

# 社会システム分析のための統合化プログラム

福井正康，田口賢士

福山平成大学経営学部経営情報学科

## 概要

我々は、社会システム分析に用いられる様々な手法を统一的に扱うソフトウェアを作成した。分析用のデータは、複数枚のシートの形式をしており、表形式、複数の表形式、グラフ構造等様々な構造に対処可能である。その例として、我々はデータの取り扱い方の全く異なる数種の分析を統合的にまとめ、1つのパッケージとした。本論文では全体的なシステムの構成と各分析についての具体的なデータ構造をまとめた。

## キーワード

ソフトウェア，社会システム分析，統計，OR

## 1章 はじめに

我々は社会システム分析を、「社会現象の因果性、構造的性、法則性を見出すための分析」と位置づけているが、この中で用いられる分析手法は、統計学や OR はもちろん、定性的な量を扱う手法や分析の方法論に至るまで、その範囲は極めて広い。分析にはコンピュータを用いることが殆どであるので、ソフトウェアの整備状況は作業効率に大きく作用する。この中で特に統計学については、統合的に多くの検定や分析を扱う優れたソフトウェアが多く、様々な分析を効率良く行うことが出来る。これに対して、OR の分野では各分析用の単体のソフトウェアとしてはよく出来たものは多いが、多数の分析を統合化したソフトウェアは殆どない。これはその他の分野についても同じで、分析手法やデータ構造の大きな違いからやむを得ないことも知れない。しかし、社会システム分析のように多くの分析手法を実行し、その中から意味を抽出するような方法においては、1つ1つのソフトウェアの操作の習得もその価格も無視出来ない問題となる。特に、教育機関においてこれら多くの分析を教育する場合、予算的にも、学生の実習時間にも制約があり、コンピュータを用いた実践的教育を十分行えるとは言い難い。

そこで、今後の教育と研究への応用のために、我々は統計学、OR、その他の分析手法の多くを統合化したソフトウェアの作成を開始した。我々はこれまで一度このような統合化プログラムを構想したことがあったが<sup>1)</sup>、データ形式として統計ソフトウェアに見られるような1枚のデータシート以上のものは考えていなかった。今回再度この問題を扱うにあたり、以下のことを原則とした。

1. 出来るだけ多くの社会システム分析の処理を統合的に扱えること
2. 教育的効果を重視すること
3. エディタや出力画面等、共通な部分は出来るだけ共用すること
4. 必要最低限のルールを守れば、各分析独立にプログラミング出来ること
5. 既存のワープロや表計算ソフトとデータの互換性があること
6. 常にその時代のパソコンの代表的なプラットフォーム上で動作すること
7. ソースコードを含めたフリーソフトとして公開すること

最初の項目の「分析処理を統合的に扱う」もっとも簡単なデータ構造として、我々はそれぞれ行数列数の異なる複数枚のシート形式を採用した。これは、意外と広い適用範囲をもっている。グラフ構造などを、行列形式で表すことにより、殆どの分析はこの枠内に入る。但し、データの構造を汎用的にしたことにより、入力に多少知識が要求されるようになったが、もし必要とあれば、各分析毎に入力用のシェルを用意することで入力を簡単化出来る。もちろんこれは他の分析に影響しないように独立に作成することを必要とする。

次に、このソフトウェアは主として教育現場での活用が考えられており、作成に当たって教育効果を考える必要がある。即ち、教育には分析結果だけでなくその過程も重要な意味を持つ

ている。そのため、アルゴリズムの確立された重要な分析手法（例えば線形計画法のシンプレックス法等）では計算の実行過程を表示する機能を付けることを考えたい。また、シミュレーション等の実験ツールとしても利用出来れば、より高い教育効果が期待出来る。

我々は現在のパソコンの代表的なプラットフォームとして、Windows 95 を想定している。そのため開発言語として、Visual Basic を選択した。これは Microsoft 社が推進している開発環境で、当分の間継続するであろうし、将来的にも資産の継承はある程度保証されるであろう。また、Windows 上だと既存のソフトウェアとのデータの互換性はクリップボードを経由することによって容易に実現される。

このソフトウェアは、ある程度実際の講義で使用した段階で公開し、常に維持管理を続けながら、フリーソフトウェアとしてインターネット上で配布する予定である。大学で学んだ社会システム分析の考え方を社会でも役立てて欲しいからである。また Visual BASIC は習得が簡単であるので、比較的容易に誰でもソースコードを変更することが出来、必要な処理を追加出来る。新しい分析の追加や改善箇所は、著者まで知らせていただければプログラムに反映させる。

本論文では、最初にシステムの概要について説明し、次に個々の分析についてエディタへのデータ記入形式を中心に話を進め、それらを共通的に保存するためのファイル形式について述べる。ここまでは利用者側からみた仕様であるが、開発には複数の協力者が必要となる。そこで、本論文がプログラム開発者向けの資料となるように、内部データや内部仕様についても補遺で言及する。プログラミングに際して、入力部分に当たるエディタの内部的なデータ形式、結果表示に用いられるフォームの利用方法、共通的に利用される関数やプロシージャの仕様が分かれば、プログラムは他の分析と独立に作成出来る。今後シリーズとして各分析の詳細を発表して行く予定であるが、その際内部資料の差分も付加する。それと同時にユーザーズマニュアルと開発者向け資料の改訂を進める予定である。

## 2章 システムの概要

システムはエディタ、分析プログラム、結果表示フォームの3つからなり、エディタと結果表示の部分が外部のファイルやプログラムとデータのやり取りを行う部分となる。全体的な構成は、図 2-1 に示すが、この構造は従来の統計パッケージに類似している。ただ一つ異なるところは、通常、データから分析へ、分析から結果表示へという形をとるのに対して、このソフトでは分析結果から逆にデータを作る場合もある、というところである。例えば AHP 等はその代表で、分析部分で一対比較を行い結果をデータに書き込むようにする。

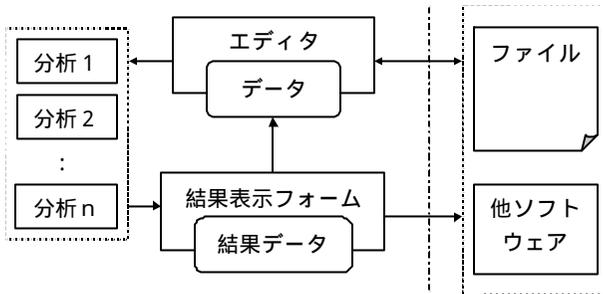


図 2.1 ソフトウェアの構造

様々な分析を取り入れるために、データ構造は最も重要な要素である。データとしては、シート形式のもの、複数のシート形式のもの、グラフ構造のものを考え、ページ毎に行数列数が異なる複数のシート構造

を採用した。これによって殆どすべての分析を組み込むことが出来る。例は次章で示す。メモリの節約とスピード化から、以前のプログラム<sup>1)</sup>ではデータにリスト構造を用い、メモリのオーバーフローに対処するために、ワークファイルを利用していたが、ここではプログラミングの簡単化のためにデータは3次元配列とし、Windows 上のプログラムではメモリ管理を OS が行ってくれるので、MS-DOS の時代のようなワークファイルを利用することはしなかった。これによってプログラミングが大幅に簡単になった。以後実際の分析処理に従って、プログラムの概略を説明する。

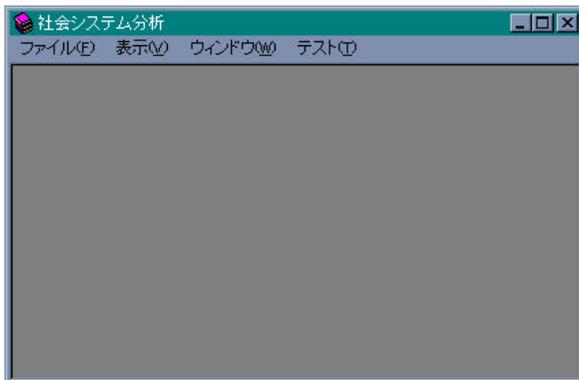


図 2.2 社会システム分析プログラム起動状態

まず、プログラムを起動させると、図 2.2 のような Window 表示用のフォームが表示される。

ここでは、メニュー「ファイル」を選択し、ファイルの新規作成が開くかを決定する。ファイルを開くと図 2.3 のようなエディタ画面となる。

エディタ画面の左下に全体のページの中の現在のページ、その

右に分析の種類メモ、備考にデータ入力の際のパラメータの覚えが記されている。データは4枚のシートである。

次にエディタに最低限必要な機能について述べる。

1. セルへの簡単な入力と訂正
2. ファイルの新規作成、読み込み、上書き保存、名前を変えて保存
3. セルの書式設定
4. 行や列の削除と追加
5. セルの切り取り、コピー、貼り付け
6. セルの計算機能

	1	2	3	4	5	6	7
1	4.88	291.5	0	0.71	0	0	0
2	0	24.51	0.05	0.97	0	0.28	0
3	0	0.18	130.73	0.66	0	0	0
4	0.66	1.99	0.81	39.07	0	0	0
5	0	1.29	2.26	0	16.71	0.45	0.14
6	0	0	0	0.03	0	41.22	0
7	0	0	1.58	0	0	0.55	1.58
8	0	0.72	0	0.63	0	0.34	0
9	16.51	10.37	17.02	0.64	5.88	4.07	3.79
10	0.02	0.23	2.86	0.05	3.75	1.59	4.24
11	0	0.35	1.18	0.05	1.28	0.1	8.16
12	2.44	0.64	3.51	0.62	2.4	0.02	0.58
13	3.67	9.15	21.08	4.05	6.21	6.59	1.84
14	0	0.74	2.78	0.04	1.22	1.63	1.55
15	1.84	6.44	14.87	2.29	4.79	4.01	2.86
16	2.14	0.48	5.57	0.27	1.51	0.99	3.48

図 2.3 エディタ画面

これらは、標準的なソフトに必ず含まれている機能で、特に説明するまでもない。現在、機能的にはどこにないが、5.の計算機能を除いて全て含まれている。今後機能強化を図る予定である。この他にはフィールド名とレコード名の入力、及び分析の実行機能が必要である。エディタは分析処理のフォームを呼び出す要のような役割をする。

社会システム分析  
ファイル(F) 表示(V) ウィンドウ(W) テスト(T)

データ編集 産業連関分析

基準ページ: 1

産業数: 19

国内需要項目数: 1

輸出項目数: 0

輸入項目数: 1

非競争輸入項目数: 0

粗付加価値項目数: 1

誤差項目数: 0

輸入の取扱い

- 競争輸入型
- 非競争輸入型
- 混合型

非競争輸入型を競争輸入型に  
変換して計算する

逆行列

- (I-A)逆行列
- (I-(I-M)A)逆行列

投入係数表 グラフ Exit

付加価値係数表

輸入係数表

Leontief逆行列 グラフ

影響力・感応度係数

前方・後方連関指標

生産誘発額 生産誘発係数

最終需要依存度

輸入誘発額 輸入誘発係数

付加価値誘発額 付加価値誘発係数

総合輸入・付加価値係数

単位構造

産業選択 [ ] 表 グラフ

2時点比較 (RAS法)

比較ページ [ ] 間隔 [ ]

図 2.4 産業連関分析メニュー

多くの分析を1つのエディタで扱うために、それぞれの分析用の最低限度の形式を守ったデータの入力が必要である。この部分がソフトの操作上最も分かりにくい部分である。いくつかの例と解説書が必要であろう。

それぞれの分析フォームは、他とは完全に独立に、特徴的に作られる。現在作られているのは、産業連関分析<sup>2),3)</sup>、KSIM<sup>4)</sup>、ISM<sup>4),5)</sup>、AHP<sup>6),7)</sup>、デシジョンツリー<sup>5)</sup>、線形計画法<sup>8)</sup>である。後程それぞれデータ構造の特徴を説明するが、この中で特に詳細に作っているのが、産業連関分析とKSIMである。これらは実際の分析に利用している。その他の分析については、今後機能強化を図り、多くの分析や検定を組み込んでいく必要がある。例として図2.4に産業連関分析のメニューの一部を表示する。

これは、実用性を考えたため、教育用としてはメニューが少し複雑すぎるかも知れない。必要な記入欄にデータを入れ、知りたい項目のコマンドボタンをクリックするだけで簡単に結果が得られる。

分析の出力結果は、個々のフォームに付随した個別の出力形式の他に、3つの共通の出力用フォームにも表示出来る。即ち、テキストボックスによる文章出力、グリッドコントロールによる行列出力、グラフコントロールによるグラフ出力である。

図 2.5 結果のグリッド出力

メンテナンスの都合上、出来る限りこれらの共通フォームを利用することを薦めたい。ここにグリッドやグラフには必要最小限の書式設定の機能を付けているが、教育用にはかえって好



--	--	--	--	--	--	--	--	--	--

このプログラムではまだ作られていないが、一般的な統計ソフトでは、行にレコード、列にフィールドをとって、データを1枚のシートとして与える。

線形計画法<sup>8)</sup>のデータは、制約式の係数行列と等号または不等号、その比較値を入れ、その下に目的関数の係数を入力する。最大化・最小化の選択は分析メニューの中で指定する。

表 3.2 線形計画法のデータ

		変数名					空欄	
空欄							等号・不等号	比較する値
	制約式の係数行列							
	目的関数の係数ベクトル							

意志決定支援手法の1つ AHP<sup>6),7)</sup>では、データ構造は少し複雑である。最初の1枚には全ての代替案や評価基準の項目全体の関係が行列的に示される。即ち、経路で繋がっている場合は、行列上のその部分に1を入力する。この構造によって、シートの2枚目以降の項目名が決まり、値は分析の実行中に指定する。即ち、1枚目のデータにより、分析画面には各項目について1対比較の画面が現われ、その項目を埋めて重要度や適合度を求め、納得のいくものであれば、データをシートに保存する。次回の分析からは1対比較のデータを利用出来る。

表 3.3 AHP のデータ

		全項目名 (代替案・評価基準)				
全項目名						
	2 値関係行列					

		項目名			
項目名					
		1 対比較			

2 頁目以降

産業連関分析<sup>2),3)</sup>では、産業連関表がそのままシート上に表示される。但し、全体の生産高についてはそのまま残すが、小計については全て省いて入力する。また、産業連関表の年次比較のために、複数の年次について入力することを考えて、複数枚の産業連関表を扱えるようにしている。その他に分析に必要なデータは、産業数、国内需要項目数、輸出項目数、輸入項目数、付加価値項目数等であるが、これらは分析画面内で指定するようになっている。

表 3.4 競争輸入型産業連関表のデータ

	産業名				項目名			項目名	名
産業名									
		投入産出額			国内需要・輸出			輸入	合計
項目名		付加価値			同じものが複数枚				
名		合計							

KSIM<sup>4)</sup>は「第3次産業が盛んになる」とか「経済発展が進む」等の因果的なネットワークにおいて相互の影響の強さを段階的に仮定して、定性的に時間的なシミュレーションを行う手法で、結果の正確さよりも直感的に傾向をみる場合の補助となる。データは、各変数の初期値(1行目)と、ある変数から他の変数への影響力の強さを表す数値で与えられるが、後者は行に始点、列に終点を取って行列形式で表せばよい。分析画面では、シミュレーションの期間、その間の時間の区切り、影響力の強さを補正するパラメータ等のデータを指定する。

表 3.5 KSIM のデータ構造

	変数名					
初期値	変数の初期値					
変数名						
		影響力の強さを与える関係行列				

ISM<sup>4),5)</sup>は、社会問題、経済問題、細かくは演説や論説の論理展開等について、それらの因果構造をグラフ的に解析する手法であるが、グラフ構造は行列表示することが出来るので、KSIMの影響力を与える行列と同じようにデータ化出来る。但し、複数の人の意見を集約する必要もあるので、シートは複数枚使用出来るようにしている。

表 3.6 ISM のデータ構造

	変数名					
変						
		2値関係行列				


同じものが複数枚

デシジョンツリー<sup>5)</sup>については、1つの独立した分析として加えることが適当であるか検討中であるので、ここではデータ構造について特に説明しない。しかし、ツリー構造である以上行列表示が可能であり、このプログラムで用いている複数枚のシートのデータ形式の枠内に収まる。

### 3.2 ファイルのデータ形式

データ構造から、それを保存するためのファイル構造が決まる。以下にこのソフトウェアの保存ファイルの構造を示す。非常に簡単な形なので、他のソフトで利用することは簡単である。

表 3.7 ファイル構造

ページ数(n),最大行数(mr),最大列数(mc) ページ 1 行数( $r_1$ ),ページ 1 列数( $c_1$ ) 予備,フィールド名 1,フィールド名 2, ... ,フィールド名 $c_1$ レコード名 1, data11, data12, data13, ... , data1 $c_1$ レコード名 2, data21, data22, data23, ... , data2 $c_1$ . . . . . レコード名 $r_1$ , datar <sub>1</sub> 1, datar <sub>2</sub> 2, datar <sub>3</sub> 3, ... , datar <sub>1</sub> $c_1$ . . . . . . . . . . ページ n 行数( $r_n$ ),ページ n 列数( $c_n$ ) 予備,フィールド名 1,フィールド名 2, ... ,フィールド名 $c_n$ レコード名 1, data11, data12, data13, ... , data1 $c_n$ レコード名 2, data21, data22, data23, ... , data2 $c_n$ . . . . . レコード名 $r_n$ , datar <sub>n</sub> 1, datar <sub>n</sub> 2, datar <sub>n</sub> 3, ... , datar <sub>n</sub> $c_n$ 分析名 (覚え書き) パラメータ (覚え書き)
---

各ページ毎に指定される行数と列数には、フィールド名やレコード名の部分は含まれない。最後の2行は、分析名とその分析で利用するパラメータの覚え書きである。パラメータ等は比較的忘れやすいので、きちんと書いておくと便利である。このファイルの構造と、データは3次元配列であるということから分かるように、1次元はページ、残りの2次元が行と列を表す。各ページ毎の行数と列数が異なるが、データとしては最大行数と最大列数のシートを必要分だけ用意する形となるので、効率的に多少悪くなるが、取扱いの簡単化からこの形式を選んだ。

## 4章 考察と将来の展望

本ソフトウェアでは開発用言語として、プログラミングの容易性と資産の継続性を考えて Visual BASIC を選択した。ある程度実用的にまとまった段階で C++ に移植することも考えてもよいが、プログラミングがすべてのユーザーに容易であるとは言えない。また、インターネットでの利用を考えると、Java での記述にも興味がある。しかし、アプレットからのローカルファイルの扱いに規制があることやグラフの表示に相当な労力を要することから、しばらくは取り扱わないことにする。またこのソフトは現在エディタ機能が貧弱である。そこで、Microsoft Excel の VBA で同機能のものを作成してみることは興味深い。最終的には大規模なプログラムになるので、移植は開発効率とプログラムのスピードを考慮して慎重に検討したい。Visual Basic は Ver.5.0 から完全なコンパイル機能を持つようになり、処理速度も実行ファイルの取扱いも進歩した。これらの状況から、多くの人（学生も含めて）気軽に参加出来る Visual Basic は当面最良の選択と思われる。

我々は将来的に以下のような分析メニューとそれを利用する教育を考えている。

### 1. 基本統計

単に個々の統計量の計算や検定だけでなく、データの種類や分布に応じて検定手順をガイドする機能を有する統計的検定ツールとする<sup>10)</sup>。

### 2. 多変量解析

重回帰分析、判別分析、主成分分析、因子分析、クラスター分析、数量化等基本的な分析手法で、結果は必要な項目を絞り込んで表示する。

### 3. OR

線形計画法、PERT/CPM、待ち行列、在庫管理、品質管理、DEA、AHP 等基本的な手法を、データ構造を吟味しながら導入する。特に待ち行列、在庫管理等はシミュレーションを活用した実験ツールの形を考える。

### 4. その他の分析

ISM、DEMATEL、KSIM、デシジョンツリー、効用理論、産業連関分析等、数値的に扱えないデータまでも含めた分析の中で、どの部分に計算機が利用出来るかまでも含めて考える。

これら以外にも多くの分析が存在し、様々なデータ形式が必要となる。今後、これらの分析を扱いながら、このソフトウェアの可能性を試して行きたい。また、これを講義に活用し、学生の理解度や満足度を調査したい。

分析プログラムを作成する段階で、Visual Basic のバージョンが変わり、グリッドとグラフに影響が現れた。特にグラフについては、元のバージョンのものと大きく異なり、仕様の変更が

起こっている。この論文においても、グラフの表示が現在のものと異なっている。今後、より使い易いグラフ表示機能を提供するために、プログラムに改良を加えて行かなければならない。

## 参考文献

- 1) 福井正康, 佐藤真司, 田口賢士, 統計学・経営科学分析システム UASM1 - 基本システム -, 福山大学経済学論集, 17 巻 2・3 合併号, 110-136, 1993.
- 2) 宮沢健一編, 産業連関分析入門, 日本経済新聞社, 1991.
- 3) 鳥居泰彦, 経済発展理論, 東洋経済新報社, 1979.
- 4) 榎木義一他編, 参加型システムズ・アプローチ, 日刊工業新聞社, 1981.
- 5) 藤田恒夫, 原田雅顕, 決定分析入門, 共立出版, 1989.
- 6) 利根薫, ゲーム感覚意思決定法 - AHP 入門 -, 日科技連出版, 1986.
- 7) 利根薫・眞鍋龍太郎編, AHP 事例集, 日科技連出版, 1990.
- 8) 三根久, オペレーションズ・リサーチ上下, 朝倉書店, 1974, 1976.
- 9) 福井正康, 田口賢士, 社会システム分析のための統合化プログラム 2 - 産業連関分析・KSIM・AHP -, 福山平成大学経営情報研究, 3 号, 1998.
- 10) 丹後俊郎, 古川俊之監修, 医学への統計学, 朝倉書店, 1983.
- 11) 篠崎壽夫・松下祐輔編, 工学のための応用数値算法入門(上), コロナ社, 1986.
- 12) 脇本和昌・垂水共之・田中豊編, パソコン統計解析ハンドブック 基礎統計編, 共立出版, 1984.

## 補遺 開発用資料

ここでは分析プログラムを作成するに当たって必要となる情報を提供する目的で、内部変数や利用可能な関数等の詳細な説明を行う。特にエディタ上のデータの形式と、結果表示のフォームの利用法はプログラム作成に不可欠である。これらの資産を利用する際、本プログラムではグローバル変数と一部のプロシージャが重要な役割を演じるが、以下ではまずこれらの意味と利用法について説明する。ここで用いる変数にはグローバル変数であることが明らかのように、変数名の最初に `g_` を付けている。今後グローバル変数と明記しない場合でも `g_` の有無で意味を判断してもらいたい。

また、汎用的な関数の作成に関しては、他言語への移植の可能性を考えて、以下のことに注意した。

1. 変数の型定義は必ず行う。
2. 変数や関数の型は、出来るだけ C 言語等で標準的に用いられているものを使う。

特に、Variant 型については、非常に使い勝手が良いので苦慮したが、特別な場合を除いて使用

は控えることにした。これ等以外に、プログラムに関して一般的に言われている注意は守るよう心掛けた。

## A.1 エディタ上のグローバル変数

エディタは複数枚の大きさの異なるシートの集合である。全体のシートのページ数は (グローバル) 変数 `g_pagenumber` で与えられる。これは、もちろん最初の 1 枚を 1 と数える。現在表示されているページは変数 `g_currentpage` で与えられる。これは最初のページを 0 とする。

次に各ページのグリッドの行数と列数はそれぞれ配列変数、`g_rownumber`(ページ番号)と `g_colnumber`(ページ番号)で与えられる。ここにページ番号は先頭ページを 0 とする。例えば現在ページの行数を知りたい場合は `g_rownumber(g_currentpage)` でよい。これらはすべて Integer 型である。グリッドは上部に 1 行、左部に 1 列、フィールド名とレコード名を入れるセルがあるが、行数と列数はこの名前部分を除いた数である。

計算データは変数 `g_grid`(ページ番号,行番号,列番号)に入っている。行番号と列番号について、`g_grid(*,0,*)`と `g_grid(*,*,0)`はそれぞれフィールド名とレコード名を表しており、実際のデータは `g_grid(page, 1, 1)`から始まっている。`g_grid()`の型は数値の場合も文字列の場合もあることを想定し、Visual Basic 特有の Variant 型とした。メモリはその分多く消費されるが、データの桁揃え等柔軟な処理が可能となる。グリッドに表示されているデータは、これらを書式指定して String 型に変換したものである。以上まとめると表 A.1 のようになる。

表 A.1 エディタ上のデータ

変数定義	備考
Global <code>g_pagenumber</code> as Integer	シートの枚数 [1-]
Global <code>g_currentpage</code> as Integer	現在開いているシートの頁 [0-] ( <code>g_pagenumber-1</code> )
Global <code>g_rownumber()</code> as Integer	シートの行数 <code>g_rownumber(頁)</code> (0 行目は数えない)
Global <code>g_colnumber()</code> as Integer	シートの列数 <code>g_colnumber(頁)</code> (0 列目は数えない)
Global <code>g_maxrownumber</code> as Integer	シートの最大行数
Global <code>g_maxcolnumber</code> as Integer	シートの最大列数
Global <code>g_grid()</code> as Variant	データ <code>g_grid(頁, 行, 列)</code> (行列名を含む)

## A.2 結果表示のグローバル変数

次に、分析結果表示用のグローバル変数について説明する。分析結果の表示には、開発者独自に作成する方法の他に、共通の環境としてテキスト (ボックス) 表示、グリッド (コントロール) 表示、グラフ (コントロール) 表示が用意されている。通常の記事出力の表示には、テキスト表示が適している。また、行列やベクトルで表される結果にはグリッド表示、それらのグラフを見る場合はグラフ表示である。グラフ表示では、結果の表示に最適なグラフを分析毎にデフォルトで設定しておく必要がある。

結果のテキスト表示には、変数 g\_retextcaption に結果表示フォームのタイトルを入れ、フォームをロードした後、resprint プロシージャや resprints プロシージャを用いて結果を出力する。例えば、以下のようにする。

```
Redim g_retextdata(1)
g_retextcaption = "平均値と標準偏差"
Load frmResultText
frmResultText.Show
resprint "平均      = " & Format(mean, "0.000")
resprints "標準偏差 = " & Format(dev, "0.000")
```

ここに resprint プロシージャは改行を含み、resprints プロシージャは改行を含まない。また、frmResultText.Show の位置は文末でも良い。

以下にテキストボックス表示の結果出力に利用するグローバル変数とプロシージャについてまとめる。

表 A.2 テキスト表示のためのグローバル変数とプロシージャ

変数定義	備考
Global g_retextcaption as String	テキスト表示フォームのタイトル
Sub resprint(str as String)	テキストの表示 (改行を含む)
Sub resprints(str as String)	テキストの表示 (改行を含まない)

結果のグリッド表示も g\_resgridcaption にフォーム名を入れ、表示するデータは g\_resgriddata()に入れる。その際、g\_resgridrow と g\_resgridcol にグリッドの行数と列数を入れるが、最初の行と列は、0 行目と 0 列目としてフィールド名とレコード名に割り振られており、行数と列数に含まれない。利用法は例えば以下のようにする。

```
g_resgridcaption = "投入係数行列"
g_resgridrow = 10
g_resgridcol = 10
Redim g_resgriddata(g_resgridrow,g_resgridcol)
For i=1 To g_resgridrow
  g_resgriddata(i,0) = "産業 " & Format(i)
  g_resgriddata(0,i) = "産業 " & Format(i)
  For j=1 To g_resgridcol
    g_retextdata(i,j) = a(i-1,j-1)
  Next j
Next i
Load frmResultGrid
frmResultGrid.Show
```

以下にグリッドコントロール表示の結果出力に利用するグローバル変数についてまとめる。

表 A.3 グリッド表示のためのグローバル変数

変数定義	備考
Global g_resgridcaption as String	グリッド表示フォームのタイトル

Global g_resgridrow as Integer	グリッドの行数（最初の行は含まない）
Global g_resgridcol as Integer	グリッドの列数（最初の列は含まない）
Global g_resgriddata() as Variant	グリッドの内容（0行目 0列目には名前が入る）

結果のグラフ表示もグリッド表示と殆ど変わらない。g\_resgraphcaption にフォーム名を入れ、表示するデータは g\_resgraphdata() に入れる。その際、g\_resgraphrow と g\_resgraphcol にグラフデータの行数と列数を入れるが、最初の行と列は、グリッド表示同様、行数と列数に含まれない。グリッド表示と異なる点はグラフの種類を設定しなければならないことである。そのために、ここでは、g\_graphtype と g\_graphstyle<sup>\*)</sup> の変数を用意してある。これらは、Visual Basic グラフコントロールの GraphType プロパティと GraphStyle<sup>\*)</sup> プロパティにそれぞれ対応している。マニュアルを参照していただきたい。

これらのグローバル変数の利用法は例えば以下のようにする。

```

g_resgraphcaption = "シミュレーション結果"
'元の Version での記述*)
  g_graphtype = 6      '折れ線グラフ
  g_graphstyle = 0    '標準
'新しい Version での記述
  g_graphtype = VtChChartType2dLine ' 2次元折れ線グラフ
g_resgraphrow = 20
g_resgraphcol = 4
ReDim g_resgraphdata(g_resgraphrow, g_resgraphcol)
For i=1 to g_resgraphcol
  g_resgraphdata(0,i) = "変数 " & Format(i)
Next i
For i=1 To g_resgraphrow
  g_resgraphdata(i,0) = Format(i/2#,"#0.0")
  For j=1 To g_resgraphcol
    g_resgraphdata(i,j) = data(i-1,j-1)
  Next j
Next i
Load frmResultGraph
frmResultGraph.Show

```

グラフ出力について以下に利用可能なグローバル変数をまとめる。

表 A.4 グラフ表示のためのグローバル変数

変数定義	備考
Global g_graphtype as Integer	グラフコントロールプロパティ GraphType に対応 <sup>*)</sup>
Global g_graphstyle as Integer <sup>*)</sup>	グラフコントロールプロパティ GraphStyle に対応 <sup>*)</sup>
Global g_resgraphcaption as String	グラフ表示フォームのタイトル

<sup>\*)</sup> Visual Basic のバージョン変更により、グラフ表示に大きな影響があった。g\_graphstyle 変数はなくなり、g\_graphtype 変数はプロパティ ChartType に対応するようになった。

Global g_resgraphrow as Integer	グラフの行数（最初の行は含まない）
Global g_resgraphcol as Integer	グラフの列数（最初の列は含まない）
Global g_resgraphdata() as Variant	グラフの内容（0行目0列目には名前が入る）

これらの他に数字選択の場合に利用するグローバル変数を定義する。数字選択は "1,3,5-8" 等の文字列表記から数字 1,3,5,6,7,8 を認識するもので、様々な場面で利用される。これには Sub select\_field(str as String) というプロシージャを利用する。例えば、select\_field "1,3,5-8" とすると、グローバル変数 g\_repnumber に数字の個数の 6、g\_repvar(0) から g\_repvar(5) に 1,3,5,6,7,8 の値が入る。統計学のソフトではこういった変数選択をよく行う。しかし、このような文字列で打ち込む形式より、別に変数選択のフォームを設けて、候補から選択させる方が分かり易い場合も多く、この形式も今後作成しなければならない。

### A.3 数学関数

ここでは、このプログラムで作成された数学関数について説明する。特に行列計算において、今後のプログラム作成に必要不可欠なものを取り上げている<sup>11)</sup>。

行列式の計算には mat\_det() 関数を用いる。

定義 Function mat\_det(matrix() As Double, col As Integer) As Double  
 機能 行列式を求める。  
 引数 matrix(): 行列式を求めたい行列 col: 列(行)数  
 戻値 行列式の値

逆行列の計算は mat\_inv() 関数を用いる。求めた逆行列は変数 revmat() に返され、戻値には正則か正則でないかの結果が返される。

定義 Function mat\_inv(matrix() As Double, col As Integer, revmat() As Double) As Integer  
 機能 逆行列を求める。  
 引数 matrix(): 元の行列 col: 列(行)数 revmat(): 逆行列が返される  
 戻値 -1: 正則 0: 正則でない(求まらない)

正値対称行列  $\mathbf{M}$  の三角行列  $\mathbf{S}$  (左下半分に値) による表現  $\mathbf{M} = \mathbf{S} \cdot \mathbf{S}'$  を求めるには、関数 mat\_tri() を用いる。

定義 Function mat\_tri(matrix() As Double, col As Integer, trimat() As Double) As Integer  
 機能 正値対称行列を分解して三角行列を求める。  
 引数 matrix(): 正値対称行列 col: 列(行)数 trimat(): 三角行列が返される  
 戻値 -1: 成功 0: 失敗

実正方行列の最大固有値とそれに対する固有ベクトルを求めるには、べき乗法を用いた関数 igen\_vector1() が用意されている。

定義 Function igen\_vector1(matrix() As Double, col As Integer, ival As Double, ivec() As Double) As Integer

機能 正方行列の最大固有値とそれに対応する固有ベクトルをべき乗法を用いて求める。  
引数 matrix(): 正方行列 col: 列(行)数 ival: 参照型として固有値が返される  
ivec(): 固有ベクトルが返される  
戻値 -1: 成功 0: 失敗

実対称行列の全ての固有値と固有ベクトルを求めるには、Jacobi 法を用いた関数 `igen_vector2()` を用いる。

定義 Function `igen_vector2(matrix() As Double, col As Integer, ival() As Double, ivec() As Double) As Integer`  
機能 実対称行列の全ての固有値と固有ベクトルを Jacobi 法を用いて求める。  
引数 matrix(): 実対称行列 col: 列(行)数 ival(): 固有値が返される  
ivec(): 2次元配列として各列にそれぞれの固有ベクトルが格納される  
戻値 -1: 成功 0: 失敗

現在は行列関数だけしか作られていないが、分析プログラムの作成を進めるに当たって必要な関数は共通に使えるようにする。

## A.4 統計関数

統計検定に欠かすことの出来ない、正規分布、 $\chi^2$  分布、F 分布、t 分布については、値から上側確率、逆に上側確率から値を求める関数を用意している<sup>12)</sup>。これら以外の関数、例えばノンパラメトリック検定に関するものや、シミュレーションに不可欠な様々な乱数発生等については順次作成して行く予定である。以下に現在共通に利用出来る関数をまとめておく。

定義 Function `pnorm_dist(u As Double) As Double`  
機能 正規分布のパーセント点から上側確率を求める。  
引数 u: 正規分布のパーセント点  
戻値 正規分布の上側確率

定義 Function `vnorm_dist(al As Double) As Double`  
機能 正規分布の上側確率からパーセント点を求める  
引数 al: 正規分布の上側確率  
戻値 正規分布のパーセント点 エラーの場合は `G_DATAERROR`

定義 Function `pchi2_dist(df As Integer, chi2 As Double) As Double`  
機能  $\chi^2$  分布のパーセント点から上側確率を求める。  
引数 df: 自由度 chi2:  $\chi^2$  分布のパーセント点  
戻値  $\chi^2$  分布の上側確率 エラーの場合は `G_DATAERROR`

定義 Function `vchi2_dist(df As Integer, al As Double) As Double`  
機能  $\chi^2$  分布の上側確率からパーセント点を求める。  
引数 df: 自由度 al:  $\chi^2$  分布の上側確率  
戻値  $\chi^2$  分布のパーセント点 エラーの場合は `G_DATAERROR`

定義 Function `pf_dist(df1 As Integer, df2 As Integer, vl As Double) As Double`  
機能 F 分布のパーセント点から上側確率を求める。  
引数 df1: 自由度 1 df2: 自由度 2 vl: F 分布のパーセント点

戻値 F 分布の上側確率 エラーの場合は G\_DATAERROR

定義 Function vf\_dist(df1 As Integer, df2 As Integer, al As Double) As Double

機能 F 分布の上側確率からパーセント点を求める。

引数 df1: 自由度 1 df2: 自由度 2 al: 上側確率

戻値 F 分布のパーセント点 エラーの場合は G\_DATAERROR

定義 Function pt\_dist(df As Integer, t As Double) As Double

機能 t 分布のパーセント点から上側確率を求める。

引数 df: 自由度 t: t 分布のパーセント点

戻値 t 分布の上側確率 エラーの場合は G\_DATAERROR

定義 Function vt\_dist(df As Integer, al As Double) As Double

機能 t 分布の上側確率からパーセント点を求める。

引数 df: 自由度 al: 上側確率

戻値 t 分布のパーセント点 エラーの場合は G\_DATAERROR

# Multi-purpose Program for Social System Analysis

Masayasu FUKUI and Katashi TAGUCHI

Department of Management Information, Faculty of Management,  
Fukuyama Heisei University

**Abstract:**

Unified software for analyzing social phenomena including business management is proposed. The software can be applied to analysis on any form of datum such as tables and structural graphs. For the purpose, various relevant analytical methods are consolidated into a package. System structure of the software is introduced in this paper.

**Keywords:**

computer software, social system analysis, statistical analysis, OR