香川大学経済論叢
第67巻 第3・4号 1995年2月 143-181

経済分析とコンピュータ(1)

——RATSを中心としたシステム——

大野拓行

I はじめに

筆者が初めてコンピュータ（電卓は別として）に触れたのは，学生時代におけるFORTRANの実習においてであったと思う。その当時は，カード穿孔機を用いて，プログラムとデータを用意してバッチジョブを依頼していた。最初に作成したプログラムは簡単な四則演算のプログラムであったが，自分の作成したプログラムのアウトプットを手にしたときの感激は新鮮なものであった。

時代は流れ，現在では別にプログラム言語を知らなくても，手軽に高度な統計的分析が可能な状況になってきている。しかし，振り返ってみると，最初に修得したFORTRANの知識は現在に至るまで，筆者の研究・教育において重要な位置を占めていることが分かってきた。単発的な統計分析においては，目的に合ったパッケージ・ソフトを探しだし，データはキーボードから入力して，必要とする結果さえ得られればそれでよい。しかし，継続的な研究や学生に対する統計教育を考えると，単に，パッケージ・ソフトを使いこなすだけでなく，その周辺のこと，例えば，ソフト間での結果のやり取り，データベースの利用，結果のプレゼンテーションなど，経済分析をシステムとして捉える必要がある。その際，既存のリソースを結びつけるのにプログラミング言語の知識は重要である。

大野（1987）は，パソコンにおいて現在ほど経済分析用ソフトが一般的でなかった時期において，プログラム言語CとMS—DOSのバッチ処理の特性を活かして，比較的簡単で，拡張性のある経済分析システムを提唱したものであった。しかし，その後，TSPやRATS等に代表されるように，パソコンでも，高

度な統計的手法が利用できるようになり，筆者も自分でプログラムを書くのではなく，それらのソフトを利用して，研究・教育を行ってきた。ここでは，筆者が，ここ数年，研究・教育に活用してきたRATSを中心としたシステムについて解説していきたい。

学術情報センターが行った『統計データに関する利用動向及び需要調査　報告書』(1994)によると，統計分析処理の約80%がパソコンでなされており，使用ソフトとしては，表計算ソフトLotus 1-2-3が55.7%を占めており，次いで，個人作成ソフト，SAS，SPSSと続き，TSPも第7位に入っている（残念ながら，RATSはベスト10には入っていない)。

松本（1994）にもあるように，経済分析用のソフトは種々なものがあるが，回帰分析を中心としたソフトで，パソコンで最も広く利用されているものはTSPであろう。筆者の実感から言えば，通常の研究・教育に使用される機能面からは，TSP，RATSもそれほど差があるとは思えないが，日本での利用を考えた場合，TSPが日本語マニュアルが完備されているのに対して，RATSは未だに日本語マニュアルが存在していない点が，その利用頻度の差に大きく影響していると思われる。(1)

筆者がTSPではなく，RATSを中心としたシステムを構築した理由は主にそのハード面の制約のためであった。すなわち，以前のバージョンのTSPにおいては，数値演算コプロセッサが必須(2)であったし，メモリ管理の制約より日本語FEPとの同居が不可能であった。その点RATSは数値演算コプロセッサも必要とせず，日本語FEPとの同居も可能であり，しかも1.2 MBのフロッピィ・ディスクにプログラム，エディター，日本語辞書も格納できるコンパクト性は，フロッピィベースでの研究・教育を行う際には，大変，便利である。

ここで，紹介するシステムは，RATS，Vzエディター，Lotus 1-2-3，東洋

(1)　また，TSPはNECのPC 98シリーズ版がリリースしているのに対して，RATSはそのままでは，PC 98シリーズでは完全には動作しないことも利用頻度の差になっていると思われる。しかし，この問題はPDS (Public Domain Software) であるSIMを利用することにより，解決できる。

(2)　TSPの最新バージョンである4.2 Bでは，この制約は解消している。

経済新報社のマクロデータファイルを組み合わせたシステムであり，先に述べたようにRATSの日本語マニュアルがない現状を考えて，RATSの簡単なマニュアルにもなっている。(3)

II RATSによる回帰分析

RATS (Regression Analysis of Time Series) はVAR ECONOMETRICS社製の経済分析用ソフトである。本節では，簡単な消費関数の例でRATSの基本的なコマンドを見ていくことにする。

1. RATSプログラムの概要

（プログラム1）にRATSのプログラムを掲載している。

- 基本的にRATSは，プログラムの最初から順番にコマンド（命令）を読み込み，そのコマンドに従って，特定の作業を実行していくソフトである。
- 個々のコマンドはそれ自身で一連の作業をするようになっている（例えば，LINREGは指定された系列を用いて線形回帰を行い，係数推定値，各種統計量を計算し，表示するコマンドである）。そのため，FORTRANなどに比較して各コマンドの独立性が高いといえる。しかし，各コマンドが完全に独立しているというわけではなく，例えば，LINREGコマンドで回帰分析を行うためには，使用する系列がそれまでにDATAコマンドによって，読み込まれている必要がある，というように作業の流れに従って各コマンドが関連し合って，全体としてまとまった作業をするのである。
- RATSによって回帰分析を行うためには，RATSに，何を，どの様な順序でさせるかを，RATSが理解できる言葉（コマンド）で，記述する必要がある。RATSのプログラムとは，コマンドを作業順序に従って並べた作業手順書（ファイル）である。RATSにプログラムを与えると，RATSはコマンドに従って所定の作業を行うのである。

---

(3) RATSもバージョン・アップして，現在では4.xxとなっているが，ここで，紹介するシステムはバージョン3.02を使用している。

（プログラム１）

```
( 1)*
( 2)*  sample program  消費関数の計測
( 3)*
( 4)*
( 5)
( 6) *分析期間の設定
( 7) CALENDAR_70          ;* 開始期    CAL 70 と書いてもよい
( 8) ALLOCATE_0_85：1     ;* 終了期    0 は常に書くこと，：1 を忘れな
                                       いようにする
( 9)
(10) *データの読み込み
(11) DATA (UNIT = INPUT, ORG = OBS) _70：1_85：1_YY_X1_X2
(12) _86_105_103
        :
        :
(28) 160_190_295
(29) * 新しい系列の作成
(30) SET_LOGY_70：1_85：1_ = _LOG (YY(T))
(31) SET_LOGX1_70：1_85：1_ = _LOG (X1(T))
(32) SET_LOGX2_70：1_85：1_ = _LOG (X2(T))
(33)
(34) * 系列の出力
(35) PRINT (DATES)_70：1_85：1_YY_X1_X2_LOGY_LOGX1_LOGX2
(36)
(37) * 最小２乗法
(38) *(1)Y = a+b X1
(39) LINREG_YY_70：1_85：1_RESI1
(40) #CONSTANT_X1
(41) PRJ_YHAT1_70：1_85：1
(42) PRINT (DATES)_70：1_85：1_YY_YHAT1_RESI1
(43) GRAPH_2
(44) #YY
(45) #YHAT1
(46)
(47) *(2)Y = a+b X1+c X2
(48) LINREG_YY_70：1_85：1_RESI2
(49) #CONSTANT_X1_X2
(50) PRJ_YHAT2_70：1_85：1
(51) PRINT (DATES)_70：1_85：1_YY_YHAT2_RESI2
(52) GRAPH_2
(53) #YY
```

```
(54) #YHAT2
(55)
(56) *(3)LOGY = a+b LOGX1
(57) LINREG_LOGY_70 : 1_85 : 1_RESI3
(58) #CONSTANT_LOGX1
(59) PRJ_YHAT3_70 : 1_85 : 1
(60) PRINT (DATES)_70 : 1_85 : 1_LOGY_YHAT3_RESI3
(61) GRAPH_2
(62) #LOGY
(63) #YHAT3
(64)
(65) *(4)LOGY = a+b LOGX1+c LOGX2
(66) LINREG_LOGY_70 : 1_85 : 1_RESI4
(67) #CONSTANT_LOGX1_LOGX2
(68) PRJ_YHAT4_70 : 1_85 : 1
(69) PRINT (DATES)_70 : 1_85 : 1_LOGY_YHAT4_RESI4
(70) GRAPH_2
(71) #LOGY
(72) #YHAT4
(73)
(74) *(5)Y = a+b X1+c Y(-1)
(75) LINREG_YY_71 : 1_85 : 1_RESI5
(76) #CONSTANT_X1_YY {1}
(77) PRJ_YHAT5_71 : 1_85 : 1
(78) PRINT (DATES)_71 : 1_85 : 1_YY_YHAT5_RESI5
(79) GRAPH_2
(80) #YY
(81) #YHAT5
(82)
(83) *系列の1 2 3ファイルへの出力
(84) OPEN_COPY_SAMPLE. WKS
(85) COPY (FORMAT = WKS, ORG = OBS, DATES)_70 : 1_85 : 1_$
(86) YY_LOGY_YHAT1_YHAT2_YHAT3_YHAT4
(87)
(88) *プログラムの終わり
(89) END
```

(注) ・各行の最初の( )で示された数字はプログラムとは関係ない説明のための行番号である。

・プログラム中の下線(_)は空白を表している。

RATS全体としては140以上のコマンドを有するが，プログラム1では，そのうち11個(CALENDER ALLOCATE DATA SET PRINT LINREG PRJ

GRAPH OPEN COPY END) が使用されているに過ぎない。しかし，この程度のコマンドで通常の回帰分析には十分なのである。

コマンドの説明に入る前に，プログラムの流れには直接関係しない，空白行，注釈行，継続行，複数コマンド行について説明しておく。

(1) 空白行

プログラム1には5行目，9行目のように何も書かれていない行がいくつかある。これらは空白行と呼ばれ，プログラムには関係しない。プログラムを見やすくするためだけにある。

(2) 注釈行

その行における最初の文字がアステリスク（*）で始まる行は注釈行（コメント行）と呼ばれ，プログラムの実行には影響しない。注釈行が空白行と異なる点は，アステリスクの後に自由にコメントなどが書き込める点である。RATS はアメリカ製のソフトなので，本来，日本語は受け付けないが，注釈行には日本語を書き込める。プログラム1では，1—4行，6，10行目などが注釈行である。

(3) 継続行

一つのコマンドが一行（80カラム）に納まらない場合には，そのコマンドが次行に継続することを指定する必要がある。行末に$を書くことによって継続を指定する。RATS のプログラムは1行が80カラム以内で，80を越えた部分は，RATS により無視される。1つのコマンドは1行に記述するのが普通であるが，多数の系列を読み込んだり，出力したりする場合，系列名が1行に納まらない場合がある。このような場合，その行の後ろに$を書き，その行に対する記述を次行に継続することができる。$を何行にも書くことにより，1つのコマンドに対する長い記述が可能となる。プログラム1の85—86行目が継続行の例である。何行にも渡る例として以下のようなものがある。

(例1　継続行およびデータベースからの読み込み[4])

```
OPEN DATA D: ZAISEI.DB
DATA (FORMAT = RATS) /    $
C       CN      DSLB    DUMMY   DUMMY1  EB      $
EH      EI      ELB     EM      EO      ES      $
ET      EX      G       GN      GNP     GNPN    $
IM      IN      LAT     LB      LT   $
LTBC    LTFE    LTIC    LTO     LTP     NTD     $
P       PC      PLAND   RO      RT      SLB     $
T1      T2      T3      TCP     TIN     TLQ     $
TX3     W       WT      YCP     YDP     YP
```

(4)　複数コマンド行（マルチ・ステートメント）

それぞれのコマンドをセミコロン（；）で区切ることにより複数のコマンドを1行に書くことができる。これは通常，短いコマンドの場合やコマンドに簡単な注釈を付ける場合に使用する。プログラム1の7，8行目はコマンドに対する注釈の例である。複数のコマンドを1つの行に記述しようと，1行に1つずつ記述しようと実行上の差異はない。複数コマンド行はSETコマンドを用いていくつもの新しい変数を作成する場合など，プログラムを見やすくするために，よく利用される。

(例2　複数コマンド行および関数)

```
SET Y1 = LOG (X2(T)) ; SET Y2 = EXP (X2(T))
SET Z1 = SIN (X3(T)) ; SET Z2 = COS (X3(T))
```

2.　コマンドの構成要素

コマンドは非常に簡単なものから高度に複雑なものまである。以下に示すコマンド（グラフを画面に表示するコマンドGRAPH）は1つのコマンドを構成する全ての要素を含んでいるので，これを参考にして，コマンドの構成要素に

(4)　これは，筆者が作成した地方財政統計データベースZAISEI.DBからのデータ読み込みの例である。

ついて説明しておく。

```
GRAPH (DATES, KEY = NONE, HEADER = 'Retail Price') 2
# FRFARM
# PRRETAIL
```

(コマンド名) GRAPH[5]

コマンド名(GRAPH)は行の先頭に書かれるものであるが，第1カラムから書き始める必要はない。また，コマンド名は最初の3文字のみがRATSによって識別されるため，GRAPHをGRAと省略することができる。

(オプション) (DATES, KEY = NONE, HEADER = 'Retail Price')

コマンド名の直後に続くのがオプションである。

- オプションは括弧に囲む。ただし，左括弧はコマンド名の直後に書く必要がある。
- 異なったオプションはコンマで区切る。
- コマンドがオプションを必要としない場合やオプションの既定値を変更する必要がない場合にはオプションを付けない。プログラム1の43行目などがオプションがつかない場合の例である。

オプションはそのコマンドをどのように実行するかを指定するものである。プログラム1の35行目のPRINTは系列を出力するコマンドであるが，オプションDATESはデータを日付付きで出力することを指定するものである。

ほとんどのオプションは他の指定をしない限り，前もって割り当てられる既定値を持っている。例えば，コマンドGRAPHはグラフを見やすいものにするために20以上のオプションを持っているが，既定値のみでも，十分見やすいグラフを作成することが可能である。

(5) RATSにおけるGRAPHコマンドは脚注(1)で述べたSIMにより，PC98シリーズでも利用可能である。

(パラメーター) 2

ほとんどのコマンドでパラメーターはオプションの次に書かれる（オプションが無い場合はコマンド名の次)。パラメーターは通常，コマンドが適用される系列や系列要素を指定するものである。

・異なるパラメーター間，オプションとパラメーター間，コマンド名とパラメーター間は1つ以上の空白で区切る。

・パラメーター・フィールドには式も書くことができる。

上の例のGRAPHコマンドでは2という値の単一のパラメーターが書かれていて，これは2つの系列をグラフすることを示している。ほとんどのコマンドは，多くても5，6個の決まった数のパラメーターを持つ。しかしながら，DATAコマンドに代表されるような，ある種の重要なコマンドでは，パラメーター・フィールドに系列名のリストを書くことができ，コマンドが多数のパラメーターを持つことが可能である。しかし，RATSでは単一のコマンドに書くことができるパラメーターの数を200までに制限している。(例1）参照

(補助カード) #PRFARM, #PRRETAIL

ある種のコマンド (GRAPH, LINREGなど) ではコマンド行に付加して補助的情報が必要となる。補助的情報はコマンドをより特定化するためのもので，例えば，LINREGのようなコマンドでは説明変数が補助カードに書かれる。

上の例のGRAPHコマンドでは2行の補助カードがプロットする2つの系列を特定化している (PRFARMとPRRETAIL)。

補助カードはその行の最初の文字として＃を書く。また，補助カードに書く情報が1行に書けない場合には最初の行の行末に$を書いて補助カードを継続することができる。

例：LINREG M1 1948：1 1988：6
#CONSTANT CPR {1 TO 12} CPI {1 TO 12} M1 {1 TO 12} $
IPI {1 TO 12}

3. 主要なコマンド

パソコンに統計的な仕事をさせる場合の手続きは大きく

1. データの入力
2. 計算
3. 結果の出力

の3つに分類できる。FORTRAN でも，BASIC でも，RATS でもこの種の仕事のためにプログラムを書く場合は，データをどの様な方法でパソコンに読み込ませるのか，どの様な計算をさせたいのか，計算結果をどの様な形で表現するかを考えてプログラムを書いていく必要がある。

統計的な仕事のために，RATS は多くのコマンドを用意しているが，それらは次のように分類できる。

A．プログラム設定のためのコマンド (CALENDAR，ALLOCATE，END など)

B．データの入出力関係のコマンド (DATA，PRINT，GRAPH，OPEN，COPY など)

C．計算のためのコマンド (SET，LINREG，PRJ など)

以下，RATS の主要コマンドを説明していくことにする。

A．プログラム設定のためのコマンド

一般的な RATS プログラムは，CALENDAR コマンドで始まり，END コマンドで終わる。また，ALLOCATE コマンドは CALLENDAR コマンドと対で使用され，プログラム全体の分析期間を設定する。

(1) CALENDAR

CALENDAR コマンドはプログラムの先頭に書き，プログラム全体の分析開始期，および使用データの種類を設定する。プログラム1においても，1－6行までは，注釈行，空白行なので，実質的なプログラムの先頭は7行目の CALENDAR である。RATS はパネル・データ，クロスセクショ

ン・データなどに対する統計的分析も可能であるが，よく使用されるのは月次，四半期，年次などの時系列データ (Time Series) に対する統計的分析である。時系列データに対するCALENDARコマンドの書き方は，

CALENDAR a b c

である。ここで，パラメーターa，b，cは以下のような意味を持つ。

a：分析開始年　(西暦は4桁でも下2桁でも可)

b：分析開始期　(年次 = 1　四半期 = 1-4　月次 = 1-12)

c：データの種別 (年次 = 1　四半期 = 4　月次 = 12)

ただし，年次データの場合はb，cが省略可能である。

例：CALENDAR 1955 1 1　1955年から始まる年次データ

CALENDAR 1960 3 4　1960年第3四半期から始まる四半期データ

CALENDAR 1970 8 12 1970年8月から始まる月次データ

コマンド名は最初3文字に，西暦は下2桁に，年次データの場合パラメーターb，cが省略可能なことを考えれば，上の例はそれぞれ，CAL 55, CAL 60 3 4, CAL 70 8 12と省略して書くこともできる。プログラム1の7行目は分析開始が1970年で，年次データを使用することを示している。

(2) ALLOCATE

ALLOCATEコマンドは分析終了期を設定するものである。書き方は

ALLOCATE a b

である。パラメーターの意味は，

a：= 0　(通常の分析ではゼロとしておく)

b：プログラム全体の分析終了期

ここで，予測などを行う場合には最終予測期を指定する必要がある。また，RATS においては，ALLOCATE コマンドを含めて，ほとんどのコマンドにおいて，そのコマンドを適用する期間を指定するが，時点の表記方法に特徴がある。

- データが四半期，月次の場合（CALENDAR コマンドにより区別される）例えば 70：4 とあれば，四半期データの場合は 70 年第 4 四半期を示しており，月次データの場合は 70 年 4 月を示している。
- データが年次の場合も，四半期，月次と表記法を統一している。例えば，70 年を示すには，70：1 と表記する。年次データの場合，：1 には意味がないが，これがないとエラーとなる。

プログラム 1 の 8 行目では，分析終了期が 85 年であることを示している。

(3) END

END コマンドは，プログラムの終わりを示す。END が記述されている行以下は実行されないので，プログラムのテストなどの目的で，プログラムの途中に，一時的に END コマンドを書くことができる。

B．データの入出力関係のコマンド

最も基本的なデータの読み込みのためのコマンドが DATA であり，データの印刷のためのコマンドが PRINT である。OPEN コマンドは他のファイルからデータを読み込んだり，データを他のファイルに書き出したりするために，ファイルをオープンするコマンドである。GRAPH コマンドはデータのグラフを画面に表示するもの，COPY コマンドは PRINT コマンドに比較して，柔軟な出力を得るためのコマンドである。

(4) OPEN，DATA

RATS におけるデータの読み込み方法は大きく 2 つある。

(4-1) プログラムに書き込まれたデータを読み込む

(4-2) 他のファイルからデータを読み込む

RATS の既定値では他のファイルからデータを読み込むようになっている。しかし，データが当該分析にしか利用されない場合や，データ量が少ない場合は，データを直接プログラムに書き込むことが可能である。プログラム 1 はそのような例で，11 行目の DATA コマンドのオプション UNIT = INPUT で，データを次行から書いていることを指定している。

(4-1) プログラムに書き込まれたデータを読み込む

一般的書式は

DATA (UNIT = INPUT, ORG = a) b c s1 s2 . . . .

ここで，オプション a，パラメーター b，c，s 1，s 2 . . . は

a：データ読み込みの方向の選択。OBS か VAR を選択。
プログラム 1 の 3 つの系列(消費支出，可処分所得，資産残高)を読み込む場合，読み込み方についてふたとおり考えられる。

1) OBS (OBServation の略)
最初に 70 年の 3 つのデータを読み込み，次に 71 年の 3 つのデータというように，観測時点毎にデータを読み込む。

2) VAR (VARiables の略) [既定値]
最初に消費支出のデータを 70 年から 85 年まで読み込み，次いで可処分所得のデータというように，系列毎にデータを読み込んでいく。

b，c：データの開始期と終了期。年次データの場合：1 を忘れないよう

s1 . . . .：データの系列名を読み込む系列数だけ並べる。系列名が 1 行 (80 カラム) に納まらない場合は継続行を使用する。例 2 を参照。

(RATS における系列名)

RATS ではデータは系列名によって扱われる。(プログラム 1) で説明すると，消費支出の 70 年から 85 年までの 16 個のデータが，プログラム 1 の 11 行目，DATA コマンドで YY という系列名で読み込まれ，以後，プログラム 1 では消費支出は YY として扱うことになる。同様に可処分所得は X1，資産残高は X2 として，扱うことになる。なお，系列における特定時点のデータ，例えば 75 年の消費支出の値はプログラム内では YY (75：1) と表す。データに対する系列名は，DATA コマンドでデータを読み込む時や，SET コマンドで新しい系列を作成する時に，自由にネーミング可能であるが，以下の制約がある。

1) 長さは 8 文字以内，日本語は使用できない
2) 系列名はアルファベット，数字，_，$ を使用して作成するが，先頭はアルファベットで始める必要がある。
3) RATS の予約語(特定の目的に使用する語，コマンド名，関数名など) は使用できない。特に X，Y，Z は予約語なので，系列名として使用しないように注意すること。

プログラム 1 の使用例 (11 行目)，

DATA (UNIT = INPUT, ORG = OBS) 70：1 85：1 YY X1 X2

では，3 つの系列，名前を YY，X1，X2 としている(系列名として Y を使用していない点に注意)，の 70 年から 85 年のデータを読み込むように RATS に命令している。この時，オプション UNIT = INPUT でデータは次の行から記述していること，ORG = OBS でデータは観測時点毎に記述

していることを指定している。

オプションUNIT = INPUTを指定した場合，データを次行から記述していく必要がある。記述方法についてもオプションで指定可能であるが，既定値はフリー・フォーマットと呼ばれる記述方法である。その方法は簡単で，

1) データを読み込ませたい順番に記述する
2) 個々のデータは1つ以上の空白で区切る
3) 1行に書くデータの個数は80カラム以内であれば制限がない
4) データが1行に納まらない場合は次行に続けるが，継続行のマーク$は必要ない

という規則に従って，データを記述していく。

(4-2) 他のファイルからデータを読み込む

RATSは次のような形式のファイルからデータを直接読み込める。

1) FORTRAN FORMAT (FREE FORMATを含む) ファイル
2) WKS, DIF, PRNファイル
3) RATS FORMATファイル

このうち，2)は表計算ソフトのLotus 1-2-3, Excelなどのファイルからのデータの読み込みである。表計算ソフト側でのデータの記述方法等については英文マニュアルの2．3に述べられている（ただし，日本語Lotus 1-2-3におけるWJ1, WJ2ファイルは直接読み込み不可能なためExcelなどを経由する必要がある)。また，3) のRATS FORMAT形式ファイルは計量モデル分析などに使用するためのRATS特有のデータベースで，大量のデータに対して高速でランダムなアクセスが可能である。RATS FORMATについては，RATSのデータベース機能の節で説明する。RATSで他のファイルからデータを読み込む場合には，DATAコマンド

に先だって，OPEN コマンドで使用ファイル名を RATS に指定する必要がある。例えば，プログラム 1 で 12 行目から 28 行目に記述しているデータを別のファイル（例えば，SAMPLE.DAT）にして，B ドライブの FD に保存しているものとしよう。この時，SAMPLE.DAT (FREE FORMAT ファイル) からデータを読み込むには，プログラム 1 の 11 行目から 27 行目までは以下の 2 行で代替すればよい。

```
OPEN DATA  B: SAMPLE.DAT
DATA (ORG = OBS) 70 : 1  85 : 1 YY X1 X2
```

プログラムにデータを記述する場合は，DATA コマンドのオプション UNIT = INPUT を指定する必要があったが，RATS では外部ファイルからのデータ読み込みが既定値なので，UNIT = オプションは必要ない。また，外部データの場合，その形式 (WKS, DIF 等) を FORMAT = オプションで指定する必要があるが，FREE FORMAT が既定値なため，ここでは指定する必要がない。

2 番目の例として，ドライブ B の FD にある，表計算ファイル LOTUS.WKS から同様のデータを読み込むためには，

```
OPEN DATA B: LOTUS.WKS
DATA (FORMAT = WKS, ORG = OBS) 70 : 1 85 : 1  YY X1 X2
```

のようにする。

最後に，例 1 はドライブ D にある，RATS 形式のデータベース ZAISEI.DB から C から YP までの 47 系列を読み込む命令である。

(5) PRINT

PRINT は RATS の基本的な系列出力のためのコマンドである。一般的な

書き方は，

```
PRINT (DATES) a b s1 s2 . . .
```

であり，パラメーターa，b，s1，s2……の意味は

a，b：出力の期間指定

s1……：出力したい系列名を系列数だけ並べる。系列名が1行（80カラム）に納まらない場合は継続行を使用する。

なお，系列名を省略すると，プログラムの先頭から当該PRINTコマンドまでに読み込まれたり，新たに作成された全ての系列を出力する。

プログラム1では数カ所でPRINTコマンドが使用されている。35行目では，回帰分析に入る前に，読み込んだ系列YY，X1，X2とSETコマンドで作成した系列LOGY，LOGX1，LOGX2の6系列を出力している。統計的分析においてはデータが正しく（意図したように）セットされていることが前提条件であるので，データ・チェックのために分析に入る前に，使用データを出力して確認する習慣をつけるようにすべきである。42，51，60，69，78行目は各計測式ごとに，現実値，計測値，残差を出力する命令である。42行目のPRINTコマンドの出力結果を以下に示す。

ENTRY	YY	1	YHAT1	8	RESI1	7
70：1	86.0000		81.7913		4.20869	
71：1	91.0000		87.0541		3.94588	
		⋮				
84：1	156.000		152.839		3.16079	
85：1	160.000		156.348		3.65225	

(6) GRAPH

回帰分析においてはデータのグラフ化，現実値と推定値のグラフ化などグラフが有効な判断材料となる。RATSのGRAPHコマンドではオプションを指定することによって複雑なグラフまで描けるが，既定値のまま（オプションなし）でも十分に実用的なグラフを作成可能である。GRAPHコマンド（オプションなし）の一般的な書き方は

```
GRAPH  a
#s1
. . . . . .
```

となり，パラメーターa，s1. . . の意味は

a：グラフ化する系列数。次行以下の補助カード（グラフ化する系列数だけある）で系列名を指定する。

s1：グラフ化する系列名

プログラム1の43—45行がGRAPHコマンドの使用例である。43行目で，グラフ化したい系列数が2つであることを指定し，それに続く2行の補助カードで，系列名を指定している。

プログラムの実行時において，GRAPHコマンドがあると，グラフを画面表示し，プログラムは一時停止する。プログラムを続けるにはリターン・キーを押せばよい。また，画面に表示されたグラフのハード・コピーを得るには，グラフが画面に表示されている状態でSHIFTキーを押しながら，COPYキーを押せばよい（プリンターが印字可能な状態であること[6]）。

(7) OPEN，COPY

RATSはそれ自身で，十分に自己完結的なソフトであるが，RATS形式のデータベースの内容や計算結果は他のソフト（FORTRANプログラムや表計算ソフト）で利用可能である。RATSの基本的な出力コマンドはPRINTであるが，より柔軟な出力であるためにCOPYコマンドを用意している。COPYコマンドの使用法は外部ファイルからのデータの読み込みの場合と同様で，まず，OPENコマンドで出力用のファイルを指定する。プログラム1の84—86行目，

```
(84) OPEN_COPY_SAMPLE. WKS
```

(6) 藤本喬雄氏作成のPDS GCOPYを利用した。

(85) COPY (FORMAT = WKS, ORG = OBS, DATES)_70：1_85：1_$
(86) YY_LOGY_YHAT1_YHAT2_YHAT3_YHAT4

は現実値と予測値(YY, LOGY, YHAT1, YHAT2, YHAT3, YHAT4)をLotus 1-2-3用のWKSファイルに出力するものである。(84)行目のOPENコマンドで出力用ファイルとして，SAMPLE.WKSをオープンする。(85)行目のCOPYコマンドで，6系列の70年から85年までの値をファイルに出力する。その際，オプションFORMAT = WKSで出力先のファイルがWKS形式ファイルであること，ORG = OBSで1系列に1列を使用すること，DATESで第1列に日付を出力することを指定している。日本語版1-2-3ではワークシートの内容をWKSファイルとして保存不可能であるが，WKSファイルは直接読み込めるので，このようにして出力したWKSファイルをLotus 1-2-3に読み込んで利用可能である。

SAMPLE.WKSを日本語Lotus 1-2-3に読み込んだ時のワークシートは以下のようになる。

	YY	LOGY	YHAT1	YHAT2	YHAT3	YHAT4
70：1	86	4.4543472	81.791308	85.893159	4.4236430	4.4475804
71：1	91	4.5108595	87.054115	90.451696	4.4819064	4.5045532
			：			
83：1	152	5.0238805	148.45353	151.49544	4.9945738	5.0211702
84：1	156	5.0498560	152.83920	156.77739	5.0231443	5.0530858
85：1	160	5.0751738	156.34774	161.67038	5.0454530	5.0818069

C．計算のためのコマンド

RATSは最も基本的な線形回帰からウエイト付き，2段階，3段階，非線形，制約条件付き等の最小2乗法，リッジ回帰，分布ラグ，SUR，各種回帰に関するテスト，Box-Jenkinsモデル，時系列分析，同時方程式体系，モンテカルロ実験，スペクトラル分析，カルマンフィルターなど回帰分析に関連する種々の統計的手法を網羅している。また，新しい推定法の開発用のコマンドも用意されている。RATSの基本的なデータの入出力をマスターした後は，自分の目的

とする分析手法に関するコマンドの使用法に関して，英文マニュアルの該当箇所を参照にすることになる。ここでは，最も基本的な線形回帰のためのLINREGコマンド，計測式から推定値を計算するPRJコマンド，変数変換のためのSETコマンドを説明することにする。

(8) LINREG

一般的な書き方は

```
LINREG a b c d
#list of explanatory variables
```

であり，ここで，パラメーターは，

a：被説明変数の系列名

b，c：計測の開始期，終了期

d：残差系列に付ける名前

補助カードに説明変数の系列名を記述する。この時，次の点に注意すること。

- 説明変数間は1つ以上の空白で区切る。
- 定数項を含む場合は，補助カードにCONSTANTと記述する。
- 説明変数にラグ付き変数（ZZt-1のように今期以前の値）を含む場合の表記法はZZ｛1｝と｛　｝の中にラグ数を書く。また，説明変数として，ZZt-1, ZZt-2, ZZt-3というように連続したラグ変数を含む場合はZZ{1 TO 3}というように一括して記述可能である。ラグ変数を含む場合には計測期間がずれることにも注意すること。

プログラム1では5本のモデルが計測されている。なお，LINREGコマンドに使用する系列は，当該コマンドより前にDATAコマンドによって読み込まれているか，SETコマンドにより作成されている必要がある。プログラム1では，

YY ：消費支出 11行目のDATAコマンドで読み込み
X1 ：可処分所得 11行目のDATAコマンドで読み込み
X2 ：資産残高 11行目のDATAコマンドで読み込み
LOGY ：LOG (YY) 30行目のSETコマンドで作成
LOGX1：LOG (X1) 31行目のSETコマンドで作成
LOGX2：LOG (X2) 32行目のSETコマンドで作成

となっている。

プログラム1での最初のLINREGコマンドは，39—40行目，

```
(39) LINREG_YY_70：1_85：1_RESI1
(40) #CONSTANT_X1
```

であり，70年から85年までのデータを用いて，YY(消費支出)をCONSTANT（定数項）とX1（可処分所得）で説明したときの，最小自乗推定値を求めるものである。また，残差（現実値と推定値の差）をRESI1という系列名で作成することも指定している。単純回帰の例である。

このLINREGコマンドに対する出力結果は以下のようになる。

```
DEPENDENT VARIABLE  1     YY
FROM       70：1 UNTIL       85：1
TOTAL OBSERVATIONS       16       SKIPPED/MISSING          0
USABLE OBSERVATIONS      16       DEGREES OF FREEDOM      14
R**2                   .97159701  RBAR**2              .96956822
SSR                 228.42931     SEE                4.0393538
DURBIN-WATSON        .35438084
Q(    8) =    36.6117        SIGNIFICANCE LEVEL   .135643 E-04
NO.      LABEL        VAR  LAG  COEFFICIENT   STAND. ERROR
***      *******      ***  ***   ***********   ***********
  1       CONSTANT      0    0    -10.30782     6.301549
  2       X1            2    0      .8771346    .4008124 E-01
T-STATISTIC
***********
-1.635760
 21.88392
```

RATSはアメリカ製のソフトであるので当然メッセージも英語である

が，筆者のシステムではVZエディターのマクロ機能を利用してメッセージを日本語にできるようにしている。計測結果がVzエディターに表示されている状態でCTRLキーを押しながら¥キーを押すと，回帰結果に対するメッセージが日本語に変換され，上の出力例は以下のように日本語化される。

```
被説明変数              1    YY
開始期  70     終了期     85
全観測データ数          16        欠損値数          0
使用観測値数            16        自由度               14
決定係数          .97159701       修正済決定係数    .96956822
残差平方和        228.42931       標準誤差          4.0393538
D—W比             .35438084
Q(    8) =    36.6117          有意水準          .135643 E-04
NO.  変数名  VAR   ラグ      回帰係数        標準偏差          t 値
***  ******* ***    ***     ************   ************    ***********
  1  定数項    0      0    −10.30782       6.301549        −1.635760
  2  X1        2      0      .8771346      .4008124 E-01    21.88392
```

計測結果の評価は計量経済学の重要なテーマであるが，ここでは，定数項に対する推定値が−10.31（対応するt値が−1.64）で，X1（可処分所得）に対する係数推定値が0.877（t値が21.9）であること，また，決定係数が0.971，DW比が0.354であることを指摘するだけにしておく。

プログラム1の2番目のLINREGコマンドは48—49行目，

```
(48) LINREG_YY_70：1_85：1_RESI2
(49) #CONSTANT_X1_X2
```

であり，YY（消費支出）をCONSTANT（定数項）とX1（可処分所得）とX2（資産残高）で説明しようとする，重回帰の例である。LINREGコマンドの使用法において，重回帰が単純回帰と異なる点は，補助カードに書かれる系列数だけである。また，この例では残差系列をRESI2という名前で作成しているが，残差を計算しない場合は書く必要はないし，ここでRESI1という名前を指定すると39—40行目のLINREGコマン

ドで作成した残差が削除され，今回の残差がRESI1という名前で作成される。このLINREGコマンドに対する出力結果は以下のようになる。

```
被説明変数                1     YY
開始期     70     終了期     85
全観測データ数           16     欠損値数              0
使用観測値数             16     自由度                13
決定係数          .99691661     修正済決定係数     .99644224
残差平方和        24.797971     標準誤差           1.3811358
D－W比          1.44463173
Q (    8) = 10.5789          有意水準          .226720
NO. 変数名  VAR ラグ    回帰係数        標準偏差          t 値
***  *******  ***  ***  ************  ************  ************
  1  定数項     0    0   22.13703      3.808331       5.812790
  2  X1         2    0    .3889630     .4919574 E-01  7.906436
  3  X2         3    0    .2224759     .2153263 E-01  10.33204
```

計測結果の見方は単純回帰の場合と同じであるが，重回帰の場合は決定係数ではなく，自由度修正済み決定係数を使用することに注意すること。

3番目，4番目のLINREGコマンドは57—58，66—67行目，

```
(57) LINREG_LOGY_70：1_85：1_RESI3
(58) #CONSTANT_LOGX1

(66) LINREG_LOGY_70：1_85：1_RESI4
(67) #CONSTANT_LOGX1_LOGX2
```

で，対数線形回帰の例である。この例でわかるように，DATAコマンドで読み込んだ系列をそのままではなく，変換して回帰分析に使用する場合にはLINREGコマンドを使用する前にSETコマンドで新たな系列として作成する必要がある。

5番目のLINREGコマンドは75—76行目，

```
(75) LINREG_YY_71：1_85：1_RESI5
(76) #CONSTANT_X1_YY {1}
```

で，YY（消費支出）をCONSTANT（定数項）とX1（可処分所得）とYY{1}前期の消費支出で説明していて，ラグ付き変数を含む場合であ

る。この例からも分かるように，ラグ変数は系列名〔ラグ数〕で表す。また，計測期間が71年から85年までとラグ数に応じてずれることに注意すること。このLINREGコマンドに対する出力結果は

```
被説明変数              1     YY
開始期     71     終了期     85
全観測データ数          15        欠損値数              0
使用観測値数            15        自由度                12
決定係数          .98695486        修正済決定係数     .98478067
残差平方和       82.859245        標準誤差          2.6277247
D—W比         1.83162493
Q(    7) = 5.67337          有意水準          .578363
NO.   変数名  VAR  ラグ   回帰係数          標準偏差          t 値
***   *******  ***   ***   ************   ************   ************
  1    定数項    0     0    3.754611       6.690282        .5612036
  2    X1        2     0     .1981436       .1728891       1.146074
  3    YY        1     1     .7552592       .1812600       4.166718
```

となり，説明変数の3行目（YYの行）のラグの欄が1となっておりYYの前期が説明変数であることがわかる。

(9) PRJ

PRJコマンドは一番近いLINREGコマンドによる推定式から推定値を計算するものである。書き方は

PRJ　a b c

で，パラメーターの意味は

a：推定値に付ける系列名

b，c：推定値を計算する期間

である。プログラム1では計測式ごとにPRJコマンドを用いて推定値を計算している。例えば，41行目の

PRJ_YHAT1_70：1_85：1

では，すぐ前のLINREGコマンド（39—40行目）からの計測式から70年から85年までの推定値を計算し，その系列にYHAT1という名前を付けている。PRJコマンドは回帰式から予測値を計算する場合にも利用するが，この場合，予測期間について説明変数系列のデータが準備されている必要がある。また，PRJコマンドで作成された系列やLINREGコマンドで作成された残差は，そのままでは見ることはできず，PRINTコマンドなどで出力する必要がある。

プログラム1では5本の回帰式を計測しており，各回帰式ごとにLINREG, PRJ, PRINT, GRAPHという4つのコマンドが使用されているが，回帰分析ではいつでも，これらのコマンドを使用するというわけではない。回帰分析のコマンドはLINREGであり，それのみで十分な結果を得ることができる。だから，推定値を計算する必要がないのならPRJコマンドは必要ないし，現実値，推定値，残差の値を出力する必要がないならPRINTコマンドは必要ないし，グラフが必要ないなら，GRAPHコマンドは必要ない。

(10) SET

SETコマンドは，それまでに存在する系列を使用して，新しい系列を作成するためのものである。一般的な書き方は，

SET a b c ＝計算式

で，ここで，パラメーターの意味は

a：計算値に付ける系列名

b，c：計算期間

＝の後に計算式を書くが＝の前後には必ず1つ以上の空白を置くこと。

(計算式の書き方)

・定数，既に存在する系列，+，−，＊(乗算)，/ (割り算)，( )，＊＊(べき乗)，関数を使用する。

・既に存在する系列の表記法は，例えば系列名が ZZZ とすると

今期の値　ZZZ (T)

前期の値　ZZZ (T-1)

ラグの値　ZZZ (T-k)　k：ラグ数

特定期の値　ZZZ (70：1)　70 年の値

となる。

・次のような関数が用意されている（一部)。

LOG　自然対数　EXP　自然対数の逆数　SQRT　ルート

SIN　サイン　COS　コサイン

関数の使用法は，関数名（式）である。

プログラム 1 では，30—32 行目で，SET コマンドが使用されている。例えば，

(30) SET_LOGY__70：1_85：1_ = _LOG (YY(T))

では，YY (消費支出) の自然対数を計算し，LOGY という名前を付けている。以下，計算式の書き方の例を挙げておく。

1) 前期と今期の平均

= (ZZZ (T)+ZZZ (T-1))/2

2) 成長率

= (ZZZ (T)−ZZZ (T-1))/ZZZ (T-1)

3) 指数化

= ZZZ (T)/ZZZ (80：1)＊100　80 年を 100 とした指数化

SET コマンドを使用する場合は以下の点に注意すること。

・= の前後は 1 つ以上の空白をおくこと。

・(,) の対応に注意すること。

・系列を使用する時は，T を忘れないように。

・ラグ変数を使用する時は，計算期間に注意すること。

(11) STATISTICS

(プログラム1)には使用されていないが，STATISTICS コマンドを用いると系列の平均，分散を簡単に得ることが可能である。書き方は

STATISTICS sl... a b

であり，パラメーターの意味は

sl...：平均，分散を計算する系列名

a，b：計算期間

である。

4. プログラムの簡略化

RATS プログラムは実行に関係しない，空白行，注釈行の削除，複数コマンド行の多用により短くすることは可能であるが，それをすると理解しづらいプログラムになりがちである。しかし，次の2点を利用すると，簡単にプログラムを簡略化することができ，入力ミスを少なくすることができる。

(1) コマンド名の簡略化

先に述べたように，RATS ではコマンド名は先頭の3文字しか意味が無いので，英語に弱い人は，STATISTICS と打つよりは STA と打った方が間違いが少なくなる。

(2) 既定期間の使用

RATS ではコマンド毎にその適用期間（計算期間）を指定するが，既定期間を使用することにより，これを簡略化することが可能である。

プログラムの CALLENDAR，ALLOCATE コマンドで設定した期間が最初の既定期間となる。例えば，プログラム1では70年から85年までが，最初の既定期間となる。以後，SMPL コマンドで既定期間を変更しない限り，コマンドの適用範囲が既定期間と同じ時は，コマンドにおけ

る開始期と終了期の指定を/で代替できる（/の前後は必ず1つ以上の空白を置くこと)。

例えば，11行目のDATAコマンドは読み込み期間が既定期間と同じであるので，

DATA (UNIT = INPUT, ORG = OBS) / YY X1 X2

と書ける。プログラム1では，70：1 85：1の部分をすべて，/に変えることができる。また，既定期間の利用に際しては以下の省略が可能である。

・SETコマンドなどでは計算期間が既定期間と同じ場合には期間指定を省略できる。例えば，30行目は

SET LOGX = LOG (YY(T))

書くことが可能である。

・コマンドの最後のパラメーターが期間指定で，それが既定期間と同じ場合には期間指定を省略できる。例えば，41行目のPRJコマンドは最後のパラメーターが期間指定70：1 85：1で，これは既定期間と同じなので，41行目はPRJ YHAT1と書くことが可能である。また，39—40行目のLINREGコマンド，

(39) LINREG_YY_70：1_85：1_RESI1
(40) #CONSTANT_X1

においても，もし，残差を計算する必要がないなら，RESI1は必要なく，その場合，コマンド行（39行目）の最後のパラメーターは期間指定70：1 85：1となり，この期間は既定期間と同じなので，省略可能で，39行目はLINREG YYと書くことが可能である。

(SMPLコマンド)

既定期間はSMPLコマンドにより変更できる。

SMPL a b

a，b：既定期間の開始期，終了期

SMPLコマンドで既定期間を変更すると，次のSMPLコマンドで再び

既定期間を変更するまで，RATS コマンドに対する既定期間が，このコマンドで指定した期間となる。ただし，a，b で指定できる期間は CALLENDAR, ALLOCATE コマンドで設定した期間内である必要がある。

プログラム 1 をコマンド名の簡略化と既定期間を利用して書き直すと以下のようになる。

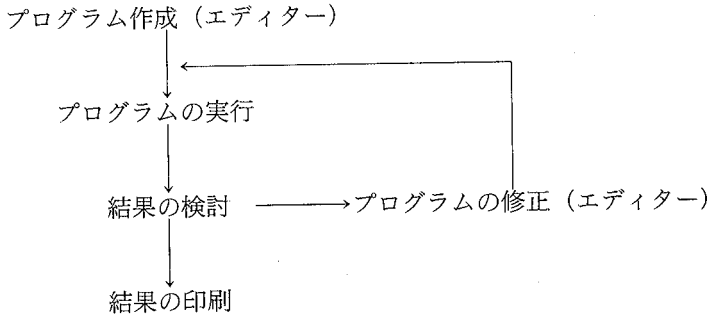
```
*
*  sample program      消費関数の計測
*  プログラム中の_は空白を表す

* 分析期間の設定
CAL 70          ; * 開始期      CAL 70 と書いてもよい
ALL 0 85:1   ; * 終了期      0 は常に書くこと，:1 を忘れないように
* データの読み込み
DATA (UNIT = INPUT, ORG = OBS) / YY X1 X2
 (数値データの記述はプログラム 1 と同じ)
* 新しい系列の作成
SET LOGY = LOG (YY(T))
SET LOGX1 = LOG (X1(T))
SET LOGX2 = LOG (X2(T))

* 系列の出力
PRINT (DATES) / YY X1 X2 LOGY LOGX1 LOGX2

* 最小 2 乗法
* (1) Y = a+b X1
LIN YY / RESI1
#CONSTANT X1
PRJ YHAT1
PRINT (DATES) / YY YHAT1 RESI1
GRA 2
#YY
#YHAT1

* (2) Y = a+b X1+c X2
LIN YY / RESI2
#CONSTANT X1 X2
PRJ YHAT2
PRINT (DATES) / YY YHAT2 RESI2
```

```
GRA 2
#YY
#YHAT2

* (3) LOGY = a+b LOGX1
LIN LOGY / RESI3
#CONSTANT LOGX1
PRJ YHAT3
PRINT (DATES) / LOGY YHAT3 RESI3
GRA 2
#LOGY
#YHAT3

* (4) LOGY = a+b LOGX1+c LOGX2
LIN LOGY / RESI4
#CONSTANT LOGX1 LOGX2
PRJ YHAT4
PRINT (DATES) / LOGY YHAT4 RESI4
GRA 2
#LOGY
#YHAT4

* (5) Y = a+b X1+c Y(-1)
LIN YY 71 : 1 85 : 1 RESI5
#CONSTANT X1 YY {1}
PRJ YHAT5 71 : 1 85 : 1
PRINT (DATES) 71 : 1 85 : 1 YY YHAT5 RESI5
GRA 2
#YY
#YHAT5

* 系列の1 2 3ファイルへの出力
OPEN COPY SAMPLE.WKS
COPY (FORMAT = WKS, ORG = OBS, DATES) / YY LOGY YHAT1 $
YHAT2 YHAT3 YHAT4

* プログラムの終わり
END
```

5. RATSプログラムの基本的な実行方法

RATSプログラムを実行するには，予め，エディターでプログラムを作成す

る必要がある[7]。また，出力結果に対する検討もエディター画面で行い，エラーの修正等を行い，再びRATSを実行し，最終的に満足できる（エラーのない）結果を面画で確認の上，プリンターに出力するというのが一般的なやり方である。これを図に書くと以下のようになる。



ここで，重要なことは

・作業ではRATSとエディターを行ったり来たりすること。

・エディターでは結果ファイルを見て，プログラムファイルを修正すること。

である。以上のことは，この種のソフトに馴染んでいない人が，よく戸惑う点なので注意を要する。

RATSプログラムの基本的な実行方法は以下の2通りである（この他に対話型の実行，統合環境における実行もある）。なお，以下の説明ではカレント・ディレクトリまたは，パス指定したディレクトリにRATS.EXE等のRATS関係のファイルが存在すると仮定している。

1) RATS〈Program file〉

例えば，ドライブBのFDにSAMPLE.PRGというRATSプログラムファイルがある時，

(7) RATSにはRATSEDITと呼ばれるエディターも内蔵された対話型使用法もあるが，筆者のシステムは他のソフトとのデータの互換性とメッセージの日本語化などを考慮して，Vzエディターと組み合わせたシステムとした。

RATS B: SAMPLE. PRG

とすると，RATSが起動し，プログラムを実行し，結果を画面に表示していく。この使用法では，結果が画面に表示されるため，表示後の検討ができないという大きな欠点があるため，あまり利用されない。しかし，プログラムのエラー等によりRATSが暴走し，次の2)の方法による結果ファイルが作成されない場合に，暴走の原因を探る手段として有効である。

2) RATS 〈Program file〉〈Output file〉

これが，一般的な使用法である。プログラムファイルを読み込んで，結果をアウトプット・ファイルに出力する。例としては

RATS B: SAMPLE. PRG B: SAMPLE. OUT

がある。この方法では実行中には画面には何も表示されなく，実行終了後，エディターでB: SAMPLE. OUTを見ることになる。

上の2）番目の方法が一般的な実行方法であるが，これでも初心者には煩雑なので，エディターをVzエディターに特定化して，作業の一部をBATファイルにしたRATSシステムFDを作成し，教育用にはこれを使用する。

6. RATSシステムFDの使用法

1) AドライブにRATSシステムFD，BドライブにFORMAT済みFD(自分のFD) を入れパソコンを起動する。

2) 画面にA>が表示されたら，B：でBドライブにする。プロンプトがB>になっていることを確認。

3) Vzエディターでプログラムの作成あるいは修正をする。

新規プログラムの作成

VZ プログラムファイル名

として，Vzを起動する。このときプログラムファイル名は任意であるが

'英数字8文字以内'+'.PRG'

とすること。例えば，TEST. PRG SAMPLE. PRG 156. PRGのように，

ファイル名の最後は .PRG とすること。

既存プログラムの修正

VZ 既存プログラム名

として，Vz を起動する。プログラム名を忘れた場合には Vz だけで起動し，ファイル名入力でリターンのみとすると，ファイル一覧が表示されるので目的のファイルにカーソルを合わせてリターンを押し，選択する。

プログラム作成あるいは修正後，ファイルを保存して Vz エディターを終了する。

4) プログラムの実行

RUN プログラムファイル名

と打ち込む。ただし，プログラムファイル名の最後の .PRG は付けないこと。例えば，プログラムファイルが SAMPLE.PRG である場合，

RUN SAMPLE

とする。

5) 計算結果の表示

実行が終了すると，自動的にプログラムと計算結果が Vz に読み込まれ，画面は計算結果の Vz 画面となる。この画面で，

(f・2 キーを押すと)

プログラムが表示され，再び f・2 キーで計算結果の表示とトルグする。

(f・4 キーを押すと)

画面が上下に 2 分割され，上画面にプログラム，下画面に計算結果が表示され，カーソルは下画面で点滅しているはずである。

6) エラーの修正

エラーの修正は下画面の計算結果を見ながら，エラー箇所を探し，付録の Error and Warning メッセージ表を参照して，上画面のプログラムの該当箇所を修正する。なお，上下画面のカーソルの移動は f・2 キーで行う。エラー修正後の再実行はファイルを保存して Vz を終了し，4)に戻る。

7) メッセージの日本語化

計算結果が正しい（エラーがない）ことを確認して，Vz 画面に計算結果が表示されている状態でCTRL キーを押しながら¥キーを押すとLINREG コマンドに対する出力結果が日本語化される（LINREG コマンドが多用されている場合には変換に時間がかかる場合がある）。

8) 計算結果の印刷[8]

1. Vz 画面に計算結果が表示されていることを確認
2. プリンターが接続されていることの確認
3. 印刷したい部分の先頭にカーソルを移動
4. CTRL キーを押しながらBキーを押す
5. カーソルを印刷したい部分の最後まで移動
6. CTRL キーを押しながら，Kキー，Oキーを押す

これで，プリンターに印刷が開始される。

9) 作業の終了

Vz を終了させ，プロンプトB〉を表示させる。FD ユニットのランプが点滅していないことを確認してFD を取り出す。メイン・スイッチ，プリンター・スイッチをOFF にする。

7. エラーへの対処

最初からエラーのないプログラムを作成することは至難の技である。プログラミングの上達はエラーに対して冷静で，論理的な対応ができるかどうかにある。以下，RATS のエラーへの対処について述べる。

(1) ENV コマンドの利用

初心者が最初に戸惑うのは，RATS からのエラー・メッセージの多さである。プログラムの先頭付近では，分析期間の設定（CALENDAR, ALLOCATE）とか，データの読み込み（OPEN, DATA）などの分析の基礎とな

(8) Vz エディターのマクロによるPDS を利用。

るコマンドが使用され，それらのコマンドの記述にエラーがあると，その後の分析が無効となり，エラーの洪水となる。さらにエラーがエラーを呼び，プログラムの暴走となりかねない。エラーの洪水，プログラムの暴走を予防するためにENVコマンドでエラーが一定数を越えるとプログラムの実行を停止することができる。ENVコマンドの書き方は，

ENV ERR = a WAR = b

で，a，bは，

a：エラーの最大数

b：警告の最大数

である。普通a = 10, b = 50程度にしておけばよい。ENVコマンドはCALENDARコマンドの前に置き，例えばプログラム1では，

```
ENV ERR = 10 WAR = 50
CALENDAR 70
ALLOCATE 0 85 : 1
```

のように使用する。

(2) エラー・メッセージの見方

例えば，プログラム1の11行目が

```
DATA ((UNIT = INPUT, ORG = OBS) 70 : 1 85 : 1 YY X1 X2
```

と間違えた場合（左カッコが一つ余分）の実行後のエラー・メッセージは

```
DATA ((UNIT = INPUT, ORG = OBS) 70 : 1 85 : 1 YY X1 X2
****ERROR DATUM  INSTRUCTION OR OPTION :
    (UN  ****                                    1)
****ERROR 41 このオプションは識別できません。      2)
```

となり[9]，1)の下線部((UN)がエラーの箇所を示し，2)がエラーに対するメッセージである。この例では，「オプション(UNが識別できない」とい

(9) 2)のエラーメッセージは本来英語で表示されるが，筆者のシステムでは，これも日本語化している。

うものである。このエラーを論理的に考えると，コマンド名(DATA)の後に(があるので，RATSは次にオプションが書かれていると判断する。しかし，最初の(の次は(UNITとなっているので，RATSはこれをオプションとして識別できないのである。このように，エラー・メッセージはエラーそのものを示すものではなく，エラーに対するヒントを与えるものであることに注意すること。

(3) エラーが1つ起こると，それが原因して，いくつかのエラーを連続することが多い。よって，エラーの処理は最初のエラーに重点を置くようにして最初のエラーを修正後，プログラムを再実行し，1つずつエラーを少なくする習慣をつけるようにすること。

(4) 初心者がよく起こすエラーに次のようなものがある。

- コマンド名，オプション名，系列名などのタイプミス
- 空白が必要なところに空白がない
- コロン（：）とセミコロン（；）の区別
- コンマ（,）とピリオド（.）の区別
- 年次データにおける：1
- =,/の前後のスペース
- SETコマンドの右辺の系列名における(T)の付け忘れ
- カッコの対応関係

III RATSと他のソフトのデータの相互利用

1. RATSの結果の「一太郎」での利用

一太郎はテキストファイルを読み込むことが可能なので，これを利用する。ただし，RATSの出力ファイルは特に変更しない限り，1行80カラムになっている点に注意すること。以下，RATSの出力ファイルがSAMPLE.OUTとしてBドライブのFDにあるとして，これを「一太郎」に読み込む手順を説明する。

1) 「一太郎」を起動する。

2) メニューの「印刷」,「スタイル」で一行の文字数を 80 に変更する。

3) メニューの「ファイル」,「読み込み」を選択し，ファイル名を B: SAMPLE. OUT と入力する。

4) 印刷したときの見やすさを考えて，RATS の結果部分を 1/2 改行にしておくとよい。

2. RATS と Lotus 1-2-3

(1) RATS のデータを Lotus 1-2-3 で利用

COPY コマンドを利用することにより，RATS のデータを Lotus 1-2-3 で読み込み可能なファイル形式で出力可能である。RATS における 2 つの系列 X1, X2 を Lotus 1-2-3 形式ファイル TEST. WJ2 として保存したい場合は，

```
OPEN COPY B: TEST. WJ2
COPY (FORMAT = WKS, ORG = OBS, DATES) / X1 X2
```

とすればよい。

(2) Lotus 1-2-3 のデータを RATS で利用

RATS はアメリカ製のソフトのため，日本語 Lotus 1-2-3 の WJ2 形式のファイルを直接読み込むことはできないが，Lotus 1-2-3 のファイル出力を利用することにより，Lotus 1-2-3 のデータを利用することが可能である。

3. 東洋経済マクロデータファイルの利用

東洋経済新報社のマクロデータファイルは日本の主要経済データを 1965 年から最近年次までを FD に収録したものである。

(1) ECONOMATE マクロデータファイルの変換について

ECONOMATE マクロデータファイルは N 88-BASIC で作成されたラ

ンダムアクセスファイルである。その構造はデータファイルに付属しているプログラム DISP. DAT を解析することにより判明する。ここでは，DISP. DAT を修正することにより，RATS 形式のデータファイルを作成することにした。その手続きは以下の通りである。

・マクロデータファイル—＞シーケンシャルファイル

先ず，マクロデータファイルを ECONOMATE 以外で利用可能にするためのプログラムを作成した。TORATS. BAS のマクロデータファイルからのデータの読み込み部分は DISP. DAT を参考にした。

・シーケンシャルファイル—＞RATS 形式データベース

上で作成したシーケンシャルファイルから RATS 形式データベースを作成するための RATS プログラムを作成した。

(2) RATS での利用

RATS 形式の ECONOMATE データファイル MACRO. DB を用いて RATS で分析をする場合は

1. MACRO. DB ファイルを自分の FD にコピーする。
2. プログラムのデータ読み込みの部分を以下のようにする。

```
OPEN DATA B: MACRO. DB.
DATA (FORMAT = RATS) 1965：1 1987：1
```

使用する系列名のリスト

なお，系列名の一部は RATS により禁止されているので，変数名が変わっているものがあることに注意する必要がある。

以下は RATS での利用例のプログラムである。

```
CAL 1965
ALL 0 1987：1
OPEN DATA B: MACRO. DB
DATA (FORMAT = RATS) / HCMD CPIMC CPI PC YDP
PRINT (DATES)
SMPL 1970：1 1987：1
SET MEDICAL = HCMD (T)/CPIMD (T) * 100
SET RPRICE = CPIMD (T) / CPI (T)
```

```
SET YD = YDP (T) / PC (T) * 100
LINREG MEDICAL / RESI
#CONSTANT YD
LINREG MEDICAL / RESI2
#CONSTANT YD RPRICE
PRINT (DATES) / MEDICAL RPRICE RESI
END
```

(3) 日本語 Lotus 1-2-3 での利用

RATS の WKS ファイルへの系列出力の機能を利用すれば，MACRO. DB の系列を WKS ファイルに出力し，日本語 Lotus 1-2-3 での使用が可能となる。

この目的で作成したプログラムが TOWKS. PRG であり，利用に際して適当に変更して利用する。

```
*
* TOWKS. PRG
*                                    Program by T. Ohno
CAL 1965 1 1
ALL 0 1987 : 1
OPEN DATA B: MACRO. DB
DATA (FORMAT = RATS) / $
GNP CP CG IH IP IG JP JG EX M
PRINT (DATES)
OPEN COPY B: SAMPLE. WJ2
COPY (FORMAT = WKS, ORG = OBS, DATES) / $
GNP CP CG IH IP IG JP JG EX M
END
```

参 考 文 献

大野拓行「新しい経済分析システムの提唱」『香川大学経済論叢』，第 60 巻　第 1 号，1987 年
松本和幸「パソコンによる経済分析の勧め」『ESP』，№271，1994 年
学術情報センター　『統計データに関する利用動向及び需要調査　報告書』，1994 年
ビレッジセンター　『VZ Editor Version 1.5 ユーザーズマニュアル』，1989 年
Thomas A. D. 'USER'S MANUAL. RATS Version 3.02' VAR Econometrics, Inc. 1989.