つまり ca の回帰式の中で jp についての係数が全て 0 という帰無仮説の検定)下のように入力する。

```
> causality(var2cajp,cause="msci.jp")
$Granger

        Granger causality H0: msci.jp do not Granger-cause msci.ca
data:  VAR object var2cajp
F-Test = 1.8891, df1 = 2, df2 = 2766, p-value = 0.1514

$Instant
        H0: No instantaneous causality between: msci.jp and msci.ca
data:  VAR object var2cajp
Chi-squared = 37.4895, df = 1, p-value = 9.19e-10
```

すると上記のように出力される。ここでグレンジャーの因果性の検定に使用するのは$Grander の部分の F-Test の値である($Instant の部分は無視してよい。これはグレンジャーの瞬時的因果性というものの検定でここでは関係ない)。教科書のやり方、つまり r (制約数)×F 値が帰無仮説の下で漸近的に$\chi^2(r)$に従うという方法でやるのであればこの F 値に制約数をかけたものを使わないといけない(上記の P 値は通常の F 検定のものだと思われるがこちらの方の p 値で判断しても漸近的には問題ない)5%有意水準で検定する場合は p-value が 0.05 より小さければ棄却となるので、上記の結果より jp から ca へグレンジャー因果性がないという仮説は棄却されない事がわかる(つまり因果性はないという事)。また同様に ca から jp へグレンジャー因果性があるかどうかを検定すると棄却される。(確かめて下さい)。

causality()関数によるグレンジャー因果性検定は cause で指定した変数が他の残りの変数に対してグレンジャー因果性を持っているかどうかを**同時に**検定するので、例えば 3 変数 VAR のような場合ある 1 つの変数が他のもう 1 つの変数のみへグレンジャー因果性がないという帰無仮説の検定には使えない(causality()関数は他の 2 つの変数へグレンジャー因果性がないという帰無仮説を同時に検定している)。

4. インパルス応答関数を描く

4.1 非直交化インパルス応答関数

先ほどの cajp データにおいて jp の ca の誤差項へのインパルス応答関数の値を求めてみよう。10 期先までのインパルス応答関数を求めるとすると

```
> irfjpca=irf(var2cajp,response="msci.jp",impulse="msci.ca",
+ n.ahead=10,orth=FALSE)
```

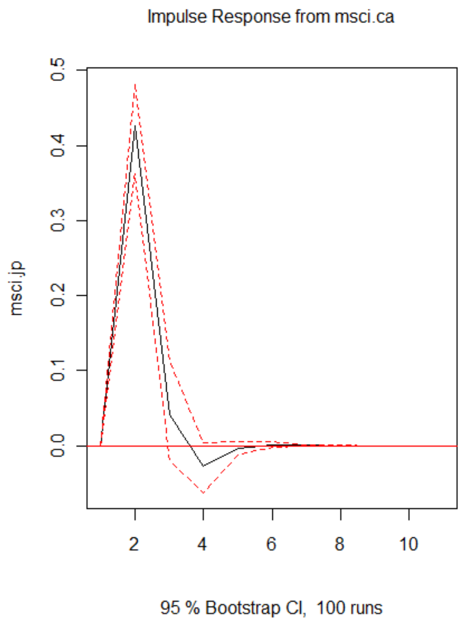
とする(結果に irfjpca という名前を付けた)。2 行目の最初に "+" がついているが、これは打ち込むコマンドが長くなる時に、エンターキーで行を変えてそのままコマンドの続きを打ち込む事ができるが、その時に "+" が表示される。カッコ内の 1 つ目には推定結果が保存してあるデータの名前を入力する(ここでは var2cajp)。ここで response には応答する方の変数名、impulse にはインパルスを与える誤差項を含む方の変数名を入力する。orth は直交化インパルス応答関数を求めるときには TRUE 、非直交化インパルス応答関数を求めるときには FALSE とする(上では非直交化インパルス応答関数を求めるため orth = FALSE としてある)。インパルス応答関数の値は

```
> irfjpca$irf
$msci.ca
           msci.jp
 [1,]  0.000000e+00
 [2,]  4.255916e-01
 [3,]  4.055399e-02
 [4,] -2.729690e-02
 [5,] -4.049398e-03
 [6,]  5.896724e-04
 [7,]  3.375793e-04
 [8,]  3.575985e-05
 [9,] -1.824694e-05
[10,] -4.927381e-06
[11,]  3.625408e-07
```

となっている。**一番目の値([1,]の横の値)は無視してよい**。2 番目の値からが求めたいインパルス応答関数の 1 期先の値である。またこれを図示したいときには

```
> plot(irfjpca)
```

とすればよい。下のような図が表示される。ここで赤線はインパルス応答関数(の推定値)の 95% 信頼区間を表している。(これは先ほどの"irfjpca=irf(…)" というコマンドのカッコの中に引数として ci = 0.99 などとすれば変更可能)



信頼区間を表示しないようにするには引数として boot=FALSE とすればよい。

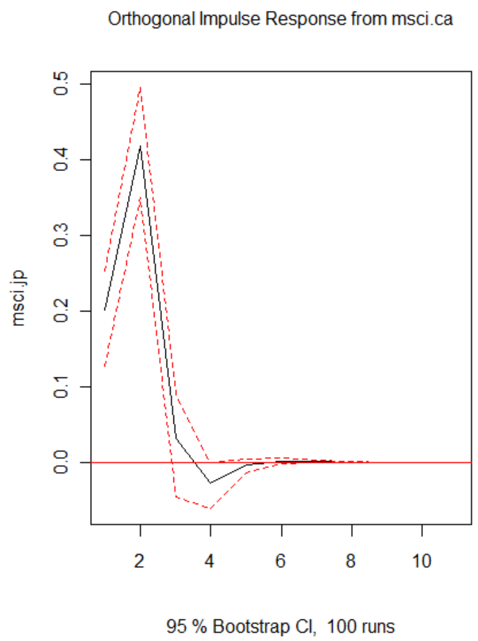
4.2 直交化インパルス応答関数

先ほどの関数の引数にて orth = TRUE とするだけである。結果は

```
> oirfjpca=irf(var2cajp,response="msci.jp",impulse="msci.ca",
+ n.ahead=10,orth=TRUE)
> oirfjpca$irf
$msci.ca
            msci.jp
 [1,]  2.017855e-01
 [2,]  4.178779e-01
 [3,]  3.164517e-02
 [4,] -2.795479e-02
 [5,] -3.859541e-03
 [6,]  7.070019e-04
 [7,]  3.432050e-04
 [8,]  2.841477e-05
 [9,] -1.946564e-05
[10,] -4.720331e-06
[11,]  4.687482e-07
```

(ここでも一番上の値は無視してよい。2 番目からが 1 期先のインパルス応答関数の推定値)

```
> plot(oirfjpca)
```


Orthogonal Impulse Response from msci.ca
msci.jp
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
10
95 % Bootstrap CI, 100 runs

となる。