

Numerical recipes for Econometrics
Part 2(J6 版)

Masato Shimura
JCD02773@nifty.ne.jp

2008 年 8 月 28 日

目次

第 II 部	第 2 部 同時方程式	137
第 6 章	景気動向指数の分析 3 態	139
6.1	主成分分析	139
6.2	景気動向指数のウェーブレット分解	146
6.3	景気動向指数とランダムウォーク	156
6.4	Reference	157
第 7 章	シンプレクス法	161
7.1	simplexnr	161
第 8 章	産業連関表	165
8.1	産業連関表のフォーマット	166
第 9 章	時系列解析 Time Series	173
9.1	自己相関	173
9.2	Autoregression 自己回帰	179
9.3	多変量時系列 VAR	190
9.4	インパルス応答	210
9.5	関数一覧	214
第 10 章	同時方程式の行列による解法	215
10.1	線形同時方程式	215
10.2	同時方程式の行列表現	217
10.3	誘導型と構造型	224
10.4	識別	227
10.5	ファイナルテスト	228
10.6	幾つかの線形同時方程式モデル	229
10.7	Klein Model	238
10.8	Reference	242

第 11 章	カルマンフィルタ	243
11.1	状態空間表現	243
11.2	Kalman Filter の構成	244
11.3	一期先予測	245
11.4	Working Example	246
11.5	固定区間平滑化	255
11.6	divbyfst	258
11.7	Script	258
付録 A	J のインストールとスタート	261
A.1	J の入手とインストール	261
A.2	J の起動	261
A.3	ディレクトリ/フォルダ	263
付録 B	J 言語超入門	265
B.1	アイバーソン言語の特色	265
B.2	数学用の動詞	269
B.3	小技	271
B.4	BOX	274
B.5	SCRIPT を書く	277
B.6	関数型定義	278
B.7	plot	281
付録 C	データの入出力	283
C.1	CSV ファイル	283
C.2	EXCEL などとの OLE	285
C.3	ADDONS の入手とインストール	286
付録 D	数学付録	287
D.1	行列の除算	287
D.2	逆行列	288
D.3	行列計算の手順	289
D.4	モデルの同時方程式を解く	290
索引		292

第 II 部

第 2 部 同時方程式

第6章

景気動向指数の分析 3 態

ESRI(社会経済研究所)の発表する30を越える月次または四半期の景気動向指数の長期系列は誰でもDLでき貴重である。2008年4月分からはCIを景気判断の軸に据え、さらに全てのデータの月次化が行われた。ESRIは景気のピーク・ボトム判定も行っているが、今がピークだとかボトムだとかは決して公表しない。

主成分分析 主成分分析を用いるとトレンド、サイクルなど景気の要素が分析できる。

ウェーブレット分解 景気動向指数のウェーブレット分解を試みる。

ランダムウォーク CIのランダムウォーク変換をベンチマークとして用いて景気動向の動勢を図示する

6.1 主成分分析

データの相関行列から固有ベクトルを求め、標準化されたデータに右からかけて内積を求めると、

$$X_0 T' = X^* \begin{bmatrix} x_{11}^* & x_{12}^* & \cdots & x_{1p}^* \\ x_{21}^* & x_{22}^* & \cdots & x_{2p}^* \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1}^* & x_{n2}^* & \cdots & x_{np}^* \end{bmatrix}$$

と変換される。

統計データから互いに無関係な因子を取り出し、観測値をこれらの因子の線形結合で説明することを主成分分析という。

6.1.1 景気動向指数の簡約

景気動向指数としてESRI(社会経済研究所)が公表しているのは、先行系列(L1-L12)、一致系列(C1-C11)、遅行系列(L1-L7)の35系列である。2008年4月から全て月次統計となった。

この先行系列、一致系列、遅行系列を主成分分析により、何本かの主成分系列に簡約化する。

一致系列では、生産関連の統計が多く含まれるが、USAでは、生産指数、個人所得、非農業雇用者数、商業販売額)の4系列のみである。このことは、固有値や累積寄与率に関

連してくる。

6.1.2 主成分分析 Principal Component Analsys

データの相関行列の固有ベクトルを求め、標準化したデータとの内積を求めると主成分が得られる。

数式で書くと複雑だが、プログラムや処理手順は簡潔である。

スクリプトは既出の関数と線形計算の LAPACK を用いると極めて簡潔である。

よく用いられる次の C_n の指標 (BI6) で試算してみよう。

- C1 鉱工業生産指数
- C5 製造業所定外労働時間指数
- C7 商業販売額指数 (小売業)
- C8 商業販売額指数 (卸売業)
- C10 中小企業売上高
- C11 有効求人倍率

グラフは標準化しているので、各指標の動向が一瞥できる。

6.1.3 経過と解説

Script BI6 の相関行列を作成しここから固有ベクトルを得る。固有ベクトルを標準化した BI6 にかける

```
syuseibun=: 3 : '(stand y.) +/ . * >{:dgeev_jlapack_ cortable y.'
```

データの読み込み 最初にデータを読み込む

```
read_data '' NB. _1 が出ればエラー
```

```
a=. pick_bi6 DAT_SNA
```

相関行列 相関行列を求める。 cortable a

```
a=. cortable pick_bi6 DAT_SNA
      1  0.468756  _0.632152  _0.385004  0.913095  0.475017
0.468756      1  _0.0742828  0.295779  0.525768  0.638448
_0.632152  _0.0742828      1  0.686128  _0.565724  0.148559
_0.385004  0.295779  0.686128      1  _0.320095  0.318727
0.913095  0.525768  _0.565724  _0.320095      1  0.537468
0.475017  0.638448  0.148559  0.318727  0.537468      1
```

固有値と固有ベクトル 固有値と固有ベクトルをを求める。固有値を主成分毎に (大きい順に) 累計すると累積寄与率が求められる。


```

1{ dgeev_jlapack_ a
+-----+
|2.03459 3.0672 0.081997 0.436681 0.15145 0.228086|
+-----+

```

固有ベクトル

```

> 2{ dgeev_jlapack_ a
_0.0442894 0.546101 0.723495 _0.143569 _0.28344 _0.274621
0.43893 0.342318 0.0449327 0.665072 _0.101441 0.485313
0.44832 _0.377384 0.0944257 _0.414525 _0.626296 0.28913
0.589283 _0.214286 _0.0135727 0.262405 0.03711 _0.7324
0.0197349 0.54613 _0.682019 _0.158203 _0.409414 _0.208695
0.506699 0.313044 _0.0168825 _0.520914 0.590024 0.159671

```

```
(stand a) +/ . * > {: dgeev_jlapack_ cortable a
```

主成分スコア 固有ベクトルを標準化した各指標にかけると主成分スコアが得られる。

主成分スコア (最後の 10 個) 左の列から第 1 第 2 主成分

```

7j3 ": 10{. calc_principal pick_bi6 DAT_SNA
_4.803 0.137 _1.942 0.755 0.405 0.523
_5.385 0.532 _2.332 0.645 _0.212 0.601
_5.980 1.285 _2.599 0.420 _0.993 0.592
_5.844 1.270 _2.609 0.461 _1.004 0.582
_5.518 0.459 _2.344 0.639 _0.638 0.551
_5.287 0.314 _1.850 0.336 _0.395 0.497
_5.057 0.154 _1.518 0.286 _0.279 0.439
_4.988 0.062 _1.366 0.365 _0.191 0.476
_4.375 _0.482 _0.644 _0.176 0.729 0.321
_4.769 0.180 _1.009 _0.025 0.008 0.370

```

累積寄与率 累積寄与率は、各固有値を大きい順に並べ、固有値の合計で割ればよい。第 2 主成分までで 85% を占める。

```

calc_cum pick_bi6 DAT_SNA
0 3.0672 0.5112
1 2.03459 0.850298
2 0.436681 0.923078

```

```

3 0.228086 0.961092
4 0.15145 0.986334
5 0.081997      1

calc_cum=: 3 : 0
EV=:>1{ dgeev_jlapack_ cortable y
(; +/ L:0 <\ \:~ EV) % +/ EV
)

```

グラフ 主成分分析では、第一主成分と第二主成分をクロスさせた散布図で表し主成分の意味を解釈することが多い。本稿では、X に時間軸を取ってグラフに表す。次のグラフ上で、高度成長、安定成長、バブル、失われた 10 年、新たな出発が、どのように景気動向指数の指標を集約したなかでの表われるかを眺めてみよう。

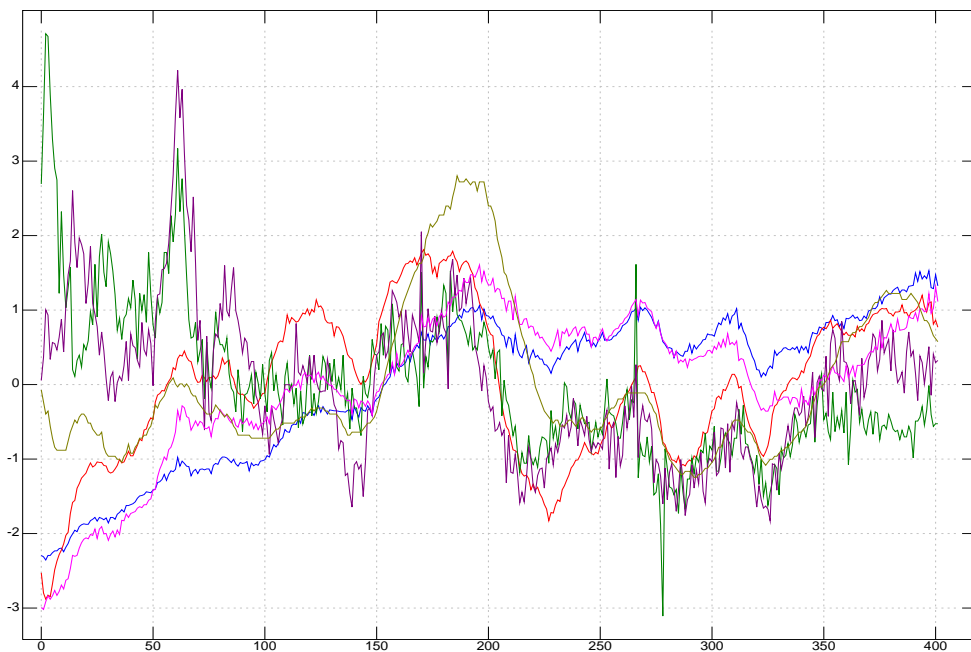


図 6.1 BI:1980/4-2007/6 の標準化

固有値の大きい順に固有ベクトルを見てゆくと第 1 主成分は全てにマイナスとなっており、景気の逆指数となっている。第 2 ベクトルは全てにプラスである。景気の高谷 同時に景気の高谷、谷を縦線のライム色とブルーで示してある。概ね景気の高谷とは一致している。

第一主成分は勢いを喪失したトレンド、第 2 主成分はサイクル、第 3 主成分はポテンシャルと考えられる。

グラフの出力先 J の temp の下に graph_sna のディレクトリを作っておく。

NB. * a=. form_sna DAT_SNA

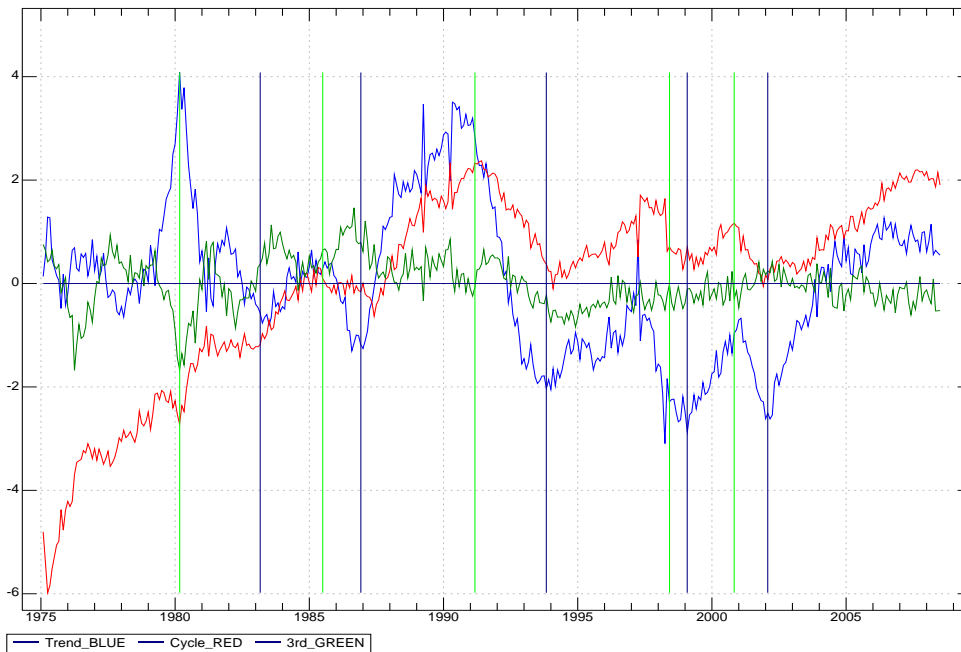


図 6.2 主成分 BI:1980/4-2007/6

```
NB. * MT plot_principal form_sna DAT_SNA
```

```
NB. * MT plot_sna_all DAT_SNA
```

```
*1
```

6.1.4 Peak Bottom

ESRI は景気の基準日付を公表している。

<http://www.esri.cao.go.jp/jp/stat/di/011221hiduke/betsuhyou2.html>

これと、主成分分析で得たグラフを重ねてある。Lime が Peak、Blue が Bottom である。
主成分分析自体に誤差もあるが、景気判断のタイミングの検証もできよう。

```
NB. * a=. form_sna DAT_SNA
```

```
NB. * MT plot_principal form_sna DAT_SNA
```

```
NB. * MT plot_sna_all DAT_SNA
```

J の temp の下に graph_sna のディレクトリを作っておく。

*1 1 枚出力するとダイアログボックスが出て止まる。do_plot.bat のファイルを作成して echo off と書いて J の config の External programs に認識させておくと連続出来たが J602 では不調である

6.1.5 関数一覧

データの読み込み	<i>read_data</i> <i>pick_bi6</i>	read_data ” a=. pick_bi6 DAT_SNA
相関係数行列	<i>cortable</i>	cortable a
固有値	<i>dgeev_jlapack_</i>	> 2 { dgeev_jlapack_ cortable a
主成分を求める	<i>calc_principal</i>	calc_principal a
主成分の累積寄与率	<i>calc_cumcalc_cum</i> a	
主成分のグラフ	<i>plot_bi6</i>	plot_bi6 a

6.1.6 Housekeeping

主成分分析の関数 `principal_main.ijs`

この章は独立した関数群で構成されている。別に読み込む必要がある。

サイレント ロード RUN → FILE でダイアログ ボックスが現れる。ここで `principal_main.ijs'` を読み込むとフォルダが一致してエラーが出なければ直ちに実行できる。

```
classes/numeric_recipe/principal/principal_main.ijs
classes/numeric_recipe/principal/principal_calc0.ijs
```

*2

LAPACK `appendix` 参照

LAPACK は FORTRAN で書かれた線形計算の関数群である。単独でも使えるが、これの一部を J の組み込み関数として利用できる。

`dgeev` は固有値の計算ルーチンであり、次により組み込むことができる

*3

```
require '~addons/math/lapack/lapack.ijs'
require '~addons/math/lapack/dgeev.ijs'
```

ロケールを用いているので `dgeev_jlapack_` と `morimoto_hokkaido_` 式の関数呼び出しで使用する。

データ 用いるデータ ESRI からダウンロードする。

*2 `principal_main.ijs` は単なる load を中心としたファイルである。ディレクトリの構成を変えるときはこのファイルを変更する。

*3 ディレクトリはユーザーにあわせ適宜変更されたい

<http://www.esri.cao.go.jp/jp/stat/menu.html>

csv ファイル (using.csv) は EXCEL などから export する。

csv ファイルに出力するときには事前に次の整形が必要である。

1. カンマは外す (あると 2 つの数に分離する)
2. 空白は 0 や 99999 を入れておく (空白があるとその行だけ右詰めすることがある)
3. 出力後のファイルをエディタ^{*4} で開いて最下行の後ろにゴミの行が入っていたら削除しておく

データの読み込み (ファイルネームやディレクトリは適宜変更)

```
DAT_SNA=:readcsv jpath '~user/classes/numeric_rcipe/data/using.csv' NB. copy
```

スクリプトのハウスキューピング `numeric_recipe_main.ijs` のファイルのハウスキューピング

1. last month の手動書き換え (ESRI データの最新月に)
NB. -----Class---INIT-----
NB. last year;month
init_sna=: 13 : '2008;06' NB. change here
2. init_anal の書き換え 12 13 24 31 ; 10 14 NB. L11 L11A;L9B L11B
最初の 12 13 24 31 は 12 13 24 31 列 (0 オリジン) (L11 L11A など) のデータの最新月などがそろっていない。(自動で削除する)
次の 10 14 (L9B L11B) はテスト段階でこれを含めると逆行列がとれなかった項である。^{*5}
NB. -----
init_anal=:init_anal2=: 3 : 0
12 13 24 31 ; 10 14 NB. L11 L11A;L9B L11B
NB. 12 13 24 31 is yellow (not full)
NB. 10 14 is not-reverse (maybe constast)
)

6.2 景気動向指数のウェーブレット分解

ウェーブレットは非定常や分散の変化する時系列に適應できる。配列を用いた多重解像度分解の計算手法を整理し、幾つかの経済時系列へ応用する。

^{*4} notepad で可

^{*5} 時々 check して、とれれば変更する。

波型の解析では大小の \sin, \cos の組み合わせで波形をシュミレートするフーリエ変換が知られている。フーリエ変換は定常波には強いが、非定常や分散の変化する波の場合は複雑な計算を必要とした。

ウェーブレット小史 ウェーブレットはフランスの石油探査会社の技師 *Jean Morlet* が 1980 年代初頭に考案したとされる。モレはマルセイユの理論物理学者 *Grossmann* のところに持ち込み、マルセイユで理論的基礎が構築された。^{*6} 1985 年にフランス人の数学者 *Yves Meyer* (ドフィンヌ大) が同僚の物理学者にウェーブレットを紹介されて興味を持ち、マルセイユへ赴いて共同研究を行い、数学面の理論化を行った。1986 年にペンシルバニア大学でメイエの講演を聴いたコンピュータ画像を研究していたフランス人の大学院生 *Stephane Mallat* が、過去にも信号や画像の処理など工学分野で類似の手法が多く研究されていることを指摘し、メイエとの共同研究で多重解像度解析の理論が誕生した。^{*7} 波の周辺には音楽家と同じ名前がよく現れる。

6.2.1 多重解像度分解・カスケード分解

次のテーブルの no を 2 個ずつ (重複しないで) 次のように計算する

足して 2 で割る	スケーリング係数	上段	<i>low pass filter</i>
引いて 2 で割る	ウェーブレット係数	下段 bold	<i>high pass filter</i>

更に $n1$ のスケーリング係数 (細字) のみを取り出して 2 個ずつを同様に組み合わせで計算する。 $n2, n3..$ も同様に計算する。

このような単純な計算で波がカスケードに分解できる。下降サンプリングと言われる。

スケーリング関数の部分のみを取り出してグラフにすると波の分解されている様子が分かる。

これは多重解像度解析と呼ばれる離散データを順次低周波から高周波への成分で表したデータの線形結合に分解する手法である。

```

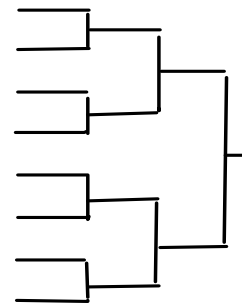
cascade_hwt  K0
+-----+
| 0 6.0653 7.3576 6.6939 5.4134 4.1042 2.9872 2.1138 |
+-----+

```

^{*6} この素敵な囁きを持つ言葉に従来からの同様の論考やアイデアも吸い寄せられて、短時間に理論が形成された。信号や画像処理系の利用が先行したので、用語や記述法も電気系か数学系の難解なものが多い

^{*7} このような単純なアルゴリズムが最近まで残っていたことが不思議である。カスケード分解は画像系の人達はメモリと計算速度に眼が行き、美しい構造を見逃していたようだ

$n_0(\text{data})$	n_1	n_2	n_3
0	3.03265	5.0292	4.34192
6.0653	-3.03265		
7.3576	7.02575	-1.99655	
6.6939	0.33185		
5.4134	4.7588	3.65465	0.687275
4.1042	0.6546		
2.9872	2.5505	1.10415	
2.1138	0.4367		



2^i
 1 2 4 8 16 32 64 128 256 512 1024 2048

図 6.3 トーナメント表

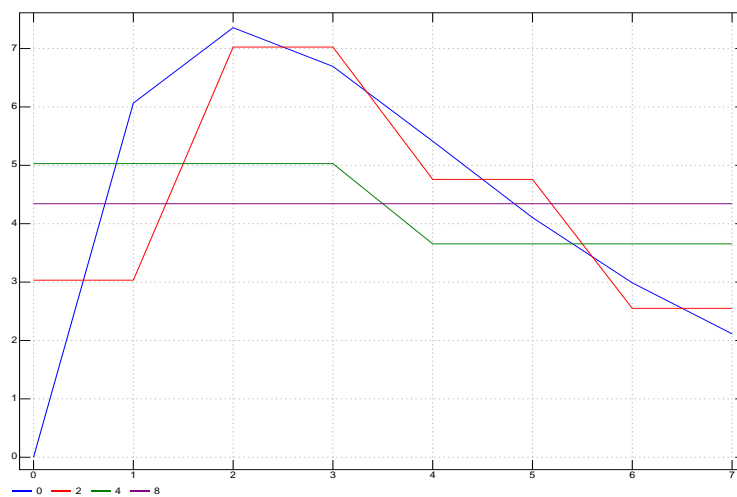


図 6.4 ハールの多重解像度分解

```
|3.03265 _3.03265 7.02575 0.33185 4.7588 0.6546 2.5505 0.4367|
+-----+
|5.0292 _1.99655 3.65465 1.10415 |
+-----+
|4.34192 0.687275 |
+-----+
```

トーナメント 囲碁や将棋の NHK 杯や高校野球のトーナメントは 1, 2 回戦を終え 32 名(校)を残す。以降は隣同士が勝ち残りを争う手法はウェーブレットの多重解像度解析と同様である。経済の時系列データ等が 2^n になることは希であり、残ったデータの取り扱いには工夫が要る。

6.2.2 高速ハール・ウェーブレット変換

高速と唱っているが、単なる線形計算である。

先の足して2で割る、引いて2で割るがハール・ウェーブレットのコアである。

Alfred Haar によって1909年に提案された。

次のマトリクスに辿り着く迄の道筋を簡単にフォローする。奇数行 (a) はスケーリング関数を、偶数行 (c) はウェーブレット関数を計算する。

$$\begin{pmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix}$$

```

mat_hwt_sub K0
0.5 0.5 0 0 0 0 0 0
0.5 -0.5 0 0 0 0 0 0
0 0 0.5 0.5 0 0 0 0
0 0 0.5 -0.5 0 0 0 0
0 0 0 0 0.5 0.5 0 0
0 0 0 0 0.5 -0.5 0 0
0 0 0 0 0 0 0.5 0.5
0 0 0 0 0 0 0.5 -0.5
    
```

ϕ Scaling	$\phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$	ψ Wavelet	$\phi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$
-------------------	--	-------------------	---

$$0\phi_{(0,0.5)} + 6.0653\phi_{(0.5,1)} = \frac{0 + 6.0653}{2}\phi_{(0,1)} + \frac{0 - 6.0653}{2}\psi_{(0,1)} = 3.03265\phi_{(0,1)} - 3.03265\psi_{(0,1)}$$

$H_2 = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}$ の形をデータ (2^n に合わせる。) の個数に展開すれば最初のマトリクスとなる。カスケードの計算過程で n が $\frac{1}{2}$ ずつ縮小していくのでマトリク

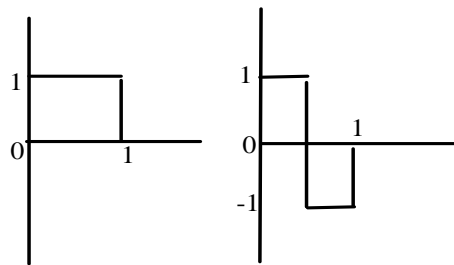


図 6.5 mother wavlet ϕ , scaling ψ

スもその都度縮小形に作り変える。

ハール・ウェーブレットの逆変換

ウェーブレットは逆変換が可能である。スケーリングとウェーブレットの値をサン
ドイチにして小さなマトリクスから逆に組み上げていく。

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} a_0 \\ c_1 \\ a_2 \\ c_3 \\ a_4 \\ c_5 \\ a_6 \\ c_7 \end{pmatrix}$$

0	$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \times \begin{pmatrix} a_0 \\ c_0 \end{pmatrix}$	<pre> a1=.4.34192 0.687275 (1 1,:1 _1) +/ . * a1 1 1 1 _1 (1 1,:1 _1) +/ . * a1 5.0292 3.65465 </pre>
---	---	---

```

3 reverse_hwt cascade_hwt K0
5.0292 3.65465
2 reverse_hwt cascade_hwt K0
3.03265 7.02575 4.7588 2.5505
1 reverse_hwt cascade_hwt K0
0 6.0653 7.3576 6.6939 5.4134 4.1042 2.9872 2.1138

```

6.2.3 ドビッシー・ウェーブレット

*8*9

ドビッシー基底

ドビッシは 1987 年にカスケードアルゴリズムでのコンパクト台を持つウェーブレット基底を考案した。

D4

N2(D4) の場合。なお、(N1(D2)) は *Haar* 基底と同じである。

高速ドビッシウェーブレット変換とエッジ問題

D4 の係数を次式によりマトリクスにする。 $N = 2^k$ である。最後の 2 行がマトリクスをはみ出す。データを周期的と見なして回り込みを許すか最後の 2 行のデータを *AR* (自己回帰) などで推計してこの部分の計算を工夫する必要がある。

$$\begin{pmatrix} a_0^{n-1} \\ a_1^{n-1} \\ a_2^{n-1} \\ a_3^{n-1} \\ a_4^{n-1} \\ a_5^{n-1} \\ a_6^{n-1} \\ a_7^{n-1} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{pmatrix} \begin{pmatrix} a_0^n \\ a_1^n \\ a_2^n \\ a_3^n \\ a_4^n \\ a_5^n \\ a_6^n \\ a_7^n \end{pmatrix}$$

*10

*8 響きの良いフランス語読みとする。

*9 *Ingrid Daubechies* ベルギー生まれの女性研究者。ベルギーで ph.D を取り、ベル研究所などを経て現在プリンストン大学。一時、マルセイユグループにいたようだ。

*10 トーナメントの階層と同じで最後の 2 段に特に効いている

Scaling 規格化の場合。規格化を行わない

場合は $\frac{1 + \sqrt{3}}{4}$ の様に $\sqrt{2}$ を落とす。

双方とも用いられているようだ。

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{3 + \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{3 - \sqrt{3}}{4\sqrt{2}}$$

$$h_0 = \frac{1 - \sqrt{3}}{4\sqrt{2}}$$

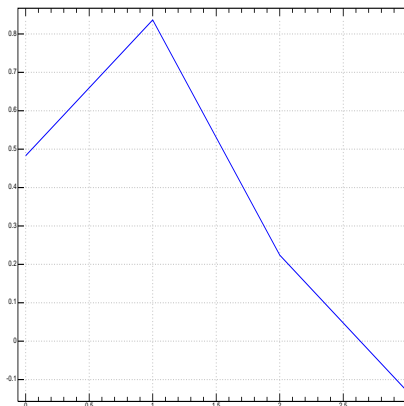


図 6.6 mother wavletφ

Wavelet $g_0 = h_3$

$$g_1 = -h_2$$

$$g_2 = h_1$$

$$g_3 = -h_0$$

```
dp_40 ''
0.482963 0.836516 0.224144 _0.12941
dp_41 ''
0.683013 1.18301 0.316987 _0.183013
```

```
6j2 ": mat_dwt4_sub K0 NB. 配列部分 回り込み許容
0.48 0.84 0.22 _0.13 0.00 0.00 0.00 0.00
_0.13 _0.22 0.84 _0.48 0.00 0.00 0.00 0.00
0.00 0.00 0.48 0.84 0.22 _0.13 0.00 0.00
0.00 0.00 _0.13 _0.22 0.84 _0.48 0.00 0.00
0.00 0.00 0.00 0.00 0.48 0.84 0.22 _0.13
0.00 0.00 0.00 0.00 _0.13 _0.22 0.84 _0.48
0.22 _0.13 0.00 0.00 0.00 0.00 0.48 0.84
0.84 _0.48 0.00 0.00 0.00 0.00 _0.13 _0.22
```

6.2.4 D(4) の逆変換

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} h_2 & g_2 & h_0 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_3 & g_3 & h_1 & g_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & h_2 & g_2 & h_0 & g_0 & 0 & 0 & 0 & 0 \\ & & h_3 & g_3 & h_1 & g_1 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & h_2 & g_2 & h_0 & g_0 & 0 & 0 \\ & & 0 & 0 & h_3 & g_3 & h_1 & g_1 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & h_2 & g_2 & h_0 & g_0 \\ & & 0 & 0 & 0 & 0 & h_3 & g_3 & h_1 & g_1 \end{pmatrix} \times \begin{pmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{pmatrix}$$

6.2.5 景気動向指数への応用

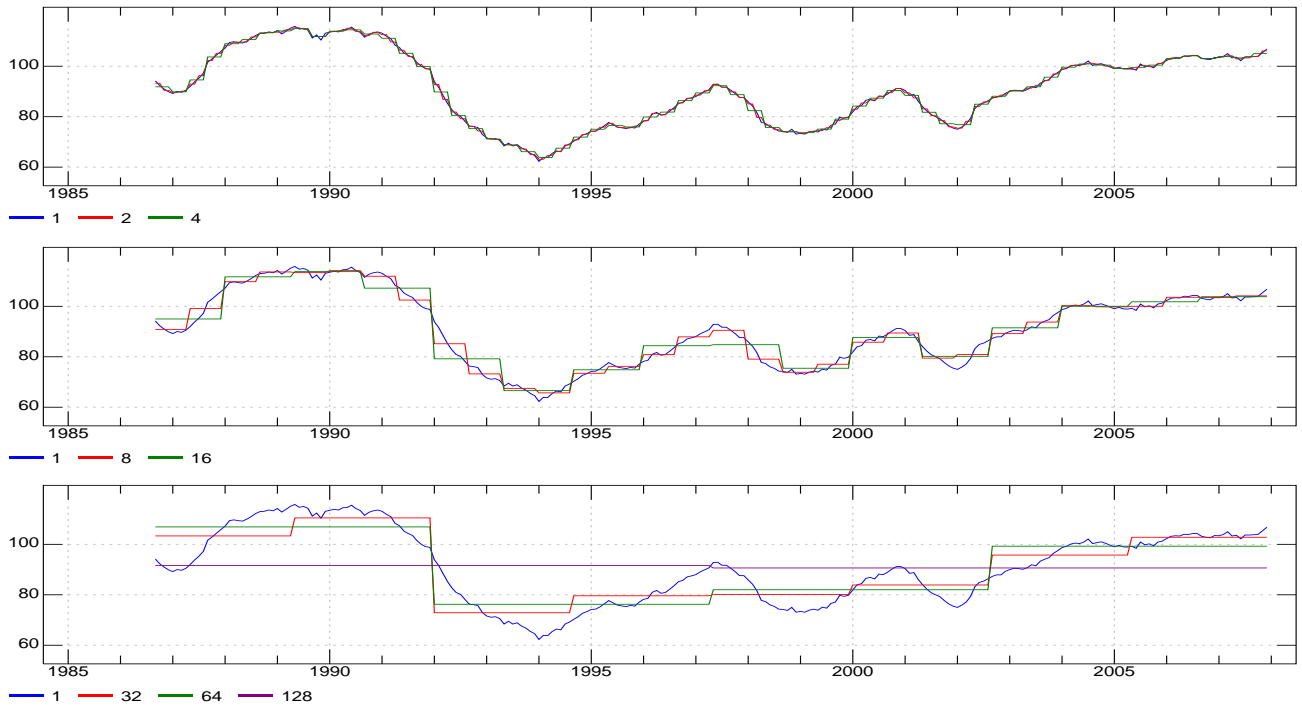


図 6.7 C17 電力消費量 Haar

6.2.6 関数一覧

classify	main	sub
file	<i>wavelet_0.ijs</i>	
haar	cascade_hwt	pick_hwt_sub reverse_hwt_sub mat_hwt_sub
daubechies	cascade_dwt40 NB. norm cascade_dwt41 NB. non norm	mat_dwt4_sub0 dp4_0 NB. norm mat_dwt4_sub1 dp4_1 NB.non norm

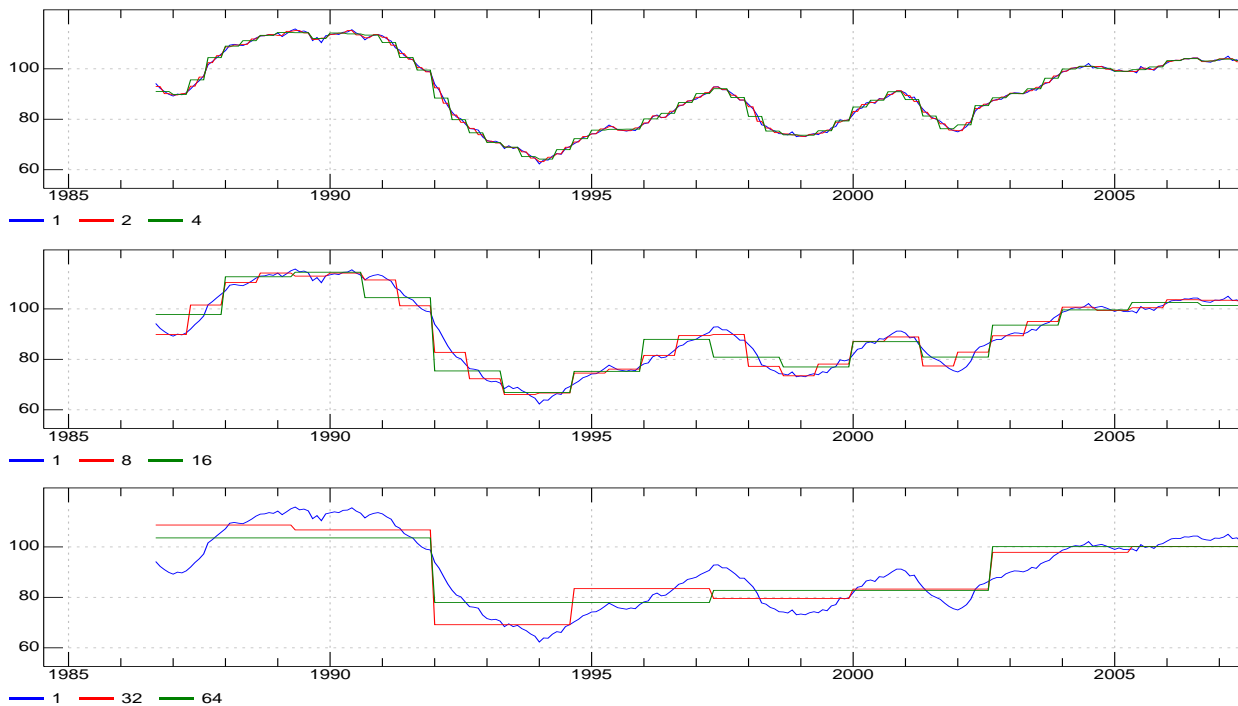


図 6.8 C17 電力消費量 dw4

common		pick_scale pick_wavelet norm0
plot	plot_cascade plot_cascade2	
sna	wavelet_sna plot_cascade3	mk_xaxis_wavelet read_trend_data

6.2.7 Script

Haar Wavelet

NB. 内積 (+/ . *) でハールマトリクスとデータを計算する。

2 個になるまで反復

```
cascade_hwt=: 3 : 0
```

NB. cascade Haar wavelet

```
TIME_IND=: I. (2^i. # y) e. # y NB. index for repeat
```

```
ANS=.< TMP=. y
```

```

for_CTR. i. TIME_IND do.
MAT=. mat_hhwt_sub TMP
N0=. MAT +/ . * TMP
ANS=. ANS,<N0
TMP=. pick_hwt_sub N0 NB. pick upper of sandwich
end.
,. ANS
)

```

NB. スケーリング部分の値のみを取り出す

```
pick_hwt_sub=: 3 : '(-.*(2|i.# y))# y' NB. pick odd is scaling
```

NB. マトリクスへの展開。|.(("0 1) と回転させて実数 (0.5,-0.5 etc) を所定位置へ配置

```

mat_hwt_sub=: 3 : 0
NB. mat for highspeed Haar wavelet
INDX =.-: # y NB. half of number
TMP0=. ;("2),.INDX # < 2 2 $ 0.5 0.5 0.5 _0.5
TMP1=. TMP0 ,. (;(# TMP0),((# y)-2))$0 NB. add 0 part
IND3=-. ;2 # L:0 {@> INDX{. +: i. # y NB. make rotate index
IND3 |. ("0 1) TMP1
)

```

NB. スケーリングの値を取り出し、グラフ用に Expand。汎用に副詞 (1:0) としている。

```

plot_cascade=: 1 : 0
TMP0=: u y
IND=: {@> }.2^i. # TMP0
TMP1=. (-.&* (L:0) 2 | L:0 i. L:0 , # L:0 }.TMP0) # L:0 }. TMP0
DAT=. ;("1)({. TMP0),IND # L:0 TMP1
pd 'reset'
pd 'keypos open bottom'
pd 'keystyle open horizontal'
pd 'key ',": 0,;IND
pd DAT

```

```
pd 'show'
)
```

6.3 景気動向指数とランダムウォーク

ESRI (経済社会総合研究所) が作成する景気動向指数は 2008 年 4 月から CI(Composit index) 中心に改訂された。

http://www.esri.cao.go.jp/jp/stat/di/menu_di.html

この CI(先行一致 遅行) の指数を用いてランダムウォークを作成し、このランダムウォークをベンチマークとして各指数と比較してみる。筈にも成長の良いのとそうでないのがあり、栗の実にも実のつきの良いのと悪いのがある。

ランダムウォークは前後 2 月の増減をプラス、マイナスで表す。時間を極限化したブラウン運動と異なり、一步一步プラスかマイナスに必ず進む。従ってベンチマークに成りうる。

6.3.1 ランダムウォーク

経過と解説

CI-L CI の Leading の最初の 10 個の例

```
10{. ; 1{"1 CI
```

```
76.1 76 73.7 73.7 72.3 72.1 71 69.9 69.7 70.1
```

重複した 2 か月の比較 2<\ a のように非負の 2 は 2 個ずつ重複して組み合わせる。rotate(|.) で前後を反転

```
2<\ a
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|76.1 76|76 73.7|73.7 73.7|73.7 72.3|72.3 72.1|72.1 71|71 69.9|69.9 69.7
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

引く 2(-/"1@|.)\ a

```
_0.1 _2.3 0 _1.4 _0.2 _1.1 _1.1 _0.2 0.4
```

プラスマイナス プラスまたは 0 かマイナスかを判別する。0 は>: で処理

```
(2(-/"1@|.)\ a)>: 0
```

```
0 0 1 0 0 0 0 0 1
```

ランダムウォークに変換 +1 -1 に変換する

```
((2(-/"1@|.)\ a)>: 0){_1 1
```

```
_1 _1 1 _1 _1 _1 _1 _1 1
```

累積 累積する。<\ で一気に成


```

5{. <\ ((2(-/"1@|.)\ a)>: 0){_1 1
+--+-----+-----+-----+-----+-----+
|_1|_1 _1|_1 _1 1|_1 _1 1 _1|_1 _1 1 _1 _1|
+--+-----+-----+-----+-----+

```

累積の和 和を求める。

```

(+/\ ((2(-/"1@|.)\ a)>: 0){_1 1
_1 _2 _1 _2 _3 _4 _5 _6 _5

```

6.3.2 グラフ

データは 1980/1 から 2008/6 まで 2005=100

ESRI 発表の景気の山(ライム)谷(ブルー)を付加した。^{*11}

L は CI の方が RW より振幅が小さい。100 のラインは上方にある。ピークがよく双耳峰になることがある。

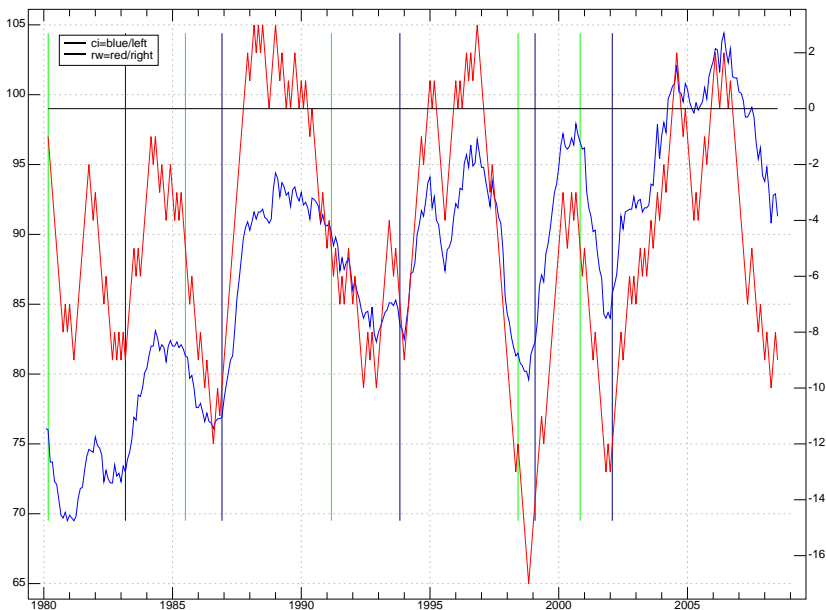


図 6.9 Leading

バブル経済の時は CI がふき上がっている。最近の動向は歩調がそろっている。
収縮期の減少は LG の方が小さい。

6.4 Reference

Yves Nievergelt, 松本・雑元・茂呂訳「ウェーブレット変換の基礎」森北出版 2004

^{*11} 最新のピーク時期は未だ発表されていない。

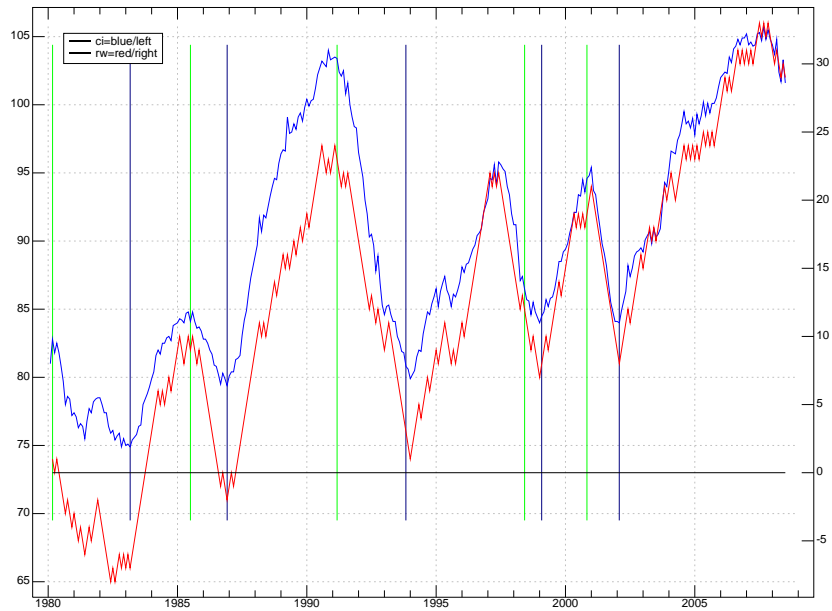


図 6.10 Coincidence

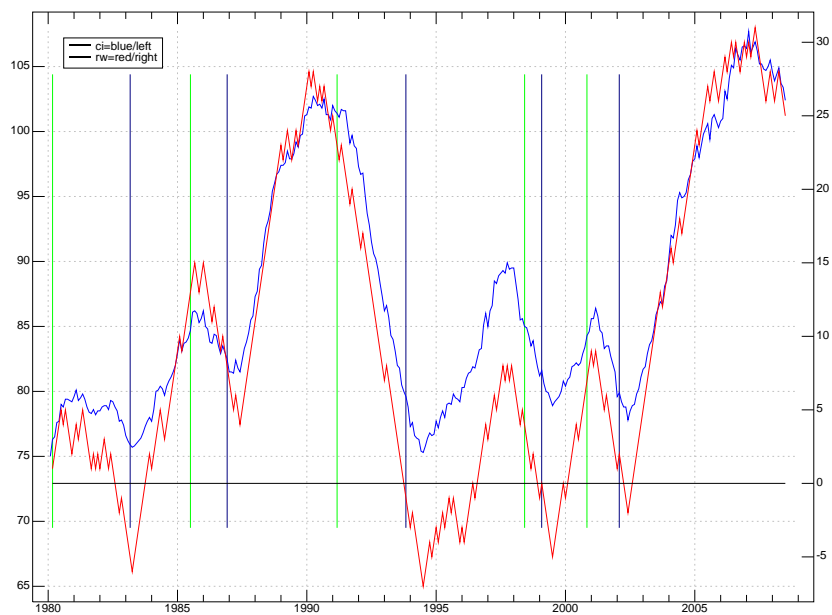


図 6.11 Lagged

金谷健一「これなら分かる応用数学教室」共立出版 2003

佐竹玄一郎 編著「経済の統計的分析」中央経済社 2004

第 7 章

シンプレックス法

J に含まれている強力なスクリプトによる線形計画法の一つであるシンプレックス法

しんぷれっくすほう@シンプレックス法の計算を行う。

線形計画は、変数が多数の 1 次不等式によって制約されている場合に、与えられた 1 次式を最適化する。これの効率的な解法がシンプレックス法である。シンプレックス法は G.B.Dantzig(1914-2005) が 1947 年に提唱した。

7.1 | simplexnr

J の `system/packages/math` に Henly Rich の貢献による `simplexnr.ijs` が入っている。

この `simplexnr` を用いて、シンプレックス法を解いてみよう。

*1

Worked Example J の `simplexnr` の example

Maximize $x_1 + x_2 + 3x_3 - 0.5x_4$

subject to

$$x_1 + 2x_3 \leq 740$$

$$2x_2 - 7x_4 \leq 0$$

$$x_2 - x_3 + 2x_4 \geq 0.5$$

$$x_1 + x_2 + x_3 + x_4 = 9$$

入力フォーマットは box を; で連結する。最後のボックスは \leq is _1, \geq is 1, $=$ is 0 を付加する。

```
mat3
+-----+-----+-----+
| 1 1 3 _0.5 | 1 0 2 0 | 740 0 0.5 9 | _1 _1 1 0 |
|           | 0 2 0 _7 |           |           |
```

*1 simplexnr は H.Ritch が Numeric Receipt より作成したもので精度がよい。

```

|          |0 1 -1 2|          |          |
|          |1 1  1 1|          |          |
+-----+-----+-----+-----+
      simplexnr mat3
+-----+-----+
|0|0 3.325 4.725 0.95|17.025|
+-----+-----+

```

Working Example スラック変数

不等式の小さい方の辺に非負の数を加えると等式になる。

この加える変数をスラック変数という。式は = になる。

スラック変数は1個 (λ) の場合と、単位行列に相応する x_n を付加する場合がある。

$$f = 29x_1 + 45x_2 \rightarrow \max$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

$$2x_1 + 8x_2 + x_3 = 60$$

$$4x_1 + 4x_2 + x_4 = 60$$

```
ax7=. 29 45 0 0 ;(2 4 $ 2 8 1 0 4 4 0 1);60 60;0 0
```

```

+-----+-----+-----+-----+
|29 45 0 0|2 8 1 0|60 60|0 0|
|          |4 4 0 1|          |          |
+-----+-----+-----+-----+

```

```
simplexnr ax7
```

```

+-----+-----+
|0|10 5 0 0|515|
+-----+-----+

```

$$x_1 = 10, x_2 = 5, f = 515$$

Working example $f = 150x_1 + 300x_2 \rightarrow \max$

$$2x_1 + x_2 + x_3 = 16$$

$$x_1 + x_2 + x_4 = 8$$

$$x_2 + x_5 = 3.5$$

```
ax3=.150 300 0 0 0 ;(3 5 $ 2 1 1 0 0 1 1 0 1 0 0 1 0 0 1);16 8 3.5 ;0 0 0
```

```

+-----+-----+-----+-----+
|150 300 0 0 0|2 1 1 0 0|16 8 3.5|0 0 0|
|          |1 1 0 1 0|          |          |

```

```
|          |0 1 0 0 1|          |          |
+-----+-----+-----+-----+
```

```
simplexnr ax3
```

```
+-----+-----+-----+-----+
```

```
|0|4.5 3.5 3.5 0 0|1725|
```

```
+-----+-----+-----+-----+
```

$x_1 = 4.5, x_2 = 3.5, x_3 = 3.5, f = 1725$

Working Example 双対問題

$f = x_1 + 2x_2 \rightarrow \max$

$x_1 \geq 0, x_2 \geq 0$

$2x_1 - x_2 \leq 7$

$3x_1 + x_2 \leq 10$

$-x_1 + 2x_2 \leq 18$

ax5=: 1 2 ;(3 2 \$ 2 _1 3 1 _1 2);7 10 18;_1 _1 _1

```
simplexnr ax5
```

```
+-----+-----+-----+-----+
```

```
|1 2| 2 _1|7 10 18|_1 _1 _1|
```

```
| | 3 1|          |          |
```

```
| |_1 2|          |          |
```

```
+-----+-----+-----+-----+
```

```
simplexnr ax5
```

```
+-----+-----+-----+-----+
```

```
|0|0.285714 9.14286|18.5714|
```

```
+-----+-----+-----+-----+
```

$x_1 = \frac{2}{7}, x_2 = \frac{64}{7}, f = \frac{130}{7}$

スラック変数を加えた場合

ax4=. 1 2 0 0 0 ;(3 5 \$ 2 _1 1 0 0 3 1 0 1 0 _1 2 0 0 1);7 10 18 ; 0 0 0

```
simplexnr ax4
```

```
+-----+-----+-----+-----+
```

```
|1 2 0 0 0| 2 _1 1 0 0|7 10 18|0 0 0|
```

```
|          | 3 1 0 1 0|          |          |
```

```
|          |_1 2 0 0 1|          |          |
```

```
+-----+-----+-----+-----+
```

```
simplexnr ax4
```

```

+-----+
|0|0.285714 9.14286 15.5714 0 0|18.5714|
+-----+

```

双対問題は転置した形である。

$$g = 7y_1 + 10y_2 + 18y_3 \rightarrow \min$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

$$2y_1 + 3y_2 + y_3 \geq 1$$

$$-y_1 + y_2 + 2y_3 \geq 2$$

標準形に直す

$$g' = -7y_1 - 10y_2 - 18y_3 \rightarrow \max$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

$$-2y_1 - 3y_2 + y_3 \leq 1$$

$$y_1 + y_2 - 2y_3 \leq 2$$

```

ax6=. _7 _10 _18;(2 3 $ _2 _3 1 1 _1 _2);_1 _2 ;_1 _1

```

```

+-----+
|_7 _10 _18|_2 _3 1|_1 _2|_1 _1|
|          | 1 _1 _2|      |
+-----+

```

```

simplexnr ax6

```

```

+-----+
|0|0 0.571429 0.714286|_18.5714|
+-----+

```

$$y_1 = 0, x_2 = \frac{4}{7}, y_3 = \frac{5}{7}, f = \frac{130}{7}$$

7.1.1 関数一覧

シンプレックス法	<i>simplexnr</i>	simplexnr table
----------	------------------	-----------------

7.1.2 Reference

金谷健一 これなら分かる最適化数学 共立出版 2005

第 8 章

産業連関表

APL と J の父 *K.E.Iverson* は最初のコンピューターによる産業連関表の計算をサポートし、数学に強いコンピューター言語の研究の契機となった。



図 8.1 W.Leontief

産業連関表の原型は W・レオンチェフ (Wassily Leontief 1906-1992) が 1936 年に独力で完成させた。レオンチェフは当時のレニングラードで生まれ、レニングラード大学で哲学、社会学、経済学を専攻した。革命の混乱期の 1925 年に出国を許され、東ベルリンのフンボルト大学で投入産出分析の研究を続け、1931 年にアメリカに渡り、1932 年からハーバード大学に在籍、1946 年に経済学部教授に就任した。1973 年に経済学でノーベル記念賞を受賞している。IBM がハーバード大学の協力を得て開発した、真空管コンピューター MARKII の第 1 号はペンタゴンに、第 2 号はハーバード大学に入っている。このうなりをあげるリレー式の初期のコンピューターを使ったレオンチェフの産業連関表の計算をハーバードの数学とコンピューターサイエンスの研究者であった若き日の *K・E・Iverson* がサポートしたことは知られている。当時のコンピューターの能力をから、部門セクタを 500 から 50 に縮小して、48 時間回し続けて解を得た。(現在では 500 で 1 秒以内)

8.1 産業連関表のフォーマット

		中間 需要	最終需要	輸入	総生産
		農業 工業			
中間 農業		X_{11}	F_1	EX_1	X_1
投入 工業		X_{21}	F_2	EX_2	X_2
	粗付加価値	V_1			
	総生産	X_1			

$$\text{需給バランス式 } X_{11} + X_{12} + F_1 = X_1$$

$$\text{(行方向) } X_{21} + X_{22} + F_2 = X_2$$

$$X_{11} + X_{21} + V_1 = X_1$$

$$\text{(縦方向) } X_{12} + X_{22} + V_2 = X_2$$

8.1.1 投入係数表

	農業	工業
農業	$a_{11} = \frac{X_{11}}{X_1}$	$a_{12} = \frac{X_{12}}{X_2}$
工業	$a_{21} = \frac{X_{21}}{X_1}$	$a_{22} = \frac{X_{22}}{X_2}$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

需給バランス式の行列表示

$$AX + F = X$$

Working Example 2×2 表

		中間 需要	最終 需要	輸入	総生産
		農業 工業	消費 投資	輸出	
中間 農業		12	15	5	40
投入 工業		8	60	28	100
	粗付加価値	20	25		
	総生産	40	100		

入力テーブルの形式

粗付加価値を右の最後の行に付加して、マトリクスを整えた。ここではこの入力形式を用いる。

NB. NB. shirasago 2nd tab 9-9

DAT_SHIRA2=: 12 15 18 5 5 _15 , 8 60 12 10 28 _18,:20 25

DAT_SHIRA2
 12 15 18 5 5 _15
 8 60 12 10 28 _18
 20 25 0 0 0 0

投入係数	$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \frac{12}{40} & \frac{15}{100} \\ \frac{8}{40} & \frac{60}{100} \end{pmatrix} = \begin{pmatrix} 0.3 & 0.15 \\ 0.2 & 0.6 \end{pmatrix}$ $\begin{pmatrix} \frac{X_{11}}{X_1} & \frac{X_{12}}{X_2} \\ \frac{X_{21}}{X_1} & \frac{X_{22}}{X_2} \end{pmatrix}$
レオンテ イエフ逆 行列	$(I - A)^{-1}$ 1.6 0.6 0.8 2.8
レオンテ イエフ逆 行列（輸 出入型）	$B = (I - (I - \bar{M})A)^{-1}$ $\frac{1}{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.3 & 0 \\ 0 & 0.2 \end{pmatrix} \right) \begin{pmatrix} 0.3 & 0.15 \\ 0.2 & 0.6 \end{pmatrix}}$ 1.3198 0.266497 0.406091 2.00508
輸入係数	$m_i = \frac{M_i}{\sum_{j=1}^n X_{ij} + F_{ci} + F_{li}} =$ 農業： $m_1 = \frac{M_1}{X_{11} + X_{12} + F_{c1} + F_{l1}} = \frac{15}{12 + 15 + 18 + 5} = 0.3$ 工業： $m_2 = \frac{M_2}{X_{21} + X_{22} + F_{c2} + F_{l2}} = \frac{18}{8 + 60 + 12 + 10} = 0.2$ $\bar{M} = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.2 \end{pmatrix}$

leon_inv_sub2 DAT_SHIRO2	
B	M-bar
+-----+-----+	
1.3198 0.266497 0.3 0	
0.406091 2.00508 0 0.2	
+-----+-----+	
レオンチェフ逆行列	$B = \frac{1}{I - (I - \bar{M})A} \begin{pmatrix} 1.3198 & 0.266497 \\ 0.406091 & 2.00508 \end{pmatrix}$
国内自給率	$\Gamma = I - \bar{M} \begin{pmatrix} 0.7 & 0 \\ 0 & 0.8 \end{pmatrix}$
消費の生産誘発額	$B\Gamma \begin{pmatrix} 18 \\ 12 \end{pmatrix}$
投資の生産誘発額	$B\Gamma \begin{pmatrix} 5 \\ 10 \end{pmatrix}$
輸出の生産誘発額	$B \begin{pmatrix} 5 \\ 28 \end{pmatrix}$
誘発額 消費、投資、輸出の誘発額を求める	

calc_leon_inv2 DAT_SHIRA2	
Agriculture Industrial	
+-----+-----+	
consumption 19.1878 24.3655	
+-----+-----+	
investment 6.75127 17.4619	
+-----+-----+	
export 14.0609 58.1726	
+-----+-----+	

誘発係数 消費、投資、輸出の誘発係数を求める。

消費の生産誘発係数	$\frac{BGF_c}{iF_c}$	BGF_c	$\begin{pmatrix} 18 \\ 12 \end{pmatrix}$
投資の生産誘発係数	$\frac{BGF_I}{iF_I}$	BGF_c	$\begin{pmatrix} 5 \\ 10 \end{pmatrix}$
輸出の生産誘発係数	$\frac{BF_E}{iF_E}$	BGF_c	$\begin{pmatrix} 5 \\ 28 \end{pmatrix}$

```
calc_leon_inv3 DAT_SHIRA2
```

```
Agriculture Industrial
```

```
+-----+-----+
|consumption|0.639594 0.812183|
+-----+-----+
|investment |0.450085 1.16413 |
+-----+-----+
|export      |0.426088 1.76281 |
+-----+-----+
```

4×4 産業連関表 拡張モデル

4×4の拡張モデルで検証してみよう。入力フォーマットは2×2と同様で、粗付加価値を最終列に移してある。

	農業	製造業	建設	サービス	消費	投資	輸出	輸入	粗付加価値
農業	6	21	3	2	17	2	1	12	20
製造業	8	120	35	24	58	33	45	23	105
建設	2	18	1	6	0	73	0	0	45
サービス	4	36	16	28	105	6	8	3	140

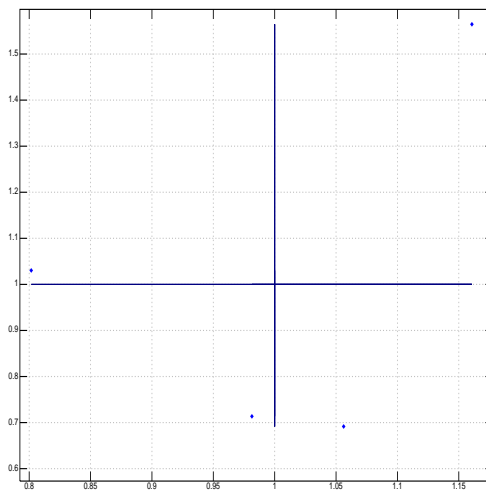
```
DAT_SHIRA3
```

```
6 21 3 2 17 2 1 _12
8 120 35 24 58 33 45 _23
2 18 1 6 0 73 0 0
4 36 16 28 105 6 8 _3
20 105 45 140 0 0 0 0
```

ここで大切なことは、輸入はマイナスで入力することである。

```
leon_inf_resp DAT_SHIRA3
```

	農業	工業	建設	サービス	感応度係数
農業	1.15891	0.109245	0.0665054	0.0265408	0.713543
工業	0.41731	1.71777	0.605561	0.244083	1.56459
建設	0.090065	0.117789	1.05877	0.0526786	0.691576
サービス	0.206	0.269412	0.284049	1.20599	1.03029
影響力係数	0.981452	1.16069	1.0562	0.801655	0



<i>Service</i>	<i>Industry</i>
<i>Agriculture</i>	<i>Construction</i>

図 8.2 X 影響力係数 Y 感応度係数

生産誘発額

calc_leon_inv2 DAT_SHIRA3

農業 工業 建設 サービス

```

+-----+-----+-----+-----+
|consumption|23.6218 122.047 12.8835 141.692|
+-----+-----+-----+-----+
|investment |10.091 98.2828 81.3045 36.3303 |
+-----+-----+-----+-----+
|export      |6.28726 79.6698 5.81199 21.9775|
+-----+-----+-----+-----+
    
```

生産誘発係数

calc_leon_inv3 DAT_SHIRA3

農業 工業 建設 サービス

```

+-----+-----+-----+-----+
|consumption|0.131232 0.678041 0.0715752 0.787179|
+-----+-----+-----+-----+
    
```

```
|investment |0.0885172 0.86213 0.713197 0.318687 |
+-----+-----+
|export     |0.116431 1.47537 0.10763 0.40699   |
+-----+-----+
```

8.1.2 関数一覧

$(I - (I - \bar{M})A)^{-1}$ 型レオン ティエフ逆行列と輸入係数	<i>leon_inv_sub2</i>	leon_inv_sub2 y
感応度係数と影響力係数を 求める	<i>leon_inf_resp</i>	leon_inf_resp y
生産誘発額を求める	<i>calc_leon_inv2</i>	calc_leon_inv2 y
生産誘発係数を求める	<i>calc_leon_inv3</i>	calc_leon_inv3 y

第9章

時系列解析 Time Series

最初の章で J を用いた易しい時系列モデルの作成方法を示した。ここでは ユール・ウオーカー法 やレビンソンのアルゴリズム、AIC によるモデル選択の方法を示す。少ない変数による多変量時系列モデルは中規模の経済モデルに匹敵すると言われる。レビンソンのアルゴリズムを用いて、多変量 ユール・ウオーカー法 を作成する。インパルス応答関数により、変数相互の時間による相関の様子を分析する。

時系列とは、一定の等間隔にとらえられた時間間隔毎に並んだ観測値の列であり、経済や工業、医学を初めとし、様々な分野で観測されている。気温、GDP、貨幣供給量、株価、計器の圧力、脳波、etc.

```
load time series    time_main.ijs
```

time series は独立の class である。

```
classes/calculus/time/time_main.ijs
```

RUN → Files でダイアログボックスがでてくる。ここからサイレントロードを行う。^{*1}

9.1 自己相関

ある確率過程が定常であるならば、時間の経過での変化の関係をとらえることによって、時間領域での性質を要約することができる。

時系列 y_n と時刻を k だけシフトした y_{n-k} を時系列 y_n の自己共分散関数 (autocovariance function) といい、時間が y_{n-k} だけ経過したとき、確率変数がどのように経過するかを示している。

自己相関係数 (autocorrelation) は、標準偏差で規格化された自己共分散であり、自己共分散、自己相関関数は時系列の過去の変動との関連の強さを表現したものである。

ここでは自己共分散関数と自己相関関数を求める。

$$ACF(k) = \frac{\sum_{t=k-1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

^{*1} フォルダの構成が異なる時は main ファイルを修正するか 1 本ずつロードする

Working Example

1. DN90=:3 6 8 4 4 8

定常過程の場合、 $X(t)$ の期待値は、時刻 t に依存しない。自己共分散 $Cov[X(t_1), X(t_2)]$ も時刻 t_1, t_2 には依存しないで、その時間差 $\tau = t_1 - t_2$ のみに依存する。

順を追って書き下してみよう。

経過と説明

1. (`<\.DN90`), `.|.(<\DN90`

最初に例題 DN90 のすべての組み合わせを作成する。時間 t は新しいほうが右にくる。右の欄は上から 原型、 $Y_t, Y_{t-1}, Y_{t-2}, Y_{t-3} \dots$ と新しい順に落としている。左の欄は頭の方から古い時間を順に落とし、右の欄とサイズを合わせた組み合わせを作成する。

```
+-----+-----+
|3 6 8 4 4 8|3 6 8 4 4 8|
+-----+-----+
|6 8 4 4 8 |3 6 8 4 4 |
+-----+-----+
|8 4 4 8    |3 6 8 4    |
+-----+-----+
|4 4 8      |3 6 8      |
+-----+-----+
|4 8        |3 6        |
+-----+-----+
|8          |3          |
+-----+-----+
```

2. 各要素から平均を引く。ACF では \bar{Y} に原系列の平均を用いる。

*2

```
tmp=: ((<\.DN90),.|. @(<\ ) DN90)- L:0 (+/%#) DN90
```

```
+-----+-----+
|_2.5 0.5 2.5 _1.5 _1.5 2.5|_2.5 0.5 2.5 _1.5 _1.5 2.5|
+-----+-----+
|0.5 2.5 _1.5 _1.5 2.5    |_2.5 0.5 2.5 _1.5 _1.5    |
+-----+-----+
```

*2 Paerson ではずらした後の各個数の平均を用いる。

```

|2.5 _1.5 _1.5 2.5          |_2.5 0.5 2.5 _1.5          |
+-----+-----+
|_1.5 _1.5 2.5            |_2.5 0.5 2.5              |
+-----+-----+
|_1.5 2.5                  |_2.5 0.5                  |
+-----+-----+
|2.5                        |_2.5                        |
+-----+-----+

```

3. 各行毎に左右の要素を掛け合わせる。

```

*/ L:0 ,. ({"1 tmp) ,: L:0 {"1 tmp
+-----+
|6.25 0.25 6.25 2.25 2.25 6.25|
+-----+
|_1.25 1.25 _3.75 2.25 _3.75 |
+-----+
|_6.25 _0.75 _3.75 _3.75      |
+-----+
|3.75 _0.75 6.25                |
+-----+
|3.75 1.25                      |
+-----+
|_6.25                          |
+-----+

```

4. 各行 (次数) ごとに合計すると自己共分散が求まる

```

;+/ L:0 TMP1

```

```

23.5 _5.25 _14.5 9.25 5 _6.25

```

5. 分散で除算する

```

TMP1 % (+/ )@:( ^&2)@ dev y'

```

NB. 自己相関

```

1 _0.223404 _0.617021 0.393617 0.212766 _0.265957

```

6. ACF

```

acf DN90

```

NB. Cov % Var

```

1 _0.223404 _0.617021 0.393617 0.212766 _0.265957

```

$A_t A_t$	$A_t A_{t-1}$	$A_t A_{t-2}$	$A_t A_{t-3}$	$A_t A_{t-4}$	$A_t A_{t-5}$
1	-0.223404	-0.617021	0.393617	0.212766	-0.265957

J 言語ノート

1. J の `outfix(<\.)`, `infix(<\)` と `rotate(1.)` を組合せ, $(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})$ を纏めて作っている.
2. `L:0` はボックスの中へ除算 (-) を作用させる。

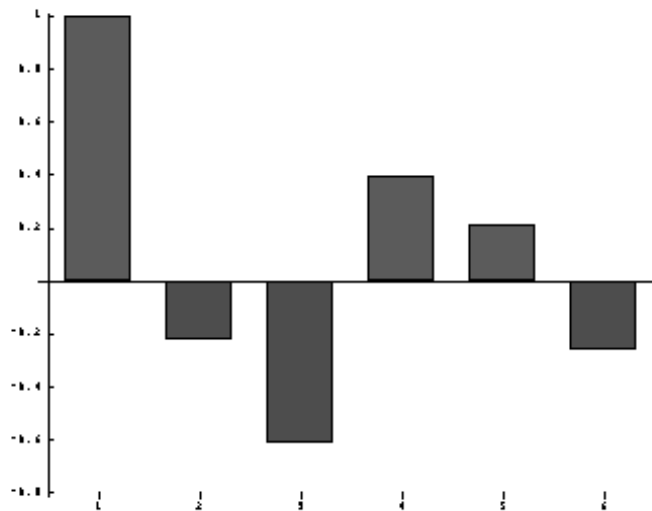


図 9.1 ACF

Script

次のスクリプトは, 自己相関関数を一度に求めてしまうものである。

$$ACF(k) = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

SCRIPT

```

    acf=:3 : 0
NB. refine ACF2 code
NB. ACF autocorrelation coefficients (many times at once)
C1=. (<\. y) ,. |.@(<\)y
C3=. C1 - L:0 C2=. mean y NB. dev
C4=.> +/ L:0 */ L:0 <"2 >C3 NB. change box style
NB. sum (Y_t - Y')*(Y_(t-k) - Y')
C4 % +/ (y - mean y)^2 NB. C4 % VAR y
)

```

J 言語ノート Tacit 関数型の定義

J は洗練された関数型言語を目指して、関数定義のためのモードとして Tacit definition(陰関数定義)を採用した。これは、従来の Explicit definition(明示的定義)と同時に、混在して使用できる。

Tacit は関数記述のためのエレガントで優れた柔軟な手法で、数式との対比が分かりやすい反面スクリプトのメンテナンスの労力は大きい。数学などの関数は、業務プログラムと異なり、一度定義してしまえばそれほど手入れは必要ないので、学術系に tacit 愛好者が多い。

tacit(関数型)で定義すると次のようになる。

ACF は \bar{Y} が同一なので, dev:=-(+ / % #) で最初に平均を引いてある.acf_0 が分子、acf_1 が分母を定義して、最後に除算している。一行での記述も可能である。

NB. ACF tacit definition(monad)

NB. Usage: acf_t y.

acf_01:=>@(+ / L:0)@((<\.) * L:0 |.@(<\)) NB. numerator

acf_02:= (+ /)&(^&2) NB. denominator

acf_t:= (acf_01 % acf_02)@dev NB. Main

acf_t2:=((>@(+ / L:0)@((<\.) * L:0 |.@(<\))) % (+ /)&(^&2))@dev NB. ACF

9.1.1 偏自己相関 PACF

偏自己相関は現在の観測値と k 期前の観測値の間の相関を表現するもので、高次の自己相関は低次の自己相関の影響を受けているがその間にある要素の影響を取り除いた値である。

PACF partial auctcovariance function 偏自己共分散

PARCOR partial autocorreration coefficient 偏自己相関

解説

$$r_{Y_t Y_{t-k}} = \frac{COV(Y_t, Y_{t-k})}{S_{Y_t} S_{Y_{t-k}}}$$

自己相関係数が、時系列における時点間の関係の強さを示すものであるのに対し、偏自己相関係数は、与えられたラグの期間内の影響を取り除いた時の関係の強さを示す。モデルの次数を示すときに用いられる。

ここで示すのは、*Pearson autocorrelation coefficient PACF* である。

This mesure is more properly calleed the *Pearson correlation coefficient*

this correlation coefficient can be interpreted as the average standard deviation change in Y associated with a one standard deviation change in X .

X の標準偏差の変化率が Y の標準偏差の変化に及ぼす影響が見られる。 Y と X の影響の強さと方向 (+ -) をあらわす。

1 と -1 の間の値をとる。

*3

```
(acf a),. pacf a
      1      1
_0.223404  _0.25
_0.617021  _0.911322
 0.393617  0.802955
 0.212766      1
_0.265957      0
```

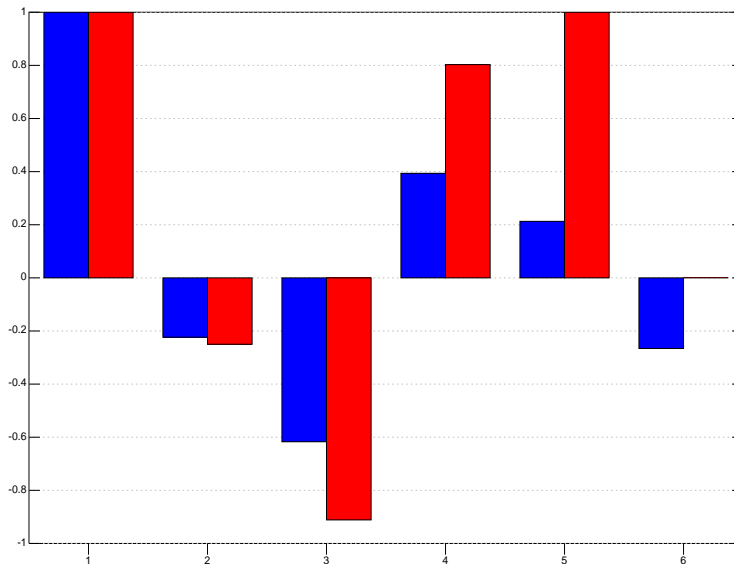


図 9.2 ACF=blue PACF=red

Script

```
pacf=: 3 : 0
```

NB. Pearsons autocorellations (many times at once)

```
C1=: (<\. y.) ,. |.@(<\)y.
```

NB. Cov(Yt Yt-k)

```
C3=: (C2=:> +/ L:0 */L:0 <"2 > dev L:0 C1) % N=:> {. "1 <:@# L:0 C1
```

NB. Syt Syt-k

```
C5=:*/ "1 > %: L:0 (C4=:+/ L:0 *: L:0 dev L:0 C1) % L:0 N2=:<:@# L:0 C1
```

*3 偏自己相関係数は,levinson のアルゴリズムによる計算過程で得られる.n levin acf data で出てくる各次数の回帰係数の最終の項が偏自己相関係数である。

C3 % C5
)

出典

Stephen A. DeLurgio [Forecasting Principles and Applications] McGRAW-HILL 1998

9.2 Autoregression 自己回帰

時系列モデルを用いると時系列の特徴を簡潔に表現できる。ここでは自己回帰モデル (autoregressive model, AR) を当てはめる問題を取り扱う。

最初にシンプルな自己回帰の方法を紹介したが、AR の解法は他に Yule-Walker 法、パーク法、ハウスホルダー法などがある。

ここでは、多変量自己回帰はレビンソンのアルゴリズムを用いたユール・ウォーカー法を用いるので、導出のため AR の同法を紹介している。(読み飛ばしてもよい。)

9.2.1 Yule-Walker 法

最もポピュラーな Yule-Walker 法は ACF から正規方程式を求める方法である。

Yull-Walker 方程式

$$C_M a_M = c_M$$

ACF

$$c(k) = \frac{1}{N} \sum_{s=1}^{N-k} x(s)x(s+k), (k = 0, 1, 2, \dots, M)$$

Yull-Walker 行列

$$C_M = \begin{bmatrix} c(0) & c(1) & \dots & c(M-2) & c(M-1) \\ c(1) & c(0) & c(1) & \dots & c(M-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c(M-2) & \dots & c(1) & c(0) & c(1) \\ c(M-1) & c(M-2) & \dots & c(1) & c(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ C_p \end{bmatrix}$$

$$c_M = (c(1), c(2), \dots, c(M))^T$$

自己相関 ACF(k) を用いて C_M を作れば、この方程式はクラメル法やガウス法で解くことができ Yull-Walker 法はこの yull-walker 方程式を直接解くことにより AR 係数の推定値を求められる。

メインフレームよりも高速で、多量のメモリーを積んだ最近の PC ではレビンソンのアルゴリズムを用いなくとも、APL や J の強力な逆行列機能で、直ちにこの方程

式の解を得ることができる.

経過と結果

```
acf DN90=: 3 6 8 4 4 8
```

```
1 _0.223404 _0.617021 0.393617 0.212766 _0.265957
```

Yull-walker 方程式 acf の結果を 3 次の C_M に、右に c_M を連結したもの
3 yw DN90 NB. YW 行列

1	_0.223404	_0.617021	_0.223404
_0.223404	1	_0.223404	_0.617021
_0.617021	_0.223404	1	0.393617

上の YW 行列をクラメル法で解いたもの。

```
cr=: %."1 NB. Cramer Method
```

```
3 yw DN90
```

```
1 _8.88178e_16 6.66134e_16 _0.37704
_2.22045e_16 1 4.44089e_16 _0.70024
_6.66134e_16 _8.32667e_16 1 0.00453847
```

$$y(t) = -0.37704x(t-1) - 0.70024x(t-2) + 0.00453847x(t-3)$$

Script

```
yw_sub0=:4 : 0
NB. Yull- Waker sub
NB. make C(m) Matrix
NB. for univariate AR
T1=: (|.}. M) , M=: y. NB. reverce and connect
T2=:> x. # < T1 NB. copy x. vertical
T3=: ({. $ T2), <: {: $ T2 NB. decrease $ to (0 _1)
T4=:(-x.) {. ("1) T3 $ , T2 NB. take backward (_x.) column
)
```



```

yw=:4 : 0
NB. Yule Walker Method(Univariate)
NB. Data type is Yoko list
NB.  $X(t)=X1(t-1)+ X2(t-2) + X3(t-3).....$ 
NB. Usage: x. jisuu // y. Data
Y1=: acf y.
Y3=:}. Y2=: (>: x.) {. Y1
Y4=: (x. yw_sub0 Y2),. Y3
Y5=: cr Y4 NB. cramer decomposition
)

NB. 上の部分解を求める
y4=: 4 : '(x. yw_sub0 Y2),. }. Y2=:(>: x.) {. acf y.'
```

Yull-Walker 法で当てはめたモデルは, 定常性をもつ。
時間を逆転させた, 過去に向かっての推計が可能なことも, 時系列の特色である。

9.2.2 レビンソンのアルゴリズム

レビンソンのアルゴリズムは最初に偏相関係数を求め, これをもとに回帰係数を計算する方法である。一変数の場合には, J は直接ユールウオーカー方程式を解くことができるが、ここでは, 多変量自己回帰係数を求めるアルゴリズムの理解のために, 作成した。

一変数

レビンソンのアルゴリズムとそのままスクリプトに落としたものを対比させてみる。このままでは次数毎に書き加える作業が必要である。

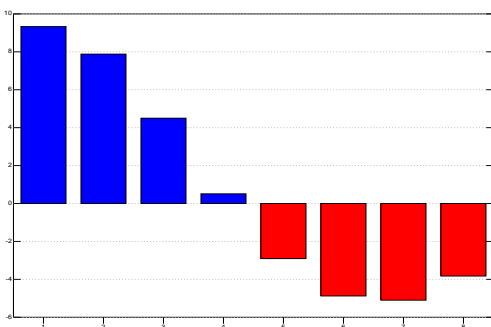


図 9.3 sample of ACF

Sample DATA

NB. Hirota & Ikoma 確率過程の数理 p.139

CX=: 9.333543 7.879308 4.500708 0.512286 _2.906882

CX=: CX, _4.876834 _5.100308 _3.826481

[0 次]

$$= V_0 = C_0$$

[1 次]

$$K_{1(1)} = \frac{C_1}{V_0}$$

$$V_1 = V_0(1 - K_{1(1)}^2)$$

[2 次]

$$K_{2(2)} = \frac{1}{V_1}(C_2 - K_1 C_1)$$

$$K_{2(1)} = K_{1(1)} - K_{2(2)}K_{1(1)}$$

$$V_2 = V_1(1 - (K_{2(2)})^2)$$

[3 次]

$$K_{3(3)} = \frac{1}{V_2} \left(C_3 - \sum \begin{pmatrix} K_{2(1)}C_2 \\ K_{2(2)}C_1 \end{pmatrix} \right)$$

$$K_{3(1)} = K_{2(1)} - K_{3(3)}K_{2(2)}$$

$$K_{3(2)} = K_{2(2)} - K_{3(3)}K_{2(1)}$$

$$V_3 = V_2(1 - (K_{3(3)})^2)$$

[4 次]

$$K_{4(4)} = \frac{1}{V_4} \left(C_4 - \sum \begin{pmatrix} K_{3(1)}C_3 \\ K_{3(2)}C_2 \\ K_{3(3)}C_1 \end{pmatrix} \right)$$

$$K_{4(1)} = K_{3(1)} - K_{4(4)}K_{3(3)}$$

$$K_{4(2)} = K_{3(2)} - K_{4(4)}K_{3(2)}$$

$$K_{4(3)} = K_{3(3)} - K_{4(4)}K_{3(1)}$$

$$V_4 = V_3(1 - (K_{4(4)})^2)$$

levin0=: 3 : 0

NB. Usage: u acf y

NB. -----0-----

V0=: { . y

ANS=: 0;V0

NB. -----1-----

K1=: (1{y})%V0

V1=: K0*1-K1^2

ANS=: ANS, : K1;V1

NB. -----2-----

K22=: (%V1)*(2{y})-K1*1{y}

K21=: K1-K22*K1

V2=: V1*1-K22^2

ANS=: ANS, (K21 ,K22);V2

NB. -----3-----

K33=: (%V2)*(3{y})-+/(K21, K22) * 2 1{y}

K32=: K22- */ K33,K21

K31=: K21-*/ K33,K22

V3=: V2*(1-K33^2)

ANS=: ANS, (K31 ,K32,K33);V3

NB. -----

K44=: (%V3)*(4{y})-+/(K31, K32,K33) *3 2 1{y}

K43=: K33- */ K44,K31

K42=: K32- */ K44,K32

K41=: K31- */ K44,K33

V4=: V3*(1-K44^2)

ANS=: ANS, (K41 ,K42,K43,K44);V4

)

4 levin DN91

自己回帰係数

分散

```

+-----+-----+
|0                |9.33354 |
+-----+-----+
|0.844193         |2.68189 |
+-----+-----+
|1.52126 _0.802026 |0.956775|
+-----+-----+
|1.50865 _0.778118 _0.0157164 |0.956538|
+-----+-----+
|1.50777 _0.821906 0.0691834 _0.0562752|0.953509|
+-----+-----+

```

Script

```

levin=: 4 : 0
NB. Usage: x levin acf y
NB. -----0-----
V0=: {. y
ANS=: 0;V0
NB. -----1-----
K1=: (%V0)*1{y
V1=: V0*1-K1^2
ANS=: ANS,: K1;V1
NB. -----
COUNTER=: 2
while. COUNTER < x do.
ANS=: ANS,ANS levin_sub }.(>:COUNTER) {. y
COUNTER=. >: COUNTER
end.
ANS
)

```

```

levin_sub=: 4 : 0
'KX V'=: {: x NB. ANS
Y0=: y NB. acf / (i. COUNTER){y
KK=: (%V)*({:Y0)-+/KX *}.|. Y0
KN=:|. (|. KX)- KK * KX
VX=: V * (1-KK^2)

```

```
(KN, KK); VX
)
```

右の欄は分散 σ^2 。回帰係数は, 上から 0 次, 1 次.. 5 次 (=x.) まで同時に求めている。AR の係数は, 次数に応じ、各行に表示される。右端の列は, 分散 σ^2 である。
(5 levin acf data のように acf を伴って実行する)

```
3 levin acf a
+-----+-----+
|0          |1          |
+-----+-----+
|_0.223404  |0.950091|
+-----+-----+
|_0.380226  _0.701965  |0.481928|
+-----+-----+
|_0.37704  _0.70024  0.00453847|0.481918|
+-----+-----+
3 yw a
          1 _8.88178e_16  6.66134e_16  _0.37704
_2.22045e_16          1 4.44089e_16  _0.70024
_6.66134e_16 _8.32667e_16          1 0.00453847
```

Yull-Walker 法で求めた結果と合致している。

9.2.3 AIC

AIC (A Information Criterion ,Akaike Information Criterion) は秩父セメントのキルン (焼成窯) の自動制御の導入過程で赤池弘次により考案された指標である。2003 年に AIC 誕生 30 周年の記念シンポジウムが横浜で催された。制御工学の要請で開発された理論であるが、幅広く高次方程式を使用する諸分野で使い込まれた堅牢な方式である。

AIC は高次の回帰係数のモデル選択に、簡潔に、有効な指標を示し、モデル選択の恣意性をなくし、適切な選択を行ってくれる。

AR の赤池情報量基準は次の式による。最適次数は AIC の最少値を選択すればよい。(マイナスでも, 最少値を選択すればよい。)

*4

$$AIC_m = -2l(\hat{\theta}) + 2(\text{パラメータ数})$$

*4 AIC は, 次数が上がると高次の桁の微小な数の調整を行っており, 最終的には計算限界で計算機が落ちる。従って, AIC の値と実際の数値を見ながら, 適当な次数を選択するとよい。

$$= N(\log 2\pi\hat{\sigma}_m^2 + 1) + N + 2(m + 1)$$

$$\log FPE(M) = \log \hat{\sigma}_M^2 + \log\left(1 + \frac{M}{N}\right) - \log\left(1 - \frac{M}{N}\right)$$

$$= \log \hat{\sigma}_M^2 + \frac{2M}{N}$$

M 次数

Working Example load 'user/classes/time/data/time_data.ijs'
東京の2月の最低気温のデータ DN12

```
3 exam_ar0 DN12
+-----+-----+-----+
|mean=6.904|corr=0.536786|AIC=41.5126|
+-----+-----+-----+
```

AIC を纏めて計算し最適次数を探る Script。AIC の推移を見ながら次数を選択する。11-12 次の AIC が適当なことを示している。(13 次は計算限界を超える)

```
15 loop_aic_ar0 DN12
2 41.6
3 41.5126
4 41.4362
5 41.7886
6 42.5812
7 41.5457
8 39.743
9 37.7295
10 33.0499
11 27.8573
12 27.4468
13 99999
14 99999
15 99999
```

```
11 linefit_ar0 DN12

exam_ar0=: 4 : 0
'MEAN R AIC'= . x exam_ar0_sub y
('mean=', (': (+/%#) y)); ('corr=', ": R); 'AIC=', ": AIC
)
```

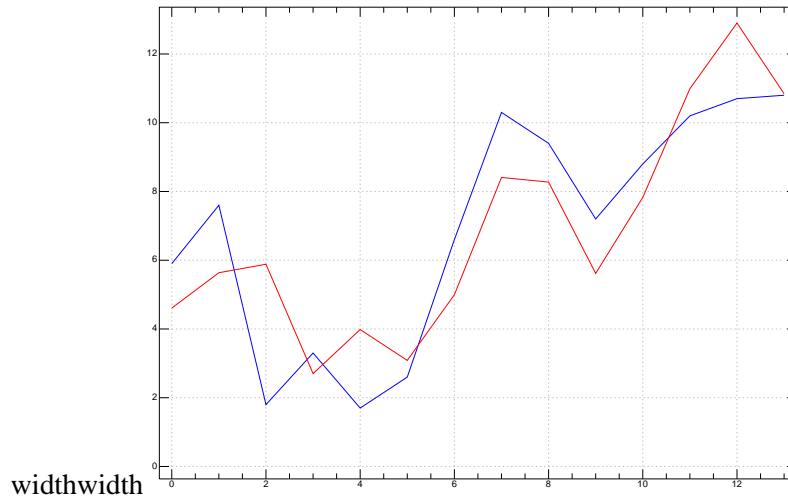


图 9.4 data=blue fitted=red

```

exam_ar0_sub=:4 : 0
NB. find AIC value
NB. Usage: x aictest y // x is jisuu // y is data
A0=.}:|.(“1) x>\Y0=.dev y      NB. drop last line
Q=. +/ *: (x}. Y0)- A0 +/ . * f=. x ar0 y NB. y- estim_ar y
R=. 1 - Q % +/ *: x}. Y0
AIC=(N * ^ . Q% N=(# y)-x )+ +: x
((+/%#) y); R;AIC
)

loop_aic_ar0=: 4 : 0
NB. 15 loop_aic_ar0 DS0
ANS=. <'
COUNTER=. 2
while. COUNTER <: x do.
try.
AIC=. {; COUNTER exam_ar0_sub y
catch. AIC=. 99999 end.
ANS=. ANS ,<COUNTER,AIC
COUNTER=. >: COUNTER
end.
;("1),. }.ANS
)

```

9.2.4 時系列の推計

時系列の最適な回帰係数を求めた後に、推計と予測を行う。

時間の流れに従い次のような組み合わせを作る。EXCELなどのデータは、新しいものを下に追加している。(逆の場合もあるが)

推計と予測は、例えば1から10の数で、次のような4次の組み合わせを作ると $x_{t-4}x_{t-3}x_{t-2}x_{t-1}$ になっている。

最下行(13)は,unknownであり,過去データからの1期先の予測に該当する。

```
|. ("0) 4>\i. 10

_1 _2 _3 _4
-----
3  2  1  0 | 4
4  3  2  1 | 5
5  4  3  2 | 6
6  5  4  3 | 7
7  6  5  4 | 8
8  7  6  5 | 9
-----|---
9  8  7  6 |(?)
```

1. 0は4から見て,4期前のデータであることを示している。
2. 指定した次数は原データの頭の方から落とす。(0から3)
3. 更に,最後の一個のデータは回帰に用いないので,落とす。(10)

10の自己回帰は(9 8 7 6)で行う。(10 9 8 7)| 11は予測に用いる。

推計は,左の組み合わせに,回帰係数 $A(m)$ を掛ける。この $i.10$ の数は回帰計算ができないので,先の気温の実データで自己回帰を行う。(次数は説明のため4次とする)

```
4 ar0 DN12
```

$$f = 0.973496_{t-1} - 0.548562_{t-2} + 0.232687_{t-3} + 0.0641936_{t-4}$$

```
estim_ar0=:4 : 0
```

```
NB. plot |: 3 estim_ar0 DS0
```

4 estim_ar0 DN12	1.8	6.09458
DN12 fitted	3.3	0.0203373
-----	1.7	5.6591
8.5 9.05466	2.6	1.94067
7.6 6.90825	6.6	3.10712
10.1 7.85527	10.3	6.77167
11.5 10.9773	9.4	8.65796
6.9 10.6693	7.2	7.02516
2 5.53323	8.8	6.80318
2.3 3.43283	10.2	10.0441
5.9 5.27571	10.7	9.98953
7.6 6.88093	10.8	9.85271

```

TMP0=:}:|. "1 x >\ y - MEAN=. +/%# y
(x }.y),.MEAN + TMP0 +/ . * (x }. y)%. TMP0
)

```

回帰係数から回帰の値を得る一般的な方法によっている。}: A1 は最終行を落とし
ており、この部分は、予測に用いる。

最初に、平均 0 のデータに加工して計算したので、平均を加えて、戻しておく。

9.2.5 時系列の予測

先に 1 から 10 までのデータを 4 次で折り畳んだ場合を示したが、ラインの下の行
(10987) に回帰係数をかけると一期先 (10) の予測が得られる。このダイアモン
ド片のような貴重な点を次々と原データの下に付け加えてけば、予測ができる。自己
回帰の予測には、自己回帰係数を固定した方式と、その都度回帰を行って、修正を加
える方式がある。通常は、固定でよい。ここでは、予測は 6 期までに固定したが、少
し変更を加えれば、予測の期数を増やすことはできる。

-1	-2	-3	-4	
3	2	1	0	4
4	3	2	1	5
5	4	3	2	6
6	5	4	3	7
7	6	5	4	8
8	7	6	5	9
9	8	7	6	(?10)

10	9	8	7	(?)
----	---	---	---	-----

```

    pred_ar0_sub=:4 : 0
f=. x ar0 y
MEAN + ({:|.(("1)x>\ y - MEAN=. (+/%#) y)+/ . * f
NB. prediction
)
pred_ar0=: 4 : '(x estim_ar0 y),0, x pred_ar0_sub y'

5 pred_ar0 DN12          1.7    5.77883
7.6    7.15113          2.6    2.07597
10.1   7.93671          6.6    2.49396
11.5   11.2182         10.3   7.07564
6.9    10.625           9.4    8.80091
2      5.32776           7.2    7.24792
2.3    3.26615           8.8    7.06044
5.9    5.14296          10.2   10.3148
7.6    6.52544          10.7   9.97611
1.8    5.82719          10.8   9.69078
3.3 _0.0696318
-----
0      9.63514 prediction

```

9.3 多変量時系列 VAR

多変量自己回帰モデルは、複数の変数がベクトルで表現されるので、VAR(vector autoregressive model) と呼ばれる。

9.3.1 相互相関関数 Cross-Correlation Coeffiencence

<i>ccf2</i>	相互相関係数 (多変数)	
<i>ccf3</i>	相互共分散 (多変数)	
<i>ccf</i>	相互相関	plot に便利 ' bar'plot ccf 0 1 {"1 dat 2 変数の相互共分散の部分

多変量の定常確率過程のデータを相互に時間を一コマずつづらして、相互の関係を見るのに、自己相関係数の多変量版とも言うべき、相互共分散と相互相関係数が有用である。

9.3.2 $CCF(k)$ crosscorrelation function

相互共分散関数は次により求められる。

$$CCF(k) = \frac{\sum_{t=1+k}^n (Y_t - \bar{Y})(X_{t-k} - \bar{X})}{\sqrt{\sum_{t=1}^n (Y_t - \bar{Y})^2 \sum_{t=1}^n (X_t - \bar{X})^2}}$$

または、

$$COV_{xy}(k) = \frac{1}{n} \sum_{t=1+k}^n (Y_t - \bar{Y})(X_{t-k} - \bar{X}), k \geq 0$$

$$COV_{xy}(k) = \frac{1}{n} \sum_{t=1-k}^n (Y_{t-k} - \bar{Y})(X_t - \bar{X}), k \geq 0$$

$$CCF(k) = \frac{COV_{xy}(k)}{S_x S_y}$$

$COV_{xy}(k) = \frac{1}{n} \sum_{t=1+k}^n (Y_t - \bar{Y})(X_{t-k} - \bar{X})$ $COV_{xy}(k) = \frac{1}{n} \sum_{t=1-k}^n (Y_{t-k} - \bar{Y})(X_t - \bar{X})$	<pre>ccf_t0=: :(L:0)@ .@(<\)@dev (<\.)@dev +/. * L:0 NB. 1/n は最終行で</pre>
$S_x S_y$	<pre>ccf_t1=: (*/)@:%:@((+/@: (^&2)@dev % #)</pre>
$CCF(k) = \frac{COV_{xy}(k)}{S_x S_y}$	<pre>ccf_t=: (ccf_t0 (%L:0) #) % L:0 ccf_t1</pre>

$$\rho_{xy}(k) = \gamma_{xy}(k) \sqrt{\gamma_{xx}(0)\gamma_{yy}(0)}$$

$$\gamma_{xy}(-k) = E[(X_t - \mu_x)(Y_{t-k} - \mu_y)]$$

9.3.3 相互相関

相互相関は幾つかの時系列のデータを相互にラグを取った場合の相関係数である。時間の推移とともに相関の強弱が変化する過程が plot によくあらわされる。時系列データは、最初に平均 0 のデータにしておく。dev dev=: -"1 (+/%#)。従って,ccf2 dev 0 1 2 {"1 dat のように dev や基準化 stand を適宜使い分ける。

Working Example

```
DN92=: 6 2 $ 26 4 16 6 18 3 35 7 21 2.5 29 3
```

```
Y X
-----
26 4
16 6
18 3
35 7
```

21 2.5

29 3

*5

X_{t+n} はデータをリバーズして, ccf2 を計算すればよい.

(|. ccf |."1 DN92),}. ccf DN92

$$COV_{xy}(k) = \frac{1}{n} \sum_{t=1+k}^n (Y_t - \bar{Y})(X_{t-k} - \bar{X})$$

次のようなデータ配列を作る。*6

|\.<\ DN92

```

+-----+-----+-----+-----+-----+-----+
|26  4|26  4|26  4|26  4|26  4|26  4| |
|16  6|16  6|16  6|16  6|16  6|   |
|18  3|18  3|18  3|18  3|   |   |
|35  7|35  7|35  7|   |   |   |
|21 2.5|21 2.5|   |   |   |   |
|29  3|   |   |   |   |   |   |
+-----+-----+-----+-----+-----+-----+

```

<\. DN92

```

+-----+-----+-----+-----+-----+-----+
|26  4|16  6|18  3|35  7|21 2.5|29  3| |
|16  6|18  3|35  7|21 2.5|29  3|   |
|18  3|35  7|21 2.5|29  3|   |   |
|35  7|21 2.5|29  3|   |   |   |
|21 2.5|29  3|   |   |   |   |   |
|29  3|   |   |   |   |   |   |
+-----+-----+-----+-----+-----+-----+

```

上の方をを transpose| : して、上下のマトリクスの内積をとると Cov $\sum_{t=1+k}^n (Y_t - \bar{Y})(X_{t-k} - \bar{X})$ が計算できる。

*7

*5 issue: DeLurgio Forecasting Principles and Applications P135

*6 (事前に-"1(+/%#) しておく)

*7 経過の説明で、dev は省略してある

```
(|: L:0 |. <\ DN92)+/ . * (L:0) <\. DN92
+-----+-----+-----+-----+-----+
| 3763  638.5| 2678 480.5|2421  340|1768 276|1010 113|754 78|
|638.5 125.25|496.5   88| 548 82.5| 353  52| 258  28|116 12|
+-----+-----+-----+-----+-----+
```

SD	C4%(L:0) # y. is $\frac{1}{\#n}$
$\sqrt{\sum_{t=1}^n (Y_t - \bar{Y})^2 \sum_{t=1}^n (X_t - \bar{X})^2}$	*/ %: (+/ ^&2 C2)% # n
C5 11.0149	

結果は次の Box であらわされる。

```
6j2 ": L:0 ccf2 DN92
+-----+-----+-----+-----+-----+
|  3.92  0.34| _1.23 _0.28| _0.42 _0.42|  0.24  0.41| _0.69  0.11|
|  0.34  0.26| _0.60 _0.13|  0.57  0.06| _0.22 _0.03|  0.14 _0.03|
+-----+-----+-----+-----+-----+
```

Variables	$Y_t X_{t+2}$	$Y_t X_{t+1}$	$Y_t X_t$	$T_t X_{t-1}$	$Y_t X_{t-2}$
CCF	_0.416103	_0.280554	0.336665	_0.597045	0.571196

Example IBN-NY-London

IBM 株の NY London 取引所のデータの最初の 10 data (1994-Jan-Dec) n=242

```
DN50=. ".@> readcsv 'classes/calculus/time/data/ibmnyln.csv'
```

Cross Correlations between NY & London IBM Stock Prices (NY Price)

$NewYork_t, London_{t-1}$

London NY

```

38.625  56.5
38.879 57.625
39.125   59
39.375  59.5
40.063  58.5
39.375 58.875
39.438  59.25
39.625 58.625
 39.25 58.125
39.063  58.75
    
```

from S.A.Delurgio Forecasting principles and Applications P549

相互相関係数の Box を利用しやすいように Matrix に展開する *seiretu* を作成した
 acf は最初の値は, 0 次であって必ず 0 となるが, ccf2 の 0 次は 1 とならない. (acf
 を用いた自己回帰では, 1 を取り除いている.)

```

          0    _1    _2    _3    _4    _5    _6
-----
-6: | (0.970) 0.981  0.959  0.941  0.926  0.910  0.897
-12: | 0.882  0.865  0.847  0.832  0.819  0.803  0.787
    
```

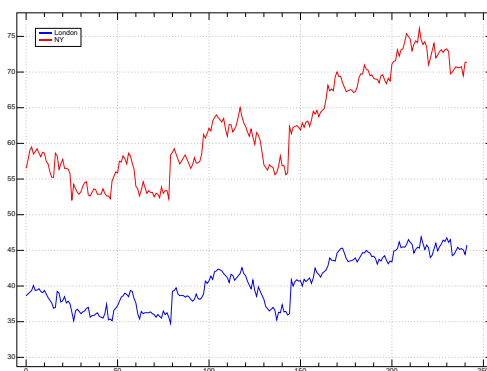


図 9.5 NY-LONDON Stock price(1994,242days)

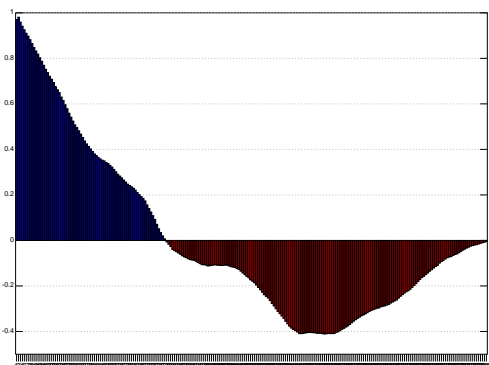


図 9.6 ccf of NY-LONDON Stock price

$CCF(-1)$ denote that the correlation NY and $London_{t+1}$ is 0.9808 and NY and $London_{t-1}$ is 0.951

ccf2 DN50

-----+-----+-----+-----+

```
| 0.50274 0.970074|0.487095 *0.950981|0.475845 0.935504|0.466009 0.92109| +
|0.970074 1.9891|*0.980794 1.95279|0.959201 1.92062|0.941186 1.89618| -
+-----+-----+-----+-----+
```

Stock Price of NY \& London(1994 01-12)

```
NY 自己相関 LN -k
LN +k LN 自己相関
```

Script

```
ccf2=: 3 : 0
NB. Correlation coefficient (multi variate )
NB. divide Sxx Syy
NB. Usage: ccf2 n (data matrix is tate type)
C2=: (dev=: - "1 +/ % #) y.
C4=: (|: L:0 |. <\ C2) +/ . * (L:0) <\. C2
C5=:*/ %: ( +/ ^&2 C2) % N=: # y. NB. +/ each SD
(C4 % L:0 N) % L:0 C5 NB. Cov / SD
)
```

9.3.4 多変量自己回帰・VAR

自己回帰 (Autoregression) を多変量に拡張した多変量自己回帰の方法にも、ユール・ウオーカー法、バーク法、ハウスホルダー法がある。

ユールウオーカー法・Yule-Walker Method

$$C(m) = \frac{1}{N} \sum_{N=m+1}^N x(n)x(n-m)$$

$C(m)$ は自己共分散関数の推定値で $k \times k$ 行列。

Yule-Walker 方程式は次のようになる。

$$\begin{pmatrix} C(0) & C(-1) & \dots & C(1-M) \\ C(-1) & C(0) & \dots & C(2-M) \\ \dots & \dots & \dots & \dots \\ C(1-M) & C(2-M) & \dots & C(0) \end{pmatrix} \begin{pmatrix} A(1) \\ A(2) \\ \dots \\ A(M) \end{pmatrix} = \begin{pmatrix} C(1) \\ C(2) \\ \dots \\ C(M) \end{pmatrix}$$

経過と説明

```
dat=: ?. 10 2 $ 20
```

相互共分散行列の生成

ccf3 dat

```

+-----+-----+-----+-----+-----+
|34.61  11|4.631 _9.57|_2.128 _3.11|_0.137 _1.53|_10.116 11.81|
|  11 27.2|_2.28 _15.5|  3.39  5.2| _4.74  _8.7|  4.64  8|
+-----+-----+-----+-----+-----+

```

Yule-Walker Matrix の生成

3 mgmain dat

```

+-----+-----+-----+-----+
|34.61  11 |4.631 _9.57|_2.128 _3.11|4.631 _9.57 |
|  11 27.2 |_2.28 _15.5|  3.39  5.2|_2.28 _15.5 |
+-----+-----+-----+-----+
|4.631 _9.57 |34.61  11 |4.631 _9.57 |_2.128 _3.11|
|_2.28 _15.5 |  11 27.2 |_2.28 _15.5 |  3.39  5.2|
+-----+-----+-----+-----+
|_2.128 _3.11|4.631 _9.57|34.61  11 |_0.137 _1.53|
|  3.39  5.2|_2.28 _15.5|  11 27.2 | _4.74  _8.7|
+-----+-----+-----+-----+

```

レビンソンのアルゴリズム・Levinson Algolism

Jはネステッドアレーをサポートしていないのでマトリックスを要素とするマトリックスの逆行列をいきなり求めることが出来ない。

多変数の Yule-Walker 方程式の解法として、レビンソンのアルゴリズム（多変数版）があり、一変数と同様に偏相関係数を求めることにより、高次の回帰係数を順次求めることができるので、この方法によった。

(1) 初期値の設定 ($m = 0$)

$$W_0 = Z_0 = C(0)$$

$$AIC(0) = N \log |W_0|$$

(2) レビンソンのアルゴリズム ($m = 1, 2, \dots, M$)

$$E(m) = C(m) - \sum_{i=1}^{m-1} A_{m-1}(i)C(m-i)$$

$$A_m(m) = E(m)Z_{m-1}^{-1}$$

$$B_m(m) = E(m)^T W_{m-1}^{-1}$$

$$A_m(i) = A_{m-1}(i) - A_m(m)B_{m-1}(m-i), (i = 1, \dots, m-1)$$

$$B_m(i) = B_{m-1}(i) - B_m(m)A_{m-1}(m-i), (i = 1, \dots, m-1)$$

$$W_m = C(0) - \sum_{i=1}^m A_m(i)C(i)^T$$

$$Z_m = C(0) - \sum_{i=1}^m B_m(i)C(i)$$

$$AIC(m) = N \log |W_m| + 2mk^2$$

経過と結果

[0 次]

$$W_0 = Z_0 = C(0)$$

[1 次]

$$E(1) = C(1)$$

$$A_1(1) = E(1)Z_0^{-1}$$

$$B_1(1) = E(1)^T W_0^{-1}$$

$$W_1 = C(0) - A_1(1)C(1)^T$$

$$V_1 = Z_1 = C(0) - B_1(1)C(1)$$

[2 次]

$$E(2) = C(2) - A_1(1)C(1)$$

$$A_2(2) = E(2)Z_1^{-1}$$

$$B_2(2) = E(2)^T W_1^{-1}$$

$$A_2(1) = A_1(1) - A_2(2)B_1(1)$$

$$B_2(1) = B_1(1) - B_2(2)A_1(1)$$

$$W_2 = C(0) - \sum \begin{pmatrix} A_2(1)C(1)^T \\ A_2(2)C(2)^T \end{pmatrix}$$

$$V_2 = Z_2 = C(0) - \sum \begin{pmatrix} B_2(1)C(1) \\ B_2(2)C(2) \end{pmatrix}$$

[3 次]

$$E(3) = C(3) - \sum \begin{pmatrix} A_2(1)C(2) \\ A_2(2)C(1) \end{pmatrix}$$

$$A_3(3) = E(3)Z_2^{-1}$$

$$B_3(3) = E(3)^T W_2^{-1}$$

$$\begin{pmatrix} A_3(1) = A_2(1) - A_3(3)B_2(2) \\ A_3(2) = A_2(2) - A_3(3)B_2(1) \end{pmatrix}$$

$$\begin{pmatrix} B_3(1) = B_2(1) - B_3(3)A_2(2) \\ B_3(2) = B_2(2) - B_3(3)A_2(1) \end{pmatrix}$$

$$W_3 = C(0) - \sum \begin{pmatrix} A_3(1)C(1)^T \\ A_3(2)C(2)^T \\ A_3(3)C(3)^T \end{pmatrix}$$

$$V_3 = Z_3 = C(0) - \sum \begin{pmatrix} B_3(1)C(1) \\ B_3(2)C(2) \\ B_3(3)C(3) \end{pmatrix}$$

[4次]

$$E(4) = C(4) - \sum \begin{pmatrix} A_3(1)C(3) \\ A_3(2)C(2) \\ A_3(3)C(1) \end{pmatrix}$$

$$A_4(4) = E(4)Z_3^{-1}$$

$$B_4(4) = E^T(4)^T W_3^{-1}$$

$$\begin{pmatrix} A_4(1) = A_3(1) - A_4(4)B_3(3) \\ A_4(2) = A_3(2) - A_4(4)B_3(2) \\ A_4(3) = A_3(3) - A_4(4)B_3(1) \end{pmatrix}$$

$$\begin{pmatrix} B_4(1) = B_3(1) - B_4(4)A_3(3) \\ B_4(2) = B_3(2) - B_4(4)A_3(2) \\ B_4(3) = B_3(3) - B_4(4)A_3(1) \end{pmatrix}$$

$$W_4 = C(0) - \sum \begin{pmatrix} A_4(1)C(1)^T \\ A_4(2)C(2)^T \\ A_4(3)C(3)^T \\ A_4(4)C(4)^T \end{pmatrix}$$

$$V_4 = Z_4 = C(0) - \sum \begin{pmatrix} B_4(1)C(1) \\ B_4(2)C(2) \\ B_4(3)C(3) \\ B_4(4)C(4) \end{pmatrix}$$

VAR 回帰

多変量自己回 帰係数を求め る	<i>varmain</i>	<i>x. varmain y.</i> x 次数 y data(Matrix)
-----------------------	----------------	--

9.3.5 自己回帰係数を求める

原データは、太平洋を航行中の船舶のエンジンの回転数と、横揺れ (pitching) を 2 秒毎 ($\Delta t = 2sec$) に計測したデータ。ここでは、原データの図と相互相関係数が提示されている。

d1 NB. 相互相関係数

```

+-----+-----+-----+-----+
|9.22 4.38|6.89 1.48|2.58 _1.66|_1.68 _3.48|
|4.38 8.99|5.32 6.75|4.38 2.53| 2.52 _0.87|
+-----+-----+-----+-----+
    
```

(出典 北川 多変量自己回帰モデルの推定 P.111)

多変量時系列 $X(n)$ は平均値 0 で、定常性を仮定している。

$$R(m) = E[X(n)X(n-m)^T] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N X(n)X(n-1)^T$$

相互共分散行列 $R(m)$ は $C(m)$ により推定できる。

$$C(m) = \frac{1}{N} \sum_{n=m+1}^N X(n)X(n-m)^T$$

M 個の行列 $\hat{A}(1), \dots, \hat{A}(M)$ に関する次の連立方程式を解くと、自己回帰係数の推定値 $\hat{A}(1), \dots, \hat{A}(M)$ が求められる。

更に、白色雑音の分散共分散行列の推定値 $\hat{V}_m = C(0) - \sum_{j=1}^M \hat{A}(j)C(-j)$ が得られる。

$$C(m) = \sum_{j=1}^M A(j)C(m-j)^T, (m = 1, \dots, M)$$

多変量自己回帰モデルの解法として、レビンソンのアルゴリズムを用いる。多変量のレビンソンのアルゴリズムは、前向きが多変量自己回帰モデル $A(M)$ と、後ろ向きの多変量自己回帰モデル $B(M)$ を計算しなければならない。

`varmain` の係数行列の output の表示は、次のようになっている。

結果は B_m と V を用いる。

[1 次]

$A_1(1) \quad B_1(1)$

$W_1 \quad V_1$

1 varmain d1

```

+-----+-----+-----+-----+
|0.870577 _0.259525| 0.606553 0.296251|
|0.286669 0.611167|_0.255242 0.87519|
+-----+-----+-----+-----+
|3.60582 1.50032 |3.46479 1.48261 |
|1.50032 3.33954 |1.48261 3.46022 |
+-----+-----+-----+-----+
    
```

[2 次]

$A_2(1)$ $A_2(2)$
 $B_2(1)$ $B_2(2)$
 W_2 V_2

2 varmain d1

```

+-----+-----+
| 1.16812 0.000710479| _0.53891 _0.114926 |
|0.132933      1.12252|0.00664515 _0.586522 |
+-----+-----+
|      1.09854 0.149265| _0.565279 0.000462227|
|_0.00867256      1.1921|_0.0986828      _0.560433|
+-----+-----+
| 2.37021 0.811994 | 2.31337 0.807092 |
|0.811994      2.1606 |0.807092      2.21026 |
+-----+-----+

```

[3次]

$A_3(1)$ $A_3(2)$ $A_3(3)$
 $B_3(1)$ $B_3(2)$ $B_3(3)$
 W_3 V_3

6j3":(L:0) 3 varmain d1

```

+-----+-----+-----+
| 1.069 0.047|_0.329_0.186|_0.190 0.083|
| 0.197 1.222|_0.083_0.811| 0.083 0.178|
+-----+-----+-----+
| 0.967 0.266|_0.316_0.275|_0.241 0.246|
|_0.047 1.324|_0.050_0.828|_0.069 0.239|
+-----+-----+-----+
| 2.297 0.837| 2.141 0.701|          |
| 0.837 2.051| 0.701 2.103|          |
+-----+-----+-----+

```

9.3.6 AIC

多変量自己回帰モデルの当てはめも、情報量規準 AIC により、最適次数を決定する。AIC の小さい方のモデルを選ばばよい。

$AIC = -2(\text{最大対数尤度}) + 2(\text{パラメータ数})$

$$L = -\frac{N}{2}(k \log 2\pi + \log |\hat{V}_m| + k)$$

次数に無関係な定数を無視すると最終的には次のようになる

$$AIC^*(m) = N \log |\hat{V}_m| + 2mk^2$$

V_m : W_i

m : 次数

k : 変数の数

AIC の例 7次の方が AIC の値は小さい。

```

7 aic_v DN50
_1707.82
6 aic_v DN50
_1712.79
5 aic_v DN50
_1719.21
4 aic_v DN50
_1719.33
3 aic_v DN50
_1717.16

```

Script

```

aic_v=: 4 : 0
NB. Usage: // x. jisuu //y. is data (tate Multi variate)
NB. Usage x. aic_v dev2 y. //
2 is already combined
NB. AV1=: ccf2 y.
x. varmain ccf2 y.
( 2*x.* (1{$ y.)^2) + (# y.) * ^ . tmp1=: -/ . * tmp0=: >{. {: ANS
)

```

9.3.7 TIMSAC のデータで

統計数理研究所の時系列解析プログラム TIMSAC が公表されている。TIMSAC の多変量時系列解析のサンプルデータを用いて見よう。(n=500)

*8

```
TDAT=: ".@> readcsv jpath '~user/classes/calculus/time/data/timsac.csv'
```

```
TDAT
```

```
713 5508 652
709 5511 650
709 5509 686
710 5506 697
712 5503 701
714 5502 706
716 5500 709
716 5499 710
718 5499 710
720 5496 708
721 5491 713
724 5490 714
726 5485 714
728 5484 735
730 5482 736
(n= 500)
```

```
自己相関係数
```

```
8j5 ": L:0 ccf2 TDAT
```

```
+-----+-----+-----+
| 0.01343 0.00407 0.01425| 0.01325 0.00395 0.01387| 0.01284 0.00384 0.01342|
| 0.00407 0.02163 0.00098| 0.00422 0.02145 0.00057| 0.00438 0.02120 0.00112|
| 0.01425 0.00098 0.10175| 0.01449 0.00161 0.06427| 0.01432 0.00189 0.00243|
+-----+-----+-----+
```

```
4 varmain ccf2 TDAT
```

```
8j4 ": (L:0) 4 varmain ccf2 TDAT
```

```
+-----+-----+-----+
| 1.6684 _0.0235 0.0044| _0.7385 0.0199 _0.0039| 0.0823 0.0206 0.0025|
+-----+-----+-----+
```

*8 TIMSAC のサンプルデータは FORTRAN のベタフォーマットでパラメータも書き込むようになっている。J の様式に変換した。

```

| _0.0481  1.1108  0.0112|  0.0607  0.0091  0.0029| _0.0333  0.0226 _0.0089|
|  1.3395 _0.8626  0.9041| _0.7803  0.1268 _0.5066| _0.0823 _0.0068 _0.2187|
+-----+-----+-----+
|  1.6821 _0.0007  0.0024| _0.7512  0.0120 _0.0006|  0.0770 _0.0096  0.0014|
| _0.0265  1.0900 _0.0064|  0.0161  0.0184  0.0094|  0.0361  0.0316  0.0026|
|  0.5731 _0.1970  0.9113|  0.0174  0.1294 _0.5150|  0.8239 _0.5512 _0.2251|
+-----+-----+-----+
|  0.0002  0.0000  0.0001|  0.0002  0.0000  0.0000|
|  0.0000  0.0003  0.0002|  0.0000  0.0003  0.0000|
|  0.0001  0.0002  0.0291|  0.0000  0.0000  0.0294|
+-----+-----+-----+

```

上段 Amm 3次まで掲載
 中断 Bmm 3次まで掲載
 下段 W V

9.3.8 推計と予測

<i>varmain</i>	多変量自己回帰係数を 求める	
<i>var_estim</i>	多変量自己回帰 多変量自己回帰	非ループ型 ループ型
<i>plot_var</i>	plot	

9.3.9 推計

前向き of m 次多変量自己回帰モデル

$$X(n) = \sum_{i=1}^m A_m(i)X(n-i) + U(n), \quad E[U(n)U(n)^T] = W_m$$

後向き of m 次多変量自己回帰モデル

$$X(n) = \sum_{i=1}^m B_m(i)X(n-i) + V(n), \quad E[V(n)V(n)^T] = Z_m$$

ここでは、AR と同じように次のような組み合わせで前向きに予測を行う。 $t_{-3} t_{-2} t_{-1} t_0$ となる。多変量回帰用のデータテーブルを作成する。(実際は *dev2* で平均を 0 にし、計算の後、最後に平均を足して戻す。)

名詞の名を *T0* とし、一度に生成する補助関数 *make_T0* を作成した。予測に用いる行は *make_T0* の方で落としてある。

<pre> a 0 10 1 11 2 12 3 13 4 14 5 15 6 16 7 17 8 18 9 19 </pre>	<pre> make_T0 で指定次数の3次元での次の ような組み合わせのテーブルを作る。 3 make_T0 a +-----+-----+-----+ 0 10 1 11 2 12 +-----+-----+-----+ 1 11 2 12 3 13 +-----+-----+-----+ 2 12 3 13 4 14 +-----+-----+-----+ 3 13 4 14 5 15 +-----+-----+-----+ 4 14 5 15 6 16 +-----+-----+-----+ 5 15 6 16 7 17 +-----+-----+-----+ 6 16 7 17 8 18 +-----+-----+-----+ 7 17 8 18 9 19 +-----+-----+-----+ </pre>
--	---

予測や回帰の場合には、最新のデータを `lotate(.)` で `top` に持ってきた方が、理解もしやすいし、 t_{-1} などを作る場合の間違いも少ない。回帰係数は、`lotate` しても同じであり、最後に、グラフィックに渡す段階で、元に戻してやればよい。時系列の場合は、`lotate` した最上行は、予測に用いるので、回帰の段階では落としておく。

この段階では、 t_{-3}, t_{-2}, t_{-1} になっており、回帰係数とは逆なので、どちらかを `lotate` させる。ここでは、回帰係数の方を回してある。

```

}. |.3 make_T0 a
+-----+-----+-----+
|6 16|7 17|8 18|
+-----+-----+-----+
|5 15|6 16|7 17|
+-----+-----+-----+

```



```
| 4 14|5 15|6 16|
+----+----+----+
| 3 13|4 14|5 15|
+----+----+----+
| 2 12|3 13|4 14|
+----+----+----+
| 1 11|2 12|3 13|
+----+----+----+
| 0 10|1 11|2 12|
+----+----+----+
```

ここの Box は 3 次元である。この場合は、*vector* と *Matrix* の計算は次の表のように、(メモリーの消費を恐れずに) *Vector* を #*n* 個 Copy して *Matrix* と *Matrix* にしてしまうか、Loop により *Vector* と *Vector* にして計算するかのどちらかである。*var_pred* は *Matrix* 計算型である。

a2=. 1 2;3 4;5 6 a2 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ >7# <a2 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+ 1 2 3 4 5 6 +---+---+---+	}. .3 make_T0 a +-----+-----+-----+ 6 16 7 17 8 18 +-----+-----+-----+ 5 15 6 16 7 17 +-----+-----+-----+ 4 14 5 15 6 16 +-----+-----+-----+ 3 13 4 14 5 15 +-----+-----+-----+ 2 12 3 13 4 14 +-----+-----+-----+ 1 11 2 12 3 13 +-----+-----+-----+ 0 10 1 11 2 12 +-----+-----+-----+	(> 7#<a2) +/ . * (L:0) }. . 3 make_T0 a +-----+-----+-----+ 38 89 148 +-----+-----+-----+ 35 82 137 +-----+-----+-----+ 32 75 126 +-----+-----+-----+ 29 68 115 +-----+-----+-----+ 26 61 104 +-----+-----+-----+ 23 54 93 +-----+-----+-----+ 20 47 82 +-----+-----+-----+
---	--	--

IBM Stock Price NY-LONDON の推計の結果 (最後の10個)

```

3 varmain ccf2 DN50
+-----+-----+-----+
|0.799279 0.0413291 |0.00752259 0.047732 |_0.0450348 0.0187558|
| 1.26671 0.481737 | _0.554715 0.262278 | _0.465082 0.126917|
+-----+-----+-----+
| 0.350795 0.520549| 0.301107 _0.216706 | 0.0622926 _0.164721|
|_0.0472608 0.93022|_0.104913 _0.0130778 |_0.0436674 0.163747|
+-----+-----+-----+
|0.0295905 0.0133707| 0.0119072 0.00944001| |
|0.0133707 0.0314022|0.00944001 0.0705072| |
+-----+-----+-----+

```

Amm

Bmm

W, V(Z)

*9

```
_10{. 3 var_estim DN50
45.9694 72.6058
46.094 71.5998
44.1805 68.0859
44.266 69.2492
44.8073 71.1367
45.2212 71.4892
44.9532 70.5977
45.0312 70.7419
44.851 70.0119
44.3646 69.8571
```

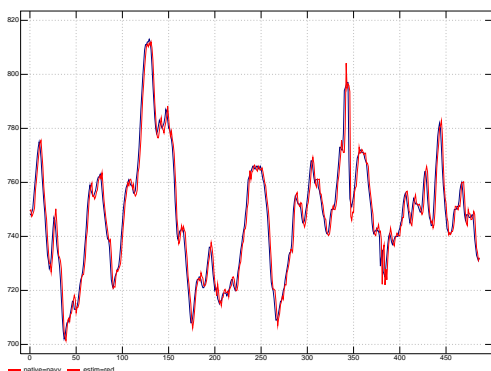


図 9.7 変数 1, 一実数と回帰

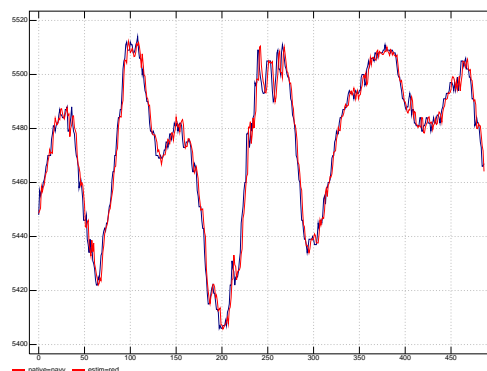


図 9.8 変数 2 一実数と回帰

Script

```
make_T0=:4 : ' > x.<\,({: $ y.)<\"1 y.'NB.not drop last for forecast
```

```
var_estim=: 4 : 0
```

```
NB. reg var //reg_v333
```

*9 表示を短くするため 3 次とし、AIC は用いていない。

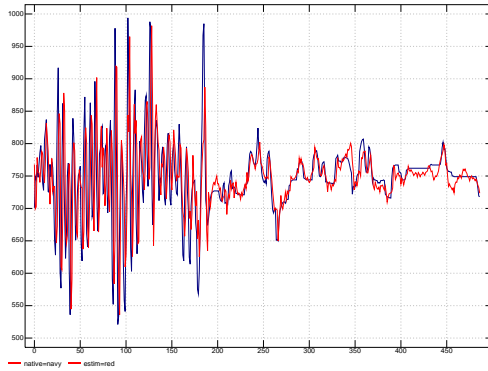


図 9.9 変数 3 - 実数と回帰

```

NB. Usage: x. reg_v2 y.
NB. X. jisuu y. data( type tate)
Y0=: dev2 y.
MEAN=: (mean=. +/ % #) y. NB. mean of each y.
T0=: }. |. x. make_T0 Y0 NB. drop head for pred
AV3=: {. x. varmain ccf2 y. NB. take Amm
AV4=: (|.("1) >(# T0)#<AV3) +/ . * (L:0) T0 NB. |.("1) Amm is t_x. ..t_x.-1
AV5=: |. >MEAN + L:0 +/ L:0 <"2 > AV4 NB. lotate up/dn
)
    
```

*10

9.3.10 予測

$$\hat{X}(n+1) = \sum_{i=1}^m A(i)X(n+1-i)$$

$\hat{X}(n+1)$ の予測誤差の分散共分散行列 W_m

pred	<i>pred_var</i> <i>pred_var6</i>	一期先予測 6期先まで予測 (係数非更新)
plot	<i>plot_pred6</i>	6期先まで plot x. 次数 データのカラム e.g. 12 0 e.g. 12 0 <i>plot_pred6 ldat</i>

*10 VAR は t_0 の関係で 1 拍微妙にずれる。取りあえず DAT で調整してある

data	予測に用いる部分
a2	
6 37	TP NB. 最終行
22 26	(dev2 していない。)
10 2	
33 33	TP
46 19	+-----+-----+-----+
25 41	26 33 0 19 3 20
1 2	+-----+-----+-----+
26 33	
0 19	
3 20	

最終行に回帰係数を掛け、Box を開いて、項毎に足し合わせ、平均を加えれば、予測値となる。pred_var6 は回帰係数を毎回更新する型である。

```
_7{. (9 ;4) var_pred TDAT
      732    5466    731
      731    5466    719
      732    5467    718
733.079 5465.27 731.151
734.715 5463.99 743.254
      736.52 5462.67    746.6
738.347 5461.3 739.223
```

Script

```
var_pred=: 4 : 0
NB. reg var and write ANS to AV00 OK!! Fine
NB. Usage: x. pred_v3n y. //prediction one step only //use sub
NB. X. jisuu y. data( type vertical/tate)
MEAN=: (mean=. +/%) y.
Y0=: dev2 y. NB. not drop last line (use pred)
```

```

AV3=: {. x. varmain ccf2 y.
T0=: x. make_T0 Y0                      NB. Xn-1 matrix
AV5=: +/> AV3 +/ . * L:0 |. {: T0       NB. reverse T0 order to time
ANS_VP=:MEAN + AV5
)

```

9.4 インパルス応答

インパルス応答は時間領域における伝達特性の解析である。^{*11}インパルスとはシステムに突然加えられた衝撃であり、それがどのように相互相関係数を介して伝わるかを分析する。

$$Z(s) = \Phi Z(s-1) + V(s)$$

状態方程式を AR 係数行列を用いて作成する。

$$\Phi = \begin{bmatrix} A(1) & A(2) & A(3) & \cdots & A(M-1) & A(M) \\ I & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 \end{bmatrix}$$

9.4.1 2変数の状態方程式の例と計算過程

状態方程式

$$Z(s) = \Phi Z(s-1) + V(s)$$

$$\Phi = \begin{bmatrix} A(1) & A(2) \\ I & 0 \end{bmatrix} = \begin{bmatrix} a_{11}(1) & a_{12}(1) & a_{11}(2) & a_{12}(2) \\ a_{21}(1) & a_{22}(1) & a_{21}(2) & a_{22}(2) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$V(s)$ はノイズ項からなる行列

$$V(s) = \begin{bmatrix} E(s) \\ 0 \end{bmatrix} = \begin{bmatrix} e_1(s) \\ 2_2(s) \\ 0 \\ 0 \end{bmatrix}$$

$$Z(0) = \begin{bmatrix} x(0) \\ x(-1) \end{bmatrix} = \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_1(-1) \\ x_2(-1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

*11 周波数領域では周波数伝達関数が受け持つ。

$$\begin{aligned}
 Z(1) &= \Phi Z(0) + V(1) = \begin{bmatrix} A(1) & A(2) \\ I & 0 \end{bmatrix} Z(0) + \begin{bmatrix} E(1) \\ 0 \end{bmatrix} \\
 &= \begin{bmatrix} a_{11}(1) & a_{12}(1) & a_{11}(2) & a_{12}(2) \\ a_{21}(1) & a_{22}(1) & a_{21}(2) & a_{22}(2) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 Z(2) &= \begin{bmatrix} a_{11}(1) & a_{12}(1) & a_{11}(2) & a_{12}(2) \\ a_{21}(1) & a_{22}(1) & a_{21}(2) & a_{22}(2) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11}(1) \\ a_{12}(1) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} z_{11}(2) \\ z_{21}(2) \\ z_{12}(2) \\ z_{22}(2) \end{bmatrix} \\
 Z(3) &= \begin{bmatrix} a_{11}(1) & a_{12}(1) & a_{11}(2) & a_{12}(2) \\ a_{21}(1) & a_{22}(1) & a_{21}(2) & a_{22}(2) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11}(1) \\ a_{12}(1) \\ 0 \\ 0 \end{bmatrix}
 \end{aligned}$$

この部分を反復して内積演算を行えばよい。回数は(Φの項数で止めているが)20,30回と指定してもよい。

$$\begin{bmatrix} a_{11}(1) \\ a_{12}(1) \\ 0 \\ 0 \end{bmatrix}$$

経過と説明 状態方程式

```
impulse_mat_sub 2 varmain ccf2 WDAT
```

```

_0.042  0.492  _0.082  0.06  0.338  0.128
 0.428  0.028  _0.059  0.288  0.008  0.116
 0.518  0.050  0.103  0.240  0.066  _0.004
      1    0    0    0    0    0
      0    1    0    0    0    0
      0    0    1    0    0    0
    
```

インパルスの矢

```

, .ARROW
+-----+
| 1 0 0 0 0 0 | A
+-----+
| 0 1 0 0 0 0 | B
    
```

```

+-----+
|0 0 1 0 0 0| C
+-----+

      impulse 2 varmain ccf2 WDAT
+-----+
|          1          0          0          |
|_0.0417323 0.428024 0.518282          |
| 0.230277 0.252217 0.293588          |
| 0.299799 0.133131 0.178242          |
| 0.175606 0.220172 0.251121          | A
| 0.166637 0.172743 0.207996          |
| 0.178595 0.142043 0.172113          |
+-----+
|          0          1          0          |
|_0.492332 0.0286744 0.050011          |
| 0.32835 0.200217 0.328189          |
|_0.103808 0.274437 0.334063          | B
|_0.233325 0.163805 0.19267          |
|_0.197415 0.159746 0.190602          |
|_0.149186 0.1662 0.196016          |
+-----+
|          0          0          1          |
|_0.0820309 _0.0599104 0.10349          |
| 0.0942977 0.0737952 _0.0396384          |
| 0.0237331 0.0337471 0.02428          | C
| 0.0392207 0.0316638 0.0442415          |
| 0.0263197 0.0244553 0.0343134          |
| 0.026924 0.0261446 0.0297238          |
+-----+

```

サンプルデータのインパルス応答（閉鎖系）の図

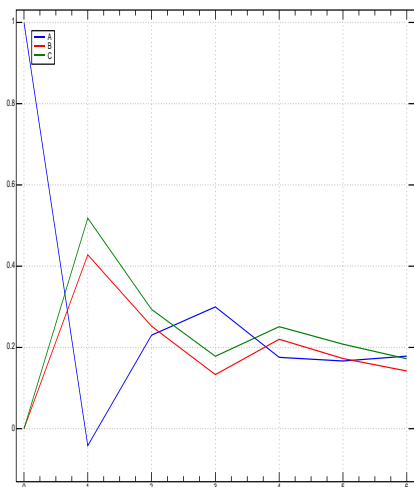


図 9.10 Impulse to A)

A へのインパルス入力少し遅れて B に伝わり、ついで C に伝わる。

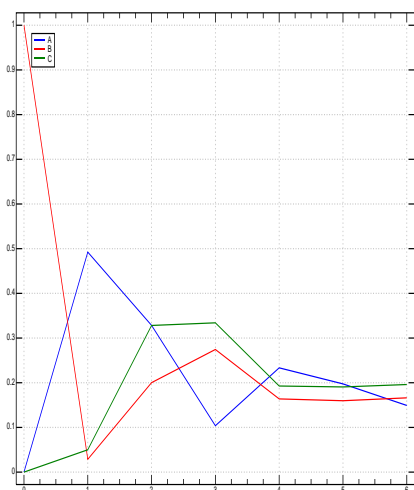


図 9.11 Impulse to B

B にインパルス入力すると A に伝わり、少し遅れて C に伝わる。

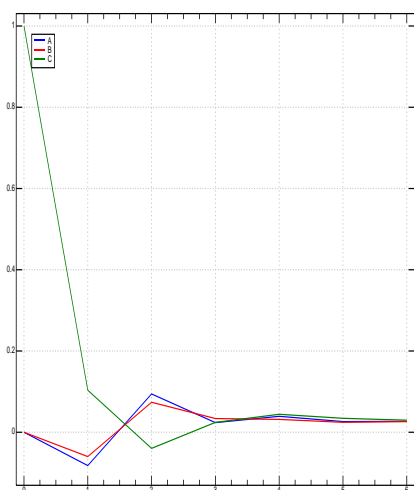


図 9.12 Impulse to C

C へのインパルス入力は A,B に伝わらない。A と B とは相互に制御しあっているが C によっては制御されていない

9.5 関数一覧

自己相関係数	<i>acf</i>	acf y
偏自己相関係数	<i>pacf</i>	pacf y
単純法	<i>ar0</i>	m ar0 y
Yull-walker 法	<i>yw</i> <i>exam_ar0</i> <i>linefit_ar0</i>	m yw y m exam_ar0 y m linefit_ar0 y
レビンソン	<i>levin</i>	m levin y
相互共分散	<i>ccf2</i>	ccf2 y
多変量自己回帰	<i>varmain</i>	m varmain ccf2 y
多変量自己回帰の AIC	<i>aic_v</i>	m aic_v y
多変量の推計	<i>var_estim</i>	m var_estim y
多変量の予測	<i>var_pred</i>	m var_pred y
インパルス応答	<i>impulse</i>	impulse m varmain ccf2 y

9.5.1 参考文献

Stephen A. DeLurgio [Forecasting Principles and Applications] McGRAW-HILL 1998

赤池弘次・中川東一郎 「ダイナミックシステムの統計的解析と制御(新版)」サイエンス社 1972/2000

石黒真木夫「予測と AR モデル」(尾崎・北川編 時系列解析の方法 朝倉書店) 1998

北川源四郎 多変量時系列モデル [時系列解析の方法] 朝倉書店 1998

廣田薫・生駒哲一「確率過程の数理(数理工学基礎シリーズ) 朝倉書店 2001

和田孝雄「生体のゆらぎとリズム」講談社サイエンスブック 1997

第10章

同時方程式の行列による解法

行列を用いた線形同時方程式モデルの作成と計算を行うなう。

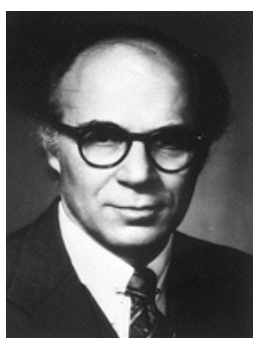


図 10.1 L.Klein(1920-)

1920年ネブラスカ州オマハに生まれる。UC Berkley で経済学と数学を学び、MIT でサミュエルソンの下で博士号を得る。シカゴ大学におかれていたコールズ委員会で計量モデルを作成。マッカーシー旋風を避けてヨーロッパに渡って Oxford でイギリスの経済モデルの作成に従事。1958年に帰国しペンシルバニア大学教授に就任。1980年にノーベル記念経済学賞を受賞

10.1 線形同時方程式

同時方程式は多入力、多出力の連立方程式システムが特色である。1950年にコールズ委員会でのクライン作成のモデル1が計量経済学の船出となったと言われる。

10.1.1 小さなアナログモデルから始めよう

Working Example リンゴのモデル

リンゴの数量、市場価格、購買者の所得、日照時間から需要関数、供給関数を求める。些か無謀なモデルであるが、モデルを求める手順の好例であるので検討してみよう。

構造型方程式	$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$ $Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$	
「内生変数」	$Q_t =$ リンゴの数量 $P_t =$ リンゴの市場価格	連立方程式の体系の中で相互依存により決定される変数
「外生変数」	$Y_t =$ 購買者の所得 $T_t =$ 日照時間	この体系外で与えられる値

t 年	Q_0 数量 千個	P_0 市場価格 円	Y_0 需要者の所得 万円	T_0 一日平均日照時間 時間
0	57	78	28	7
1	55	96	29	4.1
2	66	87	32	7.2
3	65	98	33	5.4
4	71	104	35	5.8
5	74	105	36	6.7
6	71	110	36	5
7	71	113	38	6.3

データの出典:白砂堤津耶「例題で学ぶ・初歩からの計量経済学」1998/2007 日本評論社

構造型方程式から誘導方程式を導く

内生変数を含む式は、Haavelmo^{*1}バイアスと呼ばれる歪みを持ち、不偏性と一致性を持たない。そこで、外生変数と先決変数とで構成される誘導方程式を導き、次いで、構造型方程式に戻すと、一致性は確保できる。誘導型を求めるため例題の数式を展開すると次のようになる。

$$\begin{aligned} \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t} &= \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t} \\ \alpha_1 P_t - \beta_1 P_t &= -\alpha_0 - \alpha_2 Y_t + u_{1t} + \beta_0 + \beta_2 T_t + u_{2t} \\ P_t(\alpha_1 - \beta_1) &= -\alpha_0 + \beta_0 - \alpha_2 Y_t + \beta_2 T_t - u_{1t} + u_{2t} \\ P_t &= \frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 Y_t}{\alpha_1 - \beta_1} + \frac{\beta_2 T_t}{\alpha_1 - \beta_1} + \frac{-u_{1t} + u_{2t}}{\alpha_1 - \beta_1} \end{aligned}$$

$$\begin{aligned} Q_t &= \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t} \\ &= \alpha_0 + \alpha_1 \left(\frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 Y_t}{\alpha_1 - \beta_1} + \frac{\beta_2 T_t}{\alpha_1 - \beta_1} + \frac{-u_{1t} + u_{2t}}{\alpha_1 - \beta_1} \right) + \alpha_2 Y_t + u_{1t} \\ &= \alpha_0 + \frac{\alpha_1(-\alpha_0 + \beta_0)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-\alpha_2 Y_t)}{\alpha_1 - \beta_1} + \frac{\alpha_1(\beta_2 T_t)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-u_{1t} + u_{2t})}{\alpha_1 - \beta_1} + \alpha_2 Y_t + u_{1t} \\ &= \frac{\alpha_0(\alpha_1 - \beta_1) + \alpha_1(-\alpha_0 + \beta_0)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-\alpha_2 Y_t) + \alpha_2 Y_t(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1} + \frac{\alpha_1(\beta_2 T_t)}{\alpha_1 - \beta_1} + \\ &\quad \frac{(-\alpha_1 u_{1t} + \alpha_1 u_{2t}) + (\alpha_1 u_{1t} - \beta_1 u_{1t})}{\alpha_1 - \beta_1} \\ &= \frac{\alpha_1 \beta_0 - \alpha_0 \beta_1}{\alpha_1 - \beta_1} + \frac{-\alpha_2 \beta_1 Y_t}{\alpha_1 - \beta_1} + \frac{\alpha_1 \beta_2 T_t}{\alpha_1 - \beta_1} + \frac{\alpha_1 u_{2t} - \beta_1 u_{1t}}{\alpha_1 - \beta_1} \end{aligned}$$

誘導型パラメータを次のように置くと下表右欄の誘導方程式を導くことができる

^{*1} Trigve M. Haavelmo 1911-1999 Norway 1989 ノーベル記念賞

$$\begin{aligned} \pi_{10} &= \frac{\alpha_1\beta_0 - \alpha_0\beta_1}{\alpha_1 - \beta_1} & \pi_{11} &= \frac{-\alpha_2\beta_1}{\alpha_1 - \beta_1} & \pi_{12} &= \frac{\alpha_1\beta_2}{\alpha_1 - \beta_1} \\ \pi_{20} &= \frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} & \pi_{21} &= \frac{-\alpha_2}{\alpha_1 - \beta_1} & \pi_{22} &= \frac{\beta_2}{\alpha_1 - \beta_1} \\ u_{1t} &= \frac{-u_{t1} + u_{2t}}{\alpha_1 - \beta_1} & u_{2t} &= \frac{\alpha_1 u_{2t} - \beta_1 u_{1t}}{\alpha_1 - \beta_1} \end{aligned}$$

構造型方程式	誘導方程式
$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$	$Q_t = \pi_{10} + \pi_{11} Y_t + \pi_{12} T_t + u_{1t}$
$Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$	$P_t = \pi_{20} + \pi_{21} Y_t + \pi_{22} T_t + u_{2t}$

内生変数と外生変数の区分は微妙であって判然とした区分は存在しない。モデルの中の変数が「自立的に、そしてモデル内の他の変数とは独立に変動すると期待できるかどうか」*2を一応の目安として、行列表現で右の項に記述する外生変数と先決内生変数を先決変数とする。内生変数は左欄の推計対象の変数である。

10.2 同時方程式の行列表現

$$\begin{aligned} Q_t - \alpha_1 P_t &= \alpha_0 + \alpha_2 Y_t + u_{1t} \\ Q_t - \beta_1 P_t &= \beta_0 + \beta_2 T_t + (u_{1t} + u_{2t}) \end{aligned}$$

内生変数を左に集め、右の項は外生変数と先決内生変数 (X_{t-1} 型など) で記述する。この区分は、大きな水槽の中を浸透膜や浸透壁で分けするようなもので、変数自体は相互に行き来できなくなるが、パラメーターは透過して、行き来できる。*3

第1式

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_2 \\ 0 \end{bmatrix} + u_{1t}$$

第2式

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 \\ -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \beta_0 \\ 0 \\ \beta_2 \end{bmatrix} + u_{2t}$$

全体を行列で表す

*2 W.H.Green 計量経済分析 II(2000)p835

*3 モデルによっては、元気な金魚が上を飛び越して往来している場合も見受けられる。

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix} + (u_1 + u_2)_t$$

行列の成分を表すと

$$\begin{bmatrix} Q_1 & P_1 \\ Q_2 & P_2 \\ \vdots & \vdots \\ Q_n & P_n \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_1 & T_1 \\ 1 & Y_2 & T_2 \\ \vdots & \vdots & \vdots \\ 1 & Y_n & T_n \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix} + (u_1 + u_2)_t$$

構造型を記号で表現すると

$$YB = Z\Gamma + E$$

誘導型の行列表現

行列表現では、誘導型は逆行列を用いて簡潔に表せる。

$$Y = Z\Gamma B^{-1} + EB^{-1}$$

これは、方程式体系の同時決定部分を解いたものである。誤差項を除くと次のようになる。

$$Y = Z\Gamma B^{-1}$$

J の高度な配列計算機能を用いると簡単に計算できる。

B のパラメータは、B と Γ の作成の時に定義式や明示的に移項した項の場合は数式通りとする。 Γ の定義式は数式通りとする。それ以外の OLS のパラメータは成り行きとする。

$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix}$	$\begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix}$
<pre>b=. shira_sub2 DN100 1 0.282913 1 _0.68098</pre>	<pre>]g=. shira_sub3 DN100 _5.16659 3.00044 0 _28.6958 0 4.77707</pre>

経過と説明 最初に OLS で構造型のパラメータを求め、行列に組み込む。

<p>OLS で構造型のパラメータを求める</p>	<pre>shira_sub0 DN100 _5.16659 _0.282913 3.00044 _28.6958 0.68098 4.77707 Q = -5.16659 - 0.282913P + 3.00044Y Q = -28.6958 + 0.68098P + 4.77707T</pre>	<p>各変数の単独の OLS</p>
<p>B $\begin{bmatrix} 1 & -\alpha_1 \\ 1 & -\beta_1 \end{bmatrix}$</p>	<pre>]b=. shira_sub2 DN100 1 0.282913 1 _0.68098</pre>	
<p>B^{-1} B の逆行列</p>	<pre>%. shira_sub2 DN100 0.706489 0.293511 1.03746 _1.03746</pre>	<p>1 の部分にもウエイトがかかる。</p>
<p>Γ $\begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ \beta_0 & 0 & \beta_2 \end{bmatrix}$</p>	<pre>]g=. shira_sub3 DN100 _5.16659 3.00044 0 _28.6958 0 4.77707</pre>	<p>外生変数と先決内生変数 Z のパラメータ部分である。 行の数は Z の変数の数で、 列の数はモデルの式の数で 決まる。</p>

10.2.1 モデルの同時方程式を解く

外生変数がマトリクスの場合

$Y = B^{-1}\Gamma$ は $\frac{\Gamma}{B}$ であるがマトリクスの列のサイズ (#y) が異なる場合は % . で行列除算をすることが出来ない。従って B^{-1} の右から Γ を掛け合わせ内積を求める。

$Y = B^{-1}\Gamma$ システムの回帰式 (誘導型)	$fx = (\% . b) + / . * g$ $_12.0727 \ 2.11978 \ 1.40212$ $24.4106 \ 3.11283 \ _4.95602$ $\hat{Q} = -12.0727 + 2.11978Y + 1.40212T$ $\hat{P} = 24.4106 + 3.11283Y - 4.95602T$	
---	--	--

$Z\Gamma$ を B で割れば Y の推計値 (\hat{Y}) を求められるがマトリクスのサイズが異なる場合には $Z\Gamma \% . B$ は直接計算できないので逆行列 $\% . B$ を求めて左から $Z\Gamma$ と内積演算を行う。

B $\begin{bmatrix} 1 & -\alpha_1 \\ 1 & -\beta_1 \end{bmatrix}$ b $1 \ 0.282913$ $1 \ _0.68098$	逆行列 B^{-1} $\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} -\beta_1 & \alpha_1 \\ -1 & 1 \end{bmatrix}$ $\% . b$ $0.706489 \ 0.293511$ $1.03746 \ _1.03746$
--	--

<p>内積計算では右パラメーターは(自動で)転置され、列が(同じサイズの)左パラメーターに掛け合わされる。</p>	$\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} -\beta_1 & +\alpha_1 \\ -1 & +1 \end{bmatrix}, \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix}$ <p>最終的にはこの形で B^{-1} と Γ が行 × 行で掛け合わされる。</p>
<p>誘導型のパラメーターを推計している。</p>	$\begin{aligned} (-\beta_1, +\alpha_1) \times (\alpha_0, \beta_0) &= -\alpha_0\beta_1 + \alpha_1\beta_0 = \pi_{10} \\ (-\beta_1, +\alpha_1) \times (\alpha_2, 0) &= -\alpha_2\beta_1 = \pi_{11} \\ (-\beta_1, +\alpha_1) \times (0, \beta_2) &= \alpha_1\beta_2 = \pi_{12} \\ (-1, +1) \times (\alpha_0, \beta_0) &= -\alpha_0 + \beta_0 = \pi_{20} \\ (-1, +1) \times (\alpha_2, 0) &= -\alpha_2 = \pi_{21} \\ (-1, +1) \times (0, \beta_2) &= \beta_2 = \pi_{22} \end{aligned}$ <p>($\frac{1}{\alpha_1 - \beta_1}$ の記述をを省略した。)</p>

誘導型の行列表現で誘導型のパラメータが計算されている。

10.2.2 誘導型モデル

行列を用いると誘導型のモデルが容易に作成できる。誘導型モデルは

1. バイアスを含んでいる。
2. 誘導型は識別の問題は生じない
3. 簡易モデルとして十分な精度を持っている。

\hat{Q} 推計値	先決変数に回帰式をかける (1, .2 3 {"1 DN100}) 1 28 7 1 29 4.1 1 32 7.2 1 33 5.4 1 35 5.8 1 36 6.7 1 36 5 1 38 6.3 (1, .2 3 {"1 DN100}) +/ . * : fx 57.096 76.8778 55.1496 94.363 65.8555 88.3379 65.4514 100.372 70.2518 104.615 73.6335 103.267 71.2499 111.692 77.3122 111.475	$Y = B^{-1}\Gamma$ 誘導型を直接適用 一行でモデルの計算ができる。これは全体テストである
------------------	---	---

構造型 $YB = Z\Gamma$ から誘導型 $Y = ZB^{-1}\Gamma$ に変換する。このとき、 B の逆行列 B^{-1} 即ち内生変数に移った OLS のパラメーターの逆行列が先決変数に残った OLS のパラメーターに作用して同時方程式システムのパラメーターを決定する。この白砂モデルは適正識別である。 B と Γ のサイズの差は 1 でこれは Γ の OLS の定数項部分である。

```
'key Q P hQ hP' plot ;("2) ,. |: L:0 shirasago_model DN100
pd 'eps /temp/shira_0.eps'
```

$$YB = Z\Gamma \rightarrow Y = ZB^{-1}\Gamma$$

この一行で同時方程式は計算できる。

```
shirasago_model DN100
```

Y Y^

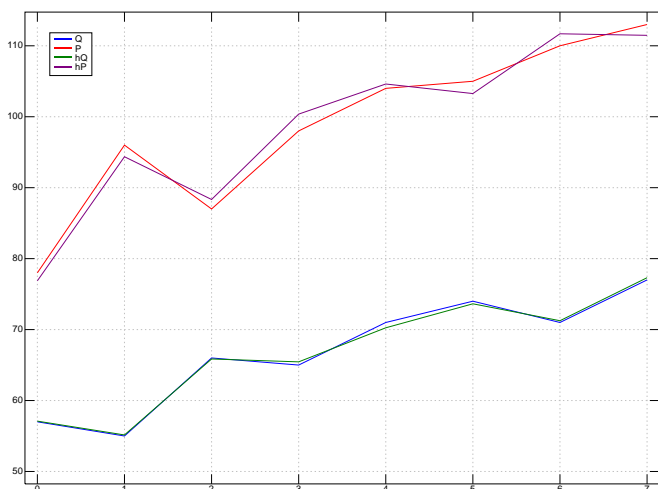


図 10.2 QP

shirasago_model DN100

```

+-----+-----+
| 57 78| 57.096 76.8778|
| 55 96| 55.1496 94.363|
| 66 87| 65.8555 88.3379|
| 65 98| 65.4514 100.372|
| 71 104| 70.2518 104.615|
| 74 105| 73.6335 103.267|
| 71 110| 71.2499 111.692|
| 77 113| 77.3122 111.475|
+-----+-----+
    
```

誘導型の回帰テスト。後に 2 段階最少自乗法を用いた場合と比較する。

(shira_reg DN100) exam_simal 1 0 1 0 divenex DN100

```

+-----+-----+-----+
| f      | _12.0727 2.11978 1.40212| 24.4106 3.11283 _4.95602|
+-----+-----+-----+
| corr   | 99.7418                | 97.9522                |
+-----+-----+-----+
| AIC    | _9.79884                | 13.2994                |
+-----+-----+-----+
| t value| _6.35087 42.0734 8.39618| 3.0314 14.5851 _7.0059 |
+-----+-----+-----+
    
```

10.3 誘導型と構造型

10.3.1 誘導型から構造型を導く

$B^{-1}\Gamma$ は誘導型をダイレクトに導く。線形代数の B^{-1} と外生変数の内積を求めるワンステップが入っているので誘導型でも十分な精度を持っている。

2 段階最少自乗法などの変数操作は、微少な精度改善に役立つ場合には、最終段階で（答案、論文、ビジネス）絞り込んだモデルを加工する場合に用いれば良い。

誘導型 $B^{-1}\Gamma$	$B^{-1}\Gamma$ は誘導型をダイレクトに導く $\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} -\beta_1 & +\alpha_1 \\ -1 & +1 \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ \beta_0 & 0 & \beta_2 \end{bmatrix}$ (%.b) +/ . * g _12.0727 2.11978 1.40212 24.4106 3.11283 _4.95602 $\hat{Q} = -12.0727 + 2.11978Y + 1.40212T$ $\hat{P} = 24.4106 + 3.11283Y - 4.95602T$	$Q_t = \pi_{10} + \pi_{11}Y_t + \pi_{12}T_t + u_{1t}$ $P_t = \pi_{20} + \pi_{21}Y_t + \pi_{22}T_t + u_{2t}$
-----------------------	--	--

10.3.2 間接最少自乗法

間接最少 2 乗法は適正識別の場合に限られる。

間接最少自乗法は構造型の仕組みを理解するのに適している。

以下のスクリプトは鈴木義一郎による。

```
a=. { |: 0 1{"1 DN100

fx=. ;("1) a ils 2 3{"1 DN100
_5.16711 _0.316224 3.09914
_30.606 0.696486 4.84058
```

(需要関数) $\hat{Q}_t = -5.16711 - 0.316224P_t + 3.09914Y_t$

(供給関数) $\hat{Q}_t = -30.606 + 0.696486P_t + 4.84058T_t$

適正識別の場合は間接最少自乗法を用いることができる。間接最少自乗法で対応関係を整理してみよう。

構造型方程式

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$$

「内生変数」

$$Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$$

Q_t =リンゴの数量

P_t =リンゴの市場価格

構造型方程式の行列型への整理

「外生変数」

$$Q_t - \alpha_1 P_t = \alpha_0 + \alpha_2 Y_t + u_{1t} \text{ (第 1 式)}$$

Y_t = 購買者の所得

$$Q_t - \beta_1 P_t = \beta_0 + \beta_2 T_t + u_{2t} \text{ (第 2 式)}$$

T_t =日照時間

$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$ $Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$ $\hat{\alpha}_0 = \pi_{10} - \frac{\hat{\pi}_{12}\hat{\pi}_{20}}{\hat{\pi}_{22}}$ $\hat{\alpha}_1 = \frac{\hat{\pi}_{12}}{\hat{\pi}_{22}}$ $\hat{\alpha}_2 = \pi_{11} - \frac{\hat{\pi}_{12}\hat{\pi}_{21}}{\hat{\pi}_{22}}$ $\hat{\beta}_0 = \pi_{10} - \frac{\hat{\pi}_{11}\hat{\pi}_{20}}{\hat{\pi}_{21}}$ $\hat{\beta}_1 = \frac{\hat{\pi}_{11}}{\hat{\pi}_{21}}$ $\hat{\beta}_2 = \pi_{12} - \frac{\hat{\pi}_{11}\hat{\pi}_{22}}{\hat{\pi}_{21}}$	$\frac{-\alpha_0\beta_1 + \alpha_1\beta_0}{\alpha_1 - \beta_1} = \pi_{10}$ $\frac{-\alpha_2\beta_1}{\alpha_1 - \beta_1} = \pi_{11}$ $\frac{\alpha_1\beta_2}{\alpha_1\beta_2} = \pi_{12}$ $\frac{\alpha_1 - \beta_1}{-\alpha_0 + \beta_0} = \pi_{20}$ $\frac{\alpha_1 - \beta_1}{-\alpha_2} = \pi_{21}$ $\frac{\beta_2}{\alpha_1 - \beta_1} = \pi_{22}$
---	--

これらの式を $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2$ について解けばよいが、このような 2 式でも対応関係は一見で分かるものではない。

10.3.3 2 段階最少二乗法

過剰識別の場合でも識別できていれば誘導型から構造型を求める方法に 2 段階最少自乗法がある。構造方程式の中で説明変数となっている内生変数の理論値を求め、その値を用いて再度 OLS を適用するものである。

このケースは適正識別であるので間接最少自乗法と一致する。

```

] fx=. ;("1) ({|: 0 1{"1 DN100)tsls 2 3{"1 DN100
_5.16711 _0.316224 3.09914
_30.606 0.696486 4.84058
    
```

(需要関数) $\hat{Q}_t = -5.16711 - 0.316224P_t + 3.09914Y_t$

(供給関数) $\hat{Q}_t = -30.606 + 0.696486P_t + 4.84058T_t$

P の理論値 \hat{P} を求める。

```

(1;2 3) exchange_ts1s DN100
57 77.3476 28 7
55 94.2694 29 4.1
66 88.6326 32 7.2
65 100.297 33 5.4
71 104.505 35 5.8
74 103.264 36 6.7
71 111.389 36 5
77 111.296 38 6.3

```

\hat{P} を最初の OLS のデータと差し替える。直接に 2 段階最少自乗法で求めた解と一致する。

```

shira_sub0 (1;2 3) exchange_ts1s DN100
_5.16711 _0.316224 3.09914
_30.606 0.696486 4.84058

```

OLS のテスト。(誘導型と比較してみる。)

```

reg0 reg_exam_ad 1 2 0{"1 (1;2 3) exchange_ts1s DN100
+-----+-----+
|f=    |_5.16711 _0.316224 3.09914|
+-----+-----+
|corr=:|99.7679          |
+-----+-----+
|AIC:  |_10.6507          |
+-----+-----+
|DW=   |2.00405           |
+-----+-----+
|t=:   |_3.2324 _9.54609 28.403 |
+-----+-----+

```

```

reg0 reg_exam_ad 1 3 0{"1 (1;2 3) exchange_ts1s DN100
+-----+-----+
|f=    |_30.606 0.696486 4.84058|
+-----+-----+
|corr=:|99.7679          |
+-----+-----+
|AIC:  |_10.6507          |
+-----+-----+

```

```
|DW=    |2.00405          |
+-----+-----+
|t=:    |_14.2655 44.6178 28.403 |
+-----+-----+
```

Script

```
NB. ----- model Shirasago p191-----
NB. OLS
shira_sub0=: 3 : ' ; ("1) ,.reg0 L:0 (1 2 0 {"1 y); 1 3 0 {"1 y'
NB. Beta
shira_sub2=: 3 : ' 1 1, . _1 _1 * 1 {"1 shira_sub0 y ' NB. DN100
NB. Gamma
shira_sub3=: 3 : ' 2 3 $ 1 1 0 1 0 1 expand ; 0 2 {"1 shira_sub0 y' NB.DN100
NB. solve model
shirasago_model=: 3 : 0
NB. u. DN100
BETA=: shira_sub2 y.
GAMMA=: shira_sub3 y.
(0 1 {"1 y.);(1,.2 3 {"1 y.) +/ .* |: (%. BETA) +/ . * GAMMA
NB. ,(shira_sub0 y.);REGNEW=(%. |: BETA) +/ . * |: GAMMA
)
```

10.4 識別

誘導型では識別の問題は生じないが、構造型に組み替えると同じ内生変数が数度計算されるので、識別が重要になる。識別は需要関数、供給関数を類似した変数で説明できるかと言うところからスタートすることが多い。しかし AIC で練り込まれた OLS は変数の数も多く多次元空間を形作っている。これらの多くは過剰識別と説明される。

適正識別は、四つ葉のクローバーを捜すようである。

モデルの行の中の 0 の数 (係数に関する先験的制約)= R

モデル全体の内生変数の数= G

	識別不能	正確に識別	過剰識別
--	------	-------	------

次数条件	$R \geq G - 1$ $R < G - 1$	$R = G - 1$	$R > G - 1$
ランク条件	$R \geq G - 1$ のとき $rank(A) < G - 1$	$rank(A) = G - 1$	$rank(A) = G - 1$

識別はモデル全体の先決変数を数えた上で、式一本毎に行う。モデル全体の係数配列を作る

過剰識別では

1. 間接最少二乗法は適正識別以外では使えない
2. 過剰識別は 2 段階最少二乗法で推計できる

モデル (1) の識別。

	Q	P	Y	T
(1)	1	α_1	α_2	0
(2)	1	β_1	0	β_2

$$G = 2$$

$$R = 1, 1$$

$$R = G - 1$$

(1) $rank A = \beta_2$ は 0 ではない。→ $rank A = 1$

(2) $rank A = \alpha_2$ は 0 でない。

10.5 ファイナルテスト

トータルテストは内生変数の推計値と観測値を比較することであるが、誘導型のモデル推計の過程で推計値が既に求まっている。

ファイナルテストでは先決内生変数にも、一期前に求めた推計値を入れて内生変数をチェックするものである。

$$\text{平均平方誤差 (RMS E)} = \sqrt{\frac{1}{t} \sum (Y - \hat{Y})^2}$$

$$\text{平均平方誤差率 (RMS E R)} = \sqrt{\frac{1}{t} \sum \left(\frac{Y - \hat{Y}}{Y} \right)^2}$$

$$\text{タイルの } U = \sqrt{\frac{\frac{1}{t} \sum (Y - \hat{Y})^2}{\frac{1}{t} \sum Y^2}}$$

t = サンプル数

Summary

誘導型回帰式

```
|: (shira_sub3 DN100) +/ . * %. shira_sub2 DN100
_1.76517 1.79187 1.38296
21.6202 3.14747 _4.68075
```

$Q = -1.76517 + 1.79187Y + 1.38296T$

$Q = 21.6202 + 3.14747Y -_4.68075T$

Total Test

shirasago_model DN100

+-----+-----+

|57 78|58.0881 76.9841|

|55 96|55.8693 93.7057|

|66 87|65.5322 88.6378|

|65 98|64.8347 100.211|

|71 104|68.9716 104.633|

|74 105|72.0082 103.568|

|71 110|69.6571 111.525|

|71 113|75.0387 111.735|

+-----+-----+

rmse shirasago_model DN100

RMSE

RMSER

+-----+-----+

|3.76711 3.1814|0.000725635 0.000268078|

+-----+-----+

10.6 幾つかの線形同時方程式モデル

10.6.1 白砂の小さな経済モデル (2)

$$\text{消費関数} \quad C_t = \alpha_0 + \alpha_1 Y_t + \alpha_2 C_{t-1} + u_t$$

定義式 $Y_t = C_t + Z_t$ 定義式なのでパラメータは不要

$$\begin{aligned}
 C_t - \alpha_1 Y_t &= \alpha_0 + \alpha_2 C_{t-1} + u_t \\
 -C_t + Y_t &= +Z_t \\
 \begin{bmatrix} C & Y \end{bmatrix} \begin{bmatrix} 1 & -\alpha_1 \\ 1 & 1 \end{bmatrix} &= \begin{bmatrix} 1 & C_{t-1} & Z \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

(1995+(i.12)), . SHIRA2

	Yt	Ct	Zt
1995	0	70	0
1996	100	76	24
1997	108	82	26
1998	110	84	26
1999	117	87	30
2000	116	87	29
2001	124	91	33
2002	131	95	36
2003	136	98	38
2004	134	97	37
2005	142	102	40
2006	149	105	44

A 国のマクロ経済データ

Y_t 国内総生産

C_t 消費支出、 C_{t-1} 1 期前の消費支出

Z_t 投資等

単位 = 10 億ドル

OLS

shira2_sub1 SHIRA2

17.5102 0.491617 0.143806

$$C_t = 17.5102 + 0.491617Y_t + 0.143806C_{t-1}$$

B	shira2_sub2 SHIRA2 1 _0.491617 _1 1
Γ	shira2_sub3 SHIRA2 17.5102 0.143806 0 0 0 1
$B^{-1}\Gamma$	shira2_reg SHIRA2 34.4428 0.28287 0.967019 34.4428 0.28287 1.96702

推計値

```

shira2_model SHIRA2
+-----+-----+
| 76 100|77.4522 101.452|
| 82 108|81.0834 107.083|
| 84 110|82.7807 108.781|
| 87 117|87.2145 117.214|
| 87 116|87.0961 116.096|
| 91 124|90.9641 123.964|
| 95 131|94.9967 130.997|
| 98 136|98.0622 136.062|
| 97 134|97.9438 134.944|
|102 142|100.562 140.562|
|105 149|105.844 149.844|
+-----+-----+

```

10.6.2 ギャリー型効用関数

$$\text{大根の需要関数 } Q_1 = \alpha_0 + \alpha_1 \frac{Y}{P_1} + \alpha_2 \frac{P_2}{P_1}$$

$$Y = P_1 Q_1 + P_2 Q_2 \rightarrow Q_2 = \frac{Y}{P_2} - \frac{P_1 Q_1}{P_2}$$

Q_1 は一見重相関回帰であるが、誘導型の（大根の）需要関数でもある。

$$\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{Y}{P_1} & \frac{Y}{P_2} & \frac{P_2}{P_1} & \frac{P_1 Q_1}{P_2} \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_1 & 0 & \alpha_2 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

```

shira3_sub1 SHIRA3
125.745 0.0016752 -56.0874

```

$$Q_1 = 125.745 + 0.0016752 \frac{Y}{P_1} - 56.0874 \frac{P_2}{P_1}$$

ここから構造方程式であるベルヌイ・ラプラス型効用関数（ギャリー型効用関数）が導出できる。

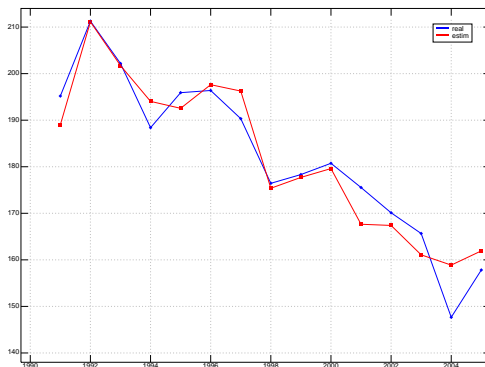
shira3_sub1 giary SHIRA3

$$a_1 = \frac{\alpha_0}{\alpha_1 - 1}, a_2 = \frac{\alpha_2}{\alpha_1}, b_1 = \alpha_1, b_2 = 1 - \alpha_1$$

```

+-----+-----+-----+-----+
|_125.956|_33481.1|0.0016752|0.998325|
+-----+-----+-----+-----+
    
```

$\hat{U} = 0.0016752 \log(-125.956 + Q_1) + 0.998325 \log(-33481.1 + Q_2)$ (ギアリー型効用関数)



赤 (理論値) が大根の消費量 (青) のターニングポイントをうまくとらえている。

図 10.3 大根の需要関数

家計のエネルギー需要モデル 2 費目分割モデルでのベルヌイ・ラプラス型モデル。

$$u = \alpha_1 \log(a_1 + q_1) + \alpha_2 \log(a_2 + q_2)$$

$\alpha_1 \log(a_1 + q_1)$ Oil

$\alpha_2 \log(a_2 + q_2)$ その他の消費

各財の限界効用

$$\frac{\alpha_1}{a_1 + q_1}$$

$\frac{\alpha_2}{a_2 + q_2}$
 q_1 , 消費量

q_2 , 価格

$$u_1 > 0$$

$$u_2 > 0$$

支出均等式: $p_1 q_1 = p_2 q_2$

$$\frac{\alpha_1}{(a_1 + q_1)p_1} = \frac{\alpha_2}{(a_2 + q_2)p_2}$$

連立して解く

$$q_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2} \frac{y}{p_1} + \frac{\alpha_1 \alpha_2}{\alpha_1 + \alpha_2} \frac{p_2}{p_1} - \frac{\alpha_2 a_1}{\alpha_1 + \alpha_2}$$

$$q_1 = A_1 \frac{y}{p_1} + B_1 \frac{p_2}{p_1} + C_1$$

$$A_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2}$$

$$B_1 = \frac{\alpha_1 \alpha_2}{\alpha_1 + \alpha_2}$$

$$C_1 = \frac{-\alpha_2 a_1}{\alpha_1 + \alpha_2}$$

実質所得 = $\frac{\text{名目所得 } y}{\text{石油価格 } p_1}$ と石油の相対価格 $\frac{p_2}{p_1}$ の 1 次関数で示される。

$$E = \alpha_0 + \alpha_1 \frac{Y}{P_1} + \alpha_2 \frac{P_2}{P_1}$$

支出均等式 : $Y = p_2 q_2 + p_1 q_1 \rightarrow q_2 = \frac{Y}{p_2} - \frac{p_1 q_1}{p_2}$

E は重相関回帰であるが、誘導型の (家計の石油) 需要関数でもある。

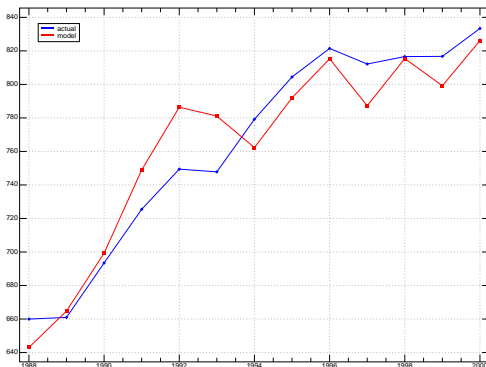
$$\begin{bmatrix} E & q_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{Y}{P_1} & \frac{Y}{P_2} & \frac{P_2}{P_1} & \frac{P_1 q_2}{P_2} \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_1 & 0 & \alpha_2 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

```
sawa_sub1 SAWA
_534.362 0.0296555 964.831
```

$$E = \alpha_0 + \alpha_1 \frac{Y}{P_1} + \alpha_2 \frac{P_2}{P_1}$$

$$E = -534.362 + 0.0296555 \frac{Y}{P_1} + 964.831 \frac{P_2}{P_1}$$

これは OLS で求めた重回帰係数であるが、誘導型のモデルともなっている。B のマトリクスが単位行列なので同時推計を行うまでもないが、同時推計のスク립トにのせると結果は次のようになる。



赤 (理論値) が石油の消費量 (青) のターニングポイントをうまくとらえている。石油の消費は景気変動ほど局所でアップダウンしない。

図 10.4 家計の石油の需要関数

B	<pre>sawa_sub2 SAWA 1 0 0 1</pre>
Γ	<pre>sawa_sub3 SAWA _534.362 0.0296555 0 964.831 0 0 0 1 0 _1</pre>
REG	<pre>sawada_reg SAWA _534.362 0.0296555 0 964.831 0 0 0 1 0 _1</pre>
	<pre>({"1 SAWA),. }:"1 sawada_model SAWA 1988 659.979 643.035 1989 660.981 664.753 1990 693.457 699.285 1991 725.468 748.958 1992 749.437 786.383 1993 747.814 781.117 1994 779.209 762.27 1995 804.412 791.981 1996 821.468 815.243 1997 812.158 787.262 1998 816.586 815.407 1999 816.718 799.203 2000 833.36 826.152</pre>

10.6.3 蓑谷の例題-小さな経済モデル

家計消費支出 (実質)	$C = \alpha_0 + \alpha_1 Y + \alpha_2 W + u_1$	(1)
民間企業設備投資 (実質)	$I = \beta_0 + \beta_1 V_{-1} - \beta_2 V_{-2} + \beta_3 T_{-1} + u_2$	(2)
家計可処分所得	$Y = \gamma_0 + \gamma_1 V + u_3$	(3)
貨幣供給	$M = \delta_0 + \delta_1 V - \delta_2 R + u_4$	(4)
国内総生産 (実質)	$V = C + I + Z$	(5)

内生変数を左辺に移項する。右辺は外生変数と先決内生変数が残る。

1	$C - \alpha_1 Y$	$= \alpha_0 + \alpha_2 W + u_1$
2	I	$= \beta_0 + \beta_1 V_{-1} - \beta_2 V_{-2} + \beta_3 I_{-1} + u_2$
3	$Y - \gamma_1 V$	$= \gamma_0 + u_3$
4	$M - \delta_1 V$	$= \delta_0 - \delta_2 R + u_4$
5	$V - C - I$	$= Z$

行列で表すと

	C	I	Y	M	V		1	W	V_{-1}	V_{-2}	I_{-1}	R	Z
(1)	1	0	$-\alpha_1$	0	0	(1)	α_0	α_2	0	0	0	0	0
(2)	0	1	0	0	0	(2)	β_0	0	β_1	$-\beta_2$	β_3	0	0
(3)	0	0	1	0	$-\gamma_1$	(3)	γ_0	0	0	0	0	0	0
(4)	0	0	0	1	$-\delta_1$	(4)	$-\delta_0$	0	0	0	0	$-\delta_2$	0
(5)	-1	-1	0	0	1	(5)	0	0	0	0	0	0	1

mino_sub2 の +- は移項を伴うので、明示的に符合を指定した。

```

mino_sub2 MINO
      1 0          0          0 _1
      0 1          0          0 _1
_0.619812 0          1          0 0
      0 0          0          1 0
      0 0 _0.618411 _1.21051 1
    
```

mino_sub3 の +- は元の式の定義式で指定されたものは明示的に、回帰式のは成り行きにとした。

```

mino_sub3 MINO
    
```

```

14.4634  _7.77127 22.3375 _11.0481 0
0.0823724      0      0      0 0
      0 0.595827      0      0 0
      0 _0.535173      0      0 0
      0 0.67628      0      0 0
      0      0      0 _10.9407 0
      0      0      0      0 1
    
```

スペースの関係で *transpose* |:する。行列の誘導型である。

```
|:  (%. mino_sub2 MINO) +/ . * mino_sub3 MINO
```

```

41.073  _7.77127      42.9316  29.2638      33.3017
0.133569      0 0.0826007 0.161687      0.133569
0.370324 0.595827      0.597478  1.16953      0.966151
0.332626 0.535173      0.536656  1.05048      0.867799
0.420328 0.67628      0.678155  1.32745      1.09661
_5.76962e_15      0 1.82199e_15 10.9407 1.21466e_15
0.62153      0      1.00277  1.96287      1.62153
    
```

トータルテスト

```
,. rmse minotani_model MINO
```

```

+-----+
|17.3873 6.65889 28.6999 37.1346 17.4736      |
+-----+
|0.00376736 0.00652471 0.00650619 0.00674606 0.00130856|
+-----+
    
```

識別

C	I	Y	M	V	W	V_{-1}	V_{-2}	I_{-1}	R	Z
1	0	α_1	0	0	α_2	0	0	0	0	0
0	1	0	0	0	0	β_1	β_2	β_3	0	0
0	0	1	0	$-\gamma_1$	0	0	0	0	0	0
0	0	0	1	$-\delta_1$	0	0	0	0	$-\delta_2$	0
-1	-1	0	0	1	0	0	0	0	0	1

$$R = 87987 > 4$$

$$G = 5 - 1 = 4$$

(1) 式のランク条件 8 カラムから 4×4 の行列式の組み合わせを作り、どれかが 0 にならなければ過剰識別。全てが 0 になれば識別不能である。

$$\begin{array}{ccc|cccc} 1 & 0 & 0 & \beta_1 & \beta_2 & \beta_3 & 0 & 0 \\ 0 & 0 & -\gamma_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\delta_1 & 0 & 0 & 0 & -\delta_2 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

システムの回帰係数

この養谷のモデルは過剰識別である。

```
(mino_sub3 MINO) %. mino_sub2 MINO
 41.073      0.133569 0.370324 _0.332626 0.420328      0  0.62153
_7.77127 9.14517e_18 0.595827 _0.535173 0.67628      0 4.996e_16
 42.9316     0.0826007 0.597478 _0.536656 0.678155      0  1.00277
 29.2638     0.161687  1.16953  _1.05048  1.32745 _10.9407  1.96287
 33.3017     0.133569 0.966151 _0.867799  1.09661      0  1.62153
```

minotani_model MINO

```
+-----+
|69.1303 10.7293 81.2368 78.1929 111.482|NB. Original Y
|76.4408 13.379 89.2311 86.4925 123.918|
|84.1501 17.0362 99.5354 94.7557 137.745|
| 92.29 20.6077 109.297 103.765 154.925|
|101.664 26.799 121.983 114.882 173.726|
+-----+
|83.3412 11.4471 100.511 62.1061 126.411| NB. Estimate Y
|85.0209 9.75124 102.032 65.083 128.87|
|91.6111 15.4479 111.152 84.0293 143.618|
|99.1147 19.5048 121.683 103.549 160.647|
|106.226 24.7564 131.33 122.433 176.247|
+-----+
```

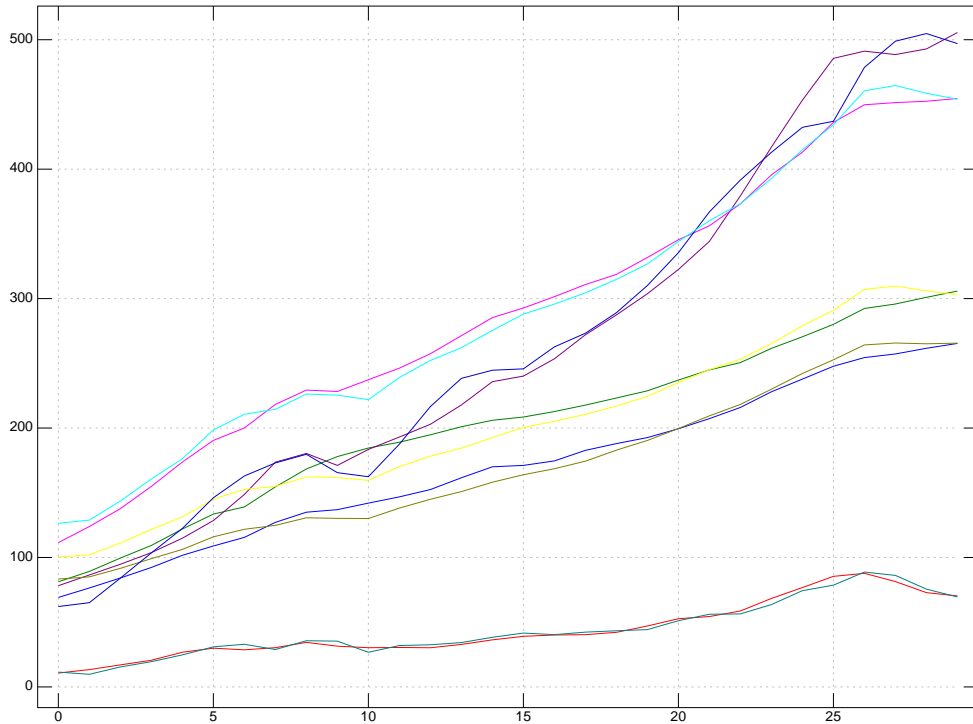


図 10.5 Minotani model-1

10.7 Klein Model

Klein(1950) の有名なマクロ経済のモデルである。

このクラインのモデル 1 は僅か 8 本の方程式で構成され、1921 年から 40 年にかけての大不況の経緯をものもの見事にトレースし、これを持って、経済学は一人前の科学として船出を遂げたと言われる。

*4

(消費)	$C_t = \alpha_0 + \alpha_1 P_t + \alpha_2 P_{t-1} + \alpha_3 (WP_t + WG_t) + e_{1t}$
(投資)	$I_t = \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \beta_3 K_{t-1} + e_{2t}$
(民間賃金)	$WP_t = \gamma_0 + \gamma_1 X_t + \gamma_2 X_{t-1} + \gamma_3 A_t + e_{3t}$
(均衡需要)	$X_t = C_t + I_t + G_t$
(民間利潤)	$P_t = X_t - T_t - WP_t$
(資本ストック)	$K_t = K_{t-1} + I_t$

*4 佐和 隆光 計量経済学への誘い 大蔵省財政金融研究所 「ファイナンシャルレビュー」 June4-1992

外生変数	G 政府非賃金支出 T 間接法人税 + 純輸出 WG 政府賃金支出 A 1930 年から測ったトレンド
先決変数	ラグ付き資本ストック 民間利潤 総需要
内生変数	各式の左に置いてある

内生変数を左に送って整理する。

$$\begin{array}{l|l}
 1 & C_t - \alpha_1 P_t - \alpha_3 W P_t = \alpha_0 + \alpha_2 P_{t-1} + \alpha_3 W G_t + e_{1t} \\
 2 & I_t - \beta_1 P_t = \beta_0 + \beta_2 P_{t-1} + \beta_3 K_{t-1} + e_{2t} \\
 3 & W P_t - \gamma_1 X_t = \gamma_0 + \gamma_2 X_{t-1} + \gamma_3 A_t + e_{3t} \\
 4 & X_t - C_t - I_t = G_t \\
 5 & P_t - X_t + W P_t = -T_t \\
 6 & K_t - I_t = K_{t-1}
 \end{array}$$

行列型で表現する。(上のナンバーは式の番号)

$$\begin{array}{c|cccccc|c|cccccc}
 & C_t & I_t & W P_t & X_t & P_t & K_t & & 1 & P_{t-1} & W G_t & K_{t-1} & X_{t-1} & A_t & G_t & T_t \\
 (1) & 1 & 0 & -\alpha_3 & 0 & -\alpha_1 & 0 & (1) & \alpha_0 & \alpha_2 & \alpha_3 & 0 & 0 & 0 & 0 & 0 \\
 (2) & 0 & 1 & 0 & 0 & -\beta_1 & 0 & (2) & \beta_0 & \beta_2 & 0 & \beta_3 & 0 & 0 & 0 & 0 \\
 (3) & 0 & 0 & 1 & -\gamma_1 & 0 & 0 & = (3) & \gamma_0 & 0 & 0 & 0 & \gamma_2 & \gamma_3 & 0 & 0 \\
 (4) & -1 & -1 & 0 & 1 & 0 & 0 & (4) & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 (5) & 0 & 0 & 1 & -1 & 1 & 0 & (5) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 (6) & 0 & -1 & 0 & 0 & 0 & 1 & (6) & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 & + (u_1 + u_2 + u_3)i & & & & & & & & & & & & & & &
 \end{array}$$

Klein の用いた数値

NB.	YEAR	C*	P	W	I	(K-1)	X	W'	G	T
	1920	39.8	12.7	28.8	2.7	180.1	44.9	2.2	2.4	3.4
	1921	41.9	12.4	25.5	-0.2	182.8	45.6	2.7	3.9	7.7
	1922	45.0	16.9	29.3	1.9	182.6	50.1	2.9	3.2	3.9
	1923	49.2	18.4	34.1	5.2	184.5	57.2	2.9	2.8	4.7
	1924	50.6	19.4	33.9	3.0	189.7	57.1	3.1	3.5	3.8
	1925	52.6	20.1	35.4	5.1	192.7	61.0	3.2	3.3	5.5
	1926	55.1	19.6	37.4	5.6	197.8	64.0	3.3	3.3	7.0

1927	56.2	19.8	37.9	4.2	203.4	64.4	3.6	4.0	6.7
1928	57.3	21.1	39.2	3.0	207.6	64.5	3.7	4.2	4.2
1929	57.8	21.7	41.3	5.1	210.6	67.0	4.0	4.1	4.0
1930	55.0	15.6	37.9	1.0	215.7	61.2	4.2	5.2	7.7
1931	50.9	11.4	34.5	-3.4	216.7	53.4	4.8	5.9	7.5
1932	45.6	7.0	29.0	-6.2	213.3	44.3	5.3	4.9	8.3
1933	46.5	11.2	28.5	-5.1	207.1	45.1	5.6	3.7	5.4
1934	48.7	12.3	30.6	-3.0	202.0	49.7	6.0	4.0	6.8
1935	51.3	14.0	33.2	-1.3	199.0	54.4	6.1	4.4	7.2
1936	57.7	17.6	36.8	2.1	197.7	62.7	7.4	2.9	8.3
1937	58.7	17.3	41.0	2.0	199.8	65.0	6.7	4.3	6.7
1938	57.5	15.3	38.2	-1.9	201.8	60.9	7.7	5.3	7.4
1939	61.6	19.0	41.6	1.3	199.9	69.5	7.8	6.6	8.9
1940	65.0	21.1	45.0	3.3	201.2	75.7	8.0	7.4	9.6
1941	69.7	23.5	53.3	4.9	204.5	88.4	8.5	13.8	11.6

Source D.N.Gjarati, Basic Econometrics 4th Edition Table 20.5

OLS	<pre>]DN101=. klein_sub0 KLEIN (1) 16.2366 0.192934 0.0898849 0.796219 (2) 8.36456 0.514518 0.225604 _0.0977402 (3) 0.194592 0.439477 0.14609 0.130245 </pre>
<i>B</i>	<pre> klein_sub2 KLEIN 1 0 _0.796219 0 _0.192934 0 0 1 0 0 _0.514518 0 0 0 1 _0.439477 0 0 _1 _1 0 1 0 0 0 0 1 _1 1 0 0 _1 0 0 0 1 </pre>

Γ	<pre> klein_sub3 KLEIN 16.2366 0.0898849 0.796219 0 0 0 0 0 8.36456 0.225604 0 _0.0977402 0 0 0 0 0.194592 0 0 0 0.14609 0.130245 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 _1 0 0 0 1 0 0 0 0 </pre>
REG	<pre> : (%. klein_sub2 KLEIN) +/- . * klein_sub3 KLEIN (1) (2) (3) (4) (5) (6) 60.832 36.268 42.8678 97.1 54.2322 36.268 0.659878 0.584474 0.546864 1.24435 0.697488 0.584474 2.23474 0.905702 1.38015 3.14045 1.76029 0.905702 _0.176587 _0.20892 _0.169421 _0.385507 _0.216086 0.79108 0.111563 _0.0604149 0.168568 0.0511479 _0.11742 _0.0604149 0.0994628 _0.0538624 0.150286 0.0456005 _0.104685 _0.0538624 1.8067 1.1375 1.73338 3.9442 2.21081 1.1375 _1.47109 _1.31925 _1.22629 _2.79033 _2.56405 _1.31925 </pre> <p>スペースの関係で <i>transpose</i> :した</p>

RMSE
<pre>.. rmse klein_model KLEIN</pre>
<pre>+-----+ 5.51403 4.28168 4.1196 9.60386 5.8179 8.05186 +-----+ 0.0027136 0.872179 0.0035421 0.00640492 0.0275251 0.000443801 +-----+</pre>

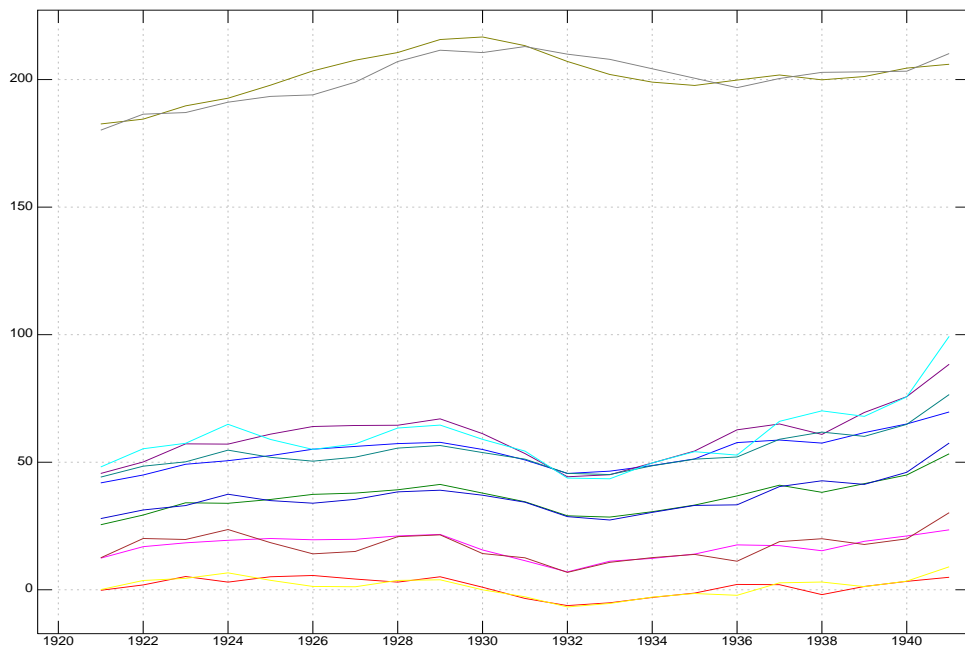


図 10.6 L.Klein

10.8 Reference

- Lawrence Klein [Economic Theory and Econometrics] Basil Blackwell 1985
 R.S.Pindyck D.L.Rubinstein[Econometric Models and Economic Forecasts]
 4th.Edition McGraw-Hill 1997
 Damodar N.Gujarati [Basic Econometrics] 4th.Edition McGraw-Hill 2003
 白砂堤津耶「例題で学ぶ・初歩からの計量経済学」1998/2007 日本評論社
 辻村江太郎「計量経済学」岩波全書 1981
 蓑谷千鳳彦 「計量経済学」多賀出版 1997

第 11 章

カルマンフィルタ

制御工学分野で開発されたカルマン・フィルタは多入力、多出力の柔軟なシステムで、平滑化や予測など適用範囲は広い。

Rudolf Emil Kalman(1930 ハンガリーブダペスト生まれ) 学部と修士は MIT で、Ph.D はコロンビア大学で。カルマンフィルタの最初の論文は 1960 年に機械関係の雑誌に発表された。Baltimore の Institute of Advanced Study の頃である。1967 年に NASA の Stanley F Schmidt を訪問し、Kalman Filter はアポロ計画で用いられた。Stanford Univ, Florida Univ の教授を歴任。

11.1 状態空間表現

電気通信の分野では 1940 年頃から *Wiener* によるウィーナーフィルタが用いられた。周波数領域で、スペクトル分解法により最適フィルタの伝達関数を求めることができ、レーダーなどに用いられた。非定常と有限観測時間の場合に拡張が難しかった。

1960 年に *Kalman* は、非定常、有限時間観測という要求を満たすフィルタを提唱した。*Wiener* の周波数領域を、時間領域で処理し、信号も回路も差分方程式で表して、定常性と無限観測時間という仮定を取り除き最適制御を行うことに成功した。カルマンフィルタはシステムパラメータや観測雑音パラメータを統計的性質を利用してシステムの状態変数や信号を推計する方法で、離散時間系に対して、マトリクス演算を主体にした逐次型の推定アルゴリズムになっている。

$$\text{システムモデル} \quad x_n = F_n x_{n-1} + G_n v_n$$

$$\text{観測モデル} \quad y_n = H_n x_n + w_n$$

表記法は何種類がある。ここでは現在時間を $x_{(n-1)}$ とする方法による。

F_n	遷移行列	$k \times k$	現象を表す物理法則や実験などで既知
-------	------	--------------	-------------------

G_n	駆動行列	$k \times m$	システムの状態変化に加わる線形の確率的変動 同じく既知
H_n	観測行列	$l \times k$	同じく既知
v_n	システムノイズ、 状態ノイズ		平均ベクトル 0、分散共分散行列 Q_n に従う m 次元の正規白色雑音
w_n	観測ノイズ		平均ベクトル 0、分散共分散行列 R_n に従う l 次元の正規白色雑音

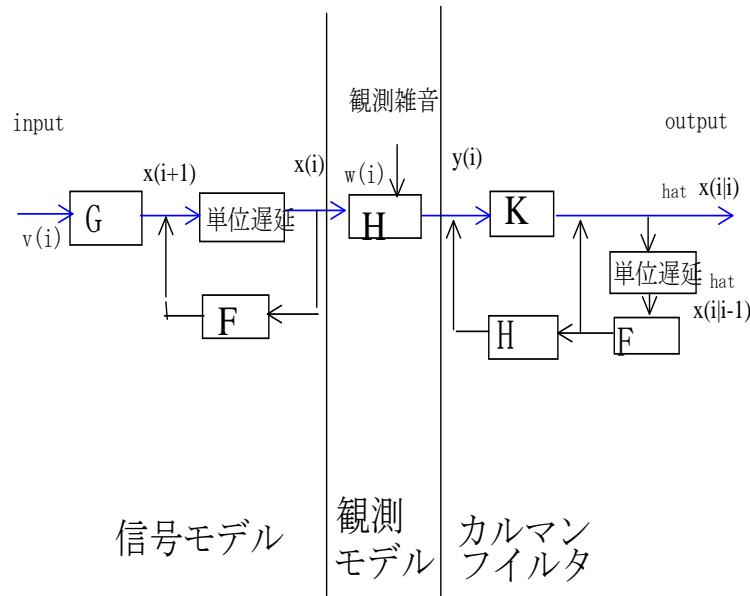


図 11.1

11.2 Kalman Filter の構成

Kalman Filter とその応用	
一期先予測	$x_{n n-1} = F_n x_{n-1 n-1}$ $V_{n n-1} = F_n V_{n-1 n-1} F_n^T + G_n Q_n G_n^T$

Filter	$K_n = V_{n n-1} H_n^T (H_n V_{n n-1} H_n^T + R_n)^{-1}$ $x_{n n} = x_{n n-1} + K_n (y_n - H_n x_{n n-1}) \text{ (ろ波推定値)}$ $V_{n n} = (I - K_n H_n) V_{n n-1}$ $V_{n n} = V_{n n-1} - K_n H_n V_{n n-1}$ $V_{n+1 n} = F_n V_{n n} F_n^T + G_n Q_n G_n^T$	(Kalman Gain) (フィルタ方程式) (推定誤差共分散行列)
固定区間平滑化	$A_n = V_{n n} F_n^T V_{n+1 n}^{-1}$ $x_{n N} = x_{n n} + A_n (x_{n+1 N} - x_{n+1 n})$ $V_{n N} = V_{n n} + A_n (V_{n+1 N} - V_{n+1 n}) A_n^T$	(平滑ゲイン) (平滑推定値) (平滑推定誤差共分散行列)
長期予測	$x_{n+2 n} = F_{n+2} x_{n+1 n}$ $V_{n+2 n} = F_{n+2} V_{n+1 n} F_{n+2}^T + G_{n+2} Q_{n+2} G_{n+2}^T$	

direction

$p(x(n-1)|Y(n-1))$ → $p(x(n)|Y(n-1))$ → $p(x(n+1)|Y(n-1))$
 Start 予測 長期予測

↓ろ波

$p(x(n-1)|Y(n))$ ← $p(x(n)|Y(n))$ → $p(x(n+1)|Y(n))$
 平滑化 予測

↓ろ波

$p(x(n+1)|Y(n+1))$

11.3 一期先予測

時刻 $n - 1$ におけるろ波の分布 $p(x_{n-1} y_{n-1})$ は既に推定されている。この分布に従う確率変数 (ベクトル) を $x_{n-1 n-1}$ とする	$x_{n-1 n-1}$ は既知
---	-------------------

予測値平均ベクトル	$x_{n n-1} = F_n x_{n-1 n-1}$	$x_{n n-1}$ は x_n の予測値 (平均) ベクトル。 x_{n-1} のフィルタ $x_{n-1 n-1}$ に推移行列 F をかける
	$y_{n n-1} = H_n x_{n n-1} (+W_n)$	
分散共分散行列	$V_{n n-1} = F_n V_{n-1 n-1} F_n^T + G_n Q_n G_n^T$	第 1 項は F_n による変換の影響 第 2 項はシステムノイズ v_n の影響を表す

11.3.1 Filtering

Kalman Gain K	$K_n = \frac{V_{n n-1} H_n^T}{H_n V_{n n-1} H_n^T + R_n}$	$H_n V_{n n-1} H_n^T + R_n$ は分散共分散行列
推計値	$x_{n n} = x_{n n-1} + K_n (y_n - H_n x_{n n-1})$ $y_{n n} = H_n x_{n n}$	$y_n - H_n x_{n n-1}$ は y_n の予測誤差
分散共分散行列の更新	$V_{n n} = (I - K_n H_n) V_{n n-1}$ or. $V_{n n} = V_{n n-1} - K_n H_n V_{n n-1}$	$K_n H_n V_{n n-1}$ は観測値 y_n

x_{n-1} のフィルタ	$x_{n-1 n-1}$
推移行列	F_n
F_n による変換の影響	$F_n V_{n-1 n-1} F_n^T$
システムノイズ v_n の影響	$G_n Q_n G_n^T$
Y_n の予測誤差	$Y_n - H_n x_{n n-1}$
予測誤差の分散共分散行列	$H_n V_{n n-1} H_n^T + R_n$

11.4 Working Example

電力消費量と経済変数の関係のサンプルデータ

出典:杉原敏夫「適応的モデルによる経済時系列分析」1996 工学図書

用いるデータは全データを初年度の数値で割り倍率に変換したデータである。このため *divbyfst* を用意した。

経済指標のインデックス

1	2	3	4	5	6
発電量	人口	鉱工業生産指数	実質 GDP	外国為替	原油輸入量
1000 万 KWH	万人	1990=100	10*8Yenn	Yen\$	10000KL
27223	10254	39.5	147992	357.8	16688
30759	10467	45	165899	357.7	19583
32699	10610	46.2	179171	314.8	22104

取得する変数を指定する

X=:1 1 3 5 6;3 3 4;4 4;5 5;6 6 NB. 自然数での列順

X0 NB. for Lagged

```
+-----+-----+-----+-----+
|1 1 3 5 6|3 3 4|4 4|5 5|6 6| NB. 自然数での列順
+-----+-----+-----+-----+
```

X2 NB. for multiple or single equation

```
+-----+-----+-----+
|1 2 3 6|3 4|4|5|6|
+-----+-----+-----+
```

F/状態方程式

カルマンフィルタは多変数入力、多変数出力である。Example は重相関係のモデルであるので、状態方程式は重相関行列の束である。

説明変数の組み合わせ（多変数）。ここでは AIC と決定係数、F 検定、t 検定を用いて選んでいる。

カルマンフィルタの変数は $x_{1k+1,k}$ のように一期ずらして用いる。自己の変数を一期遅らして用いて、推移を記述する。

$$x_{1k+1,k} = 0.484135 + 1.10611x_{1k} - 0.181381x_{3k} - 0.111986x_{5k} - 0.157438x_{6k}$$

$$x_{3k+1,k} = 0.159392 + 0.846931x_{3k} + 0.0854512x_{4k}$$

$$x_{4k+1,k} = 0.0938716 + 0.989689x_{4k}$$

$$x_{5k+1,k} = 0.0177466 + 0.925283x_{5k}$$

$$x_{6k+1} = 0.291118 + 0.809596x_{6k}$$

mk_F は $X0$ で変数を重複して指定すると自動的に一期遅らせた係数を求め、 F のマトリクスを作成する。

```

X0 mk_F divbyfst SGDAT
      1      0      0      0      0      0
0.484135 1.10611 _0.181381      0 _0.111986 _0.157438
0.159392      0 0.846931 0.0854512      0      0
0.0938716      0      0 0.989689      0      0
0.0177466      0      0      0 0.925283      0
0.291118      0      0      0      0 0.809596

```

H /観測方程式

2 番目の変数 (人口) は用いていないので指定しない。

```
mk_H 1 3 4 5 6
```

$H_k = NB$. 観測行列

```

0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1

```

$y_k = H_k x_n + w_k$ NB. 観測方程式

$$\begin{bmatrix} y_{1k} \\ y_{3k} \\ y_{4k} \\ y_{5k} \\ y_{6k} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_{1k} \\ x_{3k} \\ x_{4k} \\ x_{5k} \\ x_{6k} \end{bmatrix}$$

W /観測ノイズ

残差平方和を n で除した値を用いている。

```

X2
+-----+-----+
| 1 2 3 6|3 4|4|5|6|
+-----+-----+

```

y を各ボックスの最後に置く。トレンド型の単回帰が一度にうまく計算できない。Loop に戻す。

```

X2 mk_W0 divbyfst SGDAT
0.0108862          0          0          0          0
          0 0.00752976          0          0          0
          0          0 0.00299007          0          0
          0          0          0 0.00456877          0
          0          0          0          0 0.0413174

```

V/状態雑音

状態雑音は当初に計算してしまう方法とフィルタの更新毎に計算する方法がある。当初に一律に計算する方法は各変数の標準偏差を用いた。

kalman_0, *kalman_1* とスクリプトを 2 本用意した。

```

X0 mk_V0 divbyfst SGDAT
0          0          0          0          0          0
0 0.00284196          0          0          0          0
0          0 0.00833951          0          0          0
0          0          0 0.00121597          0          0
0          0          0          0 0.00473336          0
0          0          0          0          0 0.00926743

```

状態空間表現は経済などと異なった時間表現を用いるので慣れが必要である。状態 $z(n)$ から見て $j < n$ の場合は観測値 (+ まで) が n より前なので n までの状態を推計する。これを予測という。

$j = n$ になると丁度時間が合っているので、見繕いをする。これを濾波と言う。

$j = N$ の場合は n を過ぎているので過去を振り返って観測期間全体を見渡して状態を推定する。これを平滑化と言う。

				$z(n)$	N
予測	$j < n$	$Y(i)$	+	j	n
					N
濾波	$j = n$	$Y(n)$	+	$j = n$	N
平滑化	$j > n$	$Y(N)$			+
					$j = N$

11.4.1 Script の構成

フィルタを構成する	$compose F, H, V, W$	<pre>'X0 X1'=: x. 'KF KH KV KW'=: x kalman_multi_sub0 y</pre>
観測値入力	Input $x_{n-1 n-1}$	<pre>for_ctr i. # y. do. DAT0=:1,.(<: ~./:~ ; X0){"1 y TMPDAT=: ctr{DAT0</pre>
予測値	$x_{n n-1} = F_n x_{n-1 n-1}$ $y_{n n-1} = H_n x_{n n-1}$	<pre>KY=:KH mp KX=:KF mp TMPDAT NB. xn,n-1</pre>
Kalman Gain	$KG = \frac{V_{n n-1} H_n^T}{H_n V_{n n-1} H_n^T + W_n}$	<pre>KG=: (KV mp :KH) mp %.KW+KH mp KV mp :KH 逆行列除算でなく逆行列を右から内積演算</pre>

推定値	$x_{n n} = x_{n n-1} + KG(y_n - H_n x_{n n-1})$ $y_{n n} = H_n x_{n n-1}$	$KPRY =: KH \text{ mp } KPRX =: KX$ $+ KG \text{ mp } (\}.TMPDAT) - KY$
-----	---	---

11.4.2 計算

観測値とフィルタリングの結果及びフィルタを通す前の一期先予測値

(X0;<X2) kalman_f0	(X0;<X2) kalman_1
divbyfst SGDAT	divbyfst SGDAT
観測値 予測値 推計値	観測値 予測値 推計値
xn n-1 xn n	native xn n-1 xn n
1 1 1	(X0;<X2) kalman_f1 divbyfst SGDAT
1.12989 1.11057 1.23058	1 1.11057 1.13944
1.20115 1.20973 1.29353	1.12989 1.11916 1.23058
1.34401 1.27441 1.42391	1.20115 1.16256 1.29353
1.49098 1.40737 1.51304	1.34401 1.33802 1.42391
1.46097 1.50847 1.4897	1.49098 1.51159 1.51304
1.52088 1.48375 1.59931	1.46097 1.46866 1.4897
1.64725 1.58307 1.71265	1.52088 1.52564 1.59931
1.7191 1.69911 1.78938	1.64725 1.64946 1.71265
1.8257 1.77483 1.91163	1.7191 1.71976 1.78938
1.9161 1.89384 1.96767	1.8257 1.82515 1.91163
1.88829 1.95699 1.95636	1.9161 1.91522 1.96767
1.92168 1.94227 2.01039	1.88829 1.88697 1.95636
1.91926 1.99203 2.01631	1.92168 1.92043 2.01039
2.04059 1.99622 2.14887	1.91926 1.91822 2.01631
2.13863 2.12646 2.21354	2.04059 2.03778 2.14887
2.21845 2.19803 2.31982	2.13863 2.14229 2.21354
2.20957 2.29884 2.32718	2.21845 2.21942 2.31982
2.35154 2.30283 2.49512	2.20957 2.20983 2.32718
2.44929 2.4654 2.55454	2.35154 2.35153 2.49512
2.58855 2.53275 2.66336	2.44929 2.44957 2.55454
2.7829 2.64787 2.84494	2.58855 2.58888 2.66336
2.87665 2.8321 2.93084	2.7829 2.78321 2.84494
2.89557 2.91962 2.97237	2.87665 2.87689 2.93084
2.92293 2.95647 3.02271	2.89557 2.89567 2.97237
3.08724 3.00205 3.18978	2.92293 2.92297 3.02271
	3.08724 3.08728 3.18978

予測値、推計値の相関係数

原データと予測値、推計値の相関係数を求める。相関係数は良好な値が得られている。プロトタイプ (f0) とノイズの微調整 (f1) の効果がはっきりと現れていない

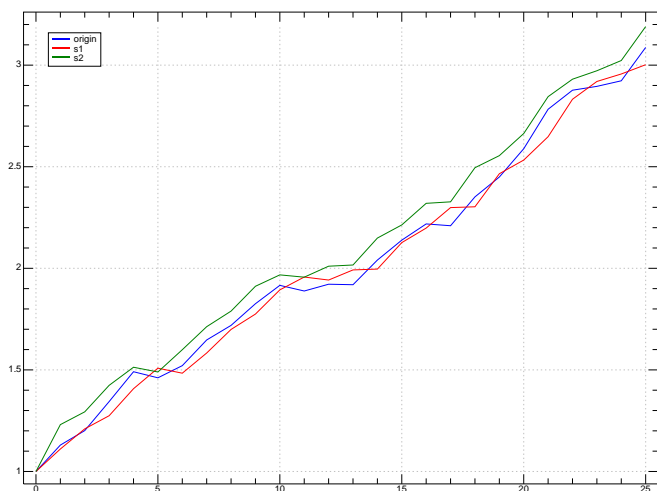


図 11.2 V 固定

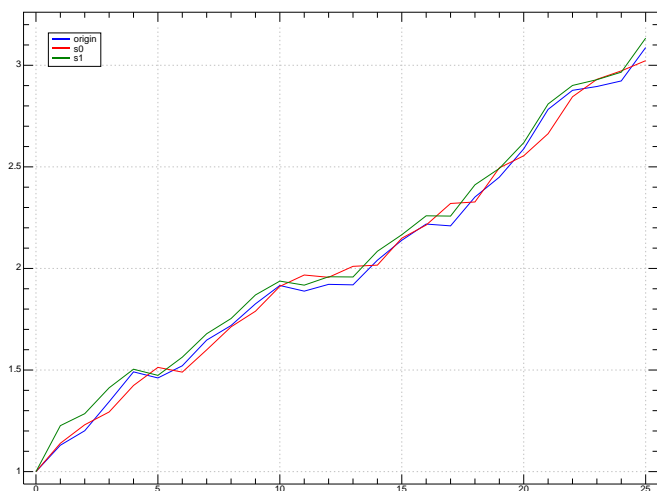


図 11.3 V 更新

```
cortable (X0;<X2) kalman_f0 a
```

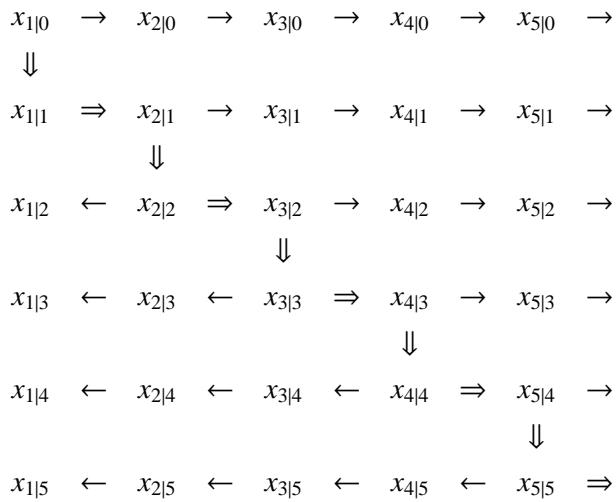
```
原データ 予測値 推計値
```

```
1 0.99592 0.998849
0.99592 1 0.995667
0.998849 0.995667 1
```

```
cortable (X0;<X2) kalman_f1 a
```

```
1 0.999236 0.998849
0.999236 1 0.998542
0.998849 0.998542 1
```

11.5 固定区間平滑化



\Rightarrow 予測 $x_{n|n-1}$
 \downarrow フィルタ $x_{n|n}$
 \rightarrow 長期予測
 \leftarrow 平滑化 $x_{n|N}$

$$A_n = V_{n|n} F_{n+1}^T (V_{n+1|n})^{-1}$$

$$x_{n|N} = x_{n|n} + A_n (x_{n+1|N} - x_{n+1|n})$$

$$V_{n|N} = V_{n|n} + A_n (V_{n+1|N} - V_{n+1|n}) A_n^T$$

$$A_n = V_{n|n} F_{n+1}^T (V_{n+1|n})^{-1}$$

$$N = 2$$

$$x_{1|2} = x_{1|1} + A_1(x_{2|2} - x_{2|1})$$

$$N = 3$$

$$x_{2|3} = x_{2|2} + A_2(x_{3|3} - x_{3|2})$$

$$x_{1|3} = x_{1|1} + A_1(x_{2|3} - x_{2|1})$$

$$N = 4$$

$$x_{3|4} = x_{3|3} + A_3(x_{4|4} - x_{4|3})$$

$$x_{2|4} = x_{2|2} + A_2(x_{3|4} - x_{3|2})$$

$$x_{1|4} = x_{1|1} + A_1(x_{2|4} - x_{2|1})$$

$$N = 5$$

$$x_{4|5} = x_{4|4} + A_4(x_{5|5} - x_{5|4})$$

$$x_{3|5} = x_{3|3} + A_3(x_{4|5} - x_{4|3})$$

$$x_{2|5} = x_{2|2} + A_2(x_{3|5} - x_{3|2})$$

$$x_{1|5} = x_{1|1} + A_1(x_{2|5} - x_{2|1})$$

$$A_n = V_{n|n} F_{n+1}^T (V_{n+1|n})^{-1}$$

$$V_{n|N} = V_{n|n} + A_n(V_{n+1|N} - V_{n+1|n})A_n^T$$

$$A_1 = V_{1|1} F^T (V_{2|1})^{-1}$$

$$A_2 = V_{2|2} F^T (V_{3|2})^{-1}$$

$$A_3 = V_{3|3} F^T (V_{4|3})^{-1}$$

$$A_4 = V_{4|4} F^T (V_{5|4})^{-1}$$

$$N = 2$$

$$V_{1|2} = V_{1|1} + A_1(V_{2|2} - V_{2|1})A_1^T$$

$$N = 3$$

$$V_{2|3} = V_{2|2} + A_2(V_{3|3} - V_{3|2})A_2^T$$

$$V_{1|3} = V_{1|1} + A_1(V_{2|3} - V_{2|1})A_1^T$$

$$N = 4$$

$$V_{3|4} = V_{3|3} + A_3(V_{4|4} - V_{4|3})A_3^T$$

$$V_{2|4} = V_{2|2} + A_2(V_{3|4} - V_{3|2})A_2^T$$

$$V_{1|4} = V_{1|1} + A_1(V_{2|4} - V_{2|1})A_1^T$$

$$N = 5$$

$$V_{4|5} = V_{4|4} + A_4(V_{5|5} - V_{5|4})A_4^T$$

$$V_{3|5} = V_{3|3} + A_3(V_{4|5} - V_{4|3})A_3^T$$

$$V_{2|5} = V_{2|2} + A_2(V_{3|5} - V_{3|2})A_2^T$$

$$V_{1|5} = V_{1|1} + A_1(V_{2|5} - V_{2|1})A_1^T$$

11.5.1 Sample data

標準誤差=誤差の標準偏差

No	Year	発電量 1000 KWH	人口 万人	鉱工業 生産指 数 1990=100 10*8Yen	実質GDP Yen/\$	外国為替 Yen/\$	原油輸入量 10000KL
1	1969	27223	10254	39.5	147992	357.8	16688
2	1970	30759	10467	45	165899	357.7	19583
3	1971	32699	10610	46.2	179171	314.8	22104
4	1972	36588	10760	49.6	188309	301.1	23833
5	1973	40589	10910	57.1	204880	280	28667
6	1974	39772	11057	54.8	214714	300.9	28048
7	1975	41403	11194	48.7	214714	305.2	26281
8	1976	44843	11309	54.1	223088	293	26859
9	1977	46799	11417	56.4	232011	240	27789
10	1978	49701	11519	60.1	242916	195.1	27018
11	1979	52162	11616	64.5	255062	239.9	28049
12	1980	51405	11706	67.6	269090	203.6	25683
13	1981	52314	11788	68.2	277970	220.3	23024
15	1982	52248	11869	68.5	287143	235.3	21469
15	1983	55551	11948	70.5	296619	232	20779
16	1984	58220	12031	77.3	309671	251.6	21460
17	1985	60393	12105	80.2	323915	200.6	19833
18	1986	60151	12167	80	333309	160.1	19452
19	1987	64016	12226	82.1	349769	122	18538
20	1988	66677	12278	90.7	363567	125.9	19385
21	1989	70468	12325	96.1	380709	143.4	20969
22	1990	75759	12361	100	399043	135.4	22876
23	1991	78311	12404	101.7	416038	125.3	24270
24	1992	78826	12445	95.5	420622	124.7	25123
25	1993	79571	12476	91.2	419765	111.9	25514
26	1994	84044	12503	92	421922	99.8	27081

11.6 divbyfst

```

divbyfst SGDAT
      1      1      1      1      1      1
1.12989 1.02077 1.13924  1.121 0.999721 1.17348
1.20115 1.03472 1.16962 1.21068 0.879821 1.32454
1.34401 1.04935  1.2557 1.27243 0.841532 1.42815
1.49098 1.06398 1.44557  1.3844  0.78256 1.71782
1.46097 1.07831 1.38734 1.45085 0.840973 1.68073
1.52088 1.09167 1.23291 1.45085  0.85299 1.57484
1.64725 1.10289 1.36962 1.50743 0.818893 1.60948
  1.7191 1.11342 1.42785 1.56773 0.670766 1.66521
  1.8257 1.12337 1.52152 1.64141 0.545277 1.61901
  1.9161 1.13283 1.63291 1.72349 0.670486 1.68079
1.88829  1.1416 1.71139 1.81827 0.569033 1.53901
1.92168  1.1496 1.72658 1.87828 0.615707 1.37967
1.91926  1.1575 1.73418 1.94026  0.65763 1.28649
2.04059  1.1652 1.78481 2.00429 0.648407 1.24515
2.13863  1.1733 1.95696 2.09248 0.703186 1.28595
2.21845 1.18051 2.03038 2.18873 0.560648 1.18846
2.20957 1.18656 2.02532 2.25221 0.447457 1.16563
2.35154 1.19232 2.07848 2.36343 0.340973 1.11086
2.44929 1.19739  2.2962 2.45667 0.351873 1.16161
2.58855 1.20197 2.43291  2.5725 0.400783 1.25653
  2.7829 1.20548 2.53165 2.69638 0.378424 1.37081
2.87665 1.20967 2.57468 2.81122 0.350196 1.45434
2.89557 1.21367 2.41772 2.84219 0.348519 1.50545
2.92293  1.2167 2.30886  2.8364 0.312745 1.52888
3.08724 1.21933 2.32911 2.85098 0.278927 1.62278

```

11.7 Script

```
kalman_f0=: 4 : 0
```

```

NB. (X0;<X2) u divbyfst SGDAT
TMPMAT=. x kalman_f0_calc y NB. xn|n,xn_n-1
({."1 y),.(1,){."1 TMPMAT),.{"1 TMPMAT
)

    kalman_f0_calc=:4 : 0
NB. x is (X0;<X2)
NB. usage: (X0;<X2) kalman_0 a=. divbyfst data
'X0 X1'=: x
'KF KH KV KW'=: x kalman_multi_sub0 y
ANSP=. ANSF=. <'
DAT1=: 1,.(<: ~./:~ ; X0){"1 y
NB. find yk(Ky KPRY)
NB. -----
for_ctr. i. # y do.
    TMPDAT=. ctr{DAT1 NB. by one step is 1 raw
    KY=. KH mp KX=:KF mp TMPDAT      NB. xn,n-1
    ANSF=. ANSF,<KY
    KG=.KV mp (|:KH) mp %. KW +KH mp KV mp |:KH NB. mp %. is mp and reverse
    NB. KG is kalman gain
    NB. calc order of formura //mp %./
    KPRX=. KX + KG mp (}.TMPDAT) - KY
    KPRY =. KH mp KPRX
    ANSP=. ANSP,<KPRY
end.
XX=. 0 5 {"1(;"1),.}.ANSP),,;"1),.}.ANSF NB. xn|n,xn|n-1
)

```

第 A 章

J のインストールとスタート

A.1 J の入手とインストール

<http://www.jssoftware.com>

からダウンロードする。

DOWNLOAD → Stable で各自のプラットフォームを選ぶと最新の版がダウンロードできる。

WIN32/WIN64 MAC/OSX (PPC,INTEL) LINUX/32 64 POCKETPC(CE4.0) がある。

インストーラに従って YES を押していけばインストールできる。

インストール先はデフォルトでは Documents add Settings であり

j602

j602-user

の 2 つのフォルダに分かれる。

ユーザライセンス: 現在は無料。^{*1}

J602 からはレジストリが切断された。USB や CD-ROM に入れておいて、USB や CD-ROM から起動できる。

HDD のどこにでも copy (move) して置くことができる。

自分の好みの環境を作れる。

j602 の下に classes を移した方が便利である。

A.2 J の起動

A.2.1 実行画面-ijx

J の起動はウィンドウ上の J のアイコンか START から J を探し、クリックすると上の画面が現れる。ijx の画面で実行環境である。

J はインタプリタ (通訳) であり、対話しながら 1 行ずつスクリプトを吟味して仕上げていく。^{*2}速度はコンパイラの方が勝るが、J は C と J のマクロで書かれたか

^{*1} Licence Key を要求される場合はメールでユーザー登録をすれば直ちにキーを送ってくる。

^{*2} コンパイラ (翻訳) 形式の言語は、機械語に変換したプログラムが実行環境でアプリケーションとして提



図 A.1 ijx 画面

なり高速な言語で、時には C に速度競争を挑んで勝つことがある。
画面に次のように打ち込んでみて、6 と出れば正常に機能している。

```
3+3
6
```

A.2.2 スクリプト記述用画面-ijx

スクリプトを書いたり、編集する画面である。

新規作成	<i>File</i> → <i>New.ijx</i>	
編集用にファイルを読み込む	<i>File</i> → <i>Open (temp)</i> <i>File</i> → <i>Open User/System/Recent</i>	間違って書き込みする事を防止するため read only がかかっている。編集する場合は <i>Edit</i> の <i>Read only</i> のチェックを外す 編集用に読み込んだ場合は <i>Run</i> → <i>Window</i> か <i>Ctrl + W</i> で ijx にファイルを認識させる。
実行用にファイルを読み込む	<i>Run</i> → <i>File</i>	

もう 1 枚の画面が現れる。

J は数学と対話しながらスクリプトを創っていくことができる。

ユーザーも多くが大学や研究所の人達なので、多くのソースコードが公開されている。

インタプリッタはソースコードが暗号化されていない限り見ることができるが、

供される。プログラムのソースコードが提供されることは希である。



図 A.2 ijs 画面

誤って書き換えてしまう事故も起こりやすい。

提供された SCRIPT はワープロやエディタで読んだり、印刷して用い、通常は編集画面の *Read only* を外さないようにしたい。

編集画面をロードするには *Ctrl+W* か *run* → *Window* のもう 1 動作が必要である。単にロードする場合は、上の *Run* → *File* をマウスでクリックするとダイアログボックスが現れるので、こちらからロードする。

スクリプトを作成する場合は、ijs ウィンドウを大きくして、*Window* → *Tile* にすると上下に画面が分かれる。

ijs の画面に書いたスクリプトは、マウスで *Run* → *Window* で *ijx* 画面に渡すか、キー操作で *Ctrl W* を同時に押して、実行画面に渡し、実行画面に移る。

A.2.3 画面の切り替え

ijx *ijs* の画面の切り替え方法。何枚もの *ijs* を開いたときに便利。

1. マウス
2. *F12* キー
3. *ALT Tab*
4. *WINDOW* → *FORM* で フォームマネージャーを用いる。

ijx *ijx* の同時表示は *J* の *WINDOW* をクリックして *Cascade*, *Tile* などを選択する

A.3 ディレクトリ/フォルダ

J602 はレジストリを切断したので、USB や CDROM から起動できる。副作用としてフォルダの取り扱いが少し面倒になった。

ユーザーファイル *jpath* を用いると (*jpath* が認識できれば) `require jpath '~user/classes/numeric_reci` のように扱うことができる。

main.ijs xx_main.ijs といえばバッチファイルでファイルの読み込みや毎回行うハウスキューピング用のファイルを記入してある。各自のフォルダ構成に合わせて、変更する。

プロジェクト Jの run に本格的なプロジェクトマネージャが入っている。Jのシステムには入っているツール類やユーザーファイルを一括管理できる。^{*3}

^{*3} 筆者は手軽に xx_main.ijs' で済ませている。

第 B 章

J 言語超入門

B.1 アイバーソン言語の特色

数値計算言語 J は APL2 や FORTRAN90/95 と並ぶ最高水準の数値計算言語である。APL のロジックを一人で考えた *K.E.Iverson* のキャリアを下に *Roger.Hui* の強力を得て開発されてから 20 年近い歴史を有し、今も *K.E.Iverson* の故郷カナダのトロントを拠点に改良が重ねられている。^{*1}

^{*2}

言語 J 言語であるので名詞、動詞、副詞がある。

J	他の言語	数学
名詞	変数	
動詞	関数	関数
副詞		作用素

1. 名詞は $a = . 1 2 3$ や $CLASS0 = : 2 2 \$ 2 3 4 5$ のように数列や文字列を定義する。
2. 英字であれば大文字でも小文字でもよいが、漢字や先頭に数字がくる語は用いることができない。
3. 定義は $=$ 、 $:=$ をもちいる。 $=$ は等しいかどうかの比較を行う動詞である。
4. 動詞は $+ - * \%$ のように名詞に作用して計算する。副詞は $+ / 1 2 3 4$ の $/$ のように独自で名詞に作用するのではなく動詞に作用してパワーアップする。

^{*3}

キーボード記号を用いる 数学は記号で記述する。アイバーソンは APL を開発したときに、英語やフランス語などの特定の言葉に依存することは一切やめ、 $\times, \div, \rho, \iota$ の様なギリシャ文字や数学記号に似せた記号をを独自に作成した。J はこれをキーボード記号に載せ変えた。

^{*1} 数式処理言語ではないがプログラムで数式処理はできる

^{*2} APL 数学とコンピューターの巧な連携を考えていたアイバーソンがハーバードから IBM に移ってほとんど一人でアルゴリズムを作成した言語 A Programming Language。

^{*3} $:=$ はグローバル定義。プログラムを定義したり、名詞を読み込ますときに用いる。プログラムの中で用いると、終了後も名詞などが破棄されないで残ってしまう。

記号は* . *:のようにキーボード記号そのままと、後ろにピリオド(ドット) コロンの付いたものの3種類があり、それぞれ機能は異なる。キーボード記号が足りないのでこのように拡張した。更に、A. a. p: q:のようにアルファベットの大文字、小文字にピリオド、コロンをつけたものも使用している。

全て記号で完遂するというポリシーをJ開発の後継者は完全には継承せず、ループでは *for, while, if, end* などの用法を導入した。他にも幾つかの英語が用いられている。

記号は自由に再定義できる 記号や、アルファベットに割り付けられた機能は全て好きな名前でも再定義できる。(漢字や2バイト文字は未だ使えないが)

tasu=:plus=: +

waru=: %

など各国の言葉で自由に定義できる。

型やサイズの宣言は不要 スクリプトの冒頭で型やサイズを宣言をする必要はない。

配列を自由に扱える Jは一般多元配列環の通った数少ない言語である。

スカラ、ベクトル、(多元)マトリクスを自由に(ループを用いなくて)扱うことができる。

記号には単項と両項がある $3+3$ の $+$ は $+$ の前後に引数をとるので両項である。

$+3j.2$ の $+$ は右引数に(のみ)働き、複素数の共役で $3j.2$ となる。このように単項と両項には(連想できる)別個の機能が割り付けられている。

計算順序は後ろから前へ プログラムの解釈は後ろから順に単項か両項を区別しながら自動的に行われる。慣れてしまえばこの方が楽であるが、最初は紛らわしいときは括弧を用いるとよい。

$3*2-5$ NB. $3*(2-5)$

_9

$(3*2)-5$

1

動詞(関数)は全て平等 かけ算や割り算が優先することはない。(最初は括弧で見やすくするとよい)数学者で創世記のコンピューター科学者あったからできる大胆な発想である。

豊富なグラフィックス機能 Jが著名な大規模ビジネスツールのグラフィックスに組み込まれているらしい。単独で専用のグラフィックツール並の使いやすしい機能を備えている。

B.1.1 高級関数電卓

最初に高級関数電卓のように *ijx* の画面のみで体験しよう。

最近次の ReferenceCard(J602) を *Henry · Rich* が公開した。^{*4}

http://202.67.223.49/jwiki/HenryRich?action=AttachFile&do=get&target=JRefCardv601_20060413.pdf

http://homepage3.asagaya_avenue/の APL/ASSOCIATION に Quick Reference が 2 つあがっており、数値例が丁寧に載っている。

また J の HELP の中に J の PRIMER が入っている。

ここでは *Reference CARD* を前提にすすめる。

1. 四則演算

Arithmetic Dyad 最初の 4 行をタイプする。

```

2+8
10
2-8
_6
2*8
16
2%8
0.25

```

_6 アイバーソンは数学の曖昧さを嫌い、演算のマイナスとマイナス符号を区別し、数の前につく符号はアンダーバー (_) とした。3 - _6 と記述でき括弧は不要。

タイプライタから PC のキーボードに移行するときになぜ \times, \div の様な基本キーを入れ忘れたかは謎であるが、皆苦労している。割算は似ている % に割り付けた。(パーセント用の記号は無い)

```

2^8 Exponent 256 2^8
2^.8 Log 3 log_2 8
3%:8 Root 2 ^3 sqrt 8

```

2. 各記号は単項と両項の 2 面の顔を持つ。

Shorthands 単項

```

*: 9 Square 81 9^2
%: 9 Squareroot 3 sqrt 9
3%:8 Root 2 ^3 sqrt 8
^ 1 Exponential 2.718 e

```

3. スカラ ベクトル マトリクスを同等に扱える。

数学と J の用語を示す。数学のベクトルやマトリクスの計算にも事務計算にも適用できる。J は一般多元配列環の通った数少ない言語である。いちいちループを用いる必要はない。

^{*4} google で J;referencecard でも出てくる

ベクトル、マトリクスの数字の間にはスペースを入れる。

<i>math</i>	<i>J</i>	例
スカラ	アトム	3人
ベクトル	リスト	あるクラスの生徒の身長一覧
マトリクス	テーブル	ミスコンテスト出場者の3サイズの表 レポート ある会社の各支店の月別売り上げ

*5

ベクトル] a0=. 2 4 6	a0 + a0
リスト	2 4 6	4 8 12
マトリクス] a1=. 2 3 \$ 3 4 5 6 7 8	a1 + a1
テーブル	3 4 5 6 7 8	6 8 10 12 14 16

\$ Shape は両項では 2 3 \$ 1 2 3 4 5 6 のようにマトリクスの形を指定し、単項ではマトリクスやベクトルのサイズを示す。

4. ベクトルやマトリクスの演算では作用する方向を明示的に示さなければならないことがある。これをランクと言い、"0 "1 "2 とあらわす。それぞれ、スカラ、ベクトル、マトリクス方向に作用する。なにも指定しないとデフォルト(の指定)に従う。

```
a1 +"1 a0
```

```
5 8 11
```

```
8 11 14
```

5. マトリクス計算用の動詞が用意されている。

逆行列	a2	%. a2
	2 0 5	0.26638 _0.102041 0.0429646
	3 8 16	_0.286788 0.0816327 0.0182599
	18 19 7	0.0934479 0.0408163 _0.0171858

*5 ベクトルとリスト、マトリクスとテーブルの用語は実際には混同して使われている

内積	<pre> a1 3 4 5 6 7 8 </pre>	<pre> a1 +/ . * a2 108 127 114 177 208 198 </pre>
行列の割算		<pre> 3 4 5 % . a2 0.6058 _0.442535 0.35768 </pre>

逆行列は正則か縦長のマトリクスで計算できる。数学でいうランクが不足した場合は、エラーになる。

数学のランクは行列式 `-/ . *`で確認する。

B.2 数学用の動詞

Σ	<pre> +/ 2 3 4 5 6 20 </pre>	<p><code>/</code>は副詞で $2+3+4+5+6$ と作用する。 副詞は動詞をパワーアップする。<code>/</code>は作用素</p>
Π	<pre> */ 2 3 4 5 6 720 </pre>	<p>$2*3*4*5*6$</p>
外積	<pre> */~>:i.9 1 2 3 4 5 6 7 8 9 2 4 6 8 10 12 14 16 18 3 6 9 12 15 18 21 24 27 4 8 12 16 20 24 28 32 36 5 10 15 20 25 30 35 40 45 6 12 18 24 30 36 42 48 54 7 14 21 28 35 42 49 56 63 8 16 24 32 40 48 56 64 72 9 18 27 36 45 54 63 72 81 </pre>	<p>九九の表 <code>i. i:</code> は指定した数の数列を打ち出す。 <code>i. 3</code> → 0 1 2 <code>i: 3</code> → _3 _2 _1 0 1 2 3</p>

<p>GCD</p>	<pre> 36j2 +. 24j3 12 36j2 +. 24j3 1j2 </pre>	<p>j は複素数 共役複素数もサポート</p>
<p>LCM</p>	<pre> 36 *. 24 72 36j2 *. 24j3 234j_312 </pre>	
<p>一様乱数</p>	<pre> 5?20 6 7 10 11 3 </pre>	<p>?. は同じ乱数を打ち出す</p>
<p>行列式の値</p>	<pre> a=. 3 3 \$ 9?12 1 7 5 4 6 10 2 3 8 -/. * a _66 </pre>	
<p>多項式</p>	<pre> x^4 - 10x^3 + 36x^2 - 58x + 35 = 0 p. 35 _58 36 _10 1 +-+-----+ 1 4.41421 2j1 2j_1 1.58579 +-+-----+ 4.41421 ± 2j1 1.58579 x^5 + 1 = 0 p. 1 0 0 0 0 1 0.809017j0.587785 0.809017j_0.587785 _0.309017j_0.951057 _1 </pre>	<p>最初のボックスは反復回数 p. への入力は定数項、低次の項から順に高次の項へ。 該当がなければ 0 回帰係数につけると推計値が求められる。</p> <pre> 1 3 2 &p. i.5 1 6 15 28 45 </pre>

微分	$f = x \sin x$ $f' = \sin x + x \cos x$ <pre>f=.] * sin@] f a 0.0397339 0.155767 0.338785 0.573885 f''0 (D.1) a=. 0.2 0.4 0.6 0.8 0.394683 0.757843 1.05984 1.27472</pre>	積分は PACKAGE の math に SCRIPT で入っている。 D.1 は 1 階微分
----	---	---

B.3 小技

オリジンは 0 数列は 0 1 2 3 4 5 の様に 0 から始まる。最近の言語では一般的であるが、クラシック言語になじんだ人には違和感があるらしい。

```
i.5
0 1 2 3 4
```

```
i:5
_5 _4 _3 _2 _1 0 1 2 3 4 5
```

```
i. 3 3
0 1 2
3 4 5
6 7 8
```

取り出し(0) 最初と最後尾。縦方向はランク(”1)で(以下同じ)

```
{. 0 1 2 3 4 5
0
```

```
{. i. 3 3
0 1 2
```

```
{: 0 1 2 3 4 5
5
```

取り出し(1) 箇所指定、まとめて

```
0 2 { 0 1 2 3 4 5
0 2
```

```

    0{i.3 3
0 1 2

    0{"1 i.3 3
0 3 6
落とし(0) 最初と最後尾
    }. i. 3 3
3 4 5
6 7 8

    }. 0 1 2 3 4 5
1 2 3 4 5

    }:"1 i. 3 3
0 1
3 4
6 7
落とし(1) 箇所指定、まとめて
    2 }. i. 3 3
6 7 8

    2}.("1) i.3 3
2
5
8
置き換え アmend (}). かなり複雑である。
    100 (2 3)} i. 5
0 1 100 100 4

    100 (0 2;2 3)} i. 4 4 NB. point
0 1 100 3
4 5 6 7
8 9 10 100
12 13 14 15

    100 (<0 2;1 3)} i. 4 4 NB.raw 0 2 column 1 3
0 100 2 100
4 5 6 7

```

```

      8 100 10 100
      12 13 14 15
連結  両項の, ; ,. ,: はベクトルやマトリクスを連結する。ベクトル、マトリ
      クスは同一サイズであること
      1 2 3,4 5 6  NB. Append
      1 2 3 4 5 6

      1,4 5 6
      1 4 5 6

      1 2 3;1 2 3  NB. Link
+-----+-----+
|1 2 3|1 2 3|
+-----+-----+

      1 2 3,: 4 5 6  NB. Laminate
      1 2 3
      4 5 6

      (i. 3 3),. 4 5 6  NB. Stich
      0 1 2 4
      3 4 5 5
      6 7 8 6
型と変形  # $は単項ではサイズを取り、$の両項は指定サイズに変形する。#の
      両項は右引数で指定した数でのコピー
      # i. 3 3  NB. # tally
      3
      $ i. 3 3  NB. $ shape
      3 3

      2 5 $ i.10
      0 1 2 3 4
      5 6 7 8 9

      ? 3 3 $ 20  NB. random numbers
      9 8 15
      17 4 0
      1 2 17

```

```
4# 1 2 NB. copy
1 1 1 1 2 2 2 2
```

```
4# <1 2 NB. boxed copy
+---+---+---+---+
|1 2|1 2|1 2|1 2|
+---+---+---+---+
```

ほぐすく , ; の単項はほぐし

```
,i.3 3
0 1 2 3 4 5 6 7 8
```

```
; 4# <1 2 NB. boxed-copy and raze
1 2 1 2 1 2 1 2
```

B.4 BOX

Jでサポートされた新しい形式。高次元の複雑なマトリクスのランクを気にしないで計算できる。

<p><はボックスを作る。>は開く。 ランクにより、組上がるボックスの形が異なる。</p>	<pre> i. 3 3 3 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 <はボックスを作り、>は開く </pre>	<p>ランク ("0 "1 "2) はそれぞれスカラ、ベクトル、マトリクスに作用する。</p> <pre> <"0 i. 3 3 +---+---+ 0 1 2 +---+---+ 3 4 5 +---+---+ 6 7 8 +---+---+ <"1 i. 3 3 3 +-----+-----+-----+ 0 1 2 3 4 5 6 7 8 +-----+-----+-----+ 9 10 11 12 13 14 15 16 17 +-----+-----+-----+ 18 19 20 21 22 23 24 25 26 +-----+-----+-----+ <"2 i. 3 3 3 +-----+-----+-----+ 0 1 2 9 10 11 18 19 20 3 4 5 12 13 14 21 22 23 6 7 8 15 16 17 24 25 26 +-----+-----+-----+ </pre>
<p>ボックスの中に作用させるときは L:0 を用いる。</p>	<p>多重ボックスでは L:1 を用いることもある。</p>	<pre> +/(L:0) <"1 i. 3 3 3 +---+---+ 3 12 21 +---+---+ 30 39 48 +---+---+ 57 66 75 +---+---+ +/"1 (L:0) <"2 i. 3 3 3 +-----+-----+-----+ </pre>

文字列と併用 もできる。	'a b' は文字列。	<pre> 'a b';1 2 3 +---+-----+ a b 1 2 3 +---+-----+ </pre>
-----------------	-------------	--

ボックスを駆使しているとランクが複雑になる。\$で確認して、場合によってはランクを変更する。

```

    $ i. 3 3
    3 3

```

同じように見えても 1 3 3 など 1 ランク上のものが出ることがある。計算できたとしても所期の結果とは異なる。

prefix,suffix prefix(<\) や suffix(<\. は時系列では大いに活躍する。

```

    <\ i.5      NB. prefix
    +---+-----+-----+-----+
    |0|0 1|0 1 2|0 1 2 3|0 1 2 3 4|
    +---+-----+-----+-----+
    <\. i.5     NB. suffix
    +-----+-----+-----+-----+
    |0 1 2 3 4|1 2 3 4|2 3 4|3 4|4|
    +-----+-----+-----+-----+

```

```

    +/ L:0 <\ i.10
    +---+---+---+---+---+---+---+---+
    |0|1|3|6|10|15|21|28|36|45|
    +---+---+---+---+---+---+---+---+

```

```

    ; +/ L:0 <\ i.10  NB.raze is open
    0 1 3 6 10 15 21 28 36 45

```

```

    移動平均    3<\i.10    NB. 3 項移動平均
    +-----+-----+-----+-----+-----+-----+-----+
    |0 1 2|1 2 3|2 3 4|3 4 5|4 5 6|5 6 7|6 7 8|7 8 9|
    +-----+-----+-----+-----+-----+-----+-----+

```

```

    ;(+/%#) (L:0) 3<\i.10

```


1 2 3 4 5 6 7 8

B.5 SCRIPT を書く

部品から作る 短い動詞でも一々書くのは面倒なので、スクリプトで保存しておく。本格的なプログラムだと、数千行を越える。APL や J は最初から小さいプログラムを部品として創り、組み合わせて動かす様になっていたの、構造、オブジェクト、インヘリタンスなどの業界用語の理解に時間をとられたり、惑わされることはない。信頼できる本や J のフォーラムに書かれている達人が作成したスクリプトも、著作権が問題なければ自由に組み込める。(非商業用) 部品は簡潔に作り、よくテストをしておく。一個の腐った蜜柑、特に数学関数の動詞の不具合は箱全部をだめにする。

明示型定義と関数型定義 J には関数型定義が最初からサポートされた。^{*6}

数学	明示型	関数型
$\frac{\sum n}{n}$	<pre>mean0=: 3 : 0 (+ / y) % # y) mean0 i.10 4.5</pre>	<pre>mean1 =: + / % # mean1 i.10 4.5</pre>

3 : 0 は単項の動詞で、右にのみ引数 (y) をとる定義法です。mean =: verb_define とする方法もある。両項は左引数 (x) と右引数 (y) をとり、4 : 0 で定義する。^{*7*8}

関数型は数式により近く簡潔に定義する方法である。^{*9}

明示型 (Explicit definition) はクラシックな方法であるが、引数をはっきり示し、後ろから前の単一ルールが適用されるので、読みやすく、メンテナンスも楽である。関数型はエレガントなので、数式の定義など一度定義してしまえば余り変更しないものに用いる方がよい。明示型と関数型は相互に混在できる。

J 言語メモ

^{*6} アイバーソンは関数型定義を作りたいからもう仕様が固まっていた APL2 でなく、新しい J を作ったと思われる節がある。その後 APL2 でも関数型定義はサポートされた。

^{*7} dyad_define でもよい

^{*8} 3 個の場合は x や y に 2 個入れる。

^{*9} C などでは手続き型のプログラムのブロックを関数ということがある。

y	右引数
+/	Σ
%	÷
#	n を数える動詞、 <i>tally</i> と古風な英国英語で呼ばれている
)	動詞の定義の最終行でここで閉じる (必須)

i.10 数列生成 0 1 2 3 4 5 6 7 8 9

明示型は引数をはっきり書く

実行は後ろから前へ 明示型の統一ルール。関数型は複雑になる。

回帰係数を求める。定数項	左引数は x , 右引数は y で
を求める場合は X の前に 1	小文字である。数学とは一
を並べる。	致しない。 ^{*10}

$\frac{y}{X}$	reg=: 4 : 'x % . 1, . y'
---------------	--------------------------

インタプリッタを活用 まずは ijx でサンプルデータを作るか読み込む。

次にこのデータを実際に ijs の画面で動かしながらスクリプトの部分を作ってみる。巧く動いたら Shift + ↓ で反転させてコピーして ijs の画面に張り付ける。データ名を xy に直してスクリプトを組んでいく。

経過を追って動作が確認されているのでデバックという手順は不要。

動詞の中に動詞を組み込む。 永い一つのスクリプトを作るより、スクリプトは小さな動作の確認できる単位で作って、最後に全体を通すスクリプトで纏める。

```
mean=: +/ % #
```

```
dev=: ] - mean
```

NB. $x - \bar{x}$

```
dev i.5
```

```
_2 _1 0 1 2
```

名前が重複したときは後読み優先 警告なしに書き換えるので注意が必要。

names ' ' で使っている名前の一覧が出てくる。

B.6 関数型定義

J で新しく用いられた定義方法。^{*11}

動詞を中心に定義する \sum_n

```
mean=: +/ % #
```

```
a=:.>:i.10
```

^{*11} APL も追従した

```
1 2 3 4 5 6 7 8 9 10
```

```
mean a
```

```
5.5
```

フォーク 単項のフォーク、`a=:>i.10` に先ず両端の2つの動詞が働いて $\Sigma(+/a)$ と $n(\#a)$ を実行し、次に `%` を実行する。(後ろなら前へではない。)この形が食器や農機具のフォークの形と同じであることならフォークと呼ばれる。

```

      f      %
 /      \   /      \
g      g +/      #
|      |   |      |
y      y y      y

```

両項の場合は次のようになる。

```

      f
 /      \
g      g
|      |
x      y

```

引数 右引数は `[`, 左引数は `]` (単項では書かなくともよい)

接続詞 `&` `@` 数字と動詞を接続する場合は`&`, 動詞と動詞を連ねる場合は`@`と`&`接続詞で連結された動詞は一つの動詞として働く。

単回帰、重回帰の関数 `reg` を定義する。Matrix-decide(`%.`) の強力な機能でシンプルに定義できる。

```

reg=: [ % . 1& ,.@] NB. dyad fork

] a=: 5?.20          NB. random number fixed seed
6 3 19 15 10

] b=: ?.5 3 $ 50    NB. shape
46 5 29
2 4 39
10 7 10
44 46 28
13 18 1

```

```

a reg b          NB. multiple regression
13.0488 0.000515423 0.150922 _0.227821
f = 13.0488 + 0.000515423x0 + 0.150922x1 - 0.227821x2

```

\hat{Y} を求める。

```
a, . (1, . b) +/ . * (a reg b) NB. inner product
```

```
6 7.22028
```

```
3 4.76847
```

```
19 11.8322
```

```
15 13.6349
```

```
10 15.5442
```

グラフへ進む

```
load 'plot'
```

NB. same require 'plot'

```
'key y hatY' plot |: a, . (1, . b) +/ . * (a reg b)
```

```
pd 'eps temp/appendix_reg0' NB. file is eps type
```

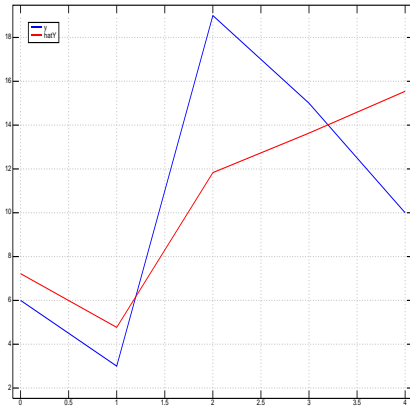


図 B.1 重相関の推計値

関数型での反復計算 $\hat{\cdot}(10)$ $\hat{\cdot}(>:i.10)$ で反復計算を行う。ループは不要

```
(1 + r) * (1 + r) * ... (1 + r)
```

```
fr=: ] * >:@[ NB. 金利計算
```

```
0.1 0.2 0.3 0.4 0.6 fr ^:(>:i.10) 1
```

```

1.1      1.2      1.3      1.4      1.6
1.21     1.44     1.69     1.96     2.56
1.331    1.728    2.197    2.744    4.096
1.4641   2.0736   2.8561   3.8416   6.5536
1.61051  2.48832  3.71293  5.37824  10.4858
1.77156  2.98598  4.82681  7.52954  16.7772
1.94872  3.58318  6.27485  10.5414  26.8435
2.14359  4.29982  8.15731  14.7579  42.9497
2.35795  5.15978  10.6045  20.661   68.7195
2.59374  6.19174  13.7858  28.9255  109.951

```

関数型と明示型の共存 明示型の中で関数型を使うことができる。無理に全部を関数型で作る必要はない。

B.7 plot

ビジネス、サイエンスグラフの使用法

最初にグラフ用のライブラリを読み込む。ijx 場面で次のようにタイプするか ijs の最初の方に書いていく

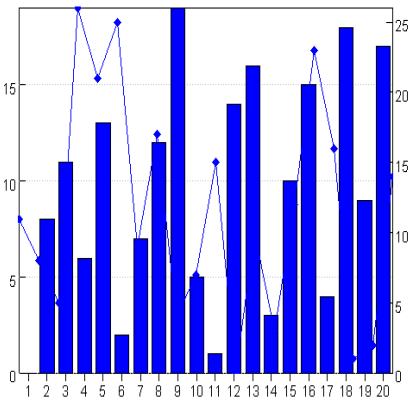
```
require 'plot numeric trig' NB. 3点セット
```

trig は三角関数、numeric は便利なツール類が入っているので同時に読み込んでおく。

```
plot sin i.20
```

```
'bar'plot 20?30
```

```
'line,stick' plot cos steps 0 10 100 NB. 0<-->10 with 100 steps
```



2軸(Y軸)のビジネスグラフ。最初の reset (または new) と最後の show は必須

```
plot_test=: 3 : 0
pd 'reset'
pd 'type bar'
pd 20 ?20
pd 'y2axis'
pd 'type line ,marker'
pd 20?30
pd 'show'
)
```

2画面のビジネスグラフを本格的に定義する。

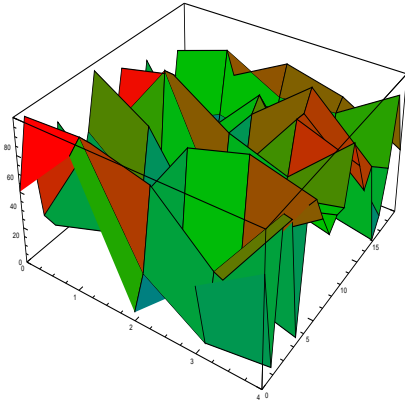


図 B.2 cos sin

```
'surface' plot 5 20 $ 100?100
pd 'eps temp_append_sin1.eps'
```

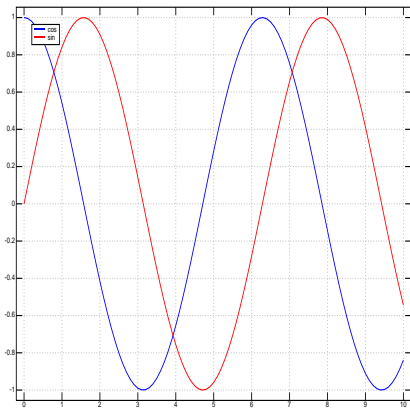


図 B.3 cos sin

```
plot_test =:3 : 0
NB. -----
pd 'reset'
pd 'sub 0 0 1000 _40'
pd 'sub 2 1 '
NB. -----0 0-----
pd 'new'
pd 'keypos ib'
pd 'keystyle lho'
pd 'key cos sine'
pd 'color navy,green'
pd a; sin a=. steps 0 10 100
NB. -----0 1-----
pd 'new'
pd 'type surface'
pd 5 20 $ 100$100
pd 'show'
)
```

第 C 章

データの入出力

C.1 CSV ファイル

最近配布されているファイルは EXCEL か CSV 形式が大半である。EXCEL は窓や MAC との連携はよくできているが、他のプラットフォームでは扱えない。最近では MS-OFFICE のクローンも増えてきた。

CSV(Camma Separated Values) とは、カンマで区切られたファイルで、永い伝統を持ったファイル形式である。筆者は 20 万件のデータ程度は CSV 形式で取り扱っており、手軽で便利である。

C.1.1 EXCEL から CSV への出力

著者はファイルは EXCEL で管理し、数値部分のみ EXCEL から CSV に出力している。CSV への出力は、ファイルをセーブするときに、[名前を付けて保存] を指定し、ダイアログボックスで保存するファイル形式を CSV と指定すればよく、元のファイルは変更されない。

ここでの約束は

カンマ カンマはつけない。(データがカンマで分離されることが多い) 出力する数値の範囲を先にマウスなどで選んで、書式 → セル → 数値でカンマのチェックを外しておく。

空白 空白は 0 や 99999 など埋めておく(その行のみ左詰めになることがある。)

下部の余白に付くゴミ csv ファイルの最下行の下にゴミの行ができたならばノートパッドなどで削除しておく(そのままだと空白のボックスが入ることがある)

アクセスなどのデータベースも CSV 出力機能を持っている。ファイル出力で csv を選べばよい。

C.1.2 csv ファイルの読み込み

データの読み込みは csv が便利である。20 万件程度のファイルはメモリが確保できれば一度に読み込める。

最初に ファイルを取り扱うライブラリを読み込む。

```
require 'files csv'
```

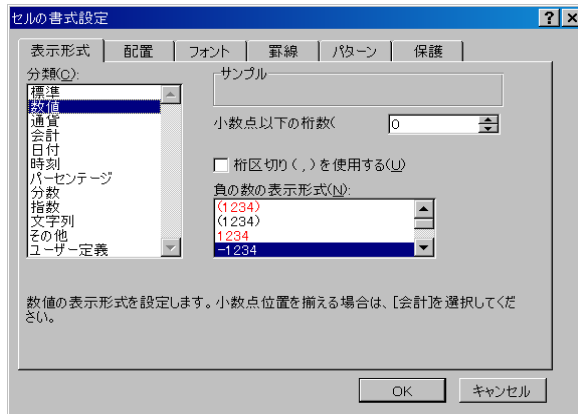


図 C.1 カンマを外す

読み込み パスを指定する。

```
DAT=: ".@> readcsv 'user/classes/numeric_recipe/data/csv/foo.csv'
```

DIRCSV などとパスを定義しておくとも簡易になる。

```
DIRCSV=: jpath '~user/classes/numeric_recipe/data/csv/'
```

jpath は j602-user の user を探してくれる。チルド (~) を付ける。

数値化 csv ファイルはテキストファイルで box に入っている。

```
DAT=. ".@> readcsv DIRCSV, 'foo.csv'
```

CSV ファイルの中身は数字の形をしていても文字である。J から読み込んだときは一つずつボックスに入っている。

数値の部分のみを CSV ファイルにしてある場合は上記のように ".@> で数値化して、ボックスを開いておく。

マトリクスの構成如何で ".@> で開いたときに崩れることがある。

文字列と混在のファイルでは文字列を落としてから数値化を行う。

確認 以上で名詞 DAT に数値データが入った。先ず DAT と打ってデータがざっと出れば次に +/ DAT として、縦に合計ができれば数値化もできている。

文字列も入った BOX ボックスの混合配列はそのままでは数値化できない。文字列の部分を取り出し数値化を行う。

C.1.3 CSV ファイルへの出力

require 'files csv' が必要

```
(i.5 5) writecsv '/temp/test.csv'
```

```
120
```

複雑な計算の場合は予め済ませておき、 writecsv の左引数は名詞 (a=.) を用いる。

J602 はパスが複雑になったので出力先は各自の *temp* 等を用いた方がよい

C.2 EXCEL などとの OLE

OLE(Object Linking and Embedding) とは EXCEL と他のアプリケーションを生で繋いでデータやプログラムなどをやりとりする機能である。

EXCEL 以外にも無償のスプレッドシートが幾つか公開されてきている。

最近 J のライブラリの *tara* が公開された。J のホームページからダウンロードでき、EXCEL 以外にオープンオルグの CALC でも利用できる。

C.2.1 tara

tara は ADDON で利用できる EXCEL や Openolg Calc の *bif8* 形式のファイルを直接読み込んだり書き込むことができる。(EXCEL 本体などは用いない。EXCEL2003 まで。EXCEL2007 では出力形式を 2003 様式にする。)

`tara.ijs require jpath '~addons/tables/tara/tara.ijs'` を読み込む。

```
read DAT=. readexcel jpath '~user/classes/numeric_recipe/data/xls/shirasago.xls
```

```
  b=. 'Sheet1' readexcelstring jpath '~addons/tables/tara/test.xls' NB. worksheet by r
ボックスを外す  tara で読み込んだファイルは数値になっている。>でオープン
する。
```

```
write 最初に conew で挨拶しておく。書き込みと save の 2 ステップに分かれ
る。書き込みアドレスの先頭を (0 0) 等と指定する。
```

書き込みはファイルを壊すことがあるので新しいファイル名を指定するか、複製したファイルを用いること。

```
bi=: '' conew 'biffbook' NB. create an workbook
writenumber__bi 0 0 ; i.6000 100
```

```
writestring__bi 6000 0 ; <<"(1) 6000 100 7$'ABCDEFGHJKLMNOPQRSTUVWXYZ'
```

```
save__bi jpath '~temp/tara9.xls'
```

詳細は Studio|Labs に *tara* が入っている

C.3 ADDONS の入手とインストール

アドオンで固有値の計算に用いる LAPACK と、フーリエ解析用のフーリエの使用法である。

C.3.1 LAPACK

LAPACK は FORTRAN の汎用数値計算パッケージである。これを J から利用できる。

インターネットに繋がった環境なら RUN →PackageManager で JSoftware と繋いで一覧から必要なものをダウンロードする。

*1

LAB は演習とマニュアルをかねている。

Ctrl + J で進んでゆく。

Coffee Brake

エディタ エディタと言われて編集者と思った人と、何ですかと思った人は少し続きを読んでください。極少数の人達の共通語彙ではエディタとはプログラムや別に整形させる文を書く専用のワープロソフト（ツール）で、競技用の自転車かモーターバイクの様に小柄で精悍な余計なことを一切しないシンプルな道具です。手元に無い人はウィンドウズのアクセサリに入っているノートパッドを使って下さい。

*2

ファイル名 ファイル名は英文で統一し、漢字や 2 バイトコードは用いていない。J はまだ漢字名のファイルを読まない。多くのプログラムを組む人にとっては奇異とは思わないが、日常で漢字ファイル名を使っているアプリケーションユーザーには負担になることもある。^{*3}

ファイルは data1.csv の様に .xxxx とピリオッドの後ろに 3 文字程度でファイルの種類を表している部分がある。この部分は決して書きかえないこと。

*1 win, Linux, そして darwin は MAC 用

*2 MS-Word は薦めません。

*3 ユニコード化が進んでいるので、早晚解決すると思うが

第 D 章

数学付録

D.1 行列の除算

J は行の数と同じなら列の数の子となる行列でも行列除算ができる。APL のドミノ関数から受け継いだ `matrix-inverse(%)` という強力な機能が備わっている。

J の辞書の % の項には

$\frac{A}{B}$ は $B^{-1}A$ であると記載されている。

拡大クラメル法はこの行列除算である。

```
cr=: %."1
```

1. example

```
a=. ? 3 4 $ 100
46 55 79 52
54 39 60 57
60 94 46 78
```

2. 拡大クラメル法

```
cr=. %."1
cr a
1 0 0 0.955809
0 1 0 0.257753
0 0 1 -0.0777672
```

3. $\frac{A}{B}$

$$\frac{\begin{bmatrix} 46 & 55 & 79 & 52 \\ 54 & 39 & 60 & 57 \\ 60 & 94 & 46 & 78 \end{bmatrix}}{\begin{bmatrix} 46 & 55 & 79 \\ 54 & 39 & 60 \\ 60 & 94 & 46 \end{bmatrix}} = \begin{bmatrix} 1 & 0 & 0 & 0.955809 \\ 0 & 1 & 0 & 0.257753 \\ 0 & 0 & 1 & -0.0777672 \end{bmatrix}$$

4. $B^{-1}A$ on J

```
(%.}: "1 a) +/ . * a
      1 0 3.46945e_16 0.955809
_1.11022e_16 1 _1.11022e_16 0.257753
_1.11022e_16 0          1 _0.0777672
```

5. $B^{-1}A$

$$\begin{bmatrix} -0.0382276 & 0.0486641 & 0.00217677 \\ 0.0110926 & -0.0260814 & 0.014969 \\ 0.0271947 & -0.0101781 & -0.0116889 \end{bmatrix} \begin{bmatrix} 46 & 55 & 79 & 52 \\ 54 & 39 & 60 & 57 \\ 60 & 94 & 46 & 78 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0.955809 \\ 0 & 1 & 0 & 0.257753 \\ 0 & 0 & 1 & -0.0777672 \end{bmatrix}$$

D.2 逆行列

$AA^{-1} = I$ となる行列を逆行列と言う。 I は単位行列である。

正方の行列ならば右に同じサイズの単位行列を並べて、次のように掃き出すと逆行列が求められる。

$$\begin{array}{ccc|ccc} 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 3 & 0 & 0 & 1 \end{array}$$

配列計算言語は行列除算機能を持っている。Jでは *Matrix inverse Matrix divide* (%.) である。%. で一発で縦長の行列の逆行列が作成できるので回帰分析の計算が見通しよく簡単にできる。

実際の例でデータから分散共分散行列を求めるときは正則な行列になるが、本書のような正規方程式を経ないで生データからいきなり回帰を行うような手法を採るとき、データが正方になることはほとんど期待できないが、Jは縦長の行列からダイレクトに逆行列を求めることができる。

J言語ノート

1. 逆行列を求める

D0=. 3 3 \$ 2 1 2 1 3 1 1 1 3 D0 2 1 2 1 3 1 1 1 3	Jで逆行列を求める %. D0 0.8 _0.1 _0.5 _0.2 0.4 0 _0.2 _0.1 0.5	ガウス法による掃き出し。 gaussj D0,./~i.3 1 0 0 0.8 _0.1 _0.5 0 1 0 _0.2 0.4 0 0 0 1 _0.2 _0.1 0.5
---	--	--

2. 行列式

行列式が0の場合はランクが不足して逆行列が作れない。行列式の値はサラスの法則が有名であるが手計算では3×3程度が計算の限界である。Jの行列式を求めるプリミティブ-/ . *で求められる。

```
-/ . * d
10
```

3. 単位行列

単位行列を作るのは何種類かの方法があるが、次の方法が容易である。

```
=/~ i.3
1 0 0
0 1 0
0 0 1
```

D.3 行列計算の手順

加算 A	B	A + B
a=. :2 3 \$>: i.6	b=. : 2 3 \$ 7+ i.6	a+b
1 4	7 10	8 14
2 5	8 11	10 16
3 6	9 12	12 18

内積	C	AC
	$c = \begin{pmatrix} 2 & 2 \\ 7 & 9 \\ 8 & 10 \end{pmatrix}$	$a + \begin{pmatrix} 7 & 1.4 \\ 39 & 49 \\ 54 & 68 \\ 69 & 87 \end{pmatrix} * c$

$$\left\{ \begin{array}{ll} (1 \times 7) + (4 \times 8) & (1 \times 9) + (4 \times 10) \\ (2 \times 7) + (5 \times 8) & (2 \times 9) + (5 \times 10) \\ (3 \times 7) + (6 \times 8) & (3 \times 9) + (6 \times 10) \end{array} \right\}$$

D.4 モデルの同時方程式を解く

外生変数がベクトルの場合

連立方程式は経済システムの要素間の相互依存関係を分析するのに有効である。

例えば次のような線形モデルの連立方程式を解く場合を考える。内生変数は左に寄せて連立モデルに組み込む。

$$\begin{aligned} C_t &= 10 + 0.4Y_t + 0.5C_{t-1} && \rightarrow C_t - 0.4Y_t = 10 + 0.5C_{t-1} \\ I_t &= 5 + 0.2Y_t && \rightarrow I_t - 0.2Y_t = 5 \\ Y_t &= C_t + I_t + G_t && \rightarrow Y_t - C_t - I_t = G_t \end{aligned}$$

$$\begin{bmatrix} C & I & Y \\ 1 & 0 & -0.4 \\ 0 & 1 & -0.2 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} C \\ I \\ Y \end{bmatrix} = \begin{bmatrix} 10 + 0.5C_{t-1} \\ 5 \\ G \end{bmatrix}$$

$C_{t-1} = 10, G = 10$ と数値を与えれば、線形連立一次方程式は逆行列[※]を用いて簡単に解を得られる。ここではクラメル法を用いた。

$$DN101 = \begin{pmatrix} 3 & 3 & 1 & 0 & -0.4 & 0 & 1 & -0.2 & -1 & -1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & -0.4 \\ 0 & 1 & -0.2 \end{pmatrix}$$

```

_1 _1    1

      DN101, . 15 5 10

1  0  _0.4 15 NB. 10 + 0.5*10
0  1  _0.2  5
_1 _1    1 10

cr DN101, . 15 5 10 NB. cramer method
      1 8.88178e_16 1.33227e_15 45
_1.94289e_15      1 8.32667e_16 20
_6.21725e_15 2.22045e_15      1 75

```

*1

$[CIY]$ を線形変換したものが $\begin{bmatrix} 10 + 0.5C_{t-1} & 5 & G \end{bmatrix}$ であり、 $[CIY]$ を逆行列を用いて割り戻して求めるプロセスである。

*1 数値計算では0ならず微少な値(ゴミ)がでることがよくある

索引

- t 検定, 248
- K.E.Iverson, 165
- 赤池, 184, 214
- インパルス, 210–214
- ウェーブレット基底, 151
- AIC, 184, 185, 200, 227, 248
- ACF, 174, 177, 179
- ESRI, 139, 143, 145, 146
- LAPACK, 140, 145, 286
- 外生変数, 215–217, 219, 225, 235, 239
- 確率過程, 173
- 過剰識別, 227
- カスケード分解, 147
- カルマンフィルタ, 243
- 間接最少自乗法, 224
- 観測方程式, 249
- ギアリー, 231, 232
- 規格化, 152
- 逆行列, 146, 179, 196, 218–220, 222, 251, 288–291
- 行列式, 289
- クラメル法, 179, 180, 290
- 構造型方程式, 215–217, 225
- 固有値, 139, 140, 145
- 固有ベクトル, 139, 140, 142
- サラスの法則, 289
- 産業連関表, 165
- 散布図, 142
- 識別, 222, 224, 227, 228, 236, 237
- 時系列解析, 173, 201
- 自己回帰, 179, 187, 188, 194, 195
- 重回帰, 233
- 状態方程式, 210, 211, 248
- 情報量規準, 200
- スラック変数, 162
- 線形計画, 161
- 線形同時方程式, 215, 229
- 相関行列, 139, 140
- 相互相関係数, 194, 210
- 双対問題, 164
- 多重解像度分解, 146
- 多変量自己回帰, 179, 181, 190, 195, 198–200, 203
- 単位行列, 162, 233, 288, 289
- 適正識別, 227
- ドビッシ, 151
- 内生変数, 216, 217, 222, 225, 227, 228, 235, 239, 290
- 2 段階最少自乗法, 225
- PACF, 177
- ハール, 149
- 標準化, 139–141
- 分散共分散行列, 199, 208, 244, 246, 288
- 偏自己相関, 177
- 誘導型方程式, 216, 217
- Yule-Walker, 179, 195
- レオンチェフ, 165
- レオンティエフ逆行列, 167
- レビンソン, 179, 181, 196, 199, 214