

# 経済統計分析における ExcelVBA の利用について

大野 拓行

## I はじめに

ExcelVBA は Excel に搭載されているプログラム言語である。Microsoft では 1991 年に Windows 用のアプリケーション作成用にプログラム言語 Visual Basic を発表した。その後、Visual Basic を基に Office の各製品 (Application) のマクロ作成用のプログラム言語である VBA (Visual Basic for Applications) を開発してきた。日本国内では 1994 年発売の Office 4.2 における Excel に初めて VBA が搭載され、1999 年発売の Office 2000 において全ての Office 製品のマクロ言語が VBA に統一された<sup>(1)</sup>。本稿のテーマは経済統計分析における ExcelVBA の利用について、であるが本論に入るまえに経済統計分析におけるコンピュータの利用の仕方を概観しておくことは重要である。

経済統計分析におけるコンピュータの利用の歴史を考えると、その初期 (1960~70 年代) においては、汎用 (大型) コンピュータが主に利用されていた<sup>(2)</sup>。分析に必要なプログラムは多くの場合、分析者自らが FORTRAN などのプログラム言語で作成していた。その後、コンピュータの普及に伴って、

---

(1) VBA は VB (Visual Basic) に基づいて開発された言語であるが、VB とは異なる点も多くある。また、ExcelVBA と同様に、WordVBA、AccessVBA、…などがある。さらに ExcelVBA も Excel のバージョンアップとともにバージョンアップされており、Excel 2002 に搭載されている VBA は Excel 2002 VBA (バージョン 6.3) である。本稿で提示したプログラムは Excel 2002 VBA および Excel 2000 VBA (バージョン 6.0) で動作確認を行っている。また、Excel 97 VBA では動作しないことも確認している。

(2) 筆者が汎用コンピュータをはじめて利用したのは 1973 年で、まだ、紙テープとパンチカードが混在していた時代であり、バッチ処理が主流であった。

データ分析が広汎に行われるようになると、SAS や SPSS に代表される統計分析用のパッケージ・ソフトウェアの利用が主流となり、先進的な分析手法の開発や、パッケージ・ソフトウェアの利用が困難な場合にのみ、独自のプログラムを作成するようになってきた。自ら独自のプログラムを作成するケースは、いまでは、次第に少なくなりつつある。

最近では主流になったパソコンでのデータ分析においても、パソコンの黎明期においては、汎用コンピュータの場合と同様に BASIC などの言語で分析目的にあうプログラムを作成していたが、パソコンの能力の向上と共に、統計分析の環境も大きく変化してきたと考えられる。すなわち、

① 汎用コンピュータで開発された専用ソフトウェアのパソコンへの移植版、あるいはパソコンのために開発された専用ソフトウェアを利用できる環境。

② 表計算ソフト（その代表格が Microsoft 社の Excel）を利用する環境。の2極化である。データ分析の必要性の高まりと高性能パソコンの普及につれて、多くの人々がパソコンによるデータ分析を行うようになってきたが、その大部分は②の環境であることも事実であろう。

筆者はこれまで、学生や社会人の方から「Excel で…の分析はできますか？」という質問を、時々、受けてきた。その際の答えの1つは、「Excel では難しいが SAS や TSP をつかえば簡単にできますよ。」というものであり、この場合、質問をした人は「そうですか」と言って少し困った顔をするが納得する。困った顔をする理由は、SAS や TSP を使える環境にない、ソフトの値段、操作法を学ぶ必要がある、現在抱えている問題に対してそこまでする必要があるのか、…など様々だと推測できる。質問した人の中で、その後、SAS や TSP を使用して問題を解決できたと報告してきた人は質問者が研究者以外では皆無であった。

一方、質問に対する回答が「Excel では難しいがアドインソフト…ではできますよ」というものであれば、多くの社会人の方は満足して帰り、多くの人「問題は解決できました」と報告に来る。社会で忙しく働いている人にとって

は、2～4万円は高くないかもしれない<sup>(3)</sup>。

社会におけるデータ分析の多くが Excel を中心に行われていることは事実であろう。また、Excel で困難な統計手法には手を出さないという傾向もあるのではないだろうか。アドインソフトや無償で入手可能なアドインは Excel 利用者にとっては便利なものである。しかし、その多くは統計データ一般に対するもので、経済統計データの特性に必ずしも対応したものとなっていないと筆者は考えている。

分析目的にフィットした独自の Excel 用プログラムを作成することは困難と思われがちであるが、ExcelVBA を利用すれば、他のプログラム言語に比較して極めて簡単に Excel のアドインソフトが開発可能なのである。また、プログラムを開発するまでには行かなくても、ExcelVBA の簡単な構文を知るだけでも、Excel による作業効率が飛躍的に高まるのである。

以上のような考えに基づいた本稿の目的は①経済統計データの特性に対応した実用プログラムを提示すること、②プログラムの解説を通じて、Excel ユーザーに ExcelVBA 利用の利点を示すこと、にある。

## II Excel 2002 VBA によるプログラムの実例<sup>(4)</sup>

マクロ (VBA で記述されたプログラム) は Excel ブックのモジュールに記述する。Excel にはデータを表示するためのワークシート、グラフを表示するためのグラフシートが用意されているが、マクロを記述するための専用シートがモジュールである<sup>(5)</sup>と考えるとイメージしやすい。モジュールを編集するためには、Excel から VBE (Visual Basic Editor) を起動する<sup>(6)</sup>。通常マクロを記述する標準モジュールには Module 1, Module 2, …と複数モジュールを作成すること

(3) これが学生だと、ほとんど購入はしない。あとで学生のレポートを見ると、分析内容を通常の Excel 操作でできる範囲に限定しているのがほとんどである。

(4) プログラムの作成は文末の文献を参考にすると共に <http://www.moug.net/> に開設されている、「Q&A 掲示板 ExcelVBA」を活用させて頂いた。質問に回答して頂いた多くの回答者の方にこの場を借りて感謝いたします。また、佐藤幹子助手は初案の段階から本稿を読んでいただき、有益なコメントをいただき感謝しています。

が可能であり、個々のモジュールは宣言セクションと1つ以上のプロシージャ(マクロ)からなる。<sup>(7)</sup>以下、実際のプログラムでExcelVBAの特徴を見ていくことにするが、すべてのコードを解説することはできないので、プログラムの全コードを本稿の付録に記載している<sup>(8)</sup>ので適宜、参照されたい。

本プログラムの標準モジュールは以下の5つのモジュールから構成されている。

Module 1：度数分布表作成のための情報

Module 2：度数分布表の作成

Module 3：度数分布表からの基本統計量

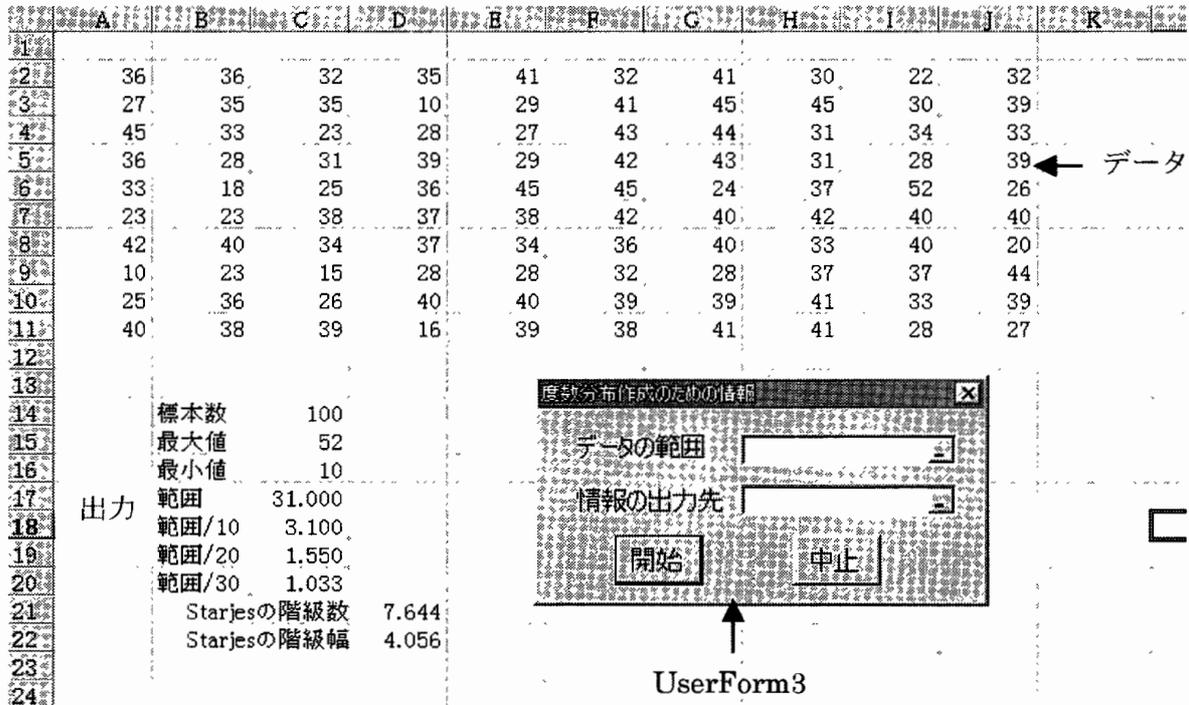
Module 4：ヒストグラムの作図

Module 5：度数分布表の再構成

- 
- (5) モジュールには「標準モジュール」、「ユーザーフォーム」、「クラスモジュール」、「Excel オブジェクト」の4種類あり、最も用いられるのが「標準モジュール」である。本稿においては特に断らない限り、モジュールは「標準モジュール」を意味する。Excelにおける「マクロ記録」により作成されるコードも「標準モジュール」に記録される。注意すべき点は、①同じマクロでも記述されるモジュールにより動作が異なること、②Excelブックが削除されるとそれに付随するモジュールもすべて自動的に削除されること、である。
- (6) ひとつのVBEの起動方法はExcelのメニューバーから、「表示」-「ツールバー」を選択し、「Visual Basic」をチェックして、「Visual Basic」ツールバーを表示し、「Visual Basic Editor」ボタンをクリックすることである。本稿においては、VBEの使用法には触れないが効率的なコーディングのためにはVBEに習熟する必要がある。特に、コーディング支援のための「自動クイックヒント」と「自動メンバ表示」は便利であろう。
- (7) 複数のモジュールのイメージはExcelブックにSheet 1, Sheet 2, …と複数のワークシートを作成できることをイメージすると理解しやすい。個々のモジュールの宣言セクションにはそのモジュール全体の設定などを記述するが、特に必要なければ記述する必要はない。プロシージャという用語の内容はほぼマクロという用語と同等と思われるが、プログラミングの世界ではプロシージャという用語が多用されているので本稿でも以後、プロシージャと表記する。
- (8) また、アドイン形式のファイルを、サンプルファイルと共に、<http://www.ec.kagawa-u.ac.jp/~ohno/>に置いているので、Ⅱの「8. アドインの利用」を参考にExcelに組み込んで利用できる。

### 1. Module 1：度数分布表作成のための情報

図1 Module 1 の実行画面



Module 1 は度数分布表の作成に際して有用な情報である，データ数，最大値，最小値，範囲，範囲／10，範囲／20，範囲／30，Starjes の階級数，Starjes の階級幅<sup>(9)</sup>を計算するものである。モジュールは宣言セクション（行番号 1）と 1つのプロシージャ（行番号 3～52）とからなっている<sup>(10)</sup>。

（1行目）：このモジュール内では使用する変数は全て宣言文により宣言することを明示<sup>(11)</sup>。

（3行目）：プロシージャ myFreq 1 の開始。1つのプロシージャはプロシ

(9) データ数を n とすると，Starjes の階級数は  $1 + \ln(n) / \ln(2)$  で定義される。ここでは範囲を Starjes の階級数で除したものを Starjes の階級幅とした。

(10) コードに付けた行番号は説明のためのもので，実際のプログラムでは付加されない。

(11) なじみのない言語で記述されたコードを見るとき，変数名と言語のキーワードは区別することが困難である。Module 1 において使用されている変数は，cb 10 hani myBtn myDataSheet myMax myMin myRange myRange 2 myTitle n starjes x y の 13 変数である。このうち，hani myBtn starjes の 3 変数は 5，6 行目の Dim 文で宣言されている。他の変数は Public 変数（すべてのモジュールのプロシージャで使用可能な変数）として Module 3 の宣言セクションにおいて宣言されている。

ジャ名 ( ) で始まり End Sub で終わる。Module 1 では 3 行目の Sub myFreq 1 ( ) から 52 行目の End Sub までが 1 つのプロシージャであり、プロシージャ名は myFreq 1 である。

(5 ~ 6 行) : このプロシージャのみで使用する変数<sup>(12)</sup>を宣言<sup>(13)</sup>。

(8 行目) : 変数 myTitle に文字列を入力。

(9 行目) : アクティブ・ワークシート名を変数 myDataSheet に代入。この行はプロパティの値を取得するための VBA の基本構文である。ブック、シート、セルなど Excel の構成要素は VBA でなんらかの操作を行う際に、その操作の対象となるのでオブジェクトと呼ばれ、オブジェクトが持つ属性はプロパティと呼ばれる。この行は、「オブジェクト ActiveSheet の Name プロパティを変数 myDataSheet に代入する<sup>(14)</sup>」を意味する。

(11 行目) : UserForm 3 をワークシートに表示する。ユーザーフォームは VBA のコードと組み合わせることでキーボードからのデータの取得などを Windows 風に操作可能とする手法であり、後に詳しく見る。UserForm 3 では計算に使用するセル範囲が変数 myRange に、計算結果を表示するセル番地が変数 myRange 2 に代入される。ユーザーフォームもオブジェクトの 1 つであり、この行はオブジェクトに対してある種の操作を実行するための VBA の基本構文である。オブジェクトに対して実行できる操作はメソッドと呼ばれる。この行は、「オブジェクト UserForm 3 に対してメソッド Show を実行する」を意味する。

(12 行目) : UserForm 3 において、「中止」ボタンや「×」ボタンがクリックされた場合、このプロシージャを終了する<sup>(15)</sup>。

---

(12) VBA には、Integer (整数型)、Double (倍精度浮動小数点型)、String (文字列)、…のほかに、Object (オブジェクト型) のような VBA の特色をあらわすデータ型も用意されている。

(13) モジュール内に複数のプロシージャがある場合、モジュールの宣言セクションの Dim ステートメントで宣言した変数は、そのモジュール内のすべてのプロシージャで利用可能となる。

(14) Module 1 において同種の構文は、16、17 行目に見出せる。

(15) UserForm 3 において、「中止」ボタンや「×」ボタンが押されたりした場合、変数 Cb10 が True になるようにしている。

(14 行目)：画面の動きを停止する。プロシージャの実行時にワークシートの内容が目まぐるしく変化し、画面がちらつくのを抑止している。この行はオブジェクトのプロパティの値を変更するための VBA の基本構文である。この行は「オブジェクト Application (=Excel) の ScreenUpdating プロパティの値を False に変更する」を意味する。50 行目で画面の動きを復活している。

(16, 17 行目)：UserForm 3 で計算に使用するデータ範囲が変数 myRange に代入されているので、16 行目でデータ範囲の行数を変数 x に、17 行目でデータ範囲の列数を変数 y に代入している。

(19~29 行)：セルに文字列を入力。ここでは、With~End With 構文<sup>(16)</sup>を利用しているが、例えば 21 行目は、本来 myRange 2. Cells (2, 1). Value="最大値"である。内容は「MyRange 2 を (1, 1) とした時の (2, 1) セルの値を"最大値"とする」を意味している。MyRange 2 には UserForm 3 で計算結果を表示するセル番地が入力されているので、例えば MyRange 2 が B 14 セルであるとする、B 15 セルに"最大値"が表示されることになる。

(31~33 行)：Excel のワークシート関数を利用して、データ数、最大値、最小値を求める。

(34 行目)：範囲の計算。

(35 行目)：Starjes の階級数<sup>(17)</sup>の計算。

(37~48 行)：計算結果をワークシートのセルに表示。47 行目で指定したセル範囲の小数桁を揃えるサブルーチン<sup>(17)</sup>を呼び出しているが、このサブルーチンについては Module 3 のところで触れる。

(50 行目)：画面の動きを復活する。

(52 行目)：プロシージャ myFreq 1 の終了。

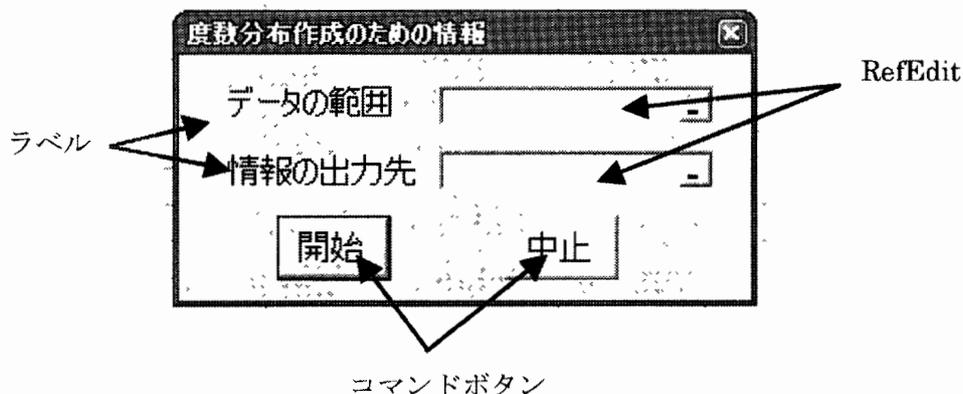
---

(16) 記述を簡略化するための構文。With~End With 間のコードにおいて共通するオブジェクト、プロパティなどを With のあとに記述し、With~End With 間のコードにおいては、With のあとに記述したオブジェクト、プロパティを、(ピリオド)で明示的に示す。

(17) ここでは、自然対数を求める VBA 関数 Log() を使用している。VBA には 140 あまりの関数が用意されているが、これと Excel のワークシート関数を混同しないように注意が必要である。

## 2. UserForm 3

図2 UserForm 3



ユーザーフォームの上に VBA で用意されている Active コントロールと呼ばれる部品を配置し、<sup>(18)</sup> 各々の Active コントロールの作業内容を VBA で記述することにより、独自のダイアログボックスを簡単に作成可能である。

Excel 2002 VBA で用意されている Active コントロールは 15 種類あるが、UserForm 3 ではその中の 3 種類が利用されている。

- ① ラベル：ユーザーフォーム上に文字列を表示するための表示専用のコントロール。UserForm 3 では 2 つのラベルを使用し、文字列「データの範囲」、「情報の出力先」を表示している。
- ② RefEdit：主にワークシート上のセル範囲を取得するために用いられるコントロール。UserForm 3 では 2 つの RefEdit を使用し、データ範囲と出力先の取得に利用されている。
- ③ コマンドボタン：処理の開始や処理のキャンセルをプログラムに伝達する時に使用するコントロール。UserForm 3 では 2 つのコマンドボタンを使用し、作業の「開始」、「中止」を選択できるようにしている。

UserForm にコントロールを配置したら、次に、各コントロールの作業内容を VBA で記述する。UserForm 3 ではラベルは表示専用であるのでその必要はないが、コマンドボタンではボタンがクリックされた時の作業（イベントプロ

(18) ユーザーフォームのデザインや Active コントロールの方法については多くの VBA 入門書に述べられているのでここでは触れない。

シージャ) を記述する必要がある。

UserForm 3 のコードで特徴を見てみることにするが、その前に次の点に留意しておく。

- ① ユーザーフォームに対するコードは標準モジュールではなく、「フォーム」と呼ばれるモジュールに作成する。
- ② ユーザーフォームに配置したコントロールには名前が自動的に割り当てられる。UserForm 3 では、

RefEdit 1：データ範囲の入力用 RefEdit

RefEdit 2：出力先の入力用 RefEdit

CommandButton 1：「開始」ボタン

CommandButton 2：「中止」ボタン

となっている。

- ③ プロシージャ名は各コントロールに対するイベントによって決められている。

付録のコードより、UserForm 3 のコードは4つのイベントプロシージャからなっていることがわかる。

(2～15)：「開始」ボタンがクリックされた時の処理。RefEdit 1、RefEdit 2 に入力データがあることを確認し、変数 myRange に RefEdit 1 の範囲、変数 myRange 2 に RefEdit 2 の範囲を代入し、UserForm 3 を非表示にする。

(18～21)：「中止」ボタンがクリックされた時の処理。変数 cb 10 の値を True にし、UserForm 3 を非表示にする。

(24～26)：ユーザーフォームが表示される直前の処理（初期化）。UserForm 3 が表示されると RefEdit 1 を最初にフォーカスする。<sup>(19)</sup>

(29～32)：ユーザーフォームの右上隅にある「×」ボタンがクリックされた時の処理。変数 cb 10 の値を True にし、UserForm 3 を非表示にする。

---

(19) ユーザーフォームのコントロール間のフォーカスの順番は「タブオーダー」コマンドにより変更する。

### 3. Module 2：度数分布表の作成

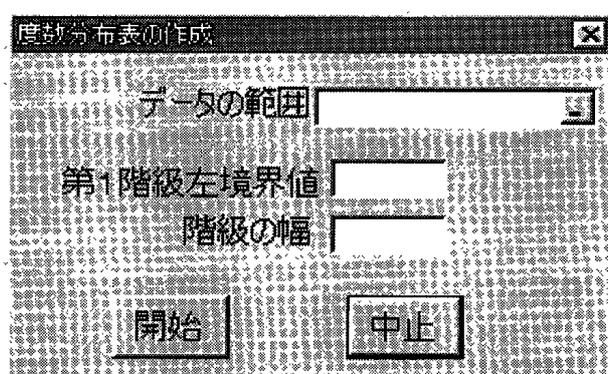
このプロシージャは分類されてデータを度数分布表に分類するものである。Excelの「分析ツール」には「ヒストグラム」コマンドがあるが、以下のような不便な点がある。

- ① 度数分布表の階級は（左境界値，右境界値）で示されるか，階級値で代表されるのが普通であるがExcelの「ヒストグラム」コマンドでは階級は右境界値で代表されており，利用に際しては注意が必要である。
- ② 階級区分は〇〇以上～△△未満というものが日常的に最もよく見かけるものであるが，Excelの「ヒストグラム」コマンドでは，〇〇より大～△△以下となり，作成される度数分布表の見方に注意が必要である。
- ③ 作成されるグラフが棒グラフで，項目軸に表示されるのが階級の右境界値であることなど，利用に際しては修正が必要である。

このような不便を解消するために作成したのがModule 2である。UserForm 4により，データのセル範囲，第1階級の左境界値，階級の幅を指定すると，新しいワークシートに，データ，度数分布表，ヒストグラムを作成するものである。モジュールはプロシージャ myFreq 2（行番号3～60）と関数サブルーチン myFreq（行番号63～70）からなる。付録のコードから特徴を見ていく。

(11)：UserForm 4 を表示する。

図3 UserForm 4



UserForm 4 においては以下のことが行われる。①データのセル範囲が変数 myRange に，第1階級の左境界値が変数 startv に，階級の幅が変数 cwidth に代

入される。②データから最大値 myMax, 最小値 myMin を計算する。③新しいワークシート (名前は myWorkSheet) が挿入され, データが新しいワークシートにコピーされる。

(18) : ワークシート関数 Count により, データ数 n の計算。

(19) : ワークシート myWorkSheet を選択。

(21) : 度数分布表の左上隅セルを選択。変数 x に 16 行目でデータ範囲の行数が代入されているので, データの下に 2 行空け, B 列が度数分布表の左上隅セルとなる。

図 4 Module 2 の出力の一部

左境界値	右境界値	階級値	度数	累積度数	相対度数	累積度数%
10	14	12	2	2	2.00%	2.00%
14	18	16	2	4	2.00%	4.00%
18	22	20	2	6	2.00%	6.00%
22	26	24	8	14	8.00%	14.00%
26	30	28	14	28	14.00%	28.00%
30	34	32	14	42	14.00%	42.00%
34	38	36	17	59	17.00%	59.00%
38	42	40	27	86	27.00%	86.00%
42	46	44	13	99	13.00%	99.00%
46	50	48	0	99	0.00%	99.00%
50	54	52	1	100	1.00%	100.00%

(23~29) : 度数分布表の表頭ラベルの表示。<sup>(20)</sup>

(30) : 第 1 階級左境界値 (startv) の表示。

(32~43) : Do~Loop While 構文。繰り返し条件は右境界値が最大値以下。

(34, 36) : 右境界値と階級値の計算。

(37) : 度数の計算。関数サブルーチン myFreq はデータにおいて指定した数値未満のデータの個数を返すので, myFreq (右境界値) - myFreq (左境界値) で度数が得られる。

(38) : 累積度数。Val 関数の利用は  $i = 2$  の時の対策。

(20) 23 行目以下は正確に記述すれば, .cells (1,1). Value="左境界値"であるが, ここにあるように, . Value は省略可能である。

(39, 40) : 相対度数と累積度数 (%表示) の計算。

(45, 46) : 変数 myRange 2 に相対度数と累積度数の数値領域を代入し, サブルーチン s\_Percent により, %表示にする。サブルーチンは Module 3 にある。

(47~49) : 変数 myRange 2 に度数分布表全体の領域 .Cells(1, 1). CurrentRegion を代入し, サブルーチン s\_表頭 () により表頭の書式設定, サブルーチン s\_罫線 1 により表全体に罫線を引く。サブルーチンは Module 3 にある。

(50, 51) : 変数 myRange 2 に階級値と度数の数値領域を代入し, サブルーチン myHist 2 によりヒストグラムを作成する。サブルーチンは Module 4 にある。

(54~58) : 変数 myRange 2 にヒストグラムを表示したい左上隅のセル番地を代入し, ヒストグラムを移動。

(63~70) : 指定した数値未満のデータの個数を返す関数サブルーチン myFreq。67 行目で検索条件の文字列を作成し, 68 行目でワークシート関数 CountIf により個数を求める。

#### 4. Module 3 : 度数分布表からの基本統計量

このモジュールはワークシート上にある度数分布表 (階級値と度数) から平均, 分散, 標準偏差, 変動係数, 四分位数, 四分位範囲, 四分位偏差, モード, 歪度, 尖度を計算するものである。Excel の「分析ツール」には「基本統計量」コマンドがあるが, それは度数分布表のように分類されたデータに対応したものではない。経済統計分析においては分類されたデータを用いての分析が多くあり, データの分布を知るために分類されたデータの基本統計量を「分析ツール」に頼らず計算する必要がある。また, そのようなケースにおいても平均, 分散の計算がほとんどで, 経済統計データの分布形状の特徴を示す指標である, 四分位数, 歪度, 尖度などが計算されることは希である。このような状況を改善するために作成したのが Module 3 である。UserForm 1 により度数分布表の階級値と度数の数値系列のセル範囲を指定すると, 新しいワークシートに, 基本統計量を表示するものである。Module 3 は宣言セクション (行番号 1~10), プロシージャ BasicStat (行番号 12~148) とプロシージャから呼び

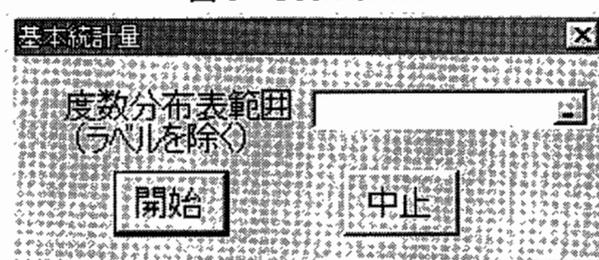
出される7つのサブルーチン（行番号151～）からなる。付録のコードから特徴を見ていく。

(3～10)：すべてのモジュールで利用可能な変数を Public 文で宣言。VBA の特徴的なものとして、4行目で、myRange, myRange 2 を Range 変数(オブジェクト型)として宣言している。<sup>(21)</sup>

(13～19)：プロシージャ BasicStat でのみ使用可能な変数を宣言。

(27)：UserForm 1 を表示する。

図5 UserForm 1



UserForm 1 では、変数 myRange に度数分布表の階級値と度数の数値系列のセル範囲が、変数 x に階級数が代入される。また、新しいワークシート（名前は myWorkSheet）が挿入され、階級値と度数の数値系列がコピーされる。

(32)：新しいワークシート myWorkSheet を選択。

(33, 34)： $xf$  (階級値×度数),  $x^2f$  の計算式の入力。

ワークシート myWorkSheet には1列目に階級値、2列目に度数がそれぞれ1～x行にコピーされている。このデータを用いて3列目に  $xf$  の計算式を代入しているのが33行目である。計算式の入力領域は Range (Cells(1,3), Cells(x,3)) であり、式を R1C1 形式で設定するので FormulaR1C1 プロパティを用いて Range(Cells(1,3), Cells(x,3)).FormulaR1C1 となる。計算式は文字列として代入する。

(36～38)：累積度数, 累積度数 (%表示) の計算式の入力。計算式に変数を含める場合の表記法に注意 (38行目)。

(21) Range 型変数には、116行目のように Set コマンドでセル領域を代入できる。

図6 Module 3 の出力例

	A	B	C	D	E	F	G	H	I	J	K
1											
2	x	f	xf	xxf	z <sup>3</sup> f	z <sup>4</sup> f	累積度数	累積度数%			
3	12	2	24	288	-47.748	137.487	2	2.00%			
4	16	2	32	512	-26.495	62.692	4	4.00%			
5	20	2	40	800	-12.723	23.574	6	6.00%			
6	24	8	192	4608	-19.233	25.765	14	14.00%			
7	28	14	392	10976	-7.900	6.528	28	28.00%			
8	32	14	448	14336	-0.430	0.135	42	42.00%			
9	36	17	612	22032	0.136	0.027	59	59.00%			
10	40	27	1080	43200	9.805	6.995	86	86.00%			
11	44	13	572	25168	23.998	29.438	99	99.00%			
12	48	0	0	0	0.000	0.000	99	99.00%			
13	52	1	52	2704	11.440	25.777	100	100.00%			
14	合計	100	3444	124624	-69.150	318.420					
15											
16											
17											
18											
19											
20											
21											
22											
23											

基本統計量	
度数合計	100
階級数	11
階級幅	4.000
平均	34.440
分散(n)	60.126
分散(n-1)	60.734
標準偏差(n)	7.754
標準偏差(n-1)	7.793
変動係数	0.226
第1四分位数	29.143
中央値(Median)	35.882
第3四分位数	40.370
四分位範囲	11.228
四分位偏差	5.614
最頻値(Mode)	39.667
歪度(Skewness)	-0.713
尖度(Kurtosis)	0.324

(39) : x + 1 行目に列合計の計算式の入力。ワークシート関数はそのまま記述可能である。

(41~49) : データ数 n, xf, xxf, 平均 xmean, 分散 s<sup>2</sup>P, 不偏分散 s<sup>2</sup>, 標準偏差 sP, 不偏標準偏差 s, 変動係数 cv の計算。<sup>(22)</sup>

(51~90) : 歪度, 尖度を除く統計指標の表示。84~86 行目で四分位数を求める関数サブルーチン quartile, 89 行目でモードを求める関数サブルーチン mode を呼び出している。これらのサブルーチンについては後述する。<sup>(23)</sup>

(92~108) : 尖度 Skewness, 歪度 Kurtosis の計算と表示。<sup>(23)</sup>

(22) 47, 48 行目では, 平方根を求める VBA 関数 Sqr() が使用されているが, この文は平方根を求めるワークシート関数 Sqrt() を用いる文とは構文が異なる。Module 1 の 31~33 を参照。

(23) 使用しているアルゴリズムはこれまでの統計指標と同じであるが, 簡単に説明しておく。92 行目で  $((x_i - \bar{x})/s)^3 f_i$ , 94 行目で  $((x_i - \bar{x})/s)^4 f_i$  を求め, 97, 98 行目で  $z^3 f = \sum((x_i - \bar{x})/s)^3 f_i$ ,  $z^4 f = \sum((x_i - \bar{x})/s)^4 f_i$  として, 101, 102 行目で,  $Skewness = z^3 f \times n / (n-1)(n-2)$   $Kurtosis = z^4 f \times n(n+1) / (n-1)(n-2)(n-3) - 3(n-1)^2 / (n-2)(n-3)$  を求めている。正規分では尖度, 歪度ともに 0 である。尖度が 0 より大きい (小さい) 場合, その分布は正規分布に比して裾野が広がっており, 歪度が 0 より大きい (小さい) 場合, その分布は右 (左) に歪んでいることを示している。

(112~143)：計算結果の書式設定など。113~119 行目は書式設定(小数の桁, %)。121~122 行で先頭に 2 行挿入し, 123~130 行でラベルを表示している。132~134 は計算表の表頭処理と罫線処理, 136~142 行は基本統計表の表頭処理と罫線処理である。

(151~166)：第  $k$  四分位数を求める関数サブルーチン `quartile`。度数分布表からの第  $k$  四分位数は「度数分布表から作成したヒストグラムの面積を左から  $k/4$  に分割する  $x$  軸の値」と定義できる。ヒストグラム全体の面積はデータ総数  $n$  で表すことができるので, 面積  $n * k/4$  に対応する値を求めることになる。第  $k$  四分位数が含まれる階級を  $i$  とし, その左境界値を  $stv$ , 1 つ前の階級までの累積度数を  $cumf1$  とおくと, 第  $k$  四分位数  $= stv + 階級幅 * (n * k/4 - cumf1) / 第 i 階級の度数$  となる。このアルゴリズムをコード化したのがサブルーチン `quartile` である。159~162 行の `Do~Loop While` 構文で第  $k$  四分位数が含まれる階級  $i$  を, 163 行で  $stv$  を, 164 行で  $cumf1$  を, そして 165 行で第  $k$  四分位数を求めている。

(169~191)：最頻値 (モード) を求める関数サブルーチン `mode`。モードは最大度数を持つ階級が 1 つある時にのみ存在し, その定義は,

$$mode = stv + \frac{haba \times (f_0 - f_{-1})}{(f_0 - f_{-1}) + (f_0 - f_{+1})}$$

である。ここで,  $stv$ ,  $f_0$  はモードがある階級の左境界値と度数,  $haba$  は階級幅,  $f_{-1}$  は 1 つ下の階級の度数,  $f_{+1}$  は 1 つ上の階級の度数である。関数サブルーチン `mode` では 175 行で変数 `myRange` に度数のセル範囲をセットし, 178 行で最大値 `myMax` を求めている。180 行で最大度数を持つ階級の数調べ, その数が 1 以上であれば, モードは「無し」とし 181 行で最大度数を持つ階級数が 1 つの場合にモードを計算している。

(194~)：書式設定等のためのサブルーチン群。194~199 行が選択範囲の列幅をオートフィットするサブルーチン `s_列 Autofit`, 201~203 行が選択範囲の数値の小数点以下の桁数を 3 桁に揃えるサブルーチン `s_小数`, 206~213 行が選択範囲のセルすべての回りに罫線を引くサブルーチン `s_罫線 1`, 216~222

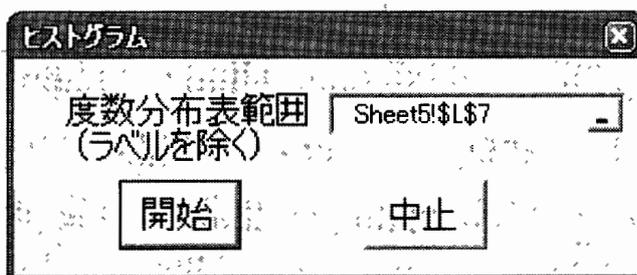
が選択範囲のセルの内容をセンタリングし、セルの背景色を薄緑にするサブルーチン s\_表頭、225~228 が選択範囲の値を小数点以下 2 桁で % 表示するサブルーチン s\_Percent である。これらのサブルーチンは Excel において当該操作を「マクロ記録」し、生成されるコードを修正することにより簡単に作成可能である。<sup>(24)</sup> これらのサブルーチンは他のモジュールからも呼び出し可能である。

## 5. Module 4 : ヒストグラムの作図

Module 2 は分類されていないデータから度数分布表とヒストグラムを作成するものであったが、Module 4 はデータが度数分布表に分類されている場合のヒストグラムの作成を行う。モジュールはプロシージャ myHist 1 (行番号 3~10) とそれから呼び出されるサブルーチン myHist 2 からなる。サブルーチン myHist 2 は Module 2 から呼び出される。

(3~10) : プロシージャ myHist 1。このプロシージャは UserForm 5 で階級値と度数の数値系列のセル範囲を変数 myRange に代入し、myRange を引数としてサブルーチン myHist 2 を呼び出すだけのものである。

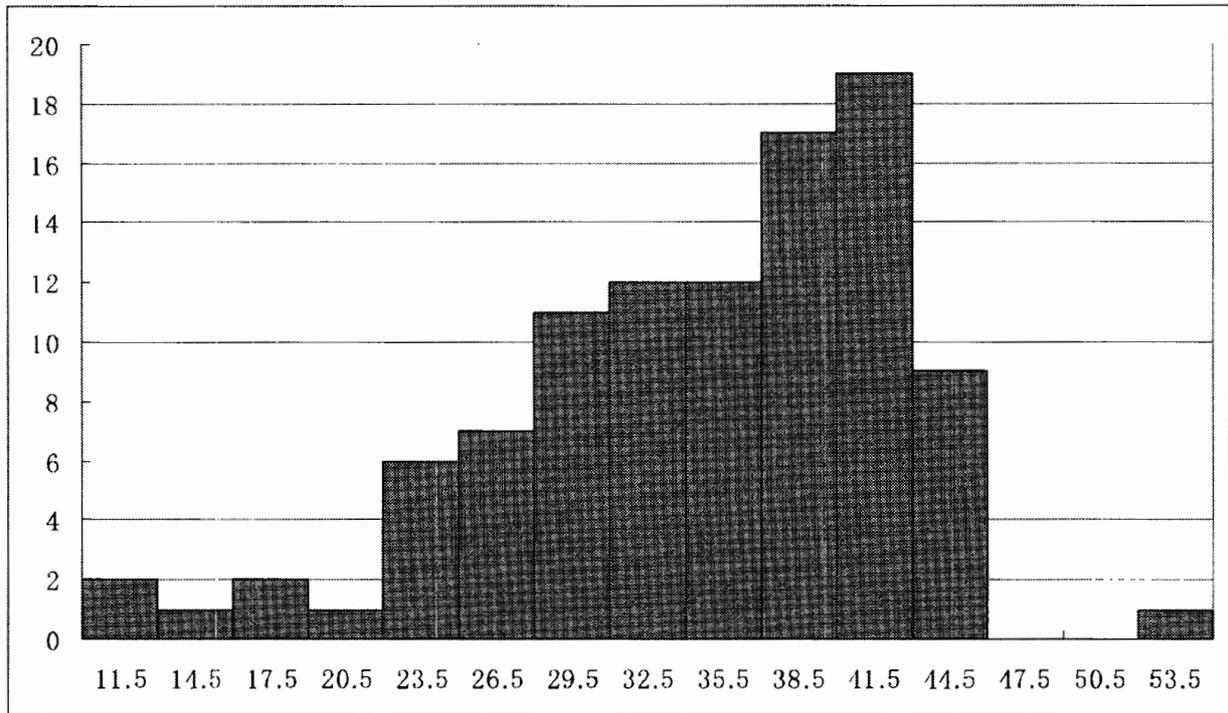
図 7 UserForm 5



(13~45) : myRange のデータからヒストグラムを作成し、それをワークシートに表示する。グラフの作成を VBA でプログラムする際には、ワークシート上でプログラムしたい操作を「マクロ記録」し、生成されたコードを修正して

(24) ただし、「マクロ記録」による生成されるコードは不必要なコードも多く含むので、どのコードが本質的なコードかを見極める必要がある。

図8 Module 4 の出力例



いくのが効率的である。サブルーチン `myHist 2` もそのようにして作成したものであるが重要だと思われる点を以下に述べる。

(21)：新規グラフシートを挿入する。グラフはグラフシートに作成し、必要に応じて、ワークシートに移動する。

(22)：グラフの種類の設定。`xlColumnClustered` が集合棒グラフを表す。

(23)：グラフに使用するデータのセル範囲。この場合、`myRange` である。

`PlotBy := xlColumns` はデータ系列が列方向であることを表す。

(26)：項目軸ラベルとして、`myRange.Columns(1)` = 階級値系列を指定。

(27)：グラフをワークシートへ移動。

(28)：「凡例」を非表示にする。

(29~35)：プロットエリアの書式設定。

(36~42)：グラフ系列の書式設定。

## 6. Module 5：度数分布表の再構成

貯蓄現在高のデータに代表されるように経済統計データにはその分布が大きく歪んでいるものが多くある。このような統計データの度数分布表は各階級の階級幅が同一ではない場合が普通である。

図9 度数分布表の再構成の実例

元の度数分布表			再構成後の度数分布表			
左境界値	右境界値	度数	左境界値	右境界値	階級値	度数
0	100	225	0	100	50	225
100	200	282	100	200	150	282
200	300	290	200	300	250	290
300	400	274	300	400	350	274
400	500	253	400	500	450	253
500	700	408	500	600	550	204
700	1000	409	600	700	650	204
1000	1500	380	700	800	750	136.3333
1500	2000	208	800	900	850	136.3333
2000	4400	456	900	1000	950	136.3333
			1000	1100	1050	76
			:	:	:	:
			:	:	:	:
			3900	4000	3950	19
			4000	4100	4050	19
			4100	4200	4150	19
			4200	4300	4250	19
			4300	4400	4350	19

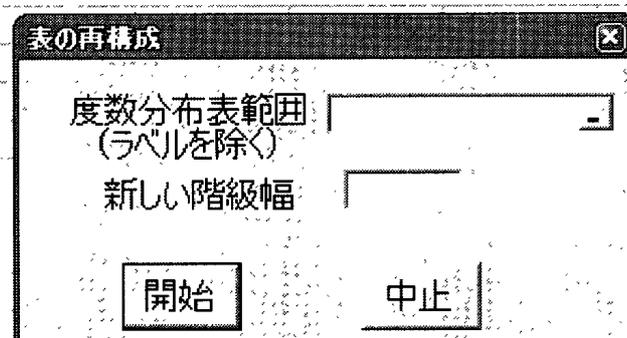
左の「元の度数分布表」では第5階級までは階級幅は100であるが、第6階級では200、第7階級では300、第8、9階級では500、最後の階級では2400<sup>(25)</sup>となっている。このような度数分布表からExcelで分布形状を正しく表すヒストグラムを描くには最小の階級幅を用いて度数分布表を再構成する必要がある。この作業を自動化したのがModule 5である。

モジュールはプロシージャreconstと、その中で呼び出される関数サブルーチンmyCumfからなる。付録のコードから特徴を見ていく。

(25) 例示の最終階級の階級幅が異常に大きいと思うかも知れないが、所得データや資産データをはじめ、経済統計データにおいては最終階級の階級幅が非常に広いケースは多くある。

(10) : UserForm 2 を表示する。

図 10 UserForm 2



UserForm 2 では度数分布表の左境界値、右境界値および度数の数値系列のセル範囲<sup>(26)</sup>が変数 `myRange` に、再構成後の階級幅が変数 `haba` に代入される。さらに、新しいワークシートが挿入され、`myRange` のデータがコピーされる。

(18, 19) : 新しいシートの第 1 列には左境界値系列、第 2 列には右境界値系列、第 3 列には度数系列がコピーされているので、4 列目に累積度数を計算する。

(23~36) : 再構成後の度数分布表の作成。第 7 列に左境界値、第 8 列に右境界値、第 9 列に階級値を表示する。30 行目は `c1` : 左境界値、`c2` : 右境界値、`myCumf` : 指定した値までのデータ数を求める関数サブルーチン、を用いて度数を計算している。

(37~49) : 飾りつけと不要となる列の削除。

(56~69) : 引数で与えられた数値までのデータ数を計算する。<sup>(27)</sup>

## 7. アドインと新規メニューの追加

作成されたプロシージャは「Visual Basic」ツールバーから「マクロ実行」ボタンで表示される一覧表から実行したいマクロ (プロシージャ) を選択し、「実

(26) 指定する系列が UserForm 1 と異なることに注意が必要である。

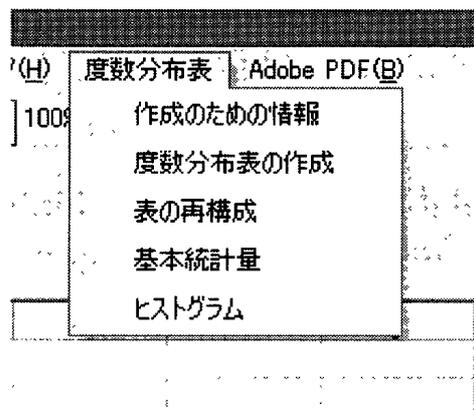
(27) 作業内容は Module 2 の `myFreq` と同じであるが、`myFreq` が分類されていないデータからの計算であるためワークシート関数 `CountIF` 利用の単純なものであるのに比して度数分布表からの計算なので少し複雑になっている。アルゴリズムは Module 3 における四分位数を求める関数サブルーチン `quartile` と同様なものである。

行」ボタンをクリックすることにより実行可能である。しかし、Excel のマクロはアドインファイルとして配布される場合が多い。この理由は、アドイン形式のファイルは利用者によってコードが改変されたり、意図しないモジュールが実行されたりする心配がないためである。

このようにアドイン形式のファイル（以下、アドインと略）は作成者にとってはプログラムの配布方法として優れたものであるが、アドインに含まれるプロシージャは「Visual Basic」ツールバーからは実行できなくなる。そのため、アドインには Excel に組み込んだ時、アドインに含まれるプロシージャをメニューバーやツールバーに登録するプロシージャを作成しておく必要がある<sup>(29)</sup>。

ここでは、メニューバーへ「度数分布」というポップアップ・メニューを登録し、その下に5つのメニュー」を配置している。

図 11 メニューバーへの追加



付録にあるコードでその特徴をしてみる。モジュールはこのアドインがインストールされた時の処理を記述したイベントプロシージャ `Workbook_AddinUninstall()` とアンインストールされた時の処理を記述したイベントプロシージャ `Workbook_AddinUninstall()` からなっている。

(8) : メニューバーを変数 `myCB` に代入。4 行目で `myCB` はオブジェクト型

(28) VBA はインタプリタ言語であるが、アドインファイルは FORTRAN などコンパイル言語における実行形式ファイルのようなものと考えられることができる。

(29) プロシージャは「Microsoft Excel Objects」の `ThisWorkbook` モジュールに作成する。

の CommanBar 変数として宣言している。

(10)：変数 myCB に Popup タイプのメニューを追加し、追加したメニューを myCBCtrl とする。

(11)：変数 myCBCtrl のラベルを「度数分布」とする。

(13)：変数 myCBCtrl に Button タイプのメニューを追加し、追加したメニューを myCBCtrl 2 とする。

(14)：変数 myCBCtrl 2 のラベルを「作成のための情報」とする。

(16)：メニューの「作成のための情報」がクリックされたら、プロシージャ myFreq 1 を実行する。

(18~36)：残り 4 つのプロシージャについて、13~16 と同様な作業の繰り返し。

(39~41)：このアドインがアンインストールされた時には、メニューバーを初期状態に戻す。

## 8. アドインの利用

アドインを利用するためには Excel に組み込む必要がある。

- ① アドインを C：¥Documents and Settings¥（ユーザー名）¥Application Data¥Microsoft¥AddIns<sup>(30)</sup> に置く。
- ② Excel のメニューから「ツール」-「アドイン」を選択すると、①のフォルダにあるアドインの一覧が表示される<sup>(31)</sup>ので、目的のアドインをチェックし、「OK」ボタンをクリックする。
- ③ 目的のアドインがメニューバーに追加される。なお、アドインをアンインストールするには、②で表示されるアドイン一覧においてアンインストールするアドインのチェックを外す。

---

(30) このパスは WindowsXP の場合である。Windows 98 では、C：¥Windows¥Application Data¥Microsoft¥AddIns となる。

(31) ①の AddIns フォルダ以外にアドインがある場合には、「参照」ボタンをクリックする。

### Ⅲ ま と め

前節では Excel 2002 VBA によるプログラムの実例によりその特徴を見てきた。それからわかる ExcelVBA の特徴は通常のプログラム言語が持つ、変数の宣言、変数処理、分岐処理、繰り返し処理に加えて、

- ① オブジェクト、プロパティ、メソッドを用いた構文
- ② 「マクロ記録」によるコードの作成
- ③ ユーザーフォームによるダイアログボックス
- ④ アドイン形式による Excel への組み込み

を通じて Excel との密接な連携をとることができる点にある。

#### ① オブジェクト、プロパティ、メソッドを用いた構文

付録のコードを見ると、コードの大部分がオブジェクト、プロパティ、メソッドというキーワードを組み合わせて命令を実行していることがわかる。これらの構文は大きく次の3つに分類可能である。

- ・オブジェクトのプロパティの値を取得する構文
- ・オブジェクトのプロパティ値を設定する構文
- ・メソッドでオブジェクトを操作する構文

VBA のコードはオブジェクト、プロパティ、メソッドの名前が比較的長いので、プログラムの解読が難しいと思われがちであるが、この3つの基本構文を理解すれば、極めて分かり易いコードであると言える。<sup>(32)</sup> オブジェクト、プロパティ、メソッドの入力は VBE (Visual Basic Editor) における「自動クイックヒント」と「自動メンバ表示」を利用すれば効率的に行うことができる。

#### ② 「マクロ記録」によるコードの自動作成

Excel を用いてデータ分析を行う場合、個々の作業の細かい点（例えば対象とするデータ範囲）は異なるが作業内容自体は同じ操作を繰り返ししなくては

(32) 前節で示したプログラムにおいても、解読が難しいのは VBA の基本構文ではなく、筆者の独自のアルゴリズムをコード化した部分である。

ならないケースは多々ある。このようなケースにおいて Excel の「マクロ記録」機能で作業のコア部分のコードを自動生成し、必要な修正を施すのみで目的とするプログラムを作成可能である。既知のプログラム言語でも、何もないところから目的とするプログラムを作成することが困難な場合が多いので、「マクロ記録」によるコードの自動作成を利用できる点は VBA の大きな特徴といえる。

### ③ ユーザーフォームによるダイアログボックス

プログラムの使い勝手の良さは入力環境の良し悪しにも依存している。その点で、Windows 風のダイアログボックスを作成可能な VBA はユーザーに違和感なく受け入れられる。通常のプログラムにおける入力は数種のコントロールで対応可能<sup>(33)</sup>であり、前節のプログラムで見たように1つユーザーフォームを作成するとそれを修正することにより、簡単に別の目的のユーザーフォームは作成できる。

### ④ アドイン形式による Excel への組み込み

Excel の多くのユーザーは何らかの形でアドインを使うであろう。特に統計分析を行っている者で「分析ツール」の存在をしらない者は皆無であろう。作成したプログラムをアドイン形式で配布することは作成者、利用者、双方にとっても有益なものである。

Excel は現時点で最も利用範囲の広いソフトウェアである。経済統計分析においても Excel を利用することにより分析を効率的に行うことができる。しかし、Excel 自体は決して経済統計分析の専用ソフトではないので、経済統計データに特有な分析を行うためにはある種の工夫が要求される。しかし、要求される工夫が面倒なケースもあり、そのことが分析内容を通常の Excel 操作で可

---

(33) 本稿のユーザーフォームでは、ラベル、RefEdit、コマンドボタンの他にテキストボックスのコントロールを使用している。その他のよく利用されるコントロールとしては、2つの状態を切り替えるチェックボックス、複数の選択肢から1つを選択するオプションボタン、リストボックスがある。

能な範囲に限定してしまう傾向もある。ExcelVBAはExcelに付属しているプログラム言語であるが、それを利用することによりExcelをより便利な経済統計分析の専用のソフトとして利用可能となるのである。また、完全なプログラムでなくても、Excelの「マクロ記録」で作成されたコードを修正して利用したり、単に、入手可能なアドインをExcelに組み込むという形でExcelVBAを利用するだけでもデータ分析にとっては有益であろう。

### 参 考 文 献

- エクスメディア(1999),『EXCEL関数活用 SUPER MASTER』,エクスメディア  
大村あつし(2002),『かんたんプログラミング Excel 2002 VBA 基礎編』,技術評論社  
大村あつし(2002),『かんたんプログラミング Excel 2002 VBA 応用編』,技術評論社  
大村あつし(2002),『かんたんプログラミング Excel 2002 VBA コントロール・関数編』,技術評論社  
大村あつし(2002),『ExcelVBAによる Win 32 API プログラミング入門』,エーアイ出版  
大村あつし・牧村あきこ(2002),『かんたんプログラミング Excel 2002 VBA 演習編』,技術評論社  
大村あつし・渡辺ひかる(2003),『ExcelVBA 実用サンプルコレクション』,ソフトバンクパブリッシング  
山賀弘(2002),『Excel 2002 VBA マグナム辞典』,技術評論社

(付録) コード

UserForm におけるコードは類似しているため、UserForm 3 のコードのみを記載している。

#### Module 1

```
1      Option Explicit
2
3      Sub myFreq 1 ()
4
5          Dim myBtn As Integer
6          Dim hani As Double, starjes As Double
7
8          myTitle = "度数分布表作成のための情報"
9          myDataSheet = ActiveSheet.Name
10
11         UserForm 3.Show
12         If Cb 10 = True Then Exit Sub
13
14         Application.ScreenUpdating = False
15
16         x = myRange.Rows.Count      '行数
17         y = myRange.Columns.Count   '列数
18
19         With myRange 2
20             .Cells(1, 1).Value = "標本数"
21             .Cells(2, 1).Value = "最大値"
22             .Cells(3, 1).Value = "最小値"
23             .Cells(4, 1).Value = "範囲"
24             .Cells(5, 1).Value = "範囲/10"
25             .Cells(6, 1).Value = "範囲/20"
26             .Cells(7, 1).Value = "範囲/30"
27             .Cells(8, 1).Value = "    Starjes の階級数"
28             .Cells(9, 1).Value = "    Starjes の階級幅"
29         End With
30
31         n = Application.WorksheetFunction.CountA(myRange)
32         myMax = Application.WorksheetFunction.Max(myRange)
33         myMin = Application.WorksheetFunction.Min(myRange)
34         hani = (myMax + myMin) / 2
35         starjes = 1 + Log(n) / Log(2)
36
37         With myRange 2
38             .Cells(1, 2).Value = n
39             .Cells(2, 2).Value = myMax
40             .Cells(3, 2).Value = myMin
41             .Cells(4, 2).Value = hani
```

```
42         .Cells(5, 2).Value=hani/10
43         .Cells(6, 2).Value=hani/20
44         .Cells(7, 2).Value=hani/30
45         .Cells(8, 3).Value=starjes
46         .Cells(9, 3).Value=hani / starjes
47         Call s_小数(Range(.Cells(4, 2), .Cells(9, 3)))
48     End With
49
50         Application.ScreenUpdating=True
51
52 End Sub
```

## UserForm 3

```
1     Option Explicit
2     Private Sub CommandButton1_Click()
3         If RefEdit1.Value=""Then
4             MsgBox"データ範囲が選択されていません"
5             Exit Sub
6         ElseIf RefEdit2.Value=""Then
7             MsgBox"出力先が選択されていません"
8             Exit Sub
9         Else
10            Set myRange=Range(RefEdit1.Value)
11            Set myRange2=Range(RefEdit2.Value)
12        End If
13        Cb10=False
14        UserForm3.Hide
15    End Sub
16
17
18    Private Sub CommandButton2_Click()
19        Cb10=True
20        UserForm3.Hide
21    End Sub
22
23
24    Private Sub UserForm_Initialize()
25        RefEdit1.SetFocus
26    End Sub
27
28
29    Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
30        Cb10=True
31        UserForm3.Hide
32    End Sub
```

## Module 2

```

1      Option Explicit
2
3      Sub myFreq 2()
4
5          Dim c 1 As Double, c 2 As Double
6          Dim i As Long
7
8          myTitle="度数分布表の作成"
9          myDataSheet=ActiveSheet.Name
10
11         UserForm 4.Show
12         If Cb 10=True Then Exit Sub
13
14         Application.ScreenUpdating=False
15
16         x=myRange.Rows.Count      '行数
17         y=myRange.Columns.Count    '列数
18         n=Application.WorksheetFunction.Count(myRange)
19         Worksheets(myWorkSheet).Select
20
21         Cells(x+3,2).Select
22         With Selection
23             .Cells(1,1)="左境界値"
24             .Cells(1,2)="右境界値"
25             .Cells(1,3)="階級値"
26             .Cells(1,4)="度数"
27             .Cells(1,5)="累積度数"
28             .Cells(1,6)="相対度数"
29             .Cells(1,7)="累積度数%"
30             .Cells(2,1)=startv
31             i=2
32             Do
33                 c 1=.Cells(i,1)
34                 .Cells(i,2)=c 1+cwidth
35                 c 2=.Cells(i,2)
36                 .Cells(i,3)=(c 1+c 2)/2
37                 .Cells(i,4)=myFreq(c 2)-myFreq(c 1)
38                 .Cells(i,5)=Val(.Cells(i-1,5))+.Cells(i,4)
39                 .Cells(i,6)=.Cells(i,4)/n
40                 .Cells(i,7)=.Cells(i,5)/n
41                 i=i+1
42                 .Cells(i,1)=c 2
43             Loop While c 2<=myMax
44             .Cells(i,1)=""
45             Set myRange 2=Range(.Cells(2,6),.Cells(i-1,7))

```

```

46         Call s_Percent(myRange 2)
47         Set myRange 2=.Cells(1,1).CurrentRegion
48         Call s_表頭(myRange 2)
49         Call s_罫線 1(myRange 2)
50         Set myRange 2=Range(.Cells(2,3),.Cells(i-1,4))
51         Call myHist 2(myRange 2)
52     End With
53
54     Set myRange 2=Range(Cells(x+i+1,2), Cells(x+i+1,2))
55     With ActiveSheet.ChartObjects(1)
56         .Top=myRange 2.Top
57         .Left=myRange 2.Left
58     End With
59
60 End Sub
61
62
63 Function myFreq(c As Double) As Long
64
65     Dim myStr As String
66
67     myStr="<"&c
68     myFreq=Application.WorksheetFunction.CountIf(myRange, myStr)
69
70 End Function

```

## Module 3

```

1     Option Explicit
2
3     Public myMsg As String, myTitle As String
4     Public myRange As Range, myRange 2 As Range
5     Public x As Integer, y As Integer
6     Public myDataSheet As String, myWorkSheet As String
7     Public Cb 10 As Integer
8     Public n As Long
9     Public myMax As Double, myMin As Double
10    Public startv As Double, cwidth As Double
11
12    Sub BasicStat()
13        Dim xmean As Double, cv As Double
14        Dim s 2 P As Double, s 2 As Double
15        Dim sP As Double, s As Double
16        Dim xf As Double, xxf As Double
17        Dim z 3 f As Double, z 4 f As Double
18        Dim sx 3 As Double, sx 4 As Double

```

```

19 Dim skewness As Double, kurtosis As Double
20
21 myTitle = "基本統計量"
22 myDataSheet = ActiveSheet.Name
23 ActiveSheet.Select
24
25 '度数分布表の選択
26
27 UserForm1.Show
28 If Cb10 = True Then Exit Sub
29
30 Application.ScreenUpdating = False
31
32 Worksheets(myWorksheet).Select
33 Range(Cells(1, 3), Cells(x, 3)).FormulaR1C1 = "=RC[-2]*RC[-1]"
34 Range(Cells(1, 4), Cells(x, 4)).FormulaR1C1 = "=RC[-3]*RC[-1]"
35
36 Cells(1, 7).Value = Cells(1, 2).Value
37 Range(Cells(2, 7), Cells(x, 7)).FormulaR1C1 = "=R[-1]C+RC[-5]"
38 Range(Cells(1, 8), Cells(x, 8)).FormulaR1C1 = "=RC[-1]/R"&x+1&"C2"
39 Range(Cells(x+1, 2), Cells(x+1, 6)).FormulaR1C1 = "=SUM(R["&-x&"C:R[-1]C)"
40 Cells(x+1, 1).Value = "合計"
41 n = Cells(x+1, 2).Value
42 xf = Cells(x+1, 3).Value
43 xxf = Cells(x+1, 4).Value
44 xmean = xf/n
45 s2P = (xxf - xf * xf/n)/n
46 s2 = (xxf - xf * xf/n)/(n-1)
47 sP = Sqr(s2P)
48 s = Sqr(s2)
49 cv = s/xmean
50
51 With Cells(1, 10)
52 .Cells(1, 1).Value = "基本統計量"
53 .Cells(2, 1).Value = "度数合計"
54 .Cells(3, 1).Value = "階級数"
55 .Cells(4, 1).Value = "階級幅"
56
57 .Cells(6, 1).Value = "平均"
58 .Cells(7, 1).Value = "分散 (n)"
59 .Cells(8, 1).Value = "分散 (n-1)"
60 .Cells(9, 1).Value = "標準偏差 (n)"
61 .Cells(10, 1).Value = "標準偏差 (n-1)"
62 .Cells(11, 1).Value = "変動係数"
63
64 .Cells(13, 1).Value = "第1四分位数"
65 .Cells(14, 1).Value = "中央値 (Median)"

```

```

66      .Cells(15, 1).Value="第3四分位数"
67      .Cells(16, 1).Value="四分位範囲"
68      .Cells(17, 1).Value="四分位偏差"
69      .Cells(18, 1).Value="最頻値(Mode)"
70
71      .Cells(20, 1).Value="歪度(Skewness)"
72      .Cells(21, 1).Value="尖度(Kurtosis)"
73
74      .Cells(2, 2).Value=n
75      .Cells(3, 2).Value=x
76      .Cells(4, 2).Value=.Cells(2, 1).Value-.Cells(1, 1).Value
77      .Cells(6, 2).Value=xmean
78      .Cells(7, 2).Value=s 2 P
79      .Cells(8, 2).Value=s 2
80      .Cells(9, 2)=sP
81      .Cells(10, 2)=s
82      .Cells(11, 2)=cv
83
84      .Cells(13, 2).Value=quartile(1)
85      .Cells(14, 2).Value=quartile(2)
86      .Cells(15, 2).Value=quartile(3)
87      .Cells(16, 2).Value=.Cells(15, 2).Value-.Cells(13, 2).Value
88      .Cells(17, 2).Value=.Cells(16, 2)/2
89      .Cells(18, 2).Value=mode()
90      End With
91
92      Range(Cells(1, 5), Cells(x, 5)).FormulaR1C1="(((RC[-4]"&-xmean&")/"&_
93      s&")^3)*RC[-3]"
94      Range(Cells(1, 6), Cells(x, 6)).FormulaR1C1="(((RC[-5]"&-xmean&")/"&_
95      s&")^4)*RC[-4]"
96
97      z3f=Cells(x+1, 5).Value
98      z4f=Cells(x+1, 6).Value
99
100     If n>3000 Then GoTo myEnd
101     skewness=n * z3f/((n-1)*(n-2))
102     kurtosis=n*(n+1)*z4f/((n-1)*(n-2)*(n-3))_
103     -3*(n-1)^2/((n-2)*(n-3))
104
105     With Cells(1, 10)
106         .Cells(20, 2).Value=skewness
107         .Cells(21, 2).Value=kurtosis
108     End With
109
110
111     myEnd:
112     '飾り付け

```

```
113 Set myRange=Range(Cells(4, 11), Cells(21, 11))
114 Call s_小数(myRange)
115 Set myRange=Range(Cells(1, 5), Cells(x+1, 6))
116 Call s_小数(myRange)
117
118 Set myRange=Range(Cells(1, 8), Cells(x, 8))
119 Call s_Percent(myRange)
120
121 Rows("1:2").Select
122 Selection.Insert Shift:=xlDown
123 Cells(2, 1).Value="x"
124 Cells(2, 2).Value="f"
125 Cells(2, 3).Value="xf"
126 Cells(2, 4).Value="xxf"
127 Cells(2, 5).Value="z ^ 3 f"
128 Cells(2, 6).Value="z ^ 4 f"
129 Cells(2, 7).Value="累積度数"
130 Cells(2, 8).Value="累積度数%"
131
132 Set myRange=Range(Cells(2, 1), Cells(x+3, 8))
133 Call s_表頭(myRange)
134 Call s_罫線1(myRange)
135
136 Range(Cells(3, 10), Cells(3, 11)).Merge
137
138 Set myRange=Range(Cells(3, 10), Cells(23, 11))
139 Call s_表頭(myRange)
140 Call s_罫線1(myRange)
141 Columns("A:K").Select
142 Selection.Columns.AutoFit
143 Cells(1, 1).Select
144 Application.DisplayAlerts=True
145
146 Exit Sub
147
148 End Sub
149
150
151 Function quartile(k As Integer) As Double
152
153 Dim haba As Double, stv As Double
154 Dim cumf As Double, i As Integer
155 Dim cumf1 As Double
156
157 haba=Cells(2, 1).Value-Cells(1, 1).Value
158 cumf=0:i=0
159 Do
```

```

160         i=i+1
161         cumf=cumf+Cells(i,2).Value
162     Loop While cumf <= n * k/4
163     stv=Cells(i,1).Value-haba/2
164     cumf1=cumf-Cells(i,2).Value
165     quartile=stv+haba*(n * k/4-cumf1)/Cells(i,2).Value
166 End Function
167
168
169 Function mode() As Variant
170     Dim haba As Double, stv As Double
171     Dim myMax As Double, i As Integer
172     Dim myStr As String
173
174     haba=Cells(2,1).Value-Cells(1,1).Value
175     Set myRange=Range(Cells(1,2), Cells(x,2))
176
177     With Application.WorksheetFunction
178         myMax=.Max(myRange)
179         myStr=Str(myMax)
180         If .CountIf(myRange, myStr) > 1 Then
181             mode="なし"
182         ElseIf .Match(myMax, myRange, 0) < 2 Then
183             mode=Cells(1,1).Value
184         Else
185             i=.Match(myMax, myRange, 0)
186             stv=Cells(i,1).Value-haba/2
187             mode=stv+(haba*(Cells(i,2).Value-Cells(i-1,2).Value))/_
188                 (2*Cells(i,2).Value-Cells(i-1,2).Value-Cells(i+1,2).Value)
189         End If
190     End With
191 End Function
192
193
194 Sub s_列 Autofit(myRange As Range)
195     With myRange
196         Range(.Columns(1), .Columns(.Columns.Count))._
197             Columns.AutoFit
198     End With
199 End Sub
200
201 Sub s_小数(myRange As Range)
202     myRange.NumberFormatLocal="0.000"
203 End Sub
204
205
206 Sub s_罫線1(myRange As Range)

```

```
207 myRange.Borders(xlEdgeLeft).Weight = xlThin
208 myRange.Borders(xlEdgeTop).Weight = xlThin
209 myRange.Borders(xlEdgeBottom).Weight = xlThin
210 myRange.Borders(xlEdgeRight).Weight = xlThin
211 myRange.Borders(xlInsideVertical).Weight = xlThin
212 myRange.Borders(xlInsideHorizontal).Weight = xlThin
213 End Sub
214
215
216 Sub s_表頭(myRange As Range)
217     With myRange.Rows(1)
218         .Interior.ColorIndex = 35
219         .Borders(xlEdgeBottom).Weight = xlMedium
220         .HorizontalAlignment = xlCenter
221     End With
222 End Sub
223
224
225 Sub s_Percent(myRange As Range)
226     myRange.Style = "Percent"
227     myRange.NumberFormatLocal = "0.00%"
228 End Sub
```

#### Module 4

```
1 Option Explicit
2
3 Sub myHist 1()
4
5     UserForm 5.Show
6     If Cb 10 = True Then Exit Sub
7
8     Call myHist 2(myRange)
9
10 End Sub
11
12
13 Sub myHist 2(myRange As Range)
14
15     Dim myBook As String, myGSheet As String
16
17     x = myRange.Rows.Count
18     myBook = ActiveWorkbook.Name
19     myGSheet = ActiveSheet.Name
20
21     Charts.Add
```

```

22     ActiveChart.ChartType = xlColumnClustered
23     ActiveChart.SetSourceData Source:=myRange, PlotBy _
24         :=xlColumns
25     ActiveChart.SeriesCollection(1).Delete
26     ActiveChart.SeriesCollection(1).XValues = myRange.Columns(1)
27     ActiveChart.Location Where:=xlLocationAsObject, Name:=myGSheet
28     ActiveChart.HasLegend = False
29     ActiveChart.PlotArea.Select
30     With Selection.Border
31         .ColorIndex = 16
32         .Weight = xlThin
33         .LineStyle = xlContinuous
34     End With
35     Selection.Interior.ColorIndex = xlNone
36     ActiveChart.SeriesCollection(1).Select
37     With ActiveChart.ChartGroups(1)
38         .Overlap = 0
39         .GapWidth = 0
40         .HasSeriesLines = False
41         .VaryByCategories = False
42     End With
43     Windows(myBook).Activate
44
45 End Sub

```

## Module 5

```

1     Option Explicit
2     Public haba As Double
3
4     Sub reconst()
5         Dim c 1 As Double, c 2 As Double
6         Dim i As Integer
7
8         ' myTitle = "度数分布表の再構成"
9
10        UserForm 2.Show
11        If Cb 10 = True Then Exit Sub
12
13        Application.ScreenUpdating = False
14
15        Set myRange = Cells(1, 1).CurrentRegion
16        x = myRange.Rows.Count
17
18        Cells(1, 4).Value = Cells(1, 3).Value
19        Range(Cells(2, 4), Cells(x, 4)).Formula = "=R[-1]C+RC[-1]"

```

```
20
21     Cells(1, 7) = Cells(1, 1).Value
22
23     i = 1
24     Do
25         Cells(i, 8).Value = Cells(i, 7).Value + haba
26         c 1 = Cells(i, 7).Value
27         c 2 = Cells(i, 8).Value
28         Cells(i, 9).Value = (c 1 + c 2) / 2
29         If c 2 > Cells(x, 2).Value Then c 2 = Cells(x, 2).Value
30         Cells(i, 10).Value = myCumf(c 2) - myCumf(c 1)
31         If c 2 >= Cells(x, 2).Value Then GoTo myEnd
32         i = i + 1
33         Cells(i, 7) = Cells(i - 1, 7) + haba
34     Loop While c 2 <= Cells(x, 2).Value
35
36 myEnd:
37     Rows("1:2").Select
38     Selection.Insert Shift:=xlDown
39     Cells(2, 7).Value = "左境界値"
40     Cells(2, 8).Value = "右境界値"
41     Cells(2, 9).Value = "階級値"
42     Cells(2, 10).Value = "度数"
43
44     Set myRange = Cells(2, 7).CurrentRegion
45     Call s_表頭(myRange)
46     Call s_罫線 1(myRange)
47
48     Columns("A:E").Delete
49     Cells(1, 1).Select
50
51     Application.ScreenUpdating = True
52
53 End Sub
54
55
56 Function myCumf(c As Double) As Double
57     Dim i As Integer
58     Dim cum 2 As Double
59     i = 0
60     Do
61         i = i + 1
62     Loop While Cells(i, 2) < c
63     cum 2 = Cells(i, 3) * (c - Cells(i, 1)) / (Cells(i, 2) - Cells(i, 1))
64     If i = 1 Then
65         myCumf = cum 2
66     Else
```

```
67         myCumf=Cells(i-1,4)+cum 2
68     End If
69 End Function
```

This Workbook モジュール

```
1     Option Explicit
2
3     Private Sub Workbook_AddinInstall()
4         Dim myCB As CommandBar
5         Dim myCBCtrl As CommandBarControl
6         Dim myCBCtrl 2 As CommandBarControl
7
8         Set myCB=Application.CommandBars("Worksheet Menu Bar")
9
10        Set myCBCtrl=myCB.Controls.Add(Type:=msoControlPopup)
11        myCBCtrl.Caption="度数分布表"
12
13        Set myCBCtrl 2=myCBCtrl.Controls.Add(Type:=msoControlButton)
14        myCBCtrl 2.Caption="作成のための情報"
15        myCBCtrl 2.Style=msoButtonIconAndCaption
16        myCBCtrl 2.OnAction="myFreq 1"
17
18        Set myCBCtrl 2=myCBCtrl.Controls.Add(Type:=msoControlButton)
19        myCBCtrl 2.Caption="度数分布表の作成"
20        myCBCtrl 2.Style=msoButtonIconAndCaption
21        myCBCtrl 2.OnAction="myFreq 2"
22
23        Set myCBCtrl 2=myCBCtrl.Controls.Add(Type:=msoControlButton)
24        myCBCtrl 2.Caption="表の再構成"
25        myCBCtrl 2.Style=msoButtonIconAndCaption
26        myCBCtrl 2.OnAction="reconst"
27
28        Set myCBCtrl 2=myCBCtrl.Controls.Add(Type:=msoControlButton)
29        myCBCtrl 2.Caption="基本統計量"
30        myCBCtrl 2.Style=msoButtonIconAndCaption
31        myCBCtrl 2.OnAction="BasicStat"
32
33        Set myCBCtrl 2=myCBCtrl.Controls.Add(Type:=msoControlButton)
34        myCBCtrl 2.Caption="ヒストグラム"
35        myCBCtrl 2.Style=msoButtonIconAndCaption
36        myCBCtrl 2.OnAction="myHist 1"
37    End Sub
38
39    Private Sub Workbook_AddinUninstall()
40        Application.CommandBars("Worksheet Menu Bar").Reset
41    End Sub
```